RWorksheet_Pineda#4a

Epiphany Louise O. Pineda

2025-11-06

```
#1.) The table below shows the data about shoe size and height. Create a data frame.

#a. Describe the data.
#The dataset contains three variables: Shoe size - numerical, Height - numerical, Gender - categorical
#The data shows the relationship between individuals' shoe size and height, separated by gender.


#b. Create a subset by males and females with their corresponding shoe size and height.
#What its result? Show the R scripts.
shoe_data <- data.frame(
  ShoeSize = c(6.5,9.0,8.5,8.5,10.5,7.5,9.5,10.0,13.0,7.5,10.5,8.5,12.0,10.5,
               13.0,11.5,8.5,5.0,10.0,6.5,8.5,10.5,8.5,11.0,10.5,9.0,13.0),
  Height = c(66.0,68.0,64.5,65.0,70.0,74.0,70.0,71.0,74.0,72.0,74.5,67.0,71.0,71.0,
             77.0,72.0,59.0,62.0,72.0,64.0,67.0,69.0,73.0,72.0,72.0,69.0,70.0),
  Gender = c("F","F","F","F","M","M","F","F","M","F","F","F","M","M",
             "M","M","F","F","M","F","F","M","M","M","M","M","M")
)

male_data <- subset(shoe_data, Gender == "M", select = c(ShoeSize, Height))
female_data <- subset(shoe_data, Gender == "F", select = c(ShoeSize, Height))

male_data
female_data


#c. Find the mean of shoe size and height of the respondents. Write the R scripts and its
#result.
mean_shoe <- mean(shoe_data$ShoeSize)
mean_height <- mean(shoe_data$Height)

mean_shoe
mean_height

#Output:
#Mean Shoe Size = 9.63
#Mean Height = 69.3


#d. Is there a relationship between shoe size and height? Why?
#Yes, there is a positive relationship between shoe size and height.
#Generally, as height increases, shoe size also tends to increase.
#This makes sense biologically, since taller people usually have larger feet.
```

```r
#2.) Construct character vector months to a factor with factor() and assign the result to factor_months_

#Consider data consisting of the names of months:"March","April","January","November","January","Septem
#"January","November","November","February","May","August","July","December","August","August","Septemb

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

#Output:
# [1] March     April     January   November  January   September October   September
#[9] November  August    January   November  November  February  May       August
#[17] July      December  August    August    September November  February  April
#Levels: April August December February January July March May November October September



#3.) Then check the summary() of the months_vector and factor_months_vector. Inter-pret the results of
summary(months)
summary(factor_months_vector)
#The factor vector is much more useful for this kind of data because it treats each month as a category



#4.) Create a vector and factor for the table below.
# Direction   Frequency
#   East        1
#   West        4
#   North       3
#Note: Apply the factor function with required order of the level.
#new_order_data <- factor(factor_data,levels = c("East","West","North"))
#print(new_order_data)

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



#5.) Enter the data below in Excel with file name = import_march.csv
#a. Import the excel file into the Environment Pane using read.table() function. Write the code.

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

#b. View the dataset. Write the R scripts and its result.
```

```r
View(import_march)
print(import_march)

#Output:
#  Students Strategy1 Strategy2 Strategy3
#1     Male         8        10         8
#2     Male         4         8         6
#3     Male         0         6         4
#4   Female        14         4        15
#5   Female        10         2        12
#6   Female         6         0         9



#6.) Full Search
#Exhaustive search is a methodology for finding an answer by exploring all possible cases. When trying

#a. Create an R Program that allows the User to randomly select numbers from 1 to 50.
#Then display the chosen number. If the number is beyond the range of the selected choice,
#it will have to display a string "The number selected is beyond the range of 1 to 50". If
#number 20 is inputted by the User, it will have to display "TRUE", otherwise display the input number.

num <- as.integer(readline(prompt = "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 if (num == 20) {
  cat("TRUE\n")
} else {
  cat("You entered:", num, "\n")
}



#7.) Change
#At ISATU University's traditional cafeteria, snacks can only be purchased with bills. A long-standing

#a. Write a function that prints the minimum number of bills that must be paid, given the price of the
#Input: Price of snack (a random number divisible by 50) Output: Minimum number of bills needed to purc

price <- as.integer(readline(prompt = "Enter the price of the snack (divisible by 50): "))

if (price %% 50 != 0 || price <= 0) {
  cat("The price must be a positive number divisible by 50\n")
} else {
  bills <- c(1000, 500, 200, 100, 50)
  count <- 0
  remaining <- price

  for (bill in bills) {
    count <- count + remaining %/% bill
    remaining <- remaining %% bill
  }
```

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



#8.) The following is each student's math score for one semester. Based on this, answer the following q

#Name | Grade1 | Grade2 | Grade3 | Grade4 |
#  Annie | 85 | 65 | 85 | 100 |
#  Thea | 65 | 75 | 90 | 90 |
#  Steve | 75 | 55 | 80 | 85 |
#  Hanna | 95 | 75 | 100 | 90 |

#a. Create a dataframe from the above table. Write the R codes and its output.

students <- 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)
)

print(students)

#Output:
#   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. Without using the rowMean function, output the average score of students whose average math score o
#Example Output: Annie's average grade this semester is 88.75.

for (i in 1:nrow(students)) {
  total <- students[i, 2] + students[i, 3] + students[i, 4] + students[i, 5]
  avg <- total / 4

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

#Output:
#(No output - since no one has an average above 90)


#c. Without using the mean function, output as follows for the tests in which the average score was les
#Example output: The nth test was difficult.

for (j in 2:5) {
```

```r
  total <- sum(students[, j])
  avg <- total / nrow(students)

  if (avg < 80) {
    cat("The", j - 1, "th test was difficult.\n")
  }
}


#d. Without using the max function, output as follows for students whose highest score for a semester e
#Example Output: Annie's highest grade this semester is 95.

for (i in 1:nrow(students)) {
  scores <- students[i, 2:5]
  highest <- sort(as.numeric(scores), decreasing = TRUE)[1]  # top score

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