Time Series Forecasting on SFO Air Passenger Statistics

Jinxiang Ma

2022-11-28

Contents

Executive Summary

In this project, we perform time series analysis on the passenger data of San Francisco International Airport(SFO) from 2005 to 2019. The dataset used in this project can be obtained from DataSF.com and air passenger data was updated on a monthly basis. Each row in this dataset represents information of each passenger(Operating Airline, Activities Type, Passenger Count at each flight)

With this dataset, we analyze the historical air passenger data and build a SARIMA model to forecast 12 month period in future.

Data Cleaning

First, we read data from the CSV file and concatenate each passengers in each flights into monthly data.

```
at_data = read.csv("Air_Traffic_Passenger_Statistics.csv", sep = ",",header = TRUE)
at_data$Passenger.Count <-  as.numeric(gsub(",","",at_data$Passenger.Count))
```

```
library(dplyr)
```

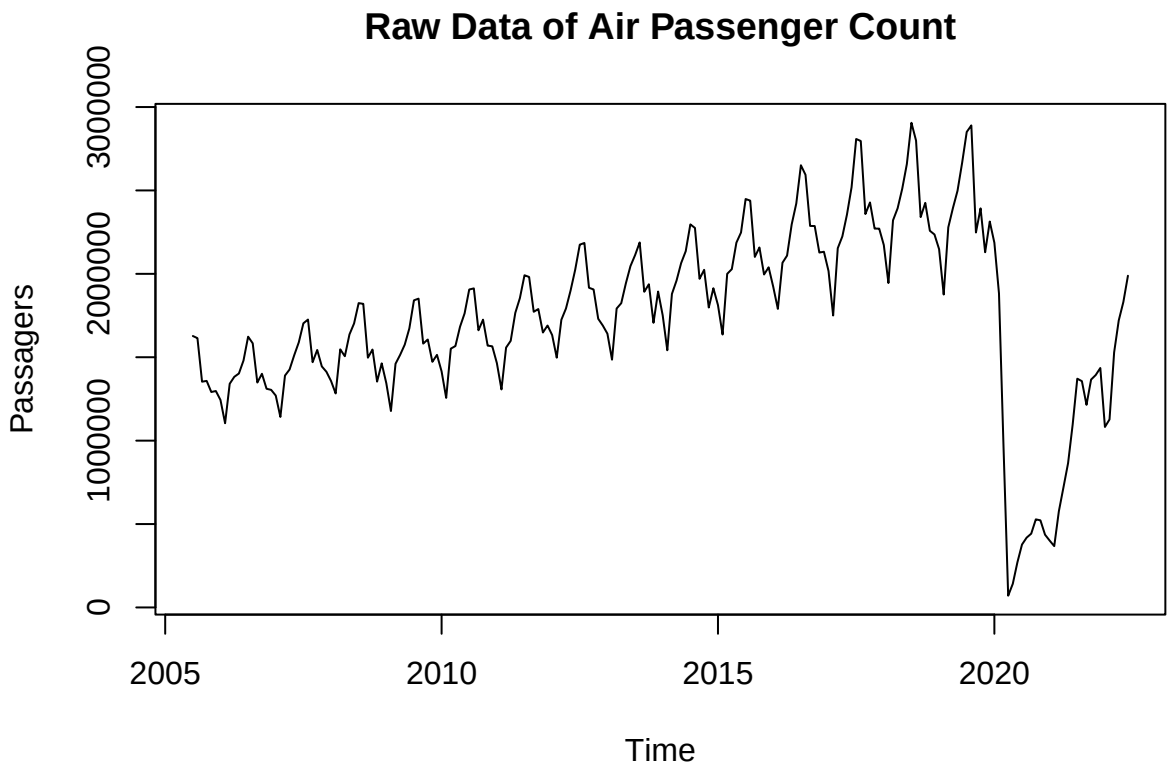
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
at_df = at_data %>%
  filter(at_data$Activity.Type.Code == "Deplaned") %>%
  select(Activity.Period, Passenger.Count) %>%
  group_by(Activity.Period)%>%
  summarize(Monthly_Passenger = sum(Passenger.Count))
```

```
at_ts = ts(at_df[,2], start = c(2005, 7), frequency = 12)
plot.ts(at_ts, main = 'Raw Data of Air Passenger Count', ylab = 'Passagers')
```

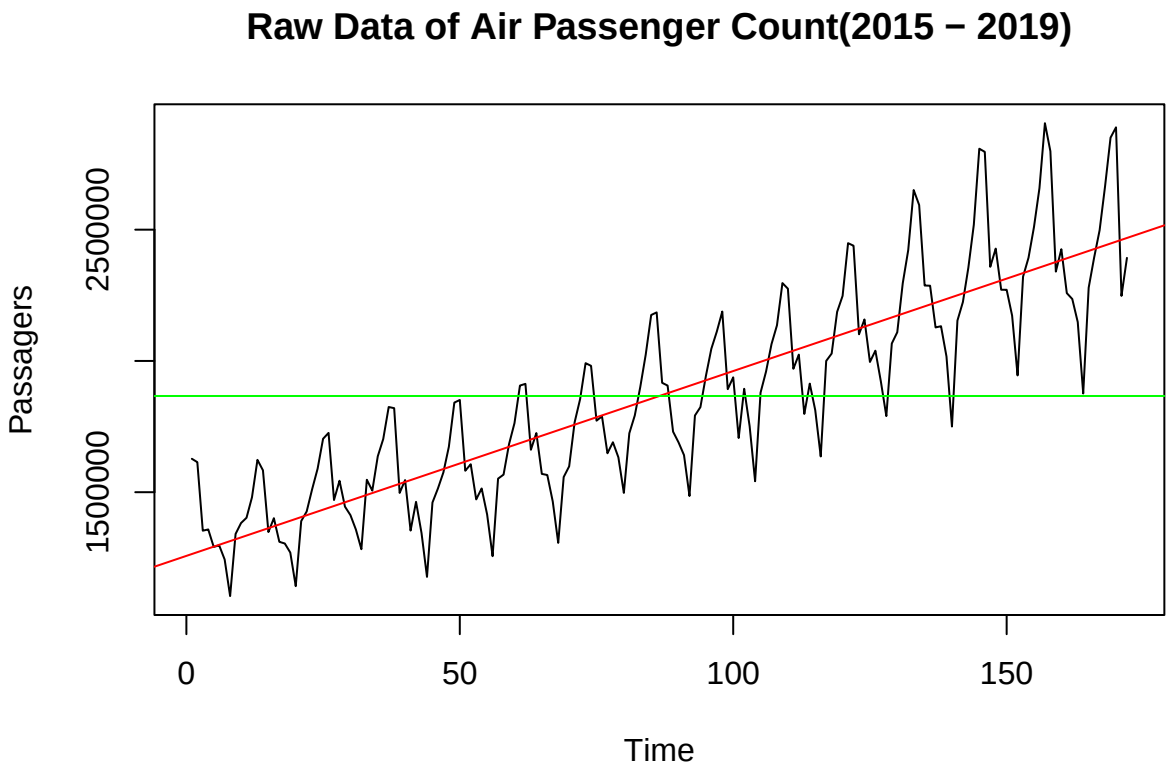


Base on the raw data of air passenger of SFO Airport, we see that the amount of passenger was increasing

every year before 2019 and exhibit a seasonal trend. We observe a sharp decline of passenger amount in 2020 due to the impact of COVID-19 pandemic. Hence, We decided to remove the air passenger data after 2020.

Preliminary Data Analysis

```
at_df = at_df[c(1:172), ]
plot.ts(at_df$Monthly_Passenger, main = 'Raw Data of Air Passenger Count(2015 - 2019)', ylab = 'Passage
len = length(at_df$Monthly_Passenger)
fit <- lm(at_df$Monthly_Passenger~as.numeric(1:len))
abline(fit, col="red")
abline(h=mean(at_df$Monthly_Passenger), col="green")
```



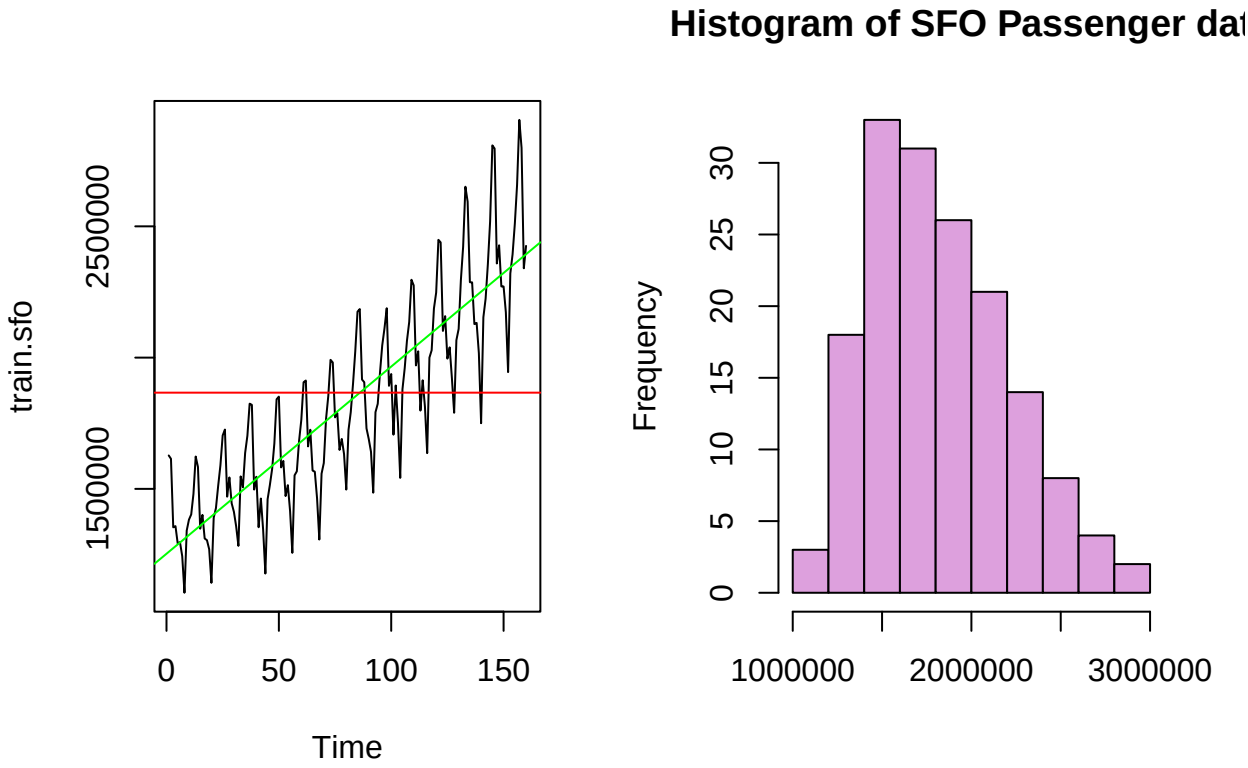
We split the SFO Passenger data into training set and test set.

```
train.sfo <- at_df$Monthly_Passenger[c(1:160)]
test.sfo <- at_df$Monthly_Passenger[c(160:172)]
```

Next, we need to check the distribution(trend and seasonality) of SFO air passenger data and see if any transformation is needed. We created a time series plot and histogram for training data. There is an obvious upward seasonal trend in the time series plot and from the histogram we see that our data is slightly right skewed, with mean value around 1,500,000.

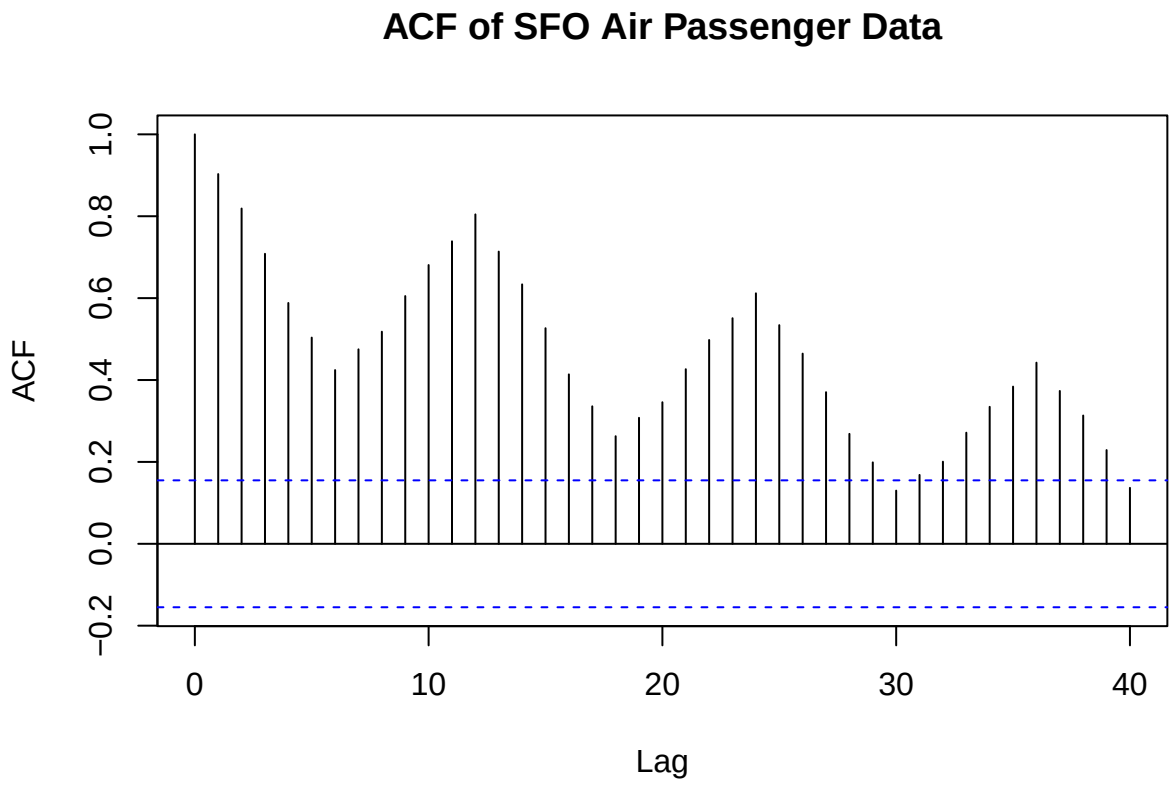
```
par(mfrow=c(1,2))
plot.ts(train.sfo)
fit <- lm(train.sfo~as.numeric(1:length(train.sfo)))
abline(fit, col = 'green')
abline(h = mean(at_df$Monthly_Passenger), col = 'red')
```

```
hist(train.sfo, col = "plum", xlab = "", main = "Histogram of SFO Passenger data")
```



If we plot the ACF of our training data, we can see periodic peaks at every 12 lags. We corresponds to our previous observation that our data has a strong seasonal trend (12 months).

```
acf(train.sfo, lag.max = 40, main = "ACF of SFO Air Passenger Data")
```



Data Tranformation

Calculating λ

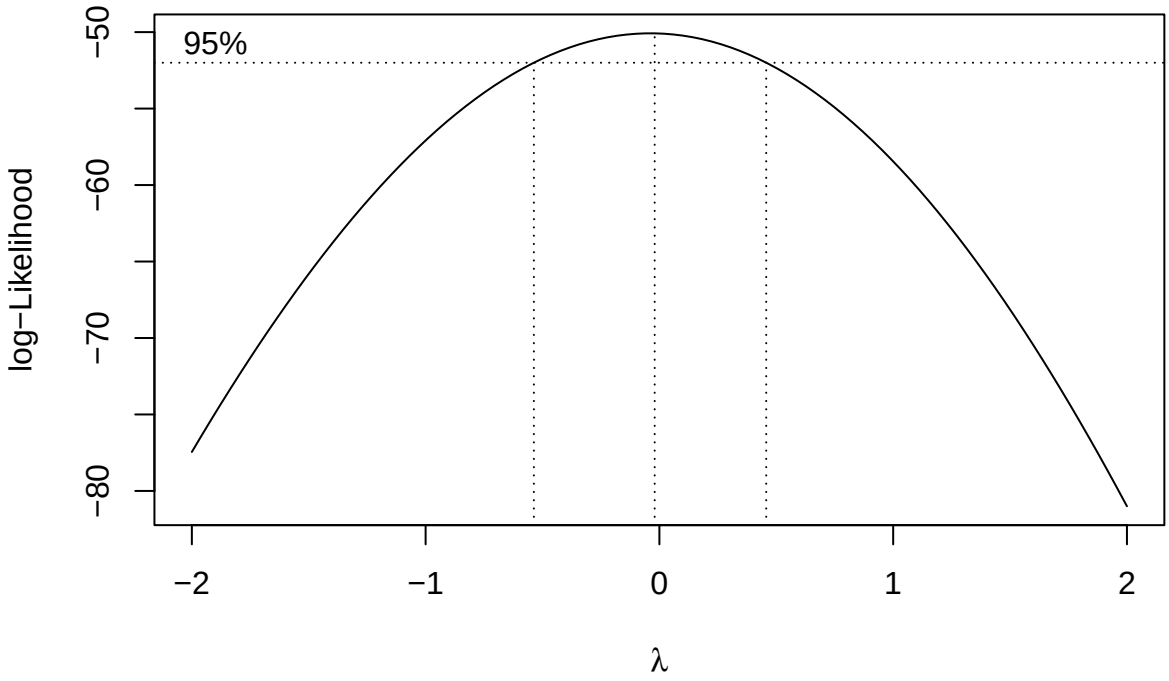
Since the air passenger data is skewed to the right with non constant variance, we decided to perform box-cox transformation and log transformation.

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
BoxCox_SFO = boxcox(train.sfo~as.numeric(1: length(train.sfo)))
```

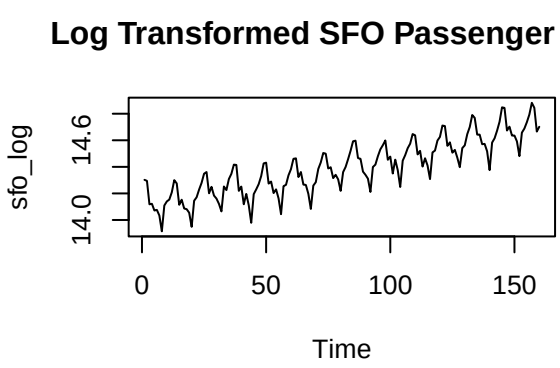


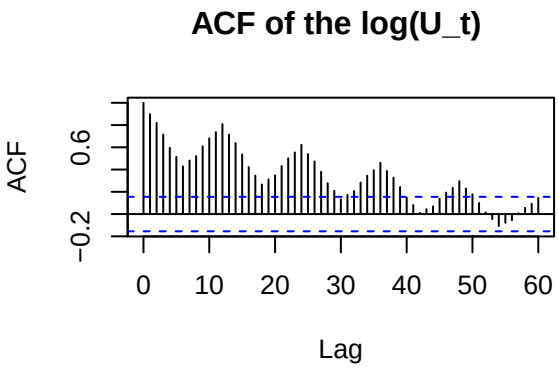

```
lambda = BoxCox_SFO$x[which(BoxCox_SFO$y == max(BoxCox_SFO$y))]
lambda
```

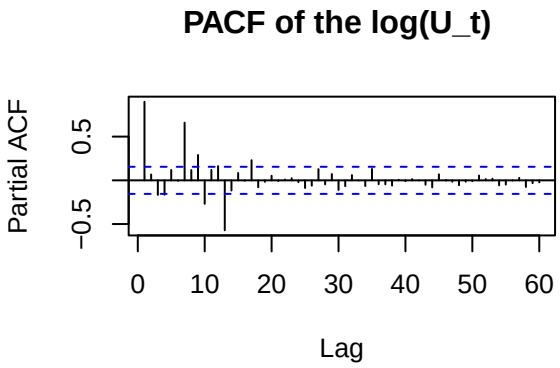
```
## [1] -0.02020202
```

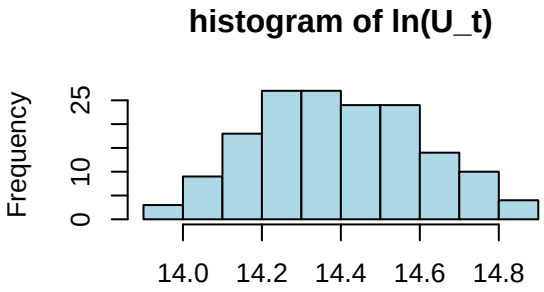
Now we get $\lambda = -0.02020202$. Since the confidence interval of lambda covers zero, I decided to use log transformation. Here, we used U_t to represent the first 160 observation of the original data, which is our training set.

```
par(mfrow = c(2,2))
sfo_BC = (1/lambda)*(train.sfo^lambda - 1)
sfo_log = log(train.sfo)
plot.ts(sfo_log, main = "Log Transformed SFO Passenger Data")
acf(sfo_log, lag.max = 60, main = "ACF of the log(U_t) ")
pacf(sfo_log, lag.max = 60, main = "PACF of the log(U_t) ")
hist(sfo_log, col="light blue", xlab="", main="histogram of ln(U_t)")
```

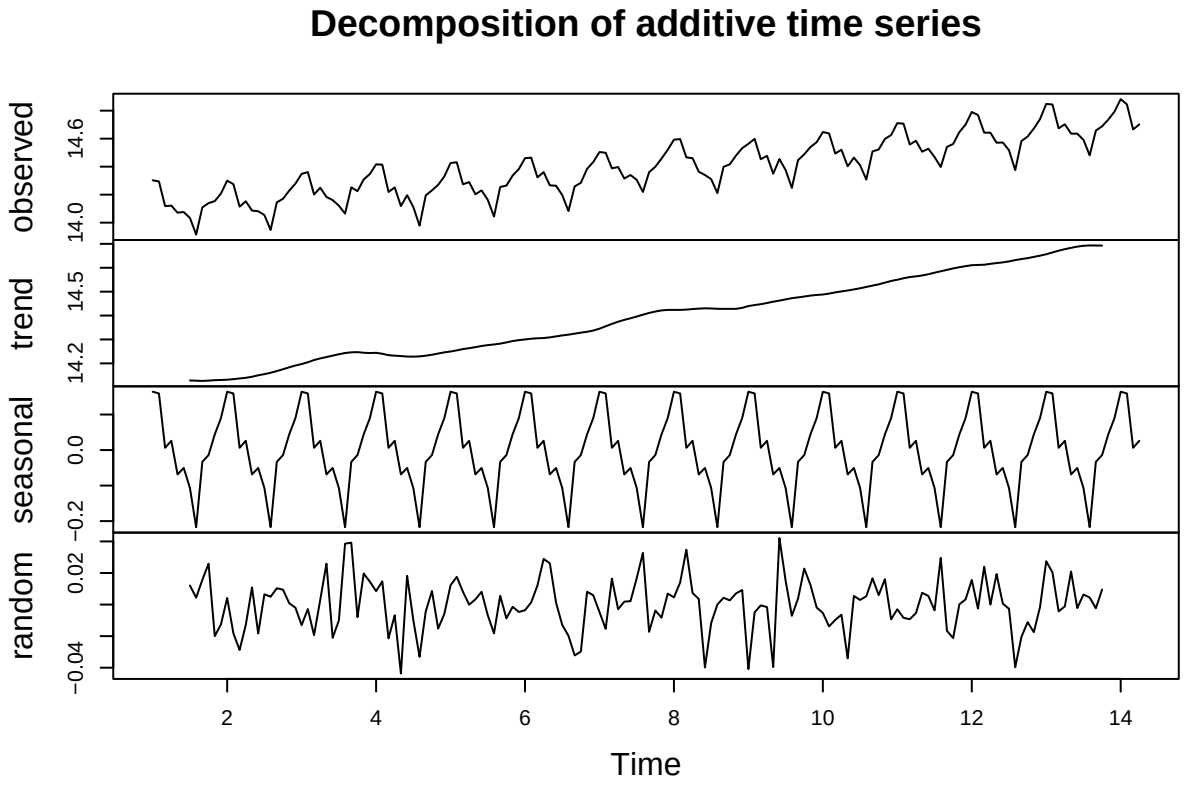









```
y <- ts(as.ts(sfo_log), frequency = 12)
plot(decompose(y))
```



Differentiating at Lag 12, $\nabla_{12} \ln(U_t)$

Now we differentiate sfo.log at log 12. We can still see a trend in the ACF and the time series is not stationary. The histogram shows that our data is left skewed.

```
par(mfrow = c(2,2))
var(sfo_log)
```

```
## [1] 0.04405847
```

```
sfo_log_12 <-  diff(sfo_log, lag = 12)
plot.ts(sfo_log_12, main = "ln(U_t) differencd at lag 12", ylab = expression(nabla[12]~~ln(U[t])))
var(sfo_log_12)
```

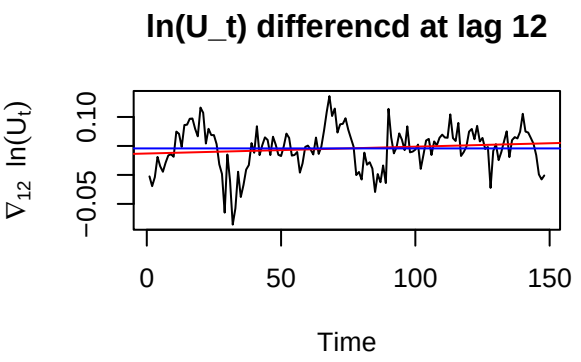
```
## [1] 0.001365461
```

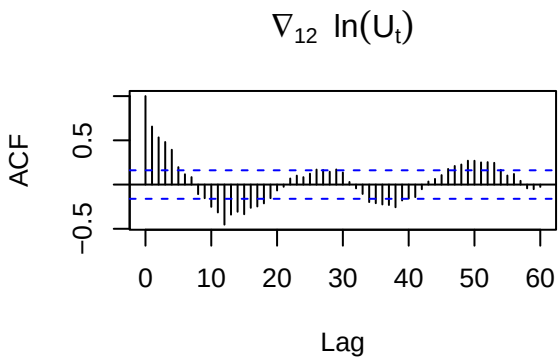
```
fit <- lm(sfo_log_12~as.numeric(1:length(sfo_log_12)))
abline(fit, col = "red")
mean(sfo_log_12)
```

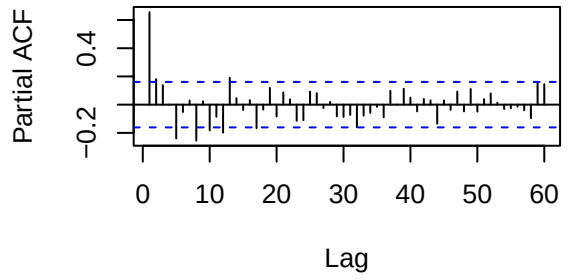
```
## [1] 0.0457408
```

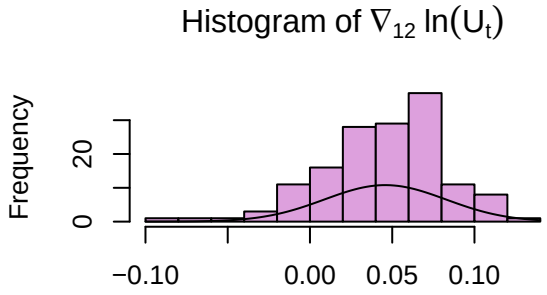
```
abline(h = mean(sfo_log_12), col = "blue")
acf(sfo_log_12, lag.max =60, main = expression(nabla[12]~~ln(U[t])) )
pacf(sfo_log_12, lag.max = 60, main = "")
hist(sfo_log_12, col = "plum", xlab="", main = expression(Histogram~of~nabla[12]~ln(U[t])))
mean <- mean(sfo_log_12)
std <- sqrt(var(sfo_log_12))
```

```
curve(dnorm(x, mean, std), add = TRUE)
```









Differentiating at Lag 1, $\nabla_1 \ln(U_t)$

Now, we difference the log-transformed data at lag 1. Although we removed the trend and the time series is stationary, we still observe strong seasonality in the ACF plot, so we need to differentiate this time series at both lag 1 and lag 12.

```
par(mfrow= c(2,2))
var(sfo_log)
```

```
## [1] 0.04405847
```

```
sfo_log_1 <-  diff(sfo_log, lag = 1)
plot.ts(sfo_log_1, main = "ln(U_t) differenced at lag 1", ylab = expression(nabla[1]~~ln(U[t])))
var(sfo_log_1)
```

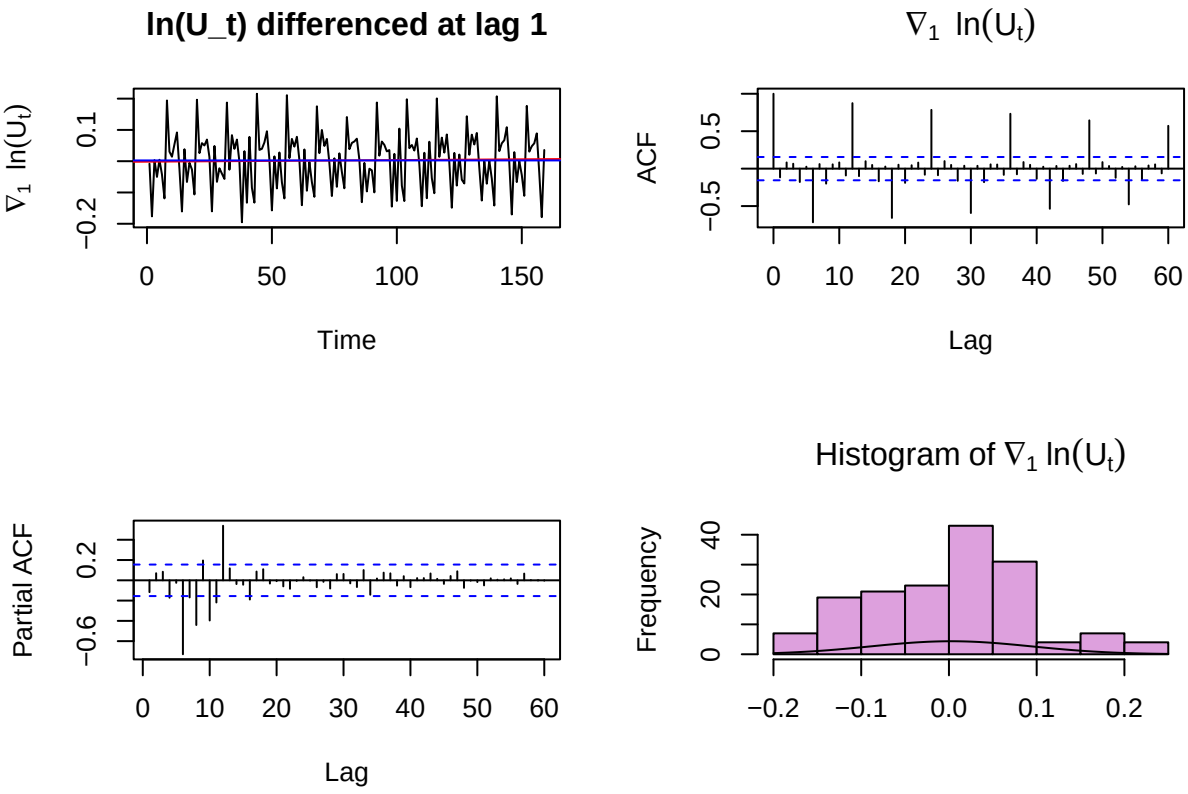
```
## [1] 0.008382194
```

```
fit <- lm(sfo_log_1~as.numeric(1:length(sfo_log_1)))
abline(fit, col = "red")
mean(sfo_log_1)
```

```
## [1] 0.002510284
```

```
abline(h = mean(sfo_log_1), col = "blue")
acf(sfo_log_1, lag.max = 60, main = expression(nabla[1]~~ln(U[t])))
pacf(sfo_log_1, lag.max = 60, main = "")
```

```
hist(sfo_log_1, col = "plum", xlab="", main = expression(Histogram~of~nabla[1]~ln(U[t])))
mean <- mean(sfo_log_1)
std <- sqrt(var(sfo_log_1))
curve(dnorm(x, mean, std), add = TRUE)
```



Differentiating at Lag 1 and Lag 12, $\nabla_1\nabla_{12}\ln(U_t)$

If we differentiate $\ln(U_t)$ at both lag 12 and lag 1, we successfully eliminated trend and seasonality and stabilize the variance. The histogram of $\nabla_1\nabla_{12}\ln(U_t)$ is Symmetric and Gaussian, with a mean value around 0.

```
par(mfrow= c(2, 2))
sfo_12_1 <- diff(sfo_log_12, lag = 1)
plot.ts(sfo_12_1, main = "ln(U_t) differenced at lag 12 and lag 1", ylab = expression(nabla[1]~nabla[12]
fit <- lm(sfo_12_1~ as.numeric(1:length(sfo_12_1)))
abline(fit, col = "red")
mean(sfo_12_1)
```

```
## [1] 1.049508e-05
```

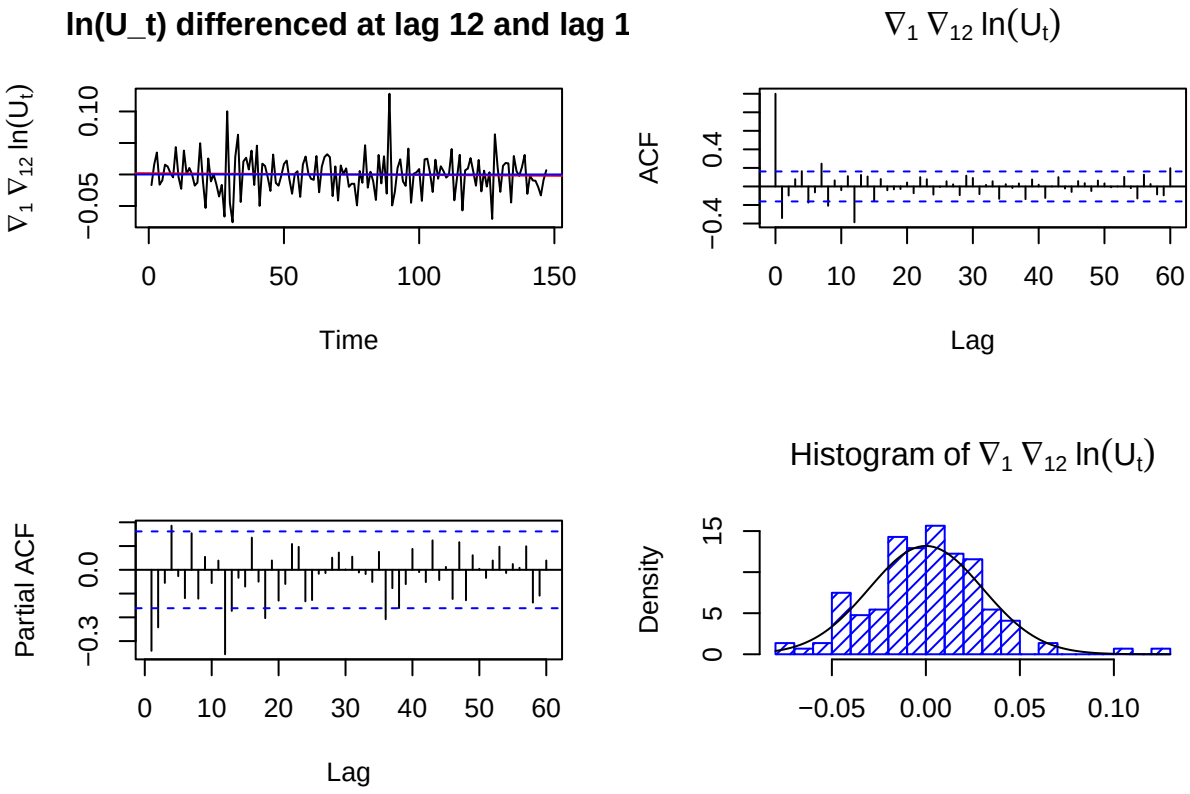
```
abline(h = mean(sfo_12_1), col = "blue")
var(sfo_12_1)
```

```
## [1] 0.0009151471
```

```
acf(sfo_12_1, lag.max = 60, main = expression(nabla[1]~nabla[12]~ln(U[t])))

pacf(sfo_12_1, lag.max = 60, main = "")
```

```
hist(sfo_12_1, density = 20, breaks = 20, col= "blue", xlab = "", prob = TRUE, main = expression(Histogr
mean <- mean(sfo_12_1)
std <- sqrt(var(sfo_12_1))
curve(dnorm(x, mean, std), add = TRUE)
```



Summary Statistics

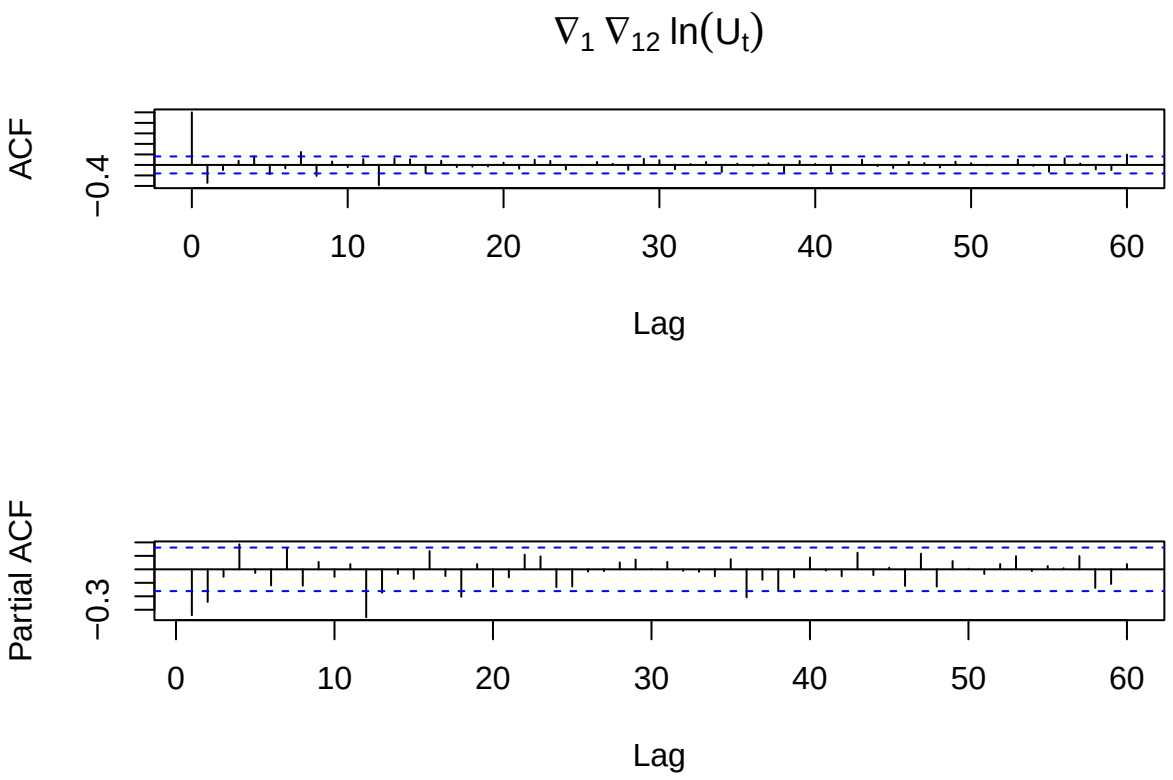
The following table show the summary statistics of each level of differentiation. We can see that differentiating at both lag 1 and lag 12 yield the lowest mean(almost zero) and variance. Hence, $\nabla_1\nabla_{12}\ln(U_t)$ is the best choice.

	Mean	Variance
$\ln(U_t)$	14.39611	0.04405
$\nabla_1 \ln(U_t)$	0.00838	0.002510
$\nabla_{12} \ln(U_t)$	0.04574	0.001365
$\nabla_1\nabla_{12} \ln(U_t)$	1.0495×10^{-5}	0.000915

Identifing Models

```
par(mfrow = c(2,1))
acf(sfo_12_1, lag.max = 60, main = expression(nabla[1]~nabla[12]~ln(U[t])))

pacf(sfo_12_1, lag.max = 60, main = "")
```



According to the acf and pacf of the $\nabla_1 \nabla_{12} \ln(U_t)$, the potential parameter for our models would be:

p = 1, 2, 4 d = 1 q = 1, 5 P = 1 D = 1 Q = 1

We compute the AICc score in a matrix to find which model minimizes the AICc score.

```
library(qpcR)
```

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

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

```
## This build of rgl does not include OpenGL functions.  Use
##  rglwidget() to display results, e.g. via options(rgl.printRglwidget = TRUE).
```

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

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

```
aiccs <- matrix(NA, nr = 4, nc = 5)
dimnames(aiccs) = list(p = 1:4, q = 1:5)

plist = c(1,2,4)
qlist = c(1,5)
for(p in plist)
{
  for(q in qlist)
  {
    aiccs[p, q] = AICc(arima(sfo_log, order = c(p, 1, q), seasonal = list(order = c(1, 1, 1), period =
  }
```

```
}
aiccs
```

```
##    q
## p           1  2  3  4         5
##   1 -688.6679 NA NA NA -691.4925
##   2 -688.8979 NA NA NA -689.8211
##   3        NA NA NA NA        NA
##   4 -689.1456 NA NA NA -688.5713
```

Now, we have calculated the AICc score for every possible model of $\nabla_1\nabla_{12}\ln(U_t)$. From the matrix, we can see that $SARIMA(1,1,5),(1,1,1)_{12}$ has the lowest AICc value, so we let $SARIMA(1,1,5),(1,1,1)_{12}$ be model A, and $SARIMA(2,1,5),(1,1,1)_{12}$ has the second lowest AICc value, but we excluded this model due to the principle of parsimony. $SARIMA(2,1,5),(1,1,1)_{12}$ has the third lowest AICc value, so we let this model to be Model B.

Model A

```
arima(sfo_log, order = c(1, 1, 5), seasonal = list(order = c(1, 1, 1), period = 12, method = "ML"))
```

```
##
## Call:
## arima(x = sfo_log, order = c(1, 1, 5), seasonal = list(order = c(1, 1, 1), period = 12,
##     method = "ML"))
##
## Coefficients:
##          ar1      ma1     ma2     ma3     ma4      ma5    sar1     sma1
##       0.1881  -0.6649  0.0566  0.0012  0.1129  -0.3476  0.0755  -0.9998
## s.e.  0.2531   0.2423  0.1338  0.0986  0.1158   0.1052  0.0958   0.1093
##
## sigma^2 estimated as 0.000376:  log likelihood = 355.22,  aic = -692.45
```

Since the AR and SAR component is not significant in this model(confidence interval covers zero), I decided to remove parameter by changing this model to $SARIMA(0,1,5),(0,1,1)_{12}$.

```
arima(sfo_log, order = c(0, 1, 5), seasonal = list(order = c(0, 1, 1), period = 12, method = "ML"))
```

```
##
## Call:
## arima(x = sfo_log, order = c(0, 1, 5), seasonal = list(order = c(0, 1, 1), period = 12,
##     method = "ML"))
##
## Coefficients:
##          ma1      ma2      ma3     ma4      ma5     sma1
##       -0.4966  -0.0105  -0.0131  0.1463  -0.3522  -0.9997
## s.e.   0.0814   0.0871   0.0942  0.1040   0.1058   0.1233
##
## sigma^2 estimated as 0.0003761:  log likelihood = 354.72,  aic = -695.44
```

Now, we have a cleaner model. Base on the coefficients of this model, we know that ma2 and ma3 is not significant. Hence, we remove those parameters in the next step:

```
#eliminating MA2 and MA3 parameters
fit.A <- arima(sfo_log, order = c(0, 1, 5), seasonal = list(order = c(0, 1, 1), period = 12, method = "
fit.A
```

```
##
```

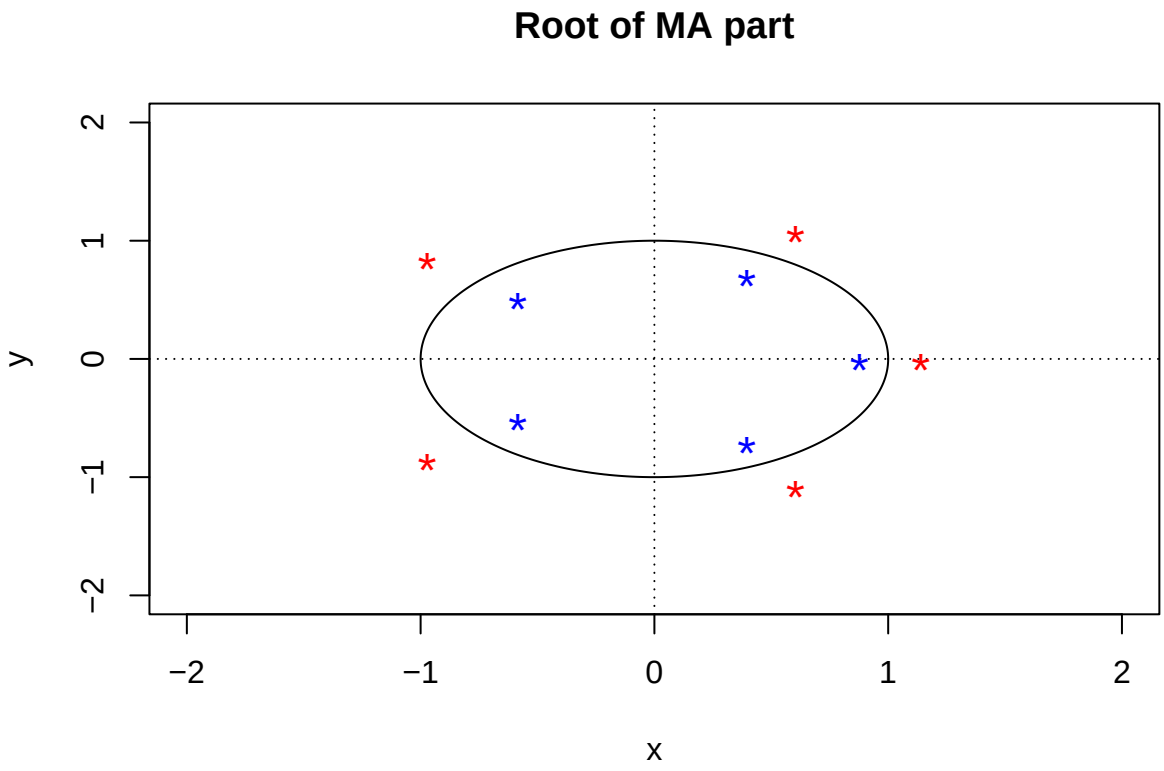
```
## Call:
## arima(x = sfo_log, order = c(0, 1, 5), seasonal = list(order = c(0, 1, 1), period = 12,
##     method = "ML"), fixed = c(NA, 0, 0, NA, NA, NA))
##
## Coefficients:
##           ma1  ma2  ma3     ma4      ma5     sma1
##       -0.4999    0    0  0.1388  -0.3460  -0.9997
## s.e.   0.0789    0    0  0.0952   0.1011   0.1245
##
## sigma^2 estimated as 0.0003764:  log likelihood = 354.7,  aic = -699.39
```

We have obtain our first model: Model A. This model can be expressed as:

$$\nabla_1\nabla_{12}\ln(U_t) = (1-0.4999_{(0.0789)}B+0.1388_{(0.0952)}B^4-0.346_{(0.1011)}B^5)(1-0.9997_{(0.1245)}B^{12})Z_t, \hat{\sigma}_z^2 = 0.0003764$$

Check Invertibility

```
plot.roots(NULL, polyroot(c(1,-0.4999,0,0,0.1388,-0.3460)), main = "Root of MA part")
```

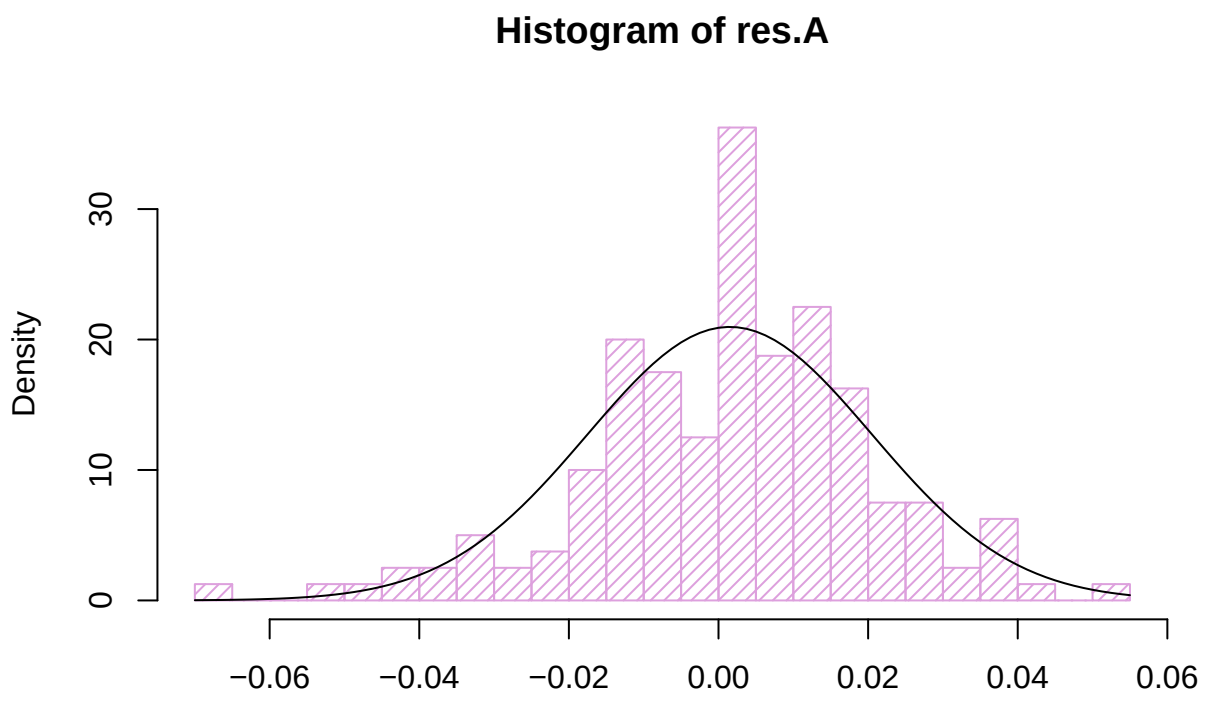


All roots(red color) are outside of the unit circle and inverse root(blue color) are inside the unit circle. Hence, Model A is causal and invertible.

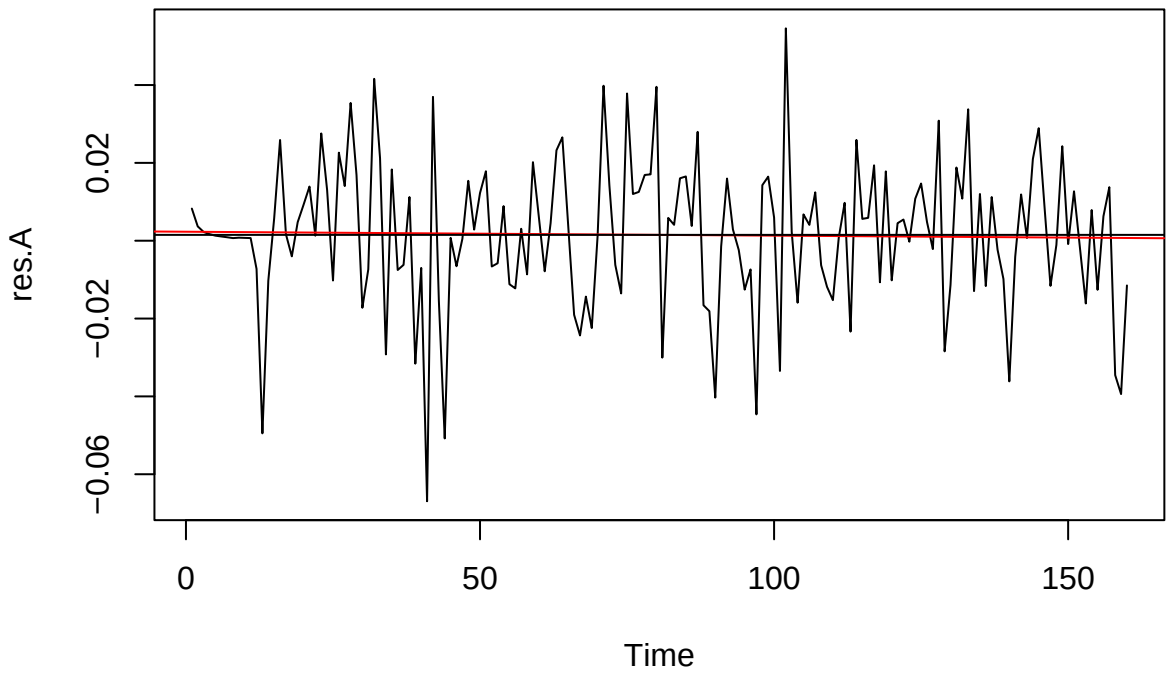
Dignostic Checking for Model A

```
res.A <- residuals(fit.A)
hist(res.A, density = 20, breaks = 20, col = "plum",xlab = "", prob = TRUE)
m <- mean(res.A)
```

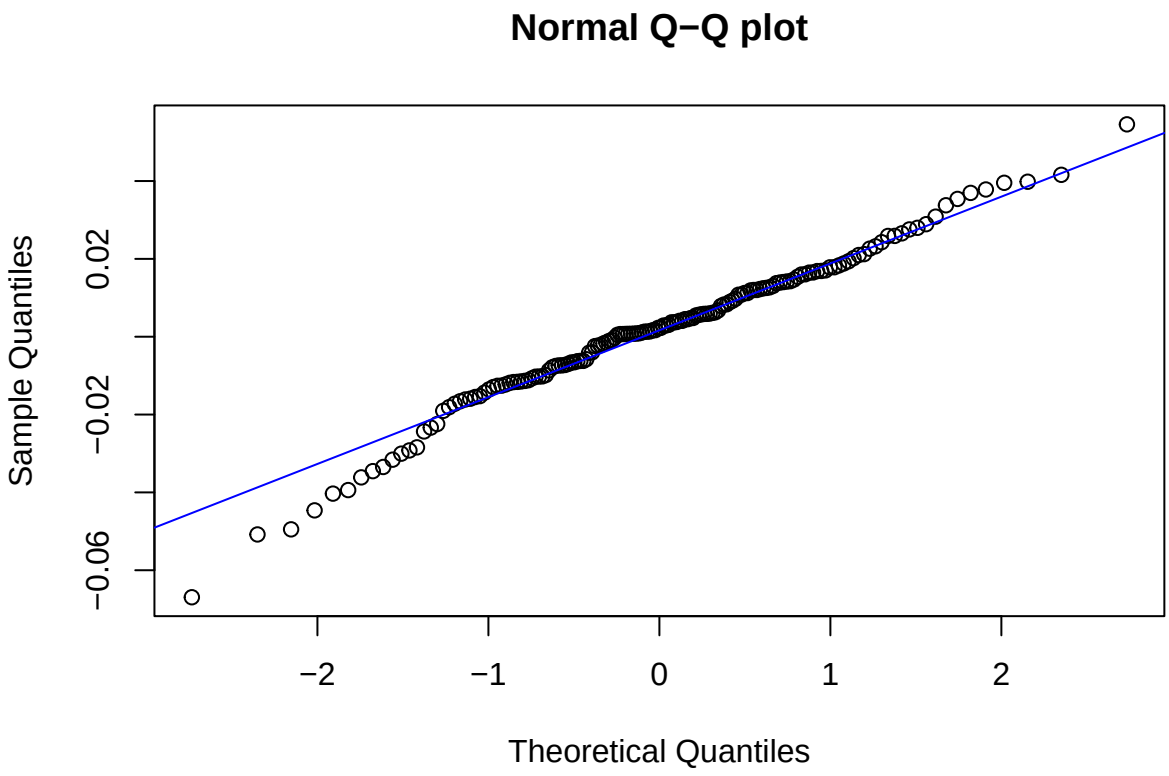
```
std <- sqrt(var(res.A))
curve(dnorm(x, m ,std), add = TRUE)
```



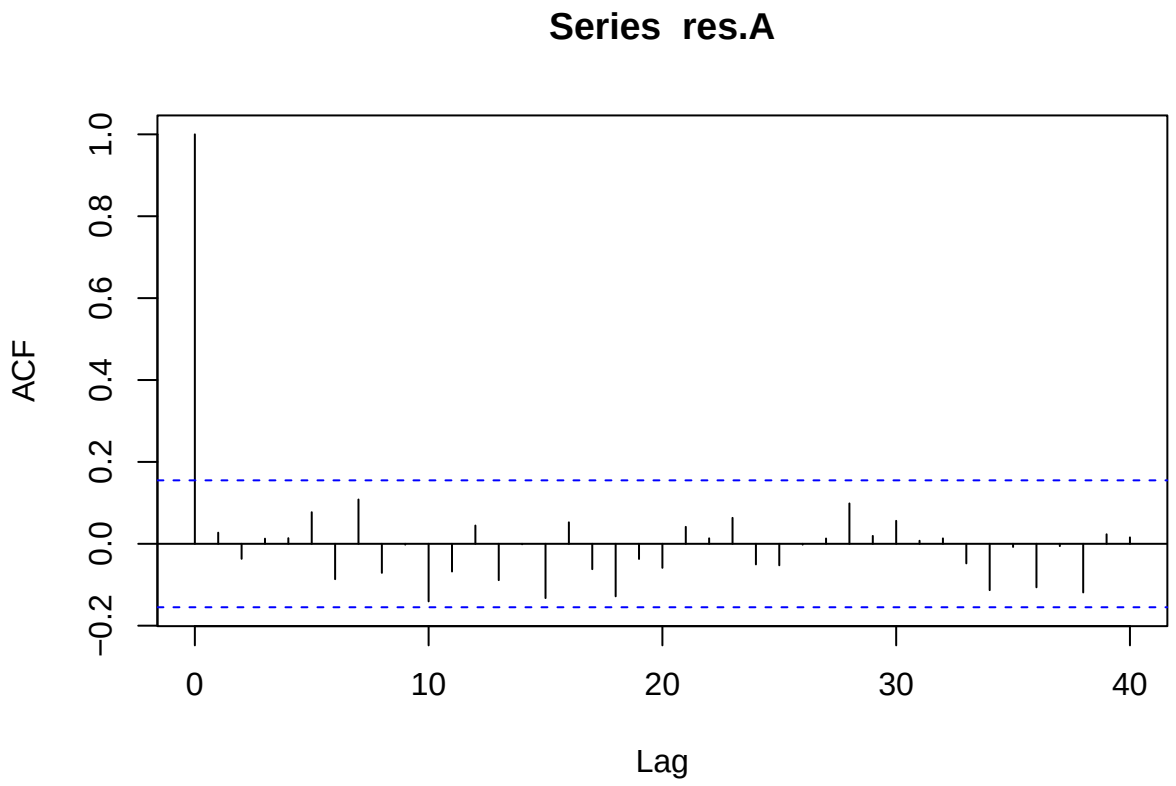

```
plot.ts(res.A)
fitt <- lm(res.A~as.numeric(1:length(res.A)));abline(fitt,col = "red")
abline(h = mean(res.A, col = "blue"))
```



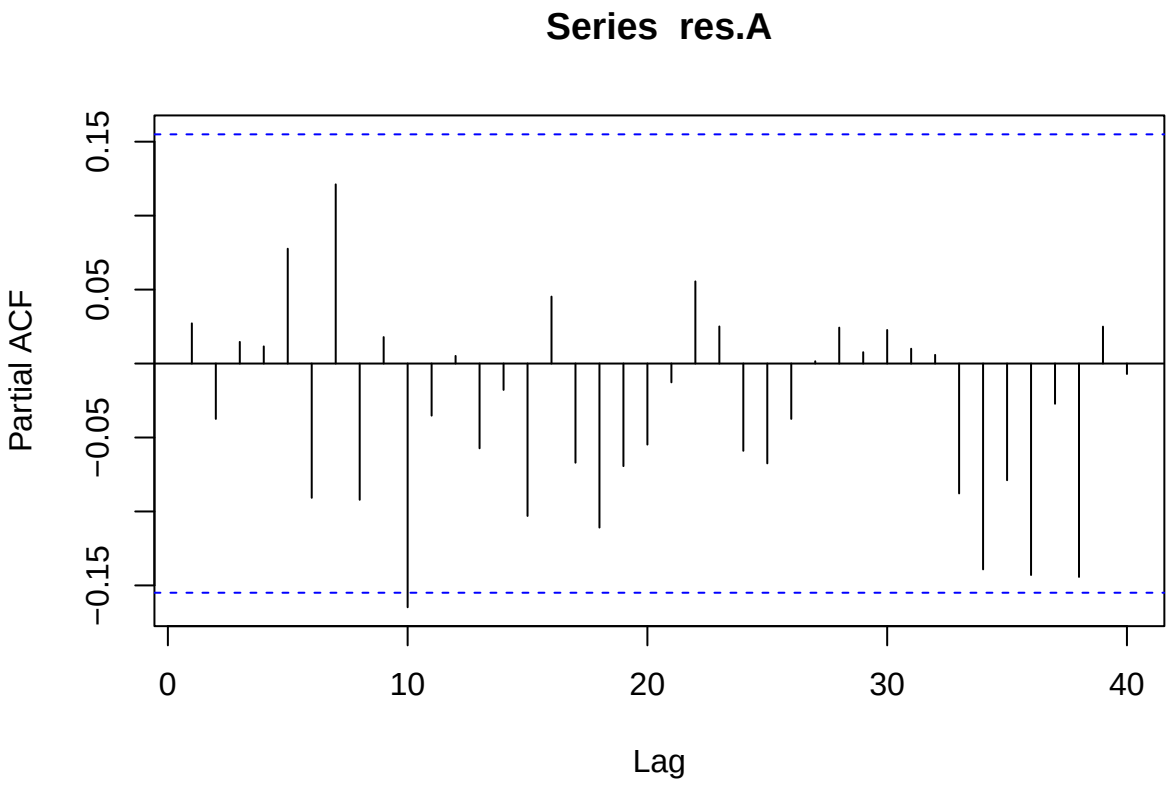

```
qqnorm(res.A, main = "Normal Q-Q plot")
qqline(res.A, col = "blue")
```




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




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



From the above plot, we can see that there is no trend nor visible change of variance for model A. There is no seasonality in the time series plot. The residual looks normal in histogram and Q-Q plot. The sample mean is almost zero: 0.00150196. The acf and pacf of residuals are within confidence interval and can be counted as zeros. Now we perform diagnostic test for Model A, since model A has four parameters, we set "fitdf" to 4.

```
Box.test(res.A, lag = 12, type = c("Box-Pierce"), fitdf = 4)
```

```
##
##  Box-Pierce test
##
## data:  res.A
## X-squared = 9.4283, df = 8, p-value = 0.3075
```

```
Box.test(res.A, lag = 12, type = c("Ljung-Box"), fitdf = 4)
```

```
##
##  Box-Ljung test
##
## data:  res.A
## X-squared = 10.053, df = 8, p-value = 0.2613
```

```
Box.test(res.A^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  res.A^2
## X-squared = 10.191, df = 12, p-value = 0.5992
```

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

```
##
## Call:
## ar(x = res.A, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.000362
```

Model A passed all diagnostic test and all p-value are larger than 0.05. Next, we will check Model B.

Model B

```r
arima(sfo_log, order = c(4, 1, 0), seasonal = list(order = c(1, 1, 1), period = 12, method = "ML"))
```

```
##
## Call:
## arima(x = sfo_log, order = c(4, 1, 0), seasonal = list(order = c(1, 1, 1), period = 12,
##     method = "ML"))
##
## Coefficients:
##           ar1      ar2      ar3     ar4    sar1     sma1
##       -0.4405  -0.2568  -0.0271  0.1613  0.0939  -0.9999
## s.e.   0.0816   0.0910   0.0932  0.0833  0.0911   0.0987
##
## sigma^2 estimated as 0.000394:  log likelihood = 352.92,  aic = -691.84
```

```r
fit.B <- arima(sfo_log, order = c(4, 1, 0), seasonal = list(order = c(1, 1, 1), period = 12, method = "
```

```
## Warning in arima(sfo_log, order = c(4, 1, 0), seasonal = list(order = c(1, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```r
fit.B
```

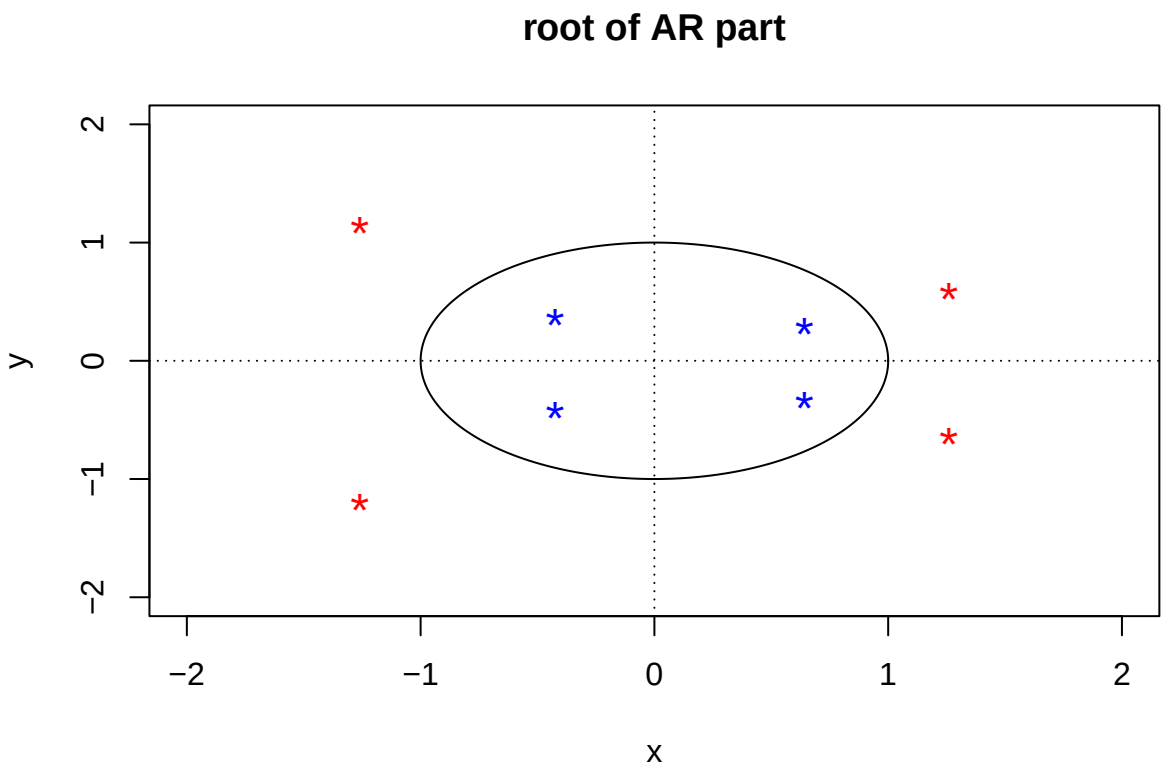
```
##
## Call:
## arima(x = sfo_log, order = c(4, 1, 0), seasonal = list(order = c(1, 1, 1), period = 12,
##     method = "ML"), fixed = c(NA, NA, 0, NA, NA, NA))
##
## Coefficients:
##           ar1      ar2  ar3     ar4    sar1     sma1
##       -0.4337  -0.2433    0  0.1717  0.0979  -1.0001
## s.e.   0.0782   0.0781    0  0.0752  0.0900   0.0991
##
## sigma^2 estimated as 0.0003944:  log likelihood = 352.88,  aic = -693.76
```

We have obtain our second model: Model B. This model can be expressed as:

$$\nabla_1\nabla_{12}\ln(U_t) = (1-0.4337_{(0.0782)}B-0.2433_{(0.0781)}B^2-0.1717_{(0.0752)}B^4)(1-0.0979_{(0.09)}B^{12})X_t-(1-1.0001_{(0.0991)}B^{12})Z_t, \hat{\sigma}_z^2 =$$

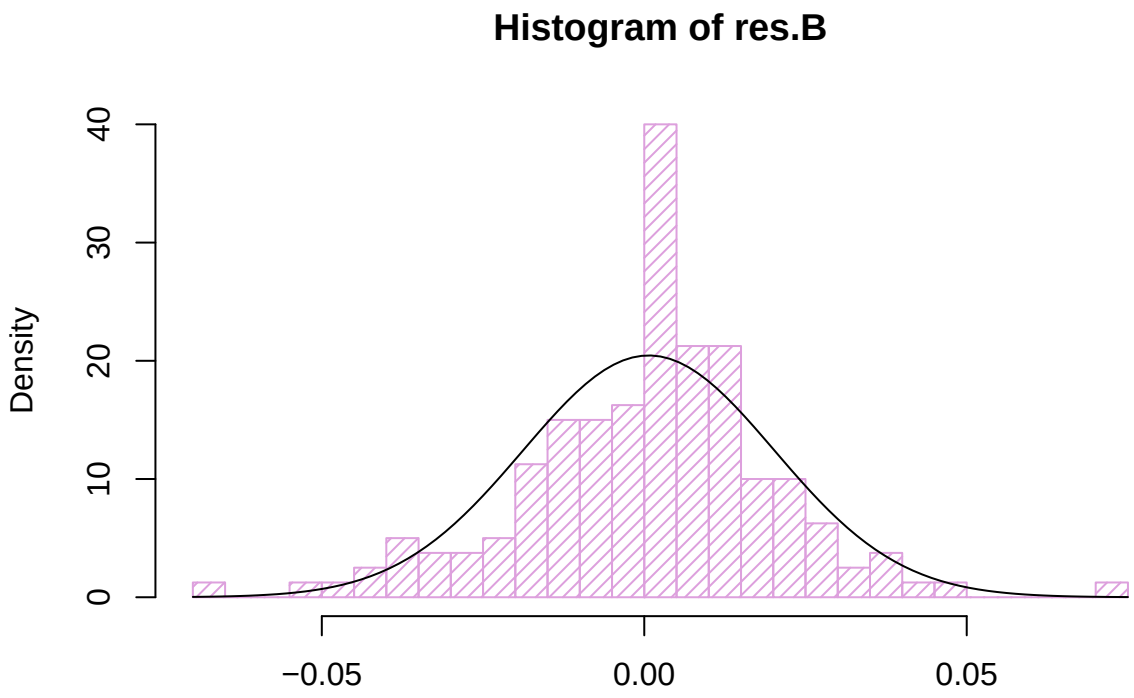
Check Invertibility

```r
plot.roots(NULL, polyroot(c(1, -0.4337, -0.2433, 0, 0.1717)), main = "root of AR part")
```

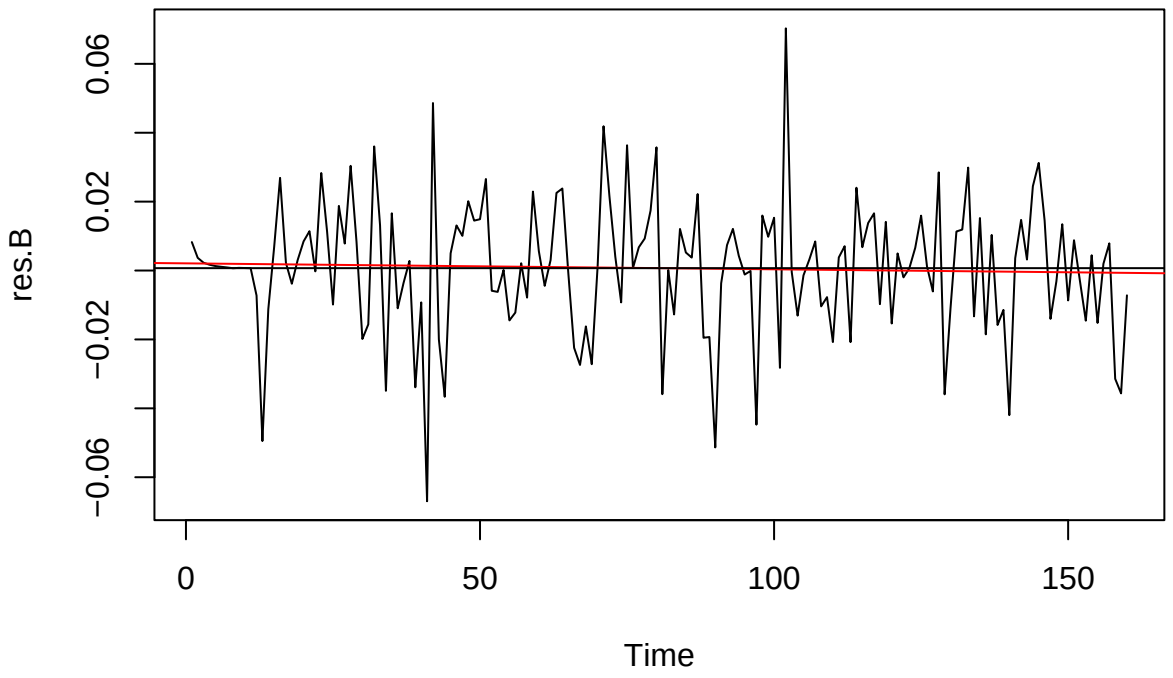


All roots(red color) are outside of the unit circle and inverse root(blue color) are inside the unit circle. Hence, Model B is causal and invertible.

```
res.B <- residuals(fit.B)
hist(res.B, density = 20, breaks = 20, col = "plum",xlab = "", prob = TRUE)
m <- mean(res.B)
std <- sqrt(var(res.B))
curve(dnorm(x, m ,std), add = TRUE)
```

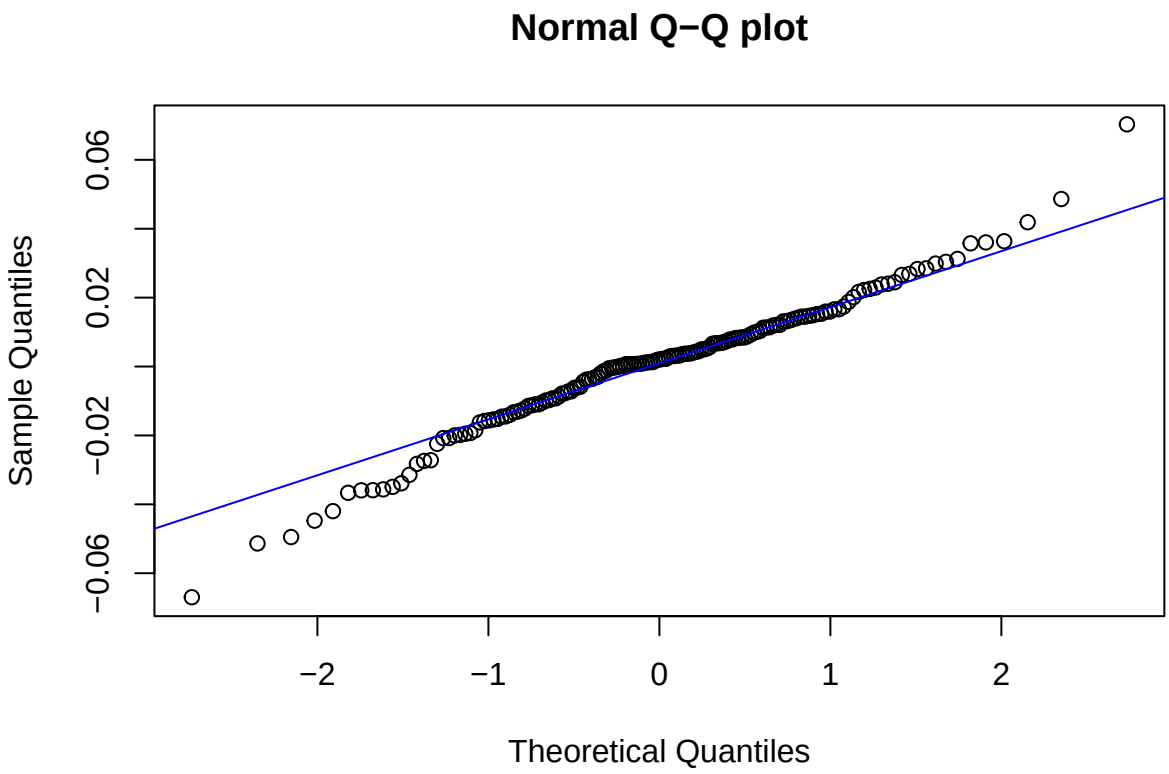



```
plot.ts(res.B)
fitt <- lm(res.B~as.numeric(1:length(res.B)));abline(fitt,col = "red")
abline(h = mean(res.B, col = "blue"))
```

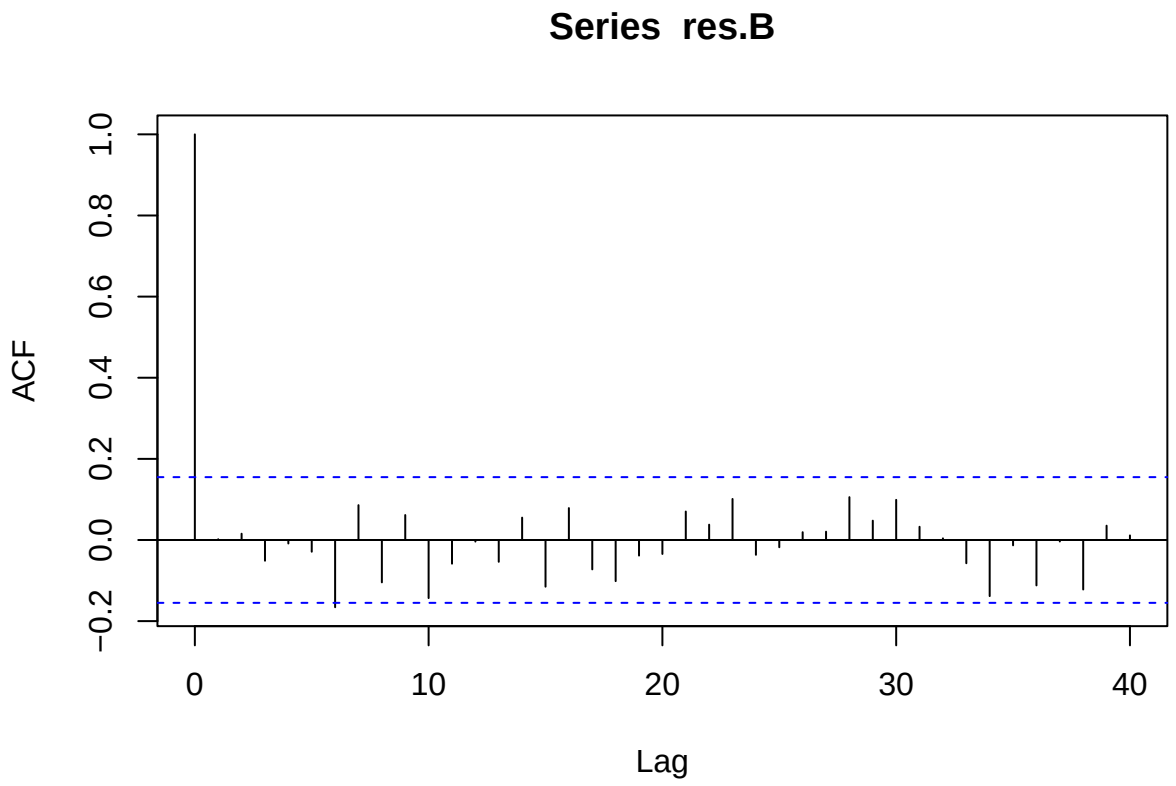


Dignostic Checking for Model B

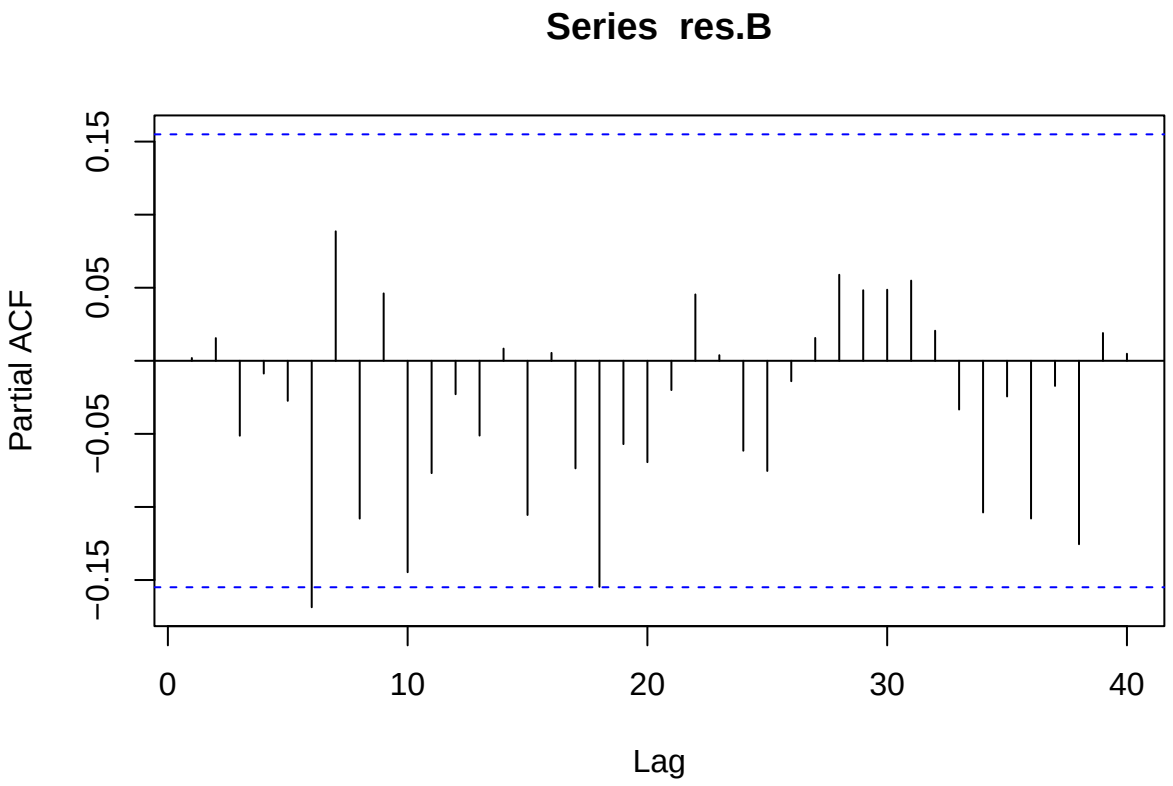
```
qqnorm(res.B, main = "Normal Q-Q plot")
qqline(res.B, col = "blue")
```




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




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



From the above plot, we can see that there is no trend nor visible change of variance for model B. There is no seasonality in the time series plot. The residual looks normal in histogram and Q-Q plot. The sample mean is almost zero: 0.0006918772 The ACF and PACF of residuals are within confidence interval and can be counted as zeros. Now we perform diagnostic test for Model A, since model B has five parameters, we set "fitdf" to 5.

```
Box.test(res.B, lag = 12, type = c("Box-Pierce"), fitdf = 5)
```

```
##
##  Box-Pierce test
##
## data:  res.B
## X-squared = 12.363, df = 7, p-value = 0.08925
```

```
Box.test(res.B, lag = 12, type = c("Ljung-Box"), fitdf = 5)
```

```
##
##  Box-Ljung test
##
## data:  res.B
## X-squared = 13.151, df = 7, p-value = 0.06852
```

```
Box.test(res.B^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  res.B^2
## X-squared = 8.0594, df = 12, p-value = 0.7805
```

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

```
##
## Call:
## ar(x = res.B, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.0003805
```

Both Model A and Model B passed the diagnostic test. We need to choose the best model for forecasting. Here, we compare the AICc score of Model A and Model B.

AICc for Model A

```r
AICc(arima(sfo_log, order = c(0, 1, 5), seasonal = list(order = c(0, 1, 1), period = 12, method = "ML")
```

```
## [1] -696.8706
```

AICc for Model B

```r
AICc(arima(sfo_log, order = c(4, 1, 0), seasonal = list(order = c(1, 1, 1), period = 12, method = "ML")
```

```
## Warning in arima(sfo_log, order = c(4, 1, 0), seasonal = list(order = c(1, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```
## [1] -693.2068
```

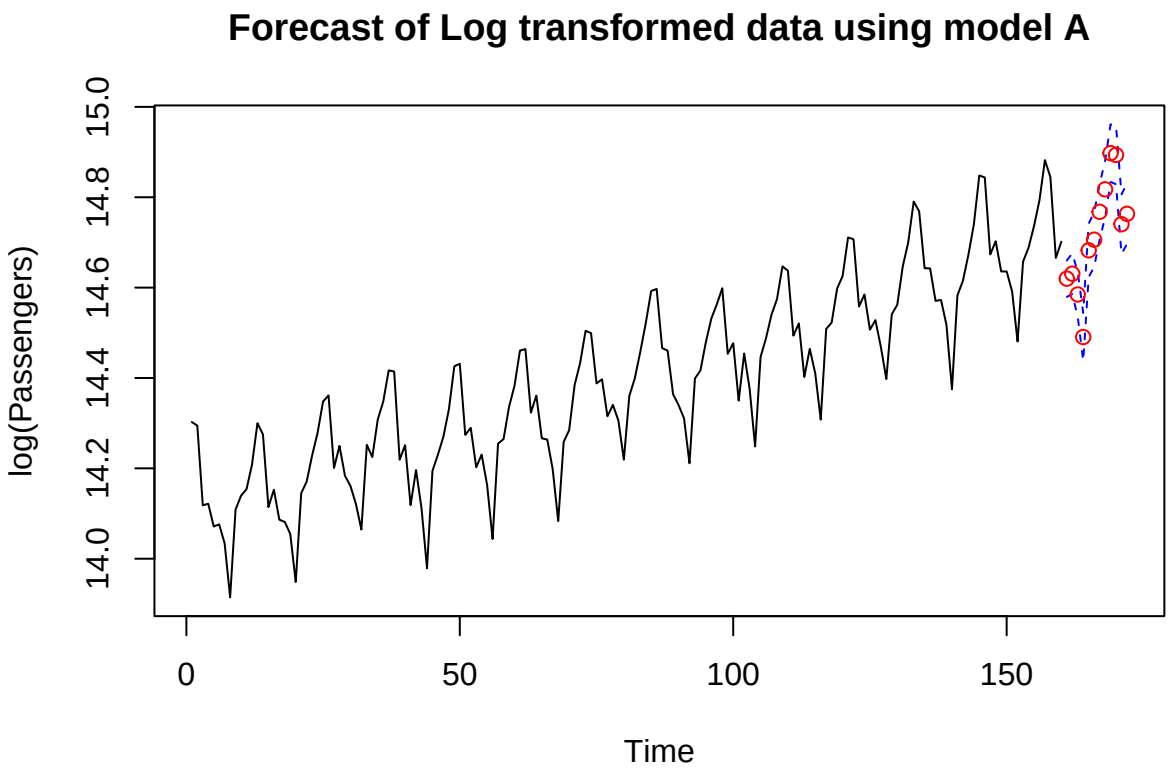
It turns out that Model A has the lowest AICc score, which is -696.8706. Hence, Model A will be chosen as our final model.

Forecasting

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

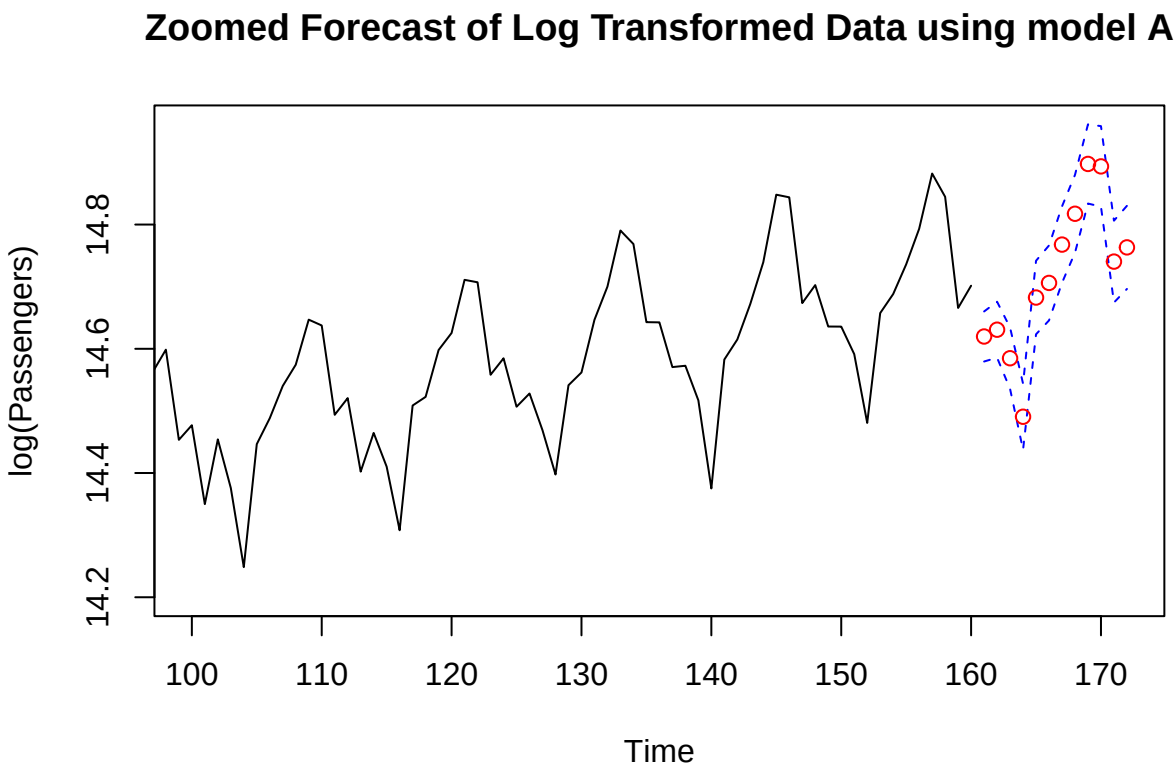
Forecast of Log transformed data using model A

```r
U.tr = pred.tr$pred + 2*pred.tr$se
L.tr = pred.tr$pred - 2*pred.tr$se
ts.plot(sfo_log, xlim = c(1, length(sfo_log)+12), ylim = c(min(sfo_log), max(U.tr)), main = "Forecast o
lines(U.tr, col = "blue", lty = "dashed")
lines(L.tr, col = "blue", lty = "dashed")
points((length(sfo_log)+1):(length(sfo_log)+12), pred.tr$pred, col = "red")
```



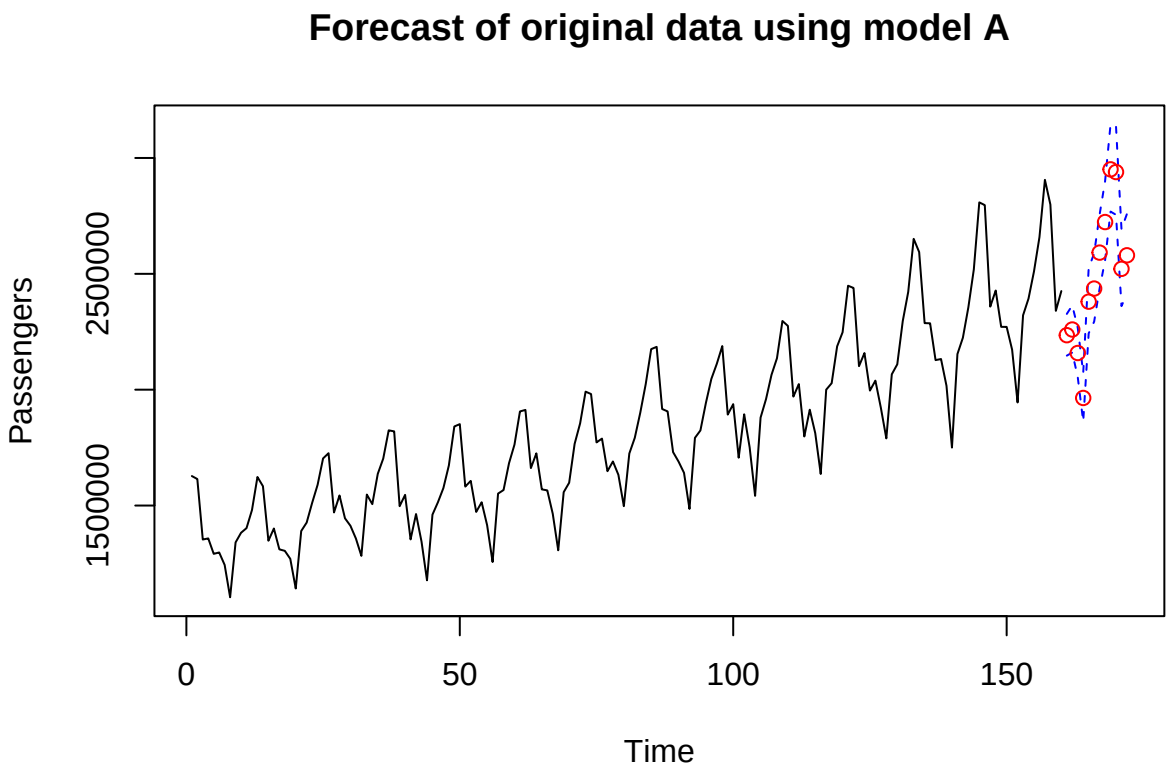
Zoomed Forecast of Log Transformed Data using model A

```
ts.plot(sfo_log, xlim = c(100, length(sfo_log)+12), ylim = c(14.2, max(U.tr)), main = "Zoomed Forecast
lines(U.tr, col = "blue", lty = "dashed")
lines(L.tr, col = "blue", lty = "dashed")
points((length(sfo_log)+1):(length(sfo_log)+12), pred.tr$pred, col = "red")
```



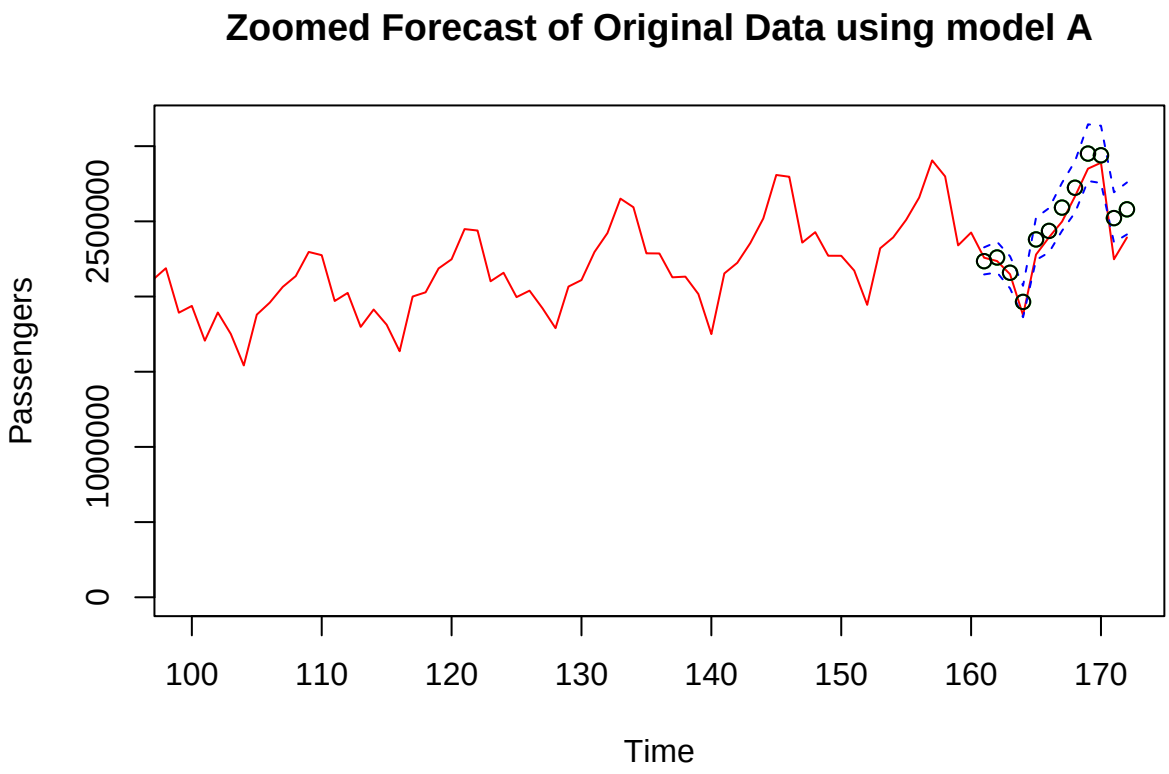
Forecast of original data using model A

```
pred.orig <-  exp(pred.tr$pred)
U = exp(U.tr)
L = exp(L.tr)
ts.plot(train.sfo, xlim = c(1, length(train.sfo)+12), ylim = c(min(train.sfo), max(U)), main = "Forecas
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points((length(train.sfo)+1):(length(train.sfo)+12), pred.orig, col = "red")
```



Zoomed Forecast of Original Data using model A

```
ts.plot(at_df$Monthly_Passenger, xlim = c(100,length(train.sfo)+12), ylim = c(250,max(U)), col="red", ma
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(train.sfo)+1):(length(train.sfo)+12), pred.orig, col="green")
points((length(train.sfo)+1):(length(train.sfo)+12), pred.orig, col="black")
```



Conclusion

At the end, we decide to choose the $SARIMA(0,1,5),(0,1,1)_{12}$ model to predict the SFO airport passenger data, this model passed all diagnostic test and perform well in forecasting future data, as we observe that the confidence interval of our model successfully captures our test data.