UBC Library Research Commons

Workshop: Intro to R and R Studio

Data Analysis Team:

- Matty Jullamon (GAA)
- Amir Michalovich (GAA)
- Jeremy Buhler (Data Librarian)
- Sarah Parker (Data Librarian)

Pre-workshop setup

Download and install R

For Windows:

1. Visit R Project (https://www.r-project.org/) to learn about R versions.
2. Download and install R from your preferred CRAN mirror here (https://cran.r-project.org/mirrors.html)
    A. Choose "0-Cloud" or a mirror site near you.

For Mac:

1. Check that your macOS system is up-to-date
2. Download and install R from The Comprehensive R Archive Network (https://cran.r-project.org/)

Download and install R studio

For Windows and Mac:

1. Download and install R Studio from here (https://rstudio.com/products/rstudio/download/#download)

Learning Objectives

- Become familiar with R and R studio environment.
- Learn the basic R programming language.
- Learn how to explore data in R.

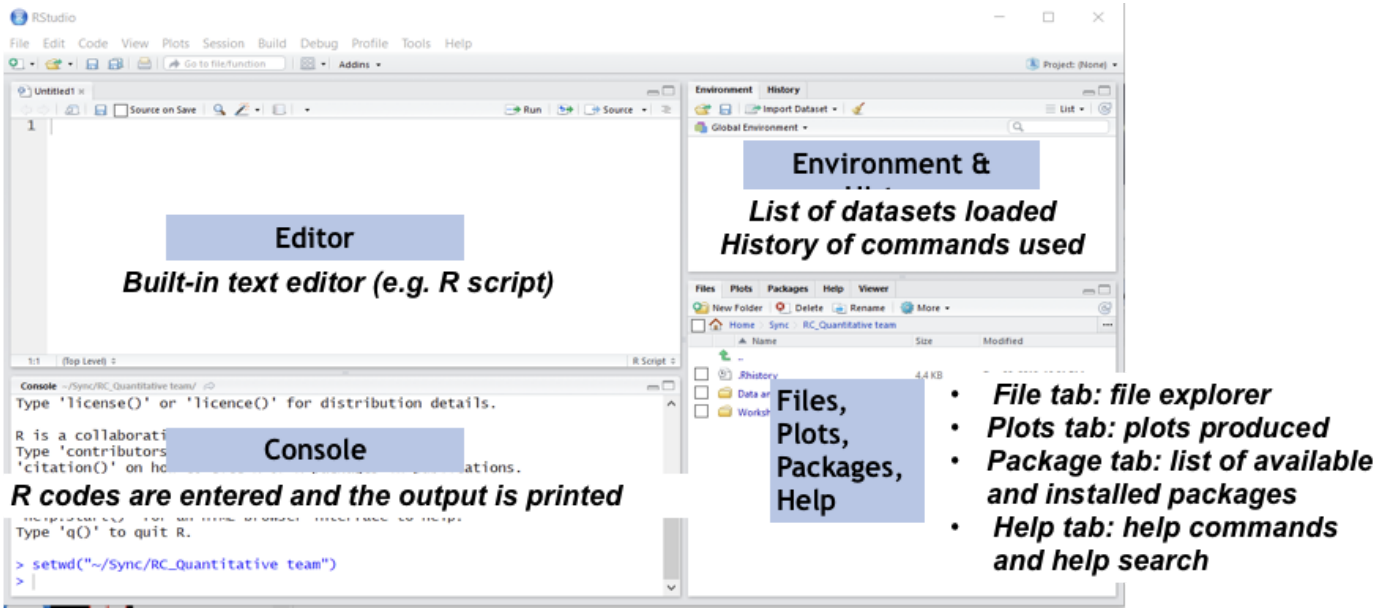
R vs. R Studio

- **R** is a programming language. We use R to run our codes and see their output.
- **R Studio** is another program which manages R in a user-friendly environment.

That is why, it is highly recommended to install both programs because they work together.

R Studio Environment

- Open R studio
- To create a new R script, click File > New file > R Script

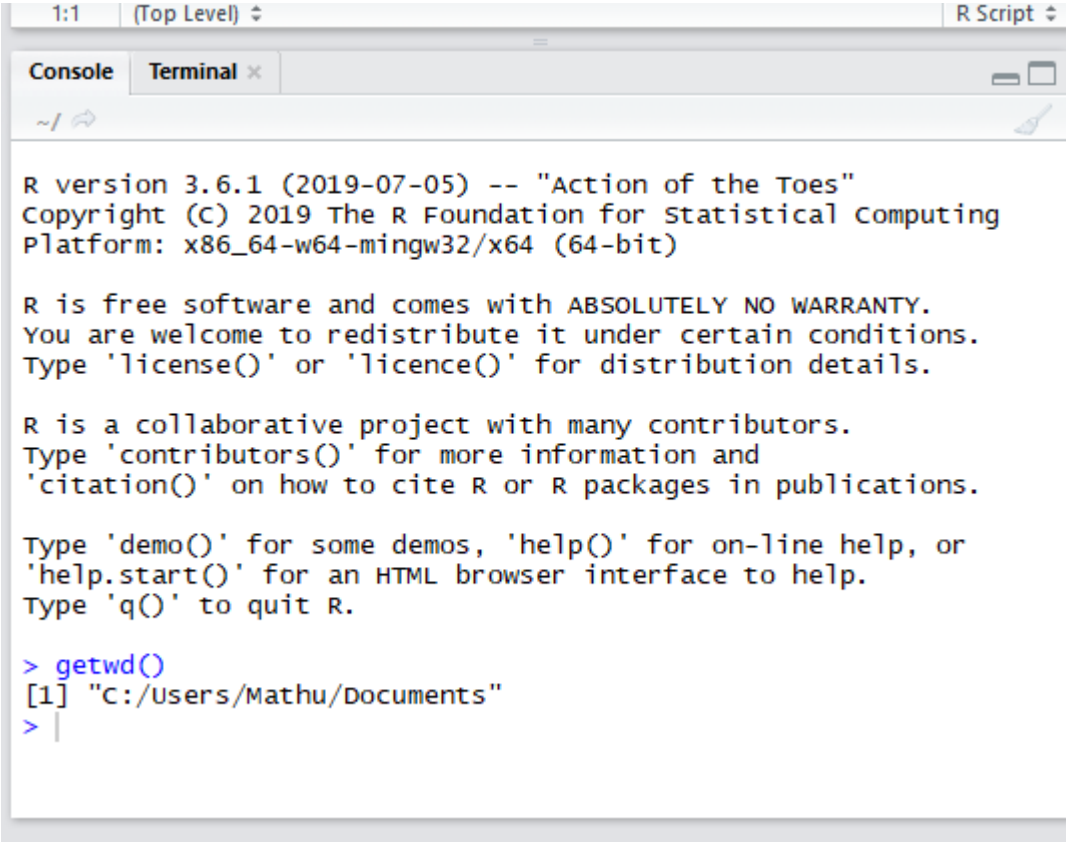


Working Directory

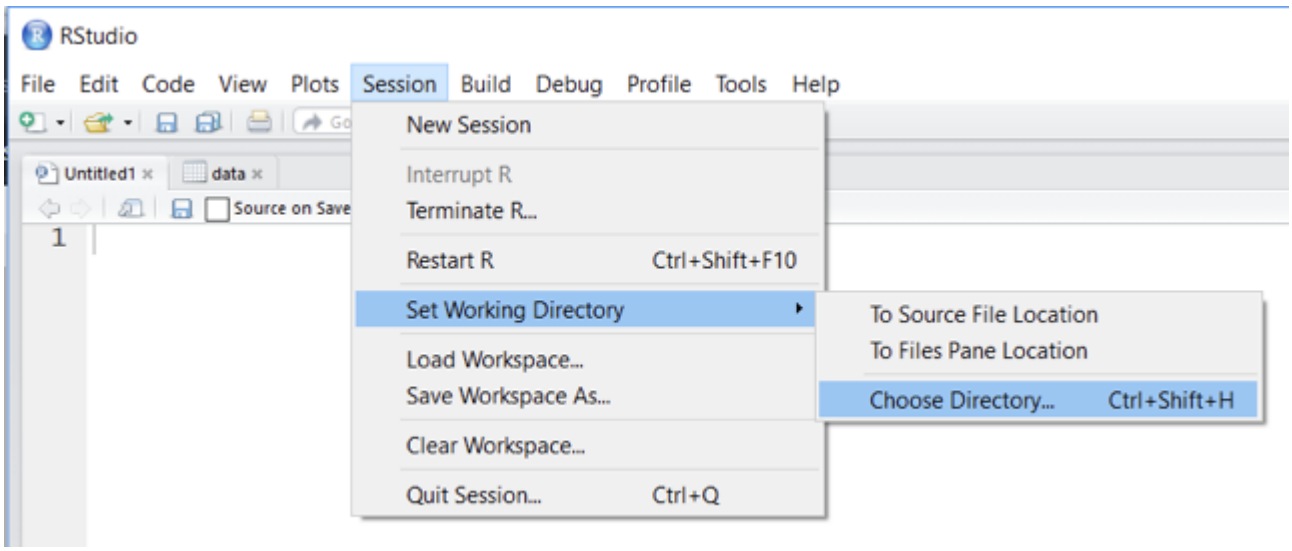
- Working directory is a folder/path where R reads and saves files.

Check your current working directory

- To check your current working directory, write and run `getwd()` in the console.



Set or Change your current working directory



- If you know the filepath to your new working directory, you can also write and run `setwd(dir)` in the console.

Data Input

Data types and structures

There are different types of data that can be stored in R. In many cases, our data is organized and stored as DataFrame.

A DataFrame:

Character	Numeric	Factor
A	1	High
B	2	Medium
C	3	Low

- Has multiple columns and rows
- Contains different types of variables
  - Numeric variables: numeric values with and without decimal places
  - Categorical variables: qualitative data that can be represented by characters or factors

Basic R commands

An operator is a symbol that intructs R to perform specific operation. Here are some basic operators which can be used in R:

Mathematical operations

Description: This allows R to perform methematics operations.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
<- or =	Assign value from the right to the left

Relational operations
Description: This allows R to compare variables.

Operator	Description
==	Equal to
!=	Not Equal to
>	Great than
<	Less than
>=	Great than or equal to
<=	Less than or equal to

Exercise #1

Calculate

- 1+1
- 4-9
- 19/3
- 14*6
- 4^3

Assign values to variables

- a <- 4
- a
- b <- 5
- b

Question: What do you notice in the Environment pane?

Calculate

- c <- 6 + a
- c
- d <- a + b + c
- d
- c==d

Getting Started

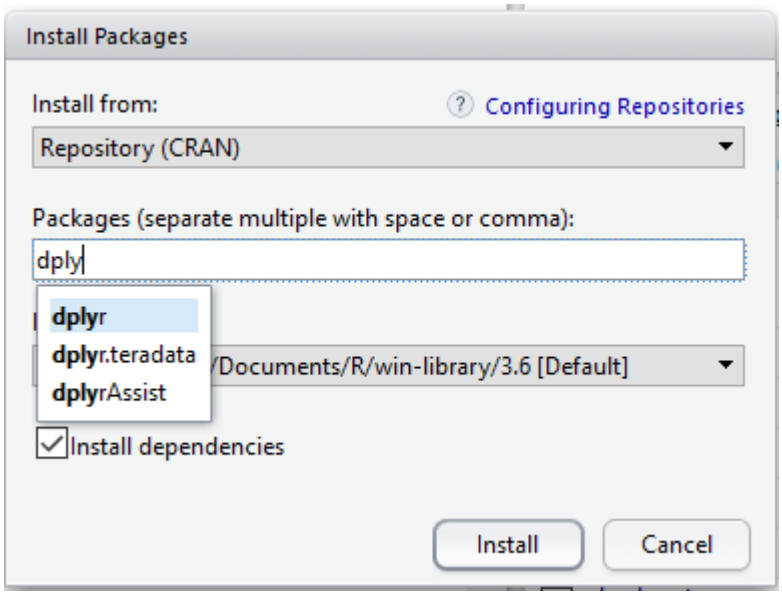
R package

R package is a library of prewritten code designed for a particular task or a collection of tasks. For today, we will mainly use 'dplyr' and 'ggplot2'.

Installing a new package

There are two options to install a new package.

1. Under Tools -> Install Package -> Search for "dplyr" and "ggplot2"



1. Write and run the following codes:

install.packages("dplyr")

install.packages("gglot2")

Loading installed packages

Use the library command to load any installed packages.

library(dplyr)

library(ggplot2)

Importing data from built-in R datasets

In [121]:
```
cars<-mtcars
head(cars) #see the first six rows of the dataframe

#How can you see the entire dataset?
# View(cars)

# How can you check the data structure?
# str(cars)
```

A data.frame: 6 × 11

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

Question: How many observations (number of rows)? How many variables (number of columns)?

- nrow(cars) #To check the number of rows
- ncol(cars) #To check the number of columns

Extra Info: Importing data from external data sources

CSV data file

- read.csv(file="mtcars.csv")

SPSS, SAS, or Stata data file

- install.packages("foreign")
- library(foreign)
- read.spss() # SPSS
- read.dta() # Stata
- read.ssd() # SAS

Data Manipulation

About the data

Motor Trend Car Road Tests: "The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models)" (R-core@R-project.org, 2020)

?mtcars #Execute this code to check the descriptions

`mtcars`

Format

A data frame with 32 observations on 11 (numeric) variables.

[, 1] mpg	Miles/(US) gallon
[, 2] cyl	Number of cylinders
[, 3] disp	Displacement (cu.in.)
[, 4] hp	Gross horsepower
[, 5] drat	Rear axle ratio
[, 6] wt	Weight (1000 lbs)
[, 7] qsec	1/4 mile time
[, 8] vs	Engine (0 = V-shaped, 1 = straight)
[, 9] am	Transmission (0 = automatic, 1 = manual)
[,10] gear	Number of forward gears
[,11] carb	Number of carburetors

Note

Henderson and Velleman (1981) comment in a footnote to Table 1: 'Hocking [original transcriber]'s noncrucial coding of the Mazda's rotary engine as a straight six-cylinder engine and the Porsche's flat engine as a V engine, as well as the inclusion of the diesel Mercedes 240D, have been retained to enable direct comparisons to be made with previous analyses.'

Source

Henderson and Velleman (1981), Building multiple regression models interactively. *Biometrics*, **37**, 391–411.

Using dplyr verbs to manage data

select ( )

- To selects columns based on columns names
- Useful when your data has many columns and you only need a subset of them

In [34]:
```
# Select specific columns such as the mpg, wt, vs, am, gear, carb
select(cars, mpg, wt, vs, am, gear, carb)
```

A data.frame: 32 × 6

	mpg	wt	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Mazda RX4	21.0	2.620	0	1	4	4
Mazda RX4 Wag	21.0	2.875	0	1	4	4
Datsun 710	22.8	2.320	1	1	4	1
Hornet 4 Drive	21.4	3.215	1	0	3	1
Hornet Sportabout	18.7	3.440	0	0	3	2
Valiant	18.1	3.460	1	0	3	1
Duster 360	14.3	3.570	0	0	3	4
Merc 240D	24.4	3.190	1	0	4	2
Merc 230	22.8	3.150	1	0	4	2
Merc 280	19.2	3.440	1	0	4	4
Merc 280C	17.8	3.440	1	0	4	4
Merc 450SE	16.4	4.070	0	0	3	3
Merc 450SL	17.3	3.730	0	0	3	3
Merc 450SLC	15.2	3.780	0	0	3	3
Cadillac Fleetwood	10.4	5.250	0	0	3	4
Lincoln Continental	10.4	5.424	0	0	3	4
Chrysler Imperial	14.7	5.345	0	0	3	4
Fiat 128	32.4	2.200	1	1	4	1
Honda Civic	30.4	1.615	1	1	4	2
Toyota Corolla	33.9	1.835	1	1	4	1
Toyota Corona	21.5	2.465	1	0	3	1
Dodge Challenger	15.5	3.520	0	0	3	2
AMC Javelin	15.2	3.435	0	0	3	2
Camaro Z28	13.3	3.840	0	0	3	4
Pontiac Firebird	19.2	3.845	0	0	3	2
Fiat X1-9	27.3	1.935	1	1	4	1
Porsche 914-2	26.0	2.140	0	1	5	2
Lotus Europa	30.4	1.513	1	1	5	2
Ford Pantera L	15.8	3.170	0	1	5	4
Ferrari Dino	19.7	2.770	0	1	5	6
Maserati Bora	15.0	3.570	0	1	5	8
Volvo 142E	21.4	2.780	1	1	4	2

In [35]:
```
# Select all columns except for qsec
select(cars, -qsec) # use subtraction operator
```

A data.frame: 32 × 10

	mpg	cyl	disp	hp	drat	wt	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Mazda RX4	21.0	6	160.0	110	3.90	2.620	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	1	1	4	2

filter ( )

Filter rows based on conditions

In [37]:
```
#Filter for cars with 3 gears
filter(cars, gear == "3")
```

A data.frame: 15 × 11

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2

In [39]:
```
# Filter rows for cars with 3 gears and have displacement greater than 300 cub
ic inches

filter(cars, gear == "3", disp > 300.00)
```

A data.frame: 9 × 11

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Duster 360	14.3	8	360	245	3.21	3.570	15.84	0	0	3	4
Cadillac Fleetwood	10.4	8	472	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440	230	3.23	5.345	17.42	0	0	3	4
Dodge Challenger	15.5	8	318	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400	175	3.08	3.845	17.05	0	0	3	2

Question: What relational operator should you use for greater than and equal to 300 cubic inches?

Solution:

filter(cars, gear == "3", disp >= 300.00)

In [40]:
```
# Filter rows for cars with 3 or 4 gears
# Use vertical line for Or condition

filter(cars, gear == "3" | gear == "4")
```

A data.frame: 27 × 11

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

arrange ( )

Arrange rows by ascending or decending order

In [41]:
```
# Arrange the number of cylinders from small to large
arrange(cars, cyl)
```

A data.frame: 32 × 11

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8

In [42]:
```
# Arrange the number of cylinders from large to small
arrange(cars, desc(cyl))
```

A data.frame: 32 × 11

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

mutate( )

Add new columns on previously existing data with mutate ().

In [68]:
```
# Create a new column called 'l_100km'. To covert miles per US gallon to the n
umber of liters per 100km, divide 235.215 by mpg
mutate(cars, l_100km = 235.215/mpg)
```

A data.frame: 32 × 12

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	l_100km
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4	11.200714
21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4	11.200714
22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1	10.316447
21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1	10.991355
18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2	12.578342
18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1	12.995304
14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4	16.448601
24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2	9.639959
22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2	10.316447
19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4	12.250781
17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4	13.214326
16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3	14.342378
17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3	13.596243
15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3	15.474671
10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4	22.616827
10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4	22.616827
14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4	16.001020
32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1	7.259722
30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2	7.737336
33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1	6.938496
21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1	10.940233
15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2	15.175161
15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2	15.474671
13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4	17.685338
19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2	12.250781
27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1	8.615934
26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2	9.046731
30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2	7.737336
15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4	14.887025
19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6	11.939848
15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8	15.681000
21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2	10.991355

Multiple Functions

Piping

What if you want to perform multiple functions in R? Use Pipe operator (%>%) in the dplyr package. It allows you to perform multiple functions without using nested parentheses. This is how piping looks like:

In [100]:
```
#DataFrame%>%
    #function to execute first %>%
        #function to execute second %>%
            #function to execute third
```

In [102]:
```
cars %>%
  select(wt,vs,gear) %>% #select wt,vs, gear columns
    filter(gear == 3 | gear == 4) %>% #filter cars with 3 or 4 gears
      group_by(vs) %>% # split data into the types of engine
        summarise(mean_wt= mean(wt)) %>%#summarize the mean weight of each gro
up
            mutate(mean_wt= round(mean_wt,2)) # Use mutate to modify existing
 columns. Round mean weight to 2 decimal points.
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

A tibble: 2 × 2

vs	mean_wt
<dbl>	<dbl>
0	3.91
1	2.70

Exercise #2

Calculate fuel consumption in liter per km for cars with 6 cylinders

1. Filter for cars with 6 cyls
2. Use mutate to convert mpg to liters per 100 km
3. View the mean of liters per 100 km for cars with 6 cylinders

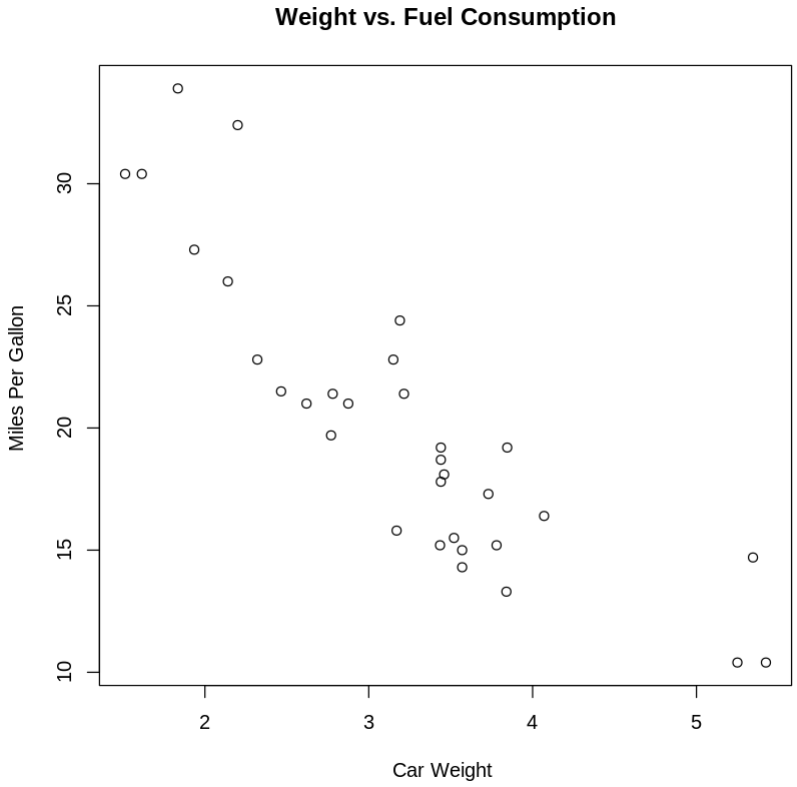
In [ ]:
```
# Answers:

cars %>%
  filter(cyl == 6) %>%
    mutate(l_100km = 235.215/mpg) %>%
      summarise(mean_l_100km= mean(l_100km))
```

Basic Plots in R

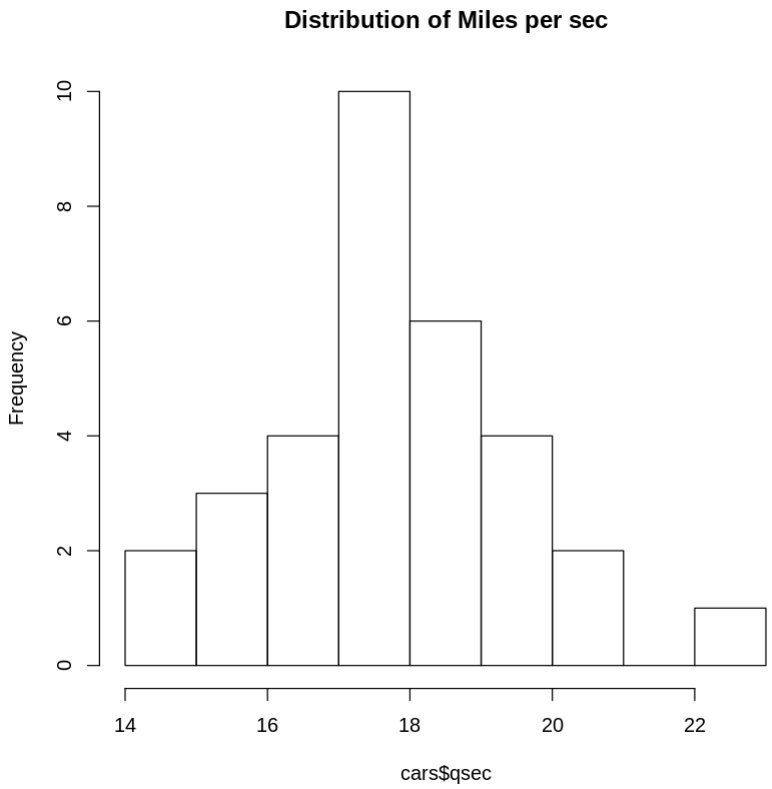
Scatterplot

In [108]:
```
# Show relationships between two variables
plot(cars$wt,cars$mpg, main = "Weight vs. Fuel Consumption", # Use $ to call a
column from the dataframe
   xlab="Car Weight ", ylab="Miles Per Gallon ") # Rename x and y axis
```



Histogram

In [112]:
```
# visualize the distribution of a continuous data
hist(cars$qsec, main = "Distribution of miles per sec")
```



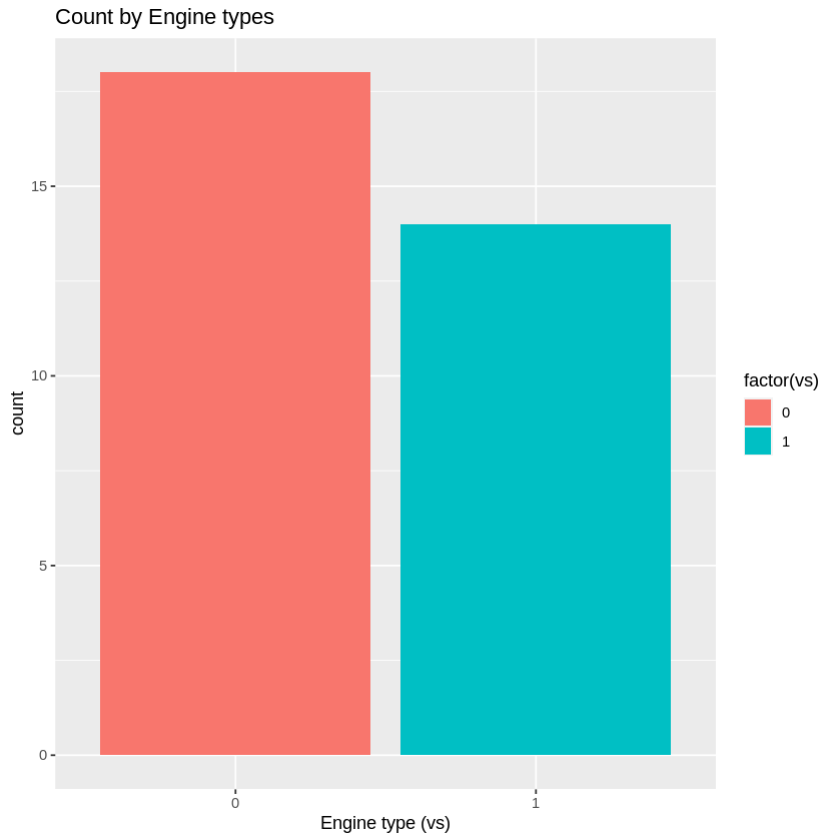
Question: How do you rename cars$qsec to qsec?

Solution:

hist(cars$qsec, main = "Distribution of miles per sec", xlab= "qsec")

Barchart with ggplot2

In [122]:
```
# Plot count value by Engine type
p <- ggplot(cars, aes(factor(vs), fill= factor(vs)))+
    geom_bar()+
    labs(title="Count by Engine types ",x="Engine type (vs)", y = "count")
p
```



Question: What is the difference between a histogram and a bar chart?

Line Graph with ggplot2

Note: Use a line graph to display change over time

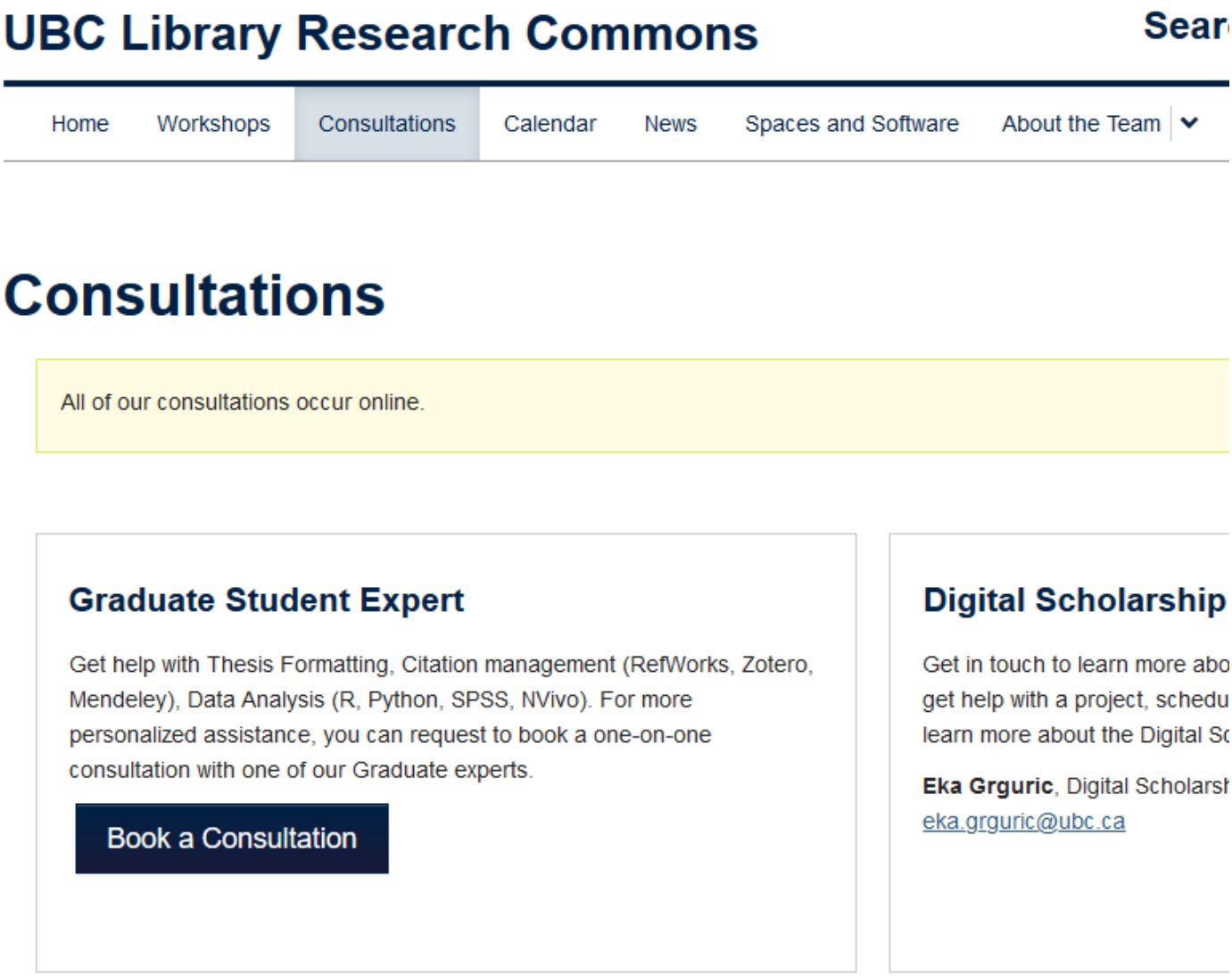
In [117]:
```
# a <- ggplot(cars, aes(x=wt, y=mpg)) +
#     geom_line()
# a
```

Save dataframe as CSV file

In [119]:
```
write.csv(cars, file= "cars.csv")
```

Questions?

Research Commons has a team of GAAs happy to help you analyze your data. Book a consultation online.



Reference

R-core@R-project.org.(2020,June 15). *mtcars*. https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/mtcars (https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/mtcars)