

JMP® 協同分析：JMP® 產品系列與 Python、R 的結合使用

白皮書

目錄

簡介	1
JMP和Python	1
利用 JMP Pro 模型生成 Python 代碼.....	1
有關連接 JMP 和 Python 的基本知識.....	3
JMP的一項額外小功能—在圖形中添加圖像標記.....	5
JMP 與 Python 協同工作的其他簡單示例.....	6
JMP和R	7
有關連接 JMP 和 R 的基本知識	7
JMP 與 R 協同工作的其他簡單示例.....	11
高級示例	11
使用 t-SNE 和 UMAP外掛程式進行資料視覺化:	12
SVM外掛程式:	14
JMP到R的外掛程式生成器:	
在沒有JSL的情況下使用 JMP 到 R 的連接.....	15
結論	15

簡介

JMP 是一款功能強大, 適用於 Windows 和 Mac 桌面的資料視覺化和統計分析軟體, JMP 獨特、動態的互動式分析, 讓資料探索充滿吸引力並能夠提供深刻的見解, 同時 JMP 還包含許多高級分析選項, 滿足了資料探索人員和分析師的需求。不過, 在某些情況下, 您可能仍然希望 (或需要) 將 JMP 與開源工具 (如 Python 或 R) 結合起來使用。

可能的情形例如:

- 公司統計人員禁止使用未經驗證的軟體 (例如 R) 進行基礎分析, 但他們可能希望結合使用R套裝軟體與 JMP 中經驗證的方法, 探索出新方法。
- Python 或 R 程式師希望充分利用 JMP 中提供的動態連結、高級視覺化或強大的分析功能。
- 公司的資料分析部門可能希望開發一個 JMP 外掛程式, 該外掛程式允許將某些R 或 Python 代碼作為 JMP 中的外掛程式運行, 以便未來的用戶可以與 GUI 外掛程式交互而無需自己編寫 R 或 Python 代碼。
- 統計學老師可能希望在課堂中引入特定的開源程式包, 學生只需點擊即可使用, 而無需自己編碼。

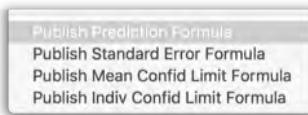
無論您出於何種原因需要將開源 (或其他) 工具與 JMP 軟體 GUI 相連接, 本指南都將幫助您如何在 JMP 中連接 Python 和 R。¹

JMP® 和 Python

利用 JMP Pro 模型生成 Python 代碼

您可以使用 JMP Pro, 為在 JMP 中創建的預測模型生成 Python 代碼。當您希望在 JMP Pro 中構建、探索和比較模型, 但要將其部署於不同的生產環境中時, 此功能將派上用場。JMP Pro 提供了一種簡便的方法, 可以構建多個模型並比較競爭模型, 甚至可以在“公式存儲庫”平臺中創建模型平均值 (需組合)。

首先, 使用 JMP “分析”功能表下的工具擬合模型。要將模型保存到“公式存儲庫”, 可以使用“發佈機率公式”(用於類別模型)或“發佈預測公式”(用於預測模型)選項。通常, 這兩個選項都可以在模型結果的紅色三角形選項中或“保存行”下找到。



¹除了 R 和 Python 之外, JMP 還可以連接到 SAS 和 Matlab; 而且除 Python 外, JMP 還會生成 C、JavaScript、SAS 和 SQL 模型評分代碼。如需瞭解有關上述 JMP 連接、入門技巧和示例的更多資訊, 請訪問 jmp.com/support/help/en/15.1/#page/jmp/extending-jmp.shtml。

一旦選擇了關鍵字“發佈”（或使用“從行添加公式”選項），您的模型就會發送到“公式存儲庫”。 “公式存儲庫”是一個平臺，通過該平臺您可以保存一個或多個競爭模型，比較競爭模型，創建組合模型並將其部署於各種生產環境中。

“生成Python代碼”是每個單獨模型（例如用於預測重量的“擬合最小平方法”模型）紅色三角形下的一個選項，或者是“公式存儲庫”紅色三角形下的一個選項，該選項可用於為“公式存儲庫”視窗中保存的全部模型生成代碼。

生成的 Python 代碼（如下所示）包括一些必需的依賴項，這些依賴項會在代碼的第一部分中予以調用。其次是版權資訊，然後是一些用來創建變數名和類型的Python 代碼，最後才是公式。該公式將用於預測新觀測記錄中預測變數已知時的響應變數（在本示例中，預測變數為年齡、身高和性別）。

```
Fit_Least_Squares_weight
from __future__ import division
import jmp_score as jmp
from math import *
import numpy as np

"""
Copyright(C) 2018 SAS Institute Inc. All rights reserved.

Notice:
The following permissions are granted provided that the
above copyright and this notice appear in the score code and
any related documentation. Permission to copy, modify
and distribute the score code generated using
JMP(R) software is limited to customers of SAS Institute Inc. ("SAS")
and successive third parties, all without any warranty, express or
implied, or any other obligation by SAS. SAS and all other SAS
Institute Inc. product and service names are registered
trademarks or trademarks of SAS Institute Inc. in the USA
and other countries. Except as contained in this notice,
the name of the SAS Institute Inc. and JMP shall not be used in
the advertising or promotion of products or services without
prior written authorization from SAS Institute Inc.
"""

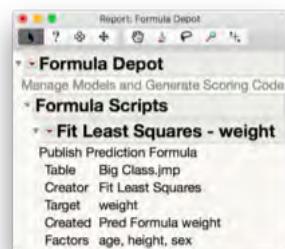
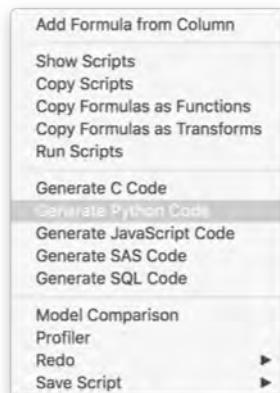
Python code generated by JMP v14.1.0

def getModelMetadata():
    return {"creator": "Fit Least Squares", "modelName": "weight",
            "table": "Big Class.jmp", "version": "14.1.0", "timestamp": "2018-08-08T04:02:07Z"}

def getInputMetadata():
    return {
        "age": "float",
        "height": "float",
        "sex": "str"
    }

def getOutputMetadata():
    return {
        "Pred Formula weight": "float"
    }

def score(indata, outdata):
    _temp_0 = np.nan
    _temp_1 = np.nan
    if (indata[u"sex"] == u"F"):
        _temp_0 = 2.0492068900361
    elif (indata[u"sex"] == u"M"):
        _temp_0 = -2.0492068900361
    else:
        _temp_0 = np.nan
    if (jmp.numeq(indata[u"age"], 12)):
        _temp_1 = 0
    elif ((jmp.numeq(indata[u"age"], 13))):
        _temp_1 = 13.6855931672148
    elif ((jmp.numeq(indata[u"age"], 14))):
        _temp_1 = -25.8472610743993
    elif ((jmp.numeq(indata[u"age"], 15))):
        _temp_1 = -19.7698493666985
    elif ((jmp.numeq(indata[u"age"], 16))):
        _temp_1 = -10.1590212259529
    elif ((jmp.numeq(indata[u"age"], 17))):
        _temp_1 = 2.52025614257322
    else:
        _temp_1 = np.nan
    outdata[u"Pred Formula weight"] = -176.033203126788 + 4.72294023921341 * indata[u"height"]
    + _temp_0 + _temp_1
    return outdata[u"Pred Formula weight"]
```



您可以造訪 jmp.com/support/help/en/15.1/#page/jmp/formula-depot.shtml，瞭解有關該主題的更多資訊。

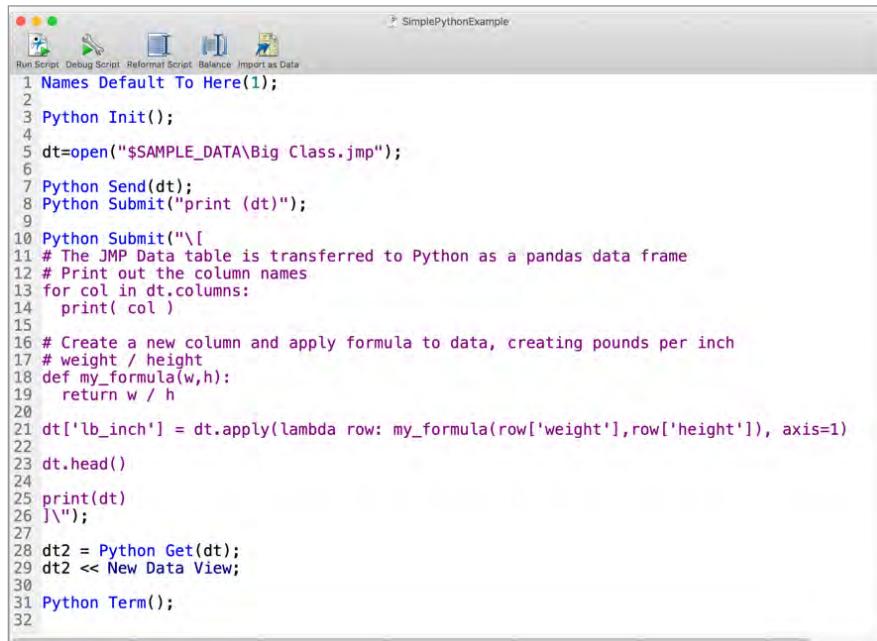
有關連接 JMP 和 Python 的基本知識

除了從 JMP 內部生成 Python 模型評分代碼外，您還可以打開 JMP 和 Python 之間的連接，並在 Python 和 JMP 之間發送和接收資料、結果及操作（通過 JMP 和 Python 代碼組合）。（JMP 和 R 之間的連接也採用相同的方式。）為了在 JMP 中使用 Python 連接，您將使用 JMP 指令碼語言 (JSL) 封裝 Python 代碼。

下面舉一個簡單的例子來進行說明。您需要在電腦上安裝 Python²，不過這是大多數新電腦的標準配置。

- (1) 從 JMP 中打開 Python 連接。
- (2) 執行一項操作，比如將資料從 JMP 發送到 Python。
- (3) 執行另一項操作，例如直接向 Python 提交 Python 代碼以創建新變數。
- (4) 再執行另一項操作，例如將結果或資料從 Python 返回到 JMP（可能在 JMP 中將結果視覺化）。
- (5) 關閉 Python 連接。

下面是該簡單示例的代碼。為了學習如何操作，您首先需要打開 JMP。接著，轉至“檔 > 新建 > 新建腳本”，打開一個新的空白腳本視窗。然後輸入以下內容：



```

1 Names Default To Here();
2
3 Python Init();
4
5 dt=open("$SAMPLE_DATA\Big Class.jmp");
6
7 Python Send(dt);
8 Python Submit("print (dt)");
9
10 Python Submit("[
11 # The JMP Data table is transferred to Python as a pandas data frame
12 # Print out the column names
13 for col in dt.columns:
14     print( col )
15
16 # Create a new column and apply formula to data, creating pounds per inch
17 # weight / height
18 def my_formula(w,h):
19     return w / h
20
21 dt['lb_inch'] = dt.apply(lambda row: my_formula(row['weight'],row['height']), axis=1)
22
23 dt.head()
24
25 print(dt)
26 ]");
27
28 dt2 = Python Get(dt);
29 dt2 << New Data View;
30
31 Python Term();
32

```

上面的腳本中有 9 段 JSL 代碼（以藍色表示）。第 1 列有助於確保我們創建的任何檔或變數引用都與此腳本相關。腳本第 3 列 "Python Init();"，打開了 Python。³

² python.org。

³ 有關 Python 安裝和版本管理的詳細資訊，可造訪：jmp.com/support/help/en/15.1/#page/jmp/install-python.shtml#，jmp.com/support/help/en/15.1/#page/jmp/troubleshooting-the-jmp-python-integration.shtml#ww822804。

第 5 列，“`dt=Open("$SAMPLE_DATA/Big Class.jmp");`” 要求 JMP 打開 JMP 樣本資料目錄中的一個資料集，並為其指定名稱 “`dt`”⁴。本示例中的資料集名稱為 “`Big Class.jmp`”。

第 7 列，“`Python Send(dt);`” 將 “`dt`” 資料表從 JMP 發送至 Python。JMP 資料表將作為 Pandas 資料框傳輸至 Python。第 8 列 “`Python Submit("print(dt)");`”，向 Python 提交某個 Python 代碼。引號內的信息是 Python 腳本；該列的其餘部分是 JSL 代碼包裝。如果需要，可以使用 JMP 日誌視窗監控您的代碼提交並進行故障排除。轉至 JMP 功能表列，然後選擇 “視窗→日誌” 打開 JMP 日誌。在這裡，您可以看到 Python 內部的所有錯誤消息或結果。例如，如果我們在運行此代碼後檢查日誌，則將在日誌中看到以下 Python 輸出：

```
dt=open("$SAMPLE_DATA\Big Class.jmp");

Data Table( "Big Class.jmp" )

Python Send(dt);
Python Submit("print (dt)");

      name  age  sex  height  weight
0      KATIE  12    F      59      95
1      LOUISE  12    F      61     123
2      JANE    12    F      55      74
3      JACLYN  12    F      66     145
4      LILLIE  12    F      52      64
```

第 10-26 列包含一個較長的 “`Python Submit()`” 語句。這段嵌入在 JSL 代碼 “`Python Submit()`” 語句中的 Python 代碼，將列印（日誌中的）行名稱，然後在最終列印新資料表之前，創建一個由公式 “`lb_inch=weight/height`” 構建的新行，新資料表將在日誌中把這一新行添加至來源資料表中：

```
name
age
sex
height
weight
      name  age  sex  height  weight  lb_inch
0      KATIE  12    F      59      95  1.610169
1      LOUISE  12    F      61     123  2.016393
2      JANE    12    F      55      74  1.345455
3      JACLYN  12    F      66     145  2.196970
4      LILLIE  12    F      52      64  1.230769
5      TIM     12    M      60      84  1.400000
6      JAMES   12    M      61     128  2.098361
```

第 28 列和第 29 列將更新後的資料表返回軟體的工作記憶體中，並將其作為新的 JMP 資料表予以打開。

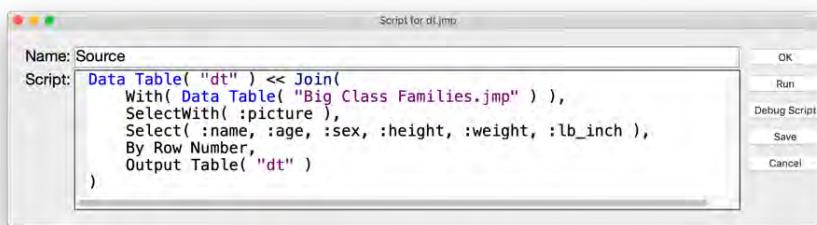
最後一列代碼 “`Python Term()`” 終止與 Python 的連接。

⁴ 注意，在 R 實例中，我們使用了略微不同的 JSL 代碼來指向一個打開的資料檔案。這是指向資料的另一種方法，在此例中，可使用 JSL 打開資料檔案。

JMP 的一項額外小功能 – 在圖形中添加圖像標記

雖然該簡單示例的最後部分沒有進一步使用 JMP 到 Python 的連接，但它確實說明了 JMP 與 Python（或 R 等其他已連接軟體）的結合使用可能帶來的好處。具體來說，我們可以在 JMP 和 Python 之間交互工作，運行規定的 Python 代碼或特定的 Python 包，然後使用 JMP 的動態功能改進最終的分析或視覺化效果。

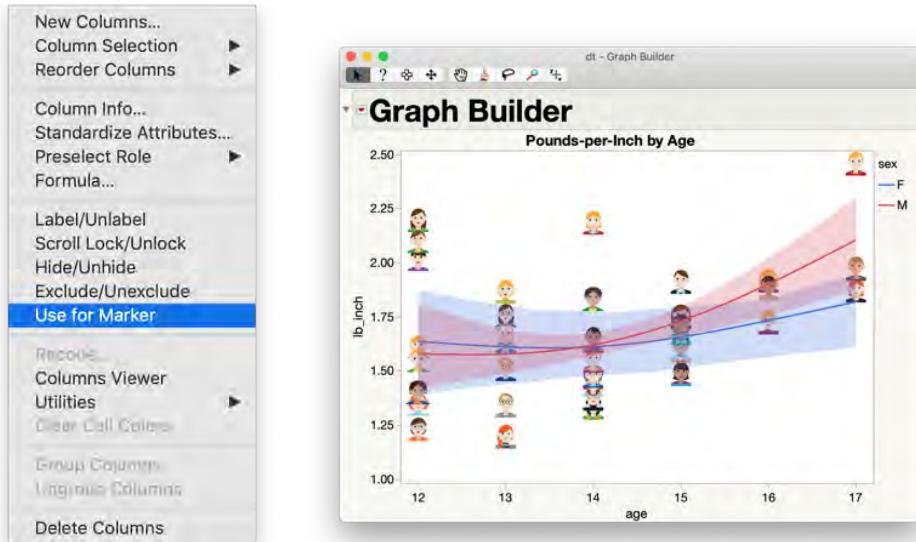
例如，將 “dt” 資料表返回 JMP 之後，我們可以利用 JMP 頗具特色的視覺化功能來改進我們的圖形，方法很簡單，只需點擊幾下滑鼠或通過簡短的 JSL 腳本即可。為了說明這一點，我們將 “dt” 表格和 Big Class Families 樣本資料集（可從 JMP 的樣本資料庫中獲得）連接起來，將一行圖像添加至新 “dt” 中。首先，打開 Big Class Families（通過點擊 “幫助 > 樣本資料庫 > Big Class Families” 或運行代碼 “dt = Open(“\$SAMPLE_DATA/Big Class.jmp”);”），使用 “表 > 連接”（或以下腳本）將圖像列連接到我們的 “dt” 表。



這些步驟將一個名為 “圖片” 的行添加到我們的資料集中，並將來自 “Big Class Families” 資料表的學生圖像導入到我們新的合併資料表中。

	picture	name	age	sex	height	weight	lb_inch
1		ROBERT	17	M	70	172	2.46
2		ALFRED		F			
3		ALICE					
4		AMY					
5		BARBARA					
6		34 others					
7		KATIE	12	F	51	95	1.6101...
8		LOUISE	12	F	59	123	2.0163...
9		JANE	12	F	55	74	1.3454...
10		JACLYN	12	F	66	145	2.1969...
11		LILLIE	12	F	52	64	1.2307...
12		TIM	12	M	60	84	1.4
13		JAMES	12	M	61	128	2.0983...
14		ROBERT	12	M	51	79	1.5490...
15		BARBARA	13	F	60	112	1.8666...

現在，我們可以使用圖像作為圖中的標記，構建新的“每英寸磅數”這一變數的視覺化。首先，點擊“圖片”行並轉至“行 > 用於標記”，將圖像行設為標記。



在這個簡單的示例中，我們使用了從 JMP 到 Python 的連接來打開 JMP 中的資料，將其發送到 Python，在 Python 中創建新行，並將新行添加到 Python pandas 資料框中，然後返回更新後的資料，形成一個 JMP 資料表，我們可以在 JMP 中對返回後的資料表進行分析和進一步操作。

JMP 與 Python 協同工作的其他簡單示例

您可以在“JMP幫助”中找到更多關於JMP與Python交互作用的簡單案例，只需前往“JMP腳本 → 擴展JMP → 與Python協同工作”或造訪 jmp.com/support/help/en/15.1/#page/jmp/additional-python-integration-examples.shtml 即可。

您可以在此找到以下代碼：

- 向Python發送資料表。
- 在Python中創建對象。
- Python中的矩陣計算。

JMP® 和 R

通過連接JMP和開源軟體R⁵，您可以輕鬆地利用R功能及JMP軟體的動態和互動式方法進行分析和視覺化。您還可以為R套件創建易於使用的功能表介面，並結合JMP和R的元素。當您在電腦上同時安裝了JMP和R時，即可實現上述功能。“腳本索引”（“幫助 > 腳本索引 > R 函數”）中列舉並說明了執行這些連接的 R 函數（包含示例）。

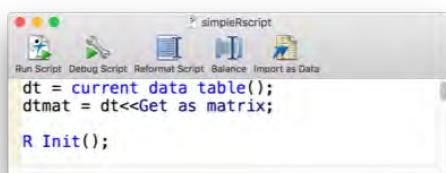
有關連接 JMP 和 R 的基本知識

為了在 JMP 中使用 R 連接，您需要使用 JMP 指令碼語言 (JSL) 來封裝您的 R 代碼。

例如，我們將一些資料從 JMP 移至 R 進行分析和操作，然後再將其返回 JMP。這個過程非常簡單。我們將編寫一些 JSL 代碼來打開 R，將資料從 JMP 發送到 R，再將 R 代碼發送到 R，最後將資料從 R 返回 JMP。

首先，確保在同一台電腦上安裝了 R 和 JMP。⁶

接下來，通過“檔 > 新建 > 新建腳本”打開 JMP 中的一個腳本視窗。

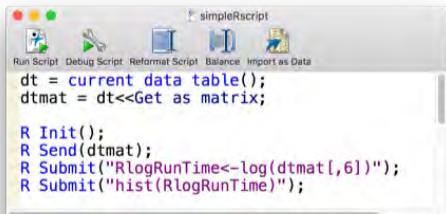


我們還要在 JMP 中打開一個樣本資料集。轉至“幫助 > 樣本資料庫”，並打開 Fitness.jmp 資料表。為了將其發送給 R，我們需要使用兩列簡單的 JSL。第一列從本質上確認了打開的資料表 Fitness.jmp，並將其重命名為“dt”。第二列將新命名的“dt”資料表轉換為矩陣格式。這是因為 R 的資料結構選項與 JMP 略有不同，而矩陣格式是一個簡單的框架，可以在 JMP 和 R 中輕鬆使用。接下來，我們從 JMP 中調用 R。如果 R 安裝在同一台電腦上，JMP 就可以找到它；您無需設置連接或將 JMP 指向 R 安裝。您只需輸入 JSL "R Init();"。JMP 將在該電腦上搜索 R，然後對其進行初始化（或將其打開）。

⁵ cran.r-project.org。

⁶ 有關 R 安裝和版本管理的詳細資訊，可造訪：jmp.com/support/help/en/15.1/#page/jmp/installing-r.shtml。

現在我們可以在 JMP 和 R 之間來回發送資訊了。下面的第一列代碼將資料的矩陣版本 dmat 發送給 R。第二列代碼則將 R 代碼發送至 R 中執行。



```

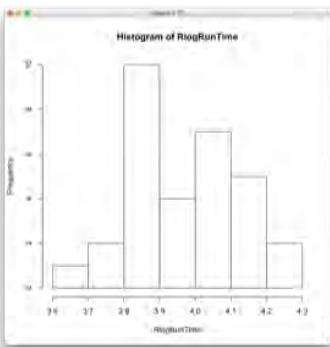
Run Script Debug Script Reformat Script Balance Import as Data
simpleRscript
dt = current data table();
dmat = dt<<Get as matrix;

R Init();
R Send(dmat);
R Submit("RlogRunTime<-log(dmat[,6])");
R Submit("hist(RlogRunTime)");

```

列 `RlogRunTime<-log(dmat[,6]);` 獲取輸入資料矩陣的第六行（即“Mile RunTime”，來自資料集Fitness），並對其進行對數變換。這些結果現名為 “RlogRunTime”（因為這是我們在 R 代碼片段中為其指定的名稱）。

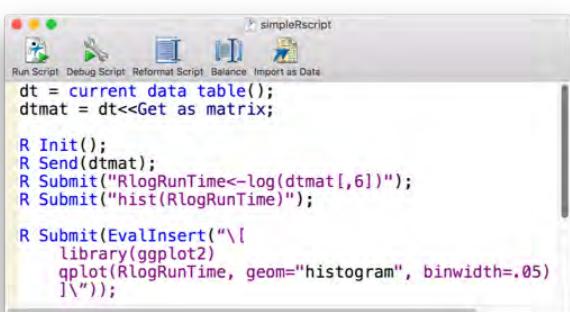
列 `R Submit("hist(RlogRunTime)");` 在 R 中生成一個長條圖，一旦運行此列代碼後，即會在桌面上彈出該長條圖。



要運行代碼，請選擇一列行或多列行（每行必須以 ";" 結束），然後點擊腳本視窗頂部的“運行腳本”按鈕：



如果我們要通過JSL包發送多列R代碼，可以使用“`R Submit(EvalInsert("\\[____]\\\"))`”代替簡單的“`R Submit("____")`”，正如您將在以下R代碼中觀察到的一樣，我們會在 R 中使用“`ggplot2`”套件來創建長條圖：



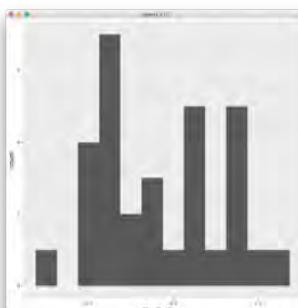
```

Run Script Debug Script Reformat Script Balance Import as Data
simpleRscript
dt = current data table();
dmat = dt<<Get as matrix;

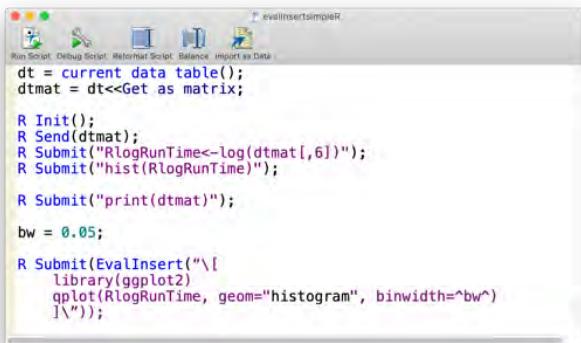
R Init();
R Send(dmat);
R Submit("RlogRunTime<-log(dmat[,6])");
R Submit("hist(RlogRunTime)");

R Submit(EvalInsert("\\[
  library(ggplot2)
  qplot(RlogRunTime, geom='histogram', binwidth=.05)
]\""));

```



通常，當您需要在 JSL 的字串中進行替換或計算運算式時，“**EvalInsert()**”也會發揮重要作用。例如，我們可以在 JSL 腳本中將 R 長條圖的箱寬指定為變數，然後使用 “**EvalInsert()**” 代碼來調用該變數，如下所示：



```
Run Script Debug Script Reformat Script Balance Import as Data
evalInsertsimpleR
dt = current data table();
dmat = dt<<Get as matrix;

R Init();
R Send(dmat);
R Submit("RlogRunTime<-log(dmat[,6])");
R Submit("hist(RlogRunTime)");

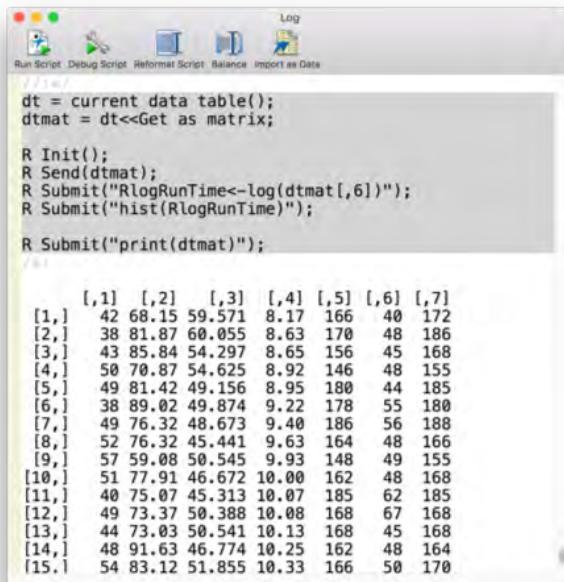
R Submit("print(dmat)");

bw = 0.05;

R Submit(EvalInsert("\
library(ggplot2)
plot(RlogRunTime, geom="histogram", binwidth=^bw^")
\"));


```

如果需要，可使用 JMP 日誌視窗監控代碼提交和故障排除。轉至 JMP 功能表列，然後選擇“窗口 → 日誌”打開 JMP 日誌。在這裡，您可以看到 R 內部的所有錯誤消息或結果。



```
Log
Run Script Debug Script Reformat Script Balance Import as Data
evalInsertsimpleR
dt = current data table();
dmat = dt<<Get as matrix;

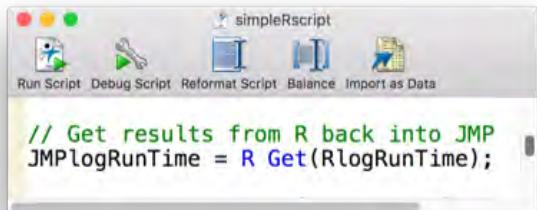
R Init();
R Send(dmat);
R Submit("RlogRunTime<-log(dmat[,6])");
R Submit("hist(RlogRunTime)");

R Submit("print(dmat)");

[,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]
[1,] 42 68.15 59.571 8.17 166 40 172
[2,] 38 81.87 60.055 8.63 170 48 186
[3,] 43 85.84 54.297 8.65 156 45 168
[4,] 58 70.87 54.625 8.92 146 48 155
[5,] 49 81.42 49.156 8.95 180 44 185
[6,] 38 89.02 49.874 9.22 178 55 180
[7,] 49 76.32 48.673 9.40 186 56 188
[8,] 52 76.32 45.441 9.63 164 48 166
[9,] 57 59.08 50.545 9.93 148 49 155
[10,] 51 77.91 46.672 10.00 162 48 168
[11,] 40 75.07 45.313 10.07 185 62 185
[12,] 49 73.37 50.388 10.08 168 67 168
[13,] 44 73.03 50.541 10.13 168 45 168
[14,] 48 91.63 46.774 10.25 162 48 164
[15,] 54 83.12 51.855 10.33 166 50 170
```

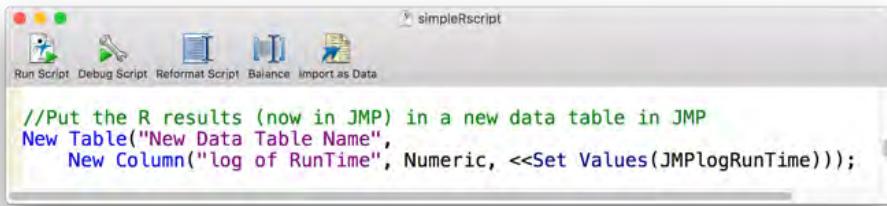
當然，我們也可以在 JMP 中使用分佈平臺創建此長條圖。我們可以通過兩種方式選擇使用 JMP 功能，如下所示。

首先，我們需要將 “**RlogRunTime**” 從 R 返回至 JMP。這部分代碼採用的是新轉換的資料，即 R 中的 “**RlogRunTime**”，然後以新名稱 “**JMPlogRunTime**” 將其返回 JMP。



```
// Get results from R back into JMP
JMPlogRunTime = R Get(RlogRunTime);
```

我們還需要將此新的 JMP 資料從內部記憶體轉移至 JMP 中的新資料表，以便我們可以將其保存下來並進行下一步操作。下面的代碼創建了一個新的 JMP 資料表，其中一行由 “**JMPlogRunTime**” 資料構成。

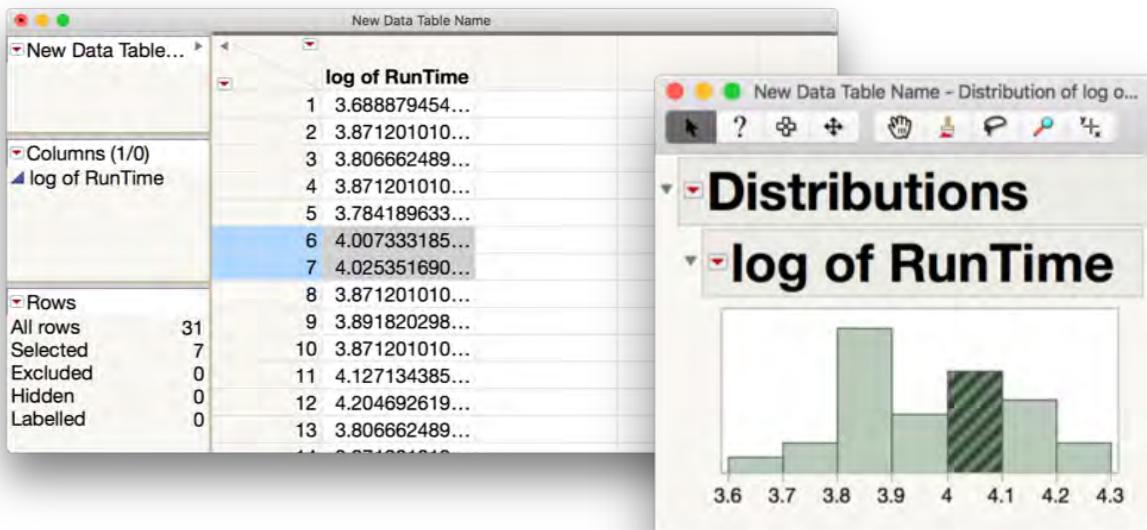


```
//Put the R results (now in JMP) in a new data table in JMP
New Table("New Data Table Name",
  New Column("log of RunTime", Numeric, <<Set Values(JMPlogRunTime)));
```

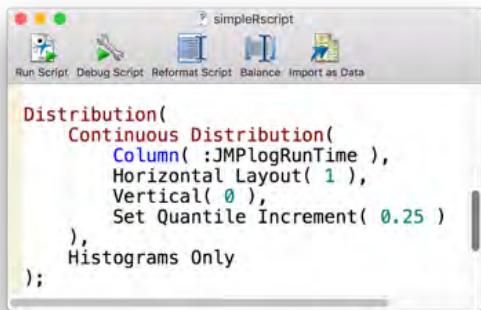
然後，我們可以選擇以這兩種方式中的任意一種，進一步使用 JMP 功能，如下所示。

(1) 通過點擊功能表使用 JMP 功能，在新的 JMP 資料表上進行操作。

在示例中，我們可以使用 “分析 > 分佈” 選項在 JMP 中創建互動式長條圖。



(2) 使用 JSL 在腳本視窗中繼續操作，以調用那些與點擊操作效果相同的 JMP 函數。如果您熟悉 JSL，則可以直接在腳本視窗中編寫更多 JMP 操作。如果您不熟悉與點擊操作對應的 JSL，則可以通過一次點擊創建操作，進而輕鬆地生成該 JSL。接著從結果視窗中點擊紅色三角形，然後選擇“將腳本複製到腳本視窗”。要為新建資料表更新腳本，請編輯 JSL 腳本，以便可將其應用於合適的變數。



JMP 與 R 協同工作的其他簡單示例

在“JMP幫助”功能表和“腳本指南→拓展 JMP → 與R協同工作”項下搜索“R”，就可以找到關於 JMP 和 R 如何協同工作的更多簡單示例。

在“JMP幫助”中，您可以找到以下代碼：

- 向R發送數據表。
- 在R中創建對象。
- 使用R函數和圖形。
- R中的簡單矩陣加法。
- 在R中獲取Bootstrap置信區間。

高級示例

一旦您理解了將 JMP 介面與 R 和 Python 結合使用的基本知識，就可以採用有趣的互動式方法對其進行擴展。

以下是一些示例的連結：

JMP 到 GGPlot 的互動式調用程式

此腳本通過JMP與R的連接，使用R套件 `ggplot2`⁸ 來創建平滑曲線的密度圖⁷（按分組變數的水準進行劃分）。

⁷ 注意，從 JMP 15 版本開始，這種類型的長條圖視圖也是 JMP 圖形生成器中的一個本地選項，因此在本示例中使用 R 僅為說明之用而非必要選項。

⁸ 參見 cran.r-project.org/web/packages/ggplot2/index.html。

要執行此操作，請選擇一個資料檔案，然後打開並運行此腳本，該腳本將打開一個含有已打開資料表中列的對話方塊。通過與此提示進行交互，創建生成的 R 圖。



要使用此腳本，您無需編寫 R 或 JSL 代碼。但是，如果您想創建一些類似的內容，底層的 JSL (包含 R 代碼片段) 就在此處顯示 (可從該示例的連結予以下載)：

The figure displays three windows from the JMP interface:

- R Script Editor:** Shows the R code for the 'R Plotting' script. It includes sections for 'Plot Functions with R v5', 'DensityWithMeans', 'DensityWithMeans', and 'RunClose'. The 'DensityWithMeans' section contains R code for creating a density plot with means for each group.
- JSL Script Window:** Shows the JSL code for the 'R Plotting' script. It includes sections for 'LoadScript', 'DrawWindow', 'CreateFieldsAndButtons', and 'R Plotting'. The 'R Plotting' section contains JSL code for creating a window and buttons to run the R script.
- R Plotting Window:** Shows the 'Imported Plot' dialog from the R Plotting interface, identical to the one in the first figure.

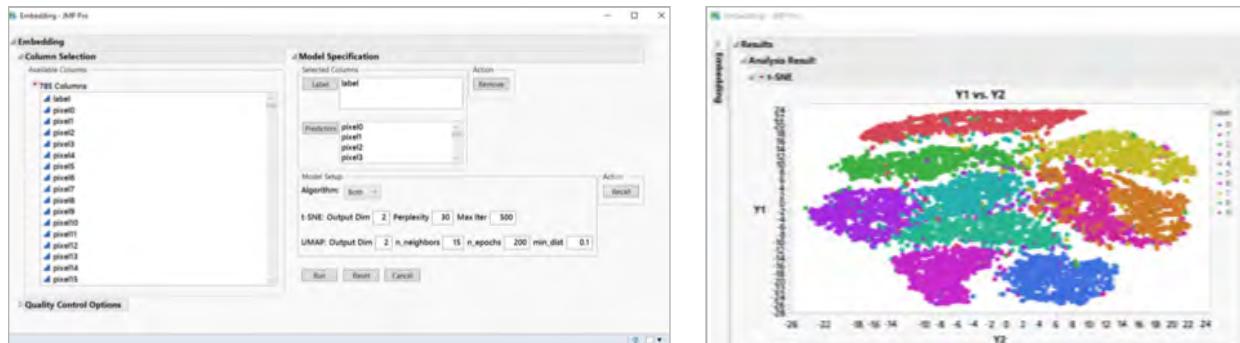
如需瞭解更多資訊並下載此外掛程式，請造訪 JMP 社區：community.jmp.com/docs/DOC-6472。

使用 t-SNE 和 UMAP 外掛程式進行資料視覺化：

此腳本使用了 R 中的 t-SNE 和 UMAP 套件⁹，當在 JMP 外掛程式中進行封裝後，此外掛程式將更加簡單易用。

t 分佈隨機近鄰嵌入 (t-SNE) 和均勻流形近似與投影 (UMAP) 是近年來在影像處理、文本挖掘和基因組學等領域日益流行的非線性降維與視覺化演算法。此外掛程式為這兩個R套件提供了一個使用者友好的介面，支援資料表導航、資料品質控制、稀疏處理、直觀參數化和互動式結果解釋。

⁹ 參見 cran.r-project.org/web/packages/Rtsne/ 和 cran.r-project.org/web/packages/umap/。



在創建類似外掛程式時，您只需要一位用戶開發R或Python代碼以及封裝它的JSL代碼。然後，該用戶將使用JMP外掛程式生成器將其封裝在一個JMP外掛程式中。¹⁰ 生成的.jmpaddin檔可以通過電子郵件發送給其他使用者，也可以發佈到網路上。¹¹ 此過程允許後續用戶運行該外掛程式，無需編寫、查看或編輯代碼——而是通過與JMP易於使用的點擊顯示交互進行。

我們無法看到在後臺運行的代碼，它的開始過程如下（R 代碼為紫色）：

```

/* scripts/embedding_031419
Run Script Debug Script Reformat Script Balance Import as Data
*/
Name: Embedding
Description: This is an add-in that provide access to t-SNE and UMAP R packages.
It has enables basic quality control, sparsity handling, parameter specification,
and result interpretations.
Author: Meijian Guan
Version: v1.2 3/14/2019
ChangeLog:
  v1.2: Fixed a bug for Rtsne package where too many columns would cause stack overflow.
  v1.1: Fixed a bug that could cause "Issues found in R..." error message. Fixed a bug when both algorithms are selected and no label is selected.
  v1: Initial version
*/
Names.Default.To.Here();
Include("JSL_Utils.vsl");
if(hostIs("Windows"),_slash="\\";_slash="/");
_addinPath=$ADDIN_HOME("con.jmp.embedding");
if(hostIs("Windows"),
  _addinPath=_Convert.File.Path(_addinPath_,windows));
//clear Log();
//closeAllDataTables,NoSave();
//remove("here")<>remove(namespace("here")<>getkeys());
//label=labelY;
//label<0;
//data4R=data2R;
//algorthm="Both";
//algorthm="Both";
//data2R=inDataRep(inData,predictor,labelY);

//mtx2R=getData2R<>Get as Matrix;
//mtx4R=Sparse.SVD(mtx2R,20);
//data4R=asTable(mtx4R)<>Invisible();
//dim(data4R);
//dim(data4R);
//dim(data4R);
talk2R=function(data4R,label,algorthm,dim=2,perplexity=2,iter=500, n_comp=2, n_neigb=15, n_epoch=200, dist=0.1,
  {Default Local};

LabelText="";
if(label==labelY);
 LabelText=eval.insert("\n
  Labels<-inData$inq[,names(inData$inq) %in% label]
  Labels<-as.list(Labels)
  \n");
 LabelText="";
);
print(algorthm||" is selected!");
Text$eval.insert("\n
  data4R=asData.frame(data4R)
  label=unlist(label)
  cat("Labels: ",label," \n")
  #print(length(label))
  cat("\n");

#remove duplicated observations from both datasets
inData=inData[inData4R[!(duplicated(data4R[,names(data4R) %in% label])),] #allow excluding multiple labels
#remove duplicated data4R[!(duplicated(data4R[,names(data4R) %in% label])),]
cat("dim of inData$inq is: ",dim(inData$inq),"\n")
#Labels$inq
cat("Labels$inq for Run: ","\n");
if("algorthm"]=="t-SNE"{
  cat("algorthm are running t-SNE", "\n")
  library("Rtsne")
  inData=inData[,matrix(inData$inq[,names(inData$inq) %in% label])
  cat("dim of inData$inq is: ",dim(inData$inq),"\n")
  tsne <- Rtsne(inData$inq, dim = "dim", perplexity="perplexity", verbose=TRUE, max_iter = ^iter^,pca=F)
  output=outputY
  head(outputY)
}else if("algorthm"]=="UMAP"{
  cat("algorthm are running UMAP", "\n")
  inData$inq
  inData$inq
}
  
```

¹⁰ 如需瞭解如何從JSL腳本創建JMP外掛程式的更多資訊，請參見 jmp.com/content/dam/jmp/documents/en/academic/learning-library/01-add-in-builder.pdf。

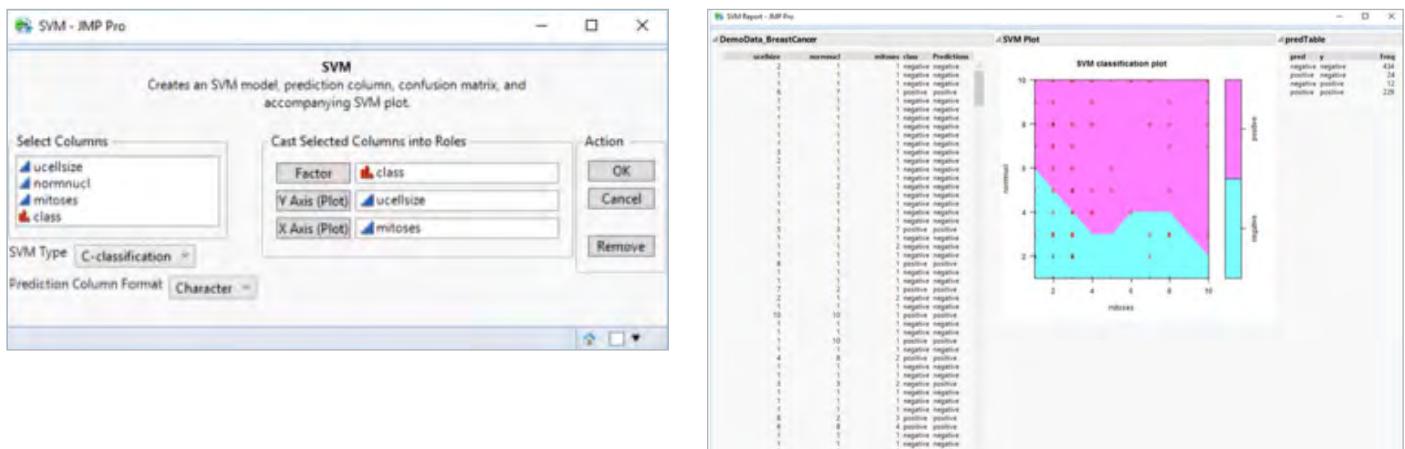
¹¹ 造訪 community.jmp.com，將其發佈到JMP用戶社區，以便其他用戶也可以找到。

一旦編寫完成，後續用戶根本無需查看此代碼或與此代碼進行互動。此功能允許您一次性編寫並將其封裝在外掛程式之中。現在，您擁有了一個點擊介面，可以調用 R 或 Python 函數。

如需瞭解更多資訊並下載此外掛程式，請造訪 JMP 社區：community.jmp.com/t5/JMP-Add-Ins/Data-visualization-with-t-SNE-and-UMAP/ta-p/177969。

SVM 外掛程式：

此外掛程式會從 e1071 R 套件中調用 svm 功能。¹² SVM 外掛程式的用戶與此對話方塊進行交互，以獲取生成的 SVM¹³ 分類報表：



用戶無法看到後臺運行的代碼：

```
library(e1071)
library(gridExtra)

theText <- paste("x <- subset(dt, select=-", colnames(factor)[1], ")")
eval(parse(text=theText))
y <- factor
svm_model <- svm(x, y, type=svmType)
modelText <- paste("second.svm <- svm(", colnames(factor)[1], " ~ ., data = dt)")
eval(parse(text=modelText))

summary(svm_model)
summary(second.svm)
pred <- predict(svm_model, x)
pred <- as.vector(pred)

if(is.list(y)) {y <- unlist(y)}
predTable <- table(pred,y)
predTable <- as.data.frame(predTable)

eval(parse(paste(colnames(y_var)[1], "<- as.vector(as.matrix(y_var))")))
eval(parse(text=paste(colnames(x_var)[1], "<- as.vector(as.matrix(x_var))"))

max1 <- eval(parse(paste("max(", colnames(y_var)[1], ",)")))
max2 <- eval(parse(paste("max(", colnames(x_var)[1], ",)")))
gridSize <- eval(parse(paste("max(c(", max1, ",", max2, ",))"))

eval(parse(text=paste("plot(second.svm, dt, ", colnames(y_var)[1], " ~ ",
colnames(x_var)[1], ", fill=TRUE, grid=gridSize)")))
```

使用 JMP 到 R 的外掛程式生成器來創建此外掛程式將更加容易（參見下一個示例）。通過 JMP 到 R 的外掛程式生成器，您可以完全跳過 JSL 編碼且僅需與某些對話方塊進行互動，在提示框中放入 R 或 Python 代碼。

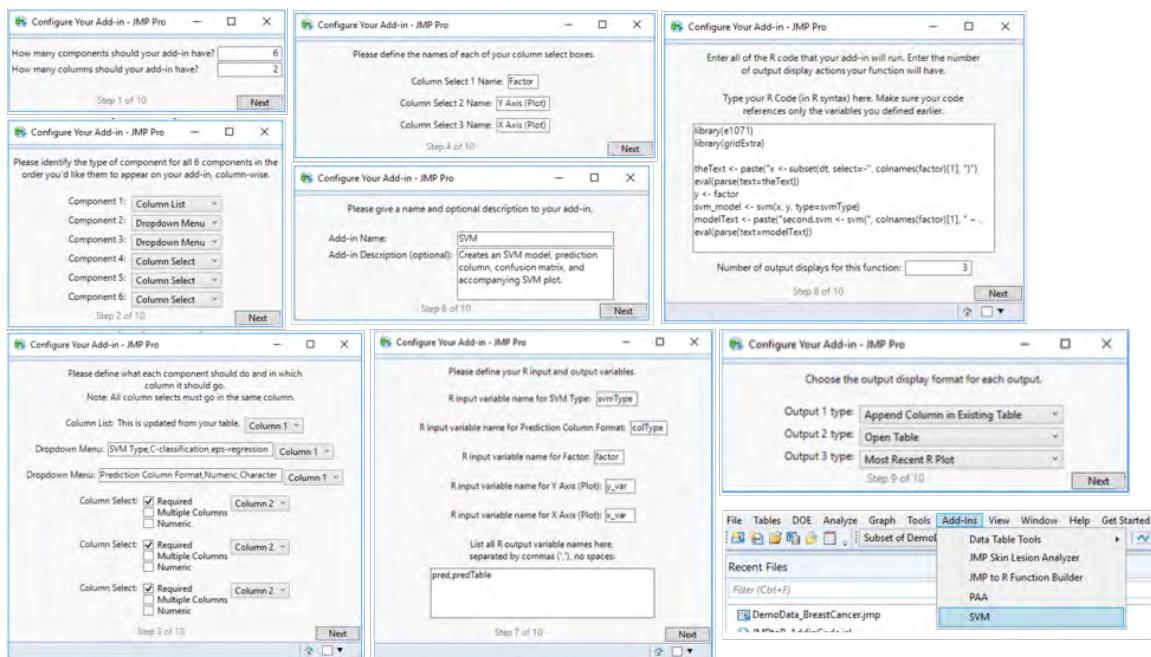
¹² 參見 cran.r-project.org/web/packages/e1071/index.html。

¹³ JMP Pro 在 15 版本的預測建模套件中增加了 SVM 平臺，因此，此示例旨在說明和 R 連接的強大功能，但 JMP Pro 15 的使用者實際上並不需要此項功能。

JMP 到 R 的外掛程式生成器：在沒有 JSL 的情況下使用 JMP 到 R 的連接

該工具允許用戶定義和使用執行 R 功能的 JMP 定制外掛程式。要使用此介面生成器，您首先需要開發要發送給 R 的 R 代碼，但是無需編寫任何 JSL 代碼。類似地，新點擊介面的終端使用者將無需使用或看到 R 代碼——他們只會看到易於使用的點擊介面。（注意：此外掛程式僅適用於 Windows 作業系統，並不適用於 Mac。）

此外掛程式允許您在使用 JMP 和 R 連接時跳過 JSL 部分。取而代之的是，您將與一些對話方塊提示進行互動，以創建要包含在外掛程式中的方框、按鈕以及提示。此處顯示了大部分的步驟：



上述 SVM 外掛程式示例是使用 JMP to R Add-In Builder 創建的。

如需瞭解更多資訊並下載 JMP to R Add-In Builder 外掛程式，請造訪 JMP 社區：

community.jmp.com/t5/JMP-Add-Ins/The-JMP-to-R-Add-In-Builder/ta-p/43879。

結論

在本指南中，我們介紹了 Python 評分代碼的生成、JMP 腳本中的 Python 連接以及 JMP 腳本中的 R 連接。我們還提供了其他示例的連結，包括一些在 JMP 中使用外掛程式生成器功能的 R 高級示例。有關這些主題的更多資訊，請參見本文中提供的註腳和連結。如果您有任何疑問、需要關於實現某種技術的幫助或單純想分享您的發現，請造訪 community.jmp.com，發文至我們的 JMP 用戶社區。

關於 SAS 和 JMP

全球分析領導者SAS 於1989 年開發了JMP（發音為“jump”），科學家、工程師和其他資料分析師能夠通過視覺化和互動式的方式深入挖掘和分析資料。如今，JMP 已由一款單一產品發展成為一個統計發現工具系列，JMP 產品家族中的每款產品都是針對特定需求而量身定制的。SAS 聯合創始人兼執行副總裁 John Sall 負責領導 JMP 事業部。

