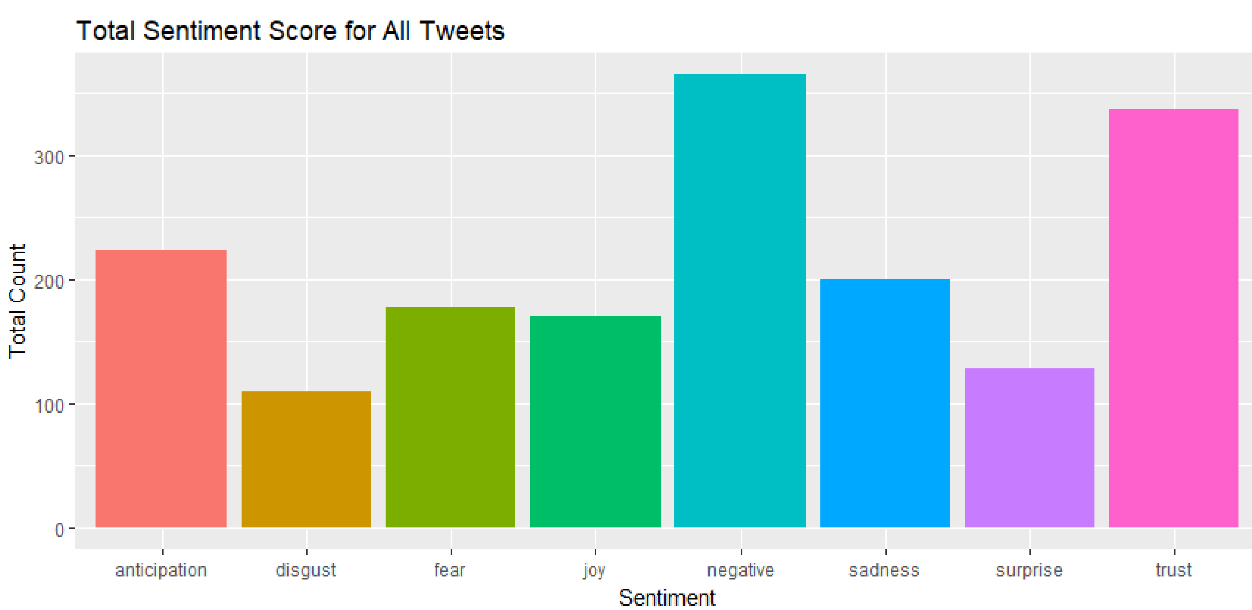
SENTIMENT ANALYSIS TO PREDICT THE MOOD OF MARKETS FOR ELECTION 2020

INDRANI MAZUMDAR


```
>consumer_key <- 'cJtGtqAedMvcD80nvfTHElvXD'
>consumer_secret <- 'b45Xz8tlCcwUHS3i6T7HlZK54S727BGsHucR8EJediYLfvncO9'
>access_token <- '937050894427918336-IkGkAXmq35p0Tn3GHge0i6YNGHOqg0l'
>access_secret <- 'X968TdvL5hN2YoQOj233p7vrPHSHxfPMR1MWmYwlqaG6D'
>setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)  #Extract data from Titter
>Markettweets<-userTimeline("realdonaldtrump", "wsj", n=3200)  #Twitter handles used @RealDonalTrump and @WSJ
>Markettweets<-twListToDF(Markettweets)                                   #Convert twitter list into data frames
>Markettweets$created <- ymd_hms(Markettweets$created)       #storing date and time of the day values
>Markettweets$created <- with_tz(Markettweets$created, "America/New_York")  #date’s time zone attributes
>Markettweets$clean_text <- str_replace_all(Markettweets$text, "@\\w+", "")
> Sentiment <- get_nrc_sentiment(Markettweets$clean_text)  #loads texts and calculates presence of 8 emotions
> Markettweets_senti <- cbind(Markettweets, Sentiment)      #Combine by rows and columns
> sentimentTotals <- data.frame(colSums(Markettweets_senti[,c(19:26)]))
> names(sentimentTotals) <- "count"
> sentimentTotals <- cbind("sentiment" = rownames(sentimentTotals), sentimentTotals)
> rownames(sentimentTotals) <- NULL
> ggplot(data = sentimentTotals, aes(x = sentiment, y = count)) +    #powerful graphics to create plot
+ geom_bar(aes(fill = sentiment), stat = "identity") +
+ theme(legend.position = "none") +
+ xlab("Sentiment") + ylab("Total Count") + ggtitle("Total Sentiment Score for All Tweets")
```

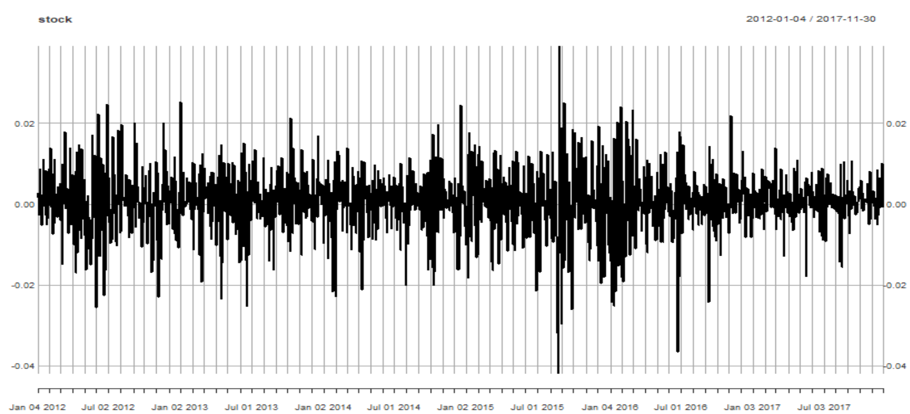


Here, in this analysis I have tried to show the mood of the market in 2020, i.e., the emotional tone behind the words and understanding the attitude, opinions and emotions expressed. I have used “Twitter” as the source and “RealDonalTrump” and “WSJ” twitter handles to gauge the public opinion on markets. According to this analysis, we can see that the negative sentiment is dominant.

FORECATING S&P500 RETURNS USING HISTORIC DATA

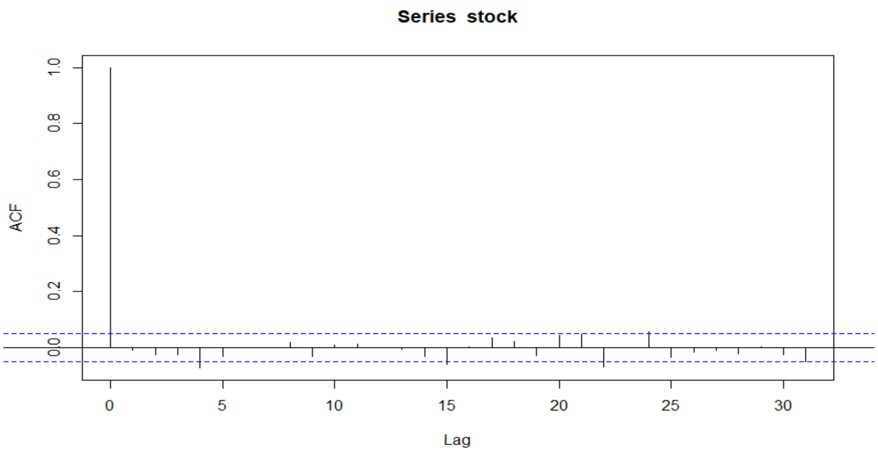
For this section, I have used SPDR S&P500 etf and collected the data from Yahoo Finance

```
> getSymbols('SPY', from='2012-01-01', to='2017-12-01')
[1] "SPY"                                   #Extracted data from Yahoo Finance
> Adjclosing_prices = SPY[,6]
> stock = diff(log(Adjclosing_prices),lag=1)  # Calculated log returns
> stock = stock[!is.na(stock)]
> plot(stock)
```

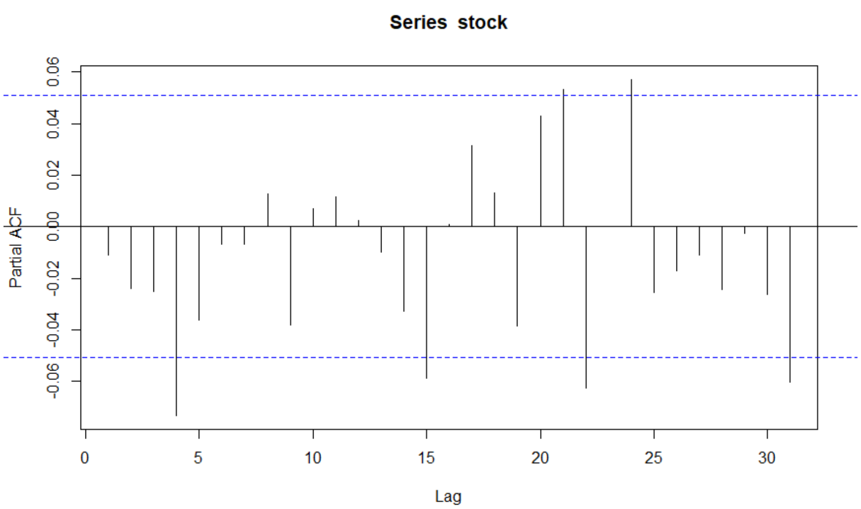



```
> adfTest((stock))                # Performed ADF test

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETERS
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -27.8071
  P VALUE:
    0.01
```

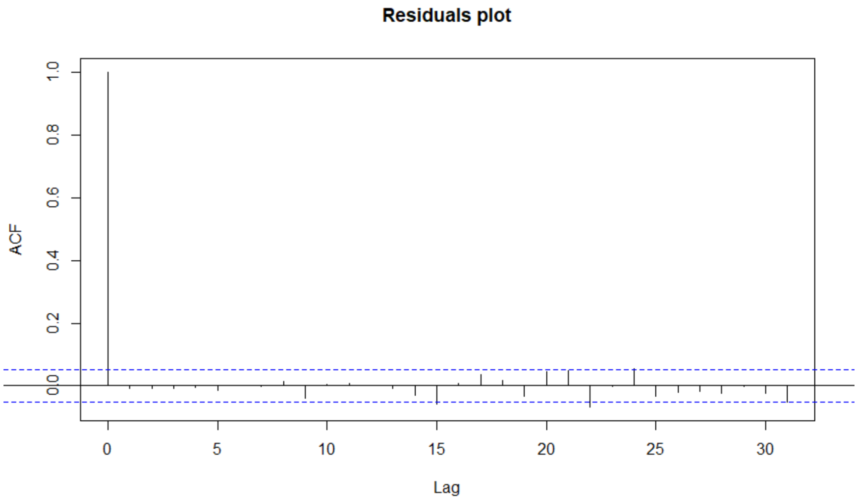
```
> acf(stock)          #Applied ACF and PACF tests
```




```
> pacf(stock)
```

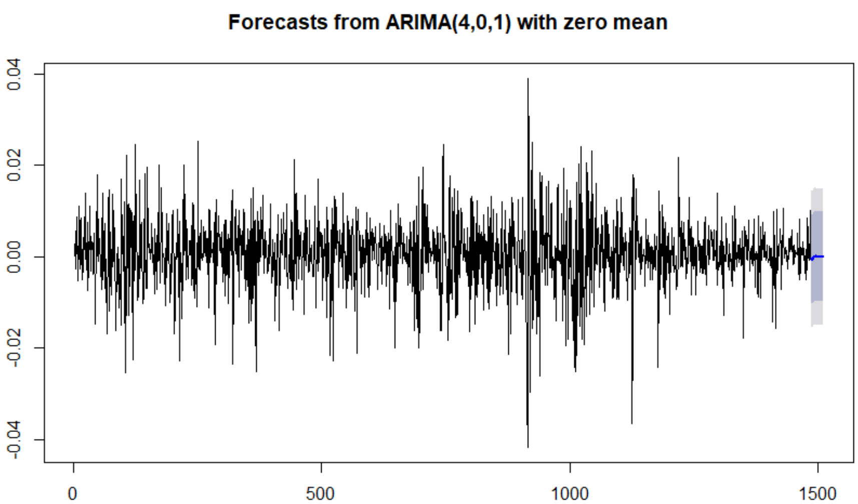



```
> fit = arima(stock, order = c(4, 0, 1),include.mean=FALSE) # Created ARIMA model
> acf(fit$residuals,main="Residuals plot")                  #plot the acf of residuals
```



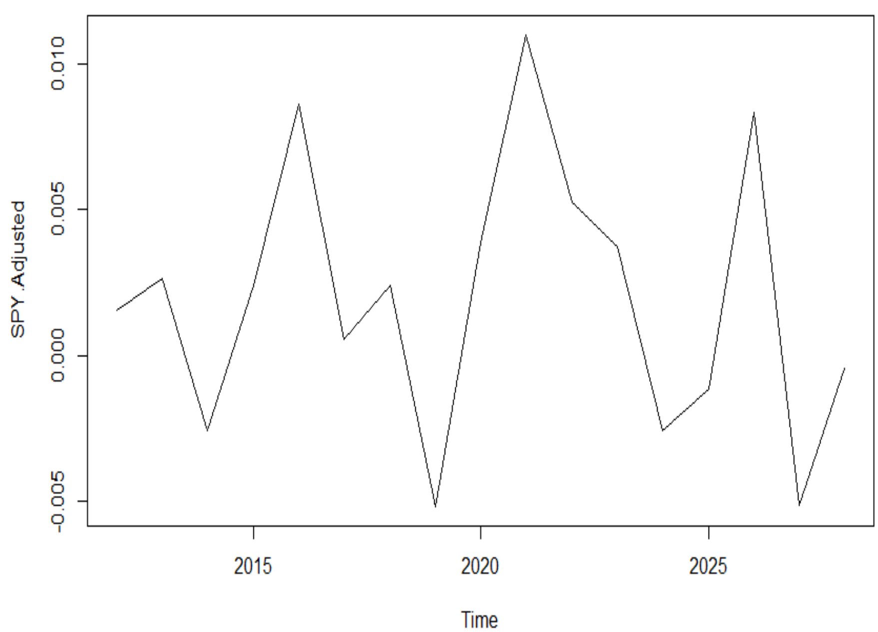

```
> summary(fit)

Call:
arima(x = stock, order = c(4, 0, 1), include.mean = FALSE)

Coefficients:
         ar1      ar2      ar3      ar4      ma1
      0.3385  -0.0177  -0.0130  -0.0607  -0.3469
s.e.  0.2745   0.0274   0.0278   0.0291   0.2746

sigma^2 estimated as 5.775e-05:  log likelihood = 5149.55,  aic = -10287.1
```

```
> p= forecast(fit,25)          # Forecasting of model
> plot(p)
```




```
> spy<-ts(stock, start=c(2012,1), end=c(2017,12), frequency=1)  #Time Series of Return
> plot(spy)
```


```
> forecastspy<-ts(forecast(stock), start=c(2012,1), end=c(2017,12), frequency=1)
>plot(forecastspy) #forecast of returns
```



Here, we can have received the forecast of S&P 500 returns (SPY ETF). The plots obtained tells that a dip is expected in 2018, the stock market will see a dip. In 2019, the stock market is likely to see a sharp decline and will rebound and start recovering and will follow an upward trend in 2020.