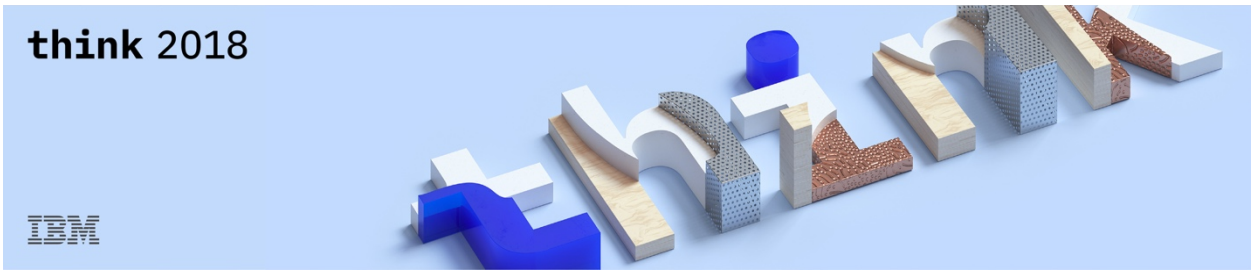


Lab Center – Hands-on Lab

2328

Extending Modeling Capabilities in IBM SPSS Modeler Using R

Jim Mott, Ph.D., IBM, jcmott@us.ibm.com

Jos den Ronden, IBM, jdenronden@nl.ibm.com

Table of Contents

Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results like those stated here.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

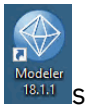
U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Workshop 1: Use R code to create boxplots in IBM SPSS Modeler.

You work for a telecommunications firm and want to use Regression using R code to predict the Total Bill for customers. Before you decide to use the Gender field as a predictor, you want to examine some boxplots to determine if there are any noticeable differences in the Total Bill between males and females. Because Modeler does not have a boxplot node, you will rely on R code instead.

Dataset:	**telco churn data.txt**
Modeler Stream:	**workshop_1_start.str**
Text file:	**code for R programs.txt**
Data folder	**C:\Training\2328**

Task 1. Start IBM SPSS Modeler and set the working folder.

1. Double-click the Modeler 18.1.1 

shortcut on the desktop.

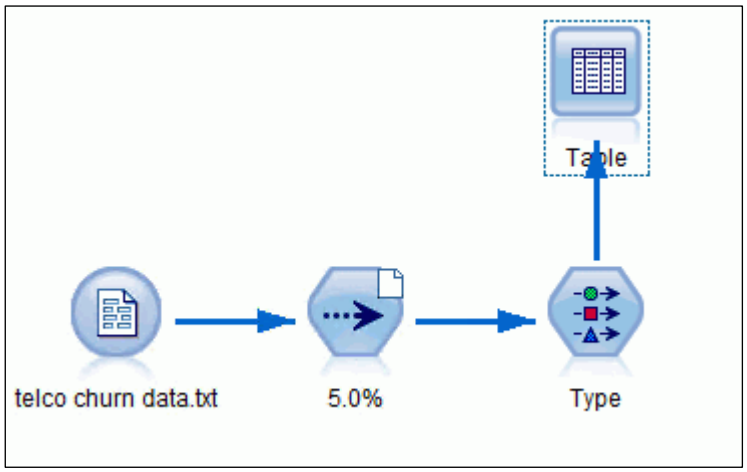
   You will know that it is ready when it presents a Welcome dialog box. You will not make use of the dialog box.

2. Click **Cancel** to close the Welcome dialog box. You will set the working folder.
3. From the **File** menu, click **Set Directory**.
4. Click the **Look In** drop down, browse to **C:\Training\2328**, and then click **Set**.

Task 2. Open the stream and examine the data.

1. From the **File** menu, click **Open Stream**, click **workshop_1_start.str**, located in the **C:\Training\2328** folder, and then click **Open**.

   The results appear as follows:

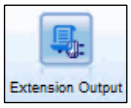


   The stream opens a text data file, samples the records, caches the data, instantiates the data in a Type node, and then requests a Table.

2. Right-click the **Table** node, and then select **Run**. The data contains 9 fields, and 903 records. The data is cached at the Sample node.
3. Close the **Table** output node.

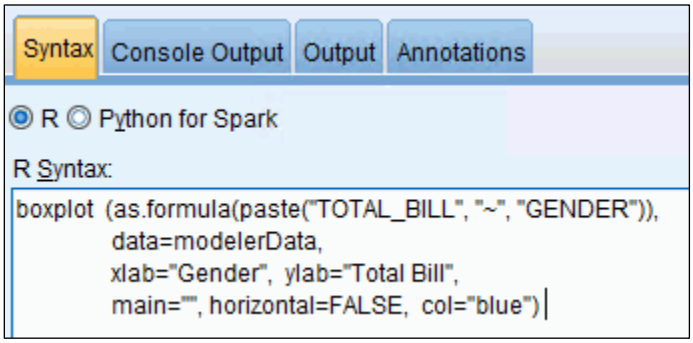
Task 3. Add and configure an Extension Output node for an R boxplot.

You will use an Extension Output node and examine the R code in this task. You will get the R code from a text file.

1. Switch to **Windows Explorer**, browse to **C:\Training\2328**, and open **code for R programs.txt** in Notepad.
2. Copy the four lines below **# CODE FOR R BOXPLOT**.
3. Close the text file.
4. Switch to **Modeler**.
5. From the **Output** palette, add an **Extension Output** node 

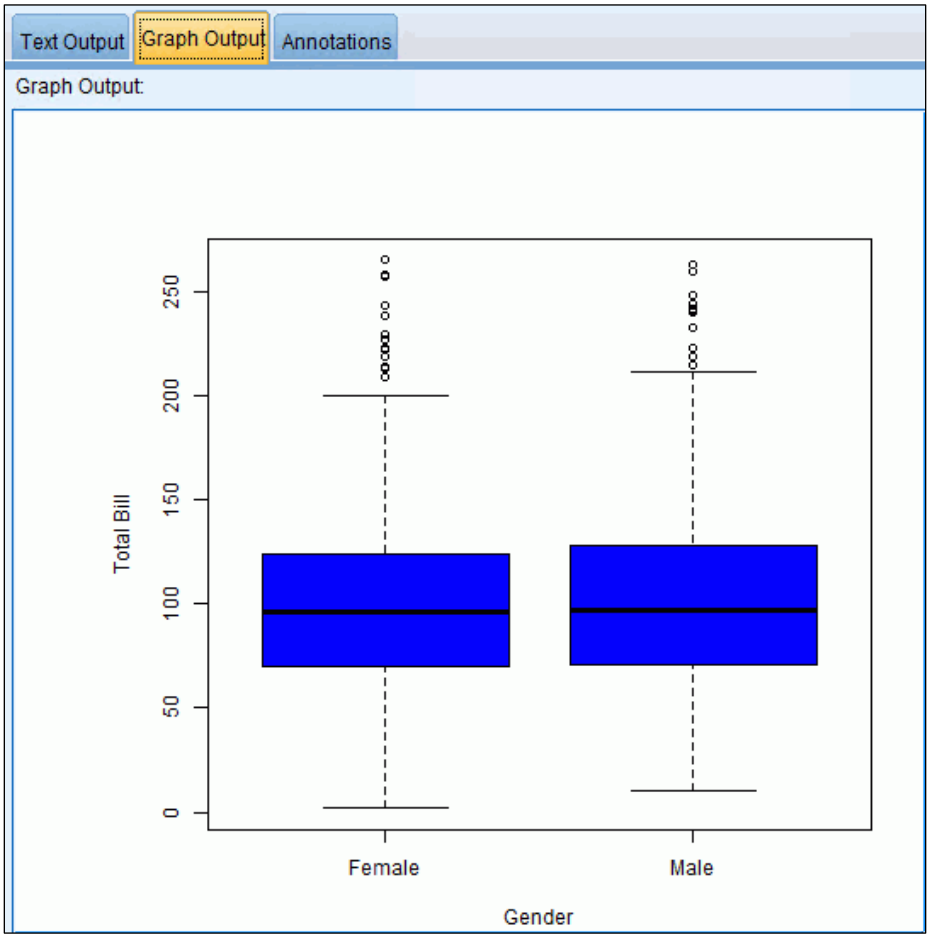
downstream from the **Type** node.
6. Edit the **Extension Output** node, and then:
   - ensure that the **Syntax** tab is selected
   - ensure that **R** is selected as the syntax language
   - paste the code you copied into the **R syntax** box

   The results appear as follows:



7. Click **Run.**

8. Click the **Graph** tab.
   The results appear as follows:



   The boxplot is generated. Males and females do not seem to differ when it comes to their bill. As a result, you will not use it as a predictor of Total_Bill in your regression analysis.

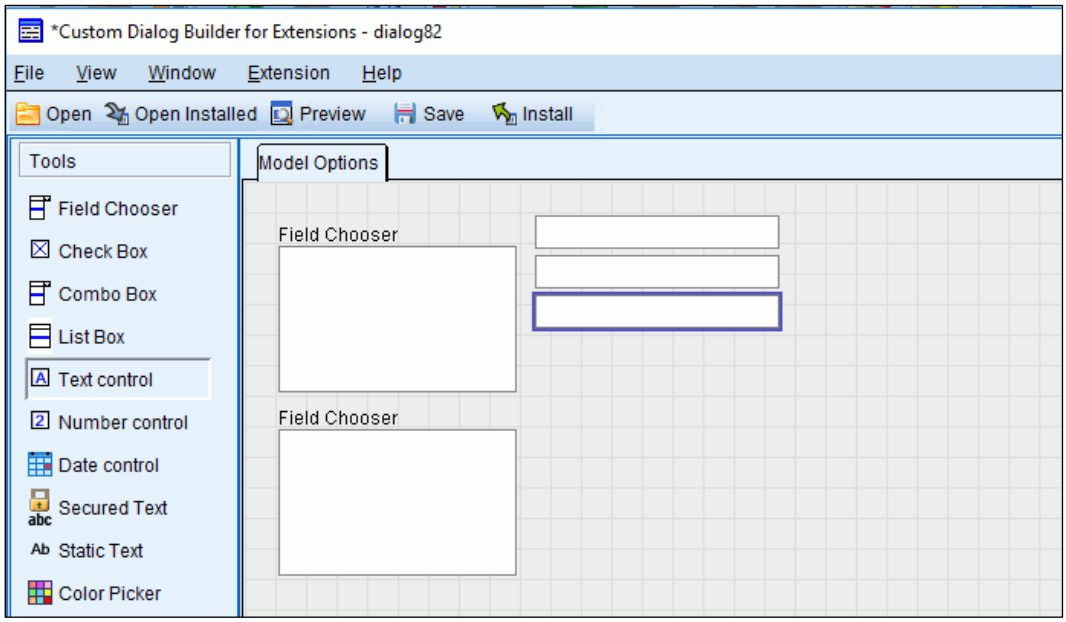
9. Close the **R Output** window.

Task 4. Create a dialog box for an R boxplot.

The boxplot code run in Task 2 is not very flexible because the fields were already specified in the code. In this task, you will use the Custom Node Dialog Builder to create your own interface for the boxplot.

1. From the **Extensions** menu, click **Custom Node Dialog Builder**.
2. Ensure that the **Tools Palette** is available. If not, select **View** and click **Tools Palette**.
3. From the **Tools Palette**, put two **Field Chooser** items on the dialog canvas, the second below the first.
4. From the **Tools Palette**, put three **Text control** items on the dialog canvas, to the right of the **Field Chooser** items.

   The results appear as follows:



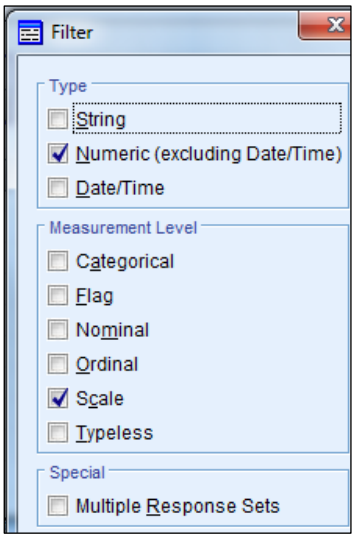
You will replace the text you have now, and add more text to it. Also, you will change names to better identify the items.

You will start with the first Field Chooser control. This control represents the numeric, continuous field that you want to request the boxplot for.

5. Click on the upper **Field Chooser** item to display its properties, and then make the following changes:

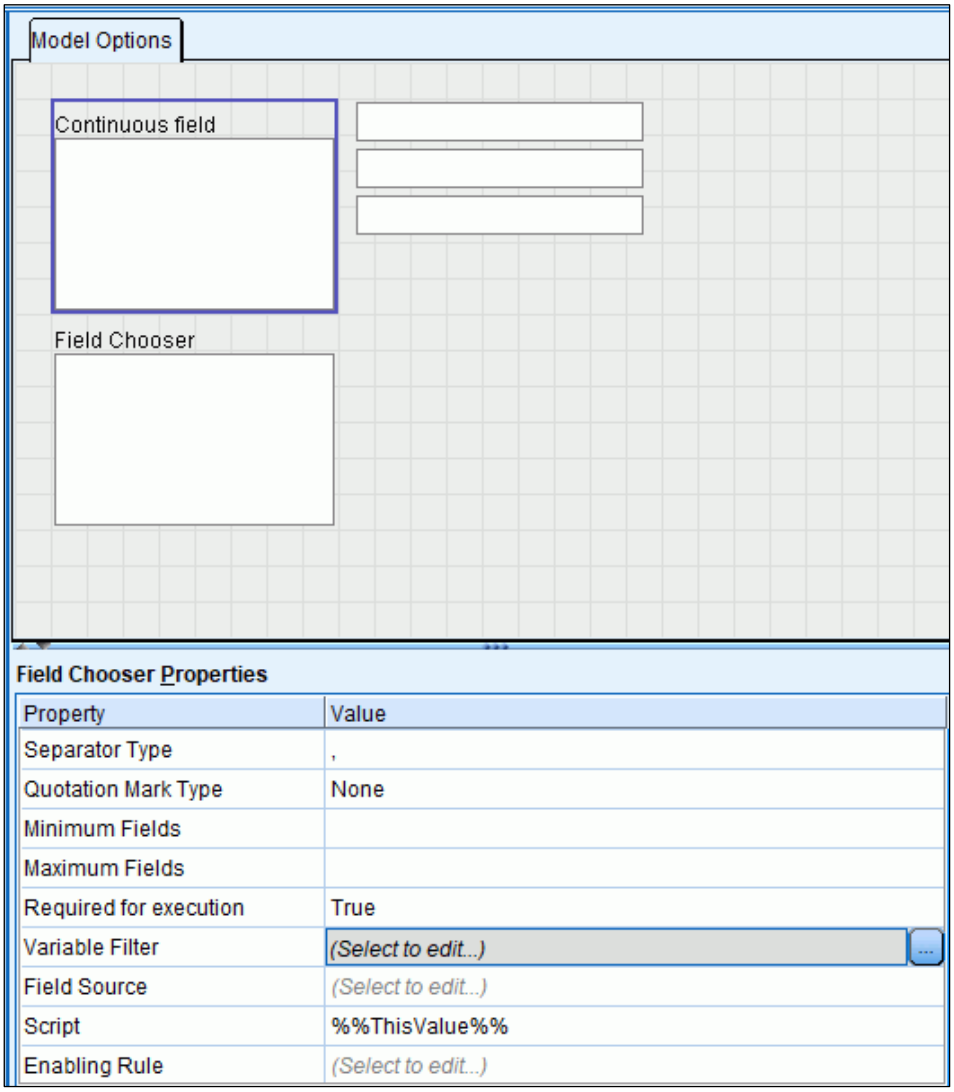
   - for **Identifier**, replace the current value by **contfield**
   - for **Title**, replace **Field Chooser** by **Continuous field**
   - for **Required for execution**, replace **False** by **True**
   - for **Variable Filter**, click in the cell ***(Select to edit...)***, and then click the **ellipses** 
   - for **Filter**, ensure that only the options **Type - Numeric** and **Measurement Level - Scale** are enabled

     The results appear as follows:

   - click **OK** close the **Filter** dialog box
   - scroll to the bottom of the **Field Chooser Properties**

The results appear as follows:



The Script property specifies the syntax to be generated by this control at run time. It has a default value of %%ThisValue%%. This specifies that the syntax generated by the control will consist of the run-time value of the control, which will be one of the continuous fields in your dataset. You will keep this default.

6. Repeat step **5** for the **second Field Chooser** control, the field that defines the groups. Change its properties to:

   - for **Identifier**, replace the current value by **catfield**
   - for **Title**, replace **Field Chooser** by **Categorical field**
   - for **Required for execution**, replace **False** by **True**
   - for **Variable Filter**, for **Type** only enable **Type String** and **Numeric,** and for **Measurement Level** only enable **Categorical**, **Flag**, **Nominal** and **Ordinal**
   - click **OK** to close the **Filter** dialog box

   You will also change the properties for the text controls. The upper two controls set the labels for the axes, the bottom control determines the color.

7. Click the upper **Text control** item, and then:

   - for **Identifier**, replace the current value by **contfieldlabel**
   - for **Title**, type **Axis label**

8. Click the upper **Text control** item, and then:

   - for **Identifier**, replace the current value by **catfieldlabel**
   - for **Title**, type **Axis label**

9. Click the bottom **Text control** item, and then:

   - for **Identifier**, replace the current value by **color**
   - for **Title**, type **Color**
   - for **Default Value**, type **blue**

   You will also change the dialog's properties.

10. Click anywhere in the grey work area to set focus on the dialog box itself, and then in the properties section:

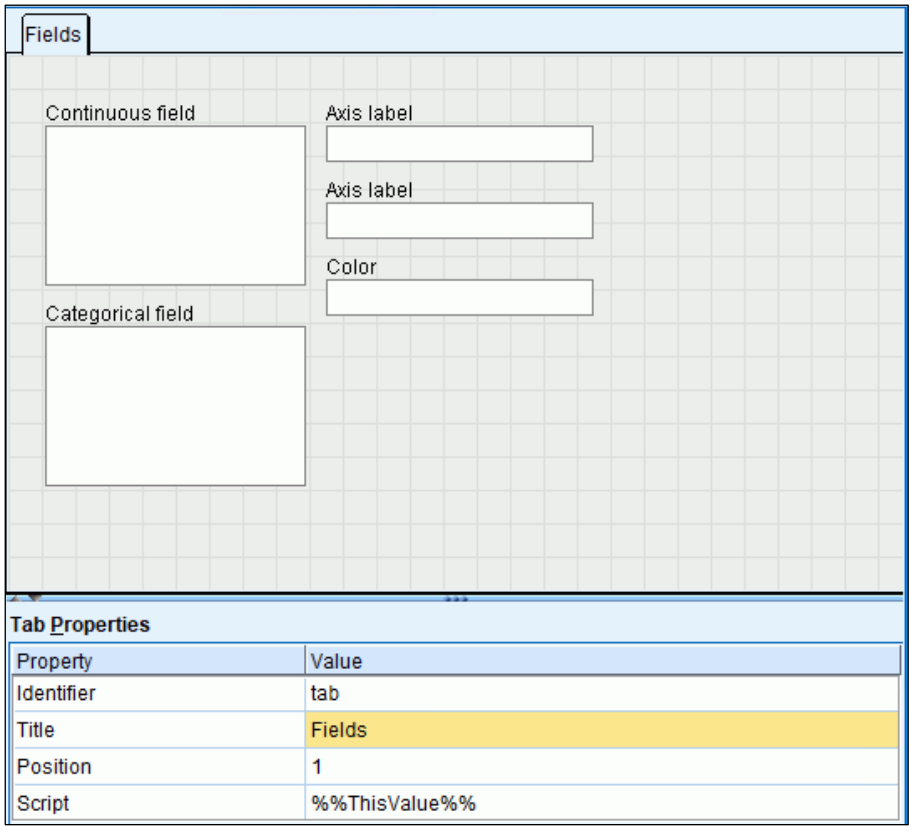
    - for **Dialog Name**, replace the current text by **RBoxplot**
    - for **Title**, type **R_Boxplot** (spaces are not allowed in the title!)

    The dialog's properties also set the type of node (Import, Model Process, Output, Export), and determine in which palette the node will appear. The boxplot node that you are designing will produce a boxplot, which is a piece of output. Therefore you will choose Output node.

11. With the dialog still selected, in the **Properties** area, scroll to **Modeler Properties**:
    - set **Node Type** to **Output**
    - for **Palette**, select **Output**

    Finally, you will change the caption of the tab.
12. Click the tab that reads **Model Options**, and then:
    - for **Identifier**, type **tab**
    - for **Title**, replace **Model Options** by **Fields**

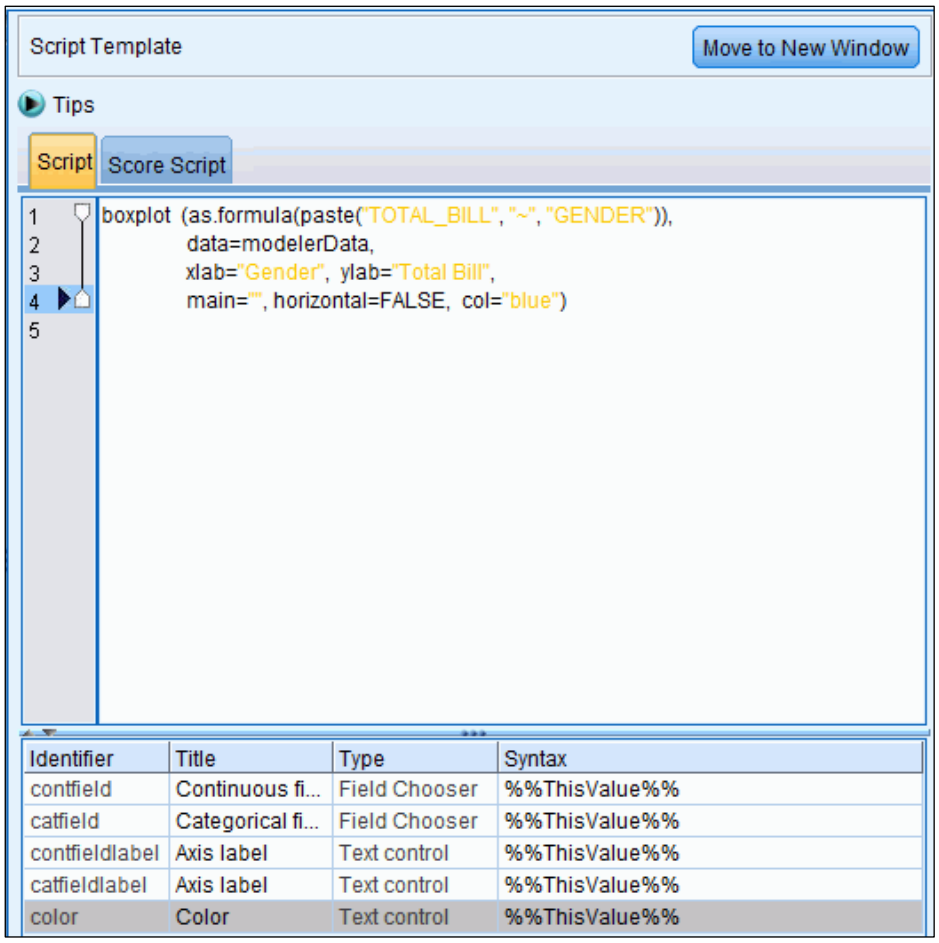
    The results appear as follows:



    You will attach the R code that must be executed when the node is executed. As before, you will get the R code from the help file.
13. Switch to **code for R programs.txt**. (This file should still be open in Notepad, if not, browse to **C:\Train\2328**, and open **code for R programs.txt** in Notepad.)
14. Copy the same four lines below **# CODE FOR R BOXPLOT** (do not close the text file).
15. Switch to **Modeler**.

16. In the S**cript template** box, paste the copied code. Ensure the Script tab is selected.

    The results appear as follows:

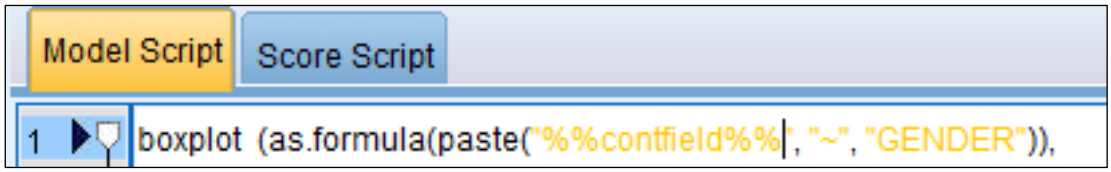


    This code will always return a boxplot of TOTAL_BILL by GENDER, in blue, with axis labels "Total Bill" and "Gender". You will need to replace these fixed values by the selections made in the dialog box. You can do this by using the identifiers as placeholders.

17. Use your cursor to highlight the string **TOTAL_BILL**
18. Click the **Delete** key. Be sure to leave the cursor between the two double-quote marks.

19. Press **<CTRL>** + **<SPACEBAR>**, and pick the control identifier **%%contfield%%** from the list.

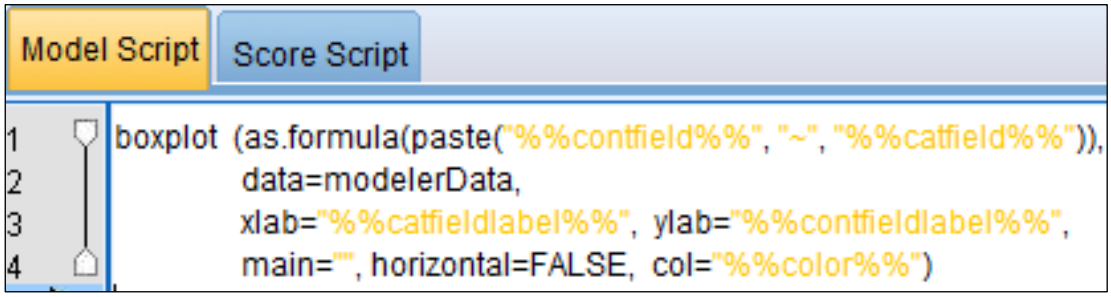
    The results appear as follows:



20. Repeat steps **18** to **20**, to:

    - replace **GENDER** by **%%catfield%%**
    - for **xlab**, replace **Gender** by **%%catfieldlabel%%**
    - for **ylab**, replace **Total Bill** by **%%contfieldlabel%%**
    - replace **blue** by **%%color%%**

    The results appear as follows:



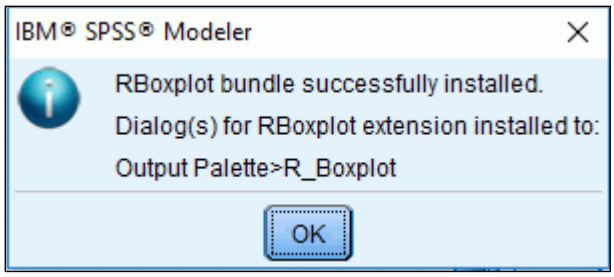
    As the dialog box will be an Output node (and not a Modeling node), the Score Script tab is irrelevant. You could click the Score Script tab, but you cannot enter code there.

    Now that you have entered the code, you will save and install the dialog box.

21. From the **File** menu, click **Install**.
22. In the **Name** box, type **RBoxplot**
23. In the **Summary** box, type **Create boxplots with R code**

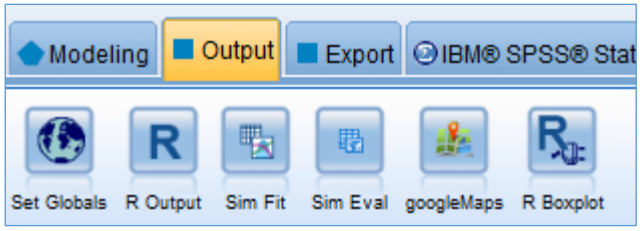
24. Click **Continue**.

    You are notified that the dialog box is installed:



25. Click **OK**.
26. From the **File** menu, click **Save**, and name the file **RBoxplot.mpe** (mpe is the extension for extension bundles). Note: you can save the file to the **Documents** folder.
27. From the **File** menu, click **Close** to close the **Custom Dialog Builder**.
28. Examine the **Output** palette.

    The results appear similar to the following:



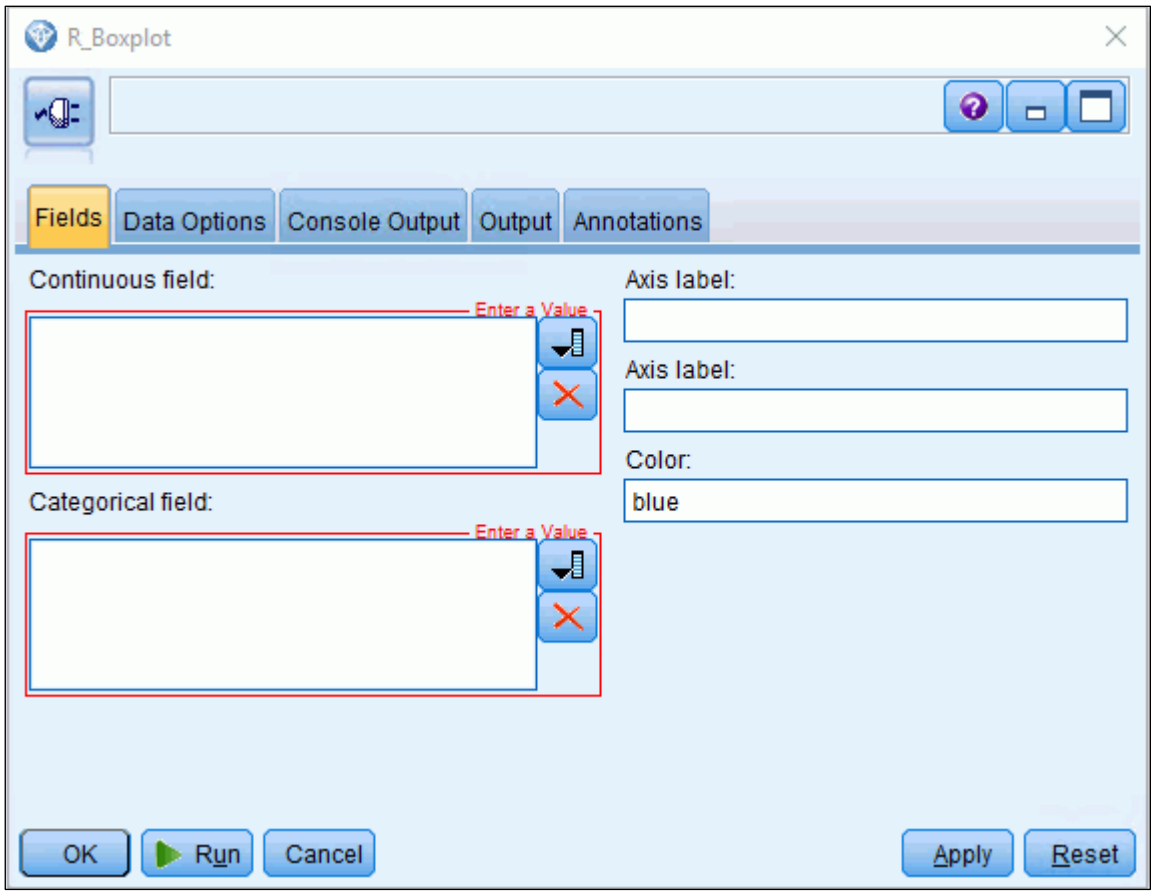
    The R Boxplot on the right is the custom node, just created in the Custom Dialog Builder.

    You will try out the new node.

29. Add the **R Boxplot** node downstream from the **Type** node.

30. Edit the **R Boxplot** node.

The results appear as follows:

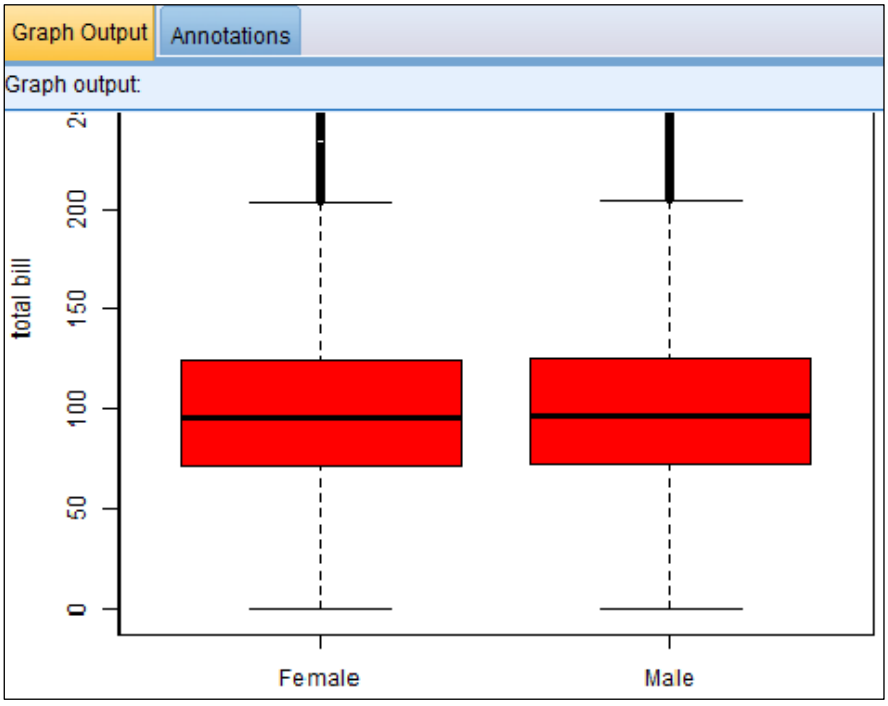


The first tab is entitled Fields. The other tabs, as well as the buttons, are automatically added.

Notice that it is required to select a continuous field and categorical field.

- for **Continuous field**, select **TOTAL_BILL** (notice that only continuous fields are presented, thanks to the filter)
- for **Categorical field**, select **GENDER**
- for the first **Axis label**, type **total bill**
- for the second **Axis label**, type **gender**
- for **color**, replace **blue** by **red**
- click **Run**

The results appear as follows:



31. Close the output window, and click **No** when asked to save it.

    From now on you will have the R Boxplot node in the Output palette. You can share the custom dialog box with colleagues by copying the file RBoxplot.mpe to the C:\ProgramData\IBM\SPSS\Modeler\18.1\CDB folder on their machines (assuming that you and your colleagues work in a client-only Windows 10 environment). Note that if the ProgramData folder is hidden, you will have to unhide it to perform this operation.

    You will create a clean slate for the next task.

32. From the **File** menu, click **Close Stream** without saving the changes.
33. From the **File** menu, click **New Stream**.

    Do not exit IBM SPSS Modeler. Leave it open for the next workshop.

Workshop 2: Use R code to add a new fields to Modeler data.

You work for a telecommunications firm and want to analyze the relationship between the continuous field TOTAL_BILL and several other categorical fields such as GENDER and CHURN, but first you want to create a categorical version of the field based on the decile rank for how much each customer is billed. This will mean adding a new column to the Modeler data.

Dataset: **telco churn data.txt**

Modeler Stream: **workshop_2_start.str**

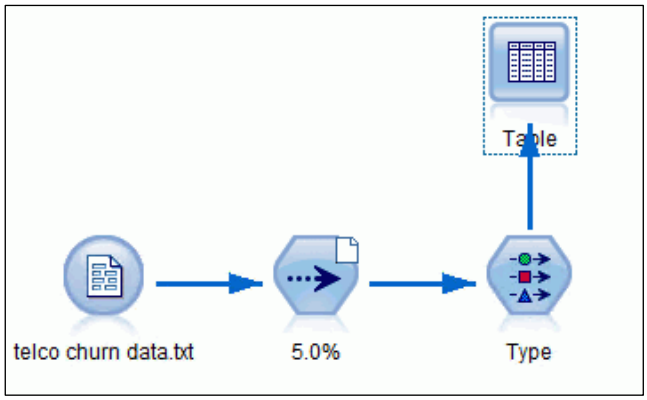
Text file: **code for R programs.txt**

Data folder **C:\Training\2328**

Task 1. Open the Modeler stream and examine the data.

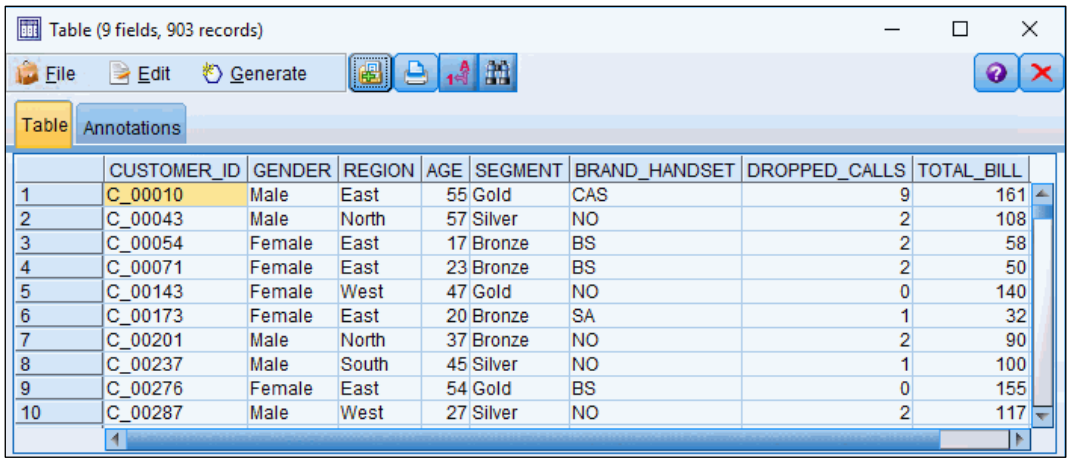
1. From the **File** menu, click **Open Stream**, click **workshop_2_start.str**, located in the **C:\2328**, and then click **Open**.

   The results appear as follows:



2. Run the **Table** node.

   The results appear as follows:



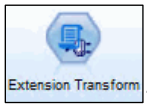
	CUSTOMER_ID	GENDER	REGION	AGE	SEGMENT	BRAND_HANDSET	DROPPED_CALLS	TOTAL_BILL
1	C_00010	Male	East	55	Gold	CAS	9	161
2	C_00043	Male	North	57	Silver	NO	2	108
3	C_00054	Female	East	17	Bronze	BS	2	58
4	C_00071	Female	East	23	Bronze	BS	2	50
5	C_00143	Female	West	47	Gold	NO	0	140
6	C_00173	Female	East	20	Bronze	SA	1	32
7	C_00201	Male	North	37	Bronze	NO	2	90
8	C_00237	Male	South	45	Silver	NO	1	100
9	C_00276	Female	East	54	Gold	BS	0	155
10	C_00287	Male	West	27	Silver	NO	2	117

   The data has 9 fields and 903 records. The field you are interested in, TOTAL_BILL, is on the right. This is the field you want to group into decile ranks.

3. Close the **Table** node.

Task 2. Add a new field based on decile rank of total amount billed.

You will need to use an Extension Transform node to add the new field to the data.

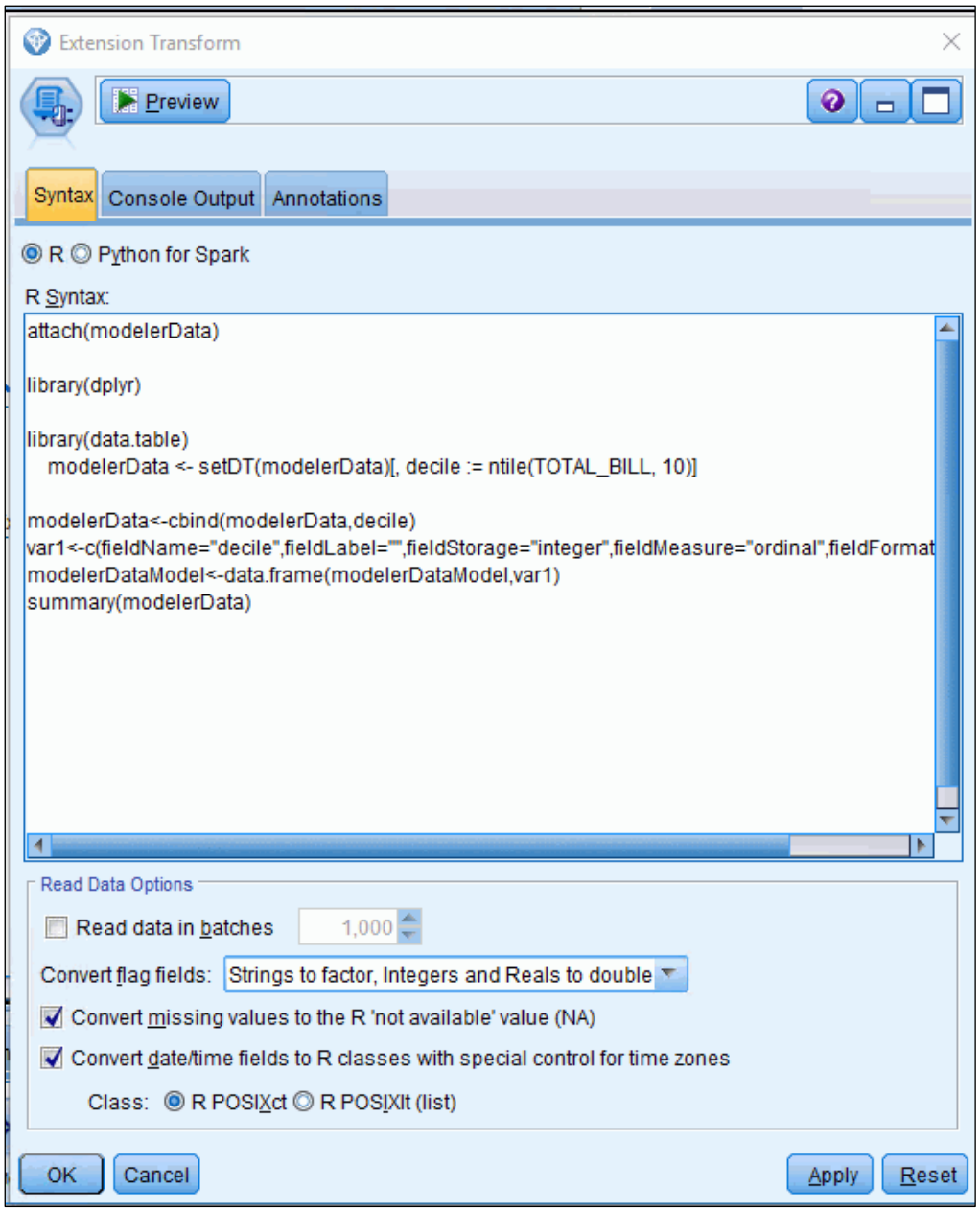
1. Switch to **Windows Explorer**, browse to **C:\Training\2328**, and open **code for R programs.txt** in Notepad.
2. Copy the six lines below **# R CODE FOR ADDING A DECILE FIELD TO THE DATA.**
3. Switch to **Modeler**.
4. From the **Records** palette, attach an **Extension Transform** node 

to the **Type** node.

5. Edit the **Extension Transform** node and then:

   - ensure that **R** is selected as the syntax language
   - paste the code you copied into the **R syntax** box

   The results appear as follows:



Line # 1 attaches the database to the R search path

Line # 2 loads the functions available in the dplyr package

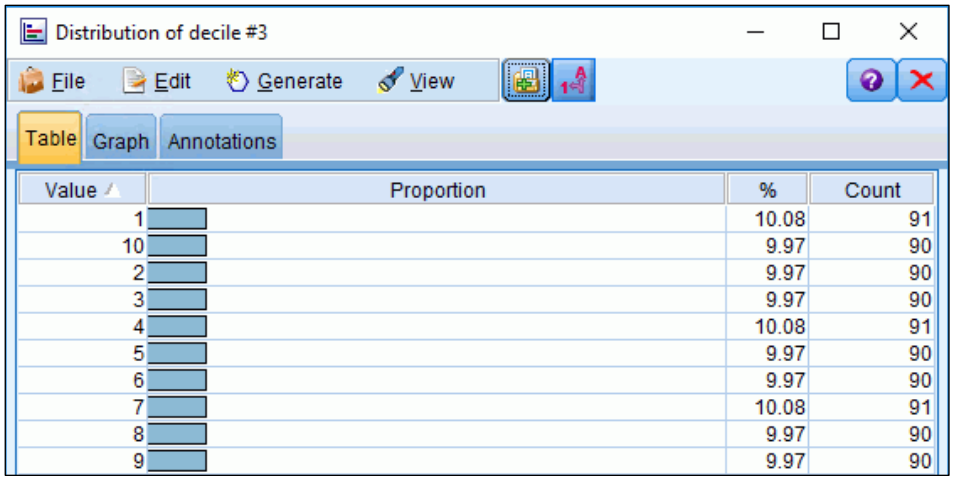
Line # 3 uses the ntile function in the dplyr package to assign decile values to TOTAL_BILL

Lines # 4 & 5 takes care of the field storing the deciles

Line # 6 returns the R data frame back to Modeler

6. Close the **Extension Transform** dialog box.
7. From the **Graph** palette, add a **Distribution** node to the **Extension Transform** node.
8. Edit the **Distribution** node and in the **Field** box, select **decile**.
9. Click **Run**.

The results are as follows:



Value	Proportion	%	Count
1		10.08	91
10		9.97	90
2		9.97	90
3		9.97	90
4		10.08	91
5		9.97	90
6		9.97	90
7		10.08	91
8		9.97	90
9		9.97	90

The values of TOTAL_BILL have been successfully ranked into deciles and stored in the new field, decile.

10. Close the Distribution node output.

You will create a clean slate for the next workshop.

11. From the **File** menu, click **Close Stream** without saving the changes.
12. From the **File** menu, click **New Stream**.

Do not exit IBM SPSS Modeler. Leave it open for the next workshop.

Workshop 3. Add and configure an extension model node to run R regression.

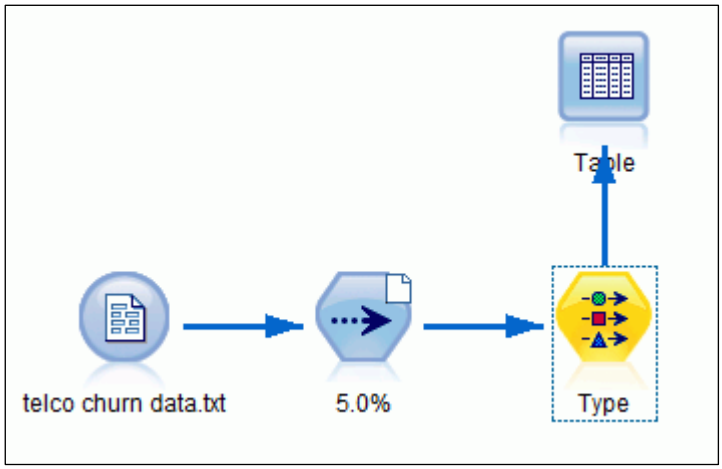
You work for a telecommunications firm and want to use regression using R code to predict the Total Bill for customers. One of your colleagues already created an extension bundle called RLinear.cfe which runs regression and sent it to you. You will install the bundle run your analyses.

Dataset:	**telco churn data.txt**
Modeler Stream:	**workshop_3_start.str**
Text file:	**code for R programs.txt**
Data folder	**C:\Training\2328**

Task 1. Open the Modeler stream and examine the data.

1. From the **File** menu, click **Open Stream**, click **workshop_3_start.str**, located in the **C:\2328**, and then click **Open**.

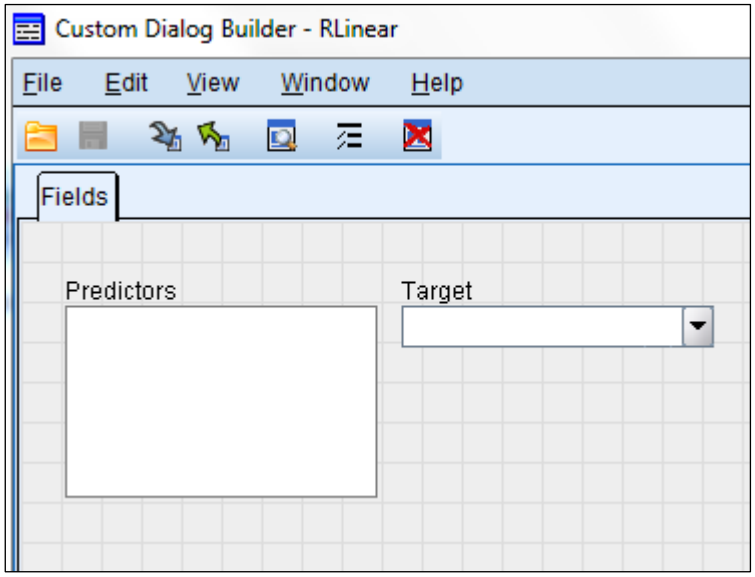
   The results appear as follows:



Task 2. Install the RLinear bundle.

1. From the **Extensions** menu, click **Custom Node Dialog Builder.**
2. From the **File** menu, click **Open**, click **RLinear.mpe**, located in the **C:\Training\2328** folder, and then click **Open**.

   The results appear as follows:



   You can specify the predictors and target.

3. Click on the **work area** to set focus on the dialog box itself, scroll to the **Dialog Properties** area, and then examine the **Modeler Properties**.

   The results appear as follows:

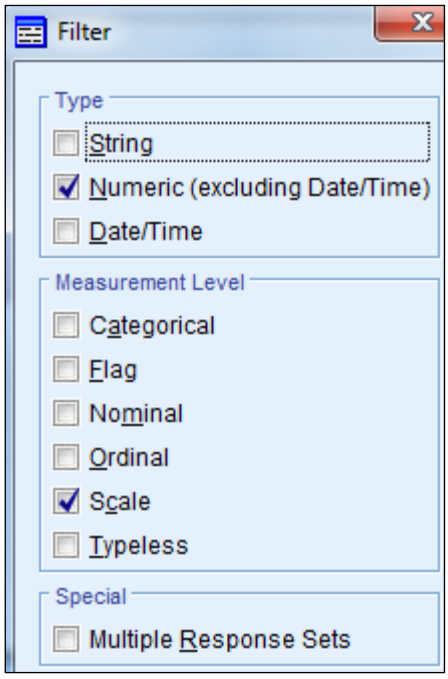
Modeler Properties	
Node Type	Model
Palette	Modeling (Classification)

   The Node Type is Model, which means that the node will generate a model nugget and has the capability to add predictions to the dataset.

   The node will be located in the Modeling palette, Classification subpalette.

4. Click the **Predictors** item to set focus on it, and in the **Field Chooser Properties** area, click in the **Variable Filter** cell, and then click the ellipses.

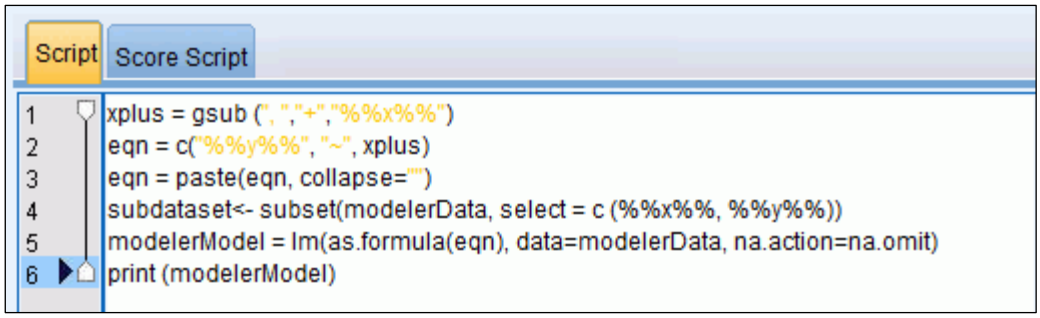
   The results are as follows:



   Only numeric fields of Scale measurement will appear in the list of predictors.

5. Click **Cancel** to close the **Filter** dialog box.
6. In the **Script Template**, click the **Script** tab.

   The results appear as follows:



   Lines #2 & 3 adds the dependent field and combines the dependent and independent fields into a string.

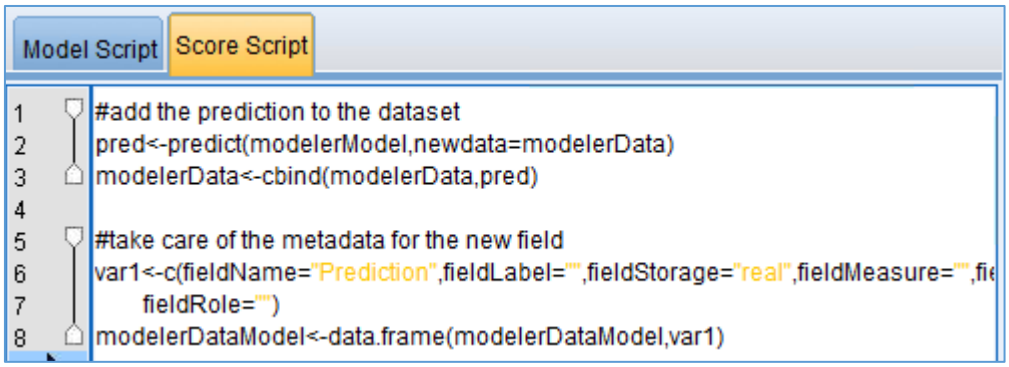
   Line #4 gets a selection of the data from Modeler.

   Line #5 runs the model and stores the results in a field named ModelerModel.

   Line #6 prints the results

7. Click the **Score Script** tab.

   The results appear as follows:



   Lines #2 through #7 take care of the field storing the predictions.

   Line #8 returns the R data frame back to Modeler.

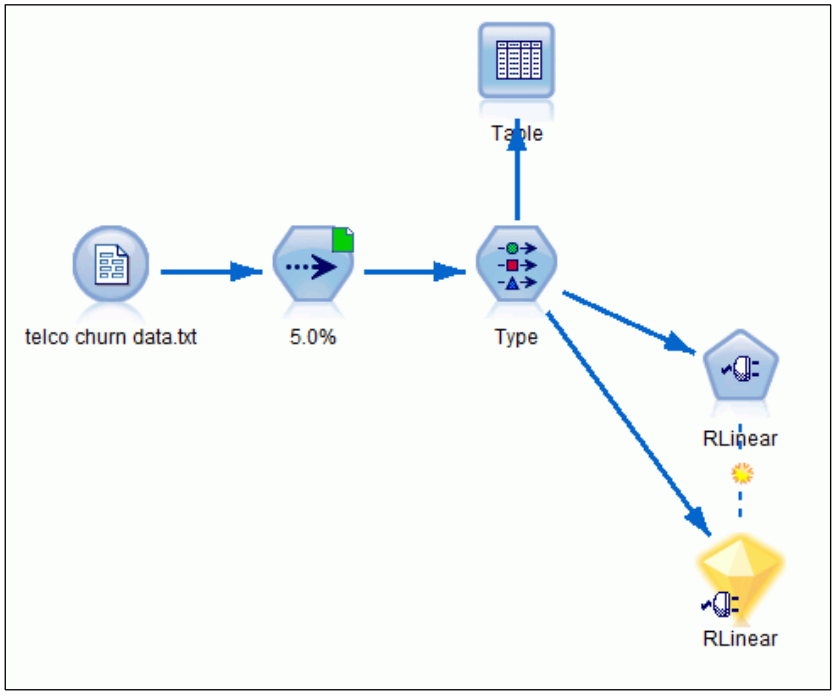
8. From the **File** menu, click **Install**.
9. Click **OK**.
10. Close the **Custom Dialog Builder** window.

11. From the **Modeling** palette, **Supervised** subpalette, add the **RLinear** 

node downstream from the **Type** node, and then:

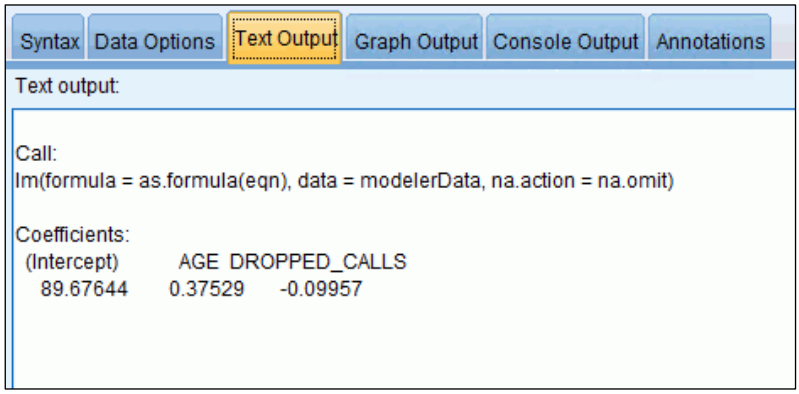
    - Edit the **RLinear** node. For **Predictors**, select **AGE**, **DROPPED_CALLS**
    - for **Target**, select **TOTAL_BILL**
    - click **Run**

    The results appear as follows:



12. Edit the **R Linear model nugget**, and then click the **Text Output** tab.

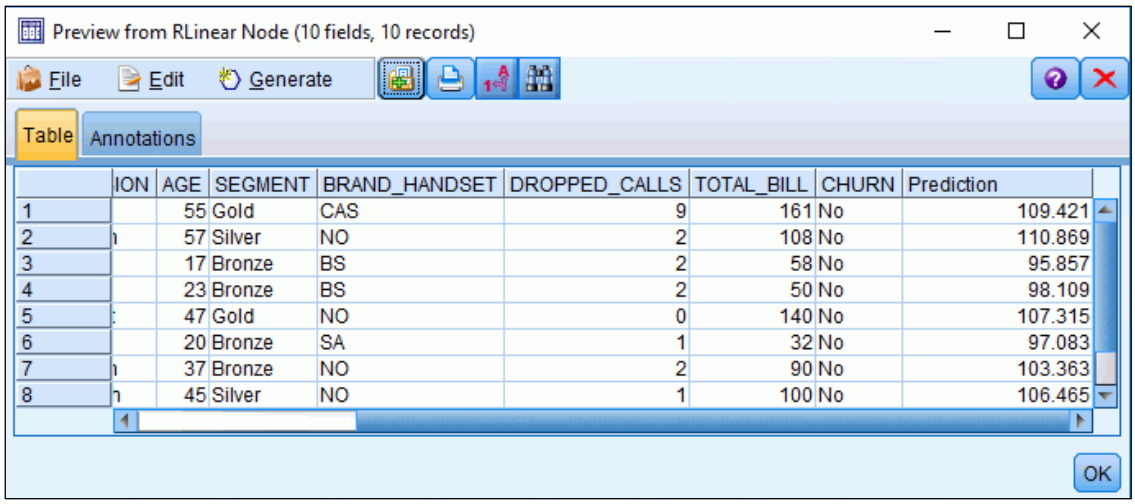
    The results are as follows:



    This piece of output gives the regression coefficients.

13. Click **Preview** and scroll to the last field.

    The results are as follows:



	ION	AGE	SEGMENT	BRAND_HANDSET	DROPPED_CALLS	TOTAL_BILL	CHURN	Prediction
1		55	Gold	CAS	9	161	No	109.421
2	h	57	Silver	NO	2	108	No	110.869
3		17	Bronze	BS	2	58	No	95.857
4		23	Bronze	BS	2	50	No	98.109
5	:	47	Gold	NO	0	140	No	107.315
6		20	Bronze	SA	1	32	No	97.083
7	h	37	Bronze	NO	2	90	No	103.363
8	h	45	Silver	NO	1	100	No	106.465

    The Prediction field is added by the model R Linear model nugget and stores the prediction from that model.

    You will create a clean slate for the next workshop.

14. From the **File** menu, click **Close Stream** without saving the changes.
15. From the **File** menu, click **New Stream**.

    Do not exit IBM SPSS Modeler. Leave it open for the next task.

Workshop 4. Use an R package to perform geospatial analysis.

A taxi company wants to understand demand on New Year's Eve. It has geospatial and time data for phones from people who opted in to their smart-taxi phone application. The data were collected in Central London, as well as Islington and King's Cross. The company wants to use Modeler together with R and Google Maps node to determine if there are enough taxis at certain locations and times to meet demand.

Datasets: **PhoneLocationData.csv**

TaxiLocationData.csv

Modeler Stream **workshop_4_start.str**

Data folder **C:\Training\2328**

Task 1. Open the Modeler stream and examine the data.

You will open and examine a stream that runs an R googleMaps node.

1. Open **workshop_4_start.str**.
2. Run the node named **TABLE 1**.

   The results appear as follows:

	TimeStamp	Latitude	Longitude	Phone_App_User_ID
1	2013-12-31 07:57:00	51.521	-0.114	40054
2	2013-12-31 08:09:00	51.520	-0.127	40054
3	2013-12-31 08:09:00	51.521	-0.136	40054
4	2013-12-31 08:15:00	51.521	-0.135	40054
5	2013-12-31 08:20:00	51.521	-0.128	40054
6	2013-12-31 08:39:00	51.521	-0.137	40054
7	2013-12-31 09:09:00	51.521	-0.136	40054
8	2013-12-31 09:12:00	51.521	-0.137	40054
9	2013-12-31 09:39:00	51.521	-0.136	40054
10	2013-12-31 10:09:00	51.520	-0.135	40054

   This data represents the location and time of customers who used their smart-taxi phone application.

   Modeler can encode the coordinates and time into a so-called Space-Time-box. For the Space component, Modeler reworks the coordinates to a so-called geohash, which represents a certain geographical area. For the Time component, Modeler reworks the exact time to a time period. You can specify the window for location (the size of the geographical area) and the window for time (the time period) in the Space-Time-Box node.

3. Close the **Table** output window.

   The upper part of the stream uses the Space-Time-Boxes node and aggregates the data. An approximately 1 square mile window for location, and a 15 minutes window was used for time

4. Run the node named **TABLE 2**.

   The results appear as follows:

	People_Density	STB_GH5_15MINS
1	9	gcpvj|2013-12-31 07:45:00|2013-12-31 08:00:00
2	12	gcpvj|2013-12-31 08:00:00|2013-12-31 08:15:00
3	1	gcpvh|2013-12-31 08:00:00|2013-12-31 08:15:00
4	1	gcpvh|2013-12-31 08:15:00|2013-12-31 08:30:00
5	2	gcpvh|2013-12-31 10:15:00|2013-12-31 10:30:00
6	3	gcpvh|2013-12-31 10:30:00|2013-12-31 10:45:00
7	4	gcpvh|2013-12-31 10:45:00|2013-12-31 11:00:00
8	5	gcpvh|2013-12-31 11:00:00|2013-12-31 11:15:00
9	6	gcpvh|2013-12-31 11:15:00|2013-12-31 11:30:00
10	5	gcpvh|2013-12-31 11:30:00|2013-12-31 11:45:00

   Each record gives the number of people in a certain area (the geohash) at a certain time. For example, there were nine people who opted in to the smart taxi phone app at geohash gcpvj (a particular area in London), between 07:45^{h} and 08:00^{h} (the first record).

5. Close the **Table** output window.
6. Run the node named **TABLE 3**.

   The results appear as follows:

	TimeStamp	Latitude	Longitude	Taxi_Number
1	2013-12-31 00:00:00	51.520	-0.115	40056
2	2013-12-31 00:11:00	51.520	-0.114	40056
3	2013-12-31 00:41:00	51.520	-0.114	40056
4	2013-12-31 01:11:00	51.520	-0.114	40056
5	2013-12-31 01:41:00	51.520	-0.115	40056
6	2013-12-31 02:11:00	51.520	-0.114	40056
7	2013-12-31 02:41:00	51.520	-0.115	40056
8	2013-12-31 03:11:00	51.520	-0.115	40056
9	2013-12-31 03:41:00	51.520	-0.114	40056
10	2013-12-31 04:11:00	51.520	-0.115	40056

   This data gives the location and time of taxis.

   This data is also aggregated to time-space boxes, using the same window for time (15 minutes) and space (1 square mile).

7. Close the **Table** output window.

8. Run the node named **TABLE 4**.

   The results appear as follows:

	Taxi_Density	STB_GH5_15MINS
1	18	gcpvj|2013-12-31 00:00:00|2013-12-31 00:15:00
2	15	gcpvj|2013-12-31 08:30:00|2013-12-31 08:45:00
3	15	gcpvj|2013-12-31 08:45:00|2013-12-31 09:00:00
4	19	gcpvj|2013-12-31 09:00:00|2013-12-31 09:15:00
5	20	gcpvj|2013-12-31 09:15:00|2013-12-31 09:30:00

   The data provides the details of the availability of taxis, in a certain area, at a certain time.

9. Close the **Table** output window.

   The two datasets are merged, and a field is derived that gives the Taxi/People ratio.

10. Run the node named **TABLE 5**.

    The results appear as follows:

	STB_GH5_15MINS	Taxi_Density	People_Density	Taxi_To_People_Ratio
1	gbgjw|2013-12-31 16:45:00|2013-12-31 17:00:00	1	1	1.000
2	gbgjw|2013-12-31 17:00:00|2013-12-31 17:15:00	1	1	1.000
3	gbgjw|2013-12-31 17:15:00|2013-12-31 17:30:00	1	2	0.500
4	gbgjw|2013-12-31 17:30:00|2013-12-31 17:45:00	1	1	1.000
5	gbgjw|2013-12-31 18:00:00|2013-12-31 18:15:00	1	1	1.000

    For each area and timeframe it is known whether there are enough taxis to meet demand. For example, there is a shortage in geohash gbgjw, between 17:15^{h} and 17:30 ^{h} (the third record).

11. Close the **Table** output window.

    The 5% of Space-Time-Boxes with the lowest taxi to people ratios have been selected because those are the locations that the taxi company needs to focus on the most. Also the geohash area are computed.

12. Run the node named **TABLE 6**.

    The results appear similar to the following:

Taxi_To_People_Ratio	Taxi_To_People_Ratio_TILE20	TAXIS_NEEDED_HERE_Latitude	TAXIS_NEEDED_HERE_Longitude
0.200	1	51.526	-0.154
0.125	1	51.526	-0.110
0.167	1	51.526	-0.110
0.160	1	51.526	-0.110
0.192	1	51.526	-0.110
0.160	1	51.526	-0.110
0.143	1	51.526	-0.110
0.217	1	51.526	-0.110
0.235	1	51.526	-0.110
0.190	1	51.526	-0.110

    It appears that nine of the top 10 space-time-boxes with highest shortage are at the same location.

13. Close the **Table** output window.

Task 2. Install the GoogleMaps extension.

The GoogleMaps node, an extension that can be downloaded from Extension Hub, enables you to view the location on the map. You will not actually download the extension during this workshop, because it is already installed. (Note that you must have an internet connection to install extensions).

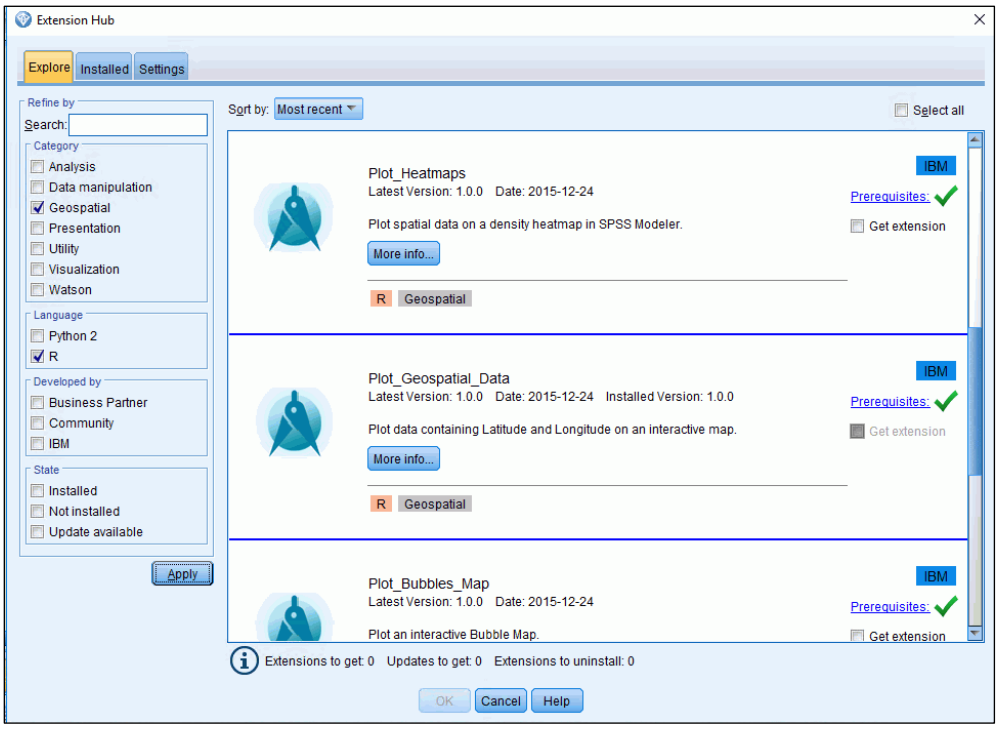
1. From the **Extensions** menu, click **Extension Hub**.

   From the menu on the left, check

   - **Geospatial**
   - **R**

2. Click **Apply** to list all the R extensions pertaining to Geospatial analysis.

   The results are as follows:

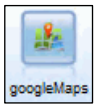


   The Plot_Geospatial_Data extension is already installed. Notice on the right that the Get Extension option under Prerequisites is disabled. If it was not, you would check the Get extension box and then click OK to download and install the extension. For example, the Get Extension option is enabled for the Plot_Heatmaps extension.

   Because the Plot_Geospatial_Data extension is already installed, you will click cancel.

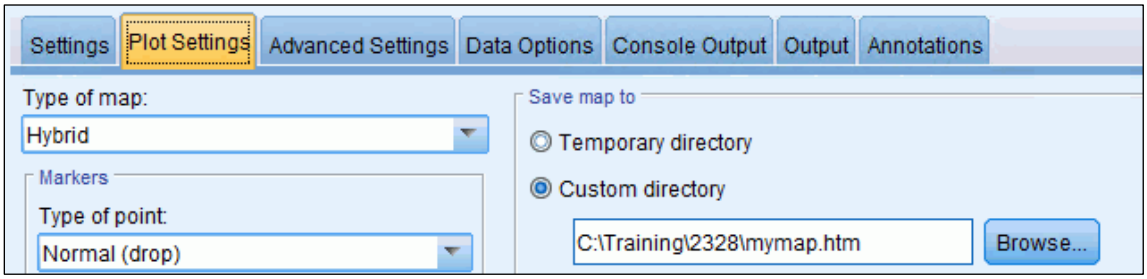
3. Click **Cancel**.

Task 3. Use the GoogleMaps node to identify locations where more taxis are needed.

1. From the **Output** palette, attach a **googleMaps** 

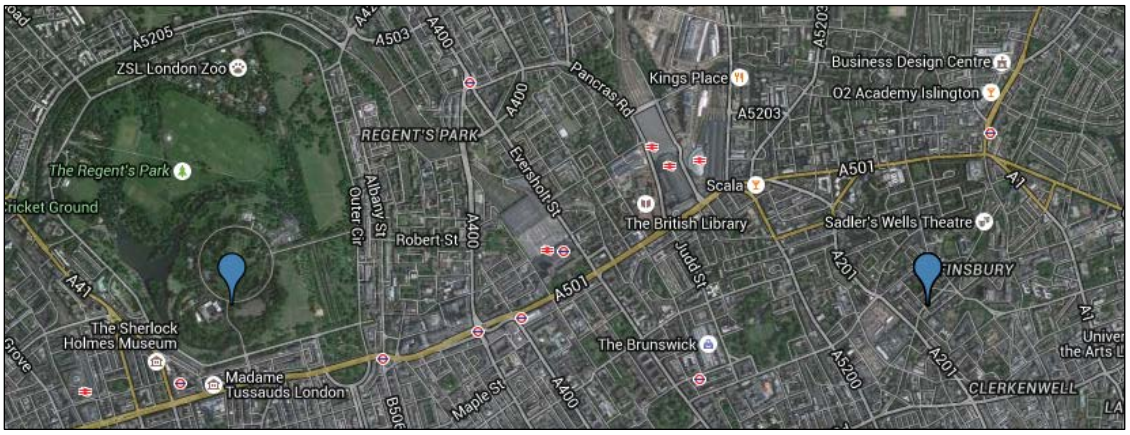
node downstream from the **Derive** node labelled **Longitude**.
2. Edit the **googleMaps** node.
3. Click the **Settings** tab (if necessary)
   - In the **Latitude Field** box, select **Taxis_Needed_Here_Latitude**
   - In the **Longitude Field** box, select **Taxis_Needed_Here_Longitude**
4. Click the **Plot Settings** tab. Under Save map to
   - Click **Custom directory**
   - Browse for **C:\Training\2328**
   - Save the map to the file **mymap.htm**. If you get a message that the file already exists, click **Yes** to replace it.

     The results are as follows:



5. Click **Run**. If you get a message that "Internet Explorer restricted this webpage from running scripts or ActiveX controls", click "Allow blocked content".

   If you have an internet connection, the results will appear similar to the following (if you do not have an internet connection, please read through the following steps).



6. Hover your cursor over the point on the right (labeled INSBURY).

   The results appear as follows:

   This is the location of nine records. The data displayed represent the last record (record #10).

7. Switch back to **Modeler**.
8. From the **File** menu, click **Close All Streams** to close all open streams. Do not save a stream when asked.

We Value Your Feedback!

- Don’t forget to submit your Think 2018 session and speaker feedback! Your feedback is very important to us – we use it to continually improve the conference.
- Access the Think 2018 agenda tool to quickly submit your surveys from your smartphone, laptop or conference kiosk.