

## RWorksheet\_Gerona#4

Mariel M. Geron

### 1a. Describe the Data

```
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, 10.5, 13.0, 11.5, 8.5),
  Height = c(66.0, 68.0, 64.5, 65.0, 70.0, 64.0, 70.0, 71.0, 72.0, 64.0, 74.75, 67.0, 71.0, 71.0, 77.0, 71.0, 66.0, 68.0, 64.5, 65.0, 70.0),
  Gender = c("F", "F", "F", "F", "M", "F", "F", "F", "M", "F", "M", "F", "M", "M", "M", "M", "F", "F", "M", "M", "M")
)
```

### 1b. Filter females

```
females <- subset(data, Gender == "F", select = c(Shoe_Size, Height))
print("Female data:")
```

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

```
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
## 24	8.5	69.0

### 1c. Filter males

```
males <- subset(data, Gender == "M", select = c(Shoe_Size, Height))
print("Male data:")
```

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

```
print(males)
```

```
## Shoe_Size Height
```

```
## 5      10.5  70.00
## 9      13.0  72.00
## 11     10.5  74.75
## 13     12.0  71.00
## 14     10.5  71.00
## 15     13.0  77.00
## 16     11.5  72.00
## 19     10.0  72.00
## 22       8.5  67.00
## 23     10.5  73.00
## 25     10.5  72.00
## 26     11.0  70.00
## 27       9.0  69.00
## 28     13.0  70.00
```

## 1d. Calculate means

```
mean_shoe_size <- mean(data$Shoe_Size)
mean_height <- mean(data$Height)
cat("Mean Shoe Size:", mean_shoe_size, "\n")
```

```
## Mean Shoe Size: 9.410714
```

```
cat("Mean Height:", mean_height, "\n")
```

```
## Mean Height: 68.58036
```

## 1e. Check relationship between shoe size and height

```
correlation <- cor(data$Shoe_Size, data$Height)
if (abs(correlation) < 0.1) {
  cat("No significant linear relationship between shoe size and height (Correlation:", correlation, ")\n")
} else {
  cat("There is a significant relationship (Correlation:", correlation, ")\n")
}
```

```
## There is a significant relationship (Correlation: 0.7751877 )
```

## 2. FACTOR

```
Months <- c("March", "April", "January", "November", "January", "September", "October", "September", "November")
factor_Months <- factor(Months)
print("Factor Months:")
```

```
## [1] "Factor Months:"
```

```
print(factor_Months)
```

```
## [1] March      April      January    November   January    September  October
## [8] September  November   August     January    November   November   February
## [15] May        August
## 9 Levels: April August February January March May November ... September
```

### 3. Summary

```
cat("Summary of Months:\n")
```

```
## Summary of Months:
```

```
print(summary(Months))
```

```
##      Length      Class      Mode
```

```
##           16 character character
```

```
cat("Summary of Factor Months:\n")
```

```
## Summary of Factor Months:
```

```
print(summary(factor_Months))
```

```
##      April      August February  January      March      May  November  October
```

```
##           1          2          1          3          1          1          4          1
```

```
## September
```

```
##           2
```

### 4. Vector and Frequency

```
Directions <- c("East", "West", "North")
```

```
Frequency <- c(1, 4, 3)
```

```
cat("Directions:\n")
```

```
## Directions:
```

```
print(Directions)
```

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

```
cat("Frequency:\n")
```

```
## Frequency:
```

```
print(Frequency)
```

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

### 4. Factor with specific order

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

```
print("Ordered Factor Data:")
```

```
## [1] "Ordered Factor Data:"
```

```
print(factor_data)
```

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

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

## 5. Read CSV data

```
data <- read.table("/cloud/project/Worksheet 4/import_march (1).csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)
print("Imported Data:")
```

```
## [1] "Imported Data:"
```

```
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
```

## 6. Exhaustive search function

```
exhaustive_search <- function(selected_number) {
  if (selected_number < 1 || selected_number > 50) {
    return("The number selected is beyond the range of 1 to 50")
  } else if (selected_number == 20) {
    return("TRUE")
  } else {
    return(as.character(selected_number))
  }
}
```

```
set.seed(Sys.time())
random_number <- sample(1:50, 1)
cat("The chosen number is:", random_number, "\n")
```

```
## The chosen number is: 35
```

```
result <- exhaustive_search(random_number)
cat("Result:", result, "\n")
```

```
## Result: 35
```

## 7. Minimum bills function

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

  if (price %% 50 != 0) {
    return("Price must be a multiple of 50.")
  }

  for (bill in bills) {
    while (price >= bill) {
      price <- price - bill
    }
  }
}
```

```

bill_count <- bill_count + 1
}
}

return(bill_count)
}

price_of_snack <- 2700
cat("Minimum number of bills needed:", min_bills(price_of_snack), "\n")

## Minimum number of bills needed: 4

```

## 8a. Create data frame for student grades

```

grades_data <- data.frame(
  Name = c("Annie", "Thea", "Steve", "Hanna"),
  Grade1 = c(85, 75, 75, 95),
  Grade2 = c(65, 75, 55, 75),
  Grade3 = c(85, 90, 80, 100),
  Grade4 = c(100, 90, 85, 90)
)

print("Student Grades Data:")

```

```
## [1] "Student Grades Data:"
```

```
print(grades_data)
```

```
##      Name Grade1 Grade2 Grade3 Grade4
## 1 Annie      85      65      85     100
## 2 Thea       75      75      90      90
## 3 Steve      75      55      80      85
## 4 Hanna      95      75     100      90
```

## 8b. Average grade calculation

```
cat("Students with average grade >= 88.75:\n")
```

```
## Students with average grade >= 88.75:
```

```
avg_scores <- rowMeans(grades_data[, 2:5]) # Calculate average scores for all students
```

```
# Loop through the students and print those with an average >= 88.75
```

```

for (i in seq_along(avg_scores)) {
  if (avg_scores[i] >= 88.75) {
    cat(grades_data$Name[i], "'s average grade this semester is", round(avg_scores[i], 2), "\n")
  }
}

```

```
## Hanna 's average grade this semester is 90
```

## 8c. Find highest scores

```
cat("Students with highest score > 90:\n")

## Students with highest score > 90:

results <- c()
for (i in 1:nrow(grades_data)) {
  highest_score <- max(grades_data[i, 2:5])
  if (highest_score > 90) {
    results <- c(results, paste(grades_data$Name[i], "'s highest grade this semester is", highest_score))
  }
}

cat(results, "\n")
```

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

#8d. Output of students' highest grade this semester

```
students_scores <- data.frame(
  student = c("Annie", "Bob", "Cathy", "David"),
  semester_scores = I(list(c(88, 95, 80), c(72, 85, 91), c(90, 88, 92), c(89, 70, 78)))
)

for (i in 1:nrow(students_scores)) {
  scores <- students_scores$semester_scores[[i]]
  highest_score <- sort(scores, decreasing = TRUE)[1]

  if (highest_score > 90) {
    cat(students_scores$student[i], "'s highest grade this semester is ", highest_score, ".\n", sep = "
  }
}
```

## Annie's highest grade this semester is 95.

## Bob's highest grade this semester is 91.

## Cathy's highest grade this semester is 92.