RWorksheet_ESTOCE#4a.Rmd

John Rick Estoce

2025-11-03

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

a. Describe the data.

```
household <- data.frame(
  ShoeSize = 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, 10.5, 8.5, 11.0, 9.0, 13.0),
  Height = c(66.0, 68.0, 64.5, 65.0, 70.0, 64.0, 70.0, 73.0, 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, 73.0, 69.0, 70.0, 69.0, 70.0),
  Gender = c("F", "F", "F", "F", "F", "F", "F", "M", "M", "F",
             "F", "F", "M", "M", "M", "M", "F", "F", "F", "F",
             "F", "M", "M", "M", "M", "M")
)

household
```

```
##    ShoeSize 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      F
## 6       7.0   64.0      F
## 7       9.5   70.0      F
## 8       9.0   73.0      M
## 9      13.0   72.0      M
## 10      7.5   64.0      F
## 11     10.5   74.5      F
## 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      F
## 20      6.5   66.0      F
## 21      7.5   64.0      F
## 22     10.5   73.0      M
## 23      8.5   69.0      M
## 24     11.0   70.0      M
## 25      9.0   69.0      M
```

```
## 26     13.0   70.0      M
```

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

```r
# Create data frame

# Subset by gender
male <- subset(household, Gender == "M", select = c(ShoeSize, Height))
female <- subset(household, Gender == "F", select = c(ShoeSize, Height))

# Display subsets
male
```

```
##    ShoeSize Height
## 8       9.0     73
## 9      13.0     72
## 13     12.0     71
## 14     10.5     71
## 15     13.0     77
## 16     11.5     72
## 22     10.5     73
## 23      8.5     69
## 24     11.0     70
## 25      9.0     69
## 26     13.0     70
```

```r
female
```

```
##    ShoeSize 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
## 10      7.5   64.0
## 11     10.5   74.5
## 12      8.5   67.0
## 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
```

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

```r
# Mean of all respondents
mean_shoesize <- mean(household$ShoeSize)
mean_height <- mean(household$Height)

mean_shoesize
```

```
## [1] 9.403846
```

```r
mean_height
```

```
## [1] 68.57692
```

d. Is there a relationship between shoe size and height? Why? Yes — there appears to be a positive relationship between shoe size and height. From observation:

Taller people tend to have larger shoe sizes.

Males generally have both larger shoe sizes and greater height than females.

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_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
```

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?

```
# Create the character vector
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")

# Convert to factor
factor_months_vector <- factor(months)

# Check summaries
summary(months)
```

```
##    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
```

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(rep("East", 1), rep("West", 4), rep("North", 3))
factor_data <- factor(direction, levels = c("East", "West", "North"))
print(factor_data)
```

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

```
## Levels: East West North
```

```r
summary(factor_data)
```

```
##  East  West North
##     1     4     3
```

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

```r
library(readxl)
import_march <- read_excel("import_march.xlsx")
print(import_march)
```

```
## # A tibble: 6 x 4
##   Students `Strategy 1` `Strategy 2` `Strategy 3`
##   <chr>           <dbl>        <dbl>        <dbl>
## 1 Male                8           10            8
## 2 <NA>                4            8            6
## 3 <NA>                0            6            4
## 4 Female             14            4           15
## 5 <NA>               10            2           12
## 6 <NA>                6            6            9
```

6. Full Search

```r
num <- 15
numbers <- 1:50
found <- FALSE

for (i in numbers) {
  if (i == num) {
    found <- TRUE
    break
  }
}

if (!found) {
  cat("The number selected is beyond the range of 1 to 50\n")
} else if (num == 20) {
  cat("TRUE\n")
} else {
  cat("You selected:", num, "\n")
}
```

```
## You selected: 15
```

7 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.

```r
minimum_bills <- function(price) {

  bills <- c(1000, 500, 200, 100, 50)

  remaining <- price
  count <- 0
```

```r
  for (bill in bills) {
    if (remaining >= bill) {
      num <- remaining %/% bill
      count <- count + num
      remaining <- remaining - num * bill
    }
  }

  cat("Minimum number of bills needed:", count, "\n")
}

price_of_snack <- 1350
minimum_bills(price_of_snack)
```

```
## Minimum number of bills needed: 4
```

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

a. Create a data frame 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. Example Output: Annie's average grade this semester is 88.75.

```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)
)

# Loop through each student to calculate average
for (i in 1:nrow(grades)) {
  total <- grades$Grade1[i] + grades$Grade2[i] + grades$Grade3[i] + grades$Grade4[i]
  avg <- total / 4
  if (avg >= 90) {
    cat(grades$Name[i], "'s average grade this semester is", avg, "\n")
```

```
  }
}
```

```
## 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.

```
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)
)

for (i in 2:ncol(grades)) {
  total <- sum(grades[, i])
  avg <- total / nrow(grades)

  if (avg < 80) {
    cat("The", i - 1, "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.

```
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)
)

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

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