Canada Unemployment predictions

Importing dataset:

```
rm(list=ls())
data_canada=read.csv("Canada_Unemployment.csv")
head(data_canada)
```

```
##   REF_DATE Labour_force Population Unemployment Employment_rate
## 1  1976-01     10369700   15015900       733000           0.630
## 2  1976-02     10389800   15049000       730000           0.631
## 3  1976-03     10395700   15081200       691500           0.632
## 4  1976-04     10451300   15113400       713100           0.633
## 5  1976-05     10446100   15145500       720000           0.632
## 6  1976-06     10469600   15177600       721300           0.631
##   Participation_rate Unemployment_rate Unemployed_Male Unemployed_Female  CPI
## 1              0.615             0.071          419000            314000 30.3
## 2              0.615             0.069          416200            313800 30.5
## 3              0.614             0.064          396000            295500 30.6
## 4              0.616             0.068          402000            311100 30.7
## 5              0.614             0.069          402200            317800 30.9
## 6              0.614             0.071          413500            307800 31.1
```

```
tail(data_canada)
```

```
##     REF_DATE Labour_force Population Unemployment Employment_rate
## 562  2022-10     20872200   24836000      1085800           0.757
## 563  2022-11     20881300   24853700      1068200           0.758
## 564  2022-12     20925800   24877100      1043400           0.759
## 565  2023-01     21078300   24914600      1046000           0.762
## 566  2023-02     21120500   24957600      1066400           0.762
## 567  2023-03     21141800   26422403      1053000           0.763
##     Participation_rate Unemployment_rate Unemployed_Male Unemployed_Female
## 562              0.654             0.053          563100            522800
## 563              0.653             0.052          583500            484700
## 564              0.654             0.051          567700            475700
## 565              0.657             0.052          568700            477300
## 566              0.657             0.051          569900            496500
## 567              0.656             0.050          572900            480000
##       CPI
## 562 153.8
## 563 154.3
## 564 154.3
## 565 154.7
## 566 154.9
## 567 155.1
```

Feature Scaling-creating a new variable:

```
data_canada$male_to_female_unemp=round((data_canada$Unemployed_Male/data_canada$Unemployed_Femal
e),4)
```

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:

```
library(regclass)
```

```
## Warning: package 'regclass' was built under R version 4.2.3
```

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

```
## Warning: package 'bestglm' was built under R version 4.2.3
```

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

```
## Warning: package 'leaps' was built under R version 4.2.3
```

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

```
## Warning: package 'VGAM' was built under R version 4.2.3
```

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

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

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

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

```
## Warning: package 'randomForest' was built under R version 4.2.3
```

```
## 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+Population+Unemployment+Employment_rate+Particip
ation_rate+CPI+male_to_female_unemp, data = data_canada))
```

```
##         Labour_force           Population         Unemployment
##           389.557968           272.137972            20.824760
##      Employment_rate   Participation_rate                  CPI
##           176.391468            19.262308           146.939746
## male_to_female_unemp
##             1.832133
```

Removing Labour_force:

```
VIF(lm(formula = Unemployment_rate ~ Population+Unemployment+Employment_rate+Participation_rate+CP
I+male_to_female_unemp, data = data_canada))
```

```
##           Population         Unemployment      Employment_rate
##            56.027555            18.044405           134.617066
##   Participation_rate                  CPI male_to_female_unemp
##            16.588392           146.757188             1.713528
```

Removing Employment_rate:

```
VIF(lm(formula = Unemployment_rate ~ Population+Unemployment+Participation_rate+CPI+male_to_female
_unemp, data = data_canada))
```

```
##           Population         Unemployment   Participation_rate
##            55.828555             1.677055             1.533431
##                  CPI male_to_female_unemp
##            61.617510             1.346959
```

Removing CPI:

```
VIF(lm(formula = Unemployment_rate ~ Population+Unemployment+Participation_rate+male_to_female_une
mp, data = data_canada))
```

```
##           Population         Unemployment   Participation_rate
##             1.749257             1.420270             1.411392
## male_to_female_unemp
##             1.287246
```

This is the final set of external variables without multicollinearity.

Train-Test split of the dataset - last 6 months of the data

would be taken into testing part:

```
df_canada_train1=data_canada[1:(nrow(data_canada)-6),]
df_canada_test1=data_canada[(nrow(data_canada)-5):nrow(data_canada),]
head(df_canada_train1)
```

```
##   REF_DATE Labour_force Population Unemployment Employment_rate
## 1  1976-01     10369700   15015900       733000           0.630
## 2  1976-02     10389800   15049000       730000           0.631
## 3  1976-03     10395700   15081200       691500           0.632
## 4  1976-04     10451300   15113400       713100           0.633
## 5  1976-05     10446100   15145500       720000           0.632
## 6  1976-06     10469600   15177600       721300           0.631
##   Participation_rate Unemployment_rate Unemployed_Male Unemployed_Female  CPI
## 1              0.615             0.071          419000            314000 30.3
## 2              0.615             0.069          416200            313800 30.5
## 3              0.614             0.064          396000            295500 30.6
## 4              0.616             0.068          402000            311100 30.7
## 5              0.614             0.069          402200            317800 30.9
## 6              0.614             0.071          413500            307800 31.1
##   male_to_female_unemp
## 1               1.3344
## 2               1.3263
## 3               1.3401
## 4               1.2922
## 5               1.2656
## 6               1.3434
```

- 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:

```
library(fpp2)
```

```
## Warning: package 'fpp2' was built under R version 4.2.3
```

```
## 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
```

```
## Warning: package 'forecast' was built under R version 4.2.2
```

```
## Warning: package 'fma' was built under R version 4.2.2
```

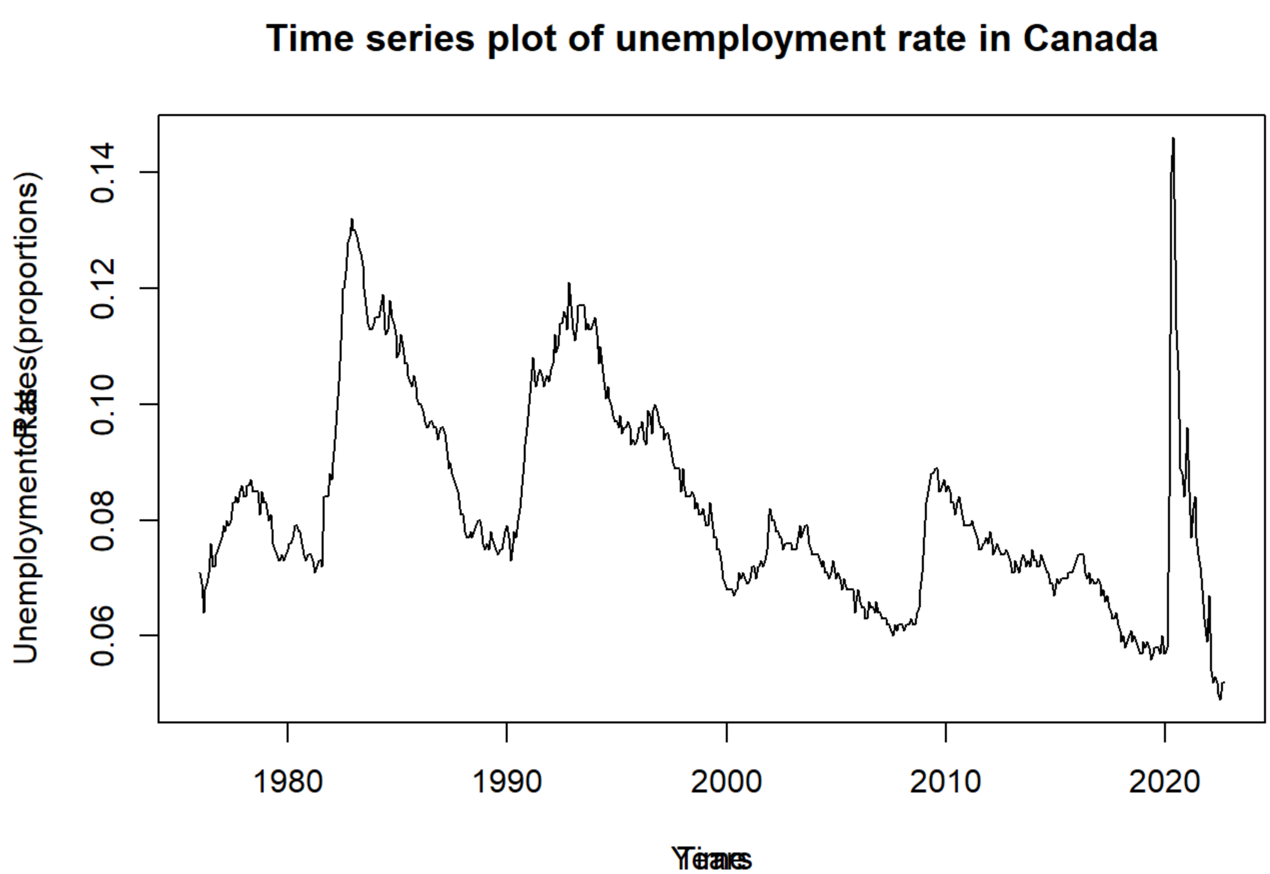
```
## Warning: package 'expsmooth' was built under R version 4.2.2
```

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

```
library(urca)
```

```
## Warning: package 'urca' was built under R version 4.2.3
```

```
df.ts=ts(df_canada_train1$Unemployment_rate, frequency = 12, start = c(1976,1))
plot(df.ts)
title(main="Time series plot of unemployment rate in Canada", xlab="Years", ylab = "Unemployment R
ates(proportions)")
```



Testing stationarity:

```
df_canada_train1[,"Unemployment_rate"] %>%
  ur.kpss() %>%
  summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 2.6899
##
## 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:

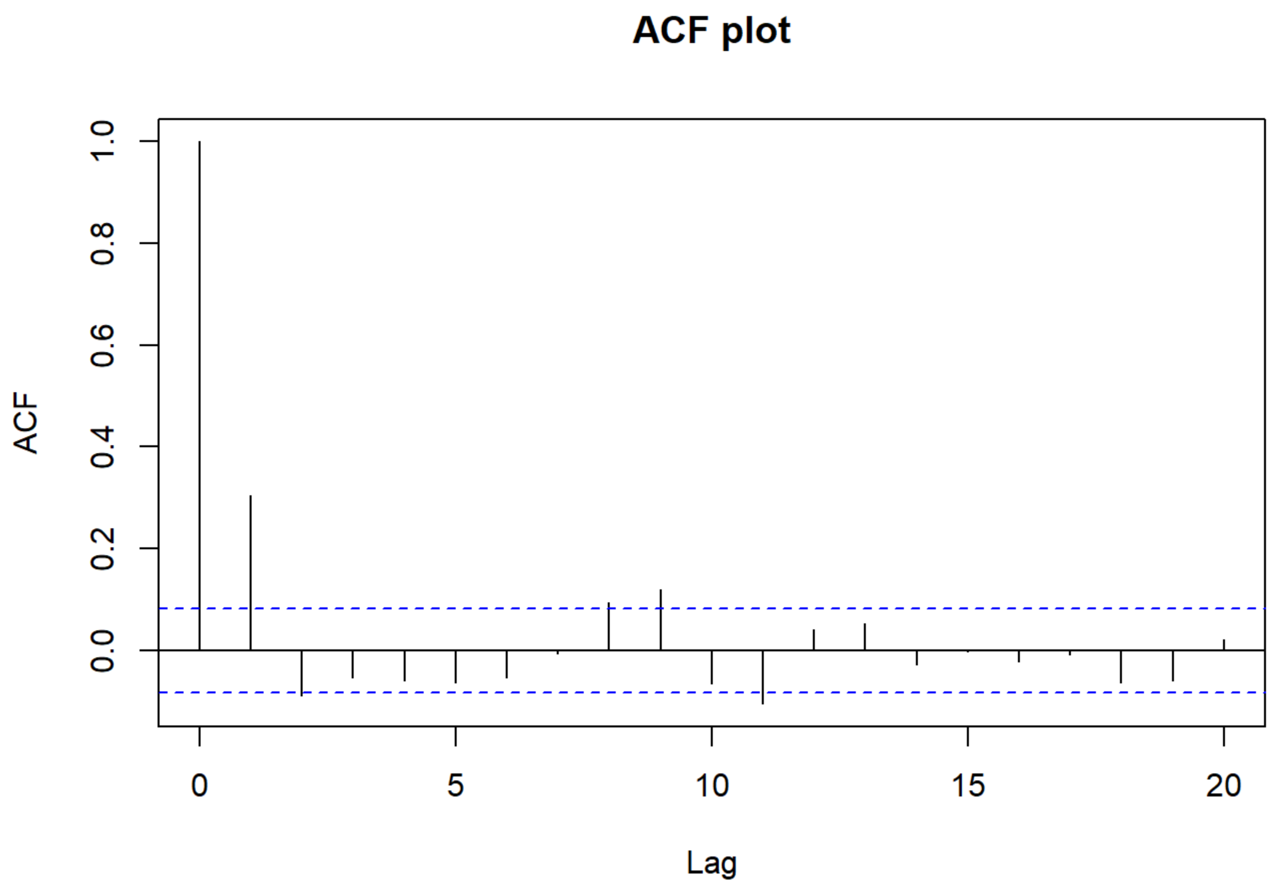
```
diff(df_canada_train1[,"Unemployment_rate"]) %>%
  ur.kpss() %>%
  summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.0717
##
## 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:

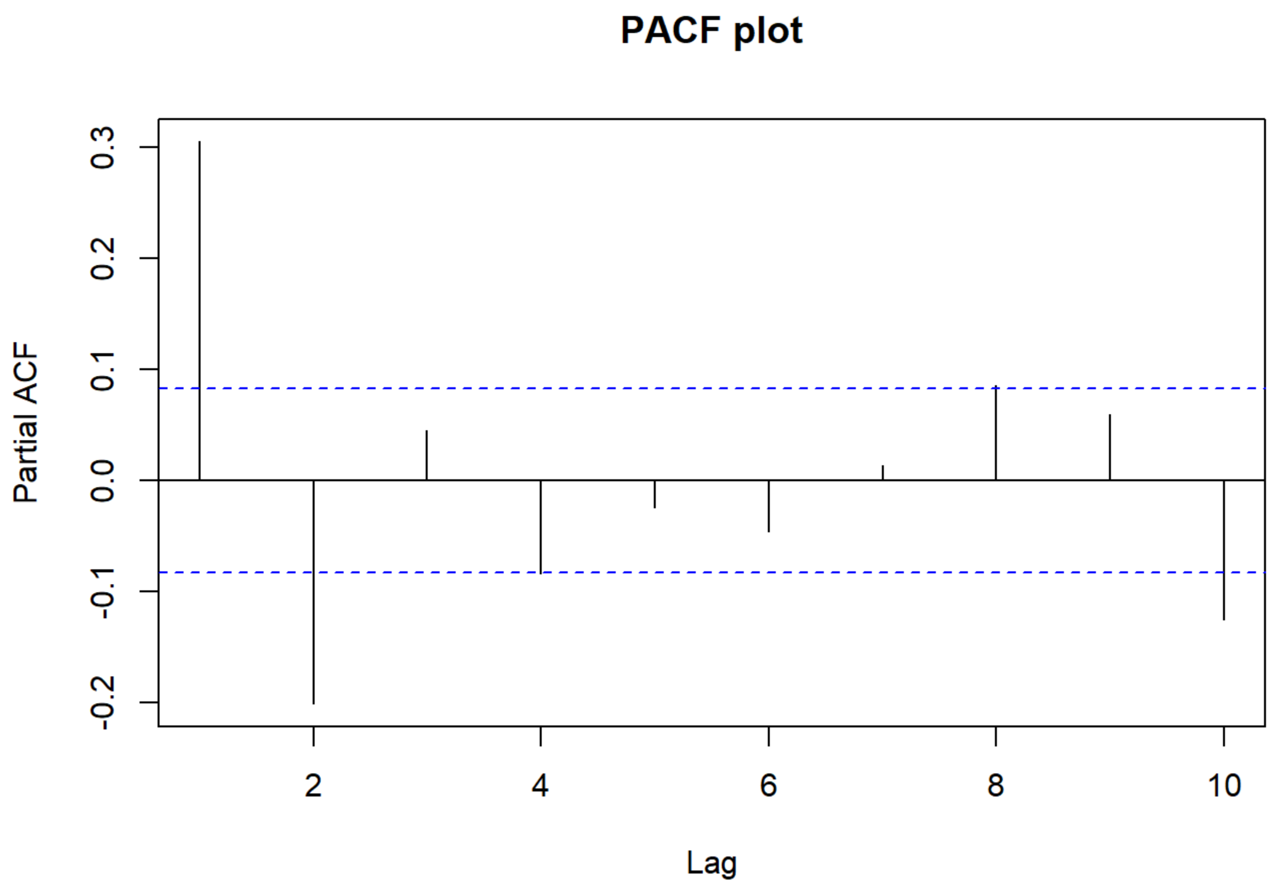
```
par(mfrow=c(1,1))
acf(diff(df_canada_train1$Unemployment_rate), lag.max = 20, main = "ACF plot")
```



p can be taken as 0 or 1 based on the number of significant lags.

PACF plot:

```
par(mfrow=c(1,1))
pacf(diff(df_canada_train1$Unemployment_rate), lag.max = 10, main = "PACF plot")
```



q can be 1 or 2, since we can see the 1st & 2nd lags are significant.

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

Starting with the value of p & q as 1 & 2 respectively with the rest of the regressors:

```
est_train=arima(df_canada_train1$Unemployment_rate, order=c(1,1,2), xreg = as.matrix(df_canada_tra
in1[,c(3,4,6,11)]), method = "ML")
summary(est_train)
```

```
##
## Call:
## arima(x = df_canada_train1$Unemployment_rate, order = c(1, 1, 2), xreg = as.matrix(df_canada_tr
ain1[,
##     c(3, 4, 6, 11)]), method = "ML")
##
## Coefficients:
```

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

```
##           ar1     ma1      ma2  Population  Unemployment  Participation_rate
##       -0.5824  0.3907  -0.1876           0             0             -0.1351
## s.e.   0.1714  0.1631   0.0306         NaN           NaN              0.0028
##       male_to_female_unemp
##                     0.0012
## s.e.                0.0009
##
## sigma^2 estimated as 9.188e-07:  log likelihood = 3097.43,  aic = -6178.87
##
## Training set error measures:
##                          ME         RMSE          MAE         MPE      MAPE
## Training set -2.442301e-05 0.0009576814 0.0006686623 -0.02762593 0.8163421
##                    MASE         ACF1
## Training set 0.3667492 -0.008262735
```

Test of significance of individual coefficients:

```
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                   -5.8240e-01  1.7140e-01  -3.3979 0.0006789 ***
## ma1                    3.9073e-01  1.6314e-01   2.3951 0.0166170 *
## ma2                   -1.8762e-01  3.0647e-02  -6.1218 9.252e-10 ***
## Population            -2.3494e-09         NaN      NaN       NaN
## Unemployment           5.7617e-08         NaN      NaN       NaN
## Participation_rate    -1.3506e-01  2.7841e-03 -48.5108 < 2.2e-16 ***
## male_to_female_unemp  1.2427e-03  8.9801e-04   1.3838 0.1664231
## ---
## 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_canada_train1$Unemployment_rate, order=c(1,1,2), xreg = as.matrix(df_canada_train1
[,c(6)]), method = "ML")
summary(est_1)
```

```
##
## Call:
## arima(x = df_canada_train1$Unemployment_rate, order = c(1, 1, 2), xreg = as.matrix(df_canada_tr
ain1[,
##     c(6)]), method = "ML")
##
## Coefficients:
##           ar1     ma1     ma2  as.matrix(df_canada_train1[, c(6)])
##       -0.9165  1.0689  0.1101                              -0.6544
## s.e.   0.0532  0.0727  0.0520                               0.0571
##
## sigma^2 estimated as 9.586e-06:  log likelihood = 2440.79,  aic = -4871.59
##
## Training set error measures:
##                        ME        RMSE         MAE         MPE     MAPE     MASE
## Training set 8.884524e-06 0.003093443 0.002084236 -0.07119637 2.501766 1.143166
##                     ACF1
## Training set 0.001095158
```

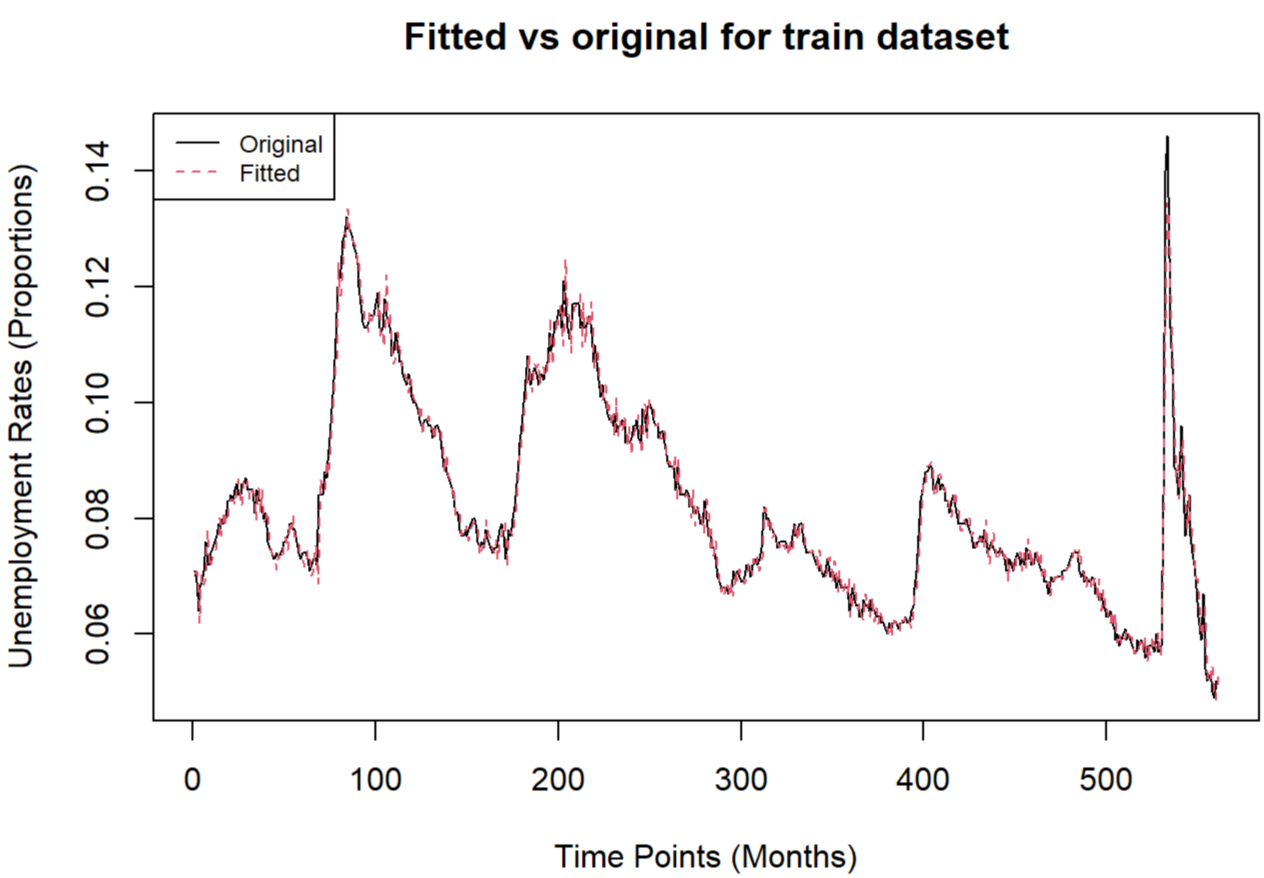
Test of significance of individual coefficients:

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

```
##
## z test of coefficients:
##
##                                      Estimate Std. Error  z value Pr(>|z|)
## ar1                                 -0.916528   0.053246 -17.2131  < 2e-16 ***
## ma1                                  1.068920   0.072691  14.7050  < 2e-16 ***
## ma2                                  0.110108   0.052046   2.1156  0.03438 *
## as.matrix(df_canada_train1[, c(6)]) -0.654419   0.057071 -11.4667  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plot of Fitted vs Original values for train dataset:

```
res=residuals(est_1)
data_fit=df_canada_train1$Unemployment_rate-res
ts.plot(df_canada_train1$Unemployment_rate, type="l", main="Fitted vs original for train dataset",
xlab="Time Points (Months)", ylab="Unemployment Rates (Proportions)")
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_canada_test1[, c(6)]), se.fit=FALSE, me
thod="ML")
```

Predicted values:

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

```
## [1] 0.05066340 0.05117934 0.05065184 0.04857226 0.04867888 0.04923558
```

Original values:

```
print(df_canada_test1$Unemployment_rate)
```

```
## [1] 0.053 0.052 0.051 0.052 0.051 0.050
```

Performane on test dataset:

MAPE (in %):

```
(1/length(df_canada_test1$Unemployment_rate))*(sum(abs(df_canada_test1$Unemployment_rate-as.vector
(test_pred))/abs(df_canada_test1$Unemployment_rate)))*100
```

```
## [1] 3.223566
```

RMSE:

```
sqrt(mean((df_canada_test1$Unemployment_rate-test_pred)^2))
```

```
## [1] 0.001998989
```

Seems to be working, more or less well, for future datasets.

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

Checking stationarity:

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

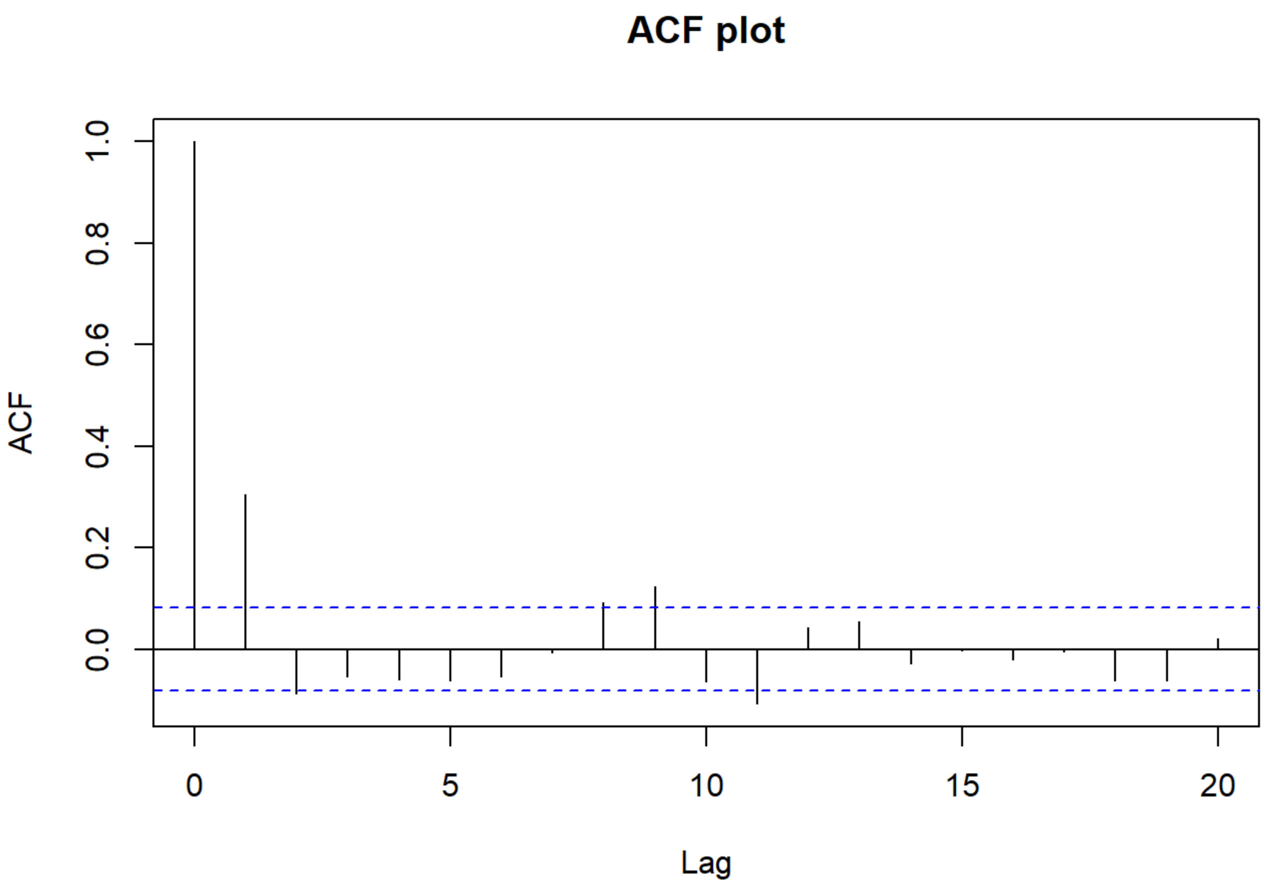
```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 2.8207
##
## Critical value for a significance level of:
##                  10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

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

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.0746
##
## Critical value for a significance level of:
##                  10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

ACF plot:

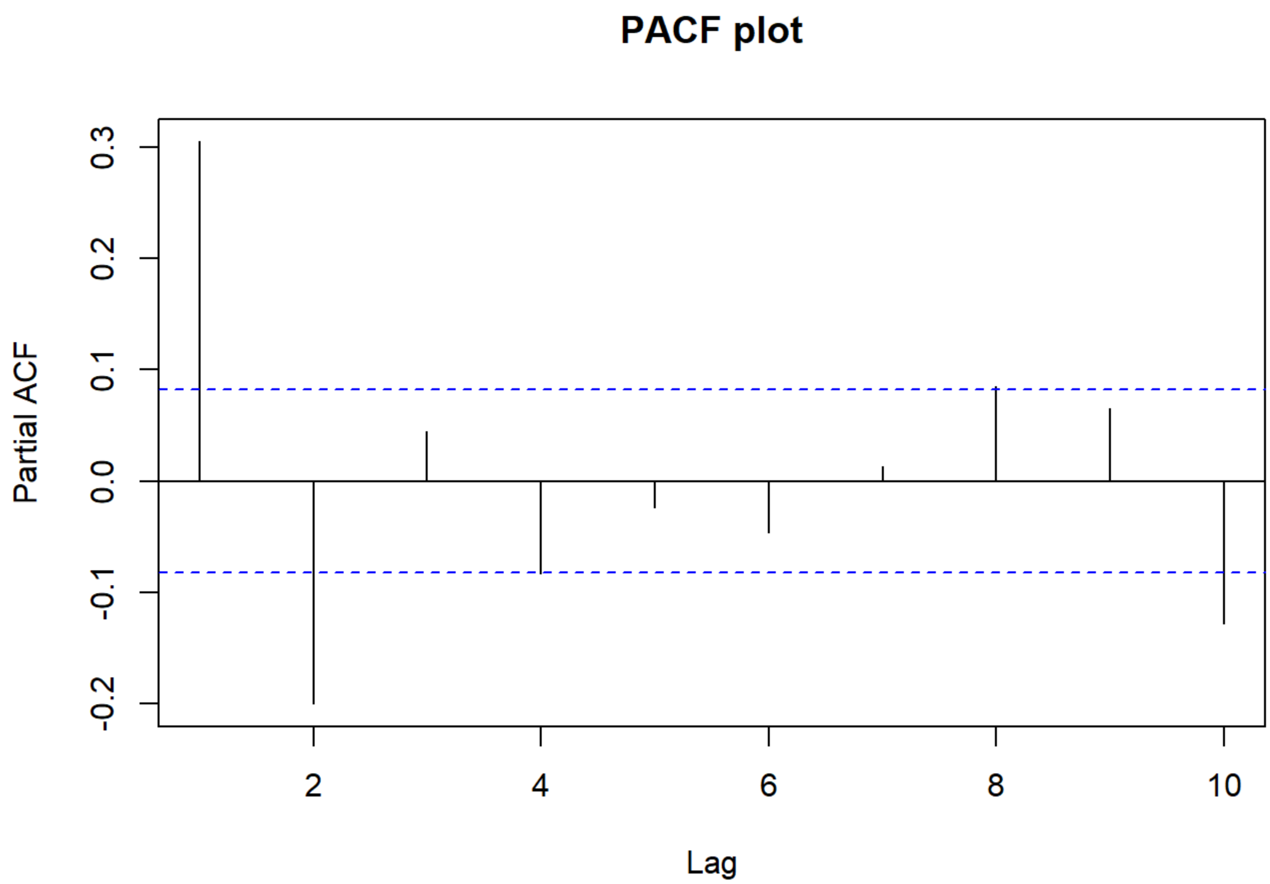
```
par(mfrow=c(1,1))
acf(diff(data_canada$Unemployment_rate), lag.max = 20, main = "ACF plot")
```



p can be taken as 1 since only the 1st lag is looking to be significant.

PACF plot:

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



q can be taken as 2 since the 2nd lag is looking to be significant.

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

```
est_2=arima(data_canada$Unemployment_rate, order=c(1,1,2), xreg = as.matrix(data_canada[,c(6)]), m
ethod = "ML")
```

```
## Warning in arima(data_canada$Unemployment_rate, order = c(1, 1, 2), xreg =
## as.matrix(data_canada[, : possible convergence problem: optim gave code = 1
```

```
summary(est_2)
```

```
##
## Call:
## arima(x = data_canada$Unemployment_rate, order = c(1, 1, 2), xreg = as.matrix(data_canada[,
##     c(6)]), method = "ML")
##
## Coefficients:
##           ar1     ma1     ma2  as.matrix(data_canada[, c(6)])
##       -0.9139  1.0661  0.1085                         -0.6538
## s.e.   0.0531  0.0723  0.0517                          0.0568
##
## sigma^2 estimated as 9.523e-06:  log likelihood = 2468.82,  aic = -4927.64
##
## Training set error measures:
##                        ME        RMSE        MAE         MPE     MAPE     MASE
## Training set 9.690324e-06 0.003083255 0.00207956 -0.06916943 2.509043 1.146087
##                      ACF1
## Training set -0.000393414
```

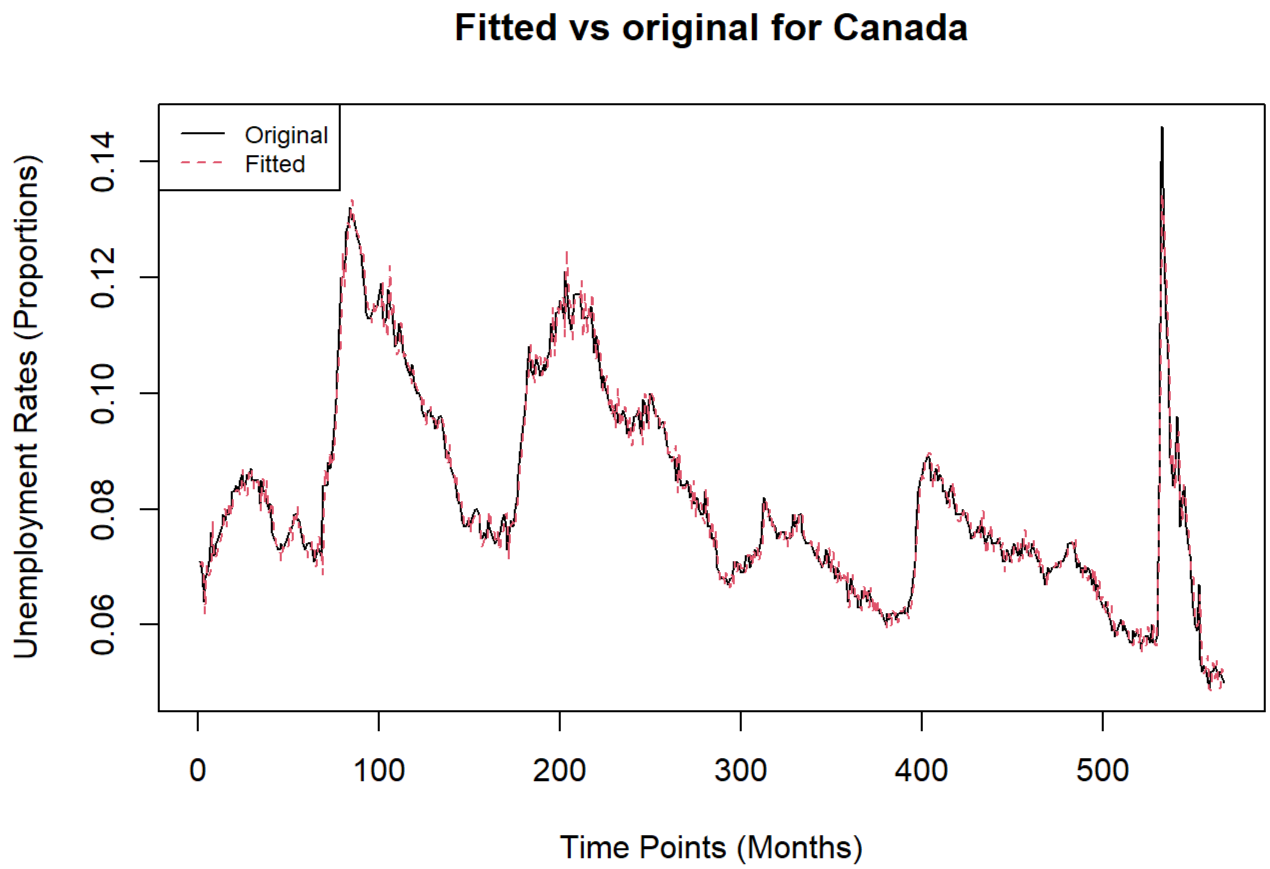
Test of significance of individual coefficients:

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

```
##
## z test of coefficients:
##
##                                 Estimate Std. Error  z value Pr(>|z|)
## ar1                            -0.913855   0.053060 -17.2231  < 2e-16 ***
## ma1                             1.066073   0.072316  14.7419  < 2e-16 ***
## ma2                             0.108476   0.051721   2.0973  0.03596 *
## as.matrix(data_canada[, c(6)]) -0.653783   0.056812 -11.5078  < 2e-16 ***
## ---
## 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_canada$Unemployment_rate-res
ts.plot(data_canada$Unemployment_rate, type="l", main="Fitted vs original for Canada", xlab="Time
Points (Months)", ylab="Unemployment Rates (Proportions)")
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)
```



We need the values of Participation Rate of April, May 23 since it is the significant external variable used in the model. It would be forecasted using ARIMA.

```r
est_prate=arima(data_canada$Participation_rate, order=c(1,1,2), method = "ML")
summary(est_prate)
```

```
##
## Call:
## arima(x = data_canada$Participation_rate, order = c(1, 1, 2), method = "ML")
##
## Coefficients:
##          ar1      ma1      ma2
##       0.4091  -0.2961  -0.3379
## s.e.  0.0939   0.0884   0.0379
##
## sigma^2 estimated as 6.27e-06:  log likelihood = 2587.03,  aic = -5166.06
##
## Training set error measures:
##                        ME        RMSE         MAE        MPE      MAPE     MASE
## Training set 0.0001167148 0.002501856 0.001463115 0.01732193 0.2242535 1.117575
##                     ACF1
## Training set 0.002322251
```

```r
library(lmtest)
coeftest(est_prate)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1  0.409059   0.093927  4.3551  1.33e-05 ***
## ma1 -0.296096   0.088402 -3.3494 0.0008098 ***
## ma2 -0.337875   0.037905 -8.9138 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
future_prate=predict(est_prate, n.ahead=2, se.fit=FALSE, method="ML")
print(future_prate)
```

```
## Time Series:
## Start = 568
## End = 569
## Frequency = 1
## [1] 0.6556483 0.6555701
```

Using these values as input, the May,23 forecast of Unemployment rate is obtained as:

```
future_unemp_pred=predict(est_2, n.ahead=2, newxreg = as.matrix(c(0.6556483,0.6555701)), se.fit=FA
LSE, method="ML")
upper=as.vector(future_unemp_pred)+(1.96*(sqrt(est_2$sigma2)))
lower=as.vector(future_unemp_pred)-(1.96*(sqrt(est_2$sigma2)))
print(as.vector(future_unemp_pred)[2])
```

```
## [1] 0.05012941
```

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] 0.0561778
```

Lower limit for May 2023 forecast:

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

```
## [1] 0.04408102
```

May 23 forecast - 5.013 %

Upper & Lower limits - (4.408 %, 5.618 %)