STAT 456 Homework 1

Hailey Lee

02/01/2024

Instructions:

1. Please use R to finish all the questions below. Although in some simple cases, you may obtain the solution directly without using R, you still need to provide the corresponding R code.
2. You are liable for missing points if you don't include output;
3. Whenever possible, please run saved variables, so our TA knows if your code goes the right way and assigns partial credits even if your final answer is wrong.
4. Please submit your solutions in .rmd file and .pdf file compiled via the R markdown through Blackboard.

1. What is the output of the following commands? Try to predict the solutions before you type in the commands. We define:

```
x <- c(5, 2, 1, 4); xx <- c(1, 10, 15, 18); y <- rev(10:15);
yy <- seq(length = 10, from  = 9, by = 2)
z <- c(TRUE, FALSE, TRUE, TRUE); w <- c("Marie", "Betty", "Peter");
v <- c("I", "list", "key", "standard", "attraction")
```

(a)

```
order(x)
```

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

```
sum(x)
```

```
## [1] 12
```

```
range(x)
```

```
## [1] 1 5
```

```
length(y)
```

```
## [1] 6
```

```
sum(yy)
```

```
## [1] 180
```

(b)

```
c(xx, yy, 12)
```

```
##  [1]  1 10 15 18  9 11 13 15 17 19 21 23 25 27 12
```

(c)

```
xx - x
```

```
## [1] -4  8 14 14
```

```
c(x, 12) * y
```

```
## Warning in c(x, 12) * y: longer object length is not a multiple of shorter
## object length
```

```
## [1]  75  28  13  48 132  50
```

```
1:6 + 1
```

```
## [1] 2 3 4 5 6 7
```

```
1:9 + 1:2
```

```
## Warning in 1:9 + 1:2: longer object length is not a multiple of shorter object
## length

## [1]  2  4  4  6  6  8  8 10 10
```

```r
log(xx)
```

```
## [1] 0.000000 2.302585 2.708050 2.890372
```

(d)

```r
x <= 2
```

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

```r
(x < 2) & z
```

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

```r
x < (2 & z)
```

```
## [1] FALSE FALSE FALSE FALSE
```

```r
(x == 1) + 2
```

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

(e)

```r
substring(w, 2, 4)
```

```
## [1] "ari" "ett" "ete"
```

```r
paste(substring(w,1,2), substring(w, 5, 5), sep = "..")
```

```
## [1] "Ma..e" "Be..y" "Pe..r"
```

```r
paste(substring(v, 1, 2), collapse = " ")
```

```
## [1] "I li ke st at"
```

(f)

```r
rbind(x,xx)
```

```
##    [,1] [,2] [,3] [,4]
## x     5    2    1    4
## xx    1   10   15   18
```

```r
cbind(2, 6:1, rep(c(3, 1, 4), 2), seq(1, 1.6, by = 0.05))
```

```
## Warning in cbind(2, 6:1, rep(c(3, 1, 4), 2), seq(1, 1.6, by = 0.05)): number of
## rows of result is not a multiple of vector length (arg 2)

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

```
## [1,]    2    6    3 1.00
## [2,]    2    5    1 1.05
## [3,]    2    4    4 1.10
## [4,]    2    3    3 1.15
## [5,]    2    2    1 1.20
## [6,]    2    1    4 1.25
## [7,]    2    6    3 1.30
## [8,]    2    5    1 1.35
## [9,]    2    4    4 1.40
## [10,]   2    3    3 1.45
## [11,]   2    2    1 1.50
## [12,]   2    1    4 1.55
## [13,]   2    6    3 1.60
```

2. Give the R commands to do the following:

   (a) Use `rep()` to create a vector that has the following elements: 2.7, 8.0, 3.0, 2.7, 8.0, 3.0.

```r
rep(c(2.7, 8.0, 3.0), times = 2)
```

```
## [1] 2.7 8.0 3.0 2.7 8.0 3.0
```

   (b) Use `seq()` with the length.out argument to create a vector that has the following elements: 0.0, 0.5, 1.0, 1.5, 2.0.

```r
seq(0.0, 2.0, by = 0.5)
```

```
## [1] 0.0 0.5 1.0 1.5 2.0
```

   (c) Let `x` be a vector of length three or greater that contains numeric elements. Write some R commands that calculate the sample mean of all of the elements of `x` except the smallest and largest. For example, if `x = (1,2,4,5,6,7)`, you should output the mean of `(2,4,5,6)`. If `x = (1,1,2,3,4,4)`, you should output the mean of `x = (2,3)`. [Hint: Make use of `which`, `min`, `max`, `mean`.]

```r
x <- c(1, 1, 2, 3, 4, 4)
mean(x[-c(which.max(x), which.min(x))])
```

```
## [1] 2.5
```

   (d) Let `x` be a vector containing numerical elements, each of which is a positive integer. Write a single line of R code that replaces each element that is a perfect square with zero. For example, if `x=(1,2,3,4)`, then you should return `(0,2,3,0)`.

```r
x <- 1:4
x[sqrt(x)%%1 == 0] <- 0
x
```

```
## [1] 0 2 3 0
```

3. Suppose you have been keeping track your study times for over three weeks (20 days) and you find the following times in hours:

   `7   10   12   5   6   9.5   13   15   8   11   6   8.5   3   9   14   9.5 10.5   6   7.5   12`

   Use R to answer the following questions.

   (a) Use the function `max` and `min` to find the longest and shortest study time, the function `mean` and `median` to find the average and the median number.

```
study <- c(7, 10, 12, 5, 6, 9.5, 13, 15, 8, 11, 6, 8.5, 3, 9, 14, 9.5, 10.5, 6, 7.5, 12)
max(study)
```

```
## [1] 15
```

```
min(study)
```

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

```
mean(study)
```

```
## [1] 9.125
```

```
median(study)
```

```
## [1] 9.25
```

   (b) The 10.5 was a mistake. It should have been 13. How can you fix this? Do so, and then find the new average.

```
study[study==10.5]<-13
mean(study)
```

```
## [1] 9.25
```

   (c) How many times was your study time 10 hours or more a day?

```
sum(study >= 10)
```

```
## [1] 8
```

   (d) What percent of your study time are less than 8 hours/day?

```
(sum(study < 8)/length(study))*100
```

```
## [1] 35
```

4. Your department's utility bill varies from Jan to Dec. Suppose it has the following monthly amounts:

   `20 33 37 31 32 30 NA 47 41 33 48 52`

   Enter this data into a variable called `Utility`.

```r
Utility <- c(20, 33, 37, 31, 32, 30, NA, 47, 41, 33, 48, 52)
Utility
```

```
##  [1] 20 33 37 31 32 30 NA 47 41 33 48 52
```

(a) Give labels for each value from Jan to Dec (use abbreviation).[Hint: Make use of `month.abb`.]

```r
names(Utility) <- month.abb
Utility
```

```
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
##  20  33  37  31  32  30  NA  47  41  33  48  52
```

(b) We missed one bill for July, replace the number with the mean of the yearly bill (11 months). [Hint: Make use of the `na.rm` option in the `mean`} function.]

```r
Utility["Jul"] <- mean(Utility, na.rm=TRUE)
Utility
```

```
##      Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 20.00000 33.00000 37.00000 31.00000 32.00000 30.00000 36.72727 47.00000
##      Sep      Oct      Nov      Dec
## 41.00000 33.00000 48.00000 52.00000
```

(c) After updated the new data using part (b), what is the variance of the yearly bill (12 months)? What is the standard deviation (12 months)?

```r
var(Utility)
```

```
## [1] 81.10744
```

```r
sd(Utility)
```

```
## [1] 9.005967
```

5. The built-in data set `mtcars` contains information about cars from a 1974 Motor Trend issue. Load the dataset (`data(mtcars)`) and try to answer the following:

```r
data(mtcars)
```

(a) What are the variable names? (Try `names`.)

```r
names(mtcars)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

(b) What is the maximum mpg?

```r
max(mtcars$mpg)
```

```
## [1] 33.9
```

(c) Which car has this?

```
rownames(mtcars[mtcars$mpg == 33.9,])
```

```
## [1] "Toyota Corolla"
```

(d) What horsepower (hp) does the `Valiant` have?

```
mtcars["Valiant",]$hp
```

```
## [1] 105
```

(e) What are the first 5 cars listed?

```
mtcars[1:5,]
```

```
##                    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
```

(f) What are all the values for the Mercedes 450slc (`Merc 450SLC`)?

```
mtcars[rownames(mtcars) == "Merc 450SLC",]
```

```
##              mpg cyl  disp  hp drat   wt qsec vs am gear carb
## Merc 450SLC 15.2   8 275.8 180 3.07 3.78   18  0  0    3    3
```

6. In `library(lattice)`, there is a build-in dataset named `barley`, which is a data frame with 120 observations on the following 4 variables.

   - `yield`: Yield (averaged across three blocks) in bushels/acre.
   - `variety`: Factor with levels `Svansota`, `No. 462`, `Manchuria`, `No. 475`, `Velvet`, `Peatland`, `Glabron`, `No. 457`, `Wisconsin No. 38`, `Trebi`.
   - `year`: Factor with levels 1932, 1931.
   - `site`: Factor with 6 levels: `Grand Rapids`, `Duluth`, `University Farm`, `Morris`, `Crookston`, `Waseca`.

   ```
   library(lattice)
   data(barley)
   ```

   (a) For variable `yield`, find the mean, standard deviation and quantiles of at 10, 20, 30, . . . , 90%.

```
mean(barley$yield)
```

```
## [1] 34.42056
```

```r
sd(barley$yield)
```

```
## [1] 10.33471
```

```r
quantile(barley$yield, na.rm = TRUE, c(0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0
```

```
##      10%      20%      30%      40%      50%      60%      70%      80%
## 22.49667 26.08000 28.09000 29.94667 32.86667 35.13333 38.97333 43.32000
##      90%
## 47.45666
```

(b) For values of `yield` obtained in 1931, find the mean, standard deviation and quantiles of at 10, 20, 30, ..., 90%. Repeat the above for values of `yield` obtained in 1932.

```r
mean(barley[barley$year == 1931,]$yield)
```

```
## [1] 37.07778
```

```r
sd(barley[barley$year == 1931,]$yield)
```

```
## [1] 10.62984
```

```r
quantile(barley[barley$year == 1931,]$yield, na.rm = TRUE, c(0.10, 0.20, 0.30, 0.40, 0.5
```

```
##      10%      20%      30%      40%      50%      60%      70%      80%
## 25.76000 28.58000 29.66667 32.10000 34.20000 38.60000 42.04000 46.80000
##      90%
## 49.90334
```

```r
mean(barley[barley$year == 1932,]$yield)
```

```
## [1] 31.76333
```

```r
sd(barley[barley$year == 1932,]$yield)
```

```
## [1] 9.384457
```

```r
quantile(barley[barley$year == 1932,]$yield, na.rm = TRUE, c(0.10, 0.20, 0.30, 0.40, 0.5
```

```
##      10%      20%      30%      40%      50%      60%      70%      80%
## 20.63333 22.55334 26.35334 27.81334 30.98334 33.81333 36.26333 39.32000
##      90%
## 44.28000
```

(c) Retrieve all rows of the Barley data in 1931 where the yield lies above the 90% quantile calculated in (b) and name it as `df`.

```r
df <- barley[barley$year == 1931 & barley$yield > 49.90334,]
df
```

```
##       yield          variety year   site
```

```
## 8  55.20000           Glabron 1931 Waseca
## 20 50.23333            Velvet 1931 Waseca
## 26 63.83330             Trebi 1931 Waseca
## 32 58.10000           No. 457 1931 Waseca
## 38 65.76670           No. 462 1931 Waseca
## 56 58.80000 Wisconsin No. 38 1931 Waseca
```