UWES R Tutorial

Andrew Girgis

Discord QR Code

Download R and RStudio

This will be a hands on workshop so I encourage all participants to download R and RStudio before the workshop and follow along.

R is the programming language and RStudio is an integrated development environment (IDE) for R. You need to install both before you are ready to start programming.

- **Download R:** https://cran.r-project.org/
- **Download RStudio:** https://posit.co/download/rstudio-desktop/

Overview

- Introduction to R
  - What is R?
  - Why use R?
  - R vs Excel
- Data structures in R
  - Vectors, Matrices, Data frames, Lists
- loading and viewing data
  - Loading built-in datasets
  - Reading CSV files
  - Subsetting data
- Basic data analysis in R
  - Summary statistics
  - Simple Visualization

Objective + Classes

Objective

- Understand the basics of R programming.
- Get familiar and comfortable with RStudio IDE.
- Learn how to load and visualize data in R.
- Prepare you for future courses and research projects

Classes

- ECON 211: Introduction to Mathematical Economics
- ECON 221: Statistics for Economics
- ECON 322: Econometric Analysis 1
- ECON 323: Econometric Analysis 2
- + Many more

Getting started with R

R and RStudio are free and open source software environment for statistical computing and graphics. It compiles and runs on a wide variety of platforms, including Windows and MacOS. R has a large and active user community and is widely used in academia and industry. It has many built in tools for statistical data analysis and visualization.

RStudio IDE

RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

Lets look at components of the RStudio IDE.

RStudio IDE

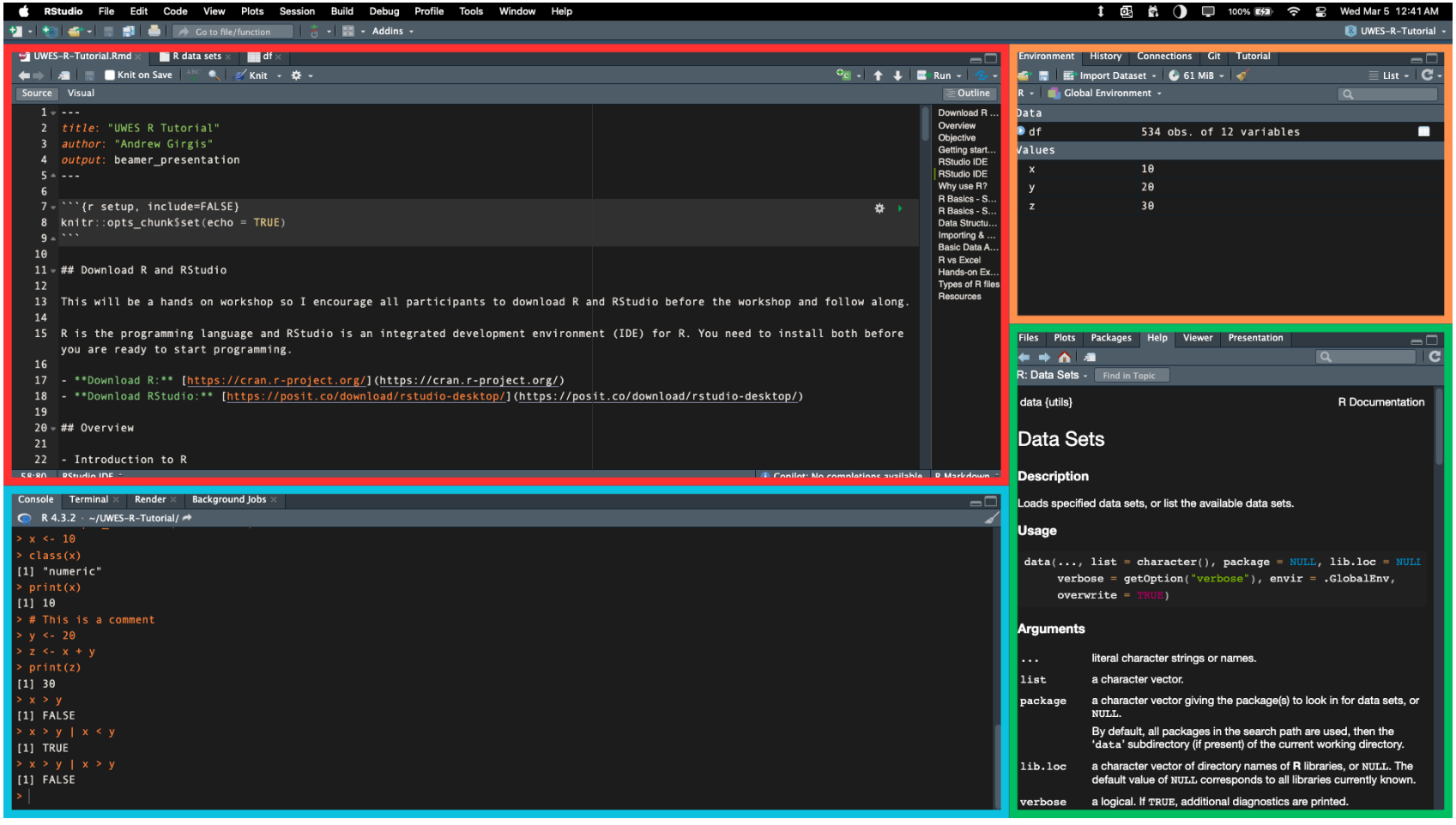


Figure 2: RStudio IDE

RStudio IDE

- Source Editor: For writing and executing R scripts.
- Console: For executing R commands.
- Environment/History: For viewing objects and history.
- Files/Plots/Packages/Help: For managing files, plots, packages, and help.

Why use R?

- R's role in data science and economic research.
- Comparison with other tools like Excel, Python, and Stata.
- Real-world examples of economists and analysts using R.

R Basics - Syntax & Data Types

- Variables and assignment

```
x <- 10
```

- Data types: Numeric, Character, Logical, Factor

```
class(x)
```

```
## [1] "numeric"
```

- Printing values

```
print(x)
```

```
## [1] 10
```

- Comments in R

```
# This is a comment
```

R Basics - Syntax & Data Types

- Arithmetic operations

```
y <- 20
z <- x + y
print(z)
```

```
## [1] 30
```

- Comparison operators

```
x > y
```

```
## [1] FALSE
```

- Logical operators

```
x > y | x > y
```

```
## [1] FALSE
```

R Basics - Syntax & Data Types

Data Type	**Example**	**Description**
Numeric	`x <- 10.5`	Decimal numbers (double/float)
Integer	`x <- 10L`	Whole numbers
Character	`x <- "Hello"`	Text or string values
Logical	`x <- TRUE`	Boolean values (`TRUE` or `FALSE`)
Factor	`"Male", "Female"`	Categorical variables
Complex	`x <- 3 + 2i`	Complex numbers

Table 1: Data Types in R

Data Structures in R

- Vectors

```
v <- c(1, 2, 3, 4, 5, 6)
```

- Matrices

```
m <- matrix(v, nrow = 3)
```

- Data frames

```
df <- data.frame(name=c("A", "B"), age=c(21, 25))
```

- Lists (list(name=“John”, age=30, salary=4000))

```
l <- list(name="John", age=30, salary=4000)
```

Indexing in R

- Indexing vectors (v[1])

```
v[1]
```

```
## [1] 1
```

- Indexing matrices (m[1, 2])

```
m[1, 2]
```

```
## [1] 4
```

Indexing in R

- Indexing data frames (df[1, 2])

```
df$age
```

```
## [1] 21 25
```

- Indexing lists

```
l$name
```

```
## [1] "John"
```

Importing & Manipulating Data

- Loading built-in datasets (data(mtcars))

```
data(mtcars)
```

- Reading CSV files (read.csv("data.csv"))

```
library(readr)
wage_df <- read_csv("wageData.csv")
```

Importing & Manipulating Data

- Viewing data (head(df))

```
head(wage_df)[1:2,]
```

```
## # A tibble: 2 x 12
##    ...1  wage education experience   age ethnicity regio
##   <dbl> <dbl>     <dbl>      <dbl> <dbl> <chr>     <chr>
## 1     1  5.1           8         21    35 hispanic  other
## 2  1100  4.95          9         42    57 cauc      other
## # i 3 more variables: sector <chr>, union <chr>, married
```

Importing & Manipulating Data

- Subsetting data (df[1:10,])

```
subset(wage_df[1:10,])
```

```
## # A tibble: 10 x 12
##      ...1  wage education experience   age ethnicity regi
##     <dbl> <dbl>     <dbl>      <dbl> <dbl> <chr>     <chr
##  1     1  5.1           8         21    35 hispanic  othe
##  2  1100  4.95          9         42    57 cauc      othe
##  3     2  6.67         12          1    19 cauc      othe
##  4     3  4            12          4    22 cauc      othe
##  5     4  7.5          12         17    35 cauc      othe
##  6     5 13.1          13          9    28 cauc      othe
##  7     6  4.45         10         27    43 cauc      sout
##  8     7 19.5          12          9    27 cauc      othe
##  9     8 13.3          16         11    33 cauc      othe
## 10     9  8.75         12          9    27 cauc      othe
## # i 3 more variables: sector <chr>, union <chr>, marriec
```

Importing & Manipulating Data

- Filtering with logical conditions (subset(df, age $>$ 25))

```
subset(wage_df, age > 25)
```

```
## # A tibble: 443 x 12
##      ...1  wage education experience   age ethnicity regi
##     <dbl> <dbl>     <dbl>      <dbl> <dbl> <chr>     <chr
##  1     1  5.1           8         21    35 hispanic  othe
##  2  1100  4.95          9         42    57 cauc      othe
##  3     4  7.5          12         17    35 cauc      othe
##  4     5 13.1          13          9    28 cauc      othe
##  5     6  4.45         10         27    43 cauc      sout
##  6     7 19.5          12          9    27 cauc      othe
##  7     8 13.3          16         11    33 cauc      othe
##  8     9  8.75         12          9    27 cauc      othe
##  9    10 11.4          12         17    35 cauc      othe
## 10    11 11.5          12         19    37 cauc      othe
## # i 433 more rows
## # i 3 more variables: sector <chr>, union <chr>, marriec
```

Basic Data Analysis in R

- Summary statistics

```
summary(mtcars$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.40   15.43   19.20   20.09   22.80   33.90
```

- Calculating mean, median, standard deviation

```
mean(mtcars$mpg)
```

```
## [1] 20.09062
```

```
median(mtcars$mpg)
```

```
## [1] 19.2
```

```
sd(mtcars$mpg)
```

```
## [1] 6.026948
```

Basic Data Analysis in R

- Calculating correlation

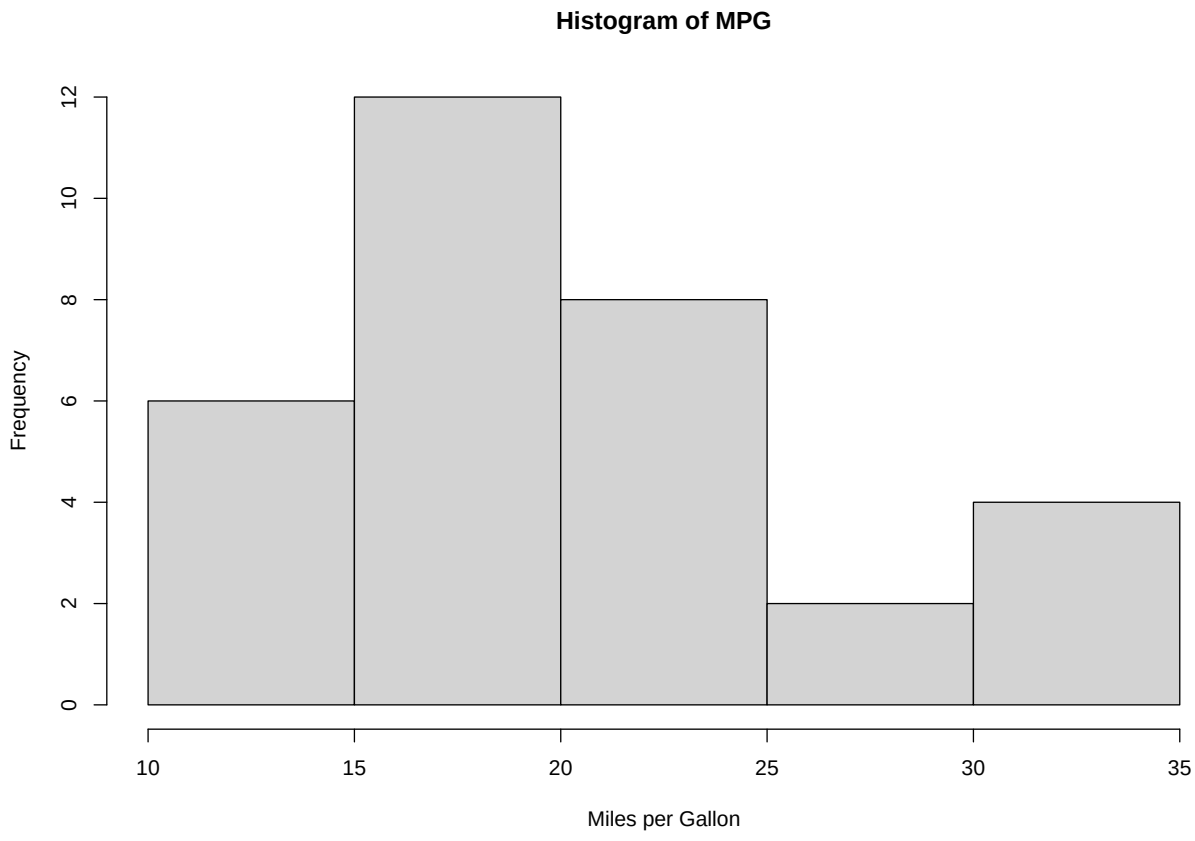
```
cor(mtcars$mpg, mtcars$wt)
```

```
## [1] -0.8676594
```

Basic Data Analysis in R

- Simple plotting
  - Histogram

```
hist(mtcars$mpg,
     main="Histogram of MPG",
     xlab="Miles per Gallon")
```



Basic Data Analysis in R

- Simple plotting
  - Scatterplot

```
plot(mtcars$mpg, mtcars$wt,
     main="Scatterplot of MPG vs Weight",
     xlab="Miles per Gallon", ylab="Weight")
```

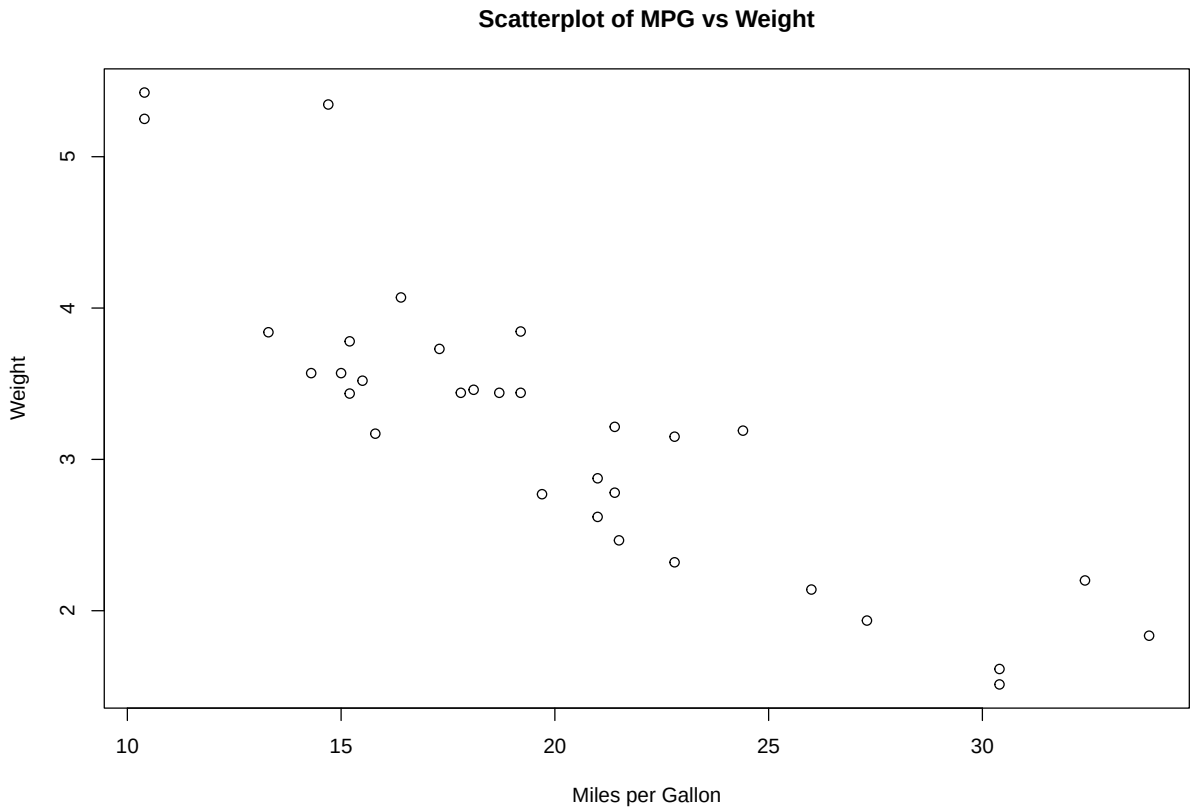
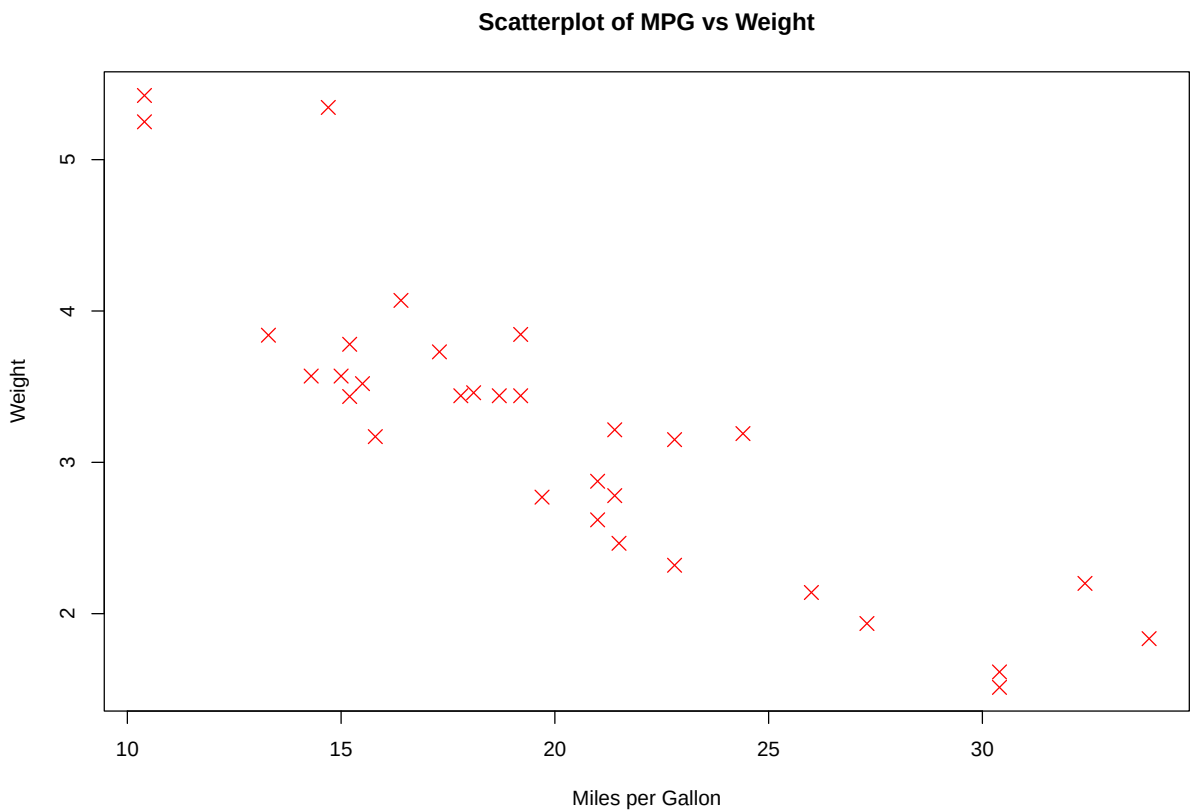


Figure 3: Scatterplot of MPG vs Weight

Customizing Plots

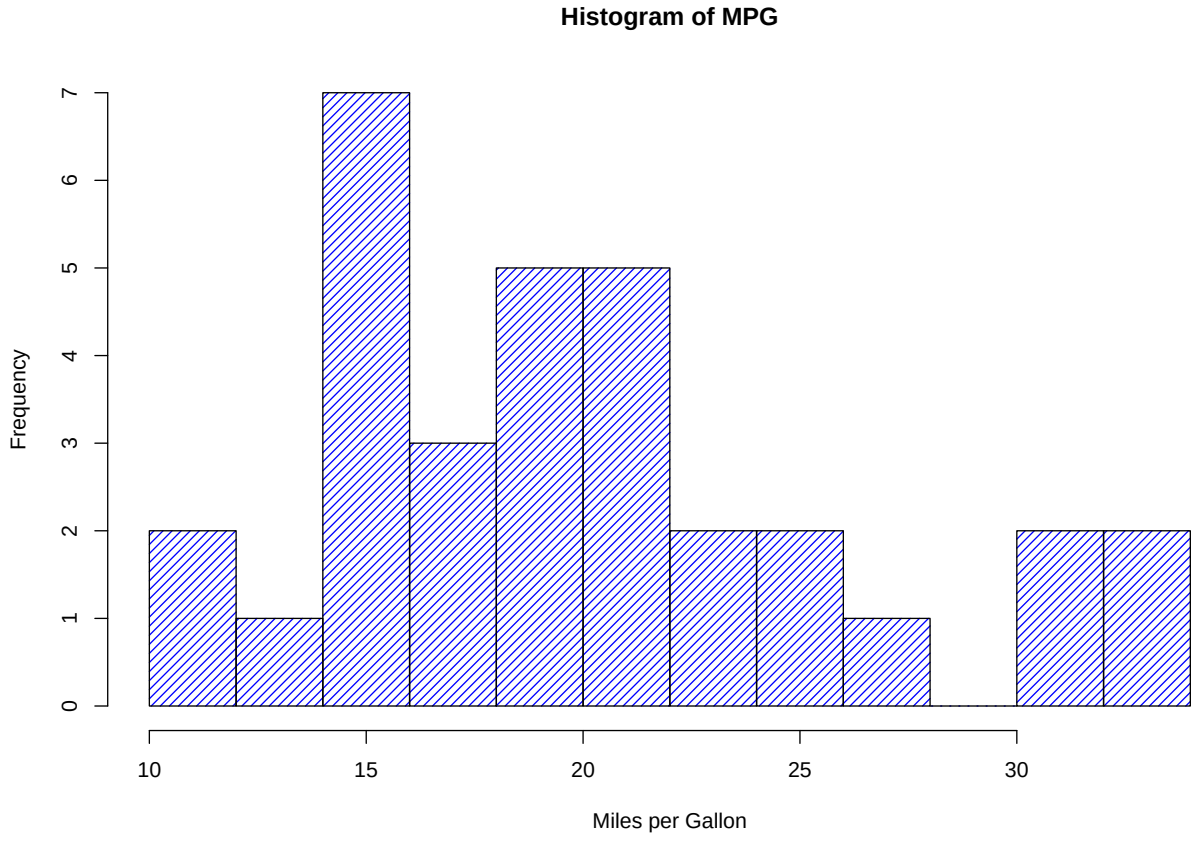
R has a wide variety of options for customizing plots. You can change the color, size, shape, and labels of the plot elements.

```
plot(mtcars$mpg, mtcars$wt,
     main="Scatterplot of MPG vs Weight",
     xlab="Miles per Gallon", ylab="Weight",
     col="red", pch=4, cex=1.5)
```



Customizing Plots

```
hist(mtcars$mpg,
     main="Histogram of MPG",
     xlab="Miles per Gallon",
     density=20, angle=45,
     col="blue", border="black", breaks=10)
```



R vs Excel

- Big data: R can handle large datasets that Excel can't.
- Open source + libraries: R has a large number of libraries for data analysis.
- Automation: R can automate repetitive tasks.
- Flexibility: R can handle complex data manipulation and analysis.
- Reproducibility: R scripts can be shared and reproduced.
- Visualization: R has better visualization capabilities.

Types of R files

- R script (.R)
  - R Shiny
- R Markdown (.Rmd)
  - Presentations
  - Reports
  - Dashboards

R Markdown

R Markdown is a file format that allows you to create dynamic documents with R code embedded in them. You can create presentations, reports, and dashboards using R Markdown.

Code chunks are enclosed in three backticks and the language name (e.g., {r}). The code is executed when the document is rendered.

Text can be formatted using Markdown/LaTeX syntax. The document can be rendered to HTML, PDF, or Word.

Resources

- help function

```
help(mean)
?mean
```

- R documentation https://www.rdocumentation.org/
- RStudio cheat sheets https://www.rstudio.com/resources/cheatsheets/
- R for Data Science https://r4ds.had.co.nz/
- R graphics customizations https://r-charts.com/base-r/#:~:text=CUSTOMIZATION,arguments%20of%20the%20par%20function.

Hands-on Exercise

Activity: - Load a dataset from R (Hint: use data() to find built-in datasets) - Calculate summary statistics of a column in the data (mean, median, standard deviation). - Hint: use ?[name of dataset] to find the columns in the dataset. - Plot a histogram with some customizations. - Plot a scatterplot with some customizations.

Thank you!

I hope you enjoyed the workshop. Please feel free to reach out to me if you have any questions or need help with R programming.

- Email: a4girgis@uwaterloo.ca