14

CASE STUDY

OVERVIEW

This chapter will walk you through a case study that utilizes many of the techniques and methods you have learned and pulls them together in a complete Tableau workbook. You will use a complex data source and transform the source data into a series of worksheets with advanced charts and interactivity. This chapter will conclude with the design of a polished dashboard that will enable detailed analysis by the user.

By the end of this chapter, you will be able to build a Tableau workbook and dashboard using complex data, enhanced with your own calculated dimensions and measures. You will also know how to effectively organize your data and design compelling charts to populate a highly interactive dashboard.

INTRODUCTION

Tableau workbooks and dashboards can provide powerful functionality and insights for end users, but only when they have been carefully constructed. Due to poor design and/or lack of focus, a lot of Tableau dashboards are hardly ever used. Many other dashboards are used frequently yet make it difficult for the end user to get the answers they seek. In most cases, these flaws are related to the dashboard designer failing to understand basic design principles or the full range of Tableau's capabilities.

In this chapter, you will work through a case study using a complex dataset and perform a series of steps to design a compelling dashboard. You will tap into many of Tableau's powerful capabilities in this design process and will design using data visualization best practices. The result will be a dashboard that is attractive, intuitive, and highly interactive, allowing users to derive insights quickly and easily.

The final dashboard will look like this:

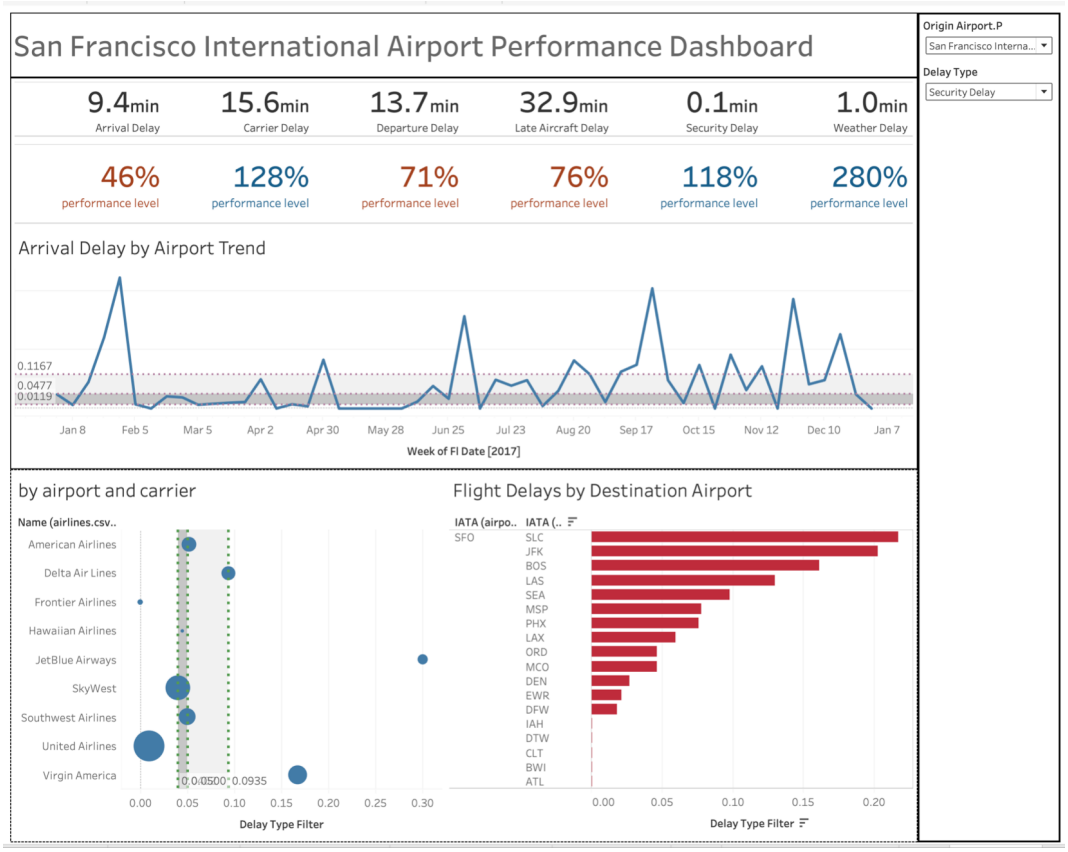

Figure 14.1: Final view of the dashboard

To arrive at the final dashboard, you will need to work through several steps, as detailed in the next section.

PERFORMING TABLEAU ANALYSIS

Many steps are involved in creating an effective Tableau workbook and dashboard for users to gain insight into their data and to take actionable steps based on these insights. This journey always begins with data and concludes with a workbook or dashboard delivered to end users. Along the way, there are many intermediate steps as you prepare the data and fashion it into something insightful for the user. Here is a list of steps in the process you'll perform in this chapter:

- Acquiring the source data
- Setting up the Tableau data source
- Building an extract
- Inspecting and validating the data
- Enhancing the data using aliases, calculated fields, and groups
- Building interactivity through parameters and filters
- Adding worksheets
- Creating the right chart types for data display
- Analyzing the data
- Designing a dashboard
- Adding interactivity to the dashboard
- Testing the dashboard
- Making final edits

As you can see, there are many steps involved in designing a great solution. However, the payoff will be worth the effort, as users return to a workbook or dashboard frequently to find answers to their questions and to discover new patterns in the data. You will work through each of these steps using your selected dataset for this process.

DOWNLOADING THE SOURCE DATA

Your source data is focused on airline flight delays in the United States in 2017. You will also use related data that will provide airline and airport details, including full names for each entity. Together, these data sources will provide you with a rich, highly detailed dataset to facilitate some interesting analyses.

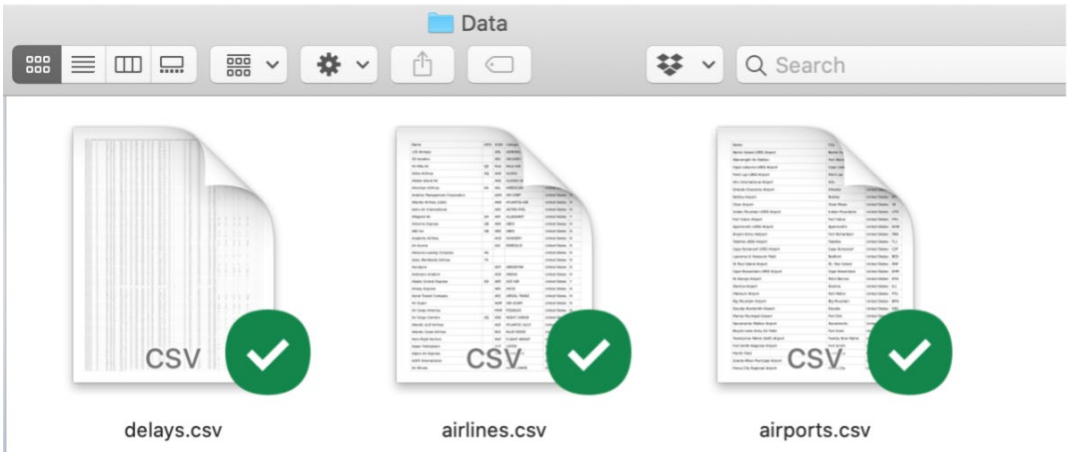

Figure 14.2: Data files in a folder

Below are short descriptions of your data sources:

- **airports.csv codes**: This .csv file contains all United States airport names, cities, the country/region, and International Air Transport Association (IATA) codes. It is important to know that each airport has a unique IATA code, which you will use to join other data sources in the chapter.
- **airlines.csv codes**: This file contains a list of all United States airlines, with their names, IATA codes, ICAO codes, callsigns, countries, and states where they are active. You are going to be using this file to help identify which airlines are the best/worst performing from a particular airport by joining it to the delays data later. This can be found at the following address: https://www.kaggle.com/open-flights/airline-database.
- **delays.csv**: This file contains row-by-row data of flight origins, destination airports, and any kind of delays including arrival delays, security delays, weather delays, and so on. You will use `op_carrier` as well as origin/dest IATA airport codes to join the preceding two `airports.csv`/`airlines.csv` files to create your dashboard. This file can be found at the following address: https://www.kaggle.com/yuanyuwendymu/airline-delay-and-cancellation-data-2009-2018.

> **NOTE**
>
> **`Delays.csv`** file is a .zip file which contains airline delay data of January and February 2017 (instead of the data of entire year 2017, to adhere to Github's uploading policy of 25mbs). The exercise/activities in this chapter were created with data of entire year 2017, so your KPIs/Screenshots might differ than the screenshots placed in the chapter. If you decide to download the data from Kaggle links mentioned above, the data file or column names possibly might have changed (as some data alterations were made for the easier flow-through of the exercises).It is highly recommend that you download the data from the GitHub repository: http://packt.link/ze8rT.

SETTING UP THE DATA SOURCE

Everything starts with the data when you build a Tableau solution. The data may come from Excel, internal databases (Oracle, Hadoop, SQL Server, and so on), or external sources. Regardless of the data source origin, you will need to set up a data source for Tableau to work from. In some cases, this will be as simple as copying and pasting existing SQL code and having Tableau run a daily refresh. Other cases will require joining multiple data sources, and perhaps even creating unions within the data source when needed.

In this case study, you will use joins to merge multiple source files that have different levels of granularity as well as file sizes. This will provide insight into structuring a complex data source; your own data sources may be simpler than this example.

ADDING THE DATA SOURCE

We are using a rather complex set of data sources in this chapter as it allows you to enhance your content beyond what a single primary data source would have permitted. Using these multiple sources will not only aid you in the context of this chapter but will permit further exploration and analysis not included here.

Each of these source files is unique – you have the **`delays.csv`** data, along with the **`airline.csv`** and **`airport.csv`** source files, for a total of three files. The annual files contain the measures you will analyze, along with some basic dimensions including flight dates, origin and destination airports, and the airline carrier. You will use the airline and airport files as reference (lookup) sources, enabling you to use more detailed title and location information.

CREATING JOINS IN THE DATA SOURCE

Start adding the data source into Tableau as part of your data preparation, where you add the airline and airport source files. **`airline.csv`**/**`airport.csv`** are joined to the **`delays.csv`** source based on matching fields.

DELAYS.CSV TO AIRLINES.CSV JOIN:

You will first make a duplicate version of both **`airlines.csv`**/**`airports.csv`** data by right-clicking on each and selecting **`Duplicate`**. You do this so that you can have separate reference tables for origin and destination airports. This is perhaps less critical for the airline source, although there are some likely use cases we won't explore in this chapter.

The airlines.csv to **`delays.csv`** table join will be joined using **`OP_CARRIER`** (**`delays.csv`**) to IATA (**`airlines.csv`**). Note that we are using a left join from **`delays.csv`** to each airline source; this will allow Tableau to have every record from the **`delays.csv`** table, even when there is no match in the airline data.

DELAYS.CSV TO AIRPORTS.CSV JOIN:

You'll follow a similar process with the **`airports.csv`** data, with one important difference. The first airport table will join on the IATA value, connecting to the Dest field in **`delays.csv`**. This will provide airport reference data for all destination airports in your analysis. The second airport table will again use the **`IATA`** value but will join on the **`Origin`** field in the **`delays.csv`** data. You will now have a separate reference for all origin airports. This allows you to build analyses with both the destination and origin airport locations.

EXERCISE 14.01: CONNECTING AND EXTRACTING DATA SOURCES

As part of the Tableau development team, you are tasked with using the instructions listed below to connect, join, and load the data files into Tableau.

Perform the following steps to complete this exercise:

Data GitHub link: http://packt.link/ze8rT.

1. Download the data source files from the GitHub repository listed above. Save these files into an easily accessible folder (perhaps the **`My Tableau Repository > Data Sources`** directory). If you downloaded all the files as **`.zip`** files, unzip the files if you haven't already done so.

2. Open Tableau and select the **`Data`** > **`New Data Source`** menu item.

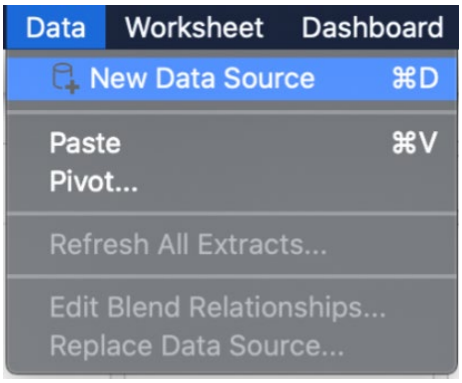

Figure 14.3: Adding a new data source

3. Select the **`Text file`** option since all your data files are **`.csv`** files.

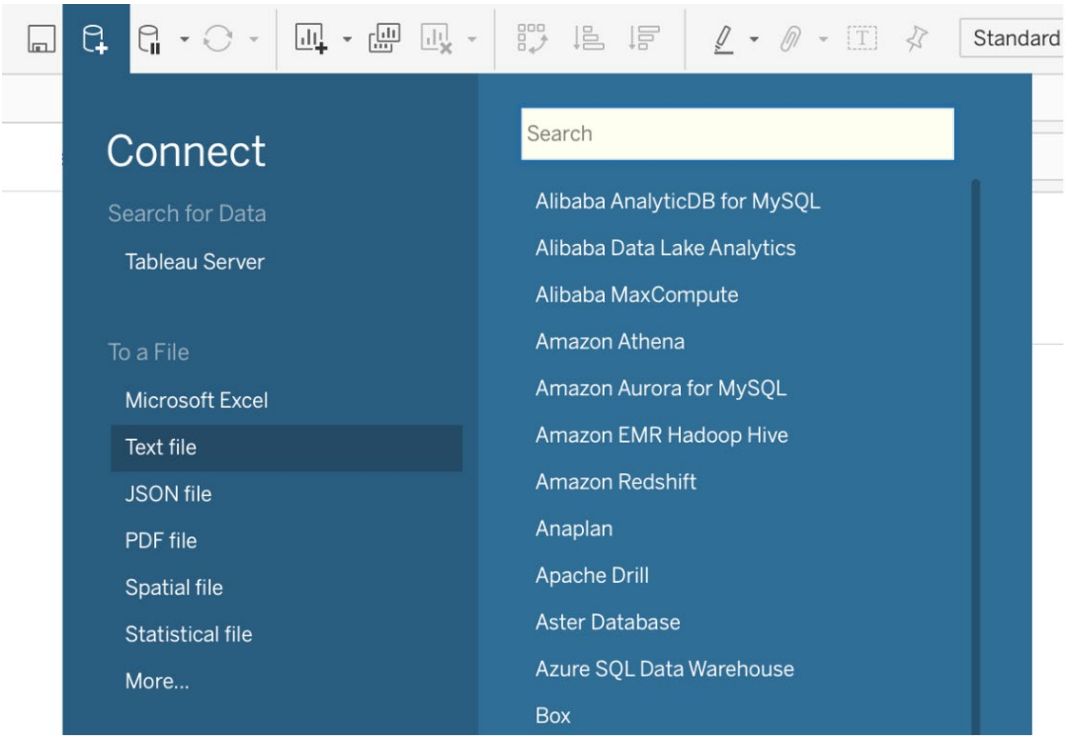

Figure 14.4: Adding a text file as a data source

4. Add the **`delays.csv`** and **`airlines.csv`** files to the data source pane.

5. Left join the **`delays.csv`** file with **`airlines.csv`** on **`[Op Carrier] = [Iata]`**, as shown below:

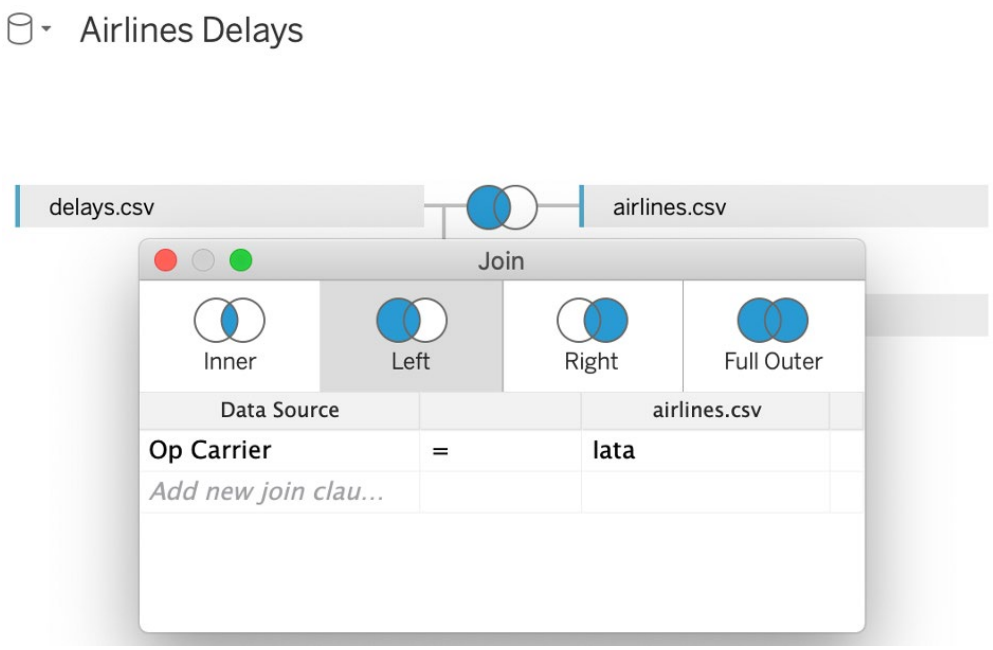

Figure 14.5: Joins in Tableau

6. Duplicate **`airlines.csv`** and rename it to **`airlines2.csv`**, and left join with **`delays.csv`** on **`[Op Carrier] = [Iata]`** as shown in as shown below:

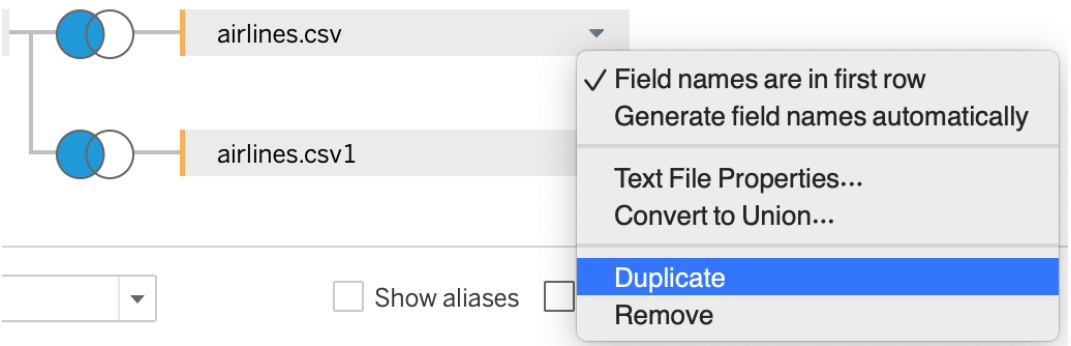

Figure 14.6: Duplicating data files in Tableau

7. Add the **`airports.csv`** file and left join **`delays.csv`** to airports.csv on **`[Origin] = [IATA]`** as shown below

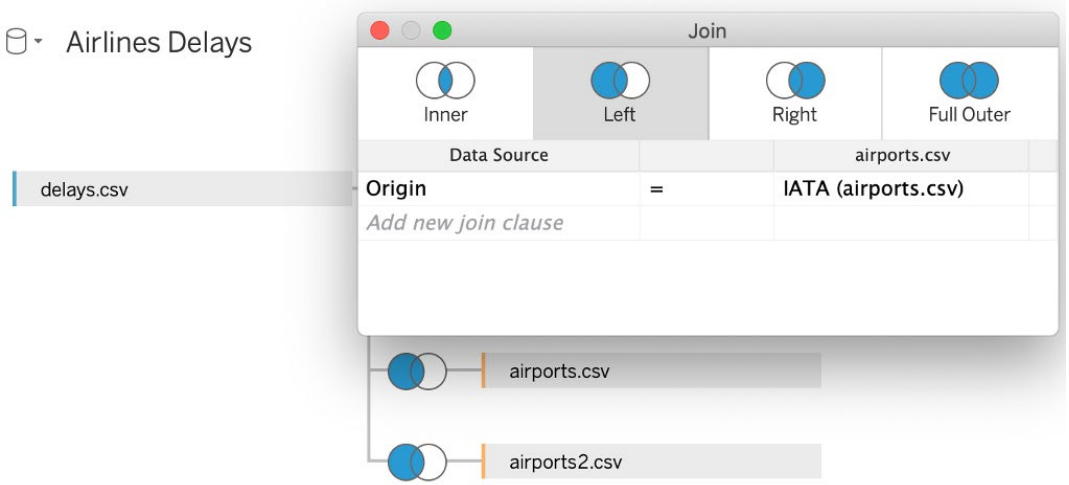

Figure 14.7: Left joins in Tableau

8. Duplicate and rename the file to **`airports2.csv`**, and left join **`delays.csv`** to **`airports.csv`** on **`[Origin] = [IATA]`** as shown in *Step 7*

   The final data files and joins should look like the following screenshot:

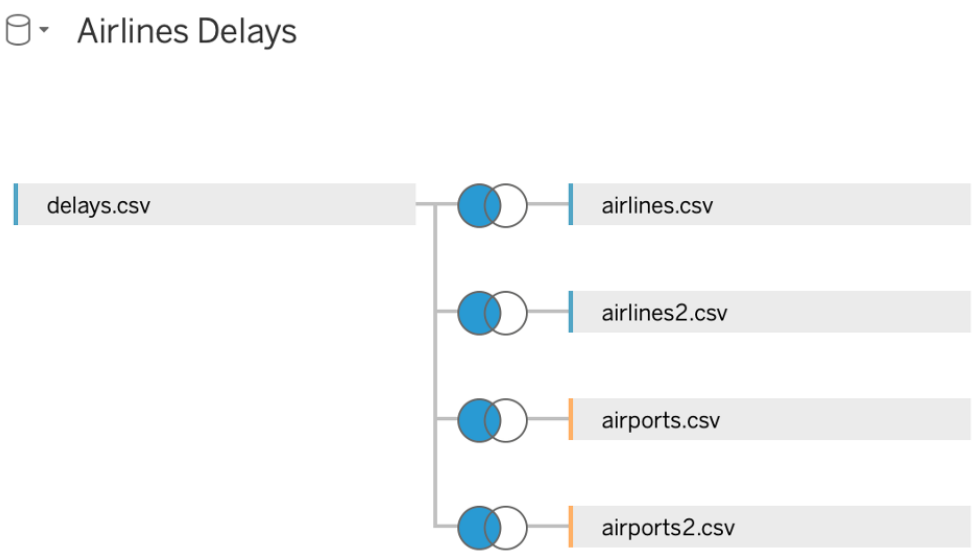

Figure 14.8: All the joins for the dashboard

9. Next, to optimize your workflow, change your data connection from **`Live`** to **`Extract`**. When you create an extract, Tableau combines multiple sources into a single file, which will improve the user response and make it easier to load the Tableau file the next time you open the file. To change the connection from **`Live`** to **`Extract`**, click on **`Extract`** under **`Connection`** as shown.

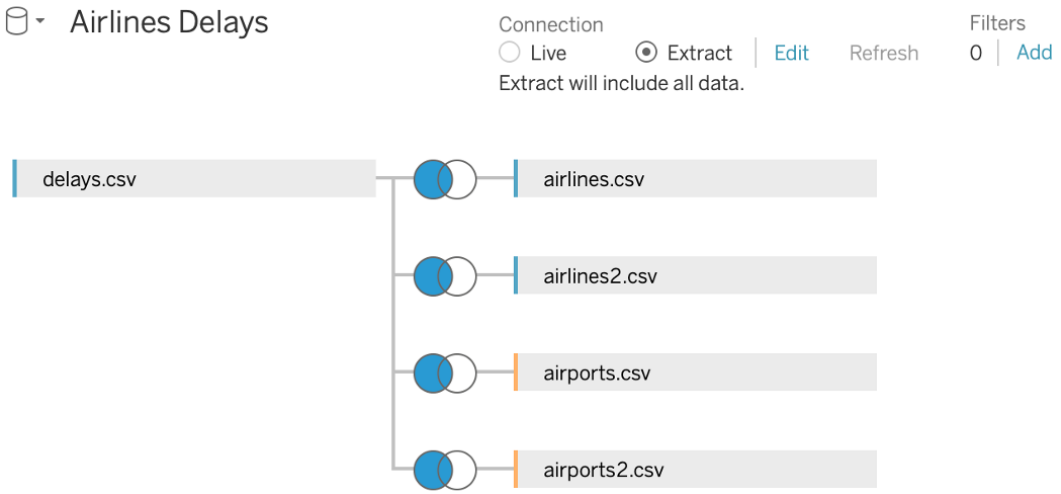

Figure 14.9: Creating an extract

10. To save your extract file, click on one of the sheets. The moment you do that, Tableau will start creating extracts by importing the data into Tableau's data engine:

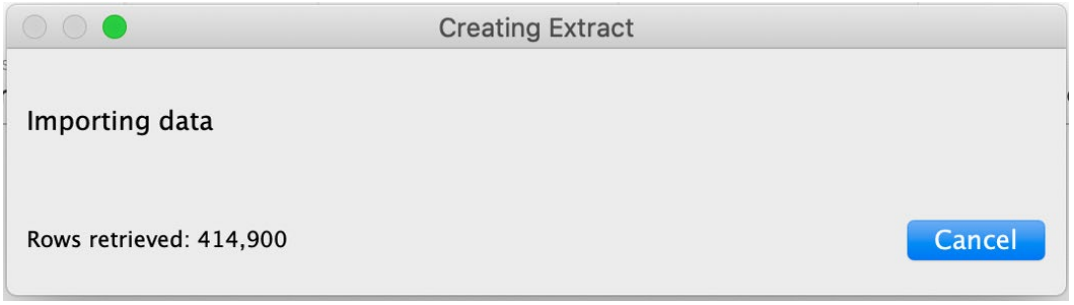

Figure 14.10: Tableau importing data for extract

Once Tableau finishes creating the extract, it will ask you to save the extract locally. Save the file locally as required. This concludes the *Connecting and Extracting Data Sources* exercise.

You have now created all the joins needed to build worksheets using complete terminology to provide detailed charts and tooltips that are easy for your user to decode. In the upcoming section, you will work through the process of validating the data that was loaded and take steps to clean/rename fields, accordingly.

DATA VALIDATION

One of the first steps after loading the data into Tableau is to validate the dimensions and measures. At this stage, you are not looking for true data analysis; that comes later. What you should be doing is validating field names and values as you try to make sense of the data. One way to do this with dimensions is to simply click the **Aliases** menu option and look at the values for each dimension. Ideally, you would see values that make sense within the data context. If, however, there is a lot of nonsense or missing values, you will need to spend some additional time on data preparation.

Another step you should take is to verify that Tableau correctly identified your data fields. In many cases, a field with only numeric values will land in Tableau as a measure. Often, these fields will be identifying codes rather than measures, so you will want to make sure to move such fields to the **Dimensions** space in the **Data** tab. Similarly, some measures will be incorrectly identified as dimensions, especially if they have null values present. In this case, you will want to move the field to the **Measures** space on the **Data** tab.

Let's look at the newly loaded data to see where there may be issues that need to be addressed.

DIMENSIONS

After your data source is loaded and the extract has been run, you will have the opportunity to examine the **Dimensions** and **Measures** sections of the **Data** tab. This is where you can correct any misclassified fields, as well as renaming fields to more intuitive values. Your dimensions will be organized based on the data source they are associated with; this is the default setting. This can easily be switched to a folder view—something that makes sense in many instances. Here, though, we will stick with the default view.

Recall from the data source structure that you repeated with both the airlines and airport files, using one for the origin information and the other for destination details. You can verify the data source by selecting the two airline files:

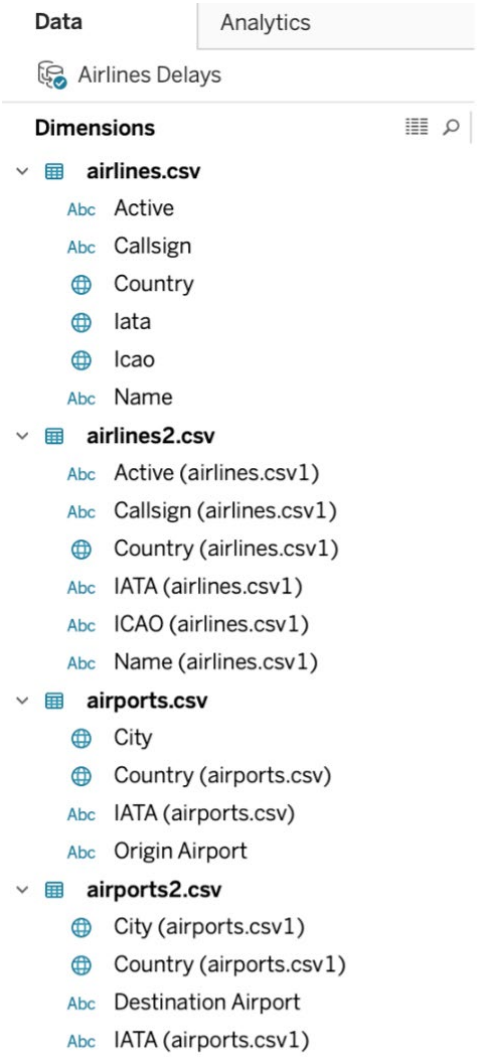

Figure 14.11: All dimensions and measures of the Airlines Delays data

Note how Tableau differentiates the field names by referring to the specific data source.

Your airport data is handled in the same fashion. If you noticed in the preceding screenshot, the **[Origin]** and **[Dest]** column has been renamed to **[Origin Airport]** in **airports.csv** and **[Destination Airport]** in airports2.csv. You can take the additional step of renaming your airport field to reflect the destination and origin values:

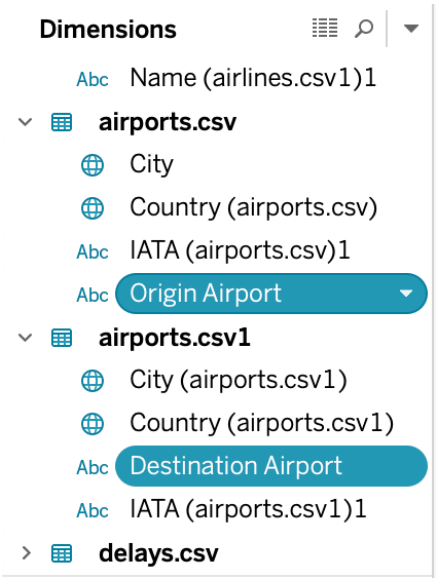

Figure 14.12: Origin and destination airport

Your **delays.csv** contains the primary data you will be using to build worksheets and the dashboard. This is where nearly all your measures and most of your dimensions are contained. The dimensions you can build from include the following:

- **CANCELLATION_CODE**
- **DEST** (for destination airport)
- **FL_DATE** (for flight date)
- **OP_CARRIER** (airline abbreviations)
- **ORIGIN** (for origin airport)

While the number of dimensions is limited, these fields will enable you to build analytics using flights between airports by date and by carrier. This gives you an opportunity for detailed analytics across these dimensions, especially when you see the many available measures included in the data.

MEASURES

Now, have a look at how Tableau defined your measures:

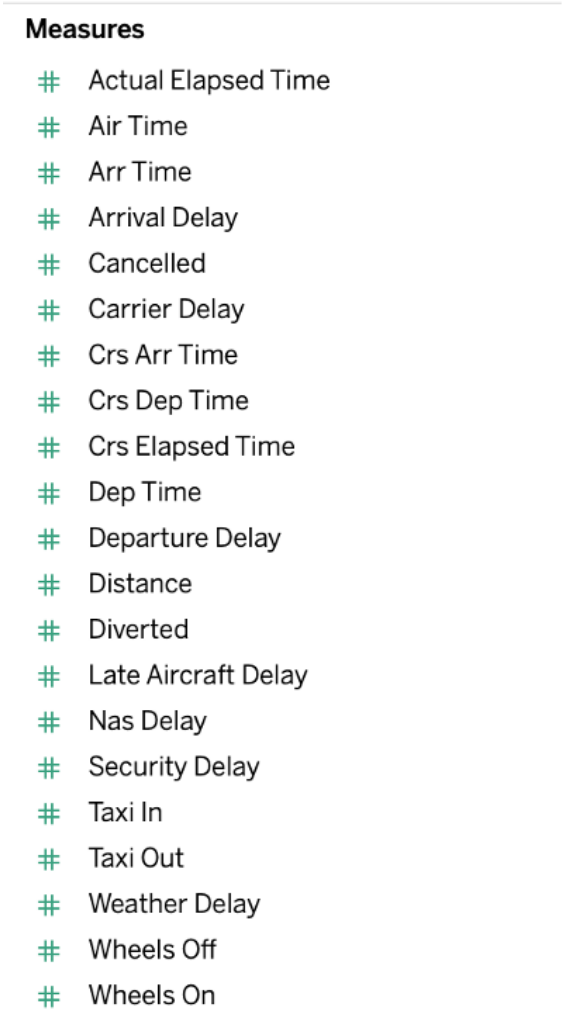

Figure 14.13: Measures in Airlines Delays data

Note that your **`delays.csv`** data source contains all your measures aside from the geographic coordinates contained in the two airport sources. These will be available to you in the event that you choose to place all airports on a map or to show traffic between airports using maps. Otherwise, everything you work on in this chapter will be based on measures from the **`delays.csv`** data source, with a focus on the multiple delay measures included here. You could just as easily have elected to focus on time or distance measures in building out additional worksheets and dashboards.

ENHANCING THE DATA

Even when your source data is highly detailed, odds are that you will need to create additional measures, aggregations, groups, or other calculations that make it easier to navigate a workbook. You can also choose to rename fields, as you did previously, so that they become more intuitive for the workbook user. Tableau also makes it easy to set default formats for your measures, making things easier for the eventual user. These formats can be updated as you walk through the next exercise step by step. With this in mind, let's check the data to see how you can improve on the existing data and add new fields to benefit your users.

ORGANIZING THE DATA

As noted earlier, in this case, the default views for dimensions and measures will work well. However, if you prefer to see the data organized by folders, this can be done easily. Simply right-click anywhere within the **`Data`** tab and select the **`Group by Folder`** menu option:

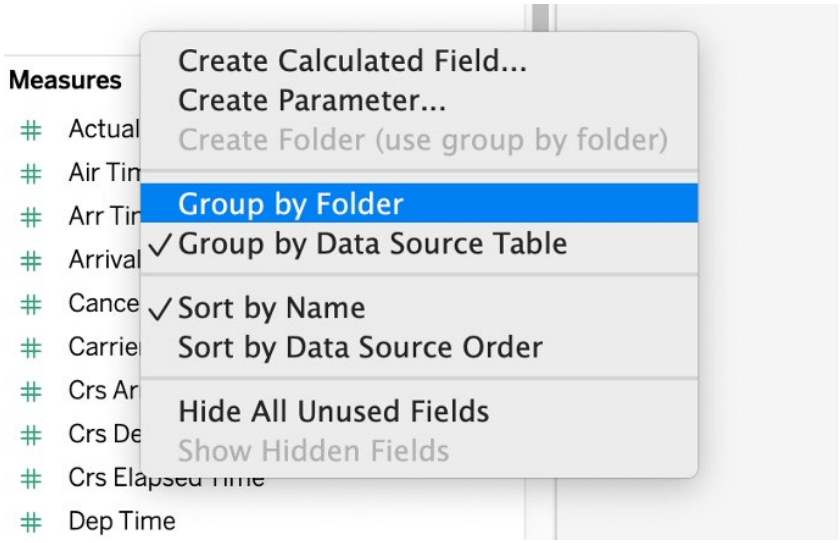

Figure 14.14: Grouping data by folder

This is often the best choice when using a single data source with many dimensions or measures, as it allows you to customize the organization of your data.

FORMATTING THE DATA

One of the great time-saving actions in Tableau is to set default formats for all data fields. This will make it much easier to create charts and crosstabs that display values in the appropriate format. You can still select a different format when needed, but setting default formats is worth the effort. In this section, you'll see how easy it is to set default formats for Tableau measures.

Generally, you would be less concerned with formatting dimensions in Tableau, although options do exist. But your main concerns here tend to revolve around geographic fields, and making sure Tableau recognizes them in the manner you are seeking. Fields with names like country, state, or province tend to be recognized at a very high rate as geographic variables, but you may have instances where the names are less than obvious, and you need to let Tableau know by applying a Geographic Role value. As with many Tableau options, a simple right-click will provide these menu options for you.

Measures are of greater concern—mainly so you don't need to continually format values for every worksheet where a measure is used. Setting these formats upfront will save valuable time later as your workbooks become more complex.

There are two primary types of formatting we will cover here. The first is the default aggregation for each measure, the second concerns the numeric format Tableau will use for a specific measure. Tableau tends to assume that the default aggregation for numeric fields should be a sum, when the reality in this case is that averages are far more useful. Certainly, there are cases where you might keep **`sum`** as the default, but when you are attempting to compare performance across different airports of varying sizes, **`sum`** tells you little to nothing about your data. You will now practice how to enhance and format data in the next exercise.

EXERCISE 14.02: ENHANCING AND FORMATTING THE DATA

In this exercise, you will continue your earlier data exploration and validate a couple of data fields as well as changing aggregation levels for some of your measures.

Perform the following steps to complete this exercise:

1. Ensure that the **`City`** and **`Country`** fields are correctly aligned by right-clicking and selecting **`Geographic Role`** from the menu across from your dataset, as shown below:

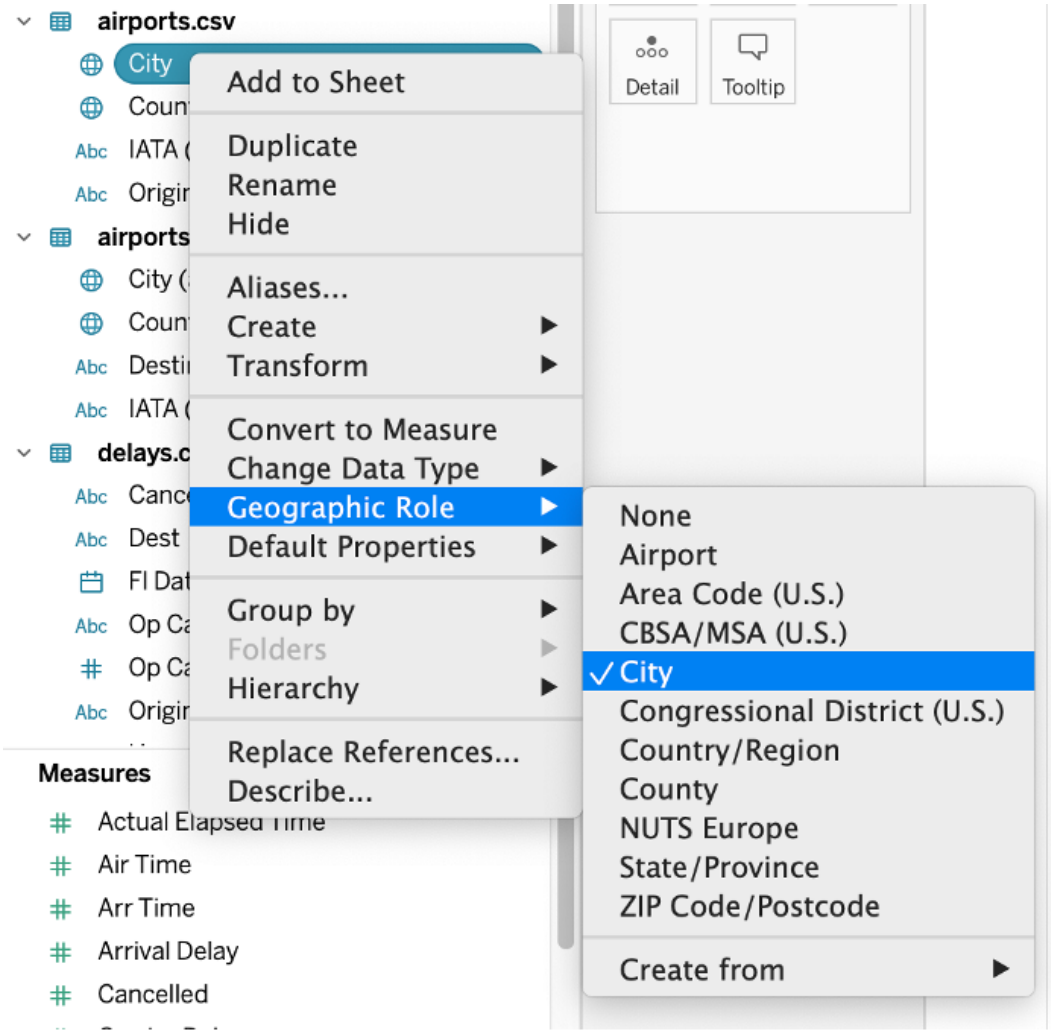

Figure 14.15: Changing Geographic Role for a dimension

2. Search for **`delay`** using the search icon under the **`Data`** pane. Then, change the default aggregation for all your **`delay`** measures to be **`Average`** instead of the default **`Sum`** aggregation by right-clicking and selecting **`Default Properties`** > **`Aggregation`** > **`Average`** as shown. We want to show average delays across **`[Arrival Delay]`**, **`[Carrier Delay]`**, **`[Departure Delay]`**, **`[Late Aircraft Delay]`**, **`[Security Delay]`**, and **`[Weather Delay]`**. We do not want the sum of all delays because this means adding all the delays together and presenting the result as an aggregated number, which is not the goal here. The goal is to show average delays across categories for each carrier/airport.

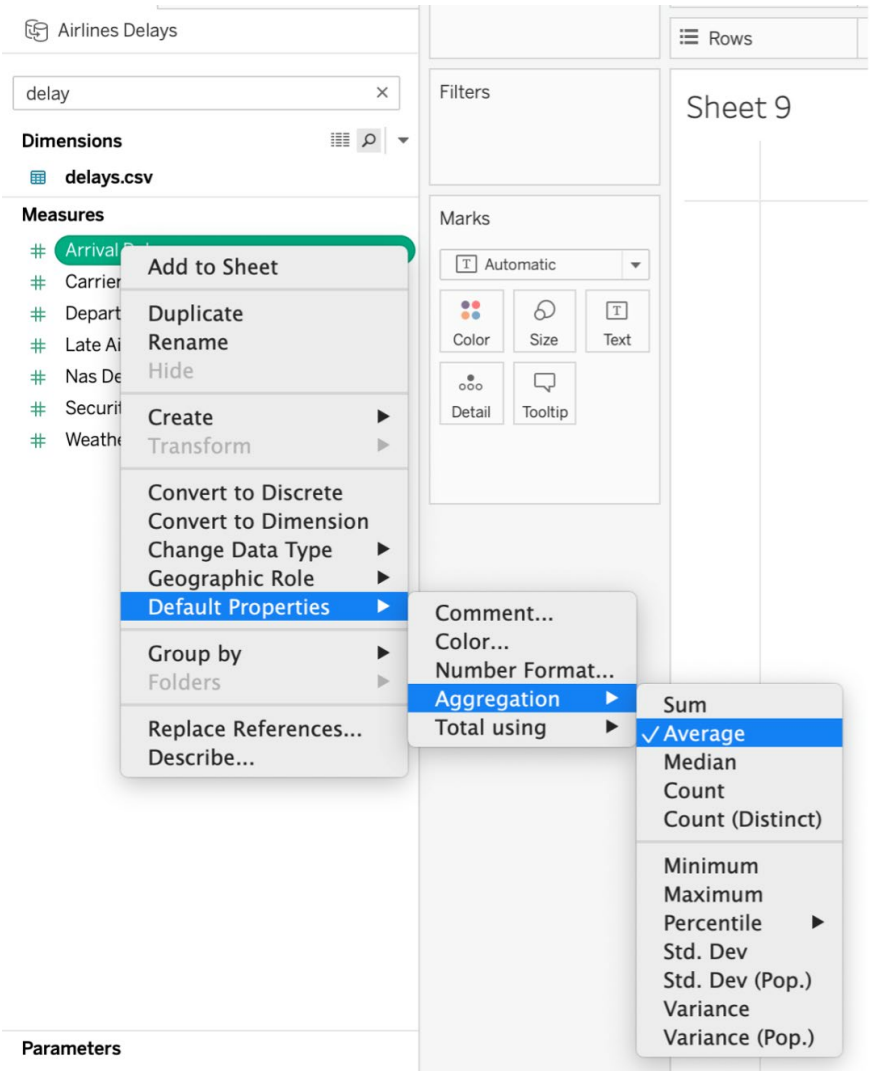

Figure 14.16: Changing the aggregation level of a measure

3. Set all your **`delay`** fields to **`Number (decimal)`** format. Search for **`delay`** as in the previous step, then right-click and select **`Change Data Type`**, and select **`Number (decimal)`** as shown below:

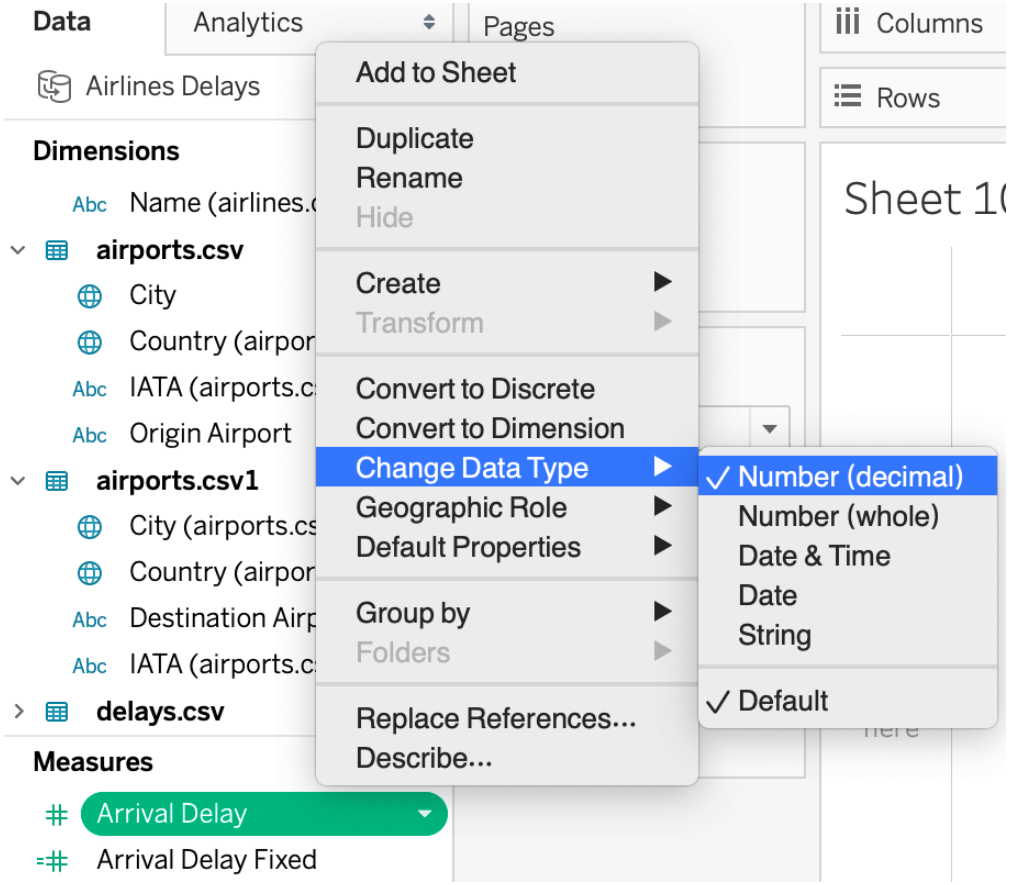

Figure 14.17: Changing the data type of a measure

4. Finally, format your "delay" measures to be one decimal. To accomplish this step, right-click on any of the delay measures and click on **`Default Properties`** > **`Number Format`** as shown below:

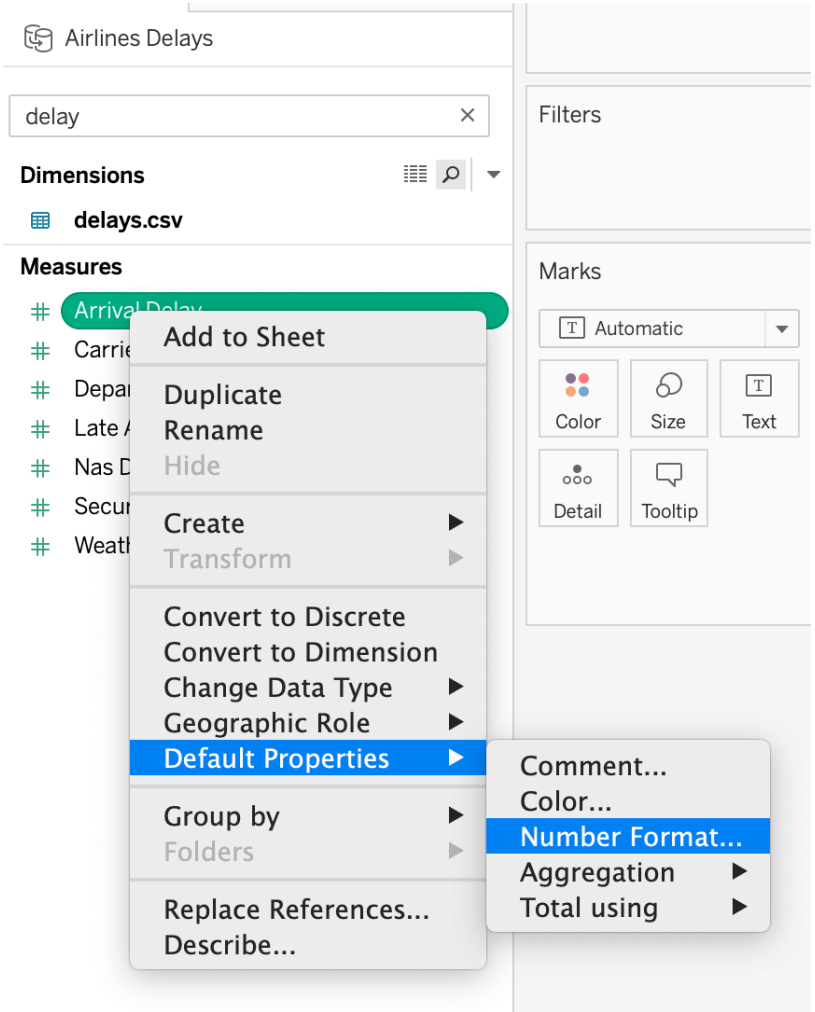

Figure 14.18: Formatting numbers

5. In the number format popup, select **`Number (Custom)`** and set the decimal places to **`1`**. You can keep the rest as is:

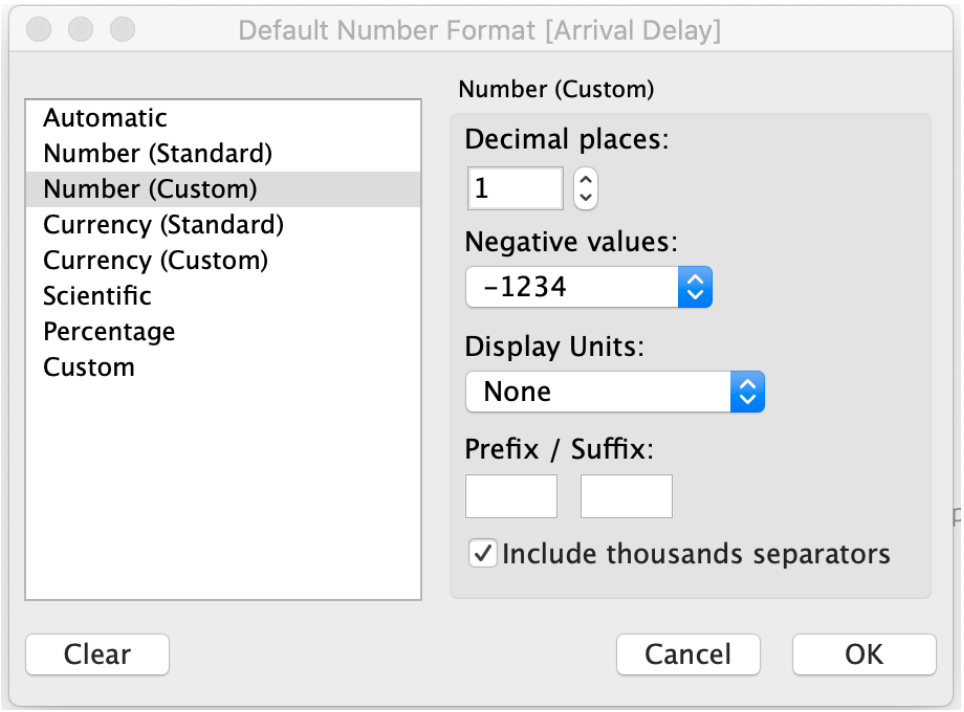

Figure 14.19: Changing the data type of a measure

In this exercise, you learned how to validate data fields, and the updated basic aggregation and formatting options. These steps will make subsequent worksheets and dashboards much easier to work with as you move into more complex scenarios.

BUILDING CALCULATED MEASURES

Few datasets contain all the measures you would like, for a variety of reasons. This is generally a good thing, however, as many measures that are pre-calculated wind up limiting your ability to work with the data in a more flexible manner. In an ideal world, the measures you load into Tableau will primarily be counts or simple measures rather than complex calculations. This allows you, as a dashboard designer/developer, to develop an array of calculations to help the user analyze information.

In this section, we will walk through multiple calculations that will make your dashboard more useful for the eventual user. The data source does provide some simple numbers detailing several types of flight delays. From these, you can build additional measures using some Tableau calculations. You will work with six fields from the source data:

- Arrival Delay (**ARR_DELAY**)
- Carrier Delay (**CARRIER_DELAY**)
- Departure Delay (**DEP_DELAY**)
- Late Aircraft Delay (**LATE_AIRCRAFT_DELAY**)
- Security Delay (**SECURITY_DELAY**)
- Weather Delay (**WEATHER_DELAY**)

You have already changed the default aggregation for all the **delay** fields from **SUM** to **AVERAGE** in the previous exercise. One of the goals in this dashboard is to compare a single airport to all airports so you can assess performance across each of the six measures just discussed. To accomplish this, you will use two steps: the first is to create a **FIXED** calculation based on each measure to set the overall average for each type of delay. Once this has been done, the second step will be to create a calculation where the individual airport is compared to the overall average.

FIXED CALCULATION:

Let's start with the **FIXED** calculation. The **FIXED** command can be used to provide what Tableau calls Level of Detail (LOD) functions, which we covered comprehensively in *Chapter 9, Analysis: Creating and Using Level of Detail (LOD) Calculations*, where numbers can be compared across different levels in a hierarchy. This is both powerful and useful as you build worksheets and dashboards. The following is an example of a fixed calculated measure for the **Arrival Delay** measure that you will build in *Exercise 14.03*:

Sample Arrival Delay Fixed:

Figure 14.20: FIXED LOD Arrival Delay formula

Notice that this example uses the **`AVG`** function to calculate the average value for **`Arrival Delay`**. If you used only that calculation, your value would change each time you filter for a new airport. However, when you use the **`FIXED LOD`** calculation, Tableau ignores filters and continues to calculate the **`Arrival Delay`** value for the entire dataset because you are explicitly asking Tableau to calculate across the dataset and ignore any filters.

Fixed calculations can also be executed at lower levels of detail; for example, you could add the flight date to the formula and see arrival delays averaged for each date in your dataset.

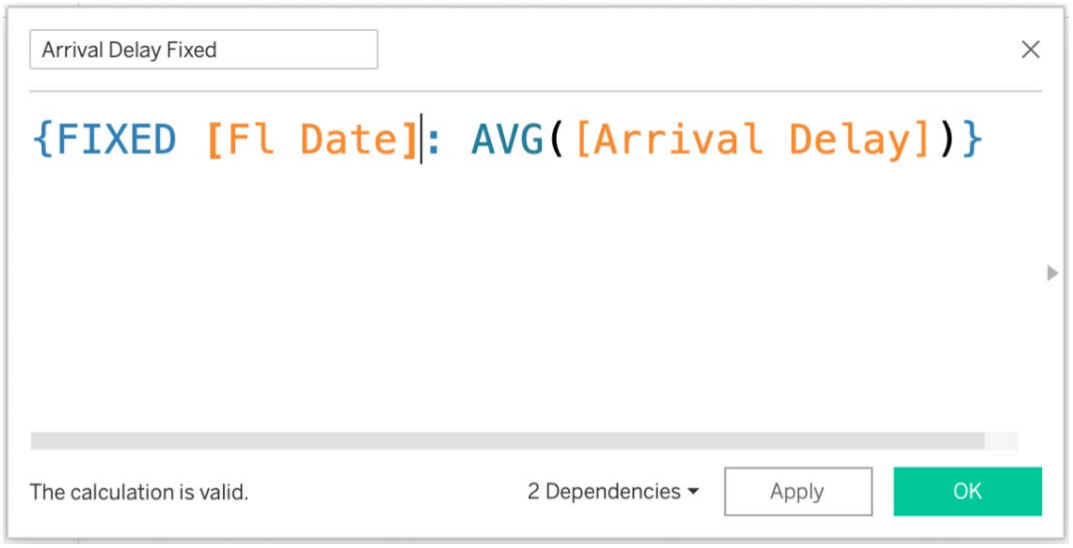

Figure 14.21: FIXED LOD Arrival Delay formula with dates

DELAY PERFORMANCE CALCULATED FIELD:

In the exercise below, you will build another set of six calculated fields to calculate the delay performance, after which you will add new calculated fields to try to calculate the delay performance level of a particular airport compared to the average delay performance across all the airports in the USA.

Sample Create Arrival Delay Performance Calculated Field

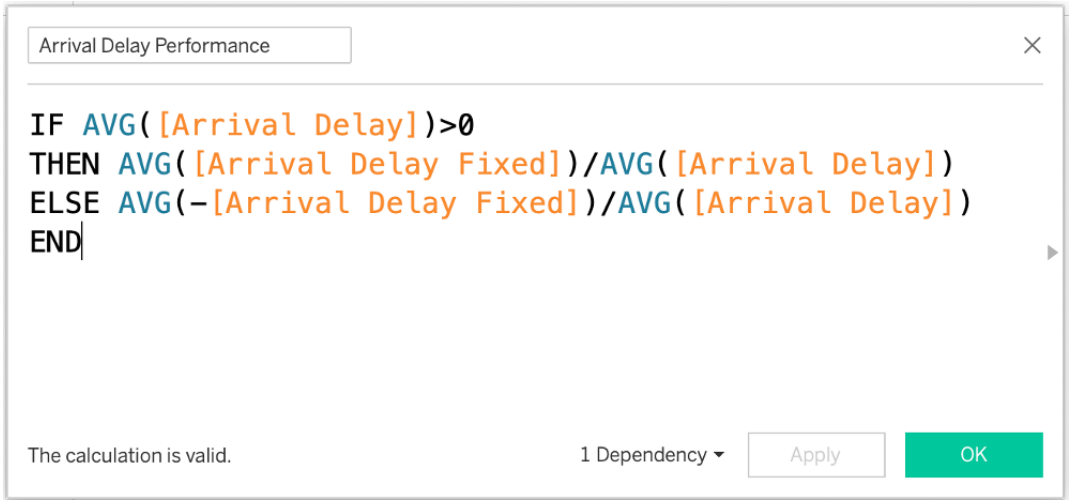

Figure 14.22: Arrival Delay Performance calculated field

Note that this example used an **`IF`** statement for arrival delays. Since these delays are sometimes negative average values, you need to account for those cases in which flights arrive early. For your remaining performance calculations, this is not necessary. You simply calculate the total versus the delay for an individual airport (or airline), as shown here:

Sample Carrier Delay Performance

Figure 14.23: Carrier Delay performance

You will build these calculated fields in your Tableau worksheet one by one in the following exercise.

EXERCISE 14.03: CREATING CALCULATED FIELDS

Continuing from our previous exercise, in this one, you will create multiple calculated fields to compare individual airports (or airlines) to overall averages using the Tableau **`FIXED`** calculation.

Perform the following steps to complete this exercise:

1. Rename the six measures containing your delay data if you have not already done so. The six measures should be named **`Arrival Delay`**, **`Carrier Delay`**, **`Departure Delay`**, **`Late Aircraft Delay`**, **`Security Delay`**, and **`Weather Delay`**.
2. Create **`FIXED`** calculated fields: **`Arrival Delay`** Fixed. As discussed in the section above, create six different fixed calculated fields for all your arrival dimensions to calculate the **`AVG`** delay across these six dimensions for the whole dataset.

Figure 14.24: Arrival Delay Fixed LOD formula

3. After saving the [Arrival Delay Fixed] calculated field, repeat the same step for the other five different delays with the following formulas:

- Arrival Delay Fixed: **`FIXED:AVG([Arrival Delay])}`**
- Carrier Delay Fixed: **`FIXED:AVG([Carrier Delay])}`**
- Departure Delay Fixed: **`FIXED:AVG([Departure Delay])}`**
- Late Aircraft Delay Fixed: **`FIXED:AVG([Late Aircraft Delay])}`**
- Security Delay Fixed: **`FIXED:AVG([Security Delay])}`**
- Weather Delay Fixed: **`FIXED:AVG([Weather Delay])}`**

  Once you have created the above calculated fields, you should have the following **`delay`** fields in your **`Data`** pane:

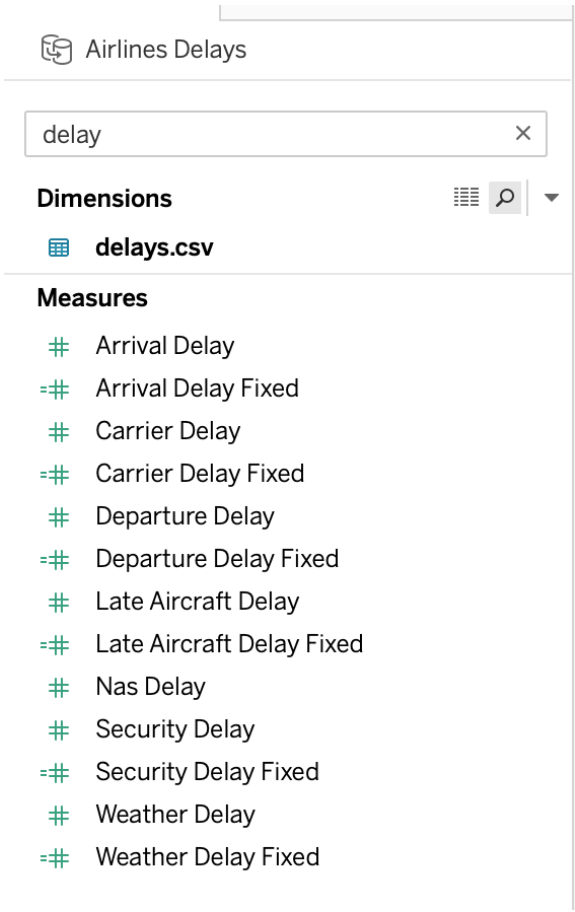

Figure 14.25: All delay fields

4. Next, build another set of six calculated fields to calculate the delay performance of individual airports (or airlines) and compare them across the dataset.

   Create Arrival Delay Performance Calculated Field:

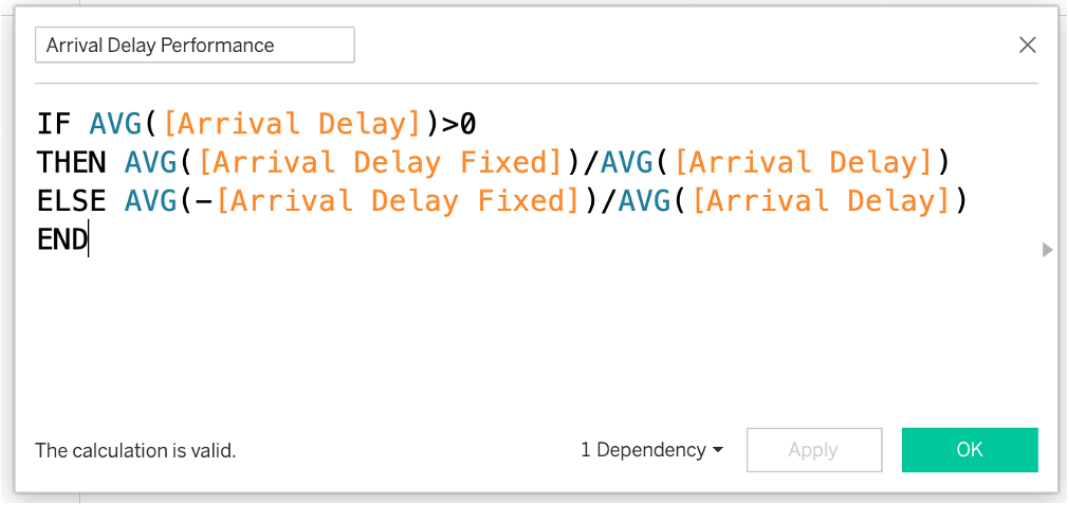

Figure 14.26: Arrival Delay Performance field

Formulas:

- Arrival Delay Performance:

```
IF AVG([Arrival Delay])>0
THEN AVG([Arrival Delay Fixed])/AVG([Arrival Delay])
ELSE AVG(-[Arrival Delay Fixed])/AVG([Arrival Delay])
END
```

- Carrier Delay Performance:

```
AVG([Carrier Delay Fixed])/AVG([Carrier Delay])
```

- Departure Delay Performance:

```
AVG([Departure Delay Fixed])/AVG([Departure Delay])
```

- Late Aircraft Delay Performance:

```
AVG([Late Aircraft Delay Fixed])/AVG([Late Aircraft Delay])
```

- Security Delay Performance:

```
AVG([Security Delay Fixed])/AVG([Security Delay])
```

- Carrier Delay Performance:

```
AVG([Weather Delay Fixed])/AVG([Weather Delay])
```

5. Set the default number format to a percentage with 0 decimal places for each of the six performance measures. Right-click on one of the performance measures as follows:

   `[Carrier Delay Performance]` > **`Default Properties`** > **`Number Format`** and make changes appropriately or as shown:

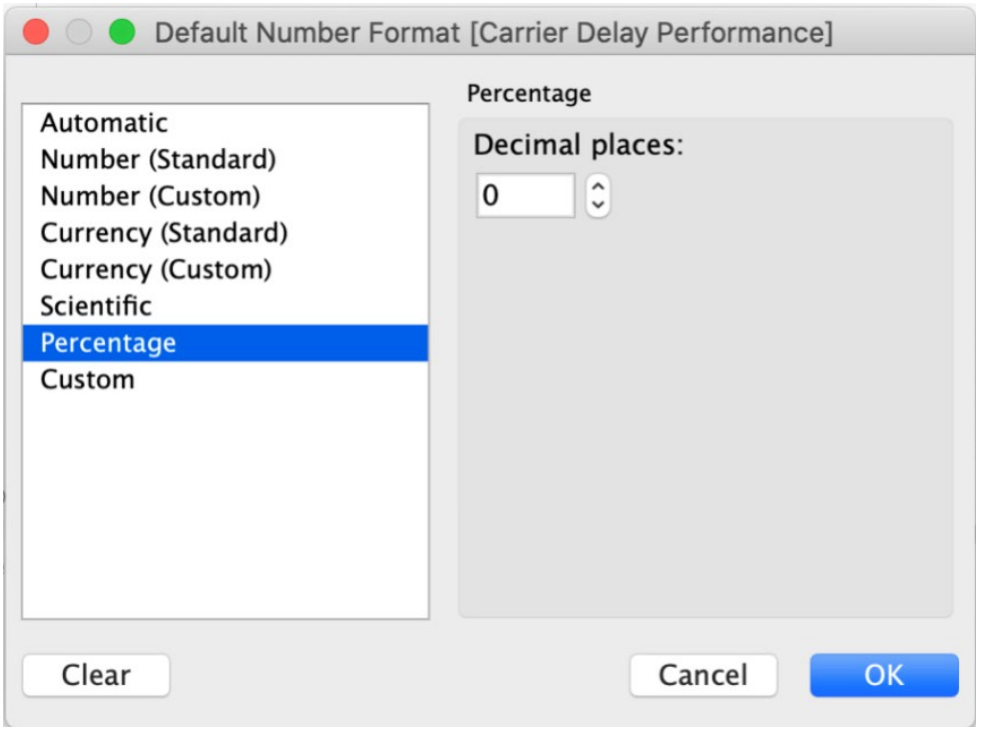

Figure 14.27: Percentage number format

6. Verify that your calculated fields are giving appropriate results. To do this, drag [Carrier Delay Performance] to the Text marks card. Don't worry about the correctness of the metrics at this moment, as this will be sorted later. The goal here is to check that your numbers are properly formatted. Here is what [Carrier Delay Performance] looks like:

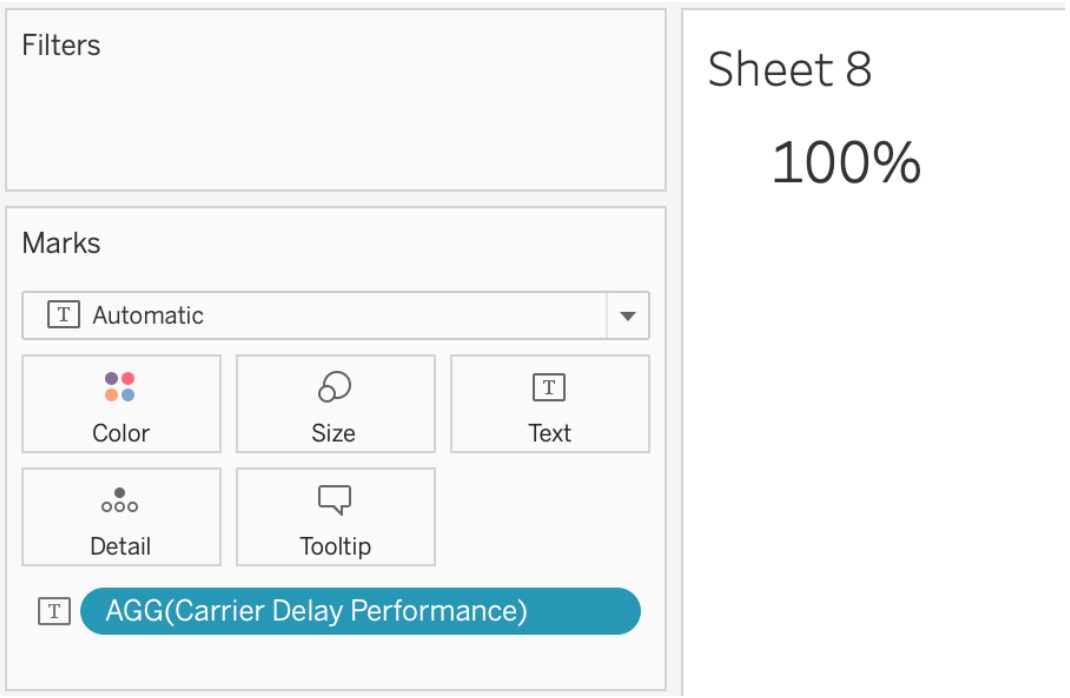

Figure 14.28: Validating the fields

Congratulations, you have now successfully created calculations that will ultimately be used in your Performance Scoreboard Dashboard at the end. Next up, you will learn how to add interactivity to your dashboards via parameters.

BUILDING INTERACTIVITY

One of the most essential elements in building Tableau solutions is to deliver interactivity to the user. This can be done in multiple ways (by creating parameters, filters, and actions) that allow users to update data values based on their needs. Many users can then use a single workbook or dashboard to answer an almost unlimited set of questions. In this section, you will create parameters and filters to add this interactivity to your final dashboard.

CREATING PARAMETERS

We covered parameters in incredible detail in *Chapter 11, Tableau Interactivity Part One*. Parameters, as stated previously, are very useful in Tableau for a variety of reasons. Here are some simple use cases that make parameters an essential part of any workbook or dashboard. You can use parameters to do the following:

- Create user selections that map to a calculated field for use as a filter.
- Compensate for noisy or dirty data by presenting correctly spelled options to the user.
- Turn cryptic data abbreviations into something more recognizable.
- Limit the input options for a user.

In the following example, you are going to add a few parameters to make it easy for users to navigate your workbook. Since you are creating a dashboard that displays a scoreboard at the origin airport level, it will be helpful to provide a parameter selection where users can easily select a single airport. Your source data in the **`delays.csv`** file provides abbreviated titles, which won't be helpful for most users. However, the secondary sources you joined do provide full airport names, which will make it simple to create an airport parameter.

Here's what your parameter looks like after you choose to populate it with data from the **`Origin Airport`** field (we will walk through the process of creating these in the next exercise):

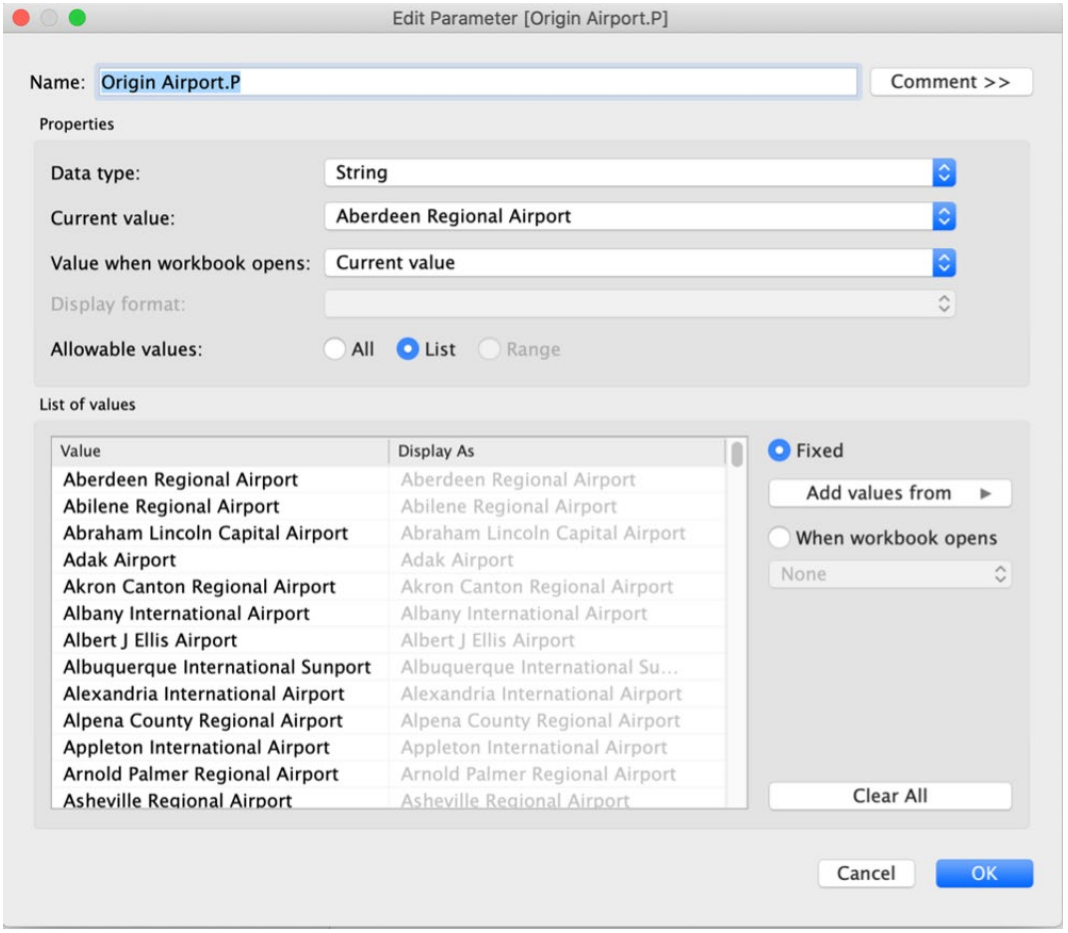

Figure. 14.29: Creating Origin Airport parameters

This parameter will be used shortly with a filter, ultimately enabling users to update an entire scoreboard with one simple drop-down selection.

Another powerful parameter is where users can select the type of delay they wish to analyze for a particular chart that you will add in your dashboard below. In this instance, since all delay types are measures, you need to build the parameter manually. This is easily done by setting the parameter to accept string values from a list, as shown below:

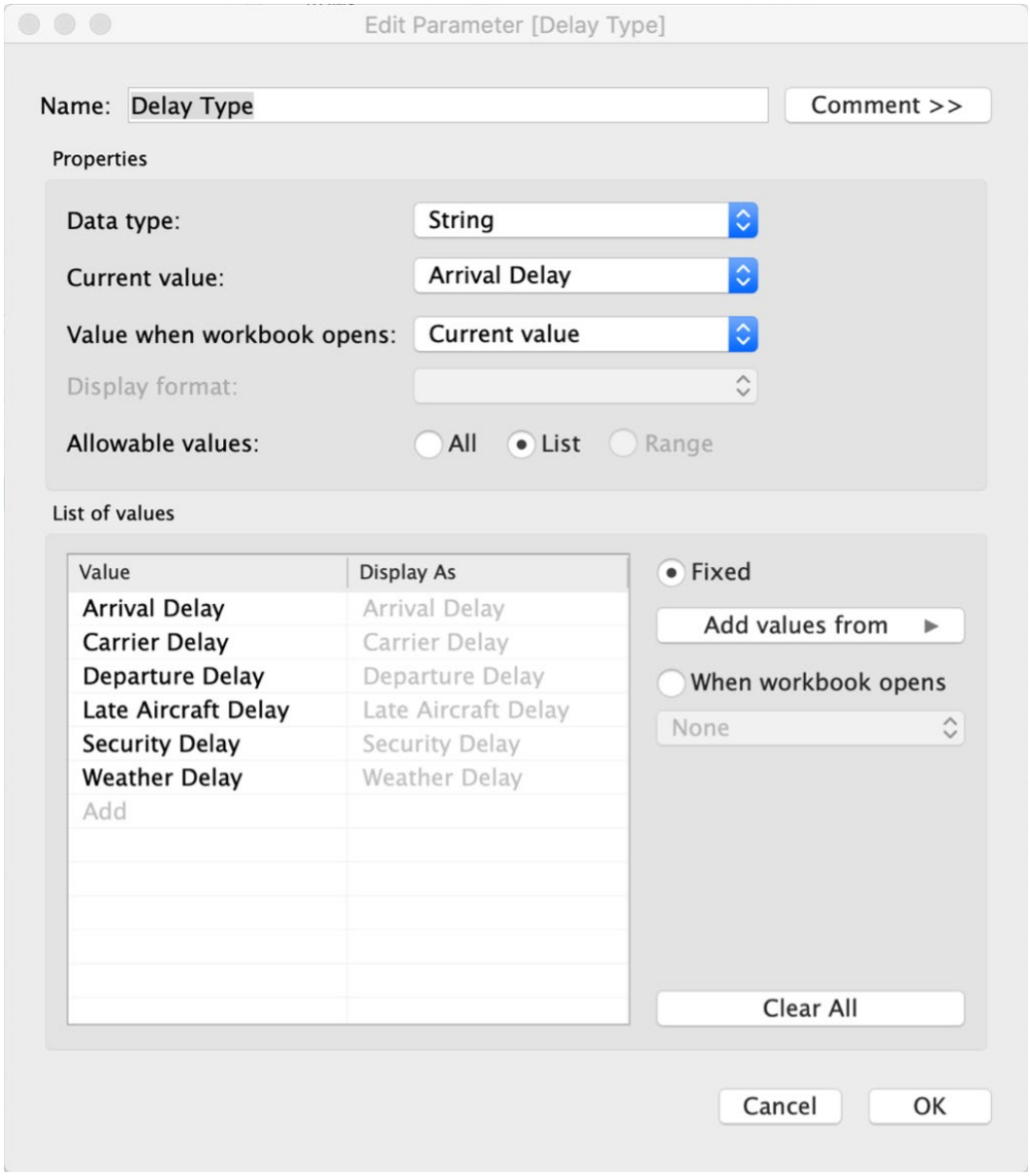

Figure 14.30: Creating a Delay Type parameter

Note how we have entered clear terms for the user to select, so there is no effort wasted in attempting to understand the options. As with the origin airport parameter, this will also be tied to a filter that allows users to control the content of the dashboard. However, if you recall from the parameter chapter, a parameter is only useful when it is used in a calculated filter to select a dimension or filter on a specific column. So, you will have to create a calculated filter so that you can use parameter selection by the end user to update the values in the worksheet.

CREATING CALCULATED DIMENSIONS

Calculated dimensions provide a lot of utility, especially as they can be used with parameters to create filters that help direct user input. In this case, you will create a calculated field to work with your **`Delay Type`** parameter, which will enable users to update a worksheet or dashboard using a very simple set of radio button options. The goal here is to tell your field how to work with input from the parameter.

To do this, you right-click and select **`Create`** > **`Calculated`** Field, and then enter the following calculation:

```
IF [Delay Type] = 'Arrival Delay' THEN [Arrival Delay]
ELSEIF [Delay Type] = 'Carrier Delay' THEN [Carrier Delay]
ELSEIF [Delay Type] = 'Departure Delay' THEN [Departure Delay]
ELSEIF [Delay Type] = 'Late Aircraft Delay' THEN [Late Aircraft Delay]
ELSEIF [Delay Type] = 'Security Delay' THEN [Security Delay]
ELSEIF [Delay Type] = 'Weather Delay' THEN [Weather Delay]
END
```

The **`[Delay Type]`** parameter will give end users the ability to change the type of measure they want to view on the scoreboard. Similar dimension/measure swapping was covered in *Chapters 11: Tableau Interactivity- Part 1* and *Chapter 12: Tableau Interactivity Part 2*.

BUILDING FILTERS

Filters are a powerful tool in Tableau, both as standalone simple filters and when paired with parameters or calculated fields. You will most often add several filters to make data discovery easier for your audience. Some of these will be simple filters based on your source data, while others will use formulas or calculated fields to help users sift through the data.

One helpful filter for this dataset will be used to focus on a minimum number of records to display in any given situation. In other words, the goal here is to hide data below a certain threshold, so you can better focus on more meaningful information. This is about the simplest filter you can build in Tableau. Here's a look at how it is used:

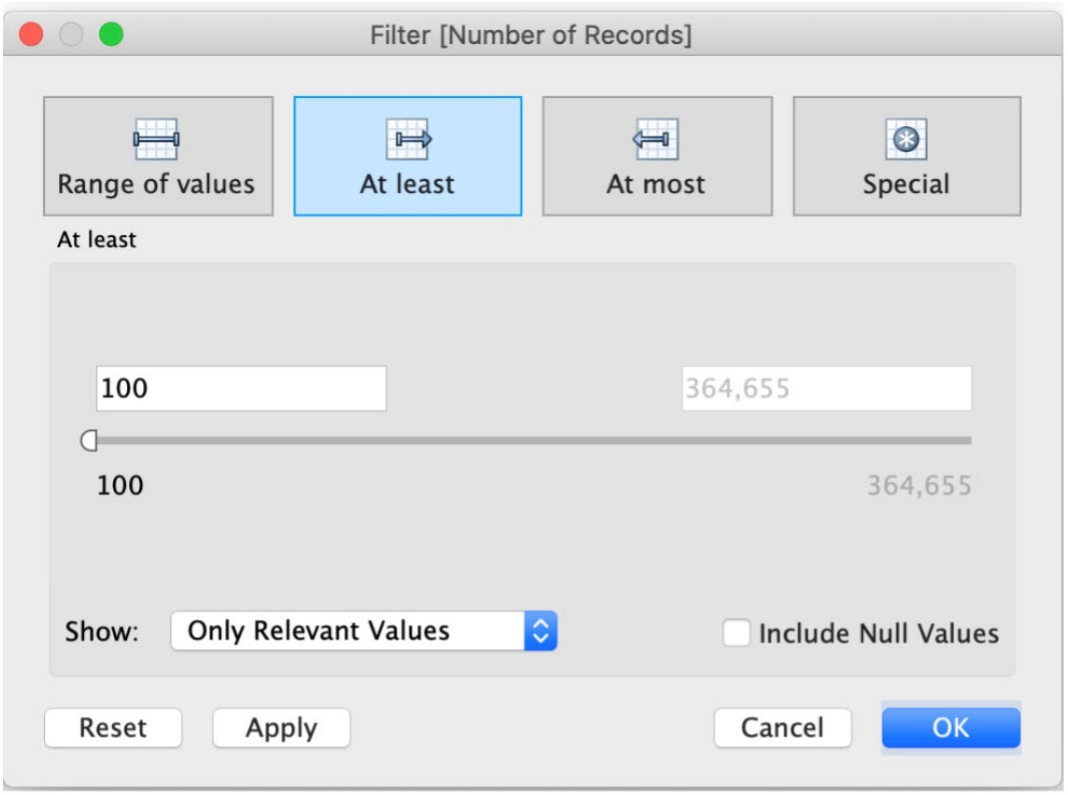

Figure 14.31: Filtering on Number of Records

In this case, the filter is set to display only dimensions with 100 or more corresponding records. You can easily set this level higher if you wish, or you could turn it around and filter out the largest values or a specific range, as needed. While this filter can be highly useful, it is applied to all levels of detail in the same manner. As you get deeper into the data, you may not see records as expected until the filter is removed or updated.

A second filter you will add is based on the airline of origin. In this case, the goal is to limit the number of airlines displayed on any given chart. Tableau filters have a very useful built-in Top N function you can employ in this way:

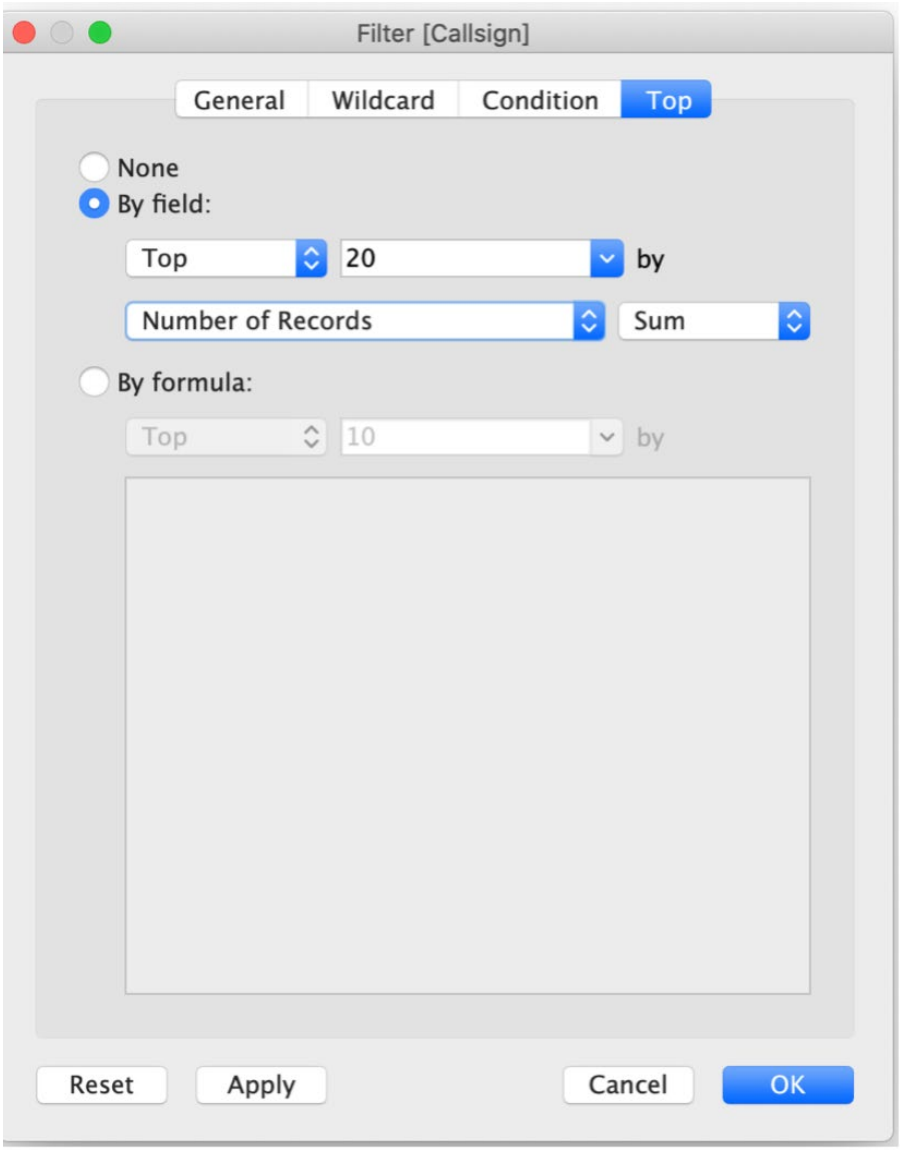

Figure 14.32: Top 20 Number of Records settings

The preceding filter displays the top 20 airlines by flight volume (number of records). This will update automatically as you filter by airport and provide a more robust method of filtering in contrast to the simple numeric filter you used earlier.

A third filter will provide the option to limit date ranges for all worksheets where it is applied. You build this using the **`FL_DATE`** field, giving users the ability to focus on a specific range of flight dates. Note that you need a green pill (for continuous dates) to build this filter. A blue pill (discrete dates) will provide specific date options, such as month and year, rather than the continuous range selected here.

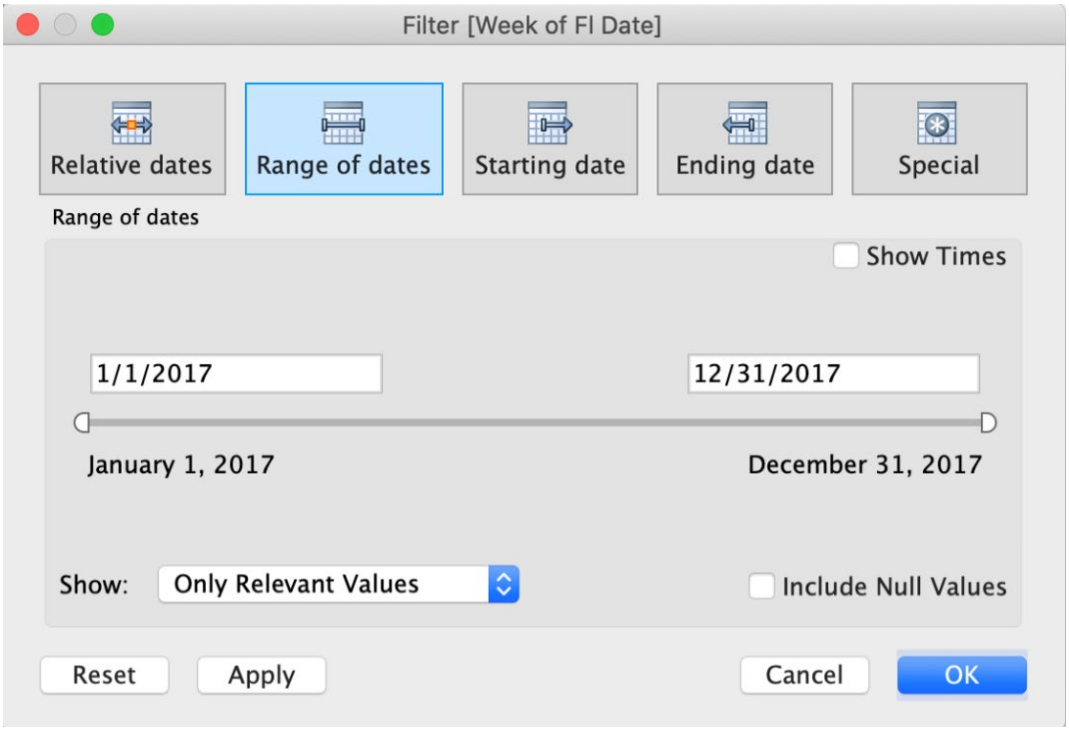

Figure 14.33: Filtering on flight date

Next, you will use a filter for your origin airport. Recall how you built the origin airport parameter earlier. You will now use it as part of your airport filter logic. This requires a calculation in which you reference the selected parameter value:

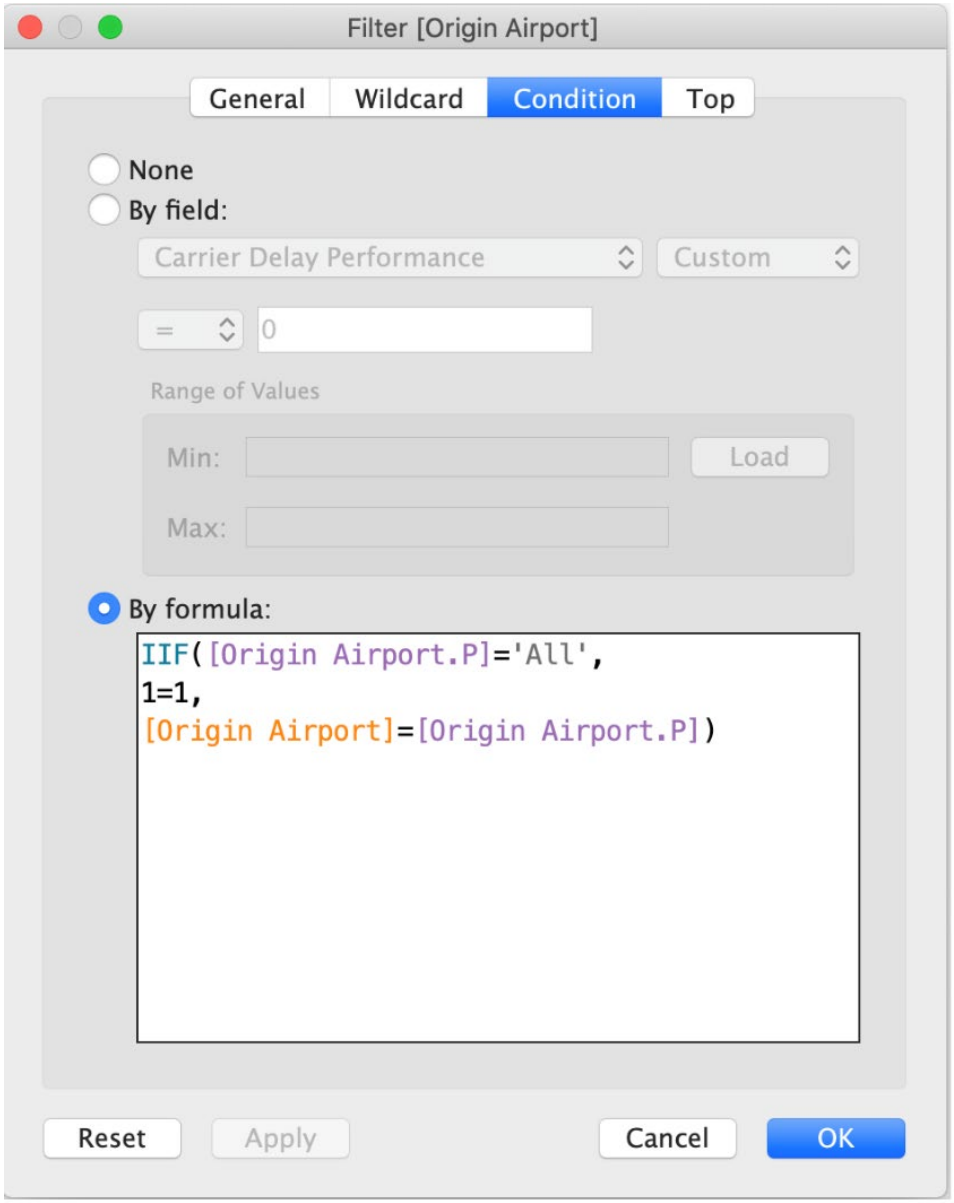

Figure 14.34: Using a conditional filter formula

You start by dragging the **Origin Airport** dimension to the **Filter** pane and add the **IIF** calculation. This tells the filter to display data for all airports when the user selects **All** from the parameter option list. Otherwise, data should display only for the specifically selected airport. Since you originally built the parameter from the dimension values, these values will be an exact match, making this process quite simple.

The final step in applying filters will take you back to the calculated field you built previously. Since this filter displays measure values, it can be used to populate the charts where it is applied. In this case, you simply drag the **Delay Type Filter** pill to either the **rows** or **columns** mark, depending on the chart type. The result will be a display with the selected **delay type** measure filtered by the previous four filters you built. You'll see how this all comes together visually in the next section.

EXERCISE 14.04: CREATING FILTERS AND PARAMETERS

This exercise will walk through each of the parameters/filters built in previous sections and the steps to take to create them. Previous sections have explained the reasoning for these filters; now it is time to start building them.

Perform the following steps to complete this exercise:

1. Using the same Tableau workbook as in *Exercise 14.3*, create an **Origin Airport** parameter so that your users can select an airport on which the whole workbook will be filtered.
2. Right-click anywhere under the **Parameter** section of the **Data** pane to create a new parameter named **Origin Airport.p** (where **.p** is used to differentiate between parameter and other dimensions). Select **String** for Data type and set **Allowable values** to **List**.
3. Before you load your **Origin Airport** from a field, manually add **All** for **Value** and **All Airports** for **Display** As since you also want your end users to select **All Airports** and not just a single one.

4. Once you have typed in **`All Airports`**, click on the **`Add values from`** field and select **`Origin Airport`**.

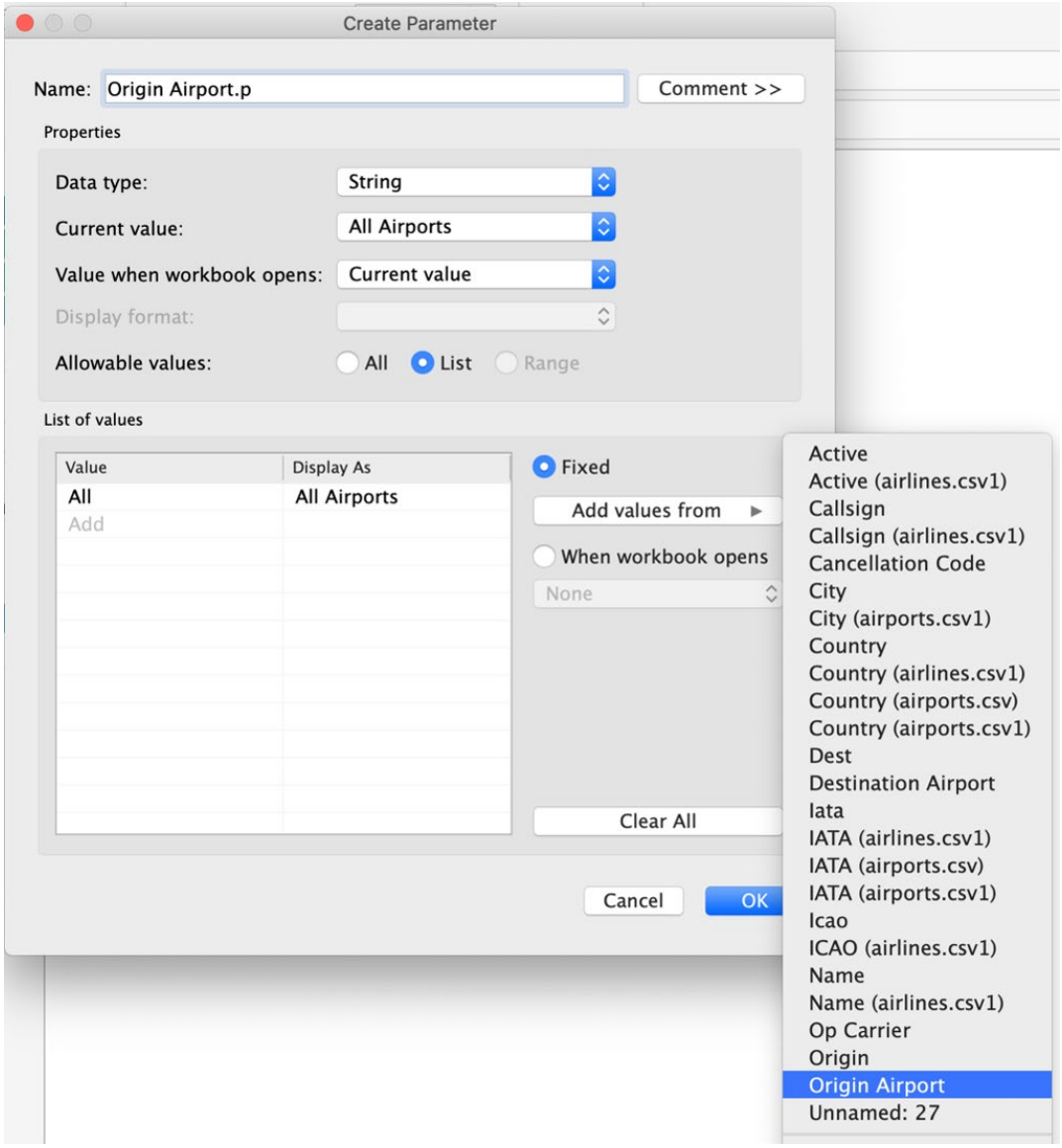

Figure 14.35: Adding parameter values from a dimension

5. After you click on **`Origin Airport`**, all the values from **`[Origin Airport]`** columns should load below **`All Airports`** and click **`OK`** once the column is loaded as shown below:

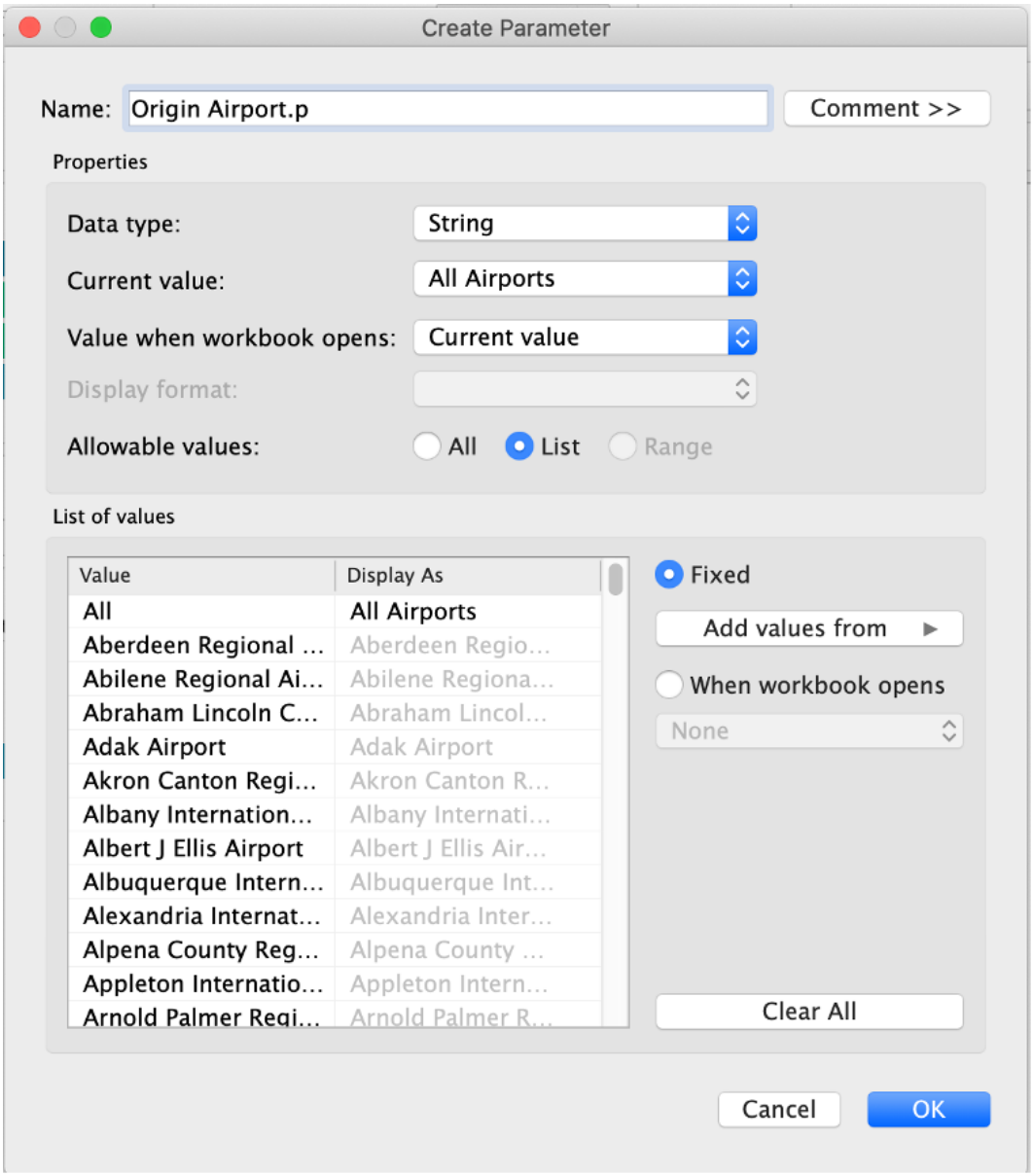

Figure 14.36: Manually adding the "All" value option

6. Manually create the **`Delay Type`** parameter by adding the **`Arrival Delay`**, **`Carrier Delay`**, **`Departure Delay`**, **`Late Aircraft Delay`**, **`Security Delay`**, and **`Weather Delay`** values.

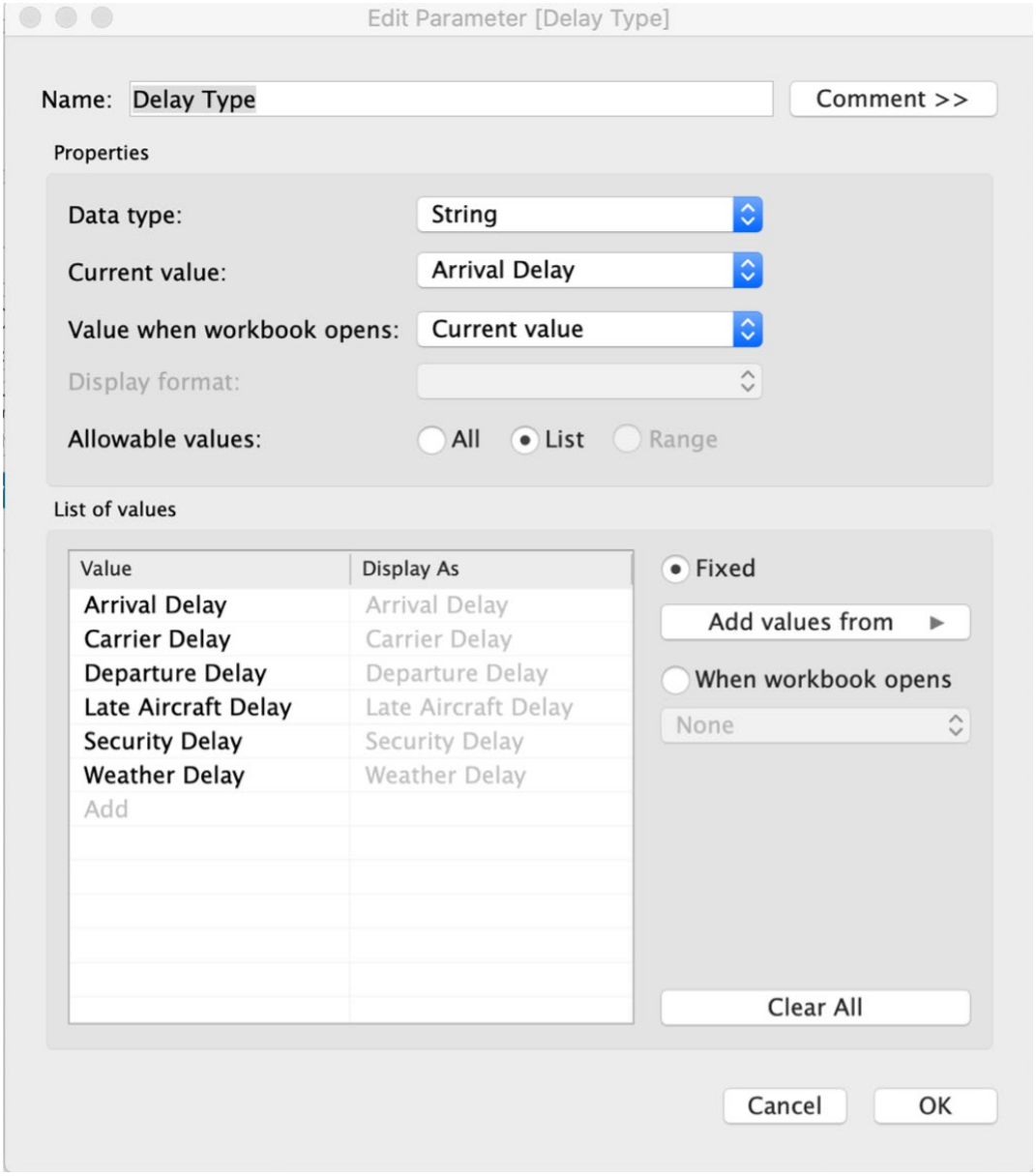

Figure 14.37: Adding the Delay Type parameter

7. Create the **`Delay Type Filter`** calculated field measure using the following calculation:

```
IF [Delay Type] = 'Arrival Delay' THEN [Arrival Delay] ELSEIF [Delay
Type] = 'Carrier Delay' THEN [Carrier Delay] ELSEIF [Delay Type] =
'Departure Delay' THEN [Departure Delay] ELSEIF [Delay Type] = 'Late
Aircraft Delay' THEN [Late Aircraft Delay] ELSEIF [Delay Type] =
'Security Delay' THEN [Security Delay]ELSEIF [Delay Type] = 'Weather
Delay' THEN [Weather Delay] END
```

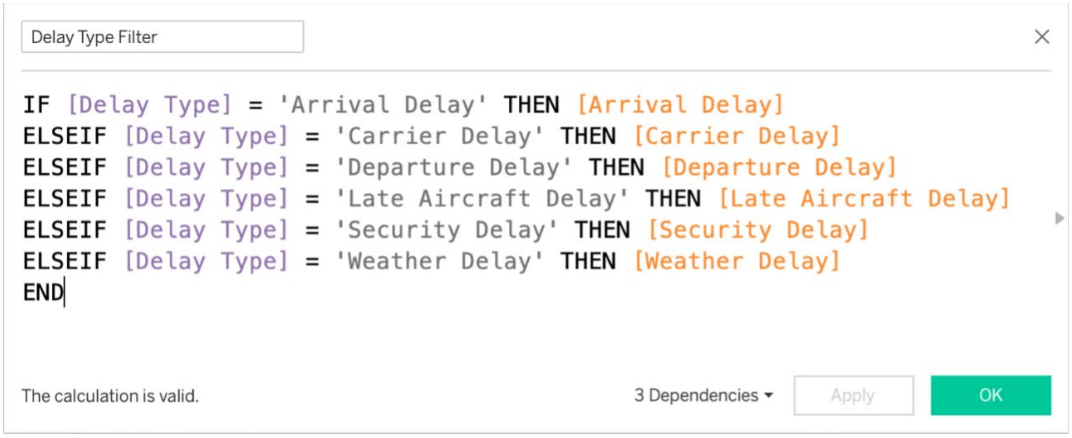

Figure 14.38: Delay Type Filter formula

Essentially, this calculated field is used for measure swapping using parameters that have previously been covered in the book. Whenever the **`[Delay Type]`** parameter value is changed, the measures in your selected worksheet will change to the selected value. That is where the **`IF`** statement comes in.

8. Build the Top N filter using the **`Callsign`** dimension from the second airline source (for origin flights) by dragging it to the filters card. Set it to the **`Top 20`** values by **`Number of Records`**, set to the **`Sum`** operator:

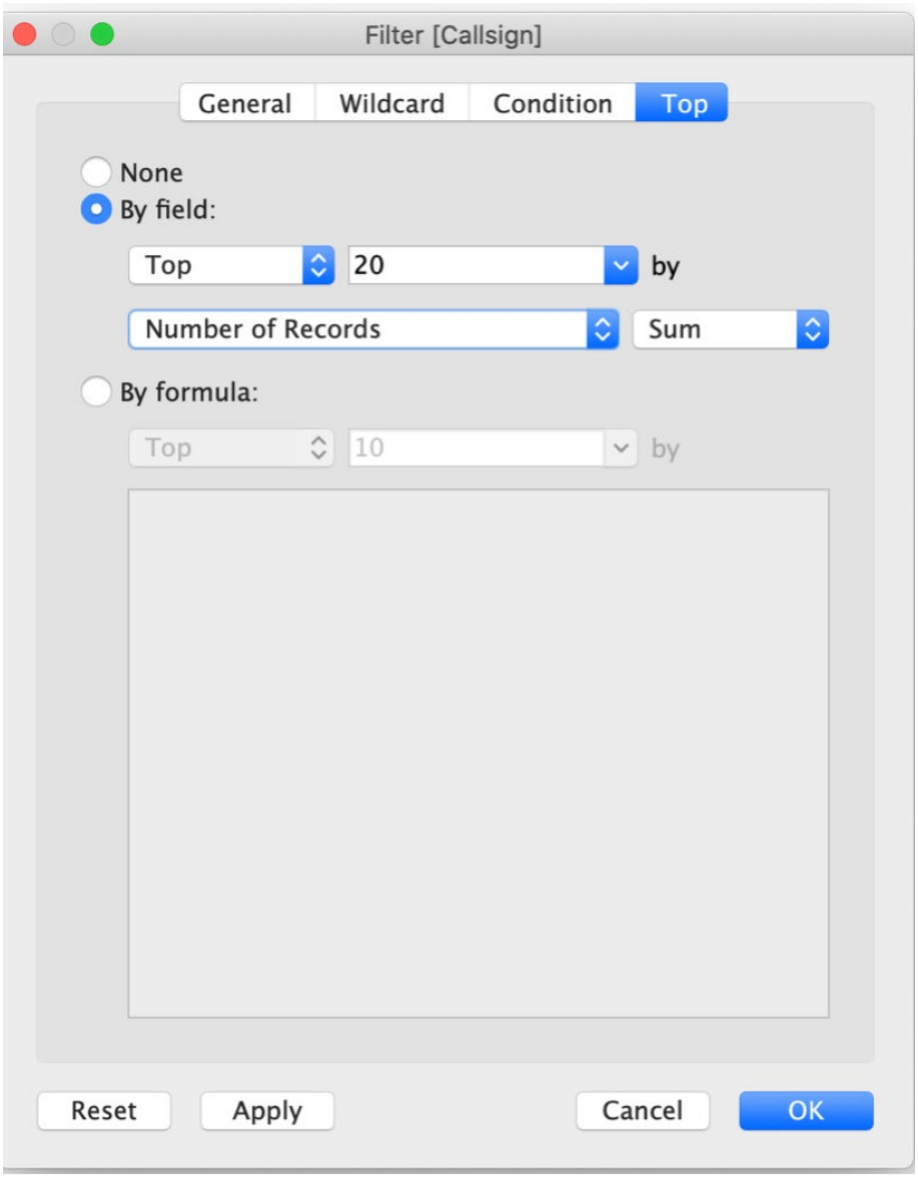

Figure 14.39: Selecting Top N records

9. Add a continuous date filter using the **`FL_DATE`** field. Use the **`Range of dates`** option and set it to show all dates in the date range:

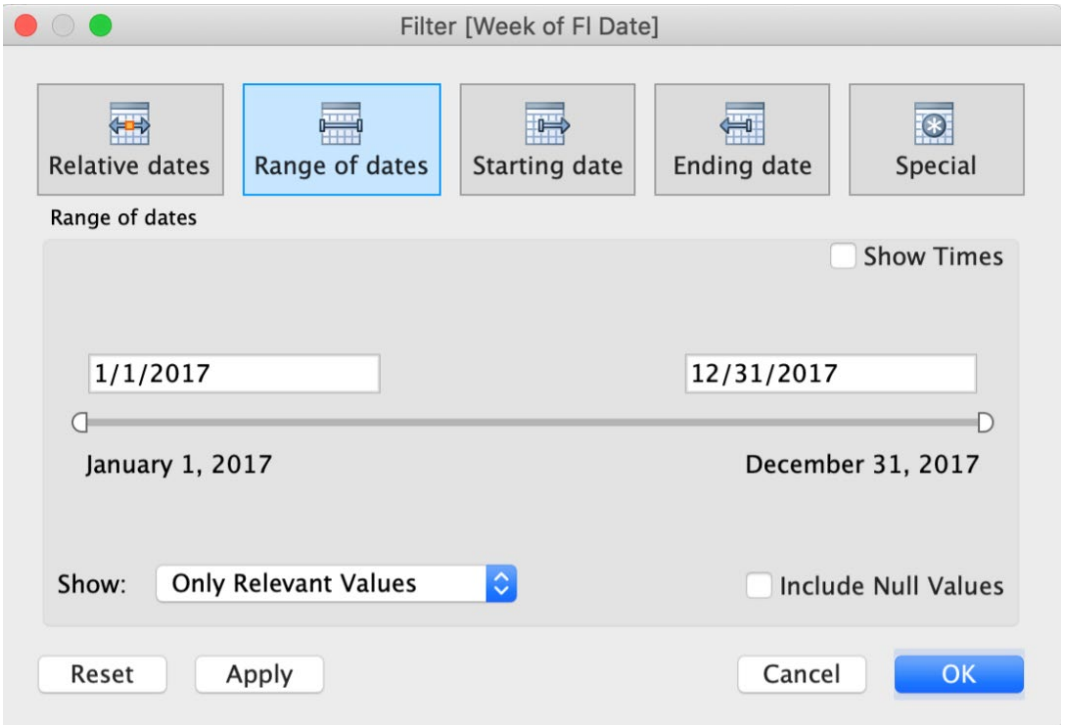

Figure 14.40: Filtering on flight FL_DATE

10. Create the **`Origin Airport`** filter by dragging the **`Origin Airport`** dimension to the **`Filter`** pane in any worksheet. Add the following calculation to the **`By formula`** section on the Condition tab:

```
IIF([Parameters.Origin Airport]='All', 1=1, [Origin Airport] =
[Parameters].[Origin Airport]):
```

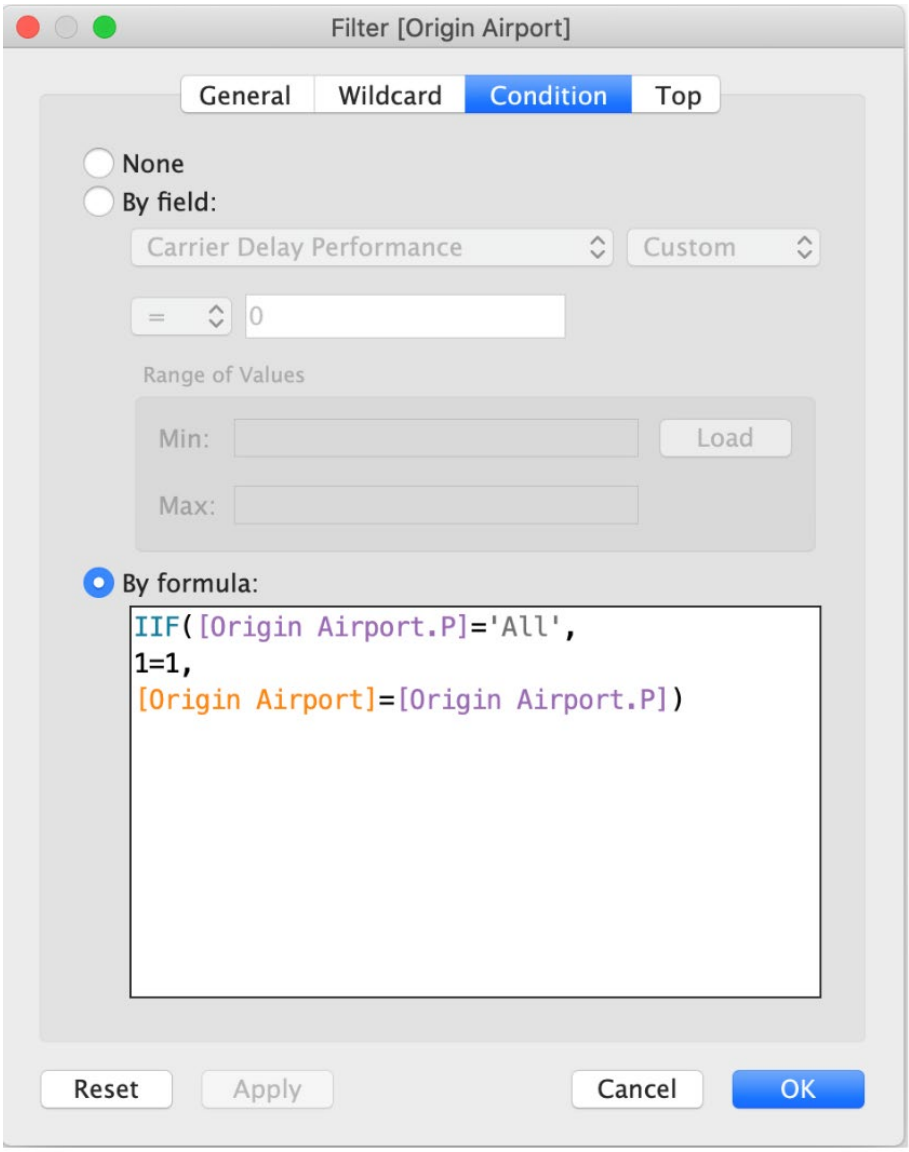

Figure 14.41: Utilizing a conditional filter formula

The formula tells the filter to display data for all airports when the user selects **`All`** from the [**`Origin Airport.p`**] parameter option list since you manually added the **`All Airports`** option for this purpose. Else, end users can only select specified airports.

This wraps up our section on adding more interactivity and filters to your dashboard. We covered combinations of parameters and filters that will power your worksheets and dashboard. In the next section, you will start creating worksheets where all the calculated fields, parameters, and filters will be utilized.

ADDING WORKSHEETS

Worksheets are where any Tableau analysis starts and can be used to display many chart types, crosstabs, text, and many other options. Since worksheets are easy to add during the design process, you will use many of them within this project. A limited number will wind up on your dashboard, and some may be removed from the workbook during the cleanup process at the end of your work. What you will use many worksheets for is to analyze different data combinations and chart types as you seek to deliver the optimal insights and formats for the user.

IDENTIFYING FOCUS DATA

One of the critical aspects of building useful workbooks and dashboards is knowing which data elements to focus on. In some cases, you will know at least some of the data that must be included (that is, corporate KPIs), while in other instances, you will have to spend time discovering what is important. Your dataset in this chapter will be new to most users, so some time needs to be spent visually analyzing the data to understand critical elements.

Examine this dataset to understand where your focus should lie, so you can then direct your efforts most efficiently. Your primary data elements include the following:

- Airline carrier information dimensions
- Airport information dimensions
- Airport location dimensions
- Flight date dimensions
- Airtime measures
- Flight time measures
- Flight delay measures

- Taxi measures
- Wheels on and off measures
- Canceled flight information measures

Merging all of these would create endless combinations but would also lead to a dashboard without focus. It is therefore wiser to narrow down the list initially and build the content for a single dashboard, knowing that you can always select other combinations in the future for a subsequent dashboard.

In this case, the focus will be on airport and airline delays. There will still be thousands of potential data points, but the dashboard content will adhere to a specific theme and the user will be able to focus on this specific subset of the information. The following section will discuss the chart types that may be most beneficial for displaying these data elements.

SELECTING CHART TYPES

The best chart for a worksheet depends on the type of analysis you seek to provide. There are many guides to chart visualization that all follow the same general principles. We have covered multiple chart type options in detail in *Chapter 4, Exploring Comparison and Composition* and *Chapter 5, Data Exploration, Distribution, and Relationships*. Depending on the end goal, your chart selection will differ from sheet to sheet. Tableau will make chart recommendations based on the number of dimensions and measures you are using in a worksheet, but it is best not to depend solely on this recommendation. It is in your best interests to test multiple chart types with the data you are displaying, since Tableau makes it simple to experiment using the **`Show Me`** chart menu.

Let's walk through each of the worksheets in the workbook and identify the best choice for each display and why that is the case. Note also that your ultimate dashboard will typically look best when it has a variety of different charts, tables, and summary panes, all held together using consistent stylistic elements.

You'll begin with a sheet labeled **by Airport Trend**, where the goal is to display variation over time for a specific delay type. This is a rather simple chart to assemble, given the time element and a single measure you wish to display. In most cases, a line chart will be the best option whenever there is a time element on the horizontal axis, especially one with several years of daily data, as in your case. While line charts may seem simple, you can enhance this chart to add context to the data by using a distribution range, as you'll see in the next exercise where you will create trend line charts to showcase the overall trend of measures.

EXERCISE 14.05: CREATING A TREND LINE CHART

In this exercise, you will create the **Delay by Airport Trend** chart where you can visualize how delays change over time.

Perform the following steps to complete this exercise:

1. Following on from the previous exercise, create a blank worksheet named **by airport trend**. Drag the **FL_DATE** dimension to the **Columns** card. Set it to **WEEK** and keep it continuous (not discrete):

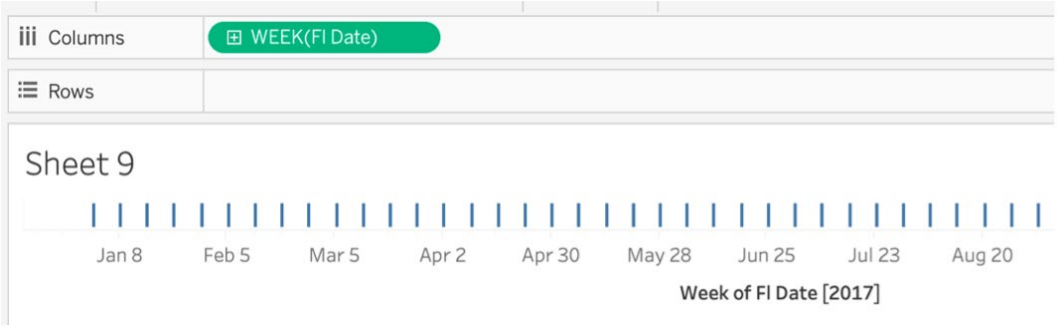

Figure 14.42: WEEK(FL_DATE) view

2. Drag the **`Delay Type Filter`** to the **`Rows`** card. Make sure it is set to **`AVG`** (average):

> **NOTE**
>
> The graphs will slightly different due to sample data in use vs entire data (700 MB) which was used writing the book.

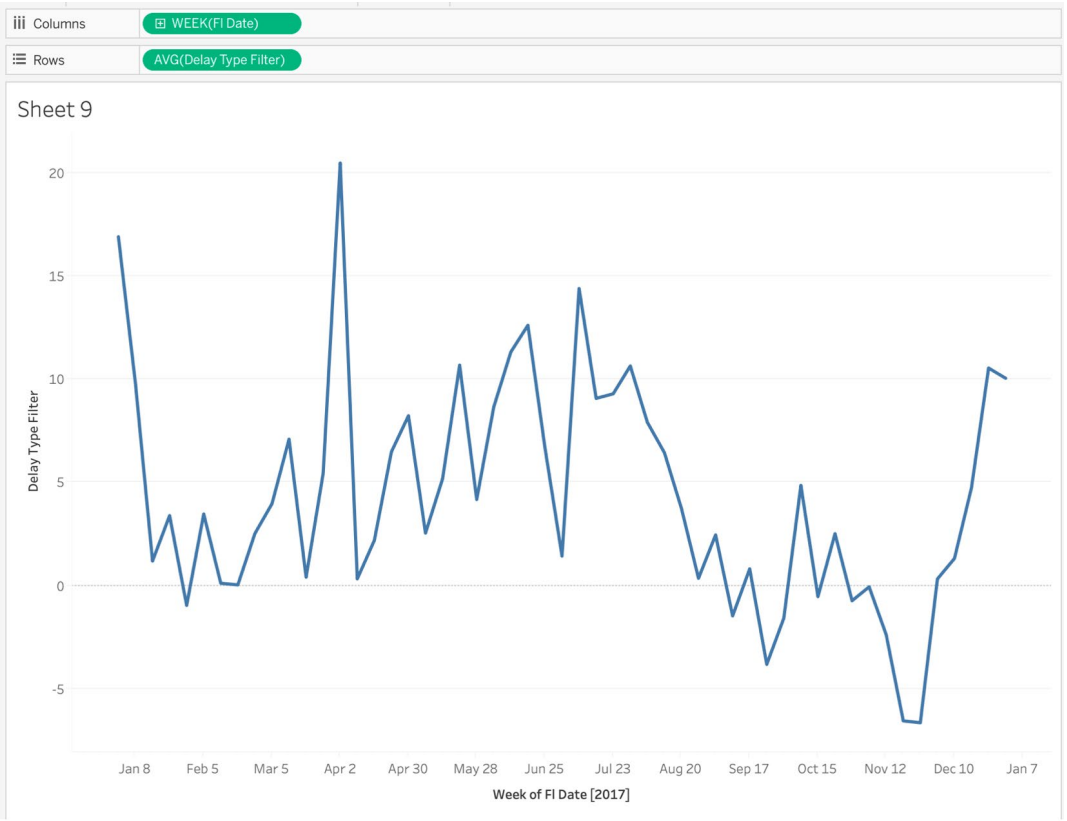

Figure 14.43: Delay Type Filter by flight date line chart

3. Add filters for **`SUM(Number of Records)`**, **`Callsign`** (from **`airlines.csv1`**), **`Origin Airport`** (from **`airports.csv1`**), and **`FL_DATE`** (set to **`WEEK`** and continuous). Origin Airport should use the same formula you used in *Exercise 14.4*.

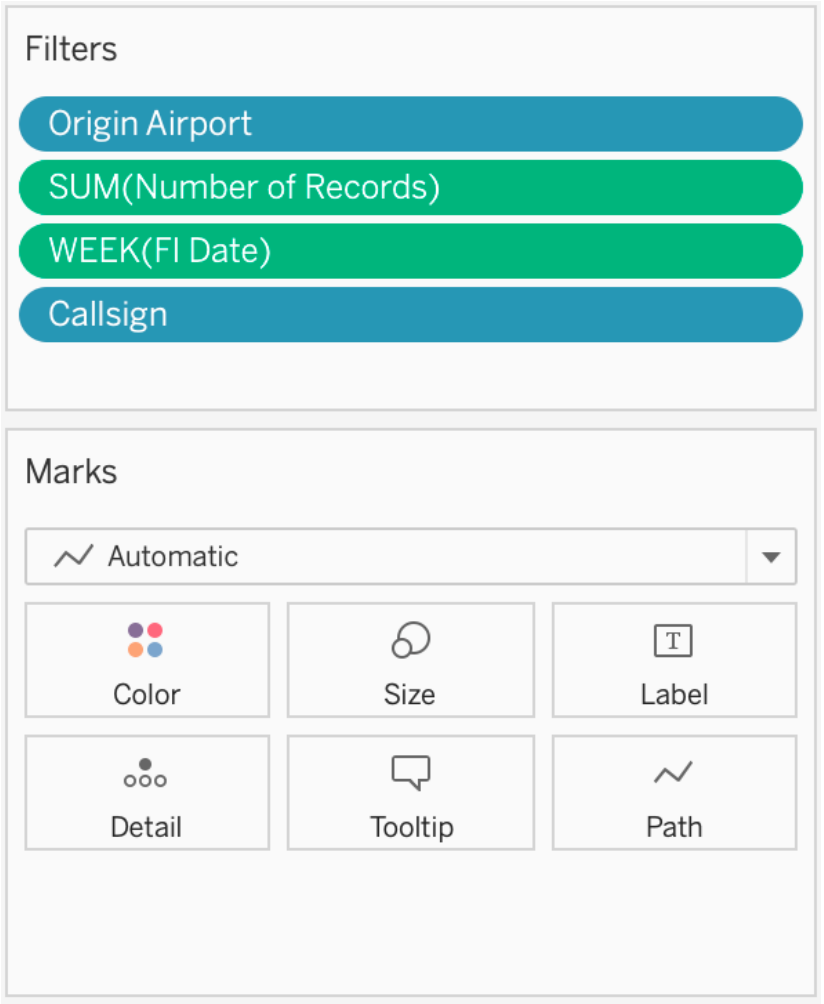

Figure 14.44: Marks card filters

4. Change the **`Marks`** type to **`Line`** if it's not already a line chart.

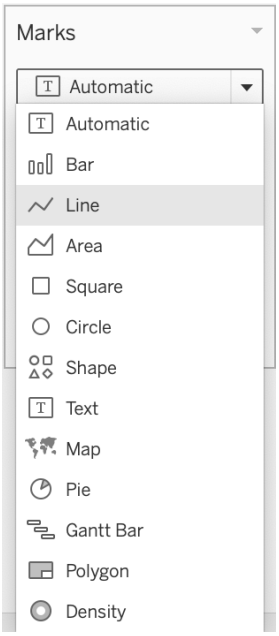

Figure 14.45: Marks type selection

5. Navigate to the **`Analytics`** pane and add **`Distribution Band`** to the pane as shown below:

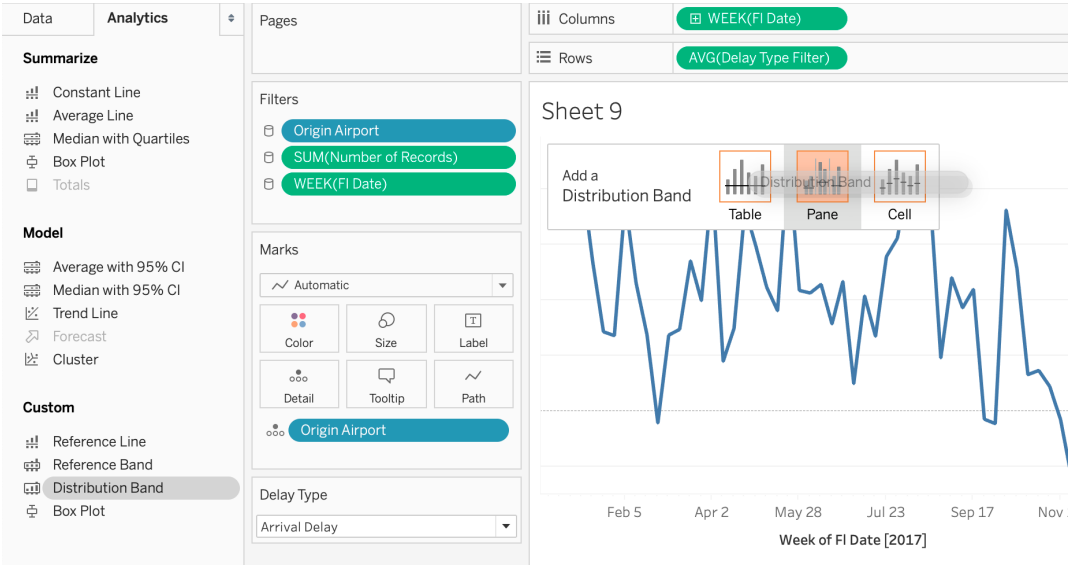

Figure 14.46: Dragging Distribution Band to the view

6. In the settings box, click on **`Distribution`** > **`Per Pane`**, select **`Quartiles`** for **`Value`**, and set **`Value`** for **`Label`**. You can experiment with **`Line`** and **`Fill`** as you desire.

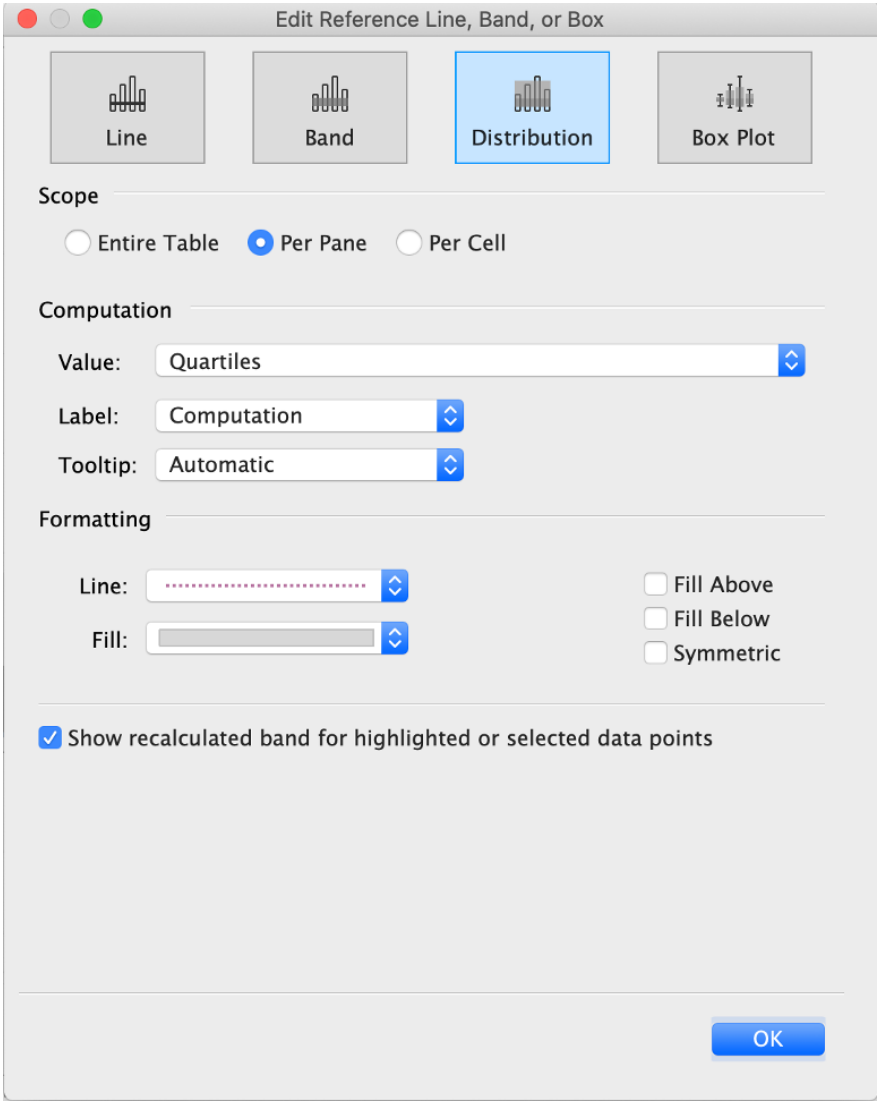

Figure 14.47: Distribution band settings

7. Hide the axis header by unchecking the **`Show Header`** option.

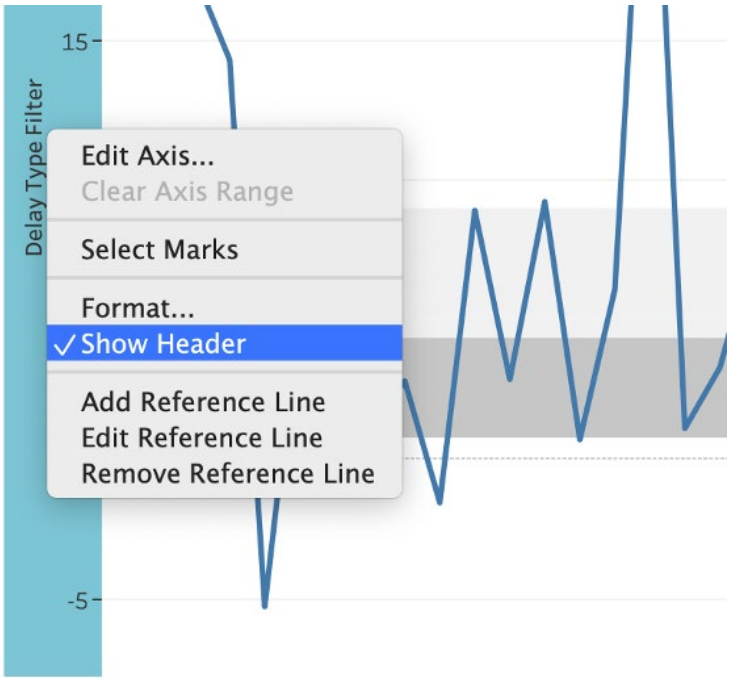

Figure 14.48: Hiding/showing header

8. Show your **`[Delay Type]`** parameter, so that end users can select the type of delay metric they want to see as part of the chart. Right-click on the **`[Delay Type]`** as well as **`[Origin Airport]`** parameter as shown:

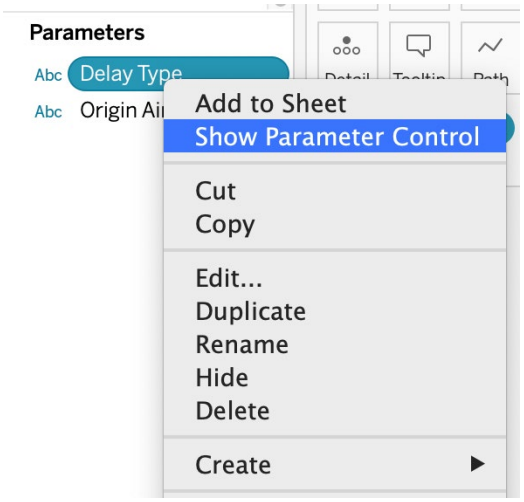

Figure 14.49: Show Parameter Control

9. Add the **`Origin Airport`** dimension as **`Detail`** on the **`Marks`** pane as you want to use the **`Origin Airport`** name in your tooltip next.

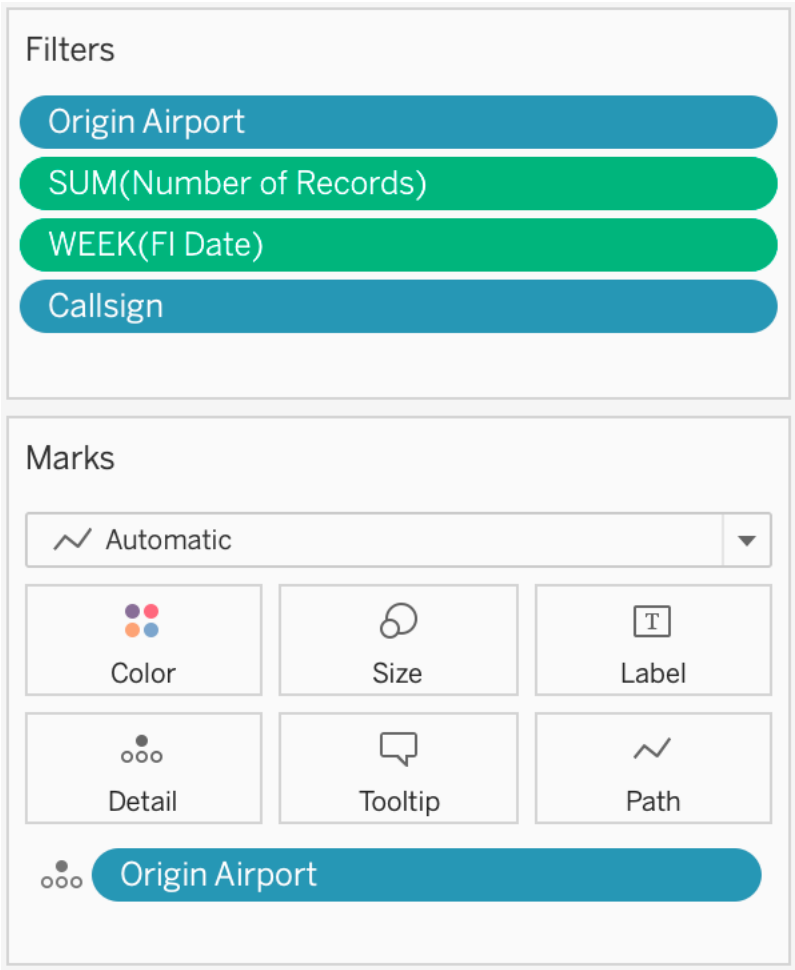

Figure 14.50: Adding Origin Airport as Detail on the Marks card

10. Create a tooltip detailing the display information: The week of **<WEEK(Fl Date)>** had an average **<Parameters.Delay Type>** of **<AVG(Delay Type Filter)>** minutes of flights from <Parameters.Origin Airport.P> as shown in the following image:

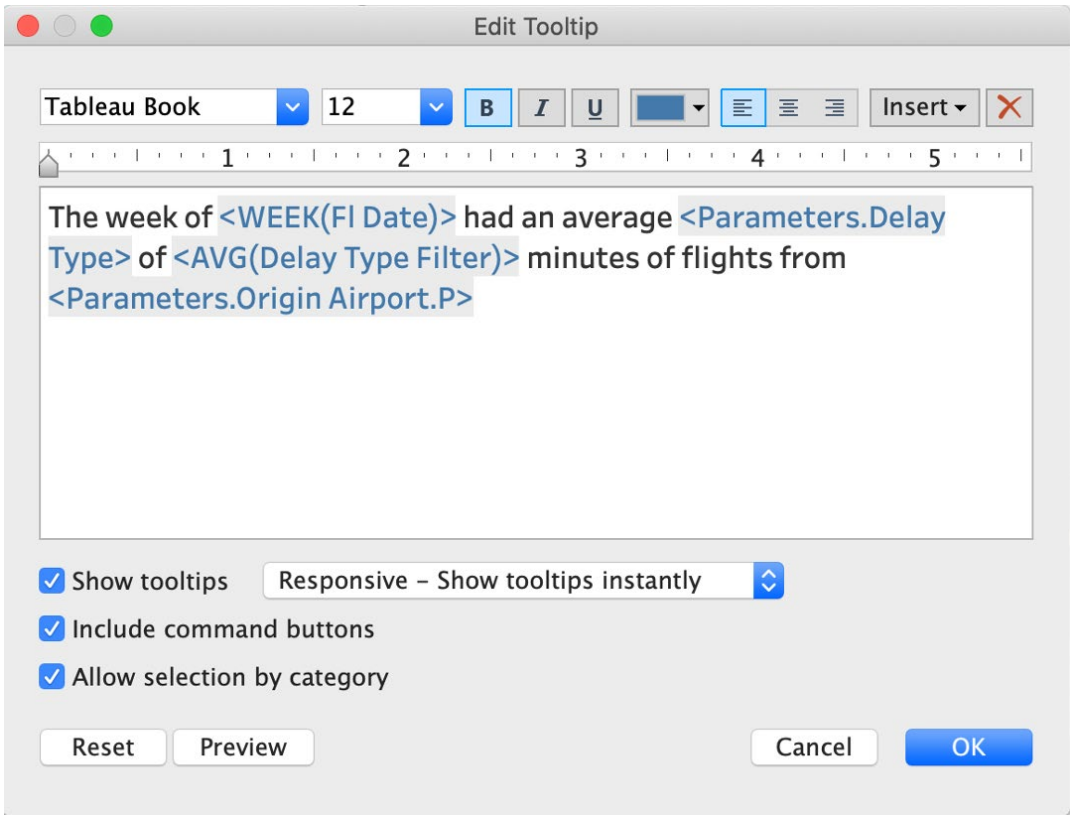

Figure 14.51: Dynamic tooltip editing

11. Use the dimensions/measures/filters/parameters in the view to create a contextual tooltip. The preceding will result in the following finished product with a contextual tooltip whenever you hover over one of the weeks.

Here is the finished product:

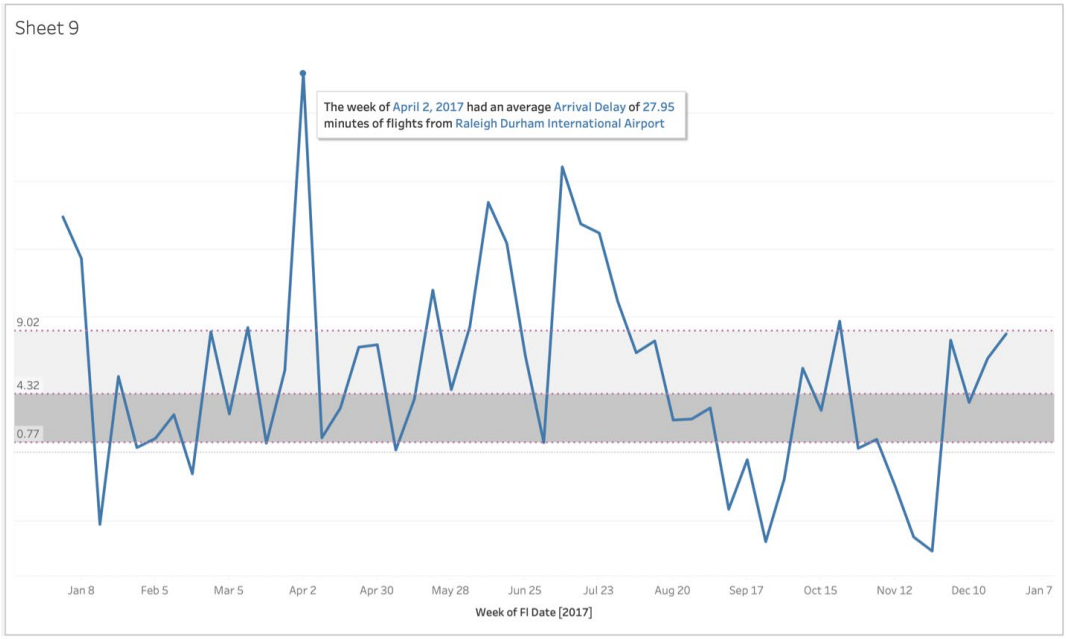

Figure 14.52: Final trend line chart view

In this exercise, you created a line chart to track delay trends—complete with quartile ranges, tooltips, and multiple filters. This chart will become part of your dashboard later in this chapter.

Your second worksheet and chart will examine delays by carrier for a specific airport, as selected by the user. Unlike the prior chart, you are not working with a time axis, so the chart options are expanded slightly. It would be easy to default to a bar chart for this type of data, but a dot plot would be the more elegant solution.

This chart will have airports along the vertical axis (it is easier to display full names this way) and delay measures along the horizontal axis. You will also size your dots (circles) based on the number of flights per airline from the selected airport. This provides a third dimension to the data that would be more difficult to achieve using a bar chart. It will also help you see how large carriers are performing relative to secondary carriers at any given airport.

You will build this chart in the following exercise.

EXERCISE 14.06: CREATE A DOT PLOT

Continuing from the previous exercise, in this exercise, you will create a chart that displays delay data by airline for a selected origin airport.

Perform the following steps to complete this exercise:

1. Add a worksheet and name it **`by airport & carrier`**.
2. Drag the **`Delay`** Type **`Filter`** measure to the **`Columns`** shelf. Make sure it is set to **`AVG (average)`**, then drag the **`Name`** field from the second airline data source to the **`Rows`** shelf (for origin data).

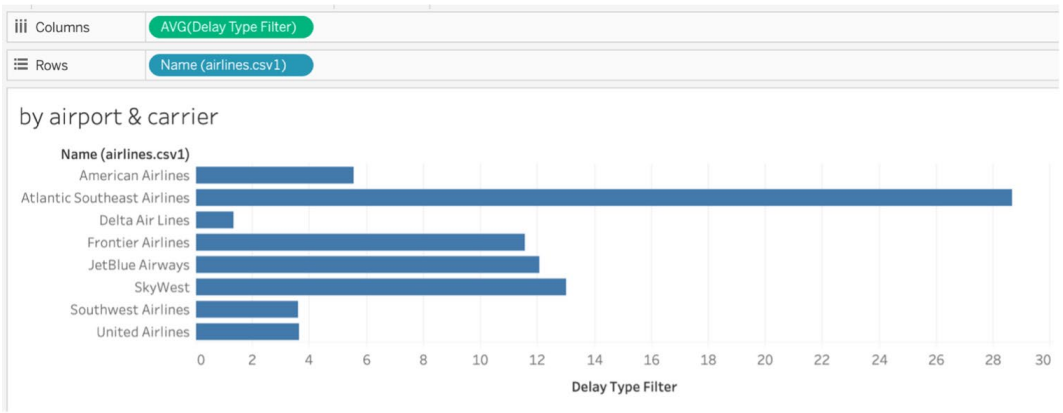

Figure 14.53: Delay type by airport & carrier

3. Add filters for **`SUM(Number of Records)`**, **`Callsign`** (from **`airlines.csv1`**), **`Origin Airport`** (from **`airports.csv1`**), and **`FL_DATE`** (set to **`WEEK`** and continuous). **`Origin Airport`** should use the same formula as in Exercise 14.4. You can avoid manually creating each of the filters by applying all the filters with **`All Using This Data Source`** as you will be using the same filter across all the following exercises.

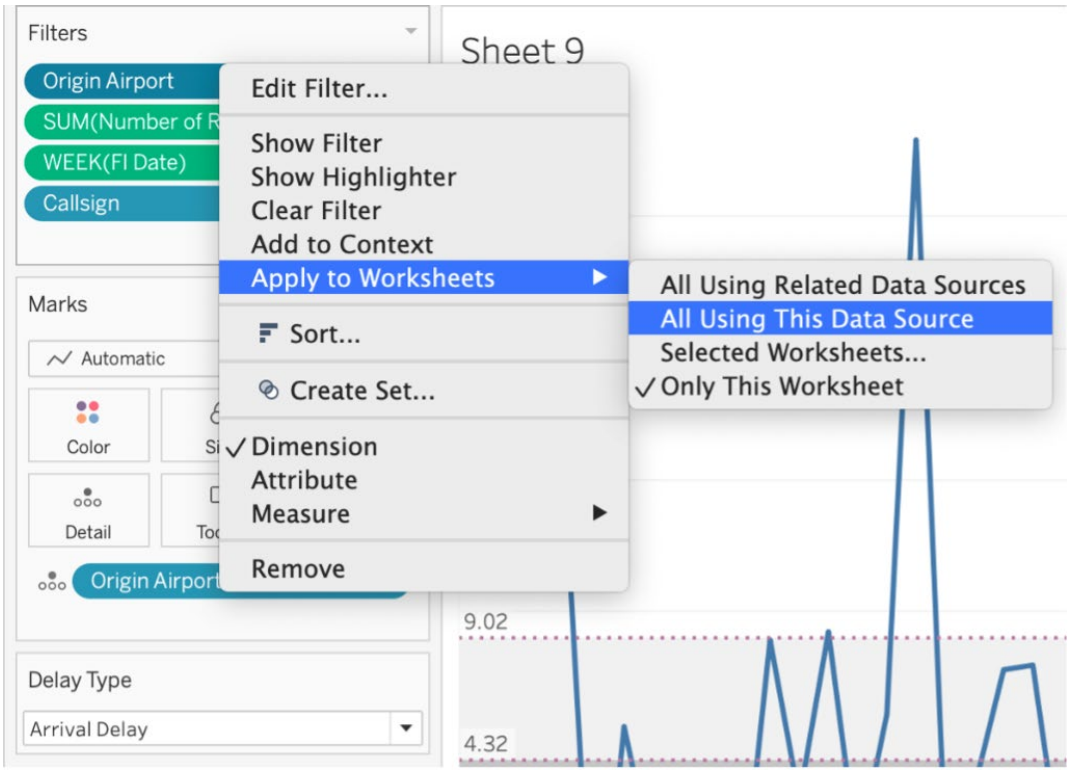

Figure 14.54: Applying a filter to all worksheets using the same data source

4. Select **`Circle`** from the **`Marks`** card:

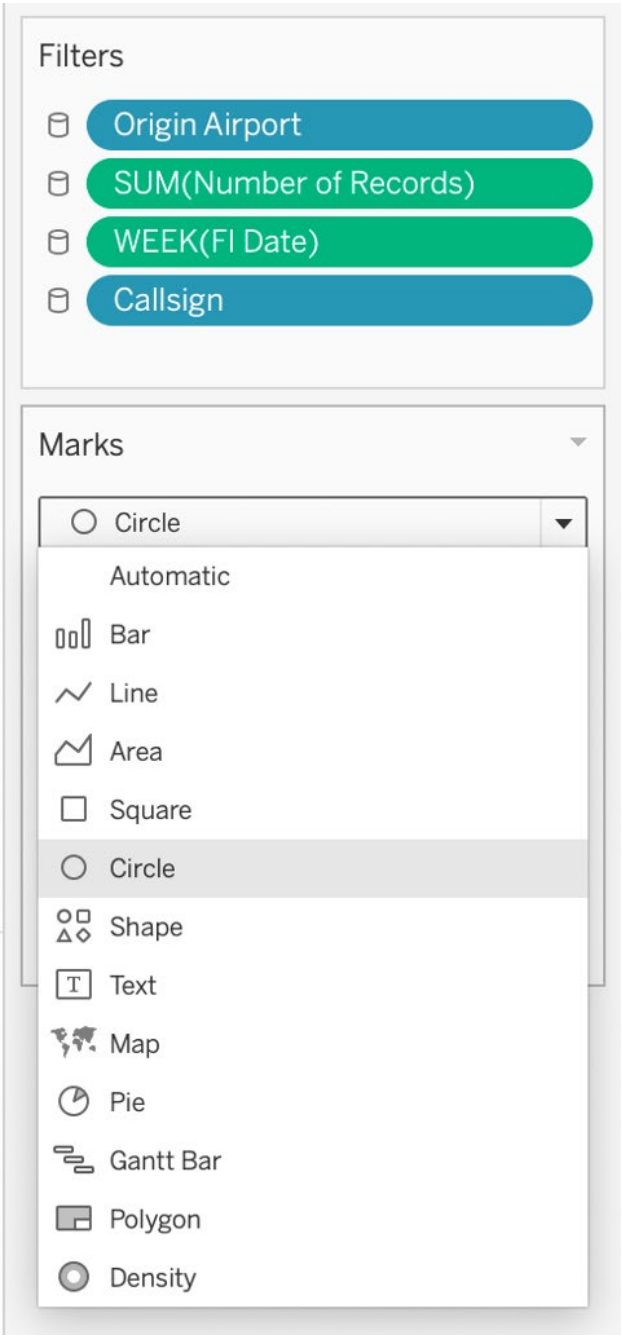

Figure 14.55: Changing the Marks type to Circle

5. Add **`SUM(Number of Records)`** to the **`Size`** card in the **`Marks`** pane and also add **`[Name(airlines.csv1]`** to the **`Tooltip`** card, as shown below:

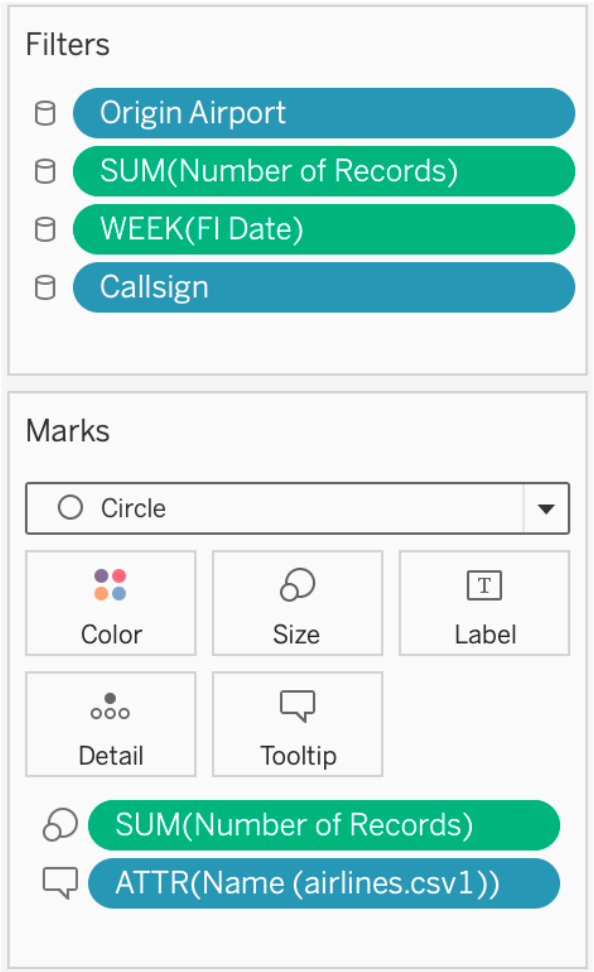

Figure 14.56: Adding SUM(Number of Records) to the Size Marks card

6. Add a reference line to the columns using the **`Distribution`** button. Set to **`Per Pane`** and **`Quartiles`** and set **`Label`** to **`Value`**:

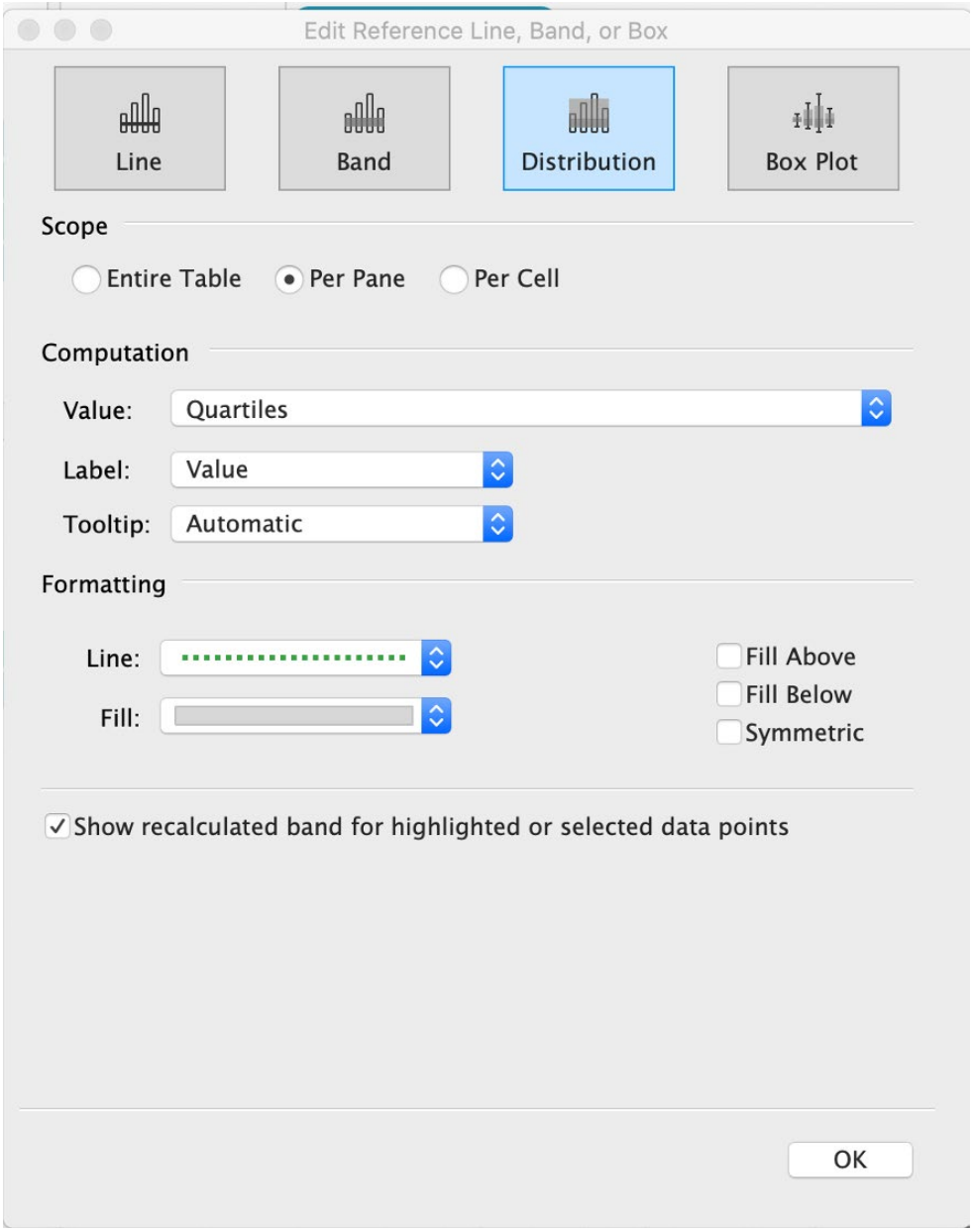

Figure 14.57: Editing distribution bands with quartiles

7. Hide the axis header by unchecking the **`Show Header`** option, then show the **`[Delay Type]`** and **`[Origin Airport.p]`** parameters in the view.

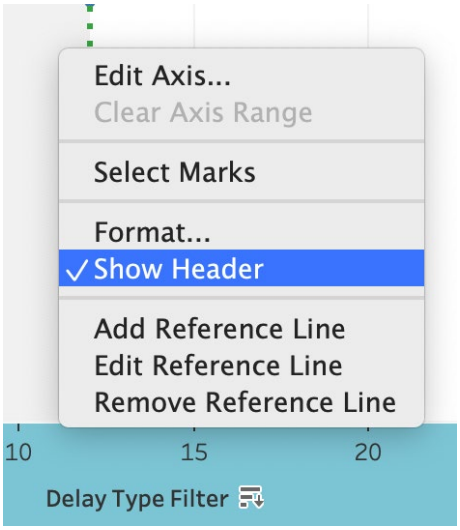

Figure 14.58: Show/Hide Header – 2

8. Add a tooltip that displays the following information: **<Name (airlines.csv1)>** has **AVG**. **<Parameters.Delay Type>** of **<AVG(Delay Type Filter)>** minutes of flights originating from **<Parameters.Origin.Airport.P>**. It should look like the image below:

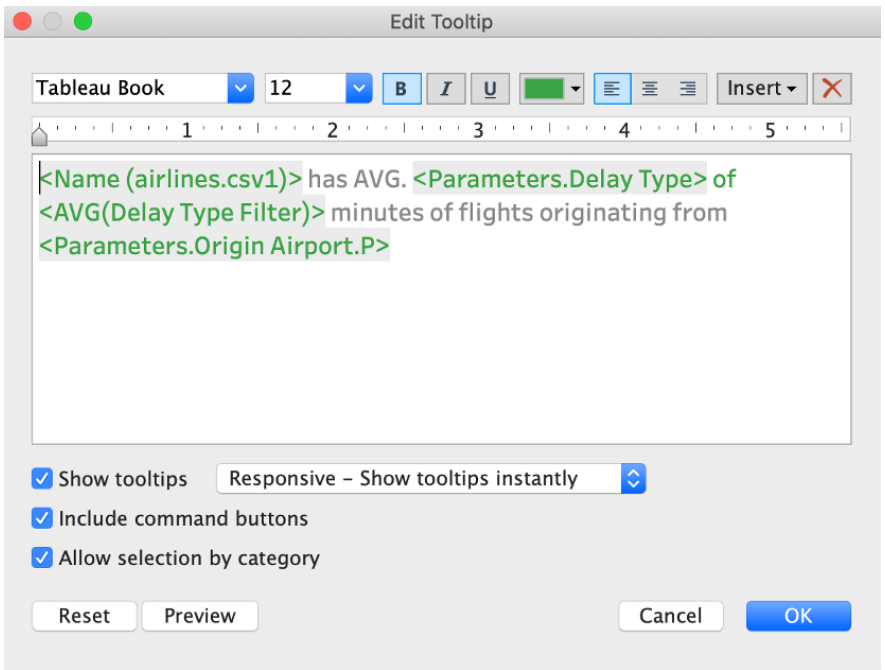

Figure 14.59: Editing Tooltip – 2

Your finished chart should look something like this when you hover over one of the airline bubbles:

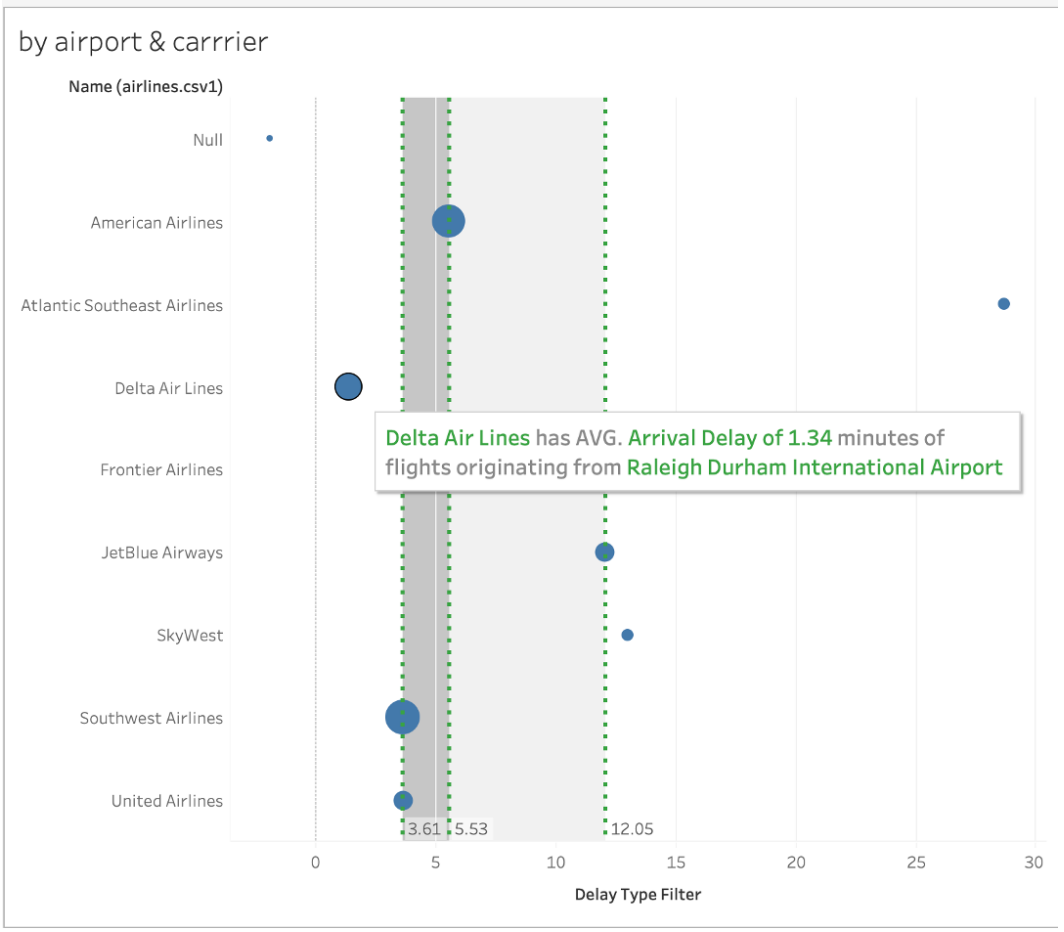

Figure 14.60: Final view of delay by airport & carrier

In this exercise, you created a worksheet that uses several filters and parameters to enable users to view airline performance for a selected airport, sized by the number of flights originating at the selected airport. This worksheet will be an integral part of your Performance Scoreboard dashboard.

The next chart will focus on flight delays by type at the airport level, which will provide the opportunity to see which airports are most subject to delay times. You want this chart to be sorted by your measure, rather than by airport name or code, since that will deliver the data in the simplest manner for a user to visually interpret. You want to see delay performance from worst to best, so sorting on the measure is your best option.

Now, to determine which chart type works best for this information, you could use another dot plot (as you did for the previous chart) and perform your sorting function. However, in this case, your data might best be displayed using horizontal bars, which can then be colored by delay time. It is possible to use multiple colors for the bars, but you should be cautious about this approach—especially since you are already sorting from highest to lowest delay. Color, in this case, might provide a distraction rather than adding insight to the chart.

In the following exercise, you will build this chart using common sense design and visualization principles.

EXERCISE 14.07: CREATING DELAY CHARTS: DELAYS BY DESTINATION AIRPORT

In this exercise, you will continue from *Exercise 14.6* and create a chart that displays delay data by the destination airport, sorted from worst to best performance. This will allow users to see where the problems are for a single origin airport, complementing your airlines and airports charts from the other two exercises:

1. Add a worksheet named **`Flight Delays by Destination Airport`**.
2. Drag the **`Delay Type Filter`** measure to the **`Columns`** card. Make sure it is set to **`AV`**G (average), and drag the **`IATA`** dimension from the **`airports.csv1`** folder (origin) to the **`Rows`** card.

3. Drag the second **IATA** dimension from the **airports.csv** folder (destination) to the **Rows** card to the right of the first **IATA** dimension. Note that RDU is pre-selected because that is the default airport for the **Origin Airport.p** parameter that you created previously.

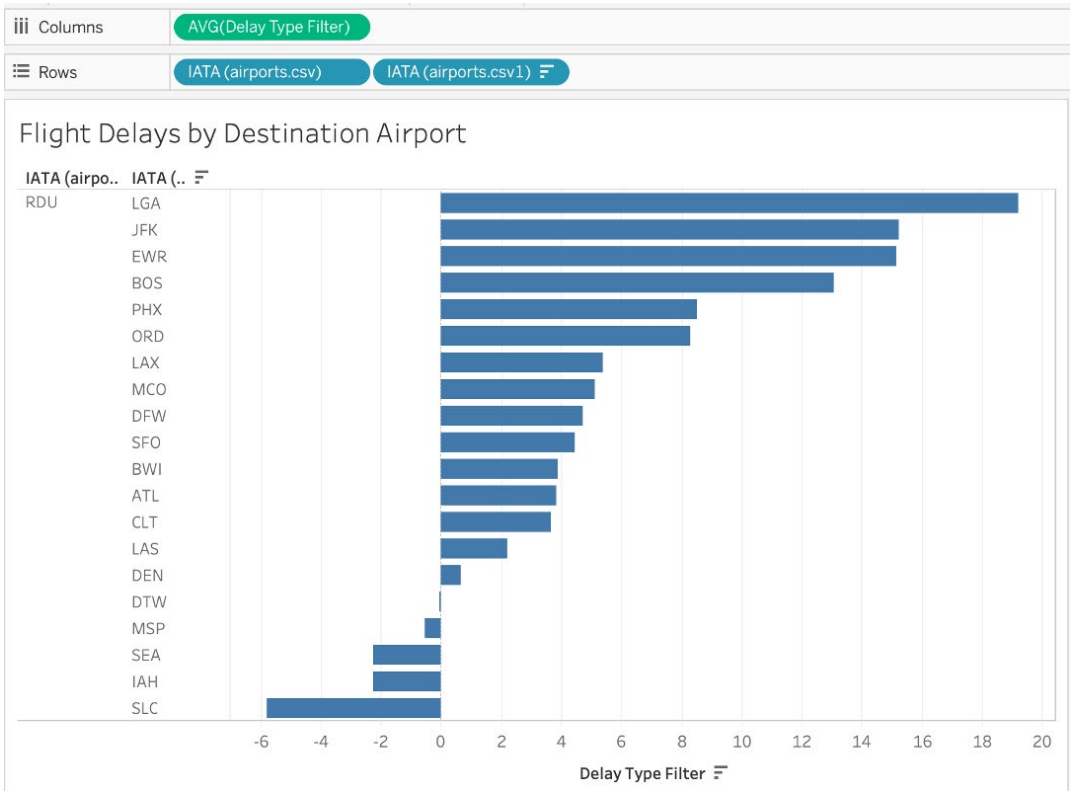

Figure 14.61: Flight Delays by Destination Airport

4. Verify that all four previous filters, **`[Origin Airport]`**, **`[SUM(Number of Records)]`**, **`[Callsign]`**, and **`[WEEK(FL_DATE)]`** are already present on the **`Filter`** card. If so, add a filter for **`[Destination Airport]`** and set the **`Destination Airport`** filter to show the Top 20 airports based on the **`SUM(Number of Records)`** measure from the origin airport.

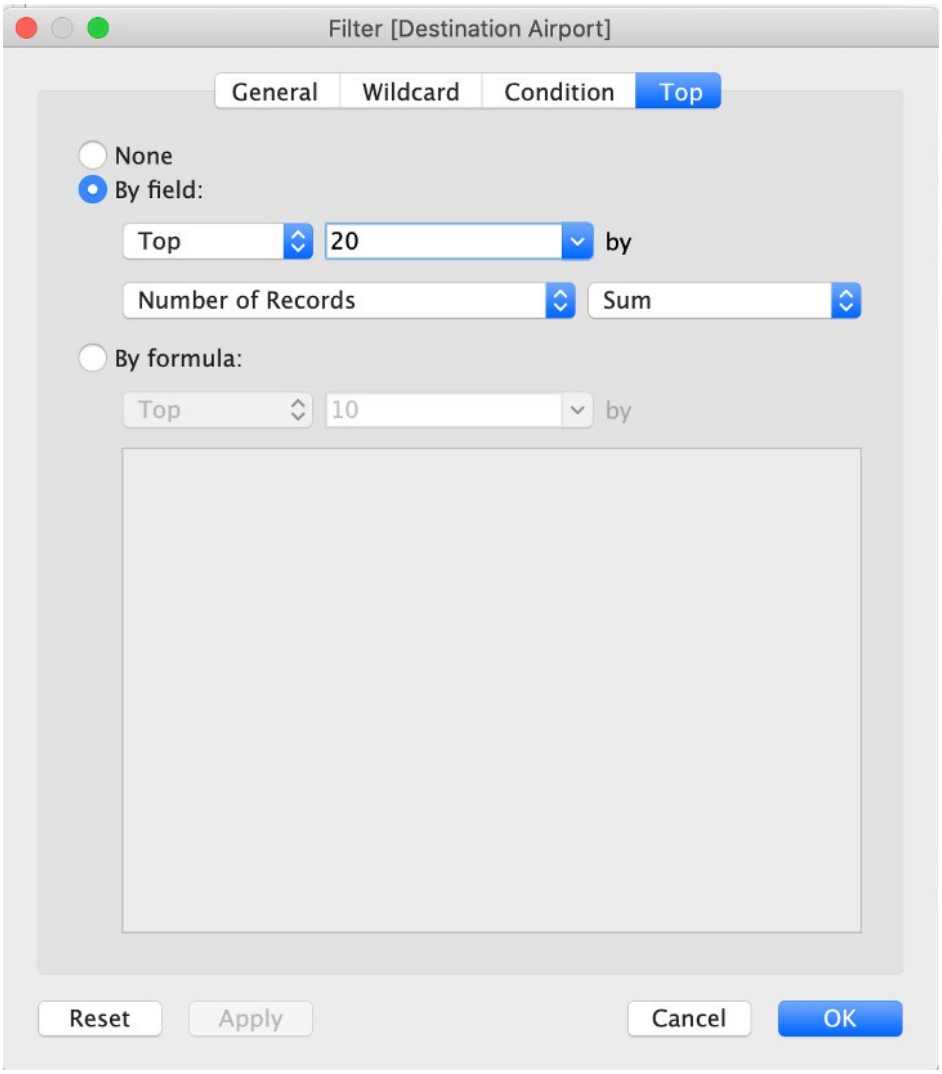

Figure 14.62: Showing Top 20 by Destination Airport

5. If Tableau does not create a horizontal bar chart by default, you can manually change the chart type to **`Horizontal Bar`** from the **`show me`** menu.
6. Drag the **`Delay Type Filter`** measure to the **`Color`** card in the **`Marks`** pane:

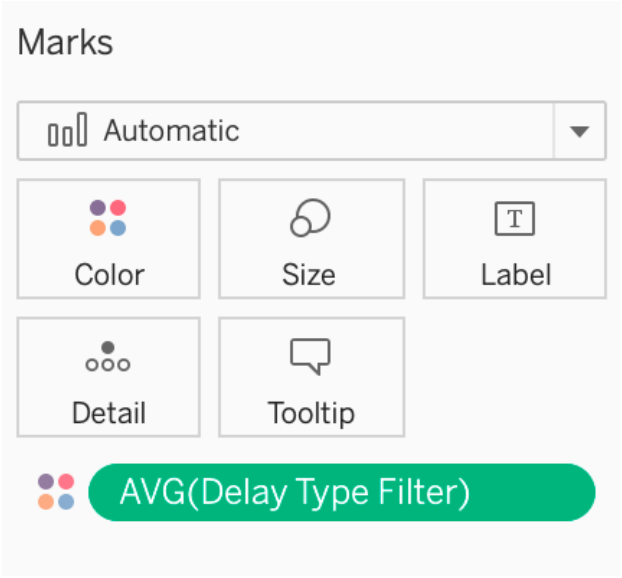

Figure 14.63: Adding AVG(Delay Type Filter) on the Color Marks card

7. You want to show only two colors, so edit the colors by clicking on the **`Color`** marks card and using **`Stepped Color`** to select **`2`** steps, with 0 at the center. Set your left color to blue and the right color to red:

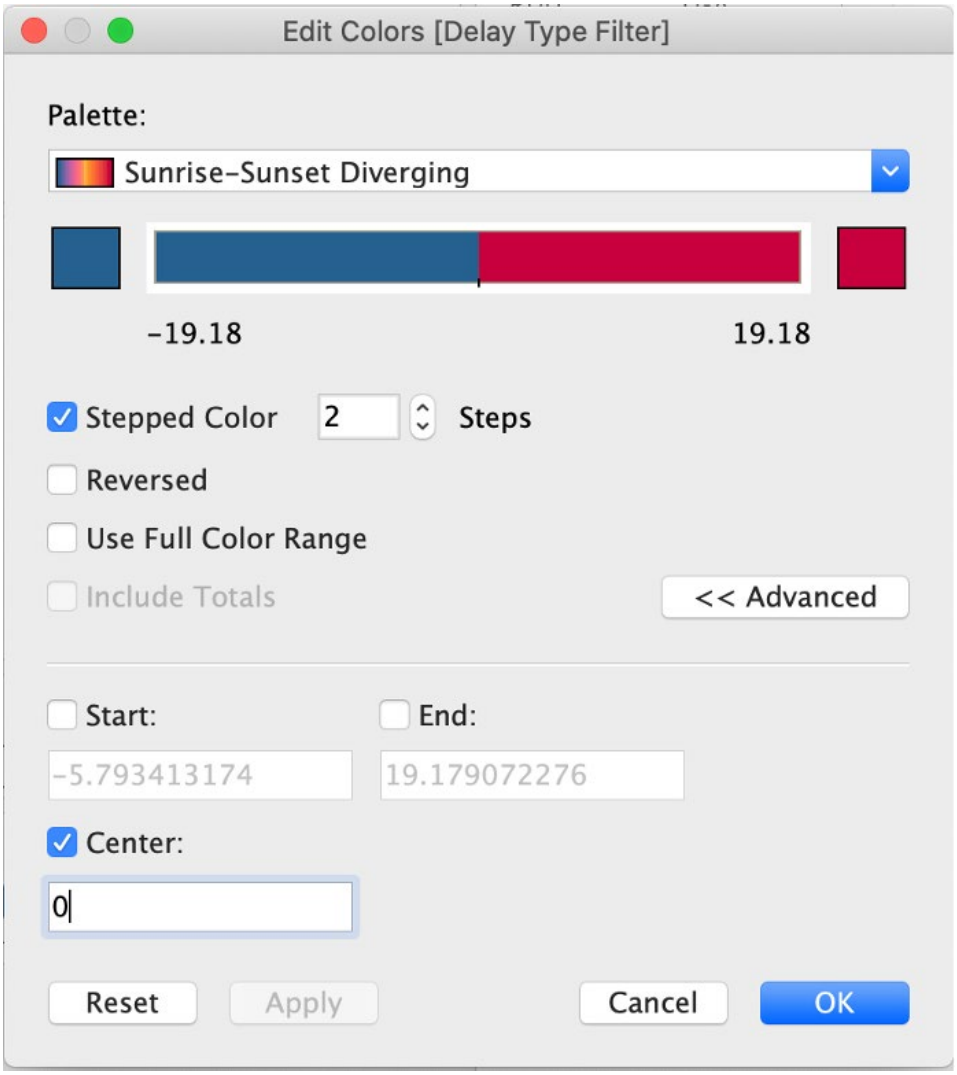

Figure 14.64: Editing color settings

8. Drag the **`IATA`**, **`Origin Airport`**, and **`Destination Airport`** fields to the **`Detail`** or **`Tooltip`** card in the **`Marks`** pane so that you can use them for tooltips in the next step:

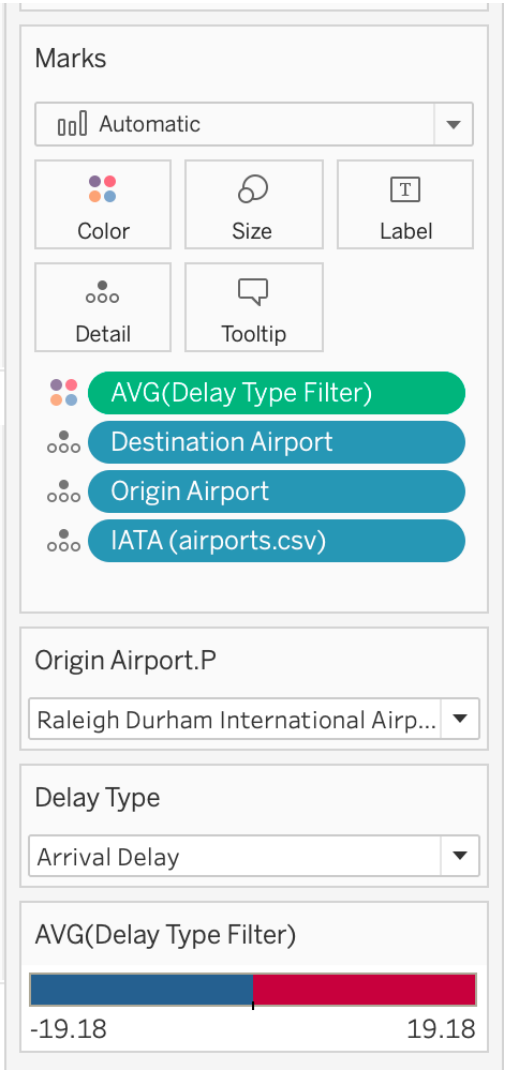

Figure 14.65: Adding IATA, Origin Airport, and Destination Airport to the detail card

9. Create a tooltip by clicking on **`Tooltip`** on the **`Marks`** card and use the following text: Flights from **`<Parameters.Origin Airport.P>`** to **`<Destination Airport>`** averaged **`<AVG(Delay Type Filter)>`** minutes **`<Parameters.Delay Type>`**.

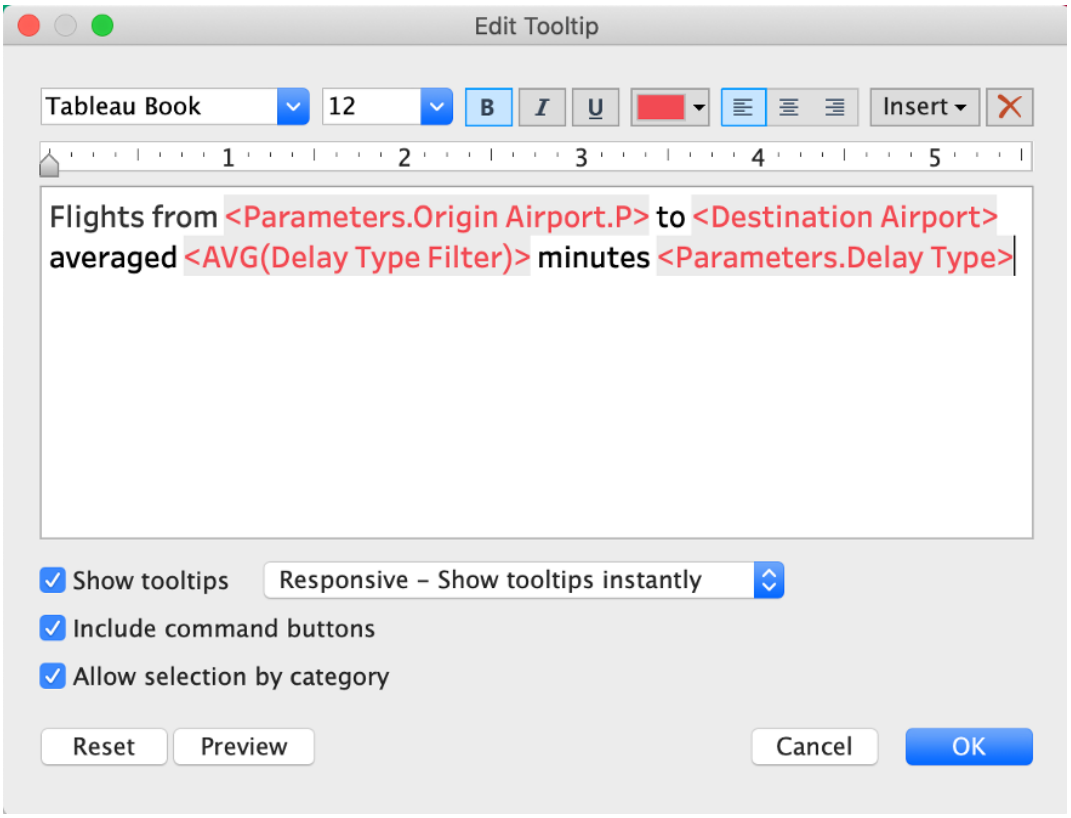

Figure 14.66: Editing the tooltip for the Delays by Destination Airport view

Here's what your finished chart should look like:

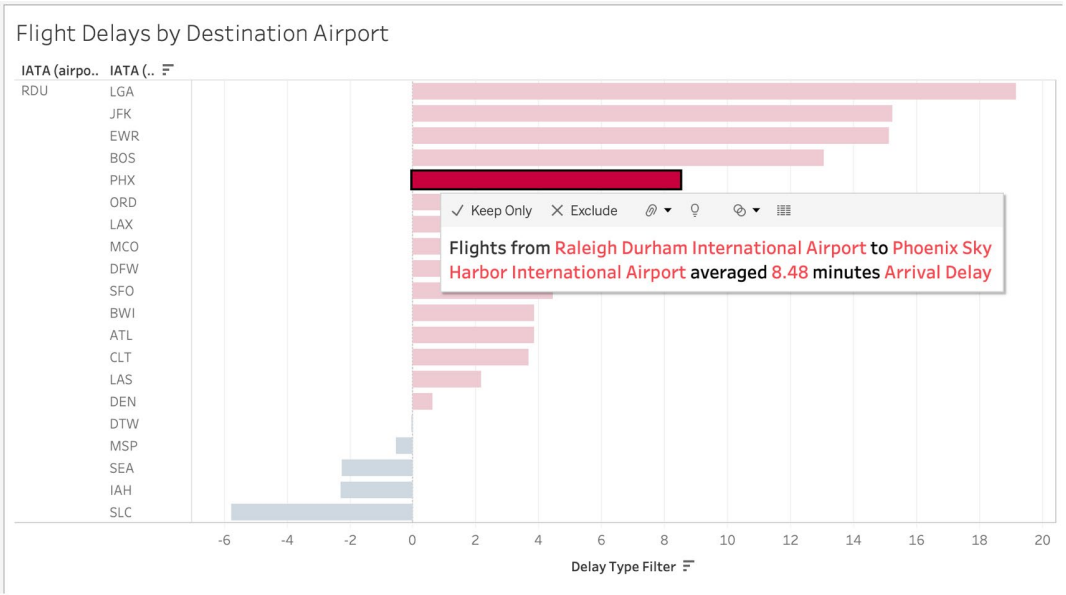

Figure 14.67: Final Flight Delays by Destination Airport view

The output above shows the worst flight delays from Raleigh Durham International Airport. For example, RDU to PHX has an 8.5-minute average **`Arrival Delay`**.

In this exercise, you created a chart that displays delay times from your origin airport to the 20 most popular destination airports and sorted and colored the data by delay times. This worksheet will be included in the Performance Scoreboard dashboard in the final section.

The next worksheet will focus on aggregate performance metrics displayed in a scoreboard type of text format. These types of displays are frequently included at the top of a dashboard page, as they provide easily digestible summary information typically aggregated at the sum or average levels. This then creates a natural flow for the dashboard, where users can quickly look at the aggregate data before navigating to the details included in the charts below.

In many cases, you will build each of these summary panes using a single measure, and then repeat the process several times. This works well when you have just two or three of these panes but starts getting more challenging from a design perspective when you wish to display more than a few aggregate measures. In this case, there are six distinct delay types you are tracking, so adding each as a single item can be cumbersome. This would also mean six individual worksheets to feed the dashboard (one per measure).

The solution then is to create a single crosstab type of display using all six measures in a single worksheet. You can then apply uniform styling and a common tooltip to provide a great deal of synergy to the process, as you will do in the following exercise.

EXERCISE 14.08: CREATING AN INDIVIDUAL AIRPORT DELAY MEASURE SCOREBOARD

As mentioned in the previous section, the goal of this exercise is to use only one worksheet to create six scoreboards using crosstab and measure selection. This scoreboard will display the average delay across your six delay metrics for the **`Origin Airport`** that your end users select using the **`[Origin Airport.p]`** parameters.

Perform the following steps to complete this exercise:

1. Add a new worksheet and name it **`Delays by Origin Airport`**.
2. Drag the **`Measure Names`** dimension to the **`Columns`** card, then right-click and click on **`Filter`** as shown below:

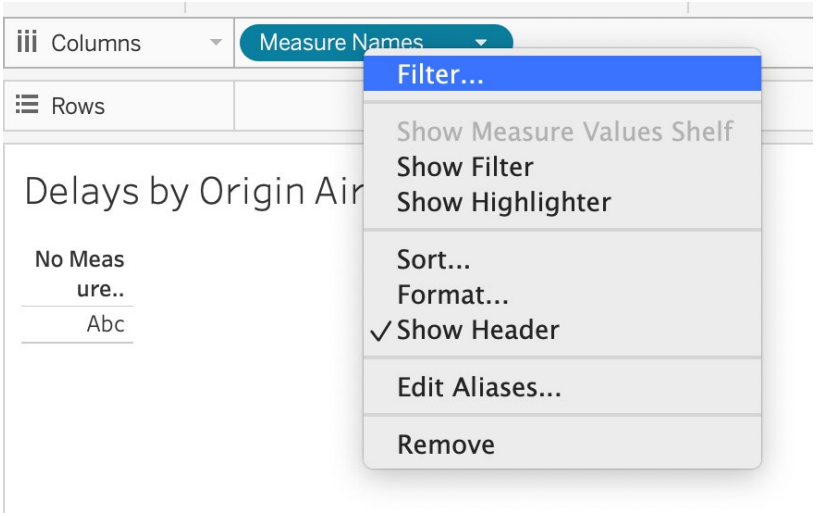

Figure 14.68: Using Measure Names filters for the delay scoreboard

3. Select only those six delay metrics that have been used in the other exercises in this chapter: the **`Arrival Delay`**, **`Carrier Delay`**, **`Departure Delay`**, **`Late Aircraft Delay`**, **`Security Delay`**, and **`Weather Delay`** measures.

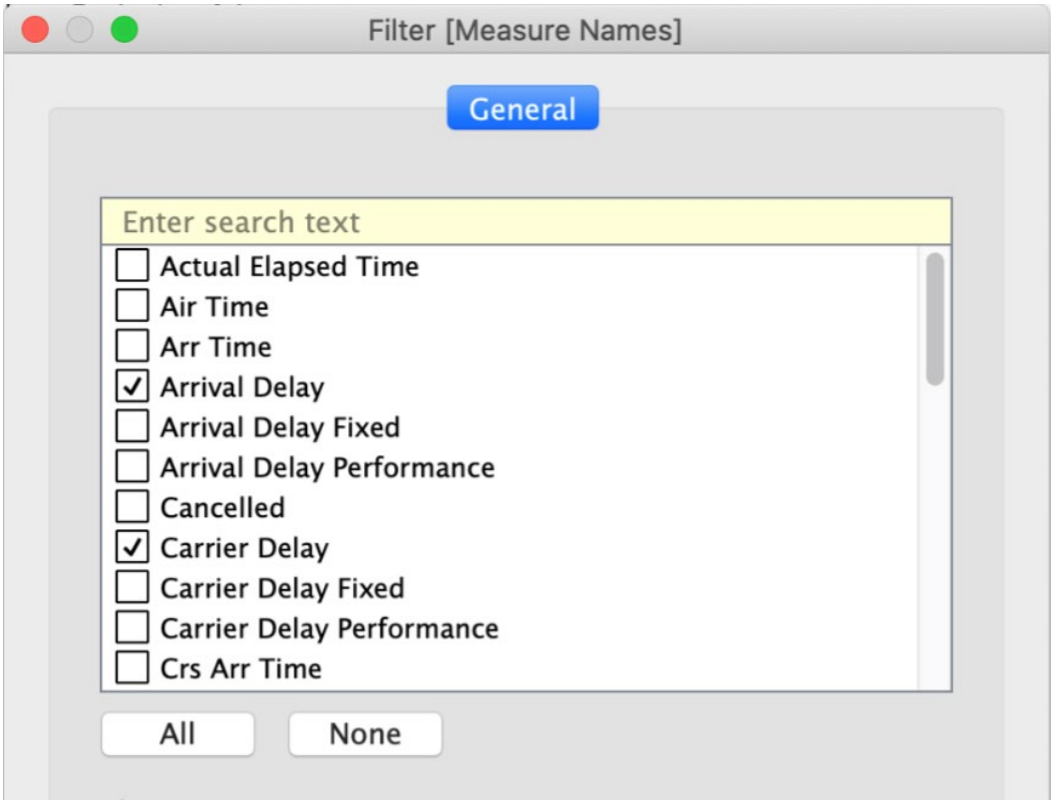

Figure 14.69: Selecting measures for our view

4. You should ideally have all your four filters loaded onto the worksheet the moment you create the new sheet. Verify that these filters were loaded onto the new sheet you created.

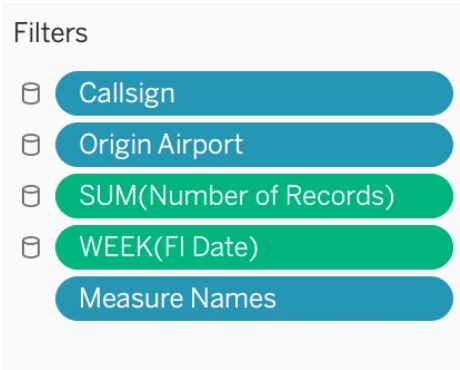

Figure 14.70: Adding all previous filters

5. Drag **`[Measure Values]`** from the **`DATA`** pane onto the **`Text`** Marks card to load the metrics values of these dimensions onto the view. Then, drag [**`Measure Names`**] onto the **`Text`** card since your goal is to convert this crosstab into a scoreboard (covered in the next step).

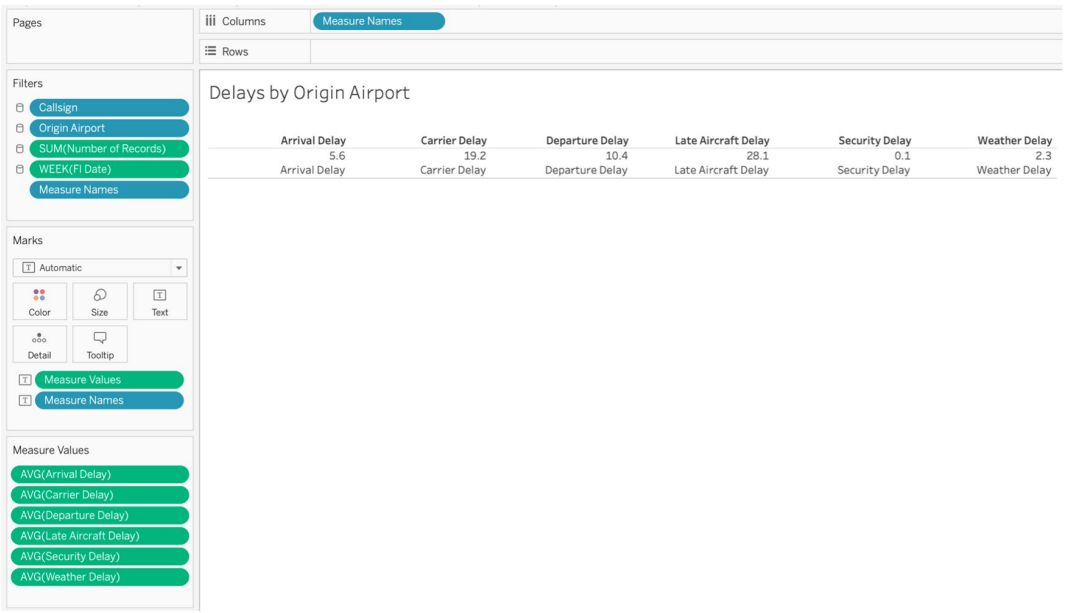

Figure 14.71: Add [Measure Values] to the Text Marks card

6. Edit the **`Text Marks`** using the **`Text`** card to make it look more scoreboard-driven instead of a crosstab. Click on the **`Text`** card and edit the marks using the **`Text`** card by increasing **`<Measure Values>`** to a font size of **`24`** and keeping **`<Measure Names>`** as **`9`** in the **`Marks`** pane. Also, add **`min`** at the end of **`<Measure Values>`** to signify to the end users that the values are in minutes.

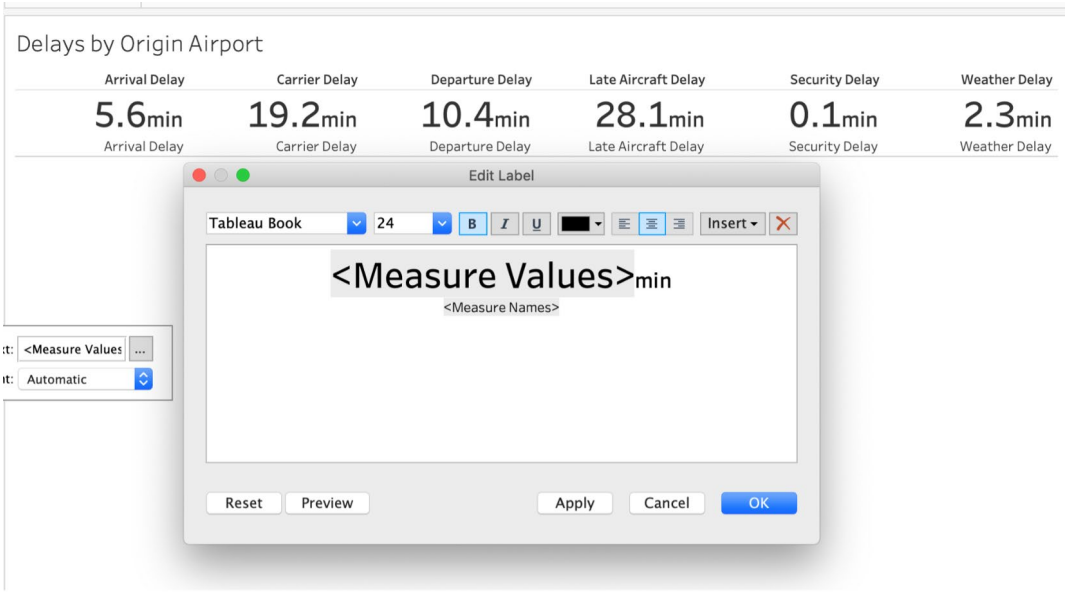

Figure 14.72: Editing a label

7. It can be difficult for end users to tell which airport is selected if they don't focus on **`[Origin Airport.p]`**, so to make it easier, add the parameter value into your worksheet title with the following text:

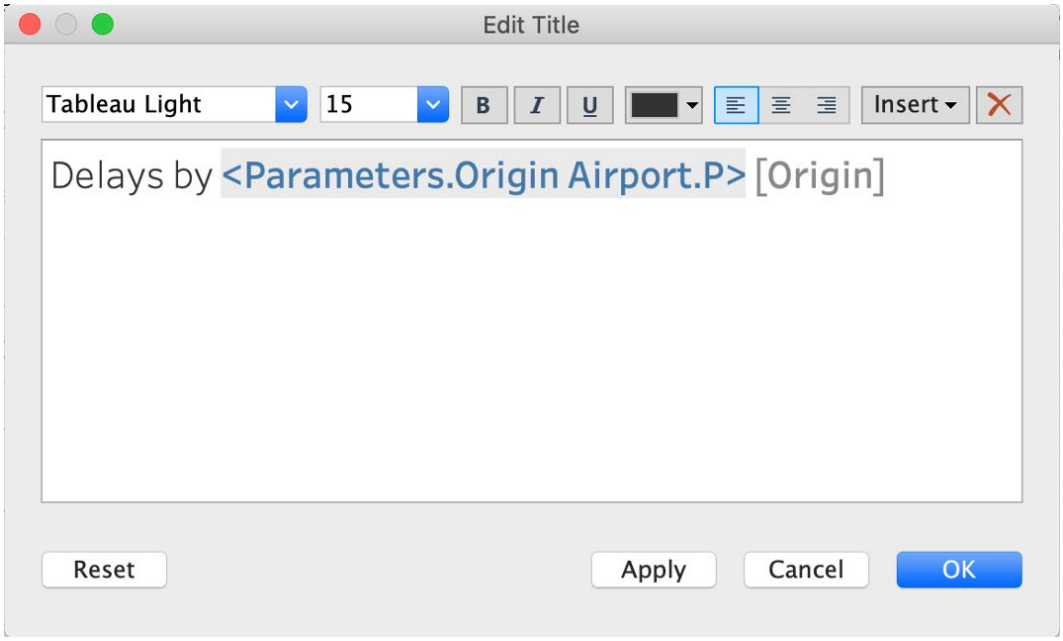

Figure 14.73: Editing the title

Here is what your output will be:

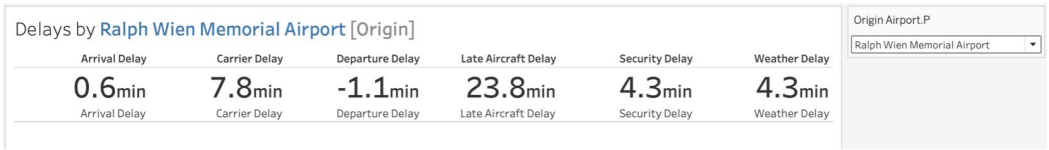

Figure 14.74: Final Delays by <Airport Name> Origin worksheet

The preceding scoreboard displays your six KPIs for Ralph Wien Memorial Airport.

In this exercise, you created a scoreboard to track six delay measures in one worksheet for a user-selected **`[Origin Airport.p]`** parameter. This sheet will become an essential component of your dashboard in the next section.

There is one more worksheet to build as part of your dashboard. This new sheet will allow your end users to compare their **[Origin Airport.p]** airport to the performance overall across the dataset. You will color code these performance metrics to make it even easier for the end users to grasp the analysis. You will achieve this by utilizing the **fixed** calculated fields that you created earlier in the exercise, which essentially gave you the average delay across categories across your dataset.

In many cases, dashboards are created with little to no context with which to judge whether a set of numbers reflects good or bad performance. You have already created charts that show changes over time, comparisons by destination airport, and comparisons by airline within a single airport. But you have not yet created something to put this into context.

Using the crosstab approach, you will now use a set of comparative measures to show how a single airport compares to all airports across each of the six measures in the following exercise.

EXERCISE 14.09: CREATING AN INDIVIDUAL AIRPORT PERFORMANCE VERSUS OVERALL SCOREBOARD

In this exercise, you will create a delay measure scoreboard detailing average delay times for each of the six measures you are tracking but, instead of showcasing the scoreboard for a single airport, you will compare the single airport performance with the overall average and color code their performance:

1. Add a new worksheet and name it **Performance vs. All Airports**. Also show the **[Origin Airport.p]** parameter control by right-clicking the parameters and selecting **Show Parameter Control**.
2. Drag the **<Measure Names>** dimension to the **Columns** card. Also, drag **<Measure Values>** onto the **Text Marks** card.
3. Filter on **Measure Names** to include only the **Arrival Delay Performance**, **Carrier Delay Performance**, **Departure Delay Performance**, **Late Aircraft Delay Performance**, **Security Delay Performance**, and **Weather Delay Performance** measures:

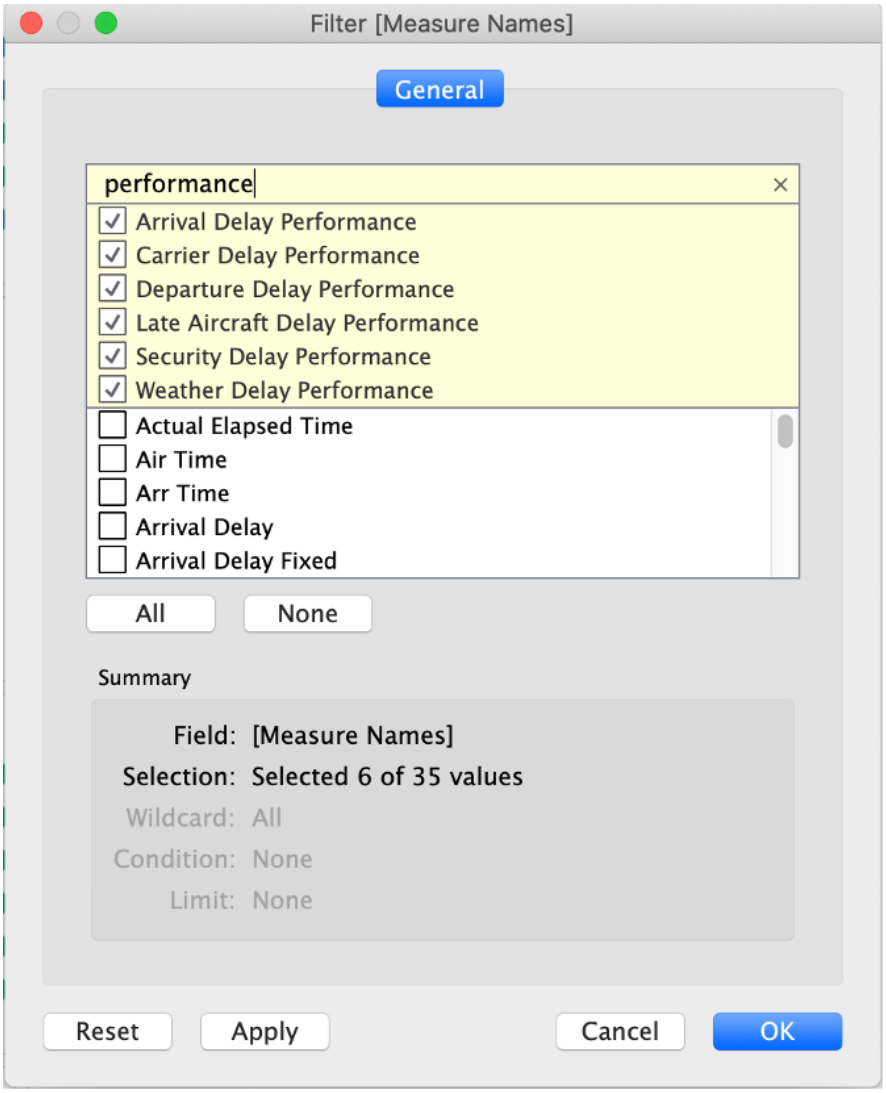

Figure 14.75: Using Measure Name filters for the performance scoreboard

4. You should ideally have all four filters loaded onto the worksheet the moment you create the new sheet. Verify that these filters were added under the Filters shelf.

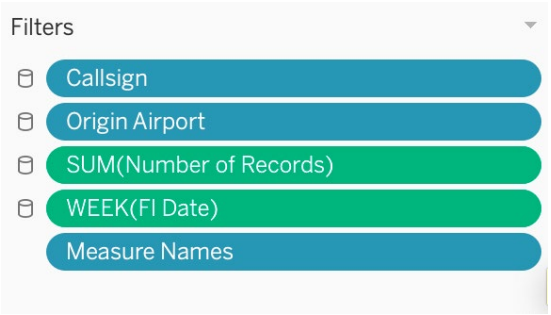

Figure 14.76: Validating that all previous filters are added to the view

5. Edit the **`Text Marks`** using the **`Text`** card to make it look more scoreboard-driven instead of a crosstab. Click on the **`Text`** card and edit the marks using the Text card by increasing <Measure Values> to a font size of 24. Also, add performance level on the second line as shown:

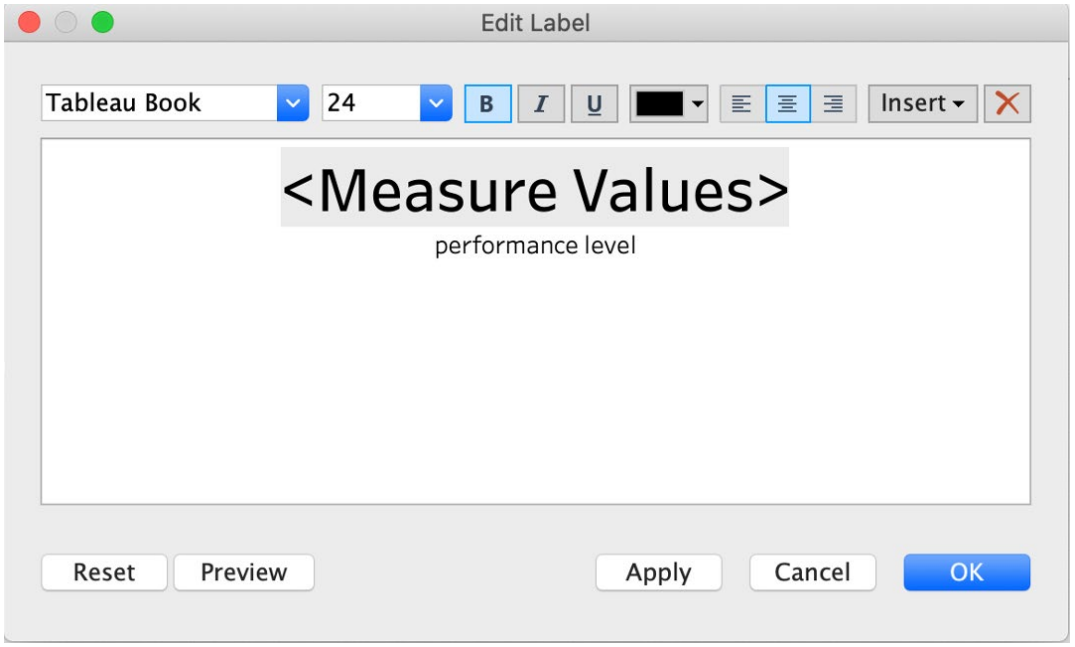

Figure 14.77: Editing a label for the performance scoreboard

6. Add the **`Measure Values`** measure to the **`Color`** card in the **`Marks`** pane. Use **`Stepped color`** with **2** steps and set the center value as 1. Set the left color to red and the right color to blue.

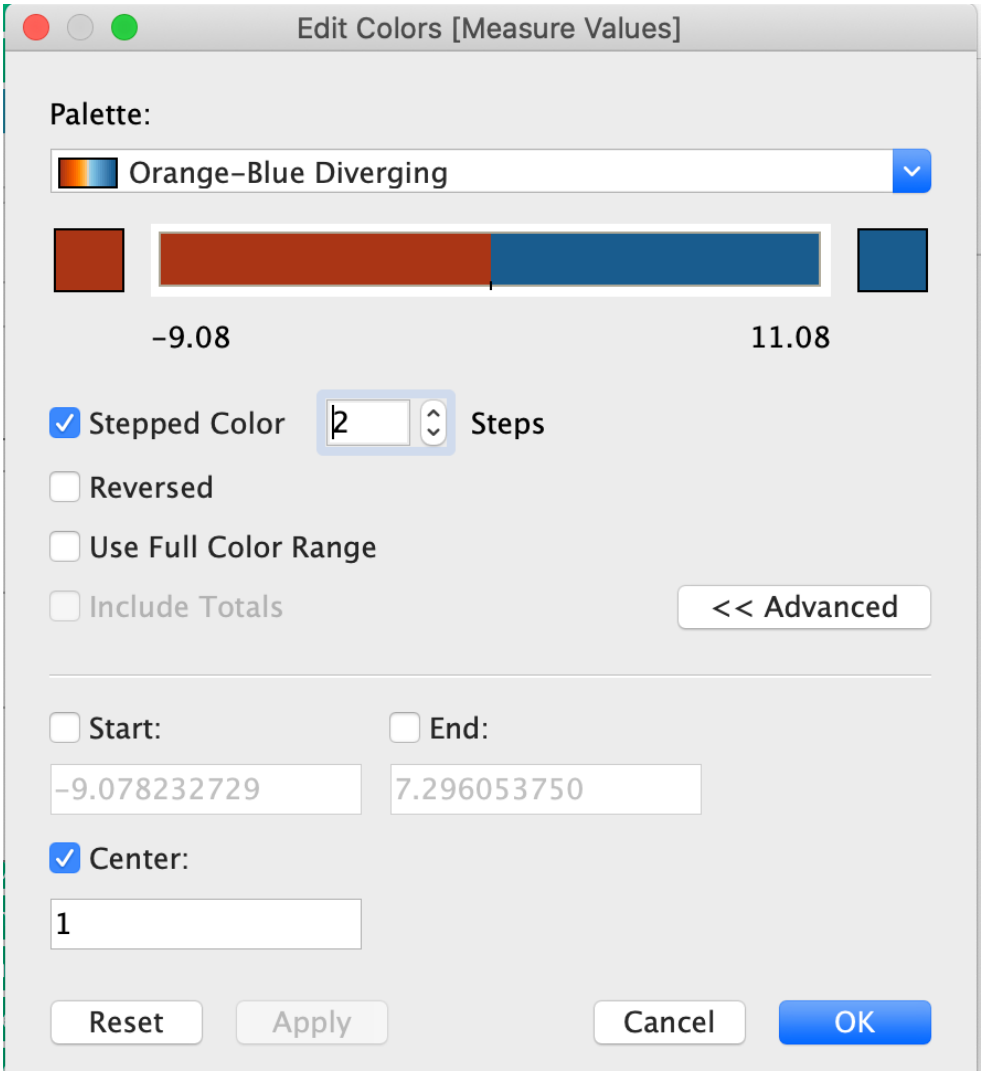

Figure 14.78: Editing colors for the performance scoreboard

Your output will be the following:

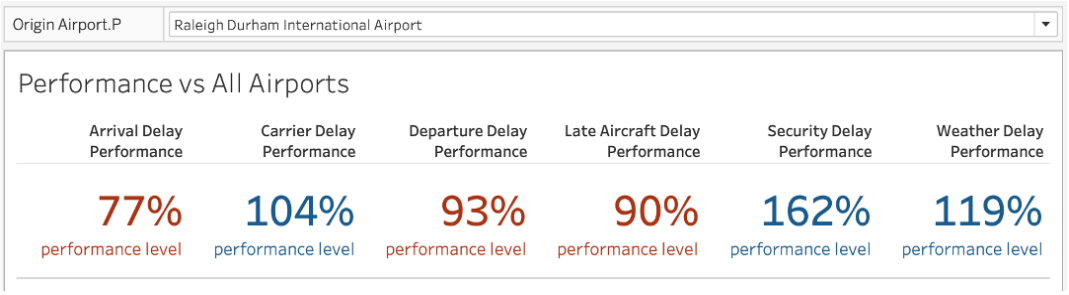

Figure 14.79: Final Performance vs All Airports scoreboard

The preceding output shows the performance of **`[Origin Airport.p]`** selected by the user, which in the screenshot shows **`Raleigh Durham International Airport`**. If the performance of a single airport was worse than the overall average (less than 100% in this case), the metric is colored red (meaning the airport performed worse than the overall average across the dataset); while if the performance was better than the overall average (greater than 100% in this case), the metric is colored blue. This quick way to compare actual metrics versus the average allows the end user to get contextual information about overall performance, which can be missing in a lot of reports/dashboards.

ANALYZING THE DATA

Now that the key worksheets have been created and optimized, it is important to take the time to visually interrogate the data to make certain the charts are displaying as expected and the data makes sense. In theory, you could have done this second requirement earlier in the process, but performing a visual inspection of the data will often alert you to patterns or inconsistencies much more efficiently than analyzing data in a crosstab or other format where little context is provided.

Visual inspection is easily done in Tableau, especially when you have taken the time to build informative charts with filters. You can simply click through the data on a worksheet by selecting different filter or parameter options, watching the data refresh in real time, especially if we have created an extract. Another benefit of spending time on visual analysis is that you become familiar with both the Tableau workbook and the underlying data, making it easy to solve user questions.

Take some time to make sure each worksheet is functioning as expected, and that filters are applied consistently before moving on to creating the dashboard.

Here are a couple of visual inspections you can do:

- **Performance vs All Airports**: Select a different airport from the **`[Origin Airport.p]`** parameter and check whether the percentage of the performance is updated. For example, the following screenshot shows the output for Chicago O'Hare International Airport:

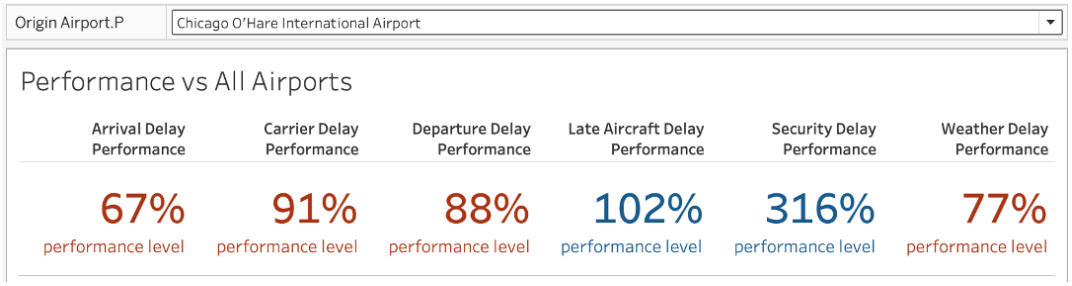

Figure 14.80: Performance vs All Airports view

- **Flight Delays by Destination Airport**: Since you selected Chicago O'Hare International Airport in the [Origin Airport.p] parameter, all other worksheets that use that parameter should be automatically updated as parameters are applied across the whole Tableau workbook and not to a specific worksheet. The following screenshot shows the output of Flight Delays by Destination Airport:

Figure 14.81: Flight Delays by Destination Airport view

You can continue to verify other visual cues across all the developed worksheets. This wraps up this section on creating individual worksheets. Now it is time to combine these into one dashboard, which we will do next.

BUILDING A DASHBOARD

A dashboard provides a summary version of your worksheets, with each one sized to fit on a single page (normally). This presents you with some challenges in laying out multiple instances of full-page worksheets into a single view, so you need to be intelligent in how you approach this task. You previously highlighted which of your worksheets will be part of the dashboard, as it would be difficult to present them all without cluttering the page and confusing users. This section will cover formatting and designing the dashboard, before proceeding to adding content and building interactivity.

FORMATTING THE WORKBOOK

Tableau makes it easy to set global values for a workbook, which will then propagate to any dashboards. It is helpful to set font and chart settings at this level, knowing they can be adjusted at the worksheet level as needed.

A good practice with fonts is to use styles and sizes that support your dashboard while not overwhelming the information in your charts. Depending on the font style, worksheet titles are often most effective between 11 and 15px and should be in a darker color (unless you have a dark dashboard background). A general rule for visual presentation is to use dark on light or light on dark and avoid colors in the mid-range, at least for title purposes.

Worksheet and tooltip fonts should be sized smaller than your worksheet titles, generally. This will help the dashboard flow naturally from a visual perspective. One last note on choosing a default font: many fonts will not align well using numeric data. If you have row or column values that use numeric codes, you may wind up unhappy with the sloppy appearance delivered by many font types. As with many Tableau features, this is easy to test for by toggling through different fonts from the **`Format`** > **`Workbook`** menu.

Recent versions of Tableau have introduced default chart formatting into the **`Format Workbook`** pane. These allow you to set defaults for grid lines, axis ticks, trend lines, and more. If you want a consistent baseline style that applies across the workbook, start here. You may wind up adjusting settings based on individual chart styles but selecting default options upfront may save some time.

LAYING OUT THE DASHBOARD

Laying out a Tableau dashboard is a mix of art and science. You should use best practice data visualization methods as a starting point for how your finished dashboard should appear to the user, and then work within Tableau to make that happen. It isn't quite a drag and drop, WYSIWYG process, but following some basic layout rules will make things as seamless as possible. You have already seen the finished dashboard at the start of the chapter and how it follows a simple layout flow from top to bottom:

- Title at the top of the page
- A summary scoreboard of delay measures
- A second scoreboard with performance measures aligned with the first scoreboard
- A single wide chart showing historical trend data
- Side-by-side charts providing a higher level of detail at the bottom of the layout
- Filters and parameters along the right margin of the content

This is the flow you will use in the next section as you populate it with individual pieces of content.

ADDING DASHBOARD CONTENT

You begin with the title, a component that every good dashboard should contain. You use this to alert the user to what they can expect to find in the content below. Ideally, this title is variable, updating based on date or dimension selections. Note that this effect is most easily achieved using parameters; text objects do not easily work with filters or other content.

Here's what your title formula looks like:

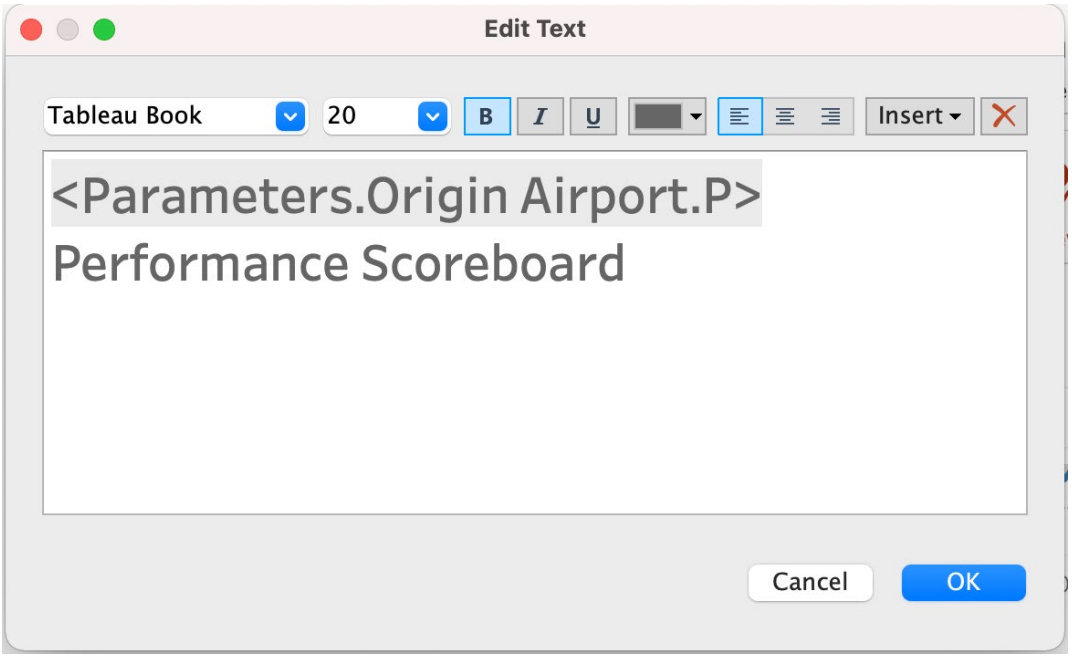

Figure 14.82: Editing the title for the dashboard

Here, you have manually entered the desired title and then added a variable component using the **`Origin Airport`** parameter. Now, every time a user selects a new airport, the dashboard title will automatically update. This is a simple yet powerful capability that will help users maintain context especially if they distribute print versions or screenshots of the dashboard.

As was discussed previously, dashboards typically flow from top to bottom and left to right. Using this principle tells you to locate summary measures near the top of the dashboard, reserving more detailed charts for below the summary sections. Content will be added in the following order:

- **`Delays by Origin Airport`** worksheet
- **`Performance vs. All Airports`** worksheet
- **`By Airport Trend`** worksheet
- **`By Airport and Carrier`** worksheet
- **`Flight Delays by Destination Airport`** worksheet
- **`Origin Airport`** parameter

- **`Origin airport`** map
- **`Flight date range`** filter
- **`Delay Type`** parameter

Once you have added these elements, our job is to make sure they are displayed properly and provide the look and feel you are seeking for the dashboard. Finally, you will bring together all the worksheets that you have created previously in the next activity.

ACTIVITY 14.01: MERGING WORKSHEETS TO CREATE A DASHBOARD

In this activity, you will build the dashboard following a series of steps, adding content from top to bottom, as discussed in the previous section. Note that Tableau will frequently add second sets of filters and parameters as well as legends from the worksheet level. These can be removed at any point in the process by deleting the container in which they appear.

Perform the following steps to complete this exercise:

1. Add a new dashboard page and title it **`Airline/Airport Performance Scoreboard`**.
2. Set the size to **`1500 x 1200px`**.
3. Add the title text, labeling it **`Performance Scoreboard`** and reference the **`Origin Airport`** parameter to complete the title.
4. Add a vertical container that occupies the remainder of the space below the title.
5. Add a second vertical container to the right of the first one. Size it to accommodate filters and parameters.
6. Add the **`Delays by Origin Airport`** worksheet to the large vertical container.
7. Add the **`Performance vs. All Airports`** worksheet beneath the prior worksheet.
8. Add the **`Carrier Delay by Airport Trend`** worksheet beneath the two scoreboard items.
9. Add a horizontal container inside the large vertical container. Add the by Airport Carrier and **`Flight Delays by Destination Airport`** charts.

10. Make sure your filters, parameters, and **`Airport Ma`**p sheet are all contained in the narrow vertical container to the right.

    This should result in the following output:

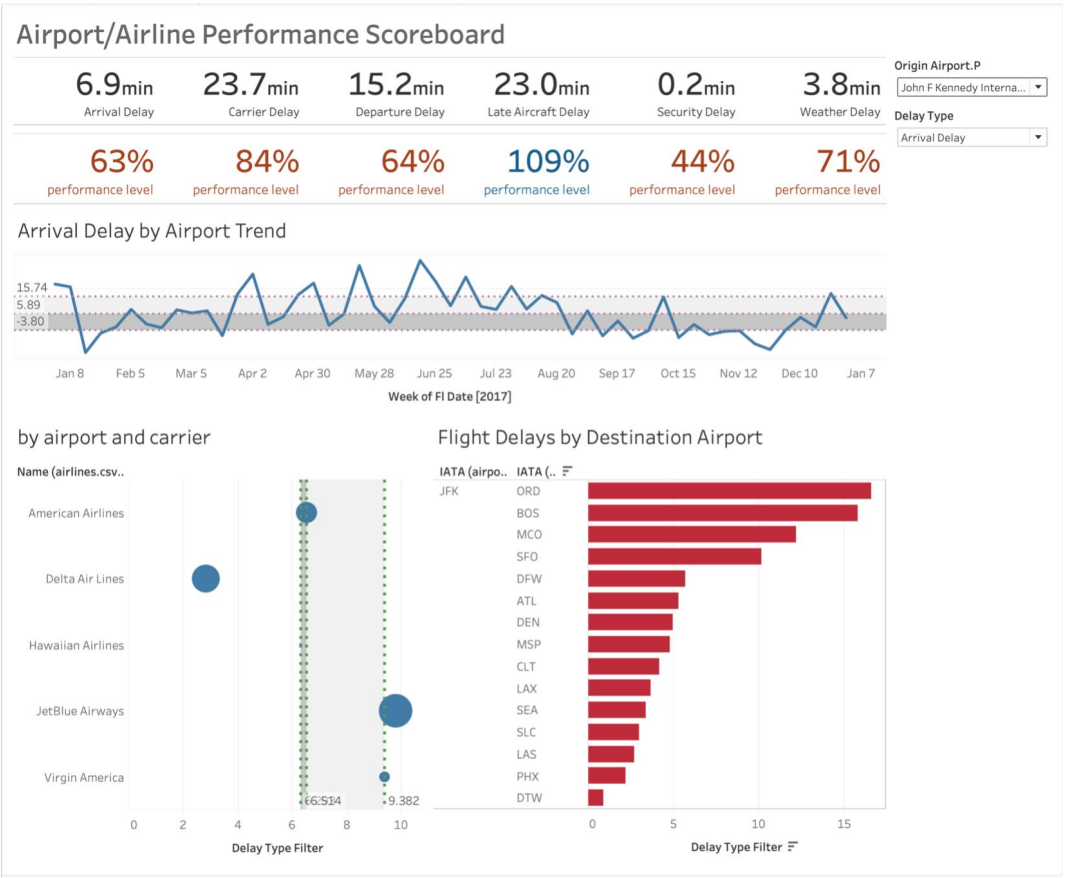

Figure 14.83: Expected output for Activity 14.1

> **NOTE**
>
> The solution to this activity can be found here: https://packt.link/CTCxk

FINISHING TOUCHES

With all functionality working as expected, you can move on to make cosmetic updates. Here's what you should look for:

- **Spacing**: Add blank objects to the dashboard if charts need spacing around them.
- **Fonts**: Are your titles too large, too small, too bold, or not bold enough? Edit them individually or by using the **`Format`** > **`Workbook`** options.
- **Charts**: Are your gridlines too prominent? Edit at the chart level or by using **`Format`** > **`Workbook`** to update all of them.
- **Appearance**: How do your filters and parameters appear to the user? Too large, too small, or just right? Remember that you don't want to distract users from the content, so size these elements accordingly.
- **Colors**: Is your color scheme appealing and intuitive? Edit as needed at the worksheet level, but try to be consistent across worksheets that appear within the dashboard.

If you feel comfortable with all these elements and how they work together, then you should have a dashboard that is highly effective at communicating information to the user.

SUMMARY

In this chapter, you worked to create an effective and powerful dashboard using multiple data sources. Along the way, you added calculated fields, parameters, filters, and actions to help users understand the wealth of information in the data. You also used several types of charts to lend insight into the dataset, including boxplots, dot plots, trend charts, vertical bar charts, and performance scoreboards.

You utilized data visualization best practices to make certain you were communicating information to the user without adding distracting visual elements. The intelligent use of spacing, fonts, colors, and chart types allowed you to design a dashboard that is easy to use and interpret while still presenting multiple layers of data.

This also completes the book. In the initial chapters of this book, we went through the basics of Tableau, considering data and its visual elements for reporting and dashboarding while also covering the essential skills we need to learn to create effective data visualizations. We then briefly covered Tableau Prep and performed data manipulations such as pivots, groupings, and aggregations.

The next couple of chapters solely focused on data exploration in Tableau, guiding you through exercises and activities in which you performed comparison and composition in Tableau and created your first chart. You then went on to create advanced chart types in the *Chapter 5, Distribution and Relationships* chapter, creating dual-axis and quadrant charts. This chapter set the foundation for exploring advanced settings and eventually led to geographical data, such as the NYC taxi data, which you utilized to create custom maps and more.

For the next three chapters, you focused on the analysis component of Tableau, wherein you created your first calculated fields and explored table calculations in contrast to calculated fields. You also learned how to write logical statement formulas as well as manipulating strings and dates, and thereafter encountered one of the most important concepts in Tableau in depth: LOD calculations. You eventually ended your journey by diving deep into dashboards and storyboards and the addition of various interactive elements to the dashboards, including but not limited to filters, sets, parameters, and actions such as Set, URL, and Parameter. All of this finally culminated in this current chapter, which consisted of a case study regarding the utilization of multiple data sources and the practical implementation of all the skills you learned throughout the course of this book.

Having walked you through Tableau and all its exciting features, the next step now is to continue exploring Tableau and getting better at it! To stay on top of Tableau and its development, you can follow Tableau on its social channels or subscribe to its email newsletter and be informed when new features are launched. Good luck with your Tableau-driven growth.