Group Project Report

Group D8

3/16/2020

```r
library(doSNOW)
library(forecast)
library(ggfortify)
library(parallel)
library(quantmod)
library(tcltk)
options('getSymbols.warning4.0' = F)
```

Using data from Yahoo! Finance

```r
getSymbols(Symbols = '^GSPC',
           src     = 'yahoo',
           auto.assign = T,
           from    = '2019-01-01',
           to      = '2020-01-01')
```

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

```r
data <- GSPC[, 'GSPC.Close']
head(data)
```

```
##            GSPC.Close
## 2019-01-02    2510.03
## 2019-01-03    2447.89
## 2019-01-04    2531.94
## 2019-01-07    2549.69
## 2019-01-08    2574.41
## 2019-01-09    2584.96
```

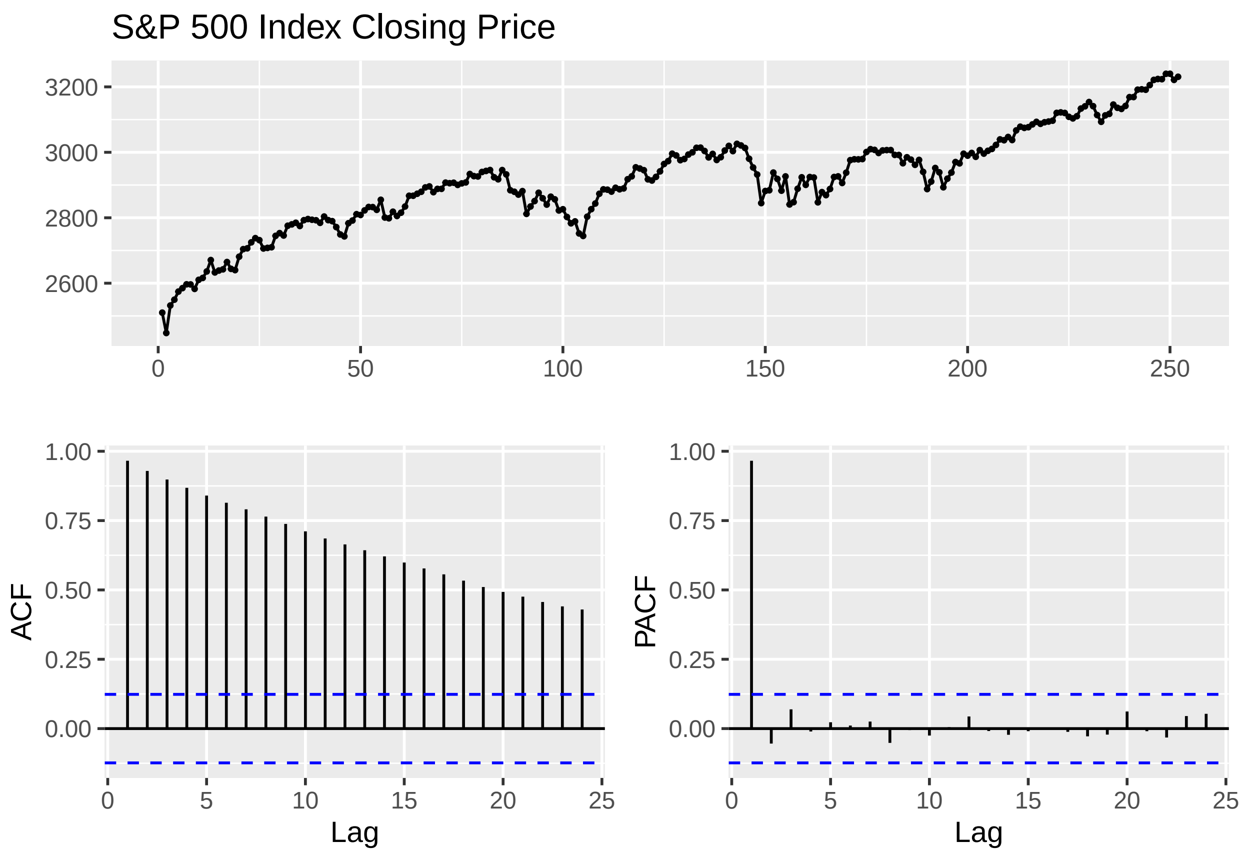
```r
tail(data)
```

```
##            GSPC.Close
## 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
## 2019-12-31    3230.78
```

```r
sp500 <- ts(data)
```

Time Series Plot

```
ggtsdisplay(sp500, main = 'S&P 500 Index Closing Price')
```



The time series plot shows an increasing linear trend with no obvious seasonality.

ACF plot shows a very slow decay in time suggesting that the series might not be stationary.

PACF cuts off at lag 1

Portmanteau Test

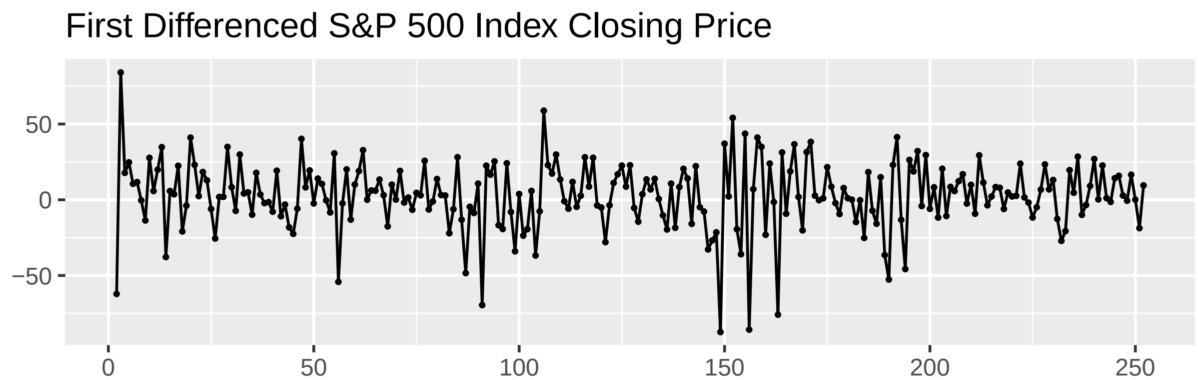
```
Box.test(sp500, lag = 25, type = 'Box-Pierce')
```

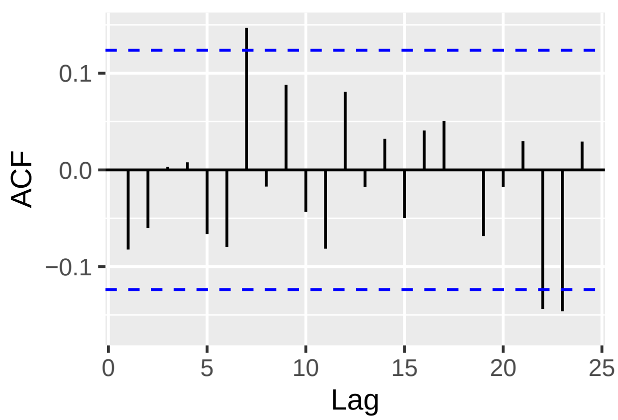
```
##
##  Box-Pierce test
##
## data:  sp500
## X-squared = 2891.3, df = 25, p-value < 2.2e-16
```

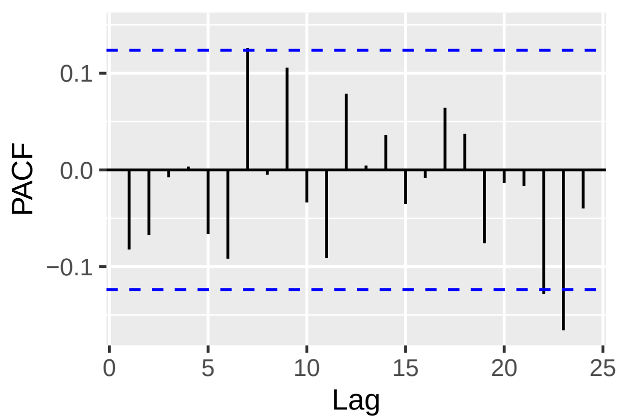
The p-value of the portmanteau test for residuals is less than 0.05. Therefore, we have sufficient evidence to reject the null hypothesis that the series is white noise.

First Differenced Time Series Plot

```
sp500 %>%
  diff() %>%
  ggtsdisplay(main = 'First Differenced S&P 500 Index Closing Price')
```







The differenced series plot shows variation without apparend trend.

ACF plot shows significant values around lower lags and the values seem to decay slowly in time.

Portmanteau Test

```
sp500 %>%
  diff() %>%
  Box.test(lag = 25, type = 'Box-Pierce')
```

```
##
##  Box-Pierce test
##
## data:  .
## X-squared = 30.757, df = 25, p-value = 0.1972
```

The p-value of the portmanteau test for residuals is greater than 0.05. Therefore, we do not have sufficient evidence to reject the null hypothesis that the differenced series is white noise.

Time Series Cross Validation for Model Selection

```
cluster <- makeSOCKcluster(detectCores(logical = T) - 1)
registerDoSNOW(cluster)

nfolds <- length(sp500) - 1
```

```r
# Leave-one-out
# kfolds <- 21:nfolds
# 12-Fold (~1 month rolling window)
kfolds <- round(seq(21, nfolds, length.out = 12))

# For Progress Bar window:
pb <- tkProgressBar(max = length(kfolds))
opts <- list(progress = function(n) setTkProgressBar(pb, n))
# For console output:
# opts <- list(progress = function(n) cat(sprintf('Fold %d is complete\n', n)))

fit_arima <- function(x, p, q) {
  model <- tryCatch({
      return(Arima(x, order = c(p, 1, q), include.constant = T))
  }, error = function(e) {
    tryCatch({
      return(Arima(x, order = c(p, 1, q), include.constant = T, method = 'ML'))
    }, error = function(e) {
      return(Arima(x, order = c(p, 1, q), include.constant = T, method = 'ML', transform.pars = F))
    })
  })
  return(model)
}

score <- foreach(k = kfolds, .options.snow = opts, .packages = c('forecast')) %dopar% {
  # initialize data.frame for each thread
  spe <- data.frame(matrix(0, 5, 5), row.names = c('0', '1', '2', '3', '4'))
  colnames(spe) <- rownames(spe)
  # Split sp500 data into train and validation set
  train <- sp500[1:k]
  validation <- sp500[k + 1]
  for (p in 0:4) {
    for (q in 0:4) {
      if (p == 0 && q == 0) next # Skip ARIMA(0, 1, 0)
      model <- fit_arima(x = train, p, q)
      y_hat <- forecast(model, h = 1)$mean[1]
      spe[as.character(p), as.character(q)] <- (y_hat - validation)^2
    }
  }
  return(spe)
}
close(pb)

rmspe <- sqrt(Reduce('+', score) / length(score))
result <- data.frame(matrix(ncol = 3, nrow = 0))
colnames(result) <- c('p', 'q', 'RMSPE')
for (i in 1:5) {
  for (j in 1:5) {
    if (i == 1 && j == 1) next
    result[nrow(result) + 1,] <- c(i - 1, j - 1, rmspe[i, j])
  }
}
print(result[order(result$RMSPE)[1:5], ])
```

```
##    p q    RMSPE
## 10 2 0 23.62269
## 15 3 0 23.66274
## 1  0 1 23.81094
## 2  0 2 23.94131
## 5  1 0 23.98198
```

From the result table, we can see that the model ARIMA(2, 1, 0) has the lowest mean squared prediction error.

Fit best model on the full train data

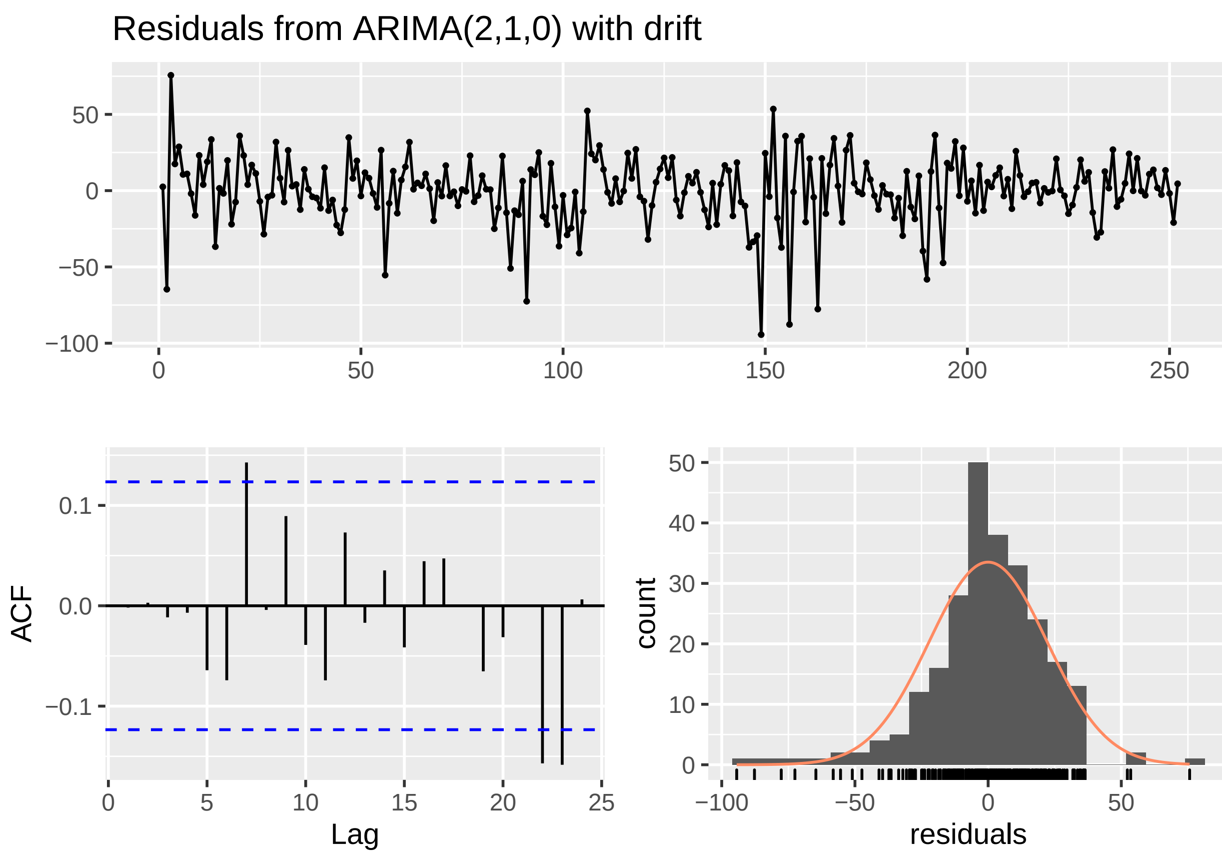
```r
best_model <- Arima(sp500, order = c(2, 1, 0), include.constant = T)
summary(best_model)
```

```
## Series: sp500
## ARIMA(2,1,0) with drift
##
## Coefficients:
##           ar1      ar2   drift
##       -0.0877  -0.0690  2.8889
## s.e.   0.0638   0.0656  1.2111
##
## sigma^2 estimated as 497.8:  log likelihood=-1134.01
## AIC=2276.03   AICc=2276.19   BIC=2290.13
##
## Training set error measures:
##                        ME     RMSE     MAE         MPE      MAPE      MASE
## Training set -0.006066974 22.13254 15.8986 -0.00158631 0.5537549 0.9726619
##                      ACF1
## Training set -0.001787564
```

Residual Analysis

```r
ggtsdiag(best_model)
```


```r
checkresiduals(best_model, lag = 25)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0) with drift
## Q* = 30.627, df = 22, p-value = 0.104
##
## Model df: 3.   Total lags used: 25
```

We can see that there is no pattern apparent in the residuals analysis plot. The acf values are not significant for lags other than 0. THe p-values for Ljung-Box test are also large suggesting nothing untoward about the fit of the model.

Forecasting

Retrieve the next 5 closing prices

```
getSymbols(Symbols = '^GSPC',
           src      = 'yahoo',
           auto.assign = T,
           from     = '2020-01-01',
           to       = '2020-03-31')
```

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

```
data <- GSPC[1:5, 'GSPC.Close']
head(data)
```

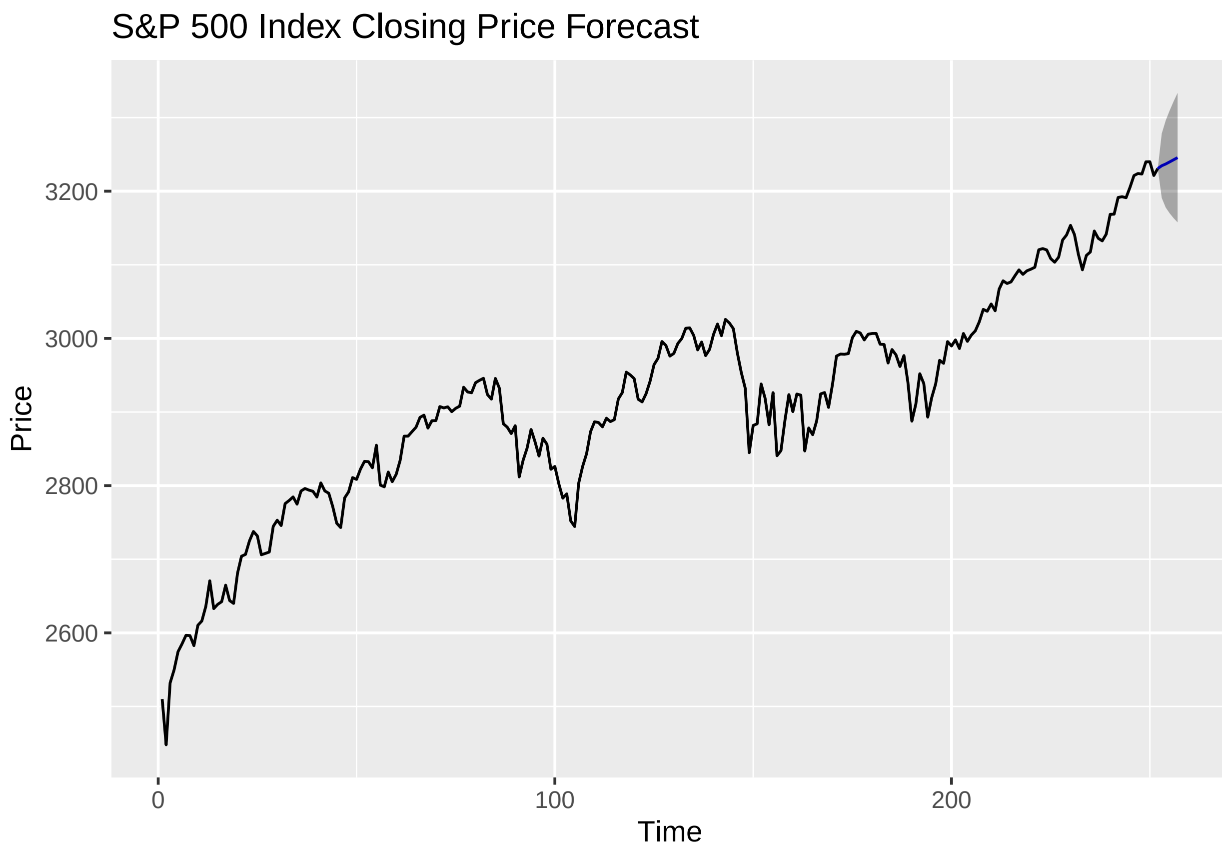
```
##              GSPC.Close
```

```
## 2020-01-02    3257.85
## 2020-01-03    3234.85
## 2020-01-06    3246.28
## 2020-01-07    3237.18
## 2020-01-08    3253.05
```

```
test <- as.vector(data)
```

Forecast

```
forecast <- forecast(best_model, h = 5, level = 95)
autoplot(forecast) +
  ggtitle(label = 'S&P 500 Index Closing Price Forecast') +
  ylab(label = 'Price') +
  xlab(label = 'Time')
```



Evaluate MSE

```
pred <- as.vector(forecast$mean)
accuracy(pred, test)
```

```
##                ME     RMSE      MAE       MPE      MAPE
## Test set 5.905451 11.60757 8.961729 0.1811685 0.2755987
```

Prediction vs Actual

```
forecast_data <- data.frame(date = index(data),
                            price = c(pred, test),
                            predicted = pred,
                            actual = test)
ggplot(forecast_data, aes(x = date, y = predicted)) +
  geom_line(aes(color = 'Predicted')) +
  geom_line(aes(y = actual, color = 'Actual')) +
  xlab(label = 'Time') +
  ylab(label = 'Closing Price (US$)') +
  ggtitle(label = 'S&P 500 Index Closing Price Forecast') +
  theme(legend.title = element_blank())
```