Lego,Worksheet#4

2023-10-25

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

```
library(readr)
Shoe_Data <- read_csv("Shoe Data.csv")
```

```
## Rows: 28 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Gender
## dbl (2): Shoe Size, Height
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Shoe_Data[,1:2]
```

```
## # A tibble: 28 x 2
##    `Shoe Size` Height
##          <dbl>  <dbl>
##  1         6.5   66
##  2         9     68
##  3         8.5   64.5
##  4         8.5   65
##  5        10.5   70
##  6         7     64
##  7         9.5   70
##  8         9     71
##  9        13     72
## 10         7.5   64
## # i 18 more rows
```

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

```
male<-subset(Shoe_Data, Gender == "M" )
male
```

```
## # 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
```

```
female<-subset(Shoe_Data, Gender == "F")
female
```

```
## # 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
```

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

```
mean(Shoe_Data$`Shoe Size`)
```

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

```
mean(Shoe_Data$`Shoe Height`)
```

```
## Warning: Unknown or uninitialised column: `Shoe Height`.
```

```
## Warning in mean.default(Shoe_Data$`Shoe Height`): argument is not numeric or
## logical: returning NA
```

```
## [1] NA
```

##d. Is there a relationship between shoe size and height? Why? ##Yes there is a relationship between shoe size and height because as you can see in the table taller individuals have larger shoe size.The taller the respondents is ther larger their shoe size is.

##2. Construct character vector months to a factor with factor() and assign the result tofactor_months_vector. ##Print out factor_months_vector and assert that R prints out the factor levels below the actual values.

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

```
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
```

Interpret the results of both vectors. Are they both equally useful in this case? ##The summary of "months_vector" will reflect the original order and frequency of months in data, while the "factor_months_vector" summary will display the months in a sorted order based on the levels of the factor. So, they may not be equally useful, and the choice between them depends on specific analysis needs.

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

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

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

```
frequency
```

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

```
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
```

```
new_order_data <- factor(direction,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

```
library(readr)
import_march <- read_csv("import_march.csv")
```

```
## Rows: 6 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Students
## dbl (3): Strategy 1, Strategy 2, Strategy 3
## 
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
import_march
```

```
## # A tibble: 6 x 4
##   Students `Strategy 1` `Strategy 2` `Strategy 3`
##   <chr>           <dbl>        <dbl>        <dbl>
## 1 Male                8           10            8
## 2 <NA>                4            8            6
## 3 <NA>                0            6            4
## 4 Female             14            4           15
## 5 <NA>               10            2           12
## 6 <NA>                6            0            9
```

##6. If else

```r
int6<- readline(prompt= "Enter a number 1 to 50")
```

```
## Enter a number 1 to 50
```

```r
if(int6>=50){
  print("THe number selected is beyond the range of 1")
}else if (int6==20){
  print (TRUE)
}else {
  int6
}
```

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

##7At 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.

```r
library(latexpdf)
bills<- c(50, 100, 200, 500, 1000)
price<-(readline(prompt= "Enter price a price: "))
```

```
## Enter price a price:
```

```r
  if (price == 50){
    cat("Minimum bills is: ")
    cat(bills[c(1)])
  }else if (price <= 150){
    cat("Minimum bills is: ")
    cat(bills [c(1,2)])
  }else if (price <= 450){
    cat("Minimum bills is: ")
    cat(bills [c(1,2,3)])
  }else if (price <= 950){
    cat("Minimum bills is: ")
    cat(bills [c(1,2,3,4)])
  }else if (price >= 1000){
    cat("Minimum bills is: ")
    cat(bills [c(1,2,3,4,5)])
  }
```

```
## Minimum bills is: 50 100
```

##8The 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)

math<-data.frame(
  Name = c(name),
  Grade1 = c(grade1),
  Grade2 = c(grade2),
  Grade3 = c(grade3),
  Grade4 = c(grade4)
)
math
```

```
##    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.

```
Average <- (math$Grade1 + math$Grade2 + math$Grade3 + math$Grade4) / 4
math$Average<-c(Average)
high_average_students <- math[math$Average >90 ]
i<-c(1:4)
if (math$Average[1]> 90){
  cat(math$Average[1], "'s average grade this semester is", math$Average[1], ".\n")
}else if (math$Average[2]> 90){
  cat(math$Average[2], "'s average grade this semester is", math$Average[2], ".\n")
}else if (math$Average[3]> 90){
  cat(math$Average[3], "'s average grade this semester is", math$Average[3], ".\n")
}else if (math$Average[4]> 90){
  cat(math$Average[4], "'s average grade this semester is", math$Average[4], ".\n")
}
```

##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.

```
test<-c("1st", "2nd", "3rd", "4th")
ave1<-sum(math$Grade1)/4
ave2<-sum(math$Grade2)/4
ave3<-sum(math$Grade3)/4
ave4<-sum(math$Grade4)/4

if (ave1<80){
  cat("The",test[1], "test was difficult")
}else if (ave2<80){
  cat("The",test[2], "test was difficult")
}else if (ave3<80){
  cat("The",test[3], "test was difficult")
}else if (ave4<80){
  cat("The",test[4], "test was difficult")
}
```

```
## The 2nd 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.

```
for (i in 1:4){
  if(math$Grade1[i]>90){
    cat(math$Name[i], "highest grade this semester is", round (math$Grade1[i],2), ".\n")
}}
```

```
## Hanna highest grade this semester is 95 .
```

```
for (i in 1:4){
  if(math$Grade2[i]>90){
    cat(math$Name[i], "highest grade this semester is", round (math$Grade2[i],2), ".\n")
}}
```

```
for (i in 1:4){
  if(math$Grade3[i]>90){
    cat(math$Name[i], "highest grade this semester is", round (math$Grade3[i],2), ".\n")
}}
```

```
## Hanna highest grade this semester is 100 .
```

```
for (i in 1:4){
  if(math$Grade4[i]>90){
    cat(math$Name[i], "highest grade this semester is", round (math$Grade4[i],2), ".\n")
}}
```

```
## Annie highest grade this semester is 100 .
```