Worksheet-4a in R

Riza Angelique Pelaez

2024-10-14

```r
# 1. Shoe size, height, and gender data

shoe_size <- c(6.5, 9.0, 8.5, 8.5, 10.5, 7.0, 9.0, 9.5, 10.0, 7.5,
               8.0, 8.5, 10.0, 8.5, 9.0, 12.0, 8.5, 13.0, 11.5, 8.5,
               5.0, 10.0, 6.5, 8.5, 10.5, 8.5, 10.0, 11.0, 9.0, 13.0)

height <- c(66.0, 68.0, 64.5, 65.0, 70.0, 64.0, 71.0, 70.0, 70.5, 71.0,
            67.0, 65.0, 72.0, 66.0, 67.0, 74.5, 69.0, 77.0, 72.0, 59.0,
            62.0, 72.0, 64.0, 67.0, 69.0, 70.0, 75.0, 70.0, 69.0, 70.0)

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

# a. Describe the dataDescribe the data
df <- data.frame(Shoe_Size = shoe_size, Height = height, Gender = gender)
summary(df)
```

```
##    Shoe_Size          Height         Gender
##  Min.   : 5.000   Min.   :59.00   Length:30
##  1st Qu.: 8.500   1st Qu.:66.00   Class :character
##  Median : 9.000   Median :69.00   Mode  :character
##  Mean   : 9.183   Mean   :68.55
##  3rd Qu.:10.000   3rd Qu.:70.88
##  Max.   :13.000   Max.   :77.00
```

```r
# b. A subsets for males and females with their corresponding shoe size and height
males <- subset(df, Gender == "M")
females <- subset(df, Gender == "F")

males
```

```
##    Shoe_Size Height Gender
## 5       10.5   70.0      M
## 8        9.5   70.0      M
## 10       7.5   71.0      M
## 11       8.0   67.0      M
## 13      10.0   72.0      M
## 14       8.5   66.0      M
## 15       9.0   67.0      M
## 16      12.0   74.5      M
## 18      13.0   77.0      M
## 19      11.5   72.0      M
## 22      10.0   72.0      M
```

```
## 24       8.5   67.0      M
## 26       8.5   70.0      M
## 27      10.0   75.0      M
## 28      11.0   70.0      M
## 29       9.0   69.0      M
## 30      13.0   70.0      M
```

```
females
```

```
##    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        9.0   71.0      F
## 9       10.0   70.5      F
## 12       8.5   65.0      F
## 17       8.5   69.0      F
## 20       8.5   59.0      F
## 21       5.0   62.0      F
## 23       6.5   64.0      F
## 25      10.5   69.0      F
```

```
# c. Find the mean of shoe size and height of the respondents
mean_shoe_size <- mean(df$Shoe_Size)
mean_height <- mean(df$Height)

mean_shoe_size
```

```
## [1] 9.183333
```

```
mean_height
```

```
## [1] 68.55
```

```
# d. Is there a relationship between shoe size and height?
# A positive correlation between shoe size and height suggests that taller individuals tend to have lar

correlation <- cor(shoe_size, height)
correlation
```

```
## [1] 0.7210568
```

```
# 2. Construct a character vector of months and convert it to a factor

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)
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
```

```
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
```

```
# 3. Create a vector and factor for the direction table and reorder it
direction <- c("East", "West", "North")
frequency <- c(1, 4, 3)
factor_direction <- factor(direction, levels = c("East", "West", "North"))
factor_direction
```

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

```
# 4. Creating a vector for Direction
direction <- c("East", "West", "North")
frequency <- c(1, 4, 3)
factor_direction <- factor(direction, levels = c("East", "West", "North"))

print(factor_direction)
```

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

```
print(frequency)
```

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

```
# 5. Import the excel file into the environment
#a. Import the excel file into the Environment Pane using read.table() function
dataset <- read.table("import_march.csv", header = TRUE, sep = ",")

#b. View the dataset
View(dataset)
```

```
# 6. Exhaustive search function
exhaustive_search <- function() {
  chosen_number <- as.integer(readline(prompt = "Choose a number between 1 and 50: "))

  if (is.na(chosen_number)) {
    print("Invalid input. Please enter a number.")
  } else if (chosen_number < 1 || chosen_number > 50) {
    print("The number selected is beyond the range of 1 to 50")
  } else if (chosen_number == 20) {
    print("TRUE")
  } else {
    print(paste("The selected number is:", chosen_number))
  }
}
```

```r
exhaustive_search()
```

```
## Choose a number between 1 and 50:
## [1] "Invalid input. Please enter a number."
```

```r
# 7. Minimum number of bills function
calculate_minimum_bills <- function(price) {
  if (price %% 50 != 0) {
    stop("The price must be divisible by 50.")
  }

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

  total_bills <- 0

  for (bill in bills) {
    if (price >= bill) {
      count <- price %/% bill  # Number of this bill
      total_bills <- total_bills + count  # Update total bills
      price <- price %% bill  # Remaining price to pay
    }
  }

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

# Test the function with a sample price
calculate_minimum_bills(1700)
```

```
## Minimum number of bills needed: 3
```

```r
# 8. Math scores dataframe
names <- 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)

df_grades <- data.frame(Name = names, Grade1 = grade1, Grade2 = grade2, Grade3 = grade3, Grade4 = grade

# a. Create a dataframe
df_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
```

```r
# b. Average score of students with an average score over 90
for(i in 1:nrow(df_grades)) {
  avg_score <- (df_grades$Grade1[i] + df_grades$Grade2[i] + df_grades$Grade3[i] + df_grades$Grade4[i]) /
  if(avg_score > 90) {
    print(paste(df_grades$Name[i], "'s average grade this semester is", avg_score))
  }
}
```

```r
# c. Tests where the average score is less than 80
for(j in 2:5) {
  avg_test_score <- mean(df_grades[, j])
  if(avg_test_score < 80) {
    print(paste("The", j-1, "th test was difficult."))
  }
}
```

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

```r
# d. Highest score above 90 for students
for(i in 1:nrow(df_grades)) {
  max_score <- max(df_grades$Grade1[i], df_grades$Grade2[i], df_grades$Grade3[i], df_grades$Grade4[i])
  if(max_score > 90) {
    print(paste(df_grades$Name[i], "'s highest grade this semester is", max_score))
  }
}
```

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