title: "Exercise3"
author: "Thai pham - T00727094"
date: "2024-03-07"
output:
pdf_document: default

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.2
```

```
## Warning: package 'tibble' was built under R version 4.3.2
```

```
## Warning: package 'tidyr' was built under R version 4.3.2
```

```
## Warning: package 'readr' was built under R version 4.3.2
```

```
## Warning: package 'purrr' was built under R version 4.3.2
```

```
## Warning: package 'dplyr' was built under R version 4.3.2
```

```
## Warning: package 'stringr' was built under R version 4.3.2
```

```
## Warning: package 'forcats' was built under R version 4.3.2
```

```
## Warning: package 'lubridate' was built under R version 4.3.2
```

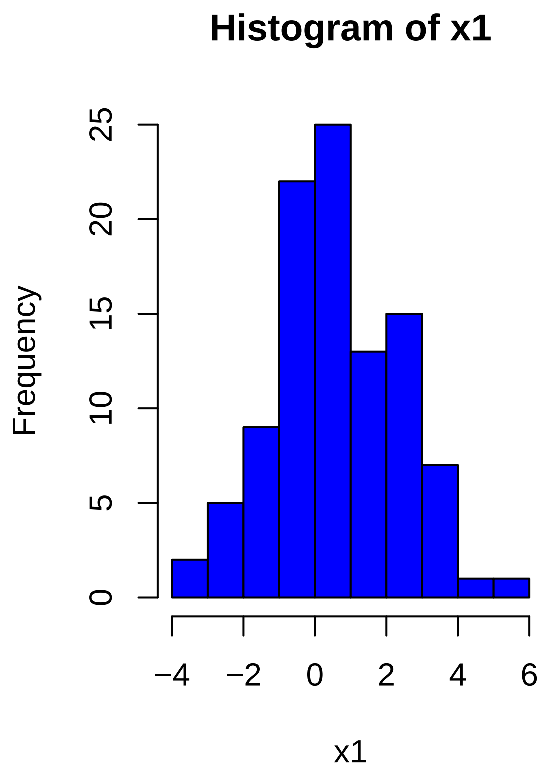
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

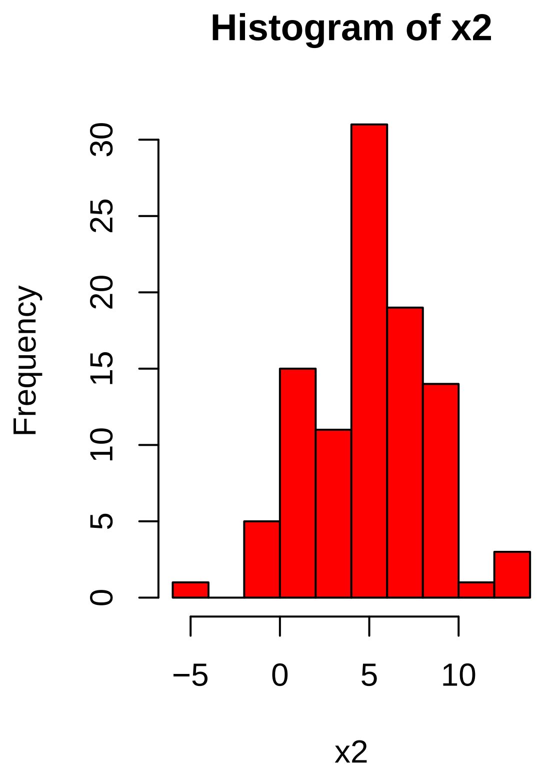
```
library(ggplot2)
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.3.2
```

# 1. Generate two vectors, x1 and x2, containing 100 observations drawn from N(1.1, 3.5) and N(5, 10), respectively. Plot the data in side by side histogram with different colors and check normality. Make sure to use set.seed(5420) prior to starting to ensure consistent results. [2.5]

```
set.seed(5420)
x1<- rnorm(100, mean = 1.1, sd = sqrt(3.5))
x2<- rnorm(100, mean = 5, sd = sqrt(10))
par(mfrow = c(1, 2))
hist(x1, col="blue")
hist(x2, col="red")
```

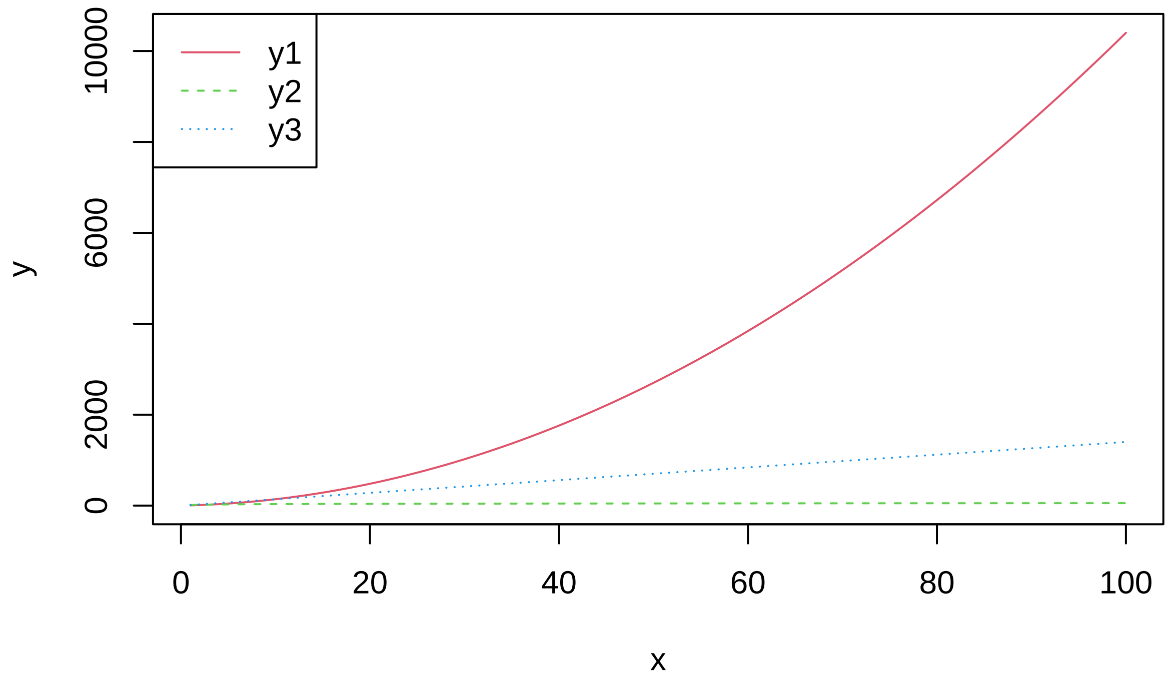



Based on the histograms, it can be said that both data follow nomal distribution.

## 2. Generate a line plot that shows the values of three functions for the inputs 0 to 100. Use different colors for three functions and label the three functions with proper legend. [2.5]

```
x <- 1:100
y1 <- x^2 + 4*x
y2 <- 9*log(x) + 12
y3 <- 14*x
plot(x,y1, type="l", lty=1,col=2, ylab="y", xlab="x")
lines(x, y2, lty=2, col=3)
lines(x, y3, lty=3,col=4)
legend("topleft", legend = c("y1", "y2", "y3"), lty=1:3, col = 2:4)
```




## 3.

(a) Make a set of data set with 4 variables. For 2 of the variables, the data should be partially correlated and 1 variable should not be correlated and 1 variable will be a categorical variable with 3 groups (say, 1,2 and 3). Generate a pairwise scatter plot of this data by categorical variable. Use different colors and points/shape for each groups of categorical variable. Give a proper title of the plot. Make sure to use set.seed(5420) prior to starting to ensure consistent results. [2.5]

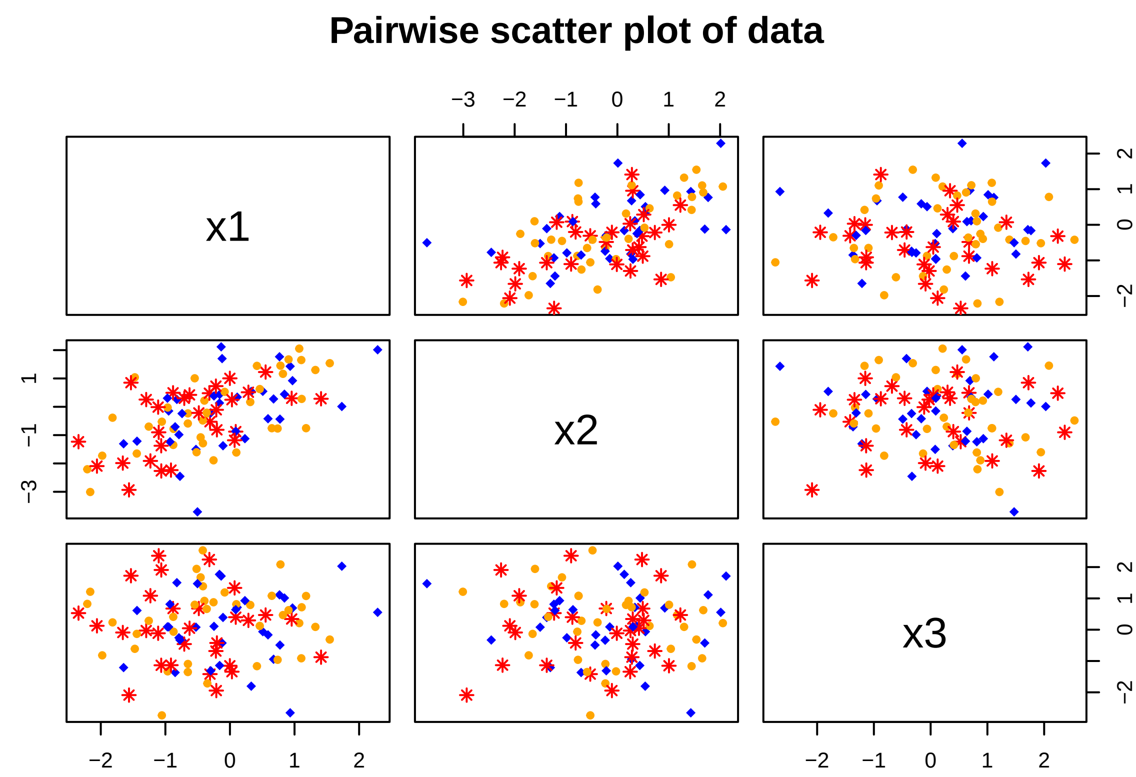
```r
set.seed(5420)
# Number of observations
n <- 100
# Generate partially correlated variables
x1 <- rnorm(n)
x2 <- x1 + rnorm(n, mean = 0, sd = 1)
# Generate uncorrelated variable
x3 <- rnorm(n)
# Generate categorical variable with three groups
cat_var <- sample(1:3, n, replace = TRUE)
# Combine variables into a data frame
data <- data.frame(x1, x2, x3, cat_var)
# Display first few rows of the dataset
head(data)
```

```
##           x1         x2          x3 cat_var
## 1 -1.2965681  0.2551352 -0.02714719       3
## 2 -1.0635849 -2.2650139  1.90807193       3
## 3 -0.1215601  1.7010522 -0.42527421       1
## 4 -0.3089153 -0.5285024 -1.41879417       3
## 5 -0.1074952 -1.3751796  0.39133902       1
## 6 -0.8731790 -0.7798999 -0.06420372       2
```

```r
# Check correlation matrix
correlation_matrix <- cor(data)
# Print correlation matrix
print(correlation_matrix)
```

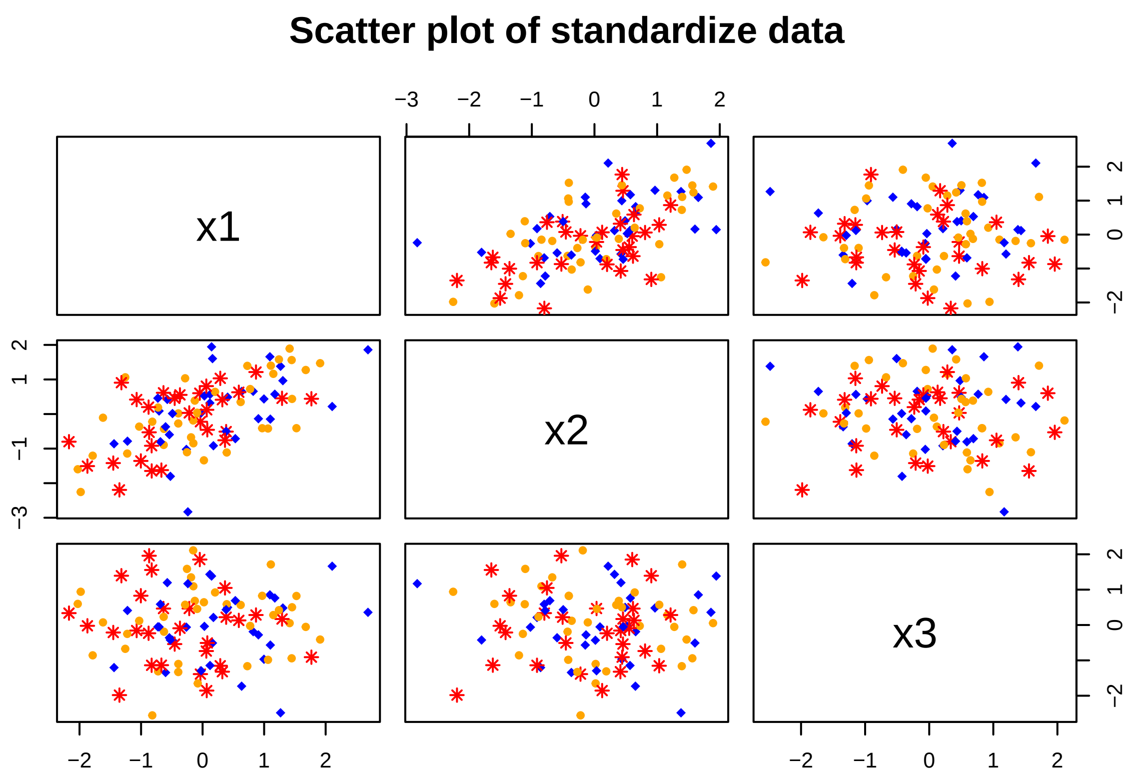
```
##                  x1          x2          x3     cat_var
## x1       1.00000000  0.59074726  0.04338939 -0.23167614
## x2       0.59074726  1.00000000 -0.05950376 -0.10761375
## x3       0.04338939 -0.05950376  1.00000000 -0.02143918
## cat_var -0.23167614 -0.10761375 -0.02143918  1.00000000
```

```r
# Define colors and points by group
colors <- c("blue", "orange", "red")
points <- c(18, 20, 8)

# Create scatter plot
pairs(data[, 1:3],
      col = colors[data$cat_var],  # Change color by group
      pch = points[data$cat_var],  # Change points by group
      main = "Pairwise scatter plot of data")
```




**(b) Standardize (or normalize) the data from part 3(a) and draw another plot with the scaled data. Remember we only standardize the quantitative variables. [2.5]**

```
# Standardize (normalize) quantitative variables
data_scaled<- scale(data[1:3])

# Create scatter plot with scaled data
pairs(data_scaled[, 1:3],
      col = colors[data$cat_var],  # Change color by group
      pch = points[data$cat_var],  # Change points by group
      main = "Scatter plot of standardize data")
```




4. In this exercise you will create some simulated data and will fit simple linear regression models to it. Make sure to use set.seed(542024) prior to starting part (a) to ensure consistent results. [10]

(a) Using the 'rnorm()' function, create a vector, x, containing 100 observations drawn from a N(0, 1.5) distribution. This represents a feature, X.

```
set.seed(542024)
x<-rnorm(100,mean = 0, sd = sqrt(1.5))
```

(b) Using the 'rnorm()' function, create a vector, e, containing 100 observations drawn from a normal distribution with mean zero and variance 0.20.

```
e<-rnorm(100, mean = 0, sd = sqrt(0.20))
```

(c) Using x and e, generate a vector y according to the model Y = 2.2 + 0.8X + e (1)

```
y<--2.2 + 0.8*x + e
```

### (d) What is the length of the vector y? What are the values of 0 and 1 in this linear model? Comment on the correlation between y and x based on this model

```r
#Length of vector y
length(y)
```

```
## [1] 100
```

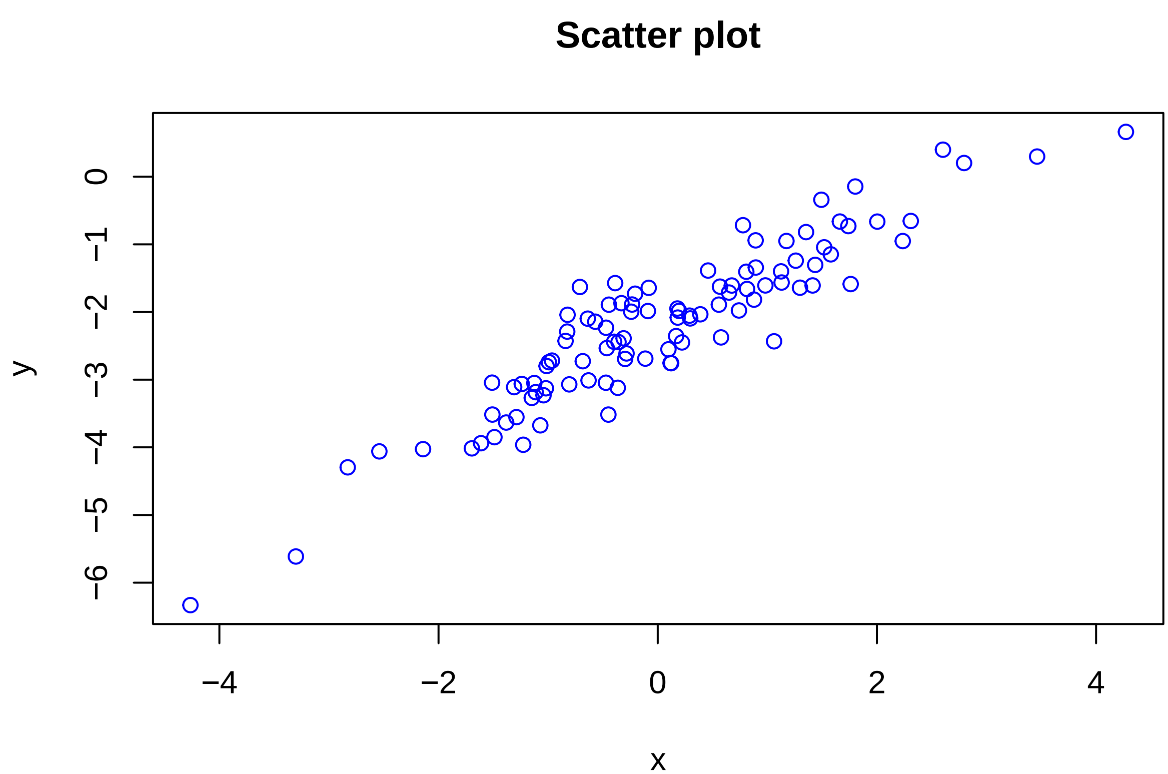
```r
#Correlation between y and x
cor(x,y)
```

```
## [1] 0.9273727
```

In this linear model, 0 = -2.2, 1 = 0.8. The correlation coefficient of 0.90756 indicates a relatively high level of correlation, suggesting a strong positive relationship between variables x and y.

### (e) Create a scatter plot displaying the relationship between x and y. Comment on what you observe.

```r
plot(x,y, col = "blue", main = "Scatter plot")
```




Based on the scatter plot, there appears to be a linear relationship between variables x and y, albeit with some noise introduced by variable e.