Final Project: Budapest Chicken Pox Cases

Aaron Armbruster

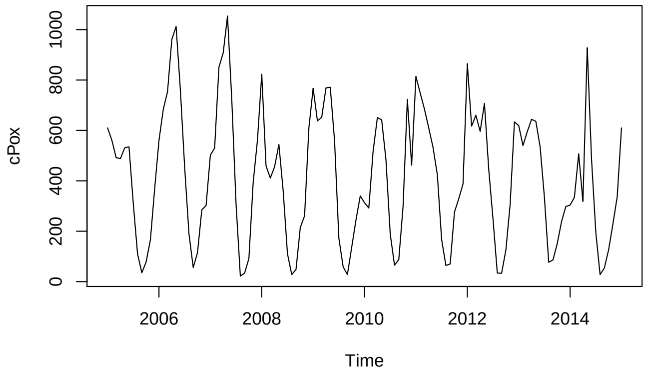
11/14/2021

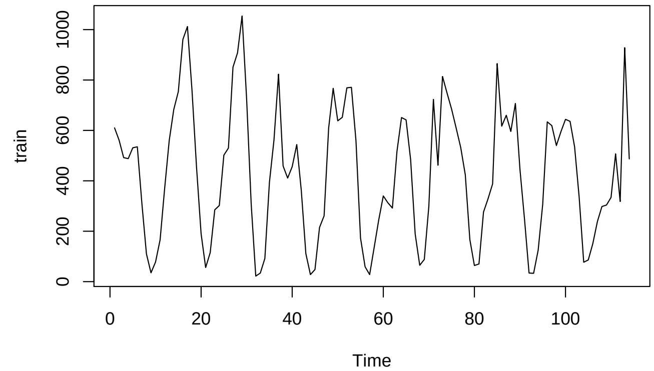
Summary

The data I chose for this project is chicken pox cases in Budapest Hungary. I was intrested in exploring this data set because of the current pandemic we are living in today. I was surprised to see that the data shows very consistent seasonality with very little showing in the way of it tapering off which I believe will continue. I used Box-Cox transformations and differencing to get the data stationary and with a low variance, which I was successful. From the stationary data set I was able to propose multiple models and eventually choose the best one that also was able to forecast well.

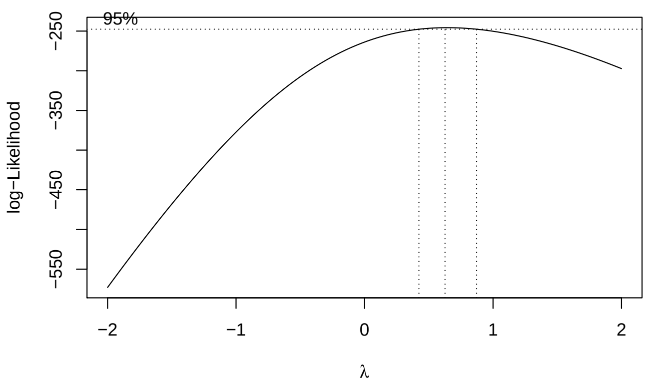
Introduction

I believe the data set I have chosen is interesting because chicken pox is a disease that is entirely preventable in countries that have the means and infrastructure to provide vaccines and in a country like Hungary which is considered a highly developed country there are still many cases even in its most populous city Budapest. My hope is to be able to forecast the data. The techniques I will be using are box-cox transformations, differnecing, utilizing ACF & PACF graphs, AICc, and the many diagnostic checking tools to asses if my chosen model is in fact a good fit. The end results were satisfying, my selected model passed all necessary diagnostic checks. Moreover, my goal to make a proper forecast worked well because the forecasts of points lined up with the test data furthering my belief that my model is in fact a good fit. Basing my analysis off the forecasts of the points it is very likely the seasonal trend will continue into the future which is what I assumed from the very beginning.



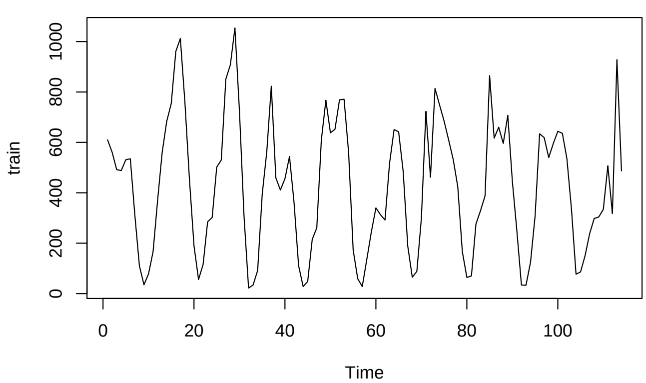


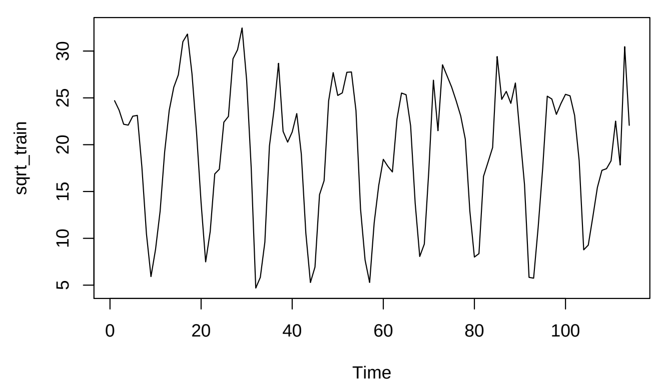
The data shows clear seasonality trend with relatively consistent variance, but no real apparent trend in any direction.




```
## Box-Cox lambda Value: 0.6262626
```

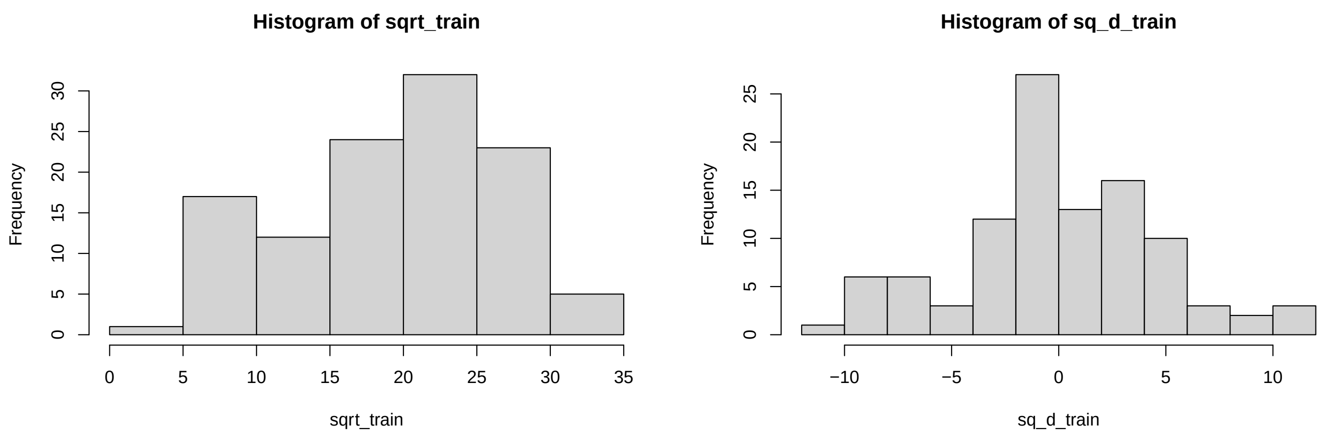
Possibly square root transformation for the data because .5 is in the confidence interval.





The variance seems to stabilize so the square root transformations will be chosen.

```
## Variance of the data before differencing: 51.97122
```

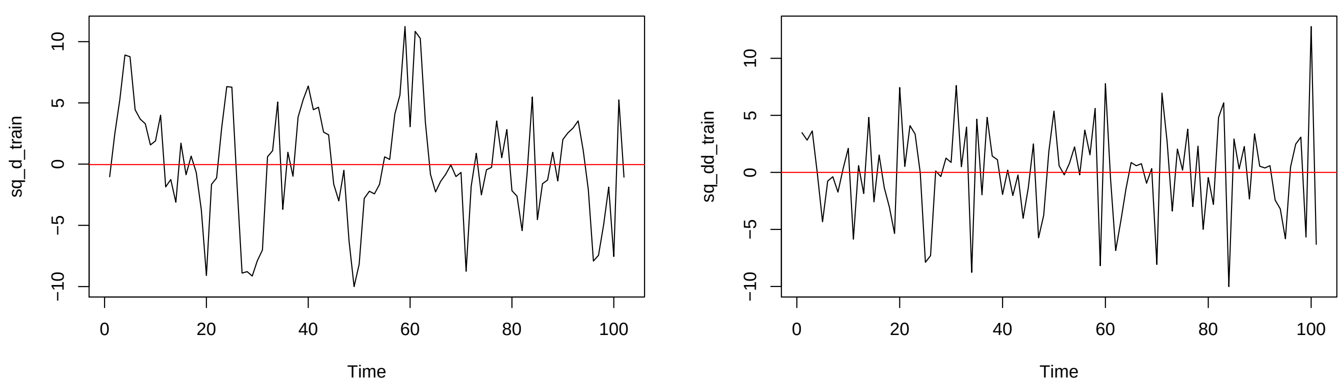


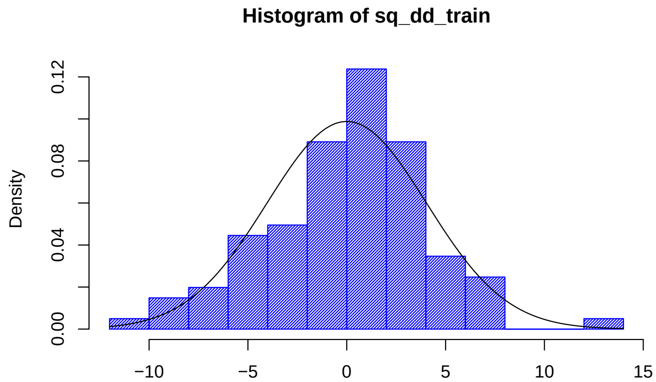
Histogram shows approximately normal data and the differencing at lag 12 removed the seasonality and decreased the variance.

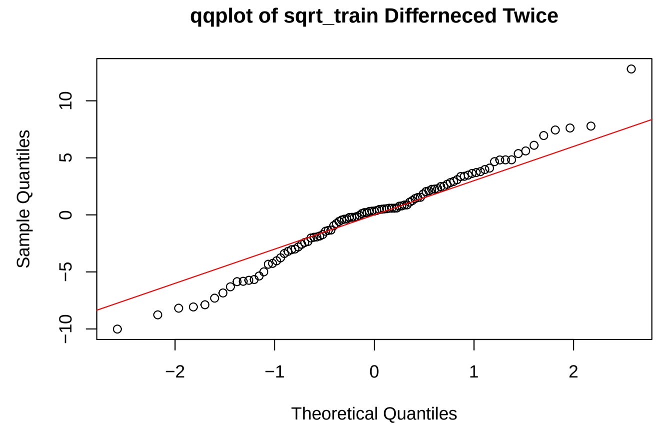
```
## Variance after differencing once at lag 12: 21.21782
```

```
## Variance after differencing at again but at lag 1: 16.29473
```

Diffrencing again at lag 1 shows the variance is still decreasing and now the data is ready.



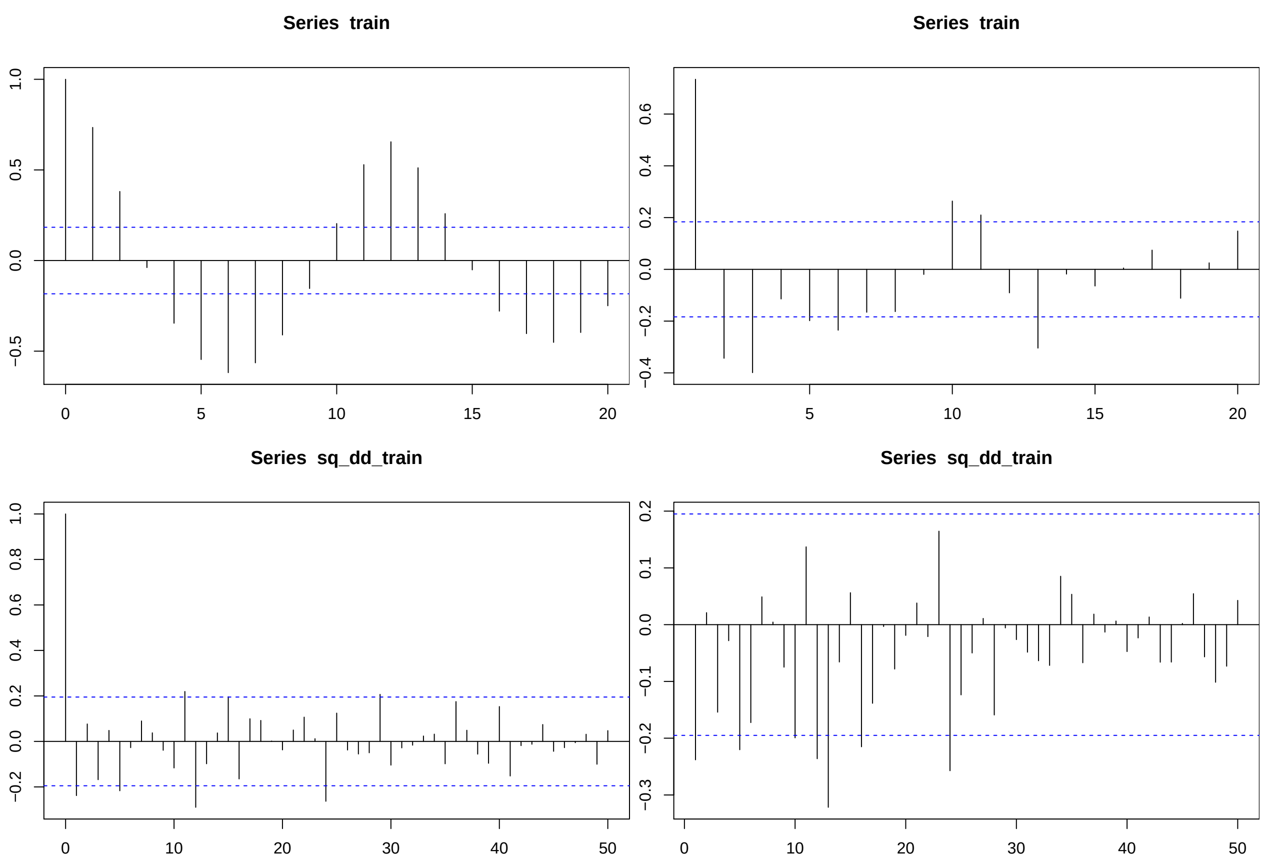




The differencing has made the data stationary as there is trend or seasonality in the data anymore. The transformed and differenced data shows the data may be normal judging from the qqplot and histogram.

```
##
##  Shapiro-Wilk normality test
##
## data:  sq_dd_train
## W = 0.98464, p-value = 0.2928
```

The P value is greater than 0.05 so the assumption of normality will not be rejected.



From the ACF and PACF possible models would all be SARIMA. There is a spike at lag 12 and 24 so Q = 2 then q = 1 or 0 because lag 11 and lag 1 are outside confidence intervals. Because of the differencing D = 1 and d = 1. From PACF again spikes at 12 and 24 so P = 2 and p = 5 or 1. The model will need to have some coefficients set to zero.

```
## [1] 536.5085
```

```
## [1] 528.9993
```

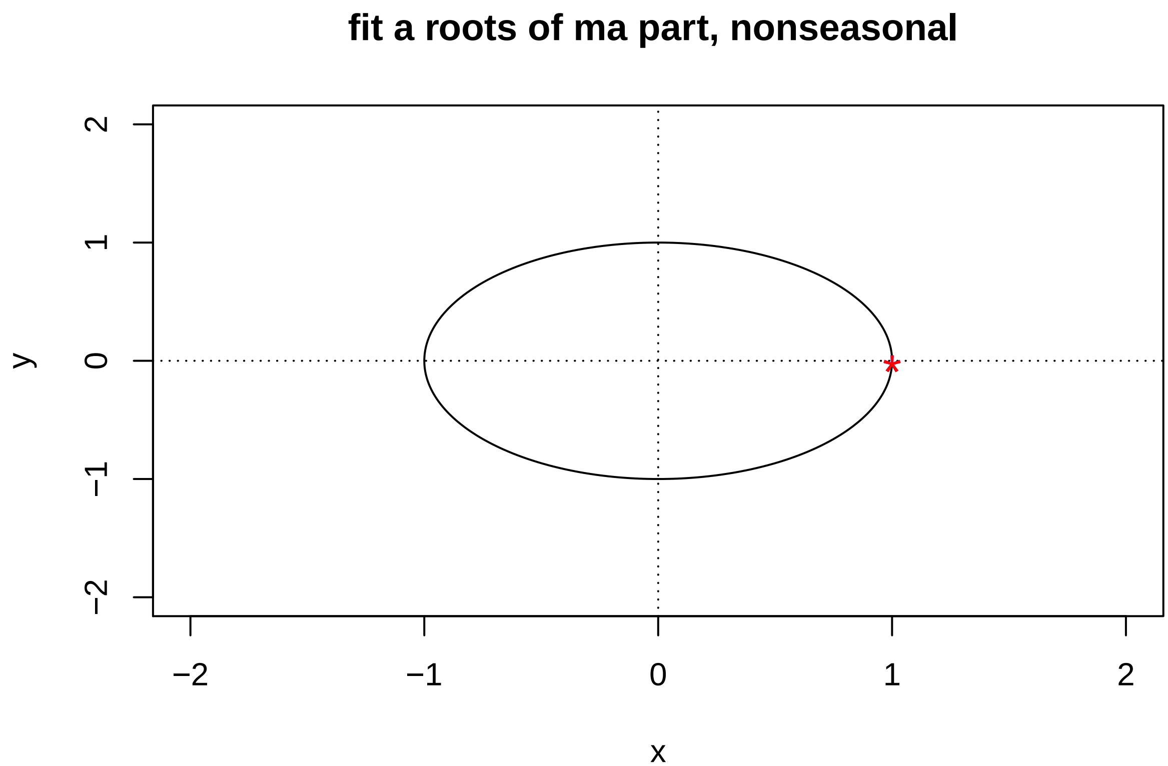
```
## [1] 535.1262
```

```
## [1] 537.6072
```

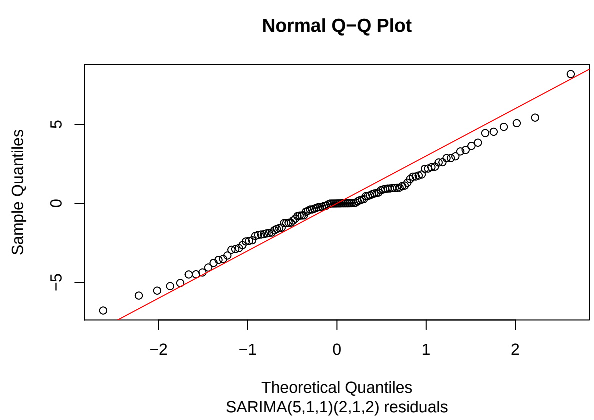
Choose lowest two AICc and beginning diagnostic checking.

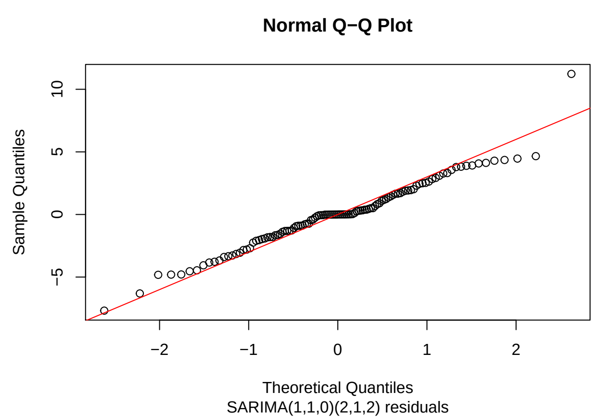
```
##
## Call:
## arima(x = sqrt_train, order = c(5, 1, 1), seasonal = list(order = c(2, 1, 2),
##     period = 12), fixed = c(NA, NA, NA, NA, NA, NA, NA, NA, 0, NA), method = "ML")
##
## Coefficients:
##          ar1     ar2      ar3      ar4      ar5      ma1     sar1     sar2
##       0.5633  0.2194  -0.1974  -0.0188  -0.0656  -1.0000  -0.7456  -0.0923
## s.e.  0.1022  0.1235   0.1320   0.1373   0.1171   0.0508   0.1833   0.1937
##       sma1     sma2
##          0  -0.7353
## s.e.     0   0.3649
##
## sigma^2 estimated as 7.04:  log likelihood = -253.81,  aic = 527.63
```

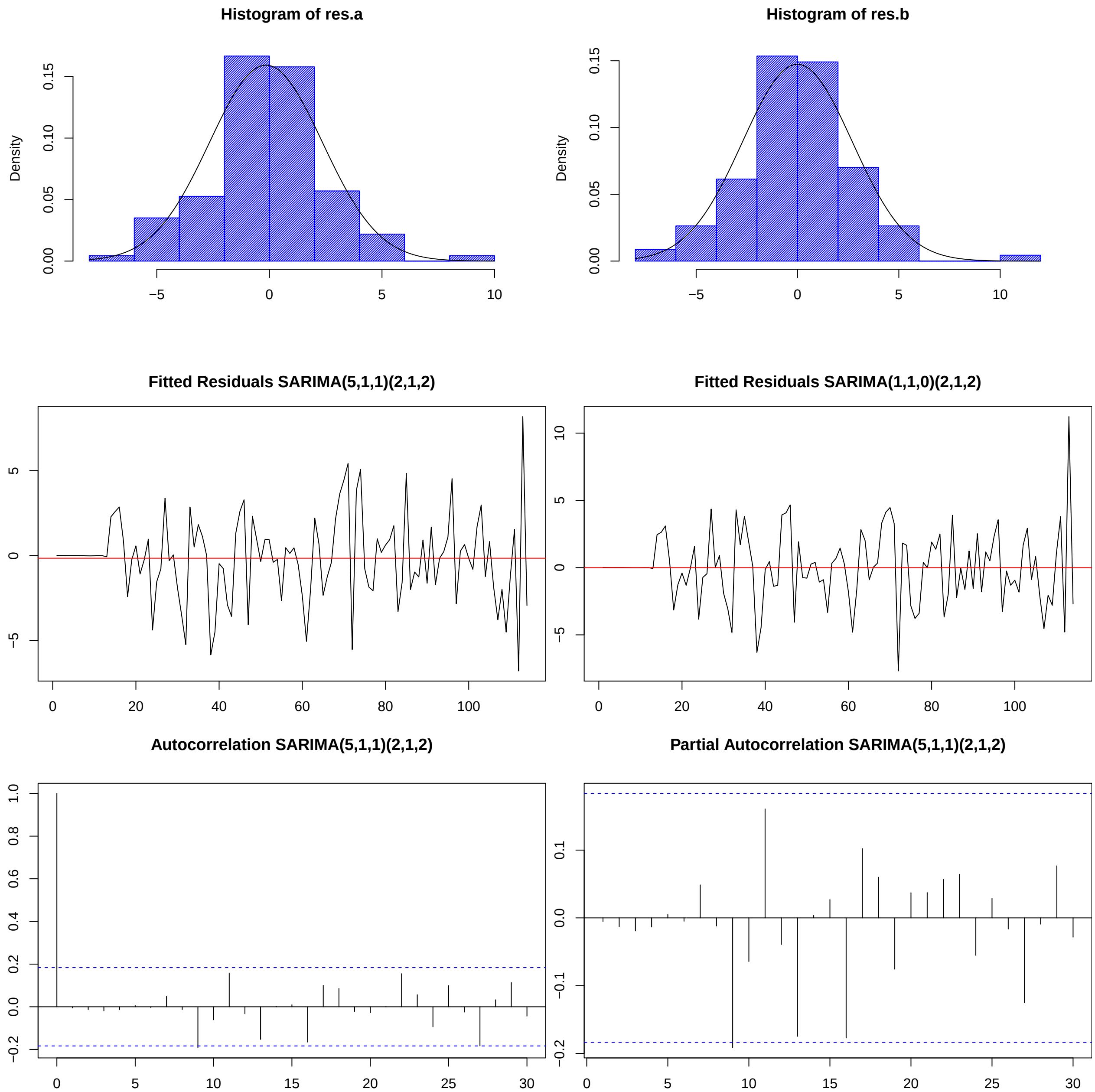
```
##
## Call:
## arima(x = sqrt_train, order = c(1, 1, 0), seasonal = list(order = c(2, 1, 2),
##     period = 12), fixed = c(NA, NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##           ar1     sar1     sar2     sma1     sma2
##       -0.2308  -0.4303  -0.2166  -0.3845  -0.4963
## s.e.   0.1002   0.3451   0.1957   0.5254   0.5761
##
## sigma^2 estimated as 8.196:  log likelihood = -261.29,  aic = 534.57
```



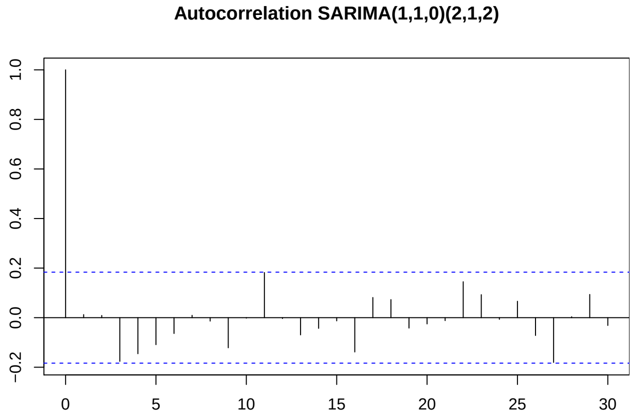
MA part of fit a is invertible

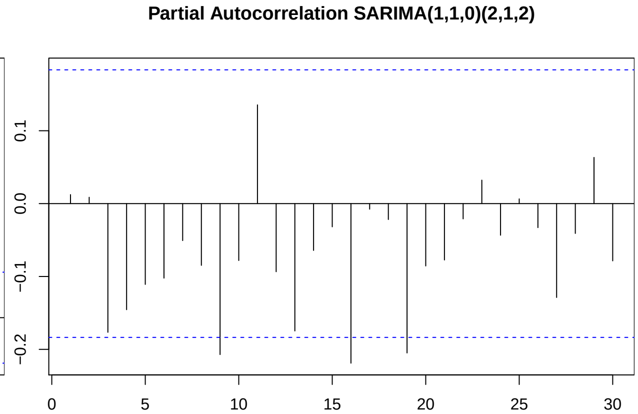






Histogram of res.a
Density
Histogram of res.b
Density
Fitted Residuals SARIMA(5,1,1)(2,1,2)
Fitted Residuals SARIMA(1,1,0)(2,1,2)
Autocorrelation SARIMA(5,1,1)(2,1,2)
Partial Autocorrelation SARIMA(5,1,1)(2,1,2)





Both time series plots of the residuals show stationary data. The residuals for the fit.a with SARIMA(5,1,1)(2,1,2) look normal and the ACF and PACF show the lags are within the confidence intervals. The residuals from the fit.b are also inside confidence intervals so both models have possibility of being sufficient and passing all diagnostic checks.

```
##
##  Shapiro-Wilk normality test
##
## data:  res.a
## W = 0.98173, p-value = 0.1222
```

```
##
##  Box-Pierce test
##
## data:  res.a
## X-squared = 10.626, df = 7, p-value = 0.1558
```

```
##
##  Box-Ljung test
##
## data:  res.a
## X-squared = 11.913, df = 7, p-value = 0.1034
```

```
##
##  Box-Ljung test
##
## data:  (res.a)^2
## X-squared = 21.696, df = 13, p-value = 0.06026
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.b
## W = 0.97017, p-value = 0.01194
```

```
##
##  Box-Pierce test
##
## data:  res.b
## X-squared = 13.843, df = 12, p-value = 0.3108
```

```
##
##  Box-Ljung test
##
## data:  res.b
## X-squared = 14.994, df = 12, p-value = 0.2418
```

```
##
##  Box-Ljung test
##
## data:  (res.b)^2
## X-squared = 6.5066, df = 13, p-value = 0.9258
```

```
##
## Call:
## ar(x = res.a, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  6.269
```

```
##
## Call:
## ar(x = res.b, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  7.326
```

Both models can be fit into a AR(0), but only fit.a SARIMA(5,1,1)(2,1,2) model passes all diagnostic tests as all P values are above 0.05. fit.b did not pass the the the Shapiro-Wilk test for normality so fit.a with a SARIMA(5,1,1)(2,1,2) model is what will be used for forecasting.

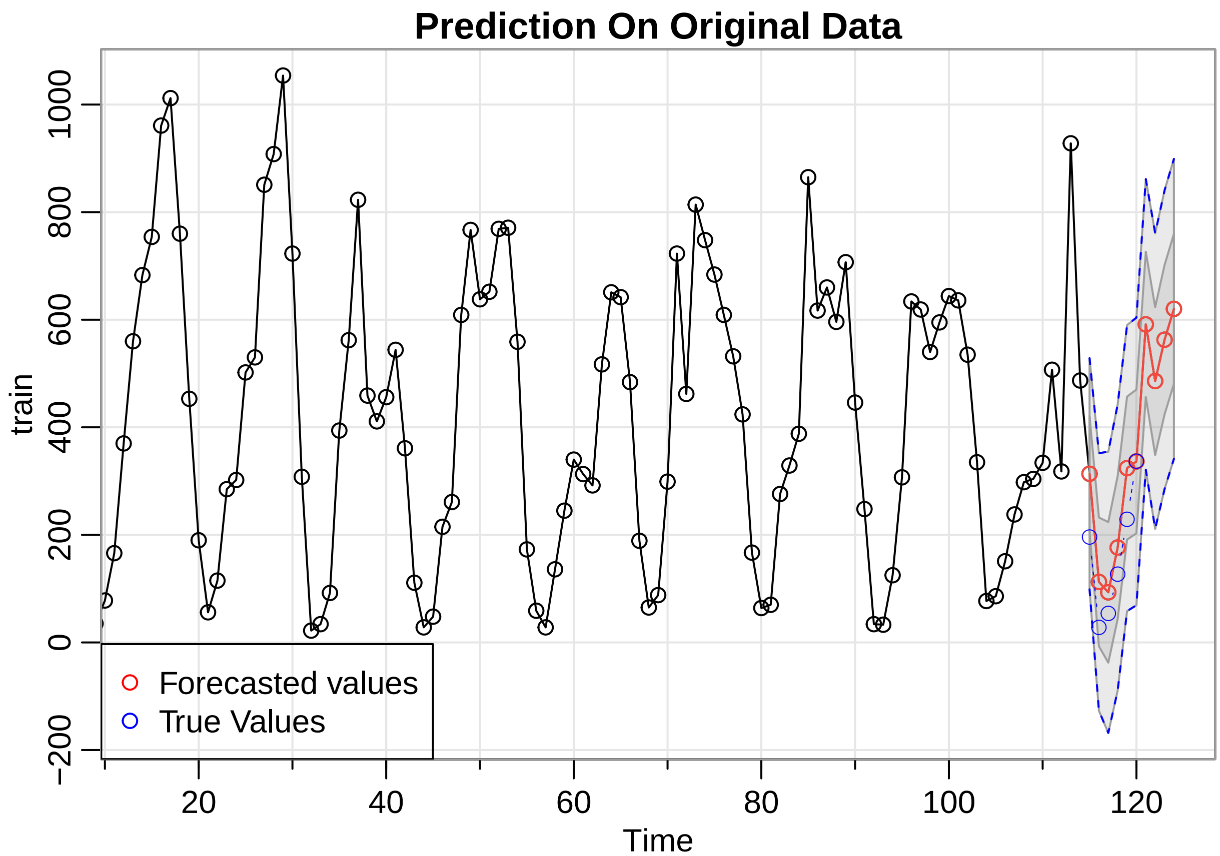
$\phi(B) = 1-0.5904B-(-0.0640)B^2-(-0.1098)B^5$ $\Phi(B) = 1-0.0536B^{60}$ $\theta(B) = 1-B$ $\Theta(B) = 1-0.998B^{24}$
$\nabla^d = (1-B)$ $\nabla_s^D = (1-B^{12})$
SARIMA(5,1,1)(2,1,2) model equation

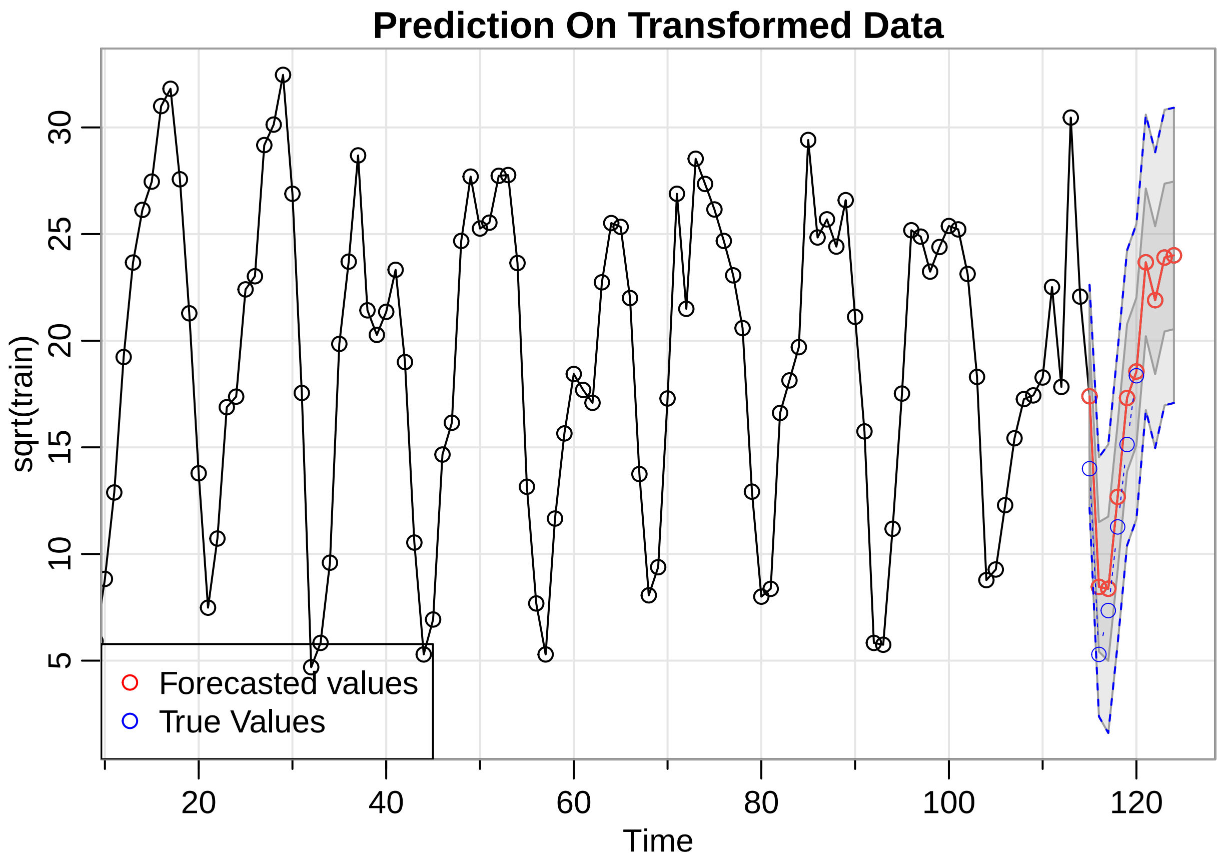
$$(1-0.0536B^{60})*(1-B^{12})*(1-B)*(1-0.5904B-(-0.0640)B^2-(-0.1098)B^5) = (1-0.998B^{24})*(1-B)*(Z_t)$$

Where Z_t is the non stationary time series sqrt_train

```
##     Point Forecast       Lo 80    Hi 80      Lo 95    Hi 95
## 115      17.832722 14.357407 21.30804 12.5176861 23.14776
## 116       8.238706  4.210010 12.26740  2.0773472 14.40006
## 117       8.258005  3.764270 12.75174  1.3854304 15.13058
## 118      12.005624  7.411086 16.60016  4.9788853 19.03236
## 119      15.947869 11.316148 20.57959  8.8642628 23.03148
## 120      18.060920 13.430973 22.69087 10.9800274 25.14181
## 121      22.112083 17.480030 26.74414 15.0279696 29.19620
## 122      20.016265 15.380112 24.65242 12.9258811 27.10665
## 123      22.740632 18.105163 27.37610 15.6512939 29.82997
## 124      22.319914 17.687889 26.95194 15.2358436 29.40398
## 125      26.924604 22.296481 31.55273 19.8465006 34.00271
## 126      21.393085 16.767100 26.01907 14.3182510 28.46792
## 127      15.151452 10.390378 19.91253  7.8700178 22.43289
## 128       7.092589  2.271791 11.91339 -0.2801847 14.46536
## 129       6.722612  1.842616 11.60261 -0.7406981 14.18592
```

```
## 130        12.294699  7.392593 17.19681  4.7975751 19.79182
## 131        18.241167 13.325432 23.15690 10.7231991 25.75913
## 132        21.424154 16.505104 26.34320 13.9011158 28.94719
## 133        24.822569 19.902592 29.74254 17.2981142 32.34702
## 134        23.370630 18.455459 28.28580 15.8535245 30.88774
## 135        24.362316 19.451453 29.27318 16.8517991 31.87283
## 136        25.031729 20.124656 29.93880 17.5270091 32.53645
## 137        25.248423 20.343842 30.15300 17.7475134 32.74933
## 138        22.237357 17.334295 27.14042 14.7387710 29.73594
```





Running forecast(fit.a) will ensure the model is infact invertible otherwise it would throw an error. The SARIMA(5,1,1)(2,1,2) model forecasts the points well as the forecasted points coincide with the true points. It is clear that the seasonal trend will continue into the future.

Conclusion

I was able to achieve my goal of forecasting the data. From the forecasts it is clear that the trend will continue unless some other factor is changed. The model I used to forecast was a SARIMA(5,1,1)(2,1,2)12 with model equation:

$(1-0.0536B^{60})*(1-B^{12})*(1-B)*(1-0.5904B-(-0.0640)B^2-(-0.1098)B^5) = (1-0.998B^{24})*(1-B)*(Z_t)$

Where Z_t is the non stationary time series sqrt_train

References

Lecture Notes and Slides from PSTAT 174

Data Used: https://archive.ics.uci.edu/ml/datasets/Hungarian+Chickenpox+Cases

Appendix

```r
knitr::opts_chunk$set(echo = FALSE)
library(tsdl)
library(forecast)
require(forecast)
library(astsa)
require(MASS)
library(MuMIn)
library(qpcR)

hungary_chickenpox.csv = read.csv("c_pox_monthly.csv",nrows = 121)

cPox <- ts(hungary_chickenpox.csv[,2],start = c(2005),end = c(2015),frequency = 12)

train <- cPox[c(1:114)]
test <- cPox[c(115:120)]



ts.plot(cPox)
ts.plot(train)  # Ts plot shows seasonal plot with minimal to no trend


bcTransform <- boxcox(train~ as.numeric(1:length(train)))
cat("Box-Cox lambda Value:", bcTransform$x[which(bcTransform$y == max(bcTransform$y))])
# lambda ~ 0.61 so sqrt transformation

sqrt_train <- sqrt(train)


ts.plot(train)
ts.plot(sqrt_train)

cat("Variance of the data before differencing:",var(sqrt_train))

hist(sqrt_train)


sq_d_train <- diff(sqrt_train, 12) # Remove seasonality

hist(sq_d_train)

# Variance is decreasing
cat("Variance after differencing once at lag 12:",var(sq_d_train))

sq_dd_train <- diff(sq_d_train, 1)# Remove trend


cat("Variance after differencing at again but at lag 1:",var(sq_dd_train))
# Variance has decreased even more. Will use differenced data
ts.plot(sq_d_train)
abline(h=mean(sq_d_train), col="red")
```

```r
ts.plot(sq_dd_train)
abline(h=mean(sq_dd_train), col="red") # seasonality has been removed and data is now stationary


hist(sq_dd_train, density=50,breaks=10, col="blue", xlab="", prob=TRUE)
curve(dnorm(x,mean= mean(sq_dd_train),sd = sqrt(var(sq_dd_train))),add = TRUE)
# Transformed Histogram shows approx normal dist

qqnorm(sq_dd_train,main = "qqplot of sqrt_train Differneced Twice")
abline(coef = c(0,3),col = "red") # Straight line

shapiro.test(sq_dd_train) # P val > 0.05 so we do not reject assumption of normality


op <- par(mar = c(3,2,3,0))
acf(train)
pacf(train)



acf(sq_dd_train,lag.max = 50)
pacf(sq_dd_train,lag.max = 50)


# Choosing models using AICc function
AICc(arima(sqrt_train, order=c(1,1,1),
           seasonal = list(order = c(2,1,2), period = 12), method="ML")) # 536.5085
AICc(arima(sqrt_train, order=c(5,1,1),
           seasonal = list(order = c(2,1,2), period = 12), method="ML")) # 528.9993
AICc(arima(sqrt_train, order=c(1,1,0),
           seasonal = list(order = c(2,1,2), period = 12), method="ML")) # 535.1262
AICc(arima(sqrt_train, order=c(5,1,0),
           seasonal = list(order = c(2,1,2), period = 12), method="ML")) # 537.6072


#fit.a <- arima(sqrt_train, order=c(5,1,1),
 #              seasonal = list(order = c(2,1,2), period = 12), method="ML",fixed=c(NA,0,0,NA,NA,NA,NA,

fit.b<- arima(sqrt_train, order=c(1,1,0),
               seasonal = list(order = c(2,1,2), period = 12), method="ML",fixed = c(NA,NA,NA,NA,NA))

fit.a <- arima(sqrt_train, order=c(5,1,1),
              seasonal = list(order = c(2,1,2), period = 12), method="ML",fixed=c(NA,NA,NA,NA,NA,NA,NA,

fit.a

fit.b


source("plotroots.R")
plot.roots(NULL,polyroot(c(1, -1)), main="fit a roots of ma part, nonseasonal ")
```

```r
res.a <- residuals(fit.a)
res.b <- residuals(fit.b)

qqnorm(res.a)
title(sub = "SARIMA(5,1,1)(2,1,2) residuals")
abline(coef = c(0,3),col = "red") # Straight line

qqnorm(res.b)
title(sub = "SARIMA(1,1,0)(2,1,2) residuals")
abline(coef = c(0,3),col = "red") # straight line as well

hist(res.a,density=50,breaks=10, col="blue", xlab="", prob=TRUE)
curve(dnorm(x,mean= mean(res.a),sd = sqrt(var(res.a))),add = TRUE)

hist(res.b,density=50,breaks=10, col="blue", xlab="", prob=TRUE)
curve(dnorm(x,mean= mean(res.b),sd = sqrt(var(res.b))),add = TRUE)

op <- par(mar = c(3,2,3,0))

ts.plot(res.a,main = "Fitted Residuals SARIMA(5,1,1)(2,1,2)"); abline(h = mean(res.a), col = "red")
ts.plot(res.b,main = "Fitted Residuals SARIMA(1,1,0)(2,1,2)"); abline(h = mean(res.b), col = "red")

acf(res.a,main = "Autocorrelation SARIMA(5,1,1)(2,1,2)",lag.max = 30)
pacf(res.a,main = "Partial Autocorrelation SARIMA(5,1,1)(2,1,2)",lag.max = 30)


acf(res.b,main = "Autocorrelation SARIMA(1,1,0)(2,1,2)",lag.max = 30)
pacf(res.b,main = "Partial Autocorrelation SARIMA(1,1,0)(2,1,2)",lag.max = 30)




# Test for the normality of residuals for model a:
shapiro.test(res.a)

# p val needs to be > 0.05

## Test for independence of residuals:
Box.test(res.a, lag=13, type=c("Box-Pierce"), fitdf=6)

Box.test(res.a, lag=13, type=c("Ljung-Box"), fitdf=6)

Box.test((res.a)^2, lag=13, type=c("Ljung-Box"), fitdf=0)
# all p val for model a > 0.05 so assumption of independence isn't rejected


# Test for the normality of residuals for model b:

shapiro.test(res.b)

## Test for independence of residuals for model b:
Box.test(res.b, lag=13, type=c("Box-Pierce"), fitdf=1)
```

```r
Box.test(res.b, lag=13, type=c("Ljung-Box"), fitdf=1)

Box.test((res.b)^2, lag=13, type=c("Ljung-Box"), fitdf=0)





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

ar(res.b, aic = TRUE, order.max = NULL, method = c("yule-walker"))
# fit into AR(0)


forecast(fit.a)

mypred <- sarima.for(train,n.ahead = 10, p=5, d=0, q=1, P=2, D=1, Q=2, S=12)
U.tr= mypred$pred + 2*mypred$se
L.tr= mypred$pred - 2*mypred$se

title("Prediction On Original Data")
lines(seq(115,120),test,col="blue",type = "b",lty = "dashed",ljoin = 1,lwd= 0.5)
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
legend("bottomleft", pch=1, col=c("red", "blue"),
legend=c("Forecasted values", "True Values"))

mypred <- sarima.for(sqrt(train),n.ahead = 10, p=5, d=1, q=1, P=2, D=1, Q=2, S=12)
U.tr= mypred$pred + 2*mypred$se
L.tr= mypred$pred - 2*mypred$se

title("Prediction On Transformed Data")
lines(seq(115,120),sqrt(test),col="blue",type = "b",lty = "dashed",ljoin = 1,lwd= 0.5)
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
legend("bottomleft", pch=1, col=c("red", "blue"),
legend=c("Forecasted values", "True Values"))
```