Amazon

Sahba Fahandezh

September 8, 2018

```r
library(readxl)
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.5.1
```

```r
require(astsa)
```

```
## Loading required package: astsa
```

```
##
## Attaching package: 'astsa'
```

```
## The following object is masked from 'package:forecast':
##
##     gas
```

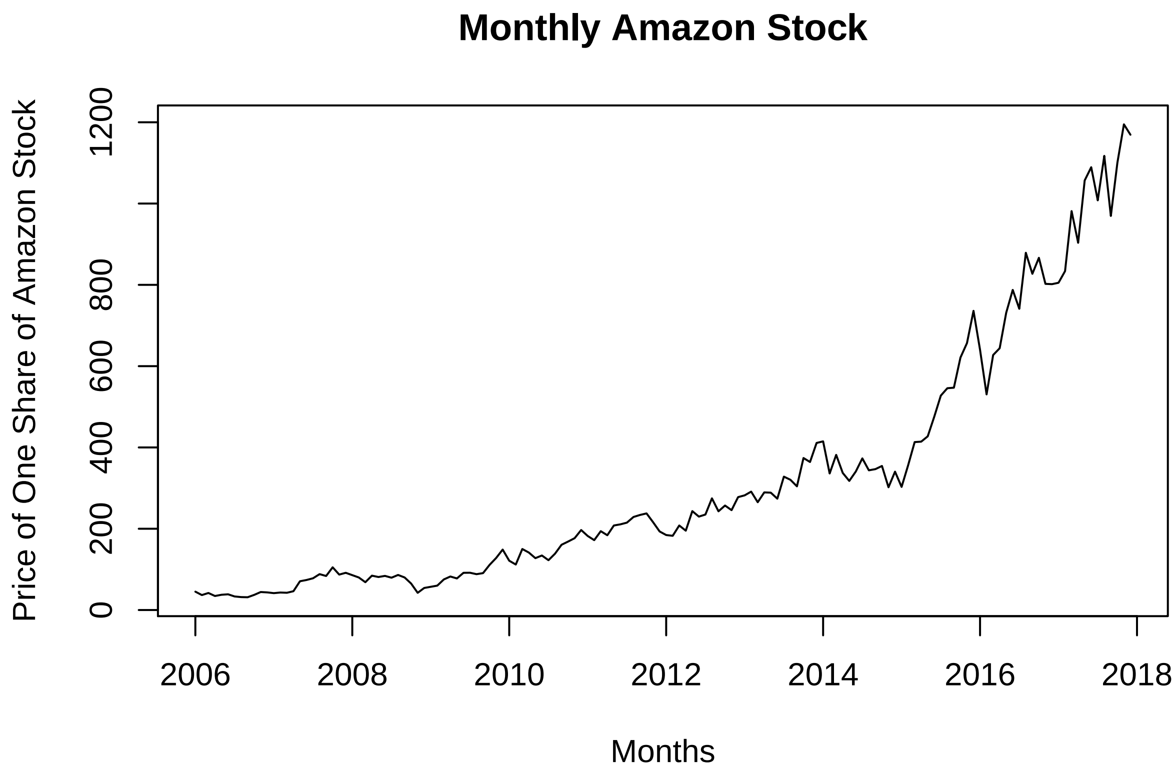
```r
library(dplyr)
```

```
##
## 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
```

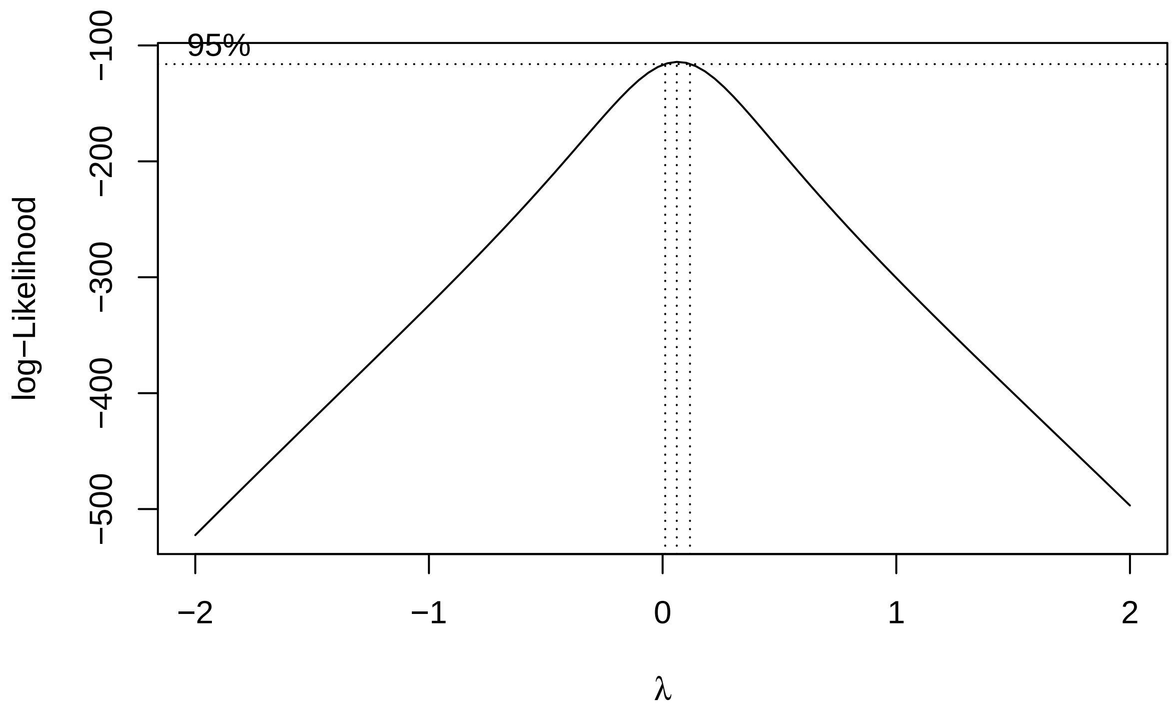
```r
#Converting data series into Time Series object
#Amazon Stock Market From Jan 2006 to Jan 2018 = AMZN

AMZN <- read.csv("C:/Sahba/Projects/AMZN_2006-01-01_to_2018-01-01.csv")
AMZN <- AMZN[,-c(2:4,6:9)]
AMZN <- na.omit(AMZN)
View(AMZN)

TS.AMZN <- ts(AMZN$Monthly, start = c(2006,1),deltat = 1/12)
plot.ts(TS.AMZN, xlab = "Months", ylab = "Price of One Share of Amazon Stock", main= "Monthly Amazon St
```




```r
library(MASS)
```

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

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

```r
t <- 1:length(AMZN$Monthly)
bctransform <- boxcox(AMZN$Monthly~t)
```




```
lambda <- bctransform$x[which(bctransform$y == max(bctransform$y))]

#BoxCox transformation
data.bcNTS <- (1/lambda)*((AMZN$Monthly^lambda)-1)
#Square Root Transformation
data.sqrtNTS <- sqrt(AMZN$Monthly)
#Log Transformation
data.logNTS <- log(AMZN$Monthly)

#Ploting Box Cox, Squareroot, and Log transformations to compare and contrats
op <- par(mfrow = c(1,3))

plot.ts(data.bcNTS, main = "BC")
bc.lm <- lm(data.bcNTS~as.numeric(1:length(data.bcNTS)))
bc.ci <- predict(bc.lm, interval='prediction')
```

```
## Warning in predict.lm(bc.lm, interval = "prediction"): predictions on current data refer to _future_
```

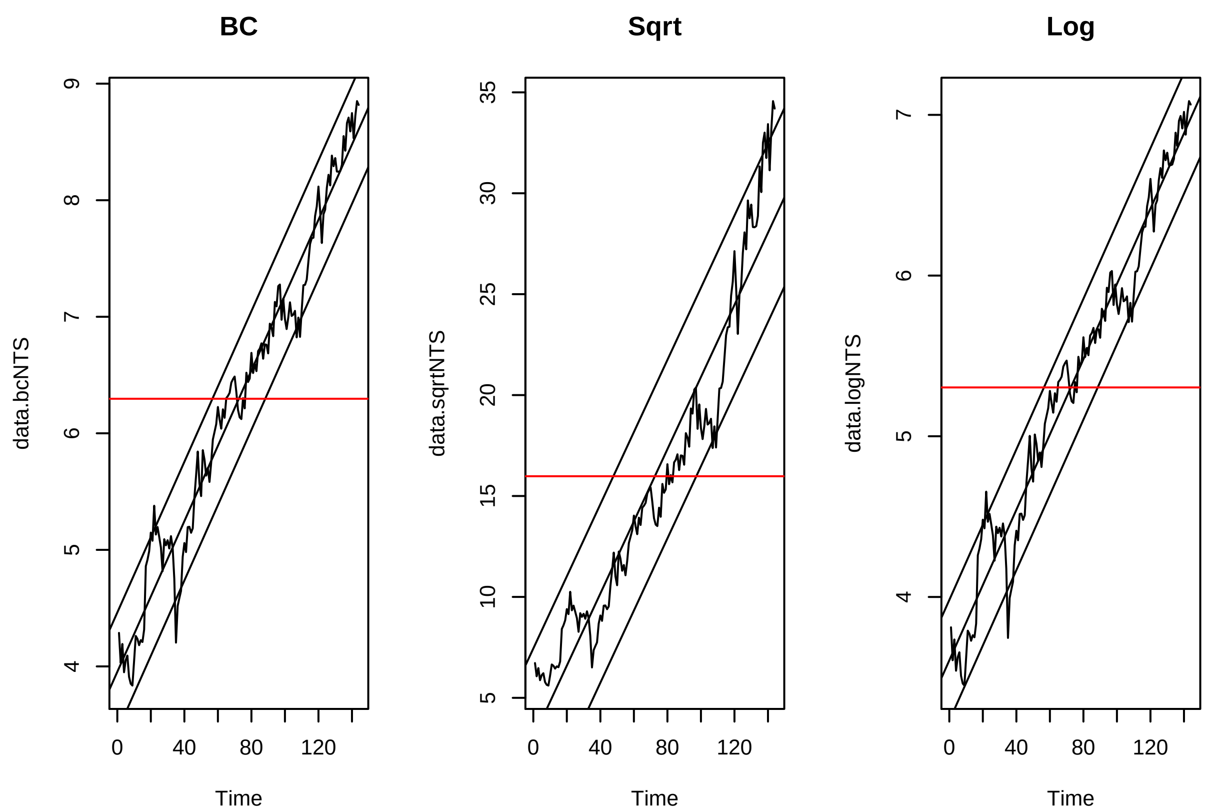
```
abline(bc.lm)
abline(lm(bc.ci[,2]~as.numeric(1:length(data.bcNTS))))
abline(lm(bc.ci[,3]~as.numeric(1:length(data.bcNTS))))
abline(h=mean(data.bcNTS), col="red")

plot.ts(data.sqrtNTS, main = "Sqrt")
sqrt.lm <- lm(data.sqrtNTS~as.numeric(1:length(data.sqrtNTS)))
sqrt.ci <- predict(sqrt.lm, interval='prediction')
```

```
## Warning in predict.lm(sqrt.lm, interval = "prediction"): predictions on current data refer to _future
```

```r
abline(sqrt.lm)
abline(lm(sqrt.ci[,2]~as.numeric(1:length(data.sqrtNTS))))
abline(lm(sqrt.ci[,3]~as.numeric(1:length(data.sqrtNTS))))
abline(h=mean(data.sqrtNTS), col="red")

plot.ts(data.logNTS, main = "Log")
log.lm <- lm(data.logNTS~as.numeric(1:length(data.logNTS)))
log.ci <- predict(log.lm, interval='prediction')
```

```
## Warning in predict.lm(log.lm, interval = "prediction"): predictions on current data refer to _future_
```

```r
abline(log.lm)
abline(lm(log.ci[,2]~as.numeric(1:length(data.logNTS))))
abline(lm(log.ci[,3]~as.numeric(1:length(data.logNTS))))
abline(h=mean(data.logNTS), col="red")
```




```r
par(op)

var(data.bcNTS)
```

```
## [1] 1.882244
```

```r
var(data.sqrtNTS)
```

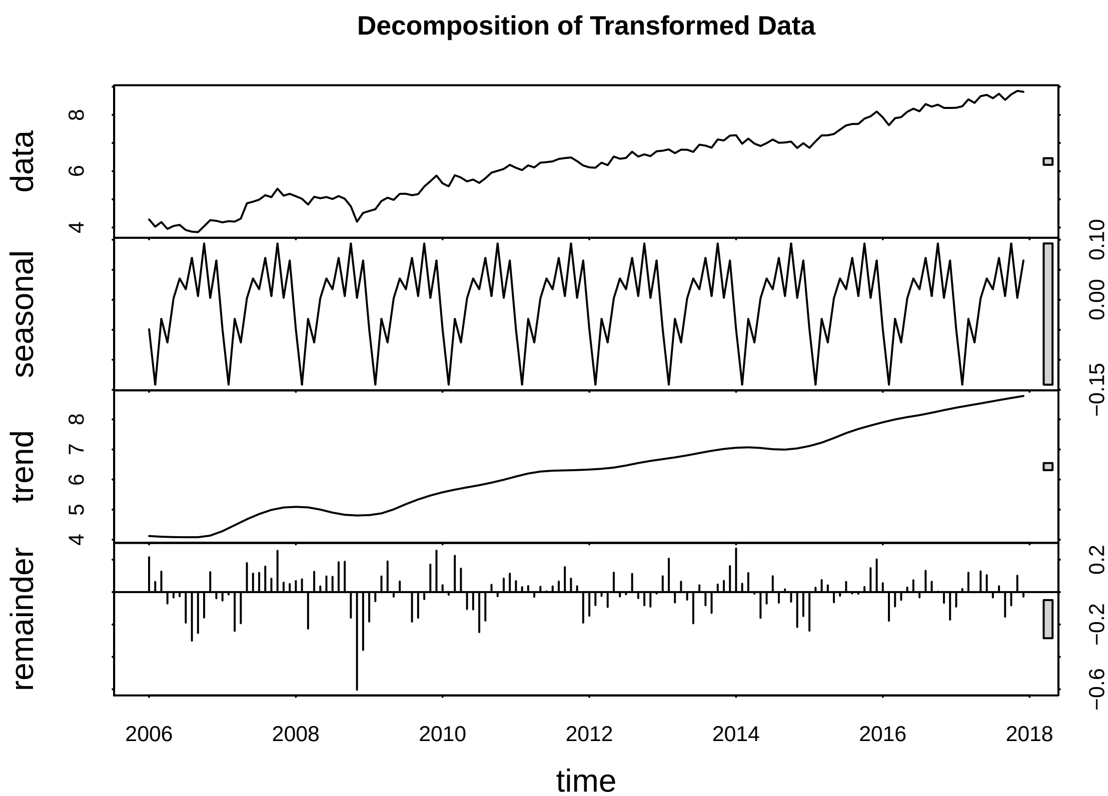
```
## [1] 60.47593
```

```r
var(data.logNTS)
```

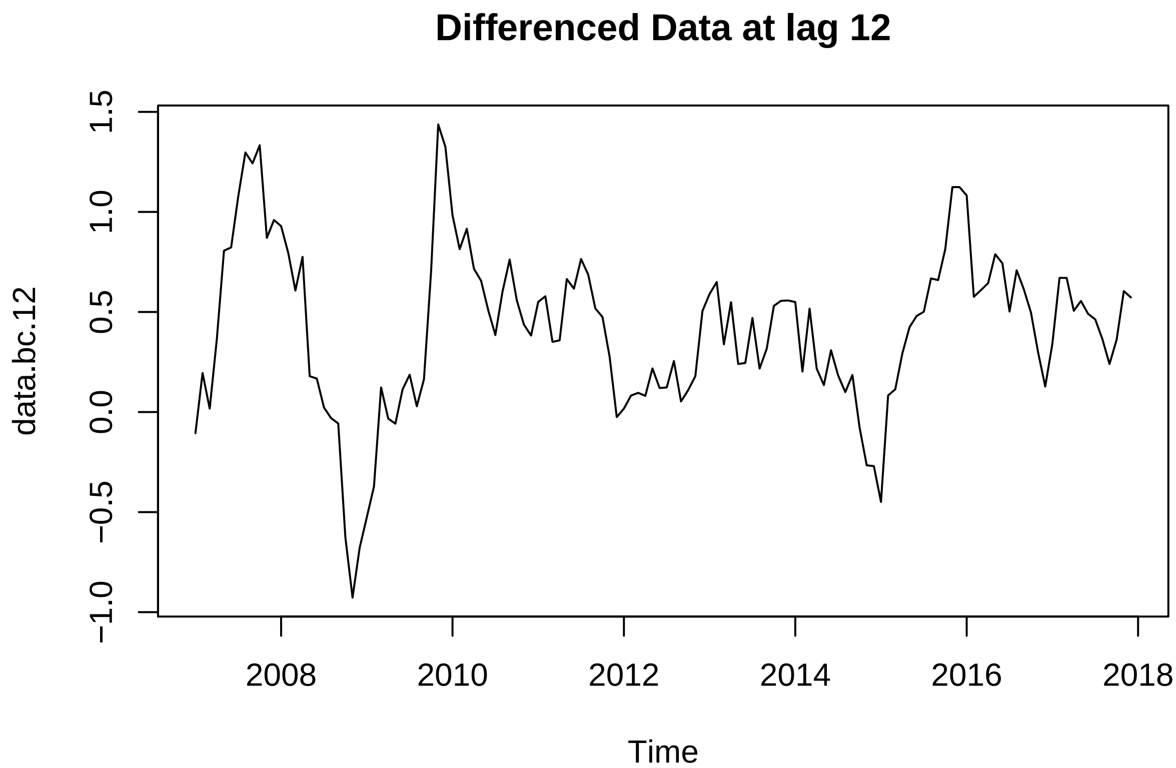
```
## [1] 0.9899928
```

```r
# square transformation is not good since it doesn't have a linear trend.
# So we have boxcox and log which is very similar. We choose log since its variance is slighlty smaller
```

```r
#APPENDIX D - REMOVING SEASONALITY AND TREND

data.bc <- ts((1/lambda)*((TS.AMZN^lambda)-1),start = c(2006,1),deltat = 1/12)

#Decomposition of transformed data
y <- stl(data.bc,s.window = "period")
plot(y,main = "Decomposition of Transformed Data")
```




```r
# we see that there is both seasonality and a trend

# Differencing at lag 12 to remove seasonality
data.bc.12 <- diff(data.bc, lag = 12)
plot(data.bc.12, main = "Differenced Data at lag 12")
```



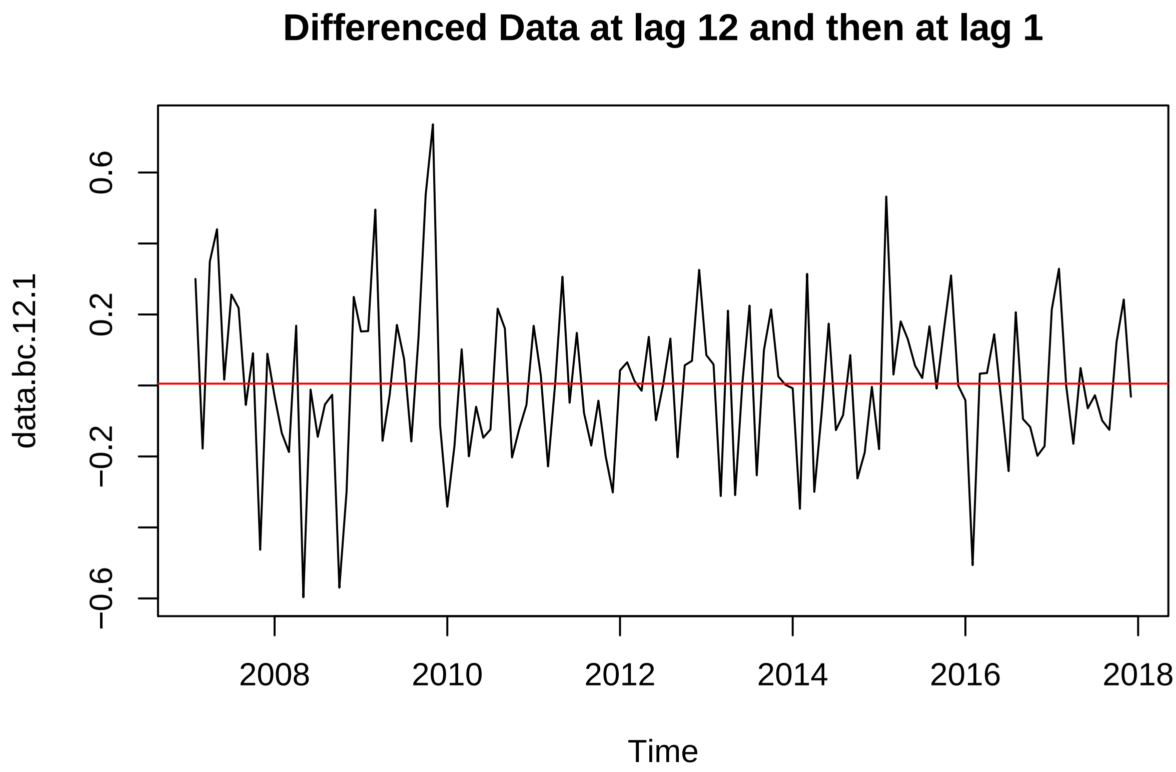

```r
# Differencing at lag 1 to remove trend
data.bc.12.1 <- diff(data.bc.12, lag = 1)
var(data.bc.12.1)
```

```
## [1] 0.04677938
```

```r
#Test to see if we should difference once more at lag 1
data.bc.12.2 <- diff(data.bc.12, lag = 1, differences = 2)
var(data.bc.12.2)
```

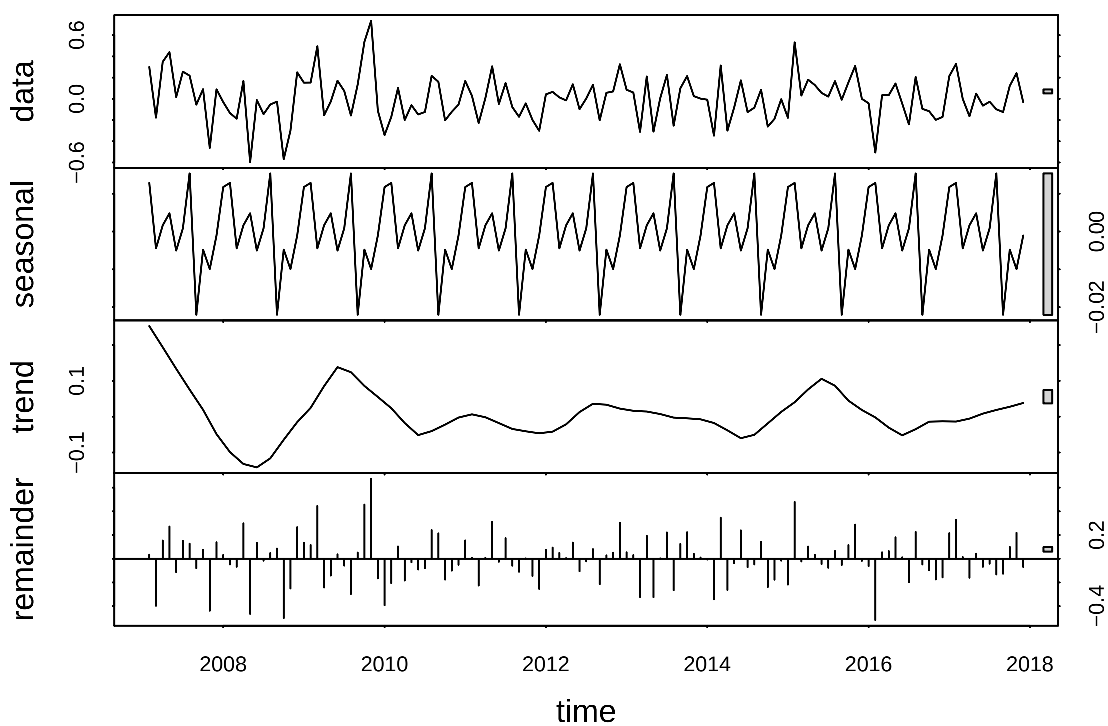
```
## [1] 0.08926294
```

```r
#We see that variance increases which means we have overdifferenced so we go back
#to using the data that is only differenced once at lag 1


plot.ts(data.bc.12.1, main = "Differenced Data at lag 12 and then at lag 1")
abline(h=mean(data.bc.12.1), col="red")
```



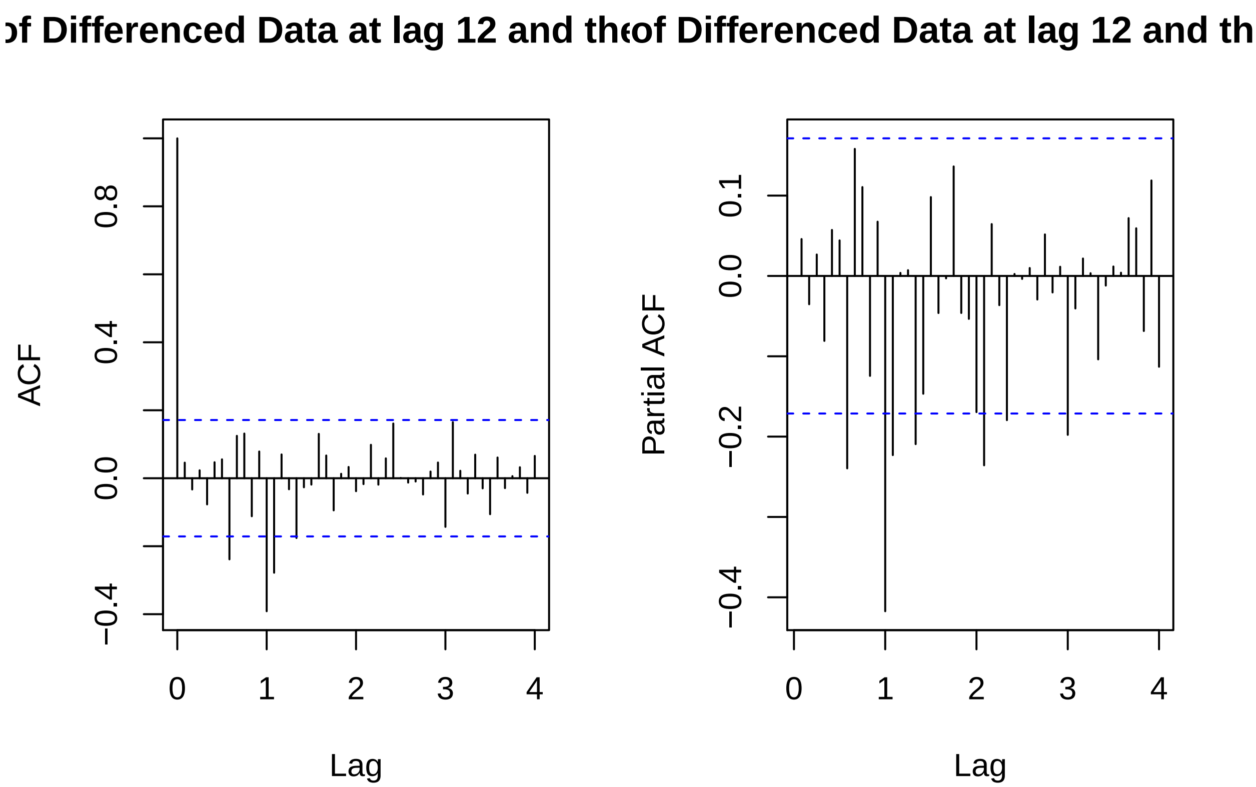

```
#Decomposition After Differencing
z <- stl(data.bc.12.1,s.window = "period")
plot(z,main = "Decomposition of Differenced Data at lag 12 and then at lag 1")
```

Decomposition of Differenced Data at lag 12 and then at lag 1




```r
# We see that this data is not stationary and does not have seasonality
```

```r
#APPENDIX E - MODEL IDENTIFICATION

op <- par(mfrow = c(1,2))

acf(data.bc.12.1, lag.max = 48, main = "ACF of Differenced Data at lag 12 and then at lag 1")
pacf(data.bc.12.1, lag.max = 48, main = "PACF of Differenced Data at lag 12 and then at lag 1")
```




```r
par(op)

# We've seen lag 2 for ACF and a Geometric decay for PACF which belongs to MA series.
# SARIMA(2,0,0)X(1,0,1)12
```

```r
auto.arima(data.bc.12.1, allowdrift = FALSE, trace = TRUE)
```

```
##
##  ARIMA(2,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)            with non-zero mean : -26.31115
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : -46.52617
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)            with zero mean     : -28.29832
##  ARIMA(1,0,0)            with non-zero mean : -24.49429
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : -56.20141
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : -56.93108
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : -56.46651
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : -58.91733
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : -50.52186
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : -58.18478
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : -58.43925
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : Inf
##
##  Best model: ARIMA(0,0,0)(2,0,0)[12] with zero mean
```

```
## Series: data.bc.12.1
## ARIMA(0,0,0)(2,0,0)[12] with zero mean
##
## Coefficients:
##          sar1     sar2
##       -0.5703  -0.3197
## s.e.   0.0892   0.0942
##
## sigma^2 estimated as 0.0348:  log likelihood=32.55
## AIC=-59.11   AICc=-58.92   BIC=-50.48
```

```
# gives best model to be ARIMA(0,0,0)(2,0,0)[12] with zero mean
```

```
ar(data.bc.12.1, method = "yule-walker")
```

```
##
## Call:
## ar(x = data.bc.12.1, method = "yule-walker")
##
## Coefficients:
##       1        2        3        4        5        6        7        8
##  0.0126  -0.0193  -0.0658  -0.0732  -0.0523   0.0053  -0.2004   0.1559
##       9       10       11       12       13       14       15       16
##  0.0851  -0.1505   0.0952  -0.3781  -0.2328  -0.0108   0.0127  -0.2030
##      17
## -0.1467
##
## Order selected 17  sigma^2 estimated as  0.0346
```

```
#Gives AR(12)
```

```
#We compare the following 6 models

fit.1020 <- arima(data.bc.12.1, order = c(1,0,0), seasonal = list(order = c(2,0,1),period = 12),
                  method = "ML", include.mean = FALSE)
fit.0120 <- arima(data.bc.12.1, order = c(0,0,1), seasonal = list(order = c(2,0,1),period = 12),
                  method = "ML", include.mean = FALSE)
fit.0020 <- arima(data.bc.12.1, order = c(0,0,0), seasonal = list(order = c(2,0,1),period = 12),
                  method = "ML", include.mean = TRUE)
fit.0120 <- arima(data.bc.12.1, order = c(0,0,1), seasonal = list(order = c(2,0,1),period = 12),
                  method = "ML", include.mean = FALSE)
fit.1020 <- arima(data.bc.12.1, order = c(1,0,0), seasonal = list(order = c(2,0,1),period = 12),
                  method = "ML", include.mean = FALSE)
fit.ar <- arima(data.bc.12.1, order = c(6,0,0), seasonal = list(order = c(0,0,0),period = 12),
                  method = "ML", include.mean = FALSE)


library(qpcR)
```

```
## Warning: package 'qpcR' was built under R version 3.5.1

## Loading required package: minpack.lm

## Warning: package 'minpack.lm' was built under R version 3.5.1

## Loading required package: rgl

## Warning: package 'rgl' was built under R version 3.5.1
```

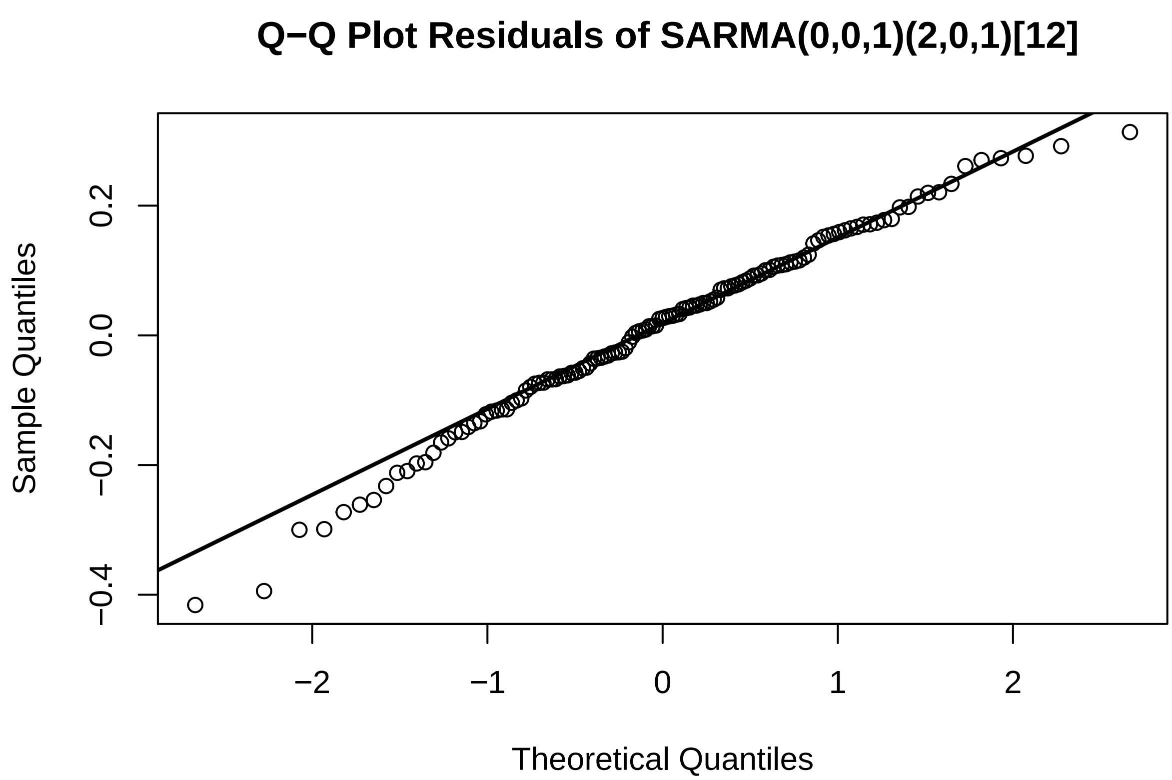
```
## Loading required package: robustbase

## Warning: package 'robustbase' was built under R version 3.5.1

## Loading required package: Matrix
```

```
matrix(c(AICc(fit.1020),AICc(fit.0120),AICc(fit.0020),AICc(fit.0120),
         AICc(fit.1020),AICc(fit.ar)),nrow = 1,
        dimnames = list("AICc",c("fit.1020","fit.0120","fit.0020","fit.0120"
                                  ,"fit.1020","fit.ar")))
```

```
##       fit.1020  fit.0120  fit.0020  fit.0120  fit.1020    fit.ar
## AICc -90.99846 -91.16659 -90.57665 -91.16659 -90.99846 -17.80489
```

```
#We see that the model with the lowest AICc is SARMA(0,0,1)(2,0,1)[12] (w/ 0 mean) with a value
#of -91.16659
```

```
#APPENDIX F - DIAGNOSTIC CHECKS

#Diagnostics for SARMA(0,0,1)(2,0,1)[12]

res0120 <- residuals(fit.0120)

#Normality of errors

qqnorm(res0120, main = " Q-Q Plot Residuals of SARMA(0,0,1)(2,0,1)[12]")
qqline(res0120, lw = 2)
```



```r
#Q-Q Plot is fairly straight so it passes
shapiro.test(res0120)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res0120
## W = 0.98756, p-value = 0.2839
```

```r
#Shapiro-Wilk normality test
#Passes shapiro.test

#Detection of Serial correlation
#chose lag h=min(2m,T/5) based on https://robjhyndman.com/hyndsight/ljung-box-test/

h <- min(2*12,length(res0120)/5)
h
```

```
## [1] 24
```

```r
#Box-Pierce test:

Box.test(res0120, lag = 12, type = "Box-Pierce", fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res0120
## X-squared = 16.295, df = 10, p-value = 0.09149
```

```r
Box.test(res0120, lag = h, type = "Box-Pierce", fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res0120
## X-squared = 45.59, df = 22, p-value = 0.002232
```

```r
#Box-Ljung test:

Box.test(res0120, lag = 12, type = "Ljung", fitdf = 2)
```

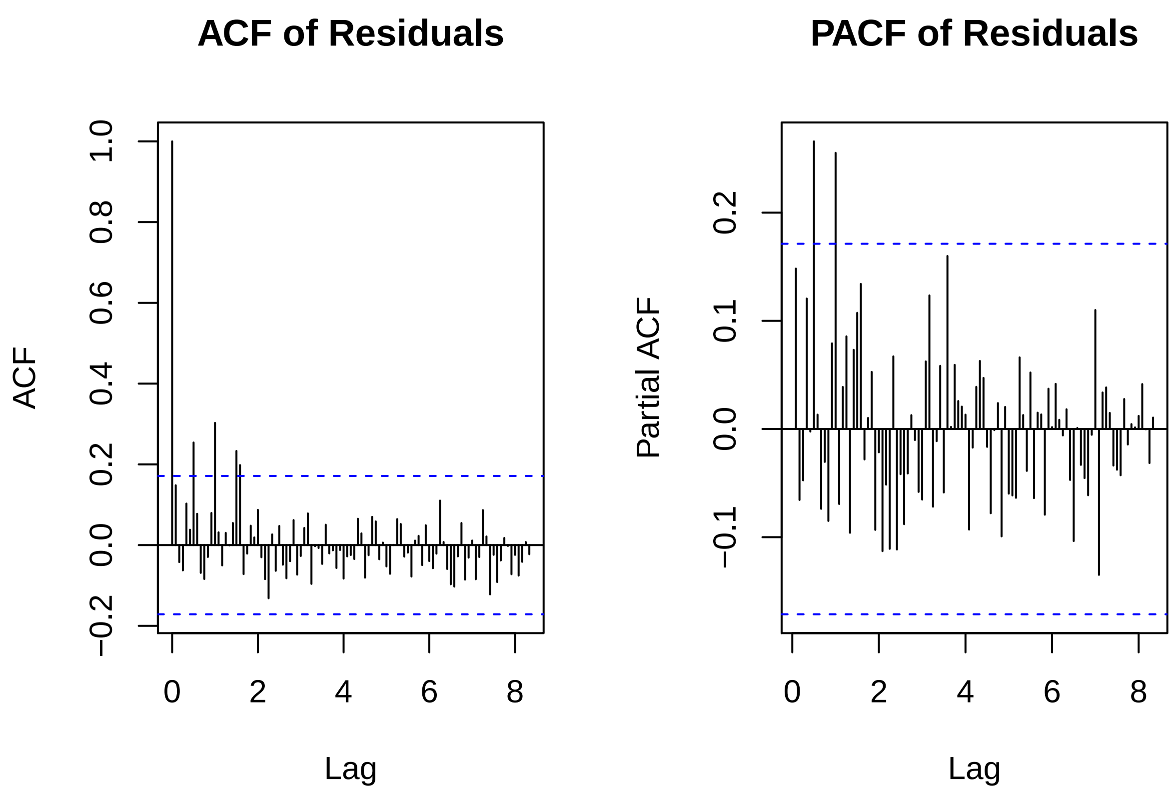
```
##
##  Box-Ljung test
##
## data:  res0120
## X-squared = 17.557, df = 10, p-value = 0.06292
```

```r
Box.test(res0120, lag = h, type = "Ljung", fitdf = 2)
```

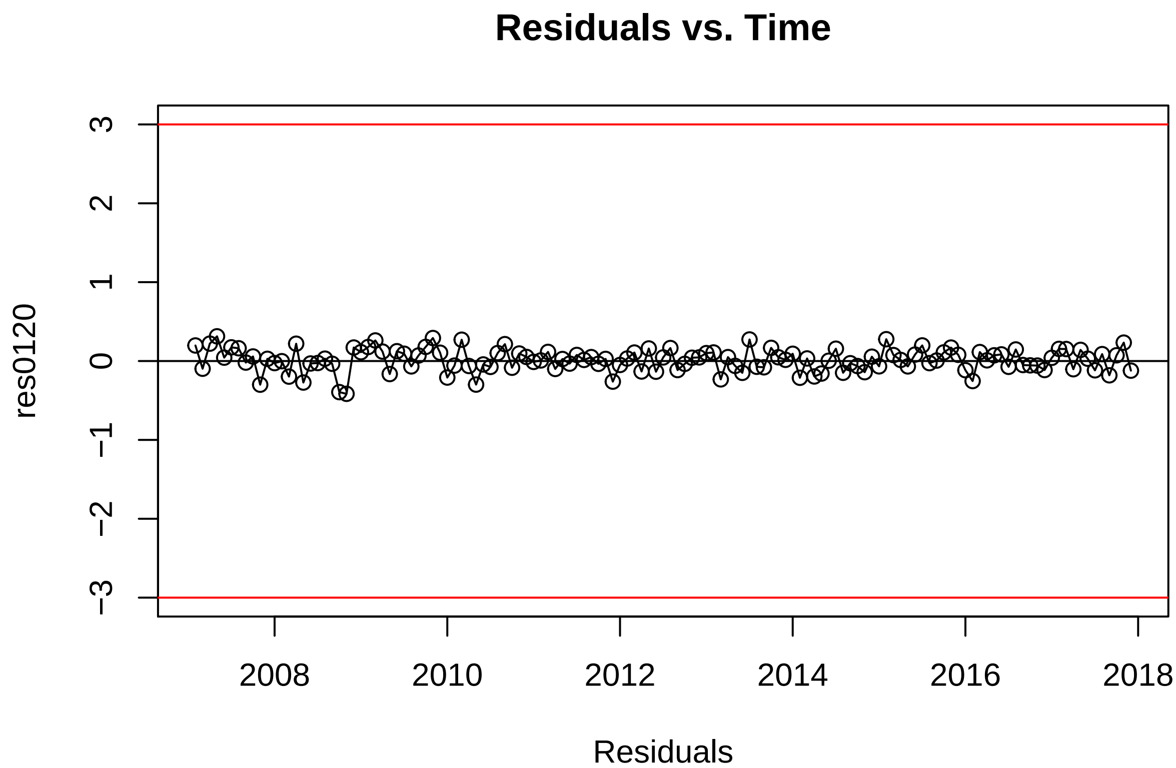
```
##
##  Box-Ljung test
##
## data:  res0120
## X-squared = 51.583, df = 22, p-value = 0.0003568
```

```r
# Passes tests at a significance level alpha = 0.05
```

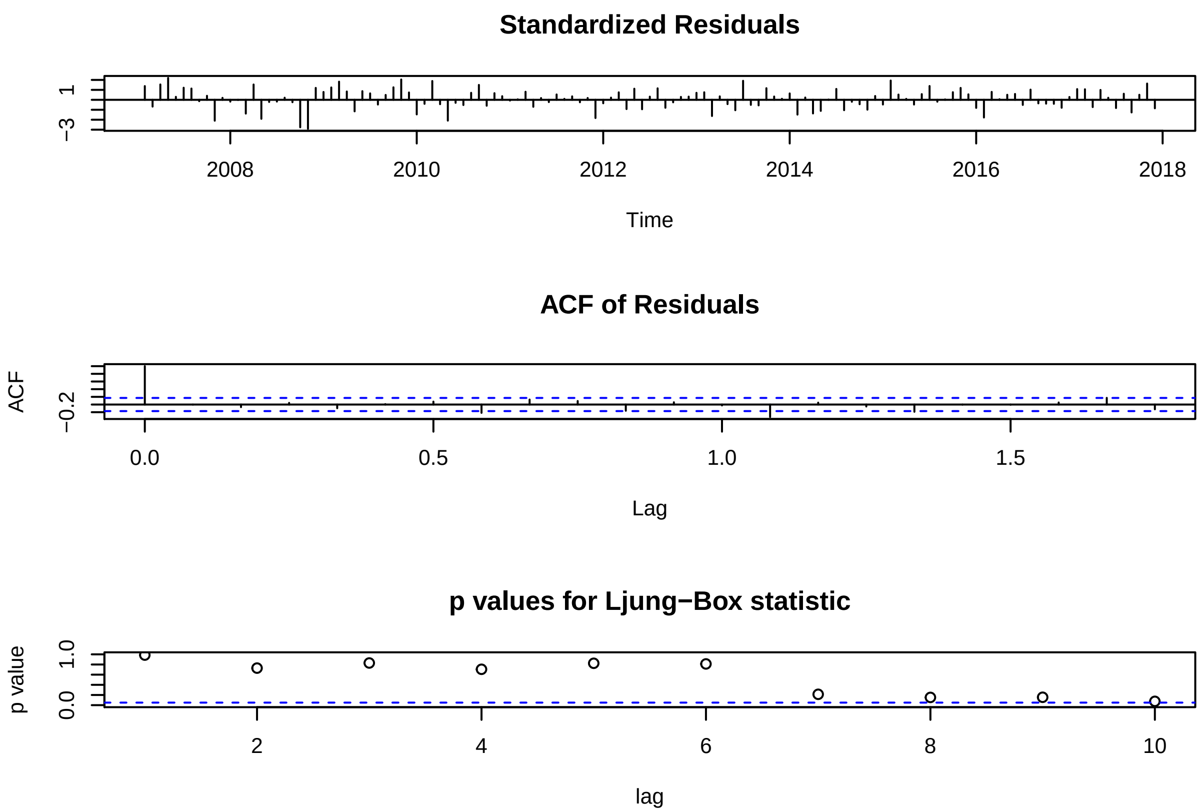
```r
# Detecting Heteroscedasticity, i.e. testing for constant variance

op <- par(mfrow = c(1,2))
acf(res0120^2, lag.max = 100, main = "ACF of Residuals")
pacf(res0120^2, lag.max = 100, main = "PACF of Residuals")
```




```r
par(op)
plot(res0120, ylim = c(-3,3), type = "o", main = "Residuals vs. Time", xlab = "Residuals")
abline(h=3,col="red")
abline(h=-3,col="red")
abline(h=-0)
```




```r
tsdiag(fit.0120)
```



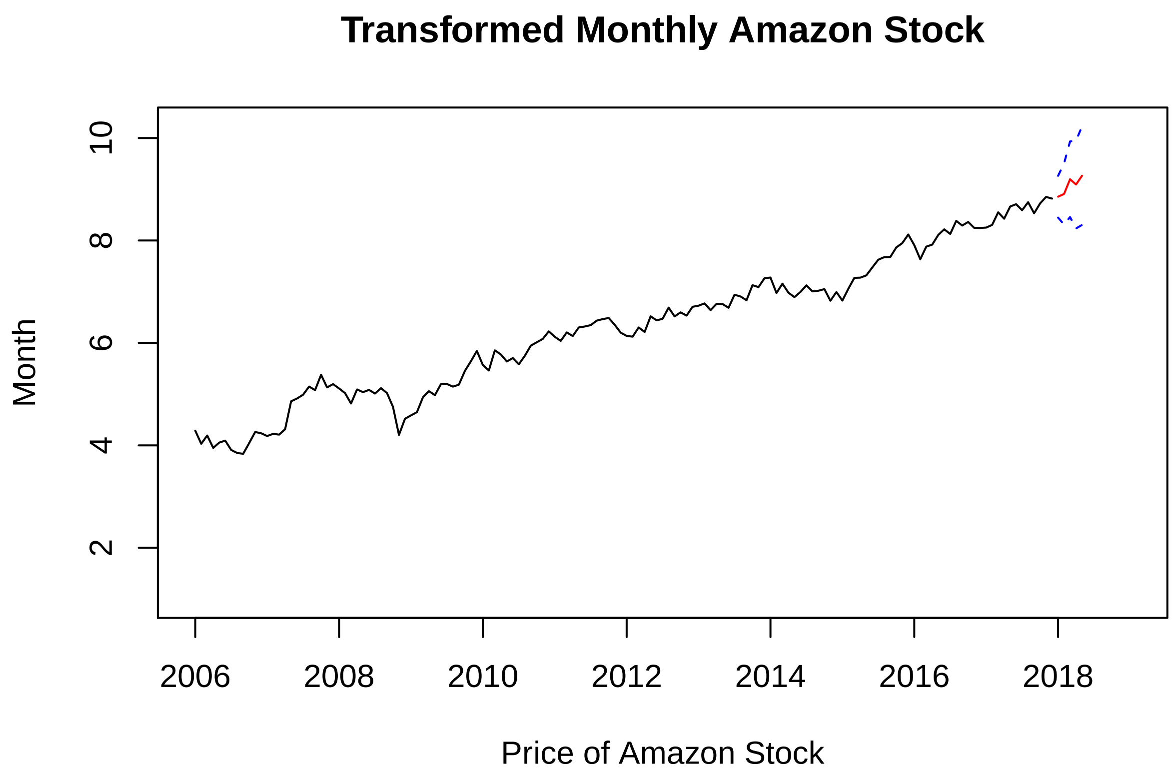

```
# Passes all tests
```

```
#APPENDIX G - FORECASTING

fit <- arima(data.bc,order = c(0,1,1), seasonal = c(order = c(2,1,1), period =8), method = "ML", includ
fit
```

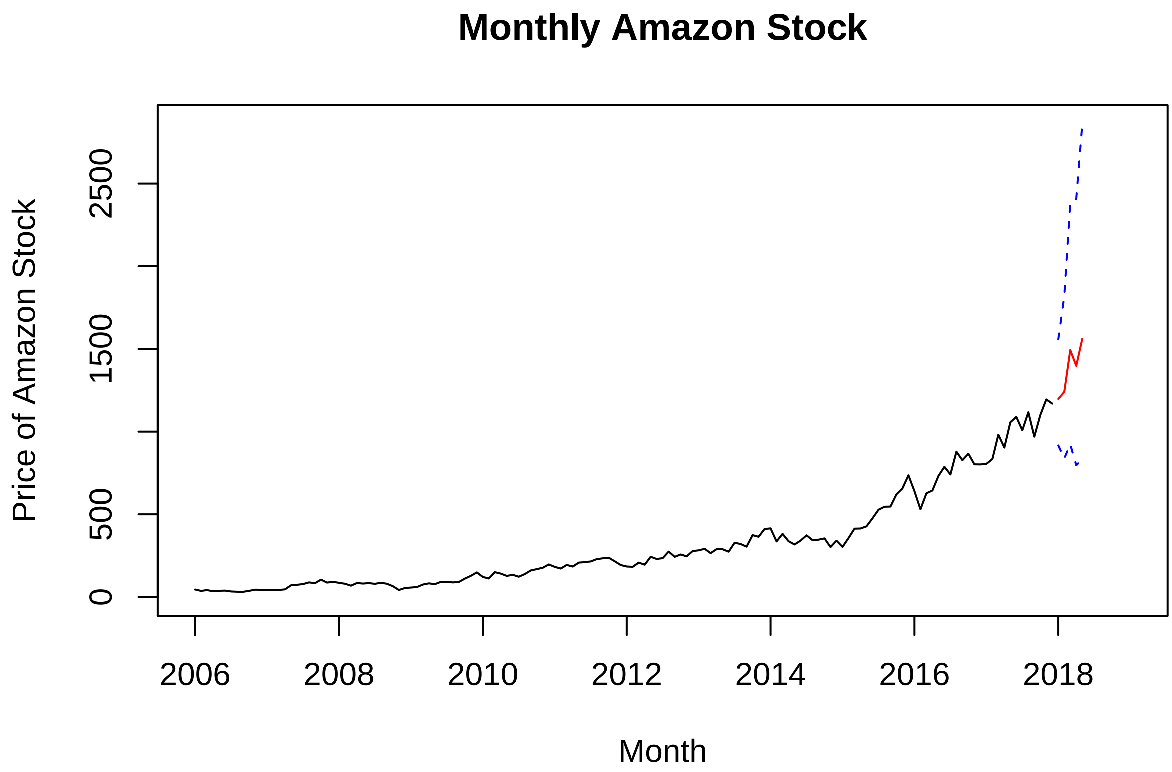
```
##
## Call:
## arima(x = data.bc, order = c(0, 1, 1), seasonal = c(order = c(2, 1, 1), period = 8),
##     include.mean = FALSE, method = "ML")
##
## Coefficients:
##          ma1    sar1     sar2    sma1
##       0.0692  0.0521  -0.2063  0.1180
## s.e.  0.0901  0.7206   0.1498  0.7514
##
## sigma^2 estimated as 0.04322:  log likelihood = 19.39,  aic = -28.79
```

```
#model is: diff1(diff12(Xt)) = (1+0.4599B)(1+0.9210B^12)Zt, sigma^2 = 1.003
#predict transformed data

pred.tfm <- predict(fit, n.ahead = 5)
U.tfm <- pred.tfm$pred + 1.96*pred.tfm$se
L.tfm <- pred.tfm$pred - 1.96*pred.tfm$se

plot.ts(data.bc, ylim = c(1,max(U.tfm)),xlim = c(2006,2019),
         main = "Transformed Monthly Amazon Stock",
```

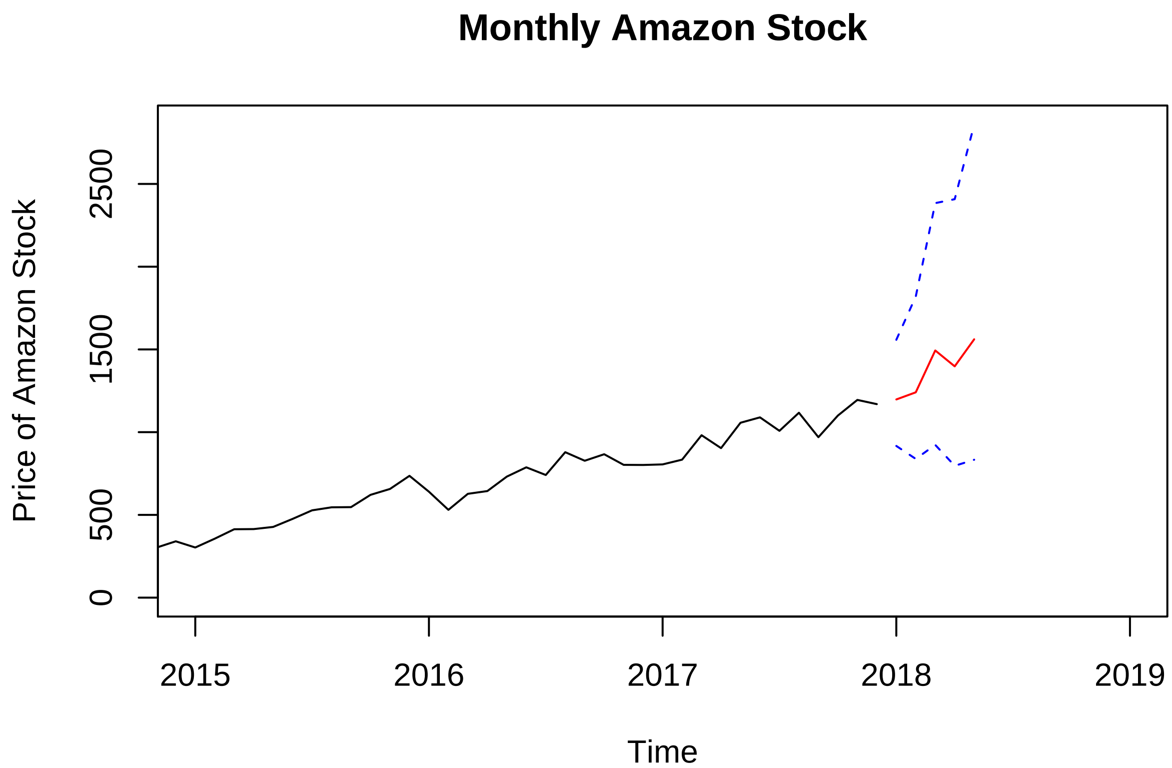
```
        xlab = "Price of Amazon Stock", ylab = "Month")
lines(U.tfm, col = "blue", lty="dashed")
lines(L.tfm, col = "blue", lty="dashed")
points(pred.tfm$pred, col = "red", type = "l")
```




```
#Transform predicted data to match original data
pred <- ((lambda*pred.tfm$pred)+1)^(1/lambda)
U <- ((lambda*U.tfm)+1)^(1/lambda)
L <- ((lambda*L.tfm)+1)^(1/lambda)

ts.plot(TS.AMZN, xlim = c(2006,2019), xlab = "Month", ylab = "Price of Amazon Stock",
        main= "Monthly Amazon Stock", ylim = c(0,max(U)+1))
points(pred, col = "red", type = "l")
lines(U, col = "blue", lty="dashed")
lines(L, col = "blue", lty="dashed")
```



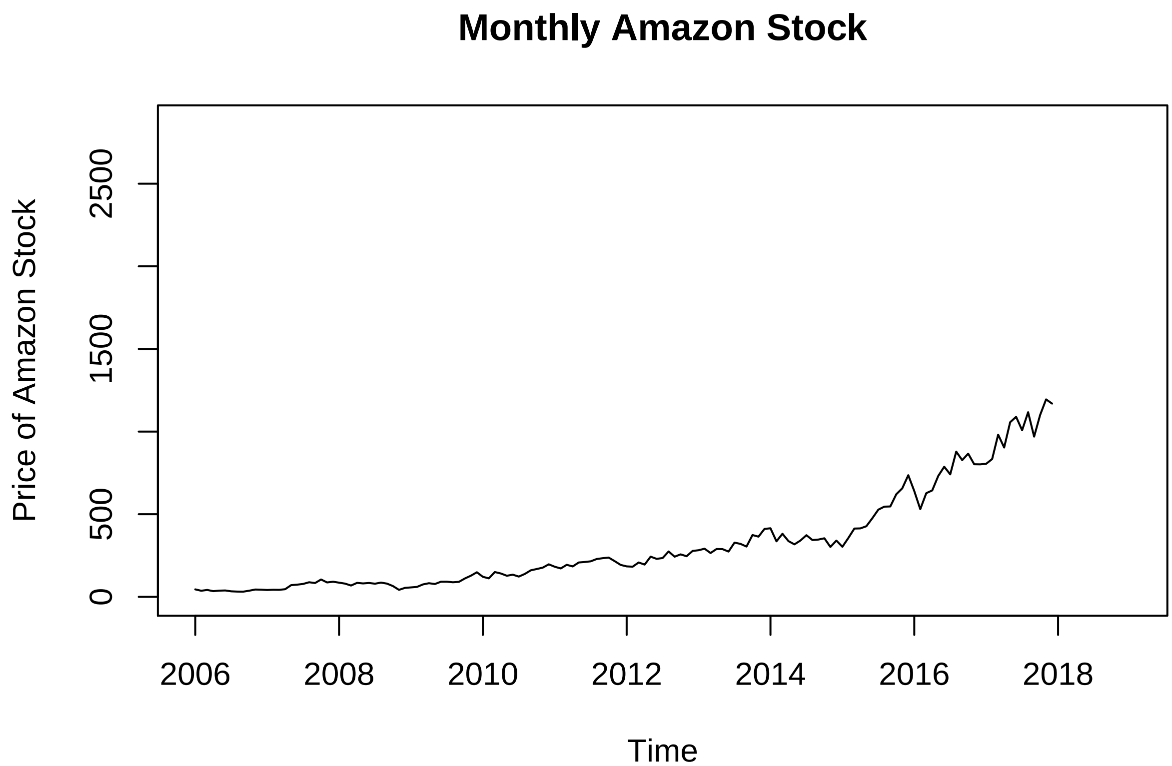

```r
ts.plot(TS.AMZN, xlim = c(2015,2019), xlab = "Time", ylab = "Price of Amazon Stock",
          main= "Monthly Amazon Stock", ylim = c(0,max(U)+1))
points(pred, col = "red", type = "l")
lines(U, col = "blue", lty="dashed")
lines(L, col = "blue", lty="dashed")
```




```
pred
```

```
##            Jan      Feb      Mar      Apr      May
## 2018 1197.650 1240.707 1493.587 1397.761 1561.267
```

```
data.bc2 <- ts(data.bc[1:127],start = c(1966,1),deltat = 1/12)
fit3 <- arima(data.bc2,order = c(0,1,1), seasonal = c(order = c(2,1,1), period =12), method =
"ML", include.mean = FALSE, transform.pars = FALSE,  fixed = c(0.0692,0.0521,-0.2063,0.1180))

pred.tfm2 <- predict(fit3, n.ahead = 15)
U.tfm2 <- pred.tfm2$pred + 1.96*pred.tfm2$se
L.tfm2 <- pred.tfm2$pred - 1.96*pred.tfm2$se

pred2 <- ((lambda*pred.tfm2$pred)+1)^(1/lambda)
U2 <- ((lambda*U.tfm2)+1)^(1/lambda)
L2 <- ((lambda*L.tfm2)+1)^(1/lambda)

ts.plot(TS.AMZN, xlim = c(2006,2019), xlab = "Time", ylab = "Price of Amazon Stock",
main= "Monthly Amazon Stock", ylim = c(0,max(U)+1))
points(pred2, col = "red", type = "l")
lines(U2, col = "blue", lty="dashed")
lines(L2, col = "blue", lty="dashed")
```



Monthly Amazon Stock
Price of Amazon Stock
0
500
1500
2500
2006
2008
2010
2012
2014
2016
2018
Time