USA Unemployment predictions

Importing dataset:

```
rm(list=ls())
data_usa=read.csv("USA_Unemployment.csv")
head(data_usa)
```

```
##     Dates Unemployment_Rate Labour_Force Employment_Rate CPI_Urban_Customers
## 1 1977-01             0.075     97208000       0.6456636                58.7
## 2 1977-02             0.076     97785000       0.6468344                59.3
## 3 1977-03             0.074     98115000       0.6491476                59.6
## 4 1977-04             0.072     98330000       0.6515833                60.0
## 5 1977-05             0.070     98665000       0.6526598                60.2
## 6 1977-06             0.072     99093000       0.6547839                60.5
##   Working_Age_Population Inactivity_Rates Unemployed_male Unemployed_female
## 1              135147081         0.301604         3920000           3360000
## 2              135350374         0.299043         4034000           3409000
## 3              135571699         0.297712         3847000           3460000
## 4              135839107         0.297630         3689000           3370000
## 5              136142088         0.297739         3729000           3182000
## 6              136339646         0.294509         3715000           3419000
##   Unemployed_all
## 1        7280000
## 2        7443000
## 3        7307000
## 4        7059000
## 5        6911000
## 6        7134000
```

```
tail(data_usa)
```

```
##       Dates Unemployment_Rate Labour_Force Employment_Rate CPI_Urban_Customers
## 550 2022-10             0.037    164646000       0.7124190             297.987
## 551 2022-11             0.036    164527000       0.7131987             298.598
## 552 2022-12             0.035    164966000       0.7157694             298.990
## 553 2023-01             0.034    165832000       0.7170108             300.536
## 554 2023-02             0.036    166251000       0.7174641             301.648
## 555 2023-03             0.035    166731000       0.7191223             301.808
##     Working_Age_Population Inactivity_Rates Unemployed_male Unemployed_female
## 550              207461858         0.260060         3212000           2841000
## 551              207524882         0.259655         3236000           2764000
## 552              207531208         0.258100         2984000           2738000
## 553              208159165         0.256395         3147000           2546000
## 554              208277722         0.255079         3208000           2728000
## 555              223490114         0.253967         3223000           2617000
##     Unemployed_all
## 550        6053000
## 551        6000000
## 552        5722000
## 553        5694000
## 554        5936000
## 555        5839000
```

Feature Scaling - Creating a new variable:

```
data_usa$male_to_female_unemp=round((data_usa$Unemployed_male/data_usa$Unemployed_female),4)
head(data_usa[,c(11)])
```

```
## [1] 1.1667 1.1833 1.1118 1.0947 1.1719 1.0866
```

This is to incorporate the factor - whether female are getting more unemployed or not compared to males over the years - as an external variable for overall unemployment rate.

Checking Multicollinearity using VIFs:

```
suppressWarnings(library(regclass))
```

```
## Loading required package: bestglm
```

```
## Loading required package: leaps
```

```
## Loading required package: VGAM
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```
## Loading required package: rpart
```

```
## Loading required package: randomForest
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
## Important regclass change from 1.3:
## All functions that had a . in the name now have an _
## all.correlations -> all_correlations, cor.demo -> cor_demo, etc.
```

```
VIF(lm(formula = Unemployment_Rate ~ Labour_Force+Employment_Rate+CPI_Urban_Customers+Working_Age_
Population+Inactivity_Rates+Unemployed_all+male_to_female_unemp, data = data_usa))
```

```
##          Labour_Force       Employment_Rate   CPI_Urban_Customers
##           1034.247763            420.770226             70.597817
## Working_Age_Population      Inactivity_Rates        Unemployed_all
##            707.580037            221.035376             91.230505
##   male_to_female_unemp
##              2.592428
```

Removing Labour_Force:

```
VIF(lm(formula = Unemployment_Rate ~ Employment_Rate+CPI_Urban_Customers+Working_Age_Population+In
activity_Rates+Unemployed_all+male_to_female_unemp, data = data_usa))
```

```
##        Employment_Rate    CPI_Urban_Customers Working_Age_Population
##             393.301669              37.621061              75.827802
##       Inactivity_Rates         Unemployed_all   male_to_female_unemp
##             220.869816              87.630731               2.555619
```

Removing Employment_Rate:

```
VIF(lm(formula = Unemployment_Rate ~ CPI_Urban_Customers+Working_Age_Population+Inactivity_Rates+U
nemployed_all+male_to_female_unemp, data = data_usa))
```

```
##    CPI_Urban_Customers Working_Age_Population       Inactivity_Rates
##              36.137200              36.961599               1.174889
##         Unemployed_all   male_to_female_unemp
##               1.798097               1.493898
```

Removing Working_Age_Population:

```
VIF(lm(formula = Unemployment_Rate ~ CPI_Urban_Customers+Inactivity_Rates+Unemployed_all+male_to_f
emale_unemp, data = data_usa))
```

```
##  CPI_Urban_Customers     Inactivity_Rates       Unemployed_all
##             1.083589             1.170820             1.690721
## male_to_female_unemp
##             1.485351
```

This is the final set of variables free from multicollinearity.

Train-Test split of the dataset - last 6 months of the data would be taken into testing part:

```
df_usa_train1=data_usa[1:(nrow(data_usa)-6),]
df_usa_test1=data_usa[(nrow(data_usa)-5):nrow(data_usa),]
head(df_usa_train1)
```

```
##     Dates Unemployment_Rate Labour_Force Employment_Rate CPI_Urban_Customers
## 1 1977-01             0.075     97208000       0.6456636                58.7
## 2 1977-02             0.076     97785000       0.6468344                59.3
## 3 1977-03             0.074     98115000       0.6491476                59.6
## 4 1977-04             0.072     98330000       0.6515833                60.0
## 5 1977-05             0.070     98665000       0.6526598                60.2
## 6 1977-06             0.072     99093000       0.6547839                60.5
##   Working_Age_Population Inactivity_Rates Unemployed_male Unemployed_female
## 1              135147081         0.301604         3920000           3360000
## 2              135350374         0.299043         4034000           3409000
## 3              135571699         0.297712         3847000           3460000
## 4              135839107         0.297630         3689000           3370000
## 5              136142088         0.297739         3729000           3182000
## 6              136339646         0.294509         3715000           3419000
##   Unemployed_all male_to_female_unemp
## 1        7280000               1.1667
## 2        7443000               1.1833
## 3        7307000               1.1118
## 4        7059000               1.0947
## 5        6911000               1.1719
## 6        7134000               1.0866
```

- The model would be trained on the train dataset.
- And the performance of the fitted model would be checked on the test dataset.
- If this performs fairly well, this model would be considered to get the future forecasts.

Time series plot:

```
suppressWarnings(library(fpp2))
```

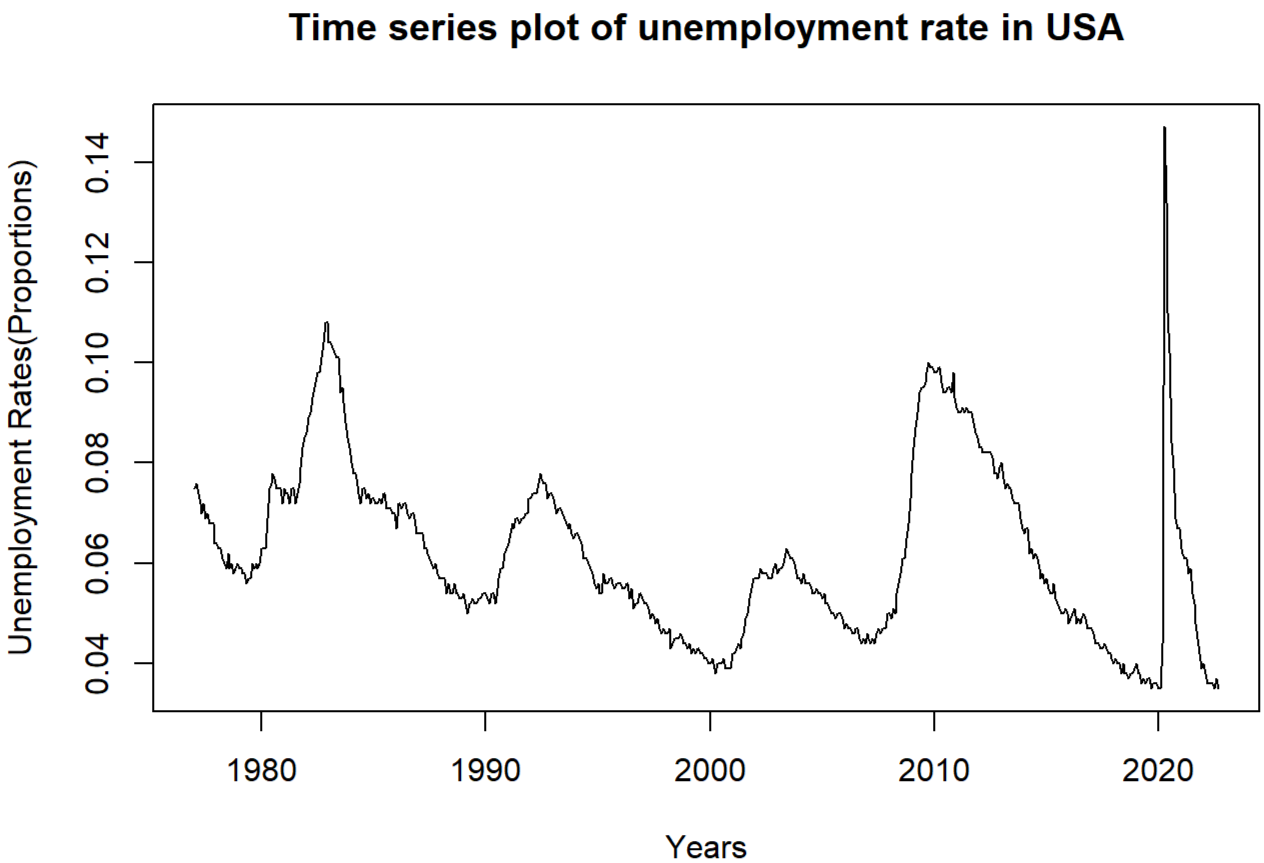
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## ── Attaching packages ──────────────────────────────────────────── fpp2 2.5 ──
```

```
## ✔ ggplot2   3.3.6     ✔ fma       2.4
## ✔ forecast  8.18      ✔ expsmooth 2.3
```

```
## ── Conflicts ───────────────────────────────────────────────── fpp2_conflicts ──
## ✖ ggplot2::margin() masks randomForest::margin()
```

```
suppressWarnings(library(urca))
df.ts=ts(df_usa_train1$Unemployment_Rate, frequency = 12, start = c(1977,1))
plot(df.ts,xlab="Years",ylab="Unemployment Rates(Proportions)")
title(main="Time series plot of unemployment rate in USA")
```



Testing stationarity:

```
df_usa_train1[,"Unemployment_Rate"] %>%
  ur.kpss() %>%
  summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.8804
##
## Critical value for a significance level of:
##                  10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

This series is non-stationary - 1st order differencing would be necessary.

Testing stationarity after 1st order differencing:

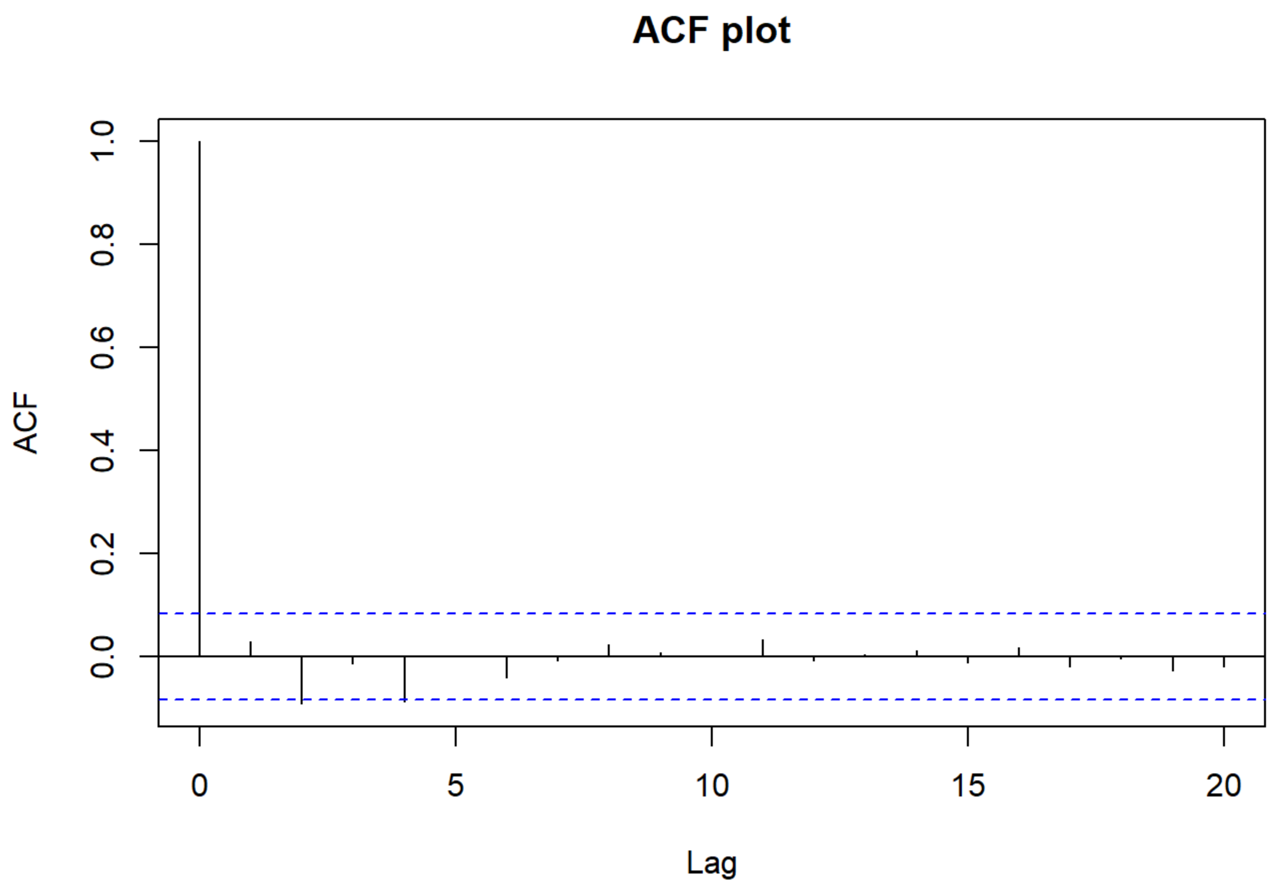
```
diff(df_usa_train1[,"Unemployment_Rate"]) %>%
  ur.kpss() %>%
  summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.0337
##
## Critical value for a significance level of:
##                  10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

The 1st order differences are stationary.

ACF plot:

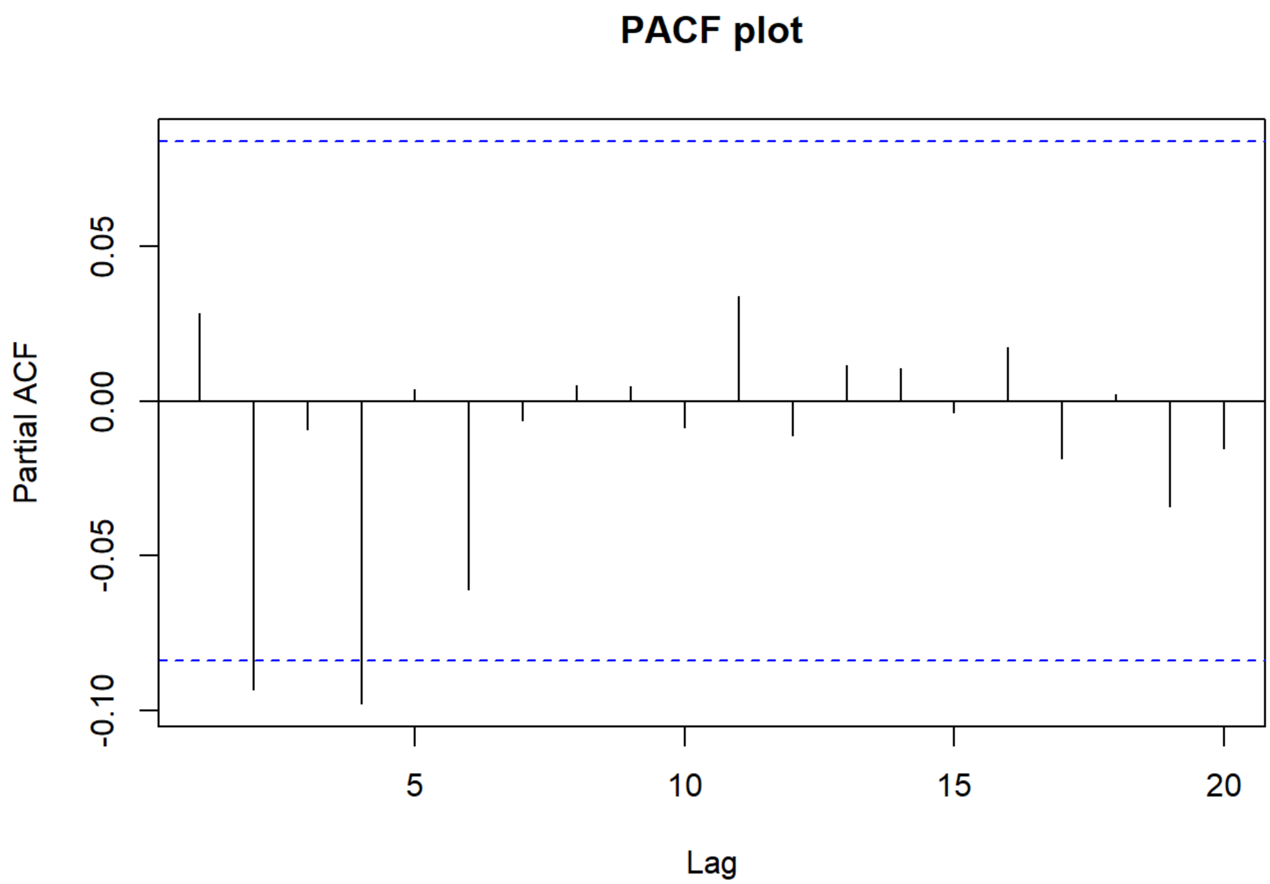
```
par(mfrow=c(1,1))
acf(diff(df_usa_train1$Unemployment_Rate), lag.max = 20, main = "ACF plot")
```



p can be taken as 0 or 2 based on the no. of significant lags.

PACF plot:

```
par(mfrow=c(1,1))
pacf(diff(df_usa_train1$Unemployment_Rate), lag.max = 20, main = "PACF plot")
```



q can be 0/2/4, based on the no. of significant lags.

Fitting ARIMAX model ignoring the variables that were eliminated due to high VIF:

Starting with the value of p & q as 2 and with the rest of the regressors:

```
est_train=arima(df_usa_train1$Unemployment_Rate, order=c(2,1,2), xreg = as.matrix(df_usa_train1[,c
(5,7,10,11)]), method = "ML")
summary(est_train)
```

```
##
## Call:
## arima(x = df_usa_train1$Unemployment_Rate, order = c(2, 1, 2), xreg = as.matrix(df_usa_train1[,
##     c(5, 7, 10, 11)]), method = "ML")
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##          ar1      ar2      ma1     ma2  CPI_Urban_Customers  Inactivity_Rates
##       1.1141  -0.2348  -1.4018  0.5876                    0            0.0081
## s.e.  0.1291   0.1237   0.1120  0.1072                  NaN            0.0025
##       Unemployed_all  male_to_female_unemp
##                    0                0.0013
## s.e.             NaN                0.0004
##
## sigma^2 estimated as 2.492e-07:  log likelihood = 3388.34,  aic = -6758.68
##
## Training set error measures:
##                         ME         RMSE          MAE         MPE     MAPE
## Training set -3.100807e-05 0.0004987726 0.0003870953 -0.05083704 0.646192
##                   MASE         ACF1
## Training set 0.2438255 -0.009527048
```

Test of significance of individual coefficients:

```
suppressWarnings(library(lmtest))
```

```
## Loading required package: zoo
```

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

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

```
##
## Attaching package: 'lmtest'
```

```
## The following object is masked from 'package:VGAM':
##
##     lrtest
```

```
coeftest(est_train)
```

```
## Warning in sqrt(diag(se)): NaNs produced
```

```
##
## z test of coefficients:
##
##                         Estimate  Std. Error  z value  Pr(>|z|)
## ar1                   1.1141e+00  1.2912e-01   8.6283 < 2.2e-16 ***
## ar2                  -2.3478e-01  1.2369e-01  -1.8981 0.0576798 .
## ma1                  -1.4018e+00  1.1198e-01 -12.5188 < 2.2e-16 ***
## ma2                   5.8756e-01  1.0715e-01   5.4834 4.173e-08 ***
## CPI_Urban_Customers  -1.0458e-05         NaN      NaN       NaN
## Inactivity_Rates      8.1123e-03  2.4503e-03   3.3107 0.0009305 ***
## Unemployed_all        6.5061e-09         NaN      NaN       NaN
## male_to_female_unemp  1.2983e-03  4.4057e-04   2.9468 0.0032113 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We need to remove the variables producing NaNs & the insignificant variables.

After doing that, the summary & test of significances of the final model would look like:

```
est_1=arima(df_usa_train1$Unemployment_Rate, order=c(1,1,2), xreg = as.matrix(df_usa_train1[,c(7,1
0,11)]), method = "ML")
summary(est_1)
```

```
##
## Call:
## arima(x = df_usa_train1$Unemployment_Rate, order = c(1, 1, 2), xreg = as.matrix(df_usa_train1[,
##     c(7, 10, 11)]), method = "ML")
##
## Coefficients:
##          ar1      ma1     ma2  Inactivity_Rates  Unemployed_all
##       0.8856  -1.1989  0.3846            0.0080               0
## s.e.  0.0384   0.0545  0.0447            0.0026               0
##       male_to_female_unemp
##                     0.0013
## s.e.                0.0005
##
## sigma^2 estimated as 2.509e-07:  log likelihood = 3386.54,  aic = -6759.08
##
## Training set error measures:
##                         ME         RMSE          MAE         MPE      MAPE
## Training set -3.214901e-05 0.0005004425 0.0003875992 -0.05301923 0.6464173
##                   MASE       ACF1
## Training set 0.2441429 0.01940615
```

Test of significance of coefficients:

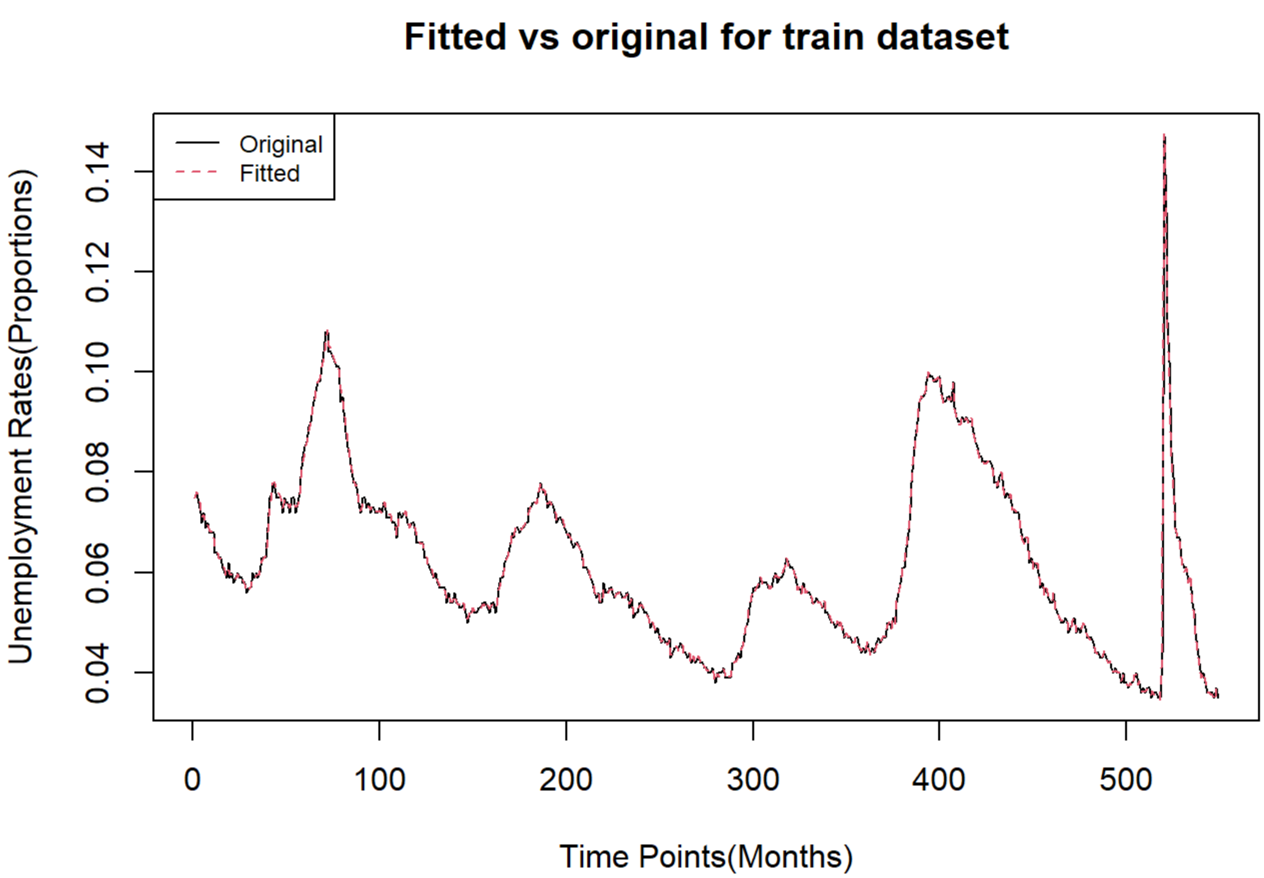
```
suppressWarnings(library(lmtest))
coeftest(est_1)
```

```
##
## z test of coefficients:
##
##                        Estimate  Std. Error  z value  Pr(>|z|)
## ar1                  8.8559e-01  3.8416e-02  23.0525 < 2.2e-16 ***
## ma1                 -1.1989e+00  5.4503e-02 -21.9970 < 2.2e-16 ***
## ma2                  3.8456e-01  4.4743e-02   8.5949 < 2.2e-16 ***
## Inactivity_Rates     8.0043e-03  2.6169e-03   3.0586  0.002223 **
## Unemployed_all       6.5121e-09  1.4973e-11 434.9242 < 2.2e-16 ***
## male_to_female_unemp  1.3232e-03  4.7798e-04   2.7684  0.005633 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Thus all the final parameters are kept which are significant in prediction of the target variable.

Plot of Fitted vs Original values for train dataset:

```
res=residuals(est_1)
data_fit=df_usa_train1$Unemployment_Rate-res
ts.plot(df_usa_train1$Unemployment_Rate, type="l", xlab="Time Points(Months)", ylab="Unemployment Rates(Proportions)", main="Fitted vs original for train dataset")
points(data_fit, type="l", col=2, lty=2)
legend("topleft",c("Original","Fitted"), col=c(1,2), lty=c(1,2), cex=0.75)
```



Predictions of unemployment rates for the test dataset

using above fitted model:

```
test_pred=predict(est_1, n.ahead=6, newxreg = as.matrix(df_usa_test1[, c(7,10,11)]), se.fit=FALSE,
method="ML")
```

Predicted values:

```
print(as.vector(test_pred))
```

```
## [1] 0.03695535 0.03668309 0.03477340 0.03478898 0.03629079 0.03573789
```

Original values:

```
print(df_usa_test1$Unemployment_Rate)
```

```
## [1] 0.037 0.036 0.035 0.034 0.036 0.035
```

Performane on test dataset:

MAPE:

```
(1/length(df_usa_test1$Unemployment_Rate))*(sum(abs(df_usa_test1$Unemployment_Rate-as.vector(test_
pred))/abs(df_usa_test1$Unemployment_Rate)))*100
```

```
## [1] 1.317021
```

RMSE:

```
sqrt(mean((df_usa_test1$Unemployment_Rate-as.vector(test_pred))^2))
```

```
## [1] 0.0005433676
```

Thus, it is working more or less well for future datasets.

Now going with the same approach with the actual dataset for getting the future forecast of May,23:

Checking stationarity:

```
data_usa[,"Unemployment_Rate"] %>%
  ur.kpss() %>%
  summary()
```

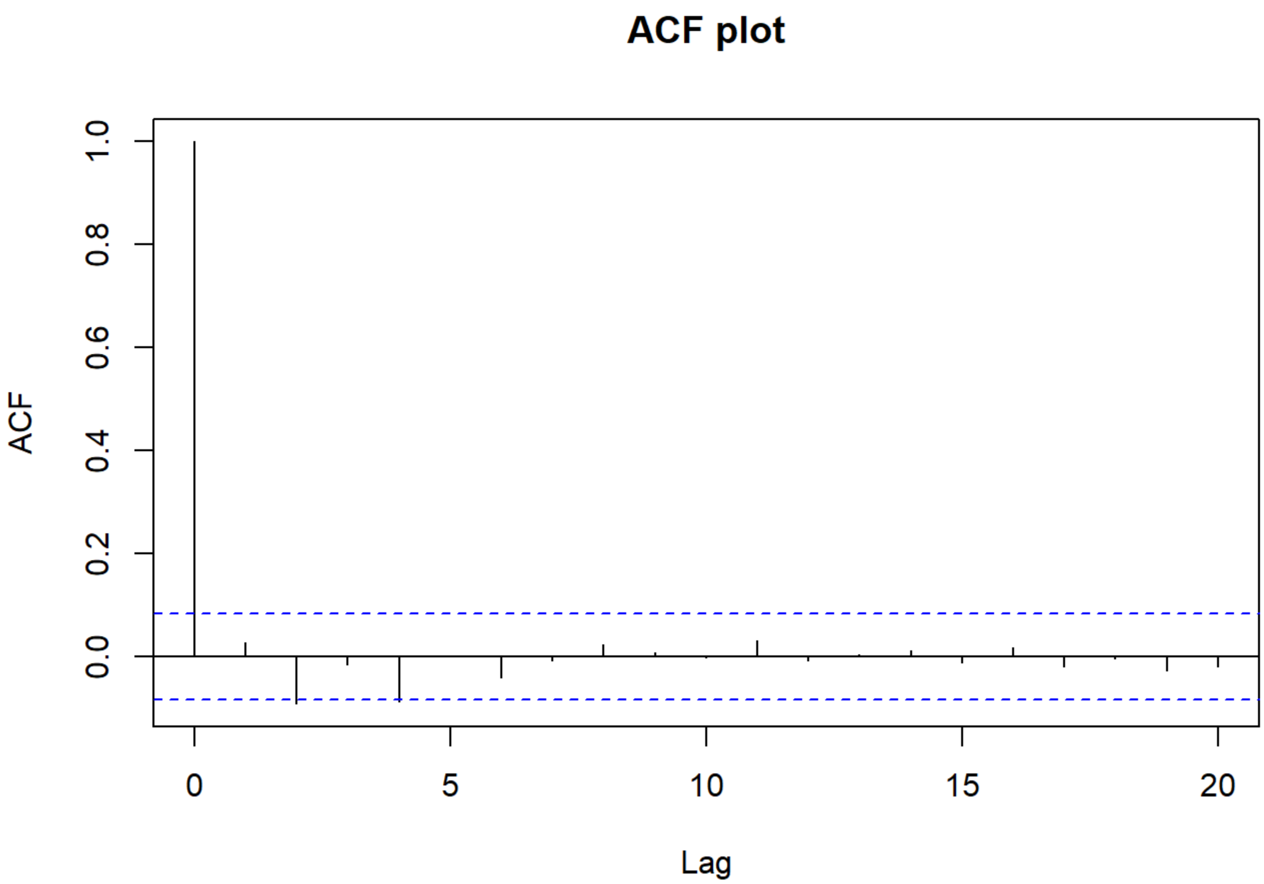
```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.953
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
diff(data_usa[,"Unemployment_Rate"]) %>%
  ur.kpss() %>%
  summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.0329
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

ACF plot:

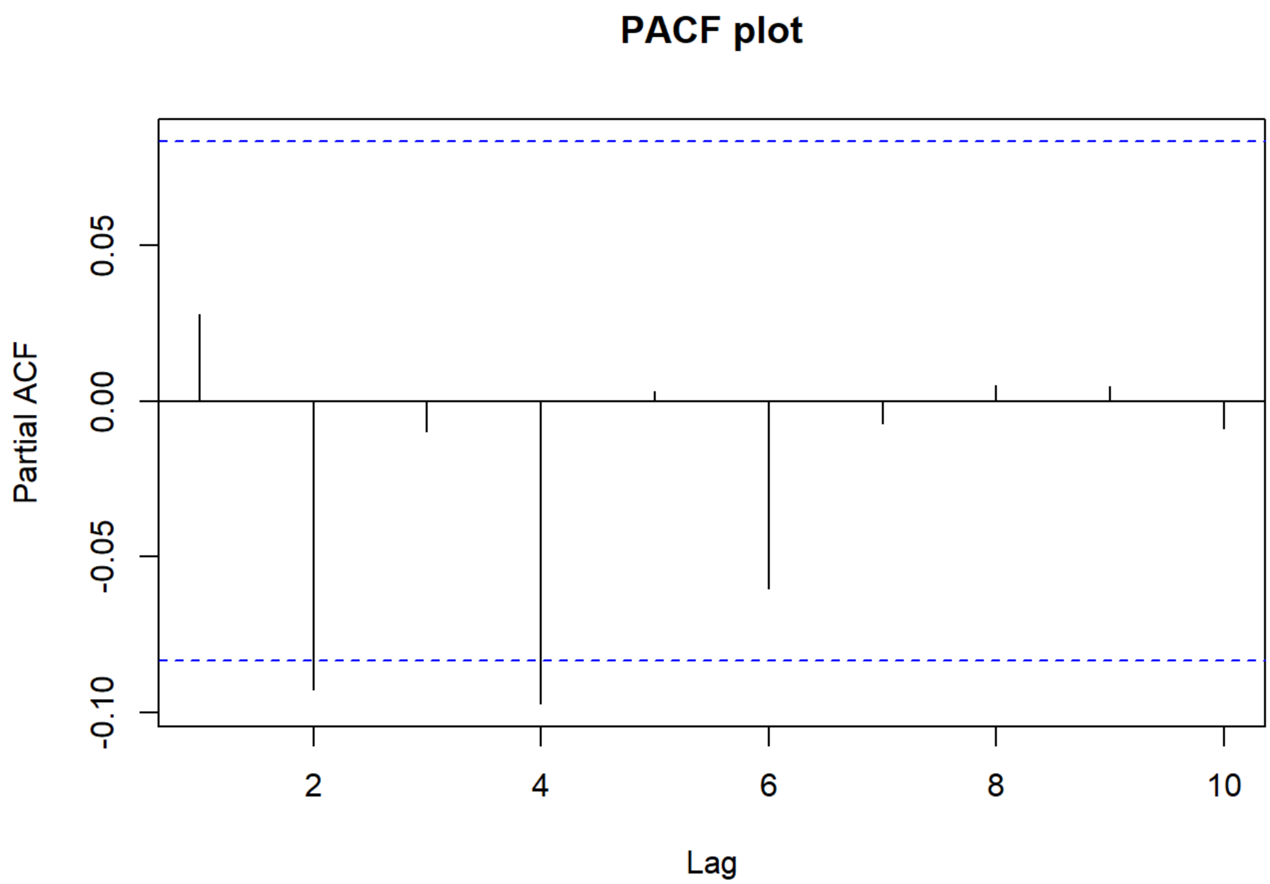
```
par(mfrow=c(1,1))
acf(diff(data_usa$Unemployment_Rate), lag.max = 20, main = "ACF plot")
```



p can be taken as 0/1/2, based on the no. of significant lags.

PACF plot:

```
par(mfrow=c(1,1))
pacf(diff(data_usa$Unemployment_Rate), lag.max = 10, main = "PACF plot")
```



q can be taken as 0/2/4, based on the no. of significant lags.

Fitting the model that we tested before - on the actual data:

```
est_2=arima(data_usa$Unemployment_Rate, order=c(1,1,2), xreg = as.matrix(data_usa[,c(7,10,11)]), m
ethod = "ML")
summary(est_2)
```

```
##
## Call:
## arima(x = data_usa$Unemployment_Rate, order = c(1, 1, 2), xreg = as.matrix(data_usa[,
##     c(7, 10, 11)]), method = "ML")
##
## Coefficients:
##          ar1      ma1     ma2  Inactivity_Rates  Unemployed_all
##       0.8830  -1.2014  0.3900            0.0086               0
## s.e.  0.0387   0.0544  0.0439            0.0026               0
##       male_to_female_unemp
##                     0.0012
## s.e.                0.0005
##
## sigma^2 estimated as 2.511e-07:  log likelihood = 3423.37,  aic = -6832.73
##
## Training set error measures:
##                         ME         RMSE          MAE         MPE      MAPE
## Training set -3.333523e-05 0.0005006755 0.0003880647 -0.05660767 0.6525266
##                   MASE       ACF1
## Training set 0.2448609 0.01671521
```

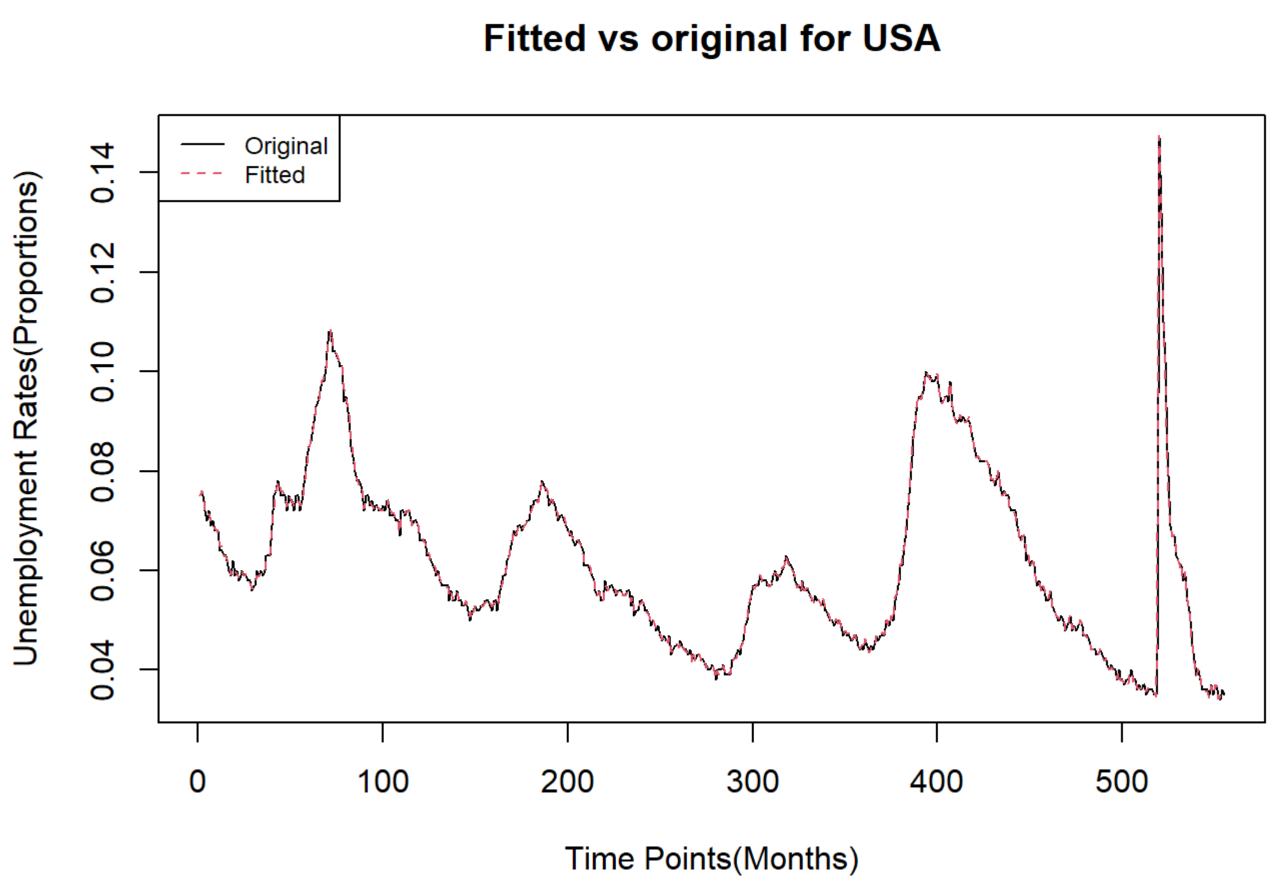
Test of significance of individual coefficients:

```
suppressWarnings(library(lmtest))
coeftest(est_2)
```

```
##
## z test of coefficients:
##
##                         Estimate  Std. Error  z value  Pr(>|z|)
## ar1                   8.8300e-01  3.8685e-02  22.8256 < 2.2e-16 ***
## ma1                  -1.2014e+00  5.4377e-02 -22.0931 < 2.2e-16 ***
## ma2                   3.8998e-01  4.3948e-02   8.8738 < 2.2e-16 ***
## Inactivity_Rates      8.5623e-03  2.5832e-03   3.3146 0.0009178 ***
## Unemployed_all        6.5090e-09  1.4765e-11 440.8317 < 2.2e-16 ***
## male_to_female_unemp  1.1804e-03  4.7279e-04   2.4967 0.0125363 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plot of Fitted vs Original on the actual data:

```
res=residuals(est_2)
data_fit=data_usa$Unemployment_Rate-res
ts.plot(data_usa$Unemployment_Rate, type="l", xlab="Time Points(Months)", ylab="Unemployment Rates
(Proportions)", main="Fitted vs original for USA")
points(data_fit, type="l", col=2, lty=2)
legend("topleft",c("Original","Fitted"), col=c(1,2), lty=c(1,2), cex=0.75)
```



Need forecasts of Inactivity rates, overall unemployment level & the uemployment level of males, females for April, May 2023 to use them for getting the forecast of Unemployment rates.

```
auto.arima(data_usa$Inactivity_Rates)
```

```
## Series: data_usa$Inactivity_Rates
## ARIMA(1,2,2)
##
## Coefficients:
##          ar1      ma1     ma2
##       0.5759  -1.7326  0.7395
## s.e.  0.1043   0.0845  0.0837
##
## sigma^2 = 3.675e-06:  log likelihood = 2674.93
## AIC=-5341.87   AICc=-5341.79   BIC=-5324.61
```

```
est_inactivity_rates=arima(data_usa$Inactivity_Rates, order=c(1,2,2), method = "ML")
future_inactivity_rates=predict(est_inactivity_rates, n.ahead=2, se.fit=FALSE, method="ML")
print(future_inactivity_rates)
```

```
## Time Series:
## Start = 556
## End = 557
## Frequency = 1
## [1] 0.2542647 0.2543310
```

```
auto.arima(data_usa$Unemployed_all)
```

```
## Series: data_usa$Unemployed_all
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1     ar2      ma1      ma2
##       0.1755  0.4709  -0.1387  -0.6135
## s.e.  0.1876  0.1929   0.1678   0.1723
##
## sigma^2 = 5.496e+11:  log likelihood = -8272.1
## AIC=16554.2   AICc=16554.31   BIC=16575.79
```

```
est_Unemployed_all=arima(data_usa$Unemployed_all, order=c(2,1,2), method = "ML")
future_Unemployed_all=predict(est_Unemployed_all, n.ahead=2, se.fit=FALSE, method="ML")
print(future_Unemployed_all)
```

```
## Time Series:
## Start = 556
## End = 557
## Frequency = 1
## [1] 5826283 5869699
```

```
auto.arima(data_usa$male_to_female_unemp)
```

```
## Series: data_usa$male_to_female_unemp
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ma1    mean
##       0.9622  -0.4083  1.2075
## s.e.  0.0120   0.0395  0.0282
##
## sigma^2 = 0.001979:  log likelihood = 940.56
## AIC=-1873.13   AICc=-1873.05   BIC=-1855.85
```

```
est_male_to_female_unemp=arima(data_usa$male_to_female_unemp, order=c(1,0,1), method = "ML")
future_male_to_female_unemp=predict(est_male_to_female_unemp, n.ahead=2, se.fit=FALSE, method="M
L")
print(future_male_to_female_unemp)
```

```
## Time Series:
## Start = 556
## End = 557
## Frequency = 1
## [1] 1.210746 1.210623
```

```
april_may_inputs=data.frame(as.vector(future_inactivity_rates), as.vector(future_Unemployed_all),
as.vector(future_male_to_female_unemp))
```

Obtaining prediction of Unemployment rate for May 2023:

```
future_unemp_pred=predict(est_2, n.ahead=2, newxreg = as.matrix(april_may_inputs[, c(1,2,3)]), se.
fit=FALSE, method="ML")
print(as.vector(future_unemp_pred)[2])
```

```
## [1] 0.03519756
```

Upper & Lower limits (95% C.I.s):

```
upper=as.vector(future_unemp_pred)+(1.96*(sqrt(est_2$sigma2)))
lower=as.vector(future_unemp_pred)-(1.96*(sqrt(est_2$sigma2)))
```

Upper limit for May 2023 forecast:

```
print(as.vector(upper)[2])
```

```
## [1] 0.03617977
```

Lower limit for May 2023 forecast:

```
print(as.vector(lower)[2])
```

```
## [1] 0.03421535
```

May 23 forecast - 3.52 %

Upper & Lower limits - (3.42 %, 3.62 %)