RWorksheet_parita#4a.Rmd

Daven M. Parita

2025-12-03

1.

```
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
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, 7
gender <- c("F", "F", "F", "F", "M", "F", "F", "F", "M", "F", "M", "F", "M", "M", "M", "M", "F", "F", "
shoes_df <- data.frame(
  "Shoe size" = shoe_size,
  "Height" = height,
  "Gender" = gender
)
View(shoes_df)
```

a.

The data is a class of dataframe with a total of 28 observations of 3 variables: Shoe size, Height, and Gender.

b.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
maleSet <- shoes_df %>%
  filter(Gender == "M") %>%
  select(Shoe.size, Height)
```

```r
maleSet2 <- subset(shoes_df, gender == "M")
femaleSet2 <- subset(shoes_df, gender == "F")

femaleSet <- shoes_df %>%
  filter(Gender == "F") %>%
  select(Shoe.size, Height)

maleSet
```

```
##    Shoe.size Height
## 1       10.5   70.0
## 2       13.0   72.0
## 3       10.5   74.5
## 4       12.0   71.0
## 5       10.5   71.0
## 6       13.0   77.0
## 7       11.5   72.0
## 8       10.0   72.0
## 9        8.5   67.0
## 10      10.5   73.0
## 11      10.5   72.0
## 12      11.0   70.0
## 13       9.0   69.0
## 14      13.0   70.0
```

```r
femaleSet
```

```
##    Shoe.size Height
## 1        6.5   66.0
## 2        9.0   68.0
## 3        8.5   64.5
## 4        8.5   65.0
## 5        7.0   64.0
## 6        9.5   70.0
## 7        9.0   71.0
## 8        7.5   64.0
## 9        8.5   67.0
## 10       8.5   59.0
## 11       5.0   62.0
## 12       6.5   66.0
## 13       7.5   64.0
## 14       8.5   69.0
```

```r
maleSet2
```

```
##    Shoe.size 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
```

```
femaleSet2
```

```
##    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        9.5   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
```

c.

```
shoe_size_mean <- mean(shoes_df$Shoe.size)
cat("the mean of Shoe size is:", shoe_size_mean)
```

```
## the mean of Shoe size is: 9.410714
```

```
height_mean <- mean(shoes_df$Height)
cat("\nthe mean of height is:", height_mean)
```

```
##
## the mean of height is: 68.57143
```

d.

The relationship between shoe size and height is quite linear. The taller the height value, the bigger the shoe size is. Although this is not the case for some of the observations.

Factors

2.

```
months_vector <- c("March","April","January","November","January", "September","October","September","N

factor_months_vector <- factor(months_vector)
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.

```
summary(months_vector)
```

```
##    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.

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

factor_directions <- factor(directions, levels = c("East", "West", "North"))
print(factor_directions)
```

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

5.

a.

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

b.

```
View(strategy_excel)
```

Using Conditional Statements (IF-ELSE)

6.

a.

```
random_number <- sample(1:50, 1)

cat("--- Random Selection ---\n")
```

```
## --- Random Selection ---
```

```
cat("The randomly chosen number (1 to 50) is:", random_number, "\n")
```

```
## The randomly chosen number (1 to 50) is: 19
```

```
cat("------------------------\n\n")
```

```
## ------------------------
```

```
user_input_str <- readline(prompt = "Enter a whole number: ")
```

```
## Enter a whole number:
```

```
user_number <- as.numeric(user_input_str)

cat("\n--- Result ---\n")
```

```
##
## --- Result ---
```

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

```
## Invalid input. Please enter a number.
```

```r
cat("--------------\n")
```

```
## --------------
```

7.

a.

```r
BILL_DENOMINATIONS <- c(1000, 500, 200, 100, 50)

calculate_min_bills <- function(price) {
  if (price <= 0 || price %% 50 != 0) {
    stop("Error: Price must be a positive number divisible by 50.")
  }

  remaining_price <- price
  total_bills <- 0
  bill_counts <- list()

  cat("--- Snack Purchase Analysis ---\n")
  cat(paste("Price of Snack:", price, "Pesos\n"))
  cat("-------------------------------\n")

  for (bill in BILL_DENOMINATIONS) {
    count <- floor(remaining_price / bill)

    if (count > 0) {
      bill_counts[[as.character(bill)]] <- count

      remaining_price <- remaining_price - (count * bill)

      total_bills <- total_bills + count
    }
  }

  cat("Breakdown of Bills Used:\n")
  for (bill_value in names(bill_counts)) {
    cat(paste("  ", bill_counts[[bill_value]], "x", bill_value, "Pesos\n"))
  }

  cat(paste("\nMinimum number of bills needed:", total_bills, "\n"))
  cat("-------------------------------\n")

  invisible(total_bills)
}

calculate_min_bills(1000)
```

```
## --- Snack Purchase Analysis ---
## Price of Snack: 1000 Pesos
## -------------------------------
```

```
## Breakdown of Bills Used:
##    1 x 1000 Pesos
##
## Minimum number of bills needed: 1
## ------------------------------
```

8.

a.

```
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)

math_scores <- data.frame(Name, Grade1, Grade2, Grade3, Grade4)
```

b.

```
math_scores$Average <- apply(math_scores[, 2:5], 1, mean)

print_student_averages <- function(df) {
  for (i in 1:nrow(df)) {
    cat(paste0(df$Name[i], "'s average grade this semester is ", df$Average[i], ".\n"))
  }
}

print_student_averages(math_scores)
```

```
## 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.
```

c.

```
grade_averages <- apply(math_scores[, 2:5], 2, mean)

difficult_tests <- names(grade_averages[grade_averages < 80])

for (test_name in difficult_tests) {
  test_number <- sub("Grade", "", test_name)
  cat(paste0("The ", test_number, "th test was difficult (Avg: ", grade_averages[test_name], ").\n"))
}
```

```
## The 2th test was difficult (Avg: 67.5).
```

d.

```
highest_scores <- apply(math_scores[, 2:5], 1, function(x) {
  return(sort(x, decreasing = FALSE)[length(x)])
})

math_scores$Highest <- highest_scores

high_score_students <- subset(math_scores, Highest > 90)

for (i in 1:nrow(high_score_students)) {
    cat(paste0(high_score_students$Name[i], "'s highest grade this semester is ", high_score_students$Hi
}
```

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