Modeling Monthly Chickenpox Cases in New York City 1931-1972

Skyler Ataide

Abstract:

The goal of this project is to create a time-series model based on real-life data and predict future data points by means of data forecasting. The project is based on a time-series dataset of the monthly reported number of cases of chickenpox in New York City, from January 1931- June 1972. The motivation for running a time series analysis on this dataset is to gain a greater understanding of any trends/patterns that existed in the chickenpox epidemic from 1931-1972. This analysis holds great value because it gives further insight into predicting potential future spikes in the chickenpox epidemic. In present times where we currently deal with the Covid-19 pandemic, it is especially interesting to conduct research on and analyze the patterns/trends of a past epidemic in order to understand how current and future pandemics may behave over time. Additionally, time-series analysis of the chickenpox dataset can aid in understanding if the chickenpox disease is related to seasonality.

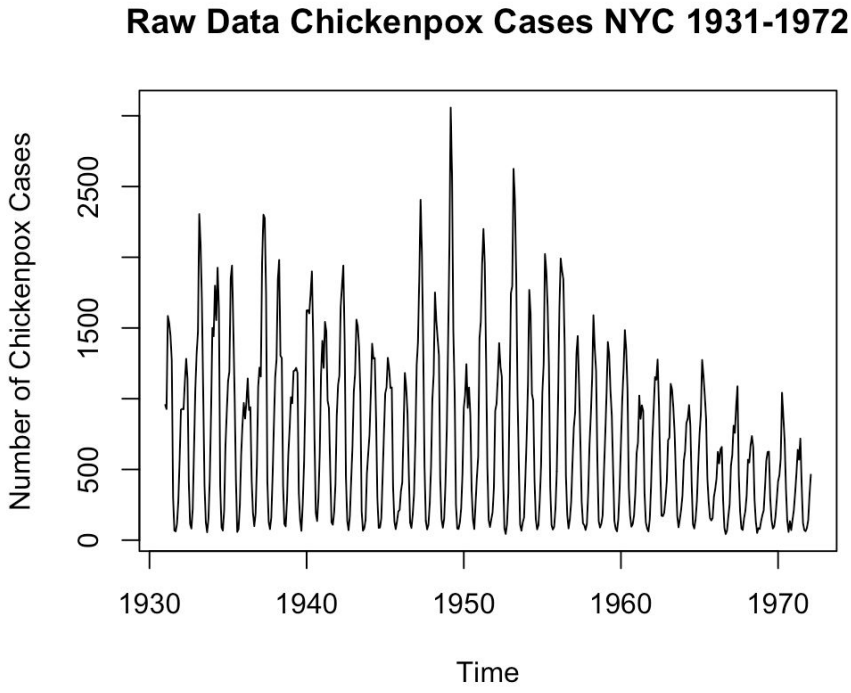
The dataset consists of 498 observations in total. Those observations are split into a training set consisting of 444 observations and a test set consisting of 54 observations, or approximately a 90/10 split respectively. First, an exploration of the original data is conducted to observe that the original data is non-stationary and displays seasonality. In order to stabilize the variance, a Box-Cox transformation is performed which yields improved results. Applying a log transformation to the data further stabilizes the data. A decomposition of the log transformed data confirms seasonality and shows that trend still exists. To best solve these problems, the data is differenced at lag 12 for seasonality and again at lag 1 for trend . The histogram of this transformed and differenced data now supports the assumption that the data is stationary. By plotting the ACF and PACF of this stationary data, we are able to determine the AR and MA terms of the model. Candidate models to fit the data are determined; $SARIMA(3,1,4)x(1,1,1)_{12}$, $SARIMA(3,1,4)x(2,1,2)_{12}$, $SARIMA(4,1,4)x(2,1,2)_{12}$, and $SARIMA(4,1,4)x(1,1,1)_{12}$, among others. Based on the AIC criterion as well as through diagnostic checking, $SARIMA(3,1,4)x(2,1,2)_{12}$ is ultimately decided on as the best model of fit. Diagnostic checking includes: Shapiro-Wilk normality test, Box-Pierce test, Box-Ljung test, and the McLeod-Li test. Plotting of the ACF and PACF of the residuals confirms that they are within the confidence intervals. In the end, twelve observations are forecasted using the best fit model.

Introduction

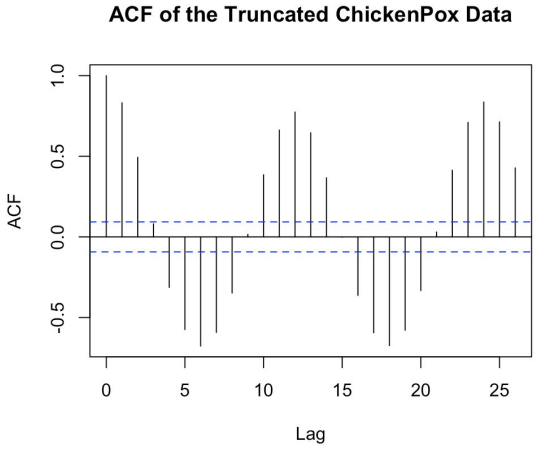
This research project will seek to address a few research questions regarding the chickenpox epidemic in New York City from 1931-1972. First, it will seek to discover whether there exists any seasonality in the number of cases in chickenpox in order to understand which seasons can be considered “peak seasons” for the epidemic. Next, it will seek to understand whether or not there exists a trend in the number of cases of chickenpox in NYC between 1931 and 1972. Finally, the overarching goal of this project will be to predict future values of chickenpox cases in NYC via forecasting. A deeper understanding of any trends/seasonality from this data can be valuable because it can help create general awareness for future spikes in the epidemic. It may also be valuable in helping to understand the trends and life cycle of future epidemics that may incur. The time series dataset used for analysis is "Monthly reported number of chickenpox, New York city, 1931-1972", from the source Hipel and McLeod (1994). The software used to analyze the time series data is R. The dataset records the monthly number of chickenpox cases in New York City beginning in January 1931 and ending in June 1972. The dataset contains a total of 498 observations, which are split into a training set of 444 observations and a test set of 54 observations. In the initial analysis of the original data, it is determined that the data is non-stationary. Differencing is used on the data in order to remove trend and seasonality, while Box-Cox and log transformations are applied to best normalize the data. AIC criterion analysis and analysis of ACF and PACF plots are conducted on the stationary data to determine the best candidate models to fit the data. Ultimately two top candidate models are determined: $SARIMA(3,1,4)x(2,1,2)_{12}$ and $SARIMA(4,1,4)x(2,1,2)_{12}$. Model diagnostic testing is run on both models including: Shapiro-Wilk normality test, Box-Pierce test, Box-Ljung test, McLeod-Li test, and analysis of residual ACF/PACF plots. Ultimately, $SARIMA(3,1,4)x(2,1,2)_{12}$ is determined as the model of best fit, with two insignificant model coefficients excluded. Using this model to forecast twelve observations ahead shows a spike in the number of cases of chickenpox in NYC, followed by a dip in the number of cases. Comparison of forecasted values with actual values from the test set confirms that our model accurately forecasts future observations.

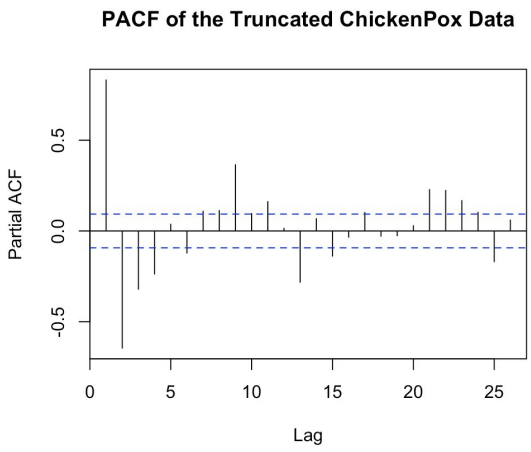
Initial Time Series Analysis

Using R, a time series of the original data was plotted to get an idea of its general form and to look for seasonality and/or trend in the data.

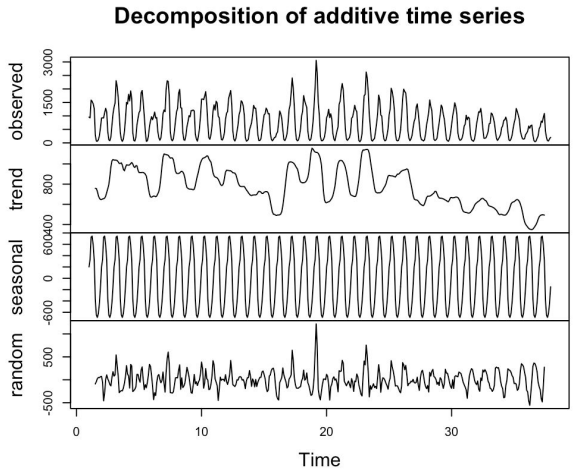


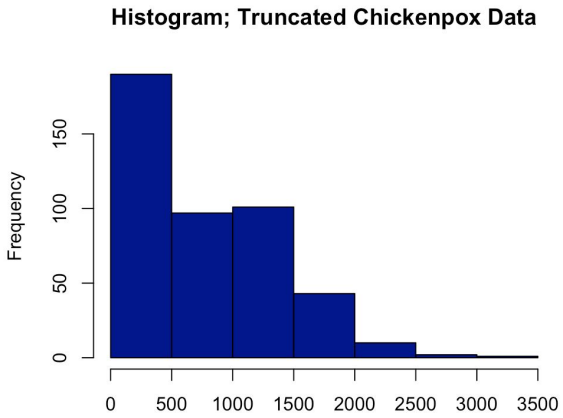
The plot of the original data does not display any obvious increasing or decreasing trends. A decreasing trend may exist roughly after the largest spike that occurs in the year 1949. The time series appears to be highly non-stationary due to changing variance and apparent seasonality.





Plots of the ACF and PACF of the truncated chickenpox data confirm non-stationarity. The plot of the ACF of the truncated data displays clear seasonality in the data. ACF's remain large and at lag 12 it appears to repeat the previous cycle. The plots of ACF and PACF validate our initial assumption that that data is non-stationary and displays trend.

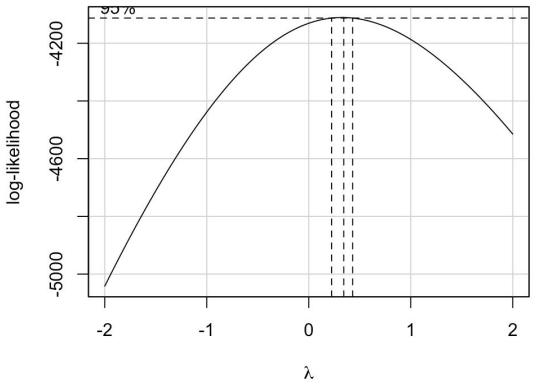


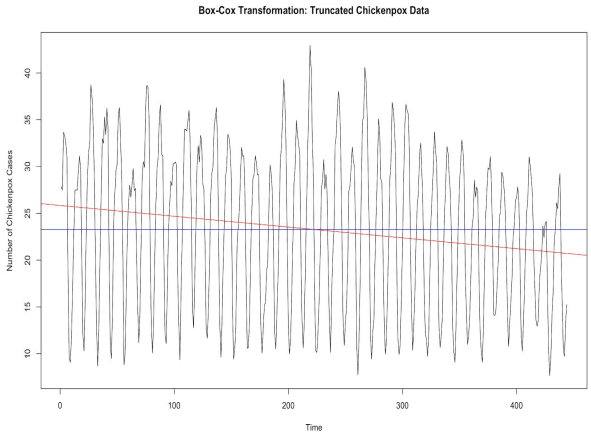


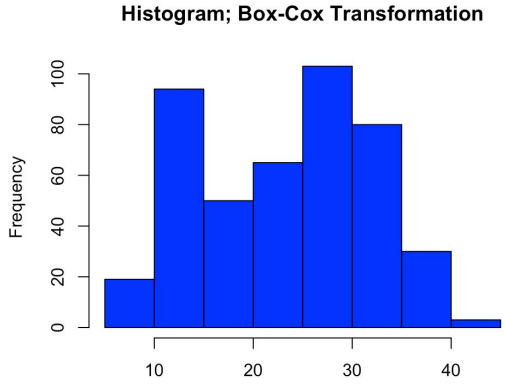
From the decomposition, the lower bound of the data appears to be constant, whereas the peak is constantly changing. From the decomposition seasonal plot, the data shows a strong seasonality pattern. The trend plot shows a more clear decreasing trend in the data than the plot of the original data. The data itself is not stationary since the variance is constantly changing. The histogram of the chickenpox data also shows that the data is badly skewed and further confirms non-stationarity.

Box-Cox Transformation

A Box-Cox transformation can help to normalize the original data. It is applied to the original data in order to help stabilize the variance. A 95% confidence interval must be applied in order to determine the optimal value of λ, which will maximize our log-likelihood.

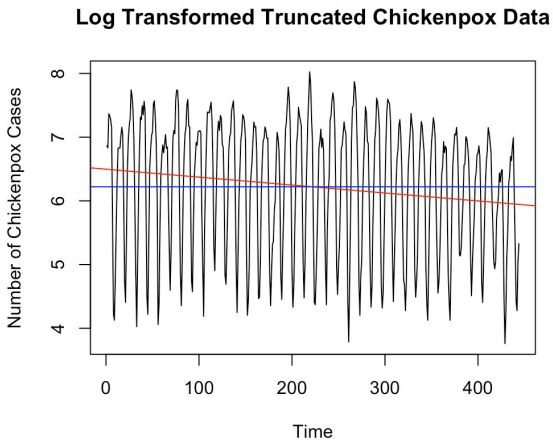


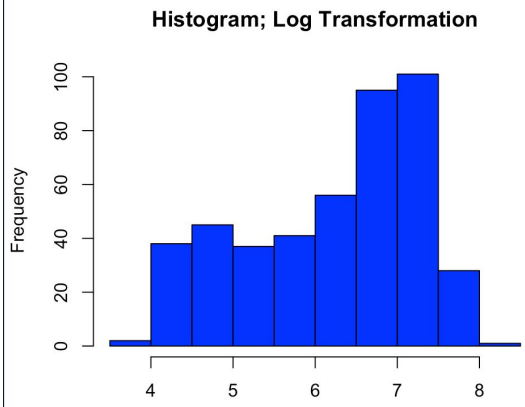




The Box-Cox transformation gives a value of λ = 0.3434343. Plotting the data with the Box-Cox transformation applied still displays trend and a slightly skewed histogram that can further be improved. Zero is not contained within the 95% confidence interval and is not the value that maximized log-likelihood. The variance of the data after the Box-Cox transformation is equal to 72.63398, which can be improved. In an effort to further stabilize the variance, a log transformation must be applied to the original data.

Log Transformation

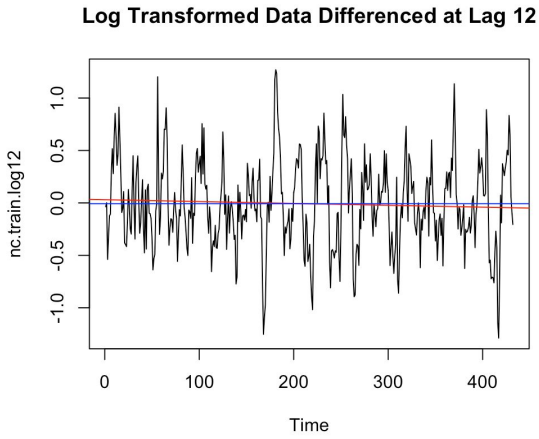


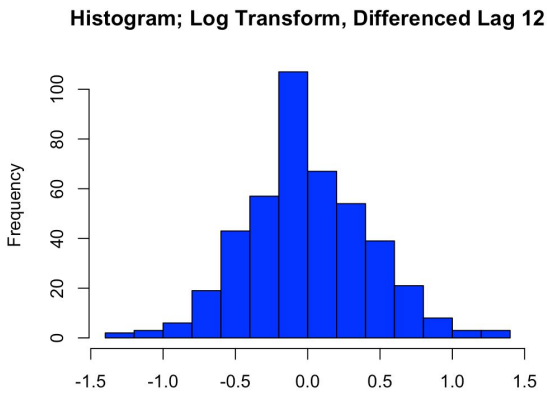


The log transformation appears to better stabilize the data than the Box-Cox transformation alone. The variance of the data after the log transformation is equal to 1.099665, which is much lower than the Box-Cox transformation. However, the histogram of the log-transformed data still appears to be skewed, and the plot of the log transformation still displays a clear trend line. Our best bet at this point to make the data stationary is to try differencing.

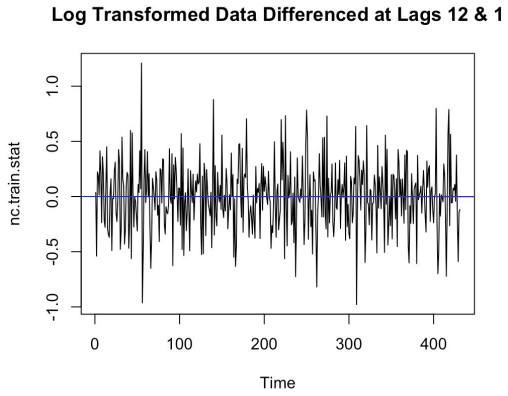
Differencing

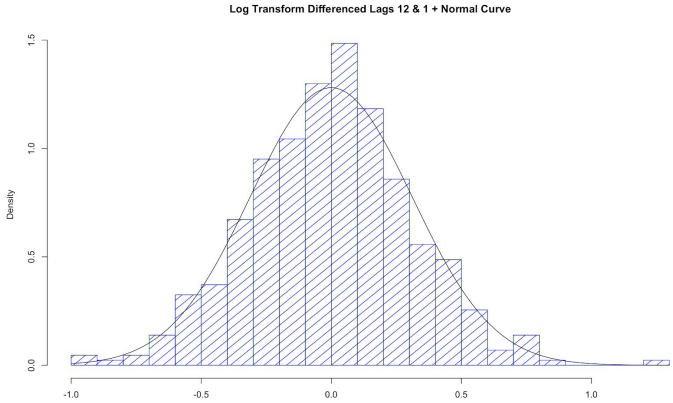
The first attempt at differencing the data will be made at lag 12, to account for the seasonality of the data.





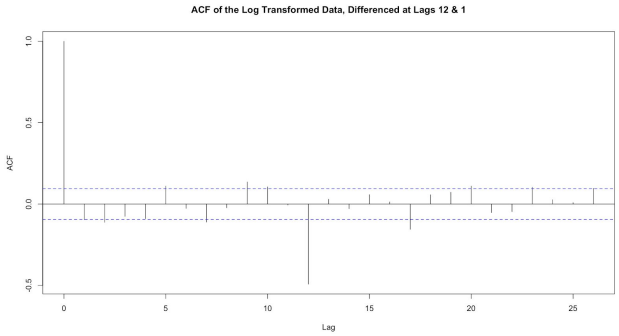
Differencing at lag 12 gets us very close to a stationary dataset. The variance of the log-transformed data differenced at lag 12 is equal to 0.1759289, which is the lowest variance obtained thus far. We will difference again at lag 1 to see if it further stabilizes the variance.

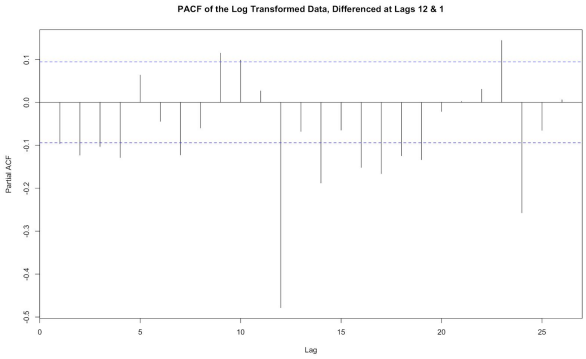




Differencing at lag 12 for seasonality and again at lag 1 for trend gave us variance equal to 0.09697977. This is the lowest variance we have obtained yet. The plot of the log transformed data differenced at lags 12 and 1 no longer displays seasonality or trend. The histogram of this data is near symmetric and closely fits that of a Gaussian distribution. The data now looks stationary, but now we must check the ACF and PACF of the log transformed and differenced data in order to confirm.

Analyze ACF and PACF





The plots display the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of the log transformed data differenced at lag 12 and at lag 1. Plot of the ACF displays a sharp spike at lag 12, but no other significant spikes. Analysis of the PACF is particularly interesting as it displays exponential decay in the seasonal lags of the PACF; in other words there are significant spikes at lags 12, 24, 36 ... etc.

Candidate Model Identification

The data will have to be fit to a SARIMA model since it has been differenced at lag 12 and lag 1. Regarding the potential model parameters, s=12 for the period, D=1 for differencing once at lag 12, and d=1 for differencing once at lag 1. We have to look for potential values of P, Q, p, and q. Based on the ACF/PACF plots, possible values for p = q = 1,2,3,4 and possible values for P = Q = 0,1, 2. These values were determined based on lags in which the spikes extended outside of the confidence intervals in the ACF/PACF plots. Since these values are all very close, AIC criterion is used to determine which values of p, q, P, and Q best suit the data and provide the lowest AIC value.

Top 5 Models Producing the Lowest AIC (R Code):

1. arima(nc.train.log, order=c(3,1,4), seasonal = list(order = c(1,1,1), period = 12), method="ML")
   AICc(arima(nc.train.log, order=c(3,1,4), seasonal = list(order = c(1,1,1), period = 12), method="ML"))
   [1] -64.83467

2. arima(nc.train.log, order=c(3,1,4), seasonal = list(order = c(2,1,2), period = 12), fixed = c(0, NA, NA, NA,NA,NA,NA,NA,0,NA,NA), method="ML")
   AICc(arima(nc.train.log, order=c(3,1,4), seasonal = list(order = c(2,1,2), period = 12), fixed = c(0, NA, NA, NA,NA,NA,NA,NA,0,NA,NA), method="ML"))
   [1] -69.73792

3. arima(nc.train.log, order=c(4,1,4), seasonal = list(order = c(1,1,1), period = 12), method="ML")
   AICc(arima(nc.train.log, order=c(4,1,4), seasonal = list(order = c(1,1,1), period = 12), method="ML"))
   [1] -64.48235

4. arima(nc.train.log, order=c(4,1,4), seasonal = list(order = c(2,1,2), period = 12), fixed = c(NA,NA,NA,NA,NA,0,NA,NA,NA,0,NA,NA), method="ML")
   AICc(arima(nc.train.log, order=c(4,1,4), seasonal = list(order = c(2,1,2), period = 12), fixed = c(NA,NA,NA,NA,NA,0,NA,NA,NA,0,NA,NA), method="ML"))
   [1] -68.44992

5. arima(nc.train.log, order=c(2,1,4), seasonal = list(order = c(0,1,1), period = 12), method="ML")

AICc(arima(nc.train.log, order=c(2,1,4), seasonal = list(order = c(0,1,1), period = 12), method="ML"))
[1] -55.26053

The top two models, Model A and Model B respectively, are based off of the SARIMA model parameters that provide the lowest AIC value.

Model A:
SARIMA(3,1,4)x(2,1,2)$_{12}$

$\nabla_1 \nabla_{12} \ln(U_t) = (1 + 0.53B^2 - 0.62B^3)(1 - B^{12})Z_t = (1 - 0.24B + 0.34B^2 - 0.94B^3 - 0.13B^4)(1 - 0.97B^{12} - 0.32B^{13} - 1.17B^{14})\, \varepsilon_t$,

where ε_t represents a white noise process.

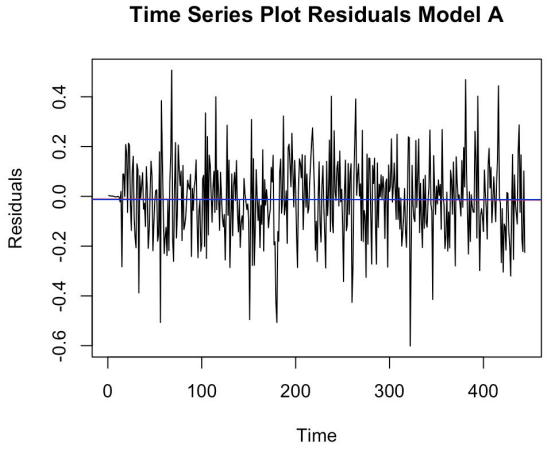
Model B:
SARIMA(4,1,4)x(2,1,2)$_{12}$

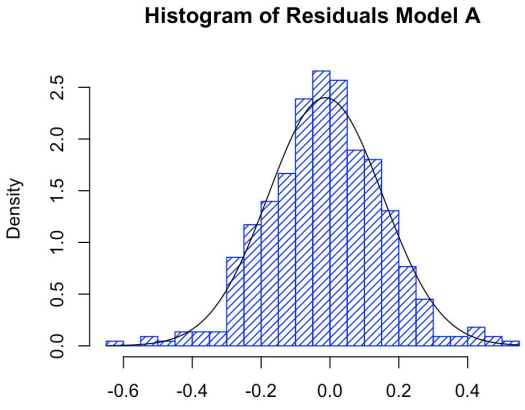
$\nabla_1 \nabla_{12} \ln(U_t) = (1 + 0.9B + 0.45B^2 - 0.19B^3 - 0.63B^4)(1-B^{12})Z_t = (1 + 0.60B - 0.63B^3 - 0.99B^4)(1 - 0.98B^{12} + 0.16B^{14} - 0.73B^{15})\, \varepsilon_t$,

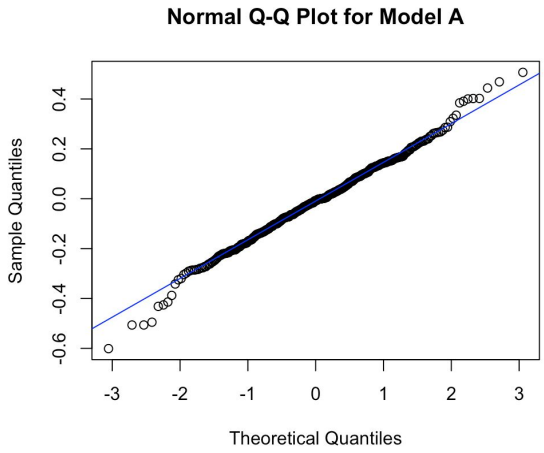
where ε_t represents a white noise process.

Diagnostic Checking for Model A

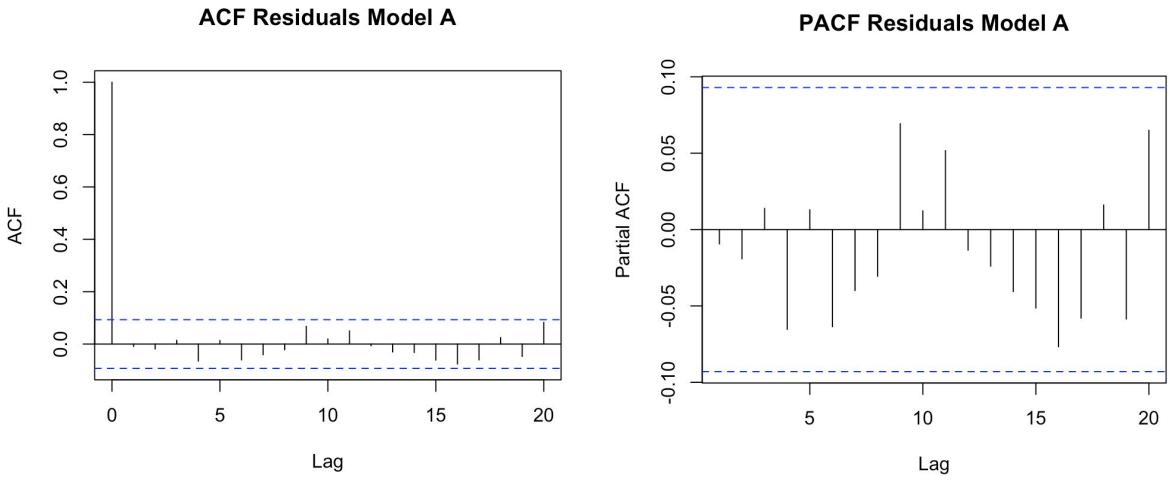
SARIMA(3,1,4)x(2,1,2)$_{12}$







Plotting the residuals of Model A displays virtually no trend. The histogram of residuals for Model A also looks good and is not skewed. The sample mean of residuals is almost zero and is equal to -0.01314788. Although there are some deviations in the Normal-QQ plot of the residuals of Model A, they are quite far and the Normal-QQ plot looks okay. For normal distribution, 95% of the values should be between +/- 2 and that is certainly the case in the Normal-QQ plot of residuals for Model A.



All ACF lags are within the confidence interval with the exception of lag 0. All PACF lags of residuals for Model A are within the confidence intervals and can be counted as zeros. The plots resemble the ACF and PACF of white noise.

<u>Shapiro-Wilk Normality Test</u>

The Shapiro-Wilk test is used on the residuals of Model A to determine whether or not the residuals come from a normal distribution. A p-value greater than α=0.05 means that we fail to reject the null hypothesis that the residuals follow a normal distribution. The results were as follows:

Shapiro-Wilk normality test

data: res
W = 0.99355, p-value = 0.05526

The p-value is greater than α=0.05, so we fail to reject the null hypothesis and conclude that the residuals of Model A follow a normal distribution.

Box-Pierce Test

The Box-Pierce test is used on the residuals of Model A to test the null hypothesis that the residuals of Model A are independent of one another. The results were as follows:

Box-Pierce test

data: res
X-squared = 20.121, df = 12, p-value = 0.06484

The p-value is greater than α=0.05, so we fail to reject the null hypothesis and conclude that the residuals of model A are independent.

Box-Ljung Test

The Box-Ljung test is used on the residuals of Model A in order to test for the absence of serial autocorrelation. A p-value greater than α=0.05 means that we fail to reject the null hypothesis that the residuals of Model A do not show any lack of fit. The results were as follows:

Box-Ljung test

data: res
X-squared = 20.827, df = 12, p-value = 0.05297

The p-value is greater than α=0.05, so we fail to reject the null hypothesis and conclude that the residuals of Model A do not show lack of fit. Furthermore, it can be concluded that the residuals of Model A have small autocorrelation.

McLeod-Li Test

The McLeod-Li test used on the squared residuals of Model A tests for the presence of autoregressive conditional heteroscedasticity. A p-value greater than α=0.05 means that we fail to reject the null hypothesis that there is no autocorrelation in the squared residuals up to lag 21. The results were as follows:

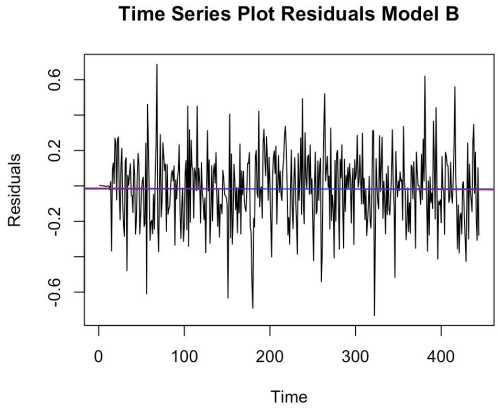
Box-Ljung test

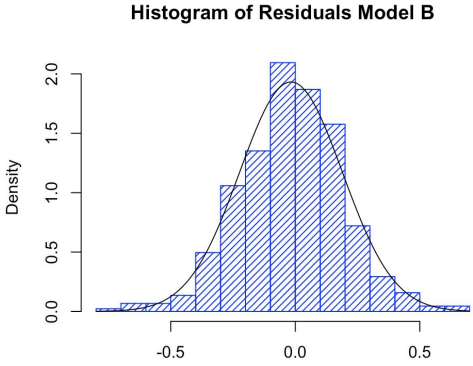
data: res^2
X-squared = 31.517, df = 21, p-value = 0.06547

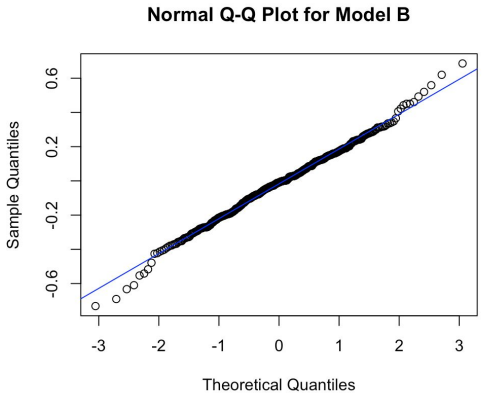
The p-value is greater than α=0.05, so we fail to reject the null hypothesis and conclude that there is no autocorrelation in the squared residuals of Model A.

Diagnostic Checking for Model B

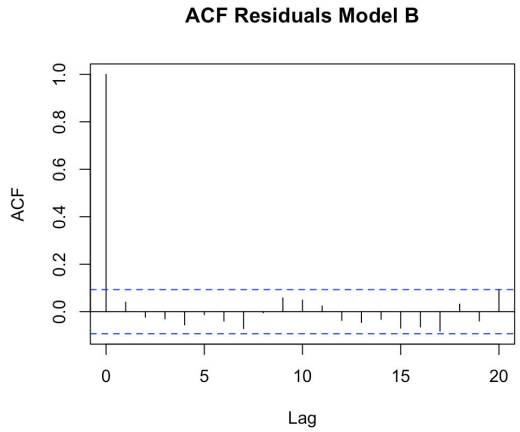
SARIMA(4,1,4)x(2,1,2)$_{12}$

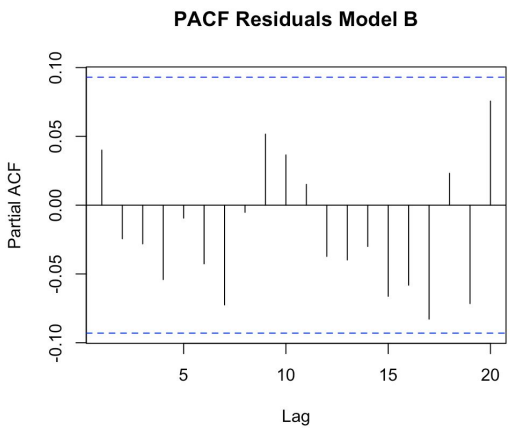






Plotting the residuals of Model B displays little to no trend. The histogram of residuals for Model B looks okay. The sample mean of residuals is almost zero and is equal to -0.01719532. The histogram of residuals of Model B looks good and is not skewed. The Normal-QQ plot looks good and 95% of points appear to be between +/- 2.





The ACF and PACF of residuals for Model B closely resemble that of Model A. All ACF and PACF lags are contained within the confidence interval with the exception of lag 0 in the ACF. The ACF and PACF of the residuals of Model B resemble that of white noise. Up to this point, Model B looks like an equally appropriate model as Model A to fit the data. Further use of tests will be the determining factor in deciding which model to use.

Shapiro-Wilk Normality Test

Shapiro-Wilk normality test

data: res
W = 0.99442, p-value = 0.1054

The p-value is greater than α=0.05, and we fail to reject the null hypothesis. It can be concluded that the residuals of Model B follow a normal distribution.

Box-Pierce Test

Box-Pierce test

data: res
X-squared = 22.6, df = 11, p-value = 0.02012

The p-value for the Box-Pierce test is not greater than α=0.05 level of significance. The residuals for Model B fail the Box-Pierce test and we can conclude that the residuals are not independent.

Box-Ljung Test

Box-Ljung test

data: res
X-squared = 23.39, df = 11, p-value = 0.01557

The p-value is not greater than α=0.05 and we reject the null hypothesis. From the Box-Ljung test we can conclude that the residuals for Model B display a lack of fit, and that there is autocorrelation in the residuals.

McLeod-Li Test

Box-Ljung test

data: res^2
X-squared = 30.797, df = 21, p-value = 0.0771

The p-value is greater than α=0.05 and we fail to reject the null hypothesis. It can be concluded that the squared residuals of Model B have no autocorrelation.

Choosing the Final Model

While the time series plots, histograms, Normal-QQ plots, and ACF/PACF plots of Model A and Model B are all interestingly similar, further diagnostic testing reveals that Model A is more suitable to fit the data than Model B. Model A passed all of the diagnostic tests which included: Shapiro-Wilk Normality test, Box-Pierce test, Box-Ljung test, and the McLeod-Li test. Model B, however, failed the Box-Pierce test as well as the Box-Ljung test. From these failed tests we conclude that the autocorrelations of the residuals for Model B are large and that Model B shows a significant lack of fit. It is decided to use Model A as the final model to fit the data and forecast future data points. The model diagnosis conducted on Model A concluded that the model satisfied the constant variance assumption, the independence assumption, and the normality assumption.

Final Model for the Logarithm Transform:

From the analysis conducted, the fitted and final model, $\ln(U_t)$, follows a SARIMA$(3,1,4)\times(2,1,2)_{12}$ model. In more detailed form it is:

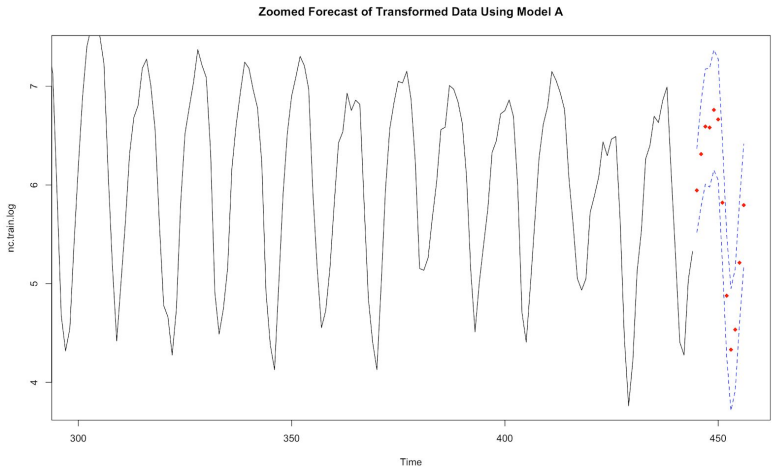
$\nabla_1 \nabla_{12} \ln(U_t) = (1 + 0.53B^2 - 0.62B^3)(1 - B^{12})Z_t = (1 - 0.24B + 0.34B^2 - 0.94B^3 - 0.13B^4)(1 - 0.97B^{12} - 0.32B^{13} - 1.17B^{14})\,\varepsilon_t$,

where ε_t represents a white noise process.

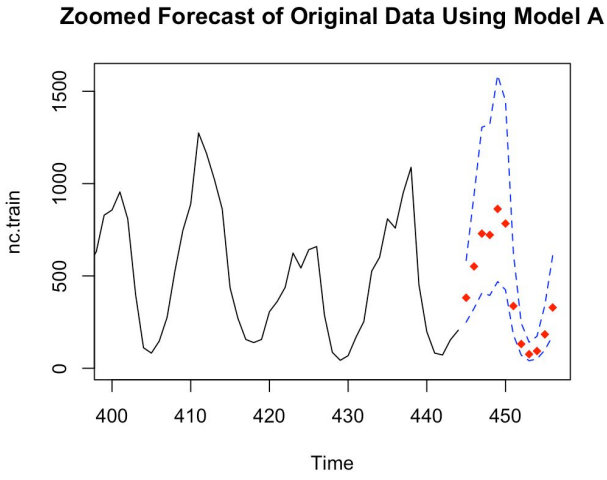
The model coefficients were determined using the arima function in R to fit the time series model.

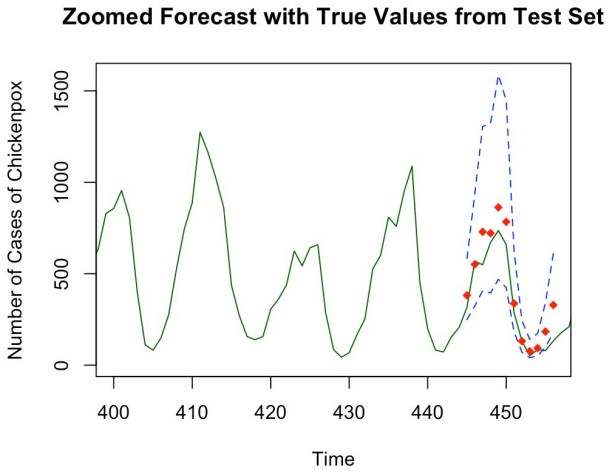
Forecasting Original Data

Using the SARIMA(3,1,4)x(2,1,2)$_{12}$ model (Model A), 10 observations are forecasted, first on the transformed data and then again on the original training set and original full dataset.



The forecast of transformed data using Model A looks good. All 12 of the future values predicted by Model A fall within the confidence interval.





The data was log transformed, so we get the data back to its original form by performing e to the power. The forecast of the original data using Model A also looks good. First, the 12 predicted points are applied to the plot of the training set of data (nc.train) containing 444 observations. As displayed in the plot, all predicted future points fall within the confidence interval. Next, the observations predicted by Model A are plotted with the original data, this time including both the training and test sets for a total of all 498 observations. As can be seen, the actual observations that were contained in the test set all fall within the confidence interval of our model forecast. These true observations from the test set also closely align with the predicted values plotted by our model forecast.

Conclusions

Based on the time series analysis conducted on the data "Monthly reported number of chickenpox, New York city, 1931-1972", a SARIMA(3,1,4)x(2,1,2)$_{12}$ model was able to be applied after transforming and differencing to make the original data stationary. This model can be represented by the model equation:

$\nabla_1 \nabla_{12} \ln(U_t) = (1 + 0.53B^2 - 0.62B^3)(1 - B^{12})Z_t = (1 - 0.24B + 0.34B^2 - 0.94B^3 - 0.13B^4)(1 - 0.97B^{12} - 0.32B^{13} - 1.17B^{14})\, \varepsilon_t$

By applying this model to the stationary data, we are able to predict future observations within a 95% confidence interval. The goals from the beginning of the project were achieved, as we were able to detect both trend and seasonality in the original data, and hence accommodated for these factors. Predicting twelve future observations, or one year into the future, shows that the number of cases of chickenpox in NYC will likely rise again and then dip. The first six predicted observations displayed an upward trend in the number of cases, and the last six predicted observations displayed a downward trend in the number of cases. The conclusion may be reached that the number of cases of chickenpox increases in the hot summer months and then recesses again in the cooler winter months. The true observations from the test set support this theory and align closely with the forecasted observations made for twelve months. The ability to forecast a year ahead with the number of cases of chickenpox in NYC holds great value for a number of reasons. First off, greater public awareness can be made for peak seasons of the disease and for predicted spikes in the disease via forecasting. Much like with what we are experiencing in present times with Covid-19, the ability to forecast future number of cases of the disease is one of the most powerful tools to use against diseases in order to “flatten the curve”.

References:

Forecasting: Principles and Practice. (n.d.). Retrieved from https://otexts.com/fpp2/seasonal-arima.html

Graves, A. (2018, January 7). Time Series Forecasting with a SARIMA Model. Retrieved from https://towardsdatascience.com/time-series-forecasting-with-a-sarima-model-db051b7ae459

Professor Feldman Lecture Notes and Slides

Hipel, K. W., & McLeod, A. I. (1994). Time series modelling of water resources and environmental sciences. Amsterdam: Elsvier.

Stephanie. (2018, September 7). Ljung Box Test: Definition. Retrieved from https://www.statisticshowto.com/ljung-box-test/

Appendix:

```
library(tsdl)
tsdl

tsdl <- subset(tsdl,"Health")
tsdl

attributes(tsdl[[6]])

View(tsdl[[6]])

num_cases <-ts(as.vector(tsdl[[6]]))

plot.ts(num_cases, ylab = "Number of Chickenpox Cases")

nt = length(num_cases)

## Add trend line to data plot:
fit <- lm(num_cases ~ as.numeric(1:nt)); abline(fit, col="red", ylab = "Number of Chickenpox
Cases")

mean(num_cases)

## Add mean (constant) to the data plot:
abline(h=mean(num_cases), col="blue")

## Plot chickenpox data with years on x=axis:
tsdata <- ts(num_cases, start = c(1931,1), end = c(1972,2), frequency = 12)

ts.plot(tsdata, main = "Raw Data Chickenpox Cases NYC 1931-1972", ylab="Number of
Chickenpox Cases")

fit <- lm(tsdata ~ as.numeric(1:length(tsdata))); abline(fit, col="red")

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

```
## Partition dataset to two parts for model training and model validation; work with training set:
nc.train = ts(num_cases[c(1:444)]) # Training Set
nc.test = ts(num_cases[c(445:498)]) # Test Set

## Chickenpox Training (truncated) Set: 444 observations
plot.ts(nc.train,  main = "Training Set:Truncated Chickenpox Cases NYC", ylab="Number of Chickenpox Cases")

fit <- lm(nc.train ~ as.numeric(1:length(nc.train))); abline(fit, col="red")

abline(h=mean(num_cases), col="blue")

## Immediate Observations: Non-constant of variance and mean. No obvious increasing or decreasing trend. Highly Non-Stationary.

## Confirm non-stationarity of original data (truncated)(nc.train)
## Plot Histogram Training Set:
hist(nc.train, col="dark blue", xlab="", main="Histogram; Truncated Chickenpox Data")
## Observations: Histogram is badly skewed.

## Plot acf of Training Set
acf(nc.train,lag.max=NULL, main="ACF of the Truncated ChickenPox Data")
## Observations: Seasonality is obvious, at lag 12 it starts to repeat the previous cycle. ACFs remain large.

pacf(nc.train,lag.max=NULL, main="PACF of the Truncated ChickenPox Data")

## Decomposition of the Log-Transformed truncated data:
library(ggplot2)
library(ggfortify)

y <- ts(as.ts(nc.train.stat), frequency = 12)
decomp <- decompose(y, type = "additive")
plot(decomp)
```

Decomposition Observations: the dataset has a strong seasonality pattern. the lower bound of the data is constant, whereas the peak is changing. there is an overall decreasing trend in the pattern. The data itself is not stationary as the variance is changing.

Box-Cox Transformation:
```
bcTransform <- boxCox(nc.train~ as.numeric(1:length(nc.train)), main="Box-Cox Transformation")
```

Command to give the value of lambda in the Box-Cox plot.
```
bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
```
lambda = 0.3434343

Perform transformations, plot transformed data, histograms:
```
lambda=bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
nc.train.bc = ts((1/lambda)*(nc.train^lambda-1))
nc.train.log <- ts(log(nc.train))
```

Plot of truncated data after Box-Cox Transformation
```
plot.ts(nc.train.bc, main="Box-Cox Transformation: Truncated Chickenpox Data", ylab="Number of Chickenpox Cases")
fit <- lm(nc.train.bc ~ as.numeric(1:length(nc.train.bc))); abline(fit, col="red")
mean(nc.train.bc)
abline(h=mean(nc.train.bc), col="blue")
```

Histogram of truncated data after Box-Cox Transformation
```
hist(nc.train.bc, col="blue", xlab="", main="Histogram; Box-Cox Transformation")
var(nc.train.bc)
```

Variance = 72.63398

Plot of truncated data after Log Transformation
```
plot.ts(nc.train.log, main="Log Transformed Truncated Chickenpox Data", ylab="Number of Chickenpox Cases")

fit <- lm(nc.train.log ~ as.numeric(1:length(nc.train.log))); abline(fit, col="red")
mean(nc.train.log)
abline(h=mean(nc.train.log), col="blue")

var(nc.train.log)
```

Observations: Variance = 1.099665

Histogram of the truncated data after Log Transform:
hist(nc.train.log, col="blue", xlab="", main="Histogram; Log Transformation")

Differencing of the Log-Transformed truncated data:
Differencing at Lag 12:
nc.train.log12 <- ts(diff(nc.train.log, lag=12))
plot.ts(nc.train.log12, main="Log Transformed Data Differenced at Lag 12")

fit <- lm(nc.train.log12 ~ as.numeric(1:length(nc.train.log12))); abline(fit, col="red")
mean(nc.train.log12)
abline(h=mean(nc.train.log12), col="blue")

var(nc.train.log12)
Variance = 0.1759289 log transformed and differenced at lag 12

Differencing apt.log at Lag 12 and Lag 1:
nc.train.stat <- ts(diff(nc.train.log12, lag=1))
plot.ts(nc.train.stat, main="Log Transformed Data Differenced at Lags 12 & 1")

fit <- lm(nc.train.stat ~ as.numeric(1:length(nc.train.stat))); abline(fit, col="red")

mean(nc.train.stat)
abline(h=mean(nc.train.stat), col="blue")

var(nc.train.stat)
Observations: Variance = 0.09697977 LOWEST VARIANCE YET! No more seasonality. No more Trend. The data looks stationary, but now we must check the acf's to confirm.

Plot of ACF of log-transformed truncated data:
acf(nc.train.log, lag.max=NULL, main="ACF of the log(U_t)")
Observations: Non-stationarity. Seasonality is apparent.

Plot of ACF of log-transformed truncated data Differenced at lag 12:
acf(nc.train.log12, lag.max=NULL, main="ACF of the log(U_t), differenced at lag 12")
Observations: Seasonality, non-stationarity.

```
## Plot of ACF of log-transformed truncated data Differenced at lag 12 and at Lag 1:
acf(nc.train.stat, lag.max=NULL, main="ACF of the Log Transformed Data, Differenced at Lags 12 & 1")
## Observations: ACF decay corresponds to a stationary process

## Decision: Work with the log-transformed truncated data differenced at lag 12 and lag 1.

par(mfrow=c(1,1))

## Compare histograms of log-transformed truncated data:
## Histogram of log-transformed truncated data:
hist(nc.train.log, col="blue", xlab="", main="Histogram; Log Transformation")

##Histogram of log transformed truncated data differenced at lag 12
hist(nc.train.log12, col="blue", xlab="", main="Histogram; Log Transform, Differenced Lag 12")

## Histogram of log-transformed truncated data differenced at lag 12 and lag 1:
hist(nc.train.stat, col="blue", xlab="", main="Histogram; Log Transform Differenced Lags 12 & 1")
## Observations: This histogram looks symmetric and almost Gaussian.

## Histogram of transformed and differenced data with normal curve:
hist(nc.train.stat, density=20,breaks=20, col="blue", xlab="", prob=TRUE, main="Log Transform Differenced Lags 12 & 1 + Normal Curve")
m<-mean(nc.train.stat)
std<- sqrt(var(nc.train.stat))
curve( dnorm(x,m,std), add=TRUE )

## Plot PACF:
pacf(nc.train.stat, lag.max=NULL, main="PACF of the Log Transformed Data, Differenced at Lags 12 & 1")

auto.arima(nc.train.log, D=1, d=1)


test(nc.train.stat)

arima(nc.train.log, order=c(0,1,2), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -15.53
```

```
arima(nc.train.log, order=c(0,1,3), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -22.26

arima(nc.train.log, order=c(0,1,4), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -55.02

arima(nc.train.log, order=c(0,1,4), seasonal = list(order = c(0,1,2), period = 12), method="ML")
## aic = -55.48

arima(nc.train.log, order=c(1,1,0), seasonal = list(order = c(0,1,0), period = 12), method="ML")
## aic = 216.47

arima(nc.train.log, order=c(2,1,0), seasonal = list(order = c(0,1,0), period = 12), method="ML")
## aic = 211.87

arima(nc.train.log, order=c(3,1,0), seasonal = list(order = c(0,1,0), period = 12), method="ML")
## aic = 209.21

arima(nc.train.log, order=c(1,1,0), seasonal = list(order = c(1,1,0), period = 12), method="ML")
## aic = 90.5

arima(nc.train.log, order=c(2,1,0), seasonal = list(order = c(1,1,0), period = 12), method="ML")
## aic = 80.84

arima(nc.train.log, order=c(3,1,0), seasonal = list(order = c(1,1,0), period = 12), method="ML")
## aic = 81.95

arima(nc.train.log, order=c(2,1,0), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -7.85

arima(nc.train.log, order=c(3,1,0), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -7.75

arima(nc.train.log, order=c(0,1,2), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -15.53

arima(nc.train.log, order=c(1,1,2), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -55.9
```

```
arima(nc.train.log, order=c(2,1,2), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -57.19

arima(nc.train.log, order=c(3,1,2), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -55.27

arima(nc.train.log, order=c(1,1,3), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -53.92

arima(nc.train.log, order=c(2,1,3), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -55.25

arima(nc.train.log, order=c(3,1,3), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -55.16

arima(nc.train.log, order=c(0,1,4), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -55.02

arima(nc.train.log, order=c(1,1,4), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -54.68

arima(nc.train.log, order=c(2,1,4), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## aic = -55.6


arima(nc.train.log, order=c(3,1,4), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## AIC = -62.85

arima(nc.train.log, order=c(4,1,4), seasonal = list(order = c(0,1,1), period = 12), method="ML")
## AIC = -57.32

arima(nc.train.log, order=c(3,1,4), seasonal = list(order = c(1,1,1), period = 12), method="ML")
## AIC = -65.36

arima(nc.train.log, order=c(4,1,4), seasonal = list(order = c(1,1,1), period = 12), method="ML")
## AIC = -65.11
```

```
arima(nc.train.log, order=c(4,1,4), seasonal = list(order = c(2,1,2), period = 12), method="ML")
## AIC = -67.09

#### MODEL B 2nd Lowest AIC
arima(nc.train.log, order=c(4,1,4), seasonal = list(order = c(2,1,2), period = 12), fixed =
c(NA,NA,NA,NA,NA,0,NA,NA,NA,0,NA,NA), method="ML")
## aic = -69.08

arima(nc.train.log, order=c(3,1,4), seasonal = list(order = c(2,1,2), period = 12), method="ML")
## aic = -67.7

#### MODEL A 1st Lowest AIC
arima(nc.train.log, order=c(3,1,4), seasonal = list(order = c(2,1,2), period = 12), fixed = c(0, NA,
NA, NA,NA,NA,NA,NA,0,NA,NA), method="ML")
## aic = -69.39

arima(nc.train.log, order=c(4,1,4), seasonal = list(order = c(2,1,1), period = 12), method="ML")
## AIC = -57.32

fit <- arima(nc.train.log, order=c(4,1,4), seasonal = list(order = c(2,1,2), period = 12), fixed =
c(NA,NA,NA,NA,NA,0,NA,NA,NA,0,NA,NA), method="ML")

res <- residuals(fit)

hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE, main = "Histogram of Residuals
Model B")

m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )

m

plot.ts(res,  main = "Time Series Plot Residuals Model B", ylab="Residuals")

fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")

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

```
qqnorm(res,main= "Normal Q-Q Plot for Model B")

qqline(res,col="blue")

acf(res, lag.max=20, main = "ACF Residuals Model B")

pacf(res, lag.max=20, main = "PACF Residuals Model B")

shapiro.test(res)

Box.test(res, lag = 21, type = c("Box-Pierce"), fitdf = 10)

Box.test(res, lag = 21, type = c("Ljung-Box"), fitdf = 10)

Box.test(res^2, lag = 21, type = c("Ljung-Box"), fitdf = 0)

acf(res^2, lag.max=NULL)

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

install.packages("forecast")
library(forecast)

fit.A <- arima(nc.train.log, order=c(3,1,4), seasonal = list(order = c(2,1,2), period = 12), fixed =
c(0, NA, NA, NA,NA,NA,NA,NA,0,NA,NA), method="ML")

forecast(fit.A)

pred.tr <- predict(fit.A, n.ahead =12)

U.tr= pred.tr$pred + 2*pred.tr$se

L.tr= pred.tr$pred - 2*pred.tr$se

ts.plot(nc.train.log, xlim=c(300,length(nc.train.log)+12), ylim = c(min(nc.train.log),max(U.tr)),
main="Zoomed Forecast of Transformed Data Using Model A")

lines(U.tr, col="blue", lty="dashed")
```

```
lines(L.tr, col="blue", lty="dashed")

points((length(nc.train.log)+1):(length(nc.train.log)+12), pred.tr$pred, col="red", pch=18)

pred.orig <- exp(pred.tr$pred)

U= exp(U.tr)

L= exp(L.tr)

ts.plot(nc.train, xlim=c(1,length(nc.train)+12), ylim = c(min(nc.train),max(U)))

lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")

points((length(nc.train)+1):(length(nc.train)+12), pred.orig, col="red",pch=18)

ts.plot(nc.train, xlim = c(400,length(nc.train)+12), ylim = c(1,max(U)), main="Zoomed Forecast
of Original Data Using Model A")

lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")

points((length(nc.train)+1):(length(nc.train)+12), pred.orig, col="red", pch=18)

ts.plot(num_cases, xlim = c(400,length(nc.train)+12), ylim = c(1,max(U)), col="dark green", ylab
= "Number of Cases of Chickenpox", main="Zoomed Forecast with True Values from Test Set")

lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")

points((length(nc.train)+1):(length(nc.train)+12), pred.orig, col="red", pch=18)

ts.plot(num_cases, xlim = c(300,length(nc.train)+54), ylim = c(300,max(U)),col="red")

ts.plot(num_cases, xlim = c(200,length(nc.train)+12), ylim = c(250,max(U)), col="red")

lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
```

```
points((length(nc.train)+1):(length(nc.train)+12), pred.orig, col="green")
points((length(nc.train)+1):(length(nc.train)+12), pred.orig, col="black")
```