Group Project Report

Group D8

3/16/2020

```
library(forecast)
library(ggfortify)
library(ggplot2)
library(quantmod)
options('getSymbols.warning4.0' = F)
```

January 2018 - December 2019 (Daily)

Using data from Yahoo! Finance

```
getSymbols(Symbols = '^GSPC',
           src     = 'yahoo',
           auto.assign = T,
           from    = '2018-01-01',
           to      = '2019-12-31')
```

```
## [1] "^GSPC"
```

```
sp500 <- GSPC[, 'GSPC.Close']
sp500 %>% str
```

```
## An 'xts' object on 2018-01-02/2019-12-30 containing:
##   Data: num [1:502, 1] 2696 2713 2724 2743 2748 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr "GSPC.Close"
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
## List of 2
##  $ src    : chr "yahoo"
##  $ updated: POSIXct[1:1], format: "2020-03-31 18:48:01"
```

```
sp500 %>% head
```

```
##            GSPC.Close
## 2018-01-02    2695.81
## 2018-01-03    2713.06
## 2018-01-04    2723.99
## 2018-01-05    2743.15
## 2018-01-08    2747.71
## 2018-01-09    2751.29
```

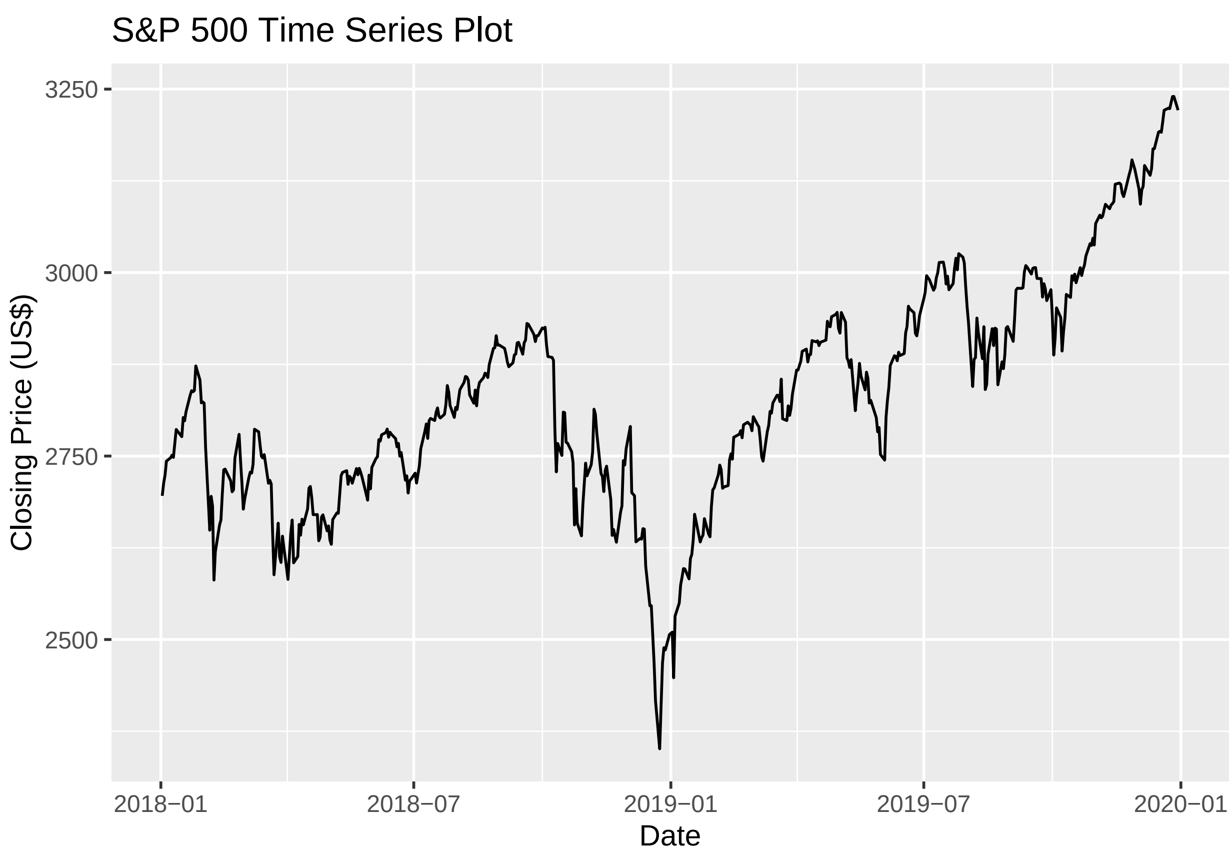
```
sp500 %>% tail
```

```
##            GSPC.Close
```

```
## 2019-12-20    3221.22
## 2019-12-23    3224.01
## 2019-12-24    3223.38
## 2019-12-26    3239.91
## 2019-12-27    3240.02
## 2019-12-30    3221.29
```

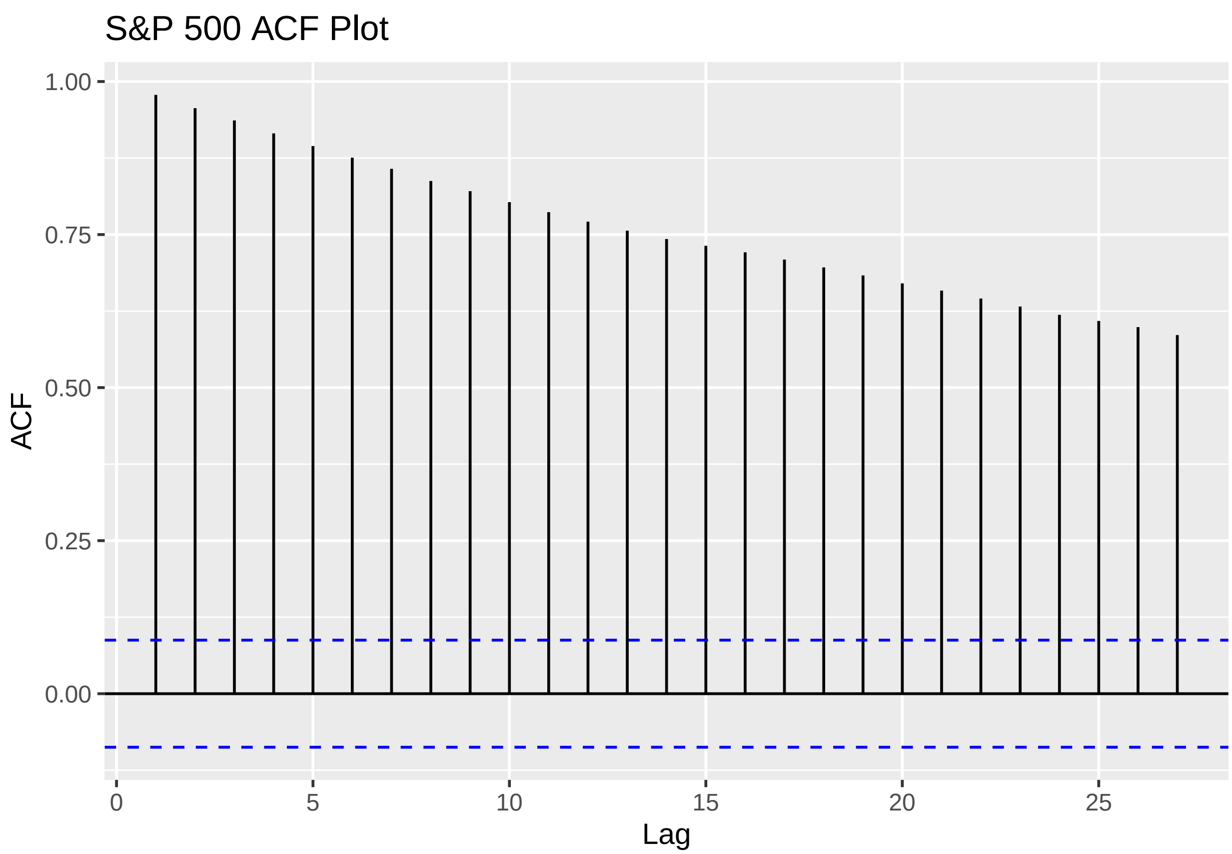
Time Series Plot

```
sp500 %>% autoplot +
  xlab(label = 'Date') +
  ylab(label = 'Closing Price (US$)') +
  ggtitle(label = 'S&P 500 Time Series Plot')
```



ACF Plot

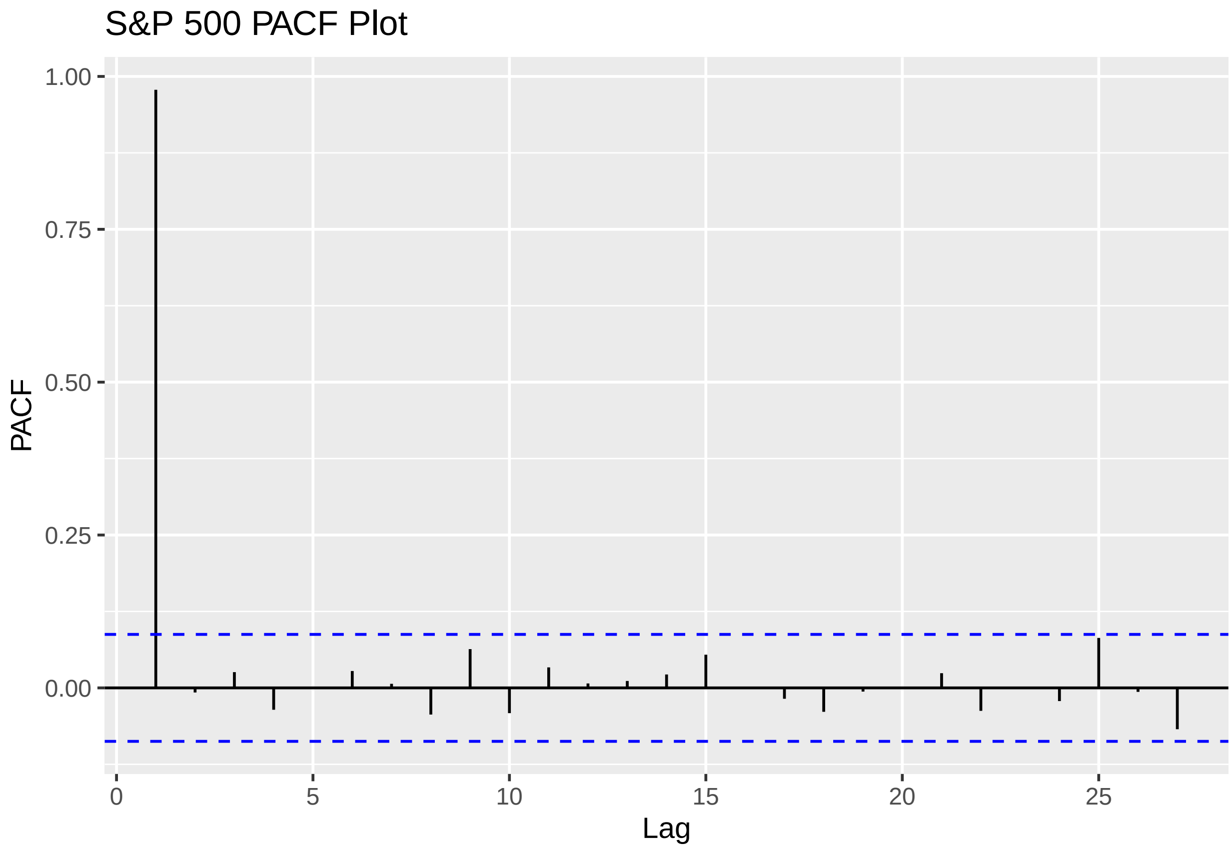
```
sp500 %>% ggAcf +
  ggtitle(label = 'S&P 500 ACF Plot')
```



The ACF values seem to be slowly decaying in time.

PACF Plot

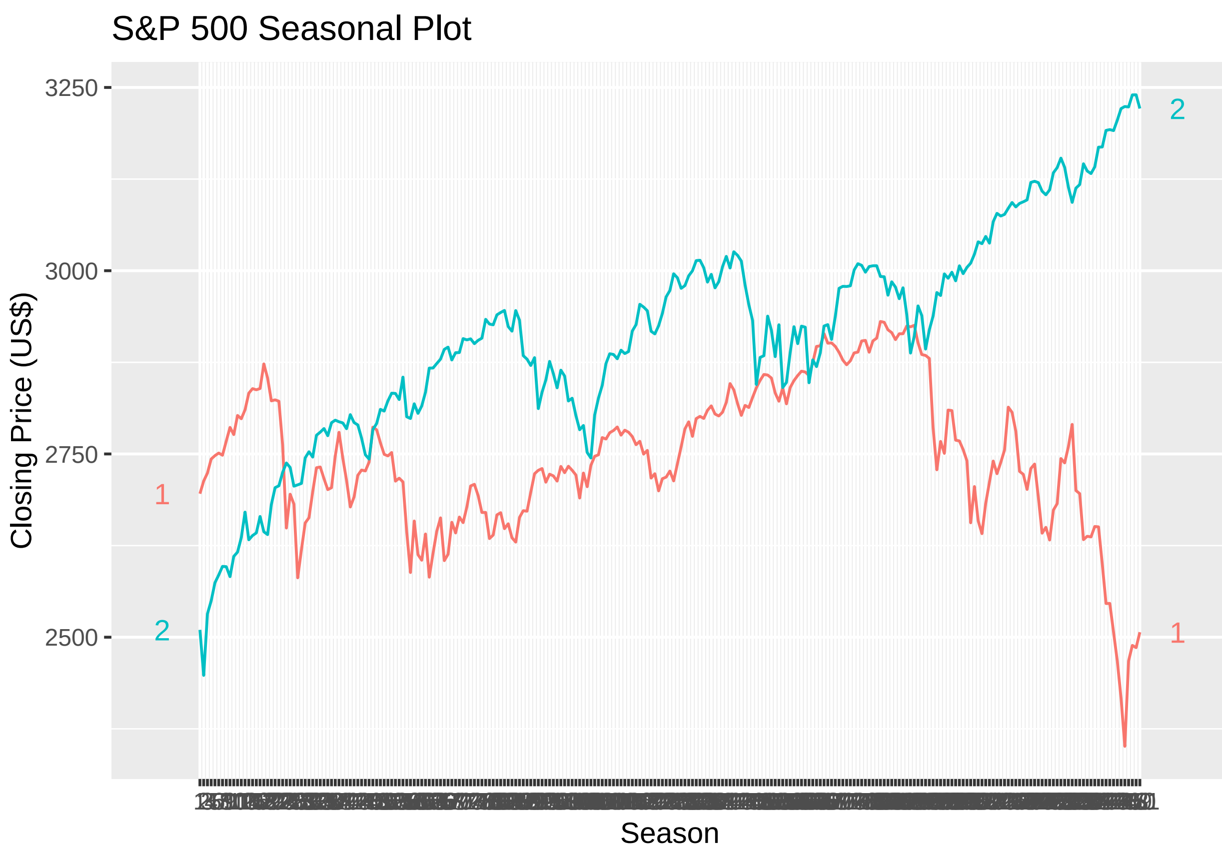
```
sp500 %>% ggPacf +
  ggtitle(label = 'S&P 500 PACF Plot')
```



PACF plot shows a significant value at lag 1 suggesting an AR(1) Model.

Seasonal Plot

```
sp500 %>%
  ts(frequency = 251) %>%
  ggseasonplot(year.labels = T,
               year.labels.left=T) +
  ylab('Closing Price (US$)') +
  ggtitle('S&P 500 Seasonal Plot')
```



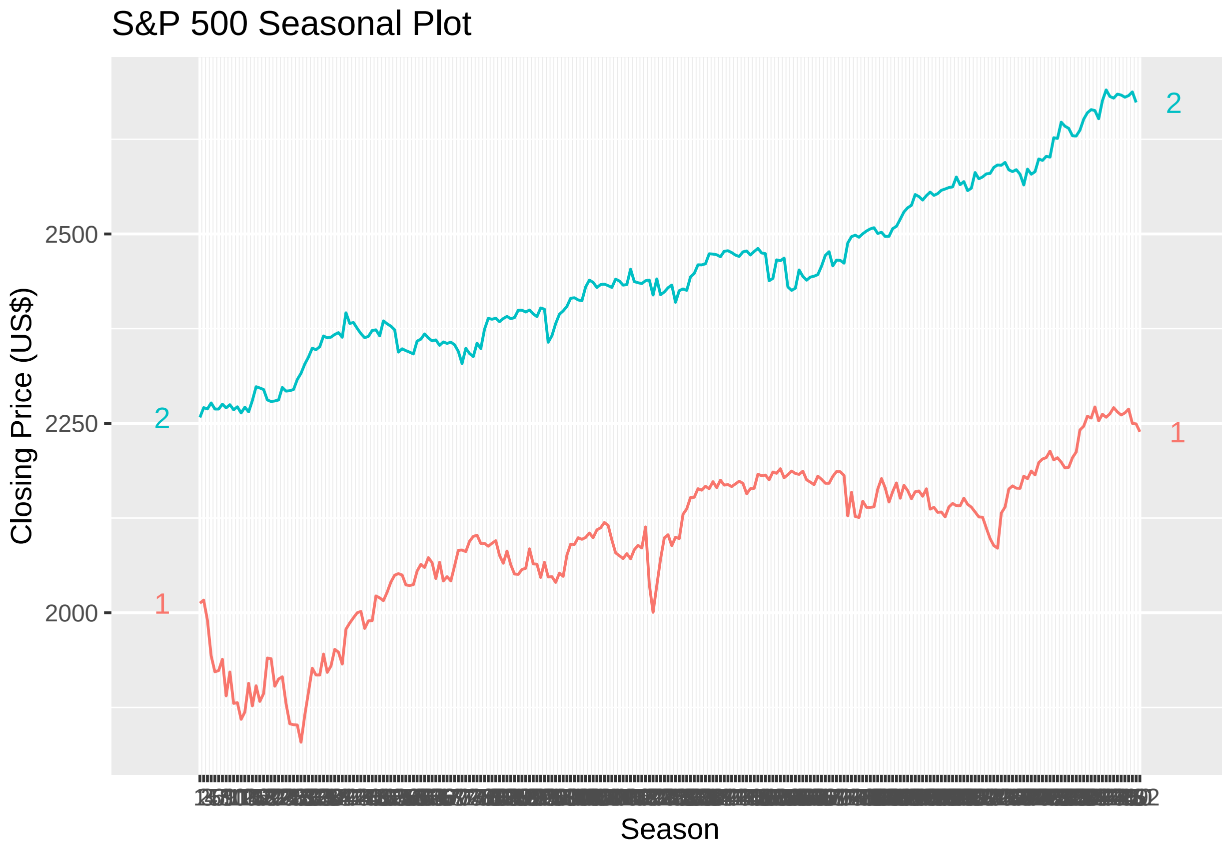
The downward trend in year 2018 is the Global Stock Market Downturn which happened on 20 Sep 2018 https://en.wikipedia.org/wiki/List_of_stock_market_crashes_and_bear_markets

Alternative Seasonal Plot (2016 - 2017 to avoid market crashes)

```
getSymbols(Symbols = '^GSPC',
           src     = 'yahoo',
           auto.assign = T,
           from    = '2016-01-01',
           to      = '2017-12-31')
```

```
## [1] "^GSPC"
```

```
sp500_2 <- GSPC[, 'GSPC.Close']
sp500_2 %>%
  ts(frequency = 252) %>%
  ggseasonplot(year.labels = T,
               year.labels.left=T) +
  ylab('Closing Price (US$)') +
  ggtitle('S&P 500 Seasonal Plot')
```




```r
# TODO: Refactor this! (for testing)
sp500 <- sp500_2
```

If we use 2016 - 2017, we would get a better daily trend

Test for Stationarity

```r
Box.test(sp500, lag = 25, type = 'Ljung-Box')
```

```
##
##  Box-Ljung test
##
## data:  sp500
## X-squared = 10364, df = 25, p-value < 2.2e-16
```

We can see that the p-value is very small (< 0.05) so we have sufficient evidence to reject the null hypothesis that the process is stationary.

Transform Data by taking first differences

```r
sp500_diff <- diff(x = sp500)[-1]
```

Test for Stationary (First Difference)

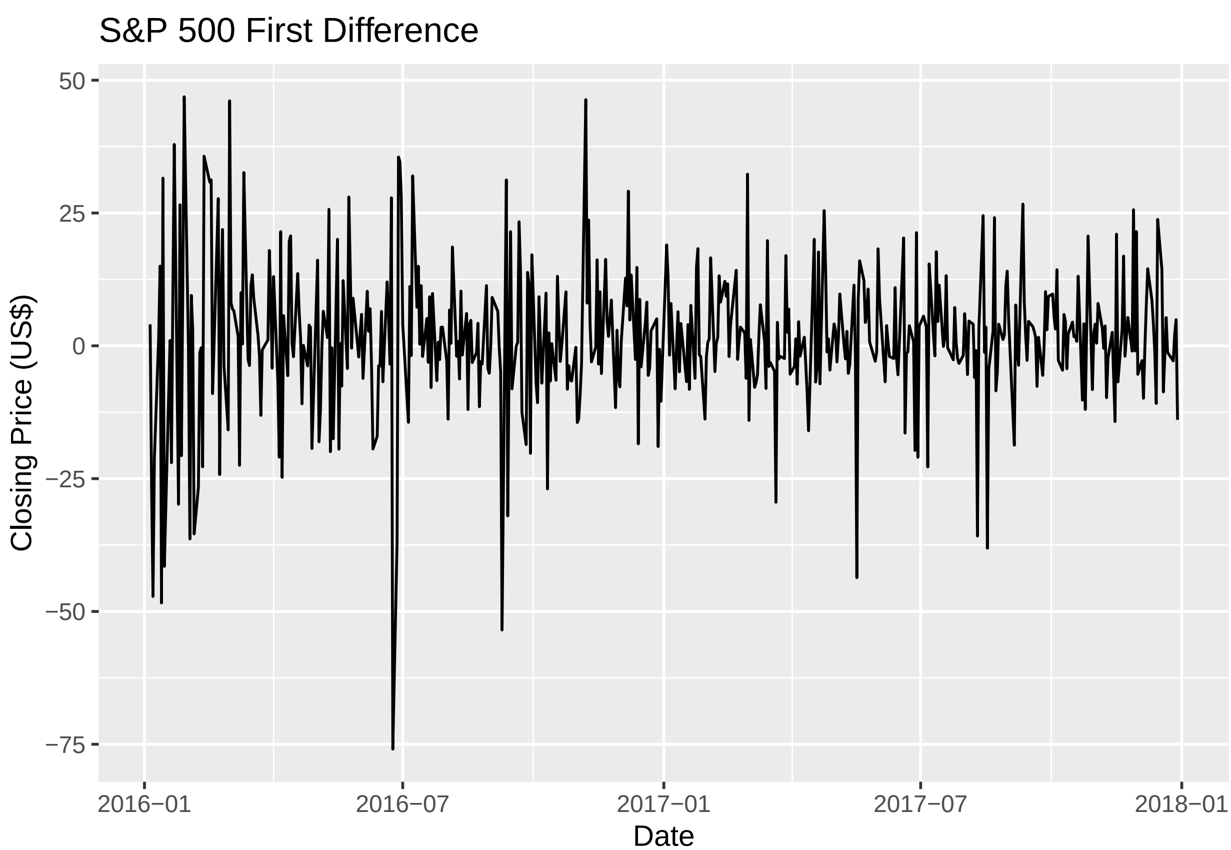
```r
Box.test(sp500_diff, lag = 25, type = 'Ljung-Box')
```

```
##
##  Box-Ljung test
##
## data:  sp500_diff
## X-squared = 24.817, df = 25, p-value = 0.4727
```

After taking the first difference, the p-value is 0.1295 (> 0.05) so we do not have sufficient evidence to reject the null hypothesis that the process is stationary.

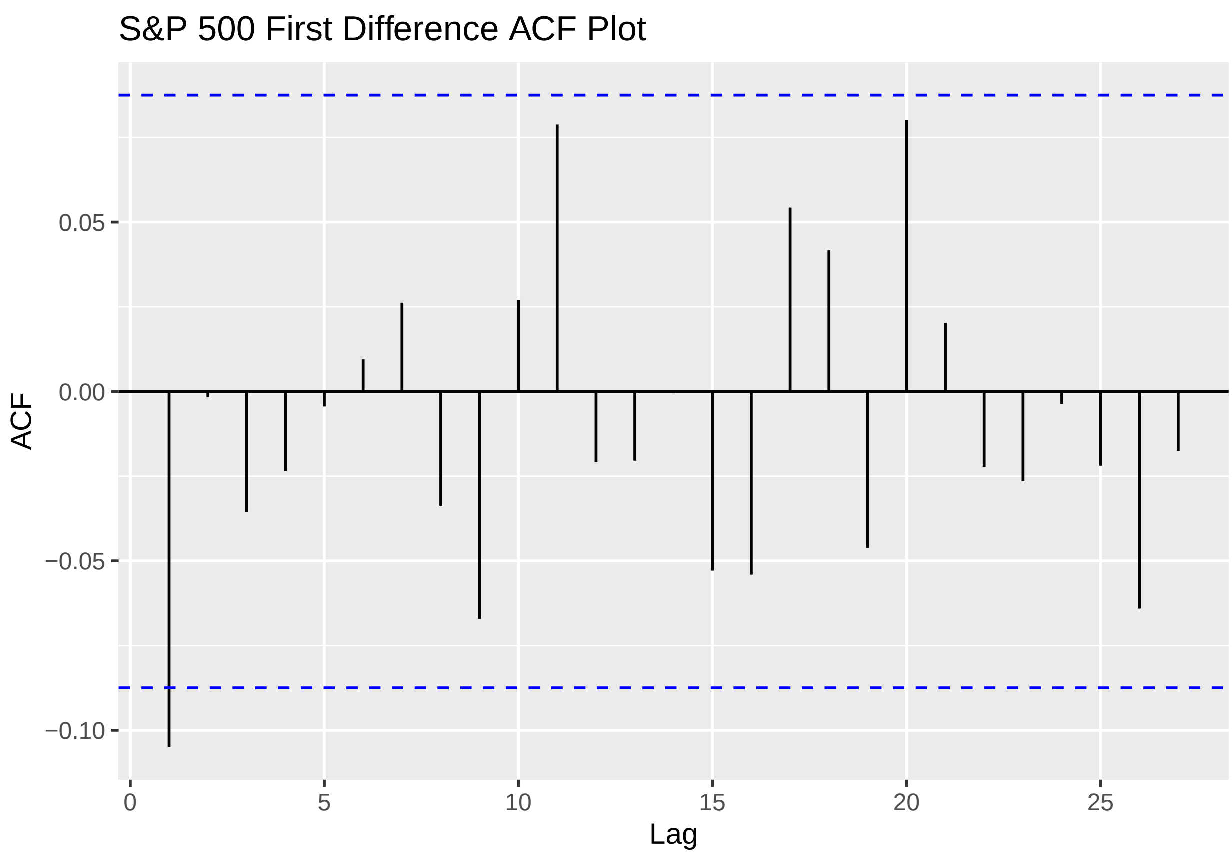
Time Series First Difference Plot

```r
sp500_diff %>% autoplot +
  xlab(label = 'Date') +
  ylab(label = 'Closing Price (US$)') +
  ggtitle(label = 'S&P 500 First Difference')
```



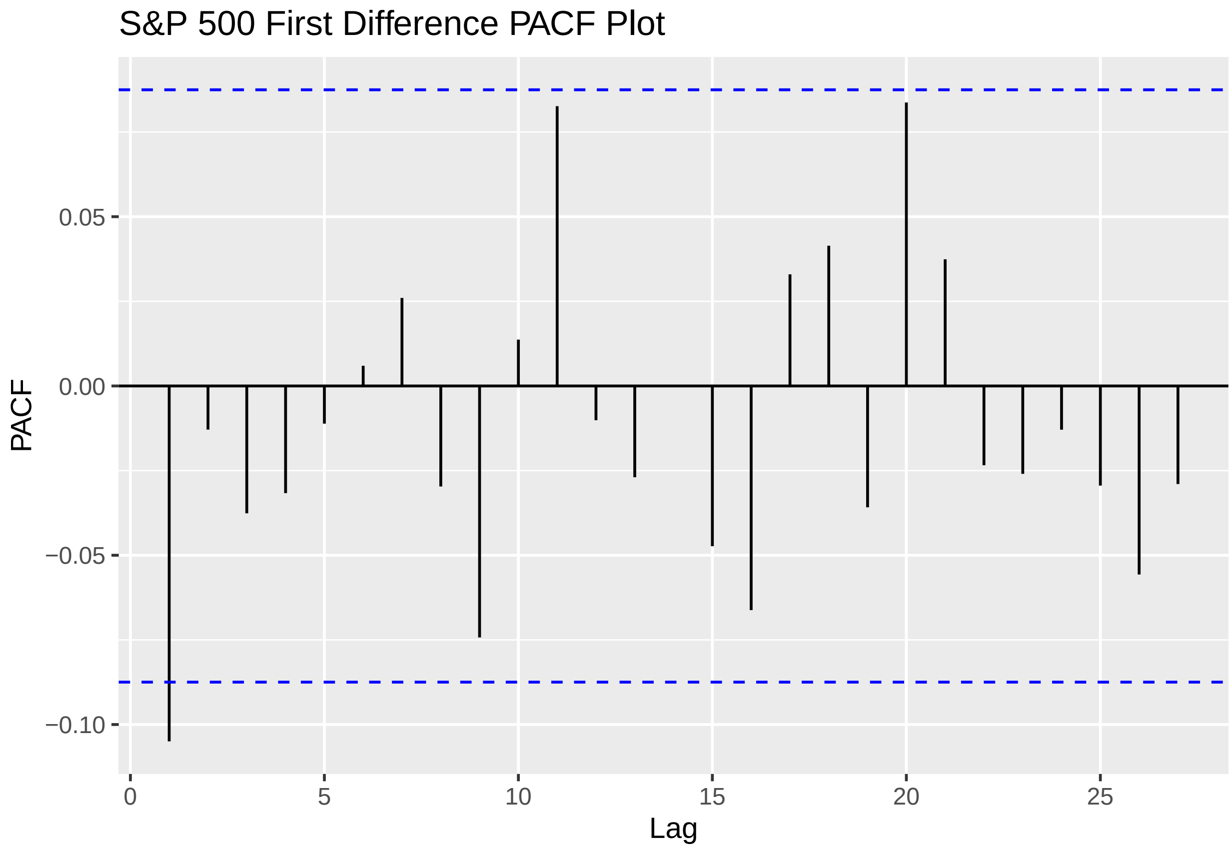
First Difference ACF Plot

```r
sp500_diff %>% ggAcf +
  ggtitle(label = 'S&P 500 First Difference ACF Plot')
```



First Difference PACF Plot

```
sp500_diff %>% ggPacf +
  ggtitle(label = 'S&P 500 First Difference PACF Plot')
```



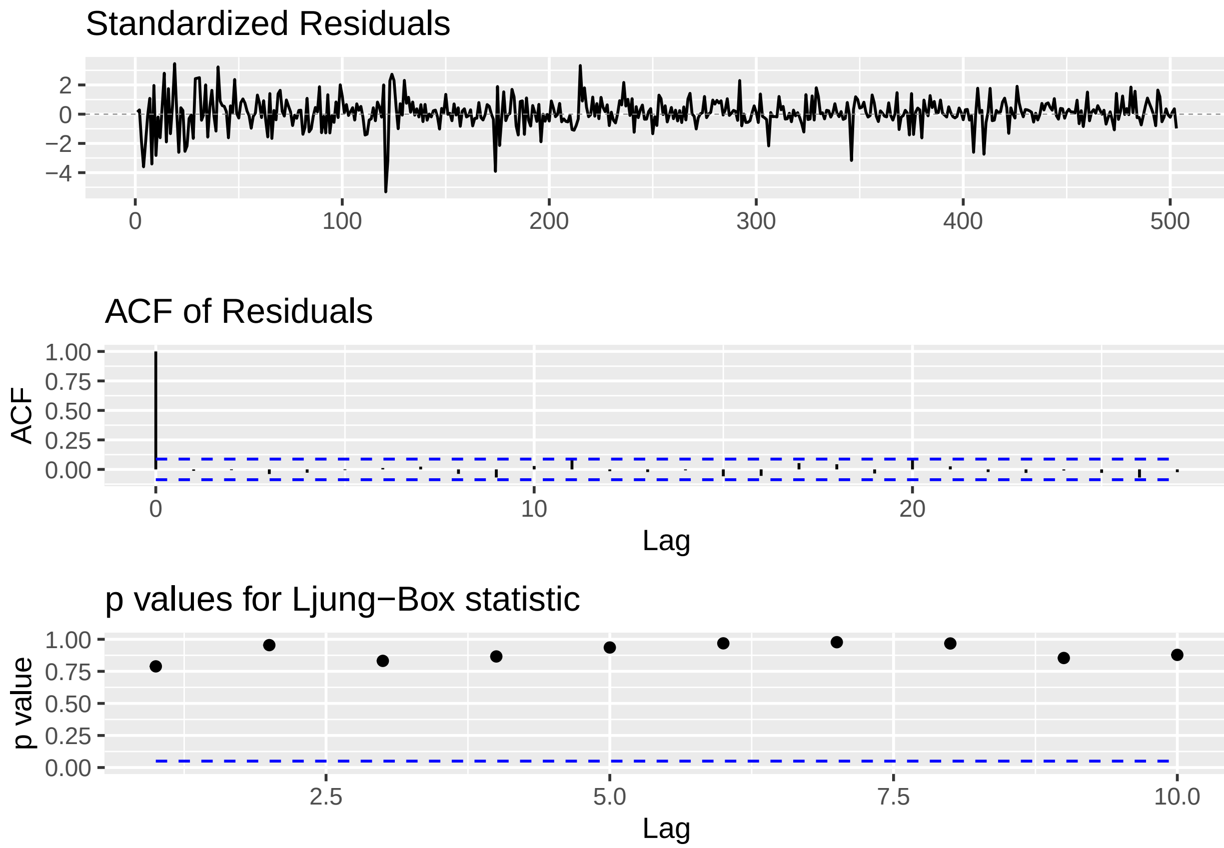
From both plots, we can see that the ACF cuts off at lag 1 and PACF tails off. Thus, MA(1) seems to be the model that best fits our data.

```
model <- arima(x = sp500, order = c(0, 1, 1))
summary(model)
```

```
##
## Call:
## arima(x = sp500, order = c(0, 1, 1))
##
## Coefficients:
##           ma1
##       -0.0949
## s.e.   0.0444
##
## sigma^2 estimated as 190.1:  log likelihood = -2029.47,  aic = 4062.93
##
## Training set error measures:
##                    ME     RMSE      MAE        MPE      MAPE      MASE
## Training set 1.458567 13.77479 9.480134 0.06035895 0.4336831 0.9937431
##                     ACF1
## Training set -0.01190945
```

Residual Analysis

```
model %>% ggtsdiag
```



Portmanteau Test

```
model$residuals %>% Box.test(lag = 25)
```

```
##
##  Box-Pierce test
##
## data:  .
## X-squared = 19.576, df = 25, p-value = 0.7687
```

The p-value of the portmanteau test for residuals is much larger than 0.05. This indicates that the fitted model is appropriate.

Forecasting

Test Set

```
# TODO: Refactor this! This should be done together with train set
getSymbols(Symbols = '^GSPC',
           src     = 'yahoo',
           auto.assign = T,
           from    = '2017-01-01',
           to      = '2020-02-29')
```

```
## [1] "^GSPC"
```

```
sp500_test <- GSPC[, 'GSPC.Close']
sp500_test <- diff(sp500_test)
```

TODO: Evaluate model

```
sp500_forecast <- forecast(model, h = 40)
plot(sp500_forecast)
```