NDH802 R Application - Live session 1

Huong Nguyen

First thing first

Set your working directory

. . . so you (and your files) don't get lost later. To do that, you either type

```
setwd("/Users/hu.4622/Documents/GitHub/NDH802") #This is my WD, change it to yours!
```

with the path to the folder you want to store your working files (e.g., R script, data tables, results). You can also do that by clicking Session/Set Working Directory. More on this in the live session.

Try your code!

Make sure your code chunk works before you go to the next chunk, or else you don't know since when it starts to go wrong and the diagnostic will be more challenging. Simply run your code every time you write!

Clean your code!

As a result of trying different code, your script can get messy and unreadable. Make some time to clean your code, i.e., delete the redundant and write comments on what you keep. You may not believe it now, but you will forget. Therefore, note down your thought process! Comment! By the way, you can select a code chunk then hit Shift + Ctrl (Cmd) + a to make it look neat(er)

Save your work!

Continuously hit Ctrl (or Cmd) + S. Your future selves will thank you later! In case you forget (though I hope you don't), everything you've run (**not** typed) are stored in History. One more reason to run your code.

It's perfectly fine to copy code. . .

. . . from the demos, books, articles, internet, but **not** your friends for the assignments. Make sure you understand the code you copy, or only copy the code you can make sense. Sometimes we simply want to make it work, but it's more important to understand why and/or when it works. Or else, you won't remember it or worse, you may misuse it the next time.

When you don't know something,

- First thing you can try is to ask R by typing in the console `?thecommand`, e.g., `?summary`. R Studio will show the command's description and usage in Help windows on your bottom right.
- If you don't find that helpful, ask Google. For example, "summary() in R". You'll find plenty of explanations in various ways (e.g., R documentation, blog post, video). I hope you find at least one of them helpful. When you learn programming, people from Stack Overflow are your best friends.
- Reach out! Ask me, Emelie, your friends, anyone you think may have or can find the answers.

You're here to learn statistics. R is supposed to be a helpful tool, don't let it be your obstacle. Having said that, it's perfectly fine to get stuck sometimes. Everyone does, even the experienced programmers. So, no worries, keep calm and code on!

Practice makes perfect

Unfortunately, there is no way to learn R (or any programming languages) without trying, failing, fixing up, learning from your mistakes, repeat. I hope I can make it less frustrating for you. To do that, I strongly encourage you to do the preparations before the live/Q&A sessions and ask if you have questions. Also, it would be nice that you try the assignments right after the live sessions when your memories are still fresh.

Working with data in R

Load your data

If you are familiar with Excel, this step is like opening a file. You can type in

```
salaries = read.csv("https://bit.ly/3r918BW") #data stored on cloud
```

or click on Import Data set on the top right of your screen.

Explore your data

These commands give you a quick overview of your data. I recommend you to try and run them in Rstudio, but NOT to knit them. It's gonna be a bit messy. To mask a command, we make them a "comment" by placing # in front of the command. Things after # are not code and won't be executed.

```
#View(salaries)
head(salaries)
tail(salaries)
summary(salaries)
#rename a column
```

Select columns from a data frame

There are numerous way to select a column. The three most basic way are:

```
#here you refer to the column by the columns names
salaries$sal_expected
salaries[,"sal_expected"]

#here you refer to the column by the column position from the left ie 1 is the first column from the le
salaries[,1]
```

When to use which?
It's all about your personal preference. I introduce several ways to give you options, **not** to confuse you. You can totally explore and see which you're more comfortable with. This is also applicable for most of other commands.

To select more than 1 column:

```
#here you refer to the column by the columns names
salaries[,c("sal_expected", "fairpay", "program")]

#here you refer to the column by the column position from the left
salaries[,c(1,3,10)]
```

Select rows from a data frame

There are numerous way to select a row. The most basic way (with base R) are:

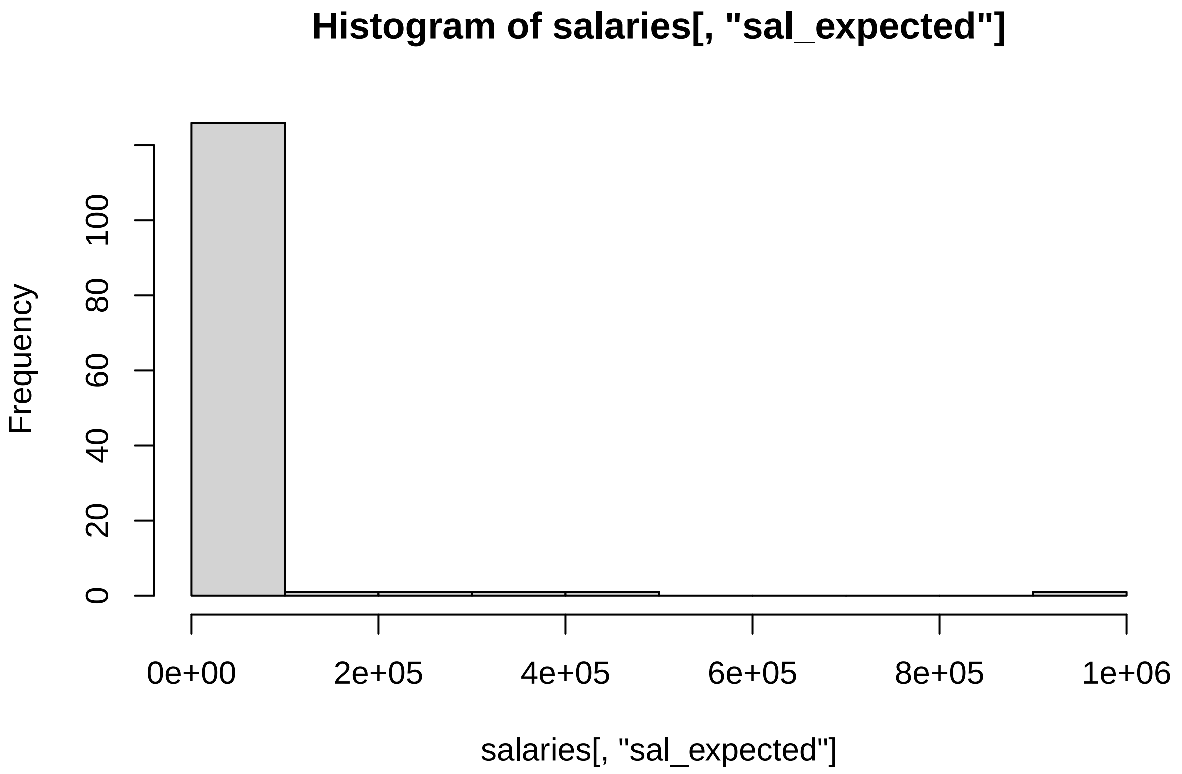
```
salaries[salaries$program == "RM",]
salaries[salaries$whatisyourgender1isfemale == 1,]
```

Let's take a closer look

Assume you want to explore the salary expectation from the students, which is the variable `sal_expected`. Two of my most frequent used methods are histogram and box plot.

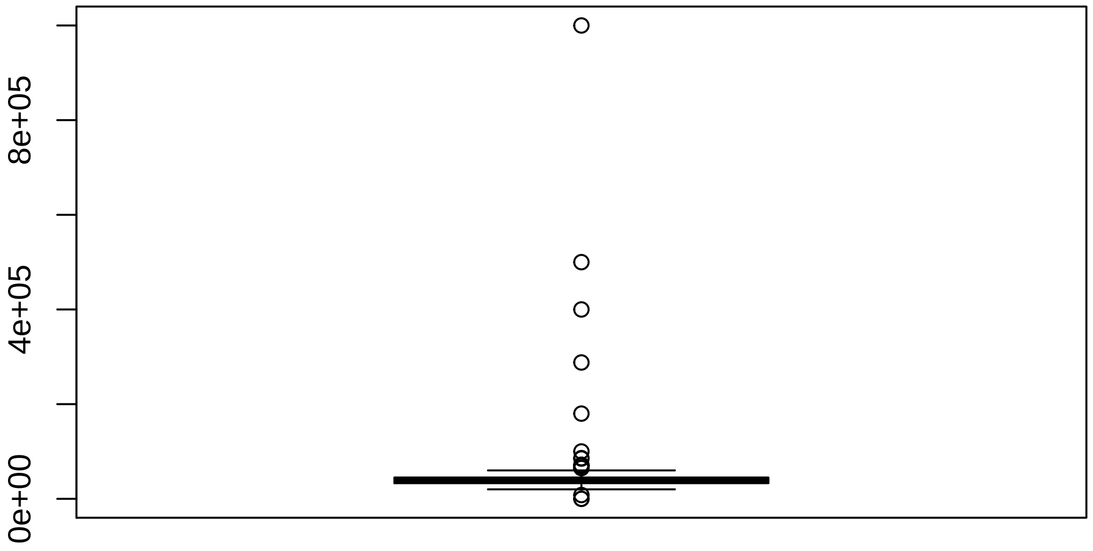
Let's first try plotting a **histogram**. If you're like "a histo what?", take a quick look here, then come back.

```
hist(salaries[,"sal_expected"]) #this in the assignment will get 0.5pt
```



This looks neither aesthetically appealing nor informative, right? How do you think we can make it better?

Now let's try **box plot**. Similarly, if you need a quick understanding of box plot, take a look here, then come back.



This looks nothing like the video. Why do you think it is and can you help me fix it?

Means and standard deviation (SD)

Calculate the mean and SD of a variable

Intuitively, you would type

```
mean(salaries[,"sal_expected"])
```

```
## [1] NA
```

```
sd(salaries[,"sal_expected"])
```

```
## [1] NA
```

but it doesn't work. why?

```
#mean(salaries[,"sal_expected"], na.rm = TRUE)
#sd(salaries[,"sal_expected"], na.rm = TRUE)
```

Can I calculate the mean of salary expected of the RM, and the mean of salary expected of the BE students separately?

```
#mean(salaries[salaries$program == "BE","sal_expected"], na.rm = TRUE)
#mean(salaries[salaries$program == "RM","sal_expected"], na.rm = TRUE)
```

Can I calculate the mean of salary expected of the female RM student?

```
#mean(salaries[salaries$program == "BE" & salaries$whatisyourgender1isfemale == "1", "sal_expected"], n
```

Probability theory

Let A be an event. To calculate N_A in base R, the simplest way is to use `table()`. For example, assume we want to know the number of BE (N_{BE}) and RM (N_{RM}) students in our data, we can

```
table(salaries[,"program"])
```

```
##
## BE RM
## 95 38
```

Similarly, let's see how many male (N_{male}) and female (N_{female}) students we have in the data

```
table(salaries[,"whatisyourgender1isfemale"])
```

```
##
##  1  2
## 44 81
```

We can do something similar to a pivot table with `table()`

```
table(salaries[,"program"], salaries[,"whatisyourgender1isfemale"])
```

```
##
##       1  2
##   BE 23 68
##   RM 21 13
```

From the table above, can you tell

- how many **RM female students** we have in the data?

- if RM and BE are **mutually exclusive/collectively exhaustive** events?

Your turn

(1) Plot a histogram for `worktime`.
(2) Make a box plot for `idealpay`.
(3) Make a box plot for `idealpay`, for RM and BE students separately.
(4) Compute the mean and variance of `fairpay`.
(5) Compute the mean and variance of `fairpay` of the female students.
(6) Compute the mean and variance of `fairpay` of the students who are female and belong to RM program.
(7) How many male students and female students do we have in the data set? Formally, compute N_{male} and N_{female}
(8) How many *students who are female and belong to RM program* do we have in the data set? compute $N_{female \cap RM}$