Final Study Guide

Stat 133, Spring 2018, Prof. Sanchez

1) What function do you use to load a package?

2) What command do you run to see the manual documentation of the function `"read.table"`?

3) Match the provided descriptions with the following commands.

a. `load(ggplot2)`
b. `help(ggplot)`
c. `install.packages("ggplot2")`
d. `g <- ggplot()`
e. `library(ggplot2)`
f. `man(ggplot)`
g. `obj <- ggplot2()`

- Creates an object of class `ggplot`.
- Downloads the package `ggplot2` from CRAN.
- Loads the package `ggplot2` to the current session.
- Displays the manual documentation of `ggplot`.

4) Write down a code example of the recycling rule in R.

5) What is the data type of each of the following vectors:

- `x`: where `x <- c(TRUE, FALSE)`
- `y`: where `y <- c(x, 10)`
- `z`: where `z <- c(y, 10, "a")`

6) In R, what is the typical symbol to represent missing values?

7) What are the different (data) types of missing values that R provides?

8) What are the main differences between R matrices and R data frames?

9) Which of the following commands reloads R objects from an R binary file `z.RData`:

- `library(z.RData)`
- `download.file("z.RData")`
- `load("z.RData")`
- `knit("z.RData")`

10) Write down the command to check if a variable `x` is greater than 5? (HINT: This command should return a `TRUE/FALSE` value)

11) Suppose `y <- c(1, 4, 9, 16, 25)`. Write down the R command to return a vector `z`, in which each element of `z` is the square root of each element of the vector `y`.

12) Write down 2 different R commands to return the first five elements of a vector `x` (assume `x` has more than 5 elements).

13) Consider a data.frame object `df`. You can use vectors of indices `index1` and `index2` to subscript the data frame like this: `df[index1, index2]`. Which of the following are NOT valid options for indexing (i.e. subscripting) `df`:

- Numeric vectors
- Missing values
- Logical vectors
- Blank spaces

14) What is the output of this command:

```
c(1, 2, 3, 4, 5) * 2
```

15) What is the output of this command:

```
1:3^2
```

16) What is the output of this command:

```
(1:5)*2
```

17) What is the output of this command:

```
var<-3
Var*2
```

18) What is the output of this command:

```
x<-2
2x<-2*x
```

19) What is the output of this command:

```
sqrt4 <- sqrt(4)
sqrt4
```

20) What is the output of this command:

```
a number <- 16
```

21) Why the following comparisons return `TRUE`:

```
1 == TRUE
```

```
## [1] TRUE
```

```
0 == FALSE
```

```
## [1] TRUE
```

22) How do you use the function `seq()` to create the following vector?

```
 [1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0
```

23) Give an example using the function `rep()` to create the following vector:

```
[1] 1 1 2 2 3 3
```

24) Give an example using the function `rep()` to create the following vector:

```
[1] 1 2 3 1 2 3
```

25) Give an example of R code using the function `matrix()` that will give you the following output:

```
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
```

26) Give an example of R code using the function `matrix()` that will give you the following output:

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
```

27) You can use the colon operator `":"` to generate a sequence of numbers. For instance `1:3`. How can you get the help documentation for the colon operator?

28) Every time you quit R, a message pops-up with the following question: `Save workspace image to /.RData?`.

- What is the so called *workspace image*?
- What type of file is `.RData`?
- What happens if you choose the `Save` option?

29) In RStudio, one of the panes has the tabs "Environment, History". What is the content of the "History" tab?

30) In RStudio, one of the panes has the tabs "Files, Plots, Packages, Help, Viewer". If you click on the "Files" tab you will see the files of your home directory. There, you should be able to see a file called `.Rhistory`. What does this file contain?

31) When you start a new R session, a message with similar content to the text below appears on the console:

```
R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (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.
```

- What happens when you type: `license()`?
- What happens when you type: `contributors()`?
- What happens when you type: `citation()`?
- What happens when you type: `demo()`?

32) When knitting a file, what happens to those code chunks that use the option `echo = FALSE`?

33) When knitting a file, what happens to those code chunks that use the option `eval = FALSE`?

34) When knitting a file, what happens to those code chunks that use the option `results = 'hide'`?

35) When knitting a file, what happens to those code chunks that use the option `comment = ""`?

36) A code chunk in an `.Rmd` file can be assigned a unique name by specifying a label inside the curly braces, like this: `{r some-label}`. When knitting a file, what happens if you have two different code chunks with the same label?

37) What does the function `getwd()` do?

38) What does the function `setwd()` do?

39) Say you have a data table in a CSV file supposedly called `dataset.csv`. When trying to read the file with `read.csv()` from your working directory you get the following error message:

```
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file 'dataset.csv': No such file or directory
```

- What does the error indicate?
- What could be causing the error?
- How could you try to solve the error?

40) The reading-table functions use the following parameters: `header`, `sep`, `dec`, `row.names`, `colClasses`, `stringsAsFactors`. Explain what they do, and give an example of the value that each of these parameters can take.

41) What does `read.table()` and friends—e.g. `read.csv()`, `read.delim()`—do by default to columns with characters?

42) Explain the concept of *vectorization* a.k.a. vectorized operations.

43) Explain the concept of atomic structures in R.

44) The following question was posted on piazza in a previous edition of Stat 133. What would you answer to the student:

> read_csv() was working before, but now I am trying to restart my homework 2 and it says “Error: could not find function ‘read_csv’. I installed the package readr and I am on the right directory. How do I get out of read_csv error?

45) Explain the use of brackets `[ ]` and parentheses `( )` in R.

46) Consider a data frame `df` with two columns `x` and `y`. What happens if you try to extract a column that is not in `df`, e.g. `df$z`?

47) Name three functions for inspecting the contents of a data frame.

48) Which command will fail to return the first five elements of a vector `x`? (assume `x` has more than 5 elements).

a. `x[1:5]`
b. `x[c(1,2,3,4,5)]`
c. `head(x, n = 5)`
d. `x[seq(1, 5)]`
e. `x(1:5)`

Consider the following data frame `df`:

```
     first     last gender born        spell
1    Harry   Potter   male 1980 sectumsempra
2 Hermione  Granger female 1979    alohomora
3      Ron  Weasley   male 1980   riddikulus
4     Luna Lovegood female 1981      episkey
```

49) Refer to the data frame `df`. What commands will fail to return the data of individuals born in 1980?

a. `df[c(TRUE, FALSE, TRUE, FALSE), ]`
b. `df[df[,4] == 1980, ]`
c. `df[df$born == 1980]`
d. `df[df$born == 1980, ]`
e. `df[ ,df$born == 1980]`

50) Refer to the data frame `df`. Your friend is trying to display the first three rows on columns 1 (`first`) and 2 (`last`), by unsuccessfully using the command: `df[1:3, 1 & 2]`

```
df[1:3, 1 & 2]
```

```
##      first    last gender born        spell
## 1    Harry  Potter   male 1980 sectumsempra
## 2 Hermione Granger female 1979    alohomora
## 3      Ron Weasley   male 1980   riddikulus
```

a. Why does the command above print all columns?

b. Write a command that would correctly display the first two columns.

51) Refer to the data frame `df`. Write a command that would give you the following data from `df`:

```
         spell    first
1 sectumsempra    Harry
2    alohomora Hermione
3   riddikulus      Ron
4      episkey     Luna
```

52) Refer to the data frame `df`. Select the command that does NOT provide you information about the data frame `df`:

a) `head(df)`
b) `str(df)`
c) `tail(df)`
d) `rm(df)`
e) `summary(df)`

Consider the following tibble `sw`:

```
# star wars data frame
sw
```

```
# A tibble: 4 x 4
  name   gender height weight
  <chr>  <chr>   <dbl>  <int>
1 Anakin male     1.88     84
2 Padme  female   1.65     45
3 Luke   male     1.72     77
4 Leia   female   1.50     49
```

53) Refer to the data frame `sw`. Using `"dplyr"`, which of the following commands gives you the data of female individuals:

```
## # A tibble: 2 x 4
##   name  gender height weight
##   <chr> <chr>   <dbl>  <int>
## 1 Padme female   1.65     45
## 2 Leia  female   1.50     49
```

a) `filter(sw, gender == female)`
b) `select(sw, gender == 'female')`
c) `select(sw, gender == 'female')`
d) `filter(sw, gender == 'female')`

54) Refer to the data frame `sw`. Using the pipe operator `"%>%"` in `"dplyr"`, which of the following commands gives you the data of male individuals:

```
## # A tibble: 2 x 4
##   name   gender height weight
##   <chr>  <chr>   <dbl>  <int>
## 1 Anakin male     1.88     84
## 2 Luke   male     1.72     77
```

a) `sw %>% select(gender == 'male')`

b) `sw %>% group_by(gender == 'male')`
c) `sw %>% filter(gender == 'male')`
d) `sw %>% filter(by == 'male')`

55) Refer to the data frame `sw`. Using `"dplyr"`, which of the following commands would give you `name` and `height` arranged by `height` as follows:

```
## # A tibble: 4 x 2
##   name   height
##   <chr>   <dbl>
## 1 Leia     1.50
## 2 Padme    1.65
## 3 Luke     1.72
## 4 Anakin   1.88
```

a) `sw %>% arrange(height) %>% select(name, height)`
b) `sw %>% select(name, height) %>% arrange(height)`
c) `sw %>% select(name, height) %>% arrange(desc(height))`
d) `sw %>% filter(name, height) %>% arrange(height)`

56) Refer to the data frame `sw`. Using the pipe operator `"%>%"` in `"dplyr"`, which of the following commands gives you average height:

```
## # A tibble: 1 x 1
##   avg_height
##        <dbl>
## 1       1.69
```

a) `sw %>% select(height) %>% summarise(avg_height = mean(height))`
b) `sw %>% select(height) %>% avg_height = mean(height)`
c) `sw %>% select(height) %>% summarise(avg_height = mean(weight))`
d) `sw %>% select(height) %>% summarise(avg_height = median(height))`

57) Write a dplyr command that gives the average weight by gender:

```
## # A tibble: 2 x 2
##   gender avg_weight
##   <chr>       <dbl>
## 1 female       47.0
## 2 male         80.5
```

58) A student is trying to implement the following formula in R:

$$e^{\frac{-(X-\mu)^2}{2\sigma^2}}$$

However, the student gets unexpected results when using the code:

```
exp(-(x - mu)^2 / 2 * sigma^2)
```

Explain the problem and correct the code.

59) Indicate whether the following statements are True of False.

- CSV format is a comma-delimited format.

- All data values in a plain text file are stored as a series of characters.
- The first row in a csv file is always used to indicate column names.
- Delimited format files allow you to define any sort of complex data structure.
- The ubiquity of spreadsheets, and its format of row-and-columns, make them the most efficient way to represent data.
- Spreadsheet data files stored in a riched format (e.g. Excel) can be opened in a text editor to inspect its contents.
- Spreadsheet software (e.g. Excel) can be used to view or explore field-delimited files.

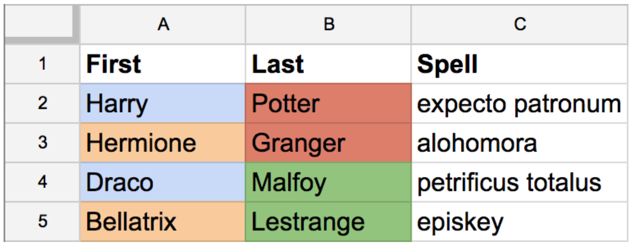
60) Consider the following table (containing some "messy" data), and assume it is stored in a `csv` file:

name	gender	height	weight	status
Anakin	male	1.88m	84kg	1 = jedi
Padme	Female	1.65m	???	2 = queen
Luke	MALE	1.72m	77kg	1 = jedi
Leia	female	150cm	49kg	3 = princess

One of your friends has to do an exploratory analysis of the data table, and she asks for your help. Name four data cleaning aspects that you would discuss with your friend before doing any exploration.

61) Based on the previous data set, you suggest your friend to create a data dictionary (i.e. metadata). How would you create such dictionary? (verbal description, no code).

Consider a spreadsheet with the following content



	A	B	C
1	**First**	**Last**	**Spell**
2	Harry	Potter	expecto patronum
3	Hermione	Granger	alohomora
4	Draco	Malfoy	petrificus totalus
5	Bellatrix	Lestrange	episkey

- The cells `Harry` and `Draco` have a *blue* background indicating male gender.
- The cells `Hermione` and `Bellatrix` have an *orange* background indicating female gender.
- The cells `Potter` and `Granger` have a *maroon* background indicating house of Gryffindor.
- The cells `Malfoy` and `Lestrange` have a *green* background indicating house of Slytherin.

62) What is the issue with using highlighting features to codify information in a spreadsheet?

63) Describe how would you modify the content in the spreadsheet with Harry Potter's data to avoid highlighting cells. (verbal description, no code).

64) Below is a list of possible names for an R vector. Some names are valid and some are invalid. Identify the invalid names, and explain why they are invalid.

a. `3var_name`

b. `var_name4`

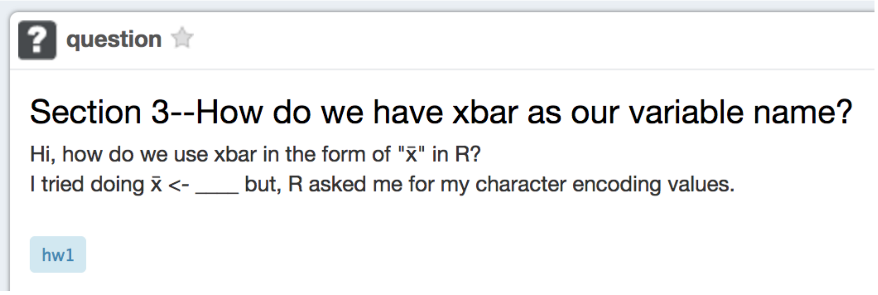
c. `_var_name`

d. `.var_name`

e. `VarName`

f. `var-name`

65) Consider the following question posted in piazza, associated with the formula to calculate the mean (or average): $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.



Section 3--How do we have xbar as our variable name?

Hi, how do we use xbar in the form of "x̄" in R?
I tried doing x̄ <- ____ but, R asked me for my character encoding values.

What is the issue with the attempted code, and how would you help to solve it?

66) This question is based on a post from piazza (slitghtly adapted).

Consider the previous tibble `sw` and the four commands below:

```
# star wars data frame
sw
```

```
# A tibble: 4 x 4
  name   gender height weight
  <chr>  <chr>   <dbl>  <int>
1 Anakin male     1.88     84
2 Padme  female   1.65     45
3 Luke   male     1.72     77
4 Leia   female   1.50     49
```

```
typeof(sw[["weight"]])
```

```
## [1] "integer"
```

```
typeof(sw$weight)
```

```
## [1] "integer"
```

```
typeof(sw["weight"])
```

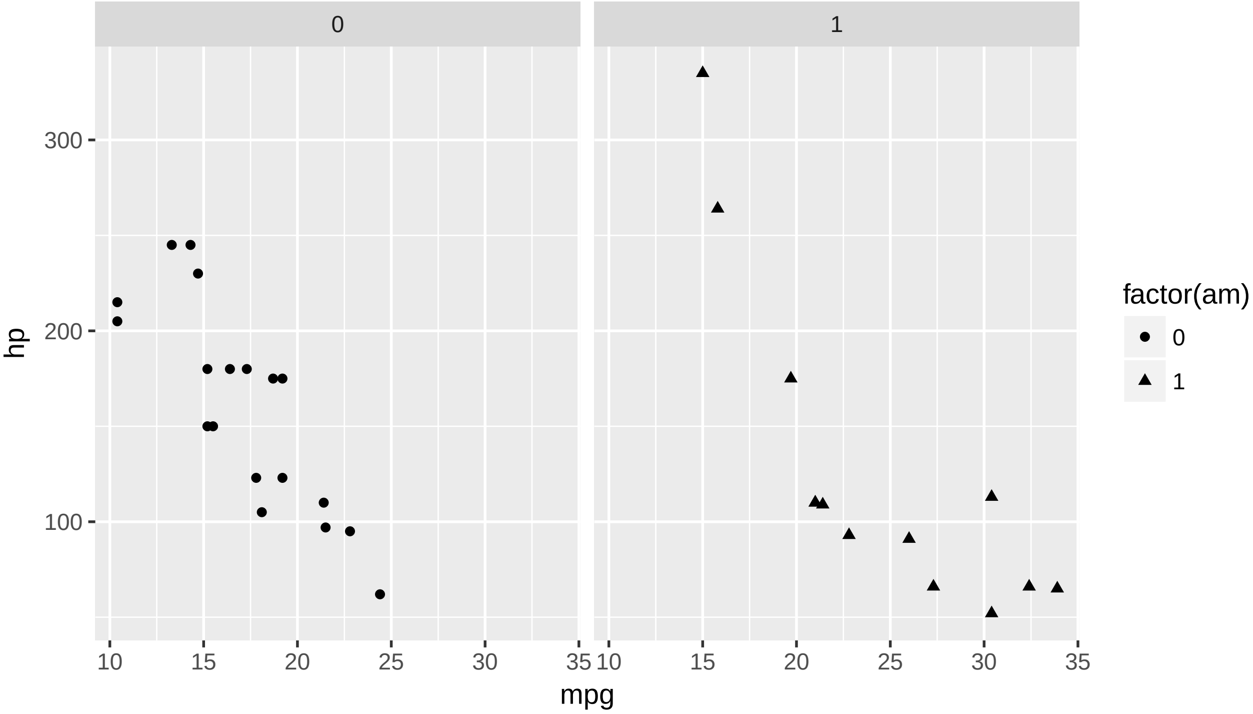
```
## [1] "list"
```

```
typeof(sw[,"weight"])
```

```
## [1] "list"
```

Why do `typeof(sw[["weight"]])` and `typeof(sw$weight)` return `"integer"`, while `typeof(sw["weight"])` and `typeof(sw[,"weight"])` return `"list"`?

67) Consider the following plot obtained using the data `mtcars`:



What call generates the previous figure:

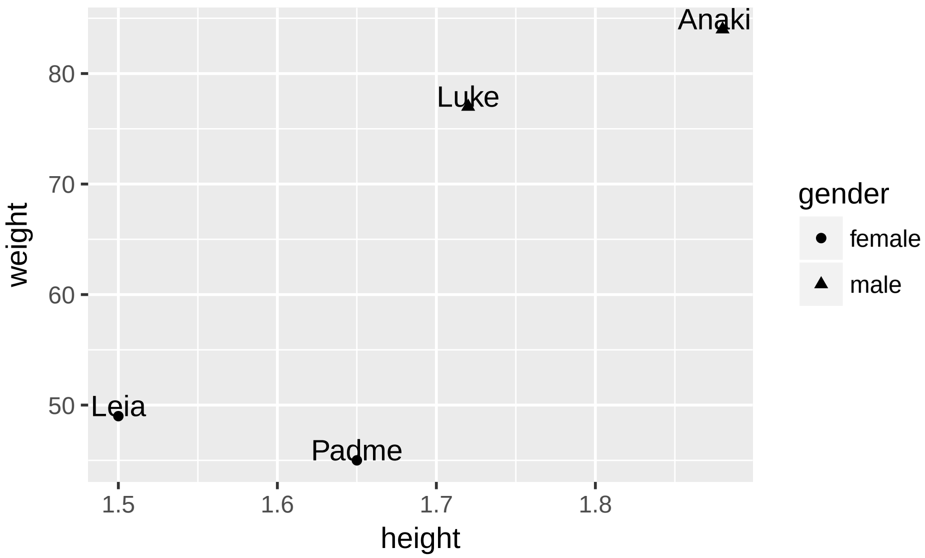
```
# option 1
ggplot(mtcars, aes(x = hp, y = mpg)) +
  geom_point(aes(shape = factor(am))) +
  facet_wrap(~ am)
```

```
# option 2
ggplot(mtcars, aes(x = mpg, y = hp)) +
  geom_text(size = factor(am)) +
  facet_wrap(~ am)
```

```
# option 3
ggplot(mtcars, aes(x = mpg, y = hp)) +
  geom_point(aes(shape = factor(am))) +
  facet_frames(~ am)
```

```
# option 4
ggplot(mtcars, aes(x = mpg, y = hp)) +
  geom_point(aes(shape = factor(am))) +
  facet_wrap(~ am)
```

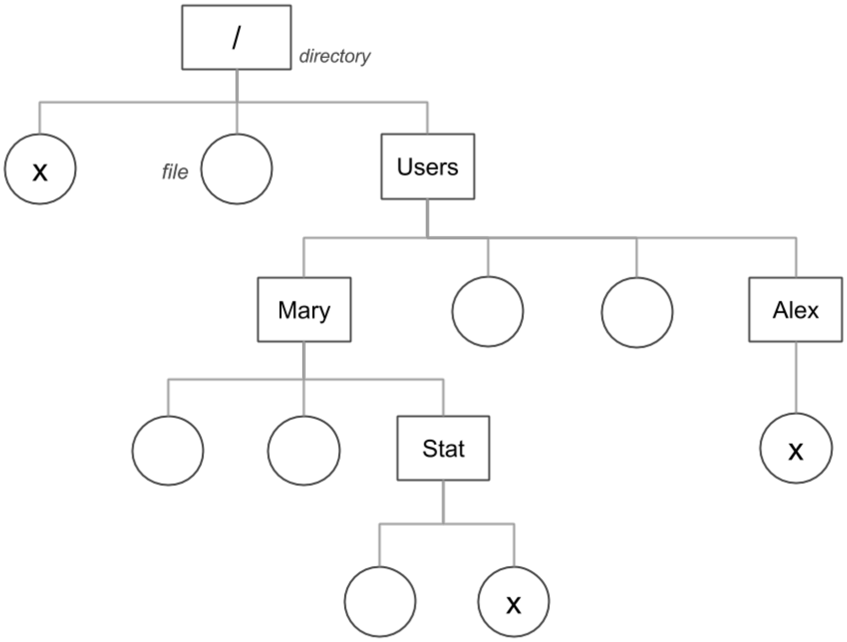
68) Consider the following ggplot based on the tibble `sw`:



Fill in the blanks:

```
ggplot(data = ______, aes(x = _________, y = _________))  ___
  geom__________(______(shape = gender))  ___
  geom__________(aes(label = name))
```

Consider the filesystem illustrated in the scheme below:



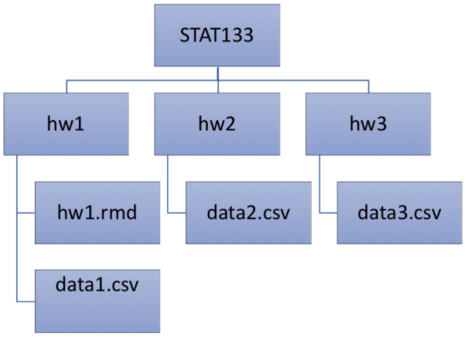
69) Write the absolute path name to the `x` file in directory `Alex`.

70) Write the relative path from `Alex` to `Stat`.

71) Select the option that indicates the relative path to file `x` at the top from within the directory `Stat`:

a) `../Mary/x`
b) `../../../x`
c) `/x`
d) `/Users/Mary/Stat/x`

72) Consider the filesystem illustrated in the scheme below:



a. Suppose you are at `STAT133` directory, write the shell command to navigate to directory `hw1`.

b. Suppose you are at `hw1` directory, write the shell command to list contents in your current directory in long format.

c. Suppose you are at `hw1` directory, write the shell command to create a new subdirectory `image` in your working directory.

d. Suppose you are at `hw1` directory, write the shell command to rename `data2.csv` as `newdata.csv`

e. Suppose you are at `hw1` directory. Explain what the command `rm *.csv` does?

f. Suppose you are at `hw1` directory. Explain what the command `cp ./data1.csv ../hw2/data2.csv` does?

73) Suupose you are working at the command line (e.g. terminal, gitbash) and your working directory contains a CSV file `data.csv`. Explain what the follwoing commands are doing:

a. `wc data.csv`

b. `wc -l data.csv`

c. `head data.csv`

d. `tail data.csv`

e. `less data.csv`

f. `head -n 11 data.csv > newdata.csv`

g. `cut -d "," -f 3 data.csv`

h. `cut -d "," -f 3 data.csv | tail +2 > newfile.csv`

74) Assume that `a` and `b` are two numbers. Indicate whether the following commands are valid or invalid.

a) `if a > b {z <- a + b}`

b) `if (a > b) z <- a + b else y <- a * b`

c) `if (a > b) then z <- a + b`

d) `if (a > b) z <- a + b`

e) `if (a > b) {z <- a + b} else {y <- a * b}`

75) For a set of numbers $x_1, x_2, x_3, \ldots, x_n$, the power mean with exponent p is defined as:

$$\left(\frac{1}{n} \sum_{i}^{n} x_i^p \right)^{1/p} = \frac{1}{n} (x_1^p + \cdots + x_n^p)^p$$

Assume that the values for $x_1, x_2, x_3, \ldots, x_n$ are given by the following vector `x`:

```
# vector of numbers
x <- 1:10
```

One of your friends wrote a function in R to calculate the power mean as:

```
# function to compute power mean
# input: a numeric vector
# output: power mean
power-mean <- function(y, p) {
  n <- length(x)
  summation <- 0
  for (i in 1:n) {
    summation + (x[i]^p)
  }
  summation <- (1/n) * summation
  return(summation^(1/p)
}
```

Your friend asks you to review the code and look for any bugs. Help your friend find any errors, and write code that fixes the function.

76) The formula of the weighted average is:

$$avg = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_n x_n}{w_1 + w_2 + \cdots + w_n}$$

Assume that the values for x and w are given by the following vectors:

```
# vectors of values and weights
x <- 1:10
w <- seq(from = 0.1, to = 1, by = 0.1)
```

One of your friends wrote R code to implement such formula in the following way:

```r
# computation of the numerator
numerator <- 0
for (j in 1:10) {
  numerator <- numerator + (w[j] * x[j])
}

# computation of the denominator
denominator <- 0
for (i in 1:10) {
  denominator <- denominator + w[j]
}

# computation of weighted average
avg <- denominator / numerator
```

Your friend asks you to review the code and look for any bugs. What would you tell your friend?

77) The Compound Interest Formula is given by:

$$A = P\left(1 + \frac{r}{n}\right)^{nt}$$

where:

- P = principal amount (the initial amount you borrow or deposit)
- r = annual interest rate (as a decimal)
- t = number of years the amount is deposited or borrowed for.
- n = number of times the interest is compounded per year
- A = amount of money accumulated after n years, including interest.

Write a function `amount()` to implement the formula of the amount with compound interest. Use descriptive names for the arguments. In other words, don't use `P` or `r`, for principal or interest; instead choose a more descriptive name. Also, give default values to all the arguments. This function should return the corresponding amount of money.

78) What does the following code print out at each iteration?

```r
f <- 1
g <- 1

for (i in 1:5) {
  print(g)
  g <- f - g
  f <- f + g
}
```

79) Consider the gaussian (Normal) function, given in the equation below, and the code that implements such equation.

$$f(x) = \frac{1}{\sqrt{2\pi}s} exp\left\{-\frac{1}{2}\left(\frac{x-m}{s}\right)^2\right\}$$

```r
# gaussian function
f<-function(x=1,m=0,s=1){
a<-1/(sqrt(2*pi))
b<-exp(-0.5*((x-m)/s)^2)
a*(1/s)*b
}
```

The function works and it has no bugs... but it is hard to review the code at first glance. Name at least four aspects that you would change to improve readability.

80) The code below takes each element in the vector `x` and transforms it by calling `floor()` and adding an integer `K`. The transformed values are stored in the vector `y`. Rewrite the code below without using a loop or an apply function:

```r
x <- rnorm(50, 0, 2)
y <- x
for (i in 1:length(y)) {
  K <- 2
  y[i] <- floor(x[i]) + K
}
```

81) Use logical subsetting to rewrite the following code, eliminating the need for any loops or if statement (Don't write a function).

```r
x <- rnorm(20)
y <- x

for (i in 1:length(x)) {
   if (x[i] < -1) {
     y[i] <- 0
   } else {
     if (x[i] > 1) {
       y[i] <- 1
     } else y[i] <- 3 * x[i]^2
   }
}
```

82) Match the provided concepts with the corresponding descriptions:

```
 - dot                        - caret: ^
 - regular expression         - [:xdigit:]
 - metacharacters             - asterisk: *
 - character class            - \\
```

a) _____ characters with special meaning, that do not match themselves literally

b) _____ match hexadecimal digits

c) _____ match only one out of several characters

d) _____ pattern describing a certain amount of text

e) _____ used for escaping characters in R

f) _____ match the preceding token zero or more times

g) _____ typing it after an opening bracket negates the character class

h) _____ matches a single character

83) Consider the following character vector:

```
pns <- c("pan", "pen", "pin", "pon", "pun", "p.n", "p1n")
```

What elements are matched by the following regular expressions (write the name of the elements, NOT their positions). *Note*: matched wither by `grep()` or `str_detect()`

a. `"[ei]"`

b. `"p.n"`

c. `"p[[:alpha:]]n"`

d. `"p[0-9]n"`

e. `"p..n"`

f. `"\\d"`

g. `"\\."`

84) Consider the following character vector:

```
food <- c("burrito", "burger", "pizza", "salad")
```

What elements are matched by the following regular expressions (write the name of the elements, NOT their positions):

a. `"[ei]"`

b. `"r[r]"`

c. `"[alpha]"`

d. `"[[:alpha:]]"`

e. `"\\w+"`