

Rworksheet_simpron#4a

Michael T. Simpron

2025-11-16

#1.

#A.

```
table <- data.frame (  
  Shoe_Size = shoe_size,  
  Shoe_Height = shoe_height  
)
```

```
print(table)

##      Shoe_Size Shoe_Height
## 1          6.5       66.0
## 2          9.0       68.0
## 3          8.5       64.5
## 4          8.5       65.0
## 5         10.5      70.0
## 6          7.5       64.0
## 7          9.5      70.0
## 8          9.0      71.0
## 9         13.0      72.0
## 10         7.5       64.0
## 11         10.5      74.5
## 12         8.5       67.0
## 13         12.0      71.0
## 14         10.5      71.0
## 15         13.0      77.0
## 16         11.5      72.0
## 17         8.5       59.0
## 18         5.0       62.0
## 19         10.0      72.0
## 20         6.5       66.0
## 21         7.5       64.0
## 22         8.5       67.0
## 23         10.5      73.0
## 24         8.5       69.0
## 25         10.5      72.0
## 26         11.0      70.0
## 27         9.0       69.0
## 28         13.0      70.0
```

```

write.csv(table, "shoe_data.csv", row.names = FALSE)

#It create a data base containing the shoe size and height.

#B.
table$gender <- c("Male", "Male", "Female", "Male", "Female", "Male", "Female", "Male", "Female",
                  "Male", "Female", "Male", "Female", "Male", "Female", "Male", "Female", "Male", "Female",
                  "Male", "Female", "Male", "Female", "Male", "Female", "Male", "Female", "Male", "Female")

print(table)

##      Shoe_Size Shoe_Height gender
## 1        6.5       66.0   Male
## 2        9.0       68.0   Male
## 3        8.5       64.5 Female
## 4        8.5       65.0   Male
## 5       10.5       70.0 Female
## 6        7.5       64.0   Male
## 7        9.5       70.0 Female
## 8        9.0       71.0   Male
## 9       13.0       72.0 Female
## 10       7.5       64.0   Male
## 11      10.5       74.5 Female
## 12       8.5       67.0   Male
## 13      12.0       71.0 Female
## 14      10.5       71.0   Male
## 15      13.0       77.0 Female
## 16      11.5       72.0   Male
## 17       8.5       59.0 Female
## 18       5.0       62.0   Male
## 19      10.0       72.0 Female
## 20       6.5       66.0   Male
## 21       7.5       64.0 Female
## 22       8.5       67.0   Male
## 23      10.5       73.0 Female
## 24       8.5       69.0   Male
## 25      10.5       72.0 Female
## 26      11.0       70.0   Male
## 27       9.0       69.0 Female
## 28      13.0       70.0   Male

males <- subset(table, gender == "Male")
females <- subset(table, gender == "Female")

print(males)

##      Shoe_Size Shoe_Height gender
## 1        6.5       66   Male
## 2        9.0       68   Male
## 4        8.5       65   Male
## 6        7.5       64   Male
## 8        9.0       71   Male

```

```

## 10      7.5      64  Male
## 12      8.5      67  Male
## 14     10.5      71  Male
## 16     11.5      72  Male
## 18      5.0      62  Male
## 20      6.5      66  Male
## 22      8.5      67  Male
## 24      8.5      69  Male
## 26     11.0      70  Male
## 28     13.0      70  Male

```

```
print(females)
```

```

##   Shoe_Size Shoe_Height gender
## 3      8.5       64.5 Female
## 5     10.5       70.0 Female
## 7      9.5       70.0 Female
## 9     13.0       72.0 Female
## 11     10.5       74.5 Female
## 13     12.0       71.0 Female
## 15     13.0       77.0 Female
## 17      8.5       59.0 Female
## 19     10.0       72.0 Female
## 21      7.5       64.0 Female
## 23     10.5       73.0 Female
## 25     10.5       72.0 Female
## 27      9.0       69.0 Female

```

#C.

```
mean(table$Shoe_Height)
```

```
## [1] 68.57143
```

#D.

#Yes there are, because taller individuals tend to have bigger footsize.

###2.

```

months <- c("March", "April", "January", "November", "January", "September", "October", "September", "November"
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.

```

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.

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

table <- data.frame (
  Direction = direction,
  Frequency = frequency
)
print(table)

##   Direction Frequency
## 1      Easy         1
## 2      West         4
## 3     North         3

factor_direction <- factor(direction,levels = c("Easy", "West", "North"))
print(factor_direction)

## [1] Easy  West  North
## Levels: Easy West North

####5.

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

```

```

search_number <- function(target, vec) {
  for (v in vec) {
    if (v == target) {
      return(TRUE)
    }
  }
  return(FALSE)
}

# Ask the user to input a number (or you could simulate with sample)
input_str <- readline(prompt = "Please enter a number (1-50): ")

```

Please enter a number (1-50):

```

x <- as.numeric(input_str)

if (is.na(x)) {
  cat("You did not enter a valid number.\n")
} else if (x < 1 || x > 50) {
  cat("The number selected is beyond the range of 1 to 50\n")
} else {

  vec <- 1:50
  found <- search_number(x, vec)

  if (found) {
    if (x == 20) {
      cat("TRUE\n")
    } else {
      cat(x, "\n")
    }
  } else {
    cat("Number not found in the search space.\n")
  }
}

```

You did not enter a valid number.

####7.

```

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

  if (price %% 50 != 0) {
    cat("Error: Price must be divisible by 50.\n")
    return(NULL)
  }

  count <- 0
  remaining <- price

  #large first

```

```

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

cat("Minimum number of bills needed:", count, "\n")
return(count)
}

min_bills(300)

```

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

```
## [1] 2
```

```
min_bills(1000)
```

```
## Minimum number of bills needed: 1
```

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

```
min_bills(650)
```

```
## Minimum number of bills needed: 3
```

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

```
## #8.
```

```
#A.
grades_df <- 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_df
```

	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

```
#B.
averages <- (grades_df$Grade1 + grades_df$Grade2 + grades_df$Grade3 + grades_df$Grade4) / 4

high_achievers <- grades_df[averages >= 90, ]
high_averages <- averages[averages >= 90]

for (i in 1:nrow(high_achievers)) {
  cat(high_achievers>Name[i], "'s average grade this semester is ", high_averages[i], ".\n", sep = "")}
```

Hanna's average grade this semester is 90.

```
#C.
avg_grade1 <- sum(grades_df$Grade1) / nrow(grades_df)
avg_grade2 <- sum(grades_df$Grade2) / nrow(grades_df)
avg_grade3 <- sum(grades_df$Grade3) / nrow(grades_df)
avg_grade4 <- sum(grades_df$Grade4) / nrow(grades_df)

test_avgs <- c(avg_grade1, avg_grade2, avg_grade3, avg_grade4)
test_names <- c("Grade1", "Grade2", "Grade3", "Grade4")

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

The Grade2 was difficult.

```
#D.
highest_grades <- apply(grades_df[, 2:5], 1, function(row) {
  sorted <- sort(as.numeric(row))
  sorted[length(sorted])})
```

for (i in 1:nrow(grades_df)) {
 name <- grades_df>Name[i]
 high_score <- highest_grades[i]
 if (high_score > 90) {
 cat(name, "'s highest grade this semester is ", high_score, ".\n", sep = "")}}

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