Lab3 Q2

Michelle Kim, Sue Yang, Legg Yeung

August 2, 2017

```
# copied
library(moments)
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.3.3
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.3.3
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 3.3.3
```

```
library(effects)
```

```
## Warning: package 'effects' was built under R version 3.3.3
```

```
unem.data = read.csv("UNRATENSA.csv", header = T)
auto.data = read.csv("TOTALNSA.csv", header = T)
```

Question 2: SARIMA

It is Dec 31, 2016 and you work for a non-partisan think tank focusing on the state of the US economy. You are interested in forecasting the unemployment rate through 2017 (and then 2020) to use it as a benchmark against the incoming administrations economic performance. Use the dataset UNRATENSA.csv and answer the following:

(A) Build a SARIMA model using the unemployment data and produce a 1 year forecast and then a 4 year forecast. Because it is Dec 31, 2016, leave out 2016 as your test data.

```
unem.ts = ts(unem.data$UNRATENSA, frequency = 12, start = c(1948,1))
auto.ts = ts(auto.data$TOTALNSA, frequency = 12, start = c(1976,1))
unem.train = unem.data[0:816,]
unem.test = unem.data[817:834,]
unem.train.ts = ts(unem.train$UNRATENSA)
unem.test.ts = ts(unem.test$UNRATENSA)
```

Construct and Loop through search spaces

From the EDA, we speculated the following models:

- Seasonal Differenced Series: SARIMA(4,0,1)(3,1,Q)
- First Differenced Series: SARIMA(9,1,q)(1,0,1)
- First and Seasonal Differenced Series: SARIMA(4,1,0)(0,1,1)

Because the EDA was inconclusive on differencing strategies, we define the following search spaces:

- Seasonal Differenced Series: SARIMA(1:4,0,0:3)(0:4,1,0:3)
- First Differenced Series: SARIMA(1:9,1,0:3)(0:3,0,0:3)

- First and Seasonal Differenced Series: SARIMA(0:5,1,0:3)(0:3,1,0:3)

where SARIMA(p,d,q)(P,D,Q)_12 is the general form of model.

```
# READ LOOP RESULTS FROM THE CSV FILES

# Seasonal Differenced Series : SARIMA(1:4,0,0:3)(0:4,1,0:3)
bestAIC <- 10000
unem.diff12.df = data.frame("p" = 0, "q" = 0, "P" = 0, "Q" = 0, "aic" = bestAIC)

for(p in 1:4){
  for (q in 0:3){
    for (P in 0:4){
      for (Q in 0:3){
        cat(p,q,P,Q,"\n")
        try(m <- Arima(unem.train.ts, order = c(p, 0, q), seasonal = list(order = c(P, 1, Q), period = 

        if(m$aic < bestAIC) # update if this model attain better aic
        { bestAIC = m$aic
          bestFit = m
          bestModel = c( p, q, P, Q)
          cat(p,q,P,Q,as.numeric(bestAIC), "\n")
          unem.diff12.df = rbind(unem.diff12.df,
                                 data.frame("p" = p, "q" = q, "P" = P, "Q" = Q, "aic" = bestAIC))}
      }
    }
  }
}

unem.diff12.df = unem.diff12.df[seq(dim(unem.diff12.df)[1],1),]
write.csv(x = unem.diff12.df, file = "unem.diff12.df.csv")


# First Differenced Series: SARIMA(9,1,0:3)(0:3,0,0:3)

bestAIC <- 10000

unem.diff.df = data.frame("p" = 0, "q" = 0, "P" = 0, "Q" = 0, "aic" = bestAIC)

for(p in 1:9){
  for (q in 0:3){
    for (P in 0:3){
      for (Q in 0:3){
        cat(p,q,P,Q,"\n")
        try(m <- Arima(unem.train.ts, order = c(p, 1, q), seasonal = list(order = c(P, 0, Q), period = 
            silent = TRUE)

        if(m$aic < bestAIC) # update if this model attain better aic
        { bestAIC = m$aic
        bestFit = m
        bestModel = c( p, q, P, Q)
        cat(p,q,P,Q,as.numeric(bestAIC), "\n")
        unem.diff.df = rbind(unem.diff.df,
                               data.frame("p" = p, "q" = q, "P" = P, "Q" = Q, "aic" = bestAIC))}
      }
```

```
    }
  }
}

unem.diff.df = unem.diff.df[seq(dim(unem.diff.df)[1],1),]
write.csv(x = unem.diff.df, file = "unem.diff.df.csv")



# First and Seasonal Differenced Series: SARIMA(0:5,1,0:3)(0:3,1,0:3)

bestAIC <- 10000

unem.diff.diff12.df = data.frame("p" = 0, "q" = 0, "P" = 0, "Q" = 0, "aic" = bestAIC)

for(p in 0:5){
  for (q in 0:3){
    for (P in 0:3){
      for (Q in 0:3){
        cat(p,q,P,Q,"\n")
        try(m <- Arima(unem.train.ts, order = c(p, 1, q), seasonal = list(order = c(P, 1, Q), period = 
            silent = TRUE)

        if(m$aic < bestAIC) # update if this model attain better aic
        { bestAIC = m$aic
        bestFit = m
        bestModel = c( p, q, P, Q)
        cat(p,q,P,Q,as.numeric(bestAIC), "\n")
        unem.diff.diff12.df = rbind(unem.diff.diff12.df,
                              data.frame("p" = p, "q" = q, "P" = P, "Q" = Q, "aic" = bestAIC))}
      }
    }
  }
}

unem.diff.diff12.df = unem.diff.diff12.df[seq(dim(unem.diff.diff12.df)[1],1),]
write.csv(x = unem.diff.diff12.df, file = "unem.diff.diff12.df.csv")
```

Seasonal Differenced Search Result

```
unem.diff12.df = read.csv("unem.diff12.df.csv")
cat("Top candidates for seasonal differenced model: d = 0, D = 1 \n")
```

```
## Top candidates for seasonal differenced model: d = 0, D = 1
```

```
unem.diff12.df[,3:7]
```

```
##    p q P Q         aic
## 1  4 3 0 1   -37.665823
## 2  3 1 0 1   -36.060627
## 3  2 3 2 3   -35.619890
## 4  2 2 3 3   -35.614309
## 5  2 2 0 1   -35.562127
```

```
## 6  2 1 3 3   -34.076835
## 7  2 1 0 1   -32.641143
## 8  1 3 2 3   -11.353745
## 9  1 3 0 1    -9.276264
## 10 1 2 2 3    -4.640501
## 11 1 2 0 1    -3.786174
## 12 1 1 3 3    28.093231
## 13 1 1 0 1    29.258551
## 14 1 0 2 3    40.238515
## 15 1 0 0 1    44.889942
## 16 1 0 0 0   418.189658
## 17 0 0 0 0 10000.000000
```

First Differenced Search Result

```
unem.diff.df = read.csv("unem.diff.df.csv")
cat("Top candidates for first differenced model: d = 1, D = 0 \n")
```

```
## Top candidates for first differenced model: d = 1, D = 0
```

```
unem.diff.df[,3:7]
```

```
##    p q P Q          aic
## 1  8 3 1 1 -3.046631e+01
## 2  2 3 2 1 -2.184229e+01
## 3  2 1 1 1 -4.512136e+00
## 4  1 2 1 1 -4.140882e+00
## 5  1 1 3 3 -1.155557e+00
## 6  1 1 1 1 -8.886923e-03
## 7  1 0 1 1  3.607811e+01
## 8  1 0 1 0  3.279158e+02
## 9  1 0 0 3  4.500052e+02
## 10 1 0 0 2  5.447998e+02
## 11 1 0 0 1  7.235003e+02
## 12 1 0 0 0  1.081045e+03
## 13 0 0 0 0  1.000000e+04
```

First and Seasonal Differenced Search Result

```
unem.diff.diff12.df = read.csv("unem.diff.diff12.df.csv")
cat("Top candidates for first and seasonal differenced model: d = 1, D = 1 \n")
```

```
## Top candidates for first and seasonal differenced model: d = 1, D = 1
```

```
unem.diff.diff12.df[,3:7]
```

```
##    p q P Q         aic
## 1  2 3 1 1   -28.398517
## 2  2 1 0 1   -18.350352
## 3  1 2 3 3   -18.274790
## 4  1 2 0 1   -18.019477
## 5  1 1 3 3   -14.602928
## 6  1 1 0 1   -12.696370
## 7  0 3 2 3    -4.160424
```

```
## 8  0 3 0 1    -2.766081
## 9  0 2 2 3     0.697895
## 10 0 2 0 1     1.640301
## 11 0 1 3 3    30.312582
## 12 0 1 0 1    32.707970
## 13 0 0 2 3    42.599118
## 14 0 0 0 1    47.126227
## 15 0 0 0 0   427.226229
## 16 0 0 0 0 10000.000000
```

Using the results above, we attempt to simplify the order of the top candidate for each search space by comparing their residuals against respective lower order models.

Seasonal Differenced Models (d = 0 , D = 1)

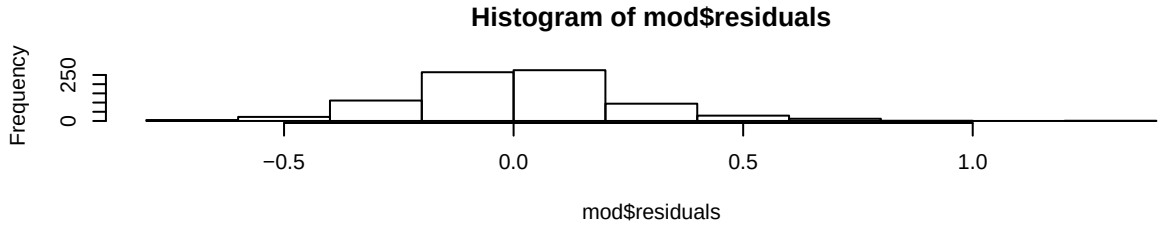
```
# top (4,0,3)(0,1,1)
m.diff12.1 <- Arima(unem.train.ts, order = c(4, 0, 3),
                    seasonal = list(order = c(0, 1, 1), period = 12))

# 2nd (3,0,1)(0,1,1)
m.diff12.2 <- Arima(unem.train.ts, order = c(3, 0, 1),
                    seasonal = list(order = c(0, 1, 1), period = 12))

# 3rd (2,0,3)(2,1,3)
#m.diff12.3 <- Arima(unem.train.ts, order = c(2, 0, 3),
#                    seasonal = list(order = c(2, 1, 3), period = 12))
# 4th (2,0,2)(3,1,3)
#m.diff12.4 <- Arima(unem.train.ts, order = c(3, 0, 1),
#                    seasonal = list(order = c(0, 1, 1), period = 12))
# 5th (2,0,2)(0,1,1)
#m.diff12.5 <- Arima(unem.train.ts, order = c(2, 0, 2),
#                    seasonal = list(order = c(0, 1, 1), period = 12))
# 6th (2,0,1)(3,1,3)
#m.diff12.6 <- Arima(unem.train.ts, order = c(2, 0, 1),
#                    seasonal = list(order = c(3, 1, 3), period = 12))

# 7th (2,0,1)(0,1,1)
m.diff12.7 <- Arima(unem.train.ts, order = c(2, 0, 1),
                    seasonal = list(order = c(0, 1, 1), period = 12))

# 8th (1,0,3)(2,1,3)
m.diff12.8 <- Arima(unem.train.ts, order = c(1, 0, 3),
                    seasonal = list(order = c(2, 1, 3), period = 12))
```

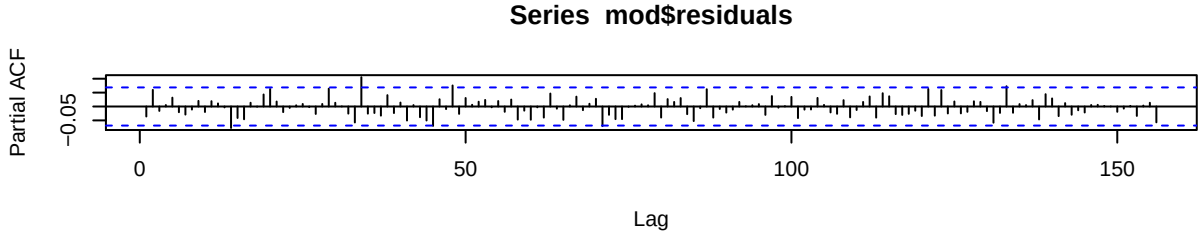
```
# Function to print residual charts
print_resid_chart <- function(mod) {
  cat("Model SARIMA ", c(mod$arma[1], mod$arma[6], mod$arma[2],
                         mod$arma[3], mod$arma[7], mod$arma[4]) , ":\n")
  cat("AIC: ",(mod$aic),"\n")
  cat("BIC: ",(mod$bic),"\n")
  par(mfrow=c(3,1))
  hist(mod$residuals)
  acf(mod$residuals, 156)
```

```
  pacf(mod$residuals, 156)
}
```

```
# top (4,0,3)(0,1,1)
print_resid_chart(m.diff12.1)
```

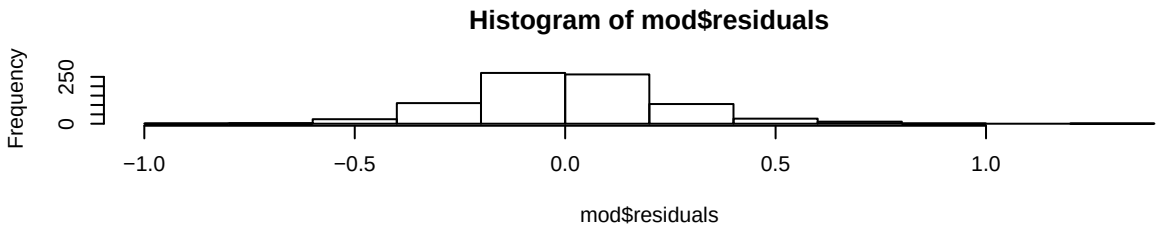
```
## Model SARIMA  4 0 3 0 1 1 :
## AIC:  -37.66582
## BIC:  4.540571
```

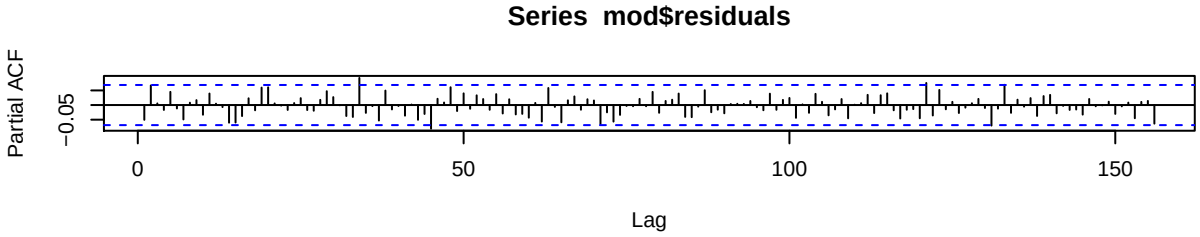





```
#print_resid_chart(m.diff12.2)
#print_resid_chart(m.diff12.3,2)
#print_resid_chart(m.diff12.4,2)
#print_resid_chart(m.diff12.5,2)
#print_resid_chart(m.diff12.6,2)

# 7th (2,0,1)(0,1,1)
print_resid_chart(m.diff12.7)
```

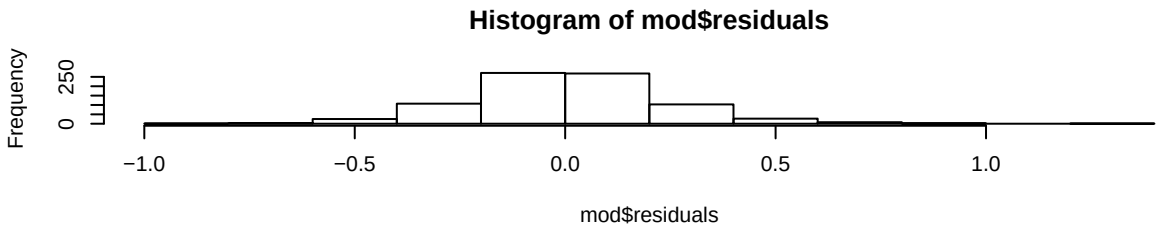
```
## Model SARIMA  2 0 1 0 1 1 :
## AIC:  -32.64114
## BIC:  -9.193147
```

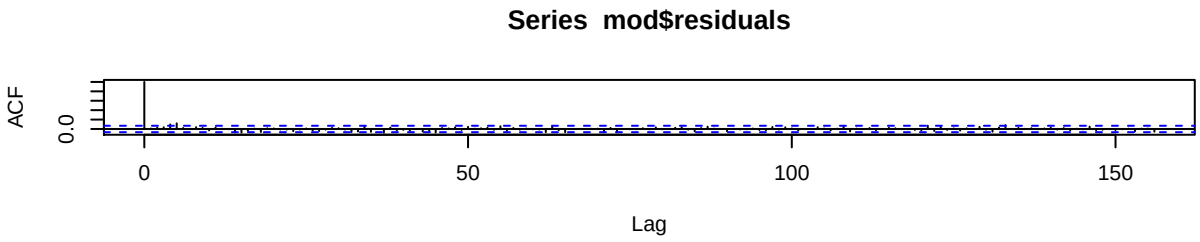


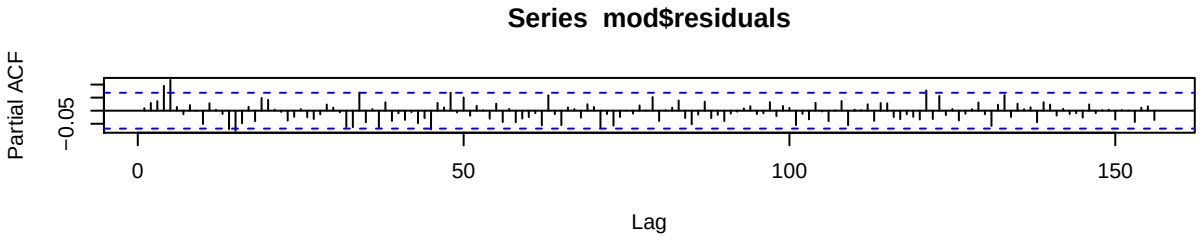



```
# 8th (1,0,3)(2,1,3)
print_resid_chart(m.diff12.8)
```

```
## Model SARIMA  1 0 3 2 1 3 :
## AIC:  -11.35374
## BIC:  35.54225
```








```
# top (4,0,3)(0,1,1)
Box.test(m.diff12.1$residuals, lag=25, type = c("Ljung-Box"))
```

```
##
##  Box-Ljung test
##
## data:  m.diff12.1$residuals
## X-squared = 20.043, df = 25, p-value = 0.7446
```

```
# 7th (2,0,1)(0,1,1)
Box.test(m.diff12.7$residuals, lag=25, type = c("Ljung-Box"))
```

```
##
##  Box-Ljung test
##
## data:  m.diff12.7$residuals
## X-squared = 25.836, df = 25, p-value = 0.4164
```

For seasonal differenced models, we reduced from the top AIC candidate, SARIMA(4,0,3)(0,1,1) to 7th candidate SARIMA(2,0,1)(0,1,1) and retain white noise residual behavior. 8th candidate SARIMA(1,0,3)(2,1,3) starts to show significant pacfs at lag 4 and 5, therefore we stop searching from there downwards. We will keep these two candidates to compare out of sample performance:

- SARIMA(4,0,3)(0,1,1)
- SARIMA(2,0,1)(0,1,1)

We perform the Box-Ljung test above for our residual series to see if residuals for these models are independently distributed. Test hypothesis is as follows:

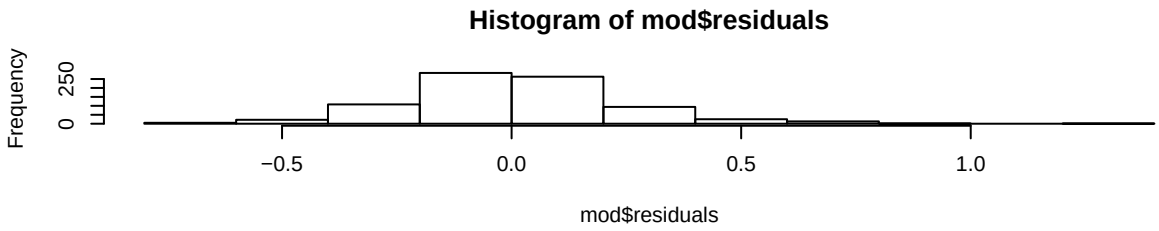
- Ho : The residuals are independently distributed
- Ha : The residuals are not independently distributed
- The tests fail to reject the null hypothesis, thus support that the residuals resemble white noise.

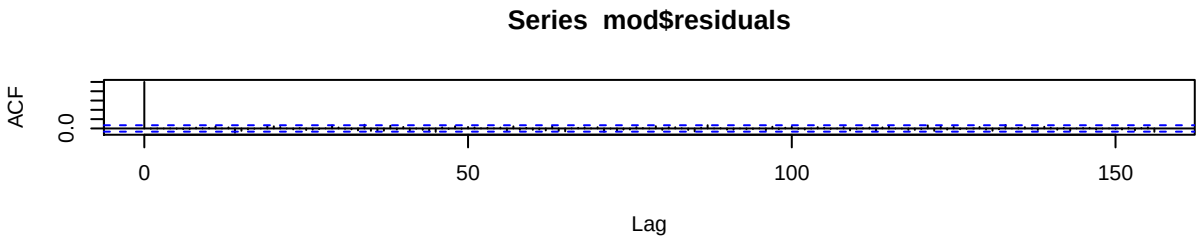
First Differenced Models (d = 1 , D = 0)

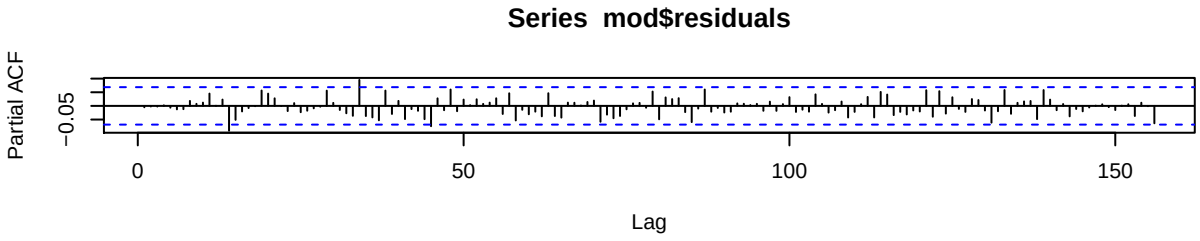
```
# top (8,1,3)(1,0,1)
m.diff.1 <- Arima(unem.train.ts, order = c(8, 1, 3),
                  seasonal = list(order = c(1, 0, 1), period = 12))

# 2nd (2,1,3)(2,0,1)
#m.diff.2 <- Arima(unem.train.ts, order = c(2, 1, 3), seasonal = list(order = c(2, 0, 1), period = 12))

# 3rd (2,1,1)(1,0,1)
m.diff.3 <- Arima(unem.train.ts, order = c(2, 1, 1), seasonal = list(order = c(1, 0, 1), period = 12))

# 4th (1,1,2)(1,0,1)
#m.diff.4 <- Arima(unem.train.ts, order = c(1, 1, 2), seasonal = list(order = c(1, 0, 1), period = 12))
# 5th (1,1,1)(3,0,3)
#m.diff.5 <- Arima(unem.train.ts, order = c(1, 1, 1), seasonal = list(order = c(3, 0, 3), period = 12))

# 6th (1,1,1)(1,0,1)
m.diff.6 <- Arima(unem.train.ts, order = c(1, 1, 1),
                  seasonal = list(order = c(1, 0, 1), period = 12))

# 7th (1,1,0)(1,0,1)
m.diff.7 <- Arima(unem.train.ts, order = c(1, 1, 0),
                  seasonal = list(order = c(1, 0, 1), period = 12))

# 8th (1,1,0)(1,0,0)
# m.diff.8 <- Arima(unem.train.ts, order = c(1, 1, 0),
#                   seasonal = list(order = c(1, 0, 0), period = 12))
```

```
# top (8,1,3)(1,0,1)
print_resid_chart(m.diff.1)
```

```
## Model SARIMA  8 1 3 1 0 1 :
## AIC:  -30.46631
## BIC:  35.37832
```

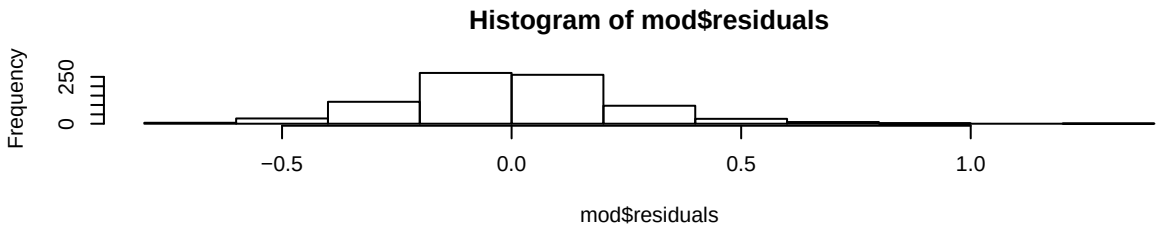


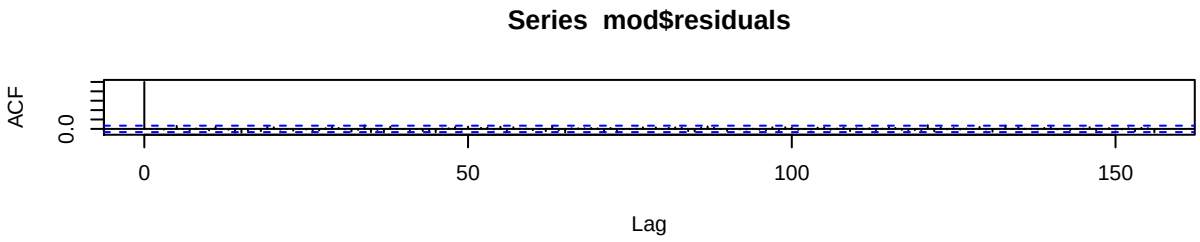





```
#print_resid_chart(m.diff.2)

# third (2,1,1)(1,0,1)
print_resid_chart(m.diff.3)
```

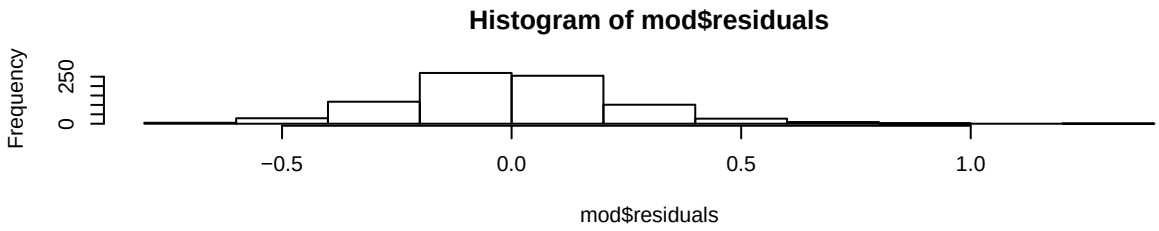
```
## Model SARIMA  2 1 1 1 0 1 :
## AIC:  -4.512136
## BIC:  23.70699
```

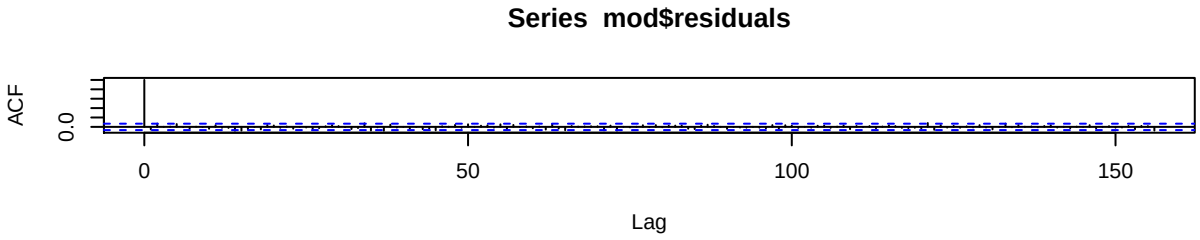


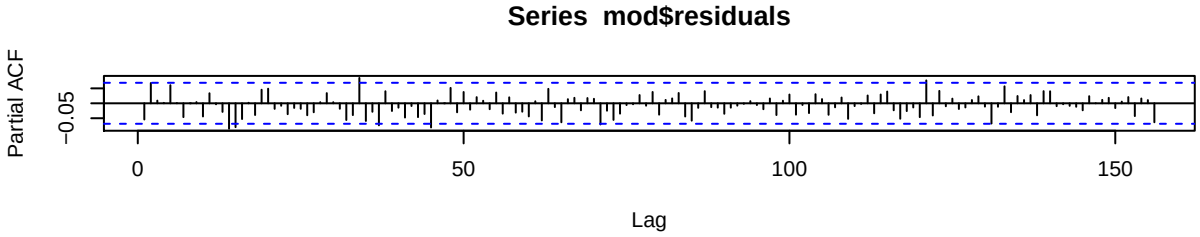



```r
# 4th (1,1,2)(1,0,1)
#print_resid_chart(m.diff.4)
# 5th (1,1,1)(3,0,3)
#print_resid_chart(m.diff.5)

# 6th (1,1,1)(1,0,1)
print_resid_chart(m.diff.6)
```

```
## Model SARIMA  1 1 1 1 0 1 :
## AIC:  -0.008886923
## BIC:  23.50705
```

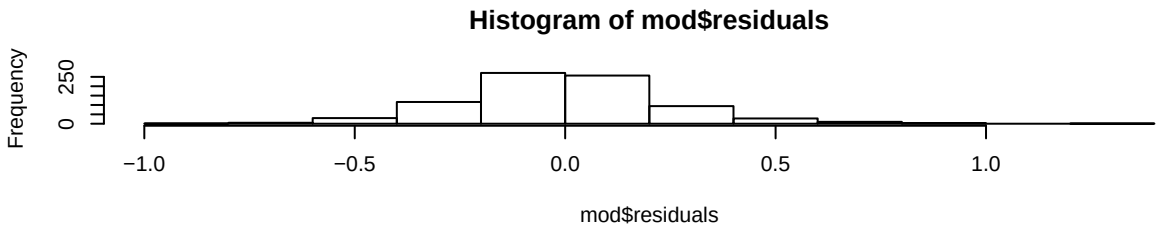


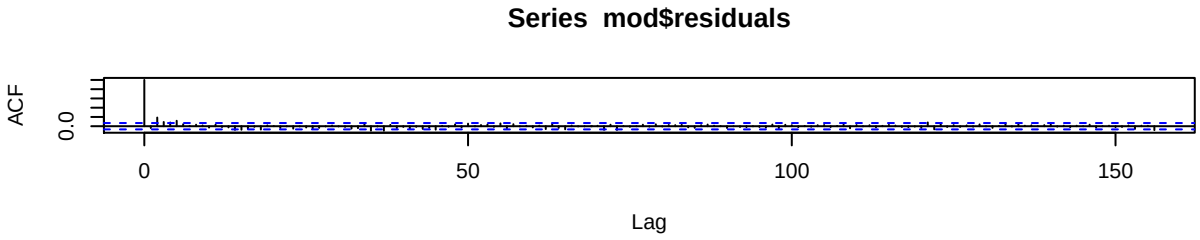


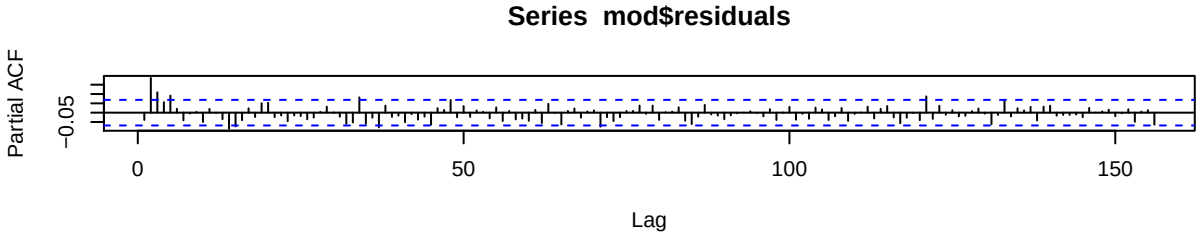



```
# 7th (1,1,0)(1,0,1)
print_resid_chart(m.diff.7)
```

```
## Model SARIMA  1 1 0 1 0 1 :
## AIC:  36.07811
## BIC:  54.89086
```








```
# 8th (1,1,0)(1,0,0)
# print_resid_chart(m.diff.8)
```

```
# top (8,1,3)(1,0,1)
Box.test(m.diff.1$residuals, lag=25, type = c("Ljung-Box"))
```

```
##
##  Box-Ljung test
##
## data:  m.diff.1$residuals
## X-squared = 19.01, df = 25, p-value = 0.7966
```

```
# third (2,1,1)(1,0,1)
Box.test(m.diff.3$residuals, lag=25, type = c("Ljung-Box"))
```

```
##
##  Box-Ljung test
##
## data:  m.diff.3$residuals
## X-squared = 25.946, df = 25, p-value = 0.4105
```

```
# 6th (1,1,1)(1,0,1)
Box.test(m.diff.6$residuals, lag=25, type = c("Ljung-Box"))
```

```
##
##  Box-Ljung test
##
## data:  m.diff.6$residuals
```

```
## X-squared = 32.498, df = 25, p-value = 0.1441
```

For first differenced models, we reduced from the top AIC candidate, SARIMA(8,1,3)(1,0,1) to 3rd candidate SARIMA(2,1,1)(1,0,1) and retain white noise residual behavior. 6th candidate SARIMA(1,1,1)(1,0,1) is still satisfactory but more of its lags are slightly closer to the cut-off. 7th candidate SARIMA(1,1,0)(1,0,1) starts to show significant pacfs at lag 2, 3 and 5, and 8th candidate SARIMA(1,1,0)(1,0,0) starts to show significant pacfs at seasonal lags, therefore we stop searching from there downwards. We will keep the first three candidates to compare out of sample performance:

- SARIMA(8,1,3)(1,0,1)
- SARIMA(2,1,1)(1,0,1)
- SARIMA(1,1,1)(1,0,1)

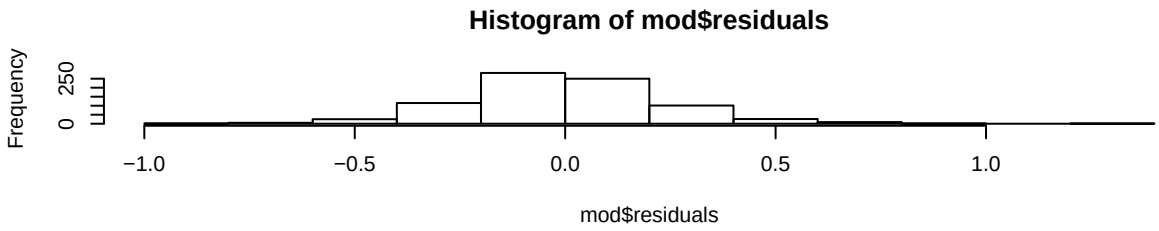
The Box-Ljung tests above rejected the null hypothesis for all three models to support that our residuals are independently distributed.

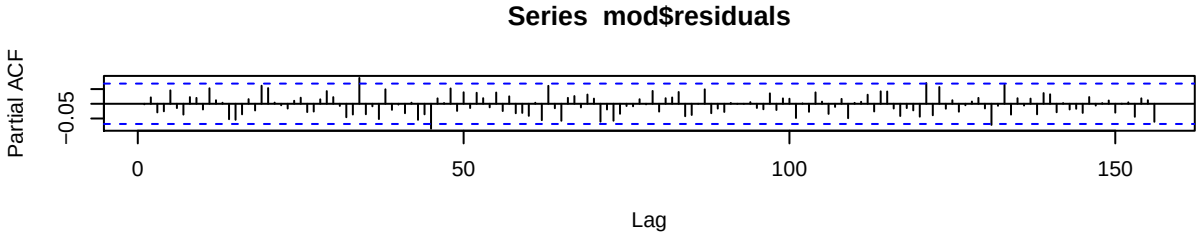
First and Seasonal Differenced Models (d = 1 , D = 1)

```
# top (2,1,3)(1,1,1)
m.diff.diff12.1 <- Arima(unem.train.ts, order = c(2, 1, 3),
                         seasonal = list(order = c(1, 1, 1), period = 12))

# 2nd (2,1,1)(0,1,1)
m.diff.diff12.2 <- Arima(unem.train.ts, order = c(2, 1, 1),
                         seasonal = list(order = c(0, 1, 1), period = 12))
# 3rd (1,1,2)(3,1,3)
m.diff.diff12.3 <- Arima(unem.train.ts, order = c(1, 1, 2),
                         seasonal = list(order = c(3, 1, 3), period = 12))

# 4th (1,1,2)(0,1,1)
#m.diff.diff12.4 <- Arima(unem.train.ts, order = c(1, 1, 2),
#                          seasonal = list(order = c(0, 1, 1), period = 12))
# 5th (1,1,1)(3,1,3)
#m.diff.diff12.5 <- Arima(unem.train.ts, order = c(1, 1, 1),
#                          seasonal = list(order = c(3, 1, 3), period = 12))
# 6th (1,1,1)(0,1,1)
#m.diff.diff12.6 <- Arima(unem.train.ts, order = c(1, 1, 1),
#                           seasonal = list(order = c(0, 1, 1), period = 12))

# 7th (0,1,3)(2,1,3)
m.diff.diff12.7 <- Arima(unem.train.ts, order = c(0, 1, 3),
                         seasonal = list(order = c(2, 1, 3), period = 12))
```

```
# top (2,1,3)(1,1,1)
print_resid_chart(m.diff.diff12.1)
```

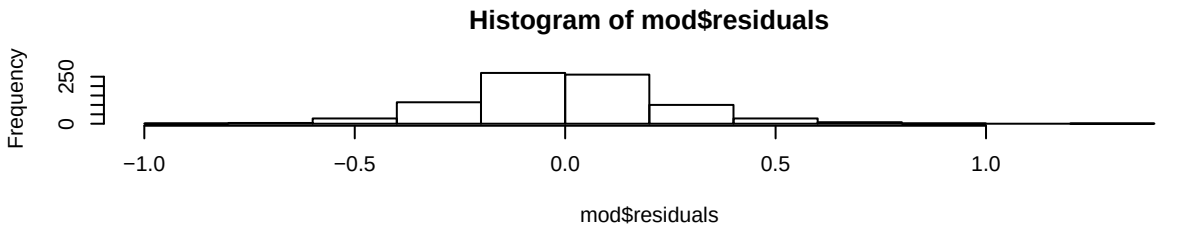
```
## Model SARIMA  2 1 3 1 1 1 :
## AIC:  -28.39852
## BIC:  9.10832
```

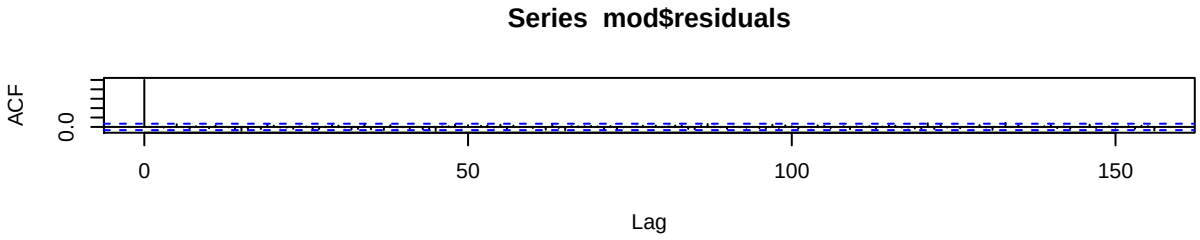


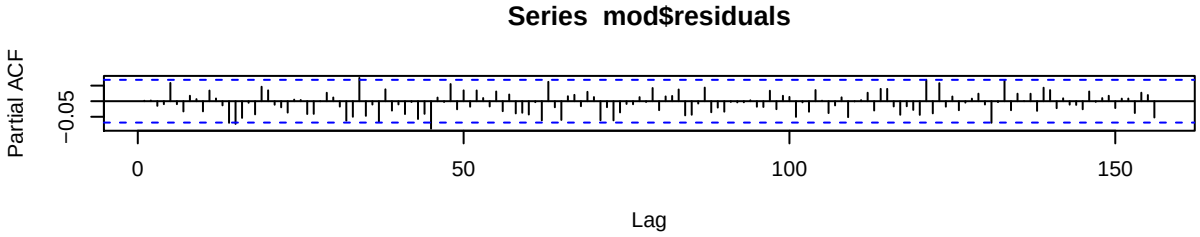



```
# 2nd (2,1,1)(0,1,1)
print_resid_chart(m.diff.diff12.2)
```

```
## Model SARIMA  2 1 1 0 1 1 :
## AIC:  -18.35035
## BIC:  5.091422
```

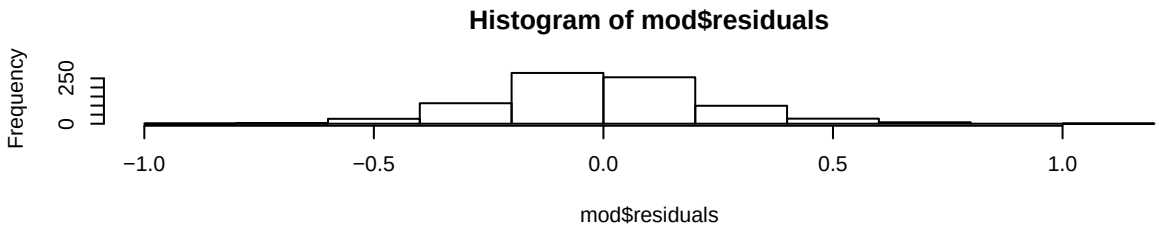


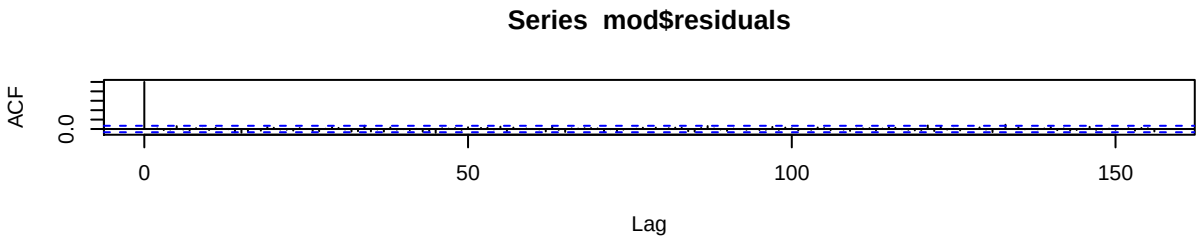





```r
# 3rd (1,1,2)(3,1,3)
print_resid_chart(m.diff.diff12.3)
```

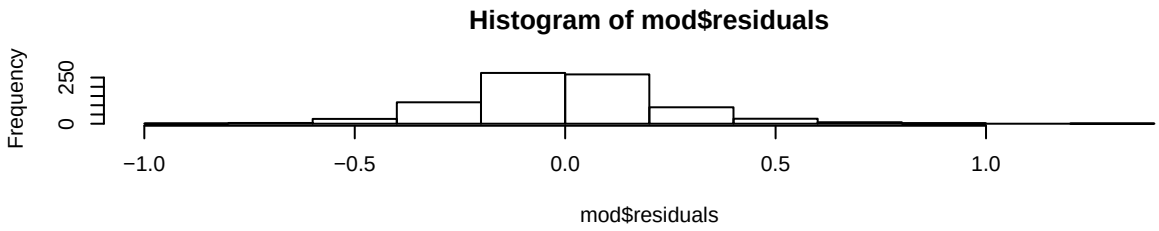
```
## Model SARIMA  1 1 2 3 1 3 :
## AIC:  -18.27479
## BIC:  28.60876
```

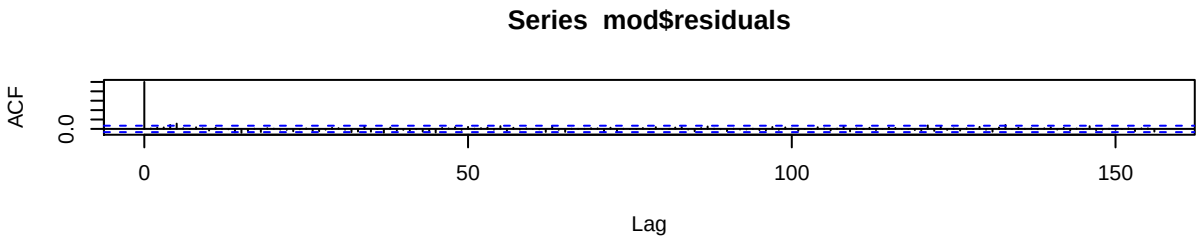


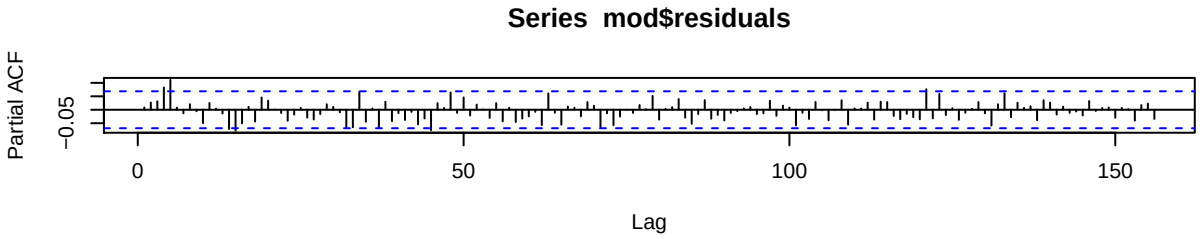



```
# 4th (1,1,2)(0,1,1)
#print_resid_chart(m.diff.diff12.4)
# 5th (1,1,1)(3,1,3)
#print_resid_chart(m.diff.diff12.5)
# 6th (1,1,1)(0,1,1)
#print_resid_chart(m.diff.diff12.6)

# 7th (0,1,3)(2,1,3)
print_resid_chart(m.diff.diff12.7)
```

```
## Model SARIMA  0 1 3 2 1 3 :
## AIC:  -4.160424
## BIC:  38.03477
```








```
# top (2,1,3)(1,1,1)
Box.test(m.diff.diff12.1$residuals, lag=25, type = c("Ljung-Box"))
```

```
##
##  Box-Ljung test
##
## data:  m.diff.diff12.1$residuals
## X-squared = 20.591, df = 25, p-value = 0.7152
```

```
# 2nd (2,1,1)(0,1,1)
Box.test(m.diff.diff12.2$residuals, lag=25, type = c("Ljung-Box"))
```

```
##
##  Box-Ljung test
##
## data:  m.diff.diff12.2$residuals
## X-squared = 22.384, df = 25, p-value = 0.6135
```

```
# 3rd (1,1,2)(3,1,3)
Box.test(m.diff.diff12.3$residuals, lag=25, type = c("Ljung-Box"))
```

```
##
##  Box-Ljung test
##
## data:  m.diff.diff12.3$residuals
## X-squared = 21.515, df = 25, p-value = 0.6636
```

For first and seasonal differenced models, we reduced from the top AIC candidate, SARIMA(2,1,3)(1,1,1) to

2nd candidate SARIMA(2,1,1)(0,1,1) and 3rd candidate(1,1,2)(3,1,3) with similar white noise behaviors. The 4th candidate SARIMA(1,1,2)(0,1,1), 5th candidate SARIMA(1,1,1)(3,1,3) and 6th SARIMA(1,1,1)(0,1,1) are still satisfactory but more of their lags are slightly closer to the cut-off (plots skipped here). 7th candidate SARIMA(0,1,3)(2,1,3) starts to show significant pacfs at lag 5 and 6, therefore we stop searching from there downwards. We will keep these three candidates to compare out of sample performance:

- SARIMA(2,1,3)(1,1,1)
- SARIMA(2,1,1)(0,1,1)
- SARIMA(1,1,2)(3,1,3)

The Box-Ljung tests above rejected the null hypothesis for all three models to support that our residuals are independently distributed.

Compare models based on out of sample errors uptil Jun 2017

(A.i)How well does your model predict the unemployment rate up until June 2017?

```
# candidate models
# SARIMA(4,0,3)(0,1,1) m.diff12.1
# SARIMA(2,0,1)(0,1,1) m.diff12.7
# SARIMA(8,1,3)(1,0,1) m.diff.1
# SARIMA(2,1,1)(1,0,1) m.diff.3
# SARIMA(1,1,1)(1,0,1) m.diff.6
# SARIMA(2,1,3)(1,1,1) m.diff.diff12.1
# SARIMA(2,1,1)(0,1,1) m.diff.diff12.2
# SARIMA(1,1,2)(3,1,3) m.diff.diff12.3

candidate_mods = list(m.diff12.1,m.diff12.7,m.diff.1,m.diff.3,m.diff.6,m.diff.diff12.1,m.diff.diff12.2,m
```

```
# function to get RMSE
get_RMSE = function(test.df, mod, ahead){
  f = forecast(mod, ahead)$mean
  sq.error = (test.df$UNRATENSA - f)^2
  rmse = sqrt(mean(sq.error))
  return (data.frame( "p" = mod$arma[1], "d" = mod$arma[6], "q" = mod$arma[2],
                      "P" = mod$arma[3], "D" = mod$arma[7], "Q" = mod$arma[4],
                      "RMSE" = rmse))
}
```

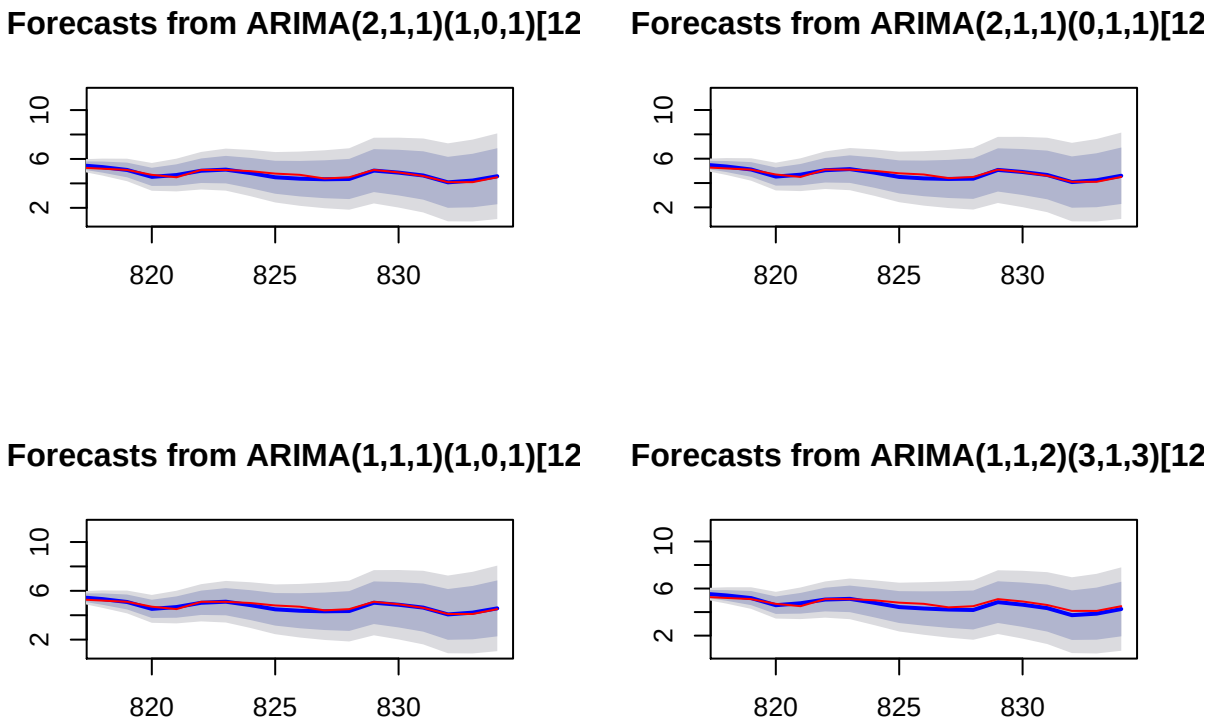
```
RMSE.df = data.frame()
for (i in 1:length(candidate_mods)) {
  add.df = get_RMSE(unem.test, candidate_mods[[i]], 18)
  RMSE.df = rbind(RMSE.df, add.df)
}
RMSE.df = RMSE.df[order(RMSE.df$RMSE),]
RMSE.df
```

```
##   p d q P D Q      RMSE
## 4 2 1 1 1 0 1 0.1472144
## 7 2 1 1 0 1 1 0.1510539
## 5 1 1 1 1 0 1 0.1516677
## 8 1 1 2 3 1 3 0.2519264
## 3 8 1 3 1 0 1 0.8150629
## 1 4 0 3 0 1 1 0.8636125
## 6 2 1 3 1 1 1 0.8867261
## 2 2 0 1 0 1 1 0.8890228
```

Out of sample errors tells us that SARIMA(2,1,1)(1,0,1), SARIMA(2,1,1)(0,1,1), SARIMA(1,1,1)(1,0,1) and SARIMA(1,1,2)(3,1,3) perform superior to the other candidates and their RMSEs are very close. We plot their forecasts to further compare these three.

```
unem.actual.ts = ts(unem.data$UNRATENSA)
# (2,1,1)(1,0,1)
f.diff.3<-forecast(m.diff.3,18)
# (2,1,1)(0,1,1)
f.diff.diff12.2<-forecast(m.diff.diff12.2,18)
# (1,1,1)(1,0,1)
f.diff.6<-forecast(m.diff.6,18)
# (1,1,2)(3,1,3)
f.diff.diff12.3<-forecast(m.diff.diff12.3,18)
```

```
par(mfrow = c(2,2))

plot(f.diff.3, xlim = c(818,834))
lines(unem.actual.ts,type="l",col="red")
plot(f.diff.diff12.2, xlim = c(818,834))
lines(unem.actual.ts,type="l",col="red")
plot(f.diff.6, xlim = c(818,834))
lines(unem.actual.ts,type="l",col="red")
plot(f.diff.diff12.3, xlim = c(818,834))
lines(unem.actual.ts,type="l",col="red")
```



All four models predict closely to the test data. SARIMA(2,1,1)(1,0,1) performing the closest in the time plots and it has the lowest RMSE of 0.147. We will use this model to forecast unemployment rate until 2020.

Before that, the model specification is given by the following:

```
m.diff.3$coef
```

```
##        ar1        ar2        ma1       sar1       sma1
##  0.5985857  0.1241366 -0.4845903  0.9918375 -0.7582691
```

```
# characteristic equation for differenced AR component
Mod(polyroot(c(1,-0.5986,-0.1241)))
```

```
## [1] 1.313101 6.136631
```

```
# characteristic equation for differenced SAR component
Mod(polyroot(c(1,-0.9918)))
```

```
## [1] 1.008268
```

```
# characteristic equation for differenced MA component
Mod(polyroot(c(1,-0.4846)))
```

```
## [1] 2.063558
```

```
# characteristic equation for differenced SMA component
Mod(polyroot(c(1,-0.7583)))
```

```
## [1] 1.318739
```

$$(1-\Theta B^{12})(1-\theta_1 B-\theta_2 B^2)(1-B)x_t = (1+\Phi B^{12})(1+\phi B)w_t$$

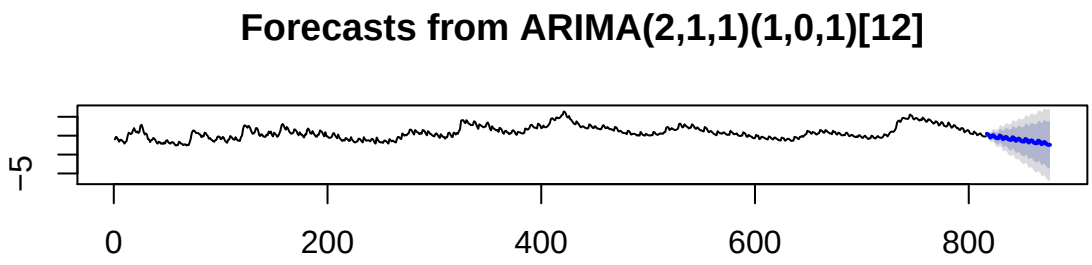
where $\Theta = +0.9918$, $\theta_1 = +0.5986$, $\theta_2 = +0.1241$, $\Phi = -0.7583$, $\phi = -0.4846$.

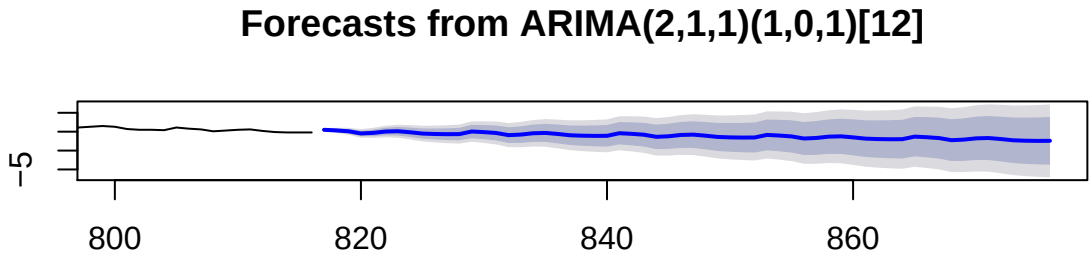
- A unit root on the left specified by $(1-B)$, which was taken care of by first differencing.
- The first differenced AR component has characteristic equation $1-0.5986B-0.1241B^2=0$, the roots for B are 1.313 and 6.136. The first differenced seasonal component has characteristic equation $1-0.9918B^{12}=0$, the root for B^{12} is 1.008. All roots exceed unity which means the first differenced series is stationary.
- The first differenced MA component has characteristic equation $1-0.4846B=0$, the roots for B is 2.064. The first differenced seasonal MA component has characteristic equation $1-0.7583B^{12}=0$, the roots for B^{12} is 1.319. All roots exceed unity which means the first differenced series is invertible.

Forecast until 2020

(A.ii) What does the unemployment rate look like at the end of 2020? How credible is this estimate?

```
#817 2016 Jan
#forecast.sarima.ts = ts(forecast(m.diff.3,h=60)$mean, start = c(2016,1), frequency = 12)

par(mfrow = c(2,1))

#forecast out to Dec 2020
plot(forecast(m.diff.3,h=60))
plot(forecast(m.diff.3,h=60),xlim=c(800,876))
```






```
f.diff.3.2020<-forecast(m.diff.3,h=60)
cat("Forecast 2020 December -- Expected Value",
    f.diff.3.2020$mean[60], "\n")
```

```
## Forecast 2020 December -- Expected Value 2.583579
```

```
cat("Forecast 2020 December -- 95% Lower Confidence Bound",
    f.diff.3.2020$lower[60,2], "\n")
```

```
## Forecast 2020 December -- 95% Lower Confidence Bound -7.054678
```

```
cat("Forecast 2020 December -- 95% Upper Confidence Bound",
    f.diff.3.2020$upper[60,2], "\n")
```

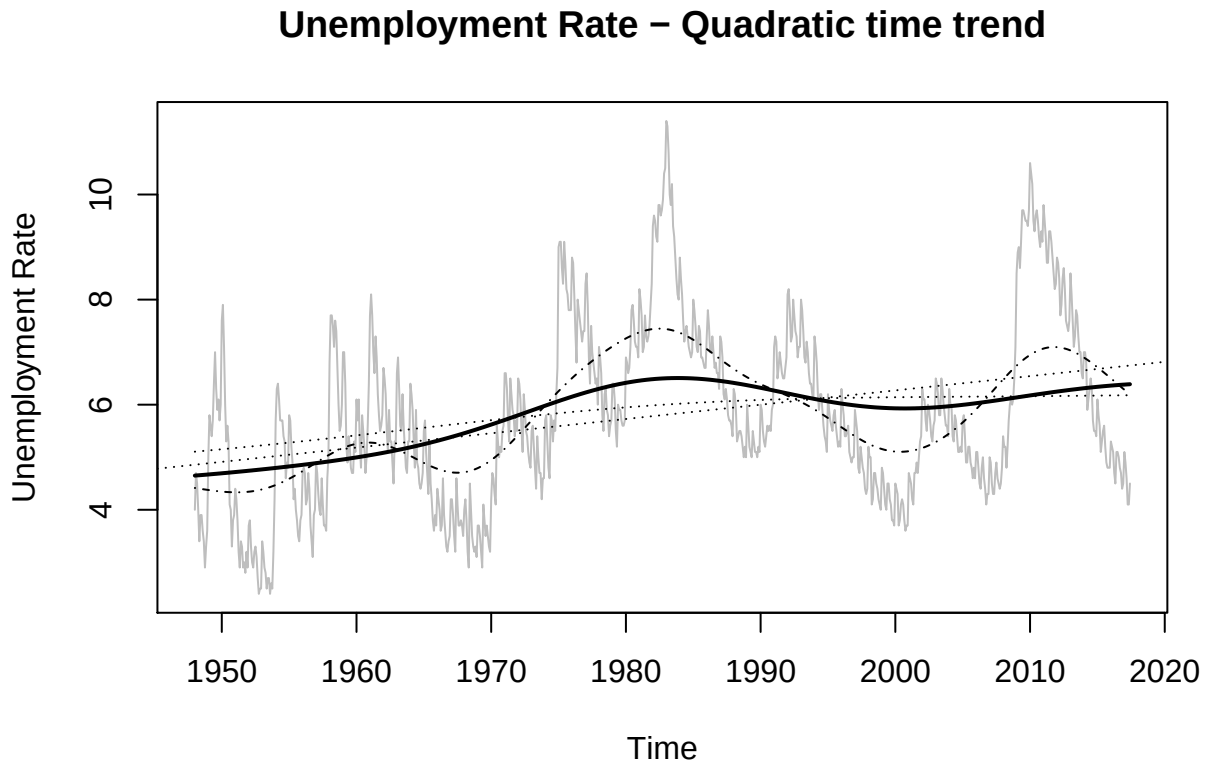
```
## Forecast 2020 December -- 95% Upper Confidence Bound 12.22184
```

As we can see from the forecast time plots above, the confidence bound of forecast expand drastically towards 2020. Although the mean value is expected to be 2.5835, this estimate is not very credible. The wide point estimates of confidence bound covers zero so we don't even know if unemployment will be positive or negative!

(B) Build a linear time-regression and incorporate seasonal effects. Be sure to evaluate the residuals and assess this model on the basis of the assumptions of the classical linear model, and then produce a 1 year and a 4 year forecast.

```
# Kernel smoothing
unem.k.smooth.widest = ksmooth(time(unem.ts),
                               unem.ts, kernel = c("normal"),
                               bandwidth = 50)
```

```r
unem.k.smooth.wide = ksmooth(time(unem.ts),
                             unem.ts, kernel = c("normal"),
                             bandwidth = 25)

unem.k.smooth.narrow = ksmooth(time(unem.ts),
                               unem.ts, kernel = c("normal"),
                               bandwidth = 10)
# Make plot
plot(unem.ts, col = "gray", ylab = "Unemployment Rate",
     main = "Unemployment Rate - Quadratic time trend")
lines(unem.k.smooth.widest$x, unem.k.smooth.widest$y,
      col = "black", lty = "dotted")
lines(unem.k.smooth.wide$x, unem.k.smooth.wide$y,
      col = "black", lty = "solid", lwd = 2)
lines(unem.k.smooth.narrow$x, unem.k.smooth.narrow$y,
      col = "black", lty = "dotdash")
abline(lm(unem.ts~time(unem.ts)), lty = "dotted", col = "black")
```



Continuing the insights from our EDA, the time series of unemployment rate has an upward trend and show some quadratic behavior. Therefore we can estimate a model with quadratic term of time. We attempt to explain seasonal behavior using a dummy variable for each month. Candidate Models are:

$$X_t = \beta_0 + \beta_t time + \beta_m month + \epsilon$$

$$X_t = \beta_0 + \beta_t time + \beta_{t2} time^2 + \beta_{\mathbf{m}} \mathbf{month} + \epsilon$$

where time is an annual unit, each additional month is expressed as a fraction of the year. "month" is an categorical variable that is broken down into indicator variables.

```
# Preparing data
unem.ts = ts(unem.data$UNRATENSA, frequency = 12, start = c(1948,1))

unem.train.ts = window(unem.ts, end = c(2015,12))
unem.test.ts = window(unem.ts, start = c(2016,01))

y.lm = as.numeric(unem.train.ts)
t.lm = as.numeric(time(unem.train.ts))
mon.lm = as.factor(cycle(unem.train.ts))
```

```
unem.lm = lm(y.lm ~ t.lm + mon.lm)
unem.lm.quad = lm(y.lm ~ t.lm + I(t.lm^2) + mon.lm)

AIC(unem.lm)
```

```
## [1] 3036.805
```

```
AIC(unem.lm.quad)
```

```
## [1] 3008.384
```

```
summary(unem.lm.quad)
```

```
##
## Call:
## lm(formula = y.lm ~ t.lm + I(t.lm^2) + mon.lm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6856 -1.0799 -0.2239  0.9789  4.6227
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.389e+03  6.040e+02  -5.611 2.77e-08 ***
## t.lm         3.396e+00  6.095e-01   5.571 3.46e-08 ***
## I(t.lm^2)   -8.487e-04  1.538e-04  -5.519 4.59e-08 ***
## mon.lm2     -1.149e-02  2.596e-01  -0.044 0.964698
## mon.lm3     -2.906e-01  2.596e-01  -1.119 0.263324
## mon.lm4     -8.050e-01  2.596e-01  -3.101 0.001999 **
## mon.lm5     -9.518e-01  2.596e-01  -3.666 0.000263 ***
## mon.lm6     -3.515e-01  2.596e-01  -1.354 0.176211
## mon.lm7     -5.144e-01  2.596e-01  -1.981 0.047917 *
## mon.lm8     -8.229e-01  2.596e-01  -3.169 0.001586 **
## mon.lm9     -1.000e+00  2.596e-01  -3.853 0.000126 ***
## mon.lm10    -1.169e+00  2.596e-01  -4.503 7.69e-06 ***
## mon.lm11    -9.982e-01  2.596e-01  -3.845 0.000130 ***
## mon.lm12    -9.596e-01  2.596e-01  -3.696 0.000234 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.514 on 802 degrees of freedom
## Multiple R-squared:  0.2132, Adjusted R-squared:  0.2004
## F-statistic: 16.71 on 13 and 802 DF,  p-value: < 2.2e-16
```

```
library(sandwich)
```

```
## Warning: package 'sandwich' was built under R version 3.3.3
```

```
car::linearHypothesis(unem.lm.quad, c("mon.lm2 = 0", "mon.lm3 = 0",
                                      "mon.lm4 = 0", "mon.lm5 = 0",
                                      "mon.lm6 = 0", "mon.lm7 = 0",
                                      "mon.lm8 = 0", "mon.lm9 = 0",
                                      "mon.lm10 = 0", "mon.lm11 = 0",
                                      "mon.lm12 = 0"), vcov = vcovHC)
```

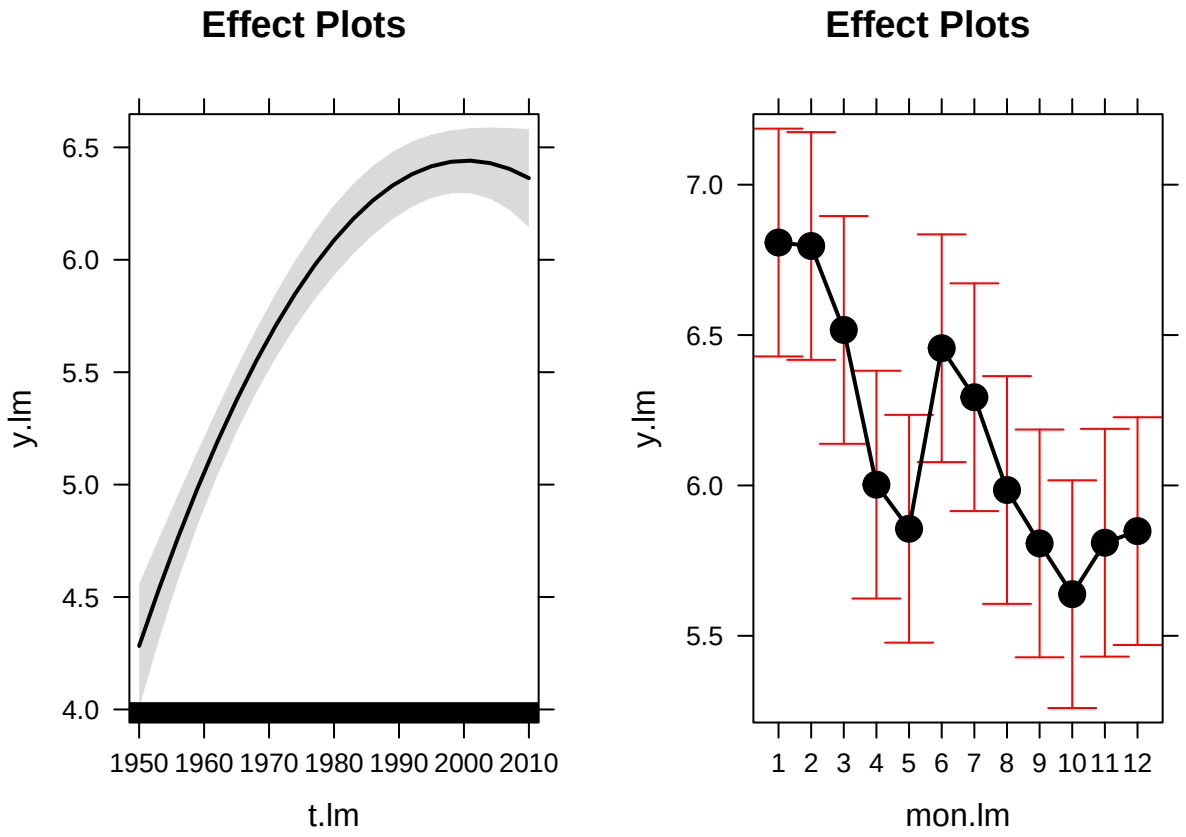
```
## Linear hypothesis test
##
## Hypothesis:
## mon.lm2 = 0
## mon.lm3 = 0
## mon.lm4 = 0
## mon.lm5 = 0
## mon.lm6 = 0
## mon.lm7 = 0
## mon.lm8 = 0
## mon.lm9 = 0
## mon.lm10 = 0
## mon.lm11 = 0
## mon.lm12 = 0
##
## Model 1: restricted model
## Model 2: y.lm ~ t.lm + I(t.lm^2) + mon.lm
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1    813
## 2    802 11 4.6588 6.382e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our quadratic model perform slightly better than the linear model by AIC and Adjusted R-squared. The F-statistic and F-test p-value strongly reject that null hypothesis that our coefficients are not jointly significant thus support explanatory power of our model. The second F-test result also provide evidence that our month variable is significant. We will proceed with this quadratic model for a better fit. Our estimated model is specified as:

$$\hat{X}_t = -3389 + 3.396time - 0.0008487time^2 + \beta_m month - 0.805 \cdot I(Apr) - 0.9518 \cdot I(May) - 0.5144 \cdot I(Jul) - 0.8229 \cdot I(Aug) - 1 \cdot I(Sep$$

Notice that we have dropped some month indicator variables in the specification, because they are not statistical different from the base variable January. Along with the negative sign of the significant indicator variables, the estimations agrees with our earlier month plot in the EDA section that the beginning and middle of each year tend to have higher unemployment rates.

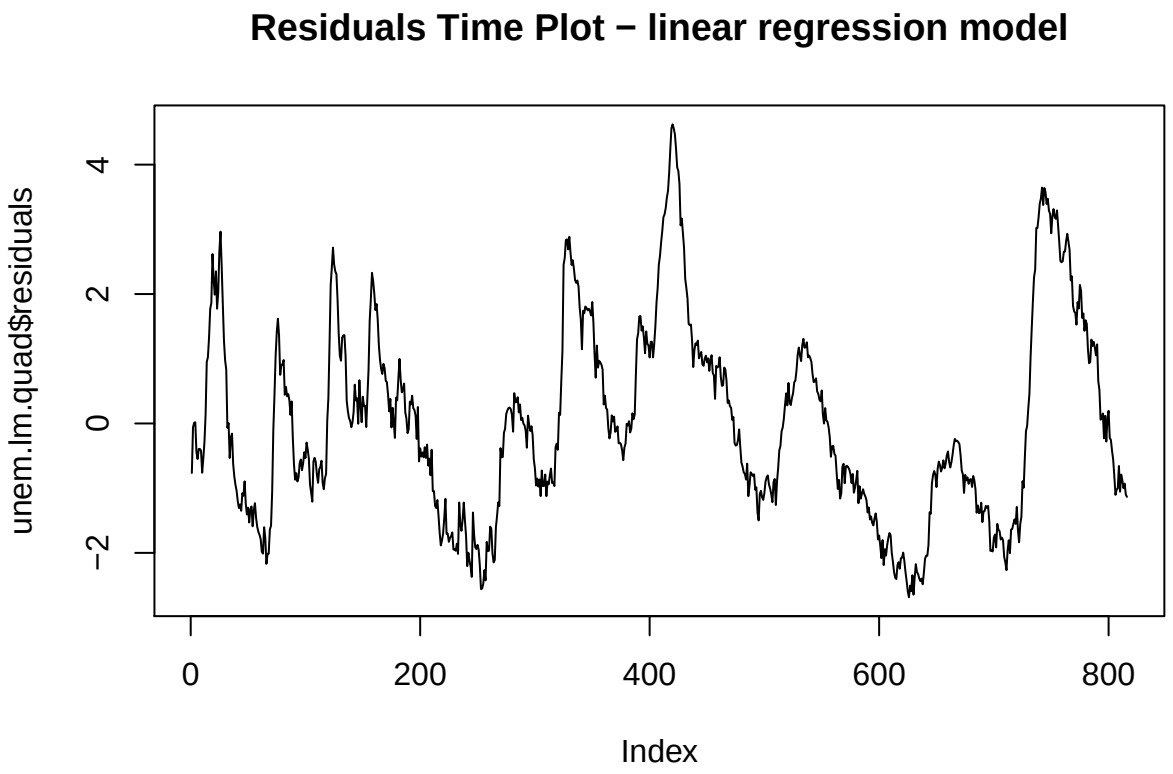
```
plot(effects::allEffects(unem.lm.quad)[c(1,3)],
     main = "Effect Plots")
```



The time effect plot above shows the fitted curve with a positive, gradually leveling slope and expanding confidence interval with the increase in time. Also, the month effect plot show discrete levels and the model clearly discriminate early and mid year from the other months. Both plots align with our EDA findings. To further examine the internal validity of our model, we evaluate the validity of our model by the 6 CLM assumptions:

1. Linearity in Parameters: This is a weak assumption, we have specified our model with linear coefficients.
2. Random sample of data: We clealy have violated this assumption because the observations are serially correlated, as demonstrated in the autocorrelation plots in the EDA. The strongly time dependent residuals plotted below demonstrates that our observations could not have been independent in the first place.

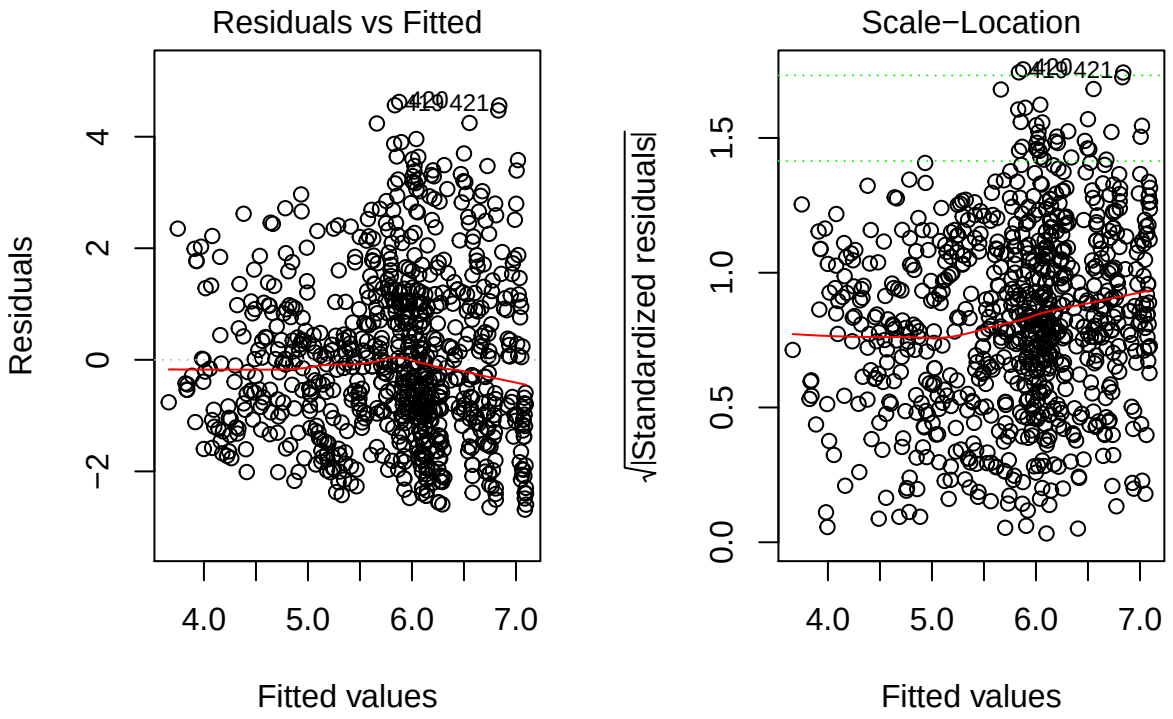
```
plot(unem.lm.quad$residuals, type = "l",
     main = "Residuals Time Plot - linear regression model")
```



3. No perfect co-linearity: Each year has 12 months, therefore the time variable is not correlated with month.

4. Zero-conditional mean. From the residuals vs fitted value plot, there is a clear curvature of the loess curve from line zero. We tried a separate model with the cubic term of time but the curvature was only flipped not flattened. We could be missing an important variable here. This assumption is violated.

```
par(mfrow = c(1,2))
plot(unem.lm.quad, which = 1)
plot(unem.lm.quad, which = 3)
abline(h = c(sqrt(2),sqrt(3)), col = "green", lty = "dotted")
```



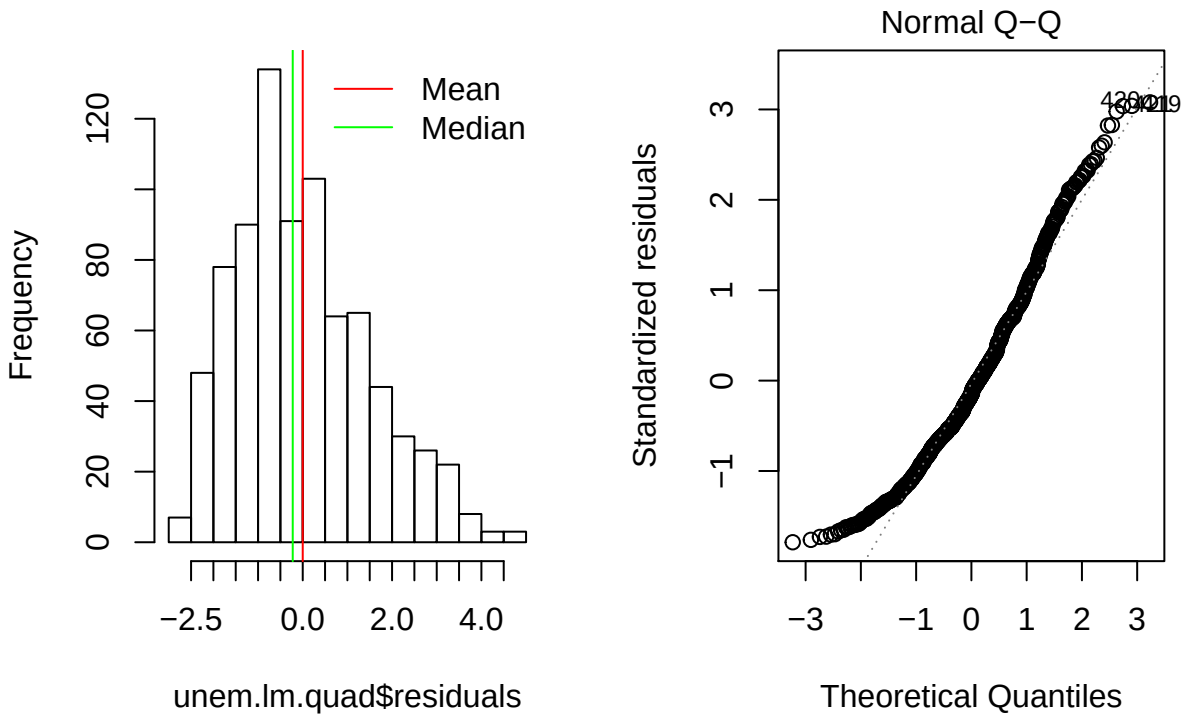
5. Homoskedasticity of errors: The variance of residuals noticeably expand towards higher fitted values. The Loess curve on the scale-location plot clearly picks up at the same time. The Breusch-Pagan test strongly reject the null hypothesis of homoskedasticity. This assumption is violated.

```
lmtest::bptest(unem.lm.quad)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  unem.lm.quad
## BP = 29.662, df = 13, p-value = 0.005268
```

6. Normally distributed error: From the normal QQ plot and histogram, our residuals are clearly right skewed. The Shapiro test also strongly reject the null hypothesis that our residuals are normally distributed.

```
par(mfrow = c(1,2))
hist(unem.lm.quad$residuals, xaxt = "n",
     main = "Residuals of linear regression model")
axis(side = 1, at = seq(-2.5,4.5,0.5))
abline(v = mean(unem.lm.quad$residuals), col = "red")
abline(v = median(unem.lm.quad$residuals), col = "green")
legend("topright", legend = c("Mean", "Median"),
       col = c("red", "green"), bty = "n",
       lty = c("solid", "solid"))

plot(unem.lm.quad, which = 2)
```

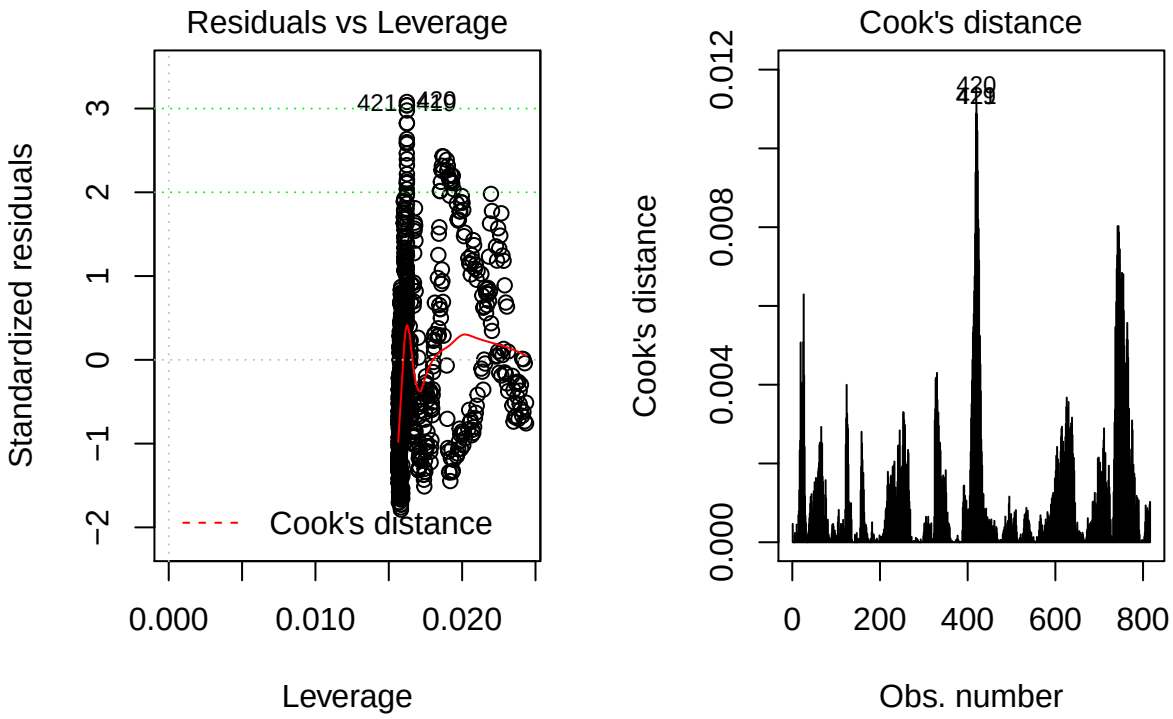



```
shapiro.test(unem.lm.quad$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  unem.lm.quad$residuals
## W = 0.96731, p-value = 1.513e-12
```

7. Outlier Analysis (Not an CLM assumption): From the scale-location plot above, we see a number of standardized residuals lying outside the threshold of 2 and 3 standard deviations. This says that our model represents variations in our data poorly. None of the observations are close to cook's distance of 0.5 so there are no extreme outliers.

```
par(mfrow = c(1,2))
plot(unem.lm.quad, which = 5)
abline(h = c(2,3), col = "green", lty = "dotted")
plot(unem.lm.quad, which = 4)
```



(B.i) How well does your model predict the unemployment rate up until June 2017? (B.ii) What does the unemployment rate look like at the end of 2020? How credible is this estimate? (B.iii) Compare this forecast to the one produced by the SARIMA model. What do you notice?

Prediction up until 2017

```
test.time = seq(2016,2017.417,by = (1/12))
test.month = c(seq(1,12),seq(1,6))

test.prediction = predict.lm(object = unem.lm.quad, se.fit = T,
           newdata = data.frame(t.lm = test.time, mon.lm = as.factor(test.month)))
```

```
data.frame("year" = floor(test.time),
           "month" = test.month,
           "lm mean" = test.prediction$fit,
           "lm error" = test.prediction$fit - unem.test$UNRATENSA,
           "SARIMA mean" = as.numeric(f.diff.3$mean),
           "SARIMA error" = as.numeric(f.diff.3$mean) - unem.test$UNRATENSA)
```

```
##    year month  lm.mean lm.error SARIMA.mean SARIMA.error
## 1  2016     1 6.891331 1.591331    5.498936  0.198936381
## 2  2016     2 6.877627 1.677627    5.333462  0.133461831
## 3  2016     3 6.596276 1.496276    5.098298 -0.001701837
## 4  2016     4 6.079631 1.379631    4.533827 -0.166172602
## 5  2016     5 5.930634 1.430634    4.683256  0.183255574
## 6  2016     6 6.528695 1.428695    5.037484 -0.062515551
```

```
## 7  2016     7 6.363520 1.263520    5.125235  0.025234937
## 8  2016     8 6.052758 1.052758    4.844588 -0.155411579
## 9  2016     9 5.872877 1.072877    4.500414 -0.299585809
## 10 2016    10 5.701821 1.001821    4.384869 -0.315130538
## 11 2016    11 5.870470 1.470470    4.338786 -0.061213655
## 12 2016    12 5.906766 1.406766    4.360144 -0.139855527
## 13 2017     1 6.864047 1.764047    5.046995 -0.053004672
## 14 2017     2 6.850201 1.950201    4.881704 -0.018296486
## 15 2017     3 6.568709 1.968709    4.646968  0.046968193
## 16 2017     4 6.051923 1.951923    4.086067 -0.013932679
## 17 2017     5 5.902783 1.802783    4.233470  0.133469609
## 18 2017     6 6.500703 2.000703    4.584196  0.084195733
```

```
sarima.err = as.numeric(f.diff.3$mean) - unem.test$UNRATENSA
lm.err = test.prediction$fit - unem.test$UNRATENSA

sarima.rmse = sqrt(mean(sarima.err^2))
lm.rmse = sqrt(mean(lm.err^2))

cat("SARIMA RMSE: ",sarima.rmse, "\n")
```

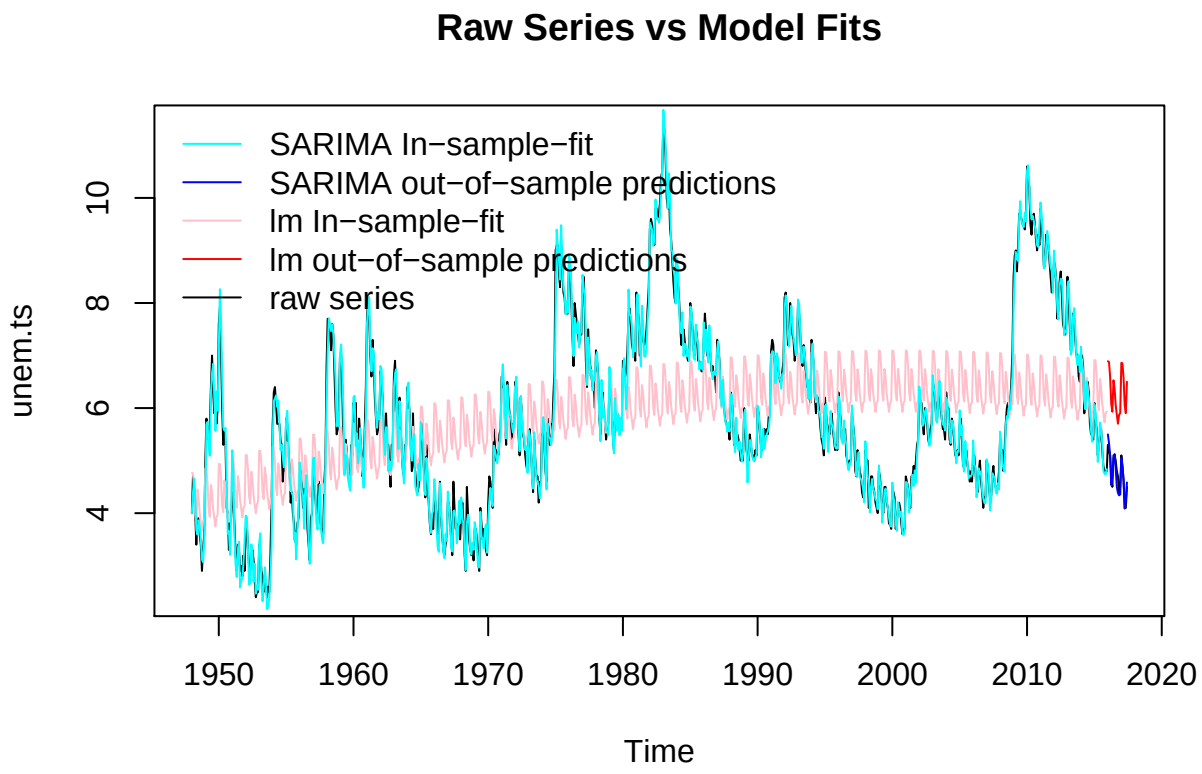
```
## SARIMA RMSE:  0.1472144
```

```
cat("Linear model RMSE: ",lm.rmse, "\n")
```

```
## Linear model RMSE:  1.571119
```

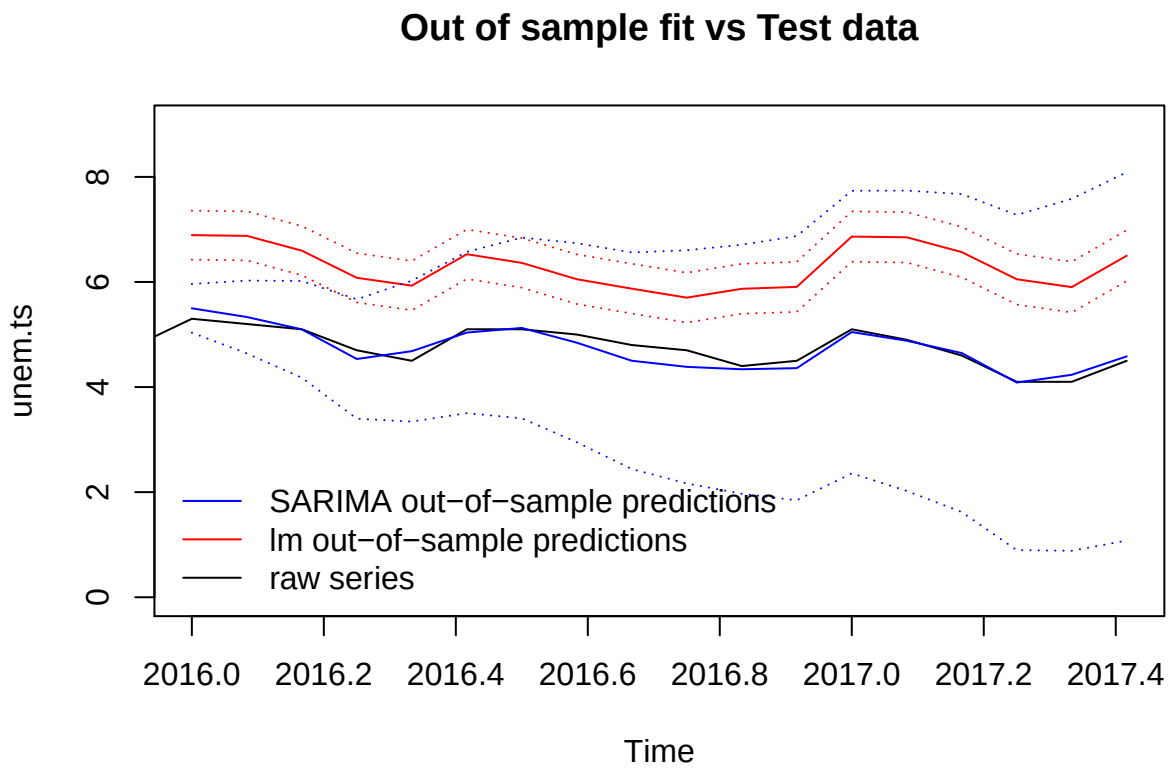
In comparison with the ARIMA model, the linear regression model has much poorer out-of-sample performance. The above table shows that the linear model predictions are consistently off by 1 to 2 units, while the SARIMA model predictions are only off from -0.4 to +0.2 units. Also, the RMSE of our SARIMA model is only less than 10% of the RMSE of the linear model.

```
plot(unem.ts, xlim = c(1948,2017.417),
     main = "Raw Series vs Model Fits")

lines(y = unem.lm.quad$fit, x = t.lm, col = "pink")
lines(y = test.prediction$fit, x = test.time, col = "red")

lines(y = m.diff.3$fitted, x = t.lm, col = "cyan")
lines(y = as.numeric(f.diff.3$mean), x = test.time, col = "blue")

legend("topleft", col = c("cyan","blue","pink","red","black"),
       legend = c("SARIMA In-sample-fit",
                  "SARIMA out-of-sample predictions",
                  "lm In-sample-fit",
                  "lm out-of-sample predictions",
                  "raw series"),
       bty = "n", lty = c("solid","solid","solid","solid","solid"))
```



The above long-term time plot shows that the fitted curve if our linear model fails to pick up most of the random walk variations in the raw series, while the step by step fit of the SARIMA model is quite close. Unless we fit a very high order term for time, it's impossible for the linear model to pick up such variations. On the other hand, seasonal patterns is reasonably approximated by both models, we can see the small ripples in the raw series replicated and matched by both fitted curves.

```
plot(unem.ts, xlim = c(2016,2017.417), ylim = c(0,9),
     main = "Out of sample fit vs Test data")

lines(y = test.prediction$fit, x = test.time, col = "red")
lines(y = test.prediction$fit + 1.96* test.prediction$se.fit,
      x = test.time, col = "red",lty = "dotted")
lines(y = test.prediction$fit - 1.96* test.prediction$se.fit,
      x = test.time, col = "red",lty = "dotted")

lines(y = f.diff.3$mean, x = test.time, col = "blue")
lines(y = as.numeric(f.diff.3$lower[,2]),
      x = test.time, col = "blue", lty = "dotted")
lines(y = as.numeric(f.diff.3$upper[,2]),
      x = test.time, col = "blue", lty = "dotted")

legend("bottomleft", col = c("blue","red","black"),
       legend = c("SARIMA out-of-sample predictions",
                  "lm out-of-sample predictions",
                  "raw series"),
       bty = "n", lty = c("solid","solid","solid"))
```



Within our test region, we see a clear trade-off between the two models. On the one hand, the mean estimates of the SARIMA model is quite close to the test data but its confidence bound widens drastically towards mid 2017 so our predictions lack precision. On the other hand for the linear model, even the lower bound of the estimates are consistently far above the test data. However, the confidence bound is quite narrow. Therefore, the SARIMA model makes less biased but less precise estimates and the linear model makes more biased but more precise estimates.

Forecast in 2020

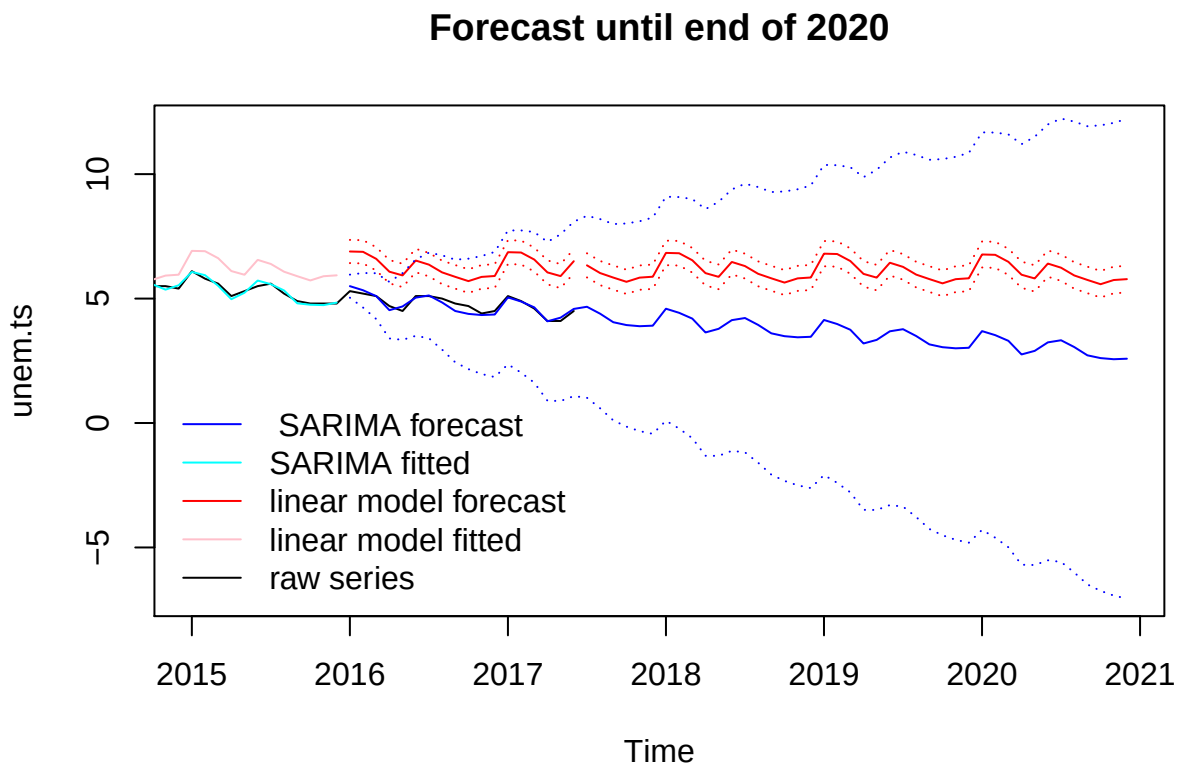
```
new.time = seq(2017.500,2020.917,by = (1/12))
new.month = c(seq(7,12),seq(1,12),seq(1,12),seq(1,12))

new.prediction = predict.lm(object = unem.lm.quad, se.fit = T,
           newdata = data.frame(t.lm = new.time, mon.lm = as.factor(new.month)))


new.df = data.frame("year" = floor(new.time),
           "month" = new.month,
           "lm mean estimate" = new.prediction$fit,
           "lm upper estimate" = new.prediction$fit + 1.96* new.prediction$se.fit,
           "lm lower estimate" = new.prediction$fit - 1.96* new.prediction$se.fit)

tail(new.df, 12)
```

```
##    year month lm.mean.estimate lm.upper.estimate lm.lower.estimate
```

```
## 31 2020     1         6.772009          7.294879          6.249138
## 32 2020     2         6.757739          7.281795          6.233683
## 33 2020     3         6.475822          7.001068          5.950576
## 34 2020     4         5.958612          6.485052          5.432171
## 35 2020     5         5.809048          6.336688          5.281408
## 36 2020     6         6.406543          6.935386          5.877700
## 37 2020     7         6.240803          6.770854          5.710752
## 38 2020     8         5.929474          6.460737          5.398211
## 39 2020     9         5.749028          6.281507          5.216549
## 40 2020    10         5.577406          6.111105          5.043706
## 41 2020    11         5.745489          6.280414          5.210565
## 42 2020    12         5.781220          6.317373          5.245066
```

The above table shows that forecast of the linear model in 2020 oscillates within 5.5 percent and 6.8 percent. Its confidence intervals don't seem to fluctuate much either.

```
plot(unem.ts, xlim = c(2015,2020.917), ylim = c(-7,12),
     main = "Forecast until end of 2020")

lines(y = unem.lm.quad$fit, x = t.lm, col = "pink")

lines(y = test.prediction$fit, x = test.time, col = "red")
lines(y = test.prediction$fit + 1.96* test.prediction$se.fit,
      x = test.time, col = "red",lty = "dotted")
lines(y = test.prediction$fit - 1.96* test.prediction$se.fit,
      x = test.time, col = "red",lty = "dotted")

lines(y = new.prediction$fit, x = new.time, col = "red")
lines(y = new.prediction$fit + 1.96* new.prediction$se.fit,
      x = new.time, col = "red",lty = "dotted")
lines(y = new.prediction$fit - 1.96* new.prediction$se.fit,
      x = new.time, col = "red",lty = "dotted")

lines(y = m.diff.3$fitted, x = t.lm, col = "cyan")

lines(y = f.diff.3.2020$mean, x = c(test.time,new.time), col = "blue")
lines(y = as.numeric(f.diff.3.2020$lower[,2]),
      x = c(test.time,new.time), col = "blue", lty = "dotted")
lines(y = as.numeric(f.diff.3.2020$upper[,2]),
      x = c(test.time,new.time), col = "blue", lty = "dotted")

legend("bottomleft", col = c("blue","cyan","red","pink", "black"),
       legend = c(" SARIMA forecast", "SARIMA fitted",
                  "linear model forecast", "linear model fitted",
                  "raw series"),
       bty = "n", lty = c("solid","solid","solid","solid"))
```



The time plot above show that the linear model's mean forecasts oscillates around level 6 without clear upward or downward overall trend. So does its confidence bound. On the other hand the SARIMA model's mean forecast continues the slight downward trend of the raw series and its confidence bound expands drastically to even below zero in 201, which doesn't make much sense. For these reasons, we deem neither models credible to forecast unemployment rate as far as the end of 2020.