Lab 7

PSTAT W 174/274

We will continue analyzing the monthly Boston armed robberies from last lab. But this time, we will focus on the last two stages: diagnostics and forecasting.

Model Diagnostics

Summary of R commands for estimation:

1. Check normality assumptions:

```
#plot the histogram of the residuals:
hist(residuals)
#q-q plot
qqnorm(residuals)
qqline(residuals)
```

2. Portmanteau Statistics:

```
#Shapiro test for normality
shapiro.test(residuals)
#Box-Pierce test
Box.test(residuals, type = c("Box-Pierce"), lag = h, fitdf)
#Ljung-Box test
Box.test(residuals, type = c("Ljung-Box"), lag = h, fitdf)
#McLeod-Li test
Box.test(residuals^2, type = c("Ljung-Box"), lag = h, fitdf)
```

   For the choice of fitdf and lag, please see Lecture 11 slide 29.

3. Check if the residuals are AR(0):

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

4. Check ACF and PACF

```
par(mfrow=c(2,1))
acf(residuals, lag.max=40,main="")
title("ACF of the residuals")
pacf(residuals, lag.max=40,main="")
title("PACF of the residuals")
```

Model Forecasting.

1. Forecast n observations ahead:

```
library(forecast)
pred.tr <- predict(model, n.ahead = n)
```

2. Get the 95% CI for the prediction:

```
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
```

Boston armed robberies data continued

```
#To see the details, please see lab 6:
library(tsdl)
crime_data = tsdl[[484]]
#train test split
train_data = crime_data[c(1: 108)]
test_data = crime_data[c(109:118)]
#box-cox
crime.bc = train_data^0.3
#remove trend
crime.log_11 <- diff(crime.bc, lag=1)
#fit the model
#model B:
fit4 = arima(crime.bc, order=c(0,1,1), seasonal = list(order = c(3,0,0)
                        , period = 12) ,fixed = c(NA, NA, 0, NA), method="ML")
```

From lab 6, we end up with our model to be model B:

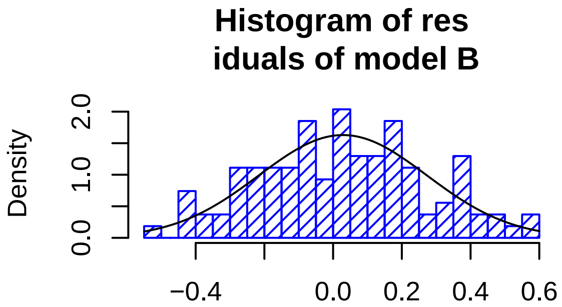
$$(1 - 0.2112B^{12} - 0.2858B^{36})(1 - B)X_t = (1 - 0.4427B)Z_t$$

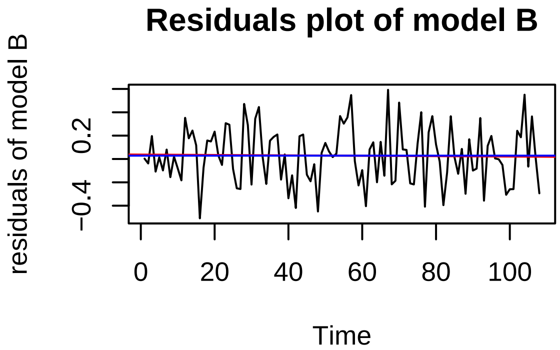
and $X_t = U_t^{1/3}$ where U_t was the original data.

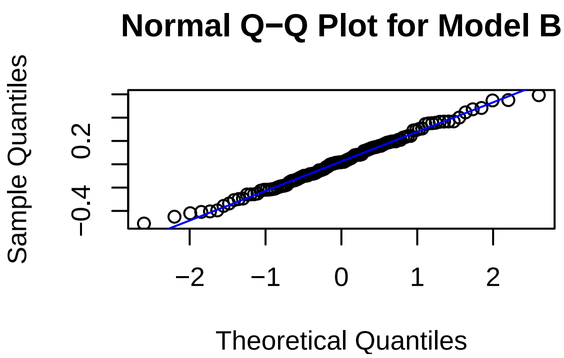
9. Diagnostics Checking

To check if the residuals of model B follow White Noise distribution, we perform several diagnostic tools in this section. We first check normality assumptions and get the plots below.

```
res4 = residuals(fit4)
par(mfrow=c(2,2))
hist(res4,density=20,breaks=20, col="blue", xlab="", prob=TRUE,main="Histogram of res
iduals of model B")
m <- mean(res4)
std <- sqrt(var(res4))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res4,ylab= "residuals of model B",main="Residuals plot of model B")
fitt <- lm(res4 ~ as.numeric(1:length(res4)))
abline(fitt, col="red")
abline(h=mean(res4), col="blue")
qqnorm(res4,main= "Normal Q-Q Plot for Model B")
qqline(res4,col="blue")
```







We can observe that it roughly follow a normal distribution from the histogram and q-q plot. Also, there is no trend or obvious seasonality from the time series plot of the residuals.

Then, we also check for several independence assumptions by Portmanteau Statistics.

```
#Shapiro test for normality
shapiro.test(res4)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res4
## W = 0.99087, p-value = 0.6868
```

```
#Box-Pierce test
Box.test(res4, lag = 10, type = c("Box-Pierce"), fitdf = 3)
```

```
##
##  Box-Pierce test
##
## data:  res4
## X-squared = 10.894, df = 7, p-value = 0.1433
```

```
#Ljung-Box test
Box.test(res4, lag = 10, type = c("Ljung-Box"), fitdf = 3)
```

```
##
##  Box-Ljung test
##
## data:  res4
## X-squared = 11.779, df = 7, p-value = 0.1081
```

```r
#McLeod-Li test
Box.test(res4^2, lag = 10, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  res4^2
## X-squared = 5.8101, df = 10, p-value = 0.831
```

lag: Here, we choose lag = 10 because we have n=108 observations and the square root is about 10.

fitdf: Also, Model 4 has 3 estimated coefficients. Therefore, fitdf = 3.

We can see that the p-values are larger than 0.05 for all the tests. Therefore, model B passes all tests.

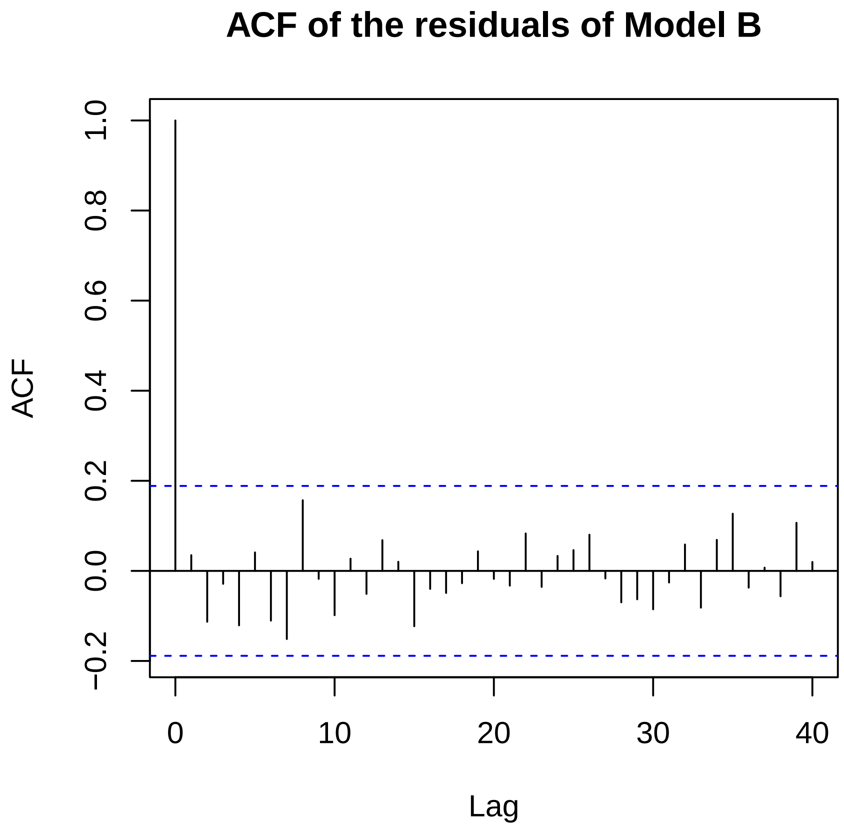
Next, we can see if ar() function will recommend order 0 result for the residuals, which means that the residuals are WN.

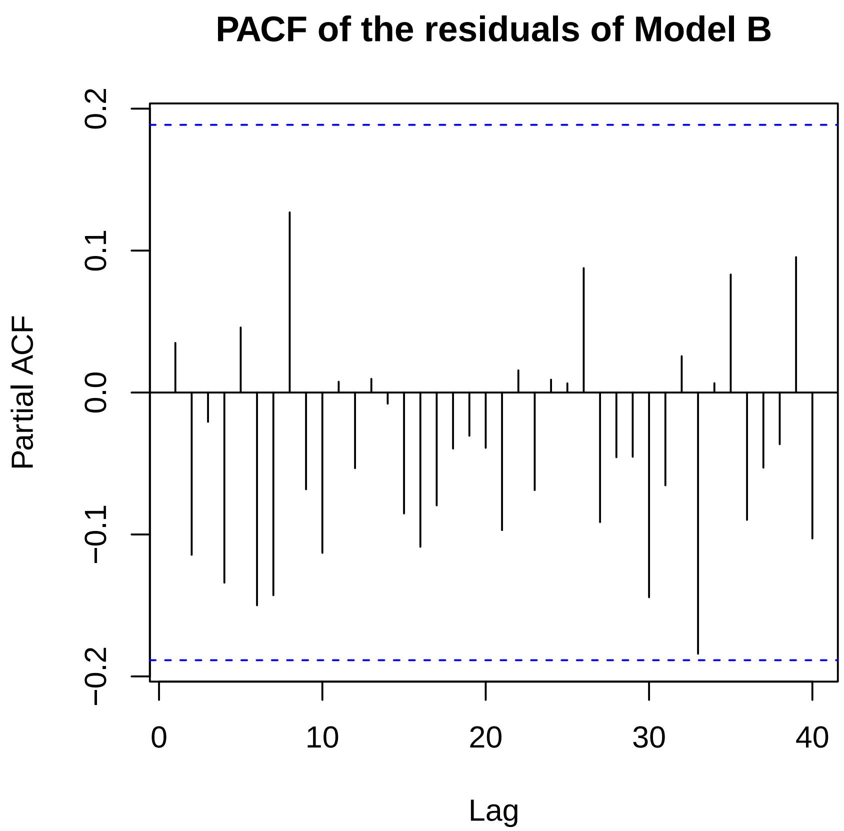
```r
ar(res4, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
##
## Call:
## ar(x = res4, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.05995
```

Lastly, we create ACF and PACF of the residuals. ACF and PACF values at all lags are within the confidence interval. So the residuals can be seen as White Noise.

```r
par(mfrow=c(2,1))
acf(res4, lag.max=40,main="")
title("ACF of the residuals of Model B")
pacf(res4, lag.max=40,main="")
title("PACF of the residuals of Model B")
```


ACF of the residuals of Model B
ACF
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
10
20
30
40
Lag


PACF of the residuals of Model B
Partial ACF
0.2
0.1
0.0
−0.1
−0.2
0
10
20
30
40
Lag

Therefore, model B passes all the diagnostics checking and is ready to use for forecasting.

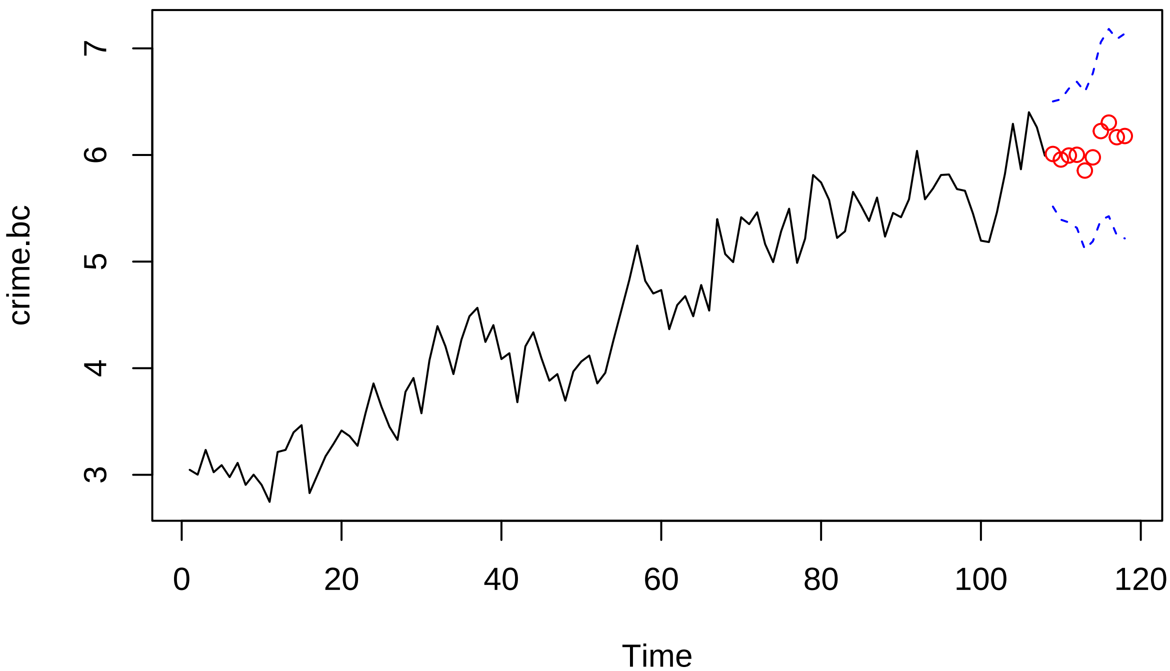
10. Forecasting

In forecasting section, we predict the armed robberies from Jan 1975- Oct 1975 based on our model and then compare with the true val- ues.

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
pred.tr <- predict(fit4, n.ahead = 10)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(crime.bc, xlim=c(1,length(crime.bc)+10), ylim = c(min(crime.bc),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(crime.bc)+1):(length(crime.bc)+10), pred.tr$pred, col="red")
```

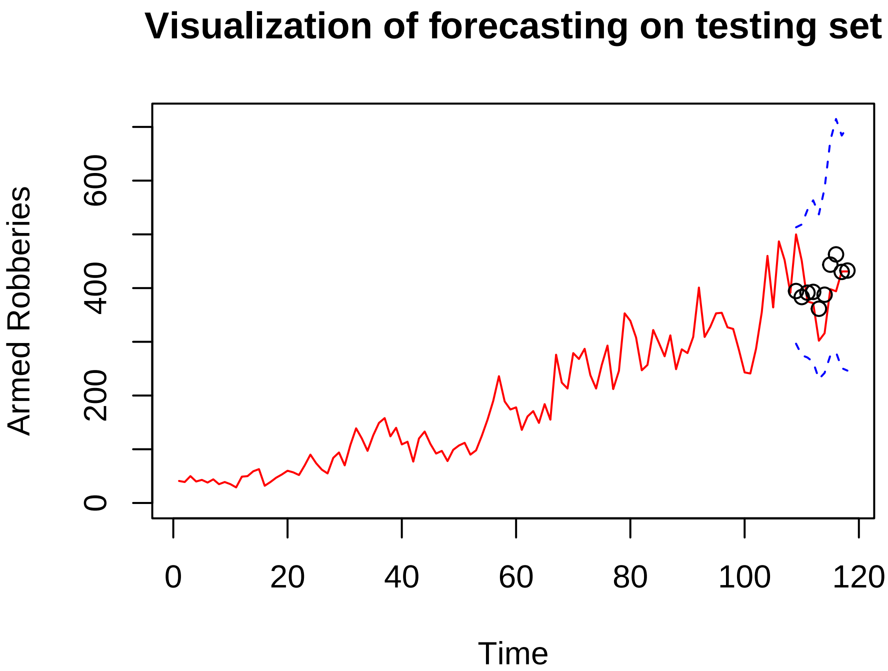


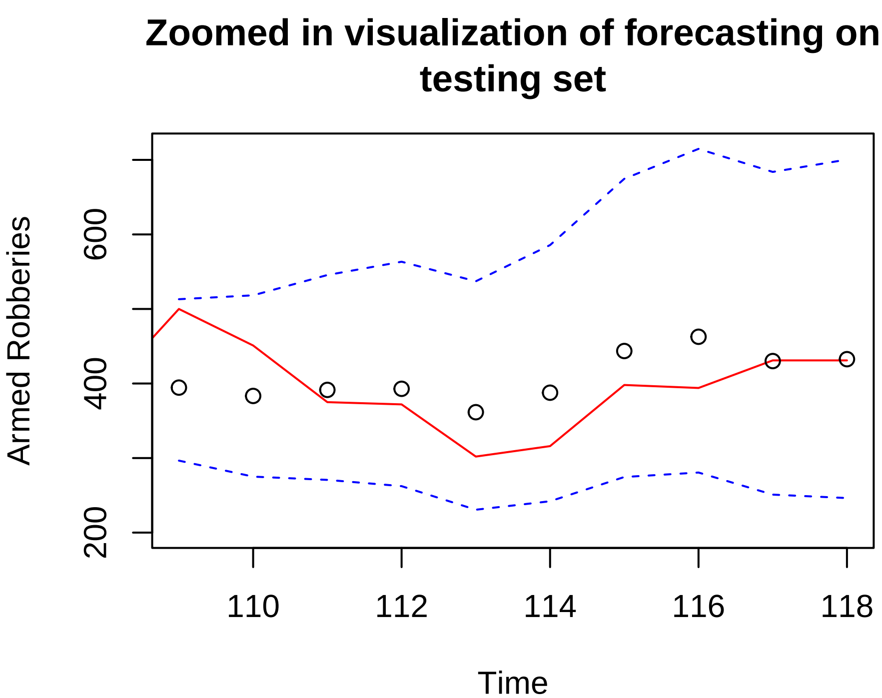
The above figure is the forecast on the transformed data. The true values are within the confidence interval of the forecasting. If we want to compare with the true values of the last ten months, we need to convert the forecasting values back to the scale before box-cox transformation.

This part shows how to convert the data back to the scale before box-cox transformation and compare the true values with predicted values.

```
pred.orig <- (pred.tr$pred)^(1/0.3)
U= (U.tr)^(1/0.3)
L= (L.tr)^(1/0.3)
par(mfrow=c(2,1))
ts.plot(as.numeric(crime_data), ylim = c(0,max(U)),col="red",
        ylab="Armed Robberies",main="Visualization of forecasting on testing set")
lines(U, col="blue", lty="dashed")
```

```
lines(L, col="blue", lty="dashed")
points((length(train_data)+1):(length(train_data)+10), pred.orig, col="black")
ts.plot(as.numeric(crime_data), xlim = c(109,length(train_data)+10), ylim = c(200,max
(U)),col="red",ylab="Armed Robberies",main="Zoomed in visualization of forecasting on
testing set")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(train_data)+1):(length(train_data)+10), pred.orig, col="black")
```





The true values are within the confidence interval of the forecasting. If we see the zoomed version from Lower part, the forecasting results are very close to the true values. Therefore, our model performs well in forecasting for future armed robberies in Boston.