Forecasting Auto Registrations in California

Wilber Delgado

2023-12-13

Abstract Summary

The foundation of my project stems from a Motor Vehicle Registration data set provided by data.gov. From this data set i decided that it would be interesting to study and attempt to forecast the motor vehicle registrations in California, as I am a California native. The primary objective is to gain insights into how the number of street-legal cars has evolved over the years, and to attempt to discover patterns that can be effectively forecasted. For this Time series analysis will be a key component in identifying patterns and possible models that my data set follows, which can then be used to predict the following few years after.

Report

Starting off, I began with the data set “Motor Vehicle Registrations Dashboard data”, derived from data.gov. When I first read in the data, I noticed that the original dataset contained 6 columns; year, state, Auto, Bus, Truck, and Motorcycle. The years ranged from 1900 to 2020, and contained annual data of every state. Since I was more interested in my native state of California and regular automobiles, I derived a subset of this data that only contained the Auto values from California. Using this I plotted the values to check if there would be any problems that I could address from the beginning. I found that there was a major decline in around 2010 which could have a significant impact on the forecasting. Even after using Box-Cox transformation the major decline was too great to keep in my data set so I decided to take a subset from 1900 to 2009 to minimize any forecasting issues. **Appendix pages 3-6**

Following this I was to plot my new dataset and its respective ACF and PACF, and noticed that there was an unstable variance. To address this I decided I would use the BoxCox transformation to see if it would fix it. After this it seemed that the variance issue was solved, but my data was still not stationary as is required for creating models for forecasting, so to address this I decided to differentiate the BoxCox transformed dataset at lag 1. Looking at the plot, ACF,m and PACF i could see that it did help make it more stationary. It seemed as though it could possibly be more stationary than it already was, but after I differed a second time at lag 1, the variance increased, which is a sign of over differencing so I continued with the data that was only different once. **Appendix pages 7-14**

After transforming and differencing my data to make it as stationary as possible, I plotted the ACF and PACF,which now did portray stationarity, of it to determine possible models to fit. Looking at the PACF I can observe a strong positive spike at lag 1, with subsequent partial autocorrelations being within the confidence interval. In the ACF I could also see a strong positive spike at lag 1, with subsequent autocorrelations being within the confidence interval. Based on these charts, it could be possible for p=1 and q=1, so to test these I proceeded to test an AR(1) and ARMA(1,1) model. **Appendix pages 15-16**

After fitting both the AR(1) and ARMA(1,1) model I found that the AR(1) model had a lower AICc than the ARMA(1,1) indicating that it could be a better fit. Out of interest I decided to do a diagnostics check with the ARMA(1,1) model. From the plotted roots I could see that the model was invertible and stationary. Also looking at the histogram and QQ plot of the residuals it did seem to be as normal as possible considering that my original

data was only 109 points. After doing the Shapiro Wilk, Box-Pierce, Ljung Box, and Mcleod Li test I found that all the p values were larger than 0.05 which means it passed them and was a good fit for the data. I then proceeded to do a diagnostics test on the AR(1) model. Since the $|\phi_1| < 1$ AR(1) model is invertible and stationary. The histogram and QQ plot both were reasonably normal for the size of the data set. The AR(1) model also did pass the Shapiro Wilk, Box-Pierce, Ljung Box, and Mcleod Li test with them all having greater p values than 0.05. Also the Yule Walker test showed the order selected being 0 which shows no significant autocorrelation in the residual, which also reflects the ACF and PACF of the residuals for the AR(1) model. Considering both of my chosen models seemed like good fits for my data, I decided to stick with the AR(1) model for forecasting because it has a lower number of estimated values and since its AICc was lower than that of the ARMA(1,1) model. **Appendix pages 16-31**

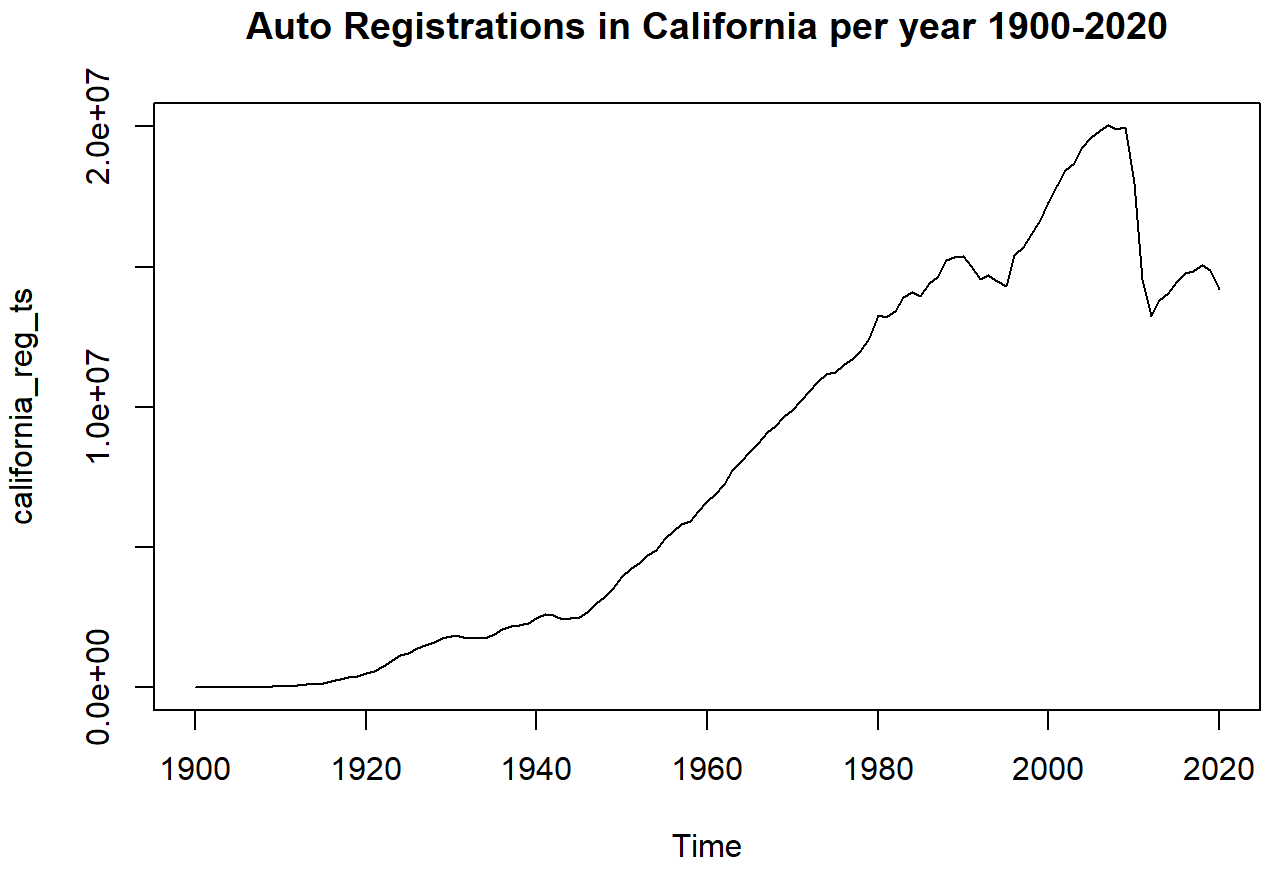
Furthermore, after choosing the AR(1) model, $(1 - 0.4843B)(1 - B)(X_t) = Z_t$ and $X_t = \nabla_1 U_t^{0.403}$ where U_t is the original data, I began the process of forecasting by creating a fit with the Box-Cox transformed dataset with p=1 and i=1. After I plotted the predicted points and prediction interval which indeed seemed to match the original Box-Cox transformed data. Next, I plotted the prediction on the Box-Cox transformed data, and the results showed to be fairly accurate the true points were within the confidence interval. **Appendix pages 31-32**

After this I plotted the prediction model to the training set, with my original data set plotted for comparison. The predicted points were a little lower than the original data set line, but they were still within the prediction interval. In the zoomed-in visualization of forecasting on the testing set I could see that the first two points were on the actual line and the following ones were just a little below. **Appendix pages 32-34**

Overall, I was able to take the data set of Motor Vehicle Registrations from data.gov and format the data set to show only my variables of interest which is Auto Registrations in California from 1900-2020 to create an AR(1) model, $(1 - 0.4843B)(1 - B)(X_t) = Z_t$ and $X_t = \nabla_1 U_t^{0.403}$ where U_t is the original data, Ultimately my model performs well in forecasting for future California Auto Registrations. Finally, I would like to thank Professor Raya Feldman for the help and guidance she offered me when working on this project.

Appendix

```
setwd("D:/")
vehicle_reg = read.csv("Motor_Vehicle_Registrations_Dashboard_data.csv")

# Filter rows where the state is "California" and select only "year" and "Auto" columns
california_reg <- subset(vehicle_reg, state == "California", select = c("year", "Auto"))

california_reg_ts <- ts(california_reg$Auto, start = c(1900, 1), frequency = 1)

# Plot the time series from 1900 to 2020
ts.plot(california_reg_ts, main="Auto Registrations in California per year 1900-2020")
```



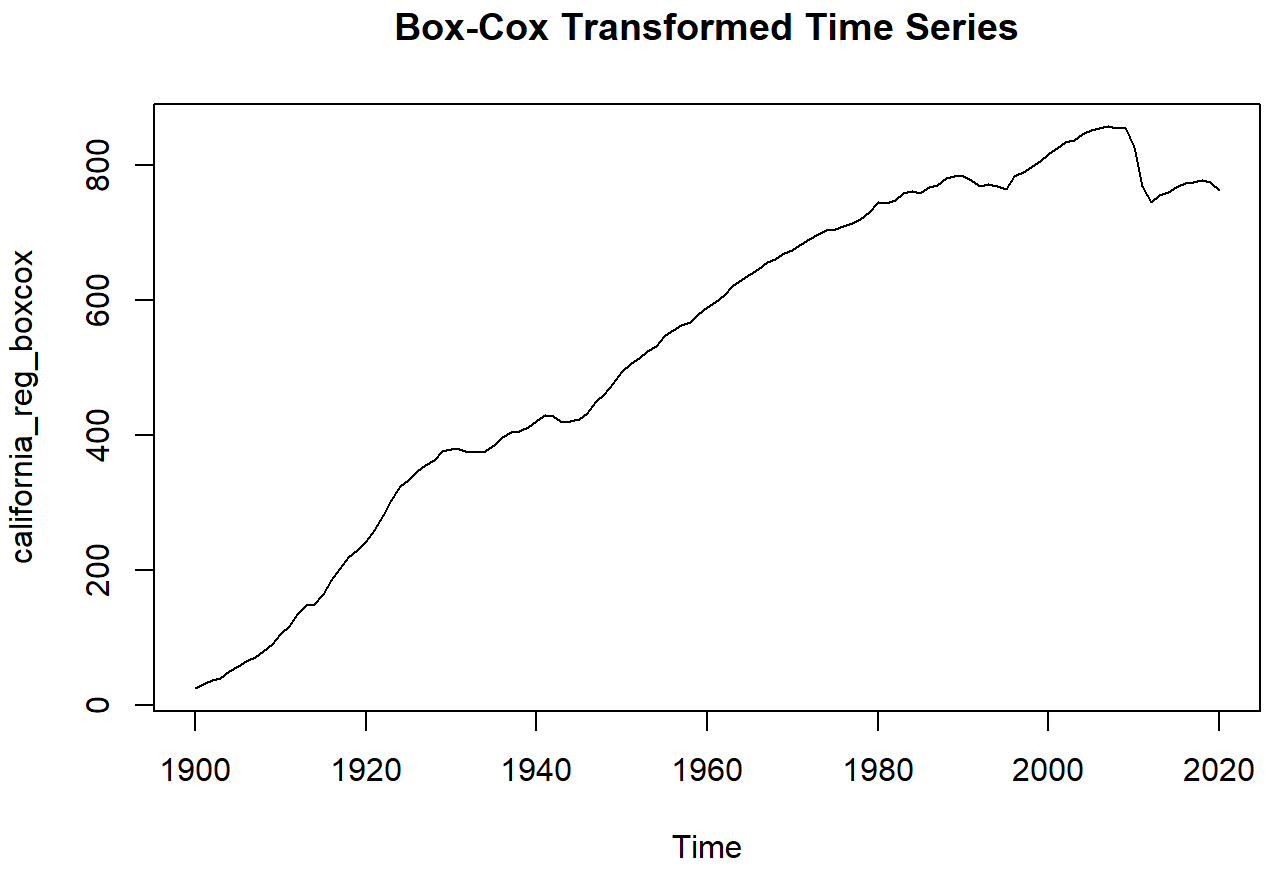
Here I plot the Auto Registrations in California, and see the big dip in 2010. I will see if Box-Cox will help with that big dip.

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.3.2
```

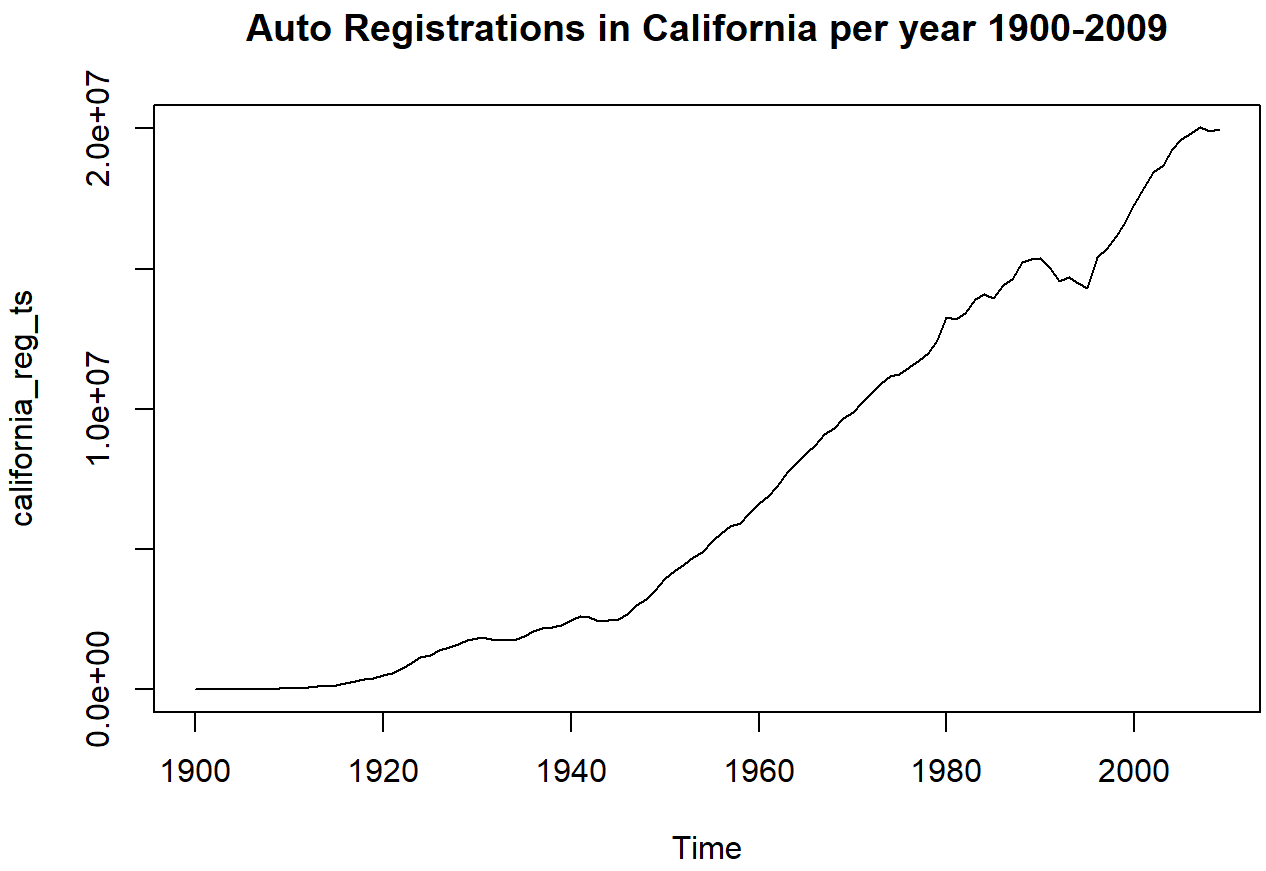
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
# Apply Box-Cox transformation
california_reg_boxcox <- BoxCox(california_reg_ts, lambda = "auto")

# Plot the transformed time series
plot(california_reg_boxcox, main = "Box-Cox Transformed Time Series")
```



The box cox transformation seems to have made the plot exponential up until 2010. Therefore I will work with 1900-2009 to avoid irregularities caused by the significant dip in 2010.

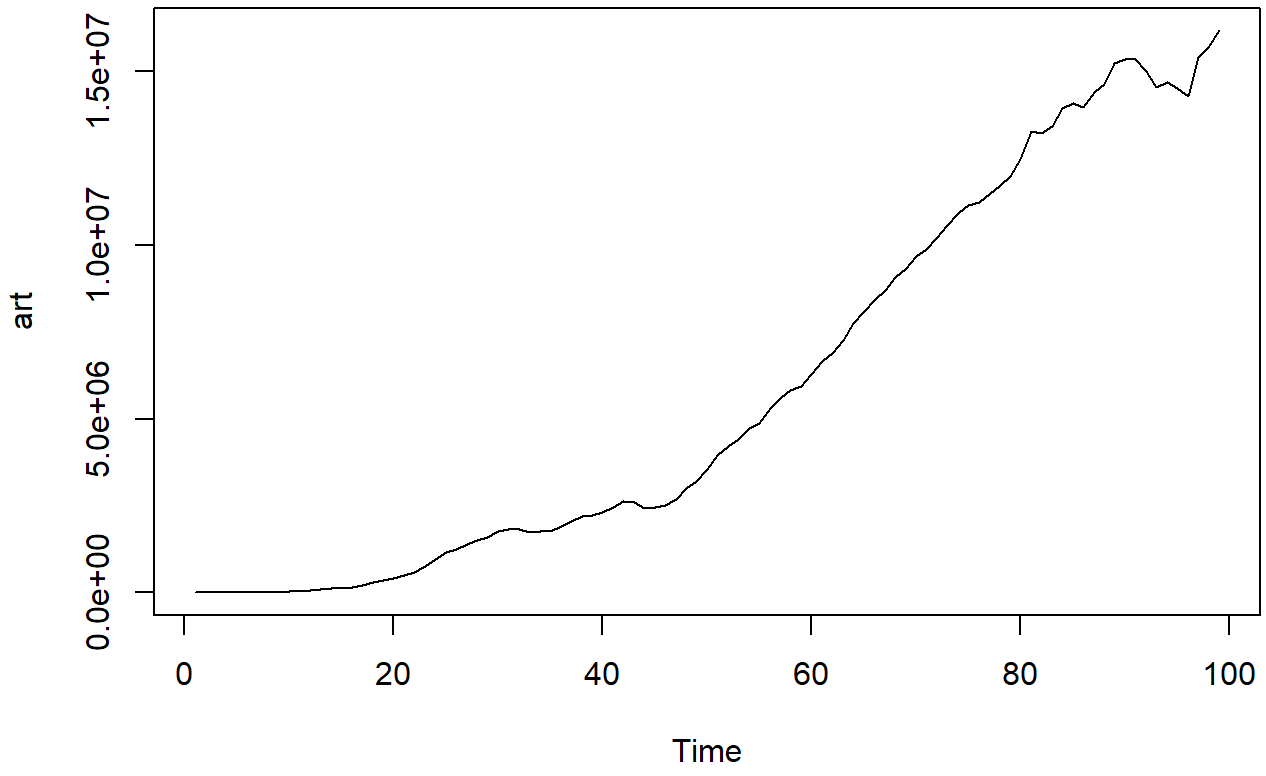
```
# Filter the data to include only the years from 1900 to 2000
california_reg_subset <- subset(california_reg, year >= 1900 & year <= 2009)

# Keep only the 'Auto' column
california_reg_subset <- california_reg_subset$Auto

# Create a time series from the subset
california_reg_ts <- ts(california_reg_subset, start = c(1900, 1), frequency = 1)

# Plot the time series from 1900 to 2000
ts.plot(california_reg_ts, main = "Auto Registrations in California per year 1900-2009")
```



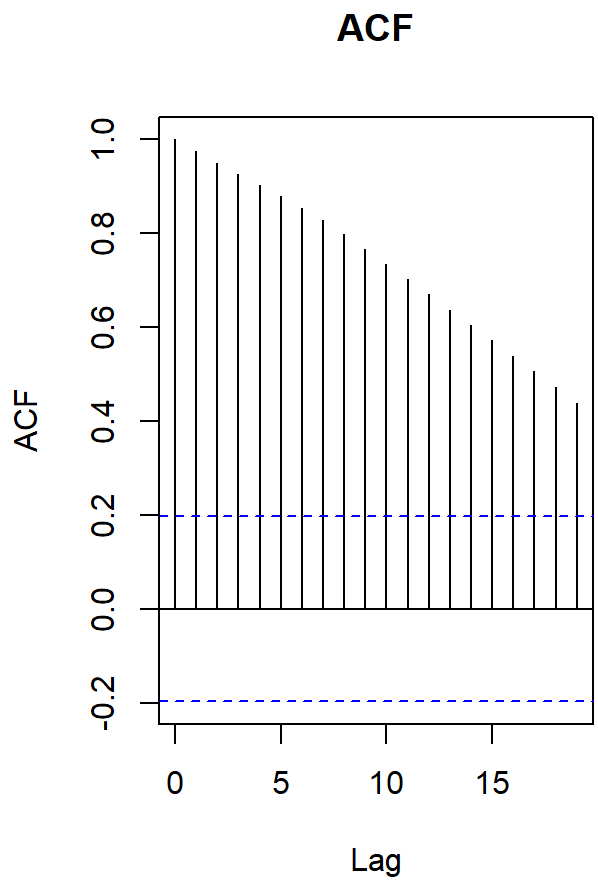
The plot looks much better now. Now I will create a training and testing set.

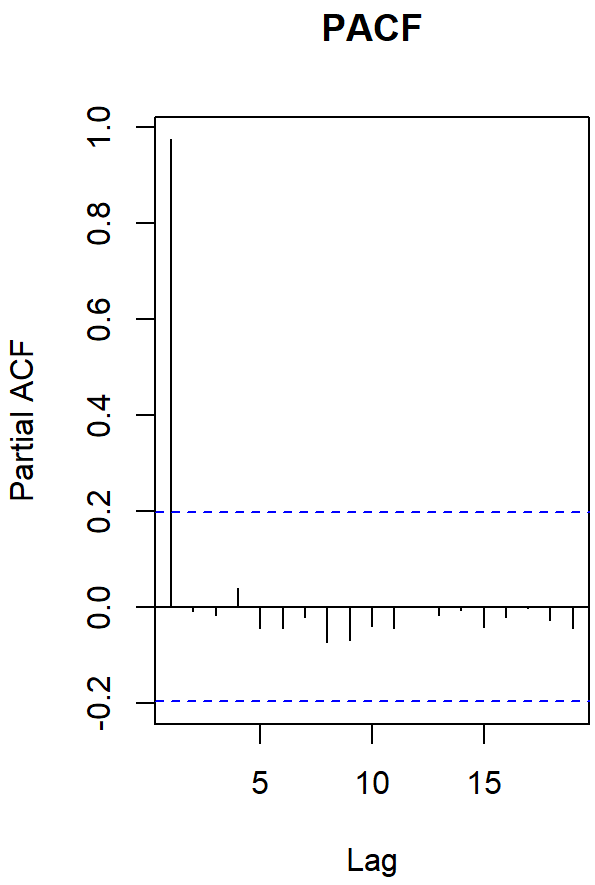
```
art = california_reg_subset[c(1:99)]
artest = california_reg_subset[c(100:110)]
```

```
plot.ts(art)
```

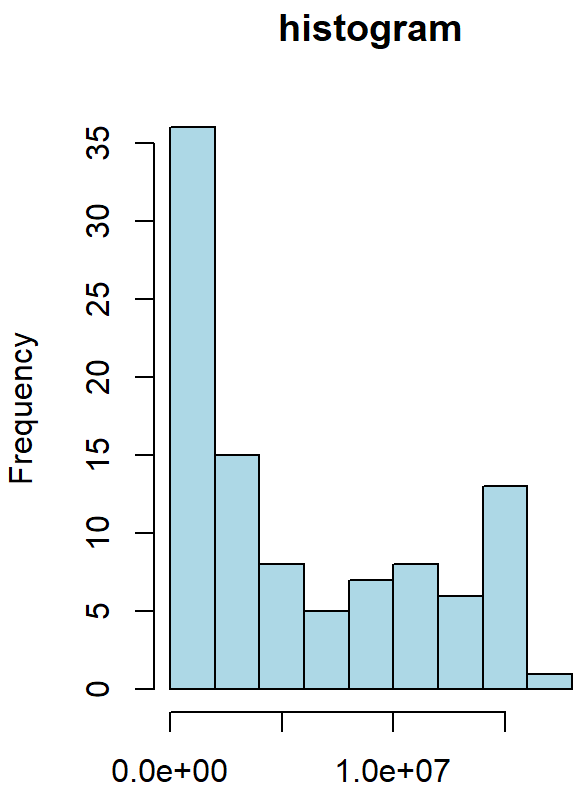



```
# Perform and plot ACF and PACF
par(mfrow = c(1, 2))
acf(art, main = "ACF")
pacf(art, main = "PACF")
```



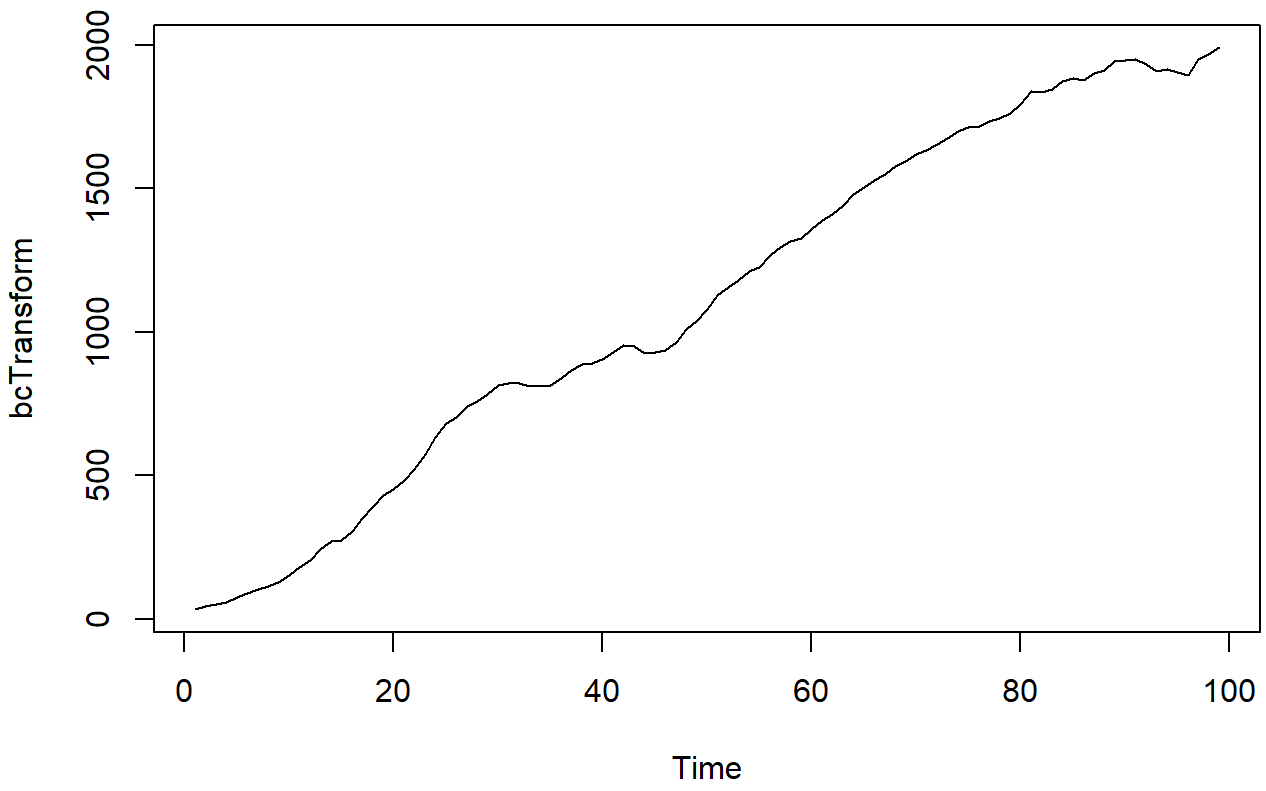



```
hist(art, col="light blue", xlab="", main="histogram")
```



There are some variance issues that can possibly be fixed by Box-Cox.

```
bcTransform <-  BoxCox(art, lambda = "auto")
plot.ts(bcTransform)
```

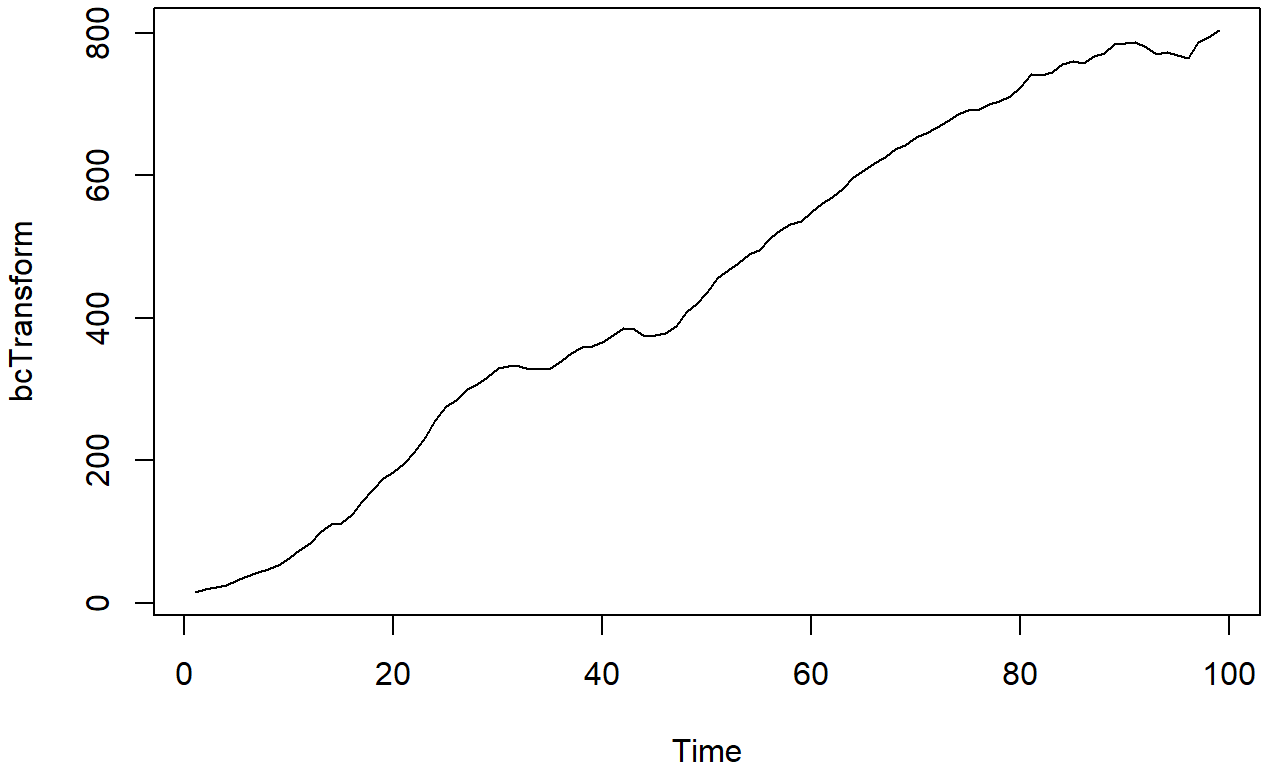



```
#print lambda
attr(bcTransform, "lambda")
```

```
## [1] 0.4029525
```

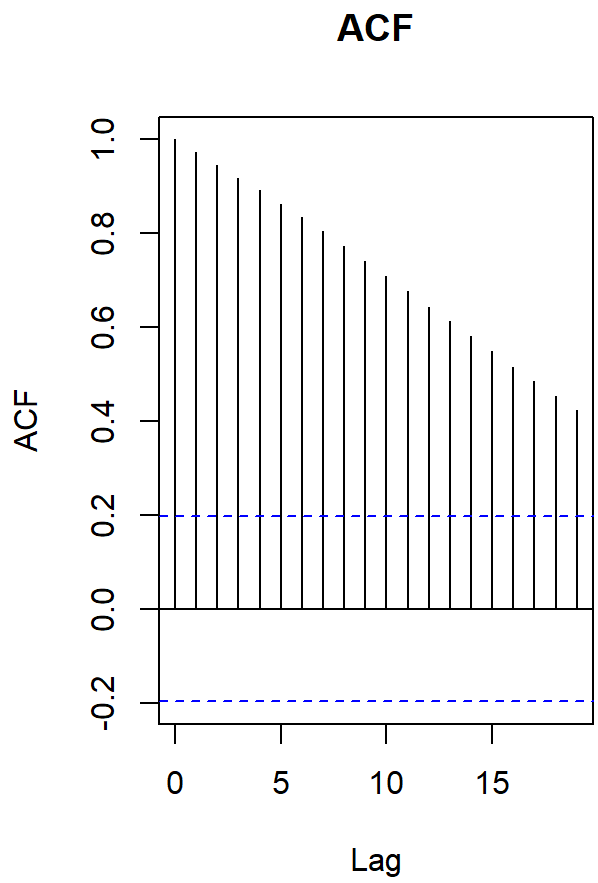
Box-Cox transformation did help and lambda = 0.403.

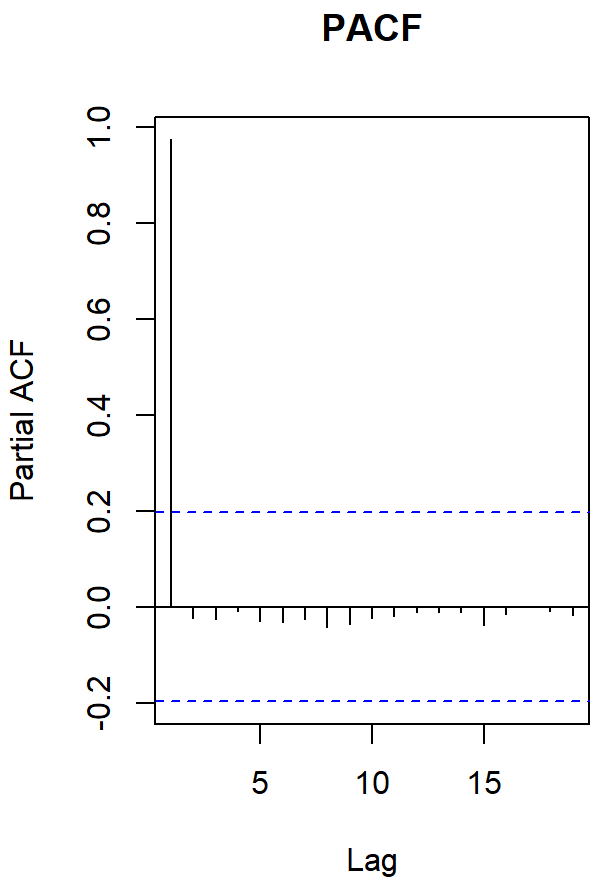
```
bcTransform <-  art^0.403
plot.ts(bcTransform)
```



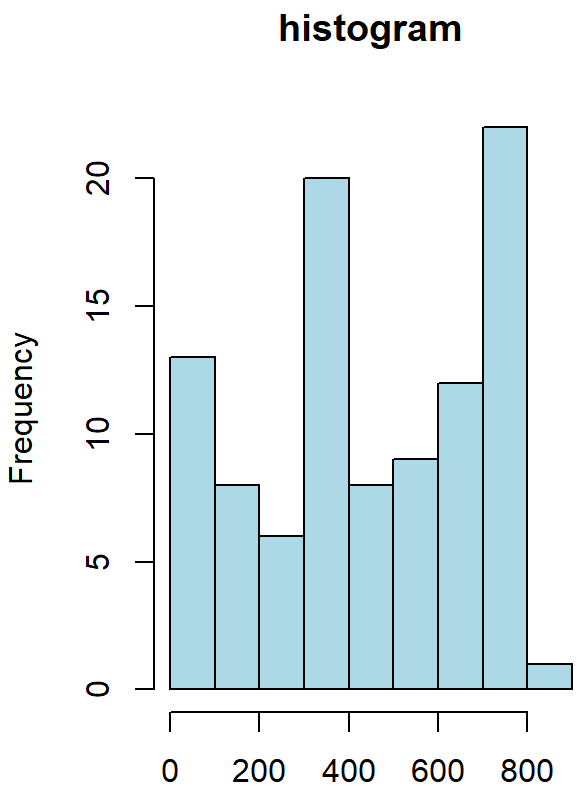
Since lambda = 0.4029525, we know that $U_t = \frac{U_t^{0.403} - 1}{0.403}$

```
# Perform and plot ACF and PACF
par(mfrow = c(1, 2))
acf(bcTransform, main = "ACF")
pacf(bcTransform, main = "PACF")
```






```
hist(bcTransform, col="light blue", xlab="", main="histogram")
```

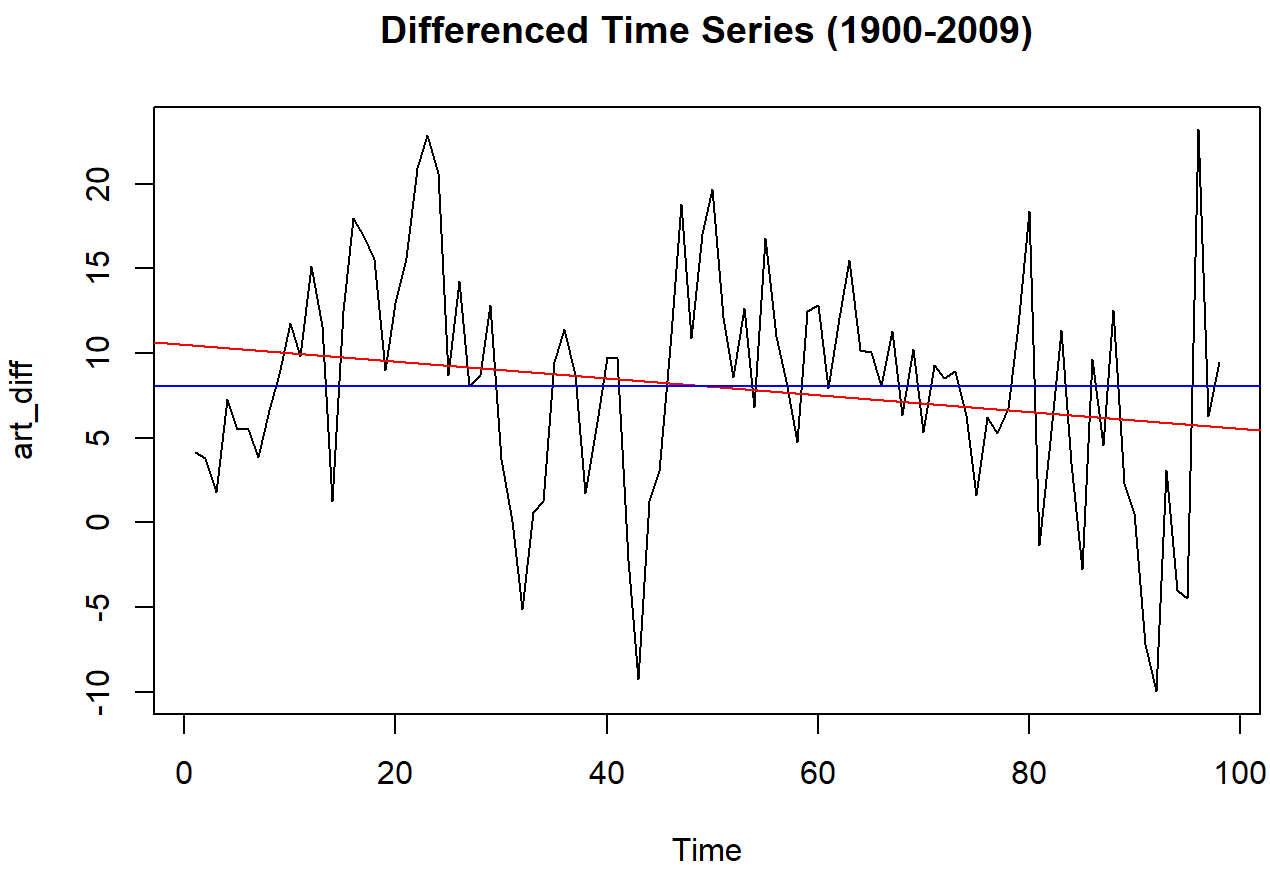


Variance issue seems better, now I have to difference to make it stationary.

```
# Difference to make the time series stationary
art_diff <- diff(bcTransform, lag = 1)

# Plot the differenced time series
ts.plot(art_diff, main = "Differenced Time Series (1900-2009)")
fit <- lm(art_diff ~ as.numeric(1:length(art_diff))); abline(fit, col="red")
mean(art_diff)
```

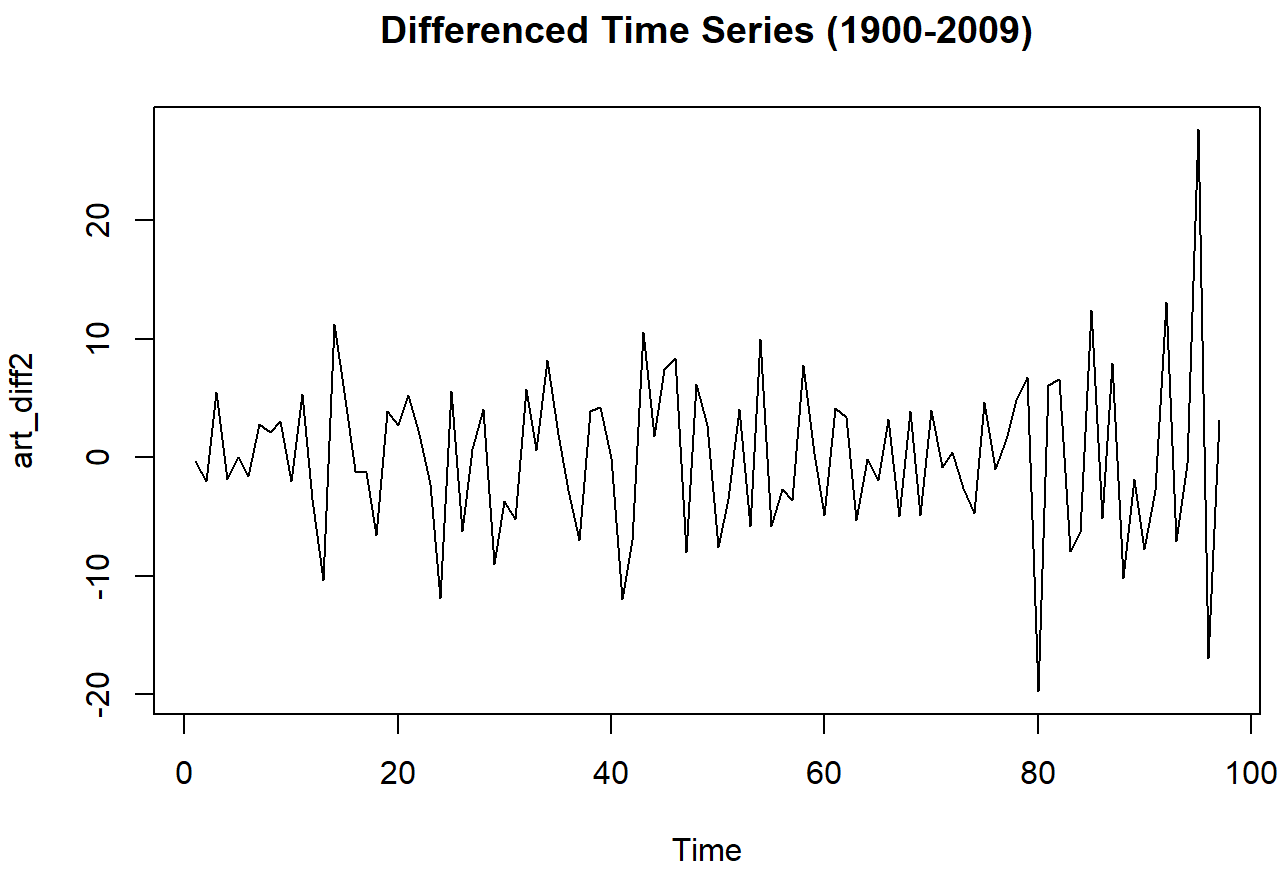
```
## [1] 8.05281
```

```
abline(h=mean(art_diff), col="blue")
```



Differencing once made it more stationary. Maybe it can be even more stationary so I will try to difference again at lag 1.

```
# Difference to make the time series stationary
art_diff2 <- diff(art_diff, lag = 1)

# Plot the differenced time series
ts.plot(art_diff2, main = "Differenced Time Series (1900-2009)")
```




```
# Variance when differenced at lag 1
cat("Variance when differenced at lag 1: ", var(art_diff), "\n")
```

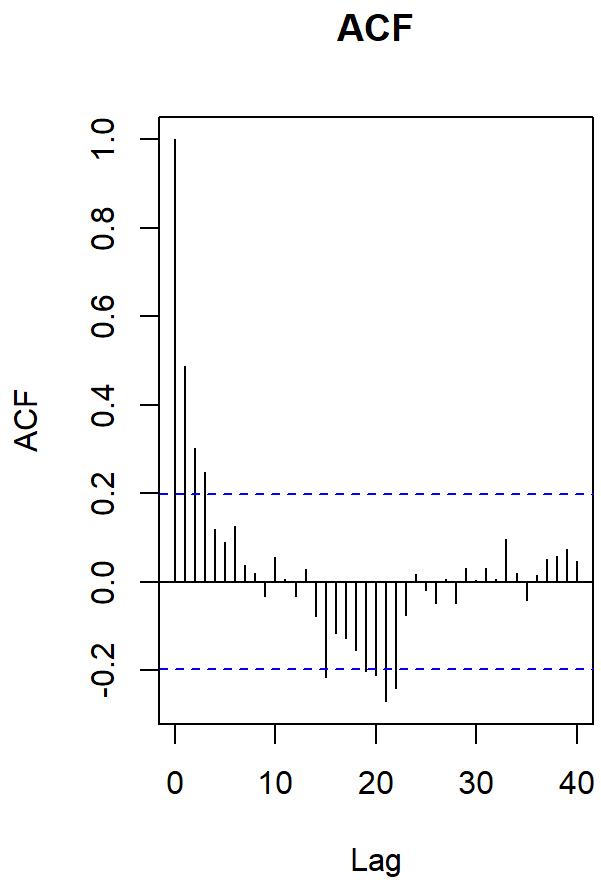
```
## Variance when differenced at lag 1:  45.09288
```

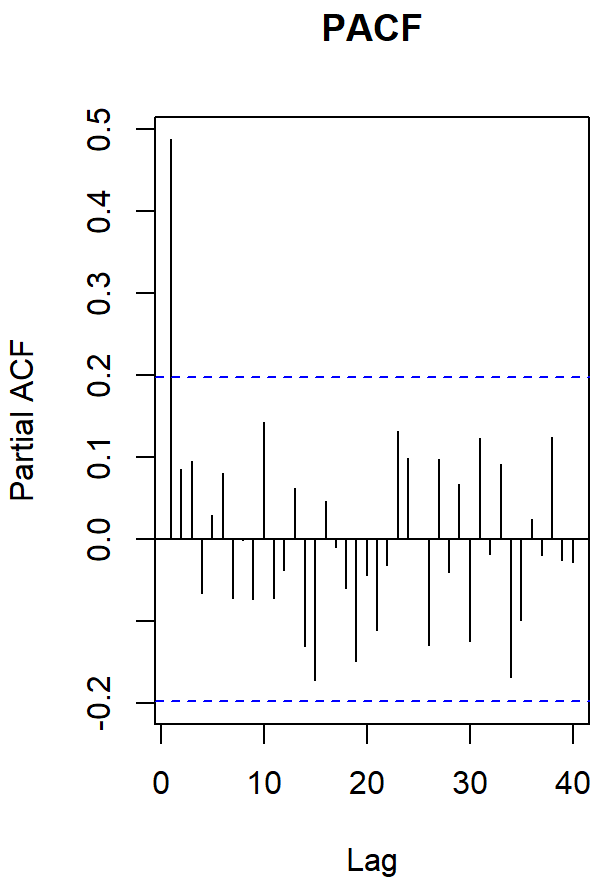
```
# Variance when differenced twice at lag 1
cat("Variance when differenced twice at lag 1: ", var(art_diff2), "\n")
```

```
## Variance when differenced twice at lag 1:  46.54215
```

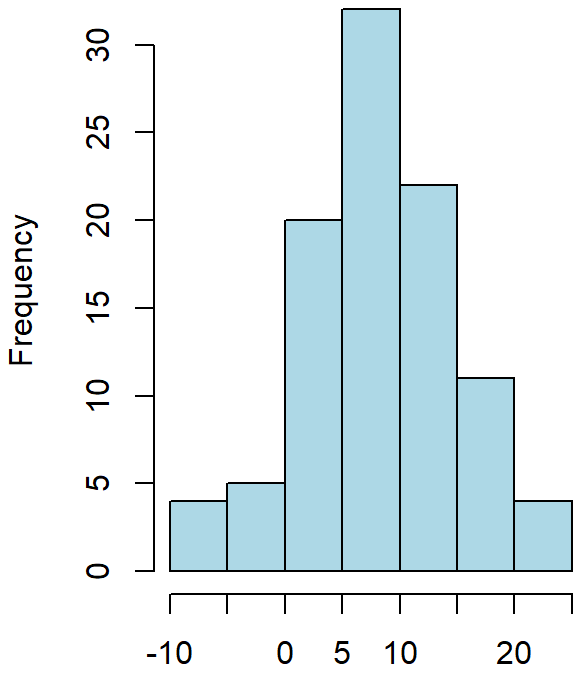
Differencing again at lag 1 made the variance go up, which is a sign of over-differencing, so I will stick with only differencing once.

```
# Perform and plot ACF and PACF
par(mfrow = c(1, 2))
acf(art_diff, lag.max = 40 ,main = "ACF")
pacf(art_diff,lag.max = 40, main = "PACF")
```






```
hist(art_diff, col="light blue", xlab="", main="BoxCox(U_t^0.403) differenced at lag 1")
```



Considering the size of my dataset, this is a fairly normal histogram. The ACF and PACF look good and possible values of p=1 and q=1 so i will test ARMA(1,1) and AR(1).

```
library(qpcR)
```

```
## Loading required package: MASS
```

```
## Loading required package: minpack.lm
```

```
## Loading required package: rgl
```

```
## Loading required package: robustbase
```

```
## Loading required package: Matrix
```

```
# Fit ARMA(1,1) model
arma_model <- arima(art_diff, order = c(1, 0, 1))

# Print model summary
arma_model
```

```
##
## Call:
## arima(x = art_diff, order = c(1, 0, 1))
##
## Coefficients:
##          ar1      ma1  intercept
##       0.6559  -0.2299     8.0087
## s.e.  0.1584   0.2089     1.2905
##
## sigma^2 estimated as 33.63:  log likelihood = -311.46,  aic = 630.92
```

```
#print AICc (Lab 6)

AICc(arma_model)
```

```
## [1] 631.175
```

$X_t - 0.656(X_{t-1}) = Z_t - 0.2299(Z_{t-1})$ $(1 - 0.656B)X_t = (1 - 0.2299B)Z_t$

and $X_t = U_t^{0.403}$ where U_t is the original data.

```
# Fit AR(1) model
ar1_model <- arima(art_diff, order = c(1, 0, 0))

# Print model summary
ar1_model
```

```
##
## Call:
## arima(x = art_diff, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.4842     8.0294
## s.e.  0.0876     1.1312
##
## sigma^2 estimated as 34:  log likelihood = -311.97,  aic = 629.95
```

```
AICc(ar1_model)
```

```
## [1] 630.0755
```

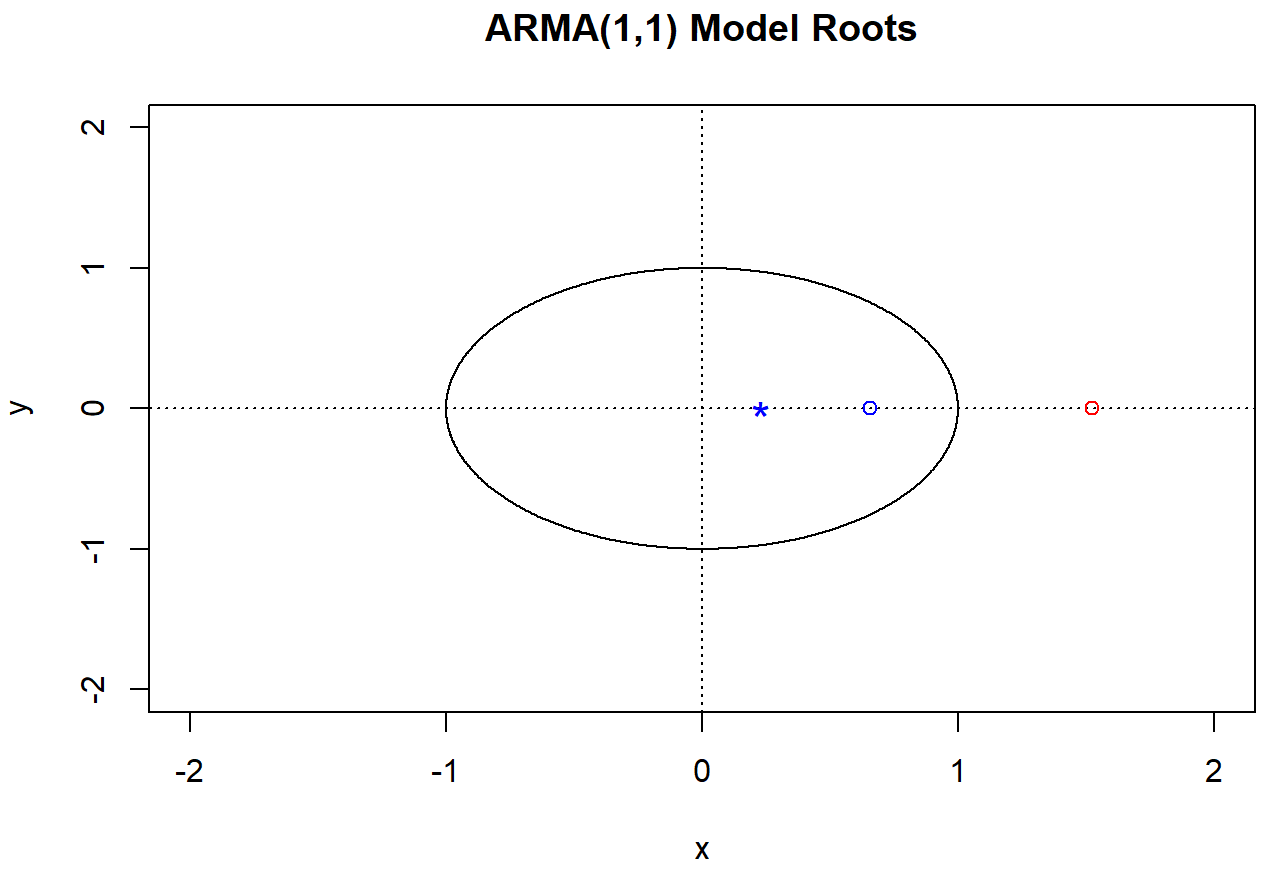
$X_t = 0.4843(X_{t-1} + Z_t)$ $(1 - 0.4843B)(1 - B)(X_t) = Z_t$

and $X_t = U_t^{0.403}$ where U_t is the original data.

The AICc for AR(1) model is lower than ARMA(1,1), indicating it might be a better fit. I will run diagnotics testing on ARMA(1,1) first and then on AR(1), and if they both pass I will stick with the AR(1) model.

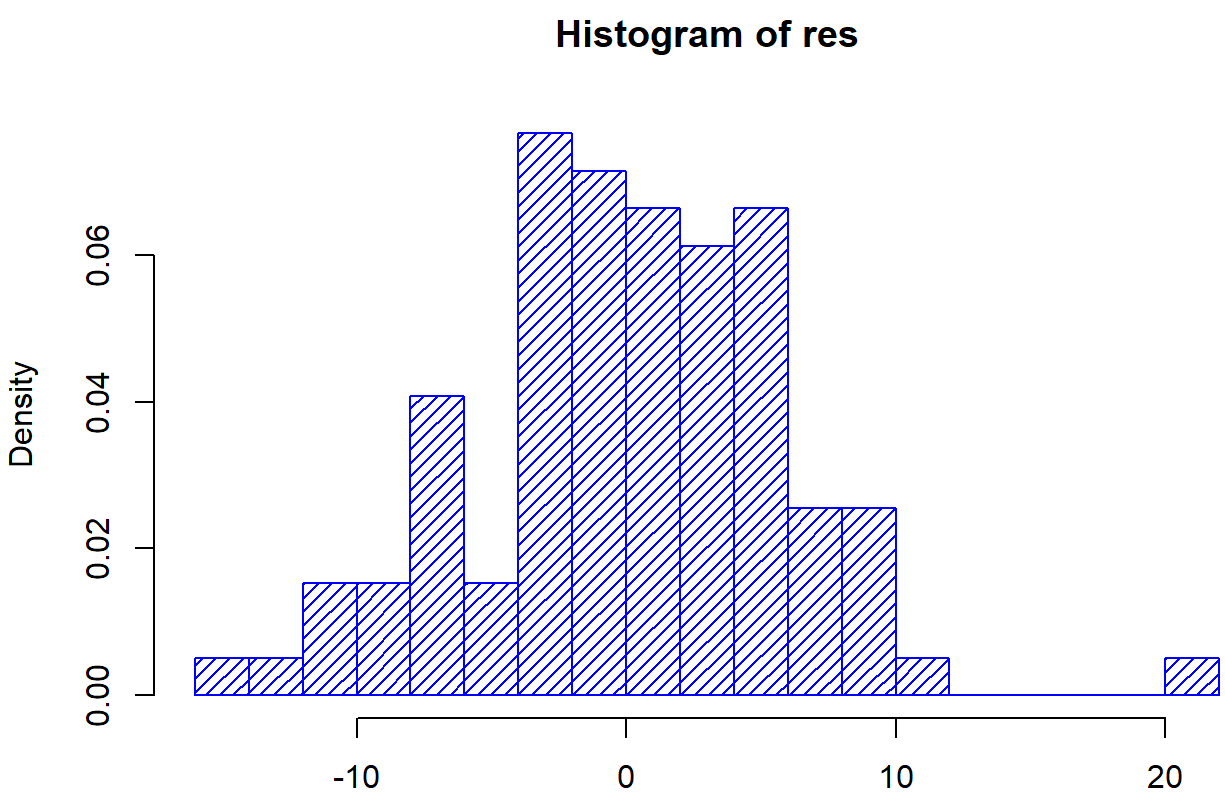
ARMA(1,1) diagnostics testing:

```
source("plot.roots.R")
# AR and MA coefficients
ar_coef <- c(0.656)
ma_coef <- c(0.2299)

# Calculate AR and MA roots
ar_roots <- polyroot(c(1, -ar_coef))
ma_roots <- polyroot(c(1, -ma_coef))

# Plot AR and MA roots
plot.roots(ar.roots = ar_roots, ma.roots = ma_roots, main="ARMA(1,1) Model Roots")
```



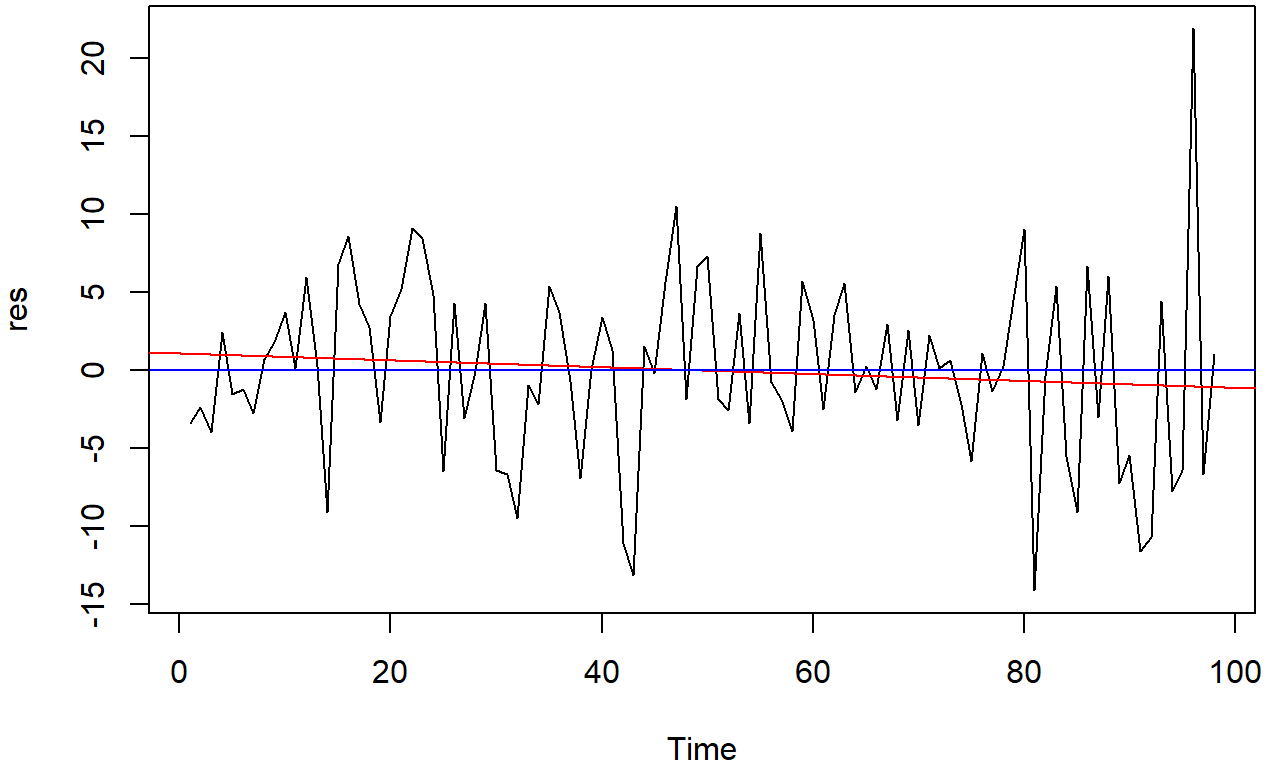
Roots pass.

```
#ARMA(1,1)
fit <- arima(art_diff, order=c(1,0,1), method="ML")
res <- residuals(fit)
hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
```



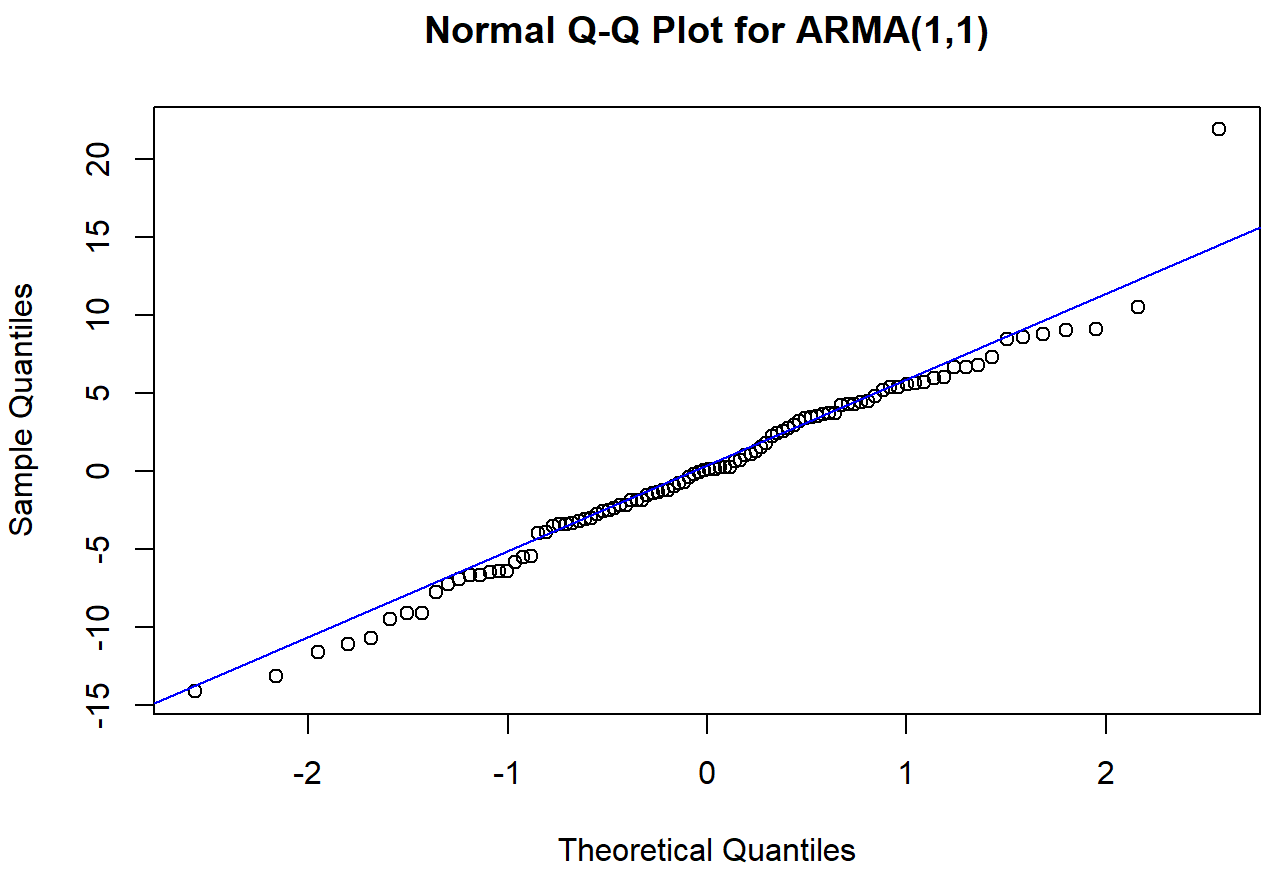
Histogram looks normal.

```
m <- mean(res)
std <- sqrt(var(res))
plot.ts(res)
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
```



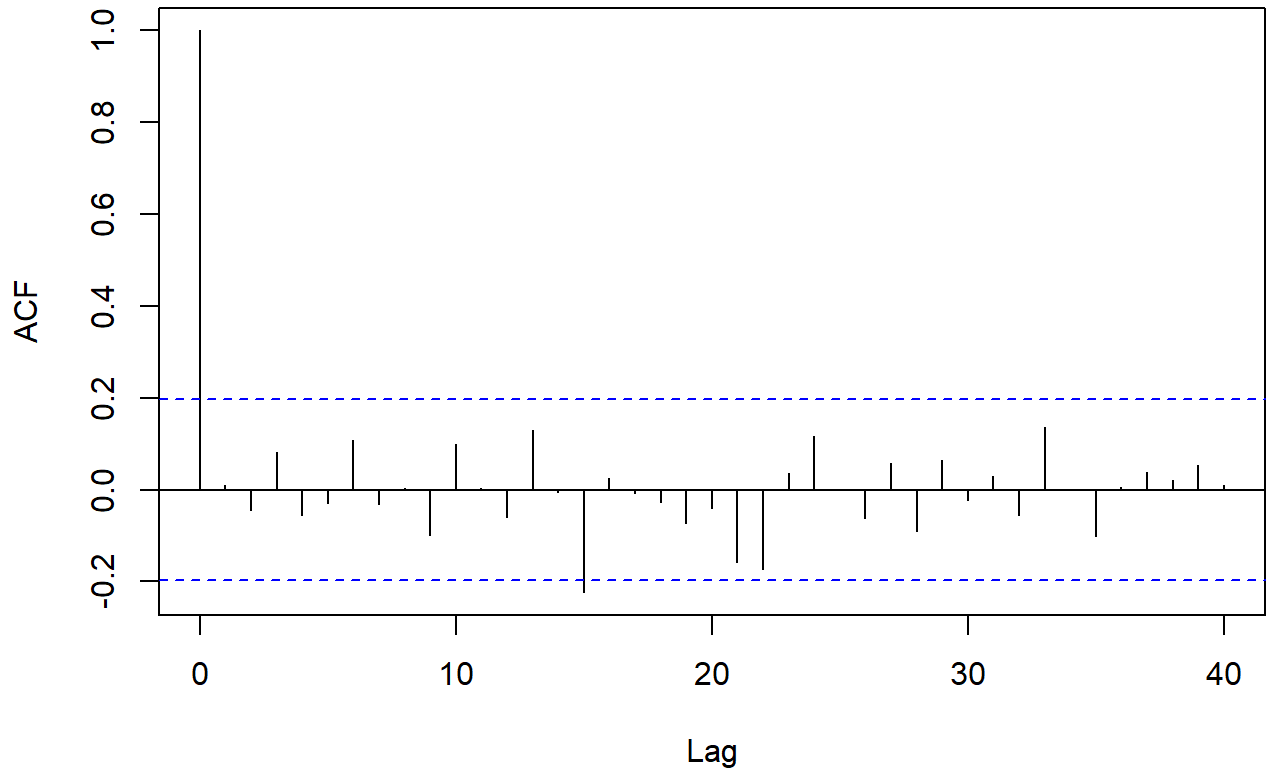
Residuals seem statioanry.

```
qqnorm(res,main= "Normal Q-Q Plot for ARMA(1,1)")
qqline(res,col="blue")
```

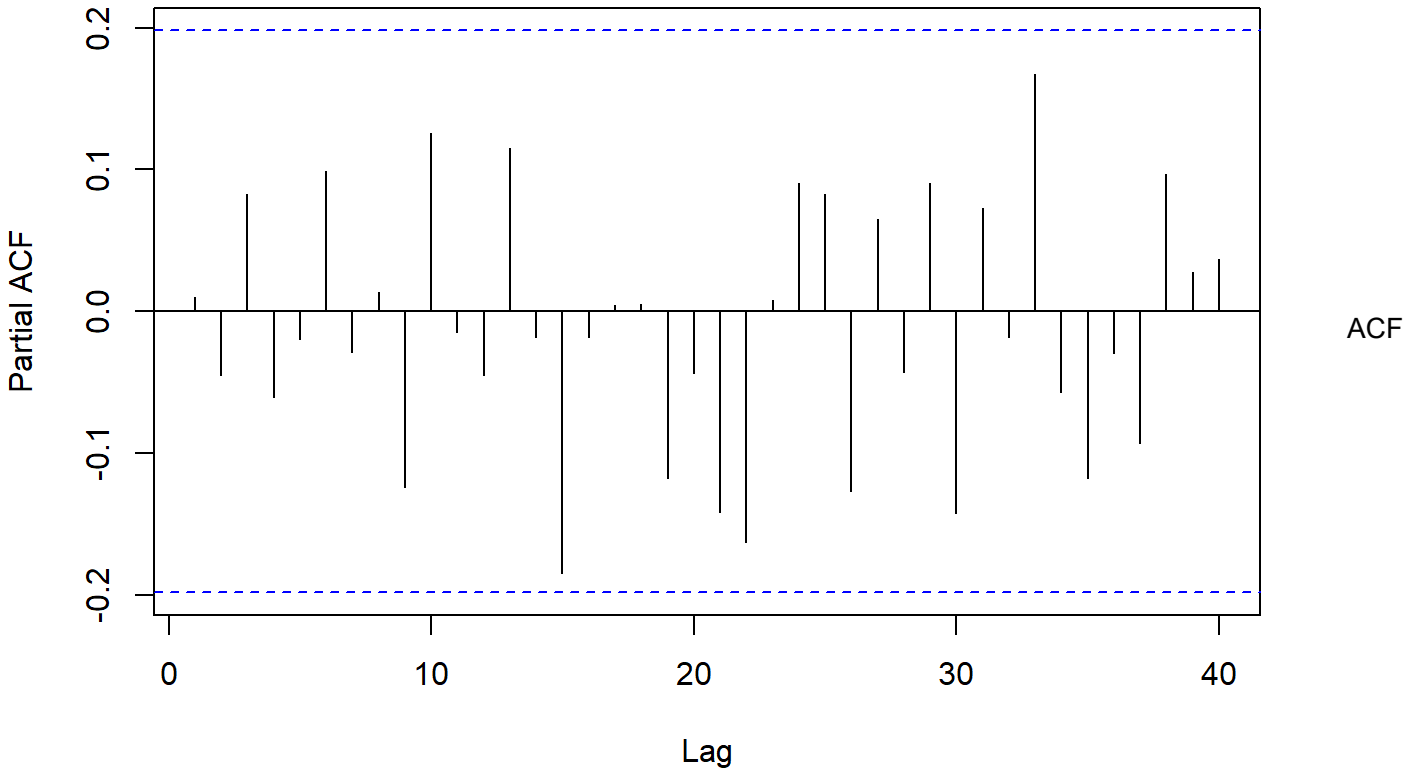


QQ plot looks normal.

```
acf(res, lag.max=40)
```




```
pacf(res, lag.max=40)
```



and PACF look fine as well, no irregularities.

```
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.97931, p-value = 0.1252
```

```
Box.test(res, lag = 10, type = c("Box-Pierce"), fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 4.4376, df = 8, p-value = 0.8156
```

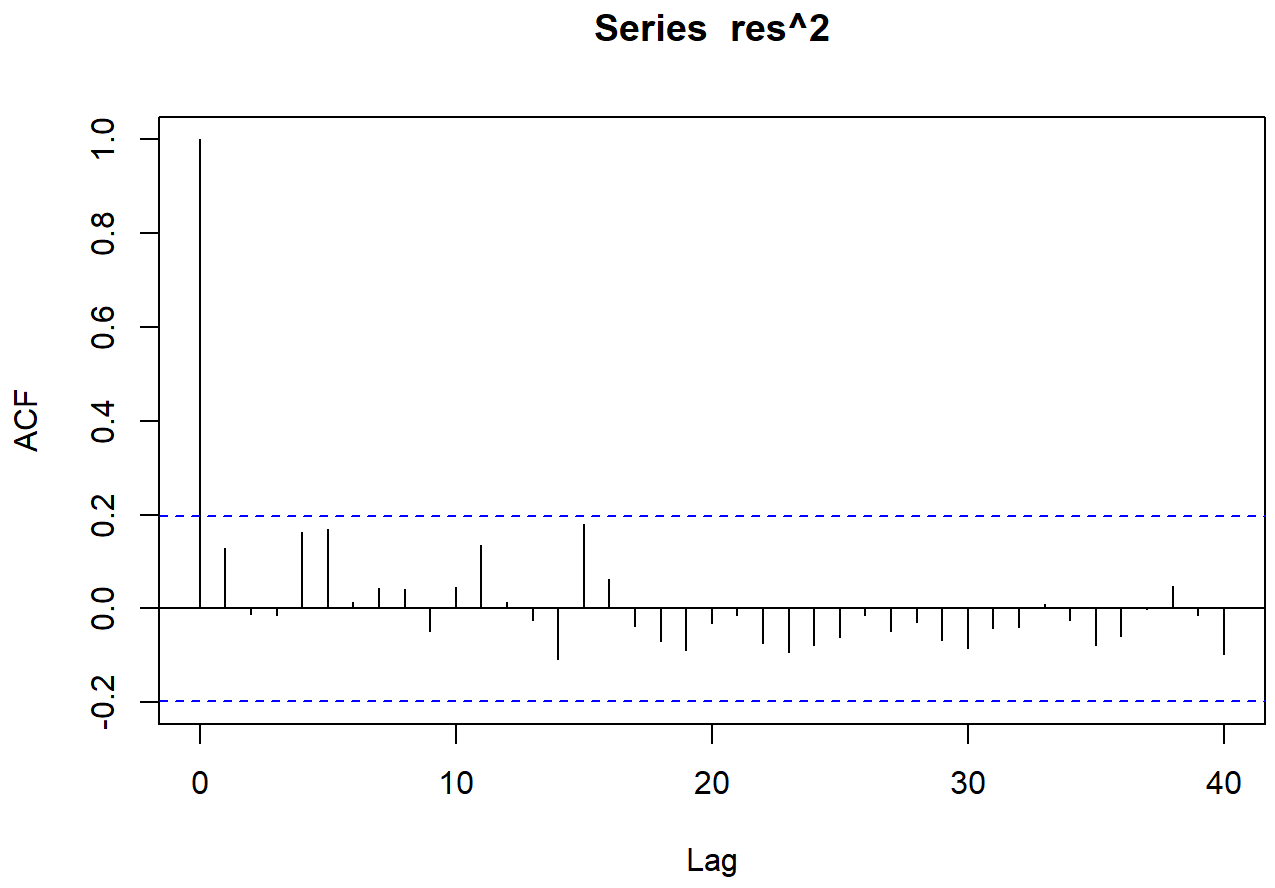
```
Box.test(res, lag = 10, type = c("Ljung-Box"), fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 4.8674, df = 8, p-value = 0.7716
```

```
Box.test(res^2, lag = 10, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  res^2
## X-squared = 8.3853, df = 10, p-value = 0.5913
```

```
acf(res^2, lag.max=40)
```




```
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

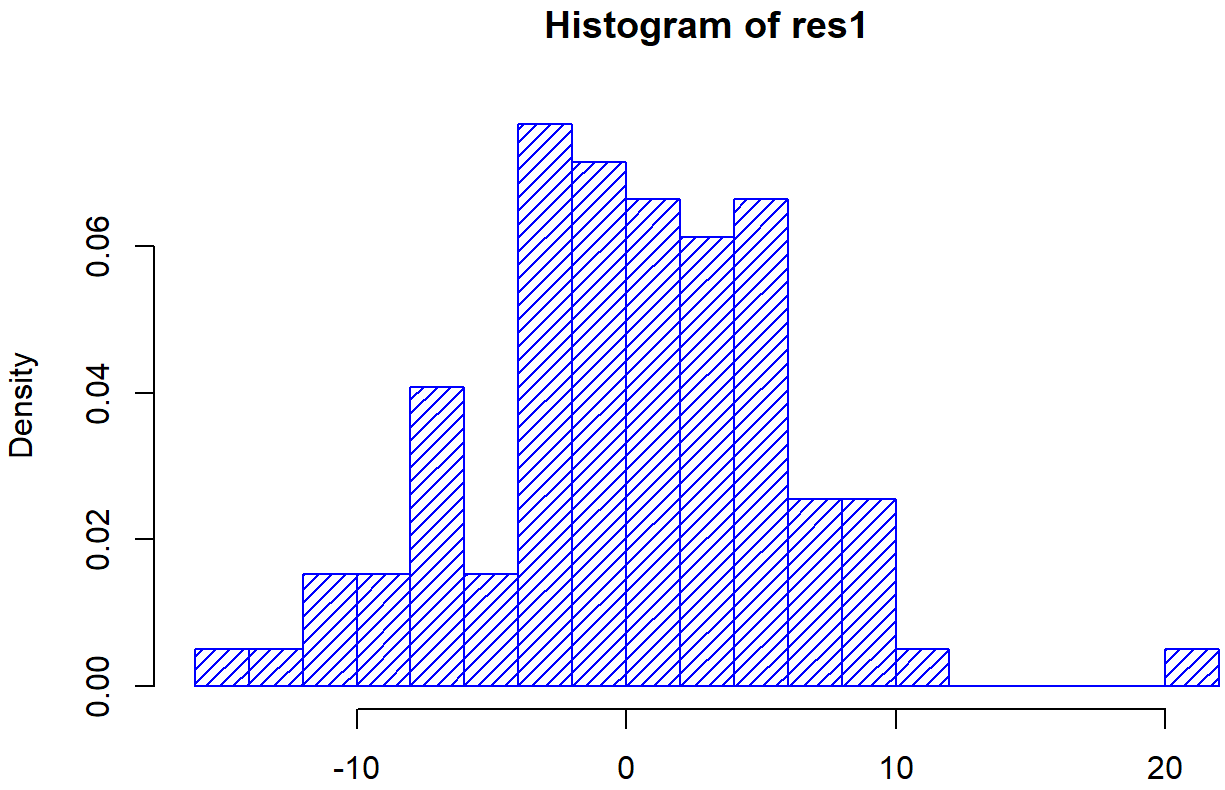
```
##
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  33.98
```

ARMA(1,1) passed all the tests, and all had p values greater tha 0.05.

AR(1) Diagnostics testing:

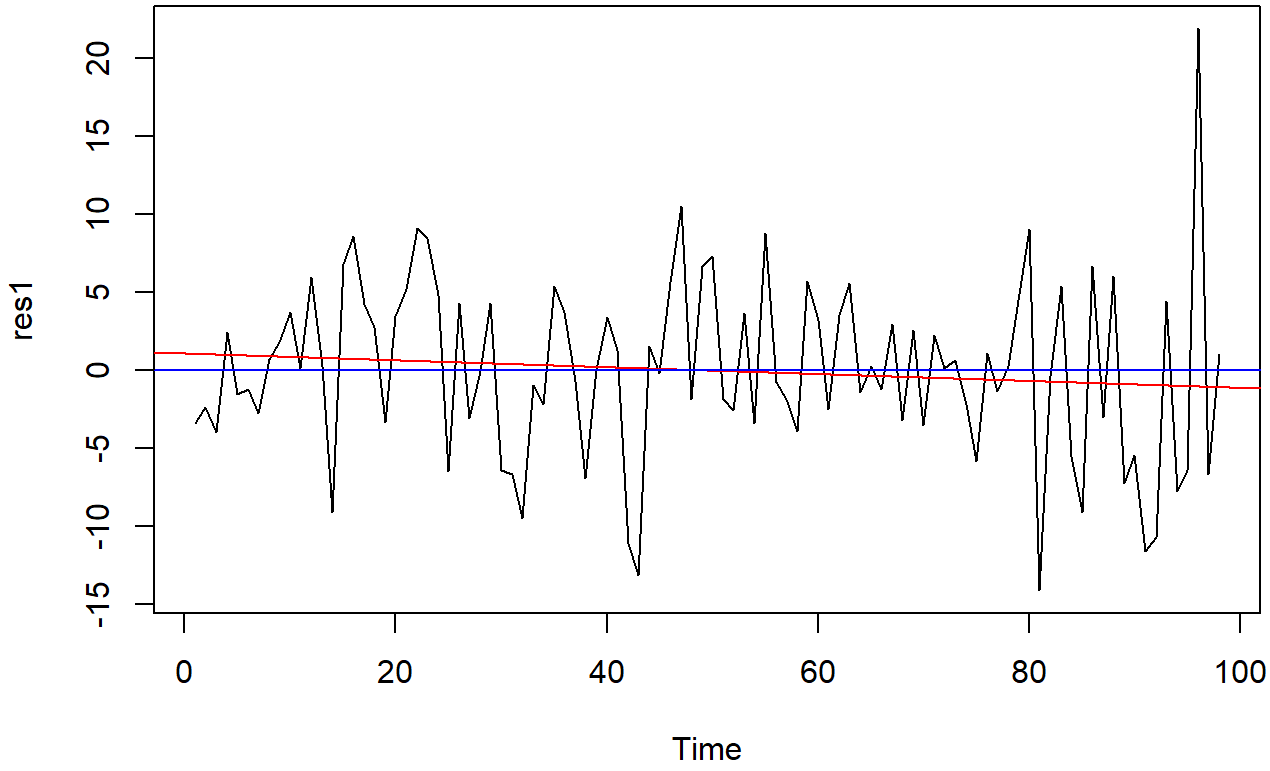
Since $|\phi_1| < 1$ AR(1) model is invertible and stationary

```
#AR(1)
fit1 <- arima(art_diff, order=c(1,0,0), method="ML")
res1 <- residuals(fit)
hist(res1,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
```



Histogram of residuals looks normal

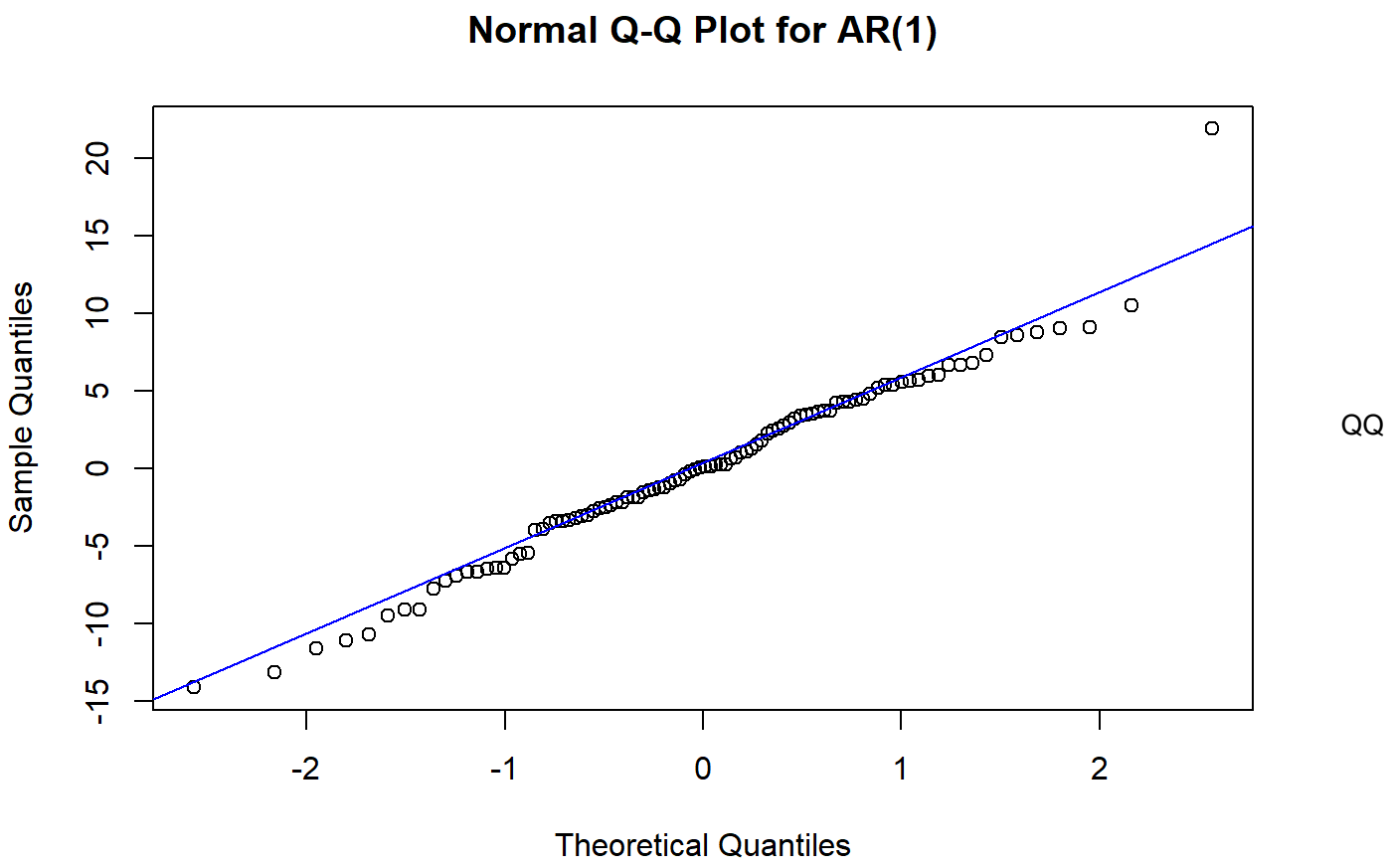
```
m1 <- mean(res1)
std1 <- sqrt(var(res1))
plot.ts(res1)
fitt1 <- lm(res1 ~ as.numeric(1:length(res1))); abline(fitt1, col="red")
abline(h=mean(res1), col="blue")
```



Plot

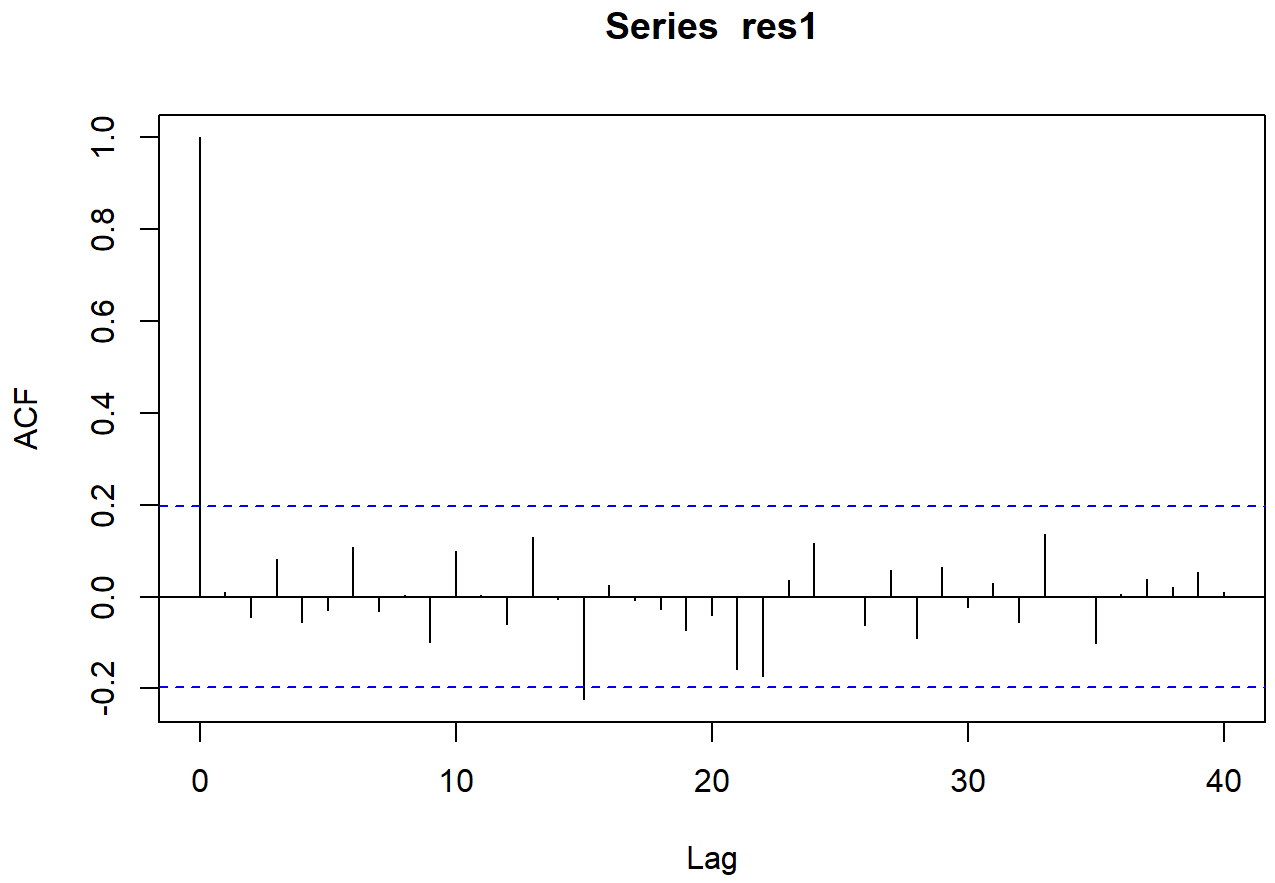
looks stationary.

```
qqnorm(res1,main= "Normal Q-Q Plot for AR(1)")
qqline(res1,col="blue")
```

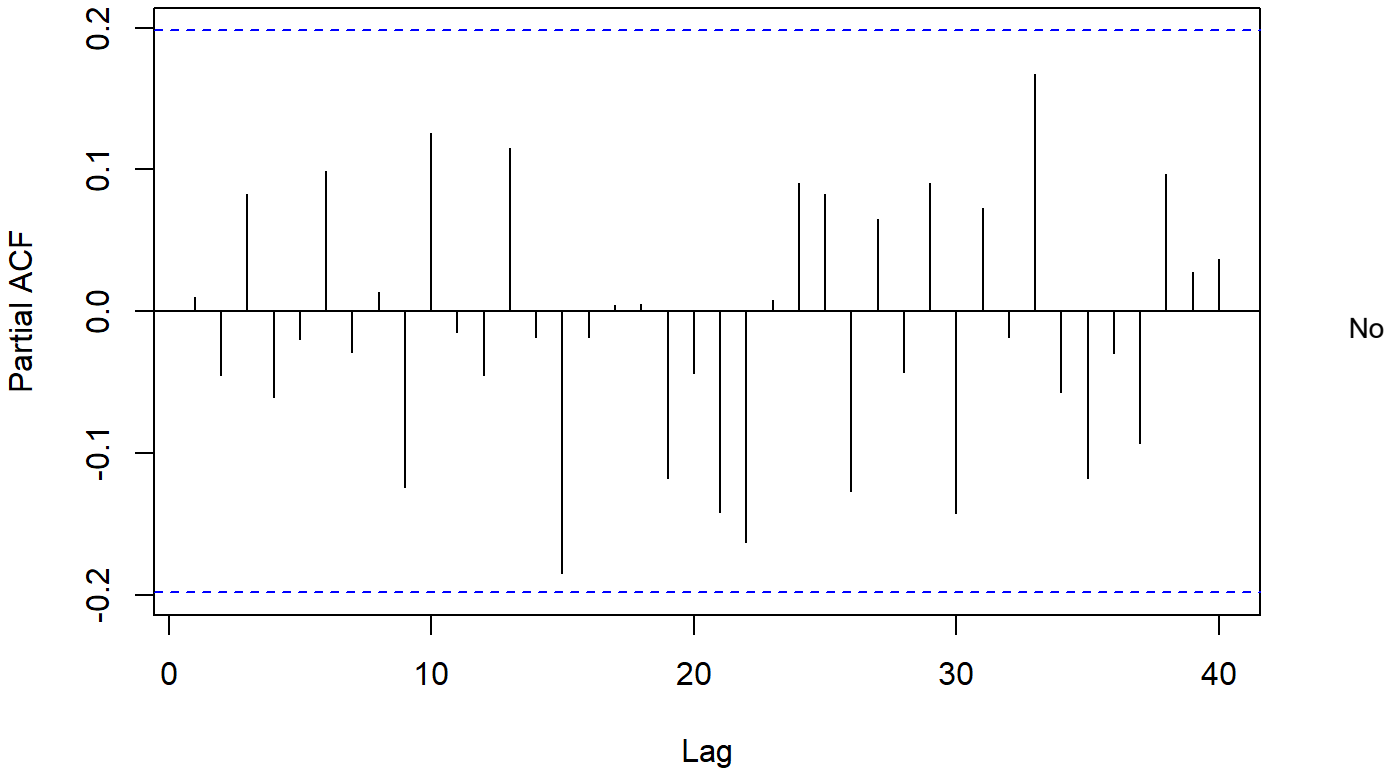


plot looks normal.

```
acf(res1, lag.max=40)
```




```
pacf(res1, lag.max=40)
```



No

irregularities in ACF and PACF or residuals.

```
shapiro.test(res1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res1
## W = 0.97931, p-value = 0.1252
```

```
Box.test(res1, lag = 10, type = c("Box-Pierce"), fitdf = 1)
```

```
##
##  Box-Pierce test
##
## data:  res1
## X-squared = 4.4376, df = 9, p-value = 0.8803
```

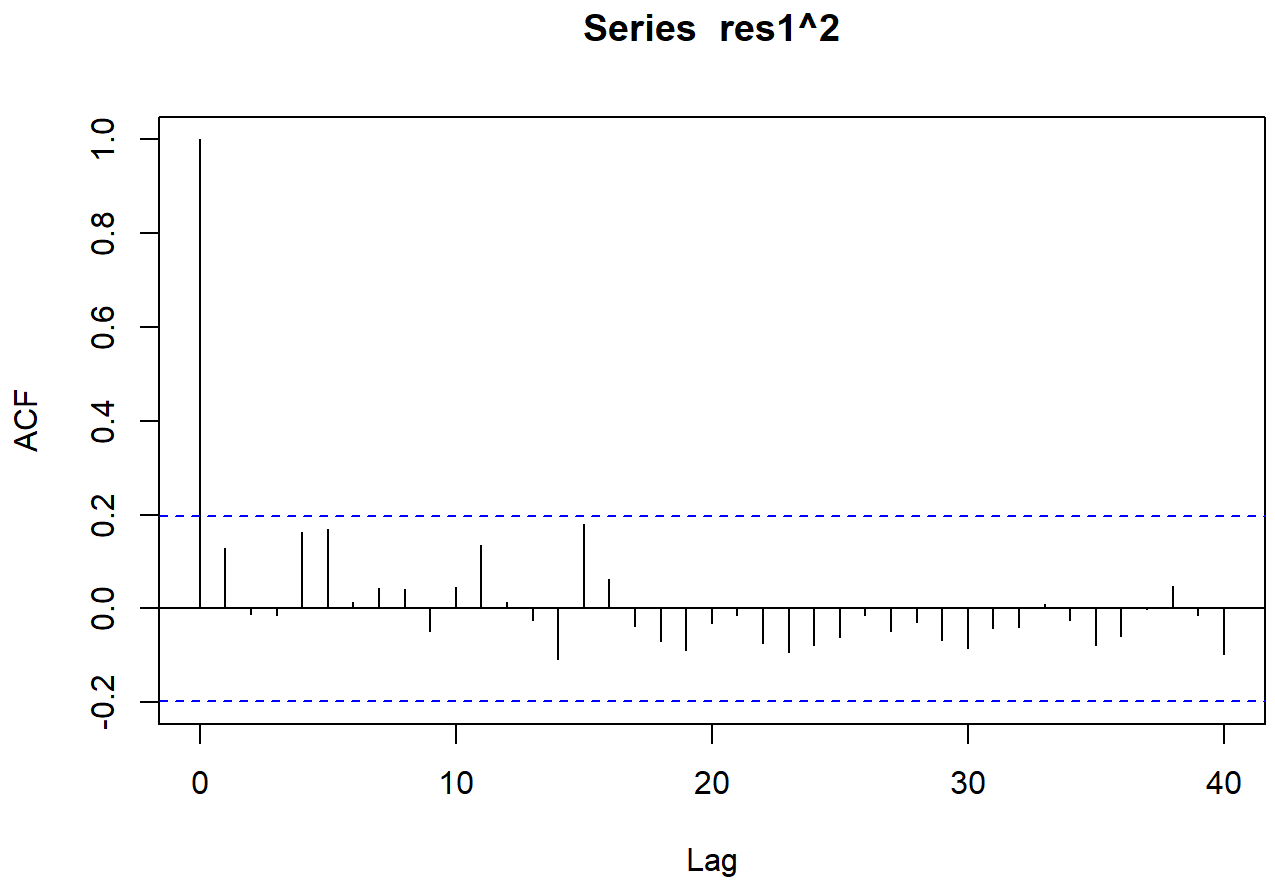
```
Box.test(res1, lag = 10, type = c("Ljung-Box"), fitdf = 1)
```

```
##
##  Box-Ljung test
##
## data:  res1
## X-squared = 4.8674, df = 9, p-value = 0.8457
```

```
Box.test(res1^2, lag = 10, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  res1^2
## X-squared = 8.3853, df = 10, p-value = 0.5913
```

```
acf(res1^2, lag.max=40)
```




```
ar(res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
##
## Call:
## ar(x = res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  33.98
```

AR(1) model passes all the diagnostics checking tests.

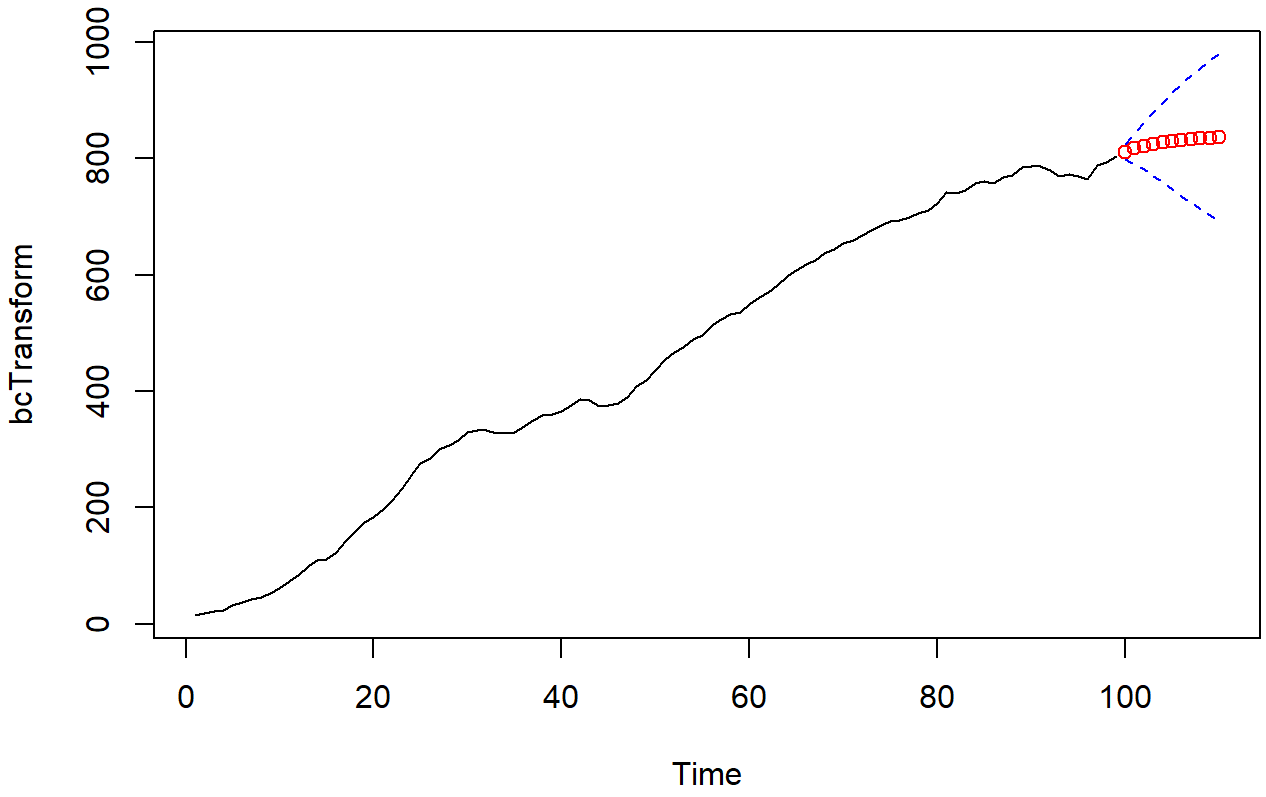
Since both models pass the diagnostic testing and since the AR(1) model has a lower number of estimated values and a lower AICc than ARMA(1,1), I will use the AR(1) model for predicting.

Fit the model and forecast on transformed and original data.

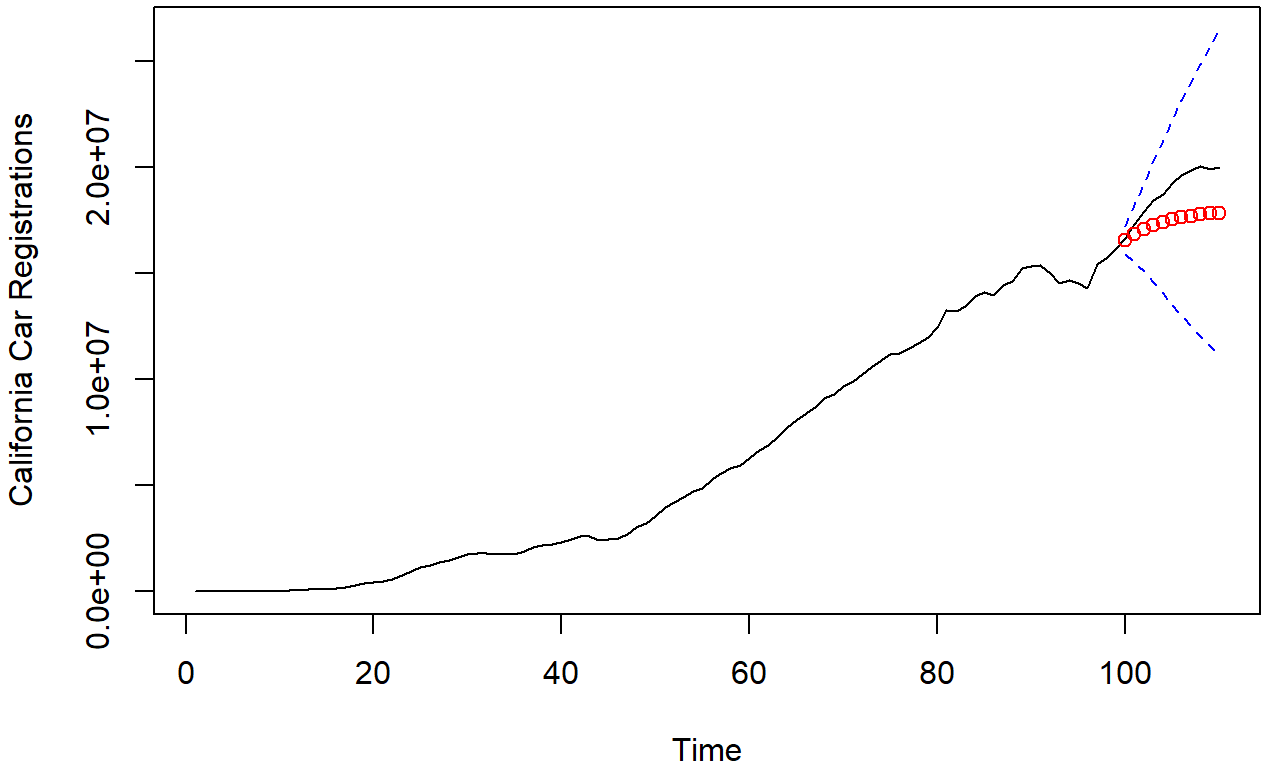
```
fit.A <- arima(bcTransform, order=c(1,1,0), method="ML")
forecast(fit.A)
```

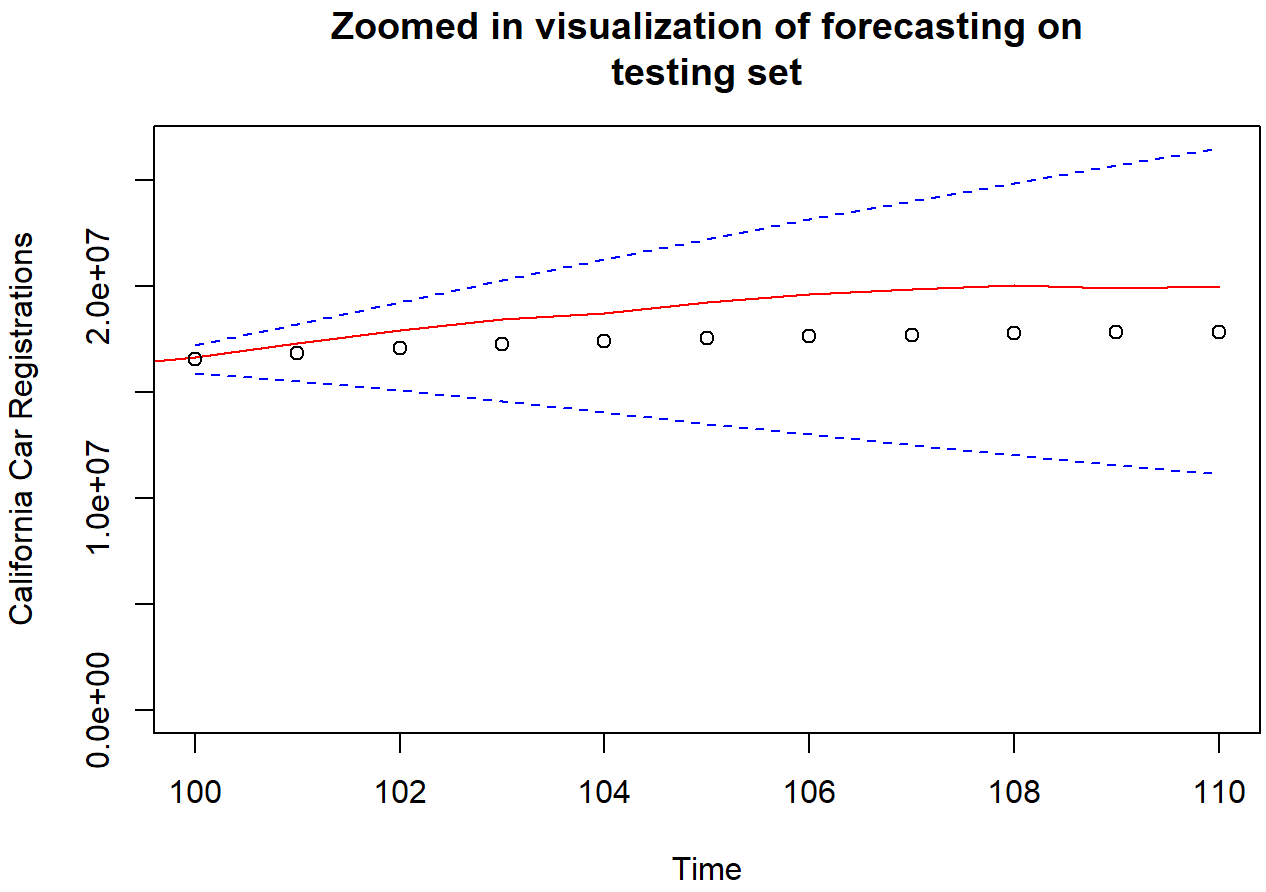
```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 100       811.2598 803.0505 819.4691 798.7048 823.8149
## 101       817.1173 800.3085 833.9261 791.4104 842.8241
## 102       821.7255 795.7919 847.6591 782.0635 861.3875
## 103       825.3509 790.1879 860.5138 771.5738 879.1279
## 104       828.2030 783.9362 872.4699 760.5027 895.9034
## 105       830.4469 777.3257 883.5681 749.2050 911.6888
## 106       832.2122 770.5496 893.8748 737.9074 926.5171
## 107       833.6010 763.7374 903.4647 726.7538 940.4483
## 108       834.6937 756.9752 912.4122 715.8335 953.5539
## 109       835.5532 750.3192 920.7873 705.1990 965.9075
```

```
pred.tr <- predict(fit.A, n.ahead = 11)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(bcTransform, xlim=c(1,length(bcTransform)+11), ylim = c(min(bcTransform),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(bcTransform)+1):(length(bcTransform)+11), pred.tr$pred, col="red")
```


bcTransform
0
200
400
600
800
1000
0
20
40
60
80
100
Time

Forecasting of original data.




Zoomed in visualization of forecasting on testing set
California Car Registrations
0.0e+00
1.0e+07
2.0e+07
100
102
104
106
108
110
Time