Untitled

Kerem Turgutlu

November 26, 2017

```r
library(tidyverse)
library(forecast)
library(lawstat)
library(tseries)
```

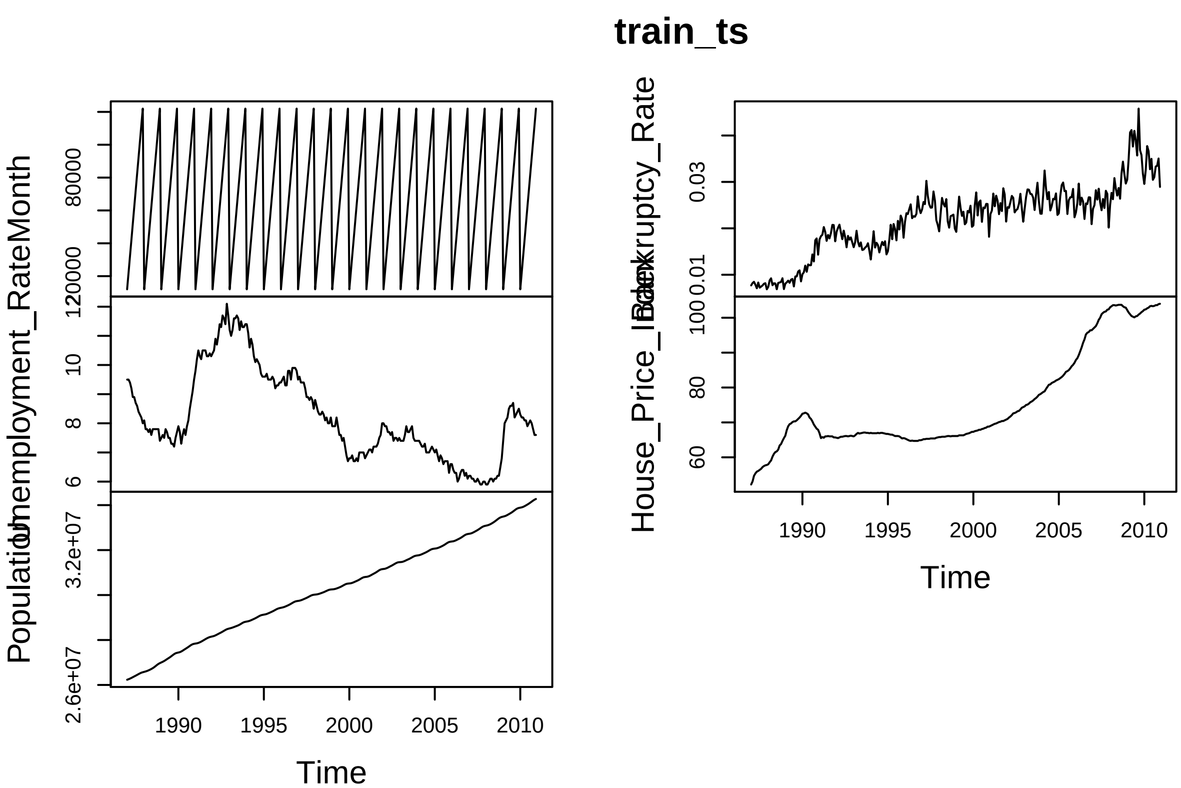
```r
train <- read.csv('train.csv')[1:288,]
test <- read.csv('test.csv')
```

```r
train %>% glimpse()
```

```
## Observations: 288
## Variables: 5
## $ Month             <int> 11987, 21987, 31987, 41987, 51987, 61987, 71...
## $ Unemployment_Rate <dbl> 9.5, 9.5, 9.4, 9.2, 8.9, 8.9, 8.7, 8.6, 8.4,...
## $ Population        <int> 26232423, 26254410, 26281420, 26313260, 2634...
## $ Bankruptcy_Rate   <dbl> 0.0077004, 0.0082196, 0.0084851, 0.0078326, ...
## $ House_Price_Index <dbl> 52.2, 53.1, 54.7, 55.4, 55.9, 56.1, 56.4, 56...
```

```r
train_ts <- ts(train,start = 1987, frequency = 12)
test_ts <- ts(test, start = 2011, frequency = 12)
```

```r
plot(train_ts)
```



There is great correlation between House_Price_Index and Bankruptcy_Rate, probably even a higher one with lagged values of House_Price_Index.

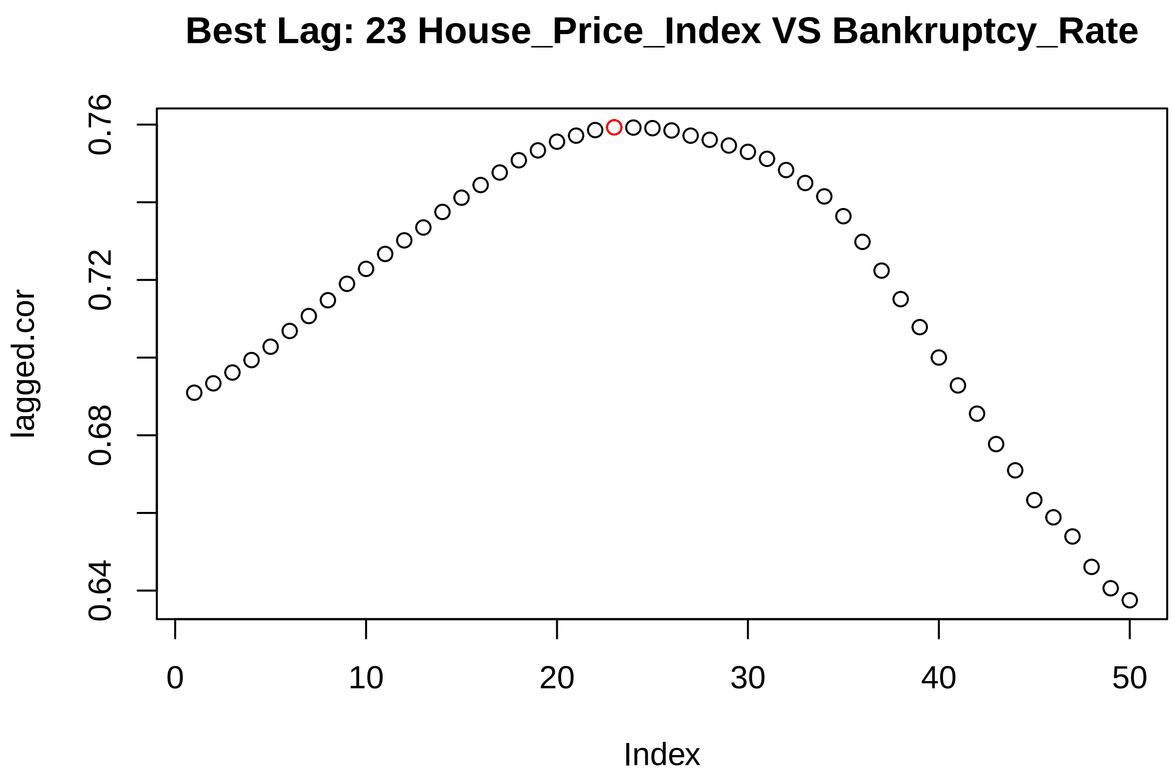
```r
cor(train)
```

```
##                        Month Unemployment_Rate Population Bankruptcy_Rate
## Month              1.00000000       -0.02322856  0.0501926     -0.00459977
## Unemployment_Rate -0.02322856        1.00000000 -0.5431182     -0.31690705
## Population         0.05019260       -0.54311821  1.0000000      0.89840496
## Bankruptcy_Rate   -0.00459977       -0.31690705  0.8984050      1.00000000
## House_Price_Index  0.04548785       -0.54305931  0.8601513      0.68970802
##                   House_Price_Index
## Month                    0.04548785
## Unemployment_Rate       -0.54305931
## Population               0.86015125
## Bankruptcy_Rate          0.68970802
## House_Price_Index        1.00000000
```

```r
lagged.cor <- c()

h = 50
for (i in (seq(h))){
  lagged_house <- lag(train$House_Price_Index, n = i)
  cor.i <- cor(lagged_house, train$Bankruptcy_Rate, use = 'complete.obs')
  lagged.cor <- c(lagged.cor, cor.i)
}
```

Lagged Correlation Plot h vs Correlation

```r
best.idx <- which.max(lagged.cor)
plot(lagged.cor)
points(best.idx, lagged.cor[best.idx], col='red')
title(paste('Best Lag:' , best.idx,'House_Price_Index VS Bankruptcy_Rate'))
```



SARIMA MODEL (Univariate Bankruptcy)

```
bankruptcy_ts <- ts(train$Bankruptcy_Rate, frequency = 12)
```

24 years data…

```
length(bankruptcy_ts) /12
```

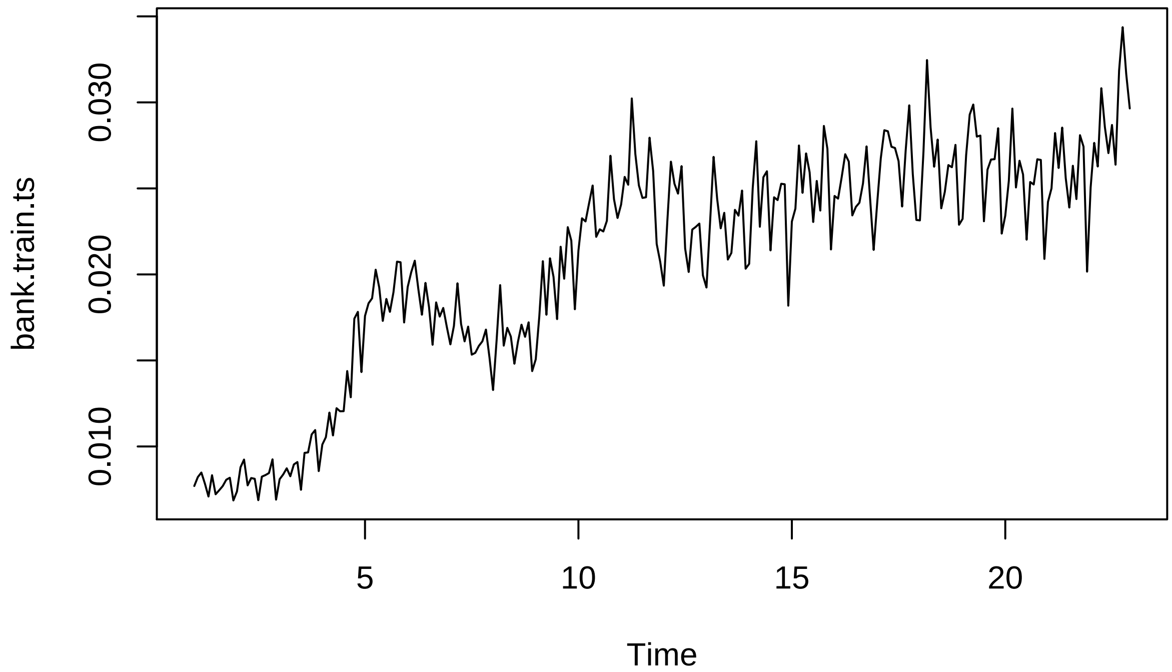
```
## [1] 24
```

Split train - valid (Last 2 Years as Valid)

```
bank.train.ts <- ts(bankruptcy_ts[1:264], frequency = 12)
bank.valid.ts <- ts(bankruptcy_ts[265:288], frequency = 12)
```

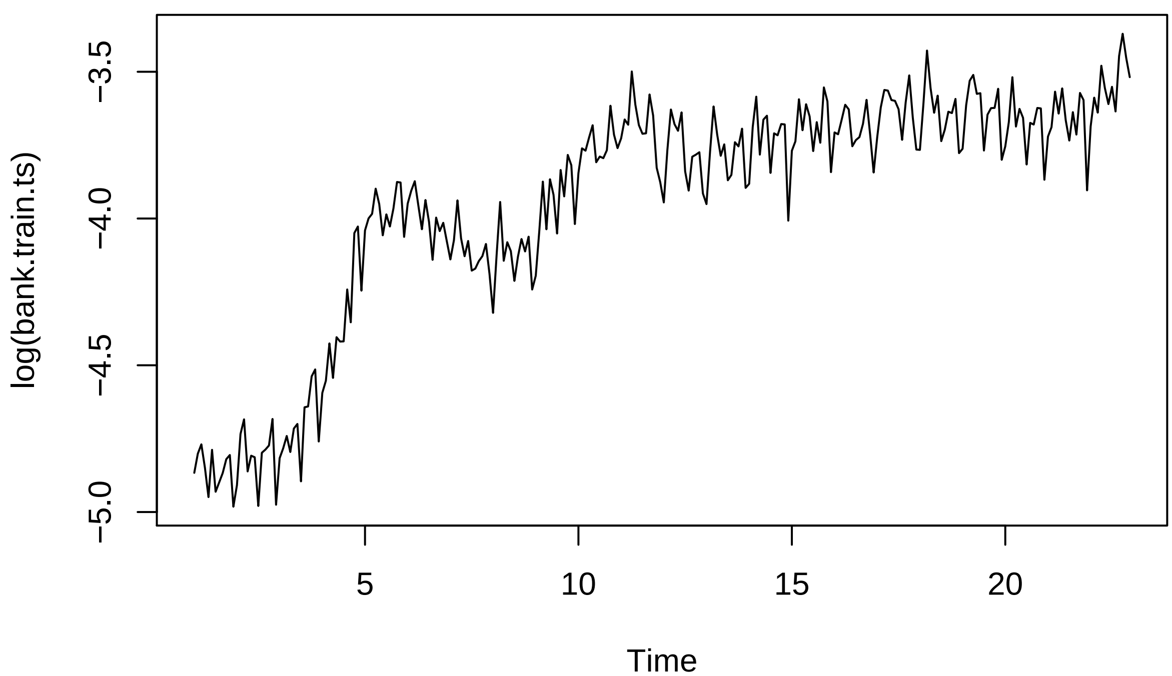
Plot Training

```
plot(bank.train.ts)
```



Try log transform, looks better.

```
bank.train.ts.log <- log(bank.train.ts)
bank.valid.ts.log <- log(bank.valid.ts)
plot(log(bank.train.ts))
```



Do 1 diff

```
ndiffs(bank.train.ts.log)
```

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

```r
bank.train.ts.log.D10 <- diff(bank.train.ts.log)
ndiffs(bank.train.ts.log.D10)
```

```
## [1] 0
```

```r
nsdiffs(bank.train.ts.log.D10)
```

```
## [1] 0
```

There seem to be no seasonality and ts is now stationary

d = 2 makes time series stationary…

```r
adf.test(bank.train.ts.log.D10, k = 48)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  bank.train.ts.log.D10
## Dickey-Fuller = -3.2651, Lag order = 48, p-value = 0.07745
## alternative hypothesis: stationary
```

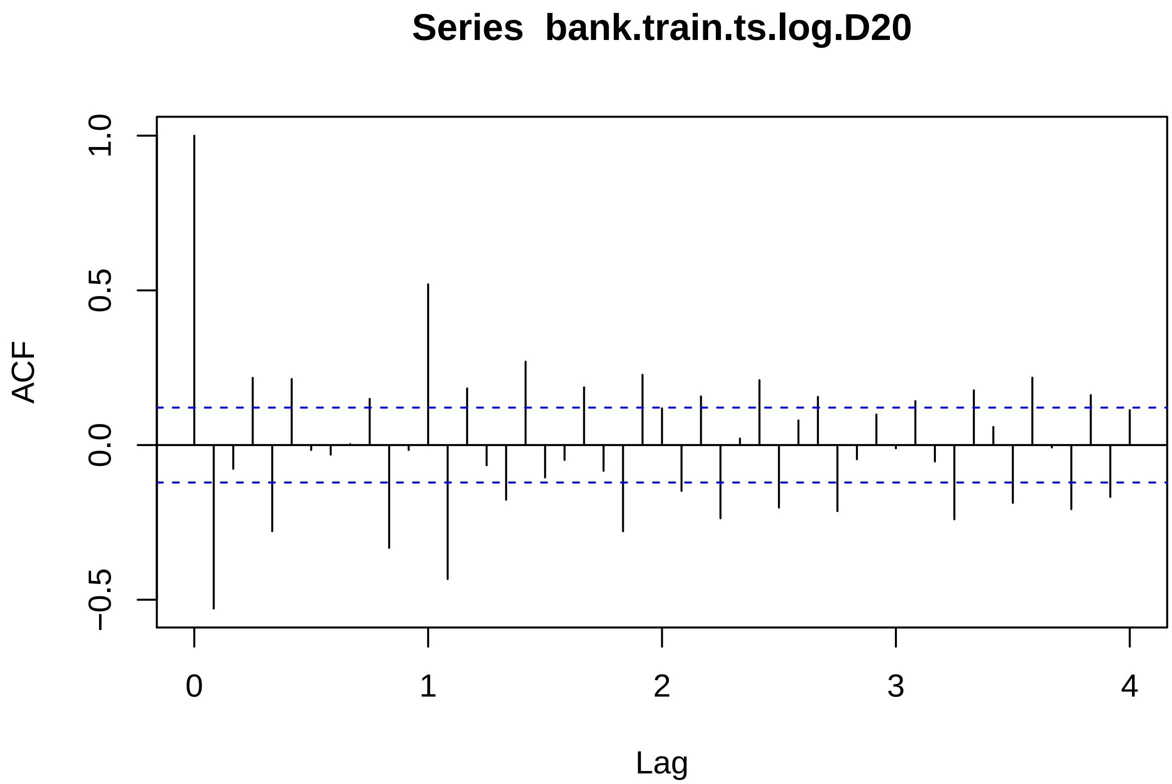
```r
adf.test(diff(bank.train.ts.log.D10), k = 48)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(bank.train.ts.log.D10)
## Dickey-Fuller = -4.5199, Lag order = 48, p-value = 0.01
## alternative hypothesis: stationary
```

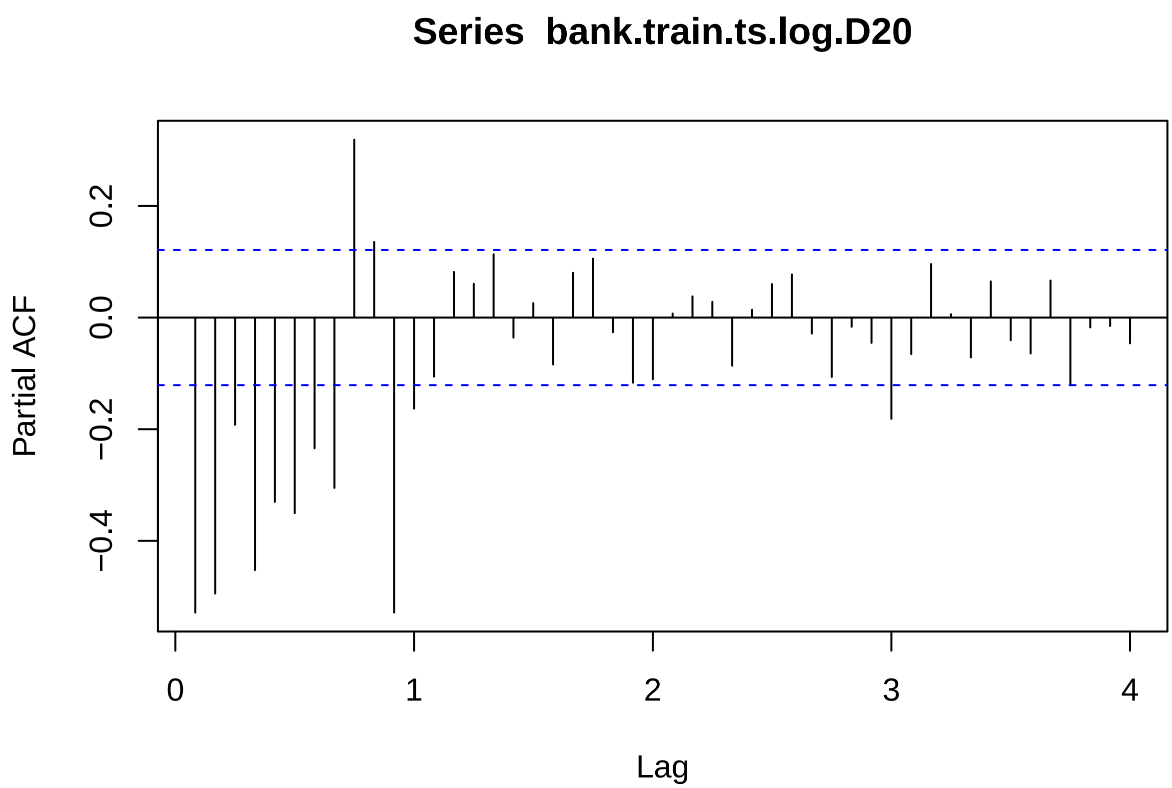
```r
bank.train.ts.log.D20 <-  diff(bank.train.ts.log.D10)
```

Pick p, q, p <= 5, q <= 2

```r
acf(bank.train.ts.log.D20, lag.max = 48)
```




```r
pacf(bank.train.ts.log.D20, 48)
```



Check auto.arima and acf plots, if any suggestions try out other models. Model seems reasonable.

```r
auto.arima(bank.train.ts.log, d=1)
```

```
## Series: bank.train.ts.log
## ARIMA(3,1,2)(1,0,0)[12]
##
## Coefficients:
##           ar1      ar2      ar3     ma1     ma2    sar1
##       -1.8164  -1.3225  -0.4436  1.2309  0.2573  0.7521
## s.e.   0.2091   0.2833   0.1179  0.2269  0.2095  0.0418
##
## sigma^2 estimated as 0.005405:  log likelihood=311.3
## AIC=-608.6   AICc=-608.16   BIC=-583.6
```

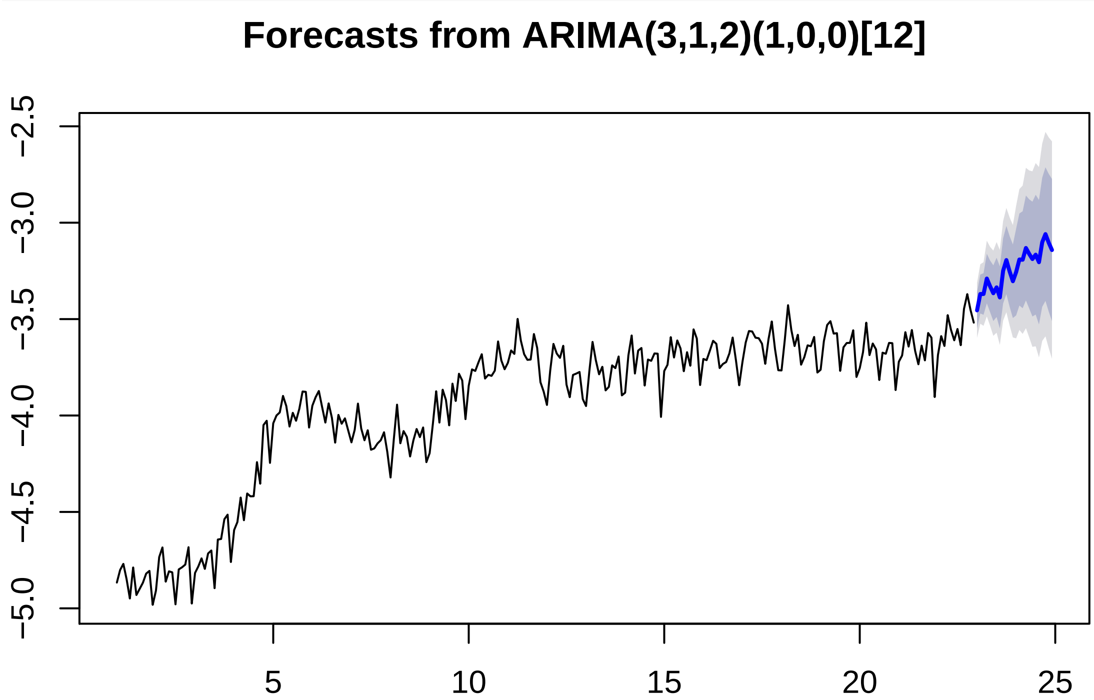
```r
arima.model.312.100 <- arima(bank.train.ts.log, order = c(3, 1, 2), seasonal = c(1, 0, 0))
```

Define rmse and make predictions

```r
#rmse
rmse <- function(true, preds){return(sqrt(mean((true - preds)**2)))}

#preds
valid.preds <- forecast(arima.model.312.100, length(bank.valid.ts))
valid.rmse <- rmse(as.numeric(exp(valid.preds$mean)), bank.valid.ts)
paste(valid.rmse)
```

```
## [1] "0.00766115438404967"
```

```r
#plot predictions
plot(valid.preds)
```



Search for optimal p, q based on rmse on validation

ARIMA model gives the best result with params: p = 3, q = 1, d = 2, rmse ~ 0.00520

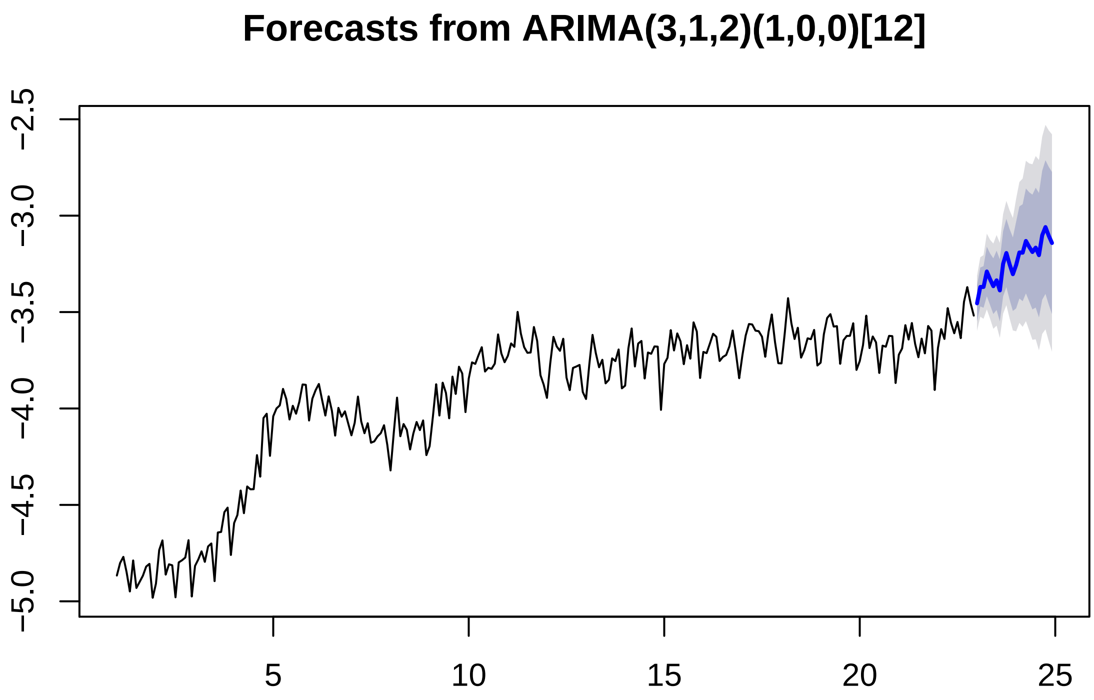
```r
valid_rmse <- function(model, valid_ts){
  valid.preds <- forecast(model, length(valid_ts))
  valid.rmse <- rmse(as.numeric(exp(valid.preds$mean)), exp(valid_ts))
  return(valid.rmse)
}


p <- seq(5)
q <- seq(2)
comb <- expand.grid(p, q)
names(comb) <- c('p', 'q')
for (i in 1:nrow(comb)){
  p <- comb[i, 'p']
  q <- comb[i, 'q']
  print(paste(p, q))
  model <- arima(bank.train.ts.log, order = c(p, 2, q), seasonal = c(0, 0, 0))
  val_rmse <- valid_rmse(model, bank.valid.ts.log)
  print(val_rmse)
  cat('\n')
}
```

```
## [1] "1 1"
## [1] 0.005742243
##
## [1] "2 1"
## [1] 0.005295898
##
## [1] "3 1"
## [1] 0.005203687
##
## [1] "4 1"
## [1] 0.005422778
##
## [1] "5 1"
## [1] 0.005748783
##
## [1] "1 2"
## [1] 0.01146711
##
## [1] "2 2"
## [1] 0.005634718
##
## [1] "3 2"
## [1] 0.01116891
##
## [1] "4 2"
## [1] 0.006402931
##
## [1] "5 2"
## [1] 0.005424213
```

Build best model check forecasts

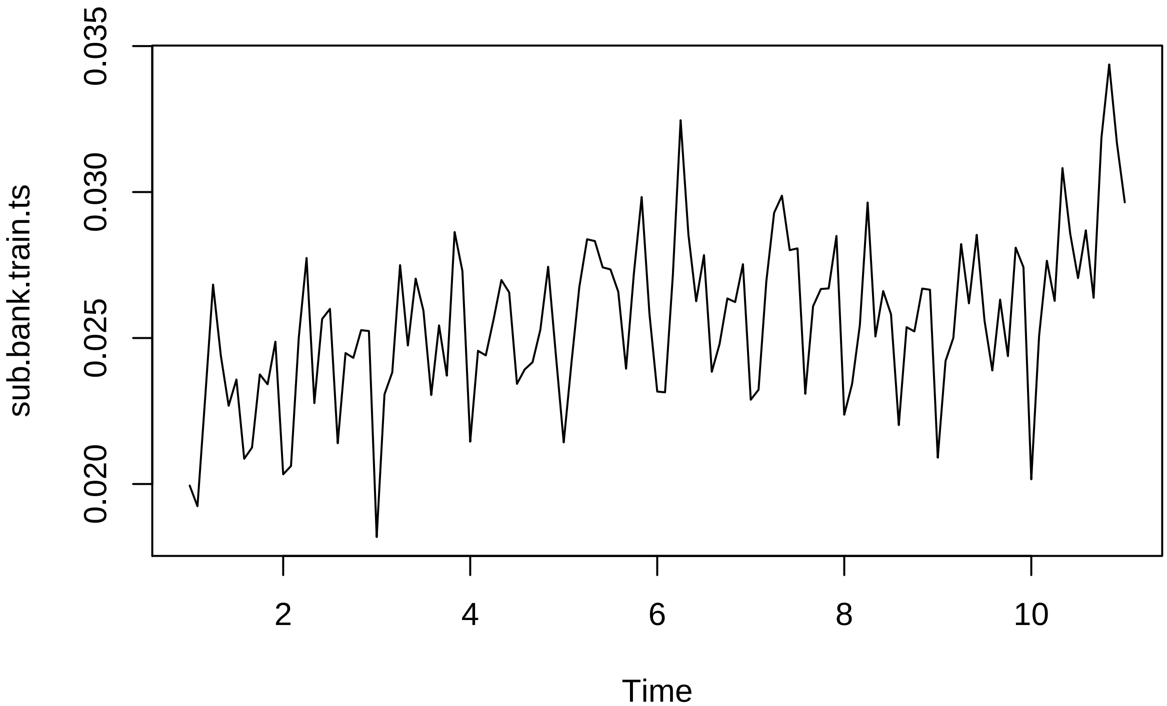
```r
best.model <- arima(bank.train.ts.log, order = c(3, 2, 1), seasonal = c(0, 0, 0))
arima.preds <- forecast(best.model, h = length(bank.valid.ts.log))
```

```r
plot(valid.preds)
```



Subset time series for SARIMA model

```r
# number of years to discard from 24 years
# we can search for optimal years to discard by search
out_years = 12
sub.bank.train.ts <- ts(bankruptcy_ts[(out_years*12):264], frequency = 12)
bank.valid.ts <- ts(bankruptcy_ts[265:288], frequency = 12)

plot(sub.bank.train.ts)
```




```
adf.test(sub.bank.train.ts, k = 12)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  sub.bank.train.ts
## Dickey-Fuller = -1.0811, Lag order = 12, p-value = 0.922
## alternative hypothesis: stationary
```

d = 1, D = 3 or 4

```
adf.test(diff(diff(sub.bank.train.ts, lag = 4)), k=12)
```

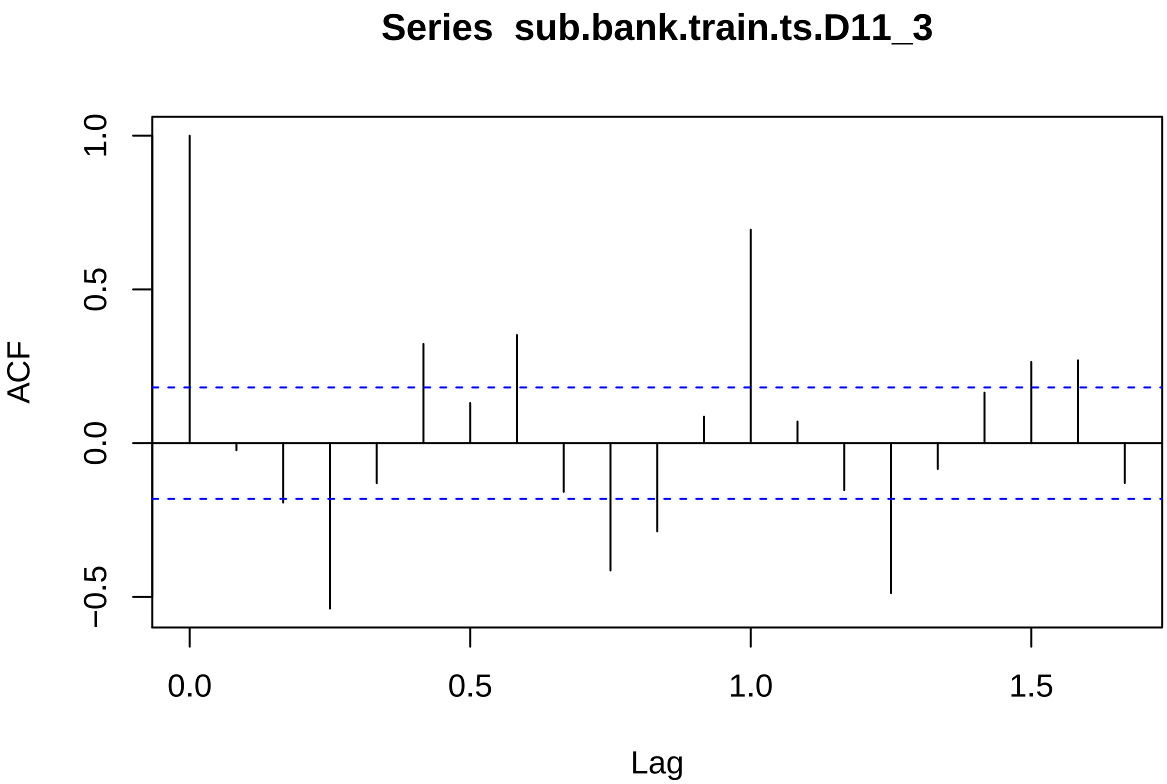
```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(diff(sub.bank.train.ts, lag = 4))
## Dickey-Fuller = -4.4165, Lag order = 12, p-value = 0.01
## alternative hypothesis: stationary
```

We have to timeseries with 1 trend diff and either 1 3 or 4 lagged seasonal diff

```
sub.bank.train.ts.D11_3 <- diff(diff(sub.bank.train.ts, lag = 3))
sub.bank.train.ts.D11_4 <- diff(diff(sub.bank.train.ts, lag = 4))
```

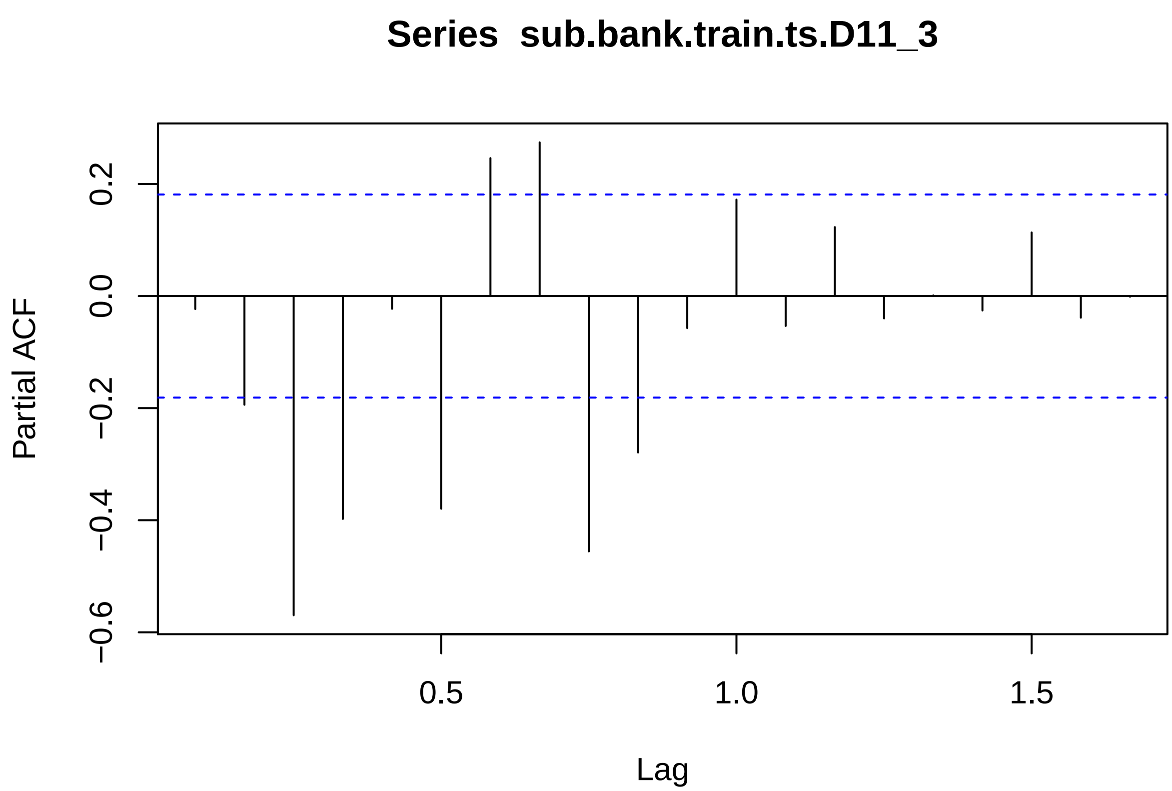
For ts with period m = 3 , Q <= 5, q <= 3

```
acf(sub.bank.train.ts.D11_3)
```



P <= 3, p <= 2

```r
pacf(sub.bank.train.ts.D11_3)
```



Do grid search for best params for ts with period = 3

```r
valid_rmse <- function(model, valid_ts){
  valid.preds <- forecast(model, length(valid_ts))
  valid.rmse <- rmse(as.numeric(valid.preds$mean), valid_ts)
  return(valid.rmse)
}


P <- seq(3)
Q <- seq(5)
p <- seq(2)
q <- seq(3)

comb <- expand.grid('p' = p, 'q' = q, 'P' = P, 'Q' = Q)


best.rmse <- Inf
best.comb <- NA
for (i in 1:nrow(comb)){
  p <- comb[i, 'p']
  q <- comb[i, 'q']
  P <- comb[i, 'P']
  Q <- comb[i, 'Q']

  model <- arima(sub.bank.train.ts, order = c(p, 1, q), seasonal = list(order = c(P, 1, Q), period = 12
  val_rmse <- valid_rmse(model, bank.valid.ts)
  if (val_rmse < best.rmse){
    best.rmse <- val_rmse
    best.comb <- c(p, q, P, Q)
  }
}
```

Best Model SARIMA (1, 1, 3) (3, 1, 2)

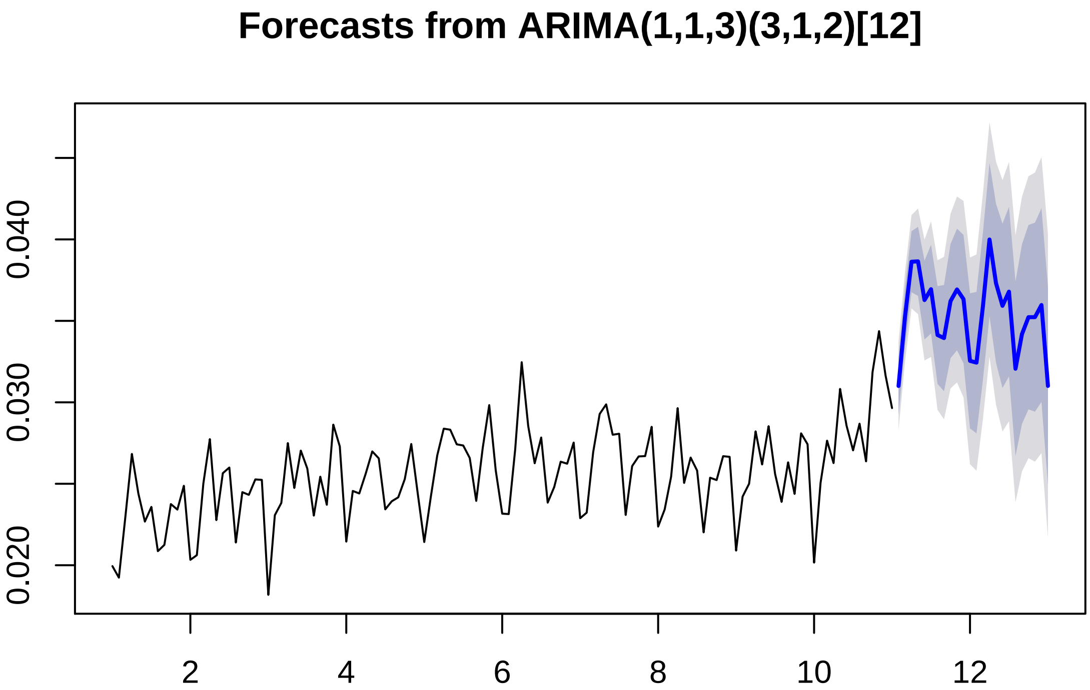
```r
model <- arima(sub.bank.train.ts, order = c(1, 1, 3), seasonal = list(order = c(3, 1, 2), period = 12),
val_rmse <- valid_rmse(model, bank.valid.ts)
sarima.preds <- forecast(model, h = length(bank.valid.ts))
paste('best rmse', best.rmse)
```

```
## [1] "best rmse 0.00296372580827692"
```

```r
paste(c('p:', 'q:', 'P:', 'Q:'), best.comb)
```

```
## [1] "p: 1" "q: 3" "P: 3" "Q: 2"
```

```r
plot(sarima.preds)
```


Forecasts from ARIMA(1,1,3)(3,1,2)[12]
0.020
0.030
0.040
2
4
6
8
10
12