RWorksheet_Magallanes#4a

Killy Magallanes

2024-10-14

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

a. It displays the shoe size, height, and gender in column

b. Create a subset by males and females with their corresponding shoe size and height. What is the result? show the Rscript

```
female_data <- subset(Household_Data, Gender == "F")

male_data <- subset(Household_Data, Gender == "M")

female_data
```

```
## # A tibble: 14 x 3
##    Shoe_size Height Gender
##        <dbl>  <dbl> <chr>
##  1       6.5   66   F
##  2       9     68   F
##  3       8.5   64.5 F
##  4       8.5   65   F
##  5       7     64   F
##  6       9.5   70   F
##  7       9     71   F
##  8       7.5   64   F
##  9       8.5   67   F
## 10       8.5   59   F
## 11       5     62   F
## 12       6.5   66   F
## 13       7.5   64   F
## 14       8.5   69   F
```

```
male_data
```

```
## # A tibble: 14 x 3
##    Shoe_size Height Gender
##        <dbl>  <dbl> <chr>
##  1      10.5   70   M
##  2      13     72   M
##  3      10.5   74.5 M
##  4      12     71   M
##  5      10.5   71   M
```

```
##  6      13     77   M
##  7      11.5   72   M
##  8      10     72   M
##  9       8.5   67   M
## 10      10.5   73   M
## 11      10.5   72   M
## 12      11     70   M
## 13       9     69   M
## 14      13     70   M
```

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

```
mean_shoesize <- mean(Household_Data$Shoe_size)
mean_shoesize
```

```
## [1] 9.410714
```

```
mean_height <- mean(Household_Data$Height)
mean_height
```

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

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

- Yes, the taller the respondent, the longer his/her feet

FACTORS

2. construct character vectors 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 thee 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", "Septembe

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
```

```
print(levels(factor_months_vector))
```

```
##  [1] "April"     "August"    "December"  "February"  "January"   "July"
##  [7] "March"     "May"       "November"  "October"   "September"
```

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?

```
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
```

Summary function will return basic information about the vector, such as the length (i.e., the total number of values) and other high-level characteristics. It returns a count of how many times each factor level appears in the vector.

4. Create a vector and factor for the table

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

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

```
frequency_vector <- c(1,4,3)
frequency_vector
```

```
## [1] 1 4 3
```

```
direction_factor <- factor(direction_vector)
direction_factor
```

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

```
frequency_factor <- factor(frequency_vector)
frequency_factor
```

```
## [1] 1 4 3
## Levels: 1 3 4
```

5. Enter the data below in Excel with the file name = import_march.csv

a. Import the excel file into the Environment pane using read.table() function. Write the code.

```
data <- read.table("C:/Users/killy/Documents/pushed/RWorksheet_Magallanes#4a/import_march.csv", header="
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

```
View(data)
```

Using Conditional Statements (IF-ELSE) 6. Full Search

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.

```
number <- readline(prompt="Enter a number from 1-50")
```

```
## Enter a number from 1-50
```

```
number
```

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

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

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

7. Change

a.

```
min_bills <- function(price) {
  bill_count <- 0

  if (price >= 1000) {
    bill_count <- bill_count + price %/% 1000
```

```
    price <- price %% 1000
  }

  if (price >= 500) {
    bill_count <- bill_count + price %/% 500
    price <- price %% 500
  }

  if (price >= 200) {
    bill_count <- bill_count + price %/% 200
    price <- price %% 200
  }

  if (price >= 100) {
    bill_count <- bill_count + price %/% 100
    price <- price %% 100
  }

  if (price >= 50) {
    bill_count <- bill_count + price %/% 50
  }

  print(bill_count)
}
min_bills(1250)
```

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

8.

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

```
Math_Score <- 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))

Math_Score
```

```
##    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.

```r
for (i in 1:nrow(Math_Score)) {
  avg_score <- sum(Math_Score[i, 2:5]) / 4
  print(paste0(Math_Score$Name[i], "'s average grade this semester is: ", avg_score))
}
```

```
## [1] "Annie's average grade this semester is: 83.75"
## [1] "Thea's average grade this semester is: 80"
## [1] "Steve's average grade this semester is: 73.75"
## [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.

```r
for (j in 2:ncol(Math_Score)) {
  avg_test_score <- sum(Math_Score[, j]) / nrow(Math_Score)
  if (avg_test_score < 80) {
    test_number <- j - 1
    print(paste("The", test_number, "th test was difficult."))
  }
}
```

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

```r
for (i in 1:nrow(Math_Score)) {
  highest_score <- Math_Score[i, 2]
  for (j in 3:ncol(Math_Score)) {
    if (Math_Score[i, j] > highest_score) {
      highest_score <- Math_Score[i, j]  # Update if a higher score is found
    }
  }
 if (highest_score > 90) {
    print(paste(Math_Score$Name[i], "'s highest grade this semester is", highest_score))
  }
}
```

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