Group Project Stuff

```
library(forecast, quietly = T)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## This is forecast 7.3
```

```
library(tseries, quietly = T)
library(vars)
```

```
## Loading required package: MASS
```

```
## Loading required package: strucchange
```

```
## Loading required package: sandwich
```

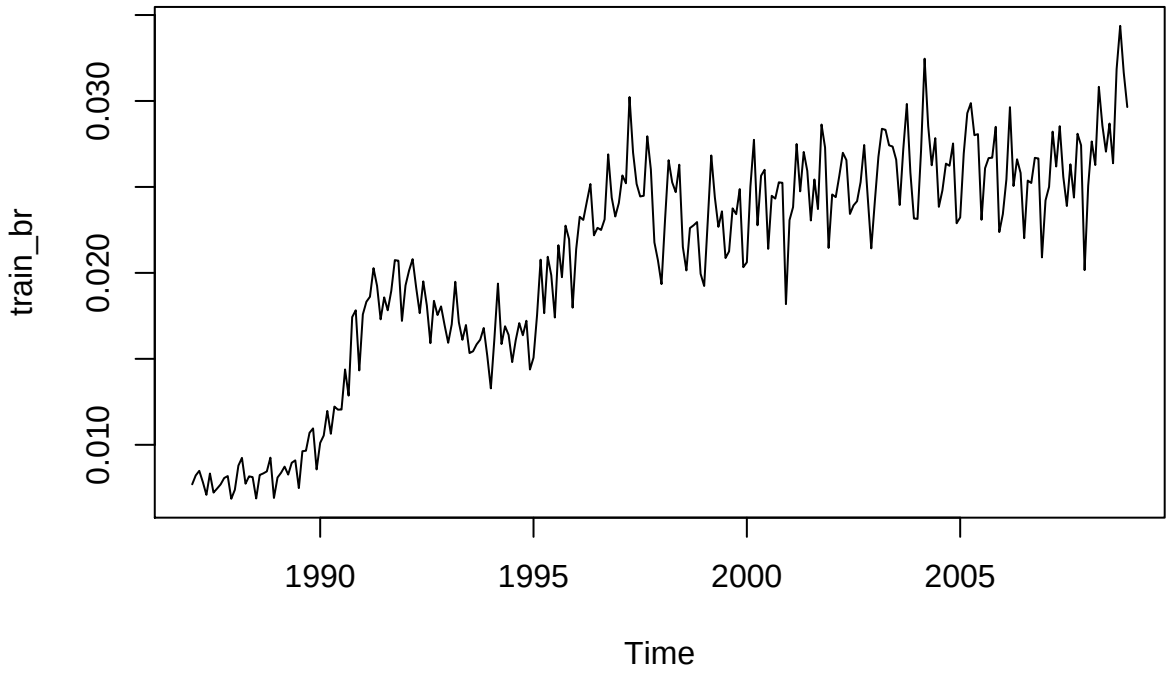
```
## Loading required package: urca
```

```
## Loading required package: lmtest
```

```
substrRight <- function(x, n){
  substr(x, nchar(x)-n+1, nchar(x))
}

substrFromRight <- function(x, n){
  substr(x, 1, nchar(x)-n)
}


rmse <- function(true, test, log=FALSE){
  if(mode(test)=="list"){
    ts <- test$pred
  }else if (mode(test)=='numeric'){
    ts <- test
  }

  if (log){
    ts <- exp(ts)
  }
  mse <- sqrt(mean( (true - ts )^2 ) )

  return(mse)
}
```

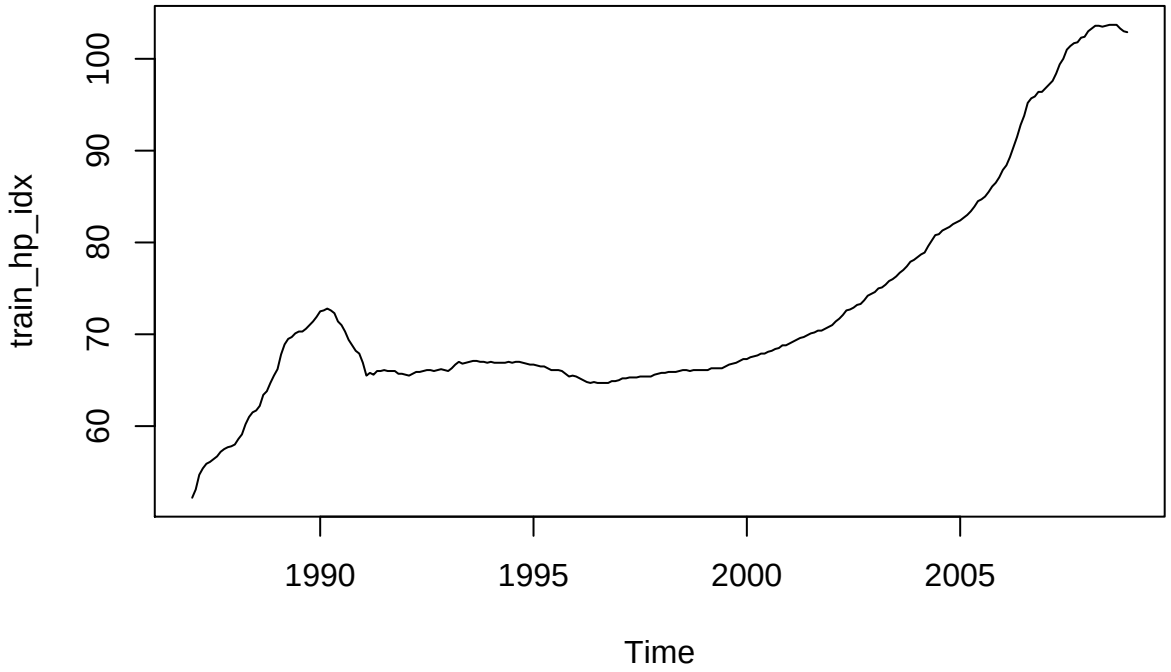
```
train <- read.csv('train.csv')
train['year'] <- substrRight(train$Month, 4)
train['month'] <- substrFromRight(train$Month, 4)
```

```r
trainees <- ts(train$Bankruptcy_Rate, start = 1987, frequency = 12)
unemployment <- ts(train$Unemployment_Rate, start = 1987, frequency = 12)
house_price_index <- ts(train$House_Price_Index, start = 1987, frequency = 12)
train_br <- window(trainees, 1987, c(1987, 22*12))
train_unemp <- window(unemployment, 1987, c(1987, 22*12))
train_hp_idx <- window(house_price_index, 1987, c(1987, 22*12))

test_br <- window(trainees, 2009, c(2009, 2*12))
test_unemp <- window(unemployment, 2009, c(2009, 2*12))
test_hp_idx <- window(house_price_index, 2009, c(2009, 2*12))
plot(train_br)
```

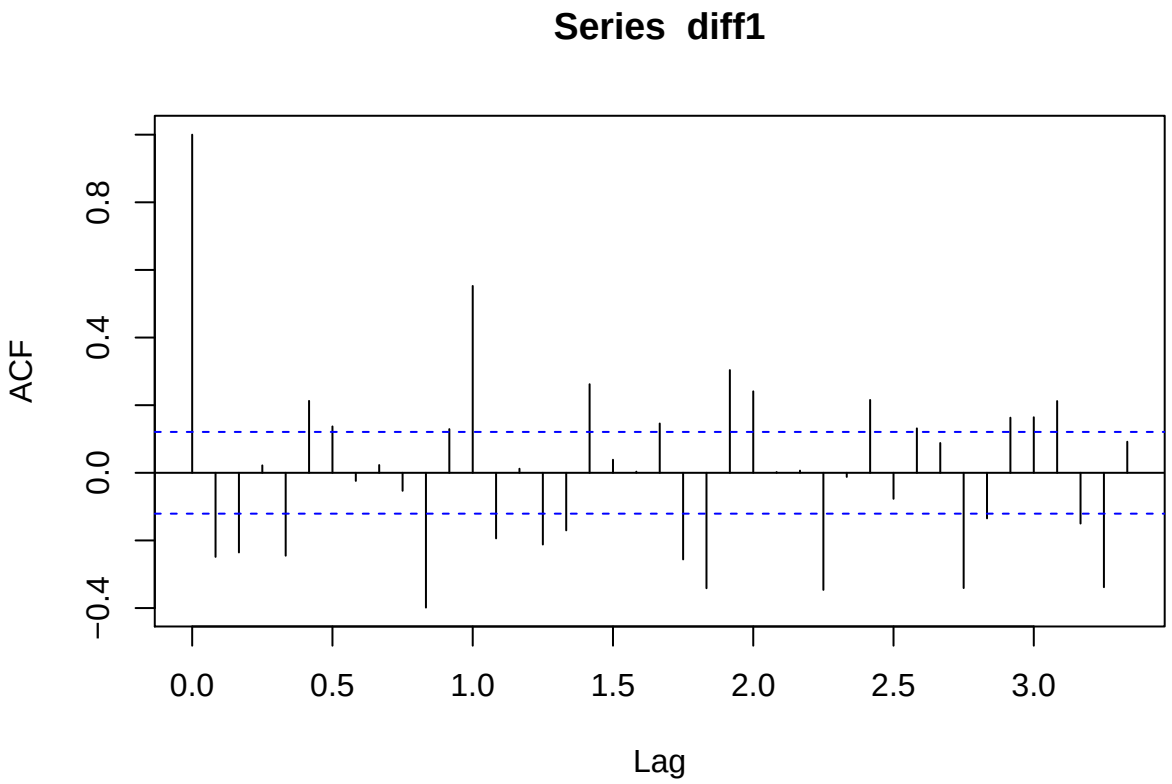



```r
plot(train_unemp)
```

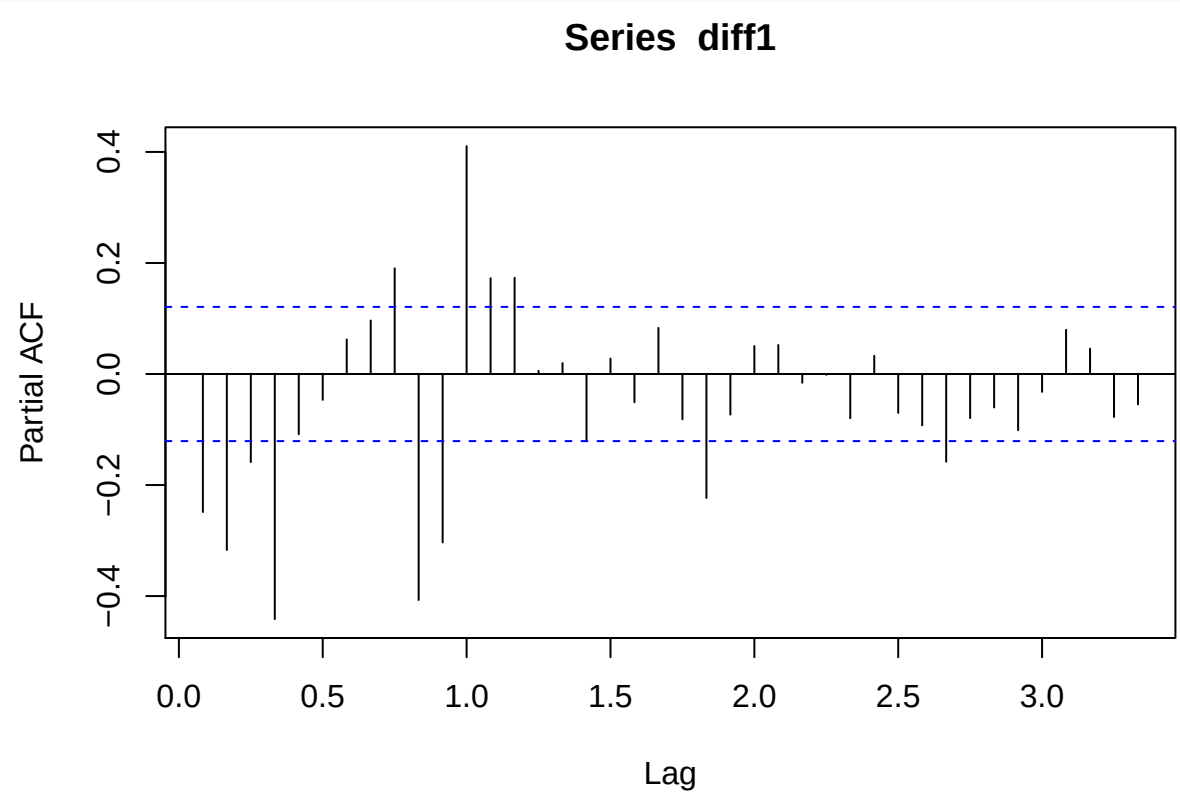

```
plot(train_hp_idx)
```




```
diff1 <- diff(train_br)
acf(diff1, lag.max = 40)
```




```
pacf(diff1, lag.max = 40)
```




```
rmses <- c()
for (i in 1:4){
```

```
  for (j in 1:4){
    for(k in 1:4){
      for(m in 1:4){
        try(
          temp_model <- arima(train_br, order = c(i,1,j),
                              seasonal = list(order = c(k,1,m), period=12),
                              xreg = data.frame(train_hp_idx, train_unemp), method = 'CSS')
        )
          print(c(rmse(test_br, predict(temp_model, n.ahead = 24,
                                        newxreg = data.frame(test_hp_idx, test_unemp))),i, j, k, m ))
          rmses <- c(rmses,rmse(test_br, predict(temp_model, n.ahead = 24,
                                        newxreg = data.frame(test_hp_idx, test_unemp))) )
      }
    }
  }
}
```

```
## [1] 0.004661236 1.000000000 1.000000000 1.000000000 1.000000000
## [1] 0.004542752 1.000000000 1.000000000 1.000000000 2.000000000
## [1] 0.004609765 1.000000000 1.000000000 1.000000000 3.000000000
## [1] 0.004071499 1.000000000 1.000000000 1.000000000 4.000000000
## [1] 0.004314708 1.000000000 1.000000000 2.000000000 1.000000000
## [1] 0.004733211 1.000000000 1.000000000 2.000000000 2.000000000
## [1] 0.003309814 1.000000000 1.000000000 2.000000000 3.000000000
## [1] 0.004869807 1.000000000 1.000000000 2.000000000 4.000000000
## [1] 0.004651137 1.000000000 1.000000000 3.000000000 1.000000000
## [1] 0.005061632 1.000000000 1.000000000 3.000000000 2.000000000
## [1] 0.00592552 1.00000000 1.00000000 3.00000000 3.00000000
## [1] 0.003470587 1.000000000 1.000000000 3.000000000 4.000000000
## [1] 0.003764689 1.000000000 1.000000000 4.000000000 1.000000000
## [1] 0.003864092 1.000000000 1.000000000 4.000000000 2.000000000
## [1] 0.003859873 1.000000000 1.000000000 4.000000000 3.000000000
## [1] 0.003868823 1.000000000 1.000000000 4.000000000 4.000000000
## [1] 0.005051271 1.000000000 2.000000000 1.000000000 1.000000000
## [1] 0.005036099 1.000000000 2.000000000 1.000000000 2.000000000
## [1] 0.005128814 1.000000000 2.000000000 1.000000000 3.000000000
## [1] 0.0046171 1.0000000 2.0000000 1.0000000 4.0000000
## [1] 0.005910355 1.000000000 2.000000000 2.000000000 1.000000000
## [1] 0.005989597 1.000000000 2.000000000 2.000000000 2.000000000
## [1] 0.005228532 1.000000000 2.000000000 2.000000000 3.000000000
## [1] 0.005253979 1.000000000 2.000000000 2.000000000 4.000000000
## [1] 0.004098835 1.000000000 2.000000000 3.000000000 1.000000000
## [1] 0.004674069 1.000000000 2.000000000 3.000000000 2.000000000
## [1] 0.005598925 1.000000000 2.000000000 3.000000000 3.000000000
## [1] 0.005010019 1.000000000 2.000000000 3.000000000 4.000000000
## [1] 0.003782138 1.000000000 2.000000000 4.000000000 1.000000000
## [1] 0.00389809 1.00000000 2.00000000 4.00000000 2.00000000
## [1] 0.003899258 1.000000000 2.000000000 4.000000000 3.000000000
## [1] 0.003900833 1.000000000 2.000000000 4.000000000 4.000000000
## [1] 0.005156427 1.000000000 3.000000000 1.000000000 1.000000000
## [1] 0.005026568 1.000000000 3.000000000 1.000000000 2.000000000
## [1] 0.004478433 1.000000000 3.000000000 1.000000000 3.000000000
## [1] 0.004595799 1.000000000 3.000000000 1.000000000 4.000000000
## [1] 0.005022843 1.000000000 3.000000000 2.000000000 1.000000000
```

```
## [1] 0.004989287 1.000000000 3.000000000 2.000000000 2.000000000
## [1] 0.003260932 1.000000000 3.000000000 2.000000000 3.000000000
## [1] 0.005242189 1.000000000 3.000000000 2.000000000 4.000000000
## [1] 0.005046002 1.000000000 3.000000000 3.000000000 1.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.004705455 1.000000000 3.000000000 3.000000000 2.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.006052003 1.000000000 3.000000000 3.000000000 3.000000000
## [1] 0.003757128 1.000000000 3.000000000 3.000000000 4.000000000
## [1] 0.003895804 1.000000000 3.000000000 4.000000000 1.000000000
## [1] 0.004042023 1.000000000 3.000000000 4.000000000 2.000000000
## [1] 0.004036464 1.000000000 3.000000000 4.000000000 3.000000000
## [1] 0.003999691 1.000000000 3.000000000 4.000000000 4.000000000
## [1] 0.004930568 1.000000000 4.000000000 1.000000000 1.000000000
## [1] 0.004834496 1.000000000 4.000000000 1.000000000 2.000000000
## [1] 0.004201345 1.000000000 4.000000000 1.000000000 3.000000000
## [1] 0.004286165 1.000000000 4.000000000 1.000000000 4.000000000
## [1] 0.004787089 1.000000000 4.000000000 2.000000000 1.000000000
## [1] 0.004796368 1.000000000 4.000000000 2.000000000 2.000000000
## [1] 0.003272609 1.000000000 4.000000000 2.000000000 3.000000000
## [1] 0.005025094 1.000000000 4.000000000 2.000000000 4.000000000
## [1] 0.004881221 1.000000000 4.000000000 3.000000000 1.000000000
## [1] 0.005558805 1.000000000 4.000000000 3.000000000 2.000000000
## [1] 0.005915466 1.000000000 4.000000000 3.000000000 3.000000000
## [1] 0.003754402 1.000000000 4.000000000 3.000000000 4.000000000
## [1] 0.003822334 1.000000000 4.000000000 4.000000000 1.000000000
## [1] 0.00406975 1.00000000 4.00000000 4.00000000 2.00000000
## [1] 0.004066263 1.000000000 4.000000000 4.000000000 3.000000000
## [1] 0.004026371 1.000000000 4.000000000 4.000000000 4.000000000
## [1] 0.004891685 2.000000000 1.000000000 1.000000000 1.000000000
## [1] 0.004858127 2.000000000 1.000000000 1.000000000 2.000000000
## [1] 0.004247809 2.000000000 1.000000000 1.000000000 3.000000000
## [1] 0.004366854 2.000000000 1.000000000 1.000000000 4.000000000
## [1] 0.004667407 2.000000000 1.000000000 2.000000000 1.000000000
## [1] 0.004675361 2.000000000 1.000000000 2.000000000 2.000000000
## [1] 0.003568134 2.000000000 1.000000000 2.000000000 3.000000000
## [1] 0.004885541 2.000000000 1.000000000 2.000000000 4.000000000
## [1] 0.005153931 2.000000000 1.000000000 3.000000000 1.000000000
## [1] 0.005171655 2.000000000 1.000000000 3.000000000 2.000000000
## [1] 0.005901834 2.000000000 1.000000000 3.000000000 3.000000000
## [1] 0.004274278 2.000000000 1.000000000 3.000000000 4.000000000
## [1] 0.00390817 2.00000000 1.00000000 4.00000000 1.00000000
## [1] 0.00401735 2.00000000 1.00000000 4.00000000 2.00000000
## [1] 0.004001397 2.000000000 1.000000000 4.000000000 3.000000000
## [1] 0.003897492 2.000000000 1.000000000 4.000000000 4.000000000
## [1] 0.004860925 2.000000000 2.000000000 1.000000000 1.000000000
## [1] 0.004778791 2.000000000 2.000000000 1.000000000 2.000000000
## [1] 0.004204817 2.000000000 2.000000000 1.000000000 3.000000000
## [1] 0.004321172 2.000000000 2.000000000 1.000000000 4.000000000
## [1] 0.004553245 2.000000000 2.000000000 2.000000000 1.000000000
```

```
## [1] 0.004454302 2.000000000 2.000000000 2.000000000 2.000000000
## [1] 0.00337025 2.00000000 2.00000000 2.00000000 3.00000000
## [1] 0.004856838 2.000000000 2.000000000 2.000000000 4.000000000
## [1] 0.005074064 2.000000000 2.000000000 3.000000000 1.000000000
## [1] 0.005081858 2.000000000 2.000000000 3.000000000 2.000000000
## [1] 0.0058199 2.0000000 2.0000000 3.0000000 3.0000000
## [1] 0.00405364 2.00000000 2.00000000 3.00000000 4.00000000
## [1] 0.003910621 2.000000000 2.000000000 4.000000000 1.000000000
## [1] 0.003974493 2.000000000 2.000000000 4.000000000 2.000000000
## [1] 0.003969633 2.000000000 2.000000000 4.000000000 3.000000000
## [1] 0.003904952 2.000000000 2.000000000 4.000000000 4.000000000
## [1] 0.004544977 2.000000000 3.000000000 1.000000000 1.000000000
## [1] 0.004561993 2.000000000 3.000000000 1.000000000 2.000000000
## [1] 0.004594979 2.000000000 3.000000000 1.000000000 3.000000000
## [1] 0.004331033 2.000000000 3.000000000 1.000000000 4.000000000
## [1] 0.005112748 2.000000000 3.000000000 2.000000000 1.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005411574 2.000000000 3.000000000 2.000000000 2.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible


## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.004613707 2.000000000 3.000000000 2.000000000 3.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible


## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005439388 2.000000000 3.000000000 2.000000000 4.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005184181 2.000000000 3.000000000 3.000000000 1.000000000
## [1] 0.004842447 2.000000000 3.000000000 3.000000000 2.000000000

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : possible convergence problem: optim gave code = 1


## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : MA part of model is not invertible

## [1] 0.004664072 2.000000000 3.000000000 3.000000000 3.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.004145986 2.000000000 3.000000000 3.000000000 4.000000000
## [1] 0.003934883 2.000000000 3.000000000 4.000000000 1.000000000
## [1] 0.003974021 2.000000000 3.000000000 4.000000000 2.000000000
## [1] 0.003968083 2.000000000 3.000000000 4.000000000 3.000000000
## [1] 0.004134613 2.000000000 3.000000000 4.000000000 4.000000000
```

```
## [1] 0.004640352 2.000000000 4.000000000 1.000000000 1.000000000
## [1] 0.004619103 2.000000000 4.000000000 1.000000000 2.000000000
## [1] 0.004639958 2.000000000 4.000000000 1.000000000 3.000000000
## [1] 0.004358391 2.000000000 4.000000000 1.000000000 4.000000000
## [1] 0.005201228 2.000000000 4.000000000 2.000000000 1.000000000
## [1] 0.004826927 2.000000000 4.000000000 2.000000000 2.000000000
## [1] 0.003350402 2.000000000 4.000000000 2.000000000 3.000000000
## [1] 0.004569758 2.000000000 4.000000000 2.000000000 4.000000000
## [1] 0.005271765 2.000000000 4.000000000 3.000000000 1.000000000
## [1] 0.00387353 2.00000000 4.00000000 3.00000000 2.00000000
## [1] 0.003817815 2.000000000 4.000000000 3.000000000 3.000000000

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : possible convergence problem: optim gave code = 1

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : MA part of model is not invertible

## [1] 0.004476854 2.000000000 4.000000000 3.000000000 4.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.003839449 2.000000000 4.000000000 4.000000000 1.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : possible convergence problem: optim gave code = 1

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.00370732 2.00000000 4.00000000 4.00000000 2.00000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : possible convergence problem: optim gave code = 1

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.003688347 2.000000000 4.000000000 4.000000000 3.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.003747176 2.000000000 4.000000000 4.000000000 4.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible
```

```
## [1] 0.004899759 3.000000000 1.000000000 1.000000000 1.000000000
## [1] 0.004861558 3.000000000 1.000000000 1.000000000 2.000000000
## [1] 0.004227852 3.000000000 1.000000000 1.000000000 3.000000000
## [1] 0.004342917 3.000000000 1.000000000 1.000000000 4.000000000
## [1] 0.004670987 3.000000000 1.000000000 2.000000000 1.000000000
## [1] 0.005258727 3.000000000 1.000000000 2.000000000 2.000000000
## [1] 0.00385613 3.00000000 1.00000000 2.00000000 3.00000000
## [1] 0.005320564 3.000000000 1.000000000 2.000000000 4.000000000
## [1] 0.00521062 3.00000000 1.00000000 3.00000000 1.00000000
## [1] 0.005127142 3.000000000 1.000000000 3.000000000 2.000000000
## [1] 0.005901172 3.000000000 1.000000000 3.000000000 3.000000000
## [1] 0.004235621 3.000000000 1.000000000 3.000000000 4.000000000
## [1] 0.003629101 3.000000000 1.000000000 4.000000000 1.000000000
## [1] 0.004001145 3.000000000 1.000000000 4.000000000 2.000000000
## [1] 0.003990664 3.000000000 1.000000000 4.000000000 3.000000000
## [1] 0.004140883 3.000000000 1.000000000 4.000000000 4.000000000
## [1] 0.004752614 3.000000000 2.000000000 1.000000000 1.000000000
## [1] 0.004660789 3.000000000 2.000000000 1.000000000 2.000000000
## [1] 0.004078144 3.000000000 2.000000000 1.000000000 3.000000000
## [1] 0.004138152 3.000000000 2.000000000 1.000000000 4.000000000
## [1] 0.004497981 3.000000000 2.000000000 2.000000000 1.000000000
## [1] 0.005132884 3.000000000 2.000000000 2.000000000 2.000000000
## [1] 0.003722984 3.000000000 2.000000000 2.000000000 3.000000000
## [1] 0.004689616 3.000000000 2.000000000 2.000000000 4.000000000
## [1] 0.005061305 3.000000000 2.000000000 3.000000000 1.000000000
## [1] 0.005057409 3.000000000 2.000000000 3.000000000 2.000000000
## [1] 0.005854648 3.000000000 2.000000000 3.000000000 3.000000000
## [1] 0.003908542 3.000000000 2.000000000 3.000000000 4.000000000
## [1] 0.003599102 3.000000000 2.000000000 4.000000000 1.000000000
## [1] 0.004003508 3.000000000 2.000000000 4.000000000 2.000000000
## [1] 0.004008186 3.000000000 2.000000000 4.000000000 3.000000000
## [1] 0.00411648 3.00000000 2.00000000 4.00000000 4.00000000
## [1] 0.004547771 3.000000000 3.000000000 1.000000000 1.000000000
## [1] 0.005860898 3.000000000 3.000000000 1.000000000 2.000000000
## [1] 0.006064065 3.000000000 3.000000000 1.000000000 3.000000000
## [1] 0.004331134 3.000000000 3.000000000 1.000000000 4.000000000
## [1] 0.005088043 3.000000000 3.000000000 2.000000000 1.000000000
## [1] 0.005330097 3.000000000 3.000000000 2.000000000 2.000000000
## [1] 0.003893774 3.000000000 3.000000000 2.000000000 3.000000000
## [1] 0.004645566 3.000000000 3.000000000 2.000000000 4.000000000
## [1] 0.005460724 3.000000000 3.000000000 3.000000000 1.000000000
## [1] 0.004712099 3.000000000 3.000000000 3.000000000 2.000000000

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : possible convergence problem: optim gave code = 1

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : MA part of model is not invertible

## [1] 0.004825764 3.000000000 3.000000000 3.000000000 3.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
```

```
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.00349815 3.00000000 3.00000000 3.00000000 4.00000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.003569085 3.000000000 3.000000000 4.000000000 1.000000000
## [1] 0.003844716 3.000000000 3.000000000 4.000000000 2.000000000
## [1] 0.003837109 3.000000000 3.000000000 4.000000000 3.000000000
## [1] 0.004063415 3.000000000 3.000000000 4.000000000 4.000000000
## [1] 0.006949112 3.000000000 4.000000000 1.000000000 1.000000000
## [1] 0.007066634 3.000000000 4.000000000 1.000000000 2.000000000
## [1] 0.007063263 3.000000000 4.000000000 1.000000000 3.000000000
## [1] 0.00738557 3.00000000 4.00000000 1.00000000 4.00000000
## [1] 0.006563994 3.000000000 4.000000000 2.000000000 1.000000000
## [1] 0.006753176 3.000000000 4.000000000 2.000000000 2.000000000
## [1] 0.006749464 3.000000000 4.000000000 2.000000000 3.000000000
## [1] 0.005540913 3.000000000 4.000000000 2.000000000 4.000000000
## [1] 0.008282714 3.000000000 4.000000000 3.000000000 1.000000000
## [1] 0.004118525 3.000000000 4.000000000 3.000000000 2.000000000

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : possible convergence problem: optim gave code = 1

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : MA part of model is not invertible

## [1] 0.0062833 3.0000000 4.0000000 3.0000000 3.0000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.0057094 3.0000000 4.0000000 3.0000000 4.0000000
## [1] 0.003566179 3.000000000 4.000000000 4.000000000 1.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.004676106 3.000000000 4.000000000 4.000000000 2.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.00477144 3.00000000 4.00000000 4.00000000 3.00000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.00398873 3.00000000 4.00000000 4.00000000 4.00000000
## [1] 0.004677125 4.000000000 1.000000000 1.000000000 1.000000000
## [1] 0.004564254 4.000000000 1.000000000 1.000000000 2.000000000
## [1] 0.004082655 4.000000000 1.000000000 1.000000000 3.000000000
## [1] 0.004144832 4.000000000 1.000000000 1.000000000 4.000000000
## [1] 0.004458979 4.000000000 1.000000000 2.000000000 1.000000000
## [1] 0.005329048 4.000000000 1.000000000 2.000000000 2.000000000
## [1] 0.003845679 4.000000000 1.000000000 2.000000000 3.000000000
```

```
## [1] 0.004596064 4.000000000 1.000000000 2.000000000 4.000000000
## [1] 0.004803543 4.000000000 1.000000000 3.000000000 1.000000000
## [1] 0.004857781 4.000000000 1.000000000 3.000000000 2.000000000
## [1] 0.005605841 4.000000000 1.000000000 3.000000000 3.000000000
## [1] 0.003881157 4.000000000 1.000000000 3.000000000 4.000000000
## [1] 0.003995131 4.000000000 1.000000000 4.000000000 1.000000000
## [1] 0.004000673 4.000000000 1.000000000 4.000000000 2.000000000
## [1] 0.004004545 4.000000000 1.000000000 4.000000000 3.000000000
## [1] 0.004234143 4.000000000 1.000000000 4.000000000 4.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005954148 4.000000000 2.000000000 1.000000000 1.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible


## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005433741 4.000000000 2.000000000 1.000000000 2.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible


## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005251137 4.000000000 2.000000000 1.000000000 3.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible


## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005023266 4.000000000 2.000000000 1.000000000 4.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.004628739 4.000000000 2.000000000 2.000000000 1.000000000
## [1] 0.005028962 4.000000000 2.000000000 2.000000000 2.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.004799309 4.000000000 2.000000000 2.000000000 3.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : possible convergence problem: optim gave code = 1

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.004684989 4.000000000 2.000000000 2.000000000 4.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
```

```
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005149849 4.000000000 2.000000000 3.000000000 1.000000000
## [1] 0.00499789 4.00000000 2.00000000 3.00000000 2.00000000
## [1] 0.005801251 4.000000000 2.000000000 3.000000000 3.000000000
## [1] 0.00419582 4.00000000 2.00000000 3.00000000 4.00000000
## [1] 0.003675497 4.000000000 2.000000000 4.000000000 1.000000000
## [1] 0.004081778 4.000000000 2.000000000 4.000000000 2.000000000
## [1] 0.004038973 4.000000000 2.000000000 4.000000000 3.000000000
## [1] 0.004359694 4.000000000 2.000000000 4.000000000 4.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.0047454 4.0000000 3.0000000 1.0000000 1.0000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005018053 4.000000000 3.000000000 1.000000000 2.000000000
## [1] 0.005138701 4.000000000 3.000000000 1.000000000 3.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.00432335 4.00000000 3.00000000 1.00000000 4.00000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005592962 4.000000000 3.000000000 2.000000000 1.000000000
## [1] 0.005021176 4.000000000 3.000000000 2.000000000 2.000000000
## [1] 0.004315733 4.000000000 3.000000000 2.000000000 3.000000000

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : possible convergence problem: optim gave code = 1

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : MA part of model is not invertible

## [1] 0.004637289 4.000000000 3.000000000 2.000000000 4.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005696701 4.000000000 3.000000000 3.000000000 1.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005694798 4.000000000 3.000000000 3.000000000 2.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.005821619 4.000000000 3.000000000 3.000000000 3.000000000
```

```
## [1] 0.004164348 4.000000000 3.000000000 3.000000000 4.000000000
## [1] 0.004017018 4.000000000 3.000000000 4.000000000 1.000000000
## [1] 0.004052028 4.000000000 3.000000000 4.000000000 2.000000000
## [1] 0.004017921 4.000000000 3.000000000 4.000000000 3.000000000

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : possible convergence problem: optim gave code = 1

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : MA part of model is not invertible

## [1] 0.004365399 4.000000000 3.000000000 4.000000000 4.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.004665854 4.000000000 4.000000000 1.000000000 1.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.00496089 4.00000000 4.00000000 1.00000000 2.00000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.004745626 4.000000000 4.000000000 1.000000000 3.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.003847044 4.000000000 4.000000000 1.000000000 4.000000000

## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible

## [1] 0.007271701 4.000000000 4.000000000 2.000000000 1.000000000
## [1] 0.007539473 4.000000000 4.000000000 2.000000000 2.000000000
## [1] 0.007463918 4.000000000 4.000000000 2.000000000 3.000000000
## [1] 0.006233083 4.000000000 4.000000000 2.000000000 4.000000000
## [1] 0.007611977 4.000000000 4.000000000 3.000000000 1.000000000
## [1] 0.00475171 4.00000000 4.00000000 3.00000000 2.00000000

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : possible convergence problem: optim gave code = 1

## Warning in arima(train_br, order = c(i, 1, j), seasonal = list(order =
## c(k, : MA part of model is not invertible
```

```
## [1] 0.006496876 4.000000000 4.000000000 3.000000000 3.000000000
```

```
## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible
```

```
## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible
```

```
## [1] 0.003873152 4.000000000 4.000000000 3.000000000 4.000000000
```

```
## Warning in predict.Arima(temp_model, n.ahead = 24, newxreg =
## data.frame(test_hp_idx, : MA part of model is not invertible
```

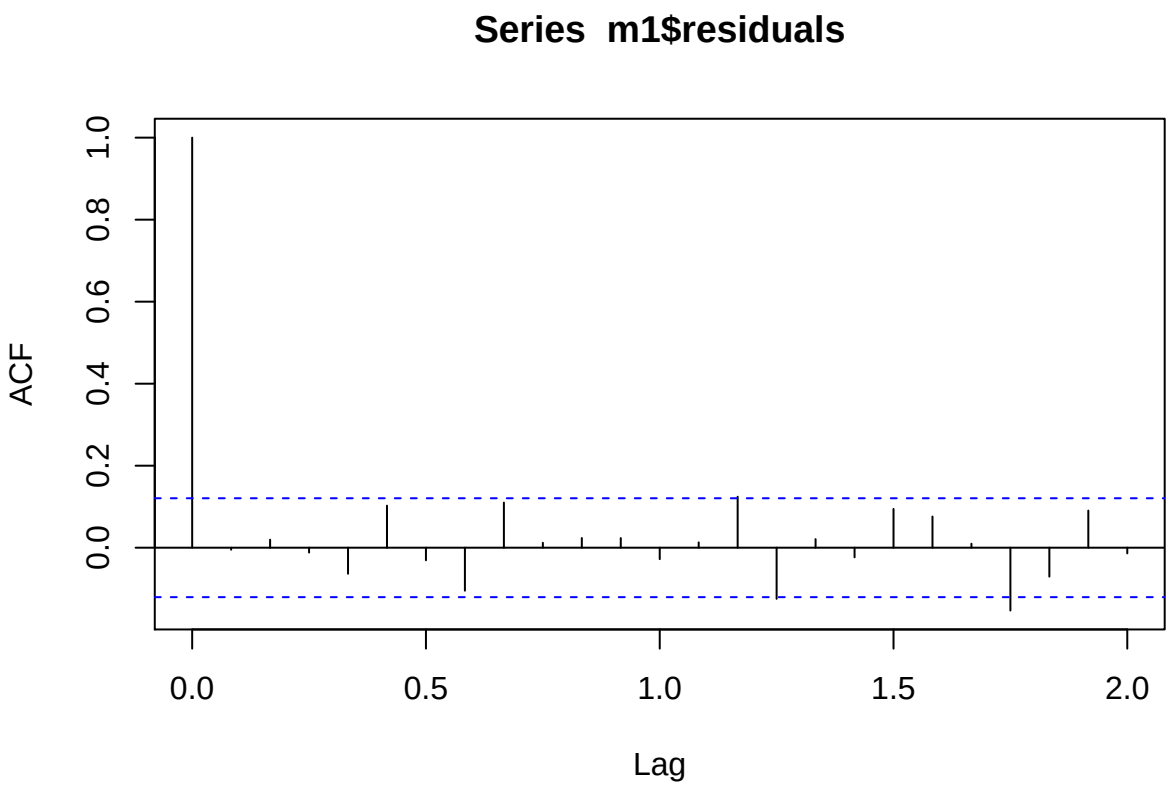
```
## [1] 0.004228405 4.000000000 4.000000000 4.000000000 1.000000000
## [1] 0.004278597 4.000000000 4.000000000 4.000000000 2.000000000
## [1] 0.004200167 4.000000000 4.000000000 4.000000000 3.000000000
## [1] 0.004447344 4.000000000 4.000000000 4.000000000 4.000000000
```

```
rmses[which.min(rmses)]
```

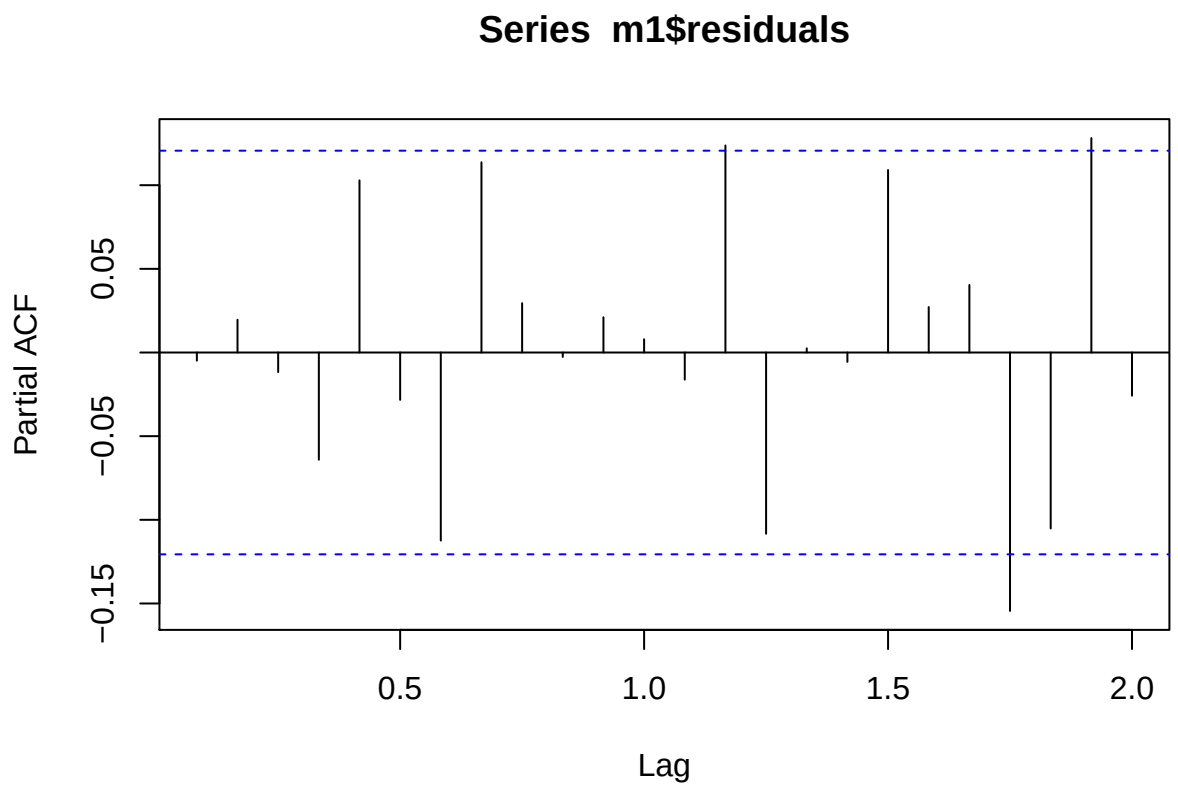
```
## [1] 0.003260932
```

Fit a SARIMAX

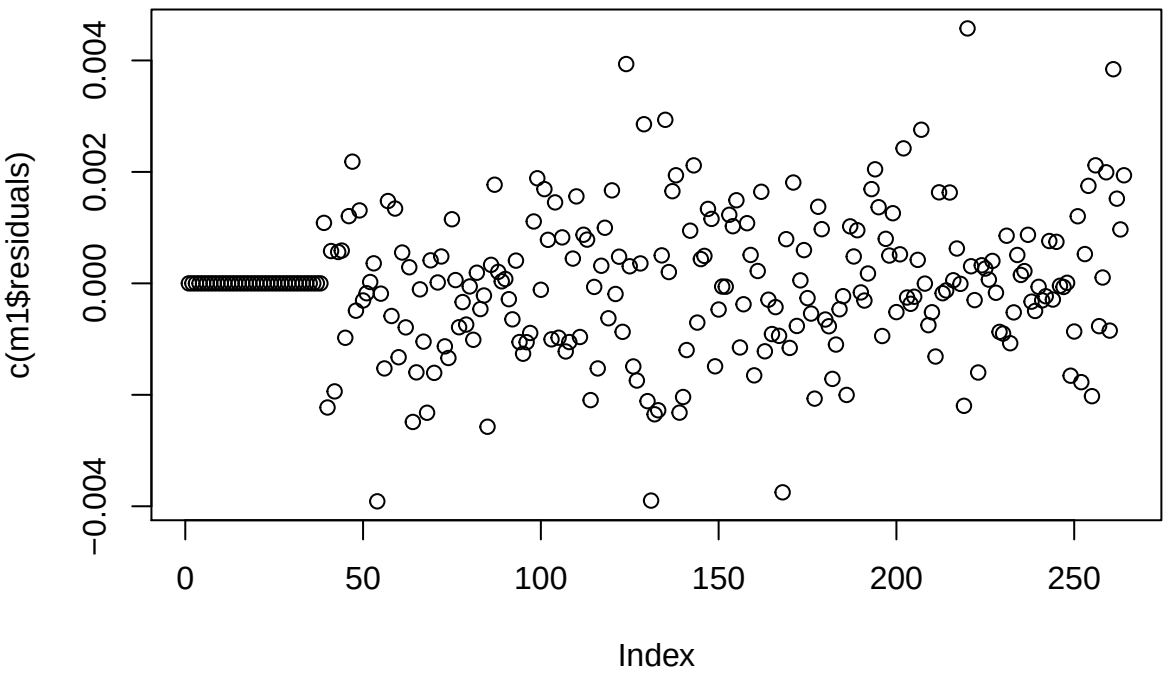
```
m1 <- arima(train_br, order = c(1,1,3),
                              seasonal = list(order = c(2,1,3), period = 12),
                              xreg = data.frame(train_hp_idx, train_unemp), method = 'CSS')

acf(m1$residuals)
```



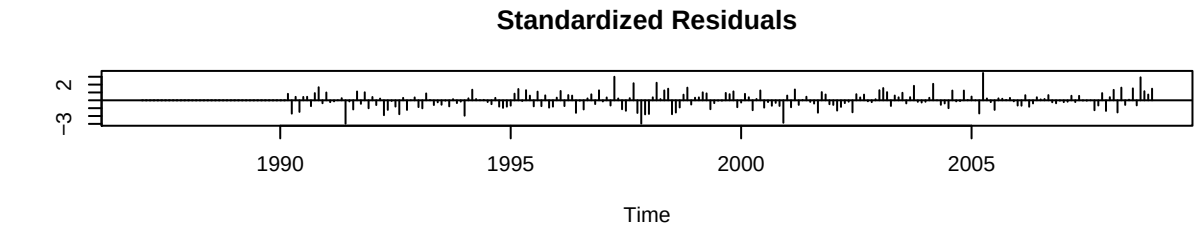
```
pacf(m1$residuals)
```

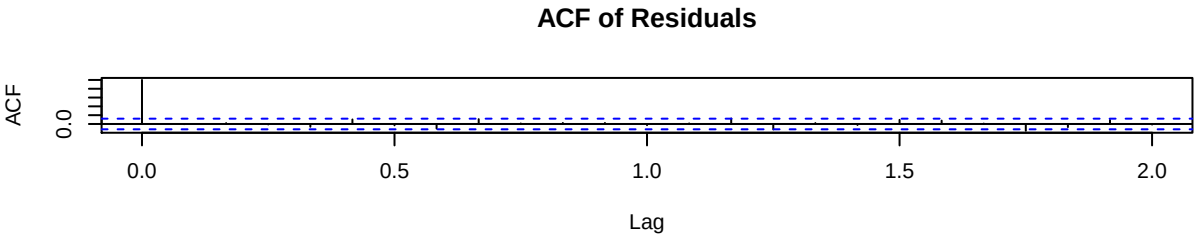


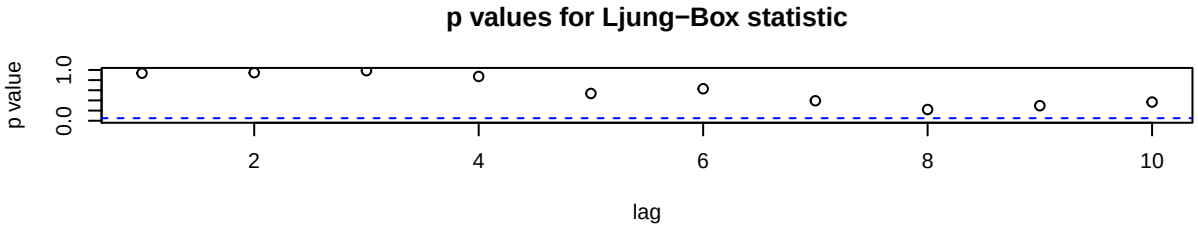

```
plot(c(m1$residuals))
```



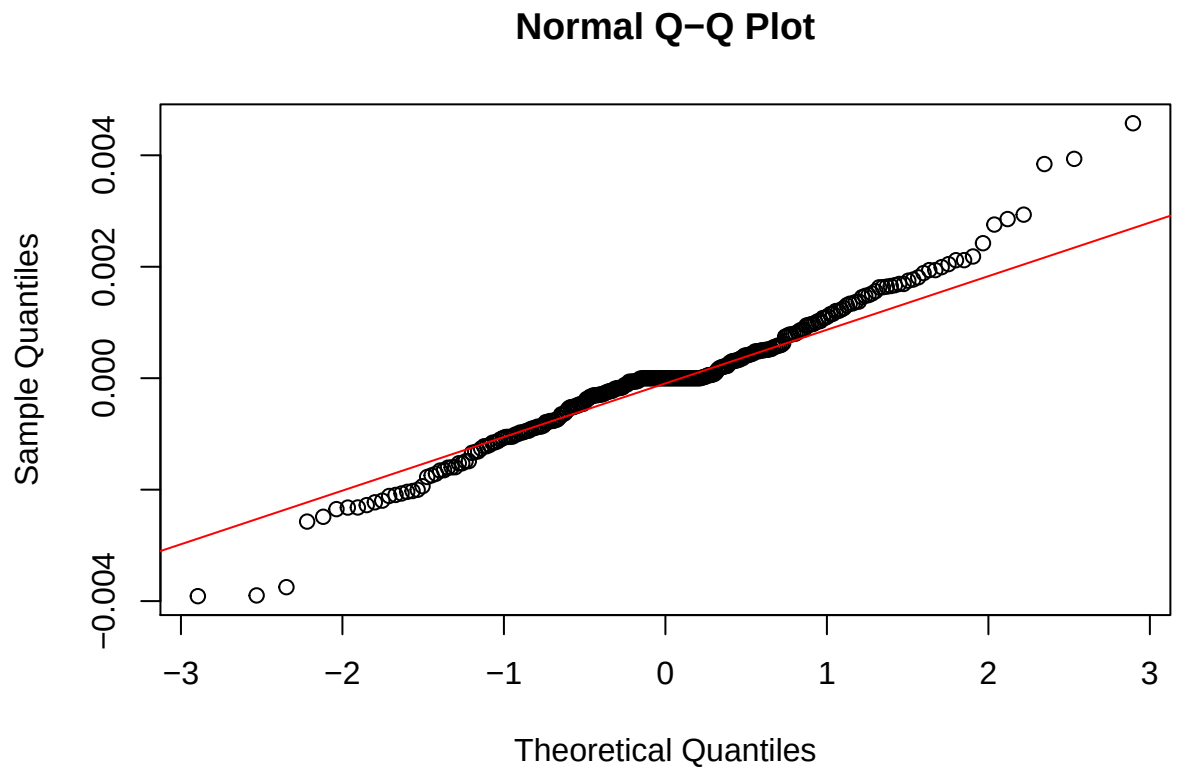

```
tsdiag(m1)
```








```
qqnorm(m1$residuals)
qqline(m1$residuals, col= "red")
```



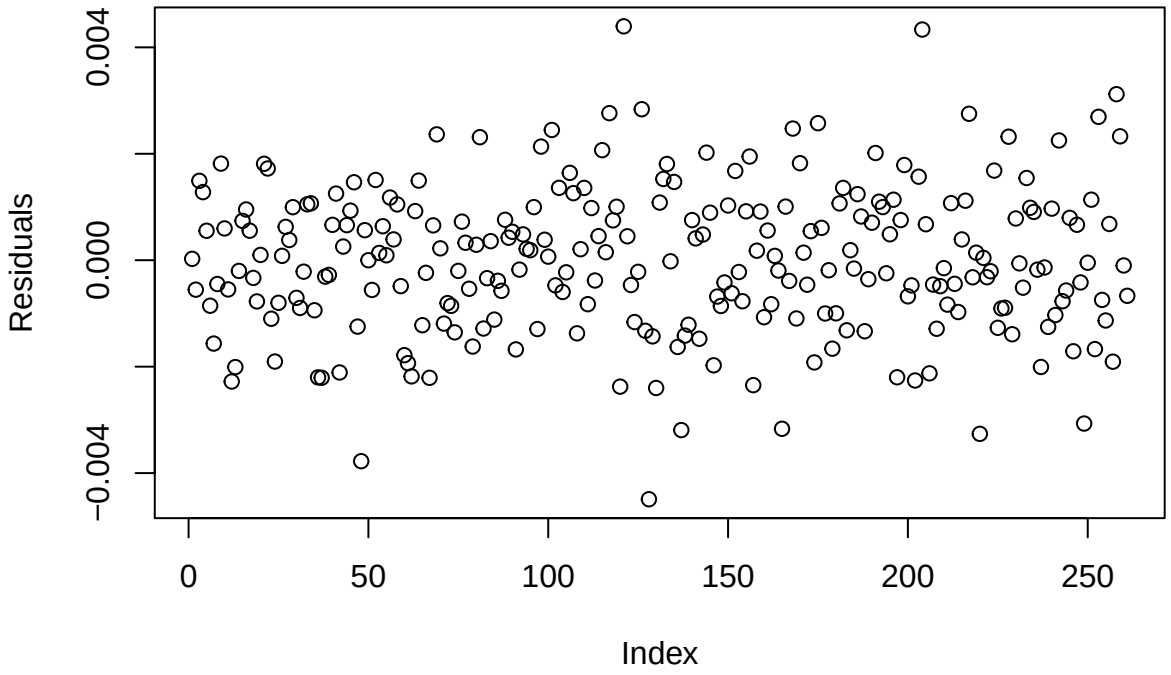
VAR

```
VARselect(y = data.frame(train_br, train_hp_idx, train_unemp), season = 12)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      6      3      3      6
##
## $criteria
##                   1             2             3             4
## AIC(n) -1.785961e+01 -1.867599e+01 -1.898483e+01 -1.895269e+01
## HQ(n)  -1.760750e+01 -1.837345e+01 -1.863188e+01 -1.854931e+01
## SC(n)  -1.723292e+01 -1.792396e+01 -1.810746e+01 -1.794998e+01
## FPE(n)  1.753274e-08  7.752325e-09  5.694887e-09  5.884227e-09
##                   5             6             7             8
## AIC(n) -1.901273e+01 -1.901484e+01 -1.896815e+01 -1.892588e+01
## HQ(n)  -1.855893e+01 -1.851062e+01 -1.841350e+01 -1.832081e+01
## SC(n)  -1.788468e+01 -1.776145e+01 -1.758942e+01 -1.742182e+01
## FPE(n)  5.545344e-09  5.538634e-09  5.809829e-09  6.068737e-09
##                   9            10
## AIC(n) -1.891718e+01 -1.892722e+01
## HQ(n)  -1.826169e+01 -1.822131e+01
## SC(n)  -1.728778e+01 -1.717248e+01
## FPE(n)  6.131453e-09  6.081415e-09
```

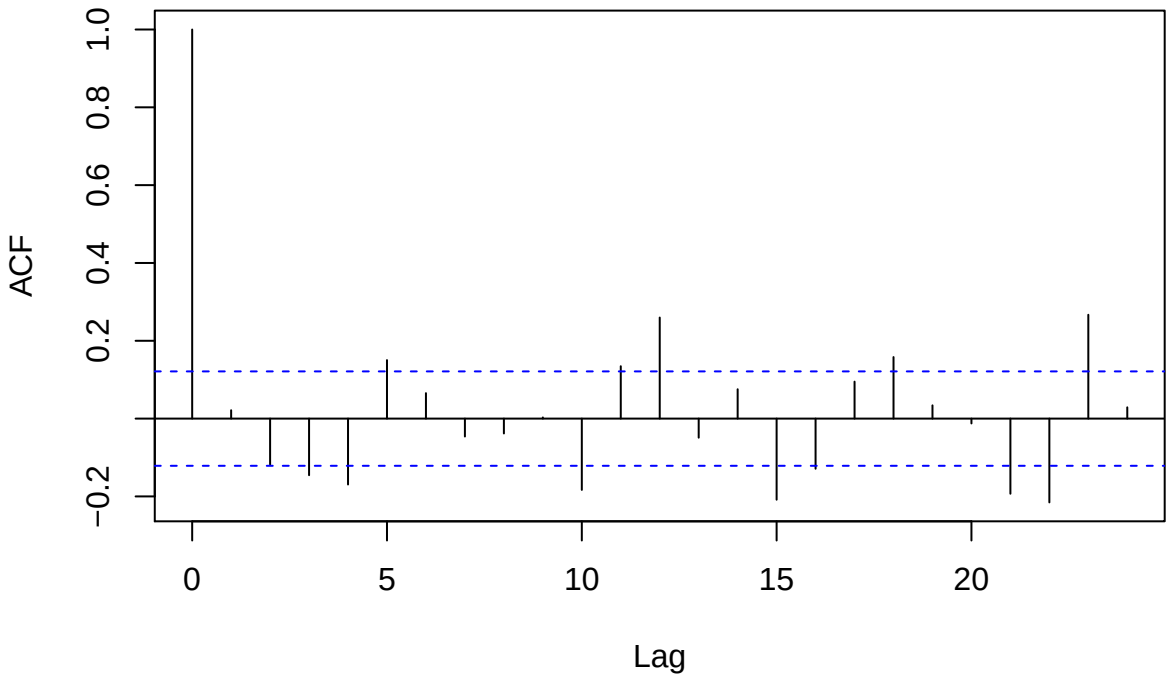
```
var.m1 <- VAR(y = data.frame(train_br, train_hp_idx, train_unemp), p = 3, season = 12)
pred.vals <- predict(var.m1, n.ahead = 24)$fcst$train_br
rmse.var <- rmse(test_br,pred.vals[,1])

e <- residuals(var.m1)
plot(e[,1], ylab='Residuals', main='Figure 12: Residual Plot')
```

Figure 12: Residual Plot




```
acf(e[,1], main='Figure 13: ACF Plot')
```

Figure 13: ACF Plot




```
pacf(e[,1], main='Figure 14: PACF plot')
```

Figure 14: PACF plot