# AUTHOR: Brian Pankau

# DATE: 23 June 2017

# CLASS: UWLAX DS710 SUMMER 2017

# ASSIGNMENT: 6-1 R program

# EXCUTE:

#

*# Problem 6-1(a): Can we detect when a marketing campaign has been successful?*

*# On homework 4, you simulated data from the TableFarm salad chain before*

*# and after the implementation of a new marketing campaign.*

*# Read the combined data (both before and after) into R.*

*# (You could do this by saving the data as a .csv file and using*

*# read.csv(), or by copying the data into a text file, separating the values*

*# by commas, and enclosing the data in c( … ) to make a vector.)*

#

# read data file

setwd("c:/Temp1")

TableFarm = read.csv("TableFarm.csv")

attach(TableFarm)

TableFarm

# Month Current\_Revenue Proposed\_Revenue Combined\_Revenue

#1 1 94000 99000 5000

#2 2 91000 109000 18000

#3 3 75000 83000 8000

#4 4 101000 117000 16000

#5 5 105000 145000 40000

#6 6 76000 145000 69000

#7 7 108000 115000 7000

#8 8 105000 145000 40000

#9 9 97000 112000 15000

#10 10 114000 145000 31000

#11 11 110000 145000 35000

#12 12 115000 145000 30000

*# 6(b) Make a scatterplot of the data.*

*# Add a vertical line to mark the month in which the new marketing campaign began, and*

*# add a legend to your plot.*

plot(Month, Combined\_Revenue, main="Table Farm Combined Revenue")

abline(v=1, col="red", lwd=3)

legend(6,60000,c("New Campaign","Revenue"), lty=c(1,1), lwd=c(2.5,2.5), col=c("red","black"))

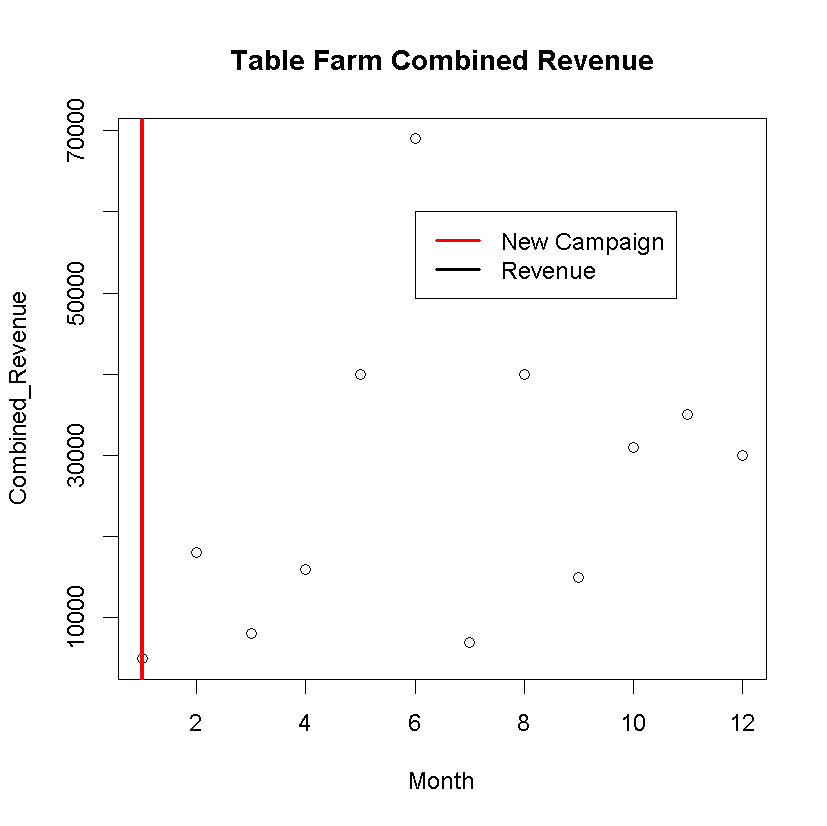


Figure 1

*# 6 (c) Make a single graph with 2 side-by-side boxplots of the revenue*

*# before and after implementing the marketing campaign.*

*# Write a few sentences describing and comparing the boxplots,*

*# and relating them to the underlying model you used to simulate the data.*

par(mfrow=c(1,2))

boxplot(Current\_Revenue ~ Month, data=TableFarm, xlab="Months", ylab="Revenue", main="Current Revenue", ylim=c(5000,150000))

boxplot(Proposed\_Revenue ~ Month, data=TableFarm, xlab="Months", ylab="Revenue", main="Proposed Revenue", ylim=c(5000,150000))

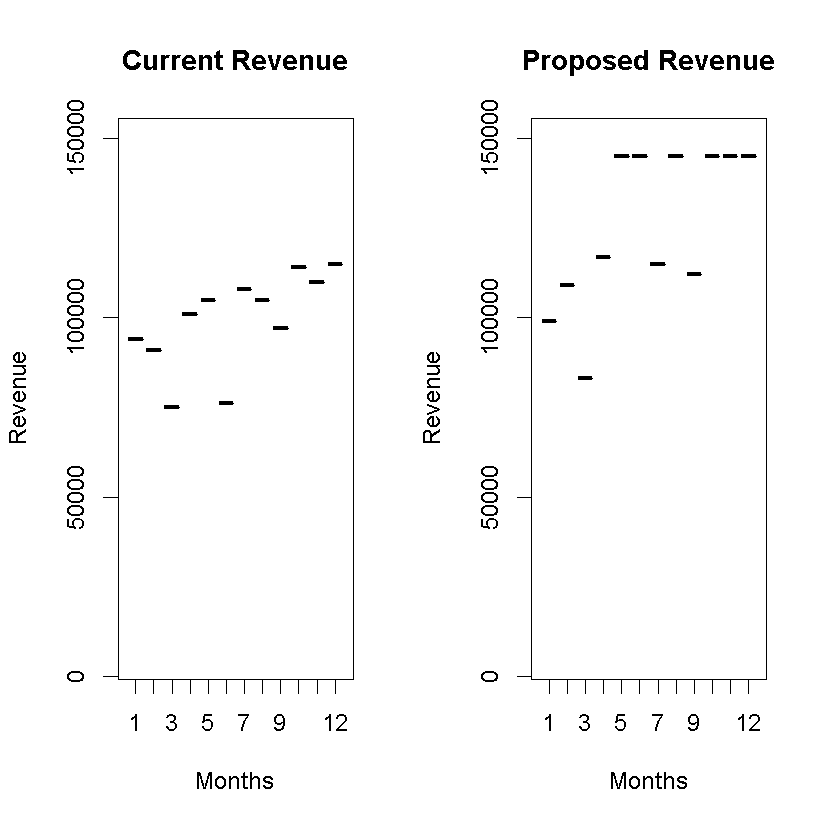


Figure 2

The proceeding chart depict the current and the proposed revenue before and after the adoption of a new advertising model for TableFarm salad chain. The Current Revenue boxplot data points are shown to be grouped within a constrained range around $100,000 which mirrors the mean annual sales ($100,000) and deviation in monthly sales from the mean ($12,000). The advertising firm predicts that its proposed advertising model will increase overall sales with a annual mean of $120,000 but also increasing the deviation in monthly sales from the mean by $25,000. This higher annual sales is depicted in the Proposed Revenue boxplot in that most of the projected monthly sales are above $100,000 and are closer to $150,000. Likewise, in comparing the Current Revenue boxplot to the Proposed Revenue boxplot, one can see that there is a smaller amount of deviation in monthly sales from the mean of the current revenue in the Current Revenue boxplot when compared to the deviation in monthly sales of the mean of the proposed revenue in the Proposed Revenue boxplot.

*# 6(d) Based on the way you simulated the data, you know that the marketing*

*# campaign was successful; that is, the data after implementing the marketing*

*# campaign was simulated from an underlying model with a higher mean than*

*# before the marketing campaign. However, in real life we probably wouldn’t*

*# know this. Based on the scatterplot and boxplots, would you be confident*

*# in claiming that the marketing campaign was successful? Why or why not?*

After comparing the two boxplots, I would assert that the proposed advertising model is superior. The justification for this claim is that an inspection of the Proposed Revenue boxplot indicates there seems to be more months above the mean of the current annual sales ($100,000) and fewer months below the current annual sales. Furthermore, although an inspection of the Proposed Revenue boxplot indicates that there is a larger variation in the deviation for the monthly sales from the proposed annual mean, it is noted that (in this iteration) there are more months that are above the proposed mean than are below the mean, visually indicating that the total revenue for the year will (most likely) be greater under the proposed revenue model when compared to the current revenue model.

*# 6(e) Write the null and alternative hypotheses for a test of whether the marketing*

*# campaign was successful. (I.e., whether the mean revenue with the marketing*

*# campaign is higher than the mean revenue before the marketing campaign.)*

The null hypothesis of the experiment assumes that there is no difference between the annual means of the two revenue models; while the alternate hypothesis is asserting that the mean of the annual sales for the proposed model statistically exceeds the mean of the current revenue model (i.e., the proposed revenue model has a positive effect).

Roughly:

H0:The proposed revenue model mean minus current revenue model mean equals 0.

H1: The proposed revenue model mean minus current revenue model mean equals greater or equal to 0.

Specifically:

H0: At the 95% confidence level, the mean sales of the proposed advertising model is no different than the mean sales of the current advertising model.

H1: At the 95% confidence level, the mean sales of the proposed advertising model is greater than the mean sales of the current advertising model.

*# 6(f) In a few sentences, explain why a 2-sample, 1-sided t-test is appropriate*

*# for testing the hypotheses in part e.*

There are two samples, Current Revenue and Proposed Revenue; thus, a two sampled test is appropriate in that the experimement is attempting to determin whether the means of the two revenue models are equalivent.

The type of test (a t-test) is appropriate for the dependent variable (revenue) is independent and unrelated between the two revenue models, is small in number, is numeric, is continuous, and is based on a normal distribution. Ideally, the variance of the two models would be similar, in this experiment the propose revenue model’s standard deviation is much higher than the current revenue’s model.

The experiment is testing whether or not the mean of the revenue from the proposed revenue model is statistically greater than the mean of the revenue from the current revenue model. In other words, does the proposed revenue model positively effect annual revenue for Table Farm? The key indicator, in this experimental design, is the type of comparsion (i.e., the word “greater”); indicating that a one-side t-test is appropriate.

If the experiment was attempting to determine if the two means were equal or if one model produced more or less revenue, then a two-side t-test would be appropriate.

Thus, in a one-sided test, the experiment would be testing the probability in a single direction (i.e., only testing if the mean of annual sales is better), while in a two-sided test the experiment would be evaluating the probability in both directions (i.e., testing multiple alternatives: the mean of annual revenue could be better, or it also could equivalent or worst).

However, the results of one-sided test is limited in that the results from the test will only be significant if the proposed revenue model outperforms the current revenue model. Stated in another way, the one-sided test will only indicate whether or not the proposed revenue model has a greater mean than the mean of the current revenue model.

*# 6(g) Conduct a 2-sample, 1-sided t-test in R. Include the R output and state*

*# your conclusion in the context of the problem.*

t.test(Proposed\_Revenue, Current\_Revenue, alternative="greater")

# Welch Two Sample t-test

#

#data: Proposed\_Revenue and Current\_Revenue

#t = 3.4988, df = 18.018, p-value = 0.00128

#alternative hypothesis: true difference in means is greater than 0

#95 percent confidence interval:

# 13198.6 Inf

#sample estimates:

#mean of x mean of y

# 125416.7 99250.0

Assessment of t-test results:

The revenue experiment is testing at the 95% confidence level which implies that the alpha value for the evaluation of the null hypothesis is 0.05: a value for which a p-value that is less that or equal to alpha is considered statistically significant for a given hypothesis. The test is attempting to determine the probability that the mean of the proposed revenue model is greater than the mean of the current revenue model (on average); thus, any p-value that results from the t-test that is less than or equal to alpha means that the null hypothesis should be rejected in favor of the alternative hypothesis. Since p-value = 0.00.128 which is lesss than alpha = 0.05, the null hypothsis is rejected in favor that the mean of the proposed revenue model is greater than the mean of the current revenue model (on average)**.**

*Problem 2 Can we detect an association between chocolate consumption and Nobel prizes?*

*2(a) On homework 4, you simulated data on countries’ per-capita chocolate consumption and number of Nobel Prize winners, using an error term ϵ (representing random “noise”). Read these data into R and make a scatterplot of the number of Nobel Prize winners versus chocolate consumption.*

# read data file

setwd("c:/Temp1")

ChocolateNobel = read.csv("Chocolate\_per\_Nobel.csv")

attach(ChocolateNobel)

ChocolateNobel

# Chocolate Nobel

#1 7.46 1.95

#2 13.04 4.39

#3 6.34 0.51

#4 11.74 4.61

#5 13.12 3.69

#6 4.96 1.46

#7 1.71 0.93

#8 13.81 3.33

#9 4.50 0.10

#10 13.56 3.85

#11 13.00 3.68

#12 4.62 1.20

#13 14.39 5.72

#14 2.26 0.00

#15 14.84 4.85

#16 3.99 0.00

#17 11.37 3.48

#18 1.35 0.00

#19 2.22 0.00

#20 13.05 5.51

#21 9.63 1.84

#22 13.14 4.74

#23 2.17 0.00

#24 10.52 2.45

#25 0.76 0.00

#26 8.24 1.49

#27 13.16 4.99

#28 9.21 0.77

#29 9.27 3.15

#30 8.02 2.09

#31 1.78 0.00

#32 4.87 1.03

#33 10.08 2.33

#34 3.27 1.32

#35 6.20 0.39

#36 3.45 0.00

#37 8.15 2.77

#38 13.18 3.37

#39 14.91 3.71

#40 13.31 4.67

#41 5.66 1.77

#42 9.79 2.49

#43 7.88 3.18

#44 13.34 4.28

#45 2.44 0.23

#46 11.71 2.01

#47 13.41 3.76

#48 11.11 5.63

#49 4.41 0.80

#50 11.97 6.10

*2(b) Fit a linear model to the data. What is the equation of the line of best fit? How does it compare to the theoretical model you used to simulate the data? Graph the line of best fit with the scatterplot.*

summary(ChocolateNobel)

# Chocolate Nobel

# Min. : 0.760 Min. :0.0000

# 1st Qu.: 4.530 1st Qu.:0.7775

# Median : 9.240 Median :2.2100

# Mean : 8.527 Mean :2.4124

# 3rd Qu.:13.047 3rd Qu.:3.7475

# Max. :14.910 Max. :6.1000

cor(Chocolate, Nobel)

# [1] 0.8858229

nc=lm(Nobel ~ Chocolate)

nc

#Call:

#lm(formula = Nobel ~ Chocolate)

#

#Coefficients:

# (Intercept) Chocolate

# -0.7688 0.3731

plot(Chocolate, Nobel, main="Nobel Laureates per Pound of Chocolate Consumed")

abline(nc, col="red", lwd=2)

Line Equation for Best Fit: number of Nobels = -0.7688 + ((0.3731) \* pounds of chocolate)

The theorical model (with error term) was: number of Nobels = (– 0.8+ random error term) + ((0.4) \* pounds of choclate) where number of nobels >= 0.

On average, for every 10 pounds of chocolate consumed, the number of nobel laureates awarded per 10 million population increases roughly by 3 which is inline with the data that was generated. Note that the model indicates that there is a limitation in that it predicts that an estimated point value of greater than 2 pounds of chocolate (0.7688/0.3731) must be consumed before any nobel laureates are predicted to be awarded.

Both models depict that there is a positive and an increasing relationship between the number of pounds of chocolate consumed and the number of nobel laureates awarded (correlation=0.88). The slope of the relationship is similar (0.4 vs 0.3731). The y-intercept for both models is negative and similar (-0.8 vs -0.7688); thus, it would seem that the application of the random error term has slightly shifted the y-intercept in a positive direction.

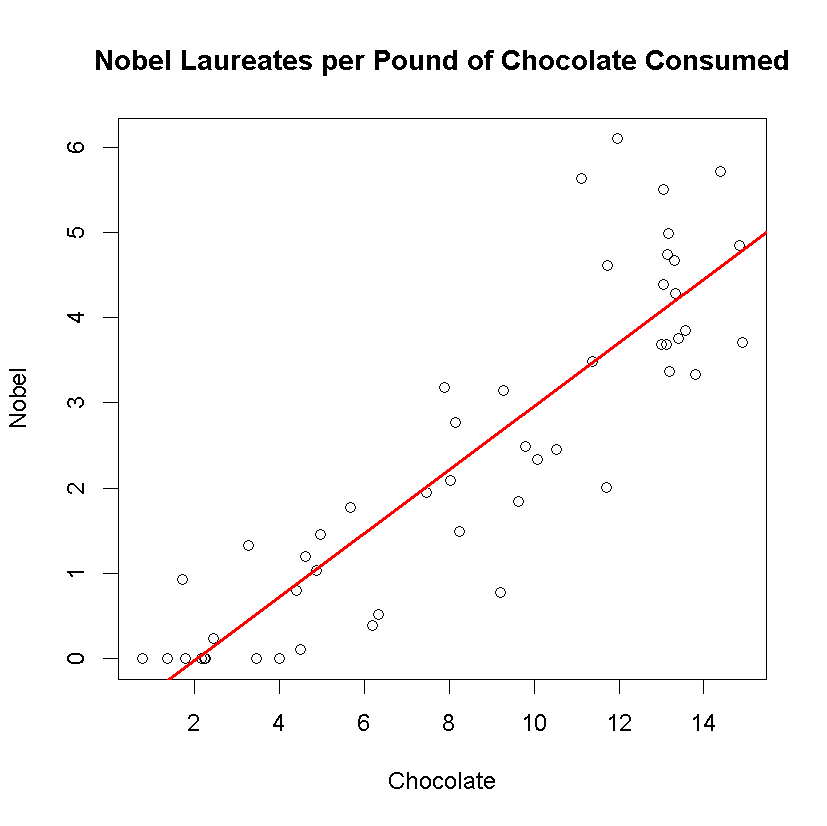


Figure 3

*2(c) State the null and alternative hypotheses for a test of whether the number of Nobel Prize winners (per 10 million population) is associated with per-capita chocolate consumption.*

H0: At the 95% confidence level, there is no relationship between the number of pounds of chocolate consumed per 10 million population and the number of Nobel Prize winners awarded.

H1: At the 95% confidence level, there is a relationship between the number of pounds of chocolate consumed per 10 million population and the number of Nobel Prize winners awarded.

*2(d) State your conclusion about the hypotheses in part c, in the context of the problem.*

The experiment is attempting to determine if there exists a relationship between the number of pounds of chocolate that is consumed per 10 million chocolate consumers in the population and the number of Noble Laureates winners that are awarded. If there is a relationship, then the experiment attempts to determine the nature of that relationship (positive or negative) and how strong that relationship is.

*2 (e) Graph the diagnostic plots for the regression. Explain what the top 2 plots tell us.*

par(mfrow =c(2,2))

> plot(nc)

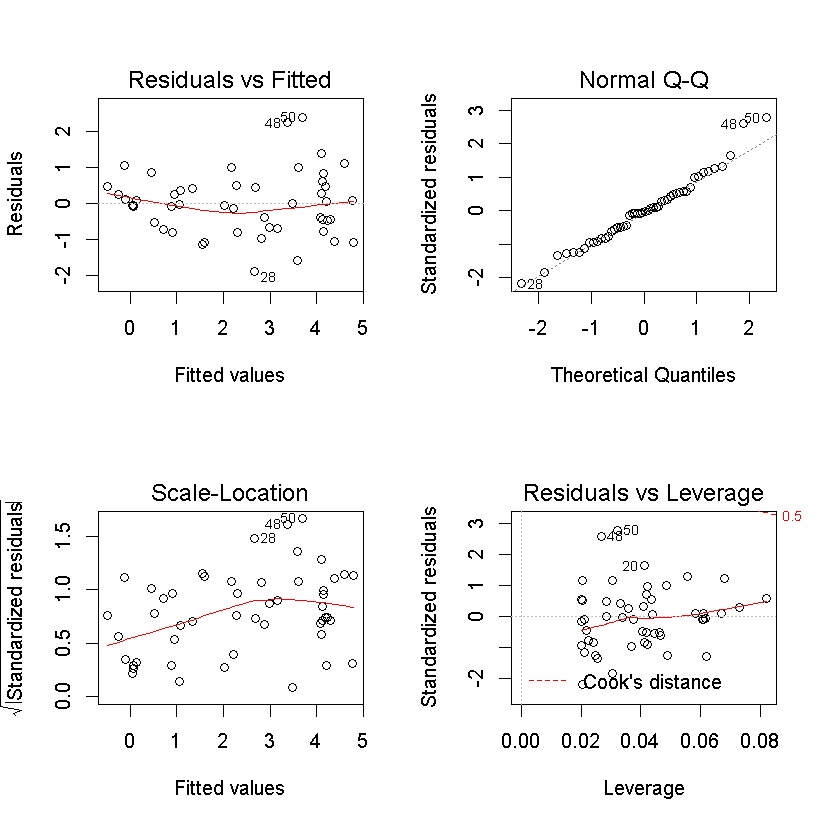


Figure 4

The Residuals vs Fitted plot shows the residuals [or the difference between the true (observed) value minus the estimated (predicted) value)] when compared to the Fitted [or predicted] values for the dependent variable. An examination of the distribution of the results of this function indicate a less than random dispersal of points, for there seems to be a greater number of point below the zero y-intercept that tends towards a parabola curve indicating a possible need to apply a squared transformation. Furthermore, it is noted that the distribution is increase left to right (a log transformation might be an interesting transformation) and that ther are odd clusters of points in the upper right-hand of the plot.

The Normal Q-Q plot indicates how well do the residuals fit a normal distribution. This is a key indicator, for in order to use linear regression, the residuals should conform to a normal distribution. This plot indicates that at the upper and lower ends of the curve the prediction model fails to conform to a normal distribution, which limits the use of the model and its validity. To improve the model, it is suggested that a squared transformation be applied.

*Problem 3: In homework 5, you counted the frequencies of letters in two encrypted texts. In this problem, you will use statistical analysis to identify the language in which the text was written, and decrypt it.*

*3 (a) Read the letter frequencies from encryptedA into R and attach the data.*

*Use the following code to make a barplot of the letter frequencies, with the letters listed in order of increasing frequency: (Here I’ve assumed that your columns were named “key” and “frequency”.)*

*encrypt\_order = order(frequency)*

*barplot( frequency [encrypt\_order], names.arg = key[encrypt\_order] )*

*Be sure you understand what this code does.*

setwd("c:/Temp1")

encryptedA = read.csv("encryptedA.csv")

attach(encryptedA)

encryptedA

key\_a frequency\_a

1 a 78

2 b 31

3 c 88

4 d 28

5 e 0

6 f 18

7 g 78

8 h 0

9 i 7

10 j 36

11 k 0

12 l 32

13 m 76

14 n 0

15 o 0

16 p 0

17 q 41

18 r 114

19 s 0

20 t 19

21 u 70

22 v 27

23 w 76

24 x 72

25 y 40

26 z 16

encrypt\_order\_a = order(frequency\_a)

barplot( frequency\_a [encrypt\_order\_a], names.arg = key\_a[encrypt\_order\_a], main="encryptedA.csv", xlab="EncryptedA Letters", ylab="Frequency of Occurrence")

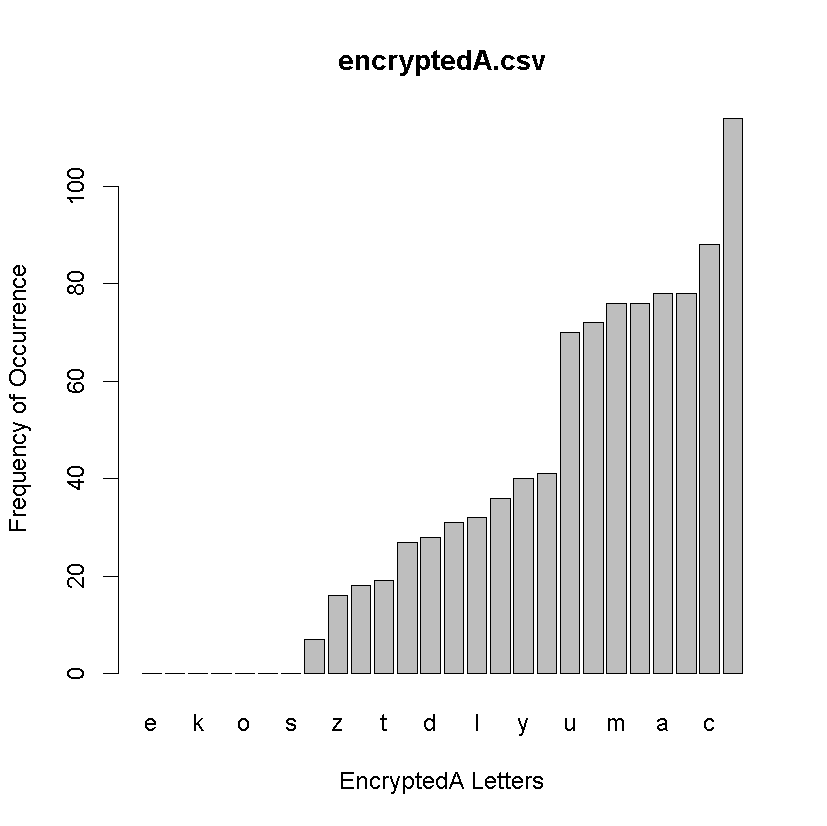


Figure 5

# Process encryptedB.csv

encryptedB = read.csv("encryptedB.csv")

attach(encryptedB)

encryptedB

> encryptedB

key\_b frequency\_b

1 a 7

2 b 3

3 c 40

4 d 29

5 e 28

6 f 1

7 g 51

8 h 48

9 i 6

10 j 19

11 k 0

12 l 11

13 m 0

14 n 79

15 o 48

16 p 23

17 q 16

18 r 2

19 s 61

20 t 31

21 u 16

22 v 122

23 w 41

24 x 93

25 y 90

26 z 61

encrypt\_order\_b = order(frequency\_b)

barplot( frequency\_b [encrypt\_order\_b], names.arg = key\_b[encrypt\_order\_b], main="encryptedB.csv", xlab="EncryptedB Letters", ylab="Frequency of Occurrence")

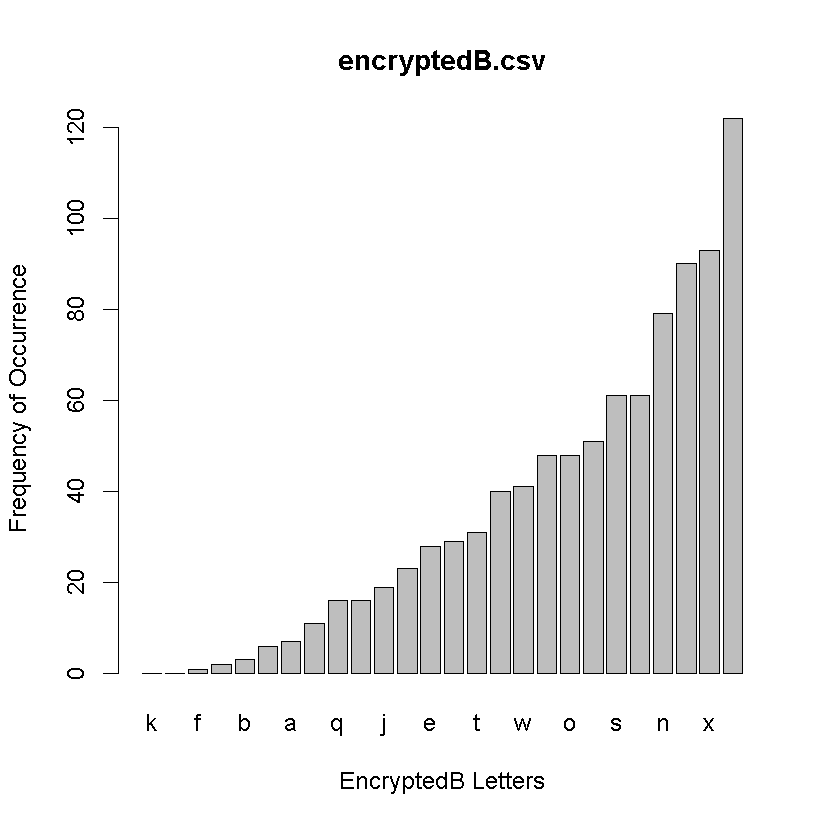


Figure 6

*3(b) The file Letter Frequencies.csv contains data on the frequencies of letters in different languages. (Source:* [*http://www.sttmedia.com/characterfrequency-english*](http://www.sttmedia.com/characterfrequency-english) *and* [*http://www.sttmedia.com/characterfrequency-welsh*](http://www.sttmedia.com/characterfrequency-welsh)*, accessed 21 August 2015. Used by permission of Stefan Trost.) Read these data into R.*

LetterFrequencies = read.csv("Letter Frequencies.csv")

attach(LetterFrequencies)

LetterFrequencies

# Letter English Welsh

#1 A 0.083441721 0.094335819

#2 B 0.015407704 0.018343076

#3 C 0.027313657 0.029127192

#4 D 0.041420710 0.099576698

#5 E 0.126063032 0.083753276

#6 F 0.020310155 0.031445273

#7 G 0.019209605 0.034368071

#8 H 0.061130565 0.039004233

#9 I 0.067133567 0.070348720

#10 J 0.002301151 0.001310220

#11 K 0.008704352 0.000000000

#12 L 0.042421211 0.050695424

#13 M 0.025312656 0.024994961

#14 N 0.068034017 0.081838339

#15 O 0.077038519 0.056339448

#16 P 0.016608304 0.009171538

#17 Q 0.000900450 0.000000000

#18 R 0.056828414 0.065712558

#19 S 0.061130565 0.029328764

#20 T 0.093746873 0.028623261

#21 U 0.028514257 0.026002822

#22 V 0.010605303 0.000000000

#23 W 0.023411706 0.040112880

#24 X 0.002001001 0.000000000

#25 Y 0.020410205 0.085567426

#26 Z 0.000600300 0.000000000

*3(c) In a single graphing window, display two bar plots:*

*A plot on top showing the encrypted frequencies, and*

*a plot below it showing the frequencies of letters in English.*

*Each plot should be sorted in order of increasing frequency.*

*Each plot should also have a title telling whether it is from the encrypted text or from plain English.*

# sort the letters by increasing frequency of occurrence for each language

encrypt\_order\_english = order(English)

encrypt\_order\_welsh = order(Welsh)

# divide the plot window into two vertical stacked windows

# plot the encrypted text A and the plain English letter frequencies in the top and bottom windows

par(mfrow=c(2,1))

barplot( frequency\_a [encrypt\_order\_a], names.arg = key\_a[encrypt\_order\_a], main=" Encrypted Text Letter Frequencies from encryptedA.csv ", xlab="EncryptedA Letters", ylab="Frequency of Occurrence")

barplot( English [encrypt\_order\_english], names.arg = Letter[encrypt\_order\_english], main="Plain English Letter Frequencies from Frequencies.csv", xlab="English Letters", ylab="Frequency of Occurrence")

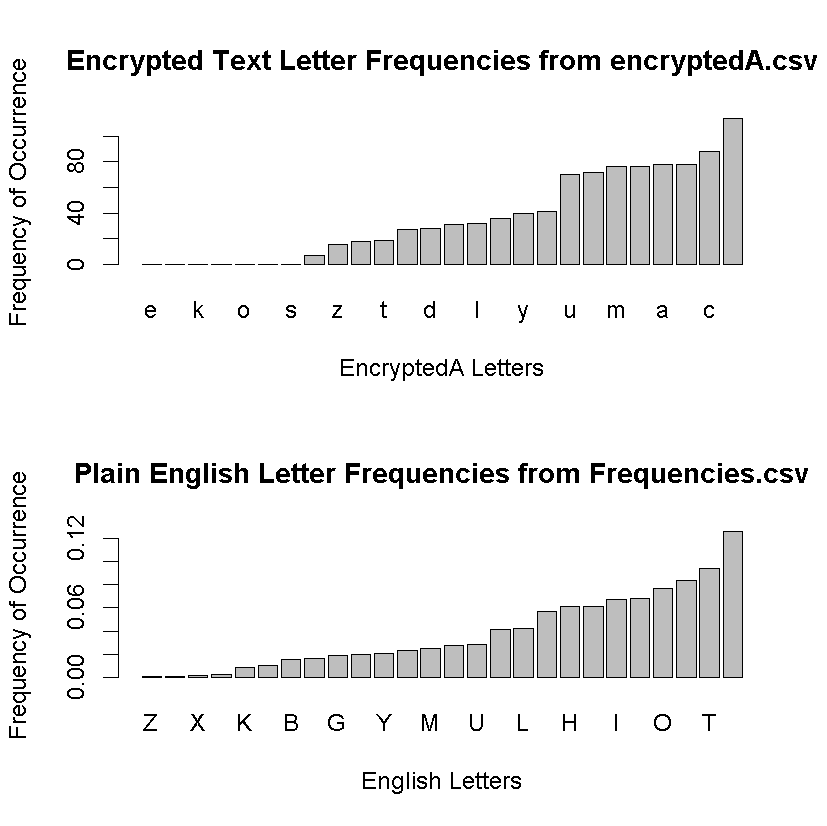


Figure 7

# plot the encrypted text A and the plain Welsh letter frequencies in the top and bottom windows

par(mfrow=c(2,1))

barplot( frequency\_a [encrypt\_order\_a], names.arg = key\_a[encrypt\_order\_a], main="Encrypted Text Letter Frequencies from encryptedA.csv", xlab="EncryptedA Letters", ylab="Frequency of Occurrence")

barplot( Welsh [encrypt\_order\_welsh], names.arg = Letter[encrypt\_order\_welsh], main="Plain Welsh Letter Frequencies from Frequencies.csv", xlab="Welsh Letters", ylab="Frequency of Occurrence")

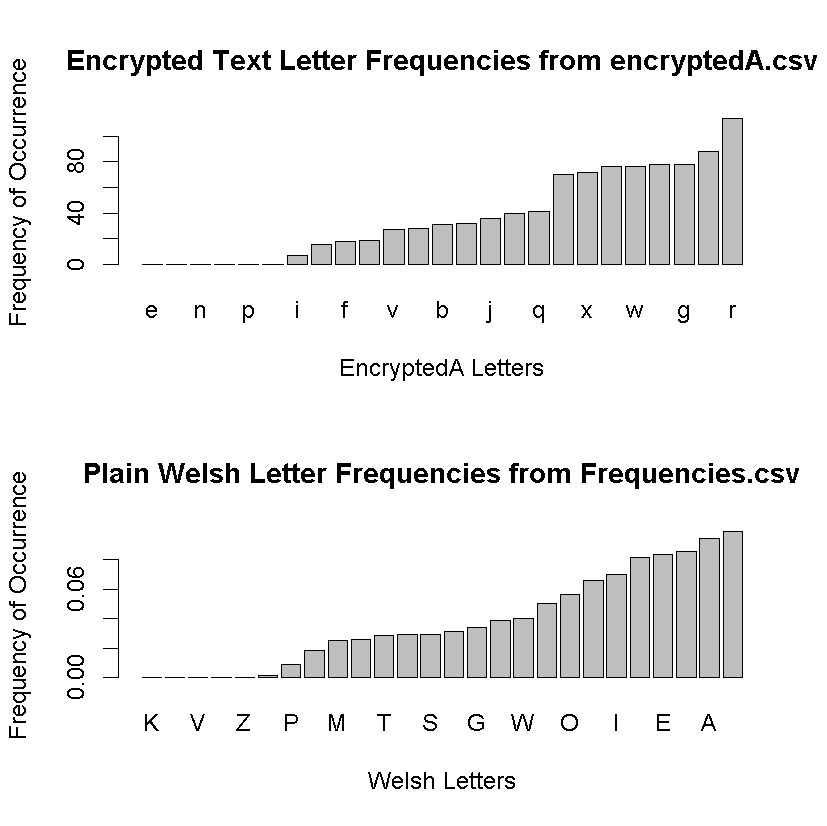


Figure 8

# Process encryptedB.csv

# plot the encrypted text B and the plain English letter frequencies in the top and bottom windows

par(mfrow=c(2,1))

barplot( frequency\_b [encrypt\_order\_b], names.arg = key\_b[encrypt\_order\_b], main=" Encrypted Text Letter Frequencies from encryptedB.csv ", xlab="EncryptedB Letters", ylab="Frequency of Occurrence")

barplot( English [encrypt\_order\_english], names.arg = Letter[encrypt\_order\_english], main="Plain English Letter Frequencies from Frequencies.csv", xlab="English Letters", ylab="Frequency of Occurrence")

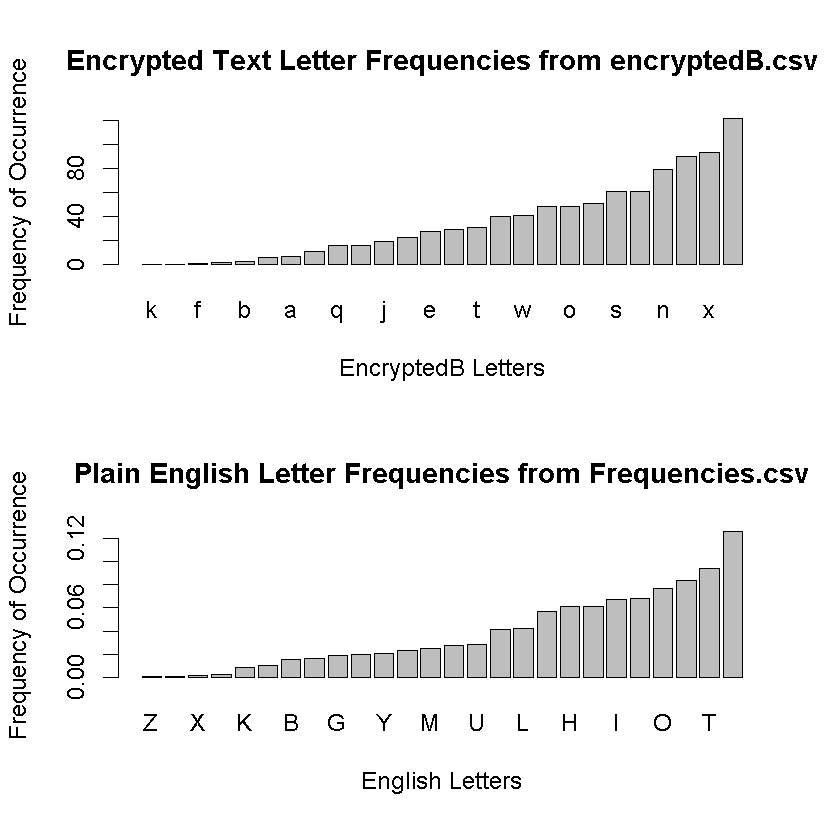


Figure 9

# plot the encrypted text B and the plain Welsh letter frequencies in the top and bottom windows

par(mfrow=c(2,1))

barplot( frequency\_b [encrypt\_order\_b], names.arg = key\_b[encrypt\_order\_b], main="Encrypted Text Letter Frequencies from encryptedB.csv", xlab="EncryptedB Letters", ylab="Frequency of Occurrence")

barplot( Welsh [encrypt\_order\_welsh], names.arg = Letter[encrypt\_order\_welsh], main="Plain Welsh Letter Frequencies from Frequencies.csv", xlab="Welsh Letters", ylab="Frequency of Occurrence")

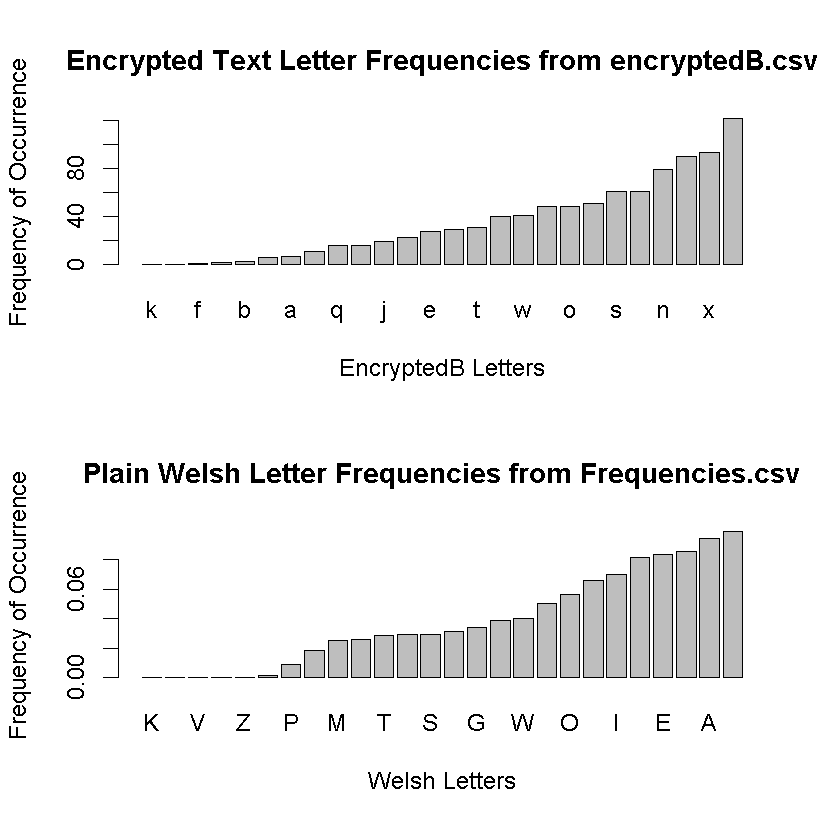


Figure 10

*3(d) Based on the* ***shape*** *of the plots, do you think it is likely that the encrypted text came from English? Explain.*

*(Note: The order of the letters along the horizontal axis of each plot will be quite different, because one plot shows the frequencies in plain English, and the other shows the frequencies in the encrypted text. So, you should ignore what letter is written below each bar when answering this question. Instead, look at things like the relative frequency of the most-common letter and the second-most common.)*

# Process encryptedA.csv

Based on a comparison of the shapes of the distributions between the encrypted text A letter frequencies and the plain English letter frequencies (Figure 7) it does not appear that the encrypted text represents encrypted English text. The justification for this conculsion is that the general distribution and shape of the two bar plots differs: the encrypted bar plot exhibits more letters that have the least frequency of occurrence count (e:i) and also plateaus towards the end of the distribution (x:r). In contrast, the bar plot (Figure 8)of the plain English seems to have a greater number of frequency counts at the left tail of the distribution, exhibits miminal plateaus, and seems to have a consistent increase in frequency counts.

Based on a comparison of the shapes of the distributions between the encrypted text A letter frequencies and the plain Welsh letter frequencies (Figure 8) it does appear that the encrypted text represents encrypted Welsh text. The justification for this conculsion is that the general distribution and shape of the two bar plots are similar: Both bar plots exhibits more letters that have the least frequency of occurrence count (e:i) and also plateaus towards the end of the distribution (x:r).

# Process encryptedB.csv

Based on a comparison of the shapes of the distributions between the encrypted text B letter frequencies and the plain Welsh letter frequencies (Figure 10) appear that the encrypted text represents encrypted Welsh text. The justification for this conculsion is that the general distribution and shape of the two bar plots are similar: Both bar plots exhibits more letters that have the least frequency of occurrence count (e:i) and also plateaus towards the end of the distribution (x:r).

Based on a comparison of the shapes of the distributions between the encrypted text B letter frequencies and the plain English letter frequencies (Figure 9) it does not appear that the encrypted text represents encrypted English text. The justification for this conculsion is that the general distribution and shape of the two bar plots differs: the encrypted bar plot exhibits more letters that have the least frequency of occurrence count (e:i) and also plateaus towards the end of the distribution (x:r). In contrast, the bar plot (Figure 10)of the plain English seems to have a greater number of frequency counts at the left tail of the distribution, exhibits miminal plateaus, and seems to have a consistent increase in frequency counts.

*3(e) We want to conduct a hypothesis test to be more precise about whether it is plausible that the text came from English.*

*To do this, we will pair up each letter in the encrypted text with a letter in English, based on the order of frequency.*

*So, encryptedA “r” is paired with English “e”, encryptedA “c” is paired with English “t”, etc.*

*Then we will test whether the resulting letter frequencies plausibly come from a random sample of English words.*

*To pair up the letters, sort the vector of counts from the encrypted text in order of increasing frequency, and store it as a new vector.*

*Then do the same thing with the vector of frequencies from English.*

*You already sorted the counts from the encrypted text in increasing order in part a) of this problem.*

*This problem is asking you to store the sorted vector as a variable, and also to sort the theoretical English frequencies in increasing order.*

**#ENCRYPTEDA**

encryptedA\_letters\_sorted = key\_a[encrypt\_order\_a]

encryptedA\_letters\_sorted

[1] e h k n o p s i z f t v d b l j y q u x m w a g c r

Levels: a b c d e f g h i j k l m n o p q r s t u v w x y z

length(encryptedA\_letters\_sorted)

#[1] 26

encryptedA\_freq\_sorted = frequency\_a[encrypt\_order\_a]

encryptedA\_freq\_sorted

[1] 0 0 0 0 0 0 0 7 16 18 19 27 28 31 32 36 40 41 70

[20] 72 76 76 78 78 88 114

length(encryptedA\_freq\_sorted)

#[1] 26

**# ENGLISH**

english\_letters\_sorted = Letter[encrypt\_order\_english]

english\_letters\_sorted

# [1] Z Q X J K V B P G F Y W M C U D L R H S I N O A T E

#Levels: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

#[1] 26

length(english\_letters\_sorted)

#[1] 26

english\_freq\_sorted = English [encrypt\_order\_english]

english\_freq\_sorted

# [1] 0.000600300 0.000900450 0.002001001 0.002301151 0.008704352 0.010605303

# [7] 0.015407704 0.016608304 0.019209605 0.020310155 0.020410205 0.023411706

# [13] 0.025312656 0.027313657 0.028514257 0.041420710 0.042421211 0.056828414

# [19] 0.061130565 0.061130565 0.067133567 0.068034017 0.077038519 0.083441721

# [25] 0.093746873 0.126063032

length(english\_freq\_sorted)

#[1] 26

**#WELSH**

welsh\_letters\_sorted = Letter[encrypt\_order\_welsh]

welsh\_letters\_sorted

#[1] K Q V X Z J P B M U T C S F G H W L O R I N E Y A D

#Levels: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

length(welsh\_letters\_sorted)

#[1] 26

welsh\_freq\_sorted = Welsh [encrypt\_order\_welsh]

welsh\_freq\_sorted

[1] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.001310220

[7] 0.009171538 0.018343076 0.024994961 0.026002822 0.028623261 0.029127192

[13] 0.029328764 0.031445273 0.034368071 0.039004233 0.040112880 0.050695424

[19] 0.056339448 0.065712558 0.070348720 0.081838339 0.083753276 0.085567426

[25] 0.094335819 0.099576698

length(welsh\_freq\_sorted)

#[1] 26

**#ENCRYPTEDB**

encryptedB\_letters\_sorted = key\_b[encrypt\_order\_b]

encryptedB\_letters\_sorted

[1] k m f r b i a l q u j p e d t c w h o g s z n y x v

Levels: a b c d e f g h i j k l m n o p q r s t u v w x y z

length(encryptedB\_letters\_sorted)

#[1] 26

encryptedB\_freq\_sorted = frequency\_b[encrypt\_order\_b]

encryptedB\_freq\_sorted

[1] 0 0 1 2 3 6 7 11 16 16 19 23 28 29 31 40 41 48 48

[20] 51 61 61 79 90 93 122

length(encryptedB\_freq\_sorted)

#[1] 26

*3(f) To pair up the letters, we need the data (the counts of letters from encryptedA.txt) and the probability model (the theoretical frequencies from Letter Frequencies.csv) to have the same number of letters.*

*Depending on how you formatted your output from Python, your letter counts may include 20 or 26 letters.*

*This is due to the fact that some letters did not appear in the encrypted text, so they appeared 0 times.*

*If necessary, prepend 6 zeroes to the count vector to make it the same length as the theoretical frequencies: count = c( rep(0, 6), count )*

**NOT NECESSARY, encryptedA.csv & encryptedB.csv both have the full 26 letters and 26 frequency counts in each file.**

*3(g) State the null and alternative hypotheses for a chi-squared Goodness of Fit test of this question.*

**encryptedA English Hypothesis**

H0: the distribution of the letters in the encryptedA.csv file is a good fit for the English letters in the Letter\_frequency.csv file.

H1: At least one of the proportions of letters in the encryptedA.csv file is different from the proprotions of the English letters in the Letter\_frequency.csv file.

**encryptedA Welsh Hypothesis**

H0: the distribution of the letters in the encryptedA.csv file is a good fit for the Welsh letters in the Letter\_frequency.csv file.

H1: At least one of the proportions of letters in the encryptedA.csv file is different from the proprotions of the Welsh letters in the Letter\_frequency.csv file.

**encryptedB English Hypothesis**

H0: the distribution of the letters in the encryptedB.csv file is a good fit for the English letters in the Letter\_frequency.csv file.

H1: At least one of the proportions of letters in the encryptedB.csv file is different from the proprotions of the English letters in the Letter\_frequency.csv file.

**encryptedA Welsh Hypothesis**

H0: the distribution of the letters in the encryptedB.csv file is a good fit for the Welsh letters in the Letter\_frequency.csv file.

H1: At least one of the proportions of letters in the encryptedB.csv file is different from the proprotions of the Welsh letters in the Letter\_frequency.csv file.

*3(h) To satisfy the assumptions of a Goodness of Fit test, we need the expected counts of each category to be greater than or equal to 5. Find the total number of letters in the encrypted text. Then multiply this number by the probabilities from Letter Frequencies.csv to get the expected counts.*

**EncryptedA**

Total sum of frequency\_a for letters a:z = 947

|  |  |  |  |
| --- | --- | --- | --- |
| key\_a | frequency\_a | frequency\_a\_expected\_english | frequency\_a\_expected\_welsh |
| a | 78 | 79.01930979 | 89.33602059 |
| b | 31 | 14.59109569 | 17.37089297 |
| c | 88 | 25.86603318 | 27.58345082 |
| d | 28 | 39.22541237 | 94.29913301 |
| e | 0 | 119.3816913 | 79.31435237 |
| f | 18 | 19.23371679 | 29.77867353 |
| g | 78 | 18.19149594 | 32.54656324 |
| h | 0 | 57.89064506 | 36.93700865 |
| i | 7 | 63.57548795 | 66.62023784 |
| j | 36 | 2.179189997 | 1.24077834 |
| k | 0 | 8.243021344 | 0 |
| l | 32 | 40.17288682 | 48.00856653 |
| m | 76 | 23.97108523 | 23.67022807 |
| n | 0 | 64.4282141 | 77.50090703 |
| o | 0 | 72.95547749 | 53.35345726 |
| p | 0 | 15.72806389 | 8.685446486 |
| q | 41 | 0.85272615 | 0 |
| r | 114 | 53.81650806 | 62.22979243 |
| s | 0 | 57.89064506 | 27.77433951 |
| t | 19 | 88.77828873 | 27.10622817 |
| u | 70 | 27.00300138 | 24.62467243 |
| v | 27 | 10.04322194 | 0 |
| w | 76 | 22.17088558 | 37.98689736 |
| x | 72 | 1.894947947 | 0 |
| y | 40 | 19.32846414 | 81.03235242 |
| z | 16 | 0.5684841 | 0 |

**encryptedB**

Total sum of frequency\_a for letters a:z = 926

|  |  |  |  |
| --- | --- | --- | --- |
| key\_b | frequency\_b | frequency\_b\_expected\_english | frequency\_b\_expected\_welsh |
| a | 7 | 77.26703365 | 87.35496839 |
| b | 3 | 14.2675339 | 16.98568838 |
| c | 40 | 25.29244638 | 26.97177979 |
| d | 29 | 38.35557746 | 92.20802235 |
| e | 28 | 116.7343676 | 77.55553358 |
| f | 1 | 18.80720353 | 29.1183228 |
| g | 51 | 17.78809423 | 31.82483375 |
| h | 48 | 56.60690319 | 36.11791976 |
| i | 6 | 62.16568304 | 65.14291472 |
| j | 19 | 2.130865826 | 1.21326372 |
| k | 0 | 8.060229952 | 0 |
| l | 11 | 39.28204139 | 46.94396262 |
| m | 0 | 23.43951946 | 23.14533389 |
| n | 79 | 62.99949974 | 75.78230191 |
| o | 48 | 71.33766859 | 52.17032885 |
| p | 23 | 15.3792895 | 8.492844188 |
| q | 16 | 0.8338167 | 0 |
| r | 2 | 52.62311136 | 60.84982871 |
| s | 61 | 56.60690319 | 27.15843546 |
| t | 31 | 86.8096044 | 26.50513969 |
| u | 16 | 26.40420198 | 24.07861317 |
| v | 122 | 9.820510578 | 0 |
| w | 41 | 21.67923976 | 37.14452688 |
| x | 93 | 1.852926926 | 0 |
| y | 90 | 18.89984983 | 79.23543648 |
| z | 61 | 0.5558778 | 0 |

*3(i) Combine categories (letters) to get expected counts that are greater than or equal to 5.* ***For example****, if you decided to combine the first two categories, you could use the code: sortEnglish\_combined = c( sum(sortEnglish[1:2]), sortEnglish[3:26] )*

*Combine the same categories in the encrypted counts.*

**encryptedA Adjusted**

|  |  |  |
| --- | --- | --- |
| **key\_a\_adj** | **frequency\_a\_adj\_english** | **Expected\_a\_adj\_english** |
| a | 78 | 79.01931 |
| b | 31 | 14.5911 |
| c | 88 | 25.86603 |
| d | 28 | 39.22541 |
| e | 0 | 119.3817 |
| f | 18 | 19.23372 |
| g | 78 | 18.1915 |
| h | 0 | 57.89065 |
| i | 7 | 63.57549 |
| jk | 36 | 10.42221 |
| l | 32 | 40.17289 |
| m | 76 | 23.97109 |
| n | 0 | 64.42821 |
| o | 0 | 72.95548 |
| pq | 41 | 16.58079 |
| r | 114 | 53.81651 |
| s | 0 | 57.89065 |
| t | 19 | 88.77829 |
| uv | 97 | 37.04622 |
| ws | 148 | 24.06583 |
| yz | 56 | 19.89695 |

frequency\_a\_adj\_english = read.csv("frequency\_a\_adj\_english.csv")

attach(frequency\_a\_adj\_english)

frequency\_a\_adj\_english

frequency\_a\_adj\_english = read.csv("frequency\_a\_adj\_english.csv")

attach(frequency\_a\_adj\_english)

frequency\_a\_adj\_english

frequency\_a\_adj\_english

1 78

2 31

3 88

4 28

5 0

6 18

7 78

8 0

9 7

10 36

11 32

12 76

13 0

14 0

15 41

16 114

17 0

18 19

19 97

20 148

21 56

Expected\_a\_adj\_english = read.csv("Expected\_a\_adj\_english.csv")

attach(Expected\_a\_adj\_english)

Expected\_a\_adj\_english

Expected\_a\_adj\_english

1 79.01931

2 14.59110

3 25.86603

4 39.22541

5 119.38169

6 19.23372

7 18.19150

8 57.89065

9 63.57549

10 10.42221

11 40.17289

12 23.97109

13 64.42821

14 72.95548

15 16.58079

16 53.81651

17 57.89065

18 88.77829

19 37.04622

20 24.06583

21 19.89695

**Letter Frequences.csv Adjusted**

|  |  |  |
| --- | --- | --- |
| Letter | English | Welsh |
| A | 0.083442 | 0.094336 |
| B | 0.015408 | 0.018343 |
| C | 0.027314 | 0.029127 |
| D | 0.041421 | 0.099577 |
| E | 0.126063 | 0.083753 |
| F | 0.02031 | 0.031445 |
| G | 0.01921 | 0.034368 |
| H | 0.061131 | 0.039004 |
| I | 0.067134 | 0.070349 |
| JK | 0.011006 | 0.00131 |
| L | 0.042421 | 0.050695 |
| M | 0.025313 | 0.024995 |
| N | 0.068034 | 0.081838 |
| O | 0.077039 | 0.056339 |
| PQ | 0.017509 | 0.009172 |
| R | 0.056828 | 0.065713 |
| S | 0.061131 | 0.029329 |
| T | 0.093747 | 0.028623 |
| UV | 0.03912 | 0.026003 |
| WX | 0.025413 | 0.040113 |
| YZ | 0.021011 | 0.085567 |

**Letter\_Frequencies\_Adj\_English Adjusted**

Letter\_Frequencies\_Adj\_English = read.csv("Letter\_Frequencies\_Adj\_English.csv")

attach(Letter\_Frequencies\_Adj\_English)

Letter\_Frequencies\_Adj\_English

English\_Frequencies

1 0.08344172

2 0.01540770

3 0.02731366

4 0.04142071

5 0.12606303

6 0.02031015

7 0.01920961

8 0.06113056

9 0.06713357

10 0.01100550

11 0.04242121

12 0.02531266

13 0.06803402

14 0.07703852

15 0.01750875

16 0.05682841

17 0.06113056

18 0.09374687

19 0.03911956

20 0.02541271

21 0.02101050

**Letter\_Frequencies\_Adj\_Welsh Adjusted**

Letter\_Frequencies\_Adj\_Welsh = read.csv("Letter\_Frequencies\_Adj\_Welsh.csv")

attach(Letter\_Frequencies\_Adj\_Welsh)

Letter\_Frequencies\_Adj\_Welsh

Welsh

1 0.094335819

2 0.018343076

3 0.029127192

4 0.099576698

5 0.083753276

6 0.031445273

7 0.034368071

8 0.039004233

9 0.070348720

10 0.001310220

11 0.050695424

12 0.024994961

13 0.081838339

14 0.056339448

15 0.009171538

16 0.065712558

17 0.029328764

18 0.028623261

19 0.026002822

20 0.040112880

21 0.085567426

**encryptedB adjusted**

|  |  |  |
| --- | --- | --- |
| **key\_b\_adj** | **Frequency\_b\_adj\_English** | **Expected\_b\_adj\_English** |
| a | 7 | 87.35496839 |
| b | 3 | 16.98568838 |
| c | 40 | 26.97177979 |
| d | 29 | 92.20802235 |
| e | 28 | 77.55553358 |
| f | 1 | 29.1183228 |
| g | 51 | 31.82483375 |
| h | 48 | 36.11791976 |
| i | 6 | 65.14291472 |
| jk | 19 | 1.21326372 |
| l | 11 | 46.94396262 |
| m | 0 | 23.14533389 |
| n | 79 | 75.78230191 |
| o | 48 | 52.17032885 |
| pq | 39 | 8.492844188 |
| r | 2 | 60.84982871 |
| s | 61 | 27.15843546 |
| t | 31 | 26.50513969 |
| uv | 138 | 24.07861317 |
| wx | 134 | 37.14452688 |
| yz | 151 | 79.23543648 |

Frequency\_b\_adj\_English = read.csv("Frequency\_b\_adj\_English.csv")

attach(Frequency\_b\_adj\_English)

Frequency\_b\_adj\_English

Frequency\_b\_adj\_English

1 7

2 3

3 40

4 29

5 28

6 1

7 51

8 48

9 6

10 19

11 11

12 0

13 79

14 48

15 39

16 2

17 61

18 31

19 138

20 134

21 151

Expected\_b\_adj\_English = read.csv("Expected\_b\_adj\_English.csv")

attach(Expected\_b\_adj\_English)

Expected\_b\_adj\_English

Expected\_b\_adj\_English

1 87.354968

2 16.985688

3 26.971780

4 92.208022

5 77.555534

6 29.118323

7 31.824834

8 36.117920

9 65.142915

10 1.213264

11 46.943963

12 23.145334

13 75.782302

14 52.170329

15 8.492844

16 60.849829

17 27.158435

18 26.505140

19 24.078613

20 37.144527

21 79.235436

encryptedB\_adj = read.csv("encryptedB\_adj.csv")

attach(encryptedB\_adj)

encryptedB\_adj

*3(j) Use R to conduct the chi-squared Goodness of Fit test.*

*If you get the warning message, “Chi-squared approximation may be incorrect,” one of two things has happened:*

*You did not combine enough categories in step i, or*

*You are using the wrong syntax for the chi-squared Goodness of Fit test.*

*If either of these things is true, your results will not be reliable.*

**encryptedA English Goodness-of-Fit Test**

H0: the distribution of the letters in the encryptedA.csv file is a good fit for the English letters in the Letter\_frequency.csv file.

H1: At least one of the proportions of letters in the encryptedA.csv file is different from the proprotions of the English letters in the Letter\_frequency.csv file.

chisq.test(frequency\_a\_adj\_english, Expected\_a\_adj\_english)

Error in chisq.test(frequency\_a\_adj\_english, Expected\_a\_adj\_english) :

'x' and 'y' must have the same length

**encryptedA Welsh Goodness-of-Fit Test**

H0: the distribution of the letters in the encryptedA.csv file is a good fit for the Welsh letters in the Letter\_frequency.csv file.

H1: At least one of the proportions of letters in the encryptedA.csv file is different from the proprotions of the Welsh letters in the Letter\_frequency.csv file.

chisq.test(frequency\_a\_adj\_english, Expected\_a\_adj\_welsh)

**encryptedB English Goodness-of-Fit Test**

H0: the distribution of the letters in the encryptedB.csv file is a good fit for the English letters in the Letter\_frequency.csv file.

H1: At least one of the proportions of letters in the encryptedB.csv file is different from the proprotions of the English letters in the Letter\_frequency.csv file.

chisq.test(frequency\_b\_adj\_english, Expected\_b\_adj\_english)

**encryptedB Welsh Goodness-of-Fit Test**

H0: the distribution of the letters in the encryptedB.csv file is a good fit for the Welsh letters in the Letter\_frequency.csv file.

H1: At least one of the proportions of letters in the encryptedB.csv file is different from the proprotions of the Welsh letters in the Letter\_frequency.csv file.

chisq.test(frequency\_b\_adj\_english, Expected\_b\_adj\_welsh)

*3(k) State your conclusion in the context of the problem.*

*Note that the null hypothesis is that the observed counts of the most-frequent letter, 2nd-most frequent letter, etc. are consistent with the theoretical frequencies. Therefore, the null hypothesis is that the text is an encrypted piece of writing in English.*

*3(l) Repeat steps h-k for Welsh, and then repeat for both languages for encryptedB.  Based on the hypothesis tests, which text do you think came from which language?  How confident are you in your assessment?*

*3(m) Optional: Try to decrypt the English text. Simon Singh’s Black Chamber website (*[*http://www.simonsingh.net/The\_Black\_Chamber/substitutioncrackingtool.html*](http://www.simonsingh.net/The_Black_Chamber/substitutioncrackingtool.html)*) will automatically substitute letters for you, so you can test different possibilities for what English plaintext letter is represented by each letter in the ciphertext. Start by substituting the letter E for the most common letter in the ciphertext. Then use frequencies of letters in the ciphertext, common patterns of letters, and experimentation to determine other substitutions.*