# Chapter 5: Data preparation and manipulation

## Introduction

This lesson includes an introduction and focus on data preparation capabilities available from the Altair Analytics Workbench Workflow perspective.

A data preparation scenario will be introduced which will be demonstrated using blocks available from the data preparation group. Other data preparation blocks will also be demonstrated prior to a summary.

Figure 1: Contents

A screenshot of a computer

Description automatically generated

Data preparation is involved and can take a considerable amount of time whether for etl, modelling, deployment or other data processing operations. There are many things to consider and may be many iterations before things are acceptable.

Figure 2: Introduction

A diagram with text and words

Description automatically generated

When preparing data questions such as what sources to include, how to join, what to include, what to summarise and what variables to derive are all considerations that require adequate time to assess and implement correctly.

When modelling, the added need to develop a dependent variable requires considerable data preparation time but done properly should make further processing run smoothly.

Figure 3: Workflow blocks

A screenshot of a computer

Description automatically generated

Altair Analytics Workbench provides an array of easy to configure data preparation blocks. These are located in the Data Preparation group and include the most commonly used techniques such as aggregate, binning, filter, impute merge, join and more.

For more complex data preparation transformations, code blocks can be used to augment any Workflow with the language of SAS, R, Python or SQL and add additional processing functionality.

To highlight Altair Analytics Workbench Workflow perspective capabilities for data preparation consider the following scenario.

Figure 4: Scenario

A diagram of data flow

Description automatically generated

Transactional information for January and February are joined into one large file and the results

aggregated and summaries created. The results are enriched with demographic details to generate the final file, which will be used to illustrate additional data preparation blocks.

## Demonstration

So on to a demonstration. To begin, a new Workflow is added to the project folder created in a previous lesson. The project is right-clicked and New > Workflow selected. The Workflow is named *data\_prep*. Finish is clicked and the Workflow is created in the project and opened.

The data used in this demonstration is located in the eLearning data folder and can be navigated to using the File Explorer. The goal here is to prepare some transactional data contained in two CSV files:

* Jan\_Altair.csv
* Feb\_Altair.csv

Double-clicking open the files and here both are opened and shown side by side.

Figure 5: Data to use

A screenshot of a computer

Description automatically generated

As can be seen both files contain transactional data and the same four variables, an id variable, and other variables observing the number of products purchased, the total price paid and the month the transaction was conducted.

There are multiple transactions for some ids and there are 1215 observations in *Jan\_Altair* and 2520 in *Feb\_Altair*, giving 3735 observations in total.

The goal is to gather these observations into one file, group by id, create summaries for number of products and price and enrich the results with demographic detail from the file *demographics\_Altair.csv*.

Opening this file from the File Explorer it can be seen it contains variables such as martial status, sex, hours worked and occupation. Notice that id values seem to be unique and there are 20000 observations in total.

The variable id will be used to link data across files and attach demographics to the summarised transactional information.

Closing all files and returning to the Workflow, the first thing to do is import the transactional data, this can be done by dragging and dropping both files from the File Explorer onto the Workflow canvas.

Both blocks have a red configuration status indicator with the message: Import not yet configured. Opening and running both blocks addresses this issue as denoted now by the green execution status indicator.

Both files are opened with the Data Profiler and split screen is used to place them side by side.

Figure 6: Files open with the Data Profiler

A screenshot of a computer

Description automatically generated

From the summary view it can be seen that there are 1215 and 2520 observations across files and both have the same 4 variables. From the Data tab variables and observations can be viewed for both files.

Figure 7: Viewing data with the Data tab

A screenshot of a computer

Description automatically generated

Again, notice that across both files there are multiple transactions associated with the same id and some id's have transactions across both months.

The Data Preparation group contains a series of point and click blocks to assist in data preparation tasks. There are blocks for aggregating, binning, filtering, imputing missing values, calculating new columns and more.

Figure 8: Data Preparation group

A screenshot of a computer

Description automatically generated

## Merge block

The Merge block can be used to merge all observations from both transactional files. This is dragged onto the Workflow canvas. Notice the configuration status messages: Block requires at least two dataset inputs and: No merge operation selected.

Figure 9: Merge block

A screenshot of a computer

Description automatically generated

The imported files are renamed to Jan and Feb respectively and then connected to the merge block and as a result, there is now only one configuration status message: No merge operation selected.

The Merge block is double-clicked to access its configuration dialog. Options are included to select the merge operation and a By Variable.

Figure 10: Merge block dialog

A screenshot of a computer

Description automatically generated

Both connected datasets are listed and the available merge operations are: concatenate, interleave, and one to one.

Concatenate stacks observations from one file on top of the other. Missing values may result if some variables are present in only one file.

Interleave works on a observation index and requires a by variable as indicated by the error message. One or more by variables can be included and will result in an output dataset ordered by the by variable, in this instance id.

The *one-to-one* option is based on observation row index and does not use a by variable. The first file is used as a base, if there are additional variables in other file(s), they will be added. Values used are taken by matching row index number. The number of observations in the final file is equal to the number of observations in the smallest input dataset, here 1215, choosing to include excess rows will results in a file size equal to the largest connected dataset and some observations will have missing values.

The appropriate option to use here is concatenate, which does not require a by variable. OK is clicked and the process runs. Opening the resulting dataset with the Data Profiler, it can be seen there are 3735 observations and four variables, as expected.

Viewing the data, 1215 January observations appear first, followed by 2520 February observations.

## Aggregate block

The next step is to group observations by id and generate some summaries such as average and total for number of products and total price.

The Aggregate block can be used for this purpose, the block is dragged from the Data Preparation group to the Workflow canvas and connected to the merged dataset.

Figure 11: Aggregate block

A screenshot of a computer

Description automatically generated

As with all newly connected blocks the configuration has not been set and the status indicator message: No function provided reflects this.

Double-clicking opens the block configuration dialog. The dialog has two pages. Grouping Variable Selection allows selection of one or more grouping variable, here *ID* is double clicked and automatically moved to the Selected Grouping Variables list. Observations will now be grouped by *ID*.

From the expressions page, new variables can be created based on aggregate functions. Options to specify the variable to base the calculation on, the function to apply and the name to assign to the calculated variable are available.

Figure 12: Expression’s page

A screenshot of a computer

Description automatically generated

Here there is interest in creating new variables that calculate the sum and average number of products purchased and total price.

To create average number of products ,The function dropdown provides a long list of possibilities, here average is selected, *num\_products* is selected from the variable selection pop up window., notice the new variable name is added automatically but can be changed if desired.by default an the suffix \_AVG is added.

Expressions can be added or deleted by selecting the appropriate icon. Clicking the Add expression below icon adds a new expression entry. This time sum selected and the same variable is chosen and the same two calculations are repeated for the variable *tot\_price*.

OK is clicked to complete the process and the resulting dataset is opened with the Data Profiler.

Figure 13: Summary View

A screenshot of a computer

Description automatically generated

As can be seen there are 239 observations and 5 variables. From the Data View it can be seen that

All transactional information has been summarised and there is now one row for each unique id.

Figure 14: Data View

A screenshot of a table

Description automatically generated

The next step is to enrich the transactional data with demographic detail contained in the file *demographics\_Altair.csv*.

First, the file is dragged onto the Workflow canvas from the File Explorer, configured and imported. Opening the file with the Data Profiler, there are 20000 observations and 8 variables. Viewing the data, it can be seen it contains an id variable and this will be used to join demographic details to the transactional data.

The join block will be used for this but prior to proceeding it is wise to ensure there are no duplicates contained in the demographics file. To do this a Sort block is added and docked with the demographic’s dataset. The sort block can be used to sort data based on one or more variables, here, *ID* is selected and the option: Remove duplicate keys chosen.

Figure 15: Sort block

A screenshot of a computer

Description automatically generated

Opening the results with the Data Profiler and from the Summary View tab it can be seen there are 20000 observations, no change from the previous file, regardless, this is good practice and something to bear in mind.

## Join block

The final step is to enrich the transactional data with demographics. At this point the file names are changed to *transactions* and *demographics* as this will make further processing easier.

To complete the process, a Join block is dragged from the Data Preparation group to the Workflow canvas. As always, the configuration settings indicator is red and displays two messages: At least two connections being required and At least one join must be defined.

Both datasets are connected to the Join block and its configuration dialog accessed, where both files are illustrated listing variables in both.

Figure 16: Join configuration

A screenshot of a computer

Description automatically generated

Here there is interest in retaining all observations in the transactions file and enriching with demographics based on the variable id.

The join variable, *ID*, is dragged from transactions to *ID* in demographics. A join is created and can be defined by either right-clicking the line joining the variables and selecting the option: Configure Join or simply by double-clicking. Either will open the Join Properties dialog.

The left dataset is demographics and the right dataset is transactions with *ID* as the column for both.

There are six join types to choose from and also six join operators. To retain all observations from transactions and matching observations from demographics the Join Type is right Outer and the Join operator selected as equals.

There are also two additional tabs in the Join block: SQL and Preview. The SQL tab outputs the code for the join and Preview illustrates what results will look like.

Clicking CTRL+S updates settings and applies the join. Returning to the Workflow and opening the resulting dataset with the Data Profiler, it can be seen that there are 239 observations, and summarised transactional information has been enriched with demographics, the data can be viewed in greater detail from the Data tab.

## Binning variables with the Binning block

At this stage all transactions have been merged, summarised and enriched with demographic detail but there may be a need to process the data further.

The data preparation group contains blocks that can assist, for example the Binning block. The block is dragged onto the resulting dataset, connected, and double-clicked to access its configuration dialog.

Figure 17: Binning block

A screenshot of a computer

Description automatically generated

The binning block can be used to create categories or bins for any variable. The dialog has a number of areas, first a variable to bin must be selected by moving it from the Unselected Variables list to the Selected Variables list, here *tot\_price\_sum* is chosen.

The binned variable is the variable that will be created and can be renamed, more on this shortly.

Once selected there are four binning types to choose from: Optimal, Equal Width, Equal Height and Winsorised. Options for each can be set from Binning Preferences accessible by clicking the gears icons.

Figure 18: Binning Preferences

A screenshot of a computer

Description automatically generated

The Default bin count and Winsorrate values apply to Equal Height, Equal Width and Winsorised binning and the optimal Binning settings apply to the option Optimal.

Equal height binning attempts to create *n* equally populated bins, whereas equal width creates bins with the same width, Winsorised binning removes a proportion of the largest and smallest values, here 20%, equal width bins are calculated and the removed observations are returned to the first and last bins respectively. This ensures that the first and last bins are adequately populated.

Optimal binning settings include the ability to choose the measure, specify the starting and maximum number of bins to create, how to deal with missing values and whether to force monotonicity.

There are also options to ensure resulting bins are adequate in terms of size. The binning type: Optimal requires selection of a Dependent variable and its treatment.

Here, there is interest in creating a new variable that records whether total price sum is either *low* or *high*. A good place to start is to choose Equal Height as the Binning type. From Binning Preferences, the default bin count is set to 2.

The Bin Variables button is clicked and this generates bins for the selected variable with bin cut points visible.

Figure 19: Equal Height binning applied

A screenshot of a computer

Description automatically generated

The Bin Statistics area relays the size and % of each bin in a frequency table. The view can be changed by hovering and selecting an appropriate icon.

Here, the icon Show as frequency chart is selected, this view shows how the created bins are distributed across the variable *tot\_price\_sum*, bear in mind this is the variable that binning is based on. The option to Show as pie chart is also available and bins can be isolated and compared by selecting them from the bins area.

Figure 20: Show as frequency chart

A graph with numbers and lines

Description automatically generated

From the Variable drop-down, *DV* is selected. Now the chart illustrates how the bins are distributed across the categories of *DV.* Changing the chart to a pie illustrates things more clearly.

Figure 21: Selecting DV as Variable

A screenshot of a graph

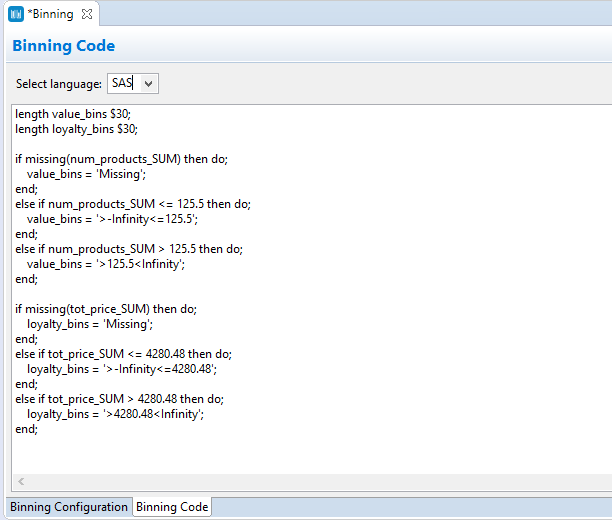
Description automatically generated

As can be seen there is little difference across the *DV* categories in terms of the bin distributions.

The new variable is given the name: *value\_bins* and num\_products\_SUM selected. The new variable to create is named to *loyalty\_bins* and the Binning type selected as Equal Height, previous settings are retained and 2 bins are created.

The binning code for the transformations applied is available from the Binning Code tab in either SAS or SQL format.

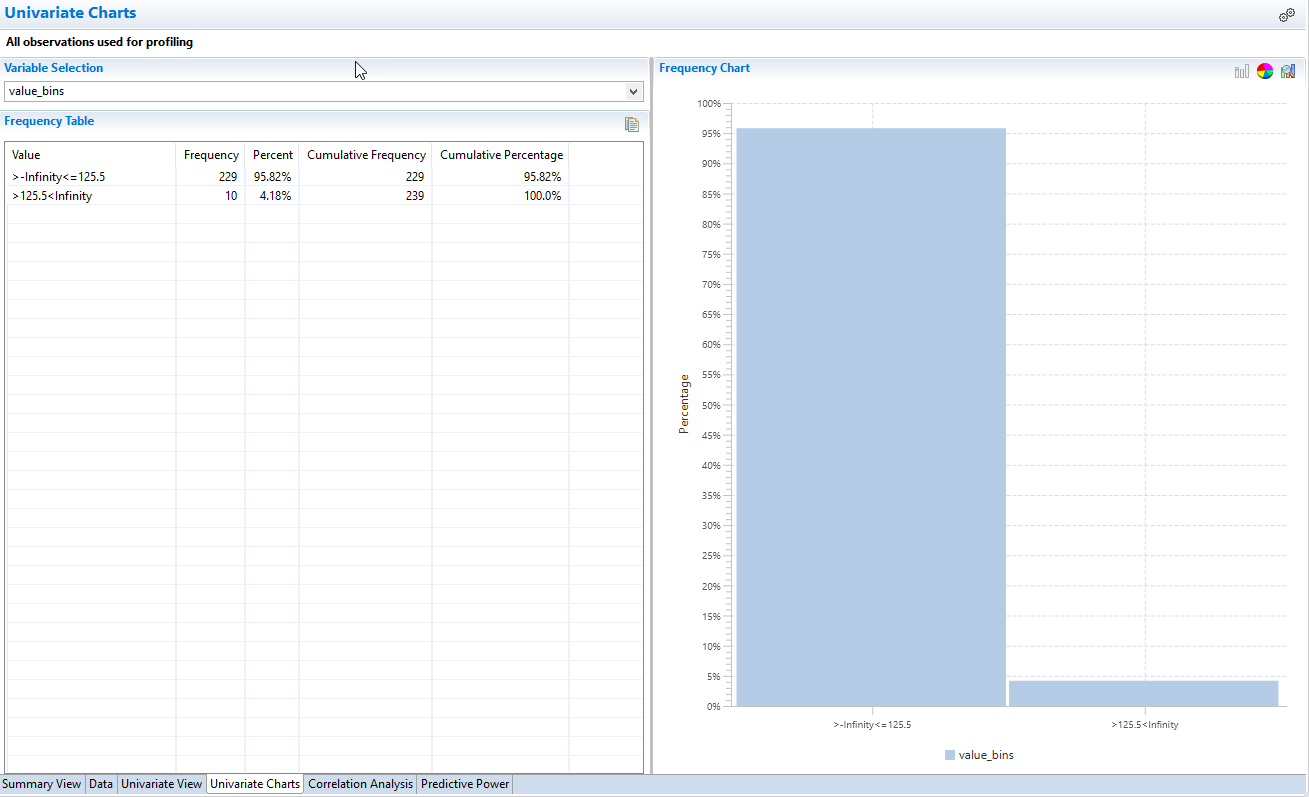
Figure 22: Binning Code



The keyboard shortcut CTRL+S is used to apply binning settings and the asterisk disappears reflecting this.

Returning to the Workflow and opening the created dataset with the Data Profiler, the variables *loyalty\_bins* and *value\_bins* have been added. Viewing bins for each variable using charts, it can be seen two bins are evident for each with an approximately equal number of observations in each bin.

Figure 23: Chart of value\_bins



Viewing the data it can be seen that the bins have cutpoints that are not appealing. To address the binning code can be copied and used in a SAS Code block and applied to the data directly. This way the bin labels can be changed to something more readily identifiable.

The binning code is copied by selecting and using the keyboard shortcut CTRL+C. A SAS code block is dragged onto the canvas and the dataset prior to the binning block attached.

Double clicking opens the SAS Code block. First an output data set is needed, and this is added by clicking the Add a new output dataset button In the Outputs area.

The binning code copied previously is pasted using CTRL+V. Now, the bin labels can be easily changed and here are updated to low and high for each variable.

The code must be wrapped in a DATA STEP. This requires the use of the command DATA, this creates a dataset, and the dataset to create is the one generated previously, this is dragged into the code and a semi-colon added.

Next, the data to process must be specified, this is done using a SET statement and the input dataset is dragged into the code and the code line completed with a semi-colon.

Finally, a RUN Statement is required and this is added at the very end with a semi-colon.

Figure 24: Completed code

A screenshot of a computer

Description automatically generated

Using CTRL+S applies the binning and the output dataset is created. Accessing the new dataset via the Data Profiler and viewing charts for the created variables shows labels have been applied and make the variable categories much easier to understand.

Figure 25: Labels applied

A screenshot of a computer screen

Description automatically generated

The binning can also be applied to other data by generating a binning model - this is accessed by right clicking the binning block.

Clicking the option configure outputs opens the binning block outputs dialog. By default a new dataset is generated but this can be suppressed by moving it to the Unselected Output list.

Moving the binning model from the Unselected Output list to the Selected Output list and clicking OK, generates the binning model block.

Figure 26: Outputting a binning model

A screenshot of a computer

Description automatically generated

The block cannot be viewed but now the binning can be applied to any appropriate dataset by connecting both the binning model and the dataset to bin to a Score block. To illustrate, the same dataset is connected and the output is automatically generated.

Figure 27: Scoring data with the binning model

A diagram of data processing

Description automatically generated

Viewing the data it can be seen that the new variables have been added.

## The Mutate block

The Mutate block can be used to apply the language of SAS or SQL statements to generate new variables. The Mutate block is dragged onto the canvas and the dataset created previously connected, and subsequently the Mutate block configuration dialog accessed.

Here there is interest in creating a new variable using those created earlier to distinguish valuable and loyal customers.

A new variable can be added by clicking the Add new variable icon, once added the variable name is changed to *value\_loyalty.*

Figure 28: Mutate block

A screenshot of a computer

Description automatically generated

Options to define the variable are available from the Configure Variable dialog, which is accessed by clicking the gears icons.

Figure 29: Configure Variable dialog

A screenshot of a computer

Description automatically generated

Three pages are available. The first, Variable, enables the properties of the variable to be specified.

As the variable being created here will contain character values, the Type Character is appropriately set. Other options such as label, length, format and informat can be set but if not, defaults are applied.

The Grouping Variable Selection page allows selection of one or more grouping variables and the Output page provides options to determine how to output results including the option to Do not output this variable.

Expression can be applied in the Expression area. A list of dataset variables and functions with help are available to assist with building expressions. These can be clicked to automatically add them to an expression.

Here case statements are used to assign a value of gold, silver, bronze based on the values of the variable’s *loyalty\_bins* and *value\_bins*.

Figure 30: Applying an expression

A screenshot of a computer

Description automatically generated

The preview provides insight as to not only the results but whether the expression has been specified correctly and here, it had.

CTRL+S applies the expression and a new dataset is output. Viewing results with the Data Profiler it can be seen that the variable has been added and from Univariate Charts its distribution can be viewed.

Figure 31: Viewing value\_loyalty

A screenshot of a computer

Description automatically generated

The resulting Workflow can be augmented with additional detail, for example if there is interest in adding transactional information for March, simply add the March dataset to the Workflow, connect to the merge node and the Workflow automatically updates and includes the new data in processing. Notice the resulting File now has 355 observations, whereas previously there were 239.

Bear in mind that the new data should correspond in format and nature to that used to build the processing chain initially to ensure the Workflow runs error free.

The SAS language code for the entire Workflow can be generated and exported to a SAS language program via canvas right-click. The code can be copied and used directly in the SAS Language or Workflow perspectives or exported and accessed as necessary.

Here the code is output to the Project folder as *data\_prep.sas*, refreshing the project shows the file and it can be accessed by double-clicking.

## Summary

 This lesson focused on data preparation and manipulation and Altair Analytics Workbench Workflow perspective capabilities.

A scenario was introduced followed by a demonstration using blocks from the data preparation group to

illustrate steps in the scenario. Once complete additional data preparation blocks were demonstrated using the results.