Justin Sampson
Stat 154
Section 101
21986614

Time-Series Modeling of Sentiment Analysis

Introduction

The scientific question motivating my work is whether or not there is any seasonality in the sentiment of a company's press releases that can be used to illuminate an internal strategy for information dissemination. For this analysis I have used British Petroleum's (to be referred to as BP henceforth) press releases to gather sentiment data. The goal is to see if the quantified sentiment values are white-noise or if they display trends and can be modeled using a standard ARIMA model. Sentiments of a press release should reflect the positivity or negativity of the news being released to the media and the public. Intuitively a news event occurring that is worthy of being released to the press should be random and follow a Poisson distribution. Further, a news event being positive or negative adds an additional layer of stochasticity. Looking at periodicities and other trends could show deviations from this Poisson Process, which could illuminate an internal strategy for how a specific company manipulates the outflow of information to the public. Methods on quantifying the sentiment of each press release are based off of Bing Liu, Minqing Hu and Junsheng Cheng's paper "Opinion Observer: Analyzing and Comparing Opinions on the Web." While current research focuses on a neurological basis for semantic relatedness, here we use a lexicon of positive and negative words to tabulate an average sentiment. There is no final check for this average due to the qualitative nature of an opinion. To move forward one must infer meaning from the relative differences over the data set rather than the exact value of the sentiment.

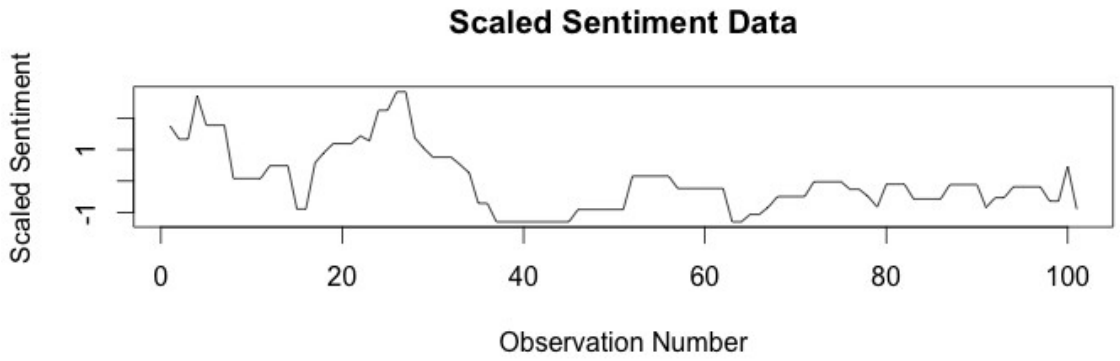
Data

The data in this analysis is scraped from the web using a self-made Python script. The first stage was compiling a list of 300 web page links from the BP Press Release interface and organizing them into a comma separated values file. These press releases span dates from November 17th, 2011 to November 17th, 2014. Each link is then run through the script where the text of the press release and the date of the press release are extracted. Natural language processing is then uses to remove punctuation from the text leaving only character-word chunks. These word chunks are then filtered against Python's Natural Language Processing ToolKit's list of stop words, commonly occurring words in the English language, and against other lists of commonly occurring words. The goal of this endeavor was keep the positive, negative, and neutral words whilst not diluting an average with words like 'or', 'and', 'because' etc. that would contribute zero to a sentiment but increase the word count (denominator) in the list. There is an attempt to match each word in the filtered press release to a word in Bing Liu, Minqing Hu and Junsheng Cheng's positive-negative lexicon. If the word is positive then a one is added to that particular press release's list, if the word is negative then a negative one is added to the list, and if the word is neither positive nor negative it is assumed to be neutral and thus a zero is added to the list. An average sentiment for a single press release by averaging the list of -1s, 1s, and 0s. This process occurs for every press release in the date range. At this stage there is a list of quantified sentiment values based on the above process with their corresponding dates of occurrence. In the date range some days have multiple press releases. This is dealt with by taking an average of the average sentiments on a day where more than one press release is released. Further some days do not have any press releases at all. This is dealt with by taking an average over ten days in the date range creating a ten day average sentiment. At this point we move on to the analysis of the

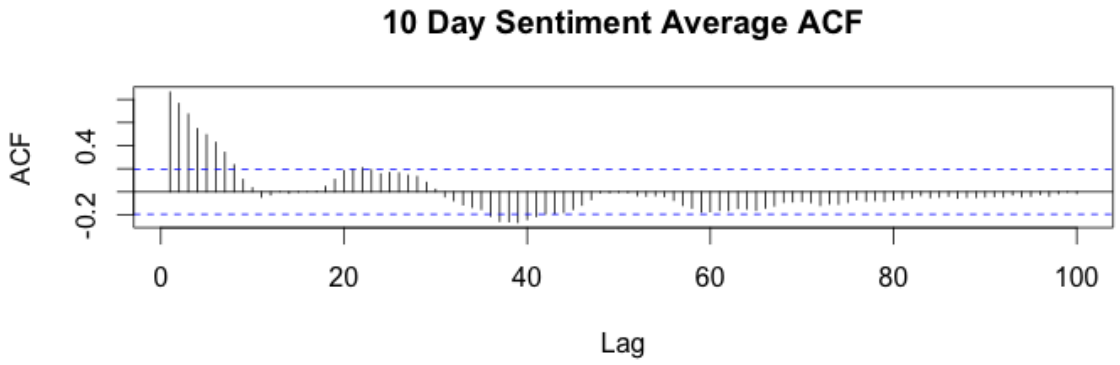
data.

Analysis: Explore

First the data is plotted to view the overall time-series.

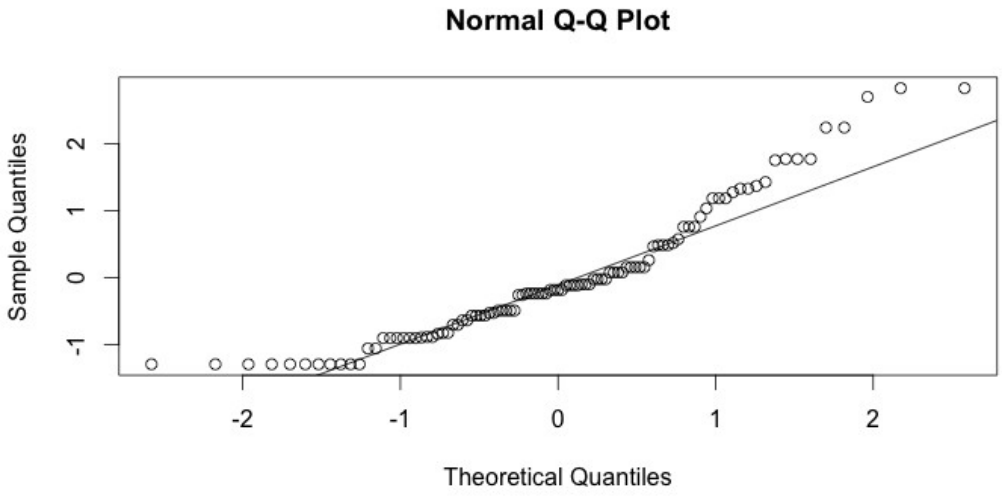


In this plot one can see a downward linear trend, perhaps a quadratic trend, and some periodicities. The next step is to plot the autocorrelation function of the data.

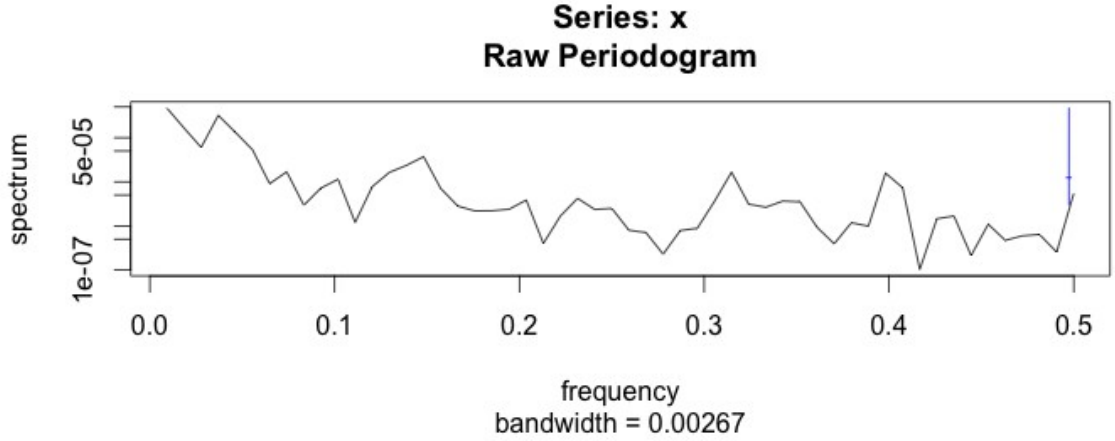


The ACF is oscillating from positive to negative over the 95-percent confidence interval while tending to zero. From this and the plot of the time-series we can infer that it is semi-stationary, but we can de-trend the data in order to make it more stationary. Using the Shapiro Test for

normality we get a p-value of 1.014e-05, meaning we can reject the null hypothesis of the data being distributed normally. We can also see this result in the following Quantile-Quantile plot.



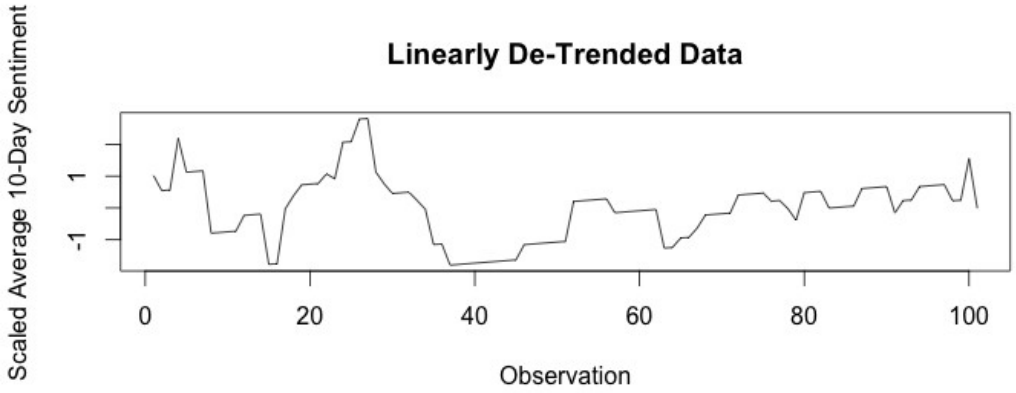
The runs test gives a p-value of 5.67e-10, that in combination with the ACF we can conclude the day is not independently distributed. When looking at the spectrum of the time-series we see clear periodicities outside of the 95-percent confidence interval, indicated by the blue line.



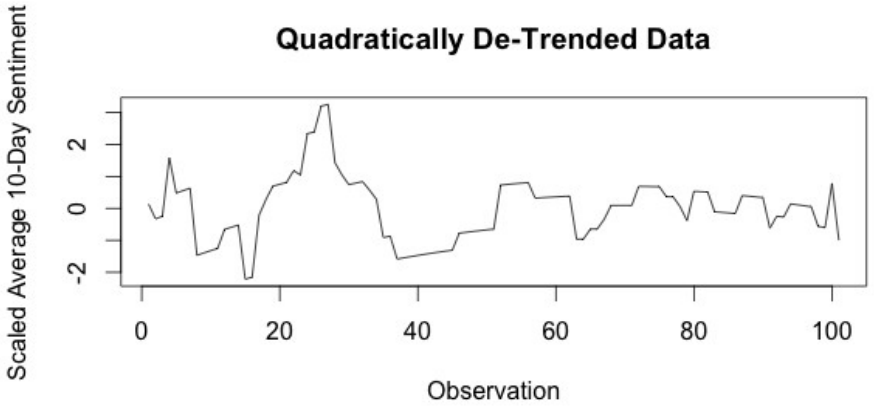
From the ACF, the Shapiro Test, the Runs test, and observing the spectrum of the data one can conclude the time-series is not white-noise.

Analysis: De-trend

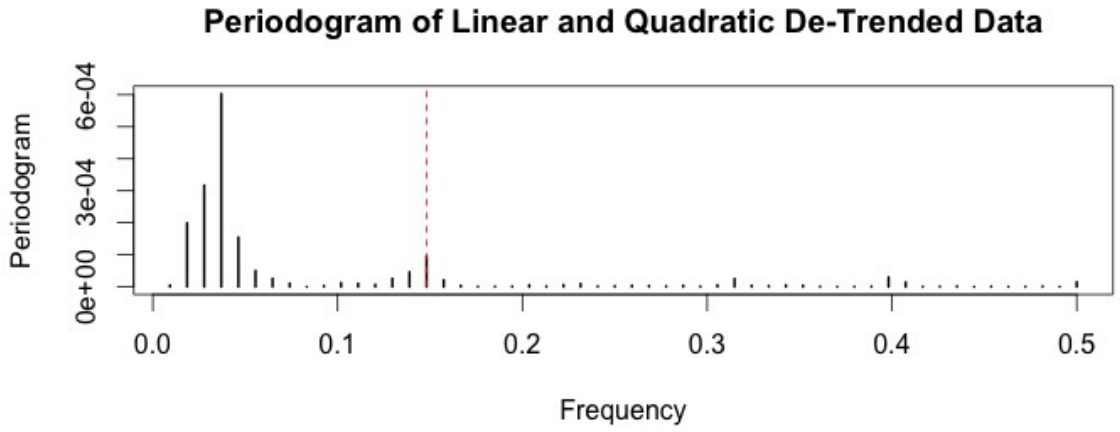
Looking at the original plot of the data a linear trend can be inferred. After linearly de-trending the time-series we observe the following plot.



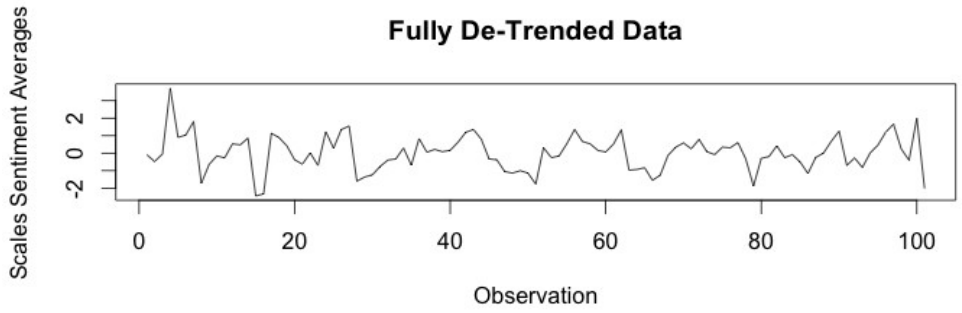
At this point the data seems to display a quadratic trend and thus we should remove it from the time-series. Upon removing the quadratic trend we observe the following plot.

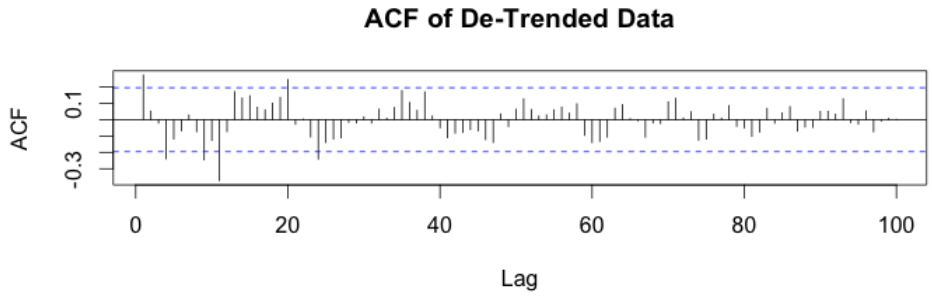


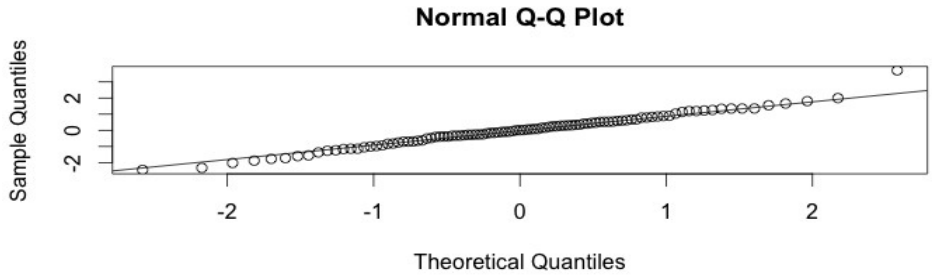
After removing the linear and quadratic trends we look to spectral analysis to remove some of the harmonic trends in the data. Looking at the periodogram several frequencies become apparent as contributing to the Fourier make-up of the series.

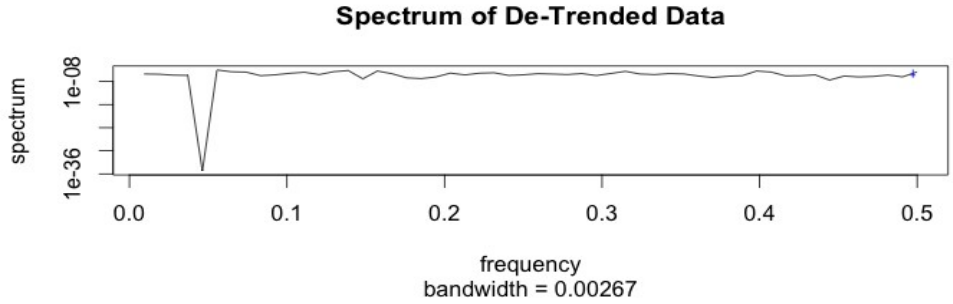


Here there is periodicity in the first few frequencies and then a final frequency of interest where the red dotted line is overlapped. This frequency corresponds to a period of about 6.8, which in the data would be 68 days since the observations are 10-day averages. This could indicate some sort of quarterly cyclical pattern in the sentiment. The lower frequencies represent trends that are over a longer period. The largest spectrum value corresponds to a frequency of 0.037, a period of 27. In this data 27 is 27 10-day averages, which would be interpreted as 270 days. This is around the end of the third-quarter of the year. One could hypothesize that BP is planning its press releases here to be of a certain type in anticipation of their business year end. By fitting values of β on to $\beta_1 \cos(2*\pi*f*t)+\beta_2*\sin(2*\pi*f*t)$ for the frequency values of interest and removing them from the time-series we obtain the following plots of the data.





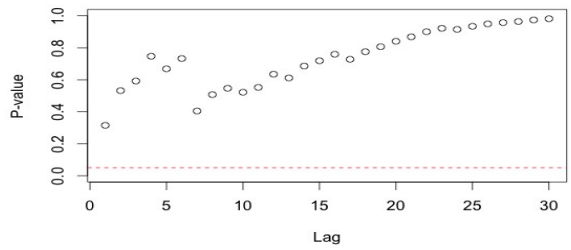




Here we can see that the data is not quite white-noise as there is still autocorrelation and a single significant frequency in the spectrum. This could be because I removed them in previous de-trending. However the data does appear to be normally distributed. We next move to the ARIMA portion of the analysis.

Analysis: ARIMA

One assumption motivating the fitting of an ARIMA model is a constant variance. This is validated through the plot of the McLeod.Li.test.

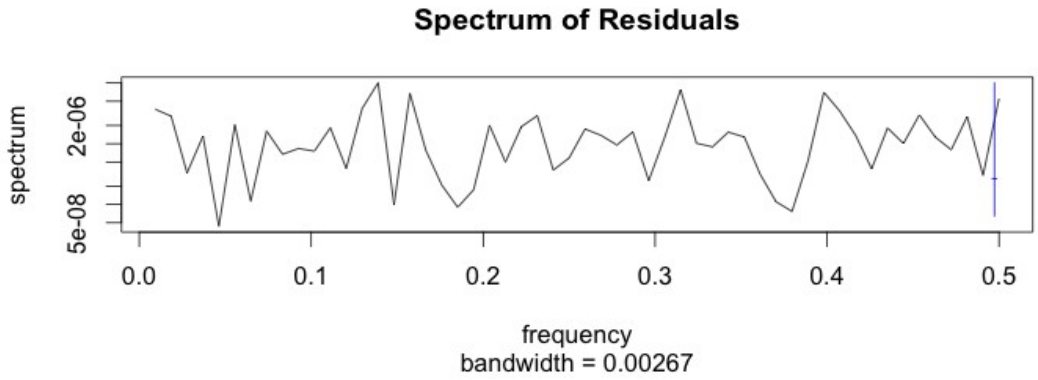


We can conclude that there is not strong evidence of autoregressive conditional heteroskedasticity, and that an ARMA model is appropriate for the de-trended data based on previous plots. Through the maximum-likelihood method we fit an ARMA(5,5), or an ARIMA(5,0,5), model to the de-trended data. The ARMA(5,5) model was selected based on the Akaike information criterion and produced a score of -1015.039. This takes the form

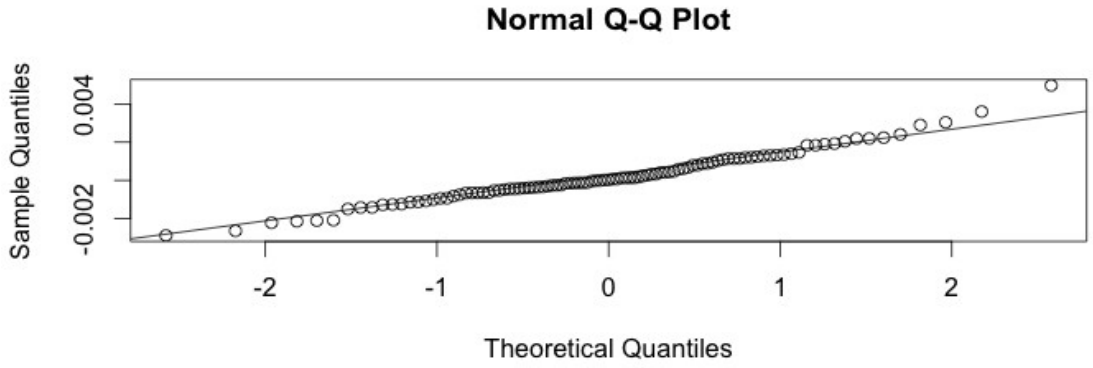
$$Y_t = c + \epsilon_t + \sum_{i=1}^{5} (\rho_i * X_n) + \sum_{i=1}^{5} \theta_i * \epsilon_n \; where \; n = t - i$$ with the following coefficients in row two and standard errors in row three.

	AR1	AR2	AR3	AR4	AR5	MA1	MA2	MA3	MA4	MA5	Int.
Coef	-0.71	1.32	0.91	-0.73	-0.6	1.05	-1.61	-1.83	0.61	0.77	0
S.E	0.09	0.09	0.14	0.06	0.09	0.13	0.16	0.19	0.13	0.12	--

Observing the following plots and out puts we can confirm the assumptions of the model that the residuals are white-noise.



Here we see that the spectrum of the residuals are all within a 95% confidence interval helping to substantiate the claim that the models residuals are actually white-noise.



Looking at the normal q-q plot we can see that the residuals are approximates normal. This combined with a p-value of 0.071 on the Shapiro test, we cannot reject the test's null hypothesis that the model's residuals are distributed normally. Finally we perform the Ljung-Box test on the residuals and get a p-value of 0.916 and cannot reject the test's null hypothesis that the errors are uncorrelated.

Conclusion

There were several issues to be dealt with in the construction of this analysis. The first issue was

dealing with unevenly spaced data. Ideally there would have been one press release a day or one press release per week. However on some days their were multiple press releases while on other days there were no press releases at all. In future work one could model a stochastic point-process and model the subsequent time-series using that additional random variable. In this analysis however the data was made equi-spaced through averaging multiple daily releases into a single day average, and then averaged again into an average sentiment over ten days. In the data there seemed to be a downward trend in the sentiments over the last several years with a slight quadratic trend. Additionally there were a few frequencies of interest that seemed to appear in the data, one of which seemed to have a period of 6.8 or 68 days since the data points were averaged over ten days. This could correspond to some sort of quarterly release or other constant release to the public. There was also a frequency corresponding to around the third quarter of the business year. After de-trending the data, based on statistical evaluation methods mentioned above, an ARMA(5,5) model seemed to fit the data the best leaving white-noise residuals. To the question of BP's strategy in public communication involves five 10-day sentiment averages and error terms yielding a fifty day backward scope, meaning that it is feasible that a current press release could be generated based on releases over a fifty window in the past. As sentiment analysis becomes more of an exact science and more computationally intensive prediction techniques, such as machine learning algorithms, are incorporated into time-series analysis a more exact pattern may emerge. Further with analysis on stock movements with an exact sentiment of released information could yield optimal strategies for information dissemination rather than identifying current strategies.

References

Bing Liu, Minqing Hu and Junsheng Cheng. "Opinion Observer: Analyzing and Comparing Opinions on the Web." Proceedings of the 14th International World Wide Web conference (WWW-2005), May 10-14, 2005, Chiba, Japan.

Jonathan Cryer and Kung-Sik Chan. Time-Series Analysis with Applications in R. January 2008.

Python Code:

```
from bs4 import BeautifulSoup
import webbrowser
from urllib import urlopen
import nltk
import string
from string import digits
import numpy as np
import csv
import socket
from nltk.corpus import stopwords

socket.setdefaulttimeout(None)

dump = '/Users/justinsampson/testing/statdata.csv'
url_file = '/Users/justinsampson/testing/Sampson.csv'


bp_words = ('bp', 'bps', 'one', 'two','three', 'four', 'five', 'six', 'seven', 'eight', 'nine','ten', 'eor', 'per', 'cent', 'pls', 'plc', 'of', 'are', 'both', 'a')
common_words = stopwords.words("english")
other_words =
['a','able','about','across','after','all','almost','also','am','among','an','and','any','are','as','at','be','because','been','but','by','can','cannot','could','dear','did','do','does','either','else','ever','every','for','from','ge
t','got','had','has','have','he','her','hers','him','his','how','however','i','if','in','into','is','it','its','just','least','let','like','likely','may','me','might','most','must','my','neither','no','nor','not','of','off','often','on','onl
y','or','other','our','own','rather','said','say','says','she','should','since','so','some','than','that','the','their','them','then','there','these','they','this','tis','to','too','twas','us','wants','was','we','were','what','when','
where','which','while','who','whom','why','will','with','would','yet']


def get_words(url):
        response = urlopen(url)
        html = response.read()
        get_words.soup = BeautifulSoup(html)

        text = get_words.soup.find('div',{'class':'nvc-press-summary'}).text
        text = text.split(' ')
        text = [line.rsplit('\n') for line in text]
        data = []
        for i in range(0,len(text)):
                data = data + text[i]


        data = [word.encode('ascii', 'ignore') for word in data]
        data = [s.translate(None, string.punctuation) for s in data]
        data = [x.lower() for x in data]

        clean = [w for w in data if not w in common_words]
        clean = [w for w in clean if not w in bp_words]
        clean = [w for w in clean if not w in other_words]
        clean = [w for w in clean if len(w) > 2]
        clean = [w for w in clean if not w.isdigit()]
```

```
		return clean

def get_date():
		date = get_words.soup.strong.string
		date = [date.encode('ascii', 'ignore')]
		date = str(date).replace('[','').replace(']','').replace("'", '')

		return date


## get sentiment

pos = open('positive-words.txt', 'r')
pos_words = [line.rstrip('\n') for line in pos.readlines()]

neg = open('negative-words.txt', 'r')
neg_words = [line.rstrip('\n') for line in neg.readlines()]

def get_sentiment(words):

		sent = []
		final = []
		for word in words:
			if word in pos_words:
				sent.append(1)
			if word in neg_words:
				sent.append(-1)
			else:
				sent.append(0)

		return np.mean(sent)
##################################

temp= open(url_file, 'rU')
urls = csv.reader(temp)

df = []
for row in urls:
		df.append([get_sentiment(get_words(row[0])),get_date()])




myfile = open(dump, 'wb')
wr = csv.writer(myfile, quoting=csv.QUOTE_ALL)
for i in range(0,len(df)):
		wr.writerow(df[i])
```

R-Code

```
library('TSA')
library(forecast)
library(date)
library(cts)

# Question: What kind of does strategy does BP use in its press releases, and can we take advantage of this through forecast/prediction?
############################## load the data
statdata = statdata[-6,]
statdata= statdata[-which(statdata == 0),]
dates = as.date(as.character(statdata[,2]), order = 'dmy')
unique_dates = unique(dates)
unique_dates = as.date(unique_dates, order = 'dmy')
date_range = as.numeric(unique_dates[length(unique_dates)]):as.numeric(unique_dates[1])
date_range = as.date(date_range)



statdata = statdata[,1]

sentiments = c()
for(i in 1:length(unique_dates)){
  if (length(which(dates == unique_dates[i])) == 1){
    sentiments = append(sentiments,statdata[i])
  }
  else{
    index = which(dates == unique_dates[i])
    average = mean(statdata[index])
    sentiments = append(sentiments,average)
  }
}

non_distorter = mean(sentiments)
full_sent = c()
for(i in 1:length(date_range)){
  index = match(date_range[i], unique_dates)
  if (is.na(index)){
    full_sent = append(full_sent, 0)
  }
  else{
```

```
   full_sent = append(full_sent, sentiments[index])
  }
}

day = mean(full_sent[1091:1097])
day10 = c()
for (i in 1:100){
  day10 = append(day10, mean(full_sent[i:(i+10)]))
}

day10 = append(day10, day)
plot(scale(day10), type= 'l', main = 'Scaled Sentiment Data', xlab = 'Observation Number', ylab = 'Scaled Sentiment')
abline(h=0)
acf(day10, lag.max=100, main = '10 Day Sentiment Average ACF')

shapiro.test(day10) #not normal
qqnorm(scale(day10)); qqline(scale(day10))
runs(day10)
spec(day10)
periodogram(day10)
w =0.148148148
lines(c(w,w),c(0,1000),col="red",lty=2)
############################## model fitting
linfit = lm(day10~t)
plot(day10, type ='l')
lines(linfit$fitted.values)
detrend = day10-linfit$fitted.values
plot(scale(detrend), type = 'l', xlab = 'Observation', ylab = 'Scaled Average 10-Day Sentiment', main = 'Linearly De-Trended Data')

quadfit = lm(detrend~t+I(t**2))
lines(quadfit$fitted.values)
noise = detrend-quadfit$fitted.values
plot(scale(noise), type = 'l', xlab = 'Observation', ylab = 'Scaled Average 10-Day Sentiment', main = 'Quadratically De-Trended Data')
periodogram(noise, main = 'Periodogram of Linear and Quadratic De-Trended Data')

harmonicfit = lm(noise~cos(2*pi*w*t)+sin(2*pi*w*t))
plot(harmonicfit$fitted.values, type = 'l')
white = noise - harmonicfit$fitted.values

harmonic2 = lm(white~cos(2*pi*periodogram(white)$freq[2]*t)+sin(2*pi*periodogram(white)$freq[2]*t))
white = white - harmonic2$fitted.values
periodogram(white)

harmonic3 =lm(white~cos(2*pi*periodogram(white)$freq[3]*t)+sin(2*pi*periodogram(white)$freq[3]*t))
white = white - harmonic3$fitted.values
periodogram(white)

harmonic4 = lm(white~cos(2*pi*periodogram(white)$freq[4]*t)+sin(2*pi*periodogram(white)$freq[4]*t))
white = white - harmonic4$fitted.values
periodogram(white)

harmonic5 = lm(white~cos(2*pi*periodogram(white)$freq[5]*t)+sin(2*pi*periodogram(white)$freq[5]*t))
white = white - harmonic5$fitted.values
periodogram(white)

plot(scale(white), type= 'l', main = 'Fully De-Trended Data', xlab = 'Observation', ylab = 'Scales Sentiment Averages')

qqnorm(scale(white)); qqline(scale(white))
periodogram(white)
spec(white, main = 'Spectrum of De-Trended Data')
acf(white, lag.max = 100, main = 'ACF of De-Trended Data')

shapiro.test(white)#cannot reject normality
##not really anything useful
Box.test(white,type='Ljung-Box')#not even close to independent#reject independence
runs(white) #reject null of independence
########################################################
pvs = c()
wilks = c()
aics = c()
for(i in 0:7){
  for (j in 0:7){
                  for(d in 0:3){
    modfit = arima(white, order = c(i, d, j), method = 'ML') ## might be legite
    box = Box.test(modfit$residuals, type = 'Ljung-Box')$p.value
    p = shapiro.test(modfit$residuals)$p.value
    pvs = append(pvs, box)
    wilks = append(wilks,p)
    aics = append(aics,modfit$aic)
  }
}
}
min(aics[which(pvs>0.05 & wilks > 0.05)])
modfit = arima(white, order = c(5,0,5), method = 'ML')#-994.1778
summary(modfit)$coeff
modfit$aic
names(summary(modfit))
modfit$coef
modfit$aic

acf(modfit$residuals, lag.max = 100)
runs(modfit$residuals)$pvalue #independent
spec(modfit$residuals, main = 'Spectrum of Residuals')
```

```
qqnorm(modfit$residuals);qqline(modfit$residuals)
hist(modfit$residuals)
shapiro.test(modfit$residuals)$p.value #cannot reject null that residuals are normal
Box.test(modfit$residuals, type = 'Ljung-Box')$p.value #no evidence to reject null that errors are uncorrelated
runs(modfit$residuals) #independent
```