Adam Hendel
DS 710 - Homework 6
R assignment

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

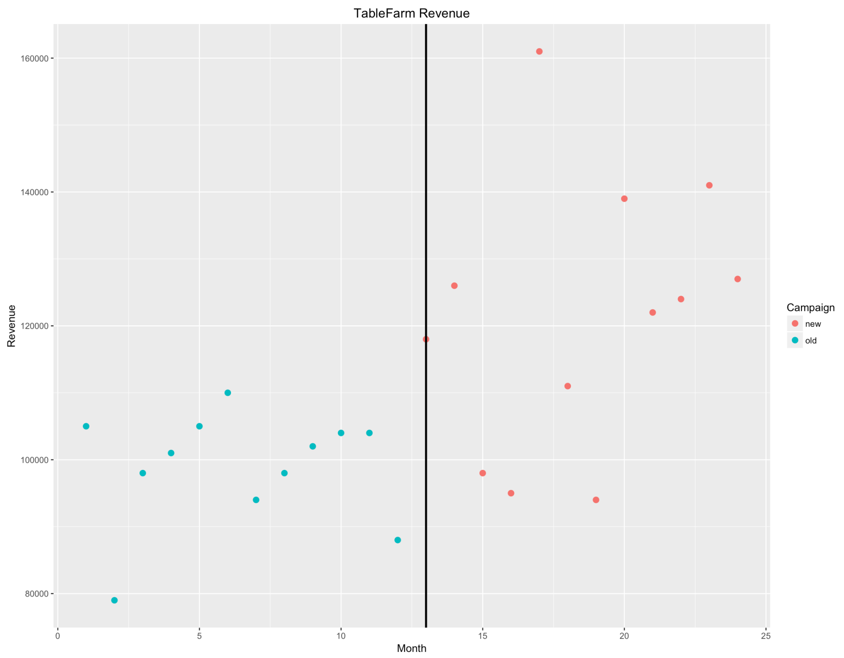
   a. On homework 4, you simulated data from the TableFarm salad chain before and after the implementation of a new marketing campaign. Read the combined data (both before and after) into R. (You could do this by saving the data as a .csv file and using read.csv(), or by copying the data into a text file, separating the values by commas, and enclosing the data in c(…) to make a vector.)

```
d <- read.csv('allRev.csv', header=FALSE)
d<-d$V1 #convert to vector
> str(d)
'data.frame':	24 obs. of  1 variable:
 $ V1: int  105000 79000 98000 101000 105000 110000 94000 98000 102000 104000 ...
```

   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.

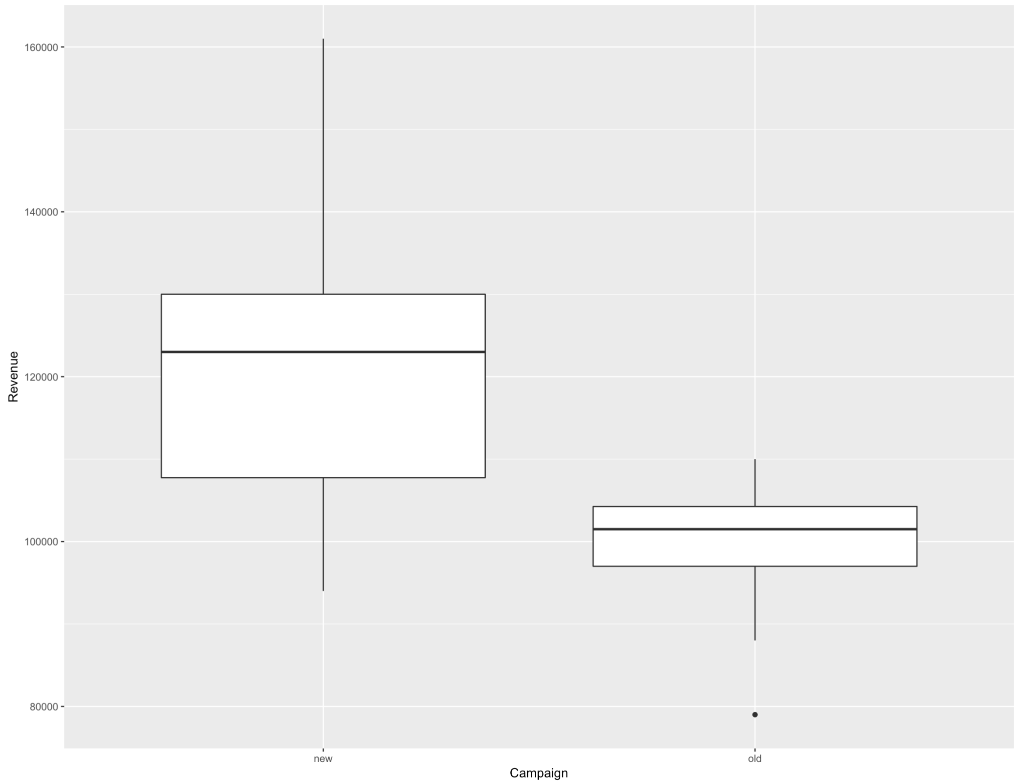
```
library(ggplot2)
d$month <- seq(1, 24, by=1)
d$category <- c(replicate(12, 'old'), replicate(12, 'new'))
names(d) <- c('Revenue', 'Month', 'Campaign')

ggplot(d, aes(x=Month, y=Revenue, col=Campaign)) +
  geom_point(size=2.5) +
  geom_vline(xintercept=13, col='black', size=1) +
  ggtitle('TableFarm Revenue') +
  theme(plot.title = element_text(hjust = 0.5))
```



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.

```
ggplot(d, aes(x=Campaign, y=Revenue)) + geom_boxplot()
```



Revenue under the new campaign is generally higher than under the old campaign, though it is also more widely distributed. We can see that the median revenue is about $20,000 higher in the new campaign and the range between 1IQR and 3IQR is also much larger. The length of the whiskers also indicate the larger spread in the new campaign. This all makes sense, since the old revenue model was based on a mean of 100,000 and a standard deviation of 12,000, and the new revenue model was based on a mean 120,000 with standard deviation of 25,000. While revenue was only increased by 20%, the standard deviation more than doubled in the new model.

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?

Bottom line: I would not be confident that the campaign was successful because we don't know how much it cost us. We would want to know the return on investment!

If I had to only utilize visual inspection of the two provided, I would say that it was successful. We might get some months with revenue lower than the old model, but the majority of months will be higher than the majority of previous months. That is, Q1 to Q4 in the new campaign are all higher than Q3 of the old campaign.

Both the scatterplot and boxplots tell us that the new campaign has potential for massively higher revenues and less chance for revenues lower than the old model.

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.)

H_o : The marketing campaign did not increase mean revenue, $\mu_{new} \leq \mu_{old}$

H_a : The marketing campaign increased mean revenue, $\mu_{new} > \mu_{old}$

f. In a few sentences, explain why a 2-sample, 1-sided t-test is appropriate for testing the hypotheses in part e.

Two sample is appropriate because we have a smaller sample size and are comparing the means of the two samples. Here, one sided is appropriate because we are only concerned if the new mean was greater than the old mean. If we wanted to know if the new mean was simply not the same as the old mean (ie greater than OR less than) then we would want to do a 2-sided t-test.

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(d$Revenue ~ d$Campaign, alternative='greater')
```

```
Welch Two Sample t-test
```

```
data:  d$Revenue by d$Campaign
t = 3.5444, df = 14.86, p-value = 0.00149
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 11280.61      Inf
sample estimates:
mean in group new mean in group old
         121333.3          99000.0
```

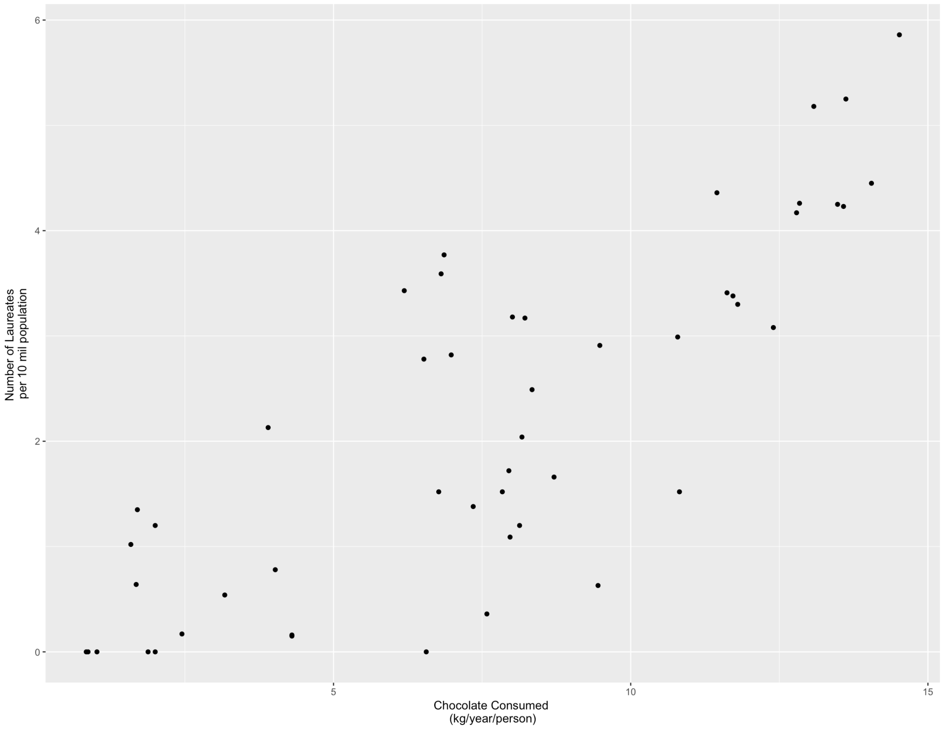
Based on a p-value of 0.00149, which is less than the significance level of 0.01, there is enough evidence to claim that the mean revenue under the new campaign is higher than the mean revenue under the old campaign.

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

   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.

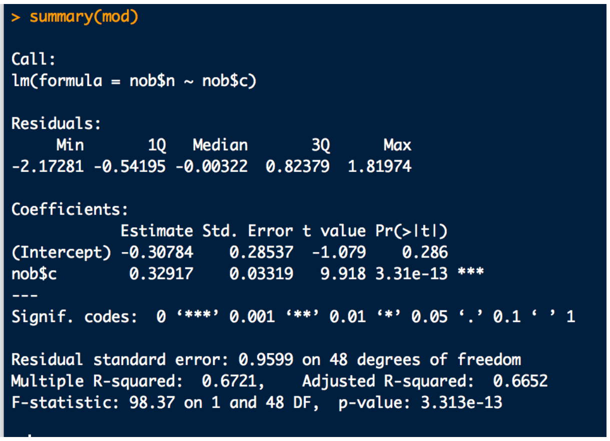
```
# read data from assignment 4
nob <- read.csv('nobel_choco.csv', header=FALSE)
# c  represents chocolate consumption in kg/year/person and
# n  represents the number of Nobel laureates per 10 million population.
names(nob) <- c('c', 'n')

# scatter plot, prize winners vs choco consumption
ggplot(nob, aes(x=c, y=n)) +
  geom_point() +
  scale_x_continuous(name='Chocolate Consumed \n(kg/year/person)') +
  scale_y_continuous(name='Number of Laureates \nper 10 mil population')
```



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.

```
# fit a linear model
mod <- lm(nob$n~nob$c)
# scatter plot, prize winners vs choco consumption
ggplot(nob, aes(x=c, y=n)) +
  geom_point() +
  scale_x_continuous(name='Chocolate Consumed \n(kg/year/person)') +
  scale_y_continuous(name='Number of Laureates \nper 10 mil population') +
  geom_abline(slope=mod$coefficients[2], intercept = mod$coefficients[1], col = 'red')
```



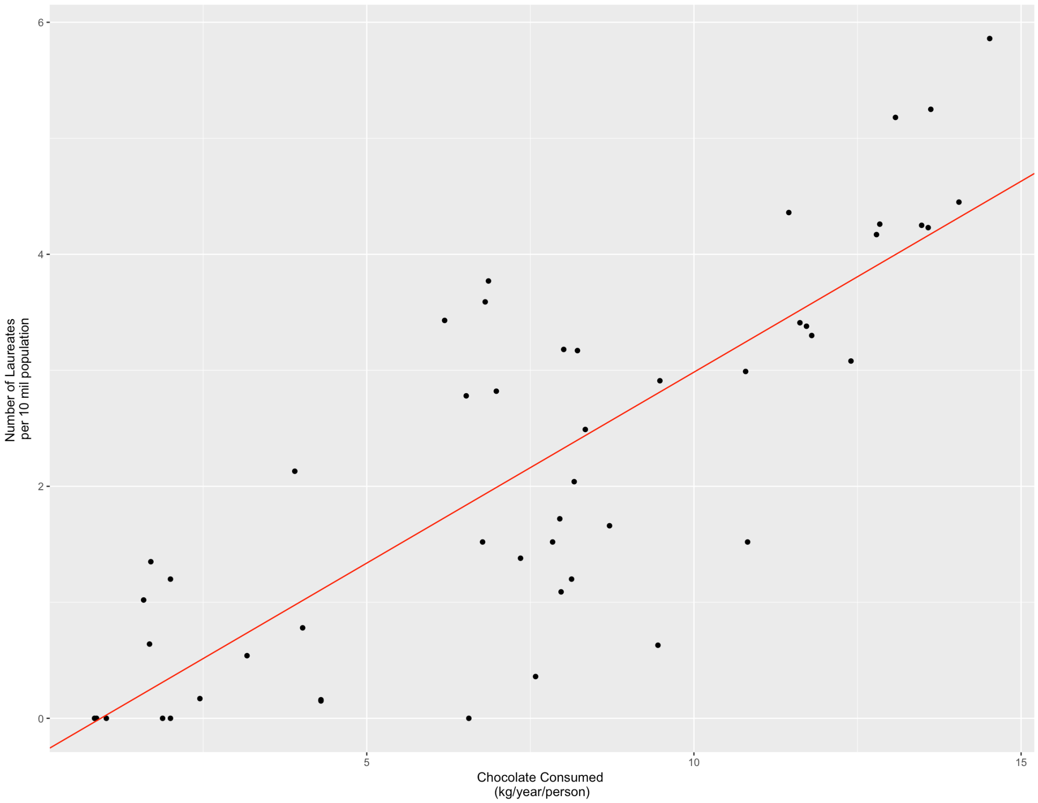
```
> summary(mod)

Call:
lm(formula = nob$n ~ nob$c)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.17281 -0.54195 -0.00322  0.82379  1.81974 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.30784    0.28537  -1.079    0.286    
nob$c        0.32917    0.03319   9.918 3.31e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9599 on 48 degrees of freedom
Multiple R-squared:  0.6721,	Adjusted R-squared:  0.6652 
F-statistic: 98.37 on 1 and 48 DF,  p-value: 3.313e-13
```



Equation of line:

laureates=0.32917*chocolate(kg/yr/perso) -0.30784

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.

H_o : The number of Nobel Prize winners (per 10 mil population) is not associated with per-capita chocolate consumption

H_a : The number of Nobel Prize winners (per 10 mil population) is associated with per-capita chocolate consumption

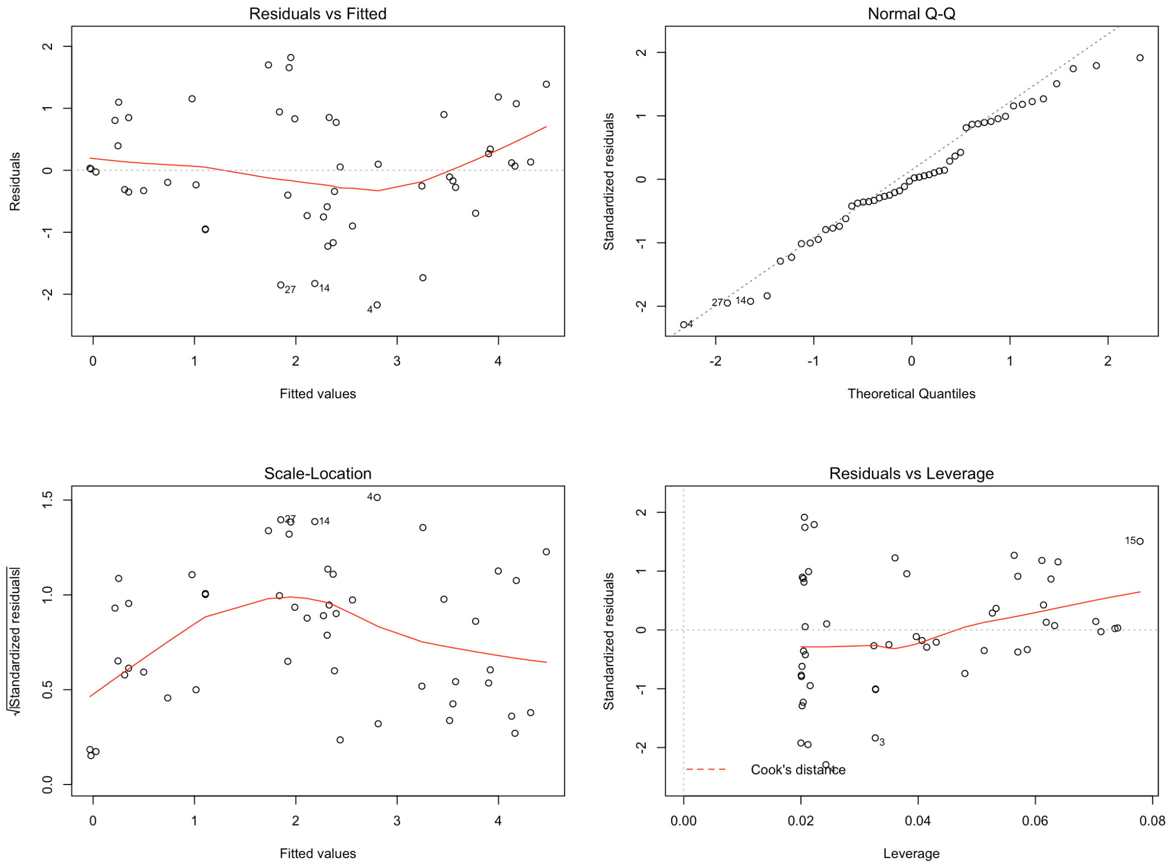
d. Use appropriate R code to test the hypotheses in part c. Include the code and output, and state your conclusion in the context of the problem.

Based on a p-value of 3.13x10^-13 (essentially zero) which is less than 0.01, in the linear model there is sufficient evidence to reject the null hypothesis.

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

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

plot(mod)
```



The upper left (residuals vs fitted) shows each points y value less the predicted y value. The x-axis are the predicted y values. We'd be hoping to have the residuals clouded around zero which would indicate that there isn't a trend in the data that we aren't accounting for. There doesn't seem to be much of a trend in this plot that would require us to transform the data or add higher degree polynomial such as x^2.

The Quantile-Quantile plot helps us understand how well the residuals fit a normal distribution, which is a rule we follow when applying linear regression. In our plot, the residuals generally follow the normal distribution, though there are a small number of points that drift away. Not a perfect fit, but it does not appear that a log-transformation would improve our results, though a polynomial might (at the risk of overfitting).

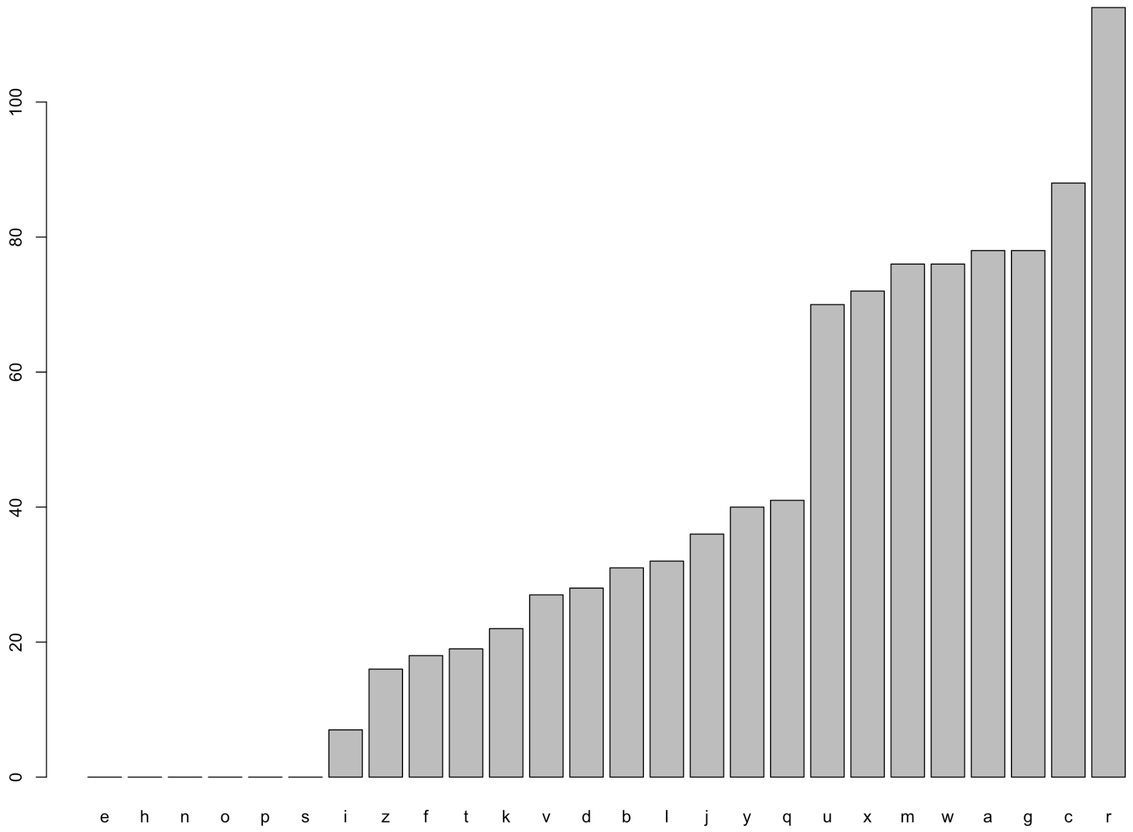
3. In homework 5, you counted the frequencies of letters in two encrypted texts. In this problem, you will use statistical analysis to identify the language in which the text was written, and decrypt it.

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.

```
encA <- read.csv("encryptedA_freq.csv", header=F)
names(encA) <- c('key', 'freq')
attach(encA)
# generate a vec with the index of the freq in ascending order
encA_order = order(freq)
# plot the data, but select it according to the order we just generated
barplot(freq[encA_order], names.arg = key[encA_order])
```



b. 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.

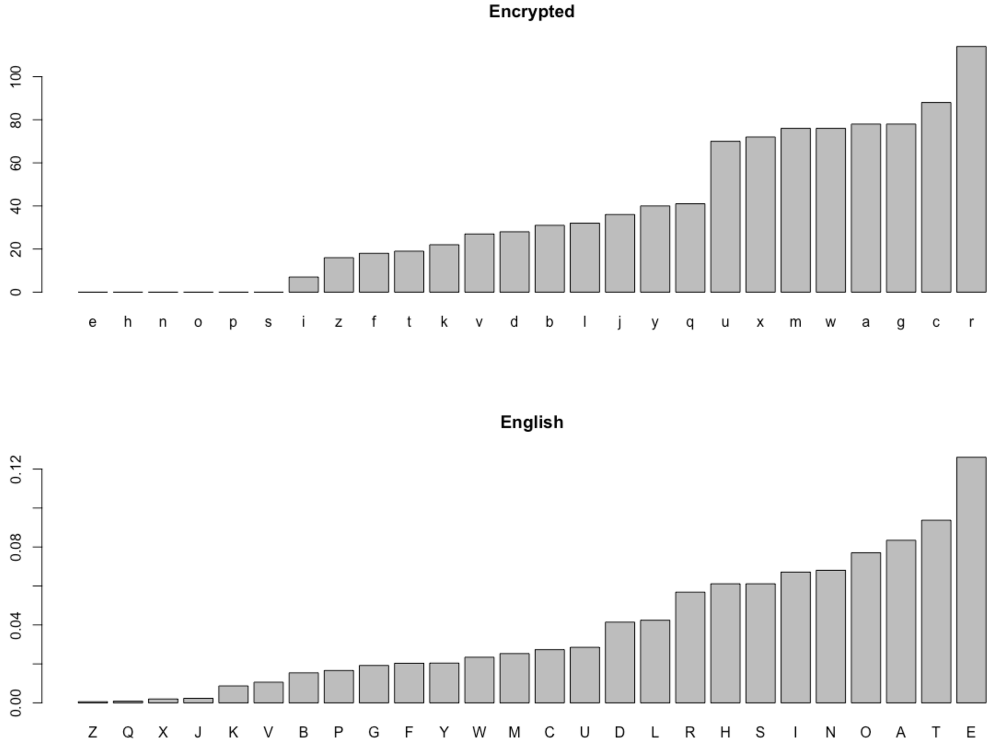
```
letterFreq <- read.csv('Letter Frequencies.csv', header=T)
```

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.

```
ltrF <- letterFreq[,c('Letter', 'English')]
```

```
names(ltrF) <-c('key', 'freq')
ltrF$type <- 'english'
encA$type <- 'encrypted'
encA <- encA[order(encA$freq),]
ltrF <- ltrF[order(ltrF$freq),]

par(mfrow=c(2,1))
barplot(encA$freq, names.arg=encA$key, main='Encrypted')
barplot(ltrF$freq, names.arg=ltrF$key, main='English')
```



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

The text is potentially English, but I am not confident. Compared to English, we see that each have 1 character than appear more often than the others, then a group of characters that are similar in frequency. There is also a steady trail off of character frequencies. What concerns me is that the encrypted text has 6 characters that do not appear in the sample, while English has only a couple with ultra low frequency.

(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.)

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.

```
encVec <- encA[order(encA$freq),'freq']

engVec <- ltrF[order(ltrF$freq),'freq']
```

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 )
```

```
length(engVec) == length(encVec) #already same length

[1] TRUE

length(engVec)

[1] 26
```

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

H_o : The distribution of frequencies for English is a good fit for the encrypted data, ie it is an English text.

H_a : The distribution of frequencies for English is not a good fit for the encrypted data, ie not an English text.

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.

```
expectedCounts<-sum(encVec)*engVec
```

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.

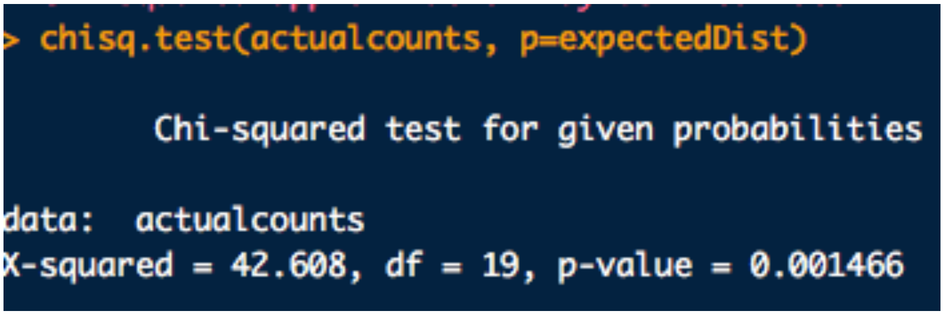
actualcounts <- c(sum(encVec[1:7]), encVec[8:26])

expectedCounts <- c(sum(expectedCounts[1:7]), expectedCounts[8:26])

expectedDist <- c(sum(engVec[1:7]), engVec[8:26])

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:
     1. You did not combine enough categories in step i, or
     2. 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.

chisq.test(actualcounts, p=expectedDist)



```
> chisq.test(actualcounts, p=expectedDist)

        Chi-squared test for given probabilities

data:  actualcounts
X-squared = 42.608, df = 19, p-value = 0.001466
```

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.

With a p-value of <0.05, there is enough evidence to reject the null hypothesis. Which means we can say that this is NOT an English text.

l. Repeat steps h-k for Welsh, and then repeat for both languages for encryptedB. Fill in the p-values you get in the following table:

Text	English	Welsh
EncryptedA	**.001466**	**0.5736**
EncryptedB	**0.5724**	**5.414e-5**

```
encB <- read.csv("encryptedB_freq.csv", header=F) # toggle B to A in file string to read other file
names(encB) <- c('key', 'freq')
encB_order = order(freq)

letterFreq <- read.csv('Letter Frequencies.csv', header=T)
ltrF <- letterFreq[,c('Letter', 'English')] # toggle English to Welsh for other translation
names(ltrF) <-c('key', 'freq')
ltrF$type <- 'english'
encA$type <- 'encrypted'

letterFreq <- read.csv('Letter Frequencies.csv', header=T)
ltrF <- letterFreq[,c('Letter', 'English')] # toggle English to Welsh for other translation
names(ltrF) <-c('key', 'freq')
ltrF$type <- 'english'
encB$type <- 'encrypted'
encVec <- encB[order(encB$freq),'freq']
engVec <- ltrF[order(ltrF$freq),'freq']

length(engVec) == length(encVec)

expectedCounts<-sum(encVec)*engVec

actualcounts <- c(sum(encVec[1:7]), encVec[8:26])
expectedCounts <- c(sum(expectedCounts[1:7]), expectedCounts[8:26])
expectedDist <- c(sum(engVec[1:7]), engVec[8:26])

chisq.test(actualcounts, p=expectedDist)
```

m. Based on the hypothesis tests, which text do you think came from which language? How confident are you in your assessment?

Encrypted A is likely Welsh, and Encrypted B is likely English if I am forced to decide. However, I have confidence that A is NOT English and B is NOT Welsh, I can't be certain that these aren't from other languages with the same alphabet. If it is certain that the only two possibilities for languages of the encrypted texts is English or Welsh, then my confidence goes to almost complete certainty.

n. 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 single .docx or .pdf file to GitHub containing your R code, R output and graphs, and your written interpretations and explanations. Include your name at the top of the file. Keep all portions of a problem together (don’t put all the R code at the end of the file).