Worksheet#4

ANGELO A. GARRIDO

2025-011-3

```r
# 1. Shoe size, height, gender data
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,
          11.5,8.5,5.0,10.0,6.5,7.5,8.5,10.5,8.5,10.5,11.0,9.0,13.0)
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,
            68.0,70.0)
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")

shoesize_Data <- data.frame(size, height, gender)
names(shoesize_Data) <- c("Size", "Height", "Gender")
str(shoesize_Data)
```

```
## 'data.frame':    28 obs. of  3 variables:
##  $ 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
# 2. Subsets

sub_male <- shoesize_Data[shoesize_Data$Gender == "M", c("Size", "Height")]
sub_female <- shoesize_Data[shoesize_Data$Gender == "F", c("Size", "Height")]
print(sub_male)
```

```
##    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
## 25 10.5   72.0
## 26 11.0   70.0
## 27  9.0   68.0
## 28 13.0   70.0
```

```r
print(sub_female)
```

```
##    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
```

```
# c. Finding the mean of shoe size and height

size_mean <- mean(shoesize_Data$Size)
height_mean <- mean(shoesize_Data$Height)

size_mean
```

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

```
height_mean
```

```
## [1] 68.53571
```

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

# As shoe size increases, height also tends to increase.
# Larger shoe sizes are associated with taller people.
```

```
# 2. Constructing character vector months

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
```

```
# 3. Checking the summary()

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
```

```
# 4. Creating vector and factor for the table.

direction <- c("East", "West", "North")
frequency <- c(1, 4, 3)

data_direction <- data.frame(direction, frequency)
factored_data <- factor(data_direction$direction,
levels = c("East", "West", "North"))

print(factored_data)
```

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

```
# 5. Importing Data from CSV

# a. Import the excel file into the Environment Pane using read.table() function.

import_march <- read.table("import_march (1).csv", header = TRUE, sep = ",")
import_march
```

```
##   Students Students.1 Students.2 Students.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
```

```
# b. View the dataset.

print(import_march)
```

```
##   Students Students.1 Students.2 Students.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. Full Search

# a. Range 1 to 50

# Exhaustive Search Program

# Ask the user to enter a number

num <- 45 #"Enter a number from 1 to 50: "

if (num < 1 || num > 50) {
cat("The number selected is beyond the range of 1 to 50\n")
```

```
} else {
  for (i in 1:50) {
    if (i == num) {
      if (num == 20) {
          cat("TRUE\n")
  } else {
          cat("You selected:", num, "\n")
  }
  break
  }
  }
}
```

```
## You selected: 45
```

```
# 7. Change - Bill Breakdown

price <- 300
count1000 <- 0
count500 <- 0
count200 <- 0
count100 <- 0
count50 <- 0

if(price%%50 != 0){
cat("Not divisible by 50.\n")
    }else{
  if(price >= 1000){
count1000 <- price %/% 1000
price <- price %% 1000
}
  if (price >= 500) {
count500 <- price %/% 500
price <- price %% 500
}
  if (price >= 200) {
count200 <- price %/% 200
price <- price %% 200
}
  if (price >= 100) {
count100 <- price %/% 100
price <- price %% 100
}
  if (price >= 50) {
count50 <- price %/% 50
price <- price %% 50
}

cat("\n=== Bills used ===\n")
cat("1000 peso bill(s):", count1000, "\n")
cat("500  peso bill(s):", count500, "\n")
cat("200  peso bill(s):", count200, "\n")
cat("100  peso bill(s):", count100, "\n")
cat("50   peso bill(s):", count50, "\n")
```

```
bills <- count1000 + count500 + count200 + count100 + count50
cat("Total number of bills needed:", bills, "\n")
}
```

```
##
## === Bills used ===
## 1000 peso bill(s): 0
## 500  peso bill(s): 0
## 200  peso bill(s): 1
## 100  peso bill(s): 1
## 50   peso bill(s): 0
## Total number of bills needed: 2
```

```
# 8. Grades Sheet Data Frame

# a. Create a dataframe from the above table.

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(95,75,100,90)

grade_sheet <- data.frame(name,grade1,grade2,grade3,grade4)
grade_sheet
```

```
##    name grade1 grade2 grade3 grade4
## 1 Annie     85     65     85     95
## 2  Thea     65     75     90     75
## 3 Steve     75     55     80    100
## 4 Hanna     95     75    100     90
```

```
# b. Compute the average score manually

average_mean <- (grade_sheet$grade1 + grade_sheet$grade2
                 + grade_sheet$grade3 + grade_sheet$grade4) / 4
print(average_mean)
```

```
## [1] 82.50 76.25 77.50 90.00
```

```
for(i in 1:nrow(grade_sheet)){
  if(average_mean[i] > 90){
    cat(grade_sheet$name[i], "scores over 90\n")
  }
}
```

```
# c. Without using the mean function, output as follows for tests where

# the average score was less than 80 out of 4 tests.

# Example output: The nth test was difficult.

test1_avg <- sum(grade_sheet$grade1) / nrow(grade_sheet)
```

```r
test2_avg <- sum(grade_sheet$grade2) / nrow(grade_sheet)
test3_avg <- sum(grade_sheet$grade3) / nrow(grade_sheet)
test4_avg <- sum(grade_sheet$grade4) / nrow(grade_sheet)

test_averages <- c(test1_avg, test2_avg, test3_avg, test4_avg)

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

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

```r
# d. Without using the max function, output as follows for students whose

# highest score for a semester exceeds 90 points.

# Example Output: Annie's highest grade this semester is 95.

for (i in 1:nrow(grade_sheet)) {
highest <- grade_sheet[i, 2]
for (j in 3:5) {
if (grade_sheet[i, j] > highest) {
highest <- grade_sheet[i, j]
}
}

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

```
## Annie 's highest grade this semester is 95 .
## Steve 's highest grade this semester is 100 .
## Hanna 's highest grade this semester is 100 .
```