

## Worksheet-4a in R

# Trexan Sanico

2024-10-18

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

```
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)
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, 65.0, 66.0)
gender <- c("F", "F", "F", "F", "M", "F", "F", "F", "M", "F", "M", "F", "M", "M", "M", "M", "F", "F")

data <- data.frame(Shoe_size = shoe_size, Height = height, Gender = gender)
```

#a. Describe the data. - It shows 28 observations, of 3 variables, shoe\_size, height, and gender.

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

```
m_subset <- subset(data, gender == "M")
f_subset <- subset(data, gender == "F")
```

m\_subset

##	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

f\_subset

##	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
```

```
data
```

```
##      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
```

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

```
mean_of_shoe_size <- mean(shoe_size)
mean_of_height <- mean(height)

shoe_size_mean <- paste("Mean of shoe size",mean_of_shoe_size)
height_mean <- paste("Mean of height", mean_of_height)

shoe_size_mean
```

```
## [1] "Mean of shoe size 9.41071428571429"
```

```
height_mean
```

```
## [1] "Mean of height 68.5714285714286"
```

#d. Is there a relationship between shoe size and height? Why? #Yes, there is a relationship between shoe size and height because taller individuals tend to have larger feet.

#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", "July", "December", "August", "August", "September", "November", "February", "April")

```
months <- 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 <- factor(months)
factor_months
```

```
## [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?

#The factor assigns levels to the months based on alphabetical order. A summary of the factor would show how many times each month appears in the vector.

```
summary(months)
```

```
##      Length      Class      Mode
##      24 character character
```

```
summary(factor_months)
```

```
##      April      August  December  February   January      July      March      May
##          2          4          1          2          3          1          1          1
## November  October  September
##          5          1          3
```

#Create a vector and factor for the table below. #Note: Apply the factor function with required order of the level. new\_order\_data <- factor(factor\_data, levels = c("East", "West", "North")) print(new\_order\_data)

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

```
data <- data.frame(direction, frequency)
data
```

```
## direction frequency
## 1      East          1
## 2      West          4
## 3      North          3
```

```
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(readxl)
excelData <- read.table("/cloud/project/Worksheet-4a/import_march.csv", header = TRUE, sep = ",")
```

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

```
print(excelData)
```

```
##  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
```