Harvey Lao 174 Project

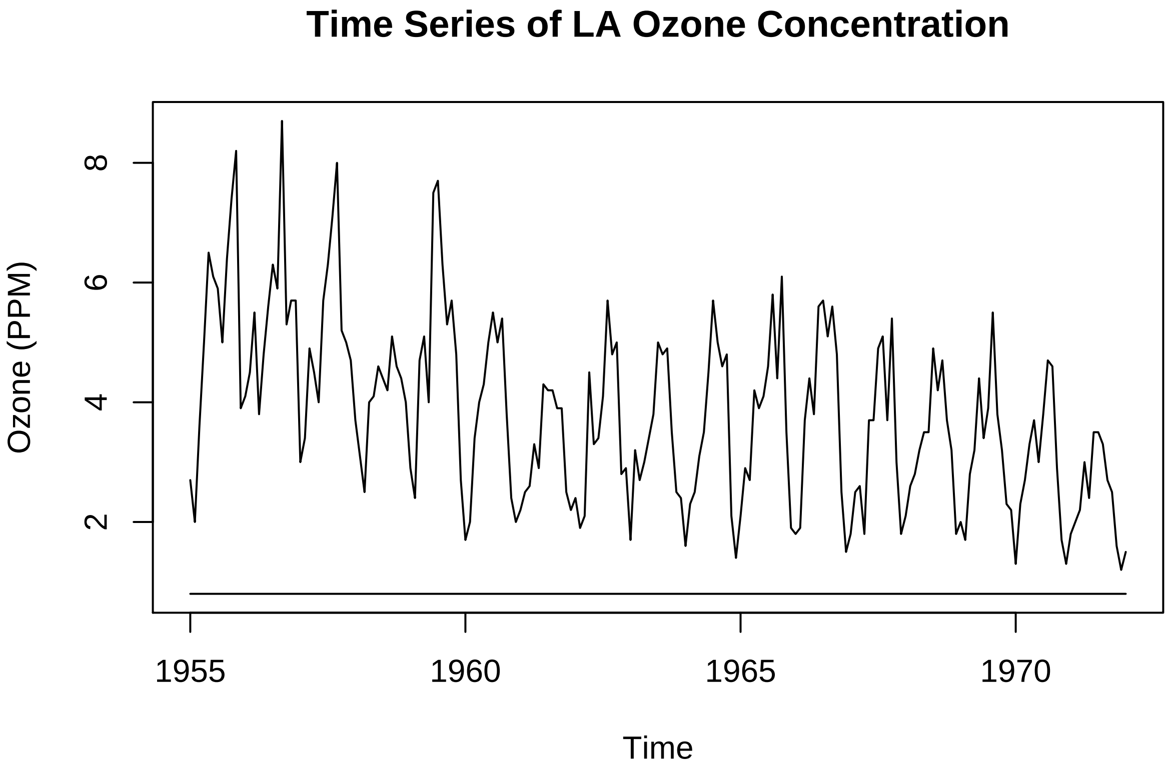
Harvey Lao

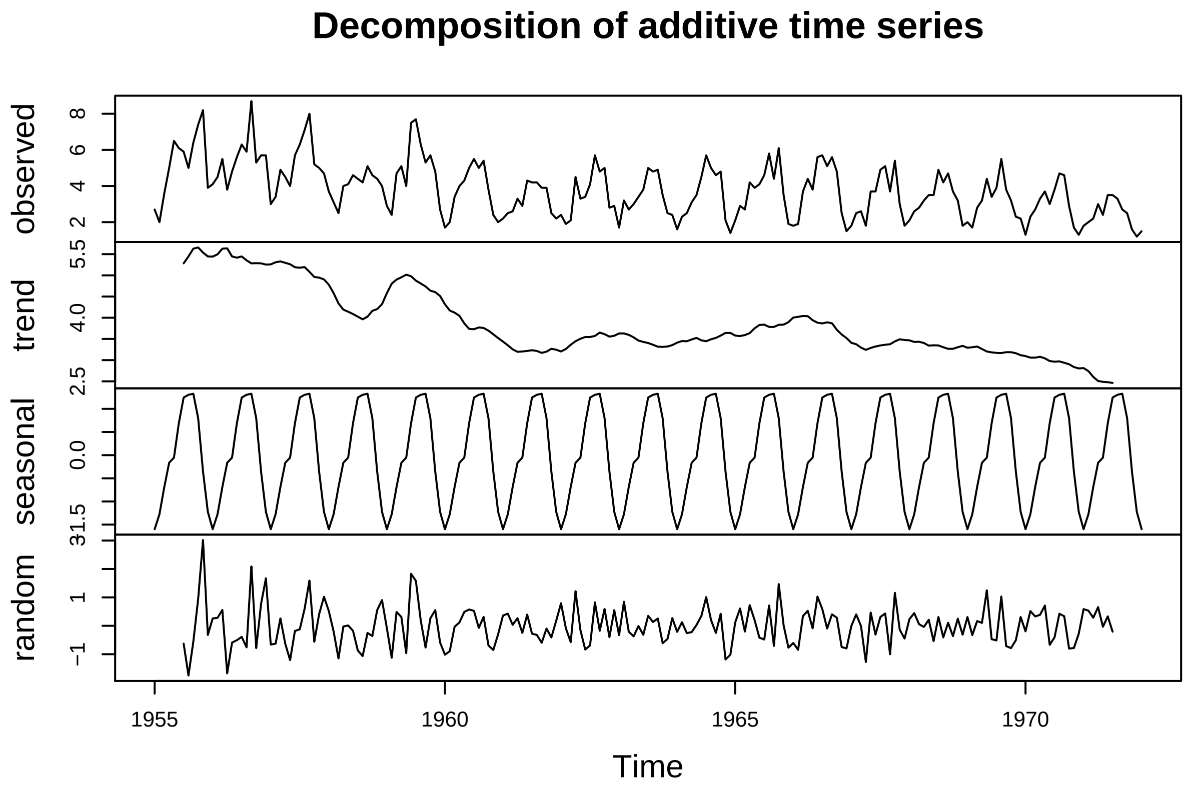
December 14, 2017

Abstract

Watching the ozone layer in our society is an important task because the ozone layer is what protects the earth from dangerous ultra-violet rays. With the recent Thomas fire so close to SB and five other fire surrounding the LA area, ozone concentration is an ironically and incredibly relevant trend to study. This report is an analysis of ozone concentration in Los Angeles between 1955 and 1972. The goal of this study is to build a model that can accurately forecast parts per million of ozone in LA. By forecasting ozone concentration, we can better prepare for the future and investigate the variation and direction of ozone in Los Angeles. Using this time series data, we can apply statistical methods to measure trend and seasonality. We can fit autoregressive models, analyze seasonality, and apply transformations and differencing. Other methods in this study include testing residuals models for normality, variance reduction, and testing causality and invertibility. After obtaining an appropriate model, a whole year of predictions are forecasted and tested on the last 12 sample observations.

Plotting and Analyzing the Time Series

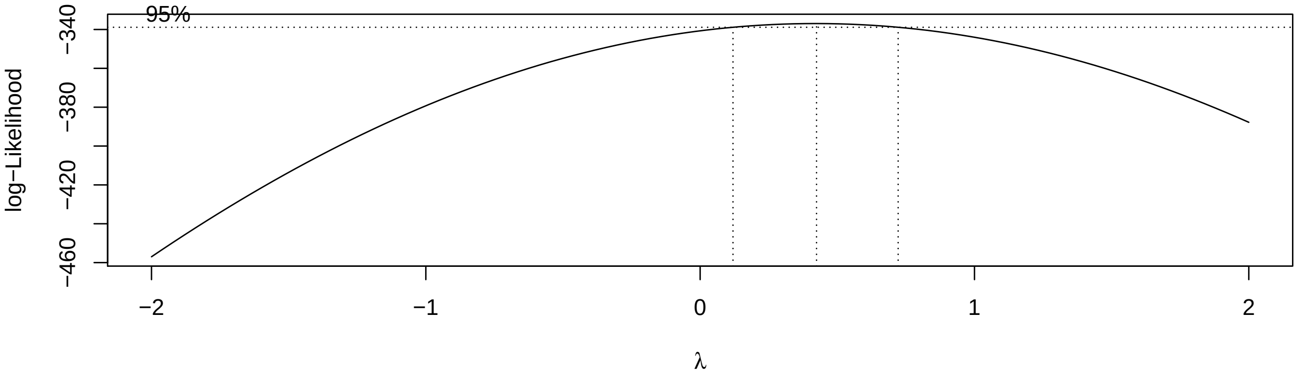




Observing the time series, we see:

- decreasing trend, change with respect to time
- strong seasonality
- some strange sharp behavior around 1958 and 1959
- potentially non-constant variance

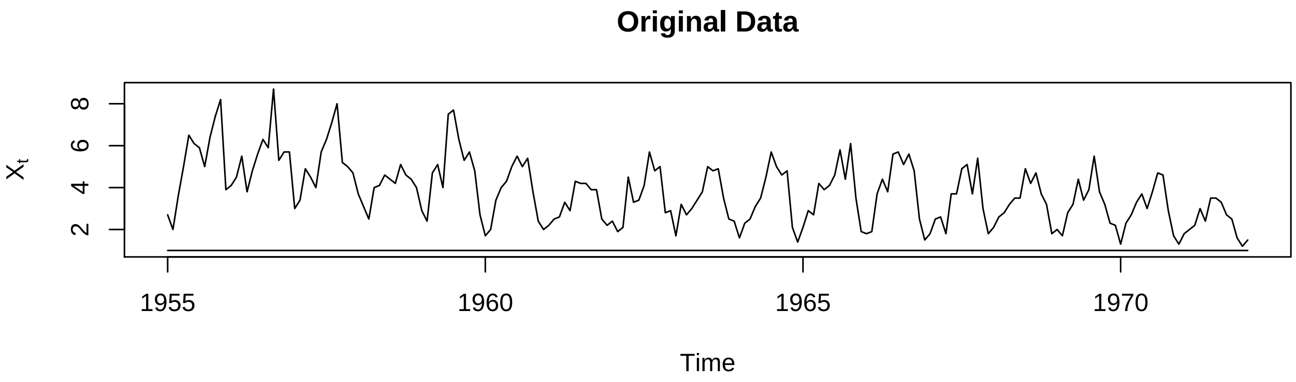
Looking at the Box-Cox interval, we see that lambda=0 does not fall within the 95% confidence interval:

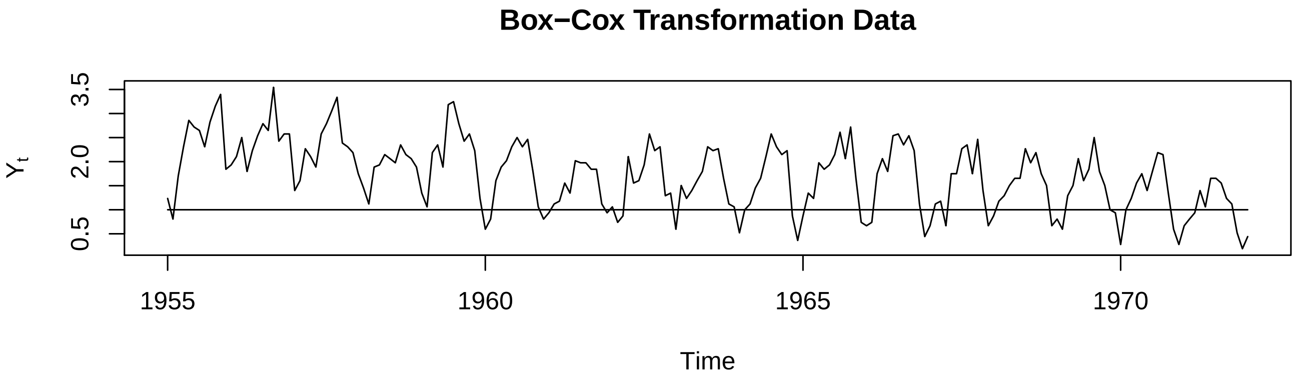


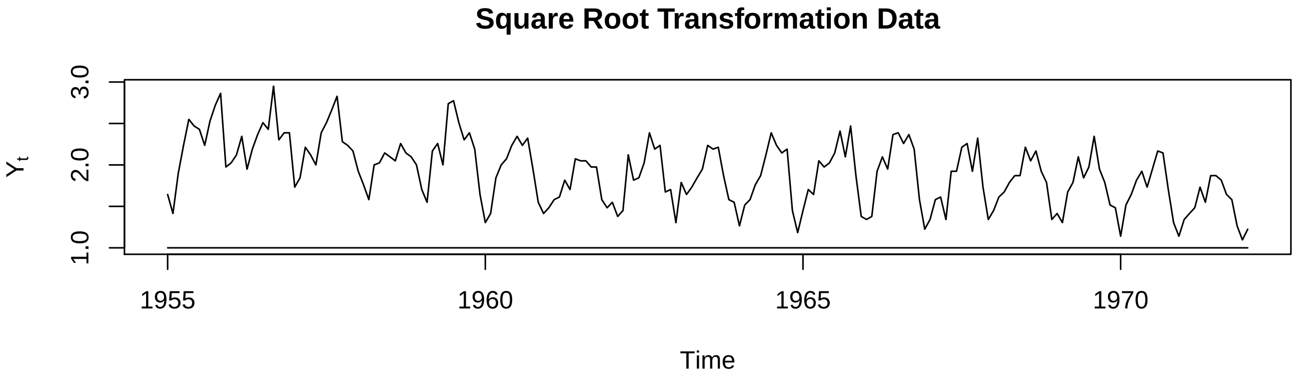
```
## Lambda is 0.4242424242
```

Because our observations of ozone concentration are all values between 0 and 9, we can consider using square root transformation. Furthermore, we observe lambda to be 0.4242, which supports our attempt to use a squre root transformation.

Transformations





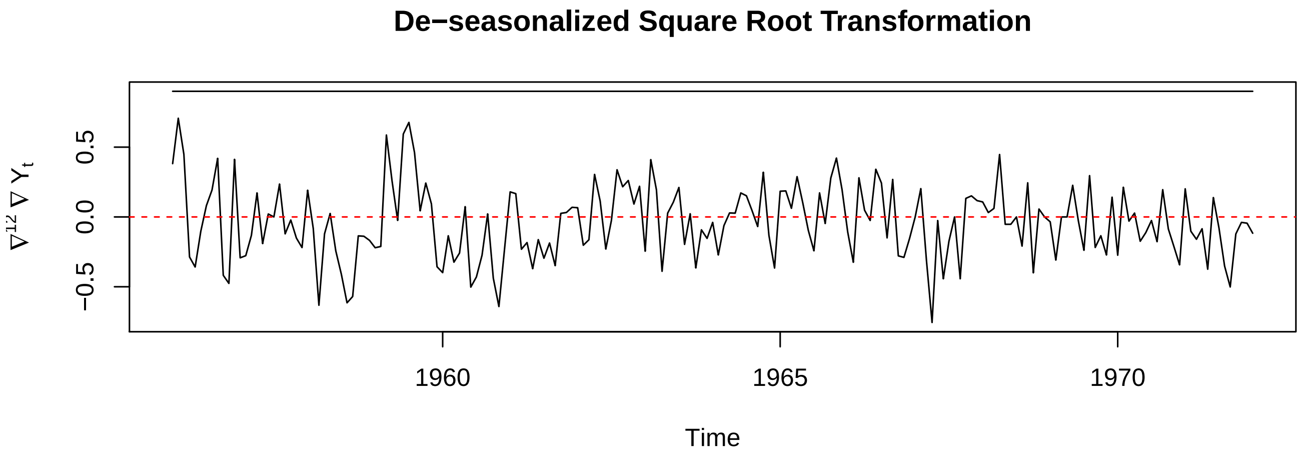


```
## The original variance was 2.251362984

## The box-cox variance was 0.4897039737

## The square root variance is 0.1485616451
```

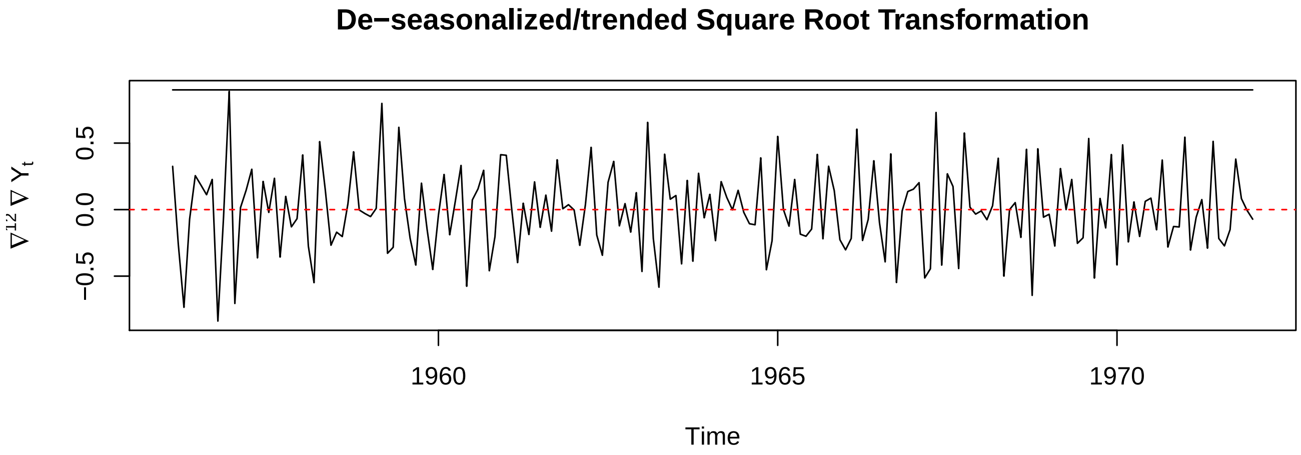
Comparing the Square Root to Box-Cox and to the original time series, we find that the plots' shapes are rather similar. However, the Square Root has a much better variance than the other transformations, and does not distort the range of the data in a significant manner. We can begin modeling with this transformation.

Differencing

First we difference at 12 because of seasonality:



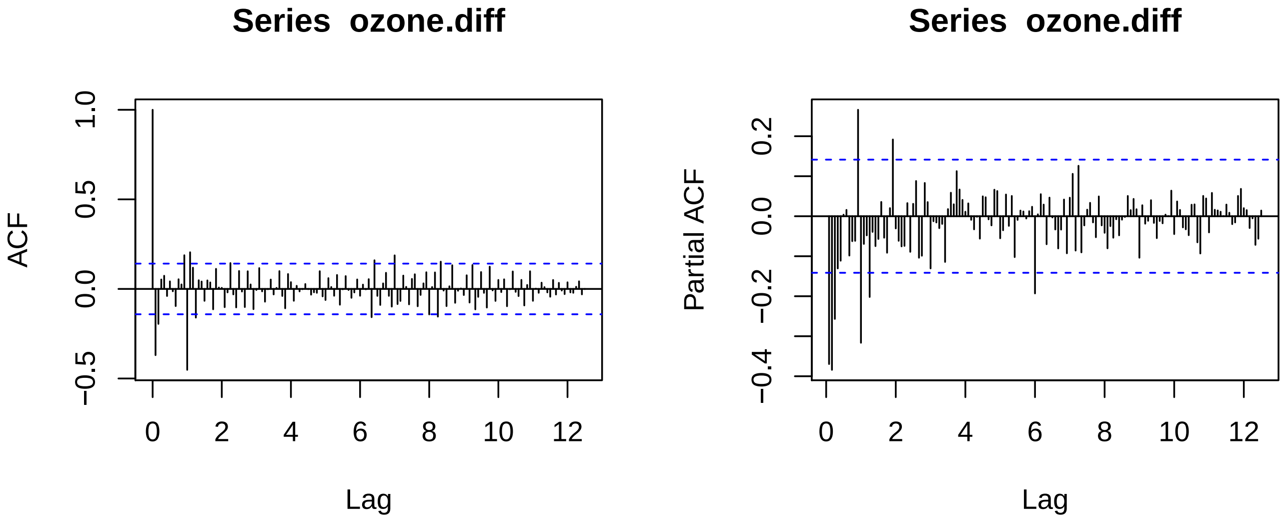
We see the seasonality has disappeared, and the mean is now 0. There is still non-constant variance present in the plot. We difference at 1 to get the following:



Although differencing at 1 increased the variance, we see that the variance is much more stable.

Analyzing ACF and PACF of Transformation

After differencing at lag 12 and lag 1, here is what we see:



The ACF and PACF of this transformation are not that clean. There are no clear models which these plots indicate, but there is sign of seasonality. We also observe a spike at lag 1 and between 1 and 2. This could be due to the irregular fluctuation around 1565 to 1559. Using the adf.test() function, we test for stationarity:

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ozone.diff
## Dickey-Fuller = -7.8603063, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

We see that differencing removed a moderate amount of noise, but there is still no clear model we can use to predict the time series. We see that the adf.test function yields a p-value of 0.01, so we can confirm our transformation is stationary. Now we attempt to generate an appropriate model.

Model Building

Because the acf and pacf are still uneasy, we have no apparent suggested models. Using the auto.arima() function to generate an optimal model gives us the following:

```
## Series: ozone.sqrt
## ARIMA(1,1,1)(2,0,0)[12]
##
## Coefficients:
##             ar1         ma1       sar1       sar2
##       0.3893377  -0.9624747  0.3595244  0.3620081
## s.e.  0.0792104   0.0231091  0.0699896  0.0715693
##
## sigma^2 estimated as 0.04683276:  log likelihood=20.37
## AIC=-30.73   AICc=-30.43   BIC=-14.14
```

We are recommended to use ARIMA(1,1,1)(2,0,0)[12]. We will test for potential models with similar seasonal components to see if the values around our suggested parameters have low AICcs and significant coefficients.

```
## [1] "D=0"
```

```
##    Q
## P              0             1             2
##   0  31.86849730   8.123069989  -3.634276932
##   1 -12.68237699 -46.867826191 -45.210950017
##   2          NA            NA            NA
```

```
## [1] "D=1"
```

```
##    Q
## P              0            1            2
##   0  28.77326599 -44.37114356 -42.56094150
##   1 -13.54199487 -42.53167427 -40.47130626
##   2 -33.76445956 -40.72192899           NA
```

Comparing the suggested model to those around it, the two best models we found are:

Model	Significant Coefficients	AICc
SARIMA(1,1,1)(1,0,1)$_{12}$	4 of 4 significant coeff	-46.8678619
SARIMA(1,1,1)(1,0,2)$_{12}$	4 of 5 significant coeff	-45.21095002

We will choose SARIMA(1,1,1)x(1,0,1)$_{12}$ as our model because it is less complex, contains a much lower AICc, and has significant coefficients. Before we forecast, we must test this model and its residuals for normality.

Model Testing and Selecting

This is a summary of our model **SARIMA(1,1,1)x(1,0,1)$_{12}$**:

```
##
## Call:
## arima(x = ozone.sqrt, order = c(1, 1, 1), seasonal = list(order = c(1, 0, 1),
##     period = 12), xreg = 1:length(ozone.sqrt), method = "ML")
##
## Coefficients:
##             ar1          ma1       sar1         sma1  1:length(ozone.sqrt)
##       0.3375076  -0.9498998  0.9705015  -0.7452183            -0.0038406
## s.e.  0.0898602   0.0398287  0.0205428   0.0754223             0.0044475
##
## sigma^2 estimated as 0.04085186:  log likelihood = 29.58,  aic = -47.17
```

Now let's test the residuals:

```
##
##  Augmented Dickey-Fuller Test
##
## data:  resid(s.101)
## Dickey-Fuller = -4.8658189, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

We see the adf.test functions yields a p-value of 0.01, so we can confirm our transformation is stationary.

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(s.101)
## W = 0.9949509, p-value = 0.7268178
```

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

```
## data:  resid(s.101)
## X-squared = 11.783578, df = 10, p-value = 0.299802
```

Our model passes both the Shapiro and Ljung-Box tests.

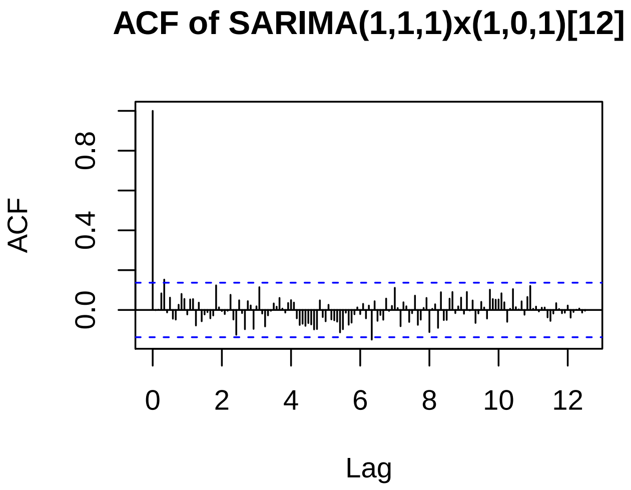
More Diagnostic Checks

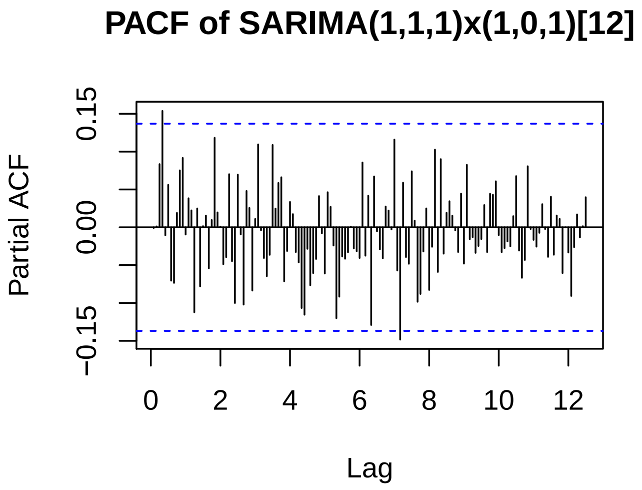
Since the model passed Shapiro and Ljung tests, we attempt the McLeod test, causality test, and invertibility test:

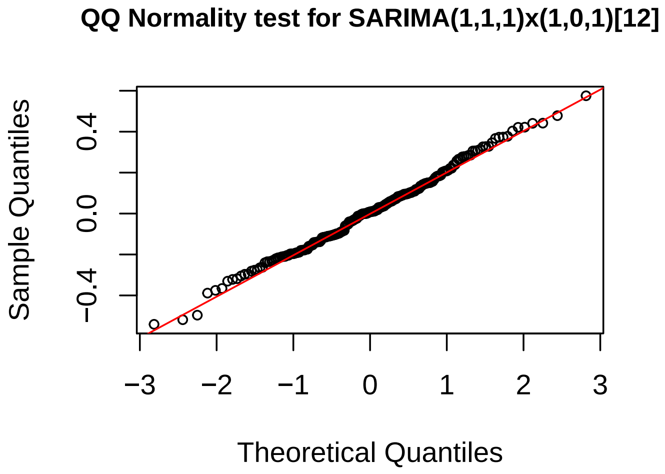
```
##
##  Box-Ljung test
##
## data:  resid(s.101)^2
## X-squared = 12.241604, df = 14, p-value = 0.586907
```

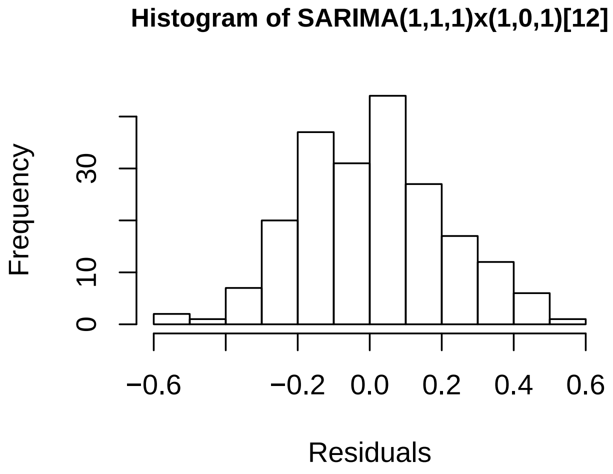
```
## Non-Causal
## Invertible
```

The model passes the McLeod Li test. Although we have a non-causal model, the model is invertible. We perform some last diagnostic checks:


ACF of SARIMA(1,1,1)x(1,0,1)[12]

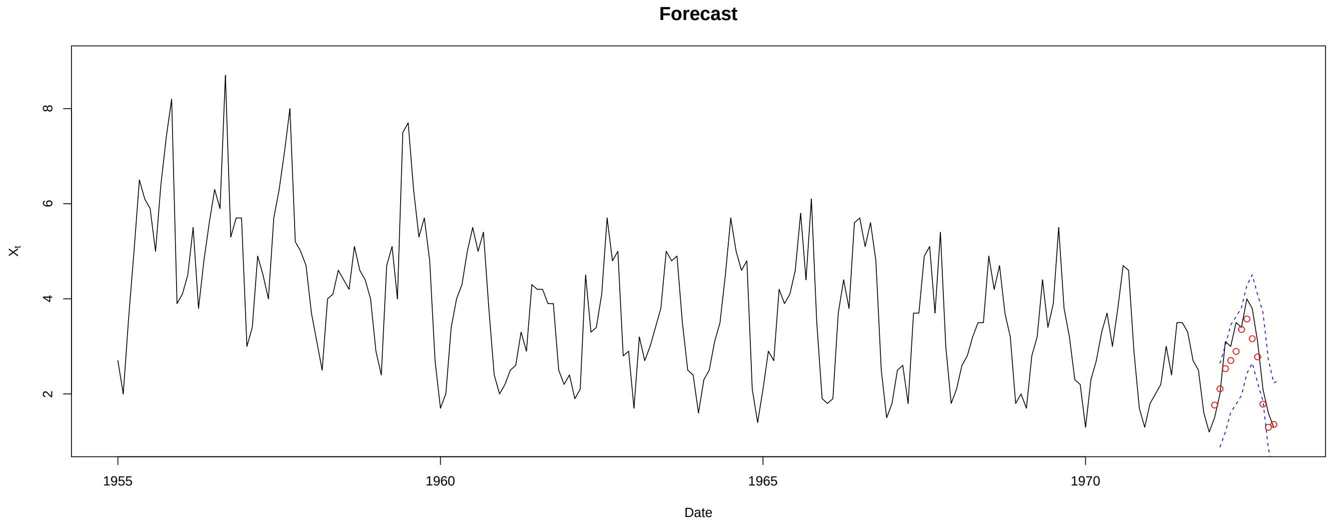

PACF of SARIMA(1,1,1)x(1,0,1)[12]


QQ Normality test for SARIMA(1,1,1)x(1,0,1)[12]


Histogram of SARIMA(1,1,1)x(1,0,1)[12]

These plots show that SARIMA(1,1,1)x(1,0,1)$_{12}$ passes the diagnositcs for normality, and is a proper model to apply forecasting.

Forecasting



Conclusion

Using many statistical tools and methods, we were able to accurately predict the next twelve months of ozone concentration in Los Angeles. Furthermore, the 12 true values we reserved as a test set falls within our 95% confidence region. We can see an apparent decreasing trend; this is a bad sign because the ozone layers protect us from dangerous ultra-violet rays of the sun. Furthermore, the time of the year is a heavy factor on the ozone concentration in Los Angeles. The ozone concentration is always at its minimum around December and January, while it steadily increases for the subsequent six months. Then after Summer, parts per million of ozone drops back down through the end of the year. This suggests that some seasonal trend is related to the fluctuation of ozone concentration. Colder climate around the new year may play a large factor in the amount of ozone in the atmosphere. For example, to keep warm, Los Angeles residents are expected to use more energy and burn more fossil fuels to keep warm during the colder season around new years. If further work were to be conducted, analyzing potential phenomenons that caused this seasonality would be quite intriguing. The final model selected for forecasting was SARIMA(1,1,1)x(1,0,1)$_{12}$.

Model Coefficients:

AR1	MA1	SAR1	SMA1
0.3375076	-0.9498998	0.9705015	-0.7452183

Final Model Equation:
$(1\text{-}0.3375076B)(1\text{-}0.9705015B^{12})Y_t = (1\text{-}0.9498998B)(1\text{-}0.7452183B^{12})Z_t$ where Z_t is Gaussian White Noise.

References

- R Studio Statistical Software Version 1.1.383

- Meteorology, Source: Hipel and McLeod (1994), in file: monthly/ozone, Description: Ozon concentration, downtown L. A., 1955-1972, https://datamarket.com/data/set/22u8/ozon-concentration-downtown-l-a-1955-1972#!ds=22u8&display=line

---

Appendix

Abstract

Plotting and Analyzing the Time Series

```r
#Obtain and plot time series
options(digits = 10)
ozone.csv = read.table('OzoneLA.csv', sep=",", header=FALSE, skip=1, nrow=205)
ozone = ts(ozone.csv[,2], start = c(1955,1), frequency = 12)
ts.plot(ozone, main = "Time Series of LA Ozone Concentration", cex.main = .8, ylab="Ozone (PPM)")
plot(decompose(ozone))
```

```r
#Box-Cox Confidence interval and visualizing lambda=0
t = 1:length(ozone)
fit = lm(ozone~t)
bcTransform = boxcox(ozone ~ t, plotit=TRUE, main = "Box-Cox and Lambda Interval")
#Calculating Lambda
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
cat("Lambda is", lambda, "\n")
```

Transformations

```r
#Box-Cox Transformation
ozone.bc = (1/lambda)*(ozone^lambda-1)
#Square Root Transformation
ozone.sqrt = sqrt(ozone)

#Plot Original, Box-Cox, Square Root
par(mfrow = c(1,1))
ts.plot(ozone, main = "Original Data", ylab = expression(X[t]), cex.main=1)
ts.plot(ozone.bc, main = "Box-Cox Transformation Data", ylab = expression(Y[t]), cex.main=1)
ts.plot(ozone.sqrt, main = "Square Root Transformation Data", ylab = expression(Y[t]), cex.main=1)
```

```r
#Finding new variances
ozone.var = var(ozone)
ozone.bc.var = var(ozone.bc)
ozone.sqrt.var = var(ozone.sqrt)

cat("The original variance was", ozone.var, "\n")
cat("The box-cox variance was", ozone.bc.var, "\n")
cat("The square root variance is", var(ozone.sqrt), "\n")
```

Differencing

```
#Differencing the square root transformation at lag = 1 and 12
par(mfrow=c(1,1))
z12 = diff(ozone.sqrt, 12)
ts.plot(z12,main = "De-seasonalized Square Root Transformation", cex.main = .9,
        ylab = expression(nabla^{12}~nabla~Y[t]))
abline(h = 0,lty = 2, col="red")
```

```
z1 = diff(z12, 1)
#Differencing again caused variance to go up, but that's fine because our variance is now constant
ts.plot(z1,main = "De-seasonalized/trended Square Root Transformation", cex.main = .9,
        ylab = expression(nabla^{12}~nabla~Y[t]))
abline(h = 0,lty = 2, col="red")
```

Analyzing ACF and PACF of Transformation

```
#Read the differenced model into a new variable name
ozone.diff=z1
#Plotting differenced ACF and PACF
par(mfrow=c(1,2))
acf(ozone.diff,lag.max=150, cex.main = .8)
pacf(ozone.diff,lag.max=150, cex.main = .8)
```

```
#Test for stationarity
adf.test(ozone.diff)
```

Model Building

```
#Auto Arima
auto.arima(ozone.sqrt)
```

```
#For loop for testing model seasonality

aiccs.d1 = matrix(NA, nr=3, nc=3)
dimnames(aiccs.d1) = list(P=0:2,Q=0:2)
#1
aiccs.d1[0+1,0+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                 list(order=c(0,1,0)), method = "ML", xreg=1:length(ozone.sqrt)))
#2
aiccs.d1[0+1,1+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                 list(order=c(0,1,1)), method = "ML", xreg=1:length(ozone.sqrt)))
#3
aiccs.d1[0+1,2+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                 list(order=c(0,1,2)), method = "ML", xreg=1:length(ozone.sqrt)))
#4
aiccs.d1[1+1,1+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                 list(order=c(1,1,1)), method = "ML", xreg=1:length(ozone.sqrt)))
#5
aiccs.d1[1+1,2+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                 list(order=c(1,1,2)), method = "ML", xreg=1:length(ozone.sqrt)))
#6
aiccs.d1[1+1,0+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                 list(order=c(1,1,0)), method = "ML", xreg=1:length(ozone.sqrt)))
#7
aiccs.d1[2+1,1+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                list(order=c(2,1,1)), method = "ML", xreg=1:length(ozone.sqrt)))
#8 #this one is NA (potentially non-convergent)
#aiccs.d1[2+1,2+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal
#=list(order=c(2,1,2)), method = "ML", xreg=1:length(ozone.sqrt)))
#9
aiccs.d1[2+1,0+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                list(order=c(2,1,0)), method = "ML", xreg=1:length(ozone.sqrt)))



aiccs.d0 = matrix(NA, nr=3, nc=3)
dimnames(aiccs.d0) = list(P=0:2,Q=0:2)
#1
aiccs.d0[0+1,0+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                list(order=c(0,0,0)), method = "ML", xreg=1:length(ozone.sqrt)))
#2
aiccs.d0[0+1,1+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                list(order=c(0,0,1)), method = "ML", xreg=1:length(ozone.sqrt)))
#3
aiccs.d0[0+1,2+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                list(order=c(0,0,2)), method = "ML", xreg=1:length(ozone.sqrt)))
#4
aiccs.d0[1+1,1+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                list(order=c(1,0,1)), method = "ML", xreg=1:length(ozone.sqrt)))
#5
aiccs.d0[1+1,2+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                list(order=c(1,0,2)), method = "ML", xreg=1:length(ozone.sqrt)))
#6
aiccs.d0[1+1,0+1] = AICc(arima(ozone.sqrt, order = c(1,1,1), seasonal =
                                list(order=c(1,0,0)), method = "ML", xreg=1:length(ozone.sqrt)))

print("D=0")
aiccs.d0
print("D=1")
aiccs.d1
```

Model	Significant Coefficients	AICc
SARIMA(1,1,1)(1,0,1)$_{12}$	4 of 4 significant coeff	-46.8678619
SARIMA(1,1,1)(1,0,2)$_{12}$	4 of 5 significant coeff	-45.21095002

Model Testing and Selecting

```
#Model Testing
s.101 = arima(ozone.sqrt, c(1,1,1), seasonal = list(order=c(1,0,1), period = 12),
             method = "ML", xreg=1:length(ozone.sqrt))
#s.102 = arima(ozone.sqrt, c(1,1,1), seasonal = list(order=c(1,0,2), period = 12),
method = "ML", xreg=1:length(ozone.sqrt))

s.101
#s.102
```

```
adf.test(resid(s.101))
#adf.test(resid(s.102))
```

```
shapiro.test(resid(s.101))
#shapiro.test(resid(s.102))
```

```r
Box.test(resid(s.101), lag = 14, type = "Ljung-Box", fitdf = 4)
#Box.test(resid(s.102), lag = 14, type = "Ljung-Box", fitdf = 5)
```

More Diagnostic Checks

```r
#McLeod Li Test
Box.test(resid(s.101)^2, lag=14, type="Ljung-Box", fit=0)
#Causality and Invertibility Test
check(specify(c(1,1,1)))
```

```r
#Diagnostic check for SARIMA(1,1,1)x(1,0,1)[12]
par(mfrow=c(1,2))
acf(resid(s.101), lag.max = 150, main = "ACF of SARIMA(1,1,1)x(1,0,1)[12]")
pacf(resid(s.101), lag.max = 150, main = "PACF of SARIMA(1,1,1)x(1,0,1)[12]")
```

```r
#Residual Plots
par(mfrow=c(1,2))
qqnorm(residuals(s.101), main = "QQ Normality test for SARIMA(1,1,1)x(1,0,1)[12]", cex=.7, cex.main = .9)
qqline(residuals(s.101), col="red", cex=.7)
hist(resid(s.101), main = "Histogram of SARIMA(1,1,1)x(1,0,1)[12]", cex.main=.9, xlab="Residuals")
```

Forecasting

```r
#Plot the Forecast
par(mfrow=c(1,1))
plot(ozone.og.ts, xlim=c(1955,1973),ylim=c(1,9), ylab = expression(X[t]), xlab = 'Date',
     main='Forecast', cex.main=1.4)
space=1/12
k=0:11*space
indextoadd=(1972)+k[1:12]
points(indextoadd,pred.orig, pch=1, col="red")
lines(u, lty="dashed", col="blue")
lines(l, lty="dashed", col="blue")
```