

R Meetup

3 Lightning Topics

23 Oct 2018

Michael Mooney

THE FOLLOWING PRESENTATION HAS BEEN RATED

	RESTRICTED
	95% Confidence Interval of less than 213 slides

®

BY CRAN The Comprehensive R Archive



CRAN The Comprehensive R Archive

cran.r-project.org

www.bioconductor.org

3 Lightning Topics

- What's new in R 3.5
- R and Bert
- R and PowerBI

About Me – Michael Mooney



Data Analyst

Using R for a long time

Not an expert in anything

Shouldn't be let outside alone

Michael.Meetup@eipsoftware.com

What's new in R 3.5

NOT MUCH!

- Implementation of ALTREP project
- Packages are byte-compiled on installation
- readLines and scan are quicker

What's new in R 3.5

- Implementation of ALTREP project

Initialization of vectors is faster. Doesn't allocate memory just start and end values

Keeps track if vector is already sorted, won't try and re-sort

Deferred coercion, until needed. Converting vector from numeric to character. i.e. `as.character(numeric_vector)`

What's new in R 3.5

- Packages are byte-compiled on installation

Base and others like dplyr were already

However now github packages will be too

What's new in R 3.5

- readLines and scan are quicker

Text buffering allows faster reading.

R and Bert – Sleeping with the Enemy



R and Bert



Bert: Basic Excel R Toolkit

Allowing almost seamless interoperability
between Excel and R

<https://bert-toolkit.com>

R and Bert

Open Source: <https://bert-toolkit.com/>

Developed by Structured Data LLC

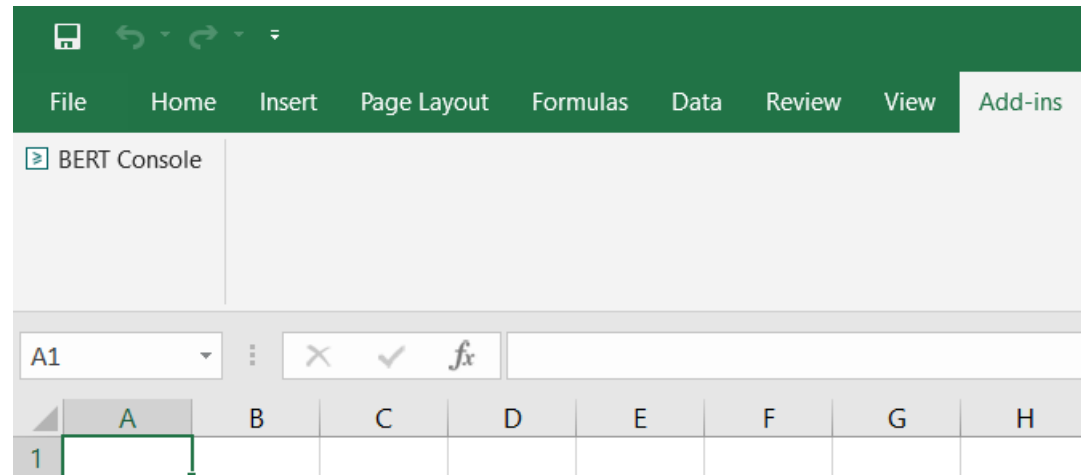
Github Repo: <https://github.com/sdllc/Basic-Excel-R-Toolkit>

R and Bert

Installing

Get file, Install, next, next, next....

After Install – Excel Add In

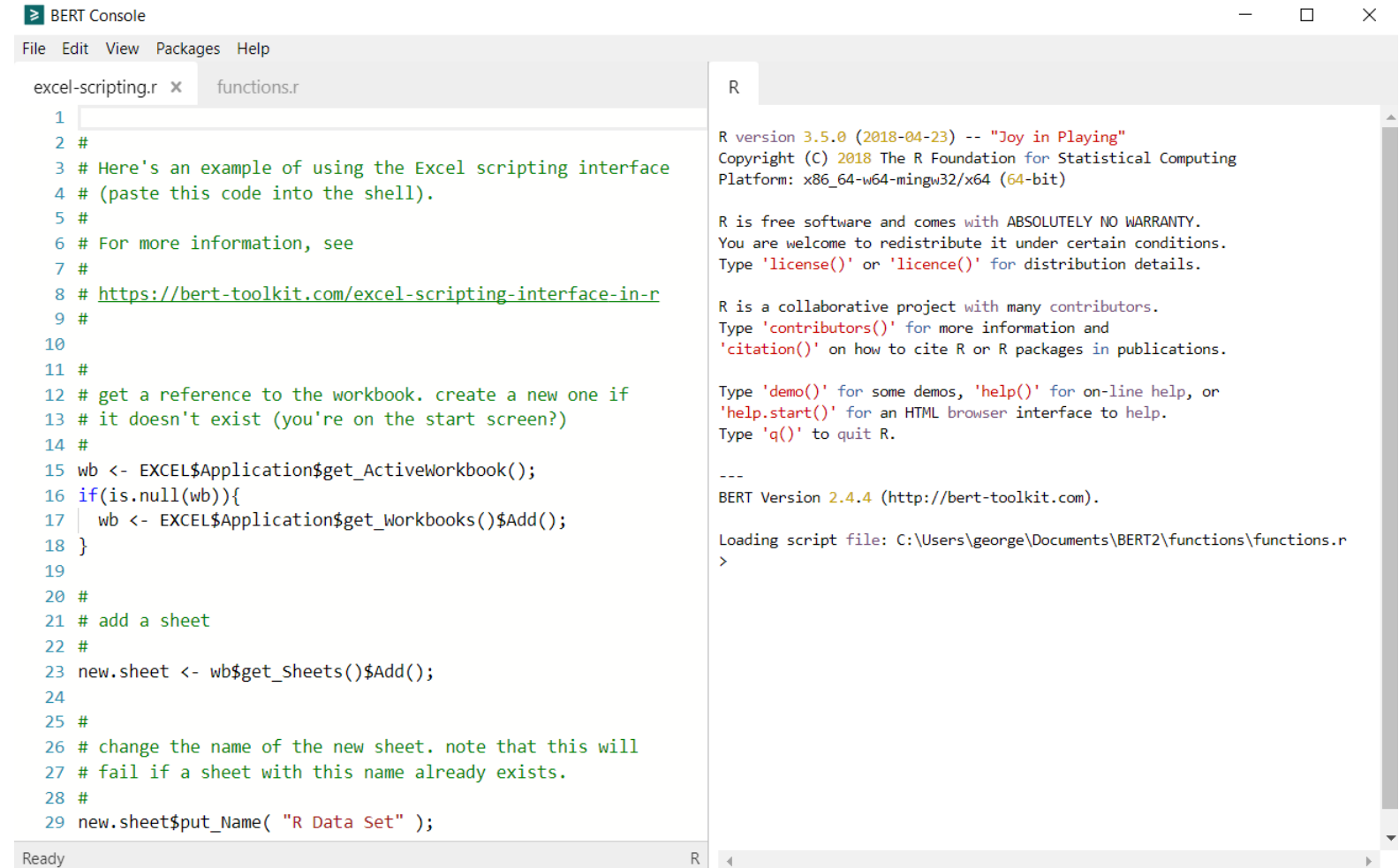


R and Bert

Bert's R
Console Window

Script file on Left

R Console on Right



The screenshot shows the BERT Console application window. The title bar reads "BERT Console". The menu bar includes "File", "Edit", "View", "Packages", and "Help". The left pane displays a script file named "functions.r" with the following content:

```
1  
2 #  
3 # Here's an example of using the Excel scripting interface  
4 # (paste this code into the shell).  
5 #  
6 # For more information, see  
7 #  
8 # https://bert-toolkit.com/excel-scripting-interface-in-r  
9 #  
10  
11 #  
12 # get a reference to the workbook. create a new one if  
13 # it doesn't exist (you're on the start screen?)  
14 #  
15 wb <- EXCEL$Application$get_ActiveWorkbook();  
16 if(is.null(wb)){  
17 | wb <- EXCEL$Application$get_Workbooks()$Add();  
18 }  
19  
20 #  
21 # add a sheet  
22 #  
23 new.sheet <- wb$get_Sheets()$Add();  
24  
25 #  
26 # change the name of the new sheet. note that this will  
27 # fail if a sheet with this name already exists.  
28 #  
29 new.sheet$put_Name( "R Data Set" );
```

The right pane shows the R console output:

```
R  
R version 3.5.0 (2018-04-23) -- "Joy in Playing"  
Copyright (C) 2018 The R Foundation for Statistical Computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
---  
BERT Version 2.4.4 (http://bert-toolkit.com).  
  
Loading script file: C:\Users\george\Documents\BERT2\functions\functions.r  
>
```

The status bar at the bottom of the window shows "Ready" on the left and "R" on the right.

R and Bert - Sending from R to Excel

```
wb <- EXCEL$Application$get_ActiveWorkbook();  
if(is.null(wb))  
{  
  wb <- EXCEL$Application$get_Workbooks()$Add();  
}  
new.sheet <- wb$get_Sheets()$Add();  
new.sheet$put_Name( "R Data Set" );  
range$put_Value( iris );
```

R and Bert – $f(x)$

R processing inside of Excel

Save R script file in ..\Documents\BERT2\functions

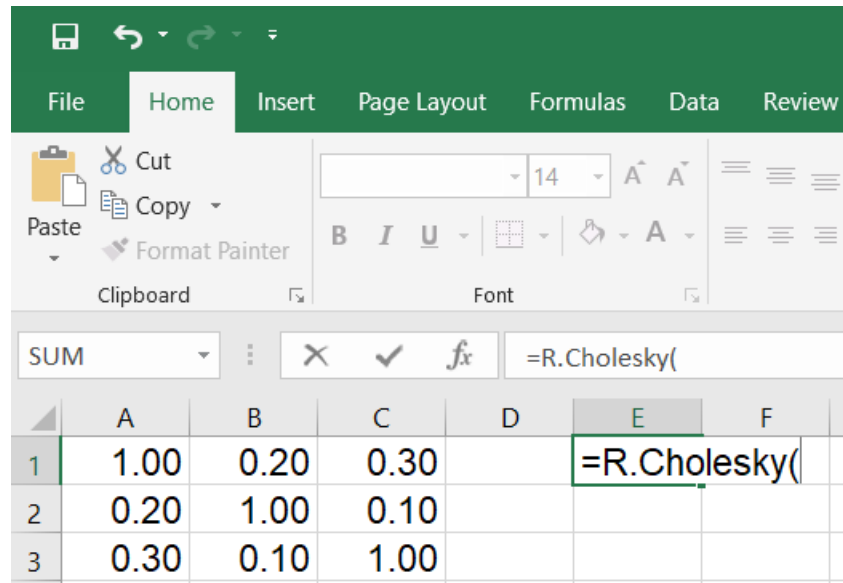
Encapsulate the Function: Cholesky decomposition of a matrix

```
#add the Cholesky decomposition of a matrix
Cholesky <- function(mat)
{
  chol(mat);
}
```

R and Bert – $f(x)$

Save the R script file.

Functions automatically available in Excel; with “R.” prefix



R and Bert – $f(x)$

R processes function

Results returned to Excel.

R returns a matrix; so formula needs to be entered as ctrl-shift-enter to put braces around formula {}

E1 ✕ ✓ <i>fx</i> {=R.Cholesky(A1:C3)}							
	A	B	C	D	E	F	G
1	1.00	0.20	0.30		1.00	0.20	0.30
2	0.20	1.00	0.10		0.00	0.98	0.04
3	0.30	0.10	1.00		0.00	0.00	0.95
4							

R and Bert – f(x)

Add Category for the R function

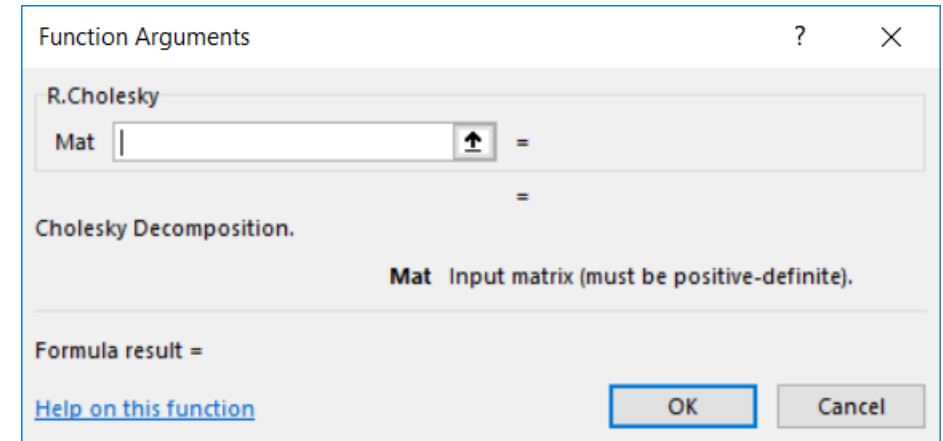
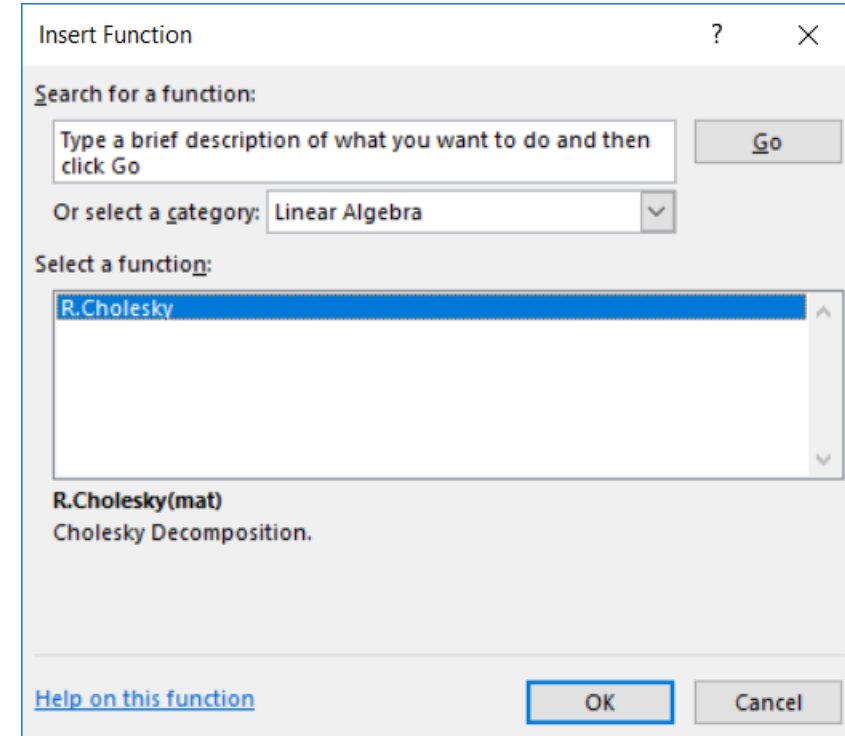
put the function into a category for Excel Functions

```
attr( "category" ) <- "Linear Algebra";
```

Add help description for R function

Add a description for the added function for helpness

```
attr( "description" ) <-  
list( "Cholesky Decomposition",  
mat="Input matrix (must be positive-definite)" );
```



R and Bert - R Plot in Excel – WHAT??!?!?

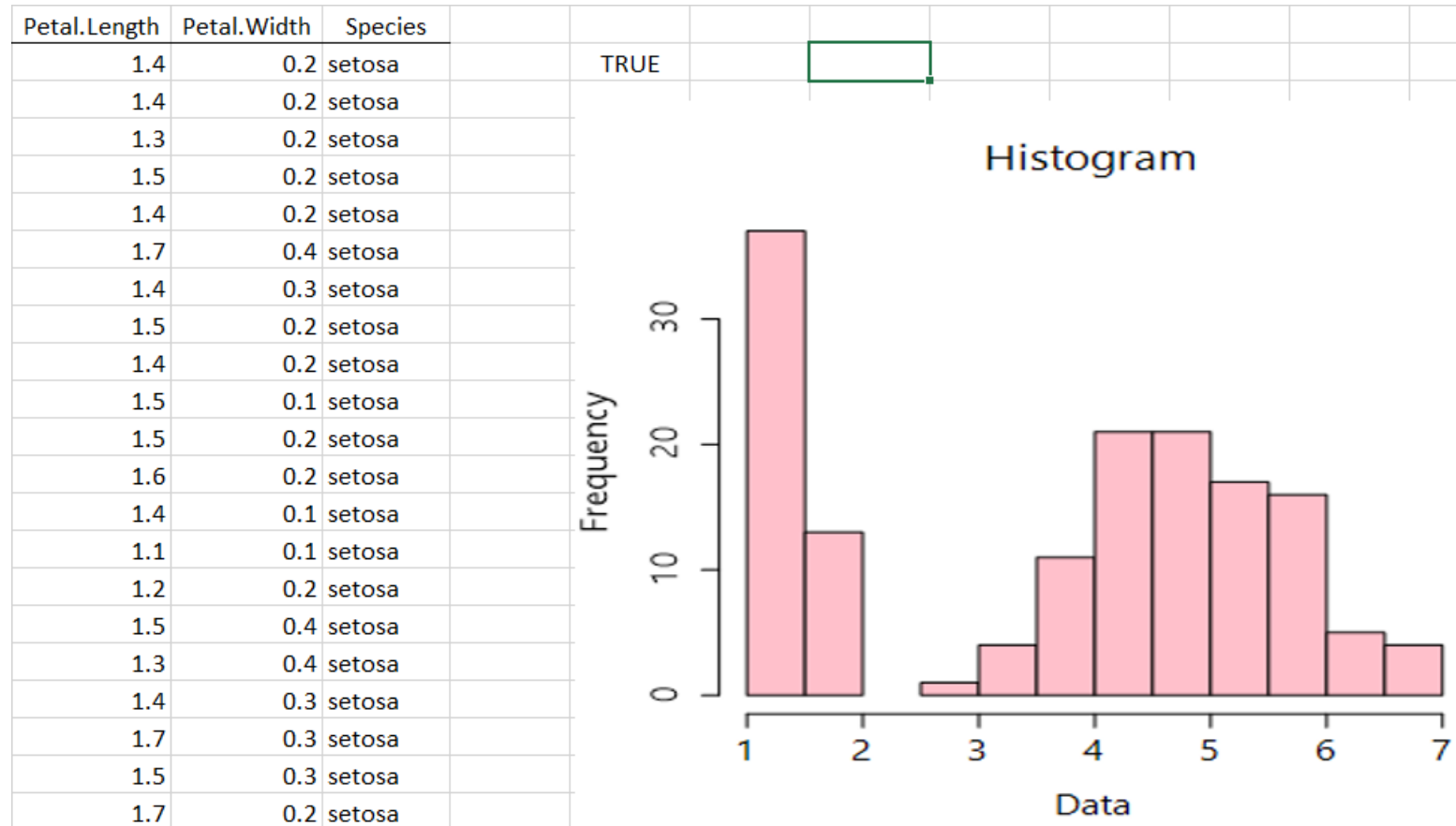
```
graph.histogram <- function(data, main="Histogram", xlabel="Data")
{
  # passing cell=T means "use the cell address as a unique identifier".
  # otherwise, use the name parameter to identify target shape.
  BERT.graphics.device(cell=T);

  # scrub the data (slightly) then generate a histogram
  x <- unlist( as.numeric( data ));
  hist( x, xlab=xlabel, main=main, col="pink", breaks=13,font.main=1);

  # we're done with the graphics device; we can shut it off, limit of 63
  # active devices so it's a good idea.
  dev.off();

  # returning TRUE indicates everything succeeded.
  TRUE
}
```

R and Bert - R Plot in Excel - WHAT??!?!?



R and PowerBI



Power BI

R and PowerBI

After starting PowerBI

File – Options

R Scripting

And the directories are set correctly

Options

GLOBAL

Data Load
Power Query Editor
DirectQuery
R scripting
Security
Privacy
Updates
Usage Data
Diagnostics
Preview features
Auto recovery

CURRENT FILE

Data Load
Regional Settings
Privacy
Auto recovery
Query reduction
Report settings

R script options

To choose a home directory for R, select a detected R installation from the drop-down list, or select Other and browse to the location you want.

Detected R home directories:

C:\Program Files\R\R-3.5.1

[How to install R](#)

To choose which R integrated development environment (IDE) you want Power BI Desktop to launch, select a detected IDE from the drop-down list, or select Other to browse to another IDE on your machine.

Detected R IDEs:

R Studio

[Learn more about R IDEs](#)

[Change temporary storage location](#)

Note: Sometimes, R custom visuals automatically install additional packages. For those to work, the temporary storage folder name must be written in Latin characters (letters in the English alphabet).

OK

Cancel

R and PowerBI – Pull Data From R

Get Data via R Script

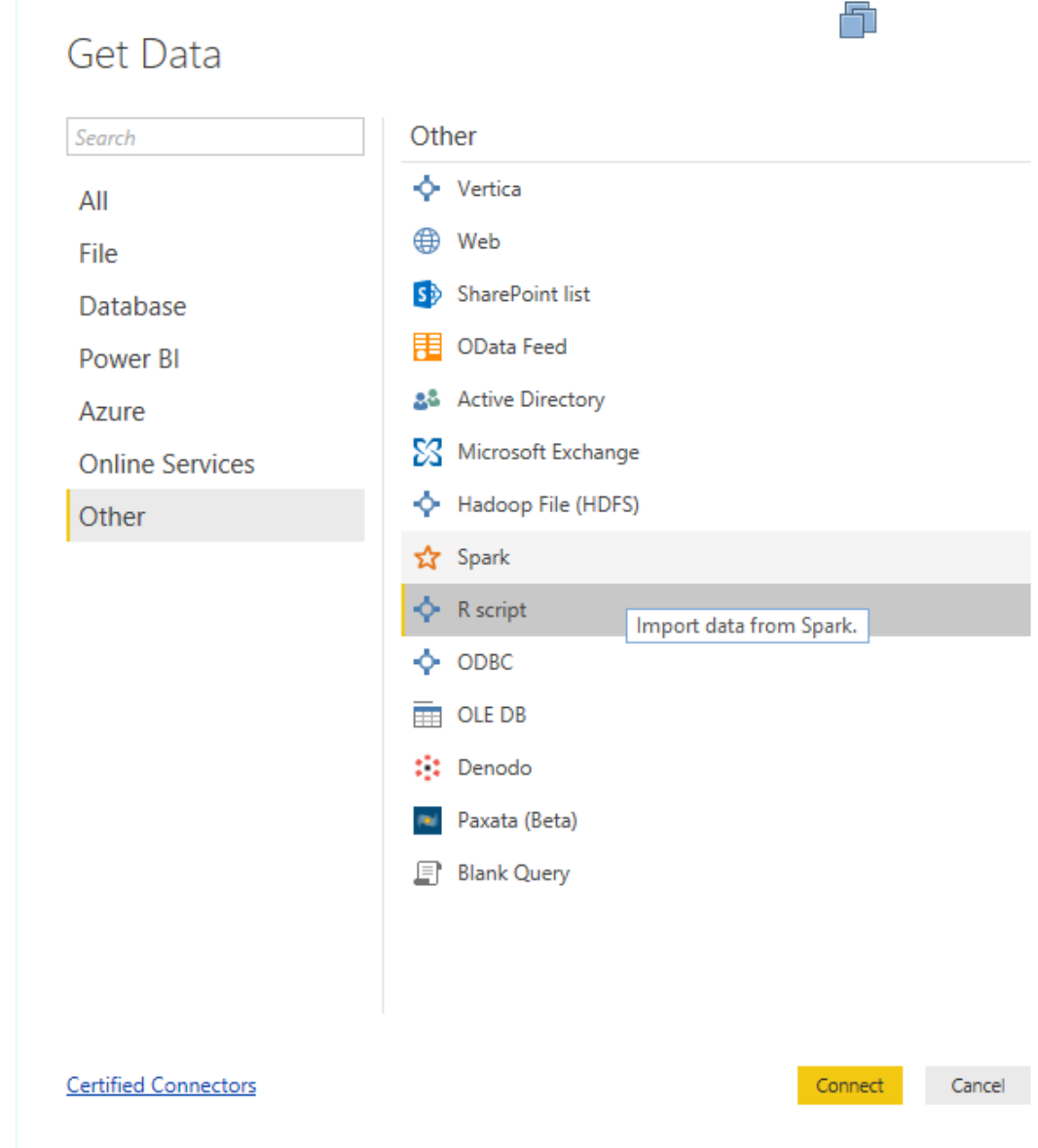
Get Data -> More... ->

Other -> R Script -> Connect Button

Must return a data frame

Can return multiple, and then select which one to use

Complex and Vector types are not supported

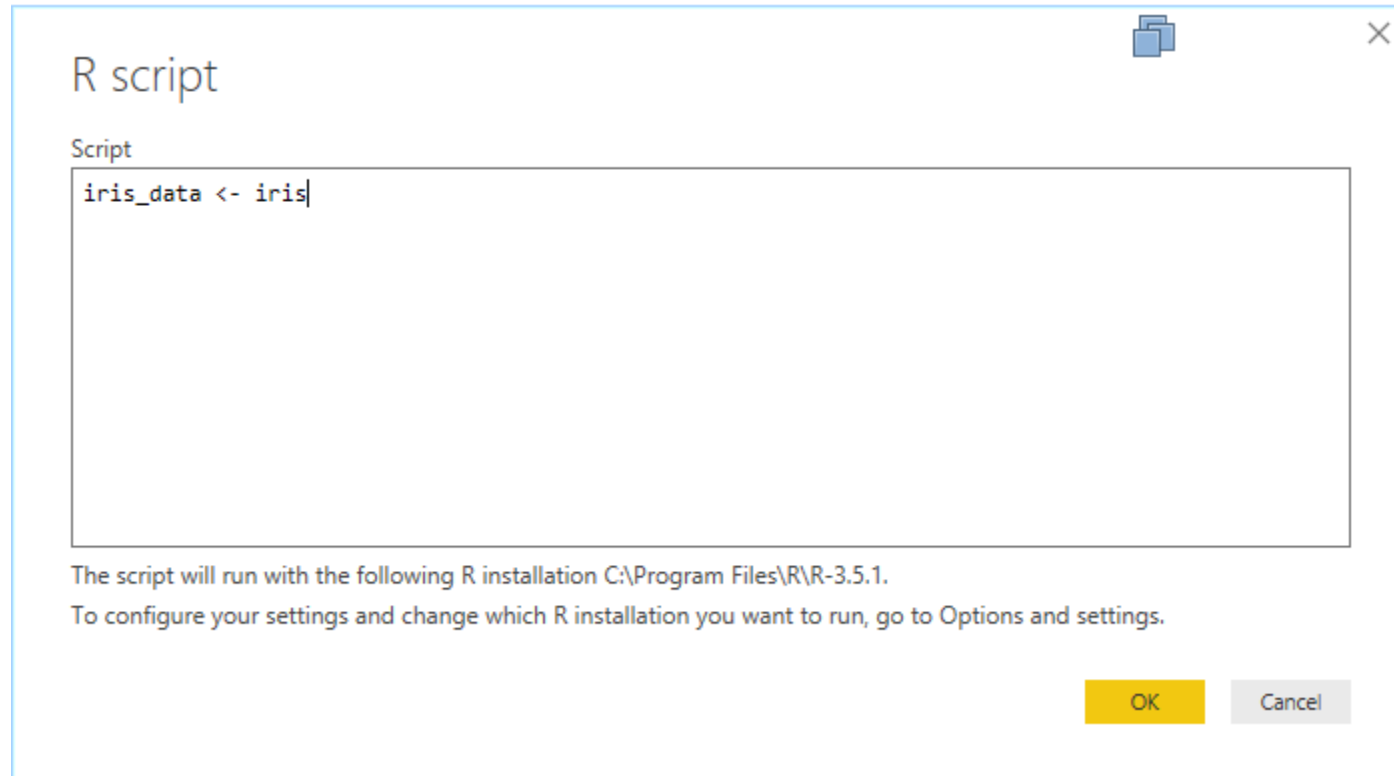


R and PowerBI – Pull Data From R

```
iris_data <- data
```

Must assign the data
to a variable for
PowerBI to pick it up

Click OK



R and PowerBI – Pull Data From R

Data Frame on left

Click the box, and
preview

Click Load

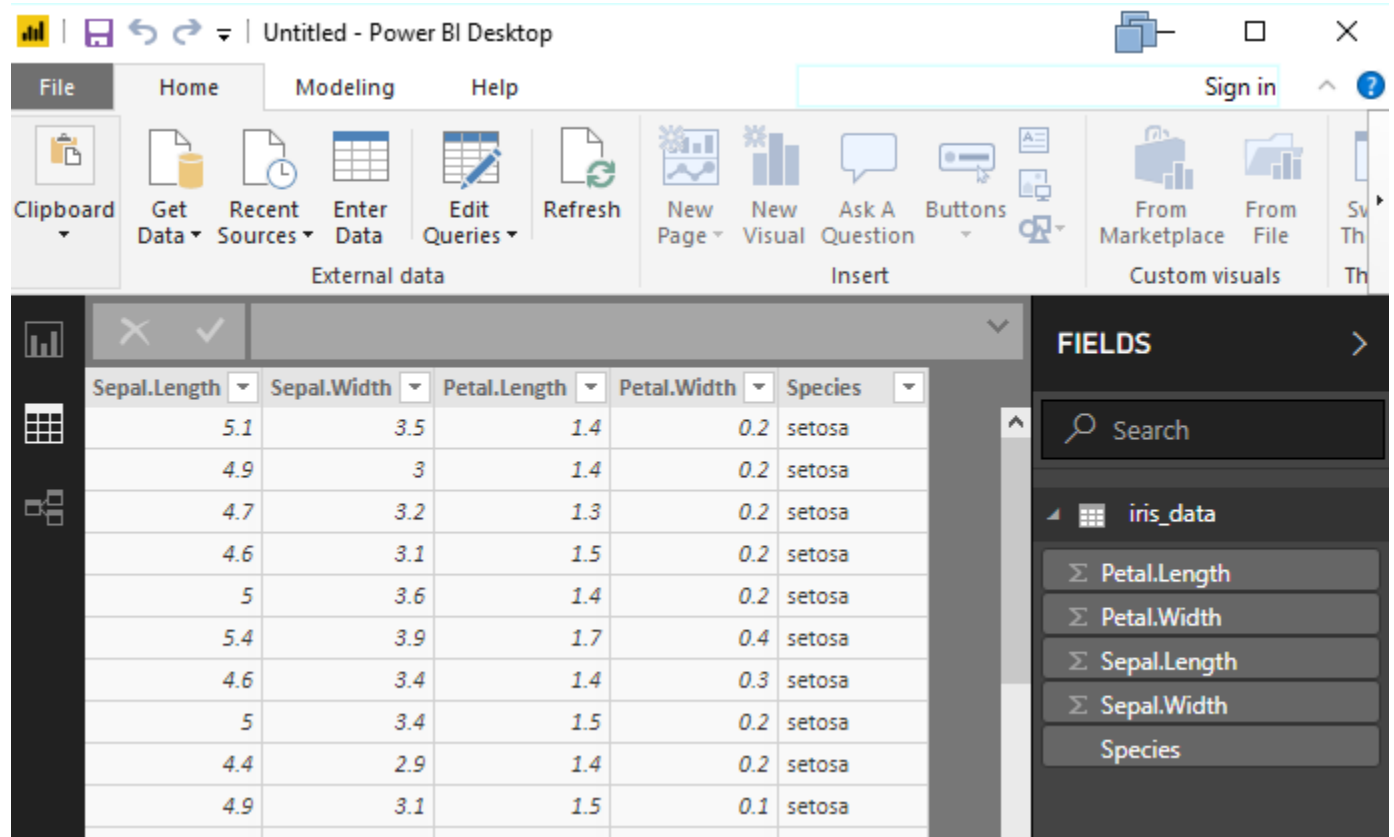
The screenshot shows the PowerBI Navigator window. On the left, under 'R [1]', the 'iris_data' data frame is selected with a checkmark. On the right, a preview of the 'iris_data' data frame is displayed as a table with 50 rows and 5 columns: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species. The table shows various measurements for the 'setosa' species. At the bottom right, there are three buttons: 'Load' (highlighted in yellow), 'Edit', and 'Cancel'.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3	1.4	0.1	setosa
4.3	3	1.1	0.1	setosa
5.8	4	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa
5.4	3.4	1.7	0.2	setosa
5.1	3.7	1.5	0.4	setosa
4.6	3.6	1	0.2	setosa

R and PowerBI – Pull Data From R

Navigator Window

Shows data and field names



The screenshot shows the Power BI Desktop interface. The top ribbon includes tabs for File, Home, Modeling, and Help. The Home tab is active, showing options like Get Data, Recent Sources, Enter Data, Edit Queries, Refresh, New Page, New Visual, Ask A Question, Buttons, From Marketplace, and From File. The Navigator window is open, displaying a table of data with columns: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species. The table contains 10 rows of data. The Fields pane on the right shows a search bar and a list of fields under the name 'iris_data', including Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, and Species.

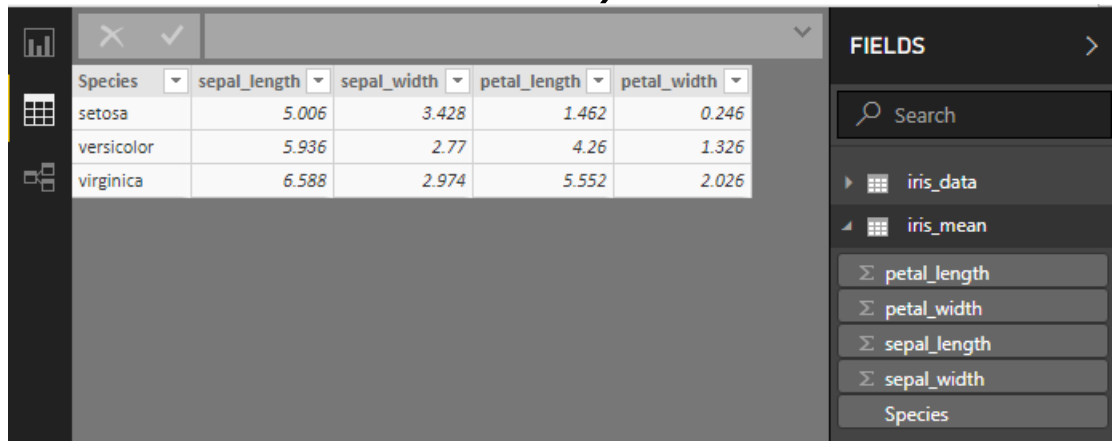
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

R and PowerBI – Pull Data From R

Make R do the scrubbing

```
library(dplyr)
```

```
iris_mean <- summarize(group_by(iris, Species)  
                        , sepal_length = mean(Sepal.Length)  
                        , sepal_width = mean(Sepal.Width)  
                        , petal_length = mean(Petal.Length)  
                        , petal_width = mean(Petal.Width)  
                        )
```



The screenshot shows the PowerBI interface. On the left, a data table displays the results of the R query. The table has five columns: Species, sepal_length, sepal_width, petal_length, and petal_width. The rows represent the three species of the iris dataset: setosa, versicolor, and virginica. On the right, the 'FIELDS' pane is visible, showing a search bar and a list of fields. The 'iris_data' table is expanded, showing its fields: petal_length, petal_width, sepal_length, sepal_width, and Species. The 'iris_mean' table is also visible, showing its fields: petal_length, petal_width, sepal_length, sepal_width, and Species.

Species	sepal_length	sepal_width	petal_length	petal_width
setosa	5.006	3.428	1.462	0.246
versicolor	5.936	2.77	4.26	1.326
virginica	6.588	2.974	5.552	2.026

R and PowerBI - Unleash the R

Unleash the R



R and PowerBI - Unleash the R

PowerBI adds Source and Navigation Steps

The screenshot displays the Power Query Editor window titled "Untitled - Power Query Editor". The ribbon includes tabs for File, Home, Transform, Add Column, View, and Help. The View tab is active, showing options for Formula Bar, Monospaced, Show whitespace, Always allow, Advanced Editor, and Query Dependencies. The main area shows the R script: `= R.Execute("iris_data <- iris")`. Below the script, a data preview table is shown with columns "Name" and "Value". The first row contains "iris_data" and "Table". A blue arrow points to the "Table" value. The left sidebar shows a list of queries: "iris_data" and "iris_mean". The right sidebar shows the "QUERY SETTINGS" panel with sections for "PROPERTIES" (Name: "iris_data") and "APPLIED STEPS" (Source, Navigation).

Queries [2]

- iris_data
- iris_mean

Formula Bar: `= R.Execute("iris_data <- iris")`

	Name	Value
1	iris_data	Table

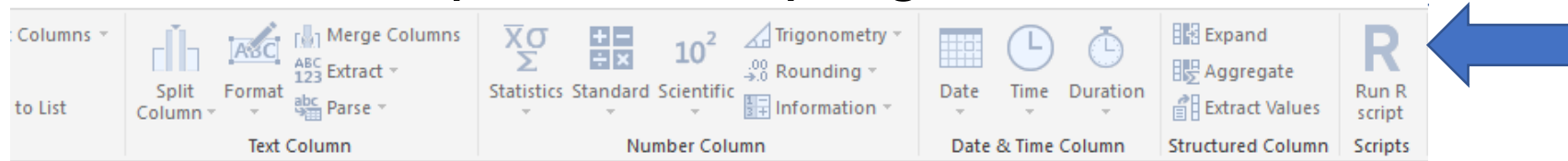
QUERY SETTINGS

- PROPERTIES
 - Name: iris_data
 - [All Properties](#)
- APPLIED STEPS
 - Source
 - Navigation

R and PowerBI - Unleash the R

From the query editor window

Click Run R Script button top Right



The screenshot shows the top toolbar of the PowerBI query editor. The 'Run R script' button, represented by a large 'R' icon, is located in the 'Scripts' section on the far right. A blue arrow points directly to this button. Other sections in the toolbar include 'Text Column', 'Number Column', and 'Date & Time Column'.

Below the toolbar, the 'Run R script' dialog box is open. It contains a text area with the following R script:

```
Script

# 'dataset' holds the input data for this script
library(dplyr)
iris_mean <- summarize(group_by(iris, Species)
                        , sepal_length = mean(Sepal.Length)
                        , sepal_width = mean(Sepal.Width)
                        , petal_length = mean(Petal.Length)
                        , petal_width = mean(Petal.Width)
                        )
```

At the bottom of the dialog, there are instructions: "The script will run with the following R installation C:\Program Files\R\R-3.5.1. To configure your settings and change which R installation you want to run, go to Options and settings." Below these instructions are 'OK' and 'Cancel' buttons.

On the right side of the dialog, the 'QUERY SETTINGS' pane is visible, showing 'PROPERTIES' with the name 'iris_data' and 'APPLIED STEPS' with 'Navigation' selected.

R and PowerBI - Unleash the R

Two Steps added:

Run R Script

iris_mean

The screenshot displays the Microsoft Power BI Desktop interface. At the top, the ribbon shows various data transformation options categorized by data type: Table, Any Column, Text Column, Number Column, and Date & Time Column. Below the ribbon, the 'Queries [2]' pane on the left lists 'iris_data' and 'iris_mean'. The main view shows the 'iris_mean' query with a formula bar containing the R script: `= #"Run R script"{{Name="iris_mean"}}[Value]`. The data table below the formula bar has columns: Species, sepal_length, sepal_width, petal_length, and petal_width, with three rows of data. On the right, the 'QUERY SETTINGS' pane is open, showing the 'APPLIED STEPS' section. A blue arrow points to the 'Run R script' step, which is currently selected and highlighted. Below it, the step '"iris_mean"' is listed with a close icon.

	Species	sepal_length	sepal_width	petal_length	petal_width
1	setosa	5.006	3.428	1.462	0.246
2	versicolor	5.936	2.77	4.26	1.326
3	virginica	6.588	2.974	5.552	2.026

R and PowerBI – Pretty Pictures Please

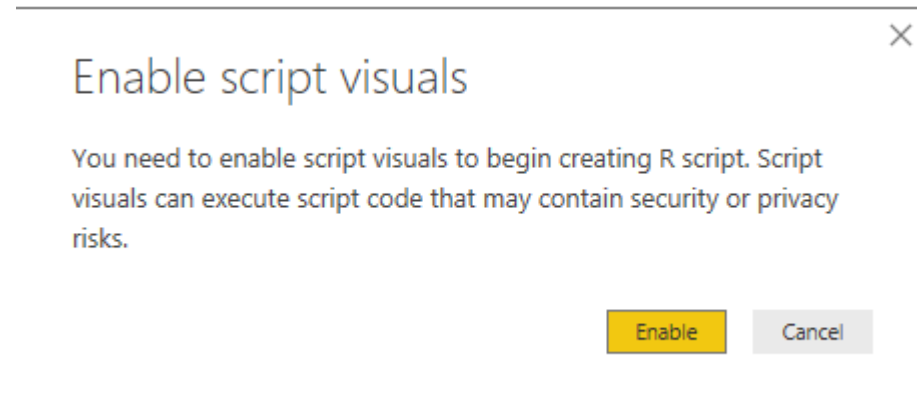
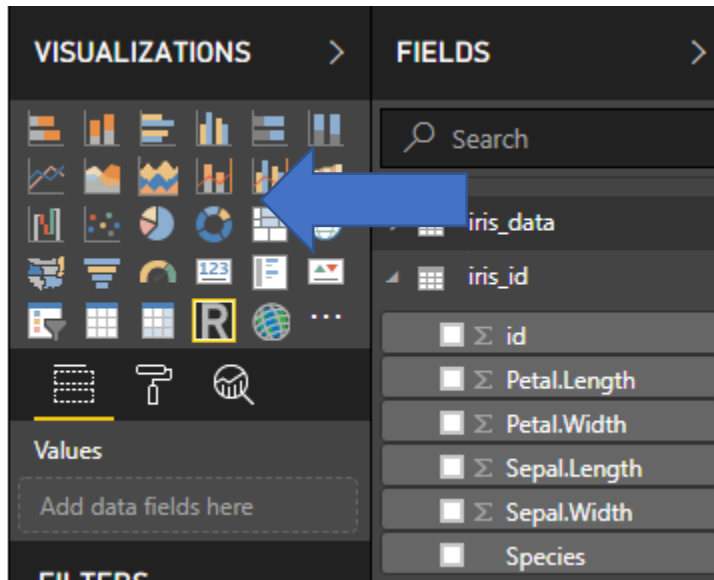


R and PowerBI – Pretty Pictures Please

Load another data frame with the iris data set

Report view and click the R button on the Visualizations pane.

Click Enable for the pop up window



R and PowerBI – Pretty Pictures Please

Copy your code to R script editor

Place code after the comments

You must refer to the data frame as **dataset**

R script editor

⚠ Duplicate rows will be removed from the data.

```
# Create dataframe
# dataset <- data.frame(id, Petal.Length, Petal.Width, Species)

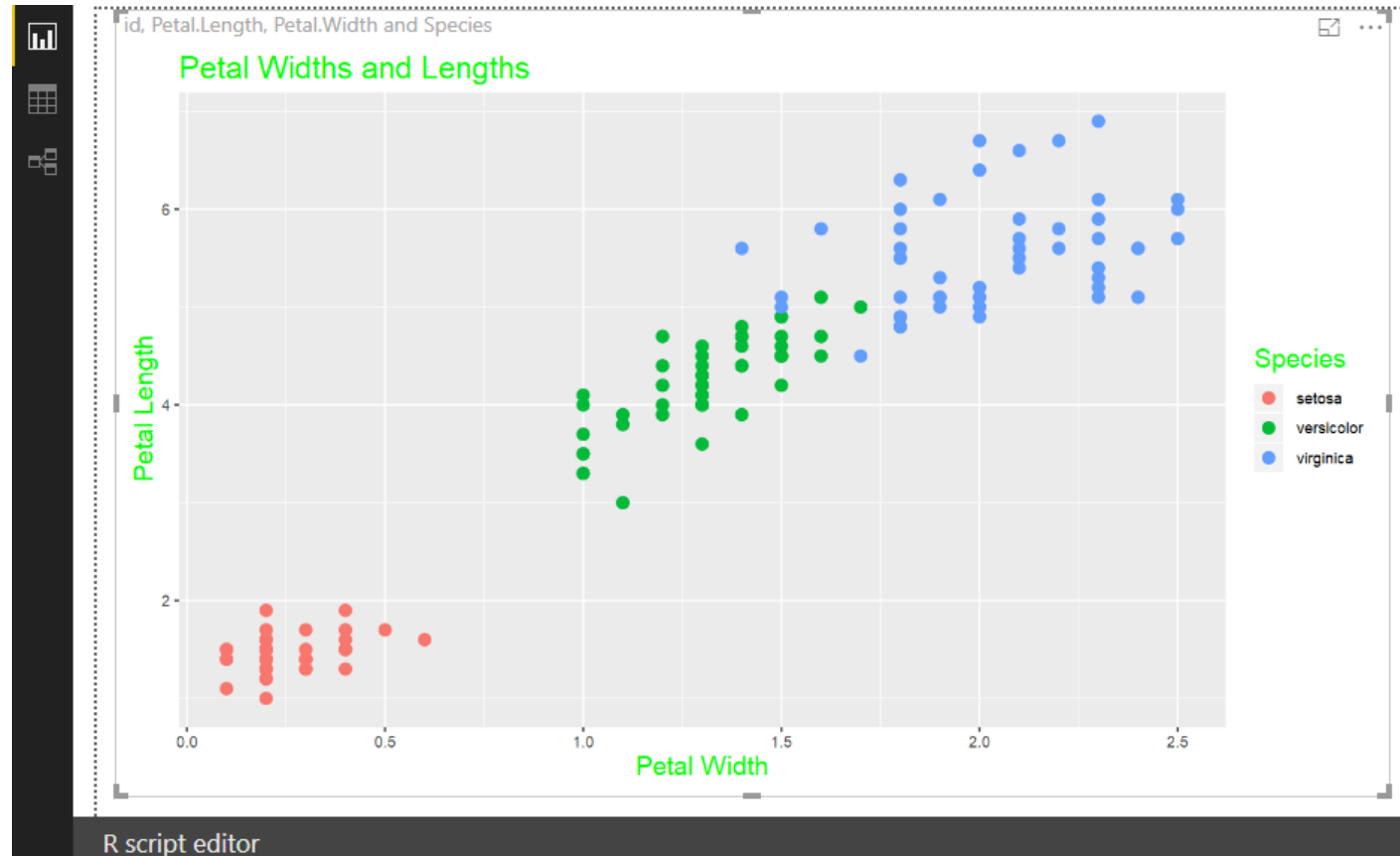
# Remove duplicated rows
# dataset <- unique(dataset)

library(ggplot2)
```

R and PowerBI – Pretty Pictures Please

Click – Run Script button

After re-sizing the graph, ctrl-r to refresh



R and PowerBI – Pretty Pictures

Please

1. Run Script
2. Script Options
3. Edit in IDE



R and PowerBI – Pretty Pictures

Please

Custom Visualizations

Click ellipses in visualization pane and can import R visualizations

3 Lightning Topics

References

Excel and R

<https://bert-toolkit.com/>

R and PowerBI

<https://powerbi.microsoft.com/en-us/downloads/>

<https://www.red-gate.com/simple-talk/sql/bi/power-bi-introduction-working-with-r-scripts-in-power-bi-desktop-part-3/>

<https://community.powerbi.com/t5/R-Script-Showcase/bd-p/RVisuals>