

Quick Start Guide for Using R with Tableau

If this is your first time using R with Tableau, start at 1. If you have R, but don't have RStudio, start from 2. If you have both, move on to 3 to begin connecting R with Tableau.

This tutorial assumes a general command of Tableau and R. For more resources on R, check the appendix.

1. Downloading R

See [this LINK](#) for the file and instructions on downloading R.

2. Downloading RStudio

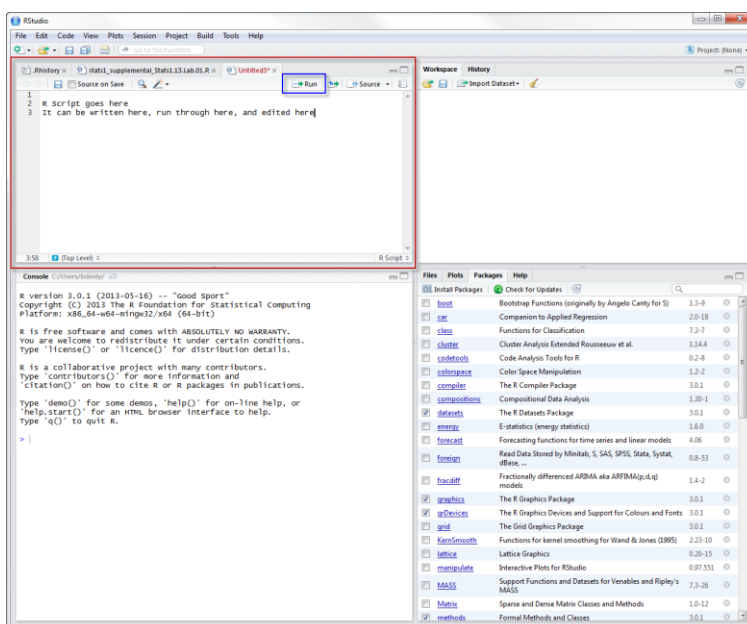
R's basic interface does not allow for editing script on the fly. Using R alone, one must write code in a text editor, save the script, then call that script in R. And when your script is wrong, R will give you a message and you'll have to go back to your text editor, correct the error, save the script again, then call it all over again.

RStudio is an integrated development environment (IDE) that solves the repetitive processes that come when using R on its own. It allows for writing, running, and editing of script to all happen within one application. It also has other nice features like file management, but that's beyond the scope of using R with Tableau. It will, though, help a lot in correcting errors that occur between Tableau and R. It will also help in testing R script quickly before you decide to implement it in Tableau.

[HERE](#) is the link to download RStudio.

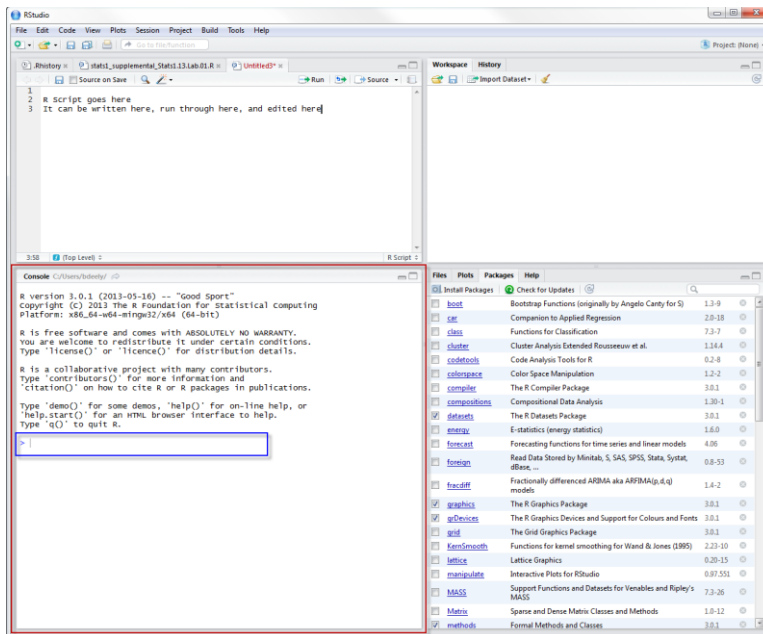
Point 1 – Top-Right Pane

In the top-right pane is the editing window. Here you can write, run, and edit R scripts. A very nice feature is that you can highlight and specify lines of code to run. This is useful when checking to make sure each section of script works properly separate from other sections. The tabs at the top can be managed just the same as Tableau's tabs for compartmentalizing work.



Point 2 – Bottom-Right Pane

In the bottom-right pane is the console. This is essentially the same as the console you see when you open R itself. Here code can be types in and run, but no editing is possible here.



3. Downloading Rserve

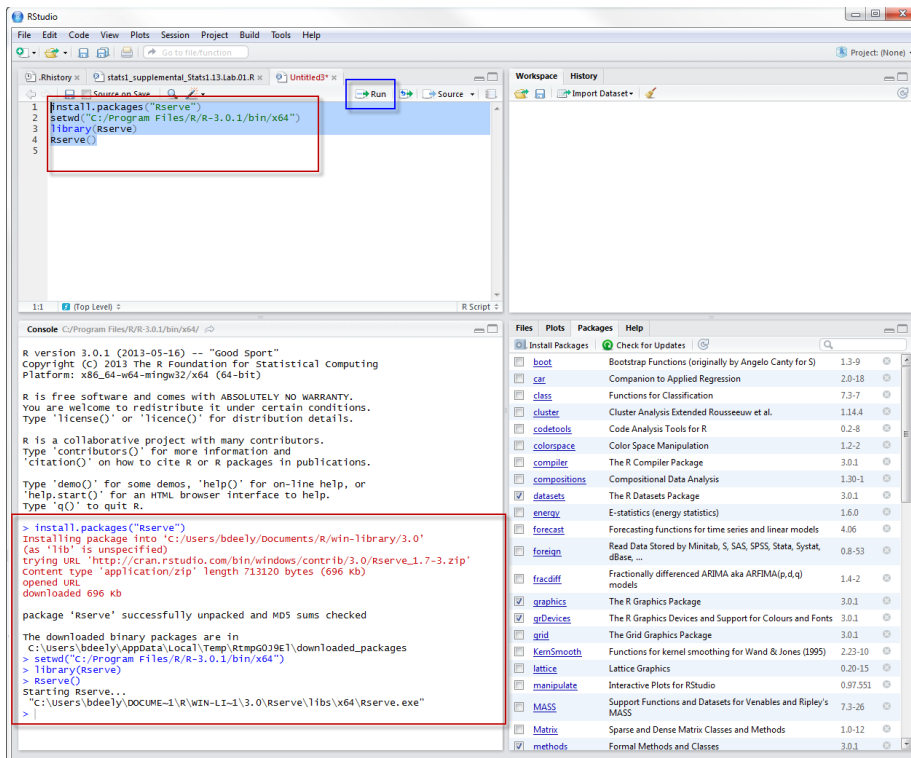
Rserve is a package for R. We will download it by writing script in R telling R to go find and download Rserve. Written here will be two examples, one using the console and the other using the editing window, on how to install Rserve. The two examples are used for illustration of why getting to know the editing pane is a good idea!

Copy and paste the following script into the editing pane as such:

```
install.packages("Rserve")
setwd("C:/Program Files/R/R-3.0.1/bin/x64")
library(Rserve)
Rserve()
```

Once you've pasted it, highlight the entire script as seen below and click on **Run** at the top-right of the edit pane. You specify certain lines of code to run separately by highlighting just the code you want to run and clicking **Run** as well.

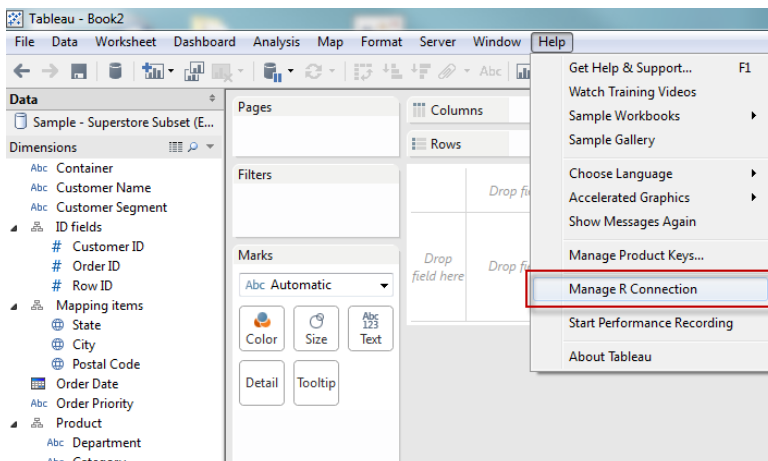
As you can see in the console, doing it this way runs the code in its entirety in one go.



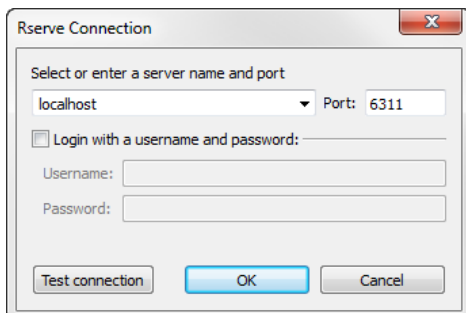
4. Connect R with Tableau

Once you have started running Rserve, connecting to Tableau is done within one easy menu.

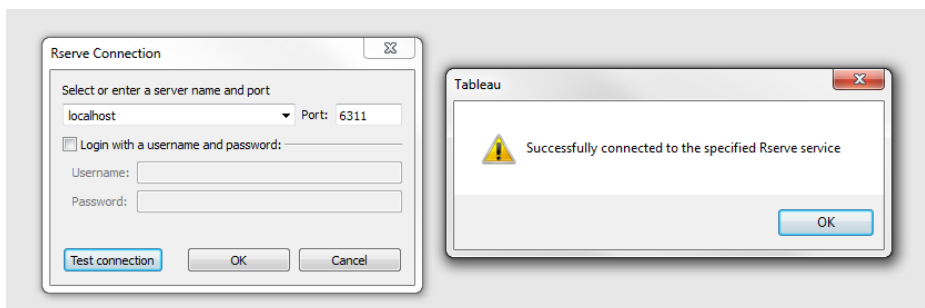
Go to **Help**, then **Manage R Connection**.



Set the server name to **Localhost** and the port to **6311**.



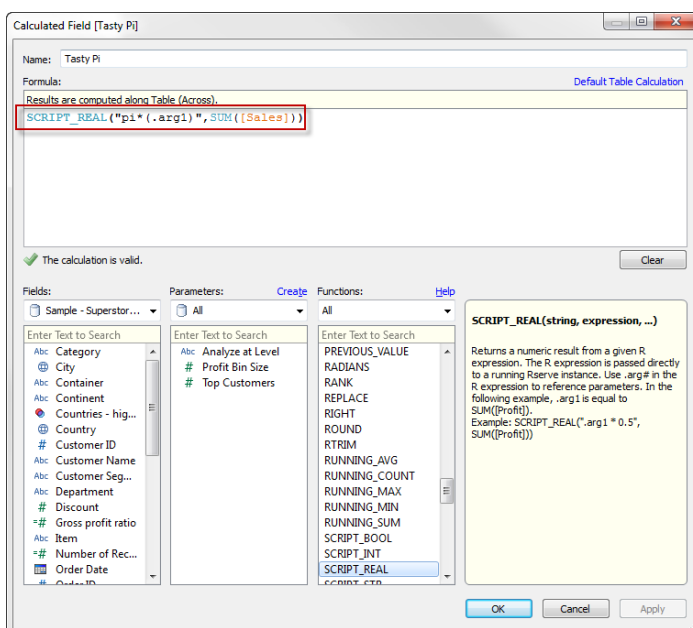
Click on **Test connection** to make sure everything runs smoothly and you should have a successful message. Click **OK** to close these menus.



5. Writing R Script in Tableau

5.1 The Basics

To start R script is written within Tableau's Calculated field menu.



Let's analyze the above sample script: `SCRIPT_REAL("pi*(.arg)",SUM([Sales]))`

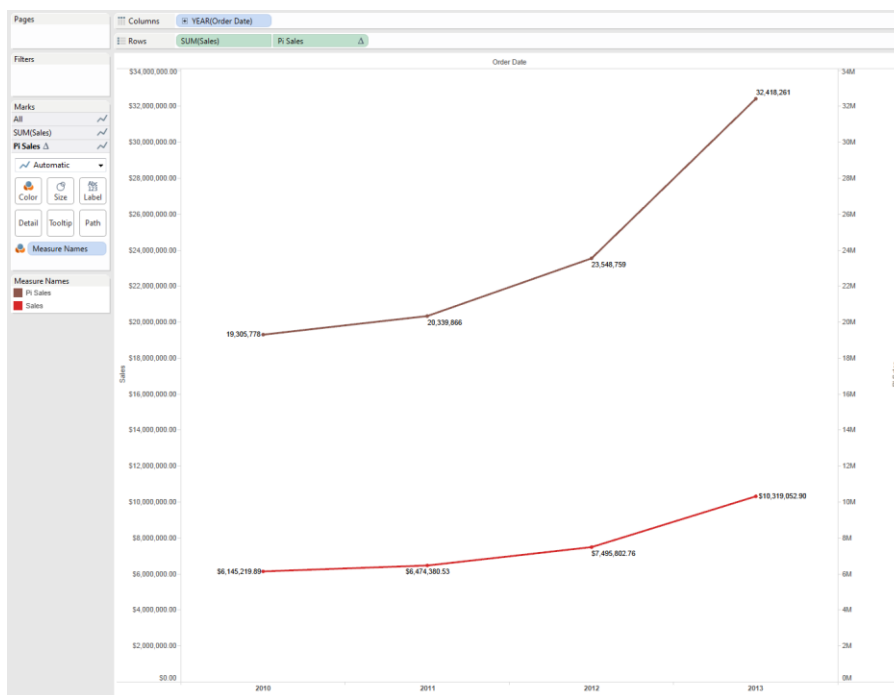
Once Rserve has been turned on and connected to Tableau, Tableau uses the `SCRIPT` function to run the R script back to R. R makes the calculations and sends them back to Tableau.

Tableau can run booleans, integers, real numbers, and strings through R, denoted as `SCRIPT_BOOL`, `SCRIPT_INT`, `SCRIPT_REAL`, and `SCRIPT_STR`, respectively.

Within the parentheses the first section is the actual R script. This must be placed in either single or double quotations. In this case we have `"pi*(.arg)"`, which is telling R to multiply the number Pi by the first argument. Within the R script, we refer to each measure we want to analyze as `.arg`.

Lastly, we place the measure(s) we want to analyze, each separated by commas.

Here's what our sales look like when multiplied by Pi.

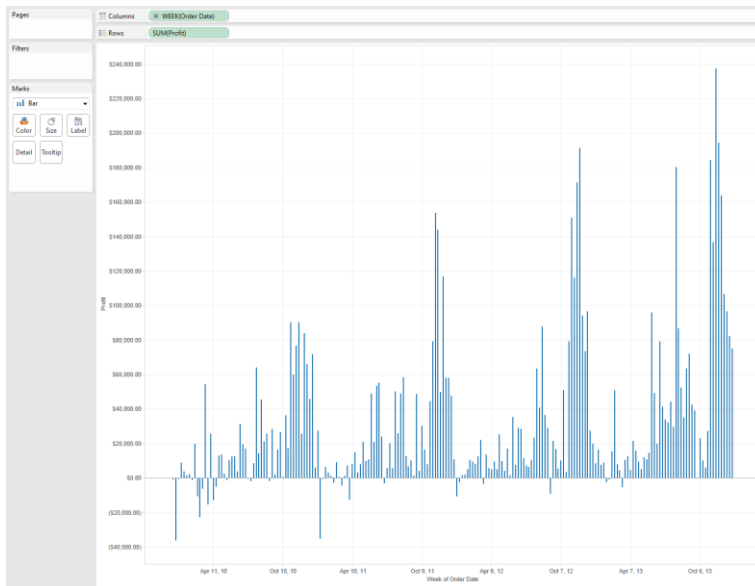


5.2 Outliers

Datasource: Sample - Superstore Sales English

Business question: what weeks had **profit** that was *significantly* different from our normal trends?

Currently, management sees their profit on a weekly basis as such:



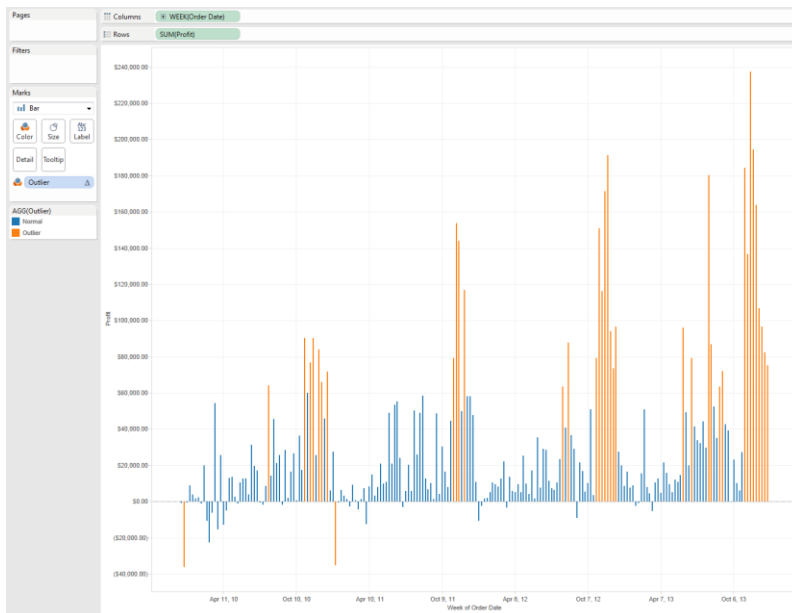
Putting profit on color would give some insight, but it still won't tell how which specific weeks were significantly different from normal trends. Let's fix that!

Create a new calculation called **Outlier** and copy and paste the following code, which will be explained below piece-by-piece. We are simply creating a KPI if/else statement.

```
IF SCRIPT_REAL("library(mvoutlier); sign2(cbind(.arg1))$wfinal01",
               SUM([Profit])) == 0
THEN "Outlier"
ELSE "Normal"
END
```

SCRIPT_REAL	Run R script using a real number
library(mvoutlier)	Call the mvoutlier package from R's library
;	Semi-colon used to separate steps or lines in R script
sign2	Calls quick algorithm to identifying multivariate outliers
cbind	
.arg1	Stand-in variable for the measure we will analyze
\$wfinal01	Sets those measures that are zero to outliers

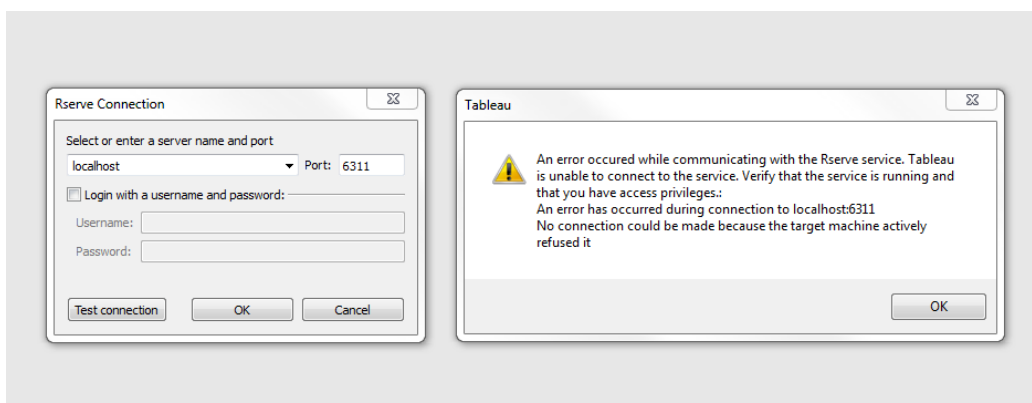
Now using the Outlier calculation on color, we have the following graph:



6. Common Error(s)

6.1 R Connection not working

Unless Rserve is up and running, you will get the following error when you try to test the connection with Tableau.



Each time you start your computer and open Tableau for the first time, you must make sure to go into R, either standalone or RStudio, and run the steps found in section 3 to turn Rserve on.

7. Additional Resources

7.1 R Resources



[Official R website](#)

[Stackoverflow](#): great place to ask questions on R functionality and scripting and for finding past solutions.

[Github](#): great place to look for R script people have written and shared

[R-bloggers](#): a great place to see what kind of things people are doing with R and visual analytics

[Coursera's data courses](#): many of the data and statistics courses use R, which makes it a great place to get your feet wet.