# Practice Session 3_answers

## Part 1 : The relationship between two Quantitative Variables/ Linear Regression

Remember that the fitted regression line is defined by the equation:

- $\hat{y} = a + bx$ , or
- $\widehat{Response} = a + b(Explanatory)$
- Residuals= observed- predicted

Where:

- Response: is the response variable or the dependent variable
- Explanatory: is the independent variable
- a: is the y- intercept
- b: is the slope of the regression line

You may use the following R functions: `plot()`, `lm()`, `cor()`, `abline()`. And you might need to download Lock5Data using Library(Lock5Data ).

**Practice 1:**

State if the following sentences are true or false.

a. We choose the linear model that passes through the most data points on scatter plot.

b. The residuals are observed y-values minus the y-value predicted by linear model.

c. Least square means that the square of the largest residuals is as small as it could be possibly be.

d. Some of the residuals from least linear model will be positive and some will be negative.

e. Least squares means that some of the squares of the residuals are minimized.

f. We write $\hat{y}$ to denote the predicted values and the y to denote the observed value.

**Answers:**

a. False; we choose a model that minimizes the sum of squared residuals (i.e., sum of squared distances between each point and the regression line).

b. True; this is the definition of a residual.

c. False; the least squares method minimizes the **total** sum of squared distances, not the largest squared distance. It takes into account all observations, not a single observation.

d. True; we see this from the residual plot normally; some values are above 0, others are below.

e. False; the least squares method minimizes the **total** sum of squared distances. The least squares method does not minimize individual squared

f. True; this is the definition of $y$ and $\hat{y}$ in regression model.

**Practice 2:**

Use data `FloridaLakes` and the two variables `Alkalinity` and `calcium`.

```
# download the data and load it into R
library(Lock5Data)
data(FloridaLakes)
```

1. Find the correlation for the `Alkalinity` and `calcium`.

```
# your code here
```

2. Using the appropriate R function, create a linear model to predict `Alkalinity` from `calcium`.

```
# your code here
```

3. Find the coefficients of the regression and write the correct equation of this linear model.

```
# your code here
```

4. Interpret the intercept and slope of the model within the context of the data.
5. Predict the `Alkalinity` level when Calcium = 2.5.

```
# your code here
```

6. The actual `Alkalinity` level was 8.5 when Calcium = 2.5. Find the residual for this data point.

```
# your code here
```

7. Find the five number summary for the variable `Calcium`.

```
# your code here
```

8. Predict the `Alkalinity` level when Calcium = 0.5.

```
# your code here
```

9. Do both of these predictions make sense given the data?

```
# your code here
```

**Answers:**

1. Find the correlation for the `Alkalinity` and `calcium`.

```
alk <- FloridaLakes$Alkalinity
cal <- FloridaLakes$Calcium

cor_alk_cal<- cor(alk , cal)
cor_alk_cal
```

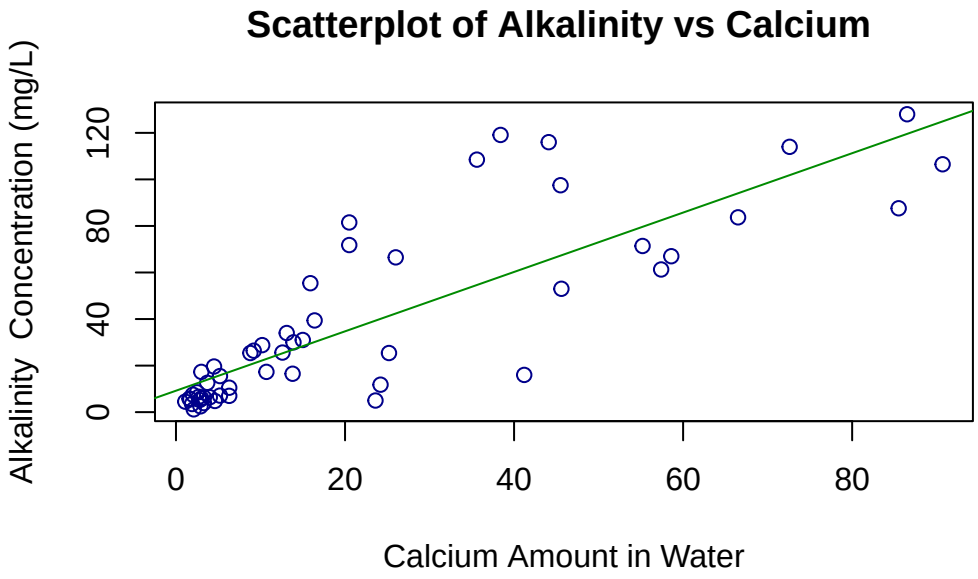
```
[1] 0.8326042
```

2. Using the appropriate R function, create a linear model to predict `calcium` from `pH`.

```r
# Fit a linear regression model data and add the regression line to the plot

alk_cal_fit <- lm( alk ~ cal)

alk_cal_fit
```

```
Call:
lm(formula = alk ~ cal)

Coefficients:
(Intercept)          cal
      9.206        1.276
```

```r
# Add the regression line to the plot
plot( alk ~ cal,

     main = "Scatterplot of Alkalinity vs Calcium ",
     xlab = "Calcium Amount in Water",
     ylab = "Alkalinity  Concentration (mg/L)",
      col = "blue4")


abline(alk_cal_fit, col = "green4")
```




3. Find the coefficients of the regression and write the correct equation of this linear model.

after running the following code you will get the intercept and the slope values of the regression line equation as follows :

$\widehat{Alkalinity} = 9.206 + 1.276 \cdot Calcium.$

```
# you can use one of the following methods:

## method 1

print(alk_cal_fit )
```

```
Call:
lm(formula = alk ~ cal)

Coefficients:
(Intercept)          cal
      9.206        1.276
```

```
## or method 2

reg_coef<-coef(alk_cal_fit)
reg_coef
```

```
(Intercept)         cal
   9.205522    1.275777
```

4. Interpret the intercept and slope of the model within the context of the data.

"Slope": for each increase of "one" unit in the calcium level there will be an increase of "1.276" in the Alkalinity level.

"Intercept": when calcium level is set to "0" the Alkalinity level will be 9.206

5. Predict the `Alkalinity` level when Calcium = 2.5.

```
# predict Alkalinity when calcium= 2.5

## method 1
(predicted_alk_2.5 <- reg_coef[1] + reg_coef[2] * 2.5)
```

```
(Intercept)
   12.39497
```

```
## method 2
(predicted_alk_0.5 <- 9.206 + 1.276 * 0.5)
```

```
[1] 9.844
```

```
## method 3: using the function `predict` and the data 39 corresponding to calcium= 2.5
(predicted_alk_2.5 <- predict(alk_cal_fit)[39])
```

```
      39
12.39497
```

6. The actual `Alkalinity` level was 8.5 when Calcium= 2.5. Find the residual for this data point.

```r
# residuals  when Calcium =2.5

## method 1:
 (res_2.5<-  8.5 - predict(alk_cal_fit)[39] )
```

```
       39
-3.894966
```

```r
# method 2:
(res_2.5<- alk_cal_fit$residuals[39])
```

```
       39
-3.894966
```

7. Find the five number summary for the variable `Calcium`.

```r
fivenum(cal)
```

```
[1]  1.1  3.3 12.6 35.6 90.7
```

8. Predict the `Alkalinity` level when Calcium = 0.5

```r
(predicted_alk_0.5 <- reg_coef[1] + reg_coef[2] * 0.5)
```

```
(Intercept)
   9.843411
```

9. Do both of these predictions make sense given the data?

The level of Calcium = 2.5 is within the range of the data. But the Level of Calcium= 0.5 id an extrapolation.

## Part 2: Review

**Practice 3:**

From the data `ICUAdmissions` create descriptive statistics ( such as: `boxplot`, `histogram`, `five number summary` ) for the variables `Infection`, `Age`, `Race`, and `HeartRate`.

1. Create descriptive statistics using `one quantitative variable`.
2. Create descriptive statistics using `one categorical variable`.
3. Descriptive statistics for `two quantitative variable`.
4. Descriptive statistics for `one quantitative and one categorical variables`.

**Answers:**

```
# download the data and load it into R
library(Lock5Data)
data(ICUAdmissions)

str(ICUAdmissions)
```

```
'data.frame':   200 obs. of  21 variables:
 $ ID           : int  8 12 14 28 32 38 40 41 42 50 ...
 $ Status       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Age          : int  27 59 77 54 87 69 63 30 35 70 ...
 $ Sex          : int  1 0 0 0 1 0 0 1 0 1 ...
 $ Race         : int  1 1 1 1 1 1 1 1 2 1 ...
 $ Service      : int  0 0 1 0 1 0 1 0 0 1 ...
 $ Cancer       : int  0 0 0 0 0 0 0 0 0 1 ...
 $ Renal        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Infection    : int  1 0 0 1 1 1 0 0 0 0 ...
 $ CPR          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Systolic     : int  142 112 100 142 110 110 104 144 108 138 ...
 $ HeartRate    : int  88 80 70 103 154 132 66 110 60 103 ...
 $ Previous     : int  0 1 0 0 1 0 0 0 0 0 ...
 $ Type         : int  1 1 0 1 1 1 0 1 1 0 ...
 $ Fracture     : int  0 0 0 1 0 0 0 0 0 0 ...
 $ PO2          : int  0 0 0 0 0 1 0 0 0 0 ...
 $ PH           : int  0 0 0 0 0 0 0 0 0 0 ...
 $ PCO2         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Bicarbonate  : int  0 0 0 0 0 1 0 0 0 0 ...
 $ Creatinine   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Consciousness: int  1 1 1 1 1 1 1 1 1 1 ...
```

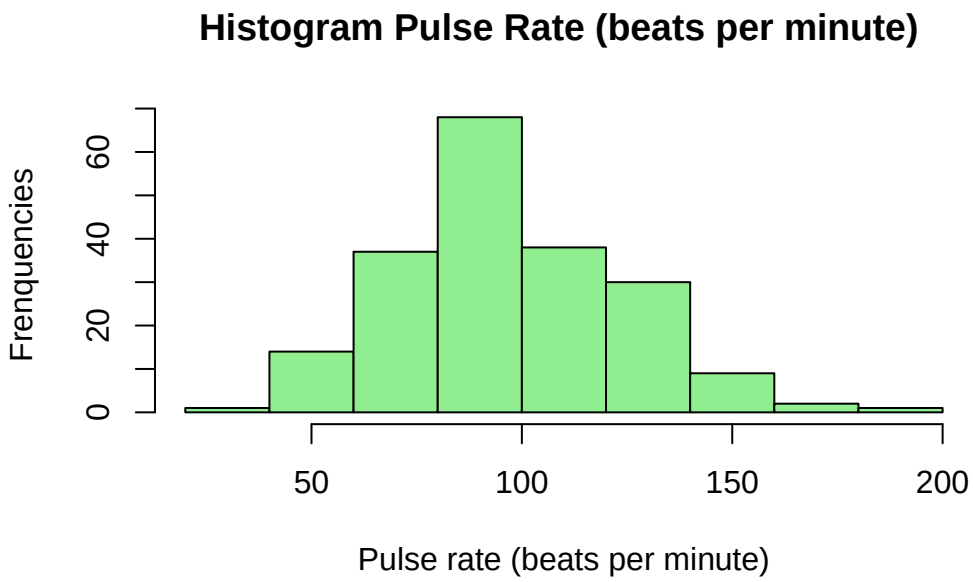
## It seems that the variables types are:

- Age : quantitative variable
- Infection: Categorical variable
- Race : Categorical variable
- HeartRate: Quantitative variable

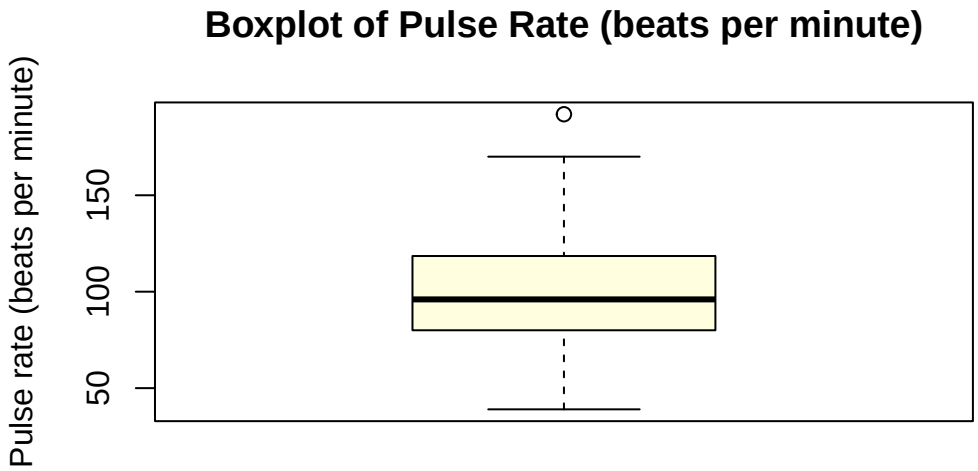
## 1. Descriptive stats for " one quantitative variable"

we can describe a quantitative variable with : `histogram`, `five number summary`, or `boxplot`

```
#par(mfrow= c(1,2) )

# Histogram for heart rate
heart_rate<- ICUAdmissions$HeartRate # Heart Rate - Pulse rate (beats per minute)

hist(heart_rate,
     main = "Histogram Pulse Rate (beats per minute) ",
     xlab = "Pulse rate (beats per minute)",
     ylab = "Frenquencies",
      col = "lightgreen")
```




```
# Boxplot for heart rate

boxplot(heart_rate,
     main = " Boxplot of Pulse Rate (beats per minute) ",
     ylab = "Pulse rate (beats per minute)",
      col = "lightyellow")
```




```
# Five numbers of age
age <- ICUAdmissions$Age
five_numbers_age <- fivenum(age)
five_numbers_age
```

```
[1] 16.0 46.5 63.0 72.0 92.0
```

## 2. Descriptive stats for " one categorical variable":

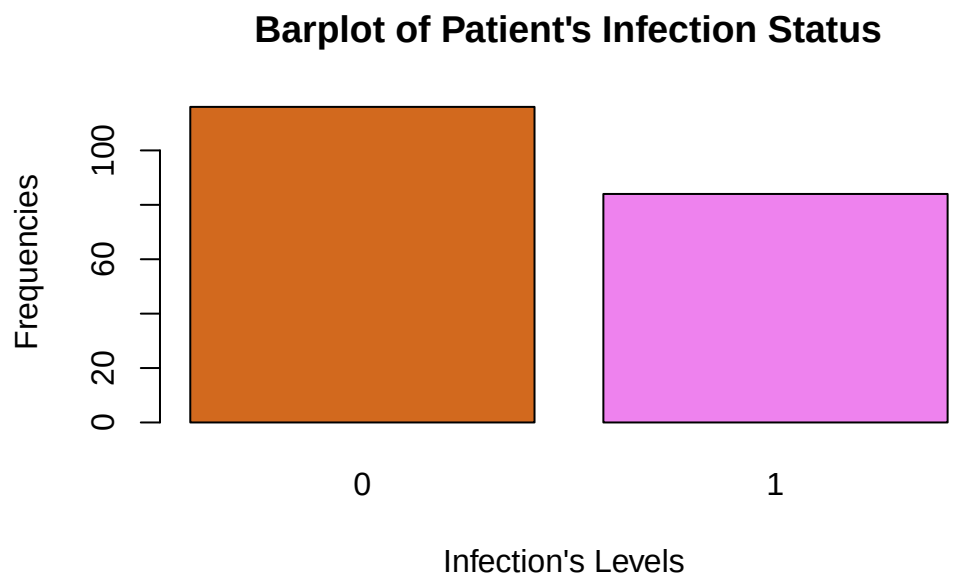
we can describe a categorical variable with : `bar plot` , `table`

```
#one categorical variable such as ( Infection)

## one way table ( Infection)

infection <-ICUAdmissions$Infection
```

```r
infection_table<- table(infection )
infection_table
```

```
infection
  0   1
116  84
```

```r
# create barplot
barplot(infection_table,

     main = " Barplot of Patient's Infection Status ",
     xlab = "Infection's Levels ",
     ylab = " Frequencies",
      col = c( "chocolate", "violet"))
```



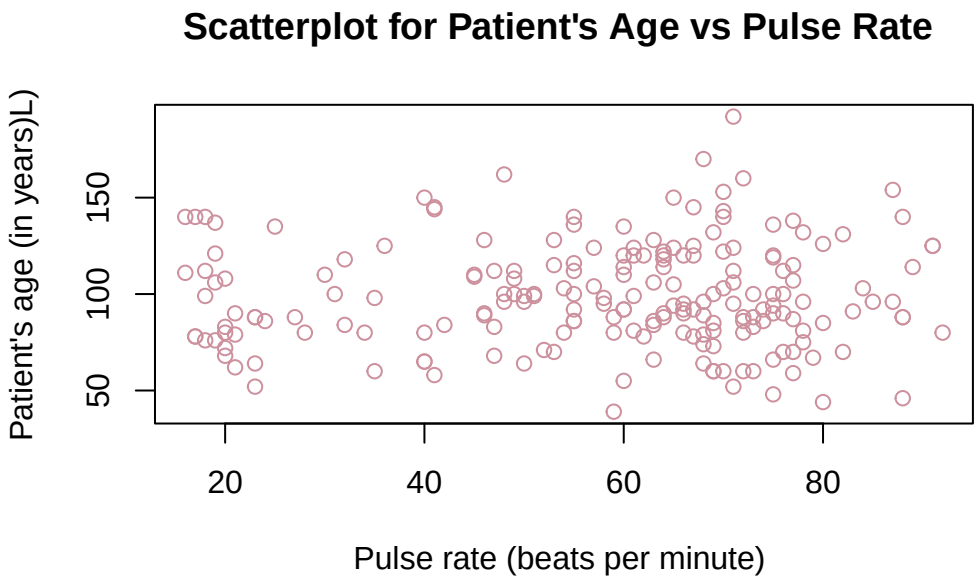

```r
## one way table proportions
(proportions<- prop.table(infection_table))
```

```
infection
   0    1
0.58 0.42
```

## 3. Descriptive stats for " two quantitative variable" :

we can describe two quantitative variable with: `scatter plot` , `correlation`

```
# scatter plot for Age and HeartRate
plot(age,  heart_rate,
     main = "Scatterplot for Patient's Age vs Pulse Rate ",
     xlab = "Pulse rate (beats per minute) ",
     ylab = "Patient's age (in years)L)",
      col = "pink3")
```




```
# correlation
cor(age, heart_rate)
```

```
[1] 0.03736843
```

## 4. Descriptive stats for " one quantitative variable vs one categorical variable" :

we can describe, `age` ( quantitative) versus `Infection` (categorical variable) with `boxpot`.

```
# boxplot of age( quantitative) versus categorical variable Infection( which has two levels )

boxplot(age  ~ infection,
     main = "Boxplot of Patient's Age vs Infection levels ",
     xlab = " Infection Levels  ",
     ylab = "Patient's age (in years)",
      col = c( "orange1", "yellow3"))
```