Conducting a basic statistical t-test in R using Dplyr and base packages

March 19 & 20, 2025

www.dartgo.org/RRADworkshops

Research Data Services
researchdatahelp@groups.dartmouth.edu
Stephen P. Gaughan
Research Facilitator - Geospatial Programmer/Analyst
603-646-9524
stephen.p.gaughan@dartmouth.edu
Hanover, NH 03755

• Reproducible •
Research
@ DARTMOUTH

DARTMOUTH

Reproducible Research

- Note: slides will be sent after the workshop
- Upcoming workshops:
  - https://dartgo.org/rradworkshops
- Research Data Services
  - Data Analysis & Visualization support: https://www.library.dartmouth.edu/research-create/data-services/analysis-visualization
  - Data Management: https://researchguides.dartmouth.edu/data_management
  - Email us at researchdatahelp@dartmouth.edu

About the Reproducible Research Group

- Joint venture of **Research Computing @ ITC** and **Research Data Services @ Library**
- Consult with us on
  - Research data management
  - Data visualization
  - Biomedical research support
  - Spatial data and GIS
  - High performance and research computing
  - Statistical software and tools
  - Economics and social sciences data
- **Meet** the people on campus that support your reproducible research lifecycle
- **Engage** in community discussions to learn from other researchers on campus
- **Attend** our workshops to **learn** practical tools and tips

https://researchguides.dartmouth.edu/data_management - - researchdatahelp@dartmouth.edu

About Research Data Services

Research Data Management

Data Management Plans for sponsored projects

Finding and using third party data

Collection and cleaning of data

Organization and documentation

Publishing and data repositories

Data Science, Data Analysis, Data Visualization

Textual, numerical, spatial data

Assistance with building reproducible workflows

Scripting in R and Python

Computational Scholarship

Computational project planning

Collections as data

Storytelling with data and visualizations

Text and data mining

Digital Humanities support

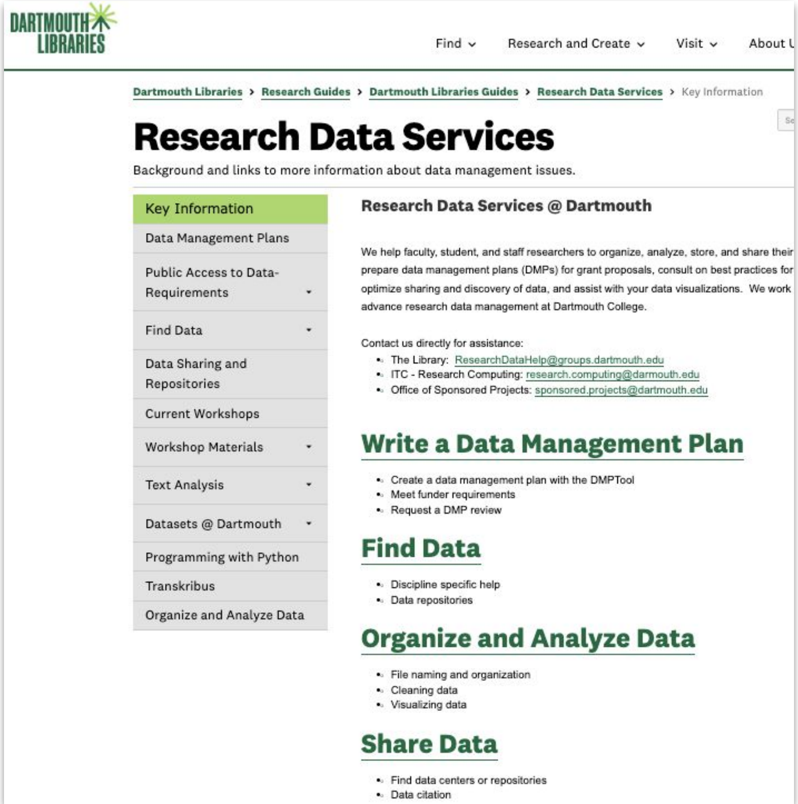
Computational Pedagogy

https://researchguides.dartmouth.edu/data_management - - researchdatahelp@dartmouth.edu

Our Mission & Services

RDS helps facilitate research by consulting on best practices with faculty, student, and staff researchers to organize, analyze, store, and share their research data.

We can help prepare data management plans (DMPs) for grant proposals, consult on best practices for storage and preservation, help optimize sharing and discovery of data, and assist with your data visualizations.



Research Data Services

Background and links to more information about data management issues.

Research Data Services @ Dartmouth

We help faculty, student, and staff researchers to organize, analyze, store, and share their
prepare data management plans (DMPs) for grant proposals, consult on best practices for
optimize sharing and discovery of data, and assist with your data visualizations. We work
advance research data management at Dartmouth College.

Contact us directly for assistance:

- The Library: ResearchDataHelp@groups.dartmouth.edu
- ITC - Research Computing: research.computing@darmouth.edu
- Office of Sponsored Projects: sponsored.projects@dartmouth.edu

Write a Data Management Plan

- Create a data management plan with the DMPTool
- Meet funder requirements
- Request a DMP review

Find Data

- Discipline specific help
- Data repositories

Organize and Analyze Data

- File naming and organization
- Cleaning data
- Visualizing data

Share Data

- Find data centers or repositories
- Data citation

https://researchguides.dartmouth.edu/data_management - - researchdatahelp@dartmouth.edu

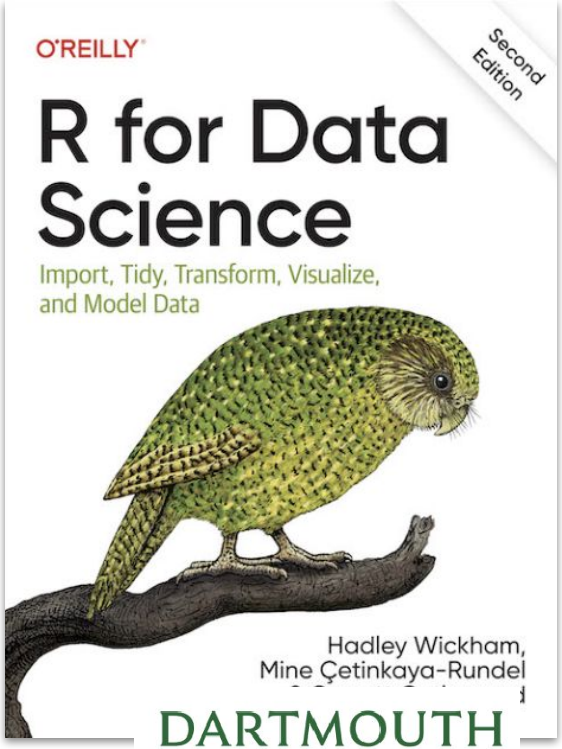
Welcome!

Base R Exploratory Data Analysis Introduction

- Install R and R Studio, an interactive development environment R
- Additional Software Carpentry R lessons: http://swcarpentry.github.io/r-novice-inflammation/
- Info and Datasets: www.dartgo.org/r-intro
- Software Carpentry Foundation: https://software-carpentry.org/
- Videos: www.dartgo.org/intro-r

Comprehensive R Network, Tidy Data, Packages, Tasks

- CRAN Vignettes - links
- Stack Overflow - R https://stackoverflow.com/questions/tagged/r
- **R for Data Science**, Hadley Wickham is one of the authors, R guru and "Tidy" data advocate https://r4ds.hadley.nz/
- R packages https://cran.r-project.org/web/packages/
- R Task views https://cran.r-project.org/
- Reproducible Research https://CRAN.R-project.org/view=ReproducibleResearch



DARTMOUTH

More Learning Resources for statistics and data science using R & RStudio

- Swirlstats interactive tutorial https://swirlstats.com/
- R Task View https://cran.r-project.org/web/views/TeachingStatistics.html
- Rpubs web publishing: https://rpubs.com/
- W3 Schools R Tutorial https://www.w3schools.com/R/
- Lots of video tutorials and techniques on Youtube and Stack Overflow (for example https://www.youtube.com/@RProgramming101

Why R?

R can:

- Assist with Exploratory Data Analysis
- Generate a wide range of **plots and visualizations**
- Import data from **CSV, Excel,** and **databases**
- Analyze datasets
- Generate statistics
- Be used to create **reproducible results** with **reusable code** that can be used on laptops for small datasets and on high-performance computers (HPC) for large datasets and complex analyses

R: a language and environment for statistical computing

R is a free software environment for statistical computing and graphics.

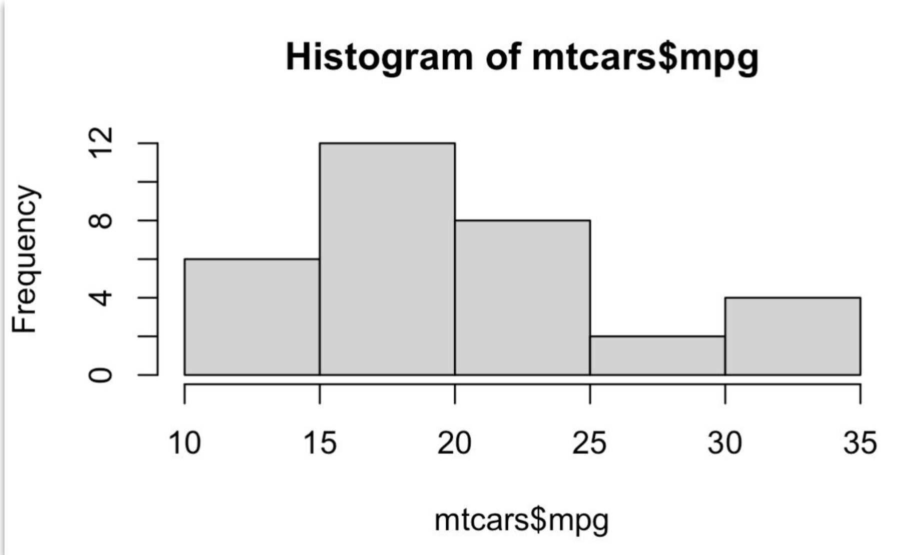
Getting Started with R

- Why R?
- What are basic exploratory statistics and exploratory data analysis (EDA)
- How can I produce statistics from a dataset -mean, max, standard dev?
- How do I run a t-test in R?
- How can I make a simple visualization / plot ?
- How can I store this in a script to automate and make repeatable
- ...and, where can I get help with R statistical functions if I need it?

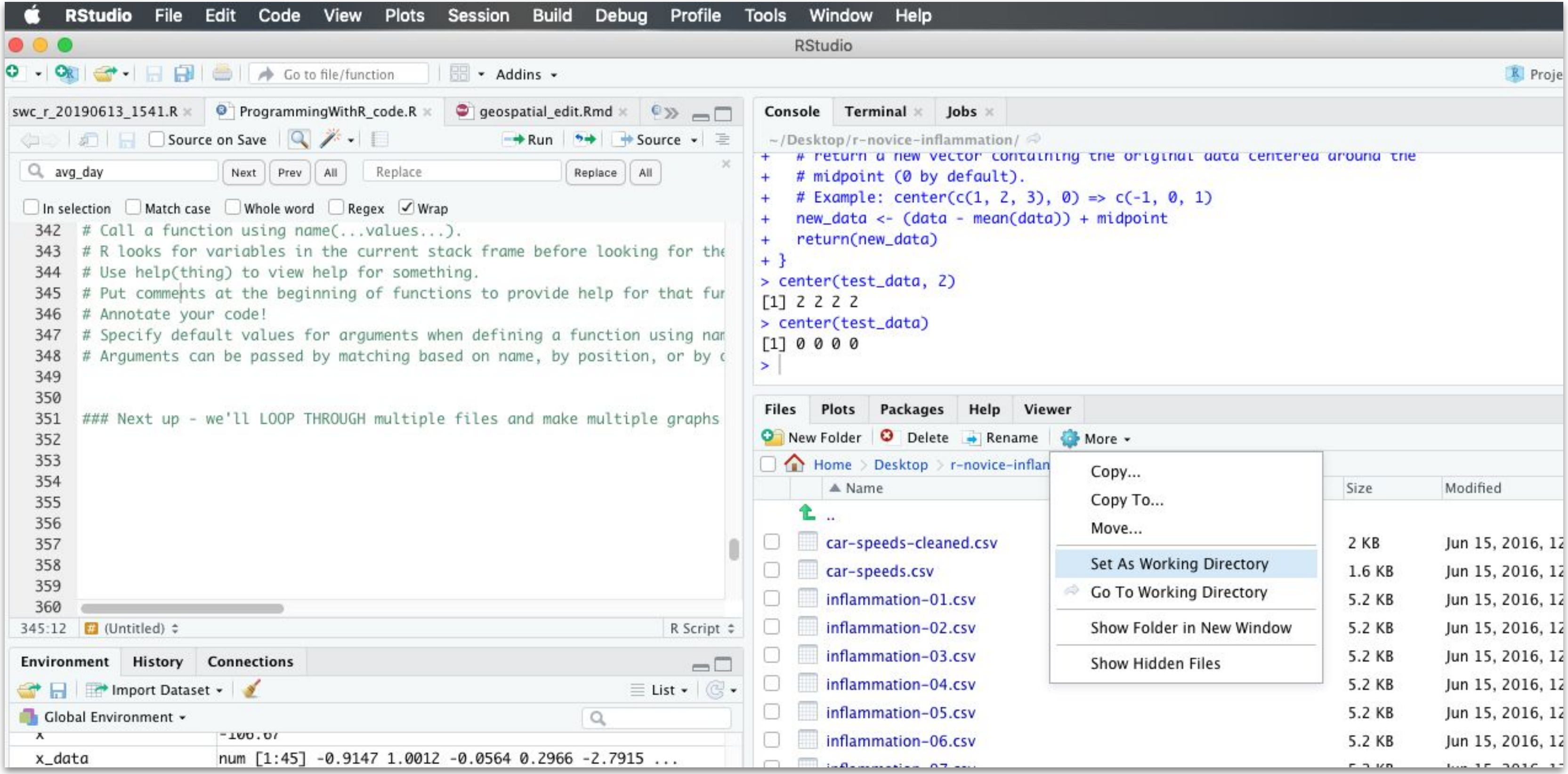
Live code & viz

Live-coding in R's built-in dataset

List of cars with their miles per gallon, number of engine cylinders, engine displacement, vehicle weight, horsepower



Set working directory



Find: avg_day

□ In selection □ Match case □ Whole word □ Regex ☑ Wrap

```
# Call a function using name(...values...).
# R looks for variables in the current stack frame before looking for the
# Use help(thing) to view help for something.
# Put comments at the beginning of functions to provide help for that fur
# Annotate your code!
# Specify default values for arguments when defining a function using nar
# Arguments can be passed by matching based on name, by position, or by c



### Next up - we'll LOOP THROUGH multiple files and make multiple graphs
```

```
~/Desktop/r-novice-inflammation/
+   # return a new vector containing the original data centered around the
+   # midpoint (0 by default).
+   # Example: center(c(1, 2, 3), 0) => c(-1, 0, 1)
+   new_data <- (data - mean(data)) + midpoint
+   return(new_data)
+ }
> center(test_data, 2)
[1] 2 2 2 2
> center(test_data)
[1] 0 0 0 0
> 
```

Home > Desktop > r-novice-inflan

Name	Size	Modified
..		
car-speeds-cleaned.csv	2 KB	Jun 15, 2016, 12
car-speeds.csv	1.6 KB	Jun 15, 2016, 12
inflammation-01.csv	5.2 KB	Jun 15, 2016, 12
inflammation-02.csv	5.2 KB	Jun 15, 2016, 12
inflammation-03.csv	5.2 KB	Jun 15, 2016, 12
inflammation-04.csv	5.2 KB	Jun 15, 2016, 12
inflammation-05.csv	5.2 KB	Jun 15, 2016, 12
inflammation-06.csv	5.2 KB	Jun 15, 2016, 12

```
x        -106.67
x_data   num [1:45] -0.9147 1.0012 -0.0564 0.2966 -2.7915 ...
```

RStudio interface tools we'll use

- Console, Terminal
- Scripts
- Environment, History
- Files, plots, packages, help - File browser GoTo working directory, set working directory
- Set As Working Directory
- Cheatsheets
- Menus - Help (cheatsheets), Code (Comment/Uncomment), Plots (Save as)

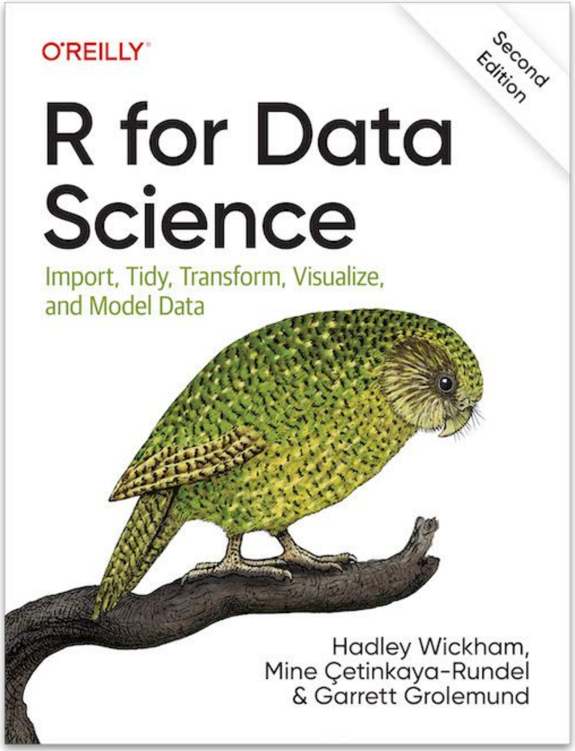
'Tidy' Data - a concept for all data, borrowed from R

Based on R, but applies to ALL data science
https://r4ds.had.co.nz/tidy-data.html
Source: R for Data Science

Tidy Data refers to a structure for arranging data where:

- Each variable has its own column, making it easier to compare across observations.
- Each observation (or case) has its own row, allowing for easy handling and aggregation of the data.
- There is one table (not multiple tables) per subject area or concept.
- There are no hidden values, and the number of rows is equal to the total count of observational units.

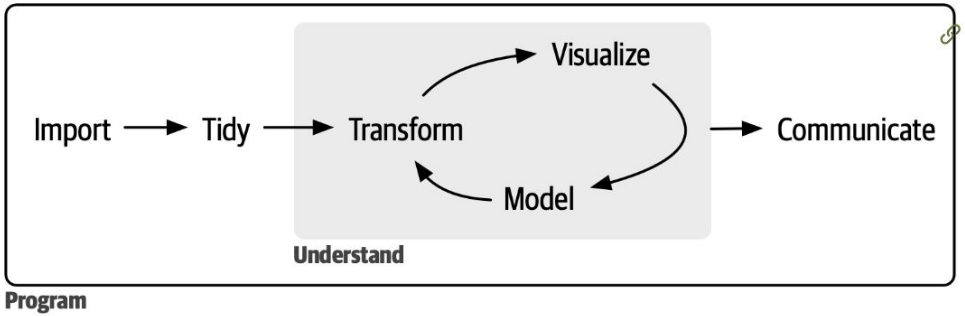
Following these principles allows for much easier data analysis, and it promotes an intuitive understanding and exploration of the data.
Note: it is ok to have multiple tables and relational database tables, but for data analysis and visualization, it is beneficial to focus on extracting data from those entities into a tidy-format data structure as described above.


O'REILLY
Second Edition
R for Data Science
Import, Tidy, Transform, Visualize, and Model Data
Hadley Wickham, Mine Çetinkaya-Rundel & Garrett Grolemund

Source: R for Data Science, Wickham et al

A typical data science process:

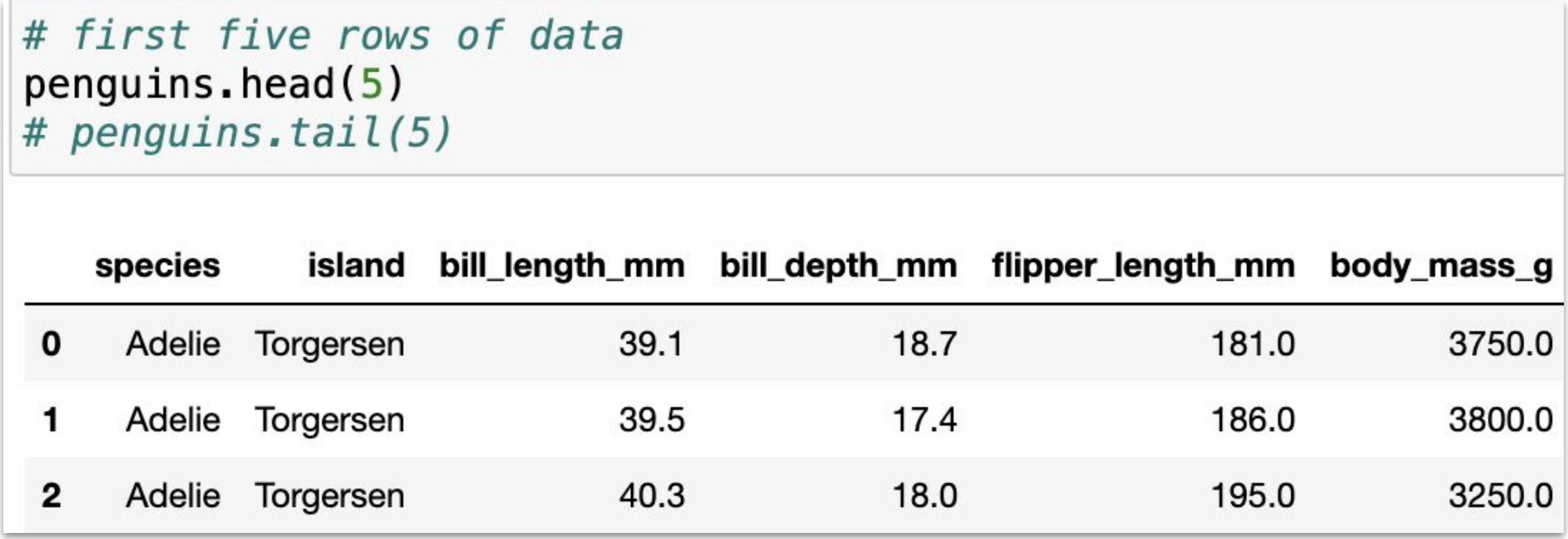
- Make a copy of raw dataset
- With the copy, import & tidy or tidy & import
- Transform/model/analyze/visualize
- Review results with colleagues, iterate process for new methods & discovery
- Communicate the results to general audience in a visual, human-readable and understandable fashion



Source: R for Data Science, Wickham et al

Tidy data example, observations

- Observations per penguin
- Rows are individuals, columns are attributes/measurements, species, location where penguin was measured, length of the bill length, flipper length, body mass

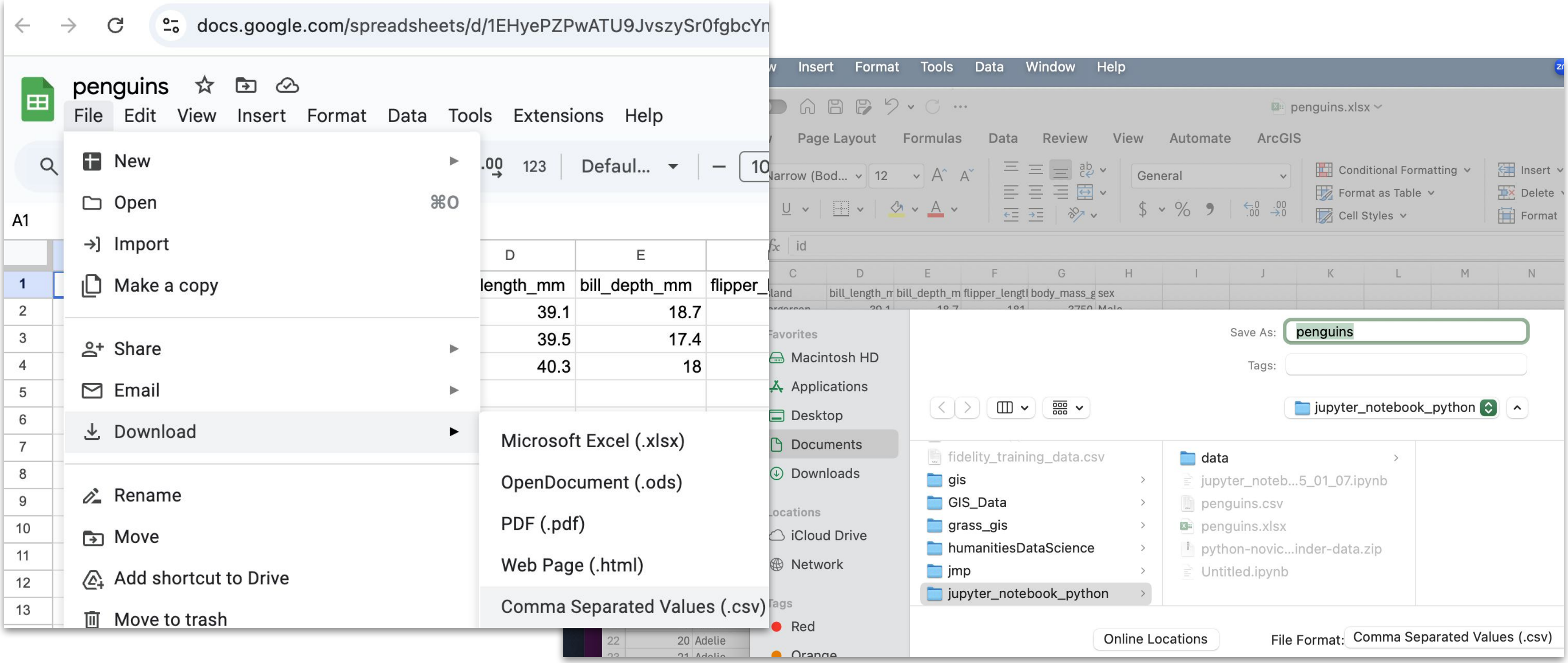


```
# first five rows of data
penguins.head(5)
# penguins.tail(5)
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0

Tidy data - Excel / Google Sheets to CSV

- File > Download or File > Save As



docs.google.com/spreadsheets/d/1EHyePZPwATU9JvszySr0fgbcYn

Save As: penguins

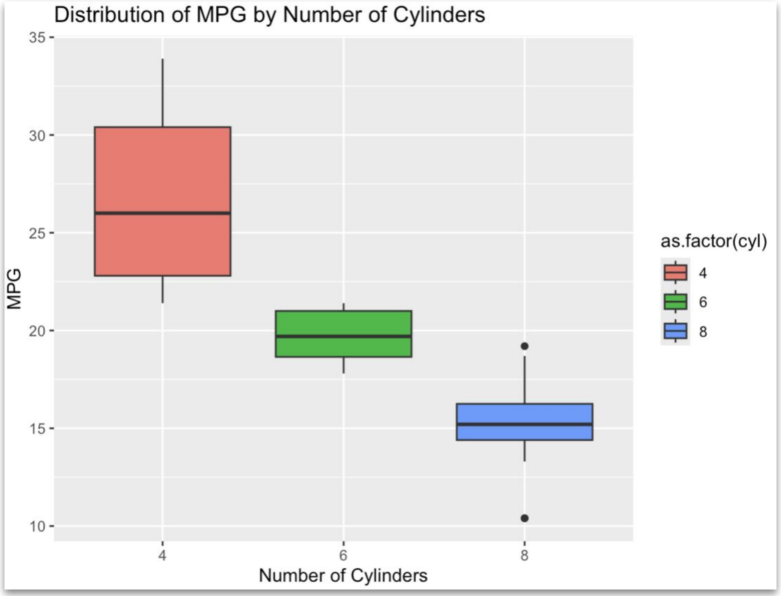
Tags:

jupyter_notebook_python

File Format: Comma Separated Values (.csv)

Get to know the data through Exploratory Data Analysis (EDA)

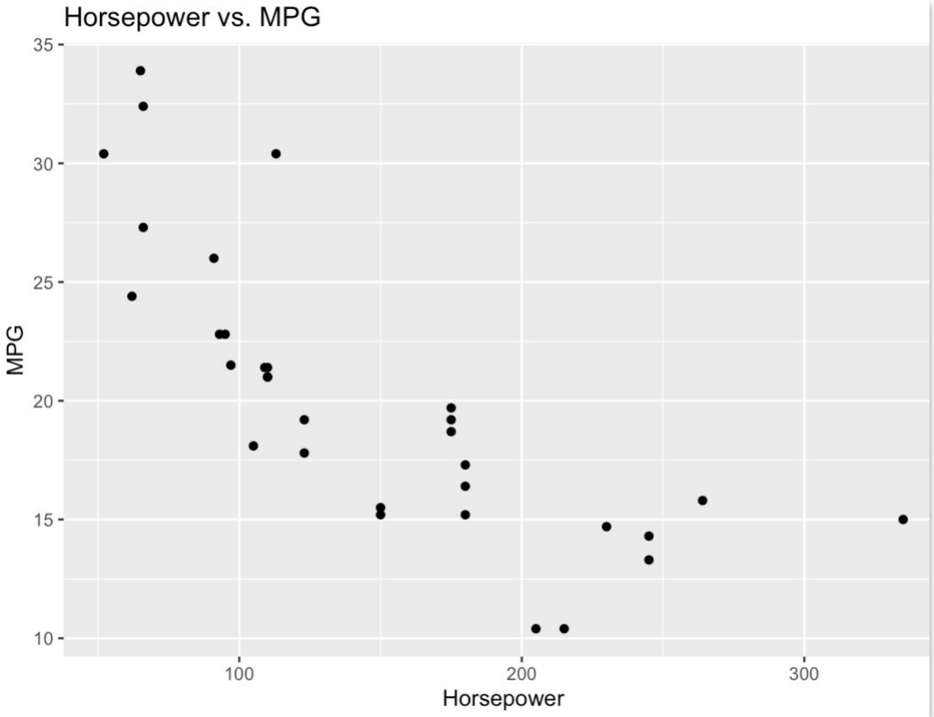
```
# View the first few rows of the mtcars dataset
head(mtcars)

# Summary statistics
summary(mtcars)

# Check for missing values
sum(is.na(mtcars))
```



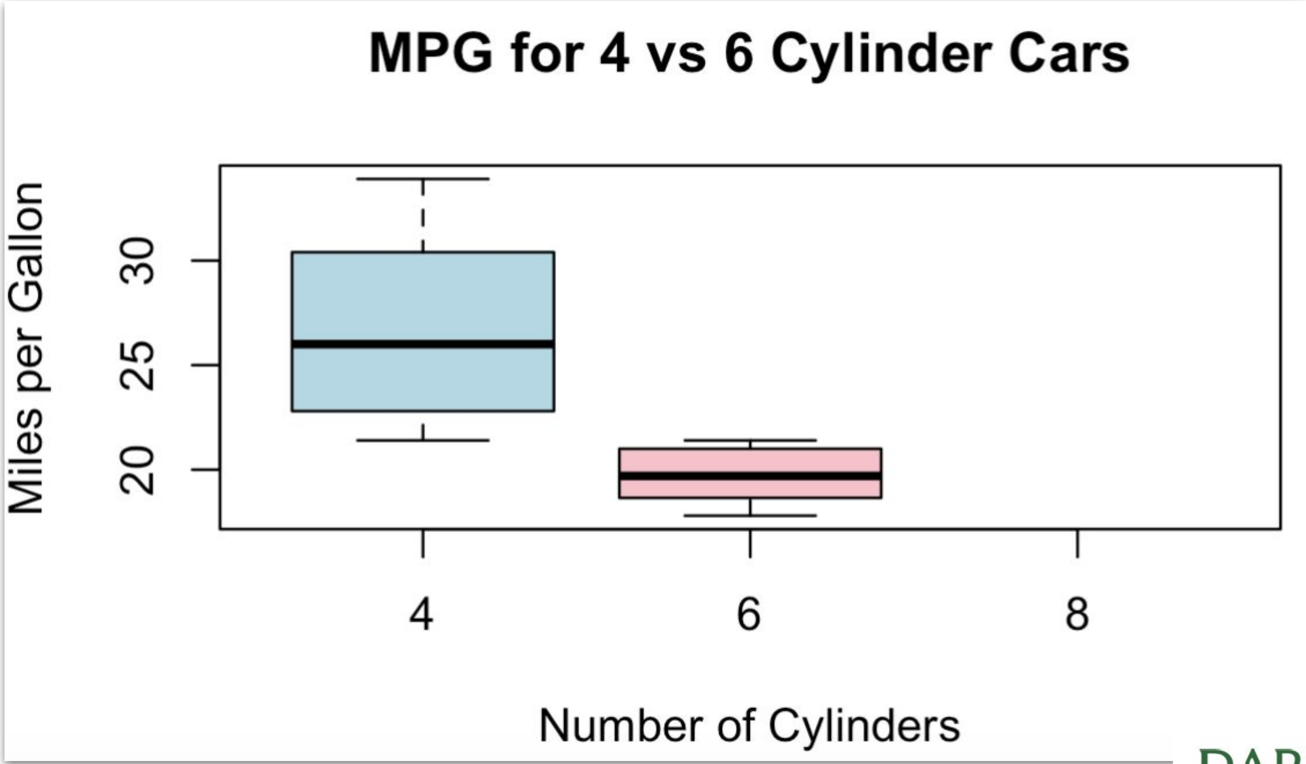
```
# Visualize the distribution of mpg (miles per gallon) by the number of cylinders
ggplot(mtcars, aes(x = as.factor(cyl), y = mpg, fill = as.factor(cyl))) +
  geom_boxplot() +
  labs(title = "Distribution of MPG by Number of Cylinders",
       x = "Number of Cylinders", y = "MPG")
```

Get to know the data
Exploratory Data Analysis

```
# Scatter plot between horsepower (hp) and mpg

ggplot(mtcars, aes(x = hp, y = mpg)) +

  geom_point() +

  labs(title = "Horsepower vs. MPG", x = "Horsepower", y = "MPG")
```



Boxplot comparison of mpg and number of cylinders



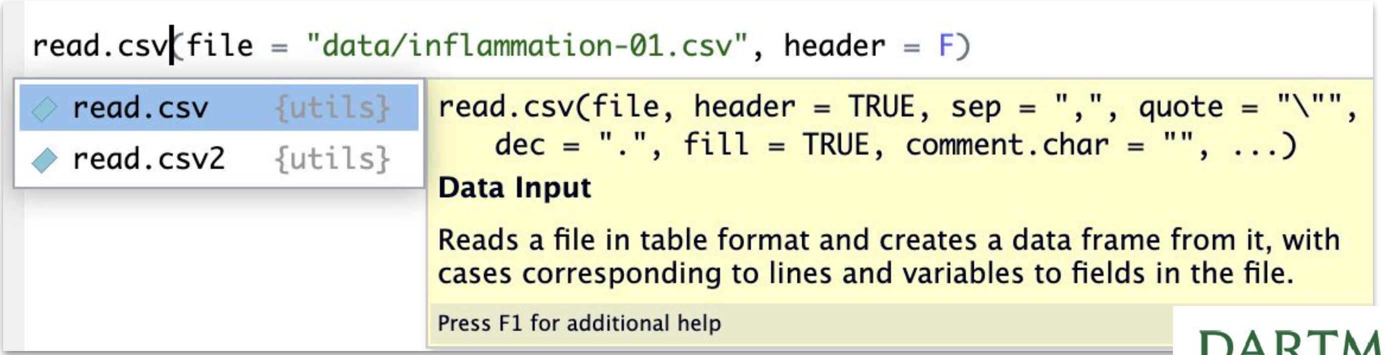
Checking the R version, loading a dataset

# The hashtag symbol indicates a comment. ALWAYS a good idea to comment your code and scripts. R.Version() will display the version, and is a good way to test R

R.Version()

# loading a dataset into R Studio (and R) dataset can be found at dartgo.org/r-intro

read.csv(file = "data/inflammation-01.csv", header = FALSE)



```
read.csv(file = "data/inflammation-01.csv", header = F)
```

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",
    dec = ".", fill = TRUE, comment.char = "", ...)
```

Data Input

Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.

Press F1 for additional help

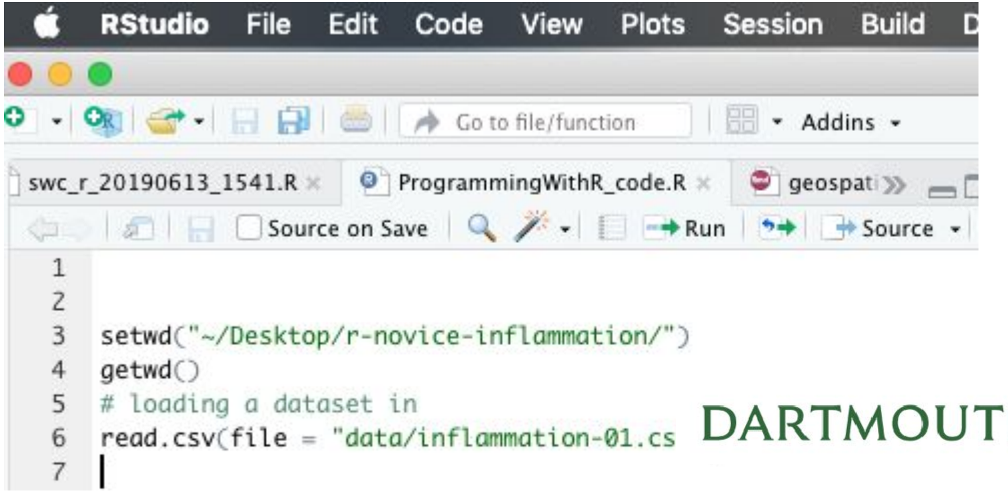
Saving commands in an R script file

File > New File > R Script

Copy the code (use the up arrow to recall the code in the console)

To run a line of code from a script, click Cmd + Return or Ctrl + Enter

Cmd + S to save the file



```
setwd("~/Desktop/r-novice-inflammation/")
getwd()
# loading a dataset in
read.csv(file = "data/inflammation-01.cs
```

DARTMOUTH

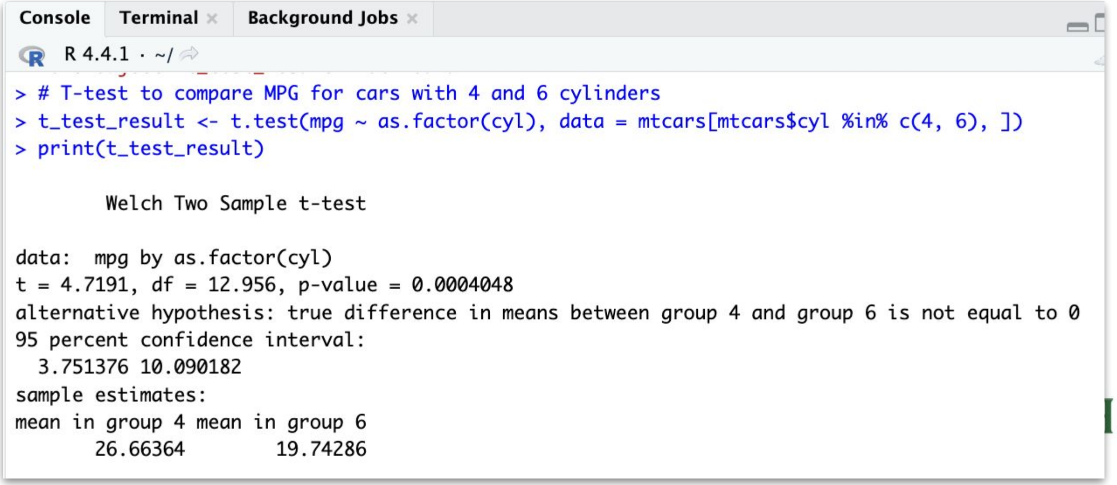
Live coding

- Setting the working environment
- Reading in a dataset
- Assigning variables
- Viewing properties of datasets
- Viewing and extracting values from datasets
- Subsetting data
- Basic Statistics

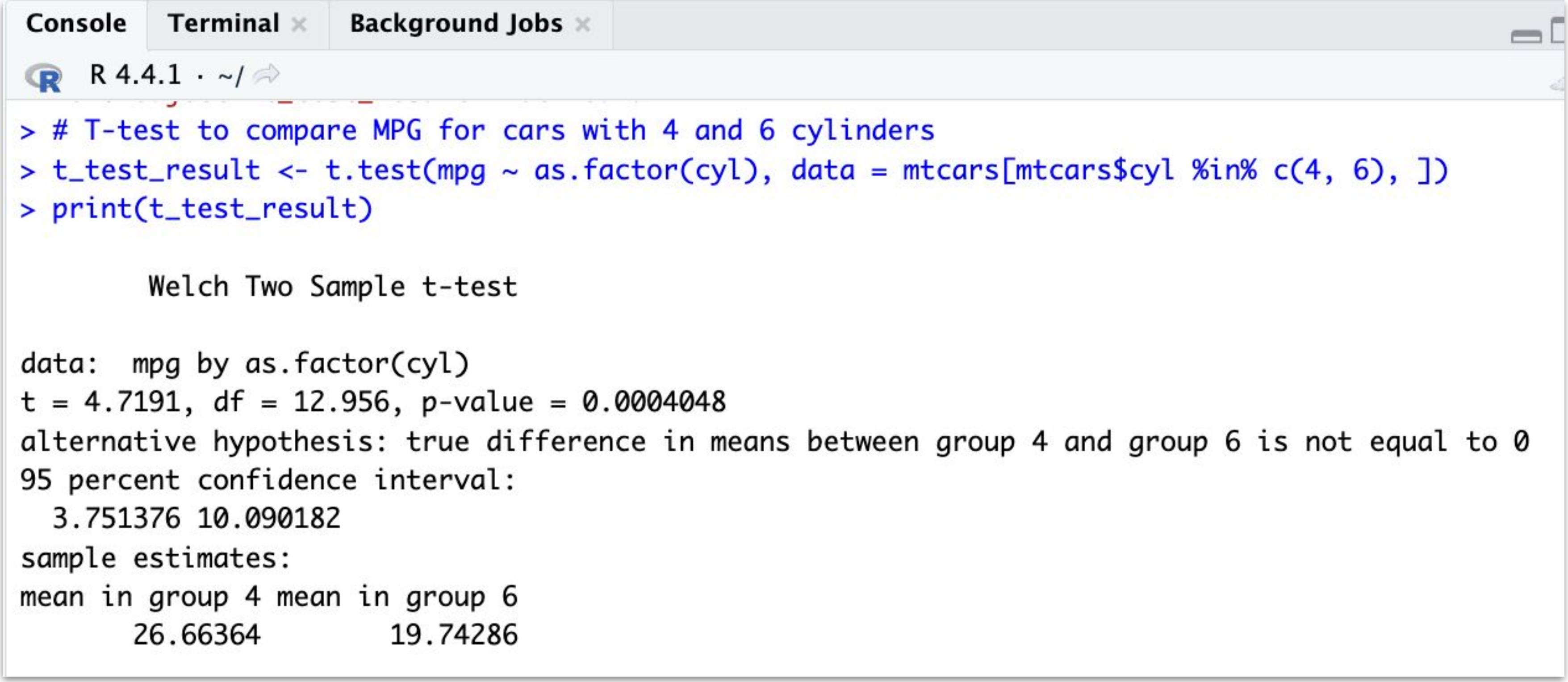
Running a t-test in R

T-test to compare MPG for cars with 4 and 6 cylinders, hypothesis = Is there a significant difference in miles-per-gallon for cars with 4 cylinders and cars with 6 cylinders?

```
t_test_result <- t.test(mpg ~ as.factor(cyl), data = mtcars[mtcars$cyl %in% c(4, 6), ])

print(t_test_result)
```



```
> # T-test to compare MPG for cars with 4 and 6 cylinders
> t_test_result <- t.test(mpg ~ as.factor(cyl), data = mtcars[mtcars$cyl %in% c(4, 6), ])
> print(t_test_result)

        Welch Two Sample t-test

data:  mpg by as.factor(cyl)
t = 4.7191, df = 12.956, p-value = 0.0004048
alternative hypothesis: true difference in means between group 4 and group 6 is not equal to 0
95 percent confidence interval:
  3.751376 10.090182
sample estimates:
mean in group 4 mean in group 6 
       26.66364        19.74286 
```

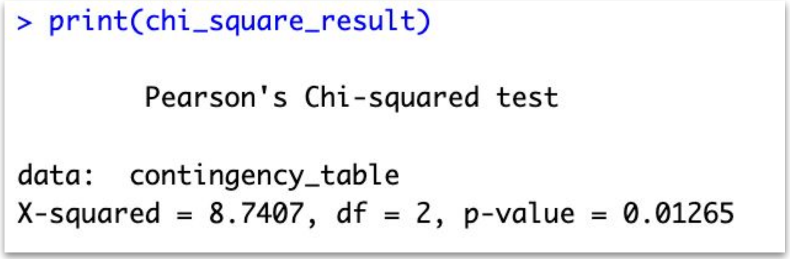


```
> # T-test to compare MPG for cars with 4 and 6 cylinders
> t_test_result <- t.test(mpg ~ as.factor(cyl), data = mtcars[mtcars$cyl %in% c(4, 6), ])
> print(t_test_result)

	Welch Two Sample t-test

data:  mpg by as.factor(cyl)
t = 4.7191, df = 12.956, p-value = 0.0004048
alternative hypothesis: true difference in means between group 4 and group 6 is not equal to 0
95 percent confidence interval:
  3.751376 10.090182
sample estimates:
mean in group 4 mean in group 6 
       26.66364        19.74286 
```

Test indicates the 'alternative hypothesis' is true, so it is likely that the mean mpg of 4cyl is different than the mean mpg of 6 cyl

Chi Square test in R

```
# Is there a significant relationship between the number of cylinders and transmission type?
# Convert cyl (cylinders) and am (transmission: 0 = automatic, 1 = manual) to factors
mtcars$cyl <- as.factor(mtcars$cyl)
mtcars$am <- as.factor(mtcars$am)


# Create a contingency table
contingency_table <- table(mtcars$cyl, mtcars$am)
print(contingency_table)

# Perform chi-square test
chi_square_result <- chisq.test(contingency_table)
print(chi_square_result)
```



```
> print(chi_square_result)

	Pearson's Chi-squared test

data:  contingency_table
X-squared = 8.7407, df = 2, p-value = 0.01265
```

Chi Square test

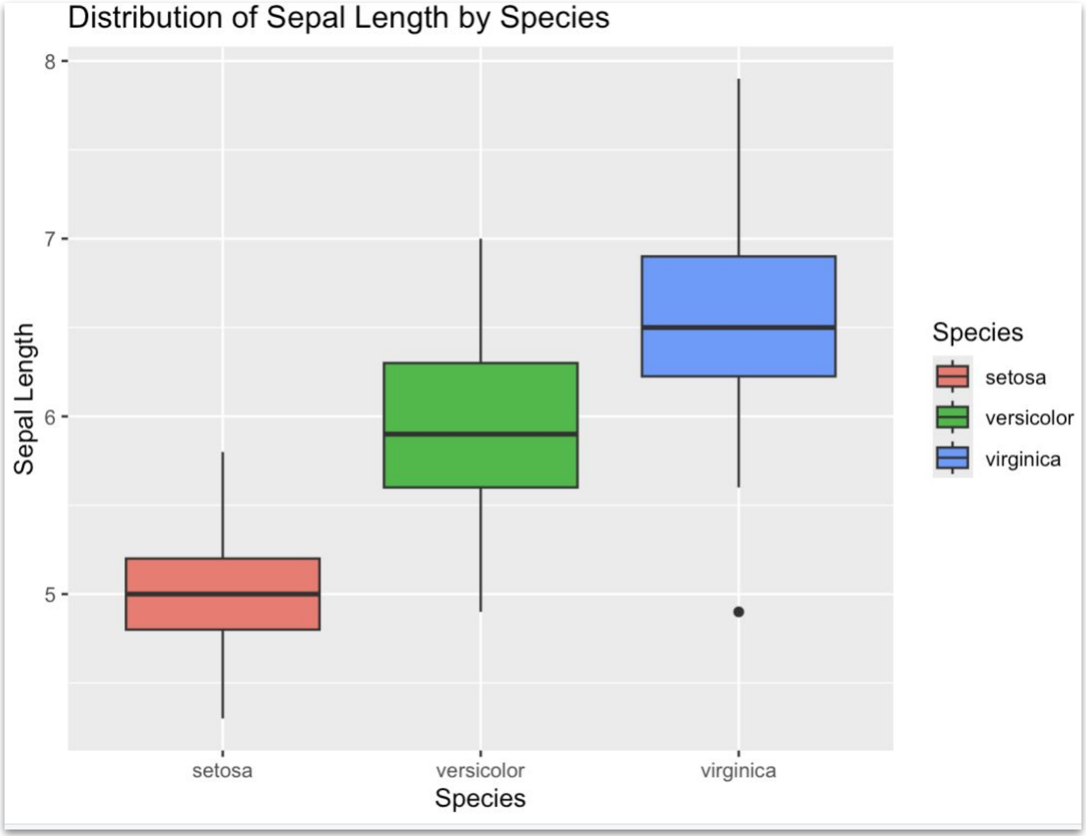
Chi-Square Test: Tests whether there's a significant relationship between the number of cylinders and transmission type.

Is the p-value of the chi-square test big enough to reject the null hypothesis?

In this case, where the p-value=0.01, the value is not big enough to reject the null hypothesis, so it appears there may be a relationship between the number of cylinders and the transmission type.

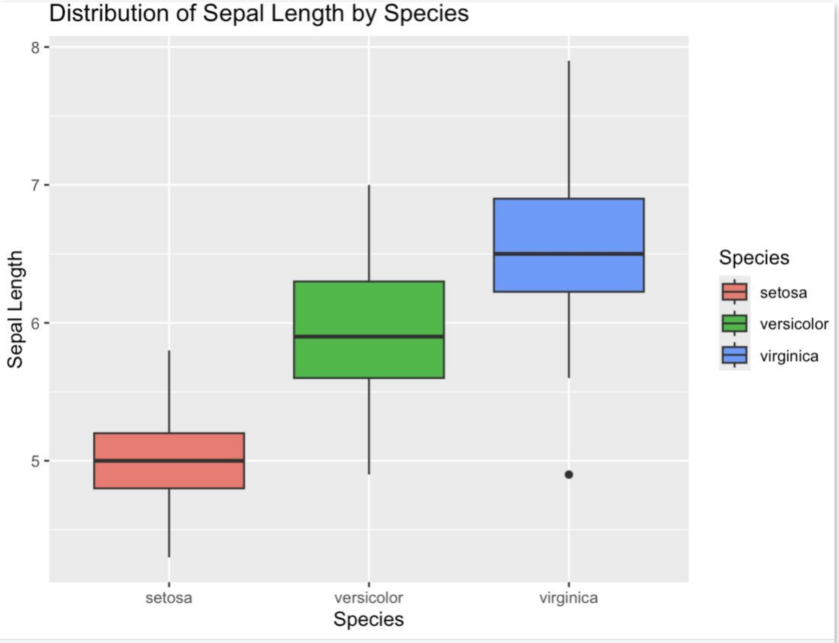
EDA visualization

- Distribution of Iris flower species with regards to the sepal length, a leaf-like portion of the flower that contains the flower bud



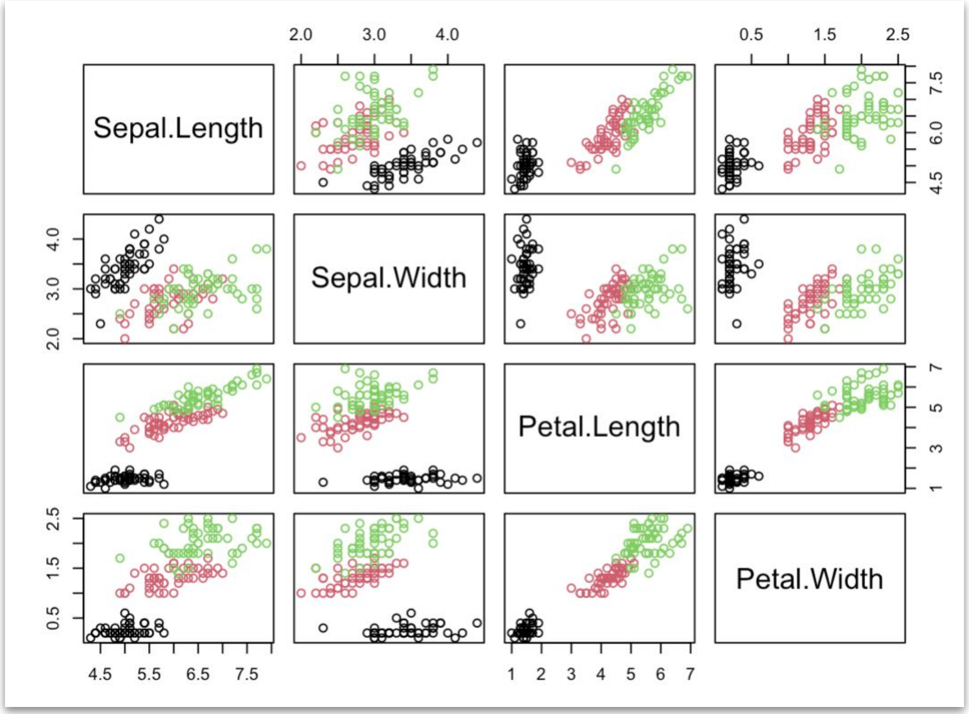
Built-in dataset, three plant species of Iris flowers

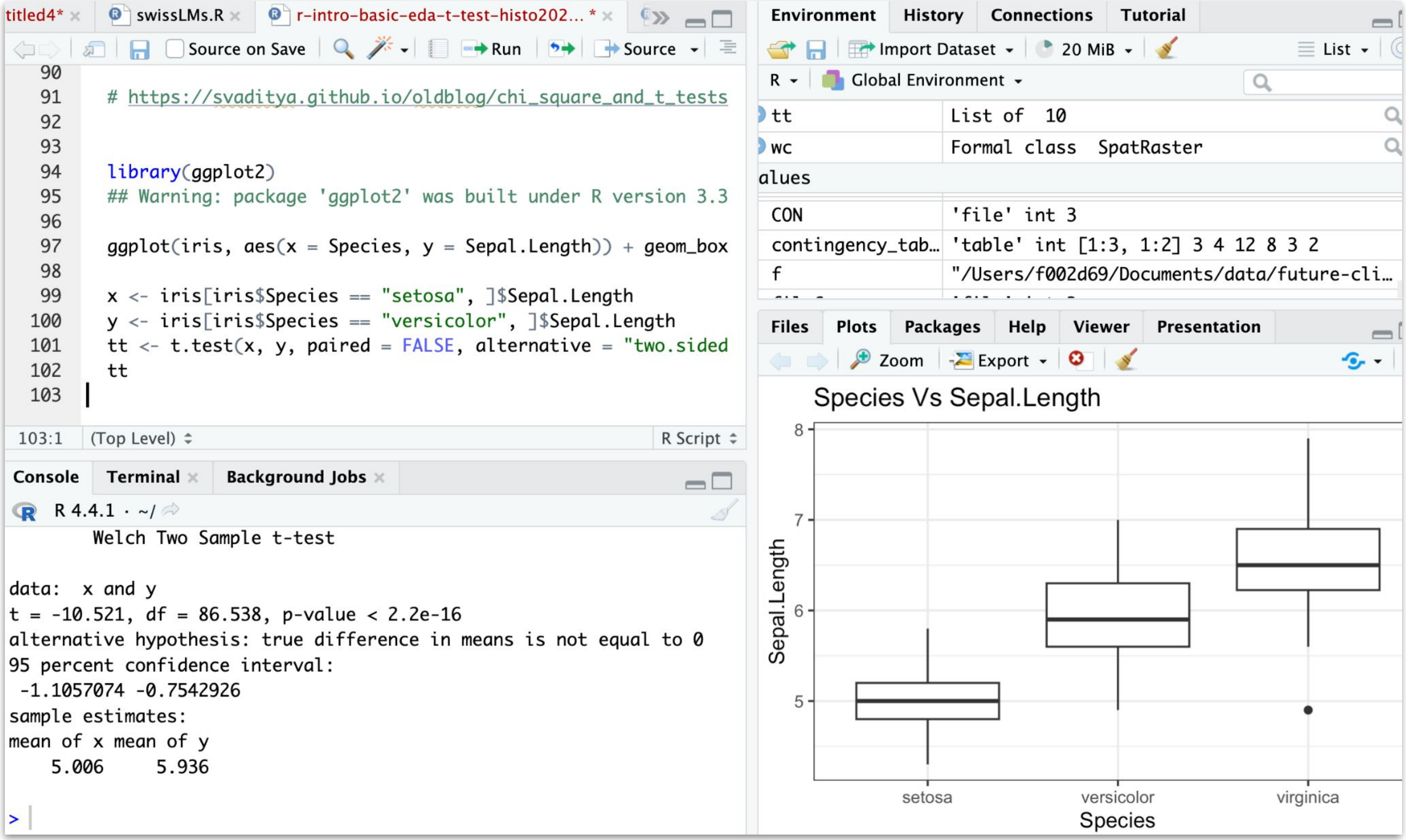
```
# Load necessary libraries
library(ggplot2)
library(dplyr)

# View the first few rows of the iris dataset
head(iris)

# Summary statistics
summary(iris)

# Check for missing values
sum(is.na(iris))

# Visualize the distribution of Sepal.Length by Species
ggplot(iris, aes(x = Species, y = Sepal.Length, fill = Species))
  geom_boxplot() +
  labs(title = "Distribution of Sepal Length by Species",
       x = "Species", y = "Sepal Length")
```



Exploratory Data Analysis with Pair Plots

```
# Pair plot to visualize relationships between variables
pairs(iris[,1:4], col = iris$Species)
```





```r
# https://svaditya.github.io/oldblog/chi_square_and_t_tests


library(ggplot2)
## Warning: package 'ggplot2' was built under R version 3.3

ggplot(iris, aes(x = Species, y = Sepal.Length)) + geom_box

x <- iris[iris$Species == "setosa", ]$Sepal.Length
y <- iris[iris$Species == "versicolor", ]$Sepal.Length
tt <- t.test(x, y, paired = FALSE, alternative = "two.sided
tt
```

```
        Welch Two Sample t-test

data:  x and y
t = -10.521, df = 86.538, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.1057074 -0.7542926
sample estimates:
mean of x mean of y
    5.006     5.936

>
```

tt	List of 10
wc	Formal class SpatRaster
alues	
CON	'file' int 3
contingency_tab…	'table' int [1:3, 1:2] 3 4 12 8 3 2
f	"/Users/f002d69/Documents/data/future-cli…

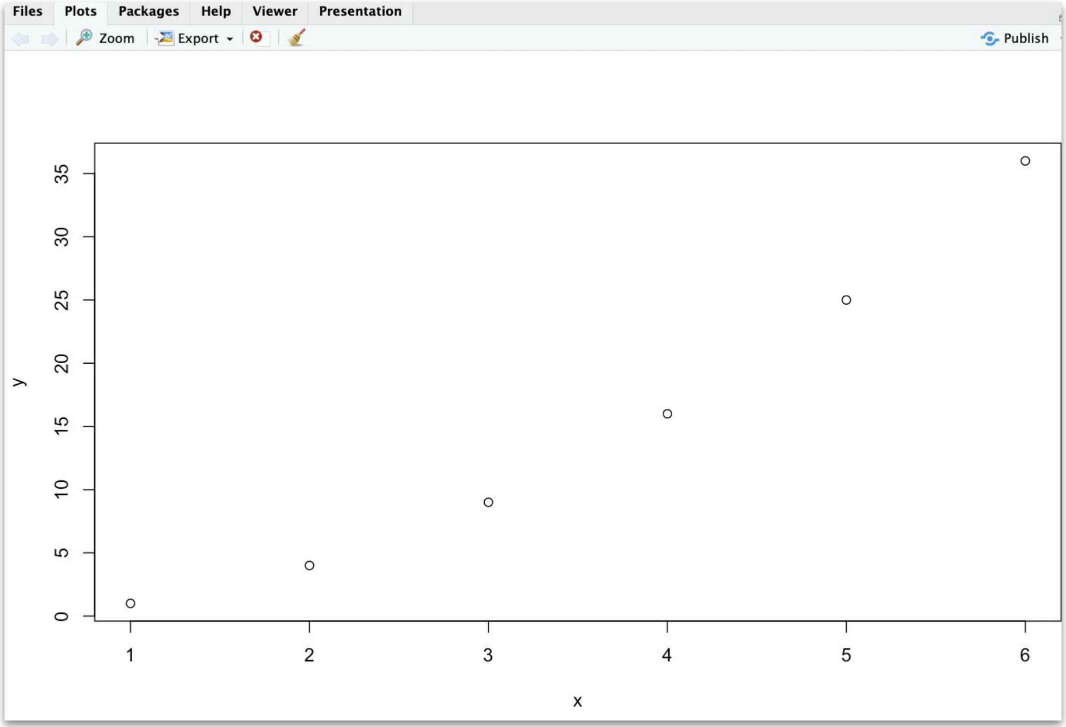
# create some basic lists (vectors)

x=c(1,2,3,4,5,6)

y=c(1,4,9,16,25,36)

# use R's base package 'graphics' to make a basic plot

plot(x,y)



Review of basic plotting in R

plot()

hist()

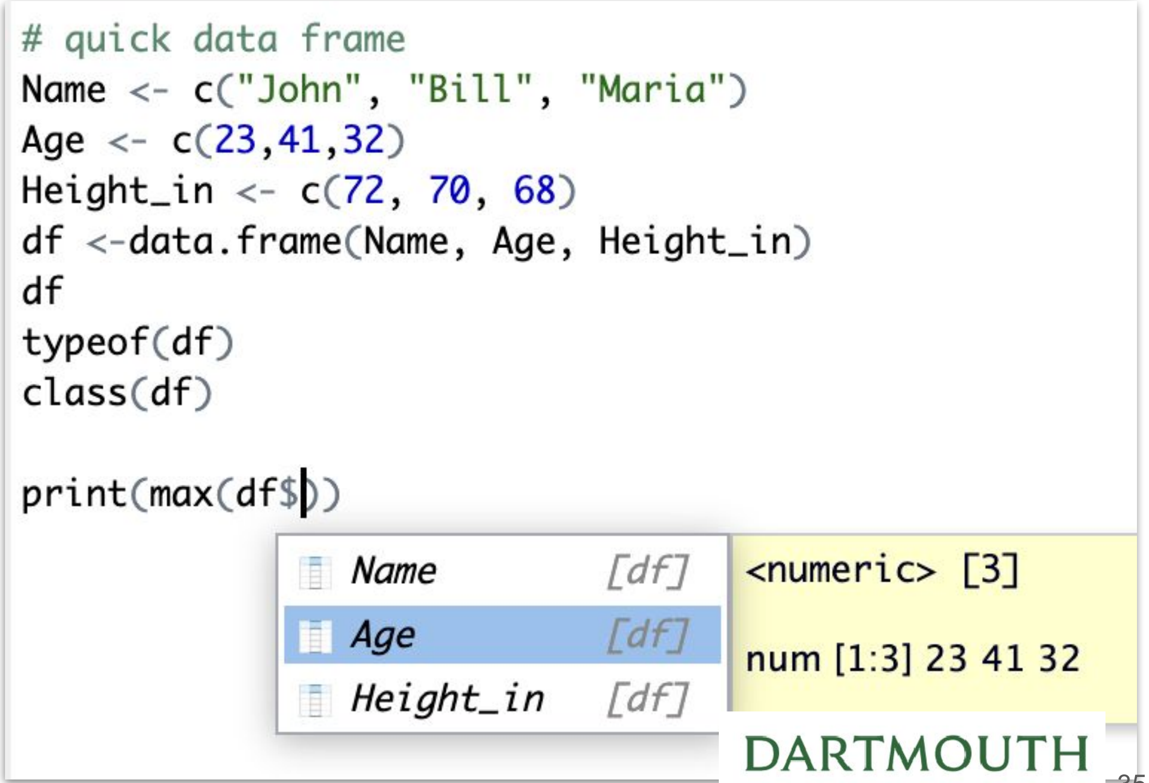
boxplot()

Simple dataframe, $ notation

Dataframes and $ notation

```
# quick data frame
Name <- c("John", "Bill", "Maria")
Age <- c(23,41,32)
Height_in <- c(72, 70, 68)
df <-data.frame(Name, Age, Height_in)
df
typeof(df)
class(df)
print(max(df$Age))
# transpose:
t(df)
```

Note: this 'tidy' data, where rows contain observations(of individuals) and columns contain values (text, number, number)



```
# quick data frame
Name <- c("John", "Bill", "Maria")
Age <- c(23,41,32)
Height_in <- c(72, 70, 68)
df <-data.frame(Name, Age, Height_in)
df
typeof(df)
class(df)

print(max(df$))
```

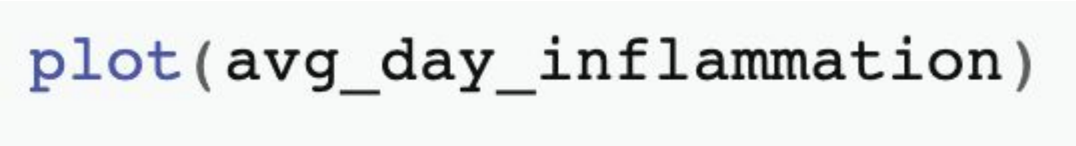
```
<numeric> [3]

num [1:3] 23 41 32
```

Loading a dataset into a variable

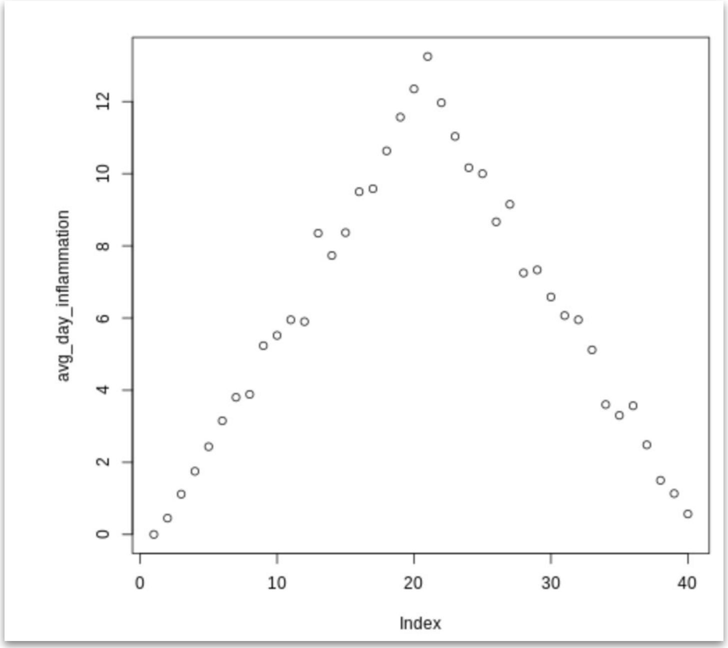
```r
#  load a dataset from a file into a variable in R Studio

dat <- read.csv(file = "data/inflammation-01.csv", header = FALSE)

#  view the first few lines of data using the head and tail functions from R's base package 'utils'

head(dat)

tail(dat)
```

EDA with data from an external csv

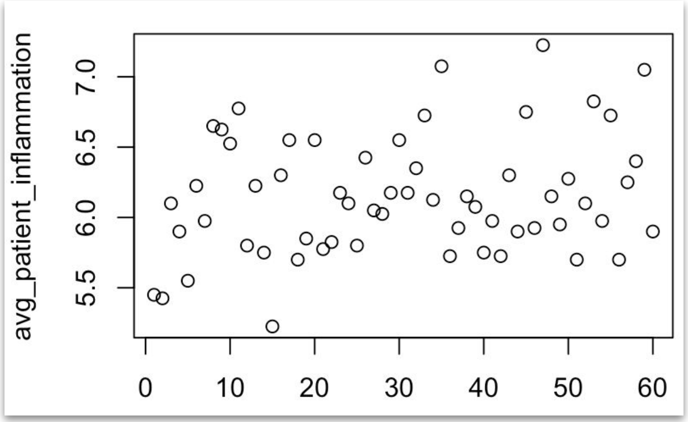
```r
# read in file
dat <- read.csv(file = "data/inflammation-01.csv",
header = FALSE)
# look at first few rows
head(dat)
# compute the mean over all 60 patients per day
avg_day_inflammation <- apply(dat, 2, mean)
# plot these values
plot(avg_day_inflammation)
```

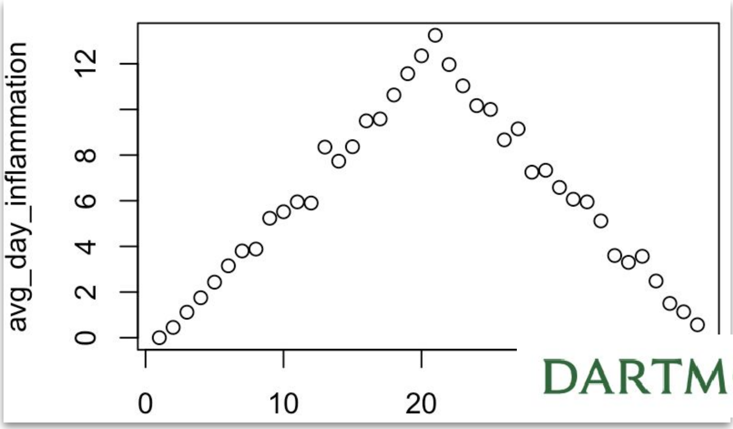


```r
plot(avg_day_inflammation)
```



```
# get the average patient data – the mean of each ROW (1) using apply
avg_patient_inflammation <- apply(dat,1,mean)
avg_patient_inflammation
plot(avg_patient_inflammation)

# next, get the average daily inflammation – mean of each COLUMN (2)
(avg_day_inflammation <- apply(dat,2, mean))
```



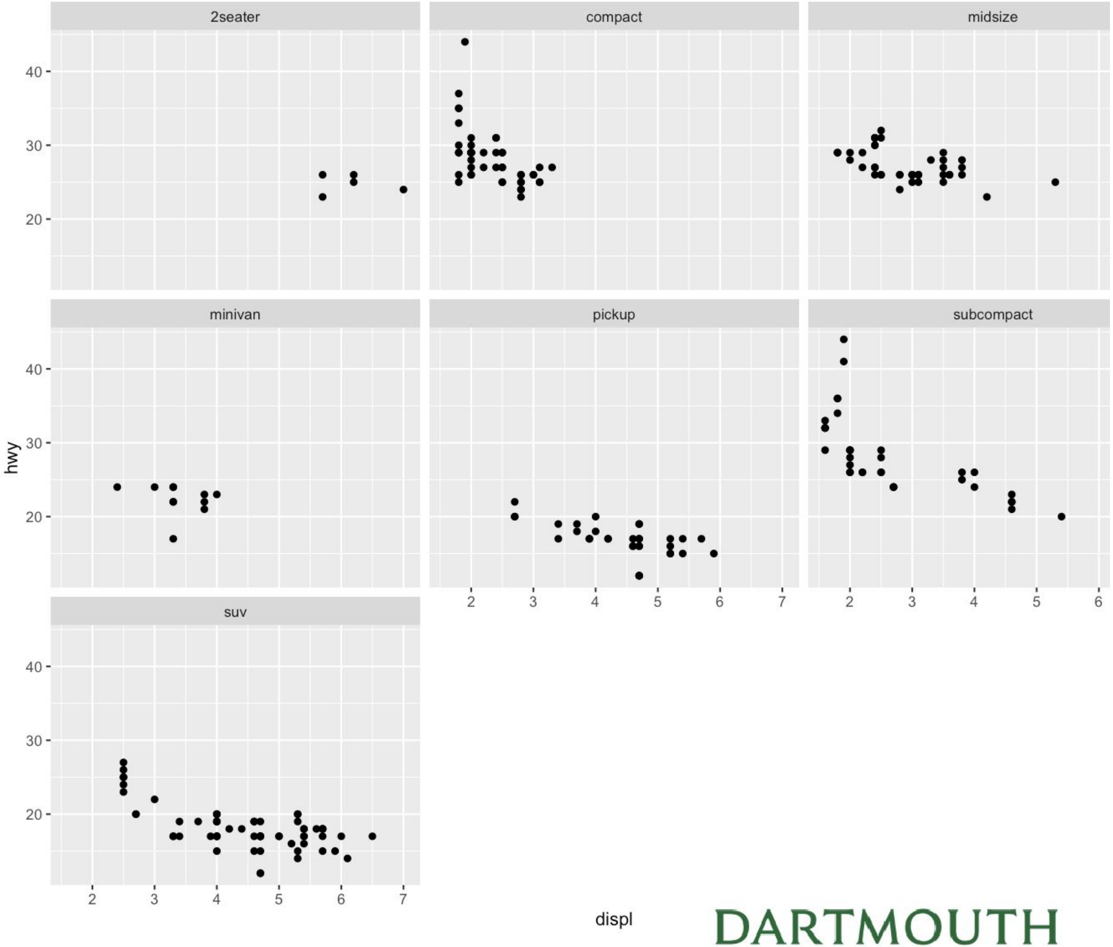


EDA - facetting data

Facets - the many sides or faces of data

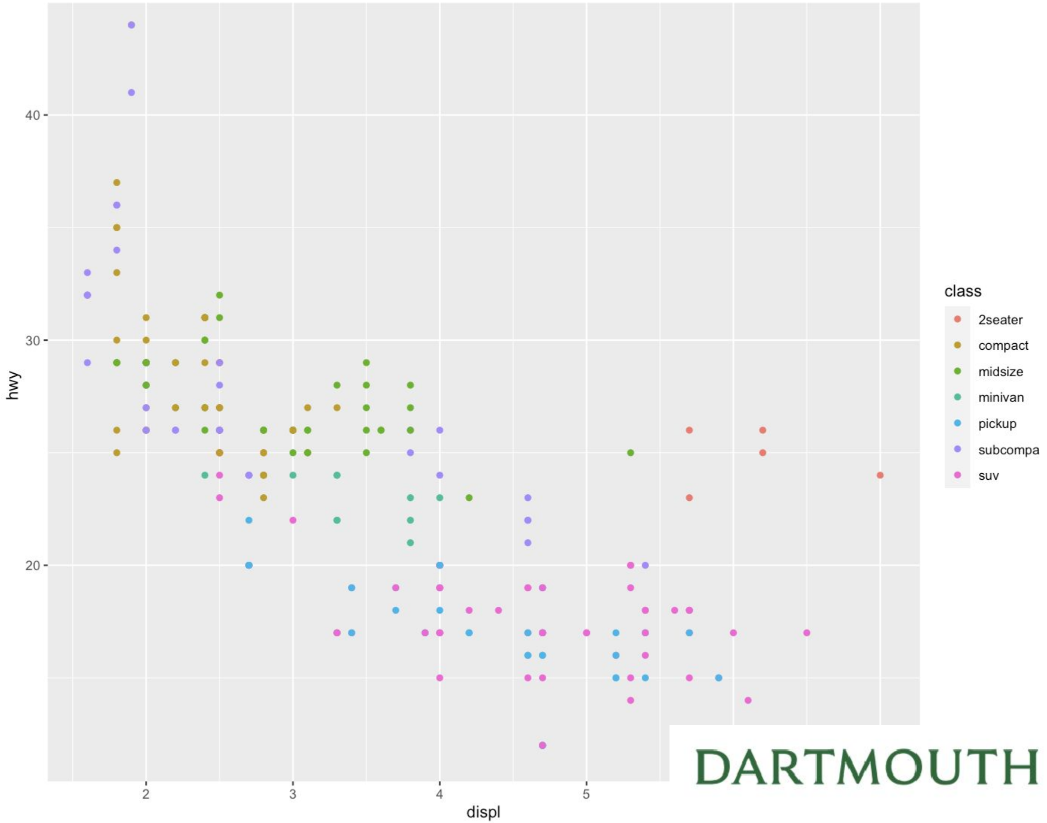
# exploratory plot, highway mpg vs displacement
# Use vars() to supply faceting variables:
p + facet_wrap(vars(class))

https://ggplot2.tidyverse.org/reference/facet_wrap.html



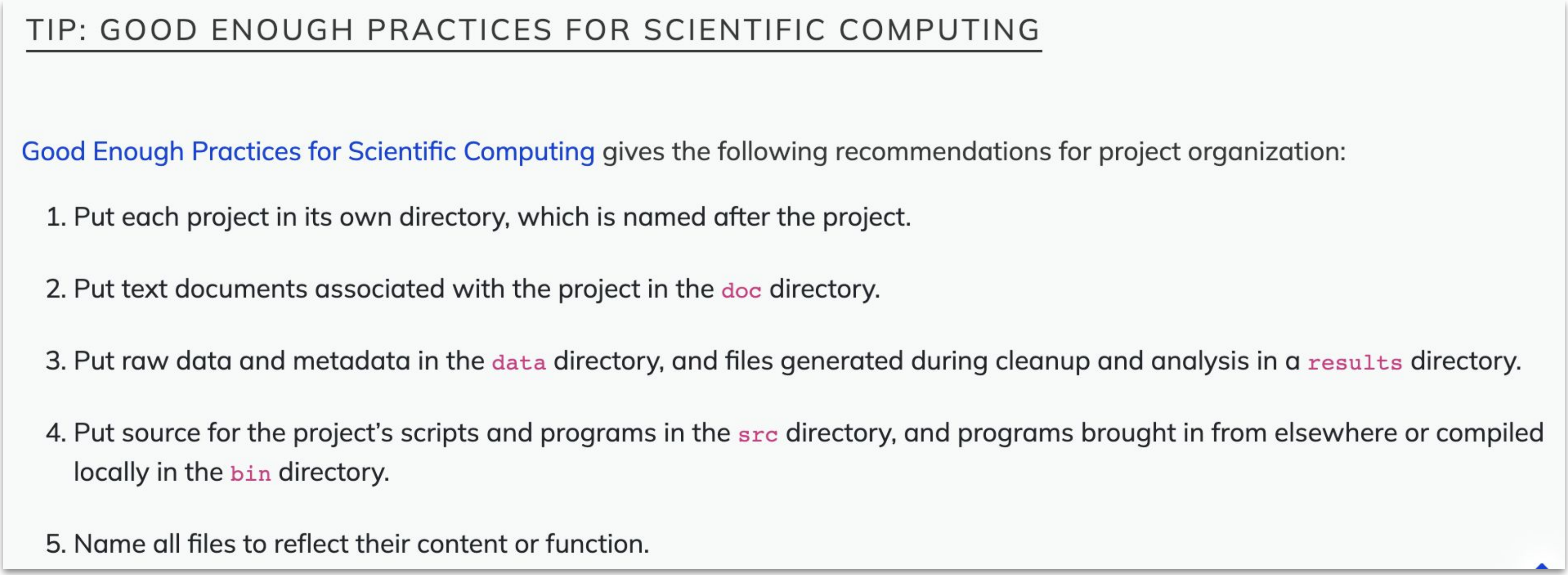
EDA facet wrap Tidy data, Tidyverse and GGPLOT

```
# The easiest way to get ggplot2 is
to install the whole tidyverse:

install.packages("tidyverse")

# Load the ggplot2 package

library(ggplot2)
```



Tidy data, Tidyverse and GGPLOT

```
# Replace 'data' with your actual data frame.
ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point()

# Create a ggplot with facet wrap
p <- ggplot(mpg, aes(displ, hwy)) + geom_point()

# exploratory plot, highway mpg vs displacement
# Use vars() to supply faceting variables:
p + facet_wrap(vars(class))
```

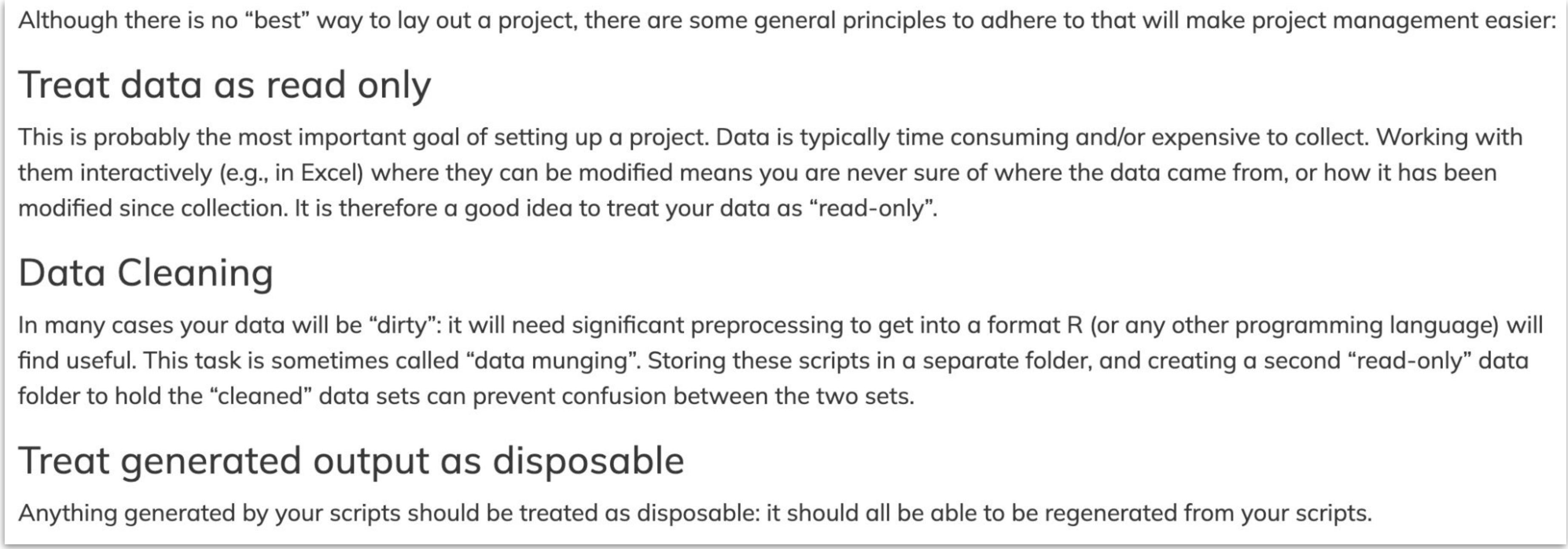


TIP: GOOD ENOUGH PRACTICES FOR SCIENTIFIC COMPUTING

Good Enough Practices for Scientific Computing gives the following recommendations for project organization:

1. Put each project in its own directory, which is named after the project.
2. Put text documents associated with the project in the `doc` directory.
3. Put raw data and metadata in the `data` directory, and files generated during cleanup and analysis in a `results` directory.
4. Put source for the project's scripts and programs in the `src` directory, and programs brought in from elsewhere or compiled locally in the `bin` directory.
5. Name all files to reflect their content or function.

Source: https://swcarpentry.github.io/r-novice-gapminder/02-project-intro.html



Although there is no “best” way to lay out a project, there are some general principles to adhere to that will make project management easier:

Treat data as read only

This is probably the most important goal of setting up a project. Data is typically time consuming and/or expensive to collect. Working with them interactively (e.g., in Excel) where they can be modified means you are never sure of where the data came from, or how it has been modified since collection. It is therefore a good idea to treat your data as “read-only”.

Data Cleaning

In many cases your data will be “dirty”: it will need significant preprocessing to get into a format R (or any other programming language) will find useful. This task is sometimes called “data munging”. Storing these scripts in a separate folder, and creating a second “read-only” data folder to hold the “cleaned” data sets can prevent confusion between the two sets.

Treat generated output as disposable

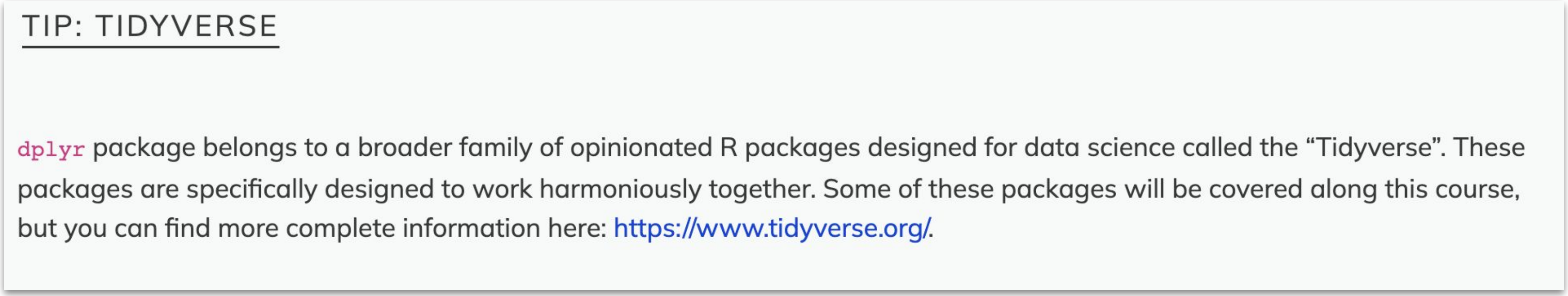
Anything generated by your scripts should be treated as disposable: it should all be able to be regenerated from your scripts.

More R topics

- Data frames https://swcarpentry.github.io/r-novice-gapminder/04-data-structures-part1.html
- Plotting with ggplot2 https://swcarpentry.github.io/r-novice-gapminder/08-plot-ggplot2.html
- Using Dplyr https://swcarpentry.github.io/r-novice-gapminder/13-dplyr.html

Tidy data & the Dplyr library

- Dplyr (see https://swcarpentry.github.io/r-novice-gapminder/13-dplyr.html)
  - select()
  - filter()
  - group_by()
  - summarize()



TIP: TIDYVERSE

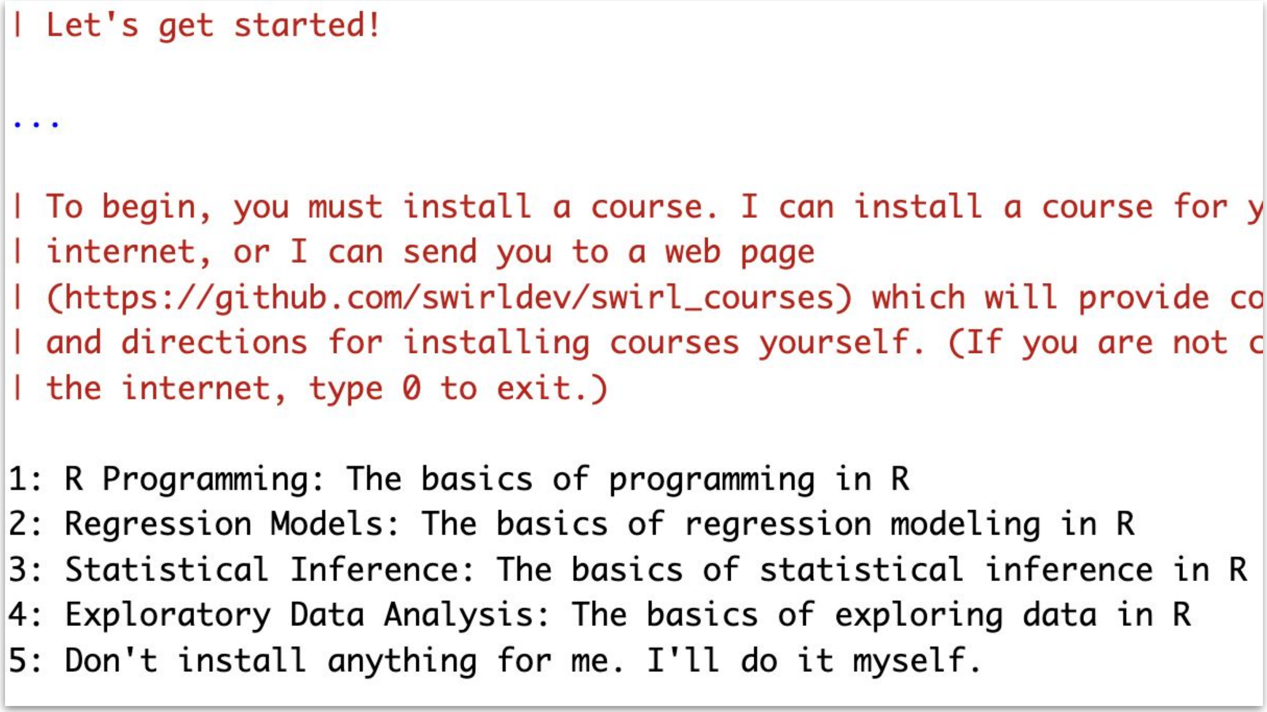
`dplyr` package belongs to a broader family of opinionated R packages designed for data science called the “Tidyverse”. These packages are specifically designed to work harmoniously together. Some of these packages will be covered along this course, but you can find more complete information here: https://www.tidyverse.org/.

Resources & Links

- Learning more R!
  - Swirlstats
  - Cheatsheets, for example https://rstudio.github.io/cheatsheets/html/rstudio-ide.html and https://rstudio.github.io/cheatsheets/html/data-visualization.html
  - Software Carpentry 'R for Reproducible Scientific Research' https://swcarpentry.github.io/r-novice-gapminder/ and W3 schools https://www.w3schools.com/r/
- Contact us:
  - https://researchguides.dartmouth.edu/data_management or ResearchDataHelp@groups.dartmouth.edu
- Upcoming workshops: dartgo.org/rradworkshops
- Materials will be sent out
  - Slides
  - Code & Data

Swirlstats

Swirlstats - https://swirlstats.com



```
| Let's get started!

...

| To begin, you must install a course. I can install a course for y
| internet, or I can send you to a web page
| (https://github.com/swirldev/swirl_courses) which will provide co
| and directions for installing courses yourself. (If you are not c
| the internet, type 0 to exit.)

1: R Programming: The basics of programming in R
2: Regression Models: The basics of regression modeling in R
3: Statistical Inference: The basics of statistical inference in R
4: Exploratory Data Analysis: The basics of exploring data in R
5: Don't install anything for me. I'll do it myself.
```

Feedback

Thanks for coming to our workshop!

We want to now learn from you about how we continue to present relevant workshops in the best way we can.

Please take a minute or so to fill out our form with your constructive feedback, we can't wait to hear from you!

dartgo.org/feedback

Questions?

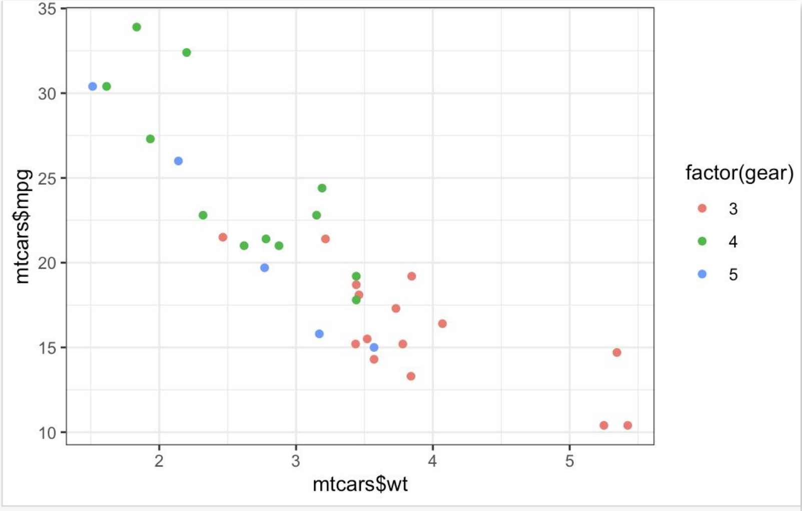
As always, feel free to reach out anytime.

Thanks for attending our workshop!

Supplemental materials

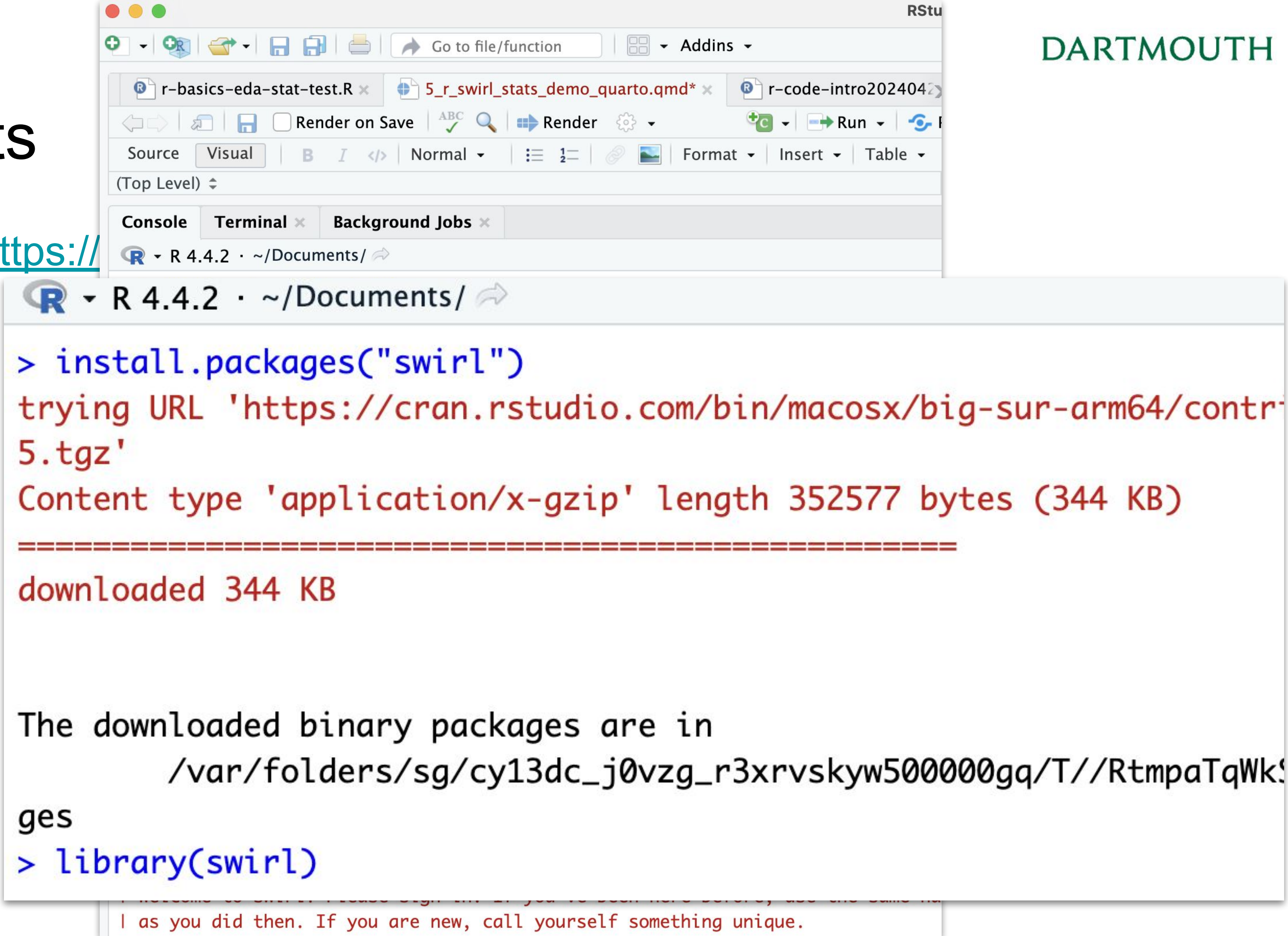
MPG vs Weight, GGPLOT

```
install.packages("ggplot2")
library(ggplot2)
mtcars

ggplot(mtcars, aes(mtcars$wt,mtcars$mpg)) +
geom_point()

ggplot(mtcars, aes(mtcars$wt,mtcars$mpg,
color=factor(gear))) + geom_point() +
theme_bw()
```



Swirlstats

Swirlstats - https://



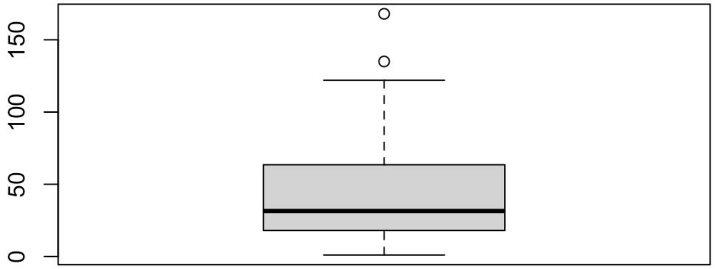
```
> install.packages("swirl")
trying URL 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contr
5.tgz'
Content type 'application/x-gzip' length 352577 bytes (344 KB)
==================================================
downloaded 344 KB


The downloaded binary packages are in
        /var/folders/sg/cy13dc_j0vzg_r3xrvskyw500000gq/T//RtmpaTqWk
ges
> library(swirl)
| as you did then. If you are new, call yourself something unique.
```

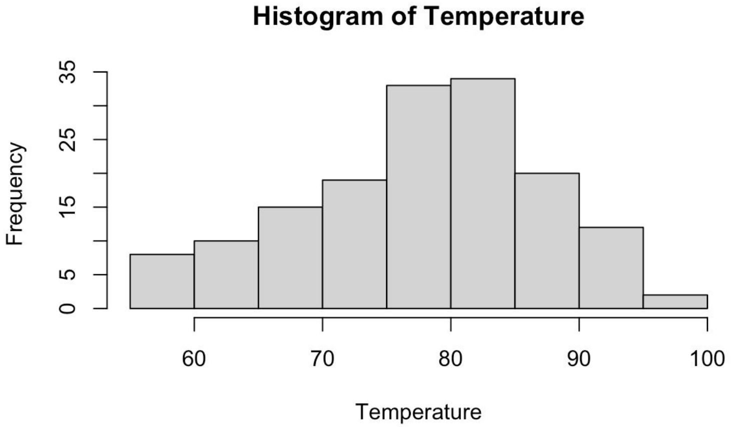
Exporting data: Write CSV

```
# write csv
# Import a dataset, look at the first six rows
carSpeeds <- read.csv(file = 'data/car-speeds.csv')
head(carSpeeds)
# subset the dataset
subset = head(carSpeeds,2)
Subset
# write a subset csv
write.csv(subset, file = 'data/car-speeds-subset.csv')
subset
```

Boxplot & Histogram

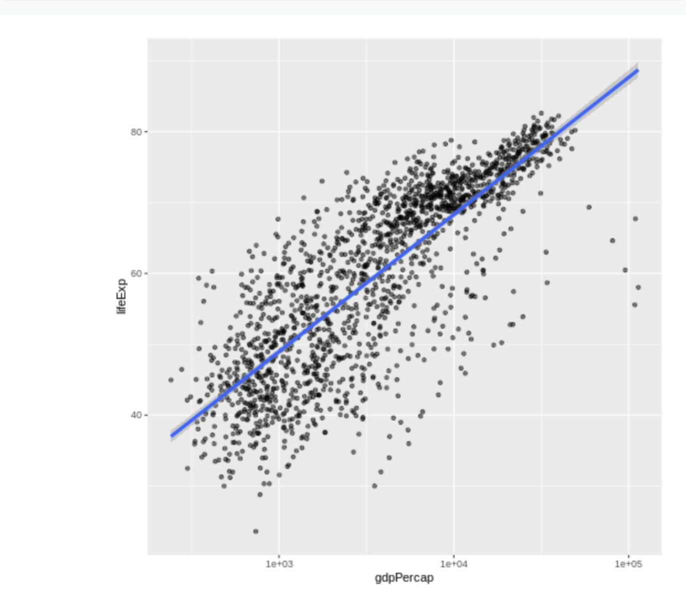
```
# Boxplot

str(airquality)

boxplot(airquality$Ozone)

#histogram

Temperature <- airquality$Temp

hist(Temperature)
```

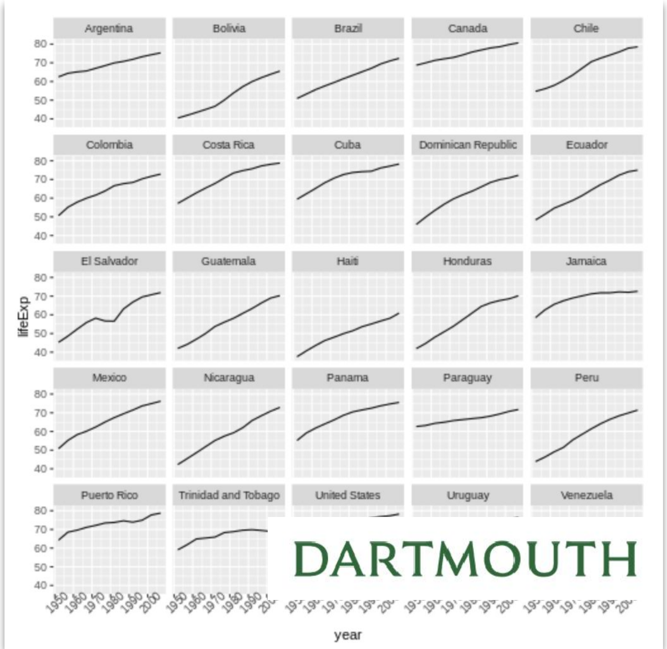




EDA and faceting with ggplot

https://swcarpentry.github.io/r-novice-gapminder/08-plot-ggplot2.html


lifeExp
gdpPercap
1e+03
1e+04
1e+05


Argentina
Bolivia
Brazil
Canada
Chile
Colombia
Costa Rica
Cuba
Dominican Republic
Ecuador
El Salvador
Guatemala
Haiti
Honduras
Jamaica
Mexico
Nicaragua
Panama
Paraguay
Peru
Puerto Rico
Trinidad and Tobago
United States
Uruguay
Venezuela
lifeExp
year