

STA 138 FQ 2020 Discussion 1

Yidong Zhou

09/30/2020

Discussion Sessions and Office Hours

TA: Yidong Zhou (ydzhou@ucdavis.edu)

We will use Piazza (access code “overdispersion” without the quotes) for this course to answer questions and facilitate discussion. Ask routine questions about course material and policies there; direct any sensitive or specifically personal matters to me or the instructor by email (**email please, not Canvas message**). In any emails, please include the text “[**STA138-FALL2020**]” (without the quotes) in the subject line.

Discussion sessions:

- Thursdays 09:00-09:50 am PT: <https://ucdstats.zoom.us/j/98190589653?pwd=K1lVTElLL1V0VVBNeMd3RklxV1dXZz09>
- Thursdays 10:00-10:50 am PT: <https://ucdstats.zoom.us/j/99471664794?pwd=eCtzRWtVeEFhQmlSd3hyL1B6b2w2Zz09>

Office hours:

- Tuesdays 09:00-11:00 am PT: <https://ucdstats.zoom.us/j/99625197827?pwd=OTFJY2FSZmozWVppMm1CSm82cVhTUT09>

All the above zoom links can be found in Canvas -> Syllabus.

Review

- Installation of R and RStudio: <https://www.datacamp.com/community/tutorials/installing-R-windows-mac-ubuntu>
- Introduction to RStudio, including file creation, panel structure.
- R basics, including variable creation, arithmetic calculation, vectors, and some basic functions: `help()`, `?`, `rep`, `seq`.

Data Manipulation

We will learn how to perform data manipulation in R programming language along with data processing. We will also overview the three operators such as subsetting, manipulation as well as sorting in R. Also, we will learn about data structures in R, how to create subsets in R and usage of R `sample()` command, ways to visualize data in R.

With the help of data structures, we can represent data in the form of data analytics. Data Manipulation in R can be carried out for further analysis and visualization. The first step is to figure out how to import data in R.

Environment panel -> Import Dataset -> From Text(readr)... -> browse -> Import

Secondly, you need to be familiar with basic **data structures** in R (`class` function):

- **Vectors**: ordered containers of primitive elements and are used for 1-dimensional data.

- **Matrices:** rectangular collections of elements and are useful when all data is of a single class that is numeric or characters.
- **Lists:** ordered containers for arbitrary elements and are used for higher dimension data, like customer data information of an organization. When data cannot be represented as an array or a data frame, the list is the best choice. This is because lists can contain all kinds of other objects, including other lists or data frames, and in that sense, they are very flexible.
- **Data Frames:** two-dimensional containers for records and variables and are used for representing data from spreadsheets etc. It is similar to a single table in the database.

(**data types:** integer, numeric, logical, character, complex.)

subset data

The process of creating samples is called subsetting. Different methods of subsetting in R are:

- **\$:** The dollar sign operator selects a single element of data. The result of this operator is always a vector when we use it with a **data frame**.
- **[[:** Similar to **\$** in R, the double square brackets operator in R also returns a single element, but it offers the flexibility of referring to the elements by position rather than by name. It can be used for data frames and lists.
- **[:** The single square bracket operator in R returns multiple elements of data. The index within the square brackets can be a numeric vector, a logical vector, or a character vector.

For example: To retrieve 5 rows and all columns of already built-in dataset **mtcars**, the below command, is used:

```
data(mtcars)
mtcars$mpg# column called mpg

## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

```
mtcars[[1]]# first column

## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

```
mtcars[1:5, ]# first five rows

##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## 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
```

```
mtcars[mtcars$cyl==8, ]# rows with 8 cylinders

##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## 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
```

sample() command in R For example, to create a sample of 10 simulations of a die, below command is used:

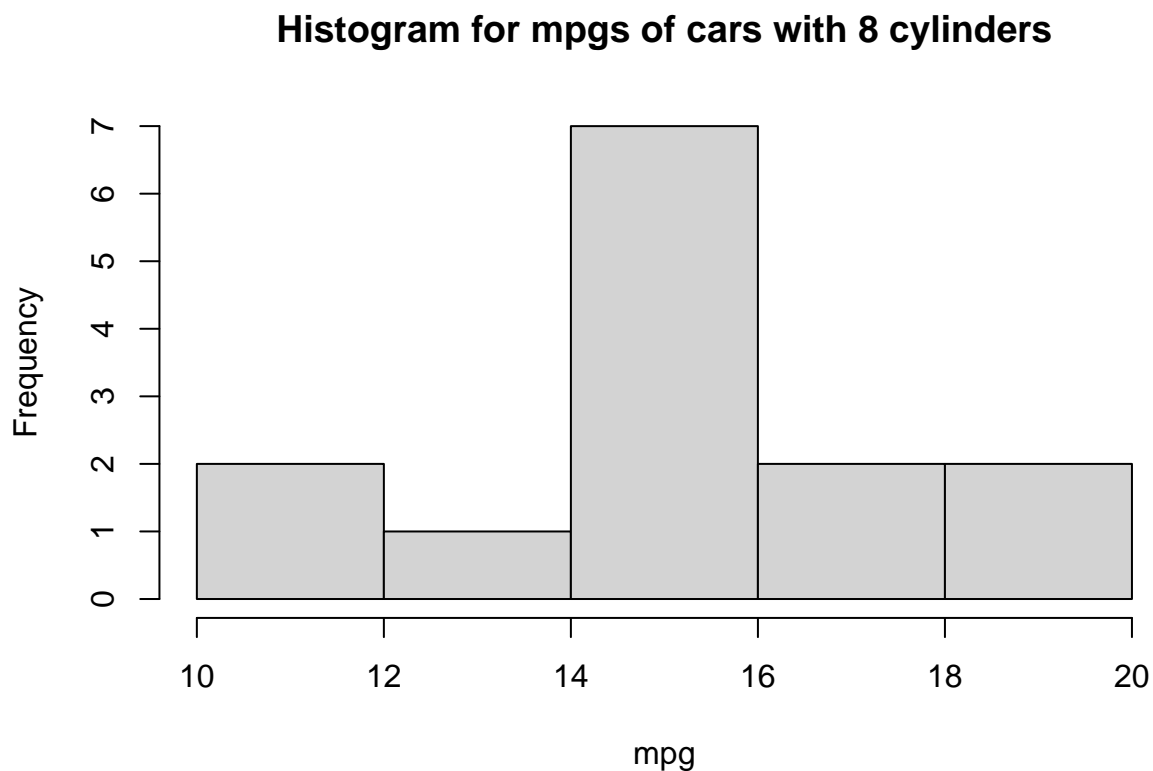
```
sample(1:6, 10, replace=TRUE)
```

```
## [1] 5 2 1 2 4 4 2 2 3 3
```

draw histograms

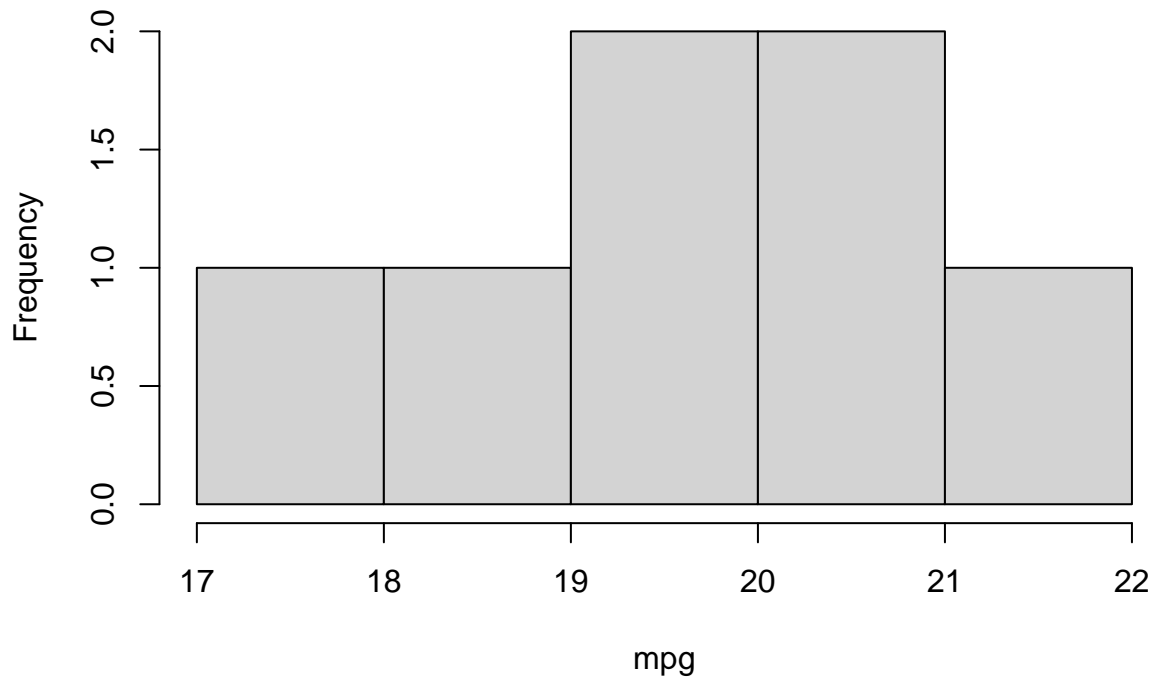
Histogram can be created using the `hist()` function in R programming language. This function takes in a vector of values for which the histogram is plotted.

```
hist(mtcars[mtcars$cyl==8, ]$mpg, main='Histogram for mpgs of cars with 8 cylinders', xlab='mpg')
```



```
hist(mtcars[mtcars$cyl==6, ]$mpg, main='Histogram for mpgs of cars with 6 cylinders', xlab='mpg')
```

Histogram for mpgs of cars with 6 cylinders



compute means and medians

```
mean(mtcars[mtcars$cyl==8, ]$mpg)
```

```
## [1] 15.1
```

```
mean(mtcars[mtcars$cyl==6, ]$mpg)
```

```
## [1] 19.74286
```

```
median(mtcars[mtcars$cyl==8, ]$mpg)
```

```
## [1] 15.2
```

```
median(mtcars[mtcars$cyl==6, ]$mpg)
```

```
## [1] 19.7
```

```
var(mtcars[mtcars$cyl==8, ]$mpg)
```

```
## [1] 6.553846
```

```
sd(mtcars[mtcars$cyl==8, ]$mpg)
```

```
## [1] 2.560048
```

```
summary(mtcars)
```

```
##      mpg      cyl      disp      hp  
##  Min.   :10.40  Min.   :4.000  Min.   : 71.1  Min.   : 52.0
```

```
## 1st Qu.:15.43 1st Qu.:4.000 1st Qu.:120.8 1st Qu.: 96.5
## Median :19.20 Median :6.000 Median :196.3 Median :123.0
## Mean :20.09 Mean :6.188 Mean :230.7 Mean :146.7
## 3rd Qu.:22.80 3rd Qu.:8.000 3rd Qu.:326.0 3rd Qu.:180.0
## Max. :33.90 Max. :8.000 Max. :472.0 Max. :335.0
##      drat      wt      qsec      vs
## Min. :2.760 Min. :1.513 Min. :14.50 Min. :0.0000
## 1st Qu.:3.080 1st Qu.:2.581 1st Qu.:16.89 1st Qu.:0.0000
## Median :3.695 Median :3.325 Median :17.71 Median :0.0000
## Mean :3.597 Mean :3.217 Mean :17.85 Mean :0.4375
## 3rd Qu.:3.920 3rd Qu.:3.610 3rd Qu.:18.90 3rd Qu.:1.0000
## Max. :4.930 Max. :5.424 Max. :22.90 Max. :1.0000
##      am      gear      carb
## Min. :0.0000 Min. :3.000 Min. :1.000
## 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.:2.000
## Median :0.0000 Median :4.000 Median :2.000
## Mean :0.4062 Mean :3.688 Mean :2.812
## 3rd Qu.:1.0000 3rd Qu.:4.000 3rd Qu.:4.000
## Max. :1.0000 Max. :5.000 Max. :8.000
```

draw mosaic plots

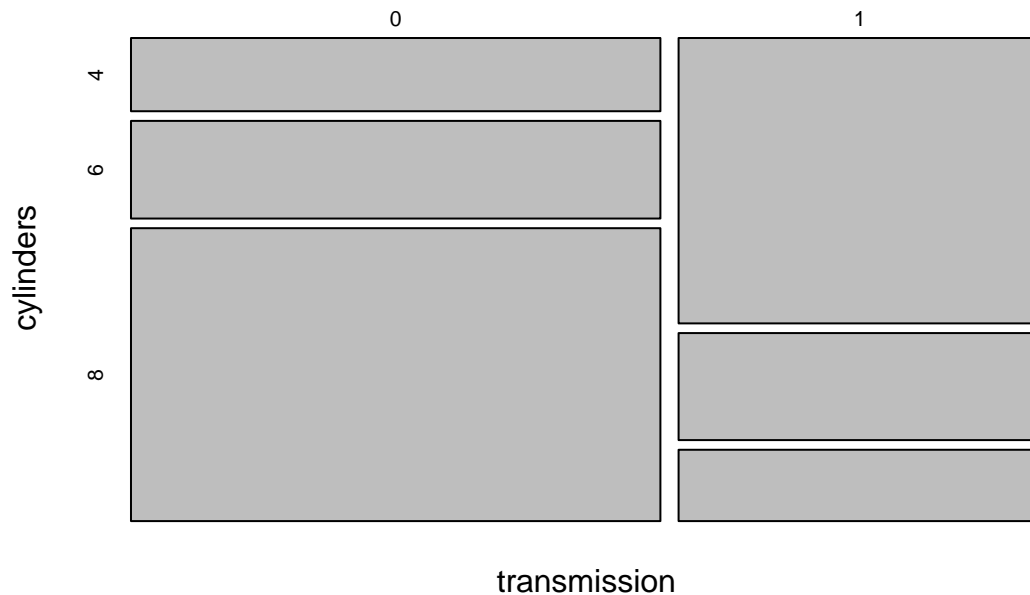
The Mosaic Plot in R Programming is very useful to visualize the data from the contingency table or two-way frequency table. The R Mosaic Plot draws a rectangle, and its height represents the proportional value.

```
table(mtcars$am, mtcars$cyl) # contingency table or two-way frequency table
```

```
##
##      4  6  8
## 0  3  4 12
## 1  8  3  2
```

```
mosaicplot(mtcars$am~mtcars$cyl, main='Mosaic plot', xlab='transmission', ylab='cylinders')
```

Mosaic plot



Code Appendix

```
data(mtcars)
mtcars$mpg# column called mpg
mtcars[[1]]# first column
mtcars[1:5, ]# first five rows
mtcars[mtcars$cyl==8, ]# rows with 8 cylinders
sample(1:6, 10, replace=TRUE)
hist(mtcars[mtcars$cyl==8, ]$mpg, main='Histogram for mpgs of cars with 8 cylinders', xlab='mpg')
hist(mtcars[mtcars$cyl==6, ]$mpg, main='Histogram for mpgs of cars with 6 cylinders', xlab='mpg')
mean(mtcars[mtcars$cyl==8, ]$mpg)
mean(mtcars[mtcars$cyl==6, ]$mpg)
median(mtcars[mtcars$cyl==8, ]$mpg)
median(mtcars[mtcars$cyl==6, ]$mpg)
var(mtcars[mtcars$cyl==8, ]$mpg)
sd(mtcars[mtcars$cyl==8, ]$mpg)
summary(mtcars)
table(mtcars$am, mtcars$cyl)# contingency table or two-way frequency table
mosaicplot(mtcars$am~mtcars$cyl, main='Mosaic plot', xlab='transmission', ylab='cylinders')
```