

R and TABLEAU

To use R with Tableau you will need to install the Rserve package: <https://cran.r-project.org/web/packages/Rserve/index.html>

You could try

```
install.packages('Rserve')
```

or try

```
install.packages('Rserve', "http://rforge.net/", type="source")
```

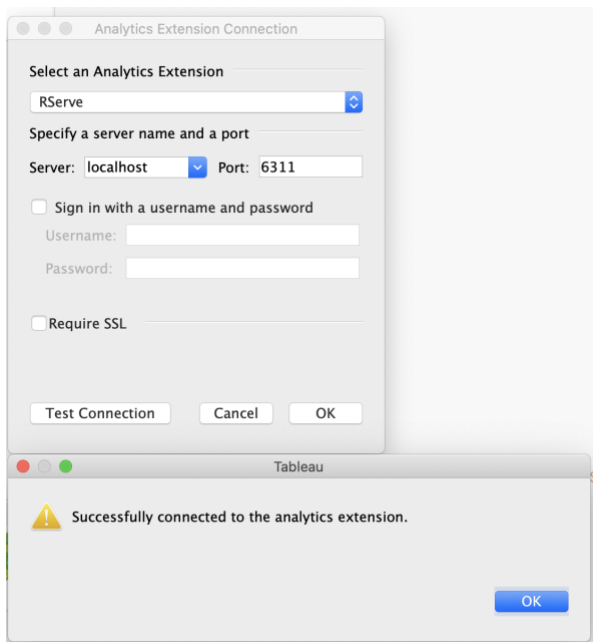
We will use it to “Connect to Tableau”. Generally, you should be able to load the package in R and start an instance of Rserve, like shown below:

```
library(Rserve)
Rserve(args = "--no-save")
```

However, a more stable approach (at least in some Mac OS systems) is to type the following command in your terminal:

```
R CMD Rserve from terminal
```

Now, make the connection and test the connection. Go to Tableau, click the **Help** menu, then **Settings and Performance**, then “**Manage Analytics Extension Connection**”



We will do this every time we want to use R-enabled workbooks (Start R, start Rserve, confirm connection)

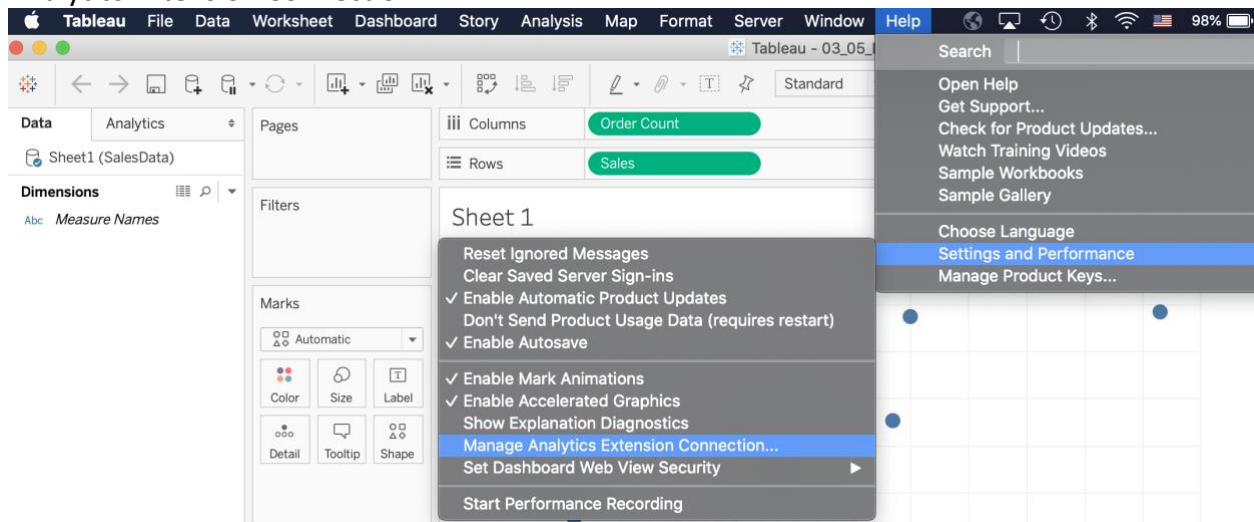
VISUALIZE LINEAR REGRESSION MODELS

Start an RServe instance:

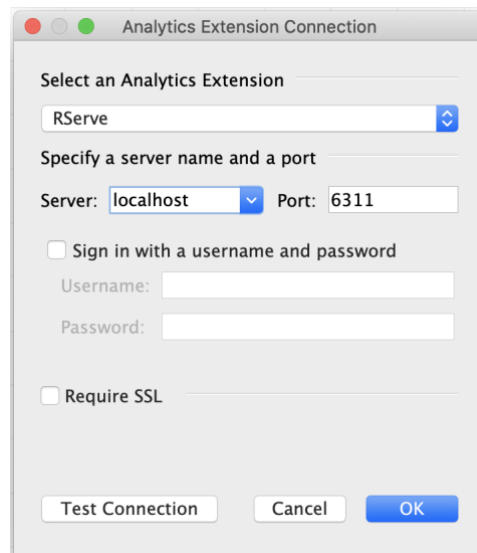
R CMD Rserve from terminal

Open the Tableau workbook 03_05_Regression.twbx

Go to Tableau's Help menu, select "Settings and Performance" and click on the "Manage Analytics Extension Connection ..."



Verify the Analytics Extension says Rserve, the Server is set to localhost, and the Port to 6311. Click on Test Connection to make sure the connection is successful.



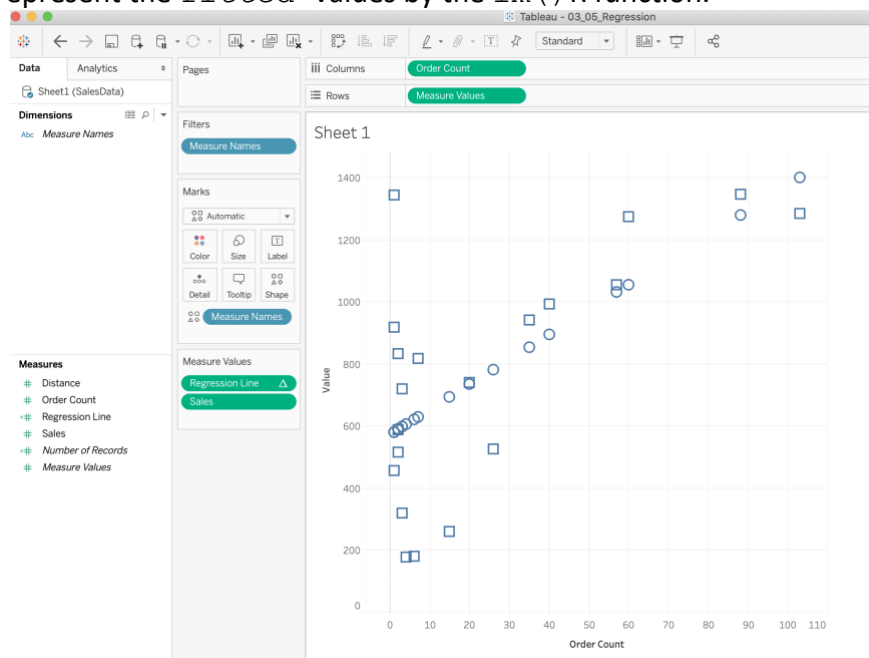
We use a **calculated field** that uses an R commands to compute the regression line.

Go to the **Analysis** menu, select **Create Calculated Field** and include:

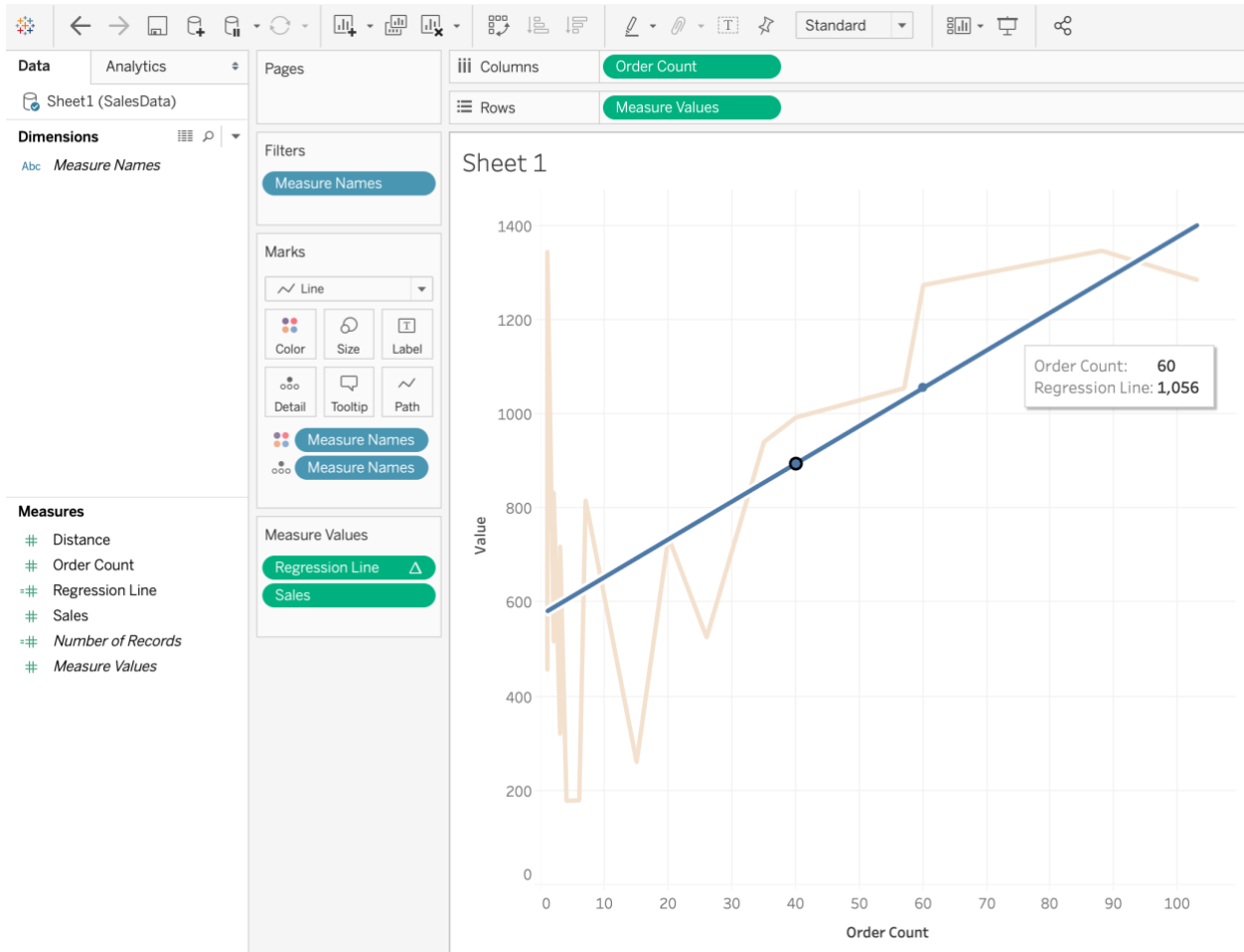


```
SCRIPT_REAL(  
"model <- lm(.arg1 ~ .arg2 )  
  
model$fitted'',  
SUM([Sales]), SUM([Order Count])  
)
```

After hitting **OK**, Regression Line has been added to the **Measures**, and it can be added to the visualization. Drag it onto the vertical axis. You will see new points (with a different marker) that represent the fitted values by the `lm()` R function.



To show the actual regression line, you can go to the **Marks** pane and select **Line** instead of **Automatic**. Additionally, drag *Measure Names* from the **Dimensions** area to the **Color** aesthetic in the **Marks** pane.



Find the documentation for the `lm()` R function at:

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/lm>

VISUALIZE CLUSTERING RESULTS

Assuming you have started an RServe instance already like described before.

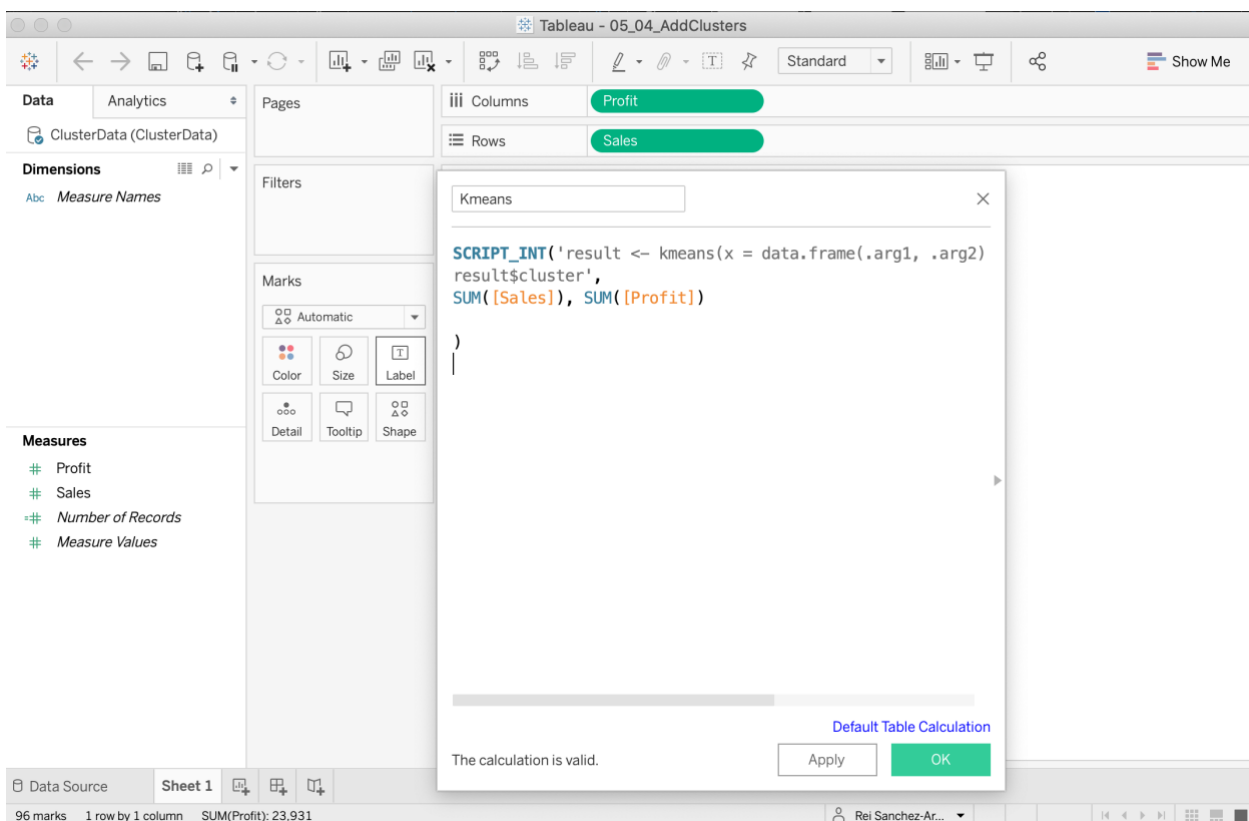
Open the Tableau workbook `05_04_AddClusters.twbx`

Go to Tableau's **Help** menu, select **"Settings and Performance"** and click on the **"Manage Analytics Extension Connection ..."**. Verify the connection is successful.

We use a **calculated field** that uses an R commands to implement the k-means clustering algorithm implemented with the `kmeans()` R function.

Go to the **Analysis** menu, select **Create Calculated Field** and include:

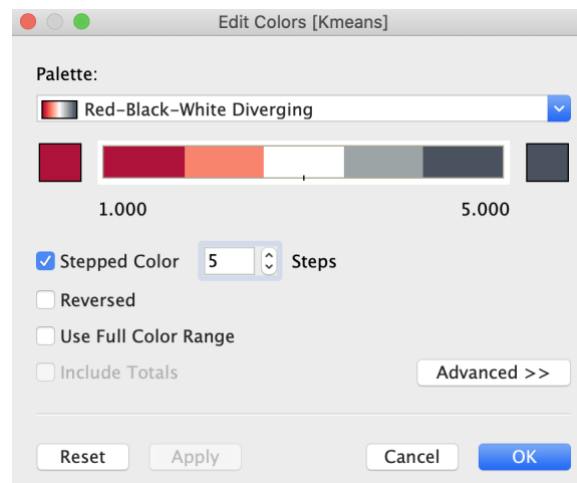
```
SCRIPT_INT('result <- kmeans(x = data.frame(.arg1, .arg2),
centers = 5, iter.max = 10)
result$cluster',
SUM([Sales]), SUM([Profit])
)
```



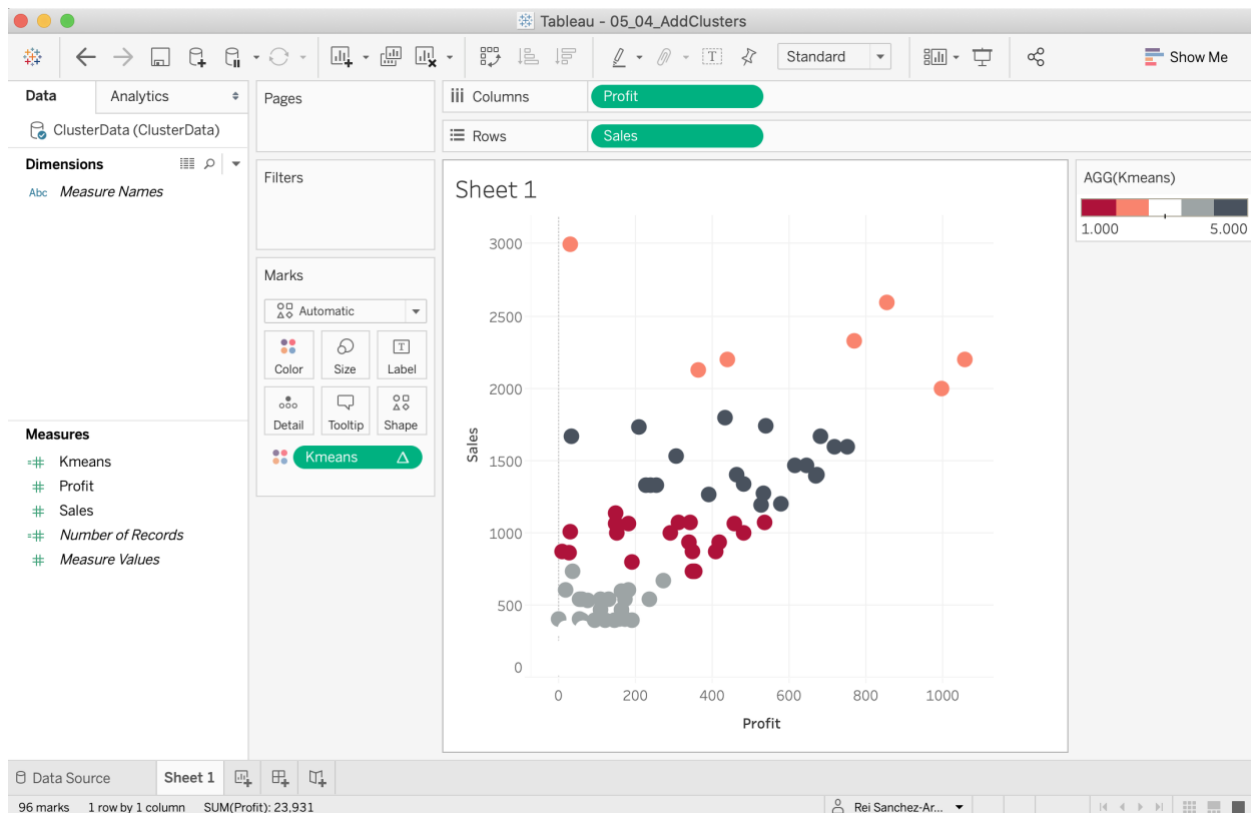
Find the documentation for this R function at:

<https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/kmeans>

Drag the new calculated field **Kmeans** onto the **Colors** mark. Go to the colors cart to modify the color palette used. For example, click **Edit colors...**, and select **Stepped color**. Pick a different one, for example by selecting for example **Red-Black-White Diverging**



The groups created by this clustering algorithms should now be more evident.



The previous two example serve as a demonstration on how to use the statistical modeling capabilities in R, with the data visualization opportunities in Tableau.