An Introduction to R

MacOdrum Library

February 23, 2021

Before We Begin

Go to:

http://madgic.library.carleton.ca/deposit/RWorkshop/PhD%20Africa/

And download the four files

Save those files in C:/Users/*YOU*/Downloads

Overview

- About R
- What can R be used for
- The First Steps
- Activities
- Bonus Data (Time Permitting)

What is R?

- Language and Environment originally designed for statistical computing and graphics
- Scripting Language
- Provides a lot of flexibility
- Can be used for multiple tasks beyond just statistical analysis

Why is R good?

- Free
- Open Source
- Simple but powerful
- Lots of help online
- Can work with all different types of data/documents
- Lots of different "packages" that are used for specific analysis
- Replication is Easy
- So much that you can do with R

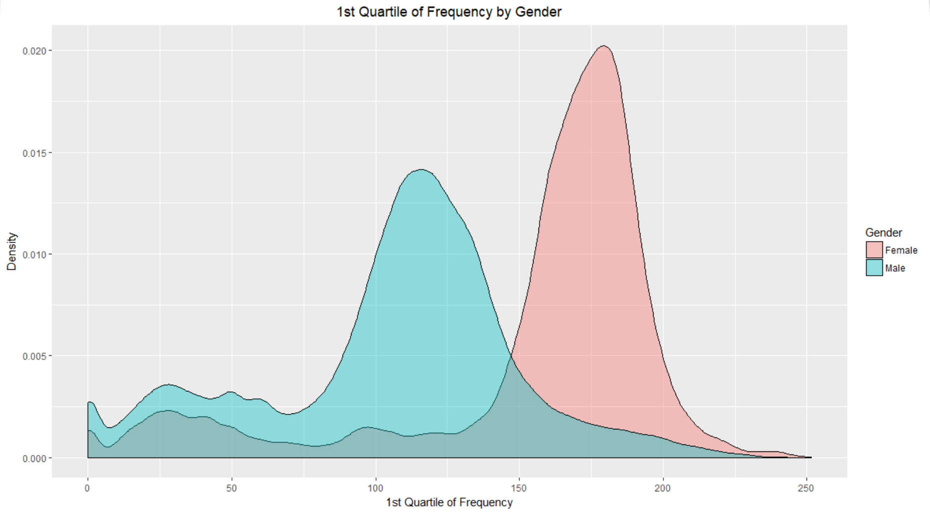
Where to get help?

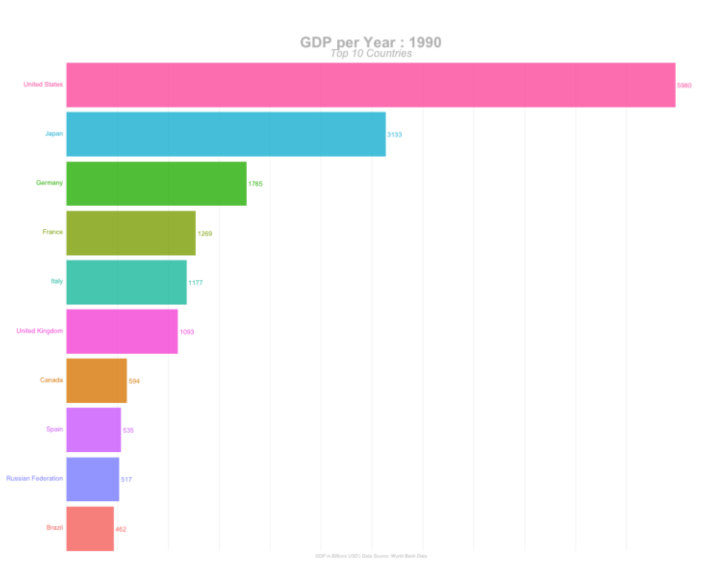
- Google
- Youtube
- Stack Overflow
- R Cheat sheets
- Lots of eBooks found in your university library

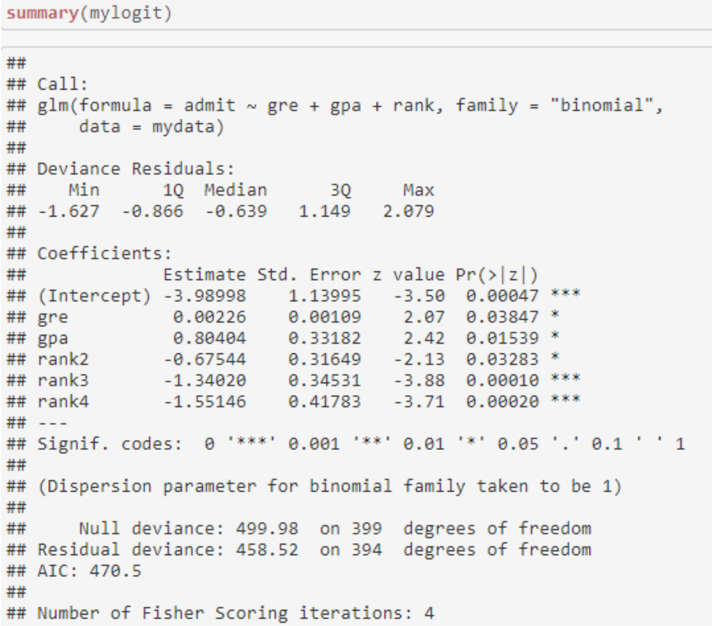
What can I do with R?

- **Statistical Analysis**
  - Basic Stats
  - Deep Learning, Machine Learning
  - Text Analysis
  - Geospatial Analysis
  - Audio Analysis
- **Cleaning Messy Data**
  - Cleaning up spreadsheets, Data files
- **Automation**
  - Web Scraping
  - Virtual Machine
  - Roll through multiple files in one go
- **Visualisations**

Basic Stats





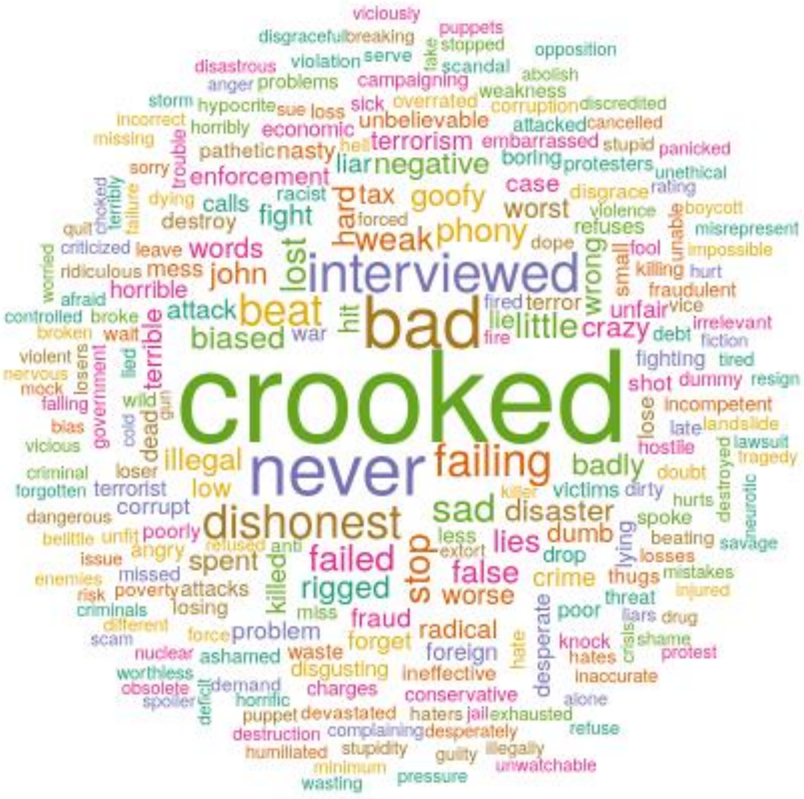


```
summary(mylogit)
```

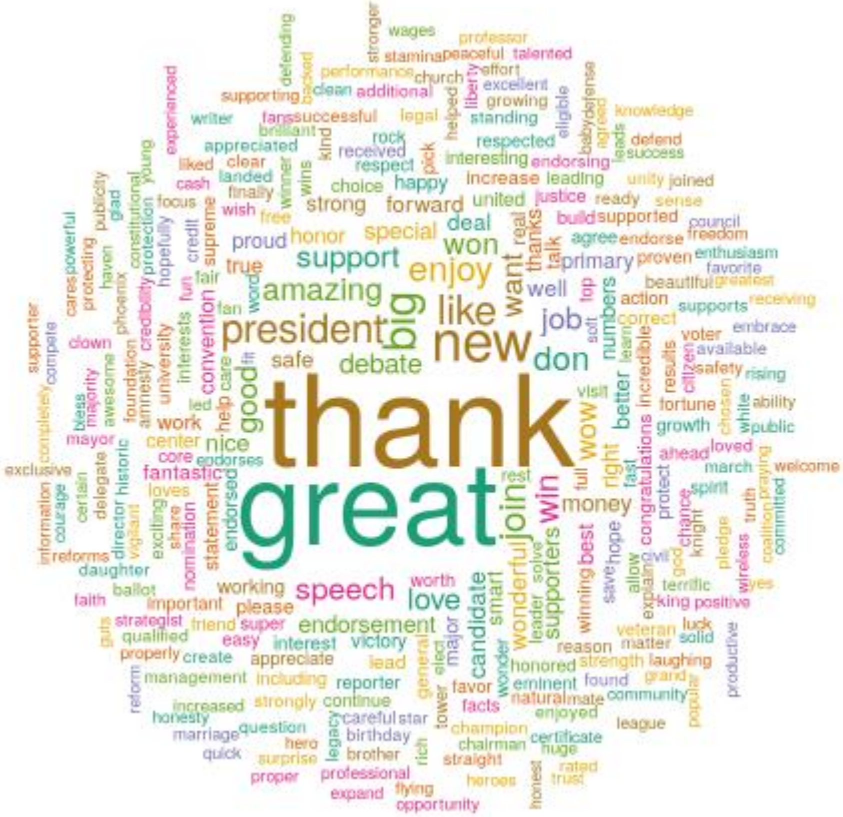
```
##
## Call:
## glm(formula = admit ~ gre + gpa + rank, family = "binomial",
##     data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.627   -0.866   -0.639    1.149    2.079
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.98998    1.13995   -3.50  0.00047 ***
## gre          0.00226    0.00109    2.07  0.03847 *
## gpa          0.80404    0.33182    2.42  0.01539 *
## rank2       -0.67544    0.31649   -2.13  0.03283 *
## rank3       -1.34020    0.34531   -3.88  0.00010 ***
## rank4       -1.55146    0.41783   -3.71  0.00020 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 458.52  on 394  degrees of freedom
## AIC: 470.5
##
## Number of Fisher Scoring iterations: 4
```

Text Analysis

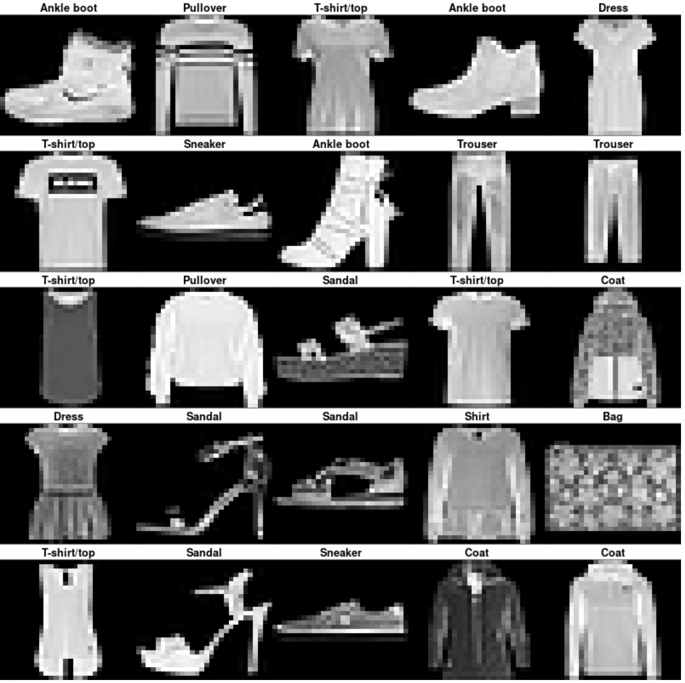
Negative Tweets

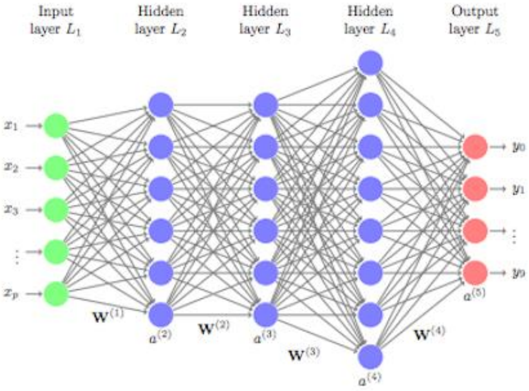


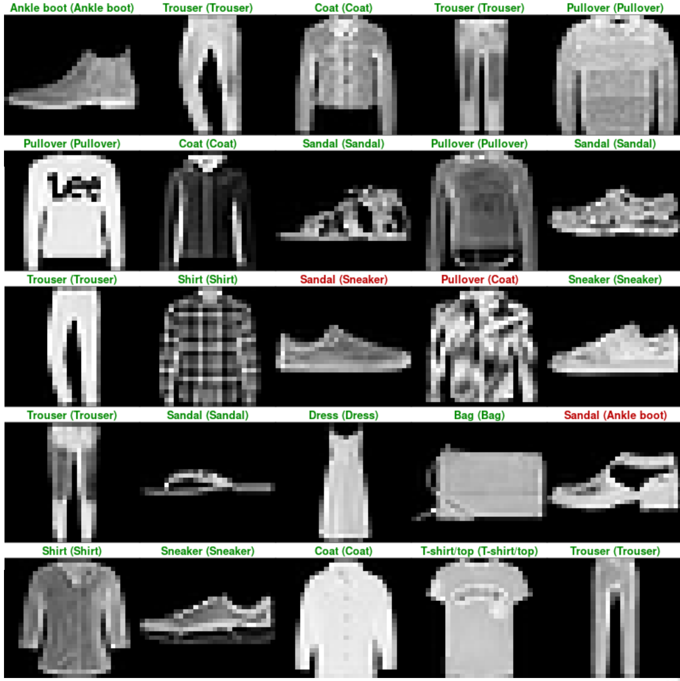
Positive Tweets



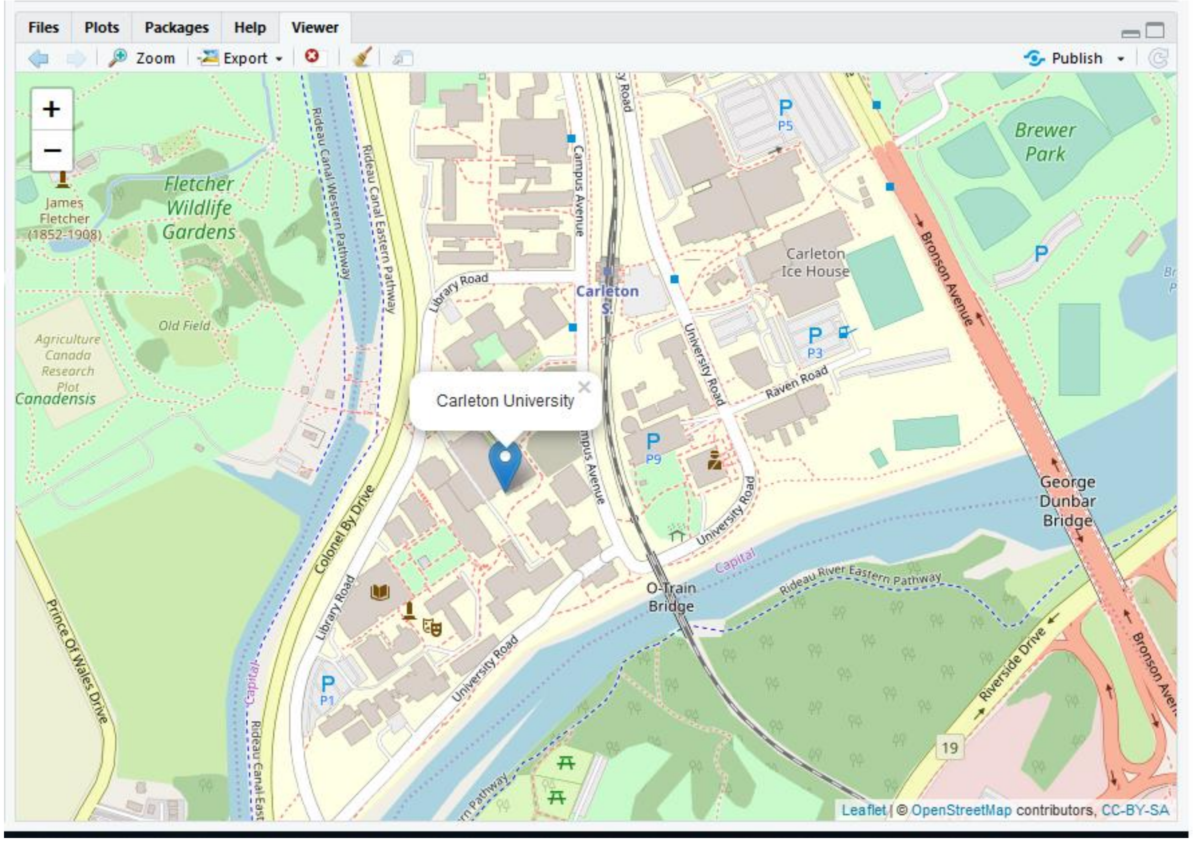
Deep Learning

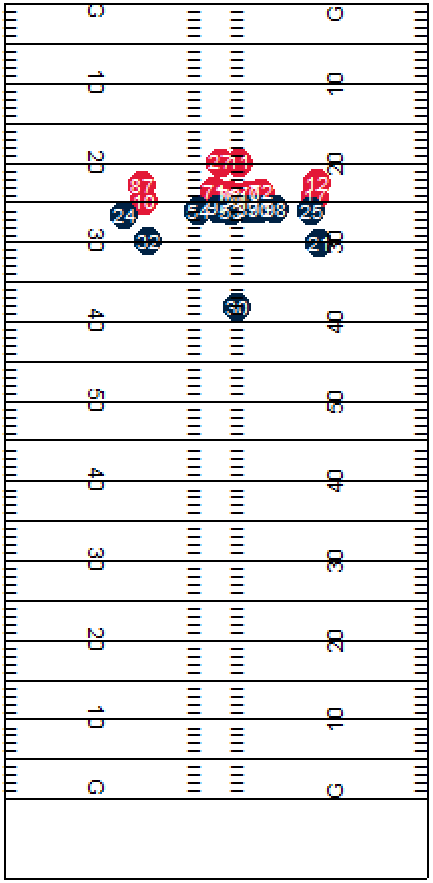






GeoSpatial and Spatial Analysis

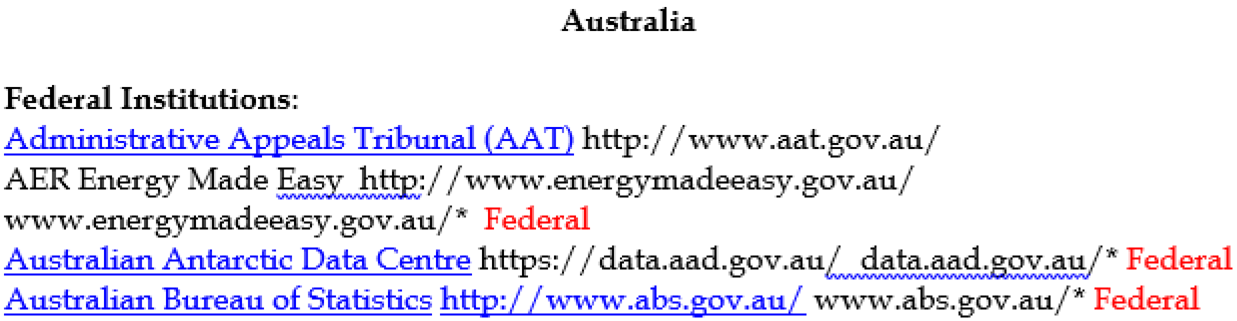




Carleton UNIVERSITY
Canada's Capital University

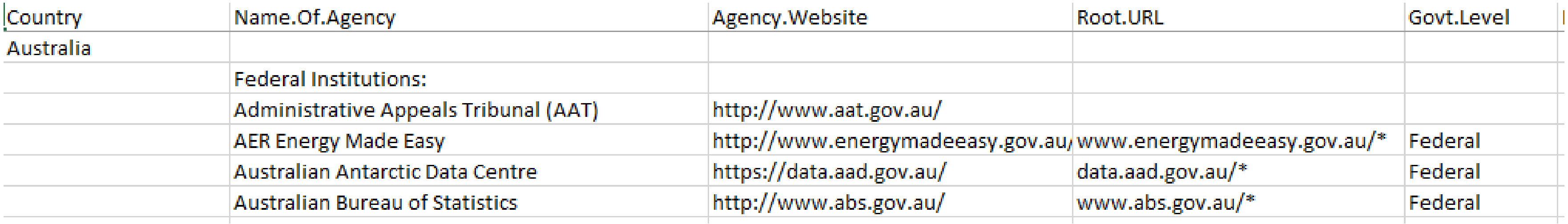
Automating

- Transform a Word Doc to Excel Spreadsheet



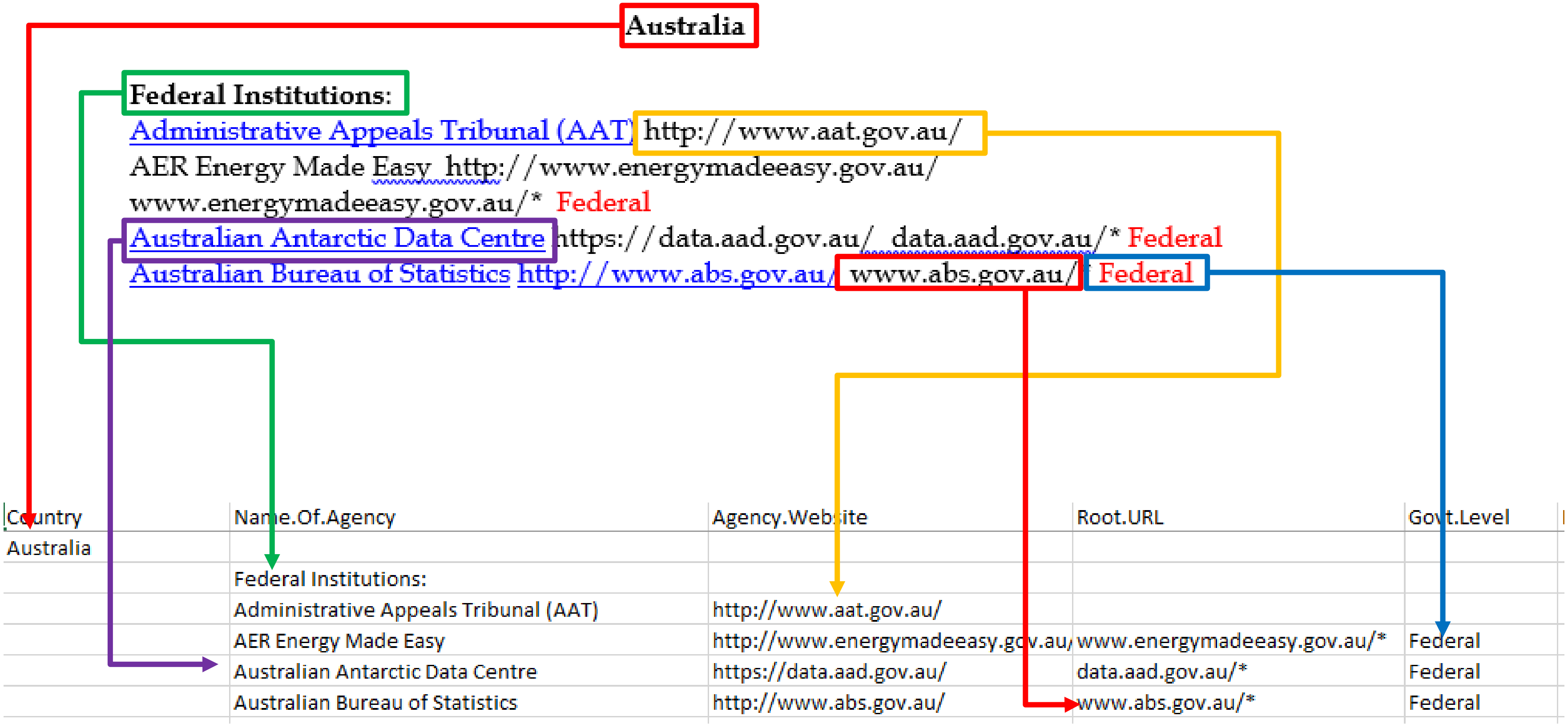
Australia

Federal Institutions:
Administrative Appeals Tribunal (AAT) http://www.aat.gov.au/
AER Energy Made Easy http://www.energymadeeasy.gov.au/
www.energymadeeasy.gov.au/* Federal
Australian Antarctic Data Centre https://data.aad.gov.au/ data.aad.gov.au/* Federal
Australian Bureau of Statistics http://www.abs.gov.au/ www.abs.gov.au/* Federal



Country	Name.Of.Agency	Agency.Website	Root.URL	Govt.Level
Australia				
	Federal Institutions:			
	Administrative Appeals Tribunal (AAT)	http://www.aat.gov.au/		
	AER Energy Made Easy	http://www.energymadeeasy.gov.au/	www.energymadeeasy.gov.au/*	Federal
	Australian Antarctic Data Centre	https://data.aad.gov.au/	data.aad.gov.au/*	Federal
	Australian Bureau of Statistics	http://www.abs.gov.au/	www.abs.gov.au/*	Federal

Automating



Australia

Federal Institutions:

Administrative Appeals Tribunal (AAT) http://www.aat.gov.au/

AER Energy Made Easy http://www.energymadeeasy.gov.au/

www.energymadeeasy.gov.au/* Federal

Australian Antarctic Data Centre https://data.aad.gov.au/ data.aad.gov.au/* Federal

Australian Bureau of Statistics http://www.abs.gov.au/ www.abs.gov.au/ Federal

Country	Name.Of.Agency	Agency.Website	Root.URL	Govt.Level
Australia				
	Federal Institutions:			
	Administrative Appeals Tribunal (AAT)	http://www.aat.gov.au/		
	AER Energy Made Easy	http://www.energymadeeasy.gov.au	www.energymadeeasy.gov.au/*	Federal
	Australian Antarctic Data Centre	https://data.aad.gov.au/	data.aad.gov.au/*	Federal
	Australian Bureau of Statistics	http://www.abs.gov.au/	www.abs.gov.au/*	Federal

Web Crawling

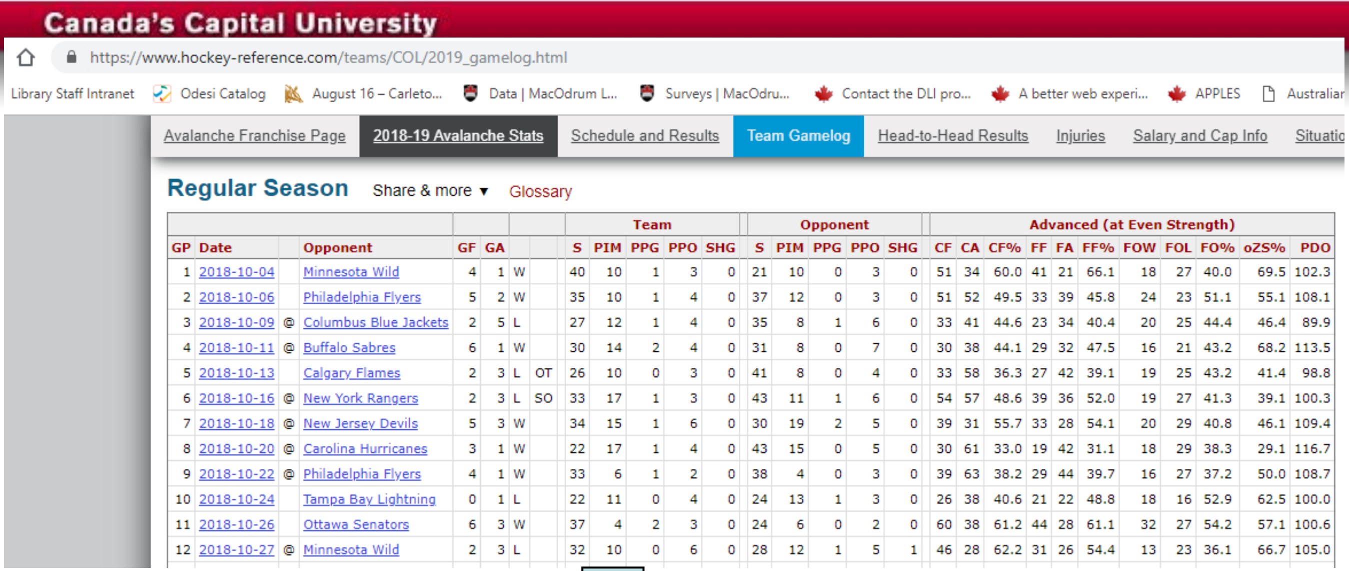
I removed the video that was here so that the slides could be sent through email.

If you are interested in learning more about web crawling, look up the RSelenium package

Carleton UNIVERSITY

Canada's Capital University

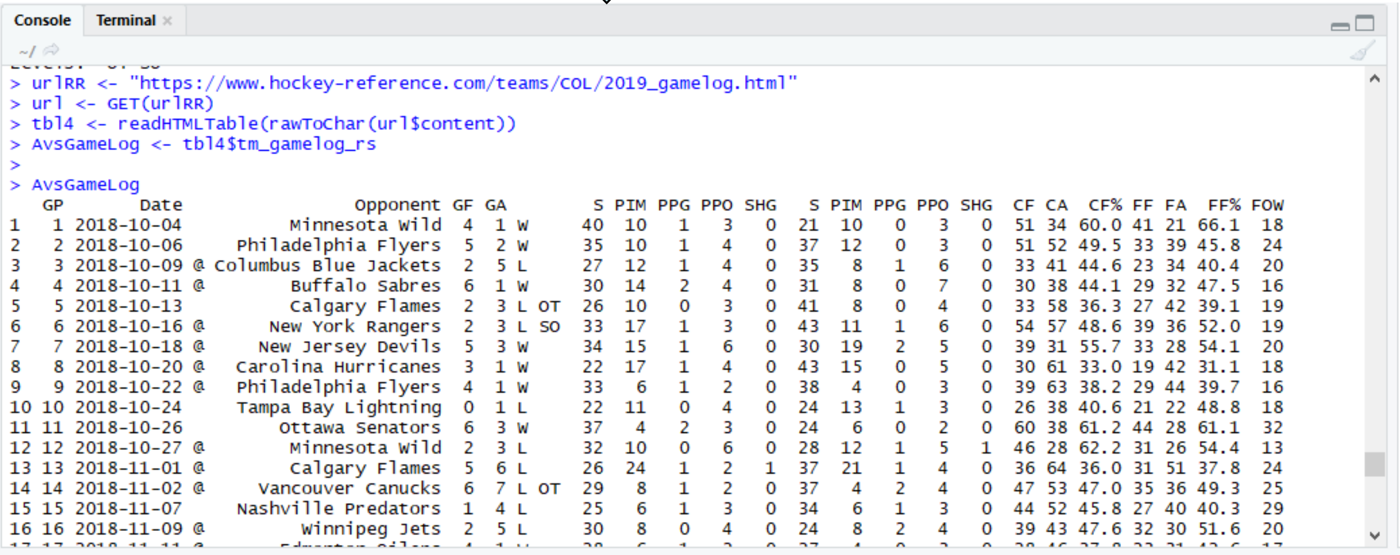
Scraping Data



Regular Season

								Team					Opponent					Advanced (at Even Strength)										
GP	Date		Opponent	GF	GA			S	PIM	PPG	PPO	SHG	S	PIM	PPG	PPO	SHG	CF	CA	CF%	FF	FA	FF%	FOW	FOL	FO%	oZS%	PDO
1	2018-10-04		Minnesota Wild	4	1	W		40	10	1	3	0	21	10	0	3	0	51	34	60.0	41	21	66.1	18	27	40.0	69.5	102.3
2	2018-10-06		Philadelphia Flyers	5	2	W		35	10	1	4	0	37	12	0	3	0	51	52	49.5	33	39	45.8	24	23	51.1	55.1	108.1
3	2018-10-09	@	Columbus Blue Jackets	2	5	L		27	12	1	4	0	35	8	1	6	0	33	41	44.6	23	34	40.4	20	25	44.4	46.4	89.9
4	2018-10-11	@	Buffalo Sabres	6	1	W		30	14	2	4	0	31	8	0	7	0	30	38	44.1	29	32	47.5	16	21	43.2	68.2	113.5
5	2018-10-13		Calgary Flames	2	3	L	OT	26	10	0	3	0	41	8	0	4	0	33	58	36.3	27	42	39.1	19	25	43.2	41.4	98.8
6	2018-10-16	@	New York Rangers	2	3	L	SO	33	17	1	3	0	43	11	1	6	0	54	57	48.6	39	36	52.0	19	27	41.3	39.1	100.3
7	2018-10-18	@	New Jersey Devils	5	3	W		34	15	1	6	0	30	19	2	5	0	39	31	55.7	33	28	54.1	20	29	40.8	46.1	109.4
8	2018-10-20	@	Carolina Hurricanes	3	1	W		22	17	1	4	0	43	15	0	5	0	30	61	33.0	19	42	31.1	18	29	38.3	29.1	116.7
9	2018-10-22	@	Philadelphia Flyers	4	1	W		33	6	1	2	0	38	4	0	3	0	39	63	38.2	29	44	39.7	16	27	37.2	50.0	108.7
10	2018-10-24		Tampa Bay Lightning	0	1	L		22	11	0	4	0	24	13	1	3	0	26	38	40.6	21	22	48.8	18	16	52.9	62.5	100.0
11	2018-10-26		Ottawa Senators	6	3	W		37	4	2	3	0	24	6	0	2	0	60	38	61.2	44	28	61.1	32	27	54.2	57.1	100.6
12	2018-10-27	@	Minnesota Wild	2	3	L		32	10	0	6	0	28	12	1	5	1	46	28	62.2	31	26	54.4	13	23	36.1	66.7	105.0

Grab Data from the Web and Bring it into R



```
> urlRR <- "https://www.hockey-reference.com/teams/COL/2019_gamelog.html"
> url <- GET(urlRR)
> tbl4 <- readHTMLTable(rawToChar(url$content))
> AvsGameLog <- tbl4$tm_gamelog_rs
>
> AvsGameLog
   GP       Date                Opponent GF GA        S PIM PPG PPO SHG   S PIM PPG PPO SHG  CF CA  CF% FF FA  FF% FOW
1   1 2018-10-04          Minnesota Wild  4  1 W     40  10   1   3   0  21  10   0   3   0  51 34 60.0 41 21 66.1  18
2   2 2018-10-06     Philadelphia Flyers  5  2 W     35  10   1   4   0  37  12   0   3   0  51 52 49.5 33 39 45.8  24
3   3 2018-10-09 @ Columbus Blue Jackets  2  5 L     27  12   1   4   0  35   8   1   6   0  33 41 44.6 23 34 40.4  20
4   4 2018-10-11 @        Buffalo Sabres  6  1 W     30  14   2   4   0  31   8   0   7   0  30 38 44.1 29 32 47.5  16
5   5 2018-10-13          Calgary Flames  2  3 L OT  26  10   0   3   0  41   8   0   4   0  33 58 36.3 27 42 39.1  19
6   6 2018-10-16 @      New York Rangers  2  3 L SO  33  17   1   3   0  43  11   1   6   0  54 57 48.6 39 36 52.0  19
7   7 2018-10-18 @     New Jersey Devils  5  3 W     34  15   1   6   0  30  19   2   5   0  39 31 55.7 33 28 54.1  20
8   8 2018-10-20 @   Carolina Hurricanes  3  1 W     22  17   1   4   0  43  15   0   5   0  30 61 33.0 19 42 31.1  18
9   9 2018-10-22 @   Philadelphia Flyers  4  1 W     33   6   1   2   0  38   4   0   3   0  39 63 38.2 29 44 39.7  16
10 10 2018-10-24     Tampa Bay Lightning  0  1 L     22  11   0   4   0  24  13   1   3   0  26 38 40.6 21 22 48.8  18
11 11 2018-10-26         Ottawa Senators  6  3 W     37   4   2   3   0  24   6   0   2   0  60 38 61.2 44 28 61.1  32
12 12 2018-10-27 @        Minnesota Wild  2  3 L     32  10   0   6   0  28  12   1   5   1  46 28 62.2 31 26 54.4  13
13 13 2018-11-01 @        Calgary Flames  5  6 L     26  24   1   2   1  37  21   1   4   0  36 64 36.0 31 51 37.8  24
14 14 2018-11-02 @     Vancouver Canucks  6  7 L OT  29   8   1   2   0  37   4   2   4   0  47 53 47.0 35 36 49.3  25
15 15 2018-11-07     Nashville Predators  1  4 L     25   6   1   3   0  34   6   1   3   0  44 52 45.8 27 40 40.3  29
16 16 2018-11-09 @        Winnipeg Jets  2  5 L     30   8   0   4   0  24   8   2   4   0  39 43 47.6 32 30 51.6  20
```

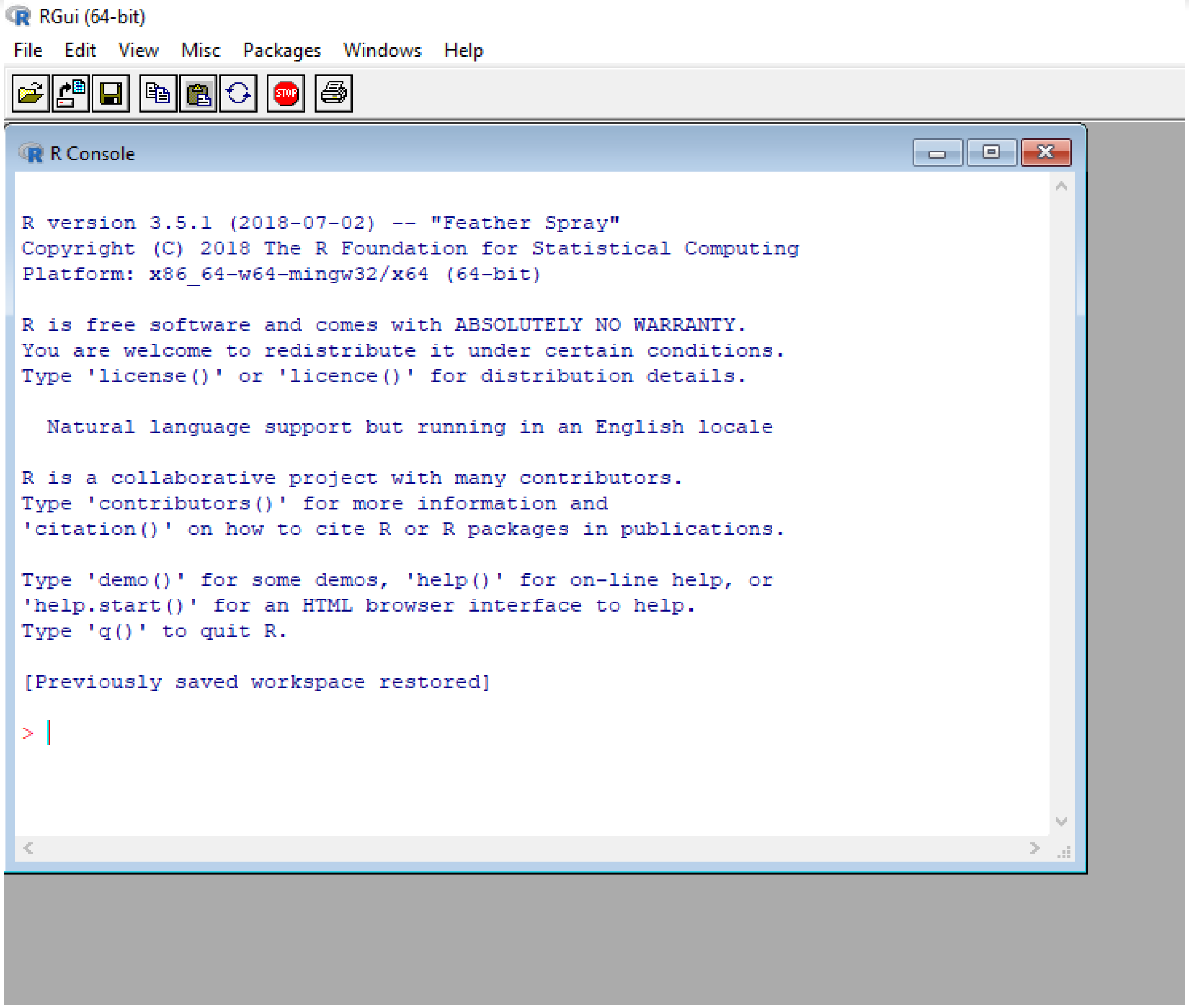
What You Can Do with R

SO MUCH
MORE

The First Steps

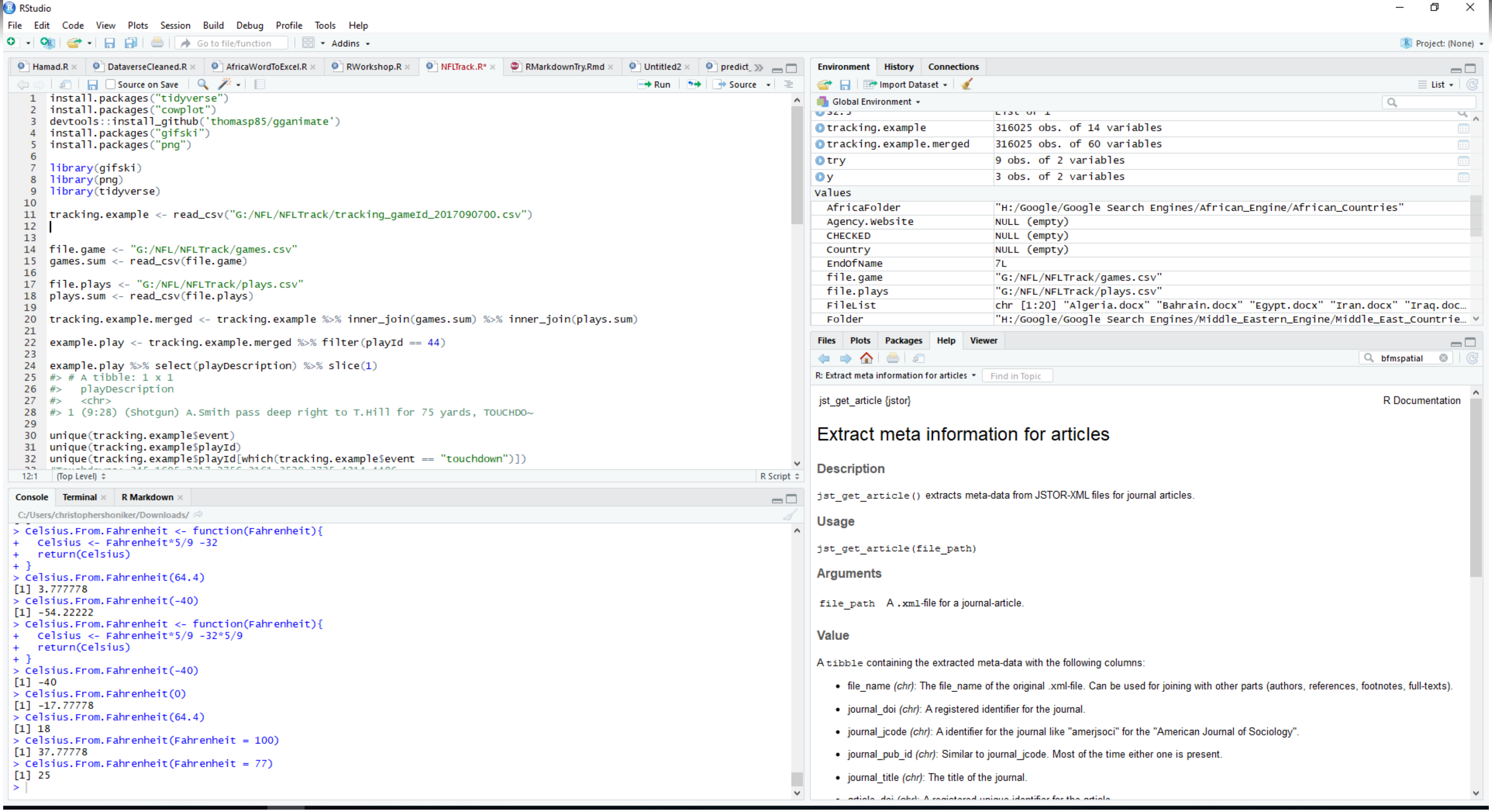
- Objects/Variables
- Data Types
- Functions
- Reading and Writing Data
- Operators
- Conditional Statements
- Loops

This is R



```
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> |
```

This is R + RStudio



```r
install.packages("tidyverse")
install.packages("cowplot")
devtools::install_github('thomasp85/gganimate')
install.packages("gifski")
install.packages("png")

library(gifski)
library(png)
library(tidyverse)

tracking.example <- read_csv("G:/NFL/NFLTrack/tracking_gameId_2017090700.csv")


file.game <- "G:/NFL/NFLTrack/games.csv"
games.sum <- read_csv(file.game)

file.plays <- "G:/NFL/NFLTrack/plays.csv"
plays.sum <- read_csv(file.plays)

tracking.example.merged <- tracking.example %>% inner_join(games.sum) %>% inner_join(plays.sum)

example.play <- tracking.example.merged %>% filter(playId == 44)

example.play %>% select(playDescription) %>% slice(1)
#> # A tibble: 1 x 1
#>   playDescription
#>   <chr>
#> 1 (9:28) (Shotgun) A.Smith pass deep right to T.Hill for 75 yards, TOUCHDO~

unique(tracking.example$event)
unique(tracking.example$playId)
unique(tracking.example$playId[which(tracking.example$event == "touchdown")])
```

```
> Celsius.From.Fahrenheit <- function(Fahrenheit){
+   Celsius <- Fahrenheit*5/9 -32
+   return(Celsius)
+ }
> Celsius.From.Fahrenheit(64.4)
[1] 3.777778
> Celsius.From.Fahrenheit(-40)
[1] -54.22222
> Celsius.From.Fahrenheit <- function(Fahrenheit){
+   Celsius <- Fahrenheit*5/9 -32*5/9
+   return(Celsius)
+ }
> Celsius.From.Fahrenheit(-40)
[1] -40
> Celsius.From.Fahrenheit(0)
[1] -17.77778
> Celsius.From.Fahrenheit(64.4)
[1] 18
> Celsius.From.Fahrenheit(Fahrenheit = 100)
[1] 37.77778
> Celsius.From.Fahrenheit(Fahrenheit = 77)
[1] 25
>
```

tracking.example	316025 obs. of 14 variables
tracking.example.merged	316025 obs. of 60 variables
try	9 obs. of 2 variables
y	3 obs. of 2 variables
Values	
AfricaFolder	"H:/Google/Google Search Engines/African_Engine/African_Countries"
Agency.Website	NULL (empty)
CHECKED	NULL (empty)
Country	NULL (empty)
EndOfName	7L
file.game	"G:/NFL/NFLTrack/games.csv"
file.plays	"G:/NFL/NFLTrack/plays.csv"
FileList	chr [1:20] "Algeria.docx" "Bahrain.docx" "Egypt.docx" "Iran.docx" "Iraq.doc…
Folder	"H:/Google/Google Search Engines/Middle_Eastern_Engine/Middle_East_Countrie…

jst_get_article {jstor}

R Documentation

Extract meta information for articles

Description

`jst_get_article()` extracts meta-data from JSTOR-XML files for journal articles.

Usage

`jst_get_article(file_path)`

Arguments

`file_path` A `.xml`-file for a journal-article.

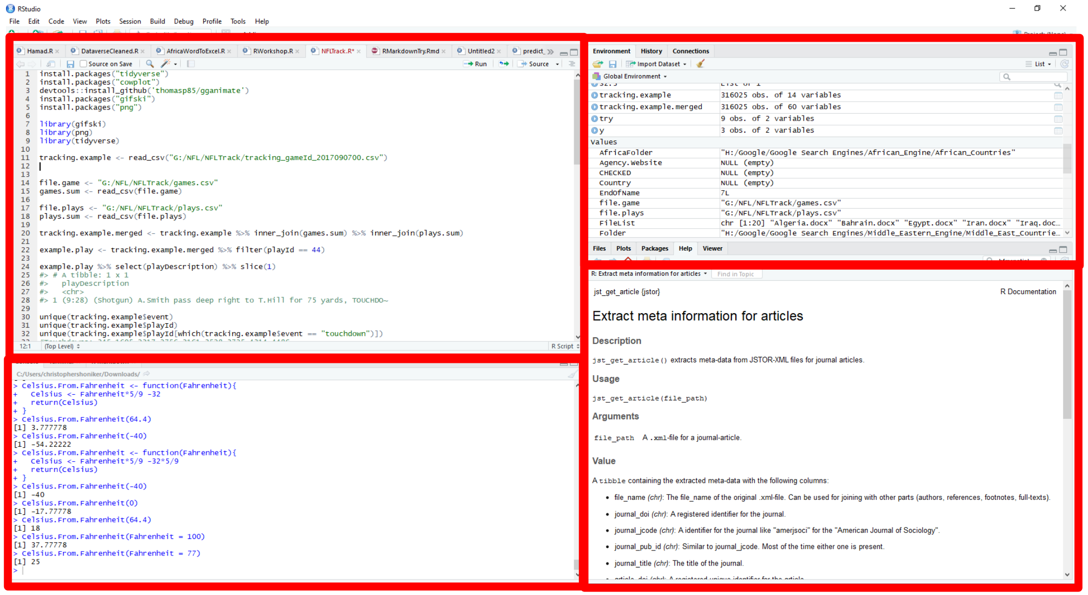
Value

A `tibble` containing the extracted meta-data with the following columns:

- file_name *(chr)*: The file_name of the original .xml-file. Can be used for joining with other parts (authors, references, footnotes, full-texts).
- journal_doi *(chr)*: A registered identifier for the journal.
- journal_jcode *(chr)*: A identifier for the journal like "amerjsoci" for the "American Journal of Sociology".
- journal_pub_id *(chr)*: Similar to journal_jcode. Most of the time either one is present.
- journal_title *(chr)*: The title of the journal.
- article_doi *(chr)*: A registered unique identifier for the article.

RStudio

- Open RStudio
- 4 Main Areas (Each Corner)
  - Script
  - Console
  - Environment/History
  - Help/Plot



```r
install.packages("tidyverse")
install.packages("cowplot")
devtools::install_github('thomasp85/gganimate')
install.packages("gifski")
install.packages("png")

library(gifski)
library(png)
library(tidyverse)

tracking.example <- read_csv("G:/NFL/NFLTrack/tracking_gameId_2017090700.csv")


file.game <- "G:/NFL/NFLTrack/games.csv"
games.sum <- read_csv(file.game)

file.plays <- "G:/NFL/NFLTrack/plays.csv"
plays.sum <- read_csv(file.plays)

tracking.example.merged <- tracking.example %>% inner_join(games.sum) %>% inner_join(plays.sum)

example.play <- tracking.example.merged %>% filter(playId == 44)

example.play %>% select(playDescription) %>% slice(1)
#> # A tibble: 1 x 1
#>   playDescription
#>   <chr>
#> 1 (9:28) (Shotgun) A.Smith pass deep right to T.Hill for 75 yards, TOUCHDO~

unique(tracking.example$event)
unique(tracking.example$playId)
unique(tracking.example$playId[which(tracking.example$event == "touchdown")])
```

C:/Users/christophershoniker/Downloads/

```
> Celsius.From.Fahrenheit <- function(Fahrenheit){
+   Celsius <- Fahrenheit*5/9 -32
+   return(Celsius)
+ }
> Celsius.From.Fahrenheit(64.4)
[1] 3.777778
> Celsius.From.Fahrenheit(-40)
[1] -54.22222
> Celsius.From.Fahrenheit <- function(Fahrenheit){
+   Celsius <- Fahrenheit*5/9 -32*5/9
+   return(Celsius)
+ }
> Celsius.From.Fahrenheit(-40)
[1] -40
> Celsius.From.Fahrenheit(0)
[1] -17.77778
> Celsius.From.Fahrenheit(64.4)
[1] 18
> Celsius.From.Fahrenheit(Fahrenheit = 100)
[1] 37.77778
> Celsius.From.Fahrenheit(Fahrenheit = 77)
[1] 25
>
```

```
tracking.example          316025 obs. of 14 variables
tracking.example.merged   316025 obs. of 60 variables
try                       9 obs. of 2 variables
y                         3 obs. of 2 variables
Values
  AfricaFolder            "H:/Google/Google Search Engines/African_Engine/African_Countries"
  Agency.Website          NULL (empty)
  CHECKED                 NULL (empty)
  Country                 NULL (empty)
  EndOfName               7L
  file.game               "G:/NFL/NFLTrack/games.csv"
  file.plays              "G:/NFL/NFLTrack/plays.csv"
  FileList                chr [1:20] "Algeria.docx" "Bahrain.docx" "Egypt.docx" "Iran.docx" "Iraq.doc…
  Folder                  "H:/Google/Google Search Engines/Middle_Eastern_Engine/Middle_East_Countrie…
```

jst_get_article {jstor} R Documentation

Extract meta information for articles

Description

`jst_get_article()` extracts meta-data from JSTOR-XML files for journal articles.

Usage

`jst_get_article(file_path)`

Arguments

`file_path` A `.xml`-file for a journal-article.

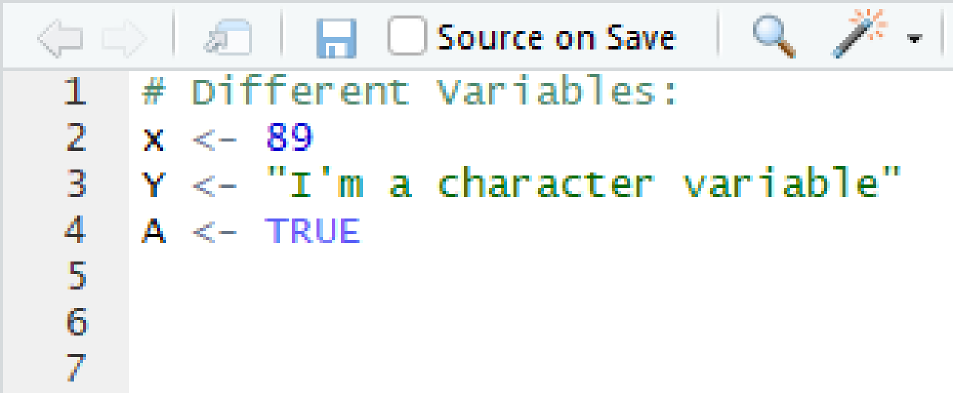
Value

A `tibble` containing the extracted meta-data with the following columns:

- file_name *(chr)*: The file_name of the original .xml-file. Can be used for joining with other parts (authors, references, footnotes, full-texts).
- journal_doi *(chr)*: A registered identifier for the journal.
- journal_jcode *(chr)*: A identifier for the journal like "amerjsoci" for the "American Journal of Sociology".
- journal_pub_id *(chr)*: Similar to journal_jcode. Most of the time either one is present.
- journal_title *(chr)*: The title of the journal.

Variables/Objects

- **Object: Essentially a variable that can store a value**

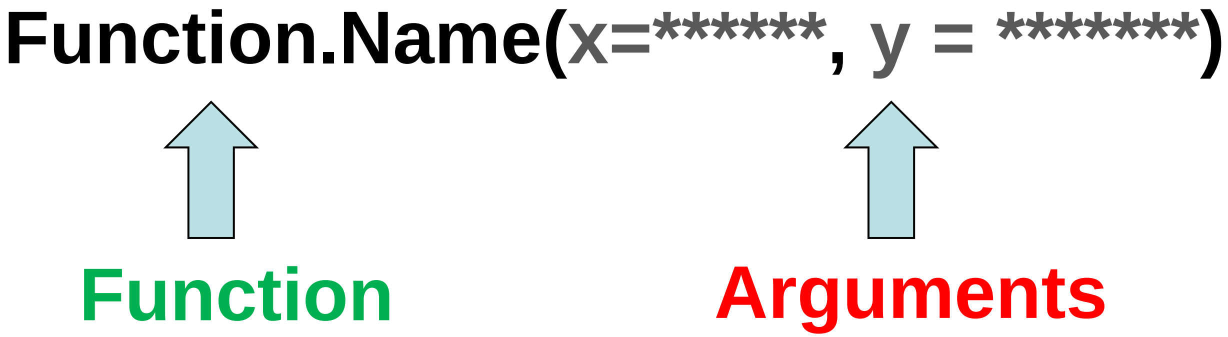


```
# Different Variables:
X <- 89
Y <- "I'm a character variable"
A <- TRUE
```

Data Types

- **Vector**
  - Collection of cells with a fixed size
- **Matrix**
  - Two-dimensional vector (row and column) “mathy”
- **Array**
  - A vector with one or more dimensions
    - *One dimensional array ~ A vector*
    - *Two dimensional array ~ A matrix*
- **List**
  - Can hold objects of different types
- **Data Frame**
  - A Table in which each column holds the same data type

Functions



Function.Name(x=****, y = *******)**

Function

Arguments

```
MyData <- read.csv(“datafile.csv” , Header = TRUE)
```

Object

(Data Frame)

Function

Arguments

Functions: Examples

print(“What we want to print”**)**

mean(A Numerical Object**)**

strsplit(x=A Character Vector, **split=**Where we want to split**)**

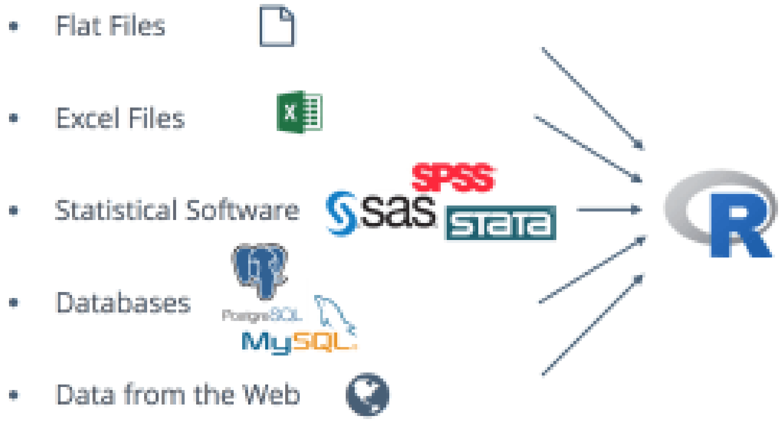
Help Function

For Help with functions:

- **Within R**
  - Type ?read.csv
  - help(read.csv)
  - Check the Help box in bottom-right of RStudio
- **Online**
  - Google!
  - StackOverflow

Reading & Writing Data

- **Different Functions:**
  - read.csv()
  - readLines()
- **Many Different Packages for Different Data**
  - sas7bdat
  - foreign
  - officer

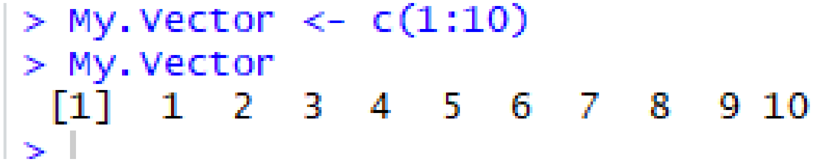


Indexing

- There are a few ways to grab a specific element or column/row of a data type
- We use square brackets [ ] to grab a specific element
- Elements are indexed in numerical order from 1 to n

Indexing

We'll make a vector called My.Vector



```
> My.Vector <- c(1:10)
> My.Vector
 [1]  1  2  3  4  5  6  7  8  9 10
> 
```

We can grab the first element by typing

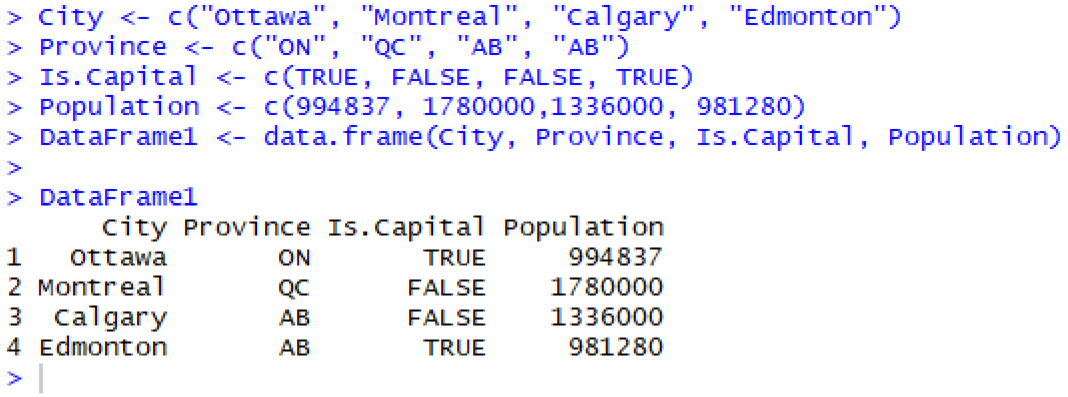
My.Vector[1]

Or the fifth element by typing

My.Vector[5]

Indexing

We'll make a DataFrame called DataFrame1



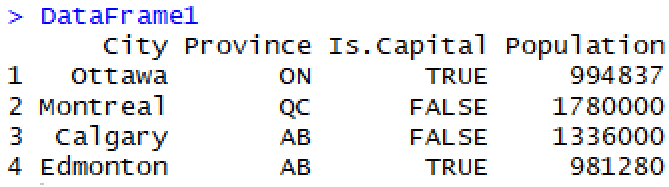
```
> City <- c("Ottawa", "Montreal", "Calgary", "Edmonton")
> Province <- c("ON", "QC", "AB", "AB")
> Is.Capital <- c(TRUE, FALSE, FALSE, TRUE)
> Population <- c(994837, 1780000,1336000, 981280)
> DataFrame1 <- data.frame(City, Province, Is.Capital, Population)
>
> DataFrame1
      City Province Is.Capital Population
1   Ottawa       ON       TRUE     994837
2 Montreal       QC      FALSE    1780000
3  Calgary       AB      FALSE    1336000
4 Edmonton       AB       TRUE     981280
> 
```

In a two-dimensional object, we have to index the row and column

Indexing

To grab a specific element, we type

DataFrame1[Row #, Column #]



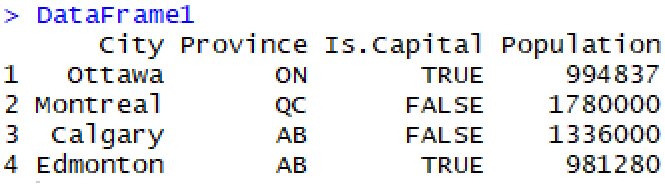
```
> DataFrame1
      City Province Is.Capital Population
1   Ottawa       ON       TRUE     994837
2 Montreal       QC      FALSE    1780000
3  Calgary       AB      FALSE    1336000
4 Edmonton       AB       TRUE     981280
```

So to get “Calgary”, we type

DataFrame1[3,1]

Indexing

To get a entire row:



```
> DataFrame1
      City Province Is.Capital Population
1   Ottawa       ON       TRUE     994837
2 Montreal       QC      FALSE    1780000
3  Calgary       AB      FALSE    1336000
4 Edmonton       AB       TRUE     981280
```

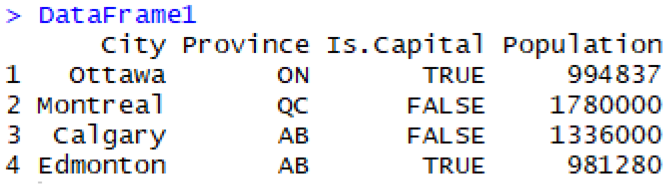
To get the third row, type:

DataFrame1[3,]

(Just the row number, but make sure to add the comma)

Indexing

To get a entire column:



```
> DataFrame1
      City Province Is.Capital Population
1   Ottawa       ON       TRUE     994837
2 Montreal       QC      FALSE    1780000
3  Calgary       AB      FALSE    1336000
4 Edmonton       AB       TRUE     981280
```

To get the third column, type:

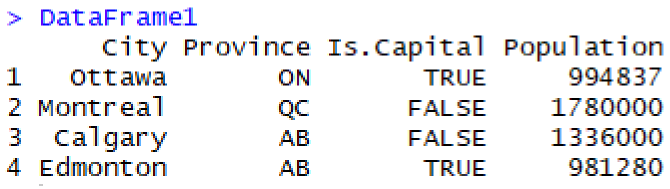
DataFrame1[,3]

(Just the row number, but make sure to add the comma)

Indexing

OR

To get a entire row or column:



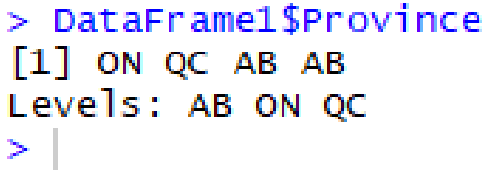
```
> DataFrame1
      City Province Is.Capital Population
1   Ottawa       ON       TRUE     994837
2 Montreal       QC      FALSE    1780000
3  Calgary       AB      FALSE    1336000
4 Edmonton       AB       TRUE     981280
```

Dataframes and Lists, have named objects like columns which can be grabbed by using $

Type

DataFrame$Column.Name

Type DataFrame1$Province will return:



```
> DataFrame1$Province
[1] ON QC AB AB
Levels: AB ON QC
> 
```

Conditional Statements

If Statement is TRUE than do this

We use curly brackets {} to open and close our "if statement"

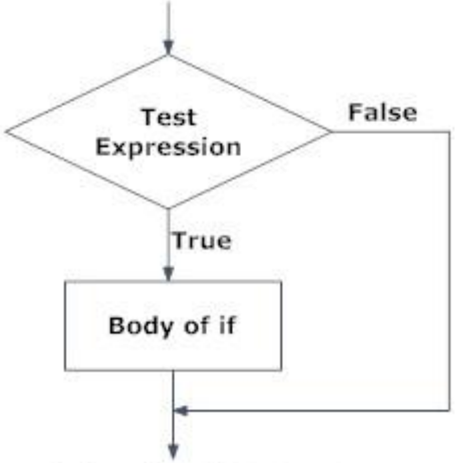


Fig: Operation of if statement

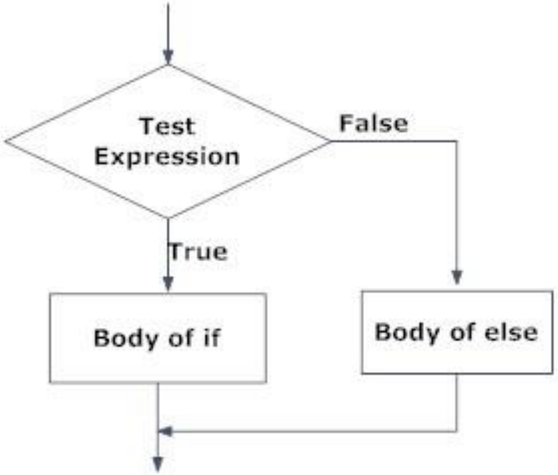


Fig: Operation of if...else statement

Conditional Statements

```
if(Conditional Statement) {
        thing to happen if Condition is TRUE
}


if(Conditional Statement){
        thing to happen if Condition is TRUE
} else {
        thing to happen if Condition is FALSE
}
```

Loops

- **Loops are used to repeat a specific task over a block of code**
- **For Loops**
  - Runs for a specific period of time (10 times, 20 times, 1000 times)
- **While Loops**
  - Runs until a specific condition is met (run until object is greater than 10, run until you encounter a specific object)

For Loops

Looks similar to our if statements

```
for (i in 1:10){
        print(i)
}
```

i is an object

1:10 is a range of numbers

print(i) will be done for each value of i

How to Get Help

- **Within R**
  - Type ?function
  - Check the Help box in bottom-right of RStudio
- **Online**
  - Google!
  - StackOverflow