Intro to R Visualization with Power BI

April 5th, 2017

datasciencedojo

data science for everyone

Who Am I?

- Dave Langer, VP of Data Science – Data Science Dojo

- 20+ years in technology:
  - Roles in development, architecture, & BI/DW/analytics.
  - Last job – Sr. Director, BI & Analytics @ Microsoft.

- Hooked on Data Science 5 years ago:
  - Extensive background in data and analytics.
  - Learned Machine Learning from 2nd place Netflix Prize winner.
  - #1 Data Scientist on YouTube.

- Joined Data Science Dojo to democratize Data Science.

Motivation

- Power BI is core to Microsoft’s analytics strategy:
    - Sustained Engineering investment.
    - Cloud, On-Premises, and Hybrid.
    - Growing community of contributors.

- Support for R Visualizations!

Why Power BI?

- Highly productive environment for:
  - Data exploration and analysis.
  - Executive Scorecards.
  - Operational Dashboards.

- Excellent solution for descriptive analytics.

- Power BI Desktop is free*!

* At the time of this presentation.

Expectation Setting

- I am assuming the following:
  - You are familiar with Power BI
  - You are familiar with R
  - You prefer R over DAX

- This is not a Power BI tutorial.
- This is not a R programming tutorial.

- We will cover practical aspects of using R visualizations from within Power BI.

Prerequisites

- To follow along you will need the following:
  - Power BI Desktop
  - R
  - RStudio

- The following R packages are required to follow along:
  - dplyr, lubridate, ggplot2, scales, qcc

- The GitHub repo has source, data, and slide files.

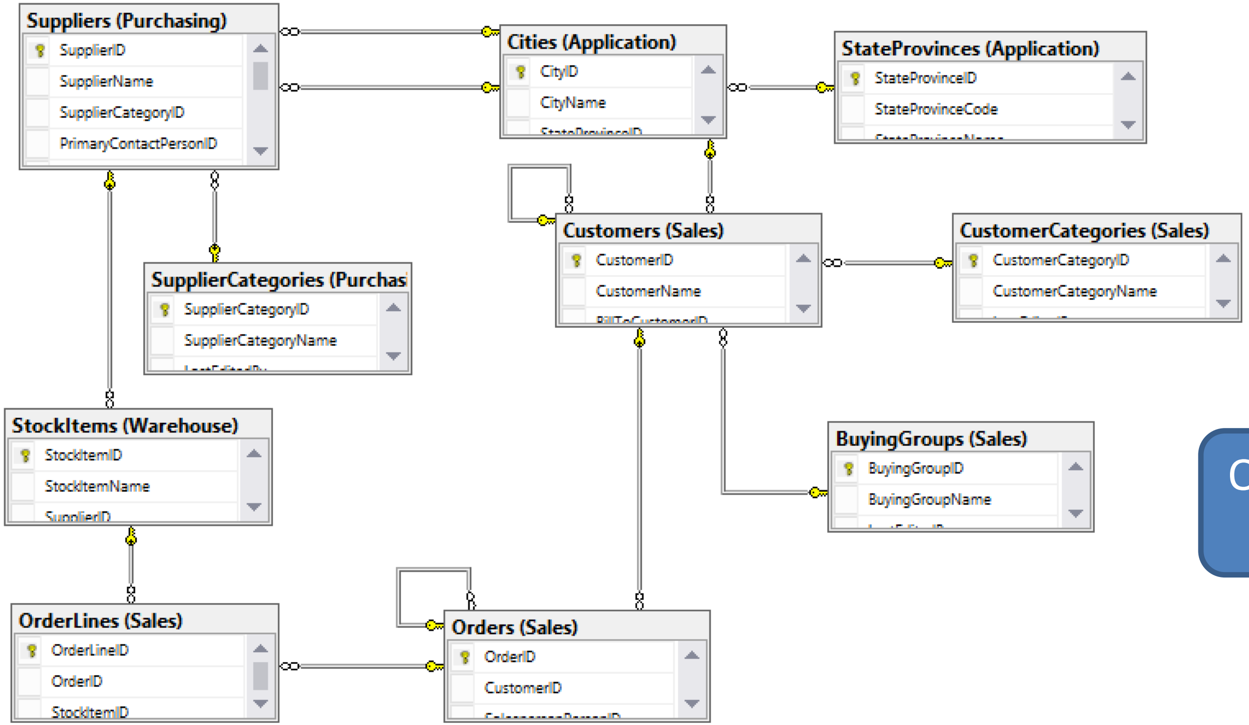
The Scenario

- You are an Analyst for Wide World Importers and have been asked to conduct an analysis on WWI Customers.
- You have access to the WWI Database.
- You have access to Power BI, but are more comfortable using R than DAX in your work.

THE DATA

datasciencedojo
data science for everyone

The Database



Suppliers (Purchasing)
SupplierID
SupplierName
SupplierCategoryID
PrimaryContactPersonID

Cities (Application)
CityID
CityName
StateProvinceID

StateProvinces (Application)
StateProvinceID
StateProvinceCode
StateProvinceName

SupplierCategories (Purchas
SupplierCategoryID
SupplierCategoryName
LastEditedBy

Customers (Sales)
CustomerID
CustomerName
BillToCustomerID

CustomerCategories (Sales)
CustomerCategoryID
CustomerCategoryName
LastEditedBy

StockItems (Warehouse)
StockItemID
StockItemName
SupplierID

BuyingGroups (Sales)
BuyingGroupID
BuyingGroupName
LastEditedBy

OrderLines (Sales)
OrderLineID
OrderID
StockItemID

Orders (Sales)
OrderID
CustomerID
SalespersonPersonID

Normalized schema, common for transactional systems.

Option #1 – Pull all tables into Power BI.

Option #2– Use SQL query to denormalize data

datasciencedojo
data science for everyone

POWER BI DESKTOP

USING R WITH POWER BI

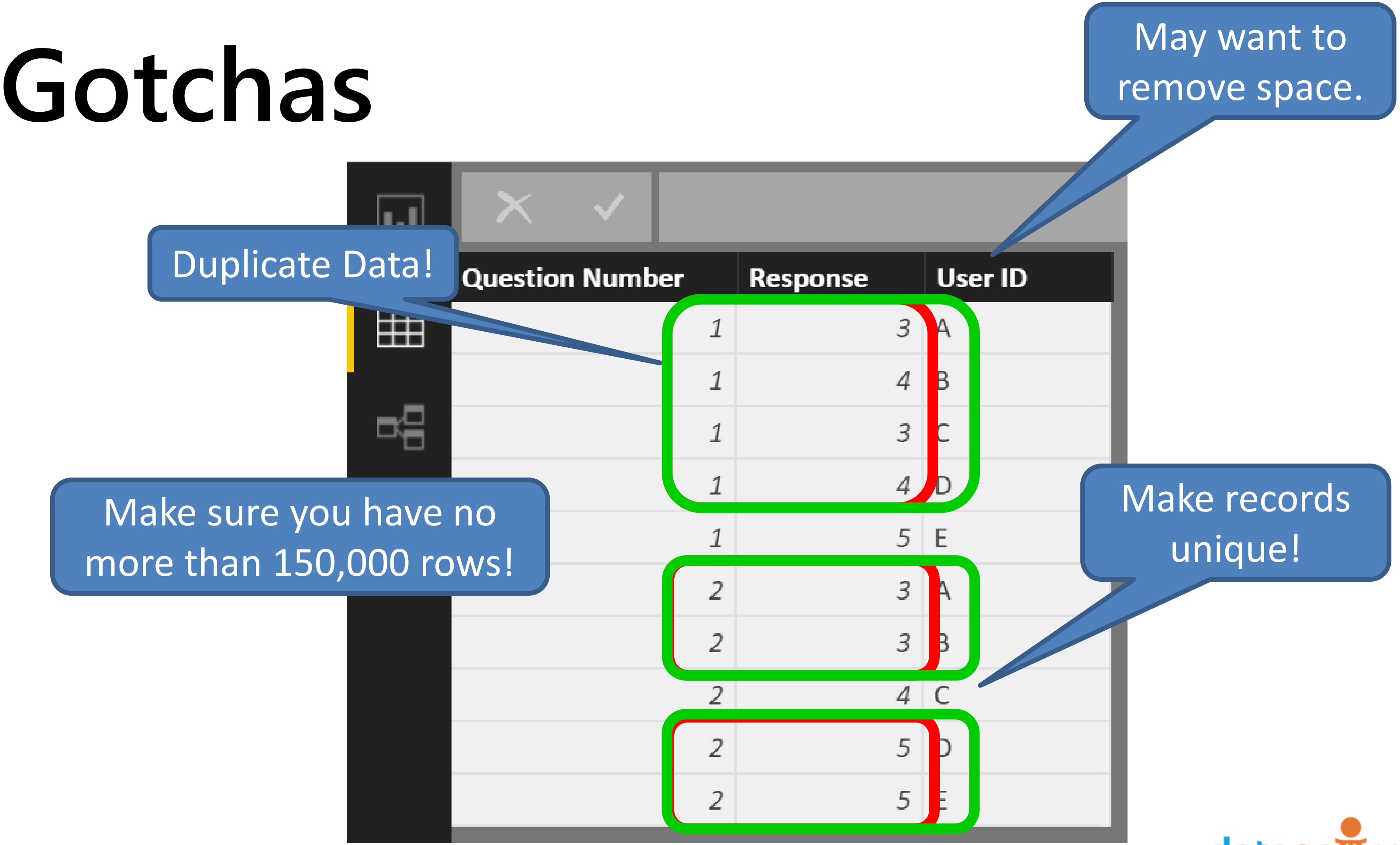
datasciencedojo

data science for everyone

Gotchas

- Power BI limits data to R at 150,000 rows.
- Power BI automatically drops duplicate rows:
  - Leveraging PKs is a good idea!
- Power BI allows for very permissive column names.

Gotchas



Question Number	Response	User ID
1	3	A
1	4	B
1	3	C
1	4	D
1	5	E
2	3	A
2	3	B
2	4	C
2	5	D
2	5	E

May want to remove space.

Duplicate Data!

Make sure you have no more than 150,000 rows!

Make records unique!

Some Best Practices

- Prefer native Power BI visuals:
  - https://app.powerbi.com/visuals/

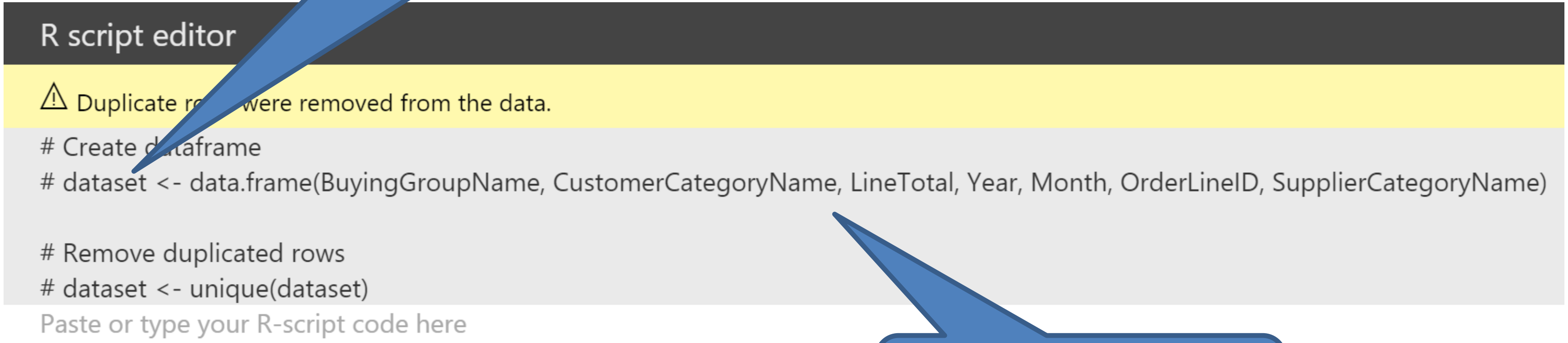
- Change columns names to be R-friendly.

- Use Power BI Slicers to reduce data sent to R.

- Use R Studio, Visual Studio R Tools, etc. to develop your visuals:
  - Coding & debugging R in Power BI is painful.
  - Write your R code to work with Power BI.

Power BI’s Interface with R

The name of your data frame is “dataset”!



R script editor

⚠ Duplicate rows were removed from the data.

```
# Create dataframe
# dataset <- data.frame(BuyingGroupName, CustomerCategoryName, LineTotal, Year, Month, OrderLineID, SupplierCategoryName)

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

Paste or type your R-script code here

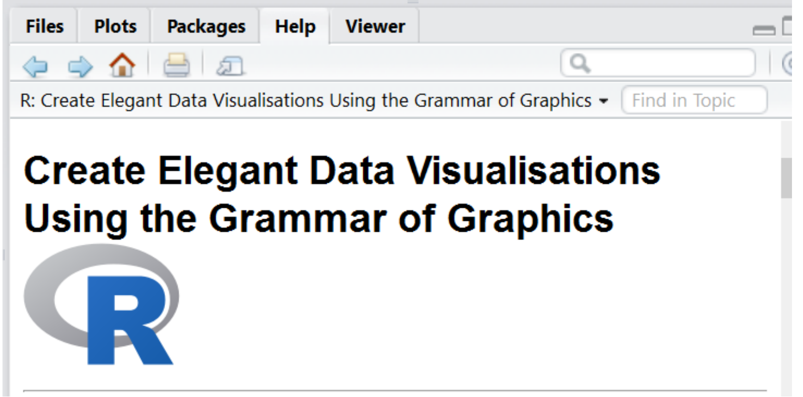
The names of your variables!

Copy and paste your R code here!

GGPLOT2 & DPLYR

ggplot2

- De facto standard visualization package in R.
- Designed for print-quality graphics.



R: Create Elegant Data Visualisations Using the Grammar of Graphics

Create Elegant Data Visualisations Using the Grammar of Graphics

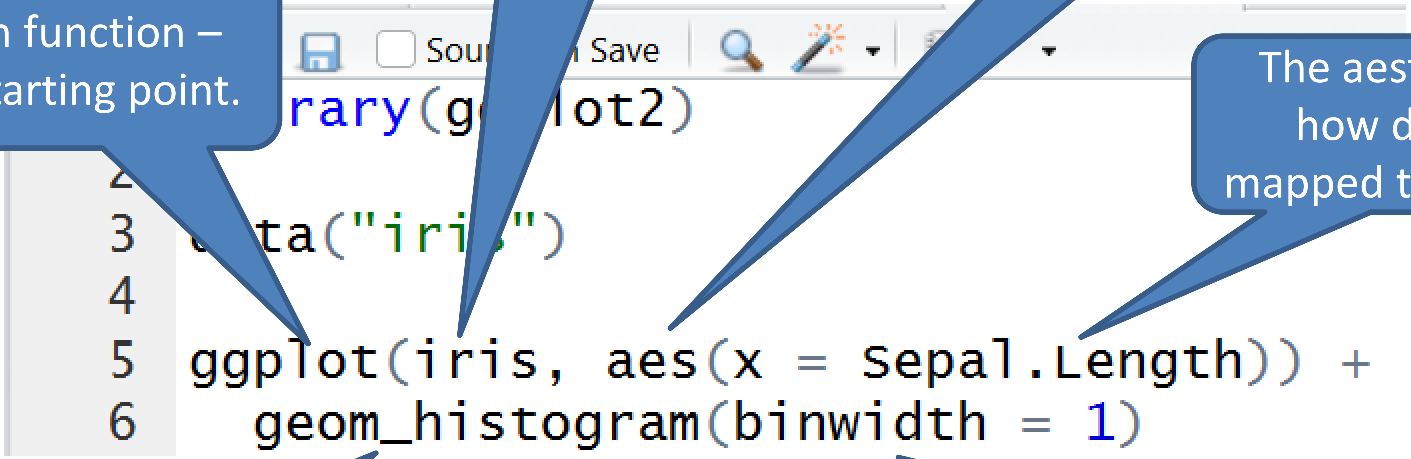
- Fine-grained control via an API focusing on layering graphical elements to build visualizations.

ggplot2

The collection of data that we're working with.

The aesthetic – how data is mapped to visuals.

Main function – the starting point.



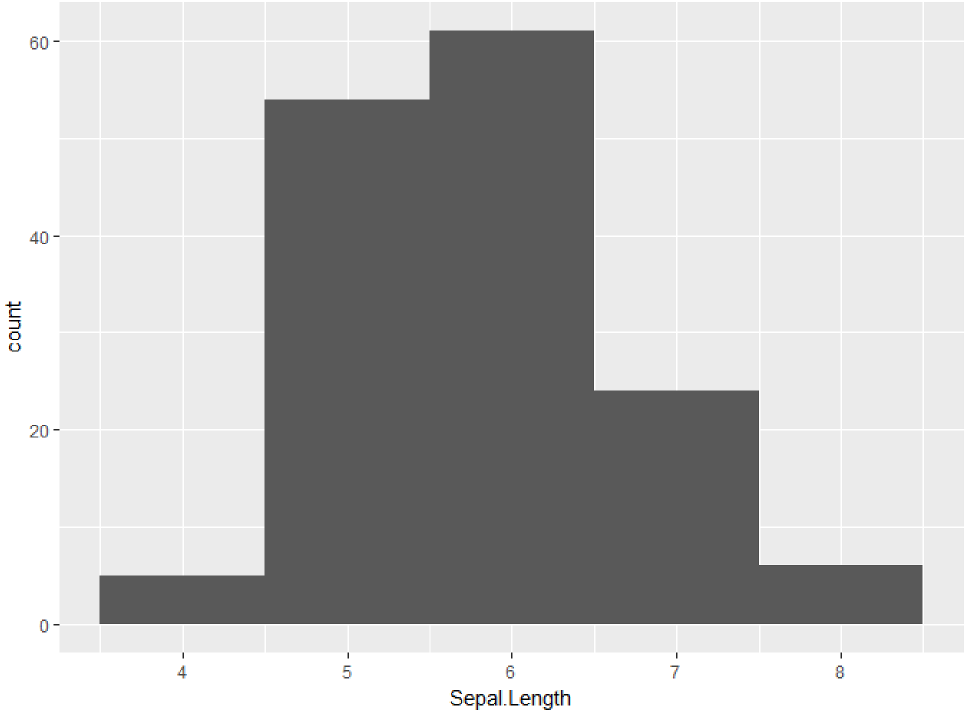
```
rary(g lot2)

ta("iri ")

ggplot(iris, aes(x = Sepal.Length)) +
  geom_histogram(binwidth = 1)
```

The aesthetic – how data is mapped to visuals.

A visual to layer on to the canvas.

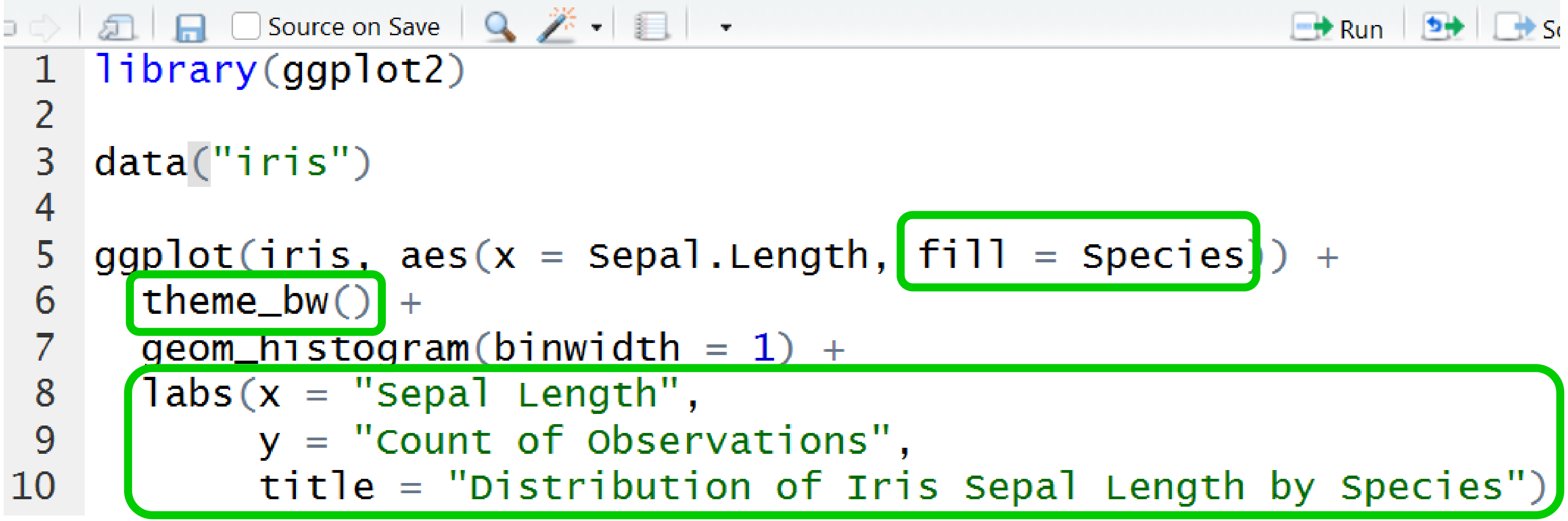
Visual parameters – control the look.

ggplot2



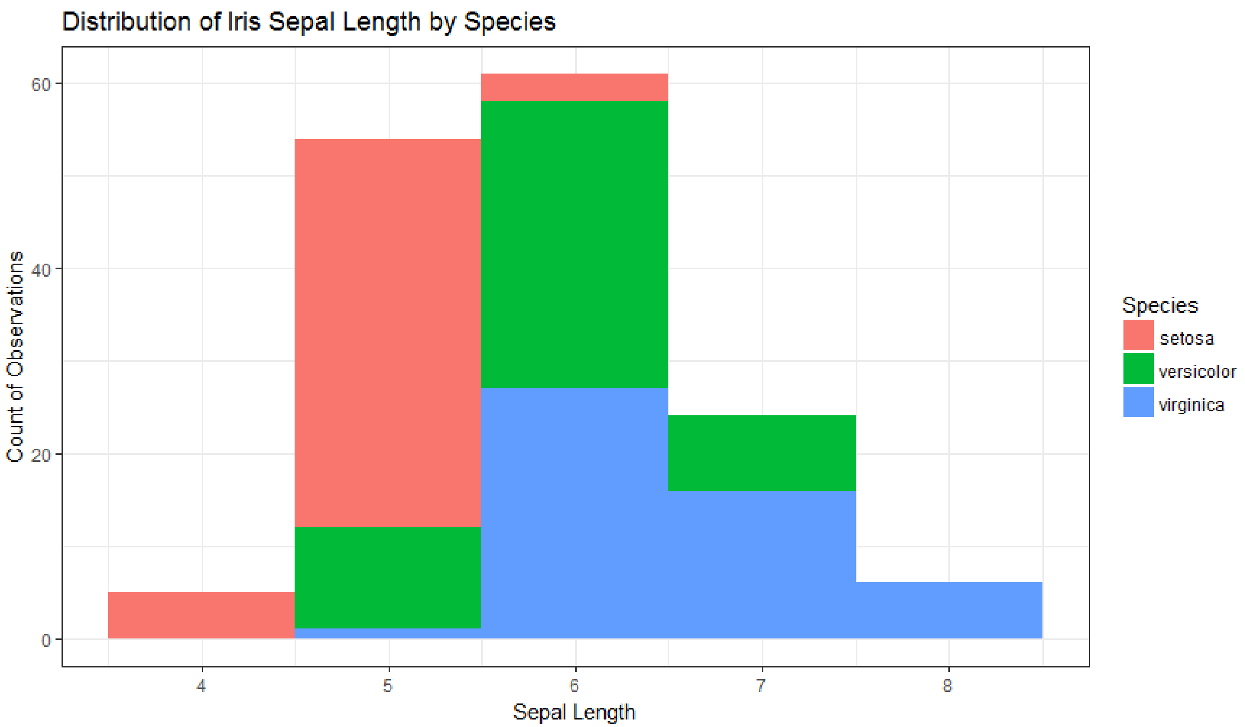
ggplot2

Incrementally build more powerful visualizations



```r
library(ggplot2)

data("iris")

ggplot(iris, aes(x = Sepal.Length, fill = Species)) +
  theme_bw() +
  geom_histogram(binwidth = 1) +
  labs(x = "Sepal Length",
       y = "Count of Observations",
       title = "Distribution of Iris Sepal Length by Species")
```

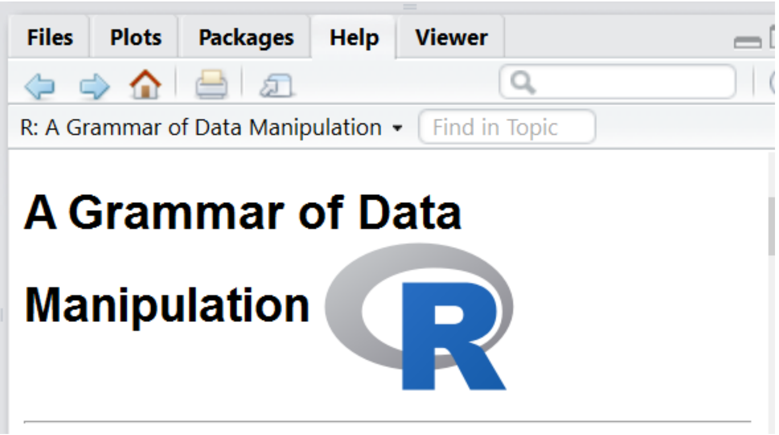
ggplot2


Distribution of Iris Sepal Length by Species
Count of Observations
60
40
20
0
4
5
6
7
8
Sepal Length
Species
setosa
versicolor
virginica

dplyr

- Popular package for data wrangling.

- Intuitive coding style based on data pieplines.



A Grammar of Data Manipulation

- Very intuitive to those with familiarity with SQL – just about everything you can do with SQL you can do with dplyr.

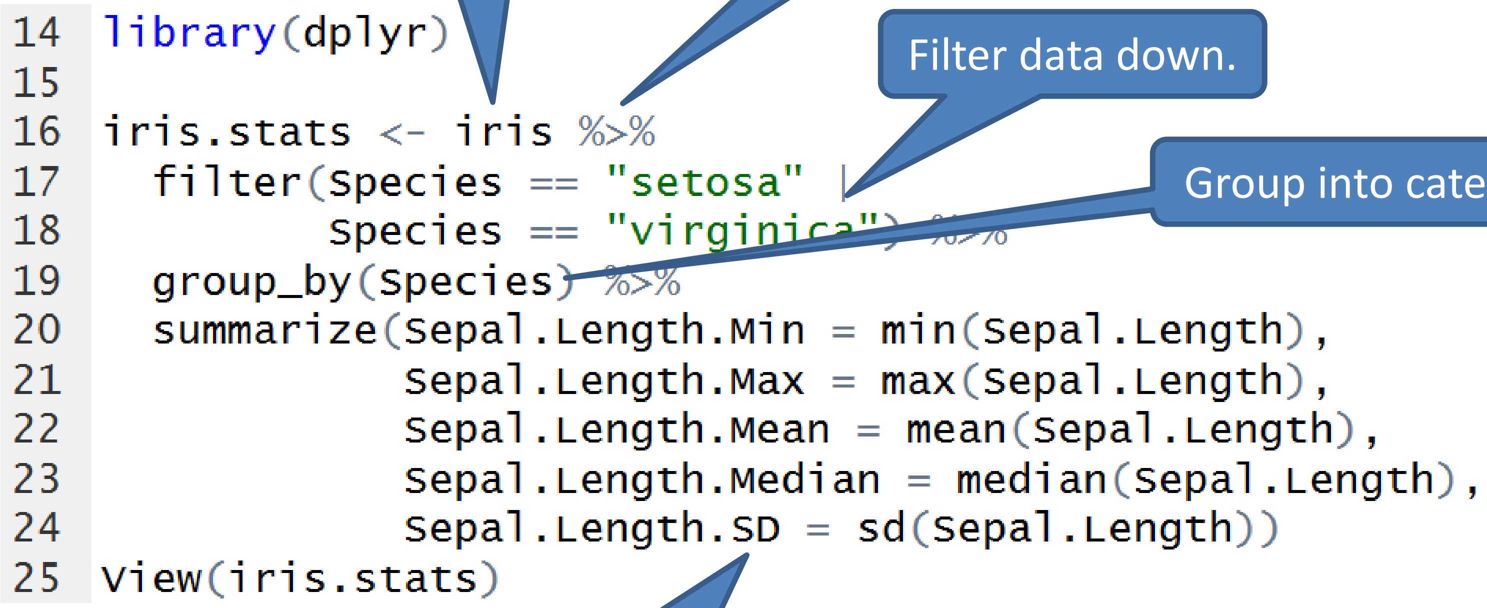
dplyr

The data we’re working with.

Pipe data to the next operation.

Filter data down.

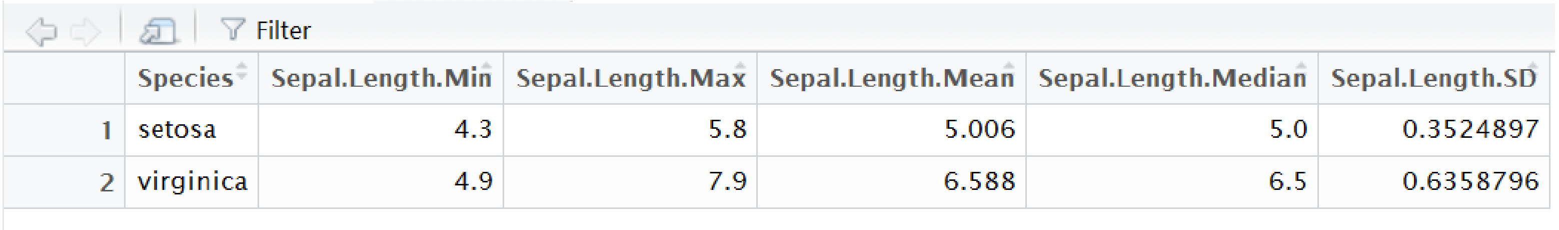
Group into categories.



```r
library(dplyr)

iris.stats <- iris %>%
  filter(Species == "setosa" |
         Species == "virginica") %>%
  group_by(Species) %>%
  summarize(Sepal.Length.Min = min(Sepal.Length),
            Sepal.Length.Max = max(Sepal.Length),
            Sepal.Length.Mean = mean(Sepal.Length),
            Sepal.Length.Median = median(Sepal.Length),
            Sepal.Length.SD = sd(Sepal.Length))
View(iris.stats)
```

Summarize groups.

dplyr



	Species	Sepal.Length.Min	Sepal.Length.Max	Sepal.Length.Mean	Sepal.Length.Median	Sepal.Length.SD
1	setosa	4.3	5.8	5.006	5.0	0.3524897
2	virginica	4.9	7.9	6.588	6.5	0.6358796

R CODE!

QUESTIONS

APPENDIX

Get the Files!

- GitHub Repo:
  - https://github.com/datasciencedojo/meetup/tree/master/r_visualization_with_power_bi

- World Wide Importers DB:
  - https://github.com/Microsoft/sql-server-samples/releases/tag/wide-world-importers-v1.0