|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | MODELLING GENE EXPRESSION USING NONLINEAR REGRESSION. | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | UCHECHUKWU FREDRICK OKONKWO STID: 11738697 |  | | | |
|  | | | | 26TH OCTOBER 2021—7089CEM INTRODUCTION TO STATISTICAL METHODS FOR DATA SCIENCE COURSE WORK—DR FEI HE |  | | | |
|  | | |  | | |  | | |

**INTRODUCTION**

|  |
| --- |
| The aim of this analysis is to select the best regression model (from a candidate set of nonlinear regression models) that can well describe the relationship between several ‘simulated’ gene expression time-series data. Gene expression is one of the most important biological processes where information from a gene is used to synthesize a functional gene product, such as protein. The expression of a gene can be controlled (or regulated) by another gene or several other genes, through a gene product (protein) called transcription factor. Understanding how genes regulate each other, i.e. gene regulation, is important to investigate a complex diseases, and how cell respond to environmental stimuli. |

The data set gene\_data.csv shows the simulated 5 gene expression time series. The first column contains the sampling time in minutes while the next 5 columns represents the time course expression data of 5 genes x1, x2, x3, x4 and x5 respectively. This 6 columns are without headers and each have 301(rows) entries in total. I assumed that the 5 genes are independent and identically distributed also Gaussian with mean(µ) equal to zero with unknown variance(σ).

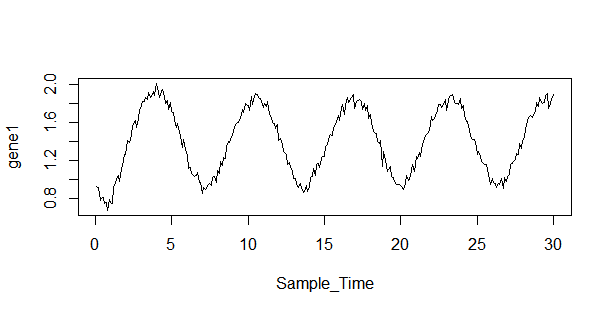
I used R-studio throughout my analysis in computing and generating plots.

**TASK 1.0: PRELIMINARY DATA ANALYSIS**

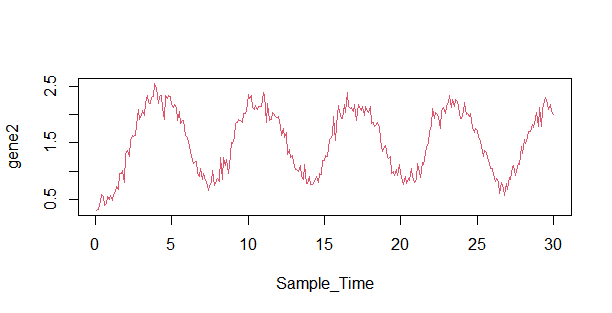
Firstly, I performed an initial exploratory analysis to investigate data sets and summarize their main characteristics. It also helps us discover patterns, spot anomalies, understand the data set variables as well as the relationship between this variables.

* 1. : Time series plots of each gene against the sampling time. This charts helps determine the trend over time, to know the nature of the data point as well as if it exhibits any pattern.

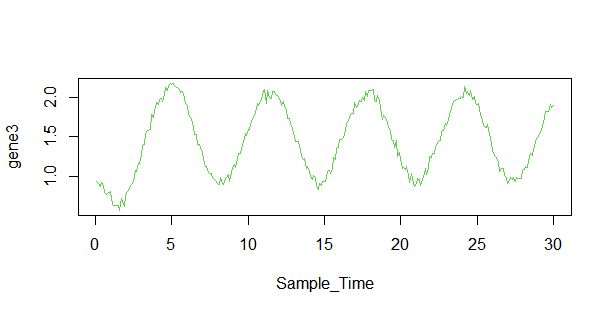
Plot 1.1.1: Gene1 against sampling time



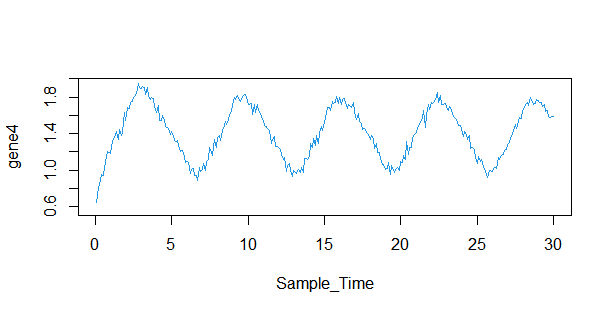
Plot 1.1.2: Gene2 against sampling time



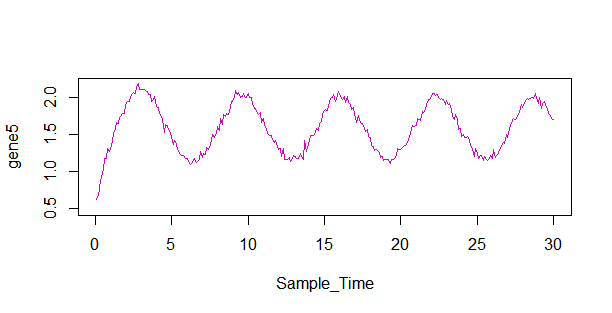
Plot 1.1.3: Gene3 against sampling time



Plot 1.1.4: Gene4 against sampling time



Plot 1.1.5: Gene5 against sampling time

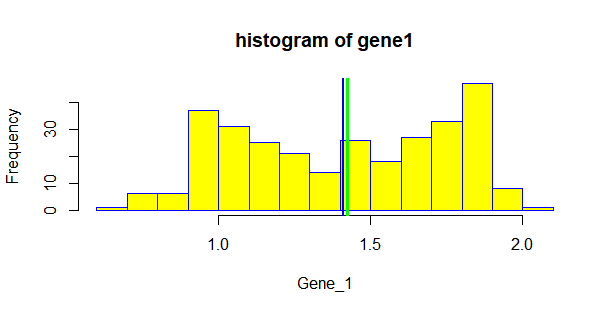


We notice a familiar trend and pattern on the 5 respective time series plots showing the movement of this gene expression data point along the sampling time in minutes. They seem to have a wavelike pattern which show a consistent fluctuation as the sample time increases. This tells us that the 5 gene expression data all have similar data structure and trend pattern. With this information obtained we proceed to determining the distribution of each gene expression. See code 1.1 in appendix.

* 1. Distribution of each gene shown using a histogram and the data summary table.

(mean = blue line; median=green line)

Plot 1.2.1

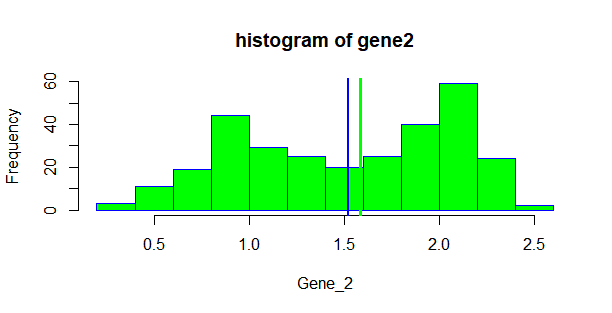


**Gene 1 data summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MIN | 1ST QU | MEDIAN | MEAN | 3RD QU | MAX |
| 0.6813 | 1.0760 | 1.4245 | 1.4085 | 1.7560 | 2.0118 |

From the histogram plot for gene1 above, we see that this data distribution is bi-modal or double-peaked which usually indicates the data has 2 groups. The Gene 1 data summary table shows a summary about the data like the mean (1.4085), median (1.4245), max (2.0118) and min (0.6813) respectively. See code 1.2.1 in appendix.

Plot 1.2.2

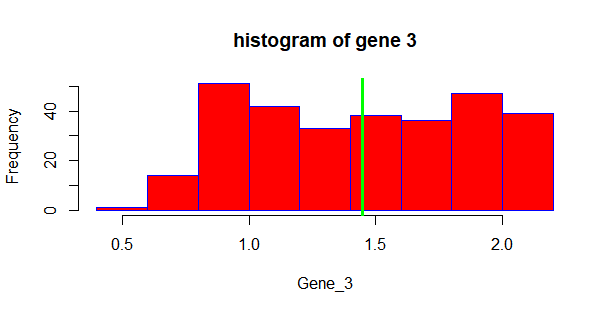


**Gene 2 data summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MIN | 1ST QU | MEDIAN | MEAN | 3RD QU | MAX |
| 0.3143 | 0.9949 | 1.5865 | 1.5237 | 2.0417 | 2.5402 |

The histogram for gene 2 also shows a bi-modal or double peaked data distribution. Which implies that the gene 2 data expression has 2 different groups or the gene expression was taken in two different batches. In the Gene 2 data summary table we see this data has a mean of 1.5237 and median of 1.5865. It also has the lowest minimum value of 0.3143 which is relatively small and the highest maximum value of 2.5402 in comparison with other gene data. See code 1.2.2 in appendix

Plot 1.2.3

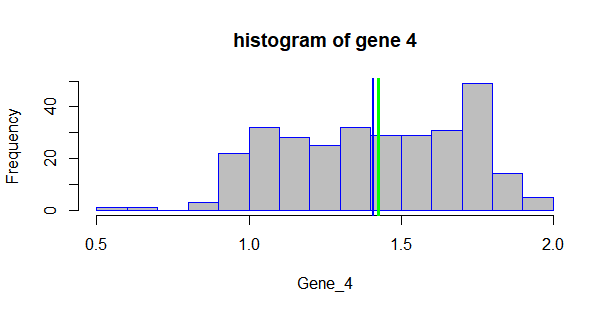


**Gene 3 data summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MIN | 1ST QU | MEDIAN | MEAN | 3RD QU | MAX |
| 0.569 | 1.037 | 1.446 | 1.447 | 1.880 | 2.174 |

The histogram of gene 3 also show a kind of multi-modal distribution. The data values on the left have the lowest frequency making it partially left skewed. This data also has a mean that is equal to its median which is 1.446. See code 1.2.3 in appendix.

Plot 1.2.4

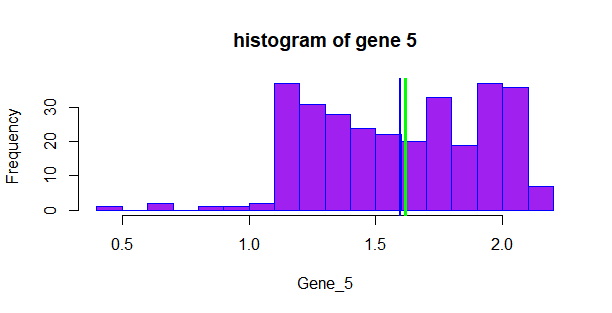


**Gene 4 data summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MIN | 1ST QU | MEDIAN | MEAN | 3RD QU | MAX |
| 0.5648 | 1.1564 | 1.4247 | 1.4101 | 1.6836 | 1.9468 |

The histogram plot 1.2.4 above shows that the gene 4 data distribution is left skewed or also called negatively skewed because there is a tail on the negative side of the number line with the peak on the positive side of the number line. Its maximum value is 1.9468. See code 1.2.4 in appendix.

Plot 1.2.5



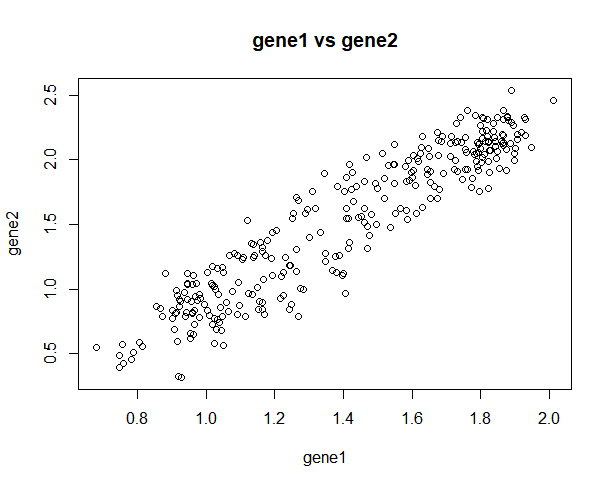
**Gene 5 data summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MIN | 1ST QU | MEDIAN | MEAN | 3RD QU | MAX |
| 0.4822 | 1.3015 | 1.6171 | 1.5956 | 1.9125 | 2.1905 |

Gene 5 data distribution is quite different compared to other gene distributions. From the plot 1.2.5 we see that the data is left skewed with the low gene expression values in the left side of the histogram. Additionally, the histogram is multi-modal, that is, we have multiple high values with high frequency. See code 1.2.5 in appendix.

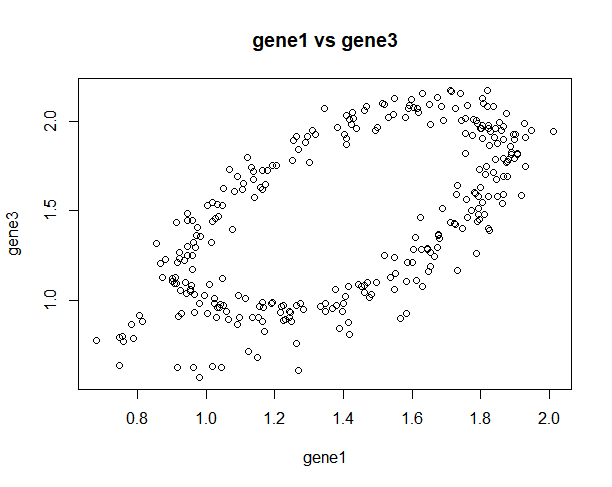
* 1. Scatter plot showing the relationship between various gene expression combination and the correlation co-efficient for this gene combinations.

Plot 1.3.1



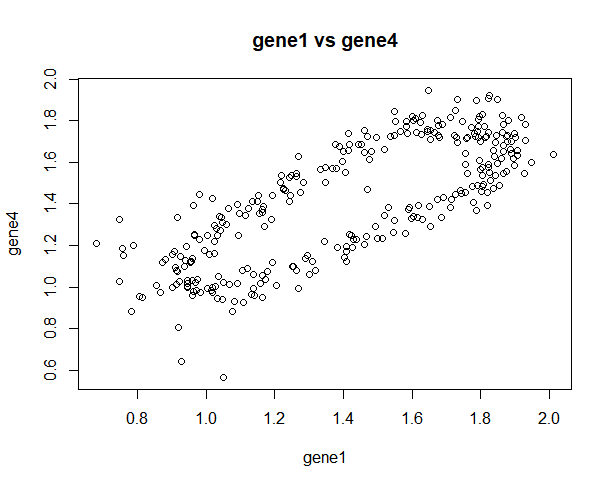
From the scatter plot we see a strong positive relationship between gene 1 and gene 2. Its correlation co-efficient is 0.9352346 which also indicates that there is strength in the linear relationship between gene 1 and gene 2. See code 1.3.1 in appendix

Plot 1.3.2



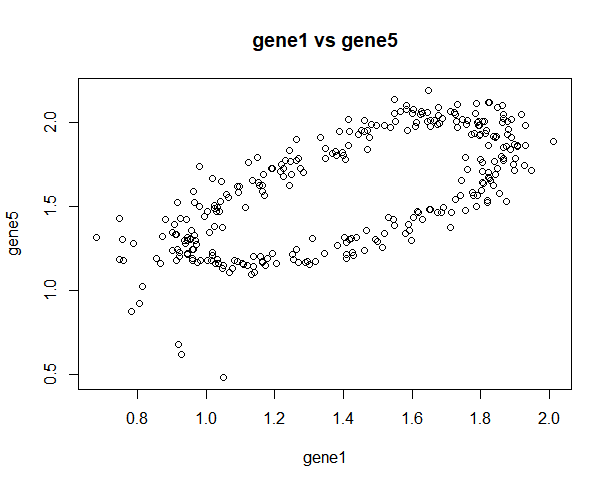
The correlation coefficient for gene 1 and gene 3 scatter plot is 0.601361 which indicates a moderately positive relationship, that is, the value of gene 3 expression on the vertical axis increases as the value of the gene 1 expression increases also. See code 1.3.2 in appendix.

Plot 1.3.3



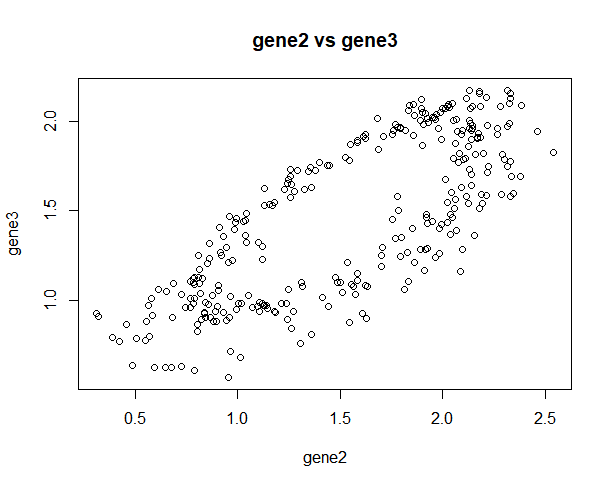
From this plot 1.3.3 we observe that there exist also a positive relationship between gene 1 and gene 4 with a positive correlation coefficient of 0.7743476. See code 1.3.3 in appendix.

Plot 1.3.4



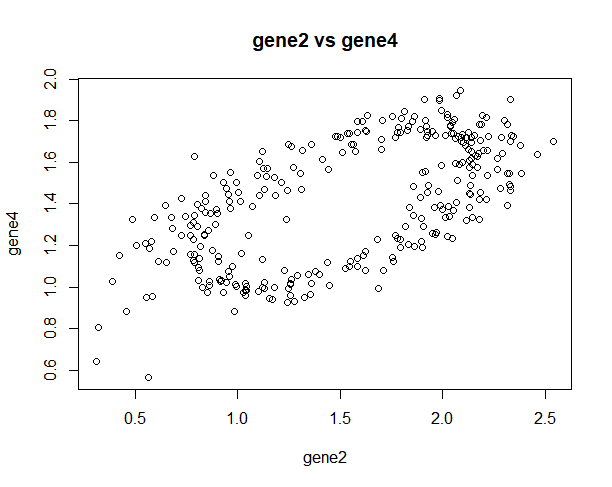
Between gene 1 and gene 5, there exist a positive relationship as we see from the plot 1.3.4 with a correlation coefficient of 0.6727781. See code 1.3.4 in appendix.

Plot 1.3.5



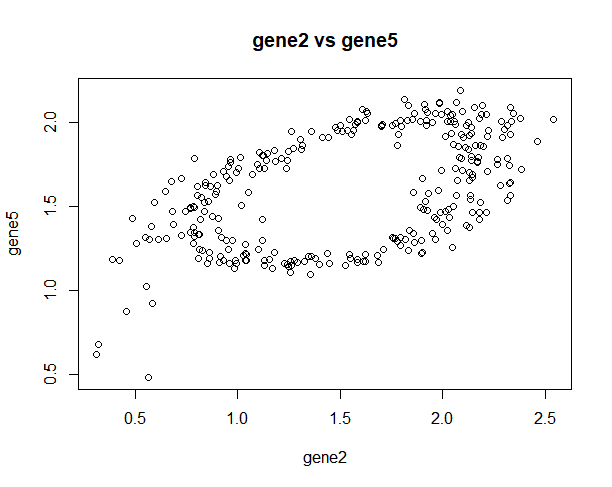
There also exist a positive relationship between gene 2 and gene 3 with correlation coefficient of 0.7632734 as we see from the scatter plot in plot 1.3.5. See code 1.3.5 in appendix.

Plot 1.3.6



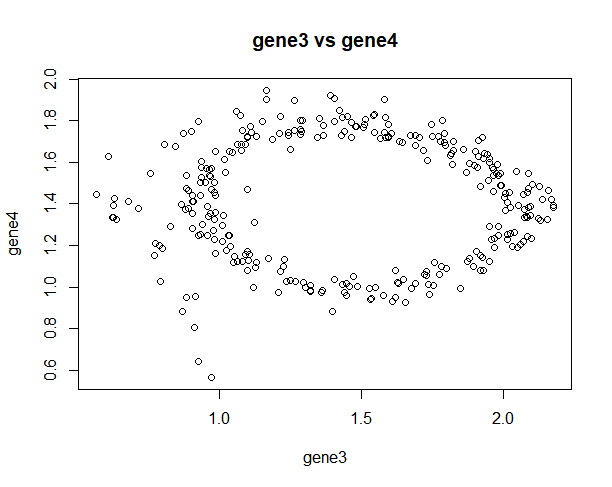
The correlation coefficient between gene 2 and gene 4 is computed as 0.6183979 and this conforms to the scatter plot shown above which indicates also that there is a positive relationship between the gene 2 and gene 4. See code 1.3.6 in appendix

Plot 1.3.7



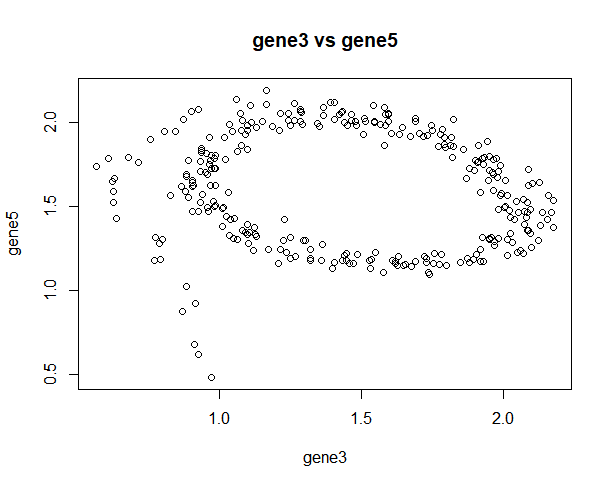
With a correlation coefficient of 0.5240552 and the scatter plot above we also see a weak positive relationship between gene 2 and gene 5. See code 1.3.7 in appendix.

Plot 1.3.8



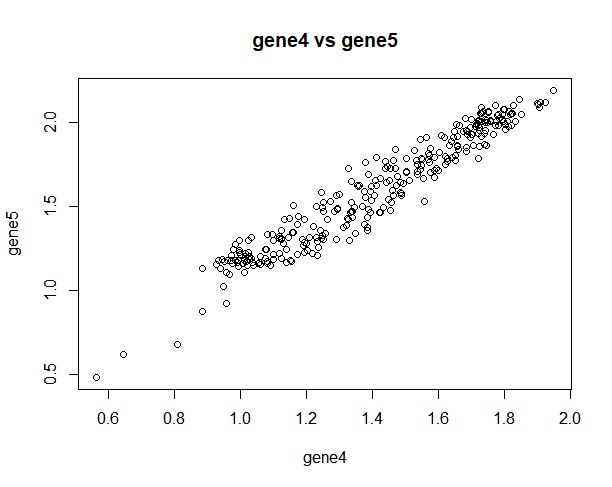
The correlation coefficient for variables in gene3 and gene 4 is 0.02139137 which indicates that there doesn’t exist a relationship between these gene expressions as we can see from the scatter plot in plot 1.3.8. See code 1.3.8 in appendix.

Plot 1.3.9



From the scatter plot above in plot 1.3.9 we see that there exist a negative relationship between the gene 3 and gene 5. The negative value of the correlation coefficient which is -0.1152138 agrees to that representation. See code 1.3.9 in appendix.

Plot 1.4.0



The scatter plot above shows a very strong positive linear relationship between gene 4 and gene 5 with the highest and strongest correlation coefficient value of 0.96672 which is approximately equal to 1 amongst other gene combination. See code 1.4.0 in appendix.

**TASK 2.0 REGRESSION- MODELING THE RELATIONSHIP BETWEEN GENE EXPRESSIONS.**

In order to determine a suitable mathematical model in explaining the relationship between the output gene 𝐲 = 𝐱2 with other input genes (i.e. x1, 𝐱3, 𝐱4, 𝐱5 ) that actually ‘regulate’ its expression, which we assume can be described by a polynomial regression model. We would try to fit the following 5 models into our data to conclude on which data best fits this data.

Candidate models are with the following structures:

Model 1: 𝑦 = 𝜃1X4 + 𝜃2 X32 + 𝜃𝑏𝑖𝑎𝑠

Model 2: 𝑦 = 𝜃1 X4 + 𝜃2 X32 + 𝜃3 X5 + 𝜃𝑏𝑖𝑎𝑠

Model 3: 𝑦 = 𝜃1 X3 + 𝜃2 𝑥4 + 𝜃3 X53 + 𝜃𝑏𝑖𝑎𝑠

Model 4: 𝑦 = 𝜃1 X4 + 𝜃2 X32 + 𝜃3 X53 + 𝜃𝑏𝑖𝑎𝑠

Model 5: 𝑦 = 𝜃1 X4 + 𝜃2 X12 + 𝜃3 X32 + 𝜃𝑏𝑖𝑎𝑠

**2.0.1** Fitting the regression models into the data set.

Model 1: 𝑦 = 𝜃1X4 + 𝜃2 X32 + 𝜃𝑏𝑖𝑎𝑠

After running the code 2.0.1 in appendix, the parameter estimates are intercept = -1, 𝜃1 = 1.16788 and 𝜃2 = 0.9452 respectively with R2 value of 0.9452 or 95% which indicates that the other gene expressions in the model explains 95% of the variation around the mean in gene 2 data which is in this case the response variable.

**2.0.2**

Model 2: 𝑦 = 𝜃1 X4 + 𝜃2 X32 + 𝜃3 X5 + 𝜃𝑏𝑖𝑎𝑠

After fitting the model 2 to the data the parameter estimates are intercept = -1.68353, 𝜃1 = 0.03892, 𝜃2 = 1.06537 and 𝜃3 = 1.00930 with R2 value of 0.9622 or 96%. See code 2.0.2 in appendix.

**2.0.3**

Model 3: 𝑦 = 𝜃1 X3 + 𝜃2 𝑥4 + 𝜃3 X53 + 𝜃𝑏𝑖𝑎𝑠

The value for the 4 parameter estimates are -1.68353,1.06537, 0.03892 and 1.00930 with R2 value to be 0.9622 or 96%. See code 2.0.3 in appendix.

**2.0.4**

Model 4: 𝑦 = 𝜃1 X4 + 𝜃2 X32 + 𝜃3 X53 + 𝜃𝑏𝑖𝑎𝑠

After running the code 2.0.4 in appendix to fit this 4th model to the data set, the parameter estimates are somewhat similar to the last two models since they involve gene expressions x3, x4, x5 respectively. The parameter estimates are as follows: -1.68353, 0.03892, 1.06537, 1.00930 with similar R2 value of 0.9622. But this is not enough information to determine how fitting this models are to the dataset.

**2.0.5**

Model 5: 𝑦 = 𝜃1 X4 + 𝜃2 X12 + 𝜃3 X32 + 𝜃𝑏𝑖𝑎𝑠

For this model 5 the parameter estimates are intercept = -1.24048, 0.72266, 0.48372, 0.73503 with R2 value equal to 0.9505 or 95%.

**2.1 Estimate model parameters 𝜽 = {𝜃1, 𝜃2, ⋯ , 𝜃𝑏𝑖𝑎𝑠 }𝑇 for every candidate model using Least Squares (𝜽 ̂= (𝐗𝑇 𝐗)−1𝐗𝑇 𝐲), using the provided input and output gene datasets (using all the data for training).**

**Table 2.1 LEAST SQUARES PARAMETER ESTIMATE VALUES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model/Parameters | 𝜃𝑏𝑖𝑎𝑠 | 𝜃1 | 𝜃2 | 𝜃3 |
| **Model 1** | **-0.8907** | **1.1838** | **0.3258** | **---** |
| **Model 2** | **-1.0257** | **-0.2241** | **0.3735** | **1.2604** |
| **Model 3** | **-0.6506** | **1.0969** | **-0.0560** | **0.1450** |
| **Model 4** | **0.4084** | **-0.5093** | **0.3983** | **0.2010** |
| **Model 5** | **-0.4191** | **0.7036** | **0.1877** | **0.2427** |

In table 2.1 above we see the least squares parameter estimates for the 5 respective models. See code 2.1.1 to 2.1.5 in appendix.

**TASK 2.2 – 2.4**

* Based on the estimated model parameters, I computed the **model residual (error) sum of squared errors (RSS)**, for every candidate model using the formula:
* I computed the log-likelihood function for every candidate model using the formula:

ln 𝑝(𝐷|𝜽 ̂) = − 𝑛/2 ln(2𝜋) – 𝑛/2 ln(𝜎̂2) – 1/2𝜎̂2 RSS

* I also computed the Akaike information criterion (AIC) and Bayesian information criterion (BIC) for each candidate model.

See codes for task 2.2 – 2.4 in appendix.

**TABLE 2.2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RSS | LOG-LIKELIHOOD FUNCTION | AIC | BIC |
| **Model 1** | **6.545379** | **298.9941** | **-591.9882** | **-580.8668** |
| **Model 2** | **4.163573** | **367.1206** | **-726.2413** | **-711.4129** |
| **Model 3** | **2.145991** | **466.8891** | **-925.7782** | **-910.9497** |
| **Model 4** | **1.191621** | **555.4316** | **-1102.863** | **-1088.035** |
| **Model 5** | **5.764454** | **318.131** | **-628.2619** | **-613.4335** |

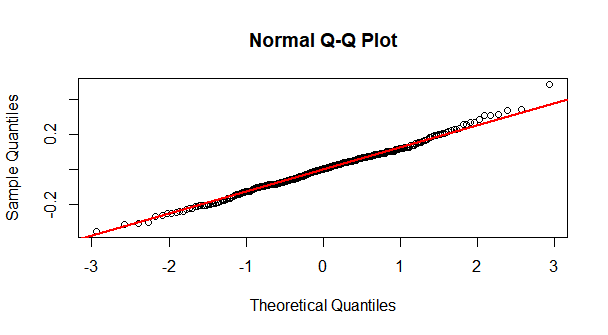
The table 2.2 above shows the computed result of the residual sum of squares, log-likelihood function, the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) for the 5 models respectively.

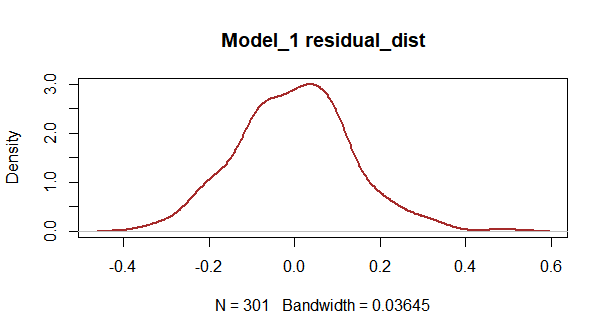
**Task 2.5:**

The distribution of model prediction errors (residuals) for each candidate model. Plot the error distributions, and evaluate if those distributions are close to Normal/Gaussian (as the output gene has additive Gaussian noise), e.g. by using Q-Q plot.

**2.5.1 MODEL 1 RESIDUAL ERROR DISTRIBUTION AND QQ PLOT**

**Plot 2.5.1**

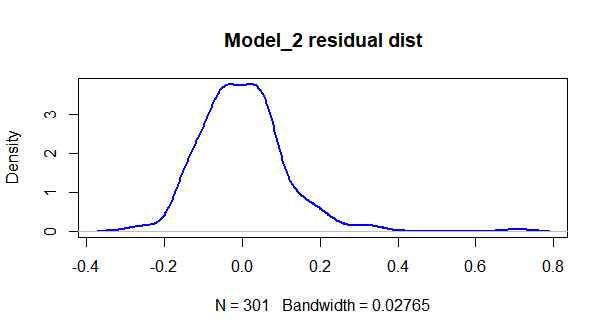
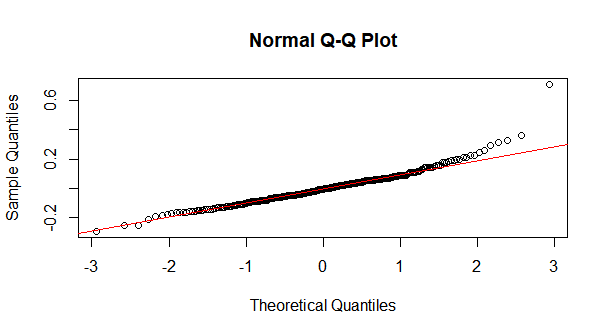




In the plots 2.5.1 above we see the match of the observed distribution with theoretical distribution. In the model 1 qq-plot we see a slight deviation at the upper tail which signifies it is partially right skewed or positively skewed but remains normally distributed as we can confirm that from the model 1 residual distribution plot. See code 2.5.1 in appendix.

**2.5.2 MODEL 2 RESIDUAL ERROR DISTRIBUTION AND QQ PLOT**

**Plot 2.5.2**

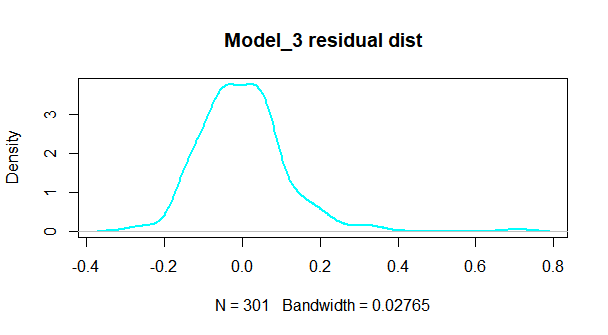
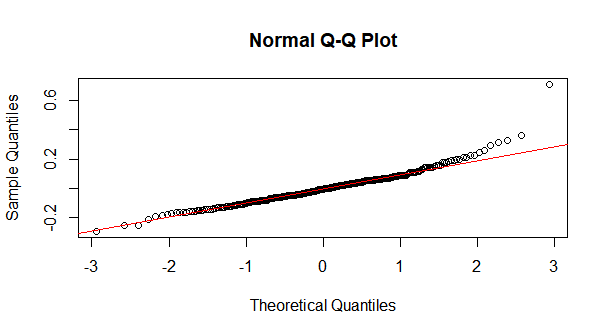


We also see in the two plots above (Plot 2.5.1) that there is a sharp deviation in the upper tail of the normal qq plot and the residual plot also confirms that it is also positive skewed with its peak density at 0.0 which also checks for the normal distribution condition.

See code 2.5.1 in appendix.

**2.5.3 MODEL 3 RESIDUAL ERROR DISTRIBUTION AND QQ PLOT**

**Plot 2.5.3**



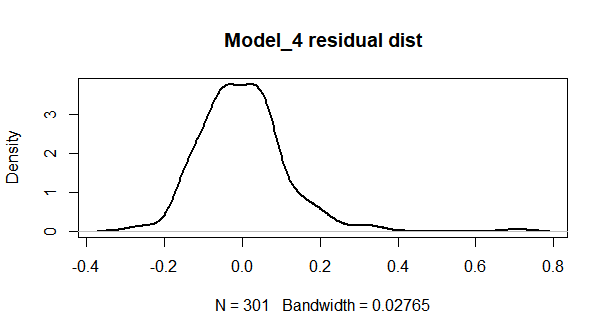
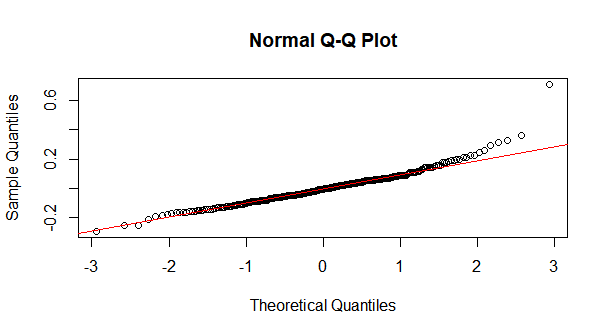
This qq normal plot and model residual as seen in the plot 2.5.3 is similar to that of the model 2.

It implies that its also right skewed.

See code 2.5.3 in appendix.

**2.5.4 MODEL 4 RESIDUAL ERROR DISTRIBUTION AND QQ PLOT**

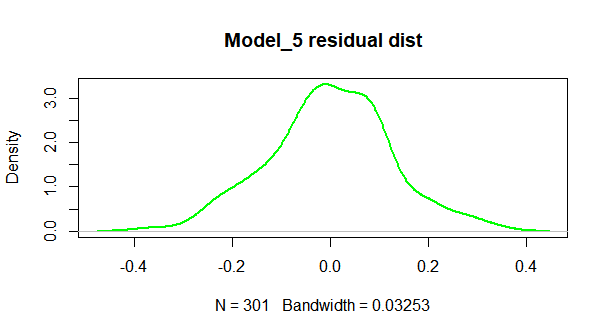
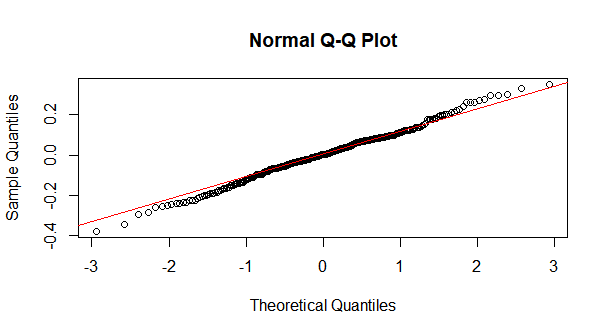
**Plot 2.5.4**



From the plot 2.5.4 above showing the residual plots and distribution we see that it is also right skewed.

**2.5.5 MODEL 5 RESIDUAL ERROR DISTRIBUTION AND QQ PLOT**

**Plot 2.5.5**



From the plot 2.5.5 above we see from the qq normal plot that there exist a deviation in the lower and upper tail from the qq line . The residual distribution shows a perfect normally distributed bell.

**Task 2.6:** Selecting the best model for this gene expression data set according to the Akaike information criterion, Bayesian information criterion and the distribution of the model residuals among the five polynomial regression models.

Akaike’s information criterion (AIC) will be used to select the best model by comparing the quality or goodness of fit among the five regression models with different number of parameters. While the Bayesian information criterion (BIC) is based on the likelihood function and it takes into account the number of observation unlike AIC.

**TABLE 2.2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RSS | LOG-LIKELIHOOD FUNCTION | AIC | BIC |
| **Model 1** | **6.545379** | **298.9941** | **-591.9882** | **-580.8668** |
| **Model 2** | **4.163573** | **367.1206** | **-726.2413** | **-711.4129** |
| **Model 3** | **2.145991** | **466.8891** | **-925.7782** | **-910.9497** |
| **Model 4** | **1.191621** | **555.4316** | **-1102.863** | **-1088.035** |
| **Model 5** | **5.764454** | **318.131** | **-628.2619** | **-613.4335** |

From the table 2.2 we see that model 4 has the significantly lowest AIC value of **-1102.863** and BIC value of **-1088.035** with the highest log-likelihood function value of **555.4316** and the lowest residual sum of squares errors equal to **1.191621** when compared to other models. Based on this result I would conclude that model 4 appears to be the model that best fits this gene expression data set.

In Plot 2.5.4 the distribution of the model residual is normally distributed but right skewed as a result of the deviation on the upper tail.

**2.7 TRAINING AND TESTING THE MODEL 4.**

Split the input and output gene dataset (𝐗 and 𝐲) into two parts: one part used to train the model, the other used for testing (e.g. 70% for training, 30% for testing). For the selected ‘best’ model, 1) estimate model parameters use the training dataset; 2) compute the model’s output/prediction on the testing data; and 3) also compute the 95% (model prediction) confidence intervals and plot them (with error bars) together with the model prediction, as well as the testing data samples.

I splitted the data into two groups,210 for training and 91 for testing. See code 2.7.1 in appendix

Fitting the model to the testing data set we were able to get the following parameters estimates:

Y = 0.086X4 + 1.098X2 + 0.967X5 – 1.73

**2.7.2 MODEL PREDICTION**

**Gene 2 expression prediction summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MIN | 1ST QU | MEDIAN | MEAN | 3RD QU | MAX |
| -0.1275 | 1.0162 | 1.5203 | 1.4989 | 2.0319 | 2.3521 |

**S**ee code 2.7.2 in appendix.

I also computed the 95% confidence interval for the product prediction.

|  |
| --- |
| fit lwr upr |
| Min. :-0.1479 Min. :-0.2129 Min. :-0.08291 |
| 1st Qu.: 0.9879 1st Qu.: 0.9628 1st Qu.: 1.01529 |
| Median : 1.4702 Median : 1.4428 Median : 1.49755 |
| Mean : 1.4806 Mean : 1.4503 Mean : 1.51092 |
| 3rd Qu.: 2.0065 3rd Qu.: 1.9711 3rd Qu.: 2.03859 |
| Max. : 2.3714 Max. : 2.3366 Max. : 2.40897 |

**APPENDIX**

R codes

setwd("C:\\Users\\Uche\\Desktop\\COVENTRY MSC\\intro\_to\_stats7089CEM")

data = read.csv("gene\_data.csv", header = F)

data = as.matrix(data)

**TASK 1.1 Time series plot**

**1.1.1**

plot(data$V1, data$V2, type = "l",

xlab = "Sample\_Time",

ylab = "gene1")

**1.1.2**

plot(data$V1, data$V3, type = "l", col = 2,

xlab = "Sample\_Time",

ylab = "gene2")

**1.1.3**

plot(data$V1, data$V4, type = "l", col = 3,

xlab = "Sample\_Time",

ylab = "gene3")

**1.1.4**

plot(data$V1, data$V5, type = "l", col = 4,

xlab = "Sample\_Time",

ylab = "gene4")

**1.1.5**

plot(data$V1, data$V6, type = "l", col = 6,

xlab = "Sample\_Time",

ylab = "gene5")

**TASK 1.2 Histogram and gene expression data summary R-code**

**1.2.1**

#gene1 histogram chart

Gene\_1<-data$V2

hist(Gene\_1,breaks = 10,col = "yellow",border = "blue", main = "histogram of gene1")

abline(v = mean(data$V2), lwd = 2 , col="blue")

abline(v = median(data$V2), lwd = 3 , col = "green")

summary(data$V2)

**1.2.2**

#gene2 histogram chart

Gene\_2<- data$V3

hist(Gene\_2,breaks = 10,col = "green" ,border = "blue", main = "histogram of gene2")

abline(v = mean(data$V3), lwd = 2 , col="blue")

abline(v = median(data$V3), lwd = 3 , col = "green")

summary(data$V3)

**1.2.3**

#gene3 histogram chart

Gene\_3<- data$V4

hist(Gene\_3,breaks = 10,col = "red",border = "blue", main = "histogram of gene 3")

abline(v = mean(data$V4), lwd = 2 , col="blue")

abline(v = median(data$V4), lwd = 3 , col = "green")

summary(data$V4)

**1.2.4**

#gene4 histogram chart

Gene\_4<-data$V5

hist(Gene\_4,breaks = 15,col = "grey",border = "blue", main = "histogram of gene 4")

abline(v = mean(data$V5), lwd = 2 , col="blue")

abline(v = median(data$V5), lwd = 3 , col = "green")

summary(data$V5)

**1.2.5**

#gene5 histogram chart

Gene\_5<-data$V6

hist(Gene\_5,breaks = 15,col = "purple",border = "blue", main = "histogram of gene 5")

abline(v = mean(data$V6), lwd = 2 , col="blue")

abline(v = median(data$V6), lwd = 3 , col = "green")

summary(data$V6)

**TASK 1.3 Scatter plot and correlation coefficient R-code to establish the relationship between gene data.**

Gene\_1<-data$V2

Gene\_2<-data$V3

Gene\_3<-data$V4

Gene\_4<-data$V5

Gene\_5<-data$V6

**1.3.1**

plot(Gene\_1,Gene\_2,

xlab = "gene1",

ylab = "gene2",

main = "gene1 vs gene2")

cor.test(Gene\_1,Gene\_2,method = "pearson")

**1.3.2**

plot(Gene\_1,Gene\_3,

xlab = "gene1",

ylab = "gene3",

main = "gene1 vs gene3")

cor.test(Gene\_1,Gene\_3, method = "pearson")

**1.3.3**

plot(Gene\_1,Gene\_4,

xlab = "gene1",

ylab = "gene4",

main = "gene1 vs gene4")

cor.test(Gene\_1,Gene\_4,method = "pearson")

**1.3.4**

plot(Gene\_1,Gene\_5,

xlab = "gene1",

ylab = "gene5",

main = "gene1 vs gene5")

cor.test(Gene\_1,Gene\_5, method = "pearson")

**1.3.5**

plot(Gene\_2,Gene\_3,

xlab = "gene2",

ylab = "gene3",

main = "gene2 vs gene3")

cor.test(Gene\_2,Gene\_3, method = "pearson")

**1.3.6**

plot(Gene\_2,Gene\_4,

xlab = "gene2",

ylab = "gene4",

main = "gene2 vs gene4")

cor.test(Gene\_2,Gene\_4,method = "pearson")

**1.3.7**

plot(Gene\_2,Gene\_5,

xlab = "gene2",

ylab = "gene5",

main = "gene2 vs gene5")

cor.test(Gene\_2,Gene\_5,method = "pearson")

**1.3.8**

plot(Gene\_3,Gene\_4,

xlab = "gene3",

ylab = "gene4",

main = "gene3 vs gene4")

cor.test(Gene\_3,Gene\_4, method = "pearson")

**1.3.9**

plot(Gene\_3,Gene\_5,

xlab = "gene3",

ylab = "gene5",

main = "gene3 vs gene5")

cor.test(Gene\_3,Gene\_5, method = "pearson")

**1.4.0**

plot(Gene\_4,Gene\_5,

xlab = "gene4",

ylab = "gene5",

main = "gene4 vs gene5")

cor.test(Gene\_4,Gene\_5,method = "pearson")

**TASK 2.0 FITTING THE REGRESSION MODELS TO THE DATA SET**

**2.0.1**

#Model 1:

y=data$V3 #gene2

x4=data$V5 #gene4

x3=data$V4 #gene3

model\_1<-lm(y~x4 + (x3^2), data)

summary(model\_1)

**2.0.2**

#Model 2:

y = data$V3 #gene2

x4 = data$V5 #gene4

x3 = data$V4 #gene3

x5 = data$V6 #gene5

model\_2<-lm(y~x4+(x3^2)+x5,data)

summary(model\_2)

**2.0.3**

#Model 3:

y = data$V3 #gene2

x3 = data$V4 #gene3

x4 = data$V5 #gene4

x5 = data$V6 #gene5

model\_3<-lm(y~x3+x4+(x5^3),data)

summary(model\_3)

**2.0.4**

#Model 4:

y = data$V3

x4 = data$V5

x3 = data$V4

x5 = data$V6

model\_4<-lm(y~x4+(x3^2)+(x5^3),data)

summary(model\_4)

**2.0.5**

#Model 5:

y = data$V3

x4 = data$V5

x1 = data$V2

x3 = data$V4

model\_5<-lm(y~x4+(x1^2)+(x3^2))

summary(model\_5)

**#Task 2.1: Estimate model parameters using Least Squares**

**2.1.1**

#model\_1 estimate

y = matrix(data$V3,nrow = 301,ncol = 1)

print(y)

x4 = matrix(data$V5,nrow = 301,ncol = 1)

x3 = matrix(data$V4,nrow = 301,ncol = 1)

ones = matrix(1 , length(y),1)

X<- cbind(ones, x4, (x3^2))

print(X)

thetaHat = solve(t(X) %\*% X) %\*% t(X) %\*% y

print(thetaHat)

**2.1.2**

#Model\_2 parameter estimator

y = matrix(data$V3,nrow = 301,ncol = 1)

print(y)

x4 = matrix(data$V5,nrow = 301,ncol = 1)

x3 = matrix(data$V4,nrow = 301,ncol = 1)

x5 = matrix(data$V6,nrow = 301,ncol = 1)

ones = matrix(1 , length(y),1)

X<- cbind(ones, x4, (x3^2) , x5)

print(X)

thetaHat = solve(t(X) %\*% X) %\*% t(X) %\*% y

print(thetaHat)

**2.1.3**

#Model 3:

y = matrix(data$V3,nrow = 301,ncol = 1)

print(y)

x3 = matrix(data$V4,nrow = 301,ncol = 1)

x4 = matrix(data$V5,nrow = 301,ncol = 1)

x5 = matrix(data$V6,nrow = 301,ncol = 1)

ones = matrix(1 , length(y),1)

X<- cbind(ones, x3, x4 , (x5^3))

print(X)

thetaHat = solve(t(X) %\*% X) %\*% t(X) %\*% y

print(thetaHat)

**2.1.4**

#Model 4 :

y = matrix(data$V3,nrow = 301,ncol = 1)

print(y)

x4 = matrix(data$V5,nrow = 301,ncol = 1)

x3 = matrix(data$V4,nrow = 301,ncol = 1)

x5 = matrix(data$V6,nrow = 301,ncol = 1)

ones = matrix(1 , length(y),1)

X<- cbind(ones, x4, (x3^2) , (x5^3))

print(X)

thetaHat = solve(t(X) %\*% X) %\*% t(X) %\*% y

print(thetaHat)

**2.1.5**

#Model 5:

y = matrix(data$V3,nrow = 301,ncol = 1)

print(y)

x4 = matrix(data$V5,nrow = 301,ncol = 1)

x1 = matrix(data$V2,nrow = 301,ncol = 1)

x3 = matrix(data$V4,nrow = 301,ncol = 1)

ones = matrix(1 , length(y),1)

X<- cbind(ones, x4, (x1^2) , (x3^2))

print(X)

thetaHat = solve(t(X) %\*% X) %\*% t(X) %\*% y

print(thetaHat)

**TASK 2.2 – 2.4**

**# RSS, LOG-LIKELIHOOD FUNCTION, AIC, BIC**

**#MODEL 1**

y\_Hat = X %\*% thetaHat

error = y - y\_Hat

#RSSE = norm(error, type = "2")^2

RSSE = sum((y-y\_Hat)^2)

print(RSSE)

#logliklihoood function

n<-nrow(y)

sigmahat2<- RSSE/(n-1)

loglik\_1<- -n/2\*(log(2\*pi))-n/2\*(log(sigmahat2))-1/2\*(sigmahat2\*RSSE)

print(loglik\_1)

#Akaike information criterion (AIC)

K = 3

AIC\_1 <- 2\*K - (2\*loglik\_1)

print(AIC\_1)

#Bayesian information criterion (BIC)

k = 3

n<-nrow(y)

BIC\_1<- k\*log(n) - (2\*loglik\_1)

print(BIC\_1)

**#MODEL 2**

y\_Hat = X %\*% thetaHat

error = y - y\_Hat

#RSSE = norm(error, type = "2")^2

RSSE = sum((y-y\_Hat)^2)

print(RSSE)

#logliklihoood function

n<-nrow(y)

sigmahat2<- RSSE/(n-1)

loglik\_2<- -n/2\*(log(2\*pi))-n/2\*(log(sigmahat2))-1/2\*(sigmahat2\*RSSE)

print(loglik\_2)

#Akaike information criterion (AIC)

K = 4

AIC\_2 <- 2\*K - (2\*loglik\_2)

print(AIC\_2)

#Bayesian information criterion (BIC)

k = 4

n<-nrow(y)

BIC\_2<- k\*log(n) - (2\*loglik\_2)

print(BIC\_2)

**#MODEL 3**

y\_Hat = X %\*% thetaHat

error = y - y\_Hat

#RSSE = norm(error, type = "2")^2

RSSE = sum((y-y\_Hat)^2)

print(RSSE)

#logliklihoood function

n<-nrow(y)

sigmahat2<- RSSE/(n-1)

loglik\_3<- -n/2\*(log(2\*pi))-n/2\*(log(sigmahat2))-1/2\*(sigmahat2\*RSSE)

print(loglik\_3)

#Akaike information criterion (AIC)

K = 4

AIC\_3 <- 2\*K - (2\*loglik\_3)

print(AIC\_3)

#Bayesian information criterion (BIC)

k = 4

n<-nrow(y)

BIC\_3<- k\*log(n) - (2\*loglik\_3)

print(BIC\_3)

**#MODEL 4**

y\_Hat = X %\*% thetaHat

error = y - y\_Hat

#RSSE = norm(error, type = "2")^2

RSSE = sum((y-y\_Hat)^2)

print(RSSE)

#logliklihoood function

n<-nrow(y)

sigmahat2<- RSSE/(n-1)

loglik\_4<- -n/2\*(log(2\*pi))-n/2\*(log(sigmahat2))-1/2\*(sigmahat2\*RSSE)

print(loglik\_4)

#Akaike information criterion (AIC)

K = 4

AIC\_4 <- 2\*K - (2\*loglik\_4)

print(AIC\_4)

#Bayesian information criterion (BIC)

k = 4

n<-nrow(y)

BIC\_4<- k\*log(n) - (2\*loglik\_4)

print(BIC\_4)

**#MODEL 5**

y\_Hat = X %\*% thetaHat

error = y - y\_Hat

#RSSE = norm(error, type = "2")^2

RSSE = sum((y-y\_Hat)^2)

print(RSSE)

#logliklihoood function

n<-nrow(y)

sigmahat2<- RSSE/(n-1)

loglik\_5<- -n/2\*(log(2\*pi))-n/2\*(log(sigmahat2))-1/2\*(sigmahat2\*RSSE)

print(loglik\_5)

#Akaike information criterion (AIC)

K = 4

AIC\_5 <- 2\*K - (2\*loglik\_5)

print(AIC\_5)

#Bayesian information criterion (BIC)

k = 4

n<-nrow(y)

BIC\_5<- k\*log(n) - (2\*loglik\_5)

print(BIC\_5)

**Task 2.5: R-Code**

**2.5.1 Model 1:**

y=data$V3 #gene2

x4=data$V5 #gene4

x3=data$V4 #gene3

model\_1<-lm(y~x4 + (x3^2), data)

summary(model\_1)

#distribution of model prediction errors Q-Q plot (residuals){TASK 2.5}

res<-resid(model\_1)

qqnorm(res)

qqline(res,lwd = 2,col = "red")

plot(density(res),col = "brown",lwd = 2,main = "Model\_1 residual\_dist")

**2.5.2 Model 2:**

y = data$V3 #gene2

x4 = data$V5 #gene4

x3 = data$V4 #gene3

x5 = data$V6 #gene5

model\_2<-lm(y~x4+(x3^2)+x5,data)

summary(model\_2)

#distribution of model prediction errors Q-Q plot (residuals){TASK 2.5}

res<-resid(model\_2)

qqnorm(res)

qqline(res, col = "red")

plot(density(res),col = "blue",lwd = 2,main = "Model\_2 residual dist")

**2.5.3 Model 3:**

y = data$V3 #gene2

x3 = data$V4 #gene3

x4 = data$V5 #gene4

x5 = data$V6 #gene5

model\_3<-lm(y~x3+x4+(x5^3),data)

summary(model\_3)

#distribution of model prediction errors Q-Q plot (residuals){TASK 2.5}

res<-resid(model\_3)

qqnorm(res)

qqline(res,col = "red")

plot(density(res),col = "cyan",lwd = 2, main = "Model\_3 residual dist")

**2.5.4 Model 4:**

y = data$V3

x4 = data$V5

x3 = data$V4

x5 = data$V6

model\_4<-lm(y~x4+(x3^2)+(x5^3),data)

summary(model\_4)

#distribution of model prediction errors Q-Q plot (residuals){TASK 2.5}

res<-resid(model\_4)

qqnorm(res)

qqline(res,col = "red")

plot(density(res),lwd = 2,main = 'Model\_4 residual dist')

**2.5.5 Model 5:**

y = data$V3

x4 = data$V5

x1 = data$V2

x3 = data$V4

model\_5<-lm(y~x4+(x1^2)+(x3^2))

summary(model\_5)

#distribution of model prediction errors Q-Q plot (residuals)

res<-resid(model\_5)

qqnorm(res)

qqline(res,col = "red")

plot(density(res),col = "green",lwd = 2,main = "Model\_5 residual dist")

**2.7 SPLITTING DATA INTO TRAINING AND TESTING**

**2.7.1**

install.packages("caTools")

library(caTools)

setwd("C:\\Users\\Uche\\Desktop\\COVENTRY MSC\\intro\_to\_stats7089CEM")

data = read.csv("gene\_data.csv", header = F)

split = sample.split(Y = data$V3, SplitRatio = 0.7)

#subsetting into Train data

train = data[split,]

#subsetting into Test data

test = data[!split,]

dim(train)

dim(test)

**2.7.2**

y = train$V3

x4 = train$V5

x3 = train$V4

x5 = train$V6

model\_4<-lm(y~x4+(x3^2)+(x5^3),train)

summary(model\_4)

#compute the model’s output/prediction on the testing data

gene2\_predict<-predict(model\_4,test)

summary(gene2\_predict)

confidence\_predict<-predict(model\_4,test,interval = "confidence")

summary(Model\_prediction)