Course Guide

Introduction to IBM SPSS Modeler and Data Science (v18.1.1)

Course code 0A008 ERC 1.0

IBM Training

Demonstration 1:
Work with IBM SPSS Modeler

Purpose:
You will become familiar with the IBM SPSS Modeler user interface.

Data file:	**demo_data_working_with_modeler.xlsx**
Data folder:	**C:\Training\0A008**
Stream file:	**unit_02_demonstration_1_start.str**
Stream file folder:	**C:\Training\0A008\02-Introduction_to_IBM_SPSS_Modeler\Start**

Note about demonstrations and exercises.

In this course, IBM SPSS Modeler will be demonstrated using the Microsoft Windows 10 Operating System, using the 64-bit Client version of IBM SPSS Modeler v18.1.1.

Also, the exercises and their solutions assume that this environment is used.

For other Operating Systems, or for connecting to IBM SPSS Modeler Server, refer to the Help in case instructions in this course do not match your results.

Task 1. Start IBM SPSS Modeler and explore the user interface.

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.

   Note: IBM SPSS Modeler 18.1.1 displays "IBM SPSS Modeler 18.1" in the Start menu and program name.

   A welcome dialog box displays at the start of a session, asking you what you would like to do. You can open a demo stream, a recently used stream, or create a new stream. You can also launch the Application Examples tutorial, which provides brief introductions to specific models, starting off from realistic business questions (you can also access these examples via the Help menu). The welcome dialog box also lets you explore the new features in the latest release of IBM SPSS Modeler.

   In this demonstration you will not use any of these features.

2. Click **Cancel** to close the welcome dialog box.

   The main menu has common entries such as File, Edit, Window, and Help. Specific IBM SPSS Modeler menu entries are Insert, View, Tools, and SuperNode.

The toolbar shows a number of standard buttons and buttons that pertain to IBM SPSS Modeler specifically.

The largest area of the IBM SPSS Modeler window is the stream canvas, the area where you will build your streams.

Across the bottom of the window, below the stream canvas, you have the palettes, the containers for the nodes. From left to right, they follow the order of data analysis, from importing data (the Sources palette), via data preparation (the Record Ops and Field Ops palettes), to output generation (the Graphs, Modeling, Output and Export palettes).

At the right side, two panes manage the work. The upper pane is the Manager pane where you manage streams, output, and models.

3. In the **Manager** pane, click the **Streams** tab, if necessary.

   This tab lists the open streams. Likewise, the Outputs tab will list output items such as tables and graphs; the Models tab will show the models that you have built.

   The Projects pane, the lower pane at the right side, organizes the work in two ways, projects and classes.

4. In the **Projects** pane, click the **CRISP-DM** tab, if necessary.

   This tab helps to organize items according to the stages of CRISP-DM. Even though some items do not involve work in IBM SPSS Modeler, the CRISP-DM tab includes all six stages of the CRISP-DM process model so that there is a central location for storing and tracking all materials associated with the project. For example, the Business Understanding stage typically involves documentation to describe the goals of the project. Such documentation can be stored in the Business Understanding folder, for future reference and inclusion in reports.

   You can also store the work according to the type of the object in the Classes tab.

5. Click the **Classes** tab.

   Objects can be added to any of the following categories: Streams, Nodes, Generated Models, Tables, Graphs & Reports, and Other (for example documents relevant to the project).

Task 2. Set defaults.

It is useful to set the default folder when opening and saving files.

1. From the **File** menu, click **Set Directory**.
2. Beside **Look in**, navigate to the **C:\Training\0A008** folder, and then click **Set**.

   IBM SPSS Modeler will automatically save a stream every five minutes. You can change this setting, if preferred.

3. Click the **Tools** menu.
4. Click **Options**, and then click **System Options**.
5. Specify the value for **Stream auto save interval (minutes)**, if different from **5**.
6. Click **OK** to close the **System Options** dialog box.

   Note: From this point forward, the instruction will just be to close the dialog box.

   You will return to the stream canvas.

   IBM SPSS Modeler can run in Traditional mode and Analytic Server mode. For this course, ensure that Traditional mode is the default.
7. Click the **Tools** menu.
8. Click **Options**, click **User Options**, click the **Mode** tab, and then ensure that the **Traditional mode** option is selected.
9. Close the **User Options** dialog box.

   If preferred, you can remove a node from a palette if it will no longer be used.

   In that case, click Tools, and then click Manage Palettes. Select the palette you want to modify and then click the Edit selection button ✎ to remove or add nodes.

   In this course, no palette will be modified.

Task 3. Create a stream (dry run).

In this task you can practice your skills in creating and editing streams. No actual data will be used in this task, so you can focus on the user interface in a dry run.

The following table outlines the operations in this task, and the appropriate palette/node for the operation.

Operation	Palette – Node
Import data from a Microsoft Excel file	Sources palette - Excel node
Sample records	Record Ops palette - Sample node
Derived a new fields	Field Ops palette - Derive node
Request a histogram graph	Graphs palette - Histogram node
Export data to an IBM SPSS Statistics file	Export palette - Statistics Export node
Add a comment to the Sample node	Sample node - context menu - New Comment

You will mimic importing data from a Microsoft Excel file. Nodes to import data are located on the Sources palette.

1. Click the **Sources** palette.

   The Excel node lets you import data from Microsoft Excel.

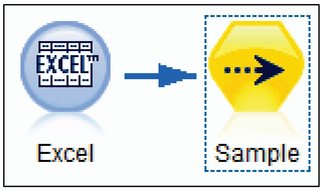
2. Double-click the **Excel** node.

   An Excel node is placed on the stream canvas.

   You will sample records. Sampling records is a record operation, so you will find the appropriate node on the Record Ops palette.

3. Click the **Record Ops** palette, and then double-click the **Sample** node.

   The results appear as follows:

   The Sample node is automatically added downstream from the Excel node, because the Excel node had focus.

   When no node has focus on the stream canvas, the new node will not be connected automatically. To learn how to connect nodes when they are not connected automatically, delete the existing connection.

4. Right-click the connection between the **Excel** node and the **Sample** node, and then click **Delete Connection**.

   You can use the middle-mouse button to connect the nodes: keeping the middle mouse button pressed, click the first node, drag to the second node, and then release the middle-mouse button. If you do not have a middle-mouse button, you can press the Alt key and use the primary mouse button (usually the left mouse button). Alternatively, right-click the node that you want to connect from, select Connect from the context menu, and then click the node you want to connect to. The latter method is demonstrated here.

5. Right-click the **Excel** node, click **Connect** from the context menu, and then click the **Sample** node.

   You restored the connection between the two nodes.

   Note: From this point forward, the instruction will just be to add the new node to the stream or downstream from another node. This implies that the nodes are connected.

   You will derive a new field. Deriving a new field is a field operation, so you will find the appropriate node on the Field Ops palette.

6. Click the **Field Ops** palette, and then double-click the **Derive** node.

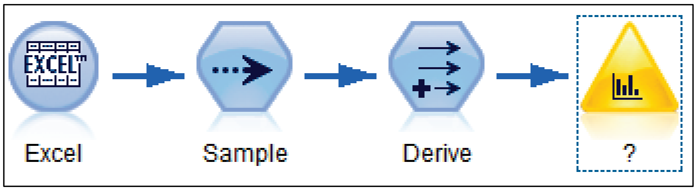
   This places a Derive node downstream from the Sample node.

   Having derived a new field, you might want to examine it graphically.

7. Click the **Graphs** palette, and then double-click the **Histogram** node.

   This places a Histogram node downstream from the Derive node.

   The results appear similar to the following:

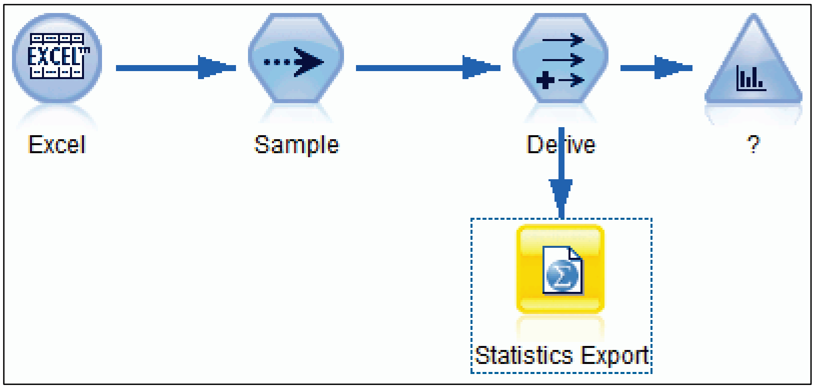


   The Histogram node asks for a field name. No data is used in this task, so leave this as it is.

   You imported data from Excel, have drawn a sample, and derived a new field. This has created a new data dataset (not actually, because this is a dry run only). You will export the new dataset to another format, IBM SPSS Statistics in this example.

8. Click the **Export** palette, and then double-click the **Statistics Export** node.

   The results appear similar to the following:



   This Statistics Export node is placed downstream from the Derive node, not downstream from the Histogram node. Recall that nodes from the Graphs, Modeling, Output, and Export palettes are terminal nodes and cannot be connected to each other. Also, conceptually, the data sits in the Derive node, not in the Histogram node, so the Statistics Export node must pull the data from the Derive node.

   You will add a comment to the Sample node. Right-clicking a node invokes a context menu with many options, among which the option to add a comment.

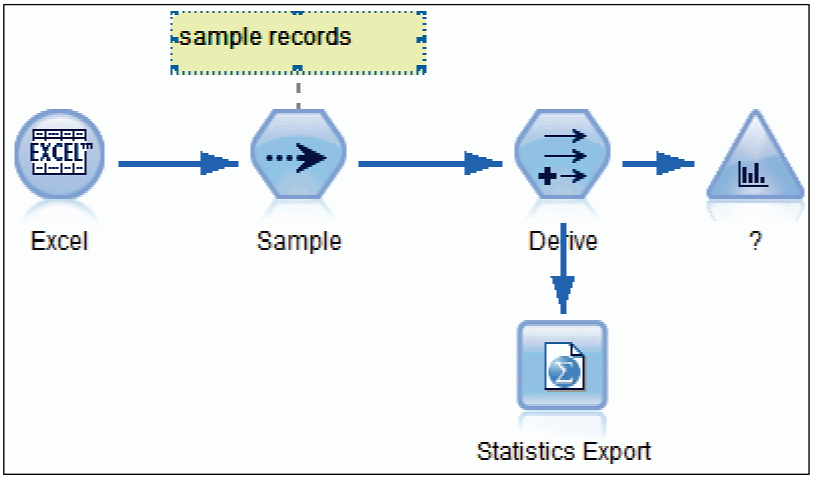
9. On the stream canvas, right-click the **Sample** node.

   You can edit the node, which opens a dialog box to set the node's options (alternatively, double-click the node to edit it). You can cut, delete, connect, disconnect, rename, and copy a node. You can also get a node from the file system (Load Node) and save a node (Save Node) to the file system. Retrieve Node and Store Node get and save a node from/to Collaborations and Deployment Services (C&DS), and are only relevant when C&DS is installed.

   In this example, you will add a comment to the Sample node.

10. Click **New Comment** from the context menu, type **sample records** in the comment box, and then click outside the comment box to remove focus.

    The results appear similar to the following:



Leave IBM SPSS Modeler and the stream open for the next task.

From this point forward, until further notice, leave IBM SPSS Modeler and the stream open from one task to another.

Task 4. Modify a stream.

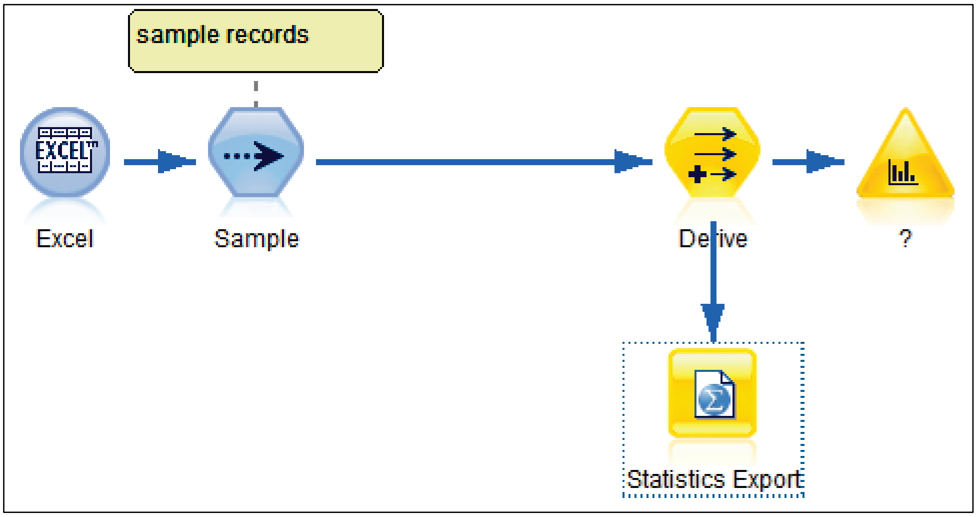
Continuing with the example, suppose that specific records must be selected after records have been sampled. Selecting records is a record operation, so the node to use can be found in the Record Ops palette. The appropriate node is named Select. (Note: The Select node might be somewhat confusing for those familiar with SQL, because the SQL SELECT statement selects fields whereas IBM SPSS Modeler's Select node selects records.)

Selecting records is a record operation, so the node to use must be found in the Record Ops palette.

You will make room for the node that needs to be inserted.

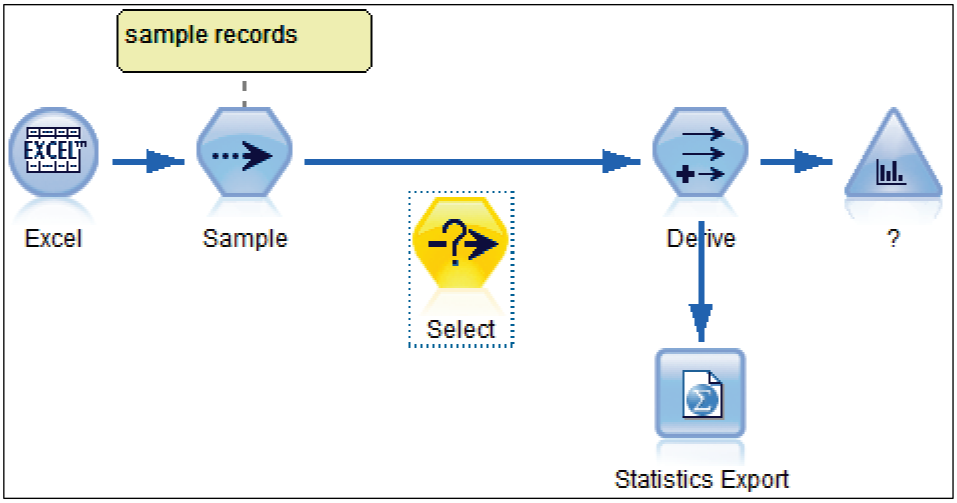
1. Using the mouse, draw a rectangle that selects the **Derive** node, the **Histogram** node, and the **Statistics Export** node, and then drag the selected objects to the right.

   The results appear similar to the following:

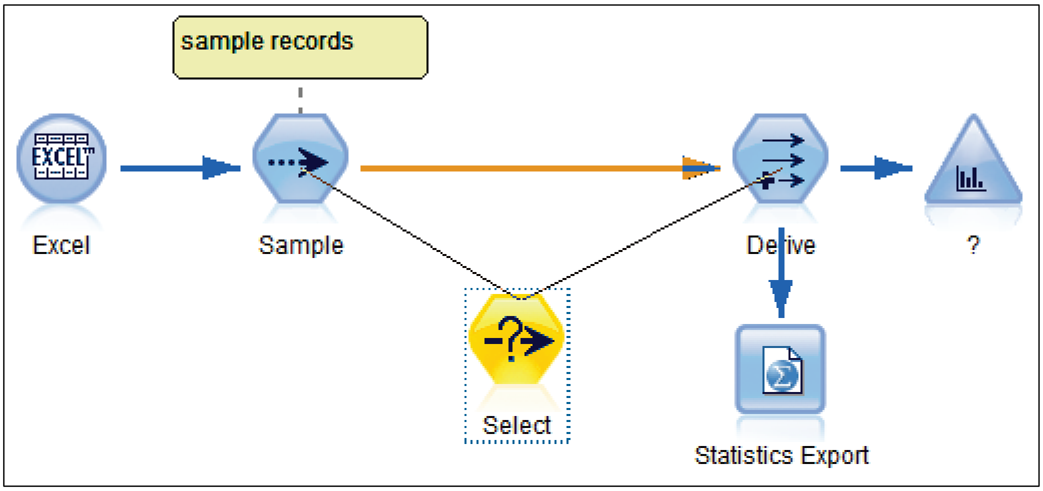


2. From the **Record Ops** palette, click the **Select** node and drag it to the stream canvas, below the connection from the **Sample** node to the **Derive** node.

   The results appear similar to the following:

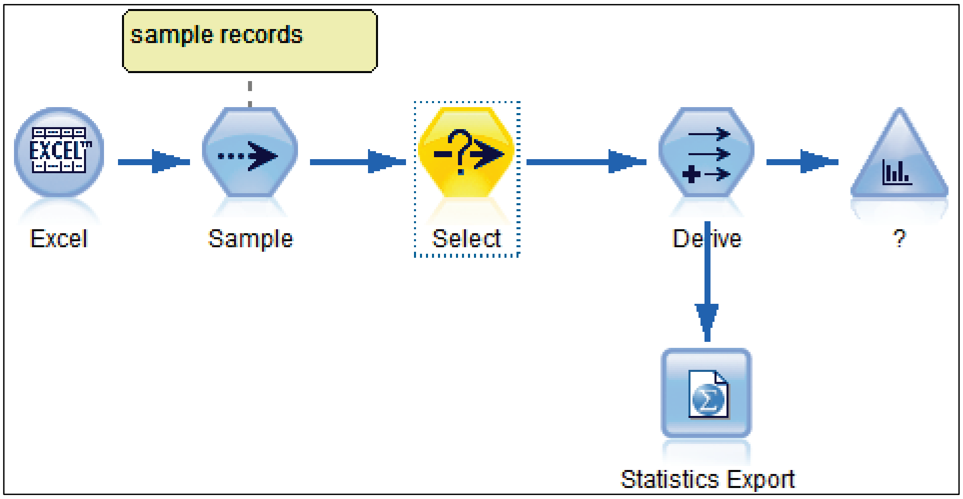


3. Click the **connection** from the **Sample** node to the **Derive** node, and then drag it over the **Select** node, as shown in the figure that follows.



4. Reposition the **Select** node, so that it is on the same height as the **Sample** and **Derive** node.

   The results appear similar to the following:



Rather than inserting a node, you may want to get rid of a node. For example, when developing a stream it is useful to have a Sample node to limit execution time, but in production mode you do not want to sample records.

There are two options to exclude a node from the analysis. You can remove a node from the stream or you can disable it. Disabling a node will leave the node in the stream, but it will be ignored when the stream is executed.

To remove a node, double-click it with the middle mouse button (or press the Alt key and double-click the left mouse button when you do not have a middle-mouse button). This will disconnect the node, and it will connect the node that is upstream from the selected node to the node that is downstream from the selected node. You can then delete the selected node.

To disable it, right-click the node, and then click Disable Node from the context menu.

You will disable the Sample node.

5. Right-click the **Sample** node, and then from the context menu click **Disable Node**.

   The node remains in the stream and is greyed out, indicating that it will be ignored when the stream is executed.

   You will encapsulate some nodes in a SuperNode, to make the stream neater. To create a SuperNode, select the nodes that you want to encapsulate in the SuperNode, right-click one of the nodes, and then select Create SuperNode from the context menu.

6. Use the mouse to draw a rectangle, so that the **Sample** (plus the comment box), **Select**, and **Derive** node are selected.

7. Right-click one of the selected nodes, and then from the context menu click **Create SuperNode**.

   The SuperNode is represented by a star icon. You will edit the SuperNode to view its content.

8. Right-click the **SuperNode**, and then click **Zoom In**.

   The SuperNode window opens and lets you view and edit the nodes.

   When you want to close the SuperNode window, click the Zoom out button. (Take care not to click the Close X button at the right top corner, because that will end the IBM SPSS Modeler session.)

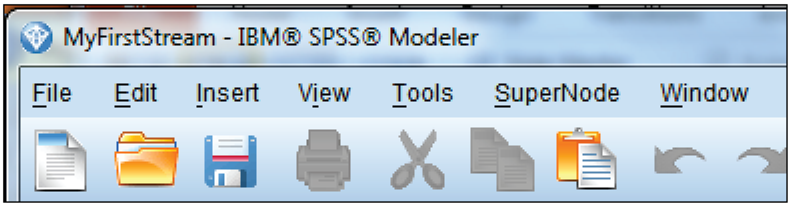
9. Click the **Zoom out** button.

   You will save your stream.

10. From the **File** menu, click **Save Stream**.

11. In **File Name** enter **MyFirstStream.str**, and then click **Save**.

    The window's title bar shows the file name, as shown below.



    Now that you saved the stream you can close it.

12. From the **File** menu, click **Close Stream**.

    When no stream is open, the stream canvas does not display. You can choose New Stream from the File menu to have the stream canvas back again. Alternatively, you can open an existing stream, as demonstrated in the next task.

Task 5. Generate a Select node from Table output.

Nodes can be placed on the stream canvas by selecting them from the appropriate palette. For some nodes there is a very efficient alternative, as demonstrated in this task.

This task uses an existing stream, unit_02_demonstration_1_task_5_start.str.

1. From the **File** menu, click **Open Stream**.
2. Navigate to the **C:\Training\0A008\02-Introduction_to_IBM_SPSS_Modeler\Start** folder, and then open **unit_02_demonstration_1_task_5_start.str**.

   Note: From this point forward, the instruction will just be to open the stream.
3. Double-click the **Microsoft Excel** node to edit it.

   Note: From this point forward, the instruction will just be to edit the node.

   The stream imports data from a Microsoft Excel named demo_data_working_with_modeler.xlsx.
4. Close the dialog box.

   The stream also includes a Table node. The Table node is one of the most popular nodes, because it lets you view the data.

   There are several ways to run the Table node. You can edit the node and then click the Run button in the dialog box. Or you can click the node to select it, and then click the Run Selection ▶ button in the toolbar. Another option is to right-click the node and choose Run from the context menu.

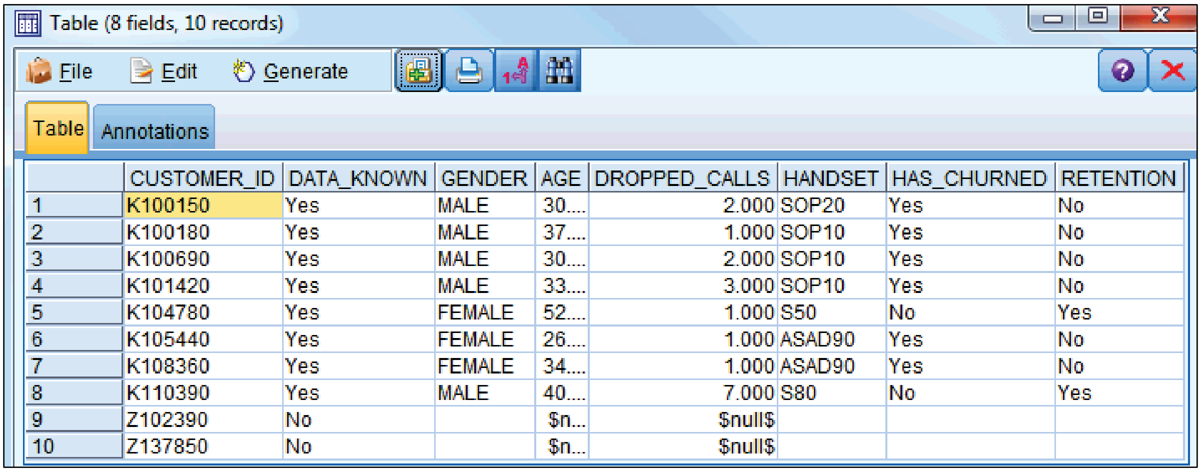
   Because the Table node is the only terminal node in the stream, you can also run the Table node by executing the entire stream by clicking the Run the current stream ▶ button.

   It is recommended to experiment with these options. Refer to the previous pages in this unit if you want to review the information about how to run nodes.

   From this point forward, the instruction will just be to run a node.

5. Run the **Table** node.

   The results appear as follows:



	CUSTOMER_ID	DATA_KNOWN	GENDER	AGE	DROPPED_CALLS	HANDSET	HAS_CHURNED	RETENTION
1	K100150	Yes	MALE	30....	2.000	SOP20	Yes	No
2	K100180	Yes	MALE	37....	1.000	SOP10	Yes	No
3	K100690	Yes	MALE	30....	2.000	SOP10	Yes	No
4	K101420	Yes	MALE	33....	3.000	SOP10	Yes	No
5	K104780	Yes	FEMALE	52....	1.000	S50	No	Yes
6	K105440	Yes	FEMALE	26....	1.000	ASAD90	Yes	No
7	K108360	Yes	FEMALE	34....	1.000	ASAD90	Yes	No
8	K110390	Yes	MALE	40....	7.000	S80	No	Yes
9	Z102390	No		$n...	$null$			
10	Z137850	No		$n...	$null$			

   Leave this Table output window open.

   Note: In a Microsoft Windows environment, if the table output window does not display, press the Alt key and press the Tab key to go through the open windows, until the Table output window displays.

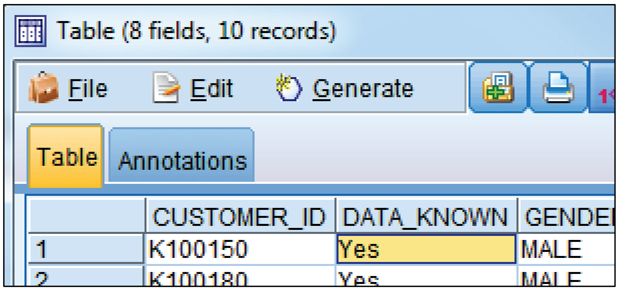
   GENDER through RETENTION are empty or $null$ for the last two records; $null$ represents IBM SPSS Modeler's undefined value.

   You want to select the records with known data. DATA_KNOWN flags whether data is known for a record.

   One way to select records with known data is to add a Select node from the Record Ops palette to the stream canvas, edit it, type the condition DATA_KNOWN = "Yes", and so forth. A quicker and more user-friendly alternative is to generate the Select node from the Table output window.

6. In the **first row**, in the **DATA_KNOWN** column, click the value **Yes**, so it is selected.

   The results appear as follows:



	CUSTOMER_ID	DATA_KNOWN	GENDE
1	K100150	Yes	MALE
2	K100180	Yes	MALE

7. From the menu, click **Generate**.

   A Select node can be generated in three ways. Refer to the previous pages in this unit for the details. Because only a single value is selected, there is no difference between Select Node ("And") and Select Node ("Or").

8. Click Select Node ("And").

   The Table output window can be closed now. There are two ways to close an output object.

9. Point the mouse to the X in the right upper corner of the output window, next to the Maximize icon (only point, do not click).

   Clicking X will close the object and store it in the Outputs tab in the Manager pane. Clicking OK in the output window has the same effect. You can view the output later by re-opening the object in the Outputs tab in the Manager pane.

10. Point the mouse to the X in the right upper corner of the output window, next to the question mark (only point, do not click).

    Clicking this X will close and delete the object and it will not be added to the Outputs tab in the Manager pane. If you want to view the output later, you need to rerun the stream.

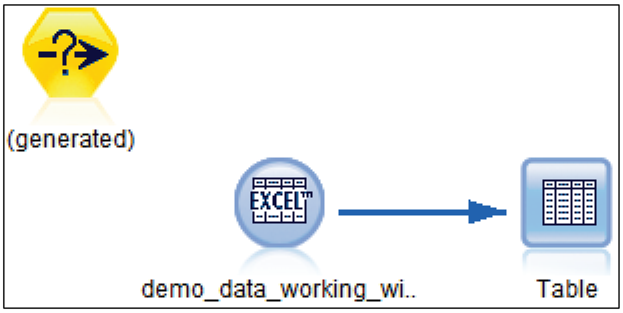
    It is recommended to experiment with both options.

11. Click **OK** to close the **Table** output window.

    Note: From this point forward, the instruction will just be to close the output window.

    Generating a node from the Table output window has placed a Select node named (generated) on the stream canvas.

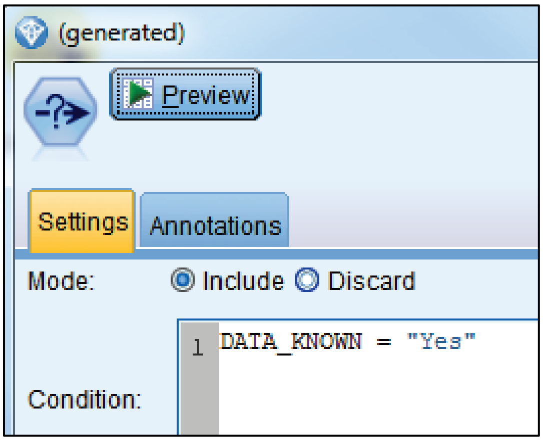
    The results appear as follows:



You will examine the node.

12. Double-click the node named **(generated)**.

    The results appear as follows:



Mode: ● Include ○ Discard

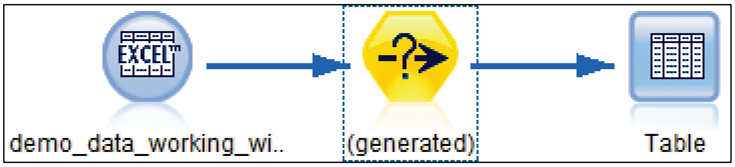
Condition:

```
DATA_KNOWN = "Yes"
```

Only records meeting the condition will be output. However, at this point the Select node is not part of the stream, so it has no effect on the data flow. The node needs to be included in the stream.

13. Close the dialog box.
14. Insert the **Select** node named **(generated)** between the **Excel** source node and the **Table** node. (To insert a node, refer to the instructions in the previous task, if preferred.)

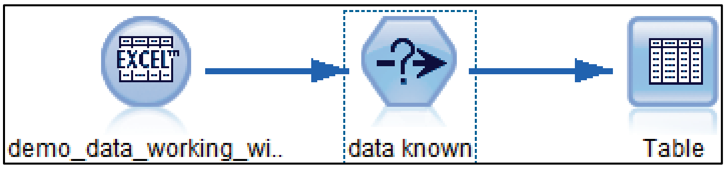
    The results appear similar to the following:



You can add a comment to the node or rename the node, to make it clear which records are selected. In this task you will rename the node.

15. Edit the **Select** node named **(generated)**.
16. Click the **Annotations** tab.
17. Beside **Name**, click **Custom**.
18. Replace the text **(generated)** by **data known**.
19. Close the dialog box.

    The results appear as follows:



You can run the Table node to check the results.

20. Run the **Table** node.

    Only records with known data are displayed.

    Selecting records by generating the appropriate Select node from table output is an extremely useful feature, which will be used throughout the course.

21. Close the **Table** output window.

    This completes the demonstration for this unit. You will create a clean state for the exercise.

22. From the **File** menu, click **Close Stream**, and then click **No** when asked to save the stream.

23. From the **File** menu, click **New Stream**.

    You will delete all output.

24. In the top-right **Manager** pane, click the **Outputs** tab.

    The results appear similar to the following:

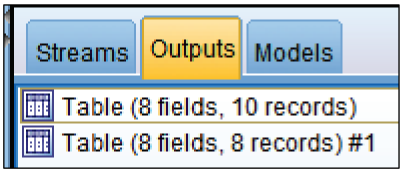


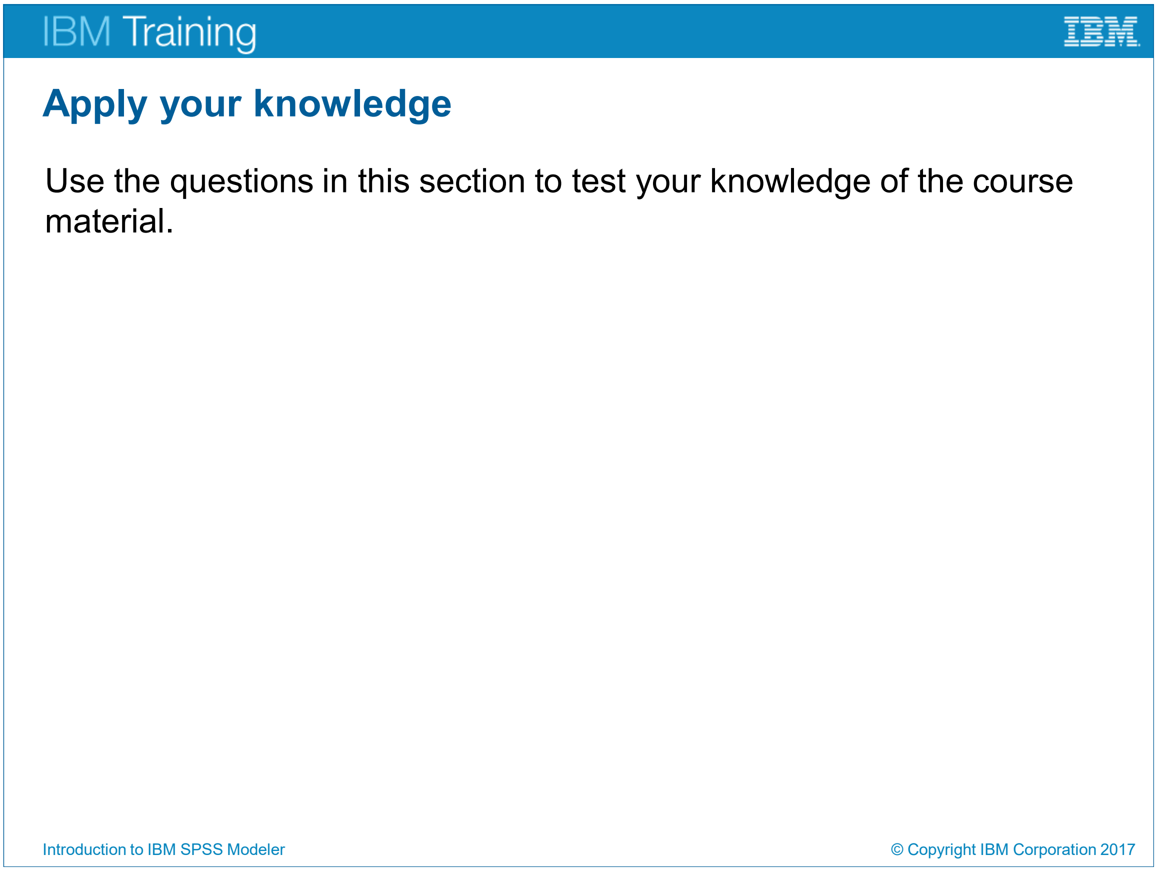
Table (8 fields, 10 records)
Table (8 fields, 8 records) #1

25. Right-click an empty area in the **Outputs** tab in the **Manager** pane, and then click **Delete all**.

    Leave IBM SPSS Modeler open for the exercise.

Results:
You have become familiar with IBM SPSS Modeler's user interface, and created, edited, opened and saved streams. Also, you learned about SuperNodes and how to generate a Select node from the Table output window.

You can find the completed streams in the **C:\Training\0A008\02-Introduction_to_IBM_SPSS_Modeler\Solutions** folder.



IBM Training

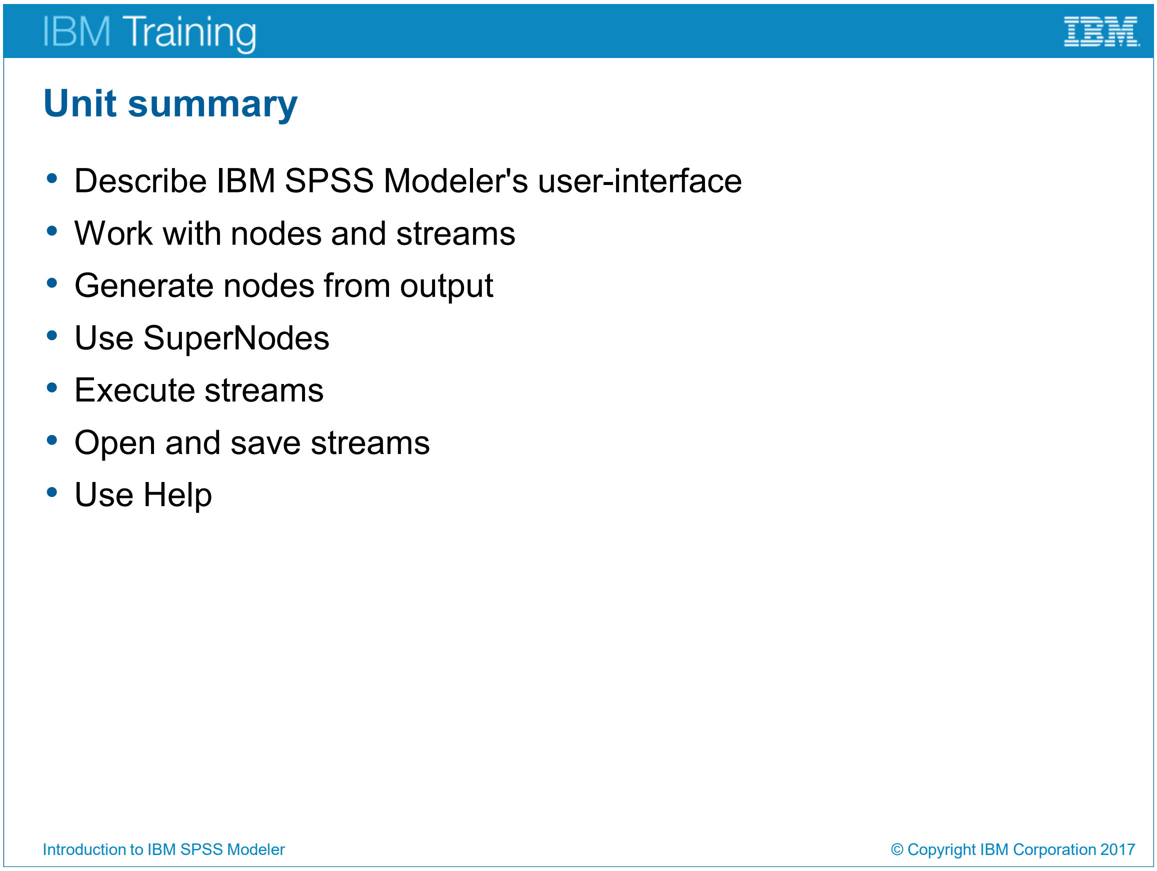
Apply your knowledge

Use the questions in this section to test your knowledge of the course material.

Introduction to IBM SPSS Modeler

© Copyright IBM Corporation 2017

Apply your knowledge



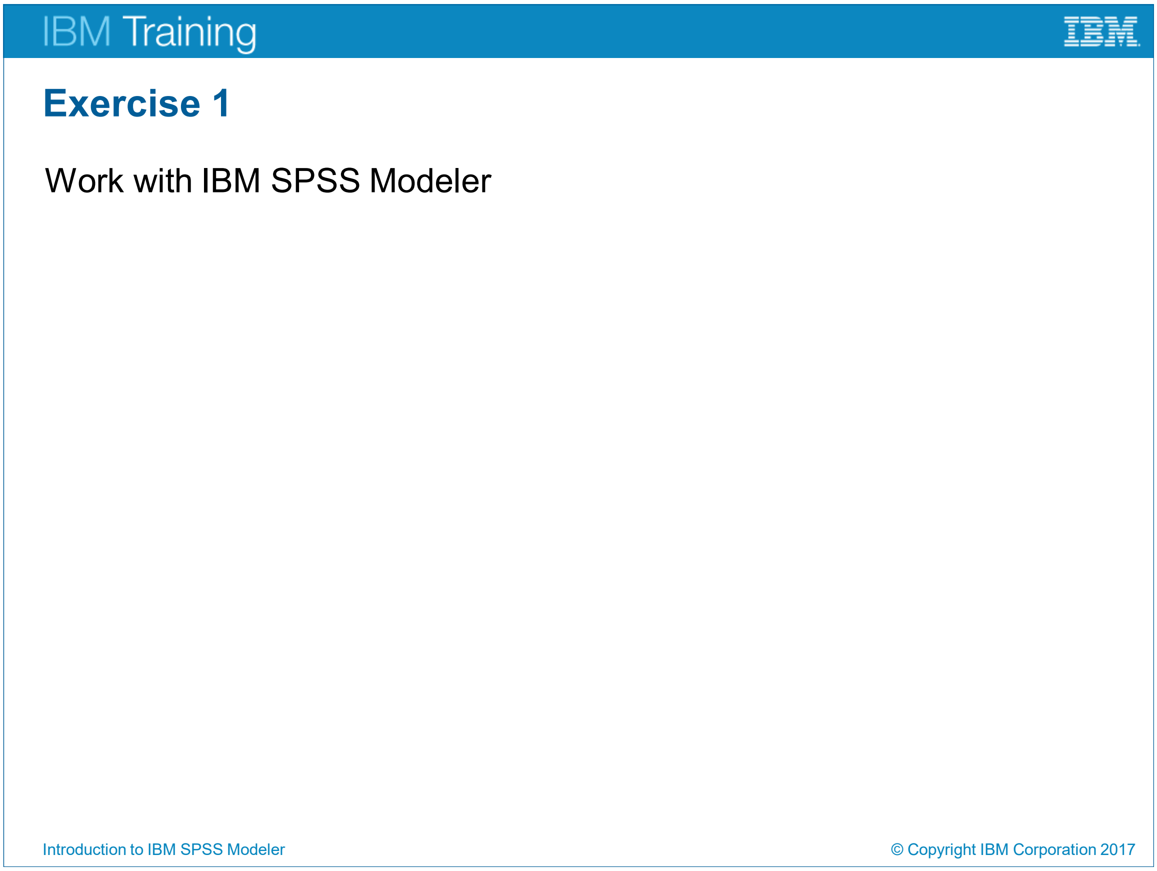
IBM Training

Unit summary

- Describe IBM SPSS Modeler's user-interface
- Work with nodes and streams
- Generate nodes from output
- Use SuperNodes
- Execute streams
- Open and save streams
- Use Help

Introduction to IBM SPSS Modeler

Unit summary



IBM Training

Exercise 1

Work with IBM SPSS Modeler

Introduction to IBM SPSS Modeler

Exercise 1: Work with IBM SPSS Modeler

Exercise 1: Work with IBM SPSS Modeler

Data file:	**exercise_data_working_with_modeler.xlsx**
Data folder:	**C:\Training\0A008**
Stream file:	**unit_02_exercise_1_start.str**
Stream file folder:	**C:\Training\0A008\02-Introduction_to_IBM_SPSS_Modeler\Start**

Practice your skills of building and running streams in this exercise. Initially no data is used, so that you can focus on IBM SPSS Modeler's interface. In the last two tasks you will use a dataset taken from a firm named ACME (a fictitious company selling sport products, online and via mail campaigns).

- Create a stream, that:
  - imports data from an IBM SPSS Statistics (.sav) file (at this moment, no specific data file is used, so place the correct node on the stream canvas, but do not specify a data file)
  - selects records from the imported IBM SPSS Statistics file (place the correct node on the stream canvas, without further specifications)
  - sorts records (place the correct node on the stream canvas, without further specifications)
  - derives a new field (place the correct node on the stream canvas, without further specifications)
  - requests a Histogram graph for the field that was just derived (place the correct node on the stream canvas, without further specifications)
  - derives a second field (place the correct node on the stream canvas, without further specifications)
  - requests a Distribution graph for the second derived field (place the correct node on the stream canvas, without further specifications)
  - exports the data to a Microsoft Excel file (place the correct node on the stream canvas, without further specifications)

- Modify the stream that you just created:
  - remove the second Derive node, but ensure that the data still flows from the data source to the Excel export node
  - remove the Distribution node that you had on the Derive node that you just removed
  - export the data to an IBM SPSS Statistics file (next to the export to Excel)
  - add a comment to the Derive node
  - add a freestanding comment to the stream, such as your name and the date that you created the stream
- Save the stream; name it **MyExercise.str**.
- You will create another stream in a new window. This new stream resembles the stream that you have just created. The only difference is that only a sample of records should be exported to the IBM SPSS Statistics file.

  To create this new stream, copy the nodes in the stream that you already have and paste them into a new window. Then, modify the stream so that only a sample of records is exported to an IBM SPSS Statistics file.
- Make this stream neat by encapsulating the Select and Sort node into a SuperNode, and rename the SuperNode to **data preparation**.

  Zoom in on the SuperNode to verify its content; if the content is okay, zoom out of the SuperNode. (To zoom out, do not click the Close ✕ button in the right-upper corner, because that will end the IBM SPSS Modeler session; instead, click the Zoom out 🔍 button.)

  In the next tasks you will work with data and select records by generating a Select node from the Table output window.
- Open **unit_02_exercise_1_start.str**, located in the **C:\Training\0A008\02-Introduction_to_IBM_SPSS_Modeler\Start** folder, and then run the **Table** node to get familiar with the data.

  How many records and fields do you have?

  Select only those customers that have received the test mailing, by generating the required **Select** node from the **Table** output window and including it in the stream.

  How many records are selected?

- Continue to work with the dataset that includes only customers that were in the test mailing. You may have noticed that GENDER is missing (UNKOWN or empty space) for some customers. Select only **female** and **male** customers.

  How many records are selected?
- Exit IBM SPSS Modeler without saving anything.

For more information about where to work and the exercise results, refer to the Tasks and Results section that follows. If you need more information to complete a task, refer to the earlier demonstration for detailed steps.

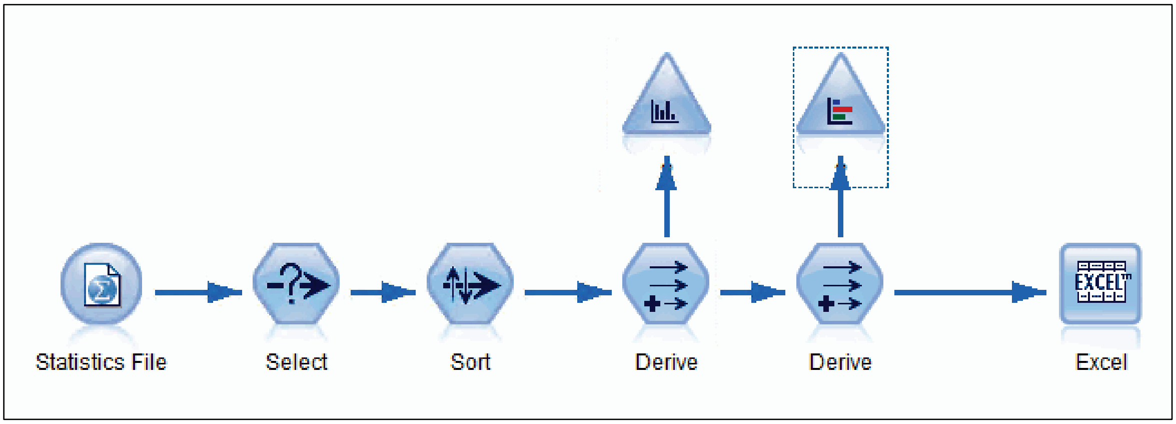
Exercise 1: Tasks and Results

Task 1. Create a stream that reads data from IBM SPSS Statistics and exports data to Microsoft Excel (dry run).

- Place the following nodes on the stream canvas:
  - **Statistics File** (Sources palette)
  - **Select** (Record Ops palette)
  - **Sort** (Record Ops palette)
  - **Derive** (Field Ops palette)
  - **Histogram** (Graphs palette)
  - **Derive** node (Field Ops palette)
  - **Distribution** (Graphs palette)
  - **Excel** (Export palette)

  Ensure the nodes are connected to form a stream.

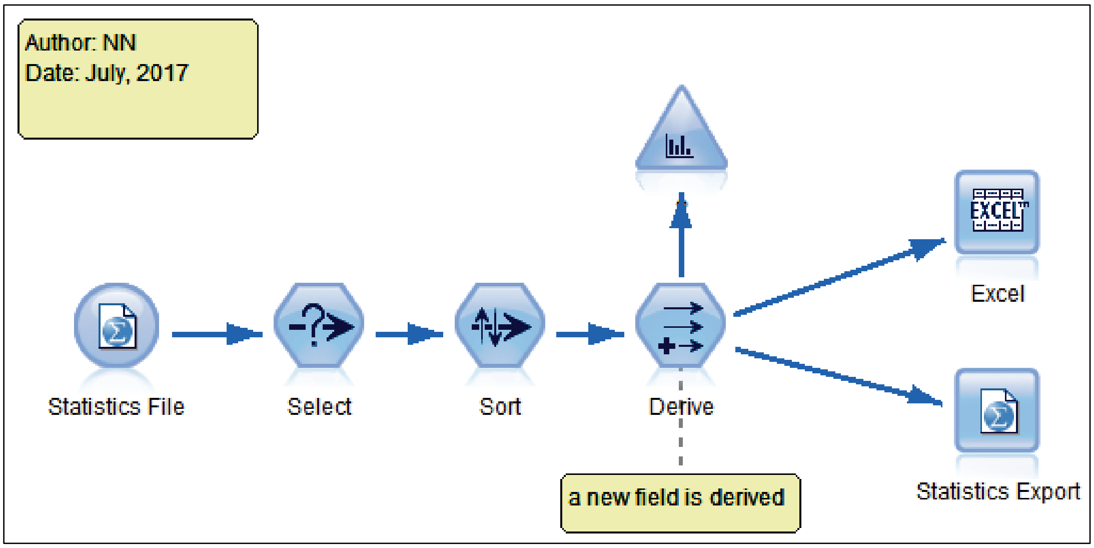
  The results appear similar to the following:



Task 2. Modify the stream.

- Delete the second **Derive** node.
- Delete the **Distribution** node.
- Connect the **Derive** node to the **Excel** node.
- From the **Export** palette, add a **Statistics Export** node downstream from the **Derive** node.
- Right-click the **Derive** node and add a comment such as **new field is derived**.
- Right-click an empty area on the stream canvas, and add a comment stating the author and date.

  Your stream should appear similar to the following:



Task 3. Save the stream.

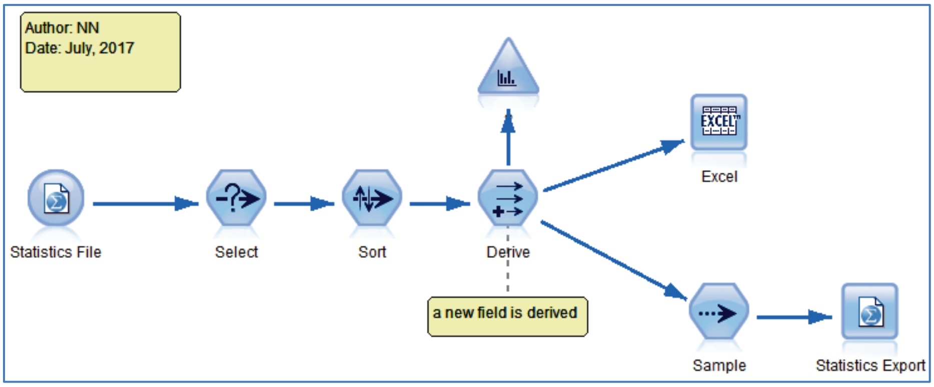
- From the **File** menu, click **Save Stream**, and type **MyExercise.str** and then click **Save**.

  The title bar will show the file name.

Task 4. Create a new stream by copying and pasting from an existing stream.

- Select all nodes (for example, by drawing a rectangle that includes all nodes, by clicking **Ctrl+A**, or by clicking the **Edit** menu, and then choosing **Select All**).
- **Copy** the selected nodes.
- From the **File** menu, click **New Stream** to open a new window.
- **Paste** the content of the clipboard to the stream canvas.
- Insert a **Sample** node between the **Derive** node and the **Statistics Export** node.

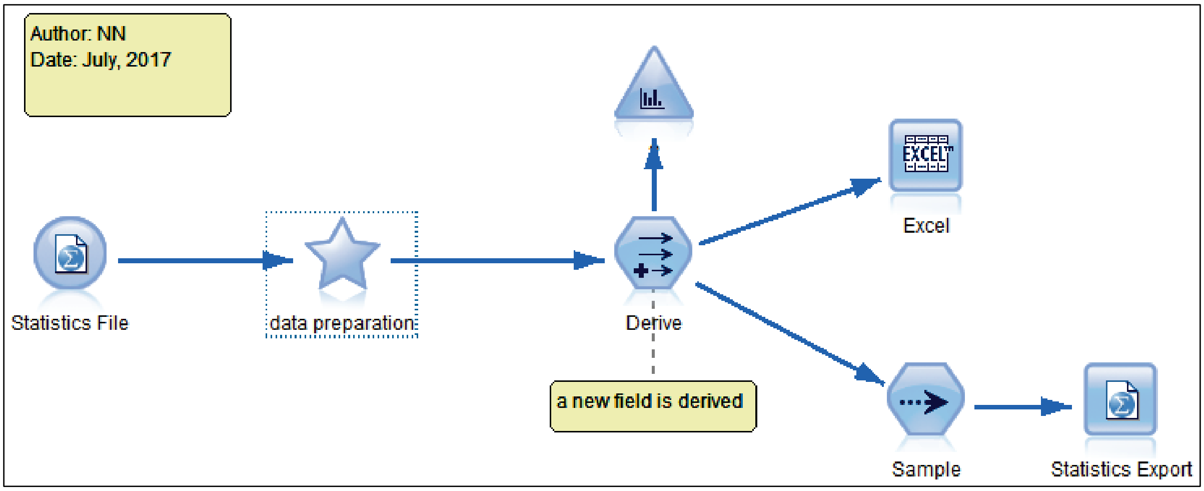
  Your stream should appear similar to the following:



Task 5. Make the stream neat by using a SuperNode.

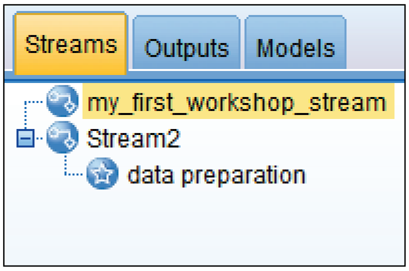
- Select the **Select** and **Sort** node (for example, by using the mouse to drag a rectangle around them).
- **Right-click** one of the selected nodes, and then click **Create SuperNode** from the context menu.
- **Right-click** the SuperNode, select **Rename and Annotate** from the context menu, click the **custom** option if necessary, and then type **data preparation**. (Alternatively, edit the SuperNode and click the Annotations tab.)

Your stream should appear similar to the following:



- Zoom in by double-clicking the SuperNode to edit it, and then click **Zoom In** Zoom In.
- Zoom out by clicking the **Zoom out of SuperNode** ⭐ toolbar button.

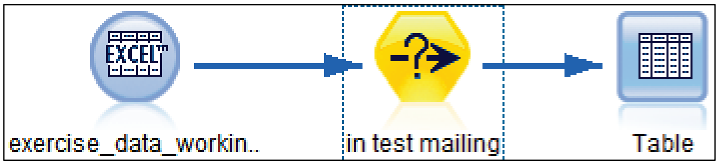
Note: you can also zoom in and zoom out from the **Streams** tab:



Task 6. Generate a Select node from Table output.

- From the **File** menu, click **Open Stream**, navigate to the **C:\Training\0A008\02-Introduction_to_IBM_SPSS_Modeler\Start** folder, open **unit_02_exercise_1_start**, and then run the **Table** node.
- In the **Table** output window, click the **yes** value in the **HAS_RECEIVED_TEST_MAILING** column, click the **Generate** menu, and then click **Select Node ("And")**.
- Insert the generated node named **(generated)** between the Excel node and Table node. Also, rename the generated node to **in test mailing**.

  The results appear similar to the following:



- Run the **Table** node.

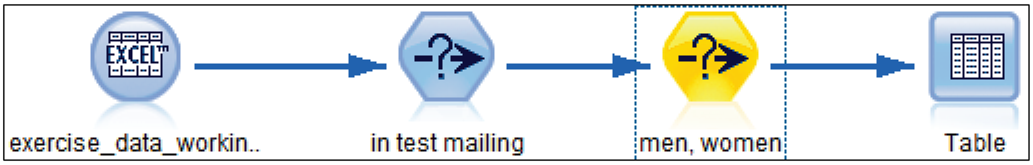
  This will show that 10,000 records are selected.

Task 7. Further record selections.

Next, select men and women from the 10,000 customers in the test mailing.

- In the **Table** output window, use **Ctrl-click** to select a **female** and **male** values, and from the **Generate** menu, click **Select Node ("Or")**.
- Insert the generated node between the **in test mailing** node and the **Table** node, and then rename the node to **men, women**.

  The results appear similar to the following:



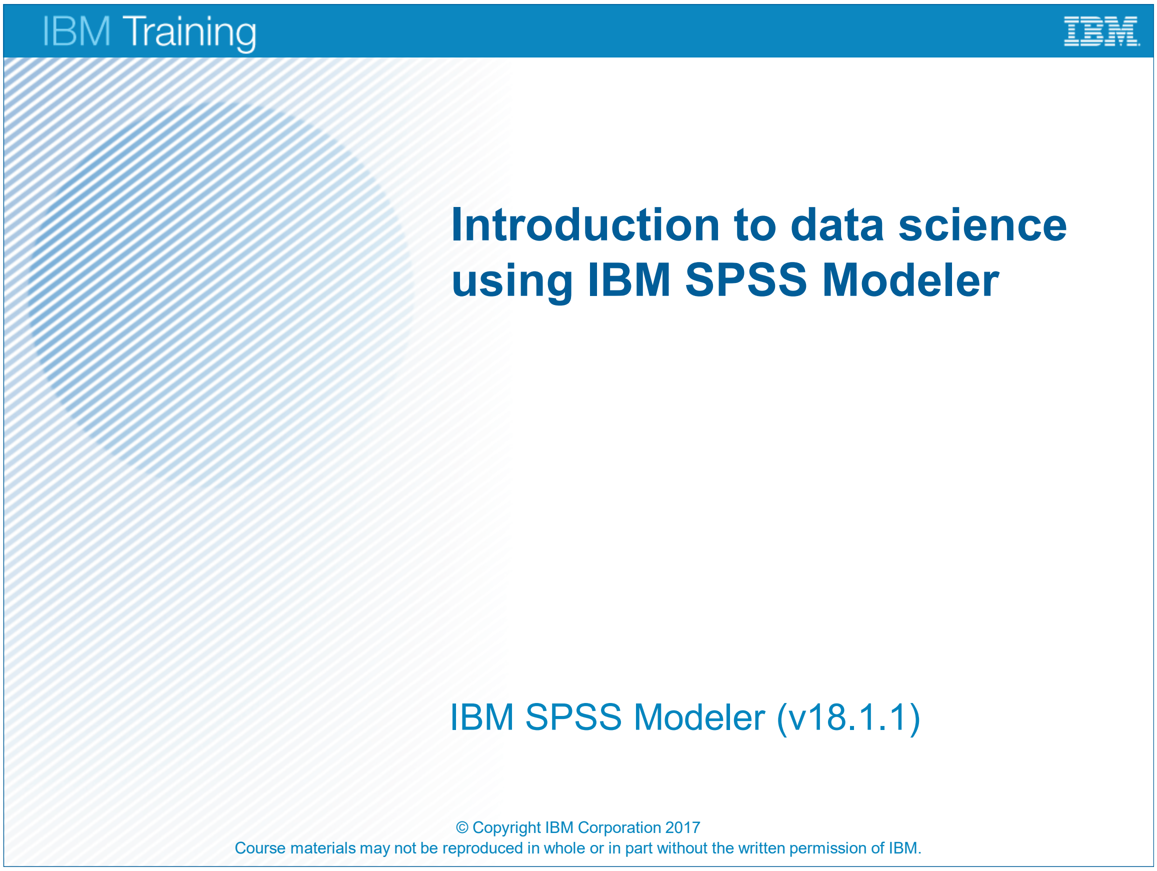
- Run the **Table** node. This will show that there are 9,980 records left.

Task 8. Exit IBM SPSS Modeler.

- From the **File** menu, click **Exit**, and then exit IBM SPSS Modeler without saving anything.

You will find the solution results in the **C:\Training\0A008\02-Introduction_to_IBM_SPSS_Modeler\Solutions** folder.

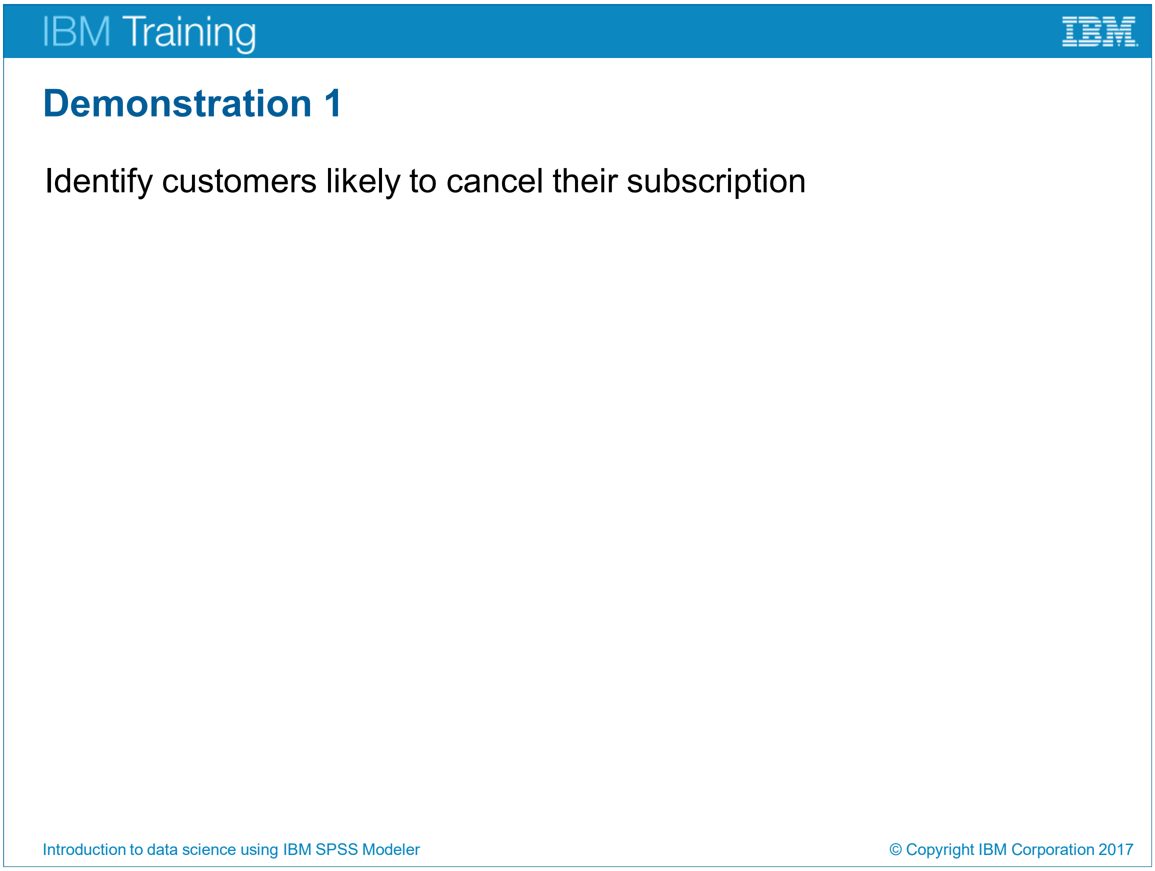
Unit 3 Introduction to data science using IBM SPSS Modeler



IBM Training

Introduction to data science using IBM SPSS Modeler

IBM SPSS Modeler (v18.1.1)



IBM Training

Demonstration 1

Identify customers likely to cancel their subscription

Introduction to data science using IBM SPSS Modeler

© Copyright IBM Corporation 2017

Demonstration 1: Identify customers likely to cancel their subscription

Demonstration 1:
Identify customers likely to cancel their subscription

Purpose:
You work as a data scientist for a telecommunications firm. You want to identify customers who are likely to cause churn by cancelling their subscription. Therefore you will build a model on historical data and apply this model to current customers.

Data files:	**telco x modeling data.xlsx**
	telco x deployment data.xlsx
Data files folder:	**C:\Training\0A008**

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

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.
2. When a welcome window displays, click **Cancel**.

   If you have already configured IBM SPSS Modeler in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A008** folder, and then click **Set**.

Task 2. Import and examine the data.

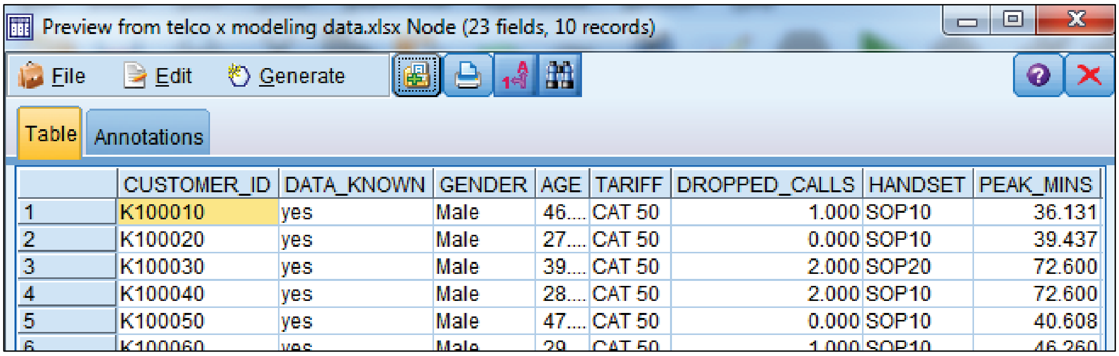
The Microsoft Excel file telco x modeling data.xlsx stores historical data from customers of a (fictitious) telecommunications firm. You will use this dataset to build a model to predict churn.

1. From the **Sources** palette, place an **Excel** node on the stream canvas.
2. Edit the **Excel** node, beside the **Import file** box, click the **Browse for file** ... and then open **telco x modeling data.xlsx**, located in the **C:\Training\0A008** folder.

   The Preview button lets you examine the first 10 records of the dataset.

3. Click the **Preview** button.

   The results appear similar to the following:



Preview from telco x modeling data.xlsx Node (23 fields, 10 records)

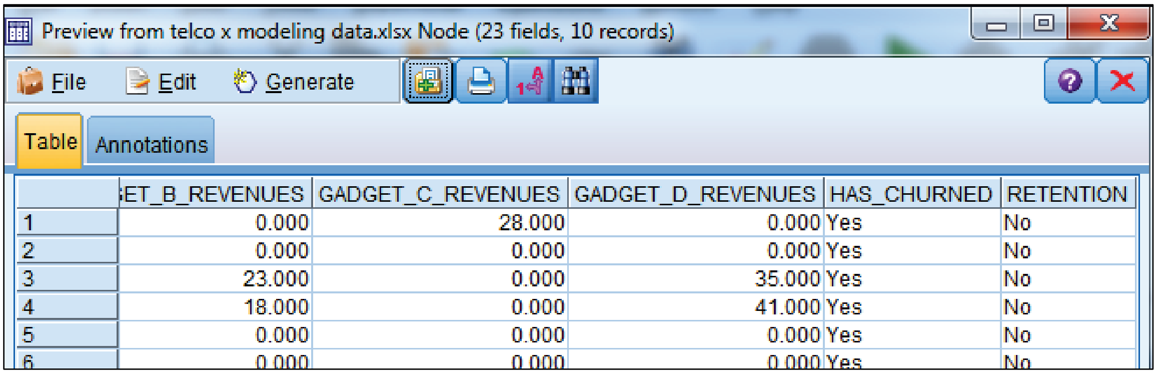
	CUSTOMER_ID	DATA_KNOWN	GENDER	AGE	TARIFF	DROPPED_CALLS	HANDSET	PEAK_MINS
1	K100010	yes	Male	46....	CAT 50	1.000	SOP10	36.131
2	K100020	yes	Male	27....	CAT 50	0.000	SOP10	39.437
3	K100030	yes	Male	39....	CAT 50	2.000	SOP20	72.600
4	K100040	yes	Male	28....	CAT 50	2.000	SOP10	72.600
5	K100050	yes	Male	47....	CAT 50	0.000	SOP10	40.608
6	K100060	yes	Male	29	CAT 50	1.000	SOP10	46.260

   The dataset includes background information (gender, age), and stores information about usage (handset used, number of minutes phoned in the peak hours).

   DATA_KNOWN flags whether a customer's data is known.

4. In the **Preview** output window, scroll to the far-right.

   The results appear similar to the following:



Preview from telco x modeling data.xlsx Node (23 fields, 10 records)

	ET_B_REVENUES	GADGET_C_REVENUES	GADGET_D_REVENUES	HAS_CHURNED	RETENTION
1	0.000	28.000	0.000	Yes	No
2	0.000	0.000	0.000	Yes	No
3	23.000	0.000	35.000	Yes	No
4	18.000	0.000	41.000	Yes	No
5	0.000	0.000	0.000	Yes	No
6	0.000	0.000	0.000	Yes	No

   The field of interest is HAS_CHURNED. This field equals Yes when the customer cancelled his subscription, No when he did not.

   RETENTION is derived from HAS_CHURNED and equals No when HAS_CHURNED equals Yes, and equals Yes when HAS_CHURNED equals No. Thus, RETENTION is the exact counterpart of HAS_CHURNED.

   If you include RETENTION as a predictor for HAS_CHURNED, a model such as CHAID will find the following rules:

   > if RETENTION = Yes then HAS_CHURNED = No
   >
   > if RETENTION = No then HAS_CHURNED = Yes

   This will be a 100% accurate but useless model, because the predictor is derived from the field you want to predict, HAS_CHURNED in this example. Therefore, this field should be left out when you build a model to predict churn.

In general, building models is not throwing in all fields in the dataset as predictor, which is actually one of the most common caveats in model building.

5. Close the **Preview** output window.
6. Close the dialog box to import the Microsoft Excel data file.

Task 3. Select the records for modeling.

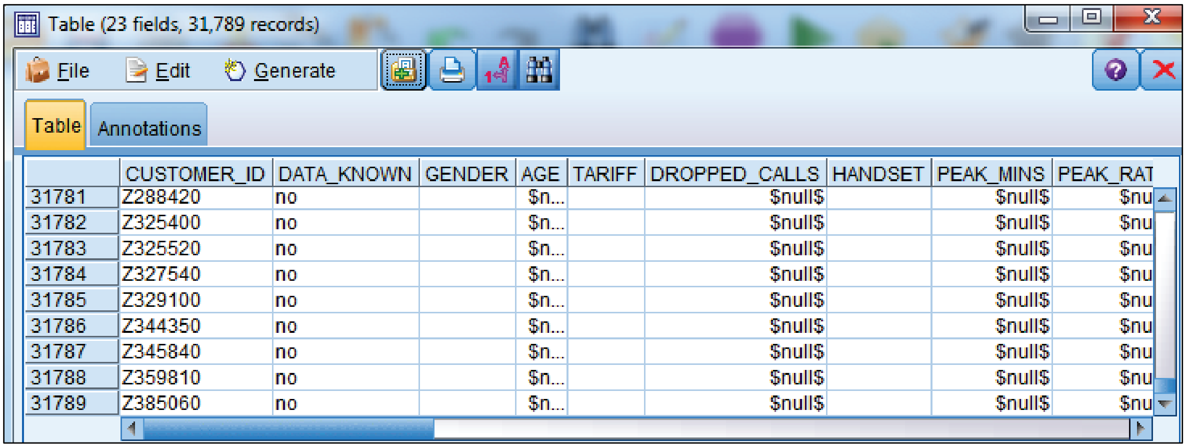
You will explore the data.

1. From the **Output** palette, add a **Table** node downstream from the **Excel** node.
2. Run the **Table** node.

   The title bar in the Table output window reads that there are 31,789 records in the dataset.

3. Scroll all the way down in the **Table** output window.

   The results appear similar to the following:



	CUSTOMER_ID	DATA_KNOWN	GENDER	AGE	TARIFF	DROPPED_CALLS	HANDSET	PEAK_MINS	PEAK_RAT
31781	Z288420	no		$n...		$null$		$null$	$nu
31782	Z325400	no		$n...		$null$		$null$	$nu
31783	Z325520	no		$n...		$null$		$null$	$nu
31784	Z327540	no		$n...		$null$		$null$	$nu
31785	Z329100	no		$n...		$null$		$null$	$nu
31786	Z344350	no		$n...		$null$		$null$	$nu
31787	Z345840	no		$n...		$null$		$null$	$nu
31788	Z359810	no		$n...		$null$		$null$	$nu
31789	Z385060	no		$n...		$null$		$null$	$nu

When DATA_KNOWN equals no, all values are empty or $null$. The $null$ value represents IBM SPSS Modeler's undefined value.

These records are not needed for modeling, so you will select only the customers whose data is known.

You will need to add a Select node downstream from the data source node. As demonstrated in the *Introduction to IBM SPSS Modeler* unit, the most efficient way is to generate the Select node from a Table output window. You will use this method to select the relevant records.

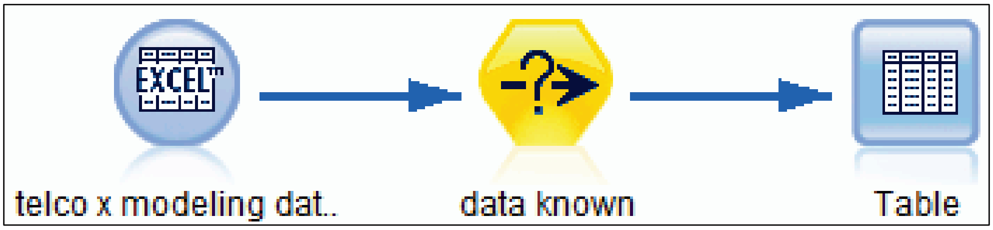
4. Scroll to the top of the **Table** output window, and click a **yes** value in the **DATA_KNOWN** column.
5. From the **Generate** menu, click **Select Node ("And")**.
6. Close the **Table** output window.

   A Select node, named (generated), is placed in the upper-left corner on the stream canvas.

7. Insert the **(generated)** node between the **Excel** source node and the **Table** node.
8. Edit the **Select** node named **(generated)**.

   The condition DATA_KNOWN = "yes" is generated. This will include only records with known data.
9. Click the **Annotations** tab.
10. Beside **Name**, click **Custom**, and then type **data known**.
11. Close the **Select** dialog box.

    The results appear similar to the following:



12. Run the **Table** node.

    The title bar in the Table output window informs you that there are 31,769 records left.
13. Close the **Table** output window.

Task 4. Build a model using historical data.

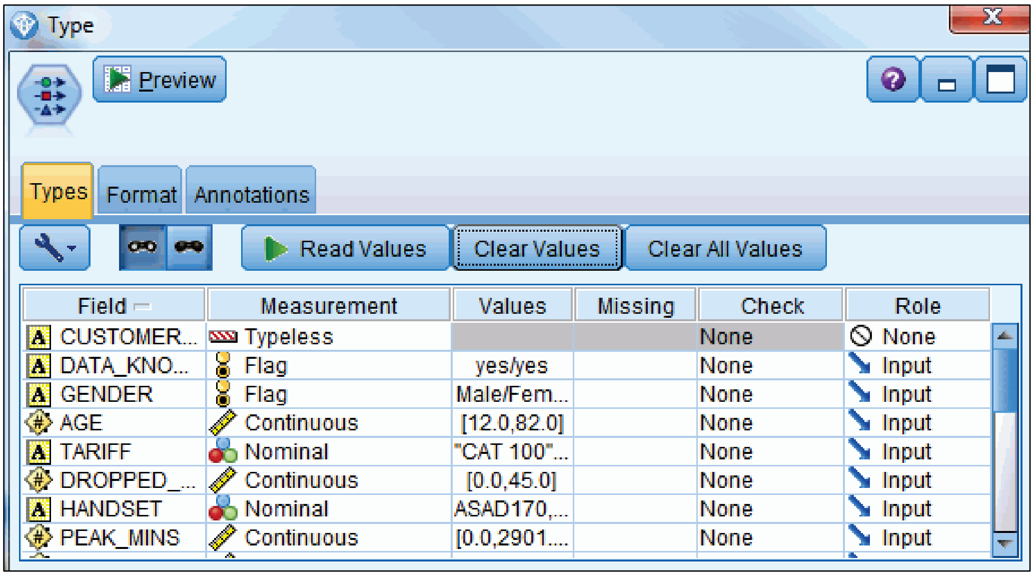
The next step is that you will use a model to predict churn. It is essential for modeling that predictors are selected and that the target field is specified. You will specify the predictors and the target in a Type node, in the Role column.

1. Make room by placing the Table node above the data known node, and then from the **Field Ops** palette, add a **Type** node downstream from the **Select** node named **data known**.
2. Edit the **Type** node.

   You can check the quality of the data by having IBM SPSS Modeler read the data.

3. Click **Read Values**.

   The results appear as follows:



Field	Measurement	Values	Missing	Check	Role
CUSTOMER...	Typeless			None	None
DATA_KNO...	Flag	yes/yes		None	Input
GENDER	Flag	Male/Fem...		None	Input
AGE	Continuous	[12.0,82.0]		None	Input
TARIFF	Nominal	"CAT 100"...		None	Input
DROPPED_...	Continuous	[0.0,45.0]		None	Input
HANDSET	Nominal	ASAD170,...		None	Input
PEAK_MINS	Continuous	[0.0,2901....		None	Input

   You can examine the Values column for out-of-range values. There appear to be no out-of-range values in this dataset.

   You will set the roles, used by the model in the next step. Fields with role Input will be used as predictors; the field that has to be predicted has role Target.

4. Ensure that the **Role** of **GENDER**, **AGE**, **TARIFF**, **DROPPED_CALLS**, and **HANDSET** is set to **Input**.

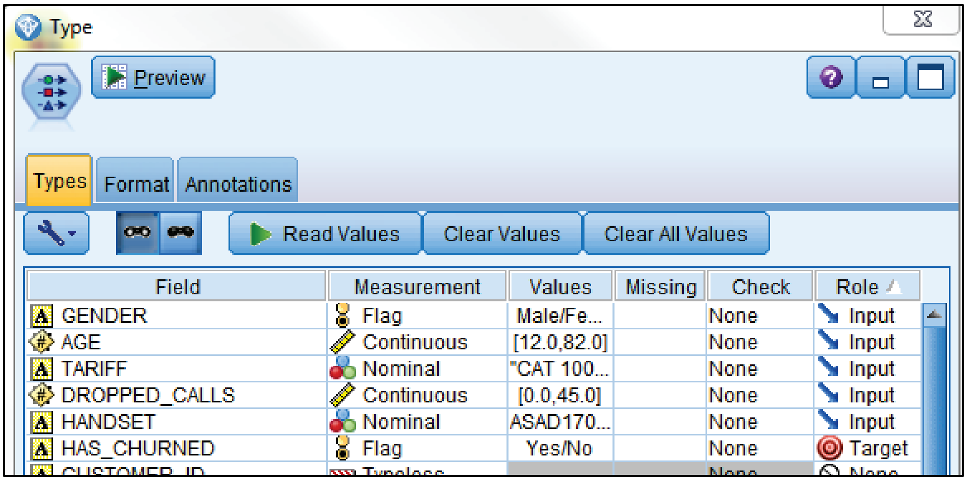
5. Set the **Role** of **HAS_CHURNED** to **Target**.

   Fields such as CUSTOMER_ID, DATA_KNOWN, and RETENTION are not relevant (recall, if you include retention you will get a 100% accurate but useless model); other fields are candidate-predictors but are excluded for the sake of a simpler model.

6. Set the **Role** for all other fields to **None**. (Use the Ctrl and Shift keys for a multiple selection; then click the cell in the Role column for one of the selected fields, and set Role to None.)

7. Click the **Role** column header twice, so that the fields are sorted according to their role.

   The results appear as follows:



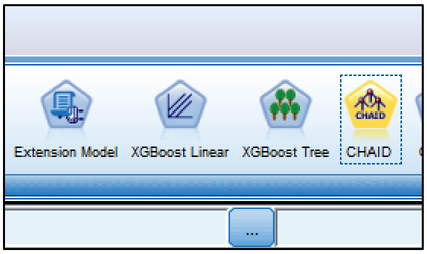
Field	Measurement	Values	Missing	Check	Role
A GENDER	Flag	Male/Fe...		None	Input
AGE	Continuous	[12.0,82.0]		None	Input
A TARIFF	Nominal	"CAT 100...		None	Input
DROPPED_CALLS	Continuous	[0.0,45.0]		None	Input
A HANDSET	Nominal	ASAD170...		None	Input
A HAS_CHURNED	Flag	Yes/No		None	Target
A CUSTOMER_ID	Typeless			None	None

8. Close the **Type** dialog box.

   Having specified predictors and target, the next step is to build a model, using one of the modeling nodes. CHAID will be used here.

9. From the **Modeling** palette, add a **CHAID** node downstream from the **Type** node.

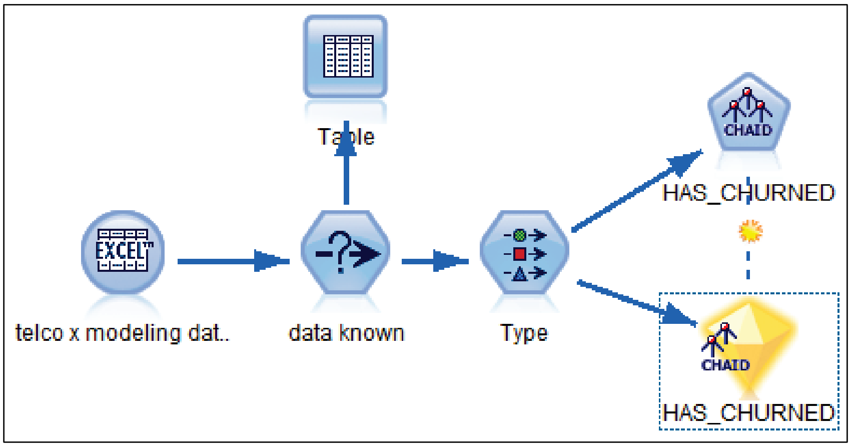
   Note: If you cannot locate the CHAID node, then, in the Modeling palette, click All at the left side, and scroll to the end right, similar to the following figure.



   The CHAID node is labeled with the target field, HAS_CHURNED.

10. Run the **CHAID** node.

    The results appear similar to the following:



    A model nugget is generated, added downstream from the Type node, and linked to the CHAID node. When you re-run the CHAID node (for example, with other predictors), the model nugget will be updated automatically because of the link.

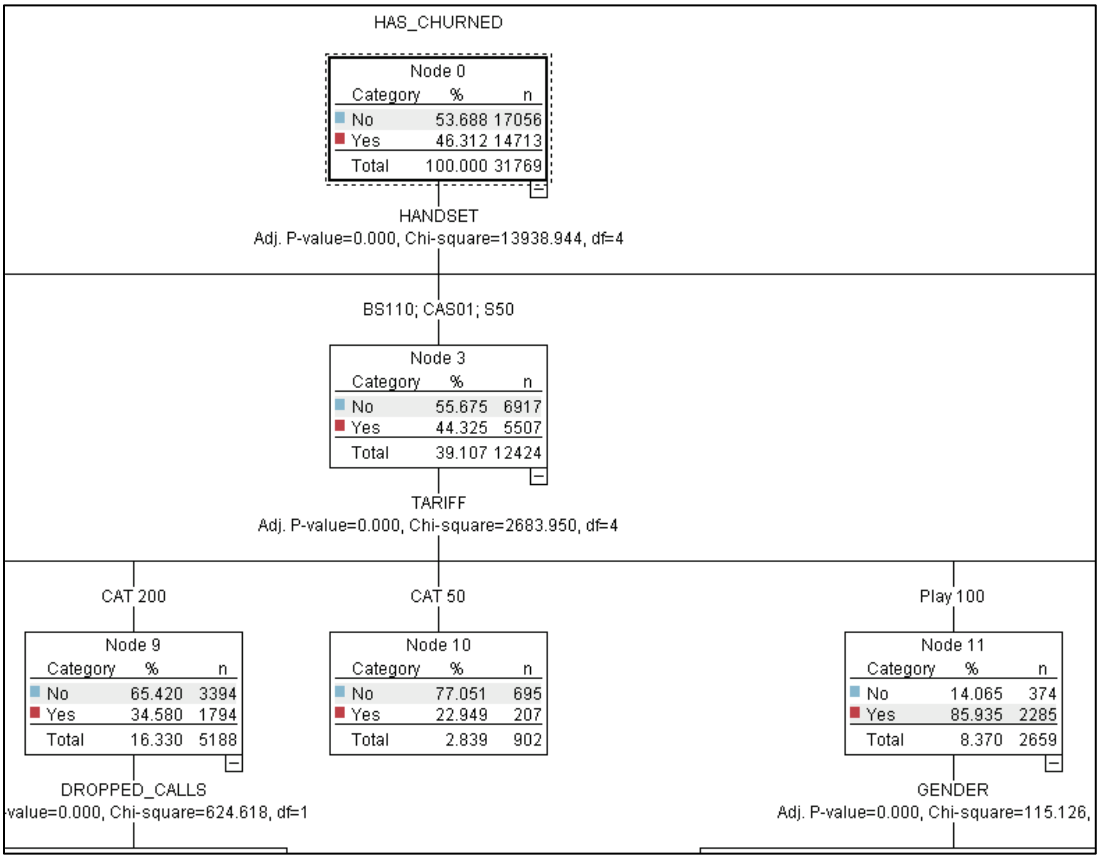
    The model nugget is also added to the Manager pane, Models tab.

    You will examine the tree in the tree viewer.

11. Edit the **CHAID** model nugget.
12. Click the **Viewer** tab.

13. Scroll to the right to display the root of the tree.

    The results appear similar to the following:



Node 0

Category	%	n
🟦 No	53.688	17056
🟥 Yes	46.312	14713
Total	100.000	31769

HANDSET
Adj. P-value=0.000, Chi-square=13938.944, df=4

Node 3

Category	%	n
🟦 No	55.675	6917
🟥 Yes	44.325	5507
Total	39.107	12424

TARIFF
Adj. P-value=0.000, Chi-square=2683.950, df=4

Node 9

Category	%	n
🟦 No	65.420	3394
🟥 Yes	34.580	1794
Total	16.330	5188

DROPPED_CALLS
-value=0.000, Chi-square=624.618, df=1

Node 10

Category	%	n
🟦 No	77.051	695
🟥 Yes	22.949	207
Total	2.839	902

Node 11

Category	%	n
🟦 No	14.065	374
🟥 Yes	85.935	2285
Total	8.370	2659

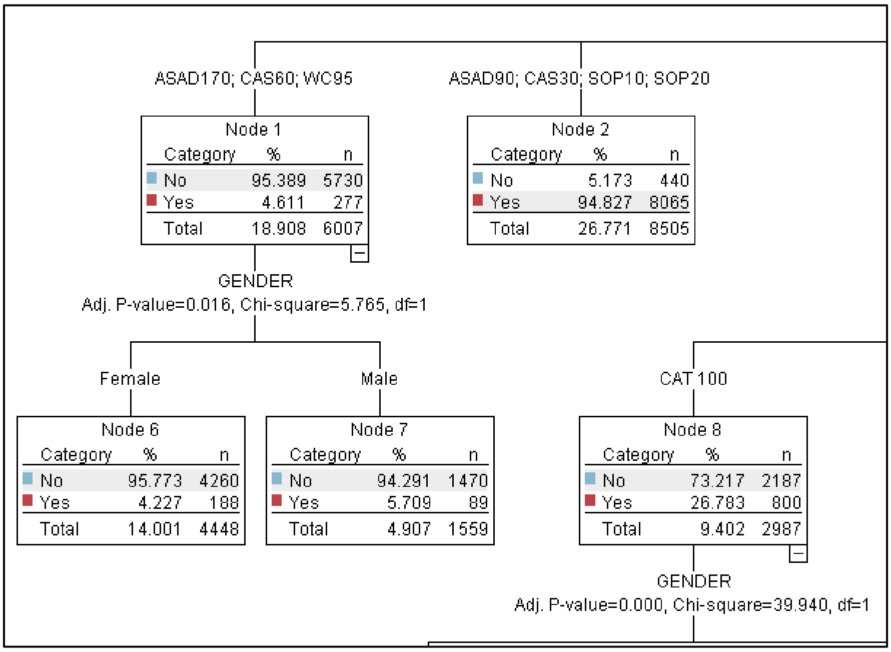
GENDER
Adj. P-value=0.000, Chi-square=115.126,

14,713 (46.3%) of the 31,769 customers have cancelled their subscription (the root node). The first split made by CHAID was on HANDSET, because that field had the most significant relationship with HAS_CHURNED. Some handsets were grouped (for example BS110, CAS01 and S50) by CHAID, because they had a similar churn rate. For the group of customers with handsets BS110, CAS01, or S50, there is a further split on TARIFF. Customers in the Play100 tariff showed the highest churn rate (85.935%). This group was split on GENDER next. Thus, within the group of customers with HANDSET BS110, CAS01 or S50 and TARIFF = Play100, there was a difference between men and women in churn rate.

You will navigate the tree to find groups with a high churn rate.

14. Scroll to the left to display the result for handsets **ASAD90**, **CAS30**, **SOP10** and **SOP20**.

The results appear similar to the following:


ASAD170; CAS60; WC95
ASAD90; CAS30; SOP10; SOP20
GENDER
Adj. P-value=0.016, Chi-square=5.765, df=1
Female
Male
CAT 100
GENDER
Adj. P-value=0.000, Chi-square=39.940, df=1

Node 1

	Category	%	n
🟦	No	95.389	5730
🟥	Yes	4.611	277
	Total	18.908	6007

Node 2

	Category	%	n
🟦	No	5.173	440
🟥	Yes	94.827	8065
	Total	26.771	8505

Node 6

	Category	%	n
🟦	No	95.773	4260
🟥	Yes	4.227	188
	Total	14.001	4448

Node 7

	Category	%	n
🟦	No	94.291	1470
🟥	Yes	5.709	89
	Total	4.907	1559

Node 8

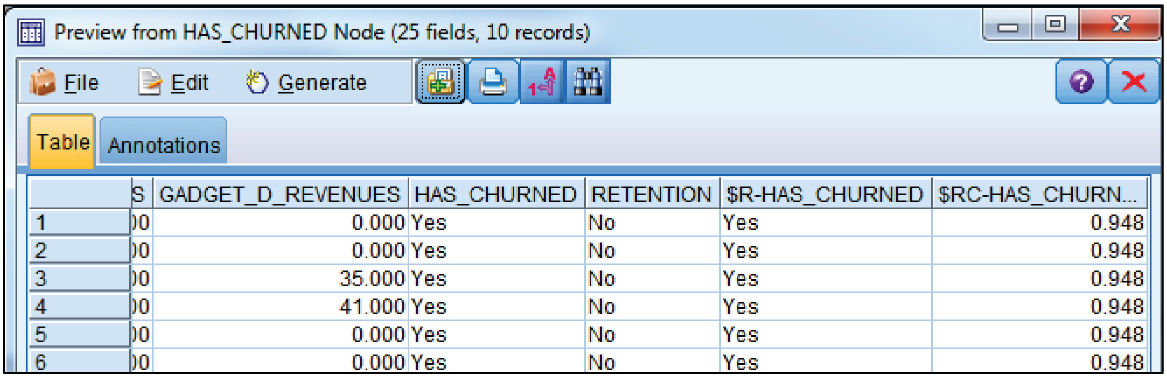
	Category	%	n
🟦	No	73.217	2187
🟥	Yes	26.783	800
	Total	9.402	2987

CHAID merged customers with handsets ASAD90, CAS30, SOP10 and SOP20, because they had similar churn rates. The churn percentage within this group was 94.827%.

The model nugget stores the rules, corresponding to the terminal nodes of the tree, and the model nugget will add two fields to the dataset. You will preview the data in the model nugget to examine these fields.

15. Click **Preview**, and then scroll to the far right.

    The results appear similar to following:



Preview from HAS_CHURNED Node (25 fields, 10 records)

	S	GADGET_D_REVENUES	HAS_CHURNED	RETENTION	$R-HAS_CHURNED	$RC-HAS_CHURN...
1	)0	0.000	Yes	No	Yes	0.948
2	)0	0.000	Yes	No	Yes	0.948
3	)0	35.000	Yes	No	Yes	0.948
4	)0	41.000	Yes	No	Yes	0.948
5	)0	0.000	Yes	No	Yes	0.948
6	)0	0.000	Yes	No	Yes	0.948

$R-HAS_CHURNED stores the predicted category for each customer. The predicted category is the most frequent category in the node to which the customer belongs. For example, a customer with handset ASAD90 belongs to the node ASAD90, CAS30, SOP10, SOP20, and 94.827% of customers in this node have churned. Therefore, the predicted category for a customer with handset ASAD90 is Yes.

$RC-HAS_CHURNED stores the confidence for the prediction. For example, in the group customers with handsets ASAD90, CAS30, SOP10 or SOP20, 94.827% of the customers have churned, so the confidence that the predicted category (Yes) is correct equals 0.94827. Another interpretation is that of probability: the probability that a customer with one of the handsets ASAD90, CAS30, SOP10 or SOP20 has churned equals 0.94827.

$RC-HAS_CHURNED cannot be equated with the "probability to cancel the subscription", because when the predicted category is "No", $RC-HAS_CHURNED actually stores the probability to be still with the company.

Notice that the first customers are predicted correctly (actual churn and predicted churn are the same).

16. Close the **Preview** output window.
17. Close the **Tree** viewer window.

Task 5. Score current customers.

Having found a model and assuming that the model is satisfactory, the model can be applied to the current customers.

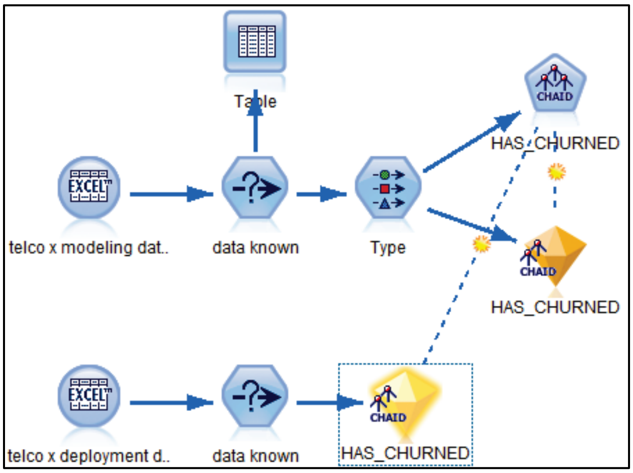
The telco x deployment data.xlsx stores the data for the current customers.

1. From the **Sources** palette, add an **Excel** source node to the stream canvas.
2. Edit the **Excel** node, and then set **Import file** to **C:\Training\0A008\telco x deployment data.xlsx**.
3. Click **Preview**.

   Data is unknown for some records, as indicated by the DATA_KNOWN field. You will only include records with known data.
4. Close the **Preview** output window.
5. Close the dialog box.

   Rather than generating the Select node, you can copy it from what you already have.
6. Right-click the **Select** node named **data known** and select **Copy Node**.
7. Right-click an empty area on the stream canvas, and then click **Paste**.
8. Add the pasted node downstream from the **Excel** data source node named **telco x deployment data.xlsx**.
9. Copy and paste the **CHAID** model nugget, and add it downstream from the **data known** Select node.

   The results appear as follows:



10. From the **Output** palette, add a **Table** node downstream from the **model nugget** named **HAS_CHURNED** in the lower part of the stream, and then run the **Table** node.

11. Scroll to the far right in the **Table** output window.

    The results appear similar to the following:

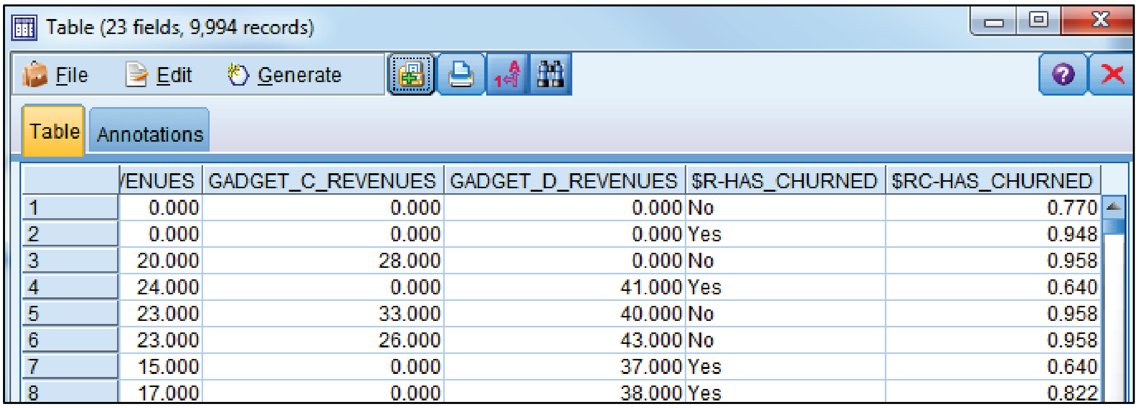


Table (23 fields, 9,994 records)

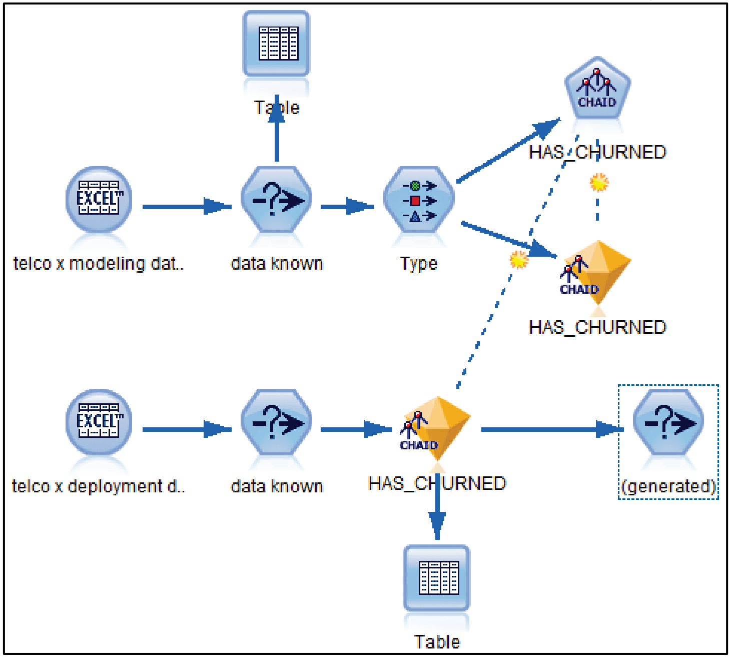
	VENUES	GADGET_C_REVENUES	GADGET_D_REVENUES	$R-HAS_CHURNED	$RC-HAS_CHURNED
1	0.000	0.000	0.000	No	0.770
2	0.000	0.000	0.000	Yes	0.948
3	20.000	28.000	0.000	No	0.958
4	24.000	0.000	41.000	Yes	0.640
5	23.000	33.000	40.000	No	0.958
6	23.000	26.000	43.000	No	0.958
7	15.000	0.000	37.000	Yes	0.640
8	17.000	0.000	38.000	Yes	0.822

Two fields are added: the predicted category and the confidence for that prediction. For example, the first customer is predicted to stay, with a probability of 0.770. The second customer is predicted to leave, with a probability of 0.948.

You will export the data for the customers who are likely to leave to a text file, where "likely" means that a customer has a greater than 0.9 probability to leave.

12. In the **Table** output window, Ctrl-click **Yes** in the **$R-HAS_CHURNED** column and **0.948** in the **$RC-HAS_CHURNED**, click the **Generate** menu, and then click **Select Node ("And")**.
13. Close the **Table** output window.
14. Add the generated **Select** node, located in the left upper corner on the stream canvas, downstream from the **CHAID** model nugget, in the lower part of the stream.

    The results appear similar to the following:



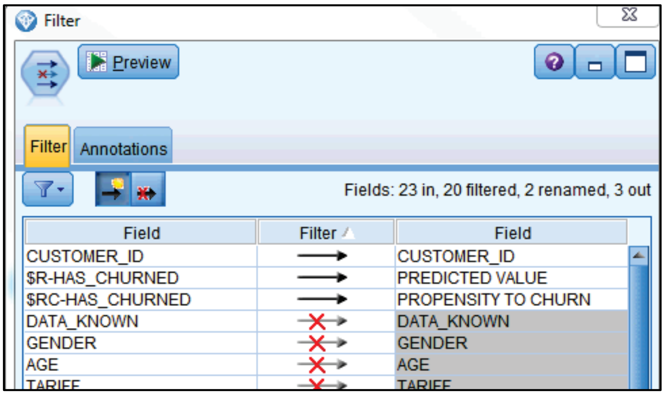
15. Edit the **Select** node named **(generated)**.
16. On the **Settings** tab, replace the expression **'$R-HAS_CHURNED' = "Yes" and '$RC-HAS_CHURNED' = 0.948160338544728** by **'$R-HAS_CHURNED' = "Yes" and '$RC-HAS_CHURNED' > 0.9**.
17. Click the **Annotations** tab, select **Custom**, and rename the node to **customers@risk**.
18. Close the dialog box.

    Before exporting the data, rename the fields that are added by the model, and ensure that only these fields together with customer_id are exported.

    You will rename and remove fields in a Filter node.
19. From the **Field Ops** palette, add a **Filter** node downstream from the **Select** node named **customers@risk**. (Ensure you use the Filter node, not the Filler node.)
20. Edit the **Filter** node, click the **Filter options menu** ▽▾ button, and then click **Remove All Fields**.
21. In the **CUSTOMER_ID** row, click the crossed arrow ✕→ once, so it turns into an arrow →.
22. Repeat the previous step for **$R-HAS_CHURNED** and **$RC-HAS_CHURNED**.
23. Rename**$R-HAS_CHURNED** to **PREDICTED VALUE**.

    Because you selected those who are predicted to churn, the confidence column now represents the propensity to churn.
24. Rename **$RC-HAS_CHURNED** to **PROPENSITY TO CHURN**.
25. Click the **Filter** column header twice.

    The results appear as follows:



Fields: 23 in, 20 filtered, 2 renamed, 3 out

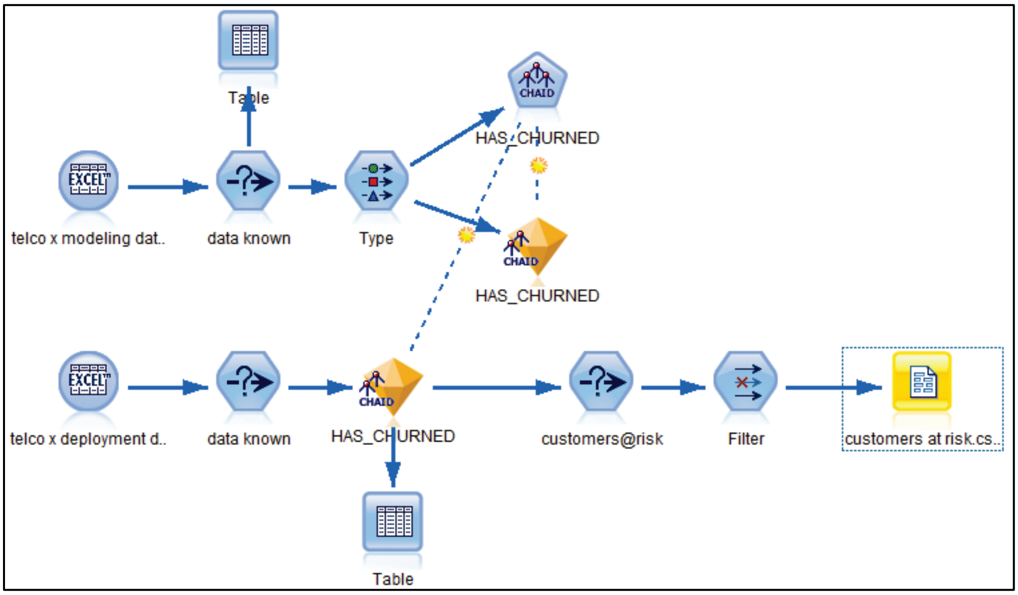
Field	Filter	Field
CUSTOMER_ID	→	CUSTOMER_ID
$R-HAS_CHURNED	→	PREDICTED VALUE
$RC-HAS_CHURNED	→	PROPENSITY TO CHURN
DATA_KNOWN	✕→	DATA_KNOWN
GENDER	✕→	GENDER
AGE	✕→	AGE
TARIFF	✕→	TARIFF

26. Close the **Filter** dialog box.

    You will export these three fields for the selected to a text file. The Flat File node will export to a comma separated text file.

27. From the **Export** palette, add a **Flat File** node downstream from the **Filter** node.
28. Edit the **Flat File** node, and then in **Export file**, type **customers at risk.csv**.
29. Close the **Flat File** node.

    The results appear similar to the following:



30. Run the **Flat File** node named **customers at risk.csv**.

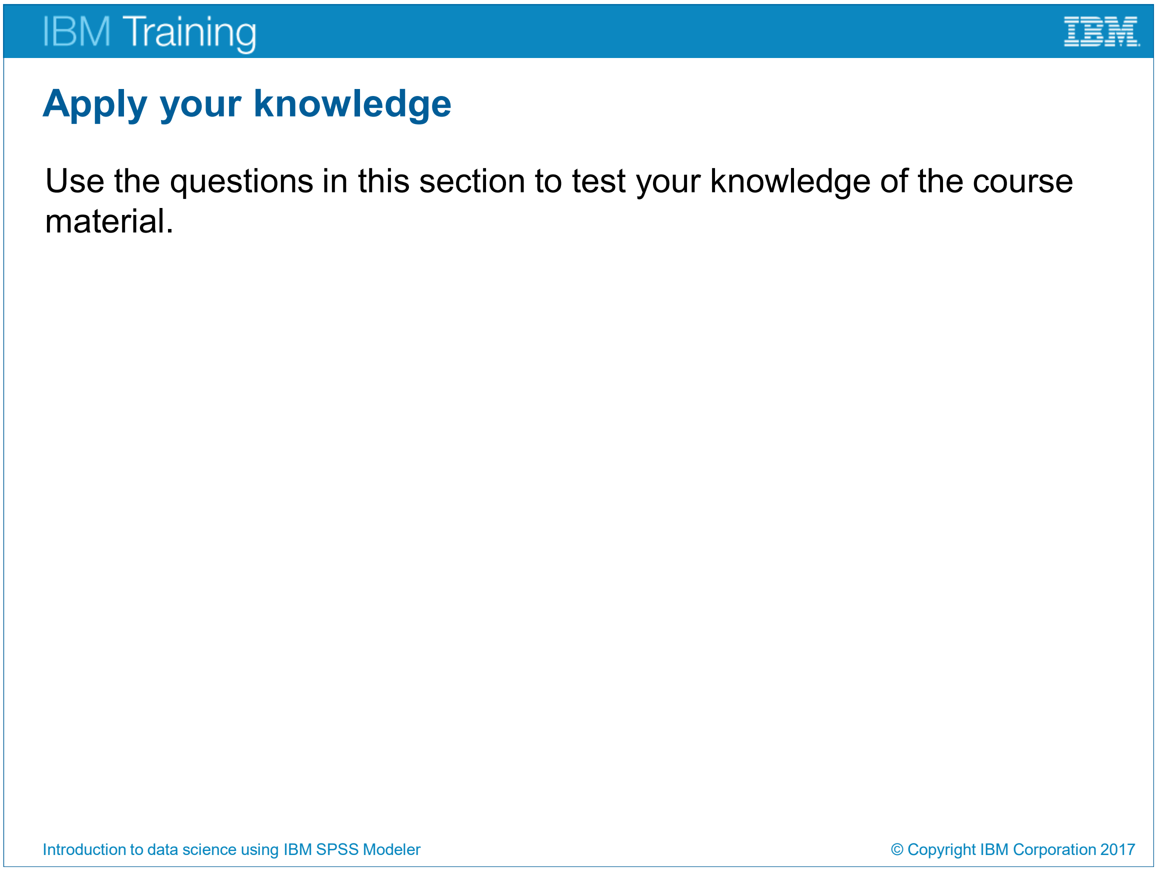
    The data for customers at risk are exported to a text file. Other departments within the organization can take it from there.

    This completes the demonstration for this unit. You will create a clean state for the exercise.
31. From the **File** menu, click **Close Stream**, and then click **No** when asked to save the stream.
32. From the **File** menu, click **New Stream**.

    Leave IBM SPSS Modeler open for the exercise.

Results:
Using historical data, you built a model to predict churn, and applied the model to current customers.

You will find the completed stream in the **C:\Training\0A008\03-Introduction_to_data_science_using_IBM_SPSS_Modeler\Solutions** folder.



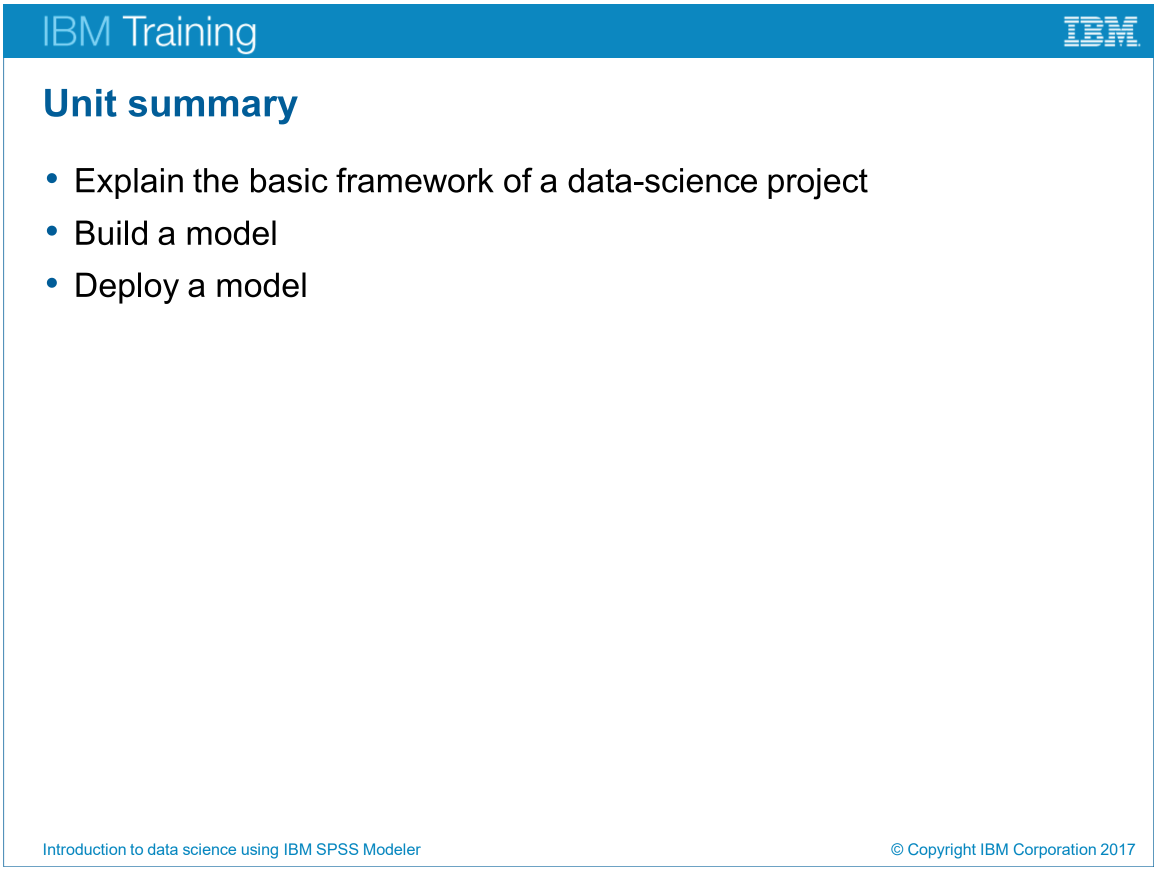
IBM Training

Apply your knowledge

Use the questions in this section to test your knowledge of the course material.

Introduction to data science using IBM SPSS Modeler

Apply your knowledge



IBM Training

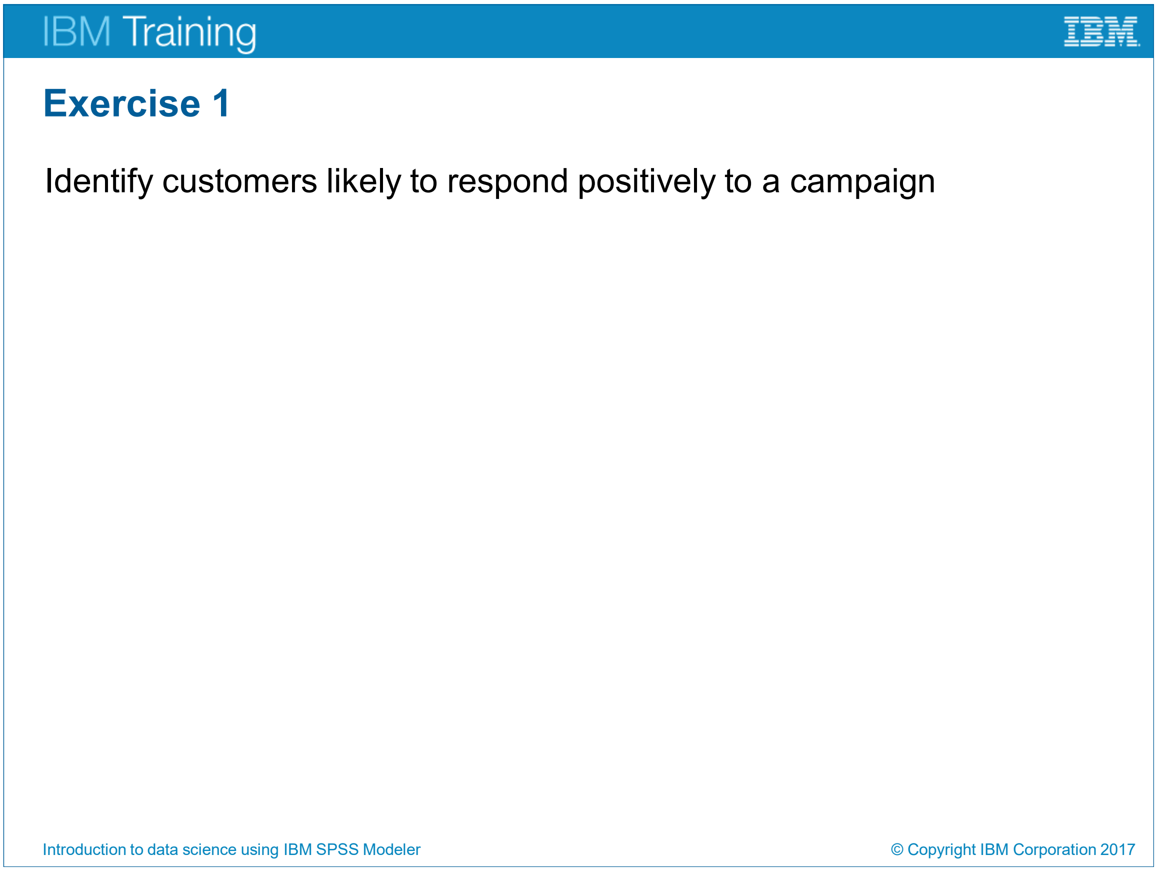
Unit summary

- Explain the basic framework of a data-science project
- Build a model
- Deploy a model

Introduction to data science using IBM SPSS Modeler

© Copyright IBM Corporation 2017

Unit summary



IBM Training

Exercise 1

Identify customers likely to respond positively to a campaign

Introduction to data science using IBM SPSS Modeler

© Copyright IBM Corporation 2017

Exercise 1: Identify customers likely to respond positively to a campaign

Exercise 1: Identify customers likely to respond positively to a campaign

Data file:	**ACME customer and rfm data.csv**
Data folder:	**C:\Training\0A008**

This exercise is about ACME, a company that sells sports products. ACME wants to promote a new product, the XL Original Orange Baseball Cap.

ACME has sent out a test mailing (by e-mail) to 10,000 randomly selected customers, and recorded the response (who purchased the XL Original Orange Baseball Cap, and who did not).

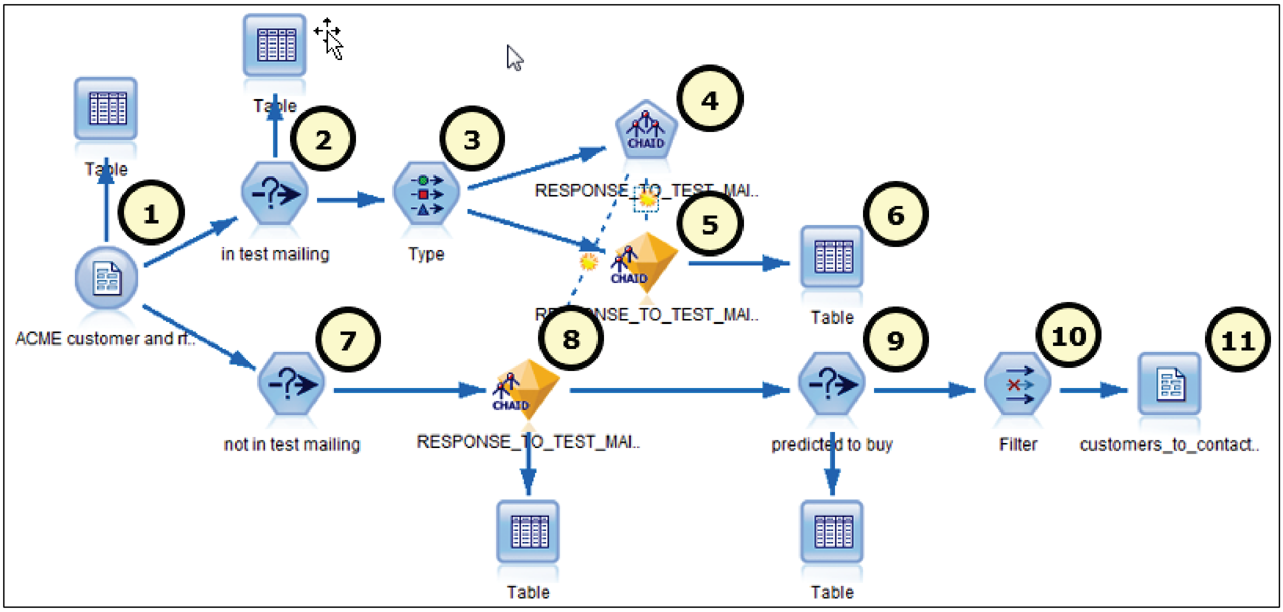
In this exercise, you will build a model using data of the test mailing. The model will (hopefully) identify groups with high response rates. You will then use the model to select the groups with high response rates in the rest of the customer database (only these groups will be included in the actual mailing for the XL Original Orange Baseball Cap).

Data is available in ACME customer and rfm data.csv and includes the following fields:

Field	**Field Description**
CUSTOMER_ID	customer's identification number
GENDER	customer's gender
EMAIL_ADDRESS	customer's e-mail
POSTAL_CODE	customer's postal code
RECENCY_BEFORE_FEB-01-2011	customer's last order date, before FEB-01-2011
FREQUENCY_BEFORE_FEB-01-2011	customer's number of orders, before FEB-01-2011
MONETARY_VALUE_BEFORE_FEB-01-2011	customer's total purchase amount, before FEB-01-2011
HAS_RECEIVED_TEST_MAILING_FEB-01-2011	flags whether the customer was in the test mailing, sent out FEB-01-2011

Field	Field Description
RESPONSE_TO_TEST_MAILING_FEB-01-2011	For customers in the test mailing, this field flags whether the customer ordered the XL Original Orange Baseball Cap; for customers not in the test mailing, this field is undefined.
ORDER_DATE	The date that the XL Original Orange Baseball Cap was ordered; this field is only valid for customers in the test mailing who ordered the XL Original Orange Baseball Cap; for customers in the test mailing who did not order the XL Original Orange Baseball Cap, and for those not in the test mailing, this field is undefined.
NUMBER_OF_DAYS_BETWEEN_TEST_MAILING_AND_ORDER_DATE	The number of days between the test mailing (FEB-01-2011) and the order date; this field is only valid for customers in the test mailing who ordered the XL Original Orange Baseball Cap; the field is undefined for customers in the test mailing who did not order the XL Original Orange Baseball Cap and for those not included in the test mailing.
ORDERED_WITHIN_MONTH	Flags whether the order date was within one month after the test mailing went out; this field is only valid for those in the test mailing who ordered the XL Original Orange Baseball Cap; the field is undefined for those in the test mailing who did not order the XL Original Orange Baseball Cap and for those not included in the test mailing.

In this exercise, you will create the following stream:



The upper branch creates the model. The lower branch applies the model to those who were not included in the test mailing, and exports the data of those who are likely to respond positively to the campaign to a text file.

The numbers in the previous figure correspond to the tasks that follow.

- From the **C:\Training\0A008** folder, import the data from **ACME customer and rfm data.csv** (a comma-separated text file; use the **Var. File** node to import the data, using the default settings for the import). Run a **Table** node to get familiar with the data.

  How many records do you have? And how many fields?

- A model will be built (in a next task) to predict the response on the test mailing. Select all those customers that were included in the test mailing. (The field **HAS_RECEIVED_TEST_MAILING_FEB-01-2011** flags whether a customer was included in the test mailing. Recall that the most efficient method to select records is to generate a Select node from the Table output window.)

  How many customers were included in the test mailing?

- Add a **Type** node (**Field Ops** palette) downstream from the **Select** node that you added in the previous task. Edit the **Type** node and then set field roles, to predict the target field **RESPONSE_TO_TEST_MAILING_FEB-01-2011**, using only **GENDER**, **RECENCY_BEFORE_FEB-01-2011**, **FREQUENCY_BEFORE_FEB-01-2011**, and **MONETARY_VALUE_BEFORE_FEB-01-2011** as inputs (predictors). Also, in the **Type** node, click the **Read Values** button to examine the values.

- Use **CHAID** to predict the target field (the **CHAID** node is located on the **Modeling** palette; after clicking the **Modeling** palette, click **All** at the left side and then locate the **CHAID** node).
- After you have executed the **CHAID** node, edit the generated **model nugget**, and then click the **Viewer** tab to examine the model.

  Which field is used as the first split field (the field under the root of the tree)? Which group shows the highest response rate, and what is the probability to respond positively for this group? (You can use the **Show or hide the tree map window** to browse the tree.)
- Run a **Table** node downstream from the **model nugget**.

  Which two fields are added to the data, and what is their interpretation?
- Select the customers that were not in the test mailing (Hint: copy the Select node that you already have on the stream canvas, and edit that.)
- Apply the model to the customers who were not in the test mailing.
- From the customers not included in the test mailing, select the customers that are expected to respond positively to the campaign (the predicted value for RESPONSE_TO_TEST_MAILING_FEB-01-2011, stored in the $R-RESPONSE_TO_TEST_MAILING_FEB-01-2011 field, equals T).

  How many customers are selected?
- Keep only the following three fields: **CUSTOMER_ID**, **$R-RESPONSE_TO_TEST_MAILING_FEB-01-2011** and **$RC-RESPONSE_TO_TEST_MAILING_FEB-01-2011**.Rename the latter two fields to **PREDICTED CATEGORY** and **CONFIDENCE FOR PREDICTION**, respectively. Recall that you can select fields and rename fields in a **Filter** node (**Field Ops** palette; ensure you choose the Filter node, not the Filler node).
- Use a **Flat File** node to export the data for the selected customers (customers expected to respond positively to the campaign) to a text file named **customers_to_contact.csv**.
- Exit IBM SPSS Modeler without saving anything.

For more information about where to work and the exercise results, refer to the Tasks and Results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1:
Tasks and Results

Task 1. Import and examine the data.

- Use the **Var. File** node (**Sources** palette) to import data from **ACME customer and rfm data.csv**, located in the **C:\Training\0A008** folder.
- Add a **Table** node (**Output** palette) downstream from the **Var. File** node.
- Run the **Table** node.

  The title bar of the Table output windows informs you that the dataset includes 12 fields and 30,000 records.

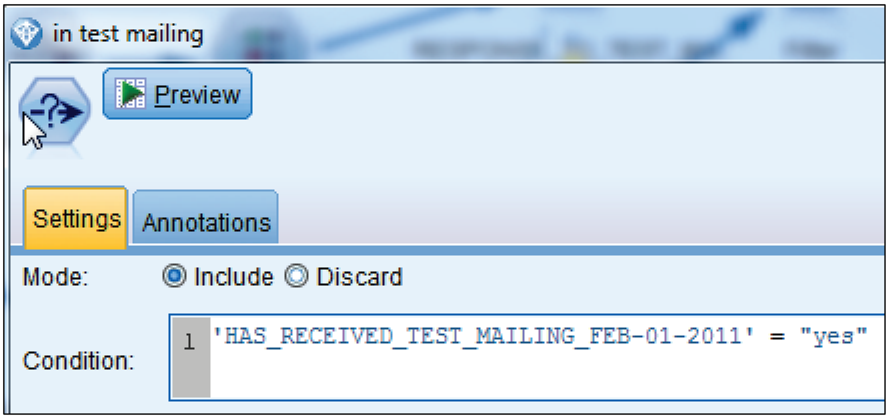
Task 2. Select the data to build the model.

- Response is only known for customers who were included in the test mailing. You need a **Select** node to select them. The **Select** node is placed downstream from the **Var. File** node.

  It is recommended to generate this **Select** node from the **Table** output window as demonstrated in unit 2 in this course. If preferred, you can rename the generated node to make clear which records are selected. An example name would be **in test mailing**. When you have generated the **Select** node from the **Table** output window, ensure that you add it downstream from the **Var. File** node.

  Alternatively, click the **Record Ops** palette, and add a **Select** node downstream from the **Var. File** node. Then, edit the **Select** node, and enter the condition: **'HAS_RECEIVED_TEST_MAILING_FEB-01-2011' = "yes"**.

  The Select dialog box appears as follows:



Mode: ● Include ○ Discard

Condition:

```
1 'HAS_RECEIVED_TEST_MAILING_FEB-01-2011' = "yes"
```

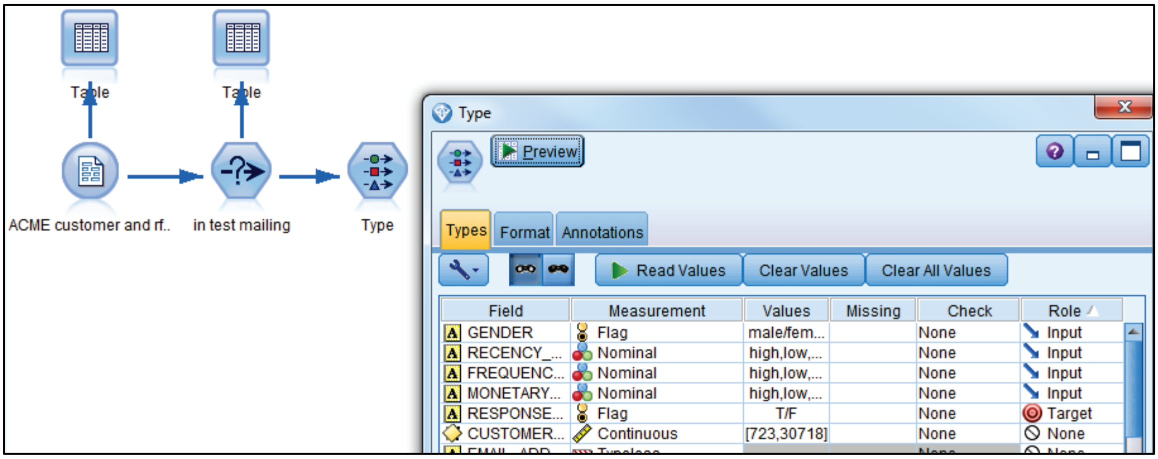
- Add a **Table** node (**Output** palette) downstream from the **Select** node.
- Run the **Table** node.

  10,000 records were included in the test mailing.

Task 3. Set field roles and read the data.

- Add a **Type** node (**Field Ops** palette) downstream from the **Select** node that you added in the previous task.
- Edit the **Type** node and then:
  - Click the cell in the **RESPONSE_TO_TEST_MAILING_FEB-01-2011** row, **Role** column, and set role to **Target**.
  - Ensure that the role of **GENDER**, **RECENCY_BEFORE_FEB-01-2011**, **FREQUENCY_BEFORE_FEB-01-2011**, and **MONETARY_VALUE_BEFORE_FEB-01-2011** is **Input**.
  - Ensure that the role for the other fields is **None**.
- Click the **Read Values** button to read the data.
  - You can sort the Role column by double-clicking the Role column header.

    The results appear similar to the following:

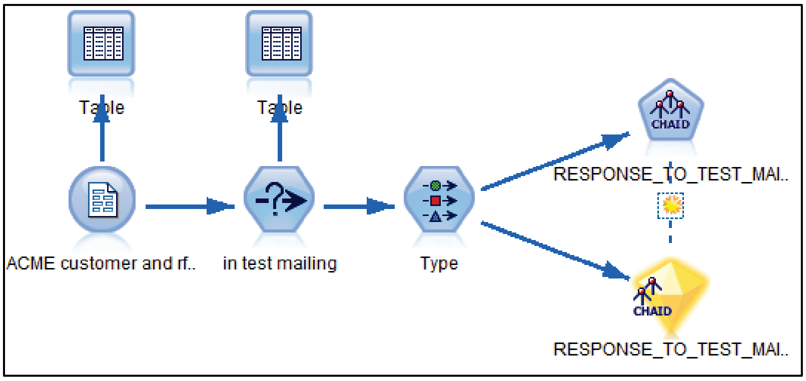


Field	Measurement	Values	Missing	Check	Role
GENDER	Flag	male/fem...		None	Input
RECENCY_...	Nominal	high,low,...		None	Input
FREQUENC...	Nominal	high,low,...		None	Input
MONETARY...	Nominal	high,low,...		None	Input
RESPONSE...	Flag	T/F		None	Target
CUSTOMER...	Continuous	[723,30718]		None	None
EMAIL_ADD...	Typeless			None	None

Task 4. Build a CHAID model.

- Add a **CHAID** node (**Modeling** palette) downstream from the **Type** node. The CHAID node will be labeled with the name of the target.
- Run the **CHAID** node.

  The results appear similar to the following:



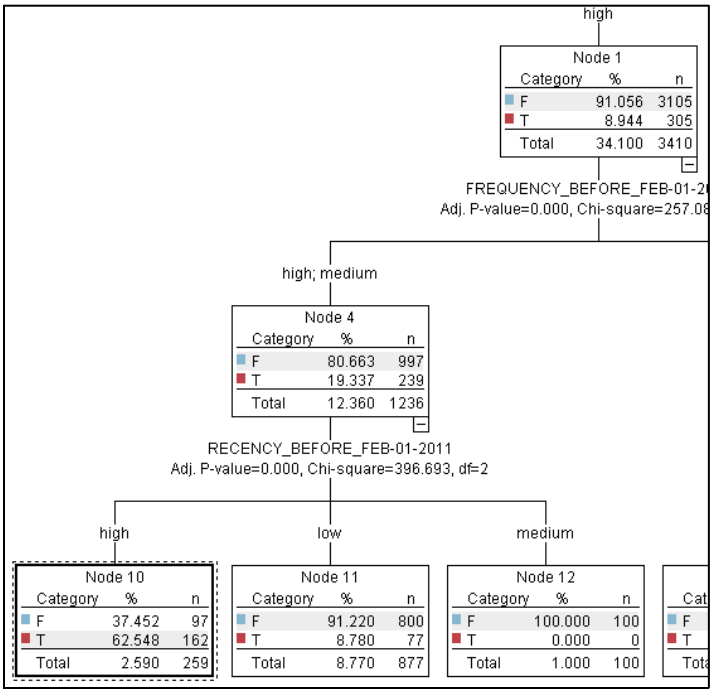
  A model nugget was automatically added downstream from the Type node.

Task 5. Examine the model.

- Edit the **model nugget**, and then click the **Viewer** tab. The first split field is MONETARY_VALUE_BEFORE_FEB-01-2011.

- Use the **Show or hide the tree map window** button to navigate the tree and locate the group with the highest response rate.

  The results appear similar to the following:



high

Node 1

Category	%	n
🟦 F	91.056	3105
🟥 T	8.944	305
Total	34.100	3410

FREQUENCY_BEFORE_FEB-01-2
Adj. P-value=0.000, Chi-square=257.08

high; medium

Node 4

Category	%	n
🟦 F	80.663	997
🟥 T	19.337	239
Total	12.360	1236

RECENCY_BEFORE_FEB-01-2011
Adj. P-value=0.000, Chi-square=396.693, df=2

high

Node 10

Category	%	n
🟦 F	37.452	97
🟥 T	62.548	162
Total	2.590	259

low

Node 11

Category	%	n
🟦 F	91.220	800
🟥 T	8.780	77
Total	8.770	877

medium

Node 12

Category	%	n
🟦 F	100.000	100
🟥 T	0.000	0
Total	1.000	100

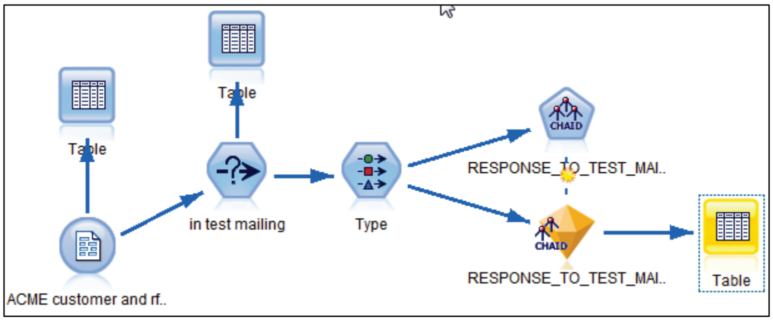
Cat
🟦 F
🟥 T
Tota

  Customers with high monetary value, medium or high frequency, and high recency have a 0.625 probability to purchase the XL Original Orange Baseball Cap.

Task 6. Interpret the fields added by the model nugget.

- Add a **Table** node (**Output** palette) downstream from the model nugget.

  The results appear similar to the following:



- Run the **Table** node.

- Scroll to the last fields in the **Table** output window.

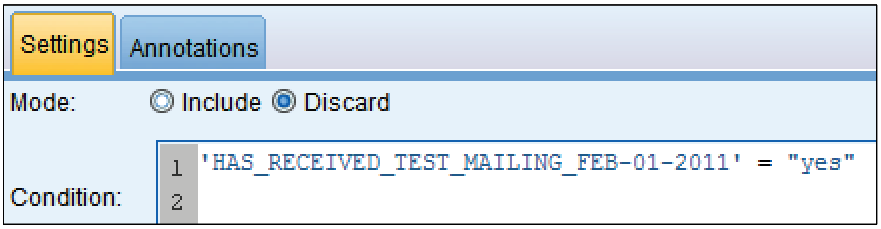
  $R-RESPONSE_TO_TEST_MAILING_FEB-01-2011 stores the predicted response, $RC-RESPONSE_TO_TEST_MAILING_FEB-01-2011 the confidence for the prediction. For example, the first record is predicted to not purchase the XL Original Orange Baseball Cap. The confidence for this prediction equals 0.953. Record #10 is predicted to purchase the XL Original Orange Baseball Cap. The confidence for the prediction is 0.625 (this customer belongs to the group with the highest response rate).

Task 7. Select the customers not included in the test mailing.

- The model has to be applied to customers that were not included in the test mailing. Therefore, a **Select** node that selects this group must be added to the stream, downstream from the **Var. File** node

  You can create the Select node in several ways. You can generate the Select node from a Table output window (demonstrated in the previous unit).

  Alternatively, right-click the Select node that you have already on the stream canvas, click Copy Node, right-click an empty area on the stream canvas, and then click Paste. Add the pasted Select node downstream from the Var. File node, and edit the expression. Or, rather than editing the expression, keep the expression as it is and change the mode from **Include** to **Discard**. In this case, the results appear as follows:



Mode: ○ Include ● Discard

Condition:

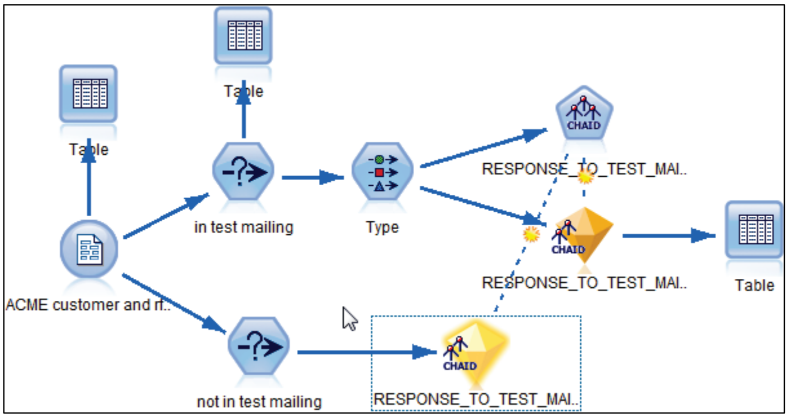
```
'HAS_RECEIVED_TEST_MAILING_FEB-01-2011' = "yes"
```

  It is recommended to rename the **Select** node, for example to **not in test mailing**. You can rename a node on the **Annotations** tab.

Task 8. Apply the model to customers not included in the test mailing.

- Applying a model to a dataset means that records are passed through the model nugget because the model contains the model rules. Therefore, copy the **model nugget** that you already have on the stream canvas, paste it, and then add it downstream from the **Select** node (named **not in test mailing** in the following figure).

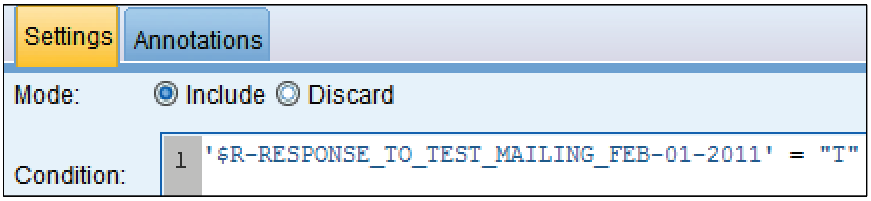
  The results appear similar to the following:



Task 9. Select customers predicted to respond positively.

- You need a Select node to select these records. The most efficient way is to generate this Select node from a Table output window. Then, add the generated Select node downstream from the model nugget. Rename the Select node, if preferred, for example **predicted to buy**.

  The Select dialog box appears as follows:



Mode: ● Include ○ Discard

Condition:

```
'$R-RESPONSE_TO_TEST_MAILING_FEB-01-2011' = "T"
```

- Add a **Table** node (**Output** palette) downstream from the **Select** node.
- Run the **Table** node to get the record count.

  254 customers are expected to respond positively to the campaign.

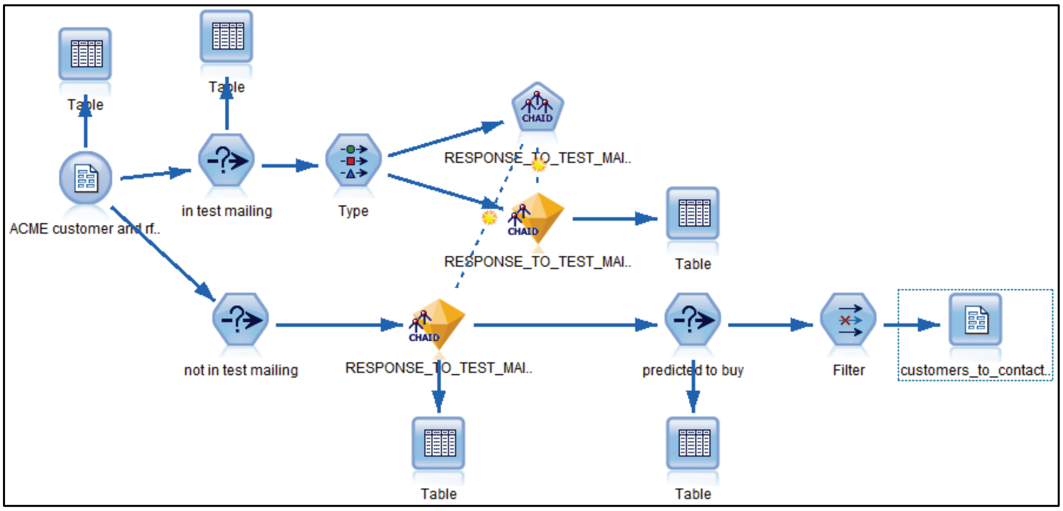
Task 10. Filter out fields and rename fields.

- Add a **Filter** node (**Field Ops** palette) downstream from the **Select** node.
- Edit the **Filter** node, and then deselect all fields except **CUSTOMER_ID**, **$R-RESPONSE_TO_TEST_MAILING_FEB-01-2011**, and **$RC-RESPONSE_TO_TEST_MAILING_FEB-01-2011**. Rename the last two fields into **PREDICTED CATEGORY** and **CONFIDENCE FOR PREDICTION**, respectively.

Task 11. Export the results.

- Add a **Flat File** node (**Export** palette) downstream from the **Filter** node, edit the node, and then specify the file name, **customers_to_contact.csv**.

  The results appear similar to the following:



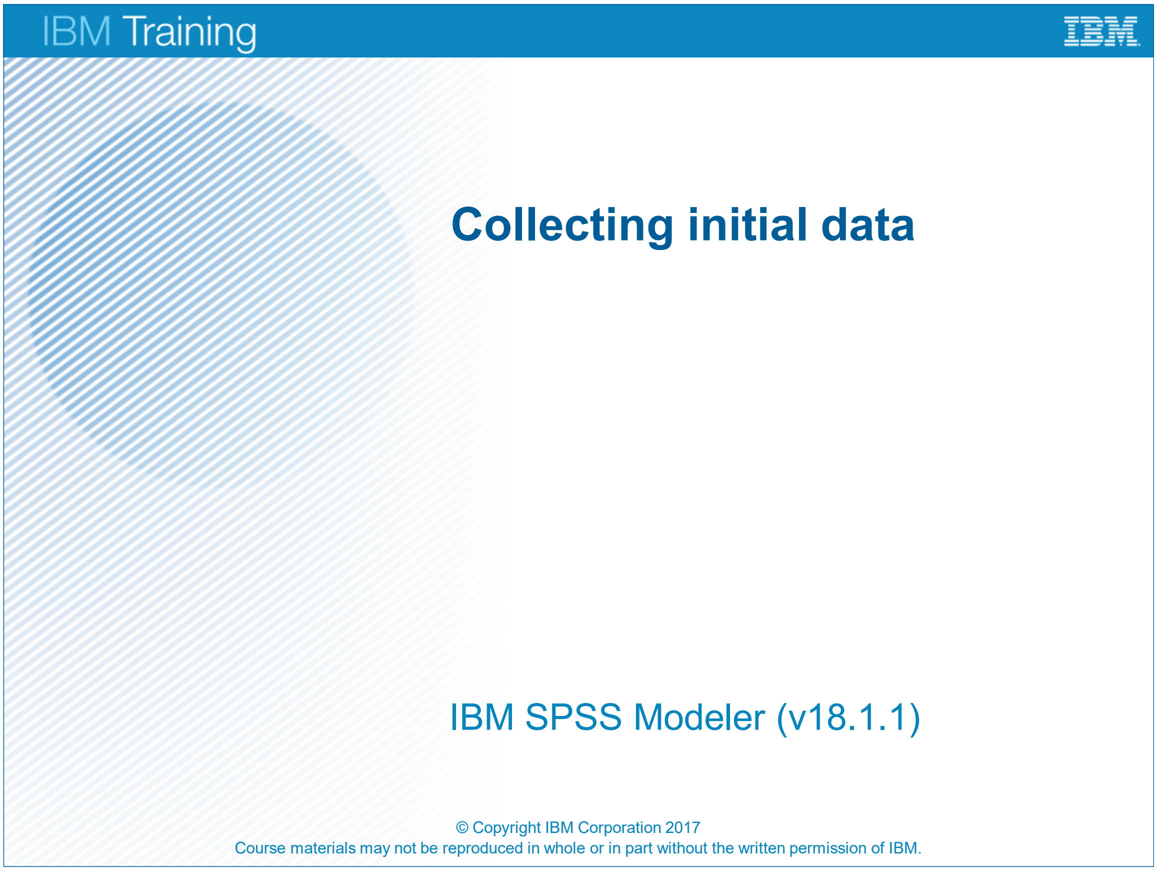
- Run the **Flat File** node to export the data.

Task 12. Exit IBM SPSS Modeler.

- From the **File** menu, click **Exit**, and then exit IBM SPSS Modeler without saving anything.

You will find the solution results in the **C:\Training\0A008\03-Introduction_to_data_science_using_IBM_SPSS_Modeler\Solutions** folder.

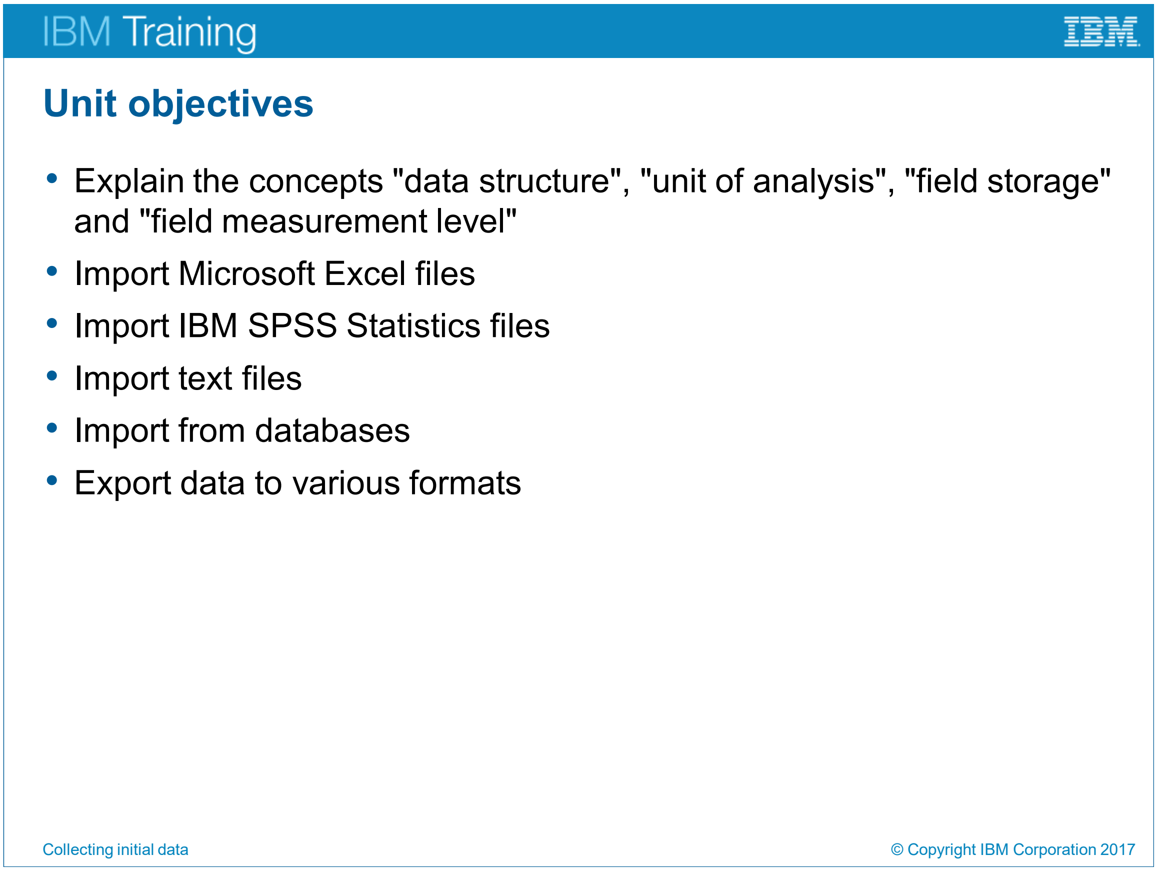
Unit 4 Collecting initial data



IBM Training

Collecting initial data

IBM SPSS Modeler (v18.1.1)



IBM Training

Unit objectives

- Explain the concepts "data structure", "unit of analysis", "field storage" and "field measurement level"
- Import Microsoft Excel files
- Import IBM SPSS Statistics files
- Import text files
- Import from databases
- Export data to various formats

Collecting initial data

© Copyright IBM Corporation 2017

Unit objectives

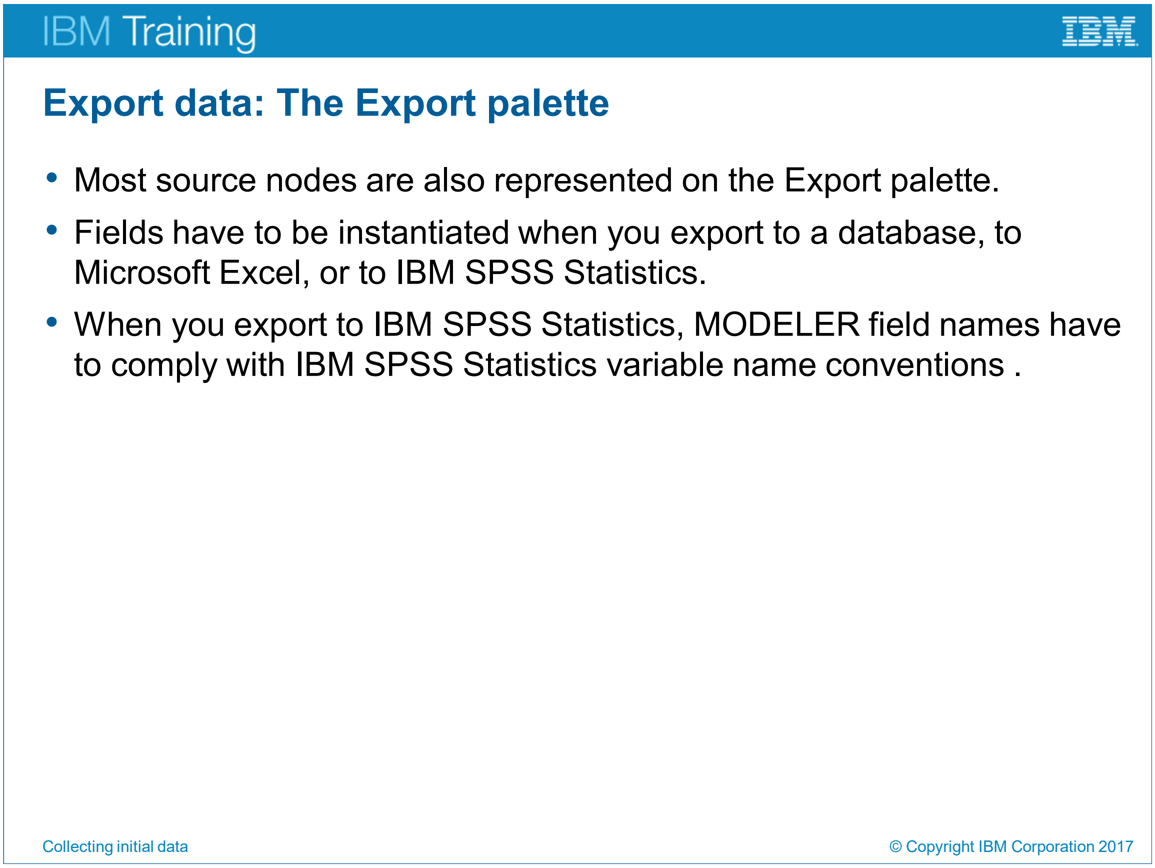
The focus in this unit is on two main tasks in the Data Understanding stage:

- Collect initial data: Getting data into IBM SPSS Modeler is the first step before any analysis can be done.
- Describe data: Describing data in terms of number of records, the number of fields, the unit of analysis, and fields' measurement levels.

First, you will be introduced to terminology such as data structure, unit of analysis, storage, and measurement level.

Before reviewing this unit you should be familiar with the following topics:

- CRISP-DM
- IBM SPSS Modeler streams, nodes, and palettes



IBM Training

Export data: The Export palette

- Most source nodes are also represented on the Export palette.
- Fields have to be instantiated when you export to a database, to Microsoft Excel, or to IBM SPSS Statistics.
- When you export to IBM SPSS Statistics, MODELER field names have to comply with IBM SPSS Statistics variable name conventions .

Collecting initial data

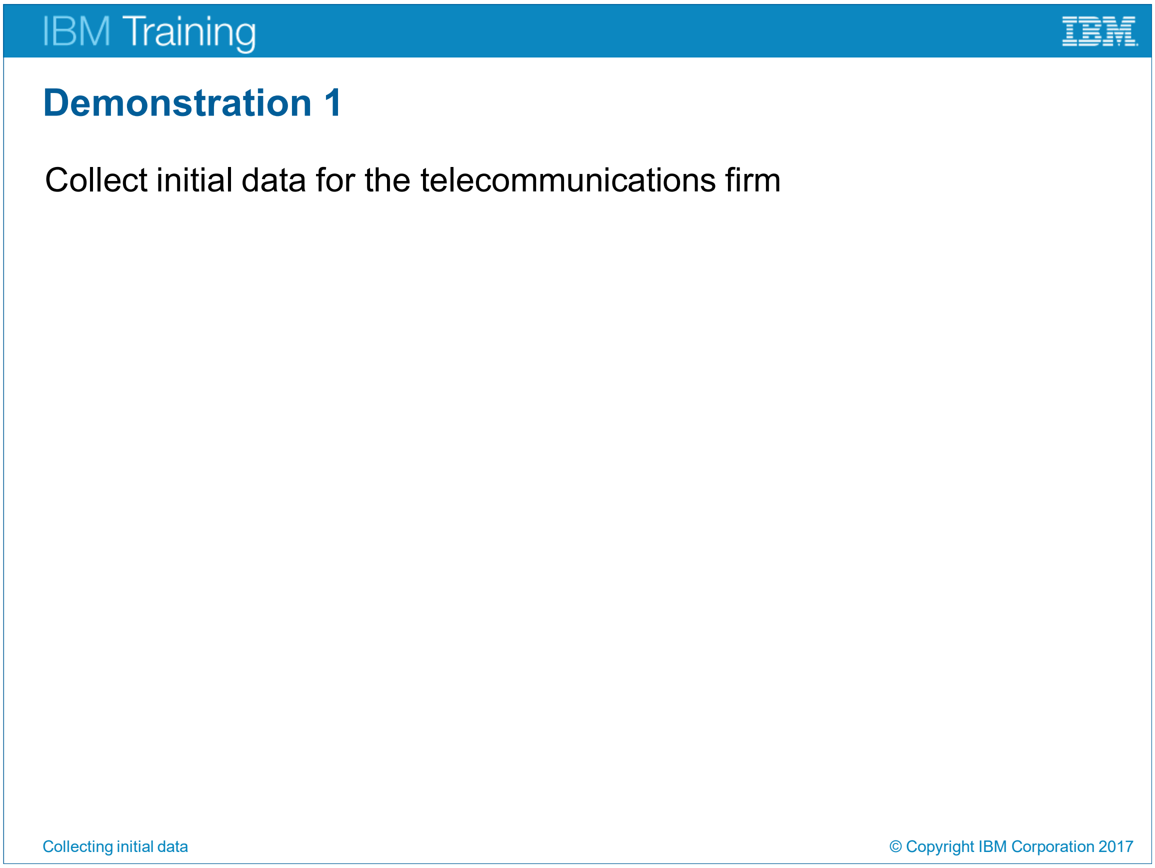
© Copyright IBM Corporation 2017

Export data: The Export palette

Almost all file types available for import have their counterpart in an export node. The Flat File exports to a delimited text file; exporting to fixed width is not supported.

When you export to a database, to Microsoft Excel, or to IBM SPSS Statistics, take into account that fields have to be instantiated, meaning a Type node has to be upstream from the export node, and values must be read in this Type node to instantiate the data.

When you export your data to an IBM SPSS Statistics file, field names have to comply with IBM SPSS Statistics field name conventions. For example, a blank are not allowed in an IBM SPSS Statistics field name. A Filter node upstream from the Statistics Export node is helpful, because it has a feature to automatically convert IBM SPSS Modeler field names to IBM SPSS Statistics field names ("variable names" in IBM SPSS Statistics).



IBM Training

Demonstration 1

Collect initial data for the telecommunications firm

Collecting initial data

© Copyright IBM Corporation 2017

Demonstration 1: Collect initial data for the telecommunications firm

Demonstration 1:
Collect initial data for the telecommunications firm

Purpose:
You work as a data scientist for a telecommunications firm. You have to import data from various sources and examine the unit of analysis and the fields' measurement levels.

Data file:	**telco x customer data.xlsx**
	telco x products.dat
	telco x call data q1.sav
Data folder:	**C:\Training\0A008**

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

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.
2. When a welcome window displays, click **Cancel**.

   If you have already configured IBM SPSS Modeler in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A008** folder, and then click **Set**.

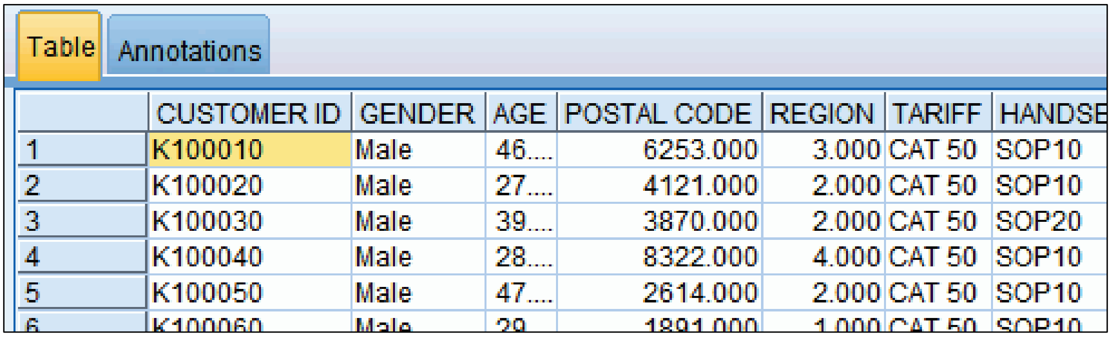
Task 2. Import a Microsoft Excel file.

In this task you will import data stored in a Microsoft Excel 2007 file, **telco x customer data.xlsx**. The Microsoft Excel file has field names in the first row.

1. From the **Sources** palette, place an **Excel** node on the stream canvas.
2. Edit the **Excel** node.
3. Click the **Data** tab, if required.
4. Beside **File type**, ensure that **Excel 2007 - 2016 (*.xlsx)** is selected.
5. Set **Import file** to **C:\Training\0A008\telco x customer data.xlsx**.

6. Ensure that the **First row has column names** option is enabled, in the left lower area of the dialog box, and then click **Preview**.

   The results appear similar to the following:



	CUSTOMER ID	GENDER	AGE	POSTAL CODE	REGION	TARIFF	HANDSE
1	K100010	Male	46....	6253.000	3.000	CAT 50	SOP10
2	K100020	Male	27....	4121.000	2.000	CAT 50	SOP10
3	K100030	Male	39....	3870.000	2.000	CAT 50	SOP20
4	K100040	Male	28....	8322.000	4.000	CAT 50	SOP10
5	K100050	Male	47....	2614.000	2.000	CAT 50	SOP10
6	K100060	Male	29	1891.000	1.000	CAT 50	SOP10

   The data import is okay.

7. Close the **Preview** output window.
8. Close the dialog box to import the Microsoft Excel file.

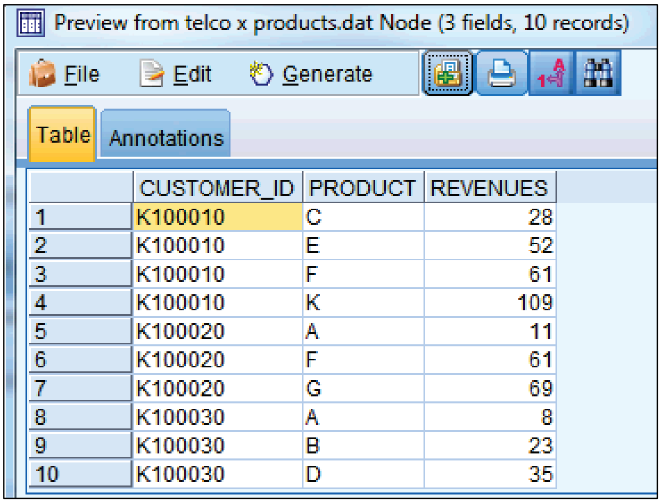
Task 3. Import a text file.

You will import data from a tab-delimited text file, telco x products.dat. The file stores information about products purchased by the customer. The file has two string fields (CUSTOMER_ID, PRODUCT) and one integer field (REVENUES).

1. From the **Sources** palette, add a **Var. File** node to the stream canvas.
2. Edit the **Var. File** node.
3. Click the **File** tab, if necessary.
4. Set the **File** field to **C:\Training\0A008\telco x products.dat**.
5. Ensure that the option **Read field names from file** is enabled.
6. In the **Field delimiters** section, clear the **Comma delimiter** option, and then enable the **Tab** option.

7. Click **Preview**.

   The results appear similar to the following:



Preview from telco x products.dat Node (3 fields, 10 records)
File
Edit
Generate
Table
Annotations

	CUSTOMER_ID	PRODUCT	REVENUES
1	K100010	C	28
2	K100010	E	52
3	K100010	F	61
4	K100010	K	109
5	K100020	A	11
6	K100020	F	61
7	K100020	G	69
8	K100030	A	8
9	K100030	B	23
10	K100030	D	35

   The data import is okay.

   A customer has as many records as he has products. A business question such as "What is the mean revenues for the customers?" cannot be answered because that would require one record per customer, and the total revenues for that customer.

8. Close the **Preview** output window.
9. Close the **Var. File** dialog box.

Task 4. Import an IBM SPSS Statistics file.

The IBM SPSS Statistics file telco x call data q1.sav stores call detail records (number of peak calls, peak minutes, etc.) for three months.

1. From the **Sources** palette, add a **Statistics File** node to the stream canvas.
2. Edit the **Statistics File** node.
3. Click the **Data** tab, if necessary.
4. Set **Import file** to **C:\Training\0A008\telco x call data q1.sav**.

   The variables and values in the file do not have labels, so you can use the default options for the import.

5. Click **Preview**.

   Each customer has three records, each record representing one month of data.

   PEAK_CALLS, OFFPEAK_CALLS, and INTERNATIONAL_CALLS have storage real in IBM SPSS Modeler, whereas they actually were integer in the IBM SPSS Statistics .sav file. To have these fields as integers in IBM SPSS Modeler, you will use the dictionary information that is contained in the IBM SPSS Statistics file.

6. Close the **Preview** output window.

7. Enable the **Use field format information to determine storage** option, in the lower left area of the dialog box.
8. Click **Preview**.

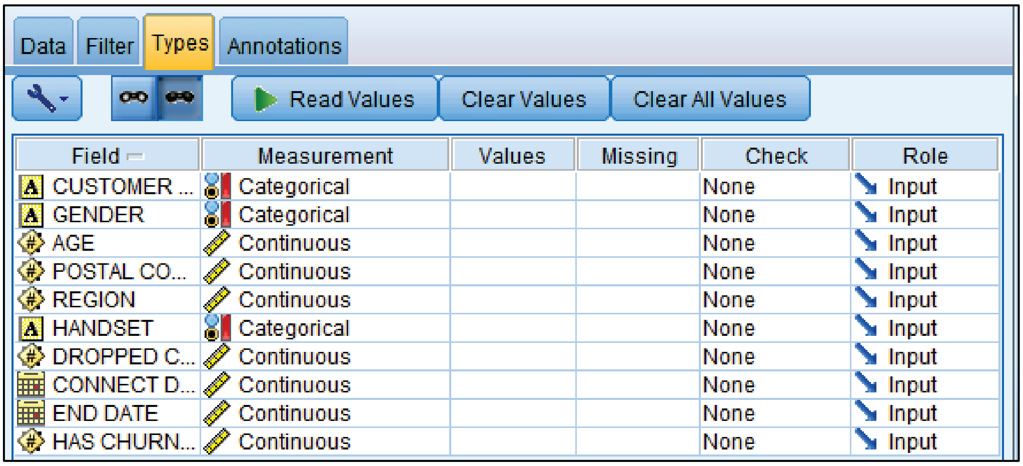
   PEAK_CALLS, OFFPEAK_CALLS, and INTERNATIONAL_CALLS display as integers now.
9. Close the **Preview** output window.
10. Close the **Statistics File** dialog box.

Task 5. Set the measurement levels.

You will check the measurement level for the fields imported from the Microsoft Excel file, telco x customer data.xlsx.

1. Edit the **Excel** source node that imports **telco x customer data.xlsx**, and then click the **Types** tab.

   The results appear similar to the following:



Field	Measurement	Values	Missing	Check	Role
CUSTOMER ...	Categorical			None	Input
GENDER	Categorical			None	Input
AGE	Continuous			None	Input
POSTAL CO...	Continuous			None	Input
REGION	Continuous			None	Input
HANDSET	Categorical			None	Input
DROPPED C...	Continuous			None	Input
CONNECT D...	Continuous			None	Input
END DATE	Continuous			None	Input
HAS CHURN...	Continuous			None	Input

The measurement level of POSTAL_CODE and REGION is set to continuous, because their values are numeric. However, the values are only of a categorical nature, so these fields should be typed as such.

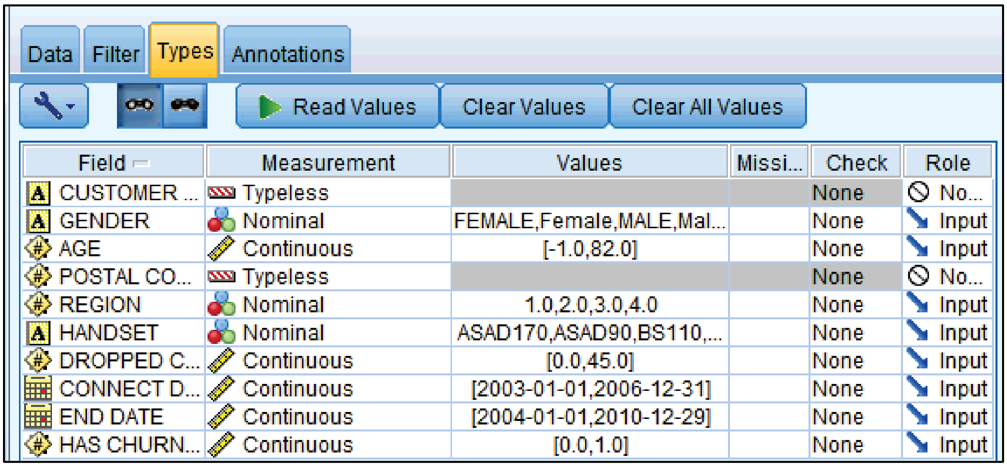
2. In the **Measurement** column, click the cell for **POSTAL_CODE**, and select **Categorical** from the dropdown.
3. In the **Measurement** column, click the cell for **REGION**, and select **Categorical** from the dropdown.

   You will read the data. The general categorical measurement level will then be instantiated to a specific measurement level.
4. Click **Read Values**.

   A message box displays, asking you if you want to read values for all fields.

5. Click **OK** to confirm.

   The results appear similar to the following:



Field	Measurement	Values	Missi...	Check	Role
CUSTOMER ...	Typeless			None	⊘ No...
GENDER	Nominal	FEMALE,Female,MALE,Mal...		None	Input
AGE	Continuous	[-1.0,82.0]		None	Input
POSTAL CO...	Typeless			None	⊘ No...
REGION	Nominal	1.0,2.0,3.0,4.0		None	Input
HANDSET	Nominal	ASAD170,ASAD90,BS110,...		None	Input
DROPPED C...	Continuous	[0.0,45.0]		None	Input
CONNECT D...	Continuous	[2003-01-01,2006-12-31]		None	Input
END DATE	Continuous	[2004-01-01,2010-12-29]		None	Input
HAS CHURN...	Continuous	[0.0,1.0]		None	Input

   REGION is instantiated to nominal, because it has more than two distinct values. CUSTOMER_ID and POSTAL_CODE are instantiated to typeless, because they have more than 250 unique values.

   The Values column shows the categories for flag and nominal fields. GENDER shows inconsistencies in spelling and that is why its measurement level is set to nominal rather than to flag.

   For continuous fields, minimum and maximum values are displayed, so that out-of-range values can be detected. For example, minimum AGE is -1, which would deserve a closer inspection.

6. Close the **Type** dialog box.

   This completes the demonstration for this unit. You will create a clean state for the exercise.

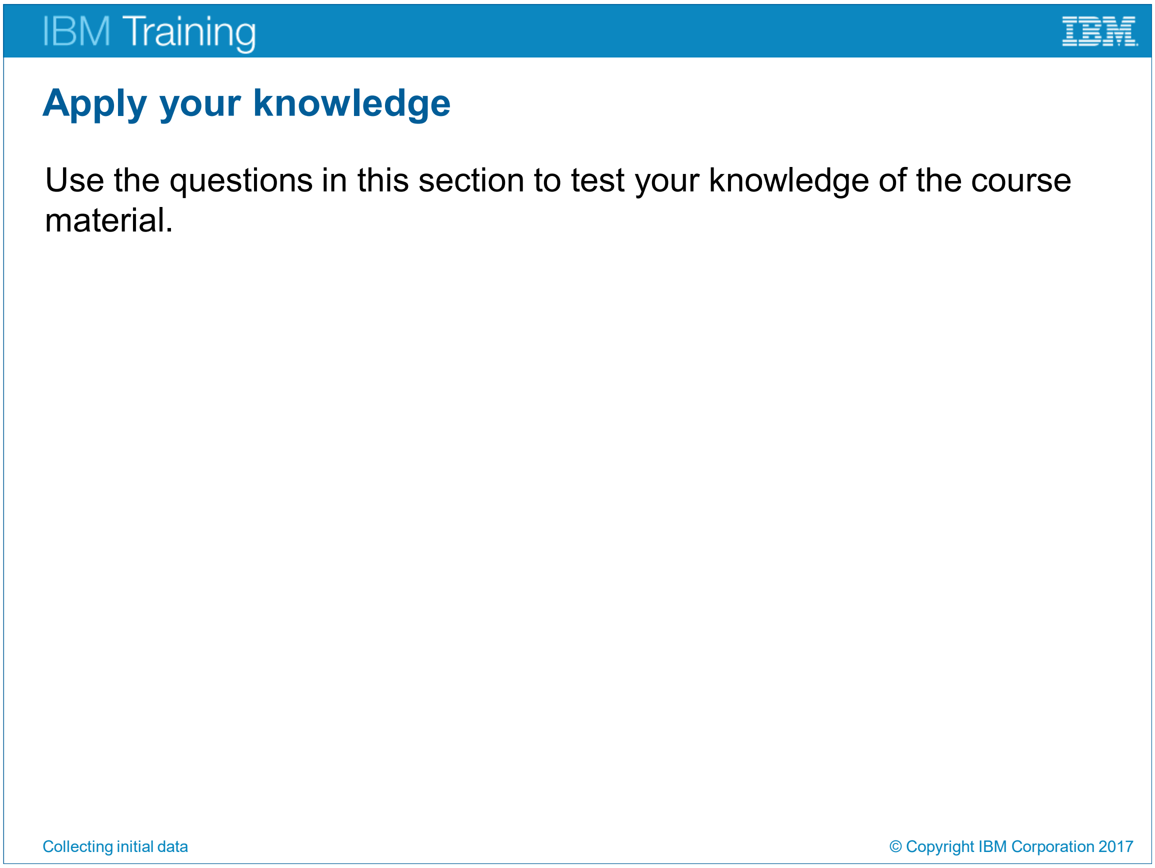
7. From the **File** menu, click **Close Stream**, and then click **No** when asked to save the stream.

8. From the **File** menu, click **New Stream**.

   Leave IBM SPSS Modeler open for the exercise.

Results:
You have imported data from various data sources and you examined the unit of analysis and the fields' measurement levels.

You will find the completed stream in the **C:\Training\0A008\04-Collecting_initial_data\Solutions** folder.



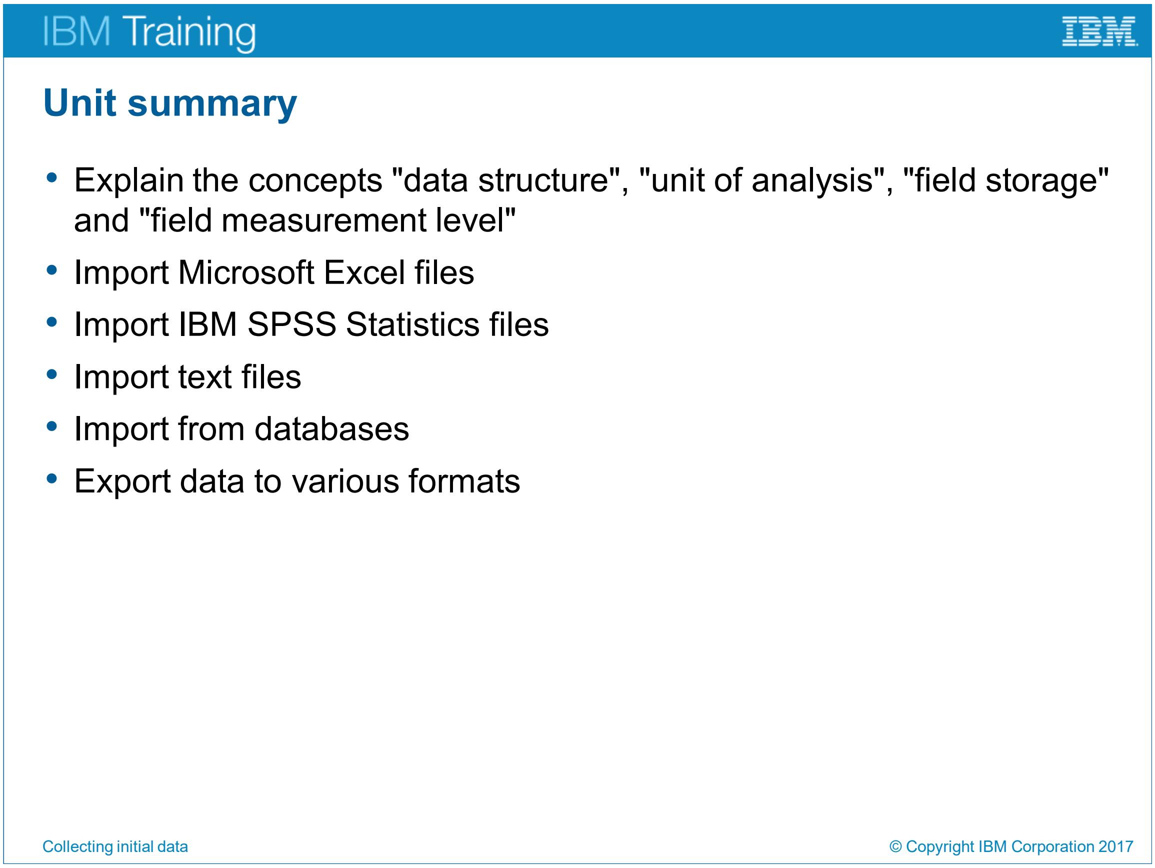
IBM Training

Apply your knowledge

Use the questions in this section to test your knowledge of the course material.

Collecting initial data

Apply your knowledge



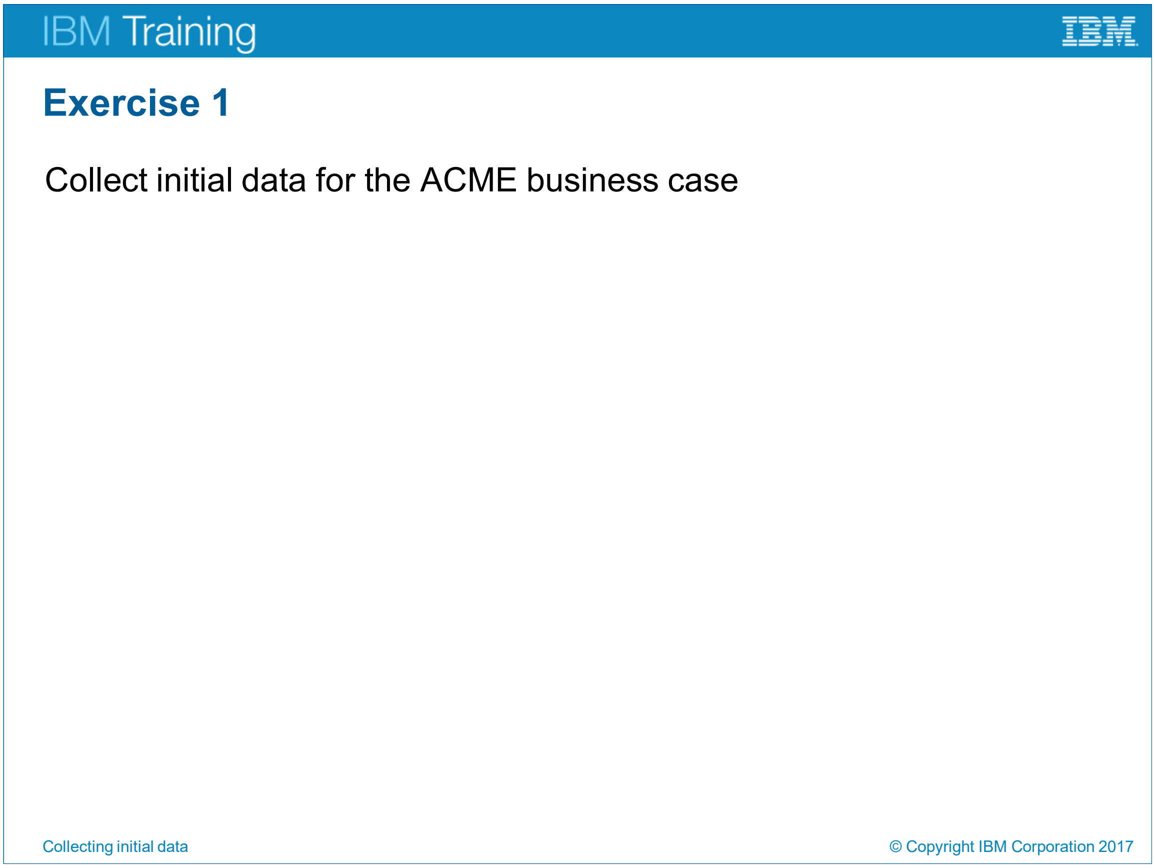
IBM Training

Unit summary

- Explain the concepts "data structure", "unit of analysis", "field storage" and "field measurement level"
- Import Microsoft Excel files
- Import IBM SPSS Statistics files
- Import text files
- Import from databases
- Export data to various formats

Collecting initial data

Unit summary



IBM Training

Exercise 1

Collect initial data for the ACME business case

Collecting initial data

Exercise 1: Collect initial data for the ACME business case

Exercise 1: Collect initial data for the ACME business case

Data file:	**ACME customers.xlsx**
	ACME purchases 1999 - 2004.dat
	ACME purchases 2005 - 2010.dat
	ACME orderlines 1999 - 2004.sav
	ACME orderlines 2005 - 2010.sav
	ACME mailing history.xlsx
	ACME postal code data.csv
Data folder:	**C:\Training\0A008**

You work for ACME, a (fictitious) company selling sport products. ACME has sent out a test mailing for one of their products. Later you will build a model to identify groups with high response rates. At this point you need to import the ACME data files.

- From the **C:\Training\0A008** folder, import the following data (ensure that each data file is imported with a separate source node, so you will have as many source nodes as data files):
    - **ACME customers.xlsx** (a Microsoft Excel file with information about customers. Ensure you select **ACME customers.xlsx**, not **ACME customer data.xlsx**. Note: The ZODIAC field denotes the astrological sign, corresponding to the customer's birth date)
    - **ACME purchases 1999 - 2004.dat** (a tab-delimited text file with purchases made by customers from 1999 to 2004)
    - **ACME purchases 2005 - 2010.dat** (a tab-delimited text file with purchases made by customers from 2005 to 2010)
    - **ACME orderlines 1999 - 2004.sav** (an IBM SPSS Statistics file with the specific items purchased from 1999 to 2004)
    - **ACME orderlines 2005 - 2010.sav** (an IBM SPSS Statistics file with the specific items purchased from 2005 to 2010)
    - **ACME mailing history.xlsx** (a Microsoft Excel file with information about which mailings were sent to a customer)
    - **ACME postal code data.csv** (a comma-separated text file with demographic information about postal codes)

- For each data source, use a **Table** node to determine the number of records and fields, and which field (/fields) identifies (/identify) a unique record.

File	# of records	# of fields	field(s) that identifies (identify) a unique record
ACME customers.xlsx	30,000	6	CUSTOMER_ID
ACME purchases 1999 - 2004.dat			
ACME purchases 2005 - 2010.dat			
ACME orderlines 1999 - 2004.sav			
ACME orderlines 2005 - 2010.sav			
ACME mailing history.xlsx			
ACME postal code data.csv			

- CUSTOMER_ID uniquely identifies a customer in ACME customers.xlsx. However, CUSTOMER_ID is not necessarily unique in ACME purchases 1999 - 2004.dat or ACME purchases 2005 - 2010.dat because a customer could have made multiple purchases.

  Likewise, PURCHASE_ID uniquely identifies a purchase in ACME purchases 1999 - 2004.dat and ACME purchases 2005 - 2010.dat, but PURCHASE_ID is not necessarily unique in ACME orderlines 1999 - 2004.sav or ACME orderlines 2005 - 2010.sav, because a particular purchase might involve multiple items (and each item is a record in the order lines datasets).

  You will verify these relationships by selecting a specific customer and trace his purchases, and the items involved in one of those purchases.

  - From the **ACME purchases 1999 - 2004.dat** dataset, select the record with **CUSTOMER_ID = 731** (recall that a Select node can be generated from a Table output window). Then, run a **Table** node to view the purchases made by this customer.

    How many purchases did customer 731 make?

    Verify that one of these purchases is the one with PURCHASE_ID = 6336.

- One of the purchases made by CUSTOMER_ID 731 is the one with PURCHASE_ID = 6336. From the **ACME orderlines 1999 - 2004.sav**, select all records where PURCHASE_ID = 6336 (recall that a Select node can be generated from a Table output window). Then, run a **Table** node to view the items involved in this purchase.

  How many items were involved in the purchase with PURCHASE = 6336?

- Ensure that CUSTOMER_ID is in upper case, in all relevant datasets.
- For the customer data (data source **ACME customers.xlsx**), go through the fields and set the appropriate measurement level.

  Note: ZODIAC represents astrological sign.

- Exit IBM SPSS Modeler without saving anything.

For more information about where to work and the exercise results, refer to the Tasks and Results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and Results

Task 1. Import data.

- **ACME customers.xlsx**: use an **Excel** node (in the dialog box, beside **File type**, ensure that **Excel 2007-2016 (*.xlsx)** is selected, then navigate to the file in the **C:\Training\0A008** folder). Note: Ensure you select **ACME customers.xlsx**, not **ACME customer data.xlsx**
- **ACME purchases 1999 - 2004.dat:** use a **Var. File** node (ensure that the delimiter is **Tab**, not Comma).
- **ACME purchases 2005 - 2010.dat**: use a **Var. File** node (ensure that the delimiter is **Tab**, not Comma).
- **ACME orderlines 1999 - 2004.sav:** use a **Statistics File** node (use default settings for **Variable names** and **Values**; enable the **Use field format information to determine storage** option to use dictionary information contained in the IBM SPSS Statistics file).
- **ACME orderlines 2005 - 2010.sav**: use a **Statistics File** node (use default settings for **Variable names** and **Values**; enable the **Use field format information to determine storage** option).
- **ACME mailing history.xlsx**: use an **Excel** node (ensure that the file type is **Excel 2007 - 2016 (*.xlsx)**.
- **ACME postal code data.csv**: use a **Var. File** node (use default settings for the import).

Task 2. Determine the unit of analysis.

- Use a **Table** node (**Output** palette) downstream from each data source to determine the number of records and fields, also to identify the field or fields that defines or define unique records.

  The results are as follows:

File	# of records	#of fields	field(s) that identifies (identify) a unique record
ACME customers.xlsx	30,000	6	CUSTOMER_ID
ACME purchases 1999 - 2004.dat	4,018	3	PURCHASE_ID
ACME purchases 2005 - 2010.dat	67,109	3	PURCHASE_ID
ACME orderlines 1999 - 2004.sav	7,426	4	ITEM_ID
ACME orderlines 2005 - 2010.sav	162,308	4	ITEM_ID
ACME mailing history.xlsx	21,000	2	customer_id, in combination with mailing
ACME postal code data.csv	4,983	4	POSTAL_CODE

Task 3. Determine the relationships between datasets.

- To select **CUSTOMER_ID = 731** from the **ACME purchases 1999 - 2004.dat** dataset, run a **Table** node downstream from the **ACME purchases 1999 - 2004.dat** dataset. Click the value **731** in the **CUSTOMER_ID** column, click **Generate** from the menu and choose **Select Node ("And")**. Add the generated **Select** node downstream from the node named **ACME purchases 1999 - 2004.dat**, add a **Table** node downstream from the **Select** node, and then run it.

  The Table output window displays three records; one of them is the purchase with PURCHASE_ID = 6336.

- In the same way, select the records with **PURCHASE_ID = 6336** from the **ACME orderlines 1999 - 2004.sav**. Ensure that the **Select** node is added downstream from the **ACME orderlines 1999 - 2004.sav** node, and then add and run a **Table** node downstream from the **Select** node.

  The purchase with PURCHASE_ID = 6336 involves three items.

  You have verified that a customer can have multiple purchases and that a purchase can have multiple order lines. You can think of the order lines as providing detailed information about purchases, like multiple items (order lines) on a cash receipt (the purchase).

  Examining the relationships between datasets is an important step to understand the data.

Task 4. Rename CUSTOMER_ID so that it is the same.

- In the mailing history data, the field names are in lower case letters. Use a **Filter** node downstream from the data source node to rename **customer_id** into **CUSTOMER_ID**. Or, change the name in the data source node, Filter tab.

  Note: When you want to merge datasets, the name of the key field in the different datasets should match, also in case.

Task 5. Set measurement levels.

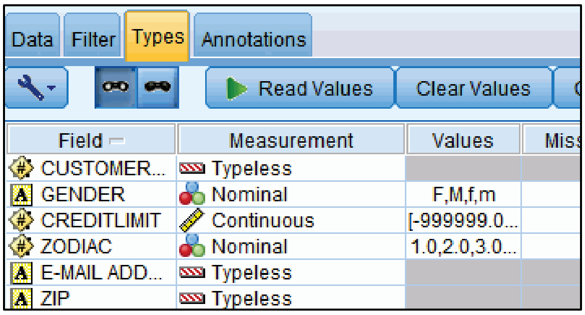
- In the customer dataset, ZODIAC is typed as continuous (because it has real storage). Set its measurement level to nominal in a **Type** node (or in the **Types** tab of the **Excel** data source node).

  CUSTOMER_ID is typed as continuous, because it has numeric storage, but it should be typeless.

- Set **CUSTOMER_ID** measurement to **Typeless**.

- Click the **Read Values** button to instantiate the data.

  The results appear as follows:



Field	Measurement	Values	Miss
CUSTOMER...	Typeless		
GENDER	Nominal	F,M,f,m	
CREDITLIMIT	Continuous	[-999999.0...	
ZODIAC	Nominal	1.0,2.0,3.0...	
E-MAIL ADD...	Typeless		
ZIP	Typeless		

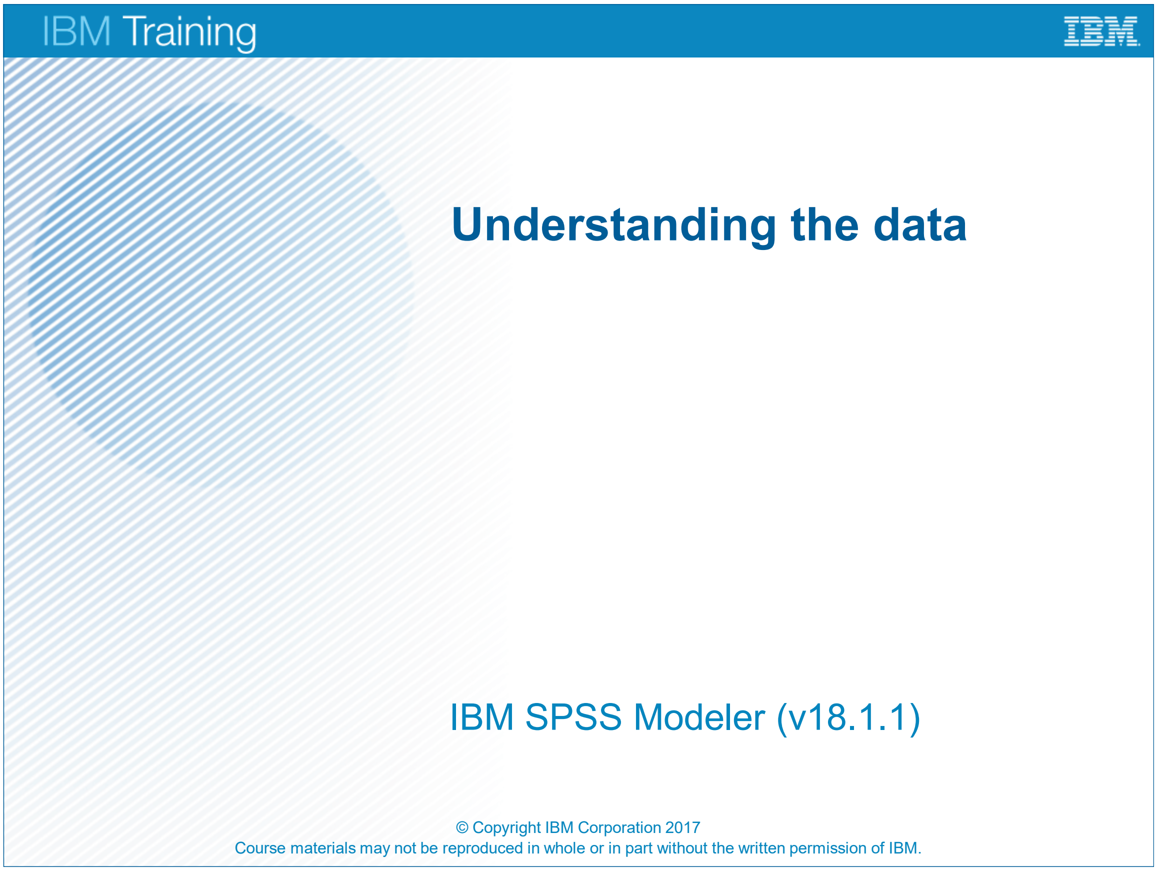
- After reviewing, click **OK** to close the dialog.

Task 6. Exit IBM SPSS Modeler.

- From the **File** menu, click **Exit**, and then exit IBM SPSS Modeler without saving anything.

You will find the solution results in the **C:\Training\0A008\04-Collecting_initial_data\Solutions** folder.

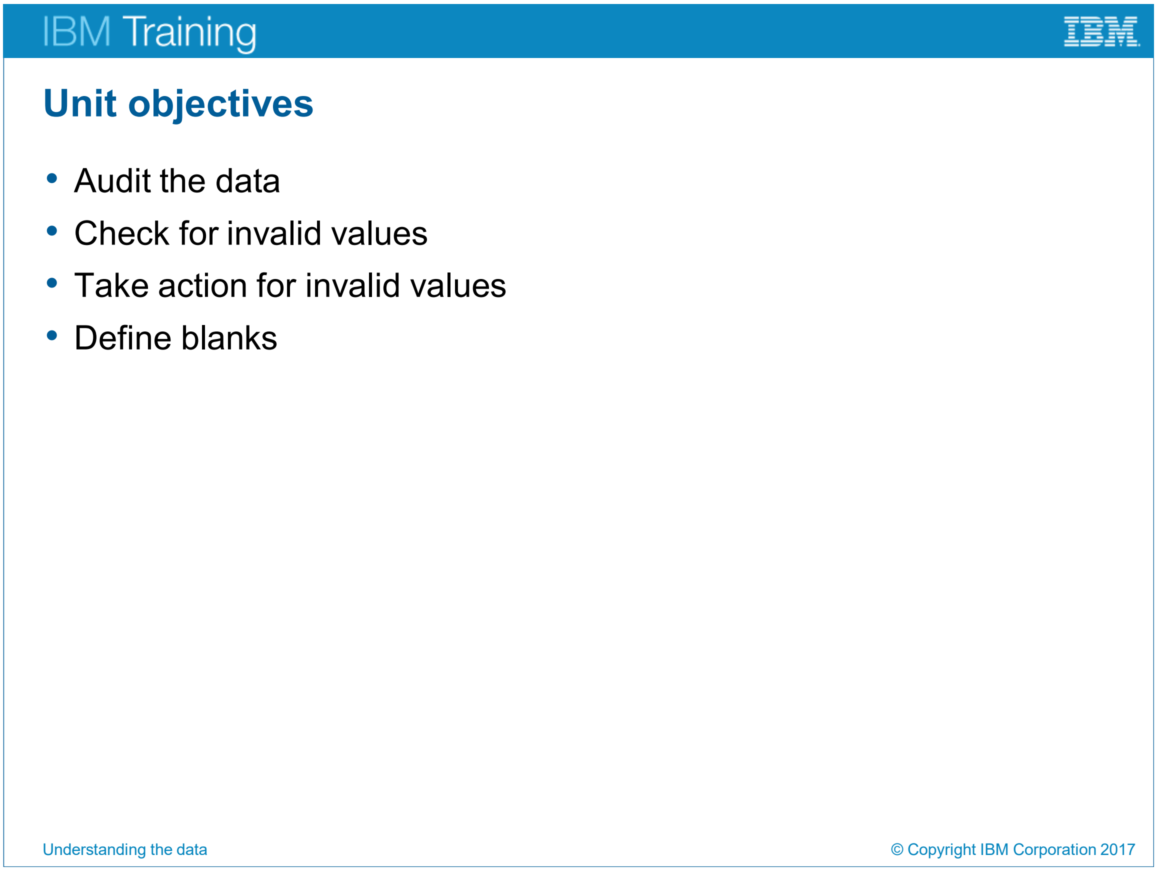
Unit 5 Understanding the data



IBM Training

Understanding the data

IBM SPSS Modeler (v18.1.1)



IBM Training

Unit objectives

- Audit the data
- Check for invalid values
- Take action for invalid values
- Define blanks

Understanding the data

© Copyright IBM Corporation 2017

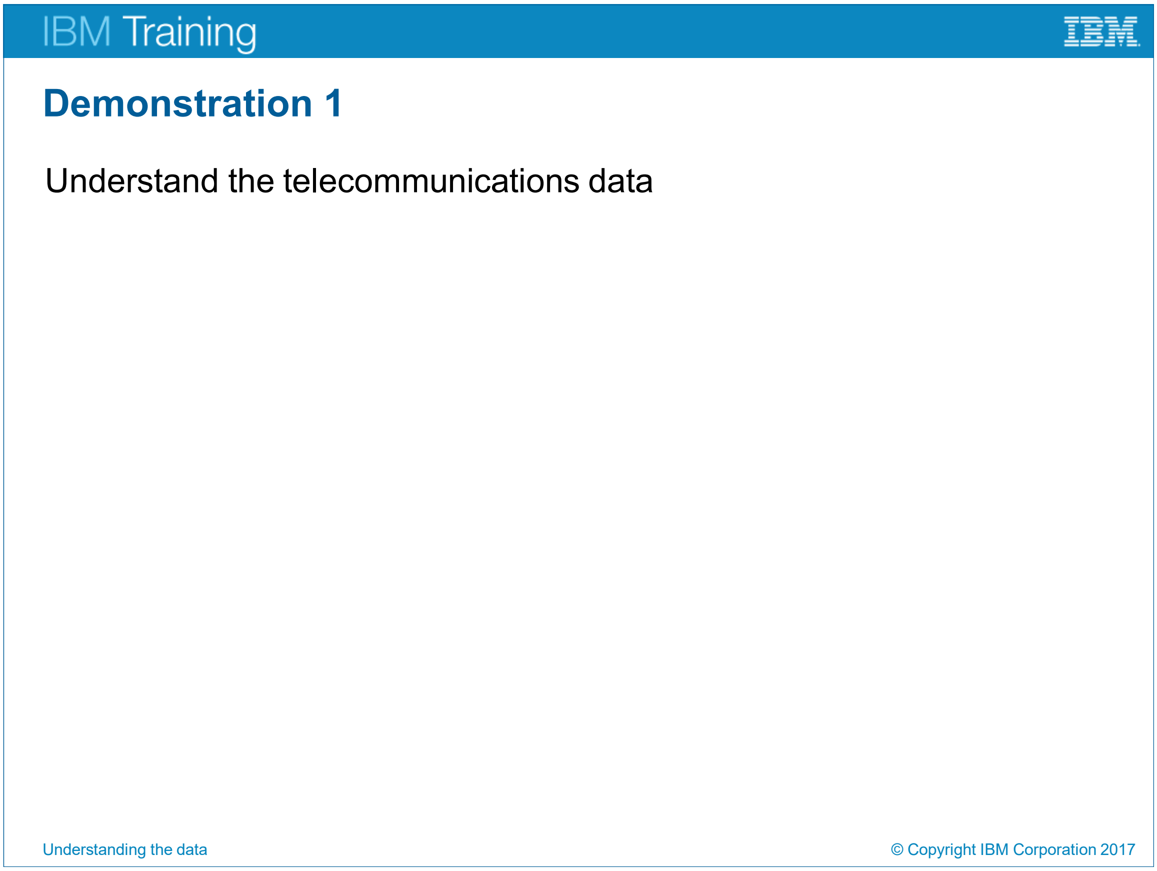
Unit objectives

Once data is read into IBM SPSS Modeler, you need to explore the data and become thoroughly familiar with its characteristics. The data most likely contains errors and missing information. Therefore, the quality of the data must be assessed before models are built. The higher the quality of the data used, the more accurate the predictions and the more useful the results.The focus in this unit is on two tasks in the Data Understanding stage in CRISP-DM:

- explore the data by running a data audit
- assess the quality of the data by reporting out-of range values and dealing with missing data

Before reviewing this unit participants should be familiar with:

- CRISP-DM
- IBM SPSS Modeler streams, nodes and palettes
- Methods to collect initial data
- Measurement levels and storages



IBM Training

Demonstration 1

Understand the telecommunications data

Understanding the data

© Copyright IBM Corporation 2017

Demonstration 1: Understand the telecommunications data

Demonstration 1: Understand the telecommunications data

Purpose:
You work as a data scientist for a telecommunications firm. You have imported data, and you need to assess the quality of the data and define blanks where appropriate.

Data file:	**telco x customer data.xlsx**
Data folder:	**C:\Training\0A008**
Stream file:	**unit_05_demonstration_1_start.str**
Stream file folder:	**C:\Training\0A008\05-Understanding_the_data\Start**

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

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.
2. When a welcome window appears, click **Cancel**.

   If you have already configured IBM SPSS Modeler in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A008** folder, and then click **Set**.

Task 2. Open the start stream and audit the data.

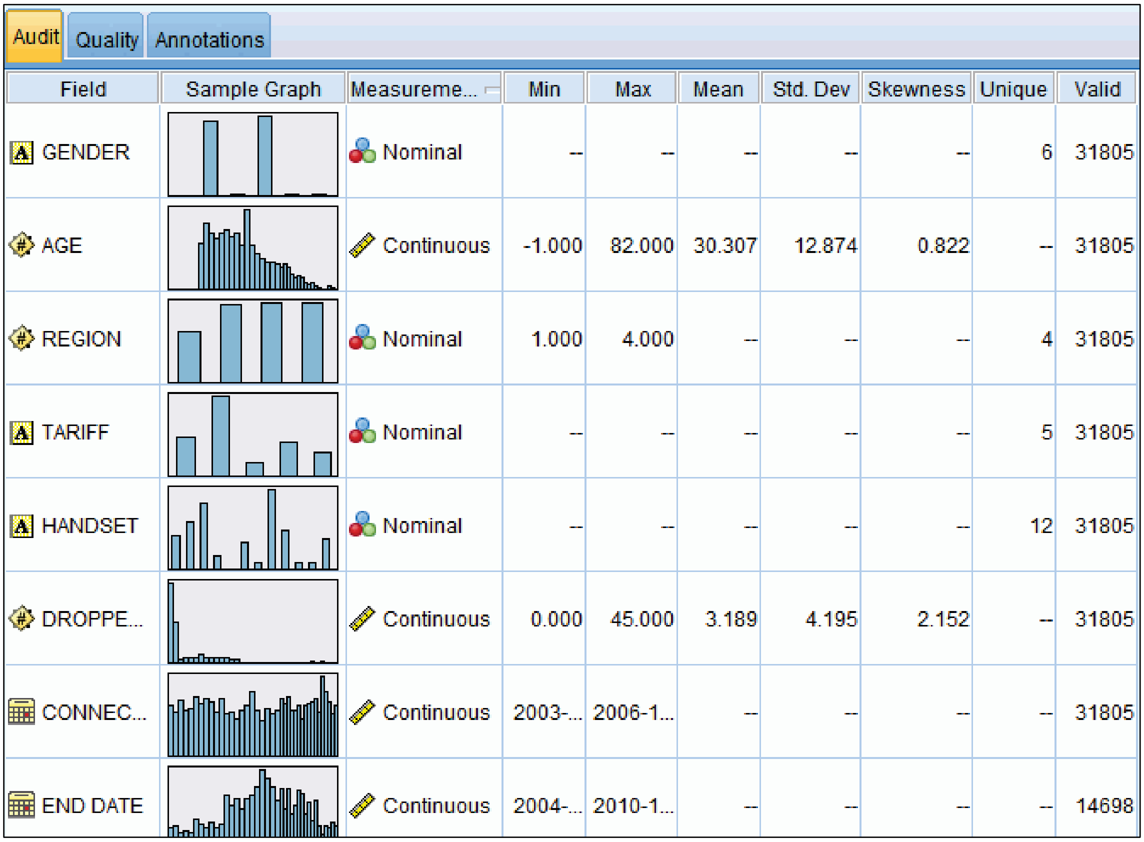
1. From the **File** menu, click **Open Stream**, navigate to **C:\Training\0A008\05-Understanding_the_data\Start** and then open **unit_05_demonstration_1_start.str**.
2. Run the **Table** node.

   The data stores demographical and usage information about customers from a telecommunications firm.
3. Scroll all the way to the right, and then scroll down so that a $null$ value appears for **END_DATE**.

   END_DATE is missing for customers that did not cancel their subscription. They are still with the company.
4. Close the **Table** output window.

5. From the **Output** palette, add a **Data Audit** node downstream from the **Type** node.
6. Run the **Data Audit** node.

   The results appear as follows:



Field	Sample Graph	Measureme...	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
GENDER		Nominal	--	--	--	--	--	6	31805
AGE		Continuous	-1.000	82.000	30.307	12.874	0.822	--	31805
REGION		Nominal	1.000	4.000	--	--	--	4	31805
TARIFF		Nominal	--	--	--	--	--	5	31805
HANDSET		Nominal	--	--	--	--	--	12	31805
DROPPE...		Continuous	0.000	45.000	3.189	4.195	2.152	--	31805
CONNEC...		Continuous	2003-...	2006-1...	--	--	--	--	31805
END DATE		Continuous	2004-...	2010-1...	--	--	--	--	14698

All but typeless field are included in the data audit report. By default, only basic statistics such as minimum, maximum, and mean are reported. Median and mode are not included because computation may be too time consuming, but you can request them in the Data Audit dialog box. You can also request more advanced statistics such as standard errors.

The number of valid values for GENDER, AGE, REGION, CONNECT_DATE is 31,805, the number of records in the dataset. END_DATE has 14,698 valid values. You will examine this field later in this demonstration.

The minimum value for AGE is -1, which points to an issue. Also, the graph for GENDER shows more than two bars, which is suspect.

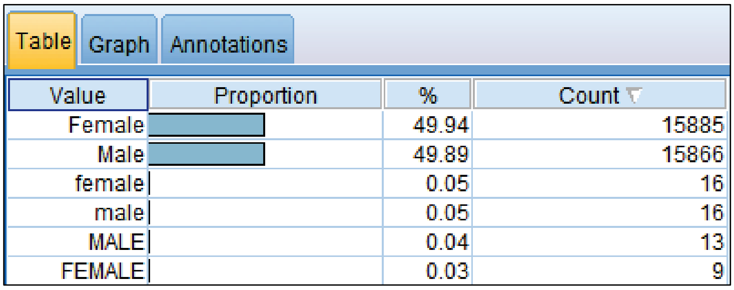
The graphs in the data audit report are thumbnails and do not show the full details. You can zoom in by double-clicking the thumbnail.

7. Double-click the thumbnail for **GENDER**.

   A Distribution output window opens.

8. In the **Distribution** output window, click the **Count** column header twice, to sort the values descending on frequency.

   The results appear similar to the following:



Value	Proportion	%	Count
Female		49.94	15885
Male		49.89	15866
female		0.05	16
male		0.05	16
MALE		0.04	13
FEMALE		0.03	9

   Female and Male have the highest frequency. The other spellings should be reclassified into Female and Male, using an appropriate field operations node. The *Deriving and reclassifying fields* unit in this course will address this issue.

   If preferred, you can copy the Distribution graph to a Microsoft Office Graphic Object. Once in Microsoft Office, you can edit the graph like any other graphic.

9. Click the **Graph** tab.
10. From the **Edit** menu, select **Copy Microsoft Office Graphic Object**.

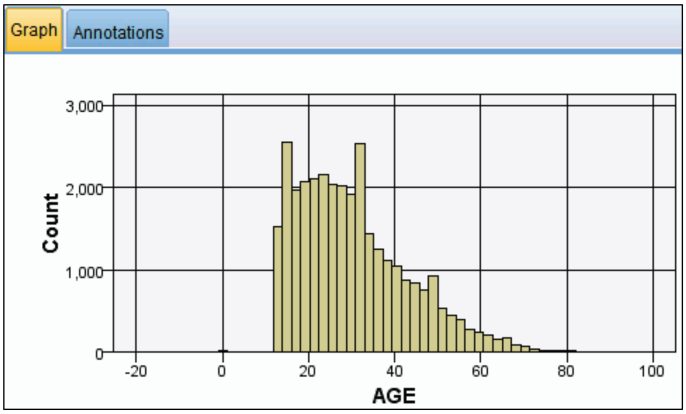
    The content will be copied to the clipboard. When you switch to a Microsoft Office application, for example Microsoft Excel, choose Paste Special, and then select Microsoft Office Drawing Object. You can then customize the graph in the Microsoft application.

    It should be noted that not all graphs support this feature, in which case the Copy Microsoft Office Graphic Object menu option will be disabled. Refer to the Help for a list of graphs that can be copied to a Microsoft Office Graphic Object.

11. Close the **Distribution** output window.
12. In the **Data Audit** output window, double-click the thumbnail for **AGE**.

    Note: If the Data Audit output window is not displayed, use Alt+Tab to scroll through the open windows until you locate it.

    The results appear similar to the following:



The bar near 0 on the x-axis points to a few records with AGE -1.

13. Close the **Histogram** output window.
14. Close the **Data Audit** output window.

Task 3. Define valid values and take action.

In the previous task you have explored the data and you have found that the value -1 was the minimum value for AGE.

In this task you will define a valid range for AGE.

1. Edit the **Type** node, and click the **Types** tab, if necessary.

   Clicking in a cell in the Values column will bring up the Values sub dialog box where you specify the set of allowable values (for categorical fields), or the range of allowable values (for continuous fields). You can also specify blank values, including the null value, in this sub dialog box. For string fields there is an extra option to declare the white space value as blank.
2. In the **Values** column, click the cell in the **AGE** row (where it reads [-1.0, 82.0]), and then click **Specify**.

   The AGE Values dialog box opens.
3. Click the **Specify values and labels** option, and then set **Lower** to **12**, and **Upper** to **90**.
4. Close the **AGE Values** dialog box.

   In the Check column, select the action that you want to be taken when an invalid value is encountered.
5. In the **Check** column, click the cell in the **AGE** row, and then select **Warn**.

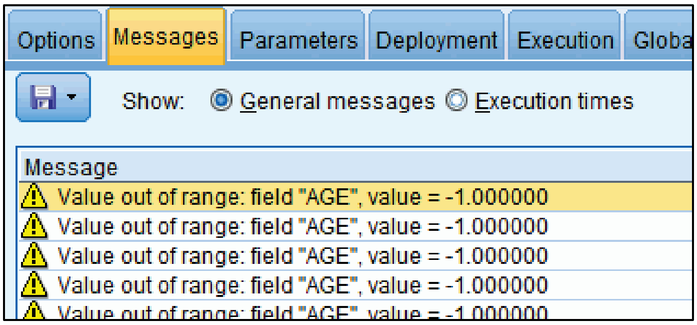
   You will rerun the Data Audit node to examine the effect of specifying a valid range.
6. Close the **Type** dialog box.
7. Run the **Data Audit** node.

   The minimum AGE value is still -1, so it seems as if nothing has changed.
8. Close the **Data Audit** output window.
9. Run the **Table** node downstream from the **Type** node.
10. Close the **Table** output window.

11. From the **Tools** menu, click **Stream Properties**, and then click **Messages**.

    (Alternatively, click the Show stream messages ⚠ button in the status bar in the bottom right corner of the window.)

    The results appear as follows:

    Show: ● General messages ○ Execution times

Message
⚠ Value out of range: field "AGE", value = -1.000000
⚠ Value out of range: field "AGE", value = -1.000000
⚠ Value out of range: field "AGE", value = -1.000000
⚠ Value out of range: field "AGE", value = -1.000000
⚠ Value out of range: field "AGE", value = -1.000000

12. Close the **Stream** messages window.

    As another example of dealing with missing values, you will discard records with an undefined ($null$) value for **END_DATE**.

13. Edit the **Type** node, click the cell in the **Check** column in the **END_DATE** row, and then set the action to **Discard**.

    You will examine the effect of removing these records.

14. Close the **Type** dialog box.

15. Rerun the **Table** node that is downstream from the **Type** node.

16. Scroll to the right, so you can view **END_DATE**, and then scroll down to observe that **END_DATE** is never $null$.

    You discarded records with END_DATE missing, and therefore you have removed the customers who did not cancel their subscription. This will make modeling later impossible because for that you need both customers who cancelled their subscription and customers who did not.

    The crucial concern is whether there is a pattern to the missing data such that there is a difference between the records with missing data and those without missing data. If there is, then your model can be misestimated and cannot be applied to the full population of interest. Removing records that have an undefined value for END_DATE is an extreme example of this.

    All in all, you will need to know the business context to take the appropriate action. Records having an undefined END_DATE cannot be discarded, so you will undo the action.

17. Close the **Table** output window.

18. Edit the **Type** node, and in the **Role** column for **END_DATE**, set the action to **None**.

19. Close the **Type** dialog box.

Task 4. Declare blank values.

In the previous task, a warning message was issued when a value was out of range for AGE. However, it is known beforehand that the value -1 can appear in the data, because it is used when AGE is unknown. This value needs to be declared as blank value, so that out-of-range messages will only report values that are out-of-range unexpectedly.

1. Edit the **Type** node.
2. In the **Missing** column, click the cell in the **AGE** row, and then click **Specify**.
3. Enable the **Define blanks** check box.
4. Under **Missing values**, type **-1**.
5. Close the **AGE Values** dialog box.

   The asterisk in the Missing column indicates that blanks are defined for AGE.
6. Close the **Type** dialog box.

   You will verify that warning messages will no longer be issued.
7. Run the **Data Audit** node.

   The minimum value for AGE is 12, instead of -1 that you had in the previous task.
8. Close the **Data Audit** output window.

   There are no stream messages, so no out-of-range values were found.

Task 5. Declare undefined values as blanks for all fields.

In the data checking process you probably do not want to discard records, or have a warning message issued, when undefined ($null$) values are encountered. To prevent that from happening, you will declare the undefined value as blank for all fields.

1. Edit the **Type** node.
2. Right-click any field, and then click **Select All** from the context menu.
3. Right-click any field, click **Set Missing** from the context menu, and then click **On (*)**.

   All fields, except the typeless fields, have an asterisk in the Missing column, indicating that blanks are defined for each of them.

   To examine the difference with the situation, you will look at the blank definitions for one of the fields.
4. In the **Missing** column, click the cell in the **REGION row**, and then select **Specify**.

   The Define blanks option is enabled, and so is the Null option. Thus, null values are declared as blank values for REGION, and the same for the other fields.

5. Close the **REGION Values** dialog box, and then close the **Type** dialog box.

   This completes the demonstration for this unit. You will create a clean state for the exercise.

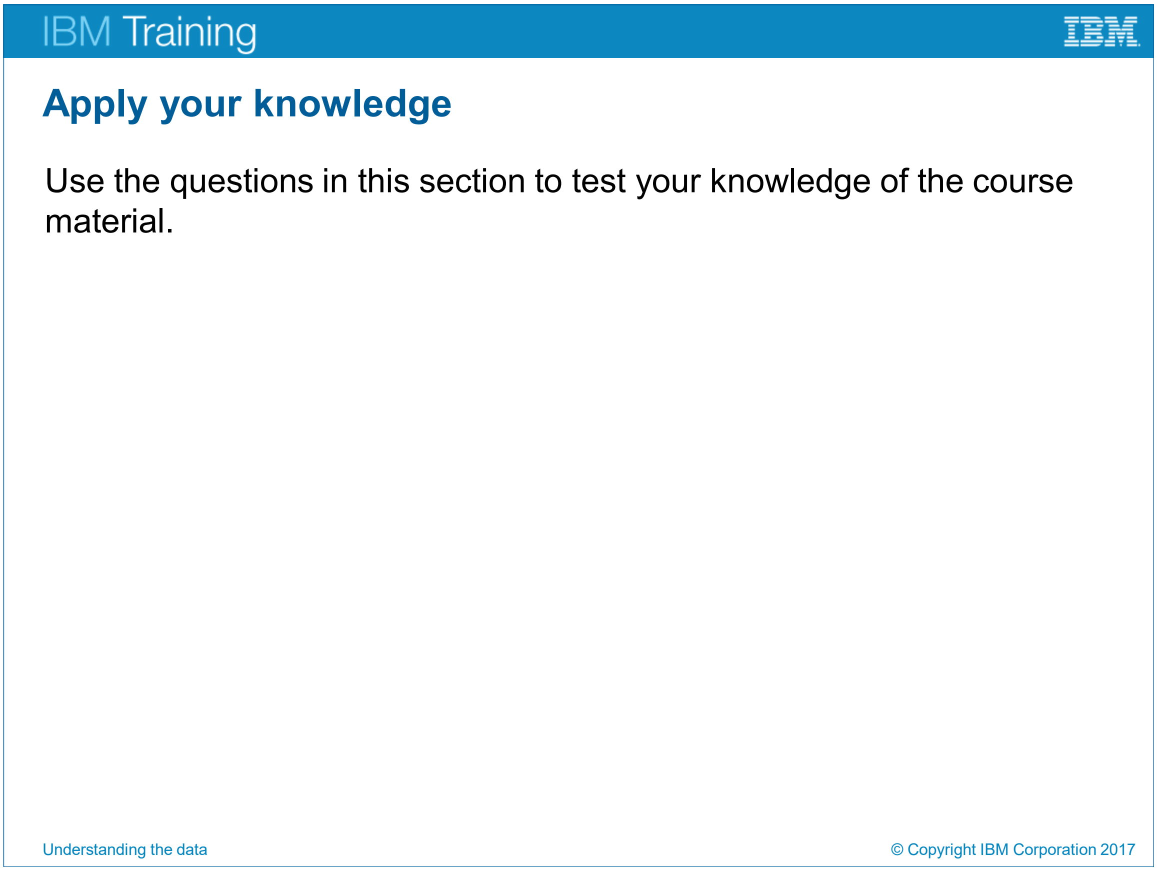
6. From the **File** menu, click **Close Stream**, and then click **No** when asked to save the stream.

7. From the **File** menu, click **New Stream**.

   Leave IBM SPSS Modeler open for the exercise.

Results:
You have imported data, assessed its quality, and defined blanks where appropriate.

You will find the completed stream in the **C:\Training\0A008\05-Understanding_the_data\Solutions** folder.



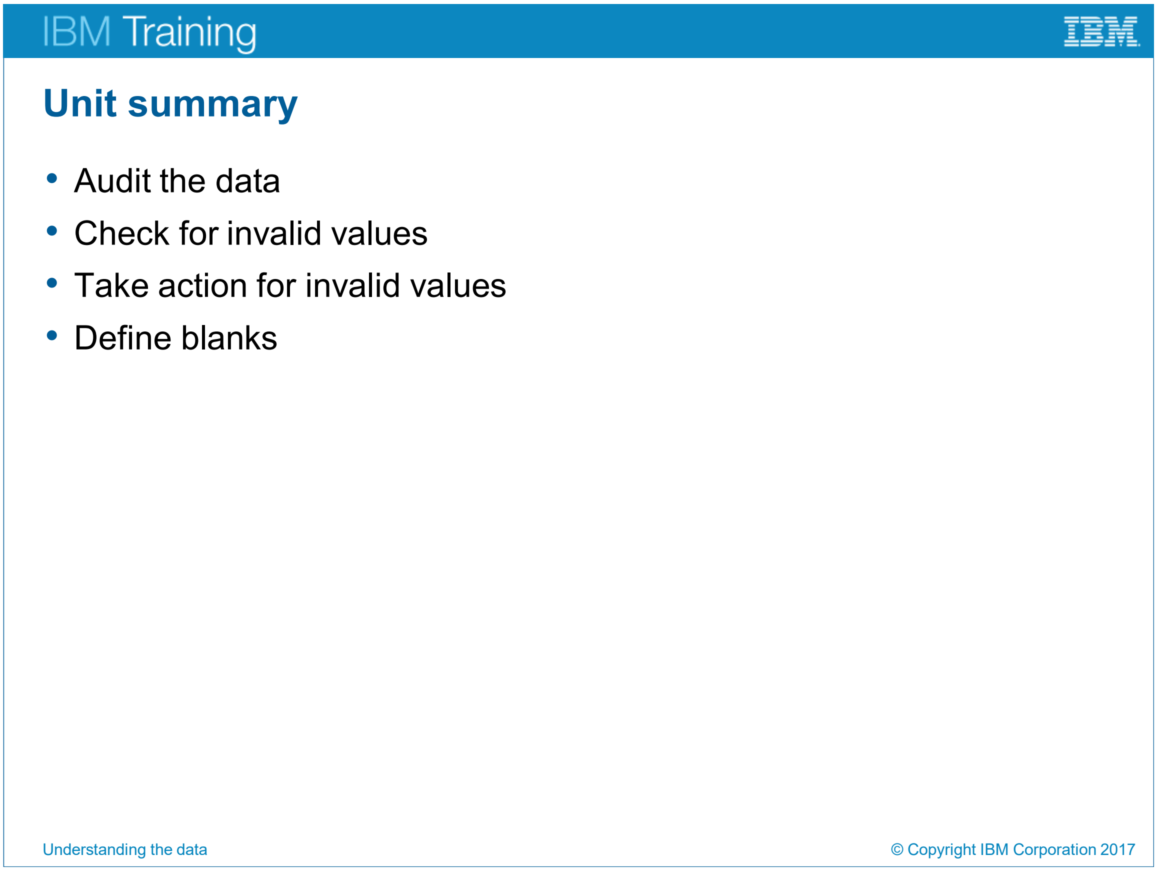
IBM Training

Apply your knowledge

Use the questions in this section to test your knowledge of the course material.

Understanding the data

Apply your knowledge



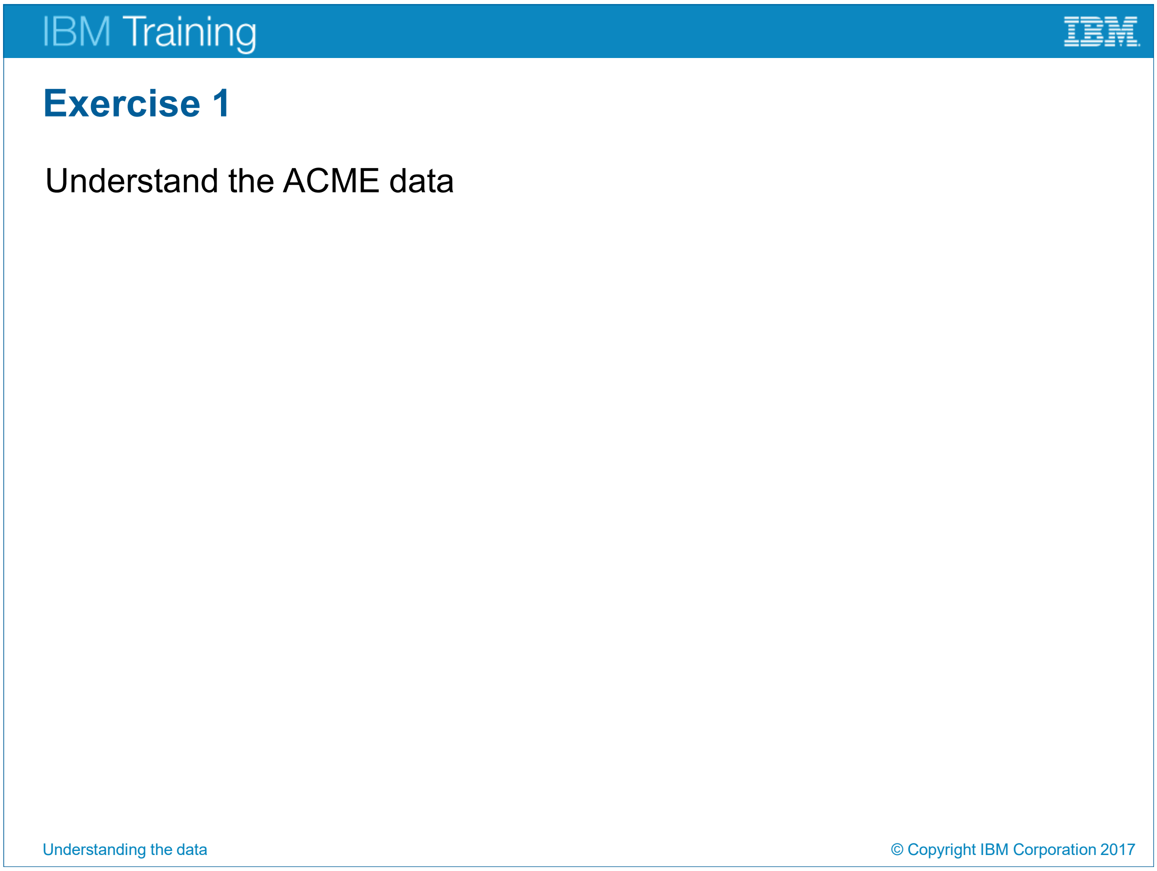
IBM Training

Unit summary

- Audit the data
- Check for invalid values
- Take action for invalid values
- Define blanks

Understanding the data

Unit summary



IBM Training

Exercise 1

Understand the ACME data

Understanding the data

Exercise 1: Understand the ACME data

Exercise 1:
Understand the ACME data

Data file:	**ACME customer data.txt**
Data folder:	**C:\Training\0A008**

You work at ACME, a company selling sport products. ACME wants to promote a new product through direct mail. ACME has sent out a test mailing and has collected data on the response for this test mailing. It is your job, in preparation of building models later, to examine the quality of one of ACME's datasets, and to take corrective action where needed.

- Import the data from **ACME customer data.txt**, located in the **C:\Training\0A008** folder. Run a **Table** node to view the data, and then audit the data.

  What is the number of records in the dataset?

  Do you trust the estimate for the mean AGE?

  What is the number of valid records for AGE? Can you explain the difference between the number of records in the dataset and the number of valid records for AGE?

- In a **Type** node, instantiate the data. Then, set a [15, 75] valid range for **AGE**. Also, ensure that a warning is issued when an out-of-range value is encountered for **AGE**. Then, run a **Data Audit** node downstream from the **Type** node, and view the stream messages.

  Which values are encountered in the data that are out- of-range?

- In the **Type** node, declare values **0** and **999** as blank values for **AGE**. Also, ensure that the $null$ value is declared as blank for **AGE**. Keep the same valid range for **AGE** as before, **[15, 75]**. Then, run a **Data Audit** node downstream from the **Type** node.

  What is the mean AGE?

- Exit IBM SPSS Modeler without saving anything.

For more information about where to work and the exercise results, refer to the Tasks and Results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and Results

Task 1. Import and examine the data.

- From the **Sources** palette, use the **Var. File** node to import data from **ACME customer data.txt**, located in the **C:\Training\0A008** folder (use default settings for the import).
- Run a **Table** node (**Output** palette).

  The dataset has 30,014 records.
- Run a **Data Audit** node (**Output** palette).

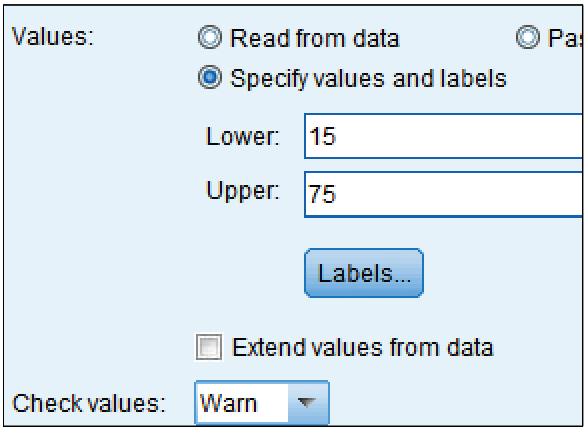
  AGE ranges from 0 to 999; mean AGE is 27.1. This statistic is incorrect, because the values 0 and 999 are included in the computation.

  The number of valid records for AGE is 30,002. Apparently, 12 records have an undefined ($null$) value for AGE.

Task 2. Set a range for AGE and issue a warning for out-of-range values.

- Add a **Type** node downstream from the **Var. File** node.
- Edit the **Type** node, click the cell in the **Values** column, **AGE** row, and then click **Specify**.
- Enable the **Specify values and labels** option, set the **Lower** and **Upper** range to **[15, 75]**, and then change the **Check values** field to **Warn**.

  The results appear as follows:



Values: ○ Read from data ○ Pa
● Specify values and labels

Lower: 15

Upper: 75

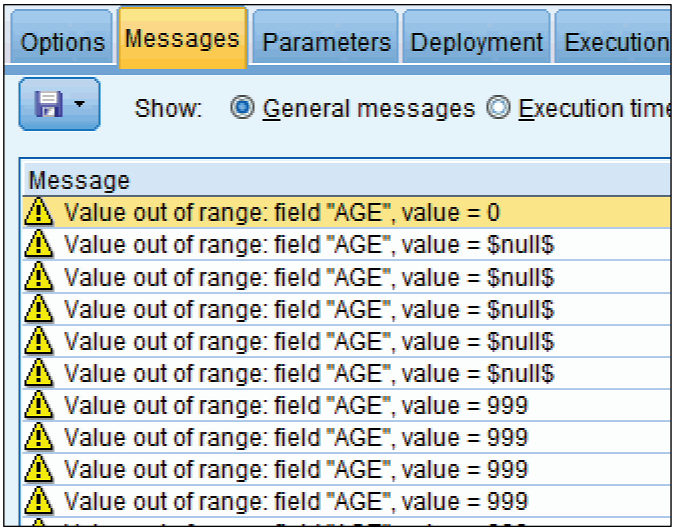
□ Extend values from data

Check values: Warn

- Run a **Data Audit** node downstream from the **Type** node.

  The Messages window will report the out-of-range values.

  The results appear as follows:



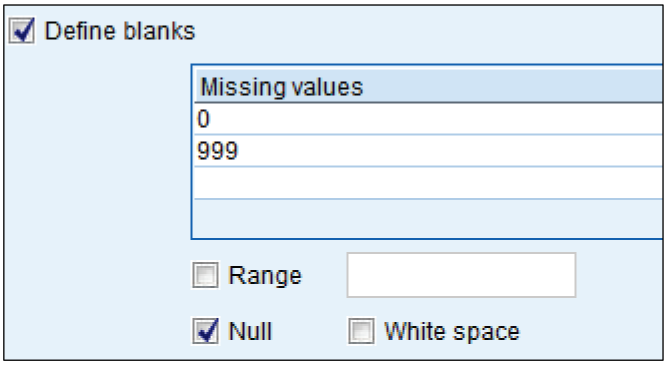
Message
Value out of range: field "AGE", value = 0
Value out of range: field "AGE", value = $null$
Value out of range: field "AGE", value = $null$
Value out of range: field "AGE", value = $null$
Value out of range: field "AGE", value = $null$
Value out of range: field "AGE", value = $null$
Value out of range: field "AGE", value = 999
Value out of range: field "AGE", value = 999
Value out of range: field "AGE", value = 999
Value out of range: field "AGE", value = 999

0, 999, and $null$ are out-of-range values.

Task 3. Declare blanks for AGE.

- Edit the **Type** node, and in the **Missing** column for **AGE**, select **Specify**.
- Enable the **Define blanks** options, and under **Missing values**, enter **0** and **999** as blanks. Also, ensure that the **Null** option is enabled.

  The results appear as follows:


☑ Define blanks

Missing values
0
999

□ Range
☑ Null
□ White space

- Add a **Data Audit** node downstream from the **Type** node, and run it.

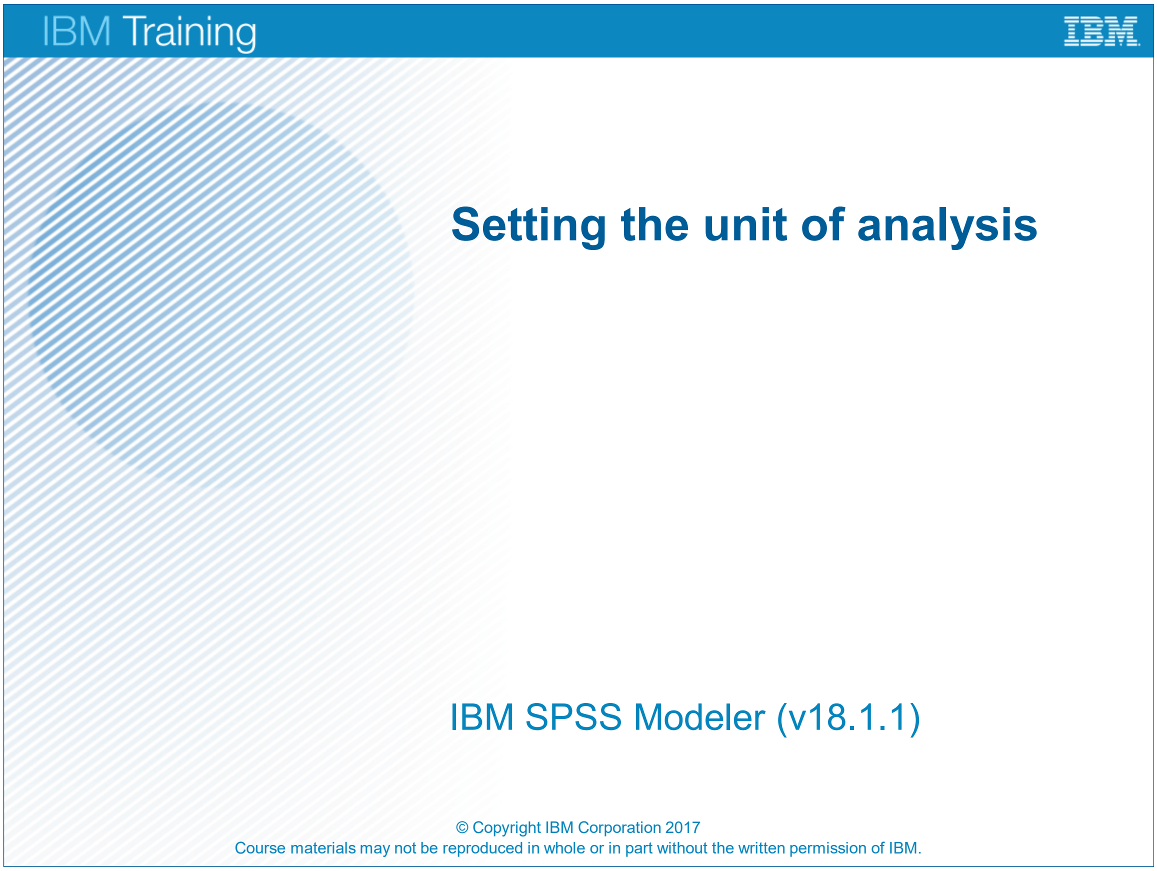
  Mean AGE is 25.145, which is the correct number (values 0 and 999 are not included in the computation because they are declared as blank values).

Task 4. Exit IBM SPSS Modeler.

- From the **File** menu, click **Exit**, and then exit IBM SPSS Modeler without saving anything.

You will find the solution results in the **C:\Training\0A008\05-Understanding_the_data\Solutions** folder.

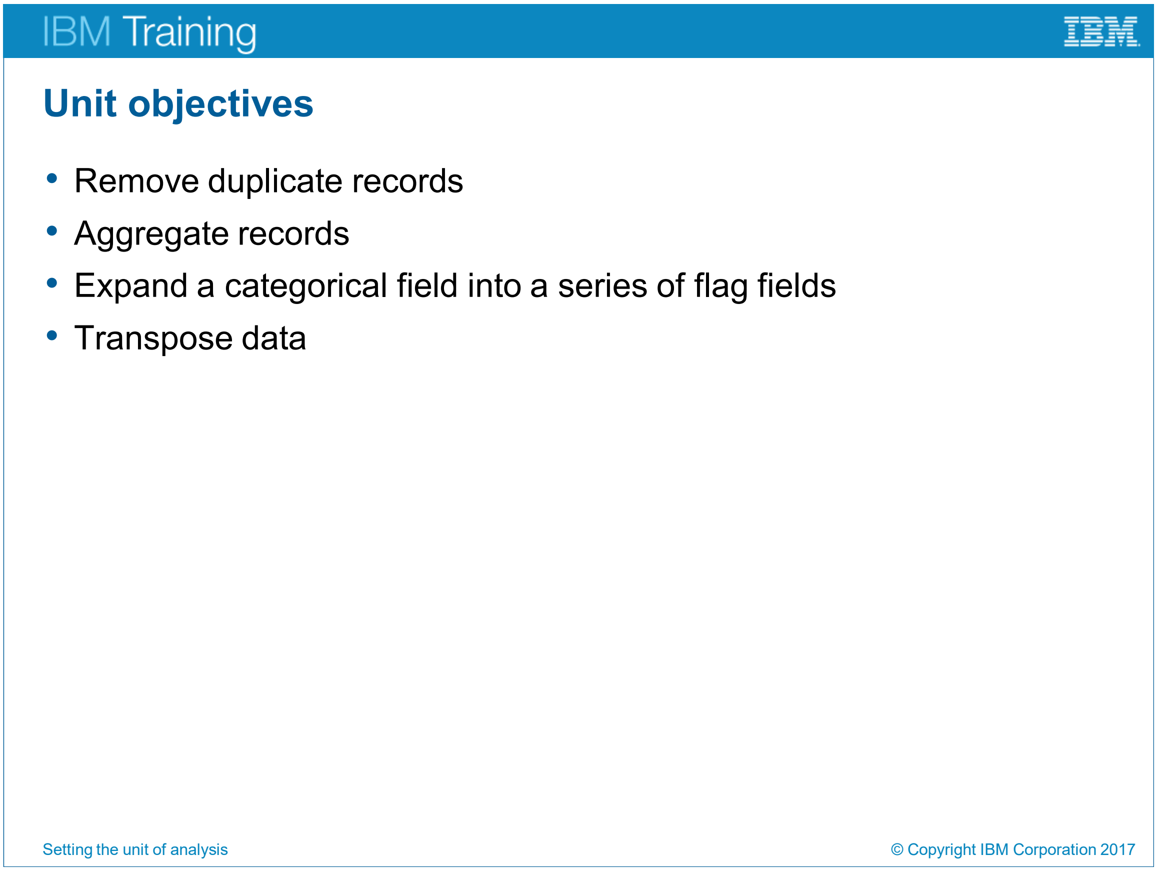
Unit 6 Setting the unit of analysis



IBM Training

Setting the unit of analysis

IBM SPSS Modeler (v18.1.1)



IBM Training

Unit objectives

- Remove duplicate records
- Aggregate records
- Expand a categorical field into a series of flag fields
- Transpose data

Setting the unit of analysis

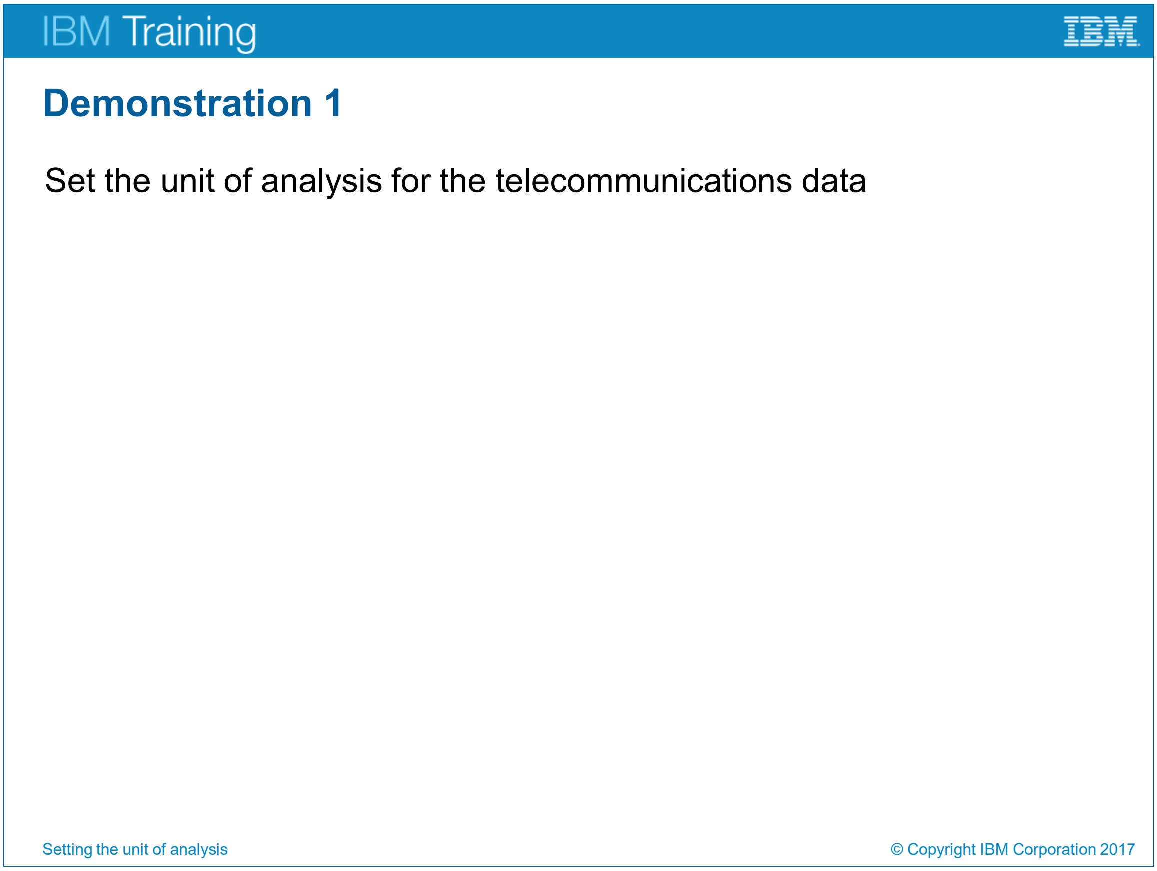
© Copyright IBM Corporation 2017

Unit objectives

After importing and exploring the data, the next task is to set the unit of analysis, one of the tasks in the Data Preparation stage in the CRISP-DM process model. This unit presents various methods how you can set the unit of analysis.

Before reviewing this unit you should be familiar with:

- CRISP-DM
- IBM SPSS Modeler streams, nodes and palettes
- Methods to collect initial data
- Measurement levels and storages
- Methods to explore the data



IBM Training

Demonstration 1

Set the unit of analysis for the telecommunications data

Setting the unit of analysis

© Copyright IBM Corporation 2017

Demonstration 1: Set the unit of analysis for the telecommunications data

Demonstration 1: Set the unit of analysis for the telecommunications data

Purpose:
You work as a data scientist for a telecommunications firm. It is your job, in order to combine the datasets later for modeling, to remove duplicate records in the customer dataset and to transform a transactional dataset into a dataset that has one record per customer.

Data file:	**telco x customer data.xlsx**
	telco x products.dat
Data folder:	**C:\Training\0A008**
Stream file:	**unit_06_demonstration_1_start.str**
Stream file folder:	**C:\Training\0A008\06-Setting_the_unit_of_analysis\Start**

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

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.
2. When a welcome window appears, click **Cancel**.

   If you have already configured IBM SPSS Modeler in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A008** folder, and then click **Set**.

Task 2. Import and examine the data.

1. From the **File** menu, click **Open Stream**, navigate to **C:\Training\0A008\06-Setting_then_unit_of_Analysis\Start** and then open **unit_06_demonstration_1_start.str**.

   You will explore the data to find out if there are any duplicate records.
2. Run the **Table** node that is downstream from the **Excel** node.

   The Table output window opens.
3. In the **Table** output window, click **Search** ; type **K136330** in the text box, and then click **Find** .
4. Scroll down a few records.

   Customer K136330 has four identical records.
5. Close the **Table** output window.

Task 3. Explore the Distinct dialog box.

1. From the **Record Ops** palette, add a **Distinct** node downstream from the **Excel** source node named **telco x customer data.xlsx**.
2. Edit the **Distinct** node.
3. Click the **Settings** tab, if necessary.

   Under Key fields for grouping, specify the field(s) that defines (define) how records must be grouped to form one output record.

   You can sort the records within each group, ascending or descending, to ensure that a particular record is the first record in the group. Sorting records in the Distinct node itself is more efficient than adding a Sort node upstream from the Distinct node.

   Mode controls how records are formed in the output dataset.
4. Click the **Mode** drop down menu.

   The Include only the first record in each group option retains only the first record of the group. Use this option if you want to remove duplicate records.

   The Discard only the first record in each group option removes the first record in the group from the dataset and retains the rest. Use this option to examine the duplicate records.

   The Create a composite record for each group option activates the Composite tab for further specifications.
5. Click **Create a composite record for each group**.
6. Click the **Composite** tab.

   Per field, specify how the new field must be built from the source field. For example, you can have the first record's value of GENDER, the last record's value of AGE, and the maximum value of DROPPED_CALLS. More advanced options are also available. For example, in a dataset where you have a record for each product purchased by the customer, you can output the most frequent purchased product per customer. To access these advanced options, click the cell for the field in question, and then click Custom.

   The Composite tab also provides the option to create an extra field, named Record_Count by default, which returns the number of input records that were grouped to form an output record.
7. Click the **Settings** tab.

Task 4. Use Distinct to remove duplicate records.

1. Beside **Mode**, select **Include only the first record in each group**.

   Records are duplicates if values for all fields are the same.

2. Under **Key fields for grouping**, click **Pick from the set of available fields** , click **All** , and then click **OK** to return to the main dialog box.

   Note: From this point forward, the instruction will just be to select the field(s).

3. Close the **Distinct** dialog box.
4. From the **Output** palette, add a **Table** node downstream from the **Distinct** node, and then run the **Table** node.

   The Table output window opens.

5. In the **Table** output window, click **Search**; type **K136330** and then click **Find**.
6. Scroll down a few records.

   You will now have only one record for this customer.

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

Task 5. Explore the Aggregate dialog box.

You will examine another data file for the demonstration in this task.

1. Run the **Table** node that is downstream from the **Var. File** node.

   A customer has as many records as he has products. For example, customer K100010 purchased 4 products.

2. Click **OK** to close table.
3. From the **Record Ops** palette, add an **Aggregate** node downstream from the **Var. File** node.
4. Edit the **Aggregate** node.

   Similar to the Distinct node, a group of records is defined by key fields. A key field such as CUSTOMER_ID will group the records of a customer into one record.

   When no key field is specified, the aggregation will be over all the records in the dataset, and thus will output one record for the entire dataset.

   If you want to retain a field value that is constant for all records in a group, such as gender for a customer, add the field to the list of key fields.

   When the data is already sorted on the key field(s), you can improve performance by clicking the Optimization tab and enabling the Keys are contiguous option.

Under Basic Aggregates, select the fields that you want to aggregate values for and select the statistic(s). You can choose Sum, Mean, Min, Max, SDev (standard deviation), Median, Count (the number of records having a non-$null$ value for the field in question), Variance, and 1st and 3rd Quartile (25th and 75th percentile). Enable the Include record count in field option to create a field that stores the number of records aggregated to form an output record.

Aggregate Expressions let you compute extra statistics. For example, although the range (the difference between maximum and minimum value) is not available as one of the statistics, it can be created by using the built-in functions MAX en MIN. When the data source is a database, you can also use functions supported by the database.

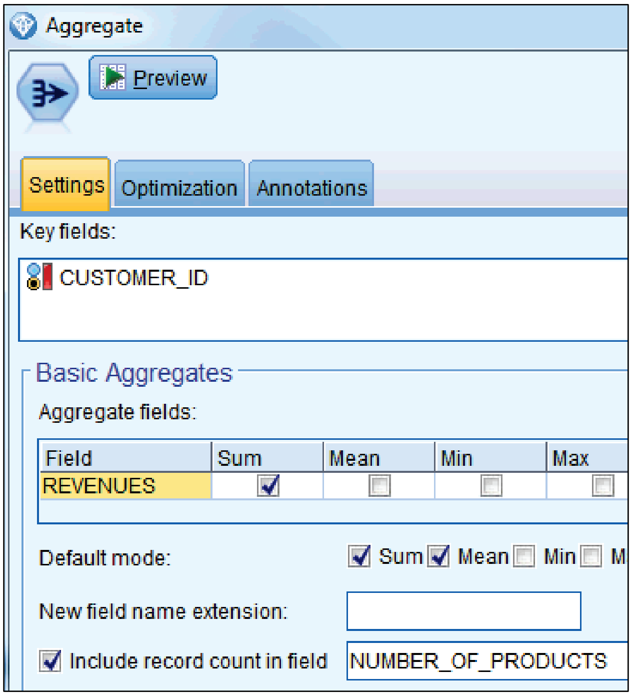
Note: User-defined blanks are included in the computation of the statistics. Think of having defined 999 as blank value for AGE. Requesting the Max statistic for AGE will return 999 although this value is declared as blank. Also the Mean statistic will be affected by this blank value. Therefore it is recommended to nullify blank values upstream from the Aggregate node.

Task 6. Obtain the number of products purchased and the total revenues per customer.

1. Under **Key fields**, select **CUSTOMER_ID**.
2. In the **Basic Aggregates** section, under **Aggregate fields**, select **REVENUES**, and then select **Sum** as only statistic.

3. In **Include record count in field**, rename the field to **NUMBER_OF_PRODUCTS**.

   The results appear as follows:



Key fields:

CUSTOMER_ID

Basic Aggregates

Aggregate fields:

Field	Sum	Mean	Min	Max
REVENUES	☑	□	□	□

Default mode: ☑ Sum ☑ Mean □ Min □ M

New field name extension:

☑ Include record count in field NUMBER_OF_PRODUCTS

4. Click **Preview**.

   There is one record per customer, with the sum of revenues stored in REVENUES_Sum. The first customer purchased 4 products, with a total revenue of 250.

5. Close the **Preview** output window.

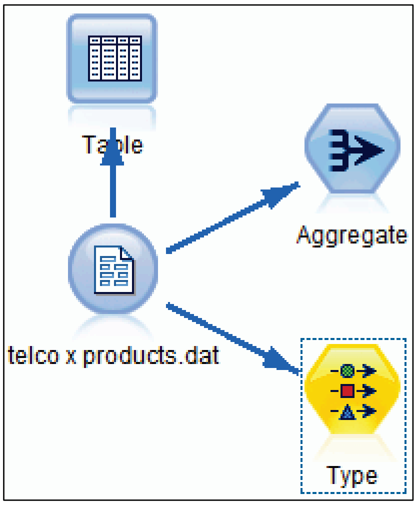
6. Close the **Aggregate** dialog box.

Task 7. Explore the SetToFlag dialog box.

When you use the SetToFlag node, it is important to instantiate the data first, so that the values of the categorical field are available in the SetToFlag node. You can instantiate the data in the Types tab of a data source node, or in a separate Type node upstream from the SetToFlag node. In this task the latter is preferred to emphasize that instantiation is a separate step.

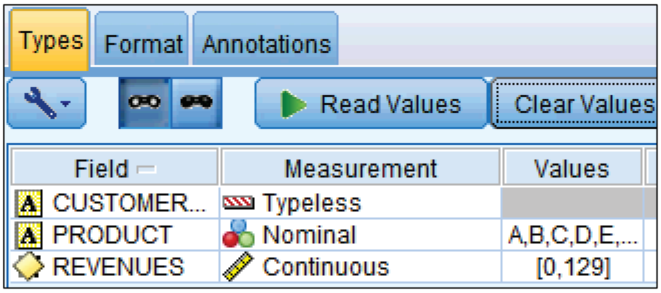
1. From the **Field Ops** palette, add a **Type** node downstream from the **Var. File** node named **telco x products.dat**.

   The results appear similar to the following:



2. Edit the **Type** node, and then click **Read Values**.

   The results appear similar to the following:



Field	Measurement	Values
CUSTOMER...	Typeless	
PRODUCT	Nominal	A,B,C,D,E,...
REVENUES	Continuous	[0,129]

   PRODUCT is instantiated.

3. Close the **Type** dialog box.
4. From the **Field Ops** palette, add a **SetToFlag** node downstream from the **Type** node.

5. Edit the **SetToFlag** node.

   On the Settings tab, under Set fields, select the categorical field that you want to expand into flags. The area under Available set values will be populated with the categories of the selected categorical field, provided that the field is instantiated. Move the categories that you want to create flag fields for to the Create flag fields area. Optionally, extend the field name for the new flag fields, either as suffix or prefix.

   If preferred, you can change the default true and false value.

   By default, the number of records output from the SetToFlag node will be the same as the number of records input to the node, because records will not be grouped. Enable the option Aggregate keys and select the appropriate key field(s) to group records.

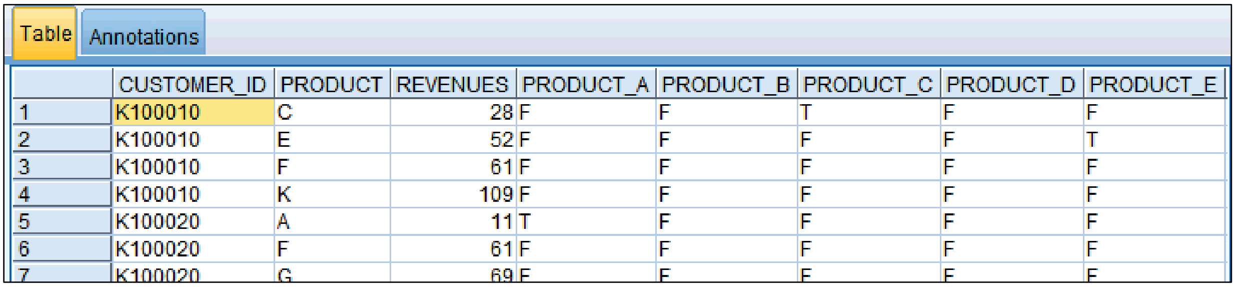
Task 8. Expand a nominal field into a series of flag fields with one record per customer.

You will create a dataset where PRODUCT is expanded into a series of flags, with only one record for each customer. To accomplish this, you will use the SetToFlag node.

1. Under **Set fields**, select **PRODUCT**.
2. Under **Available set values**, press **Ctrl+A** to select all values, and then click **Create selected flag fields** → to move them into the **Create flag fields** box.

   At this point the flag fields have been created, but there is not yet one record per customer. To verify that, preview the data.
3. Click **Preview**.

   The results appear similar to the following:

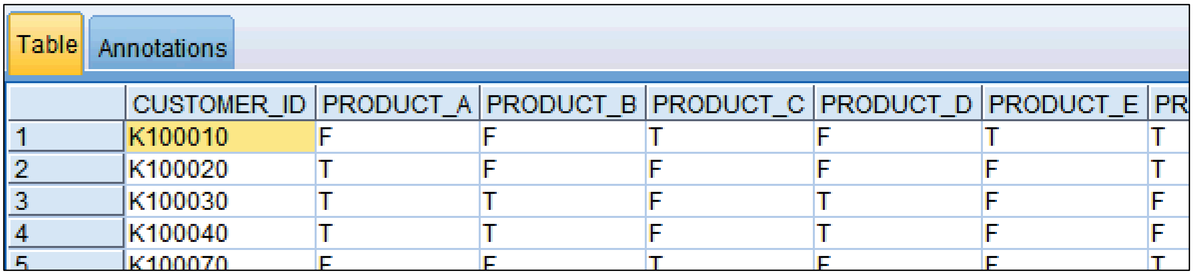


	CUSTOMER_ID	PRODUCT	REVENUES	PRODUCT_A	PRODUCT_B	PRODUCT_C	PRODUCT_D	PRODUCT_E
1	K100010	C	28	F	F	T	F	F
2	K100010	E	52	F	F	F	F	T
3	K100010	F	61	F	F	F	F	F
4	K100010	K	109	F	F	F	F	F
5	K100020	A	11	T	F	F	F	F
6	K100020	F	61	F	F	F	F	F
7	K100020	G	69	F	F	F	F	F

   A customer still has as many records as he has products. Notice that at most one value is true for a record.
4. Close the **Preview** output window.

5. Enable the **Aggregate keys** check box, and then select **CUSTOMER_ID** as field to aggregate on.
6. Click **Preview**.

   The results appear as follows:



	CUSTOMER_ID	PRODUCT_A	PRODUCT_B	PRODUCT_C	PRODUCT_D	PRODUCT_E	PR
1	K100010	F	F	T	F	T	T
2	K100020	T	F	F	F	F	T
3	K100030	T	T	F	T	F	F
4	K100040	T	T	F	T	F	F
5	K100070	F	F	T	F	F	T

   The flag fields have been created, and there is one record per customer.
7. Close the **Preview** output window.
8. Close the **SetToFlag** dialog box.

Task 9. Explore the Transpose dialog box.

The Transpose node requires that the data is instantiated upstream, so that the values are known.

1. From the **Field Ops** palette, add a **Transpose** node downstream from the **Type** node.
2. Edit the **Transpose** node.
3. Click the **Transpose method** drop down menu.

   The Both fields and records option let you change the role of rows and columns. For example, you might have two fields, where the values of the first field are placed in the first row, and the values of the second field in the second row. Thus, you will have two rows and as many columns as you have observations. To analyze this data, you need to transpose the data so that the columns become the rows (records), and the rows the columns (fields). This option can be very useful when you work with time series data, because many time series data is formatted this way.

   The Records to fields option lets you group records, as demonstrated in the next task.

   The Fields to records option will create multiple records from one record. For example, a dataset might have one record and two fields, but in order to perform the correct statistical test, you want two records and one field.

   The dialog box will reflect the mode that you select. Refer to the Help for details about the Both fields and records and Fields to record options.

   Note: The Records to fields and Fields to records methods are only supported on Windows 64-bit, Linux 64-bit, and Mac.

Task 10. Expand a continuous field into a series of continuous fields with one record per customer.

You will create a dataset where REVENUES is expanded into fields storing REVENUES per PRODUCT, with only one record for each customer. To accomplish this, you will use the Transpose node, Records to field option.

1. Beside **Transpose** method, select **Records to fields**.
2. In the **Index** box, specify the field that defines the groups, **CUSTOMER_ID** in this example.
3. In the **Fields** box, specify the categorical field by which the continuous field must be restructured, **PRODUCT** in this example.
4. In the **Value** box, specify the continuous field that must be expanded into a series of new fields, **REVENUES** in this example.

   In the lower left area of the dialog box you can specify the statistic that must be computed across records within the same group. This is relevant when a customer has multiple records with the same product. Per customer you can output the mean REVENUES for that product, the sum, minimum, maximum, median, or count.

   In this example dataset, a customer does not have the same product multiple times, so it will make no difference whether you select mean, sum, max, min, or median. Therefore you can keep the default, Mean.
5. Click **Preview**.

   The continuous field REVENUES is expanded into a series of fields that store revenues per product. For example, customer K100010 purchased product C (revenues 28) and product E (revenues 52), among others. This customer did not purchase product A, B, and D, etcetera.
6. Close the **Preview** output window.
7. Close the **Transpose** dialog box.

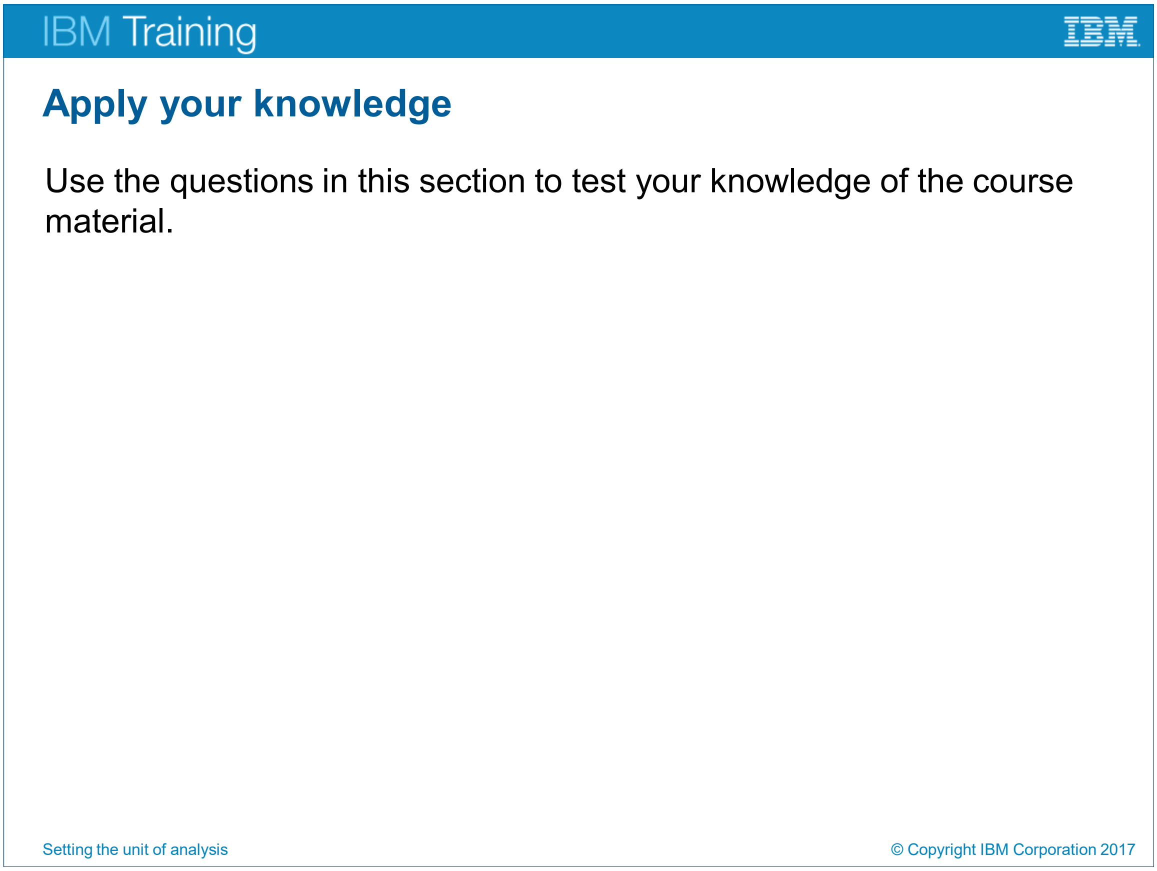
   This completes the demonstration for this unit. You will create a clean state for the exercise.

8. From the **File** menu, click **Close Stream**, and then click **No** when asked to save the stream.
9. From the **File** menu, click **New Stream**.

   Leave IBM SPSS Modeler open for the exercise.

Results:
You removed duplicate records in the customer dataset and transformed a transactional dataset into a dataset that has one record per customer. This enables you to merge these files in the next stage, and run models later.

You will find the completed stream in the **C:\Training\0A008\06-Setting_the_unit_of_analysis\Solutions** folder.



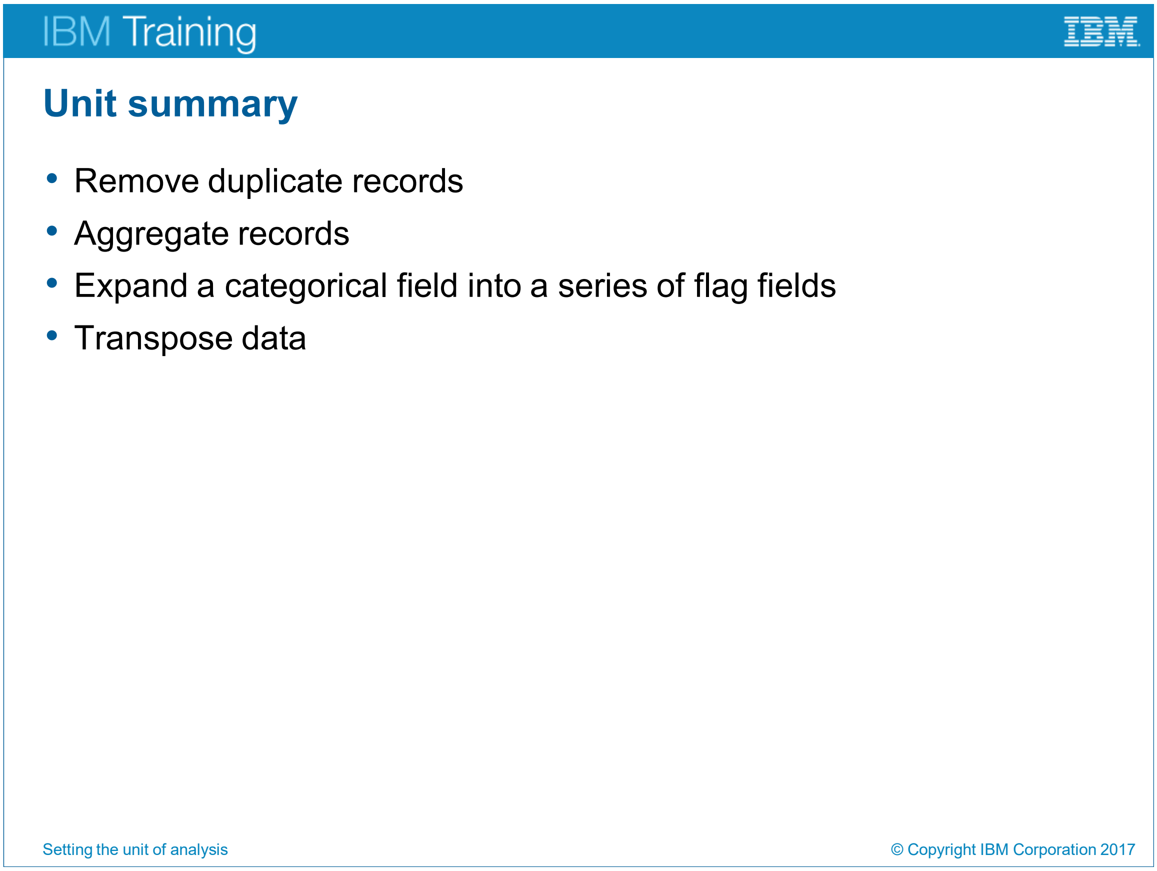
IBM Training

Apply your knowledge

Use the questions in this section to test your knowledge of the course material.

Setting the unit of analysis

Apply your knowledge



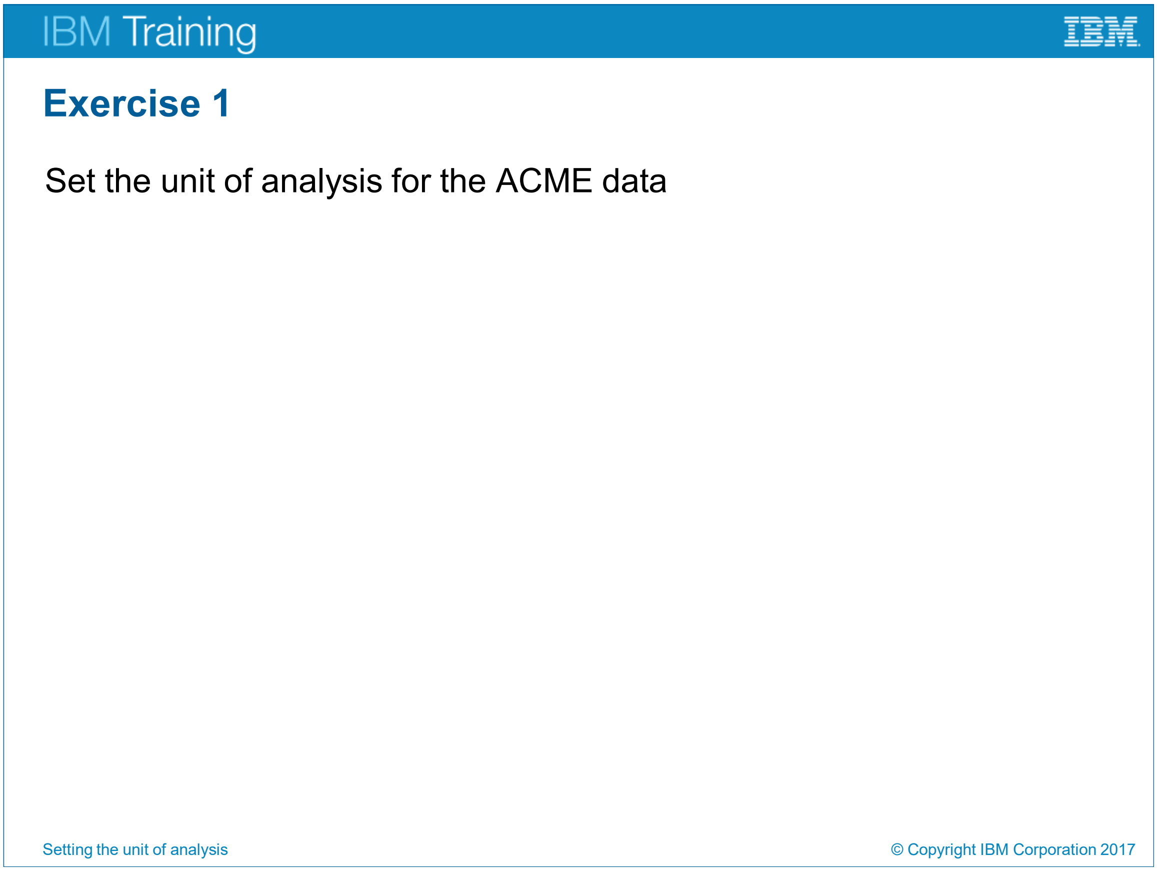
IBM Training

Unit summary

- Remove duplicate records
- Aggregate records
- Expand a categorical field into a series of flag fields
- Transpose data

Setting the unit of analysis

Unit summary



IBM Training

Exercise 1

Set the unit of analysis for the ACME data

Setting the unit of analysis

Exercise 1: Set the unit of analysis for the ACME data

Exercise 1: Set the unit of analysis for the ACME data

Data files:	**ACME customer data.xlsx**
	ACME purchases 1999 - 2004.dat
	ACME orderlines 1999 - 2004.sav
	ACME mailing history.xlsx
Data folder:	**C:\Training\0A008**
Stream file:	**unit_06_exercise_1_start.str**
Stream file folder:	**C:\Training\0A008\06-Setting_the_unit_of_analysis\Start**

You work at the database marketing department of ACME, a company that sells sports products. It is your job to import data from several sources, and create datasets with the required unit of analysis (one record per customer), so that these datasets can be merged later.

You will use an existing stream, unit_06_exercise_1_start.str, to import the data needed for this exercise.

- Open **unit_06_exercise_1_start.str**, located in the **C:\Training\0A008\06-Setting_the_unit_of_analysis\Start** folder.
- For the data source storing customer data, **ACME customer data.xlsx**, run the **Table** node that is downstream and note the number of records. Then, remove duplicate records.

  To check your results: The de-duplicated dataset has 30,000 records.
- From the data source storing purchases, **ACME purchases 1999 - 2004.dat** (each record represents a purchase made by a customer), create a dataset that has only one record per customer, with the most recent order date and the number of purchases made by the customer.

  To check your results: The new dataset has 2,739 records.
- From the data source storing orders, **ACME orderlines 1999 - 2004.sav** (each record represents one of the items purchased at one point in time by the customer), create a dataset that has only one record per purchase, with the total price per purchase (the sum of the PRICE field), and the number of items bought for each purchase.

  To check your results: The new dataset has 4,018 records.

- From the data source storing the mailing history, **ACME mailing history.xlsx** (each record represents a mailing sent to a customer), create a dataset with one record per customer, with fields indicating whether the customer was included in the first, second and/or third mailing.

  To check your results: The new dataset has 16,740 records.
- Exit IBM SPSS Modeler without saving anything.

For more information about where to work and the exercise results, refer to the Tasks and Results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1:
Tasks and Results

Task 1. Open the stream file.

- From the **File** menu, click **Open Stream**, navigate to **C:\Training\0A008\06-Setting_the_unit_of_analysis\Start**, select **unit_06_exercise_1_start.str**, and then click **Open**.

Task 2. Remove duplicate records in ACME's customer data.

- Run the **Table** node, downstream from the **ACME customer data.xlsx** source node.

  The dataset has 30,017 records.
- From the **Record Ops** palette, add a **Distinct** node downstream from the **Excel** source node named **ACME customer data.xls**.
- Edit the **Distinct** node, and then:
  - for **Mode**, select **Include only the first record in each group**
  - for **Key fields for grouping**, select all fields (select all fields, because records are identical when values for all fields are the same)
- From the **Output** palette, add a **Table** node downstream from the **Distinct** node, and then run the **Table** node.

  This will show that you have 30,000 records in the dataset.

Task 3. Create a dataset where customers are unique in ACME's purchases data.

- From the **Record Ops** palette, add an **Aggregate** node downstream from the **Var. File** source node.
- Edit the **Aggregate** node, and then:
  - under **Key fields**, select **CUSTOMER_ID**
  - under **Aggregate fields**, select **ORDERDATE**, and then select **Max**
  - enable the **Include record count in field** option, and then type a name such as **NUMBER OF PURCHASES**

  Note: Instead of using Aggregate, you can use the Distinct node, with the option to create a composite record.
- Run a **Table** node to verify that the number of records is 2,739.

Task 4. Create a dataset where purchases are unique in ACME's order lines data.

- From the **Record Ops** palette, add an **Aggregate** node downstream from the **Statistics File** source node.
- Edit the **Aggregate** node, and then:
  - for **Key fields**, select **PURCHASE_ID**
  - for **Aggregate fields**, select **PRICE**, and then for statistic, select **Sum**
  - enable the **Include record count in field** option; and type a name such as **NUMBER OF ITEMS PURCHASED**

Note: Instead of using Aggregate, you can use the Distinct node, with the option to create a composite record.

- From the **Output** palette, add a **Table** node downstream from the **Aggregate** node, and then run the **Table** node.

  This will show that you have 4,018 records in the dataset.

Task 5. Create a dataset where customers are unique in ACME's mailing history data.

- From the **Field Ops** palette, add a **Type** node downstream from the **Excel** source node named **ACME mailing history.xlsx**.
- Edit the **Type** node, and then click **Read Values** to instantiate the data (the values for the mailing field will then be available in the SetToFlag node that will be added downstream).
- From the **Field Ops** palette, add a **SetToFlag** node downstream from the **Type** node.
- Edit the **SetToFlag** node and then:
  - for **Set fields**, select **mailing**
  - under **Available set values**, select all values and move them into the **Create flag fields** box
  - enable the option **Aggregate keys**, and select **customer_id**
- From the **Output** palette, add a **Table** node downstream from the **SetToFlags** node, and then run the **Table** node.

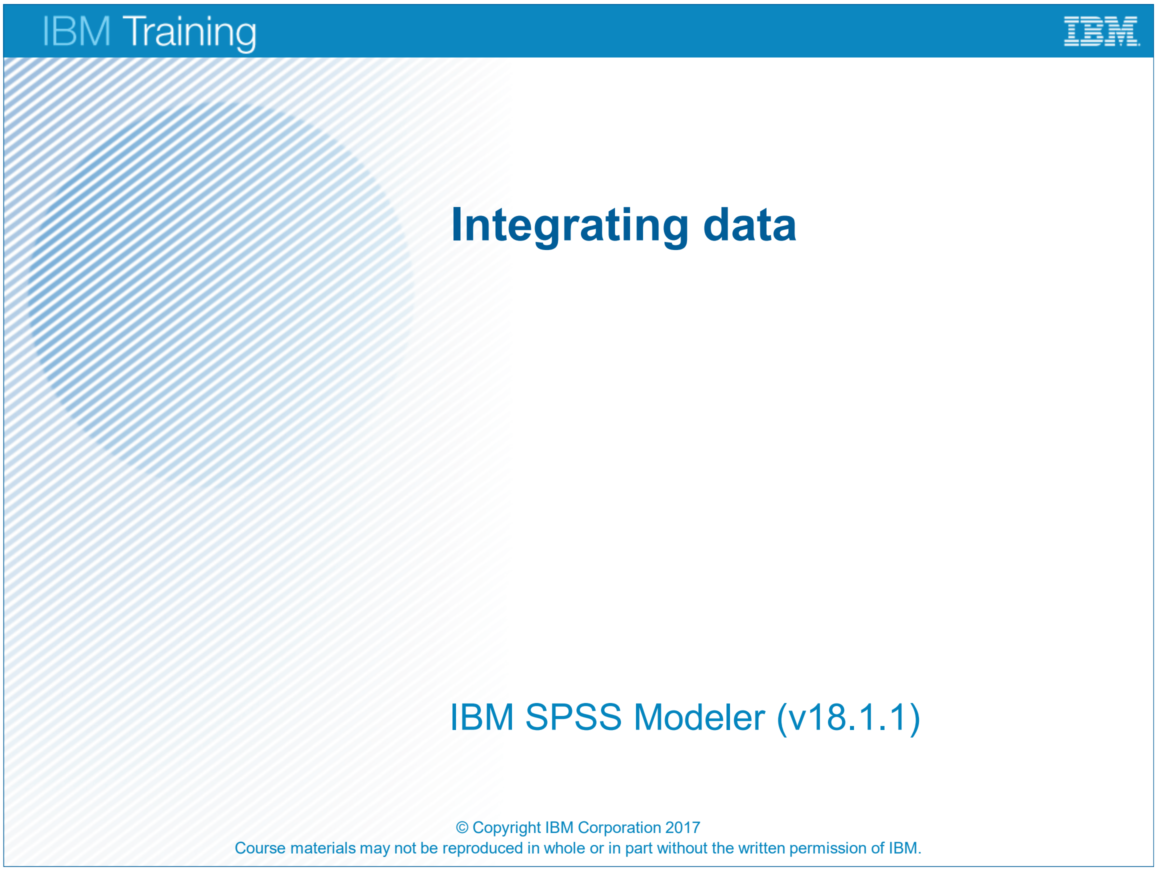
  This will show that you have 16,700 records in the dataset.

Task 6. Exit IBM SPSS Modeler.

- From the **File** menu, click **Exit**, and then exit IBM SPSS Modeler without saving anything.

You will find the solution results in the **C:\Training\0A008\06-Setting_the_unit_of_analysis\Solutions** folder.

Unit 7 Integrating data

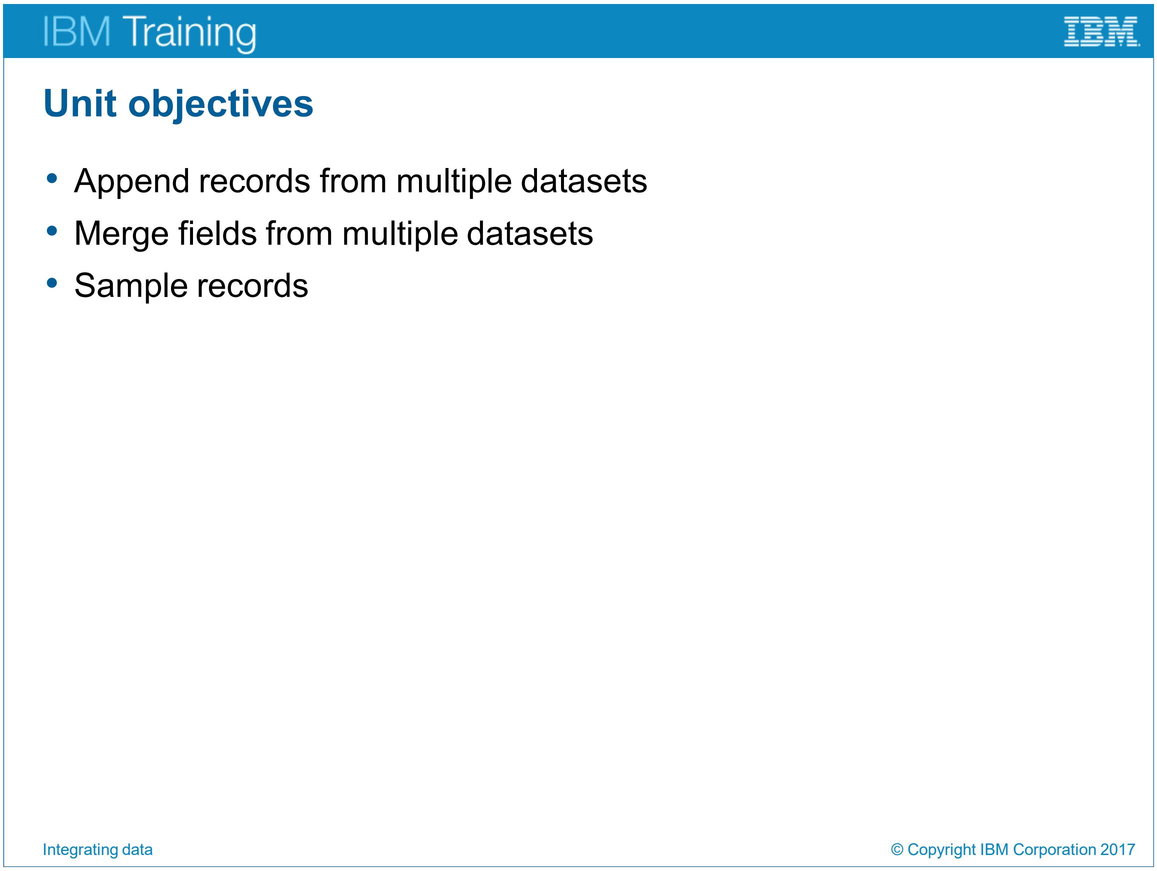


IBM Training

IBM

Integrating data

IBM SPSS Modeler (v18.1.1)



IBM Training

Unit objectives

- Append records from multiple datasets
- Merge fields from multiple datasets
- Sample records

Integrating data

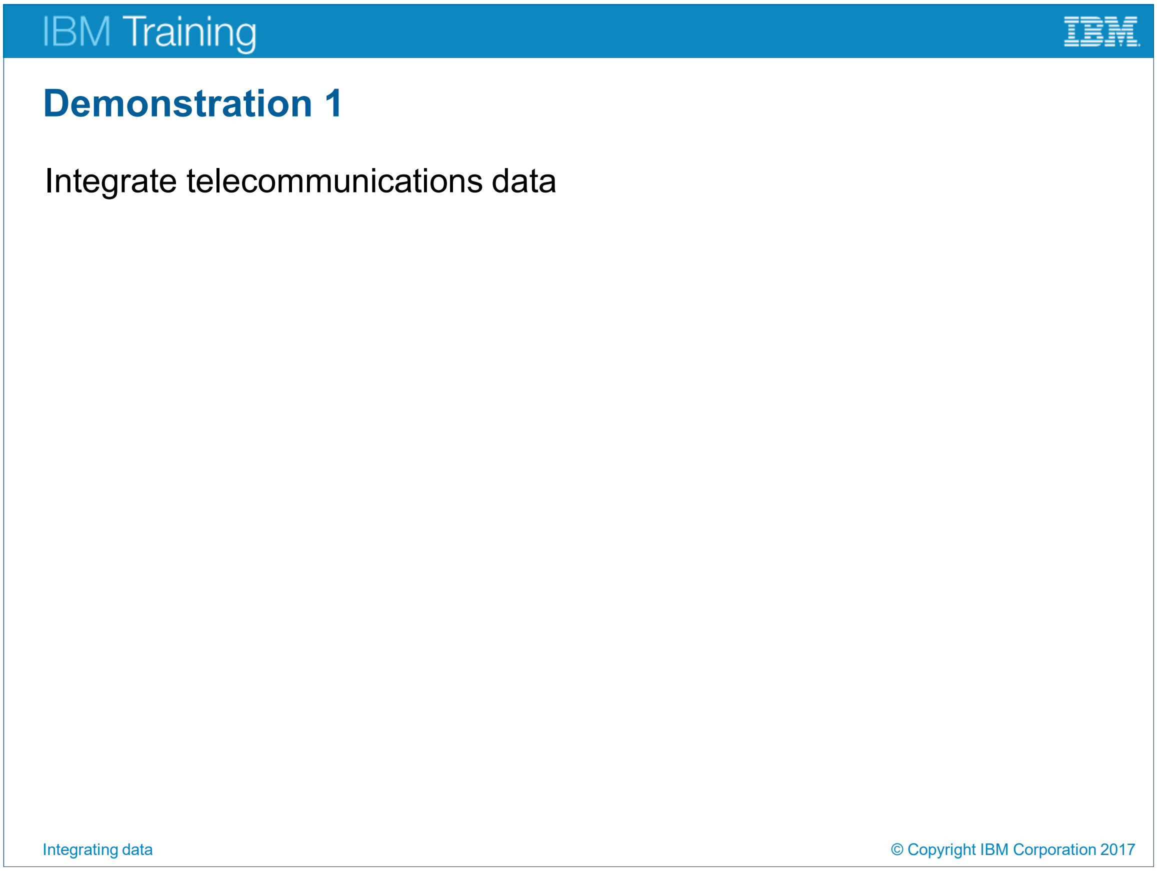
Unit objectives

Similar pieces of information may be stored in different datasets. These datasets must be combined into a single dataset for analyses. Typically, datasets are combined after the unit of analysis is set correctly for each of the datasets involved.

Combining datasets is referred to in CRISP-DM as integrating data.

Before reviewing this unit you should be familiar with:

- CRISP-DM
- IBM SPSS Modeler streams, nodes and palettes
- Methods to collect initial data
- Measurement levels and storages
- Methods to explore the data
- Methods to set the unit of analysis



IBM Training

Demonstration 1

Integrate telecommunications data

Integrating data

Demonstration 1: Integrate telecommunications data

Demonstration 1: Integrate telecommunications data

Purpose:
You work as a data scientist for a telecommunications firm and have to combine a number of datasets into a single dataset for analyses and modeling later.

Data file:	**telco x call data q1.sav**
	telco x call data q2.sav
	telco x customer data.xlsx
	telco x products.dat
	telco x tariff.dat
Data folder:	**C:\Training\0A008**
Stream file:	**unit_07_demonstration_1_start.str**
Stream file folder:	**C:\Training\0A008\07-Integrating_data\Start**

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

1. From the **Start** menu, expand, and then click **IBM SPSS Modeler 18.1**.
2. When a welcome window appears, click **Cancel**.

   If you have already configured IBM SPSS Modeler in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A008** folder, and then click **Set**.

Task 2. Open the stream.

1. From the **File** menu, click **Open Stream**, navigate to the **C:\Training\0A008\07-Integrating_data\Start** folder, click **unit_07_demonstration_1_start.str** and then click **Open**.

Task 3. Append records from two datasets.

You will combine two IBM SPSS Statistics .sav files that both store call detail records.

1. From the **Record Ops** palette, place an **Append** node on the stream canvas to the **right** of the two **Statistics File** sources nodes named **telco x call data q1.sav** and **telco x call data q2.sav**.

   Note: These data source nodes are on the stream canvas after you have opened **unit_07_demonstration_1_start.str**.

2. Connect the **Statistics File** node named **telco x call data q1.sav** to the **Append** node.
3. Connect the **Statistics File** node named **telco x call data q2.sav** to the **Append** node.
4. Edit the **Append** node.
5. Click the **Append** tab, if necessary.

   The Append tab controls how the datasets are appended.

   For Include fields from, in the lower part of the dialog box, select whether you want to use a leading dataset (the Main dataset only option) or that you want all datasets to play equal roles (the All datasets option).

   Fields that will be included in the new dataset are listed in the Output Field column.

   It is recommended to tag records by adding a field to the combined dataset whose values indicate the source dataset for each record.

   Fields can be matched by position or name (the default). In general, matching fields by position is not recommended. When matching on name, there is an extra option to enable case sensitivity. By default, the same field names, although in a different case, will match.
6. Click the **Inputs** tab.

   You can specify the main dataset (if that option was selected), and the order wherein the datasets are appended.

   You will keep these settings.
7. Click the **Append** tab.

   In this example, the field names match except those that refer to international phone calling. Fields unique to the second dataset (telco x call data q2.sav) will be lost and records coming from the second dataset will be assigned the undefined ($null$) value for INTERNATIONAL_CALLS and INTERNATIONAL_MINS.

   You can make the second dataset the main dataset by reordering the datasets in the Inputs tab. In this case, that is no solution to the problem, because data will still be lost (now for the first dataset). Neither would it help if you let both datasets play equal roles: the information about international calling would be in different fields. This can be handled downstream by further data preparation, but in this demonstration a simpler solution is preferred: INTERNATIONAL_MINS and INTERNATIONAL_CALLS in the first dataset are in the same position as INTERNAT_MINS and INTERNAT_CALLS in the second dataset, so you can match the datasets by position (in practice, you should take care in using this option).

8. Beside **Match fields by**, select **Position**.
9. Enable the **Tag records by including source dataset in field** option.
10. Rename the **Input** field to **QUARTER**.
11. Close the **Append** dialog box.

    When the Append node is processed, the first block of data that is output comes from the first dataset, followed by a block of data from the second dataset. To have a better view on the data, you will sort the data on CUSTOMER_ID and MONTH.
12. From the **Record Ops** palette, add a **Sort** node downstream from the **Append** node.
13. Edit the **Sort** node, and then under **Sort by**, click **Pick from the set of available fields** .
14. Ctrl+click **CUSTOMER_ID** and **MONTH** and then click **OK**.
15. Click **Preview**.

    The Preview output window shows that the first customer has 6 records of data, one record for each month. QUARTER indicates the data source.
16. Close the **Preview** output window.
17. Close the **Sort** dialog box.

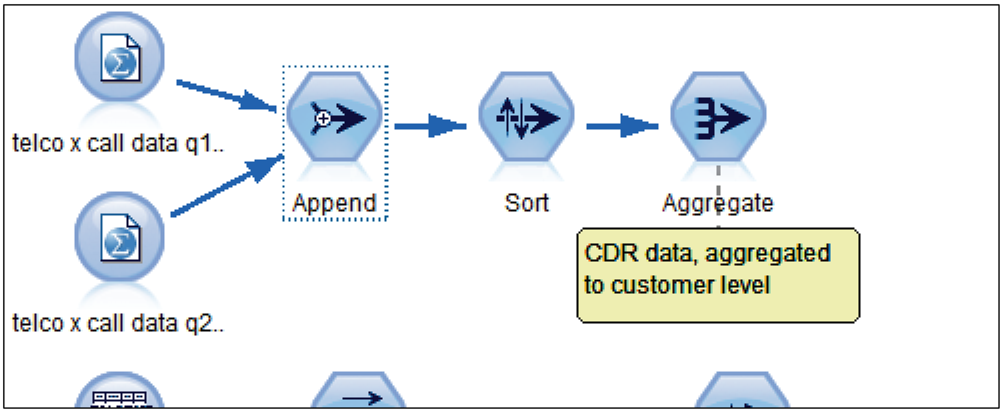
    Appending the two datasets has been completed successfully. For the analyses later in the project, however, a data structure is required with one record per customer. You can use the Distinct node or the Aggregate node to arrive at that data structure. Refer to the *Setting the Unit of Analysis* unit in this course for more information about each of these operations. Which method is preferred depends on the objective of the analysis. Here, you will aggregate the data to customer level by summing the values.
18. From the **Record Ops** palette, add an **Aggregate** node downstream from the **Sort** node.
19. Edit the **Aggregate** node, and under **Key fields**, select **CUSTOMER_ID**.
20. Under **Aggregate fields**, click **PEAK_CALLS**, Shift+click **INTERNATIONAL_MINS**, and then click **OK**.
21. Under **Aggregate fields**, click **PEAK_CALLS** and drag to **INTERNATIONAL_MINS**, and then click **Mean** for any field, to deselect Mean for all selected fields.

    You will not keep track of the number of records a customer has (it will always be 6).
22. Disable the **Include record count in field** option.

23. Click **Preview**.

    This dataset has one record per customer, as required.
24. Close the **Preview** output window.
25. Close he **Aggregate** dialog box.
26. Add the following text as a comment to the **Aggregate** node: **CDR data, aggregated to customer level**.

    The results appear similar to the following:



CDR data, aggregated to customer level

Task 4. Merge fields from three datasets.

Three datasets have to be merged:

- customer data (with duplicate records removed by using a Distinct node)
- call detail records (aggregated to a 6 month period in the previous task)
- product data (transformed into a dataset with one record per customer by using a SetToFlag node).

These datasets all have one record per customer, and thus can be merged on the key CUSTOMER_ID.

In this demonstration, the customer information dataset is taken as the leading dataset in the merge, so you will use a partial outer join.

1. From the **Record Ops** palette, drag a **Merge** node to the right of the **Distinct** node.
2. Connect the **Distinct** node to the **Merge** node.
3. Connect the **Aggregate** node to the **Merge** node.
4. Connect the **SetToFlag** node to the **Merge** node.

5. Edit the **Merge** node.

   The Merge tab controls how datasets are merged. You can merge records by order (not recommended), by using one or more key fields (the most common situation), or by specifying a (ranked) condition (with extra features when matching geospatial data). Refer to the Help for more details about this latter option.

   Under Possible keys, fields included in all input datasets are listed. Field names in IBM SPSS Modeler are case sensitive, and field names have to match in case otherwise they will not appear. You can use a Filter node upstream from the Merge node to rename fields, or you can match by condition to solve this issue.

   Enabling the option Combine duplicate key fields ensures that you will have only output key field with a given name. When this option is disabled, duplicate key fields must be renamed or excluded using the Filter tab in the Merge dialog box.

6. Beside **Merge Method**, select **Keys**.

7. Under **Possible keys**, click **CUSTOMER_ID**, and then click **Add keys** → to move it into the **Keys for merge** box.

   Select the type of the merge that you want.

8. Enable the **Include matching and selected non-matching records (partial outer join)** option.

   For a partial outer join, click Select to choose the leading dataset.

   The customer dataset must be the leading dataset in the merge, so you will mark this dataset as main dataset.

9. Click **Select**, and then ensure that the check box is enabled in the **Outer Join** column for **telco x customer data.xlsx**.

10. Close the **Merge: Select Dataset for Outer Join** sub dialog box.

    The Inputs tab determines the order in which the records are read. Also, it sets the main dataset for an anti join.

11. Click the **Inputs** tab.

    The customer dataset is listed first: fields from this dataset will be the first fields in the merged dataset, as preferred.

12. Click **Preview**.

    Three datasets are combined into one. The order of the fields is determined by the order of the datasets in the Inputs tab in the Merge node.

    The key field is the first field in the combined dataset. This will always be the case.

13. Scroll to the right, to **PRODUCT_A**.

    Some customers have undefined ($null$) values for all product fields. For example, all product fields are $null$ for record with ID K100050. This customer was not in the product dataset, so the undefined ($null$) value was assigned for the fields related to products.
14. Close the **Preview** output window.
15. Close the **Merge** dialog box.

Task 5. Enrich a dataset with aggregated data.

Apart from the three datasets merged in the previous task, there is additional information on tariffs. This dataset will be added to the single dataset that was created in the previous task.

1. Run the **Table** node, attached to the **Var. File** node named **telco x tariff.dat**.

   This dataset does not store customer data, but stores data on the aggregated level of tariffs. This will require a separate merge.
2. From the **Record Ops** palette, place a second **Merge** node on the stream canvas, to the right of the data source **telco x tariff.dat** (do not connect the Merge node to telco x tariff.dat).
3. Connect the **first Merge** node to the second **Merge** node.
4. Connect the **Var. File** node named **telco x tariff.dat** node to the second **Merge** node.
5. Edit the second **Merge** node.
6. For **Merge Method**, select **Keys**.
7. Under **Possible keys**, click **TARIFF**, and then move it into the **Keys for merge** box.
8. Enable the **Include matching and selected non-matching records (partial outer join)** option.
9. Click **Select**, and then ensure that the formerly merged dataset is the leading dataset.
10. Close the **Merge: Select Dataset for Outer Join** sub dialog box.
11. Click **Preview**, and then scroll all the way to the right.

    Fields from the tariff dataset are added to the (merged) customer data. Customers in the same tariff group will have the same values on the fields originating from the tariff dataset. From a database perspective, this is not ideal; however, IBM SPSS Modeler needs a rectangular data structure, so there is no way around this.
12. Close the **Preview** output window.
13. Close the **Merge** dialog box.

Task 6. Sample records.

From the combined dataset (storing data from four different sources), a random sample is drawn of approximately 10%, and the data is cached. Furthermore, to ensure that the student samples the same records as sampled in this demonstration, the value for the random seed will be fixed.

1. From the **Record Ops** palette, add a **Sample** node downstream from the second **Merge** node.
2. Right-click the **Sample** node, from the context menu, click **Cache** and then click **Enable**.
3. Edit the **Sample** node.

   IBM SPSS Modeler offers two sampling methods: simple and complex. The simple sampling method is presented in this unit. Refer to the *Advanced Data Preparation Using IBM SPSS Modeler* course for a presentation of complex sampling.

   You can either select or deselect records (the Include sample option and Discard sample option, respectively).

   You have three options for sampling: select the first n records in the dataset (where n needs to be specified), select every *n*th record or draw a random sample of a certain percentage. The latter option will draw a different sample each time that records pass through the Sample node. If you want to replicate a sample, type or generate a seed value for the algorithm, so that the same records will be sampled.
4. Under **Sample**, select **Random %**, and change the associated value to **10**.
5. Enable the **Repeatable partition assignment** option.
6. Set the **Seed** value to **1**.
7. Close the **Sample** dialog box.
8. From the **Output** palette, add a **Table** node downstream from the **Sample** node.
9. Run the **Table** node.

   The cache is filled (the document icon has turned green) and 3,252 records are sampled. The next time the Table node is executed, or any other node downstream from the Sample node, data will be taken from the cached file, returning the same 3,252 records (to be clear: in this case it is the effect of the cache, not because of the fixed seed value).
10. Click **OK** to close the Output window.

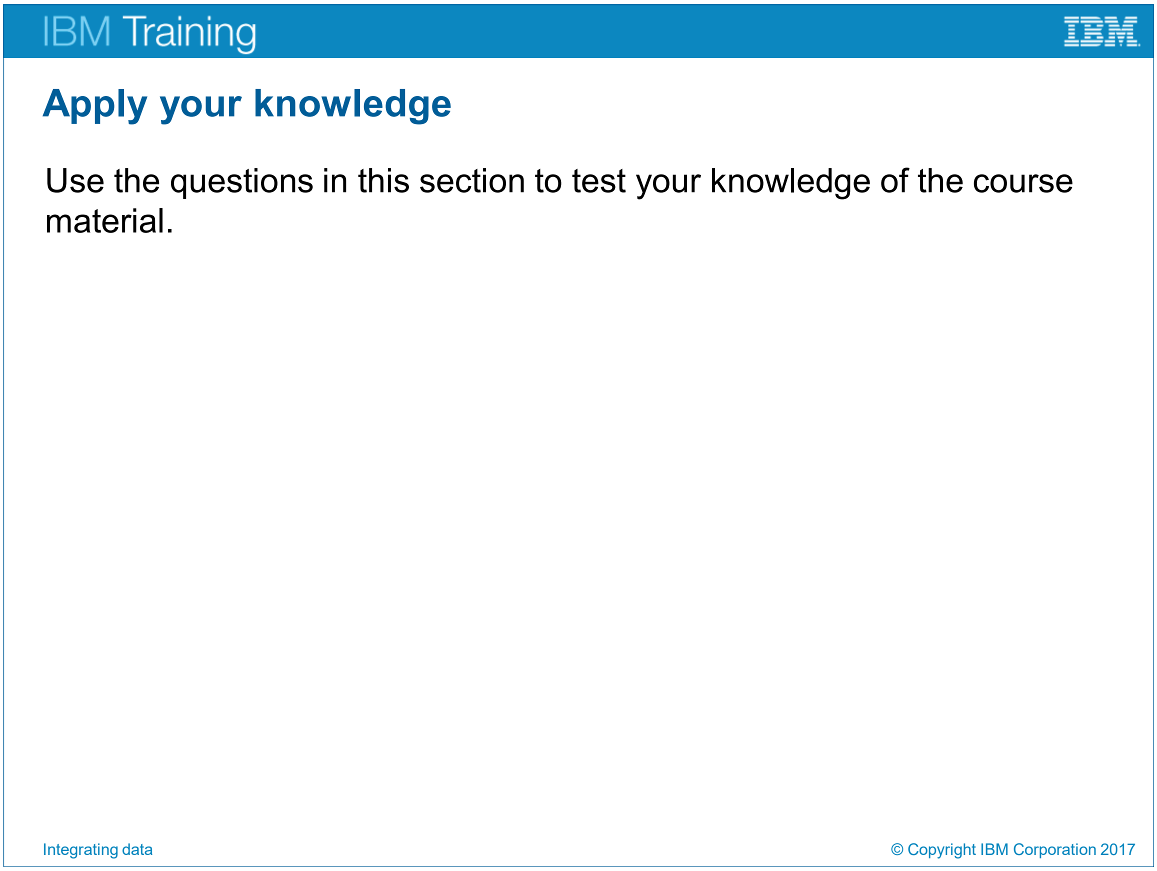
    This completes the demonstration for this unit. You will create a clean state for the exercise.

11. From the **File** menu, click **Close Stream**, and then click **No** when asked to save the stream.
12. From the **File** menu, click **New Stream**.

    Leave IBM SPSS Modeler open for the exercise.

Results:
You combined a number of datasets into a single dataset. This enables you to build models at a later stage.

You will find the completed stream in the **C:\Training\0A008\07-Integrating_data\Solutions** folder.



IBM Training

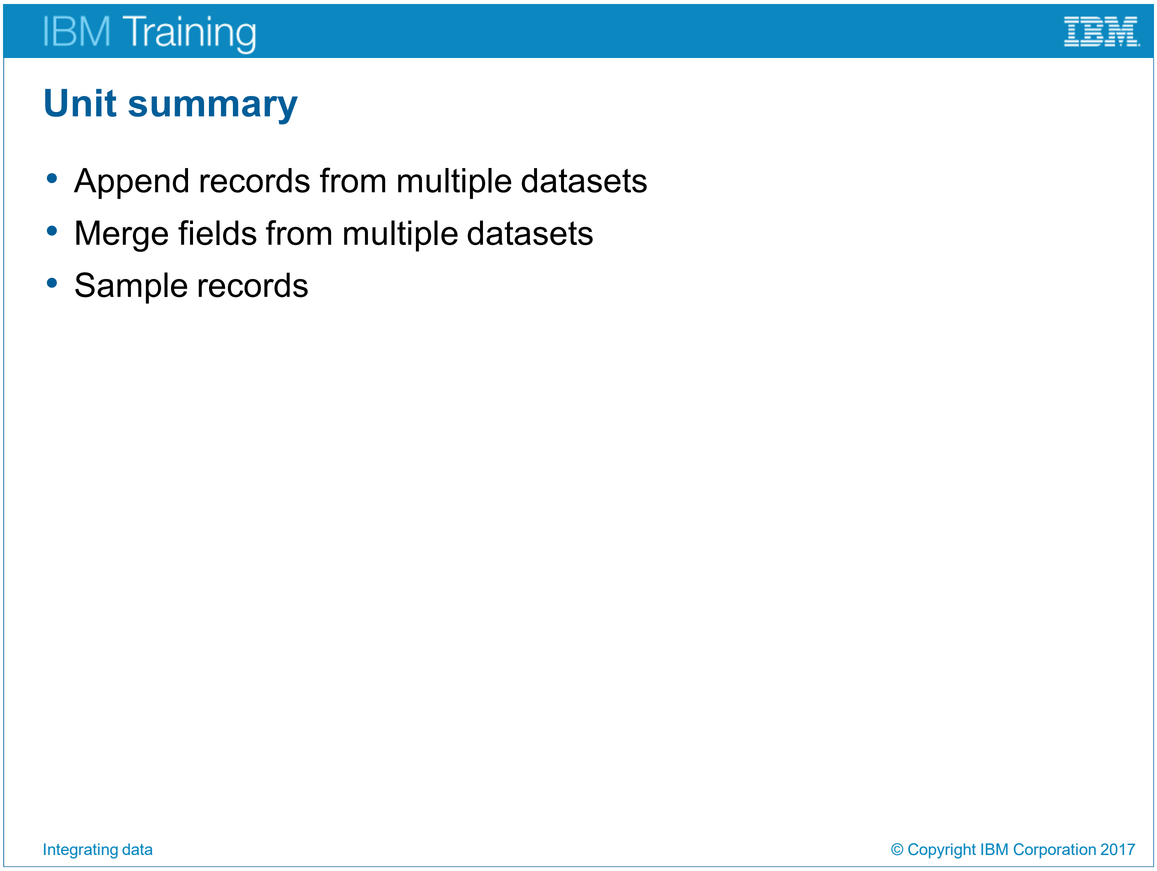
Apply your knowledge

Use the questions in this section to test your knowledge of the course material.

Integrating data

© Copyright IBM Corporation 2017

Apply your knowledge



IBM Training

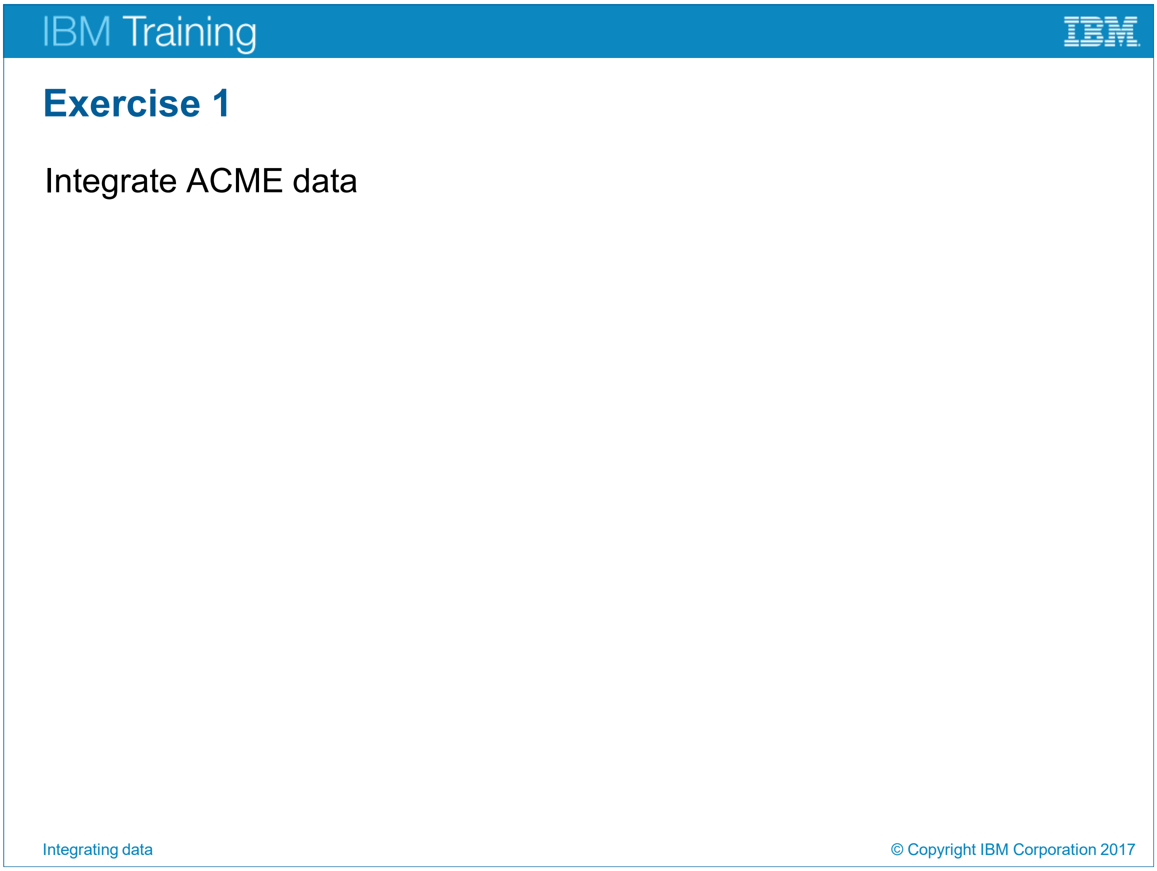
Unit summary

- Append records from multiple datasets
- Merge fields from multiple datasets
- Sample records

Integrating data

© Copyright IBM Corporation 2017

Unit summary



IBM Training

Exercise 1

Integrate ACME data

Integrating data

Exercise 1: Integrate ACME data

Exercise 1: Integrate ACME data

Data file:	**ACME purchases 1999 - 2004.dat**
	ACME purchases 2005 - 2010.dat
	ACME orderlines 1999 - 2004.sav
	ACME orderlines 2005 - 2010.sav
	ACME customer data.xlsx
	ACME mailing history.xlsx
	ACME zip data.csv
Data folder:	**C:\Training\0A008**
Stream file:	**unit_07_exercise_1_start.str**
Stream file folder:	**C:\Training\0A008\07-Integrating_data\Start**

You work for ACME, a company that sells sports products. It is your job to combine a number of datasets into a single dataset, so that models can be built using the information from all these datasets later.

- Open **unit_07_exercise_1_start.str**, located in the **C:\Training\0A008\07-Integrating_data\Start** folder.
- Two datasets store information about ACME purchases, one for the 1999 - 2004 period (**ACME purchases 1999 - 2004.dat** - 4,018 records), and another for the 2005 - 2010 period (**ACME purchases 2005 - 2010.dat** - 67,109 records). Create a single dataset storing purchases for the entire 1999 - 2010 period.

  Note: The nodes to import all the datasets are already on the stream canvas after you have opened **unit_07_exercise_1_start.str**.

  To check your results: The new dataset should have 4,018 + 67,109 = 71,127 records.
- Two datasets store information about the items that were ordered (**ACME orderlines 1999 - 2004.sav** and **ACME orderlines 2005 - 2010.sav**). Create a single dataset containing all order lines for the entire 1999 - 2010 period.

  To check your results: The new dataset should be comprised of 169,734 records.

- From the dataset storing the order lines for the entire 1999 - 2010 period (the dataset 169,734 records created in the previous task), create a dataset so that there is only one record per PURCHASE ID, and a field that stores the total price for each PURCHASE_ID (do not include a record count field).

  To check your results: There should be 71,127 records in the new dataset. Also, check that PURCHASE ID 5723 involves a total price of 948.48.

- Create a single dataset from:
  - the dataset storing the purchase information (created in the first task - 71,127 records)
  - the dataset storing aggregated information for the purchases that you created in the previous task (also 71,127 records)

  You can assume that both datasets include the same PURCHASE_ID's.

  To check your results: There should be 71,127 records in this dataset.

- For the dataset created in the previous task, ensure that there is 1 record per CUSTOMER_ID, with most recent order date, the total price the customer has paid for all purchases, and the number of purchases the customer made.

  To check your results: There should be 30,000 records in this dataset.

- Create a single dataset from:
  - the dataset having customer background information originating from **ACME customer data.xlsx** (30,000 records, already on the stream canvas)
  - the dataset that is comprised of the customer purchase information (created in the previous task)
  - the dataset with the customer mail history originating from **ACME mailing history.xlsx** (already on the stream canvas)

  The dataset with the customer information must be the leading dataset in combining these datasets. The new dataset therefore should be comprised of 30,000 records.

  Note: When combining datasets by a key field, the key field name must be identical (in name and case) in order to have the field appear in the **Possible keys** area. Use a **Filter** node where needed to ensure that the name of the key field is the same in all datasets.

- Enrich the dataset that you created in the previous task with zipcode (postal code) information stored in the file **ACME zip data.csv** (already on the stream canvas).

- Export a **25%** random sample of the data to a comma-separated text file (use **123** for the seed value); name the file **ACME sample.dat**.

  To check your results: 7,568 records will be sampled.

- Exit IBM SPSS Modeler without saving anything.

For more information about where to work and the exercise results, refer to the Tasks and Results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and Results

Task 1. Open the stream.

- From the **File** menu, click **Open Stream**, and navigate to **C:\Training\0A008\07-Integrating_data\Start**, click **unit_07_exercise_1_start.str**, and then click **Open**.

Task 2. Create a single dataset for purchases.

- **ACME purchases 1999 - 2004.dat** and **ACME purchases 2005 - 2010.dat** have the same fields, but different records. Therefore, use the **Append** node (in the Record Ops palette) to join the two datasets. No edits are needed in the Append node.
- Run a **Table** node downstream from the **Append** node to show that there are 71,127 records in the new dataset.

Task 3. Create a single dataset for order lines.

- Use the **Append** node to join the **ACME orderlines 1999 - 2004.sav** and **ACME orderlines 2005 - 2010.sav** datasets. No edits are needed in the Append node.
- Run a **Table** node downstream from the **Append** node to show that the new dataset is comprised of 169,734 records.

Task 4. Create a dataset with one record per purchase for order lines.

- From the **Records Ops** palette, add an **Aggregate** node downstream from the second **Append** node.
- Edit the **Aggregate** node, and then:
  - for **Key fields**, select **PURCHASE_ID**
  - for **Aggregate fields**, select **PRICE**, and statistic **Sum**
  - disable the **Include record count in field** option
- Run a **Table** node downstream from the **Aggregate** node to verify that the aggregated dataset has 71,127 records, and PURCHASE ID 5723 has a PRICE_Sum of 948.48.

Task 5. Create a single dataset from the purchases dataset and the (aggregated) order lines dataset.

- Add a **Merge** node to the stream canvas, and connect both the first **Append** node (originating from the appended datasets storing the purchases) and the **Aggregate** node (originating from the appended datasets storing order lines) to it.
- Edit the **Merge** node, and then:
  - for **Merge Method**, select **Keys**
  - for **Keys for merge**, select **PURCHASE_ID**
  - use the (default) **Inner join** method for merging (both datasets include the same PURCHASE IDs, so an inner, outer, or partial join will give the same results)

Task 6. Create a dataset with one record per customer from the purchases and order lines dataset.

- Add an **Aggregate** node downstream from the **Merge** node.
- Edit the **Aggregate** node.
  - for **Keys fields**, select **CUSTOMER_ID**
  - for **Aggregate fields**, select:
    - **ORDERDATE** and statistic **Max**
    - **PRICE_Sum** and statistic **Sum**
  - ensure that the option **Include record count in field** is enabled and name it **NUMBER OF PURCHASES**
- Run a **Table** node downstream from the **Aggregate** node to verify that you have 30,000 records.

Task 7. Create a single dataset for customer information, their purchases and their mailing history.

- So that the key field is the same in all three datasets, you must rename the **customer_id** field in **ACME mailing history** by adding a **Filter** node (from the Field Ops palette) downstream from the **SetToFlag** node, and then replace **customer_id** by **CUSTOMER_ID**.
- Add a **Merge** node (**Record Ops** palette) to the stream canvas, and connect the **Aggregate** node (from the previous task), the **Distinct** node and the **Filter** node to it.
- Edit the **Merge** node, and then:
  - for **Merge method**, select **Keys**, for **Keys for merge**, select **CUSTOMER_ID** (after having renamed customer_id to CUSTOMER_ID in the ACME mailing history dataset; this field is available as key field)
  - select the **Partial outer join** merge method, click **Select**, and ensure that **ACME customer data.xlsx** is selected as the leading dataset in the merge
- Run a **Table** node downstream from the **Merge** node to verify your results.

Task 8. Enrich the data with zipcode information.

- Add a **Merge** node (**Record Ops** palette) to the stream, and then connect the **Merge** node (from the previous task) and the **Var. File** node named **ACME zip data.csv** to it.
- Edit the **Merge** node, and then:
  - for **Merge Method**, select **Keys**
  - for **Keys for merge**, select **ZIP**
  - select the **Partial outer join** merge method, click **Select**, and ensure that the dataset originating from the Merge node in the previous task is the leading dataset in the merge
- Run a **Table** node downstream from the **Merge** node to verify your results.

Task 9. Export a random sample of 25%.

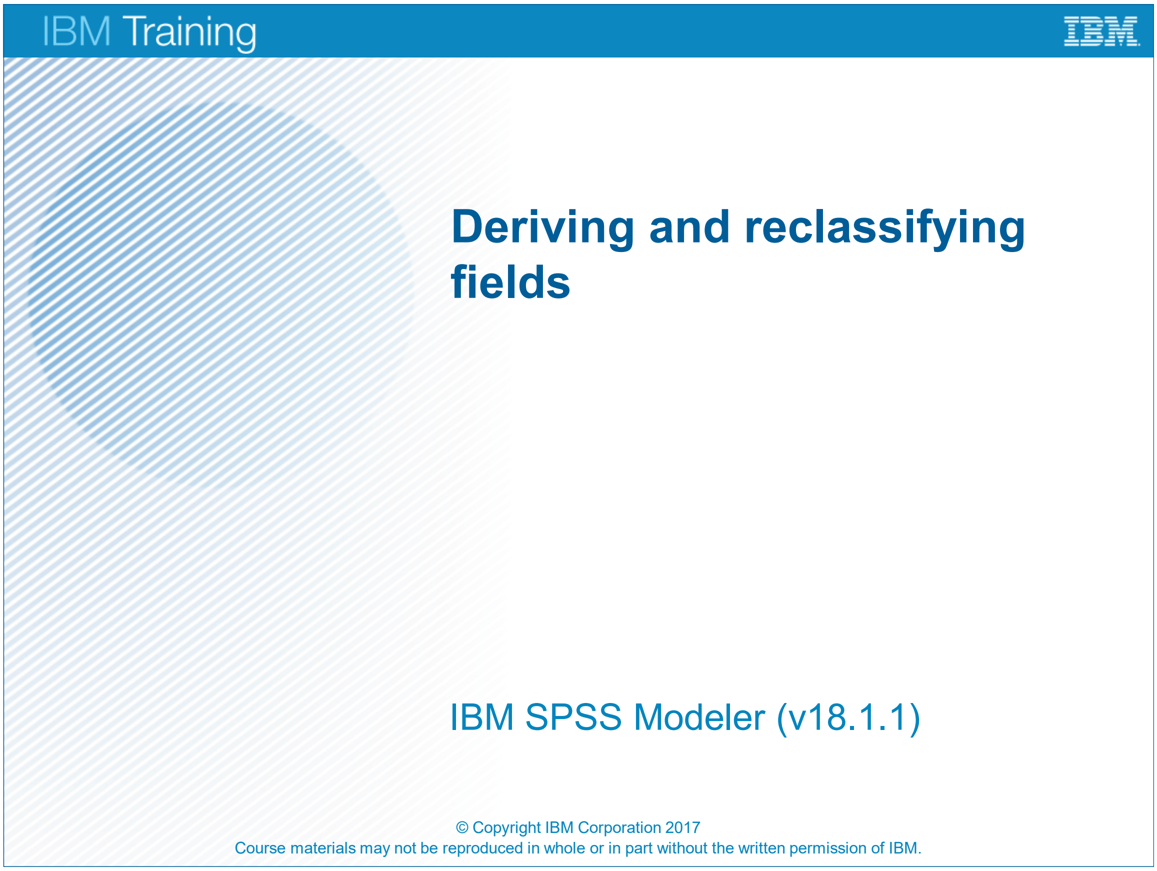
- Add a **Sample** node downstream from the **Merge** node (from the previous task).
- Edit the **Sample** node, and then:
  - for **Sample**, select **Random %**, and set the value to **25**
  - enable the option **Repeatable partition assignment**, and set the **Seed** to **123**
- Run a **Table** node downstream from the **Sample** node to verify the results (7,568 records should be sampled).
- Add a **Flat File** node (**Export** palette) downstream from the **Sample** node.
- Edit the **Flat File** node, and then, beside **Export file**, type **ACME sample.dat**.
- Click **Run** to initiate the export.

Task 10. Exit IBM SPSS Modeler.

- From the **File** menu, click **Exit**, and then exit IBM SPSS Modeler without saving anything.

You will find the solution results in the **C:\Training\0A008\07-Integrating_data\Solutions** folder.

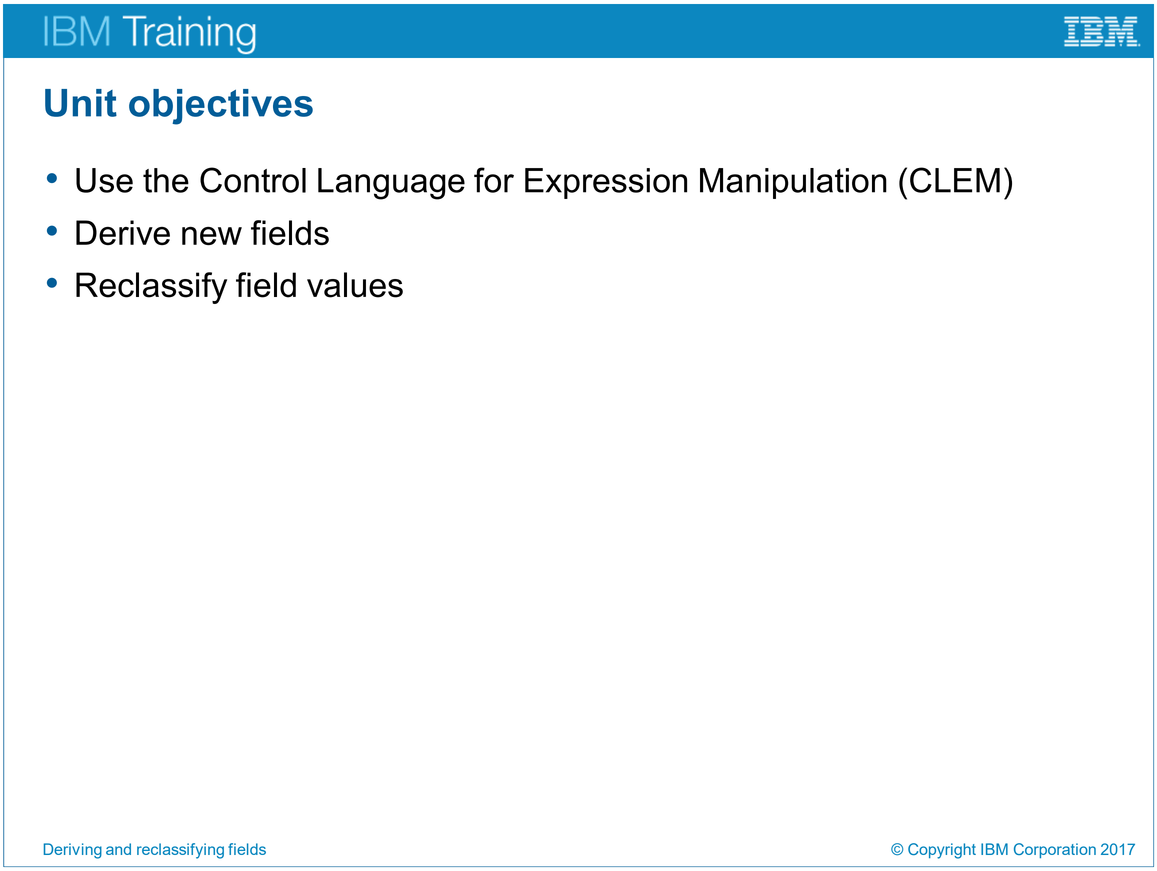
Unit 8 Deriving and reclassifying fields



IBM Training

Deriving and reclassifying fields

IBM SPSS Modeler (v18.1.1)



IBM Training

Unit objectives

- Use the Control Language for Expression Manipulation (CLEM)
- Derive new fields
- Reclassify field values

Deriving and reclassifying fields

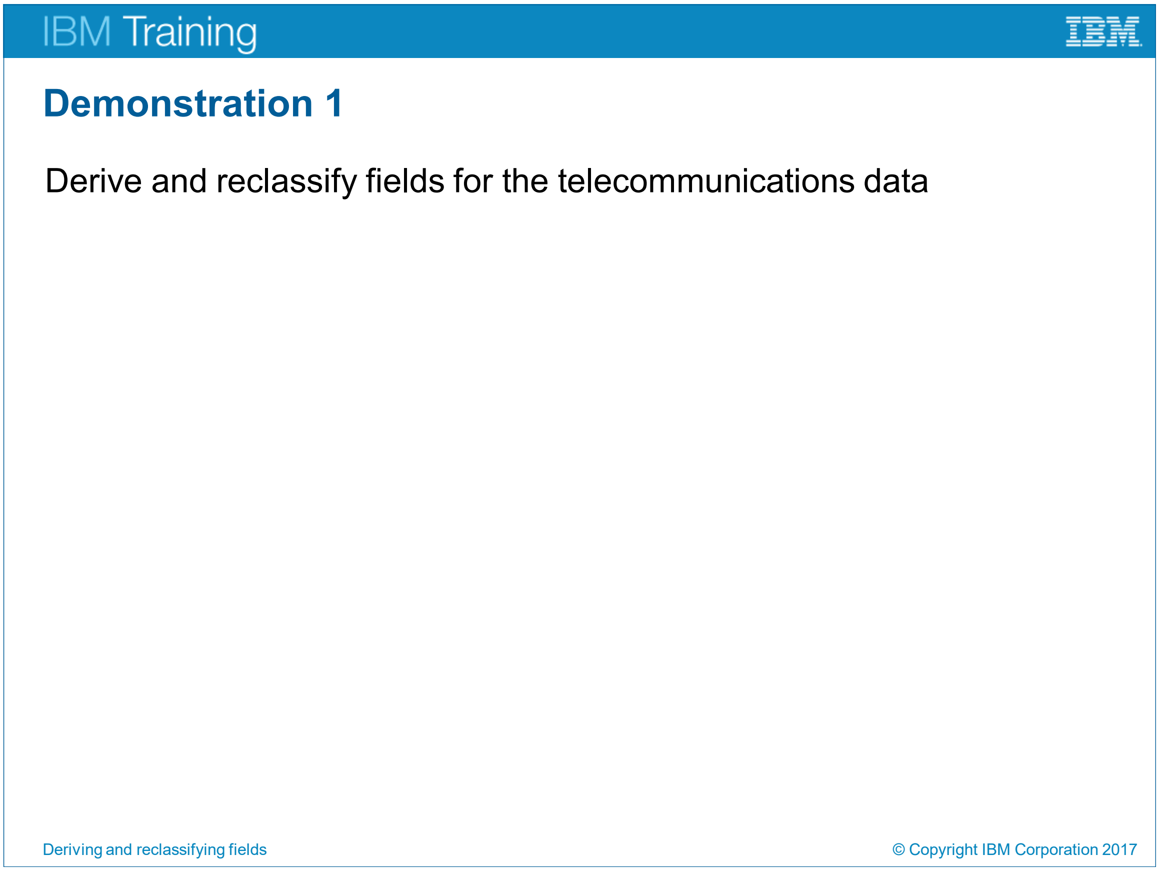
© Copyright IBM Corporation 2017

Unit objectives

The focus in this unit is on another task in the data preparation stage: construct the final dataset for modeling by cleansing and enriching the data.

Before reviewing this unit you should be familiar with:

- CRISP-DM
- IBM SPSS Modeler streams, nodes and palettes
- Methods to collect initial data
- Measurement levels and storages
- Methods to explore the data



IBM Training

Demonstration 1

Derive and reclassify fields for the telecommunications data

Deriving and reclassifying fields

Demonstration 1: Derive and reclassify fields for the telecommunications data

Demonstration 1:
Derive and reclassify fields for the telecommunications data

Purpose:
You work as a data scientist for a telecommunications firm and you need to cleanse and enrich a dataset in order to build models later.

Data file:	**telco x data.txt**
Data folder:	**C:\Training\0A008**
Stream file:	**unit_08_demonstration_1_start.str**
Stream file folder:	**C:\Training\0A008\08-Deriving_and_reclassifying_fields\Start**

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

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.
2. When a welcome window appears, click **Cancel**.

   If you have already configured IBM SPSS Modeler in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A008** folder, and then click **Set**.

Task 2. Open the start stream.

1. From the **File** menu, click **Open Stream**, navigate to the **C:\Training\0A008\08-Deriving_and_reclassifying_fields\Start** folder, click **unit_08_demonstration_1_start.str**, and then click **Open**.

Task 3. Explore the Derive dialog box.

1. From the **Field Ops** palette, add a **Derive** node downstream from the **Type** node.
2. Edit the **Derive** node.

   By default, a single field will be derived. Specify the name of this new field in the Derive field text box.

   Alternatively, you can derive multiple fields with a single Derive node. This applies when multiple fields must be derived using the same formula. For example, when you want to derive 5 new fields by multiplying 5 source fields by 100, use the multiple mode rather than 5 separate Derive nodes.

3. Beside **Derive as**, click the drop down.

   You can derive a field as:

   - Formula: An outcome of a formula. For example: a new field TAX derived as: TAX = INCOME * 0.20.
   - Flag: A T/F field. For example: a new field ADULT, True if AGE >= 21, else False.
   - Nominal: A categorical field. For example: a new field AGECAT, 1 if AGE is less than or equal to 35, 2 if AGE is greater than 35 and less than or equal to 70, and 3 if AGE is greater than 70.
   - Conditional: An outcome of a formula, but computed conditionally. For example: a new field TAX = 0.1 * INCOME if INCOME <= 100000, and TAX= 10000 + 0.2 * (INCOME – 100000) if INCOME > 100000.

   Refer to the Help or the *Advanced Data Preparation Using IBM SPSS Modeler* course for more information about the other two options (Count and State).

   The selection made for the Derive as option will be reflected in the dialog box. For example, when you derive a new field as Formula, you can type the formula in the Formula text box.

4. Beside **File type**, click the drop down.

   If you keep the default, IBM SPSS Modeler will auto-type the new field. For example, if you derive a field TAX as Formula, equal to INCOME * 0.20, TAX will be auto-typed as a continuous field. Or, when you derive a new field as Flag, it will be auto-typed as Flag. In general, you can leave the default value and let IBM SPSS Modeler determine the measurement level. As one of the few examples where you would set the measurement level manually, think of deriving a field such as AGE CATEGORY with values 1, 2 and 3. This field will be derived as nominal and autotyped as nominal, whereas its measurement level should be ordinal.

   You can type your CLEM expression, but that is not efficient, especially because field names and function names are case-sensitive. Instead of, or in conjunction with typing, you can use the Expression Builder to create expressions.

   You can invoke the Expression Builder by clicking Launch expression builder 🖩. (Note: The Expression Builder is also available in the Select dialog box).

5. Click **Launch expression builder** .

   You can build the expression by selecting and pasting the various elements (fields, functions, and operators) to the area where the expression must be specified.

   Functions are grouped by categories, such as string functions, date and time functions, numeric functions, and logical functions. When you select a function you will have a description of its use at the bottom of the dialog box.

   When you need to specify a value of a categorical field, you can pick the value from a list of values, provided that the field is instantiated. If the field is not instantiated, its values will not be available.

6. Close the **Expression Builder** window.

Task 4. Derive bill for peak, offpeak and total bill.

You will derive fields that store the bill for calling in the peak hours and the bill for calling in the off-peak hours (these fields can be valuable predictors for churn). Each field is computed by multiplying the minutes by the corresponding rate. Rates are expressed in dollar and cents and therefore you will divide by 100.

When you derive and reclassify fields, it is recommended to add the nodes to the same branch of the stream. This lets you create fields from earlier created fields. Also, adding new fields downstream in the same branch is necessary to use all of these new fields together in modeling.

1. Make the following edits in the **Derive** dialog box:
   - Derive field: **BILL_PEAK**
   - Derive as: **Formula** (the default)
   - Expression Builder: **PEAK_MINS * PEAK_RATE/100**
2. Close the **Expression Builder** window.

   You could preview the data to check the results for a few records.
3. Close the **Derive** dialog box.
4. From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **BILL_PEAK**, and edit it as follows:
   - Derive field: **BILL_OFFPEAK**
   - Derive as: **Formula** (the default)
   - Expression: **OFFPEAK_MINS * OFFPEAK_RATE / 100**

   Now that you have derived these two fields you will sum their values to get the total bill.

5. From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **BILL_OFFPEAK**, and edit it as follows:
   - Derive field: **BILL_TOTAL**
   - Derive as: **Formula** (the default)
   - Expression: **BILL_PEAK + BILL_OFFPEAK**
6. Click **Preview**, and then scroll to the last fields.

   The new fields are computed correctly.
7. Close the **Preview** output window.
8. Close the **Derive** dialog box.

Task 5. Derive more fields for modeling.

Next, you will derive a field that flags whether the total bill is greater than 0.

1. From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **BILL_TOTAL**, and edit it as follows:
   - Derive field: **BILL_TOTAL_GREATER_THAN_0**
   - Derive as: **Flag**
   - Field type: **Flag** (default value when the Derive type is Flag)
   - True when: **BILL_TOTAL > 0**
2. Close the **Derive** dialog box.

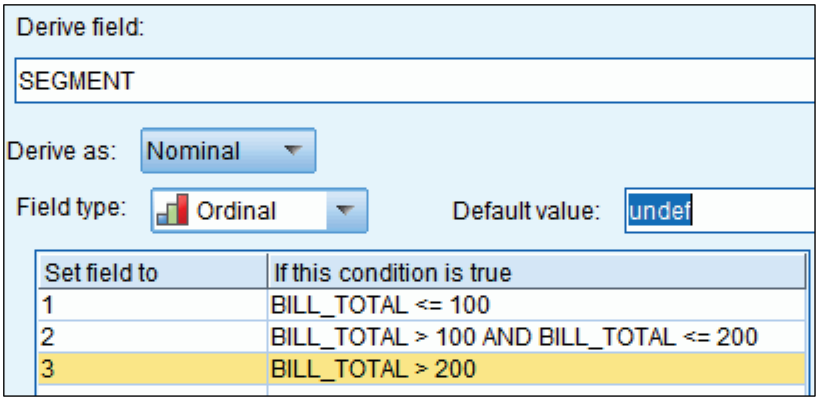
   Based on the total bill, you will derive a field named SEGMENT. This field classifies a customer into one of three categories: 1 (BILL_TOTAL <= 100), 2 (BILL_TOTAL > 100 and BILL_TOTAL <= 200), 3 (BILL_TOTAL > 200). Because this is an ordinal field you will set the field's measurement to Ordinal.
3. From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **BILL_TOTAL_GREATER_THAN_0**, and edit it as follows:
   - Derive field: **SEGMENT**
   - Derive as: **Nominal**
   - Field type: **Ordinal**
4. Under **Set field to**, enter **1**, and under **If this condition is true**, enter **BILL_TOTAL <= 100**. (Use the Expression Builder, if preferred.)

5. Repeat the previous step for the following values and expressions:
   2 BILL_TOTAL > 100 and BILL_TOTAL <= 200
   3 BILL_TOTAL > 200

   When BILL_TOTAL is undefined ($null$), the result for segment should be $null$ also. In CLEM you refer to the $null$ value by using undef.

6. Beside **Default value**, type **undef**.

   The results appear as follows:



Derive field:

SEGMENT

Derive as: Nominal

Field type: Ordinal   Default value: undef

Set field to	If this condition is true
1	BILL_TOTAL <= 100
2	BILL_TOTAL > 100 AND BILL_TOTAL <= 200
3	BILL_TOTAL > 200

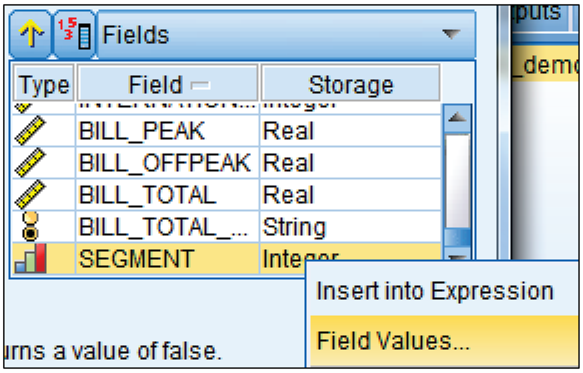
7. Close the **Derive** dialog box.

   Finally, suppose that customers in segment 3 receive a 10% discount on their total bill, whereas customers in segments 1 and 2 do not receive a discount. You will derive a field named DISCOUNT conditionally.

8. From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **SEGMENT**.
9. Edit the **Derive** node.
10. Under **Derive** field, type **DISCOUNT**.
11. Beside **Derive as** field, select **Conditional**.

    To demonstrate how to pick from a field's values, you will use the Expression Builder.

12. Click **Launch expression builder** which is next to the **If** text box.
13. Type **SEGMENT =** .
14. In the field list, right-click SEGMENT, as shown below.



Type	Field	Storage
	BILL_PEAK	Real
	BILL_OFFPEAK	Real
	BILL_TOTAL	Real
	BILL_TOTAL_...	String
	SEGMENT	Integer

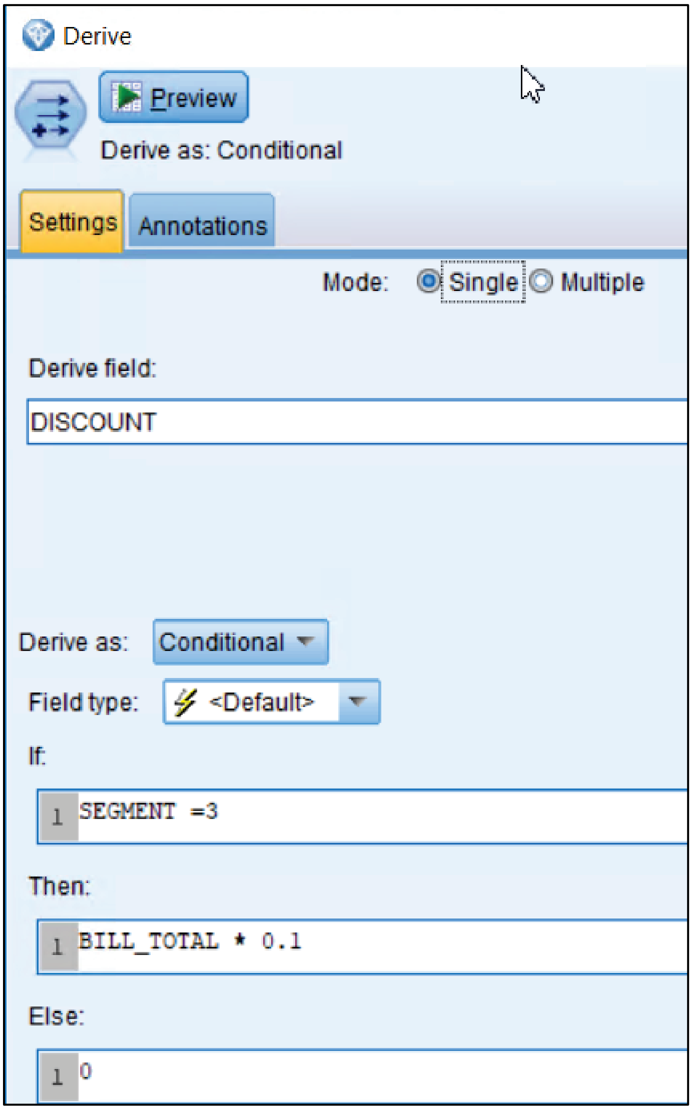
irns a value of false.

15. Click **Field Values**, click **3**, and then click **Insert**.

    The expression now reads SEGMENT =3.

16. Close the Expression Builder.
17. In the **Then** text box, type Then: **BILL_TOTAL * 0.1**.
18. In the **Else** text box, type **0**.

    The results appear as follows:



Derive as: Conditional

Mode: ● Single ○ Multiple

Derive field: DISCOUNT

Derive as: Conditional

Field type: <Default>

If:

```
SEGMENT =3
```

Then:

```
BILL_TOTAL * 0.1
```

Else:

```
0
```

19. Close the **Derive** dialog box
20. From the **Output** palette, add a **Table** node downstream from the **Derive** node named **DISCOUNT**, and then run the **Table** node.

    Customers in segment 3 have a discount equal to 10% of their total bill. Discount equals 0 for customers from segments 2 and 3.

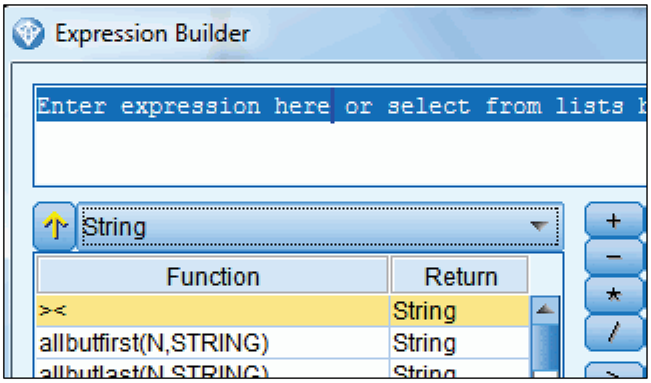
21. Close the **Table** output window.

Task 6. Cleanse data for GENDER.

1. Run the **Distribution** node for **GENDER** downstream from the **Type** node.

   The values of gender are not spelled consistently. To fix this, a new field named GENDER_OK will be derived, with values MALE and FEMALE. You can use the Reclassify node for this purpose, but it is more efficient to use the Derive node with an appropriate string function that converts lower case characters into upper case characters.
2. Close the **Distribution** output window.
3. From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **DISCOUNT**.
4. Edit the **Derive** node.
5. In the **Derive field** text box, type **GENDER_OK**.
6. Click **Launch expression builder** to launch the **Expression Builder**.
7. On the left side, click **General Functions**, and select the **String** group of functions.

   The results appear similar to the following:



Expression Builder

```
Enter expression here or select from lists b
```

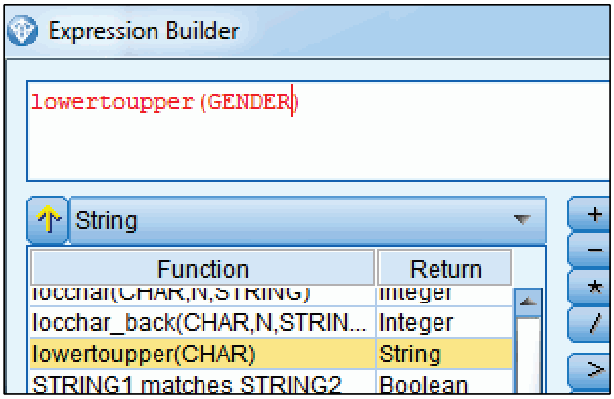
String

Function	Return
><	String
allbutfirst(N,STRING)	String
allbutlast(N,STRING)	String

8. Under **Function**, select **lowertoupper (CHAR)**, and then click **Insert** to paste it into the expression area.

9. In the right pane, under **Field**, double-click **GENDER**, to add it to the expression area. (If preferred, type the expression.)

   The results appear as follows:



```
lowertoupper(GENDER)
```

Function	Return
locchar(CHAR,N,STRING)	Integer
locchar_back(CHAR,N,STRIN...	Integer
lowertoupper(CHAR)	String
STRING1 matches STRING2	Boolean

10. Close the **Expression Builder** window.
11. Click **Preview**.
12. Close the **Preview** output window.
13. Close the **Derive** dialog box.
14. From the **Graphs** palette, add a **Distribution** node downstream from the **Derive** node named **GENDER_OK**.
15. Edit the **Distribution** node, and then beside **Field** select **GENDER_OK**.
16. Click **Run**.

    The values of the new field are okay.
17. Close the **Distribution** output window.

Task 7. Cleanse data and reclassify fields for modeling.

In the previous task, this fixed the values for GENDER by using the Derive node and adding a new field. In this task the Reclassify node will be used which has the advantage that the field's values can be overwritten.

You will also recode the handsets into a (smaller) number of brands.

1. From the **Field Ops** palette, add a **Reclassify** node downstream from the **Derive** node named **GENDER_OK**.
2. Edit the **Reclassify** node.

   Beside Mode, select Single when you want to reclassify only one field. Multiple mode is useful when the same reclassification rules apply to a number of fields. For example, when you want to recode 15 satisfaction fields that are measured on a 7-point scale into a 3-point scale, select the Multiple option rather than using 15 separate Reclassify nodes.The new values will be placed in a new field by default. Alternatively, you can choose to replace the field's values with the Existing field option.

   In this example you will overwrite the values.

3. Beside **Reclassify into**, select **Existing field**.
4. Under **Reclassify field**, select **GENDER**.

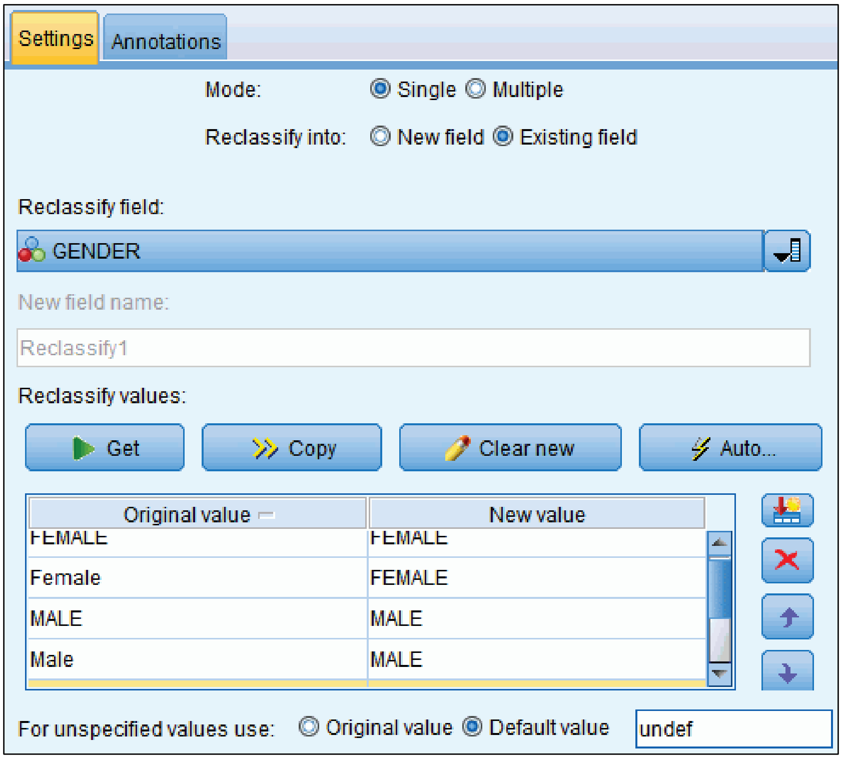
   Under Reclassify values, click the Get button to populate the Original value column with values (provided that the field is instantiated).

   You can click the Copy button to copy the values from the Original value column to the New value column. This is useful if you want to retain most of the original values, reclassifying only a few.
5. Click the **Get** button, to populate the **Original value** column.
6. Click the **Copy** button, to populate the **New value** column.

   You will recode the values to upper case.
7. Ensure that the **New value** column values are in upper case.

   When a value is encountered in the source field that is not listed in the Original value column, either the value itself or the default value undef ($null$) can be assigned. In this example, you will assign the undef value.
8. Beside **For unspecified values use**, enable the **Default value** option.

   The results appear similar to the following:



Mode: ● Single ○ Multiple

Reclassify into: ○ New field ● Existing field

Reclassify field: GENDER

New field name: Reclassify1

Reclassify values:

Original value	New value
FEMALE	FEMALE
Female	FEMALE
MALE	MALE
Male	MALE

For unspecified values use: ○ Original value ● Default value undef

9. Click the **Preview** button.

   The values of GENDER are overwritten and in upper case.
10. Close the **Preview** output window.

11. Close the **Reclassify** dialog box.

    You recoded the different spellings for GENDER into upper case, overwriting an existing field. The Derive node did not let you do this, because Derive will always add a new field.

    Overwriting the values of an existing field can also be accomplished with the Filler node. Refer to the *Advanced Data Preparation using IBM SPSS Modeler* course for more information.

    Next, you will recode the values of handset. The handsets have a brand name and a type number, for example CAS30. For modeling, only the brand name is relevant, CAS in this example.

12. From the **Field Ops** palette, add a **Reclassify** node downstream from the first **Reclassify** node, and edit it as follows:

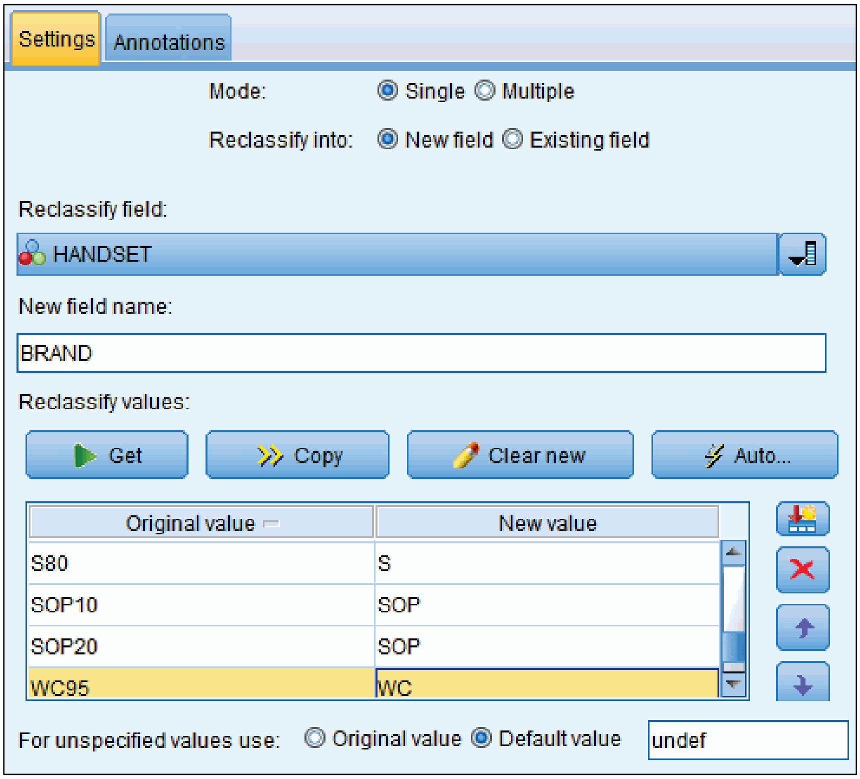
    - Reclassify field: **HANDSET**
    - New field name: **BRAND**

13. Click the **Get** button to populate the **Original values** column.
14. For the first new brand name, type **ASAD** in the **New value** column.
15. For the second new brand name, type **ASAD**, or pick **ASAD** from the drop-down list.
16. Repeat for the other brands, using only the letters from the **Original value** as the **New value**.

17. Beside **For unspecified values use**, enable the **Default value** option.

    If this Reclassify node would be applied to another dataset, with possibly different handsets, these handsets will then be recoded into the $null$ value.

    The results appear similar to the following:



Mode: ● Single ○ Multiple

Reclassify into: ● New field ○ Existing field

Reclassify field: HANDSET

New field name: BRAND

Reclassify values:

Original value	New value
S80	S
SOP10	SOP
SOP20	SOP
WC95	WC

For unspecified values use: ○ Original value ● Default value undef

18. Close the **Reclassify** dialog box.

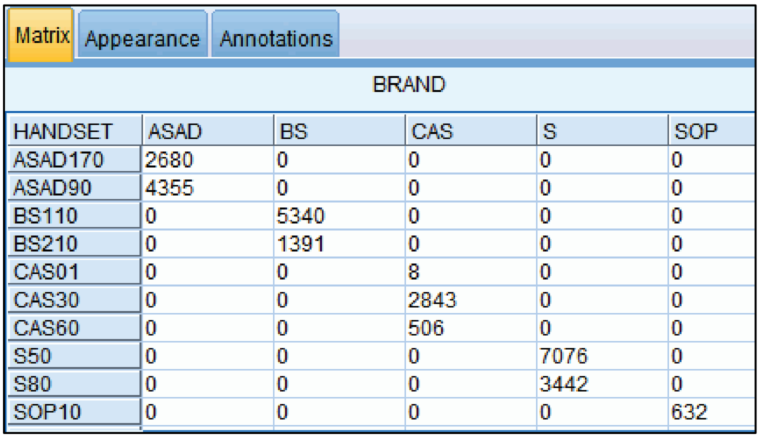
    To check the recoding for handsets into brands, you will cross tabulate the two fields. A Matrix node can be used for this purpose.

19. From the **Output** palette, add a **Matrix** node downstream from the **Reclassify** node named **BRAND**.

20. Edit the **Matrix** node, beside **Rows**, select **HANDSET**, and then beside **Columns** select **BRAND**.

21. Click **Run**.

    The results appear similar to the following:



	BRAND				
HANDSET	ASAD	BS	CAS	S	SOP
ASAD170	2680	0	0	0	0
ASAD90	4355	0	0	0	0
BS110	0	5340	0	0	0
BS210	0	1391	0	0	0
CAS01	0	0	8	0	0
CAS30	0	0	2843	0	0
CAS60	0	0	506	0	0
S50	0	0	0	7076	0
S80	0	0	0	3442	0
SOP10	0	0	0	0	632

    The handsets are reclassified into brands correctly.

22. Close the **Matrix** output window.

    This completes the demonstration for this unit. You will create a clean state for the exercise.

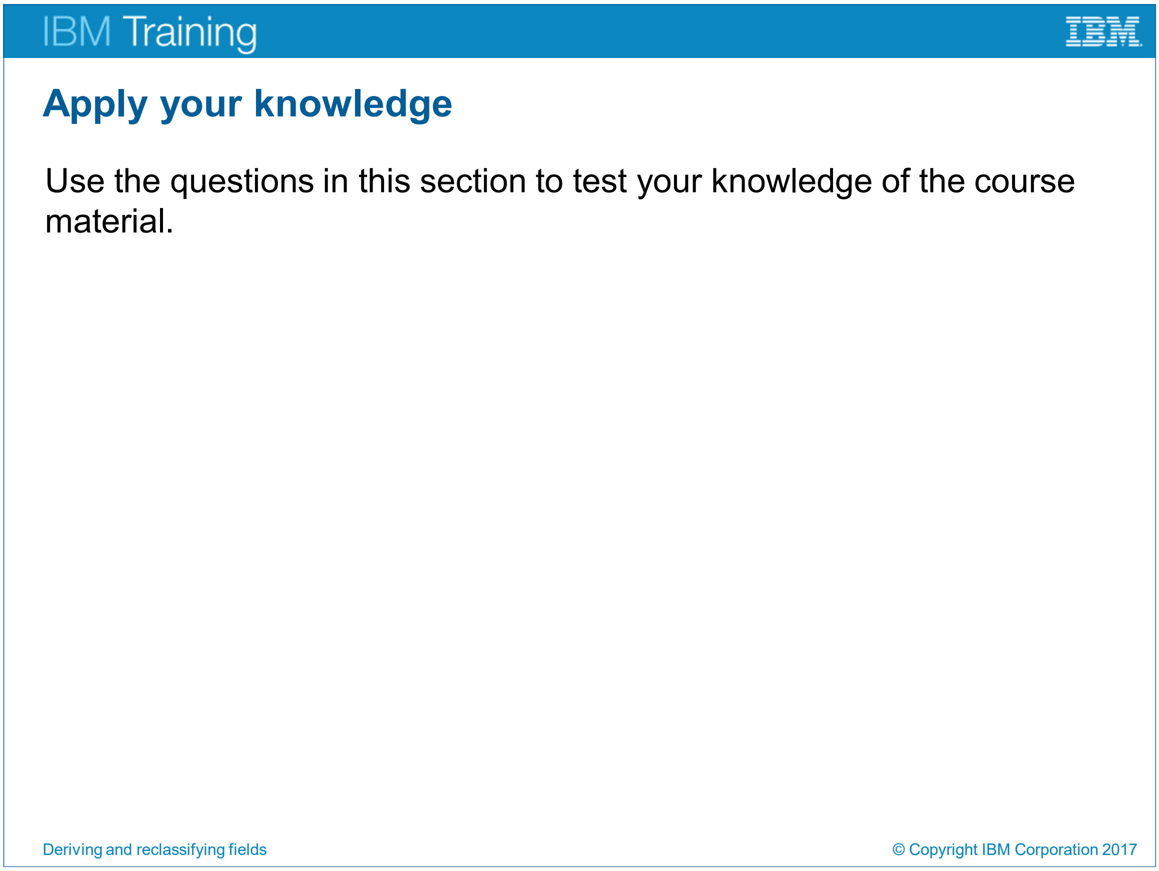
23. From the **File** menu, click **Close Stream**, and then click **No** when asked to save the stream.

24. From the **File** menu, click **New Stream**.

    Leave IBM SPSS Modeler open for the exercise.

Results:
You have cleansed and enriched the data, to build better models later.

You will find the completed stream in the **C:\Training\0A008\08-Deriving_and_reclassifying_fields\Solutions** folder.



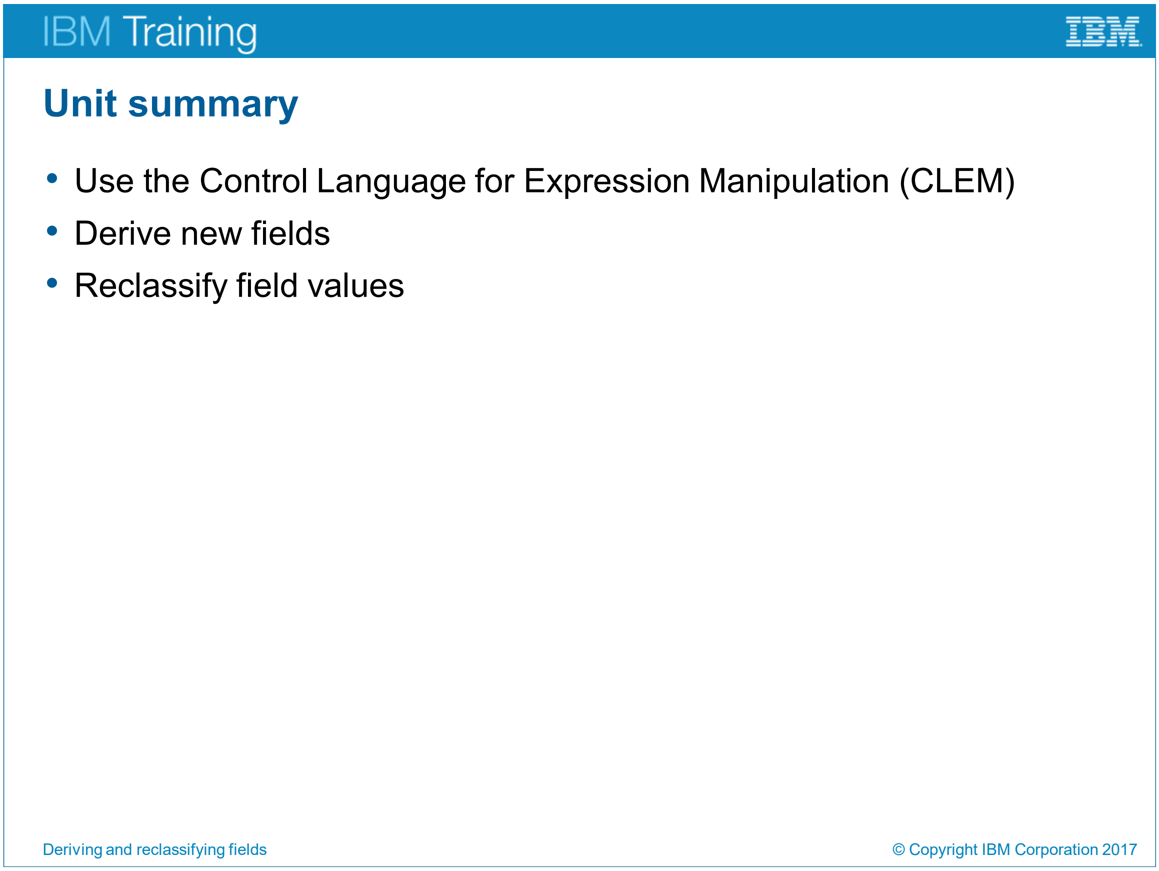
IBM Training

Apply your knowledge

Use the questions in this section to test your knowledge of the course material.

Deriving and reclassifying fields

Apply your knowledge



IBM Training

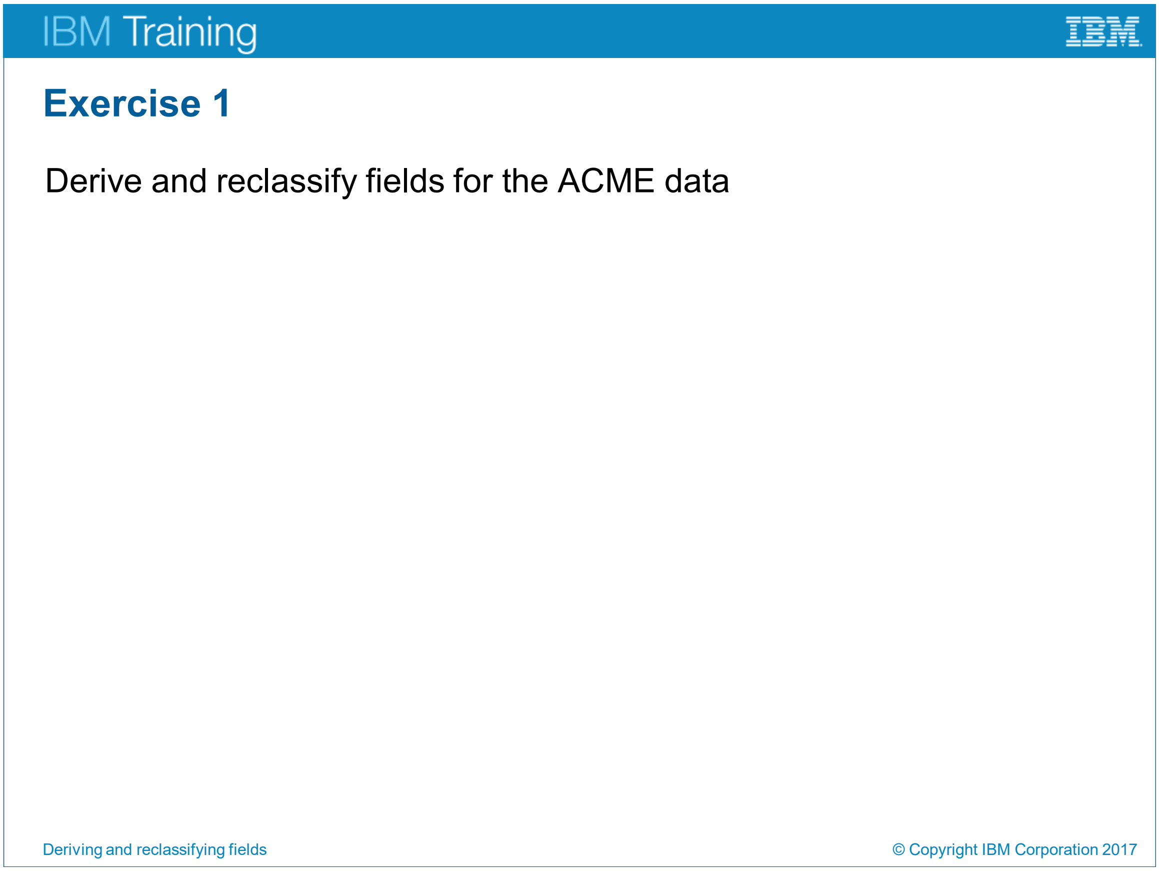
Unit summary

- Use the Control Language for Expression Manipulation (CLEM)
- Derive new fields
- Reclassify field values

Deriving and reclassifying fields

© Copyright IBM Corporation 2017

Unit summary



IBM Training

Exercise 1

Derive and reclassify fields for the ACME data

Deriving and reclassifying fields

Exercise 1: Derive and reclassify fields for the ACME data

Exercise 1: Derive and reclassify fields for the ACME data

Data file:	**ACME data part 1.dat**
Data folder:	**C:\Training\0A008**
Stream file:	**unit_08_exercise_1_start.str**
Stream file folder:	**C:\Training\0A008\08-Deriving_and_reclassifying_fields\Start**

You work at ACME where you are responsible for data-preparation. You have to cleanse and enrich the data, so that better models can be built later.

- Open **unit_08_exercise_1_start.str**, located in the **C:\Training\0A008\08-Deriving_and_reclassifying_fields\Start** folder.
- Create a field named **DIFFERENCE** that stores the difference between **CREDITLIMIT** and **AMOUNT_SPENT**. The difference should return a negative number when AMOUNT_SPENT exceeds CREDITLIMIT.

  To check your results: The customer with ID 723.000 (the first record in the dataset) has a credit limit of 9,026.000, her amount spent is 546.731, so the difference is 8,479.269.
- **CREDITLIMIT**, **AMOUNT_SPENT**, and **DIFFERENCE** are all in Euro (a European currency). Based on these fields, create three new fields that are expressed in US dollars: **CREDITLIMIT_Dollar**, **AMOUNT_SPENT_Dollar**, and **DIFFERENCE_Dollar** (1 euro equals, say, 1.1 US dollar).

  Hint: Because the formula is the same for the three fields (each input field must be multiplied by 1.1), you can use the **Multiple** mode in the **Derive** node. The new field names are the original ones, with **_Dollar** as suffix. In the **Formula** area, refer to the input fields by using **@FIELD**.

  To check your results: The customer with ID 723.000 (the first record in the dataset) has a credit limit of 9,928.6 dollars, spent 601.404 dollars, and the difference equals 9,327.196 dollars.
- Create a field that flags whether the amount spent exceeds the credit limit. Name the field **SPENT_TOO_MUCH**.

  To check your results: 2,987 customers exceeded their credit limit. (Run a Distribution node on the new field.)

- Based on amount spent (in dollars), create a field named SEGMENT with the following three categories:
  - 1-Bronze: AMOUNT_SPENT_Dollar up to 2000 dollars (including 2000)
  - 2-Silver: AMOUNT_SPENT_Dollar between 2000 dollars and 5000 dollars (excluding 2000, including 5000).
  - 3-Gold: AMOUNT_SPENT_Dollar greater than 5000 dollars

  To check your results: There are 28,218 bronze customers, 1,709 silver customers, and 73 gold customers. (Run a Distribution node on the new field.)
- Gold customers receive a bonus of 5% of their amount spent (in dollars), and bronze and silver customers get no bonus at all. Create this field; name it **BONUS**.

  To check your results: The customer with ID 723.000 (the first record in the dataset) has zero bonus.
- **GENDER** has different spellings. Cleanse the data so that its values are Female and Male.

  To check your results: Run a Distribution graph for GENDER; you should only have the values Female and Male.
- In the dataset, there is a field named **ZODIAC** (astrological sign), whose values range from 1.0 to 12.0. Recode this field into a new field named **SEASON_BORN**, with 4 categories defined as follows:
  - Winter: ZODIAC equals 12.0, 1.0 or 2.0
  - Spring: ZODIAC equals 3.0, 4.0, or 5.0
  - Summer: ZODIAC equals 6.0, 7.0 or 8.0
  - Autumn: ZODIAC equals 9.0, 10.0 or 11.0

  To check your results: Run a **Matrix** node (Output palette), with **ZODIAC** cross tabulated in the row by **SEASON_BORN** in the column.
- Optional: create a field named **ORDERDATE_YEAR**, which returns the year of the order date.
  Hint: Use an appropriate Date and Time function.

  To check your results: The customer with ID 723.000 (the first record in the dataset) ordered in 2009.
- Exit IBM SPSS Modeler without saving anything.

For more information about where to work and the exercise results, refer to the Tasks and Results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1:
Tasks and Results

Task 1. Open the start stream.

- From the **File** menu, click **Open Stream**, navigate to **C:\Training\0A008\08-Deriving_and_reclassifying_fields\Start** and then open **unit_08_exercise_1_start.str**.

Task 2. Compute the difference between amount spent and credit limit.

- Add a **Derive** node (**Field Ops** palette) downstream from the **Type** node, and edit it as follows:
  - Derive field: **DIFFERENCE**
  - Derive type: **Formula** (default)
  - Formula: **CREDITLIMIT - AMOUNT_SPENT**
- Click **Preview**, and scroll to the far right.

  The difference for the first record equals 8,479.269.

Task 3. Compute fields in dollars from fields in euros.

Notice that the same formula applies to three fields: all fields must be multiplied by 1.1.

- Add a **Derive** node (**Field Ops** palette) downstream from the **Derive** node named **DIFFERENCE**, and define it as follows:
  - Mode: **Multiple**.
  - Derive from: **CREDITLIMIT, AMOUNT_SPENT**, and **DIFFERENCE**
  - Field name extension: **_Dollar**
  - Formula: **@FIELD * 1.1**

    @FIELD refers to each of the input fields in turn.
- Click **Preview**.

  For Customer ID 723.000 (the first record in the dataset), the CREDITLIMIT_Dollar value is 9,928.600, the AMOUNT_SPENT_Dollar is 601.404, and the difference between them (DIFFERENCE_Dollar) is 9,327.196.

Task 4. Flag whether amount spent exceeds the credit limit.

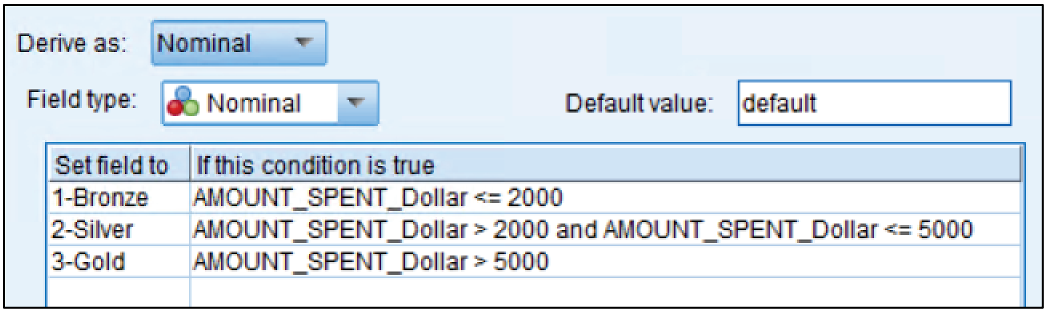
- Add a **Derive** node (**Field Ops** palette) downstream from the **Derive** node named **_Dollar**, and edit it as follows:
    - Derive field: **SPENT_TOO_MUCH**
    - Derive as: **Flag**
    - True when: **AMOUNT_SPENT_Dollar > CREDITLIMIT_Dollar**

      Alternatively, use the DIFFERENCE field created in task 2
- Add a **Distribution** node (**Graphs** palette) downstream from the **Derive** node named **SPENT_TOO_MUCH**, set the **Field** to **SPENT_TOO_MUCH**, and then click **Run**.

  The results show that 2,987 customers exceeded their credit limit.

Task 5. Create a segment field.

- Add a **Derive** node downstream from the **Derive** node named **SPENT_TOO_MUCH**, and edit as follows:
    - Derive field: **SEGMENT**
    - Derive as: **Nominal**
    - Specify the values and conditions as follows:



Derive as: Nominal

Field type: Nominal

Default value: default

Set field to	If this condition is true
1-Bronze	AMOUNT_SPENT_Dollar <= 2000
2-Silver	AMOUNT_SPENT_Dollar > 2000 and AMOUNT_SPENT_Dollar <= 5000
3-Gold	AMOUNT_SPENT_Dollar > 5000

- Add a **Distribution** node (**Graphs** palette) downstream from **SEGMENT**, set the **Field** to **SEGMENT**, and then click **Run**.

  The results show that there are 28,218 bronze customers, 1,709 silver customers, and 73 gold customers.

Task 6. Create a field returning the bonus.

- Add a **Derive** node (**Field Ops** palette) downstream from the **Derive** node named **SEGMENT**, and edit it as follows:
    - Derive field: **BONUS**
    - Derive as: **Conditional**
    - If: **SEGMENT = "3-Gold"**
    - Then: **AMOUNT_SPENT_Dollar * 0.05**
    - Else: **0**
- Click **Preview**.

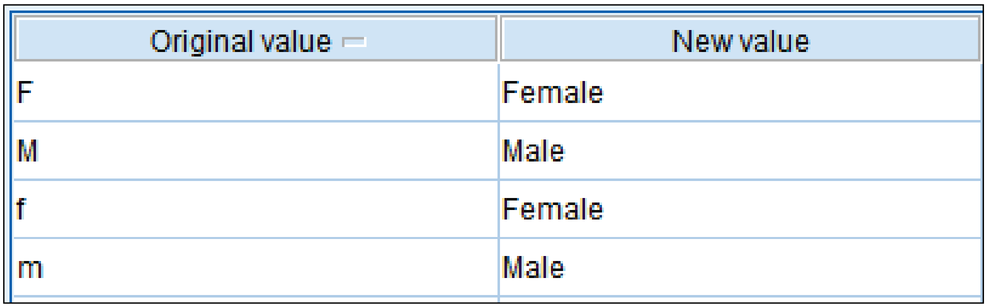
  The results show that CUSTOMER_ID 723.000 (the first record in the dataset) does not have a bonus (BONUS equals 0).

Task 7. Cleanse the data for gender.

The Derive node can be used to cleanse the data for GENDER, but Reclassify is preferred because that lets you overwrite the field's values, without adding a new field to the data.

- Add a **Reclassify** node (**Field Ops** palette) downstream from the **Derive** node named **BONUS**, and edit it as follows:
    - Reclassify into: **Existing field**
    - Reclassify field: **GENDER**
    - Click the **GET** button to populate the **Original value** column

      Note: If you receive a message that values are not available, instantiate the data upstream from the Reclassify node.
    - Under **New values**, specify **Female** and **Male**, as depicted in the figure below:



Original value	New value
F	Female
M	Male
f	Female
m	Male

- Run a **Distribution** node (**Graphs** palette) for the **GENDER** Reclassify node.

  The chart shows that the values are as requested.

Task 8. Recode zodiac into broader categories.

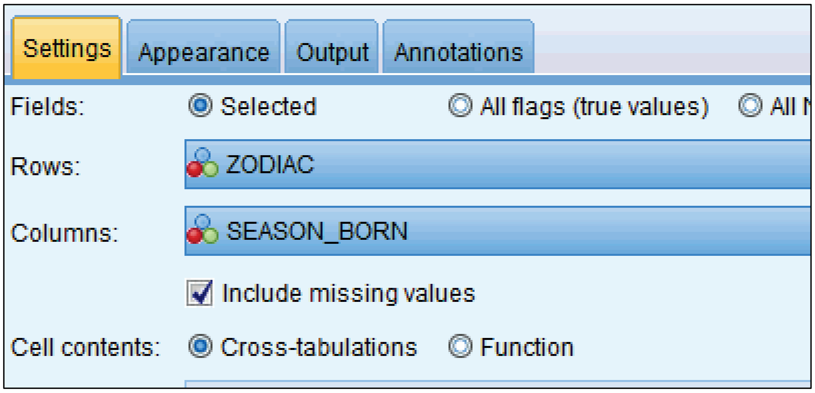
- Add a second **Reclassify** node (**Field Ops** palette) downstream from the first **Reclassify** node, and edit it as follows:
  - Reclassify into: **New field**
  - Reclassify field: **ZODIAC**
  - New field name: **SEASON_BORN**
  - Click the **GET** button to populate the **Original value** column.

    If you receive a message that values are not available, instantiate the data upstream from the Reclassify node.
  - New values: specify the values as follows:

Original value	New value
1.0	Winter
2.0	Winter
3.0	Spring
4.0	Spring
5.0	Spring
6.0	Summer
7.0	Summer
8.0	Summer
9.0	Autumn
10.0	Autumn
11.0	Autumn
12.0	Winter

- Add a **Matrix** node (**Output** palette) downstream from the **SEASON_BORN** Reclassify node, and edit it as follows:
  - Rows: **ZODIAC**
  - Columns: **SEASON_BORN**

  The results appear as follows:



Fields: ● Selected ○ All flags (true values) ○ All N

Rows: ZODIAC

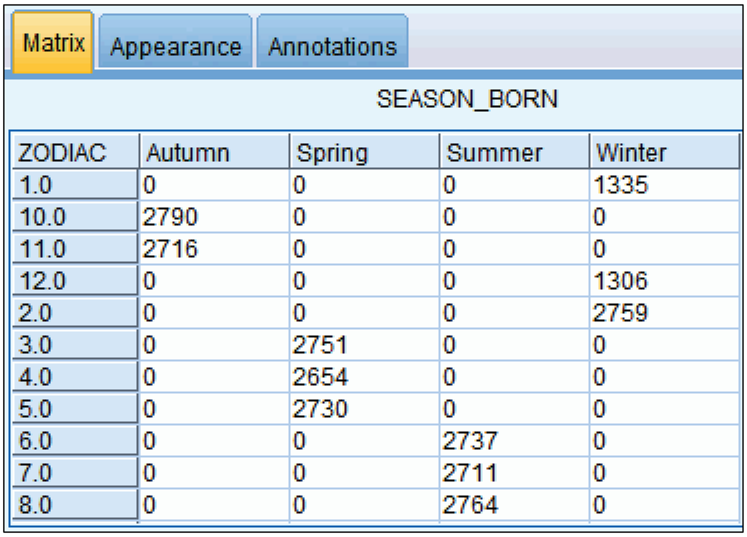
Columns: SEASON_BORN

☑ Include missing values

Cell contents: ● Cross-tabulations ○ Function

- Run the **Matrix** node.

  The results appear as follows:



SEASON_BORN

ZODIAC	Autumn	Spring	Summer	Winter
1.0	0	0	0	1335
10.0	2790	0	0	0
11.0	2716	0	0	0
12.0	0	0	0	1306
2.0	0	0	0	2759
3.0	0	2751	0	0
4.0	0	2654	0	0
5.0	0	2730	0	0
6.0	0	0	2737	0
7.0	0	0	2711	0
8.0	0	0	2764	0

The mapping from original values to new values is correct.

Task 9. Create a field named ORDERDATE_YEAR (Optional).

- Add a **Derive** node (**Field Ops** palette) downstream from the **SEASON_BORN** Reclassify node, and edit it as follows:
  - Derive field: **ORDERDATE_YEAR**
  - Formula: **datetime_year(ORDERDATE)**
- Run a table from the ORDERDATE_YEAR Derive node
- Scroll to the right to view the ORDERDATE_YEAR column

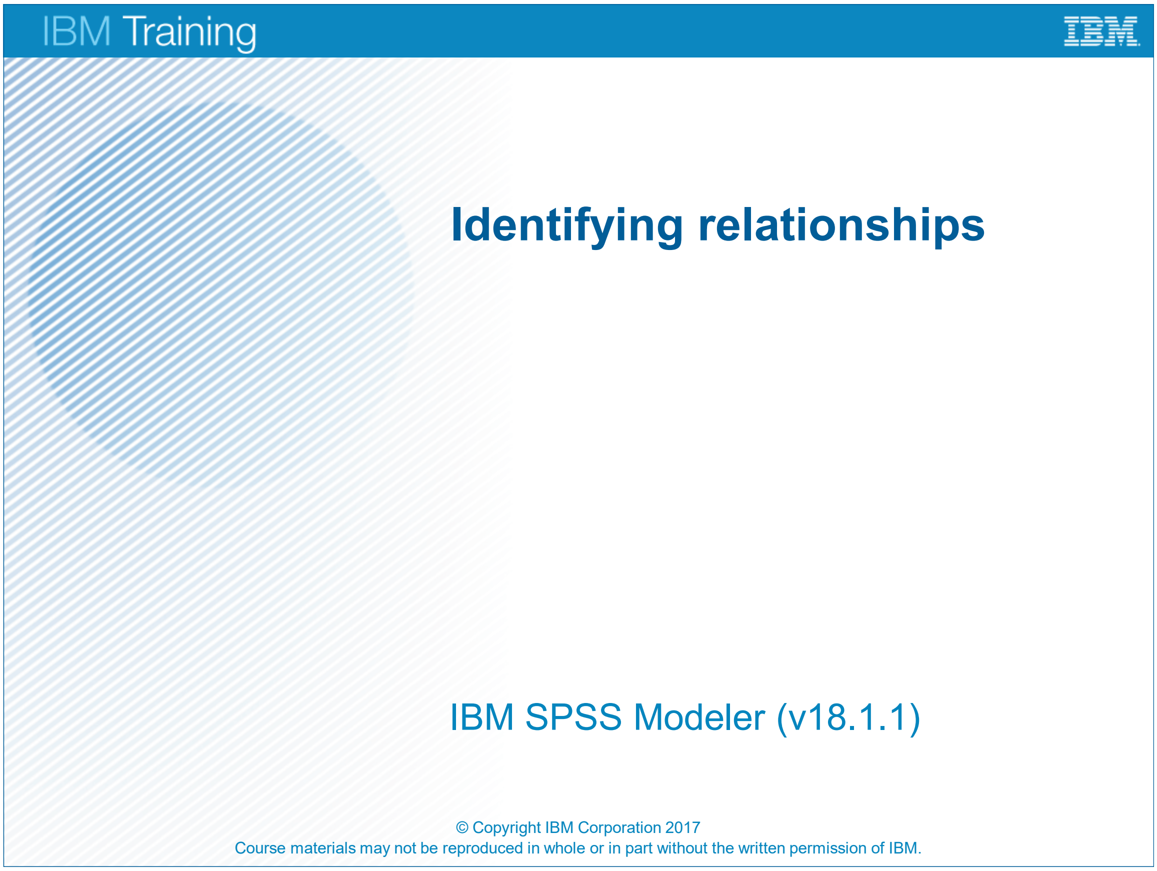
  To check your results: The customer with ID 723.000 (the first record in the dataset) ordered in 2009

Task 10. Exit IBM SPSS Modeler.

- From the **File** menu, click **Exit**, and then exit IBM SPSS Modeler without saving anything.

You will find the solution results in the **C:\Training\0A008\08-Deriving_and_reclassifying_fields\Solutions** folder.

Unit 9 Identifying relationships

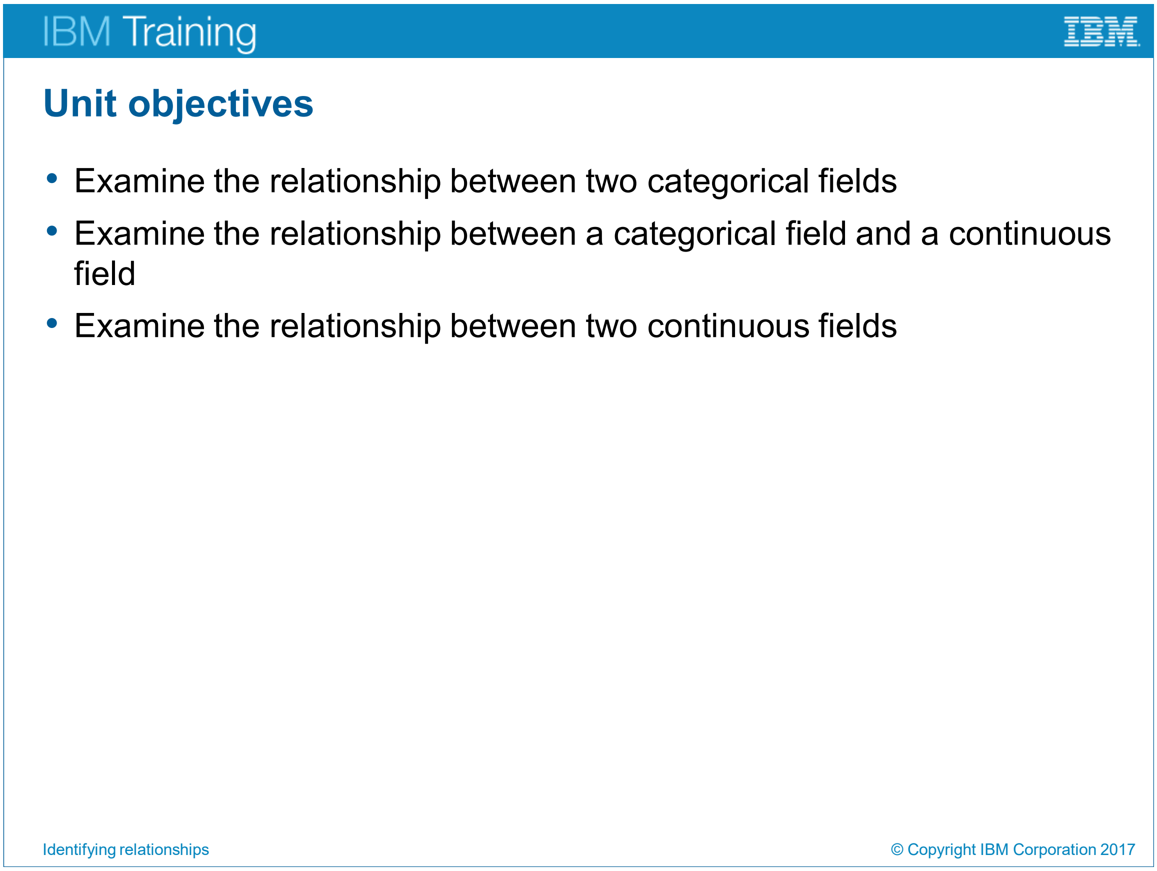


IBM Training

IBM

Identifying relationships

IBM SPSS Modeler (v18.1.1)



IBM Training

Unit objectives

- Examine the relationship between two categorical fields
- Examine the relationship between a categorical field and a continuous field
- Examine the relationship between two continuous fields

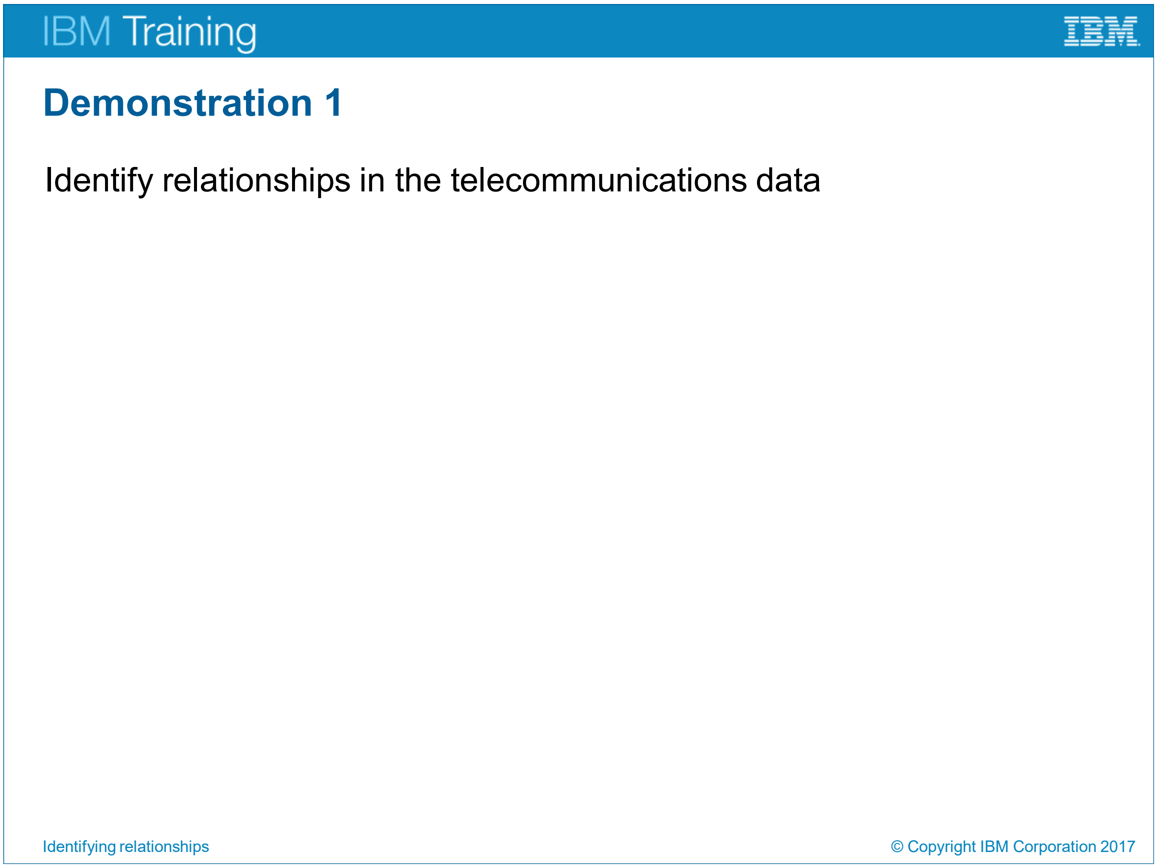
Identifying relationships

Unit objectives

Although building powerful models is key in data mining projects, investigating the relationships between the target field (churn, fraud, credit risk, response, and so forth) and the predictors can still be helpful in answering the questions that motivated the project. You may find that revenue is directly related to length of time as a customer, or that customers with a certain mobile phone plan are more likely to switch providers. Although these patterns are not substitutes for a full model, they can often be used along with a model.

This unit presents methods to examine the relationship between two fields. Before reviewing this unit you should be familiar with the following topics:

- CRISP-DM
- IBM SPSS Modeler streams, nodes and palettes
- Methods to collect initial data
- Measurement levels and storages
- Methods to explore the data



IBM Training

Demonstration 1

Identify relationships in the telecommunications data

Identifying relationships

© Copyright IBM Corporation 2017

Demonstration 1: Identify relationships in the telecommunications data

Demonstration 1: Identify relationships in the telecommunications data

Purpose:
As a data scientist in a telecommunications firm, you want to answer questions such as: Is churn related to handset? Is churn related to the number of dropped calls? And how strong is the relationship between number of products purchased and revenues?

You will answer these questions in this demonstration.

Data file:	**telco x data.txt**
Data folder:	**C:\Training\0A008**
Stream file:	**unit_09_demonstration_1_start.str**
Stream file folder:	**C:\Training\0A008\09-Identifying_relationships\Start**

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

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.
2. When a welcome window appears, click **Cancel**.

   If you have already configured IBM SPSS Modeler in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A008** folder, and then click **Set**.

Task 2. Open the start stream.

1. From the **File** menu, click **Open Stream**, navigate to the **C:\Training\0A008-Integrating_data\09-Identifying_relationships\Start** folder, click **unit_09_demonstration_1_start.str**, and then click **Open**.

Task 3. Examine the relationship between churn and handset.

You will investigate whether the type of handset is related to churn. Both fields are categorical, so you will use a Matrix and a Distribution node.

1. From the **Output** palette, add a **Matrix** node downstream from the **Type** node.
2. Edit the **Matrix** node.

3. Click the **Settings** tab, if necessary.

   Specify the row and the column field on this tab. Only categorical fields are eligible in the field lists.

   By default records with missing values on any of the fields will be included in the table (and in the computation of the Chi-square statistic).
4. Beside **Rows**, select **HANDSET**.
5. Beside **Columns**, select **CHURN**.
6. Click the **Appearance** tab.

   On this tab, request statistics such as counts and percentages. Cells with the highest or lowest values in the table can be highlighted by entering the number of cells in the Highlight top/bottom options. This feature is useful when percentages are displayed.

   Churn rates must be computed by handset, and because handsets make up the rows of the table, you will request row percentages.
7. Select the **Percentage of row** option.

   It is recommended to include row and column totals, so that percentages can be interpreted more easily.
8. Enable the **Include row and column totals** option.
9. Click **Run**.

   For customers with handset ASAD170, the churn rate is 4.627%, whereas it is 94.856% for those with handset ASAD90. The Chi-square statistic points to a significant relationship between HANDSET and CHURN (the probability is 0).

   This is not to say that all handsets differ in churn rate. For example, ASAD90 and CAS30 both have a churn rate of about 95%. Thus, although the Chi-square test tells you that there are differences in churn rates, it does not tell you which handsets differ in churn rate. Further analyses, or a model (for example, CHAID) would be needed to investigate this. Refer to the *Predictive models for Categorical Targets Using IBM SPSS Modeler* course for more information.
10. Close the **Matrix** output window.

    You will run a Distribution graph to support the findings graphically.
11. From the **Graphs** palette, add a **Distribution** node downstream from the **Type** node.
12. Edit the **Distribution** node.
13. Click the **Plot** tab, if necessary.

    On this tab, select the field whose categories will make up the bars. Notice the option to request a distribution graph for a series of flag fields. This will create one bar for each flag field, each bar representing the percentage True for the respective field.

14. Beside **Field**, select **HANDSET**.

    Also select the field to overlay the bars with.

15. Beside **Color**, select **CHURN**.

    You will enable the Normalize by color option to scale bars so that all bars take up the full width of the graph. Categories can be compared more easily that way.

16. Enable the **Normalize by color** option.
17. Click **Run**.

    The Distribution graph shows large differences in churn rates between handsets. It also shows that some handsets (ASAD90, CAS30, SOP10, SOP20) have similar high churn rates. Customers with these handsets are at risk to cancel their subscription.

18. Close the **Distribution** output window.

Task 4. Examine the relationship between churn and number of dropped calls.

In this task you will see whether the number of dropped calls is related to churn. The first field is continuous, the second field is categorical. Therefore you will use a Means node and a Histogram node.

1. From the **Output** palette, add a **Means** node downstream from the **Type** node.
2. Edit the **Means** node.

   The categorical field that defines the groups is specified under Grouping field, the continuous field for which means must be computed is specified under Test field(s).

3. For **Grouping** field, select **CHURN**.
4. For **Test field(s)**, select **DROPPED_CALLS**.
5. Click the **Options** tab.

   You can set threshold probability values to label results as important, marginal, or unimportant (or other labels if you prefer those). By default, importance values below 0.90 are considered Unimportant. Recall that Importance is equal to 1 - probability, so an importance less than 0.9 means that the probability is greater than 0.1.

   Importance values between 0.90 and 0.95 are labeled Marginal, and values greater than 0.95 are labeled Important, in agreement with the commonly used threshold of a probability of 0.05.

   You will use the default values.

6. Click **Run**.

   Customers who cancelled their subscription have 3.906 dropped calls on average, whereas active customers have 2.573 dropped calls on average. Thus, on average, the number of dropped calls is higher for those that left the company than for active customers. This difference is labeled Important (statistically significant at the 0.05 level).
7. Close the **Means** output window.

   You will graphically illustrate the relationship by running a Histogram node.
8. From the **Graphs** palette, add a **Histogram** node downstream from the **Type** node.
9. Edit the **Histogram** node.

   The dialog box resembles the dialog box of the Distribution node.
10. Beside **Field**, select **DROPPED_CALLS**.
11. Beside **Color**, select **CHURN**.

    You can normalize the bars to compare the groups more easily, if preferred (in the Histogram dialog box, click the Options tab, and then enable the Normalize by colors options). This option will not be used here.
12. Click **Run**.

    The histogram confirms that higher values for dropped calls go along with higher churn rates.

    Rather than using CHURN as Color field, you can use it as Panel field, so that you will get two histograms, one for each group. It is recommended to experiment with these options.
13. Close the **Histogram** output window.

Task 5. Examine the relationship between number of products and revenues.

You will explore the relationship between two continuous fields, number of products and revenues.

1. From the **Output** palette, add a **Statistics** node downstream from the **Type** node.
2. Edit the **Statistics** node.

   Correlations will be computed between fields specified in the Examine list and fields specified in the Correlate list.
3. Beside **Examine**, select **REVENUES**.
4. Beside **Correlate**, select **NUMBER_OF_PRODUCTS**.

5. Click the **Correlations Settings** button.

   You can change the thresholds for the importance labels, and the labels themselves. By default, the strength of the correlation is defined by the importance (1 - probability). Even very small correlations can be statistically significant when the dataset has many records. For example, with 100,000 records, a correlation of 0.01 will be statistically significant.

   An alternative way to assess the strength of the relation is based on the absolute value of the correlation. In general, it is recommended that correlations are labeled based on importance rather than by absolute value since the first decision about a correlation is whether it is significant (important). On the other hand, if your dataset uses thousands, maybe millions of records, almost all correlations will be significant and show an importance of 1. Thus, the larger the sample size, the more you should rely on the actual value of the correlation. The smaller the sample size, look first at the importance, then at the actual value of the correlation.
6. Close the **Correlation Settings** sub dialog box.
7. Click **Run**.

   The correlation indicates a strong relationship (meaning significant, because the default correlation settings were used) between the fields. This may come as no surprise, because number of products is the number of products the customer purchased and revenues is the total price he paid for it.
8. Close the **Statistics** output window.
9. From the **Graphs** palette, add a **Plot** node downstream from the **Type** node.
10. Edit the **Plot** node.
11. For **X field**, select **NUMBER_OF_PRODUCTS**.
12. For **Y field**, select **REVENUES**.
13. Click **Run**.

    The plot shows a linear trend.
14. Close the **Plot** output window.

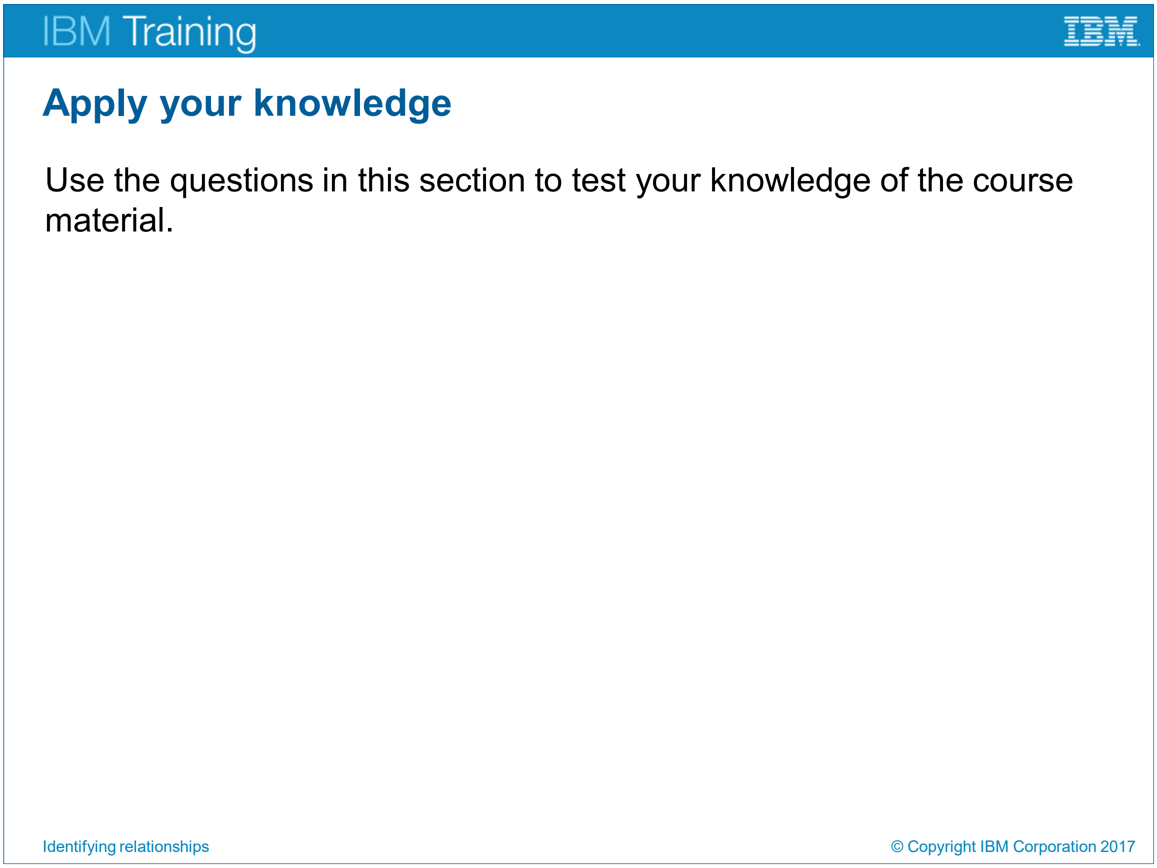
    This completes the demonstration for this unit. You will create a clean state for the exercise.

15. From the **File** menu, click **Close Stream**, and then click **No** when asked to save the stream.
16. From the **File** menu, click **New Stream**.

    Leave IBM SPSS Modeler open for the exercise.

Results:
You examined the relationships between churn and a number of other fields. You used a procedure that was appropriate for the measurement level of the fields in question.

You will find the completed stream in the **C:\Training\0A008\09-Identifying_relationships\Solutions** folder.



IBM Training

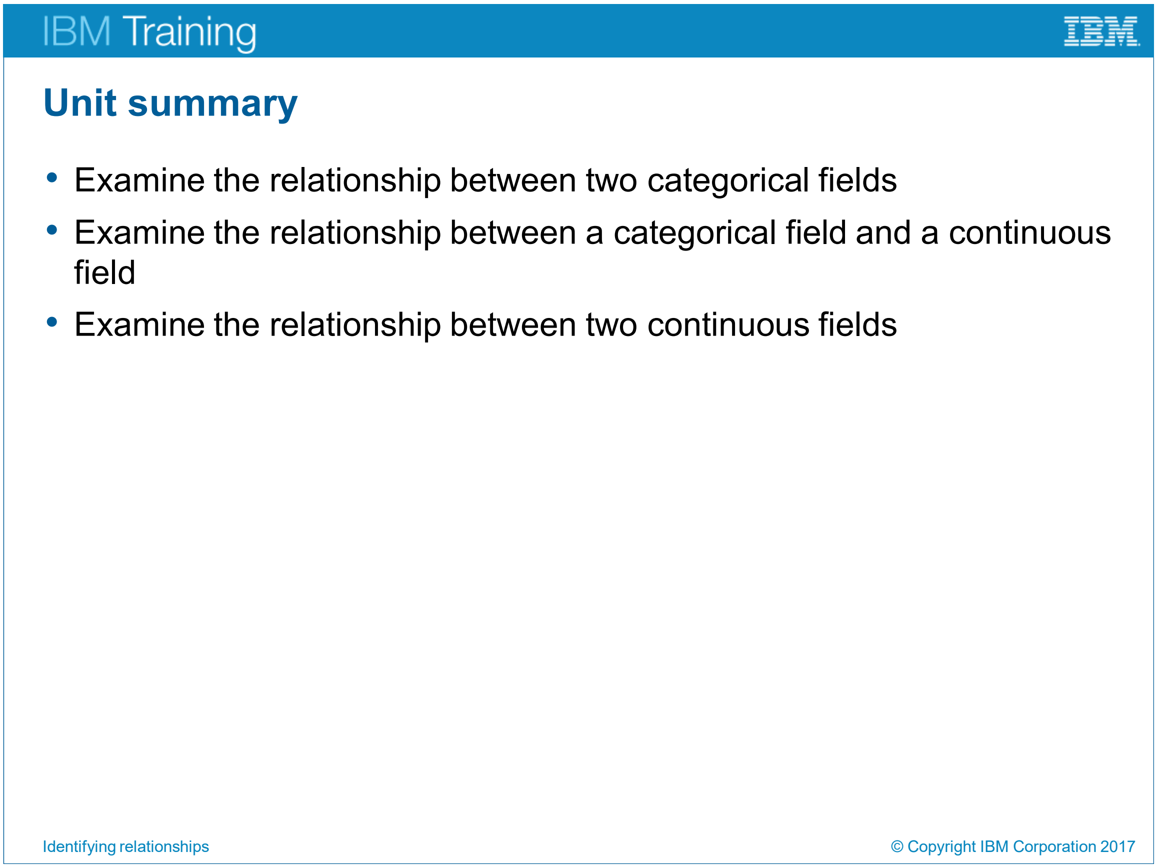
Apply your knowledge

Use the questions in this section to test your knowledge of the course material.

Identifying relationships

© Copyright IBM Corporation 2017

Apply your knowledge



IBM Training

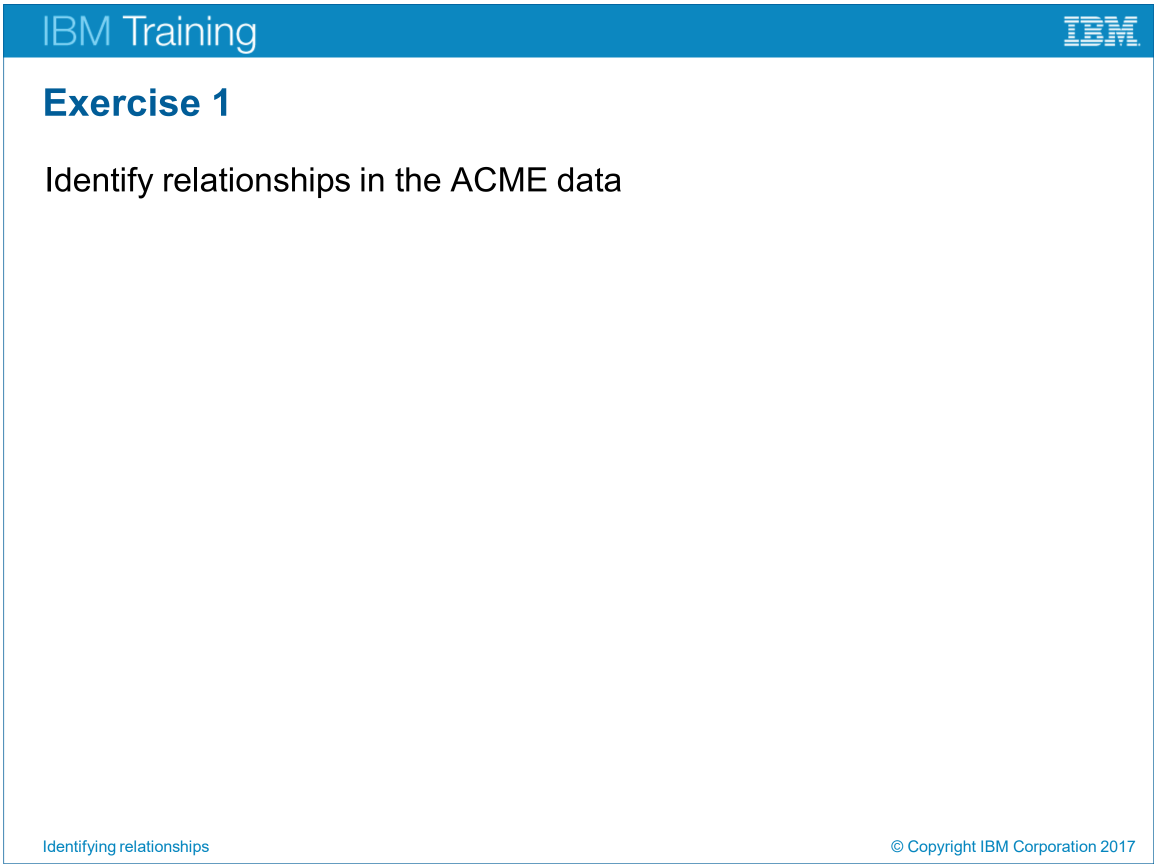
Unit summary

- Examine the relationship between two categorical fields
- Examine the relationship between a categorical field and a continuous field
- Examine the relationship between two continuous fields

Identifying relationships

© Copyright IBM Corporation 2017

Unit summary



IBM Training

Exercise 1

Identify relationships in the ACME data

Identifying relationships

Exercise 1: Identify relationships in the ACME data

Exercise 1: Identify relationships in the ACME data

Data file:	**ACME analysis data.csv**
Data folder:	**C:\Training\0A008**
Stream file:	**unit_09_exercise_1_start.str**
Stream file folder:	**C:\Training\0A008\09-Identifying_relationships\Start**

You work for ACME, a company that sells sports products. Before you start building models you want to examine the relationships in the data, focusing on which fields are related to response.

- Open **unit_09_exercise_1_start.str**, located in the **C:\Training\0A008\09-Identifying_relationships\Start** folder.
- Examine the relationship between **RESPONSE_TO_TEST_MAILING** and **GENDER**, both in tabular and graphical output.

  What percentage of women have responded positively to the test mailing, and what is this percentage for men?

  Is the association between the two fields statistically significant?

  Does the graph support this conclusion?
- Which of the following relationships are statistically significant?
  - The relationship between **MONETARY_VALUE** and **RESPONSE_TO_TEST_MAILING**.
  - The relationship between **FREQUENCY** and **RESPONSE_TO_TEST_MAILING**.
  - The relationship between **RECENCY** and **RESPONSE_TO_TEST_MAILING**.
- Examine the relationship between **RESPONSE_TO_TEST_MAILING** and **CREDITLIMIT**, both in tabular and graphical output (normalize the chart by color, by enabling the appropriate option on the Options tab).

  What is the mean credit limit for those who responded positively to the test mailing, and what is the mean credit limit for those who did not respond positively to the test mailing?
- Exit IBM SPSS Modeler without saving anything.

For more information about where to work and the exercise results, refer to the Tasks and Results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and Results

Task 1. Open the start file.

- From the **File** menu, click **Open Stream**, navigate to the **C:\Training\0A008\09-Identifying_relationships\Start** folder, select **unit_09_exercise_1_start.str** and then open it.

Task 2. Examine the relationship between response and gender.

Both fields are categorical, so use a Matrix node for tabular output, and a Distribution node for graphical output.

- Add a **Matrix** node (**Output** palette) downstream from the **Type** node.
- Edit the **Matrix** node.
- On the **Settings** tab, beside **Rows**, select **GENDER**, beside **Columns** select **RESPONSE_TO_TEST_MAILING**.
- On the **Appearance** tab, select **Percentage of row** (you want to compare men and women and these categories make up the rows of the table, per specification on the Settings tab); also enable the **Include row and column totals** option (to interpret the percentages more easily).

  Note: Rather than having GENDER in the row, RESPONSE_TO_TEST_MAILING in the column, you can have it the other way around. In that case, request column percentages.
- Run the **Matrix** node.

  About 2.9% of the 6,139 women responded positively to the mailing, versus 4.6% of the 3,861 men. Although the difference in percentages is small, it is statistically significant (probability is 0, smaller than the threshold of 0.05). The fact that such a small difference is statistically significant is on the account of the sample size (10,000).
- Add a **Distribution** graph (**Graphs** palette) downstream from the **Type** node.
- Edit the **Distribution** node.
- Beside **Field**, select **GENDER**.
- Beside **Color**, select **RESPONSE_TO_TEST_MAILING**.
- Enable the **Normalize by color** option.
- Run the **Distribution** node.

  There is a small difference between men and women in response rate.

Task 3. Examine the relationship between response and RFM fields.

Recency (how long ago did the customer purchase a product from the company), monetary value (the total amount of all purchases made by the customer), and frequency (how many times did the customer purchase a product) have proven to be important predictors in many projects. These fields are known as RFM fields.

In this dataset, the RFM fields are categorical, so again a Matrix node is needed to assess their relationship with response to the test mailing.

The Matrix node only enables you to specify one row field, and one column field, so you have to examine the relationships one-by-one.

- Add a **Matrix** node (**Output** palette) downstream from the **Type** node.
- Edit the **Matrix** node.
- On the **Settings** tab, for **Rows**, select **MONETARY_VALUE**, for **Columns** select **RESPONSE_TO_TEST_MAILING**.
- On the **Appearance** tab, select **Percentage of row** (we want to compare the categories of monetary value, for example if those who are in the low category have a higher response rate than those in the category high).
- Click **Run**.

  About 0.4% of the 3,354 low category customers responded positively to the mailing, whereas this percentage equals 8.9 for the high category. The relationship is statistically significant (probability equals 0). Although the test does not tell you where the differences lie (for example, do the response percentages for the low and medium category differ?), it tells you that monetary value is an important field for response, and it should be included in modeling. (Note: a model such as CHAID will tell you which categories differ.)

- Repeat the previous steps using **FREQUENCY** as row field.

  Again, the test tells you that there is a statistically significant relationship between frequency and response, although the differences between the categories are smaller than for monetary value.

- Repeat the previous steps using **RECENCY** as row field.

  Again, the relationship is statistically significant.

  All in all, the RFM fields are candidates for including them in a model to predict response. A model, such as CHAID, will, based on statistical criteria, group the categories that have a similar response, and will also show the interaction between the predictors.

Task 4. Examine the relationship between response and credit limit.

In this case, use the Means and the Histogram node, because it pertains to the relationship between a categorical and a continuous field.

- Add a **Means** node (**Output** palette) to the **Type** node.
- Edit the **Means** node.
- Under **Grouping field**, select **RESPONSE_TO_TEST_MAILING**.
- Under **Test field(s)**, select **CREDITLIMIT**.
- Click **Run**.

  The credit limit for those that did not respond to the test mailing equals 4670.3, whereas the credit limit is 4898.4 for those that did respond positively to the test mailing. Thus, those with higher credit limits are more inclined to respond positively to the test mailing. This difference, however, is labeled Unimportant from a statistical point of view.

  A graph enables you to examine the relationship closer.

- Add a **Histogram** node (**Graphs** palette) downstream from the **Type** node.
- Edit the **Histogram** node.
- Under **Field**, select **CREDITLIMIT**.
- Under **Color**, select **RESPONSE_TO_TEST_MAILING**.
- On the **Options** tab, enable the **Normalize by color** option.
- Run the **Histogram** node.

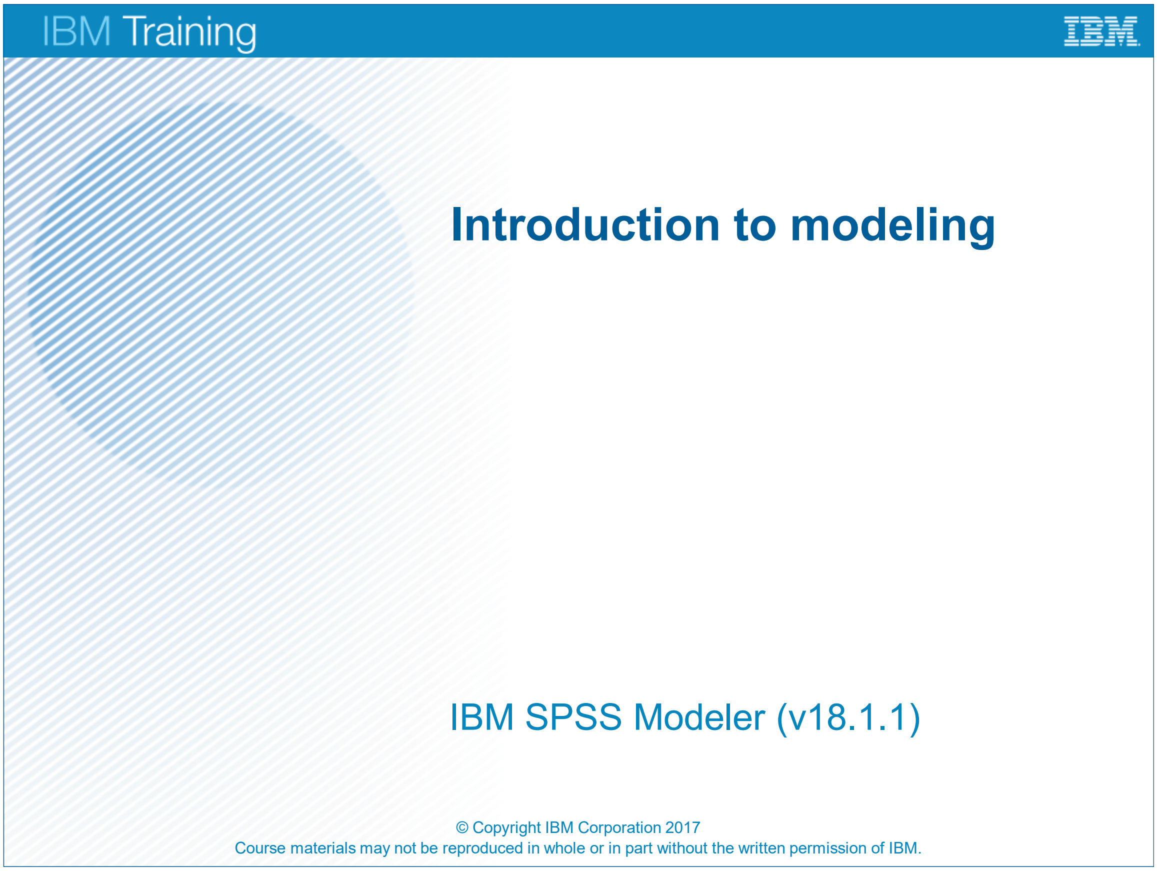
  The graph supports the finding that there is no relationship between credit limit and response to the test mailing.

Task 5. Exit IBM SPSS Modeler.

- From the **File** menu, click **Exit**, and then exit IBM SPSS Modeler without saving anything.

You will find the solution results in the **C:\Training\0A008\09-Identifying_relationships\Solutions** folder.

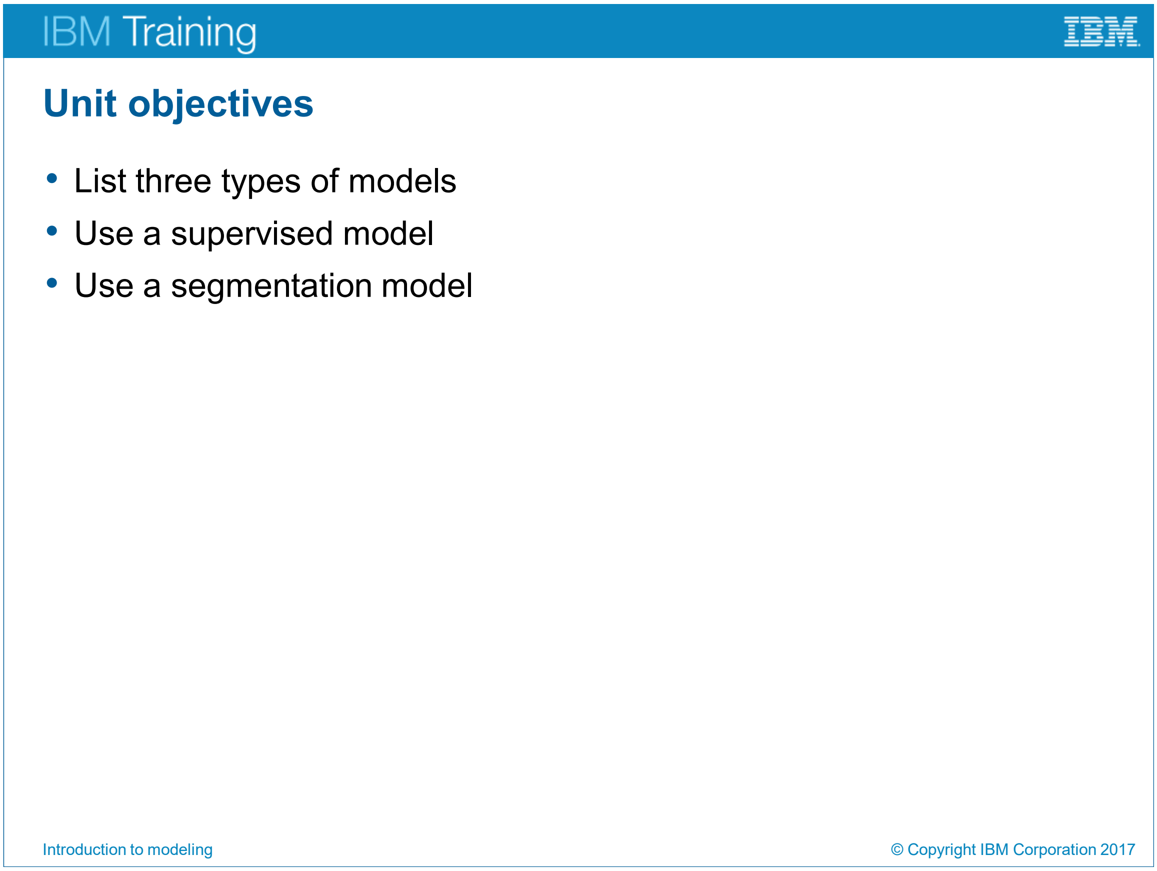
Unit 10 Introduction to modeling



IBM Training

Introduction to modeling

IBM SPSS Modeler (v18.1.1)



IBM Training

Unit objectives

- List three types of models
- Use a supervised model
- Use a segmentation model

Introduction to modeling

© Copyright IBM Corporation 2017

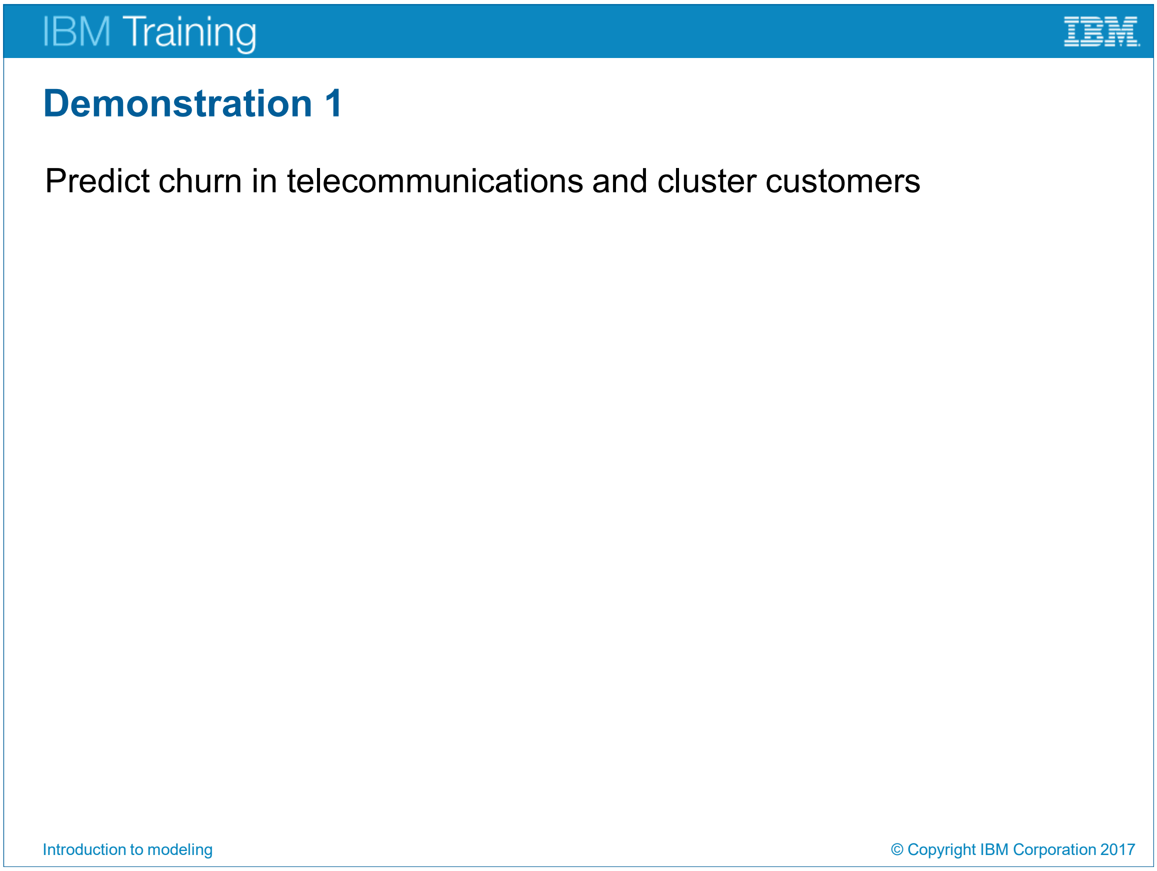
Unit objectives

This unit focuses on the Modeling stage in the CRISP-DM process model.

IBM SPSS Modeler offers many modeling nodes, all located in the Modeling palette. These modeling nodes can be classified into three categories, depending on the type of model.This unit starts off with an overview of types of models and then presents two specific models.

Before reviewing this unit you should be familiar with:

- CRISP-DM
- IBM SPSS Modeler streams, nodes and palettes
- Methods to collect initial data
- Measurement levels and storages
- Methods to explore the data
- Methods to examine the relationship between two fields



IBM Training

Demonstration 1

Predict churn in telecommunications and cluster customers

Introduction to modeling

© Copyright IBM Corporation 2017

Demonstration 1: Predict churn in telecommunications and cluster customers

Demonstration 1: Predict churn in telecommunications and cluster customers

Purpose:
You work as a data scientist for a telecommunications firm. You will first predict churn by using CHAID and Neural Net. You will also compare the accuracy of these models. Then, you will use TwoStep to find groups (clusters) of similar customers, based on usage.

Data file:	**telco x modeling data.xlsx**
Data file:	**C:\Training\0A008**
Stream file:	**unit_10_demonstration_1_start.str**
Stream file folder:	**C:\Training\0A008\10-Introduction_to_modeling\Start**

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

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.
2. When a welcome window appears, click **Cancel**.

   If you have already configured IBM SPSS Modeler in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A008** folder, and then click **Set**.

Task 2. Open the start stream.

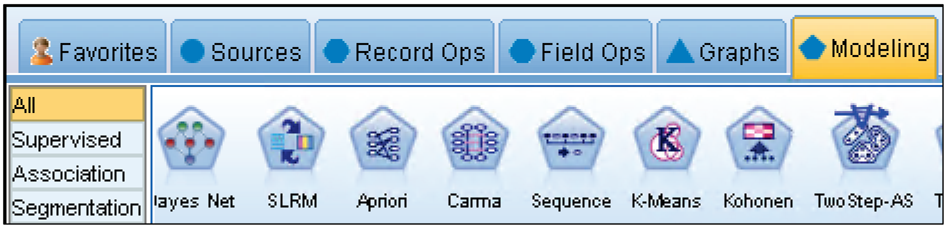
1. From the **File** menu, click **Open Stream**, navigate to the **C:\Training\0A008\10-Introduction_to_modeling\Start** folder, click **unit_10_demonstration_1_start.str**, and then click **Open**.

Task 3. Predict churn.

1. Scroll to the upper branch of the stream, which is labeled **Prediction churn**.
2. Edit the **Type** node.

   The role of HAS_CHURNED is set to target; it will be predicted by fields with role Input. Fields with role None will not be used in the model.
3. Close the **Type** dialog box.
4. Click the **Modeling** palette.

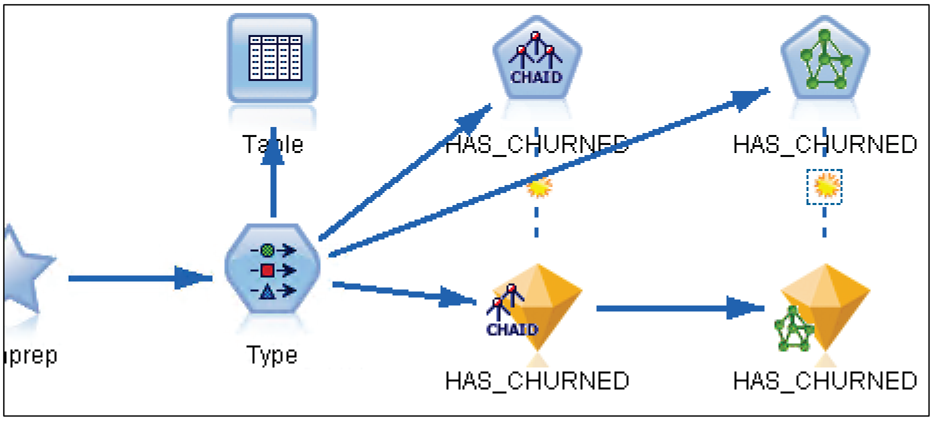
5. Click the **All** sub palette on the left side, as shown in the figure below:



6. Scroll all to the right in the list of models to locate CHAID, and then add the **CHAID** node downstream from the **Type** node.
7. Scroll to the left in the list of models to locate **Neural Net**.
8. Add a **Neural Net** node downstream from the **Type** node.
9. Run the **CHAID** node, and then run the **Neural Net** node.

   When the nodes have been executed, two model nuggets are added downstream from the Type node (and linked to the respective modeling node). To compare the models, you will rearrange the nodes so that both model nuggets are in the same branch of the stream.
10. Disconnect the **Neural Net** model nugget from the **Type** node, place it downstream from the **CHAID** model nugget, and then connect the **CHAID** model nugget to the **Neural Net** model nugget, so that the **Neural Net** node model nugget is downstream from the **CHAID** model nugget.

    The results appear similar to the following:

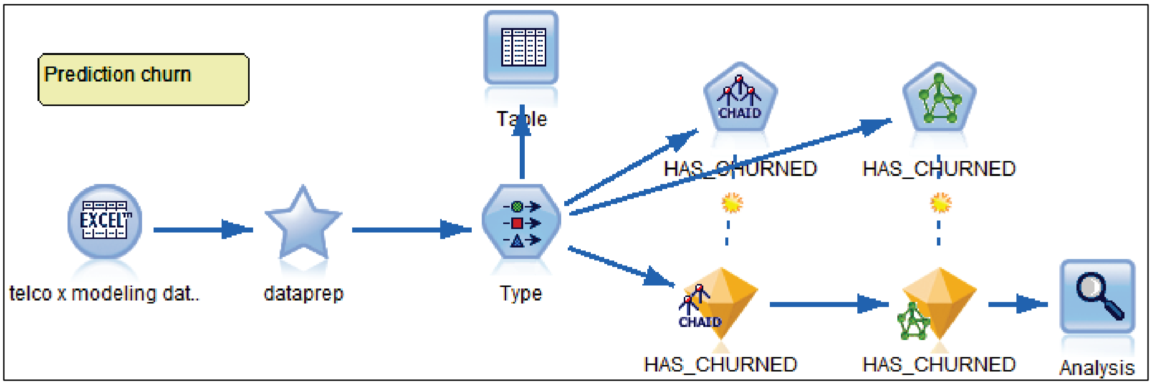


You can use an Analysis node, located in the Output palette, to evaluate the models. The Analysis node compares the target's actual values to the target's predicted values. In a perfect model, the actual values and predicted values coincide. A perfect model will occur in trivial datasets only, or indicates that the wrong predictors have been used.

11. From the **Output** palette, add an **Analysis** node downstream from the **Neural Net** model nugget.
12. Edit the **Analysis** node.
13. Enable the **Coincidence matrices (for symbolic targets)** option.

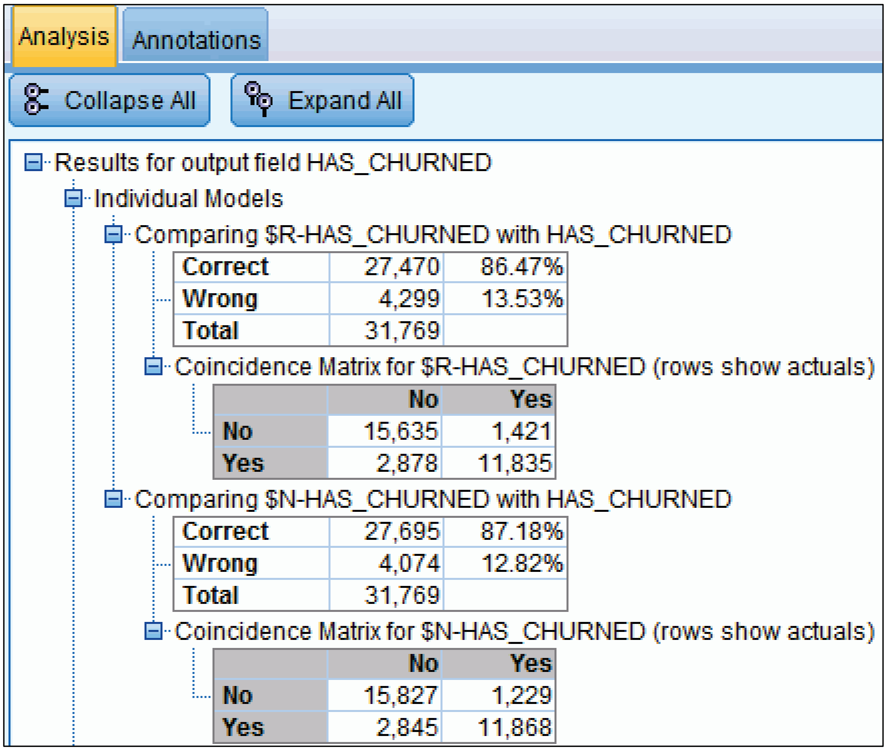
14. Close the **Analysis** dialog box.

    The results appear similar to the following:



15. Run the **Analysis** node.

    The results appear similar to the following:



Results for output field HAS_CHURNED

Individual Models

Comparing $R-HAS_CHURNED with HAS_CHURNED

Correct	27,470	86.47%
Wrong	4,299	13.53%
Total	31,769	

Coincidence Matrix for $R-HAS_CHURNED (rows show actuals)

	No	Yes
No	15,635	1,421
Yes	2,878	11,835

Comparing $N-HAS_CHURNED with HAS_CHURNED

Correct	27,695	87.18%
Wrong	4,074	12.82%
Total	31,769	

Coincidence Matrix for $N-HAS_CHURNED (rows show actuals)

	No	Yes
No	15,827	1,229
Yes	2,845	11,868

The coincidence matrix shows the target's actual values in the rows and the target's predicted values in the column. In data science, the coincidence matrix is also referred to as confusion table or confusion matrix.

The presentation order in the output follows the order of models in the stream: $R-HAS_CHURNED refers to the CHAID model, $N-HAS_CHURNED to the Neural Net model.

Examining the coincidence matrix for CHAID ($R-HAS_CHURNED), the model predicts that 15,635 actual active customers will be active, and that 11,835 customers who left will cancel their subscription. This makes a total of 15,635 + 11,835 = 27,470 correct predictions.

On the other side, 1,421 actual active customers were predicted to cancel their subscription, and 2,878 actual churned customers were predicted to be active. Therefore, there were 1,421 + 2,878 = 4,299 wrong predictions. The accuracy of the CHAID model is the percentage of correct predictions: 27,470 / 31,769 * 100 = 86.47%.

The Neural Net model ($N- churn) is more accurate than the CHAID model (accuracy is: 87.18%). However, the CHAID model provides insight: the tree shows the profile of customers who are likely to end their subscription. It is a business decision which model to take. Or, maybe you want to run other models and examine their accuracy.

As a note, in data science, it is common practice to build models on a training set, and evaluate models on a testing set. The Partition node, located in the Field Ops palette, is designed to split the data into a training and testing set. Refer to the online Help for more information about the Partition node; or refer to the *Advanced Data Preparation Using IBM SPSS Modeler* course.

16. Close the **Analysis** output window.

Task 4. Create homogeneous groups of customers.

1. Scroll to the lower branch of the stream, which is labeled **Creating clusters**.

   You will use TwoStep clustering because this model will find the number of clusters automatically.

2. Edit the **Type** node.

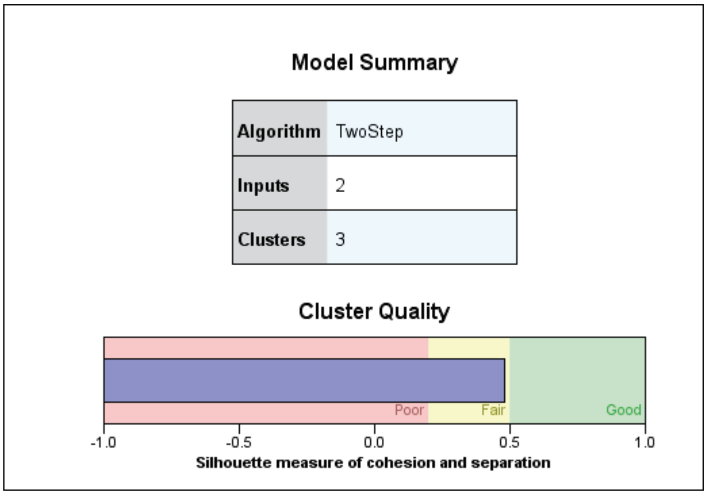
   Two fields, BILL_PEAK and BILL_OFFPEAK, have role Input, so the clusters will be formed on the basis of these two fields. Records with similar values for BILL_PEAK and BILL_OFFPEAK will be placed into the same cluster.

   No field is designated as target.

3. Close the **Type** dialog box.
4. Click the **Modeling** palette, and then, on the left side, click the **Segmentation** sub palette.
5. Add a **TwoStep** node downstream from the **Type** node.
6. Run the **TwoStep** node.

7. Edit the **TwoStep** model nugget that was generated.

   The results appear as follows:



Model Summary

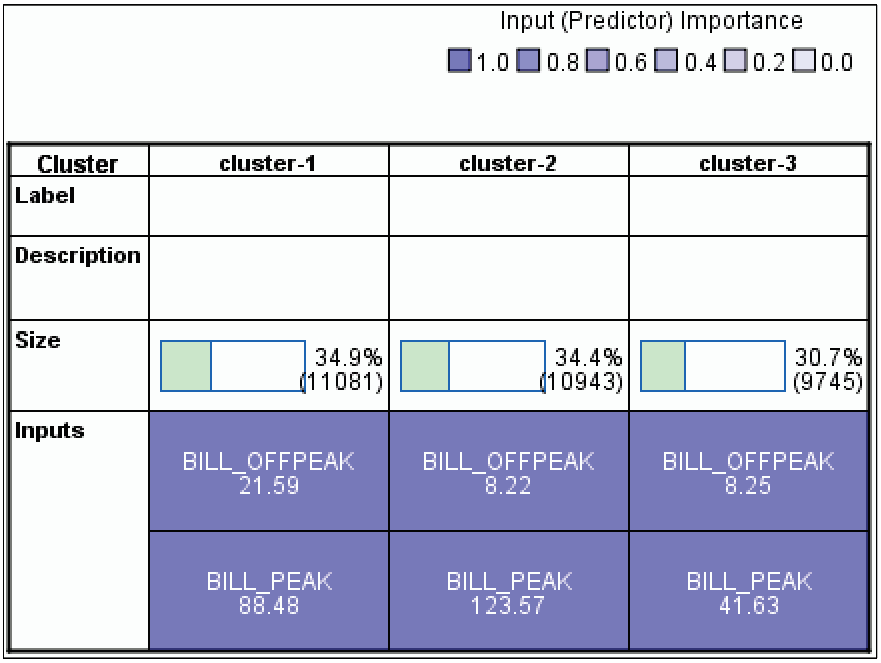
Algorithm	TwoStep
Inputs	2
Clusters	3

Cluster Quality

Three clusters are created; the solution is acceptable ("fair").

8. At the bottom of the left pane, beside **View**, select **Clusters**.

   The results appear similar to the following:



Input (Predictor) Importance

▨ 1.0 ▨ 0.8 ▨ 0.6 ▨ 0.4 ▨ 0.2 □ 0.0

Cluster	cluster-1	cluster-2	cluster-3
Label			
Description			
Size	34.9% (11081)	34.4% (10943)	30.7% (9745)
Inputs	BILL_OFFPEAK 21.59	BILL_OFFPEAK 8.22	BILL_OFFPEAK 8.25
	BILL_PEAK 88.48	BILL_PEAK 123.57	BILL_PEAK 41.63

Cluster 1 is comprised of customers with a, relatively, high off-peak usage; cluster 2 is characterized by a high usage in the peak hours; cluster 3 is a segment of customers who are inactive.

It is up to you to decide if this solution is useful. You could also explore if the cluster field is a valuable predictor for churn.

9. Close the **TwoStep** model nugget.

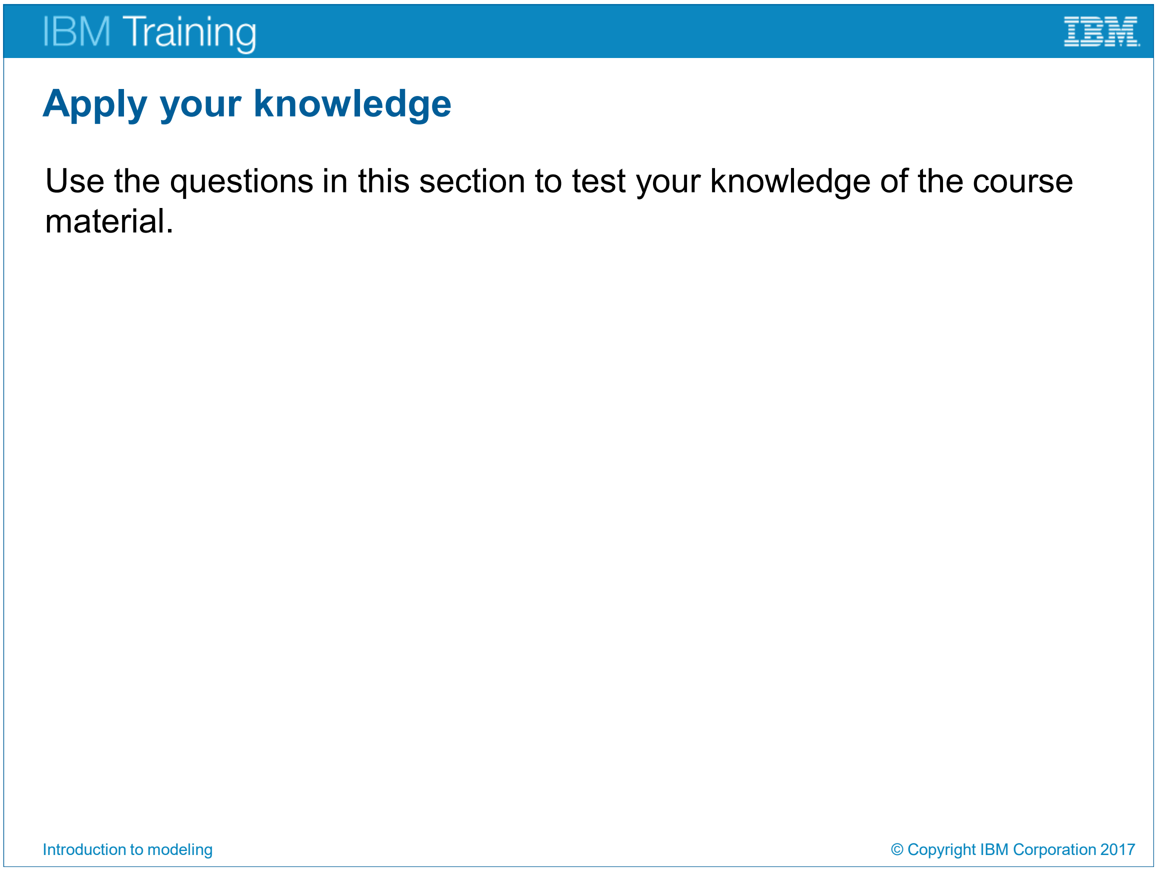
   This completes the demonstration for this unit. You will create a clean state for the exercise.

10. Close the **TwoStep** dialog box.
11. From the **File** menu, click **Close Stream**, and then click **No** when asked to save the stream.
12. From the **File** menu, click **New Stream**.

    Leave IBM SPSS Modeler open for the exercise.

Results:
You predicted churn with two supervised models, CHAID model and Neural Net, and you compared the accuracy of the models. You also looked for clusters of similar customers, using the TwoStep segmentation model.

You will find the completed stream in the **C:\Training\0A008\10-Introduction_to_modeling\Solutions** folder.



IBM Training

Apply your knowledge

Use the questions in this section to test your knowledge of the course material.

Introduction to modeling

Apply your knowledge

Question 1: Based on transactional data such as minutes of outgoing calls, minutes of incoming calls, and text messaging, a telecommunications firm clusters its customers and finds groups such as "leaders" and "followers". This is an example of which type of model?

A. A supervised model

B. A segmentation model

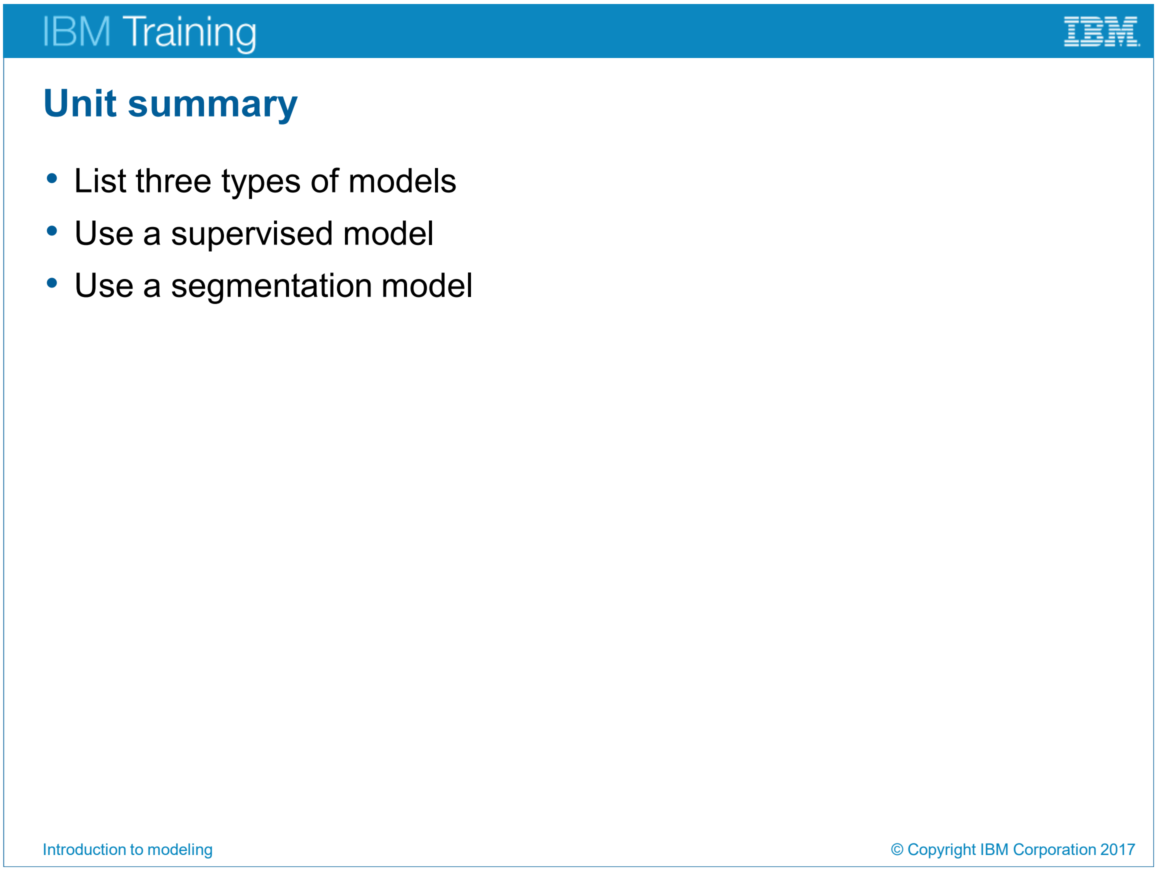
C. An association model

Question 2: A retailer runs an analysis on what customers have in their shopping carts to find out which product combinations are popular. This is an example of which type of model?

A. A supervised model

B. A segmentation model

C. An association model



IBM Training

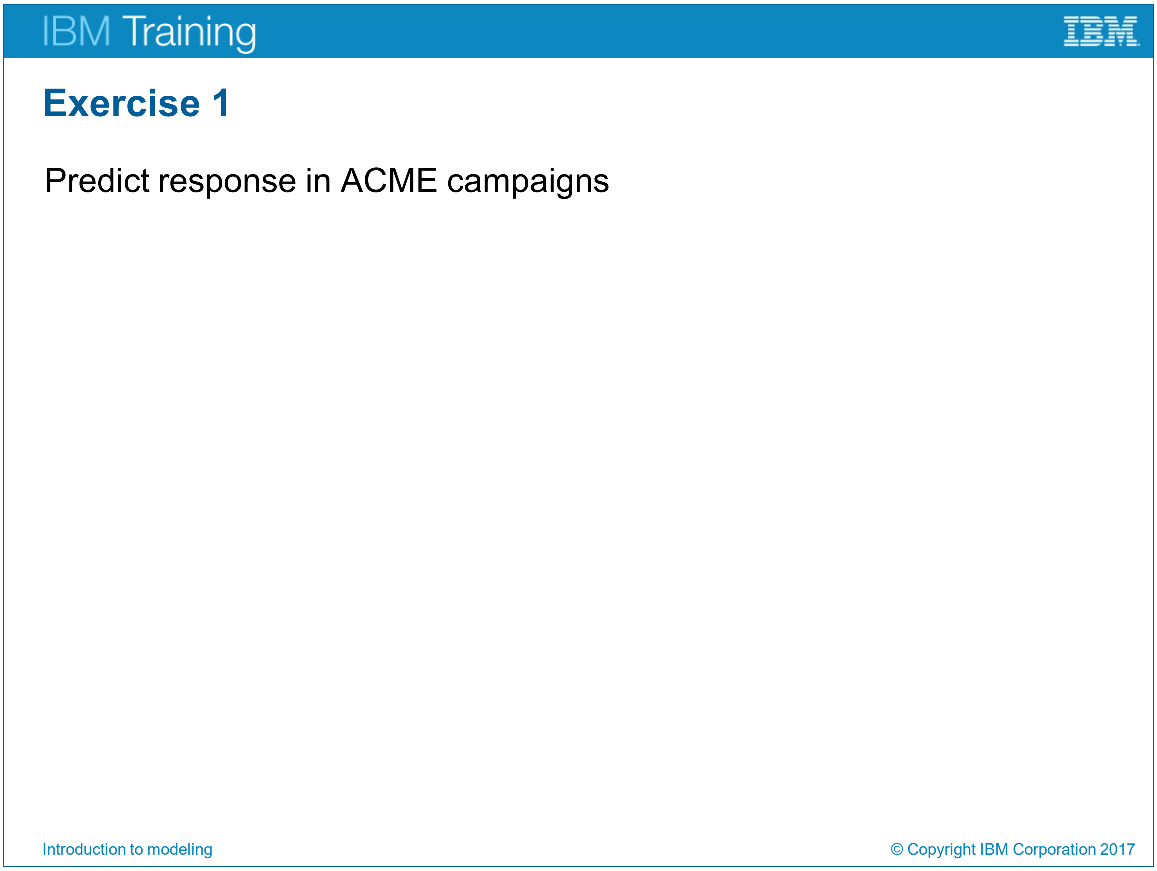
Unit summary

- List three types of models
- Use a supervised model
- Use a segmentation model

Introduction to modeling

© Copyright IBM Corporation 2017

Unit summary



IBM Training

Exercise 1

Predict response in ACME campaigns

Introduction to modeling

Exercise 1: Predict response in ACME campaigns

Exercise 1: Predict response in ACME campaigns

Data file:	**ACME analysis data.csv**
Data file:	**C:\Training\0A008**
Stream file:	**unit_10_exercise_1_start.str**
Stream file folder:	**C:\Training\0A008\10-Introduction_to_modeling\Start**

You work at ACME, where you prepared a dataset for modeling. Now the time has come to build a model to predict response to the test mailing, and, if the model is satisfactory, to apply it to the customers that were not included in the test mailing. Also, you will use a segmentation model to find clusters of customers in the ACME database.

- Open **unit_10_exercise_1_start.str**, located in the **C:\Training\0A008\10-Introduction_to_modeling\Start** folder.
- Set roles in the **Type** node, so that **RESPONSE_TO_TEST_MAILING** will be predicted using **RECENCY, FREQUENCY** and **MONETARY_VALUE** as predictors.
- Use **CHAID** to predict the target.

  When the model nugget has been added to your stream, examine the tree. What is the predicted response for customers in the high frequency category, high recency category and high monetary value category?
- Use an **Analysis** node (**Output** palette) to assess the model's accuracy. Edit the **Analysis** node and ask for the coincidence matrix.

  What is the accuracy (percentage of records predicted correctly)? How many of the customers who responded positively have been identified as such by the model?
- Assuming that the model that was created is satisfactory, apply the model to the customers that were not in the test mailing, and select all those customers that are predicted to respond positively; export their data to a text file named **prospects for mailing.txt**. (If IBM SPSS Modeler issues a message to overwrite the file, click OK.)

  How many of the customers who were not included in the test mailing, are predicted to respond?
- In the lower stream, set the roles so that you can cluster records using **RECENCY, FREQUENCY** and **MONETARY_VALUE** as inputs.

- Use the **TwoStep** segmentation model to cluster records (all 30,000 customers). Then, edit the model nugget that was generated.

  How many clusters are found? Is the solution acceptable?

  Profile the clusters in terms of the input fields**.**
- Exit IBM SPSS Modeler without saving anything.

For more information about where to work and the exercise results, refer to the Tasks and Results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and Results

Task 1. Open the start stream.

- From the **File** menu, click **Open Stream**, navigate to **C:\Training\0A008\10-Introduction_to_modeling\Start**, and then open **unit_10_exercise_1_start.str**.

Task 2. Set roles.

- Edit the upper **Type** node, set **Role** for **RECENCY**, **FREQUENCY** and **MONETARY_VALUE** to **Input**, and then set **Role** for **RESPONSE_TO_TEST_MAILING** to **Target**.

Task 3. Use CHAID to predict response.

A model will be built on only those customers that were included in the test mailing.

- Add a **CHAID** node (**Modeling** palette) downstream from the **Type** node, and then run the **CHAID** node.

  A model nugget is generated, connected to the Type node, and linked to the CHAID node.

Task 4. Assess the model's accuracy.

- Add an **Analysis** node (**Output** palette) downstream from the **CHAID** model nugget.
- Edit the **Analysis** node, and then enable the **Coincidence matrices (for symbolic targets)** option.
- Click **Run**.

  The accuracy is 97.04%; of the 199+162 = 361 customers who actually responded positively to the mailing (refer t the row labeled T), 162 customers were identified as such.

Task 5. Use the model to score other customers.

Assuming that the accuracy is satisfactory, apply the model to the customers that were not in the test mailing: select these customers, and then use the model nugget.

- Copy and paste the **CHAID** model nugget downstream from the **Select** node named **NOT IN TEST MAILING**.
- Add a **Select** node downstream from the model nugget, and then add the expression **'$R-RESPONSE_TO_TEST_MAILING' = "T"** (alternatively, generate the **Select** node from a **Table** output window).
- Add a **Table** node downstream from the **Select** node, and then run the **Table** node.

  This will show that 254 customers are predicted to respond.
- To export the data for the 254 customers predicted to respond, add a **Flat File** node (**Export** palette) downstream from the **Select** node, edit the **Flat file** node, specify the file name **prospects for mailing.txt** and then run the **Flat File** node.

  If IBM SPSS Modeler issues a message to overwrite the file, click **OK**.

Task 6. Set roles to cluster customers.

- Scroll to the lower part of the stream canvas.
- Edit the **Type** node, and then set **Role** for **RECENCY**, **FREQUENCY** and **MONETARY_VALUE** to **Input**. Ensure that all other fields have **Role None**.

Task 7. Create clusters of customers.

- Add a **Two-Step** node (**Modeling** palette - **Segmentation** sub palette) downstream from the **Type** node, and then run the **TwoStep** node.
- Edit the **TwoStep** model nugget.

  Three clusters are found; the solution is acceptable ("fair").
- In the left pane, at the bottom of the window, beside **View**, select **Clusters**.

  Cluster 1 is a group of medium usage customers. Cluster 3 is a group of high value customers, although with low frequency. Customers in Cluster 2 represent low monetary value, although they have many transactions.

Task 8. Exit IBM SPSS Modeler.

- From the **File** menu, click **Exit**, and then exit IBM SPSS Modeler without saving anything.

You will find the solution results in the **C:\Training\0A008\10-Introduction_to_modeling\Solutions** folder.