Japan Unemployment predictions

Importing dataset:

```
rm(list=ls())
data_japan=read.csv("Japan_Unemployment.csv")
head(data_japan)
```

```
##      TIME Unemployment_Rate Employment_Rate_15_64 Working_Age_Population_15_64
## 1 1975-11             0.021              0.659374                     75654440
## 2 1975-12             0.021              0.659842                     75747382
## 3 1976-01             0.021              0.661498                     75797614
## 4 1976-02             0.020              0.662158                     75767387
## 5 1976-03             0.020              0.662846                     75839650
## 6 1976-04             0.021              0.659644                     75934952
##   Labour_Force_15_64 Economic_Inactivity_Rates_15_64 Unemployed_Males
## 1           51010000                        0.326014           740000
## 2           50550000                        0.323541           800000
## 3           49820000                        0.324121           800000
## 4           49910000                        0.324116           720000
## 5           50830000                        0.323705           730000
## 6           51390000                        0.326397           770000
##   Unemployed_Females Unemployed_all  CPI
## 1             370000        1100000 54.7
## 2             340000        1140000 54.7
## 3             330000        1130000 55.7
## 4             370000        1090000 56.1
## 5             320000        1050000 56.3
## 6             340000        1110000 57.6
```

```
tail(data_japan)
```

```
##        TIME Unemployment_Rate Employment_Rate_15_64
## 564 2022-10             0.026              0.786715
## 565 2022-11             0.025              0.784647
## 566 2022-12             0.025              0.785700
## 567 2023-01             0.024              0.787192
## 568 2023-02             0.026              0.783979
## 569 2023-03             0.028              0.782000
##     Working_Age_Population_15_64 Labour_Force_15_64
## 564                     74003919           59970000
## 565                     73829426           59500000
## 566                     73844960           59550000
## 567                     73832099           59360000
## 568                     73586132           59110000
## 569                     73930000           59580000
##     Economic_Inactivity_Rates_15_64 Unemployed_Males Unemployed_Females
## 564                        0.191664          1070000             710000
## 565                        0.193953          1040000             690000
## 566                        0.193039          1030000             680000
## 567                        0.192086          1000000             670000
## 568                        0.193870          1080000             710000
## 569                        0.194000          1150000             800000
##     Unemployed_all   CPI
## 564        1780000 103.7
## 565        1730000 103.9
## 566        1710000 104.1
## 567        1670000 104.7
## 568        1800000 104.0
## 569        1950000 104.4
```

Data of Labour force from 2011-03 to 2011-08 were missing due to earthquake. For these months approximate value has been put there based on the participation rate during that year.

Feature Scaling - Creating a new variable:

```
data_japan$male_to_female_unemp=round((data_japan$Unemployed_Males/data_japan$Unemployed_Females),
4)
print(data_japan$male_to_female_unemp[1:10])
```

```
##  [1] 2.0000 2.3529 2.4242 1.9459 2.2812 2.2647 2.2647 2.1471 2.2727 2.1471
```

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 ~ Employment_Rate_15_64+Working_Age_Population_15_64+Labour_For
ce_15_64+Economic_Inactivity_Rates_15_64+Unemployed_all+CPI+male_to_female_unemp, data = data_japa
n))
```

```
##           Employment_Rate_15_64    Working_Age_Population_15_64
##                     13066.90798                        53.90800
##              Labour_Force_15_64 Economic_Inactivity_Rates_15_64
##                        38.32807                     15137.06140
##                  Unemployed_all                             CPI
##                       907.96954                        35.93361
##            male_to_female_unemp
##                         3.84045
```

Removing Economic_Inactivity_Rates_15_64:

```
VIF(lm(formula = Unemployment_Rate ~ Employment_Rate_15_64+Working_Age_Population_15_64+Labour_For
ce_15_64+Unemployed_all+CPI+male_to_female_unemp, data = data_japan))
```

```
##        Employment_Rate_15_64 Working_Age_Population_15_64
##                    43.252975                    41.809405
##           Labour_Force_15_64               Unemployed_all
##                    38.071781                     6.499345
##                          CPI         male_to_female_unemp
##                    34.340895                     3.755630
```

Removing Employment_Rate_15_64:

```
VIF(lm(formula = Unemployment_Rate ~ Working_Age_Population_15_64+Labour_Force_15_64+Unemployed_al
l+CPI+male_to_female_unemp, data = data_japan))
```

```
## Working_Age_Population_15_64           Labour_Force_15_64
##                     6.612017                    21.955752
##               Unemployed_all                          CPI
##                     2.995417                    20.483774
##         male_to_female_unemp
##                     3.571736
```

Removing Labour_Force_15_64:

```
VIF(lm(formula = Unemployment_Rate ~ Working_Age_Population_15_64+Unemployed_all+CPI+male_to_femal
e_unemp, data = data_japan))
```

```
## Working_Age_Population_15_64               Unemployed_all
##                     1.633869                     2.975656
##                          CPI         male_to_female_unemp
##                     4.360960                     3.223064
```

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_japan_train1=data_japan[1:(nrow(data_japan)-6),]
df_japan_test1=data_japan[(nrow(data_japan)-5):nrow(data_japan),]
head(df_japan_train1)
```

```
##      TIME Unemployment_Rate Employment_Rate_15_64 Working_Age_Population_15_64
## 1 1975-11             0.021              0.659374                     75654440
## 2 1975-12             0.021              0.659842                     75747382
## 3 1976-01             0.021              0.661498                     75797614
## 4 1976-02             0.020              0.662158                     75767387
## 5 1976-03             0.020              0.662846                     75839650
## 6 1976-04             0.021              0.659644                     75934952
##   Labour_Force_15_64 Economic_Inactivity_Rates_15_64 Unemployed_Males
## 1           51010000                        0.326014           740000
## 2           50550000                        0.323541           800000
## 3           49820000                        0.324121           800000
## 4           49910000                        0.324116           720000
## 5           50830000                        0.323705           730000
## 6           51390000                        0.326397           770000
##   Unemployed_Females Unemployed_all  CPI male_to_female_unemp
## 1             370000        1100000 54.7               2.0000
## 2             340000        1140000 54.7               2.3529
## 3             330000        1130000 55.7               2.4242
## 4             370000        1090000 56.1               1.9459
## 5             320000        1050000 56.3               2.2812
## 6             340000        1110000 57.6               2.2647
```

- 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))
```

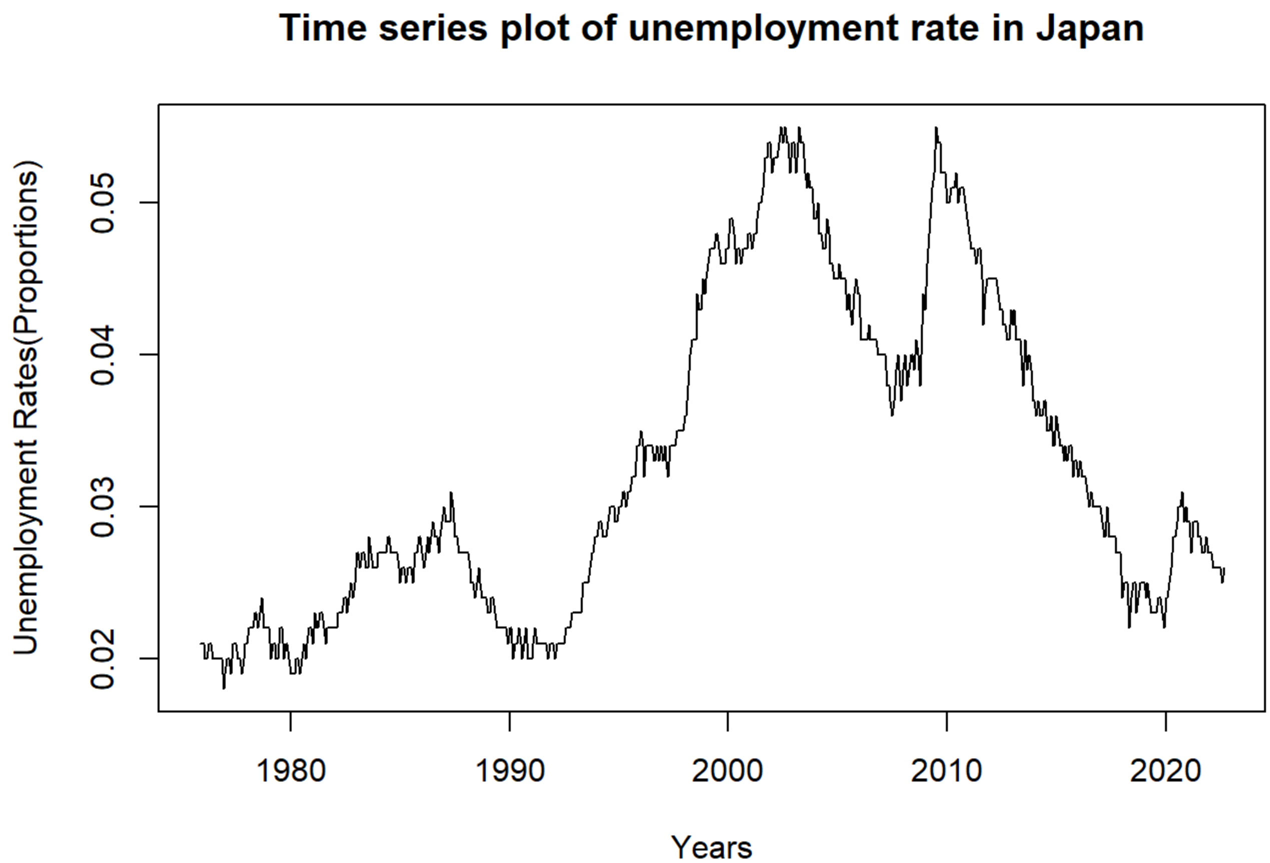
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## ── 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_japan_train1$Unemployment_Rate, frequency = 12, start = c(1975,11))
plot(df.ts,xlab="Years",ylab="Unemployment Rates(Proportions)")
title(main="Time series plot of unemployment rate in Japan")
```



Testing stationarity:

```
df_japan_train1[,"Unemployment_Rate"] %>%
  ur.kpss() %>%
  summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 3.152
##
## 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:

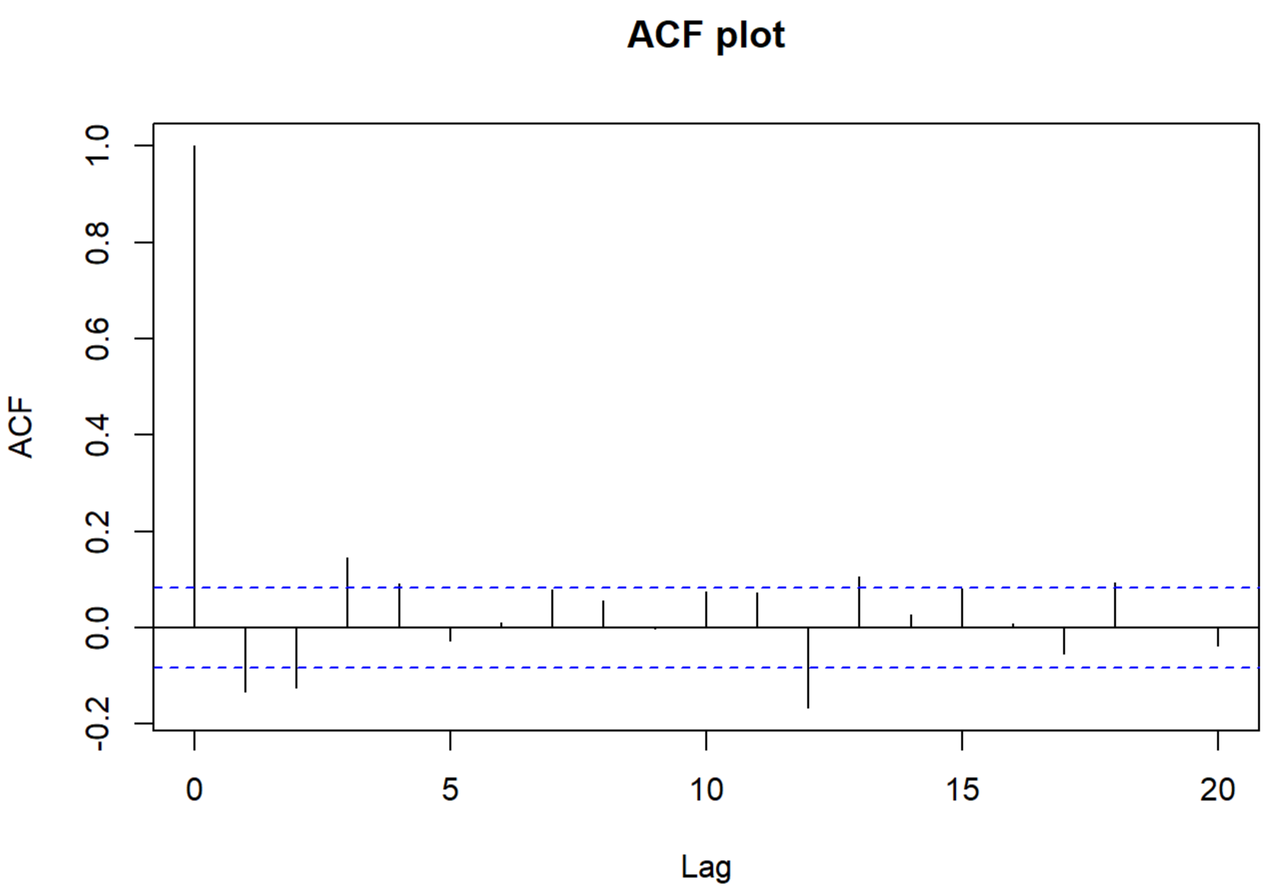
```
diff(df_japan_train1[,"Unemployment_Rate"]) %>%
  ur.kpss() %>%
  summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.363
##
## 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 can be said to be stationary at 5% level of significance.

ACF plot:

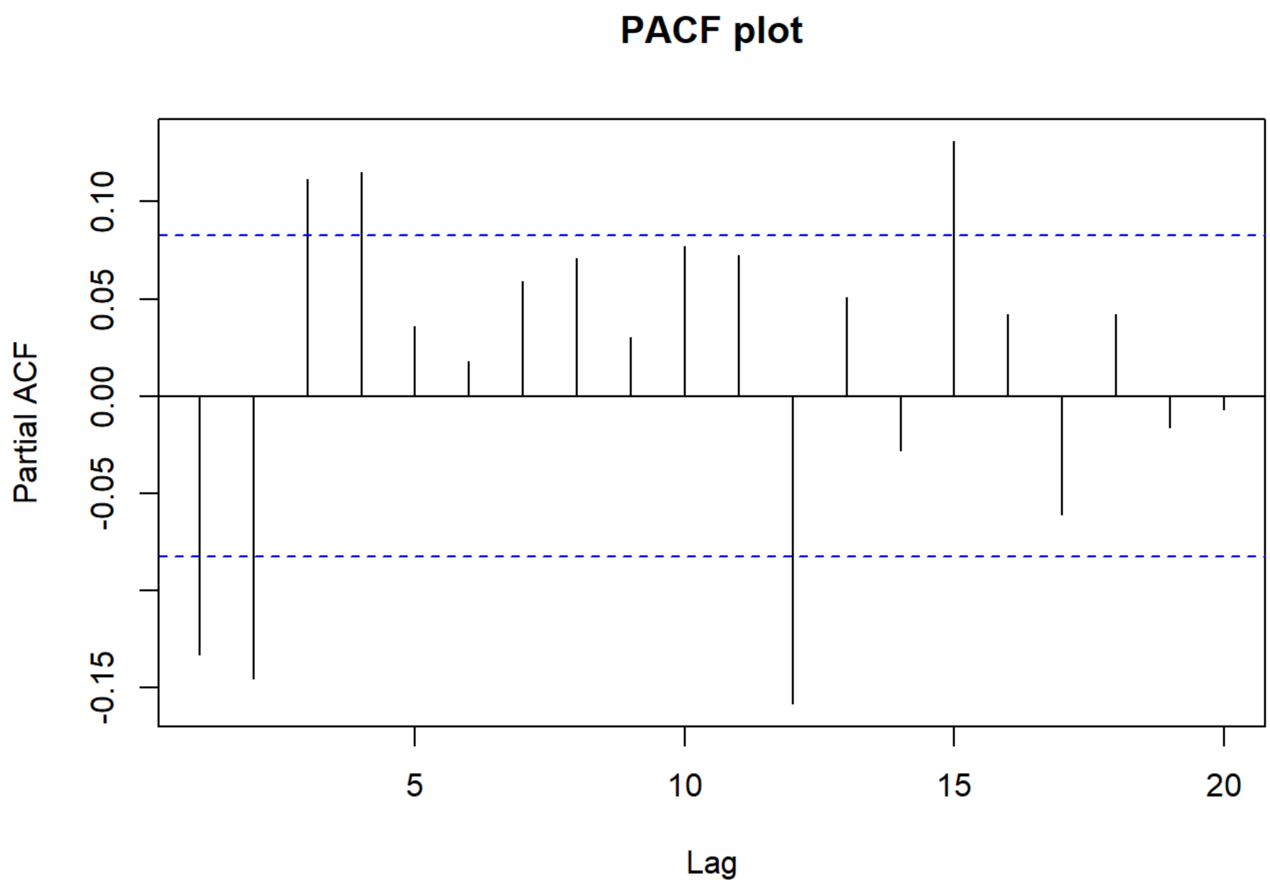
```
par(mfrow=c(1,1))
acf(diff(df_japan_train1$Unemployment_Rate), lag.max = 20, main = "ACF plot")
```



p=0/1/2/3, based on the no. of significant lags.

PACF plot:

```
par(mfrow=c(1,1))
pacf(diff(df_japan_train1$Unemployment_Rate), lag.max = 20, main = "PACF plot")
```



q=0/1/2/3/4, 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 3 & 4 respectively with the rest of the regressors:

```
est_train=arima(df_japan_train1$Unemployment_Rate, order=c(3,1,4), xreg = as.matrix(df_japan_train
1[,c(4,9,10,11)]), method = "ML")
summary(est_train)
```

```
##
## Call:
## arima(x = df_japan_train1$Unemployment_Rate, order = c(3, 1, 4), xreg = as.matrix(df_japan_trai
n1[,
##     c(4, 9, 10, 11)]), method = "ML")
##
## Coefficients:
```

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

```
##           ar1      ar2     ar3      ma1     ma2      ma3     ma4
##       -0.3987  -0.1169  0.1735  -0.3173  -0.249  -0.2492  0.1409
## s.e.      NaN      NaN     NaN      NaN     NaN   0.0487     NaN
##       Working_Age_Population_15_64  Unemployed_all  CPI  male_to_female_unemp
##                                  0               0    0                 1e-04
## s.e.                           NaN             NaN    0                 2e-04
##
## sigma^2 estimated as 1.041e-07:  log likelihood = 3719.99,  aic = -7415.98
##
## Training set error measures:
##                         ME         RMSE          MAE       MPE      MAPE
## Training set -3.189237e-05 0.0003223791 0.0002686316 -0.129861 0.9110315
##                   MASE        ACF1
## Training set 0.3620406 -0.01475248
```

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                         -3.9867e-01         NaN     NaN       NaN
## ar2                         -1.1686e-01         NaN     NaN       NaN
## ar3                          1.7354e-01         NaN     NaN       NaN
## ma1                         -3.1732e-01         NaN     NaN       NaN
## ma2                         -2.4897e-01         NaN     NaN       NaN
## ma3                         -2.4916e-01  4.8692e-02 -5.1170 3.104e-07 ***
## ma4                          1.4091e-01         NaN     NaN       NaN
## Working_Age_Population_15_64 -1.6996e-10         NaN     NaN       NaN
## Unemployed_all               1.5128e-08         NaN     NaN       NaN
## CPI                         -2.7653e-05  2.5209e-05 -1.0970    0.2727
## male_to_female_unemp         6.1606e-05  1.6440e-04  0.3747    0.7079
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

After removing the variables producing NaNs & the insignificant variables one by one, the final model results are:

```
est_2=arima(df_japan_train1$Unemployment_Rate, order=c(2,1,1), method = "ML")
summary(est_2)
```

```
##
## Call:
## arima(x = df_japan_train1$Unemployment_Rate, order = c(2, 1, 1), method = "ML")
##
## Coefficients:
##           ar1      ar2     ma1
##       -0.4377  -0.1918  0.2921
## s.e.   0.1685   0.0436  0.1687
##
## sigma^2 estimated as 1.05e-06:  log likelihood = 3070.86,  aic = -6133.71
##
## Training set error measures:
##                        ME        RMSE          MAE         MPE     MAPE
## Training set 1.097945e-05 0.001023989 0.0007720725 -0.01822717 2.471948
##                  MASE        ACF1
## Training set 1.040539 0.005877635
```

Test of significance of coefficients:

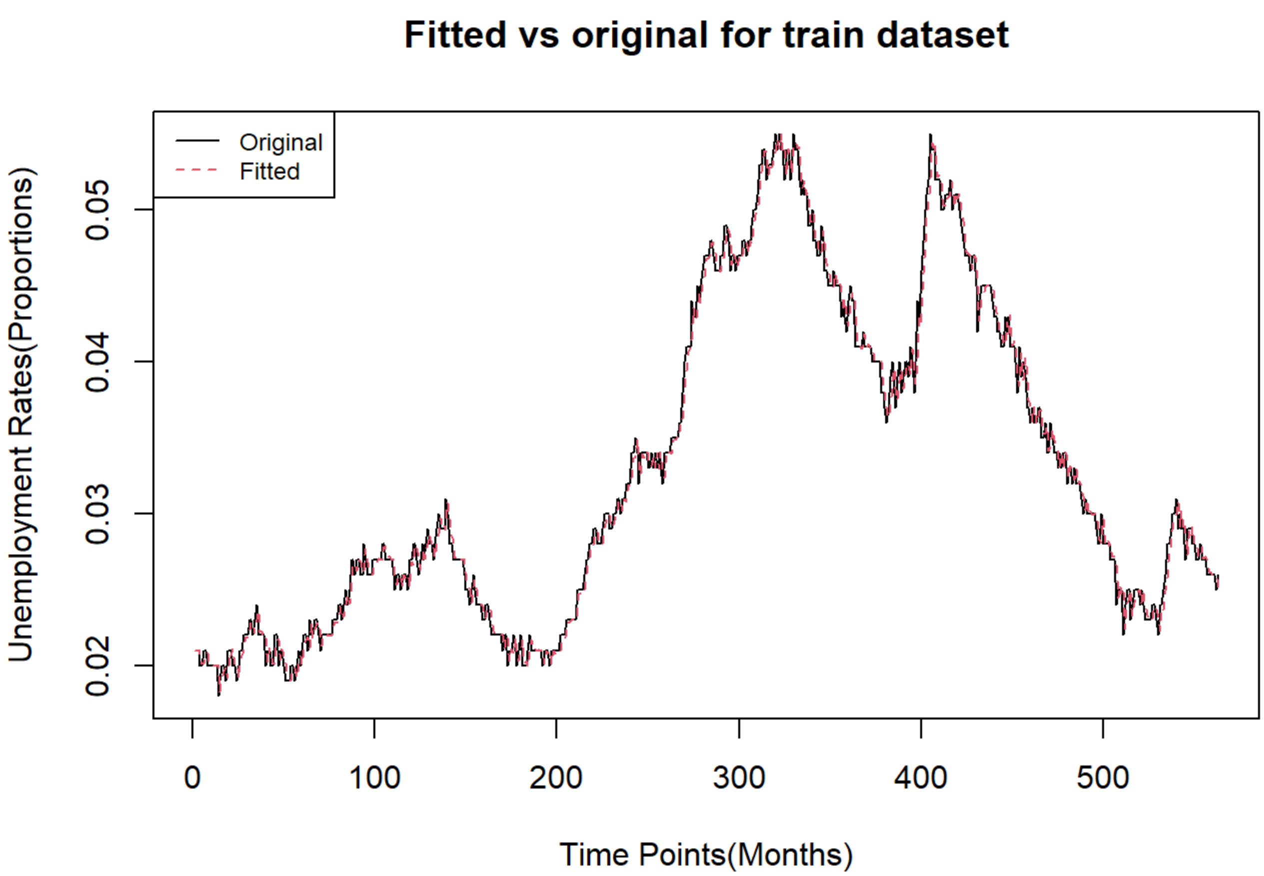
```
suppressWarnings(library(lmtest))
coeftest(est_2)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.437677   0.168526 -2.5971  0.009402 **
## ar2 -0.191840   0.043626 -4.3973 1.096e-05 ***
## ma1  0.292116   0.168749  1.7311  0.083440 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the final fitted model, only the significant parameters are kept.

Plot of Fitted vs Original values for train dataset:

```
res2=residuals(est_2)
data_fit2=df_japan_train1$Unemployment_Rate-res2
ts.plot(df_japan_train1$Unemployment_Rate, type="l", xlab="Time Points(Months)", ylab="Unemploymen
t Rates(Proportions)", main="Fitted vs original for train dataset")
points(data_fit2, 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 the above fitted model:

```
test_pred2=predict(est_2, n.ahead=6, se.fit=FALSE, method="ML")
```

Predicted values:

```
print(as.vector(test_pred2))
```

```
## [1] 0.02600345 0.02581010 0.02589406 0.02589441 0.02587815 0.02588520
```

Original values:

```
print(df_japan_test1$Unemployment_Rate)
```

```
## [1] 0.026 0.025 0.025 0.024 0.026 0.028
```

Performane on test dataset:

MAPE (in %):

```
(1/length(df_japan_test1$Unemployment_Rate))*(sum(abs(df_japan_test1$Unemployment_Rate-as.vector(t
est_pred2))/abs(df_japan_test1$Unemployment_Rate)))*100
```

```
## [1] 3.790801
```

RMSE:

```
sqrt(mean((df_japan_test1$Unemployment_Rate-as.vector(test_pred2))^2))
```

```
## [1] 0.001260399
```

Thus we can say that the final model we had chosen is working well, more or less, for future datasets.

Now using this model, we would fit the actual data of Japan from Nov, 1975 to March, 2023 & obtain forecast for April.

Checking stationarity:

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

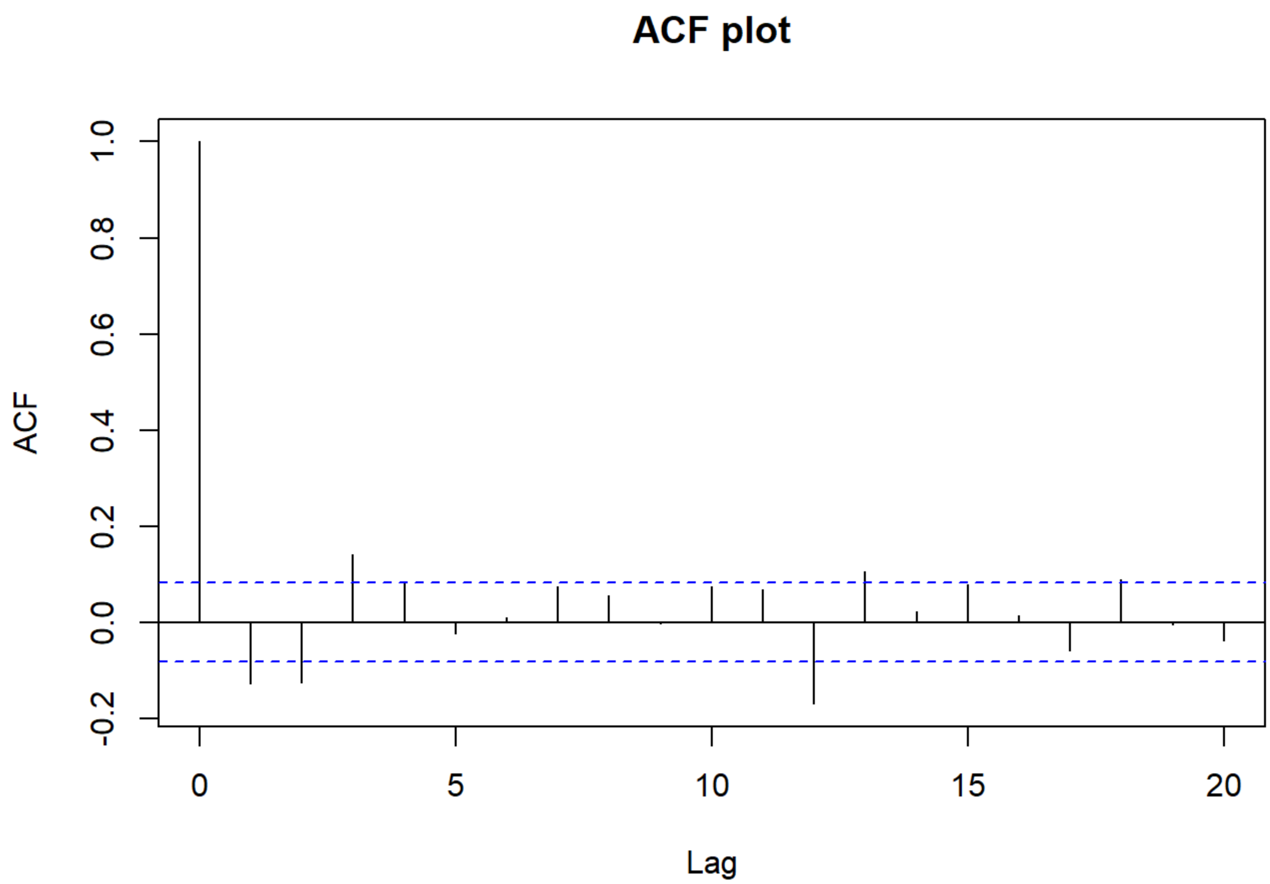
```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 3.0511
##
## Critical value for a significance level of:
##                  10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

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

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.3141
##
## Critical value for a significance level of:
##                  10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

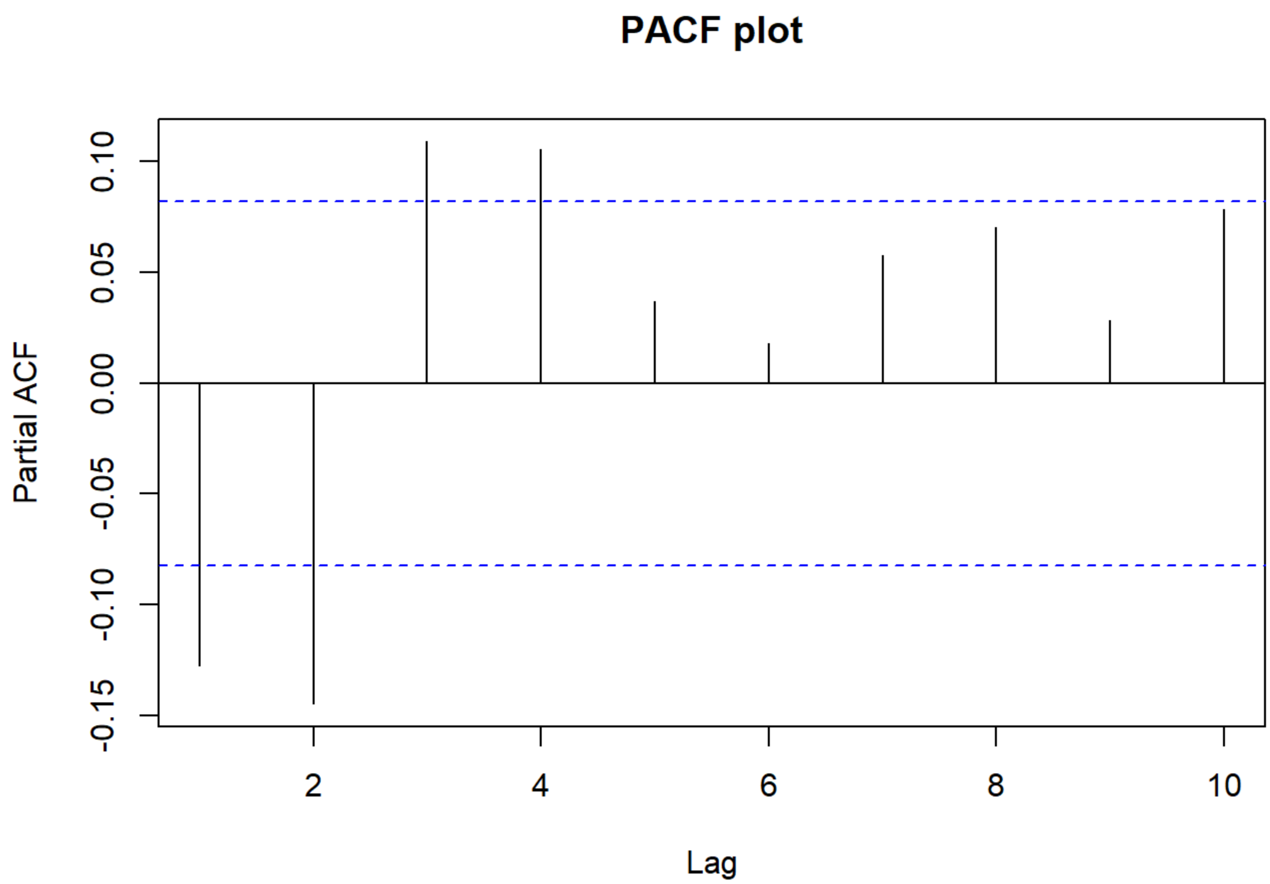
ACF plot:

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



PACF plot:

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



Fitting the model:

```
est_3=arima(data_japan$Unemployment_Rate, order=c(2,1,1), method = "ML")
summary(est_3)
```

```
##
## Call:
## arima(x = data_japan$Unemployment_Rate, order = c(2, 1, 1), method = "ML")
##
## Coefficients:
##           ar1      ar2     ma1
##       -0.4380  -0.1932  0.2965
## s.e.   0.1701   0.0434  0.1706
##
## sigma^2 estimated as 1.057e-06:  log likelihood = 3101.85,  aic = -6195.69
##
## Training set error measures:
##                        ME        RMSE          MAE          MPE     MAPE
## Training set 1.413312e-05 0.001027239 0.0007748223 -0.007266005 2.487642
##                  MASE        ACF1
## Training set 1.040423 0.005446084
```

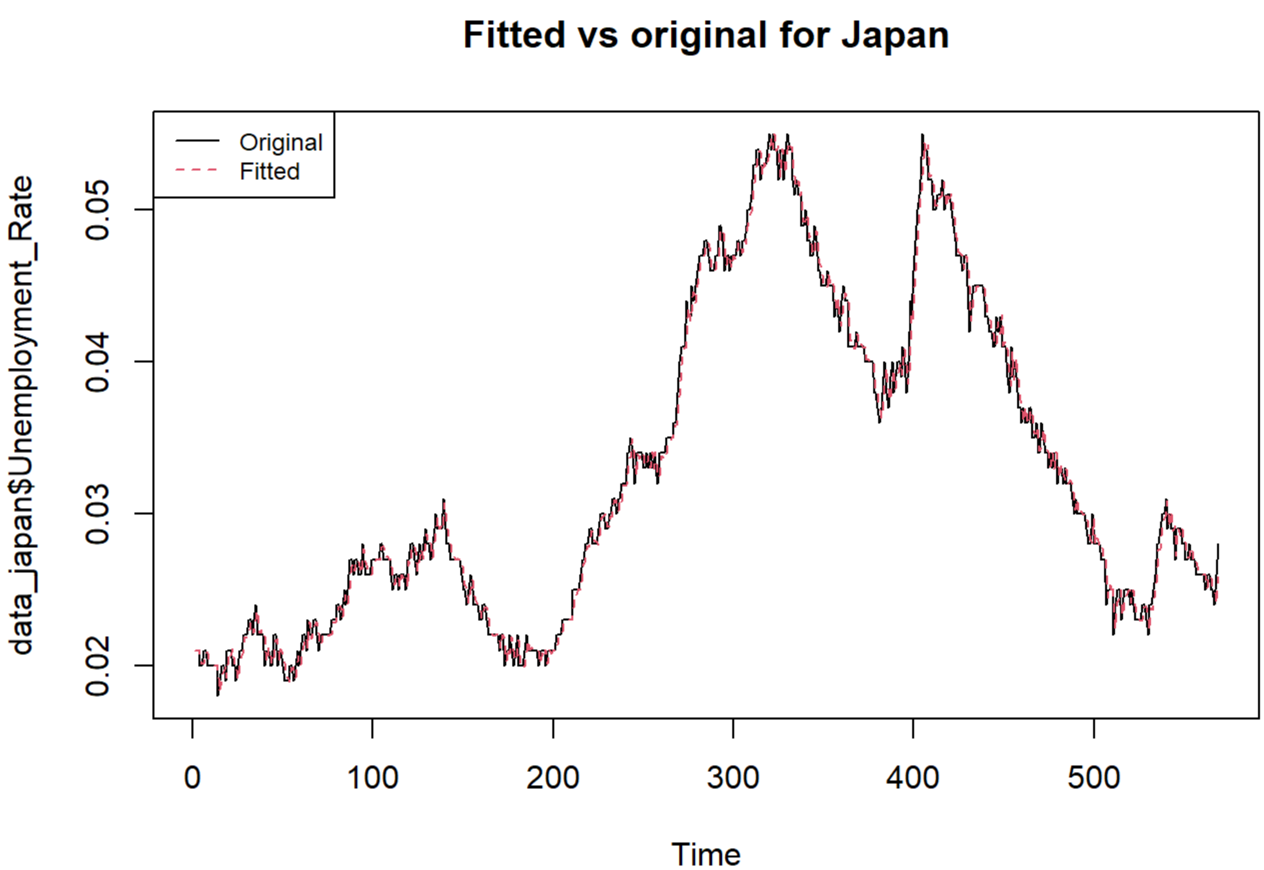
Test of significance of individual coefficients:

```
suppressWarnings(library(lmtest))
coeftest(est_3)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.438002   0.170148 -2.5742   0.01005 *
## ar2 -0.193180   0.043404 -4.4507 8.558e-06 ***
## ma1  0.296510   0.170577  1.7383   0.08216 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plot of Fitted vs Original on the actual data:

```
res=residuals(est_3)
data_fit=data_japan$Unemployment_Rate-res
ts.plot(data_japan$Unemployment_Rate, type="l", main="Fitted vs original for Japan")
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)
```



Getting April forecast:

```
future_unemp_pred=predict(est_3, n.ahead=1, se.fit=FALSE, method="ML")
print(future_unemp_pred)
```

```
## Time Series:
## Start = 570
## End = 570
## Frequency = 1
## [1] 0.02736623
```

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

```
upper=as.vector(future_unemp_pred)+(1.96*(sqrt(est_3$sigma2)))
lower=as.vector(future_unemp_pred)-(1.96*(sqrt(est_3$sigma2)))
```

Upper limit for April 2023 forecast:

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

```
## [1] 0.02938139
```

Lower limit for April 2023 forecast:

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

```
## [1] 0.02535107
```

April 23 forecast - 2.74 %

Upper & Lower limits - (2.535 %, 2.938 %)