RWorksheet_Moquete#4a.Rmd

Renz Moquete

2025-11-11

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', '

# Combine the vectors into a data frame
shoe_height_data <- data.frame(
  Shoe_Size = shoe_size,
  Height = height,
  Gender = gender
)
```

a. Describe the data

The data shows the an organize table of data such as shoe size, height and gender.

```
str(shoe_height_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" ...
```

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

What its result? Show the R scripts.

```
males_data <- subset(shoe_height_data, Gender == 'M')

# Create a subset for Females
females_data <- subset(shoe_height_data, Gender == 'F')

# Print the results to the console
print("Male Subset:")
```

```
## [1] "Male Subset:"
```

```
print(males_data)
```

```
##    Shoe_Size Height Gender
## 5       10.5   70.0      M
## 9       13.0   72.0      M
## 11      10.5   74.5      M
## 13      12.0   71.0      M
```

```
## 14      10.5   71.0      M
## 15      13.0   77.0      M
## 16      11.5   72.0      M
## 19      10.0   72.0      M
## 22       8.5   67.0      M
## 23      10.5   73.0      M
## 25      10.5   72.0      M
## 26      11.0   70.0      M
## 27       9.0   69.0      M
## 28      13.0   70.0      M
```

```
print("Female Subset:")
```

```
## [1] "Female Subset:"
```

```
print(females_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
## 6        7.0   64.0      F
## 7        9.5   70.0      F
## 8        9.0   71.0      F
## 10       7.5   64.0      F
## 12       8.5   67.0      F
## 17       8.5   59.0      F
## 18       5.0   62.0      F
## 20       6.5   66.0      F
## 21       7.5   64.0      F
## 24       8.5   69.0      F
```

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

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

# Calculate the mean of the 'Height' column
mean_height <- mean(shoe_height_data$Height)

# Print the results to the console
mean_height
```

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

```
mean_shoe_size
```

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

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

Answer: Yes, there is a strong relationship between shoe size and height.

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

# Calculate the mean of the 'Height' column
mean_height <- mean(shoe_height_data$Height)
```

```r
# Print the results to the console
mean_height
```

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

```r
mean_shoe_size
```

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

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
# 1. Create a character vector with the names of the months
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")

# 2. Convert the character vector to a factor
factor_months_vector <- factor(months_vector)

# 3. Print the resulting factor to the console
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?

```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
```

Interpretation: The summary of a character vector is purely structural. It tells you what the data is (text) and how much of it there is (24 items), but it provides zero insight into the content of the data. It doesn't know that "January" appeared multiple times or how many unique months there are. For data analysis, this summary is not very useful.

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

```r
direction <- c("East", "West", "North")   # Vector of directions
frequency <- c(1, 4, 3)                   # Vector of frequencies

# Create factor with required order of levels
direction_factor <- factor(direction, levels = c("East", "West", "North"))

print(direction)
```

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

```r
print(frequency)
```

```
## [1] 1 4 3
```

```r
print(direction_factor)
```

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

6. Full Search

a. Create an R Program that allows the User to randomly select numbers from 1 to 50.

```r
number <- as.numeric(readline(prompt = "Select numbers 1-50: "))
```

```
## Select numbers 1-50:
```

```r
number = 20

if (number > 50) {
  print("The number selected is beyond the range of 1 to 50")
} else if (number == 20) {
  print("True")
} else {
  print(number)
}
```

```
## [1] "True"
```

7. Change

a. Write a function that prints the minimum number of bills that must be paid, given the price of the snack.

```r
min_bills <- function(price) {
  bills <- c(1000, 500, 200, 100, 50)
  count <- 0

  for (bill in bills) {
    n <- price %/% bill   #
    count <- count + n
    price <- price %% bill
  }

  return(count)
}

min_bills(1350)
```

```
## [1] 4
```

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.

```
math_scores <- 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)
)

print(math_scores)
```

```
##    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.

```
average_scores <- (math_scores$Grade1 + math_scores$Grade2 + math_scores$Grade3 + math_scores$Grade4) /

average_scores
```

```
## [1] 83.75 80.00 73.75 90.00
```

```
high_achievers <- math_scores[average_scores > 90, ]

high_achievers
```

```
## [1] Name   Grade1 Grade2 Grade3 Grade4
## <0 rows> (or 0-length row.names)
```

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.

```
average_scores <- (math_scores$Grade1 + math_scores$Grade2 + math_scores$Grade3 + math_scores$Grade4) /

average_scores
```

```
## [1] 83.75 80.00 73.75 90.00
```

```
low_scores <- math_scores[average_scores < 80, ]

low_scores
```

```
##    Name Grade1 Grade2 Grade3 Grade4
## 3 Steve     75     55     80     85
```

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

```r
for (i in 1:nrow(math_scores)) {

  grades <- c(math_scores$Grade1[i], math_scores$Grade2[i], math_scores$Grade3[i], math_scores$Grade4[i]


  highest <- grades[1]
  for (g in grades) {
    if (g > highest) {
      highest <- g
    }
  }

  if (highest > 90) {
    cat(math_scores$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
```