UK Unemployment predictions

Importing dataset:

```
rm(list=ls())
data_uk=read.csv("UK_Unemployment.csv")
head(data_uk)
```

```
##       DATE Unemployed_Over_12_months Monthly_GDP CPI_Inflation_Rate
## 1 1997 JAN                    863000        63.8              0.021
## 2 1997 FEB                    840000        64.8              0.019
## 3 1997 MAR                    813000        64.9              0.017
## 4 1997 APR                    789000        65.2              0.016
## 5 1997 MAY                    771000        64.7              0.016
## 6 1997 JUN                    746000        65.2              0.017
##   Unemployment_rate Employment_Rate Employment Avg_actual_weekly_hours_of_work
## 1             0.075           0.706   26258000                            38.5
## 2             0.073           0.708   26324000                            38.6
## 3             0.072           0.709   26381000                            38.8
## 4             0.072           0.709   26428000                            38.7
## 5             0.072           0.710   26450000                            38.6
## 6             0.073           0.709   26514000                            38.5
##   Temporary_Workers Economic_Inactivity_Rate Unemployed_Female Unemployed_Male
## 1           1701000                    0.236            817000         1364000
## 2           1714000                    0.236            800000         1331000
## 3           1739000                    0.236            786000         1298000
## 4           1761000                    0.236            784000         1268000
## 5           1765000                    0.236            763000         1284000
## 6           1792000                    0.234            783000         1267000
```

```
tail(data_uk)
```

```
##          DATE Unemployed_Over_12_months Monthly_GDP CPI_Inflation_Rate
## 309 2022 SEP                    309000        99.7              0.101
## 310 2022 OCT                    284000       100.4              0.111
## 311 2022 NOV                    274000       100.5              0.107
## 312 2022 DEC                    270000       100.0              0.105
## 313 2023 JAN                    289000       100.4              0.101
## 314 2023 FEB                    298000       100.4              0.104
##     Unemployment_rate Employment_Rate Employment
## 309             0.037           0.756   32739000
## 310             0.037           0.756   32773000
## 311             0.037           0.756   32781000
## 312             0.037           0.757   32813000
## 313             0.037           0.758   32839000
## 314             0.038           0.756   32950000
##     Avg_actual_weekly_hours_of_work Temporary_Workers Economic_Inactivity_Rate
## 309                            36.2           1620000                    0.216
## 310                            36.2           1670000                    0.215
## 311                            36.2           1690000                    0.215
## 312                            36.4           1655000                    0.214
## 313                            36.6           1658000                    0.213
## 314                            36.6           1650000                    0.211
##     Unemployed_Female Unemployed_Male
## 309            576000          648000
## 310            575000          672000
## 311            565000          679000
## 312            589000          681000
## 313            578000          674000
## 314            593000          700000
```

Feature Scaling - Creating a new variable:

```
data_uk$male_to_female_unemp=round((data_uk$Unemployed_Male/data_uk$Unemployed_Female),4)
print(data_uk$male_to_female_unemp[10])
```

```
## [1] 1.5959
```

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 ~ Unemployed_Over_12_months+Monthly_GDP+CPI_Inflation_Rate+Empl
oyment_Rate+Employment+Avg_actual_weekly_hours_of_work+Temporary_Workers+Economic_Inactivity_Rate+
male_to_female_unemp, data = data_uk))
```

```
##       Unemployed_Over_12_months                     Monthly_GDP
##                       16.561283                       59.855401
##              CPI_Inflation_Rate                 Employment_Rate
##                        1.486485                       50.991741
##                      Employment Avg_actual_weekly_hours_of_work
##                      129.794988                        5.950788
##               Temporary_Workers        Economic_Inactivity_Rate
##                        2.957720                       41.083590
##            male_to_female_unemp
##                        8.324337
```

Removing Employment:

```
VIF(lm(formula = Unemployment_rate ~ Unemployed_Over_12_months+Monthly_GDP+CPI_Inflation_Rate+Empl
oyment_Rate+Avg_actual_weekly_hours_of_work+Temporary_Workers+Economic_Inactivity_Rate+male_to_fem
ale_unemp, data = data_uk))
```

```
##       Unemployed_Over_12_months                     Monthly_GDP
##                       16.440794                       13.846021
##              CPI_Inflation_Rate                 Employment_Rate
##                        1.483674                       50.776713
## Avg_actual_weekly_hours_of_work               Temporary_Workers
##                        2.448179                        2.916613
##        Economic_Inactivity_Rate            male_to_female_unemp
##                       28.765769                        8.290965
```

Removing Employment_Rate:

```
VIF(lm(formula = Unemployment_rate ~ Unemployed_Over_12_months+Monthly_GDP+CPI_Inflation_Rate+Avg_
actual_weekly_hours_of_work+Temporary_Workers+Economic_Inactivity_Rate+male_to_female_unemp, data
= data_uk))
```

```
##       Unemployed_Over_12_months                     Monthly_GDP
##                        1.343520                       13.822688
##              CPI_Inflation_Rate Avg_actual_weekly_hours_of_work
##                        1.472392                        2.101835
##               Temporary_Workers        Economic_Inactivity_Rate
##                        2.289429                        7.419036
##            male_to_female_unemp
##                        7.611162
```

Removing Monthly GDP:

```
VIF(lm(formula = Unemployment_rate ~ Unemployed_Over_12_months+CPI_Inflation_Rate+Avg_actual_weekl
y_hours_of_work+Temporary_Workers+Economic_Inactivity_Rate+male_to_female_unemp, data = data_uk))
```

```
##       Unemployed_Over_12_months              CPI_Inflation_Rate
##                        1.233932                        1.088983
## Avg_actual_weekly_hours_of_work               Temporary_Workers
##                        1.623927                        1.283611
##        Economic_Inactivity_Rate            male_to_female_unemp
##                        4.238719                        3.792523
```

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_uk_train1=data_uk[1:(nrow(data_uk)-6),]
df_uk_test1=data_uk[(nrow(data_uk)-5):nrow(data_uk),]
head(df_uk_train1)
```

```
##       DATE Unemployed_Over_12_months Monthly_GDP CPI_Inflation_Rate
## 1 1997 JAN                    863000        63.8              0.021
## 2 1997 FEB                    840000        64.8              0.019
## 3 1997 MAR                    813000        64.9              0.017
## 4 1997 APR                    789000        65.2              0.016
## 5 1997 MAY                    771000        64.7              0.016
## 6 1997 JUN                    746000        65.2              0.017
##   Unemployment_rate Employment_Rate Employment Avg_actual_weekly_hours_of_work
## 1             0.075           0.706   26258000                            38.5
## 2             0.073           0.708   26324000                            38.6
## 3             0.072           0.709   26381000                            38.8
## 4             0.072           0.709   26428000                            38.7
## 5             0.072           0.710   26450000                            38.6
## 6             0.073           0.709   26514000                            38.5
##   Temporary_Workers Economic_Inactivity_Rate Unemployed_Female Unemployed_Male
## 1           1701000                    0.236            817000         1364000
## 2           1714000                    0.236            800000         1331000
## 3           1739000                    0.236            786000         1298000
## 4           1761000                    0.236            784000         1268000
## 5           1765000                    0.236            763000         1284000
## 6           1792000                    0.234            783000         1267000
##   male_to_female_unemp
## 1               1.6695
## 2               1.6638
## 3               1.6514
## 4               1.6173
## 5               1.6828
## 6               1.6181
```

- 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))
```

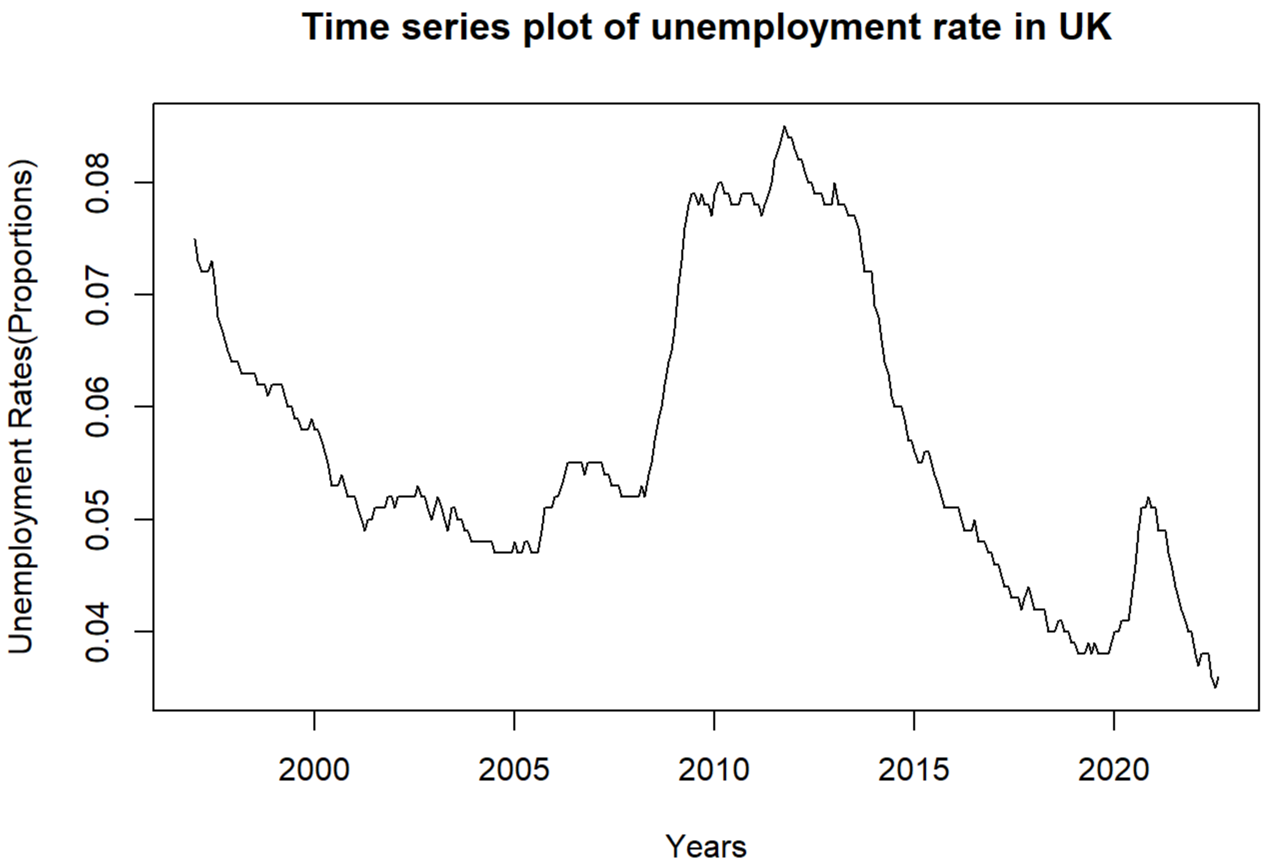
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## ── 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_uk_train1$Unemployment_rate, frequency = 12, start = c(1997,1))
plot(df.ts,xlab="Years",ylab="Unemployment Rates(Proportions)")
title(main="Time series plot of unemployment rate in UK")
```



Testing stationarity:

```
df_uk_train1[,"Unemployment_rate"] %>%
  ur.kpss() %>%
  summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.9483
##
## 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:

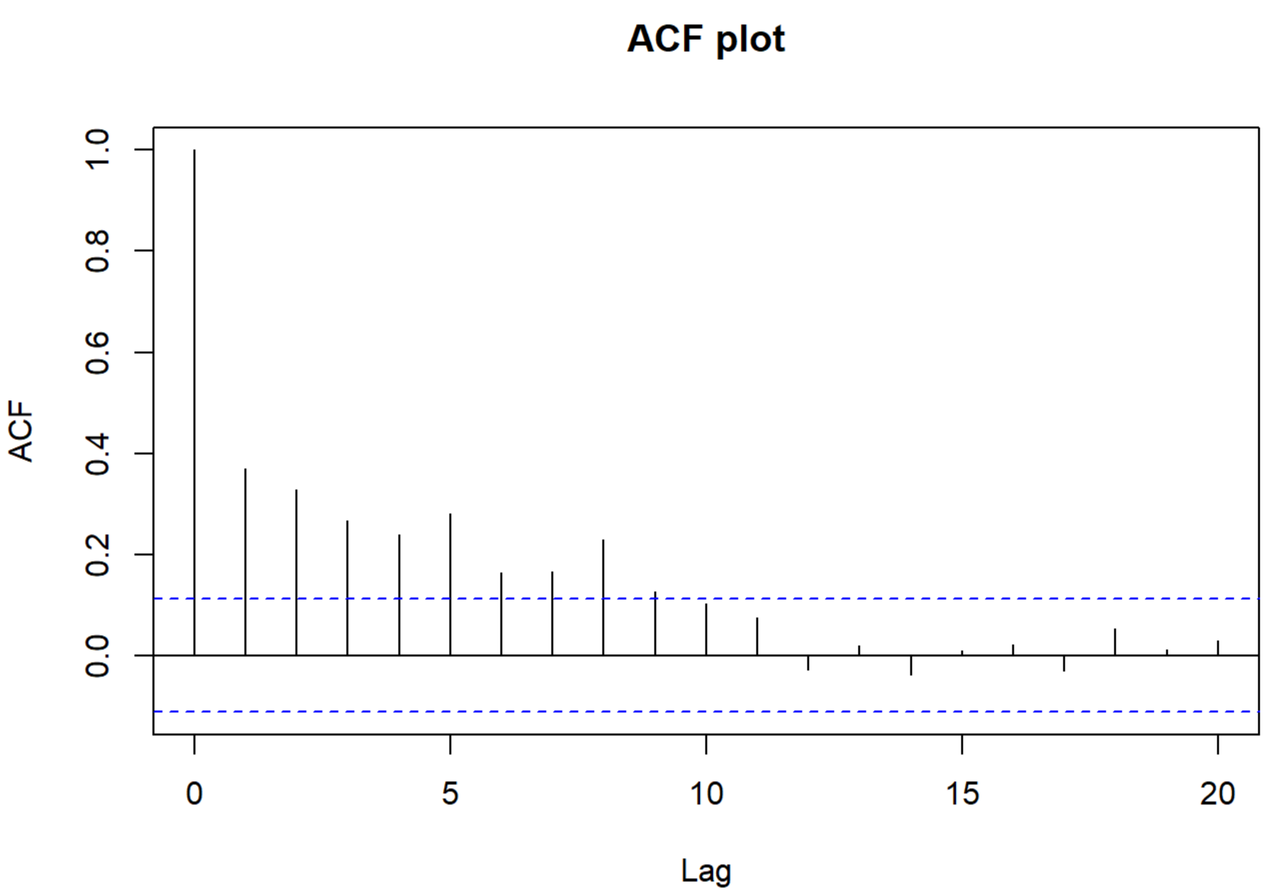
```
diff(df_uk_train1[,"Unemployment_rate"]) %>%
  ur.kpss() %>%
  summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.2604
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

1st order differences are stationary.

ACF plot:

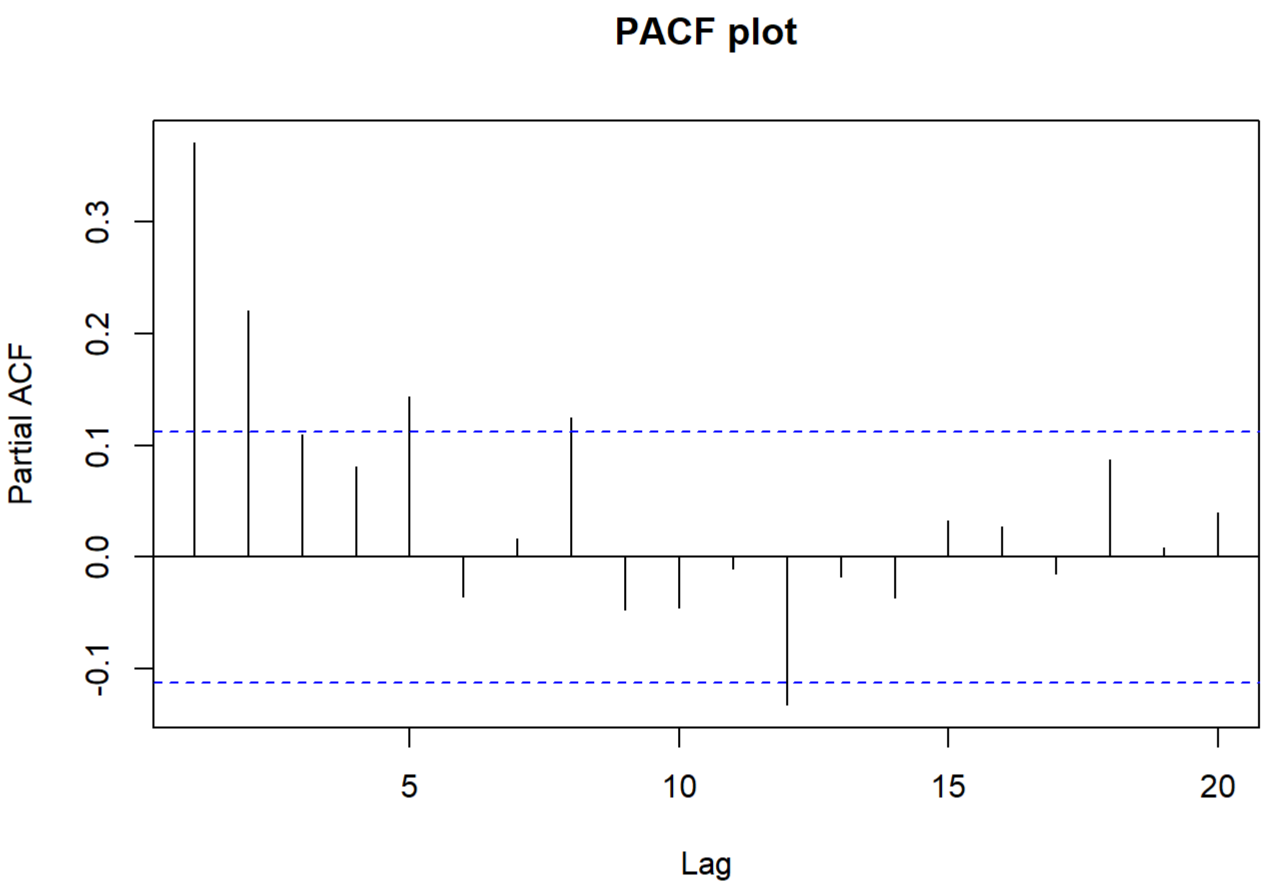
```
par(mfrow=c(1,1))
acf(diff(df_uk_train1$Unemployment_rate), lag.max = 20, main = "ACF plot")
```



p can be taken as 0/1/2/3/4 based on the no. of significant lags.

PACF plot:

```
par(mfrow=c(1,1))
pacf(diff(df_uk_train1$Unemployment_rate), lag.max = 20, main = "PACF plot")
```



q can be 0/1/2, 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 as 4 & q as 2 and with the rest of the regressors:

```
est_train=arima(df_uk_train1$Unemployment_rate, order=c(4,1,2), xreg = as.matrix(df_uk_train1[,c
(2,4,8,9,10,13)]), method = "ML")
summary(est_train)
```

```
##
## Call:
## arima(x = df_uk_train1$Unemployment_rate, order = c(4, 1, 2), xreg = as.matrix(df_uk_train1[,
##     c(2, 4, 8, 9, 10, 13)]), method = "ML")
##
## Coefficients:
```

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

```
##           ar1     ar2     ar3      ar4     ma1      ma2
##       -0.0044  0.7791  0.0419  -0.0124  0.2357  -0.5456
## s.e.   0.4054  0.3163  0.0896   0.0844  0.4063   0.2449
##       Unemployed_Over_12_months  CPI_Inflation_Rate
##                               0              0.0025
## s.e.                        NaN              0.0167
##       Avg_actual_weekly_hours_of_work  Temporary_Workers
##                                 4e-04                  0
## s.e.                            1e-04                NaN
##       Economic_Inactivity_Rate  male_to_female_unemp
##                        -0.0495                0.0030
## s.e.                    0.0228                0.0017
##
## sigma^2 estimated as 8.102e-07:  log likelihood = 1717.13,  aic = -3408.26
##
## Training set error measures:
##                        ME         RMSE          MAE         MPE     MAPE
## Training set -3.44748e-05 0.0008986814 0.0006971506 -0.05305371 1.283931
##                   MASE         ACF1
## Training set 0.9772842 -0.005538302
```

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                            -4.4440e-03  4.0537e-01 -0.0110 0.991253
## ar2                             7.7913e-01  3.1631e-01  2.4632 0.013772 *
## ar3                             4.1877e-02  8.9600e-02  0.4674 0.640226
## ar4                            -1.2389e-02  8.4355e-02 -0.1469 0.883240
## ma1                             2.3571e-01  4.0626e-01  0.5802 0.561790
## ma2                            -5.4563e-01  2.4489e-01 -2.2281 0.025876 *
## Unemployed_Over_12_months      -5.0639e-09         NaN     NaN      NaN
## CPI_Inflation_Rate              2.4962e-03  1.6685e-02  0.1496 0.881075
## Avg_actual_weekly_hours_of_work  3.8369e-04  1.4804e-04  2.5917 0.009549 **
## Temporary_Workers              -2.4379e-09         NaN     NaN      NaN
## Economic_Inactivity_Rate       -4.9530e-02  2.2757e-02 -2.1764 0.029523 *
## male_to_female_unemp            2.9634e-03  1.6919e-03  1.7515 0.079852 .
## ---
## 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_uk_train1$Unemployment_rate, order=c(1,1,1), xreg = as.matrix(df_uk_train1[,c(8,1
3)]), method = "ML")
summary(est_1)
```

```
##
## Call:
## arima(x = df_uk_train1$Unemployment_rate, order = c(1, 1, 1), xreg = as.matrix(df_uk_train1[,
##     c(8, 13)]), method = "ML")
##
## Coefficients:
##          ar1      ma1  Avg_actual_weekly_hours_of_work  male_to_female_unemp
##       0.8939  -0.6560                            4e-04                0.0028
## s.e.  0.0414   0.0672                            2e-04                0.0017
##
## sigma^2 estimated as 8.181e-07:  log likelihood = 1715.68,  aic = -3421.36
##
## Training set error measures:
##                         ME         RMSE         MAE         MPE     MAPE
## Training set -3.404678e-05 0.0009030436 0.00070459 -0.05141442 1.295943
##                   MASE        ACF1
## Training set 0.9877129 -0.02211437
```

Test of significance of coefficients:

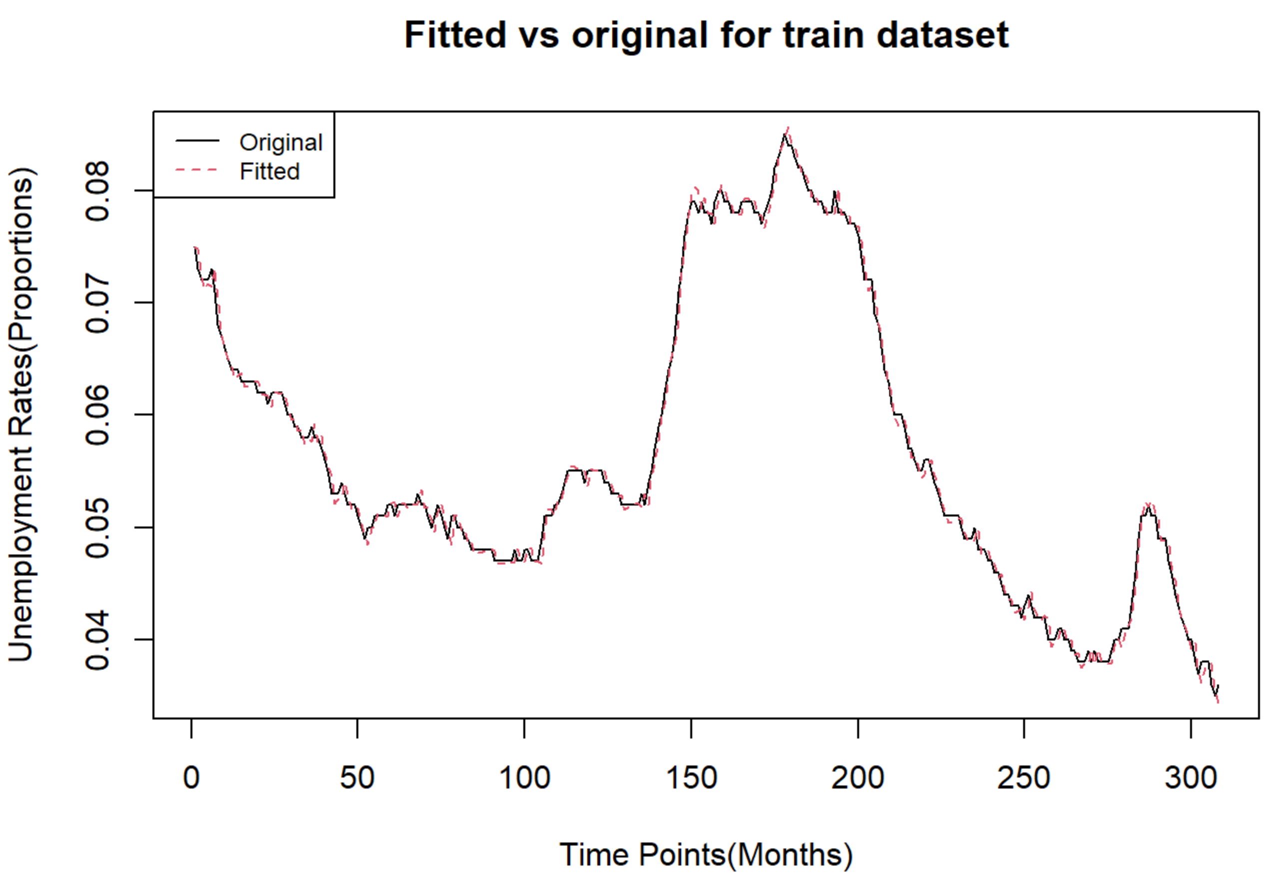
```
suppressWarnings(library(lmtest))
coeftest(est_1)
```

```
##
## z test of coefficients:
##
##                              Estimate  Std. Error z value Pr(>|z|)
## ar1                        0.89392974  0.04144593 21.5686  < 2e-16 ***
## ma1                       -0.65597529  0.06723064 -9.7571  < 2e-16 ***
## Avg_actual_weekly_hours_of_work  0.00037148  0.00020344  1.8260  0.06785 .
## male_to_female_unemp       0.00277229  0.00168480  1.6455  0.09987 .
## ---
## 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_uk_train1$Unemployment_rate-res
ts.plot(df_uk_train1$Unemployment_rate, type="l", xlab="Time Points(Months)", ylab="Unemployment R
ates(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_uk_test1[, c(8,13)]), se.fit=FALSE, met
hod="ML")
```

Predicted values:

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

```
## [1] 0.03587447 0.03590770 0.03592086 0.03579848 0.03583741 0.03582090
```

Original values:

```
print(df_uk_test1$Unemployment_rate)
```

```
## [1] 0.037 0.037 0.037 0.037 0.037 0.038
```

Performance on test dataset:

MAPE (in %):

```
(1/length(df_uk_test1$Unemployment_rate))*(sum(abs(df_uk_test1$Unemployment_rate-as.vector(test_pr
ed))/abs(df_uk_test1$Unemployment_rate)))*100
```

```
## [1] 3.505785
```

RMSE:

```
sqrt(mean((df_uk_test1$Unemployment_rate-as.vector(test_pred))^2))
```

```
## [1] 0.001364321
```

Thus, the fitted model is working well, more or less, for future dataset.

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

Checking stationarity:

```
data_uk[,"Unemployment_rate"] %>%
  ur.kpss() %>%
  summary()
```

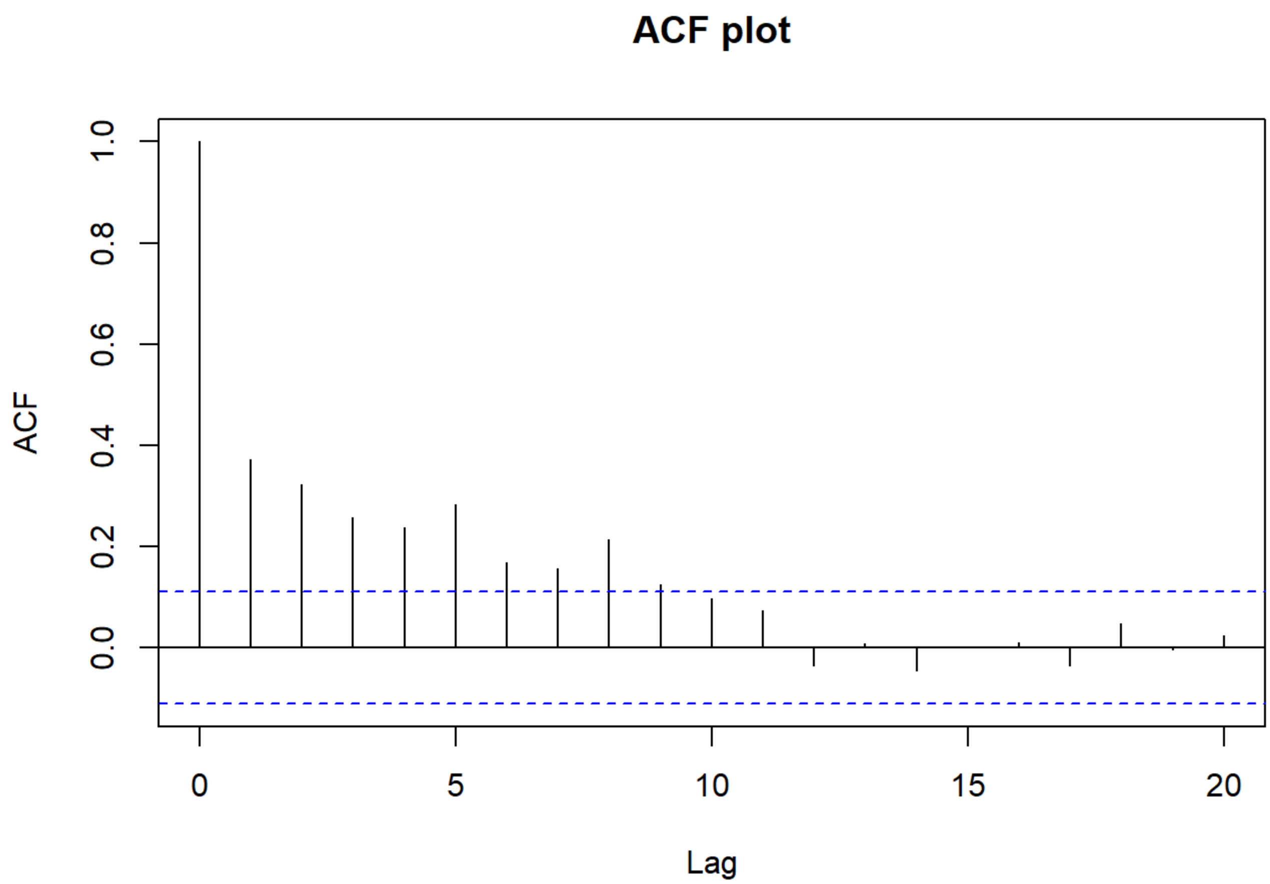
```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 1.0643
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
diff(data_uk[,"Unemployment_rate"]) %>%
  ur.kpss() %>%
  summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.2396
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

ACF plot:

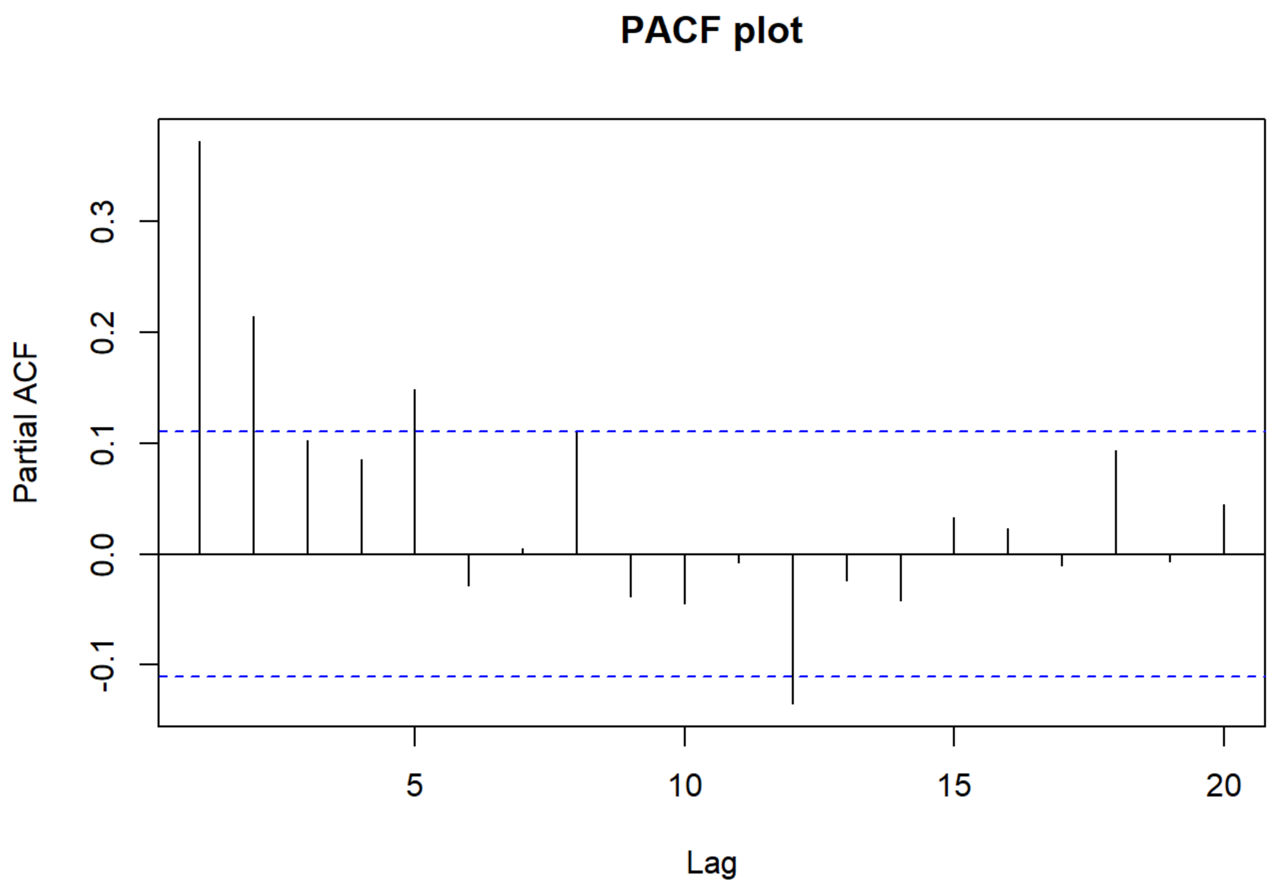
```
par(mfrow=c(1,1))
acf(diff(data_uk$Unemployment_rate), lag.max = 20, main = "ACF plot")
```



p can be taken as 0/1/2/3/4, based on the no. of significant lags.

PACF plot:

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



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

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

```
est_actual=arima(data_uk$Unemployment_rate, order=c(1,1,1), xreg = as.matrix(data_uk[,c(8,13)]), m
ethod = "ML")
summary(est_actual)
```

```
##
## Call:
## arima(x = data_uk$Unemployment_rate, order = c(1, 1, 1), xreg = as.matrix(data_uk[,
##     c(8, 13)]), method = "ML")
##
## Coefficients:
##          ar1      ma1  Avg_actual_weekly_hours_of_work  male_to_female_unemp
##       0.8913  -0.6520                            4e-04                0.0028
## s.e.  0.0417   0.0671                            2e-04                0.0017
##
## sigma^2 estimated as 8.099e-07:  log likelihood = 1750.8,  aic = -3491.6
##
## Training set error measures:
##                         ME        RMSE          MAE        MPE     MAPE
## Training set -2.921492e-05 0.000898517 0.0006998858 -0.0392716 1.294598
##                   MASE        ACF1
## Training set 0.9912409 -0.01646729
```

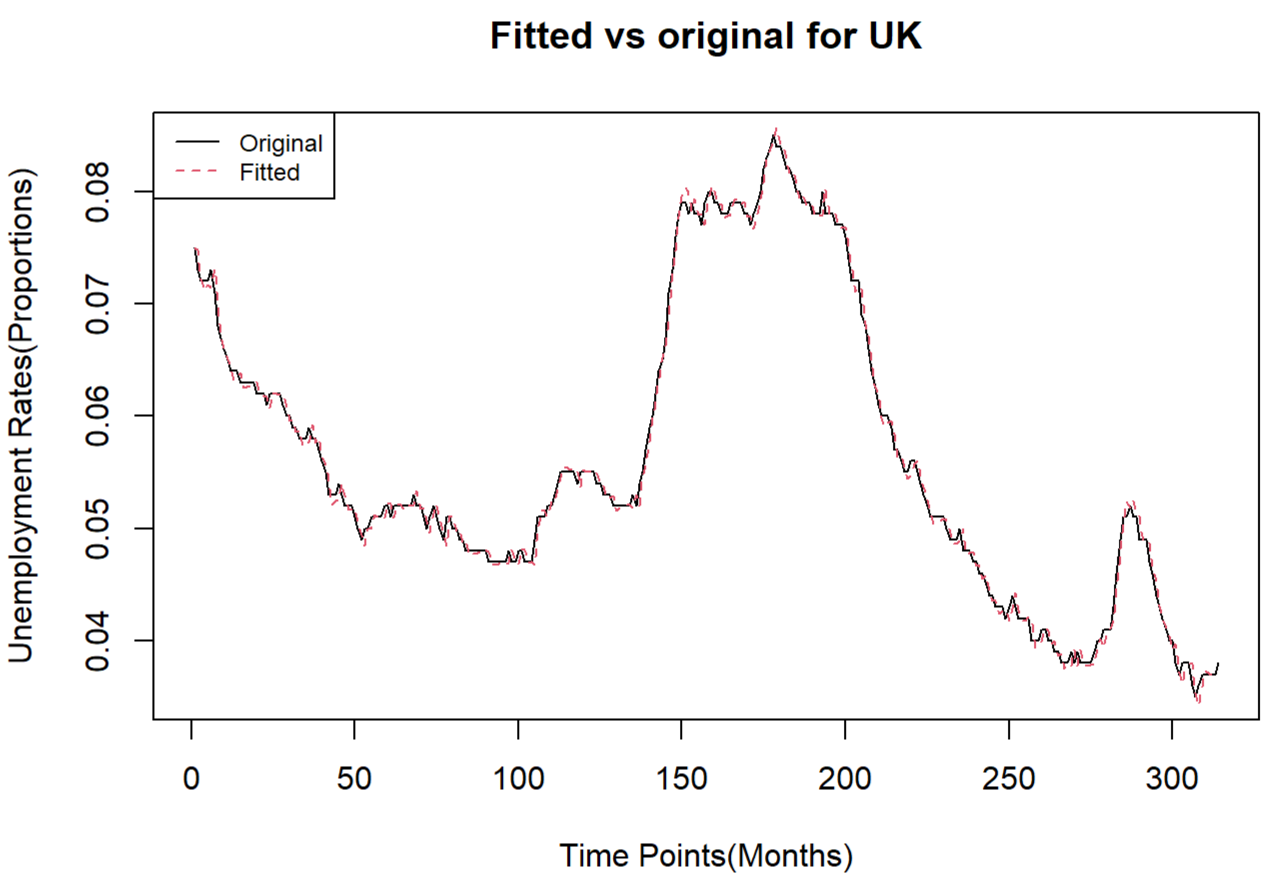
Test of significance of individual coefficients:

```
suppressWarnings(library(lmtest))
coeftest(est_actual)
```

```
##
## z test of coefficients:
##
##                                   Estimate  Std. Error z value Pr(>|z|)
## ar1                             0.89125948  0.04169167 21.3774  < 2e-16 ***
## ma1                            -0.65196174  0.06713170 -9.7117  < 2e-16 ***
## Avg_actual_weekly_hours_of_work  0.00036209  0.00020207  1.7920  0.07314 .
## male_to_female_unemp            0.00281557  0.00165424  1.7020  0.08875 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plot of Fitted vs Original on the actual data:

```
res=residuals(est_actual)
data_fit=data_uk$Unemployment_rate-res
ts.plot(data_uk$Unemployment_rate, type="l", xlab="Time Points(Months)", ylab="Unemployment Rates
(Proportions)", main="Fitted vs original for UK")
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 forecast of Avg_actual_weekly_hours_of_work and Male / Female unemployment for the month of March 2023.

```
auto.arima(data_uk$Avg_actual_weekly_hours_of_work)
```

```
## Series: data_uk$Avg_actual_weekly_hours_of_work
## ARIMA(1,1,3)
##
## Coefficients:
##          ar1      ma1      ma2      ma3
##       0.8286  -0.2063  -0.1188  -0.5902
## s.e.  0.0531   0.0578   0.0501   0.0493
##
## sigma^2 = 0.04241:  log likelihood = 51.73
## AIC=-93.46   AICc=-93.26   BIC=-74.72
```

```
est_aawhw=arima(data_uk$Avg_actual_weekly_hours_of_work, order=c(1,1,3))
future_aawhw=predict(est_aawhw, n.ahead=1, se.fit=FALSE)
print(future_aawhw)
```

```
## Time Series:
## Start = 315
## End = 315
## Frequency = 1
## [1] 36.48415
```

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

```
## Series: data_uk$male_to_female_unemp
## ARIMA(0,1,1)
##
## Coefficients:
##           ma1
##       -0.1889
## s.e.   0.0615
##
## sigma^2 = 0.0008014:  log likelihood = 672.07
## AIC=-1340.14   AICc=-1340.1   BIC=-1332.64
```

```
est_male_to_female_unemp=arima(data_uk$male_to_female_unemp, order=c(0,1,1))
future_male_to_female_unemp=predict(est_male_to_female_unemp, n.ahead=1, se.fit=FALSE)
print(future_male_to_female_unemp)
```

```
## Time Series:
## Start = 315
## End = 315
## Frequency = 1
## [1] 1.1776
```

Obtaining prediction of Unemployment rate for March

2023:

```
march_input=data.frame(as.vector(future_aawhw), as.vector(future_male_to_female_unemp))

future_unemp_pred=predict(est_actual, n.ahead=1, newxreg = as.matrix(march_input[, c(1,2)]), se.fi
t=FALSE, method="ML")
print(as.vector(future_unemp_pred))
```

```
## [1] 0.03818014
```

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

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

Upper limit for March 2023 forecast:

```
print(as.vector(upper))
```

```
## [1] 0.03994403
```

Lower limit for March 2023 forecast:

```
print(as.vector(lower))
```

```
## [1] 0.03641625
```

March 23 forecast - 3.818 %

Upper & Lower limits - (3.642 %, 3.9944 %)