

IBM SPSS Modeler 18.5 Python Scripting
和自動化手冊



附註

在使用本資訊及其支援的產品之前，請先閱讀第 417 頁的『[注意事項](#)』中的資訊。

產品資訊

此版本適用於版本 18 版次 4 修訂版 IBM® SPSS Modeler 及所有後續版次與修訂版，除非新版中另有指示。

© Copyright International Business Machines Corporation .

目錄

第 1 章 Scripting 和 Scripting 語言.....	1
Scripting 概觀.....	1
Script 類型.....	1
串流 Script.....	1
串流 Script 範例：訓練神經網路.....	2
Jython 代碼大小限制.....	3
獨立式 Script.....	3
獨立式 Script 範例：儲存和載入模型.....	3
獨立式 Script 範例：產生功能選項模式.....	4
SuperNode Script.....	4
SuperNode Script 範例.....	5
串流中的迴圈和條件執行.....	5
串流中的迴圈.....	6
串流中的條件執行.....	8
執行和中斷 Script.....	9
尋找/取代.....	10
第 2 章 Scripting 語言.....	13
Scripting 語言概觀.....	13
Python 和 Jython.....	13
Python Scripting.....	13
作業.....	13
清單.....	14
字串.....	15
備註.....	16
陳述式語法.....	16
ID.....	17
代碼區塊.....	17
將引數傳送給 Script.....	17
範例.....	18
數學方法.....	18
使用非 ASCII 字元.....	20
面向物件的程式設計.....	21
定義類別.....	21
建立類別實例.....	21
向類別實例新增屬性.....	21
定義類別屬性和方法.....	22
隱藏變數.....	22
繼承.....	22
第 3 章 在 IBM SPSS Modeler 中進行 Scripting.....	25
Script 類型.....	25
串流、SuperNode 串流和圖表.....	25
串流.....	25
SuperNode 串流.....	25
圖表.....	25
執行串流.....	25
Scripting 環境定義.....	26
參照現有節點.....	26
尋找節點.....	26

設定內容.....	27
建立節點以及修改串流.....	28
建立節點.....	28
鏈結和取消鏈結節點.....	29
匯入、替換和刪除節點.....	30
遍訪串流中的節點.....	30
清除或除去項目.....	31
取得節點的相關資訊.....	31
第 4 章 Scripting API.....	35
Scripting API 簡介.....	35
範例 1: 使用自訂過濾器搜尋節點.....	35
範例 2: 容許使用者基於其專用權獲取目錄或檔案資訊.....	35
meta 資料: 有關資料的資訊.....	36
存取產生的物件.....	38
處理錯誤.....	39
Stream、Session 及 SuperNode 參數.....	40
廣域值.....	43
使用多個串流 - 獨立式 Script.....	44
第 5 章 Scripting 技巧.....	45
修改串流執行.....	45
對節點執行迴圈.....	45
存取 IBM SPSS Collaboration and Deployment Services 儲存庫 中的物件.....	45
產生經過編碼的密碼.....	47
Script 檢查.....	48
從指令行執行 Scripting.....	48
與先前版本的相容性.....	48
存取串流執行結果.....	48
表格內容模型.....	49
XML 內容模型.....	51
JSON 內容模型.....	52
欄統計資料內容模型和成對統計量內容模型.....	53
第 6 章 指令行引數.....	57
呼叫軟體.....	57
使用指令行引數.....	57
系統引數.....	58
參數的引數.....	59
伺服器連線引數.....	60
IBM SPSS Collaboration and Deployment Services 儲存庫 連線引數.....	61
IBM SPSS Analytic Server 連線引數.....	61
組合多個引數.....	62
第 7 章 內容參照資訊.....	63
內容參照概觀.....	63
內容語法.....	63
節點和串流內容範例.....	64
節點內容概觀.....	65
通用節點內容.....	65
第 8 章 串流內容.....	67
第 9 章 來源節點內容.....	71
來源節點一般內容.....	71
asimport 內容.....	78

cognosimport 節點內容.....	79
databasenode 內容.....	81
datacollectionimportnode 內容.....	83
excelimportnode 內容.....	86
extensionimportnode 內容.....	88
fixedfilenode 內容.....	90
gsdata_import 節點內容.....	94
jsonimportnode 內容.....	94
sasimportnode 內容.....	94
simgennode 內容.....	95
statisticsimportnode 內容.....	97
tm1odataimport 節點內容.....	97
tm1import 節點內容 (已淘汰)	98
twcimport 節點內容.....	99
userinputnode 內容.....	100
variablefilenode 內容.....	101
xmlimportnode 內容.....	106
第 10 章 記錄作業節點內容.....	107
appendnode 內容.....	107
aggregatenode 內容.....	107
balancenode 內容.....	108
cplexoptnode 內容.....	109
derive_stbnode 內容.....	111
distinctnode 內容.....	113
extensionprocessnode 內容.....	115
mergenode 內容.....	116
rfmaggregatenode 內容.....	118
sampleneode 內容.....	120
selectnode 內容.....	122
sortnode 內容.....	122
spacetimeboxes 內容.....	123
streamingtimeseries 內容.....	124
第 11 章 欄位作業節點內容.....	133
anonymizenode 內容.....	133
autodatapreprenode 內容.....	134
astimeintervalsnode 內容.....	138
binningnode 內容.....	138
derivenode 內容.....	141
ensemblenode 內容.....	144
fillernode 內容.....	145
filternode 內容.....	146
historynode 內容.....	147
partitionnode 內容.....	148
reclassifynode 內容.....	149
reordernode 內容.....	149
reprojectnode 內容.....	150
restructurenode 內容.....	151
rfanalysisnode 內容.....	151
settoflagnode 內容.....	152
statistictransformnode 內容.....	153
timeintervalsnode 內容 (已淘汰)	153
transposenode 內容.....	159
typenode 內容.....	161
第 12 章 圖形節點內容.....	169

圖形節點一般內容.....	169
collectionnode 內容.....	170
distributionnode 內容.....	171
evaluationnode 內容.....	172
graphboardnode 內容.....	174
histogramnode 內容.....	178
mapvisualization 內容.....	179
multiplotnode 內容.....	182
plotnode 內容.....	183
timeplotnode 內容.....	185
eplotnode 內容.....	187
tsnode 內容.....	187
webnode 內容.....	189

第 13 章 建模節點內容..... 191

一般建模節點內容.....	191
anomalydetectionnode 內容.....	191
apriorinode 內容.....	193
associationrulesnode 內容.....	194
autoclassifiernode 內容.....	197
設定演算法內容.....	199
autoclusternode 內容.....	199
autonumericnode 內容.....	201
bayesnetnode 內容.....	203
c50node 內容.....	205
carmanode 內容.....	206
cartnode 內容.....	207
chaidnode 內容.....	210
coxregnode 內容.....	212
decisionlistnode 內容.....	215
discriminantnode 內容.....	216
extensionmodelnode 內容.....	218
factornode 內容.....	220
featureselectionnode 內容.....	222
genlinnode 內容.....	224
glmmnode 內容.....	229
gle 內容.....	233
kmeansnode 內容.....	240
kmeansasnode 內容.....	241
knnnode 內容.....	242
kohonenode 內容.....	244
linearnode 內容.....	245
linearasnode 內容.....	247
logregnode 內容.....	248
lsvmnode 內容.....	253
neuralnetnode 內容.....	254
neuralnetworknode 內容.....	257
questnode 內容.....	259
randomtrees 內容.....	261
regressionnode 內容.....	263
sequencenode 內容.....	266
slrmnode 內容.....	267
statisticsmodelnode 內容.....	268
stpnode 內容.....	268
svmnode 內容.....	274
tcmnode 內容.....	275
ts 內容.....	280

treeas 內容.....	290
twostepnode 內容.....	292
twostepAS 內容.....	293
第 14 章 模型區塊節點內容.....	297
applyanomalydetectionnode 內容.....	297
applyapriorinode 內容.....	297
applyassociationrulesnode 內容.....	298
applyautoclassifiernode 內容.....	298
applyautoclusternode 內容.....	300
applyautonumericnode 內容.....	300
applybayesnetnode 內容.....	300
applyc50node 內容.....	301
applycarmanode 內容.....	301
applycartnode 內容.....	301
applychaidnode 內容.....	301
applycoxregnode 內容.....	302
applydecisionlistnode 內容.....	302
applydiscriminantnode 內容.....	303
applyextension 內容.....	303
applyfactornode 內容.....	304
applyfeatureselectionnode 內容.....	305
applygeneralizedlinearnode 內容.....	305
applyglmnode 內容.....	305
applygle 內容.....	306
applygmm 內容.....	306
applykmeansnode 內容.....	306
applyknnnode 內容.....	306
applykohonenode 內容.....	307
applylinearnode 內容.....	307
applylinearasnode 內容.....	307
applylogregnode 內容.....	307
applysvmnode 內容.....	308
applyneuralnetnode 內容.....	308
applyneuralnetworknode 內容.....	308
applyocsvmnode 內容.....	309
applyquestnode 內容.....	309
applyrandomtrees 內容.....	310
asapplyregressionnode 內容.....	310
applyselflearningnode 內容.....	310
applysequencenode 內容.....	311
applysvmnode 內容.....	311
applystpnode 內容.....	311
applytcmnode 內容.....	311
applyts 內容.....	312
applytimeseriesnode 內容 (已淘汰)	312
applytreeas 內容.....	312
applytwostepnode 內容.....	313
applytwostepAS 內容.....	313
applyxgboosttreenode 內容.....	313
applyxgboostlinearnode 內容.....	313
hdbscannugget 內容.....	314
kdeapply 內容.....	314
第 15 章 資料庫建模節點內容.....	315
Microsoft 建模的節點內容.....	315
Microsoft 建模節點內容.....	315

Microsoft 模型塊內容.....	317
Oracle 建模的節點內容.....	319
Oracle 建模節點內容.....	319
Oracle 模型區塊內容.....	326
IBM Netezza Analytics 建模節點內容.....	326
Netezza 建模節點內容.....	326
Netezza 模型區塊內容.....	340
第 16 章 輸出節點內容.....	341
analysisnode 內容.....	341
dataauditnode 內容.....	342
extensionoutputnode 內容.....	344
kdeexport 內容.....	345
matrixnode 內容.....	346
meansnode 內容.....	348
reportnode 內容.....	350
setglobalsnode 內容.....	351
simevalnode 內容.....	352
simfitnode 內容.....	353
statisticsnode 內容.....	354
statisticsoutputnode 內容.....	355
tablenode 內容.....	355
transformnode 內容.....	358
第 17 章 匯出節點內容.....	361
通用匯出節點內容.....	361
asexport 內容.....	361
cognosexportnode 內容.....	362
databaseexportnode 內容.....	364
datacollectionexportnode 內容.....	368
excelexportnode 內容.....	369
extensionexportnode 內容.....	370
jsonexportnode 內容.....	371
outputfilenode 內容.....	371
sasexportnode 內容.....	372
statisticsexportnode 內容.....	373
tm1odataexport 節點內容.....	373
tm1export 節點內容 (已淘汰)	375
xmlexportnode 內容.....	376
第 18 章 IBM SPSS Statistics 節點內容.....	379
statisticsimportnode 內容.....	379
statisticstransformnode 內容.....	379
statisticsmodelnode 內容.....	380
statisticsoutputnode 內容.....	380
statisticsexportnode 內容.....	381
第 19 章 Python 節點內容.....	383
gmm 內容.....	383
hdbscannode 內容.....	383
kdemodel 內容.....	385
kdeexport 內容.....	386
gmm 內容.....	387
ocsvmnode 內容.....	387
rfnode 內容.....	389
smotnode 內容.....	391
tsnnode 內容.....	391

xgboostlinearnode 內容.....	393
xgboosttreenode 內容.....	394
第 20 章 Spark 節點內容.....	397
isotonicasnode 內容.....	397
kmeansasnode 內容.....	397
multilayerperceptronnnode 內容.....	398
xgboostasnode 內容.....	398
第 21 章 SuperNode 內容.....	401
附錄 A: 節點名稱參照.....	403
模型塊名稱.....	403
避免重複的模型名稱.....	405
輸出類型名稱.....	405
附錄 B: 從舊式 Scripting 移轉至 Python Scripting.....	407
舊式 Script 移轉概觀.....	407
一般差異.....	407
Scripting 環境定義.....	407
指令與函數.....	407
文字與註解.....	407
運算子.....	408
條件句與迴圈.....	409
變數.....	410
節點、輸出及模型類型.....	410
內容名稱.....	410
節點參照.....	411
取得及設定內容.....	411
編輯串流.....	412
節點作業.....	412
迴圈.....	413
執行串流.....	414
透過檔案系統及儲存庫存取物件.....	414
串流作業.....	415
模型作業.....	415
文件輸出作業.....	416
舊式 Scripting 與 Python Scripting 之間的其他差異.....	416
注意事項.....	417
商標.....	418
產品說明文件的條款.....	418
索引.....	419

第 1 章 Scripting 和 Scripting 語言

Scripting 概觀

IBM SPSS Modeler 中的 Script 編寫是用於在使用者介面上實現過程自動化的強大工具。您使用滑鼠或鍵盤進行的動作，借助 Script 同樣可以完成，而且使用 Script 可以自動化那些手動執行將造成大量重複動作且高耗時的作業。

Script 的作用包括：

- 限制在串流中執行節點的特定順序。
- 設定節點內容並使用 CLEM（表示式操作控制語言）的子集合來執行衍生。
- 指定通常包含使用者互動的動作的自動執行順序，例如您可以建立一個模型，然後對其進行測試。
- 設定需要實際使用者互動的複合程序，例如需要重複模型產生和正在測試的交叉驗證步驟。
- 設定串流操作過程 - 例如，您可以提取一個模型訓練串流，執行它，然後自動生成對應的模型測試串流。

本章提供串流級 Script、獨立式 Script 以及 IBM SPSS Modeler 介面 SuperNode 內 Script 的高級說明和範例。有關 Scripting 語言、語法和指令的進一步資訊，請參閱接下來的章節。

註：

您無法匯入和執行在 IBM SPSS Modeler 中的 IBM SPSS Statistics 中建立的 Script。

Script 類型

IBM SPSS Modeler 使用三種類型的 Script：

- **串流 Script** 儲存為串流內容然後和指定串流一起儲存和載入。例如，可以撰寫自動化訓練和套用模型區塊處理程序的串流 Script。您還可以指定無論何時執行特定串流，都應該執行 Script 而不是串流的畫布內容。
- **獨立式 Script** 與任何特定串流都沒有關聯，它們儲存在外部文字檔中。例如，可以使用獨立式 Script 同時操作多個串流。
- **SuperNode Script** 儲存為 SuperNode 串流內容。SuperNode Script 只在終端 SuperNode 中可用。您可以使用 SuperNode Script 控制 SuperNode 內容的執行序列。對於非終端（來源或過程）SuperNode，可以直接在串流 Script 中為 SuperNode 或它包含的節點定義內容。

串流 Script

Script 可用於自訂特定串流中的作業並與該串流一起儲存。串流 Script 可用於指定某個串流中終端節點的特定執行順序。可以使用「串流 Script」對話框來編輯與目前串流一起儲存的 Script。

從「串流內容」對話框存取串流 Script 標籤：

1. 從工具功能表中，選擇：

串流內容 > 執行

2. 按一下執行標籤以處理目前串流的 Script。

使用串流 Script 對話框頂部的工具列圖示來執行下列作業：

- 將先前存在的獨立式 Script 內容匯入到視窗。
- 將 Script 儲存為文字檔。
- 列印 Script。
- 附加預設 Script。
- 編輯 Script（復原、剪下、複製、貼上及其他常見的編輯功能）。

- 執行整個目前 Script。
- 執行某個 Script 中的選定行。
- 在執行期間停止 Script。（只有在 Script 處於執行狀態的情況下，才會啟用此圖示。）
- 檢查 Script 的語法，如果發現任何錯誤，就將其顯示在對話框的下部窗格中進行檢閱。

註：從 16.0 版開始，SPSS Modeler 使用 Python Scripting 語言。16.0 之前的所有版本都使用 SPSS Modeler 所特有的 Scripting 語言，現稱為舊式 Scripting。根據您所使用的 Script 類型，在執行標籤上，選取預設（選用 Script）執行方式，然後選取 Python 或舊式。

您可以指定執行串流時是否執行 Script。要在每次執行串流時都執行 Script，並且按照 Script 的執行順序執行，請選取執行該 Script。此設定在串流層次提供自動化以便更快地建置模型。但是，預設設定為在執行串流的過程中忽略此 Script。即使選取選項忽略此 Script，也可以直接從此對話框執行 Script。

Script 編輯器提供了下列功能，這些功能有助於 Script 編寫：

- 語法強調顯示；將強調顯示關鍵字、文字值（例如字串和數字）以及備註。
- 行編號。
- 區塊比對；當游標處於程式區塊的開始位置時，還將強調顯示對應的結束區塊。
- 建議的自動補全。

可以使用 IBM SPSS Modeler 顯示喜好設定來自訂語法螢光筆使用的顏色和文字樣式。要存取顯示喜好設定，請選擇工具 > 選項 > 使用者選項並選擇語法標籤。

通過從快速功能表中選取自動建議或者按 Ctrl + Space，可以存取建議語法補全的清單。使用游標鍵在清單中上下移動，然後按 Enter 鍵可插入所選文字。要結束自動建議方式而不修改現有文字，請按 Esc 鍵。

除錯標籤顯示除錯訊息，並且可以用於在執行 Script 後立即對 Script 狀態進行評估。偵錯標籤包含一個唯讀文字區域和單行輸入文字欄位。文字區域顯示由 Script 傳送到標準輸出或標準錯誤（例如，通過錯誤訊息文字）的文字。輸入文字欄位將接收來自使用者的輸入。然後，將在對話框內最近執行的 Script 環境定義（稱為 Scripting 環境定義）中對此輸入進行評估。文字區域包含指令和生成的輸出，以便使用者能夠請參閱指令追蹤。文字輸入欄位始終包含命令提示字元（對於舊式 Script 而言為 -->）。

在下列情況下，將建立新的 Scripting 環境定義：

- 使用執行此 Script 或執行所選行來執行 Script。
- Scripting 語言會發生變更。

如果建立了新的 Scripting 環境定義，那麼將清除文字區域。

註：在 Script 窗格外執行串流不會修改 Script 窗格的 Script 環境定義。在該執行過程中建立的任何變數的值在 Script 對話框中都不可見。

串流 Script 範例：訓練神經網路

當執行時，串流可用於訓練神經網路模型。通常，要測試模型，可以執行建模節點以便將該模型新增到串流中，建立適當的連線，然後執行「分析」節點。

借助 IBM SPSS Modeler Script，您可以在建立模型區塊之後，實現模型區塊正在測試過程的自動化。例如，可以從「串流內容」對話框（工具 > 串流內容 > Script）執行下列串流 Script 來測試示範串流 druglearn.str（位於 IBM SPSS Modeler 安裝架構下的 /Demos/streams/ 資料夾中）：

```
stream = modeler.script.stream()
neuralnetnode = stream.findByType("neuralnetwork", None)
results = []
neuralnetnode.run(results)
appliernode = stream.createModelApplierAt(results[0], "Drug", 594, 187)
analysisnode = stream.createAt("analysis", "Drug", 688, 187)
typenode = stream.findByType("type", None)
stream.linkBetween(appliernode, typenode, analysisnode)
analysisnode.run([])
```

下列項目符號說明此 Script 範例中的每一行。

- 第一行定義指向目前串流的變數。
- 在第二行中，Script 將尋找「類神經網路」建置器節點。
- 在第三行中，Script 將建立可以在其中儲存執行結果的清單。
- 在第四行中，將建立「神經網路」模型區塊。此模型塊儲存在第三行中定義的清單內。
- 在第五行中，將為此模型區塊建立模型套用節點並將此節點放入串流畫布中。
- 在第六行中，將建立稱為 **Drug** 的分析節點。
- 在第七行中，Script 將尋找 type 節點。
- 在第八行中，Script 將連接第五行中在類型節點與分析節點之間建立的模型套用節點。
- 最後，執行分析節點以生成分析報告。

可以使用 Script 從頭開始（從空白畫布開始）建立並執行串流。

Jython 代碼大小限制

Jython 將每個 Script 編譯為 Java 位元組碼，然後由 Java 虛擬機器 (JVM) 執行。但是，Java 對單個位元組碼檔案的大小施加了限制。因此，當 Jython 試圖載入位元組碼時，可能會導致 JVM 當機。IBM SPSS Modeler 無法阻止這種情況發生。

請確保您使用適當的編碼做法（例如，使用變數或函數計算一般中間值，從而盡量減少複製代碼）撰寫 Jython Script。如有需要，可能需要將代碼分割成若干個來源檔案，或者使用模組進行定義，這些模組繼而編譯為個別的位元組碼檔案。

獨立式 Script

「獨立式 Script」對話框用於建立或編輯儲存為文字檔的 Script。它顯示了檔案名稱，提供了用於載入、儲存、匯入和執行 Script 的機能。

要存取「獨立式 Script」對話框，請執行以下操作：

從主功能表中，選擇：

工具 > 獨立式 Script

對串流 Script 可用的工具列和 Script 語法檢查選項對獨立式 Script 同樣適用。請參閱第 1 頁的『串流 Script』主題，以取得更多資訊。

獨立式 Script 範例：儲存和載入模型

獨立式 Script 可用於串流操作。假設有兩個串流，第一個串流建立模型並產生規則集，第二個串流則通過現有資料欄位，採用圖示的方式對規則集進行探索。該案例的獨立式 Script 可能具有如下形式：

```
taskrunner = modeler.script.session().getTaskRunner()

# Modify this to the correct Modeler installation Demos folder.
# Note use of forward slash and trailing slash.
installation = "C:/Program Files/IBM/SPSS/Modeler/19/Demos/"

# First load the model builder stream from file and build a model
druglearn_stream = taskrunner.openStreamFromFile(installation + "streams/druglearn.str", True)
results = []
druglearn_stream.findByType("c50", None).run(results)

# Save the model to file
taskrunner.saveModelToFile(results[0], "rule.gm")

# Now load the plot stream, read the model from file and insert it into the stream
drugplot_stream = taskrunner.openStreamFromFile(installation + "streams/drugplot.str", True)
model = taskrunner.openModelFromFile("rule.gm", True)
modelapplier = drugplot_stream.createModelApplier(model, "Drug")

# Now find the plot node, disconnect it and connect the
# model applier node between the derive node and the plot node
derivencode = drugplot_stream.findByType("derive", None)
plotnode = drugplot_stream.findByType("plot", None)
drugplot_stream.disconnect(plotnode)
```

```
modelapplier.setPositionBetween(derivinode, plotnode)
drugplot_stream.linkBetween(modelapplier, derivinode, plotnode)
plotnode.setPropertyValue("color_field", "$C-Drug")
plotnode.run([])
```

註：要瞭解有關一般 Scripting 語言的更多資訊，請參閱第 13 頁的『Scripting 語言概觀』。

獨立式 Script 範例：產生功能選項模式

首先開啟一個空白畫布，在此範例中將建立一個串流，該串流產生一個功能選項模型，套用此模型並建立一個表格，該表格包含有對於指定目標而言重要性最高的 15 個欄位。

```
stream = modeler.script.session().createProcessorStream("featureselection",
True)

statisticsimportnode = stream.createAt("statisticsimport", "Statistics
File", 150, 97)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/
customer_dbase.sav")

typenode = stream.createAt("type", "Type", 258, 97)
typenode.setKeyedPropertyValue("direction", "response_01", "Target")

featureselectionnode = stream.createAt("featureselection", "Feature
Selection", 366, 97)
featureselectionnode.setPropertyValue("top_n", 15)
featureselectionnode.setPropertyValue("max_missing_values", 80.0)
featureselectionnode.setPropertyValue("selection_mode", "TopN")
featureselectionnode.setPropertyValue("important_label", "Check Me Out!")
featureselectionnode.setPropertyValue("criteria", "Likelihood")

stream.link(statisticsimportnode, typenode)
stream.link(typenode, featureselectionnode)
models = []
featureselectionnode.run(models)

# Assumes the stream automatically places model apply nodes in the stream
applynode = stream.findByType("applyfeatureselection", None)
tablenode = stream.createAt("table", "Table", applynode.getXPosition() + 96,
applynode.getYPosition())
stream.link(applynode, tablenode)
tablenode.run([])
```

此 Script 建立了一個用以讀入資料的來源節點，使用「類型」節點將欄位 response_01 的角色（方向）設定為 Target，然後建立並執行「功能選擇」節點。此 Script 還連接串流畫布上的各個節點和位置以生成可讀取的佈置。然後結果模型區塊與表格節點相連接，「表格」節點列出了內容 selection_mode 和 top_n 所確定的 15 個最重要的欄位。請參閱第 222 頁的『featureselectionnode 內容』主題，以取得更多資訊。

SuperNode Script

通過使用 IBM SPSS Modeler 的 Scripting 語言，可以建立和儲存所有終端 SuperNode 中的 Script。這些 Script 只在終端 SuperNode 中可用，並且常在建立範本串流或用於強制 SuperNode 內容以特定順序執行時使用。使用 SuperNode Script，您也可以在此串流中執行多個 Script。

例如，假設需要指定一個複合串流的執行順序，並且 SuperNode 包含若干個包括設定廣域值節點的節點，而執行設定廣域值節點又需要在衍生用於「繪圖」節點的新欄位之前進行。這種情況下，可以建立一個首先執行設定廣域值節點的 SuperNode Script。然後由此節點計算的值（例如，平均值或標準差）可以在執行「圖形」節點時使用。

在 SuperNode Script 中也可以指定節點內容，操作方法與在其他 Script 中的進行的操作一樣。另外，您可以直接從串流 Script 變更和定義任何 SuperNode 或其封裝節點的內容。請參閱第 401 頁的『第 21 章 SuperNode 內容』主題，以取得更多資訊。此方法適用於來源和處理 SuperNode 以及終端 SuperNode。

註: 因為只有終端 SuperNode 能夠執行自身 Script, 所以 SuperNode 對話框的 Script 標籤只在用於終端 SuperNode 時可用。

從主畫布開啟 SuperNode Script 對話框:

從串流畫布選擇終端 SuperNode, 然後從 SuperNode 功能表選擇:

SuperNode Script...

從放大 SuperNode 畫布開啟 SuperNode Script 對話框:

用滑鼠右鍵按一下 SuperNode 畫布, 然後從快速功能表中選擇:

SuperNode Script...

SuperNode Script 範例

下列 SuperNode Script 宣告 SuperNode 中終端節點的執行順序。此順序可確保首先執行設定廣域節點, 以便隨後執行其他節點時可使用由此節點計算的值。

```
execute 'Set Globals'  
execute 'gains'  
execute 'profit'  
execute 'age v. $CC-pep'  
execute 'Table'
```

鎖定和解除鎖定 SuperNode。

下列範例說明如何鎖定和解除鎖定 SuperNode:

```
stream = modeler.script.stream()  
superNode=stream.findByID('id854RNTSD5MB')  
# unlock one super node  
print 'unlock the super node with password abcd'  
if superNode.unlock('abcd'):  
    print 'unlocked.'  
else:  
    print 'invalid password.'  
# lock one super node  
print 'lock the super node with password abcd'  
superNode.lock('abcd')
```

串流中的迴圈和條件執行

從 16.0 版開始, 通過 SPSS Modeler, 您可以選取各個對話框中的值在串流中建立一些基本 Script, 而無需使用 Scripting 語言直接撰寫指令 / 指示。可通過此方式建立的兩種主要類型的 Script 是簡單迴圈以及在符合條件時執行節點的方式。

可以組合串流中的迴圈規則和條件執行規則。例如, 您可能具有來自世界各地製造商的汽車銷售相關資料。您可以在串流中設定一個用於處理資料的迴圈, 從而按製造國家識別詳細資料, 並將資料輸出到各個顯示了詳細資料 (例如, 按型號排列的銷售量, 按製造商和引擎大小排列的排放層次等) 的圖形。如果您希望僅分析歐洲資訊, 那麼還可以向迴圈新增條件, 以阻止針對總部設在美國和亞洲的製造商建立圖形。

註: 由於迴圈和條件式執行均以背景 Script 為基礎, 因此它們僅適用於執行的整個串流。

- **迴圈** 使用迴圈可自動化重複作業。例如, 這可能意味著向串流新增給定號碼的節點, 並且每次變更一個節點參數。另外, 您可以控制串流或分支執行給定次數, 如下列範例所示:
 - 按給定次數執行串流, 並每次變更來源。
 - 按給定次數執行串流, 每次變更變數的值。
 - 按給定次數執行串流, 每次執行時額外輸入一個欄位。
 - 按給定次數建置模型, 並每次變更模型設定。

- **條件執行** 您可以使用它根據預先定義的條件來控制終端節點的執行方式，可能的範例如下：
 - 根據給定的值是 `true` 還是 `false`，控制是否執行節點。
 - 定義節點迴圈是平行還是循序執行。

迴圈和條件執行都是在「串流內容」對話框中的「執行」標籤中設定的。任何在條件或迴圈要求中使用的節點都隨附加到這些節點的附加符號一起顯示在串流畫布上，此符號用於指示這些節點將參與迴圈和條件執行。

您可以通過下列三種方式中的其中一種來存取「執行」標籤：

- 使用主對話框頂部的功能表：
 1. 從「工具」功能表中，選擇：
 - 串流內容 > 執行**
 2. 按一下「執行」標籤以處理目前串流的 Script。
- 從串流中：
 1. 用滑鼠右鍵按一下節點，然後選擇**迴圈/條件執行**。
 2. 選取相關子功能表選項。
- 從主對話框頂部的圖形工具列中，按一下串流內容圖示。

如果這是您第一次設定迴圈或條件執行詳細資料，請在「執行」標籤上選取**迴圈/條件執行**執行方式，然後選取**條件**或**迴圈**子標籤。

串流中的迴圈

通過迴圈，您可以自動化串流中的重複作業；可能的範例如下：

- 按給定次數執行串流，並每次變更來源。
- 按給定次數執行串流，每次變更變數的值。
- 按給定次數執行串流，每次執行時額外輸入一個欄位。
- 按給定次數建置模型，並每次變更模型設定。

可以在串流「執行」標籤的**迴圈**子標籤上設定要符合的條件。若要顯示子標籤，請選取**迴圈/條件執行**執行模式。

如果設定了**迴圈/條件式執行**執行方式，那麼在您執行串流時，您定義的所有迴圈要求都將生效。您可以選擇性地產生迴圈需求的 Script 程式碼，並按一下「迴圈」子標籤右下角中的**貼上...**，將它貼到 Script 編輯器中；主要「執行」標籤會顯示變更，以顯示**預設值（選用 Script）**執行模式，且 Script 位於標籤頂端部分。這意味著，您可以先使用多個迴圈對話框選項來定義迴圈結構，然後再產生可在 Script 編輯器中進行進一步自訂的 Script。請注意，當您按一下**貼上...**時，您已定義的任何條件式執行需求也會顯示在產生的 Script 中。

重要：如果您在 IBM SPSS Collaboration and Deployment Services 工作中執行某個 SPSS Modeler 串流，那麼可以覆蓋您在此串流中設定的迴圈變數。這是因為，IBM SPSS Collaboration and Deployment Services 工作編輯器項目將置換 SPSS Modeler 項目。例如，如果您在串流中設定了某個迴圈變數以便為每個迴圈建立不同的輸出檔案名稱，那麼這些檔案將在 SPSS Modeler 中正確命名，但由 IBM SPSS Collaboration and Deployment Services Deployment Manager 的「結果」標籤中輸入的固定項目置換。

要設定迴圈，請完成下列步驟：

1. 建立疊代索引鍵以定義將在串流中執行的主要迴圈結構。有關進一步資訊，請參閱[建立疊代索引鍵](#)。
2. 在需要時，定義一個或多個疊代變數。有關進一步資訊，請參閱[建立疊代變數](#)。
3. 您建立的疊代和所有變數都將顯示在該子標籤的主要部分中。預設情況下，將按顯示順序執行疊代；要在清單中上下移動疊代，請按一下疊代以將其選中，然後使用該子標籤右側欄中的向上或向下箭頭變更順序。

建立用於串流中的迴圈的疊代索引鍵

使用疊代索引鍵可以定義將在串流中執行的主要迴圈結構。例如，如果要對汽車銷售進行分析，那麼可以建立串流參數製造國家，並將其用作疊代索引鍵；在執行串流時，此關鍵字將設定為各個疊代過程中您的資料中的各個不同的國家值。使用「定義疊代索引鍵」對話框可以設定關鍵字。

若要開啟對話框，請選取「迴圈」子標籤左下角的**疊代索引鍵...** 按鈕，或用滑鼠右鍵按一下串流中的任何節點並選取 **迴圈/條件式執行 > 定義疊代索引鍵 (欄位)** 或 **迴圈/條件式執行 > 定義疊代索引鍵 (值)**。如果是從串流中開啟此對話框，那麼系統可能會自動為您填寫一些欄位，例如，節點的名稱。

要設定疊代索引鍵，請填寫下列欄位：

疊代目標。 您可以選取下列其中一個選項：

- **串流參數 - 欄位。** 使用此選項可建立一個迴圈，用於將現有串流參數的值依次設定為各個指定欄位。
- **串流參數 - 值。** 使用此選項可建立一個迴圈，用於將現有串流參數的值依次設定為各個指定值。
- **節點內容 - 欄位。** 使用此選項可建立一個迴圈，用於將節點內容的值依次設定為各個指定欄位。
- **節點內容 - 值。** 使用此選項可建立一個迴圈，用於將節點內容的值依次設定為各個指定值。

設定內容。 選擇將在每次執行迴圈時設定其值的項目。您可以選取下列其中一個選項：

- **參數。** 僅當您選取**串流參數 - 欄位**或**串流參數 - 值**時才可用。從可用清單中選取所需參數。
- **節點。** 僅在您選取**節點內容 - 欄位**或**節點內容 - 值**的情況下可用。選取要對其設定迴圈的節點。按一下瀏覽按鈕以開啟「選取節點」對話框並選擇您想要的節點；如果列出的節點太多，您可以透過選取下列其中一個節點類別來過濾顯示以僅顯示特定類型的節點：來源、程序、圖形、建模、輸出、匯出或套用模型。
- **內容。** 僅在您選取**節點內容 - 欄位**或**節點內容 - 值**的情況下可用。從可用清單中選取節點的內容。

要使用的欄位。 僅當您選取**串流參數 - 欄位**或**節點內容 - 欄位**時才可用。選擇節點中要用於提供疊代值的欄位。您可以選取下列其中一個選項：

- **節點。** 僅當您選取**串流參數 - 欄位**時才可用。選取要為其設定迴圈的包含詳細資料的節點。按一下瀏覽按鈕以開啟「選取節點」對話框並選擇您想要的節點；如果列出的節點太多，您可以透過選取下列其中一個節點類別來過濾顯示以僅顯示特定類型的節點：來源、程序、圖形、建模、輸出、匯出或套用模型。
- **欄位清單。** 按一下右邊欄中的清單按鈕可顯示「選取欄位」對話框，您可以在此對話框中選取節點中要用於提供疊代資料的欄位。如需相關資訊，請參閱第 8 頁的『[選取用於疊代的欄位](#)』。

要使用的值。 僅當您選取**串流參數 - 值**或**節點內容 - 值**時才可用。選擇選取的欄位內要用作疊代值的值。您可以選取下列其中一個選項：

- **節點。** 僅當您選取**串流參數 - 值**時才可用。選取要為其設定迴圈的包含詳細資料的節點。按一下瀏覽按鈕以開啟「選取節點」對話框並選擇您想要的節點；如果列出的節點太多，您可以透過選取下列其中一個節點類別來過濾顯示以僅顯示特定類型的節點：來源、程序、圖形、建模、輸出、匯出或套用模型。
- **欄位清單。** 選取節點中用於提供疊代資料的欄位。
- **值清單。** 按一下右邊欄中的清單按鈕可顯示「選取值」對話框，您可以在此對話框中選取要用於提供疊代資料的欄位值。

建立用於串流中的迴圈的疊代變數

您可以使用疊代變數在每次執行迴圈時變更串流中的串流參數值或選定節點的內容值。例如，如果串流迴圈將對汽車銷售資料進行分析並使用製造國家作為疊代索引鍵，那麼您可能會具有一個按型號顯示銷售額的圖形輸出，以及另一顯示了廢氣排放資訊的圖形輸出。在這些情況下，您可以建立疊代變數，這些變數將為生成的圖形建立新標題，例如瑞典汽車排放和按型號排列的日本汽車銷售額。使用「定義疊代變數」對話框可以設定任何您需要的變數。

若要開啟對話框，請選取「迴圈」子標籤左下角的**新增變數...** 按鈕，或用滑鼠右鍵按一下串流中的任何節點並選取：**迴圈/條件式執行 > 定義疊代變數**。

要設定疊代變數，請填寫下列欄位：

變更。 選取要修正的屬性的類型。可以從**串流參數**或**節點內容**中進行選擇。

- 如果選擇**串流參數**，請選擇所需參數，然後通過迴圈的各個疊代，使用下列其中一個選項（如果在串流中可用）定義應該將該參數設定為的值。
 - **廣域變數**。選取應該將串流參數設定為的廣域變數。
 - **表格輸出蜂巢 Cell**。要將串流參數設定為表格輸出蜂巢 Cell 中的值，請從清單中選取表格，然後輸入要使用的**列和欄**。
 - **手動輸入**。如果要手動為此參數輸入將在各個疊代中採用的值，請選取此選項。傳回到「迴圈」子標籤時，將建立一個可在其中輸入所需文字的新欄。
- 如果選擇**節點內容**，請選擇所需節點以及該節點的其中一個內容，然後設定要用於該內容的值。通過使用下列其中一個選項，可以設定新內容值：
 - **單獨**。內容值將使用疊代索引鍵值。如需相關資訊，請參閱第 7 頁的『[建立用於串流中的迴圈的疊代索引鍵](#)』。
 - **作為資源字首**。使用疊代索引鍵值作為在**資源**欄位中輸入的內容的字首。
 - **作為資源字尾**。使用疊代索引鍵值作為在**資源**欄位中輸入的內容的字尾。

如果選取字首或字尾選項，那麼系統將提示您向**資源**欄位新增附加文字。例如，如果疊代索引鍵值為製造國家並且您選取**作為資源字首**，那麼可以在此欄位中輸入 - 按型號排列的銷售額。

選取用於疊代的欄位

建立疊代時，您可以使用「選取欄位」對話框選取一個或多個欄位。

排序方式 可以通過選取下列其中一個選項對可用的欄位進行排序以供檢視：

- **自然** 當欄位沿資料串流向下傳遞至現行節點時，檢視欄位的順序。
- **名稱** 採用按字母順序對欄位進行排序以便於檢視。
- **類型** 檢視欄位時按其測量層次排序。此選項在選取採用特定測量層級的欄位時有用。

從清單中一次選取一個欄位或使用按住 **Ctrl** 並點選或 **Shift** 並點選方法來選取多個欄位。您還可以使用清單下面的按鈕來基於測量層級選取欄位群組，或者選取或取消選取表格中的所有欄位。

請注意，可供選擇的欄位將被過濾，以僅顯示適用於您使用的串流參數或節點內容的欄位。例如，如果您使用的是儲存類型為字串的串流參數，那麼將僅顯示儲存類型為字串的欄位。

串流中的條件執行


通過條件式執行，您可以根據符合您所定義條件的串流內容來控制終端節點的執行方式；可能的範例如下：

- 根據給定的值是 **true** 還是 **false**，控制是否執行節點。
- 定義節點迴圈是平行還是循序執行。

可以在串流「執行」標籤的**條件式**子標籤上設定要符合的條件。若要顯示子標籤，請選取**迴圈/條件執行**執行模式。

如果設定了**迴圈/條件執行**執行方式，那麼在您執行串流時，您定義的所有條件執行要求都將生效。您可以選擇性地產生條件式執行需求的 **Script** 程式碼，並按一下「條件式」子標籤右下角的**貼上 ...**，將它貼上至 **Script** 編輯器；主要「執行」標籤顯示畫面會變更，以顯示**預設值（選用 Script）**執行模式，並且 **Script** 位於標籤的頂端部分。這意味著，您可以先使用多個迴圈對話框選項來定義條件，然後再產生可在 **Script** 編輯器中進行進一步自訂的 **Script**。請注意，當您按一下**貼上 ...**時，您已定義的任何迴圈需求也會顯示在產生的 **Script** 中。

要設定條件，請完成下列步驟：

1. 在「條件式」子標籤的右側欄中，按一下「新增條件」按鈕 ，以開啟「**新增條件式執行陳述式**」對話框。在此對話框中，可以指定執行節點所必須符合的條件。
2. 在「**新增條件式執行陳述式**」對話框中，指定下列內容：

- a. **節點**。選取要對其設定條件執行的節點。按一下瀏覽按鈕以開啟「選取節點」對話框並選擇所需節點；如果列出的節點過多，那麼可以對顯示結果進行過濾，以按下列其中一個種類顯示節點：「匯出」、「圖形」、「建模」或「輸出」節點。
- b. **作為依據的條件**。指定執行節點所必須符合的條件。您可以從下列四個選項中選擇其中一個：**串流參數**、**廣域變數**、**表格輸出蜂巢 Cell** 或**始終滿足**。在對話框下半部分中輸入的詳細資料由您選擇的條件控制。
 - **串流參數** 從提供的清單中選擇參數，然後選擇該參數的**運算子**；例如，運算子可以是大於、等於、少於和介於之間等等。然後，輸入**值**或**最小值**和**最大值**，具體取決於運算子。
 - **廣域變數**。從提供的清單中選取變數；例如，這可能包含平均數、總和、最小值、最大值或標準差。然後您選取需要的**運算子**及**值**。
 - **表格輸出蜂巢 Cell**。從可用清單中選擇表格節點，然後選擇表格中的**列**和**欄**。然後您選取需要的**運算子**及**值**。
 - **始終滿足**。如果必須始終執行節點，請選取此選項。選取此選項後，將無需選取其他參數。
3. 重複步驟 1 和 2 所需次數，直到您設定了所有需要的條件。所選節點和執行該節點前所必須符合的條件將分別顯示在該子標籤主要部分中的**執行節點**和**如果符合此條件欄**中。
4. 依預設，將按顯示順序執行節點和條件；要在清單中上下移動節點和條件，請按一下節點或條件以將其選中，然後使用該子標籤右側欄中向上或向下箭頭變更順序。

另外，您可以在「條件」子標籤的底部設定下列選項：

- **依順序對所有條件進行求值**。選取此選項可按條件在該子標籤上的顯示順序對各個條件進行求值。對所有條件進行求值後，將立即執行那些條件求值為 "true" 的節點。
- **一次執行一個節點**。只有已選取**依順序對所有條件進行求值**時才可用。選中此選項表示，如果某個條件求值為 "true"，那麼將先執行與該條件關聯的節點，然後再對下一個條件進行求值。
- **在首次命中之前進行求值**。選中此選項表示，將僅執行第一個根據您指定的條件傳回 "true" 求值的節點。

執行和中斷 Script

可以通過多種方法來執行 Script。例如，在串流 Script 或獨立式 Script 對話框中，「執行此 Script」按鈕將執行完整的 Script：



圖 1: 「執行此 Script」按鈕

「執行選定的行」按鈕用於執行您在 Script 中選取的單一行或者相鄰行所組成的區塊：



圖 2: 「執行選定的行」按鈕

可以使用下列方式執行 Script：

- 在串流 Script 或獨立式 Script 對話框中，按一下「執行此 Script」或「執行選定的行」按鈕。
- 在**執行此 Script** 設定為預設執行方式的情況下執行串流。
- 啟動後以交互模式使用 `-execute` 旗標。請參閱第 57 頁的『使用指令行引數』主題，以取得更多資訊。

註：如果在「SuperNode Script」對話框中選取**執行此 Script**，則將在執行 SuperNode 時執行 SuperNode Script。

中斷 Script 執行

「串流 Script」對話框中的紅色「停止」按鈕將在 Script 執行過程中被啟動。使用此按鈕可以放棄 Script 和任何目前串流的執行。

尋找/取代

可在編輯 Script 或表示式文字的位置（包括 Script 編輯器和 CLEM 表示式建置器）或定義「報告」節點中的範本時使用「尋找/取代」對話框。在上述任一區域編輯文字時，按 **Ctrl+F** 可存取此對話框，並確保游標的對焦點位於文字區域中。例如，處理「填入器」節點時，可以通過「設定」標籤的任一文字區域或表示式建置器中的文字欄位存取此對話框。

1. 當游標位於文字區中時，按 **Ctrl+F** 可存取「尋找/取代」對話框。
2. 輸入要搜尋的文字，或從最近搜尋項目下拉清單中選擇。
3. 輸入取代文字（如果有的話）。
4. 按一下**尋找**下一個開始搜尋。
5. 按一下**取代**取代目前選定的內容，或按一下**全部置換**更新所有項或選定的實例。
6. 每次作業完成後，此對話框將關閉。從任一文字區域中按 **F3** 鍵，可重複上一次尋找作業，或按 **Ctrl+F**，可再次存取該對話框。

搜尋選項

區分觀察值。 指定尋找作業是否區分觀察值；例如 *myvar* 是否與 *myVar* 相符。無論怎樣設定，取代文字始終完全按照已輸入插入。

僅尋找全字。 指定尋找作業是否相符單詞中內部的文字。如果已選取，*spider* 的搜尋結果將不會包括 *spiderman* 或 *spider-man*。

一般表示式。 指定是否使用正規表示式語法（請參閱下一節）。如果已選取，**僅限於整個單詞**選項將停用並且會忽略其值。

僅選取的文字。 控制項使用**全部置換**選項時的搜尋範圍。

正規表示式語法

使用正規表示式，您可以搜尋特殊字元（如定位字元或換行字元）、字元類別或範圍（如 *a* 到 *d*）、任何數字或非數字以及界限（如行首或行尾）。受支援的表示式類型如下。

字元	相符項
x	字元 x
\\	反斜線字元
\\0n	含八進位值的字元 0n (0 <= n <= 7)
\\0nn	含八進位值的字元 0nn (0 <= n <= 7)
\\0mnn	含八進位值的字元 0mnn (0 <= m <= 3, 0 <= n <= 7)
\\xhh	含十六進位值的字元 0xhh
\\uhhhh	含十六進位值的字元 0xhhhh
\\t	定位點 ('\\u0009')
\\n	換行符 ('\\u000A')
\\r	回車符 ('\\u000D')
\\f	換頁符 ('\\u000C')
\\a	警示（鈴聲）字元 ('\\u0007')
\\e	跳出符 ('\\u001B')
\\cx	x 對應的控制字元

字元類別	相符項
[abc]	a、b、或 c (簡單類別)
[^abc]	除 a、b、或 c 之外的所有字元 (相減法)
[a-zA-Z]	a 到 z 或 A 到 Z, 包含 (範圍)
[a-d[m-p]]	a 到 d 或 m 到 p (聯集)。也可指定為 [a-dm-p]
[a-z&&[def]]	a 到 z 和 d、e、或 f (交集)
[a-z&&[^bc]]	a 到 z, 除 b 和 c 外 (相減法)。也可指定為 [ad-z]
[a-z&&[^m-p]]	a 到 z, 而非 m 到 p (相減法)。也可指定為 [a-lq-z]

預先定義的字元類別	相符項
.	任意字元 (可能或不可能與行終止符相符)
\d	任意數字: [0-9]
\D	非數字: [^0-9]
\s	空格字元: [\t\n\x0B\f\r]
\S	非空格字元: [^\s]
\w	單詞字元: [a-zA-Z_0-9]
\W	非單詞字元: [^\w]

界限比對符	相符項
^	行首
\$	行尾
\b	單詞界限
\B	非單詞界限
\A	輸入的開頭
\Z	除最後終止符外 (如果有), 輸入的結尾
\z	輸入的結尾

第 2 章 Scripting 語言

Scripting 語言概觀

通過 IBM SPSS Modeler 的 Scripting 工具，您可以建立一些 Script，這些 Script 可以在 SPSS Modeler 使用者介面上執行、操作輸出物件並執行指令語法。您可以在 SPSS Modeler 中直接執行這些 Script。

IBM SPSS Modeler 中的 Script 以 Scripting 語言 Python 編寫。IBM SPSS Modeler 所使用的基於 Java 的 Python 實作稱為 Jython。Scripting 語言包含下列功能：

- 用來參照節點、串流、專案、輸出及其他 IBM SPSS Modeler 物件的格式。
- 可以用來操作這些物件的一組 Scripting 陳述式或指令。
- 用來設定變數、參數及其他物件的值的 Scripting 表示式語言。
- 支援字面文字的評論、接續及封鎖。

下列各節說明了 Python Scripting 語言、Python 的 Jython 實作以及在 IBM SPSS Modeler 內進行 Scripting 的入門基本語法。後面的章節中會提供特定內容及指令的相關資訊。

Python 和 Jython

Jython 是 Python Scripting 語言的實作，它以 Java 語言撰寫並與 Java 平台進行整合。Python 是一種面向物件的功能強大的 Scripting 語言。Jython 很有用，因為它具有成熟 Script 語言的生產力特性，而且與 Python 不同，Jython 可以在任何支援 Java 虛擬機器 (JVM) 的環境中執行。這意味著您在撰寫程式時可以使用 JVM 上的 Java 程式庫。通過 Jython，您可以利用此差異並使用 Python 語言的語法和大部分功能

作為一種 Scripting 語言，Python（及其 Jython 實作）易於學習並能夠高效地進行編碼，而且具備建立執行程式所需要的最小結構。可以在交互方式下已輸入代碼，即一次已輸入一行。Python 是一種解釋性 Scripting 語言；它沒有 Java 中的預編譯步驟。Python 程式僅僅是文字檔，系統將在輸入這些檔案時對其進行解釋（在解析語法錯誤後）。簡單表示式（例如已定義的值）以及更加複合的動作（例如函數定義）將立即執行並可供使用。可以快速測試任何對代碼進行的變更。但是，Script 解譯確實存在一些缺點。例如，由於使用未定義的變數不是編譯器錯誤，因此只有在執行使用了該變數的陳述式的情況下，才會偵測到此錯誤。在這種情況下，可以編輯並執行程式以偵錯錯誤。

Python 將所有內容（包含所有資料和代碼）視為物件。因此，您可以使用多行代碼來操作這些物件。某些選取類型（例如數字和字串）將被更加方便地視為值而不是物件；Python 受支援此行為。有一個受支援的空值。此空值具有保留名稱 None。

有關 Python 和 Jython Scripting 的更深入簡介以及一些範例 Script，請參閱 <http://www.ibm.com/developerworks/java/tutorials/j-jython1/j-jython1.html> 和 <http://www.ibm.com/developerworks/java/tutorials/j-jython2/j-jython2.html>。

Python Scripting

本 Python Scripting 語言手冊簡介了在 IBM SPSS Modeler 中編製 Script 時最可能使用的元件，其中包括概念和程式設計基礎。這將為您提供足夠的知識來開發自己的 Python Script，以便在 IBM SPSS Modeler 中使用。

作業

使用等號 (=) 完成指派。例如，若要將值 "3" 指派給稱為 "x" 的變數，您可以使用下列陳述式：

```
x = 3
```

等號還可用於將字串類型資料指派給變數。例如，要將值 "a string value" 指派給變數 "y"，您可以使用下列陳述式：

```
y = "a string value"
```

下表列出了一些常用比較作業和數值型作業及其說明。

作業	說明
$x < y$	x 是否少於 y?
$x > y$	x 是否大於 y?
$x \leq y$	x 是否小於或等於 y?
$x \geq y$	x 是否大於或等於 y?
$x == y$	x 是否等於 y?
$x != y$	x 不等於 y?
$x <> y$	x 不等於 y?
$x + y$	將 y 與 x 相加
$x - y$	從 x 中減去 y
$x * y$	將 x 乘以 y
x / y	將 x 除以 y
$x ** y$	求 x 的 y 次幂

清單

清單是元素序列。清單可以包含任意號碼的元素，而清單的元素可以是任何類型的物件。也可以將清單視為陣列。隨著新增、移除或取代元素，清單中的元素數目可能會增加或減少。

範例

<code>[]</code>	任何空白清單。
<code>[1]</code>	包含單個元素（整數）的清單。
<code>["Mike", 10, "Don", 20]</code>	包含 4 個元素（兩個字串元素和兩個整數元素）的清單。
<code>[[], [7], [8, 9]]</code>	清單的清單。每個子清單都是一個空白清單或整數元素清單。
<pre>x = 7; y = 2; z = 3; [1, x, y, x + y]</pre>	整數清單。此範例說明了變數和表示式的使用。

您可以向變數分配清單，例如：

```
mylist1 = ["one", "two", "three"]
```

然後，可以存取清單的特定元素，例如：

```
mylist[0]
```

這將導致下列輸出：

```
one
```


方括弧 ([]) 中的數字稱為指標，它指向清單中的某個特定元素。將從 0 開始對清單中的元素編製索引。

您也可以選取清單中的一系列元素；這稱為切割。例如，`x[1:3]` 會選取 `x` 的第二和第三個元素。結束索引是超出選取範圍的索引。

字串

字串是一個被視為值的不可變字元序列。字串支援所有生成新字串的不可變序列函數和運算子。例如，`"abcdef"[1:4]` 將生成輸出 `"bcd"`。

在 Python 中，字元由長度為 1 的字串代表。

字串文字通過使用單重參照或三引號來定義。使用單引號定義的字串不能跨行，而使用三引號定義的字串可以跨行。字串可以用單引號 (') 或雙引號 (") 括住。引號字元可能包含另一個未跳出的引號字元或跳出的引號字元，亦即前面是反斜線 (\) 字元。

範例

```
"This is a string"
'This is also a string'
"It's a string"
'This book is called "Python Scripting and Automation Guide".'
"This is an escape quote (\") in a quoted string"
```

Python 剖析器將自動連結多個以空格分隔的字串。這樣您可以更輕鬆地鍵入長字串，並且更容易在單個字串中混用引號類型，例如：

```
"This string uses ' and " 'that string uses "."
```

這會導致下列輸出：

```
This string uses ' and that string uses ".
```

字串支援一些有用的方法。下表中會提供一些方法。

方法	用法
<code>s.capitalize()</code>	對 <code>s</code> 執行首字母大寫
<code>s.count(ss {,start {,end}})</code>	計算 <code>ss</code> 在 <code>s[start:end]</code> 中的出現次數
<code>s.startswith(str {, start {, end}})</code> <code>s.endswith(str {, start {, end}})</code>	測試以請參閱 <code>s</code> 是否以 <code>str</code> 開頭 測試以請參閱 <code>s</code> 是否以 <code>str</code> 結尾
<code>s.expandtabs({size})</code>	將 Tab 取代為空格，預設 <code>size</code> 為 8
<code>s.find(str {, start {, end}})</code> <code>s.rfind(str {, start {, end}})</code>	在 <code>s</code> 中尋找 <code>str</code> 的第一個指標；如果找不到，那麼結果為 -1。 <code>rfind</code> 由右至左進行搜尋。
<code>s.index(str {, start {, end}})</code> <code>s.rindex(str {, start {, end}})</code>	在 <code>s</code> 中尋找 <code>str</code> 的第一個指標；如果找不到，那麼將引起 <code>ValueError</code> 。 <code>rindex</code> 從右到左進行搜尋。
<code>s.isalnum</code>	測試以請參閱字串是否為英數字串
<code>s.isalpha</code>	測試以請參閱字串是否為英文字母字串
<code>s.isnum</code>	測試以請參閱字串是否為數值字串
<code>s.isupper</code>	測試以請參閱字串是否為全部大寫
<code>s.islower</code>	測試以請參閱字串是否為全部小寫

表 6: 字串方法 (繼續)

方法	用法
<code>s.isspace</code>	測試以請參閱字串是否全是空格
<code>s.istitle</code>	測試以請參閱字串是否為首字母大寫的英數字串序列
<code>s.lower()</code> <code>s.upper()</code> <code>s.swapcase()</code> <code>s.title()</code>	轉換為全部小寫 轉換為全部大寫 轉換為觀察值顛倒 轉換為全部首字母大寫
<code>s.join(seq)</code>	將 <code>seq</code> 中的字串結合起來，以 <code>s</code> 作為分隔字元
<code>s.splitlines({keep})</code>	將 <code>s</code> 分割為多行，如果 <code>keep</code> 為 <code>true</code> ，那麼將使用換行
<code>s.split({sep {, max}})</code>	使用 <code>sep</code> (預設 <code>sep</code> 為空格) 將 <code>s</code> 分割為「單字」，最多分割 <code>max</code> 次
<code>s.ljust(width)</code> <code>s.rjust(width)</code> <code>s.center(width)</code> <code>s.zfill(width)</code>	在寬度為 <code>width</code> 的欄位中，將字串靠左對齊 在寬度為 <code>width</code> 的欄位中，將字串靠右對齊 在寬度為 <code>width</code> 的欄位中，將字串居中排列整齊 用 0 進行填充。
<code>s.lstrip()</code> <code>s.rstrip()</code> <code>s.strip()</code>	移除前導空格 移除尾部空格 移除前導和尾部空格
<code>s.translate(str {, delc})</code>	在移除 <code>delc</code> 中的任何字元之後，使用表格來轉換 <code>s</code> 。 <code>str</code> 應該是長度為 <code>== 256</code> 的字串。
<code>s.replace(old, new {, max})</code>	使用字串 <code>new</code> 取代所有或 <code>max</code> 個 <code>old</code> 字串

備註

註解是由井號 (或井字號) # 所建立的註解。同一行上井號之後的所有文字都會被視為註解的一部分，且會被忽略。備註可以從何欄開始。下列範例說明備註的使用：

```
#The HelloWorld application is one of the most simple
print 'Hello World' # print the Hello World line
```

陳述式語法

Python 的陳述式語法非常簡單。通常，每個原始碼行都是單一陳述式。除 `expression` 和 `assignment` 陳述式外，每個陳述式都由一個關鍵字名稱 (例如 `if` 或 `for`) 引入。可以在代碼中任何陳述式之間的任意位置插入空白行或備註行。如果某一行中有多個陳述式，那麼必須使用分號 (;) 來分隔每個陳述式。

超長陳述式可以分為多行。在這種情況下，要分到下一行的陳述式必須以反斜線 (\) 結尾，例如：

```
x = "A loooooooooooooooooooooooooong string" + \
    "another loooooooooooooooooooooooooong string"
```

如果某個結構括在圓括弧 (())、方括弧 ([]) 或大括弧 ({}) 內，那麼陳述式可以在任何逗點後面分為新行，而不必插入反斜線，例如：

```
x = (1, 2, 3, "hello",
    "goodbye", 4, 5, 6)
```

ID

ID 用於對變數、函數、類別和關鍵字進行命名。ID 可以是任意長度，但必須以大寫或小寫字母字元或底線字元 (`_`) 開頭。以底線開頭的名稱通常會保留給內部或專用名稱。在第一個字元後面，ID 可以包含任意數字，以及英文字母、0 到 9 的數字和底線字元的組合。

Jython 中的一些保留單字不可用於對變數、函數或類別進行命名。這些保留字分為下列種類：

- **陳述式引導詞：** `assert`、`break`、`class`、`continue`、`def`、`del`、`elif`、`else`、`except`、`exec`、`finally`、`for`、`from`、`global`、`if`、`import`、`pass`、`print`、`raise`、`return`、`try` 和 `while`
- **參數引導詞：** `as`、`import` 和 `in`
- **運算子：** `and`、`in`、`is`、`lambda`、`not` 和 `or`

關鍵字使用不當通常會生成 `SyntaxError`。

代碼區塊

代碼區塊是在預期的單個陳述式的位置使用的陳述式群組。代碼區塊可以跟隨下列任何陳述式：`if`、`elif`、`else`、`for`、`while`、`try`、`except`、`def` 和 `class`。這些陳述式將引入帶有冒號字元 (`:`) 的代碼區塊，例如：

```
if x == 1:
    y = 2
    z = 3
elif:
    y = 4
    z = 5
```

使用縮排對代碼區塊進行定界，而不是像 Java 一樣使用大括弧。代碼區塊中的所有行都必須縮排到同一位置。這是因為對縮排的變更指示代碼區塊結束。通常，每一級縮排四個空格。建議使用空格而不是 Tab 來縮排行。不得混用空格和 Tab。模組的最外層區塊中的行必須從第一欄開始，否則將發生 `SyntaxError`。

組成代碼區塊的陳述式（以及後面的冒號）也可以包括在一行中，並以分號分隔，例如：

```
if x == 1: y = 2; z = 3;
```

將引數傳送給 Script

將引數傳送給 Script 非常有用，因為這表示可以重複使用 Script 而無需進行修改。在指令行中傳送的引數將作為清單 `sys.argv` 中的值進行傳送。使用指令 `len(sys.argv)` 可以獲取所傳送值的數目。例如：

```
import sys
print "test1"
print sys.argv[0]
print sys.argv[1]
print len(sys.argv)
```

在此範例中，`import` 指令用於匯入整個 `sys` 類別，以便可以使用這個類別中存在的方法，例如 `argv`。

可以使用下列行呼叫此範例中的 Script：

```
/u/mjloos/test1 mike don
```

結果為下列輸出：

```
/u/mjloos/test1 mike don
test1
mike
don
3
```

範例

`print` 關鍵字將列印緊跟其後的引數。如果陳述式後跟逗點，那麼不會在輸出中新增一行。例如：

```
print "This demonstrates the use of a",  
print " comma at the end of a print statement."
```

這將導致下列輸出：

```
This demonstrates the use of a comma at the end of a print statement.
```

`for` 陳述式用於疊代代碼區塊。例如：

```
mylist1 = ["one", "two", "three"]  
for lv in mylist1:  
    print lv  
    continue
```

在此範例中，將為清單 `mylist1` 分配 3 個字串。然後，將列印該清單的元素，每個元素佔用一行。這將導致下列輸出：

```
one  
two  
three
```

在此範例中，反覆運算子 `lv` 將依次採用清單 `mylist1` 中每個元素的值，因為 `for` 迴圈用於實作每個元素的代碼區塊。反覆運算子可以是任意長度的任何有效 ID。

`if` 陳述式是條件式陳述式。該陳述式將對條件進行求值，並根據求值結果傳回 `true` 或 `false`。例如：

```
mylist1 = ["one", "two", "three"]  
for lv in mylist1:  
    if lv == "two"  
        print "The value of lv is ", lv  
    else  
        print "The value of lv is not two, but ", lv  
    continue
```

在此範例中，對反覆運算子 `lv` 進行了求值。如果 `lv` 的值为 `two`，那麼將傳回一個字串，該字串不同於 `lv` 的值不是 `two` 時傳回的字串。這會導致下列輸出：

```
The value of lv is not two, but one  
The value of lv is two  
The value of lv is not two, but three
```

數學方法

您可以從 `math` 模組存取有用的數學方法。下表中會提供一些方法。除非另有說明，否則所有值將作為浮點數傳回。

表 7: 數學方法	
方法	用法
<code>math.ceil(x)</code>	將 <code>x</code> 的上限作為浮點數傳回，即大於或等於 <code>x</code> 的最小整數
<code>math.copysign(x, y)</code>	傳回帶有 <code>y</code> 符號的 <code>x</code> 。 <code>copysign(1, -0.0)</code> 傳回 <code>-1</code>
<code>math.fabs(x)</code>	傳回 <code>x</code> 的絕對值
<code>math.factorial(x)</code>	傳回 <code>x</code> 階乘。如果 <code>x</code> 是負數或非整數，那麼將發生 <code>ValueError</code> 。

表 7: 數學方法 (繼續)

方法	用法
<code>math.floor(x)</code>	將 x 的下限作為浮點數傳回，即小於或等於 x 的最大整數
<code>math.frexp(x)</code>	傳回 x 的假數 (m) 和指數 (e) 作為配對 (m, e)。 m 是浮點數， e 是整數， 這樣剛好滿足 $x == m * 2^{**}e$ 。 如果 x 為零， 則傳回 $(0.0, 0)$ ， 否則傳回 $0.5 <= \text{abs}(m) < 1$ 。
<code>math.fsum(iterable)</code>	傳回 <code>iterable</code> 中值的精確浮點總和
<code>math.isinf(x)</code>	檢查浮點數 x 是正不定式還是負不定式
<code>math.isnan(x)</code>	檢查浮點數 x 是否為 NaN (非數字)
<code>math.ldexp(x, i)</code>	傳回 $x * (2^{**}i)$ 。 此方法本質上是函數 <code>frexp</code> 的反函數。
<code>math.modf(x)</code>	傳回 x 的小數和整數部分。 這兩個結果都帶有 x 的符號， 並且都是浮點數。
<code>math.trunc(x)</code>	傳回已截斷為 Integral 的 Real 值 x 。
<code>math.exp(x)</code>	傳回 $e^{**}x$
<code>math.log(x[, base])</code>	傳回以給定值 <code>base</code> 為底的 x 的對數。 如果未指定 <code>base</code> ， 那麼將傳回 x 的自然對數。
<code>math.log1p(x)</code>	傳回 $1+x$ (base e) 的自然對數
<code>math.log10(x)</code>	傳回以 10 為底的 x 的對數
<code>math.pow(x, y)</code>	傳回 x 次方 y 。 即使 x 是零或 NaN， <code>pow(1.0, x)</code> 和 <code>pow(x, 0.0)</code> 一律會傳回 1。
<code>math.sqrt(x)</code>	傳回 x 的平方根

除數學函數外， 還提供了一些有用的三角函數法。 下表格列出了這些方法。

表 8: 三角方法

方法	用法
<code>math.acos(x)</code>	傳回以弧度表示的 x 的反餘弦
<code>math.asin(x)</code>	傳回以弧度表示的 x 的正弦
<code>math.atan(x)</code>	傳回以弧度表示的 x 的正切
<code>math.atan2(y, x)</code>	傳回以弧度表示的 <code>atan(y / x)</code> 。
<code>math.cos(x)</code>	傳回以弧度表示的 x 的餘弦。
<code>math.hypot(x, y)</code>	傳回歐基里得範數 $\text{sqrt}(x*x + y*y)$ 。 這是從原點到點 (x, y) 的向量的長度。
<code>math.sin(x)</code>	傳回以弧度表示的 x 的正弦
<code>math.tan(x)</code>	傳回以弧度表示的 x 的正切
<code>math.degrees(x)</code>	將角度 x 從弧度轉換為度
<code>math.radians(x)</code>	將角度 x 從度轉換為弧度
<code>math.acosh(x)</code>	傳回 x 的反雙曲餘弦值

表 8: 三角方法 (繼續)	
方法	用法
<code>math.asinh(x)</code>	傳回 x 的反雙曲正弦值
<code>math.atanh(x)</code>	傳回 x 的反雙曲線正切值
<code>math.cosh(x)</code>	傳回 x 的雙曲餘弦值
<code>math.sinh(x)</code>	傳回 x 的雙曲餘弦值
<code>math.tanh(x)</code>	傳回 x 的雙曲線正切值

還有兩個數學常數。 `math.pi` 的值為數學常數 pi。 `math.e` 的值為數學常數 e。

使用非 ASCII 字元

要使用非 ASCII 字元，Python 需要明確地將字串編碼和解碼為 Unicode。在 IBM SPSS Modeler 中，假定 Python Script 採用 UTF-8 進行編碼，這是支援非 ASCII 字元的標準 Unicode 編碼。下列 Script 將執行編譯，這是因為 SPSS Modeler 已將 Python 編譯器設定為 UTF-8。

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", "テストノード", 96, 64)
```

但是，產生的節點將具有不正確的標籤。



圖 3: 錯誤顯示的包含非 ASCII 字元的節點標記

標籤不正確，因為 Python 已將字串文字自身轉換為 ASCII 字串。

Python 通過在字串文字前新增 u 字元字首來支援指定 Unicode 字串文字：

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", u"テストノード", 96, 64)
```

這將建立 Unicode 字串，並且將正確顯示標籤。



圖 4: 正確顯示的包含非 ASCII 字元的節點標記

使用 Python 和 Unicode 是一個非常大的主題，它超出了本文件的範圍。提供了多數對此主題進行了更詳細介紹的書籍和線上資源。

面向物件的程式設計

面向物件的程式設計基於在程式中創建目標問題模型的概念。面向物件的程式設計減少了程式設計錯誤並促進了代碼的重複使用。Python 是一種面向物件的語言。以 Python 定義的物件具有下列特徵：

- **身分模型。** 每個物件都必須截然不同，並且必須可以對此特徵進行測試。is 和 is not 測試可用於此目的。
- **狀態。** 每個物件都必須能夠儲存狀態。屬性（例如欄位和實例變數）可用於此目的。
- **行為。** 每個物件都必須能夠操作其狀態。方法可用於此目的。

Python 提供了支援面向物件的程式設計的下列特徵：

- **基於類別的物件建立。** 類別是用於建立物件的範本。物件是具有關聯行為的資料結構。
- **多型性繼承。** Python 支援單繼承和多重繼承。所有 Python 實例方法都具有多型性，並且可以由子類別置換。
- **具有隱藏資料的封裝。** Python 容許隱藏屬性。隱藏後，將只能通過類別的方法從類別外部存取這些屬性。類別實作了用於修改資料的方法。

定義類別

在 Python 類別中，可以定義變數和方法。與 Java 不同，您可以在 Python 中對每個來源檔案（或模組）定義任意數目的公用類別。因此，可以認為 Python 中的模組類似於 Java 中的套件。

在 Python 中，類別是使用 class 陳述式定義的。class 陳述式的格式如下：

```
class name (superclasses): statement
```

或

```
class name (superclasses):  
    assignment  
    .  
    .  
    function  
    .  
    .
```

定義類別時，您可以選擇提供零個以上的指派陳述式。這些陳述式將建立該類別的所有實例共用的類別屬性。您還可以提供零個以上的函數定義。這些函數定義將建立方法。超類別清單是選用的。

在同一範圍中（即模組、函數或類別中），類別名稱應該唯一。您可以將多重變數定義為參照同一類別。

建立類別實例

類別用於儲存類別（或共用）屬性，或者用於建立類別實例。要建立某個類別的實例，請將該類別作為函數進行呼叫。例如，請考慮下列類別：

```
class MyClass:  
    pass
```

這裡使用了 pass 陳述式，因為需要一個陳述式來完成類別，但不需要以程式設計方式執行任何動作。

下列陳述式將建立類別 MyClass 的實例：

```
x = MyClass()
```

向類別實例新增屬性

與 Java 不同，用戶端可以在 Python 中向類別實例新增屬性。只有一個實例會發生變更。例如，要向實例 x 新增屬性，請在該實例上設定新值：

```
x.attr1 = 1  
x.attr2 = 2
```

```
.  
.  
x.attrN = n
```

定義類別屬性和方法

任何鏈結在類別中的變數都是類別屬性。任何在類別中定義的函數都是方法。方法接收類別的實例（通常稱為 `self`）作為第一個引數。例如，要定義一些類別屬性和方法，您可以輸入下列代碼：

```
class MyClass  
    attr1 = 10          #class attributes  
    attr2 = "hello"  
  
    def method1(self):  
        print MyClass.attr1  #reference the class attribute  
  
    def method2(self):  
        print MyClass.attr2  #reference the class attribute  
  
    def method3(self, text):  
        self.text = text      #instance attribute  
        print text, self.text #print my argument and my attribute  
  
    method4 = method3  #make an alias for method3
```

在類別中，您應該使用類別名稱限定所有對類別屬性的參照，例如 `MyClass.attr1`。應該使用 `self` 變數限定所有對實例屬性的參照；例如 `self.text`。在類別外部，您應該使用類別名稱（例如 `MyClass.attr1`）或使用類別的實例（例如 `x.attr1`，其中 `x` 是類別的實例）限定所有對類別屬性的參照。在類別外部，應該使用類別的實例限定所有對實例變數的參照；例如 `x.text`。

隱藏變數

可以通過建立專用變數來隱藏資料。專用變數只能由類別自身進行存取。如果您宣告格式為 `__xxx` 或 `__xxx_yyy`（即，帶有兩個前導底線）的名稱，那麼 Python 剖析器將自動向宣告的名稱新增類別名稱以建立隱藏變數，例如：

```
class MyClass:  
    __attr = 10  #private class attribute  
  
    def method1(self):  
        pass  
  
    def method2(self, p1, p2):  
        pass  
  
    def __privateMethod(self, text):  
        self.__text = text  #private attribute
```

與 Java 不同，在 Python 中必須使用 `self` 限定所有對實例變數的參照；未隱含地使用 `this`。

繼承

從類別進行繼承的功能是面向物件的程式設計的基礎。Python 同時支援單繼承和多重繼承。單繼承表示只能存在一個超類別。多重繼承表示可以存在多個超類別。

繼承是透過對其他類別設定子類別來實作。任意數目的 Python 類別都可以是超類別。在 Python 的 Jython 實作中，只能從一個 Java 類別進行直接或間接繼承。無需提供超類別。

超類別中的任何屬性或方法也包含在任何子類別中，並且只要屬性或方法未隱藏，類別自身或任何用戶端就可以使用這些屬性或方法。可以在任何位置使用子類別的任何實例，並且可以使用超類別的實例；這是多型性範例。這些特徵支持重複使用和輕鬆延伸。

範例

```
class Class1: pass  #no inheritance  
  
class Class2: pass  
  
class Class3(Class1): pass  #single inheritance
```



```
class Class4(Class3, Class2): pass    #multiple inheritance
```


第 3 章 在 IBM SPSS Modeler 中進行 Scripting

Script 類型

在 IBM SPSS Modeler 中，有 3 種類型的 Script：

- 串流 *Script*，用於控制單個串流的執行並且它儲存在串流中。
- *SuperNode Script*，用於控制 SuperNode 的行為。
- 獨立式或階段作業 *Script*，可用於跨多個不同的串流協調執行。

提供了在 IBM SPSS Modeler 中的 Script 內使用的多種方法，您可以使用這些方法來存取各種 SPSS Modeler 功能。另外，這些方法還在第 35 頁的『第 4 章 Scripting API』中用於建立更進階的函數。

串流、SuperNode 串流和圖表

多數情況下，串流這個詞表示同一內容，無論是從檔案載入的串流還是 SuperNode 中使用的串流都是如此。通常情況下，它表示連接在一起並可執行的節點收集。但是，Script 中的作業並非在所有地方都受支援，這表示 Script 編寫人員應該知道他們正在使用哪些串流變式。

串流

串流是主 IBM SPSS Modeler 文件類型。可對其進行儲存、載入、編輯和執行。串流還可以有參數、廣域值、Script、以及與其相關聯的其他資訊。

SuperNode 串流

SuperNode 串流是 SuperNode 中所用的串流類型。與標準串流一樣，它包含鏈結在一起的節點。SuperNode 串流與標準串流存在若干不同之處：

- 參數以及任何 Script 與具有 SuperNode 串流的 SuperNode 相關聯，而不是與 SuperNode 串流本身相關聯。
- SuperNode 串流有附加的輸入與輸出連接器節點，這取決於 SuperNode 的類型。這些連接器節點用於將資訊導入和衍生 SuperNode 串流，並且在建立 SuperNode 之後會自動建立這些連接器節點。

圖表

圖表這個詞涵蓋標準串流和 SuperNode 串流均支援的功能，例如新增和刪除節點，以及修改節點之間的連線。

執行串流

下列範例將執行串流中的所有可執行節點，並且它是最簡單的串流 Script 類型：

```
modeler.script.stream().runAll(None)
```

下列範例也將執行串流中的所有可執行節點：

```
stream = modeler.script.stream()
stream.runAll(None)
```

在此範例中，串流儲存在名為 `stream` 的變數中。將串流儲存在變數中非常有用，因為 Script 通常用於修改串流或串流中的節點。創建用於儲存串流的變數將生成一個更加簡潔的 Script。

Scripting 環境定義

`modeler.script` 模組提供了在其中執行 Script 的環境定義。此模組將在執行時期自動匯入 SPSS Modeler Script 中。此模組定義了以下 4 個函數，這些函數提供可以存取其執行環境的 Script：

- `session()` 函數，用於傳回 Script 的階段作業。此階段作業定義語言環境等資訊以及將用於執行任何串流的 SPSS Modeler 後端（本端處理程序或聯網 SPSS Modeler Server）。
- `stream()` 函數，此函數可以與串流 Script 及 SuperNode Script 配合使用。此函數將傳回擁有正在執行的串流 Script 或 SuperNode Script 的串流。
- `diagram()` 函數，此函數可以與 SuperNode Script 配合使用。此函數將傳回 SuperNode 內的圖。對於其他 Script 類型，此函數傳回的內容與 `stream()` 函數相同。
- `supernode()` 函數，此函數可以與 SuperNode Script 配合使用。此函數將傳回擁有正在執行的 Script 的 SuperNode。

下表概述了這四個函數及其輸出。

Script 類型	<code>session()</code>	<code>stream()</code>	<code>diagram()</code>	<code>supernode()</code>
單獨	傳回階段作業	在呼叫該 Script 時傳回目前受管串流（例如，通過批次方式 <code>-stream</code> 選項傳送的串流）或 <code>None</code> 。	與 <code>stream()</code> 相同	不適用
串流	傳回階段作業	傳回串流	與 <code>stream()</code> 相同	不適用
SuperNode	傳回階段作業	傳回串流	傳回 SuperNode 串流	傳回 SuperNode

`modeler.script` 模組還定義了一種使用結束碼終止 Script 的方式。`exit(exit-code)` 函數可防止執行 Script 並傳回所提供的整數結束碼。

為串流定義的其中一種方法是 `runAll(List)`。此方法將執行所有可執行節點。執行節點時產生的任何模型或輸出都將新增到所提供的清單中。

串流執行通常產生模型和圖形等輸出以及其他輸出。若要擷取此輸出，Script 可以向清單提供一個起始設定的變數，例如：

```
stream = modeler.script.stream()
results = []
stream.runAll(results)
```

執行完成後，可以從 `results` 清單存取執行所產生的任何物件。

參照現有節點

串流通常預先建置了一些參數，在執行串流之前必須先修改這些參數。修改這些參數涉及下列作業：

1. 在相關串流中找到節點。
2. 變更節點和/或串流設定。

尋找節點

串流提供了多種查找現有節點的方式。這些方法在下表進行彙總。

表 10: 用來尋找現有節點的方法

方法	傳回類型	說明
<code>s.findAll(type, label)</code>	集合	傳回具有指定類型和標籤的所有節點的清單。類型或標籤可以為 <code>None</code> ，在這種情況下將使用其他參數。
<code>s.findAll(filter, recursive)</code>	集合	傳回指定過濾器所接受的所有節點的集合。如果遞歸旗標為 <code>True</code> ，那麼還將搜尋指定串流內的任何 <code>SuperNode</code> 。
<code>s.findById(id)</code>	節點	傳回具有所提供 ID 的節點或 <code>None</code> （如果不存在此類節點）。搜尋範圍限制為目前串流。
<code>s.findByType(type, label)</code>	節點	傳回具有所提供類型和/或標籤的節點。類型或名稱可以為 <code>None</code> ，在這種情況下將使用其他參數。如果有多個節點相符，那麼將選擇並傳回任意一個節點。如果沒有相符的節點，那麼回覆值為 <code>None</code> 。
<code>s.findDownstream(fromNodes)</code>	集合	從所提供的節點清單中進行搜尋，並傳回所提供節點下游的節點集合。傳回的清單包括原始提供的節點。
<code>s.findUpstream(fromNodes)</code>	集合	從所提供的節點清單中進行搜尋，並傳回所提供節點上游的節點集合。傳回的清單包括原始提供的節點。
<code>s.findProcessorForID(String id, boolean recursive)</code>	節點	傳回具有所提供 ID 的節點或 <code>None</code> （如果不存在此類節點）。如果遞迴旗標為 <code>true</code> ，則也會搜尋此圖表內的所有複合節點。

例如，如果串流包含 Script 需要存取的單個「過濾器」節點，那麼可以使用下列 Script 來找到該「過濾器」節點：

```
stream = modeler.script.stream()
node = stream.findByType("filter", None)
...
```

另外，如果已經知道節點的 ID（如節點對話框的「註釋」標籤中所示），那麼可以使用此 ID 來尋找節點，例如：

```
stream = modeler.script.stream()
node = stream.findById("id32FJT71G2") # the filter node ID
...
```

設定內容

節點、串流、模型和輸出都具有可以存取並在大多數情況下可以設定的內容。這些內容通常用於修改物件的行為或外觀。下表概述了可用於存取和設定物件內容的方法。

表 11: 用來存取及設定物件內容的方法

方法	傳回類型	說明
<code>p.getPropertyValue(propertyName)</code>	物件	傳回指定內容的值或 None (如果不存在此內容)。
<code>p.setPropertyValue(propertyName, value)</code>	不適用	設定指定內容的值。
<code>p.setPropertyValues(properties)</code>	不適用	設定指定內容的值。內容圖中的每個項目都包含一個代表內容名的鍵，以及應指定給該內容的值。
<code>p.getKeyedPropertyValue(propertyName, keyName)</code>	物件	傳回指定內容的值及關聯的鍵或 None (如果不存在此內容或鍵)。
<code>p.setKeyedPropertyValue(propertyName, keyName, value)</code>	不適用	設定指定內容的值和鍵。

例如，如果要設定位於串流的開始位置的「變數檔案」節點的值，那麼可以使用下列 Script:

```
stream = modeler.script.stream()
node = stream.findByType("variablefile", None)
node.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")
...
```

或者，您可能希望根據「過濾器」節點來過濾欄位。在這種情況下，還將根據欄位名稱鍵入值，例如：

```
stream = modeler.script.stream()
# Locate the filter node ...
node = stream.findByType("filter", None)
# ... and filter out the "Na" field
node.setKeyedPropertyValue("include", "Na", False)
```

建立節點以及修改串流

在某些狀況下，您可能希望向現有串流新增新節點。向現有串流新增節點通常涉及下列作業：

1. 建立節點。
2. 將節點鏈結到現有串流。

建立節點

串流提供了多種建立節點的方式。這些方法在下表進行彙總。

表 12: 用來建立節點的方法

方法	傳回類型	說明
<code>s.create(nodeType, name)</code>	節點	建立具有指定類型的節點並將其新增到指定的串流中。
<code>s.createAt(nodeType, name, x, y)</code>	節點	建立具有指定類型的節點並將其新增到指定串流中的指定位置。如果 $x < 0$ 或 $y < 0$ ，那麼未設定位置。
<code>s.createModelApplier(modelOutput, name)</code>	節點	建立衍生自所提供的模型輸出物件的模型套用者節點。

例如，要在串流中建立新的「類型」節點，您可以使用下列 Script：

```
stream = modeler.script.stream()
# Create a new type node
node = stream.create("type", "My Type")
```

鏈結和取消鏈結節點

在串流中建立新節點時，這個新節點必須先連接到節點序列，然後才可使用。串流提供了多種鏈結節點和取消鏈結節點的方法。這些方法在下表進行彙總。

方法	傳回類型	說明
<code>s.link(source, target)</code>	不適用	在來源節點與目標節點之間建立新鏈結。
<code>s.link(source, targets)</code>	不適用	在來源節點與所提供清單中的每個目標節點之間建立新鏈結。
<code>s.linkBetween(inserted, source, target)</code>	不適用	連接兩個其他節點實例（來源節點和目標節點）之間的節點，並將已插入節點的位置設定為位於這兩個節點實例之間。將首先移除來源節點與目標節點之間的任何直接鏈結。
<code>s.linkPath(path)</code>	不適用	在節點實例之間建立新路徑。第一個節點將鏈結至第二個節點，而第二個節點將鏈結至第三個節點，依此類推。
<code>s.unlink(source, target)</code>	不適用	移除來源節點與目標節點之間的任何直接鏈結。
<code>s.unlink(source, targets)</code>	不適用	移除來源節點與目標清單中每個物件之間的任何直接鏈結。
<code>s.unlinkPath(path)</code>	不適用	移除節點實例之間存在的任何路徑。
<code>s.disconnect(node)</code>	不適用	移除所提供節點與指定串流中任何其他節點之間的任何鏈結。
<code>s.isValidLink(source, target)</code>	<i>boolean</i>	如果在指定的來源節點與目標節點之間建立鏈結是有效的，那麼此方法將傳回 <code>True</code> 。此方法將檢查這兩個物件是否的指定串流，來源節點是否可以提供鏈結以及目標節點是否可以接收鏈結，並確認建立此類鏈結不會在串流中引起循環。

下面的範例 Script 將執行 5 項作業：

1. 建立「變數檔案」輸入節點、「過濾器」節點和「表格」輸出節點。
2. 將這些節點連接到一起。
3. 在「變數檔案」輸入節點上設定檔案名稱。
4. 根據生成的輸出過濾欄位「藥品」。

5. 執行「表格」節點。

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", "My File Input ", 96, 64)
filternode = stream.createAt("filter", "Filter", 192, 64)
tablenode = stream.createAt("table", "Table", 288, 64)
stream.link(filenode, filternode)
stream.link(filternode, tablenode)
filenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
filternode.setKeyedPropertyValue("include", "Drug", False)
results = []
tablenode.run(results)
```

匯入、替換和刪除節點

與建立和連接節點一樣，通常需要取代和刪除串流中的節點。下表格概述了可用於匯入、取代和刪除節點的方法。

方法	傳回類型	說明
<code>s.replace(originalNode, replacementNode, discardOriginal)</code>	不適用	取代指定串流中的指定節點。原始節點和取代節點都必須的指定串流。
<code>s.insert(source, nodes, newIDs)</code>	清單	在所提供的清單中插入節點的副本。假定所提供清單中的所有節點都包含在指定串流中。 <code>newIDs</code> 旗標指示應該為每個節點產生新 ID，還是應該複製並使用現有 ID。假定串流中所有節點都具有唯一 ID，這樣在來源串流與指定串流相同的情況下，必須將此旗標設定為 <code>True</code> 。此方法將傳回新插入節點的清單，此清單中未定義節點的順序（即，順序不一定與輸入清單中節點的順序相同）。
<code>s.delete(node)</code>	不適用	從指定串流中刪除指定節點。該節點必須的指定串流。
<code>s.deleteAll(nodes)</code>	不適用	從指定串流中刪除所有指定節點。收集中的所有節點都必須的指定串流。
<code>s.clear()</code>	不適用	從指定串流中刪除所有節點。

遍訪串流中的節點

識別特定節點上游或下游節點是一項常見需求。串流提供了一些可用於識別這些節點的方法。這些方法在下表進行彙總。

方法	傳回類型	說明
<code>s.iterator()</code>	反覆運算子	傳回針對指定串流中包含的節點物件的反覆運算子。如果在 <code>next()</code> 函數的兩次呼叫之間對串流進行了修改，那麼將取消定義反覆運算子的行為。

表 15: 用來識別上游及下游節點的方法 (繼續)

方法	傳回類型	說明
<code>s.predecessorAt(node, index)</code>	節點	傳回所提供節點的指定直接前置作業或 <code>None</code> (如果索引超出範圍)。
<code>s.predecessorCount(node)</code>	<i>int</i>	傳回所提供節點的直接前置作業數。
<code>s.predecessors(node)</code>	清單	傳回所提供節點的直接前置作業。
<code>s.successorAt(node, index)</code>	節點	傳回所提供節點的指定直接後置作業或 <code>None</code> (如果索引超出範圍)。
<code>s.successorCount(node)</code>	<i>int</i>	傳回所提供節點的直接後置作業數。
<code>s.successors(node)</code>	清單	傳回所提供節點的直接後置作業。

清除或除去項目

舊的 Scripting 支援 `clear` 指令的各種用法，例如：

- `clear outputs` 用於從管理程式選用區中刪除所有輸出項目。
- `clear generated palette` 用於從模型選用區中清除所有模型區塊。
- `clear stream` 用於移除串流的內容。

Python Scripting 支援一組相似的函數；`removeAll()` 指令用於清除串流、輸出以及模型管理程式。例如：

- 清除串流管理器：

```
session = modeler.script.session()
session.getStreamManager.removeAll()
```

- 清除輸出管理器：

```
session = modeler.script.session()
session.getDocumentOutputManager().removeAll()
```

- 清除模型管理器：

```
session = modeler.script.session()
session.getModelOutputManager().removeAll()
```

取得節點的相關資訊

節點分為數個不同的種類，例如資料匯入和匯出節點、模型建置節點和其他類型的節點。每個節點都提供了一些可用於尋找該節點的相關資訊的方法。

下表概述了可用於獲取節點的 ID、名稱和標籤的方法。

表 16: 用來取得節點之 ID、名稱及標籤的方法

方法	傳回類型	說明
<code>n.getLabel()</code>	<i>string</i>	傳回指定節點的顯示標籤。只有在內容 <code>custom_name</code> 是非空白字串並且內容 <code>use_custom_name</code> 未進行設定的情況下，標籤才是前一內容的值；否則，標籤是 <code>getName()</code> 的值。
<code>n.setLabel(label)</code>	不適用	設定指定節點的顯示標籤。如果新標籤為非空白字串，那麼會將其指定給內容 <code>custom_name</code> ，並且會將 <code>False</code> 指定給內容 <code>use_custom_name</code> ，以使指定的標籤優先；否則，會將空字串指定給內容 <code>custom_name</code> ，並將 <code>True</code> 指定給內容 <code>use_custom_name</code> 。
<code>n.getName()</code>	<i>string</i>	傳回指定節點的名稱。
<code>n.getID()</code>	<i>string</i>	傳回指定節點的 ID。每次建立新節點時都會建立一個新 ID。將節點作為串流的組成部分進行儲存時，ID 將隨節點一起持續儲存，以便在開啟串流時保留節點 ID。但是，如果將已儲存的節點插入串流中，那麼會將已插入的節點視為新物件並為其配置一個新 ID。

下表格概述了可用於獲取節點的其他相關資訊的方法。

表 17: 用來取得節點之相關資訊的方法

方法	傳回類型	說明
<code>n.getTypeName()</code>	<i>string</i>	傳回此節點的 Scripting 名稱。此名稱即為可用於建立該節點的新實例的名稱。
<code>n.isInitial()</code>	布林	如果此節點是初始節點（即，位於串流的開始位置的節點），那麼此方法將傳回 <code>True</code> 。
<code>n.isInline()</code>	布林	如果此節點是行內節點（即，位於串流中部的節點），那麼此方法將傳回 <code>True</code> 。
<code>n.isTerminal()</code>	布林	如果此節點是終端節點（即，位於串流的結束位置的節點），那麼此方法將傳回 <code>True</code> 。
<code>n.getXPosition()</code>	<i>int</i>	傳回節點在串流中的 x 位置偏移量。
<code>n.getYPosition()</code>	<i>int</i>	傳回節點在串流中的 y 位置偏移量。
<code>n.setXYPosition(x, y)</code>	不適用	設定節點在串流中的位置。

表 17: 用來取得節點之相關資訊的方法 (繼續)

方法	傳回類型	說明
<code>n.setPositionBetween(source, target)</code>	不適用	設定節點在串流中的位置，以使其位於所提供的節點之間。
<code>n.isCacheEnabled()</code>	布林	如果已啟用快取，那麼此方法將傳回 <code>True</code> ，否則將傳回 <code>False</code> 。
<code>n.setCacheEnabled(val)</code>	不適用	對此物件啟用或取消快取。如果快取已滿並且快取功能進入停用狀態，那麼快取將進行清倉。
<code>n.isCacheFull()</code>	布林	如果快取已滿，則傳回 <code>True</code> ，否則將傳回 <code>False</code> 。
<code>n.flushCache()</code>	不適用	清空此節點的快取。如果快取未啟用或者未滿，那麼此方法不起任何作用。

第 4 章 Scripting API

Scripting API 簡介

Scripting API 提供對各種不同的 SPSS Modeler 功能的存取。目前說明的所有方法是 API 的組成部分，可以在 Script 內隱式地存取這些方法而不執行進一步的匯入。但是，如果要參照 API 類別，則必須使用下列陳述式明確地匯入 API：

```
import modeler.api
```

此匯入陳述式是多數 Scripting API 範例所必要的。

在《IBM SPSS Modeler Python Scripting API 參考手冊》文件中，可以找到透過 Scripting API 提供的類別、方法和參數的完整手冊。

範例 1：使用自訂過濾器搜尋節點

第 26 頁的『尋找節點』一節提供了關於使用節點的類型名稱作為搜尋準則在串流中搜尋節點的範例。在某些狀況下，需要執行更加通用的搜尋，並且此搜尋可使用 NodeFilter 類別和串流 findAll() 方法來實作。這種搜尋包括下列兩個步驟：

1. 建立用於延伸 NodeFilter 並實作 accept() 方法的自訂版本的新類別。
2. 使用此新類別的實例來呼叫串流 findAll() 方法。這將傳回所有符合 accept() 方法中定義的準則的節點。

下列範例顯示如何在串流中搜尋已啟用節點快取的節點。所傳回的節點清單可用於清除或停用這些節點的快取。

```
import modeler.api

class CacheFilter(modeler.api.NodeFilter):
    """A node filter for nodes with caching enabled"""
    def accept(this, node):
        return node.isCacheEnabled()

cachingnodes = modeler.script.stream().findAll(CacheFilter(), False)
```

範例 2：容許使用者基於其專用權獲取目錄或檔案資訊

為避免向使用者開啟 PSAPI，可以通過呼叫 PSAPI 函數建立檔案系統物件來使用名為 session.getServerFileSystem() 的方法。

下列範例顯示了如何容許使用者基於連接至 IBM SPSS Modeler Server 的使用者的專用權來取得目錄或檔案資訊。

```
import modeler.api
stream = modeler.script.stream()
sourceNode = stream.findByID('')
session = modeler.script.session()
fileSystem = session.getServerFileSystem()
parameter = stream.getParameterValue('VPATH')
serverDirectory = fileSystem.getServerFile(parameter)
files = fileSystem.getFiles(serverDirectory)
for f in files:
    if f.isDirectory():
        print 'Directory:'
    else:
        print 'File:'
        sourceNode.setPropertyValue('full_filename', f.getPath())
        break
    print f.getName(), f.getPath()
stream.execute()
```

meta 資料：有關資料的資訊

由於串流中的節點相互連接在一起，因此能夠提供每個節點中可用欄或欄位的資訊。例如，在 Modeler UI 中，您可以使用此資訊來選取要排序或聚合的欄位。此資訊被稱為資料模型。

Script 還可以透過查看進出節點的欄位來存取資料模型。對於某些節點，輸入與輸出資料模型是相同的，例如，「排序」節點僅對記錄進行重新排序，但不變更資料模型。某些節點（例如，「衍生」節點）可新增新的欄位。而其他節點（例如，「過濾器」節點）可重新命名或移除欄位。

在下列範例中，Script 採用標準 IBM SPSS Modeler druglearn.str 串流，並針對每一個欄位建置一個模型，其中一個輸入欄位已捨棄。按以下操作執行：

1. 從「類型」節點存取輸出資料模型。
2. 對輸出資料模型中的每個欄位進行迴圈。
3. 修改每個輸入欄位的「過濾器」節點。
4. 變更要建立的模型名稱。
5. 執行模式建置節點。

註：在 druglearn.str 串流中執行 Script 之前，請記得將 Scripting 語言設定為 Python（該串流是在舊版 IBM SPSS Modeler 中建立的，因此串流 Scripting 語言設定為 Legacy）。

```
import modeler.api

stream = modeler.script.stream()
filternode = stream.findByType("filter", None)
typenode = stream.findByType("type", None)
c50node = stream.findByType("c50", None)
# Always use a custom model name
c50node.setPropertyValue("use_model_name", True)

lastRemoved = None
fields = typenode.getOutputDataModel()
for field in fields:
    # If this is the target field then ignore it
    if field.getModelingRole() == modeler.api.ModelingRole.OUT:
        continue

    # Re-enable the field that was most recently removed
    if lastRemoved != None:
        filternode.setKeyedPropertyValue("include", lastRemoved, True)

    # Remove the field
    lastRemoved = field.getColumnName()
    filternode.setKeyedPropertyValue("include", lastRemoved, False)

    # Set the name of the new model then run the build
    c50node.setPropertyValue("model_name", "Exclude " + lastRemoved)
    c50node.run([])
```

DataModel 物件提供了存取資料模型中欄位資訊或欄資訊的多種方法。這些方法在下表進行彙總。

方法	傳回類型	說明
d.getColumnCount()	int	傳回資料模型中的欄個數。
d.columnIterator()	反覆運算子	傳回按「自然」插入順序傳回各欄的反覆運算子。反覆運算子傳回欄實例。
d.nameIterator()	反覆運算子	傳回按「自然」插入順序傳回各欄名稱的反覆運算子。
d.contains(name)	布林	如果此 DataModel 中存在含有所供名稱的欄，那麼傳回 True，否則傳回 False。

表 18: 用來存取欄位或直欄的相關資訊的 *DataModel* 物件方法 (繼續)

方法	傳回類型	說明
<code>d.getColumn(name)</code>	直條圖	傳回含有指定名稱的欄。
<code>d.getColumnGroup(name)</code>	ColumnGroup	傳回指定的欄群組，如果不存在此類欄群組，則傳回 None。
<code>d.getColumnGroupCount()</code>	<i>int</i>	傳回此資料模型中的欄群組數。
<code>d.columnGroupIterator()</code>	反覆運算子	依次傳回用於傳回各個欄群組的反覆運算子。
<code>d.toArray()</code>	直欄[]	按欄陣列傳回資料模型。按「自然」插入順序對欄進行排序。

每個欄位（欄物件）包含用於存取欄資訊的多種方法。下方表格顯示了這些方法選擇。

表 19: 用來存取直欄的相關資訊的 *Column* 物件方法

方法	傳回類型	說明
<code>c.getColumnName()</code>	<i>string</i>	傳回欄名稱。
<code>c.getColumnLabel()</code>	<i>string</i>	傳回欄的標籤，如果不存在與欄相關聯的任何標籤，則傳回空白字串。
<code>c.getMeasureType()</code>	MeasureType	傳回欄的測量類型。
<code>c.getStorageType()</code>	StorageType	傳回欄的儲存類型。
<code>c.isMeasureDiscrete()</code>	布林	如果是離散欄，那麼傳回 True。會將作為集合或旗標的欄視為離散欄。
<code>c.isModelOutputColumn()</code>	布林	如果是模型輸出欄，那麼傳回 True。
<code>c.isStorageDatetime()</code>	布林	如果欄的儲存為時間、日期或時間戳記值，那麼傳回 True。
<code>c.isStorageNumeric()</code>	布林	如果欄的儲存是整數或實數，那麼傳回 True。
<code>c.isValidValue(value)</code>	布林	如果指定的值對此儲存有效，那麼傳回 True，如果已知有效的欄值，那麼傳回 valid。
<code>c.getModelingRole()</code>	ModelingRole	傳回欄的建模角色。
<code>c.getSetValues()</code>	物件[]	傳回欄的有效值陣列，如果值不明或者欄不是集合，則傳回 None。
<code>c.getValueLabel(value)</code>	<i>string</i>	傳回欄中值的標籤，如果不存在與值相關聯的任何標籤，那麼傳回空字串。
<code>c.getFalseFlag()</code>	物件	傳回欄的 "false" 指標值，如果值不明或者欄不是旗標，則傳回 None。
<code>c.getTrueFlag()</code>	物件	傳回欄的 "true" 指標值，如果值不明或者欄不是旗標，則傳回 None。

表 19: 用來存取直欄的相關資訊的 Column 物件方法 (繼續)

方法	傳回類型	說明
<code>c.getLowerBound()</code>	物件	傳回欄中值的下限值，如果值未知或者欄不是連續的，則傳回 None。
<code>c.getUpperBound()</code>	物件	傳回欄中值的上限值，如果值不明或者欄不是連續的，則傳回 None。

請注意，存取欄資訊的大多數方法在 DataModel 物件本身中定義了等效方法。例如，下列兩個陳述式是等效的：

```
dataModel.getColumn("someName").getModelingRole()
dataModel.getModelingRole("someName")
```

存取產生的物件

執行某個串流通常涉及產生其他輸出物件。這些附加物件可能是新模型，也可能是提供要在後續執行中使用的資訊的輸出。

在以下範例中，`druglearn.str` 串流再次用作串流的起點。在此範例中，將執行串流中的所有節點，並且結果將儲存在清單中。然後，Script 會迴圈執行結果，執行後產生的任何模型輸出都會儲存為 IBM SPSS Modeler 模型 (.gm) 檔案，且模型會匯出為 PMML。

```
import modeler.api

stream = modeler.script.stream()

# Set this to an existing folder on your system.
# Include a trailing directory separator
modelFolder = "C:/temp/models/"

# Execute the stream
models = []
stream.runAll(models)

# Save any models that were created
taskrunner = modeler.script.session().getTaskRunner()
for model in models:
    # If the stream execution built other outputs then ignore them
    if not(isinstance(model, modeler.api.ModelOutput)):
        continue

    label = model.getLabel()
    algorithm = model.getModelDetail().getAlgorithmName()

    # save each model...
    modelFile = modelFolder + label + algorithm + ".gm"
    taskrunner.saveModelToFile(model, modelFile)

    # ...and export each model PMML...
    modelFile = modelFolder + label + algorithm + ".xml"
    taskrunner.exportModelToFile(model, modelFile, modeler.api.FileFormat.XML)
```

作業執行者類別提供了一種便捷方式來執行各項一般作業。下表概述了此類別中提供的方法。

表 20: 用來執行一般作業的作業執行器類別的方法

方法	傳回類型	說明
<code>t.createStream(name, autoConnect, autoManage)</code>	串流	建立並傳回新的串流。請注意，必須以不公開方式建立串流而不向使用者顯示這些串流的代碼應該將 <code>autoManage</code> 旗標設定為 <code>False</code> 。

表 20: 用來執行一般作業的作業執行器類別的方法 (繼續)

方法	傳回類型	說明
<code>t.exportDocumentToFile(documentOutput, filename, fileFormat)</code>	不適用	使用指定的檔案格式將串流說明匯出至檔案。
<code>t.exportModelToFile(modelOutput, filename, fileFormat)</code>	不適用	使用指定的檔案格式將模型匯出至檔案。
<code>t.exportStreamToFile(stream, filename, fileFormat)</code>	不適用	使用指定的檔案格式將串流匯出至檔案。
<code>t.insertNodeFromFile(filename, diagram)</code>	節點	從指定檔案中讀取並傳回節點，然後將其插入所提供的圖中。請注意，此方法可用於同時讀取節點物件和 SuperNode 物件。
<code>t.openDocumentFromFile(filename, autoManage)</code>	文件輸出	從指定檔案讀取並傳回文件。
<code>t.openModelFromFile(filename, autoManage)</code>	模型輸出	從指定檔案讀取並傳回模型。
<code>t.openStreamFromFile(filename, autoManage)</code>	串流	從指定檔案讀取並傳回串流。
<code>t.saveDocumentToFile(documentOutput, filename)</code>	不適用	將文件儲存到指定的檔案位置。
<code>t.saveModelToFile(modelOutput, filename)</code>	不適用	將模型儲存到指定的檔案位置。
<code>t.saveStreamToFile(stream, filename)</code>	不適用	將串流儲存到指定的檔案位置。

處理錯誤

Python 語言提供了透過 `try...except` 代碼區塊來處理錯誤的方法。可以在 Script 內使用此方法來捕獲例外狀況，並處理將導致 Script 終止的問題。

在以下範例 Script 中，已試圖從 IBM SPSS Collaboration and Deployment Services Repository 中擷取模型。此作業可能會導致擲出例外狀況，例如可能未正確設定儲存庫登入認證或者儲存庫路徑錯誤。在 Script 中，這可能會導致擲出 `ModelerException` (IBM SPSS Modeler 產生的所有異常狀況都衍生自 `modeler.api.ModelerException`)。

```
import modeler.api

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # print goes to the Modeler UI script panel Debug tab
    print "Everything OK"
except modeler.api.ModelerException, e:
    print "An error occurred:", e.getMessage()
```

註: 某些 Scripting 作業可能會導致擲出標準 Java 例外狀況; 這些例外狀況並非衍生自 `ModelerException`。要捕獲這些例外狀況, 可以使用附加的 `except` 區塊來捕獲所有 Java 例外狀況, 例如:

```
import modeler.api
import java.lang.Exception

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # print goes to the Modeler UI script panel Debug tab
    print "Everything OK"
except modeler.api.ModelerException, e:
    print "An error occurred:", e.getMessage()
except java.lang.Exception, e:
    print "A Java exception occurred:", e.getMessage()
```

Stream、Session 及 SuperNode 參數

參數提供了一種有用方法, 用於在執行時期傳送值而不是直接在 Script 中將這些值寫入程式中。參數及其值的定義方式與串流相同, 即定義為串流或 SuperNode 的參數表格中的項目或指令行中的參數。串流類別和 SuperNode 類別實作了一組由 `ParameterProvider` 物件定義的函數, 如下表格所示。階段作業提供了 `getParameters()` 呼叫, 此呼叫將傳回定義這些函數的物件。

方法	傳回類型	說明
<code>p.parameterIterator()</code>	反覆運算子	傳回此物件的參數名稱的反覆運算子。
<code>p.getParameterDefinition(parameterName)</code>	參數定義	傳回具有指定名稱的參數的參數定義或 <code>None</code> (如果此提供者中不存在此類參數)。結果可以是呼叫此方法時的定義 Snapshot, 並且不需要反映通過此提供者對該參數進行的任何後續修改。
<code>p.getParameterLabel(parameterName)</code>	<code>string</code>	傳回指定參數的標籤或 <code>None</code> (如果不存在此類參數)。
<code>p.setParameterLabel(parameterName, label)</code>	不適用	設定指定參數的標籤。
<code>p.getParameterStorage(parameterName)</code>	參數儲存	傳回指定參數的儲存或 <code>None</code> (如果不存在此類參數)。
<code>p.setParameterStorage(parameterName, storage)</code>	不適用	設定指定參數的儲存。
<code>p.getParameterType(parameterName)</code>	參數類型	傳回指定參數的類型或 <code>None</code> (如果不存在此類參數)。
<code>p.setParameterType(parameterName, type)</code>	不適用	設定指定參數的類型。
<code>p.getParameterValue(parameterName)</code>	物件	傳回指定參數的值或 <code>None</code> (如果不存在此類參數)。
<code>p.setParameterValue(parameterName, value)</code>	不適用	設定指定參數的值。

在下列範例中，Script 聚合了一些 Telco 資料以尋找具有最低平均值收入資料的區域。然後，將使用此區域設定一個串流參數。接下來，將在「選取」節點中使用該串流參數從資料中排除此區域，然後根據剩餘的資料建立流動模型。

此範例並不真實，這是因為 Script 本身產生了「選取」節點，並因此已經在「選取」節點表示式中直接產生正確的值。但是，串流通常是預先構建的，因此通過這種方式設定參數提供了有用的範例。

範例 Script 的第一部分用於建立串流參數，該串流參數將包含平均值收入最低的區域。另外，此 Script 還將在聚集分支和模型建置分支中建立節點，並將這些節點連接在一起。

```
import modeler.api

stream = modeler.script.stream()

# Initialize a stream parameter
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# First create the aggregation branch to compute the average income per region
statisticsimportnode = stream.createAt("statisticsimport", "SPSS File", 114, 142)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/telco.sav")
statisticsimportnode.setPropertyValue("use_field_format_for_storage", True)

aggregatenode = modeler.script.stream().createAt("aggregate", "Aggregate", 294, 142)
aggregatenode.setPropertyValue("keys", ["region"])
aggregatenode.setKeyedPropertyValue("aggregates", "income", ["Mean"])

tablenode = modeler.script.stream().createAt("table", "Table", 462, 142)

stream.link(statisticsimportnode, aggregatenode)
stream.link(aggregatenode, tablenode)

selectnode = stream.createAt("select", "Select", 210, 232)
selectnode.setPropertyValue("mode", "Discard")
# Reference the stream parameter in the selection
selectnode.setPropertyValue("condition", "'region' = '$P-LowestRegion'")

typenode = stream.createAt("type", "Type", 366, 232)
typenode.setKeyedPropertyValue("direction", "churn", "Target")

c50node = stream.createAt("c50", "C5.0", 534, 232)

stream.link(statisticsimportnode, selectnode)
stream.link(selectnode, typenode)
stream.link(typenode, c50node)
```

此範例 Script 將建立下列串流。

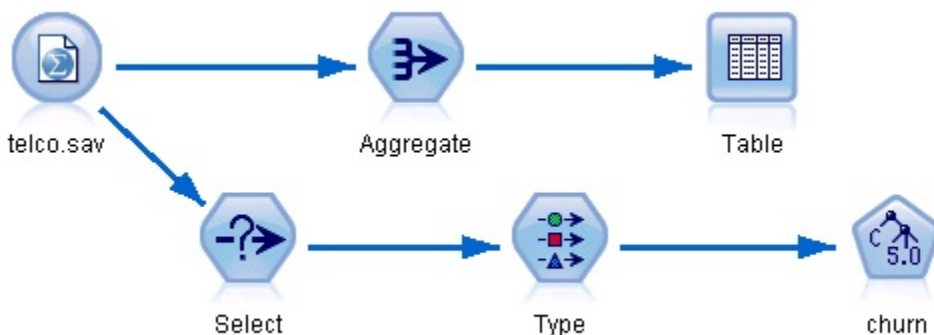


圖 5: 範例 Script 生成的串流

範例 Script 的下列部分用於執行位於聚集分支末尾的「表格」節點。

```
# First execute the table node
results = []
tablenode.run(results)
```

範例 Script 的下列部分用於存取執行「表格」節點所產生的表格輸出。隨後，此 Script 將對表格中的各列執行反覆運算，以查找平均值收入最低的區域。

```
# Running the table node should produce a single table as output
table = results[0]

# table output contains a RowSet so we can access values as rows and columns
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

# From the way the aggregate node is defined, the first column
# contains the region and the second contains the average income
row = 0
rowcount = rowset.getRowCount()
while row < rowcount:
    if rowset.getValueAt(row, 1) < min_income:
        min_income = rowset.getValueAt(row, 1)
        min_region = rowset.getValueAt(row, 0)
    row += 1
```

Script 的下列部分使用平均值收入最低的區域來設定先前建立的 "LowestRegion" 串流參數。然後，在從訓練資料中排除了指定區域的情況下，此 Script 將執行模型建置器。

```
# Check that a value was assigned
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)

# Finally run the model builder with the selection criteria
c50node.run([])
```

完整的範例 Script 如下方所示。

```
import modeler.api

stream = modeler.script.stream()

# Create a stream parameter
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# First create the aggregation branch to compute the average income per region
statisticsimportnode = stream.createAt("statisticsimport", "SPSS File", 114, 142)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/telco.sav")
statisticsimportnode.setPropertyValue("use_field_format_for_storage", True)

aggregatenode = modeler.script.stream().createAt("aggregate", "Aggregate", 294, 142)
aggregatenode.setPropertyValue("keys", ["region"])
aggregatenode.setKeyedPropertyValue("aggregates", "income", ["Mean"])

tablenode = modeler.script.stream().createAt("table", "Table", 462, 142)

stream.link(statisticsimportnode, aggregatenode)
stream.link(aggregatenode, tablenode)

selectnode = stream.createAt("select", "Select", 210, 232)
selectnode.setPropertyValue("mode", "Discard")
# Reference the stream parameter in the selection
selectnode.setPropertyValue("condition", "'region' = '$P-LowestRegion'")

typenode = stream.createAt("type", "Type", 366, 232)
typenode.setKeyedPropertyValue("direction", "churn", "Target")

c50node = stream.createAt("c50", "C5.0", 534, 232)

stream.link(statisticsimportnode, selectnode)
stream.link(selectnode, typenode)
stream.link(typenode, c50node)

# First execute the table node
results = []
tablenode.run(results)

# Running the table node should produce a single table as output
table = results[0]
```

```

# table output contains a RowSet so we can access values as rows and columns
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

# From the way the aggregate node is defined, the first column
# contains the region and the second contains the average income
row = 0
rowcount = rowset.getRowCount()
while row < rowcount:
    if rowset.getValueAt(row, 1) < min_income:
        min_income = rowset.getValueAt(row, 1)
        min_region = rowset.getValueAt(row, 0)
    row += 1

# Check that a value was assigned
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)

# Finally run the model builder with the selection criteria
c50node.run([])

```

廣域值

廣域值用於計算指定欄位的各項彙總統計量。可以在串流內部的任何位置存取這些彙總值。在串流中，可以依名稱存取廣域值，這一點與串流參數相似。廣域值與串流參數的差異在於，關聯值將在「設定廣域值」節點執行時期自動進行更新，而不是通過 Scripting 或指令行來指定。可以通過呼叫串流的 `getGlobalValues()` 方法來存取該串流的廣域值。

GlobalValues 物件定義下表格中顯示的函數。

方法	傳回類型	說明
<code>g.fieldNameIterator()</code>	反覆運算子	針對至少具有一個廣域值的每個欄位名稱，傳回一個反覆運算子。
<code>g.getValue(type, fieldName)</code>	物件	傳回指定類型和欄位名稱的廣域值或 None（如果找不到值）。雖然未來的功能可能會傳回各種值類型，但通常預期的回覆值為數字。
<code>g.getValues(fieldName)</code>	對映	傳回包含指定欄位名稱的已知項目的圖或 None（如果該欄位沒有現有項目）。

GlobalValues.Type 定義可用的彙總統計量的類型。可用的彙總統計量如下：

- MAX: 欄位的最大值。
- MEAN: 欄位的平均值。
- MIN: 欄位的最小值。
- STDDEV: 欄位的標準差。
- SUM: 欄位中值的總和。

例如，下列 Script 將存取「收入」欄位的平均數，此平均數由「設定廣域值」節點計算：

```

import modeler.api

globals = modeler.script.stream().getGlobalValues()
mean_income = globals.getValue(modeler.api.GlobalValues.Type.MEAN, "income")

```

使用多個串流 - 獨立式 Script

要使用多個串流，必須使用獨立式 Script。可以在 IBM SPSS Modeler UI 內編輯和執行獨立式 Script，也可以在批次模式下將獨立式 Script 作為指令行參數進行傳送。

下列獨立式 Script 將開啟兩個串流。其中一個串流用於建置模型，而第二個串流用於繪製預測值的分佈。

```
# Change to the appropriate location for your system
demosDir = "C:/Program Files/IBM/SPSS/Modeler/18.5.0/DEMOS/streams/"

session = modeler.script.session()
tasks = session.getTaskRunner()

# Open the model build stream, locate the C5.0 node and run it
buildstream = tasks.openStreamFromFile(demosDir + "druglearn.str", True)
c50node = buildstream.findByType("c50", None)
results = []
c50node.run(results)

# Now open the plot stream, find the Na_to_K derive and the histogram
plotstream = tasks.openStreamFromFile(demosDir + "drugplot.str", True)
derivenode = plotstream.findByType("derive", None)
histogramnode = plotstream.findByType("histogram", None)

# Create a model applier node, insert it between the derive and histogram nodes
# then run the histogram
applyc50 = plotstream.createModelApplier(results[0], results[0].getName())
applyc50.setPositionBetween(derivenode, histogramnode)
plotstream.linkBetween(applyc50, derivenode, histogramnode)
histogramnode.setPropertyValue("color_field", "$C-Drug")
histogramnode.run([])

# Finally, tidy up the streams
buildstream.close()
plotstream.close()
```

下列範例顯示還可以通過哪種方式對開啟的串流執行疊代（所有串流都在「串流」標籤中開啟）。請注意，只有獨立式 Script 才支援這種方式。

```
for stream in modeler.script.streams():
    print stream.getName()
```

第 5 章 Scripting 技巧

本章簡要介紹使用 Script 的技巧和技術，包括修改串流執行、在 Script 中採用編碼密碼以及存取 IBM SPSS Collaboration and Deployment Services 儲存庫中的物件等。

修改串流執行

執行串流時，將按預設情形下的最佳化順序來執行其終端節點。某些情況下，您可能更喜歡以其他順序來執行。要修改串流的執行順序，請在串流內容對話框的「執行」標籤上完成下列步驟：

1. 打開一個空白 Script。
2. 按一下工具列上的**附加預設 Script** 按鈕來新增預設串流 Script。
3. 將預設串流 Script 中陳述式的順序變更為您希望的執行順序。

對節點執行迴圈

您可以使用 for 迴圈對串流中的所有節點進行迴圈。例如，下列兩個 Script 範例用於對所有節點進行迴圈並將「過濾器」節點中的欄位名稱變更為大寫。

可以在具有「過濾器」節點的任何串流中使用此 Script，即使實際上不過濾任何欄位也是如此。只需新增傳送所有欄位的過濾器節點即可在整個面板上將欄位名稱變更為大寫。

```
# Alternative 1: using the data model nameIterator() function
stream = modeler.script.stream()
for node in stream.iterator():
    if (node.getTypeName() == "filter"):
        # nameIterator() returns the field names
        for field in node.getInputDataModel().nameIterator():
            newname = field.upper()
            node.setKeyedPropertyValue("new_name", field, newname)
```

```
# Alternative 2: using the data model iterator() function
stream = modeler.script.stream()
for node in stream.iterator():
    if (node.getTypeName() == "filter"):
        # iterator() returns the field objects so we need
        # to call getColumnName() to get the name
        for field in node.getInputDataModel().iterator():
            newname = field.getColumnName().upper()
            node.setKeyedPropertyValue("new_name", field.getColumnName(), newname)
```

此 Script 在目前串流的所有節點中進行迴圈，並檢查每個節點是否為過濾節點。如果是，那麼 Script 將迴圈該節點中的每個欄位，並使用 `field.upper()` 或 `field.getColumnName().upper()` 函數將名稱變更為大寫。

存取 IBM SPSS Collaboration and Deployment Services 儲存庫中的物件

如果您有 IBM SPSS Collaboration and Deployment Services 儲存庫的授權，那麼可以使用 Script 指令在儲存庫中儲存和擷取物件。通過使用此儲存庫，您可以在企業應用程式、工具和解決方案環境定義中管理資料採礦模型和相關預測物件的生命循環。

連接至 IBM SPSS Collaboration and Deployment Services 儲存庫

要存取儲存庫，必須首先通過 SPSS Modeler 使用者介面的工具功能表或通過指令行建立與該儲存庫的有效連線。如需相關資訊，請參閱第 61 頁的『[IBM SPSS Collaboration and Deployment Services 儲存庫 連線引數](#)』。

存取儲存庫

可以通過階段作業存取儲存庫，例如：

```
repo = modeler.script.session().getRepository()
```

從儲存庫中擷取物件

在 Script 中，使用 `retrieve*` 函數來存取各種物件，包括串流、模型、輸出和節點。下表格中顯示了關於擷取函數的摘要。

物件類型	儲存庫功能
串流	<code>repo.retrieveStream(String path, String version, String label, Boolean autoManage)</code>
型號	<code>repo.retrieveModel(String path, String version, String label, Boolean autoManage)</code>
輸出	<code>repo.retrieveDocument(String path, String version, String label, Boolean autoManage)</code>
節點 (Node)	<code>repo.retrieveProcessor(String path, String version, String label, ProcessorDiagram diagram)</code>

例如，您可以使用下列函數從儲存庫中擷取串流：

```
stream = repo.retrieveStream("/projects/retention/risk_score.str", None, "production", True)
```

此範例將從指定的資料夾中擷取 `risk_score.str` 串流。標籤 `production` 識別要擷取的串流版本，而最後一個參數指定 SPSS Modeler 要管理串流（例如，如果 SPSS Modeler 使用者介面可見，則串流會出現在串流標籤中）。作為替代方案，要使用未添加標籤的特定版本，請使用下列函數：

```
stream = repo.retrieveStream("/projects/retention/risk_score.str", "0:2015-10-12 14:15:41.281", None, True)
```

註：如果版本和標籤參數都為 `None`，那麼將傳回最新版本。

在儲存庫中儲存物件

要使用 *Scripting* 在儲存庫中儲存物件，請使用 `store*` 函數。下表格中顯示了關於 `store` 函數的摘要。

物件類型	儲存庫功能
串流	<code>repo.storeStream(ProcessorStream stream, String path, String label)</code>
型號	<code>repo.storeModel(ModelOutput modelOutput, String path, String label)</code>
輸出	<code>repo.storeDocument(DocumentOutput documentOutput, String path, String label)</code>
節點	<code>repo.storeProcessor(Processor node, String path, String label)</code>

例如，您可以使用下列函數儲存新版本的 `risk_score.str` 串流：

```
versionId = repo.storeStream(stream, "/projects/retention/risk_score.str", "test")
```

此範例將儲存新版本的串流，並將 `"test"` 標籤與此串流進行關聯，然後傳回新建立的版本的版本標示。

註：如果您不想將標籤與該新版本進行關聯，請對該標籤傳送 `None`。

管理儲存庫資料夾

通過使用儲存庫中的資料夾，您可以將物件編入邏輯群組中，從而更易于請參閱相關的物件。通過使用 `createFolder()` 函數建立資料夾，如下列範例中所示：

```
newpath = repo.createFolder("/projects", "cross-sell")
```

此範例將在 `/projects` 資料夾中建立名為 `"cross-sell"` 的新資料夾。該函數將傳回新資料夾的完整路徑。

要重新命名某個資料夾，請使用 `renameFolder()` 函數：

```
repo.renameFolder("/projects/cross-sell", "cross-sell-Q1")
```

第一個參數是要重新命名的資料夾的完整路徑，而第二個參數是指定給該資料夾的新名稱。

要刪除空白資料夾，請使用 `deleteFolder()` 函數：

```
repo.deleteFolder("/projects/cross-sell")
```

鎖定和解除鎖定物件

對於 Script，您可以鎖定一個物件，以防止其他使用者更新任一現有版本或建立新版本。還可以解除鎖定已鎖定的物件。

鎖定和解除鎖定物件的語法為：

```
repo.lockFile(REPOSITORY_PATH)
repo.lockFile(URI)

repo.unlockFile(REPOSITORY_PATH)
repo.unlockFile(URI)
```

對於儲存和擷取物件，`REPOSITORY_PATH` 指出物件在儲存庫中的位置。路徑必須用英文引號引起並以正斜線作為定界字元。路徑不區分觀察值。

```
repo.lockFile("/myfolder/Stream1.str")
repo.unlockFile("/myfolder/Stream1.str")
```

除此之外，還可以使用統一資源識別碼 (URI) 而非儲存庫路徑來給出物件的位置。URI 必須包含字首 `spsscr:`，同時必須完全括在引號中。只有正斜線可以作為路徑定界字元，空格必須以編碼形式出現。即在路徑中以 `%20` 代替空格。URI 不區分觀察值。範例如下：

```
repo.lockFile("spsscr:///myfolder/Stream1.str")
repo.unlockFile("spsscr:///myfolder/Stream1.str")
```

注意，物件鎖定適用於物件的所有版本 - 您無法鎖定或解除鎖定單個版本。

產生經過編碼的密碼

某些情況下，可能需要在 Script 中包含密碼，例如，您可能需要存取受密碼保護的資料來源。編碼密碼可用在：

- 資料庫來源和輸出節點的節點內容
- 登入到伺服器的指令行引數
- 儲存在 `.par` 檔案（由匯出節點的「發行」標籤產生的參數檔案）中的資料庫連線內容

通過此使用者介面，可以使用一個工具根據 Blowfish 演算法來產生編碼密碼（有關詳細資訊，請參閱 <http://www.schneier.com/blowfish.html>）。編碼後，可以複製密碼並將其儲存到 Script 檔案和指令行引數中。用於 `databasenode` 和 `databaseexportnode` 的節點內容 `epassword` 儲存編碼密碼。

1. 要產生編碼密碼，請從「工具」功能表中選擇：

密碼編碼...

2. 在「密碼」文字框中指定一個密碼。
3. 按一下 **編碼** 為自己的密碼產生一個隨機編碼。
4. 按一下「複製」按鈕將編碼密碼複製到剪貼簿。
5. 將此密碼貼上到所需的 Script 或參數中。

Script 檢查

通過按一下「獨立式 Script」對話框工具列上的紅色檢查按鈕，可以快速檢查所有類型 Script 的語法。



圖 6: 串流 Script 工具列圖示

Script 檢查將就您編碼中的錯誤發出警示並給出改善建議。要檢視錯誤行，請按一下該對話框下半部分的意見。此時將以紅色強調顯示錯誤。

從指令行執行 Scripting

通過編寫 Script 可以執行通常在使用者介面中執行的作業。啟動 IBM SPSS Modeler 時，只需在指令行中指定並執行一個單獨 Script。例如：

```
client -script scores.txt -execute
```

-script 旗標表示載入指定 Script，而 -execute 旗標表示執行該 Script 檔案中的所有指令。

與先前版本的相容性

在以前版本的 IBM SPSS Modeler 中建立的 Script 通常應該無需變更就可以在目前版本中運行。不過，模型區塊現在可以自動插入到串流中（此為預設設定），並可替代或補充串流中此類型的現有模型區塊。是否實際發生這種情況取決於將模型新增至串流和取代前一個模型選項（工具 > 選項 > 使用者選項 > 通知）的設定。例如，您可能需要修改以前版本中的 Script，在該版本中模型區塊取代是通過刪除現有模型區塊並插入新的模型區塊來完成。

在目前版本中建立的 Script 在以前的版本中可能無法正常運行。

如果在舊版本中建立的 Script 使用了已被取代（或已淘汰）的指令，則使用舊形式指令的 Script 仍然受支援，但將顯示一則警告訊息。例如，舊的 `generated` 關鍵字已被 `model` 取代，且 `clear generated` 已被 `clear generated palette` 取代。沿用舊形式的 Script 仍然可以執行，但將顯示一條警告消息。

存取串流執行結果

多數 IBM SPSS Modeler 節點生成輸出物件，例如模型、圖表和表格資料。其中許多輸出包含 Script 可用來引導後續執行的有用值。這些值群組到內容儲存器（簡稱為儲存器）中，您可以通過用於識別各個儲存器的標籤或識別來存取這些儲存器。存取這些值的方式取決於該儲存器所使用的格式或「內容模型」。

例如，多數預測模型輸出使用稱為 PMML 的 XML 變式來代表關於模型的資訊，例如決策樹狀結構在每個分割點使用哪些欄位，或者神經網路中的神經元如何連接以及以何種強度進行連接。使用 PMML 的模型輸出提供了可用於存取資訊的 XML 內容模型。例如：

```
stream = modeler.script.stream()
# Assume the stream contains a single C5.0 model builder node
# and that the datasource, predictors and targets have already been
# set up
modelbuilder = stream.findByType("c50", None)
results = []
modelbuilder.run(results)
modeloutput = results[0]

# check how many contents are there and what are their names
```

```

tags = modeloutput.getContentModelTags()
print "Content Model Tags :" , tags

# Now that we have the C5.0 model output object, access the
# relevant content model
cm = modeloutput.getContentModel("PMML")
if (cm != None) :
    # The PMML content model is a generic XML-based content model that
    # uses XPath syntax. Use that to find the names of the data fields.
    # The call returns a list of strings match the XPath values
    dataFieldNames = cm.getStringValues("/PMML/DataDictionary/DataField",
    "name")
    print "Data Field Names:" , dataFieldNames

```

在 Script 中，IBM SPSS Modeler 支援下列內容模型：

- **表格內容模型**用於存取以列和欄表示的簡單表格資料。
- **XML 內容模型**用於存取以 XML 格式儲存的內容。
- **JSON 內容模型**用於存取以 JSON 格式儲存的內容。
- **直欄統計量內容模型**用於存取有關特定欄位的彙總統計量。
- **成對直欄統計量內容模型**用於存取兩個欄位之間的彙總統計量或者兩個不同欄位之間的值。

請注意，下列節點不包含上述內容模型：

- 時間序列
- 區別
- SLRM
- TCM
- 所有 Python 節點
- 所有 Spark 節點
- 所有資料庫建模節點
- 延伸模型
- STP

表格內容模型

表格內容模型提供了一個簡單的模型，用於存取簡單的列資料和直欄資料。特定欄中的值必須全都採用同一類型的儲存（例如字串或整數）。

API

表 25: API		
傳回	方法	說明
int	getRowCount()	傳回這個表格中的列個數。
int	getColumnCount()	傳回這個表格中的欄個數。
String	getColumnName(int columnIndex)	傳回指定欄指標處的欄的名稱。直欄索引啟動於 0。
StorageType	getStorageType(int columnIndex)	傳回指定指標處的欄的儲存類型。直欄索引啟動於 0。
Object	getValueAt(int rowIndex, int columnIndex)	傳回指定列指標和欄指標處的值。列索引和欄索引起始於 0。

表 25: API (繼續)

傳回	方法	說明
void	reset()	清除與此內容模型相關聯的任何內部儲存體。

節點和輸出

此表格會列出建置包括此類型內容模型之輸出的節點。

表 26: 節點和輸出

節點名稱	輸出名稱	儲存器 ID
table	table	"table"

範例 Script

```

stream = modeler.script.stream()
from modeler.api import StorageType

# Set up the variable file import node
varfilenode = stream.createAt("variablefile", "DRUG Data", 96, 96)
varfilenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")

# Next create the aggregate node and connect it to the variable file node
aggregatenode = stream.createAt("aggregate", "Aggregate", 192, 96)
stream.link(varfilenode, aggregetenode)

# Configure the aggregate node
aggregatenode.setPropertyValue("keys", ["Drug"])
aggregatenode.setKeyedPropertyValue("aggregates", "Age", ["Min", "Max"])
aggregatenode.setKeyedPropertyValue("aggregates", "Na", ["Mean", "SDev"])

# Then create the table output node and connect it to the aggregate node
tablenode = stream.createAt("table", "Table", 288, 96)
stream.link(aggregatenode, tablenode)

# Execute the table node and capture the resulting table output object
results = []
tablenode.run(results)
tableoutput = results[0]

# Access the table output's content model
tablecontent = tableoutput.getContentModel("table")

# For each column, print column name, type and the first row
# of values from the table content
col = 0
while col < tablecontent.getColumnCount():
    print tablecontent.getColumnName(col), \
          tablecontent.getStorageType(col), \
          tablecontent.getValueAt(0, col)
    col = col + 1

```

在 Scripting 「除錯」 標籤中，輸出將類似於：

```

Age_Min Integer 15
Age_Max Integer 74
Na_Mean Real 0.730851098901
Na_SDev Real 0.116669731242

```

XML 內容模型

XML 內容模型用於存取基於 XML 的內容。

XML 內容模型支援存取基於 XPath 表示式的元件。XPath 表示式是一些字串，這些字串定義呼叫者所需的元素或屬性。XML 內容模型隱藏了構造各種物件以及編譯表示式的細節，而 XPath 支援人員通常需要這些細節。這使得從 Python Script 中進行呼叫更加簡單。

XML 內容模型包含一個以字串形式傳回 XML 文件的函數。這使 Python Script 使用者能夠使用其偏愛的 Python 程式庫來剖析 XML。

API

表 27: API		
傳回	方法	說明
String	getXMLAsString()	以字串形式傳回 XML。
number	getNumericValue(String xpath)	傳回評估路徑的結果，傳回類型為數值（例如，計算與路徑表示式相符的元素號碼）。
boolean	getBooleanValue(String xpath)	傳回對所指定路徑表示式求值所得的 Boolean 結果。
String	getStringValue(String xpath, String attribute)	傳回與指定路徑相符的屬性值或 XML 節點值。
List of strings	getStringValues(String xpath, String attribute)	傳回一個清單，其中包含所有與指定路徑相符的屬性值或 XML 節點值。
List of lists of strings	getValuesList(String xpath, <List of strings> attributes, boolean includeValue)	傳回一個清單，其中包含所有與指定路徑相符的屬性值，有需要時還包含 XML 節點值。
Hash table (key:string, value:list of string)	getValuesMap(String xpath, String keyAttribute, <List of strings> attributes, boolean includeValue)	傳回一個雜湊表，這個表格使用鍵屬性或 XML 節點值作為鍵，並使用一系列指定的屬性值作為表格值。
boolean	isNamespaceAware()	傳回 XML 剖析器是否應瞭解名稱空間。預設值是 False。
void	setNamespaceAware(boolean value)	設定 XML 剖析器是否應瞭解名稱空間。這還將呼叫 reset()，以確保後續呼叫採用所作的變更。
void	reset()	清除與此內容模型（例如，快取的 DOM 物件）相關聯的任何內部儲存體。

節點和輸出

此表格會列出建置包括此類型內容模型之輸出的節點。

表 28: 節點和輸出		
節點名稱	輸出名稱	儲存器 ID
Most model builders	Most generated models	"PMML"
"autodataprep"	n/a	"PMML"

範例 Script

用於存取內容的 Python Script 代碼可能如下所示：

```
results = []
modelbuilder.run(results)
modeloutput = results[0]
cm = modeloutput.getContentModel("PMML")

dataFieldNames = cm.getStringValues("/PMML/DataDictionary/DataField", "name")
predictedNames = cm.getStringValues("//MiningSchema/
MiningField[@usageType='predicted']", "name")
```

JSON 內容模型

JSON 內容模型用於提供對 JSON 格式內容的支援。此模型提供了基本 API，用於在假定呼叫者知道所要存取的值的情況下，容許呼叫者擷取值。

API

表 29: API		
傳回	方法	說明
String	getJSONAsString()	以字串形式傳回 JSON 內容。
Object	getObjectAt(<List of object> path, JSONArtifact artifact) throws Exception	傳回指定路徑處的物件。提供的根構件可能為空值，在這種情況下，將使用內容的根。傳回的值可能是字串、整數、實數或布林，或者是 JSON 構件（JSON 物件或 JSON 陣列）。
Hash table (key:object, value:object)	getChildValuesAt(<List of object> path, JSONArtifact artifact) throws Exception	如果指定的路徑指向 JSON 物件，那麼傳回該路徑的子代值，否則傳回空值。表格中的鍵是字串，而相關聯的值可以是字串、整數、實數或布林，或者是 JSON 構件（JSON 物件或 JSON 陣列）。
List of objects	getChildrenAt(<List of object> path path, JSONArtifact artifact) throws Exception	如果指定的路徑指向 JSON 陣列，那麼傳回該路徑處的物件的清單，否則傳回空值。傳回的值可能是字串、整數、實數或布林，或者是 JSON 構件（JSON 物件或 JSON 陣列）。
void	reset()	清除與此內容模型（例如，快取的 DOM 物件）相關聯的任何內部儲存體。

節點和輸出

此表格會列出建置包括此類型內容模型之輸出的節點。

表 30: 節點和輸出		
節點名稱	輸出名稱	儲存器 ID
"applykmeansas"	n/a	"JSON_MV"
"applyxgboosttree"	n/a	"JSON_MV"

範例 Script

下列 Script 會擷取 JSON 檔案：

```
applykmeansas = stream.findByType("applykmeansas",None)
mvjson = applykmeansas.getContentModel("JSON_MV")
print(mvjson.getJSONAsString())
```

```
applyxgboosttree = stream.findByType("applyxgboosttree",None)
mvjson = applyxgboosttree.getContentModel("JSON_MV")
print(mvjson.getJSONAsString())
```

欄統計資料內容模型和成對統計量內容模型

欄統計資料內容模型用於存取那些可以為每個欄位計算的統計資料（單變數統計量）。成對統計量內容模型用於存取可以在一成對欄位之間或者某個欄位中的一成對值之間計算的統計資料。

可能的統計資料測量如下所示：

- Count
- UniqueCount
- ValidCount
- Mean
- Sum
- Min
- Max
- Range
- Variance
- StandardDeviation
- StandardErrorOfMean
- Skewness
- SkewnessStandardError
- Kurtosis
- KurtosisStandardError
- Median
- Mode
- Pearson
- Covariance
- TTest

- FTest

某些值僅適用於單欄統計資料，而其他值僅適用於成對統計量。

生成這些值的節點包括：

- 「統計資料」節點生成欄統計資料，在指定了相關性欄位時，還可以生成成對統計量。
- 「資料審核」節點生成欄統計資料，在指定了覆蓋欄位時，還可以生成配對統計資料。
- 「平均數」節點在比較一成對欄位或者將某個欄位的值與其他欄位摘要進行比較時生成成對統計量。

可用的內容模型和統計資料既取決於特定節點的功能，也取決於該節點內的設定。

ColumnStatsContentModel API

傳回	方法	說明
List<StatisticType>	getAvailableStatistics()	傳回此模型中可用的統計資料。並非所有欄位都必須具有所有統計資料的值。
List<String>	getAvailableColumns()	傳回已計算其統計資料的直欄名稱。
Number	getStatistic(String column, StatisticType statistic)	傳回與該欄相關聯的統計量值。
void	reset()	清除與此內容模型相關聯的任何內部儲存體。

PairwiseStatsContentModel API

傳回	方法	說明
List<StatisticType>	getAvailableStatistics()	傳回此模型中可用的統計資料。並非所有欄位都必須具有所有統計資料的值。
List<String>	getAvailablePrimaryColumns()	傳回已計算其統計資料的主直欄名稱。
List<Object>	getAvailablePrimaryValues()	傳回已計算其統計資料的主欄的值。
List<String>	getAvailableSecondaryColumns()	傳回已計算其統計資料的輔助直欄名稱。
Number	getStatistic(String primaryColumn, String secondaryColumn, StatisticType statistic)	傳回與各個欄相關聯的統計量值。
Number	getStatistic(String primaryColumn, Object primaryValue, String secondaryColumn, StatisticType statistic)	傳回與主欄值和輔助欄相關聯的統計量值。

表 32: *PairwiseStatsContentModel* API (繼續)

傳回	方法	說明
void	reset()	清除與此內容模型相關聯的任何內部儲存體。

節點和輸出

此表格會列出建置包括此類型內容模型之輸出的節點。

表 33: 節點和輸出

節點名稱	輸出名稱	儲存器 ID	附註:
"means" (平均數節點)	"means"	"columnStatistics"	
"means" (平均數節點)	"means"	"pairwiseStatistics"	
"dataaudit" (「資料審核」節點)	"means"	"columnStatistics"	
"statistics" (統計資料節點)	"statistics"	"columnStatistics"	僅當檢查特定欄位時才會產生。
"statistics" (統計資料節點)	"statistics"	"pairwiseStatistics"	僅當關聯欄位時才會產生。

範例 Script

```

from modeler.api import StatisticType
stream = modeler.script.stream()

# Set up the input data
varfile = stream.createAt("variablefile", "File", 96, 96)
varfile.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")

# Now create the statistics node. This can produce both
# column statistics and pairwise statistics
statisticsnode = stream.createAt("statistics", "Stats", 192, 96)
statisticsnode.setPropertyValue("examine", ["Age", "Na", "K"])
statisticsnode.setPropertyValue("correlate", ["Age", "Na", "K"])
stream.link(varfile, statisticsnode)

results = []
statisticsnode.run(results)
statsoutput = results[0]
statscm = statsoutput.getContentModel("columnStatistics")
if (statscm != None):
    cols = statscm.getAvailableColumns()
    stats = statscm.getAvailableStatistics()
    print "Column stats:", cols[0], str(stats[0]), " = ",
    statscm.getStatistic(cols[0], stats[0])

```

```
statscm = statsoutput.getContentModel("pairwiseStatistics")
if (statscm != None):
    pcols = statscm.getAvailablePrimaryColumns()
    scols = statscm.getAvailableSecondaryColumns()
    stats = statscm.getAvailableStatistics()
    corr = statscm.getStatistic(pcols[0], scols[0], StatisticType.Pearson)
    print "Pairwise stats:", pcols[0], scols[0], " Pearson = ", corr
```

第 6 章 指令行引數

呼叫軟體

您可以使用作業系統的指令行來啟動 IBM SPSS Modeler，如下所示。

Microsoft Windows

1. 在安裝了 IBM SPSS Modeler 的電腦上，開啟 DOS 或命令提示字元視窗。
2. 切換至 IBM SPSS Modeler (例如， [Installpath]\Program Files\IBM\SPSS\Modeler\18.5\bin)。
3. 如果要以互動模式啟動 IBM SPSS Modeler 介面，請輸入 modelerclient 指令，後面接著必要的引數；例如：

```
modelerclient -stream report.str -execute
```

您可以使用可用的引數 (旗標) 來連接至伺服器、載入串流、執行 Script 或視需要指定其他參數。

Mac OS

1. 找出 IBM SPSS Modeler 的 Mac OS 指令路徑 (例如 [Installpath]/Applications/IBM/SPSS/Modeler/18.5/IBM SPSS Modeler.app/Contents/MacOS)。
2. 若要以互動模式啟動 IBM SPSS Modeler 介面，請執行後接必要引數的 Modeler 指令；例如：

```
./modeler -stream report.str -execute
```

使用指令行引數

您可以將指令行引數（也稱為旗標）附加至起始 modelerclient 指令，以變更 IBM SPSS Modeler 的呼叫。

存在多種可用的指令行引數類型，本節的隨後內容將對其進行說明。

引數類型	說明位置
系統引數	請參閱第 58 頁的『系統引數』主題，以取得更多資訊。
參數的引數	請參閱第 59 頁的『參數的引數』主題，以取得更多資訊。
伺服器連線引數	請參閱第 60 頁的『伺服器連線引數』主題，以取得更多資訊。
IBM SPSS Collaboration and Deployment Services 儲存庫 連線引數	請參閱第 61 頁的『IBM SPSS Collaboration and Deployment Services 儲存庫 連線引數』主題，以取得更多資訊。
IBM SPSS Analytic Server 連線引數	請參閱第 61 頁的『IBM SPSS Analytic Server 連線引數』主題，以取得更多資訊。

例如，可以使用 `-server`、`-stream` 和 `-execute` 旗標來連接至伺服器，然後載入並執行串流，如下所示：

```
modelerclient -server -hostname myserver -port 80 -username dminer
-password 1234 -stream mystream.str -execute
```

請注意，當針對本端用戶端安裝執行時，不需要輸入伺服器連線引數。

可以用雙引號括起包含空格的參數值，例如：

```
modelerclient -stream mystream.str -Pusername="Joe User" -execute
```

您也可以使用 `-state` 和 `-script` 旗標，以這種方式分別執行 IBM SPSS Modeler 狀態和 Script。

註：如果在指令中使用結構化參數，那麼必須在引號前面加上反斜線。這將阻止在解譯字串期間移除該引號。

除錯指令行引數

若要對指令行除錯，請使用 `modelerclient` 指令搭配所需的引數來啟動 IBM SPSS Modeler。這樣可以驗證指令是否將如預期般執行。您還可以在「工作階段參數」對話框（「工具」功能表 -> 「設定工作階段參數」）中對由指令行傳送的參數進行確認。

系統引數

下表說明可用於使用者介面指令行呼叫的系統引數。

表 35: 系統引數	
引數	行為/說明
@ <commandFile>	@ 符號後跟檔名，此檔案用於指定指令清單。當 <code>modelerclient</code> 遇到以 @ 開頭的引數時，它將在該檔案中對指令進行操作，就如同在指令行中一樣。如需相關資訊，請參閱第 62 頁的『組合多個引數』主題。
-directory <dir>	設定預設工作目錄。在本端模式下，該目錄同時用於存放資料和輸出。範例： <code>-directory c:/</code> 或 <code>-directory c:\\</code>
-server_directory <dir>	為資料設定預設伺服器目錄。通過 <code>-directory</code> 旗標指定的工作目錄將用於輸出。
-execute	在啟動後執行啟動時所載入的串流、狀態或 Script。如果在串流或狀態之外還載入了 Script，則 Script 將單獨執行。
-stream <stream>	啟動時載入指定的串流。可以指定多個串流，但是最後一個指定的串流將被設定為目前串流。
-stream_password <password>	啟動時，載入已加密密碼的串流。例如，您可以執行如下指令： <code>clem.exe -stream enc-stream1.str -stream_password Pass1234 -execute</code>
-script <script>	啟動時載入指定的獨立式 Script。如下方所述，除串流或狀態之外，此標記還可用於指定 Script，但在啟動時僅可載入一個 Script。
-model <model>	在啟動時載入指定的產生的模型（.gm 格式的檔案）。
-state <state>	在啟動時，載入指定的已儲存狀態。
-project <project>	載入指定專案。在啟動時僅可載入一個專案。
-output <output>	在啟動時載入已儲存的輸出項目（.cou 格式的檔案）。
-help	顯示指令行引數清單。指定此選項後，將忽略所有其他引數並顯示說明畫面。
-P <name>=<value>	用來設定啟動參數。也可用來設定節點內容（位置參數）。

註: 也可以在使用者介面中設定預設目錄。要存取上述選項, 請在「檔案」功能表中選擇設定工作目錄或設定伺服器目錄。

載入多個檔案

指令行模式下, 您可以通過重複輸入每個載入物件的相關引數, 在啟動時載入多個串流、狀態和輸出。例如, 要載入和執行兩個稱為 `report.str` 和 `train.str` 的串流, 您可以使用如下指令:

```
modelerclient -stream report.str -stream train.str -execute
```

從 IBM SPSS Collaboration and Deployment Services 儲存庫 載入物件

因為您可以從檔案或從 IBM SPSS Collaboration and Deployment Services 儲存庫 (如果已授權) 載入特定物件, 所以檔名字首 `spsscr:` 及 (選用) `file:` (針對磁碟上的物件) 會指示 IBM SPSS Modeler 尋找物件的位置。字首可與下列旗標配合使用:

- `-stream`
- `-script`
- `-output`
- `-model`
- `-project`

您可以使用字首建立 URI 以指定物件的位置, 例如 `-stream "spsscr:///folder_1/scoring_stream.str"`。如果存在 `spsscr:` 字首, 則需要在相同指令中指定與 IBM SPSS Collaboration and Deployment Services 儲存庫 的有效連線。因此, 完整的指令應形如以下的範例:

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080  
-spsscr_username myusername -spsscr_password mypassword  
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

注意, 在指令行中 必須使用 URI。不支援像 `REPOSITORY_PATH` 這樣的簡單路徑。(此種路徑僅適用於 Script。) 有關 IBM SPSS Collaboration and Deployment Services 儲存庫 中物件的 URI 的詳細資料, 請參閱第 45 頁的『存取 IBM SPSS Collaboration and Deployment Services 儲存庫 中的物件』主題。

參數的引數

參數可用作在 IBM SPSS Modeler 的指令行執行期間的旗標。在指令行引數中, 使用 `-P` 旗標, 以格式 `-P <name>=<value>` 表示參數。

參數可以是以下任一項:

- **簡單參數** (即, 直接在 CLEM 表示式中使用的參數)。
- **位置參數**, 也稱為節點內容。此類參數可用於修改串流中節點的設定。如需相關資訊, 請參閱第 65 頁的『節點內容概觀』。
- **指令行參數**, 用於更改對 IBM SPSS Modeler 的呼叫。

例如, 您可以提供資料來源使用者名稱和密碼作為指令行旗標, 如下所示:

```
modelerclient -stream response.str -P:databasenode.datasources="{\"ORA  
10gR2\",user1,mypsw,false}"
```

其格式與 `databasenode` 節點內容的 `datasources` 參數格式相同。如需相關資訊, 請參閱: 第 81 頁的『`databasenode` 內容』。

如果傳遞已編碼的密碼, 最後一個參數應設為 `true`。另請注意, 在資料庫使用者名稱及密碼前面不應使用前導空格 (當然, 除非您的資料庫使用者名稱及密碼實際包含前導空格)。

註: 如果指定此節點, 那麼必須將節點名稱括在雙引號內, 並使用反斜線對參照進行跳出。例如, 如果上述範例中的資料來源節點名為 Source_ABC, 那麼此項目如下所示:

```
modelerclient -stream response.str
-P:databasenode.\"Source_ABC\".datasource="{\"ORA 10gR2\",
user1,mypsw,true}"
```

用於識別結構化參數的參照前還需要有反斜線, 如下列 TM1 資料來源範例中所示:

```
clemb -server -hostname 9.115.21.169 -port 28053 -username administrator
-execute -stream C:\Share\TM1_Script.str -P:tm1import.pm_host="http://9.115.21.163:9510/
pmhub/pm"
-P:tm1import.tm1_connection="{\"SData\", \"\", \"admin\", \"apple\"}"
-P:tm1import.selected_view="{\"SalesPriorCube\", \"salesmargin%\"}"
```

註: 如果 datasource 內容中的資料庫名稱包含一個或多個空格、句點 (也稱為「句號」) 或底線, 那麼您可以使用「反斜線雙引號」格式將其作為字串進行處理。例如: "{ \"db2v9.7.6_linux\" }" 或 "{ \"TDATA 131\" }"。此外, 一律以雙引號和大括弧括住 datasource 字串值, 如下列範例所示: "{ \"SQL Server\", spssuser,abcd1234,false}"。

伺服器連線引數

-server 旗標會告知 IBM SPSS Modeler 應該連接至公用伺服器, 而旗標 -hostname、-use_ssl、-port、-username、-password 及 -domain 會用來告知 IBM SPSS Modeler 如何連接至公用伺服器。如果未指定 -server 引數, 則使用預設或本端伺服器。

範例

連接至公用伺服器:

```
modelerclient -server -hostname myserver -port 80 -username dminer
-password 1234 -stream mystream.str -execute
```

連接至伺服器叢集:

```
modelerclient -server -cluster "QA Machines" \
-spsscr_hostname pes_host -spsscr_port 8080 \
-spsscr_username asmith -spsscr_epassword xyz
```

請注意, 連接至伺服器叢集需要透過 IBM SPSS Collaboration and Deployment Services 的處理程序協調器, 因此 -cluster 引數必須與儲存庫連線選項 (spsscr_*) 一起使用。如需相關資訊, 請參閱主題 [第 61 頁的『IBM SPSS Collaboration and Deployment Services 儲存庫 連線引數』](#)。

引數	行為/說明
-server	以伺服器模式執行 IBM SPSS Modeler, 同時使用旗標 -hostname、-port、-username、-password 和 -domain 連接至公用伺服器。
-hostname <name>	伺服器的主機名稱。僅在伺服器模式中可用。
-use_ssl	指定連線應該使用 SSL (Secure Sockets Layer)。這是選用旗標; 預設值為不使用 SSL。
-port <number>	指定伺服器的埠號。僅在伺服器模式中可用。
-cluster <name>	指定指向伺服器叢集 (而不是已命名的伺服器) 的連線; 此引數可用來替代 hostname、port 和 use_ssl 引數。名稱為叢集名, 或識別 IBM SPSS Collaboration and Deployment Services 儲存庫中叢集的唯一 URI。伺服器叢集由 IBM SPSS Collaboration and Deployment Services 中的過程協調器管理。請參閱第 61 頁的『IBM SPSS Collaboration and Deployment Services 儲存庫 連線引數』主題, 以取得更多資訊。

表 36: 伺服器連線引數 (繼續)

引數	行為/說明
-username <name>	這是用於登入伺服器的使用者名稱。僅在伺服器模式中可用。
-password <password>	這是用於登入伺服器的密碼。僅在伺服器模式中可用。 註: 如果未使用 -password 引數, 那麼系統將提示您輸入密碼。
-epassword <encodedpasswordstring>	這是用於登入伺服器的經過編碼的密碼。僅在伺服器模式中可用。 註: 可以從 IBM SPSS Modeler 應用程式的「工具」功能表中產生編碼密碼。
-domain <name>	這是用於登入到伺服器的網域。僅在伺服器模式中可用。
-P <name>=<value>	用來設定啟動參數。也可用來設定節點內容 (位置參數)。

IBM SPSS Collaboration and Deployment Services 儲存庫 連線引數

如果想通過指令行來儲存或擷取 IBM SPSS Collaboration and Deployment Services 中的物件, 則必須指定一個指向該 IBM SPSS Collaboration and Deployment Services 儲存庫 的有效連線。例如:

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080
-spsscr_username myusername -spsscr_password mypassword
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

下表列出可用來設定連線的引數清單。

表 37: IBM SPSS Collaboration and Deployment Services 儲存庫 連線引數

引數	行為/說明
-spsscr_hostname <hostname or IP address>	安裝 IBM SPSS Collaboration and Deployment Services 儲存庫 的伺服器的主機名稱或 IP 位址。
-spsscr_port <number>	IBM SPSS Collaboration and Deployment Services 儲存庫 接受連線的埠號 (通常, 預設值為 8080)。
-spsscr_use_ssl	指定連線應該使用 SSL (Secure Sockets Layer)。這是選用旗標; 預設值為不使用 SSL。
-spsscr_username <name>	登入到 IBM SPSS Collaboration and Deployment Services 儲存庫 的使用者名稱。
-spsscr_password <password>	登入到 IBM SPSS Collaboration and Deployment Services 儲存庫 的密碼。
-spsscr_epassword <encoded password>	用來登入 IBM SPSS Collaboration and Deployment Services 儲存庫 的編碼密碼。
-spsscr_providername <name>	用於登入 IBM SPSS Collaboration and Deployment Services 儲存庫 的鑑別提供者 (Active Directory 或 LDAP)。如果使用原生 (本端儲存庫) 提供者, 那麼這不是必要的。

IBM SPSS Analytic Server 連線引數

如果要通過指令行儲存或擷取 IBM SPSS Analytic Server 中的物件, 那麼必須指定與 IBM SPSS Analytic Server 的有效連線。

註: 分析伺服器的預設位置是從 SPSS Modeler Server 獲取的。使用者也可以通過工具 > **Analytic Server 連線** 定義自己的分析伺服器 連線。

下表列出可用來設定連線的引數清單。

表 38: IBM SPSS Analytic Server 連線引數

引數	行為/說明
-analytic_server_username	用於登入 IBM SPSS Analytic Server 的使用者名稱。
-analytic_server_password	用於登入 IBM SPSS Analytic Server 的密碼。
-analytic_server_epassword	用於登入 IBM SPSS Analytic Server 的經過編碼的密碼。
-analytic_server_credential	用於登入 IBM SPSS Analytic Server 的認證。

組合多個引數

通過在檔名前使用 @ 符號，可以在呼叫時指定的指令檔中合併多個引數。這將使您可以縮短指令行呼叫，並且可以克服作業系統關於指令長度的限制。例如，下列啟動指令會使用 <commandFileName> 所參照檔案中指定的引數。

```
modelerclient @<commandFileName>
```

如果需用空格，則請用引號將指令檔的檔名和路徑括起來，如下所示：

```
modelerclient @ "C:\Program  
Files\IBM\SPSS\Modeler\nn\scripts\my_command_file.txt"
```

指令檔中可以包含在之前啟動中個別指定的所有引數，每行一個引數。例如：

```
-stream report.str  
-Porder.full_filename=APR_orders.dat  
-Preport.filename=APR_report.txt  
-execute
```

當寫入和參照指令檔時，必須遵循以下限制：

- 每條指令佔用一行。
- 不要在指令檔中內含 @CommandFile 引數。

第 7 章 內容參照資訊

內容參照概觀

可以為節點、串流、專案和 SuperNode 指定多個不同的內容。某些內容在所有節點中通用，例如「名稱」、「註釋」和「工具提示」，有些內容則只針對某些特定的節點類型。其他內容涉及高級串流作業，例如快取或 SuperNode 行為。可通過標準使用者介面（例如當開啟對話框編輯節點選項時）存取內容，還可以多種其他方式使用內容。

- 可通過 Script 修改內容（如本章所述）。如需相關資訊，請參閱第 63 頁的『內容語法』。
- 可在 SuperNode 參數中應用節點內容。
- 在啟動 IBM SPSS Modeler 時，節點內容也可以用作指令行選項的一部分（使用 -P 旗標）。

在 IBM SPSS Modeler 的 Script 編寫環境定義中，節點和串流內容通常稱為**位置參數**。在本手冊中，它們指的是節點或串流的內容。

內容語法

可以使用下列語法設定內容

```
OBJECT.setPropertyValue(PROPERTY, VALUE)
```

或：

```
OBJECT.setKeyedPropertyValue(PROPERTY, KEY, VALUE)
```

可以使用下列語法擷取內容的值：

```
VARIABLE = OBJECT.getPropertyValue(PROPERTY)
```

或：

```
VARIABLE = OBJECT.getKeyedPropertyValue(PROPERTY, KEY)
```

其中 OBJECT 是節點或輸出，PROPERTY 是表示式參照的節點內容的名稱，而 KEY 是含索引鍵內容的索引鍵值。例如，下列語法用於尋找過濾器節點，然後設定預設值以包含所有欄位並根據下游資料過濾 Age 欄位：

```
filternode = modeler.script.stream().findByType("filter", None)
filternode.setPropertyValue("default_include", True)
filternode.setKeyedPropertyValue("include", "Age", False)
```

使用串流 `findByType(TYPE, LABEL)` 函數可以找到 IBM SPSS Modeler 中使用的所有節點。必須至少指定一個 TYPE 或 LABEL。

結構化內容

Script 編寫通過結構化內容，增強語法剖析清晰度的方式有二：

- 指定複合節點內容名稱的結構，例如類型節點、過濾節點或平衡節點。
- 提供一種可一次指定多種內容的格式。

複合介面的結構化

帶有表格和其他複合介面的節點（類型、過濾和平衡節點）的 Script 必須遵循特定的結構以便正確剖析。這些內容需要的名稱比單個 ID 的名稱更複雜，此名稱為關鍵字。例如，過濾器節點內，每個可用欄位（在其

上游)均處於開啟或關的狀態。為了參照此資訊,「過濾器」節點為每個欄位(無論此欄位為 true 還是 false)儲存一個資訊項目。此內容的值可能為(或指定為) True 或 False。假設名為 mynode 的「過濾器」節點在其上游端具有名為 Age 的欄位。要關閉此字段,請將具有關鍵字 Age 的內容 include 的值設定為 False,如下所示:

```
mynode.setKeyedPropertyValue("include", "Age", False)
```

為設定多重內容而結構化

對於多數節點而言,您可以一次指定節點或串流的多個內容。這稱為**多重集合指令**或**設定區塊**。

在某些情況下,結構化內容相當複合。範例如下:

```
sortnode.setPropertyValue("keys", [{"K", "Descending"}, {"Age", "Ascending"}, {"Na", "Descending"}])
```

結構化內容的另一個優勢在於,在某個節點穩定之前可以在該節點上設定若干個內容。依預設,多重集合將在基於單個內容設定的動作運行之前,在區塊中設定所有內容。例如,如果在定義固定檔案節點時分兩步設定欄位內容,由於節點在兩個設定均生效之前是不一致的,將會導致發生錯誤。以多重集合方式定義內容,可使得在更新資料模型前設定上述兩個內容,從而避免上述問題發生。

縮寫

在節點內容語法中使用標準縮寫。瞭解縮寫有助於構建 Script。

縮寫	意義
abs	絕對值
len	長度
最小值	最小值
max	最大值
correl	相關
covar	共變異
num	數值或數值型
pct	百分比
transp	透明
xval	交叉驗證
var	變異或變數 (在來源節點中)

節點和串流內容範例

在 IBM SPSS Modeler 中可以變化方式使用節點和串流內容。此類屬性最常用作 Script 的一部分,作為**獨立式 Script**的一部分用以實現多個串流或作業的自動化;或用作**串流 Script**的一部分用以實現單個串流內部的過程自動化。還可通過在 SuperNode 內使用節點內容來指定節點參數。就最基礎的層次而言,內容還可用作指令行選項來啟動 IBM SPSS Modeler。將 -p 引數用作指令行呼叫的一部分時,可以使用串流內容來變更串流設定。

內容	意義
s.max_size	參照節點 s 的內容 max_size。

內容	意義
s:samplenode.max_size	請參照名為 s 的節點的內容 max_size, 必須為「樣本」節點。
:samplenode.max_size	參照目前串流中樣本節點的內容 max_size (只能有一個樣本節點)。
s:sample.max_size	請參照名為 s 的節點的內容 max_size, 必須為「樣本」節點。
t.direction.Age	涉及到「類型」節點 t 中年齡欄位的角色。
:.max_size	*** 非法操作 *** 必須指定節點名稱或節點類型。

範例 s:sample.max_size 說明不一定要寫出節點類型的全稱。

範例 t.direction.Age 說明, 當某個節點屬性比帶有個別值的個別屬性更複雜時, 某些屬性名稱可以自行結構化。此類屬性稱為**結構化**或**複合**內容。

節點內容概觀

每種類型的節點均有其合法內容集, 每種內容也有其類型。該類型可以是常用類型數字、旗標或字串, 此時內容設定將強制至正確類型。如果無法強制操作, 則將出現錯誤。另外, 通過內容參照可以指定合法值的範圍, 例如 Discard、PairAndDiscard 和 IncludeAsText, 此時如果採用其他值, 則將出現錯誤。應通過採用值 true 或 false 來讀取或設定旗標內容。(設定值時也可識別如下變異值: Off、OFF、off、No、NO、no、n、N、f、F、false、False、FALSE 或 0, 但在某些情況下讀取內容值時會出錯。所有其他值都將被當作 true。使用 true 和 false 時保持一致將可以避免混淆。) 在本手冊的參照表格中, 內容說明欄對結構化內容進行了說明, 並給出了內容的使用格式。

通用節點內容

IBM SPSS Modeler 中的很多內容通用於所有節點 (包括 SuperNode)。

內容名稱	資料類型	內容說明
use_custom_name	旗標	
name	string	讀取畫布中某個節點名稱的唯讀內容 (自動或自訂)。
custom_name	string	指定節點的自訂名稱。
tooltip	string	
annotation	string	
keywords	string	指定與物件關聯的關鍵字清單的結構化屬性 (例如 ["Keyword1" "Keyword2"])。
cache_enabled	旗標	
node_type	source_supernode process_supernode terminal_supernode 所有為進行 Scripting	按類型參照節點的唯讀內容。例如, 除依名稱參照節點 (例如 real_income) 之外, 還可以指定類型 (例如 userinputnode 或 filternode)。

SuperNode 內容以及所有其他節點的內容均將個別討論。請參閱第 401 頁的『[第 21 章 SuperNode 內容](#)』主題，以取得更多資訊。

第 8 章 串流內容

通過 Script 編寫可以控制多種串流內容。要參照串流內容，必須將執行方法設定為使用 Script:

```
stream = modeler.script.stream()
stream.setPropertyValue("execute_method", "Script")
```

範例

節點內容用於參照目前串流中的節點。如下所示的串流 Script 可作為一個範例:

```
stream = modeler.script.stream()
annotation = stream.getPropertyValue("annotation")

annotation = annotation + "\n\nThis stream is called \"" + stream.getLabel() + "\" and
contains the following nodes:\n"

for node in stream.iterator():
    annotation = annotation + "\n" + node.getTypeName() + " node called \"" + node.getLabel()
    + "\""

stream.setPropertyValue("annotation", annotation)
```

此範例使用節點內容建立了一個包含串流中所有節點的清單，並將該清單寫入串流的註解中。生成的註解具有如下形式:

```
This stream is called "druglearn" and contains the following nodes:

type node called "Define Types"
derive node called "Na_to_K"
variablefile node called "DRUG1n"
neuralnetwork node called "Drug"
c50 node called "Drug"
filter node called "Discard Fields"
```

串流內容的具體說明見於下表格。

表 42: 串流內容		
內容名稱	資料類型	內容說明
execute_method	Normal	
	Script	

表 42: 串流內容 (繼續)

內容名稱	資料類型	內容說明
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	
date_baseline	數字	
date_2digit_baseline	數字	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	
time_rollover	旗標	
import_datetime_as_string	旗標	
decimal_places	數字	
decimal_symbol	Default Period Comma	
angles_in_radians	旗標	
use_max_set_size	旗標	
max_set_size	數字	

表 42: 串流內容 (繼續)

內容名稱	資料類型	內容說明
ruleset_evaluation	Voting FirstHit	
refresh_source_nodes	旗標	用於在串流執行中自動重新整理來源節點。
script	string	
annotation	string	
name	string	註: 此內容為唯讀。如果想要變更串流的名稱, 您應該使用其他名稱加以儲存。
parameters		使用此內容可以從獨立式 Script 中更新串流參數。
nodes		詳細資訊參見下方。
encoding	SystemDefault "UTF-8"	
stream_rewriting	boolean	
stream_rewriting_maximise_sql	boolean	
stream_rewriting_optimise_client_execution	boolean	
stream_rewriting_optimise_syntax_execution	boolean	
enable_parallelism	boolean	
sql_generation	boolean	
database_caching	boolean	
sql_logging	boolean	
sql_generation_logging	boolean	
sql_log_native	boolean	
sql_log_prettyprint	boolean	
record_count_suppress_input	boolean	
record_count_feedback_interval	整數	
use_stream_auto_create_node_settings	boolean	如果值為 true, 那麼將使用特定於串流的設定, 否則將使用使用者喜好設定。

表 42: 串流內容 (繼續)

內容名稱	資料類型	內容說明
create_model_applier_for_new_models	boolean	如果值為 true，那麼在模型建置器建立新模型並且沒有處於作用中狀態的更新鏈結時，將新增一個新的模型套用者。 註: 如果您使用的是 IBM SPSS Modeler Batch 第 15 版，那麼必須在 Script 中明確地新增模型套用者。
create_model_applier_update_links	createEnabled createDisabled doNotCreate	定義自動新增模型套用者節點時建立的鏈結類型。
create_source_node_from_builders	boolean	如果值為 true，那麼在來源建置器建立新的來源輸出並且沒有處於作用中狀態的更新鏈結時，將新增一個新的來源節點。
create_source_node_update_links	createEnabled createDisabled doNotCreate	定義自動新增來源節點時建立的鏈結類型。
has_coordinate_system	boolean	如果設置為 true，那麼將座標系統套用於整個串流。
coordinate_system	string	選取的投影座標系統的名稱。
deployment_area	ModelRefresh Scoring None	選擇您要部署串流的方式。如果此值設定為 None，那麼不會使用其他部署項目。
scoring_terminal_node_id	string	選擇串流中的評分分支。它可以是串流中的任何終端節點。
scoring_node_id	string	選擇評分分支中的塊。
model_build_node_id	string	選擇串流中的建模節點。

第 9 章 來源節點內容

來源節點一般內容

所有來源節點的通用內容如下方所示，後面的主題是具體節點的相關資訊。

範例 1

```
varfilenode = modeler.script.stream().create("variablefile", "Var. File")
varfilenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
varfilenode.setKeyedPropertyValue("check", "Age", "None")
varfilenode.setKeyedPropertyValue("values", "Age", [1, 100])
varfilenode.setKeyedPropertyValue("type", "Age", "Range")
varfilenode.setKeyedPropertyValue("direction", "Age", "Input")
```

範例 2

此 Script 假定指定的資料檔案包含名為 Region 的欄位，該欄位代表多線串。

```
from modeler.api import StorageType
from modeler.api import MeasureType

# Create a Variable File node that reads the data set containing
# the "Region" field
varfilenode = modeler.script.stream().create("variablefile", "My Geo Data")
varfilenode.setPropertyValue("full_filename", "C:/mydata/mygeodata.csv")
varfilenode.setPropertyValue("treat_square_brackets_as_lists", True)

# Override the storage type to be a list...
varfilenode.setKeyedPropertyValue("custom_storage_type", "Region",
StorageType.LIST)
# ...and specify the type if values in the list and the list depth
varfilenode.setKeyedPropertyValue("custom_list_storage_type", "Region",
StorageType.INTEGER)
varfilenode.setKeyedPropertyValue("custom_list_depth", "Region", 2)

# Now change the measurement to indentify the field as a geospatial value...
varfilenode.setKeyedPropertyValue("measure_type", "Region",
MeasureType.GEOSPATIAL)
# ...and finally specify the necessary information about the specific
# type of geospatial object
varfilenode.setKeyedPropertyValue("geo_type", "Region", "MultiLineString")
varfilenode.setKeyedPropertyValue("geo_coordinates", "Region", "2D")
varfilenode.setKeyedPropertyValue("has_coordinate_system", "Region", True)
varfilenode.setKeyedPropertyValue("coordinate_system", "Region",
"ETRS_1989_EPSG_Arctic_zone_5-47")
```

表 43: 來源節點一般內容

內容名稱	資料類型	內容說明
direction	Input Target Both None Partition Split Frequency RecordID	欄位角色的索引內容。 用法格式: NODE.direction.FIELDNAME 註: 現在已淘汰值 In 和 Out。在未來版本中可能會置換對它們的支援。
type	Range Flag Set Typeless Discrete Ordered Set Default	欄位類型。如果將該內容設定為 <i>Default</i> , 那麼將清除所有 values 內容設定, 如果將 value_mode 設定為 <i>Specify</i> , 那麼它將重新設定為 <i>Read</i> 。如果 value_mode 已設定為 <i>Pass</i> 或 <i>Read</i> , 那麼它將不受 type 設定的影響。 用法格式: NODE.type.FIELDNAME
storage	Unknown String Integer Real Time Date Timestamp	欄位儲存類型的唯讀索引內容。 用法格式: NODE.storage.FIELDNAME

表 43: 來源節點一般內容 (繼續)

內容名稱	資料類型	內容說明
check	None Nullify Coerce Discard Warn Abort	欄位類型及範圍檢查的索引內容。 用法格式: NODE.check.FIELDNAME
values	[value value]	對於連續型 (範圍) 欄位而言, 第一個是最小值, 最後一個是最大值。對於列名 (集合) 欄位, 請指定所有值。針對旗標欄位, 第一個值代表 <i>false</i> , 最後一個值代表 <i>true</i> 。設定該內容將自動把 <i>value_mode</i> 內容設定為 <i>Specify</i> 。儲存器是根據清單中的第一個值確定的, 例如, 如果第一個值為 <i>string</i> , 那麼儲存器將設定為 <i>String</i> 。 用法格式: NODE.values.FIELDNAME
value_mode	Read Pass Read+ Current Specify	確定下一次資料傳送設定某個欄位值的方式。 用法格式: NODE.value_mode.FIELDNAME 注意, 不能將此內容直接設定為 <i>Specify</i> ; 要使用特定值, 需設定 <i>values</i> 內容。
default_value_mode	Read Pass	指定用預設方式設定所有欄位值。 用法格式: NODE.default_value_mode 可以通過使用 <i>value_mode</i> 內容, 置換特定欄位的這個設定。
extend_values	旗標	當 <i>value_mode</i> 設定為 <i>Read</i> 時將套用。設為 <i>T</i> 則將新讀取的值新增到任意現有欄位值。設定為 <i>F</i> 則捨棄現有值並添加新讀取值。 用法格式: NODE.extend_values.FIELDNAME

表 43: 來源節點一般內容 (繼續)

內容名稱	資料類型	內容說明
value_labels	string	用於指定值標籤。請注意，必須先指定值。
enable_missing	旗標	當設定為 <i>T</i> 時，則啟動對欄位遺漏值的追蹤。 用法格式： NODE.enable_missing.FIELDNAME
missing_values	[value value...]	指定表示遺漏資料的資料值。 用法格式： NODE.missing_values.FIELDNAME
range_missing	旗標	此內容設定為 <i>T</i> 時，指定是否為欄位定義遺漏值（空白）範圍。 用法格式： NODE.range_missing.FIELDNAME
missing_lower	string	當 range_missing 為 true 時，指定遺漏值範圍的下界。 用法格式： NODE.missing_lower.FIELDNAME
missing_upper	string	當 range_missing 為 true 時，指定遺漏值範圍的上限。 用法格式： NODE.missing_upper.FIELDNAME
null_missing	旗標	當此內容設定為 <i>T</i> 時，空值（在本軟體中顯示為 \$null\$ 的未定義值）被視為遺漏值。 用法格式： NODE.null_missing.FIELDNAME
whitespace_missing	旗標	當該內容設定為 <i>T</i> 時，僅包含空白（空格、Tab 和換行符）的值將被當成遺漏值。 用法格式： NODE.whitespace_missing.FIELDNAME
description	string	用於指定欄位標籤或說明。

表 43: 來源節點一般內容 (繼續)

內容名稱	資料類型	內容說明
default_include	旗標	索引內容，用來指定預設行為是傳送欄位還是過濾欄位： NODE.default_include 範例： set mynode:filternode.default_include = false
include	旗標	用於指出是包含還是過濾單個欄位的鍵控內容： NODE.include.FIELDNAME.
new_name	string	
measure_type	Range / MeasureType.RANGE Discrete / MeasureType.DISCRETE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS Collection / MeasureType.COLLECTION Geospatial / MeasureType.GEOSPATIAL	此索引內容類似於 type，因為可以用來定義與欄位相關聯的測量。不同的是 Python Scripting，還可以為 setter 函數傳遞其中一個 MeasureType 值，而 getter 一律在 MeasureType 值上傳回。

表 43: 來源節點一般內容 (繼續)

內容名稱	資料類型	內容說明
collection_measure	Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS	針對集合欄位 (深度為 0 的清單), 此索引內容會定義與基礎值相關聯的測量類型。
geo_type	Point MultiPoint LineString MultiLineString Polygon MultiPolygon	針對地理空間欄位, 此索引內容會定義此欄位代表的地理空間物件類型。這應與值的清單深度一致。
has_coordinate_system	<i>boolean</i>	對於地理空間欄位, 此內容會定義此欄位是否具有座標系統
coordinate_system	<i>string</i>	針對地理空間欄位, 此索引內容會定義此欄位的座標系統。

表 43: 來源節點一般內容 (繼續)

內容名稱	資料類型	內容說明
custom_storage_type	Unknown / MeasureType.UNKNOWN String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP List / MeasureType.LIST	此索引內容類似於 custom_storage，因為可以用來定義欄位的置換儲存。不同的是 Python Scripting，還可以為 setter 函數傳遞其中一個 StorageType 值，而 getter 一律在 StorageType 值上傳回。
custom_list_storage_type	String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP	針對清單欄位，此索引內容會指定基礎值的儲存類型。
custom_list_depth	整數	針對清單欄位，此索引內容會指定欄位的深度

表 43: 來源節點一般內容 (繼續)

內容名稱	資料類型	內容說明
max_list_length	整數	僅可用於測量層級為地理空間或集合的資料。透過指定清單可以包含的元素數目來設定清單的長度上限。
max_string_length	整數	僅可用於無類型資料，且僅可在產生 SQL 來建立表格時使用。輸入資料中最大字串的值；這會產生表格中大小足以包含字串的直欄。

asimport 內容

分析伺服器 來源可讓您在「Hadoop 分散式檔案系統 (HDFS)」上執行串流。

範例

```
node.setPropertyValue("use_default_as", False)
node.setPropertyValue("connection",
["false", "9.119.141.141", "9080", "analyticserver", "ibm", "admin", "admin", "false", "", "", "", "", "", "])
```

表 44: asimport 內容

asimport 內容	資料類型	內容說明
data_source	string	資料來源的名稱。
use_default_as	boolean	如果設為 True，則會使用伺服器 options.cfg 檔案中配置的預設分析伺服器連線。如果設為 False，則使用此節點的連線。
connection	["string", "string", "string", "string", "string", "string", "string", "string", "string"]	包含分析伺服器連線詳細資料的清單內容。格式為： ["is_secure_connect", "server_url", "server_port", "context_root", "consumer", "user_name", "password", "use-kerberos-auth", "kerberos-krb5-config-file-path", "kerberos-jaas-config-file-path", "kerberos-krb5-service-principal-name", "enable-kerberos-debug"] 其中： is_secure_connect: 指出是否使用安全連線，且連線為 true 或 false。 use-kerberos-auth: 指出是否使用 kerberos 鑑別，其值為 true 或 false。 enable-kerberos-debug: 指出是否使用 kerberos 鑑別的除錯模式，其值為 true 或 false。

cognosimport 節點內容



IBM Cognos 來源節點會從 Cognos Analytics 資料庫匯入資料。

範例

```
node = stream.create("cognosimport", "My node")
node.setPropertyValue("cognos_connection", ["http://mycogsrv1:9300/p2pd/
servlet/dispatch",
True, "", "", ""])
node.setPropertyValue("cognos_package_name", "/Public Folders/GOSALES")
node.setPropertyValue("cognos_items", ["[GreatOutdoors].[BRANCH].
[BRANCH_CODE]", "[GreatOutdoors]
.[BRANCH].[COUNTRY_CODE]"])
```

表 45: cognosimport 節點內容

cognosimport 節點內容	資料類型	內容說明
mode	Data Report	指定是匯入 Cognos 資料（預設）還是報告。

表 45: *cognosimport* 節點內容 (繼續)

cognosimport 節點內容	資料類型	內容說明
cognos_connection	["string",flag,"string", "string","string"]	<p>包含 Cognos 伺服器連線詳細資料的清單內容。格式為： ["Cognos_server_URL", login_mode, "namespace", "username", "password"]</p> <p>其中：</p> <p>Cognos_server_URL 是包含來源之 Cognos 伺服器的 URL。</p> <p>login_mode 指出是否使用匿名登入 (為 true 或 false)；如果設為 true，則下列欄位應該設為 ""。</p> <p>namespace 指定用來登入伺服器的安全鑑別提供者。</p> <p>username 和 password 是用來登入 Cognos 伺服器的使用者名稱和密碼。</p> <p>還會提供下列節點而非 login_mode：</p> <ul style="list-style-type: none"> • anonymousMode。例如： ['Cognos_server_url', 'anonymousMode', "namespace", "username", "password"] • credentialMode。例如： ['Cognos_server_url', 'credentialMode', "namespace", "username", "password"] • storedCredentialMode。例如： ['Cognos_server_url', 'storedCredentialMode', "stored_credential_name"] <p>其中 stored_credential_name 是儲存庫中 Cognos 認證的名稱。</p>
cognos_package_name	string	<p>您要從中匯入資料物件的 Cognos 套件的路徑和名稱，例如：</p> <p>/Public Folders/GOSALES</p> <p>註：只有正斜線才有效。</p>

表 45: cognosimport 節點內容 (繼續)

cognosimport 節點內容	資料類型	內容說明
cognos_items	<code>["field","field", ... ,"field"]</code>	要匯入的一個或多個資料物件的名稱。 <i>field</i> 格式為 <code>[namespace]. [query_subject].[query_item]</code>
cognos_filters	欄位	匯入資料前要套用的一個或多個過濾器的名稱。
cognos_data_parameters	清單	資料的提示參數的值。名稱/值配對以方括弧括住，多個配對以逗點區隔，而整個字串以方括弧括住。 格式： <code>[["param1", "value"], ..., ["paramN", "value"]]</code>
cognos_report_directory	欄位	要從中匯入報告的資料夾或包的 Cognos 路徑，例如： <code>/Public Folders/GOSALES</code> 註：只有正斜線才有效。
cognos_report_name	欄位	要匯入的報告的報告位置中的路徑和名稱。
cognos_report_parameters	清單	報告參數的值。名稱/值配對以方括弧括住，多個配對以逗點區隔，而整個字串以方括弧括住。 格式： <code>[["param1", "value"], ..., ["paramN", "value"]]</code>

databasenode 內容



您可以使用「資料庫」節點來透過 ODBC（開放式資料庫連線功能）從多種其他套件中匯入資料，包括 Microsoft SQL Server、Db2、Oracle 等。

範例

```
import modeler.api
stream = modeler.script.stream()
node = stream.create("database", "My node")
node.setPropertyValue("mode", "Table")
node.setPropertyValue("query", "SELECT * FROM drug1n")
node.setPropertyValue("datasource", "Drug1n_db")
node.setPropertyValue("username", "spss")
node.setPropertyValue("password", "spss")
node.setPropertyValue("tablename", ".Drug1n")
```

表 46: databasemode 內容

databasemode 內容	資料類型	內容說明
mode	Table Query	指定表格以透過使用對話框控制項連接至資料庫表格，或指定查詢以透過使用 SQL 查詢選取的資料庫。
datasource	string	資料庫名稱（另請參閱下方的附註）。
username	string	資料庫連線詳細資料（另請參閱下方的附註）。
password	string	
credential	string	IBM SPSS Collaboration and Deployment Services 中儲存的認證的名稱。此內容可用於替代 username 和 password 內容。此認證的使用者名稱和密碼必須與存取資料庫所需的使用者名稱和密碼相符。
use_credential		設定為 True 或 False。
epassword	string	指定一個編碼密碼來替代在 Script 中將密碼寫在程式中。 請參閱第 47 頁的『產生經過編碼的密碼』主題，以取得更多資訊。在執行過程中，此內容為唯讀。
tablename	string	要存取的表格名稱。
strip_spaces	None Left Right Both	捨棄字串中前端和尾部空格的選項。
use_quotes	AsNeeded Always Never	指定在向資料庫傳送查詢時是否用引號括起表格和欄的名稱（例如在包含空格或標點符號的情況下）。
query	string	指定要提交查詢所對應的 SQL 編碼。

註: 如果 datasource 內容中的資料庫名稱包含空格，那麼您還可以使用下列格式的單個資料來源內容，以代替使用 datasource、username 和 password 的各個內容：

表 47: *datanode* 內容 - 資料來源專用

datanode 內容	資料類型	內容說明
datasource	string	<p>格式:</p> <p>[database_name,username,password[,true false]]</p> <p>最後一個參數與經過加密的密碼配合使用。如果將其設為 true，將會在使用之前對密碼進行解密。</p>

如果您要變更資料來源，也可使用此格式；不過，如果您只想變更使用者名稱或密碼，則可使用 `username` 或 `password` 內容。

datacollectionimportnode 內容



資料收集 資料匯入節點根據市場調查產品所使用的 資料收集 資料模型來匯入意見調查資料。必須安裝 資料收集 資料程式庫才可使用此節點。

範例

```
node = stream.create("datacollectionimport", "My node")
node.setPropertyValue("metadata_name", "mrQvDsc")
node.setPropertyValue("metadata_file", "C:/Program Files/IBM/SPSS/
DataCollection/DDL/Data/
Quanvert/Museum/museum.pkd")
node.setPropertyValue("casedata_name", "mrQvDsc")
node.setPropertyValue("casedata_source_type", "File")
node.setPropertyValue("casedata_file", "C:/Program Files/IBM/SPSS/
DataCollection/DDL/Data/
Quanvert/Museum/museum.pkd")
node.setPropertyValue("import_system_variables", "Common")
node.setPropertyValue("import_multi_response", "MultipleFlags")
```

表 48: *datacollectionimportnode* 內容

datacollectionimportnode 內容	資料類型	內容說明
metadata_name	<i>string</i>	<p>MDSC 的名稱。特殊值 DimensionsMDD 表示應使用標準 資料收集 meta 資料文件。其他可能的值包含：</p> <p>mrADODsc</p> <p>mrI2dDsc</p> <p>mrLogDsc</p> <p>mrQdiDrsDsc</p> <p>mrQvDsc</p> <p>mrSampleReportingMDSC</p> <p>mrSavDsc</p> <p>mrSCDsc</p> <p>mrScriptMDSC</p> <p>特殊值 none 指示不存在 MDSC。</p>
metadata_file	<i>string</i>	儲存 meta 資料的檔案的名稱。

表 48: datacollectionimportnode 內容 (繼續)

datacollectionimportnode 內容	資料類型	內容說明
casedata_name	string	CDSC 的名稱。 可能的值包含： mrADODsc mrI2dDsc mrLogDsc mrPunchDSC mrQdiDrsDsc mrQvDsc mrRdbDsc2 mrSavDsc mrScDSC mrXmlDsc 特殊值 none 指示不存在 CDSC。
casedata_source_type	Unknown File Folder UDL DSN	指出 CDSC 的來源類型。
casedata_file	string	當 casedata_source_type 為 <i>File</i> 時，則指定包含案例資料的檔案。
casedata_folder	string	當 casedata_source_type 為 <i>Folder</i> 時，則指定包含案例資料的資料夾。
casedata_udl_string	string	當 casedata_source_type 為 <i>UDL</i> 時，則為包含案例資料的資料來源指定 OLD-DB 連線字串。
casedata_dsn_string	string	當 casedata_source_type 為 <i>DSN</i> ，則為資料來源指定 ODBC 連線字串。
casedata_project	string	從 資料收集 資料庫中讀取觀察值資料時，可以輸入專案的名稱。對於所有其他觀察值資料類型，此設定應該留為空白。

表 48: datacollectionimportnode 內容 (繼續)

datacollectionimportnode 內容	資料類型	內容說明
version_import_mode	All Latest Specify	定義版本處理方式。
specific_version	string	當 version_import_mode 為 Specify 時，則定義要匯入案例資料的版本。
use_language	string	定義是否應使用指定語言的標籤。
language	string	如果 use_language 的值為 True，則定義匯入時要使用的語言碼。語言碼應為案例資料中的某一可用代碼。
use_context	string	定義是否應匯入特定的環境定義。環境定義可用於改變與回應相關的說明。
context	string	如果 use_context 的值為 true，則定義匯入環境定義。環境定義應是案例資料中的某一可用環境定義。
use_label_type	string	定義是否應匯入指定標籤類型。
label_type	string	如果 use_label_type 的值為 true，則定義要匯入的標籤類型。標籤類型應是案例資料中的某一可用標籤類型。
user_id	string	對於要求明確地登入的資料庫，可通過提供使用者 ID 和密碼來存取資料來源。
password	string	
import_system_variables	Common None All	指定要匯入哪些系統變數。
import_codes_variables	旗標	
import_sourcefile_variables	旗標	
import_multi_response	MultipleFlags Single	

excelimportnode 內容



Excel 匯入節點可從 Microsoft Excel 以 .xlsx 檔案格式匯入資料。不需要 ODBC 資料來源。

範例

```
#To use a named range:
node = stream.create("excelimport", "My node")
node.setPropertyValue("excel_file_type", "Excel2007")
node.setPropertyValue("full_filename", "C:/drug.xlsx")
node.setPropertyValue("use_named_range", True)
node.setPropertyValue("named_range", "DRUG")
node.setPropertyValue("read_field_names", True)

#To use an explicit range:
node = stream.create("excelimport", "My node")
node.setPropertyValue("excel_file_type", "Excel2007")
node.setPropertyValue("full_filename", "C:/drug.xlsx")
node.setPropertyValue("worksheet_mode", "Name")
node.setPropertyValue("worksheet_name", "Drug")
node.setPropertyValue("explicit_range_start", "A1")
node.setPropertyValue("explicit_range_end", "F300")
```

表 49: *excelimportnode* 內容

excelimportnode 內容	資料類型	內容說明
excel_file_type	Excel2007	
full_filename	string	完整檔名，包括路徑。
use_named_range	布林	是否使用指定範圍。如果為 true，則將用 named_range 內容來指定讀取範圍，但忽略其他工作表和資料範圍設定。
named_range	string	
worksheet_mode	Index Name	指定是否通過指標或名稱來定義工作表。
worksheet_index	整數	要讀取工作表的指標，開始時第一個工作表的指標為 0，第二個工作表的指標為 1，以此類推。
worksheet_name	string	要讀取工作表的名稱。
data_range_mode	FirstNonBlank ExplicitRange	指定確定範圍的方式。
blank_rows	StopReading ReturnBlankRows	當 data_range_mode 為 <i>FirstNonBlank</i> 時，指定空白列處理方式。
explicit_range_start	string	當 data_range_mode 為 <i>ExplicitRange</i> 時，指定要讀取範圍的起點。
explicit_range_end	string	
read_field_names	布林	指定是否應將指定範圍的第一列用作欄位（欄）名稱。
scanLineCount	整數	指定直欄和儲存類型的掃描列數。預設值為 200。



使用「延伸匯入」節點，您可以執行 R 或 Python for Spark Script 以匯入資料。

Python for Spark 範例

```
##### Script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_importer", "extension_importer")
node.setPropertyValue("syntax_type", "Python")

python_script = """
import spss.pyspark
from pyspark.sql.types import *

cxt = spss.pyspark.runtime.getContext()

_schema = StructType([StructField('id', LongType(), nullable=False), \
StructField('age', LongType(), nullable=True), \
StructField('Sex', StringType(), nullable=True), \
StructField('BP', StringType(), nullable=True), \
StructField('Cholesterol', StringType(), nullable=True), \
StructField('K', DoubleType(), nullable=True), \
StructField('Na', DoubleType(), nullable=True), \
StructField('Drug', StringType(), nullable=True)])

if cxt.isComputeDataModelOnly():
    cxt.setSparkOutputSchema(_schema)
else:
    df = cxt.getSparkInputData()
    if df is None:
        drugList=[(1,23,'F','HIGH','HIGH',0.792535,0.031258,'drugY'), \
(2,47,'M','LOW','HIGH',0.739309,0.056468,'drugC'), \
(3,47,'M','LOW','HIGH',0.697269,0.068944,'drugC'), \
(4,28,'F','NORMAL','HIGH',0.563682,0.072289,'drugX'), \
(5,61,'F','LOW','HIGH',0.559294,0.030998,'drugY'), \
(6,22,'F','NORMAL','HIGH',0.676901,0.078647,'drugX'), \
(7,49,'F','NORMAL','HIGH',0.789637,0.048518,'drugY'), \
(8,41,'M','LOW','HIGH',0.766635,0.069461,'drugC'), \
(9,60,'M','NORMAL','HIGH',0.777205,0.05123,'drugY'), \
(10,43,'M','LOW','NORMAL',0.526102,0.027164,'drugY')]
        sqlcxt = cxt.getSparkSQLContext()
        rdd = cxt.getSparkContext().parallelize(drugList)
        print 'pyspark read data count = '+str(rdd.count())
        df = sqlcxt.createDataFrame(rdd, _schema)

    cxt.setSparkOutputData(df)
"""

node.setPropertyValue("python_syntax", python_script)
```

R 範例

```
##### Script example for R
node.setPropertyValue("syntax_type", "R")

R_script = """# 'JSON Import' Node v1.0 for IBM SPSS Modeler
# 'RJSONIO' package created by Duncan Temple Lang - http://cran.r-project.org/web/packages/RJSONIO
# 'plyr' package created by Hadley Wickham http://cran.r-project.org/web/packages/plyr
# Node developer: Danil Savine - IBM Extreme Blue 2014
# Description: This node allows you to import into SPSS a table data from a JSON.
# Install function for packages
packages <- function(x){
  x <- as.character(match.call()[[2]])
  if (!require(x,character.only=TRUE)){
    install.packages(pkgs=x,repos="http://cran.r-project.org")
  }
}
```

```

require(x,character.only=TRUE)
}
}
# packages
packages(RJSONIO)
packages(plyr)
### This function is used to generate automatically the dataModel
getMetaData <- function(data) {
  if (dim(data)[1]<=0) {

    print("Warning : modelerData has no line, all fieldStorage fields set to strings")
    getStorage <- function(x){return("string")}

  } else {

    getStorage <- function(x) {
      res <- NULL
      #if x is a factor, typeof will return an integer so we treat the case on the side
      if(is.factor(x)) {
        res <- "string"
      } else {
        res <- switch(typeof(unlist(x)),
                      integer = "integer",
                      double = "real",
                      character = "string",
                      "string")
      }
      return (res)
    }
  }

  col = vector("list", dim(data)[2])
  for (i in 1:dim(data)[2]) {
    col[[i]] <- c(fieldName=names(data[i]),
                  fieldLabel="",
                  fieldStorage=getStorage(data[i]),
                  fieldMeasure="",
                  fieldFormat="",
                  fieldRole="")
  }
  mdm<-do.call(cbind,col)
  mdm<-data.frame(mdm)
  return(mdm)
}
# From JSON to a list
txt <- readLines('C:/test.json')
formattedtxt <- paste(txt, collapse = '')
json.list <- fromJSON(formattedtxt)
# Apply path to json.list
if(strsplit(x='true', split='
',fixed=TRUE)[[1]][1]) {
  path.list <- unlist(strsplit(x='id_array', split=','))
  i = 1
  while(i<length(path.list)+1){
    if(is.null(getElement(json.list, path.list[i]))){
      json.list <- json.list[[1]]
    }else{
      json.list <- getElement(json.list, path.list[i])
      i <- i+1
    }
  }
}
}
# From list to dataframe via unlisted json
i <-1
filled <- data.frame()
while(i < length(json.list)+ 1){
  unlisted.json <- unlist(json.list[[i]])
  to.fill <- data.frame(t(as.data.frame(unlisted.json, row.names = names(unlisted.json))),
stringsAsFactors=FALSE)
  filled <- rbind.fill(filled,to.fill)
  i <- 1 + i
}
# Export to SPSS Modeler Data
modelerData <- filled
print(modelerData)
modelerDataModel <- getMetaData(modelerData)
print(modelerDataModel)
"""
node.setPropertyValue("r_syntax", R_script)

```

表 50: *extensionimportnode* 內容

extensionimportnode 內容	資料類型	內容說明
syntax_type	<i>r</i> <i>Python</i>	指定執行哪個 Script – R 或 Python（預設值是 R）。
r_syntax	<i>string</i>	要執行的 R Scripting 語法。
python_syntax	<i>string</i>	要執行的 Python Scripting 語法。

fixedfilenode 內容



「固定檔案」節點將匯入固定欄位文字檔中的資料，亦即，其欄位未定界但在同一位置啟動且具有固定長度的檔案。機器產生的資料或舊資料通常以固定欄位格式儲存。

範例

```
node = stream.create("fixedfile", "My node")
node.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node.setPropertyValue("record_len", 32)
node.setPropertyValue("skip_header", 1)
node.setPropertyValue("fields", [{"Age", 1, 3}, {"Sex", 5, 7}, {"BP", 9, 10}, {"Cholesterol", 12, 22}, {"Na", 24, 25}, {"K", 27, 27}, {"Drug", 29, 32}])
node.setPropertyValue("decimal_symbol", "Period")
node.setPropertyValue("lines_to_scan", 30)
```

表 51: *fixedfilenode* 內容

fixedfilenode 內容	資料類型	內容說明
record_len	數字	指定每條記錄中的字元數。
line_oriented	旗標	跳過每條記錄尾部的換行符。
decimal_symbol	Default Comma Period	用於資料來源中的小數點的類型。
skip_header	數字	指定第一條記錄開頭要忽略的行數。用於忽略欄標題。
auto_recognize_datetime	旗標	指定在來源資料中是否會自動識別日期或時間。
lines_to_scan	數字	
fields	清單	結構化內容。
full_filename	<i>string</i>	要讀取檔案的全稱（包含目錄）。

表 51: *fixedfilenode* 內容 (繼續)

fixedfilenode 內容	資料類型	內容說明
strip_spaces	None	匯入時捨棄字串中的前導和尾端空格。
	Left	
	Right	
	Both	
invalid_char_mode	Discard	從資料輸入中移除無效字元（空值、0 或現行編碼中任何不存在的字元），或將無效字元取代成指定的一字元符號。
	Replace	
invalid_char_replacement	<i>string</i>	
use_custom_values	旗標	
custom_storage	Unknown	
	String	
	Integer	
	Real	
	Time	
	Date	
	Timestamp	

表 51: *fixedfilenode* 內容 (繼續)

fixedfilenode 內容	資料類型	內容說明
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY"	唯有在指定了自訂儲存體時此內容才適用。

表 51: fixedfilenode 內容 (繼續)

fixedfilenode 內容	資料類型	內容說明
	"DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	唯有在指定了自訂儲存體時此內容才適用。

表 51: *fixedfilenode* 內容 (繼續)

fixedfilenode 內容	資料類型	內容說明
custom_decimal_symbol	欄位	僅當已指定自訂儲存體時才適用。
encoding	StreamDefault SystemDefault "UTF-8"	指定文字編碼方法。

gsdata_import 節點內容



使用「地理空間」來源節點將地圖或空間資料匯入資料採礦階段作業。

表 52: *gsdata_import* 節點內容

gsdata_import 節點內容	資料類型	內容說明
full_filename	string	請輸入要載入的 .shp 檔案的檔案路徑。
map_service_URL	string	請輸入要連接至地圖服務 URL。
map_name	string	僅當使用了 map_service_URL 時，此屬性才有效，並且包含地圖服務的最上層資料夾結構。

jsonimportnode 內容



JSON 來源節點會匯入 JSON 檔中的資料。

表 53: *jsonimportnode* 內容

jsonimportnode 內容	資料類型	內容說明
full_filename	string	完整檔名，包括路徑。
string_format	記錄 值	指定 JSON 字串的格式。預設值是 records。
auto_label		已在 18.2.1.1 版中新增。

sasimportnode 內容



SAS 匯入節點可將 SAS 資料匯入到 IBM SPSS Modeler 中。

範例

```
node = stream.create("sasimport", "My node")
node.setPropertyValue("format", "Windows")
node.setPropertyValue("full_filename", "C:/data/retail.sas7bdat")
node.setPropertyValue("member_name", "Test")
node.setPropertyValue("read_formats", False)
node.setPropertyValue("full_format_filename", "Test")
node.setPropertyValue("import_names", True)
```

表 54: sasimportnode 內容

sasimportnode 內容	資料類型	內容說明
format	Windows UNIX Transport SAS7 SAS8 SAS9	要匯入檔案的格式。
full_filename	string	輸入的完整檔名（包含路徑）。
member_name	string	指定要從特定 SAS 傳輸檔案中匯入的成員。
read_formats	旗標	從指定格式檔案中讀取資料格式（例如變數標籤）。
full_format_filename	string	
import_names	NamesAndLabels LabelsasNames	指定在匯入時對映變數名稱和標籤的方法。

simgennode 內容



「模擬產生」節點提供了一種產生模擬資料的簡單方法 - 使用使用者指定的統計分佈從頭開始產生資料，或者使用對現有歷程資料執行「模擬配適」節點而獲取的分佈自動產生資料。當您想要在模型輸入中存在不確定性時評估預測模型的結果時，這非常有用。

表 55: simgennode 內容

simgennode 內容	資料類型	內容說明
fields	結構化內容	請參閱範例
correlations	結構化內容	請參閱範例
keep_min_max_setting	boolean	
refit_correlations	boolean	
max_cases	整數	最小值為 1000，最大值為 2,147,483,647

表 55: *simgennode* 內容 (繼續)

simgennode 內容	資料類型	內容說明
create_iteration_field	<i>boolean</i>	
iteration_field_name	<i>string</i>	
replicate_results	<i>boolean</i>	
random_seed	整數	
parameter_xml	<i>string</i>	以字串形式傳回參數 XML

fields 範例

這是包含下列語法的結構化位置參數：

```
simgennode.setPropertyValue("fields", [
    [field1, storage, locked, [distribution1], min, max],
    [field2, storage, locked, [distribution2], min, max],
    [field3, storage, locked, [distribution3], min, max]
])
```

distribution 是分佈名稱的宣告，此名稱後跟包含屬性名稱/值成對的清單。每項分佈都以如下方式定義：

```
[distributionname, [[par1], [par2], [par3]]]

simgennode = modeler.script.stream().createAt("simgen", u"Sim Gen", 726, 322)
simgennode.setPropertyValue("fields", [[["Age", "integer", False, ["Uniform", [[["min", "1"],
["max", "2"]]]], "", ""]])
```

例如，要建立用於產生具有二項式分佈的單個欄位的節點，您可以使用下列 Script：

```
simgen_node1 = modeler.script.stream().createAt("simgen", u"Sim Gen", 200, 200)
simgen_node1.setPropertyValue("fields", [[["Education", "Real", False, ["Binomial", [[["n", 32],
["prob", 0.7]]], "", ""]])
```

二項式分佈使用兩個參數：n 和 prob。由於二項式分佈不支援最小值和最大值，因此這兩個參數將作為空字串提供。

註：您不能直接設定 *distribution*；可以將其與 *fields* 內容一起使用。

下列範例顯示所有可能的分配類型。請注意，臨界值在 *NegativeBinomialFailures* 和 *NegativeBinomialTrial* 中均輸入為 *thresh*。

```
stream = modeler.script.stream()
simgennode = stream.createAt("simgen", u"Sim Gen", 200, 200)

beta_dist = ["Field1", "Real", False, ["Beta", [[["shape1", "1"], ["shape2", "2"]]]], "", ""]
binomial_dist = ["Field2", "Real", False, ["Binomial", [[["n", "1"], ["prob", "1"]]]], "", ""]
categorical_dist = ["Field3", "String", False, ["Categorical", [[["A", 0.3], ["B", 0.5], ["C", 0.2]]], "", ""]
dice_dist = ["Field4", "Real", False, ["Dice", [[["1", "0.5"], ["2", "0.5"]]]], "", ""]
exponential_dist = ["Field5", "Real", False, ["Exponential", [[["scale", "1"]]]], "", ""]
fixed_dist = ["Field6", "Real", False, ["Fixed", [[["value", "1"]]]], "", ""]
gamma_dist = ["Field7", "Real", False, ["Gamma", [[["scale", "1"], ["shape", "1"]]]], "", ""]
lognormal_dist = ["Field8", "Real", False, ["Lognormal", [[["a", "1"], ["b", "1"]]]], "", ""]
negbinomialfailures_dist = ["Field9", "Real", False, ["NegativeBinomialFailures", [[["prob", "0.5"], ["thresh", "1"]]]], "", ""]
negbinomialtrial_dist = ["Field10", "Real", False, ["NegativeBinomialTrials", [[["prob", "0.2"], ["thresh", "1"]]]], "", ""]
normal_dist = ["Field11", "Real", False, ["Normal", [[["mean", "1"], ["stddev", "2"]]]], "", ""]
poisson_dist = ["Field12", "Real", False, ["Poisson", [[["mean", "1"]]]], "", ""]
range_dist = ["Field13", "Real", False, ["Range", [[["BEGIN", "1,3"], ["END", "2,4"], ["PROB", "[0.5],[0.5]"]]]], "", ""]
triangular_dist = ["Field14", "Real", False, ["Triangular", [[["min", "0"], ["max", "1"], ["mode", "1"]]]], "", ""]
uniform_dist = ["Field15", "Real", False, ["Uniform", [[["min", "1"], ["max", "2"]]]], "", ""]
weibull_dist = ["Field16", "Real", False, ["Weibull", [[["a", "0"], ["b", "1"], ["c", "1"]]]], "", ""]

simgennode.setPropertyValue("fields", [\
    beta_dist, \
    binomial_dist, \
    categorical_dist, \
    dice_dist, \
    exponential_dist, \
    fixed_dist, \
    gamma_dist, \
    lognormal_dist, \
    negbinomialfailures_dist, \
```

```

negbinomialtrial_dist, \
normal_dist, \
poisson_dist, \
range_dist, \
triangular_dist, \
uniform_dist, \
weibull_dist
])

```

相關係數範例

這是包含下列語法的結構化位置參數：

```

simgennode.setPropertyValue("correlations", [
  [field1, field2, correlation],
  [field1, field3, correlation],
  [field2, field3, correlation]
])

```

相關性可以是介於 +1 與 -1 之間的任何數字。您可以根據需要指定相關性數目。任何未指定的相關性都將設定為 0。如果存在任何未知欄位，那麼應該在相關性矩陣（或表格）上設定相關性值，並以紅色文字顯示該值。如果存在未知欄位，那麼無法執行節點。

statisticsimportnode 內容



IBM SPSS Statistics 檔案節點從 IBM SPSS Statistics 使用的 .sav 檔案格式以及儲存在 IBM SPSS Modeler 中的快取檔案（其也使用相同格式）讀取資料。

有關此節點內容的資訊，請參閱第 379 頁的『statisticsimportnode 內容』。

tm1odataimport 節點內容



IBM Cognos TM1 來源節點從 Cognos TM1 資料庫匯入資料。

tm1odataimport 節點內容	資料類型	內容說明
credential_type	inputCredential 或 storedCredential	用來指出認證類型。
input_credential	清單	當 credential_type 為 inputCredential 時，指定網域、使用者名稱及密碼。
stored_credential_name	string	當 credential_type 為 storedCredential 時；指定 C&DS 伺服器上認證的名稱。
selected_view	["field" "field"]	清單內容，包含所選 TM1 資料方塊的詳細資料以及資料方塊視圖（從中將資料匯入 SPSS）的名稱。例如： TM1_import.setPropertyValue("selected_view", ['plan_BudgetPlan', 'Goal Input'])
is_private_view	旗標	指定 selected_view 是否是專用視圖。預設值是 false。

表 56: *tm1odataimport* 節點內容 (繼續)

tm1odataimport 節點內容	資料類型	內容說明
selected_columns	<code>["field"]</code>	指定選取的直欄：只能指定一個項目。 例如： <code>setProperty("selected_columns", ["Measures"])</code>
selected_rows	<code>["field" "field"]</code>	指定選取的列。 例如： <code>setProperty("selected_rows", ["Dimension_1_1", "Dimension_2_1", "Dimension_3_1", "Periods"])</code>
connection_type	AdminServer TM1Server	指出連線類型。預設值為 AdminServer。
admin_host	<code>string</code>	REST API 的主機名稱的 URL。如果 <code>connection_type</code> 是 AdminServer，則此內容為必要。
server_name	<code>string</code>	從 <code>admin_host</code> 選取的 TM1 伺服器名稱。如果 <code>connection_type</code> 是 AdminServer，則此內容為必要。
server_url	<code>string</code>	TM1 伺服器 REST API 的 URL。如果 <code>connection_type</code> 是 TM1Server，則此內容為必要。

tm1import 節點內容 (已淘汰)



IBM Cognos TM1 來源節點從 Cognos TM1 資料庫匯入資料。

註: 此節點在 Modeler 18.0 中已淘汰。取代節點 Script 名稱是 *tm1odataimport*。

表 57: *tm1import* 節點內容

tm1import 節點內容	資料類型	內容說明
pm_host	<code>string</code>	註: 僅適用於 16.0 版及 17.0 版 主機名稱。例如： <code>TM1_import.setProperty("pm_host", 'http://9.191.86.82:9510/pmhub/pm')</code>
tm1_connection	<code>["field","field", ... ,"field"]</code>	註: 僅適用於 16.0 版及 17.0 版 包含 TM1 伺服器連線詳細資料的清單內容。格式為: ["TM1_Server_Name", "tm1_username", "tm1_password"] 例如： <code>TM1_import.setProperty("tm1_connection", ['Planning Sample', "admin", "apple"])</code>

表 57: tm1import 節點內容 (繼續)

tm1import 節點內容	資料類型	內容說明
selected_view	["field" "field"]	清單內容，包含所選 TM1 資料方塊的詳細資料以及資料方塊視圖（從中將資料匯入 SPSS）的名稱。例如： TM1_import.setPropertyValue("selected_view", ['plan_BudgetPlan', 'Goal Input'])
selected_column	["field"]	指定選取的直欄：只能指定一個項目。 例如： setPropertyValue("selected_columns", ["Measures"])
selected_rows	["field" "field"]	指定選取的列。 例如： setPropertyValue("selected_rows", ["Dimension_1_1", "Dimension_2_1", "Dimension_3_1", "Periods"])

twcimport 節點內容



TWC 來源節點從 The Weather Company (IBM 的一個業務) 匯入天氣資料。您可以使用它來取得某個位置的歷程或預測天氣資料。這可以協助您使用最準確且精確的可用天氣資料來制定受天氣驅動的商業解決方案，以做出更好的決策。

表 58: twcimport 節點內容

twcimport 節點內容	資料類型	內容說明
TWCDataImport.latitude	real	以格式 [-90.0~90.0] 指定緯度值
TWCDataImport.longitude	real	以格式 [-180.0~180.0] 指定經度值。
TWCDataImport.licenseKey	string	指定從 The Weather Company 獲取的授權金鑰。
TWCDataImport.measurementUnit	English Metric Hybrid	指定測量單位。可能的值為 English、Metric 或 Hybrid。Metric 是預設值。
TWCDataImport.dataType	Historical Forecast	指定要輸入的天氣資料類型。可能的值是 Historical 或 Forecast。Historical 是預設值。
TWCDataImport.startDate	整數	如果為 TWCDataImport.dataType 指定 Historical，則以格式 yyyyMMdd 指定啟動日期。

表 58: twcimport 節點內容 (繼續)

twcimport 節點內容	資料類型	內容說明
TWCDataImport.endDate	整數	如果為 TWCDataImport.dataType 指定 Historical, 則以格式 yyyyMMdd 指定結束日期。
TWCDataImport.forecastHour	6 12 24 48	如果為 TWCDataImport.dataType 指定了 Forecast, 則指定 6、12、24 或 48 作為小時值。

userinputnode 內容



「使用者輸入」節點可讓您輕鬆從頭開始建立綜合資料, 或透過變更現有資料來建立。例如, 當您要建立一個測試資料集以便建模時, 此項很有用。

範例

```
node = stream.create("userinput", "My node")
node.setPropertyValue("names", ["test1", "test2"])
node.setKeyedPropertyValue("data", "test1", "2, 4, 8")
node.setKeyedPropertyValue("custom_storage", "test1", "Integer")
node.setPropertyValue("data_mode", "Ordered")
```

表 59: userinputnode 內容

userinputnode 內容	資料類型	內容說明
data		
names		設定或傳回節點所產生的欄位名稱清單的結構化屬性。
custom_storage	Unknown String Integer Real Time Date Timestamp	可用於設定或傳回某個欄位儲存的索引通道。

表 59: *userinputnode* 內容 (繼續)

userinputnode 內容	資料類型	內容說明
data_mode	Combined Ordered	如果指定了 Combined, 則為設定值和最小/最大值的每個組合產生記錄。產生的記錄數等於每個欄位中值的數量的 product。如果指定了 Ordered, 那麼從每條記錄的每一欄中提取一個值來產生資料列。產生的記錄數等於一個與欄位相關的最大數字。將為所有資料值較少的欄位新增空值。
values		註: 此內容已由 <i>userinputnode.data</i> 取代, 不應繼續使用。

variablefilenode 內容



「變數檔案」節點可讀取自由欄位文字檔 (亦即, 其記錄包含固定數目的欄位但字元數目不同的檔案) 中的資料。對於頁首文字為固定長度且具有特定註釋類型的檔案, 此節點也很有用。

範例

```
node = stream.create("variablefile", "My node")
node.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node.setPropertyValue("read_field_names", True)
node.setPropertyValue("delimit_other", True)
node.setPropertyValue("other", ",")
node.setPropertyValue("quotes_1", "Discard")
node.setPropertyValue("decimal_symbol", "Comma")
node.setPropertyValue("invalid_char_mode", "Replace")
node.setPropertyValue("invalid_char_replacement", "|")
node.setKeyedPropertyValue("use_custom_values", "Age", True)
node.setKeyedPropertyValue("direction", "Age", "Input")
node.setKeyedPropertyValue("type", "Age", "Range")
node.setKeyedPropertyValue("values", "Age", [1, 100])
```

表 60: *variablefilenode* 內容

variablefilenode 內容	資料類型	內容說明
skip_header	數字	指定每條記錄開頭要忽略的字元數。
num_fields_auto	旗標	自動確定每條記錄中的欄位個數。記錄必須以換行符終止。
num_fields	數字	手動指定每條記錄中的欄位個數。
delimit_space	旗標	指定檔案中用於劃定欄位範圍的字元。
delimit_tab	旗標	
delimit_new_line	旗標	
delimit_non_printing	旗標	
delimit_comma	旗標	當逗點在串流中同時用作小數點和欄位分隔符號時, 將 <i>delimit_other</i> 設定為 <i>true</i> , 然後使用 <i>other</i> 內容將逗點指定為分隔符號。

表 60: *variablefilenode* 內容 (繼續)

variablefilenode 內容	資料類型	內容說明
<code>delimit_other</code>	旗標	容許使用 <code>other</code> 內容來指定自訂分隔符號。
<code>other</code>	<i>string</i>	在 <code>delimit_other</code> 為 <i>true</i> 時, 指定要使用的分隔符號。
<code>decimal_symbol</code>	Default Comma Period	指定用於資料來源中的小數點。
<code>multi_blank</code>	旗標	將多個相鄰空白定界字元視為單個定界字元處理。
<code>read_field_names</code>	旗標	將資料檔案的第一列作為欄的標籤。
<code>strip_spaces</code>	None Left Right Both	匯入時捨棄字串中的前導和尾端空格。
<code>invalid_char_mode</code>	Discard Replace	從資料輸入中移除無效字元 (空值、0 或現行編碼中任何不存在的字元), 或將無效字元取代成指定的一字元符號。
<code>invalid_char_replacement</code>	<i>string</i>	
<code>break_case_by_newline</code>	旗標	指定行分隔符號為換行符。
<code>lines_to_scan</code>	數字	指定具體資料類型的掃描行數。
<code>auto_recognize_datetime</code>	旗標	指定在來源資料中是否會自動識別日期或時間。
<code>quotes_1</code>	Discard PairAndDiscard IncludeAsText	指定匯入後單個的處理方式。
<code>quotes_2</code>	Discard PairAndDiscard IncludeAsText	指定匯入後雙的處理方式。
<code>full_filename</code>	<i>string</i>	要讀取的檔案全稱 (包含目錄)。
<code>use_custom_values</code>	旗標	

表 60: variablefilenode 內容 (繼續)

variablefilenode 內容	資料類型	內容說明
custom_storage	Unknown String Integer Real Time Date Timestamp	

表 60: variablefilenode 內容 (繼續)

variablefilenode 內容	資料類型	內容說明
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY"	僅當已指定自訂儲存體時才適用。

表 60: variablefilenode 內容 (繼續)

variablefilenode 內容	資料類型	內容說明
	"DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	僅當已指定自訂儲存體時才適用。

表 60: variablefilenode 內容 (繼續)

variablefilenode 內容	資料類型	內容說明
custom_decimal_symbol	欄位	僅當已指定自訂儲存體時才適用。
encoding	StreamDefault SystemDefault "UTF-8"	指定文字編碼方法。

xmlexportnode 內容



XML 來源節點將 XML 格式的資料匯入到串流中。可以匯入單個檔案或某個目錄中的所有檔案。還選用擇指定綱目檔，以從中讀取 XML 結構。

範例

```
node = stream.create("xmlexport", "My node")
node.setPropertyValue("full_filename", "c:/import/ebooks.xml")
node.setPropertyValue("records", "/author/name")
```

表 61: xmlexportnode 內容

xmlexportnode 內容	資料類型	內容說明
read	single directory	讀取單個資料檔案 (預設)，或目錄中的所有 XML 檔案。
recurse	旗標	指定是否另外讀取指定目錄的所有子目錄中的 XML 檔案。
full_filename	string	(必要項目) 要匯入之 XML 檔的完整路徑和檔名 (如果 read = single)。
directory_name	string	(必要) 要從中匯入 XML 檔案的目錄的完整路徑和名稱 (如果 read = directory)。
full_schema_filename	string	要從中讀取 XML 結構的 XSD 或 DTD 檔案的完整路徑和檔案名稱。如果您省略了此參數，將從 XML 來源檔案中讀取結構。
records	string	XPath 表示式 (例如, /author/name)，用以定義記錄界限。每次在來源檔案中遇到此元素，都會建立一個新記錄。
mode	read specify	讀取所有資料 (預設)，或指定要讀取的項目。
fields		要匯入的項目 (元素和屬性) 清單。清單中的每項目為 XPath 表示式。

第 10 章 記錄作業節點內容

appendnode 內容



「附加」節點用於連結各組記錄。也可以用於合併結構類似但資料不同的資料集。

範例

```
node = stream.create("append", "My node")
node.setPropertyValue("match_by", "Name")
node.setPropertyValue("match_case", True)
node.setPropertyValue("include_fields_from", "All")
node.setPropertyValue("create_tag_field", True)
node.setPropertyValue("tag_field_name", "Append_Flag")
```

表 62: appendnode 內容

appendnode 內容	資料類型	內容說明
match_by	Position Name	可以根據欄位在主資料來源中的位置或輸入資料集中欄位的名稱來附加資料集。
match_case	旗標	比對欄位名稱時啟用區分大小寫。
include_fields_from	Main All	
create_tag_field	旗標	
tag_field_name	string	

aggregatenode 內容



「合計」節點用聚合和合計的輸出記錄替代一系列輸入記錄。

範例

```
node = stream.create("aggregate", "My node")
# dbnode is a configured database import node
stream.link(dbnode, node)
node.setPropertyValue("contiguous", True)
node.setPropertyValue("keys", ["Drug"])
node.setKeyedPropertyValue("aggregates", "Age", ["Sum", "Mean"])
node.setPropertyValue("inc_record_count", True)
node.setPropertyValue("count_field", "index")
node.setPropertyValue("extension", "Aggregated_")
node.setPropertyValue("add_as", "Prefix")
```

表 63: *aggreatenode* 內容

aggreatenode 內容	資料類型	內容說明
keys	清單	列出可用作聚集鍵的欄位。例如，如果 Sex 和 Region 是索引鍵欄位，那麼 M 和 F 與 N 和 S 區域的每個唯一組合（四個唯一組合）都將具有一條聚合記錄。
contiguous	旗標	如果您知道鍵值相同的所有記錄都在輸入中群組在一起（例如，如果對索引鍵欄位上的輸入進行排序）。這樣做可以改善效能。
aggregates		一種結構化內容，它列出其值將進行聚集的數字欄位以及所選的聚集模式。
aggregate_exprs		鍵控內容，輸入衍生欄位名稱以及用於計算此名稱的聚合表示式。例如： <pre>aggreatenode.setKeyedPropertyValue("aggregate_exprs", "Na_MAX", "MAX('Na')")</pre>
extension	<i>string</i>	為複製聚合的欄位指定字首或字尾（樣本如下方）。
add_as	Suffix Prefix	
inc_record_count	旗標	建立額外欄位，指定聚集多少輸入記錄來形成每一個聚集記錄。
count_field	<i>string</i>	指定記錄計數欄位的名稱。
allow_approximation	布林	在分析伺服器中執行聚集時容許使用順序統計資料的近似值。
bin_count	整數	指定要在近似統計中使用的「分組」個數

balancenode 內容



「平衡」節點糾正資料集中的不平衡度，因而它遵循指定的條件。平衡指令透過指定的因素來調整滿足條件的記錄比例。

範例

```
node = stream.create("balance", "My node")
node.setPropertyValue("training_data_only", True)
node.setPropertyValue("directives", [[1.3, "Age > 60"], [1.5, "Na > 0.5"]])
```

表 64: *balancenode* 內容

balancenode 內容	資料類型	內容說明
directives		根據指定數字平衡欄位值比例的結構化內容（參閱下方的範例）。
training_data_only	旗標	指定僅已平衡訓練資料。如果串流中不出現分割區欄位，則忽略該選項。

此節點內容使用以下格式：

[[*number, string*] \ [*number, string*] \ ... [*number, string*]].

註：如果在表示式中內嵌字串（使用雙引號），則必須在它們之前加上跳出字元 " \ "。" \ " 字元也是行接續字元，您可以用來對齊引數以清楚說明。

cplexoptnode 內容



CPLEX Optimization 節點可讓您透過最佳化程式設計語言 (OPL) 模型檔來使用複雜數學 (CPLEX) 型最佳化。此功能之前在 IBM Analytical Decision Management 產品中有提供，它不再受支援。但您也可以使用 SPSS Modeler 中的 CPLEX 節點，而不需要 IBM Analytical Decision Management。

表 65: *cplexoptnode* 內容

cplexoptnode 內容	資料類型	內容說明
opl_model_text	<i>string</i>	將由 CPLEX Optimization 節點執行並產生最佳化結果的 OPL（最佳化程式設計語言）Script 程式。
opl_tuple_set_name	<i>string</i>	這是 OPL 模型中對應於傳入資料的值組集名稱。這不是必要項目，且一般不透過 Script 來設定。它僅應用來編輯某個選定資料來源的欄位對映。
data_input_map	結構化內容的清單	資料來源的輸入欄位對映。這不是必要項目，且一般不透過 Script 來設定。它僅應用來編輯某個選定資料來源的欄位對映。

表 65: cplexoptnode 內容 (繼續)

cplexoptnode 內容	資料類型	內容說明
md_data_input_map	結構化內容的清單	<p>OPL 中定義的每個元組與每個對應欄位資料來源 (送入的資料) 之間的欄位對映。使用者可以每個資料來源為基礎逐個編輯欄位對映。使用此 Script, 您可以直接設定該內容以一次性設定所有對映。此設定不會顯示於使用者介面中。</p> <p>清單中的每一個實體都是結構化資料:</p> <p>資料來源標籤。 資料來源的標籤, 可以在資料來源下拉清單中找到。例如, 對於 0_Products_Type, 標籤為 0。</p> <p>資料來源索引。 資料來源的實體索引 (索引)。這由連線順序決定。</p> <p>來源節點。 資料來源的來源節點 (註釋)。它可以在資料來源下拉清單中找到。例如, 對於 0_Products_Type, 來源節點為 Products。</p> <p>已連接節點。 連接現行 CPLEX Optimization 節點的先前節點。它可以在資料來源下拉清單中找到。例如, 對於 0_Products_Type, 連接節點為 Type。</p> <p>值組集名稱。 資料來源的值組集名稱。它必須與 OPL 中定義的名稱相符。</p> <p>值組欄位名稱。 資料來源的值組集欄位名稱。它必須與 OPL 值組集定義中定義的名稱相符。</p> <p>儲存體類型。 欄位儲存類型。可能的值為 int、float 或 string。</p>
		<p>資料欄位名稱。 資料來源的欄位名稱。</p> <p>範例:</p> <pre data-bbox="966 1365 1461 1627">[[0,0,'Product','Type','Products','prod_id_tup','int','prod_id'], [0,0,'Product','Type','Products','prod_name_tup','string','prod_name'], [1,1,'Components','Type','Components','comp_id_tup','int','comp_id'], [1,1,'Components','Type','Components','comp_name_tup','string','comp_name']]</pre>
opl_data_text	string	用於 OPL 的部分變數或資料的定義。
output_value_mode	string	可能的值為 raw 或 dvar。如果指定了 dvar, 那麼使用者必須在「輸出」標籤上指定 OPL 中用於輸出的物件函數變數名稱。如果指定了 raw, 那麼將直接輸出目標函數, 而與名稱無關。

表 65: cplexoptnode 內容 (繼續)

cplexoptnode 內容	資料類型	內容說明
decision_variable_name	string	OPL 中定義的目標函數變數名稱。僅當 output_value_mode 內容設定為 dvar 時，才啟用此內容。
objective_function_value_fieldname	string	要在輸出中使用的目標函數值的欄位名稱。預設值為 _OBJECTIVE。
output_tuple_set_names	string	<p>送入的資料中預先定義元組的名稱。它充當決策變數的索引，並預期與變數輸出一一起輸出。「輸出元組」必須與 OPL 中的決策變數定義一致。如果有多個索引，則元組名稱必須以逗點 (,) 結合。</p> <p>單一值組的範例為 Products, 對應的 OPL 定義為 dvar float+ Production[Products];</p> <p>多個值組的範例為 Products,Components, 對應的 OPL 定義為 dvar float+ Production[Products][Components];</p>
decision_output_map	結構化內容的清單	<p>OPL 中定義的將輸出的變數與輸出欄位之間的欄位對映。清單中的每一個實體都是結構化資料：</p> <p>變數名稱。 OPL 中要輸出的變數名稱。</p> <p>儲存體類型。 可能的值為 int、float 或 string。</p> <p>輸出欄位名稱。 結果（輸出或匯出）中的預期欄位名稱。</p> <p>範例：</p> <pre data-bbox="979 1276 1430 1352">[['Production', 'int', 'res'], ['Remark', 'string', 'res_1']]['Cost', 'float', 'res_2']]</pre>

derive_stbnode 內容



「空間時間限制」節點根據緯度、經度和時間戳記欄位衍生了空間時間限制。您還可以將頻繁的空間時間限制識別為逗留。

範例

```
node = modeler.script.stream().createAt("derive_stb", "My node", 96, 96)

# Individual Records mode
node.setPropertyValue("mode", "IndividualRecords")
node.setPropertyValue("latitude_field", "Latitude")
node.setPropertyValue("longitude_field", "Longitude")
node.setPropertyValue("timestamp_field", "OccurredAt")
node.setPropertyValue("densities", ["STB_GH7_1HOUR", "STB_GH7_30MINS"])
node.setPropertyValue("add_extension_as", "Prefix")
node.setPropertyValue("name_extension", "stb_")
```

```
# Hangouts mode
node.setPropertyValue("mode", "Hangouts")
node.setPropertyValue("hangout_density", "STB_GH7_30MINS")
node.setPropertyValue("id_field", "Event")
node.setPropertyValue("qualifying_duration", "30MINUTES")
node.setPropertyValue("min_events", 4)
node.setPropertyValue("qualifying_pct", 65)
```

表 66: Space-Time-Boxes 節點內容

derive_stbnode 內容	資料類型	內容說明
mode	IndividualRecords Hangouts	
latitude_field	欄位	
longitude_field	欄位	
timestamp_field	欄位	
hangout_density	密度	單一密度。請參閱 densities 以瞭解有效的密度值。
densities	[density,density,..., density]	每個密度都是一個字串，例如 STB_GH8_1DAY。 註: 對於哪些密度有效，存在限制。對於地理雜湊，可以使用 GH1 到 GH15 中的值。針對時間部分，可以使用下列值： EVER 1YEAR 1MONTH 1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2HOURS 1HOUR 30MINS 15MINS 10MINS 5MINS 2MINS 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SEC
id_field	欄位	

表 66: Space-Time-Boxes 節點內容 (繼續)

derive_stbnode 內容	資料類型	內容說明
qualifying_duration	1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2Hours 1HOUR 30MIN 15MIN 10MIN 5MIN 2MIN 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SECS	必須是字串。
min_events	整數	下限有效整數值為 2。
qualifying_pct	整數	必須介於 1 與 100 之間。
add_extension_as	Prefix Suffix	
name_extension	string	

distinctnode 內容



「相異」節點將移除重複記錄，方法是將第一個相異記錄傳遞到資料串流，或者捨棄第一個記錄而將任何重複記錄傳遞到資料串流。

範例

```
node = stream.create("distinct", "My node")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("fields", ["Age" "Sex"])
node.setPropertyValue("keys_pre_sorted", True)
```

表 67: distinctnode 內容

distinctnode 內容	資料類型	內容說明
mode	Include Discard	既可以將第一條相異的記錄包含在資料串流中，也可以捨棄第一條相異的記錄並將任何複製記錄傳送到資料串流。
grouping_fields	清單	列出用於確定記錄是否相同的欄位。 註: 從 IBM SPSS Modeler 16 啟動，會淘汰此內容。
composite_value	結構化屬性	請參閱下列範例。
composite_values	結構化屬性	請參閱下列範例。

表 67: *distinctnode* 內容 (繼續)

distinctnode 內容	資料類型	內容說明
inc_record_count	旗標	建立額外欄位，指定聚集多少輸入記錄來形成每一個聚集記錄。
count_field	<i>string</i>	指定記錄計數欄位的名稱。
sort_keys	結構化插槽。	註: 從 IBM SPSS Modeler 16 啟動，會淘汰此內容。
default_ascending	旗標	
low_distinct_key_count	旗標	指定您只具有少量記錄和/或少量索引鍵欄位唯一值。
keys_pre_sorted	旗標	指定具有相同索引值的所有記錄在輸入中群組在一起。
disable_sql_generation	旗標	

composite_value 內容的範例

composite_value 內容的一般格式如下:

```
node.setKeyedPropertyValue("composite_value", FIELD, FILLOPTION)
```

FILLOPTION 的格式為 [FillType, Option1, Option2, ...]。

範例:

```
node.setKeyedPropertyValue("composite_value", "Age", ["First"])
node.setKeyedPropertyValue("composite_value", "Age", ["last"])
node.setKeyedPropertyValue("composite_value", "Age", ["Total"])
node.setKeyedPropertyValue("composite_value", "Age", ["Average"])
node.setKeyedPropertyValue("composite_value", "Age", ["Min"])
node.setKeyedPropertyValue("composite_value", "Age", ["Max"])
node.setKeyedPropertyValue("composite_value", "Date", ["Earliest"])
node.setKeyedPropertyValue("composite_value", "Date", ["Latest"])
node.setKeyedPropertyValue("composite_value", "Code", ["FirstAlpha"])
node.setKeyedPropertyValue("composite_value", "Code", ["LastAlpha"])
```

自訂選項需要多個引數，這些引數以清單形式新增，例如:

```
node.setKeyedPropertyValue("composite_value", "Name", ["MostFrequent", "FirstRecord"])
node.setKeyedPropertyValue("composite_value", "Date", ["LeastFrequent", "LastRecord"])
node.setKeyedPropertyValue("composite_value", "Pending", ["IncludesValue", "T", "F"])
node.setKeyedPropertyValue("composite_value", "Marital", ["FirstMatch", "Married", "Divorced",
"Separated"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "Space"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "Comma"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "UnderScore"])
```

composite_values 內容的範例

composite_values 內容的一般格式如下:

```
node.setPropertyValue("composite_values", [
    [FIELD1, [FILLOPTION1]],
    [FIELD2, [FILLOPTION2]],
    .
    ])
```

範例:

```
node.setPropertyValue("composite_values", [
  ["Age", ["First"]],
  ["Name", ["MostFrequent", "First"]],
  ["Pending", ["IncludesValue", "T"]],
  ["Marital", ["FirstMatch", "Married", "Divorced", "Separated"]],
  ["Code", ["Concatenate", "Comma"]]
])
```

extensionprocessnode 內容



通過「延伸轉換」節點，您可以從串流中提取資料，並套用 R Scripting 或 Python for Spark Scripting 對該資料套用轉換。

Python for Spark 範例

```
#### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_process", "extension_process")
node.setPropertyValue("syntax_type", "Python")

process_script = """
import spss.pyspark.runtime
from pyspark.sql.types import *

cxt = spss.pyspark.runtime.getContext()

if cxt.isComputeDataModelOnly():
    _schema = StructType([StructField("Age", LongType(), nullable=True), \
                             StructField("Sex", StringType(), nullable=True), \
                             StructField("BP", StringType(), nullable=True), \
                             StructField("Na", DoubleType(), nullable=True), \
                             StructField("K", DoubleType(), nullable=True), \
                             StructField("Drug", StringType(), nullable=True)])
    cxt.setSparkOutputSchema(_schema)
else:
    df = cxt.getSparkInputData()
    print df.dtypes[:]
    _newDF = df.select("Age", "Sex", "BP", "Na", "K", "Drug")
    print _newDF.dtypes[:]
    cxt.setSparkOutputData(_newDF)
"""

node.setPropertyValue("python_syntax", process_script)
```

R 範例

```
#### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_syntax", "" "day<-as.Date(modelerData$dob, format=\"%Y-%m-%d\")
next_day<-day + 1
modelerData<-cbind(modelerData,next_day)
var1<-c(fieldName="Next day",fieldLabel="",fieldStorage="date",fieldMeasure="",fieldFormat="",
fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)""")
```

表 68: *extensionprocessnode* 內容

extensionprocessnode 內容	資料類型	內容說明
syntax_type	<i>r</i> <i>Python</i>	指定執行哪個 Script – R 或 Python（預設值是 R）。
r_syntax	<i>string</i>	要執行的 R Scripting 語法。
python_syntax	<i>string</i>	要執行的 Python Scripting 語法。
use_batch_size	旗標	啟用批次處理。
batch_size	整數	指定要包含在每個批次中的資料記錄數量。
convert_flags	StringsAndDoubles LogicalValues	用於轉換旗標欄位的選項。
convert_missing	旗標	將遺漏值轉換為 R NA 值的選項。
convert_datetime	旗標	用於將日期或日期時間格式的變數轉換為 R 日期/時間格式的選項。
convert_datetime_class	POSIXct POSIXlt	用於指定將日期或日期時間格式的變數轉換為何種格式的選項。

mergenode 內容



「合併」節點獲取多個輸入記錄並建立包含某些或全部輸入欄位的單個輸出記錄。這對於合併來源不同的資料 非常有用，例如內部客戶資料和已購買人群統計資料。

範例

```
node = stream.create("merge", "My node")
# assume customerdata and salesdata are configured database import nodes
stream.link(customerdata, node)
stream.link(salesdata, node)
node.setPropertyValue("method", "Keys")
node.setPropertyValue("key_fields", ["id"])
node.setPropertyValue("common_keys", True)
node.setPropertyValue("join", "PartialOuter")
node.setKeyedPropertyValue("outer_join_tag", "2", True)
node.setKeyedPropertyValue("outer_join_tag", "4", True)
node.setPropertyValue("single_large_input", True)
node.setPropertyValue("single_large_input_tag", "2")
node.setPropertyValue("use_existing_sort_keys", True)
node.setPropertyValue("existing_sort_keys", [["id", "Ascending"]])
```

表 69: *mergenode* 內容

mergenode 內容	資料類型	內容說明
method	Order Keys Condition Rankedcondition	指定記錄是否按它們在資料檔案中的列出順序進行合併，是否使用一個或多個索引鍵欄位來合併在索引鍵欄位中包含相同值的記錄，是否在滿足指定條件時合併記錄，或者是否會合併主要資料集以及所有次要資料集中的每個列配對；使用排名表示式可以按從低到高順序對任意多個相符項進行排序。
condition	<i>string</i>	如果 method 設定為 Condition，指定包含或捨棄記錄的條件。
key_fields	清單	
common_keys	旗標	
join	Inner FullOuter PartialOuter Anti	
outer_join_tag.n	旗標	在此內容中， <i>n</i> 是「選取資料集」對話框中顯示的標籤名。注意，可以指定多個標籤名，因為任何數量的資料集都可能提供不完整記錄。
single_large_input	旗標	指定是否進行最佳化，以使一個輸入與其他輸入相比具有一個相對較大的輸入值。
single_large_input_tag	<i>string</i>	按「選取大的資料集」對話框中的顯示指定標籤名。請注意，該內容的使用與 <i>outer_join_tag</i> 內容的使用略有不同（旗標與字串），因為前者只能指定一個輸入資料集。
use_existing_sort_keys	旗標	指定輸入值是否已根據一個或多個索引鍵欄位進行排序。
existing_sort_keys	[[<i>'string'</i> , 'Ascending']\ [<i>'string'</i> ", 'Descending']]	指定已排序的欄位及其排序方向。
primary_dataset	<i>string</i>	如果 method 為 Rankedcondition，請選取用於合併的主資料集。這可視為外部結合合併的左側。
rename_duplicate_fields	布林	如果 method 為 Rankedcondition，此屬性設定為 Y，並且所生成的合併資料集包含來自不同資料來源的多個同名欄位，那麼來自這些資料來源的相應標籤將新增到欄欄位標題開頭。
merge_condition	<i>string</i>	
ranking_expression	<i>string</i>	

表 69: *mergenode* 內容 (繼續)

mergenode 內容	資料類型	內容說明
Num_matches	整數	要根據 <i>merge_condition</i> 和 <i>ranking_expression</i> 傳回的相符項數。最小為 1，最大為 100。

rfmaggregatenode 內容



「回頭率、頻率、消費金額 (RFM) 聚集」節點可讓您取得客戶的歷程交易式資料、除去任何未使用的資料，並將其所有剩餘交易資料結合成單一列，其中列出客戶前次與您交易的時間、他們所進行的交易數目，以及這些交易的總消費金額值。

範例

```
node = stream.create("rfmaggregate", "My node")
node.setPropertyValue("relative_to", "Fixed")
node.setPropertyValue("reference_date", "2007-10-12")
node.setPropertyValue("id_field", "CardID")
node.setPropertyValue("date_field", "Date")
node.setPropertyValue("value_field", "Amount")
node.setPropertyValue("only_recent_transactions", True)
node.setPropertyValue("transaction_date_after", "2000-10-01")
```

表 70: *rfmaggregatenode* 內容

rfmaggregatenode 內容	資料類型	內容說明
relative_to	Fixed Today	指定計算交易回頭率的起始日期。
reference_date	日期	僅在 <i>relative_to</i> 中選擇 <i>Fixed</i> 時才可用。
contiguous	旗標	如果您的資料進行了預先排序，以便所有 ID 相同的記錄一起出現在資料串流中，那麼選取此選項可以加快處理速度。
id_field	欄位	指定該欄位以用來識別客戶及其交易。
date_field	欄位	指定將要用來計算回頭率的日期欄位。
value_field	欄位	指定該欄位以用來計算消費金額值。
extension	string	為複製聚合的欄位指定字首或字尾。
add_as	Suffix Prefix	指定是否應作為字首或字尾來新增 <i>extension</i> 。
discard_low_value_records	旗標	啟用 <i>discard_records_below</i> 設定。
discard_records_below	數字	可在計算 RFM 總計時，指定一個最小值，凡低於該值的交易詳細資料都不再被使用。值單位與選取的 <i>value</i> 欄位相關。
only_recent_transactions	旗標	啟用 <i>specify_transaction_date</i> 或 <i>transaction_within_last</i> 設定。

表 70: rfmaggregatenode 內容 (繼續)

rfmaggregatenode 內容	資料類型	內容說明
specify_transaction_date	旗標	
transaction_date_after	日期	只有已選取 specify_transaction_date 時才可用。指定交易日期以在分析時包含其之後的記錄。
transaction_within_last	數字	只有已選取 transaction_within_last 時才可用。指定從「計算新近值」傳回的數字及期間類型 (日、週、月或年)，相對於其後記錄將包括在分析中的日期。
transaction_scale	Days Weeks Months Years	只有已選取 transaction_within_last 時才可用。指定從「計算新近值」傳回的數字及期間類型 (日、週、月或年)，相對於其後記錄將包括在分析中的日期。
save_r2	旗標	顯示每個客戶次新交易的日期。
save_r3	旗標	只有已選取 save_r2 時才可用。顯示每個客戶最新交易中排名第三的交易的日期。

Rprocessnode 內容



可以使用「R 轉換」節點從 IBM(r) SPSS(r) Modeler 串流中獲取資料，並使用您自己的自訂 R Script 來修改該資料。修改資料之後，資料會傳回至串流。

範例

```
node = stream.create("rprocess", "My node")
node.setPropertyValue("custom_name", "my_node")
node.setPropertyValue("syntax", """day<-as.Date(modelerData$dob, format="%Y-%m-%d")
next_day<-day + 1
modelerData<-cbind(modelerData,next_day)
var1<-c(fieldName="Next
day",fieldLabel="",fieldStorage="date",fieldMeasure="",fieldFormat="",
fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)""")
node.setPropertyValue("convert_datetime", "POSIXct")
```

表 71: Rprocessnode 內容

Rprocessnode 內容	資料類型	內容說明
syntax	string	
convert_flags	StringsAndDoubles LogicalValues	
convert_datetime	旗標	

表 71: Rprocessnode 內容 (繼續)

Rprocessnode 內容	資料類型	內容說明
convert_datetime_class	POSIXct POSIXlt	
convert_missing	旗標	
use_batch_size	旗標	啟用批次處理
batch_size	整數	請指定要在每一個批次中包含的資料記錄數

sampleneode 內容



「樣本」節點選取記錄的子集合。支援各種樣本類型，其中包括分層樣本、叢集樣本以及非隨機（結構化）樣本。取樣對於提高效能和選取相關記錄群組或交易群組用於分析會很有用。

範例

```

/* Create two Sample nodes to extract
   different samples from the same data */

node = stream.create("sample", "My node")
node.setPropertyValue("method", "Simple")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("sample_type", "First")
node.setPropertyValue("first_n", 500)

node = stream.create("sample", "My node")
node.setPropertyValue("method", "Complex")
node.setPropertyValue("stratify_by", ["Sex", "Cholesterol"])
node.setPropertyValue("sample_units", "Proportions")
node.setPropertyValue("sample_size_proportions", "Custom")
node.setPropertyValue("sizes_proportions", [["M", "High", "Default"], ["M",
"Normal", "Default"],
["F", "High", 0.3], ["F", "Normal", 0.3]])
    
```

表 72: sampleneode 內容

sampleneode 內容	資料類型	內容說明
method	Simple Complex	
mode	Include Discard	包含或捨棄符合指定條件的記錄。
sample_type	First OneInN RandomPct	指定取樣方法。
first_n	整數	將併入或捨棄直到指定截止點的記錄。
one_in_n	數字	每隔 $n-1$ 條記錄包含或捨棄一條記錄。

表 72: *samplenode* 內容 (繼續)

samplenode 內容	資料類型	內容說明
rand_pct	數字	指定要包含或捨棄記錄的百分比。
use_max_size	旗標	啟用 maximum_size 設定。
maximum_size	整數	指定要併入在資料串流中或捨棄的最大樣本量。這是備援選項，因此指定 First 和 Include 時會被停用。
set_random_seed	旗標	啟用隨機種子設定。
random_seed	整數	指定用作隨機種子的值。
complex_sample_type	Random Systematic	
sample_units	Proportions Counts	
sample_size_proportions	Fixed Custom Variable	
sample_size_counts	Fixed Custom Variable	
fixed_proportions	數字	
fixed_counts	整數	
variable_proportions	欄位	
variable_counts	欄位	
use_min_stratum_size	旗標	
minimum_stratum_size	整數	僅當使用 Sample units=Proportions 取得複式樣本時，此選項才適用。
use_max_stratum_size	旗標	
maximum_stratum_size	整數	僅當使用 Sample units=Proportions 取得複式樣本時，此選項才適用。
clusters	欄位	
stratify_by	[field1 ... fieldn]	
specify_input_weight	旗標	
input_weight	欄位	
new_output_weight	string	

表 72: *samplenode* 內容 (繼續)

samplenode 內容	資料類型	內容說明
sizes_proportions	[[string string value] [string string value]...]	如果 sample_units=proportions 且 sample_size_proportions=Custom, 指定層欄位值每個可能組合的值。
default_proportion	數字	
sizes_counts	[[string string value] [string string value]...]	指定層欄位值每個可能的組合值。使用與 sizes_proportions 的使用相似, 但指定的是整數, 而非比例。
default_count	數字	

selectnode 內容



「選取」節點可基於特定條件從資料串流中選取或捨棄記錄子集合。例如, 可以選取有關特定銷售區域的記錄。

範例

```
node = stream.create("select", "My node")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("condition", "Age < 18")
```

表 73: *selectnode* 內容

selectnode 內容	資料類型	內容說明
mode	Include Discard	指定是包含還是捨棄選定記錄。
condition	string	包含或捨棄記錄的條件。

sortnode 內容



「排序」節點可根據一個或多個欄位的值將記錄按遞增或遞減排序。

範例

```
node = stream.create("sort", "My node")
node.setPropertyValue("keys", [["Age", "Ascending"], ["Sex", "Descending"]])
node.setPropertyValue("default_ascending", False)
node.setPropertyValue("use_existing_keys", True)
node.setPropertyValue("existing_keys", [["Age", "Ascending"]])
```

表 74: *sortnode* 內容

sortnode 內容	資料類型	內容說明
keys	清單	指定要作為排序依據的欄位。如果未指定方向, 則會使用預設值。

表 74: *sortnode* 內容 (繼續)

sortnode 內容	資料類型	內容說明
default_ascending	旗標	指定預設排序順序。
use_existing_keys	旗標	指定是否使用以前已排序欄位的排序來最佳化現在的排序。
existing_keys		指定已排序的欄位及其排序方向。使用的格式與 <i>keys</i> 內容相同。

spacetimeboxes 內容



Space-Time-Boxes (STB) 是一個 Geohash 形式的空間位置延伸。具體而言，STB 是一個英數字串，其代表分佈均勻的空間與時間區域。

表 75: *spacetimeboxes* 內容

spacetimeboxes 內容	資料類型	內容說明
mode	<i>IndividualRecords</i> <i>Hangouts</i>	
latitude_field	欄位	
longitude_field	欄位	
timestamp_field	欄位	
densities	<i>[density, density, density...]</i>	<p>每個密度都是一個字串。例如： STB_GH8_1DAY</p> <p>請注意，對於哪些密度有效，存在一些限制。</p> <p>對於地理雜湊，可以使用 GH1-GH15 中的值。</p> <p>針對時間部分，可以使用下列值：</p> <pre> EVER 1YEAR 1MONTH 1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2HOURS 1HOUR 30MINS 15MINS 10MINS 5MINS 2 MINS 1 MIN 30SECS 15SECS 10SECS 5 SECS 2 SECS 1SEC </pre>

表 75: *spacetimeboxes* 內容 (繼續)

spacetimeboxes 內容	資料類型	內容說明
field_name_extension	<i>string</i>	
add_extension_as	<i>Prefix</i> <i>Suffix</i>	
hangout_density	密度	單一密度 (請參見上方文)
id_field	欄位	
qualifying_duration	1DAY 12HOURS 8HOURS 6HOURS 4HOURS 2HOURS 1HOUR 30MIN 15MIN 10MIN 5MIN 2MIN 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SECS	這必須是字串。
min_events	整數	最小值為 2
qualifying_pct	整數	必須在範圍 1-100 中

streamingtimeseries 內容



「多媒體串流時間序列」節點可在一個步驟中建置時間序列模型並進行評分。

註: 此「串流式時間序列」節點取代 SPSS Modeler 第 18 版中已淘汰使用的原始「串流式時間序列」節點。

表 76: *streamingtimeseries* 內容

streamingtimeseries 內容	值	內容說明
targets	欄位	「串流式時間序列」節點可以預測一個或多個目標，可以選擇使用一個或多個輸入欄位作為預測值。不會使用頻率及加權欄位。請參閱第 191 頁的『 一般建模節點內容 』主題，以取得更多資訊。
candidate_inputs	[<i>field1 ... fieldN</i>]	模型所用的輸入或預測值欄位。
use_period	旗標	
date_time_field	欄位	

表 76: *streamingtimeseries* 內容 (繼續)

streamingtimeseries 內容	值	內容說明
input_interval	None Unknown Year Quarter Month Week Day Hour Hour_nonperiod Minute Minute_nonperiod Second Second_nonperiod	
period_field	欄位	
period_start_value	整數	
num_days_per_week	整數	
start_day_of_week	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	
num_hours_per_day	整數	
start_hour_of_day	整數	
timestamp_increments	整數	

表 76: *streamingtimeseries* 內容 (繼續)

streamingtimeseries 內容	值	內容說明
<code>cyclic_increments</code>	整數	
<code>cyclic_periods</code>	清單	
<code>output_interval</code>	None Year Quarter Month Week Day Hour Minute Second	
<code>is_same_interval</code>	旗標	
<code>cross_hour</code>	旗標	
<code>aggregate_and_distribute</code>	清單	
<code>aggregate_default</code>	Mean Sum Mode Min Max	
<code>distribute_default</code>	Mean Sum	
<code>group_default</code>	Mean Sum Mode Min Max	

表 76: *streamingtimeseries* 內容 (繼續)

streamingtimeseries 內容	值	內容說明
missing_imput	Linear_interp Series_mean K_mean K_median Linear_trend	
k_span_points	整數	
use_estimation_period	旗標	
estimation_period	Observations Times	
date_estimation	清單	僅當使用 date_time_field 時可用
period_estimation	清單	僅當使用 use_period 時可用
observations_type	Latest Earliest	
observations_num	整數	
observations_exclude	整數	
method	ExpertModeler Exsmooth Arima	
expert_modeler_method	ExpertModeler Exsmooth Arima	
consider_seasonal	旗標	
detect_outliers	旗標	
expert_outlier_additive	旗標	
expert_outlier_level_shift	旗標	
expert_outlier_innovational	旗標	
expert_outlier_level_shift	旗標	
expert_outlier_transient	旗標	

表 76: *streamingtimeseries* 內容 (繼續)

streamingtimeseries 內容	值	內容說明
expert_outlier_seasonal_additive	旗標	
expert_outlier_local_trend	旗標	
expert_outlier_additive_patch	旗標	
consider_newesmodels	旗標	
exsmooth_model_type	Simple HoltLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative DampedTrendAdditive DampedTrendMultiplicative MultiplicativeTrendAdditive MultiplicativeSeasonal MultiplicativeTrendMultiplicative MultiplicativeTrend	
futureValue_type_method	Compute specify	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arma.p	整數	
arma.d	整數	

表 76: *streamingtimeseries* 內容 (繼續)

streamingtimeseries 內容	值	內容說明
<code>arma.q</code>	整數	
<code>arma.sp</code>	整數	
<code>arma.sd</code>	整數	
<code>arma.sq</code>	整數	
<code>arma_transformation_type</code>	None SquareRoot NaturalLog	
<code>arma_include_constant</code>	旗標	
<code>tf_arma.p.fieldname</code>	整數	用於轉換函數。
<code>tf_arma.d.fieldname</code>	整數	用於轉換函數。
<code>tf_arma.q.fieldname</code>	整數	用於轉換函數。
<code>tf_arma.sp.fieldname</code>	整數	用於轉換函數。
<code>tf_arma.sd.fieldname</code>	整數	用於轉換函數。
<code>tf_arma.sq.fieldname</code>	整數	用於轉換函數。
<code>tf_arma.delay.fieldname</code>	整數	用於轉換函數。
<code>tf_arma.transformation_type.fieldname</code>	None SquareRoot NaturalLog	用於轉換函數。
<code>arma_detect_outliers</code>	旗標	
<code>arma_outlier_additive</code>	旗標	
<code>arma_outlier_level_shift</code>	旗標	
<code>arma_outlier_innovational</code>	旗標	
<code>arma_outlier_transient</code>	旗標	
<code>arma_outlier_seasonal_additive</code>	旗標	
<code>arma_outlier_local_trend</code>	旗標	
<code>arma_outlier_additive_patch</code>	旗標	
<code>conf_limit_pct</code>	<i>real</i>	
<code>events</code>	欄位	
<code>forecastperiods</code>	整數	
<code>extend_records_into_future</code>	旗標	
<code>conf_limits</code>	旗標	
<code>noise_res</code>	旗標	

streamingts 內容 (已淘汰)



註: 此原始「串流式時間序列」節點在 SPSS Modeler 第 18 版中已已淘汰使用, 並且取代之為旨在利用 IBM SPSS Analytic Server 的能力並處理大資料的新的「串流式時間序列」節點。

「串流時間序列」節點在某個步驟中建立「時間數列」模型並對其進行評分, 而不需要「時間間隔」節點。

範例

```
node = stream.create("streamingts", "My node")
node.setPropertyValue("deployment_force_rebuild", True)
node.setPropertyValue("deployment_rebuild_mode", "Count")
node.setPropertyValue("deployment_rebuild_count", 3)
node.setPropertyValue("deployment_rebuild_pct", 11)
node.setPropertyValue("deployment_rebuild_field", "Year")
```

表 77: *streamingts* 內容

streamingts 內容	資料類型	內容說明
custom_fields	旗標	如果 custom_fields=false, 那麼將使用上游「類型」節點的當前設定。如果 custom_fields=true, 那麼必須指定 targets 和 inputs。
targets	[field1...fieldN]	
inputs	[field1...fieldN]	
method	ExpertModeler Exsmooth Arima	
calculate_conf	旗標	
conf_limit_pct	real	
use_time_intervals_node	旗標	如果 use_time_intervals_node=true, 那麼將使用上游「時間間隔」節點的設定。如果 use_time_intervals_node=false, 那麼必須指定 interval_offset_position、interval_offset 和 interval_type。
interval_offset_position	LastObservation LastRecord	LastObservation 是指最後一個有效觀察。LastRecord 是指從最後一個記錄計數。
interval_offset	數字	
interval_type	Periods Years Quarters Months WeeksNonPeriodic DaysNonPeriodic HoursNonPeriodic MinutesNonPeriodic SecondsNonPeriodic	

表 77: *streamingts* 內容 (繼續)

streamingts 內容	資料類型	內容說明
events	欄位	
expert_modeler_method	AllModels Exsmooth Arima	
consider_seasonal	旗標	
detect_outliers	旗標	
expert_outlier_additive	旗標	
expert_outlier_level_shift	旗標	
expert_outlier_innovational	旗標	
expert_outlier_transient	旗標	
expert_outlier_seasonal_additive	旗標	
expert_outlier_local_trend	旗標	
expert_outlier_additive_patch	旗標	
exsmooth_model_type	Simple HoltLinearTrend BrownLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arma_p	整數	與用於時間序列建模節點的內容相同的內容
arma_d	整數	與用於時間序列建模節點的內容相同的內容
arma_q	整數	與用於時間序列建模節點的內容相同的內容
arma_sp	整數	與用於時間序列建模節點的內容相同的內容
arma_sd	整數	與用於時間序列建模節點的內容相同的內容
arma_sq	整數	與用於時間序列建模節點的內容相同的內容
arma_transformation_type	None SquareRoot NaturalLog	與用於時間序列建模節點的內容相同的內容
arma_include_constant	旗標	與用於時間序列建模節點的內容相同的內容
tf_arma_p. <i>fieldname</i>	整數	與用於時間序列建模節點的內容相同的內容 用於轉換函數。

表 77: *streamingts* 內容 (繼續)

streamingts 內容	資料類型	內容說明
<code>tf_arma_d.fieldname</code>	整數	與用於時間序列建模節點的內容相同的內容用於轉換函數。
<code>tf_arma_q.fieldname</code>	整數	與用於時間序列建模節點的內容相同的內容用於轉換函數。
<code>tf_arma_sp.fieldname</code>	整數	與用於時間序列建模節點的內容相同的內容用於轉換函數。
<code>tf_arma_sd.fieldname</code>	整數	與用於時間序列建模節點的內容相同的內容用於轉換函數。
<code>tf_arma_sq.fieldname</code>	整數	與用於時間序列建模節點的內容相同的內容用於轉換函數。
<code>tf_arma_delay.fieldname</code>	整數	與用於時間序列建模節點的內容相同的內容用於轉換函數。
<code>tf_arma_transformation_type.fieldname</code>	None SquareRoot NaturalLog	
<code>arma_detect_outlier_mode</code>	None Automatic	
<code>arma_outlier_additive</code>	旗標	
<code>arma_outlier_level_shift</code>	旗標	
<code>arma_outlier_innovational</code>	旗標	
<code>arma_outlier_transient</code>	旗標	
<code>arma_outlier_seasonal_additive</code>	旗標	
<code>arma_outlier_local_trend</code>	旗標	
<code>arma_outlier_additive_patch</code>	旗標	
<code>deployment_force_rebuild</code>	旗標	
<code>deployment_rebuild_mode</code>	Count Percent	
<code>deployment_rebuild_count</code>	數字	
<code>deployment_rebuild_pct</code>	數字	
<code>deployment_rebuild_field</code>	<field>	

第 11 章 欄位作業節點內容

anonymizenode 內容



「匿名化」節點用於轉換欄位名稱和欄位值的下游代表方式，從而掩飾了原始資料。如果要容許其他使用者使用敏感資料（如客戶名稱或其他詳細資料）建立模型，這種節點將十分有用。

範例

```
stream = modeler.script.stream()
varfilenode = stream.createAt("variablefile", "File", 96, 96)
varfilenode.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")
node = stream.createAt("anonymize", "My node", 192, 96)
# Anonymize node requires the input fields while setting the values
stream.link(varfilenode, node)
node.setKeyedPropertyValue("enable_anonymize", "Age", True)
node.setKeyedPropertyValue("transformation", "Age", "Random")
node.setKeyedPropertyValue("set_random_seed", "Age", True)
node.setKeyedPropertyValue("random_seed", "Age", 123)
node.setKeyedPropertyValue("enable_anonymize", "Drug", True)
node.setKeyedPropertyValue("use_prefix", "Drug", True)
node.setKeyedPropertyValue("prefix", "Drug", "myprefix")
```

表 78: anonymizenode 內容

anonymizenode 內容	資料類型	內容說明
enable_anonymize	旗標	設定為 True 時，將啟動欄位值的匿名化（相當於在「對值進行匿名化」欄中為該欄位選取是）。
use_prefix	旗標	設定為 True 時，如果已指定自訂字首，那麼將使用該字首。適用於將通過雜湊法匿名化的欄位，而且等效於在「取代數值」對話框中為該欄位選中自訂圓鈕。
prefix	string	等效於在「取代值」對話框的文字框中鍵入字首。如果未指定其他任何值，那麼預設字首是該預設值。
transformation	Random Fixed	確定通過轉換法匿名化的欄位的轉換參數是隨機的還是固定的。
set_random_seed	旗標	設定為 True 時，將使用指定的種子值（如果 transformation 也設定為 Random）。
random_seed	整數	當 set_random_seed 設定為 True 時，該值是亂數的種子。
scale	數字	當 transformation 設定為 Fixed 時，此值用於「乘數」。通常，最大刻度值為 10，但可能會減小以避免溢位。
translate	數字	當 transformation 設定為 Fixed 時，此值用於「變換」。通常，上限轉換值為 1000，但可能會減小以避免溢位。

autodatapreprenode 內容



自動資料預備 (ADP) 節點可分析您的資料並識別修正，篩選出存在問題或可能無用的欄位，並在適當的情況下衍生新的屬性，通過智能篩選和取樣技術改進效能。您可以完全自動化地使用節點，容許節點選擇並套用修正程式，或者可在進行變更前預覽變更，並按照需要接受、拒絕或修正變更。

範例

```
node = stream.create("autodataprep", "My node")
node.setPropertyValue("objective", "Balanced")
node.setPropertyValue("excluded_fields", "Filter")
node.setPropertyValue("prepare_dates_and_times", True)
node.setPropertyValue("compute_time_until_date", True)
node.setPropertyValue("reference_date", "Today")
node.setPropertyValue("units_for_date_durations", "Automatic")
```

表 79: autodatapreprenode 內容

autodatapreprenode 內容	資料類型	內容說明
objective	Balanced Speed Accuracy Custom	
custom_fields	旗標	如果為 true，則容許您為現行節點指定目標、輸入及其他欄位。如果為 false，則會使用上游「類型」節點中的現行設定。
target	欄位	指定單一目標欄位。
inputs	[field1 ... fieldN]	模型所用的輸入或預測值欄位。
use_frequency	旗標	
frequency_field	欄位	
use_weight	旗標	
weight_field	欄位	
excluded_fields	Filter None	
if_fields_do_not_match	StopExecution ClearAnalysis	
prepare_dates_and_times	旗標	控制對所有日期和時間欄位的存取
compute_time_until_date	旗標	
reference_date	Today Fixed	
fixed_date	日期	

表 79: autodatapreprenode 內容 (繼續)

autodatapreprenode 內容	資料類型	內容說明
units_for_date_durations	Automatic	
	Fixed	
fixed_date_units	Years	
	Months	
	Days	
compute_time_until_time	旗標	
reference_time	CurrentTime	
	Fixed	
fixed_time	時間	
units_for_time_durations	Automatic	
	Fixed	
fixed_date_units	Hours	
	Minutes	
	Seconds	
extract_year_from_date	旗標	
extract_month_from_date	旗標	
extract_day_from_date	旗標	
extract_hour_from_time	旗標	
extract_minute_from_time	旗標	
extract_second_from_time	旗標	
exclude_low_quality_inputs	旗標	
exclude_too_many_missing	旗標	
maximum_percentage_missing	數字	
exclude_too_many_categories	旗標	
maximum_number_categories	數字	
exclude_if_large_category	旗標	
maximum_percentage_category	數字	
prepare_inputs_and_target	旗標	

表 79: autodatapreprenode 內容 (繼續)

autodatapreprenode 內容	資料類型	內容說明
adjust_type_inputs	旗標	
adjust_type_target	旗標	
reorder_nominal_inputs	旗標	
reorder_nominal_target	旗標	
replace_outliers_inputs	旗標	
replace_outliers_target	旗標	
replace_missing_continuous_inputs	旗標	
replace_missing_continuous_target	旗標	
replace_missing_nominal_inputs	旗標	
replace_missing_nominal_target	旗標	
replace_missing_ordinal_inputs	旗標	
replace_missing_ordinal_target	旗標	
maximum_values_for_ordinal	數字	
minimum_values_for_continuous	數字	
outlier_cutoff_value	數字	
outlier_method	Replace Delete	
rescale_continuous_inputs	旗標	
rescaling_method	MinMax ZScore	
min_max_minimum	數字	
min_max_maximum	數字	
z_score_final_mean	數字	
z_score_final_sd	數字	
rescale_continuous_target	旗標	
target_final_mean	數字	
target_final_sd	數字	
transform_select_input_fields	旗標	

表 79: autodatapreprenode 內容 (繼續)

autodatapreprenode 內容	資料類型	內容說明
maximize_association_with_target	旗標	
p_value_for_merging	數字	
merge_ordinal_features	旗標	
merge_nominal_features	旗標	
minimum_cases_in_category	數字	
bin_continuous_fields	旗標	
p_value_for_binning	數字	
perform_feature_selection	旗標	
p_value_for_selection	數字	
perform_feature_construction	旗標	
transformed_target_name_extension	string	
transformed_inputs_name_extension	string	
constructed_features_root_name	string	
years_duration_name_extension	string	
months_duration_name_extension	string	
days_duration_name_extension	string	
hours_duration_name_extension	string	
minutes_duration_name_extension	string	
seconds_duration_name_extension	string	
year_cyclical_name_extension	string	
month_cyclical_name_extension	string	
day_cyclical_name_extension	string	
hour_cyclical_name_extension	string	
minute_cyclical_name_extension	string	

表 79: autodatapreinode 內容 (繼續)

autodatapreinode 內容	資料類型	內容說明
second_cyclical_name_extension	string	

astimeintervalsnode 內容



使用「時間間隔」節點可以指定時間間隔並衍生用於估計或預測的新時間欄位。支援從秒到年的全範圍時間區間。

表 80: astimeintervalsnode 內容

astimeintervalsnode 內容	資料類型	內容說明
time_field	欄位	只能接受單個連續欄位。此節點將該欄位用作聚集鍵，以轉換時間間隔。如果在此處使用了整數欄位，那麼會將其視為時間指標。
dimensions	[field1 field2 ... fieldn]	這些欄位用於根據欄位值來建立各個時間序列。
fields_to_aggregate	[field1 field2 ... fieldn]	在變更時間欄位的時間段的過程中，這些欄位將進行聚合。在離開此節點的資料中，將過濾掉所有未併入在此選取器中的欄位。

binningnode 內容



「分組」節點根據一個或多個現有連續（數值型範圍）欄位的值自動建立新的列名（集合）欄位。例如，用戶可將連續收入欄位轉換為一個包含各群組收入的新的種類欄位，作為其與平均數之間的離差。一旦您為新欄位建立了分組，就可以根據切割點來產生「衍生」節點。

範例

```
node = stream.create("binning", "My node")
node.setPropertyValue("fields", ["Na", "K"])
node.setPropertyValue("method", "Rank")
node.setPropertyValue("fixed_width_name_extension", "_binned")
node.setPropertyValue("fixed_width_add_as", "Suffix")
node.setPropertyValue("fixed_bin_method", "Count")
node.setPropertyValue("fixed_bin_count", 10)
node.setPropertyValue("fixed_bin_width", 3.5)
node.setPropertyValue("tile10", True)
```

表 81: binningnode 內容

binningnode 內容	資料類型	內容說明
fields	[field1 field2 ... fieldn]	要進行轉換的連續（數值型範圍）欄位。可以同時對多個欄位進行分級。

表 81: binningnode 內容 (繼續)

binningnode 內容	資料類型	內容說明
method	FixedWidth EqualCount Rank SDev Optimal	用於為新欄位「分組」（種類）確定分割點的方法。
rcalculate_bins	Always IfNecessary	指定是每一次執行節點時都重新計算「分組」且將資料放置在相關「分組」中，還是僅將資料新增到現有「分組」和已新增的新「分組」中。
fixed_width_name_extension	string	預設副檔名為 <i>_BIN</i> 。
fixed_width_add_as	Suffix Prefix	指定是將副檔名新增到欄位名稱的末尾（字尾）還是開頭（字首）。預設副檔名為 <i>income_BIN</i> 。
fixed_bin_method	Width Count	
fixed_bin_count	整數	指定用於確定新欄位的固定寬度「分組」（種類）數的整數。
fixed_bin_width	real	用於計算分級寬度的值（整數或實數）。
equal_count_name_extension	string	預設副檔名為 <i>_TILE</i> 。
equal_count_add_as	Suffix Prefix	指定用於使用標準 p 分位顯示法產生的欄位名稱的副檔名（字尾或字首）。預設副檔名是 <i>_TILE</i> 加上 <i>N</i> ，其中 <i>N</i> 是分位號。
tile4	旗標	產生四分位數「分組」，每個「分組」中包含 25% 的觀察值。
tile5	旗標	產生五分位數「分組」。
tile10	旗標	產生十分位數「分組」。
tile20	旗標	產生二十分位數「分組」。
tile100	旗標	產生百分位數「分組」。
use_custom_tile	旗標	
custom_tile_name_extension	string	預設副檔名為 <i>_TILEN</i> 。
custom_tile_add_as	Suffix Prefix	
custom_tile	整數	

表 81: binningnode 內容 (繼續)

binningnode 內容	資料類型	內容說明
equal_count_method	RecordCount ValueSum	RecordCount 方法是為每個分級分配相同號碼的記錄，而 ValueSum 方法是使分配記錄後每個分級中值總和相等。
tied_values_method	Next Current Random	指定要輸入的分級結值資料。
rank_order	Ascending Descending	此內容包含 Ascending (最低值標示為 1) 或 Descending (最高值標示為 1)。
rank_add_as	Suffix Prefix	此選項適用於排等級、分數等級和百分比排等級。
rank	旗標	
rank_name_extension	string	預設副檔名為 <i>_RANK</i> 。
rank_fractional	旗標	排等級觀察值，其中新欄位的值是排等級值除以非遺漏觀察值的加權和。分數排名範圍在 0-1 之間。
rank_fractional_name_ extension	string	預設副檔名為 <i>_F_RANK</i> 。
rank_pct	旗標	每個等級皆除以具備有效值的記錄數目，然後乘以 100。分數排名百分比範圍在 1-100 之間。
rank_pct_name_extension	string	預設副檔名為 <i>_P_RANK</i> 。
sdev_name_extension	string	
sdev_add_as	Suffix Prefix	
sdev_count	One Two Three	
optimal_name_extension	string	預設副檔名為 <i>_OPTIMAL</i> 。
optimal_add_as	Suffix Prefix	
optimal_supervisor_field	欄位	作為監督欄位選取的欄位，為分組選取的欄位與之相關。

表 81: binningnode 內容 (繼續)

binningnode 內容	資料類型	內容說明
optimal_merge_bins	旗標	指定將所有具有較小觀察值計數的「分組」新增到更大的相鄰「分組」。
optimal_small_bin_threshold	整數	
optimal_pre_bin	旗標	表示要進行資料集的預先分組。
optimal_max_bins	整數	指定上限以避免建立過大「分組」個數。
optimal_lower_end_point	Inclusive	
	Exclusive	
optimal_first_bin	Unbounded	
	Bounded	
optimal_last_bin	Unbounded	
	Bounded	

derivednode 內容



「衍生」節點將修改資料值或根據一個或多個現有欄位建立新欄位。它會建立類型為公式、旗標、列名、狀態、計數及條件式的欄位。

範例 1

```
# Create and configure a Flag Derive field node
node = stream.create("derive", "My node")
node.setPropertyValue("new_name", "DrugX_Flag")
node.setPropertyValue("result_type", "Flag")
node.setPropertyValue("flag_true", "1")
node.setPropertyValue("flag_false", "0")
node.setPropertyValue("flag_expr", "'Drug' == \"drugX\"")

# Create and configure a Conditional Derive field node
node = stream.create("derive", "My node")
node.setPropertyValue("result_type", "Conditional")
node.setPropertyValue("cond_if_cond", "@OFFSET(\"Age\", 1) = \"Age\"")
node.setPropertyValue("cond_then_expr", "@OFFSET(\"Age\", 1) = \"Age\" >> @INDEX")
node.setPropertyValue("cond_else_expr", "\"Age\"")
```

範例 2

此 Script 假定出現兩個分別名為 XPos 和 YPos 的數值欄，這兩個欄分別代表一個點（例如事件發生位置）的 X 和 Y 座標。此 Script 建立「衍生」節點，該節點根據特定座標系統中代表該點的 X 和 Y 座標來計算地理空間欄：

```
stream = modeler.script.stream()
# Other stream configuration code
node = stream.createAt("derive", "Location", 192, 96)
node.setPropertyValue("new_name", "Location")
```

```

node.setPropertyValue("formula_expr", "['XPos', 'YPos']")
node.setPropertyValue("formula_type", "Geospatial")
# Now we have set the general measurement type, define the
# specifics of the geospatial object
node.setPropertyValue("geo_type", "Point")
node.setPropertyValue("has_coordinate_system", True)
node.setPropertyValue("coordinate_system", "ETRS_1989_EPSG_Arctic_zone_5-47")

```

表 82: *derivemode* 內容

derivemode 內容	資料類型	內容說明
<code>new_name</code>	<i>string</i>	新欄位的名稱。
<code>mode</code>	Single Multiple	指定單個或多個欄位。
<code>fields</code>	清單	僅用於在多模式下選取多個欄位。
<code>name_extension</code>	<i>string</i>	為新欄位名稱指定副檔名。
<code>add_as</code>	Suffix Prefix	將副檔名新增為欄位名的字首（開頭）或字尾（末尾）。
<code>result_type</code>	Formula Flag Set State Count Conditional	可建立的六種類型的新欄位。
<code>formula_expr</code>	<i>string</i>	用於在衍生節點計算新欄位值的表示式。
<code>flag_expr</code>	<i>string</i>	
<code>flag_true</code>	<i>string</i>	
<code>flag_false</code>	<i>string</i>	
<code>set_default</code>	<i>string</i>	
<code>set_value_cond</code>	<i>string</i>	提供與給定值相關聯的條件的結構化屬性。
<code>state_on_val</code>	<i>string</i>	指定符合 On 條件時新欄位的值。
<code>state_off_val</code>	<i>string</i>	指定符合 Off 條件時新欄位的值。
<code>state_on_expression</code>	<i>string</i>	
<code>state_off_expression</code>	<i>string</i>	
<code>state_initial</code>	On Off	為新欄位的每個記錄分配初始值 On 或 Off。可在符合每個條件時變更此值。

表 82: *derivenode* 內容 (繼續)

derivenode 內容	資料類型	內容說明
count_initial_val	<i>string</i>	
count_inc_condition	<i>string</i>	
count_inc_expression	<i>string</i>	
count_reset_condition	<i>string</i>	
cond_if_cond	<i>string</i>	
cond_then_expr	<i>string</i>	
cond_else_expr	<i>string</i>	
formula_measure_type	Range / MeasureType.RANGE Discrete / MeasureType.DISCRETE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS Collection / MeasureType.COLLECTION Geospatial / MeasureType.GEOSPATIAL	此內容可用於定義與衍生的欄位相關聯的測量。可以向 setter 函數傳送字串或者其中一個 MeasureType 值。 getter 函數始終傳回 MeasureType 值。
collection_measure	Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS	對於收集欄位（深度為 0 的清單），此內容定義與底層值相關聯的測量類型。

表 82: *derivenode* 內容 (繼續)

derivenode 內容	資料類型	內容說明
geo_type	Point MultiPoint LineString MultiLineString Polygon MultiPolygon	對於地理空間欄位，此內容定義此欄位所代表的地理空間物件的類型。這應該與這些值的清單深度一致。
has_coordinate_system	<i>boolean</i>	對於地理空間欄位，此內容會定義此欄位是否具有座標系統
coordinate_system	<i>string</i>	對於地理空間欄位，此內容定義此欄位的座標系統。

ensembledenode 內容



整體節點結合兩個或多個模型塊，以獲得比從任何一個模型可以獲得的更準確的預測。

範例

```
# Create and configure an Ensemble node
# Use this node with the models in demos\streams\pm_binaryclassifier.str
node = stream.create("ensemble", "My node")
node.setPropertyValue("ensemble_target_field", "response")
node.setPropertyValue("filter_individual_model_output", False)
node.setPropertyValue("flag_ensemble_method", "ConfidenceWeightedVoting")
node.setPropertyValue("flag_voting_tie_selection", "HighestConfidence")
```

表 83: *ensembledenode* 內容

ensembledenode 內容	資料類型	內容說明
ensemble_target_field	欄位	為在總體中使用的所有模型指定目標欄位。
filter_individual_model_output	旗標	指定是否應抑制各個模型的評分結果。

表 83: *ensemblenode* 內容 (繼續)

ensemblenode 內容	資料類型	內容說明
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting AdjustedPropensityWeightedVoting HighestConfidence AverageRawPropensity AverageAdjustedPropensity	指定用來判定整體評分的方法。僅當選取的目標是旗標欄位時，此設定才適用。
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	指定用來判定整體評分的方法。唯有當選取的目標是標稱欄位時才會套用此設定。
flag_voting_tie_selection	Random HighestConfidence RawPropensity AdjustedPropensity	如果選取投票方法，則指定鏈結的解析方法。僅當選取的目標是旗標欄位時，此設定才適用。
set_voting_tie_selection	Random HighestConfidence	如果選取投票方法，則指定鏈結的解析方法。唯有當選取的目標是標稱欄位時才會套用此設定。
calculate_standard_error	旗標	如果目標欄位是連續的，則依預設會執行標準錯誤計算以計算測量或估計值與 true 值之間的差分；並顯示這些估計值的相符程度。

fillernode 內容



「填入器」節點會取代欄位值並變更儲存。您可以選擇根據 CLEM 條件 (例如 @BLANK(@FIELD)) 來取代值。或者，您可以選擇將所有空白或空值取代為特定值。填充值節點通常與「類型」節點一起使用以取代遺漏值。

範例

```
node = stream.create("filler", "My node")
node.setPropertyValue("fields", ["Age"])
node.setPropertyValue("replace_mode", "Always")
```

```
node.setPropertyValue("condition", "(\"Age\" > 60) and (\"Sex\" = \"M\")")
node.setPropertyValue("replace_with", "\"old man\"")
```

表 84: fillernode 內容

fillernode 內容	資料類型	內容說明
fields	清單	資料集中其值將被檢查並取代的欄位。
replace_mode	Always Conditional Blank Null BlankAndNull	可以取代所有值、空白值或空值，也可以根據指定條件進行取代。
condition	string	
replace_with	string	

filternode 內容



「過濾器」節點用於過濾（捨棄）欄位、重新命名欄位並將欄位從一個來源節點對映到另一來源節點。

範例：

```
node = stream.create("filter", "My node")
node.setPropertyValue("default_include", True)
node.setKeyedPropertyValue("new_name", "Drug", "Chemical")
node.setKeyedPropertyValue("include", "Drug", False)
```

使用 **default_include** 內容。請注意，設定 `default_include` 內容的值不會自動包含或排除所有欄位；它只確定針對目前選定欄位的預設行為。這在功能上等效於按一下「過濾器節點」對話框中的**依預設包含欄位**按鈕。例如，假設執行下列 Script：

```
node = modeler.script.stream().create("filter", "Filter")
node.setPropertyValue("default_include", False)
# Include these two fields in the list
for f in ["Age", "Sex"]:
    node.setKeyedPropertyValue("include", f, True)
```

這將導致節點傳遞欄位 `Age` 和 `Sex` 並捨棄所有其他欄位。在執行前一個 Script 之後，現在假設您將下列行新增至 Script 以命名其他兩個欄位：

```
node.setPropertyValue("default_include", False)
# Include these two fields in the list
for f in ["BP", "Na"]:
    node.setKeyedPropertyValue("include", f, True)
```

這會將另外兩個欄位新增至過濾器，以便總共傳遞四個欄位（`Age`、`Sex`、`BP`、`Na`）。換句話說，將 `default_include` 的值重設為 `False` 不會自動重設所有欄位。

此外，如果現在透過使用 Script 或在「過濾器節點」對話框中將 `default_include` 的值變更為 `True`，則會使此行為發生顛倒，即，將捨棄而非併入上方列出的四個欄位。如果有疑問，可使用「過濾器節點」對話框中的控制項進行實驗，這將有助於理解此互動效應。

表 85: <i>filternode</i> 內容		
filternode 內容	資料類型	內容說明
<code>default_include</code>	旗標	索引內容，用來指定預設行為是傳送欄位還是過濾欄位： 注意，設定此內容不會自動併入或排除所有欄位；它只確定依預設是併入還是排除所選取的欄位。有關其他備註請參見下方的範例。
<code>include</code>	旗標	用於包括和移除欄位的鍵控內容。
<code>new_name</code>	<i>string</i>	

historynode 內容



「歷史」節點將建立新欄位，其中包含之前記錄中的欄位資料。「歷程」節點通常用於序列資料，例如時間序列資料。在使用「歷程」節點之前，您可能想要使用「排序」節點對資料進行排序。

範例

```
node = stream.create("history", "My node")
node.setPropertyValue("fields", ["Drug"])
node.setPropertyValue("offset", 1)
node.setPropertyValue("span", 3)
node.setPropertyValue("unavailable", "Discard")
node.setPropertyValue("fill_with", "undef")
```

表 86: <i>historynode</i> 內容		
historynode 內容	資料類型	內容說明
<code>fields</code>	清單	需要其歷史的欄位。
<code>offset</code>	數字	指定要從中擷取歷程記錄欄位值的最新的記錄（現行記錄之前的記錄）。
<code>span</code>	數字	指定要從中擷取值的以前記錄的號碼。
<code>unavailable</code>	Discard Leave Fill	處理不含歷史值的記錄時，通常參考沒有以前的記錄作為歷史的前幾個記錄（位於資料集頂部）。
<code>fill_with</code>	String Number	指定要用於無歷史值可用的記錄的值或字串。

partitionnode 內容



分割區節點可產生分割區欄位，該欄位可將資料分割為個別的子集以便在模型建置的訓練、正在測試和驗證階段使用。

範例

```
node = stream.create("partition", "My node")
node.setPropertyValue("create_validation", True)
node.setPropertyValue("training_size", 33)
node.setPropertyValue("testing_size", 33)
node.setPropertyValue("validation_size", 33)
node.setPropertyValue("set_random_seed", True)
node.setPropertyValue("random_seed", 123)
node.setPropertyValue("value_mode", "System")
```

表 87: partitionnode 內容

partitionnode 內容	資料類型	內容說明
new_name	string	由節點產生的分割區欄位的名稱。
create_validation	旗標	指定是否應建立驗證分割區。
training_size	整數	要配置給訓練分割區的記錄所佔的百分比 (0-100)。
testing_size	整數	要配置給測試分割區的記錄所佔的百分比 (0-100)。
validation_size	整數	要配置給驗證分割區的記錄所佔的百分比 (0-100)。如果不建立驗證分割區，則會忽略。
training_label	string	訓練分割區的標籤。
testing_label	string	測試分割區的標籤。
validation_label	string	驗證分割區的標籤。如果不建立驗證分割區，則會忽略。
value_mode	System SystemAndLabel Label	指定用於代表資料中每個分割區的值。例如，訓練樣本可以代表為系統整數 1、標籤 Training 或二者的組合 1_Training。
set_random_seed	布林	指定是否應使用使用者指定的隨機種子。
random_seed	整數	使用者指定的隨機種子值。如果要使用此值，set_random_seed 必須設定為 True。
enable_sql_generation	布林	指定是否使用 SQL 回送以分配記錄到分割區。
unique_field		指定輸入欄位，用以確保以隨機但可重複的方式將記錄分配到分割區。如果要使用此值，enable_sql_generation 必須設定為 True。

reclassifynode 內容



「重新分類」節點可將一組種類值轉換為另一組值。重新分類對收合種類或重新分組分析資料非常有用。

範例

```
node = stream.create("reclassify", "My node")
node.setPropertyValue("mode", "Multiple")
node.setPropertyValue("replace_field", True)
node.setPropertyValue("field", "Drug")
node.setPropertyValue("new_name", "Chemical")
node.setPropertyValue("fields", ["Drug", "BP"])
node.setPropertyValue("name_extension", "reclassified")
node.setPropertyValue("add_as", "Prefix")
node.setKeyedPropertyValue("reclassify", "drugA", True)
node.setPropertyValue("use_default", True)
node.setPropertyValue("default", "BrandX")
node.setPropertyValue("pick_list", ["BrandX", "Placebo", "Generic"])
```

表 88: reclassifynode 內容

reclassifynode 內容	資料類型	內容說明
mode	Single Multiple	Single 重新分類一個欄位的種類。 Multiple 將啟動能夠轉換多個欄位的選項。
replace_field	旗標	
field	string	僅在「單一」模式中使用。
new_name	string	僅在「單一」模式中使用。
fields	[field1 field2 ... fieldn]	僅在「多個」模式下使用。
name_extension	string	僅在「多個」模式下使用。
add_as	Suffix Prefix	僅在「多個」模式下使用。
reclassify	string	欄位值的結構化內容。
use_default	旗標	使用預設值。
default	string	指定預設值。
pick_list	[string string ... string]	容許使用者匯入已知新值的清單以移入表格中的下拉清單。

reordernode 內容



「欄位重新排序」節點定義了用於顯示下游欄位的自然順序。此順序會影響欄位在各個位置（例如表格、清單與「欄位」選擇器）中的顯示方式。當使用廣泛的資料集使感興趣的欄位更可見時，此作業非常有用。

範例

```
node = stream.create("reorder", "My node")
node.setPropertyValue("mode", "Custom")
node.setPropertyValue("sort_by", "Storage")
node.setPropertyValue("ascending", False)
node.setPropertyValue("start_fields", ["Age", "Cholesterol"])
node.setPropertyValue("end_fields", ["Drug"])
```

表 89: *reordernode* 內容

reordernode 內容	資料類型	內容說明
mode	Custom Auto	可以自動對值進行排序，也可以指定自訂順序。
sort_by	Name Type Storage	
ascending	旗標	
start_fields	[<i>field1 field2 ... fieldn</i>]	新欄位插入到這些欄位之後。
end_fields	[<i>field1 field2 ... fieldn</i>]	新欄位插入到這些欄位之前。

reprojectnode 內容



在 SPSS Modeler 中，表示式建置器空間函數、「空間-時間預測」(STP) 節點和「地圖視覺化」節點之類的项目使用投影座標系統。使用「重新投影」節點可以變更所匯入的任何使用了地理座標系統的資料的座標系統。

表 90: *reprojectnode* 內容

reprojectnode 內容	資料類型	內容說明
reproject_fields	[<i>field1 field2 ... fieldn</i>]	列出所有要進行重新投影的欄位。
reproject_type	Streamdefault Specify	選擇如何對欄位進行重新投影。
coordinate_system	<i>string</i>	要套用至欄位的座標系統的名稱。範例： set reprojectnode.coordinate_system = "WGS_1984_World_Mercator"

restructurenode 內容



「重組」節點可將一個名義欄位或旗標欄位轉換為一組欄位（該欄位群組可移入另一個欄位的值）。例如，假設有一個名為付款類型的欄位，其值為信用、現金和借項，則會建立三個新欄位（信用、現金、借項），每一個新欄位可能包含實際付款的值。

範例

```
node = stream.create("restructure", "My node")
node.setKeyedPropertyValue("fields_from", "Drug", ["drugA", "drugX"])
node.setPropertyValue("include_field_name", True)
node.setPropertyValue("value_mode", "OtherFields")
node.setPropertyValue("value_fields", ["Age", "BP"])
```

表 91: restructurenode 內容

restructurenode 內容	資料類型	內容說明
fields_from	[category category category] all	
include_field_name	旗標	指示是否在重組的欄位名稱中使用欄位名稱。
value_mode	OtherFields Flags	表示用於為重組欄位指定值的模式。如果選擇 OtherFields，必須指定要使用哪些欄位（參閱下方文）。如果選擇 Flags，則值為數值型旗標。
value_fields	清單	如果 value_mode 是 OtherFields，則此屬性是必要的。指定使用哪些欄位作為值欄位。

rfmanalysisnode 內容



「回頭率、頻率、消費金額 (RFM) 分析」節點可讓您透過檢查客戶前次向您購買的時間（回頭率）、他們購買的頻率（頻率），以及他們在所有交易中花費的金額（消費金額），來定量判定哪些客戶可能是最佳客戶。

範例

```
node = stream.create("rfmanalysis", "My node")
node.setPropertyValue("recency", "Recency")
node.setPropertyValue("frequency", "Frequency")
node.setPropertyValue("monetary", "Monetary")
node.setPropertyValue("tied_values_method", "Next")
node.setPropertyValue("recalculate_bins", "IfNecessary")
node.setPropertyValue("recency_thresholds", [1, 500, 800, 1500, 2000, 2500])
```

表 92: rfmanalysisnode 內容

rfmanalysisnode 內容	資料類型	內容說明
recency	欄位	指定回頭率欄位。這可以是日期、時間戳記或簡單數字。
frequency	欄位	指定頻率欄位。

表 92: rfmanalysisnode 內容 (繼續)

rfmanalysisnode 內容	資料類型	內容說明
monetary	欄位	指定消費金額欄位。
recency_bins	整數	指定要產生的回頭率「分組」數量。
recency_weight	數字	指定套用至回頭率資料的加權。預設值是 100。
frequency_bins	整數	指定要產生的頻率「分組」數量。
frequency_weight	數字	指定套用至頻率資料的加權。預設值是 10。
monetary_bins	整數	指定要產生的消費金額「分組」數量。
monetary_weight	數字	指定套用至消費金額資料的加權。預設值是 1。
tied_values_method	Next Current	指定要輸入的分級連結值資料。
recalculate_bins	Always IfNecessary	
add_outliers	旗標	只有當 recalculate_bins 設為 IfNecessary 時才可用。如果已設定，則將位於下限分級以下的記錄新增到下限分級中，並且將最高分級以上的記錄新增到最高分級中。
binned_field	Recency Frequency Monetary	
recency_thresholds	值值	只有當 recalculate_bins 設為 Always 時才可用。指定回頭率「分組」的下限臨界值和上限臨界值。一個「分組」的上限臨界值用作下一個「分組」的下限臨界值例如，[10 30 60] 可定義兩個「分組」，第一個「分組」的上限臨界值和下限臨界值分別為 10 和 30，第二個「分組」的兩個臨界值分別為 30 和 60。
frequency_thresholds	值值	只有當 recalculate_bins 設為 Always 時才可用。
monetary_thresholds	值值	只有當 recalculate_bins 設為 Always 時才可用。

settoflagnode 內容



「設為旗標」節點根據定義給一或多個標準欄位的種類值衍生多個旗標欄位。

範例

```
node = stream.create("settoflag", "My node")
node.setKeyedPropertyValue("fields_from", "Drug", ["drugA", "drugX"])
node.setPropertyValue("true_value", "1")
node.setPropertyValue("false_value", "0")
node.setPropertyValue("use_extension", True)
node.setPropertyValue("extension", "Drug_Flag")
node.setPropertyValue("add_as", "Suffix")
node.setPropertyValue("aggregate", True)
node.setPropertyValue("keys", ["Cholesterol"])
```

表 93: settoflagnode 內容

settoflagnode 內容	資料類型	內容說明
fields_from	[category category category] all	
true_value	string	指定設定旗標時節點所使用的 true 值。預設值是 T。
false_value	string	指定設定旗標時節點所使用的 false。預設值是 F。
use_extension	旗標	使用副檔名作為新旗標欄位的字尾或字首。
extension	string	
add_as	Suffix Prefix	指定所新增的副檔名是字尾還是字首。
aggregate	旗標	根據索引鍵欄位將記錄群組。如果有任何記錄被設定為 true，則會啟用群組中的所有旗標欄位。
keys	清單	索引鍵欄位。

statisticstransformnode 內容



Statistics 轉換節點針對 IBM SPSS Modeler 中的資料來源執行所選的 IBM SPSS Statistics 語法指令。此節點需要 IBM SPSS Statistics 的授權副本。

有關此節點內容的資訊，請參閱第 379 頁的『[statisticstransformnode 內容](#)』。

timeintervalsnode 內容 (已淘汰)



註: 此節點在 SPSS Modeler 第 18 版中已淘汰使用，並且取代為新的「時間序列」節點。

時間間隔節點指定區間，建立用於對時間序列資料進行建模的標籤（如果需要）。如果值的間隔不均勻，那麼此節點可以根據需要填補或總計值以產生統一的記錄間隔。

範例

```

node = stream.create("timeintervals", "My node")
node.setPropertyValue("interval_type", "SecondsPerDay")
node.setPropertyValue("days_per_week", 4)
node.setPropertyValue("week_begins_on", "Tuesday")
node.setPropertyValue("hours_per_day", 10)
node.setPropertyValue("day_begins_hour", 7)
node.setPropertyValue("day_begins_minute", 5)
node.setPropertyValue("day_begins_second", 17)
node.setPropertyValue("mode", "Label")
node.setPropertyValue("year_start", 2005)
node.setPropertyValue("month_start", "January")
node.setPropertyValue("day_start", 4)
node.setKeyedPropertyValue("pad", "AGE", "MeanOfRecentPoints")
node.setPropertyValue("agg_mode", "Specify")
node.setPropertyValue("agg_set_default", "Last")

```

表 94: *timeintervals*node 內容

timeintervals node 內容	資料類型	內容說明
interval_type	None Periods CyclicPeriods Years Quarters Months DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic	
mode	Label Create	指定是要連續標記記錄還是要根據指定日期、時間戳記或時間欄位建立數列。
field	欄位	當根據資料建立數列時，指定表示每個記錄的日期或時間的欄位。

表 94: *timeintervalsnode* 內容 (繼續)

timeintervalsnode 內容	資料類型	內容說明
period_start	整數	指定期間或週期性時期的起始區間
cycle_start	整數	週期性時期的起始期間。
year_start	整數	對於適用的區間類型，指第一個區間所屬的年份。
quarter_start	整數	對於適用的區間類型，指第一個區間所屬的季度。
month_start	January February March April May June July August September October November December	
day_start	整數	
hour_start	整數	
minute_start	整數	
second_start	整數	
periods_per_cycle	整數	對於週期性時期，指每個期間中的期間數。
fiscal_year_begins	January February March April May June July August September October November December	對於季度區間，指定財政年度開始的月。

表 94: `timeintervalsnode` 內容 (繼續)

timeintervalsnode 內容	資料類型	內容說明
<code>week_begins_on</code>	Sunday Monday Tuesday Wednesday Thursday Friday Saturday Sunday	對於週期性區間（一週中的天、一天中的小時、一天中的分鐘和一天中的秒），指定一週開始的那一天。
<code>day_begins_hour</code>	整數	對於週期性區間（一天中的小時、一天中的分鐘和一天中的秒），指定一天開始的小時。可以與 <code>day_begins_minute</code> 和 <code>day_begins_second</code> 結合起來使用，以指定一個準確時間，例如 <code>8:05:01</code> 。請參見下方的使用範例。
<code>day_begins_minute</code>	整數	對於週期性區間（一天中的小時、一天中的分鐘和一天中的秒），指定一天開始的分鐘（例如 <code>8:05</code> 中的 5）。
<code>day_begins_second</code>	整數	對於週期性區間（一天中的小時、一天中的分鐘和一天中的秒），指定一天開始的秒（例如 <code>8:05:17</code> 中的 17）。
<code>days_per_week</code>	整數	對於週期性區間（一週中的天、一天中的小時、一天中的分鐘和一天中的秒），指定一週中的天數。
<code>hours_per_day</code>	整數	對於週期性區間（一天中的小時、一天中的分鐘和一天中的秒），指定一天中的小時數。

表 94: *timeintervalsnode* 內容 (繼續)

timeintervalsnode 內容	資料類型	內容說明
interval_increment	1 2 3 4 5 6 10 15 20 30	對於一天中的分鐘和一天中的秒，指定為每個記錄增加的分鐘數或秒數。
field_name_extension	<i>string</i>	
field_name_extension_as_prefix	旗標	
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	

表 94: `timeintervalsnode` 內容 (繼續)

timeintervalsnode 內容	資料類型	內容說明
<code>time_format</code>	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	
<code>aggregate</code>	Mean Sum Mode Min Max First Last TrueIfAnyTrue	指定欄位的聚集方法。
<code>pad</code>	Blank MeanOfRecentPoints True False	指定欄位的填補方法。
<code>agg_mode</code>	All Specify	指定是根據需要使用預設函數聚合或填充所有欄位，還是指定要使用的欄位和函數。
<code>agg_range_default</code>	Mean Sum Mode Min Max	指定聚合連續欄位時要使用的預設函數。

表 94: `timeintervalsnode` 內容 (繼續)

<code>timeintervalsnode</code> 內容	資料類型	內容說明
<code>agg_set_default</code>	Mode First Last	指定聚合列名欄位時要使用的預設函數。
<code>agg_flag_default</code>	TrueIfAnyTrue Mode First Last	
<code>pad_range_default</code>	Blank MeanOfRecentPoints	指定填補連續欄位時要使用的預設函數。
<code>pad_set_default</code>	Blank MostRecentValue	
<code>pad_flag_default</code>	Blank True False	
<code>max_records_to_create</code>	整數	指定填補數列時要建立的上限記錄數。
<code>estimation_from_beginning</code>	旗標	
<code>estimation_to_end</code>	旗標	
<code>estimation_start_offset</code>	整數	
<code>estimation_num_holdouts</code>	整數	
<code>create_future_records</code>	旗標	
<code>num_future_records</code>	整數	
<code>create_future_field</code>	旗標	
<code>future_field_name</code>	<i>string</i>	

transposenode 內容



「轉置」節點會交換列和欄中的資料，以便記錄變成欄位，欄位變成記錄。

範例

```
node = stream.create("transpose", "My node")
node.setPropertyValue("transposed_names", "Read")
node.setPropertyValue("read_from_field", "TimeLabel")
node.setPropertyValue("max_num_fields", "1000")
node.setPropertyValue("id_field_name", "ID")
```

表 95: transposenode 內容		
transposenode 內容	資料類型	內容說明
transpose_method	enum	指定跳至方法: 正常 (normal)、CASE 到 VAR (casetovar) 或 VAR 到 CASE (vartocase)。
transposed_names	Prefix Read	常態跳至方法的內容。可以根據指定字首自動產生新的欄位名稱, 也可以從資料的現有欄位中讀取新的欄位名稱。
prefix	string	常態跳至方法的內容。
num_new_fields	整數	常態跳至方法的內容。使用字首時, 指定要建立的新欄位的最大數量。
read_from_field	欄位	常態跳至方法的內容。從中讀取名稱的欄位。此欄位必須是一個實例化欄位, 否則執行節點時將出錯。
max_num_fields	整數	常態跳至方法的內容。當從某個欄位中讀取名稱時, 指定上限以避免建立過大的欄位個數。
transpose_type	Numeric String Custom	常態跳至方法的內容。依預設, 只能已跳至連續 (數值型範圍) 欄位, 但也可以選擇數值型欄位的自訂子集合或已跳至所有字串欄位。
transpose_fields	清單	常態跳至方法的內容。指定使用 Custom 選項時跳至的欄位。
id_field_name	欄位	常態跳至方法的內容。
transpose_casetovar_id_fields	欄位	CASE 至 VAR (casetovar) 跳至方法的內容。接受要用作指標欄位的多個欄位。 field1 ... fieldN
transpose_casetovar_columnfields	欄位	CASE 至 VAR (casetovar) 跳至方法的內容。接受要用作直欄欄位的多個欄位。 field1 ... fieldN
transpose_casetovar_valuefields	欄位	CASE 至 VAR (casetovar) 跳至方法的內容。接受要用作值欄位的多個欄位。 field1 ... fieldN

表 95: *transposenode* 內容 (繼續)

transposenode 內容	資料類型	內容說明
transpose_vartocase_id fields	欄位	VAR 到 CASE (vartocase) 跳至方法的內容。接受要用作 ID 變數欄位的多個欄位。 field1 ... fieldN
transpose_vartocase_val fields	欄位	VAR 到 CASE (vartocase) 跳至方法的內容。接受要用作值變數欄位的多個欄位。 field1 ... fieldN

typenode 內容



「類型」節點指定欄位 meta 資料和內容。例如，您可以指定每個欄位的測量層次（連續、列名、序數或旗標）、設定用於處理遺漏值和系統空值的選項、設定用於建模的欄位的角色、指定欄位和值標籤，以及為欄位指定值。

範例

```
node = stream.createAt("type", "My node", 50, 50)
node.setKeyedPropertyValue("check", "Cholesterol", "Coerce")
node.setKeyedPropertyValue("direction", "Drug", "Input")
node.setKeyedPropertyValue("type", "K", "Range")
node.setKeyedPropertyValue("values", "Drug", ["drugA", "drugB", "drugC",
"drugD", "drugX",
"drugY", "drugZ"])
node.setKeyedPropertyValue("null_missing", "BP", False)
node.setKeyedPropertyValue("whitespace_missing", "BP", False)
node.setKeyedPropertyValue("description", "BP", "Blood Pressure")
node.setKeyedPropertyValue("value_labels", "BP", [["HIGH", "High Blood
Pressure"],
["NORMAL", "normal blood pressure"]])
```

注意，某些情況下可能需要完全實例化 `type` 節點才能使其他節點正常運作，例如，設為旗標節點的 `fields from` 內容。可以只連接表格節點並執行該節點以實例化這些欄位：

```
tablenode = stream.createAt("table", "Table node", 150, 50)
stream.link(node, tablenode)
tablenode.run(None)
stream.delete(tablenode)
```

表 96: typenode 內容

typenode 內容	資料類型	內容說明
direction	Input Target Both None Partition Split Frequency RecordID	欄位角色的索引內容。 註: 現在已淘汰值 In 和 Out。 在未來版本中可能會置換對它們的支援。
type	Range Flag Set Typeless Discrete OrderedSet Default	欄位的測量層次（先前稱為欄位的「類型」）。 將 type 設定為 Default 會清除所有 values 參數設定，並且如果 value_mode 的值為 Specify，那麼它將重設為 Read。 如果 value_mode 設定為 Pass 或 Read，那麼設定 type 不會影響 value_mode。 註: 內部使用的資料類型與 type 節點中的那些變數不同。 對應關係如下所示: 範圍 -> 連續集合 -> 列名依序集合 -> 依序離散 -> 種類
storage	Unknown String Integer Real Time Date Timestamp	欄位儲存類型的唯讀索引內容。

表 96: *typenode* 內容 (繼續)

typenode 內容	資料類型	內容說明
check	None Nullify Coerce Discard Warn Abort	欄位類型及範圍檢查的索引內容。
values	[value value]	對於連續型欄位而言，第一個是最小值，最後一個是最大值。對於列名欄位，指定所有值。針對旗標欄位，第一個值代表 <i>false</i> ，最後一個值代表 <i>true</i> 。設定該內容將自動把 <i>value_mode</i> 內容設定為 <i>Specify</i> 。
value_mode	Read Pass Read+ Current Specify	確定值的設定方式。注意，不能將此內容直接設定為 <i>Specify</i> ；要使用特定值，需設定 <i>values</i> 內容。
extend_values	旗標	當 <i>value_mode</i> 設定為 <i>Read</i> 時將套用。設定為 <i>T</i> 則將新讀取的值新增到任意現有欄位值。設定為 <i>F</i> 則捨棄現有值並添加新讀取值。
enable_missing	旗標	當設定為 <i>T</i> 時，則啟動對欄位遺漏值的追蹤。
missing_values	[value value...]	指定表示遺漏資料的資料值。
range_missing	旗標	指定是否為欄位定義遺漏值（空白）範圍。
missing_lower	<i>string</i>	當 <i>range_missing</i> 為 <i>true</i> 時，指定遺漏值範圍的下界。
missing_upper	<i>string</i>	當 <i>range_missing</i> 為 <i>true</i> 時，指定遺漏值範圍的上限。
null_missing	旗標	設定為 <i>T</i> 時，無效（在軟體中顯示為 <i>\$null\$</i> 的未定義值）被視為遺漏值。
whitespace_missing	旗標	設定為 <i>T</i> 時，僅包含空白（空格、Tab 和換行符）的值被視為遺漏值。

表 96: *typenode* 內容 (繼續)

typenode 內容	資料類型	內容說明
description	<i>string</i>	為欄位指定說明。
value_labels	<i>[[Value LabelString] [Value LabelString] ...]</i>	用來為值配對指定標籤。
display_places	整數	設定欄位顯示時的小數位數 (僅適用於具有 REAL 儲存體的欄位)。值 -1 表示將使用串流預設值。
export_places	整數	為欄位設定匯出時的小數位數 (僅用於以 REAL 儲存的欄位)。值 -1 表示將使用串流預設值。
decimal_separator	DEFAULT PERIOD COMMA	為欄位設定小數點 (僅用於以 REAL 儲存的欄位)。
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	設定欄位的日期格式 (僅套用至使用 DATE 或 TIMESTAMP 儲存的欄位)。
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	設定欄位的時間格式 (僅套用至使用 TIME 或 TIMESTAMP 儲存的欄位)。

表 96: *typenode* 內容 (繼續)

typenode 內容	資料類型	內容說明
number_format	DEFAULT STANDARD SCIENTIFIC CURRENCY	為欄位設定數字顯示格式。
standard_places	整數	為欄位設定以標準格式顯示時的小數位數。值 -1 表示將使用串流預設值。請注意，現有的 <code>display_places</code> 屬性也會變更此屬性，但目前已不再使用。
scientific_places	整數	為欄位設定以科學計數格式顯示時的小數位數。值 -1 表示將使用串流預設值。
currency_places	整數	為欄位設定以貨幣格式顯示時的小數位數。值 -1 表示將使用串流預設值。
grouping_symbol	DEFAULT NONE LOCALE PERIOD COMMA SPACE	為欄位設定分組符號。
column_width	整數	設定欄位的欄寬。值 -1 會將欄寬設為 Auto。
justify	AUTO CENTER LEFT RIGHT	設定欄位的直欄對齊。

表 96: *typenode* 內容 (繼續)

typenode 內容	資料類型	內容說明
<code>measure_type</code>	Range / MeasureType.RANGE Discrete / MeasureType.DISCRETE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS Collection / MeasureType.COLLECTION Geospatial / MeasureType.GEOSPATIAL	此索引內容類似於 <code>type</code> ，因為可以用來定義與欄位相關聯的測量。不同的是 Python Scripting，還可以為 <code>setter</code> 函數傳遞其中一個 <code>MeasureType</code> 值，而 <code>getter</code> 一律在 <code>MeasureType</code> 值上傳回。
<code>collection_measure</code>	Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS	針對集合欄位 (深度為 0 的清單)，此索引內容會定義與基礎值相關聯的測量類型。
<code>geo_type</code>	Point MultiPoint LineString MultiLineString Polygon MultiPolygon	針對地理空間欄位，此索引內容會定義此欄位代表的地理空間物件類型。這應與值的清單深度一致。
<code>has_coordinate_system</code>	<i>boolean</i>	對於地理空間欄位，此內容會定義此欄位是否具有座標系統
<code>coordinate_system</code>	<i>string</i>	針對地理空間欄位，此索引內容會定義此欄位的座標系統。

表 96: *typenode* 內容 (繼續)

typenode 內容	資料類型	內容說明
custom_storage_type	Unknown / MeasureType.UNKNOWN String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP List / MeasureType.LIST	此索引內容類似於 <i>custom_storage</i> ，因為可以用來定義欄位的置換儲存。不同的是 Python Scripting，還可以為 <i>setter</i> 函數傳遞其中一個 <i>StorageType</i> 值，而 <i>getter</i> 一律在 <i>StorageType</i> 值上傳回。
custom_list_storage_type	String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP	針對清單欄位，此索引內容會指定基礎值的儲存類型。
custom_list_depth	整數	針對清單欄位，此索引內容會指定欄位的深度
max_list_length	整數	僅可用於測量層級為地理空間或集合的資料。透過指定清單可以包含的元素數目來設定清單的長度上限。
max_string_length	整數	僅可用於無類型資料，且僅可在產生 SQL 來建立表格時使用。輸入資料中最大字串的值；這會產生表格中大小足以包含字串的直欄。

第 12 章 圖形節點內容

圖形節點一般內容

本節介紹圖形節點的可用內容，包括通用內容和每種節點類型特有的內容。

一般圖形節點內容	資料類型	內容說明
title	string	指定標題。 範例: "This is a title."
caption	string	指定標題。 範例: "This is a caption."
output_mode	Screen File	指定是顯示圖形節點的輸出還是將其寫入到檔案中。
output_format	BMP JPEG PNG HTML output (.cou)	指定輸出的類型。 容許每個節點使用的確切輸出類型是不同的。
full_filename	string	為從圖形節點產生的輸出指定目標路徑和檔名。
use_graph_size	旗標	控制項是否使用下方的寬度和高度內容明確調整圖形的大小。 只影響輸出到畫面的圖形。 不適用於配送節點。
graph_width	數字	當 use_graph_size 為 True 時，以像素為單位設定圖形寬度。
graph_height	數字	當 use_graph_size 為 True 時，以像素為單位設定圖形高度。

關閉選用欄位

通過將內容值設定為 " " (空字串)，可以將選用欄位 (如繪圖的套版欄位) 關閉，如以下範例所示：

```
plotnode.setPropertyValue("color_field", "")
```

指定顏色

使用十六進位字串 (從井字 (#) 開始)，可指定標題、標註、背景和標籤的顏色。 例如，要將圖形背景設定為天藍色，請使用下列陳述式：

```
mygraphnode.setPropertyValue("graph_background", "#87CEEB")
```

此處，前兩位數 87 指定紅色內容；中間兩位數 CE 指定綠色內容；最後一欄兩位數 EB 指定藍色內容。 每一個數字可以採用 0-9 或 A-F 範圍內的值。 這些值可以一起指定紅色-綠色-藍色或 RGB 顏色。

註: 以 RGB 形式指定顏色時, 可以使用使用者介面中的欄位選擇器確定正確的顏色代碼。只需將滑鼠停留在顏色上面就可啟動含所需資訊的工具提示。

collectionnode 內容



「收集」節點顯示一個數值欄位的值相對於另一個數值欄位的值的分配。(它建立類似於直方圖的圖形。) 它可用來說明值隨時間不斷變化的變數或欄位。使用 3-D 繪圖, 您還可以包括符號軸, 用來依種類顯示分佈項目。

範例

```
node = stream.create("collection", "My node")
# "Plot" tab
node.setPropertyValue("three_D", True)
node.setPropertyValue("collect_field", "Drug")
node.setPropertyValue("over_field", "Age")
node.setPropertyValue("by_field", "BP")
node.setPropertyValue("operation", "Sum")
# "Overlay" section
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "")
# "Options" tab
node.setPropertyValue("range_mode", "Automatic")
node.setPropertyValue("range_min", 1)
node.setPropertyValue("range_max", 100)
node.setPropertyValue("bins", "ByNumber")
node.setPropertyValue("num_bins", 10)
node.setPropertyValue("bin_width", 5)
```

表 98: collectionnode 內容

collectionnode 內容	資料類型	內容說明
over_field	欄位	
over_label_auto	旗標	
over_label	string	
collect_field	欄位	
collect_label_auto	旗標	
collect_label	string	
three_D	旗標	
by_field	欄位	
by_label_auto	旗標	
by_label	string	

表 98: collectionnode 內容 (繼續)

collectionnode 內容	資料類型	內容說明
operation	Sum Mean Min Max SDev	
color_field	string	
panel_field	string	
animation_field	string	
range_mode	Automatic UserDefined	
range_min	數字	
range_max	數字	
bins	ByNumber ByWidth	
num_bins	數字	
bin_width	數字	
use_grid	旗標	
graph_background	顏色	標準圖形顏色在此小節的開頭進行說明。
page_background	顏色	標準圖形顏色在此小節的開頭進行說明。

distributionnode 內容



「分佈」節點顯示符號（種類）值（例如抵押類型或性別）的出現次數。通常，可以使用「分佈」節點來顯示資料中的不平衡，然後在建立模型前使用「平衡」節點來糾正此類不平衡。

範例

```
node = stream.create("distribution", "My node")
# "Plot" tab
node.setPropertyValue("plot", "Flags")
node.setPropertyValue("x_field", "Age")
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("normalize", True)
node.setPropertyValue("sort_mode", "ByOccurence")
node.setPropertyValue("use_proportional_scale", True)
```

表 99: *distributionnode* 內容

distributionnode 內容	資料類型	內容說明
plot	SelectedFields Flags	
x_field	欄位	
color_field	欄位	套版欄位。
normalize	旗標	
sort_mode	ByOccurence Alphabetic	
use_proportional_scale	旗標	

evaluationnode 內容



「評估」節點有助於評估和比較預測模型。評估圖顯示了模型對特定結果的預測程度。它根據預測值和預測信賴度來對記錄進行排序。它將記錄分成若干個相同大小的群組（分位數），然後從高到底為每個分位數繪製業務準則值。多個模型在圖形中顯示為個別的行。

範例

```
node = stream.create("evaluation", "My node")
# "Plot" tab
node.setPropertyValue("chart_type", "Gains")
node.setPropertyValue("cumulative", False)
node.setPropertyValue("field_detection_method", "Name")
node.setPropertyValue("inc_baseline", True)
node.setPropertyValue("n_tile", "Deciles")
node.setPropertyValue("style", "Point")
node.setPropertyValue("point_type", "Dot")
node.setPropertyValue("use_fixed_cost", True)
node.setPropertyValue("cost_value", 5.0)
node.setPropertyValue("cost_field", "Na")
node.setPropertyValue("use_fixed_revenue", True)
node.setPropertyValue("revenue_value", 30.0)
node.setPropertyValue("revenue_field", "Age")
node.setPropertyValue("use_fixed_weight", True)
node.setPropertyValue("weight_value", 2.0)
node.setPropertyValue("weight_field", "K")
```

表 100: *evaluationnode* 內容

evaluationnode 內容	資料類型	內容說明
chart_type	Gains Response Lift Profit ROI ROC	
inc_baseline	旗標	

表 100: evaluationnode 內容 (繼續)

evaluationnode 內容	資料類型	內容說明
field_detection_method	Metadata Name	
use_fixed_cost	旗標	
cost_value	數字	
cost_field	string	
use_fixed_revenue	旗標	
revenue_value	數字	
revenue_field	string	
use_fixed_weight	旗標	
weight_value	數字	
weight_field	欄位	
n_tile	Quartiles Quintiles Deciles Vingtiles Percentiles 1000-tiles	
cumulative	旗標	
style	Line Point	
point_type	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	

表 100: *evaluationnode* 內容 (繼續)

evaluationnode 內容	資料類型	內容說明
export_data	旗標	
data_filename	<i>string</i>	
delimiter	<i>string</i>	
new_line	旗標	
inc_field_names	旗標	
inc_best_line	旗標	
inc_business_rule	旗標	
business_rule_condition	<i>string</i>	
plot_score_fields	旗標	
score_fields	<i>[field1 ... fieldn]</i>	
target_field	欄位	
use_hit_condition	旗標	
hit_condition	<i>string</i>	
use_score_expression	旗標	
score_expression	<i>string</i>	
caption_auto	旗標	

graphboardnode 內容



「圖形板」節點可在一個節點中提供多數不同類型的圖形。使用此節點，可以選擇要探索的資料欄位，然後從適用於選定資料的欄位中選擇一個圖形。該節點會自動濾出不適用於欄位選項的任何圖形類型。

註: 如果您設定對圖形類型無效的內容 (例如，為直方圖指定 *y_field*)，該內容將被忽略。

註: 在 UI 中多數不同圖形類型的「詳細」標籤上，有一個摘要欄位；Scripting 目前不支援此欄位。

範例

```
node = stream.create("graphboard", "My node")
node.setPropertyValue("graph_type", "Line")
node.setPropertyValue("x_field", "K")
node.setPropertyValue("y_field", "Na")
```


表 101: graphboardnode 內容

graphboard 內容	資料類型	內容說明
graph_type	2DDotplot 3DArea 3DBar 3DDensity 3DHistogram 3DPie 3DScatterplot Area ArrowMap Bar BarCounts BarCountsMap BarMap BinnedScatter Boxplot Bubble ChoroplethMeans ChoroplethMedians ChoroplethSums ChoroplethValues	識別圖形類型。

表 101: graphboardnode 內容 (繼續)

graphboard 內容	資料類型	內容說明
	ChoroplethCounts	
	CoordinateMap	
	CoordinateChoroplethMeans	
	CoordinateChoroplethMedians	
	CoordinateChoroplethSums	
	CoordinateChoroplethValues	
	CoordinateChoroplethCounts	
	Dotplot	
	Heatmap	
	HexBinScatter	
	Histogram	
	Line	
	LineChartMap	
	LineOverlayMap	
	Parallel	
	Path	
	Pie	
	PieCountMap	
	PieCounts	
	PieMap	

表 101: graphboardnode 內容 (繼續)

graphboard 內容	資料類型	內容說明
	PointOverlayMap PolygonOverlayMap Ribbon Scatterplot SPLOM Surface	
x_field	欄位	為 x 軸指定自訂標籤。僅可用於標籤。
y_field	欄位	為 y 軸指定自訂標籤。僅可用於標籤。
z_field	欄位	在某些三維圖中使用。
color_field	欄位	在熱圖中使用。
size_field	欄位	在氣泡統計圖中使用。
categories_field	欄位	
values_field	欄位	
rows_field	欄位	
columns_field	欄位	
fields	欄位	
start_longitude_field	欄位	與參照圖中的箭頭配合使用。
end_longitude_field	欄位	
start_latitude_field	欄位	
end_latitude_field	欄位	
data_key_field	欄位	用於各種圖。
panelrow_field	string	
panelcol_field	string	
animation_field	string	
longitude_field	欄位	與地圖中的座標配合使用。
latitude_field	欄位	
map_color_field	欄位	

histogramnode 內容



「直方圖」節點顯示了數值型欄位的值的出現次數。它經常用來在資料操作和模型建置之前探索資料。與「分佈」節點相似，「直方圖」節點經常用來揭示資料中的不平衡度。

範例

```
node = stream.create("histogram", "My node")
# "Plot" tab
node.setPropertyValue("field", "Drug")
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "")
# "Options" tab
node.setPropertyValue("range_mode", "Automatic")
node.setPropertyValue("range_min", 1.0)
node.setPropertyValue("range_max", 100.0)
node.setPropertyValue("num_bins", 10)
node.setPropertyValue("bin_width", 10)
node.setPropertyValue("normalize", True)
node.setPropertyValue("separate_bands", False)
```

表 102: histogramnode 內容

histogramnode 內容	資料類型	內容說明
field	欄位	
color_field	欄位	
panel_field	欄位	
animation_field	欄位	
range_mode	Automatic UserDefined	
range_min	數字	
range_max	數字	
bins	ByNumber ByWidth	
num_bins	數字	
bin_width	數字	
normalize	旗標	
separate_bands	旗標	
x_label_auto	旗標	
x_label	string	
y_label_auto	旗標	
y_label	string	
use_grid	旗標	

表 102: histogramnode 內容 (繼續)

histogramnode 內容	資料類型	內容說明
graph_background	顏色	標準圖形顏色在此小節的開頭進行說明。
page_background	顏色	標準圖形顏色在此小節的開頭進行說明。
normal_curve	旗標	指出是否應在輸出中顯示常態分佈曲線。

mapvisualization 內容



「地圖視覺化」節點可以接受多個輸入連線並在地圖上將地理空間資料顯示為一系列層。每一層都是單一地理空間欄位；例如，基本層可能是某個國家或地區的地圖，而在該層之上，可能有一層代表道路，一層代表河流，以及一層代表城鎮。

表 103: mapvisualization 內容

mapvisualization 內容	資料類型	內容說明
tag	string	設定輸入的標籤名稱。預設標籤是一個數字，基於輸入連接至節點的順序（第一個連線標籤是 1，第二個連線標籤是 2，以此類推）
layer_field	欄位	<p>選取將資料集中的哪個地理空間欄位顯示為地圖上的層。預設選擇基於下列排序：</p> <ul style="list-style-type: none"> • 第一個 - 點 • Linestring • 多邊形 • 多點 • MultiLinestring • 最後一個 - 多個多邊形 <p>如果有兩個欄位具有相同測量類型，依預設將選取依名稱按字母順序排列的第一個欄位。</p>
color_type	boolean	指定是對所有地理空間欄位特徵套用標準顏色，還是選擇套版欄位，這樣會基於資料集中另一個欄位的值對特徵著色。可能的值是 standard 或 overlay。預設值是 standard。
color	string	<p>如果為 color_type 選取了 standard，那麼下拉清單將包含與使用者選項「顯示」標籤上的圖表種類顏色順序相同的調色盤。</p> <p>預設值是圖表種類顏色 1。</p>
color_field	欄位	如果為 color_type 選取了 overlay，那麼下拉清單將包含與選作層的地理空間欄位相同資料集中的所有欄位。

表 103: mapvisualization 內容 (繼續)

mapvisualization 內容	資料類型	內容說明
symbol_type	boolean	指定是對所有地理空間欄位記錄套用標準符號，還是選擇重疊符號，這樣會基於資料集中另一欄位的值變更點的符號圖示。可能的值是 standard 或 overlay。預設值是 standard。
symbol	string	如果為 symbol_type 選擇了 standard，那麼下拉清單將包含可以用於顯示地圖上的點的符號選擇。
symbol_field	欄位	如果為 symbol_type 選取了 overlay，那麼下拉清單將包含與選作層的地理空間欄位相同資料集中的所有列名、序數或種類欄位。
size_type	boolean	指定是對所有地理空間欄位記錄套用標準大小，還是選擇套版大小，這樣會基於資料集中另一欄位的值變更符號圖示的大小或線條粗細。可能的值是 standard 或 overlay。預設值是 standard。
size	string	如果為 size_type 選擇了 standard，那麼對於 point 或 multipoint，下拉清單將包含適用於所選符號的大小選擇。對於 linestring 或 multilinestring，該下拉清單將包含線條粗細選擇。
size_field	欄位	如果選取了 overlay 作為 size_type，則下拉清單中會包含來自選定為層之地理空間欄位所在資料集中的所有欄位。
transp_type	boolean	指定是對所有地理空間欄位記錄套用標準透明，還是選擇套版透明，這樣會基於資料集中另一欄位的值變更符號、線條或多邊形的透明度。可能的值是 standard 或 overlay。預設值是 standard。
transp	整數	<p>如果為 transp_type 選擇了 standard，那麼下拉清單將包含透明度選項，從 0%（不透明）增加到 100%（透明），增量為 10%。設定地圖上的點、線條或多邊形的透明度。</p> <p>如果選取了 overlay 作為 size_type，則下拉清單中會包含來自選定為層之地理空間欄位所在資料集中的所有欄位。</p> <p>對於作為底層的 points、multipoints、linestrings 和 multilinestrings、polygons 和 multipolygons，預設值是 0%。對於不是底層的 polygons 和 multipolygons，預設值為 50%（以避免遮蔽這些多邊形下方的層）。</p>

表 103: mapvisualization 內容 (繼續)

mapvisualization 內容	資料類型	內容說明
transp_field	欄位	如果選取了 overlay 作為 transp_type, 則下拉清單中會包含來自選定為層之地理空間欄位所在資料集中的所有欄位。
data_label_field	欄位	指定要用作地圖上的資料標籤的欄位。例如, 如果對其應用該設定的層是多邊形層, 那麼資料標籤可能是 name 欄位 - 包含每個多邊形的名稱。因此在此選取 name 欄位可能會導致在地圖上顯示這些名稱。
use_hex_binning	boolean	啟用十六進位分組並啟用所有聚集下拉清單。預設情況下, 該設定處於關閉狀態。
color_aggregation 和 transp_aggregation	string	<p>如果為使用十六進位分組的點層選取了套版欄位, 那麼該欄位的所有值都必須針對該六邊形內的所有點進行聚合。因此, 必須為任何要套用至地圖的套版欄位指定總計函數。</p> <p>可用的總計函數為:</p> <p>連續 (實數或整數儲存):</p> <ul style="list-style-type: none"> • 加總 • 平均數 • 最小值 • 最大值 • 中位數 • 第 1 個四分位數 • 第 3 個四分位數 <p>連續 (時間、日期或時間戳記儲存):</p> <ul style="list-style-type: none"> • 平均數 • 最小值 • 最大值 <p>列名/種類:</p> <ul style="list-style-type: none"> • 模式 • 最小值 • 最大值 <p>旗標:</p> <ul style="list-style-type: none"> • 如果任意一項為 true 則為 true • 如果任意一項為 false 則為 false
custom_storage	string	設定欄位的整體儲存類型。預設值為 List。如果指定了 List, 那麼將停用下列 custom_value_storage 和 list_depth 控制項。
custom_value_storage	string	設定清單中的元素的儲存類型, 而不是設定為欄位的整體儲存類型。預設值為 Real。

表 103: mapvisualization 內容 (繼續)

mapvisualization 內容	資料類型	內容說明
list_depth	整數	<p>設定清單欄位的深度。所需深度取決於地理空間欄位類型，遵循下列準則：</p> <ul style="list-style-type: none"> • 點 - 0 • 線串 - 1 • 多邊形 - 2 • 多點 - 1 • 多線串 - 2 • 多個多邊形 - 3 <p>您必須瞭解要重新轉換為清單的地理空間欄位的類型以及該類欄位的所需深度。如果設定不正確，則不能使用該欄位。</p> <p>預設值為 0，最小值為 0，最大值為 10。</p>

multiplotnode 內容



使用多重繪圖節點可建立在一個 X 欄位上顯示多個 Y 欄位的統計圖。Y 欄位被繪製為彩色的行；每條行相當於「樣式」設定為行且「X 模式」設定為排序的「繪圖」節點。要研究幾個變數隨時間的變化情況時，多重繪圖非常有用。

範例

```
node = stream.create("multiplot", "My node")
# "Plot" tab
node.setPropertyValue("x_field", "Age")
node.setPropertyValue("y_fields", ["Drug", "BP"])
node.setPropertyValue("panel_field", "Sex")
# "Overlay" section
node.setPropertyValue("animation_field", "")
node.setPropertyValue("tooltip", "test")
node.setPropertyValue("normalize", True)
node.setPropertyValue("use_overlay_expr", False)
node.setPropertyValue("overlay_expression", "test")
node.setPropertyValue("records_limit", 500)
node.setPropertyValue("if_over_limit", "PlotSample")
```

表 104: multiplotnode 內容

multiplotnode 內容	資料類型	內容說明
x_field	欄位	
y_fields	清單	
panel_field	欄位	
animation_field	欄位	
normalize	旗標	
use_overlay_expr	旗標	
overlay_expression	string	
records_limit	數字	

表 104: *multiplotnode* 內容 (繼續)

multiplotnode 內容	資料類型	內容說明
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	旗標	
x_label	<i>string</i>	
y_label_auto	旗標	
y_label	<i>string</i>	
use_grid	旗標	
graph_background	顏色	標準圖形顏色在此小節的開頭進行說明。
page_background	顏色	標準圖形顏色在此小節的開頭進行說明。

plotnode 內容



「繪圖」節點可顯示數值型欄位間的關係。可透過使用點（散佈平面圖）或線建立繪圖。

範例

```
node = stream.create("plot", "My node")
# "Plot" tab
node.setPropertyValue("three_D", True)
node.setPropertyValue("x_field", "BP")
node.setPropertyValue("y_field", "Cholesterol")
node.setPropertyValue("z_field", "Drug")
# "Overlay" section
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("size_field", "Age")
node.setPropertyValue("shape_field", "")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "BP")
node.setPropertyValue("transp_field", "")
node.setPropertyValue("style", "Point")
# "Output" tab
node.setPropertyValue("output_mode", "File")
node.setPropertyValue("output_format", "JPEG")
node.setPropertyValue("full_filename", "C:/temp/graph_output/
plot_output.jpeg")
```

表 105: *plotnode* 內容

plotnode 內容	資料類型	內容說明
x_field	欄位	為 x 軸指定自訂標籤。僅可用於標籤。
y_field	欄位	為 y 軸指定自訂標籤。僅可用於標籤。
three_D	旗標	為 y 軸指定自訂標籤。只適用於 3 維圖形中的標籤。

表 105: plotnode 內容 (繼續)

plotnode 內容	資料類型	內容說明
z_field	欄位	
color_field	欄位	套版欄位。
size_field	欄位	
shape_field	欄位	
panel_field	欄位	指定用於為每個種類繪製個別圖表的列名欄位或旗標欄位。圖表一起平鋪在一個輸出視窗中。
animation_field	欄位	指定以圖說明資料值種類（通過使用動畫建立一系列按順序顯示的圖表來說明）時所使用的列名欄位或旗標欄位。
transp_field	欄位	指定一個欄位，用來為每個種類使用不同的透明度來圖解資料值種類。不適用於線條圖。
overlay_type	None Smoother Function	指定是顯示套版函數還是 LOESS 校平器。
overlay_expression	string	指定當 overlay_type 設定為 Function 時使用的表示式。
style	Point Line	
point_type	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
x_mode	Sort Overlay AsRead	

表 105: *plotnode* 內容 (繼續)

plotnode 內容	資料類型	內容說明
x_range_mode	Automatic UserDefined	
x_range_min	數字	
x_range_max	數字	
y_range_mode	Automatic UserDefined	
y_range_min	數字	
y_range_max	數字	
z_range_mode	Automatic UserDefined	
z_range_min	數字	
z_range_max	數字	
jitter	旗標	
records_limit	數字	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	旗標	
x_label	<i>string</i>	
y_label_auto	旗標	
y_label	<i>string</i>	
z_label_auto	旗標	
z_label	<i>string</i>	
use_grid	旗標	
graph_background	顏色	標準圖形顏色在此小節的開頭進行說明。
page_background	顏色	標準圖形顏色在此小節的開頭進行說明。
use_overlay_expr	旗標	該屬性已由 <i>overlay_type</i> 替代。

timeplotnode 內容



「時間繪圖」節點顯示一個或多個時間序列資料集。通常，您將首先使用「時間間隔」節點建立 *TimeLabel* 欄位，該欄位將用於標記 x 軸。

範例

```
node = stream.create("timeplot", "My node")
node.setPropertyValue("y_fields", ["sales", "men", "women"])
node.setPropertyValue("panel", True)
node.setPropertyValue("normalize", True)
node.setPropertyValue("line", True)
node.setPropertyValue("smoother", True)
node.setPropertyValue("use_records_limit", True)
node.setPropertyValue("records_limit", 2000)
# Appearance settings
node.setPropertyValue("symbol_size", 2.0)
```

表 106: timeplotnode 內容

timeplotnode 內容	資料類型	內容說明
plot_series	Series Models	
use_custom_x_field	旗標	
x_field	欄位	
y_fields	清單	
panel	旗標	
normalize	旗標	
line	旗標	
points	旗標	
point_type	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
smoother	旗標	只有將 panel 設定為 True，才可將校平器新增到圖形中。
use_records_limit	旗標	
records_limit	整數	
symbol_size	數字	指定符號大小。
panel_layout	Horizontal Vertical	

eplotnode 內容



E-Plot (測試版) 節點顯示數值欄位之間的關係。它與「繪圖」節點類似，但選項不同，且其輸出使用特定於此節點的新圖形繪製介面。使用測試版層次節點來與新圖形繪製功能搭配使用。

表 107: eplotnode 內容

eplotnode 內容	資料類型	內容說明
x_field	string	指定要在水平 X 軸上顯示的欄位。
y_field	string	指定要在垂直 Y 軸上顯示的欄位。
color_field	string	指定要在輸出中用於色彩對映套版的欄位 (如果需要)。
size_field	string	指定要在輸出中用於大小對映套版的欄位 (如果需要)。
shape_field	string	指定要在輸出中用於形狀對映套版的欄位 (如果需要)。
interested_fields	string	指定您要在輸出中包括的欄位。
records_limit	整數	指定要在輸出中繪製的記錄數目上限。預設值是 2000。
if_over_limit	布林	指定是否要在超過 records_limit 時使用 Sample 選項或 Use all data 選項。預設值是 Sample，它會對資料進行隨機取樣，直至達到 records_limit。如果您指定 Use all data 以忽略 records_limit 並繪製所有資料點，請注意這可能會大幅降低效能。

tsnnode 內容



t-Distributed Stochastic Neighbor Embedding (t-SNE) 是用於視覺化高維度資料的工具。它會將資料點的親緣性轉換為機率。SPSS Modeler 中的這個 t-SNE 節點使用 Python 進行實作，並且需要 scikit-learn® Python 程式庫。

表 108: tsnnode 內容

tsnnode 內容	資料類型	內容說明
mode_type	string	指定 simple 或 expert 模式。
n_components	string	內嵌空間的維度 (2D 或 3D)。指定 2 或 3。預設值是 2。
method	string	指定 barnes_hut 或 exact。預設值是 barnes_hut。
init	string	內嵌的起始設定。指定 random 或 pca。預設值是 random。

表 108: tsnode 內容 (繼續)

tsnode 內容	資料類型	內容說明
target_field 從 18.2.1.1 版開始重新命名為 target	string	目標欄位名稱。它將是輸出圖形上的顏色分配表。如果未指定目標欄位，則該圖形會使用一種顏色。
perplexity	Float	混雜度與其他多重學習演算法中使用的最近鄰接項數相關。更大的資料集通常需要更大的混雜度。考量選取介於 5 和 50 之間的值。預設值是 30。
early_exaggeration	Float	控制原始空間中的自然叢集在內嵌空間中的緊密程度，以及叢集之間的空間量。預設值是 12.0。
learning_rate	Float	預設值是 200。
n_iter	整數	用於最佳化的最大疊代數。設為至少 250。預設值為 1000。
angle	Float	從點測量的遠距節點的角度大小。指定 0-1 範圍中的值。預設值是 0.5。
enable_random_seed	布林	設為 true 以啟用 random_seed 參數。預設值是 false。
random_seed	整數	要使用的亂數種子。預設值是 None。
n_iter_without_progress	整數	沒有進度的最大疊代數。預設值是 300。
min_grad_norm	string	如果梯度標準低於此臨界值，則會停止最佳化。預設值是 1.0E-7。可能的值為： <ul style="list-style-type: none"> • 1.0E-1 • 1.0E-2 • 1.0E-3 • 1.0E-4 • 1.0E-5 • 1.0E-6 • 1.0E-7 • 1.0E-8
isGridSearch	布林	設為 true 以使用數個不同的混雜度執行 t-SNE。預設值為 false。
output_Rename	布林	如果您要提供自訂名稱，請指定 true，或者指定 false 來自動命名輸出。預設值是 false。
output_to	string	指定 Screen 或 Output。預設值是 Screen。
full_filename	string	指定輸出檔名稱。
output_file_type	string	輸出檔格式。指定 HTML 或 Output object。預設值是 HTML。

webnode 內容



Web 節點說明兩個或多個符號（種類）欄位值之間關係的強度。該圖使用不同粗細的線來表示連接強度。例如，您可以使用 Web 節點來探索在電子商務網站購買一組項目之間的關係。

範例

```
node = stream.create("web", "My node")
# "Plot" tab
node.setPropertyValue("use_directed_web", True)
node.setPropertyValue("to_field", "Drug")
node.setPropertyValue("fields", ["BP", "Cholesterol", "Sex", "Drug"])
node.setPropertyValue("from_fields", ["BP", "Cholesterol", "Sex"])
node.setPropertyValue("true_flags_only", False)
node.setPropertyValue("line_values", "Absolute")
node.setPropertyValue("strong_links_heavier", True)
# "Options" tab
node.setPropertyValue("max_num_links", 300)
node.setPropertyValue("links_above", 10)
node.setPropertyValue("num_links", "ShowAll")
node.setPropertyValue("discard_links_min", True)
node.setPropertyValue("links_min_records", 5)
node.setPropertyValue("discard_links_max", True)
node.setPropertyValue("weak_below", 10)
node.setPropertyValue("strong_above", 19)
node.setPropertyValue("link_size_continuous", True)
node.setPropertyValue("web_display", "Circular")
```

表 109: webnode 內容

webnode 內容	資料類型	內容說明
use_directed_web	旗標	
fields	清單	
to_field	欄位	
from_fields	清單	
true_flags_only	旗標	
line_values	Absolute OverallPct PctLarger PctSmaller	
strong_links_heavier	旗標	
num_links	ShowMaximum ShowLinksAbove ShowAll	
max_num_links	數字	

表 109: webnode 內容 (繼續)

webnode 內容	資料類型	內容說明
links_above	數字	
discard_links_min	旗標	
links_min_records	數字	
discard_links_max	旗標	
links_max_records	數字	
weak_below	數字	
strong_above	數字	
link_size_continuous	旗標	
web_display	Circular Network Directed Grid	
graph_background	顏色	標準圖形顏色在此小節的開頭進行說明。
symbol_size	數字	指定符號大小。

第 13 章 建模節點內容

一般建模節點內容

下列內容通用於某些或所有建模節點。所有例外情況均根據需要記錄在各個建模節點的文件中。

內容	值	內容說明
custom_fields	旗標	如果為 true，則容許您為現行節點指定目標、輸入及其他欄位。如果為 false，則會使用上游「類型」節點中的現行設定。
target 或 targets	欄位 或 [field1 ... fieldN]	根據模型類型指定一個目標欄位或多個目標欄位。
inputs	[field1 ... fieldN]	模型所用的輸入或預測值欄位。
partition	欄位	
use_partitioned_data	旗標	如果已定義分割區欄位，則此選項可確保只使用訓練分割區中的資料來建置模型。
use_split_data	旗標	
splits	[field1 ... fieldn]	指定用於分割建模的一或多個欄位。僅當 use_split_data 設定為 True 時才有效。
use_frequency	旗標	特定模型所使用的加權和頻率欄位（參見每種模型類型的說明）。
frequency_field	欄位	
use_weight	旗標	
weight_field	欄位	
use_model_name	旗標	
model_name	string	自訂新模型的名稱。
mode	Simple Expert	

anomalydetectionnode 內容



異常偵測節點識別不符合「正常」資料型樣的異常觀察值或離群值。使用此節點，即使偏離值不符合任何先前已知的型樣，也可以識別偏離值，即使您未確切確定要尋找的項目也是如此。

範例

```
node = stream.create("anomalydetection", "My node")
node.setPropertyValue("anomaly_method", "PerRecords")
node.setPropertyValue("percent_records", 95)
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("peer_group_num_auto", True)
node.setPropertyValue("min_num_peer_groups", 3)
node.setPropertyValue("max_num_peer_groups", 10)
```

表 111: anomalydetectionnode 內容

anomalydetectionnode 內容	值	內容說明
inputs	[field1 ... fieldn]	異常偵測模型將根據指定的輸入欄位篩選記錄。它們不使用目標欄位。也不使用加權和頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
mode	Expert Simple	
anomaly_method	IndexLevel PerRecords NumRecords	指定用於確定將記錄旗標為異常的分割值的方法。
index_level	數字	指定用來標示例外狀況的最小截斷值。
percent_records	數字	根據訓練資料中的記錄百分比設定用於旗標記錄的臨界值。
num_records	數字	根據訓練資料中的記錄數設定用於旗標記錄的臨界值。
num_fields	整數	要對每條異常記錄報告的欄位數。
impute_missing_values	旗標	
adjustment_coeff	數字	此值用於平衡計算距離時指定給連續欄位和種類欄位的相對加權。
peer_group_num_auto	旗標	自動計算對等群組數目。
min_num_peer_groups	整數	指定 peer_group_num_auto 設定為 True 時使用的對等群組的下限號碼。
max_num_per_groups	整數	指定對等群組的最大數量。
num_peer_groups	整數	指定 peer_group_num_auto 設定為 False 時使用的對等群組數目。
noise_level	數字	確定叢集期間離群值的處理方式。指定介於 0 到 0.5 之間的值。
noise_ratio	數字	指定為元件配置的應用於噪音緩衝的記憶體部分。指定介於 0 到 0.5 之間的值。

apriorinode 內容



「事前」節點從資料擷取一組規則，即擷取資訊內容最多的規則。Apriori 節點提供五種選取規則的方法並使用複雜的編製索引模式來高效地處理大資料集。對於較大的問題，Apriori 訓練的速度通常較快；它對可保留的規則數目沒有任何限制，而且可處理最多帶有 32 個前提條件的規則。Apriori 要求輸入和輸出欄位均為種類欄位，但因為它專為處理此類型資料而進行最佳化，因而處理速度快得多。

範例

```
node = stream.create("apriori", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("partition", "Test")
# For non-transactional
node.setPropertyValue("use_transactional_data", False)
node.setPropertyValue("consequents", ["Age"])
node.setPropertyValue("antecedents", ["BP", "Cholesterol", "Drug"])
# For transactional
node.setPropertyValue("use_transactional_data", True)
node.setPropertyValue("id_field", "Age")
node.setPropertyValue("contiguous", True)
node.setPropertyValue("content_field", "Drug")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Apriori_bp_choles_drug")
node.setPropertyValue("min_supp", 7.0)
node.setPropertyValue("min_conf", 30.0)
node.setPropertyValue("max_antecedents", 7)
node.setPropertyValue("true_flags", False)
node.setPropertyValue("optimize", "Memory")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("evaluation", "ConfidenceRatio")
node.setPropertyValue("lower_bound", 7)
```

表 112: apriorinode 內容

apriorinode 內容	值	內容說明
consequents	欄位	Apriori 模型使用「結果」和「條件」代替標準目標和輸入欄位。不會使用加權和頻率欄位。請參閱第 191 頁的『 一般建模節點內容 』主題，以取得更多資訊。
antecedents	[field1 ... fieldN]	
min_supp	數字	
min_conf	數字	
max_antecedents	數字	
true_flags	旗標	
optimize	Speed Memory	
use_transactional_data	旗標	當值為 true 時，每一筆交易 ID 的評分都與其他交易 ID 無關。當要評分的資料太大而無法取得可接受的效能時，我們建議區隔資料。

表 112: *apriorinode* 內容 (繼續)

apriorinode 內容	值	內容說明
contiguous	旗標	
id_field	string	
content_field	string	
mode	Simple Expert	
evaluation	RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare	
lower_bound	數字	
optimize	Speed Memory	用來指定是否應該最佳化模型建置以提高速度或增加記憶體。

associationrulesnode 內容



「相關規則」節點與 Apriori 節點類似；但是，與 Apriori 不同，「相關規則」節點能夠處理清單資料。另外，「關聯規則」節點可以與 IBM SPSS Analytic Server 配合使用，以處理大型資料以及利用更快的平行處理功能。

表 113: *associationrulesnode* 內容

associationrulesnode 內容	資料類型	內容說明
predictions	欄位	此清單中的欄位只能顯示為規則的預測值
conditions	[field1...fieldN]	此清單中的欄位只能顯示為規則的條件
max_rule_conditions	整數	單條規則中可以包含的上限條件數。最小值為 1，最大值為 9。
max_rule_predictions	整數	單條規則中可以包含的預測的最大數量。最小值為 1，最大值為 5。
max_num_rules	整數	在規則建置過程中可以考慮的最大規則數目。最小值為 1，最大值為 10,000。

表 113: associationrulesnode 內容 (繼續)

associationrulesnode 內容	資料類型	內容說明
rule_criterion_top_n	Confidence Rulesupport Lift Conditionsupport Deployability	規則準則，此準則確定一個值，依據此值選擇模型中的前 "N" 條規則。
true_flags	布林	設定為 Y 確定規則建置期間僅考慮旗標欄位的 true 值。
rule_criterion	布林	設定為 Y 確定模型建置期間使用規則準則值來排除規則。
min_confidence	數字	0.1 到 100 - 模型所產生規則的下限必要信任層次的百分比值。如果模型產生了信賴度少於此處指定的值的規則，那麼將捨棄該規則。
min_rule_support	數字	0.1 到 100 - 模型所產生規則的下限必要規則支援度的百分比值。如果模型產生了規則支援度少於指定值的規則，那麼將捨棄該規則。
min_condition_support	數字	0.1 到 100 - 模型所產生規則的下限必要條件支援度的百分比值。如果模型產生了條件支援度少於指定值的規則，那麼將捨棄該規則。
min_lift	整數	1 到 10 - 代表模型所產生規則的下限必要提升。如果模型產生了提升層次少於指定值的規則，那麼將捨棄該規則。
exclude_rules	布林	用於選取您不希望模型在建立規則時使用的一系列相關欄位。 範例: set :gsarsnode.exclude_rules = [[field1,field2, field3],[field4, field5]] - 其中以 [] 區隔的每一個欄位清單是表格中的一列。
num_bins	整數	設定用於對連續欄位進行「分組」的自動「分組」數。最小值為 2，最大值為 10。
max_list_length	整數	套用至最大長度未知的任何清單欄位。清單中此處所指定數字之前的元素將包含在模型建立中；任何後續元素都將被捨棄。最小為 1，最大為 100。
output_confidence	布林	
output_rule_support	布林	
output_lift	布林	

表 113: associationrulesnode 內容 (繼續)

associationrulesnode 內容	資料類型	內容說明
output_condition_support	布林	
output_deployability	布林	
rules_to_display	upto all	要在輸出表格中顯示的最大規則數目。
display_upto	整數	如果在 rules_to_display 中設定了 upto, 請設定要在輸出表格中顯示的規則數目。最小值為 1。
field_transformations	布林	
records_summary	布林	
rule_statistics	布林	
most_frequent_values	布林	
most_frequent_fields	布林	
word_cloud	布林	
word_cloud_sort	Confidence Rulesupport Lift Conditionsupport Deployability	
word_cloud_display	整數	最小值為 1, 最大值為 20
max_predictions	整數	評分的每個輸入可套用的規則數上限。
criterion	Confidence Rulesupport Lift Conditionsupport Deployability	選取用來決定規則強度的測量。
allow_repeats	布林	判定評分中是否包括具有相同預測的規則。

表 113: associationrulesnode 內容 (繼續)

associationrulesnode 內容	資料類型	內容說明
check_input	NoPredictions Predictions NoCheck	

autoclassifiernode 內容



「自動分類器」節點用於建立和對比二元結果（是或否，流失或不流失等）的若干不同模型，使用戶可以選擇給定分析的最佳處理方法。由於受支援多種建模演算法，因此可以對用戶希望使用的方法、每種方法的特定選項以及對比結果的準則進行選取。節點會根據指定的選項產生一組模型，並根據您指定的準則對最佳候選項進行排名。

範例

```
node = stream.create("autoclassifier", "My node")
node.setPropertyValue("ranking_measure", "Accuracy")
node.setPropertyValue("ranking_dataset", "Training")
node.setPropertyValue("enable_accuracy_limit", True)
node.setPropertyValue("accuracy_limit", 0.9)
node.setPropertyValue("calculate_variable_importance", True)
node.setPropertyValue("use_costs", True)
node.setPropertyValue("svm", False)
```

表 114: autoclassifiernode 內容

autoclassifiernode 內容	值	內容說明
target	欄位	對於旗標目標，「自動分類器」節點要求單個目標欄位以及一個或多個輸入欄位。也可以指定加權與頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
ranking_measure	Accuracy Area_under_curve Profit Lift Num_variables	
ranking_dataset	Training Test	
number_of_models	整數	要併入模型區塊中的模型數。指定 1 和 100 之間的整數。
calculate_variable_importance	旗標	

表 114: autoclassifiernode 內容 (繼續)

autoclassifiernode 內容	值	內容說明
enable_accuracy_limit	旗標	
accuracy_limit	整數	0 到 100 之間的整數。
enable_area_under_curve_limit	旗標	
area_under_curve_limit	數字	介於 0.0 與 1.0 之間的實數。
enable_profit_limit	旗標	
profit_limit	數字	大於 0 的整數。
enable_lift_limit	旗標	
lift_limit	數字	這是大於 1.0 的實數。
enable_number_of_variables_limit	旗標	
number_of_variables_limit	數字	大於 0 的整數。
use_fixed_cost	旗標	
fixed_cost	數字	大於 0.0 的實數。
variable_cost	欄位	
use_fixed_revenue	旗標	
fixed_revenue	數字	大於 0.0 的實數。
variable_revenue	欄位	
use_fixed_weight	旗標	
fixed_weight	數字	這是大於 0.0 的實數。
variable_weight	欄位	
lift_percentile	數字	0 到 100 之間的整數。
enable_model_build_time_limit	旗標	
model_build_time_limit	數字	設定為分鐘數的整數，用於限制建立各個模型所花費的時間。
enable_stop_after_time_limit	旗標	
stop_after_time_limit	數字	設定為小時數的實數，用於限制執行自動分類器的總耗時。
enable_stop_after_valid_model_produced	旗標	
use_costs	旗標	
<algorithm>	旗標	啟用或停用特定的演算法。
<algorithm>.<property>	string	為特定的演算法設定內容值。請參閱第 199 頁的『設定演算法內容』主題，以取得更多資訊。

設定演算法內容

對於「自動分類器」、「自動數值」和「自動叢集」節點，可以使用通用格式來設定節點所使用的特定演算法的內容：

```
autonode.setKeyedPropertyValue(<algorithm>, <property>, <value>)
```

例如：

```
node.setKeyedPropertyValue("neuralnetwork", "method", "MultilayerPerceptron")
```

用於自動分類器節點的演算法名稱有 cart、chaid、quest、c50、logreg、decisionlist、bayesnet、discriminant、svm 和 knn。

用於自動數值節點的演算法名稱有 cart、chaid、neuralnetwork、genlin、svm、regression、linear 和 knn。

自動叢集節點的演算法名稱有 twostep、k-means 和 kohonen。

內容名為各演算法節點記載的標準格式。

含有句點或其他標點符號的演算法內容必須包括在單引號中，例如：

```
node.setKeyedPropertyValue("logreg", "tolerance", "1.0E-5")
```

也可以為內容分配多個值，例如：

```
node.setKeyedPropertyValue("decisionlist", "search_direction", ["Up", "Down"])
```

要啟用或取消特定演算法：

```
node.setPropertyValue("chaid", True)
```

註：如果某些演算法選項在「自動分類器」節點中無法使用，或者只能指定單一值而不是某個範圍的值，則編寫 Script 時受到的限制與以標準方式存取節點時一樣。

autoclusternode 內容



自動叢集節點估計和比較識別具有類似特性記錄群組的叢集作業模型。節點工作方式與其他自動建模節點相同，使您在一次建模運行中即可試驗多個選項組合。可以使用基本測量來比較模型，以嘗試對叢集模型的實用性進行過濾及分級，並根據特定欄位的重要性提供測量。

範例

```
node = stream.create("autocluster", "My node")
node.setPropertyValue("ranking_measure", "Silhouette")
node.setPropertyValue("ranking_dataset", "Training")
node.setPropertyValue("enable_silhouette_limit", True)
node.setPropertyValue("silhouette_limit", 5)
```

表 115: autoclusternode 內容

autoclusternode 內容	值	內容說明
evaluation	欄位	註：僅限於「自動叢集」節點。識別要計算重要性值的欄位。另外，可用於識別叢集區分此欄位的值的良好程度，從而識別模型預測此欄位的良好程度。

表 115: autoclusternode 內容 (繼續)

autoclusternode 內容	值	內容說明
ranking_measure	Silhouette Num_clusters Size_smallest_cluster Size_largest_cluster Smallest_to_largest Importance	
ranking_dataset	Training Test	
summary_limit	整數	要在報告中列出的模型的號碼。指定 1 和 100 之間的整數。
enable_silhouette_limit	旗標	
silhouette_limit	整數	0 到 100 之間的整數。
enable_number_less_limit	旗標	
number_less_limit	數字	介於 0.0 與 1.0 之間的實數。
enable_number_greater_limit	旗標	
number_greater_limit	數字	大於 0 的整數。
enable_smallest_cluster_limit	旗標	
smallest_cluster_units	Percentage Counts	
smallest_cluster_limit_percentage	數字	
smallest_cluster_limit_count	整數	大於 0 的整數。
enable_largest_cluster_limit	旗標	
largest_cluster_units	Percentage Counts	
largest_cluster_limit_percentage	數字	
largest_cluster_limit_count	整數	

表 115: *autoclusternode* 內容 (繼續)

autoclusternode 內容	值	內容說明
enable_smallest_largest_limit	旗標	
smallest_largest_limit	數字	
enable_importance_limit	旗標	
importance_limit_condition	Greater_than Less_than	
importance_limit_greater_than	數字	0 到 100 之間的整數。
importance_limit_less_than	數字	0 到 100 之間的整數。
<algorithm>	旗標	啟用或停用特定的演算法。
<algorithm>.<property>	<i>string</i>	為特定的演算法設定內容值。請參閱第 199 頁的『設定演算法內容』主題，以取得更多資訊。

autonumericnode 內容



自動數值節點使用多種不同方法估計和對比模型的連續數值範圍結果。此節點和自動分類器節點的工作方式相同，因此可以選擇要使用和要在單個建模傳送中使用多個選項組合進行測試的演算法。受支援的演算法包含神經網路、C&R 樹狀結構、CHAID、線性迴歸、廣義線性迴歸以及受支援向量機器 (SVM)。可以根據相關性、相對誤差或使用的變數數目來比較模型。

範例

```
node = stream.create("autonumeric", "My node")
node.setPropertyValue("ranking_measure", "Correlation")
node.setPropertyValue("ranking_dataset", "Training")
node.setPropertyValue("enable_correlation_limit", True)
node.setPropertyValue("correlation_limit", 0.8)
node.setPropertyValue("calculate_variable_importance", True)
node.setPropertyValue("neuralnetwork", True)
node.setPropertyValue("chaid", False)
```

表 116: *autonumericnode* 內容

autonumericnode 內容	值	內容說明
custom_fields	旗標	如果為 True，將使用自訂欄位設定代替類型節點設定。
target	欄位	「自動數值」節點要求單個目標欄位以及一個或多個輸入欄位。也可以指定加權與頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
inputs	[<i>field1 ... field2</i>]	
partition	欄位	

表 116: autonumericnode 內容 (繼續)

autonumericnode 內容	值	內容說明
use_frequency	旗標	
frequency_field	欄位	
use_weight	旗標	
weight_field	欄位	
use_partitioned_data	旗標	如果定義分割區欄位，則僅將訓練資料用於模型建置。
ranking_measure	Correlation NumberOfFields	
ranking_dataset	Test Training	
number_of_models	整數	要併入模型區塊中的模型數。指定 1 和 100 之間的整數。
calculate_variable_importance	旗標	
enable_correlation_limit	旗標	
correlation_limit	整數	
enable_number_of_fields_limit	旗標	
number_of_fields_limit	整數	
enable_relative_error_limit	旗標	
relative_error_limit	整數	
enable_model_build_time_limit	旗標	
model_build_time_limit	整數	
enable_stop_after_time_limit	旗標	
stop_after_time_limit	整數	
stop_if_valid_model	旗標	
<algorithm>	旗標	啟用或停用特定的演算法。
<algorithm>.<property>	string	為特定的演算法設定內容值。請參閱第 199 頁的『設定演算法內容』主題，以取得更多資訊。

bayesnetnode 內容



通過貝葉斯網路節點，你可以利用對真實世界認知的判斷力並結合所觀察和記錄的證據來建立機率模型。該節點著重於主要用於分類的 Tree Augmented Naïve Bayes (TAN) 和 Markov Blanket 網路。

範例

```
node = stream.create("bayesnet", "My node")
node.setPropertyValue("continue_training_existing_model", True)
node.setPropertyValue("structure_type", "MarkovBlanket")
node.setPropertyValue("use_feature_selection", True)
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("independence", "Pearson")
```

表 117: bayesnetnode 內容

bayesnetnode 內容	值	內容說明
inputs	[field1 ... fieldn]	貝葉斯網路模型使用一個目標欄位以及一個或多個輸入欄位。連續欄位將自動進行分組。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
continue_training_existing_model	旗標	
structure_type	TAN MarkovBlanket	選取要在建立貝葉斯網路時使用的結構。
use_feature_selection	旗標	
parameter_learning_method	Likelihood Bayes	指定用於預測母節點的值已知的節點之間的條件機率表格的方法。
mode	Expert Simple	
missing_values	旗標	
all_probabilities	旗標	
independence	Likelihood Pearson	指定用於確定兩個變數上的成對觀察是否彼此獨立的方法。
significance_level	數字	指定用於確定獨立性的截斷值。
maximal_conditioning_set	數字	設定用於獨立性正在測試的條件變數數目的上限。

表 117: bayesnetnode 內容 (繼續)

bayesnetnode 內容	值	內容說明
inputs_always_selected	[field1 ... fieldn]	指定建立貝葉斯網路時始終使用的資料集中的欄位。 註: 一律會選取目標欄位。
maximum_number_inputs	數字	指定建立貝葉斯網路時使用的輸入欄位的最大數目。
calculate_variable_importance	旗標	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	
adjusted_propensity_partition	Test Validation	

buildr 內容



「R 建置」節點可讓您輸入自訂 R Script，以執行 IBM SPSS Modeler 中部署的模型建置和模型評分。

範例

```
node = stream.create("buildr", "My node")
node.setPropertyValue("score_syntax", "")
result<-predict(modelerModel,newdata=modelerData)
modelerData<-cbind(modelerData,result)
var1<-
c(fieldName="NaPrediction",fieldLabel="",fieldStorage="real",fieldMeasure="",
fieldFormat="",fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)""")
```

表 118: buildr 內容

buildr 內容	值	內容說明
build_syntax	string	這是用於進行模型建置的 R Script 語法。
score_syntax	string	用於模型評分的 R Scripting 語法。
convert_flags	StringsAndDoubles LogicalValues	用於轉換旗標欄位的選項。
convert_datetime	旗標	用於將日期或日期時間格式的變數轉換為 R 日期/時間格式的選項。
convert_datetime_class	POSIXct POSIXlt	用於指定將日期或日期時間格式的變數轉換為何種格式的選項。
convert_missing	旗標	用於將遺漏值轉換為 R NA 值的選項。

buildr 內容	值	內容說明
output_html	旗標	用來將圖形顯示在 R 模型區塊中的標籤上的選項。
output_text	旗標	用來將 R 主控台文字輸出寫入 R 模型區塊中的標籤的選項。

c50node 內容



C5.0 節點建立決策樹狀結構或規則集。該模型的工作原理是在每個層次根據提供最大資訊增益的欄位分割樣本。目標欄位必須為種類欄位。容許多次分割為兩個以上的子群組。

範例

```
node = stream.create("c50", "My node")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "C5_Drug")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("output_type", "DecisionTree")
node.setPropertyValue("use_xval", True)
node.setPropertyValue("xval_num_folds", 3)
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("favor", "Generality")
node.setPropertyValue("min_child_records", 3)
# "Costs" tab
node.setPropertyValue("use_costs", True)
node.setPropertyValue("costs", [["drugA", "drugX", 2]])
```

c50node 內容	值	內容說明
target	欄位	C5.0 模型使用單個目標欄位以及一個或多個輸入欄位。還可以指定加權欄位。請參閱第 191 頁的『 一般建模節點內容 』主題，以取得更多資訊。
output_type	DecisionTree RuleSet	
group_symbolics	旗標	
use_boost	旗標	
boost_num_trials	數字	
use_xval	旗標	
xval_num_folds	數字	
mode	Simple Expert	

表 119: c50node 內容 (繼續)

c50node 內容	值	內容說明
favor	Accuracy Generality	首選精確度或通用性。
expected_noise	數字	
min_child_records	數字	
pruning_severity	數字	
use_costs	旗標	
costs	結構化	這是結構化內容。
use_winning	旗標	
use_global_pruning	旗標	預設為「啟用」 (True)。
calculate_variable_importance	旗標	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	
adjusted_propensity_partition	Test Validation	

carmanode 內容



CARMA 模型會從資料中擷取一組規則，而不需要您指定輸入或目標欄位。與 Apriori 相反，CARMA 節點提供了規則支援（同時支援前提條件和結果）的建置設定，而不僅僅是前提條件支援。這就意味著產生的規則可以用於更多應用程式，例如用於尋找產品或服務（前提條件）的清單，這些產品或服務的結果為想在節日期間促銷的商品。

範例

```
node = stream.create("carma", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("use_transactional_data", True)
node.setPropertyValue("inputs", ["BP", "Cholesterol", "Drug"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "age_bp_drug")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("min_supp", 10.0)
node.setPropertyValue("min_conf", 30.0)
node.setPropertyValue("max_size", 5)
# Expert Options
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("use_pruning", True)
node.setPropertyValue("pruning_value", 300)
node.setPropertyValue("vary_support", True)
```



```
node.setPropertyValue("estimated_transactions", 30)
node.setPropertyValue("rules_without_antecedents", True)
```

表 120: carmanode 內容

carmanode 內容	值	內容說明
inputs	[field1 ... fieldn]	CARMA 模型使用輸入欄位清單，而不使用目標欄位。不會使用加權和頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
id_field	欄位	用作模型建置 ID 欄位的欄位。
contiguous	旗標	用於指定 ID 欄位中的 ID 是否連續。
use_transactional_data	旗標	
content_field	欄位	
min_supp	number(percent)	與規則支援相關，而不是與條件支援相關。預設值是 20%。
min_conf	number(percent)	預設值是 20%。
max_size	數字	預設值是 10。
mode	Simple Expert	預設值是 Simple。
exclude_multiple	旗標	排除具有多結果的規則。預設值是 False。
use_pruning	旗標	預設值是 False。
pruning_value	數字	預設值為 500。
vary_support	旗標	
estimated_transactions	整數	
rules_without_antecedents	旗標	

cartnode 內容



分類和迴歸方法 (C&R) 樹狀結構節點產生可用於預測或分類未來觀察的決策樹狀結構。該方法通過在每一個步驟最大限度降低不純潔度，使用遞歸分區來將訓練記錄分割為組。如果樹狀結構中某個節點中 100% 的觀察值都的目標欄位的一個特定種類，那麼該節點將被認為「純潔」。目標和輸入欄位可以是數值範圍或類別（名義、序數或旗標）；所有分割都是二元（只有兩個子群組）。

範例

```
node = stream.createAt("cart", "My node", 200, 100)
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "BP", "Cholesterol"])
# "Build Options" tab, "Objective" panel
node.setPropertyValue("model_output_type", "InteractiveBuilder")
node.setPropertyValue("use_tree_directives", True)
node.setPropertyValue("tree_directives", "" "Grow Node Index 0 Children 1 2
```

```

Grow Node Index 2 Children 3 4""")
# "Build Options" tab, "Basics" panel
node.setPropertyValue("prune_tree", False)
node.setPropertyValue("use_std_err_rule", True)
node.setPropertyValue("std_err_multiplier", 3.0)
node.setPropertyValue("max_surrogates", 7)
# "Build Options" tab, "Stopping Rules" panel
node.setPropertyValue("use_percentage", True)
node.setPropertyValue("min_parent_records_pc", 5)
node.setPropertyValue("min_child_records_pc", 3)
# "Build Options" tab, "Advanced" panel
node.setPropertyValue("min_impurity", 0.0003)
node.setPropertyValue("impurity_measure", "Twoing")
# "Model Options" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Cart_Drug")

```

表 121: cartnode 內容

cartnode 內容	值	內容說明
target	欄位	C&R 樹狀結構模型需要單個目標欄位以及一個或多個輸入欄位。還可以指定頻率欄位。請參閱第 191 頁的『 一般建模節點內容 』主題，以取得更多資訊。
continue_training_existing_model	旗標	
objective	Standard Boosting Bagging psm	psm 用於極大型資料集，且需要伺服器連線。
model_output_type	Single InteractiveBuilder	
use_tree_directives	旗標	
tree_directives	string	指定生成樹狀結構的指引。指引可放在三引號中，以防止跳出新行或引號。請注意，指引可能對資料或建模選項的細小變更高度敏感，並且可能無法通用於其他資料集。
use_max_depth	Default Custom	
max_depth	整數	樹狀結構深度上限，從 0 到 1000。僅在 use_max_depth = Custom 時使用。
prune_tree	旗標	刪改樹狀結構以避免過度適合。
use_std_err	旗標	在風險中（在標準錯誤中）使用最大差異。

表 121: `cartnode` 內容 (繼續)

cartnode 內容	值	內容說明
<code>std_err_multiplier</code>	數字	最大差異。
<code>max_surrogates</code>	數字	最大代理數。
<code>use_percentage</code>	旗標	
<code>min_parent_records_pc</code>	數字	
<code>min_child_records_pc</code>	數字	
<code>min_parent_records_abs</code>	數字	
<code>min_child_records_abs</code>	數字	
<code>use_costs</code>	旗標	
<code>costs</code>	結構化	結構化內容。
<code>priors</code>	Data Equal Custom	
<code>custom_priors</code>	結構化	結構化內容。
<code>adjust_priors</code>	旗標	
<code>trails</code>	數字	推升法或自助重抽總合法的成分模型數目。
<code>set_ensemble_method</code>	Voting HighestProbability HighestMeanProbability	類別目標的預設結合規則。
<code>range_ensemble_method</code>	Mean Median	連續目標的預設結合規則。
<code>large_boost</code>	旗標	將推升法套用至極大型資料集。
<code>min_impurity</code>	數字	
<code>impurity_measure</code>	Gini Twoing Ordered	
<code>train_pct</code>	數字	過度適合預防集。
<code>set_random_seed</code>	旗標	抄寫結果選項。
<code>seed</code>	數字	
<code>calculate_variable_importance</code>	旗標	

表 121: cartnode 內容 (繼續)

cartnode 內容	值	內容說明
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	
adjusted_propensity_partition	Test Validation	

chaidnode 內容



CHAID 使用卡方統計資料來產生決策樹狀結構，以確定最佳的分割。與 C&R 樹狀結構和 QUEST 節點不同，CHAID 可產生非二元樹狀結構，表示部分分割有兩個以上的分支。目標欄位和輸入欄位可以是數值範圍（連續）或類別欄位。Exhaustive CHAID 是對 CHAID 的修改，它會執行更徹底的工作來檢查所有可能的分割，但需要更長的計算時間。

範例

```

filename = stream.createAt("variablefile", "My node", 100, 100)
filename.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node = stream.createAt("chaid", "My node", 200, 100)
stream.link(filename, node)

node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "CHAID")
node.setPropertyValue("method", "Chaid")
node.setPropertyValue("model_output_type", "InteractiveBuilder")
node.setPropertyValue("use_tree_directives", True)
node.setPropertyValue("tree_directives", "Test")
node.setPropertyValue("split_alpha", 0.03)
node.setPropertyValue("merge_alpha", 0.04)
node.setPropertyValue("chi_square", "Pearson")
node.setPropertyValue("use_percentage", False)
node.setPropertyValue("min_parent_records_abs", 40)
node.setPropertyValue("min_child_records_abs", 30)
node.setPropertyValue("epsilon", 0.003)
node.setPropertyValue("max_iterations", 75)
node.setPropertyValue("split_merged_categories", True)
node.setPropertyValue("bonferroni_adjustment", True)

```

表 122: chaidnode 內容

chaidnode 內容	值	內容說明
target	欄位	CHAID 模型需要一個目標以及一個或多個輸入欄位。還可以指定頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
continue_training_existing_model	旗標	

表 122: chaidnode 內容 (繼續)

chaidnode 內容	值	內容說明
objective	Standard Boosting Bagging psm	psm 用於極大型資料集，且需要伺服器連線。
model_output_type	Single InteractiveBuilder	
use_tree_directives	旗標	
tree_directives	string	
method	Chaid ExhaustiveChaid	
use_max_depth	Default Custom	
max_depth	整數	樹狀結構深度上限，從 0 到 1000。僅在 use_max_depth = Custom 時使用。
use_percentage	旗標	
min_parent_records_pc	數字	
min_child_records_pc	數字	
min_parent_records_abs	數字	
min_child_records_abs	數字	
use_costs	旗標	
costs	結構化	結構化內容。
trails	數字	推升法或自助重抽總合法的成分模型數目。
set_ensemble_method	Voting HighestProbability HighestMeanProbability	類別目標的預設結合規則。
range_ensemble_method	Mean Median	連續目標的預設結合規則。
large_boost	旗標	將推升法套用至極大型資料集。

表 122: chaidnode 內容 (繼續)

chaidnode 內容	值	內容說明
split_alpha	數字	用於分割的顯著性層級。
merge_alpha	數字	用於合併的顯著性層級。
bonferroni_adjustment	旗標	使用 Bonferroni 方法來調整顯著值。
split_merged_categories	旗標	容許重新分割合併種類。
chi_square	Pearson LR	用來計算方差統計值的方法：皮爾森 (Pearson) 或概似比
epsilon	數字	預期儲存格變更的最小變更。
max_iterations	數字	聚合的疊代數上限。
set_random_seed	整數	
seed	數字	
calculate_variable_importance	旗標	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	
adjusted_propensity_partition	Test Validation	
maximum_number_of_models	整數	

coxregnode 內容



使用 Cox 迴歸節點，您可以在已有審查記錄的情況下建置時間事件資料的生存模型。此模型會產生一個存活函數，可針對輸入變數的給定值，預測感興趣事件在給定時間 (T) 發生的機率。

範例

```
node = stream.create("coxreg", "My node")
node.setPropertyValue("survival_time", "tenure")
node.setPropertyValue("method", "BackwardsStepwise")
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("removal_criterion", "Conditional")
node.setPropertyValue("survival", True)
```

表 123: coxregnode 內容

coxregnode 內容	值	內容說明
survival_time	欄位	Cox 迴歸模型需要一個包含存活時間的欄位。

表 123: coxregnode 內容 (繼續)

coxregnode 內容	值	內容說明
target	欄位	Cox 迴歸模型需要一個目標欄位以及一個或多個輸入欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
method	Enter Stepwise BackwardsStepwise	
groups	欄位	
model_type	MainEffects Custom	
custom_terms	["BP*Sex" "BP*Age"]	
mode	Expert Simple	
max_iterations	數字	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	

表 123: coxregnode 內容 (繼續)

coxregnode 內容	值	內容說明
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
removal_criterion	LR Wald Conditional	
probability_entry	數字	
probability_removal	數字	
output_display	EachStep LastStep	
ci_enable	旗標	
ci_value	90 95 99	
correlation	旗標	
display_baseline	旗標	
survival	旗標	
hazard	旗標	
log_minus_log	旗標	
one_minus_survival	旗標	
separate_line	欄位	
value	數字或字串	如果未對某個欄位指定值，那麼將對該欄位使用預設選項 "Mean"。

decisionlistnode 內容



決策清單節點可識別子群組或區段，以顯示給定二元結果相對於整體的更高或更低可能性。例如，您或許在尋找那些最不可能流失的客戶或最有可能對某個商業活動作出積極回應的客戶。透過新增自訂區段和並列預覽替代模型來比較結果，您可以將自己的業務知識融入模型。決策清單模型由一組規則構成，其中每個規則具備一個條件和一個結果。規則依順序套用，相符的第一個規則將決定結果。

範例

```
node = stream.create("decisionlist", "My node")
node.setPropertyValue("search_direction", "Down")
node.setPropertyValue("target_value", 1)
node.setPropertyValue("max_rules", 4)
node.setPropertyValue("min_group_size_pct", 15)
```

表 124: decisionlistnode 內容

decisionlistnode 內容	值	內容說明
target	欄位	決策清單模型使用一個目標以及一個或多個輸入欄位。還可以指定頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
model_output_type	Model InteractiveBuilder	
search_direction	Up Down	與尋找區段相關；其中 Up 相當於「高機率」，而 Down 相當於「低機率」。
target_value	string	如果未指定，則旗標採用 true 值。
max_rules	整數	除餘數之外的上限區段數。
min_group_size	整數	下限區段大小。
min_group_size_pct	數字	下限區段大小（以百分比表示）。
confidence_level	數字	為了使輸入欄位符合新增到區段定義的條件，輸入欄位必須提高的回應概似值（提升）的下限臨界值。
max_segments_per_rule	整數	
mode	Simple Expert	
bin_method	EqualWidth EqualCount	
bin_count	數字	
max_models_per_cycle	整數	清單的搜尋寬度。
max_rules_per_cycle	整數	區段規則的搜尋寬度。
segment_growth	數字	

表 124: *decisionlistnode* 內容 (繼續)

decisionlistnode 內容	值	內容說明
include_missing	旗標	
final_results_only	旗標	
reuse_fields	旗標	容許重複使用屬性（出現在規則中的輸入欄位）。
max_alternatives	整數	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	
adjusted_propensity_partition	Test Validation	

discriminantnode 內容



區別分析比邏輯迴歸具有更嚴格的假設，但在符合這些假設時，可以是邏輯迴歸分析有價值的替代方案或補充。

範例

```
node = stream.create("discriminant", "My node")
node.setPropertyValue("target", "custcat")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("method", "Stepwise")
```

表 125: *discriminantnode* 內容

discriminantnode 內容	值	內容說明
target	欄位	判別模型需要一個目標欄位以及一個或多個輸入欄位。不會使用加權和頻率欄位。請參閱第 191 頁的『 一般建模節點內容 』主題，以取得更多資訊。
method	Enter Stepwise	
mode	Simple Expert	
prior_probabilities	AllEqual ComputeFromSizes	
covariance_matrix	WithinGroups SeparateGroups	

表 125: discriminantnode 內容 (繼續)

discriminantnode 內容	值	內容說明
means	旗標	「進階輸出」對話框中的統計資料選項。
univariate_anovas	旗標	
box_m	旗標	
within_group_covariance	旗標	
within_groups_correlation	旗標	
separate_groups_covariance	旗標	
total_covariance	旗標	
fishers	旗標	
unstandardized	旗標	
casewise_results	旗標	「進階輸出」對話框中的分類選項。
limit_to_first	數字	預設值是 10。
summary_table	旗標	
leave_one_classification	旗標	
combined_groups	旗標	
separate_groups_covariance	旗標	矩陣選項類共變異。
territorial_map	旗標	
combined_groups	旗標	統計圖選項 聯合群組。
separate_groups	旗標	統計圖選項 獨立群組。
summary_of_steps	旗標	
F_pairwise	旗標	
stepwise_method	WilksLambda UnexplainedVariance MahalanobisDistance SmallestF RaosV	
V_to_enter	數字	
criteria	UseValue UseProbability	
F_value_entry	數字	預設值是 3.84。

表 125: *discriminantnode* 內容 (繼續)

discriminantnode 內容	值	內容說明
F_value_removal	數字	預設值是 2.71。
probability_entry	數字	預設值是 0.05。
probability_removal	數字	預設值是 0.10。
calculate_variable_importance	旗標	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	
adjusted_propensity_partition	Test Validation	

extensionmodelnode 內容



通過「延伸模型」節點，可以執行 R 或 Python for Spark Script 來建立和分數結果。

Python for Spark 範例

```
#### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_build", "extension_build")
node.setPropertyValue("syntax_type", "Python")

build_script = """
import json
import spss.pyspark.runtime
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.linalg import DenseVector
from pyspark.mllib.tree import DecisionTree

cxt = spss.pyspark.runtime.getContext()
df = cxt.getSparkInputData()
schema = df.dtypes[:]

target = "Drug"
predictors = ["Age", "BP", "Sex", "Cholesterol", "Na", "K"]

def metaMap(row, schema):
    col = 0
    meta = []
    for (cname, ctype) in schema:
        if ctype == 'string':
            meta.append(set([row[col]]))
        else:
            meta.append((row[col], row[col]))
        col += 1
    return meta

def metaReduce(meta1, meta2, schema):
    col = 0
    meta = []
    for (cname, ctype) in schema:
```

```

        if ctype == 'string':
            meta.append(meta1[col].union(meta2[col]))
        else:
            meta.append((min(meta1[col][0],meta2[col][0]),max(meta1[col][1],meta2[col][1])))
        col += 1
    return meta

metadata = df.rdd.map(lambda row: metaMap(row,schema)).reduce(lambda x,y:metaReduce(x,y,schema))

def setToList(v):
    if isinstance(v,set):
        return list(v)
    return v

metadata = map(lambda x: setToList(x), metadata)
print metadata

lookup = {}
for i in range(0,len(schema)):
    lookup[schema[i][0]] = i

def row2LabeledPoint(dm,lookup,target,predictors,row):
    target_index = lookup[target]
    tval = dm[target_index].index(row[target_index])
    pvals = []
    for predictor in predictors:
        predictor_index = lookup[predictor]
        if isinstance(dm[predictor_index],list):
            pval = dm[predictor_index].index(row[predictor_index])
        else:
            pval = row[predictor_index]
        pvals.append(pval)
    return LabeledPoint(tval,DenseVector(pvals))

# count number of target classes
predictorClassCount = len(metadata[lookup[target]])

# define function to extract categorical predictor information from datamodel
def getCategoricalFeatureInfo(dm,lookup,predictors):
    info = {}
    for i in range(0,len(predictors)):
        predictor = predictors[i]
        predictor_index = lookup[predictor]
        if isinstance(dm[predictor_index],list):
            info[i] = len(dm[predictor_index])
    return info

# convert dataframe to an RDD containing LabeledPoint
lps = df.rdd.map(lambda row: row2LabeledPoint(metadata,lookup,target,predictors,row))

treeModel = DecisionTree.trainClassifier(
    lps,
    numClasses=predictorClassCount,
    categoricalFeaturesInfo=getCategoricalFeatureInfo(metadata, lookup, predictors),
    impurity='gini',
    maxDepth=5,
    maxBins=100)

_outputPath = cxt.createTemporaryFolder()
treeModel.save(cxt.getSparkContext(), _outputPath)
cxt.setModelContentFromPath("TreeModel", _outputPath)
cxt.setModelContentFromString("model.dm",json.dumps(metadata), mimeType="application/json")\
    .setModelContentFromString("model.structure",treeModel.toDebugString())

"""

node.setPropertyValue("python_build_syntax", build_script)

```

R 範例

```

#### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_build_syntax", """modelerModel <-
lm(modelerData$Na-modelerData$K,modelerData)
modelerDataModel
modelerModel
""")

```

表 126: extensionmodelnode 內容

extensionmodelnode 內容	值	內容說明
syntax_type	<i>r</i> <i>Python</i>	指定執行哪個 Script – R 或 Python（預設值是 R）。
r_build_syntax	<i>string</i>	這是用於進行模型建置的 R Script 語法。
r_score_syntax	<i>string</i>	這是用於進行模型評分的 R Script 語法。
python_build_syntax	<i>string</i>	這是用於進行模型建置的 Python Script 語法。
python_score_syntax	<i>string</i>	這是用於進行模型評分的 Python Script 語法。
convert_flags	StringsAndDoubles LogicalValues	用於轉換旗標欄位的選項。
convert_missing	旗標	用於將遺漏值轉換為 R NA 值的選項。
convert_datetime	旗標	用於將日期或日期時間格式的變數轉換為 R 日期/時間格式的選項。
convert_datetime_class	POSIXct POSIXlt	用於指定將日期或日期時間格式的變數轉換為何種格式的選項。
output_html	旗標	用來將圖形顯示在 R 模型區塊中的標籤上的選項。
output_text	旗標	用來將 R 主控台文字輸出寫入 R 模型區塊中的標籤的選項。

factornode 內容



PCA/因素節點提供強大的資料減少技術來減少資料的複雜性。主成份分析（PCA）可找出輸入欄位的線性組合，該組合最好地擷取了整個欄位集中的變異，且組合中的各個成分相互正交（相互垂直）。因數分析則試圖識別底層因素，這些因素說明觀測的欄位集合內的相關性型樣。對於這兩種方法，其共同的目標是找到可對原始欄位集合中的資訊進行有效總結的少量衍生欄位。

範例

```
node = stream.create("factor", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["BP", "Na", "K"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Factor_Age")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("method", "GLS")
# Expert options
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("complete_records", True)
node.setPropertyValue("matrix", "Covariance")
```

```

node.setPropertyValue("max_iterations", 30)
node.setPropertyValue("extract_factors", "ByFactors")
node.setPropertyValue("min_eigenvalue", 3.0)
node.setPropertyValue("max_factor", 7)
node.setPropertyValue("sort_values", True)
node.setPropertyValue("hide_values", True)
node.setPropertyValue("hide_below", 0.7)
# "Rotation" section
node.setPropertyValue("rotation", "DirectOblimin")
node.setPropertyValue("delta", 0.3)
node.setPropertyValue("kappa", 7.0)

```

表 127: factornode 內容

factornode 內容	值	內容說明
inputs	[field1 ... fieldN]	PCA/因子模型使用輸入欄位的清單，但不使用目標。不會使用加權和頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
method	PC ULS GLS ML PAF Alpha Image	
mode	Simple Expert	
max_iterations	數字	
complete_records	旗標	
matrix	Correlation Covariance	
extract_factors	ByEigenvalues ByFactors	
min_eigenvalue	數字	
max_factor	數字	

表 127: factornode 內容 (繼續)

factornode 內容	值	內容說明
rotation	None Varimax DirectOblimin Equamax Quartimax Promax	
delta	數字	如果選取 DirectOblimin 作為旋轉資料的類型，則可以指定 delta 的值。 如果未指定一個值，則將使用 delta 的預設值。
kappa	數字	如果選取 Promax 作為旋轉資料的類型，則可以指定 kappa 的值。 如果未指定一個值，則將使用 kappa 的預設值。
sort_values	旗標	
hide_values	旗標	
hide_below	數字	

featureselectionnode 內容



功能選擇節點會根據某組準則（例如遺漏百分比）篩選可移除的輸入欄位，對於保留的輸入，隨後將對其相對於指定目標的重要性進行排等級。例如，假如某個給定資料集有上千個潛在輸入，那麼哪些輸入最有可能用於對病患結果進行建模呢？

範例

```
node = stream.create("featureselection", "My node")
node.setPropertyValue("screen_single_category", True)
node.setPropertyValue("max_single_category", 95)
node.setPropertyValue("screen_missing_values", True)
node.setPropertyValue("max_missing_values", 80)
node.setPropertyValue("criteria", "Likelihood")
node.setPropertyValue("unimportant_below", 0.8)
node.setPropertyValue("important_above", 0.9)
node.setPropertyValue("important_label", "Check Me Out!")
node.setPropertyValue("selection_mode", "TopN")
node.setPropertyValue("top_n", 15)
```

有關建立和套用「功能選擇」模型的更詳細範例，請參閱第 4 頁的『獨立式 Script 範例：產生功能選項模式』。

表 128: *featureselectionnode* 內容

featureselectionnode 內容	值	內容說明
target	欄位	功能選項模式相對於指定的目標對預測值進行排名。不會使用加權和頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
screen_single_category	旗標	如果為 True，則將篩選相對於記錄總數而言同個種類中具有過多記錄的欄位。
max_single_category	數字	指定 screen_single_category 為 True 時使用的臨界值。
screen_missing_values	旗標	如果為 True，則將篩選具有過多遺漏值（以記錄總數百分比表示）的欄位。
max_missing_values	數字	
screen_num_categories	旗標	如果為 True，則將篩選相對於記錄總數而言具有過多種類的欄位。
max_num_categories	數字	
screen_std_dev	旗標	如果為 True，則將篩選標準差小於或等於指定最小值的欄位。
min_std_dev	數字	
screen_coeff_of_var	旗標	如果為 True，則將篩選變異係數小於或等於指定最小值的欄位。
min_coeff_of_var	數字	
criteria	Pearson Likelihood CramersV Lambda	對照種類目標對種類預測值進行分等級時，指定用作確定重要性值的根據的測量。
unimportant_below	數字	指定用於將變數分級為「重要」、「邊際」或「不重要」的 p 臨界值。接受 0.0 與 1.0 之間的值。
important_above	數字	接受 0.0 與 1.0 之間的值。
unimportant_label	string	指定「不重要」分等級的標籤。
marginal_label	string	
important_label	string	
selection_mode	ImportanceLevel ImportanceValue TopN	

表 128: *featureselectionnode* 內容 (繼續)

featureselectionnode 內容	值	內容說明
select_important	旗標	當 selection_mode 設為 ImportanceLevel 時，指定是否選取重要欄位。
select_marginal	旗標	當 selection_mode 設為 ImportanceLevel 時，指定是否選取一般欄位。
select_unimportant	旗標	當 selection_mode 設為 ImportanceLevel 時，指定是否選取不重要欄位。
importance_value	數字	當 selection_mode 設為 ImportanceValue 時，指定要使用的截斷值。接受 0 與 100 之間的值。
top_n	整數	當 selection_mode 設為 TopN 時，指定要使用的截斷值。接受 0 與 1000 之間的值。

genlinnode 內容



「廣義線性」模型對廣義線性模型進行了擴展，這樣應變數通過指定的鏈結函數與因子和共變數線性相關。此外，此模式允許變數具有非常態分配。它涵蓋大量統計模型的功能，包括線性迴歸、邏輯迴歸、計數資料的對數線性模型，以及區間受限存活模型。

範例

```
node = stream.create("genlin", "My node")
node.setPropertyValue("model_type", "MainAndAllTwoWayEffects")
node.setPropertyValue("offset_type", "Variable")
node.setPropertyValue("offset_field", "Claimant")
```

表 129: *genlinnode* 內容

genlinnode 內容	值	內容說明
target	欄位	廣義線性模型要求單個目標欄位（必須是名義或旗標欄位），以及一個或多個輸入欄位。還可以指定加權欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
use_weight	旗標	
weight_field	欄位	欄位類型僅為連續欄位。
target_represents_trials	旗標	
trials_type	Variable FixedValue	
trials_field	欄位	欄位類型有連續、旗標或序數。
trials_number	數字	預設值是 10。

表 129: genlinnode 內容 (繼續)

genlinnode 內容	值	內容說明
model_type	MainEffects MainAndAllTwoWayEffects	
offset_type	Variable FixedValue	
offset_field	欄位	欄位類型僅為連續欄位。
offset_value	數字	必須是實數。
base_category	Last First	
include_intercept	旗標	
mode	Simple Expert	
distribution	BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIAL	IGAUSS: 反向高斯。 NEGBIN: 負二項式。
negbin_para_type	Specify Estimate	
negbin_parameter	數字	預設值為 1。必須包含非負實數。
tweedie_parameter	數字	

表 129: genlinnode 內容 (繼續)

genlinnode 內容	值	內容說明
link_function	IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPower PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT	CLOGLOG: 互補雙對數。 LOGC: 對數互補。 NEGBIN: 負二項式。 NLOGLOG: 負對數存活函數的對數。 CUMCAUCHIT: 累積 Cauchit。 CUMCLOGLOG: 累加互補雙對數。 CUMLOGIT: 累加 Logit。 CUMNLOGLOG: 累加負對數存活函數的對數。 CUMPROBIT 累加 Probit。
power	數字	值必須是非零實數。
method	Hybrid Fisher NewtonRaphson	
max_fisher_iterations	數字	預設值為 1; 只容許正整數。
scale_method	MaxLikelihoodEstimate Deviance PearsonChiSquare FixedValue	
scale_value	數字	預設值為 1; 必須大於 0。

表 129: genlinnode 內容 (繼續)

genlinnode 內容	值	內容說明
covariance_matrix	ModelEstimator	
	RobustEstimator	
max_iterations	數字	預設值為 100; 僅允許使用非負整數。
max_step_halving	數字	預設值為 5; 僅允許使用正整數。
check_separation	旗標	
start_iteration	數字	預設值為 20; 僅容許使用正整數。
estimates_change	旗標	
estimates_change_min	數字	預設值為 1E-006; 僅容許使用正數。
estimates_change_type	Absolute	
	Relative	
loglikelihood_change	旗標	
loglikelihood_change_min	數字	只接受正數數目。
loglikelihood_change_type	Absolute	
	Relative	
hessian_convergence	旗標	
hessian_convergence_min	數字	只接受正數數目。
hessian_convergence_type	Absolute	
	Relative	
case_summary	旗標	
contrast_matrices	旗標	
descriptive_statistics	旗標	
estimable_functions	旗標	
model_info	旗標	
iteration_history	旗標	
goodness_of_fit	旗標	
print_interval	數字	預設值為 1; 必須是正整數。
model_summary	旗標	
lagrange_multiplier	旗標	
parameter_estimates	旗標	
include_exponential	旗標	
covariance_estimates	旗標	
correlation_estimates	旗標	

表 129: genlinnode 內容 (繼續)

genlinnode 內容	值	內容說明
analysis_type	TypeI TypeIII TypeIAndTypeIII	
statistics	Wald LR	
citype	Wald Profile	
tolerancelevel	數字	預設值是 0.0001。
confidence_interval	數字	預設值是 95。
loglikelihood_function	Full Kernel	
singularity_tolerance	1E-007 1E-008 1E-009 1E-010 1E-011 1E-012	
value_order	Ascending Descending DataOrder	
calculate_variable_importance	旗標	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	
adjusted_propensity_partition	Test Validation	

glmmnode 內容



概化線性混合模型 (GLMM) 延伸了線性模型，使得目標可以有非常態分佈，通過指定的連接函數與因子和共變數線性相關，並且觀察可能相關。廣義線性混合模型涵蓋多種模式，從非常態縱向資料的簡單線性迴歸，到複雜的多層級模式。

glmmnode 內容	值	內容說明
residual_subject_spec	結構化	這是指定的種類欄位的值組合，這些欄位唯一地定義資料集中的主旨。
repeated_measures	結構化	這些欄位用於識別重複的觀察。
residual_group_spec	[field1 ... fieldN]	這些欄位用於定義重複效應共變異參數的獨立集合。
residual_covariance_type	Diagonal AR1 ARMA11 COMPOUND_SYMMETRY IDENTITY TOEPLITZ UNSTRUCTURED VARIANCE_COMPONENTS	指定殘值的共變異結構。
custom_target	旗標	指出是使用上游節點中定義的目標 (<code>false</code>) 或使用 <code>target_field</code> 指定的自訂目標 (<code>true</code>)。
target_field	欄位	用作目標的欄位 (如果 <code>custom_target</code> 為 <code>true</code>)。
use_trials	旗標	指出當目標回應是在試用集中發生的數個事件時，是要使用額外欄位還是值來指定試用數目。預設值是 <code>false</code> 。
use_field_or_value	Field Value	指出是使用欄位 (預設值) 還是值來指定試用數目。
trials_field	欄位	用來指定試用數目的欄位。
trials_value	整數	用來指定試用數目的值。指定的話，最小值為 1。
use_custom_target_reference	旗標	指定是否將自訂參照種類用於類別目標。預設值是 <code>false</code> 。

表 130: *glmmnode* 內容 (繼續)

glmmnode 內容	值	內容說明
target_reference_value	<i>string</i>	要使用的參照種類 (如果 use_custom_target_reference 為 true)。
dist_link_combination	Nominal Logit GammaLog BinomialLogit PoissonLog BinomialProbit NegbinLog BinomialLogC Custom	用來分佈目標值的一般模型。選擇 Custom 可以指定 target_distribution 所提供的清單中的分佈。
target_distribution	Normal Binomial Multinomial Gamma Inverse NegativeBinomial Poisson	當 dist_link_combination 為 Custom 時目標值的分佈。

表 130: glmmnode 內容 (繼續)

glmmnode 內容	值	內容說明
link_function_type	Identity LogC Log CLOGLOG Logit NLOGLOG PROBIT POWER CAUCHIT	將目標值關聯至至預測值的鏈結函數。如果 target_distribution 是 Binomial 您可以使用任何列出的鏈結函數。如果 target_distribution 是您可以使用的 Multinomial CLOGLOG, CAUCHIT, LOGIT, NLOGLOG 或 PROBIT。如果 target_distribution 是 Binomial 或您可以使用的 Multinomial IDENTITY、LOG 或 POWER。
link_function_param	數字	要使用的鏈結函數參數值。唯有在 normal_link_function 或 link_function_type 為 POWER 才適用。
use_predefined_inputs	旗標	指出固定效應欄位是那些將上游定義為輸入欄位 (true) 的欄位，還是來自 fixed_effects_list (false) 的那些欄位。預設值為 false。
fixed_effects_list	結構化	如果 use_predefined_inputs 為 false，那麼指定將輸入欄位用作固定效應欄位。
use_intercept	旗標	如果為 true (預設值)，則將截取併入模型中。
random_effects_list	結構化	作為隨機效果指定的欄位清單。
regression_weight_field	欄位	用作分析加權欄位的欄位。
use_offset	None offset_value offset_field	指出如何指定偏移。值 None 表示未使用任何偏移。
offset_value	數字	用於偏移的值 (如果 use_offset 設為 offset_value)。
offset_field	欄位	用於偏移值的欄位 (如果 use_offset 設為 offset_field)。

表 130: glmmnode 內容 (繼續)

glmmnode 內容	值	內容說明
target_category_order	Ascending Descending Data	類別目標的排序。值 Data 指定使用資料中發現的排序。預設值是 Ascending。
inputs_category_order	Ascending Descending Data	類別預測值的排序。值 Data 指定使用資料中發現的排序。預設值是 Ascending。
max_iterations	整數	演算法將執行的疊代次數上限。非負值整數；預設值是 100。
confidence_level	整數	用於計算模型係數的區間估計值的信賴等級。非負整數；最大值為 100，預設值為 95。
degrees_of_freedom_method	Fixed Varied	指定如何計算自由度以進行顯著性測試。
test_fixed_effects_coefficients	Model Robust	計算參數估計值共變異矩陣的方法。
use_p_converge	旗標	參數聚合的選項。
p_converge	數字	空白，或任何正數值。
p_converge_type	Absolute Relative	
use_l_converge	旗標	對數概似聚合的選項。
l_converge	數字	空白，或任何正數值。
l_converge_type	Absolute Relative	
use_h_converge	旗標	Hessian 聚合的選項。
h_converge	數字	空白，或任何正數值。
h_converge_type	Absolute Relative	
max_fisher_steps	整數	
singularity_tolerance	數字	
use_model_name	旗標	指出是要指定模型的自訂名稱 (true) 還是使用系統產生的名稱 (false)。預設值為 false。

表 130: glmmnode 內容 (繼續)

glmmnode 內容	值	內容說明
model_name	string	如果 use_model_name 為 true, 則指定要使用的模型名稱。
confidence	onProbability onIncrease	用來計算評分信賴度值的基準; 最高預測機率, 或最高與次高預測機率之間的差異。
score_category_probabilities	旗標	如果為 true, 則為種類目標生成預測機率。預設值是 false。
max_categories	整數	如果 score_category_probabilities 為 true, 那麼指定儲存種類數目上限。
score_propensity	旗標	如果為 true, 則為旗標目標欄位生成傾向分數, 指示欄位結果為 "true" 的可能性。
emeans	structure	對於固定效果清單中的每個種類欄位, 指定是否生成估計邊際平均數。
covariance_list	structure	對於固定效果清單中的每個連續欄位, 指定計算估計邊際平均數時是使用平均數還是自訂值。
mean_scale	Original Transformed	指定是根據目標的原始尺度 (預設) 還是根據鏈結函數變換來計算估計邊際平均數。
comparison_adjustment_method	LSD SEQBONFERRONI SEQSIDAK	對多個對照執行假設測試時使用的調整方法。

gle 內容



GLE 延伸了線性模型, 以便目標可以有非常態分佈, 通過指定的連接函數與因子和共變數線性相關, 並且觀察可能相關。廣義線性混合模型涵蓋多種模式, 從非常態縱向資料的簡單線性迴歸, 到複雜的多層級模式。

表 131: gle 內容

gle 內容	值	內容說明
custom_target	旗標	指出是使用上游節點中定義的目標 (false) 或使用 target_field 指定的自訂目標 (true)。
target_field	欄位	用作目標的欄位 (如果 custom_target 為 true)。
use_trials	旗標	指出當目標回應是在試用集中發生的數個事件時, 是要使用額外欄位還是值來指定試用數目。預設值是 false。

表 131: *gle* 內容 (繼續)

gle 內容	值	內容說明
use_trials_field_or_value	Field Value	指出是使用欄位 (預設值) 還是值來指定試用數目。
trials_field	欄位	用來指定試用數目的欄位。
trials_value	整數	用來指定試用數目的值。指定的話, 最小值為 1。
use_custom_target_reference	旗標	指定是否將自訂參照種類用於類別目標。預設值是 <code>false</code> 。
target_reference_value	<i>string</i>	要使用的參照種類 (如果 <code>use_custom_target_reference</code> 為 <code>true</code>)。
dist_link_combination	NormalIdentity GammaLog PoissonLog NegbinLog TweedieIdentity NominalLogit BinomialLogit BinomialProbit BinomialLogC CUSTOM	用來分佈目標值的一般模型。 選擇 CUSTOM 可以根據 <code>target_distribution</code> 所提供的清單指定分佈。

表 131: *gle* 內容 (繼續)

gle 內容	值	內容說明
target_distribution	Normal Binomial Multinomial Gamma INVERSE_GAUSS NEG_BINOMIAL Poisson TWEEDIE UNKNOWN	當 dist_link_combination 為 Custom 時 目標值的分佈。

表 131: gle 內容 (繼續)

gle 內容	值	內容說明
link_function_type	UNKNOWN	用於使目標值與預測值相關的鏈結函數。如果 target_distribution 為 Binomial, 那麼您可以使用:
	IDENTITY	
	LOG	UNKNOWN
	LOGIT	IDENTITY
	PROBIT	LOG
	COMPL_LOG_LOG	LOGIT
	POWER	PROBIT
	LOG_COMPL	COMPL_LOG_LOG
	NEG_LOG_LOG	POWER
	ODDS_POWER	LOG_COMPL
	NEG_BINOMIAL	NEG_LOG_LOG
	GEN_LOGIT	ODDS_POWER
	CUMUL_LOGIT	如果 target_distribution 為 NEG_BINOMIAL, 那麼您可以使用:
	CUMUL_PROBIT	NEG_BINOMIAL.
	CUMUL_COMPL_LOG_LOG	如果 target_distribution 為 UNKNOWN, 那麼您可以使用:
	CUMUL_NEG_LOG_LOG	GEN_LOGIT
	CUMUL_CAUCHIT	CUMUL_LOGIT
	CUMUL_PROBIT	
	CUMUL_COMPL_LOG_LOG	
	CUMUL_NEG_LOG_LOG	
	CUMUL_CAUCHIT	
link_function_param	數字	要使用的 Tweedie 參數值。唯有在 normal_link_function 或 link_function_type 為 POWER 才適用。

表 131: gle 內容 (繼續)

gle 內容	值	內容說明
tweedie_param	數字	要使用的鏈結函數參數值。僅適用於 dist_link_combination 設定為 TweedieIdentity 或 link_function_type 為 TWEEDIE 的情況。
use_predefined_inputs	旗標	指示模型效應欄位是將上游定義為輸入欄位的欄位 (true) 還是來自 fixed_effects_list 的欄位 (false)。
model_effects_list	結構化	如果 use_predefined_inputs 為 false, 請指定要用作模型效應欄位的輸入欄位。
use_intercept	旗標	如果為 true (預設值), 則將截取併入模型中。
regression_weight_field	欄位	用作分析加權欄位的欄位。
use_offset	None Value Variable	指出如何指定偏移。值 None 表示未使用任何偏移。
offset_value	數字	用於偏移的值 (如果 use_offset 設為 offset_value)。
offset_field	欄位	用於偏移值的欄位 (如果 use_offset 設為 offset_field)。
target_category_order	Ascending Descending	類別目標的排序。預設值是 Ascending。
inputs_category_order	Ascending Descending	類別預測值的排序。預設值是 Ascending。
max_iterations	整數	演算法將執行的疊代次數上限。非負值整數; 預設值是 100。
confidence_level	數字	用於計算模型係數的區間估計值的信賴等級。非負整數; 最大值為 100, 預設值為 95。
test_fixed_effects_coef fecients	Model Robust	計算參數估計值共變異矩陣的方法。
detect_outliers	旗標	值為 true 時, 演算法將尋找除多項式分佈以外的所有分佈的影響離群值。
conduct_trend_analysis	旗標	值為 true 時, 此演算法會對散佈平面圖處理趨勢分析。

表 131: gle 內容 (繼續)

gle 內容	值	內容說明
estimation_method	FISHER_SCORING NEWTON_RAPHSON HYBRID	指定極大概似估計演算法。
max_fisher_iterations	整數	疊代數目上限 (如果使用 FISHER_SCORING estimation_method)。最小值為 0, 最大值為 20。
scale_parameter_method	MLE FIXED DEVIANCE PEARSON_CHISQUARE	指定要用於估計尺度參數的方法。
scale_value	數字	僅當 scale_parameter_method 設定為 Fixed 時可用。
negative_binomial_method	MLE FIXED	指定要用於估計負二項式輔助參數的方法。
negative_binomial_value	數字	僅當 negative_binomial_method 設定為 Fixed 時可用。
non_neg_least_squares	旗標	是否執行非負最小平方法。預設值是 false。
use_p_converge	旗標	參數聚合的選項。
p_converge	數字	空白, 或任何正數值。
p_converge_type	旗標	True = Absolute, False = Relative
use_l_converge	旗標	對數概似聚合的選項。
l_converge	數字	空白, 或任何正數值。
l_converge_type	旗標	True = Absolute, False = Relative
use_h_converge	旗標	Hessian 聚合的選項。
h_converge	數字	空白, 或任何正數值。
h_converge_type	旗標	True = Absolute, False = Relative
max_iterations	整數	演算法將執行的疊代次數上限。非負值整數; 預設值是 100。
sing_tolerance	整數	
use_model_selection	旗標	啟用參數臨界值和模型選擇方法控制項。
method	LASSO ELASTIC_NET FORWARD_STEPWISE RIDGE	確定所使用的模型選擇方法 (如果使用 Ridge, 那麼確定正規化方法)。

表 131: gle 內容 (繼續)

gle 內容	值	內容說明
detect_two_way_interactions	旗標	<p>值為 True 時，模型將自動偵測輸入欄位之間的雙向交互。</p> <p>僅當模型為僅主效應（即，使用者尚未建立任何高階效應時）並且所選 method 為逐步向前、Lasso 或彈性網路時，才應該啟用此控制項。</p>
automatic_penalty_params	旗標	<p>僅當模型選擇 method 為 Lasso 或彈性網路時才可用。</p> <p>使用此函數可以輸入與 Lasso 或彈性網路變數選擇方法關聯的罰參數。</p> <p>如果值為 True，那麼將使用預設值。如果值為 False，那麼將啟用罰參數，並且可以已輸入自訂值。</p>
lasso_penalty_param	數字	<p>僅當模型選取 method 是 Lasso 或 Elastic Net 並且 automatic_penalty_params 是 False 時才可用。為 Lasso 指定罰參數值。</p>
elastic_net_penalty_param1	數字	<p>僅當模型選取 method 是 Lasso 或 Elastic Net 並且 automatic_penalty_params 是 False 時才可用。為彈性網路參數 1 指定罰參數值。</p>
elastic_net_penalty_param2	數字	<p>僅當模型選取 method 是 Lasso 或 Elastic Net 並且 automatic_penalty_params 是 False 時才可用。為彈性網路參數 2 指定罰參數值。</p>
probability_entry	數字	<p>僅當選取的 method 是「逐步向前」時才可用。針對效應併入，指定 f 統計量準則的顯著性水準。</p>
probability_removal	數字	<p>僅當選取的 method 是「逐步向前」時才可用。針對效應移除，指定 f 統計量準則的顯著性水準。</p>
use_max_effects	旗標	<p>僅當選取的 method 是「逐步向前」時才可用。</p> <p>啟用 max_effects 控制項。</p> <p>當值為 False 時，所包含的預設效應數應等於提供給模型的總效應數減截距。</p>
max_effects	整數	<p>指定使用逐步向前建立方法時的效果數目上限。</p>
use_max_steps	旗標	<p>啟用 max_steps 控制項。</p> <p>當值為 False 時，預設步驟數應等於提供給模型的效應數的三倍減去截距。</p>

gle 內容	值	內容說明
max_steps	整數	指定使用逐步向前建立 method 時要執行的上限步驟數。
use_model_name	旗標	指出是要指定模型的自訂名稱 (true) 還是使用系統產生的名稱 (false)。預設值為 false。
model_name	string	如果 use_model_name 為 true, 則指定要使用的模型名稱。
usePI	旗標	如果值為 true, 那麼將計算預測值重要性。

kmeansnode 內容



K-Means 節點將資料集叢集到不同群組 (或叢集)。此方法將定義固定的叢集數目, 將記錄疊代分配給叢集, 以及調整叢集中心, 直到進一步優化無法再精確模型。k-means 使用稱為非監督式學習的程序來揭露輸入欄位集中的型樣, 而不是嘗試預測結果。

範例

```
node = stream.create("kmeans", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["Cholesterol", "BP", "Drug", "Na", "K",
"Age"])
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Kmeans_allinputs")
node.setPropertyValue("num_clusters", 9)
node.setPropertyValue("gen_distance", True)
node.setPropertyValue("cluster_label", "Number")
node.setPropertyValue("label_prefix", "Kmeans_")
node.setPropertyValue("optimize", "Speed")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("stop_on", "Custom")
node.setPropertyValue("max_iterations", 10)
node.setPropertyValue("tolerance", 3.0)
node.setPropertyValue("encoding_value", 0.3)
```

kmeansnode 內容	值	內容說明
inputs	[field1 ... fieldN]	K-means 模型在一系列輸入欄位上執行叢集分析, 但並不使用目標欄位。不會使用加權和頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題, 以取得更多資訊。
num_clusters	數字	
gen_distance	旗標	
cluster_label	String Number	

表 132: kmeansnode 內容 (繼續)

kmeansnode 內容	值	內容說明
label_prefix	string	
mode	Simple Expert	
stop_on	Default Custom	
max_iterations	數字	
tolerance	數字	
encoding_value	數字	
optimize	Speed Memory	用來指定是否應該最佳化模型建置以提高速度或增加記憶體。

kmeansasnode 內容



K-Means 是最常用的叢集演算法之一。它會將資料點叢集到若干預先定義的叢集中。SPSS Modeler 中的 K-Means-AS 節點在 Spark 中實作。如需 K-Means 演算法的詳細資料，請參閱 <https://spark.apache.org/docs/2.2.0/ml-clustering.html>。請注意，K-Means-AS 節點會針對類別變數自動執行 One-Hot 編碼。

表 133: kmeansasnode 內容

kmeansasnode 內容	值	內容說明
roleUse	string	指定 predefined 以使用預先定義的角色，或指定 custom 以使用自訂欄位指派。預設值為 predefined。
autoModel	布林	指定 true 以針對新產生的評分欄位使用預設名稱 (\$S-prediction)，或指定 false 以使用自訂名稱。預設值是 true。
features	欄位	當 roleUse 內容設為 custom 時，輸入的欄位名稱清單。
name	string	當 autoModel 內容設為 false 時，新產生的評分欄位的名稱。
clustersNum	整數	要建立的叢集數。預設值為 5。
initMode	string	起始設定演算法。可能的值為 k-means 或 random。預設值為 k-means 。
initSteps	整數	當 initMode 設為 k-means 時的起始設定步驟數。預設值是 2。
advancedSettings	布林	指定 true 以使下列四個內容可用。預設值是 false。

表 133: *kmeansasnode* 內容 (繼續)

kmeansasnode 內容	值	內容說明
maxIteration	整數	用於叢集作業的最大疊代數。預設值為 20。
tolerance	string	用來停止疊代的容錯。可能的設定為 1.0E-1、1.0E-2、...、1.0E-6。預設值為 1.0E-4。
setSeed	布林	指定 true 以使用自訂隨機種子。預設值是 false。
randomSeed	整數	當 setSeed 內容為 true 時的自訂隨機種子。

knnnode 內容



k-最近鄰接項 (KNN) 節點將新觀察值關聯到預測值空間中與其最鄰近的 *k* 個物件的種類或值 (其中 *k* 為整數)。相似的觀察值會彼此相鄰，相異的觀察值則會彼此相隔。

範例

```
node = stream.create("knn", "My node")
# Objectives tab
node.setPropertyValue("objective", "Custom")
# Settings tab - Neighbors panel
node.setPropertyValue("automatic_k_selection", False)
node.setPropertyValue("fixed_k", 2)
node.setPropertyValue("weight_by_importance", True)
# Settings tab - Analyze panel
node.setPropertyValue("save_distances", True)
```

表 134: *knnnode* 內容

knnnode 內容	值	內容說明
analysis	PredictTarget IdentifyNeighbors	
objective	Balance Speed Accuracy Custom	
normalize_ranges	旗標	
use_case_labels	旗標	用來啟用下一個選項的勾選框。
case_labels_field	欄位	
identify_focal_cases	旗標	用來啟用下一個選項的勾選框。
focal_cases_field	欄位	

表 134: knnnode 內容 (繼續)

knnnode 內容	值	內容說明
automatic_k_selection	旗標	
fixed_k	整數	只有當 automatic_k_selection 為 False 時才啟用。
minimum_k	整數	只有當 automatic_k_selection 為 True 時才啟用。
maximum_k	整數	
distance_computation	Euclidean CityBlock	
weight_by_importance	旗標	
range_predictions	Mean Median	
perform_feature_selection	旗標	
forced_entry_inputs	[field1 ... fieldN]	
stop_on_error_ratio	旗標	
number_to_select	整數	
minimum_change	數字	
validation_fold_assign_by_field	旗標	
number_of_folds	整數	只有當 validation_fold_assign_by_field 為 False 時才啟用
set_random_seed	旗標	
random_seed	數字	
folds_field	欄位	只有當 validation_fold_assign_by_field 為 True 時才啟用
all_probabilities	旗標	
save_distances	旗標	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	
adjusted_propensity_partition	Test Validation	

kohonenode 內容



Kohonen 節點會產生一種神經網路，此神經網路可用於將資料集叢集到各個差異群組。此網路訓練完成後，相似的記錄應在輸出對映中緊密地聚集，差異大的記錄則應彼此遠離。您可以通過查看模型區塊 中每個單位所擷取觀察的數量來找出規模較大的單元。這將讓您對叢集的相應數量有所估計。

範例

```
node = stream.create("kohonen", "My node")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Symbolic Cluster")
node.setPropertyValue("stop_on", "Time")
node.setPropertyValue("time", 1)
node.setPropertyValue("set_random_seed", True)
node.setPropertyValue("random_seed", 12345)
node.setPropertyValue("optimize", "Speed")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("width", 3)
node.setPropertyValue("length", 3)
node.setPropertyValue("decay_style", "Exponential")
node.setPropertyValue("phase1_neighborhood", 3)
node.setPropertyValue("phase1_eta", 0.5)
node.setPropertyValue("phase1_cycles", 10)
node.setPropertyValue("phase2_neighborhood", 1)
node.setPropertyValue("phase2_eta", 0.2)
node.setPropertyValue("phase2_cycles", 75)
```

表 135: kohonenode 內容

kohonenode 內容	值	內容說明
inputs	[field1 ... fieldN]	Kohonen 模型使用輸入欄位的清單，但不使用目標。不會使用頻率及加權欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
continue	旗標	
show_feedback	旗標	
stop_on	Default Time	
time	數字	
optimize	Speed Memory	用來指定是否應該最佳化模型建置以提高速度或增加記憶體。
cluster_label	旗標	
mode	Simple Expert	
width	數字	
length	數字	

表 135: kohonenode 內容 (繼續)

kohonenode 內容	值	內容說明
decay_style	Linear Exponential	
phase1_neighborhood	數字	
phase1_eta	數字	
phase1_cycles	數字	
phase2_neighborhood	數字	
phase2_eta	數字	
phase2_cycles	數字	

linearnode 內容



線性迴歸模型會根據目標與一個以上預測值之間的線性關係來預測連續目標。

範例

```
node = stream.create("linear", "My node")
# Build Options tab - Objectives panel
node.setPropertyValue("objective", "Standard")
# Build Options tab - Model Selection panel
node.setPropertyValue("model_selection", "BestSubsets")
node.setPropertyValue("criteria_best_subsets", "ASE")
# Build Options tab - Ensembles panel
node.setPropertyValue("combining_rule_categorical", "HighestMeanProbability")
```

表 136: linearnode 內容

linearnode 內容	值	內容說明
target	欄位	指定單一目標欄位。
inputs	[field1 ... fieldN]	模型所使用的預測值欄位。
continue_training_existing_model	旗標	
objective	Standard Bagging Boosting psm	psm 用於極大型資料集，且需要伺服器連線。
use_auto_data_preparation	旗標	
confidence_level	數字	

表 136: *linearnode* 內容 (繼續)

linearnode 內容	值	內容說明
model_selection	ForwardStepwise BestSubsets None	
criteria_forward_stepwise	AICC Fstatistics AdjustedRSquare ASE	
probability_entry	數字	
probability_removal	數字	
use_max_effects	旗標	
max_effects	數字	
use_max_steps	旗標	
max_steps	數字	
criteria_best_subsets	AICC AdjustedRSquare ASE	
combining_rule_continuou s	Mean Median	
component_models_n	數字	
use_random_seed	旗標	
random_seed	數字	
use_custom_model_name	旗標	
custom_model_name	<i>string</i>	
use_custom_name	旗標	
custom_name	<i>string</i>	
tooltip	<i>string</i>	
keywords	<i>string</i>	
annotation	<i>string</i>	

linearasnode 內容



線性迴歸模型會根據目標與一個以上預測值之間的線性關係來預測連續目標。

linearasnode 內容	值	內容說明
target	欄位	指定單一目標欄位。
inputs	[field1 ... fieldN]	模型所使用的預測值欄位。
weight_field	欄位	模型使用的分析欄位。
custom_fields	旗標	預設值是 TRUE。
intercept	旗標	預設值是 TRUE。
detect_2way_interaction	旗標	是否考慮雙向互動。預設值是 TRUE。
cin	數字	用於計算模型係數的估計的置信區間。指定一個大於 0 且小於 100 的值。預設值是 95。
factor_order	ascending descending	種類預測值的排序。預設值是 ascending。
var_select_method	ForwardStepwise BestSubsets none	要使用的模型選擇方法。預設值是 ForwardStepwise。
criteria_for_forward_stepwise	AICC Fstatistics AdjustedRSquare ASE	用於確定應向模型中新增效應還是應從模型中移除效應的統計資料。預設值是 AdjustedRSquare。
pin	數字	將向模型中新增具有少於此指定 pin 臨界值的最小 p 值的效應。預設值為 0.05。
pout	數字	將移除模型中具有大於此指定 pout 臨界值的 p 值的任何效應。預設值為 0.10。
use_custom_max_effects	旗標	是否在最終模型中使用上限效應數。預設值是 FALSE。
max_effects	數字	要在最終模型中使用的效果數目上限。預設值是 1。

表 137: *linearasnode* 內容 (繼續)

linearasnode 內容	值	內容說明
use_custom_max_steps	旗標	是否使用上限步驟數。預設值是 FALSE。
max_steps	數字	分步演算法停止之前的上限步驟數。預設值是 1。
criteria_for_best_subsets	AICC AdjustedRSquare ASE	要使用的準則方式。預設值是 AdjustedRSquare。

logregnode 內容



邏輯迴歸是一種統計技術，它可根據輸入欄位的值對記錄進行分類。它類似於線性迴歸，但採用種類目標欄位而非數值範圍。

多項式範例

```
node = stream.create("logreg", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["BP", "Cholesterol", "Age"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Log_reg Drug")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("method", "Stepwise")
node.setPropertyValue("logistic_procedure", "Multinomial")
node.setPropertyValue("multinomial_base_category", "BP")
node.setPropertyValue("model_type", "FullFactorial")
node.setPropertyValue("custom_terms", [["BP", "Sex"], ["Age"], ["Na", "K"]])
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("scale", "Pearson")
node.setPropertyValue("scale_value", 3.0)
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("tolerance", "1.0E-7")
# "Convergence..." section
node.setPropertyValue("max_iterations", 50)
node.setPropertyValue("max_steps", 3)
node.setPropertyValue("l_converge", "1.0E-3")
node.setPropertyValue("p_converge", "1.0E-7")
node.setPropertyValue("delta", 0.03)
# "Output..." section
node.setPropertyValue("summary", True)
node.setPropertyValue("likelihood_ratio", True)
node.setPropertyValue("asymptotic_correlation", True)
node.setPropertyValue("goodness_fit", True)
node.setPropertyValue("iteration_history", True)
node.setPropertyValue("history_steps", 3)
node.setPropertyValue("parameters", True)
node.setPropertyValue("confidence_interval", 90)
```

```

node.setPropertyValue("asymptotic_covariance", True)
node.setPropertyValue("classification_table", True)
# "Stepping" options
node.setPropertyValue("min_terms", 7)
node.setPropertyValue("use_max_terms", True)
node.setPropertyValue("max_terms", 10)
node.setPropertyValue("probability_entry", 3)
node.setPropertyValue("probability_removal", 5)
node.setPropertyValue("requirements", "Containment")

```

二項範例

```

node = stream.create("logreg", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Cholesterol")
node.setPropertyValue("inputs", ["BP", "Drug", "Age"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Log_reg Cholesterol")
node.setPropertyValue("multinomial_base_category", "BP")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("binomial_method", "Forwards")
node.setPropertyValue("logistic_procedure", "Binomial")
node.setPropertyValue("binomial_categorical_input", "Sex")
node.setKeyedPropertyValue("binomial_input_contrast", "Sex", "Simple")
node.setKeyedPropertyValue("binomial_input_category", "Sex", "Last")
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("scale", "Pearson")
node.setPropertyValue("scale_value", 3.0)
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("tolerance", "1.0E-7")
# "Convergence..." section
node.setPropertyValue("max_iterations", 50)
node.setPropertyValue("l_converge", "1.0E-3")
node.setPropertyValue("p_converge", "1.0E-7")
# "Output..." section
node.setPropertyValue("binomial_output_display", "at_each_step")
node.setPropertyValue("binomial_goodness_of_fit", True)
node.setPropertyValue("binomial_iteration_history", True)
node.setPropertyValue("binomial_parameters", True)
node.setPropertyValue("binomial_ci_enable", True)
node.setPropertyValue("binomial_ci", 85)
# "Stepping" options
node.setPropertyValue("binomial_removal_criterion", "LR")
node.setPropertyValue("binomial_probability_removal", 0.2)

```

表 138: logregnode 內容

logregnode 內容	值	內容說明
target	欄位	邏輯迴歸模型需要一個目標欄位以及一個或多個輸入欄位。不會使用頻率及加權欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
logistic_procedure	Binomial Multinomial	
include_constant	旗標	

表 138: logregnode 內容 (繼續)

logregnode 內容	值	內容說明
mode	Simple Expert	
method	Enter Stepwise Forwards Backwards BackwardsStepwise	
binomial_method	Enter Forwards Backwards	
model_type	MainEffects FullFactorial Custom	將 FullFactorial 指定為模型類型時，即使指定了執行步驟方法，執行步驟方法也不會執行。而是使用 Enter 方法。 如果將模型類型設定為 Custom，但未指定自訂欄位，則將建立主效應模型。
custom_terms	[[BP Sex][BP][Age]]	
multinomial_base_category	string	指定如何決定參照種類。
binomial_categorical_input	string	
binomial_input_contrast	Indicator Simple Difference Helmert Repeated Polynomial Deviation	種類輸入的含索引鍵內容，用於指定如何確定比對。

表 138: logregnode 內容 (繼續)

logregnode 內容	值	內容說明
binomial_input_category	First Last	這是種類輸入的含索引鍵內容，用於指定如何確定參照種類。
scale	None UserDefined Pearson Deviance	
scale_value	數字	
all_probabilities	旗標	
tolerance	1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10	
min_terms	數字	
use_max_terms	旗標	
max_terms	數字	
entry_criterion	Score LR	
removal_criterion	LR Wald	
probability_entry	數字	
probability_removal	數字	
binomial_probability_entry	數字	
binomial_probability_removal	數字	

表 138: logregnode 內容 (繼續)

logregnode 內容	值	內容說明
requirements	HierarchyDiscrete HierarchyAll Containment None	
max_iterations	數字	
max_steps	數字	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
delta	數字	
iteration_history	旗標	
history_steps	數字	
summary	旗標	
likelihood_ratio	旗標	
asymptotic_correlation	旗標	
goodness_fit	旗標	
parameters	旗標	
confidence_interval	數字	
asymptotic_covariance	旗標	
classification_table	旗標	

表 138: logregnode 內容 (繼續)

logregnode 內容	值	內容說明
stepwise_summary	旗標	
info_criteria	旗標	
monotonicity_measures	旗標	
binomial_output_display	at_each_step at_last_step	
binomial_goodness_of_fit	旗標	
binomial_parameters	旗標	
binomial_iteration_history	旗標	
binomial_classification_plots	旗標	
binomial_ci_enable	旗標	
binomial_ci	數字	
binomial_residual	outliers all	
binomial_residual_enable	旗標	
binomial_outlier_threshold	數字	
binomial_classification_cutoff	數字	
binomial_removal_criterion	LR Wald Conditional	
calculate_variable_importance	旗標	
calculate_raw_propensities	旗標	

lsvmnode 內容



通過線性支援向量機器 (LSVM) 節點，您可以將資料分為兩群組，而無需過度配適。LSVM 是線性的，並且可以與大量資料集配合使用，例如包含大量記錄的資料集。

表 139: lsvmnode 內容

lsvmnode 內容	值	內容說明
intercept	旗標	在模型中包含截距。預設值為 True。
target_order	Ascending Descending	為分類目標指定排序。對於連續目標，將忽略此項。預設值是 Ascending。
precision	數字	僅在目標欄位的測量層次為 Continuous 時使用。指定與迴歸方法損失的敏感性相關的參數。最小值為 0，而沒有最大值。預設值為 0.1。
exclude_missing_values	旗標	設置為 True 時，如果任何單一值遺漏，那麼將排除記錄。預設值是 False。
penalty_function	L1 L2	指定所使用的處罰函數的類型。預設值為 L2。
lambda	數字	罰（正規化）參數。
calculate_variable_importance	旗標	對於生成相應重要性測量的模型，此選項將顯示一張圖表來說明評估模型時每個預測值的相對重要性。請注意，對於某些模型，計算預測變數重要性（特別在處理大型資料集時）可能需要花費較長時間，因此依預設，預測變數重要性對某些模型處於關閉狀態。變數重要性不適用於決策清單模型。

neuralnetnode 內容

重要: 在此發行版中提供了具有增強功能的新版本的神經網路建模節點，並將在下一節 (*neuralnetwork*) 中進行介紹。儘管您仍然可以使用先前版本來建立模型並對其分數，但我們建議您將 Script 更新為使用新版本。這裡保留了先前版本的詳細資料以供您參照。

範例

```
node = stream.create("neuralnet", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("targets", ["Drug"])
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
# "Model" tab
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("method", "Dynamic")
node.setPropertyValue("train_pct", 30)
node.setPropertyValue("set_random_seed", True)
node.setPropertyValue("random_seed", 12345)
node.setPropertyValue("stop_on", "Time")
node.setPropertyValue("accuracy", 95)
node.setPropertyValue("cycles", 200)
node.setPropertyValue("time", 3)
node.setPropertyValue("optimize", "Speed")
# "Multiple Method Expert Options" section
```



```
node.setPropertyValue("m_topologies", "5 30 5; 2 20 3, 1 10 1")
node.setPropertyValue("m_non_pyramids", False)
node.setPropertyValue("m_persistence", 100)
```

表 140: *neuralnetnode* 內容

neuralnetnode 內容	值	內容說明
targets	[field1 ... fieldN]	「神經網路」節點需要一個或多個目標欄位以及一個或多個輸入欄位。將忽略頻率和加權欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
method	Quick Dynamic Multiple Prune ExhaustivePrune RBFN	
prevent_overtrain	旗標	
train_pct	數字	
set_random_seed	旗標	
random_seed	數字	
mode	Simple Expert	
stop_on	Default Accuracy Cycles Time	停止方式。
accuracy	數字	停止精確度。
cycles	數字	訓練循環數。
time	數字	訓練時間（分鐘數）。
continue	旗標	
show_feedback	旗標	
binary_encode	旗標	
use_last_model	旗標	
gen_logfile	旗標	

表 140: *neuralnetnode* 內容 (繼續)

neuralnetnode 內容	值	內容說明
logfile_name	<i>string</i>	
alpha	數字	
initial_eta	數字	
high_eta	數字	
low_eta	數字	
eta_decay_cycles	數字	
hid_layers	One Two Three	
hl_units_one	數字	
hl_units_two	數字	
hl_units_three	數字	
persistence	數字	
m_topologies	<i>string</i>	
m_non_pyramids	旗標	
m_persistence	數字	
p_hid_layers	One Two Three	
p_hl_units_one	數字	
p_hl_units_two	數字	
p_hl_units_three	數字	
p_persistence	數字	
p_hid_rate	數字	
p_hid_pers	數字	
p_inp_rate	數字	
p_inp_pers	數字	
p_overall_pers	數字	
r_persistence	數字	
r_num_clusters	數字	
r_eta_auto	旗標	
r_alpha	數字	

表 140: *neuralnetnode* 內容 (繼續)

neuralnetnode 內容	值	內容說明
r_eta	數字	
optimize	Speed Memory	用來指定是否應該最佳化模型建置以提高速度或增加記憶體。
calculate_variable_importance	旗標	註：此內容取代了先前版本中使用的 <i>sensitivity_analysis</i> 內容。仍然受支援舊內容，但建議使用 <i>calculate_variable_importance</i> 。
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	
adjusted_propensity_partition	Test Validation	

neuralnetworknode 內容



神經網路節點使用的模型是對人類大腦處理資訊的方式簡化了的模型。此模型通過模擬大量類似於神經元的抽象形式的互連簡單處理裝置而運作。神經網路是功能強大的一般函數估計器，只需要最少的統計或數學知識就可以對其進行訓練或套用。

範例

```
node = stream.create("neuralnetwork", "My node")
# Build Options tab - Objectives panel
node.setPropertyValue("objective", "Standard")
# Build Options tab - Ensembles panel
node.setPropertyValue("combining_rule_categorical", "HighestMeanProbability")
```

表 141: *neuralnetworknode* 內容

neuralnetworknode 內容	值	內容說明
targets	[field1 ... fieldN]	指定目標欄位。
inputs	[field1 ... fieldN]	模型所使用的預測值欄位。
splits	[field1 ... fieldN]	指定用於分割建模的一或多個欄位。
use_partition	旗標	如果已定義分割區欄位，則此選項可確保只使用訓練分割區中的資料來建置模型。
continue	旗標	繼續訓練現有模型。

表 141: *neuralnetworknode* 內容 (繼續)

neuralnetworknode 內容	值	內容說明
objective	Standard Bagging Boosting psm	psm 用於極大型資料集，且需要伺服器連線。
method	MultilayerPerceptron RadialBasisFunction	
use_custom_layers	旗標	
first_layer_units	數字	
second_layer_units	數字	
use_max_time	旗標	
max_time	數字	
use_max_cycles	旗標	
max_cycles	數字	
use_min_accuracy	旗標	
min_accuracy	數字	
combining_rule_categorical	Voting HighestProbability HighestMeanProbability	
combining_rule_continuou s	Mean Median	
component_models_n	數字	
overfit_prevention_pct	數字	
use_random_seed	旗標	
random_seed	數字	
missing_values	listwiseDeletion missingValueImputation	
use_model_name	<i>boolean</i>	
model_name	<i>string</i>	

表 141: *neuralnetworknode* 內容 (繼續)

neuralnetworknode 內容	值	內容說明
confidence	onProbability onIncrease	
score_category_probabilities	旗標	
max_categories	數字	
score_propensity	旗標	
use_custom_name	旗標	
custom_name	string	
tooltip	string	
keywords	string	
annotation	string	

questnode 內容



QUEST 節點可提供用於建立決策樹狀結構的二元分類法，此方法的設計目的是減少大型 C&R 樹狀結構分析所需的處理時間，同時也減少在分類樹狀結構方法中發現的趨勢以便偏愛容許有多個分割的輸入。輸入欄位可以是數值範圍（連續），但目標欄位必須是種類。所有分割都是二項式。

範例

```
node = stream.create("quest", "My node")
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
node.setPropertyValue("model_output_type", "InteractiveBuilder")
node.setPropertyValue("use_tree_directives", True)
node.setPropertyValue("max_surrogates", 5)
node.setPropertyValue("split_alpha", 0.03)
node.setPropertyValue("use_percentage", False)
node.setPropertyValue("min_parent_records_abs", 40)
node.setPropertyValue("min_child_records_abs", 30)
node.setPropertyValue("prune_tree", True)
node.setPropertyValue("use_std_err", True)
node.setPropertyValue("std_err_multiplier", 3)
```

表 142: *questnode* 內容

questnode 內容	值	內容說明
target	欄位	QUEST 模型需要一個目標欄位以及一個或多個輸入欄位。還可以指定頻率欄位。請參閱第 191 頁的『 一般建模節點內容 』主題，以取得更多資訊。
continue_training_existing_model	旗標	

表 142: questnode 內容 (繼續)

questnode 內容	值	內容說明
objective	Standard Boosting Bagging psm	psm 用於極大型資料集，且需要伺服器連線。
model_output_type	Single InteractiveBuilder	
use_tree_directives	旗標	
tree_directives	string	
use_max_depth	Default Custom	
max_depth	整數	樹狀結構深度上限，從 0 到 1000。僅在 use_max_depth = Custom 時使用。
prune_tree	旗標	刪改樹狀結構以避免過度適合。
use_std_err	旗標	在風險中（在標準錯誤中）使用最大差異。
std_err_multiplier	數字	最大差異。
max_surrogates	數字	最大代理數。
use_percentage	旗標	
min_parent_records_pc	數字	
min_child_records_pc	數字	
min_parent_records_abs	數字	
min_child_records_abs	數字	
use_costs	旗標	
costs	結構化	結構化內容。
priors	Data Equal Custom	
custom_priors	結構化	結構化內容。
adjust_priors	旗標	
trails	數字	推升法或自助重抽總合法的成分模型數目。

表 142: questnode 內容 (繼續)

questnode 內容	值	內容說明
set_ensemble_method	Voting HighestProbability HighestMeanProbability	類別目標的預設結合規則。
range_ensemble_method	Mean Median	連續目標的預設結合規則。
large_boost	旗標	將推升法套用至極大型資料集。
split_alpha	數字	用於分割的顯著性層級。
train_pct	數字	過度適合預防集。
set_random_seed	旗標	抄寫結果選項。
seed	數字	
calculate_variable_importance	旗標	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	
adjusted_propensity_partition	Test Validation	

randomtrees 內容



「隨機樹狀結構」節點類似於現有的 C&RT 節點；不過，「隨機樹狀結構」節點設計為用於處理大型資料以建立單一樹狀結構，並會在 SPSS Modeler 第 17 版中新增的輸出檢視器中顯示產生的模型。「隨機樹狀結構」節點將產生您可以對未來觀察進行預測或分類的決策樹狀結構。通過在每一個步驟最大限度降低不純潔度，此方法使用遞歸分區將訓練記錄分割為多個區段。如果樹狀結構中某個節點的全部觀察值都的目標欄位的一個特定種類，那麼系統會將該節點視為純潔。目標欄位和輸入欄位可以是數值範圍或類別（標稱、序數或旗標）欄位；所有分割都是二元的（只有兩個子群組）。

表 143: randomtrees 內容

randomtrees 內容	值	內容說明
target	欄位	在「隨機樹狀結構」節點中，模型需要單個目標欄位以及一個或多個輸入欄位。還可以指定頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
number_of_models	整數	確定要在總體建模過程中建立的模型個數。

表 143: *randomtrees* 內容 (繼續)

randomtrees 內容	值	內容說明
use_number_of_predictors	旗標	確定是否使用 number_of_predictors。
number_of_predictors	整數	指定在建立分割模型時要使用的預測值數目。
use_stop_rule_for_accuracy	旗標	確定在精確度無法提高時是否停止模型建置。
sample_size	數字	減小此值可以提高處理超大資料集時的效能。
handle_imbalanced_data	旗標	如果模型的目標是特定的旗標結果，並且所需結果與非所需結果的比例非常小，那麼資料是不平衡資料並且模型所處理的自我重複採樣可能會影響模型精確性。啟用不平衡資料處理，以便模型將擷取所需結果中的更大比例部分並產生更強大的模型。
use_weighted_sampling	旗標	為 <i>False</i> 時，每個節點的變數將以同一機率隨機進行選取。為 <i>True</i> 時，會對變數進行相應地加權和選取。
max_node_number	整數	容許各個樹狀結構中存在的節點數目上限。如果下一次分割時將超過此數字，那麼樹狀結構成長將停止。
max_depth	整數	樹狀結構成長停止之前的樹狀結構深度上限。
min_child_node_size	整數	確定分割母節點之後容許子節點中存在的下限記錄數。如果子節點包含的記錄數少於此處指定的數目，那麼不會分割母節點
use_costs	旗標	
costs	結構化	結構化內容。格式是 3 個值的清單：實際值、預測值及成本（如果該預測錯誤）。例如： <code>tree.setPropertyValue("costs", [["drugA", "drugB", 3.0], ["drugX", "drugY", 4.0]])</code>
default_cost_increase	none linear square custom	註：僅對序數目標啟用。 在成本矩陣中設定預設值。
max_pct_missing	整數	如果任何輸入中遺漏值所佔百分比超過此處指定的值，那麼將排除此輸入。下限 0，上限 100。

表 143: *randomtrees* 內容 (繼續)

randomtrees 內容	值	內容說明
exclude_single_cat_pct	整數	如果某個種類值代表的記錄百分比高於此處指定的值，那麼將從模型建置中排除整個欄位。最小值為 1，最大值為 99。
max_category_number	整數	如果欄位中種類數超過了此值，那麼將從模型建立中排除此欄位。最小值為 2。
min_field_variation	數字	如果某個連續欄位的變異係數小於此值，那麼將從模型建置中排除此欄位。
num_bins	整數	僅在資料由連續輸入組成的情況下使用。設定要用於輸入的相等頻率「分組」數；選項為 2、4、5、10、20、25、50 或 100。
topN	整數	指定要報告的規則數目。預設值為 50，最小值為 1，最大值為 1000。

regressionnode 內容



線性迴歸是一種一般統計技術，用於透過配適直線或表面來彙總資料並進行預測，從而將預測輸出值與實際輸出值之間的差異最小化。

註: 在未來的發行版中，「迴歸方法」節點將取代之為「線性」節點。我們建議從現在啟動針對線性迴歸使用線性模型。

範例

```
node = stream.create("regression", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Age")
node.setPropertyValue("inputs", ["Na", "K"])
node.setPropertyValue("partition", "Test")
node.setPropertyValue("use_weight", True)
node.setPropertyValue("weight_field", "Drug")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Regression Age")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("method", "Stepwise")
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("complete_records", False)
node.setPropertyValue("tolerance", "1.0E-3")
# "Stepping..." section
node.setPropertyValue("stepping_method", "Probability")
node.setPropertyValue("probability_entry", 0.77)
node.setPropertyValue("probability_removal", 0.88)
node.setPropertyValue("F_value_entry", 7.0)
node.setPropertyValue("F_value_removal", 8.0)
# "Output..." section
node.setPropertyValue("model_fit", True)
node.setPropertyValue("r_squared_change", True)
```

```

node.setPropertyValue("selection_criteria", True)
node.setPropertyValue("descriptives", True)
node.setPropertyValue("p_correlations", True)
node.setPropertyValue("collinearity_diagnostics", True)
node.setPropertyValue("confidence_interval", True)
node.setPropertyValue("covariance_matrix", True)
node.setPropertyValue("durbin_watson", True)

```

表 144: regressionnode 內容

regressionnode 內容	值	內容說明
target	欄位	迴歸模型需要一個目標欄位以及一個或多個輸入欄位。還可以指定加權欄位。請參閱第 191 頁的『 一般建模節點內容 』主題，以取得更多資訊。
method	Enter Stepwise Backwards Forwards	
include_constant	旗標	
use_weight	旗標	
weight_field	欄位	
mode	Simple Expert	
complete_records	旗標	

表 144: regressionnode 內容 (繼續)

regressionnode 內容	值	內容說明
tolerance	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10 1.0E-11 1.0E-12	請使用雙引號將引數括起來。
stepping_method	useP useF	useP : 使用 F 機率值 useF: 使用 F 值
probability_entry	數字	
probability_removal	數字	
F_value_entry	數字	
F_value_removal	數字	
selection_criteria	旗標	
confidence_interval	旗標	
covariance_matrix	旗標	
collinearity_diagnostics	旗標	
regression_coefficients	旗標	
exclude_fields	旗標	
durbin_watson	旗標	
model_fit	旗標	
r_squared_change	旗標	
p_correlations	旗標	

表 144: regressionnode 內容 (繼續)

regressionnode 內容	值	內容說明
descriptives	旗標	
calculate_variable_importance	旗標	

sequencenode 內容



序列節點可探索循序資料或與時間有關的資料中的相關規則。序列是一系列可能會以可預測順序發生的項目集合。例如，一個購買了剃刀和須後水的顧客可能在下次購物時購買剃須膏。「序列」節點基於 CARMA 關聯規則演算法，使用有效的兩段式方法來尋找序列。

範例

```
node = stream.create("sequence", "My node")
# "Fields" tab
node.setPropertyValue("id_field", "Age")
node.setPropertyValue("contiguous", True)
node.setPropertyValue("use_time_field", True)
node.setPropertyValue("time_field", "Date1")
node.setPropertyValue("content_fields", ["Drug", "BP"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Sequence_test")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("min_supp", 15.0)
node.setPropertyValue("min_conf", 14.0)
node.setPropertyValue("max_size", 7)
node.setPropertyValue("max_predictions", 5)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("use_max_duration", True)
node.setPropertyValue("max_duration", 3.0)
node.setPropertyValue("use_pruning", True)
node.setPropertyValue("pruning_value", 4.0)
node.setPropertyValue("set_mem_sequences", True)
node.setPropertyValue("mem_sequences", 5.0)
node.setPropertyValue("use_gaps", True)
node.setPropertyValue("min_item_gap", 20.0)
node.setPropertyValue("max_item_gap", 30.0)
```

表 145: sequencenode 內容

sequencenode 內容	值	內容說明
id_field	欄位	若要建立「序列」模型，您需要指定一個 ID 欄位以及一個選用的時間欄位，以及一個或多個內容欄位。不會使用加權和頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
time_field	欄位	
use_time_field	旗標	
content_fields	[field1 ... fieldn]	

表 145: sequencenode 內容 (繼續)

sequencenode 內容	值	內容說明
contiguous	旗標	
min_supp	數字	
min_conf	數字	
max_size	數字	
max_predictions	數字	
mode	Simple Expert	
use_max_duration	旗標	
max_duration	數字	
use_gaps	旗標	
min_item_gap	數字	
max_item_gap	數字	
use_pruning	旗標	
pruning_value	數字	
set_mem_sequences	旗標	
mem_sequences	整數	

slrmnode 內容



自我學習回應模型 (SLRM) 節點可用於建置一個模型，其中單一新觀察值或少量新觀察值可用來重新評估模型，而無需使用所有資料重新訓練模型。

範例

```
node = stream.create("slrm", "My node")
node.setPropertyValue("target", "Offer")
node.setPropertyValue("target_response", "Response")
node.setPropertyValue("inputs", ["Cust_ID", "Age", "Ave_Bal"])
```

表 146: slrmnode 內容

slrmnode 內容	值	內容說明
target	欄位	目標欄位必須是列名欄位或旗標欄位。還可以指定頻率欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
target_response	欄位	類型必須是旗標。
continue_training_existing_model	旗標	

表 146: slrmnode 內容 (繼續)

slrmnode 內容	值	內容說明
target_field_values	旗標	全部使用：使用來自來源的全部值。 指定：選取所需的值。
target_field_values_specify	[field1 ... fieldn]	
include_model_assessment	旗標	
model_assessment_random_seed	數字	必須是實數。
model_assessment_sample_size	數字	必須是實數。
model_assessment_iterations	數字	疊代次數。
display_model_evaluation	旗標	
max_predictions	數字	
randomization	數字	
scoring_random_seed	數字	
sort	Ascending Descending	指定先顯示評分最高的供應項目還是評分最低的供應項目。
model_reliability	旗標	
calculate_variable_importance	旗標	

statisticsmodelnode 內容



Statistics 模型節點使您能夠通過執行生成 PMML 的 IBM SPSS Statistics 程序分析和處理資料。此節點需要 IBM SPSS Statistics 的授權副本。

有關此節點內容的資訊，請參閱第 380 頁的『statisticsmodelnode 內容』。

stpnode 內容



空間-時間預測 (STP) 節點使用包含位置資料、預測輸入欄位 (預測值)、時間欄位和目標欄位的資料。每個位置有數個資料列，用來代表在每次測量時每個預測工具的值。分析資料之後，它可用來在分析中所使用形狀資料內的任何位置，預測目標值。

表 147: stpnode 內容

stpnode 內容	資料類型	內容說明
欄位標籤		
target	欄位	此為目標欄位。

表 147: *stpnode* 內容 (繼續)

stpnode 內容	資料類型	內容說明
location	欄位	模型的位置欄位。 只容許使用地理空間欄位。
location_label	欄位	這是要在輸出中使用的種類欄位，用於標註 location 中選擇的位置。
time_field	欄位	模型的時間欄位。 只容許使用具有連續測量的欄位，並且儲存類型必須為時間、日期、時間戳記或整數。
inputs	<i>[field1 ... fieldn]</i>	輸入欄位的清單。
時間間隔標籤		
interval_type_timestamp	Years Quarters Months Weeks Days Hours Minutes Seconds	
interval_type_date	Years Quarters Months Weeks Days	
interval_type_time	Hours Minutes Seconds	限制建立 STP 用於計算的時間指標時考慮的每週的天數。
interval_type_integer	Periods (僅限時間指標欄位，整數儲存)	資料集所要轉換為的時間間隔。 可用的選擇取決於選擇作為模型 <i>time_field</i> 的欄位的儲存類型。

表 147: stpnode 內容 (繼續)

stpnode 內容	資料類型	內容說明
period_start	整數	
start_month	January February March April May June July August September October November December	模型將開始檢索的月份（例如，如果設為 March，但資料集中的第一筆記錄是 January，則模型將跳過前兩筆記錄，並在三月開始檢索。）。
week_begins_on	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	STP 根據資料建立的時間索引的起點。
days_per_week	整數	最小值為 1，最大值為 7，增量為 1。
hours_per_day	整數	模型在某天中佔用的小時數。如果此項設定為 10，那麼模型將從 day_begins_at 時間開始編製索引並持續 10 個小時，然後跳至與 day_begins_at 值相符的下一個值等等。

表 147: stpnode 內容 (繼續)

stpnode 內容	資料類型	內容說明
day_begins_at	00:00 01:00 02:00 03:00 ... 23:00	設定模型開始編製索引的起始小時值。
interval_increment	1 2 3 4 5 6 10 12 15 20 30	此增量設定針對分鐘或秒。此項目決定了模型根據資料建立指標的位置。因此，如果增量為 30 且間隔類型為 seconds，則模型將每 30 秒從資料建立一次索引。
data_matches_interval	布林	如果設定為 N，那麼從資料到一般 interval_type 的轉換在建立模型前進行。 如果資料已具有正確的格式，並且 interval_type 以及任何相關設定與資料相符，那麼將此屬性設定為 Y 可以阻止轉換或聚集資料。 將此屬性設定為 Y 將取消所有聚集控制項。

表 147: stpnode 內容 (繼續)

stpnode 內容	資料類型	內容說明
agg_range_default	Sum Mean Min Max Median 1stQuartile 3rdQuartile	此項目決定用於連續欄位的預設聚集方法。未明確包含在自訂聚集中的所有連續欄位將使用此處指定的方法進行聚集。
custom_agg	[[field, aggregation method],[..] 展示: [['x5' 'FirstQuartile'] ['x4' 'Sum']]	結構化內容: Script 參數: custom_agg 例如: set :stpnode.custom_agg = [[field1 function] [field2 function]] 其中 function 是要與該欄位配合使用的總計函數。
基本標籤		
include_intercept	旗標	
max_autoregressive_lag	整數	最小值 1, 最大值 5, 增量為 1。這是預測所需的先前記錄數。因此, 如果設定為 5, 那麼將使用先前的 5 條記錄建立新預測。此處根據建立資料指定的記錄數將合併到模型中, 因此使用者無需在對模型進行評分時再次提供資料。
estimation_method	Parametric Nonparametric	用於對空間共變異矩陣進行建模的方法。

表 147: stpnode 內容 (繼續)

stpnode 內容	資料類型	內容說明
parametric_model	Gaussian Exponential PoweredExponential	Parametric 空間共變異模型的階數參數。
exponential_power	數字	PoweredExponential 模型的冪次層次。最小值為 1, 最大值為 2。
進階標籤		
max_missing_values	整數	模型中容許具有遺漏值的記錄所佔的上限百分比。
significance	數字	模型建立中的假設測試的顯著性水準。指定 STP 模型估計值中所有測試 (包含兩項適合度測試、作用 F 測試及係數 T 測試) 的顯著性值。
輸出標籤		
model_specifications	旗標	
temporal_summary	旗標	
location_summary	旗標	決定「位置摘要」表格是否併入在模型輸出中。
model_quality	旗標	
test_mean_structure	旗標	
mean_structure_coefficients	旗標	
autoregressive_coefficients	旗標	
test_decay_space	旗標	
parametric_spatial_covariance	旗標	
correlations_heat_map	旗標	
correlations_map	旗標	
location_clusters	旗標	
similarity_threshold	數字	這是一個臨界值, 達到此臨界值後, 認為輸出叢集足夠相似, 應合併為單個叢集。
max_number_clusters	整數	可以併入在模型輸出中的叢集數目的上限。
模型選項標籤		
use_model_name	旗標	
model_name	string	

表 147: *stpnod*e 內容 (繼續)

stpnode 內容	資料類型	內容說明
uncertainty_factor	數字	最小值為 0，最大值為 100。決定套用至未來預測的不確定性（錯誤）增量。它是預測的上限和下限。

svmnode 內容



使用支援向量機器 (SVM) 節點，可以將資料分為兩群組，而無需過度配適。SVM 可以與大量資料集配合使用，例如那些含有大量輸入欄位的資料集。

範例

```
node = stream.create("svm", "My node")
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("kernel", "Polynomial")
node.setPropertyValue("gamma", 1.5)
```

表 148: *svmnode* 內容

svmnode 內容	值	內容說明
all_probabilities	旗標	
stopping_criteria	1.0E-1 1.0E-2 1.0E-3 (default) 1.0E-4 1.0E-5 1.0E-6	決定停止最佳化演算法的時間。
regularization	數字	也稱為 C 參數。
precision	數字	僅在目標欄位的測量層次為 Continuous 時使用。
kernel	RBF (預設) Polynomial Sigmoid Linear	用於轉換的核心函數的類型。
rbf_gamma	數字	僅在 kernel 為 RBF 時使用。

表 148: svmnode 內容 (繼續)

svmnode 內容	值	內容說明
gamma	數字	僅在 kernel 為 Polynomial 或 Sigmoid 時使用。
bias	數字	
degree	數字	僅在 kernel 為 Polynomial 時使用。
calculate_variable_importance	旗標	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	
adjusted_propensity_partition	Test Validation	

tcmnode 內容



時間原因建模嘗試在時間序列資料中探索主要原因關係。在時間原因建模中，您可以指定一組目標序列以及這些目標的一組候選輸入。然後，此程序為每個目標建立自身迴歸「時間序列」模型，並且僅包含那些與目標的因果關係最為重要的輸入。

表 149: tcmnode 內容

tcmnode 內容	值	內容說明
custom_fields	布林	
dimensionlist	[dimension1 ... dimensionN]	
data_struct	Multiple Single	
metric_fields	欄位	
both_target_and_input	[f1 ... fN]	
targets	[f1 ... fN]	
candidate_inputs	[f1 ... fN]	
forced_inputs	[f1 ... fN]	
use_timestamp	Timestamp Period	

表 149: tcmnode 內容 (繼續)

tcmnode 內容	值	內容說明
input_interval	None Unknown Year Quarter Month Week Day Hour Hour_nonperiod Minute Minute_nonperiod Second Second_nonperiod	
period_field	<i>string</i>	
period_start_value	整數	
num_days_per_week	整數	
start_day_of_week	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	
num_hours_per_day	整數	
start_hour_of_day	整數	
timestamp_increments	整數	

表 149: tcmnode 內容 (繼續)

tcmnode 內容	值	內容說明
cyclic_increments	整數	
cyclic_periods	清單	
output_interval	None Year Quarter Month Week Day Hour Minute Second	
is_same_interval	Same Notsame	
cross_hour	布林	
aggregate_and_distribute	清單	
aggregate_default	Mean Sum Mode Min Max	
distribute_default	Mean Sum	

表 149: tcmnode 內容 (繼續)

tcmnode 內容	值	內容說明
group_default	Mean Sum Mode Min Max	
missing_imput	Linear_interp Series_mean K_mean K_meridian Linear_trend None	
k_mean_param	整數	
k_median_param	整數	
missing_value_threshold	整數	
conf_level	整數	
max_num_predictor	整數	
max_lag	整數	
epsilon	數字	
threshold	整數	
is_re_est	布林	
num_targets	整數	
percent_targets	整數	
fields_display	清單	
series_display	清單	
network_graph_for_target	布林	
sign_level_for_target	數字	
fit_and_outlier_for_target	布林	
sum_and_para_for_target	布林	
impact_diag_for_target	布林	

表 149: tcmnode 內容 (繼續)

tcmnode 內容	值	內容說明
impact_diag_type_for_target	Effect Cause Both	
impact_diag_level_for_target	整數	
series_plot_for_target	布林	
res_plot_for_target	布林	
top_input_for_target	布林	
forecast_table_for_target	布林	
same_as_for_target	布林	
network_graph_for_series	布林	
sign_level_for_series	數字	
fit_and_outlier_for_series	布林	
sum_and_para_for_series	布林	
impact_diagram_for_series	布林	
impact_diagram_type_for_series	Effect Cause Both	
impact_diagram_level_for_series	整數	
series_plot_for_series	布林	
residual_plot_for_series	布林	
forecast_table_for_series	布林	
outlier_root_cause_analysis	布林	
causal_levels	整數	
outlier_table	Interactive Pivot Both	
rmisp_error	布林	

表 149: tcmnode 內容 (繼續)

tcmnode 內容	值	內容說明
bic	布林	
r_square	布林	
outliers_over_time	布林	
series_transormation	布林	
use_estimation_period	布林	
estimation_period	Times Observation	
observations	清單	
observations_type	Latest Earliest	
observations_num	整數	
observations_exclude	整數	
extend_records_into_future	布林	
forecastperiods	整數	
max_num_distinct_values	整數	
display_targets	FIXEDNUMBER PERCENTAGE	
goodness_fit_measure	ROOTMEAN BIC RSQUARE	
top_input_for_series	布林	
aic	布林	
rmse	布林	

ts 內容



「時間序列」節點會為時間序列資料評估指數平滑化、單變數自身迴歸整合移動平均 (ARIMA)，以及多變數 ARIMA（或稱轉換函數）模型，然後產生未來效能的預測。此「時間序列」節點類似於 SPSS Modeler 第 18 版中已淘汰使用的先前「時間序列」節點。但是，此較新「時間序列」節點旨在利用 IBM SPSS Analytic Server 的能力來處理大資料，並在 SPSS Modeler 第 17 版中新增的輸出檢視器中顯示生成的模型。

表 150: ts 內容

ts 內容	值	內容說明
targets	欄位	「時間序列」節點會預測一或多個目標，選擇性地使用一或多個輸入欄位作為預測值。不會使用頻率及加權欄位。請參閱第 191 頁的『一般建模節點內容』主題，以取得更多資訊。
candidate_inputs	[<i>field1 ... fieldN</i>]	模型所用的輸入或預測值欄位。
use_period	旗標	
date_time_field	欄位	
input_interval	None Unknown Year Quarter Month Week Day Hour Hour_nonperiod Minute Minute_nonperiod Second Second_nonperiod	
period_field	欄位	
period_start_value	整數	
num_days_per_week	整數	

表 150: ts 內容 (繼續)

ts 內容	值	內容說明
start_day_of_week	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	
num_hours_per_day	整數	
start_hour_of_day	整數	
timestamp_increments	整數	
cyclic_increments	整數	
cyclic_periods	清單	
output_interval	None Year Quarter Month Week Day Hour Minute Second	
is_same_interval	旗標	
cross_hour	旗標	
aggregate_and_distribute	清單	

表 150: ts 內容 (繼續)

ts 內容	值	內容說明
aggregate_default	Mean Sum Mode Min Max	
distribute_default	Mean Sum	
group_default	Mean Sum Mode Min Max	
missing_imput	Linear_interp Series_mean K_mean K_median Linear_trend	
k_span_points	整數	
use_estimation_period	旗標	
estimation_period	Observations Times	
date_estimation	清單	僅當使用 date_time_field 時可用
period_estimation	清單	僅當使用 use_period 時可用
observations_type	Latest Earliest	

表 150: ts 內容 (繼續)

ts 內容	值	內容說明
observations_num	整數	
observations_exclude	整數	
method	ExpertModeler Exsmooth Arima	
expert_modeler_method	ExpertModeler Exsmooth Arima	
consider_seasonal	旗標	
detect_outliers	旗標	
expert_outlier_additive	旗標	
expert_outlier_level_shift	旗標	
expert_outlier_innovational	旗標	
expert_outlier_level_shift	旗標	
expert_outlier_transient	旗標	
expert_outlier_seasonal_additive	旗標	
expert_outlier_local_trend	旗標	
expert_outlier_additive_patch	旗標	
consider_newesmodels	旗標	

表 150: ts 內容 (繼續)

ts 內容	值	內容說明
exsmooth_model_type	Simple HoltLinearTrend BrownLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative DampedTrendAdditive DampedTrendMultiplicative MultiplicativeTrendAdditive MultiplicativeSeasonal MultiplicativeTrendMultiplicative MultiplicativeTrend	指定「指數平滑化」方法。預設值為 Simple。

表 150: ts 內容 (繼續)

ts 內容	值	內容說明
futureValue_type_method	Compute specify	<p>如果使用 Compute, 則系統會針對每一個預測值計算預測期間的「未來值」。</p> <p>針對每一個預測值, 您可以從函數清單中選擇 (空白、最近點的平均數、最近值) 或使用 specify 來手動輸入值。若要指定個別欄位及內容, 請使用 extend_metric_values 內容。例如:</p> <pre>set :ts.futureValue_type_method="specify" set :ts.extend_metric_values=[{'Market_1', 'USER_SPECIFY', [1,2,3]}, {'Market_2', 'MOST_RECENT_VALUE', ''}, {'Market_3', 'RECENT_POINTS_MEAN', ''}]</pre>
exsmooth_transformation_type	None SquareRoot NaturalLog	
arma.p	整數	
arma.d	整數	
arma.q	整數	
arma.sp	整數	
arma.sd	整數	
arma.sq	整數	
arma_transformation_type	None SquareRoot NaturalLog	
arma_include_constant	旗標	
tf_arma.p.fieldname	整數	用於轉換函數。
tf_arma.d.fieldname	整數	用於轉換函數。
tf_arma.q.fieldname	整數	用於轉換函數。
tf_arma.sp.fieldname	整數	用於轉換函數。
tf_arma.sd.fieldname	整數	用於轉換函數。

表 150: ts 內容 (繼續)

ts 內容	值	內容說明
tf_arma.sq. <i>fieldname</i>	整數	用於轉換函數。
tf_arma.delay. <i>fieldname</i>	整數	用於轉換函數。
tf_arma.transformation_type. <i>fieldname</i>	None SquareRoot NaturalLog	用於轉換函數。
arma_detect_outliers	旗標	
arma_outlier_additive	旗標	
arma_outlier_level_shift	旗標	
arma_outlier_innovational	旗標	
arma_outlier_transient	旗標	
arma_outlier_seasonal_additive	旗標	
arma_outlier_local_trend	旗標	
arma_outlier_additive_patch	旗標	
max_lags	整數	
cal_PI	旗標	
conf_limit_pct	<i>real</i>	
events	欄位	
continue	旗標	
scoring_model_only	旗標	用於時間序列數目非常大 (數萬個) 的模型。
forecastperiods	整數	
extend_records_into_future	旗標	
extend_metric_values	欄位	容許您為預測值提供未來 值。
conf_limits	旗標	
noise_res	旗標	
max_models_output	整數	用於控制項輸出中顯示的 模型的數量。預設值為 10。如果建立的模型總 數超過此值，那麼模型不 會顯示在輸出。模型仍可 用於評分。

timeseriesnode 內容 (已淘汰)



註: 此原始「時間序列」節點在 SPSS Modeler 第 18 版中已已淘汰使用, 並且取代為旨在利用 IBM SPSS Analytic Server 的能力並處理大資料的新的「時間序列」節點。

「時間序列」節點會為時間序列資料評估指數平滑化、單變數自身迴歸整合移動平均 (ARIMA), 以及多變數 ARIMA (或稱轉換函數) 模型, 然後產生未來效能的預測。在時間序列節點之前必須有時間間隔節點。

範例

```
node = stream.create("timeseries", "My node")
node.setPropertyValue("method", "Exsmooth")
node.setPropertyValue("exsmooth_model_type", "HoltsLinearTrend")
node.setPropertyValue("exsmooth_transformation_type", "None")
```

表 151: timeseriesnode 內容

timeseriesnode 內容	值	內容說明
targets	欄位	「時間序列」節點會預測一或多個目標, 選擇性地使用一或多個輸入欄位作為預測值。不會使用頻率及加權欄位。請參閱第 191 頁的『一般建模節點內容』主題, 以取得更多資訊。
continue	旗標	
method	ExpertModeler Exsmooth Arima Reuse	
expert_modeler_method	旗標	
consider_seasonal	旗標	
detect_outliers	旗標	
expert_outlier_additive	旗標	
expert_outlier_level_shift	旗標	
expert_outlier_innovational	旗標	
expert_outlier_level_shift	旗標	
expert_outlier_transient	旗標	
expert_outlier_seasonal_additive	旗標	
expert_outlier_local_trend	旗標	
expert_outlier_additive_patch	旗標	

表 151: *timeseriesnode* 內容 (繼續)

timeseriesnode 內容	值	內容說明
<i>exsmooth_model_type</i>	Simple HoltLinearTrend BrownLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
<i>exsmooth_transformation_type</i>	None SquareRoot NaturalLog	
<i>arma_p</i>	整數	
<i>arma_d</i>	整數	
<i>arma_q</i>	整數	
<i>arma_sp</i>	整數	
<i>arma_sd</i>	整數	
<i>arma_sq</i>	整數	
<i>arma_transformation_type</i>	None SquareRoot NaturalLog	
<i>arma_include_constant</i>	旗標	
<i>tf_arma_p.fieldname</i>	整數	用於轉換函數。
<i>tf_arma_d.fieldname</i>	整數	用於轉換函數。
<i>tf_arma_q.fieldname</i>	整數	用於轉換函數。
<i>tf_arma_sp.fieldname</i>	整數	用於轉換函數。
<i>tf_arma_sd.fieldname</i>	整數	用於轉換函數。
<i>tf_arma_sq.fieldname</i>	整數	用於轉換函數。
<i>tf_arma_delay.fieldname</i>	整數	用於轉換函數。

表 151: timeseriesnode 內容 (繼續)

timeseriesnode 內容	值	內容說明
tf_arma_transformation_type. fieldname	None SquareRoot NaturalLog	用於轉換函數。
arma_detect_outlier_mode	None Automatic	
arma_outlier_additive	旗標	
arma_outlier_level_shift	旗標	
arma_outlier_innovational	旗標	
arma_outlier_transient	旗標	
arma_outlier_seasonal_additive	旗標	
arma_outlier_local_trend	旗標	
arma_outlier_additive_patch	旗標	
conf_limit_pct	real	
max_lags	整數	
events	欄位	
scoring_model_only	旗標	用於時間序列數目非常大 (數萬個) 的模型。

treeas 內容



樹狀結構 AS 節點類似於現有的 CHAID 節點；但是，「樹狀結構 AS」節點旨在處理大量資料以建立單一樹狀結構，並在 SPSS Modeler 第 17 版中新增的輸出檢視器中顯示生成的模型。此節點通過使用卡方統計量 (CHAID) 來識別最佳化分割，從而產生決策樹狀結構。對 CHAID 的這一使用可產生非二元樹狀結構，意味著某些分割將具有兩個以上的分支。目標欄位和輸入欄位可以是數值範圍 (連續) 或類別欄位。「詳盡的 CHAID」是對 CHAID 的修改，它會徹底檢查所有可能的分割，但計算時間較長。

表 152: treeas 內容

treeas 內容	值	內容說明
target	欄位	在 Tree-AS 節點中，CHAID 模型需要單個目標和一個或多個輸入欄位。還可以指定頻率欄位。請參閱第 191 頁的『 一般建模節點內容 』主題，以取得更多資訊。
method	chaid exhaustive_chaid	
max_depth	整數	樹狀結構深度上限 (從 0 到 20)。預設值為 5。

表 152: *treeas* 內容 (繼續)

treeas 內容	值	內容說明
num_bins	整數	僅在資料由連續輸入組成的情況下使用。設定要用於輸入的相等頻率「分組」數；選項為 2、4、5、10、20、25、50 或 100。
record_threshold	整數	記錄數，在達到此數量的情況下在建立樹狀結構時模型將從使用 p 值切換為作用大小。預設值是 1,000,000；請按增量 10,000 將此增大或減小。
split_alpha	數字	用於分割的顯著性層級。值必須介於 0.01 到 0.99 之間。
merge_alpha	數字	用於合併的顯著性層級。值必須介於 0.01 到 0.99 之間。
bonferroni_adjustment	旗標	使用 Bonferroni 方法來調整顯著值。
effect_size_threshold_count	數字	設定在使用連續目標分割節點和合併種類時的作用大小臨界值。值必須介於 0.01 到 0.99 之間。
effect_size_threshold_cat	數字	設定使用種類目標分割節點和合併種類時的作用大小臨界值。值必須介於 0.01 到 0.99 之間。
split_merged_categories	旗標	容許重新分割合併種類。
grouping_sig_level	數字	用於確定如何形成節點群組或如何識別異常節點。
chi_square	pearson likelihood_ratio	用來計算方差統計值的方法：皮爾森 (Pearson) 或概似比
minimum_record_use	use_percentage use_absolute	
min_parent_records_pc	數字	預設值為 2。最小值為 1，最大值為 100，增量為 1。母項分支值必須高於子項分支。
min_child_records_pc	數字	預設值為 1。最小值為 1，最大值為 100，增量為 1。
min_parent_records_abs	數字	預設值為 100。最小值為 1，最大值為 100，增量為 1。母項分支值必須高於子項分支。
min_child_records_abs	數字	預設值為 50。最小值為 1，最大值為 100，增量為 1。
epsilon	數字	預期儲存格變更的最小變更。
max_iterations	數字	聚合的疊代數上限。
use_costs	旗標	

表 152: *treeas* 內容 (繼續)

treeas 內容	值	內容說明
costs	結構化	結構化內容。格式是 3 個值的清單：實際值、預測值及成本（如果該預測錯誤）。例如： tree.setPropertyValue("costs", [["drugA", "drugB", 3.0], ["drugX", "drugY", 4.0]])
default_cost_increase	none linear square custom	註：僅對序數目標啟用。 在成本矩陣中設定預設值。
calculate_conf	旗標	
display_rule_id	旗標	在評分輸出中新增一個欄位，以指出對其指派每一筆記錄的終端節點 ID。

twostepnode 內容



TwoStep 節點使用二階叢集方法。第一步進行一次資料透通，以便將原始輸入資料壓縮為可管理的一組子叢集。第二步使用階層式叢集方法將子叢集漸進地合併為更大的叢集。TwoStep 具有一個優點，就是能夠為訓練資料自動估計最佳叢集數目。它可以高效處理混合的欄位類型和大型的資料集。

範例

```
node = stream.create("twostep", "My node")
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["Age", "K", "Na", "BP"])
node.setPropertyValue("partition", "Test")
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "TwoStep_Drug")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("exclude_outliers", True)
node.setPropertyValue("cluster_label", "String")
node.setPropertyValue("label_prefix", "TwoStep_")
node.setPropertyValue("cluster_num_auto", False)
node.setPropertyValue("max_num_clusters", 9)
node.setPropertyValue("min_num_clusters", 3)
node.setPropertyValue("num_clusters", 7)
```

表 153: *twostepnode* 內容

twostepnode 內容	值	內容說明
inputs	[field1 ... fieldN]	TwoStep 模型使用輸入欄位的清單，但不使用目標。無法辨識加權和次數欄位。請參閱第 191 頁的『 一般建模節點內容 』主題，以取得更多資訊。
standardize	旗標	

表 153: *twostepnode* 內容 (繼續)

twostepnode 內容	值	內容說明
exclude_outliers	旗標	
percentage	數字	
cluster_num_auto	旗標	
min_num_clusters	數字	
max_num_clusters	數字	
num_clusters	數字	
cluster_label	String Number	
label_prefix	<i>string</i>	
distance_measure	Euclidean Loglikelihood	
clustering_criterion	AIC BIC	

twostepAS 內容



「TwoStep 叢集」是設計用來顯示資料集中自然分組（或叢集）的探索工具（原本不會加以顯示）。此程序使用的演算法有多個不錯的特徵使其有別於傳統叢集技術，例如處理種類和連續變數、叢集號碼的自動選擇以及可擴展性。

表 154: *twostepAS* 內容

twostepAS 內容	值	內容說明
inputs	[<i>f</i> 1 ... <i>f</i> <i>N</i>]	TwoStepAS 模型使用一列輸入欄位，但不使用目標欄位。無法辨識加權和次數欄位。
use_predefined_roles	Boolean	預設值=True
use_custom_field_assignments	Boolean	預設值為 False
cluster_num_auto	Boolean	預設值=True
min_num_clusters	整數	Default=2
max_num_clusters	整數	Default=15
num_clusters	整數	Default=5
clustering_criterion	AIC BIC	

表 154: twostepAS 內容 (繼續)

twostepAS 內容	值	內容說明
automatic_clustering_method	use_clustering_criterion_setting Distance_jump Minimum Maximum	
feature_importance_method	use_clustering_criterion_setting effect_size	
use_random_seed	布林 (Boolean)	
random_seed	整數	
distance_measure	Euclidean Loglikelihood	
include_outlier_clusters	布林 (Boolean)	預設值=True
num_cases_in_feature_tree_leaf_is_less_than	整數	Default=10
top_perc_outliers	整數	Default=5
initial_dist_change_threshold	整數	Default=0
leaf_node_maximum_branches	整數	Default=8
non_leaf_node_maximum_branches	整數	Default=8
max_tree_depth	整數	Default=3
adjustment_weight_on_measurement_level	整數	Default=6
memory_allocation_mb	號碼	Default=512
delayed_split	Boolean	預設值=True
fields_to_standardize	[f1 ... fN]	
adaptive_feature_selection	布林 (Boolean)	預設值=True
featureMisPercent	整數	Default=70
coefRange	數字	Default=0.05
percCasesSingleCategory	整數	Default=95
numCases	整數	Default=24
include_model_specifications	布林 (Boolean)	預設值=True
include_record_summary	布林 (Boolean)	預設值=True
include_field_transformations	Boolean	預設值=True

表 154: twostepAS 內容 (繼續)

twostepAS 內容	值	內容說明
excluded_inputs	Boolean	預設值=True
evaluate_model_quality	Boolean	預設值=True
show_feature_importance_bar chart	Boolean	預設值=True
show_feature_importance_ word_cloud	Boolean	預設值=True
show_outlier_clusters interactive_table_and_chart	Boolean	預設值=True
show_outlier_clusters_pivot_table	Boolean	預設值=True
across_cluster_feature_importance	Boolean	預設值=True
across_cluster_profiles_pivot_table	Boolean	預設值=True
withinprofiles	Boolean	預設值=True
cluster_distances	Boolean	預設值=True
cluster_label	String Number	
label_prefix	String	

第 14 章 模型區塊節點內容

模型區塊節點具有與其他節點相同的通用內容。請參閱第 65 頁的『通用節點內容』主題，以取得更多資訊。

applyanomalydetectionnode 內容

異常偵測建模節點可用於產生異常偵測模型區塊。該模型區塊的 Scripting 名稱為 *applyanomalydetectionnode*。如需 Script 化建模節點本身的相關資訊，第 191 頁的『*anomalydetectionnode* 內容』

表 155: *applyanomalydetectionnode* 內容

applyanomalydetectionnode 內容	值	內容說明
anomaly_score_method	FlagAndScore FlagOnly ScoreOnly	確定建立哪些輸出進行評分。
num_fields	整數	要報告的欄位。
discard_records	旗標	指示是否從輸出中捨棄記錄。
discard_anomalous_records	旗標	指示捨棄異常記錄還是捨棄非異常記錄。預設值是 <i>off</i> ，表示將捨棄非異常記錄。否則，如果值為 <i>on</i> ，那麼將捨棄異常記錄。僅當啟用了 <i>discard_records</i> 內容時，才會啟用此內容。

applyapriorinode 內容

可使用 Apriori 建模節點產生 Apriori 模型區塊。該模型區塊的 Scripting 名稱為 *applyapriorinode*。如需 Script 化建模節點本身的相關資訊，請參閱第 193 頁的『*apriorinode* 內容』

表 156: *applyapriorinode* 內容

applyapriorinode 內容	值	內容說明
max_predictions	數字（整數）	
ignore_unmatached	旗標	
allow_repeats	旗標	
check_basket	NoPredictions Predictions NoCheck	

表 156: *applyapriorinode* 內容 (繼續)

applyapriorinode 內容	值	內容說明
criterion	Confidence Support RuleSupport Lift Deployability	

applyassociationrulesnode 內容

「相關規則」建模節點可以用於產生關聯規則模型區塊。該模型區塊的 Scripting 名稱為 *applyassociationrulesnode*。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 194 頁的『[associationrulesnode 內容](#)』。

表 157: *applyassociationrulesnode* 內容

applyassociationrulesnode 內容	資料類型	內容說明
max_predictions	整數	評分的每個輸入可套用的規則數上限。
criterion	Confidence Rulesupport Lift Conditionsupport Deployability	選取用來決定規則強度的測量。
allow_repeats	布林	判定評分中是否包括具有相同預測的規則。
check_input	NoPredictions Predictions NoCheck	

applyautoclassifiernode 內容

「自動分類器」建模節點可用於產生「自動分類器」模型塊。此模型塊的 Scripting 名稱為 *applyautoclassifiernode*。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 197 頁的『[autoclassifiernode 內容](#)』。

表 158: applyautoclassifiernode 內容

applyautoclassifiernode 內容	值	內容說明
flag_ensemble_method	Voting EvaluationWeightedVoting ConfidenceWeightedVoting RawPropensityWeightedVoting HighestConfidence AverageRawPropensity	指定用來判定整體評分的方法。僅當選取的目標是旗標欄位時，此設定才適用。
flag_evaluation_selection	Accuracy AUC_ROC	此選項僅適用於旗標目標，用來決定選擇哪個評估測量進行評估加權投票。
filter_individual_model_output	旗標	指定是否應抑制各個模型的評分結果。
is_ensemble_update	旗標	啟用連續自動化機器學習模式，可將新的元件模型新增至現有的自動模型集，而不是取代現有的自動模型，並使用新的可用資料重新評估現有元件模型的測量。
is_auto_ensemble_weights_reevaluation	旗標	啟用自動模型加權重新評估。
use_accumulated_factor	旗標	累計因數用來計算累計測量。
accumulated_factor	number (float)	最大值為 0.99，最小值為 0.85。
use_accumulated_reducing	旗標	在模型重新整理期間，根據累計限制來執行模型減少。
accumulated_reducing_limit	number (float)	最大值為 0.7，最小值為 0.1。
use_accumulated_weighted_evaluation	旗標	累計評估測量用於在選取評估加權投票方法作為全體方法時進行投票。
flag_voting_tie_selection	Random HighestConfidence RawPropensity	如果選取投票方法，則指定鏈結的解析方法。僅當選取的目標是旗標欄位時，此設定才適用。

表 158: *applyautoclassifiernode* 內容 (繼續)

applyautoclassifiernode 內容	值	內容說明
set_ensemble_method	Voting EvaluationWeightedVoting ConfidenceWeightedVoting HighestConfidence	指定用來判定整體評分的方法。僅當選定的目標為集合欄位時，才會套用此設定。
set_voting_tie_selection	Random HighestConfidence	如果選取投票方法，則指定鏈結的解析方法。唯有當選取的目標是標稱欄位時才會套用此設定。

applyautoclusternode 內容

「自動叢集」建模節點可用於產生「自動叢集」模型區塊。該模型區塊的 Scripting 名稱為 *applyautoclusternode*。此模型區塊沒有其他內容。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 199 頁的『*autoclusternode* 內容』。

applyautonumericnode 內容

「自動數值」建模節點可用於產生「自動數值」模型區塊。此模型區塊的 Scripting 名稱為 *applyautonumericnode*。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 201 頁的『*autonumericnode* 內容』。

表 159: *applyautonumericnode* 內容

applyautonumericnode 內容	值	內容說明
calculate_standard_error	旗標	

applybayesnetnode 內容

可使用貝葉斯網路建模節點產生貝式網路模型區塊。該模型區塊的 Scripting 名稱為 *applybayesnetnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 203 頁的『*bayesnetnode* 內容』。

表 160: *applybayesnetnode* 內容

applybayesnetnode 內容	值	內容說明
all_probabilities	旗標	
raw_propensity	旗標	
adjusted_propensity	旗標	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	

applyc50node 內容

可使用 C5.0 建模節點產生 C5.0 模型塊。該模型塊的 Scripting 名稱為 *applyc50node*。如需 Script 化建模節點本身的相關資訊，請參閱第 205 頁的『[c50node 內容](#)』。

applyc50node 內容	值	內容說明
sql_generate	udf Never NoMissingValues	用來在規則集執行期間設定 SQL 產生選項。預設值是 udf。
calculate_conf	旗標	已啟用 SQL 產生時可用；此內容包含所產生樹狀結構中的信賴度計算。
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	

applycarmanode 內容

可使用 CARMA 建模節點產生 CARMA 模型區塊。該模型區塊的 Scripting 名稱為 *applycarmanode*。此模型區塊沒有其他內容。如需 Script 化建模節點本身的相關資訊，請參閱第 206 頁的『[carmanode 內容](#)』。

applycartnode 內容

可使用 C&R 樹狀結構建模節點產生 C&R 樹狀結構模型區塊。該模型區塊的 Scripting 名稱為 *applycartnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 207 頁的『[cartnode 內容](#)』。

applycartnode 內容	值	內容說明
enable_sql_generation	Never MissingValues NoMissingValues	用來在規則集執行期間設定 SQL 產生選項。
calculate_conf	旗標	已啟用 SQL 產生時可用；此內容包含所產生樹狀結構中的信賴度計算。
display_rule_id	旗標	在評分輸出中新增一個欄位，以指出對其指派每一筆記錄的終端節點 ID。
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	

applychaidnode 內容

可使用 CHAID 建模節點產生 CHAID 模型區塊。該模型區塊的 Scripting 名稱為 *applychaidnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 210 頁的『[chaidnode 內容](#)』。

表 163: *applychaidnode* 內容

applychaidnode 內容	值	內容說明
enable_sql_generation	Never MissingValues	用來在規則集執行期間設定 SQL 產生選項。
calculate_conf	旗標	
display_rule_id	旗標	在評分輸出中新增一個欄位，以指出對其指派每一筆記錄的終端節點 ID。
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	

applycoxregnode 內容

可使用 Cox 建模節點產生 Cox 模型區塊。該模型區塊的 Scripting 名稱為 *applycoxregnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 212 頁的『*coxregnode* 內容』。

表 164: *applycoxregnode* 內容

applycoxregnode 內容	值	內容說明
future_time_as	Intervals Fields	
time_interval	數字	
num_future_times	整數	
time_field	欄位	
past_survival_time	欄位	
all_probabilities	旗標	
cumulative_hazard	旗標	

applydecisionlistnode 內容

決策清單建模節點可用於產生決策清單模型區塊。該模型區塊的 Scripting 名稱為 *applydecisionlistnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 215 頁的『*decisionlistnode* 內容』。

表 165: *applydecisionlistnode* 內容

applydecisionlistnode 內容	值	內容說明
enable_sql_generation	旗標	值為 true 時，IBM SPSS Modeler 會嘗試將決策清單模型回推到 SQL。
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	

applydiscriminantnode 內容

判別建模節點可用於產生判別模型區塊。該模型區塊的 Scripting 名稱為 *applydiscriminantnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 216 頁的『discriminantnode 內容』。

applydiscriminantnode 內容	值	內容說明
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	

applyextension 內容



「延伸模型」節點可以用於產生「延伸」模型區塊。該模型區塊的 Scripting 名稱為 *applyextension*。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 218 頁的『extensionmodelnode 內容』。

Python for Spark 範例

```
##### script example for Python for Spark
applyModel = stream.findByType("extension_apply", None)

score_script = """
import json
import spss.pyspark.runtime
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.linalg import DenseVector
from pyspark.mllib.tree import DecisionTreeModel
from pyspark.sql.types import StringType, StructField

cxt = spss.pyspark.runtime.getContext()

if cxt.isComputeDataModelOnly():
    _schema = cxt.getSparkInputSchema()
    _schema.fields.append(StructField("Prediction", StringType(), nullable=True))
    cxt.setSparkOutputSchema(_schema)
else:
    df = cxt.getSparkInputData()

    _modelPath = cxt.getModelContentToPath("TreeModel")
    metadata = json.loads(cxt.getModelContentToString("model.dm"))

    schema = df.dtypes[:]
    target = "Drug"
    predictors = ["Age", "BP", "Sex", "Cholesterol", "Na", "K"]

    lookup = {}
    for i in range(0, len(schema)):
        lookup[schema[i][0]] = i

    def row2LabeledPoint(dm, lookup, target, predictors, row):
        target_index = lookup[target]
        tval = dm[target_index].index(row[target_index])
        pvals = []
        for predictor in predictors:
            predictor_index = lookup[predictor]
            if isinstance(dm[predictor_index], list):
                pval = row[predictor_index] in dm[predictor_index] and
dm[predictor_index].index(row[predictor_index]) or -1
            else:
                pval = row[predictor_index]
            pvals.append(pval)
```

```

return LabeledPoint(tval, DenseVector(pvals))

# convert dataframe to an RDD containing LabeledPoint
lps = df.rdd.map(lambda row: row2LabeledPoint(metadata,lookup,target,predictors,row))
treeModel = DecisionTreeModel.load(cxt.getSparkContext(), _modelPath);
# score the model, produces an RDD containing just double values
predictions = treeModel.predict(lps.map(lambda lp: lp.features))

def addPrediction(x,dm,lookup,target):
    result = []
    for _idx in range(0, len(x[0])):
        result.append(x[0][_idx])
    result.append(dm[lookup[target]][int(x[1])])
    return result

_schema = cxt.getSparkInputSchema()
_schema.fields.append(StructField("Prediction", StringType(), nullable=True))
rdd2 = df.rdd.zip(predictions).map(lambda x:addPrediction(x, metadata, lookup, target))
outDF = cxt.getSparkSQLContext().createDataFrame(rdd2, _schema)

"""
cxt.setSparkOutputData(outDF)
applyModel.setPropertyValue("python_syntax", score_script)

```

R 範例

```

#### script example for R
applyModel.setPropertyValue("r_syntax", """
result<-predict(modelerModel,newdata=modelerData)
modelerData<-cbind(modelerData,result)
var1<-c(fieldName="NaPrediction",fieldLabel="",fieldStorage="real",fieldMeasure="",
fieldFormat="",fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)""")

```

表 167: *applyextension* 內容

applyextension 內容	值	內容說明
r_syntax	string	用於模型評分的 R Scripting 語法。
python_syntax	string	用於模型評分的 Python Scripting 語法。
use_batch_size	旗標	啟用批次處理。
batch_size	整數	指定要併入在每個批次的資料記錄數。
convert_flags	StringsAndDoubles LogicalValues	用於轉換旗標欄位的選項。
convert_missing	旗標	將遺漏值轉換為 R NA 值的選項。
convert_datetime	旗標	用於將日期或日期時間格式的變數轉換為 R 日期/時間格式的選項。
convert_datetime_class	POSIXct POSIXlt	用於指定將日期或日期時間格式的變數轉換為何種格式的選項。

applyfactornode 內容

「主成分分析/因子」建模節點可用於產生「主成分分析/因子」模型區塊。該模型區塊的 Scripting 名稱為 *applyfactornode*。此模型區塊沒有其他內容。如需 Script 化建模節點本身的相關資訊，請參閱 [第 220 頁的『factornode 內容』](#)。

applyfeatureselectionnode 內容

「功能選擇」建模節點可用於產生「功能選擇」模型區塊。該模型區塊的 Scripting 名稱為 *applyfeatureselectionnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 222 頁的『featureselectionnode 內容』。

applyfeatureselectionnode 內容	值	內容說明
selected_ranked_fields		指定在模型瀏覽器中檢查哪些已排等級欄位。
selected_screened_fields		指定在模型瀏覽器中檢查哪些已篩選欄位。

applygeneralizedlinearnode 內容

「廣義線性 (genlin)」建模節點可用於產生「廣義線性」模型區塊。該模型區塊的 Scripting 名稱為 *applygeneralizedlinearnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 224 頁的『genlinnode 內容』。

applygeneralizedlinearnode 內容	值	內容說明
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	

applyglmnode 內容

GLMM 建模節點可用於產生 GLMM 模式塊。該模型區塊的 Scripting 名稱為 *applyglmnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 229 頁的『glmnode 內容』。

applyglmnode 內容	值	內容說明
confidence	onProbability onIncrease	用來計算評分信賴度值的基準；最高預測機率，或最高與次高預測機率之間的差異。
score_category_probabilities	旗標	如果設為 True，則會產生種類目標的預測機率。會為每一個類別建立一個欄位。預設值是 False。
max_categories	整數	要預測機率的種類數目上限。僅當 score_category_probabilities 為 True 時才使用。
score_propensity	旗標	如果設定為 True，那麼為具有旗標目標的模型生成原始傾向分數 ("True" 結果的可能性)。如果分割區處於有效，則模型還會根據測試分割區產生已調整的傾向分數。預設值是 False。

表 170: *applylmmnode* 內容 (繼續)

applylmmnode 內容	值	內容說明
enable_sql_generation	udf native	用來在串流執行期間設定 SQL 產生選項。選項包括：推回資料庫並使用 SPSS® Modeler Server 評分配接器（如果已連線到安裝了評分配接器的資料庫）進行評分，或者在 SPSS Modeler 內部評分。 預設值是 udf。

applygle 內容

可以將 GLE 建模節點用於產生 GLE 模型區塊。此模型區塊的 Scripting 名稱為 *applygle*。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 233 頁的『[gle 內容](#)』。

表 171: *applygle* 內容

applygle 內容	值	內容說明
enable_sql_generation	udf native	用來在串流執行期間設定 SQL 產生選項。選擇推回至資料庫並使用 SPSS Modeler Server 評分配接器（如果連接至已安裝評分配接器的資料庫）進行評分，或在 SPSS Modeler 內評分。

applygmm 內容

Gaussian Mixture 節點可用來產生 Gaussian Mixture 模型塊。此模型塊的 Scripting 名稱為 *applygmm*。18.2.1.1 版以及更新版本中提供下表格中的內容。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 383 頁的『[gmm 內容](#)』。

表 172: *applygmm* 內容

applygmm 內容	資料類型	內容說明
centers		
item_count		
total		
dimension		
components		
partition		

applykmeansnode 內容

K-Means 建模節點可用於產生 K-Means 模型區塊。該模型區塊的 Scripting 名稱為 *applykmeansnode*。此模型區塊沒有其他內容。如需 Script 化建模節點本身的相關資訊，請參閱第 240 頁的『[kmeansnode 內容](#)』。

applyknnnode 內容

KNN 建模節點可用於產生 KNN 模型區塊。此模型區塊的 Scripting 名稱是 *applyknnnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 242 頁的『[knnnode 內容](#)』。

表 173: <i>applyknnnode</i> 內容		
applyknnnode 內容	值	內容說明
all_probabilities	旗標	
save_distances	旗標	

applykohonennode 內容

Kohonen 建模節點可用於產生 Kohonen 模型區塊。該模型區塊的 Scripting 名稱為 *applykohonennode*。此模型區塊沒有其他內容。如需 Script 化建模節點本身的相關資訊，請參閱第 205 頁的『*c50node* 內容』。

applylinearnode 內容

可使用線性建模節點產生線性模型區塊。該模型區塊的 Scripting 名稱為 *applylinearnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 245 頁的『*linearnode* 內容』。

表 174: <i>applylinearnode</i> 內容		
linear 內容	值	內容說明
use_custom_name	旗標	
custom_name	string	
enable_sql_generation	udf native puresql	用來在串流執行期間設定 SQL 產生選項。選項包括：推回資料庫並使用 SPSS® Modeler Server 評分配接器（如果已連線到安裝了評分配接器的資料庫）進行評分，或者在 SPSS Modeler 內部評分。 預設值是 udf。

applylinearnode 內容

線性 AS 建模節點可用於產生線性 AS 模型區塊。此模型區塊的 Scripting 名稱為 *applylinearnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 247 頁的『*linearnode* 內容』。

表 175: <i>applylinearnode</i> 內容		
applylinearnode 內容	值	內容說明
enable_sql_generation	udf native	預設值是 udf。

applylogregnode 內容

「邏輯迴歸模型建模」節點可用於產生「邏輯迴歸模型」模型區塊。該模型區塊的 Scripting 名稱為 *applylogregnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 248 頁的『*logregnode* 內容』。

表 176: <i>applylogregnode</i> 內容		
applylogregnode 內容	值	內容說明
calculate_raw_propensiti es	旗標	

表 176: *applylogregnode* 內容 (繼續)

applylogregnode 內容	值	內容說明
calculate_conf	旗標	
enable_sql_generation	旗標	

applysvmnode 內容

可以使用 LSVM 建模節點來產生 LSVM 模型區塊。此模型區塊的 Scripting 名稱為 *applysvmnode*。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 253 頁的『[lsvmnode 內容](#)』。

表 177: *applysvmnode* 內容

applysvmnode 內容	值	內容說明
calculate_raw_propensities	旗標	指定是否計算原始傾向分數。
enable_sql_generation	udf native	指定使用評分配接器（如果已安裝）進行評分、在進行中評分還是在資料庫外部進行評分。

applyneuralnetnode 內容

「神經網路」建模節點可用於產生「神經網路」模型區塊。該模型區塊的 Scripting 名稱為 *applyneuralnetnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 254 頁的『[neuralnetnode 內容](#)』。

警告：在此發行版中提供了具有增強功能的新版本的神經網路模型區塊，並將在下一節 (*applyneuralnetwork*) 中進行介紹。儘管先前版本仍然可用，但我們建議您更新 Script 以使用新的版本。此處保留了先前版本的詳細資料以供參照，但會在將來的發行版中不再支援。

表 178: *applyneuralnetnode* 內容

applyneuralnetnode 內容	值	內容說明
calculate_conf	旗標	已啟用 SQL 產生時可用；此內容包含所產生樹狀結構中的信賴度計算。
enable_sql_generation	旗標	
nn_score_method	Difference SoftMax	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	

applyneuralnetworknode 內容

可使用神經網路建模節點產生神經網路模型區塊。該模型區塊的 Scripting 名稱為 *applyneuralnetworknode*。如需對建模節點自身進行 Script 的相關資訊，請參閱 [neuralnetworknode 內容](#)。

表 179: *applyneuralnetworknode* 內容

applyneuralnetworknode 內容	值	內容說明
use_custom_name	旗標	
custom_name	string	
confidence	onProbability onIncrease	
score_category_probabilities	旗標	
max_categories	數字	
score_propensity	旗標	
enable_sql_generation	udf native puresql	用來在串流執行期間設定 SQL 產生選項。選項包括：推回資料庫並使用 SPSS® Modeler Server 評分配接器（如果已連線到安裝了評分配接器的資料庫）進行評分，或者在 SPSS Modeler 內部評分。 預設值是 udf。

applyocsvmnode 內容

一級 SVM 節點可用於產生一類 SVM 模型區塊。此模型區塊的 Scripting 名稱為 *applyocsvmnode*。此模型區塊沒有其他內容。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 387 頁的『*ocsvmnode* 內容』。

applyquestnode 內容

可使用 QUEST 建模節點產生 QUEST 模型區塊。該模型區塊的 Scripting 名稱為 *applyquestnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 259 頁的『*questnode* 內容』。

表 180: *applyquestnode* 內容

applyquestnode 內容	值	內容說明
enable_sql_generation	Never MissingValues NoMissingValues	用來在規則集執行期間設定 SQL 產生選項。
calculate_conf	旗標	
display_rule_id	旗標	在評分輸出中新增一個欄位，以指出對其指派每一筆記錄的終端節點 ID。
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	

applyr 內容

「R 建置」節點可用於產生 R 模型區塊。此模型區塊的 Scripting 名稱為 *applyr*。如需 Script 化建模節點本身的相關資訊，請參閱第 204 頁的『[buildr 內容](#)』。

applyr 內容	值	內容說明
score_syntax	string	用於模型評分的 R Scripting 語法。
convert_flags	StringsAndDoubles LogicalValues	用於轉換旗標欄位的選項。
convert_datetime	旗標	用於將日期或日期時間格式的變數轉換為 R 日期/時間格式的選項。
convert_datetime_class	POSIXct POSIXlt	用於指定將日期或日期時間格式的變數轉換為何種格式的選項。
convert_missing	旗標	用於將遺漏值轉換為 R NA 值的選項。
use_batch_size	旗標	啟用批次處理
batch_size	整數	請指定要在每一個批次中包含的資料記錄數

applyrandomtrees 內容

可以使用「隨機樹狀結構」建模節點來產生隨機樹狀結構模型區塊。此模型區塊的 Scripting 名稱為 *applyrandomtrees*。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 261 頁的『[randomtrees 內容](#)』。

applyrandomtrees 內容	值	內容說明
calculate_conf	旗標	此內容包括產生的樹狀結構中的信賴度計算。
enable_sql_generation	udf native	用來在串流執行期間設定 SQL 產生選項。選擇推回至資料庫並使用 SPSS Modeler Server 評分配接器（如果連接至已安裝評分配接器的資料庫）進行評分，或在 SPSS Modeler 內評分。

asapplyregressionnode 內容

「線性迴歸」建模節點可用於產生「線性迴歸」模型區塊。該模型區塊的 Scripting 名稱為 *applyregressionnode*。此模型區塊沒有其他內容。如需 Script 化建模節點本身的相關資訊，請參閱第 263 頁的『[regressionnode 內容](#)』。

applyselflearningnode 內容

「自我學習回應模型 (SLRM)」建模節點可用於產生 SLRM 模型區塊。該模型區塊的 Scripting 名稱為 *applyselflearningnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 267 頁的『[slrmnode 內容](#)』。

表 183: *applyselflearningnode* 內容

applyselflearningnode 內容	值	內容說明
max_predictions	數字	
randomization	數字	
scoring_random_seed	數字	
sort	ascending descending	指定先顯示評分最高的供應項目還是評分最低的供應項目。
model_reliability	旗標	將「設定」標籤中的模型可靠性選項考慮在內。

applysequencenode 內容

「序列」建模節點可用於產生「序列」模型區塊。該模型區塊的 Scripting 名稱為 *applysequencenode*。此模型區塊沒有其他內容。如需 Script 化建模節點本身的相關資訊，請參閱第 266 頁的『[sequencenode 內容](#)』。

applysvmnode 內容

可使用 SVM 建模節點產生 SVM 模型區塊。該模型區塊的 Scripting 名稱為 *applysvmnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 274 頁的『[svmnode 內容](#)』。

表 184: *applysvmnode* 內容

applysvmnode 內容	值	內容說明
all_probabilities	旗標	
calculate_raw_propensities	旗標	
calculate_adjusted_propensities	旗標	

applystpnode 內容

STP 建模節點可以用於產生相關聯的模型區塊，該模型區塊在輸出檢視器中顯示模型輸出。該模型區塊的 Scripting 名稱為 *applystpnode*。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 268 頁的『[stpnode 內容](#)』。

表 185: *applystpnode* 內容

applystpnode 內容	資料類型	內容說明
uncertainty_factor	布林	下限 0, 上限 100。

applytcmnode 內容

時間原因建模 (TCM) 建模節點可用於產生 TCM 模型區塊。此模型區塊的 Scripting 名稱為 *applytcmnode*。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 275 頁的『[tcmnode 內容](#)』。

表 186: *applytcmnode* 內容

applytcmnode 內容	值	內容說明
ext_future	boolean	

表 186: *applytcmnode* 內容 (繼續)

applytcmnode 內容	值	內容說明
ext_future_num	整數	
noise_res	<i>boolean</i>	
conf_limits	<i>boolean</i>	
target_fields	清單	
target_series	清單	

applyts 內容

「時間序列」建模節點可以用來產生「時間序列」模型區塊。此模型區塊的 Scripting 名稱為 *applyts*。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 280 頁的『[ts 內容](#)』。

表 187: *applyts* 內容

applyts 內容	值	內容說明
extend_records_into_future	布林	
ext_future_num	整數	
compute_future_values_input	布林	
forecastperiods	整數	
noise_res	<i>boolean</i>	
conf_limits	<i>boolean</i>	
target_fields	清單	
target_series	清單	
includeTargets	欄位	

applytimeseriesnode 內容 (已淘汰)

「時間序列」建模節點可以用來產生「時間序列」模型區塊。該模型區塊的 Scripting 名稱為 *applytimeseriesnode*。如需 Script 化建模節點本身的相關資訊，請參閱第 288 頁的『[timeseriesnode 內容 \(已淘汰\)](#)』。

表 188: *applytimeseriesnode* 內容

applytimeseriesnode 內容	值	內容說明
calculate_conf	旗標	
calculate_residuals	旗標	

applytreeas 內容

Tree-AS 建模節點可用於產生 Tree-AS 模型區塊。此模型區塊的 Scripting 名稱為 *applytreeas*。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 290 頁的『[treeas 內容](#)』。

表 189: *applytreeas* 內容

applytreeas 內容	值	內容說明
calculate_conf	旗標	此內容包括產生的樹狀結構中的信賴度計算。
display_rule_id	旗標	在評分輸出中新增一個欄位，以指出對其指派每一筆記錄的終端節點 ID。
enable_sql_generation	udf native	用來在串流執行期間設定 SQL 產生選項。選擇推回至資料庫並使用 SPSS Modeler Server 評分配接器（如果連接至已安裝評分配接器的資料庫）進行評分，或在 SPSS Modeler 內評分。

applytwostepnode 內容

「二階」建模節點可用於產生「二階」模型區塊。該模型區塊的 Scripting 名稱為 *applytwostepnode*。此模型區塊沒有其他內容。如需 Script 化建模節點本身的相關資訊，請參閱第 292 頁的『[twostepnode 內容](#)』。

applytwostepAS 內容

二階 AS 建模節點可用於產生二階 AS 模型區塊。此模型區塊的 Scripting 名稱為 *applytwostepAS*。如需 Script 化建模節點本身的相關資訊，請參閱第 293 頁的『[twostepAS 內容](#)』。

表 190: *applytwostepAS* 內容

applytwostepAS 內容	值	內容說明
enable_sql_generation	udf native	用來在串流執行期間設定 SQL 產生選項。選項包括：推回資料庫並使用 SPSS® Modeler Server 評分配接器（如果已連線到安裝了評分配接器的資料庫）進行評分，或者在 SPSS Modeler 內部評分。 預設值是 udf。

applyxgboosttreenode 內容

XGBoost Tree 節點可用於產生 XGBoost Tree 模型區塊。此模型區塊的 Scripting 名稱為 *applyxgboosttreenode*。18.2.1.1 中已新增下表中的內容。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 394 頁的『[xgboosttreenode 內容](#)』。

表 191: *applyxgboosttreenode* 內容

applyxgboosttreenode 內容	資料類型	內容說明
use_model_name		
model_name		

applyxgboostlinearnode 內容

XGBoost Linear 節點可用於產生 XGBoost Linear 模型區塊。此模型區塊的 Scripting 名稱為 *applyxgboostlinearnode*。此模型區塊沒有其他內容。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 393 頁的『[xgboostlinearnode 內容](#)』。

hdbscannugget 內容

HDBSCAN 節點可用於產生 HDBSCAN 模型塊。此模型塊的 Scripting 名稱為 hdbscannugget。此模型區塊沒有其他內容。有關編寫建模節點自身 Script 的詳細資訊，請參閱第 383 頁的『hdbscannode 內容』。

kdeapply 內容

KDE 建模節點可用來產生 KDE 模型塊。這個模型塊的 Scripting 名稱為 kdeapply。如需編寫建模節點自身 Script 的相關資訊，請參閱第 385 頁的『kdemodel 內容』。

kdeapply 內容	資料類型	內容說明
outLogDensity 從 18.2.1.1 版開始重新命名為 out_log_density	<i>boolean</i>	指定 True 或 False 以在輸出中併入或排除日誌密度值。預設值是 False。

第 15 章 資料庫建模節點內容

IBM SPSS Modeler 支援與資料庫供應商提供的資料採礦和建模工具整合，包括 Microsoft SQL Server Analysis Services、Oracle Data Mining、和 IBM Netezza Analytics。您可以使用 IBM SPSS Modeler 應用程式自有的資料庫演算法來建立模型並對模型進行評分。還可以使用本節介紹的內容通過編寫 Script 來構建和操作資料庫模型。

例如，下面這段 Script 說明如何使用 IBM SPSS Modeler Script 編寫介面來建立 Microsoft 決策樹狀結構模型：

```
stream = modeler.script.stream()
msbuilder = stream.createAt("mstreenode", "MSBuilder", 200, 200)

msbuilder.setPropertyValue("analysis_server_name", 'localhost')
msbuilder.setPropertyValue("analysis_database_name", 'TESTDB')
msbuilder.setPropertyValue("mode", 'Expert')
msbuilder.setPropertyValue("datasource", 'LocalServer')
msbuilder.setPropertyValue("target", 'Drug')
msbuilder.setPropertyValue("inputs", ['Age', 'Sex'])
msbuilder.setPropertyValue("unique_field", 'IDX')
msbuilder.setPropertyValue("custom_fields", True)
msbuilder.setPropertyValue("model_name", 'MSDRUG')

typenode = stream.findByType("type", None)
stream.link(typenode, msbuilder)
results = []
msbuilder.run(results)
msapplier = stream.createModelApplierAt(results[0], "Drug", 200, 300)
tablenode = stream.createAt("table", "Results", 300, 300)
stream.linkBetween(msapplier, typenode, tablenode)
msapplier.setPropertyValue("sql_generate", True)
tablenode.run([])
```

Microsoft 建模的節點內容

Microsoft 建模節點內容

通用內容

Microsoft 資料庫建模節點的一般內容如下所示。

一般 Microsoft 節點內容	值	內容說明
analysis_database_name	string	Analysis Services 資料庫的名稱。
analysis_server_name	string	Analysis Services 主機的名稱。
use_transactional_data	旗標	指定輸入資料是採用表格式還是交易格式。
inputs	清單	表格資料的輸入欄位。
target	欄位	預測欄位（不適用於「MS 叢集」或「序列叢集」節點）。
unique_field	欄位	索引鍵欄位。

表 193: 一般 Microsoft 節點內容 (繼續)		
一般 Microsoft 節點內容	值	內容說明
msas_parameters	結構化	演算法參數。請參閱第 317 頁的『演算法參數』主題，以取得更多資訊。
with_drillthrough	旗標	使用「往下探查」選項。

MS 決策樹狀結構

沒有為 `mstreenode` 類型的節點定義具體內容。請參閱此區段開頭處的一般 Microsoft 內容。

MS 叢集

沒有為 `msclusternode` 類型的節點定義具體內容。請參閱此區段開頭處的一般 Microsoft 內容。

MS 關聯規則

下列特定內容可用於類型為 `msassocnode` 的節點：

表 194: <code>msassocnode</code> 內容		
<code>msassocnode</code> 內容	值	內容說明
<code>id_field</code>	欄位	識別資料中的每筆交易。
<code>trans_inputs</code>	清單	交易式資料的輸入欄位。
<code>transactional_target</code>	欄位	預測欄位（交易資料）。

MS 貝式邏輯分類演算法 (Naive Bayes)

沒有為 `msbayesnode` 類型的節點定義具體內容。請參閱此區段開頭處的一般 Microsoft 內容。

MS 線性迴歸

沒有為 `msregressionnode` 類型的節點定義具體內容。請參閱此區段開頭處的一般 Microsoft 內容。

MS 神經網路

沒有為 `msneuralnetworknode` 類型的節點定義具體內容。請參閱此區段開頭處的一般 Microsoft 內容。

MS 邏輯迴歸

沒有為 `mslogisticnode` 類型的節點定義具體內容。請參閱此區段開頭處的一般 Microsoft 內容。

MS 時間序列

沒有為 `mstimeseriesnode` 類型的節點定義具體內容。請參閱此區段開頭處的一般 Microsoft 內容。

MS 順序叢集

下列特定內容可用於類型為 `mssequenceclusternode` 的節點：

表 195: <code>mssequenceclusternode</code> 內容		
<code>mssequenceclusternode</code> 內容	值	內容說明
<code>id_field</code>	欄位	識別資料中的每筆交易。

表 195: mssequenceclusternode 內容 (繼續)

mssequenceclusternode 內容	值	內容說明
input_fields	清單	交易式資料的輸入欄位。
sequence_field	欄位	序列 ID。
target_field	欄位	預測欄位 (表格資料)。

演算法參數

每一個 Microsoft 資料庫機型都具有特定參數，可以使用 msas_parameters 內容來設定這些參數，例如：

```
stream = modeler.script.stream()
msregressionnode = stream.findByType("msregression", None)
msregressionnode.setPropertyValue("msas_parameters",
[["MAXIMUM_INPUT_ATTRIBUTES", 255],
["MAXIMUM_OUTPUT_ATTRIBUTES", 255]])
```

這些參數源自 SQL Server。如需每個節點的相關參數，請執行如下操作：

1. 將資料庫來源節點放入畫布中。
2. 開啟該資料庫來源節點。
3. 從**資料來源**下拉清單中選取一個有效來源。
4. 從**表格名稱**清單中選取一個有效表格。
5. 按一下**確定**以關閉該資料庫來源節點。
6. 附加要列出其內容的 Microsoft 資料庫建模節點。
7. 開啟該資料庫建模節點。
8. 選取**專家標籤**。

此時會顯示該節點的可用 msas_parameters 內容。

Microsoft 模型塊內容

使用 Microsoft 資料庫建模節點建立的模型塊具有下列內容。

MS 決策樹狀結構

表 196: MS 決策樹狀結構內容

applymstreenode 內容	值	說明
analysis_database_name	string	此節點可以在串流中直接進行評分。 此內容用來識別 Analysis Services 資料庫的名稱。
analysis_server_name	string	Analytic Server 主機的名稱。
datasource	string	SQL Server ODBC 資料來源名稱 (DSN) 的名稱。
sql_generate	旗標 udf	啟用 SQL 產生。

MS 線性迴歸

表 197: MS 線性迴歸內容		
appliesregressionnode 內容	值	說明
analysis_database_name	string	此節點可以在串流中直接進行評分。 此內容用來識別 Analysis Services 資料庫的名稱。
analysis_server_name	string	Analytic Server 主機的名稱。

MS 神經網路

表 198: MS 神經網路內容		
appliesneuralnetworknode 內容	值	說明
analysis_database_name	string	此節點可以在串流中直接進行評分。 此內容用來識別 Analysis Services 資料庫的名稱。
analysis_server_name	string	Analytic Server 主機的名稱。

MS 邏輯迴歸

表 199: MS 邏輯迴歸內容		
applieslogisticnode 內容	值	說明
analysis_database_name	string	此節點可以在串流中直接進行評分。 此內容用來識別 Analysis Services 資料庫的名稱。
analysis_server_name	string	Analytic Server 主機的名稱。

MS 時間序列

表 200: MS 時間序列內容		
aplymstimeseriesnode 內容	值	說明
analysis_database_name	string	此節點可以在串流中直接進行評分。 此內容用來識別 Analysis Services 資料庫的名稱。
analysis_server_name	string	Analytic Server 主機的名稱。
start_from	new_prediction historical_prediction	指定是進行未來預測還是歷程預測。
new_step	數字	定義未來預測的啟動時間段。

表 200: MS 時間序列內容 (繼續)

appliedtimeseriesnode 內容	值	說明
historical_step	數字	定義歷程預測的啟動時間段。
end_step	數字	定義預測結束時間段。

MS 順序叢集

表 201: MS 時序叢集內容

appliedsequenceclusternode 內容	值	說明
analysis_database_name	string	此節點可以在串流中直接進行評分。 此內容用來識別 Analysis Services 資料庫的名稱。
analysis_server_name	string	Analytic Server 主機的名稱。

Oracle 建模的節點內容

Oracle 建模節點內容

Oracle 資料庫建模節點的一般內容如下所示。

表 202: 一般 Oracle 節點內容

一般 Oracle 節點內容	值	內容說明
target	欄位	
inputs	欄位的清單	
partition	欄位	此欄位用於將資料分割為不同的樣本，以用於模型建置的訓練、檢定和驗證階段。
datasource		
username		
password		
epassword		
use_model_name	旗標	
model_name	string	自訂新模型的名稱。
use_partitioned_data	旗標	如果已定義分割區欄位，則此選項可確保只使用訓練分割區中的資料來建置模型。
unique_field	欄位	
auto_data_prep	旗標	啟用或取消 Oracle 自動資料準備功能（僅適用於 11g 資料庫）。
costs	結構化	結構化內容，格式如下： [[drugA drugB 1.5] [drugA drugC 2.1]]，其中 [] 中的引數是實際預測成本。

表 202: 一般 Oracle 節點內容 (繼續)

一般 Oracle 節點內容	值	內容說明
mode	Simple Expert	如在各個節點內容中所記錄, 如果設定為 Simple, 會導致忽略某些內容。
use_prediction_probability	旗標	
prediction_probability	string	
use_prediction_set	旗標	

Oracle Naive Bayes

類型為 oranbnode 的節點的可用內容如下所示:

表 203: oranbnode 內容

oranbnode 內容	值	內容說明
singleton_threshold	數字	0.0-1.0.*
pairwise_threshold	數字	0.0-1.0.*
priors	Data Equal Custom	
custom_priors	結構化	結構化內容, 格式如下: set :oranbnode.custom_priors = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]

* 如果 mode 設為 Simple, 則忽略內容。

Oracle 調適性 Bayes

類型為 oraabnnode 的節點的可用內容如下所示:

表 204: oraabnnode 內容

oraabnnode 內容	值	內容說明
model_type	SingleFeature MultiFeature NaiveBayes	
use_execution_time_limit	旗標	*
execution_time_limit	整數	值必須大於 0。 *
max_naive_bayes_predictors	整數	值必須大於 0。 *
max_predictors	整數	值必須大於 0。 *

表 204: oraabnnode 內容 (繼續)

oraabnnode 內容	值	內容說明
priors	Data Equal Custom	
custom_priors	結構化	結構化內容, 格式如下: set :oraabnnode.custom_priors = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]

* 如果 mode 設為 Simple, 則忽略內容。

Oracle 支援向量機器

類型為 orasvmnode 的節點的可用內容如下所示:

表 205: orasvmnode 內容

orasvmnode 內容	值	內容說明
active_learning	Enable Disable	
kernel_function	Linear Gaussian System	
normalization_method	zscore minmax none	
kernel_cache_size	整數	僅適用於高斯核心。值必須大於 0。*
convergence_tolerance	數字	值必須大於 0。*
use_standard_deviation	旗標	僅適用與高斯核心。*
standard_deviation	數字	值必須大於 0。*
use_epsilon	旗標	僅適用於迴歸模型。*
epsilon	數字	值必須大於 0。*
use_complexity_factor	旗標	*
complexity_factor	數字	*
use_outlier_rate	旗標	僅適用於一級變式。*

表 205: orasvmnode 內容 (繼續)

orasvmnode 內容	值	內容說明
outlier_rate	數字	僅適用於一級變式。0.0–1.0.*
weights	Data Equal Custom	
custom_weights	結構化	結構化內容，格式如下： set :orasvmnode.custom_weights = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]

* 如果 mode 設為 Simple，則忽略內容。

Oracle 廣義線性模型

類型為 oraglmnode 的節點的可用內容如下所示：

表 206: oraglmnode 內容

oraglmnode 內容	值	內容說明
normalization_method	zscore minmax none	
missing_value_handling	ReplaceWithMean UseCompleteRecords	
use_row_weights	旗標	*
row_weights_field	欄位	*
save_row_diagnostics	旗標	*
row_diagnostics_table	string	*
coefficient_confidence	數字	*
use_reference_category	旗標	*
reference_category	string	*
ridge_regression	Auto Off On	*
parameter_value	數字	*

表 206: oraglmnode 內容 (繼續)		
oraglmnode 內容	值	內容說明
vif_for_ridge	旗標	*

* 如果 mode 設為 Simple, 則忽略內容。

Oracle 決策樹狀結構

類型為 oradecisiontreenode 的節點的可用內容如下所示:

表 207: oradecisiontreenode 內容		
oradecisiontreenode 內容	值	內容說明
use_costs	旗標	
impurity_metric	Entropy Gini	
term_max_depth	整數	2–20.*
term_minpct_node	數字	0.0–10.0.*
term_minpct_split	數字	0.0–20.0.*
term_minrec_node	整數	值必須大於 0。 *
term_minrec_split	整數	值必須大於 0。 *
display_rule_ids	旗標	*

* 如果 mode 設為 Simple, 則忽略內容。

Oracle O-叢集

類型為 oraoclusternode 的節點的可用內容如下所示:

表 208: oraoclusternode 內容		
oraoclusternode 內容	值	內容說明
max_num_clusters	整數	值必須大於 0。
max_buffer	整數	值必須大於 0。 *
sensitivity	數字	0.0–1.0.*

* 如果 mode 設為 Simple, 則忽略內容。

Oracle KMeans

類型為 orakmeansnode 的節點的可用內容如下所示:

表 209: orakmeansnode 內容		
orakmeansnode 內容	值	內容說明
num_clusters	整數	值必須大於 0。

表 209: orakmeansnode 內容 (繼續)

orakmeansnode 內容	值	內容說明
normalization_method	zscore minmax none	
distance_function	Euclidean Cosine	
iterations	整數	0–20.*
conv_tolerance	數字	0.0–0.5.*
split_criterion	Variance Size	預設值為 Variance。*
num_bins	整數	值必須大於 0。*
block_growth	整數	1–5.*
min_pct_attr_support	數字	0.0–1.0.*

* 如果 mode 設為 Simple，則忽略內容。

Oracle NMF

類型為 oranmfnode 的節點的可用內容如下所示：

表 210: oranmfnode 內容

oranmfnode 內容	值	內容說明
normalization_method	minmax none	
use_num_features	旗標	*
num_features	整數	0–1。預設值由演算法根據資料估計得出。*
random_seed	數字	*
num_iterations	整數	0–500.*
conv_tolerance	數字	0.0–0.5.*
display_all_features	旗標	*

* 如果 mode 設為 Simple，則忽略內容。

Oracle Apriori

類型為 oraapriorinode 的節點的可用內容如下所示：

表 211: oraapriorinode 內容

oraapriorinode 內容	值	內容說明
content_field	欄位	
id_field	欄位	
max_rule_length	整數	2–20。
min_confidence	數字	0.0–1.0。
min_support	數字	0.0–1.0。
use_transactional_data	旗標	

Oracle 最小說明長度 (MDL)

沒有為類型為 oramdlnode 的節點定義具體內容。請參閱本章節開頭部分的通用 Oracle 內容。

Oracle 屬性重要性 (AI)

類型為 oraainode 的節點的可用內容如下所示：

表 212: oraainode 內容

oraainode 內容	值	內容說明
custom_fields	旗標	如果為 true，則容許您為現行節點指定目標、輸入及其他欄位。如果為 false，則會使用上游「類型」節點中的現行設定。
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	旗標	當 selection_mode 設為 ImportanceLevel 時，指定是否選取重要欄位。
important_label	string	指定「重要」分等級的標籤。
select_marginal	旗標	當 selection_mode 設為 ImportanceLevel 時，指定是否選取一般欄位。
marginal_label	string	指定「邊際」分等級的標籤。
important_above	數字	0.0–1.0。
select_unimportant	旗標	當 selection_mode 設為 ImportanceLevel 時，指定是否選取不重要欄位。
unimportant_label	string	指定「不重要」分等級的標籤。
unimportant_below	數字	0.0–1.0。
importance_value	數字	當 selection_mode 設為 ImportanceValue 時，指定要使用的截斷值。接受 0 與 100 之間的值。
top_n	數字	當 selection_mode 設為 TopN 時，指定要使用的截斷值。接受 0 與 1000 之間的值。

Oracle 模型區塊內容

使用 Oracle 模型建立的模型區塊具有下列內容。

Oracle Naive Bayes

沒有為 `applyoranbnnode` 類型的節點定義具體內容。

Oracle 調適性 Bayes

沒有為 `applyoraabnnode` 類型的節點定義具體內容。

Oracle 支援向量機器

沒有為 `applyorasvmnode` 類型的節點定義具體內容。

Oracle 決策樹狀結構

類型為 `applyoradecisiontreenode` 的節點的可用內容如下所示：

表 213: <code>applyoradecisiontreenode</code> 內容		
<code>applyoradecisiontreenode</code> 內容	值	內容說明
<code>use_costs</code>	旗標	
<code>display_rule_ids</code>	旗標	

Oracle O-叢集

沒有為 `applyoraoclusternode` 類型的節點定義具體內容。

Oracle KMeans

沒有為 `applyorakmeansnode` 類型的節點定義具體內容。

Oracle NMF

下列內容用於 `applyoranmfnode` 類型的節點：

表 214: <code>applyoranmfnode</code> 內容		
<code>applyoranmfnode</code> 內容	值	內容說明
<code>display_all_features</code>	旗標	

Oracle Apriori

在 Scripting 中無法套用此模型區塊。

Oracle MDL

在 Scripting 中無法套用此模型區塊。

IBM Netezza Analytics 建模節點內容

Netezza 建模節點內容

IBM Netezza 資料庫建模節點的一般內容如下所示。

表 215: 一般 Netezza 節點內容

一般 Netezza 節點內容	值	內容說明
custom_fields	旗標	如果為 true, 則容許您為現行節點指定目標、輸入及其他欄位。如果為 false, 則會使用上游「類型」節點中的現行設定。
inputs	[field1 ... fieldn]	模型所用的輸入或預測值欄位。
target	欄位	目標欄位 (連續或種類)。
record_id	欄位	要用作唯一記錄 ID 的欄位。
use_upstream_connection	旗標	如果為 true (預設), 那麼連線細節在上游節點中指定。在指定了 move_data_to_connection 時不使用。
move_data_connection	旗標	如果為 true, 則將資料移動到由 connection 指定的資料庫。在指定了 use_upstream_connection 時不使用。
connection	結構化	<p>儲存模型的 Netezza 資料庫的連線字串。結構化內容, 格式如下:</p> <pre>['odbc' '<dsn>' '<username>' '<psw>' '<catname>' '<conn_attribs>' [true false]]</pre> <p>其中:</p> <p><dsn> 是資料來源名稱</p> <p><username> 和 <psw> 是資料庫的使用者名稱和密碼</p> <p><catname> 是型錄名稱</p> <p><conn_attribs> 是連線屬性</p> <p>true false 指示是否需要密碼。</p>
table_name	string	這是用於儲存模型的資料庫表格的名稱。
use_model_name	旗標	如果為 true, 使用由 model_name 指定的名稱作為模型名稱, 否則採用系統建立的模型名稱。
model_name	string	自訂新模型的名稱。
include_input_fields	旗標	如果為 true, 向下游傳送所有輸入欄位, 否則僅傳送模型產生的 record_id 和欄位。

Netezza 決策樹狀結構

類型為 netezzadectreenode 的節點的可用內容如下所示:

表 216: netezzadectreenode 內容

netezzadectreenode 內容	值	內容說明
impurity_measure	Entropy Gini	對雜質的測量，用於評估樹狀結構的最佳分割位置。
max_tree_depth	整數	樹狀結構可以增長到的上限層次數。預設值為 62（上限可能值）。
min_improvement_splits	數字	進行分割前必須滿足的最低雜質改善。預設值為 0.01。
min_instances_split	整數	可以進行分割前餘下的下限未分割記錄數。預設值為 2（下限可能值）。
weights	結構化	各個類別的相對加權。結構化內容，格式如下： set :netezza_dectree.weights = [[drugA 0.3][drugB 0.6]] 預設情況是所有類別的加權均為 1。
pruning_measure	Acc wAcc	預設值為 Acc（精確度）。如果要在套用刪改時將類別加權考慮在內，可套用 wAcc（加權精確度）替代。
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	預設為使用 allTrainingData 來估計模型準確性。使用 partitionTrainingData 來指定要使用的訓練資料百分比，或 useOtherTable 來使用指定資料庫表格中的訓練資料集。
perc_training_data	數字	如果 prune_tree_options 設定為 partitionTrainingData，則指定用於訓練的資料所佔的百分比。
prune_seed	整數	在 prune_tree_options 設定為 partitionTrainingData 時，用於抄寫分析結果的隨機種子，預設值是 1。
pruning_table	string	用來估計模型準確性的個別刪改資料集的表格名稱。
compute_probabilities	旗標	如果為 true，會產生信賴等級（機率）欄位以及預測欄位。

Netezza K-Means

類型為 netezzakmeansnode 的節點的可用內容如下所示：

表 217: netezzakmeansnode 內容

netezzakmeansnode 內容	值	內容說明
distance_measure	Euclidean Manhattan Canberra maximum	這是用於對資料點之間的距離進行測量的方法。
num_clusters	整數	要建立的叢集數目；預設值為 3。
max_iterations	整數	演算法疊代次數，模型訓練在此之後停止；預設值為 5。
rand_seed	整數	這是用於抄寫分析結果的隨機種子；預設值為 12345。

Netezza Bayes Net

類型為 netezزابayesnode 的節點的可用內容如下所示：

表 218: netezزابayesnode 內容

netezزابayesnode 內容	值	內容說明
base_index	整數	對第一個輸入欄位指定的數值 ID，用於進行內部管理；預設值為 777。
sample_size	整數	屬性號碼非常大時的採樣大小；預設值為 10,000。
display_additional_information	旗標	如果為 true，則在訊息對話框中顯示額外的進度資訊。
type_of_prediction	best neighbors nn-neighbors	要使用的預測演算法類型：best（最相關的相鄰值）、neighbors（相鄰值的加權預測）或 nn-neighbors（非無效相鄰值）。

Netezza Naive Bayes

類型為 netezzanaivebayesnode 的節點的可用內容如下所示：

表 219: netezzanaivebayesnode 內容

netezzanaivebayesnode 內容	值	內容說明
compute_probabilities	旗標	如果為 true，會產生信賴等級（機率）欄位以及預測欄位。
use_m_estimation	旗標	如果為 true，則使用 m-estimation 技術以避免估計期間的零機率。

Netezza KNN

類型為 netezzaknnode 的節點的可用內容如下所示：

表 220: netezzaknnnode 內容

netezzaknnnode 內容	值	內容說明
weights	結構化	這是用於對各個類別指定加權的結構化內容。範例： set :netezzaknnnode.weights = [[drugA 0.3][drugB 0.6]]
distance_measure	Euclidean Manhattan Canberra Maximum	用於測量資料點之間距離的方法。
num_nearest_neighbors	整數	特定觀察值的最近鄰接項數；預設值為 3。
standardize_measurements	旗標	如果為 true，那麼在計算距離值之前，對連續輸入欄位的測量值進行標準化。
use_coresets	旗標	如果為 true，則對大型資料集使用核心集採樣以提高計算速度。

Netezza 分割叢集

類型為 netezzadivclusternode 的節點的可用內容如下所示：

表 221: netezzadivclusternode 內容

netezzadivclusternode 內容	值	內容說明
distance_measure	Euclidean Manhattan Canberra Maximum	用於測量資料點之間距離的方法。
max_iterations	整數	在模型訓練停止前執行的上限演算法疊代次數；預設值為 5。
max_tree_depth	整數	可以將資料集拆分為的上限層次數；預設值為 3。
rand_seed	整數	隨機種子，用於抄寫分析；預設值為 12345。
min_instances_split	整數	可以分割的下限記錄數，預設值為 5。
level	整數	要將記錄分數到的階層層次；預設值為 -1。

Netezza PCA

類型為 netezzapcanode 的節點的可用內容如下所示：

表 222: netezzapcanode 內容

netezzapcanode 內容	值	內容說明
center_data	旗標	如果為 true (預設值), 那麼先執行資料集中 (也稱為「平均數消去法」), 然後再執行分析。
perform_data_scaling	旗標	如果為 true, 那麼在分析前執行資料換算。這樣做可以在以不同的單位測量不同的變數時使分析不那麼任意。
force_eigensolve	旗標	如果為 true, 則使用不太準確但較快的方法來尋找主成份。
pc_number	整數	要將資料集精簡到的主成份數; 預設值為 1。

Netezza 迴歸樹狀結構

類型為 netezzaregtreenode 的節點的可用內容如下所示:

表 223: netezzaregtreenode 內容

netezzaregtreenode 內容	值	內容說明
max_tree_depth	整數	樹狀結構在根節點下方可以增長到的上限層次數; 預設值為 10。
split_evaluation_measure	Variance	類別雜質測量, 用於評估分割樹狀結構的最佳位置, 預設值 (目前唯一選項) 是 Variance。
min_improvement_splits	數字	在樹狀結構中進行新分割前要将雜質減少到的數量下限。
min_instances_split	整數	可以分割的記錄數目下限。
pruning_measure	mse r2 pearson spearman	要使用的刪改方法
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	預設為使用 allTrainingData 來估計模型準確性。使用 partitionTrainingData 來指定要使用的訓練資料百分比, 或 useOtherTable 來使用指定資料庫表格中的訓練資料集。
perc_training_data	數字	如果 prune_tree_options 設定為 PercTrainingData, 則指定用於訓練的資料所佔的百分比。
prune_seed	整數	在 prune_tree_options 設定為 PercTrainingData 時, 用於抄寫分析結果的隨機種子, 預設值是 1。
pruning_table	string	用來估計模型準確性的個別刪改資料集的表格名稱。

表 223: netezzaregtreenode 內容 (繼續)

netezzaregtreenode 內容	值	內容說明
compute_probabilities	旗標	如果為 true，則指定應該併入在輸出中的指定類別的變異。

Netezza 線性迴歸

類型為 netezzalineressionnode 的節點的可用內容如下所示：

表 224: netezzalineressionnode 內容

netezzalineressionnode 內容	值	內容說明
use_svd	旗標	如果為 true，則使用「特異值分解」矩陣代替原始矩陣，以便增加速度和數值精確度。
include_intercept	旗標	如果為 true（預設值），那麼增加解的整體精確度。
calculate_model_diagnostics	旗標	如果為 true，則對模型計算診斷資訊。

Netezza 時間序列

類型為 netezzatimeseriesnode 的節點的可用內容如下所示：

表 225: netezzatimeseriesnode 內容

netezzatimeseriesnode 內容	值	內容說明
time_points	欄位	此輸入欄位包含時間序列的日期值或時間值。
time_series_ids	欄位	此輸入欄位包含時間序列 ID；在輸入包含多個時間序列時使用。
model_table	欄位	這是用於儲存 Netezza 時間序列模型的資料庫表格的名稱。
description_table	欄位	這是包含時間序列名稱和說明的輸入表格的名稱。
seasonal_adjustment_table	欄位	這是一個輸出表格的名稱，該表格用於儲存指數平滑化或週期性趨勢分解演算法所計算的按季度調整值。
algorithm_name	SpectralAnalysis 或 spectral ExponentialSmoothing 或 esmoothing ARIMA SeasonalTrendDecomposition 或 std	這是用於「時間序列」建模的演算法。

表 225: *netezzatimeseriesnode* 內容 (繼續)

netezzatimeseriesnode 內容	值	內容說明
trend_name	N A DA M DM	指數平滑化的趨勢類型： N - 無 A - 可加性 DA - 減幅可加性 M - 相乘性 DM - 減幅相乘性
seasonality_type	N A M	指數平滑化的週期性類型： N - 無 A - 可加性 M - 相乘性
interpolation_method	linear cubicspline exponentialspline	要使用的插補方法。
timerange_setting	SD SP	要使用的時間範圍的設定： SD - 由系統確定（使用時間序列資料的完整範圍） SP - 使用者透過 <code>earliest_time</code> 和 <code>latest_time</code> 指定

表 225: *netezzatimeseriesnode* 內容 (繼續)

netezzatimeseriesnode 內容	值	內容說明
earliest_time	整數	開始值和結束值 (如果 timerange_setting 為 SP) 。
latest_time	日期 時間 時間戳記	格式應遵循 time_points 值。 例如, 如果 time_points 欄位包含日期, 則此項也應該是日期。 範例: <pre> set NZ_DT1.timerange_setting = 'SP' set NZ_DT1.earliest_time = '1921-01-01' set NZ_DT1.latest_time = '2121-01-01'</pre>

表 225: netezzatimeseriesnode 內容 (繼續)

netezzatimeseriesnode 內容	值	內容說明
arima_setting	SD SP	<p>ARIMA 演算法的設定 (僅當 algorithm_name 設定為 ARIMA 時才使用) :</p> <p>SD - 系統決定</p> <p>SP - 使用者指定</p> <p>如果 arima_setting = SP, 請使用下列參數來設定週期性值和非週期性值。範例 (僅非週期性) :</p> <pre>set NZ_DT1.algorithm_name = 'arima' set NZ_DT1.arima_setting = 'SP' set NZ_DT1.p_symbol = 'lesseq' set NZ_DT1.p = '4' set NZ_DT1.d_symbol = 'lesseq' set NZ_DT1.d = '2' set NZ_DT1.q_symbol = 'lesseq' set NZ_DT1.q = '4'</pre>
p_symbol	less	<p>ARIMA - 參數 p、d、q、sp、sd 和 sq 的運算子:</p> <p>less - 少於</p> <p>eq - 等於</p> <p>lesseq - 小於或等於</p>
d_symbol	eq	
q_symbol		
sp_symbol	lesseq	
sd_symbol		
sq_symbol		
p	整數	ARIMA - 自動關聯的非週期性程度。
q	整數	ARIMA - 非週期性衍生值。
d	整數	ARIMA - 模型中的移動平均值階數的非週期性數目。

表 225: *netezzatimeseriesnode* 內容 (繼續)

netezzatimeseriesnode 內容	值	內容說明
sp	整數	ARIMA - 自動關聯的週期性程度。
sq	整數	ARIMA - 週期性衍生值。
sd	整數	ARIMA - 模型中的移動平均值階數的週期性數目。
advanced_setting	SD SP	<p>確定如何處理進階設定：</p> <p>SD - 系統決定</p> <p>SP - 由使用者通過 <code>period</code>、<code>units_period</code> 和 <code>forecast_setting</code> 指定。</p> <p>範例：</p> <pre>set NZ_DT1.advanced_setting = 'SP' set NZ_DT1.period = 5 set NZ_DT1.units_period = 'd'</pre>
period	整數	週期性循環的長度，與 <code>units_period</code> 一起指定。不適用於光譜分析。
units_period	ms s min h d wk q y	<p><code>period</code> 的表示單位：</p> <p>ms - 毫秒</p> <p>s - 秒</p> <p>min - 分鐘</p> <p>h - 小時</p> <p>d - 日</p> <p>wk - 星期</p> <p>q - 季度</p> <p>y - 年</p> <p>例如，對於每週時間序列，請對 <code>period</code> 使用 1，並對 <code>units_period</code> 使用 wk。</p>

表 225: *netezzatimeseriesnode* 內容 (繼續)

netezzatimeseriesnode 內容	值	內容說明
forecast_setting	forecasthorizon forecasttimes	指定如何進行預測。
forecast_horizon	整數 日期 時間 時間戳記	如果 forecast_setting = forecasthorizon, 那麼指定預測結束點值。 格式應遵循 time_points 值。 例如, 如果 time_points 欄位包含日期, 則此項也應該是日期。
forecast_times	整數 日期 時間 時間戳記	如果 forecast_setting = forecasttimes, 那麼指定用於進行預測的值。 格式應遵循 time_points 值。 例如, 如果 time_points 欄位包含日期, 則此項也應該是日期。
include_history	旗標	指示是否將歷程值併入在輸出中。
include_interpolated_values	旗標	指示是否將插入值併入在輸出中。如果 include_history 為 false, 則不適用。

Netezza 廣義線性

類型為 *netezzaglmnode* 的節點的可用內容如下所示:

表 226: *netezzaglmnode* 內容

netezzaglmnode 內容	值	內容說明
dist_family	bernoulli gaussian poisson negativebinomial wald gamma	分佈類型; 預設值為 bernoulli。
dist_params	數字	要使用的分佈參數值。僅當 distribution 為 Negativebinomial 時才適用。

表 226: netezzaglmnode 內容 (繼續)

netezzaglmnode 內容	值	內容說明
trials	整數	僅當 distribution 為 Binomial 時才適用。當目標回應為發生在一組試驗中的事件數目時, target 欄位包含事件數目, trials 欄位包含試驗數。
model_table	欄位	這是用於儲存 Netezza 廣義線性模型的資料庫表格的名稱。
maxit	整數	演算法應執行的疊代數目上限; 預設值為 20。
eps	數字	指定最大錯誤值 (以科學記號表示法表示), 達到此值後, 演算法應停止尋找最適合模型。預設值為 -3, 這表示 1E-3, 即 0.001。
tol	數字	這是一個值 (以科學記號表示法表示), 低於此值的所有錯誤均被視為具有 0 值。預設值為 -7, 表示誤值若低於 1E-7 (或 0.0000001), 則被視為不顯著。

表 226: *netezzaglmnode* 內容 (繼續)

netezzaglmnode 內容	值	內容說明
link_func	identity inverse invnegative invsquare sqrt power oddspower log clog loglog cloglog logit probit gaussit cauchit canbinom cangeom cannegbinom	要使用的鏈結函數；預設值為 logit。
link_params	數字	要使用的鏈結函數參數值。僅當 link_function 為 power 或 oddspower 時才適用。

表 226: netezzaglmnode 內容 (繼續)

netezzaglmnode 內容	值	內容說明
interaction	[[[colnames1],[levels1]], [[colnames2],[levels2]], ...,[colnamesN],[levelsN]],]	指定欄位之間的互動。colnames 是輸入欄位的清單，而 level 對於每個欄位始終為 0。 範例： [[["K", "BP", "Sex", "K"], [0, 0, 0, 0]], [["Age", "Na"], [0, 0]]]
intercept	旗標	如果為 true，則在模型中包含截距。

Netezza 模型區塊內容

Netezza 資料庫模型區塊的一般內容如下所示。

表 227: 一般 Netezza 模型區塊內容

通用 Netezza 模型區塊內容	值	內容說明
connection	string	儲存模型的 Netezza 資料庫的連線字串。
table_name	string	這是用於儲存模型的資料庫表格的名稱。

其他模型區塊內容與相應建模節點的內容相同。

模型區塊的 Script 名稱如下所示。

表 228: Netezza 模型區塊的 Script 名稱

模型區塊	Script 名稱
決策樹狀結構	applynetezzagtreeode
K-Means	applynetezzagkmeansode
Bayes 網路	applynetezzagbayesode
貝式邏輯分類演算法 (Naive Bayes)	applynetezzagnaivebayesode
KNN	applynetezzagknnode
區分叢集	applynetezzagdivclusternode
PCA	applynetezzagpcanode
迴歸樹	applynetezzagregtreeode
線性迴歸	applynetezzaglineregressionode
時間序列	applynetezzagtimeseriesode
廣義線性	applynetezzaglmnode

第 16 章 輸出節點內容

輸出節點的內容與其他類型的節點內容略有不同。輸出節點內容不是參照特定的節點選項，而是儲存對輸出物件的參照。這在從表格中提取值並將其設定為串流參數時非常有用。

本節說明輸出節點的可用的 Scripting 內容。

analysisnode 內容



「分析」節點評估預測模型產生準確預測的能力。「分析」節點執行一個或多個模型塊的預測值和實際值之間的各種比較。它們也可以相互比較預測模型。

範例

```
node = stream.create("analysis", "My node")
# "Analysis" tab
node.setPropertyValue("coincidence", True)
node.setPropertyValue("performance", True)
node.setPropertyValue("confidence", True)
node.setPropertyValue("threshold", 75)
node.setPropertyValue("improve_accuracy", 3)
node.setPropertyValue("inc_user_measure", True)
# "Define User Measure..."
node.setPropertyValue("user_if", "@TARGET = @PREDICTED")
node.setPropertyValue("user_then", "101")
node.setPropertyValue("user_else", "1")
node.setPropertyValue("user_compute", ["Mean", "Sum"])
node.setPropertyValue("by_fields", ["Drug"])
# "Output" tab
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/output/analysis_out.html")
```

表 229: analysisnode 內容

analysisnode 內容	資料類型	內容說明
output_mode	Screen File	用來指定從輸出節點產生之輸出的目標位置。
use_output_name	旗標	指定是否使用自訂輸出名稱。
output_name	string	如果 use_output_name 為 true，請指定要使用的名稱。
output_format	Text (.txt) HTML (.html) Output (.cou)	用來指定輸出類型。
by_fields	清單	
full_filename	string	如果是磁碟、資料或 HTML 輸出，則為輸出檔的名稱。
coincidence	旗標	

表 229: *analysisnode* 內容 (繼續)

analysisnode 內容	資料類型	內容說明
performance	旗標	
evaluation_binary	旗標	
confidence	旗標	
threshold	數字	
improve_accuracy	數字	
field_detection_method	Metadata Name	確定預測欄位與原始目標欄位符合的方式。指定 Metadata 或 Name。
inc_user_measure	旗標	
user_if	<i>expr</i>	
user_then	<i>expr</i>	
user_else	<i>expr</i>	
user_compute	[Mean Sum Min Max SDev]	

dataauditnode 內容



「資料審核」節點將首先全面檢查資料，這些套件含每個欄位的彙總統計量、直方圖和分佈以及有關離群值、遺漏值和極端值的資訊。結果會顯示在易於讀取的矩陣中，該矩陣可以排序並用於產生完整大小的圖形和資料準備節點。

範例

```

filenode = stream.createAt("variablefile", "File", 100, 100)
filenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node = stream.createAt("dataaudit", "My node", 196, 100)
stream.link(filenode, node)
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("fields", ["Age", "Na", "K"])
node.setPropertyValue("display_graphs", True)
node.setPropertyValue("basic_stats", True)
node.setPropertyValue("advanced_stats", True)
node.setPropertyValue("median_stats", False)
node.setPropertyValue("calculate", ["Count", "Breakdown"])
node.setPropertyValue("outlier_detection_method", "std")
node.setPropertyValue("outlier_detection_std_outlier", 1.0)
node.setPropertyValue("outlier_detection_std_extreme", 3.0)
node.setPropertyValue("output_mode", "Screen")
    
```

表 230: *dataauditnode* 內容

dataauditnode 內容	資料類型	內容說明
custom_fields	旗標	
fields	[<i>field1 ... fieldN</i>]	
overlay	欄位	

表 230: dataauditnode 內容 (繼續)

dataauditnode 內容	資料類型	內容說明
display_graphs	旗標	用於開啟或關閉輸出矩陣中圖形的顯示。
basic_stats	旗標	
advanced_stats	旗標	
median_stats	旗標	
calculate	Count Breakdown	用於計算遺漏值。選取兩種計算方法中的一種、兩種，或均不選取。
outlier_detection_method	std iqr	用於指定離群值與極端值的偵測方法。
outlier_detection_std_outlier	數字	如果 outlier_detection_method 是 std, 那麼指定用於定義離群值的數字。
outlier_detection_std_extreme	數字	如果 outlier_detection_method 是 std, 那麼指定用於定義極端值的數字。
outlier_detection_iqr_outlier	數字	如果 outlier_detection_method 是 iqr, 那麼指定用於定義離群值的數字。
outlier_detection_iqr_extreme	數字	如果 outlier_detection_method 是 iqr, 那麼指定用於定義極端值的數字。
use_output_name	旗標	指定是否使用自訂輸出名稱。
output_name	string	如果 use_output_name 為 true, 請指定要使用的名稱。
output_mode	Screen File	用來指定從輸出節點產生之輸出的目標位置。
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	用來指定輸出類型。
paginate_output	旗標	當 output_format 為 HTML 時, 會導致輸出分成數頁。

表 230: dataauditnode 內容 (繼續)

dataauditnode 內容	資料類型	內容說明
lines_per_page	數字	與 paginate_output 搭配使用時，會指定輸出的每頁行數。
full_filename	string	

extensionoutputnode 內容



通過使用您自己的自訂 R 或 Python for Spark Script，可以使用「延伸輸出」節點來分析模型評分的資料和結果。分析的輸出可以是文字或圖形。輸出會新增至管理程式窗格的輸出標籤；或者，輸出可以重新導向至檔案。

Python for Spark 範例

```
##### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_output", "extension_output")
node.setPropertyValue("syntax_type", "Python")

python_script = """
import json
import spss.pyspark.runtime

cxt = spss.pyspark.runtime.getContext()
df = cxt.getSparkInputData()
schema = df.dtypes[:]
print df
"""

node.setPropertyValue("python_syntax", python_script)
```

R 範例

```
##### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_syntax", "print(modelerData$Age)")
```

表 231: extensionoutputnode 內容

extensionoutputnode 內容	資料類型	內容說明
syntax_type	<i>r</i> <i>Python</i>	指定執行哪個 Script – R 或 Python（預設值是 R）。
r_syntax	string	用於模型評分的 R Scripting 語法。
python_syntax	string	用於模型評分的 Python Scripting 語法。
convert_flags	StringsAndDoubles LogicalValues	用於轉換旗標欄位的選項。
convert_missing	旗標	將遺漏值轉換為 R NA 值的選項。

表 231: *extensionoutputnode* 內容 (繼續)

extensionoutputnode 內容	資料類型	內容說明
convert_datetime	旗標	用於將日期或日期時間格式的變數轉換為 R 日期/時間格式的選項。
convert_datetime_class	POSIXct POSIXlt	用於指定將日期或日期時間格式的變數轉換為何種格式的選項。
output_to	Screen File	指定輸出類型 (Screen 或 File)。
output_type	Graph Text	指定是產生圖形還是文字輸出。
full_filename	string	用於所產生輸出的檔案名稱。
graph_file_type	HTML COU	輸出檔案的檔案類型 (.html 或 .cou)。
text_file_type	HTML TEXT COU	指定文字輸出的檔案類型 (.html、.txt 或 .cou)。

kdeexport 內容



Kernel Density Estimation (KDE)[®] 使用 Ball Tree 或 KD Tree 演算法進行有效查詢，並且結合不受監督的學習、功能設計及資料建模的概念。鄰接項型方法 (例如 KDE) 是部分最受歡迎且最有用的密度估計技術。SPSS Modeler 中的 KDE 建模節點和 KDE 模擬節點顯示了 KDE 程式庫的核心功能及常用參數。這些節點是以 Python 來實作的。

表 232: *kdeexport* 內容

kdeexport 內容	資料類型	內容說明
bandwidth	double	預設值是 1。
kernel	string	使用的核心: gaussian 或 tophat。預設值為 gaussian。
algorithm	string	使用的樹狀結構演算法: kd_tree、ball_tree 或 auto。預設值為 auto。
metric	string	計算距離時使用的度量值。對於 kd_tree 演算法，請選擇 Euclidean、Chebyshev、Cityblock、Minkowski、Manhattan、Infinity、P、L2 或 L1。對於 ball_tree 演算法，請選擇 Euclidian、Braycurtis、Chebyshev、Canberra、Cityblock、Dice、Hamming、Infinity、Jaccard、L1、L2、Minkowski、Matching、Manhattan、P、Rogersanimoto、Russellrao、Sokalmichener、Sokalsneath 或 Kulsinski。預設值為 Euclidean。

表 232: *kdeexport* 內容 (繼續)

kdeexport 內容	資料類型	內容說明
atol	<i>Float</i>	結果所需的絕對容錯。通常，容錯較大將導致執行更快。預設值是 0.0。
rtol	<i>Float</i>	結果所需的相對容錯。通常，容錯較大將導致執行更快。預設值是 1E-8。
breadthFirst	<i>boolean</i>	設為 <code>True</code> 以使用寬度優先方法。設為 <code>False</code> 以使用深度優先方法。預設值為 <code>True</code> 。
LeafSize	整數	基礎樹狀結構的葉節點大小。預設值是 40。變更此值可能會顯著影響效能。
pValue	<i>double</i>	如果您將 Minkowski 用於度量值，請指定 P 值。預設值為 1.5。

matrixnode 內容



「矩陣」節點建立一個顯示欄位間關係的表格。它最常用來顯示兩個符號欄位之間的關係，但也可以顯示旗標欄位或數值欄位之間的關係。

範例

```
node = stream.create("matrix", "My node")
# "Settings" tab
node.setPropertyValue("fields", "Numerics")
node.setPropertyValue("row", "K")
node.setPropertyValue("column", "Na")
node.setPropertyValue("cell_contents", "Function")
node.setPropertyValue("function_field", "Age")
node.setPropertyValue("function", "Sum")
# "Appearance" tab
node.setPropertyValue("sort_mode", "Ascending")
node.setPropertyValue("highlight_top", 1)
node.setPropertyValue("highlight_bottom", 5)
node.setPropertyValue("display", ["Counts", "Expected", "Residuals"])
node.setPropertyValue("include_totals", True)
# "Output" tab
node.setPropertyValue("full_filename", "C:/output/matrix_output.html")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("paginate_output", True)
node.setPropertyValue("lines_per_page", 50)
```

表 233: *matrixnode* 內容

matrixnode 內容	資料類型	內容說明
fields	Selected Flags Numerics	
row	欄位	
column	欄位	

表 233: *matrixnode* 內容 (繼續)

matrixnode 內容	資料類型	內容說明
include_missing_values	旗標	指定在列和欄輸出中是否包含使用者遺漏值 (空白) 和系統遺漏值 (空白)。
cell_contents	CrossTabs Function	
function_field	<i>string</i>	
function	Sum Mean Min Max SDev	
sort_mode	Unsorted Ascending Descending	
highlight_top	數字	如果為非零, 則為 True。
highlight_bottom	數字	如果為非零, 則為 True。
display	[Counts Expected Residuals RowPct ColumnPct TotalPct]	
include_totals	旗標	
use_output_name	旗標	指定是否使用自訂輸出名稱。
output_name	<i>string</i>	如果 use_output_name 為 true, 請指定要使用的名稱。
output_mode	Screen File	用來指定從輸出節點產生之輸出的目標位置。

表 233: *matrixnode* 內容 (繼續)

matrixnode 內容	資料類型	內容說明
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	用來指定輸出類型。 Formatted 和 Delimited 格式都可使用修飾元 transposed, 此符號可轉置表格中的列與欄。
paginate_output	旗標	當 output_format 為 HTML 時, 會導致輸出分成數頁。
lines_per_page	數字	與 paginate_output 搭配使用時, 會指定輸出的每頁行數。
full_filename	string	

meansnode 內容



「平均值」節點會比較獨立群組之間的平均值, 或比較相關欄位的配對之間的平均值, 以測試是否存在顯著性差異。例如, 您可以比較開展促銷前後的平均營收, 或者將來自未接受促銷客戶的營收與接受促銷客戶的營收進行比較。

範例

```
node = stream.create("means", "My node")
node.setPropertyValue("means_mode", "BetweenFields")
node.setPropertyValue("paired_fields", [["OPEN_BAL", "CURR_BAL"]])
node.setPropertyValue("label_correlations", True)
node.setPropertyValue("output_view", "Advanced")
node.setPropertyValue("output_mode", "File")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/output/means_output.html")
```

表 234: *meansnode* 內容

meansnode 內容	資料類型	內容說明
means_mode	BetweenGroups BetweenFields	指定要在資料上執行的平均數統計資料的類型。
test_fields	[field1 ... fieldn]	指定當 means_mode 設定為 BetweenGroups 時的檢定欄位。
grouping_field	欄位	指定分組欄位。
paired_fields	[[field1 field2] [field3 field4] ...]	指定當 means_mode 設定為 BetweenFields 時要使用的成對欄位。

表 234: meansnode 內容 (繼續)

meansnode 內容	資料類型	內容說明
label_correlations	旗標	指定在輸出中是否顯示相關係數標籤。僅當 means_mode 設定為 BetweenFields 時才套用此設定。
correlation_mode	Probability Absolute	指定是否依概率或絕對值標示相關性。
weak_label	string	
medium_label	string	
strong_label	string	
weak_below_probability	數字	當 correlation_mode 設為 Probability 時，指定弱相關性的截斷值。這必須是介於 0 到 1 之間的值 - 例如，0.90。
strong_above_probability	數字	強相關性的分割值。
weak_below_absolute	數字	當 correlation_mode 設為 Absolute 時，指定弱相關性的截斷值。這必須是介於 0 到 1 之間的值 - 例如，0.90。
strong_above_absolute	數字	強相關性的分割值。
unimportant_label	string	
marginal_label	string	
important_label	string	
unimportant_below	數字	低欄位重要性的截斷值。這必須是介於 0 到 1 之間的值 - 例如，0.90。
important_above	數字	
use_output_name	旗標	指定是否使用自訂輸出名稱。
output_name	string	使用的名稱。
output_mode	Screen File	指定從輸出節點中產生的輸出的目標位置。
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	指定輸出的類型。
full_filename	string	

表 234: *meansnode* 內容 (繼續)

meansnode 內容	資料類型	內容說明
output_view	Simple Advanced	指定在輸出中顯示簡單視圖還是進階視圖。

reportnode 內容



報告節點可建立格式化報告，其中包含固定文字、資料及得自資料的其他表示式。透過使用文字範本定義固定文字和資料輸出建構，可以指定報告的格式。透過使用範本中的 HTML 標籤和在「輸出」標籤上設定選項，可以提供自訂文字格式。您可以在範本中使用 CLEM 表示式來併入資料值及其他條件式輸出。

範例

```
node = stream.create("report", "My node")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/report_output.html")
node.setPropertyValue("lines_per_page", 50)
node.setPropertyValue("title", "Report node created by a script")
node.setPropertyValue("highlights", False)
```

表 235: *reportnode* 內容

reportnode 內容	資料類型	內容說明
output_mode	Screen File	用來指定從輸出節點產生之輸出的目標位置。
output_format	HTML (.html) Text (.txt) Output (.cou)	用於指定檔案輸出的類型。
format	Auto Custom	用於選擇自動設置輸出格式還是使用範本中包含的 HTML 設置輸出格式。要使用範本中的 HTML 格式設置，請指定 Custom。
use_output_name	旗標	指定是否使用自訂輸出名稱。
output_name	string	如果 use_output_name 為 true，請指定要使用的名稱。
text	string	
full_filename	string	
highlights	旗標	
title	string	
lines_per_page	數字	

rouputnode 內容



「R 輸出」節點可讓您使用您自己的自訂 R Script，來分析資料及模型評分結果。分析的輸出可以是文字或圖形。輸出會新增至管理程式窗格的輸出標籤；或者，輸出可以重新導向至檔案。

表 236: rouputnode 內容

rouputnode 內容	資料類型	內容說明
syntax	string	
convert_flags	StringsAndDoubles LogicalValues	
convert_datetime	旗標	
convert_datetime_class	POSIXct POSIXlt	
convert_missing	旗標	
output_name	Auto Custom	
custom_name	string	
output_to	Screen File	
output_type	Graph Text	
full_filename	string	
graph_file_type	HTML COU	
text_file_type	HTML TEXT COU	

setglobalsnode 內容



設定廣域節點掃描資料並計算可在 CLEM 表示式中使用的彙總值。例如，您可以使用此節點來計算稱為年齡的欄位的統計量，然後透過插入函數 @GLOBAL_MEAN(age)，在 CLEM 表示式中使用年齡的整體平均數。

範例

```
node = stream.create("setglobals", "My node")
node.setKeyedPropertyValue("globals", "Na", ["Max", "Sum", "Mean"])
node.setKeyedPropertyValue("globals", "K", ["Max", "Sum", "Mean"])
node.setKeyedPropertyValue("globals", "Age", ["Max", "Sum", "Mean", "SDev"])
```

```
node.setPropertyValue("clear_first", False)
node.setPropertyValue("show_preview", True)
```

表 237: setglobalsnode 內容

setglobalsnode 內容	資料類型	內容說明
globals	[Sum Mean Min Max SDev]	結構內容，在其中，必須使用下面的語法參照要設定的欄位： node.setKeyedPropertyValue("globals", "Age", ["Max", "Sum", "Mean", "SDev"])
clear_first	旗標	
show_preview	旗標	

simevalnode 內容



「模擬評估」節點對指定的預測目標欄位進行評估，並顯示有關該目標欄位的分佈和相關性資訊。

表 238: simevalnode 內容

simevalnode 內容	資料類型	內容說明
target	欄位	
iteration	欄位	
presorted_by_iteration	boolean	
max_iterations	數字	
tornado_fields	[field1...fieldN]	
plot_pdf	boolean	
plot_cdf	boolean	
show_ref_mean	boolean	
show_ref_median	boolean	
show_ref_sigma	boolean	
num_ref_sigma	數字	
show_ref_pct	boolean	
ref_pct_bottom	數字	
ref_pct_top	數字	
show_ref_custom	boolean	
ref_custom_values	[number1...numberN]	

表 238: *simevalnode* 內容 (繼續)

simevalnode 內容	資料類型	內容說明
category_values	Category Probabilities Both	
category_groups	Categories Iterations	
create_pct_table	<i>boolean</i>	
pct_table	Quartiles Intervals Custom	
pct_intervals_num	數字	
pct_custom_values	<i>[number1...numberN]</i>	

simfitnode 內容

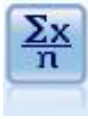


「模擬配適」節點檢查每個欄位中資料的統計分佈，並產生（或更新）「模擬產生」節點，同時將最佳配適分佈分配給每個欄位。然後，可以使用「模擬產生」節點來產生模擬資料。

表 239: *simfitnode* 內容

simfitnode 內容	資料類型	內容說明
build	Node XMLExport Both	
use_source_node_name	<i>boolean</i>	
source_node_name	<i>string</i>	正在產生或更新的來源節點的自訂名稱。
use_cases	All LimitFirstN	
use_case_limit	整數	
fit_criterion	AndersonDarling KolmogorovSmirnov	
num_bins	整數	
parameter_xml_filename	<i>string</i>	
generate_parameter_import	<i>boolean</i>	

statisticsnode 內容



「統計資料」節點可提供有關數值型欄位的基本彙總資訊。它可計算單個欄位以及欄位間的相關性的彙總統計量。

範例

```
node = stream.create("statistics", "My node")
# "Settings" tab
node.setPropertyValue("examine", ["Age", "BP", "Drug"])
node.setPropertyValue("statistics", ["mean", "sum", "sdev"])
node.setPropertyValue("correlate", ["BP", "Drug"])
# "Correlation Labels..." section
node.setPropertyValue("label_correlations", True)
node.setPropertyValue("weak_below_absolute", 0.25)
node.setPropertyValue("weak_label", "lower quartile")
node.setPropertyValue("strong_above_absolute", 0.75)
node.setPropertyValue("medium_label", "middle quartiles")
node.setPropertyValue("strong_label", "upper quartile")
# "Output" tab
node.setPropertyValue("full_filename", "c:/output/statistics_output.html")
node.setPropertyValue("output_format", "HTML")
```

表 240: statisticsnode 內容

statisticsnode 內容	資料類型	內容說明
use_output_name	旗標	指定是否使用自訂輸出名稱。
output_name	string	如果 use_output_name 為 true, 請指定要使用的名稱。
output_mode	Screen File	用來指定從輸出節點產生之輸出的目標位置。
output_format	Text (.txt) HTML (.html) Output (.cou)	用來指定輸出類型。
full_filename	string	
examine	清單	
correlate	清單	
statistics	[count mean sum min max range variance sdev semean median mode]	
correlation_mode	Probability Absolute	指定是否依概率或絕對值標示相關性。
label_correlations	旗標	

表 240: *statisticsnode* 內容 (繼續)

statisticsnode 內容	資料類型	內容說明
<i>weak_label</i>	<i>string</i>	
<i>medium_label</i>	<i>string</i>	
<i>strong_label</i>	<i>string</i>	
<i>weak_below_probability</i>	數字	當 <i>correlation_mode</i> 設為 <i>Probability</i> 時，指定弱相關性的截斷值。這必須是介於 0 到 1 之間的值 - 例如，0.90。
<i>strong_above_probability</i>	數字	強相關性的分割值。
<i>weak_below_absolute</i>	數字	當 <i>correlation_mode</i> 設為 <i>Absolute</i> 時，指定弱相關性的截斷值。這必須是介於 0 到 1 之間的值 - 例如，0.90。
<i>strong_above_absolute</i>	數字	強相關性的分割值。

statisticsoutputnode 內容



「統計量輸出」節點可呼叫 IBM SPSS Statistics 程序以分析您的 IBM SPSS Modeler 資料。可以存取許多不同的 IBM SPSS Statistics 分析程序。此節點需要 IBM SPSS Statistics 的授權副本。

有關此節點內容的資訊，請參閱第 380 頁的『*statisticsoutputnode* 內容』。

tablenode 內容



「表格」節點以表格式顯示資料，這些資料還可以寫入到檔案中。每當您需要檢查資料值或將其匯出為可輕鬆讀取的形式時，此節點非常有用。

範例

```
node = stream.create("table", "My node")
node.setPropertyValue("highlight_expr", "Age > 30")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("transpose_data", True)
node.setPropertyValue("full_filename", "C:/output/table_output.htm")
node.setPropertyValue("paginate_output", True)
node.setPropertyValue("lines_per_page", 50)
```

表 241: *tablenode* 內容

tablenode 內容	資料類型	內容說明
<i>full_filename</i>	<i>string</i>	如果是磁碟、資料或 HTML 輸出，則為輸出檔的名稱。
<i>use_output_name</i>	旗標	指定是否使用自訂輸出名稱。
<i>output_name</i>	<i>string</i>	如果 <i>use_output_name</i> 為 <i>true</i> ，請指定要使用的名稱。

表 241: *tablenode* 內容 (繼續)

tablenode 內容	資料類型	內容說明
output_mode	Screen File	用來指定從輸出節點產生之輸出的目標位置。
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	用來指定輸出類型。
transpose_data	旗標	匯出前跳至資料，使列代表欄位，欄代表記錄。
paginate_output	旗標	當 output_format 為 HTML 時，會導致輸出分成數頁。
lines_per_page	數字	與 paginate_output 搭配使用時，會指定輸出的每頁行數。
highlight_expr	string	
output	string	唯讀內容，可保留對由節點構建的最後一個表格的參照。
value_labels	[[Value LabelString] [Value LabelString] ...]	用來為值配對指定標籤。
display_places	整數	為欄位設定顯示的小數位數（僅用於儲存類型為實數的欄位）。值 -1 將使用串流預設值。
export_places	整數	為欄位設定匯出的小數位數（僅用於儲存類型為實數的欄位）。值 -1 將使用串流預設值。
decimal_separator	DEFAULT PERIOD COMMA	為欄位設定小數點（僅用於儲存類型為實數的欄位）。

表 241: *tablenode* 內容 (繼續)

tablenode 內容	資料類型	內容說明
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	設定欄位的日期格式（僅套用至使用 DATE 或 TIMESTAMP 儲存的欄位）。

表 241: *tablenode* 內容 (繼續)

tablenode 內容	資料類型	內容說明
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	設定欄位的時間格式（僅套用至使用 TIME 或 TIMESTAMP 儲存的欄位）。
column_width	整數	設定欄位的欄寬。值為 -1 表示將直欄寬度設定為 Auto。
justify	AUTO CENTER LEFT RIGHT	設定欄位的直欄對齊。

transformnode 內容



「轉換」節點可讓您選取並以視覺化方式預覽轉換結果，然後再將它們套用至選取的欄位。

範例

```
node = stream.create("transform", "My node")
node.setPropertyValue("fields", ["AGE", "INCOME"])
node.setPropertyValue("formula", "Select")
node.setPropertyValue("formula_log_n", True)
node.setPropertyValue("formula_log_n_offset", 1)
```

表 242: *transformnode* 內容

transformnode 內容	資料類型	內容說明
fields	[<i>field1</i> ... <i>fieldn</i>]	要在轉換中使用的欄位。
formula	All Select	表示應計算所有轉換還是所選轉換。
formula_inverse	旗標	表示是否應使用逆模型轉換。
formula_inverse_offset	數字	表示公式中要使用的資料偏移量。除非使用者指定，否則依預設設定為 0。
formula_log_n	旗標	表示是否應使用 \log_n 轉換。
formula_log_n_offset	數字	
formula_log_10	旗標	表示是否應使用 \log_{10} 變換。
formula_log_10_offset	數字	
formula_exponential	旗標	表示是否應使用指數 (e^x) 變換。
formula_square_root	旗標	表示是否應使用平方根轉換。
use_output_name	旗標	指定是否使用自訂輸出名稱。
output_name	<i>string</i>	如果 <i>use_output_name</i> 為 true，則指定使用的名稱。
output_mode	Screen File	用來指定從輸出節點產生之輸出的目標位置。
output_format	HTML (.html) Output (.cou)	用來指定輸出類型。
paginate_output	旗標	當 <i>output_format</i> 為 HTML 時，會導致輸出分成數頁。
lines_per_page	數字	與 <i>paginate_output</i> 搭配使用時，會指定輸出的每頁行數。
full_filename	<i>string</i>	表示要在檔案輸出中使用的檔案名稱。

第 17 章 匯出節點內容

通用匯出節點內容

下列內容通用於所有匯出節點。

內容	值	內容說明
publish_path	string	輸入供已發佈影像檔和參數檔案使用的根目錄名稱。
publish_metadata	旗標	指定是否生成介紹影像的輸入與輸出以及它們的資料模型的 meta 資料檔。
publish_use_parameters	旗標	指定是否在 *.par 檔案中包含串流參數。
publish_parameters	字串清單	指定要併入的參數。
execute_mode	export_data publish	指定是執行節點而不發佈串流，還是在執行節點時自動發佈串流。

asexport 內容

您可以使用 分析伺服器 匯出在 Hadoop 分散式檔案系統 (HDFS) 上執行串流。

範例

```
node.setPropertyValue("use_default_as", False)  
node.setPropertyValue("connection",  
["false", "9.119.141.141", "9080", "analyticserver", "ibm", "admin", "admin", "false",  
"","","",""])
```

asexport 內容	資料類型	內容說明
data_source	string	資料來源的名稱。
export_mode	string	指定是要將匯出的資料 append 指定給現有資料來源，還是要將 overwrite 指定給現有資料來源。
use_default_as	boolean	如果設為 True，則會使用伺服器 options.cfg 檔案中配置的預設分析伺服器連線。如果設為 False，則使用此節點的連線。

表 244: asexport 內容 (繼續)

asexport 內容	資料類型	內容說明
connection	["string", "string", "string", "string", "string", "string", "string", "string", "string", "string", "string", "string"]	包含 分析伺服器 連線詳細資料的清單內容。格式為： ["is_secure_connect", "server_url", "server_port", "context_root", "consumer", "user_name", "password", "use- kerberos-auth", "kerberos-krb5-config- file-path", "kerberos- jaas-config-file-path", "kerberos-krb5-service- principal-name", "enable- kerberos-debug"] 其中： is_secure_connect: 指出是否使用安全連線，且連線為 true 或 false。use-kerberos-auth: 指出是否使用 kerberos 鑑別，其值為 true 或 false。enable-kerberos-debug: 指出是否使用 kerberos 鑑別的除錯模式，其值為 true 或 false。

cognosexportnode 內容



IBM Cognos 匯出節點以 Cognos 資料庫可以讀取的格式匯出資料。

對於此節點，必須定義 Cognos 連線和 ODBC 連線。

Cognos 連線

Cognos 連線的內容如下。

表 245: cognosexportnode 內容

cognosexportnode 內容	資料類型	內容說明
cognos_connection	["string","flag","string","string","string"]	<p>包含 Cognos 伺服器連線詳細資料的清單內容。格式為： ["Cognos_server_URL", login_mode, "namespace", "username", "password"]</p> <p>其中：</p> <p>Cognos_server_URL 是包含來源之 Cognos 伺服器的 URL。</p> <p>login_mode 指出是否使用匿名登入（為 true 或 false）；如果設為 true，則下列欄位應該設為 ""。</p> <p>namespace 指定用來登入伺服器的安全鑑別提供者。</p> <p>username 和 password 是用來登入 Cognos 伺服器的使用者名稱和密碼。</p> <p>還會提供下列節點而非 login_mode：</p> <ul style="list-style-type: none"> • anonymousMode。例如： ['Cognos_server_url', 'anonymousMode', "namespace", "username", "password"] • credentialMode。例如： ['Cognos_server_url', 'credentialMode', "namespace", "username", "password"]
		<ul style="list-style-type: none"> • storedCredentialMode。例如： ['Cognos_server_url', 'storedCredentialMode', "stored_credential_name"] <p>其中 stored_credential_name 是儲存庫中 Cognos 認證的名稱。</p>

表 245: cognosexportnode 內容 (繼續)

cognosexportnode 內容	資料類型	內容說明
cognos_package_name	string	您要將資料匯出到的 Cognos 套件的路徑和名稱，例如： /Public Folders/MyPackage
cognos_datasource	string	
cognos_export_mode	Publish ExportFile	
cognos_filename	string	

ODBC 連線

ODBC 連線的內容和為下一節中的 databaseexportnode 列出的相同，除了 datasource 內容無效之外。

databaseexportnode 內容



「資料庫匯出」節點將資料寫入與 ODBC 相容的關聯式資料來源。為了寫入 ODBC 資料來源，資料來源必須存在，且您必須具有它的寫入權。

範例

```

'''
Assumes a datasource named "MyDatasource" has been configured
'''
stream = modeler.script.stream()
db_exportnode = stream.createAt("databaseexport", "DB Export", 200, 200)
applyn = stream.findByType("applyneuralnetwork", None)
stream.link(applyn, db_exportnode)

# Export tab
db_exportnode.setPropertyValue("username", "user")
db_exportnode.setPropertyValue("datasource", "MyDatasource")
db_exportnode.setPropertyValue("password", "password")
db_exportnode.setPropertyValue("table_name", "predictions")
db_exportnode.setPropertyValue("write_mode", "Create")
db_exportnode.setPropertyValue("generate_import", True)
db_exportnode.setPropertyValue("drop_existing_table", True)
db_exportnode.setPropertyValue("delete_existing_rows", True)
db_exportnode.setPropertyValue("default_string_size", 32)

# Schema dialog
db_exportnode.setKeyedPropertyValue("type", "region", "VARCHAR(10)")
db_exportnode.setKeyedPropertyValue("export_db_primarykey", "id", True)
db_exportnode.setPropertyValue("use_custom_create_table_command", True)
db_exportnode.setPropertyValue("custom_create_table_command", "My SQL Code")

# Indexes dialog
db_exportnode.setPropertyValue("use_custom_create_index_command", True)
db_exportnode.setPropertyValue("custom_create_index_command", "CREATE BITMAP
INDEX <index-name>
ON <table-name> <(index-columns)>")

```

```
db_exportnode.setKeyValueProperty("indexes", "MYINDEX", [{"fields", ["id", "region"]}]
```

表 246: databaseexportnode 內容

databaseexportnode 內容	資料類型	內容說明
datasource	string	
username	string	
password	string	
epassword	string	在執行過程中，此插槽為唯讀。要產生編碼密碼，請使用「工具」功能表中的「密碼工具」。請參閱第 47 頁的『產生經過編碼的密碼』主題，以取得更多資訊。
table_name	string	
write_mode	Create Append Merge	
map	string	用於將串流欄位名稱對映到資料庫直欄名稱（僅在 write_mode 為 Merge 的情況下才有效）。 對於合併，所有欄位必須經過對映才能匯出。資料庫中不存在的欄位名稱將新增為新欄。
key_fields	清單	指定用作索引鍵的串流欄位；map 內容顯示它在資料庫中的對應項。
join	Database Add	
drop_existing_table	旗標	
delete_existing_rows	旗標	
default_string_size	整數	
type		用來設定綱目類型的結構化內容。
generate_import	旗標	
use_custom_create_table_command	旗標	使用 custom_create_table 屬性來修改標準 CREATE TABLE SQL 指令。
custom_create_table_command	string	指定要代替標準 CREATE TABLE SQL 指令使用的字串指令。
use_batch	旗標	下列內容是用來大量載入資料庫的進階選項。Use_batch 為 true 值時將關閉向資料庫逐列確定的功能。

表 246: *databaseexportnode* 內容 (繼續)

databaseexportnode 內容	資料類型	內容說明
<code>batch_size</code>	數字	指定在向記憶體確定之前傳送給資料庫的記錄數。
<code>bulk_loading</code>	Off ODBC External	指定大量載入的類型。下面列出 ODBC 和 External 的其他選項。
<code>not_logged</code>	旗標	
<code>odbc_binding</code>	Row Column	指定通過 ODBC 批量載入時使用逐列鏈結或逐直欄式鏈結。
<code>loader_delimit_mode</code>	Tab Space Other	對於通過外部程式的批量載入，指定分隔符號的類型。選取 Other 連同 <code>loader_other_delimiter</code> 內容以指定分隔符號，例如逗點 (,)。
<code>loader_other_delimiter</code>	<i>string</i>	
<code>specify_data_file</code>	旗標	旗標為 True 時可啟動下方的 <code>data_file</code> 內容，可在其中指定批量載入到資料庫時所用的檔名以及要寫入的目標路徑。
<code>data_file</code>	<i>string</i>	
<code>specify_loader_program</code>	旗標	旗標為 True 時可啟動下方的 <code>loader_program</code> 內容，在該內容中可以指定外部載入程式 Script 或程式的名稱和位置。
<code>loader_program</code>	<i>string</i>	
<code>gen_logfile</code>	旗標	旗標為 True 值時可啟動下方的 <code>logfile_name</code> ，可在其中指定伺服器上檔案的名稱以產生錯誤日誌。
<code>logfile_name</code>	<i>string</i>	
<code>check_table_size</code>	旗標	旗標為 True 時容許進行表格檢查以確保資料庫表格大小的增加與從 IBM SPSS Modeler 匯出的列個數相符。
<code>loader_options</code>	<i>string</i>	指定載入程式的其他引數，例如 <code>-comment</code> 和 <code>-specialdir</code> 。
<code>export_db_primarykey</code>	旗標	指定給定欄位是否是主要索引鍵。

表 246: *databaseexportnode* 內容 (繼續)

databaseexportnode 內容	資料類型	內容說明
<code>use_custom_create_index_command</code>	旗標	如果標誌為 <code>true</code> ，則為所有指標啟用自訂 SQL。
<code>custom_create_index_command</code>	<i>string</i>	當已啟用自訂 SQL 時，指定用於建立指標的 SQL 指令。（可針對特定索引，置換此值，如下所示。）
<code>indexes.INDEXNAME.fields</code>		必要時建立指定的指標並列出將要包含在該指標中的欄位名稱。
<code>INDEXNAME "use_custom_create_index_command"</code>	旗標	用於啟用或取消特定指標的自訂 SQL。請參閱下表後面的範例。
<code>INDEXNAME "custom_create_index_command"</code>	<i>string</i>	指定用於指定指標的自訂 SQL。請參閱下表後面的範例。
<code>indexes.INDEXNAME.remove</code>	旗標	如果為 <code>True</code> ，那麼將從指標集中移除指定的指標。
<code>table_space</code>	<i>string</i>	指定將建立表格空間。
<code>use_partition</code>	旗標	指定將使用分佈雜湊欄位。
<code>partition_field</code>	<i>string</i>	指定分佈雜湊欄位的內容。

註: 對於部分資料庫，您可以指定建立資料庫表格以進行壓縮匯出（例如，在 SQL 中相當於 `CREATE TABLE MYTABLE (...) COMPRESS YES;`）。為了支援此功能，提供了內容 `use_compression` 和 `compression_mode`，如下所示。

表 247: 使用壓縮功能的 *databaseexportnode* 內容

databaseexportnode 內容	資料類型	內容說明
<code>use_compression</code>	布林	如果設定為 <code>True</code> ，那麼將以匯出時進行壓縮的方式建立表格。

表 247: 使用壓縮功能的 *databaseexportnode* 內容 (繼續)

databaseexportnode 內容	資料類型	內容說明
compression_mode	Row	設定 SQL Server 資料庫的壓縮層次。
	Page	
	Default	設定 Oracle 資料庫的壓縮層次。請注意, 值 OLTP、Query_High、Query_Low、Archive_High 和 Archive_Low 至少需要 Oracle 11gR2。
	Direct_Load_Operations	
	All_Operations	
	Basic	
	OLTP	
	Query_High	
	Query_Low	
Archive_High		
Archive_Low		

顯示如何針對特定指標變更 CREATE INDEX 指令的範例:

```
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX",
["use_custom_create_index_command",
True])db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX",
["custom_create_index_command",
"CREATE BITMAP INDEX <index-name> ON <table-name> <(index-columns)>"])
```

或者, 也可以通過雜湊表完成此操作:

```
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", [{"fields":["id",
"region"],
"use_custom_create_index_command":True,
"custom_create_index_command":"CREATE INDEX <index-name> ON
<table-name> <(index-columns)>"}])
```

datacollectionexportnode 內容



資料收集 匯出節點以資料收集 市場調查軟體使用的格式輸出資料。必須安裝資料收集 資料檔案庫才能使用此節點。

範例

```
stream = modeler.script.stream()
datacollectionexportnode = stream.createAt("datacollectionexport", "Data
Collection", 200, 200)
datacollectionexportnode.setPropertyValue("metadata_file", "c:\\museums.mdd")
datacollectionexportnode.setPropertyValue("merge_metadata", "Overwrite")
```

```

datacollectionexportnode.setPropertyValue("casedata_file", "c:\
\museumdata.sav")
datacollectionexportnode.setPropertyValue("generate_import", True)
datacollectionexportnode.setPropertyValue("enable_system_variables", True)

```

表 248: datacollectionexportnode 內容

datacollectionexportnode 內容	資料類型	內容說明
metadata_file	string	要匯出的 meta 資料檔的名稱。
merge_metadata	Overwrite MergeCurrent	
enable_system_variables	旗標	指定匯出的 .mdd 檔案是否應包含資料收集系統變數。
casedata_file	string	要匯出觀測資料的 .sav 檔案的名稱。
generate_import	旗標	

excelexportnode 內容



Excel 匯出節點會在 Microsoft Excel 中輸出資料。xlsx 檔案格式。您還選用擇在執行完此節點後自動啟動 Excel 並開啟匯出的檔案。

範例

```

stream = modeler.script.stream()
excelexportnode = stream.createAt("excelexport", "Excel", 200, 200)
excelexportnode.setPropertyValue("full_filename", "C:/output/myexport.xlsx")
excelexportnode.setPropertyValue("excel_file_type", "Excel2007")
excelexportnode.setPropertyValue("inc_field_names", True)
excelexportnode.setPropertyValue("inc_labels_as_cell_notes", False)
excelexportnode.setPropertyValue("launch_application", True)
excelexportnode.setPropertyValue("generate_import", True)

```

表 249: excelexportnode 內容

excelexportnode 內容	資料類型	內容說明
full_filename	string	
excel_file_type	Excel2007	
export_mode	Create Append	
inc_field_names	旗標	指定欄位名稱是否應該包含在工作表的第一列中。
start_cell	string	指定匯出的開始 Cell。
worksheet_name	string	要寫入的工作表的名稱。

表 249: *excelexportnode* 內容 (繼續)

excelexportnode 內容	資料類型	內容說明
launch_application	旗標	指定是否應在結果檔案上呼叫 Excel。請注意，必須在 Helper 應用程式對話框（「工具」功能表 > Helper 應用程式）中指定啟動 Excel 的路徑。
generate_import	旗標	指定是否應產生將讀取已匯出資料檔案的 Excel 匯入節點。

extensionexportnode 內容



使用「延伸匯出」節點，您可以執行 R 或 Python for Spark Script 以匯出資料。

Python for Spark 範例

```
##### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_export", "extension_export")
node.setPropertyValue("syntax_type", "Python")

python_script = """import spss.pyspark.runtime
from pyspark.sql import SQLContext
from pyspark.sql.types import *

cxt = spss.pyspark.runtime.getContext()
df = cxt.getSparkInputData()
print df.dtypes[:]
_newDF = df.select("Age", "Drug")
print _newDF.dtypes[:]

df.select("Age", "Drug").write.save("c:/data/ageAndDrug.json", format="json")
"""

node.setPropertyValue("python_syntax", python_script)
```

R 範例

```
##### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_syntax", """write.csv(modelerData, "C:/export.csv)""")
```

表 250: *extensionexportnode* 內容

extensionexportnode 內容	資料類型	內容說明
syntax_type	<i>r</i> <i>Python</i>	指定執行哪個 Script – R 或 Python（預設值是 R）。
r_syntax	<i>string</i>	要執行的 R Scripting 語法。
python_syntax	<i>string</i>	要執行的 Python Scripting 語法。

表 250: *extensionexportnode* 內容 (繼續)

extensionexportnode 內容	資料類型	內容說明
convert_flags	StringsAndDoubles LogicalValues	用於轉換旗標欄位的選項。
convert_missing	旗標	將遺漏值轉換為 R NA 值的選項。
convert_datetime	旗標	用於將日期或日期時間格式的變數轉換為 R 日期/時間格式的選項。
convert_datetime_class	POSIXct POSIXlt	用於指定將日期或日期時間格式的變數轉換為何種格式的選項。

jsonexportnode 內容



JSON 匯出節點輸出 JSON 格式的資料。

表 251: *jsonexportnode* 內容

jsonexportnode 內容	資料類型	內容說明
full_filename	string	完整檔名，包括路徑。
string_format	記錄 值	指定 JSON 字串的格式。預設值是 records。
generate_import	旗標	指定是否應產生將讀取已匯出資料檔案的 JSON 匯入節點。預設值是 False。

outputfilenode 內容



純文字檔案匯出節點可用於將資料輸出到已定界文字檔。它對於匯出可由其他分析或試算表軟體讀取的資料非常有用。

範例

```
stream = modeler.script.stream()
outputfile = stream.createAt("outputfile", "File Output", 200, 200)
outputfile.setPropertyValue("full_filename", "c:/output/flatfile_output.txt")
outputfile.setPropertyValue("write_mode", "Append")
outputfile.setPropertyValue("inc_field_names", False)
outputfile.setPropertyValue("use_newline_after_records", False)
outputfile.setPropertyValue("delimit_mode", "Tab")
outputfile.setPropertyValue("other_delimiter", ",")
outputfile.setPropertyValue("quote_mode", "Double")
outputfile.setPropertyValue("other_quote", "*")
outputfile.setPropertyValue("decimal_symbol", "Period")
outputfile.setPropertyValue("generate_import", True)
```

表 252: *outputfilenode* 內容

outputfilenode 內容	資料類型	內容說明
full_filename	<i>string</i>	輸出檔案的名稱。
write_mode	Overwrite Append	
inc_field_names	旗標	
use_newline_after_records	旗標	
delimit_mode	Comma Tab Space Other	
other_delimiter	<i>char</i>	
quote_mode	None Single Double Other	
other_quote	旗標	
generate_import	旗標	
encoding	StreamDefault SystemDefault "UTF-8"	

sasexportnode 內容



「SAS 匯出」節點可以 SAS 格式輸出資料，以便讀入 SAS 或與 SAS 相容的套裝軟體中。有三種可用的 SAS 檔案格式：SAS for Windows/OS2、SAS for UNIX 或 SAS 第 7/8 版。

範例

```
stream = modeler.script.stream()
sasexportnode = stream.createAt("sasexport", "SAS Export", 200, 200)
sasexportnode.setPropertyValue("full_filename", "c:/output/
SAS_output.sas7bdat")
sasexportnode.setPropertyValue("format", "SAS8")
sasexportnode.setPropertyValue("export_names", "NamesAndLabels")
sasexportnode.setPropertyValue("generate_import", True)
```

表 253: sasexportnode 內容

sasexportnode 內容	資料類型	內容說明
format	Windows UNIX SAS7 SAS8	可變內容標籤欄位。
full_filename	string	
export_names	NamesAndLabels NamesAsLabels	用來將匯出時 IBM SPSS Modeler 中的欄位名稱對映至 IBM SPSS Statistics 或 SAS 變數名稱。
generate_import	旗標	

statisticsexportnode 內容



統計量匯出節點以 IBM SPSS Statistics .sav 或 .zsav 格式輸出資料。IBM SPSS Statistics Base 及其他產品可以讀取 .sav 或 .zsav 檔案。這也是 IBM SPSS Modeler 中用於快取檔案的格式。

有關此節點內容的資訊，請參閱第 381 頁的『[statisticsexportnode 內容](#)』。

tm1odataexport 節點內容



IBM Cognos TM1 匯出節點以 Cognos TM1 資料庫可以讀取的格式匯出資料。

表 254: tm1odataexport 節點內容

tm1odataexport 節點內容	資料類型	內容說明
credential_type	inputCredential 或 storedCredential	用來指出認證類型。
input_credential	清單	當 credential_type 為 inputCredential 時，指定網域、使用者名稱及密碼。
stored_credential_name	string	當 credential_type 為 storedCredential 時；指定 C&DS 伺服器上認證的名稱。
selected_cube	欄位	您要將資料匯出至其中的資料方塊名稱。例如： TM1_export.setPropertyValue("selected_cube", "plan_BudgetPlan")

表 254: tm1odataexport 節點內容 (繼續)

tm1odataexport 節點內容	資料類型	內容說明
spss_field_to_tm1_element_mapping	清單	<p>要對映至的 tm1 元素必須的所選資料方塊視圖的直欄維度。格式為: <code>[[[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...], [[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]]</code></p> <p>有 2 個清單可說明對映資訊。將葉節點元素對映至維度對應於下面的範例 2:</p> <p>範例 1: 第一個清單: (<code>[[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...]</code>) 用於 TM1 維度對映資訊。</p> <p>每個 3 值清單表示維度對映資訊。第三個布林值用來指出它是否選取維度的元素。例如: <code>"[Field_1, Dimension_1, False]"</code> 表示 Field_1 對映至 Dimension_1; <code>"[Element_1, Dimension_2, True]"</code> 表示針對 Dimension_2 選取 Element_1。</p> <p>範例 2: 第二個清單: (<code>[[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]</code>) 用於 TM1 測量維度元素對映資訊。</p> <p>每個 3 值清單表示測量元素對映資訊。第三個布林值用來指出需要建立新元素。<code>"[Field_2, ExistMeasureElement, False]"</code> 表示 Field_2 對映至 ExistMeasureElement; <code>"[Field_3, NewMeasureElement, True]"</code> 表示 NewMeasureElement 必須是 selected_measure 中所選的測量維度, 且 Field_3 對映至它。</p>
selected_measure	<i>string</i>	<p>指定測量維度。</p> <p>範例: <code>setProperty("selected_measure", "Measures")</code></p>
connection_type	AdminServer TM1Server	指出連線類型。預設值為 AdminServer。
admin_host	<i>string</i>	REST API 的主機名稱的 URL。如果 connection_type 是 AdminServer, 則此內容為必要。
server_name	<i>string</i>	從 admin_host 選取的 TM1 伺服器的名稱。如果 connection_type 是 AdminServer, 則此內容為必要。
server_url	<i>string</i>	TM1 伺服器 REST API 的 URL。如果 connection_type 是 TM1Server, 則此內容為必要。

tm1export 節點內容 (已淘汰)



IBM Cognos TM1 匯出節點以 Cognos TM1 資料庫可以讀取的格式匯出資料。

註: 此節點在 Modeler 18.0 中已淘汰。取代節點 Script 名稱是 *tm1odataexport*。

tm1export 節點內容	資料類型	內容說明
pm_host	string	註: 僅適用於 16.0 版及 17.0 版 主機名稱。例如: TM1_export.setPropertyValue("pm_host", 'http://9.191.86.82:9510/pmhub/pm')
tm1_connection	["field","field",...,"field"]	註: 僅適用於 16.0 版及 17.0 版 包含 TM1 伺服器連線詳細資料的清單內容。格式為: ["TM1_Server_Name", "tm1_username", "tm1_password"] 例如: TM1_export.setPropertyValue("tm1_connection", ['Planning Sample', "admin" "apple"])
selected_cube	欄位	您要將資料匯出至其中的資料方塊名稱。例如: TM1_export.setPropertyValue("selected_cube", "plan_BudgetPlan")

表 255: tm1export 節點內容 (繼續)

tm1export 節點內容	資料類型	內容說明
spssfield_tm1element_mapping	清單	<p>要對映至的 tm1 元素必須的所選資料方塊視圖的直欄維度。格式為: <code>[[[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...], [[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]]</code></p> <p>有兩個清單說明對映資訊。將葉節點元素對映至與以下範例 2 對應的維度:</p> <p>範例 1: 第一個清單: (<code>[[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...]</code>) 用於 TM1 維度對映資訊。</p> <p>每個 3 值清單表示維度對映資訊。第三個布林值用來指出它是否選取維度的元素。例如: <code>"[Field_1, Dimension_1, False]"</code> 表示 Field_1 對映至 Dimension_1; <code>"[Element_1, Dimension_2, True]"</code> 表示針對 Dimension_2 選取了 Element_1。</p> <p>範例 2: 第二個清單: (<code>[[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]</code>) 用於 TM1 測量維度元素對映資訊。</p> <p>每個 3 值清單表示測量元素對映資訊。第三個布林值用來指出需要建立新元素。<code>"[Field_2, ExistMeasureElement, False]"</code> 表示 Field_2 對映至 ExistMeasureElement; <code>"[Field_3, NewMeasureElement, True]"</code> 表示 NewMeasureElement 需要是 selected_measure 中選擇的測量維度, 且 Field_3 對映至它。</p>
selected_measure	string	<p>指定測量維度。</p> <p>範例: <code>setProperty("selected_measure", "Measures")</code></p>

xmlexportnode 內容



XML 匯出節點將資料以 XML 格式輸出到檔案。您可以選擇性地建立 XML 來源節點以將匯出的資料讀回串流。

範例

```
stream = modeler.script.stream()
xmlexportnode = stream.createAt("xmlexport", "XML Export", 200, 200)
xmlexportnode.setPropertyValue("full_filename", "c:/export/data.xml")
xmlexportnode.setPropertyValue("map", [{"/catalog/book/genre", "genre"}, {"/
catalog/book/title", "title"}])
```

表 256: *xmlexportnode* 內容

xmlexportnode 內容	資料類型	內容說明
full_filename	<i>string</i>	(必要) XML 匯出檔案的完整路徑和檔案名稱。
use_xml_schema	旗標	指定是否使用 XML 綱目 (XSD 或 DTD 檔案) 控制匯出資料的結構。
full_schema_filename	<i>string</i>	要使用的 XSD 或 DTD 檔案的完整路徑和檔案名稱。如果 <i>use_xml_schema</i> 設為 <i>true</i> , 則為必要。
generate_import	旗標	產生將已匯出資料檔案讀取回到串流中的 XML 來源節點。
records	<i>string</i>	表示記錄界限的 XPath 表示式。
map	<i>string</i>	將欄位名稱對映到 XML 結構。

第 18 章 IBM SPSS Statistics 節點內容

statisticsimportnode 內容



Statistics 檔案節點從 IBM SPSS Statistics 以及儲存在 IBM SPSS Modeler 中的快取檔（也使用同一格式）使用的 .sav 或 .zsav 檔案格式讀取資料。

範例

```
stream = modeler.script.stream()
statisticsimportnode = stream.createAt("statisticsimport", "SAV Import",
200, 200)
statisticsimportnode.setPropertyValue("full_filename", "C:/data/drug1n.sav")
statisticsimportnode.setPropertyValue("import_names", True)
statisticsimportnode.setPropertyValue("import_data", True)
```

表 257: statisticsimportnode 內容

statisticsimportnode 內容	資料類型	內容說明
full_filename	string	完整檔名，包括路徑。
password	string	密碼。password 參數必須在 file_encrypted 參數之前設定。
file_encrypted	旗標	檔案是否受密碼保護。
import_names	NamesAndLabels LabelsAsNames	處理變數名稱和標籤的方法。
import_data	DataAndLabels LabelsAsData	處理值與標籤的方法。
use_field_format_for_storage	布林	指定匯入時是否使用 IBM SPSS Statistics 欄位格式資訊。

statistictransformnode 內容



Statistics 轉換節點針對 IBM SPSS Modeler 中的資料來源執行所選的 IBM SPSS Statistics 語法指令。此節點需要 IBM SPSS Statistics 的授權副本。

範例

```
stream = modeler.script.stream()
statistictransformnode = stream.createAt("statistictransform",
"Transform", 200, 200)
statistictransformnode.setPropertyValue("syntax", "COMPUTE NewVar = Na +
K.")
statistictransformnode.setKeyedPropertyValue("new_name", "NewVar", "Mixed
Drugs")
statistictransformnode.setPropertyValue("check_before_saving", True)
```

表 258: *statisticstransformnode* 內容

statisticstransformnode 內容	資料類型	內容說明
syntax	<i>string</i>	
check_before_saving	旗標	儲存已輸入項之前驗證已輸入的語法。如果語法無效，則會顯示一條錯誤訊息。
default_include	旗標	請參閱第 146 頁的『 filternode 內容 』主題，以取得更多資訊。
include	旗標	請參閱第 146 頁的『 filternode 內容 』主題，以取得更多資訊。
new_name	<i>string</i>	請參閱第 146 頁的『 filternode 內容 』主題，以取得更多資訊。

statisticsmodelnode 內容



Statistics 模型節點使您能夠通過執行生成 PMML 的 IBM SPSS Statistics 程序分析和處理資料。此節點需要 IBM SPSS Statistics 的授權副本。

範例

```
stream = modeler.script.stream()
statisticsmodelnode = stream.createAt("statisticsmodel", "Model", 200, 200)
statisticsmodelnode.setPropertyValue("syntax", "COMPUTE NewVar = Na + K.")
statisticsmodelnode.setKeyedPropertyValue("new_name", "NewVar", "Mixed Drugs")
```

statisticsmodelnode 內容	資料類型	內容說明
syntax	<i>string</i>	
default_include	旗標	請參閱第 146 頁的『 filternode 內容 』主題，以取得更多資訊。
include	旗標	請參閱第 146 頁的『 filternode 內容 』主題，以取得更多資訊。
new_name	<i>string</i>	請參閱第 146 頁的『 filternode 內容 』主題，以取得更多資訊。

statisticsoutputnode 內容



「統計量輸出」節點可呼叫 IBM SPSS Statistics 程序以分析您的 IBM SPSS Modeler 資料。可以存取許多不同的 IBM SPSS Statistics 分析程序。此節點需要 IBM SPSS Statistics 的授權副本。

範例

```
stream = modeler.script.stream()
statisticsoutputnode = stream.createAt("statisticsoutput", "Output", 200, 200)
statisticsoutputnode.setPropertyValue("syntax", "SORT CASES BY Age(A) Sex(A) BP(A) Cholesterol(A)")
```

```

statisticsoutputnode.setPropertyValue("use_output_name", False)
statisticsoutputnode.setPropertyValue("output_mode", "File")
statisticsoutputnode.setPropertyValue("full_filename", "Cases by Age, Sex
and Medical History")
statisticsoutputnode.setPropertyValue("file_type", "HTML")

```

表 259: *statisticsoutputnode* 內容

statisticsoutputnode 內容	資料類型	內容說明
mode	Dialog Syntax	選取「IBM SPSS Statistics 對話框」選項或語法編輯程式
syntax	<i>string</i>	
use_output_name	旗標	
output_name	<i>string</i>	
output_mode	Screen File	
full_filename	<i>string</i>	
file_type	HTML SPV SPW	

statisticsexportnode 內容



統計量匯出節點以 IBM SPSS Statistics *.sav* 或 *.zsav* 格式輸出資料。IBM SPSS Statistics Base 及其他產品可以讀取 *.sav* 或 *.zsav* 檔案。這也是 IBM SPSS Modeler 中用於快取檔案的格式。

範例

```

stream = modeler.script.stream()
statisticsexportnode = stream.createAt("statisticsexport", "Export", 200,
200)
statisticsexportnode.setPropertyValue("full_filename", "c:/output/
SPSS_Statistics_out.sav")
statisticsexportnode.setPropertyValue("field_names", "Names")
statisticsexportnode.setPropertyValue("launch_application", True)
statisticsexportnode.setPropertyValue("generate_import", True)

```

表 260: *statisticsexportnode* 內容

statisticsexportnode 內容	資料類型	內容說明
full_filename	<i>string</i>	

表 260: *statisticsexportnode* 內容 (繼續)

statisticsexportnode 內容	資料類型	內容說明
file_type	sav zsav	以 <i>sav</i> 或 <i>zsav</i> 格式儲存檔案。例如： <code>statisticsexportnode.setPropertyValue("file_type","sav")</code>
encrypt_file	旗標	檔案是否受密碼保護。
password	<i>string</i>	密碼。
launch_application	旗標	
export_names	NamesAndLabels NamesAsLabels	用來將匯出時 IBM SPSS Modeler 中的欄位名稱對映至 IBM SPSS Statistics 或 SAS 變數名稱。
generate_import	旗標	

第 19 章 Python 節點內容

gmm 內容



Gaussian Mixture[®] 模型是一個機率模型，其假設所有資料點都是從混合的有限數量 Gaussian 分佈（參數不明）中產生的。1 可將混合模型視為泛化 k-means 叢集作業，以納入資料收斂結構的相關資訊以及潛在 Gaussians 的中心。SPSS Modeler 中的 Gaussian Mixture 節點顯示了 Gaussian Mixture 程式庫的核心功能及常用參數。該節點是以 Python 來實作的。

表 261: gmm 內容

gmm 內容	資料類型	內容說明
use_partition	<i>boolean</i>	設為 True 或 False 以指定是否使用分割的資料。預設值是 False。
covariance_type	<i>string</i>	指定 Full、Tied、Diag 或 Spherical 來設定收斂類型。
number_component	整數	為混合成分數目指定一個整數。最小值為 1。預設值為 2。
component_lable	<i>boolean</i>	指定 True 以將叢集標籤設為字串，或指定 False 以將叢集標籤設為一個數字。預設值是 False。
label_prefix	<i>string</i>	如果使用的是字串叢集標籤，則可以指定字首。
enable_random_seed	<i>boolean</i>	若要使用隨機種子，請指定 True。預設值是 False。
random_seed	整數	如果使用隨機種子，請指定要用來產生隨機樣本的整數。
tol	<i>double</i>	指定收斂臨界值。預設值為 0.000.1。
max_iter	整數	指定要執行的疊代次數上限。預設值為 100。
init_params	<i>string</i>	設定要使用的起始設定參數。選項為 Kmeans 或 Random。
warm_start	<i>boolean</i>	指定 True 以使用最後一個適合值的求解作為下一次呼叫適合值的起始設定。預設值是 False。

hdbscannode 內容



Hierarchical Density-Based Spatial Clustering (HDBSCAN)[®] 使用非監督式學習來尋找資料集的叢集或密集區域。SPSS Modeler 中的 HDBSCAN 節點顯示了 HDBSCAN 程式庫的核心功能及常用參數。該節點在 Python 中實作，當您一開始不瞭解那是些什麼群組時，您可以使用它來將資料集叢集至不同的群組。

表 262: hdbscannode 內容

hdbscannode 內容	資料類型	內容說明
inputs	欄位	用於叢集作業的輸入欄位。
useHPO	<i>boolean</i>	指定 <code>true</code> 或 <code>false</code> 可基於 Rbfopt 啟用或停用「超參數最佳化 (HPO)」，這會自動探索最佳參數組合，以便模型在樣本上達到預期或更低的誤差率。預設值是 <code>false</code> 。
min_cluster_size	整數	叢集的大小下限。請指定一個整數。預設值為 5。
min_samples	整數	某個點要被視為核心點，其芳鄰中的樣本數。請指定一個整數。如果設定為 0，則會使用 <code>min_cluster_size</code> 。預設值是 0。
algorithm	<i>string</i>	指定要使用哪種演算法: <code>best</code> 、 <code>generic</code> 、 <code>prims_kdtree</code> 、 <code>prims_balltree</code> 、 <code>boruvka_kdtree</code> 或 <code>boruvka_balltree</code> 。預設值為 <code>best</code> 。
metric	<i>string</i>	指定在計算功能陣列中實例之間的距離時要使用的度量值: <code>euclidean</code> 、 <code>cityblock</code> 、 <code>L1</code> 、 <code>L2</code> 、 <code>manhattan</code> 、 <code>braycurtis</code> 、 <code>canberra</code> 、 <code>chebyshev</code> 、 <code>correlation</code> 、 <code>minkowski</code> 或 <code>squeclidean</code> 。預設值是 <code>euclidean</code> 。
useStringLabel	<i>boolean</i>	指定 <code>true</code> 以使用字串叢集標籤，或指定 <code>false</code> 以使用數字叢集標籤。預設值是 <code>false</code> 。
stringLabelPrefix	<i>string</i>	如果 <code>useStringLabel</code> 參數設為 <code>true</code> ，請為字串標籤字首指定一個值。預設字首為 <code>cluster</code> 。
approx_min_span_tree	<i>boolean</i>	指定 <code>true</code> 以接受近似最小跨距樹狀結構，如果您願意為了正確性而犧牲速度，請指定 <code>false</code> 。預設值是 <code>true</code> 。
cluster_selection_method	<i>string</i>	指定從壓縮樹狀結構中選取叢集所使用的方法: <code>eom</code> 或 <code>leaf</code> 。預設值為 <code>eom</code> (Excess of Mass 演算法)。
allow_single_cluster	<i>boolean</i>	如果您要容許單一叢集結果，請指定 <code>true</code> 。預設值是 <code>false</code> 。
p_value	<i>double</i>	如果將 <code>minkowski</code> 用於度量值，請指定要使用的 <code>p value</code> 。預設值為 1.5。
leaf_size	整數	如果使用空間樹狀結構演算法 (<code>boruvka_kdtree</code> 或 <code>boruvka_balltree</code>)，請指定樹狀結構葉節點中的點數。預設值為 40。
outputValidity	<i>boolean</i>	指定 <code>true</code> 或 <code>false</code> 以控制模型輸出中是否包括「有效性指標」圖表。
outputCondensed	<i>boolean</i>	指定 <code>true</code> 或 <code>false</code> 以控制模型輸出中是否包括「壓縮樹狀結構」圖表。

表 262: *hdbscannode* 內容 (繼續)

hdbscannode 內容	資料類型	內容說明
outputSingleLinkage	<i>boolean</i>	指定 true 或 false 以控制模型輸出中是否包括「單一鏈結樹狀結構」圖表。
outputMinSpan	<i>boolean</i>	指定 true 或 false 以控制模型輸出中是否包括「最小跨距樹狀結構」圖表。
is_split		已在 18.2.1.1 版中新增。

kdemodel 內容



Kernel Density Estimation (KDE)[®] 使用 Ball Tree 或 KD Tree 演算法進行有效查詢，並且結合不受監督的學習、功能設計及資料建模的概念。鄰接項型方法（例如 KDE）是部分最受歡迎且最有用的密度估計技術。SPSS Modeler 中的 KDE 建模節點和 KDE 模擬節點顯示了 KDE 程式庫的核心功能及常用參數。這些節點是以 Python 來實作的。

表 263: *kdemodel* 內容

kdemodel 內容	資料類型	內容說明
bandwidth	<i>double</i>	預設值是 1。
kernel	<i>string</i>	使用的核心: gaussian、tophat、epanechnikov、exponential、linear 或 cosine。預設值為 gaussian。
algorithm	<i>string</i>	使用的樹狀結構演算法: kd_tree、ball_tree 或 auto。預設值為 auto。
metric	<i>string</i>	計算距離時使用的度量值。對於 kd_tree 演算法，請選擇 Euclidean、Chebyshev、Cityblock、Minkowski、Manhattan、Infinity、P、L2 或 L1。對於 ball_tree 演算法，請選擇 Euclidian、Braycurtis、Chebyshev、Canberra、Cityblock、Dice、Hamming、Infinity、Jaccard、L1、L2、Minkowski、Matching、Manhattan、P、Rogersanimoto、Russellrao、Sokalmichener、Sokalsneath 或 Kulsinski。預設值為 Euclidean。
atol	<i>Float</i>	結果所需的絕對容錯。通常，容錯較大將導致執行更快。預設值是 0.0。
rtol	<i>Float</i>	結果所需的相對容錯。通常，容錯較大將導致執行更快。預設值是 1E-8。
breadthFirst 從 18.2.1.1 版開始，重新命名為 breadth_first	<i>boolean</i>	設為 True 以使用寬度優先方法。設為 False 以使用深度優先方法。預設值為 True。

表 263: *kdemodel* 內容 (繼續)

kdemodel 內容	資料類型	內容說明
LeafSize 從 18.2.1.1 版開始, 重新命名為 leaf_size	整數	基礎樹狀結構的葉節點大小。預設值是 40。變更此值可能會顯著影響效能。
pValue	<i>double</i>	如果您將 Minkowski 用於度量值, 請指定 P 值。預設值為 1.5。
custom_name		
default_node_name		
use_HPO		

kdeexport 內容



Kernel Density Estimation (KDE)[®] 使用 Ball Tree 或 KD Tree 演算法進行有效查詢, 並且結合不受監督的學習、功能設計及資料建模的概念。鄰接項型方法 (例如 KDE) 是部分最受歡迎且最有用的密度估計技術。SPSS Modeler 中的 KDE 建模節點和 KDE 模擬節點顯示了 KDE 程式庫的核心功能及常用參數。這些節點是以 Python 來實作的。

表 264: *kdeexport* 內容

kdeexport 內容	資料類型	內容說明
bandwidth	<i>double</i>	預設值是 1。
kernel	<i>string</i>	使用的核心: gaussian 或 tophat。預設值為 gaussian。
algorithm	<i>string</i>	使用的樹狀結構演算法: kd_tree、ball_tree 或 auto。預設值為 auto。
metric	<i>string</i>	計算距離時使用的度量值。對於 kd_tree 演算法, 請選擇 Euclidean、Chebyshev、Cityblock、Minkowski、Manhattan、Infinity、P、L2 或 L1。對於 ball_tree 演算法, 請選擇 Euclidian、Braycurtis、Chebyshev、Canberra、Cityblock、Dice、Hamming、Infinity、Jaccard、L1、L2、Minkowski、Matching、Manhattan、P、Rogersanimoto、Russellrao、Sokalmichener、Sokalsneath 或 Kulsinski。預設值為 Euclidean。
atol	<i>Float</i>	結果所需的絕對容錯。通常, 容錯較大將導致執行更快。預設值是 0.0。
rtol	<i>Float</i>	結果所需的相對容錯。通常, 容錯較大將導致執行更快。預設值是 1E-8。
breadthFirst	<i>boolean</i>	設為 True 以使用寬度優先方法。設為 False 以使用深度優先方法。預設值為 True。

表 264: kdeexport 內容 (繼續)

kdeexport 內容	資料類型	內容說明
LeafSize	整數	基礎樹狀結構的葉節點大小。預設值是 40。變更此值可能會顯著影響效能。
pValue	double	如果您將 Minkowski 用於度量值，請指定 P 值。預設值為 1.5。

gmm 內容



Gaussian Mixture[®] 模型是一個機率模型，其假設所有資料點都是從混合的有限數量 Gaussian 分佈（參數不明）中產生的。1 可將混合模型視為泛化 k-means 叢集作業，以納入資料收斂結構的相關資訊以及潛在 Gaussians 的中心。SPSS Modeler 中的 Gaussian Mixture 節點顯示了 Gaussian Mixture 程式庫的核心功能及常用參數。該節點是以 Python 來實作的。

表 265: gmm 內容

gmm 內容	資料類型	內容說明
use_partition	boolean	設為 True 或 False 以指定是否使用分割的資料。預設值是 False。
covariance_type	string	指定 Full、Tied、Diag 或 Spherical 來設定收斂類型。
number_component	整數	為混合成分數目指定一個整數。最小值為 1。預設值為 2。
component_lable	boolean	指定 True 以將叢集標籤設為字串，或指定 False 以將叢集標籤設為一個數字。預設值是 False。
label_prefix	string	如果使用的是字串叢集標籤，則可以指定字首。
enable_random_seed	boolean	若要使用隨機種子，請指定 True。預設值是 False。
random_seed	整數	如果使用隨機種子，請指定要用來產生隨機樣本的整數。
tol	double	指定收斂臨界值。預設值為 0.000.1。
max_iter	整數	指定要執行的疊代次數上限。預設值為 100。
init_params	string	設定要使用的起始設定參數。選項為 Kmeans 或 Random。
warm_start	boolean	指定 True 以使用最後一個適合值的求解作為下一次呼叫適合值的起始設定。預設值是 False。

ocsvmnode 內容



「一類 SVM」節點使用未受監督的學習演算法。該節點可用來偵測新事件。它將偵測給定樣本集的軟性界限，然後將新的點分類成是否的該集合。SPSS Modeler 中的這個一級 SVM 建模節點是在 Python 中實作，並且需要 scikit-learn[®] Python 程式庫。

表 266: ocsvmnode 內容

ocsvmnode 內容	資料類型	內容說明
role_use 從 18.2.1.1 版開始重新命名為 custom_fields	string	指定 predefined 以使用預先定義的角色，或指定 custom 以使用自訂欄位指派。預設為 predefined。
splits	欄位	這是用於分割的欄位名稱清單。
use_partition	布林	指定 true 或 false。預設值是 true。如果設定為 true，那麼在建立模型時，將僅使用訓練資料。
mode_type	string	這是模式。可能的值為 simple 或 expert。如果指定 simple，則將停用「專家」標籤上的所有參數。
stopping_criteria	string	這是科學記號字串。可能的值為 1.0E-1、1.0E-2、1.0E-3、1.0E-4、1.0E-5 或 1.0E-6。預設值為 1.0E-3。
precision	Float	迴歸方法精準度 (nu)。訓練錯誤及支援向量分數的範圍。請指定大於 0 並小於或等於 1.0 的數字。預設值是 0.1。
kernel	string	要在演算法中使用的核心類型。可能的值為 linear、poly、rbf、sigmoid 或 precomputed。預設值為 rbf。
enable_gamma	布林	啟用 gamma 參數。指定 true 或 false。預設值是 true。
gamma	Float	僅針對核心 rbf、poly 及 sigmoid 啟用此參數。如果 enable_gamma 參數設為 false，則此參數將設為 auto。如果設定為 true，那麼預設值是 0.1。
coef0	Float	這是核心函數中的獨立項目。僅針對 poly 核心及 sigmoid 核心啟用此參數。預設值為 0.0。
degree	整數	多項式核心函數的次數。僅針對 poly 核心啟用此參數。指定任何整數。預設值為 3。
shrinking	布林	用於指定是否要使用縮小啟發式選項。指定 true 或 false。預設值是 false。
enable_cache_size	布林	啟用 cache_size 參數。指定 true 或 false。預設值是 false。
cache_size	Float	這是核心快取的大小 (MB)。預設值為 200。
pc_type	string	這是平行座標圖形的類型。可能的選項為 independent 或 general。
lines_amount	整數	這是要包含在圖形上的上限行數。請指定介於 1 和 1000 之間的整數。

表 266: ocsvmnode 內容 (繼續)

ocsvmnode 內容	資料類型	內容說明
lines_fields_custom	布林	用於啟用 lines_fields 參數，您可以通過此參數指定要顯示在圖形輸出中的自訂欄位。如果設定為 false，那麼將顯示所有欄位。如果設定為 true，那麼僅顯示使用 lines_fields 參數指定的欄位。出於效能原因，最多將顯示 20 個欄位。
lines_fields	欄位	這是要作為縱軸包含在圖形上的欄位名稱的清單。
enable_graphic	布林	指定 true 或 false。啟用圖形輸出（如果您要節省時間並減少串流檔大小，則停用此選項）。
enable_hpo	布林	指定 true 或 false 以啟用或停用 HPO 選項。如果設為 true，將套用 Rbfopt 來自動找出「最佳」One-Class SVM 模型，這將達到使用者使用下列 target_objval 參數定義的目標目標值。
target_objval	Float	我們要達到的目標函數值（樣本上的模型誤差率）（例如，不明最佳效果的值）。如果最佳效果不明，請將此參數設為適當的值（例如，0.01）。
max_iterations	整數	用來試驗模型的最大疊代數。預設值為 1000。
max_evaluations	整數	用來試驗模型的最大函數評估數，其中焦點是精確度與速度比。預設值為 300。

rfnode 內容



「隨機森林」節點使用將樹狀結構模型用作基底模型的自助重抽總合法演算法的進階實作。SPSS Modeler 中的此「隨機森林」建模節點在 Python 中實作，並且需要 scikit-learn® Python 程式庫。

表 267: rfnode 內容

rfnode 內容	資料類型	內容說明
role_use	string	指定 predefined 以使用預先定義的角色，或指定 custom 以使用自訂欄位指派。預設為 predefined。
inputs	欄位	這是輸入的欄位名稱。
splits	欄位	這是用於分割的欄位名稱清單。
n_estimators	整數	要建置的樹狀結構數。預設值是 10。
specify_max_depth	布林	指定自訂深度上限。如果為 false，則會展開節點，直到所有葉節點都是純節點，或直到所有葉節點包含的樣本數都少於 min_samples_split。預設值是 false。

表 267: rfnode 內容 (繼續)

rfnode 內容	資料類型	內容說明
max_depth	整數	樹狀結構的深度上限。預設值是 10。
min_samples_leaf	整數	葉節點大小下限。預設值是 1。
max_features	string	尋找最佳分割時要考量的功能數： <ul style="list-style-type: none"> • 如果是 auto, 則分類器為 $\text{max_features}=\sqrt{\text{n_features}}$, 迴歸為 $\text{max_features}=\sqrt{\text{n_features}}$。 • 如果是 sqrt, 則為 $\text{max_features}=\sqrt{\text{n_features}}$。 • 如果是 log2, 則為 $\text{max_features}=\log_2(\text{n_features})$。 預設值是 auto。
bootstrap	布林	建置樹狀結構時使用重複取樣樣本。預設值是 true。
oob_score	布林	使用 out-of-bag 樣本估計一般化精確度。預設值是 false。
extreme	布林	使用極其隨機化的樹狀結構。預設值是 false。
use_random_seed	布林	指定此選項可取得抄寫結果。預設值是 false。
random_seed	整數	建置樹狀結構時要使用的亂數種子。指定任何整數。
cache_size	Float	這是核心快取的大小 (MB)。預設值是 200。
enable_random_seed	布林	用於啟用 random_seed 參數。指定 true 或 false。預設值是 false。
enable_hpo	布林	指定 true 或 false 以啟用或停用 HPO 選項。如果設為 true, 將套用 Rbfopt 來自動判定「最佳」隨機森林模型, 這將達到使用者使用下列 target_objval 參數定義的目標目標值。
target_objval	Float	您要達到的目標函數值 (樣本上的模型誤差率) (例如, 不明最佳效果的值)。如果最佳效果不明, 請將此參數設為適當的值 (例如, 0.01)。
max_iterations	整數	用來試驗模型的最大疊代數。預設值是 1000。
max_evaluations	整數	用來試驗模型的最大函數評估數, 其中焦點是精確度與速度比。預設值是 300。

smotenode 內容



Synthetic Minority Over-sampling Technique (SMOTE) 節點提供一個過度取樣演算法來處理不平衡的資料集。它提供進階方法來平衡資料。SPSS Modeler 中的 SMOTE 程序節點是在 Python 中實作，並且需要 `imbalanced-learn`® Python 程式庫。

表 268: smotenode 內容

smotenode 內容	資料類型	內容說明
target_field 從 18.2.1.1 版開始重新命名為 target	欄位	目標欄位。
sample_ratio	string	用於啟用自訂比例值。兩個選項分別為「自動」(sample_ratio_auto) 和「設定比例」(sample_ratio_manual)。
sample_ratio_value	Float	比例是少數類別中的樣本數除以多數類別中的樣本數。它必須大於 0 並小於或等於 1。預設值為 auto。
enable_random_seed	布林	如果設為 true，則將啟用 random_seed 內容。
random_seed	整數	這是由亂數產生器使用的種子。
k_neighbours	整數	這是要用於構建合成樣本的最近鄰接項的數量。預設值為 5。
m_neighbours	整數	這是要用於確定是否少數樣本處於危險狀態的最近鄰接項的數量。只有在 SMOTE 演算法類型為 borderline1 和 borderline2 時，才啟用此選項。預設值為 10。
algorithm_kind 從 18.2.1.1 版開始重新命名為 algorithm	string	SMOTE 演算法的類型: regular、borderline1 或 borderline2。
usepartition 從 18.2.1.1 版開始重新命名為 use_partition	布林	如果設定為 true，那麼僅將訓練資料用於模型建置。預設值是 true。

tsnenode 內容



t-Distributed Stochastic Neighbor Embedding (t-SNE) 是用於視覺化高維度資料的工具。它會將資料點的親緣性轉換為機率。SPSS Modeler 中的這個 t-SNE 節點使用 Python 進行實作，並且需要 `scikit-learn`® Python 程式庫。

表 269: tsnenode 內容

tsnenode 內容	資料類型	內容說明
mode_type	string	指定 simple 或 expert 模式。

表 269: tsnode 內容 (繼續)

tsnode 內容	資料類型	內容說明
n_components	string	內嵌空間的維度 (2D 或 3D) 。指定 2 或 3。預設值是 2。
method	string	指定 barnes_hut 或 exact。預設值是 barnes_hut。
init	string	內嵌的起始設定。指定 random 或 pca。預設值是 random。
target_field 從 18.2.1.1 版開始重新命名為 target	string	目標欄位名稱。它將是輸出圖形上的顏色分配表。如果未指定目標欄位，則該圖形會使用一種顏色。
perplexity	Float	混雜度與其他多重學習演算法中使用的最近鄰接項數相關。更大的資料集通常需要更大的混雜度。考量選取介於 5 和 50 之間的值。預設值是 30。
early_exaggeration	Float	控制原始空間中的自然叢集在內嵌空間中的緊密程度，以及叢集之間的空間量。預設值是 12.0。
learning_rate	Float	預設值是 200。
n_iter	整數	用於最佳化的最大疊代數。設為至少 250。預設值為 1000。
angle	Float	從點測量的遠距節點的角度大小。指定 0-1 範圍中的值。預設值是 0.5。
enable_random_seed	布林	設為 true 以啟用 random_seed 參數。預設值是 false。
random_seed	整數	要使用的亂數種子。預設值是 None。
n_iter_without_progress	整數	沒有進度的最大疊代數。預設值是 300。
min_grad_norm	string	如果梯度標準低於此臨界值，則會停止最佳化。預設值是 1.0E-7。可能的值為： <ul style="list-style-type: none"> • 1.0E-1 • 1.0E-2 • 1.0E-3 • 1.0E-4 • 1.0E-5 • 1.0E-6 • 1.0E-7 • 1.0E-8
isGridSearch	布林	設為 true 以使用數個不同的混雜度執行 t-SNE。預設值為 false。
output_Rename	布林	如果您要提供自訂名稱，請指定 true，或者指定 false 來自動命名輸出。預設值是 false。

表 269: tsnode 內容 (繼續)

tsnode 內容	資料類型	內容說明
output_to	string	指定 Screen 或 Output。預設值是 Screen。
full_filename	string	指定輸出檔名稱。
output_file_type	string	輸出檔格式。指定 HTML 或 Output object。預設值是 HTML。

xgboostlinearnode 內容



XGBoost Linear[®] 是將線性模型用作基底模型的梯度提升演算法的進階實作。推升法演算法會反覆學習弱分類器，然後將其新增至最終的強分類器。SPSS Modeler 中的 XGBoost Linear 節點使用 Python 進行實作。

表 270: xgboostlinearnode 內容

xgboostlinearnode 內容	資料類型	內容說明
TargetField 從 18.2.1.1 版開始重新命名為 target	欄位	
InputFields 從 18.2.1.1 版開始重新命名為 inputs	欄位	
alpha	double	這是 α 值線性提升系數參數。指定任何數字，0 或更大的數字。預設值是 0。
lambda	double	這是 λ 值線性提升系數參數。指定任何數字，0 或更大的數字。預設值是 1。
lambdaBias	double	這是 λ 值偏移線性提升系數參數。請指定任何數。預設值是 0。
numBoostRound 從 18.2.1.1 版開始重新命名為 num_boost_round	整數	用於模型建置的提高循環數值。指定介於 1 和 1000 之間的值。預設值是 10。
objectiveType	string	用於學習作業的目標類型。可能的值為 reg:linear、reg:logistic、reg:gamma、reg:tweedie、count:poisson、rank:pairwise、binary:logistic 或 multi。請注意，針對旗標目標，僅可以使用 binary:logistic 或 multi。如果使用了 multi，則評分結果會顯示 multi:softmax 及 multi:softprob XGBoost 目標類型。
random_seed	整數	亂數種子。介於 0 和 9999999 之間的任何數字。預設值是 0。

表 270: *xgboostlinearnode* 內容 (繼續)

xgboostlinearnode 內容	資料類型	內容說明
useHPO	布林	指定 <code>true</code> 或 <code>false</code> 以啟用或停用 HPO 選項。如果設為 <code>true</code> ，將套用 <code>Rbfopt</code> 來自動找出「最佳」One-Class SVM 模型，這將達到使用者使用 <code>target_objval</code> 參數定義的目標目標值。

xgboosttreenode 內容



XGBoost Tree[®] 是將樹狀結構模型用作基底模型的梯度提升演算法的進階實現。推升法演算法會反覆學習弱分類器，然後將其新增至最終的強分類器。XGBoost Tree 具有很高的靈活性，並提供了很多對於大多數使用者來說過於複雜的參數，因此 SPSS Modeler 中的 XGBoost Tree 節點僅顯示了核心函數和一般參數。該節點是以 Python 來實作的。

表 271: *xgboosttreenode* 內容

xgboosttreenode 內容	資料類型	內容說明
TargetField 從 18.2.1.1 版開始重新命名為 <code>target</code>	欄位	目標欄位。
InputFields 從 18.2.1.1 版開始重新命名為 <code>inputs</code>	欄位	輸入欄位。
treeMethod 從 18.2.1.1 版開始重新命名為 <code>tree_method</code>	<i>string</i>	模型建置的樹狀結構方法。可能的值為 <code>auto</code> 、 <code>exact</code> 或 <code>approx</code> 。預設值為 <code>auto</code> 。
numBoostRound 從 18.2.1.1 版開始重新命名為 <code>num_boost_round</code>	整數	用於模型建置的提高循環數值。指定介於 1 和 1000 之間的值。預設值是 10。
maxDepth 從 18.2.1.1 版開始重新命名為 <code>max_depth</code>	整數	樹狀結構成長的深度上限。請指定值 1 或更高值。預設值是 6。
minChildWeight 從 18.2.1.1 版開始重新命名為 <code>min_child_weight</code>	<i>double</i>	樹狀結構成長的最小子項加權。請指定值 0 或更高值。預設值是 1。
maxDeltaStep 從 18.2.1.1 版開始重新命名為 <code>max_delta_step</code>	<i>double</i>	樹狀結構成長的上限變化量步驟。請指定值 0 或更高值。預設值是 0。

表 271: xgboosttreenode 內容 (繼續)

xgboosttreenode 內容	資料類型	內容說明
objectiveType 從 18.2.1.1 版開始重新命名為 objective_type	string	用於學習作業的目標類型。可能的值為 reg:linear、reg:logistic、reg:gamma、reg:tweedie、count:poisson、rank:pairwise、binary:logistic 或 multi。請注意，針對旗標目標，僅可以使用 binary:logistic 或 multi。如果使用了 multi，則評分結果會顯示 multi:softmax 及 multi:softprob XGBoost 目標類型。
earlyStopping 從 18.2.1.1 版開始重新命名為 early_stopping	布林	是否使用提早停止功能。預設值是 False。
earlyStoppingRounds 從 18.2.1.1 版開始重新命名為 early_stopping_rounds	整數	驗證錯誤需要至少降低每個提早停止循環以繼續訓練。預設值是 10。
evaluationDataRatio 從 18.2.1.1 版開始重新命名為 evaluation_data_ratio	double	用於驗證錯誤的輸入資料比例。預設值是 0.3。
random_seed	整數	亂數種子。介於 0 和 9999999 之間的任何數字。預設值是 0。
sampleSize 從 18.2.1.1 版開始重新命名為 sample_size	double	用於控制過度契合的子樣本。指定介於 0.1 和 1.0 之間的值。預設值是 0.1。
eta	double	用於控制過度契合的 eta。指定介於 0 和 1 之間的值。預設值是 0.3。
gamma	double	用於控制過度契合的 γ 。指定任何數字，0 或更大的數字。預設值是 6。
colsSampleRatio 從 18.2.1.1 版開始重新命名為 col_sample_ratio	double	用於控制過度契合的 Colsample（依樹狀結構）。指定介於 0.01 和 1 之間的值。預設值是 1。
colsSampleLevel 從 18.2.1.1 版開始重新命名為 col_sample_level	double	用於控制過度契合的 Colsample（依層次）。指定介於 0.01 和 1 之間的值。預設值是 1。
lambda	double	用於控制過度契合的 λ 值。指定任何數字，0 或更大的數字。預設值是 1。
alpha	double	用於控制過度契合的 α 值。指定任何數字，0 或更大的數字。預設值是 0。

表 271: *xgboosttreemode* 內容 (繼續)

xgboosttreemode 內容	資料類型	內容說明
<p>scalePosWeight</p> <p>從 18.2.1.1 版開始重新命名為 scale_pos_weight</p>	<p><i>double</i></p>	<p>用於處理不平衡資料集的比例 POS 加權。 預設值是 1。</p>
<p>use_HPO</p> <p>針對 18.2.1.1 版已新增</p>		

第 20 章 Spark 節點內容

isotonicasnode 內容



Isotonic Regression 屬於迴歸演算法系列。SPSS Modeler 中的 Isotonic-AS 節點在 Spark 中實作。如需 Isotonic Regression 演算法的詳細資料，請參閱 <https://spark.apache.org/docs/2.2.0/mllib-isotonic-regression.html>。

表 272: isotonicasnode 內容

isotonicasnode 內容	資料類型	內容說明
label	string	此內容是為其計算 Isotonic Regression 的應變數。
features	string	此內容是一個自變數。
weightCol	string	此加權代表測量次數。預設值是 1。
isotonic	布林	此內容指出類型是 isotonic 還是 antitonic。
featureIndex	整數	此內容是功能的索引（如果 featuresCol 是向量直欄）。預設值是 0。

kmeansasnode 內容



K-Means 是最常用的叢集演算法之一。它會將資料點叢集到若干預先定義的叢集中。SPSS Modeler 中的 K-Means-AS 節點在 Spark 中實作。如需 K-Means 演算法的詳細資料，請參閱 <https://spark.apache.org/docs/2.2.0/ml-clustering.html>。請注意，K-Means-AS 節點會針對類別變數自動執行 One-Hot 編碼。

表 273: kmeansasnode 內容

kmeansasnode 內容	值	內容說明
roleUse	string	指定 predefined 以使用預先定義的角色，或指定 custom 以使用自訂欄位指派。預設值為 predefined。
autoModel	布林	指定 true 以針對新產生的評分欄位使用預設名稱 (\$S-prediction)，或指定 false 以使用自訂名稱。預設值是 true。
features	欄位	當 roleUse 內容設為 custom 時，輸入的欄位名稱清單。
name	string	當 autoModel 內容設為 false 時，新產生的評分欄位的名稱。
clustersNum	整數	要建立的叢集數。預設值為 5。

表 273: *kmeansasnode* 內容 (繼續)

kmeansasnode 內容	值	內容說明
initMode	<i>string</i>	起始設定演算法。可能的值為 <i>k-means </i> 或 <i>random</i> 。預設值為 <i>k-means </i> 。
initSteps	整數	當 <i>initMode</i> 設為 <i>k-means </i> 時的起始設定步驟數。預設值是 2。
advancedSettings	布林	指定 <i>true</i> 以使下列四個內容可用。預設值是 <i>false</i> 。
maxIteration	整數	用於叢集作業的最大疊代數。預設值為 20。
tolerance	<i>string</i>	用來停止疊代的容錯。可能的設定為 <i>1.0E-1</i> 、 <i>1.0E-2</i> 、...、 <i>1.0E-6</i> 。預設值為 <i>1.0E-4</i> 。
setSeed	布林	指定 <i>true</i> 以使用自訂隨機種子。預設值是 <i>false</i> 。
randomSeed	整數	當 <i>setSeed</i> 內容為 <i>true</i> 時的自訂隨機種子。

multilayerperceptronnode 內容



多層感知器是基於前向人工神經網絡的分類器，它由多層構成。每一層完全連接到網絡中的下一層。SPSS Modeler 中的 *MultiLayerPerceptron-AS* 節點以 Spark 來實作。如需多層感知器分類器 (MLPC) 的相關詳細資料，請參閱 <https://spark.apache.org/docs/latest/ml-classification-regression.html#multilayer-perceptron-classifier>。

表 274: *multilayerperceptronnode* 內容

multilayerperceptronnode 內容	資料類型	內容說明
features	欄位	用作預測的輸入的一或多個欄位。
label	欄位	用作預測的目標的欄位。
layers[0]	整數	要包括的感知器導數目。預設值是 1。
layers[1...<latest-1>]	整數	隱藏層的數目。預設值是 1。
layers[<latest>]	整數	輸出層的數目。預設值是 1。
seed	整數	自訂隨機種子。
maxiter	整數	要執行的最大疊代數。預設值是 10。

xgboostasnode 內容



XGBoost 是梯度提升演算法的進階實作。推升法演算法會反覆學習弱分類器，然後將其新增至最終的強分類器。XGBoost 具有很高的靈活性，並提供了很多對於大多數使用者來說過於複雜的參數，因此 SPSS Modeler 中的 *XGBoost-AS* 節點顯示了核心功能和常用參數。*XGBoost-AS* 節點使用 Spark 進行實作。

表 275: xgboostasnode 內容

xgboostasnode 內容	資料類型	內容說明
target_field	欄位	用於目標的欄位名稱清單。
input_fields	欄位	用於輸入的欄位名稱清單。
nWorkers	整數	用於訓練 XGBoost 模型的工作者數目。預設值是 1。
numThreadPerTask	整數	每個工作者使用的執行緒數。預設值是 1。
useExternalMemory	布林	是否使用外部記憶體作為快取。預設值是 false 。
boosterType	string	要使用的提升係數類型。可用的選項為 gbtree、gblinear 或 dart。預設值是 gbtree。
numBoostRound	整數	用來提升的循環數。指定值 0 或更大的值。預設值是 10。
scalePosWeight	double	用於控制正加權和負加權的平衡。預設值是 1。
randomseed	整數	這是由亂數產生器使用的種子。預設值是 0。
objectiveType	string	學習目標。可能的值為 reg:linear、reg:logistic、reg:gamma、reg:tweedie、rank:pairwise、binary:logistic 或 multi。請注意，針對旗標目標，僅可以使用 binary:logistic 或 multi。如果使用了 multi，則評分結果會顯示 multi:softmax 及 multi:softprob XGBoost 目標類型。預設值是 reg:linear。
evalMetric	string	驗證資料的評估度量值。將根據目標指派預設度量。可能的值為 rmse、mae、logloss、error、merror、mlogloss、auc、ndcg、map 或 gamma-deviance。預設值是 rmse。
lambda	double	這是有關加權的 L2 正規化項目。增大此值將使模型更保守。指定任何數字，0 或更大的數字。預設值是 1。
alpha	double	有關加權的 L1 正規化項目。增大此值將使模型更保守。指定任何數字，0 或更大的數字。預設值是 0。
lambdaBias	double	這是有關偏差的 L2 正規化術語。如果使用 gblinear 提升係數類型，則此 λ 值偏差線性提升係數參數可供使用。指定任何數字，0 或更大的數字。預設值是 0。

表 275: xgboostasnode 內容 (繼續)

xgboostasnode 內容	資料類型	內容說明
treeMethod	string	如果使用 gbtree 或 dart 提升係數類型，則可以使用樹狀結構成長的這個樹狀結構方法參數（以及後面的其他樹狀結構參數）。它指定要使用的 XGBoost 樹狀結構建構演算法。可用的選項為 auto、exact 或 approx。預設值為 auto。
maxDepth	整數	樹狀結構的深度上限。指定值 2 或更大的值。預設值是 6。
minChildWeight	double	子代中需要的實例加權 (hessian) 的總和下限。指定值 0 或更大的值。預設值是 1。
maxDeltaStep	double	容許用於每個樹狀結構的加權估計的最大差異步驟。指定值 0 或更大的值。預設值是 0。
sampleSize	double	子樣本是訓練實例的比例。請指定介於 0.1 和 1.0 之間的值。預設值是 1.0。
eta	double	這是更新步驟期間用於防止過度配適的步驟大小收縮。請指定介於 0 和 1 之間的值。預設值為 0.3。
gamma	double	這是對樹狀結構的某個葉節點進行進一步分割區所需的下限損失減小。指定任何數字，0 或更大的數字。預設值是 6。
colsSampleRatio	double	建構每個樹狀結構時直欄的子樣本比例。請指定介於 0.01 和 1 之間的值。預設值為 1。
colsSampleLevel	double	在每個層次每個分割的直欄的子樣本比例。請指定介於 0.01 和 1 之間的值。預設值是 1。
normalizeType	string	如果使用 dart 提升係數類型，則可以使用此 dart 參數及下列三個 dart 參數。此參數會設定正規化演算法。指定 tree 或 forest。預設值是 tree。
sampleType	string	取樣演算法類型。指定 uniform 或 weighted。預設值是 uniform。
rateDrop	double	dropout 比例 dart 提升係數參數。請指定介於 0.0 和 1.0 之間的值。預設值是 0.0。
skipDrop	double	跳過 dropout 之機率的 dart 提升係數參數。請指定介於 0.0 和 1.0 之間的值。預設值是 0.0。

第 21 章 SuperNode 內容

下表格中說明 SuperNode 的特定內容。注意一般節點內容也可套用至 SuperNode。

內容名稱	內容類型/值清單	內容說明
execute_method	Script Normal	
script	string	

SuperNode 參數

可使用通用格式在 Script 中建立或設定 SuperNode 參數：

```
mySuperNode.setParameterValue("minvalue", 30)
```

您可以使用以下內容擷取參數值：

```
value mySuperNode.getParameterValue("minvalue")
```

尋找現有 SuperNode

您可以使用 `findByType()` 函數在串流中尋找 SuperNode：

```
source_supernode = modeler.script.stream().findByType("source_super", None)
process_supernode = modeler.script.stream().findByType("process_super", None)
terminal_supernode = modeler.script.stream().findByType("terminal_super", None)
```

設定已封裝節點的內容

可以通過存取某個 SuperNode 中的子圖來設定封裝在該 SuperNode 中特定節點的內容。例如，假設有一個來源 SuperNode，其中封裝有「變數檔案」節點以讀取資料。可以通過存取子圖並尋找相關節點來傳送要讀取的檔案的名稱（使用 `full_filename` 內容指定），如下所示：

```
childDiagram = source_supernode.getChildDiagram()
varfilenode = childDiagram.findByType("variablefile", None)
varfilenode.setPropertyValue("full_filename", "c:/mydata.txt")
```

建立 SuperNode

如果您要從頭開始建立 SuperNode 及其內容，那麼可以通過建立 SuperNode、存取子圖並建立所需節點來以類似方式完成此操作。還必須確保 SuperNode 圖中的節點也鏈結至輸入和/或輸出連接器節點。例如，如果您要建立處理 SuperNode：

```
process_supernode = modeler.script.stream().createAt("process_super", "My SuperNode", 200, 200)
childDiagram = process_supernode.getChildDiagram()
filternode = childDiagram.createAt("filter", "My Filter", 100, 100)
childDiagram.linkFromInputConnector(filternode)
childDiagram.linkToOutputConnector(filternode)
```


附錄 A 節點名稱參照

本節會提供 IBM SPSS Modeler 中節點 Scripting 名稱的參照。

模型塊名稱

模型塊（也稱為產生的模型）可以按類型進行參照，像節點和輸出物件一樣。下表會列出模型物件參照名稱。

請注意，這些名稱專門用來參照「模型」選用區（位於 IBM SPSS Modeler 視窗的右上角）中的模型塊。若要參照新增至串流以便評分的模型節點，會使用一組字首為 apply... 的不同名稱。

附註：在正常情況下，建議同時按名稱和類型參照模型以避免混淆。

模式名稱	型號
anomalydetection	Anomaly
Apriori	Apriori
autoclassifier	自動分類器
autocluster	自動叢集
autonumeric	自動數值
bayesnet	貝氏網路
c50	C5.0
carma	Carma
大車	C&R 樹狀結構
chaid	CHAID
coxreg	Cox 迴歸
decisionlist	決策清單
discriminant	區別
因子	PCA/因素
featureselection	功能選擇
genlin	廣義線性迴歸
glimm	GLMM
kmeans	K-Means
knn	k-最近鄰接項
kohonen	Kohonen
線性	線性
logreg	邏輯迴歸
neuralnetwork	神經網路
quest	QUEST

表 277: 模型塊名稱 (建模選用區) (繼續)

模式名稱	型號
迴歸	線性迴歸
序列	序列
slrm	自我學習回應模型
statisticsmodel	IBM SPSS Statistics 模型
svm	支援向量機器
timeseries	時間序列
Twostep	兩步驟

表 278: 模型塊名稱 (資料庫建模選用區)

模式名稱	型號
db2imcluster	IBM ISW 叢集
db2imlog	IBM ISW 邏輯迴歸
db2imnb	IBM ISW Naive Bayes
db2imreg	IBM ISW 迴歸
db2imtree	IBM ISW 決策樹狀結構
msassoc	MS 關聯規則
msbayes	MS 貝式邏輯分類演算法 (Naive Bayes)
mscluster	MS 叢集
mslogistic	MS 邏輯迴歸
msneuralnetwork	MS 神經網路
msregression	MS 線性迴歸
mssequencecluster	MS 順序叢集
mstimeseries	MS 時間序列
mstree	MS 決策樹狀結構
netezzabayes	Netezza Bayes Net
netezzadectree	Netezza 決策樹狀結構
netezzadivcluster	Netezza 分割叢集
netezzaglm	Netezza 廣義線性
netezzakmeans	Netezza K-Means
netezzaknn	Netezza KNN
netezzalinereregression	Netezza 線性迴歸
netezzanaivebayes	Netezza Naive Bayes
netezzapca	Netezza PCA
netezzaregtree	Netezza 迴歸樹狀結構
netezzatimeseries	Netezza 時間序列

模式名稱	型號
oraabn	Oracle 調適性 Bayes
oraai	Oracle AI
oradecisiontree	Oracle 決策樹狀結構
oraglm	Oracle GLM
orakmeans	Oracle k-Means
oranb	Oracle Naive Bayes
oranmf	Oracle NMF
oraocluster	Oracle O-叢集
orasvm	Oracle SVM

避免重複的模型名稱

使用 Script 操作產生的模型時，請注意，容許重複的模型名稱可能會導致參照不明確。為了避免這個問題，最好要求在 Scripting 時針對產生的模型使用唯一名稱。

若要為重複的模型名稱設定選項，請執行下列動作：

1. 從功能表中選擇：
 工具 > 使用者選項
2. 按一下**通知**標籤。
3. 選取**取代前一個模型**以限制針對產生的模型重複命名。

當模型參照不明確時，SPSS Modeler 與 IBM SPSS Collaboration and Deployment Services 在 Script 執行的行為上可能有所差異。SPSS Modeler 用戶端包括選項「取代前一個模型」，這會自動取代具有相同名稱的模型（例如，其中 Script 對迴圈進行疊代以每次產生不同的模型）。然而，在 IBM SPSS Collaboration and Deployment Services 中執行相同的 Script 時，此選項無法使用。您可以為每次疊代中產生的模型重新命名來避免不明確的模型參照，也可以清除現行模型（例如，新增 `clear generated palette` 陳述式），從而避免這種狀況。

輸出類型名稱

下表會列出所有輸出物件類型和建立這些類型的節點。

輸出物件類型	節點 (Node)
analysisoutput	分析
collectionoutput	集合
dataauditoutput	資料審核
distributionoutput	分佈
evaluationoutput	評估
histogramoutput	直方圖
matrixoutput	矩陣
meansoutput	平均數

表 279: 輸出物件類型及建立這些類型的節點 (繼續)

輸出物件類型	節點 (Node)
multiplotoutput	多點
plotoutput	統計圖
qualityoutput	品質
reportdocumentoutput	此物件類型並非來自節點；它是由專案報告建立的輸出
reportoutput	報告
statisticsprocedureoutput	Statistics 輸出
statisticsoutput	統計資料
tableoutput	表格
timeplotoutput	時間繪圖
weboutput	網路

附錄 B 從舊式 Scripting 移轉至 Python Scripting

舊式 Script 移轉概觀

本節會提供在 IBM SPSS Modeler 中 Python 與舊式 Scripting 之間差異的摘要，並提供如何將舊式 Script 移轉至 Python Script 的相關資訊。在本章節中，您會找到 SPSS Modeler 標準舊式指令及等效 Python 指令的清單。

一般差異

舊式 Scripting 的設計大部分都要歸功於 OS 指令 Script。舊式 Scripting 以行導向，儘管存在部分區塊結構，例如，`if...then...else...endif` 及 `for...endfor`，但縮排一般不太顯著。

在 Python Scripting 中，縮排非常顯著，並且必須按相同的層次縮排屬於相同邏輯區塊的行。

註：您在複製及貼上 Python 程式碼時必須小心。使用定位點縮排的行在編輯器中看起來可能與使用空格縮排的行是一樣的。但 Python Script 會產生錯誤，因為不會將這些行視為同樣的縮排。

Scripting 環境定義

Scripting 環境定義會定義 Script 的執行環境，例如執行 Script 的串流或 SuperNode。在舊式 Scripting 中，環境定義是隱含的，這表示，例如，串流 Script 中的任何節點參照都會假定為位於執行 Script 的串流內。

在 Python Scripting 中，Scripting 環境定義是透過 `modeler.script` 模組明確提供的。例如，Python 串流 Script 可以使用下列程式碼存取執行 Script 的串流：

```
s = modeler.script.stream()
```

然後可以透過傳回的物件呼叫串流相關的函數。

指令與函數

舊式 Scripting 以指令導向。這表示每一行 Script 通常以要執行的指令開始，後跟參數，例如：

```
connect 'Type':typenode to :filternode  
rename :derivenode as "Compute Total"
```

Python 使用的函數通常透過定義函數的物件（模組、類別或物件）呼叫，例如：

```
stream = modeler.script.stream()  
typenode = stream.findByType("type", "Type")  
filternode = stream.findByType("filter", None)  
stream.link(typenode, filternode)  
derive.setLabel("Compute Total")
```

文字與註解

在 IBM SPSS Modeler 中常用的部分文字與註解指令在 Python Scripting 中有等效指令。這可協助您將現有的 SPSS Modeler 舊式 Script 轉換為 Python Script 以在 IBM SPSS Modeler 17 中使用。

表 280: 文字與註解的舊式 Scripting 與 Python Scripting 對映

舊式 Scripting	Python Scripting
整數，例如 4	相同
浮點數，例如 0.003	相同

表 280: 文字與註解的舊式 *Scripting* 與 *Python Scripting* 對映 (繼續)

舊式 Scripting	Python Scripting
單引號字串, 例如 'Hello'	相同 註: 包含非 ASCII 字元的字串文字必須以 u 為字首來確保它們表示為 Unicode。
雙引號字串, 例如 "Hello again"	相同 註: 包含非 ASCII 字元的字串文字必須以 u 為字首來確保它們表示為 Unicode。
長字串, 例如 <pre>"""This is a string that spans multiple lines"""</pre>	相同
清單, 例如 [1 2 3]	[1, 2, 3]
變數參照, 例如 set x = 3	x = 3
續行 (\), 例如 <pre>set x = [1 2 \ 3 4]</pre>	<pre>x = [1, 2,\ 3, 4]</pre>
區塊註解, 例如 <pre>/* This is a long comment over a line. */</pre>	<pre>""" This is a long comment over a line. """</pre>
行註解, 例如 set x = 3 # make x 3	x = 3 # make x 3
undef	None
true	True
false	False

運算子

在 IBM SPSS Modeler 中常用的部分運算子指令在 Python Scripting 中有等效指令。這可協助您將現有的 SPSS Modeler 舊式 Script 轉換為 Python Script 以在 IBM SPSS Modeler 17 中使用。

表 281: 運算子的舊式 *Scripting* 與 *Python Scripting* 對映

舊式 Scripting	Python Scripting
<pre>NUM1 + NUM2 LIST + ITEM LIST1 + LIST2</pre>	<pre>NUM1 + NUM2 LIST.append(ITEM) LIST1.extend(LIST2)</pre>
<pre>NUM1 - NUM2 LIST - ITEM</pre>	<pre>NUM1 - NUM2 LIST.remove(ITEM)</pre>
NUM1 * NUM2	NUM1 * NUM2
NUM1 / NUM2	NUM1 / NUM2

表 281: 運算子的舊式 *Scripting* 與 *Python Scripting* 對映 (繼續)

舊式 Scripting	Python Scripting
= ==	==
/= /==	!=
X ** Y	X ** Y
X < Y X <= Y X > Y X >= Y	X < Y X <= Y X > Y X >= Y
X div Y X rem Y X mod Y	X // Y X % Y X % Y
and or not(EXPR)	and or not EXPR

條件句與迴圈

在 IBM SPSS Modeler 中常用的部分條件式及迴圈指令在 *Python Scripting* 中有等效指令。這可協助您將現有的 SPSS Modeler 舊式 Script 轉換為 Python Script 以在 IBM SPSS Modeler 17 中使用。

表 282: 條件句與迴圈的舊式 *Scripting* 與 *Python Scripting* 對映

舊式 Scripting	Python Scripting
<pre>for VAR from INT1 to INT2 ... endfor</pre>	<pre>for VAR in range(INT1, INT2): ...</pre> <p>或</p> <pre>VAR = INT1 while VAR <= INT2: ... VAR += 1</pre>
<pre>for VAR in LIST ... endfor</pre>	<pre>for VAR in LIST: ...</pre>
<pre>for VAR in_fields_to NODE ... endfor</pre>	<pre>for VAR in NODE.getInputDataModel(): ...</pre>
<pre>for VAR in_fields_at NODE ... endfor</pre>	<pre>for VAR in NODE.getOutputDataModel(): ...</pre>

表 282: 條件句與迴圈的舊式 *Scripting* 與 *Python Scripting* 對映 (繼續)

舊式 Scripting	Python Scripting
<pre>if...then ... elseif...then ... else ... endif</pre>	<pre>if ...: ... elif ...: ... else: ... </pre>
<pre>with TYPE OBJECT ... endwith</pre>	無等效項
<pre>var VAR1</pre>	不需要宣告變數

變數

在舊式 *Scripting* 中，參照變數之前會宣告變數，例如：

```
var mynode
set mynode = create typenode at 96 96
```

在 *Python Scripting* 中，第一次參照變數時會建立變數，例如：

```
mynode = stream.createAt("type", "Type", 96, 96)
```

在舊式 *Scripting* 中，必須使用 ^ 運算子明確移除變數參照，例如：

```
var mynode
set mynode = create typenode at 96 96
set ^mynode.direction."Age" = Input
```

與大部分 *Scripting* 語言類似，這對 *Python scripting* 並非必需，例如：

```
mynode = stream.createAt("type", "Type", 96, 96)
mynode.setKeyedPropertyValue("direction", "Age", "Input")
```

節點、輸出及模型類型

在舊式 *Scripting* 中，不同的物件類型（節點、輸出及模型）通常會將類型附加至物件類型。例如，「衍生」節點的類型為 `derivenode`：

```
set feature_name_node = create derivenode at 96 96
```

Python 中的 *IBM SPSS Modeler API* 不包括 `node` 字尾，因此「衍生」節點的類型為 `derive`，例如：

```
feature_name_node = stream.createAt("derive", "Feature", 96, 96)
```

在舊式和 *Python Scripting* 中類型名稱的唯一差異是缺少類型字尾。

內容名稱

舊式和 *Python Scripting* 中的內容名稱都是相同的。例如，在「變數檔案」節點中，在兩個 *Scripting* 環境中定義檔案位置的內容都是 `full_filename`。

節點參照

許多舊式 Script 使用隱式搜尋來尋找並存取要修改的節點。例如，下列指令會在現行串流中搜尋標籤為「Type」的「類型」節點，然後將「Age」欄位的方向（或建模角色）設為「輸入」，將「Drug」欄位的方向設為「目標」，即要預測的值：

```
set 'Type':typenode.direction."Age" = Input
set 'Type':typenode.direction."Drug" = Target
```

在 Python Scripting 中，必須先明確找到節點物件，然後再呼叫函數來設定內容值，例如：

```
typenode = stream.findByType("type", "Type")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
typenode.setKeyedPropertyValue("direction", "Drug", "Target")
```

註：在這種情況下，「Target」字串必須用引號括住。

Python Script 也可以在 `modeler.api` 套件中使用 `ModelingRole` 列舉。

儘管 Python Scripting 版本可能更詳細，但它會帶來最佳的執行時期效能，因為節點搜尋通常僅執行一次。在舊式 Scripting 範例中，會針對每一個指令執行節點搜尋。

此外，還支援依 ID 尋找節點（節點 ID 在節點對話框的「註釋」標籤中顯示）。例如，在舊式 Scripting 中：

```
# id65EMPB9VL87 is the ID of a Type node
set @id65EMPB9VL87.direction."Age" = Input
```

下列 Script 顯示 Python Scripting 中的相同範例：

```
typenode = stream.findByID("id65EMPB9VL87")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
```

取得及設定內容

舊式 Scripting 使用 `set` 指令來指派一個值。`set` 指令後面的項目可以是內容定義。下列 Script 顯示兩種可以用來設定內容的 Script 格式：

```
set <node reference>.<property> = <value>
set <node reference>.<keyed-property>.<key> = <value>
```

在 Python Scripting 中，使用函數 `setPropertyvalue()` 及 `setKeyedPropertyValue()` 可以達到相同的結果，例如：

```
object.setPropertyvalue(property, value)
object.setKeyedPropertyValue(keyed-property, key, value)
```

在舊式 Scripting 中，使用 `get` 指令可以存取內容值，例如：

```
var n v
set n = get node :filternode
set v = ^n.name
```

在 Python Scripting 中，使用函數 `getPropertyvalue()` 可以達到相同的結果，例如：

```
n = stream.findByType("filter", None)
v = n.getPropertyvalue("name")
```

編輯串流

在舊式 Scripting 中，使用 `create` 指令建立新節點，例如：

```
var agg select
set agg = create aggregatenode at 96 96
set select = create selectnode at 164 96
```

在 Python Scripting 中，串流有各種建立節點的方法，例如：

```
stream = modeler.script.stream()
agg = stream.createAt("aggregate", "Aggregate", 96, 96)
select = stream.createAt("select", "Select", 164, 96)
```

在舊式 Scripting 中，使用 `connect` 指令建立節點之間的鏈結，例如：

```
connect ^agg to ^select
```

在 Python Scripting 中，使用 `link` 方法建立節點之間的鏈結，例如：

```
stream.link(agg, select)
```

在舊式 Scripting 中，使用 `disconnect` 指令移除節點之間的鏈結，例如：

```
disconnect ^agg from ^select
```

在 Python Scripting 中，使用 `unlink` 方法移除節點之間的鏈結，例如：

```
stream.unlink(agg, select)
```

在舊式 Scripting 中，使用 `position` 指令將節點置於串流畫布上或其他節點之間，例如：

```
position ^agg at 256 256
position ^agg between ^myselect and ^mydistinct
```

在 Python Scripting 中，使用兩種不同的方法 `setXYPosition` 及 `setPositionBetween` 可以達到相同的結果。例如：

```
agg.setXYPosition(256, 256)
agg.setPositionBetween(myselect, mydistinct)
```

節點作業

在 IBM SPSS Modeler 中常用的部分節點作業指令在 Python Scripting 中有等效指令。這可協助您將現有的 SPSS Modeler 舊式 Script 轉換為 Python Script 以在 IBM SPSS Modeler 17 中使用。

舊式 Scripting	Python Scripting
<code>create nodespec at x y</code>	<pre>stream.create(type, name) stream.createAt(type, name, x, y) stream.createBetween(type, name, preNode, postNode) stream.createModelApplier(model, name)</pre>
<code>connect fromNode to toNode</code>	<pre>stream.link(fromNode, toNode)</pre>
<code>delete node</code>	<pre>stream.delete(node)</pre>
<code>disable node</code>	<pre>stream.setEnabled(node, False)</pre>
<code>enable node</code>	<pre>stream.setEnabled(node, True)</pre>

表 283: 節點作業的舊式 *Scripting* 與 *Python Scripting* 對映 (繼續)

舊式 Scripting	Python Scripting
disconnect <i>fromNode</i> from <i>toNode</i>	<code>stream.unlink(fromNode, toNode)</code> <code>stream.disconnect(node)</code>
duplicate <i>node</i>	<code>node.duplicate()</code>
execute <i>node</i>	<code>stream.runSelected(nodes, results)</code> <code>stream.runAll(results)</code>
flush <i>node</i>	<code>node.flushCache()</code>
position <i>node</i> at x y	<code>node.setXYPosition(x, y)</code>
position <i>node</i> between <i>node1</i> and <i>node2</i>	<code>node.setPositionBetween(node1, node2)</code>
rename <i>node</i> as <i>name</i>	<code>node.setLabel(name)</code>

迴圈

在舊式 *Scripting* 中，支援兩種主要的迴圈選項：

- 計數迴圈，其中索引變數會在兩個整數範圍之間移動。
- 序列迴圈，對值序列進行迴圈，將現行值連結至迴圈變數。

下列 *Script* 是舊式 *Scripting* 中計數迴圈的範例：

```
for i from 1 to 10
  println ^i
endfor
```

下列 *Script* 是舊式 *Scripting* 中序列迴圈的範例：

```
var items
set items = [a b c d]

for i in items
  println ^i
endfor
```

還有其他迴圈類型可以使用：

- 對模型選用區中的模型進行疊代或對輸出選用區中的輸出進行疊代。
- 對進出節點的欄位進行疊代。

Python Scripting 也支援不同的迴圈類型。下列 *Script* 是 *Python Scripting* 中計數迴圈的範例：

```
i = 1
while i <= 10:
  print i
  i += 1
```

下列 *Script* 是 *Python Scripting* 中序列迴圈的範例：

```
items = ["a", "b", "c", "d"]
for i in items:
  print i
```

序列迴圈非常靈活，當其與 IBM SPSS Modeler API 方法結合時，可以支援大部分舊式 Scripting 使用案例。下列範例顯示如何在 Python Scripting 中使用序列迴圈對節點中產生的欄位進行疊代：

```
node = modeler.script.stream().findByType("filter", None)
for column in node.getOutputDataModel().columnIterator():
    print column.getColumnName()
```

執行串流

在串流執行期間，產生的模型或輸出物件會新增至其中一個物件管理程式。在舊式 Scripting 中，Script 必須從物件管理程式中找到建置的物件，或從產生輸出的節點中存取最近產生的輸出。

Python 中的串流執行則有所不同，因為透過執行產生的任何模型或輸出物件都以清單傳回，然後該清單會傳遞至執行函數。這樣存取串流執行的結果就會更為簡單。

舊式 Scripting 支援三個串流執行指令：

- `execute_all` 會執行串流中所有可執行的終端節點。
- `execute_script` 會執行串流 Script，而不管 Script 執行如何設定。
- `execute node` 會執行指定的節點。

Python Scripting 支援一組類似的函數：

- `stream.runAll(results-list)` 會執行串流中所有可執行的終端節點。
- `stream.runScript(results-list)` 會執行串流 Script，而不管 Script 執行如何設定。
- `stream.runSelected(node-array, results-list)` 會按指定節點集的提供順序執行它們。
- `node.run(results-list)` 會執行指定的節點。

在舊式 Script 中，可以使用含選用整數碼的 `exit` 指令終止串流執行，例如：

```
exit 1
```

在 Python Scripting 中，使用下列 Script 可以達到相同的結果：

```
modeler.script.exit(1)
```

透過檔案系統及儲存庫存取物件

在舊式 Scripting 中，您可以使用 `open` 指令開啟現有的串流、模型或輸出物件，例如：

```
var s
set s = open stream "c:/my streams/modeling.str"
```

在 Python Scripting 中，可從階段作業中存取的 `TaskRunner` 類別可以用來執行類似的作業，例如：

```
taskrunner = modeler.script.session().getTaskRunner()
s = taskrunner.openStreamFromFile("c:/my streams/modeling.str", True)
```

若要在舊式 Scripting 中儲存物件，您可以使用 `save` 指令，例如：

```
save stream s as "c:/my streams/new_modeling.str"
```

等效的 Python Script 方法會使用 `TaskRunner` 類別，例如：

```
taskrunner.saveStreamToFile(s, "c:/my streams/new_modeling.str")
```


在舊式 Scripting 中透過 `retrieve` 及 `store` 指令支援基於 IBM SPSS Collaboration and Deployment Services Repository 的作業，例如：

```
var s
set s = retrieve stream "/my repository folder/my_stream.str"
store stream ^s as "/my repository folder/my_stream_copy.str"
```

在 Python Scripting 中，等效的功能將透過與階段作業相關聯的「儲存庫」物件存取：

```
session = modeler.script.session()
repo = session.getRepository()
s = repo.retrieveStream("/my repository folder/my_stream.str", None, None, True)
repo.storeStream(s, "/my repository folder/my_stream_copy.str", None)
```

註：儲存庫存取需要階段作業已配置有效的儲存庫連線。

串流作業

在 IBM SPSS Modeler 中常用的部分串流作業指令在 Python Scripting 中有等效指令。這可協助您將現有的 SPSS Modeler 舊式 Script 轉換為 Python Script 以在 IBM SPSS Modeler 17 中使用。

舊式 Scripting	Python Scripting
<code>create stream DEFAULT_FILENAME</code>	<code>taskrunner.createStream(name, autoConnect, autoManage)</code>
<code>close stream</code>	<code>stream.close()</code>
<code>clear stream</code>	<code>stream.clear()</code>
<code>get stream stream</code>	無等效項
<code>load stream path</code>	無等效項
<code>open stream path</code>	<code>taskrunner.openStreamFromFile(path, autoManage)</code>
<code>save stream as path</code>	<code>taskrunner.saveStreamToFile(stream, path)</code>
<code>retrieve stream path</code>	<code>repository.retrieveStream(path, version, label, autoManage)</code>
<code>store stream as path</code>	<code>repository.storeStream(stream, path, label)</code>

模型作業

在 IBM SPSS Modeler 中常用的部分模型作業指令在 Python Scripting 中有等效指令。這可協助您將現有的 SPSS Modeler 舊式 Script 轉換為 Python Script 以在 IBM SPSS Modeler 17 中使用。

舊式 Scripting	Python Scripting
<code>open model path</code>	<code>taskrunner.openModelFromFile(path, autoManage)</code>
<code>save model as path</code>	<code>taskrunner.saveModelToFile(model, path)</code>
<code>retrieve model path</code>	<code>repository.retrieveModel(path, version, label, autoManage)</code>

表 285: 模型作業的舊式 *Scripting* 與 *Python Scripting* 對映 (繼續)

舊式 Scripting	Python Scripting
<code>store model as path</code>	<code>repository.storeModel(model, path, label)</code>

文件輸出作業

在 IBM SPSS Modeler 中常用的部分文件輸出作業指令在 Python Scripting 中有等效指令。這可協助您將現有的 SPSS Modeler 舊式 Script 轉換為 Python Script 以在 IBM SPSS Modeler 17 中使用。

表 286: 文件輸出作業的舊式 *Scripting* 與 *Python Scripting* 對映

舊式 Scripting	Python Scripting
<code>open output path</code>	<code>taskrunner.openDocumentFromFile(path, autoManage)</code>
<code>save output as path</code>	<code>taskrunner.saveDocumentToFile(output, path)</code>
<code>retrieve output path</code>	<code>repository.retrieveDocument(path, version, label, autoManage)</code>
<code>store output as path</code>	<code>repository.storeDocument(output, path, label)</code>

舊式 *Scripting* 與 *Python Scripting* 之間的其他差異

舊式 Script 可為操作 IBM SPSS Modeler 專案提供支援。Python Scripting 目前對此不提供支援。

舊式 Scripting 可為載入狀態物件（串流及模型的組合）提供部分支援。自 IBM SPSS Modeler 8.0 以來已淘汰狀態物件。Python Scripting 不支援狀態物件。

Python Scripting 可提供舊式 Scripting 中無法使用的其他功能，如下所示：

- 類別和函數定義
- 錯誤處理
- 更複雜的輸入/輸出支援
- 外部及第三方模組

注意事項

本資訊係針對 IBM 在美國所提供之產品與服務所開發。IBM 可能會以其他語言提供本材料。不過，您可以要求擁有一份該語言的產品或產品版本以取用它。

在其他國家，IBM 不見得有提供本文件所提及之各項產品、服務或功能。請洽詢當地的 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。本文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 之智慧財產權，任何功能相當之產品、程式或服務皆可取代 IBM 之產品、程式或服務。不過，任何非 IBM 之產品、程式或服務，使用者必須自行負責作業之評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。提供本文件不代表提供這些專利的授權。您可以書面提出授權查詢，來函請寄到：

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

International Business Machines Corporation 只依「現況」提供本出版品，不提供任何明示或默示之保證，其中包括且不限於不侵權、可商用性或特定目的之適用性的隱含保證。有些司法管轄區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合您。

本資訊中可能會有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。同時，IBM 得隨時改進並（或）變動本書中所提及的產品及（或）程式。

本書對於非 IBM 網站的援引只是為了方便而提供，並不對這些網站作任何認可。這些網站所提供的資料不是 IBM 本產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

IBM 得以各種 IBM 認為適當的方式使用或散布 貴客戶提供的任何資訊，而無需對 貴客戶負責。

如果本程式之獲授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於 IBM 客戶合約、IBM 國際程式授權合約或雙方之任何同等合約的條款，提供本文件所提及的授權程式與其所有適用的授權資料。

所引用的客戶範例為說明用途。實際的績效會因不同的配置與作業狀況而異。

本書所提及之非 IBM 產品資訊，係一由產品的供應商，或其出版的聲明或其他公開管道取得。IBM 未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性、或任何對產品的其他主張是否完全無誤。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

有關 IBM 未來動向的陳述，僅代表 IBM 的目標而已，並可能於未事先聲明的情況下有所變動或撤回。

此資訊包含日常企業運作所使用的資料和報告的範例。為了盡可能完整地說明，範例中包括了個人、公司行號、品牌以及產品等的名稱。所有這些名稱全為虛構，任何與實際個人或商場企業類似之處，純屬巧合。

商標

IBM、IBM 標誌及 `ibm.com` 是 International Business Machines Corp. 的商標或註冊商標。在全球許多司法管轄區註冊。其他產品與服務名稱可能是 IBM 或其他公司的商標。如需最新的 IBM 商標清單，請造訪「著作權及商標資訊」網站，網址是 www.ibm.com/legal/copytrade.shtml。

Adobe、Adobe 標誌、PostScript 及 PostScript 標誌是 Adobe Systems Incorporated 在美國及（或）其他國家或地區的註冊商標或商標。

Intel、Intel 標誌、Intel Inside、Intel Inside 標誌、Intel Centrino、Intel Centrino 標誌、Celeron、Intel Xeon、Intel SpeedStep、Itanium 及 Pentium 是 Intel Corporation 或其子公司在美國及其他國家或地區的商標或註冊商標。

Linux 是 Linus Torvalds 在美國及（或）其他國家或地區的註冊商標。

Microsoft、Windows、Windows NT 及 Windows 標誌是 Microsoft Corporation 在美國及（或）其他國家或地區的商標。

UNIX 是 The Open Group 在美國及其他國家或地區的註冊商標。

Java 和所有以 Java 為基礎的商標和標誌是 Oracle 及/或其附屬機構的商標或註冊商標。

產品說明文件的條款

這些出版品的使用許可權，係遵循下列條款而授與。

適用性

這些條款係附加於 IBM 網站的任何使用條款上。

個人使用

貴客戶可以為了非商務性的私人用途而複製這些出版品，但必須保留所有專利注意事項。未經 IBM 明示同意，您不得散佈、展示或改作該等「出版品」或其任何部分。

商業用途

貴客戶可僅在企業內部重製、散佈及顯示這些出版品，但必須保留全部的所有權聲明。未經 IBM 明示同意，您不得改作該等「出版品」，也不得於企業外複製、散佈或展示該等「出版品」或其任何部分。

權限

除了本項許可權所明確授予者之外，並未明示或暗示授予出版品或任何資訊、資料、軟體或其中的其他智慧財產的任何其他許可權、授權或權利。

IBM 保留在判定出版品的使用將損害其利益或判定未適當遵守上述指示時，撤銷此處所授予之許可權的權利。

除非完全符合所有適當的法律和規章，其中包括所有美國輸出法律和規章，否則，貴客戶不能下載、輸出或再輸出本項資訊。

IBM 不提供這些出版品內容的任何保證。這些出版品係「依現狀」提供，無任何形式（明示或暗示）的擔保，包括但不限於對適售性、無侵權、符合特定使用目的的暗示保證。

索引

Special Characters

- 「全體」節點
 - 內容 [144](#)
- 「地圖視覺化」節點
 - 內容 [179](#)
- 「空間-時間預測」節點
 - 內容 [268](#)
- 「相關規則」節點塊
 - 內容 [298](#)
- 「模擬生成」節點
 - 內容 [95](#)
- 「模擬配適」節點
 - 內容 [353](#)
- 「模擬評估」節點
 - 內容 [352](#)
- 「選取」節點
 - 內容 [122](#)
- 「IBM SPSS 統計量匯出」節點
 - 內容 [381](#)
- 「TWC 匯入」來源節點
 - 內容 [99](#)
- 一級 SVM 節點
 - 內容 [387](#)
- 已編碼的密碼
 - 新增至 Script [47](#)
- 中斷 Script [9](#)
- 內容
 - 串流 [67](#)
 - 通用 Script 編寫 [65](#)
 - 資料庫建模節點 [315](#)
 - 過濾節點 [63](#)
 - Scripting [63-65](#), [191](#), [297](#), [361](#)
 - SuperNode [401](#)
- 分析節點
 - 內容 [341](#)
- 分組節點
 - 內容 [138](#)
- 分割區節點
 - 內容 [148](#)
- 引數
 - 伺服器連線 [60](#)
 - 系統 [58](#)
 - 指令檔 [62](#)
 - IBM SPSS Analytic Server 儲存庫連接 [61](#)
 - IBM SPSS Collaboration and Deployment Services Repository 連線 [61](#)
- 支援向量機器模型
 - 節點 Scripting 內容 [274](#), [311](#)
- 代碼區塊 [17](#)
- 功能選擇模型
 - 套用 [4](#)
 - 節點 Scripting 內容 [222](#), [305](#)
 - Scripting [4](#)
- 平均數節點
 - 內容 [348](#)
- 平衡節點
 - 平衡節點 (繼續)
 - 內容 [108](#)
 - 合併節點
 - 內容 [116](#)
 - 在串流中迴圈 [5, 6](#)
 - 地理空間來源節點
 - 內容 [94](#)
 - 多媒體串流時間序列節點
 - 內容 [130](#)
 - 多媒體串流時間序列模型
 - 節點 Scripting 內容 [124](#)
 - 多集指令 [63](#)
 - 多圖節點
 - 內容 [182](#)
 - 字串
 - 轉換觀察值 [45](#)
 - 字串函數 [45](#)
 - 存取串流執行的結果
 - 表格內容模型 [49](#)
 - JSON 內容模型 [52](#)
 - XML 內容模型 [51](#)
 - 存取串流執行結果
 - 表格內容模型 [49](#)
 - JSON 內容模型 [52](#)
 - XML 內容模型 [51](#)
 - 安全
 - 已編碼的密碼 [47, 60](#)
 - 自我學習回應模型
 - 節點 Scripting 內容 [267](#), [310](#)
 - 自動分類器節點
 - 節點 Scripting 內容 [197](#)
 - 自動分類器模型
 - 節點 Scripting 內容 [298](#)
 - 自動式資料準備
 - 內容 [134](#)
 - 自動數值模型
 - 節點 Scripting 內容 [201](#), [300](#)
 - 自動叢集節點
 - 節點 Scripting 內容 [199](#)
 - 自動叢集模型
 - 節點 Scripting 內容 [300](#)
 - 串流
 - 內容 [67](#)
 - 多集指令 [63](#)
 - 修改 [28](#)
 - 迴圈 [5, 6](#)
 - 執行 [25](#)
 - 條件式執行 [5, 8](#)
 - Scripting [1, 25](#)
 - 串流的條件執行 [5, 8](#)
 - 串流執行順序
 - 隨 Script 變更 [45](#)
 - 伺服器
 - 指令行引數 [60](#)
 - 位置參數 [4, 63, 65](#)
 - 作業 [13](#)
 - 決策清單模型

- 決策清單模型 (繼續)
 - 節點 Scripting 內容 [215](#), [302](#)
- 系統
 - 指令行引數 [58](#)
- 貝式網路模型
 - 節點 Scripting 內容 [203](#), [300](#)
- 使用者輸入節點
 - 內容 [100](#)
- 來源節點
 - 內容 [71](#)
- 函數
 - 文件輸出作業 [416](#)
 - 文字 [407](#)
 - 串流作業 [415](#)
 - 物件參照 [407](#)
 - 迴圈 [409](#)
 - 條件句 [409](#)
 - 註解 [407](#)
 - 節點作業 [412](#)
 - 運算子 [408](#)
 - 模型作業 [415](#)
- 固定檔案節點
 - 內容 [90](#)
- 定向 Web 節點
 - 內容 [189](#)
- 定義方法 [22](#)
- 定義類別 [21](#)
- 定義屬性 [22](#)
- 延伸匯入節點
 - 內容 [88](#)
- 延伸匯出節點
 - 內容 [370](#)
- 延伸模型節點
 - 節點 Scripting 內容 [218](#)
- 延伸輸出節點
 - 內容 [344](#)
- 延伸轉換節點
 - 內容 [115](#)
- 物件導向 [21](#)
- 直方圖節點
 - 內容 [178](#)
- 表格內容模型 [49](#)
- 表格節點
 - 內容 [355](#)
- 附加節點
 - 內容 [107](#)
- 非 ASCII 字元 [20](#)
- 建立節點 [28-30](#)
- 建立類別 [21](#)
- 建模節點
 - 節點 Scripting 內容 [191](#)
- 指令行
 - 引數清單 [58](#), [60](#), [61](#)
 - 多個引數 [62](#)
 - 參數 [59](#)
 - 執行 IBM SPSS Modeler [57](#)
 - Scripting [48](#)
- 衍生節點
 - 內容 [141](#)
- 重組節點
 - 內容 [151](#)
- 重新分類節點
 - 內容 [149](#)
- 重新投影節點
 - 重新投影節點 (繼續)
 - 內容 [150](#)
 - 修改串流 [28](#), [30](#)
 - 差異節點
 - 內容 [113](#)
 - 座標系統重新投影
 - 內容 [150](#)
 - 時間因果模型
 - 節點 Scripting 內容 [275](#)
 - 時間序列模型
 - 節點 Scripting 內容 [280](#), [288](#), [312](#)
 - 時間間隔節點
 - 內容 [153](#)
 - 時間圖節點
 - 內容 [185](#)
 - 矩陣節點
 - 內容 [346](#)
 - 神經網路
 - 節點 Scripting 內容 [257](#), [308](#)
 - 神經網路模型
 - 節點 Scripting 內容 [254](#), [308](#)
 - 純文字檔案節點
 - 內容 [371](#)
 - 迴圈
 - 在 Script 中使用 [45](#)
 - 配送節點
 - 內容 [171](#)
 - 匿名節點
 - 內容 [133](#)
 - 區別元件模型
 - 節點 Scripting 內容 [216](#), [303](#)
 - 區塊
 - 節點 Scripting 內容 [297](#)
 - 參照節點
 - 設定內容 [27](#)
 - 尋找節點 [26](#)
 - 參數
 - Scripting [13](#)
 - SuperNode [401](#)
 - 執行 Script [9](#)
 - 執行串流 [25](#)
 - 執行順序
 - 隨 Script 變更 [45](#)
 - 密碼
 - 已編碼 [60](#)
 - 新增至 Script [47](#)
 - 排序節點
 - 內容 [122](#)
 - 清單 [14](#)
 - 產生的模型
 - Scripting 名稱 [403](#), [405](#)
 - 產生的關鍵字 [48](#)
 - 異常偵測模型
 - 節點 Scripting 內容 [191](#), [297](#)
 - 移轉
 - 一般差異 [407](#)
 - 內容名稱 [410](#)
 - 存取物件 [414](#)
 - 函數 [407](#)
 - 取得內容 [411](#)
 - 指令 [407](#)
 - 迴圈 [413](#)
 - 執行串流 [414](#)
 - 清除串流, 輸出, 和模型管理程式 [31](#)

- 移轉 (繼續)
 - 細項 [416](#)
 - 設定內容 [411](#)
 - 概觀 [407](#)
 - 節點參照 [411](#)
 - 節點類型 [410](#)
 - 模式類型 [410](#)
 - 編輯串流 [412](#)
 - 輸出類型 [410](#)
 - 檔案系統 [414](#)
 - 變數 [410](#)
 - repository [414](#)
 - Scripting 環境定義 [407](#)
- 設定內容 [27](#)
- 設定為旗標節點
 - 內容 [152](#)
- 設定廣域節點
 - 內容 [351](#)
- 陳述式 [16](#)
- 備註 [16](#)
- 最近的鄰接模型
 - 節點 Scripting 內容 [242](#)
- 報告節點
 - 內容 [350](#)
- 尋找節點 [26](#)
- 結構化內容 [63](#)
- 統計量節點
 - 內容 [354](#)
- 評估節點
 - 內容 [172](#)
- 集合節點
 - 內容 [170](#)
- 順序模型
 - 節點 Scripting 內容 [266, 311](#)
- 傳送引數 [17](#)
- 匯出節點
 - 節點 Scripting 內容 [361](#)
- 填入器節點
 - 內容 [145](#)
- 新增屬性 [21](#)
- 節點
 - 名稱參照 [403](#)
 - 在 Script 中迴圈 [45](#)
 - 刪除 [30](#)
 - 取代 [30](#)
 - 取消鏈結節點 [29](#)
 - 匯入 [30](#)
 - 資訊 [31](#)
 - 鏈結節點 [29](#)
- 節點 Scripting 內容
 - 建模節點 [191](#)
 - 匯出節點 [361](#)
 - 模型塊 [297](#)
- 資料收集來源節點
 - 內容 [83](#)
- 資料收集匯出節點
 - 內容 [368](#)
- 資料庫建模 [315](#)
- 資料庫匯出節點
 - 內容 [364](#)
- 資料庫節點
 - 內容 [81](#)
- 資料審核節點
 - 內容 [342](#)
- 跳至節點
 - 內容 [159](#)
- 遍訪節點 [30](#)
- 過濾器節點
 - 內容 [146](#)
- 圖形板節點
 - 內容 [174](#)
- 圖形節點
 - Scripting 內容 [169](#)
- 聚集節點
 - 內容 [107](#)
- 廣義線性模型
 - 節點 Scripting 內容 [224, 305](#)
- 數學方法 [18](#)
- 模型
 - Scripting 名稱 [403, 405](#)
- 模型物件
 - Scripting 名稱 [403, 405](#)
- 模型塊
 - 節點 Scripting 內容 [297](#)
 - Scripting 名稱 [403, 405](#)
- 模擬產生節點
 - 內容 [95](#)
- 模擬評估節點
 - 內容 [352](#)
- 樣本節點
 - 內容 [120](#)
- 範例 [18](#)
- 線性 AS 內容 [247](#)
- 線性 AS 模型
 - 節點 Scripting 內容 [247, 307](#)
- 線性內容 [245](#)
- 線性支援向量機器模型
 - 節點 Scripting 內容 [253, 308](#)
- 線性迴歸模型
 - 節點 Scripting 內容 [263, 310](#)
- 線性模型
 - 節點 Scripting 內容 [245, 307](#)
- 歷程節點
 - 內容 [147](#)
- 獨立式 Script [1, 3, 25](#)
- 輸出物件
 - Scripting 名稱 [405](#)
- 輸出節點
 - Scripting 內容 [341](#)
- 錯誤檢查
 - Scripting [48](#)
- 隨機森林節點
 - 內容 [389](#)
- 隨機樹狀結構模型
 - 節點 Scripting 內容 [261, 310](#)
- 隱藏變數 [22](#)
- 轉換節點
 - 內容 [358](#)
- 繪圖節點
 - 內容 [183](#)
- 關聯規則節點
 - 內容 [194](#)
- 類型節點
 - 內容 [161](#)
- 繼承 [22](#)
- 欄位
 - 在 Script 中關閉 [169](#)
- 欄位名稱

欄位名稱 (繼續)
轉換觀察值 [45](#)
欄位重新排序節點
內容 [149](#)
疊代鍵
在 Script 中迴圈 [7](#)
疊代變數
在 Script 中迴圈 [7](#)
變數
Scripting [13](#)
變數檔案節點
內容 [101](#)
邏輯迴歸模型
節點 Scripting 內容 [248](#), [307](#)

A

aggregatenode 內容 [107](#)
analysisnode 內容 [341](#)
Analytic Server 來源節點
內容 [78](#)
anomalydetectionnode 內容 [191](#)
anonymizenode 內容 [133](#)
appendnode 內容 [107](#)
applyanomalydetectionnode 內容 [297](#)
applyapriorinode 內容 [297](#)
applyassociationrulesnode 內容 [298](#)
applyautoclassifiernode 內容 [298](#)
applyautoclusternode 內容 [300](#)
applyautonumericnode 內容 [300](#)
applybayesnetnode 內容 [300](#)
applyc50node 內容 [301](#)
applycarmanode 內容 [301](#)
applycartnode 內容 [301](#)
applychaidnode 內容 [301](#)
applycoxregnode 內容 [302](#)
applydecisionlistnode 內容 [302](#)
applydiscriminantnode 內容 [303](#)
applyextension 內容 [303](#)
applyfactornode 內容 [304](#)
applyfeatureselectionnode 內容 [305](#)
applygeneralizedlinearnode 內容 [305](#)
applygle 內容 [306](#)
applyglmnode 內容 [305](#)
applygmm 內容 [306](#)
applykmeansnode 內容 [306](#)
applyknnnode 內容 [306](#)
applykohonennode 內容 [307](#)
applylinearnode 內容 [307](#)
applylogregnode 內容 [307](#)
applysvmnode 內容 [308](#)
applymslogisticnode 內容 [317](#)
applymsneuralnetworknode 內容 [317](#)
applymsregressionnode 內容 [317](#)
applymssequenceclusternode 內容 [317](#)
applymstimeseriesnode 內容 [317](#)
applymstreenode 內容 [317](#)
applynetezabayesnode 內容 [340](#)
applynetezadectreenode 內容 [340](#)
applynetezadivclusternode 內容 [340](#)
applynetezakmeansnode 內容 [340](#)
applynetezaknnnode 內容 [340](#)
applynetezalineregressionnode 內容 [340](#)

applynetezanaivebayesnode 內容 [340](#)
applynetezzapcanode 內容 [340](#)
applynetezzaregtreenode 內容 [340](#)
applyneuralnetnode 內容 [308](#)
applyneuralnetworknode 內容 [308](#)
applyocsvm 內容 [309](#)
applyoraabnnode 內容 [326](#)
applyoradecisiontreenode 內容 [326](#)
applyorakmeansnode 內容 [326](#)
applyoranbnnode 內容 [326](#)
applyoranmfnode 內容 [326](#)
applyoraoclusternode 內容 [326](#)
applyorasvmnode 內容 [326](#)
applyquestnode 內容 [309](#)
applyr 內容 [310](#)
applyrandomtrees 內容 [310](#)
applyregressionnode 內容 [310](#)
applyselflearningnode 內容 [310](#)
applysequencenode 內容 [311](#)
applystpnode 內容 [311](#)
applysvmnode 內容 [311](#)
applytcmnode 內容 [311](#)
applytimeseriesnode 內容 [312](#)
applytreeas 內容 [312](#)
applyts 內容 [312](#)
applytwostepAS 內容 [313](#)
applytwostepnode 內容 [313](#)
applyxgboostlinearnode 內容 [313](#)
applyxgboosttreenode 內容 [313](#)
apriori 模型
節點 Scripting 內容 [193](#), [297](#)
apriorinode 內容 [193](#)
AS 「時間間隔」節點
內容 [138](#)
asexport 內容 [361](#)
asimport 內容 [78](#)
associationrulesnode 內容 [194](#)
astimeintervalsnode 內容 [138](#)
autoclassifiernode 內容 [197](#)
autoclusternode 內容 [199](#)
autodatapreprenode 內容 [134](#)
autonumericnode 內容 [201](#)

B

balancenode 內容 [108](#)
bayesnet 內容 [203](#)
binningnode 內容 [138](#)
buildr 內容 [204](#)

C

C&R 樹狀結構模型
節點 Scripting 內容 [207](#), [301](#)
C5.0 模型
節點 Scripting 內容 [205](#), [301](#)
c50node 內容 [205](#)
CARMA 模型
節點 Scripting 內容 [206](#), [301](#)
carmanode 內容 [206](#)
cartnode 內容 [207](#)
CHAID 模型
節點 Scripting 內容 [210](#), [301](#)

chaidnode 內容 [210](#)
clear generated palette 指令 [48](#)
CLEM
 Scripting [1](#)
cognosimport 節點內容 [79](#)
collectionnode 內容 [170](#)
Cox 迴歸模型
 節點 Scripting 內容 [212](#), [302](#)
coxregnode 內容 [212](#)
CPLEX 最佳化節點
 內容 [109](#)
cplexoptnode 內容 [109](#)

D

dataauditnode 內容 [342](#)
databaseexportnode 內容 [364](#)
databasenode 內容 [81](#)
datacollectionexportnode 內容 [368](#)
datacollectionimportnode 內容 [83](#)
decisionlist 內容 [215](#)
derive_stbnode
 內容 [111](#)
derivnode 內容 [141](#)
diagrams [25](#)
directedwebnode 內容 [189](#)
discriminantnode 內容 [216](#)
distinctnode 內容 [113](#)
distributionnode 內容 [171](#)

E

E-Plot 節點
 內容 [187](#)
ensemblenode 內容 [144](#)
eplotnode 內容 [187](#)
evaluationnode 內容 [172](#)
Excel 來源節點
 內容 [86](#)
Excel 匯出節點
 內容 [369](#), [371](#)
excelexportnode 內容 [369](#), [371](#)
excelimportnode 內容 [86](#)
exportModelToFile [38](#)
extensionexportnode 內容 [370](#)
extensionimportnode 內容 [88](#)
extensionmodelnode 內容 [218](#)
extensionoutputnode 內容 [344](#)
extensionprocessnode 內容 [115](#)

F

factornode 內容 [220](#)
featureselectionnode 內容 [4](#), [222](#)
fillernode 內容 [145](#)
filternode 內容 [146](#)
fixedfilenode 內容 [90](#)
flags
 指令行引數 [57](#)
 組合多個旗標 [62](#)
flatfilenode 內容 [371](#)
for 指令 [45](#)

G

Gaussian Mixture 節點
 內容 [383](#), [387](#)
genlinnode 內容 [224](#)
gle 內容 [233](#)
GLE 模型
 節點 Scripting 內容 [233](#), [306](#)
GLMM 模型
 節點 Scripting 內容 [229](#), [305](#)
glmmnode 內容 [229](#)
gmm 內容 [383](#), [387](#)
graphboardnode 內容 [174](#)
gsdata_import 節點內容 [94](#)

H

HDBSCAN 節點
 內容 [383](#)
hdbscannode 內容 [383](#)
hdbscannugget 內容 [314](#)
histogramnode 內容 [178](#)
historynode 內容 [147](#)

I

IBM Cognos 來源節點
 內容 [79](#)
IBM Cognos TM1 來源節點
 內容 [97](#), [98](#)
IBM SPSS Analytic Server 儲存庫
 指令行引數 [61](#)
IBM SPSS Collaboration and Deployment Services
 Repository
 指令行引數 [61](#)
 Scripting [45](#)
IBM SPSS Modeler
 從指令行執行 [57](#)
IBM SPSS Statistics 來源節點
 內容 [379](#)
IBM SPSS Statistics 模型
 節點 Scripting 內容 [380](#)
IBM SPSS Statistics 轉換節點
 內容 [379](#)
IBM SPSS 「統計量輸出」節點
 內容 [380](#)
ID [17](#)
Isotonic-AS 節點
 內容 [397](#)
isotonicasnode 內容 [397](#)

J

JSON 內容模型 [52](#)
JSON 來源節點
 內容 [94](#)
jsonimportnode 內容 [94](#)
Jython [13](#)

K

K-Means 模型
 節點 Scripting 內容 [240](#), [306](#)

K-Means-AS 模型
節點 Scripting 內容 [241](#), [397](#)
KDE 建模節點
內容 [385](#)
KDE 模型
節點 Scripting 內容 [314](#)
KDE 模擬節點
內容 [345](#), [386](#)
kdeapply 內容 [314](#)
kdeexport 內容 [345](#), [386](#)
kdemodel 內容 [385](#)
kmeansasnode 內容 [241](#), [397](#)
kmeansnode 內容 [240](#)
KNN 模型
節點 Scripting 內容 [306](#)
knnnode 內容 [242](#)
kohonen 模型
節點 Scripting 內容 [244](#)
Kohonen 模型
節點 Scripting 內容 [307](#)
kohonenode 內容 [244](#)

L

logregnode 內容 [248](#)
lowertoupper 函數 [45](#)
LSVM 模型
節點 Scripting 內容 [253](#)
lsvmnode 內容 [253](#)

M

mapvisualization 內容 [179](#)
matrixnode 內容 [346](#)
meansnode 內容 [348](#)
mergenode 內容 [116](#)
Microsoft 模型
節點 Scripting 內容 [315](#), [317](#)
MS 決策樹狀結構
節點 Scripting 內容 [315](#), [317](#)
MS 時間序列
節點 Scripting 內容 [317](#)
MS 神經網路
節點 Scripting 內容 [315](#), [317](#)
MS 順序叢集
節點 Scripting 內容 [317](#)
MS 線性迴歸
節點 Scripting 內容 [315](#), [317](#)
MS 邏輯迴歸
節點 Scripting 內容 [315](#), [317](#)
msassocnode 內容 [315](#)
msbayesnode 內容 [315](#)
msclusternode 內容 [315](#)
mslogisticnode 內容 [315](#)
msneuralnetworknode 內容 [315](#)
msregressionnode 內容 [315](#)
mssequenceclusternode 內容 [315](#)
mstimeseriesnode 內容 [315](#)
mstreenode 內容 [315](#)
MultiLayerPerceptron-AS 節點
內容 [398](#)
multilayerperceptronnode 內容 [398](#)
multiplotnode 內容 [182](#)

N

Netezza 分割叢集作業模型
節點 Scripting 內容 [326](#), [340](#)
Netezza 決策樹狀結構模型
節點 Scripting 內容 [326](#), [340](#)
Netezza 貝式邏輯分類演算法模型
節點 Scripting 內容 [326](#), [340](#)
Netezza 時間序列模型
節點 Scripting 內容 [326](#)
Netezza 迴歸樹狀結構模型
節點 Scripting 內容 [326](#), [340](#)
Netezza 廣義線性模型
節點 Scripting 內容 [326](#)
Netezza 模型
節點 Scripting 內容 [326](#)
Netezza 線性迴歸模型
節點 Scripting 內容 [326](#), [340](#)
Netezza Bayes Net 模型
節點 Scripting 內容 [326](#), [340](#)
Netezza K-Means 模型
節點 Scripting 內容 [326](#), [340](#)
Netezza KNN 模型
節點 Scripting 內容 [326](#), [340](#)
Netezza PCA 模型
節點 Scripting 內容 [326](#), [340](#)
netezabayesnode 內容 [326](#)
netezadectreenode 內容 [326](#)
netezadivclusternode 內容 [326](#)
netezaglmnode 內容 [326](#)
netezakmeansnode 內容 [326](#)
netezaknnnode 內容 [326](#)
netezalineregressionnode 內容 [326](#)
netezanaivebayesnode 內容 [326](#)
netezapcanode 內容 [326](#)
netezaregtreenode 內容 [326](#)
netezatimeseriesnode 內容 [326](#)
neuralnetnode 內容 [254](#)
neuralnetworknode 內容 [257](#)
numericpredictornode 內容 [201](#)

O

ocsvmnode 內容 [387](#)
oraabnode 內容 [319](#)
oraainode 內容 [319](#)
oraapriorinode 內容 [319](#)
Oracle 支援向量機器模型
節點 Scripting 內容 [319](#), [326](#)
Oracle 決策樹狀結構模型
節點 Scripting 內容 [319](#), [326](#)
Oracle 廣義線性模型
節點 Scripting 內容 [319](#)
Oracle 模型
節點 Scripting 內容 [319](#)
Oracle Adaptive Bayes 模型
節點 Scripting 內容 [319](#), [326](#)
Oracle AI 模型
節點 Scripting 內容 [319](#)
Oracle Apriori 模型
節點 Scripting 內容 [319](#), [326](#)
Oracle KMeans 模型
節點 Scripting 內容 [319](#), [326](#)
Oracle MDL 模型

Oracle MDL 模型 (繼續)
節點 Scripting 內容 [319, 326](#)
Oracle Naive Bayes 模型
節點 Scripting 內容 [319, 326](#)
Oracle NMF 模型
節點 Scripting 內容 [319, 326](#)
Oracle O-叢集
節點 Scripting 內容 [319, 326](#)
oradecisiontreenode 內容 [319](#)
oraglmnode 內容 [319](#)
orakmeansnode 內容 [319](#)
oramdlnode 內容 [319](#)
oranbnode 內容 [319](#)
oranmfnode 內容 [319](#)
oraoclusternode 內容 [319](#)
orasvmnode 內容 [319](#)
outputfilenode 內容 [371](#)

P

partitionnode 內容 [148](#)
PCA 模型
節點 Scripting 內容 [220, 304](#)
PCA/因素模型
節點 Scripting 內容 [220, 304](#)
plotnode 內容 [183](#)
Python
Scripting [13](#)
Python 模型
節點 Scripting 內容 [309, 313, 314](#)
Gaussian Mixture 節點 Scripting 內容 [306](#)

Q

QUEST 模型
節點 Scripting 內容 [259, 309](#)
questnode 內容 [259](#)

R

R 建置節點
節點 Scripting 內容 [204](#)
R 輸出節點
內容 [351](#)
R 轉換節點
內容 [119](#)
randomtrees 內容 [261](#)
reclassifynode 內容 [149](#)
regressionnode 內容 [263](#)
reordernode 內容 [149](#)
reportnode 內容 [350](#)
reprojectnode 內容 [150](#)
restructurenode 內容 [151](#)
retrieve 指令 [45](#)
RFM 分析節點
內容 [151](#)
RFM 聚集節點
內容 [118](#)
rfmaggregatenode 內容 [118](#)
rfmanalysisnode 內容 [151](#)
rfnode 內容 [389](#)
routputnode 內容 [351](#)
Rprocessnode 內容 [119](#)

S

samplenode 內容 [120](#)
SAS 來源節點
內容 [94](#)
SAS 匯出節點
內容 [372](#)
sasexportnode 內容 [372](#)
sasimportnode 內容 [94](#)
Script
迴圈 [5, 6](#)
從文字檔匯入 [1](#)
條件式執行 [5, 8](#)
選取欄位 [8](#)
儲存 [1](#)
疊代鍵 [7](#)
疊代變數 [7](#)
Scripting
中斷 [9](#)
功能選擇模型 [4](#)
在 SuperNode 中 [4](#)
串流 [1, 25](#)
串流執行順序 [45](#)
使用者介面 [1, 3, 4](#)
所用縮寫 [64](#)
執行 [9](#)
從指令行 [48](#)
條件式執行 [5, 8](#)
視覺化迴圈 [5, 6](#)
通用內容 [65](#)
概觀 [1, 13](#)
圖形節點 [169](#)
與早期版本的相容性 [48](#)
語法 [13-18, 20-22](#)
獨立式 Script [1, 25](#)
輸出節點 [341](#)
選取欄位 [8](#)
錯誤檢查 [48](#)
環境定義 [26](#)
舊式 Scripting [407-409, 412, 415, 416](#)
疊代鍵 [7](#)
疊代變數 [7](#)
diagrams [25](#)
Python Scripting [407-409, 412, 415, 416](#)
SuperNode 串流 [25](#)
SuperNode Script [1, 25](#)
Scripting API
多個串流 [44](#)
存取產生的物件 [38](#)
串流參數 [40](#)
取得目錄 [35](#)
處理錯誤 [39](#)
階段作業參數 [40](#)
搜尋 [35](#)
廣域值 [43](#)
範例 [35](#)
獨立式 Script [44](#)
簡介 [35](#)
meta 資料 [36](#)
SuperNode 參數 [40](#)
selectnode 內容 [122](#)
sequencenode 內容 [266](#)
setglobalsnode 內容 [351](#)
settoflagnode 內容 [152](#)

simevalnode 內容 [352](#)
simfitnode 內容 [353](#)
simgennode 內容 [95](#)
SLRM 模型
 節點 Scripting 內容 [267, 310](#)
slrmnode 內容 [267](#)
SMOTE 節點
 內容 [391](#)
smotenode 內容 [391](#)
sortnode 內容 [122](#)
Space-Time-Boxes 節點
 內容 [111, 123](#)
Space-Time-Boxes 節點內容 [111](#)
spacetimeboxes 內容 [123](#)
statisticsexportnode 內容 [381](#)
statisticsimportnode 內容 [4, 379](#)
statisticsmodelnode 內容 [380](#)
statisticsnode 內容 [354](#)
statisticsoutputnode 內容 [380](#)
statisticstransformnode 內容 [379](#)
store 指令 [45](#)
STP 節點
 內容 [268](#)
STP 節點塊
 內容 [311](#)
stpnode 內容 [268](#)
stream.nodes 內容 [45](#)
streamingtimeseries 內容 [124](#)
streamingts 內容 [130](#)
SuperNode
 內容 [401](#)
 串流 [25](#)
 參數 [401](#)
 設定內容 [401](#)
 Script [1, 4, 5, 25](#)
 Scripting [401](#)
SVM 模型
 節點 Scripting 內容 [274](#)
svmnode 內容 [274](#)

T

t-SNE 節點
 內容 [187, 391](#)
tablnode 內容 [355](#)
tcm 模型
 節點 Scripting 內容 [311](#)
tcmnode 內容 [275](#)
timeintervalsnode 內容 [153](#)
timeplotnode 內容 [185](#)
timeseriesnode 內容 [288](#)
tm1import 節點內容 [98](#)
tm1odataimport 節點內容 [97](#)
transformnode 內容 [358](#)
transposenode 內容 [159](#)
Tree-AS 模型
 節點 Scripting 內容 [290, 312](#)
treeas 內容 [290](#)
ts 內容 [280](#)
tsnode 內容 [187, 391](#)
twcimport 節點內容 [99](#)
TwoStep 模型
 節點 Scripting 內容 [292, 313](#)
TwoStep AS 模型

TwoStep AS 模型 (繼續)
 節點 Scripting 內容 [293, 313](#)
twostepAS 內容 [293](#)
twostepnode 內容 [292](#)
typenode 內容 [4, 161](#)

U

userinputnode 內容 [100](#)

V

variablefilenode 內容 [101](#)

W

Web 節點
 內容 [189](#)
webnode 內容 [189](#)

X

XGBoost 線性節點
 內容 [393](#)
XGBoost 樹狀結構節點
 內容 [394](#)
XGBoost-AS 節點
 內容 [398](#)
xgboostasnode 內容 [398](#)
xgboostlinearnode 內容 [393](#)
xgboosttreenode 內容 [394](#)
XML 內容模型 [51](#)
XML 來源節點
 內容 [106](#)
XML 匯出節點
 內容 [376](#)
xmlexportnode 內容 [376](#)
xmlimportnode 內容 [106](#)

