RWorksheet_#4

Nicole Duero

2023-10-25

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

```
Shoe_size<-c(6.5,9.0,8.5,8.5,10.5,7.0,9.5,9.0,13.0,7.5,10.5,8.5,12.0,10.5,13.0,11.5,8.5,5.0,10.0,6.5,7.5
Height<-c(66.0,68.0,64.5,65.0,70.0,64.0,70.0,71.0,72.0,64.0,74.5,67.0,71.0,71.0,77.0,72.0,59.0,62.0,72.0
HouseholdData<-data.frame(Shoe_size,Height)
HouseholdData
```

```
##    Shoe_size Height
## 1        6.5   66.0
## 2        9.0   68.0
## 3        8.5   64.5
## 4        8.5   65.0
## 5       10.5   70.0
## 6        7.0   64.0
## 7        9.5   70.0
## 8        9.0   71.0
## 9       13.0   72.0
## 10       7.5   64.0
## 11      10.5   74.5
## 12       8.5   67.0
## 13      12.0   71.0
## 14      10.5   71.0
## 15      13.0   77.0
## 16      11.5   72.0
## 17       8.5   59.0
## 18       5.0   62.0
## 19      10.0   72.0
## 20       6.5   66.0
## 21       7.5   64.0
## 22       8.5   67.0
## 23      10.5   73.0
## 24       8.5   69.0
## 25      10.5   72.0
## 26      11.0   70.0
## 27       9.0   69.0
## 28      13.0   70.0
```

```
Gender<-c("F","F","F","F","M","F","F","F","M","F","M","F","M","M","M","M","F","F","M","F","F","M","M","M"
HouseholdDataNew<-cbind(HouseholdData, Gender)
HouseholdDataNew
```

```
##    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
```

ANSWER: The data provided contains information about individuals' shoe sizes, heights, and genders. Additionally, there are 28 observations in the dataset.

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

```
males<-subset(HouseholdDataNew, Gender=="M")
males
```

```
##    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
```

```
females<-subset(HouseholdDataNew, Gender=="F")
females
```

```
##    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
```

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

```
mShoe<-mean(HouseholdDataNew$Shoe_size)
mShoe
```

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

```
mHeight<-mean(HouseholdDataNew$Height)
mHeight
```

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

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

```
RelShoeHeight<- cor(HouseholdDataNew$Shoe_size, HouseholdDataNew$Height)
RelShoeHeight
```

```
## [1] 0.7766089
```

ANSWER: Yes, as it indicates a strong positive linear relationship between shoe size and height. This means that as shoe size increases, there is a tendency for height to also increase.

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.

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

ANSWER: Yes, they are both equally useful. If you view months as text labels, use `months_vector` as it's a character vector. For statistical operations or categorizing months, `factor_months_vector` is more suitable.

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

```
Direction<-c("East", "West","North")
Frequency<-c(1,4,3)
new_order_data<-factor(Direction, levels=c("East", "West","North"))
print(new_order_data)
```

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

```
new_order_data1<-factor(Frequency, levels=c(1,4,3))
print(new_order_data1)
```

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

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

```
read_importMarch<-read.table(file="C://Users//User//Documents//Rstudio Files//Worksheet#4//import_march
read_importMarch
```

```
##   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. 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.

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

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

```
if(UserNum == 20){
  print("TRUE")
}else if(UserNum<=50 && UserNum>=1){
   cat("The input number is:",UserNum)
}else{
    print("The number selected is beyond range of 1 to 50.")
}
```

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

7.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.

```
min_bills<-function(){
 price<-as.integer(readline(prompt="Price of snack(a random number divisible by 50):"))

 if (is.na(price)|| price %% 50 !=0){
  cat("Invalid. Should be divisible by 50.")
return()
}
  num_bills<-0
  bill_type<-c(1000,500,200,100,50)

  for(bill in bill_type){
    num_bills<-num_bills + (price %/% bill)
    price<-price %%bill
  }

cat("Minimum number of bills needed:", num_bills,"\n")
}
min_bills()
```

```
## Price of snack(a random number divisible by 50):
## Invalid. Should be divisible by 50.
```

```
## NULL
```

```
#8. 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)

StudentsScore<-data.frame(Name, Grade1, Grade2, Grade3, Grade4)
StudentsScore
```

```
##    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 o
students_above_90<-FALSE
for(g in 1:length(Name)){
  AveScore<-c((Grade1)[g]+(Grade2)[g]+(Grade3)[g]+(Grade4)[g])/4
  if(AveScore>90){
  cat(paste(Name[g], "'s average grade this semester is", round(AveScore,2),".\n"))
    students_above_90<-TRUE
  }
}
if(!students_above_90){
  print("No students has reached above 90 average math score.")
}
```

```
## [1] "No students has reached above 90 average math score."
```

```
#c. Without using the mean function, output as follows for the tests in which the average score was les
for (test_num in 1:4){
  AveScore2<-Grade1 + Grade2 + Grade3 + Grade4
  AveScore2<-AveScore2/4
  if (AveScore2[test_num]<80){
    cat("The", test_num, "test was difficult.\n")
  }
}
```

```
## The 3 test was difficult.
```

```
#d. Without using the max function, output as follows for students whose highest score for a semester e
for (h in 1:length(Name)){
  highGrade<-Grade1[h]

  if (Grade2[h]>highGrade){
    highGrade<-Grade2[h]
  }
  if (Grade3[h]>highGrade){
```

```
    highGrade<-Grade3[h]
  }
  if (Grade4[h]>highGrade){
    highGrade<-Grade4[h]
  }

  if (highGrade>90){
    cat(paste(Name[h], "'s highest grade this semester is", highGrade, ".\n"))
  }
}
```

```
## Annie 's highest grade this semester is 100 .
## Hanna 's highest grade this semester is 100 .
```