Worksheet#4a

Lorie Mae Tupaz BSIT 2B

2024-10-28

```
shoe_data <- 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, 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")
)

print(shoe_data)
```

```
##    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      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
```

```
#   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      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
```

```
#1.a

#Shoe Size ranges from 5.0 to 13.0, indicating a variety of sizes.

#Height varies from 59.0 to 77.0 inches, reflecting a diverse range of heights.

#Gender is categorized as "M" for male and "F" for female, allowing for comparisons between the two gro
```

```
#1.b
female_subset <- shoe_data[shoe_data$Gender == "F", ]

male_subset <- shoe_data[shoe_data$Gender == "M", ]

print(female_subset)
```

```
##    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
## 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
```

```
print(male_subset)
```

```
##    ShoeSize 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
```

```
#   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
#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


#   ShoeSize 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
```

```
#1.c
mean_shoe_size <- mean(shoe_data$ShoeSize)
mean_height <- mean(shoe_data$Height)

# Use paste() to concatenate the strings, avoiding special characters
paste("Mean Shoe Size:", mean_shoe_size)
```

```
## [1] "Mean Shoe Size: 9.41071428571429"
```

```
paste("Mean Height:", mean_height)
```

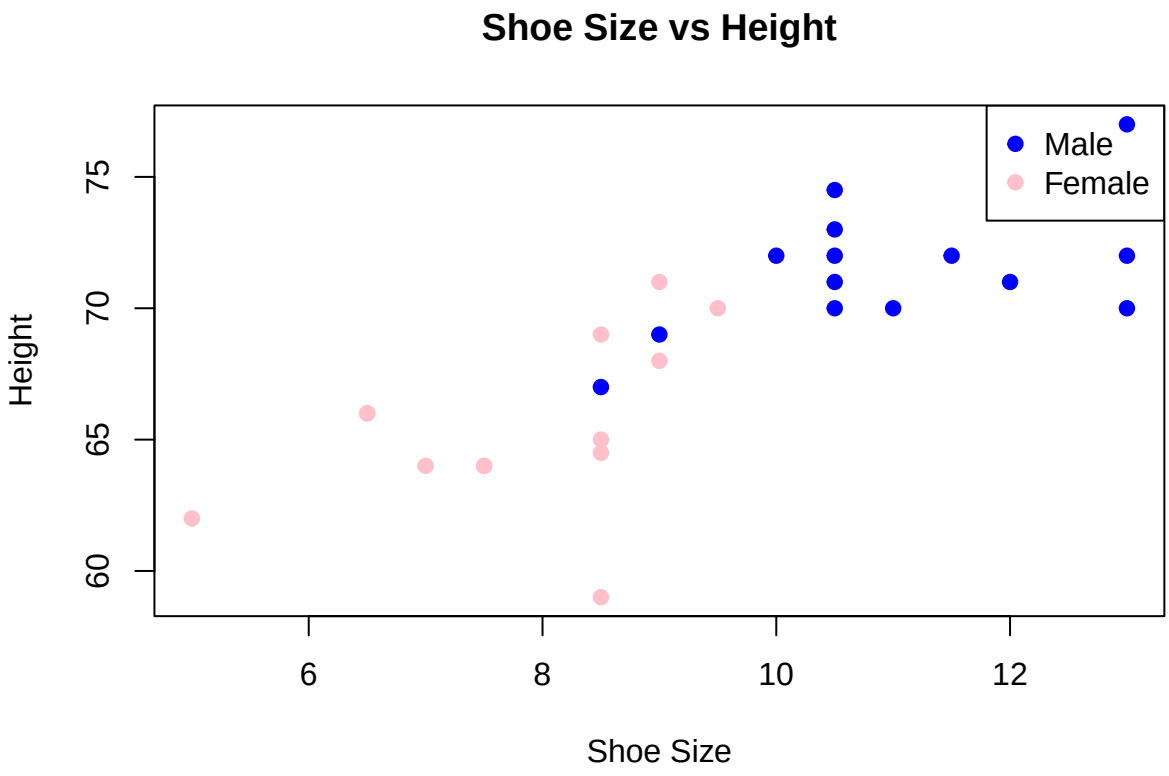
```
## [1] "Mean Height: 68.5714285714286"
```

```
#Mean Shoe Size: 9.410714
#Mean Height: 68.57143


#1.d
colors <- ifelse(shoe_data$Gender == "M", "blue", "pink")

plot(shoe_data$ShoeSize, shoe_data$Height,
     main = "Shoe Size vs Height",
     xlab = "Shoe Size",
     ylab = "Height",
     pch = 19, col = colors)

legend("topright", legend = c("Male", "Female"), col = c("blue", "pink"), pch = 19)
```




```
#Yes, there is a relationship between shoe size and height because taller individuals generally have la
```

```
#Factors

#2

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
```

```
#[1] March     April     January   November  January   September October   September November  August
#[12] November  November  February  May       August    July      December  August    August    Septemb
#[23] February  April
#Levels: April August December February January July March May November October September
```

```r
#3
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)

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
```

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

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


#No, they are not equally useful.

#months_vector shows all the months as they appear, including duplicates.

#factor_months_vector summarizes the unique months and their frequencies, making it easier to analyze t
```

```r
#4
factor_data <- c("East", "West", "North", "West", "West", "North", "West", "West")

new_order_data <- factor(factor_data, levels = c("East", "West", "North"))

print(new_order_data)
```

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

```r
summary(new_order_data)
```

```
##  East  West North
##     1     5     2
```

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

#East  West North
#    1     5     2
```

```
#5

#5.a

setwd("C:\\Users\\Client\\OneDrive\\Documents\\CS101\\RWorksheet#4\\RWorksheet_4a_Tupaz_files")

setwd("C:/Users/Client/OneDrive/Documents/CS101/RWorksheet#4/RWorksheet_4a_Tupaz_files")

# Then run the rest of your code
data <- read.table("import_march.csv", header = TRUE, sep = ",")
print(head(data))
```

```
##   Students Strategy..1 Strategy.2 Strategy.3
## 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
```

```
#  Students Strategy.1 Strategy.2 Strategy.3
#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
exhaustive_search <- function(user_input) {
  cat("You selected:", user_input, "\n")

  if (user_input < 1 || user_input > 50) {
    cat("The number selected is beyond the range of 1 to 50.\n")
  } else if (user_input == 20) {
    cat("TRUE\n")
  } else {
    cat("The input number is:", user_input, "\n")
  }
}

exhaustive_search(20)
```

```
## You selected: 20
## TRUE
```

```r
#7.
price_input <- 150  # Replace with any value you want to test

calculate_minimum_bills <- function(price) {
  denominations <- c(1000, 500, 200, 100, 50)

  bill_count <- 0

  if (price %% 50 != 0) {
    cat("Price must be a number divisible by 50.\n")
  } else {
    for (denom in denominations) {
      if (price >= denom) {
        count <- price %/% denom
        bill_count <- bill_count + count

        price <- price - (count * denom)
      }
    }

    if (bill_count > 0) {
      cat("Minimum number of bills needed:", bill_count, "\n")
    } else {
      cat("No bills needed.\n")
    }
  }
}

calculate_minimum_bills(price_input)
```

```
## Minimum number of bills needed: 2
```

```r
#8.a
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)

math_scores <- data.frame(Name = names, Grade1 = grade1, Grade2 = grade2, Grade3 = grade3, Grade4 = gra
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
```

```r
#8.b
average_scores <- numeric(length(names))

for (i in 1:length(names)) {
```

```
  average_scores[i] <- (math_scores$Grade1[i] + math_scores$Grade2[i] +
                        math_scores$Grade3[i] + math_scores$Grade4[i]) / 4
}

for (i in 1:length(names)) {
  cat(math_scores$Name[i], "'s average grade this semester is", round(average_scores[i], 2), "\n")
}
```

```
## Annie 's average grade this semester is 83.75
## Thea 's average grade this semester is 80
## Steve 's average grade this semester is 73.75
## Hanna 's average grade this semester is 90
```

```
for (i in 1:length(names)) {
  if (average_scores[i] > 90) {
    cat(math_scores$Name[i], "'s average grade this semester is", round(average_scores[i], 2), "\n")
  }
}

# 8.c
test_averages <- numeric(4)

for (j in 1:4) {
  test_averages[j] <- sum(math_scores[, j + 1]) / nrow(math_scores)
  cat("Average for", colnames(math_scores)[j + 1], "is", round(test_averages[j], 2), "\n")
}
```

```
## Average for Grade1 is 80
## Average for Grade2 is 67.5
## Average for Grade3 is 88.75
## Average for Grade4 is 91.25
```

```
#8.d
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
```

```
for (i in 1:nrow(grades)) {
  highest <- grades[i, 2]
  for (j in 3:5) {
```

```
    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
```