JMP® 시너지 효과: JMP® 및 JMP® Pro를 Python 및 R과 함께 사용하기

백서

목차

서론

JMP는 SAS에서 개발한 Windows 및 맥 데스크탑용 소프트웨어 패키지로, 데이터 시각화 및 통계 분석과 관련된 다양한 기능을 제공합니다. JMP는 대화식 구성과 동적 연결을 통해 데이터 탐색을 흥미롭고 통찰력있게 해주며, 다수의 고급 분석 옵션을 갖춰 데이터 탐색가와 분석가의 요구 사항을 충족시켜줍니다. 그럼에도, JMP를 Python 또는 R과 같은 오픈 소스 도구와 함께 사용하고 싶거나 사용해야 할 경우가 생길 수 있습니다.

다음과 같은 상황을 고려해보십시오:

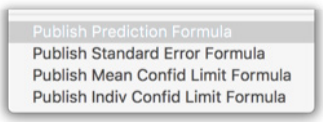
- 기업 내부 통계학자로, R과 같이 검증되지 않은 소프트웨어를 1차 분석에 사용하는 것이 금지되어 있지만, JMP의 검증된 방법과 함께 R 패키지의 새로운 방법론을 병행하여 탐구하려고 합니다.
- Python 또는 R 프로그래머가 JMP에서 사용할 수 있는 동적 연결, 고급 비주얼 기능 또는 강력한 분석 기능을 활용하려고 합니다.
- 회사의 데이터 분석 부서에서 R 또는 Python 코드를 JMP에서 실행할 수 있는 애드인으로 개발하여, 미래의 사용자들이 직접 R 또는 Python 코드를 작성하지 않고도 GUI 방식의 애드인을 사용할 수 있도록 하려고 합니다.
- 통계 수업 중 특정 오픈 소스 패키지를 소개하고 싶은데, 학생이 직접 코딩할 필요 없이 마우스로 조작할 수 있었으면 합니다.

JMP 소프트웨어의 GUI와 오픈 소스(또는 기타) 도구를 연결하는 이유가 무엇이든, 이 가이드를 참조하여 JMP에서 Python 및 R과의 연동을 시작할 수 있습니다.[1]

JMP®와 Python

JMP Pro 모델에서 Python 코드 생성

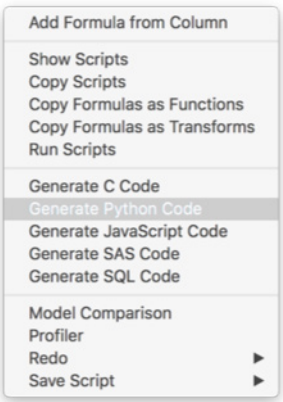
JMP Pro를 이용해 JMP에서 생성하는 예측 모형에 사용할 Python 코드를 생성할 수 있습니다. 이는 JMP Pro에서 모형을 생성 및 탐색하고, 모형들을 비교한 후 다른 개발 환경에 배포하고 싶을 때 유용합니다. JMP Pro를 사용하면 간단한 방법으로 여러 모형을 구성하고, 경쟁 모형들을 비교하고, 더 나아가 계산식 저장소 플랫폼에서 평균 모델(앙상블)을 생성할 수 있습니다.



먼저 JMP 분석 메뉴 아래의 도구들을 사용하여 모형을 적합시킵니다. 모델을 계산식 저장소에 저장하려면 "확률 계산식 게시"(분류 모형의 경우) 또는 "예측 계산식 게시"(예측 모형의 경우) 옵션을 사용합니다. 일반적으로 두 옵션 모두 모형 결과에 대한 빨간색 삼각형 옵션이나 "열 저장" 아래에서 찾을 수 있습니다.

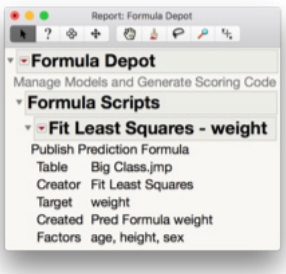
[1] JMP는 또한 SAS 및 Matlab(R과 Python 외에)에도 연결되며, Python 외에 C, JavaScript, SAS 및 SQL 모형 점수 매기기 코드도 생성합니다. 이러한 JMP 연결에 대한 자세한 정보와 시작하기 팁 및 추가 예제들을 jmp.com/support/help/en/15.1/#page/jmp/extending-jmp.shtml에서 확인할 수 있습니다.

키워드인 "게시"(또는 "열에서 계산식 추가" 옵션 사용)를 선택하면 모형이 계산식 저장소로 전송됩니다. 계산식 저장소는 경쟁 모형 하나 또는 여러 개를 저장하고 비교하며 앙상블 모형을 생성하여 다양한 개발 환경에 배포할 수 있는 플랫폼입니다.



"Python 코드 생성" 옵션은 개별 모형의 빨간색 삼각형 아래(**가중치** 예측의 경우 "최소 제곱 적합" 옵션) 또는 계산식 저장소 창에서 저장한 모형 집합에 대한 코드를 작성하는 데 필요한 계산식 저장소 빨간색 삼각형 아래에 있는 옵션입니다.

아래 제시된 대로 생성되는 Python 코드에는 코드의 첫 번째 섹션에서 호출되는 몇몇 필수 종속 요소가 포함됩니다. 그리고 저작권 정보, 변수 이름과 유형을 생성하는 몇 가지 Python 코드, 마지막으로 예측 변수(예시에서는 *연령과 신장, 성별*)에 대해 확인된 값과 함께 새로운 관측치에 대한 반응 변수를 예측하는 데 사용될 계산식이 차례로 나옵니다.



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": u"Fit Least Squares", "modelName": u"", "predicted": u"weight",
    "table": u"Big Class.jmp", "version": u"14.1.0", "timestamp": u"2018-08-08T04:02:07Z"}


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


def getOutputMetadata():
    return {
        u"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 = -19.7698493666905
    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"]
```

이 주제에 대한 추가 자료는 jmp.com/support/help/en/15.1/#page/jmp/formula-depot.shtml에서 찾아볼 수 있습니다.

JMP와 Python 연결의 기본 원리

JMP에서 Python 모델 스코어링 코드를 생성하는 것 외에, JMP와 Python 간 연결을 열고 JMP와 Python 코드 조합을 사용하여 Python과 JMP 간 데이터와 결과, 작업을 송수신할 수 있습니다. (JMP와 R 연결도 같은 방식으로 작동합니다.) JMP에서 Python 연결을 사용하려면 JMP 스크립트 언어(JSL)를 Python 코드 래퍼로 사용합니다.

설명을 위해 간단한 예시를 소개합니다. 사용자 시스템에 Python[2]이 설치되어 있어야 합니다. 대부분의 컴퓨터에서 동작하지만 버전 등에 따라 차이가 있을 있습니다.

(1) JMP에서 Python 연결을 엽니다.

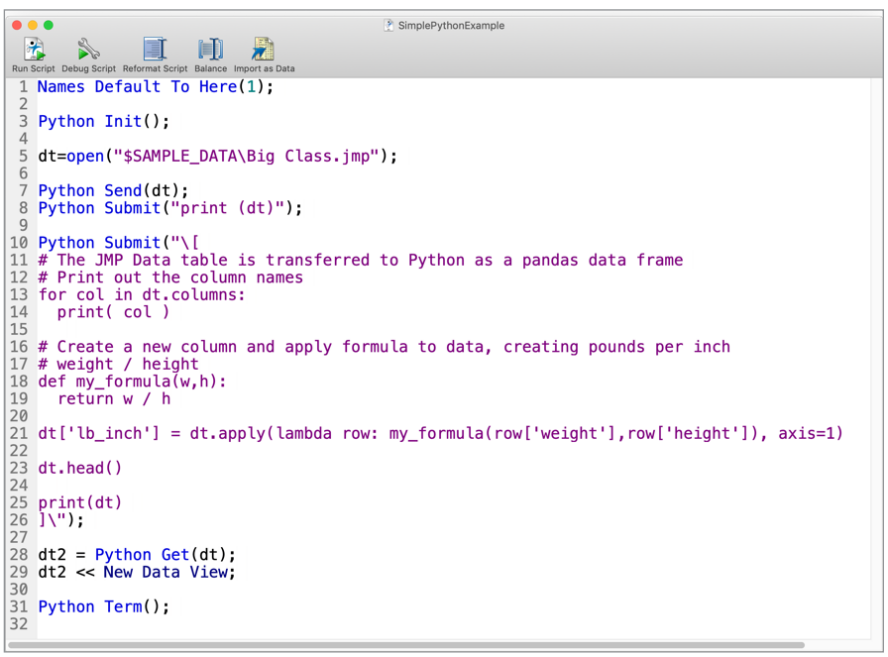
(2) JMP에서 Python으로 데이터 보내기 같은 작업 하나를 수행합니다.

(3) Python으로 Python 코드 바로 보내기 같은 다른 작업을 수행하여, 새 변수를 생성합니다.

(4) Python의 결과 또는 데이터를 다시 JMP로 불러오기(또는 JMP에서 결과 시각화) 같은 작업을 추가로 수행합니다.

(5) Python 연결을 종료합니다.

위 예제에 대한 코드입니다. 계속 진행하려면 JMP를 먼저 열어야 합니다. 계속해서 '파일 > 새로 만들기 > 새 스크립트'를 차례로 선택하여 빈 스크립트 창을 새로 엽니다. 그리고 다음 내용을 입력합니다.




```
1  Names Default To Here(1);
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
```

위 스크립트에는 9개의 JSL 코드(파란색으로 표시)가 있습니다. 1행은 생성하는 파일 또는 변수 참조를 스크립트와 연관되도록 도움을 줍니다. 스크립트의 3행 "**`Python Init`**`();`"에 의해 Python이 열립니다.[3]

[2] python.org.

[3] Python 설치 및 버전 생성에 대한 자세한 내용은 jmp.com/support/help/en/15.1/#page/jmp/install-python.shtml# 및 jmp.com/support/help/en/15.1/#page/jmp/troubleshooting-the-jmp-python-integration.shtml#ww822804에서 확인할 수 있습니다.

5행 “**dt = Open**(**“$SAMPLE_DATA/Big Class.jmp”**);"은 JMP 샘플 데이터 디렉토리에 있는 데이터 집합을 열고 이름을 “dt”로 지정하라는 명령이 JMP에 전달됩니다.[4] 예제의 데이터 집합 이름은 "Big Class.jmp"입니다.

7행 “**Python Send(dt);**"에 의해 JMP에서 Python으로 “dt” 데이터 테이블이 전송됩니다. 이 JMP 데이터 테이블은 Pandas 데이터 프레임로 전송됩니다. 8행 “**Python Submit(“print(dt)”);**”에 의해 Python으로 일부 Python 코드가 제출됩니다. 인용부호로 묶인 정보는 Python 스크립트입니다. 8행의 나머지 부분은 JSL 코드 래퍼입니다. 필요한 경우 JMP 로그 창을 사용하여 코드 제출을 모니터링하고 문제를 해결하는 데 유용합니다. JMP 메뉴 표시줄로 이동하여 '창 → 로그'를 선택하여 JMP 로그를 엽니다. 여기에서 내부 Python 오류 메시지나 결과를 확인할 수 있습니다. 예를 들어 이 코드를 실행한 후 이 시점까지 로그를 검토하면 로그에서 다음과 같은 Python 출력이 확인됩니다.

```
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
```

10-26행에는 긴 “**Python Submit();**” 명령문이 들어 있습니다. JSL 코드에 삽입된 이 부분의 Python 코드 “**Python Submit()**” 명령문에 의해 열 이름(로그에 사용된 이름)이 인쇄된 후 “lb_inch = weight/height” 계산식을 만드는 새로운 열이 생성되고, 마지막으로 로그의 기존 데이터 테이블에 새로 생성되는 열을 추가하는 새 데이터 테이블이 인쇄됩니다.

```
/*:

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
```

28행과 29행은 업데이트된 데이터 테이블을 소프트웨어의 작업 메모리로 다시 가져와 새로운 JMP 데이터 테이블로서 엽니다.

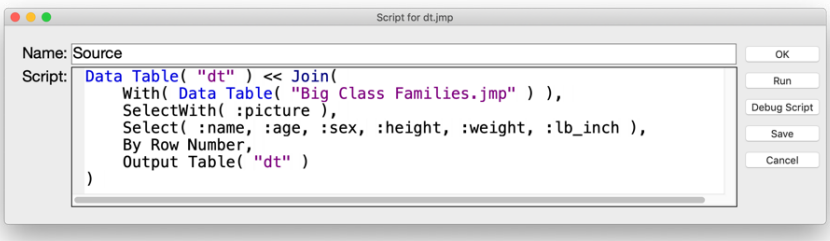
코드의 마지막 행인 “**Python Term();**”에 의해 Python과 연결이 끊어집니다.

[4] R 예제에서 데이터 파일을 열기 위해 가리키는 JSL 코드에 약간 다른 행을 사용하였음에 주목하십시오. 이 경우에는 JSL을 사용하여 데이터 파일을 열기 위해 데이터를 가리키는 또 하나의 방법입니다.

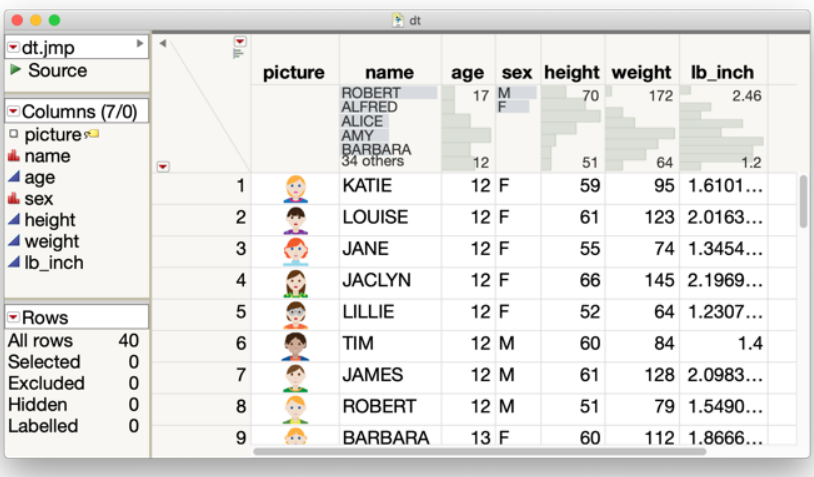
기타 JMP 활용 팁– 그래프에 이미지 마커 추가

이 단순한 예제의 마지막 섹션은 JMP to Python 연결을 더 이상 사용하지는 않지만 Python(또는 R이나 기타 연결된 소프트웨어)과 연결하여 JMP를 사용할 때 가능한 이점들을 보여줍니다. 특히 JMP와 Python 사이 호환 특성을 통해 미리 작성된 Python 코드 또는 특수 Python 패키지를 실행한 다음, JMP의 다양한 동적 기능을 사용하여 최종 분석 또는 시각화 성능을 개선할 수 있습니다.

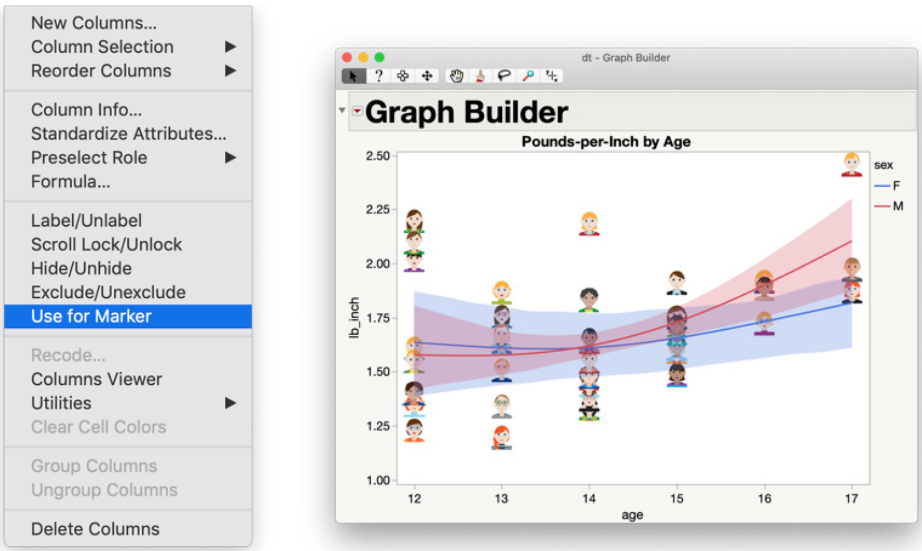
예를 들어, "dt" 데이터 테이블을 Jmp로 다시 가져온 후, 일부 우수한 Jmp 시각화 기능을 사용하여 몇 번의 클릭이나 짧은 Jsl 스크립트만으로 그래프를 개선할 수 있습니다. 이해를 돕기 위해 Jmp의 표본 데이터 라이브러리에서 사용할 수 있는 Big Class Families 표본 데이터 집합과 "dt" 데이블을 결합하여 새 "dt"에 이미지 열을 추가해보겠습니다. 먼저, Big Class Families 표본 데이터 집합을 열고('도움말 > 표본 데이터 라이브러리 > Big Class Families'를 차례로 클릭하거나 코드 “`Dt = Open("$sample_data/big Class.jmp");`” 실행) '테이블 > 결합'(또는 다음 스크립트)를 사용하여 “dt” 테이블에 이미지 열을 결합합니다.



이러한 단계를 통해 데이터 세트에 "사진"이라는 열을 추가하고 Big Class Families 데이터 테이블에 있는 학생 이미지들을 새로 결합된 데이터 테이블로 가져옵니다.



이제 이미지들을 그림에서 표식으로 사용하여 새로운 인치당 파운드(pounds-per-inch) 변수를 시각화할 수 있습니다. 먼저, "사진" 열을 클릭하고 '열 > 표식에 사용'을 클릭하여 이미지 열을 표식으로 만듭니다.



지금까지 간단한 예제에서 JMP-Python 연결을 사용하여 JMP에서 데이터를 열고, Python으로 보내고 Python에 새로운 열을 만들어 Python panda 데이터 프레임에 추가한 다음, 업데이트된 데이터를 JMP 데이터 테이블로 다시 가져와서 JMP에서 분석하고 추가 기능을 수행해보았습니다.

Python이 연결된 단순 JMP 예제

JMP 도움말에서, '스크립트 가이드 → JMP 확장 → Python과 연결하여 작업' 아래에서 또는 jmp.com/support/help/en/15.1/#page/jmp/additional-python-integration-examples.shtml을 방문하여 JMP와 Python 교호작용에 대한 간단한 다른 예들을 확인할 수 있습니다.

다음 작업에 필요한 코드가 게시되어 있습니다.

- Python으로 데이터 테이블 보내기
- Python에서 객체 생성
- Python에서 매트릭스 작업

JMP®와 R

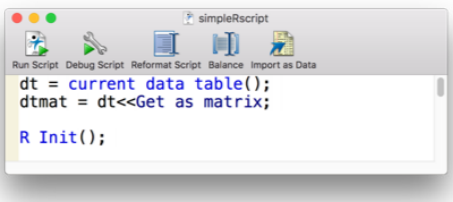
JMP와 오픈 소스 소스트웨어 R[5]을 연결하면 JMP 소프트웨어의 동적 대화형 분석 및 시각화 기능과 함께 R 기능을 사용할 수 있습니다. R 패키지에 사용하기 쉬운 메뉴 인터페이스를 만들고 JMP 요소와 R 요소를 결합할 수도 있습니다. 컴퓨터에 JMP와 R을 모두 설치하면 이러한 작업이 모두 가능합니다. 스크립트 인덱스(도움말 > 스크립트 인덱스 > R 기능)에서 연결을 수행하는 다양한 R 기능 목록(기능별 설명 및 예제 포함)을 확인할 수 있습니다.

JMP와 R 연결의 기본 원리

JMP에서 R 연결을 사용하려면 JMP 스크립트 언어(JSL)를 사용하여 R 코드를 래핑해야 합니다.

분석 및 처리를 위해 JMP에서 R로 일부 데이터를 옮겼다가 다시 JMP로 가져오는 예를 들어 보겠습니다. 진행 단계는 상당히 간단한 편입니다. 여기서는 R을 열고 JMP에서 R로 데이터를 보내고, 다시 R에서 JMP로 데이터를 가져오는 JSL 코드를 간단하게 작성할 것입니다.

먼저, 시스템에 R과 JMP가 모두 설치되어 있는지 확인합니다.



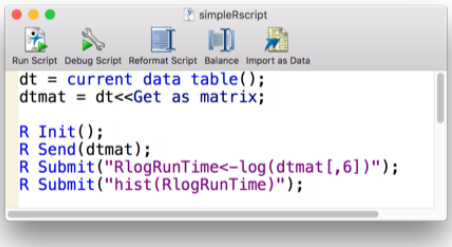
그리고 **파일 > 새로 만들기 > 새 스크립트**를 이용하여 JMP에서 스크립트 창을 엽니다.[6]

또한 JMP의 표본 데이터 집합도 열어보겠습니다. **도움말 > 표본 데이터 라이브러리**로 이동하여 **Fitness.jmp** 데이터 테이블을 엽니다. 이 테이블을 R로 보내려면 간단한 2행 JSL 코드를 사용해야 합니다. 1행은 오픈 데이터 테이블인 **Fitness.jmp**를 열고 이름을 "dt"로 바꿉니다. 2행은 이름이 바뀐 "dt" 데이터 테이블을 행렬 형식으로 변환합니다. R은 JMP와 약간 다른 데이터 구성 옵션을 제공하는데 JMP와 R에서 모두 쉽게 사용할 수 있는 단순한 프레임워크가 행렬 형식입니다. 다음에는 JMP에서 R을 호출합니다. R이 같은 시스템에 설치되어 있으면 JMP가 이를 찾기 때문에 사용자가 설치된 R로 연결을 직접 설정하거나 JMP로 R을 가리킬 필요가 없습니다. JSL 코드 행 "`R Init();`"만 입력하면 JMP가 시스템에서 R을 검색하여 초기화하거나 엽니다.

[5] cran.r-project.org.

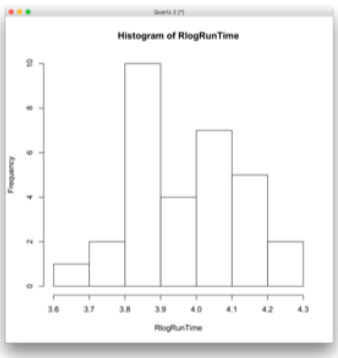
[6] R 설치 및 버전 생성에 대한 자세한 내용은 jmp.com/support/help/en/15.1/#page/jmp/installing-r.shtml에서 확인할 수 있습니다.

이제 JMP와 R 사이를 오고가며 작업할 준비가 되었습니다. 다음 코드의 1행에 의해 dtmat이라는 데이터의 행렬 버전이 R로 전송됩니다. 2행에 의해 R에서 실행될 R 코드가 전송됩니다.



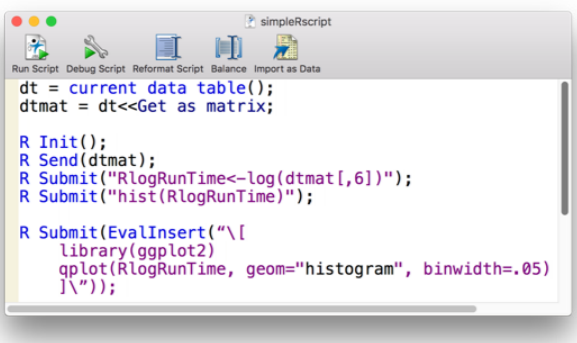
"`RlogRunTime<-log(dtmat[,6]);`" 행에 의해 입력 데이터 행렬의 6번째 열 (Fitness 데이터 세트의 "Mile Run Time")에 로그 변환이 수행됩니다. 이제 변환 결과를 "RlogRunTime"라고 합니다(R 코드 단편에서 이미 할당한 이름이기 때문임).

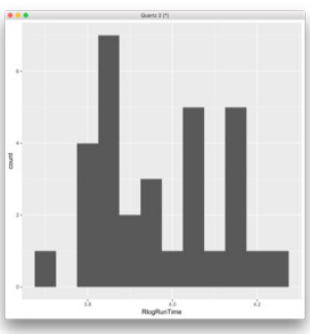
"`R Submit("hist(RlogRunTime)");`" 행에 의해 R에서 히스토그램이 생성되며, 나중에 이 코드 행을 실행하고 나면 사용자 바탕 화면에 나타납니다.



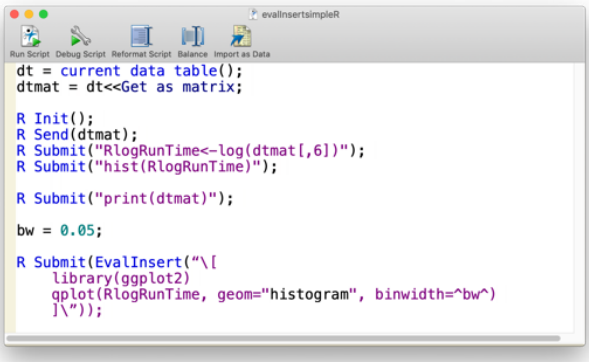
코드를 실행하려면 행을 하나 이상 선택(각 행이 ";"로 끝나야 함)하고 스크립트 창 맨 위에서 "스크립트 실행" 버튼을 클릭합니다.

여러 행으로 구성된 R 코드를 JSL 래퍼로 보내려는 경우, "`R Submit(EvalInsert("\[____]\"))`"를 단순한 "`R Submit("___")`" 대신 사용할 수 있습니다. 나중에 R에서 "`ggplot2`" 패키지를 사용하여 히스토그램을 생성하는 다음 R 코드에서 확인하게 될 것입니다.

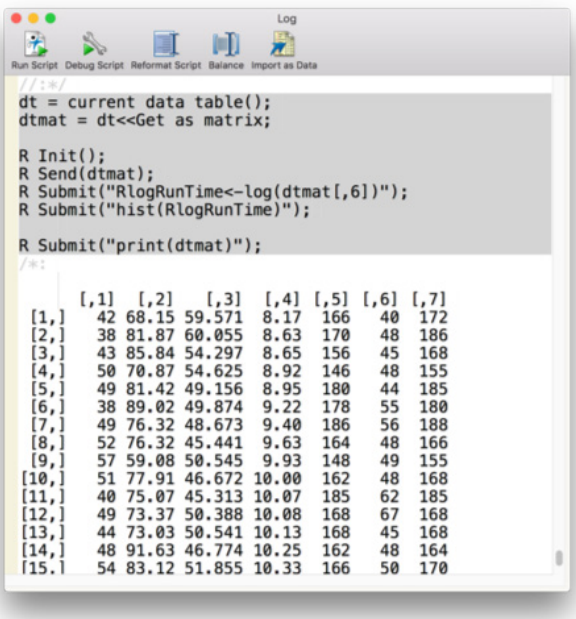




이 "**EvalInsert**()"는 일반적으로 JSL에서 대체물을 생성하거나 문자열 안의 표현식을 평가해야 할 때에도 유용합니다. 예를 들어 JSL 스크립트에서 변수로서 R 히스토그램의 계급 너비를 지정한 다음, 아래와 같이 "**EvalInsert**()" 코드를 사용하여 변수를 호출할 수 있습니다.

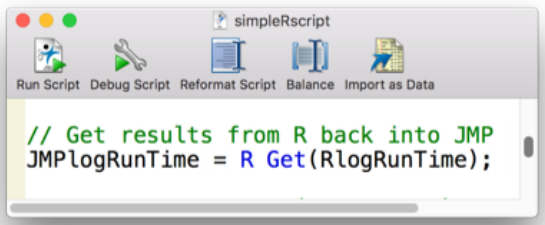


코드 제출을 모니터링하고, 필요하면 JMP 로그 창을 사용하여 문제를 해결합니다. JMP 메뉴 표시줄로 이동하여 '창 → 로그'를 선택하여 JMP 로그를 엽니다. 여기에서 모든 내부 R 오류 메시지나 결과를 확인할 수 있습니다.

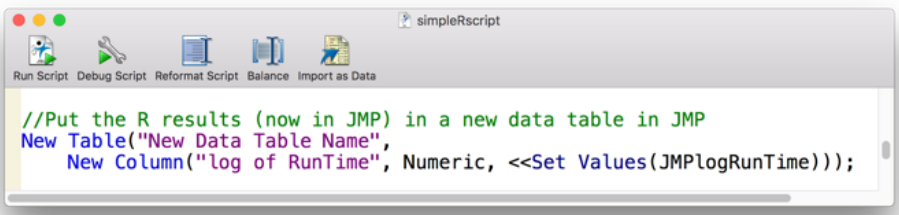


물론 JMP에서 분포 플랫폼을 사용하여 이 히스토그램을 생성할 수도 있습니다. 다음 두 가지 방법으로 JMP 함수를 선택할 수 있습니다.

먼저, "**RlogRunTime**"을 R에서 JMP로 다시 불어와야 합니다. 코드의 이 섹션에 의해 R에서 새로 변환된 "**RlogRunTime**"이라는 데이터를 취하여 "**JMPlogRunTime**"라는 새로운 이름으로 다시 JMP로 불러옵니다.



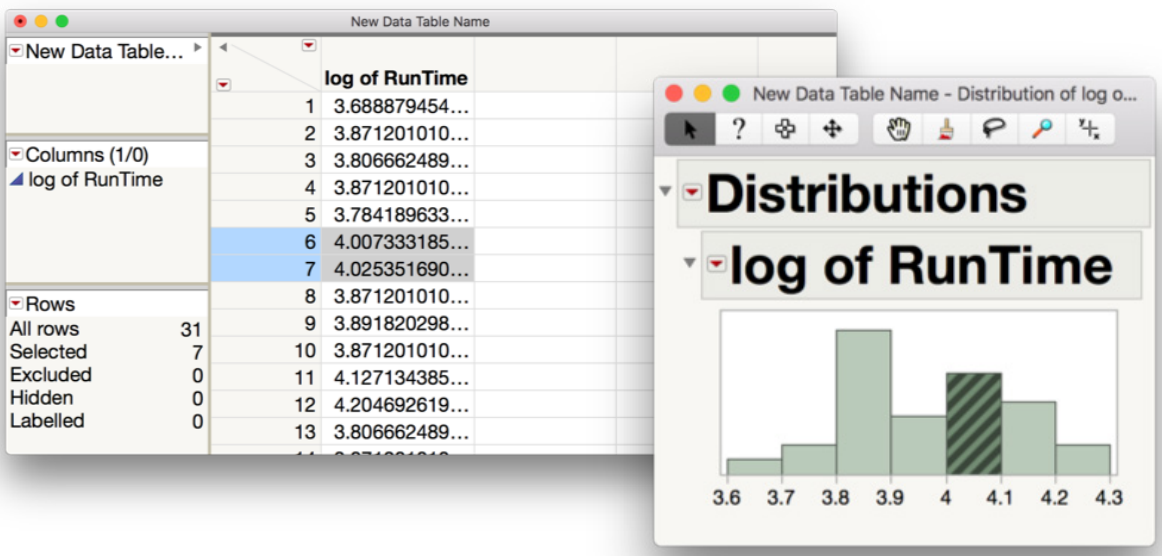
또한 새로 불러온 JMP 데이터를 저장하고 추가 작업을 수행할 수 있으려면 내부 메모리에서 JMP의 새로운 데이터 테이블로 옮겨야 합니다. 아래 코드에 의해 "**JMPlogRunTime**"의 데이터로 구성된 열 한 개가 포함된 새로운 JMP 데이터 테이블이 생성됩니다.



계속해서 다음 두 가지 방법 중 하나로 추가 JMP 기능을 사용하도록 선택할 수 있습니다.

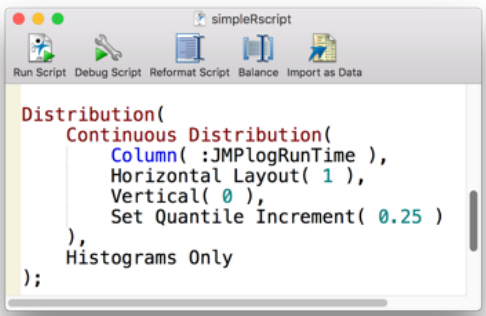
(1) 새 JMP 데이터 테이블에서 작업하는 데 마우스 클릭형 메뉴의 JMP 기능들을 사용합니다.

여기 예제에서는 '분석 > 분포' 옵션을 사용하여 JMP에서 대화형 히스토그램을 생성합니다.



(2) 동일한 마우스 클릭형 JMP 기능들을 호출하는 JSL을 사용하여 스크립트 창에서 작업을 계속합니다.

JSL 지식이 있으면 스크립트 창에서 직접 추가 JMP 작업 수행 스크립트를 작성할 수 있습니다. 마우스 클릭 작업에 해당하는 JSL에 익숙하지 않은 경우, 마우스 클릭으로 원하는 작업을 한 번 생성하는 방법으로 해당 JSL을 쉽게 작성할 수 있습니다. 그런 다음, 결과 창에서 빨간색 삼각형을 클릭하고 '스크립트 창에 스크립트 복사'를 선택합니다. 새 데이터 테이블에 맞게 스크립트를 업데이트하려면 JSL 스크립트를 편집하여 적절한 변수에 적용합니다.



R이 연결된 단순 JMP 예제

JMP 도움말에서 "R"을 검색하거나 '스크립트 가이드 → JMP 확장 → R과 연결하여 작업' 아래에서 JMP와 R 사이 교호 방식을 보여주는 간단한 예들을 확인할 수 있습니다.

JMP 도움말에서 다음 작업에 필요한 코드를 찾아볼 수 있습니다.

- R로 데이터 테이블 보내기
- R에서 객체 생성
- R 기능 및 그래픽 사용
- R에서 단순한 행렬 추가
- R에서 붓스트랩 신뢰 구간 가져오기

고급 예제

R 및 Python과 함께 JMP 인터페이스 사용의 기본 원리를 이해하였으면 흥미로운 대화식 방법으로 확장할 수 있습니다.

다음은 몇 가지 예제로 연결되는 링크입니다:

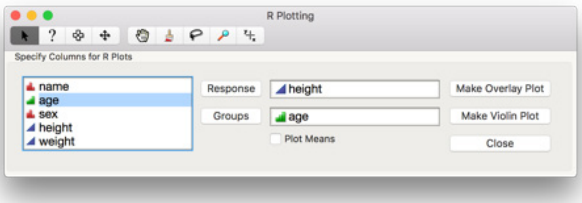
JMP와 GGPlot 간 대화식 호출 프로그램

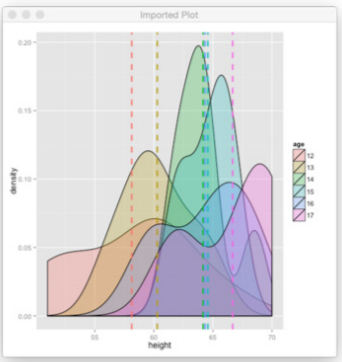
R에 JMP 연결을 활용하여 그룹화 변수 수준별로 쪼개진 평곡 밀도 그림[7](R 패키지 ggplot2 사용)[8]을 생성하는 데 이 스크립트를 활용할 수 있습니다.

[7] 이 유형의 히스토그램 보기는 버전 15부터 지원되는 JMP 그래프 빌더의 기본 옵션이므로 이 예제에서 R을 사용하는 것은 필요보다는 이해를 돕기 위한 것입니다.

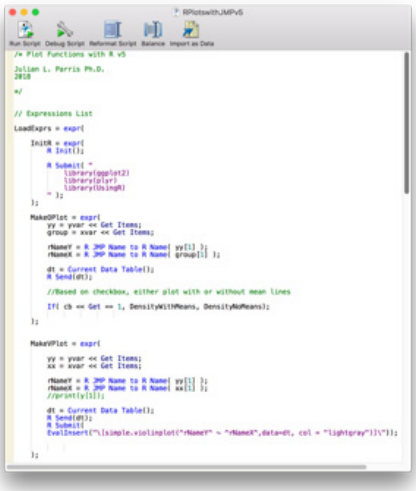
[8] cran.r-project.org/web/packages/ggplot2/index.html을 참조하십시오.

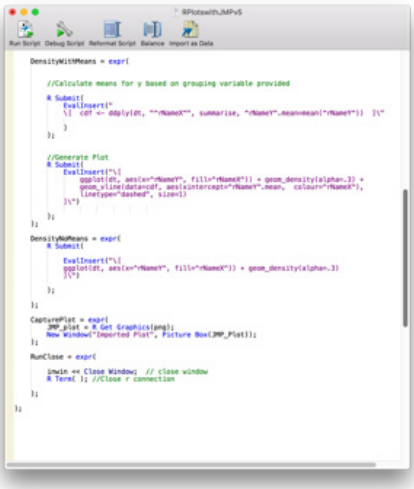
실행하려면 데이터 파일을 선택한 다음 이 스크립트를 열어서 실행합니다. 그러면 열린 데이터 테이블의 열들로 채워진 대화 상자가 열립니다. 이 대화 상자에서 작업으로 결과 R 그림이 생성됩니다.

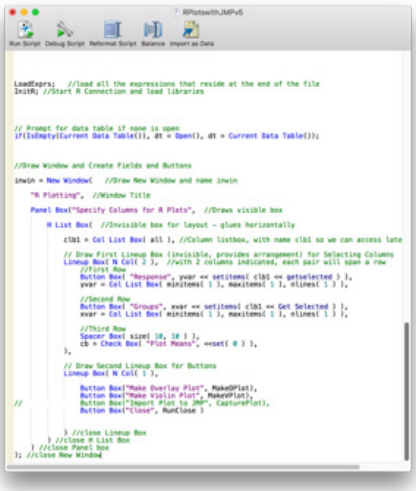




이 스크립트를 사용하기 위해 R 또는 JSL 코드를 작성할 필요는 없습니다. 그러나 비슷한 코드를 생성하고 싶은 사용자를 위해 기본 JSL(R 코드 단편 포함)을 여기에 제시합니다(이 예제 링크에서 다운로드 가능).







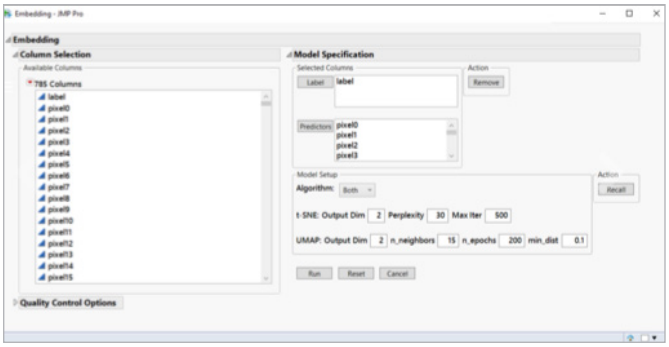
이 추가기능을 자세히 살펴보고 다운로드하려면 community.jmp.com/docs/DOC-6472의 JMP 커뮤니티를 방문하십시오.

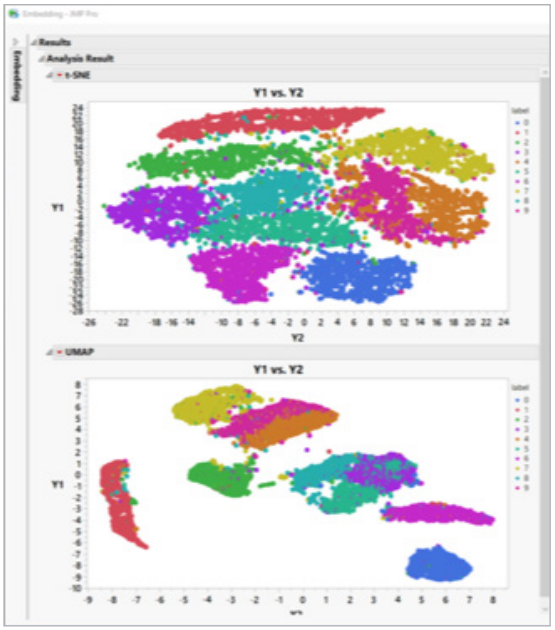
t-SNE 및 UMAP 추가기능을 사용하여 데이터 시각화

R에서 t-SNE 및 UMAP 패키지[9]를 사용하는 이 스크립트는 JMP 추가기능에 패키지로 묶일 때 한층 더 사용자 친화적인 스크립트입니다.

t-SNE(t-Distributed Stochastic Neighbor Embedding)과 UMAP(Uniform Manifold Approximation and Projection)는 비선형 차원 축소 및 시각화 알고리즘으로, 이미지 처리, 텍스트 마이닝 및 유전체학과 같은 분야에서 인기를 끌고 있습니다. 이 추가 기능은 데이터 테이블 탐색, 데이터 품질 제어, 희소성 처리, 직관적인 모수화 및 대화식 결과 해석을 가능하게 하는 사용자 중심 인터페이스를 두 가지 R 패키지에 제공합니다.

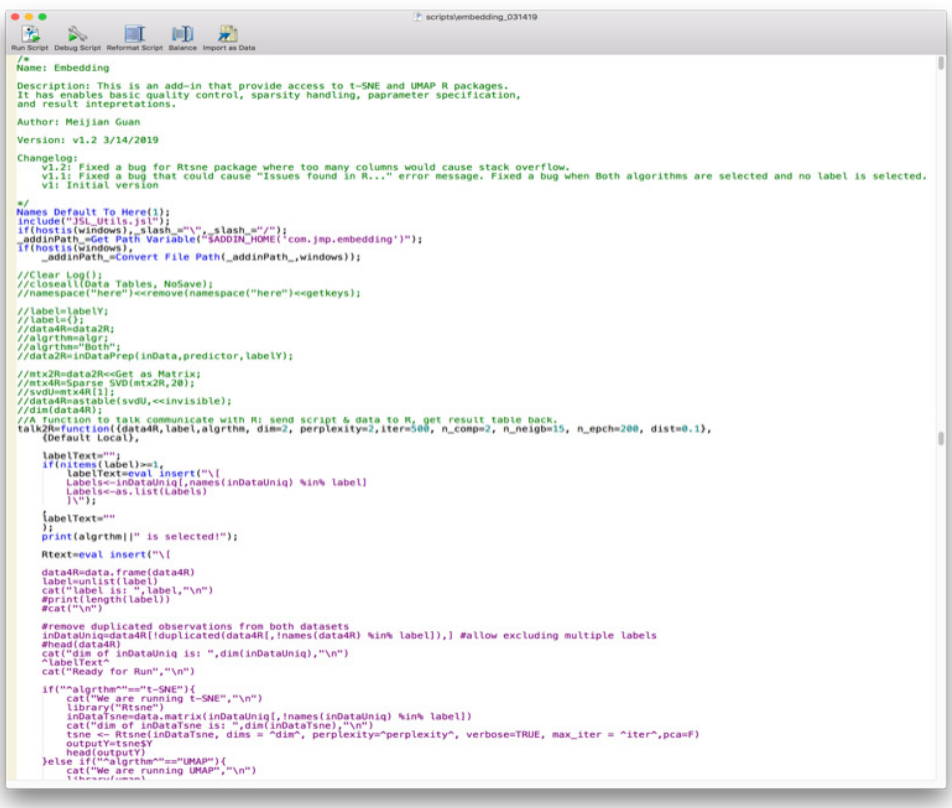
[9] cran.r-project.org/web/packages/Rtsne/ 및 cran.r-project.org/web/packages/umap/를 참조하십시오.





이와 같은 추가기능을 생성할 때 간단히 한 사용자가 R 또는 Python 코드를 작성하여 둘레를 래핑하는 JSL 코드를 추가하면 됩니다. 그런 다음 JMP 추가기능 빌더를 사용하여 JMP 추가기능에서 전체를 래핑하면 됩니다.[10] 결과로 생성되는 .jmpaddin 파일은 다른 사용자에게 이메일로 보내거나 웹사이트에 게시할 수 있습니다.[11] 이러한 방식으로 다른 사용자들이 코드를 작성, 검토 또는 편집하지 않고도 추가기능을 실행할 수 있습니다. 대신에 JMP의 사용하기 간편한 마우스 클릭 화면을 통해 작업하면 됩니다.

배경에서 실행되어 보이지 않는 코드는 다음과 같이 시작됩니다(R 코드는 보라색임).



```
/*
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, paprameter specification,
and result intepretations.

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(1);
include("JSL_Utils.jsl");
if(hostis(windows),_slash_="\",_slash_="/");
_addinPath_=Get Path Variable("$ADDIN_HOME('com.jmp.embedding')");
if(hostis(windows),
    _addinPath_=Convert File Path(_addinPath_,windows));

//Clear Log();
//closeall(Data Tables, NoSave);
//namespace("here")<<remove(namespace("here")<<getkeys);

//label=labelY;
//label={};
//data4R=data2R;
//algrthm=algr;
//algrthm="Both";
//data2R=inDataPrep(inData,predictor,labelY);

//mtx2R=data2R<<Get as Matrix;
//mtx4R=Sparse SVD(mtx2R,20);
//svdU=mtx4R[1];
//data4R=astable(svdU,<<invisible);
//dim(data4R);
//A function to talk communicate with R: send script & data to R, get result table back.
talk2R=function({data4R,label,algrthm, dim=2, perplexity=2,iter=500, n_comp=2, n_neigb=15, n_epch=200, dist=0.1},
    {Default Local},

    labelText="";
    if(nitems(label)>=1,
        labelText=eval insert("\[
        Labels<-inDataUniq[,names(inDataUniq) %in% label]
        Labels<-as.list(Labels)
        ]\");
    ,
    labelText=""
    );
    print(algrthm||" is selected!");

    Rtext=eval insert("\[

    data4R=data.frame(data4R)
    label=unlist(label)
    cat("label is: ",label,"\n")
    #print(length(label))
    #cat("\n")

    #remove duplicated observations from both datasets
    inDataUniq=data4R[!duplicated(data4R[,!names(data4R) %in% label]),] #allow excluding multiple labels
    #head(data4R)
    cat("dim of inDataUniq is: ",dim(inDataUniq),"\n")
    ^labelText^
    cat("Ready for Run","\n")

    if("^algrthm^"=="t-SNE"){
        cat("We are running t-SNE","\n")
        library("Rtsne")
        inDataTsne=data.matrix(inDataUniq[,!names(inDataUniq) %in% label])
        cat("dim of inDataTsne is: ",dim(inDataTsne),"\n")
        tsne <- Rtsne(inDataTsne, dims = ^dim^, perplexity=^perplexity^, verbose=TRUE, max_iter = ^iter^,pca=F)
        outputY=tsne$Y
        head(outputY)
    }else if("^algrthm^"=="UMAP"){
        cat("We are running UMAP","\n")
```

[10] 사용자 JSL 스크립트에서 JMP 추가기능을 생성하는 방법에 대한 자세한 내용은 jmp.com/content/dam/jmp/documents/en/academic/learning-library/01-add-in-builder.pdf를 참조하십시오.

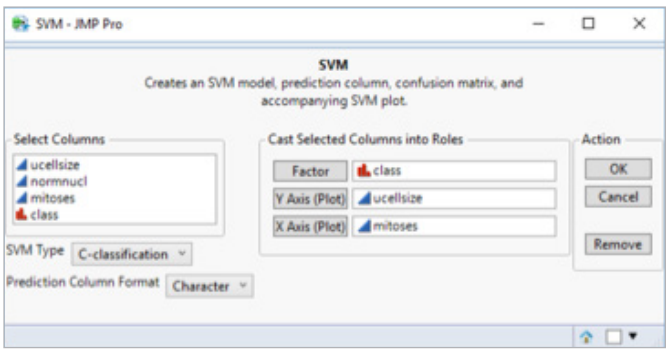
[11] 다른 사람들이 공유할 수 있도록 사용자 스크립트를 JMP 사용자 커뮤니티(community.jmp.com)에 게시합니다.

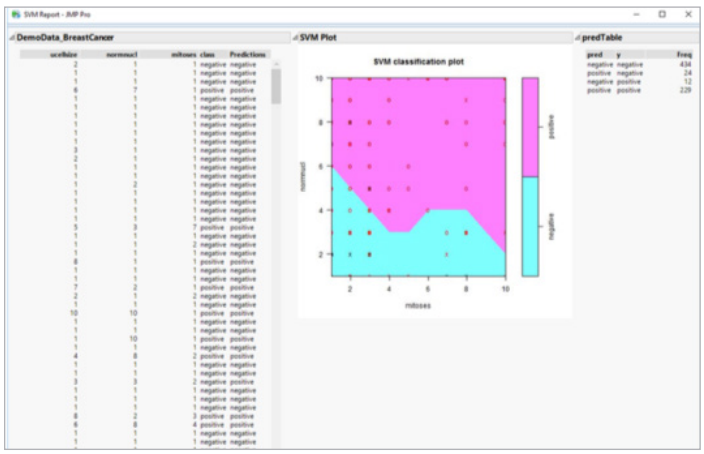
코드가 작성되면 이후 사용자들이 코드를 확인하거나 편집할 필요가 전혀 없습니다. 이 기능을 통해 코드를 한 번 작성하여 추가기능에 삽입할 수 있습니다. 이제 R 또는 Python 기능을 호출하는 마우스 클릭 인터페이스가 준비되었습니다.

이 추가기능을 자세히 살펴보고 다운로드하려면 community.jmp.com/t5/JMP-Add-Ins/Data-visualization-with-t-SNE-and-UMAP/ta-p/177969의 JMP 커뮤니티를 방문하십시오.

SVM 추가기능:

이 추가기능은 e1071 R 패키지에서 svm 함수를 호출합니다.[12] SVM 추가기능 사용자가 대화 상자에서 작업하여 SVM[13] 분류 보고서를 가져옵니다:





사용자에게 보이지 않는 것은 배경에서 실행되는 코드이기 때문입니다.

```
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(text=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(text=paste("max(", colnames(y_var)[1], ")")))
max2 <- eval(parse(text=paste("max(", colnames(x_var)[1], ")")))
gridSize <- eval(parse(text=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)")))
```

이 추가기능을 생성하는 것은 JMP - R 추가기능 빌더를 사용하는 것보다도 훨씬 쉽습니다 (다음 예제 참조). JMP - R 추가기능 빌더를 사용하면 JSL 코딩을 완전히 건너뛰고 간단히 몇몇 대화 상자에서 작업을 통해 나타나는 상자에 R 또는 Python 코드를 끌어다 놓을 수 있습니다.

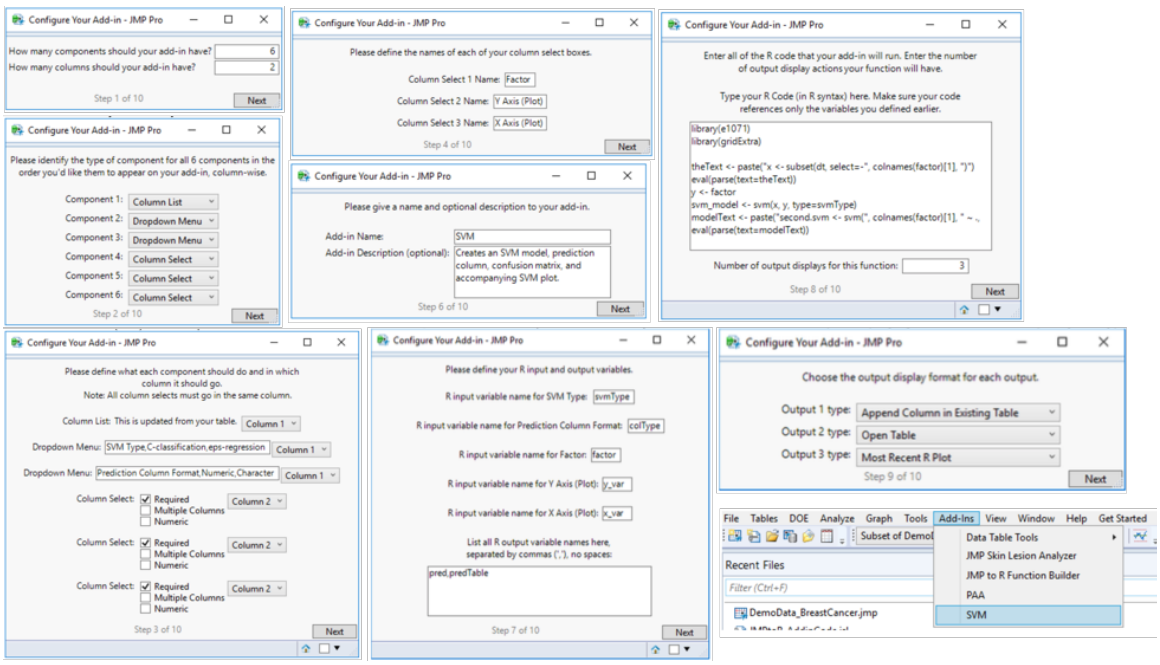
[12] cran.r-project.org/web/packages/e1071/index.html을 참조하십시오.

[13] JMP Pro는 버전 15의 예측 모형 제품군에 SVM을 추가했습니다. 따라서 이 예제는 R과 연결의 이점을 보여주기 위한 것일뿐 JMP Pro 15 사용자에게는 필요하지 않습니다.

JMP - R 추가기능 빌더: JSL없이 JMP - R 연결 사용

이 도구를 사용하면 R 함수를 실행하는 사용자 정의 JMP 추가기능을 정의하고 사용할 수 있습니다. 이 인터페이스 빌더를 사용하려면, 먼저 R로 보낼 R 코드를 작성해야 하지만 JSL 코드를 작성할 필요는 전혀 없습니다. 마찬가지로, 새로운 마우스 클릭 인터페이스의 최종 사용자도 R 코드를 사용하거나 보지 않아도 됩니다. 사용하기 간편한 마우스 클릭 인터페이스만 보이게 됩니다(참고: 이 추가기능은 Mac이 아닌 Windows OS에서만 사용됨).

이 추가기능을 사용하면 R과 연결된 JMP 사용에 해당하는 JSL 부분을 건너뛸 수 있습니다. 대신에 화면에 나타나는 몇 가지 대화 상자에서 작업을 통해 추가기능에 포함시키려는 상자와 버튼, 프롬프트를 생성할 것입니다. 대부분의 단계가 여기에 나와 있습니다.



위 예에 나온 SVM 추가기능은 이 JMP - R 추가기능 빌더를 사용하여 생성된 것입니다.

이 JMP - R 추가기능 빌더의 추가기능을 자세히 살펴보고 다운로드하려면 JMP 커뮤니티의 community.jmp.com/t5/JMP-Add-Ins/The-JMP-to-R-Add-In-Builder/ta-p/43879 글을 읽어보십시오.

결론

이 가이드에서는 Python 점수 매기기 코드 생성, JMP에서 Python 스크립트 작성, JMP에서 R 스크립트 작성을 소개했습니다. 또한 JMP의 추가기능 빌더 기능을 사용하는 몇 가지 고급 R 예제를 포함하여 다양한 예제로 연결되는 링크도 안내했습니다. 이러한 주제에 대한 자세한 정보를 보려면 이 백서 전체에서 제공하는 주석과 링크를 참조하십시오. 의문 사항이 있거나 소개된 기법을 실행하는 데 도움이 필요한 경우, 또는 단순히 학습한 내용을 공유하려는 경우, JMP 사용자 커뮤니티(community.jmp.com)에 게시하십시오.

SAS 및 JMP 정보

분석 분야를 선도하는 세계 정상급 기업 SAS에서 1989년에 과학자와 엔지니어, 기타 데이터 분석가들이 데이터를 시각적, 대화식 작업으로 탐구하고 분석할 수 있는 다양한 기능을 제공하는 JMP("점프"로 발음)를 개발했습니다. 이후 JMP는 단일 제품에서, 각각 특정한 요구 사항을 충족하도록 특별히 제작된 다양한 통계적 탐색 도구를 한데 모은 제품군으로 성장했습니다. John Sall SAS 공동창립자 겸 부사장은 JMP 사업부를 이끌고 있습니다.

SAS Institute Inc. 전세계 본사 +1 919 677 8000

JMP는 SAS에서 개발한 소프트웨어 솔루션입니다. SAS에 대해 자세히 알아보려면 **sas.com**을 방문하십시오.
미국과 캐나다 지역 JMP 영업부에는 전화(877 594 6567) 또는 **jmp.com**을 통해 문의할 수 있습니다.