PSTAT 10: Homework 1

Solution Sketches

If we find these solutions posted to the likes of Chegg, CourseHero, etc., we will cancel all future quizzes and move the corresponding weights of your final grade solely to the final exam.

Please Note:

- There are often many ways of solving each problem. Just because the solution sketches below are different than your own personal approach doesn't mean your approach is invalid!
- Do not post these solutions anywhere (e.g. Chegg, CourseHero, etc.). If we find these solutions posted online, we will change the grading structure of the course to put 100% weight on the final exam.

Instructions

- Remember to always copy the necessary files to your working directory; never edit and/or knit the files directly in the `10f22-content` folder.
- Rename your `.Rmd` file as `hw01-your-netid.Rmd`. For example, my Rmd script file will be `hw01-umaravat.Rmd`.
- Make sure to include your identifying details in the Rmd file.
- **Please keep in mind that there are 5 additional Multiple Choice questions that appear on Canvas**, in the submission portal for this homework. These questions are still considered a part of the homework, and you must complete them in order to be eligible to earn full points.
- When submitting to Canvas, follow the provided submission instructions carefully.

Exercises

Exercise 0: YAML

Update the YAML to reflect your own information (i.e. name, collaborators, etc.). If you did not collaborate with anyone, note that down in the YAML.

Exercise 1: Lecture 2 Summary

Part (a): DataTypes

Write down a bulleted list of the types of data `R` supports. Make sure your list displays properly in your final knitted document.

Solutions:

- `character` (also known as `string`)
- `double` (also known as `numeric`)
- `integer`
- `logical`

Part (b): Glossary of Functions

Fill in the following table with the functions you learned in Lecture 02. Add as many lines as you feel are necessary (remember that Worksheet 0 covered how to add rows to tables in Markdown).

Solutions:

This one is open-ended and more for your reference

R Function	*Description of Function*	*Example of the Function*
`c`	combines elements into a vector	`c(1, 2, "hello")`
`mean`	computes the mean of a vector	`mean(c(1,4,5)`
`median`	computes the median of a vector	`median(c(10,11,12,13)`
`typeof()`	returns the data type of a given object	`typeof(c("hello", "world"))`
`length()`	returns the length of a given vector	`length(1:3)`
`sort()`	returns a sorted version of a given vector	`sort(c("Cat", "Dog"))`

Exercise 2: Splitting Hairs

In this problem, we'll deal with a new function that can help manage strings. Remember- HELP files are your friend!

(a) Look up the `strsplit()` function in RStudio Help.

Solutions:

```
?strsplit
```

(b) Write down code that takes the string “hair” and returns “h” “a” “i” “r”.

Solutions:

```
strsplit("hair", "")
```

```
## [[1]]
## [1] "h" "a" "i" "r"
```

(c) The function `extract_month()`, which has been partially written below, is designed to take in a date of the form MM/DD/YY, and extract out the month. For instance, `extract_month("09/30/22")` should return `"09"`. Complete the code to ensure the function works as expected.

```
extract_month <- function(x) {
  split_date <- <REPLACE THIS WITH YOUR CODE>
  return(split_date[[1]][1]) # DO NOT CHANGE THIS LINE
}
```

Solutions:

```
extract_month <- function(x) {
  split_date <- strsplit(x, "/")
  return(split_date[[1]][1])
}
```

(d) Test your function with 3 different dates.

Solutions:

```
extract_month("09/30/22")
```

```
## [1] "09"
```

```
extract_month("10/31/22")
```

```
## [1] "10"
```

```
extract_month("08/1/20")
```

```
## [1] "08"
```

Exercise 3: Chunky Guacamole

In this problem, we'll get some more practice with basic `R` commands. For each part, you will need to insert a code chunk on the line right after the **Solutions:** line. Additionally, at least one of the parts below may result in a code chunk that contains an error; **that is fine!** Just make sure your final document still knits, even with that error (hint: what options can you pass into your code chunk to ensure that it knits, despite there being an error present? This was covered in Lecture 1)

(a) Assign the value `3` to a variable called `guac`.

Solutions:

```
guac <- 3
```

(b) Assign the value `1` to a variable called `a`.

Solutions:

```
a <- 1
```

(c) Assign the value `"yummy"` to a variable called `mole`.

Solutions:

```
mole <- "yummy"
```

(d) What should the code `guac + a` return? Confirm your answer by running the code in a code chunk.

Solutions: The code should return 4 (i.e. the result of 3 + 1):

```
guac + a
```

```
## [1] 4
```

(e) What do you think `guac + a + mole` will return? *No coding required for this part*

Solutions: Student answers may vary; I suspect students will say something to the effect of "an error."

(f) Now, run the code `guac + a + mole` and comment on the result.

Solutions:

```
guac + a + mole
```

```
## Error in guac + a + mole: non-numeric argument to binary operator
```

Remark: You will also have needed to add `error = T` into your code chunk header, to ensure that the `.Rmd` still knitted while displaying the error you received.

Exercise 4: Palmy Weather

The file `palm.csv` contains data on the prevalence of four different types of palm trees in the Santa Barbara Area. The four types of palm trees are encoded as:

- `MFP` : Mexican Fan Palm
- `KP` : King Palm
- `QP` : Queen Palm
- `FP` : Foxtail Palm

Each row of the data consists of the number of each type of palms in a certain city block; each block is coded with a unique identifying number, which is stored as `ID` .

The following code chunk can be used to load in the dataset as a dataframe, assigned to the variable `palm` :

```
palm <- read.csv("./data/palm.csv", header = T)
```

(a) How many city blocks were included in this study?

Solutions:

Since each row of the dataset corresponds to a city block, the number of city blocks surveyed is simply the number of rows of `palm` :

```
nrow(palm)
```

```
## [1] 232
```

(b) What is the average density of Mexican Fan Palms per City Block?

Solutions:

```
mean(palm$MFP)
```

```
## [1] 9.155172
```

(c) What is the median density of Foxtail Palms per City Block?

Solutions:

```
median(palm$MFP)
```

```
## [1] 9
```

(d) Which City Block contains the most King Palms?

Solutions:

```
which.max(palm$KP)
```

```
## [1] 3
```

Remark: There are multiple city blocks with the same maximum number of King Palms. That is okay; we only needed you to return the first city block with the maximum number of King Palms (which is precisely what `which.max()` does when there isn't a unique index corresponding to a maximum).

(e) Which city block contains the fewest Queen Palms?

Solutions:

```
which.min(palm$QP)
```

```
## [1] 2
```

Exercise 5: Score!

Suppose we have exam scores for 5 students: Brad, Angelina, Sandra, Michelle, and John. Their scores are 7, 5, 9, 10, and 6, respectively.

(a) Create a vector of these scores.

Solutions:

```
scores <- c(7, 5, 9, 10, 6)
```

(b) Find the mean score in two ways (using `mean` and using `sum`).

Solutions:

```
# using mean
mean(scores)
```

```
## [1] 7.4
```

```
# using sum
sum(scores) / length(scores)
```

```
## [1] 7.4
```

(c) Find the median score.

Solutions:

```
median(scores)
```

```
## [1] 7
```

(d) Assign the name of each student to their test score.

Solutions:

```
names(scores) <- c("Brad", "Angelina", "Sandra", "Michelle", "John")
```

(e) Retrieve Angelina's score in two ways.

Solutions:

```
# indexing from the left:
scores[2]
```

```
## Angelina
##        5
```

```
#  second method : using the names attribute
scores["Angelina"]
```

```
## Angelina
##        5
```

(f) Retrieve Brad's and Sandra's scores, in that order.

Solutions:

```
scores[c(1, 3)]
```

```
##   Brad Sandra
##      7      9
```

```
scores[c("Brad", "Sandra")]
```

```
##   Brad Sandra
##      7      9
```

(g) Retrieve the scores of everyone except John.

Solutions:

```
scores[-5]
```

```
##     Brad Angelina   Sandra Michelle
##        7        5        9       10
```