Rworksheet_Barrientos#4A

Barrientos, Milfrance D.

10/23/2024

```r
#1. The table below shows the data about shoe size and height. Create a data frame.
household_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, 8.5, 5.0, 10.0, 6.5, 7.5,
8.5, 10.5, 8.5, 10.5, 11.0, 9.0, 13.0),
  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, 72.0, 59.0, 62.0, 72.0, 66.0, 64.0,
67.0, 73.0, 69.0, 72.0, 70.0, 69.0, 70.0),
  Gender = c("F","F","F","F","M","F","F","F","M",
"F","M","F","M","M","M","M","F","F","M",
"F","F","M","M","F","M","M","M","M")
)
household_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
```

```r
#a. Describe the data
str(household_data)
```

```
## 'data.frame':    28 obs. of  3 variables:
##  $ Shoe_size: num  6.5 9 8.5 8.5 10.5 7 9.5 9 13 7.5 ...
##  $ Height   : num  66 68 64.5 65 70 64 70 71 72 64 ...
##  $ Gender   : chr  "F" "F" "F" "F" ...
```

```r
summary(household_data)
```

```
##    Shoe_size          Height         Gender         
##  Min.   : 5.000   Min.   :59.00   Length:28         
##  1st Qu.: 8.500   1st Qu.:65.75   Class :character  
##  Median : 9.000   Median :69.50   Mode  :character  
##  Mean   : 9.411   Mean   :68.57                     
##  3rd Qu.:10.500   3rd Qu.:71.25                     
##  Max.   :13.000   Max.   :77.00
```

```r
#b.Create a subset by males and females with their corresponding shoe size and height.
# What its result? Show the R scripts.
male <- subset(household_data, Gender == "M", select = c(Shoe_size, Height))

fem <- subset(household_data, Gender == "F", select = c(Shoe_size, Height))

male
```

```
##    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
fem
```

```
##    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
```

```r
#c.Find the mean of shoe size and height of the respondents. Write the R scripts and its result.
mean(household_data$Shoe_size)
```

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

```r
mean(household_data$Height)
```

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

```r
#d. Is there a relationship between shoe size and height? Why?
# yes, because the taller individuals tend to have larger shoe sizes.
```

```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")
months
```

```
##  [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"
```

```r
factor_months_vector <- factor(months)

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"
```

```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_months_vector)
```

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

```r
# the character vector is not as useful as the factor vector because the factor vector provides
# the number of occurrences of each level allowing for easier interpretation and understanding of
# the data.
```

```r
# 4. Create a vector and factor for the table below.
directions <- c("East", "West", "North")
frequencies <- c(1, 4, 3)
factor_data <- factor(directions, levels = c("East", "West", "North"))
print("Original Factor Data:")
```

```
## [1] "Original Factor Data:"
```

```r
print(factor_data)
```

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

```r
new_order_data <- factor(factor_data, levels = c("East", "West", "North"))
print("New Ordered Factor Data:")
```

```
## [1] "New Ordered Factor Data:"
```

```r
print(new_order_data)
```

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

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


library(readxl)

data <- read_excel("C:/PROJ/import_march.xlsx")

data
```

```
## # A tibble: 6 x 4
##   Students Strategy1 Strategy2 Strategy3
##   <chr>        <dbl>     <dbl>     <dbl>
## 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
```

```r
write.csv(data, file = "C:/PROJ/import_march.csv", row.names = FALSE)
# a. Import the excel file into the Environment Pane using read.table() function.
# Write the code.
read.table("C:/PROJ/import_march.csv", header = TRUE, sep = ",")
```

```
##   Students Strategy1 Strategy2 Strategy3
## 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
```

```r
#b. View the dataset. Write the R scripts and its result.
View(data)
```

Using Conditional Statements (IF-ELSE)

```r
# 6.

#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.

random_num <- function() {
    user_input <- as.numeric(readline(prompt = "Enter a number: "))
    if (is.na(user_input)) {
        print("Invalid input! Please enter a numeric value.")
```

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

```
## Enter a number:
## [1] "Invalid input! Please enter a numeric value."
```

```r
# 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 pes

#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) {

  if (price %% 50 != 0) {
    stop("The price must be a random number divisible by 50.")
  }


  denominations <- c(200, 100, 50)


  bill_count <- 0


  for (denom in denominations) {
    if (price >= denom) {

      bill_count <- bill_count + (price %/% denom)

      price <- price %% denom
    }
  }

  return(bill_count)
}


snack_price <- 50
result <- min_bills(snack_price)
cat("Minimum number of bills needed:", result, "\n")
```

```
## Minimum number of bills needed: 1
```

```r
#8.

students <- 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)
)
students
```

```
##    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
```

```r
# 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.

average_scores <- numeric(nrow(students))
for (i in 1:nrow(students)) {

  average_scores[i] <- sum(students[i, -1]) / (ncol(students) - 1)
}

students$AverageScore <- average_scores

for (i in 1:nrow(students)) {
  if (students$AverageScore[i] > 90) {
    cat(students$Name[i], "'s average grade this semester is", round(students$AverageScore[i], 2), ".\n
  }
}

#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.

average_scores_tests <- numeric(ncol(students) - 1)  # Exclude the Name column

for (j in 2:ncol(students)) {
  total_score <- 0
  for (i in 1:nrow(students)) {
    total_score <- total_score + students[i, j]
  }
  average_scores_tests[j - 1] <- total_score / nrow(students)
}


for (k in 1:length(average_scores_tests)) {
  if (average_scores_tests[k] < 80) {
```

```r
    cat("The test", k, "was difficult.\n")
  }
}
```

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

```r
#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.

for (i in 1:nrow(students)) {
  highest_score <- students[i, 2]
  for (j in 3:ncol(students)) {
    if (students[i, j] > highest_score) {
      highest_score <- students[i, j]
    }
  }


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

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