Rworksheet_MAMINTA#4a

#1.

#a. Describe the data.

```
Respondent <- 1:8
Gender <- c("M", "F", "F", "M", "F", "M", "F", "M")
ShoeSize <- c(9, 7, 6, 10, 6.5, 9.5, 7, 11)
Height_cm <- c(175, 160, 158, 180, 162, 176, 159, 185)
shoe_height_df <- data.frame(Respondent, Gender, ShoeSize, Height_cm)
shoe_height_df
```

```
##   Respondent Gender ShoeSize Height_cm
## 1          1      M      9.0       175
## 2          2      F      7.0       160
## 3          3      F      6.0       158
## 4          4      M     10.0       180
## 5          5      F      6.5       162
## 6          6      M      9.5       176
## 7          7      F      7.0       159
## 8          8      M     11.0       185
```

#b. Create a subset by males and females with their corresponding shoe size and height.

```
males <- subset(shoe_height_df, Gender == "M", select = c(ShoeSize, Height_cm))
females <- subset(shoe_height_df, Gender == "F", select = c(ShoeSize, Height_cm))
males
```

```
##   ShoeSize Height_cm
## 1      9.0       175
## 4     10.0       180
## 6      9.5       176
## 8     11.0       185
```

```
females
```

```
##   ShoeSize Height_cm
## 2      7.0       160
## 3      6.0       158
## 5      6.5       162
## 7      7.0       159
```

#c. Find the mean of shoe size and height of the respondents.

```r
mean_shoe_overall <- mean(shoe_height_df$ShoeSize)
mean_height_overall <- mean(shoe_height_df$Height_cm)
mean_shoe_by_gender <- tapply(shoe_height_df$ShoeSize, shoe_height_df$Gender, mean)
mean_height_by_gender <- tapply(shoe_height_df$Height_cm, shoe_height_df$Gender, mean)


mean_shoe_overall
```

```
## [1] 8.25
```

```r
mean_height_overall
```

```
## [1] 169.375
```

```r
mean_shoe_by_gender
```

```
##     F     M
## 6.625 9.875
```

```r
mean_height_by_gender
```

```
##      F      M
## 159.75 179.00
```

#d. Is there a relationship between shoe size and height? Why?

```r
#The dataset shows that people with larger shoe sizes also tend to be taller.
#This makes sense, as shoe size is often linked to body proportions, and taller
#individuals usually have bigger feet. Although the pattern is visible in the
#data, using a statistical method like correlation analysis would provide a more
#precise measure of how strong this relationship is.
```

```r
##2. Construct character vector months to a factor with factor() and assign the
#result to factor_months_vector. Print out factor_months_vector and assert that
#R prints out the factor levels below the actual values.

Months <- c(
  "March","April","January","November","January",
  "September","October","September","November","August",
  "January","November","November","February","May","August",
  "July","December","August","August","September","November","February",
  "April"
)

factor <- factor(Months)
factor
```

```
##  [1] March     April     January   November  January   September October
##  [8] September November  August    January   November  November  February
## [15] May       August    July      December  August    August    September
## [22] November  February  April
## 11 Levels: April August December February January July March May ... September
```

```r
#3. Then check the summary() of the months_vector and factor_months_vector. Inter-
#pret the results of both vectors. Are they both equally useful in this case?

summary(Months)
```

```
##    Length     Class      Mode
##        24 character character
```

```r
summary(factor)
```

```
##     April    August  December  February   January      July     March       May
##         2         4         1         2         3         1         1         1
##  November   October September
##         5         1         3
```

```r
#4. Create a vector and factor for the table below.
factor_data <- c("East", rep("West", 4), rep("North", 3))
factor_data
```

```
## [1] "East"  "West"  "West"  "West"  "West"  "North" "North" "North"
```

```r
new_order_data <- factor(factor_data, levels = c("East", "West", "North"))
print(new_order_data)
```

```
## [1] East  West  West  West  West  North North North
## Levels: East West North
```

```r
#5. Enter the data below in Excel with file name = import_march.csv

import_march <- read.table("import_march.csv", header = TRUE, sep = ",")
import_march
```

```
##   Students Strategy.1 Strategy.2 Strategy.3
## 1     Male          8         10          8
## 2                   4          8          6
## 3                   0          6          4
## 4   Female         14          4         15
## 5                  10          2         12
## 6                   6          0          9
```

```r
#6. Full Search

#a. Create an R Program that allows the User to randomly select numbers from 1 to 50.
#Then display the chosen number. If the number is beyond the range of the selected choice,
#it will have to display a string "The number selected is beyond the range of 1 to 50". If
#number 20 is inputted by the User, it will have to display "TRUE", otherwise display the
#input number.

num <- as.integer(readline("Choose a number (1 to 50): "))
```

```
## Choose a number (1 to 50):
```

```r
cat("Chosen number:", num, "\n")
```

```
## Chosen number: NA
```

```r
if (is.na(num)) {
  cat("Invalid!\n")

} else if (num < 1 || num > 50) {
  cat("Choose only from 1-50!\n")

} else if (num == 20) {
  print(TRUE)

} else {
  print(num)
}
```

```
## Invalid!
```

```r
#7. Change
#At ISATU University's traditional cafeteria, snacks can only be purchased with bills. A
#long-standing rule at the concession stand is that snacks must be purchased with as few
#coins as possible. There are three types of bills: 50 pesos, 100 pesos, 200 pesos, 500 pesos,
#1000 pesos.
#a. Write a function that prints the minimum number of bills that must be paid, given the
#price of the snack.
#Input: Price of snack (a random number divisible by 50) Output: Minimum number of bills
#needed to purchase a snack.

min_bills <- function(price) {
  bills <- c(1000, 500, 200, 100, 50)

  remaining <- price
  count <- 0

  for (b in bills) {
    if (remaining >= b) {
      count <- count + (remaining %/% b)
      remaining <- remaining %% b
    }
  }

  return(count)
}
```