Bitcoin Time Series

Group Kappa

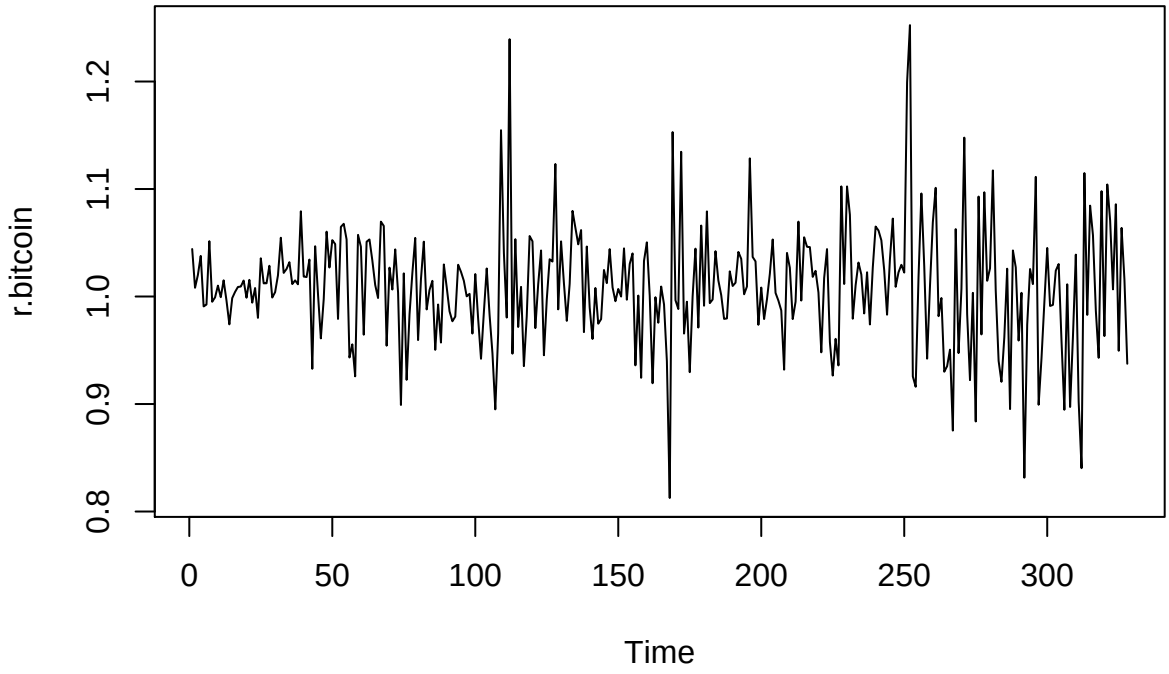
3/8/2018

Abstract

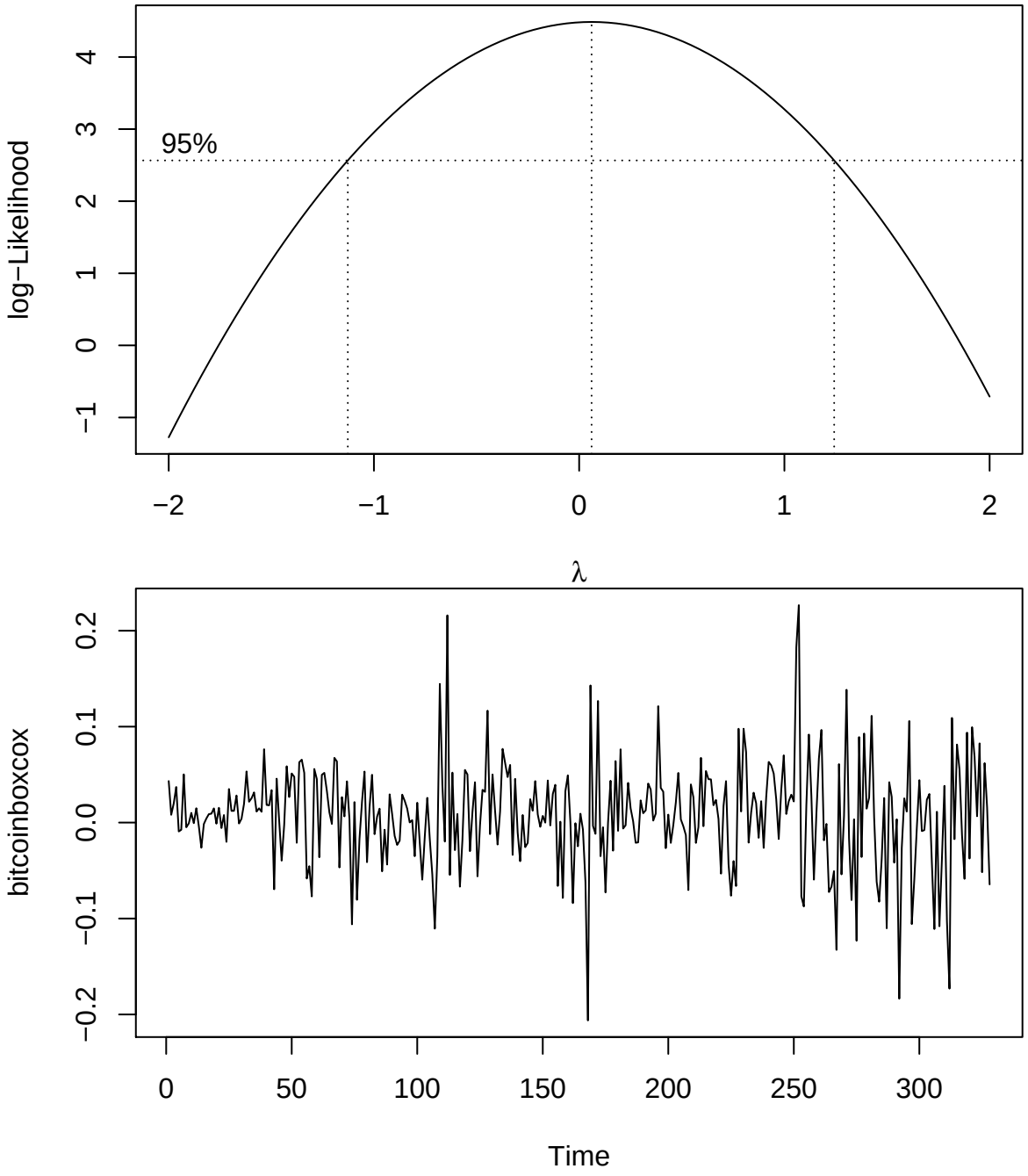
Introduction

Analysis

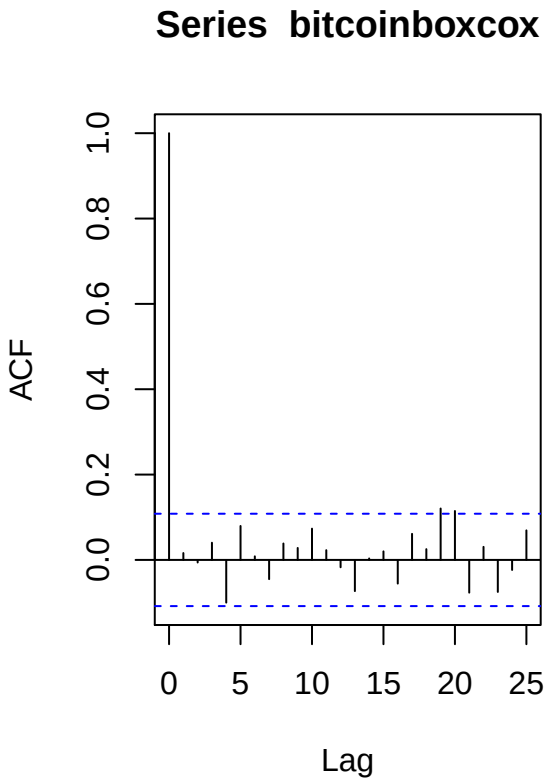
Exploratory Data Analysis

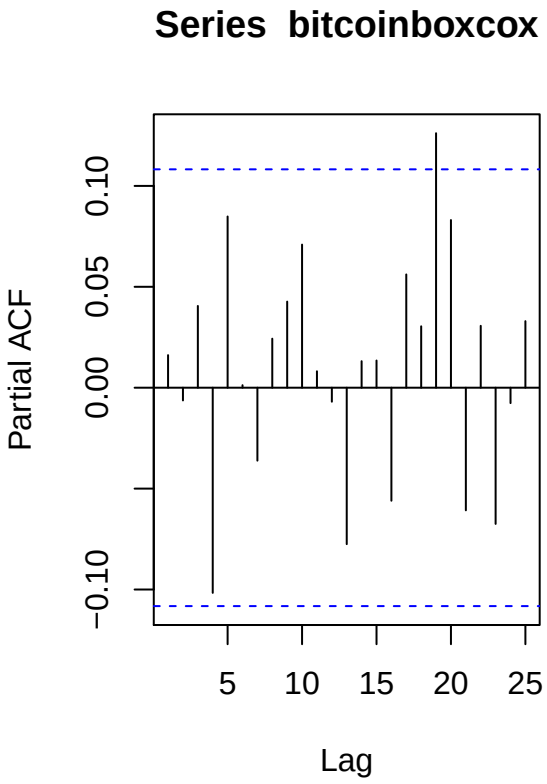


Data Transformations



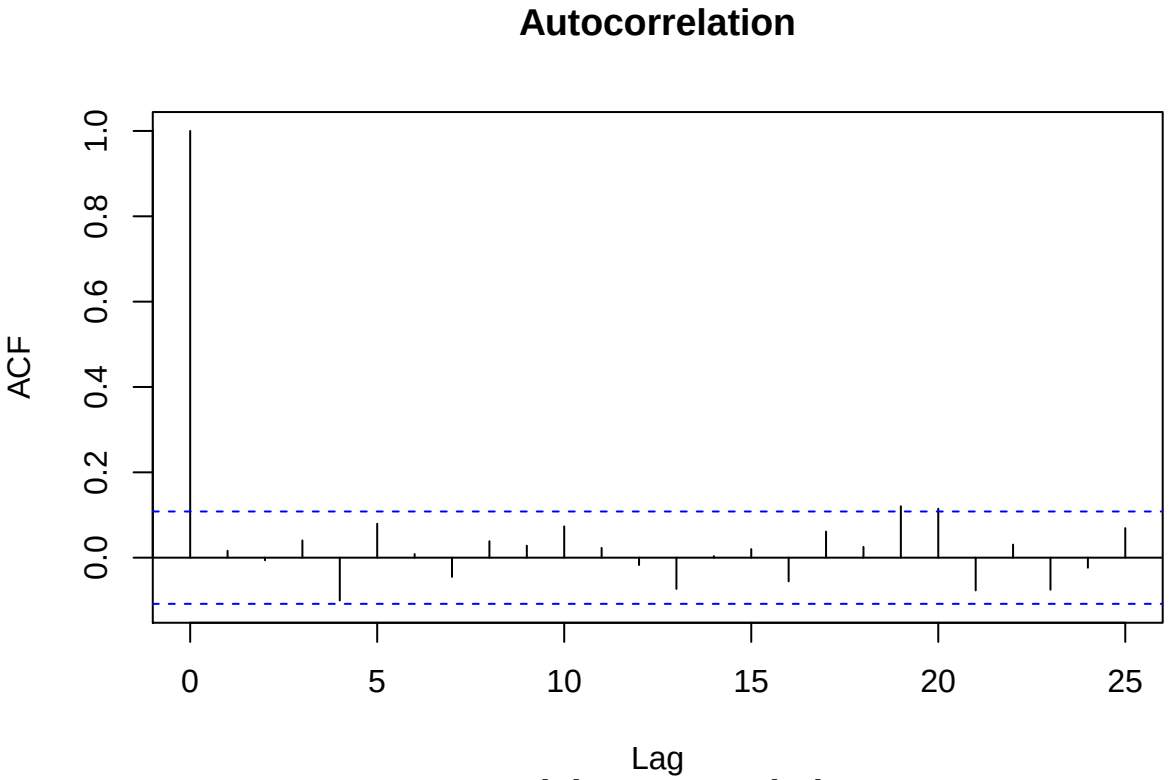
ACF and PACF Analysis

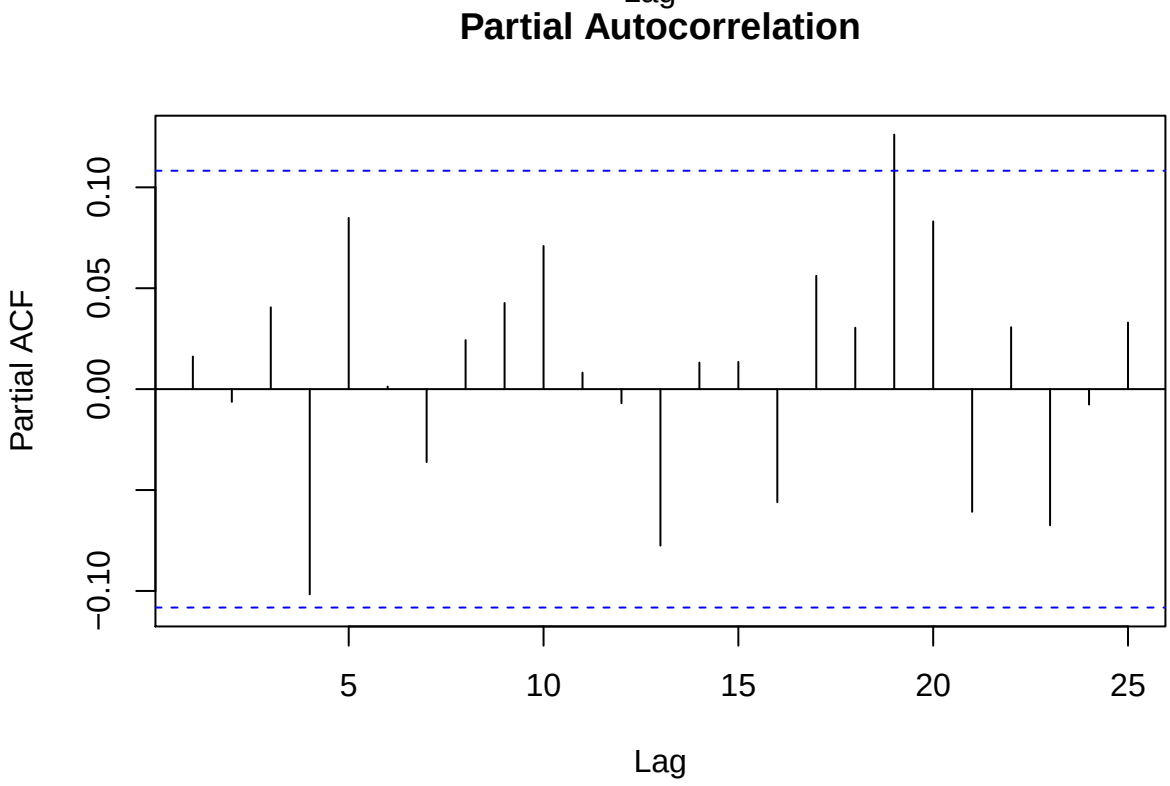




Model Diagnostics

White Noise Model (Model 1)

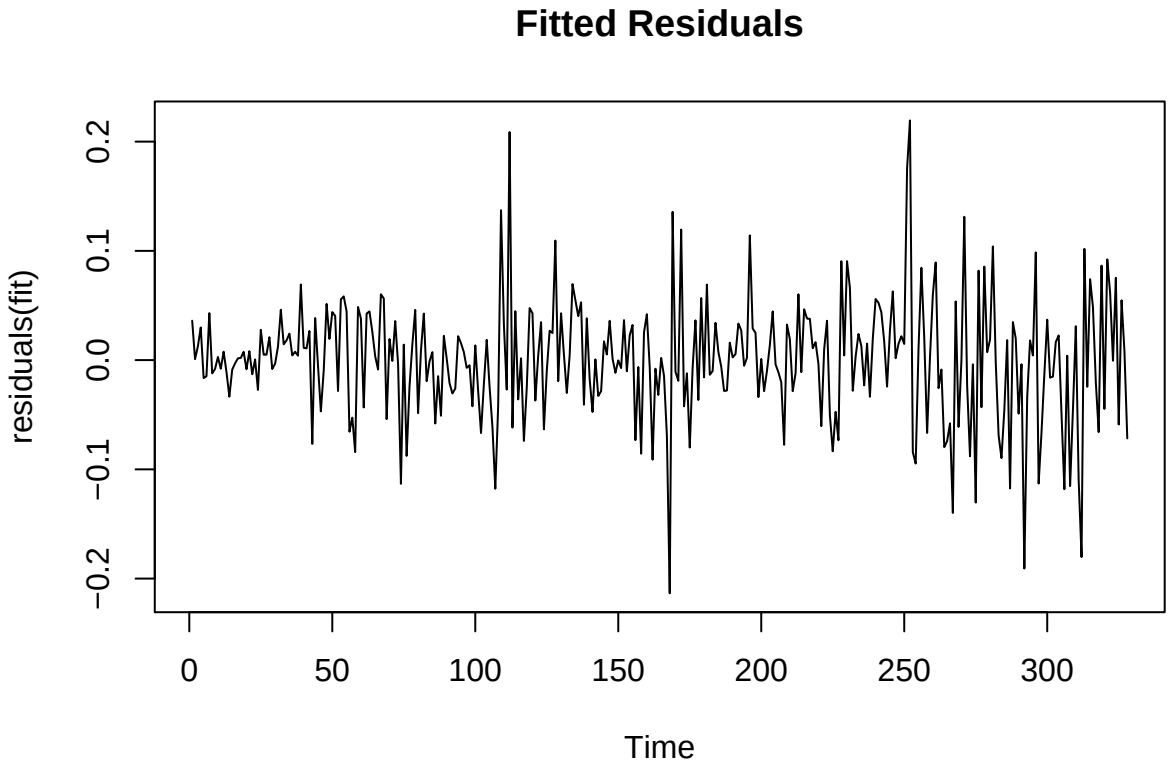


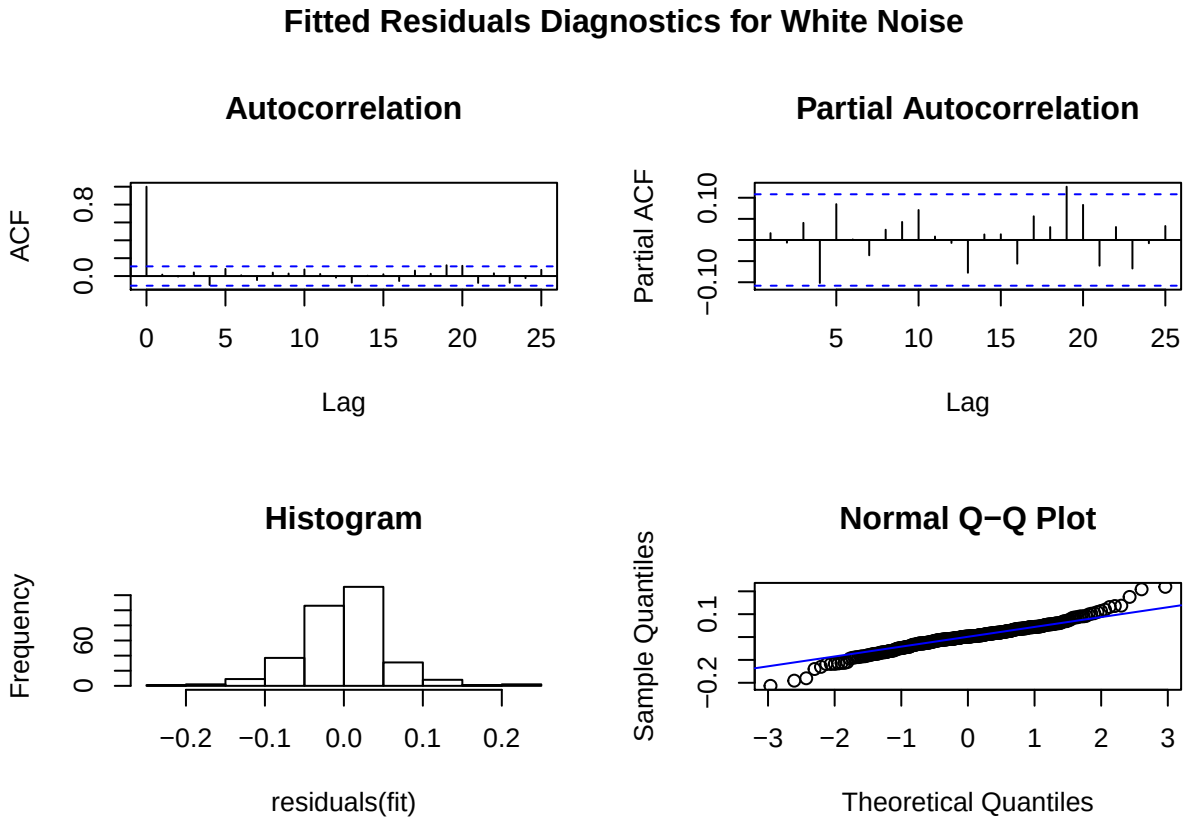


##

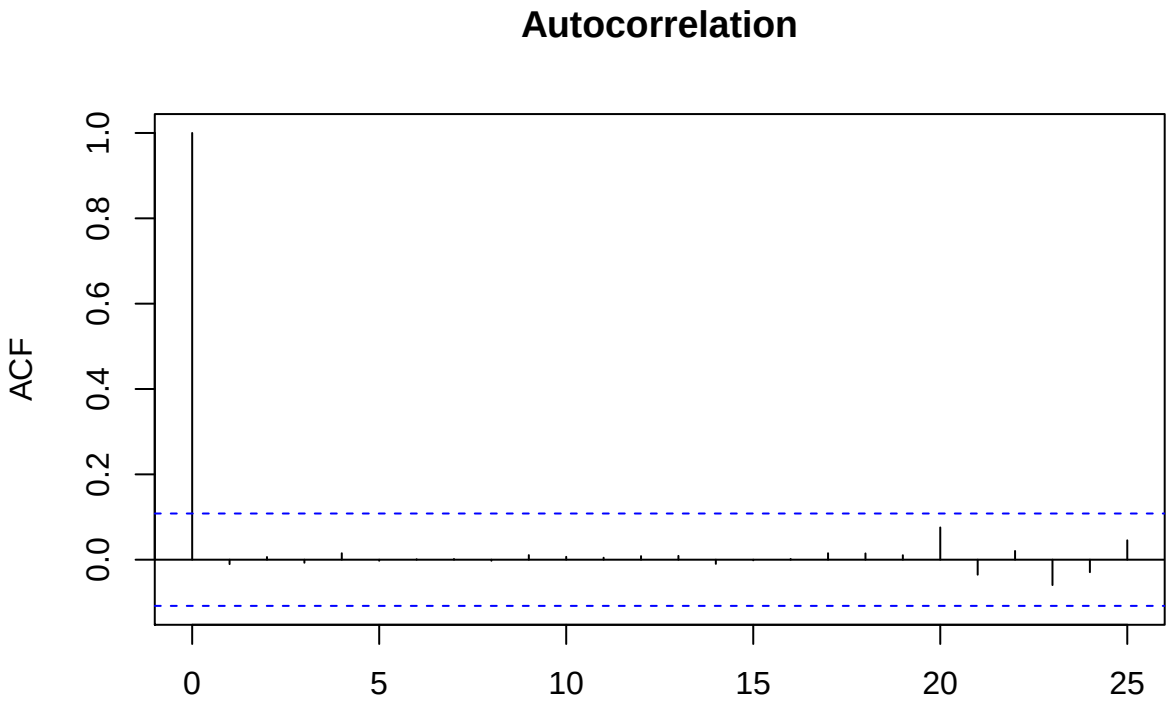
```
##  Box-Ljung test
##
## data:  resid(fit)
## X-squared = 0.086284, df = 1, p-value = 0.769
```

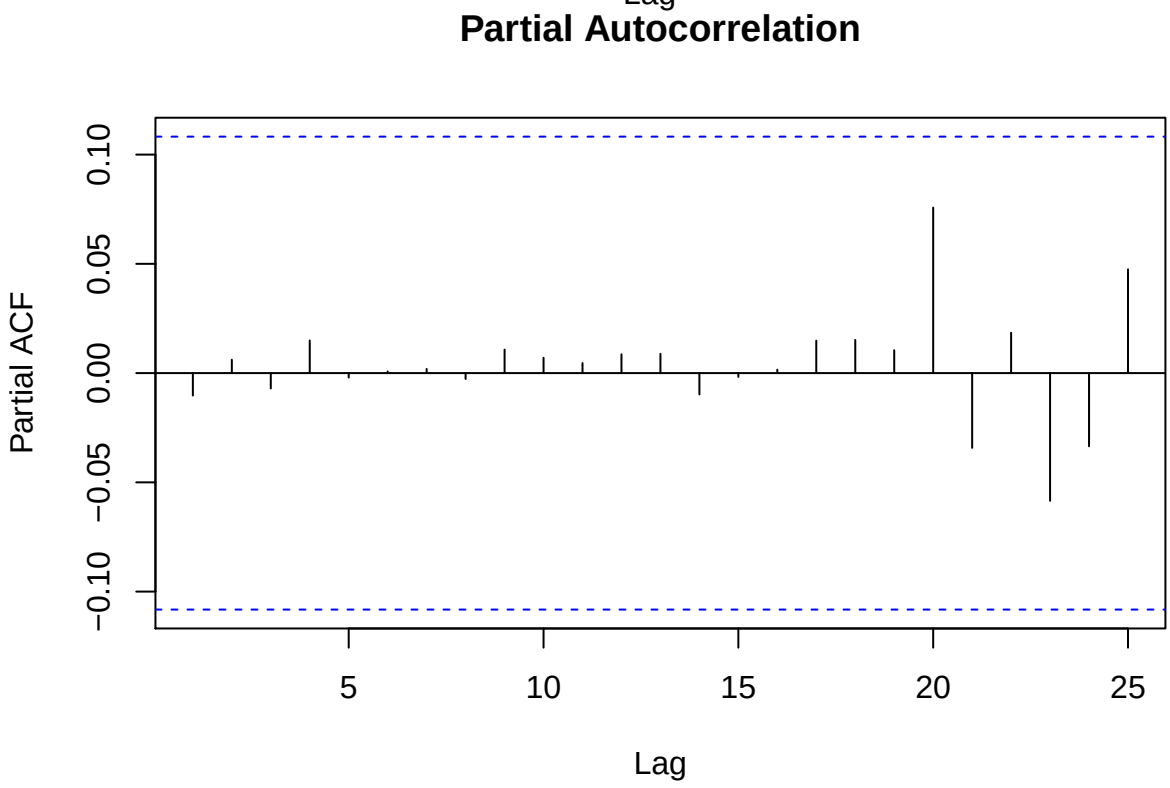
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit)
## W = 0.96744, p-value = 1.004e-06
```




Fitted Residuals Diagnostics for White Noise
Autocorrelation
ACF
0.0
0.8
0
5
10
15
20
25
Lag
Partial Autocorrelation
Partial ACF
−0.10
0.10
5
10
15
20
25
Lag
Histogram
Frequency
0
60
−0.2
−0.1
0.0
0.1
0.2
residuals(fit)
Normal Q−Q Plot
Sample Quantiles
−0.2
0.1
−3
−2
−1
0
1
2
3
Theoretical Quantiles

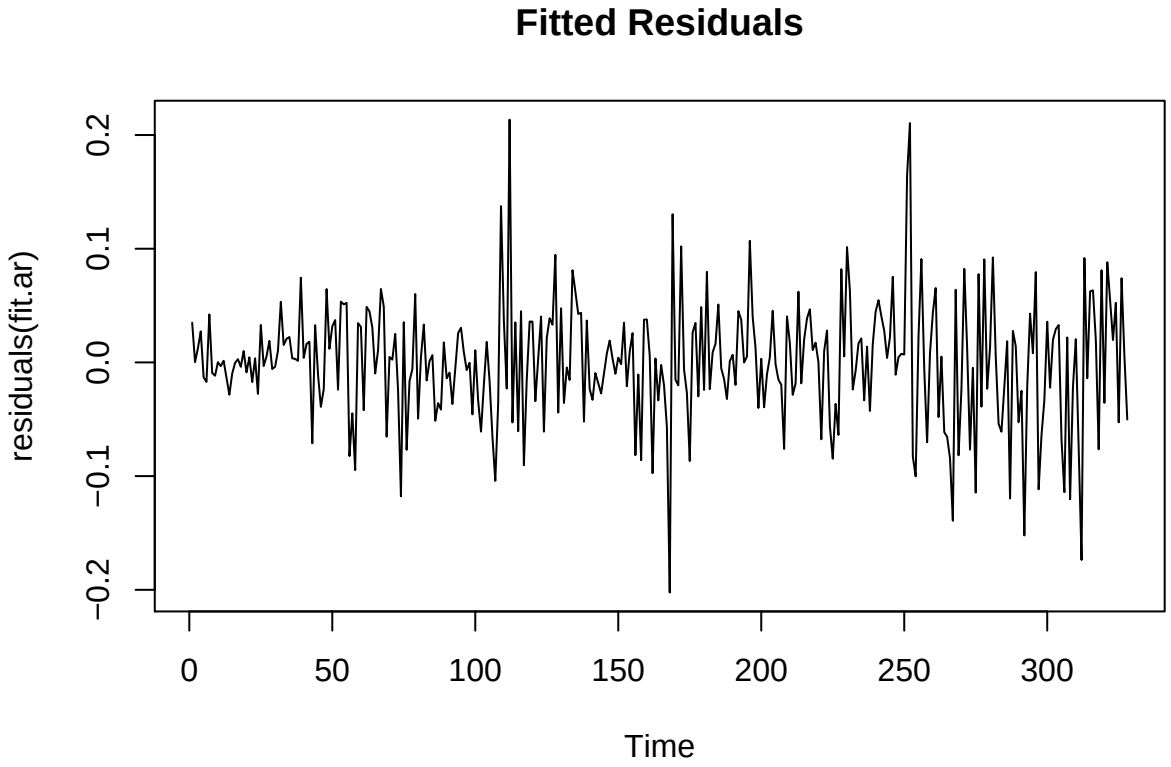
AR(19) (Model 2)

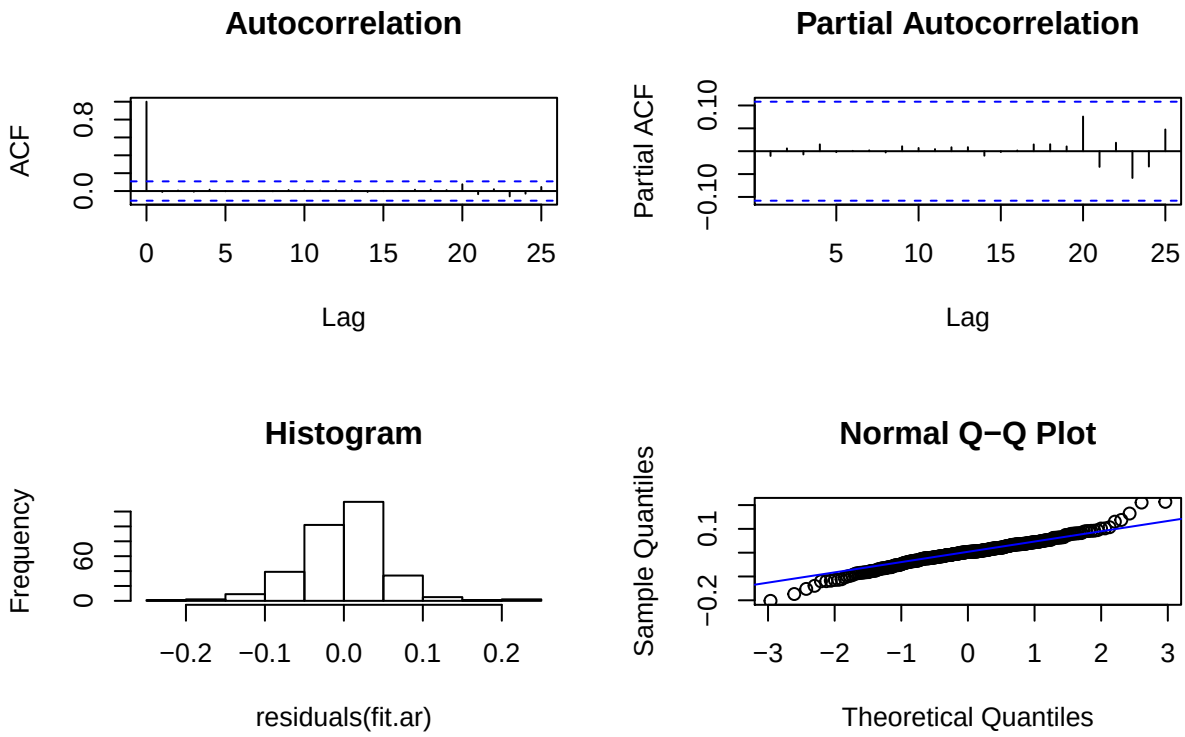




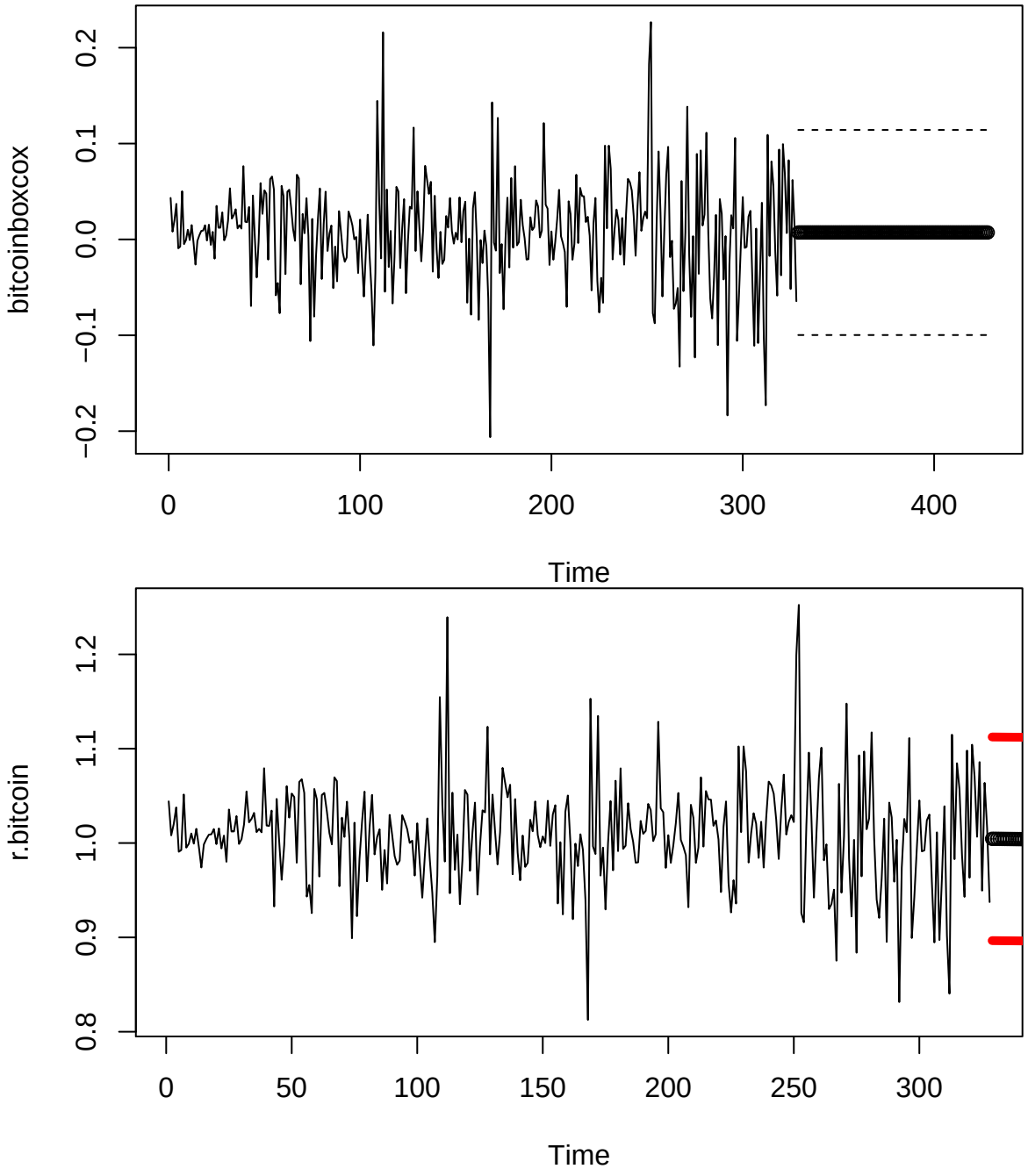

```
##
##  Box-Ljung test
##
```

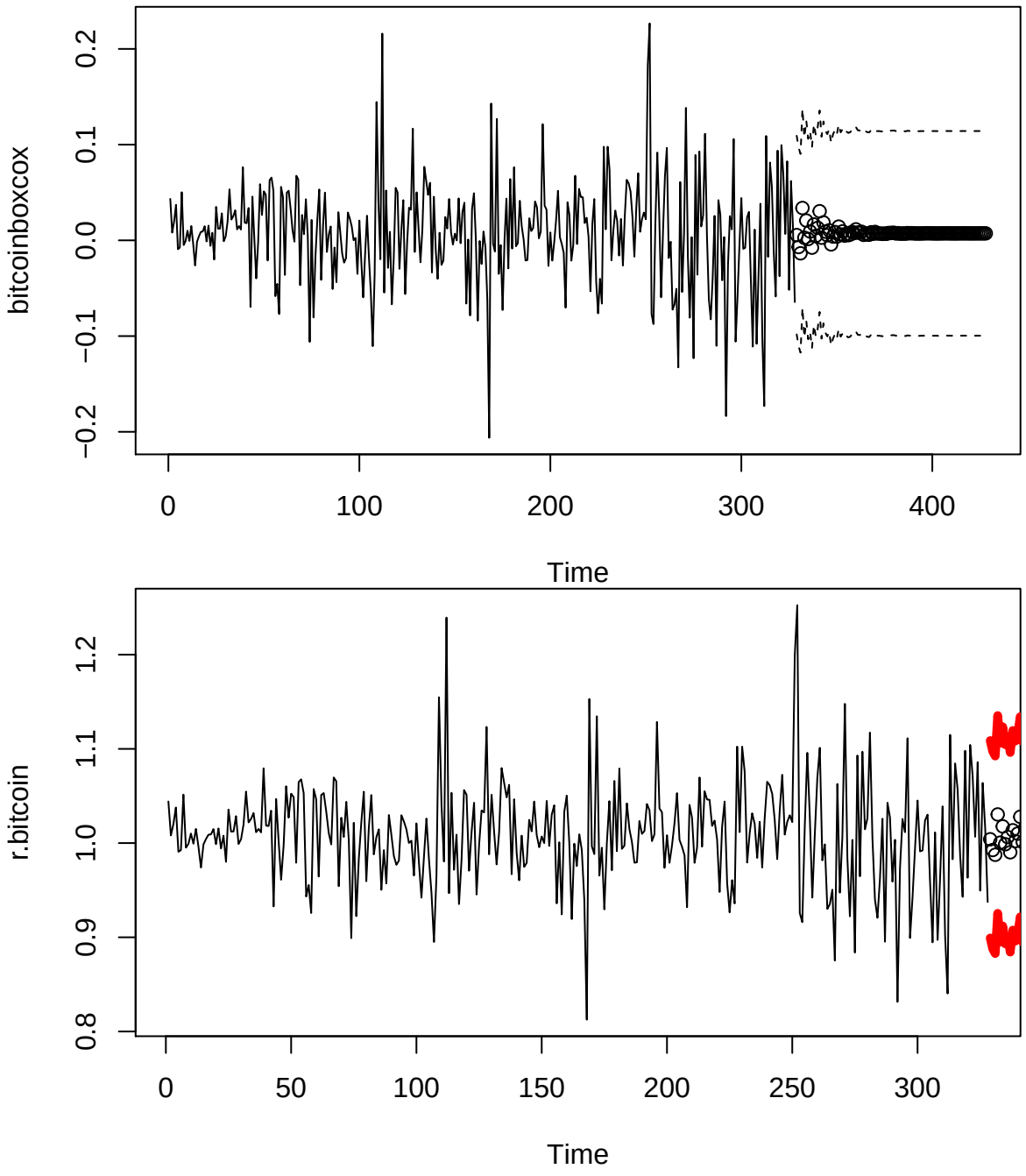
```
## data:  resid(fit.ar)
## X-squared = 0.035084, df = 1, p-value = 0.8514

##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit.ar)
## W = 0.97412, p-value = 1.271e-05
```




Fitted Residuals Diagnostics For AR
Autocorrelation
ACF
0.0
0.8
0
5
10
15
20
25
Lag
Partial Autocorrelation
Partial ACF
−0.10
0.10
5
10
15
20
25
Lag
Histogram
Frequency
0
60
−0.2
−0.1
0.0
0.1
0.2
residuals(fit.ar)
Normal Q−Q Plot
Sample Quantiles
−0.2
0.1
−3
−2
−1
0
1
2
3
Theoretical Quantiles

Forecasting





Conclusion

References

*https://www.kaggle.com/jessevent/all-crypto-currencies/data

*RStudio

*PSTAT 174 Winter 2018 Lecture Slides and Labs

Appendix

```
library(tidyverse)
library(ggplot2)
library(MASS)
library(qpcR)
library(forecast)
#For macs download https://www.xquartz.org/
library(tseries)

##Setup

#Reading in data
data <- read.csv("crypto.csv", header=TRUE)

#Choosing only Bitcoin
newdata <- subset(data, symbol=="BTC")

#Choosing the two variables
myvars <- c("date","close")
crypto <- newdata[myvars]

#Changing class of "date"
crypto$date <- as.Date(crypto$date)

#Cutting off dates before 4/1/17
bitcoin <- subset(crypto, date > "2017-03-29")

#Creating the return variable
r.bitcoin = (bitcoin[2:nrow(bitcoin),2] / bitcoin[1:(nrow(bitcoin)-1),2]-1) + 1

#Creating the return TS
mydata <- bitcoin[-c(1),]
r.bitcoinTS = data.frame(mydata[1],r.bitcoin)

#BoxCox Transform
time <- 1:length(r.bitcoin)
fit <- lm(r.bitcoin ~ time)
boxcoxtransform <- boxcox(r.bitcoin ~ time, plotit = T)
lamb <- boxcoxtransform$x[which(boxcoxtransform$y == max(boxcoxtransform$y))]
bitcoinboxcox <- (1/lamb)*(r.bitcoin^lamb-1)
ts.plot(bitcoinboxcox)

#BoxCox Transform for Dataset
time <- 1:length(r.bitcoinTS[,2])
fit <- lm(r.bitcoinTS[,2] ~ time)
boxcoxtransform <- boxcox(r.bitcoinTS[,2] ~ time, plotit = T)
lamb <- boxcoxtransform$x[which(boxcoxtransform$y == max(boxcoxtransform$y))]
bitcoinboxcoxTS <- (1/lamb)*(r.bitcoinTS[,2]^lamb-1)
ts.plot(bitcoinboxcoxTS)

#PLotting ACF/PACF
op <- par(mfrow=c(2,1))
```

```r
acf(bitcoinboxcox)
pacf(bitcoinboxcox)
par(op)

##Diagnostics

#MA(1) Model
fit = arima(bitcoinboxcox, order=c(0,0,0), method="ML")

#Testing for independence of residuals
Box.test(resid(fit), type="Ljung")

#Test for normality of residuals
shapiro.test(residuals(fit))

#Plotting Residuals of Fit
ts.plot(residuals(fit),main = "Fitted Residuals")


par(mfrow=c(1,2),oma=c(0,0,2,0))
# Plot diagnostics of residuals
op <- par(mfrow=c(2,2))
# acf
acf(residuals(fit),main = "Autocorrelation")
# pacf
pacf(residuals(fit),main = "Partial Autocorrelation")
# Histogram
hist(residuals(fit),main = "Histogram")
# q-q plot
qqnorm(residuals(fit))
qqline(residuals(fit),col ="blue")
# Add overall title
title("Fitted Residuals Diagnostics", outer=TRUE)
par(op)

##Forecasting on bitcoinboxcox dataset
mypred = predict(fit, n.ahead=10)
ts.plot(bitcoinboxcox, xlim=c(0,339))
points(x = 329:338, y = mypred$pred)
lines(329:338,mypred$pred+1.96*mypred$se,lty=2)
lines(329:338,mypred$pred-1.96*mypred$se,lty=2)


##Forecasting on bitcoinboxcox dataset with 100
mypred = predict(fit, n.ahead=100)
ts.plot(bitcoinboxcox, xlim=c(0,429))
points(x = 329:428, y = mypred$pred)
lines(329:428,mypred$pred+1.96*mypred$se,lty=2)
lines(329:428,mypred$pred-1.96*mypred$se,lty=2)

#Forecasting on original
unboxcoxed<-((bitcoinboxcox*lamb)+1)^(1/lamb)
fitma1unboxcoxed<-arima(unboxcoxed,order=c(0,0,0),method="ML",xreg=1:length(unboxcoxed))
```

```
predtrans1<-predict(fitma1unboxcoxed,n.ahead=100,newxreg=(length(unboxcoxed)+1)
                    :(length(unboxcoxed)+100))
ltrans1<-predtrans1$pred-1.96*predtrans1$se
ltrans1

utrans1<-predtrans1$pred+1.96*predtrans1$se
utrans1

ts.plot(r.bitcoin)
points(x = 329:428,predtrans1$pred,pch=1)
lines(329:428,ltrans1,col="red",lwd=5)
lines(329:428,utrans1,col="red",lwd=5)


#AR(18) Model
fit.ar = arima(bitcoinboxcox, order=c(19,0,0), method="ML")

par(mfrow=c(1,2),oma=c(0,0,2,0))
# Plot diagnostics of residuals
op <- par(mfrow=c(2,2))
# acf
acf(residuals(fit.ar),main = "Autocorrelation")
# pacf
pacf(residuals(fit.ar),main = "Partial Autocorrelation")
# Histogram
hist(residuals(fit.ar),main = "Histogram")
# q-q plot
qqnorm(residuals(fit.ar))
qqline(residuals(fit.ar),col ="blue")
# Add overall title
title("Fitted Residuals Diagnostics For AR", outer=TRUE)
par(op)

#Testing for independence of residuals
Box.test(resid(fit.ar), type="Ljung")

#Test for normality of residuals
shapiro.test(residuals(fit.ar))

##Forecasting AR Model on bitcoinboxcox dataset with 100
mypred = predict(fit.ar, n.ahead=100)
ts.plot(bitcoinboxcox, xlim=c(0,429))
points(x = 329:428, y = mypred$pred)
lines(329:428,mypred$pred+1.96*mypred$se,lty=2)
lines(329:428,mypred$pred-1.96*mypred$se,lty=2)

#Forecasting on original for AR
unboxcoxed<-((bitcoinboxcox*lamb)+1)^(1/lamb)
fitma1unboxcoxed<-arima(unboxcoxed,order=c(18,0,0),method="ML",xreg=1:length(unboxcoxed))
predtrans1<-predict(fitma1unboxcoxed,n.ahead=100,newxreg=(length(unboxcoxed)+1)
                    :(length(unboxcoxed)+100))
ltrans1<-predtrans1$pred-1.96*predtrans1$se
ltrans1
```

```
utrans1<-predtrans1$pred+1.96*predtrans1$se
utrans1

ts.plot(r.bitcoin)
points(x = 329:428,predtrans1$pred,pch=1)
lines(329:428,ltrans1,col="red",lwd=5)
lines(329:428,utrans1,col="red",lwd=5)
```