

## Topic 0: Introduction and R tutorial

8/28/2018

### Administrative

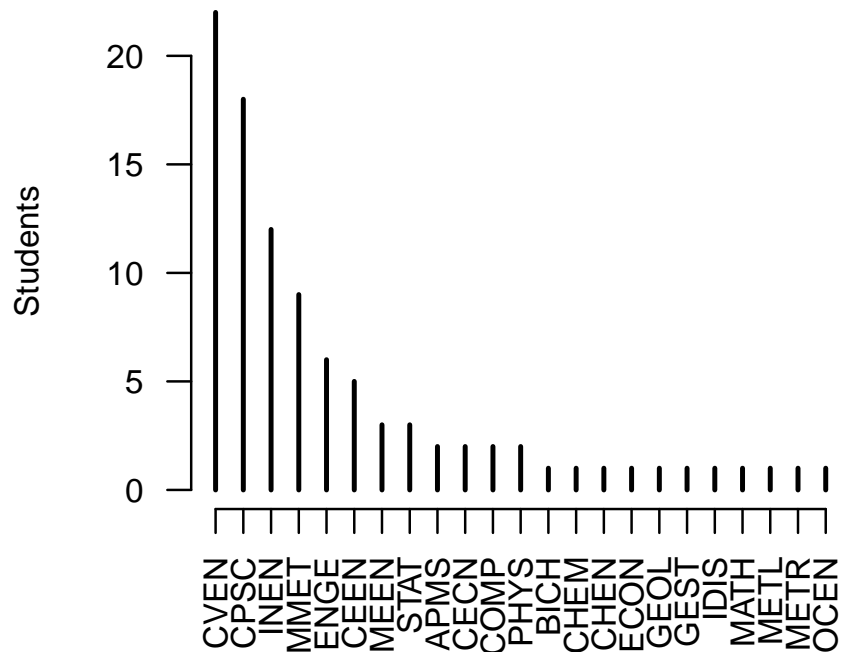
1. Get webassign
2. Accept Piazza invite
3. Read syllabus
4. Download R and Rstudio

### What is Statistics?

Statistics is the science of learning from data, and of measuring, controlling, and communicating uncertainty; and it thereby provides the navigation essential for controlling the course of scientific and societal advances.

– Marie Davidian and Thomas A. Louis, *Why Statistics?*, Science 2012.

### Why should you care



- Computer science/software engineering
  - A/B Testing

- Recommendation
- Civil engineering
  - Traffic management
  - Risk and reliability
- Industrial engineering
  - Statistical process control
  - Queuing theory
- Mechanical engineering
  - Optimal control
- Biology
  - Genome-wide association study
  - Phylogeny
- Meteorology
  - Model output statistics
  - Ensemble forecasts
- Economics
  - Dynamic stochastic general equilibrium models
  - Factor investing
- Chemistry
  - Multivariate calibration
- Humanities
  - Topic modeling
  - Distant reading
- Psychology
  - Personality testing
  - Standardized testing
- Politics
  - Polling
  - Ideal point models
- Marketing
  - Market basket analysis

*Motivating example*

In the 2015 season of the National Football League (NFL), the Houston Texans won 9 of their 16 games.

- Win percentage:  $(9 / 16) \times 100\% = 56.25\%$ .
- Is “real” win percentage better than chance (50%)?
- What is the probability of 9 or more wins out of 16 if real win percentage is 50%?

Simulate result of a season

```
p <- 0.5
season <- sample(c(0, 1), size = 16, replace = TRUE,
  prob = c(1 - p, p))
win_total <- sum(season)
season

## [1] 1 1 0 0 1 0 0 0 0 1 1 1 0 1 0 1

win_total

## [1] 8
```

Run simulation 1000 times

```
n <- 1000
win_total <- replicate(n, {
  season <- sample(c(0, 1), size = 16, replace = TRUE,
    prob = c(1 - p, p))
  sum(season)
})
```

Frequencies of win totals:

```
table(win_total)

## win_total
##  2  3  4  5  6  7  8  9 10 11 12
##  4 12 27 67 113 158 206 184 115 72 31
## 13 14
##  8  3
```

Win probabilities

```
table(win_total)/n
```

```
## win_total
##      2      3      4      5      6      7      8
## 0.004 0.012 0.027 0.067 0.113 0.158 0.206
##      9     10     11     12     13     14
## 0.184 0.115 0.072 0.031 0.008 0.003

sum(win_total >= 9)/n

## [1] 0.413
```

## Learning R

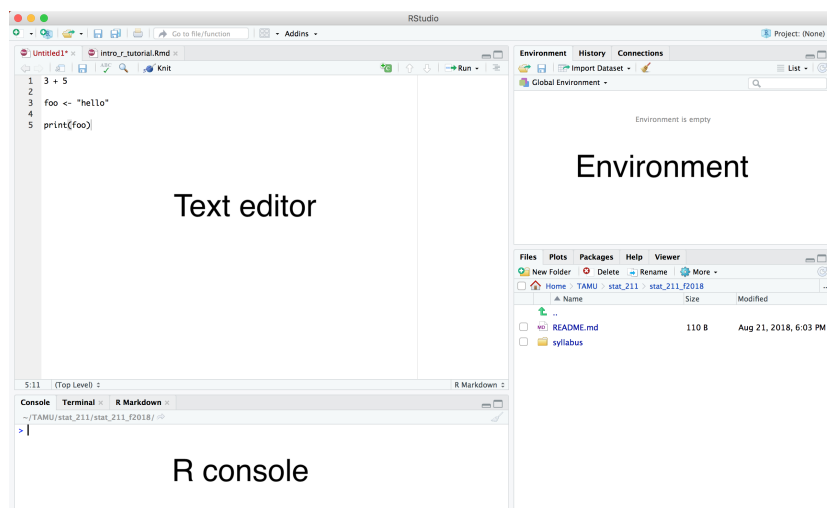
### Resources

- StackOverflow
- DataCamp introduction
- Rstudio cheatsheets

### General Advice

- Just trying something has no cost; guess and check

## Using R



## Suggested workflow

1. Open Rstudio
2. Change working directory: `Ctrl + Shift + h`
3. Create script: `Ctrl + Shift + n`

4. Write, save code in script
5. Run code
  1. source the script, or
  2. run line by line

## *R Syntax*

```
# this is a comment

# use R as a calculator
3 * (5 + sqrt(2) + pi)

## [1] 28.66742

# assignment
a <- TRUE
b = 2

# comparison
10 > 20

## [1] FALSE

is.na(NA) & (5 > b)

## [1] TRUE
```

## Control Flow

```
# conditional
if (!a) {
  print("hello")
} else {
  print("goodbye")
}

## [1] "goodbye"

# for loop
for (i in 1:10) {
  cat(i)
}

## 12345678910
```

```
# while loop
x <- 4
while (x > 0) {
  cat(x^2)
  cat(" ")
  x <- x - 1
}

## 16 9 4 1
```

### Data Types

```
# vectors
vec1 <- c(1, 5, 4, 3)
vec2 <- 1:10
vec3 <- seq(from = -4, to = 2, by = 2)

# get first element
vec1[1]

## [1] 1

# change 2nd element value
vec1[2] <- 1000
vec1

## [1] 1 1000 4 3

# get length
length(vec3)

## [1] 4

# lists
list1 <- list(1, "a", 3)
list1

## [[1]]
## [1] 1
##
## [[2]]
## [1] "a"
##
## [[3]]
## [1] 3

# get first element, not a list
list1[[1]]
```

```
## [1] 1

# get sublist, this is a list
list1[1]

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

# data frames
names <- c("Bob", "Fatima", "Pierre")
df <- data.frame(age = c(10, 15, 23), name = names)
df

##   age  name
## 1  10   Bob
## 2  15 Fatima
## 3  23 Pierre

# get a column, 3 ways to do same thing
df[, "name"]
df$name
df[, 2]

## [1] Bob    Fatima Pierre
## Levels: Bob Fatima Pierre
## [1] Bob    Fatima Pierre
## Levels: Bob Fatima Pierre
## [1] Bob    Fatima Pierre
## Levels: Bob Fatima Pierre

colnames(df)
dim(df)

## [1] "age"  "name"
## [1] 3 2
```

Reading/writing data sets

```
# write df to csv, look at directory contents
write.csv(df, "demo_file.csv", row.names = FALSE)
dir()

## [1] "demo_file.csv"
## [2] "intro_r_tutorial_files"
## [3] "intro_r_tutorial.aux"
## [4] "intro_r_tutorial.html"
## [5] "intro_r_tutorial.out"
## [6] "intro_r_tutorial.pdf"
```

```
## [7] "intro_r_tutorial.Rmd"
## [8] "roster_509_f2018.csv"
## [9] "rstudio_pic.png"
## [10] "tryingstuffuntilitworks-big.png"
```

```
# read df back in
```

```
df2 <- read.csv("demo_file.csv")
df
```

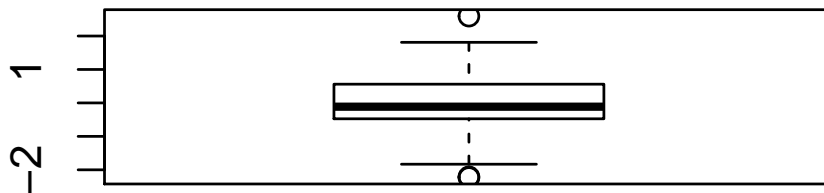
```
##   age  name
## 1  10   Bob
## 2  15 Fatima
## 3  23 Pierre
```

```
df2
```

```
##   age  name
## 1  10   Bob
## 2  15 Fatima
## 3  23 Pierre
```

### Plotting

```
y_vals <- rnorm(100)
boxplot(y_vals)
```



### Functions

```
# define function
hello_func <- function(name, response = "hello") {
  paste0(name, " says ", response)
}
```

```
# call function
hello_func("Patrick")
```

```
## [1] "Patrick says hello"
```

```
# see function
hello_func
```



```
## function(name, response = "hello") {
##   paste0(name, " says ", response)
## }
```

*# override default argument*

```
hello_func("Patrick", response = "goodbye")
```

```
## [1] "Patrick says goodbye"
```

Getting help

Use ? or help()

```
'?'(hist)
```

```
help(read.csv)
```

Packages

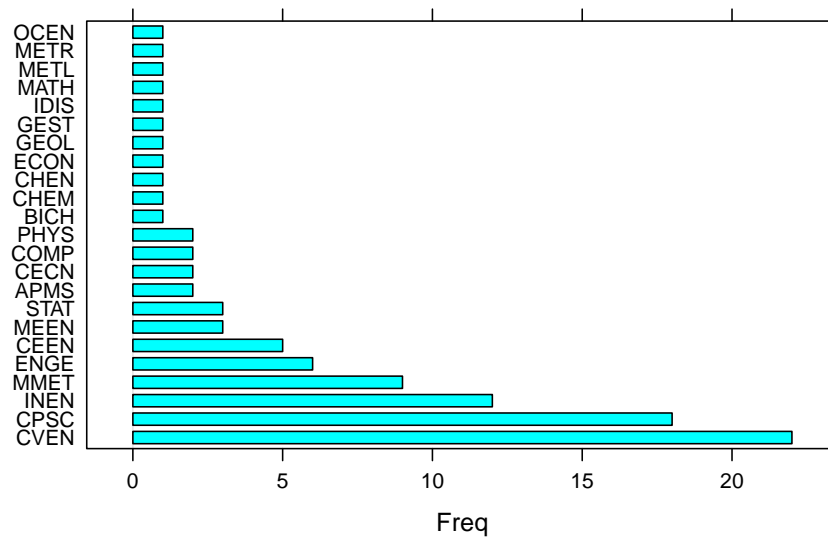
Stand on the shoulders of giants

*# download a package, lattice*

*# install.packages(lattice)*

*# call a function from a package*

```
lattice::barchart(roster_major)
```



*# load packages into environment, call*

*# function directly*

```
library(lattice)
```

```
dotplot(roster_major)
```

