Rworksheet_cadiz#4a

John Dave R. Cadiz

2024-10-21

```r
## 1. The table below shows the data about shoe size and height. Create a data frame.
shoesize <- c(6.5, 9.0, 8.5, 8.5, 7.0, 9.0, 9.5, 13.0, 7.5, 10.5, 10.5, 12.0,
              10.5, 13.0, 11.5, 8.5, 5.0, 10.0, 6.5, 7.5, 8.5, 10.5, 8.5, 10.5,
              11.0, 9.0, 13.0)

height <- c(66.0, 68.0, 65.0, 65.0, 64.0, 71.0, 72.0, 72.0, 74.5, 67.0, 74.5,
            71.0, 71.0, 77.0, 72.0, 59.0, 62.0, 72.0, 66.0, 64.0, 67.0, 73.0,
            69.0, 72.0, 70.0, 69.0, 70.0)

gender <- c("F", "F", "F", "F", "F", "F", "F", "M", "F", "M", "M", "M", "M",
            "M", "M", "F", "F", "M", "F", "F", "F", "M", "F", "M", "M", "M", "M")

shoedata <- data.frame(ShoeSize = shoesize, Height = height, Gender = gender)

print(shoedata)
```

```
##    ShoeSize Height Gender
## 1       6.5   66.0      F
## 2       9.0   68.0      F
## 3       8.5   65.0      F
## 4       8.5   65.0      F
## 5       7.0   64.0      F
## 6       9.0   71.0      F
## 7       9.5   72.0      F
## 8      13.0   72.0      M
## 9       7.5   74.5      F
## 10     10.5   67.0      M
## 11     10.5   74.5      M
## 12     12.0   71.0      M
## 13     10.5   71.0      M
## 14     13.0   77.0      M
## 15     11.5   72.0      M
## 16      8.5   59.0      F
## 17      5.0   62.0      F
## 18     10.0   72.0      M
## 19      6.5   66.0      F
## 20      7.5   64.0      F
## 21      8.5   67.0      F
## 22     10.5   73.0      M
## 23      8.5   69.0      F
## 24     10.5   72.0      M
## 25     11.0   70.0      M
## 26      9.0   69.0      M
```

```
## 27     13.0   70.0      M
```

a. Describe the data.

```
#Data frame shoedata are the data frome the table above it shows the record of
#the size, height, and gender of the user.
```

b. Create a subset by males and females with their corresponding shoe size

##and height. What its result? Show the R Scripts.

```
maledata <- subset(shoedata, Gender == "M", select = c(ShoeSize, Height))

print(maledata)
```

```
##    ShoeSize Height
## 8      13.0   72.0
## 10     10.5   67.0
## 11     10.5   74.5
## 12     12.0   71.0
## 13     10.5   71.0
## 14     13.0   77.0
## 15     11.5   72.0
## 18     10.0   72.0
## 22     10.5   73.0
## 24     10.5   72.0
## 25     11.0   70.0
## 26      9.0   69.0
## 27     13.0   70.0
```

```
femaledata <- subset(shoedata, Gender == "F", select = c(ShoeSize, Height))

print(femaledata)
```

```
##    ShoeSize Height
## 1       6.5   66.0
## 2       9.0   68.0
## 3       8.5   65.0
## 4       8.5   65.0
## 5       7.0   64.0
## 6       9.0   71.0
## 7       9.5   72.0
## 9       7.5   74.5
## 16      8.5   59.0
## 17      5.0   62.0
## 19      6.5   66.0
## 20      7.5   64.0
## 21      8.5   67.0
## 23      8.5   69.0
```

c. Find the mean of the shoe size and the height of the respondents.

##Write the R scripts and its results.

```
mean_shoesize <- mean(shoedata$ShoeSize)
print(mean_shoesize)
```

```
## [1] 9.444444
```

```
mean_height <- mean(shoedata$Height)
print(mean_height)
```

```
## [1] 69
```

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

```
# Yes, because as shown as the table above as the height increase the shoe size
#also increase, this could be a factor of increasing shoe size
#because of height of the respondent.
```

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 scripts print 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", "Janiary", "November", "November", "February", "May", "August", ##"July", "December", "August", "August", "August", "September", "November", ##"February", "April")

```
months_vector <- c("March", "April", "January", "November", "January",
                   "September", "October", "September", "November", "August",
                   "January", "November", "November", "February", "May",
                   "August", "July", "December", "August", "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    August
## [22] September 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 if both vectors. ##Are they both equally useful in this case?

```
summary_monthsvector <- summary(months_vector)
print(summary_monthsvector)
```

```
##    Length     Class      Mode
##        25 character character
```

```
summary_factormonthsvector <- summary(factor_months_vector)
print(summary_factormonthsvector)
```

```
##     April    August  December  February   January      July     March       May
##         2         5         1         2         3         1         1         1
```

```
##  November   October September
##         5         1         3
```

```
## It shows the result of the object, summary tells the length, class, and mode
##of the number of characters. In line with this its states the number of months
##mentioned, in this way it is useful to use this r code so that it sorts
##automatically and gives information regarding to the number of months tallied
##from the data above.
```

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

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

factor_data<- factor(Direction)

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

print(new_order_data)
```

```
## [1] East  West  North West  North West  North West
## 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.

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

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

```
print(dataexcel)
```

```
##   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. Full Search

Exhaustive search is a methodology for finding an answer by exploring

##all possible cases.

When tring to find a desired number in a set of given numbers, the method of

##finding the corresponding number by checkug all elements in the set one by one ##can 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 displat a string "The number selected is ##beyond the ranfe of 1 to 50". If the number is inputted by the user, ##it will have to display "TRUE", otherwise display the input number.

```
exhaustives <- function() {

  selected_number <- as.numeric(readline(prompt = "Select a number from 1 to 50: "))

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

7. Change

##At ISATU Univesity's traditional cafeteria, snacks can only be purchased with ##bills. A long-standing rule at the concession standt 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, and 1000 pesos.

a. Write a function that prints the minimum number of bills that must be

##paid, given the price of the snanck.

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)

  num_bills <- 0
  for(bill in bills) {
    count <- price %/% bill
    num_bills <- num_bills + count

    price <- price %% bill
  }

  print(paste("Minimum number of bills needed to purchase a snack: ", num_bills))
}

min_bills(1250)
```

```
## [1] "Minimum number of bills needed to purchase a snack:  3"
```

8. The following is each student's math score for one semester.

##Based on this, answer the following questions.

a. Create a data frame 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)

gradesdf <- data.frame(
  Name = name, Grade_1 = grade1, Grade_2= grade2, Grade_3 = grade3,
  Grade_4 = grade4)

print(gradesdf)
```

```
##    Name Grade_1 Grade_2 Grade_3 Grade_4
## 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 studens

##whose average math score over 90 points during the semester. ##Write the R code and its output.

Example Output: Annie's Average grade this semester is 88.75.

```
gradesdf$Average <- (gradesdf$Grade_1 + gradesdf$Grade_2 + gradesdf$Grade_3 +
                       gradesdf$Grade_4) / 4


for (i in 1:nrow(gradesdf)) {
  if (gradesdf$Average[i] > 90) {
    cat(gradesdf$Name[i], "'s Average grade this semester is",
        round(gradesdf$Average[i], 2), ".\n")
  }
}
```

c. Without using the mean functuon, 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 (j in 2:5) {
  total_test_score <- sum(gradesdf[, j])
  avg_test_score <- total_test_score / nrow(gradesdf)

  if (avg_test_score < 80) {
    test_num <- j - 1
    print(paste("The", test_num,
            ifelse(test_num == 1, "st",
            ifelse(test_num == 2, "nd",
```

```
            ifelse(test_num == 3, "rd", "th"))), "test was difficult."))
  }
}
```

```
## [1] "The 2 nd test was difficult."
```

d. Without using the max function, output as follows for students whose

##highest score gor a semester exceeds 90 points.

Example output: Annie's highest grade this semester is 95.

```
for (i in 1:nrow(gradesdf)) {

  grades <- c(gradesdf$Grade_1[i], gradesdf$Grade_2[i], gradesdf$Grade_3[i],
              gradesdf$Grade_4[i])

  highest_grade <- grades[1]

  for (grade in grades) {
    if (grade > highest_grade) {
      highest_grade <- grade
    }
  }

  if (highest_grade > 90) {
    print(paste(gradesdf$Name[i], "'s highest grade this semester is",
                highest_grade))
  }
}
```

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