

Introduction to R in Power BI

Overview of R Scripting



Overview

What is R?

Connect Data

Why use R in Power BI?

Transform Data

Setup R

Create Visuals



What is R?

R is a language and environment for statistical computing and graphics.

R provides a wide variety of statistical and graphical techniques, and is highly extensible.



Why use R in Power BI?

Easily expand the range of available **data sources**

Enrich and implement powerful **data processing logic**

Create highly **customizable visualizations**

Build reusable **custom visual elements**



Setup R

Installation

Setup R

Setup Power BI



**Power BI: Introduction to
Business Analytics**

Course Author: Joshua Michalik

Installation

Install Microsoft Open R

<https://mran.revolutionanalytics.com/download>

Install R Studio

<https://www.rstudio.com/products/rstudio/download/>



Setup R

Open R Studio

Install R packages

Use `install.packages(dplyr)`

dplyr

A fast, consistent tool for working with data frame like objects

ggplot2

Create Elegant Data Visualisations Using the Grammar of Graphics

data.table

Offers a natural and flexible syntax, for faster development.



Setup Power BI

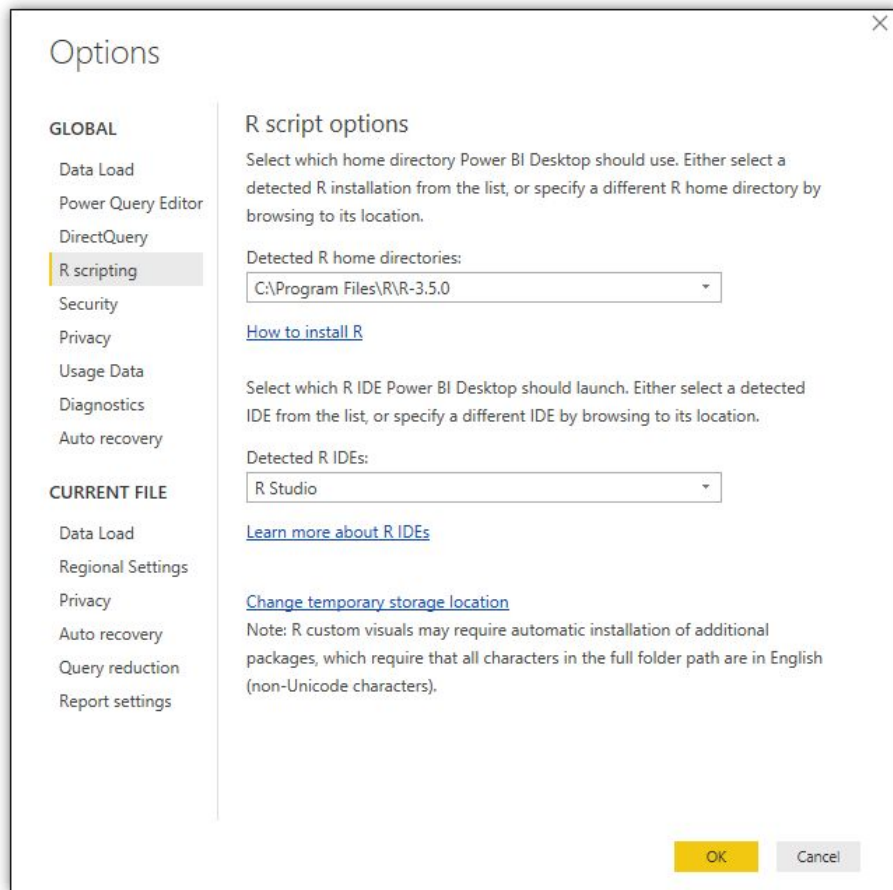
Open *File > Options and settings > Options*. Under GLOBAL, click *R scripting*

Home directories

C:\Program Files\R\R-3.x

Detected R IDE

R Studio



Helpful Tips

Use a separate R IDE when using R in Power BI

Avoid using datasets with complex and vector type columns when using R in Power BI



Connect Data

Datasets in R

Using R scripts to *Get Data*

Examples



**Power BI: Introduction to
Business Analytics**

Course Author: Joshua Michalik

Datasets in R

Run `data()` to see entire list

The screenshot shows the RStudio interface. The 'R data sets' window is open, displaying a list of datasets. The 'mtcars' dataset is highlighted. The console shows the following R code and output:

```
> boxplot(count ~ spray, data = InsectSprays, col = "lightgray")
> # *add* notches (somewhat funny here):
> boxplot(count ~ spray, data = InsectSprays,
+         notch = TRUE, add = TRUE, col = "blue")
Warning message:
In bxp(list(stats = c(7, 11, 14, 18.5, 23, 7, 12, 16.5, 18, 21, :
  Quelques indentations ("notches") dépassent des jointures ("hinges") ('box') : utilisez peut-être notch=FALSE
> data()
>
```

The boxplot on the right shows the distribution of 'count' for different 'spray' categories (A, B, D, F). The y-axis ranges from 0 to 20. The boxplots are colored blue, and notches are visible on the boxes.



Using R scripts to *Get Data*

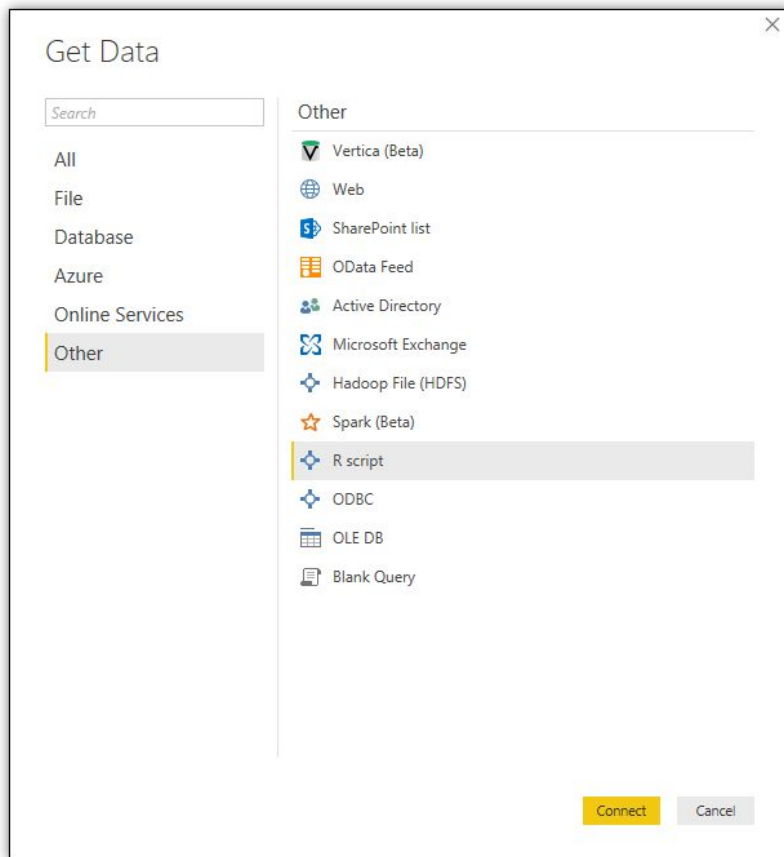
Select ***Get Data > Other > R script***

Paste R script into **Script box**

Click **OK**

Editing R Scripts

Open *Query Editor* >
double click on *Source*



Examples

Basic import

```
iris_raw <- iris
```

Advanced import with data processing

```
library(dplyr)
iris_mean <- summarize(group_by(iris, Species),
                        slength = mean(Sepal.Length), swidth = mean(Sepal.Width),
                        plength = mean(Petal.Length),| pwidth = mean(Petal.Width))
```



Transform Data

Applied Steps

Using R scripts in Query Editor



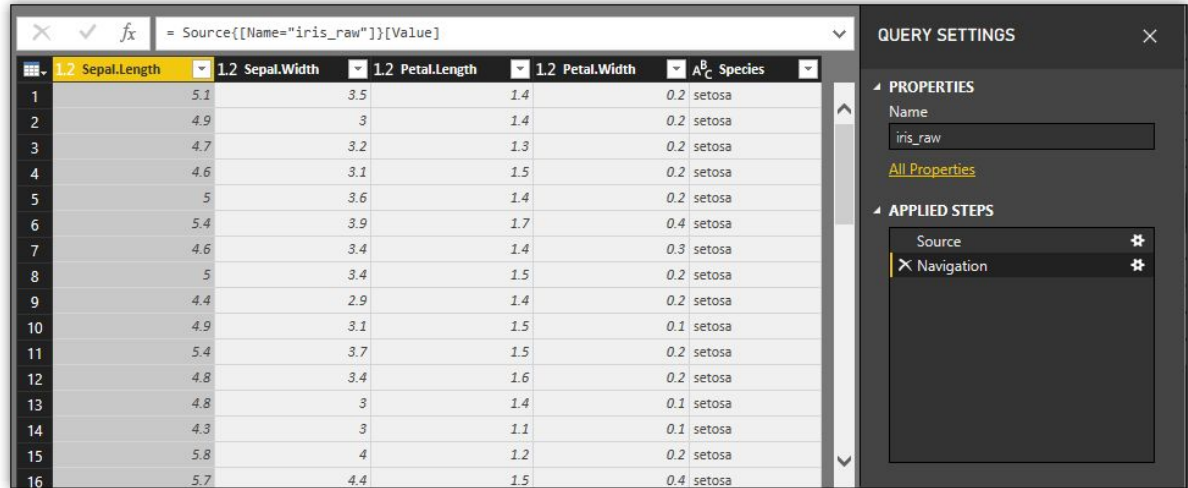
**Power BI: Introduction to
Business Analytics**

Course Author: Joshua Michalik

Applied Steps

In Power BI Desktop, click *Edit Queries* to navigate to Query Editor.

Right hand side shows *Applied Steps*



The screenshot displays the Power BI Query Editor interface. The main area shows a table with 16 rows of data. The columns are: 1.2 Sepal.Length, 1.2 Sepal.Width, 1.2 Petal.Length, 1.2 Petal.Width, and A^B_C Species. The formula bar at the top shows the query: = Source{[Name="iris_raw"]}[Value]. On the right side, the 'QUERY SETTINGS' pane is open, showing the 'APPLIED STEPS' section with a list of steps: 'Source' and 'Navigation'.

	1.2 Sepal.Length	1.2 Sepal.Width	1.2 Petal.Length	1.2 Petal.Width	A ^B _C Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3	1.4	0.1	setosa
14	4.3	3	1.1	0.1	setosa
15	5.8	4	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa

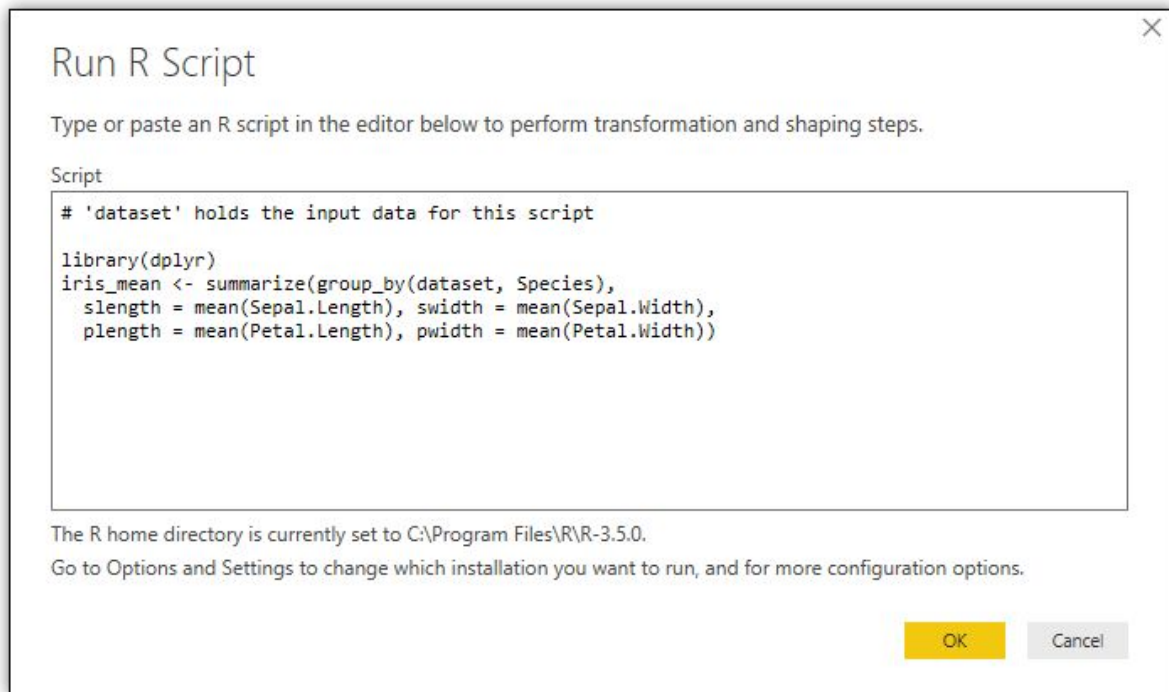


Using R scripts in *Query Editor*

In Query Editor, under Transform, click Run R script.

Input data at steps is referred to as *dataset*.

R scripts work on any dataset.



Create Visuals

Prepping Data

Adding Values to Visuals

Using R script Visuals



Preparing Data

Power BI automatically removes duplicate lines.

Use this workaround to add an ID column to a table.

```
library(data.table)
iris_id <- iris
iris_id <- setDT(iris_id, keep.rownames=TRUE)[]
setnames(iris_id, 1, "id")
iris_id$id <- as.integer(iris_id$id)
```



Adding Values to Visuals

All data to be used gets added to Values section first.

Drag or Checkbox

The screenshot displays the Power BI desktop application interface. The central area is the R script editor, which contains the following code:

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

# Remove duplicated rows
# dataset <- unique(dataset)
Paste or type your R-script code here
```

A yellow notification bar at the top of the script editor states: "Duplicate rows were removed from the data." Below the script editor, the "Page 1" tab is visible.

On the right side, the "FIELDS" pane is open, showing a search bar and a list of fields. The "iris_id" field is selected, and its sub-fields are listed with checkboxes:

- ☒ id
- ☒ Petal.Length
- ☒ Petal.Width
- ☐ Sepal.Length
- ☐ Sepal.Width
- ☒ Species

Below the "VALUES" section, the "FILTERS" section is visible, showing a list of filters:

- id (All)
- Petal.Length (All)
- Petal.Width (All)
- Species (All)



Using R script Visuals

R script editor

Edit your R code here

Edit script in External IDE

Use this to edit your R code in R Studio

Run script

Displays the visual in the Report screen



Review



Review

How would you connect to data sources using R?

What kind of data transformations can you make with R?

What visuals are most commonly made with R?

