**DS 710**

**Homework 6**

**R assignment**

1. Can we detect when a marketing campaign has been successful?

1. 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.)

> TableFarmVector = c(119000, 99000, 111000, 89000, 92000, 125000, 88000, 102000, 110000, 102000, 94000, 92000, 109000, 146000, 151000, 153000, 159000, 97000, 152000, 136000, 122000, 158000, 143000, 110000)

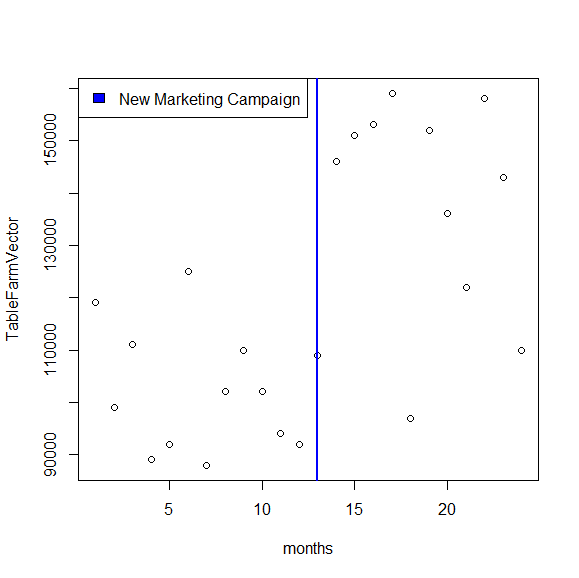
> months = c(1:24)

1. 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(months, TableFarmVector)

> abline(v = 13, col = "blue", lwd = 2)

> legend("topleft", legend = c("New Marketing Campaign"), fill = c("blue"))



1. Make 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.

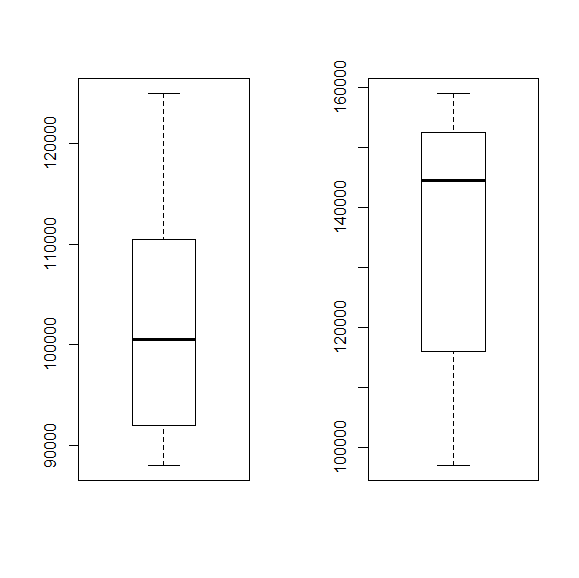
> preNewCampaign = c(119000, 99000, 111000, 89000, 92000, 125000, 88000, 102000, 110000, 102000, 94000, 92000)

> postNewCampaign = c(109000, 146000, 151000, 153000, 159000, 97000, 152000, 136000, 122000, 158000, 143000, 110000)

> par( mfrow = c( 1, 2 ) )

> boxplot(preNewCampaign)

>boxplot(postNewCampaign)



a.Left(preCampaign) b.right(postCampaign)

As you can see the box plots plotted side by side reveal that after the new campaign the revenue has moved up. The median prior to campaign was around 10,000 and after the implementation the median jumped to 14,000 and above.

1. 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?

I would not feel too good about just relying on those two. I would quite possibly increase the period of measurements, maybe even compare previous years to the current post campaign year for more in depth comparison.

1. 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.)

> mean(PreNewCampaign)

[1] 101916.7

> mean(postNewCampaign)

[1] 136333.3

𝐻0 : 𝜇1 >= 𝜇2  
𝐻↓𝑎 : 𝜇1 < 𝜇2

1. In a few sentences, explain why a 2-sample t-test is appropriate for testing the hypotheses in part e. T-test is used differentiate means of 2 population groups on whether they offer any difference. In our problem it is feasible to go ahead with this test to check for an significant difference in mean prior and post marketing campaign change.
2. Conduct a 2-sample t-test in R.  Include the R output and state your conclusion in the context of the problem.

> t.test(preNewCampaign,postNewCampaign, alternative = "less")

Welch Two Sample t-test

data: preNewCampaign and postNewCampaign

t = -4.8476, df = 17.335, p-value = 7.156e-05

alternative hypothesis: true difference in means is less than 0

95 percent confidence interval:

-Inf -22079.66

sample estimates:

mean of x mean of y

101916.7 136333.3

**The p –value is .** 7.156e-05**, less than the significance level of .01, there seems to be enough evidence to show period1(pre Campaign) and period 2(post campaign) have different means and infact higher. So we fail to reject the null hypothesis**

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

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

> choclate

[1] 5.436052 6.756622 14.851957 1.669580 1.036380 1.894941 5.357978

[8] 7.928586 5.214147 5.298095 14.249402 1.582665 8.714841 7.493554

[15] 4.269054 9.449248 5.172317 10.920410 11.416124 12.370067 7.939452

[22] 14.348084 6.186153 13.047349 7.167379 1.241972 5.155513 2.504383

[29] 14.142482 8.159183

> Ne

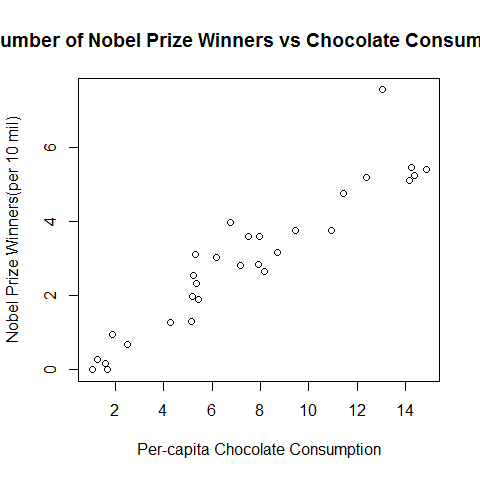
[1] 1.8852195 3.9820326 5.4056542 0.0000000 0.0000000 0.9514715 2.3198636

[8] 2.8516935 2.5417993 3.0980731 5.4442280 0.1797509 3.1768234 3.5968694

[15] 1.2656936 3.7663839 1.9796806 3.7496136 4.7478245 5.2002064 3.5874122

[22] 5.2379251 3.0388855 7.5487209 2.8277660 0.2848108 1.3008075 0.6758840

[29] 5.1189947 2.6629762

> plot(choclate,Ne, xlab = "Per-capita Chocolate Consumption", ylab = "Nobel Prize Winners(per 10 mil)", main = "Number of Nobel Prize Winners vs Chocolate Consumption")

1. 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(lm(Ne ~ choclate))

Call:

lm(formula = Ne ~ choclate)

Residuals:

Min 1Q Median 3Q Max

-0.7240 -0.3861 -0.1741 0.2686 2.2291

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -0.12764 0.24925 -0.512 0.613

choclate 0.41750 0.02941 14.194 2.57e-14 \*\*\*

---

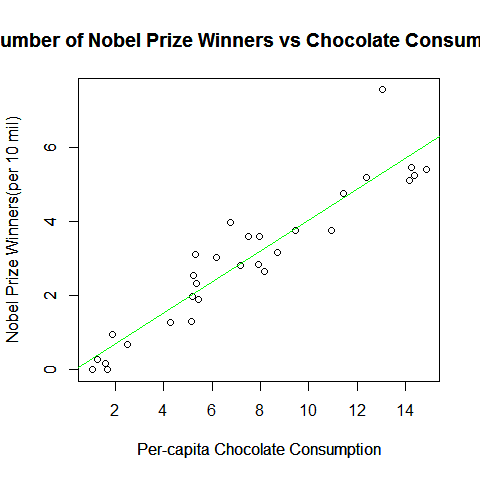
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.675 on 28 degrees of freedom

Multiple R-squared: 0.878, Adjusted R-squared: 0.8736

F-statistic: 201.5 on 1 and 28 DF, p-value: 2.575e-14

> abline(lm(Ne ~C), col = "green")



> model = lm(Ne~choclate)

> model

Call:

lm(formula = Ne ~ choclate)

Coefficients:

(Intercept) choclate

-0.1276 0.4175

Ne = (.4175 \* choclate ) - .0.1276 (Y = mx + b) Ne = nobel lauret and choclate=Choclate

1. 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.

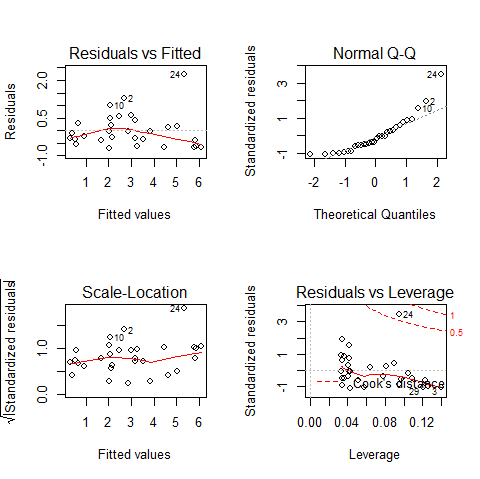
**H0 : μ1 = 0 There is no connection of eating chocolate per captia versus Nobel Prize winners**

**Ha: μ1 ≠ 0 There is a connection of eating chocolate per captia versus Nobel Prize winnerr**

1. State your conclusion about the hypotheses in part c, in the context of the problem.

We reject the null, the p value is = **2.575e-14** . So p value is less than .001. There seems to be significance evidence that points toward some kind of connection or association.

1. Graph the diagnostic plots for the regression.  Explain what they tell us.



**Residual vs fitted**: This is where we allow the graph to show upper and lower points of y = 0. In the particular graph, there seems to be a little bit fall of as it approaches the end, it seems to show more below y= 0 points towards the end.

**Normal quantile- quantile plot**: a somewhat linear regression seems to be taking place. A upward positive trend. The end of the plot seems to be not giving us the best option, maybe a transformation would help here, log.

**Scale-location**: To eliminate skewness and to enhance graph readability a particular function such square root is being applied in this graph.

Reduduals vs Leverage:

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.

1. 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 “count”.)

encrypt\_order = order(count)

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

Be sure you understand what this code does.

> encryptedA = read.csv("C:/Users/pedbv9699/Documents/GitHub/ds710assignment5/dictA1.csv")

> attach(encryptedA)

> head(encryptedA)

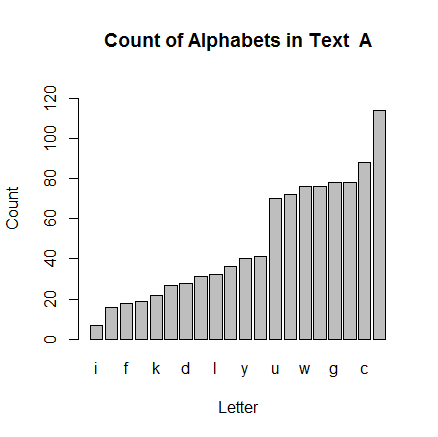
Key Count

1 d 28

2 j 36

3 w 76

> encrypt\_order = order(Count)

> barplot( Count[encrypt\_order], names.arg = Key[encrypt\_order], main = "Count of Alphabets in Text A", xlab = "Letter",ylab="Count",ylim=c(0,130))

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

> letterFreq = read.csv("C:/Users/pedbv9699/Documents/GitHub/ds710assignment6/Letter Frequencies.csv",quote="")

> attach(letterFreq)

> head(letterFreq)

Letter English Welsh

1 A 0.08344172 0.09433582

2 B 0.01540770 0.01834308

3 C 0.02731366 0.02912719

4 D 0.04142071 0.09957670

5 E 0.12606303 0.08375328

6 F 0.02031015 0.03144527

> Frequency\_order = order(English)

> Frequency\_order

[1] 26 17 24 10 11 22 2 16 7 6 25 23 13 3 21 4 12 18 8 19 9 14 15 1 20

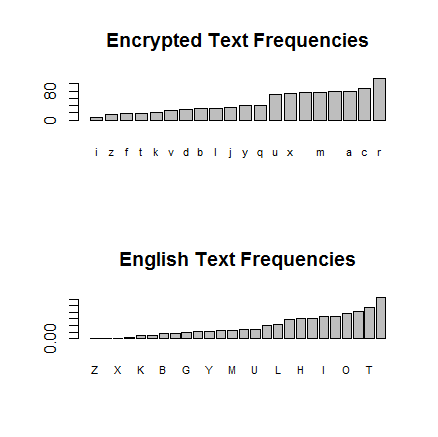
[26] 5

1. 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.

par(mfrow=c(2,1))

> barplot(Count[encrypt\_order],names.arg = Key[encrypt\_order],main = "Encrypted Text Frequencies",cex.names = .7)

> barplot(English[Frequency\_order],names.arg = Letter[Frequency\_order],main = "English Text Frequencies",cex.names = .7)



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

Plots have a strikingly similar frequencies as far as the letters go. However those letters are not the same across the board, might indicate that the Encrypted test might not be actually English.

(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.)

1. 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, encrypted1 “u” is paired with English “e”, encrypted1 “s” 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.

> Sorted\_textA = sort(Count)

> Sorted\_English = sort(letterFreq$English)

> Sorted\_Welsh = sort(letterFreq$Welsh)

> Sorted\_textA = sort(Count)

> Sorted\_textA

[1] 0 0 0 0 0 0 7 16 18 19 22 27 28 31 32 36 40

[18] 41 70 72 76 76 78 78 88 114

> Sorted\_English

[1] 0.000600300 0.000900450 0.002001001 0.002301151 0.008704352

[6] 0.010605303 0.015407704 0.016608304 0.019209605 0.020310155

[11] 0.020410205 0.023411706 0.025312656 0.027313657 0.028514257

[16] 0.041420710 0.042421211 0.056828414 0.061130565 0.061130565

[21] 0.067133567 0.068034017 0.077038519 0.083441721 0.093746873

[26] 0.126063032

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 )

> length(Sorted\_textA)

[1] 26

> length(Sorted\_English)

[1] 26

> length(Sorted\_Welsh)

[1] 26

1. No need have 26 letters already
2. State the null and alternative hypotheses for a chi-squared Goodness of Fit test of this question.

Null Hypothesis: The Letter frequency in the encrypted Text A and English is equivalent.

Alternative Hypothesis: At least one distribution of letters Text A is different from of letters frequency in English text.

1. 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.

> sum(Count)

[1] 969

Greater than 5, so we can proceed

> sum(Count)\*Sorted\_English

[1] 0.5816907 0.8725361 1.9389700 2.2298153 8.4345171

[6] 10.2765386 14.9300652 16.0934466 18.6141072 19.6805402

[11] 19.7774886 22.6859431 24.5279637 26.4669336 27.6303150

[16] 40.1366680 41.1061535 55.0667332 59.2355175 59.2355175

[21] 65.0524264 65.9249625 74.6503249 80.8550276 90.8407199

[26] 122.1550780

1. 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.

> TextA\_Sort\_Comb = c(sum(Sorted\_textA[1:4]),Sorted\_textA[5:26])

> TextA\_Sort\_Comb

[1] 0 0 0 7 16 18 19 22 27 28 31 32 36 40 41 70 72

[18] 76 76 78 78 88 114

> English\_Sort\_Comb

[1] 0.005802902 0.008704352 0.010605303 0.015407704 0.016608304

[6] 0.019209605 0.020310155 0.020410205 0.023411706 0.025312656

[11] 0.027313657 0.028514257 0.041420710 0.042421211 0.056828414

[16] 0.061130565 0.061130565 0.067133567 0.068034017 0.077038519

[21] 0.083441721 0.093746873 0.126063032

1. Use R to conduct the chi-squared Goodness of Fit test.

> chisq.test(TextA\_Sort\_Comb, p = English\_Sort\_Comb)

Chi-squared test for given probabilities

data: TextA\_Sort\_Comb

X-squared = 44.6421, df = 22, p-value = 0.002945

1. State your conclusion in the context of the problem.

We can possibly reject the null hypothesis and assume that text from A file is most likely English.

P value being .0002945 is lesser than .01 alpha. A strong indicator of rejecting null hypothesis

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?

Null Hypothesis: The Letter frequency in the encrypted Text A and Welsh is equivalent.

Alternative Hypothesis: At least one distribution of letters Text A is different from of letters frequency in Welsh text.

WELSH – Encrypted A

> Sorted\_Welsh

[1] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000

[6] 0.001310220 0.009171538 0.018343076 0.024994961 0.026002822

[11] 0.028623261 0.029127192 0.029328764 0.031445273 0.034368071

[16] 0.039004233 0.040112880 0.050695424 0.056339448 0.065712558

[21] 0.070348720 0.081838339 0.083753276 0.085567426 0.094335819

[26] 0.099576698

> Sorted\_Welsh \* sum(Count)

[1] 0.000000 0.000000 0.000000 0.000000 0.000000 1.269603 8.887220

[8] 17.774441 24.220117 25.196735 27.735940 28.224249 28.419572 30.470470

[15] 33.302661 37.795102 38.869381 49.123866 54.592925 63.675469 68.167910

[22] 79.301350 81.156924 82.914836 91.411409 96.489820

> Welsh\_Sort\_Comb = c(sum(Sorted\_Welsh[1:7]), Sorted\_Welsh[8:26])

> Welsh\_Sort\_Comb

[1] 0.01048176 0.01834308 0.02499496 0.02600282 0.02862326 0.02912719

[7] 0.02932876 0.03144527 0.03436807 0.03900423 0.04011288 0.05069542

[13] 0.05633945 0.06571256 0.07034872 0.08183834 0.08375328 0.08556743

[19] 0.09433582 0.09957670

> chisq.test(TextA\_Sort\_Comb, p = Welsh\_Sort\_Comb)

Chi-squared test for given probabilities

data: TextA\_Sort\_Comb

X-squared = 17.2398, df = 19, p-value = 0.5736

We fail to reject the null hypotheis,The P value suggests that there might NOT be enough evidence, as alpha being .05 and p- value being .5736, it is significantly larger. Insufficient evidence to say that letter frequencies from text A are different from that of Welsh Text

ENGLISH – ENCRYPTED B

> encryptedB = read.csv("C:/Users/pedbv9699/Documents/GitHub/ds710assignment5/dictB1.csv")

> attach(encryptedB)

>head(encrypted)

Key Count

1 y 92

2 f 2

3 q 17

4 k 85

5 s 61

6 b 5

Null Hypothesis: The Letter frequency in the encrypted Text B and English is equivalent.

Alternative Hypothesis: At least one distribution of letters Text B is different from of letters frequency in English text.

> Sorted\_textB = sort(encryptedB$Count)

> Sorted\_textB

[1] 2 4 5 6 13 13 14 17 19 23 23 28 30 34 43 48 48

[18] 54 56 61 61 79 85 92 97 123

> Sorted\_English \*sum(encryptedB$Count)

[1] 0.6471234 0.9706851 2.1570791 2.4806408 9.3832915

[6] 11.4325166 16.6095049 17.9037517 20.7079542 21.8943471

[11] 22.0022010 25.2378191 27.2870432 29.4441222 30.7383690

[16] 44.6515254 45.7300655 61.2610303 65.8987491 65.8987491

[21] 72.3699852 73.3406703 83.0475235 89.9501752 101.0591291

[26] 135.8959485

> English\_Sort\_ComboB = c(sum(Sorted\_English[1:4]), Sorted\_English[5:26])

> English\_Sort\_ComboB

[1] 0.005802902 0.008704352 0.010605303 0.015407704 0.016608304

[6] 0.019209605 0.020310155 0.020410205 0.023411706 0.025312656

[11] 0.027313657 0.028514257 0.041420710 0.042421211 0.056828414

[16] 0.061130565 0.061130565 0.067133567 0.068034017 0.077038519

[21] 0.083441721 0.093746873 0.126063032

English\_Sort\_ComboB \* Sum(encryptedB$Count)

Greater THAN 5

> English\_Sort\_ComboB \* sum(encryptedB$Count)

[1] 6.255528 9.383291 11.432517 16.609505 17.903752 20.707954

[7] 21.894347 22.002201 25.237819 27.287043 29.444122 30.738369

[13] 44.651525 45.730065 61.261030 65.898749 65.898749 72.369985

[19] 73.340670 83.047523 89.950175 101.059129 135.895948

> TextB\_Sort\_Comb = c(sum(Sorted\_textB[1:4]), Sorted\_textB [5:26])

> chisq.test(TextB\_Sort\_Comb, p = English\_Sort\_ComboB )

Chi-squared test for given probabilities

data: TextB\_Sort\_Comb

X-squared = 33.7073, df = 22, p-value = 0.05259

We Fail to reject the null hypothesis at alpha level of .05. There isn’t enough evidence to say that Welsh and Text B aren’t similar.

WELSH- Encrypted B

> Sorted\_Welsh

[1] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000

[6] 0.001310220 0.009171538 0.018343076 0.024994961 0.026002822

[11] 0.028623261 0.029127192 0.029328764 0.031445273 0.034368071

[16] 0.039004233 0.040112880 0.050695424 0.056339448 0.065712558

[21] 0.070348720 0.081838339 0.083753276 0.085567426 0.094335819

[26] 0.099576698

> > Sorted\_Welsh \* sum(encryptedB$Count)

[1] 0.000000 0.000000 0.000000 0.000000 0.000000 1.412417

[7] 9.886918 19.773836 26.944568 28.031042 30.855875 31.399113

[13] 31.616408 33.898004 37.048781 42.046563 43.241685 54.649667

[19] 60.733925 70.838138 75.835920 88.221729 90.286032 92.241685

[25] 101.694013 107.343680

> Welsh\_Sort\_ComboB = c(sum(Sorted\_Welsh[1:7]), Sorted\_Welsh[8:26])

> Welsh\_Sort\_ComboB

[1] 0.01048176 0.01834308 0.02499496 0.02600282 0.02862326 0.02912719

[7] 0.02932876 0.03144527 0.03436807 0.03900423 0.04011288 0.05069542

[13] 0.05633945 0.06571256 0.07034872 0.08183834 0.08375328 0.08556743

[19] 0.09433582 0.09957670

> Welsh\_Sort\_ComboB \* sum(encryptedB$Count)

[1] 11.29934 19.77384 26.94457 28.03104 30.85588 31.39911 31.61641

[8] 33.89800 37.04878 42.04656 43.24168 54.64967 60.73392 70.83814

[15] 75.83592 88.22173 90.28603 92.24169 101.69401 107.34368

TextB\_Sort\_Comb = c(sum(Sorted\_textB[1:7]), Sorted\_textB [8:26])

> chisq.test(TextB\_Sort\_Comb, p = Welsh\_Sort\_ComboB)

Chi-squared test for given probabilities

data: TextB\_Sort\_Comb

X-squared = 201.6661, df = 19, p-value < 2.2e-16

We can safely reject the null hypothesis the P-value is < 2.2e-16 a lot less than the alpha of .05 or .01. So we can say that Text B and Welsh text have at least one similarities.

1. Optional: Try to decrypt the English text. Simon Singh’s Black Chamber website (<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.

Submit a .doc, .docx, .rmd, or .pdf file to GitHub containing your R code, R output and graphs, and your written interpretations and explanations. (You may include your responses for problems 1, 2, and 3 in the same file.)