Lab 6

Pstat W 174/274

Model Estimation

Summary of `R` commands for estimation:

1. To difference a time series at lag s:

```
diff(data,lag = s)
```

2. To estimate parameters of an AR model:

```
ar(data, aic = TRUE, order.max = NULL, method = c("..."))
```

If `aic = TRUE`, Akaike's Information Criterion is used to select the order of the AR model to fit. If `aic = FALSE`, the model of order equal to `order.max` is fitted. `method` could be set equal to `"yule-walker"` or `"mle"` for different estimation methods of AR parameters.

3. To estimate parameters of an ARMA model:

```
arima(data, order = c(p, 0, q), method = c("..."))
```

The middle number in `order` is the number of times d the data should be differenced before fitting an ARMA model. For example, `order = c(2, 1, 2)` would fit an ARMA(2,2) once the original time series has been differenced (e.g., to remove the trend component). If the original data is already stationary, put 0 for this number. Refer to help file (type `?arima`) for different methods of estimation.

4. To compare models using AICC:

```
AICc(fittedModel)
```

The `qpcR` package needs to be downloaded and installed in `R` first before using this function. This function will give the Akaike's Information Criterion corrected for bias given a fitted model object from `arima()`. We want to select the model with the *lowest* AICC.

IMPORTANT: For Mac users, it is often the case that you cannot load the qpcR library and therefore cannot use AICc function.

- First downloaded and installed `qpcR` package in `R` by install.packages("qpcR")
- Then please download XQuartz in this link https://www.xquartz.org
- After downloading XQuartz, run library(qpcR)

Sometimes you will get a warning: unable to open X11 displayWarning: 'rgl.init' failed, running with 'rgl.useNULL = TRUE'. But you will be able to use the function AICc.

```
AICc(fittedModel)
```

Boston armed robberies data

1. Data Information and Visualization

The data is in the tsdl library and it is the 484th data in the library. We can see that there are 118 observations. We could also use attr() function to see the subject, source and description of a certain dataset in the tsdl library.

```
library(tsdl)
i = 484
length(tsdl[[i]])
```

```
## [1] 118
```

```
attr(tsdl[[i]], "subject")
```

```
## [1] "Crime"
```

```
attr(tsdl[[i]], "source")
```

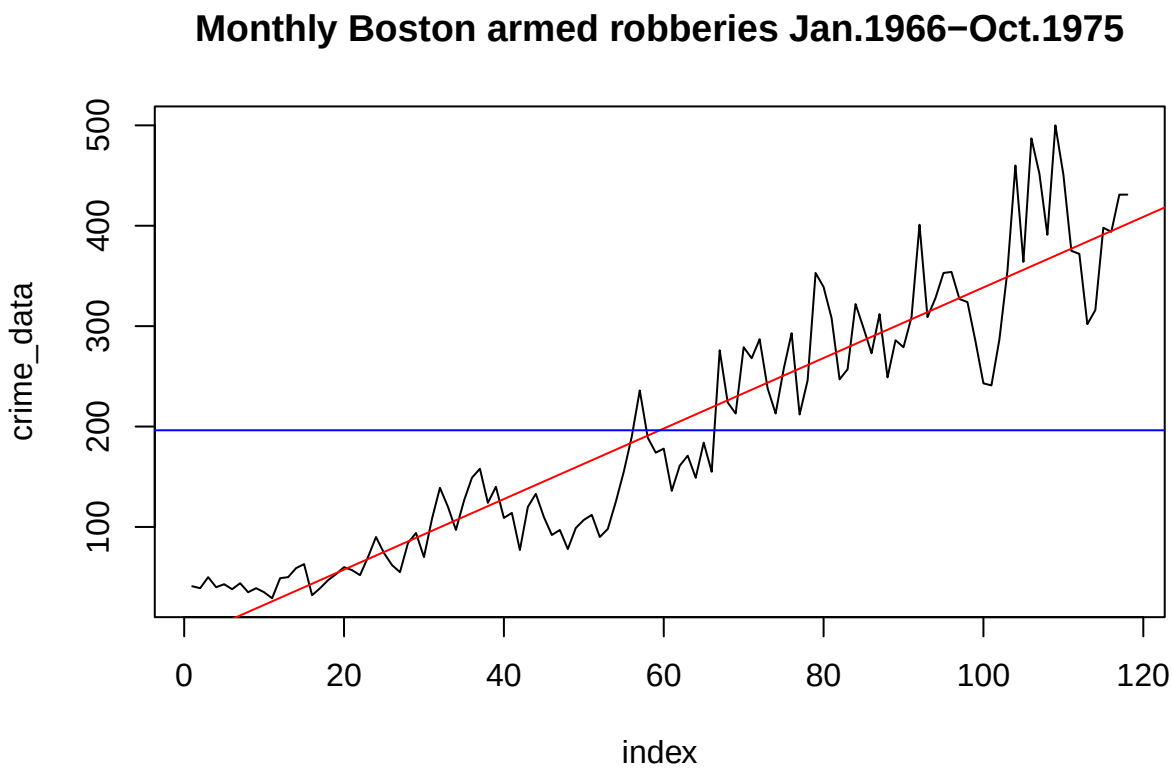
```
## [1] "McCleary & Hay (1980)"
```

```
attr(tsdl[[i]], "description")
```

```
## [1] "Monthly Boston armed robberies Jan.1966-Oct.1975 Deutsch and Alt (1977)"
```

We could first plot the dataset using

```
crime_data = tsdl[[484]]
plot(1:length(crime_data),crime_data, main =
       "Monthly Boston armed robberies Jan.1966-Oct.1975", type = 'l',xlab='index')
index = 1: length(crime_data)
trend <- lm(crime_data ~ index)
abline(trend, col="red")
abline(h=mean(crime_data) , col='blue')
```

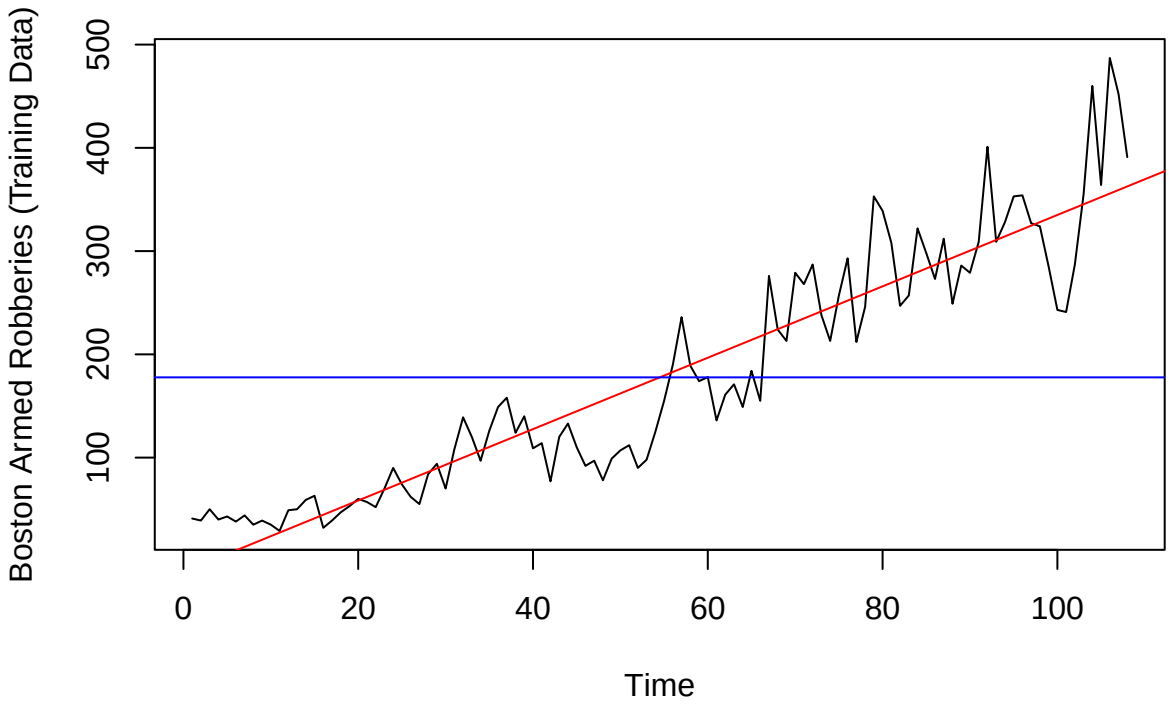


We can see that there exist a positive trend between time and robberies from the red line, which stands for the regression fit of the time series data. Also, there is a clear seasonal pattern in the data: for each year, the robberies reached minimum in the second quarter, especially in April or May; whereas reached maximum in winter periods.

2. Training/Testing split:

To see if our model is good or not, we set aside the last 10 observations(Jan 1975-Oct 1975) as the test set for forecasting and use the rest at training set.

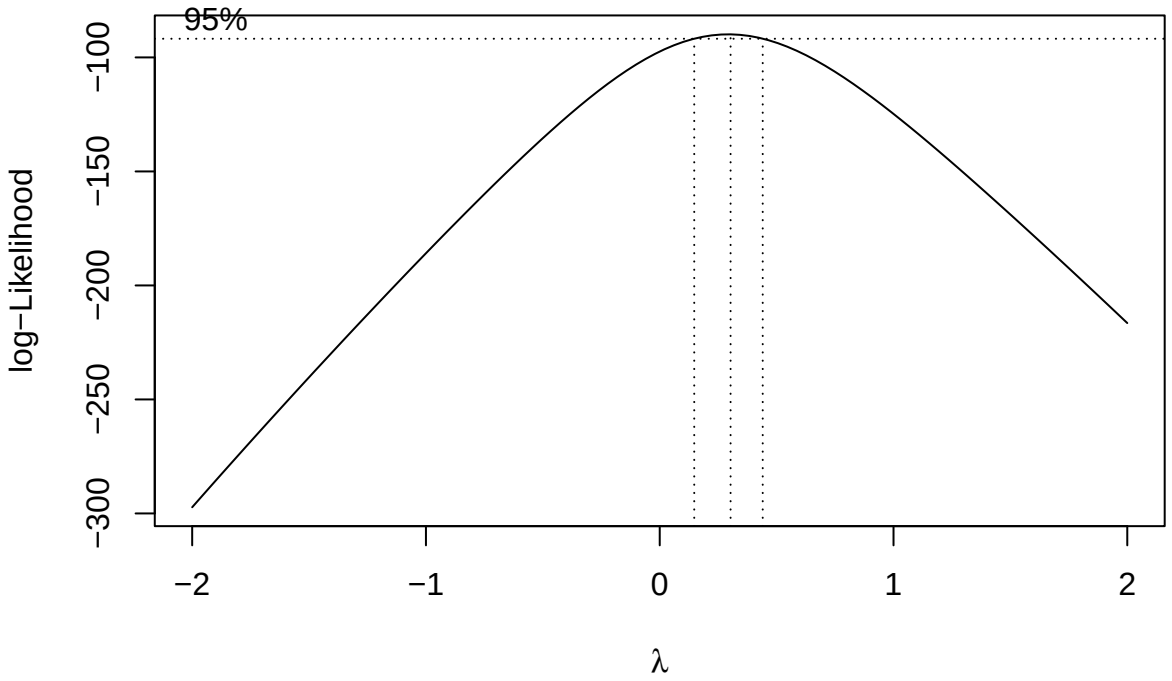
```
train_data = crime_data[c(1: 108)]
test_data = crime_data[c(109:118)]
plot.ts(as.numeric(train_data), main="", ylab="Boston Armed Robberies (Training Data)")
# to generate trend and mean:
ntr=length(as.numeric(train_data))
fit_train <- lm(as.numeric(train_data) ~ as.numeric(1:ntr))
abline(fit_train, col="red")
abline(h=mean(as.numeric(train_data)), col="blue")
```



3. Stabilize the variance: box-cox transformation

We can see that the variance of the data also increases as time increases from the plot above, so in this section, we first transform the data and stablize the variance.

```
library(MASS)
bcTransform <- boxcox(as.numeric(train_data)~ as.numeric(1:length(train_data)))
```




```
bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
```

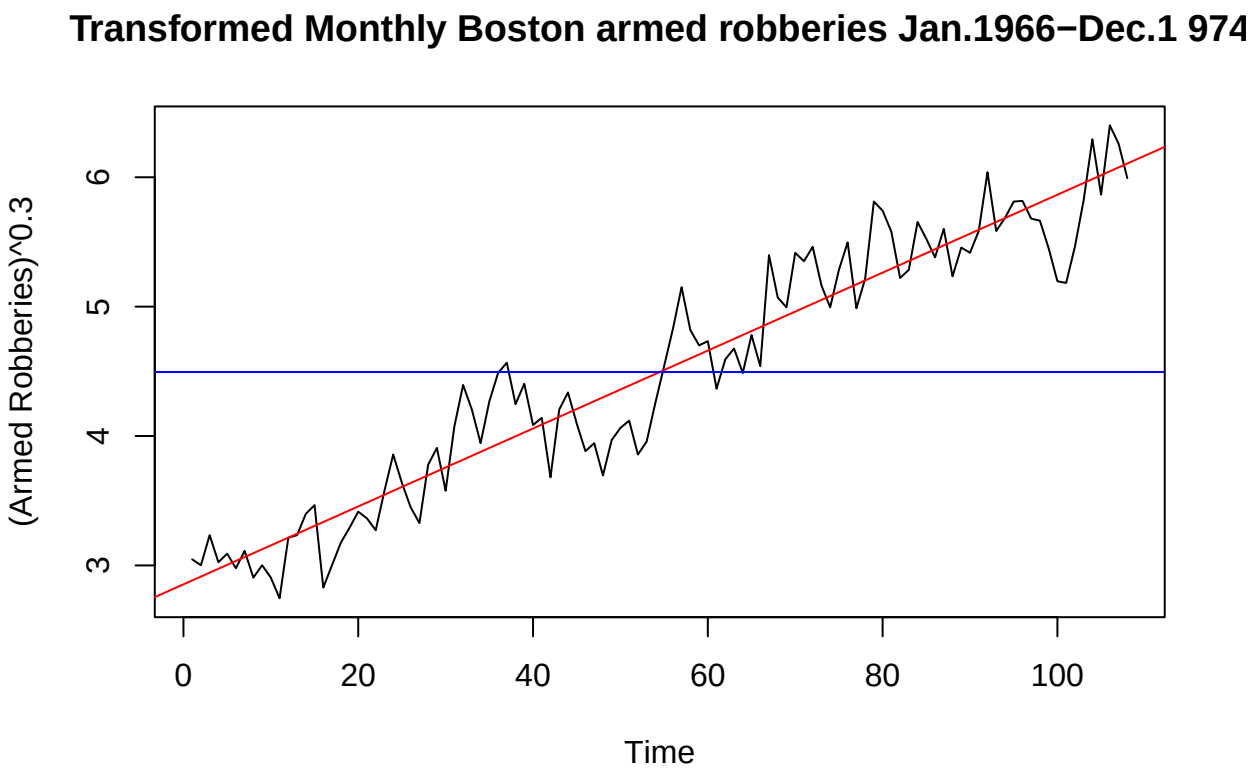
```
## [1] 0.3030303
```

From the above plot, the best λ to get the maximum log-likelihood is around 0.3, so we transformed the training data by taking the power of 0.3 and stabilize the variance.

```
crime.bc = train_data^0.3
#crime.bc = (train_data^0.3-1)/0.3
par(mfrow=c(1,2))
hist(train_data, col="light blue", xlab="", main="Histogram of Training Data")
hist(crime.bc, col="light blue", xlab="", main="Histogram of Transformed Data",breaks
 =10)
```

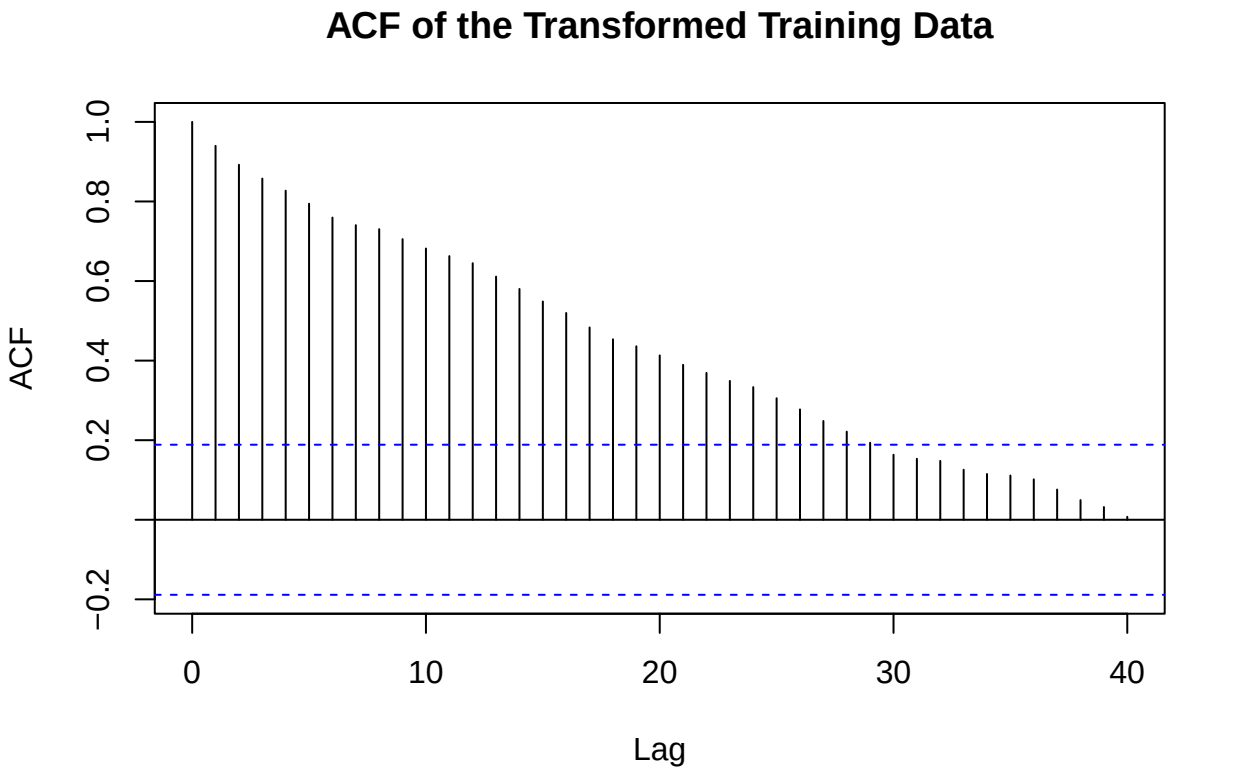
By comparing the histogram before and after box-cox, histogram of transformed data is more Gaussian. Therefore, we will use the transformed data in the following analysis.

```
crime.bc_raw = as.numeric(crime.bc)
plot.ts(crime.bc_raw, main="Transformed Monthly Boston armed robberies Jan.1966-Dec.1 974", ylab="(Arme
# to generate trend and mean:
nt=length(crime.bc_raw)
fit <- lm(crime.bc_raw ~ as.numeric(1:nt))
abline(fit, col="red")
abline(h=mean(crime.bc_raw), col="blue")
```



The variance stabilizes as time goes by but the positive trend is still presented in the data.

```
acf(crime.bc,lag.max=40,main="")
title("ACF of the Transformed Training Data")
```

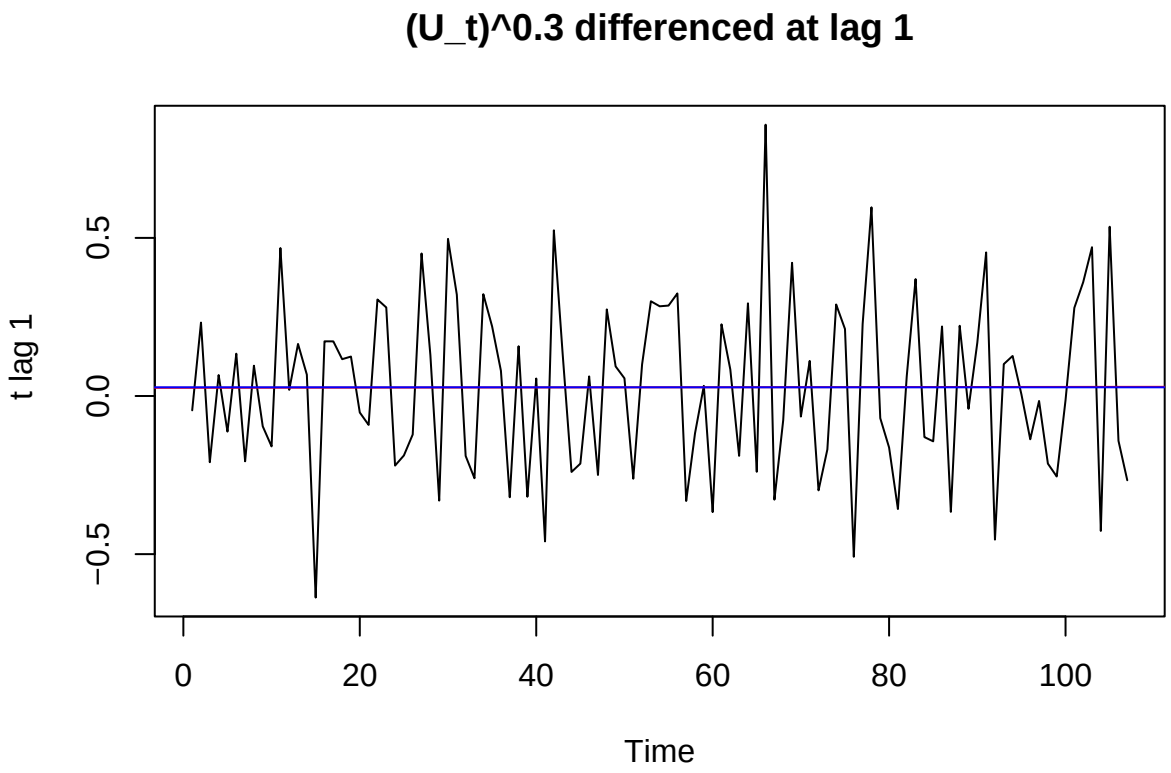


Also, from the Auto-correlation function (ACF) in the plot above, the ACF values decrease slowly. Therefore,

we consider taking difference at lag 1 to eliminate the trend.

4. Remove the trend

```
crime.lag_11 <- diff(crime.bc, lag=1)
plot.ts(crime.lag_11, main="(U_t)^0.3 differenced at lag 1",ylab="crime differenced a
 t lag 1")
fit_11 <- lm(crime.lag_11 ~ as.numeric(1:length(crime.lag_11)))
abline(fit_11, col="red")
abline(h=mean(crime.lag_11), col="blue")
```



After taking difference at lag 1, we plot the differenced data with time. We observe that the trend disappear with mean almost equal to 0. Also, the red regression fit line is also very close to the blue horizontal line for mean. Thus, we conclude that differencing at lag 1 eliminates the trend.

```
var(crime.bc)
```

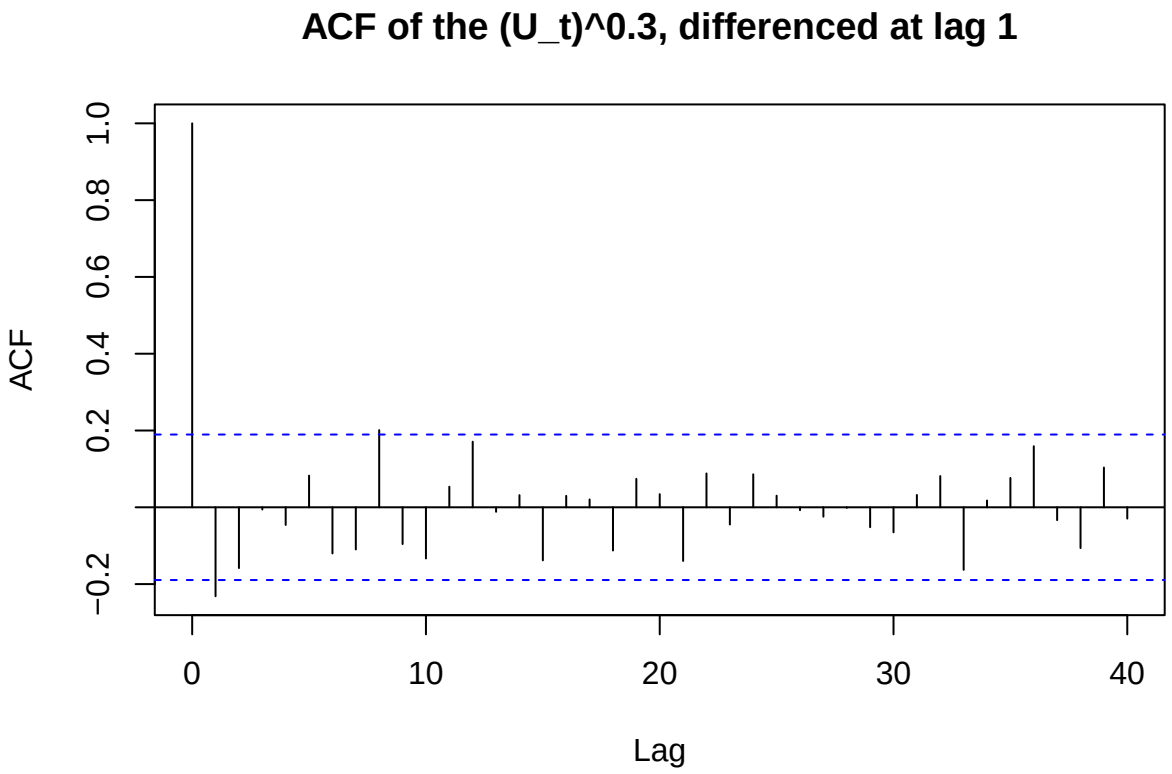
```
## [1] 0.9750683
```

```
var(crime.lag_11)
```

```
## [1] 0.07685027
```

The variance of the data decreases from 0.975 to 0.077.

```
acf(crime.lag_11, lag.max=40, main="")
 title("ACF of the (U_t)^0.3, differenced at lag 1")
```



Also, the ACF decays fast to 0 for de-trend data. Therefore, differencing at lag 1 is a reasonable approach.

5. Check: if we need to remove the seasonality

We then try differencing at lag 12 to remove possible seasonality.

First, from the plot above, we do not see obvious seasonal pattern.

```
crime.lag_12 <- diff(crime.lag_11, lag=12)
var(crime.lag_11)
```

```
## [1] 0.07685027
```

```
var(crime.lag_12)
```

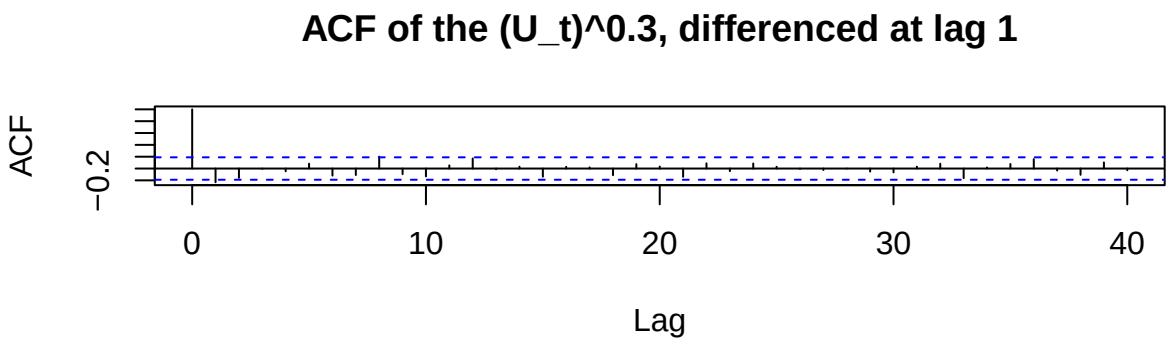
```
## [1] 0.1271474
```

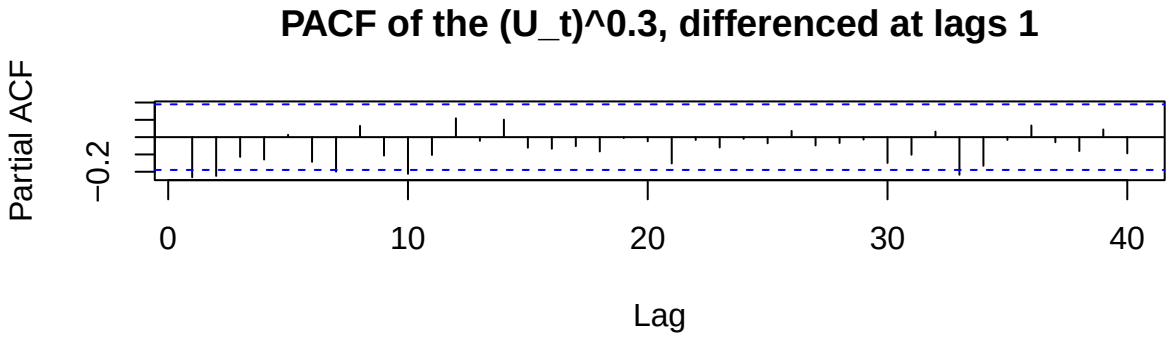
If we difference at lag 12 one time, the variance of the data increases to 0.127. Thus, we over-differnce the data. We conclude the box-cox transformed data differencing at lag 1 is stationary because there is no obvious trend or seasonality and the variance of the data is relatively constant.

6. Model Identification

In this section, we focus on model identification based on sample ACF and PACF. Given the seasonality and trend of the original data, We aim to find a suitable SARIMA model to the data.

```
par(mfrow=c(2,1))
acf(crime.lag_11, lag.max=40, main="")
title("ACF of the (U_t)^0.3, differenced at lag 1")
pacf(crime.lag_11, lag.max=40, main="")
title("PACF of the (U_t)^0.3, differenced at lags 1")
```





q: In ACF plot, only ACF at lag 1 exceeds the 95 percent confidence interval. Therefore, we pick q=1 in the moving average part.

Q: We pick Q=0 because w there is no seasonal lag in ACF.

p: We can try picking p=2 since PACF exceeds the 95 percent confidence interval at lag 2.

P: P=1 or 3 in the SARIMA model.

7. Find suitable ARMA models using maximum likelihood estimation

If you want to find pure ARMA models and would like to automatically decide with p and q to use. Then you can use a for loop and compare models using AICc.

Fit different ARMA models using maximum likelihood estimation and compare the model fits using AICC (Hint: use arima() for estimation and AICc() in library(qpcR) for model comparison - you will need to install this package into R first).

```
library(qpcR)
```

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

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

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

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

```
# Calculate AICc for ARMA models with p and q running from 0 to 5
aiccs <- matrix(NA, nr = 6, nc = 6)
dimnames(aiccs) = list(p=0:5, q=0:5)
for(p in 0:5) {
for(q in 0:5) {
```

```
aiccs[p+1,q+1] = AICc(arima(crime.bc, order = c(p,1,q), method="ML")) }
}
```

```
## Warning in arima(crime.bc, order = c(p, 1, q), method = "ML"): possible
## convergence problem: optim gave code = 1

## Warning in arima(crime.bc, order = c(p, 1, q), method = "ML"): possible
## convergence problem: optim gave code = 1

## Warning in arima(crime.bc, order = c(p, 1, q), method = "ML"): possible
## convergence problem: optim gave code = 1
```

```
aiccs
```

```
##    q
## p          0        1        2        3        4        5
##   0 31.15765 24.78325 23.84409 25.96054 27.94907 30.06108
##   1 27.96802 24.14793 25.96055 28.03600 30.03486 32.12907
##   2 25.45457 26.04958 27.90942 30.08315 24.25020 26.35334
##   3 26.84969 28.09066 30.07363 32.30562 26.39046 24.94726
##   4 28.04717 29.84801 32.26634 26.41918 28.52005 30.34819
##   5 29.86623 31.92990 34.21964 27.92067 32.29370 29.48193
```

```
(aiccs==min(aiccs))
```

```
##    q
## p       0     1     2     3     4     5
##   0 FALSE FALSE  TRUE FALSE FALSE FALSE
##   1 FALSE FALSE FALSE FALSE FALSE FALSE
##   2 FALSE FALSE FALSE FALSE FALSE FALSE
##   3 FALSE FALSE FALSE FALSE FALSE FALSE
##   4 FALSE FALSE FALSE FALSE FALSE FALSE
##   5 FALSE FALSE FALSE FALSE FALSE FALSE
```

From the loop to find the best p and q, we could get the optimal ARMA model to be ARMA(0,2).

8. Model fitting

We have two models from the model identification part (Part 6):

Model A: SARIMA $(2, 1, 1)\ (1, 0, 0)_{12}$

Model B: SARIMA $(2, 1, 1)\ (3, 0, 0)_{12}$

And one model from find suitable ARMA models using maximum likelihood estimation (Part 7):

Model C: ARIMA(0,1,2)

```
fit2 = arima(crime.bc, order=c(2,1,1), seasonal = list(order = c(1,0,0), period = 12)
, method="ML")
fit2
```

Model A:

```
##
## Call:
## arima(x = crime.bc, order = c(2, 1, 1), seasonal = list(order = c(1, 0, 0),
##     period = 12), method = "ML")
```

```
##
## Coefficients:
##          ar1      ar2      ma1    sar1
##       0.2482  -0.0565  -0.5819  0.2377
## s.e.  0.4504   0.1997   0.4436  0.1001
##
## sigma^2 estimated as 0.06518:  log likelihood = -6.19,  aic = 22.39
```

The 2 times standard errors are larger than Estimated Coeffcients of ar1, ar2. So it means that 0 is within the confidence interval of the coefficients. We set the coefficients of ar1 and ar2 to be 0, which reduces the model to be SARIMA $(0, 1, 1)\ (1, 0, 0)_{12}$

```
fit3 = arima(crime.bc, order=c(0,1,1), seasonal = list(order = c(1,0,0), period = 12)
, method="ML")
fit3
```

```
##
## Call:
## arima(x = crime.bc, order = c(0, 1, 1), seasonal = list(order = c(1, 0, 0),
##     period = 12), method = "ML")
##
## Coefficients:
##           ma1    sar1
##       -0.3891  0.2530
## s.e.   0.1052  0.0986
##
## sigma^2 estimated as 0.06643:  log likelihood = -7.24,  aic = 20.48
```

Both ma1 and sar1 are non-zero.

Therefore, the model can be written as:

$$(1 - 0.253B^{12})(1 - B)X_t = (1 - 0.3891B)Z_t$$

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

Model B: For model B, increasing the P from 1 to 3 also result in estimated coeffcients of ar1 and ar2 equal to 0. We reduce the model B to SARIMA $(0, 1, 1)\ (3, 0, 0)_{12}$

```
fit4.1 = arima(crime.bc, order=c(0,1,1), seasonal = list(order = c(3,0,0),
                                                         period = 12), method="ML")
fit4.1
```

```
##
## Call:
## arima(x = crime.bc, order = c(0, 1, 1), seasonal = list(order = c(3, 0, 0),
##     period = 12), method = "ML")
##
## Coefficients:
##           ma1    sar1    sar2    sar3
##       -0.4474  0.1959  0.0537  0.2736
## s.e.   0.1063  0.0958  0.1014  0.1104
##
## sigma^2 estimated as 0.06059:  log likelihood = -3.92,  aic = 17.84
```

sar2 is not significant from the above output. Therefore, we fix sar2 to 0 and refit the model.

```r
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")
```

```
## Warning in arima(crime.bc, order = c(0, 1, 1), seasonal = list(order = c(3, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```r
fit4
```

```
##
## Call:
## arima(x = crime.bc, order = c(0, 1, 1), seasonal = list(order = c(3, 0, 0),
##     period = 12), fixed = c(NA, NA, 0, NA), method = "ML")
##
## Coefficients:
##           ma1    sar1  sar2    sar3
##       -0.4427  0.2112     0  0.2858
## s.e.   0.1052  0.0918     0  0.1077
##
## sigma^2 estimated as 0.06077:  log likelihood = -4.06,  aic = 16.12
```

So we choose the final model B as SARIMA $(0, 1, 1)$ $(3, 0, 0)_{12}$ with coefficient of sar2 equal to 0. Therefore, the model can be written as:

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

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

```r
fit5 = arima(crime.bc, order = c(0,1,2), method="ML")
fit5
```

Model C:

```
##
## Call:
## arima(x = crime.bc, order = c(0, 1, 2), method = "ML")
##
## Coefficients:
##           ma1      ma2
##       -0.2910  -0.1752
## s.e.   0.0959   0.0991
##
## sigma^2 estimated as 0.06898:  log likelihood = -8.86,  aic = 23.73
```

From the output, ma1 and ma2 are significant. Therefore, the model can be written as:

$$(1 - B)X_t = (1 - 0.291B - 0.1752B^2)Z_t$$

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

AICc comparison:

```r
library(qpcR)
AICc(fit3)
```

```
## [1] 20.58981
```

```r
AICc(fit4)
```

```
## [1] 16.51284
```

```r
AICc(fit5)
```

```
## [1] 23.84409
```

We can see that model B: SARIMA $(0, 1, 1)$ $(3, 0, 0)_{12}$ with coefficient of sar2 equal to 0 is the best model according to AICc.

9. Check the model stationarity/invertibility:

Model A:

$$(1 - 0.253B^{12})(1 - B)X_t = (1 - 0.3891B)Z_t$$

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

- SAR part: 0.253 is smaller than 1. So the roots lie outside the unit circle.
- AR part: no AR part
- SMA part: no SMA part
- MA part: 0.3891 is smaller than 1. So the roots lie outside the unit circle.

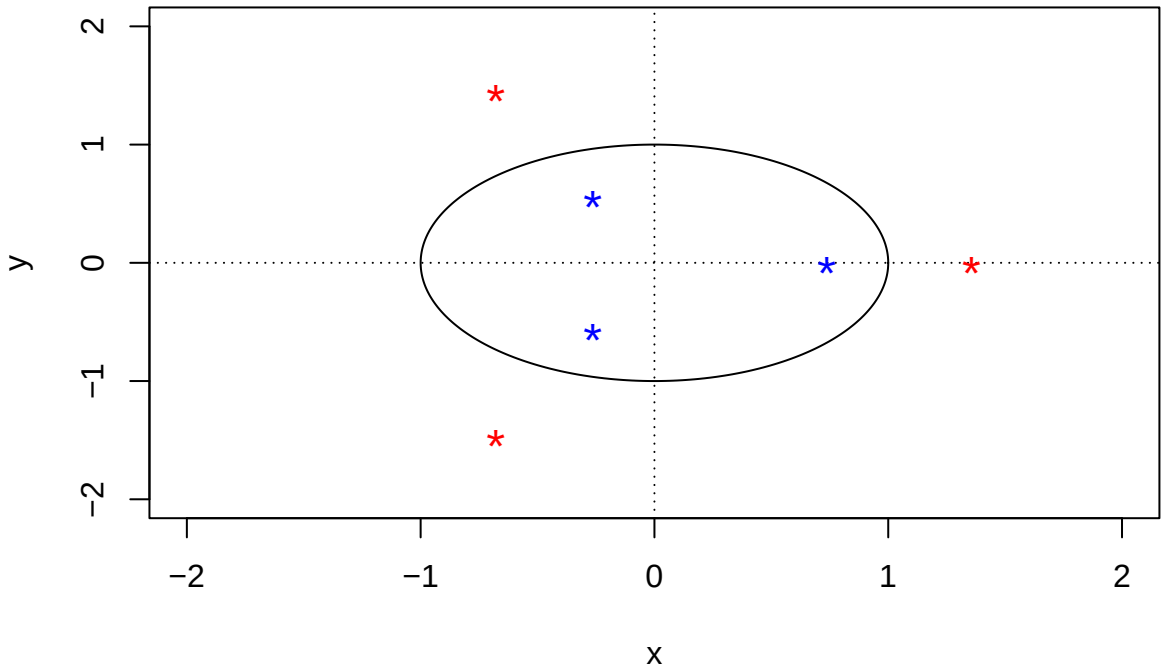
Model B:

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

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

- SAR part: The blue dots are inside the unit circle, meaning that the roots(red) are ouside the unit circle

```r
plot.roots(NULL,polyroot(c(1, -0.2112,0,-0.2858)), main="roots of SAR part of model B"
)
```



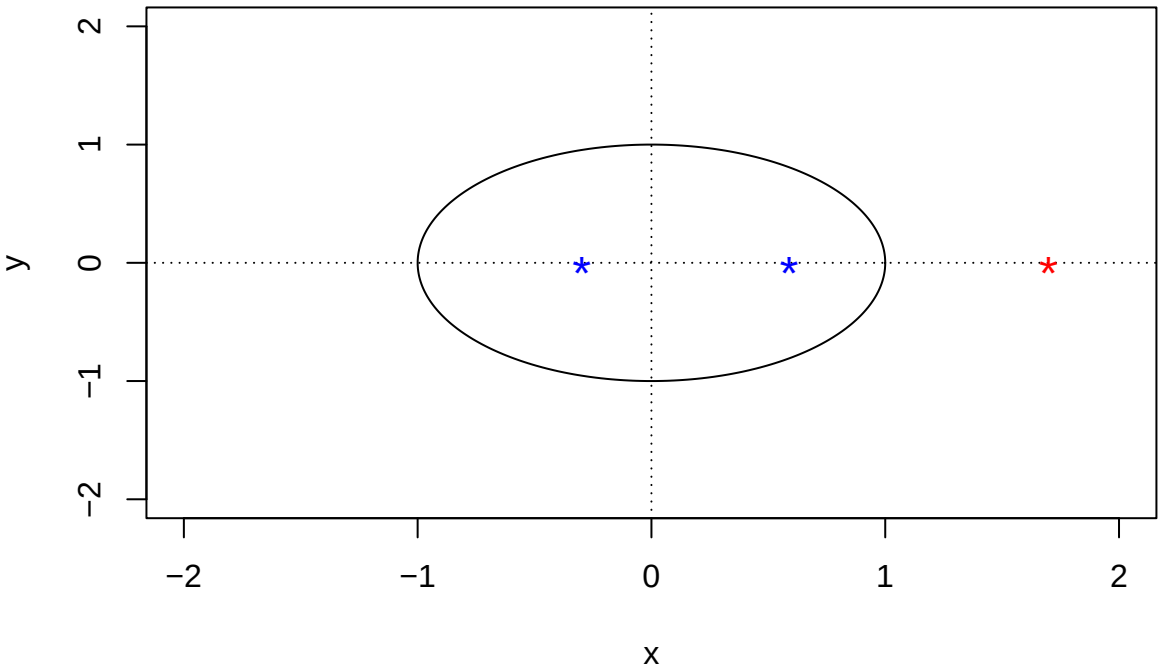
- AR part: no AR part
- SMA part: no SMA part
- MA part: 0.4427 is smaller than 1. So the roots lie outside the unit circle.

Model C:

$$(1-B)X_t = (1 - 0.291B - 0.1752B^2)Z_t$$

and $X_t = U_t^{1/3}$ where U_t was the original data. * SAR part: no SAR part * AR part: no AR part * SMA part: no SMA part * MA part: The blue dots are inside the unit circle, meaning that the roots(red) are ouside the unit circle

```
plot.roots(NULL,polyroot(c(1, -0.291,-0.1752)), main="roots of MA part of model C")
```



Therefore, all three models pass the check the model stationarity/invertibility. According to AICc, we choose model B as our final model and will be used to do diagnostics next week.

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.