Times Series Analysis of Monthly International Airline Passengers

PSTAT 174 Final Project

RYAN GAN, ISABELLE LAMBERT, LUCAS MORGAN, LUKAS POKHREL, SPENCER WU

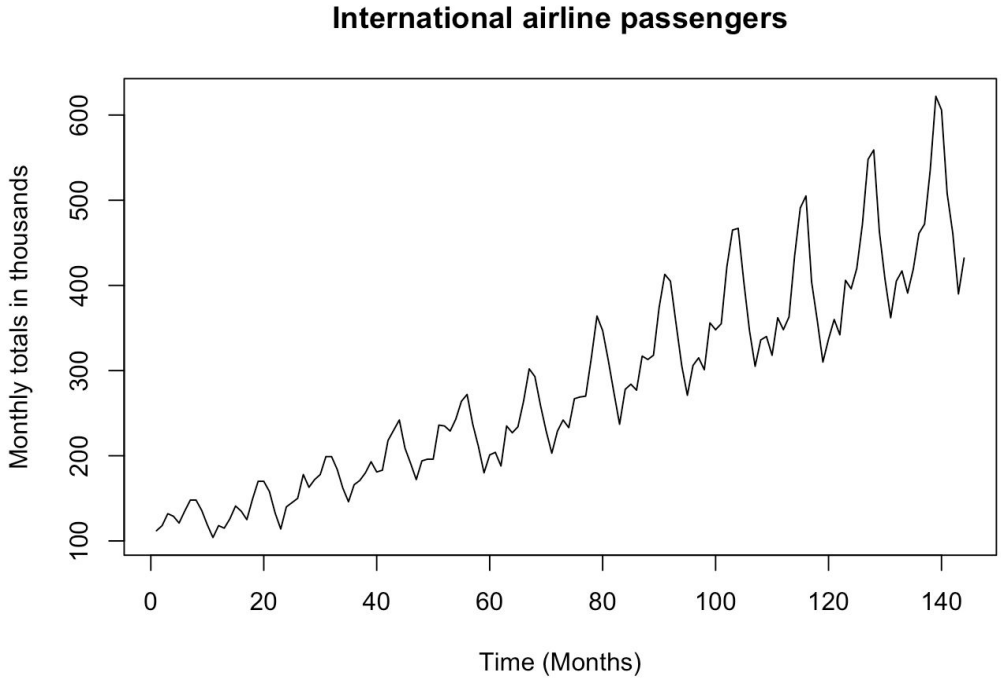
Contents

Abstract

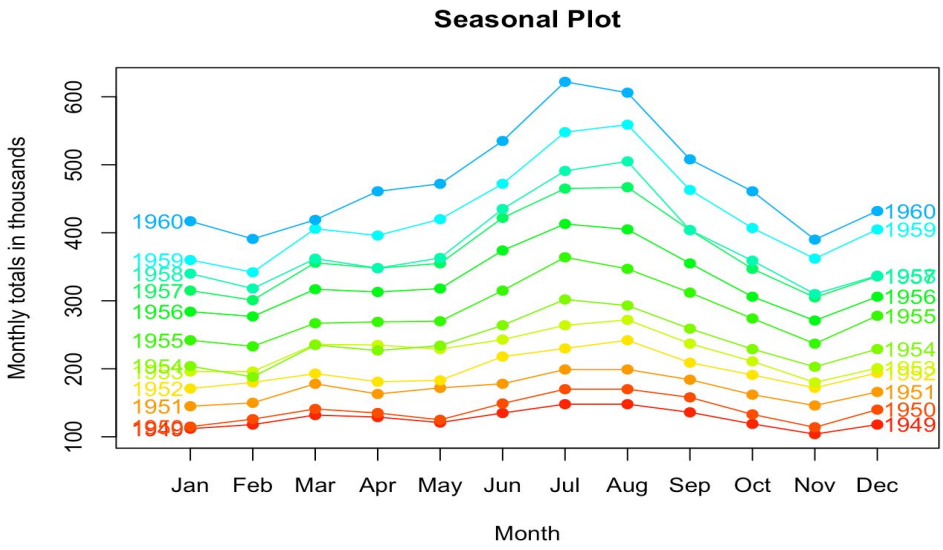
The objective of this project is to forecast the next 24 months of total international airline passengers and analyze the dataset using time series techniques. We used various data transformation methods in order to fit a seasonal autoregressive integrated moving average model to the data. Some of the methods we explored included Box-Cox, logarithm, square root, and differencing. Ultimately, we utilized a Box Cox Transformation with $\lambda=0.05$ in order to control heteroskedasticity. We then performed diagnostics to our model to check issues with normality, constant variance, and independence. We ended up using the differencing technique onto the data: once at lag 1 and once at lag 12. Ultimately, we concluded that the best fit model for our data was a SARIMA $(0, 1, 1) \times (0, 1, 1)_{12}$ model.

1.0 Introduction

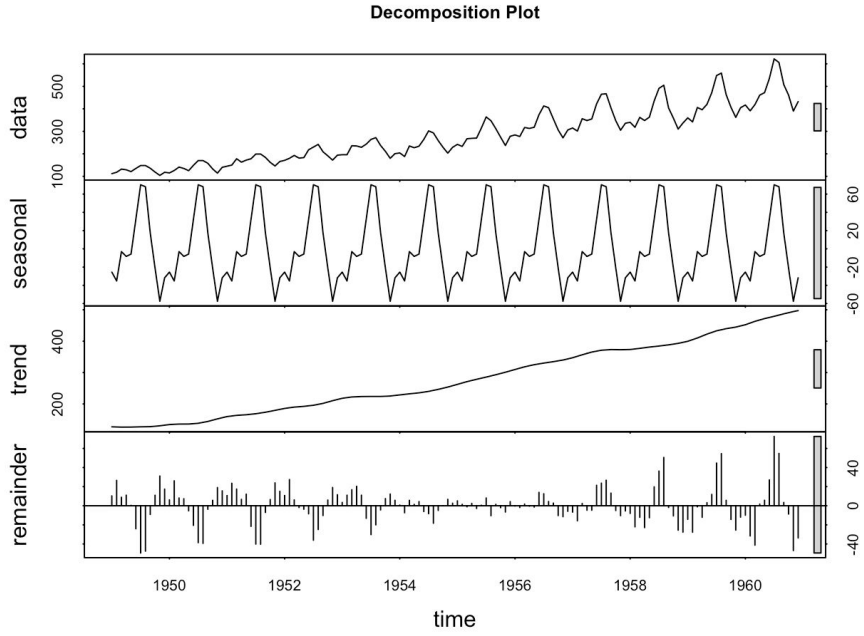
The data that we explored measured monthly international airline passengers. The two variables included time, in terms of month and year, and monthly totals of airline passengers in thousands. The data included 144 observations beginning in January 1949 and ending in December 1960. We began data exploration by plotting the time series. The plot shows clear monthly seasonality and an upward trend.



We took a closer look at the seasonality by creating a seasonal plot of the observations by year. The seasonal plot shows a clear oscillating pattern by month that is evident in each year's data. There are significant spikes downwards but followed by a consistent climbing trend that overall, displays an upward trend as the years go by.



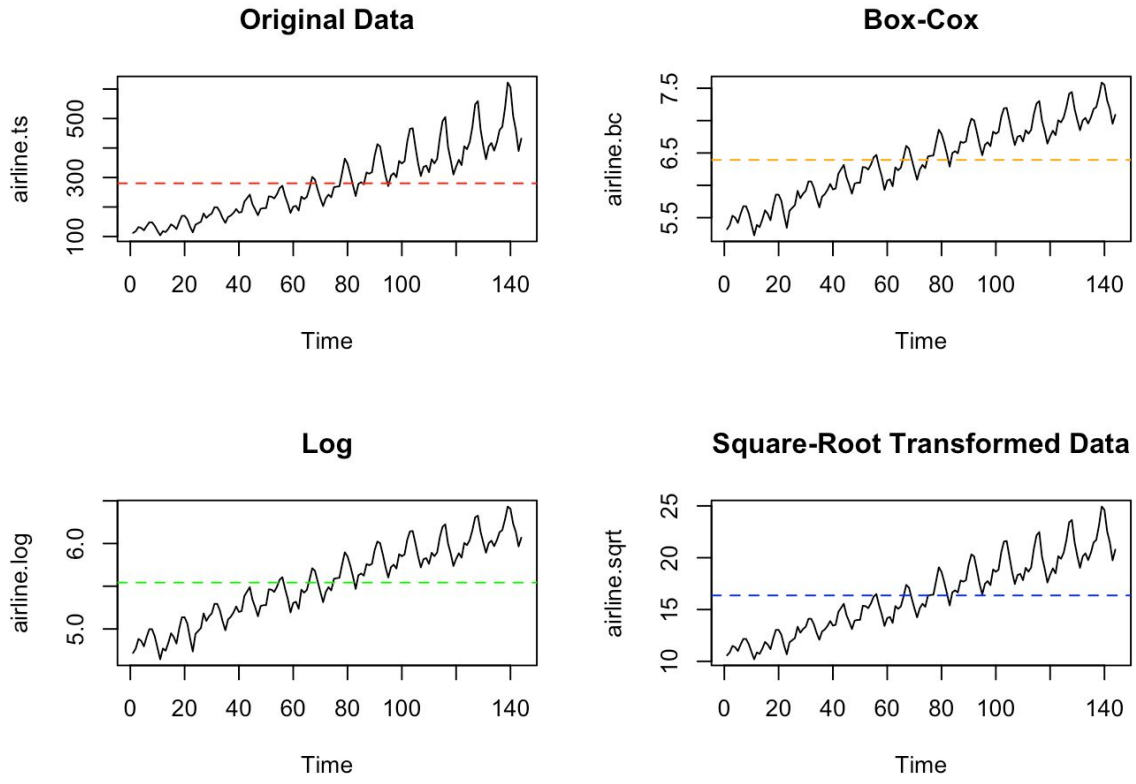
To get a further idea of how the data was impacted by the seasonality and trend, we created a decomposition plot. This plot is able to clearly show seasonality and an upward trend.



2.0 Data Transformations

2.1 Box-Cox Transform

To transform our data into a stationary series, we first looked to variance smoothing transformations. We tested three transformations: box-cox, square root, and logarithm. We calculated the ideal λ for a box-cox transformation to be 0.05. The resulting plots are below.



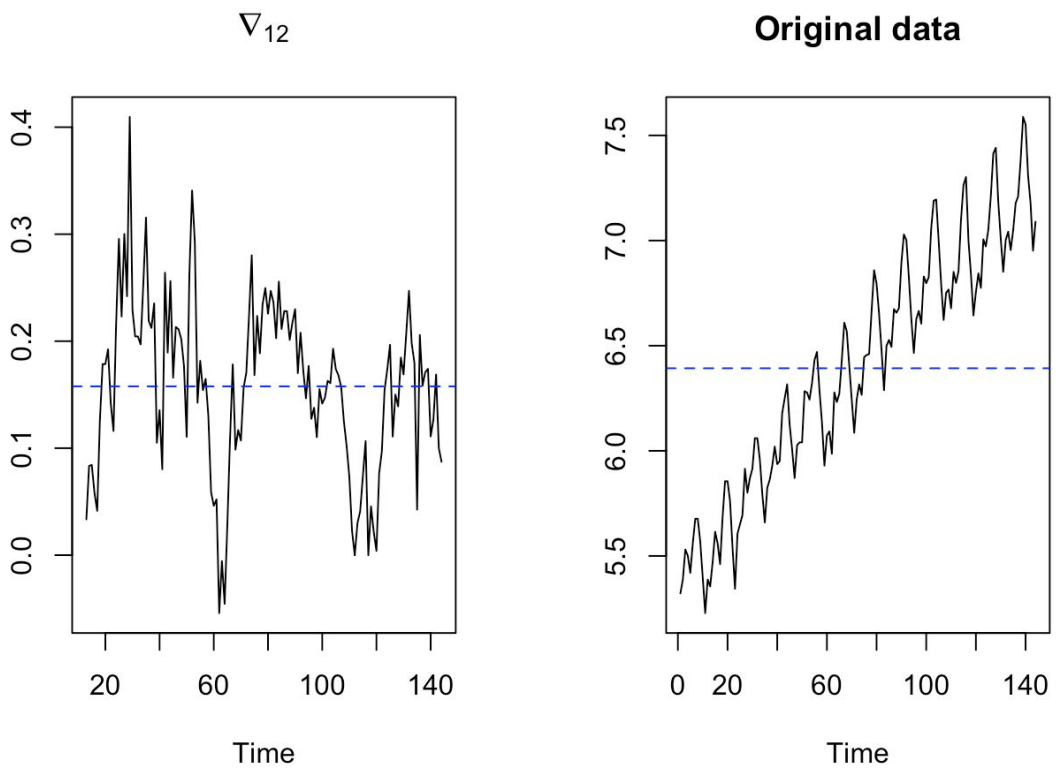
Based on the plots, the box-cox transformation and log transformation seems to do the best job at smoothing the variance over time. We will use the box-cox transformation with $\lambda=0.05$. We then looked at the resulting variances produced from each transformation we tested.

Table 1: Variances

Original	Box-Cox	Log	Square Root
14391.92	0.3384626	0.1948838	12.66472

2.2 Remove Seasonality

We were able to see from our time series plot that the data has clear monthly seasonality. To remove this for model building we differenced at lag 12. Below is the resulting plot compared to the original. The transformation seems to be adequate to remove monthly seasonality.



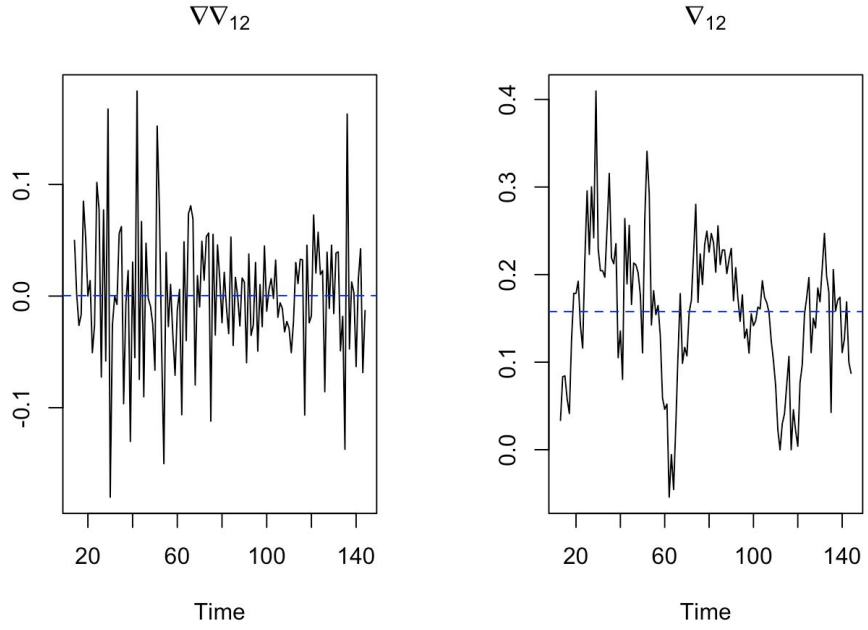
2.3 Remove Trend

The original time series plot, and decomposition plot implied there was an upward trend in observations. To remedy this we then differenced at lag 1 on our already transformed data. It decreased the variance and improved the time plot. We then differenced again at lag 1 to see if that would also improve our data. The variance increased suggesting overdifferencing.

Time

Model	Variance
Original	0.338462559648484
Differenced at lag 12	0.0064231034991497
Differenced at lag 12 and lag 1	0.00360076262606474
Differenced at lag 12, and lag 1 twice	0.00971869975429481

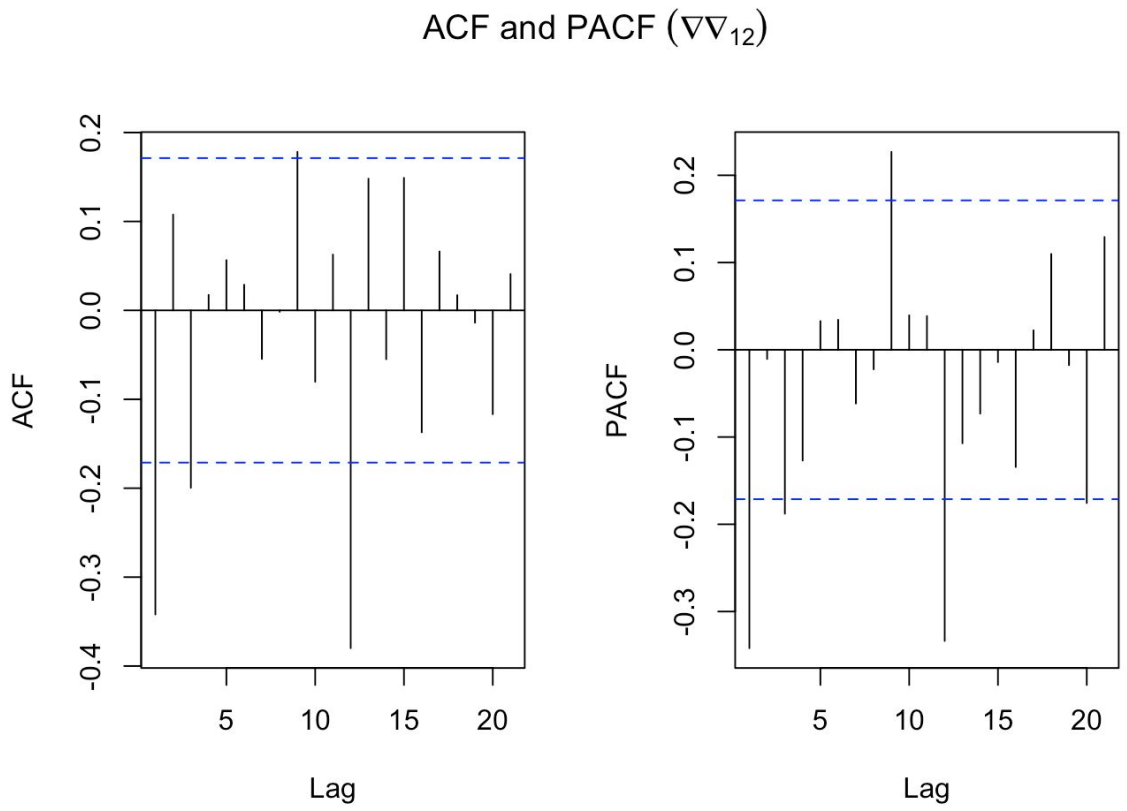
Therefore, we concluded the ideal transformation to be $\nabla \nabla_{12}$. The resulting time series plot also appears to be more stationary then the previous plots.



3.0 Model Fitting

3.1 Initial Model Estimation

To begin fitting a model to our data, we plotted the ACF and PACF graphs of the differenced dataset. The ACF graph spikes at lag 12, suggesting a SMA(1) component to our ideal model. The PACF graph is a little bit harder to read, it seems that there could be a spike at lag 12, or that it is tailing off.



3.2 Model Selection

Our objective is to fit an appropriate SARIMA model to the data. From our previous exploration we know that the appropriate d=1 and D=1. We wished to test various possibilities for AR, MA, SAR, and SMA aspects to a model. To do this, we created a for loop that looped through possible values for p,q,P,and Q. We preset d=1, and D=1. We let p and q range from 0 to 2 and P and Q from 0 to 1. We

looked at the models with the lowest AIC values. We then chose to continue with the 2 models with the lowest AIC values.

p	q	P	Q	AIC
0	1	0	1	-413.013574760653
2	1	0	1	-412.510499172408
1	2	0	1	-412.142890734553
0	1	1	1	-411.563746157757
1	1	0	1	-411.439146736834

3.3 Model Fitting

We fit our selected two models: SARIMA $(0, 1, 1) \times (0, 1, 1)_{12}$ and SARIMA $(2, 1, 1) \times (0, 1, 1)_{12}$ with the Arima function in R that utilized the maximum likelihood method.

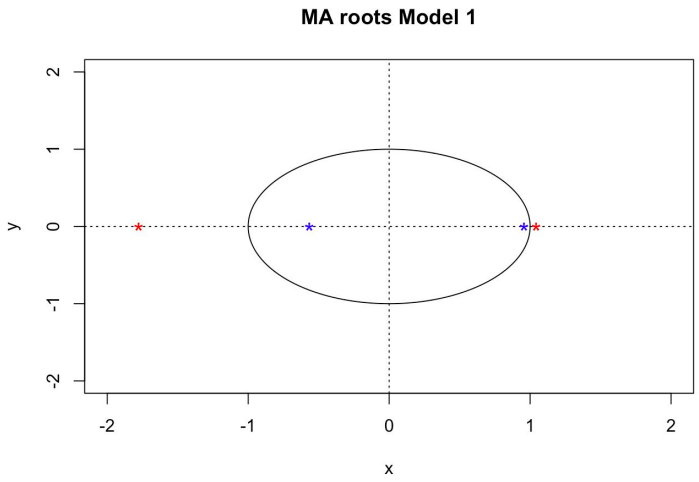
Table 2: Coefficients of SARIMA models

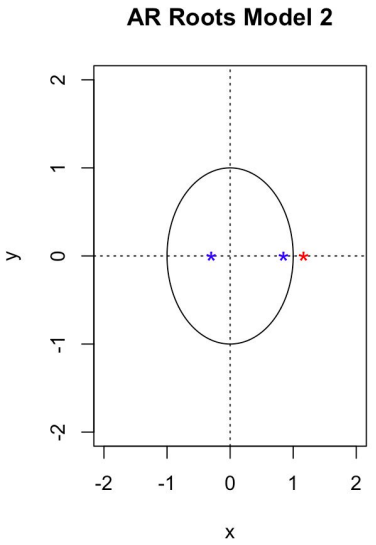
	AR(1)	AR(2)	MA(1)	SMA(1)
Model 1			-0.3947	-0.5397
Model 2	0.5639	0.2486	-0.9737	-0.5406

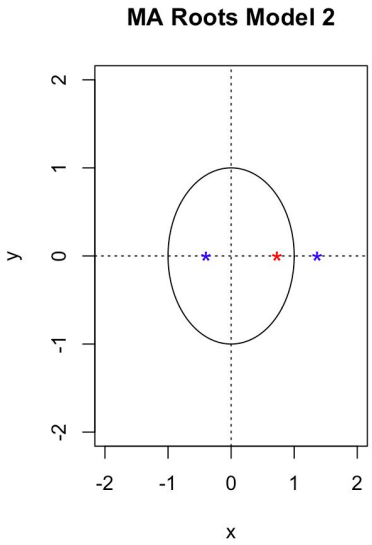
4.0 Diagnostics

4.1 Root Tests

There appears to be an MA root within the unit circle for the SARIMA$(2,1,1)(0,1,1)_{12}$ model. This indicates non-invertibility.

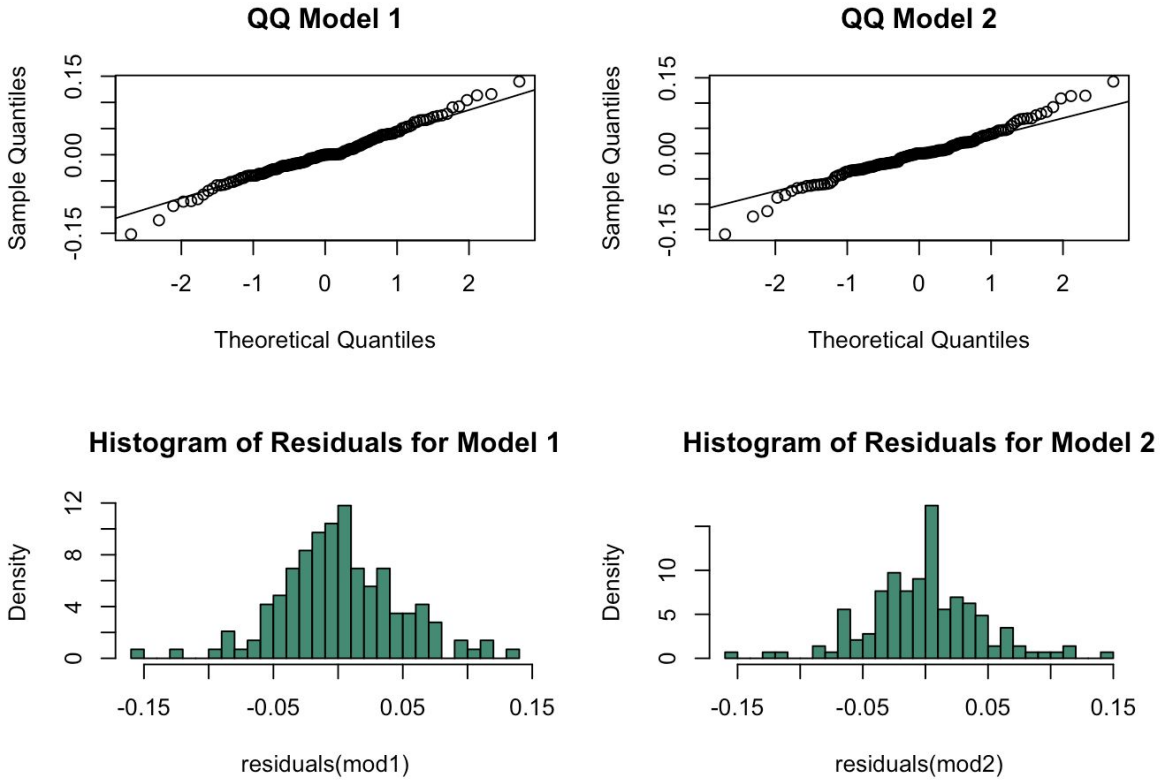






4.2 Normality

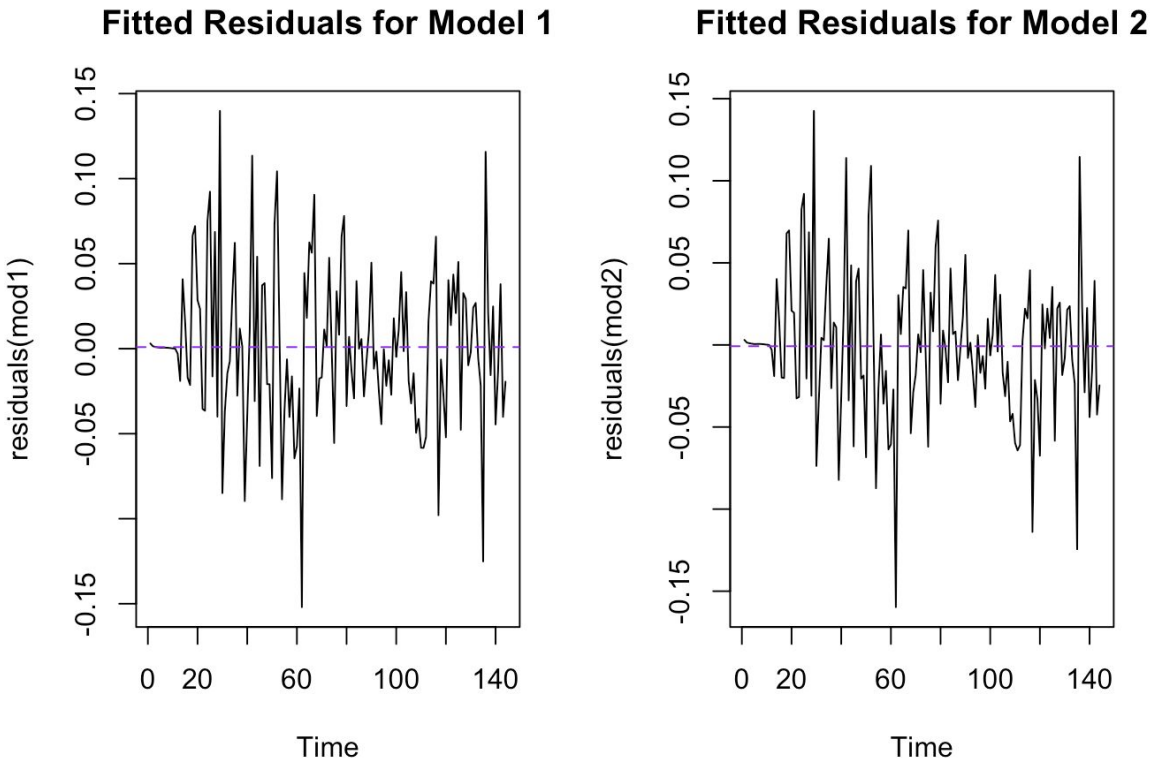
To check if the residuals of our model are normally distributed we fit our data on QQ plots and histograms. The QQ plot displayed tells us our data is normal with most of our plot points falling on the normal distribution with the exception of a few outliers on each side. The histogram shown for model 1 and model 2 are similar to the shape of a normal distribution plot. Therefore we can conclude the QQ Plots and histograms for our models appear to be reasonably normal.



The graphs below display our residuals plotted against time. To further diagnose our model we use the Shapiro-Wilk Normality to test if the residuals are approximately independent and identically distributed as Gaussian. For this test, H_0 = assumption that the residuals are normally distributed and H_1 = The residuals are not normal. Model 1 passes the Shapiro-Wilk Normality test with a p-value(α) above 0.05, Model 2 does not. Therefore we can reject normality of the residuals for the second model and accept it for Model 1.

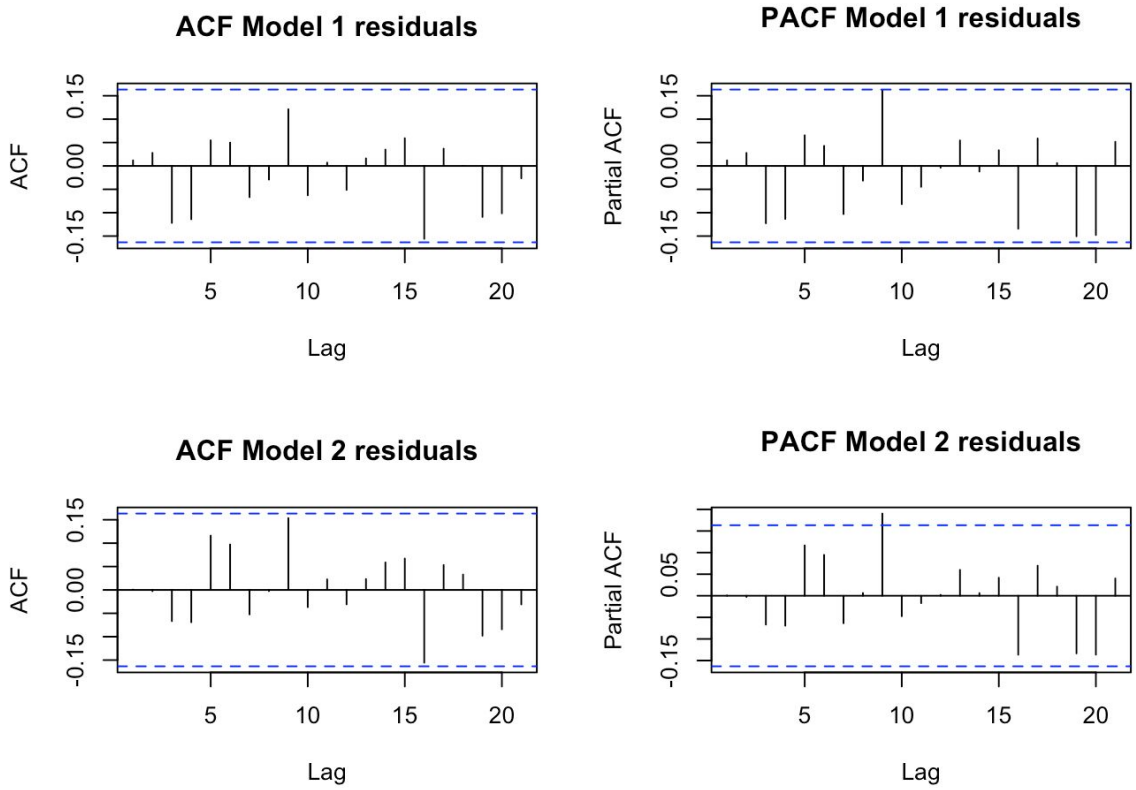
Table 3: Shapiro-Wilk Normality Test

	W-Statistic	P-Value
Model 1	0.987	0.1887666
Model 2	0.978	0.0215862



4.3 Heteroscedasticity

We want to check for heteroscedasticity which is a violation of constant variance. The ACF and PACF of the residuals for model 1 fall within the 95% confidence interval so they can all be counted as 0's. For model 2, there appears to be one point in the PACF that reaches above the confidence interval at approximately lag 9. No heteroscedasticity is detected within model 1.



4.4 Independence (Serial Correlation)

We perform two tests: the Ljung-Box Test and the Box-Pierce Test with both at level α=.05. The null hypothesis is that the residuals are serially uncorrelated. The alternative hypothesis that the residuals are not serially uncorrelated. All p-values are above 0.05, so we do not reject the assumption of serial correlation between the residuals in either model.

Table 4:

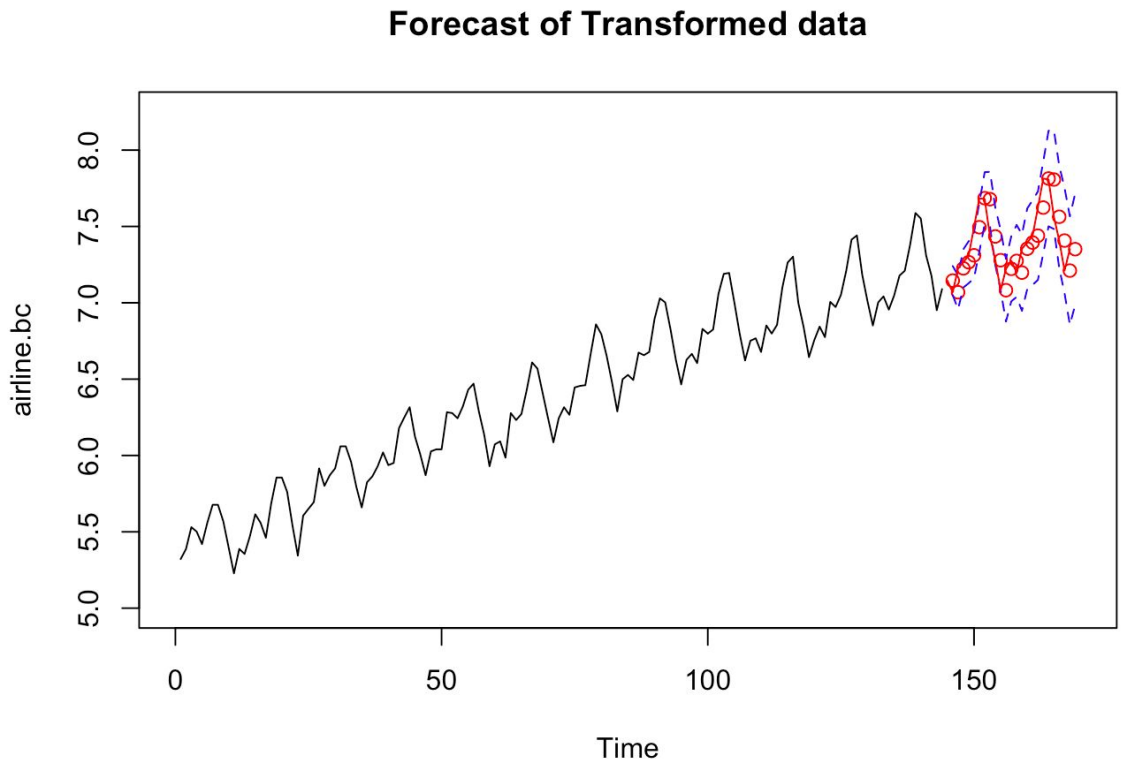
	Box-Pierce	Ljung-Box
Model 1	0.8862201	0.8850410
Model 2	0.9924235	0.9923444

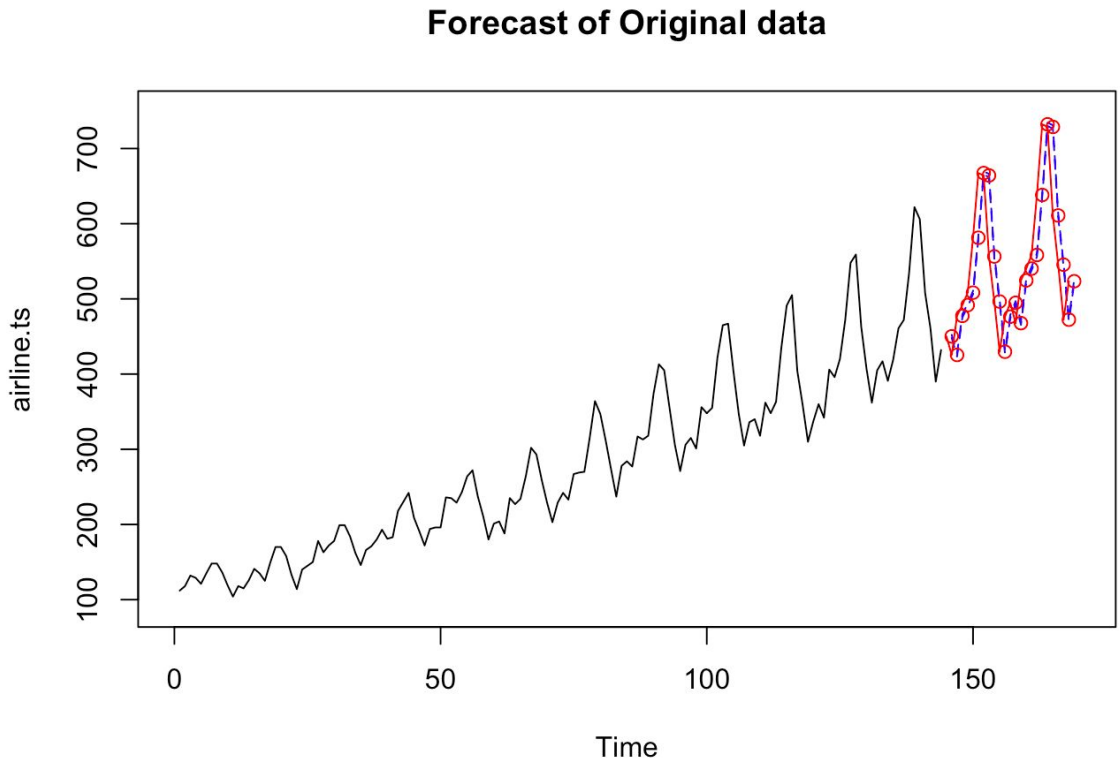
Both models pass our diagnostic checks. We decide to use the SARIMA $(0,1,1) \times (0,1,1)_{12}$ order model because it passes all diagnostic checks. Therefore, we conclude that our best fit model is:

$$X_t \nabla\nabla_{12} = Z_t (1\text{-}0.3947B)(1\text{-}0.5397 B^{12})$$

5.0 Forecasting

Here we forecast the next 24 months of the box-cox transformed data and the original data. We give the confidence intervals in both plots given by the blue lines, and our forecasted points fall within our confidence intervals. The forecasted data is reflective of the periodic cycle and indicates that our model is successful.





Conclusion

Our goal with our data analysis is to use time series techniques to analyze the monthly international airline passenger numbers to make inferences and discover trends. Our goals were achieved as we fitted a S*ARIMA*(0,1,1)x$(0,1,1)_{12}$ model with our dataset and were able to analyze it. We've predicted the next 24 estimates for each month's international airline passengers with the graph above. Also by taking a look at the decomposition plot, we notice as spring start till the summer months end, we see an upward trend for that time period in that particular year . Our downward trend in a particular year of the data begins once the fall season begins but ends up recovering once the winter season begins. We can infer from this plot the warmer climate in the summer makes for a great time to travel. Also we could assume that there could be more time for vacations during this time period as significantly more people travel. The spike from november to december/january can be explained by the demand to travel during the holiday season as people work less and are free to travel or visit family/friends.

References

1. International airline passengers: monthly totals in thousands. Jan 49 – Dec 60, Retrieved February 15, 2019, from https://datamarket.com/data/set/22u3/international-airline-passengers-monthly-totals-in-thousands-jan-49-dec-60#!ds=22u3&display=line.

Appendix

```
knitr::opts_chunk$set(echo=TRUE,
                      cache=FALSE,
                      fig.align='center')
indent1 = '    '
indent2 = paste(rep(indent1, 2), collapse='')

library(dplyr)
library(robustbase)
library(qpcR)
library(rgl)
library(MuMIn)
library(forecast)
library(MASS)
library(kableExtra)
library(TSA)
library(tseries)
library(astsa)
```

Introduction/describe data

```
airline <- read.csv("~/Downloads/international-airline-passengers.csv",
                    header=TRUE) #read in file
airline<- airline[1:144,]
airline.ts<- ts(airline[,2]) #convert data into time series format
airline.seasonal = ts(airline[,2], start=c(1949,01),
                      frequency = 12)
plot.ts(airline.ts, main = "International airline passengers",
        xlab = "Time (Months)", ylab="Monthly totals in thousands")
```

```
seasonplot(airline.seasonal, year.labels = TRUE,
           year.labels.left=TRUE, col=rainbow(20), pch=19,
           main = "Seasonal Plot", xlab = "Month", ylab = "Monthly totals in thousands")
```

```
plot(stl(airline.seasonal, s.window="periodic"), main= "Decomposition Plot")
```

Transformations

```
# three transformations
#boxcox
lambda<- BoxCox.lambda(airline.ts, method = "loglik") #lambda=0.05
airline.bc<- forecast::BoxCox(airline.ts, lambda=lambda) #tranform data

#log
airline.log <- log(airline.ts)

# sqrt
airline.sqrt <- sqrt(airline.ts)
```

```
#compare original and transformed data
par(mfrow=c(2,2))
ts.plot(airline.ts, main = "Original Data")
abline(h = mean(airline.ts),lty = 2,col="red")
ts.plot(airline.bc,main = "Box-Cox")
abline(h = mean(airline.bc),lty = 2,col="orange")
ts.plot(airline.log,main = "Log")
abline(h = mean(airline.log),lty = 2,col="green")
ts.plot(airline.sqrt,main = "Square-Root Transformed Data")
abline(h = mean(airline.sqrt),lty = 2,col="blue")
```

```
vars<- data.frame(matrix(ncol=4, nrow=0))
vars<- (rbind(vars, c(var(airline.ts), var(airline.bc),
                      var(airline.log), var(airline.sqrt))))
colnames(vars)=c("Original", "Box-Cox", "Log", "Square Root")
kable(vars, caption = "Variances", booktabs=T) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = 'center')

##We will use boxcox
```

Remove Seasonality

```
#difference at lag=12 to remove seasonality component
airline.d12 <- diff(airline.bc,lag=12, differences=1)
par(mfrow=c(1,2))
ts.plot(airline.d12, main = expression(nabla[12]), ylab="")
abline(h = mean(airline.d12),lty = 2,col="blue")
ts.plot(airline.bc, main = "Original data", ylab="", xlab = "Time")
abline(h = mean(airline.bc),lty = 2,col="blue")
```

```
#show variances before and after differencing
vars.dif<- data.frame()
vars.dif<- (rbind(vars.dif, c("Original", var(airline.bc)),
                  c("Differenced at lag 12", var(airline.d12)),
                  stringsAsFactors=F))
colnames(vars.dif)=c("Model", "Variance")
#kable(vars.dif, booktabs=T) %>%
#  kable_styling(bootstrap_options = "striped", full_width = F, position = 'center')
```

Remove trend

```
airline.d12.d1 <- diff(airline.d12,lag=1, differences=1) #difference at lag 1
par(mfrow=c(1,2))
ts.plot(airline.d12.d1, main = expression(nabla*nabla[12]), ylab="")
abline(h = mean(airline.d12.d1),lty = 2,col="blue")
ts.plot(airline.d12, main = expression(nabla[12]), ylab="", xlab = "Time")
abline(h = mean(airline.d12),lty = 2,col="blue")
```

```
airline.test <- diff(airline.d12.d1, lag=1, differences=1) #difference again to test


vars.dif<- (rbind(vars.dif, c("Differenced at lag 12 and lag 1",
```

```
                              var(airline.d12.d1)),
                  c("Differenced at lag 12, and lag 1 twice", var(airline.test))))
kable(vars.dif, booktabs=T) %>%
  kable_styling(bootstrap_options = "striped",
                full_width = T, position = 'center') %>%
  column_spec(1:2, bold = T) %>%
  row_spec(3, bold = T, color = "white", background = "#D7261E")
```

Initial Model Estimation

```
par(mfrow=c(1,2))
acf(airline.d12.d1,main="")
pacf(airline.d12.d1,ylab="PACF",main="")
title(expression("ACF and PACF "(nabla*nabla[12])),outer=T,line=-1)
```

Fitting a SARIMA model

```
chart<- data.frame() ##for loop to identify p,q,P,Q using AIC

for (p in 0:2) {
  for (q in 0:2) {
    for (P in 0:1){
      for (Q in 0:1){
        values<- c(p,q,P,Q, arima(airline.bc, order = c(p, 1, q),
                                  method = c("ML"),
                                  seasonal=list(order=c(P,1,Q), period=12))$aic)
        chart=rbind(chart,values)
      }
    }
  }
}

colnames(chart)=c("p", "q", "P", "Q", "AIC")
chart2 <- chart[order(chart$AIC),]
chart2[1:5, ] %>%
  mutate(AIC = cell_spec(AIC, color = "white", bold = T,
                         background = spec_color(1:5, end = 0.9,
                                                 option = "A", direction = -1))) %>%
  kable(escape = F) %>%
  kable_styling(c("striped", "condensed"), full_width = F)
```

```
#Model 1:
mod1<- arima(airline.bc, order = c(0, 1, 1),
             method = c("ML"), seasonal=list(order=c(0,1,1), period=12))

#Model 2:
mod2<- arima(airline.bc, order = c(2, 1, 1),
             seasonal=list(order=c(0,1,1), period=12, method = c("ML")))

model.coef<- data.frame()
mod1.coef<- c("","",-0.3947, -0.5397)
```

```r
mod2.coef<- c(0.5639,  0.2486,  -0.9737,  -0.5406)
model.coef<- rbind(model.coef, mod1.coef, mod2.coef)
colnames(model.coef)=c("AR(1)", "AR(2)", "MA(1)", "SMA(1)")
rownames(model.coef)=c("Model 1", "Model 2")
kable(model.coef, caption= "Coefficients of SARIMA models", booktabs=T) %>%
  kable_styling(bootstrap_options = "striped", full_width = F)
```

Diagnostics

```r
# path of your working directory
source("plot.roots.R.txt")
source("spec.arma.R")
#model 1
#Check models for unit roots
plot.roots(NULL,polyroot(c(1, -0.3947 , -0.5397)),
           main="SARMA(0,1,1)x(0,1,1)_12 roots of MA part")
```

```r
#model 2
#Check models for unit roots
par(mfrow=c(1,2))
plot.roots(NULL,polyroot(c(1, -0.5639 , -0.2486)),
           main="SARIMA(2,1,1)x(0,1,1)_12 roots of AR part")
plot.roots(NULL,polyroot(c(1, -0.9737 , -0.5406)),
           main="SARIMA(2,1,1)x(0,1,1)_12 roots of MA part")
```

Normality checking

```r
#Plot qq and histograms of residuals to check for normality
par(mfrow=c(2,2))
qqnorm(residuals(mod1), main="QQ Model 1")
qqline(residuals(mod1))
qqnorm(residuals(mod2), main="QQ Model 2")
qqline(residuals(mod2))
hist(residuals(mod1), main="Histogram of Residuals for Model 1",
     breaks=30, col="aquamarine4", freq = F)
hist(residuals(mod2), main="Histogram of Residuals for Model 2",
     breaks=30, col="aquamarine4", freq = F)
```

```r
# Test for normality of residuals
par(mfrow=c(1,2))
ts.plot(residuals(mod1),main = "Fitted Residuals for Model 1")
abline(h = mean(residuals(mod1)),lty = 2,col="blueviolet")
ts.plot(residuals(mod2),main = "Fitted Residuals for Model 2")
abline(h = mean(residuals(mod2)),lty = 2,col="blueviolet")
```

```r
shap.1<- shapiro.test(residuals(mod1))
shap.2<- shapiro.test(residuals(mod2))


shapiro.tests<- data.frame()
shapiro.tests<- rbind(shapiro.tests, c(round(shap.1$statistic,
```

```
                                              digits=3), shap.1$p.value),
                        c(round(shap.2$statistic, digits=3), shap.2$p.value))
colnames(shapiro.tests)=c("W-Statistic", "P-Value")
rownames(shapiro.tests)=c("Model 1", "Model 2")

kable(shapiro.tests, caption= "Shapiro-Wilk Normality Test", booktabs=T) %>%
  kable_styling(bootstrap_options = "striped", full_width = F)
```

Heteroscedasticity checking

```
par(mfrow=c(2,2))
acf(residuals(mod1), main="ACF Model 1 residuals")
pacf(residuals(mod1), main="PACF Model 1 residuals")
acf(residuals(mod2), main="ACF Model 2 residuals")
pacf(residuals(mod2), main="PACF Model 2 residuals")
```

Independence (Serial Correlation) Checking

```
#Ljung Box
ljung.1<- Box.test(residuals(mod1), type = "Ljung")
ljung.2<- Box.test(residuals(mod2), type = "Ljung")

#Box Pierce
pierce.1<- Box.test(residuals(mod1), type = "Box-Pierce")
pierce.2<- Box.test(residuals(mod2), type = "Box-Pierce")

corr.tests<- data.frame()
corr.tests<- rbind(corr.tests, c(pierce.1$p.value, ljung.1$p.value),
                   c(pierce.2$p.value, ljung.2$p.value))

colnames(corr.tests)=c("Box-Pierce", "Ljung-Box")
rownames(corr.tests)=c("Model 1", "Model 2")

kable(corr.tests, caption= "", booktabs=T) %>%
  kable_styling(bootstrap_options = "striped", full_width = F)
```

Forecasting

```
mypred2 <- predict(mod1, n.ahead=24)
ts.plot(airline.bc, main="Forecast of Transformed data", xlim=c(0,170), ylim=c(5,8.25))
points(146:169,mypred2$pred, col="red")
lines(146:169,mypred2$pred+1.96*mypred2$se,lty=2, col="blue")
lines(146:169,mypred2$pred-1.96*mypred2$se,lty=2, col="blue")
lines(mypred2$pred, col="red")
```

```
mypred.original<- forecast::InvBoxCox(mypred2$pred, lambda=lambda) #inv box cox to revert to original
mypred.original.se<- InvBoxCox(mypred2$se, lambda=lambda)
ts.plot(airline.ts,  main="Forecast of Original data", xlim=c(0,170), ylim=c(90,750))
lines(146:169,mypred.original+1.96*mypred.original.se,lty=2, col="blue")
```

```r
lines(146:169,mypred.original-1.96*mypred.original.se,lty=2, col="blue")
points(146:169,mypred.original, col="red")
lines(mypred.original, col="red")
```