

RWorksheet_Alpange#4a

Jersey Gabriel E. Alpange

2024-10-16

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

```
shoe_size <- c(8, 9, 7, 10, 6)
height <- c(170, 175, 160, 180, 155)
gender <- c("M", "M", "F", "M", "F")

df <- data.frame(shoe_size, height, gender)
df
```

```
##   shoe_size height gender
## 1         8    170      M
## 2         9    175      M
## 3         7    160      F
## 4        10    180      M
## 5         6    155      F
```

- a. Describe the data.

The data consists of shoe size, height, and gender of respondents. Each row corresponds to a respondent, and the columns represent their shoe size, height in cm, and gender (M for male, F for female).

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

```
shoe_size <- c(8, 9, 7, 10, 6)
height <- c(170, 175, 160, 180, 155)
gender <- c("M", "M", "F", "M", "F")

df <- data.frame(shoe_size, height, gender)

male <- subset(df, gender == "M")
female <- subset(df, gender == "F")

print("Male Subset:")
```

```
## [1] "Male Subset:"
print(male)
```

```
##   shoe_size height gender
## 1         8    170      M
## 2         9    175      M
## 4        10    180      M
```

```
print("Female Subset:")
```

```
## [1] "Female Subset:"
```

```
print(female)
```

```
##   shoe_size height gender
## 3         7    160      F
## 5         6    155      F
```

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

```
shoe_size <- c(8, 9, 7, 10, 6)
height <- c(170, 175, 160, 180, 155)
gender <- c("M", "M", "F", "M", "F")

df <- data.frame(shoe_size, height, gender)

mean_shoe_size <- mean(df$shoe_size)
mean_height <- mean(df$height)

print(paste("Mean Shoe Size: ", mean_shoe_size))
```

```
## [1] "Mean Shoe Size: 8"
```

```
print(paste("Mean Height: ", mean_height))
```

```
## [1] "Mean Height: 168"
```

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

```
shoe_size <- c(8, 9, 7, 10, 6)
height <- c(170, 175, 160, 180, 155)
gender <- c("M", "M", "F", "M", "F")
df <- data.frame(shoe_size, height, gender)

correlation <- cor(df$shoe_size, df$height)
print(paste("Correlation between shoe size and height: ", correlation))
```

```
## [1] "Correlation between shoe size and height: 0.99124070716193"
```

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”, “April”)

```
months_vector <- 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_vector)
print(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
```

```
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
```

3. Then check the summary() of the months_vector and factor_months_vector. | Interpret the results of both vectors. Are they both equally useful in this case?

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

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

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

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

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

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

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

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

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.

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

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

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

Using Conditional Statements (IF-ELSE) 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.

- 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.

```
exhaustive_search <- function() {
  number <- sample(1:50, 1)
  print(paste("Selected number:", number))

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

exhaustive_search()
```

```
## [1] "Selected number: 7"
## [1] "Input number: 7"
```

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.

- 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.

```
min_bills <- function(price) {
  bills <- c(1000, 500, 200, 100, 50)
  count <- 0

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

  return(count)
}
```

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")
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 <- data.frame(name, grade1, grade2, grade3, grade4)
print(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. 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.

```
for (i in 1:nrow(grades_df)) {
  avg_score <- sum(grades_df[i, 2:5]) / 4
  if (avg_score > 90) {
    print(paste(grades_df$name[i], "'s average grade this semester is", avg_score))
  }
}
```

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.

```
for (i in 2:5) {
  avg_test <- mean(grades_df[, i])
  if (avg_test < 80) {
    print(paste("The test", i - 1, "was difficult."))
  }
}
```

```
## [1] "The test 2 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.

```
for (i in 1:nrow(grades_df)) {
  max_score <- max(grades_df[i, 2:5])
  if (max_score > 90) {
    print(paste(grades_df$name[i], "'s highest grade this semester is", max_score))
  }
}
```

```
## [1] "Annie 's highest grade this semester is 100"
## [1] "Hanna 's highest grade this semester is 100"
```