RWorksheet_Freires#4a

2024-10-23

1. The table below shows the data about shoe size and height. Create a data frame.

```
shoe_size <- c(6.5, 9.0, 8.5, 8.5, 10.5, 7.0, 9.5, 9.0, 13.0, 7.5, 10.5, 8.5, 12.0, 10.5, 13.0, 11.5, 8

height <- c(66.0, 68.0, 64.5, 65.0, 70.0, 64.0, 70.0, 71.0, 72.0, 64.0, 74.5, 67.0, 71.0, 71.0, 77.0, 7

gender <- c("F", "F", "F", "F", "M", "F", "F", "F", "M", "F",  "M", "F",  "M", "M", "M", "M", "F", "F",

data_table <- data.frame(Shoe_size = shoe_size, Height = height, Gender = gender)
print(data_table)
```

```
##    Shoe_size Height Gender
## 1        6.5   66.0      F
## 2        9.0   68.0      F
## 3        8.5   64.5      F
## 4        8.5   65.0      F
## 5       10.5   70.0      M
## 6        7.0   64.0      F
## 7        9.5   70.0      F
## 8        9.0   71.0      F
## 9       13.0   72.0      M
## 10       7.5   64.0      F
## 11      10.5   74.5      M
## 12       8.5   67.0      F
## 13      12.0   71.0      M
## 14      10.5   71.0      M
## 15      13.0   77.0      M
## 16      11.5   72.0      M
## 17       8.5   59.0      F
## 18       5.0   62.0      F
## 19      10.0   72.0      M
## 20       6.5   66.0      F
## 21       7.5   64.0      F
## 22       8.5   67.0      M
## 23      10.5   73.0      M
## 24       8.5   69.0      F
## 25      10.5   72.0      M
## 26      11.0   70.0      M
## 27       9.0   69.0      M
## 28      13.0   70.0      M
```

a. Describe the data.

- The data shows the table of Shoe sizes and Height of Male and Female genders.

b. Create a subset by males and females with their corresponding shoe size and height. What its result? Show the R scripts.

```r
males <- subset(data_table, Gender == "M", select = c(Shoe_size, Height))
print(males)
```

```
##    Shoe_size Height
## 5       10.5   70.0
## 9       13.0   72.0
## 11      10.5   74.5
## 13      12.0   71.0
## 14      10.5   71.0
## 15      13.0   77.0
## 16      11.5   72.0
## 19      10.0   72.0
## 22       8.5   67.0
## 23      10.5   73.0
## 25      10.5   72.0
## 26      11.0   70.0
## 27       9.0   69.0
## 28      13.0   70.0
```

```r
females <- subset(data_table, Gender = "F", select = c(Shoe_size, Height))
```

```
## Warning: In subset.data.frame(data_table, Gender = "F", select = c(Shoe_size,
##     Height)) :
##  extra argument 'Gender' will be disregarded
```

```r
print(females)
```

```
##    Shoe_size Height
## 1        6.5   66.0
## 2        9.0   68.0
## 3        8.5   64.5
## 4        8.5   65.0
## 5       10.5   70.0
## 6        7.0   64.0
## 7        9.5   70.0
## 8        9.0   71.0
## 9       13.0   72.0
## 10       7.5   64.0
## 11      10.5   74.5
## 12       8.5   67.0
## 13      12.0   71.0
## 14      10.5   71.0
## 15      13.0   77.0
## 16      11.5   72.0
## 17       8.5   59.0
## 18       5.0   62.0
## 19      10.0   72.0
## 20       6.5   66.0
## 21       7.5   64.0
## 22       8.5   67.0
## 23      10.5   73.0
## 24       8.5   69.0
## 25      10.5   72.0
## 26      11.0   70.0
## 27       9.0   69.0
## 28      13.0   70.0
```

C. Find the mean of shoe size and height of the respondents. Write the R scripts and its result.

```
mean_shoesize <- mean(shoe_size)
print(mean_shoesize)
```

```
## [1] 9.410714
```

```
mean_height <- mean(height)
print(mean_height)
```

```
## [1] 68.57143
```

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

- In my conclusion, there is a relationship because the data has shown that most tall people have bigger shoe sizes.

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. Consider data consisting of the names of months:

“March”,“April”,“January”,“November”,“January”, “September”,“October”,“September”,“November”,“August”, “January”,“November”,“November”,“February”,“May”,“August”, “July”,“December”,“August”,“August”,“September”,“Novem
April”

```
months_vector <- c("March","April","January","November","January",
"September","October","September","November","August", "January","November","November","February","May"
print(months_vector)
```

```
##  [1] "March"     "April"     "January"   "November"  "January"   "September"
##  [7] "October"   "September" "November"  "August"    "January"   "November"
## [13] "November"  "February"  "May"       "August"    "July"      "December"
## [19] "August"    "August"    "September" "November"  "February"  "April"
```

```
factor_months_vector <- factor(months_vector)
print(factor_months_vector)
```

```
##  [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
```

3. Then check the summary() of the months_vector and factor_months_vector. | Interpret the results of both vectors. Are they both equally useful in this case?

```
summary(months_vector)
```

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

```
summary(factor_months_vector)
```

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

- My interpretation of the results of months vector is that it shows the length, class, and mode functions. wherein the length is 24, the class is character and the mode is also a character. The results of the factor months vector is showing each month and their levels. Both summary are useful in this case as it has its own function.

4. Create a vector and factor for the table below.

- Note: Apply the factor function with required order of the level. new_order_data <- factor(factor_data,levels = c("East","West","North")) print(new_order_data)

```
direction <- c("East", "West", "North")

frequency <- c(1,4,3)

data <- data.frame(direction, frequency)
print(data)
```

```
##   direction frequency
## 1      East         1
## 2      West         4
## 3     North         3
```

```
new_data <- factor(direction, levels = c("East", "West", "North"))
print(new_data)
```

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

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

a. Import the excel file into the Environment Pane using read.table() function. Write the code.

```
library(readxl)
excelData <- read.table("/cloud/project/Worksheet#4/import_march.csv", header = TRUE, sep = ",")
```

b. View the dataset. Write the R scripts and its result.

```
print(excelData)
```

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

Using Conditional Statements (IF-ELSE)

6. Full Search

Exhaustive search is a methodology for finding an answer by exploring all possible cases. When trying to find a desired number in a set of given numbers, the method of finding the corresponding number by checking all elements in the set one by one can be called an exhaustive search. Implement an exhaustive search function that meets the input/output conditions below.

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 <- readline(prompt = "Enter a number from 1 to 50: ")
```

```
## Enter a number from 1 to 50:
```

```
if(num < 1 || num > 50) {
  paste("The number selected is beyond the range of 1 to 50")
}else if(num == 20) {
  print("TRUE")
} else{
  print(num)
}
```

```
## [1] "The number selected is beyond the range of 1 to 50"
```

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 <- function() {
  price <- as.numeric(readline(prompt = "Price of snack: "))

  if (is.na(price)) {
    print("Enter minimum number of bills needed to purchase a snack.")
    return()
  }
  paste("price of snack is ", price)

  if (price < 50) {
    print("Enter minimum number of bills needed to purchase a snack.")
  } else if (price < 100) {
    print("Minimum number of bills needed to purchase a snack is 50.")
  } else if (price < 200) {
    print("Minimum number of bills needed to purchase a snack is 100.")
  } else if (price < 500) {
    print("Minimum number of bills needed to purchase a snack is 200.")
  } else if (price < 1000) {
    print("Minimum number of bills needed to purchase a snack is 500.")
  } else {
    print("Minimum number of bills needed to purchase a snack is 1000.")
  }
}

min()
```

```
## Price of snack:
## [1] "Enter minimum number of bills needed to purchase a snack."
```

```
## NULL
```

8. The following is each student's math score for one semester. Based on this, answer the following questions.

Name Grade1 Grade2 Grade3 Grade4 Annie 85 65 85 100 Thea 65 75 90 90 Steve 75 55 80 85 Hanna 95 75 100 90

a. Create a dataframe from the above table. Write the R codes and its output.

```
name <- c("Annie", "Thea", "Steve", "Hanna")
Grade1 <- c(85, 65, 79, 95)
Grade2 <- c(65, 75, 55, 75)
Grade3 <- c(85, 90, 80, 100)
Grade4 <- c(100, 90, 85, 90)

data <- data.frame(name, Grade1, Grade2, Grade3, Grade4)
print(data)
```

```
##    name Grade1 Grade2 Grade3 Grade4
## 1 Annie     85     65     85    100
## 2  Thea     65     75     90     90
## 3 Steve     79     55     80     85
## 4 Hanna     95     75    100     90
```

b. Without using the rowMean function, output the average score of students whose average math score over 90 points during the semester. write R code and its output.

Example Output: Annie’s average grade this semester is 88.75.

```
row1 <- mean(as.numeric(data[1, 2:5]))
row2 <- mean(as.numeric(data[2, 2:5]))
row3 <- mean(as.numeric(data[3, 2:5]))
row4 <- mean(as.numeric(data[4, 2:5]))

for (i in 1:nrow(data)) {
    if (row1 >= 90) {
      average <- paste(data[1, 1], "’s average grade this semester is", round(row1, 2))

    } else if(row2 >= 90) {
      average <- paste(data[2, 1], "’s average grade this semester is", round(row2, 2))

    } else if(row3 >= 90) {
      average <- paste(data[3, 1], "’s average grade this semester is", round(row3, 2))

    } else if(row4 >= 90) {
      average <- paste(data[4, 1], "’s average grade this semester is", round(row4, 2))

    } else {
      average <- "No student got the average 90 and above"
    }

}
print(average)
```

```
## [1] "Hanna ’s average grade this semester is 90"
```

c. Without using the mean function, output as follows for the tests in which the average score was less than 80 out of 4 tests. Example output: The nth test was difficult.

```
scores <- c(0, 0, 0, 0)

for (i in 1:nrow(data)) {
  scores[1] <- scores[1] + data$Grade1[i]
  scores[2] <- scores[2] + data$Grade2[i]
```

```r
  scores[3] <- scores[3] + data$Grade3[i]
  scores[4] <- scores[4] + data$Grade4[i]
}

students <- nrow(data)

for (j in 1:length(scores)) {
  average <- scores[j] / students
  if (average < 80) {
    cat("The", j, "th test was difficult.\n")
  }
}
```

```
## The 2 th test was difficult.
```

d. Without using the max function, output as follows for students whose highest score for a semester exceeds 90 points. Example Output: Annie's highest grade this semester is 95.

```r
for (i in 1:nrow(data)) {
  grades <- as.numeric(data[i, 2:5])


  highest_grade <- grades[1]
  for (grade in grades) {
    if (grade > highest_grade) {
      highest_grade <- grade
    }
  }

  if (highest_grade > 90) {
    cat(data$name[i], "'s highest grade this semester is", highest_grade, ".\n")
  }
}
```

```
## Annie 's highest grade this semester is 100 .
## Hanna 's highest grade this semester is 100 .
```