Regacho_Worksheet#4

```r
# 1a. Create data frame
shoe_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, 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, 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", "M", "M", "M")
)

# Describe the data
str(shoe_data)
```

```
## 'data.frame':    26 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(shoe_data)
```

```
##    Shoe_size          Height         Gender
##  Min.   : 5.000   Min.   :59.00   Length:26
##  1st Qu.: 8.500   1st Qu.:65.25   Class :character
##  Median : 9.000   Median :69.50   Mode  :character
##  Mean   : 9.404   Mean   :68.42
##  3rd Qu.:10.500   3rd Qu.:71.00
##  Max.   :13.000   Max.   :77.00
```

```r
## 1b. Create subsets by gender
males <- subset(shoe_data, Gender == "M", select = c(Shoe_size, Height))
females <- subset(shoe_data, Gender == "F", select = c(Shoe_size, Height))

print("Males subset:")
```

```
## [1] "Males subset:"
```

```r
print(males)
```

```
##    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
## 24      11.0   70.0
## 25       9.0   69.0
## 26      13.0   70.0
```

```
print("Females subset:")
```

```
## [1] "Females subset:"
```

```
print(females)
```

```
##    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
```

```
# 1b. Create subsets by gender
males <- subset(shoe_data, Gender == "M", select = c(Shoe_size, Height))
females <- subset(shoe_data, Gender == "F", select = c(Shoe_size, Height))

print("Males subset:")
```

```
## [1] "Males subset:"
```

```
print(males)
```

```
##    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
## 24      11.0   70.0
## 25       9.0   69.0
## 26      13.0   70.0
```

```
print("Females subset:")
```

```
## [1] "Females subset:"
```

```
print(females)
```

```
##    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
```

```
# 1c. Find mean of shoe size and height
mean_shoe <- mean(shoe_data$Shoe_size)
mean_height <- mean(shoe_data$Height)

cat("\nMean shoe size:", round(mean_shoe, 2), "\n")
```

```
##
## Mean shoe size: 9.4
```

```
cat("Mean height:", round(mean_height, 2), "\n")
```

```
## Mean height: 68.42
```

```
# Mean by gender
mean_shoe_male <- mean(males$Shoe_size)
mean_height_male <- mean(males$Height)
mean_shoe_female <- mean(females$Shoe_size)
mean_height_female <- mean(females$Height)

cat("\nMean shoe size - Males:", round(mean_shoe_male, 2), "\n")
```

```
##
## Mean shoe size - Males: 11
```

```r
cat("Mean height - Males:", round(mean_height_male, 2), "\n")
```

```
## Mean height - Males: 71.42
```

```r
cat("Mean shoe size - Females:", round(mean_shoe_female, 2), "\n")
```

```
## Mean shoe size - Females: 7.81
```

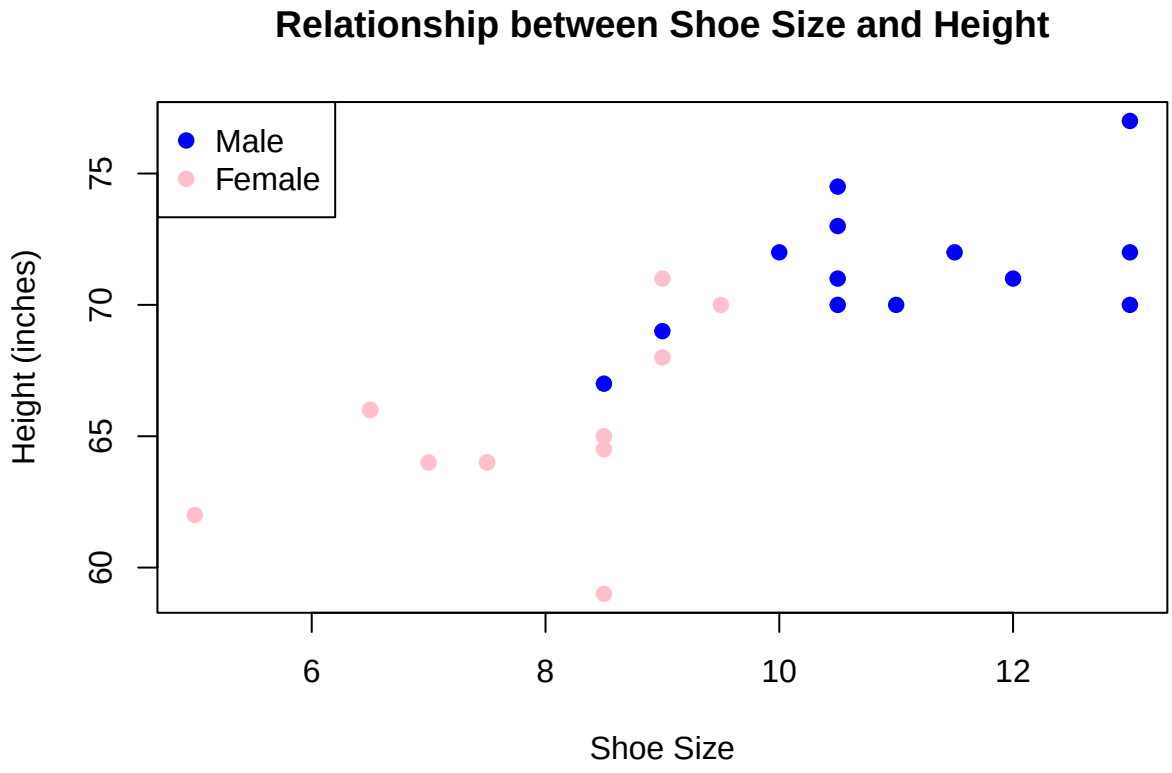
```r
cat("Mean height - Females:", round(mean_height_female, 2), "\n")
```

```
## Mean height - Females: 65.42
```

```r
# 1d. Check relationship
correlation <- cor(shoe_data$Shoe_size, shoe_data$Height)
cat("Correlation between shoe size and height:", correlation, "\n")
```

```
## Correlation between shoe size and height: 0.7795104
```

```r
# Scatter plot to visualize relationship
plot(shoe_data$Shoe_size, shoe_data$Height,
     xlab = "Shoe Size", ylab = "Height (inches)",
     main = "Relationship between Shoe Size and Height",
     col = ifelse(shoe_data$Gender == "M", "blue", "pink"),
     pch = 19)
legend("topleft", legend = c("Male", "Female"), col = c("blue", "pink"), pch = 19)
```



```r
# Linear model
lm_model <- lm(Height ~ Shoe_size, data = shoe_data)
cat("Linear model coefficient:", round(coef(lm_model)[2], 3), "\n")
```

```
## Linear model coefficient: 1.534
```

```r
# There is a strong positive relationship between shoe size and height, with a correlation coefficient
```

```r
# 2. Create a factor for months
months_vector <- c("March","April","January","November","January",
"September","October","September","November","August","January","November","November","February","May",

factor_months_vector <- factor(months_vector)
cat("Factor months vector:\n")
```

```
## Factor months vector:
```

```r
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
# 3. Check summary if both vectors
cat("\nSummary of months_vector (character):\n")
```

```
##
## Summary of months_vector (character):
```

```r
print(summary(months_vector))
```

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

```r
cat("\nSummary of factor_months_vector (character):\n")
```

```
##
## Summary of factor_months_vector (character):
```

```r
print(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 factor summmary is much more useful in this case because it provides the frequency count for each
```

```r
# 5a. First create and save the CSV file
# Create the data frame
strategy_data <- data.frame(
  Students = c("Male", "", "", "Female", "", ""),
  Strategy1 = c(8, 4, 0, 14, 10, 6),
  Strategy2 = c(10, 8, 6, 4, 2, 0),
  Strategy3 = c(8, 6, 4, 15, 12, 9)
)

# Save as CSV
write.csv(strategy_data, "import_march.csv", row.names = FALSE)

# 5a. Import the CSV file
imported_data <- read.table("import_march.csv", header = TRUE, sep = ",")

# 5b. View the dataset
cat("Imported dataset:\n")
```

```
## Imported dataset:
```

```r
print(imported_data)
```

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

```r
cat("\nStructure of imported data:\n")
```

```
##
## Structure of imported data:
```

```r
str(imported_data)
```

```
## 'data.frame':    6 obs. of  4 variables:
##  $ Students : chr  "Male" "" "" "Female" ...
##  $ Strategy1: int  8 4 0 14 10 6
##  $ Strategy2: int  10 8 6 4 2 0
##  $ Strategy3: int  8 6 4 15 12 9
```

```r
# 6. Full Search with conditional statements
full_search <- function() {
  cat("Enter a number from 1 to 50: ")
  user_input <- as.numeric(readline())
```

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

# Test function with different scenarios
cat("Testing full_search function:\n")
```

```
## Testing full_search function:
```

```
cat("If input is 20:", ifelse(20 == 20, "TRUE", "Input number: 20"), "\n")
```

```
## If input is 20: TRUE
```

```
cat("If input is 25: Input number: 25\n")
```

```
## If input is 25: Input number: 25
```

```
cat("If input is 55: The number selected is beyond the range of 1 to 50\n")
```

```
## If input is 55: The number selected is beyond the range of 1 to 50
```

```
cat("If input is 'abc': Invalid input. Please enter a number.\n")
```

```
## If input is 'abc': Invalid input. Please enter a number.
```

```
 result <- full_search()
```

```
## Enter a number from 1 to 50:
```

```
 print(result)
```

```
## [1] "Invalid input. Please enter a number."
```

```
# 7. Change calculator for bills
calculate_bills <- function(price) {
  if (price %% 50 != 0) {
    return("Price must be divisible by 50")
  }

  bills <- c(1000, 500, 200, 100, 50)
  count <- 0
```

```
  remaining <- price

  for (bill in bills) {
    if (remaining >= bill) {
      num_bills <- floor(remaining / bill)
      count <- count + num_bills
      remaining <- remaining - (num_bills * bill)
    }
  }

  return(count)
}

# Test the function with various prices
test_prices <- c(50, 100, 150, 200, 450, 500, 750, 1000, 1250, 1500)

cat("Testing bill calculator:\n")
```

```
## Testing bill calculator:
```

```
for (price in test_prices) {
  cat("For price", price, "pesos, minimum bills needed:", calculate_bills(price), "\n")
}
```

```
## For price 50 pesos, minimum bills needed: 1
## For price 100 pesos, minimum bills needed: 1
## For price 150 pesos, minimum bills needed: 2
## For price 200 pesos, minimum bills needed: 1
## For price 450 pesos, minimum bills needed: 3
## For price 500 pesos, minimum bills needed: 1
## For price 750 pesos, minimum bills needed: 3
## For price 1000 pesos, minimum bills needed: 1
## For price 1250 pesos, minimum bills needed: 3
## For price 1500 pesos, minimum bills needed: 2
```

```
# 8a. Create dataframe from student grades
student_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),
  stringsAsFactors = FALSE
)

cat("Student grades dataframe:\n")
```

```
## Student grades dataframe:
```

```
print(student_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
# 8b. Average score over 90
cat("\nStudents with average over 90:\n")
```

```
##
## Students with average over 90:
```

```r
for (i in 1:nrow(student_grades)) {
  avg_grade <- (student_grades$Grade1[i] + student_grades$Grade2[i] +
                student_grades$Grade3[i] + student_grades$Grade4[i]) / 4

  if (avg_grade > 90) {
    cat(student_grades$Name[i], "'s average grade this semester is", avg_grade, "\n")
  }
}
```

```r
# 8c. Tests with average less than 80
cat("\nTests with average less than 80:\n")
```

```
##
## Tests with average less than 80:
```

```r
test_averages <- c(
  sum(student_grades$Grade1) / nrow(student_grades),
  sum(student_grades$Grade2) / nrow(student_grades),
  sum(student_grades$Grade3) / nrow(student_grades),
  sum(student_grades$Grade4) / nrow(student_grades)
)

for (i in 1:length(test_averages)) {
  if (test_averages[i] < 80) {
    cat("The", i, "test was difficult.\n")
  }
}
```

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

```r
# 8d. Students with highest score exceeding 90
cat("\nStudents with highest score exceeding 90:\n")
```

```
##
## Students with highest score exceeding 90:
```

```r
for (i in 1:nrow(student_grades)) {
  highest_grade <- student_grades$Grade1[i]
  if (student_grades$Grade2[i] > highest_grade) highest_grade <- student_grades$Grade2[i]
  if (student_grades$Grade3[i] > highest_grade) highest_grade <- student_grades$Grade3[i]
  if (student_grades$Grade4[i] > highest_grade) highest_grade <- student_grades$Grade4[i]

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

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