CW1_ANS_1

No, it is not in the form of a data frame as the response variable is “Time measurement in min” and, therefore, not all values of the response variable are contained in the same column.

 # CREATED DATA FRAME AS BELOW:

Time_Measured_min	Musical_genre
2.0	reggae
2.5	reggae
3.0	heavy metal
3.2	heavy metal
3.5	reggae
4.0	blues
4.2	blues
4.5	heavy metal
5.0	classical music
5.1	blues
5.5	jazz
6.4	jazz
7.0	jazz
10.0	classical music
12.5	classical music

 # CREATED DATA FRAME BY R-STUDIO CODES AS BELOW:

 # 1st step:

#Create the first vector with the name Time_Measured_min.

```
Time_Measured_min <-c(5,10,12.5,7,5.5,6.4,4,4.2,5.1,2,2.5,3.5,3,4.5,3.2)
```

Create a second vector with the name Muscial_genre and in this vector use the repeat function to match with value #of Time_Measured_min.

```
Musical_genre <-c(rep("classical music",3),rep("jazz",3),rep("blues",3),rep("reggae",3),rep("heavy metal",3))
```

 # 2nd step:

#create a data frame

```
Musical_Genre_DataFrame <-data.frame (Time_Measured_min,Musical_genre)
```

 # 3rd step:

#measured time in min is in ascending order

```
df <-Musical_Genre_DataFrame[order(Musical_Genre_DataFrame$Time_Measured_min),]
```

```
df
```

 # Output:

```
   Time_Measured_min   Musical_genre
10                2.0          reggae
11                2.5          reggae
13                3.0     heavy metal
15                3.2     heavy metal
12                3.5          reggae
7                 4.0           blues
8                 4.2           blues
14                4.5     heavy metal
1                 5.0 classical music
9                 5.1           blues
5                 5.5            jazz
6                 6.4            jazz
4                 7.0            jazz
2                10.0 classical music
3                12.5 classical music
> |
```

CW1_ANS_2.a

 # 1st step:

read the "wallaby-nr-47.csv" dataset by using the read.csv command and

check variables and values in a dataset by using the head command

```
data <- read.csv("D:/Pranav -UK/Applied Statistics - R language/CW-01/wallaby-nr-47.csv")
head(data)
```

 # 2nd step:

construct a vector p that consists of the 3rd, 5th, and 7th variables of the dataset.

```
p <-data[,c(3,5,7)]
```

 # 3rd step:

#Using p, list the names of the variables.

```
list <-variable.names(p)
list
```

 # Output:

```
> list
[1] "Ear"  "Leg"  "Tail"
> |
```

CW1_ANS_2.b_i

the mean, median, max, and min values of the variable Ear

```
ear_mean <-mean(data$Ear)

ear_mean

ear_median<-median(data$Ear)

ear_median

ear_max <-max(data$Ear)

ear_max

ear_min <-min(data$Ear)

ear_min
```

 # Output:

```
> ear_mean <-mean(data$Ear)
> ear_mean
[1] 670.1667
> ear_median<-median(data$Ear)
> ear_median
[1] 676
> ear_max <-max(data$Ear)
> ear_max
[1] 707
> ear_min <-min(data$Ear)
> ear_min
[1] 613
```

CW1_ANS_2.b_ii

To find the time point in the study when ear length is above the mean, But below the median ear length.

The only available variable is 'Leng' in a dataset. Therefore, we can consider the 'Leng' variable for the time point in our further analysis

```
ear_d <- data$Ear

Leng_time_point <- data$Leng[ear_d > ear_mean & ear_d < ear_median]

Leng_time_point#output
```

```
> ear_d <- data$Ear
> Leng_time_point <- data$Leng[ear_d > ear_mean & ear_d < ear_median]
> Leng_time_point
[1] 1002
```

 # Interpretation:

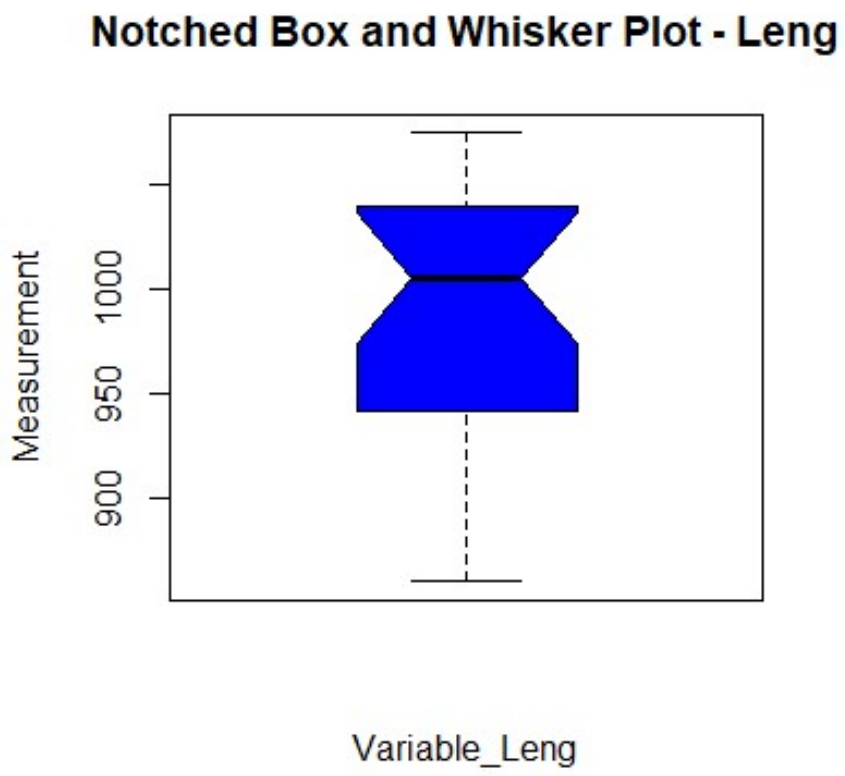
The periods in the research where ear length is greater than the mean but less than the median reflect specific lengths or times in time when Wallaby 47 showed this ear length behavior. Which might be a crucial stage of development.

 # Carefulness: It is prudent to treat measurements for the variable "Ear" with care because they can be influenced by a variety of variables, including measurement mistakes and individual variability. Outliers and anomalies can influence the mean and median, highlighting the importance of thorough data validation and management to assure the accuracy of results.

CW1_ANS_2.c_i

```
# Create Notched Box and Whisker plot for variable name Leng

boxplot(data$Leng,notch = TRUE,names = c("length"),

    col = 'blue',main="Notched Box and Whisker Plot - Leng",

    xlab="Variable_Leng",ylab="Measurement")
```

 # Output



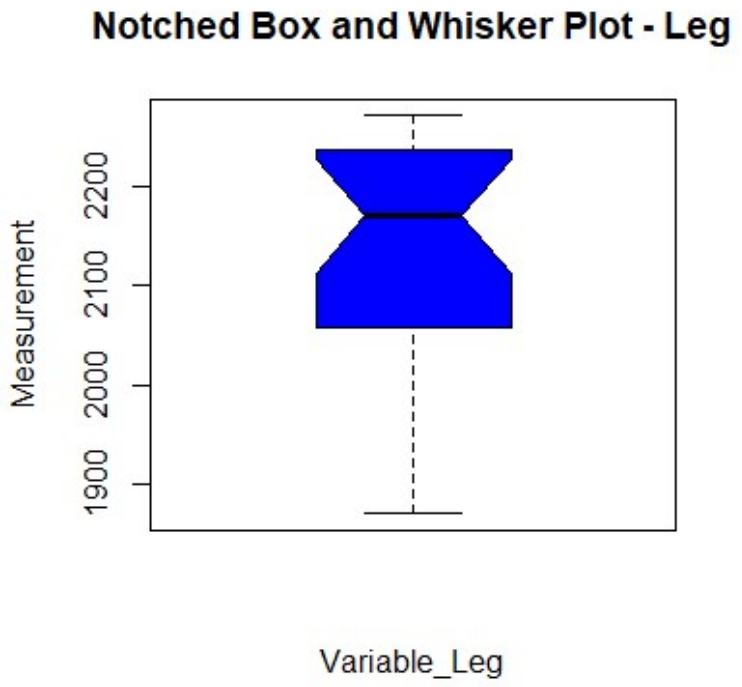
 # Interpretation:

1. **Box** represents the Interquartile range(IQR) of the given variable Leng data, which is the range between the 1st quartile (Q1) around 950 (a tenth of a millimeter), and the 3rd quartile(Q3) around 1030 (a tenth of a millimeter). The above graph shows that the wider the box means data of variable Leng is more spread out.
2. **Notches** are overlapped means there may not be a statistically significant difference in the medians.
3. **Median Values:** The horizontal lines within the box represent the median of the dataset. For the "Length" variable, the median is approximately 1000 (a tenth of a millimeter) on the y-axis.
4. **Whiskers**: the above graph shows that Whiskers are below the value of 900 (a tenth of a millimeter) and above 1050 (a tenth of a millimeter) and beyond this range are outliers.

 # Create Notched Box and Whisker plot for variable name leg

```
boxplot(data$Leg,notch = TRUE,names = c("leg"),
    col = 'blue',main="Notched Box and Whisker Plot - Leg",
    xlab="Variable_Leg",ylab="Measurement")
```

 # Output



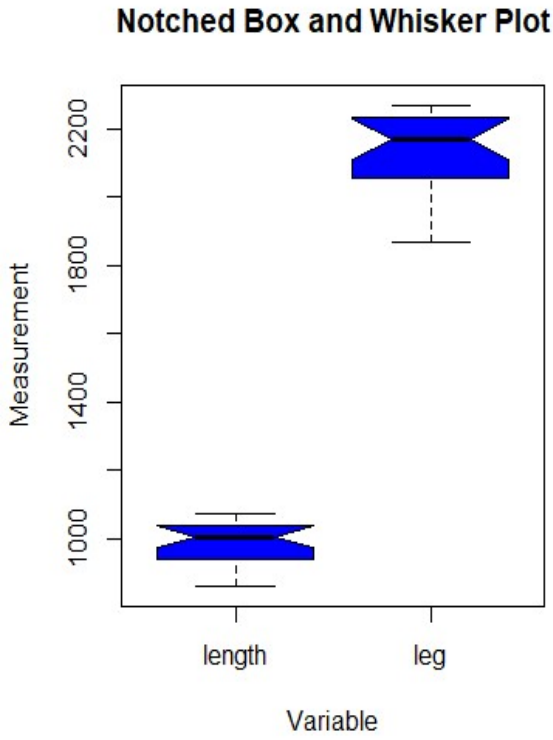
 # Interpretation:

1. **Box** represents the Interquartile range(IQR) of the given variable Leg data, which is the range between the 1st quartile (Q1) around 2060 (a tenth of a millimeter), and the 3rd quartile(Q3) around 2230 (a tenth of a millimeter). The above graph shows that the wider the box means data of variable Leg is more spread out.
2. **Notches** are overlapped means there may not be a statistically significant difference in the medians.
3. **Median Values:** The horizontal lines within the box represent the median of the dataset. For the "Leg" variable, the median is approximately 2150 (a tenth of a millimeter) on the y-axis.
4. **Whiskers**: the above graph shows that Whiskers are below the value of 1900 (a tenth of a millimeter) and above 2250 (a tenth of a millimeter) and beyond this range are outliers.

 # Create Notched Box and Whisker plot for variable name leg and leng

```
boxplot(data$Leng,data$Leg,notch = TRUE,names = c("length","leg"),
    col = 'blue',main="Notched Box and Whisker Plot",xlab="Variable",ylab="Measurement")
```

#Output:



CW1_ANS_2.c_ii

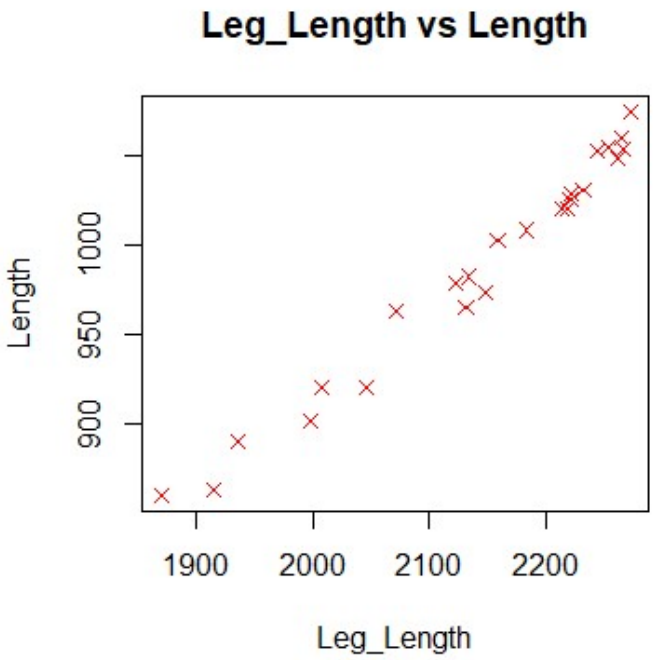
 # Interpretation of notched box and whisker plot for variable names length and leg as follows:

5. **Notches** of two boxes are overlapped; it shows that there is no strong evidence of a significant difference in medians between the groups. This means that, based on these plots, we cannot conclude that one variable significantly outperforms the other in terms of medians.
6. **Median Values:** The horizontal lines within the boxes represent the medians of the datasets. For the "Length" variable, the median is approximately 1000 (a tenth of a millimeter) on the y-axis, while for the "Leg" variable, the median is around 2150 (a tenth of a millimeter) on the y-axis. This tells us about the central tendencies of the two variables.
7. **Interquartile Range (IQR):** The height of the box, which represents the IQR, allows us to assess the spread of data within each variable. In this context, a larger IQR for the "Length" variable suggests that it has more variability compared to the "Leg" variable. This means that the range of measurements for "Length" is broader, indicating greater variability in the data compared to "Leg."

CW1_ANS_2.d

```
# Create scatterplot - leg length vs length

plot(data$Leg,data$Leng,pch=4,xlab = "Leg_Length",ylab = "Length",

     col='red',main='Leg_Length vs Length')
```

#Output:



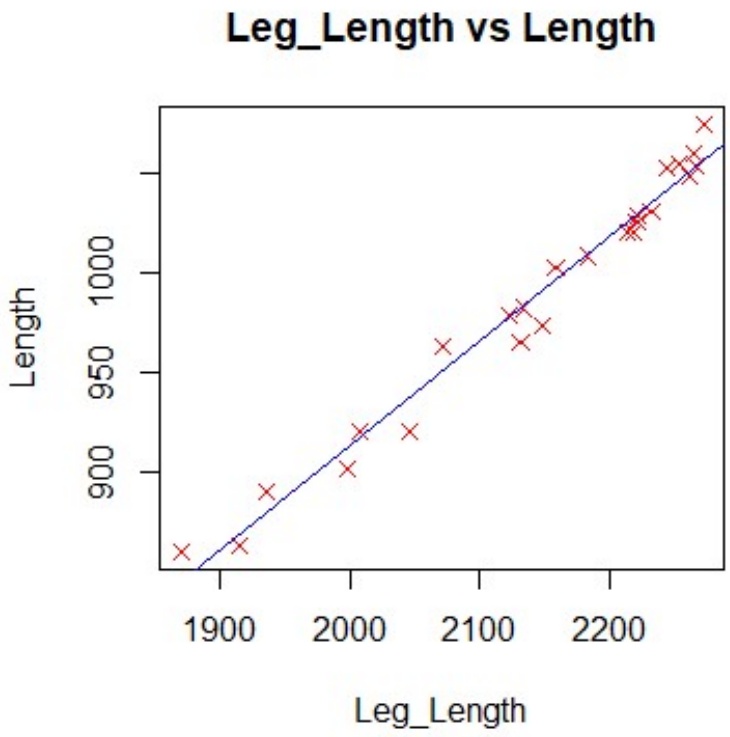
 # Interpretation:

This scatter plot helps visualize and understand the relationship between 'Leg Length' and 'Length' measurements. Also highlighting positive trends with some variability in the data.

#CW1_ANS_2.e

```
#Simple linear regression model for leng and leg variables.

length <-data$Leng

leg_length <-data$Leg

model <-lm(length ~ leg_length)

#2nd step:

#create a scatterplot for a simple linear regression model.

plot(data$Leg,data$Leng,pch=4,xlab = "Leg_Length",ylab = "Length",

   col='red',main='Leg_Length vs Length')

#3rd step:

# Draw a simple linear regression line.

abline(model,col='blue')
```

 # Output:



 # Interpretation:

The scatter plot with the least square regression line illustrates the positive linear relationship between “leg” and “leng” and allows for predictive insights, but also emphasizes the inherent data variability.

CW1_ANS_2.f

 # 1st step:

Predict value from Simple linear regression model for leng and leg variables.

model_fit_value <-predict(model)

model_fit_value

 # Output:

```
> model_fit_value
       1        2        3        4        5        6        7        8        9       10       11
844.5725 868.3214 878.8765 911.5972 916.8747 936.9294 950.6510 981.7884 982.8439 990.7602 977.0386
      12       13       14       15       16       17       18       19       20       21       22
996.0378 1009.2316 1025.0642 1027.7030 1028.7585 1034.5638 1029.2863 1040.8968 1050.3964 1051.9797 1052.5074
      23       24
1045.6466 1055.6739
```

 # 2nd step:

find the observed weight is at least 18mm larger than the predicted weight.

indices_values <-which((data$Leng-model_fit_value)>=18)

weight_of_wallaby <-data$Weight[indices_values]

Print out the weight of the wallaby

weight_of_wallaby

 # Output:

```
> indices_values <-which((data$Leng-model_fit_value)>=18)
> weight_of_wallaby <-data$weight[indices_values]
> weight_of_wallaby
[1] 62500
```

CW1_ANS_2.g

 # 1st step:

Fit linear regression model for leng (variable) and leg(variable)

r means residuals

r_model <-lm(data$Leng ~ data$Leg)

 # 2nd step:

Calculate residuals

r <- resid(r_model)

r_table <-data.frame(residuals=r)

r_table

 # Output:

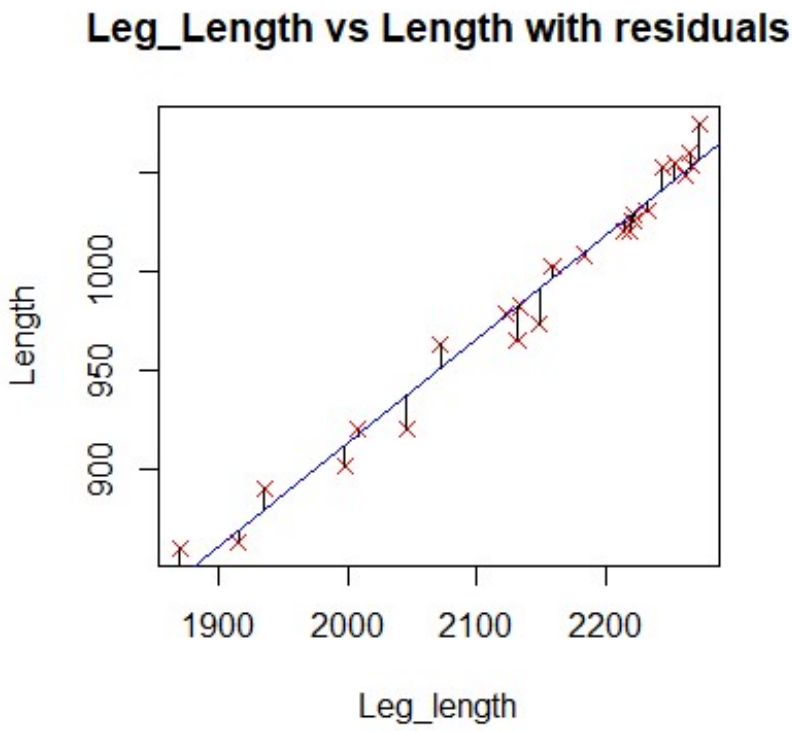
```
     residuals
1   15.4275251
2   -5.3213906
3   11.1235357
4  -10.5971926
5    3.1252706
6  -16.9293694
7   12.3490349
8  -16.7884324
9   -0.8439397
10 -17.7602450
11   0.9613508
12   5.9622182
13  -1.2316239
14  -5.0642344
15  -7.7030028
16  -3.7585101
17  -4.5638006
18  -1.2862638
19  11.1031552
20  -2.3964111
21   7.0203278
22   0.4925741
23   8.3533720
24  18.3260520
>
```

 # 3rd step:

Scatter plot for leg_length vs length with residual

```
plot(data$Leg,data$Leng,pch=4,xlab = 'Leg_length',ylab = 'Length',col='red',main='Leg_Length vs Length with residuals')

abline(r_model,col='blue')

segments(data$Leg,data$Leng,data$Leg,data$Leng-r,pch=25,col='black')
```

 # Output:



CW1_ANS_2.h

```
# Find out a correlation between leng and leg by formula.

X_bar <-mean(data$Leng)

X_bar

Y_bar <-mean(data$Leg)

Y_bar

Leng_stddev <-sd(data$Leng)

Leng_stddev

Leg_stddev <-sd(data$Leg)

Leg_stddev

Denominator <- (Leng_stddev*Leg_stddev)

Denominator

Covariance <- cov(data$Leng,data$Leg)
```

Covariance

Covariance_coefficient <-(Covariance/Denominator)

Covariance_coefficient

 # Output:

```
> X_bar <-mean(data$Leng)
> X_bar
[1] 987
> Y_bar <-mean(data$Leg)
> Y_bar
[1] 2140.875
> Leng_stddev <-sd(data$Leng)
> Leng_stddev
[1] 64.60987
> Leg_stddev <-sd(data$Leg)
> Leg_stddev
[1] 120.9536
> Denominator <- (Leng_stddev*Leg_stddev)
> Denominator
[1] 7814.794
> Covariance <- cov(data$Leng,data$Leg)
> Covariance
[1] 7720.913
> Covariance_coefficient <-(Covariance/Denominator)
> Covariance_coefficient
[1] 0.9879868
>
```

 # Conclusion:

From the above, all calculations for covariance coefficient, the value of covariance coefficient is 0.9878, which is exactly one, which shows a strong positive correlation between Leng (Variable) and Leg (variable).

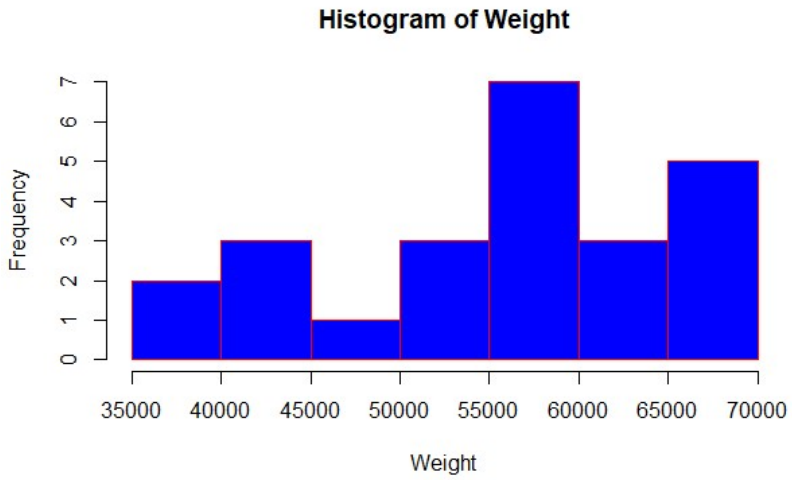
CW1_ANS_2.i

Create a Histogram of weight with 7bin.

hist(data$Weight,breaks = 7,col='blue',border = 'red',

main='Histogram of Weight', xlab = 'Weight')

 # Output:



 # Interpretation:

1. The histogram of weight shows the distribution of weight in the given dataset wallaby-nr-47.csv.
2. Weight data is divided into 7 bins as seen at the X-axis while the height of each bar represents the Frequency of weights falling into that bin as seen at the Y-axis.
3. From the above Histogram, we observe the highest wallabies weigh around 60000 (a tenth of a gram), and second highest wallabies weigh around 70000 (a tenth of a gram), and the least wallabies weigh around 50000 (a tenth of a gram).
4. Observed from the Histogram distribution, it is a right-skewed with a long tail on the right side.

CW1_ANS_2.j

 # 1st step:

```
# Create a vector to store the weight and age when weight loss is observed

weight_age_loss <-c()
```

 # 2nd step:

```
# Make a starting variable to store the previous weight and age

# p means previous

p_weight <- data$Weight[1]

p_age <- data$age[1]
```

 # 3rd step:

```
# Iterate through the data

# c means Current

for (i in 2:length(data$Weight)) {

  c_weight <-data$Weight[i]

  c_age <- data$Age[i]


  # Check if the current weight is less than the previous weight

  if(c_weight<p_weight) {

    weight_age_loss[[length(weight_age_loss)+1]] <-c(c_weight,c_age)

  }

  # Update the previous weight and age

  p_weight <- c_weight

  p_age <-c_age

}
```

 # 4th step:

```
# Print the vector containing weight and age when weight loss is observed
```

```
for (i in 1:length(weight_age_loss)){
  cat("At the point",i,"weight:",weight_age_loss[[i]][1],
      "age:",weight_age_loss[[i]][2],"\n")
}
```

 # Output:

```
At the point 1 weight: 53500 age: 723
At the point 2 weight: 58500 age: 885
At the point 3 weight: 67000 age: 1059
At the point 4 weight: 65000 age: 1088
At the point 5 weight: 62500 age: 1286
>
```

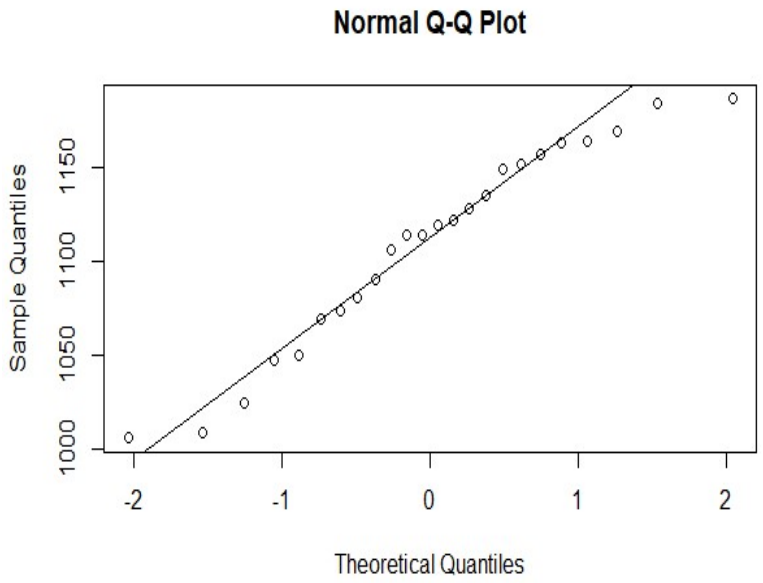
CW1_ANS_2.k

#Q-Q plots:

We can perform a Q-Q Plot to ensure that our assumption about the Variable Head is normally distributed.

```
qqnorm(data$Head)
qqline(data$Head)
```

#Output:



 # Interpretation:

From the above Q-Q Plot, We observed that data points closely follow the diagonal line, further indicating that the variable Head is normally distributed.

CW1_ANS_2.l_i

 # 1st step:

```
# Calculate the mean and standard deviation from the variable Head.

head_mean <- mean(data$Head)

head_sd <-sd(data$Head)

Lower_value <- 1090

Upper_value <- 1164
```

 # 2nd step:

```
# Calculate Z_score for Lower_value and Upper_value

z_lower <-(Lower_value - head_mean) / head_sd

z_upper <-(Upper_value - head_mean) / head_sd
```

 # 3rd step:

```
# Calculate cumulative probabilities and probability

c_p_lower <-pnorm(z_lower)

c_p_upper <-pnorm(z_upper)

probalility_between_values <- c_p_upper - c_p_lower

cat("Probability that Head is between 1090mm and 1164mm:", probalility_between_values, "\n")
```

 # Output:

The probability that the head is between 1090mm and 1164mm is 0.48282

CW1_ANS_2.l_ii

 # 1st step:

```
# Calculate the Z score for the 30th percentile (40% inside the interval) and 70th percentile (60%
outside the interval) of a standard normal distribution

mean_Head <- mean(data$Head)

sd_Head <-sd(data$Head)

p_30th <- 0.30

p_70th <- 0.70
```

 # 2nd step:

```
# Calculate z score for 30th and 70th percentile

zs_30th <- qnorm(p_30th)

zs_70th <- qnorm(p_70th)
```

 # 3rd step:

```
# Calculate values for the Head variable
```

V_30th <- mean_Head + zs_30th * sd_Head

V_70th <- mean_Head + zs_70th * sd_Head

 # 4th step:

Print the values at 30th and 70th percentile.

cat("Value for 40% inside:", V_30th, "mm\n")

cat("Value for 60% outside:", V_70th, "mm\n")

 # Output:

```
> #Print the values at 30th and 70th percentile.
> cat("Value for 40% inside:", V_30th, "mm\n")
Value for 40% inside: 1080.512 mm
> cat("Value for 60% outside:", V_70th, "mm\n")
Value for 60% outside: 1137.155 mm
>
```

CW1_ANS_3

 # 1st step:

Using Bayes' theorem,

$$P(R|F) = \frac{P(F|R).P(R)}{P(F)}$$

Where,

R means the Event that the randomly chosen car is red

P(R|F) means the probability that a randomly chosen faulty car is red.

P(R) means the probability of R being true before considering the new evidence.

P(F) means marginal probability.

P(F|R) means the conditional probabilities of a car being faulty given its color.

 # Using Bayes' theorem for randomly chosen faulty cars is green.

$$P(G|F) = \frac{P(F|G).P(G)}{P(F)}$$

Where,

G means the Event that the randomly chosen car is green.

P(G|F) means the probability that a randomly chosen faulty car is green.

P(G) means the probability of G being true before considering the new evidence.

P(F) means marginal probability.

P(F|G) means the conditional probabilities of a car being faulty given its color.

 # Given Details as below:

prob_redcar <- **0.6** # red car percentage -- **P(R)**

prob_greencar <- **0.4** # green car percentage -- **P(G)**

prob_faulty_redcar <- **0.015** # faulty red car percentage -- **P(F|R)**

prob_faulty_greencar <- **0.035** # falulty green car percentage -- **P(F|G)**

 # Probability of total Redcar (included faulty Redcars) -- P(F|R)*P(R)

prob_redcar_total <- prob_redcar*prob_faulty_redcar # **(0.015*0.6)**

prob_redcar_total # **(0.009)**

 # Output:

```
> prob_redcar_total <- prob_redcar*prob_faulty_redcar
> prob_redcar_total
[1] 0.009
```

 # Probalility of total greencar (included faulty greencars) -- P(F|G)*P(G)

prob_greencar_total <- prob_greencar*prob_faulty_greencar **# (0.035*0.4)**

prob_greencar_total # **(0.014)**

 # Output:

```
> prob_greencar_total <- prob_greencar*prob_faulty_greencar
> prob_greencar_total
[1] 0.014
```

 # Calculate using the law of probability for faulty cars -- P(F) = P(F|R)*P(R) + P(F|G)*P(G)

prob_total_faultycars <-prob_redcar_total + prob_greencar_total **# (0.009 +0.014)**

prob_total_faultycars # **(0.023)**

 # Output:

```
> prob_total_faultycars <-prob_redcar_total + prob_greencar_total
> prob_total_faultycars
[1] 0.023
```

 # 2nd step:

 # Calculate probabilities for given red car which is faulty -- P(R|F) = (P(F|R)*P(R))/P(F)

prob_redcar_faulty <- (prob_faulty_redcar*prob_redcar)/prob_total_faultycars # **(0.009/0.023)**

cat("Probability that the given red car

which is faulty (P(redcar|faulty)):", prob_redcar_faulty, "\n") # **0.3913**

 # Output:

```
> prob_redcar_faulty <- (prob_faulty_redcar*prob_redcar)/prob_total_faultycars
> cat("Probability that the given red car which is faulty (P(redcar|faulty)):", prob_redcar_faulty, "\n")
Probability that the given red car which is faulty (P(redcar|faulty)): 0.3913043
>
```

 # Calculate probabilities for given green car which is faulty -- P(G|F) = (P(F|G)*P(G))/P(F)

prob_greencar_faulty <-(prob_faulty_greencar*prob_greencar)/prob_total_faultycars # **(0.014/0.023)**

cat("Probability that the given green car

which is faulty (P(green|faulty)):",prob_greencar_faulty, "\n") # **0.6086**

 # Output:

```
> prob_greencar_faulty <-(prob_faulty_greencar*prob_greencar)/prob_total_faultycars
> cat("Probability that the given green car which is faulty (P(green|faulty)):",prob_greencar_faulty, "\n")
Probability that the given green car which is faulty (P(green|faulty)): 0.6086957
> |
```

 # Conclusion:

The probability of a **red car** being faulty is **0.3913 and the** Probability of a **green car** being faulty is **0.6086.**

From the above probabilities, both the red faulty car and the green faulty car picked up by John randomly are **green in color**.

CW1_ANS_4_i

Use Poisson distribution for calculating probability because flow occurs randomly and we have details to find out the average rate of flow per meter.

$$P(X - k) = (e^{-\lambda} * \lambda^{k})/k!$$

Where,

X is the random variable representing the number of flows.

k is the number of flows. (value of k is given which is 3).

λ is the average rate of flaws per unit length.

Now put values in the equation,

$P(X - 3) = (e^{-0.05} * (0.05)\text{^}3)/3!$

= ((0.9512)*(0.000125))/6

= 0.0000198167

= 1.98167 * 10^{-5}

Calculate the average rate of flow per meter by using R commands:

lambda <- 0.05 # We have details like one flow per 20 meters. so flow per meter is 1/20 means 0.05.

Number of flaws we have. (exactly 3 flow in 5 meter means k=3)

k<-3

probability <- dpois(k,lambda)

cat("Probability of exactly 3 flaws in a 5-meter cloth:", probability, "\n")

 # Output:

```
> probability <- dpois(k,lambda)
> cat("Probability of exactly 3 flaws in a 5-meter cloth:", probability, "\n")
Probability of exactly 3 flaws in a 5-meter cloth: 1.981728e-05
>
```

CW1_ANS_4_ii

Use Poisson Distribution for calculating probability because flow occurs randomly and we have details to find out the average rate of flow per meter.

$$P(X-k) = (e^{-\lambda} * \lambda^{k})/k!$$

Where,

X is the random variable representing the number of flows.

k is the number of flows. (value of k is given which is 0).

λ is the average rate of flaws per unit length.

Now put values in the equation,

$P(X-0) = (e^{-0.05} * (0.05)\text{^}0)/0!$

= ((0.9512)*(1))/1

= 0.9512

Calculate the average rate of flow per meter by using the R command

lambda <- 0.05 # We have details like one flow per 20 meters. so flow per meter is 1/20 means 0.05.

No. of events we need to calculate ate probability (no flow in 10 meters means k=0)

k<-0

probability <- dpois(k,lambda)

cat("Probability of no flaws in a 10-meter cloth:", probability, "\n")

 # Output:

```
> probability <- dpois(k,lambda)
> cat("Probability of no flaws in a 10-meter cloth:", probability, "\n")
Probability of no flaws in a 10-meter cloth: 0.9512294
>
```

 # REFERENCES:

1. https://learn.uea.ac.uk/ultra/courses/_144066_1/cl/outline (LECTURE NOTES AND LAB EXERCISES- PROF. KATHARINA HUBAR)
2. Francis, A. (1979) *Advanced level statistics: an integrated course*. Cheltenham: Stanley Thornes.
3. Crawley, M. J. (2015) *Statistics : an introduction using r*. Second edn. Chichester, West Sussex: Wiley.