

174 Final Project

Seth Marceno (8934838)

3/14/2020

Abstract

The problem I plan to address in this project is to forecast US dollar to Euro exchange rates in order to gain some insight into possible investment opportunities either in Europe or domestically in the United States by gauging the well-being of the respective economies. In order to do this I utilize time series models, specifically an $ARIMA(6, 1, 5)$ model, (algebraically given as: $U_t = (1 + 0.7148B^3 - 0.3156B^4 - 0.2840B^5 - 0.1465B^6)(1 - 0.7673B^3 + 0.2683B^4 + 0.1488B^5)Z_t$ where $U_t = \nabla_1 \frac{X_t^{-2}-1}{-2} = (1 - B)^1 \frac{X_t^{-2}-1}{-2}$), in order to estimate future exchange rate values. In doing so, we find that the model used gives the exchange rate for the next 10 days to be an average of about 1.1 with a standard error of around 0.00469, thus concluding that the buying power of the US dollar is slowly approaching an even 1:1 with the Euro, and continuing its downward trend. However, we find that these numbers are consistently overestimating the true values, and may not encapsulate the true volatility of these rates over time. I find that in order to accurately forecast this economic data, a different type of model, presumably an ARCH or GARCH model, may fit the data better and forecast with better accuracy.

Project Report (Main Body)

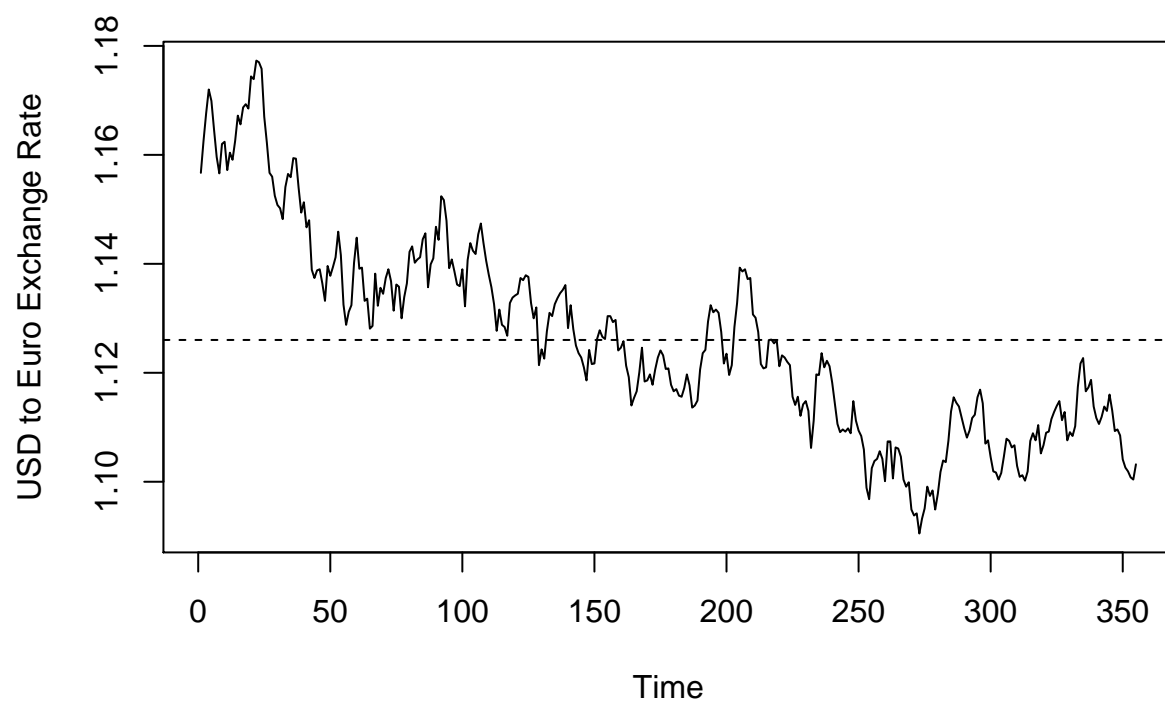
Introduction

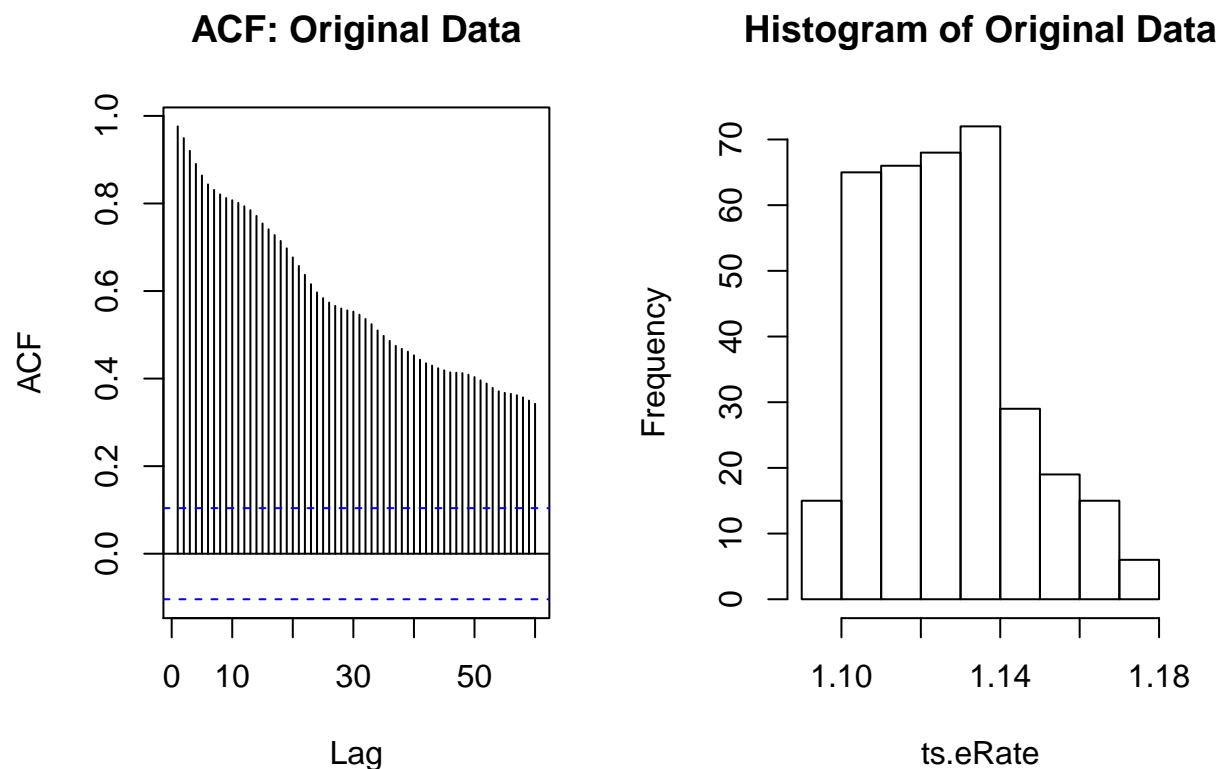
In this project I examine the exchange rate of the Euro to the US dollar from October 5, 2018 until February 28, 2020. This dataset was obtained from the Federal Reserve Bank of St. Louis' website FRED. The data I am examining is of interest because exchange rates of between the USD and other currencies are good indicators of the well-being of the United States economy. If United States currency has greater buying power over seas it is a good indicator that the economy is in a healthy place, and vice versa.

In order to do any statistical analysis, I utilized Rstudio and R. When fitting models to the data the first step that had to be done was to obtain stationary data. To do this I used a Box-Cox transformation in order to obtain constant variance, and then differenced at lag one in order to eliminate the negative trend that was in the data. Once the data was stationary I examined the autocorrelation and partial autocorrelation graphs in order to get some ideas as to what the order of the AR and MA parts of my model could be. Once I had these values, I fit all variations of an $ARIMA(p, 1, q)$ model on my data and found the model with the lowest AICC is an $ARIMA(6, 1, 5)$. Once I had my model, I did tests to fix variables to zero in order to lower the AICC. Once this was done, and I found that the model was causal, invertible, and residuals that were normally distributed with mean 0 and variance $5.264101e - 06$, I was able to continue onto forecasting. From forecasting I found that the exchange rate for the next 10 days is predicted to stay around 1.1 US dollar for every 1 Euro. This average however is a constant overestimation of the true data and does not capture the true downwards trend of the exchange rate. This indicates that the model used to forecast is not the best fit, and that an ARCH or GARCH model may be more suitable for this problem.

Exploratory Analysis

The first thing we will do is analyze the graph of our time series along with its ACF and PACF in order to determine if our data is stationary or not.





(i) **Trend:** Based off of the graph of our data, we see a clear downwards trend, a breach in stationarity. On top of this, looking at the sample ACF we see a slow decay, an indication of trend. To correct this we will look to difference our series at lag 1.

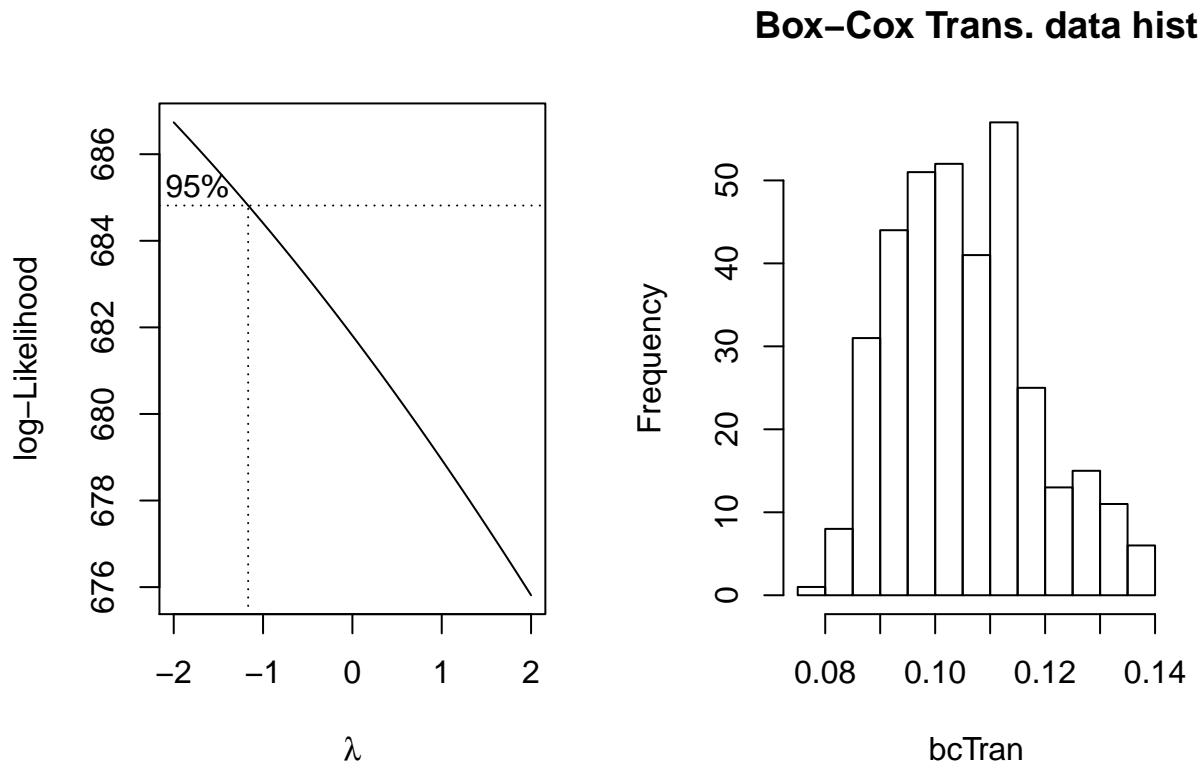
(ii) **Seasonality:** There does not seem to be any glaring seasonality just by looking at the plot of our time series. Similarly, examining the ACF graph there doesn't seem to be any cyclical nature that would imply seasonality.

(iii) **Sharp changes in behavior:** There does not seem to be anything abnormal in the plotting of the data. Overall we see some trend that must be eliminated in order to obtain stationarity; however, there does not seem to be any sharp changes in behavior.

Obtaining Stationarity

Nonconstant Variance Check

Since our histogram seems to be a bit skewed to the right, we will look at a box cox transformation of the data to see if we can correct for this.

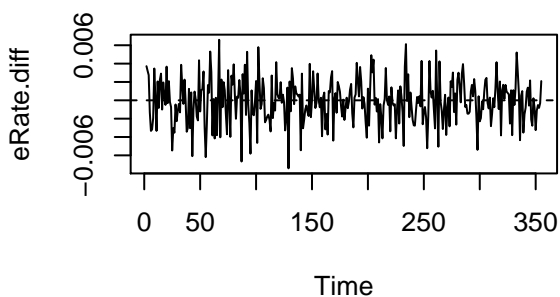
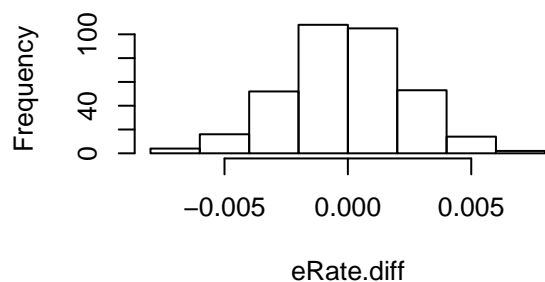


We find that the best λ is $\lambda = -2$. Thus transforming our data as $U_t = \frac{X_t^\lambda - 1}{\lambda}$ seems to correct for the skewness in the prior data. Therefore we will continue with the transformed data.

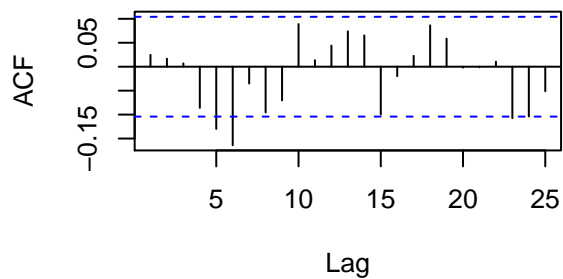
Eliminating Trend

To eliminate trend, we will difference our series at lag 1 and check the ACF to see if we still have slowly decaying autocorrelations.

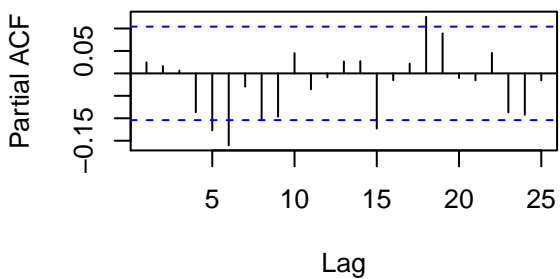
Histogram of Differenced data



Differenced ACF



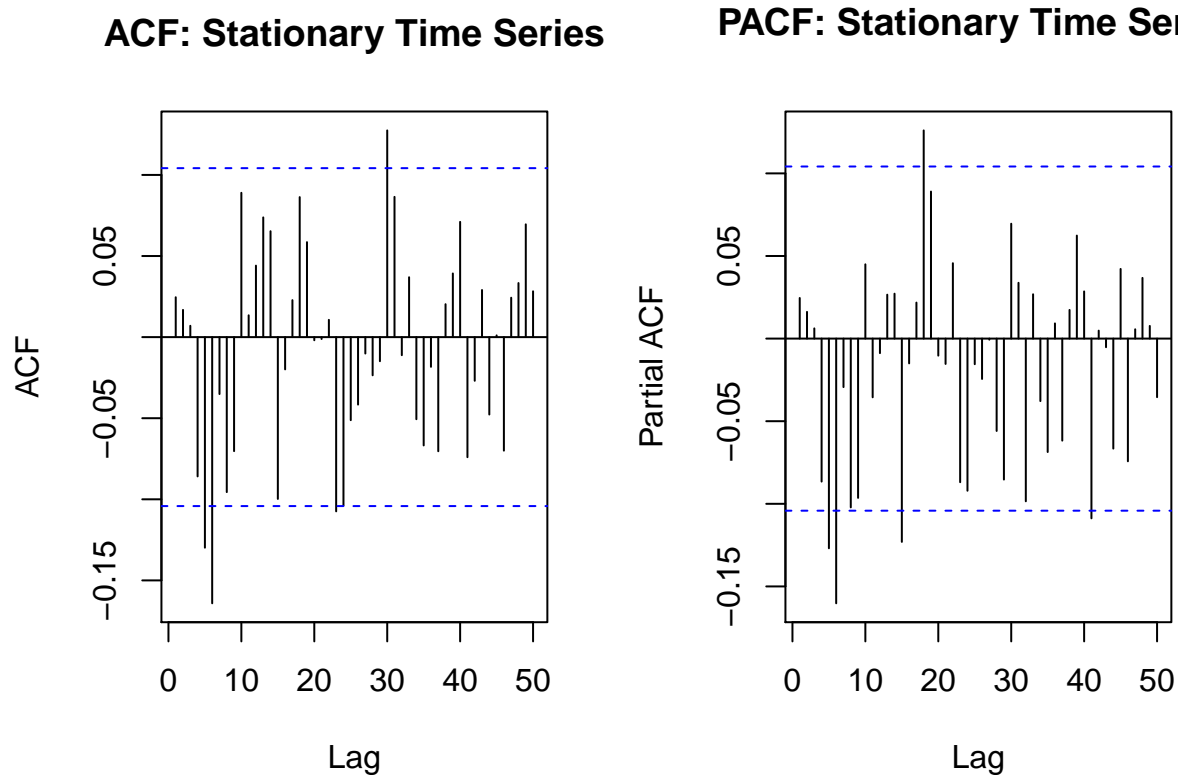
Differenced PACF



After differencing once at lag one, the data shows signs of stationarity. The variance of our data drops from 0.000163 to $5.877e - 06$, our autocorrelations no longer have slow decay, and the histogram of the data is normal and symmetric. Thus with stationary data we will continue to model estimation.

Model Estimation

To preliminarily fit our models, we will inspect the ACF/PACF graphs for our stationary series:



Modeling (p, d, q) for an *ARIMA* model:

- We applied one difference at lag 1 in order to eliminate trend, thus $d = 1$
- The PACF looks to be non-zero at lags 5, 6, 15, and 20. Thus $p = 5, 6, 15, 20$
- The ACF looks to be non-zero at lags 5, 6, and 30. Thus $q = 5, 6$ and possibly 30.

Now looking at AICC to find our p and q for our model:

```
##      q
## p      0      1      2      3      4      5      6
## 0 -3257.450 -3255.684 -3253.805 -3251.978 -3252.441 -3257.010 -3262.891
## 1 -3255.693 -3253.720 -3251.780 -3250.366 -3261.332 -3261.039 -3260.836
## 2 -3253.793 -3251.757 -3256.552 -3254.151 -3265.291 -3264.715 -3258.900
## 3 -3251.785 -3250.227 -3254.140 -3249.513 -3263.008 -3261.236 -3268.620
## 4 -3252.281 -3264.162 -3265.358 -3263.827 -3265.446 -3259.080 -3269.936
## 5 -3255.909 -3264.549 -3263.329 -3261.197 -3259.171 -3263.447 -3268.948
## 6 -3263.127 -3263.561 -3262.228 -3271.912 -3270.775 -3269.577 -3267.696
## 7 -3261.336 -3261.842 -3260.282 -3270.098 -3270.501 -3268.756 -3266.601
## 8 -3262.462 -3261.783 -3264.092 -3260.164 -3267.243 -3265.112 -3264.458
##      q
## p      7      8
## 0 -3260.827 -3259.287
## 1 -3258.735 -3257.524
## 2 -3256.917 -3262.242
## 3 -3266.658 -3265.721
```

```

##      4 -3270.431 -3270.150
##      5 -3268.844 -3269.981
##      6 -3266.680 -3267.280
##      7 -3264.528 -3265.321
##      8 -3266.165 -3259.060

##      q
## p      0      1      2      3      4      5      6      7      8
## 0 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 4 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 5 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 6 FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 7 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 8 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

```

From fitting all combinations of $ARIMA(p, 1, q)$ where $p = 0$ to 8 and $q = 0$ to 8 we find that the model with the lowest AICC value corresponds to the $ARIMA(6, 1, 3)$ model.

Fitting Models

Nine models that I will fit from the ACF/PACF graphs are the following:

- (i) $ARIMA(5, 1, 5)$
- (ii) $ARIMA(6, 1, 5)$
- (iii) $ARIMA(5, 1, 6)$
- (iv) $ARIMA(6, 1, 6)$
- (v) $ARIMA(5, 1, 15)$
- (vi) $ARIMA(6, 1, 15)$
- (vii) $ARIMA(5, 1, 20)$
- (viii) $ARIMA(6, 1, 20)$

From AICC:

- (ix) $ARIMA(6, 1, 3)$

Preliminary Fits of Models (i - ix)

```

##      q
## p      5      6
## 3 -3261.236 -3268.620
## 5 -3263.447 -3268.948
## 6 -3269.577 -3267.696
## 15 -3260.881 -3258.778

## [1] "p = 5, q = 20, AICC: -3265.17615583666"
## [1] "p = 6, q = 20, AICC: -3262.70690551844"
## [1] "p = 30, q = 5, AICC: -3254.36085806985"
## [1] "p = 30, q = 6, AICC: -3250.71380256465"
## [1] "p = 30, q = 20, AICC: -3234.25684387636"

```

- NOTE: We were given the warning that NaNs were produced, thus one variation of the above models is not suitable. If the model with the lowest AICC gives this message when we fit it to our data then we know that this model will not work.

```
##      q
## p      5      6
## 3 FALSE FALSE
## 5 FALSE FALSE
## 6  TRUE FALSE
## 15 FALSE FALSE
```

We find the model with the lowest AICC value is model (ii), $ARIMA(6, 1, 5)$. Similarly the model with the second lowest AICC value is model (iii), $ARIMA(5, 1, 6)$.

In fitting both models we are given no warnings about NaNs produced; therefore, both models are viable options. However, Since the $ARIMA(6, 1, 5)$ model has a lower AICc and has the same amount of parameters as $ARIMA(5, 1, 6)$ we will go forth checking to see if the $ARIMA(6, 1, 5)$ is both causal and invertible.

Examining the coefficients given to us for the $ARIMA(6, 1, 5)$ model:

```
##
## Call:
## arima(x = bcTran, order = c(6, 1, 5), method = "ML", optim.control = list(maxit = 1000))
##
## Coefficients:
##      ar1      ar2      ar3      ar4      ar5      ar6      ma1      ma2
##      -0.2523  0.1007  0.7032 -0.1404 -0.4254 -0.1792  0.2551 -0.0951
## s.e.    0.2887  0.2296  0.0776  0.2103  0.2076  0.0659  0.2913  0.2406
##      ma3      ma4      ma5
##      -0.7507  0.0598  0.2906
## s.e.    0.0828  0.2468  0.2324
##
## sigma^2 estimated as 5.284e-06:  log likelihood = 1647.17,  aic = -3272.35
```

Checking each respective confidence interval to determine if 0 is contained within the interval for a given coefficient, we find that possible 0 coefficients could be ϕ_1 , ϕ_2 , ϕ_4 , θ_1 , θ_2 , θ_4 , and θ_5 .

Testing all variations zeroing these coefficients we find:

```
## [1] "Lowest AICc: -3277.10193923722 phi1: 0 phi2: 0 theta1 0 theta2 0"
```

Our result is that we obtain a lower AICc when we fix coefficients ϕ_1 , ϕ_2 , θ_1 , and θ_2 to be 0.

Comparing the AICC value of our original $ARIMA(6, 1, 5)$ to our new $ARIMA(6, 1, 5)$ model with coefficients ϕ_1 , ϕ_2 , θ_1 , and θ_2 fixed as zero, we find that the AICC value drops from -3269.577 to -3277.102 . Thus we will go forth with our new $ARIMA$ model with some fixed coefficients as 0, as seen below:

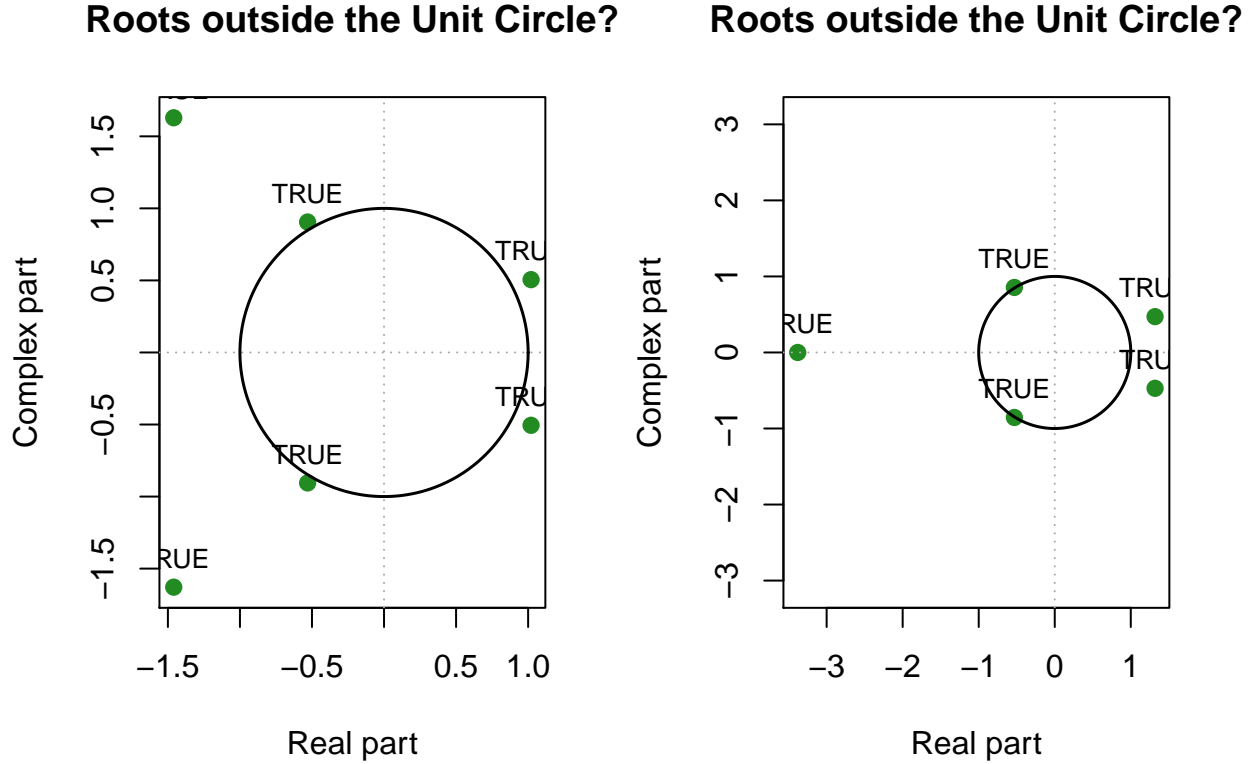
```
##
## Call:
## arima(x = bcTran, order = c(6, 1, 5), fixed = c(0, 0, NA, NA, NA, NA, 0, 0,
##      NA, NA, NA), method = "ML", optim.control = list(maxit = 1000))
##
## Coefficients:
##      ar1 ar2      ar3      ar4      ar5      ar6 ma1 ma2      ma3      ma4
##      0   0  0.7148 -0.3156 -0.2840 -0.1465  0   0 -0.7673  0.2683
## s.e.    0   0  0.0635  0.0587  0.0586  0.0497  0   0  0.0553  0.0539
##      ma5
##      0.1488
## s.e.    0.0525
```



```
##
```

```
## sigma^2 estimated as 5.291e-06: log likelihood = 1646.94, aic = -3279.87
```

Going forth with these coefficients, we will now look at the $MA(5)$ component to see if the model is invertible, and the $AR(6)$ component to determine if the model is causal. Below we see the unit circle and root checks for $\phi(z) = 1 - 0.7148z^3 + 0.3156z^4 + 0.2840z^5 + 0.1465z^6$ on the left, and $\theta(z) = 1 - 0.7673z^3 + 0.2683z^4 + 0.1488z^5$ on the right.



Thus since all of the roots for both $\phi(z)$ and $\theta(z)$ lie outside of the unit circle, we have both causality and invertibility. Therefore we will go forth with this model.

Below we have the algebraic form of the current $ARIMA(6, 1, 5)$ model:

$$U_t = (1 + 0.7148B^3 - 0.3156B^4 - 0.2840B^5 - 0.1465B^6)(1 - 0.7673B^3 + 0.2683B^4 + 0.1488B^5)Z_t$$

Where we have U_t is the stationary process $U_t = \nabla_1 \frac{X_t^{-2}-1}{-2} = (1 - B)^1 \frac{X_t^{-2}-1}{-2}$.

Diagnostic Checking

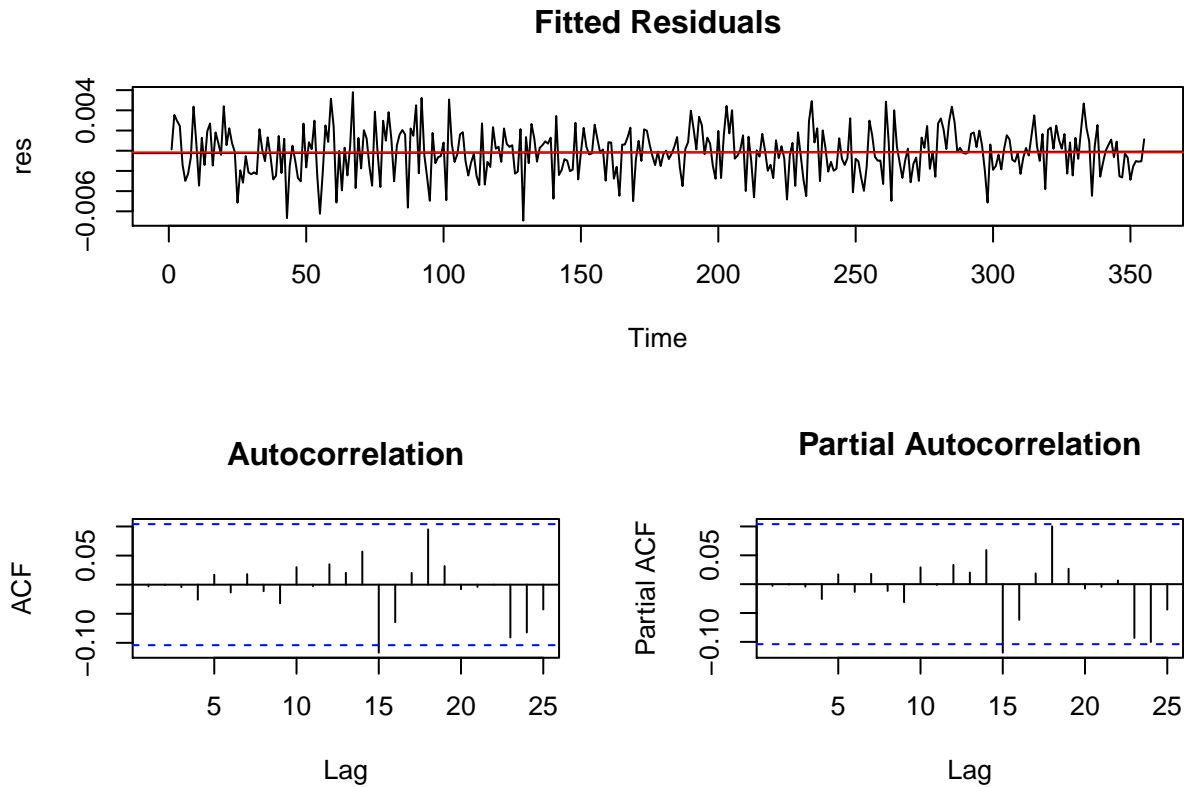
Now that we have our $ARIMA(6, 1, 5)$ model we will analyze the residuals to determine if our model is suitable for forecasting.

Residual Analysis

Checking the mean and variance of the residuals of our fit:

```
## [1] "Mean: -0.000162955379142604 Var: 5.26410134899969e-06"
```

we see a mean of about 0, and a variance of $5.264101e - 06$.



- NOTE: We have ACF and PACF lags at 15 that are outside of the confidence interval; however, we can count these as zero through the justification of Bartlett's formula.

Box-Pierce Test

```
##
## Box-Pierce test
##
## data:  res
## X-squared = 13.026, df = 12, p-value = 0.3671
```

We see a p-value of $0.3671 > \alpha = 0.05$, thus we fail to reject that our residuals are not independent.

Ljung-Box Test

```
##
## Box-Ljung test
##
## data:  res
## X-squared = 13.681, df = 12, p-value = 0.3215
```

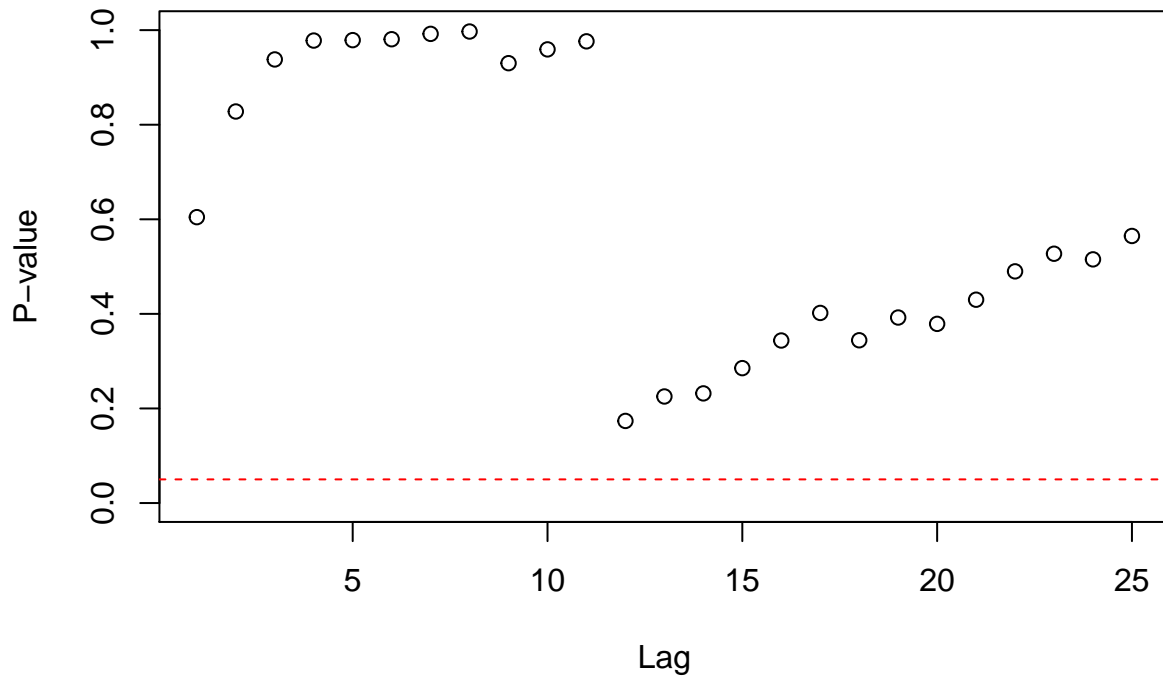
We see a p-value of $0.3215 > \alpha = 0.05$, thus we fail to reject that our autocorrelations are non-zero.

McLeod-Li Test

```
##
## Box-Ljung test
```

```
##
## data:  res^2
## X-squared = 20.198, df = 19, p-value = 0.3828
```

We see a p-value of $0.3828 > \alpha = 0.05$, thus we fail to reject that there are no autoregressive conditional heteroskedasticity among the lags considered.



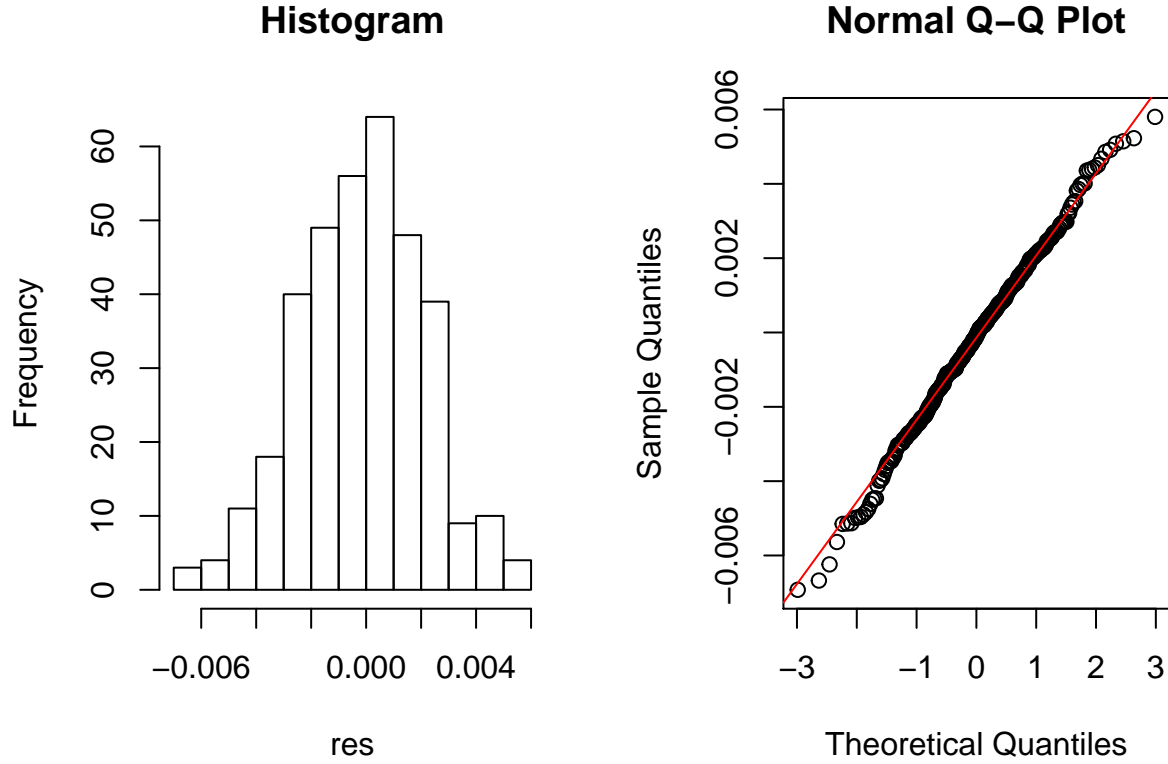
Similarly, we can see that all lags consider are above the significance threshold, and can thus be considered 0.

Shapiro Test: Normality of Residuals

```
##
## Shapiro-Wilk normality test
##
## data:  res
## W = 0.99648, p-value = 0.628
```

We see a p-value of $0.628 > \alpha = 0.05$, thus we fail to reject that our residuals are normally distributed.

Histogram and Q-Q plot of residuals



We see a symmetric and normal histogram, as well as a Q-Q plot that does not indicate any breaches in normality. Thus we conclude that our residuals are normally distributed with mean 0, and variance $5.264101e - 06$.

Hence we see that our model is suitable for forecasting.

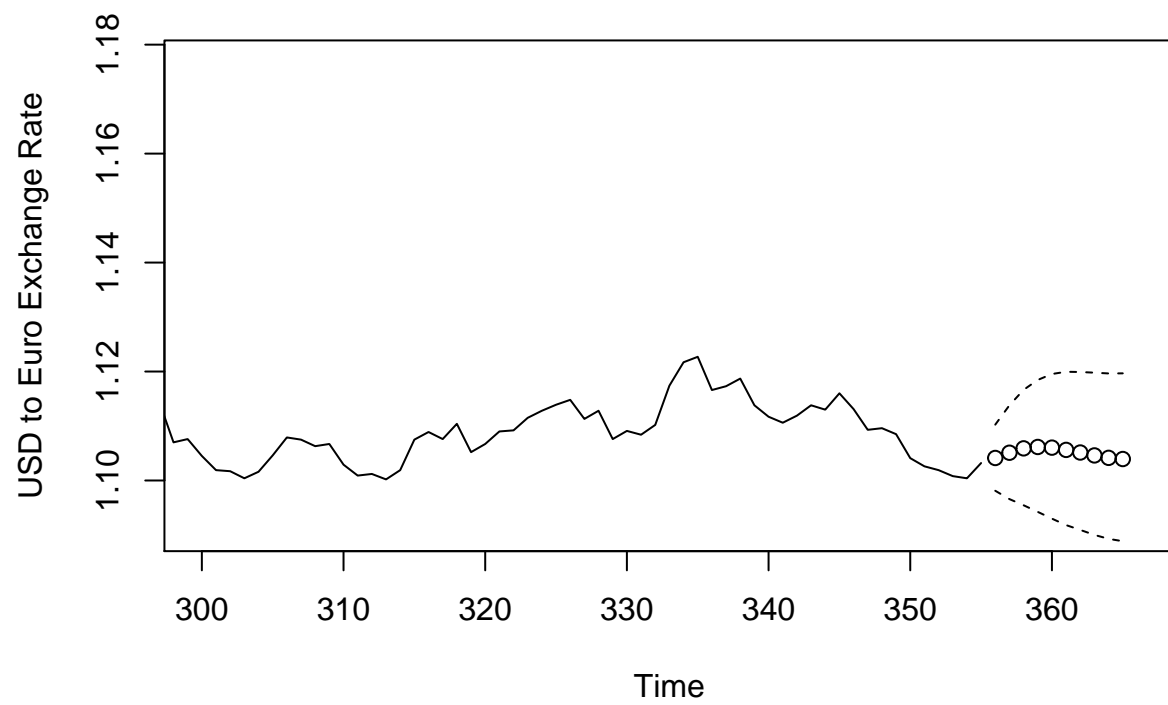
Forecasting

Now using the final $ARIMA(6, 1, 5)$ model, algebraically written as:

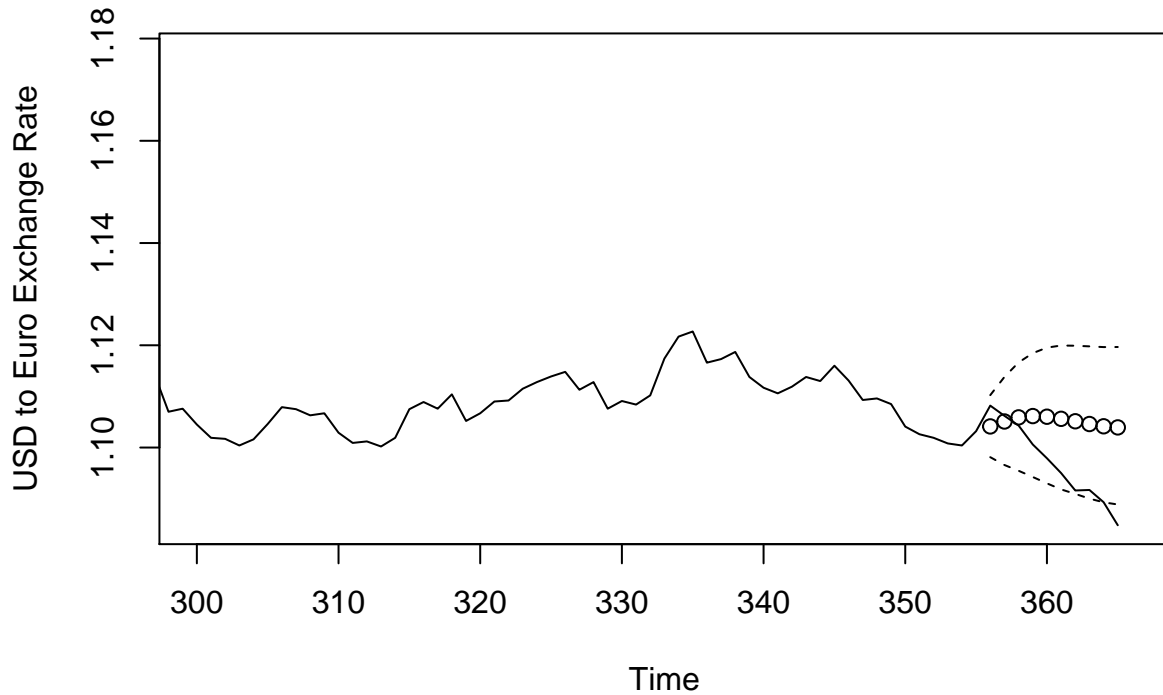
$$U_t = (1 + 0.7148B^3 - 0.3156B^4 - 0.2840B^5 - 0.1465B^6)(1 - 0.7673B^3 + 0.2683B^4 + 0.1488B^5)Z_t$$

Where we have U_t is the stationary process $U_t = \nabla_1 \frac{X_t^{-2} - 1}{-2} = (1 - B)^1 \frac{X_t^{-2} - 1}{-2}$. With $\sigma_Z^2 = 5.291e - 06$

We see graphed below the estimates of the exchange rate for the next 10 days.



Comparing these estimated values with the true values:



Predicted Values:

```
## Time Series:
## Start = 356
## End = 365
## Frequency = 1
## [1] 1.104132 1.105095 1.105896 1.106151 1.106029 1.105616 1.105134 1.104599
## [9] 1.104153 1.103937
```

True Values:

```
## [1] 1.1082 1.1062 1.1043 1.1006 1.0979 1.0950 1.0916 1.0917 1.0893 1.0848
```

We find that the estimated exchange rate for the upcoming 10 days is going to stay around 1.1 with minimal volatility. However, we notice that, compared to the true values, we are overestimating. Furthermore, at 10 days ahead we see that the true value of the exchange rate drops below the estimated lower bound of the confidence interval.

Conclusion

We see that the estimations for 10 days ahead using an $ARIMA(6,1,5)$ overestimate the true values of the exchange rate on those days. The model predicts around a constant 1.1 US dollar to 1 Euro; however, the true rate seems to be dropping from 1.1 to 1.08. The reason behind this constant overestimation may be that these predictions stay constant with minimal variance because we have that our shocks have variance $\sigma_Z^2 = 5.29e - 06$, which is practically zero, so we don't expect the model to be very volatile. Because the model is not very volatile, it does not encapsulate the true movement of the exchange rate. The goal of forecasting exchange rates is not fulfilled because this model can not confidently and accurately predict

exchange rates past a 10 day threshold. In light of this, this economic dataset would be a better fit for an ARCH or GARCH model, that is used heavily in econometrics for financial data.

References

[1] Data obtained from the Federal Reserve Bank of St. Louis website FRED: <https://fred.stlouisfed.org/series/DEXUSEU>

Appendix

```
# Reading in Exchange Rate dataset Obtained from FRED
Data <- na.omit(DEXUSEU)
eRate <- Data[c(876:(nrow(Data)-10)), 2]
ts.eRate <- as.ts(eRate[1:355])
test <- eRate[(length(eRate)-9):length(eRate)]

# Code to plot the time series data
ts.plot(ts.eRate, ylab = "USD to Euro Exchange Rate")
abline(h=mean(ts.eRate), lty = 2)

# Code to plot the autocorrelation graphs and histogram
par(mfrow = c(1, 2))
acf(ts.eRate, main = 'ACF: Original Data', lag.max = 60)
hist(ts.eRate, main = 'Histogram of Original Data')

#Code to run Box-Cox Transformation and plot resulting histogram
time <- 1:(length(ts.eRate))
vals <- ts.eRate[1:length(ts.eRate)]

par(mfrow = c(1, 2))
bc <- boxcox(vals~time)
hist(bcTran, breaks = 15, main = 'Box-Cox Trans. data hist')

lambda <- bc$x[which(bc$y == max(bc$y))]
bcTran <- ((ts.eRate^lambda) - 1)/(lambda)

#Code that takes difference at lag 1 and plots
#the resulting autocorrelations and partial autocorrelations
eRate.diff <- diff(bcTran, lag = 1)

#Plotting the series with its Histogram
par(mfrow = c(2, 2))
hist(eRate.diff, breaks = 10, main = 'Histogram of Differenced data')
```

```

ts.plot(eRate.diff)
abline(h = 0, lty = 2)
acf(eRate.diff, main = 'Differenced ACF')
pacf(eRate.diff, main = 'Differenced PACF')
var(bcTran)
var(eRate.diff)

#Plotting ACF/PACF of differenced data
par(mfrow = c(1, 2))
acf(eRate.diff, lag.max = 50, main = 'ACF: Stationary Time Series')
pacf(eRate.diff, lag.max = 50, main = 'PACF: Stationary Time Series')

#Fits all variation of ARIMA(p, 1, q) where p = 0:8 and q = 0:8,
#then adds the AICC value of the respective model to the matrix names AICC
aiccs <- matrix(NA, nr = 9, nc = 9)
dimnames(aiccs) = list(p=0:8, q=0:8)
for(p in 0:8) {
  for(q in 0:8) {
    aiccs[p+1,q+1] = AICc(arima(bcTran, order = c(p,1,q), method="ML",
      optim.control = list(maxit = 1000)))
  }
}
aiccs

#Fits all ARIMA(p,1,q) models for p = 3, 5, 6, 15, 30 and q = 5, 6, 20
#Then prints out all of the respective AICC
aiccs <- matrix(NA, nr = 4, nc = 2)
dimnames(aiccs) = list(p=c(3, 5, 6, 15), q=c(5, 6))
p <- c(3, 5, 6, 15)
q <- c(5, 6)
for(i in 1:4) {
  for(j in 1:2) {
    aiccs[i,j] = AICc(arima(bcTran, order = c(p[i], 1, q[j]), method="ML",
      optim.control = list(maxit = 1000)))
  }
}
aiccs
first <- arima(bcTran, order = c(5, 1, 20), method="ML", optim.control = list(maxit = 1000))
second <- arima(bcTran, order = c(6, 1, 20), method="ML", optim.control = list(maxit = 1000))
third <- arima(bcTran, order = c(30, 1, 5), method="ML", optim.control = list(maxit = 1000))
fourth <- arima(bcTran, order = c(30, 1, 6), method="ML", optim.control = list(maxit = 1000))
fifth <- arima(bcTran, order = c(30, 1, 20), method="ML", optim.control = list(maxit = 1000))

paste('p = 5, q = 20, AICC:', AICc(first))
paste('p = 6, q = 20, AICC:', AICc(second))
paste('p = 30, q = 5, AICC:', AICc(third))
paste('p = 30, q = 6, AICC:', AICc(fourth))

```



```

paste('p = 30, q = 20, AICC:', AICc(fifth))

#Fits ARIMA(6, 1, 5) models with all variations of
#phi1, phi2, phi4, theta1, theta2, theta4, and theta5 being fixed to zero.
#Then it prints out the model whose coefficients give the lowest AICC value
parList <- c(0, NA)
min <- list(10000)
  for(j in 1:2){
    for(k in 1:2){
      for(h in 1:2){
        for(m in 1:2){
          for(n in 1:2){
            for(s in 1:2){
              for(f in 1:2){
                mlVal <- AICc(arima(bcTran, order=c(6,1,5),
                                   fixed = c(parList[j], parList[k], NA, parList[h], NA,
                                              NA, parList[m], parList[n], NA, parList[s], parList[f]),
                                   method="ML", optim.control = list(maxit = 1000)))

                if(min[1] > mlVal){
                  min <- list(mlVal, j, k, h, m, n, s, f)
                }
              }
            }
          }
        }
      }
    }
  }

#Prints the coefficients of the fit with lowest AICC value
fit2b

#Code that checks to see if the roots of AR and MA component are outside of the unit circle
par(mfrow=c(1,2))
#Checking AR component
uc.check(pol_ = c(1, 0, 0, -0.7148, 0.3156, 0.2840, 0.1465),
         print_output = F)

#Checking MA component
uc.check(pol_ = c(1, 0, 0, -0.7673, 0.2683, 0.1488),
         print_output = F)

#Finds the residuals of my final fit and plots

```

```

#a time series of these values as well as the ACF and PACF graphs
res <- residuals(final.fit)
paste('Mean:', mean(res), 'Var:', var(res))

layout(matrix(c(1,1,2,3),2,2,byrow=T))
ts.plot(res,main = "Fitted Residuals")
t = 1:length(res)
fit.res = lm(res~t)
abline(fit.res)
abline(h = mean(res), col = "red")
# acf
acf(res,main = "Autocorrelation")
# pacf
pacf(res,main = "Partial Autocorrelation")

#Box-Pierce test for independence of residuals
Box.test(res, lag = 19, type = c("Box-Pierce"), fitdf = 7)

#Ljung-Box test to see if autocorrelations are different than 0
Box.test(res, lag = 19, type = c("Ljung-Box"), fitdf = 7)

#McLeod-Li tests
Box.test(res^2, lag = 19, type = c("Ljung-Box"), fitdf = 0)
McLeod.Li.test(final.fit)

#Shapiro-Wilk test for normality of residuals
shapiro.test(res)

#draws a histogram of residuals
hist(res,main = "Histogram")

#draws a q-q plot
qqnorm(res)
qqline(res,col ="red")

#Graphs a 10 step ahead prediction with a confidence interval around predicted points
mypred <- predict(final.fit, n.ahead=10)
ts.plot(ts.eRate, xlim=c(300,366), ylab = 'USD to Euro Exchange Rate')
points(356:365, sqrt(1/(((mypred$pred)*-2) + 1)))
lines(356:365, sqrt(1/(((mypred$pred+1.96*preds$se)*-2) + 1)), lty=2)
lines(356:365, sqrt(1/(((mypred$pred-1.96*preds$se)*-2) + 1)),lty=2)

#Graphs a 10 step ahead prediction with a confidence interval around predicted points,
#along with the line of true values
mypred <- predict(final.fit, n.ahead=10)

```

```

ts.plot(eRate, xlim=c(300,366), ylab = 'USD to Euro Exchange Rate')
points(356:365, sqrt(1/(((mypred$pred)*-2) + 1)))
lines(356:365, sqrt(1/(((mypred$pred+1.96*preds$se)*-2) + 1))), lty=2)
lines(356:365, sqrt(1/(((mypred$pred-1.96*preds$se)*-2) + 1))), lty=2)

```