

Rworksheet_Fegidero#3b

Jorealle Fegidero

2025-10-20

##1.a

```
household_data <- data.frame(  
  Sex = c("Male", "Female", "Male", "Female", "Female"),  
  No_of_Siblings_Attending_School = c(4, 6, 5, 2, 7),  
  Type_of_House = c("Wood", "Concrete", "Semi-Concrete", "Concrete", "Wood"),  
  Father_Occupation = c("Farmer", "Driver", "Others", "Farmer", "Driver"),  
  stringsAsFactors = FALSE  
)
```

##1.b

```
# Structure  
str(household_data)
```

```
## 'data.frame':    5 obs. of  4 variables:  
##  $ Sex                : chr  "Male" "Female" "Male" "Female" ...  
##  $ No_of_Siblings_Attending_School: num  4 6 5 2 7  
##  $ Type_of_House       : chr  "Wood" "Concrete" "Semi-Concrete" "Concrete" ...  
##  $ Father_Occupation   : chr  "Farmer" "Driver" "Others" "Farmer" ...
```

```
# Summary  
summary(household_data)
```

```
##      Sex                No_of_Siblings_Attending_School Type_of_House  
## Length:5                Min.      :2.0                      Length:5  
## Class :character        1st Qu.:4.0                      Class :character  
## Mode  :character        Median :5.0                      Mode  :character  
##                        Mean   :4.8  
##                        3rd Qu.:6.0  
##                        Max.   :7.0  
## Father_Occupation  
## Length:5  
## Class :character  
## Mode  :character  
##  
##  
##
```

##1.c

```
mean(household_data$No_of_Siblings_Attending_School)
```

```
## [1] 4.8
```

```
##1.d
```

```
household_data[1:2, ]
```

```
##      Sex No_of_Siblings_Attending_School Type_of_House Father_Occupation
## 1  Male                                4           Wood           Farmer
## 2 Female                                6           Concrete          Driver
```

```
##1.e
```

```
household_data[c(3,5), c(2,4)]
```

```
##      No_of_Siblings_Attending_School Father_Occupation
## 3                                5           Others
## 5                                7           Driver
```

```
##1.f
```

```
types_houses <- household_data$Type_of_House
types_houses
```

```
## [1] "Wood"           "Concrete"          "Semi-Concrete" "Concrete"
## [5] "Wood"
```

```
##1.g
```

```
subset(household_data, Sex == "Male" & Father_Occupation == "Farmer")
```

```
##      Sex No_of_Siblings_Attending_School Type_of_House Father_Occupation
## 1 Male                                4           Wood           Farmer
```

```
##1.h
```

```
subset(household_data, Sex == "Female" & No_of_Siblings_Attending_School >= 5)
```

```
##      Sex No_of_Siblings_Attending_School Type_of_House Father_Occupation
## 2 Female                                6           Concrete          Driver
## 5 Female                                7           Wood           Driver
```

```
##1.2
```

```
df <- data.frame(
  Ints = integer(),
  Doubles = double(),
  Characters = character(),
  Logicals = logical(),
  Factors = factor(),
  stringsAsFactors = FALSE
)

print("Structure of the empty dataframe:")
```

```
## [1] "Structure of the empty dataframe:"
```

```
print(str(df))
```

```
## 'data.frame': 0 obs. of 5 variables:
## $ Ints : int
## $ Doubles : num
## $ Characters: chr
## $ Logicals : logi
## $ Factors : Factor w/ 0 levels:
## NULL
```

```
##2.a
```

```
##The data frame is initialized with 0 rows but 5 columns with specified types: integer, numeric, character, logical, factor
```

```
##3.a
```

```
write.csv(household_data, "HouseholdData.csv", row.names = FALSE)
```

```
##3.b
```

```
data_imported <- read.csv("HouseholdData.csv", stringsAsFactors = FALSE)
```

```
##3.c
```

```
data_imported$Sex <- factor(data_imported$Sex, levels = c("Male", "Female"))
data_imported$Sex <- as.integer(data_imported$Sex)
data_imported$Sex
```

```
## [1] 1 2 1 2 2
```

```
##3.d
```

```
data_imported$Type_of_House <- factor(data_imported$Type_of_House,
                                     levels = c("Wood", "Concrete", "Semi-Concrete"))
data_imported$Type_of_House <- as.integer(data_imported$Type_of_House)
data_imported$Type_of_House
```

```
## [1] 1 2 3 2 1
```

```
##3.e
```

```
data_imported$Father_Occupation <- factor(data_imported$Father_Occupation,  
                                           levels = c("Farmer", "Driver", "Others"))  
data_imported$Father_Occupation <- as.integer(data_imported$Father_Occupation)  
data_imported$Father_Occupation
```

```
## [1] 1 2 3 1 2
```

```
##3.f
```

```
subset(data_imported, Sex == 2 & Father_Occupation == 2)
```

```
##   Sex No_of_Siblings_Attending_School Type_of_House Father_Occupation  
## 2    2                               6              2              2  
## 5    2                               7              1              2
```

```
##3.g
```

```
subset(data_imported, No_of_Siblings_Attending_School >= 5)
```

```
##   Sex No_of_Siblings_Attending_School Type_of_House Father_Occupation  
## 2    2                               6              2              2  
## 3    1                               5              3              3  
## 5    2                               7              1              2
```

```
##4
```

```
install.packages("ggplot2") # if not installed
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.5'  
## (as 'lib' is unspecified)
```

```
library(ggplot2)
```

```
# Create the data frame  
sentiment_data <- data.frame(  
  Date = rep(as.Date(c("2020-07-14", "2020-07-15", "2020-07-17",  
                       "2020-07-18", "2020-07-20", "2020-07-21")), each = 3),  
  Sentiment = rep(c("Negative", "Neutral", "Positive"), times = 6),  
  Count = c(2500, 1500, 1800, # July 14  
            4300, 2900, 3200, # July 15  
            3100, 1800, 2500, # July 17  
            3300, 2000, 2200, # July 18  
            2400, 1500, 1700, # July 20  
            4100, 2700, 3400) # July 21  
)
```

```
ggplot(sentiment_data, aes(x = factor(Date), y = Count, fill = Sentiment)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Sentiments Of Tweets Per Day",
       x = "Day of Date / Sentiment",
       y = "Count of Tweets") +
  theme_minimal() +
  scale_fill_manual(values = c("Negative" = "red", "Neutral" = "orange", "Positive" = "blue"))
```

