RWorksheet_rocillo#4a

Cassandra Jielin Rocillo

2024-10-17

1. The table below shows the data about shoe size and height. Create a 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,
  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,
  gender <- c("F", "F", "F", "F", "M", "F", "F", "F", "M", "F", "M", "F", "M", "M", "M", "M", "F", "F",

data <- data.frame(Shoe_size = shoe_size, Height = height, Gender = gender)
```

   a. Describe the data.

- It shows the shoe size, height and gender.

   b. Create a subset by males and females with their corresponding shoe size and height. What its result? Show the R scripts.

```
male_subset <- subset(data, gender == "M")
female_subset <- subset(data, gender == "F")

print(male_subset)
```

```
##    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
```

```
print(female_subset)
```

```
##    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
```

```
print(data)
```

```
##    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
## 5       10.5   70.0      M
## 6        7.0   64.0      F
## 7        9.5   70.0      F
## 8        9.0   71.0      F
## 9       13.0   72.0      M
## 10       7.5   64.0      F
## 11      10.5   74.5      M
## 12       8.5   67.0      F
## 13      12.0   71.0      M
## 14      10.5   71.0      M
## 15      13.0   77.0      M
## 16      11.5   72.0      M
## 17       8.5   59.0      F
## 18       5.0   62.0      F
## 19      10.0   72.0      M
## 20       6.5   66.0      F
## 21       7.5   64.0      F
## 22       8.5   67.0      M
## 23      10.5   73.0      M
## 24       8.5   69.0      F
## 25      10.5   72.0      M
## 26      11.0   70.0      M
## 27       9.0   69.0      M
## 28      13.0   70.0      M
```

c. Find the mean of shoe size and height of the respondents. Write the R scripts and its result.

```
mean_of_shoe_size <- mean(shoe_size)
mean_of_height <- mean(height)

shoe_size_mean <- paste("Mean of shoe size",mean_of_shoe_size)
height_mean <- paste("Mean of height", mean_of_height)

print(shoe_size_mean)
```

```
## [1] "Mean of shoe size 9.41071428571429"
```

```
print(height_mean)
```

```
## [1] "Mean of height 68.5714285714286"
```

d. Is there a relationship between shoe size and height? Why? Yes, there is a relationship between shoe

size and height, as taller individuals often have larger feet.

2. Construct character vector months to a factor with factor() and assign the result to factor_months_vector. Print out factor_months_vector and assert that R prints out the factor levels below the actual values. Consider data consisting of the names of months: ("March", "April", "January", "November", "January", "September", "October", "September", "November", "August", "January", "November", "November", "February", "May", "August", "July", "December", "August", "August", "September", "November", "February", "April")

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

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. Then check the summary() of the months_vector and factor_months_vector. Interpret the results of both vectors.

```
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
```

Are they both equally useful in this case? - the factor_months_vector is more useful because it shows specifically what's on your data and it categorized the months and provides the count of how many times each month repeats.

4. Create a vector and factor for the table below.

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

data <- data.frame(direction, frequency)
data
```

```
##   direction frequency
## 1      East         1
## 2      West         4
## 3     North         3
```

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

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

5. 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.

```
 library(readxl)
data <- read.table("/cloud/project/worksheet/worksheet 4/import_march.csv", header = TRUE, sep = ",")
```

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

```
print(data)
```

```
##   Student 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. Full Search Exhaustive search is a methodology for finding an answer by exploring all possible cases. When trying to find a desired number in a set of given numbers, the method of finding the corresponding number by checking all elements in the set one by one can be called an exhaustive search. Implement an exhaustive search function that meets the input/output conditions below.

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

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

```
num
```

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

```
if(num < 1 || num > 50) {
  paste("The number selected is beyond the range of 1 to 50")
}else if(num == 20) {
  print("TRUE")
} else{
  print(num)
}
```

```
## [1] "The number selected is beyond the range of 1 to 50"
```

7. Change At ISATU University's traditional cafeteria, snacks can only be purchased with bills. A long-standing rule at the concession stand is that snacks must be purchased with as few coins as possible. There are three types of bills: 50 pesos, 100 pesos, 200 pesos, 500 pesos, 1000 pesos.

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

```
price <- readline(prompt = "Price of snack: ")
```

```
## Price of snack:
```

```
paste("price of snack is ", price)
```

```
## [1] "price of snack is  "
```

```
if (price < 50) {
  print("Minimum number of bills needed to purchase a snack is 50 pesos")
} else if (price >= 50 && price < 100) {
  print("Minimum number of bills needed to purchase a snack is 100 pesos")
} else if (price >= 100 && price < 200) {
  print("Minimum number of bills needed to purchase a snack is 200 pesos")
} else if (price >= 200 && price < 500) {
  print("Minimum number of bills needed to purchase a snack is 500 pesos")
} else {
  print("Minimum number of bills needed to purchase a snack is 1000 pesos")
}
```

```
## [1] "Minimum number of bills needed to purchase a snack is 50 pesos"
```

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

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

```
Name <- c("Annie", "Thea", "Steve", "Hanna")
Grade_1 <- c(85, 75, 65, 95)
Grade_2 <- c(65, 75, 55, 75)
Grade_3 <- c(85, 90, 80, 100)
Grade_4 <- c(100, 90, 85, 90)

data <- data.frame(Name, Grade_1, Grade_2, Grade_3, Grade_4)
data
```

```
##    Name Grade_1 Grade_2 Grade_3 Grade_4
## 1 Annie      85      65      85     100
## 2  Thea      75      75      90      90
## 3 Steve      65      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 over 90 points during the semester. write R code and its output. Example Output: Annie's average grade this semester is 88.75.

```
  row_1 <- mean(as.numeric(data[1, 2:5]))
  paste(data[1, 1], "'s average grade this semester is", row_1)
```

```
## [1] "Annie 's average grade this semester is 83.75"
```

```
  row_2 <- mean(as.numeric(data[2, 2:5]))
  paste(data[2, 1], "'s average grade this semester is", row_2)
```

```
## [1] "Thea 's average grade this semester is 82.5"
```

```
  row_3 <- mean(as.numeric(data[3, 2:5]))
  paste(data[3, 1], "'s average grade this semester is", row_3)
```

```
## [1] "Steve 's average grade this semester is 71.25"
```

```
  row_4 <- mean(as.numeric(data[4, 2:5]))
  paste(data[4, 1], "'s average grade this semester is", row_4)
```

```
## [1] "Hanna 's average grade this semester is 90"
```

c. Without using the mean function, output as follows for the tests in which the average score was less than 80 out of 4 tests. Example output: The nth test was difficult.

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.