RWorksheet_Benedicto#4a.R

Miguel F. Josue Jr

2024-10-14

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

```
data <- 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, 
  Gender = c("F", "F", "F", "F", "M", "F", "F", "F", "M", "F", "M", "F", "M", "M", "M","M", "F", "F", "
)
data
```

```
##    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 table provides a summary of data about individuals' shoe size, height, and gender. Whereas shoe size and height are in a numeric data types since they are measurements while the gender(F and M) correspond to the initial characters of male and female. The table implies that there is a correlation between these variables.

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

- The R scripts displays the gender-based shoe size and height data, through the “select” function we are able to only view only the Shoe Size and Height, this is used in order to avoid redundancy.

```
male_data <- subset(data, Gender == "M",select = c(Shoe_Size, Height))

male_data
```

```
##    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
```

```
female_data <- subset(data, Gender == "F", select = c(Shoe_Size, Height))

female_data
```

```
##    Shoe_Size Height
## 1        6.5   66.0
## 2        9.0   68.0
## 3        8.5   64.5
## 4        8.5   65.0
## 6        7.0   64.0
## 7        9.5   70.0
## 8        9.0   71.0
## 10       7.5   64.0
## 12       8.5   67.0
## 17       8.5   59.0
## 18       5.0   62.0
## 20       6.5   66.0
## 21       7.5   64.0
## 24       8.5   69.0
```

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

```
mean_shoe_size <- mean(data$Shoe_Size)
mean_shoe_size
```

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

```r
mean_height <- mean(data$Height)
mean_height
```

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

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

- There is indeed a relationship between shoe size and height, as the height increases the shoe size also increase relatively. It is also observed in the data that men have a significant gap in both said variables compared to women.

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.

```r
months_vector <- 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_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
```

```r
levels(factor_months_vector)
```

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

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? The summary(month_vector) presents the length and other generic properties of the vector like Class and mode, while the summary(factor_months_vector) counts the occurrences of each month . Both these summaries provide significant data, one shows the structure of the vector while the other is more on classifying categorical data.

```r
summary(months_vector)
```

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

```r
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
```

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

```r
direction_vector <- rep(c("East", "West", "North"),c(1,4,3))

factor_data <- factor(direction_vector)

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
```

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

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

```
##   Students Strategy.1 Strategy.2 Strategy3
## 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
```

6. 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.

```r
#Full Search
exhaustive_search <- function(input_number) {
  if (is.na(input_number)||input_number < 1 || input_number > 50) {
    return("The number selected is beyond the range of 1 to 50")
  } else if (input_number == 20) {
    return("TRUE")
  } else {
    return(paste("The selected number is:", input_number))
  }
}

input_number <- as.integer(readline(prompt = "Enter a number between 1 and 50: "))
```

```
## Enter a number between 1 and 50:
```

```r
result <- exhaustive_search(input_number)
print(result)
```

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

7. 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.

```r
#Change
min_bills <- function(price) {
  available_bills <- c(1000, 500, 200, 100, 50)
  total_bills <- 0

  for (bill in available_bills) {
    if (is.na(price) || is.na(bill)||price >= bill) {
      count <- floor(price / bill)
      price <- price - count * bill
      total_bills <- total_bills + count
    }
  }

  return(total_bills)
}

price <- as.numeric(readline(prompt = "Enter the price of the snack: "))
```

```
## Enter the price of the snack:
```

```r
print(paste("Minimum number of bills needed to purchase a snack:", min_bills(price)))
```

```
## [1] "Minimum number of bills needed to purchase a snack: NA"
```

8. The following is each student's math score for one semester. Based on this, answer the following questions. A. Create a dataframe from the above table. Write the R codes and its output.

```r
grades <- data.frame(
  Name = c("Annie", "Thea", "Steve", "Hanna"),
  Grade1 = c(85, 65, 75, 95),
  Grade2 = c(65, 75, 55, 75),
  Grade3 = c(85, 90, 80, 100),
  Grade4 = c(100, 90, 85, 90)
)

grades
```

```
##    Name Grade1 Grade2 Grade3 Grade4
## 1 Annie     85     65     85    100
## 2  Thea     65     75     90     90
## 3 Steve     75     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.

```
for (i in 1:nrow(grades)) {
  total_score <- sum(grades[i, 2:5])
  avg_score <- total_score / 4
  if (avg_score > 90) {
    print(paste(grades$Name[i], "'s average grade this semester is", avg_score))
  }
}
#NO average score OVER 90 (highest is flat 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.

```
for (j in 2:5) {
  total_score <- sum(grades[, j])
  avg_score <- total_score / nrow(grades)
  if (avg_score < 80) {
    test_num <- j - 1
    print(paste("The Grade",test_num, "test was difficult."))
  }
}
```

```
## [1] "The Grade 2 test was difficult."
```

D.Without using the max function, output as follows for students whose highest score for a semester exceeds 90 points.

```
for (i in 1:nrow(grades)) {
  highest_score <- grades[i, 2]
  for (j in 3:5) {
    if (grades[i, j] > highest_score) {
      highest_score <- grades[i, j]
    }
  }
  if (highest_score > 90) {
    print(paste(grades$Name[i], "'s highest grade this semester is", highest_score))
  }
}
```

```
## [1] "Annie 's highest grade this semester is 100"
## [1] "Hanna 's highest grade this semester is 100"
```