Lab 14. Tableau Troubleshooting

In this lab, we'll cover the following recipes:

- Performance recording
- Performance troubleshooting and best practices
- Troubleshooting through log files

Lab 14. Tableau Troubleshooting

In this lab, we'll cover the following recipes:

- Performance recording
- Performance troubleshooting and best practices
- Troubleshooting through log files

Technical requirements

To follow the recipes outlined in this lab, you'll need to have Tableau 2019.x installed on your device.

Introduction

When working with Tableau, we should always strive to optimize the performance of our workbooks. Even if we develop astonishing dashboards with great functionalities, our users' experience will not be optimal if our workbook works too slow. Especially in business, time is essential. Our stakeholders often don't have time to wait for our workbook to load. Despite all the effort we put into the workbook, the overall impression of our work would be tainted. To achieve the best possible results, it's important to test the performance of the workbook during development, and address issues that may appear. In this lab, we'll learn some handy techniques for assessing and optimizing workbook performance, as well as for troubleshooting issues.

Performance recording

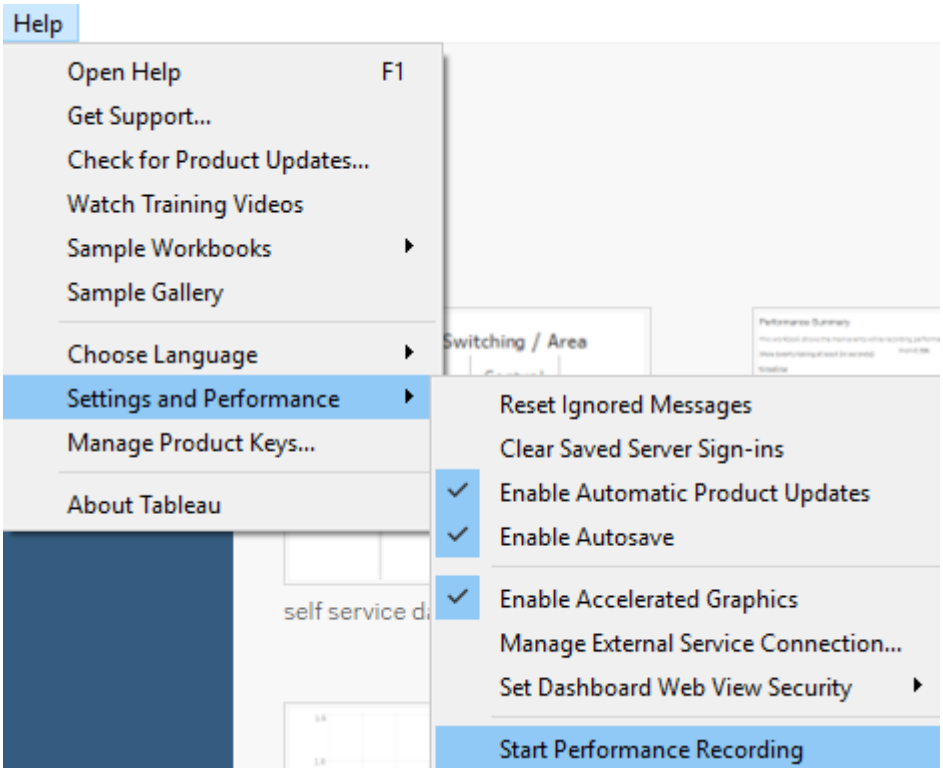
Anticipating possible performance issues, testing for them, and, if needed, solving solving them is important when creating a workbook. Poor performance can have a negative impact on the experience of our workbook users. In this recipe, you'll learn how to use Tableau's built-in performance-diagnostic tool: performance recording.

Getting ready

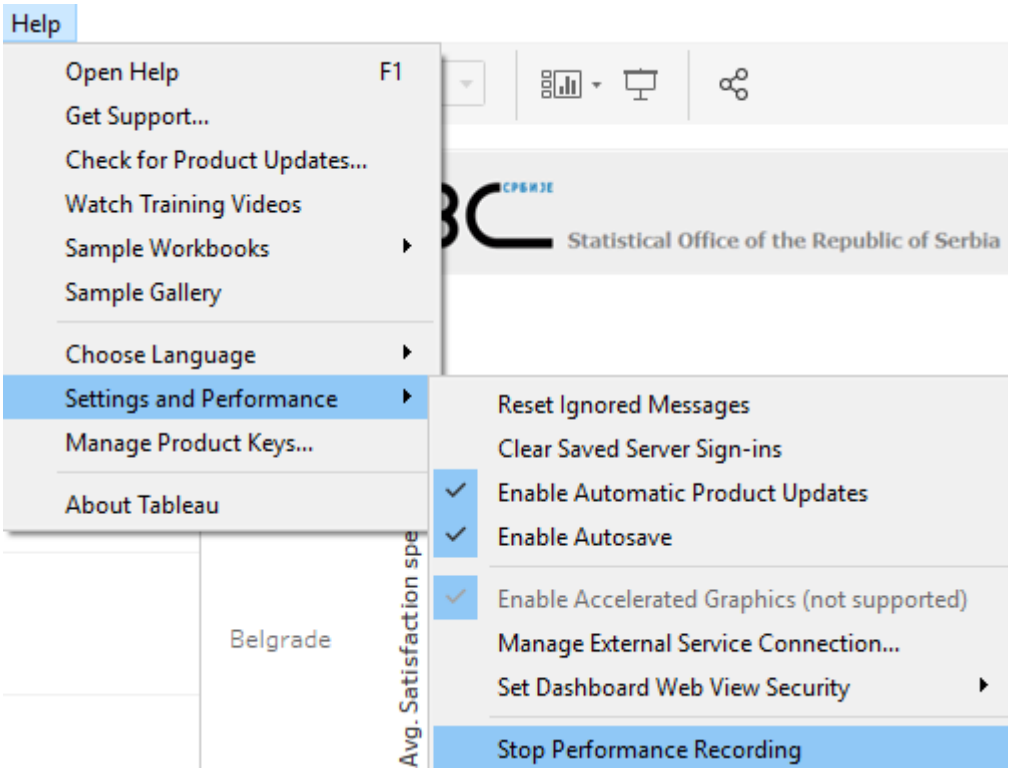
In this recipe, we'll test the performance of a dashboard. Follow the last recipe in *Building Dashboards* to create the dashboard we'll be working with in this recipe. When you're done, save the workbook to your device.

How to do it..

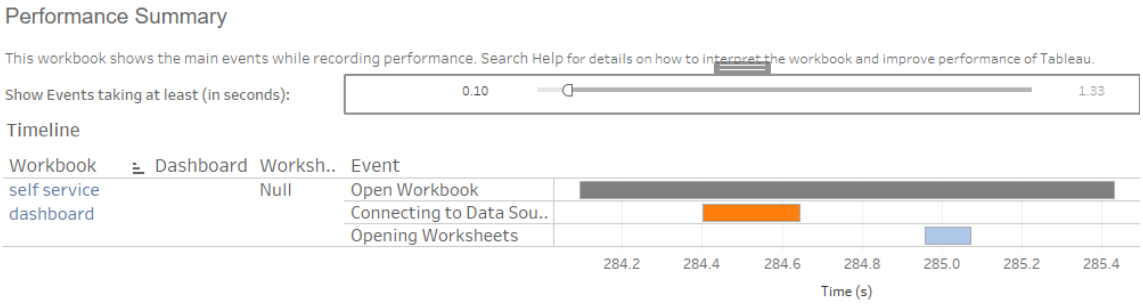
1. Launch Tableau.
2. In the main menu toolbar, select `Help`.
3. In the drop-down menu, go to `Settings and Performance`, and, from the additional drop-down menu, select `Start Performance Recording`:

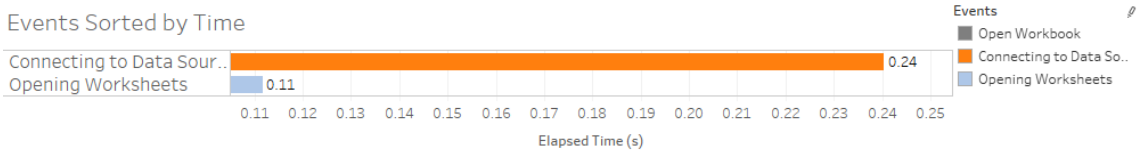


4. In the main menu toolbar, click on **`File`** and select **`Open`**... .
5. Navigate to the location on your device where you've saved the dashboard, select it, and click on **`Open`** .
6. Try playing around with the dashboard a bit---for example, filter different options.
7. In the main menu toolbar, navigate to **`Help`** .
8. In the drop-down menu, select **`Settings and Performance`** and in the additional drop-down menu, select **`Stop Performance Recording`** :



9. Once you've stopped the performance recording, you may need to wait a couple of seconds for Tableau to generate the performance recording report. Tableau will automatically open a new read-only workbook, called `PerformanceRecording`. In the following screenshot, you can examine the visualizations to see how various actions you've taken impact your workbook's performance:





Query

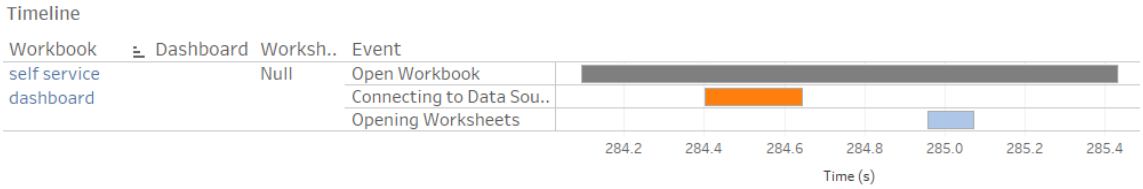
How it works...

The Tableau performance recording functionality is a built-in tool that can help you assess the performance of your workbook and identify processes that might be slowing it down. In this recipe, we started the performance recording, and performed some actions, such as filtering. After we finished recording, Tableau automatically generated a read-only workbook that contains a dashboard with the processes that have taken place since we started recording, and the respective times they took to run.

The dashboard contains the `Timeline`, `Events`, and `Query` views, which are described here:

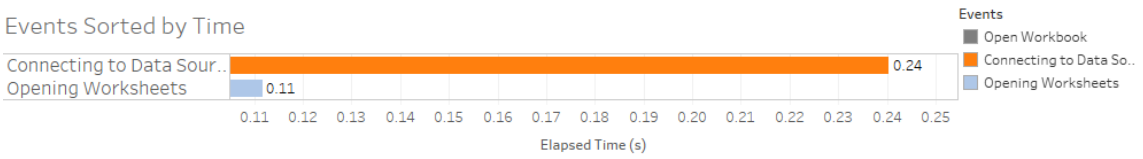
- `Timeline`: This view shows the processes that have taken place during the recording in chronological order. The processes are ordered from left to right along the

  > *x*
  >
  > ```
  > -axis, which represents the time since Tableau started. The view
  > ```
  >
  > *also provides information on the context of the events (`Workbook`, `Dashboard`, and `Worksheet`) and the nature of the event itself (`Event`). If you noticed performance issues with your workbook, this view is useful in exploring where the bottleneck occurs. Let's look at the processes that took place in the `Timeline` view, in the following screenshot:*



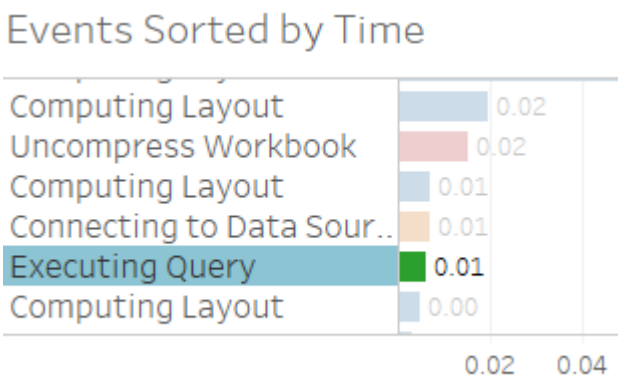
- `Events`: This view also shows the processes (events), but sorted by their duration, which is shown on the

  > *x*
  >
  > ```
  > axis. This view is very helpful because it highlights the processes
  > ```
  >
  > *that are taking the longest to run, making it easier to identify where the performance issue might be. Inthe following screenshot, we can see the processes of the `Events` view:*



Both of the preceding views can be adjusted by using the `Show Events taking at least (in seconds):` filter, located in the top of the dashboard. By default, events taking less than `0.10` second are filtered out of both views, but shorter events can also be displayed by moving the filter slider to the left. In the following screenshot, we can see how the views can be adjusted:

- `Query` : The view is, by default, empty. However, it will be populated when you select an `Executing Query` event in either the `Timeline` or `Events` view. Let's look at the `Query` view in the following screenshot:



When you select a query event, the SQL or XML text (depending on whether you're connected to a data source directly or to a published data source, respectively) of that query will be displayed in the `Query` view. You can then use the text of the query to optimize it. We can view the result in the following screenshot:

Query

```
SELECT ”Usage”.”Area” AS ”Area”,
 ”Usage”.”Settlement type” AS ”Calculation_858217245440413696”,
 AVG(CAST(”Usage”.”Internet penetration” AS DOUBLE PRECISION OR NULL)) AS ”avg:Internet penetration:ok”
FROM ”TableauTemp”.”Usage$” ”Usage”
GROUP BY 1,
 2
```

There's more...

Apart from Tableau's built-in functionality, some third-party tools geared toward the same purpose also exist. One of the more well-known ones is [*Power Tools for Tableau -- Performance Analyzer*]. Feel free to check it out: http://powertoolsfortableau.com/performance-analyzer-comes-to-workbook-tools-tableau.

Performance troubleshooting and best practices

After you've run the performance recording, you might have identified the processes that are slowing down your workbook's performance, or maybe you're just creating a workbook and would like to anticipate any potential issues. Either way, you'll want to implement some best practices in order to prevent or resolve any potential performance bottlenecks.

How to do it..

In the following sections, we'll look at the following topics:

- Limiting your data source
- Being cautious with filters
- Keeping an eye on the calculations
- Optimizing your visualizations

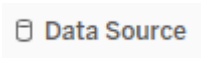
Limiting your data source

When we develop a workbook, we usually don't use all the data that we have in our database. In order to improve the performance of our workbook, it's always recommended to limit your data source to only the information that's needed.

Filtering out cases in the database

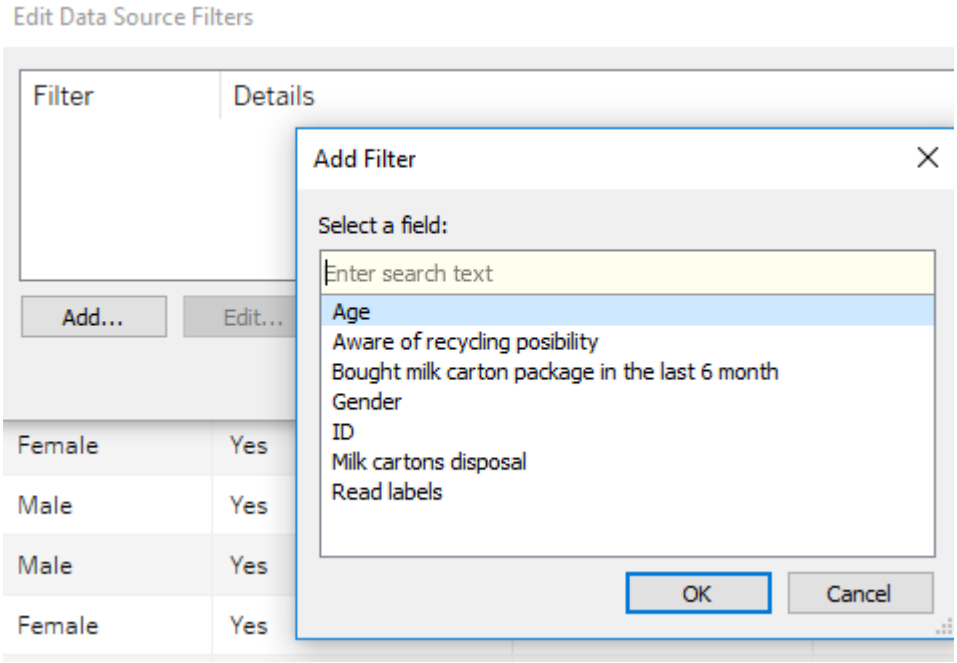
If you aren't going to use all the cases from your database, the best thing you can do is apply a filter to it:

1. Open any workbook that you want to improve, and go to the `Data Source` tab in the bottom-left corner:

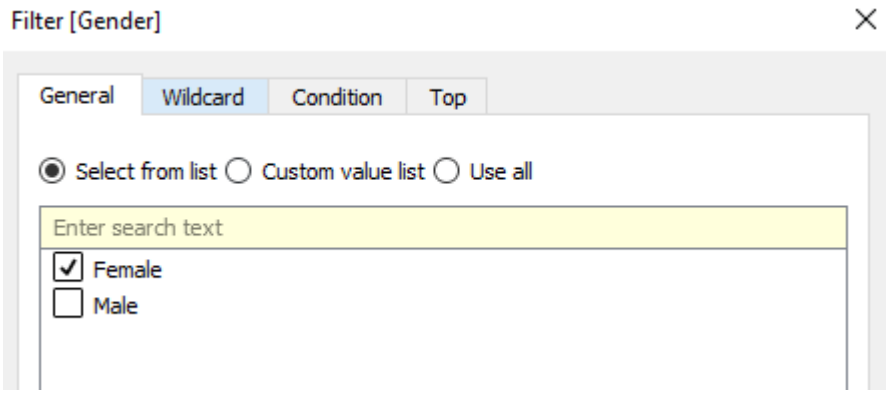


2. In the top-right corner, you can see the `Filters` section. Select `Add`:

3. The list of all of the variables in the database will be presented.
4. Chose the variable that contains the values you want to filter out:



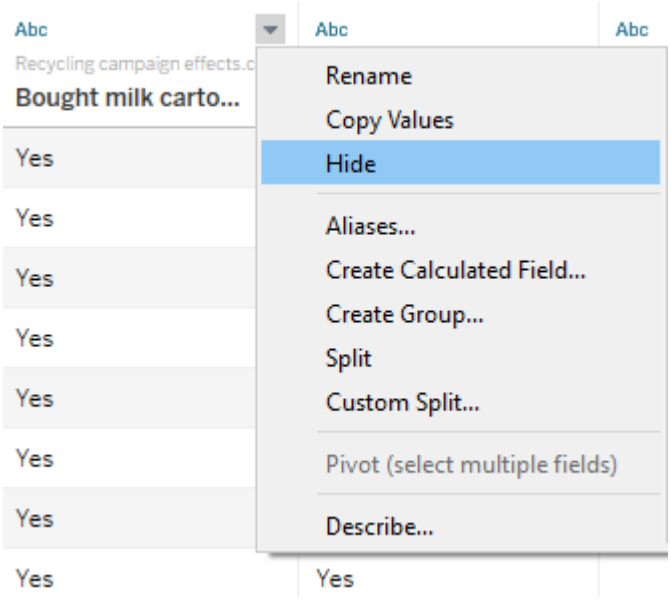
5. Select only those that you want to preserve (in our example, we want to keep only answers from female respondents), as shown in the following screenshot:



Filtering out variables

In most cases, we don't need all of the variables that we have in our database. In order to make our workbook run smoothly, we can hide variables that we don't need:

1. Open any workbook that you want to improve, and go to the `Data Source` tab in the bottom-left corner. Hover over the variable that you want to exclude until a black downward arrow appears.
2. Click on it and select `Hide`:



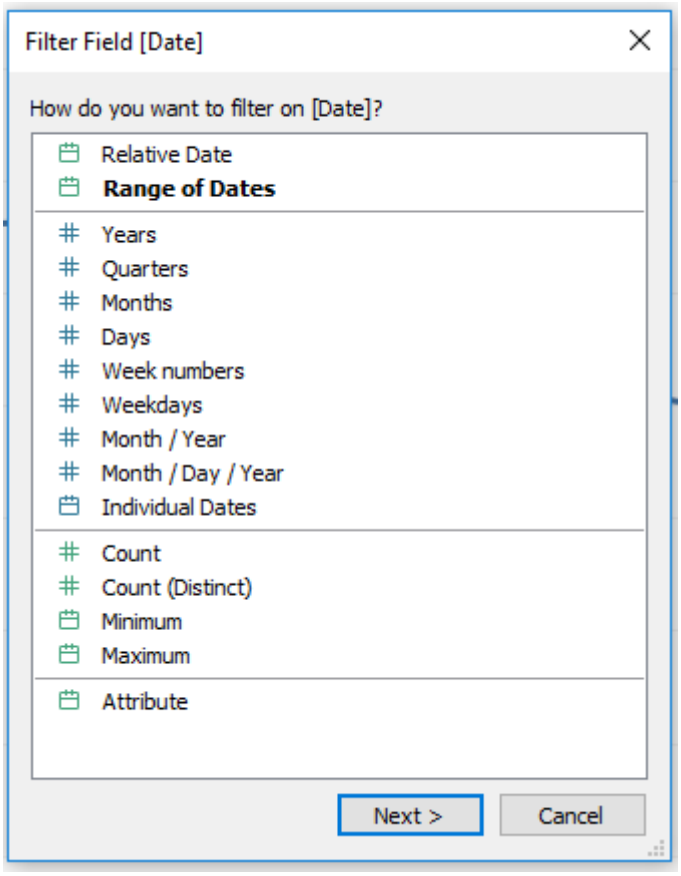
Making extracts

When we make a workbook that should communicate with a server, it's essential to decide how frequently our data should be refreshed. The default option is live. However, a live connection can be demanding and slow down your workbook. Luckily, in many situations, it isn't needed. In these situations, we can make extracts.

Being cautious with filters

Filtering is, without a doubt, a fundamental feature of Tableau. However, in some cases, filters can have a negative effect on a workbook's performance. In order to keep your workbook running optimally, try to be mindful of the following things:

- Avoid applying filters to long lists. Querying for each option in your list can be a demanding task for Tableau. In cases where this might be an issue, consider creating a calculated field to shorten the list or use parameters.
- Filtering date fields can also be a demanding task for Tableau because individual dates can amount to a lot of data points for Tableau to process. However, we often don't need to filter on the lowest level of date granularity. So, when filtering dates, always try to think about the actual level of granularity you need to filter on and go with the highest level that you can. If you can - filter out years; if not, ask yourself if it is sufficient to filter out quarters, and so on. If possible, avoid filtering on the lower levels, such as days or hours.
- When adding a date field to the `Filters` shelf, you can choose the level of granularity you would like to use, as shown in the following screenshot:



- In case you have smaller number of values in a dimension, then adding it to context can significantly improve the dashboard performance

Keeping an eye on the calculations

Calculated fields, especially when complex, can also give Tableau a hard time. However, you can counter this by trying to implement the following:

1. If there's a calculation metric that's frequently used in your workbook, consider creating it directly in the database, and not as a table calculation. This can save computational resources Tableau would otherwise have to invest in creating the views.

2. Disaggregated (row-level) calculations can be very demanding. Try to aggregate your measures when creating calculated fields.

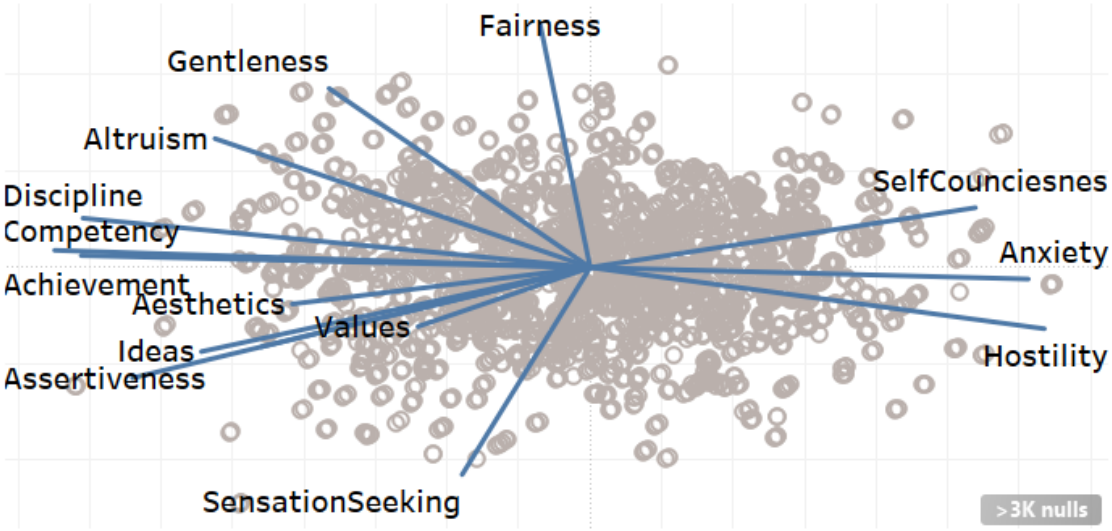
3. When alternatives are available, consider using functions that are faster to execute in your calculated fields. For example:

- Try replacing IF-ELSE statements with CASE-WHEN statements
- Try to avoid nested IF statements
- Try to skip using LOD expressions if not necessary, since they can be computationally demanding

Optimizing your visualizations

Demanding visualizations can also create some trouble and slow down your workbook. If you encounter performance issues, or simply want to make sure your workbook is going to run smoothly, pay attention to the following:

- A visualization that shows many data points, such as the one shown in the following screenshot, can slow down performance of the workbook:



The number of marks in the view is visible in the very bottom-left corner of the workbook, as shown in the following screenshot:

Note

A large number of marks can make rendering the view a very demanding task for Tableau to process. So, make sure the number of marks in your view isn't excessive.

- Advanced visualizations aren't the only ones that give Tableau a hard time. Simple cross-tabs, especially with a higher number of rows and columns can also be very computationally demanding. Try to limit the number of rows and columns used in a single visualization---a table with too many rows and columns isn't easy to read anyway.

How it works...

All of these tips are about saving computational resources and minimizing the amount of interaction your workbook has to do with the database. For example, creating extracts minimizes the interaction of the workbook with the data

base, which can be time consuming. On the other hand, creating frequently used calculations in the database minimizes the usage of computational resources.

There's more...

A range of third-party tools have been developed to help users assess and troubleshoot workbook performance. Some of them are geared specifically toward troubleshooting Tableau Server performance. Feel free to explore the options until you find the tool that addresses your needs the best.

Troubleshooting through log files

While running, Tableau records its activities in logs. If you encounter issues when working with Tableau, these logs can be extremely useful for troubleshooting.

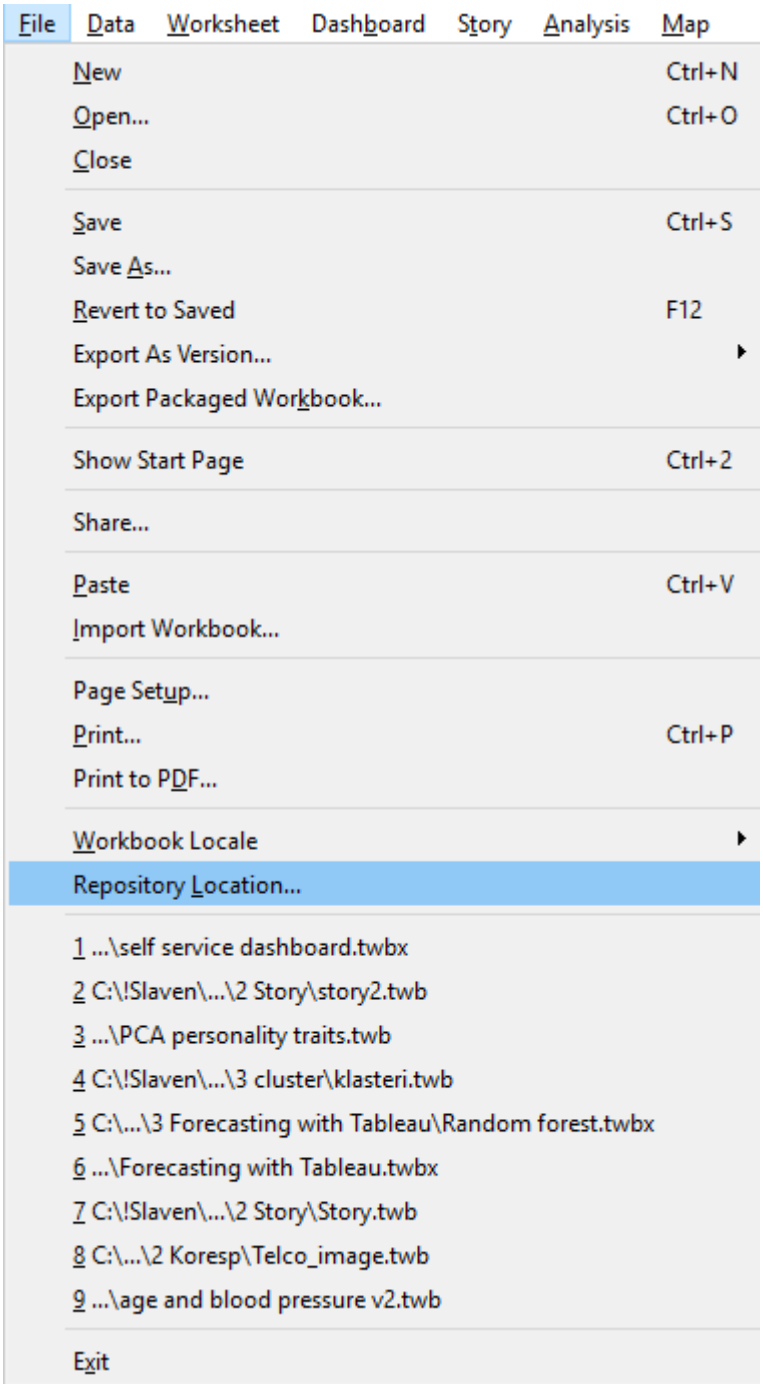
How to do it..

In this section, we'll look at the following topics:

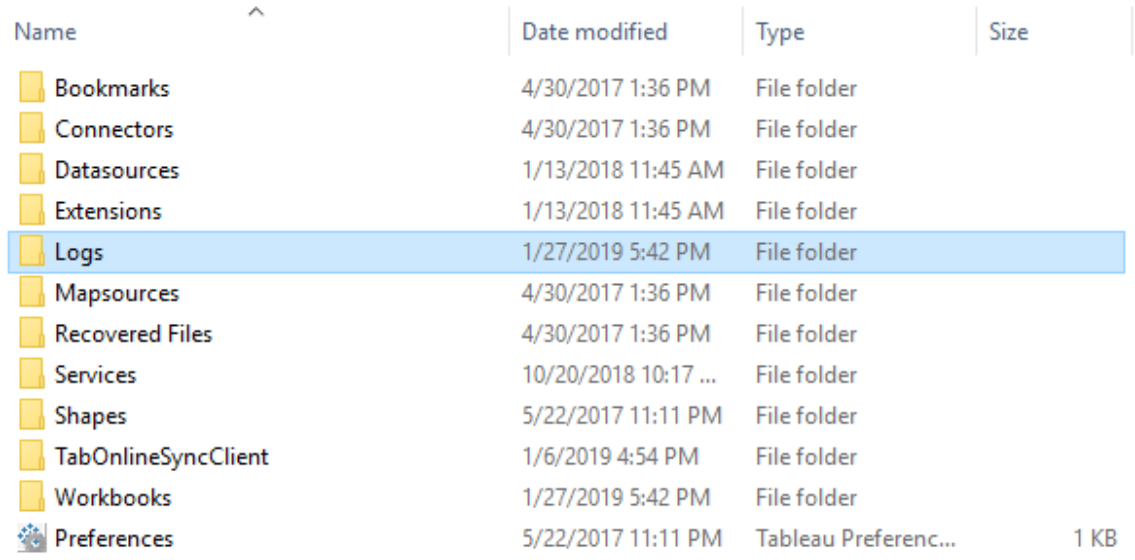
- Accessing logs
- Submitting logs to the support team

Accessing logs

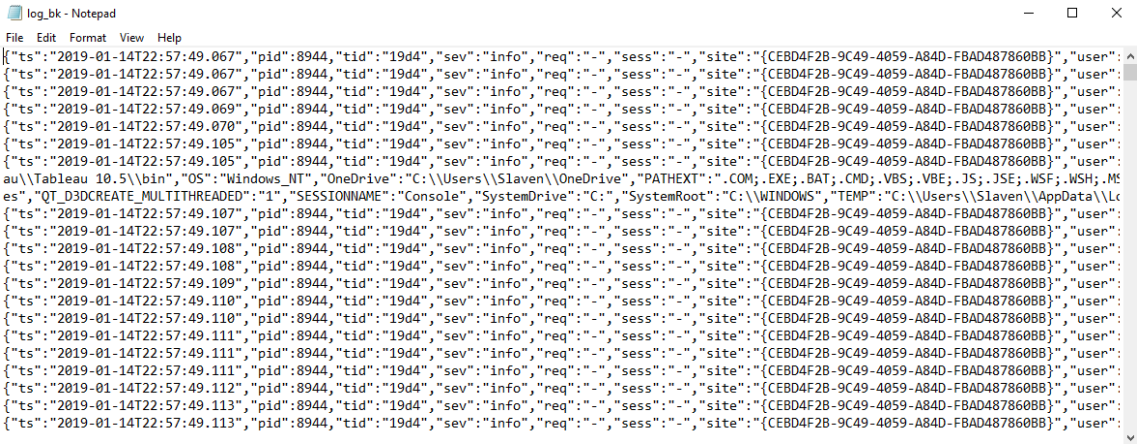
1. To find your logs, in the main menu toolbar, navigate to `File` | `Repository Location...`:



2. Alternatively, access the logs directly on your device. The default folder where the log files are saved is `/Users/<username>/Documents/My Tableau Repository`:



3. There's a wide range of issues that can occur, and it would be impossible to provide a comprehensive overview of them. However, once you've found the logs, you can use them to understand the nature of the issue you've experienced:

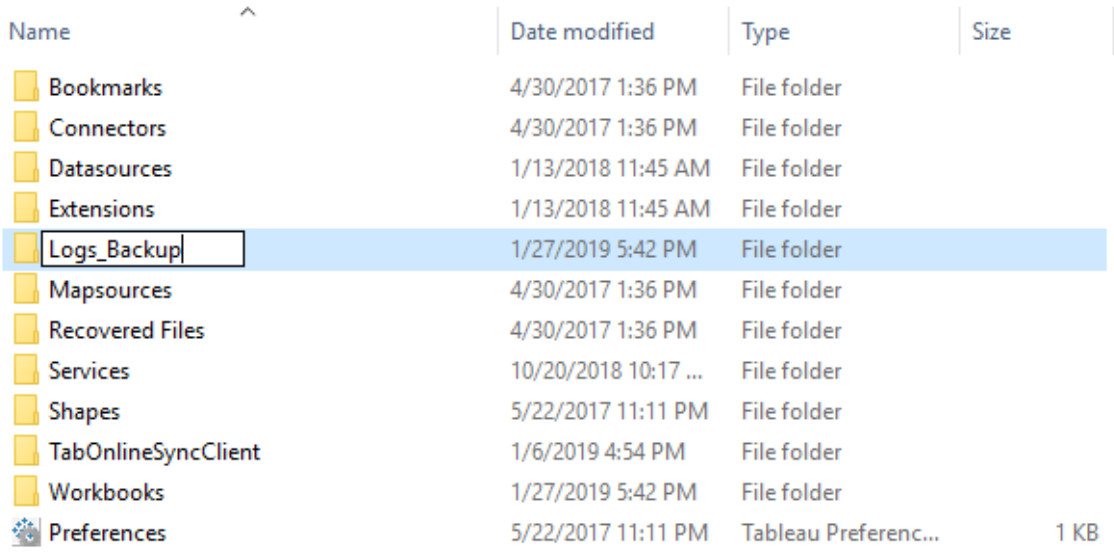


Submitting logs to the support team

If you have trouble diagnosing the problem, you can always contact Tableau support. In order to help them understand what has happened, you might be asked to provide log files where the issue has been recorded. Naturally, you'll want to provide log files in the manner that's most useful and informative for the Tableau support team.

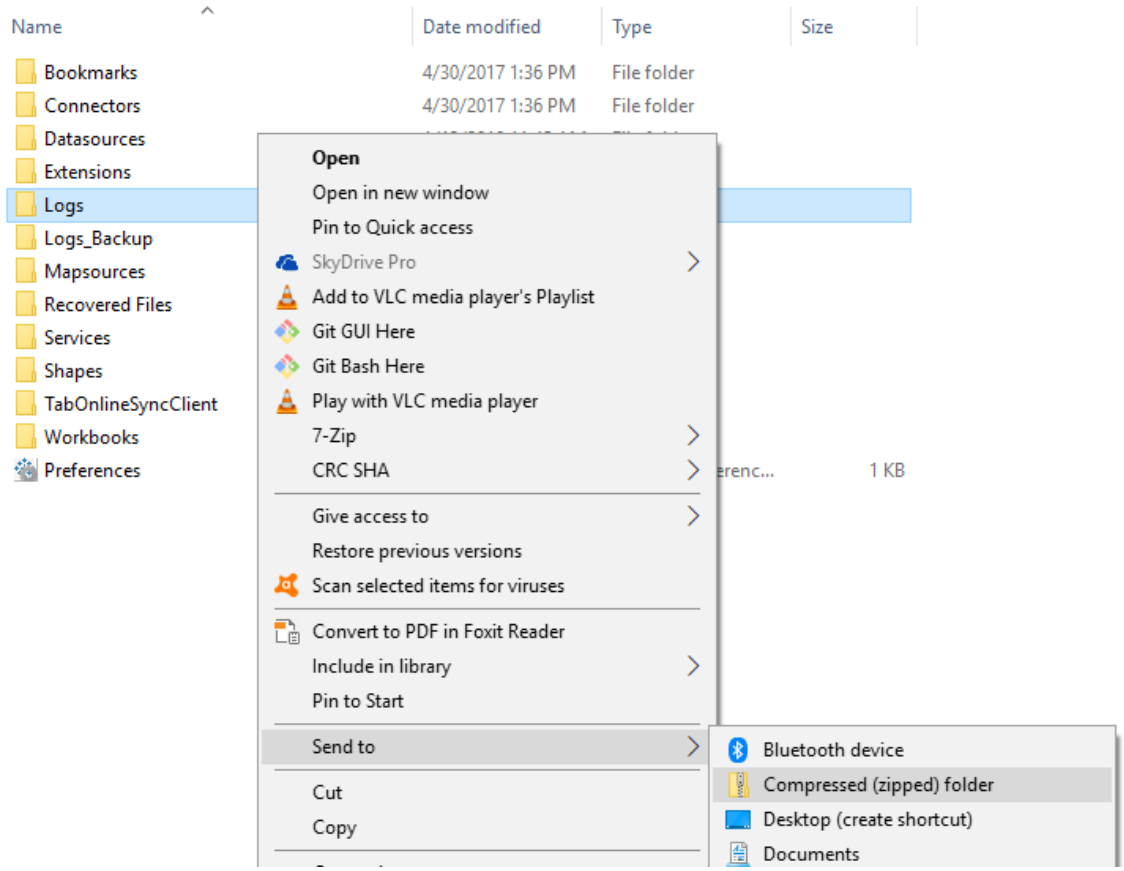
If possible, you should create clean logs. But first, create a backup log. You can do this by following these steps:

1. Navigate to the folder where your logs are located.
2. Rename the folder from `Logs` to `Logs_Backup`:



3. Create a new folder named `Logs`. `Logs` the new, clean folder where new logs will be stored, while the old folder, `Logs_Backup`, won't be used anymore. Through the following steps, you will reproduce the issue that has occurred and send the logs to Tableau support:

1. Start Tableau and take the steps required to get to the point where the problem has occurred.
2. Close all Tableau Desktop sessions, so that the errors are recorded to the log file.
3. Compress the `Logs` folder by right-clicking on it, selecting **`Send to`**, and choosing **`Compressed (zipped) folder`**, as shown in the following screenshot:



4. Send the compressed file to Tableau support.

There's more...

There are several open source applications that can make reading Tableau logs easier:

- https://github.com/tableau/tableau-log-viewer
- https://github.com/tableau/Logshark
- https://github.com/tableau/TabMon Each of the mentioned third-party software provides clear guidance for installation and usage, so feel free to explore.