```
install.packages("forecast")

install.packages("ggplot2")  # Required for checkresiduals()

library(forecast)

library(tseries)

library(ggplot2)


Dataset <- read.csv("Datasett.csv")

# Convert columns to time series (Monthly Data: frequency = 12)

CPIHome_ts <- ts(Dataset$CPI_Home, frequency=12, start=c(2015,1))

CPIForeign_ts <- ts(Dataset$CPI_Foreign, frequency=12,
start=c(2015,1))

RealExchangeRate_ts <- ts(Dataset$Real_Exchange_Rate, frequency=12,
start=c(2015,1))

Nominalexchange_ts <- ts(Dataset$Nominal_Exchange_Rate, frequency=12,
start=c(2015,1))


# Summarize raw data (before log transformation)

cat("Summary Before Log Transformation:\n")

summary(Dataset[, c("CPI_Home", "CPI_Foreign", "Real_Exchange_Rate",
"Nominal_Exchange_Rate")])




# Log transformation

Nominalexchange_logts <- log(Nominalexchange_ts)

CPIForeign_logts <- log(CPIForeign_ts)

RealExchangeRate_logts <- log(RealExchangeRate_ts)
```

```
CPIHome_logts <- log(CPIHome_ts)


# Summarize data after log transformation

cat("\nSummary After Log Transformation:\n")

summary(data.frame(CPIHome_logts, CPIForeign_logts,
RealExchangeRate_logts, Nominalexchange_logts))


# Plot log-transformed time series before differencing

par(mfrow=c(2,2))

plot.ts(Nominalexchange_logts, main="Log Nominal Exchange Rate (Before
Differencing)")

plot.ts(RealExchangeRate_logts, main="Log Real Exchange Rate (Before
Differencing)")

plot.ts(CPIHome_logts, main="Log CPI Home (Before Differencing)")

plot.ts(CPIForeign_logts, main="Log CPI Foreign (Before
Differencing)")

par(mfrow=c(1,1))  # Reset layout


# ADF test before differencing (for all 4 variables)

cat("\nADF Test Before Differencing:\n")

cat("\nADF Test: Log CPI Home\n")

adf_CPIHome_before <- adf.test(CPIHome_logts)

print(adf_CPIHome_before)

cat("\nADF Test: Log CPI Foreign\n")

adf_CPIForeign_before <- adf.test(CPIForeign_logts)

print(adf_CPIForeign_before)
```

```
cat("\nADF Test: Log Real Exchange Rate\n")

adf_RealEXR_before <- adf.test(RealExchangeRate_logts)

print(adf_RealEXR_before)

cat("\nADF Test: Log Nominal Exchange Rate\n")

adf_NominalEXR_before <- adf.test(Nominalexchange_logts)

print(adf_NominalEXR_before)


# First-order differencing (Monthly Data)

CPIHome_d1logts <- diff(CPIHome_logts, differences=1)

CPIForeign_d1logts <- diff(CPIForeign_logts, differences=1)

RealEXR_d1logts <- diff(RealExchangeRate_logts, differences=1)

NominalEXR_d1logts <- diff(Nominalexchange_logts, differences=1)



# Plot differenced series

par(mfrow=c(2,2))

plot.ts(CPIHome_d1logts, main="Log CPI Home after Differencing")

plot.ts(CPIForeign_d1logts, main="Log CPI Foreign after Differencing")

plot.ts(RealEXR_d1logts, main="Log Real Exchange Rate after
Differencing")

plot.ts(NominalEXR_d1logts, main="Log Nominal Exchange Rate after
Differencing")

par(mfrow=c(1,1))  # Reset layout


# ADF test after differencing (for all 4 variables)

cat("\nADF Test After Differencing:\n")
```

```
cat("\nADF Test: Log CPI Home (Differenced)\n")

adf_CPIHome_after <- adf.test(CPIHome_d1logts)

print(adf_CPIHome_after)

cat("\nADF Test: Log CPI Foreign (Differenced)\n")

adf_CPIForeign_after <- adf.test(CPIForeign_d1logts)

print(adf_CPIForeign_after)

cat("\nADF Test: Log Real Exchange Rate (Differenced)\n")

adf_RealEXR_after <- adf.test(RealEXR_d1logts)

print(adf_RealEXR_after)

cat("\nADF Test: Log Nominal Exchange Rate (Differenced)\n")

adf_NominalEXR_after <- adf.test(NominalEXR_d1logts)

print(adf_NominalEXR_after)



# Cointegration Test for Absolute PPP

cat("\nAbsolute PPP: Unit Root Test on Log Real Exchange Rate\n")

# ADF test for log real exchange rate

adf_RealEXR <- adf.test(RealExchangeRate_logts,
alternative="stationary")

print(adf_RealEXR)



# Cointegration Test for Relative PPP

cat("\nRelative PPP: Unit Root Test on Log CPI Ratio\n")
```

```
# Construct log CPI Ratio (RATIO APPROACH)
CPI_Ratio_logts <- CPIHome_logts - CPIForeign_logts
# ADF test for log CPI ratio
adf_CPI_Ratio <- adf.test(CPI_Ratio_logts, alternative="stationary")
print(adf_CPI_Ratio)



# OLS Regression for Relative PPP
PPP_model <- lm(Nominalexchange_logts ~ CPI_Ratio_logts)
summary(PPP_model)

# Extract residuals
PPP_residuals <- residuals(PPP_model)

# ADF test for residuals
cat("\nADF Test for OLS Residuals \n")
adf_Residuals_1 <- adf.test(PPP_residuals, alternative="stationary")
print(adf_Residuals_1)
```

```
# OLS regression (log_s on log_p and log_pstar) INSTEAD OF USING RATIO
APPROACH, WE SEPERATE PARAMETER B1,B2

cat("\nOLS Regression Model\n")

ols_model <- lm( Nominalexchange_logts~ CPIForeign_logts +
CPIHome_logts)

# Print regression results

cat("\nRegression Results:\n")

# Extract residuals from OLS regression

ols_residuals <- resid(ols_model)

# Print first few residuals to check

cat("\nFirst few residuals:\n")

# ADF Test on Residuals

cat("\nADF Test on Residuals

adf_residuals_2 <- adf.test(ols_residuals)

print(adf_residuals_2)




#PACF AND ACF PLOT


par(mfrow = c(1, 2))  # 1 row, 2 columns

acf(RealEXR_d1logts, lag.max = 20, main = "ACF of RealEXR_d1logts")

pacf(RealEXR_d1logts, lag.max = 20, main = "PACF of RealEXR_d1logts")

par(mfrow = c(1, 1))  # Reset layout
```

```
# Fit the six ARIMA models on log-transformed Real Exchange Rate
```

Fit the six ARIMA models on log-transformed Real Exchange Rate

```
fix_1 <- Arima(RealEXR_d1logts, order=c(1,0,0))  # ARIMA(1,1,0)

fix_2 <- Arima(RealEXR_d1logts, order=c(2,0,0))  # ARIMA(2,1,0)

fix_3 <- Arima(RealEXR_d1logts, order=c(0,0,1))  # ARIMA(0,1,1)

fix_4 <- Arima(RealEXR_d1logts, order=c(0,0,2))  # ARIMA(0,1,2)

fix_5 <- Arima(RealEXR_d1logts, order=c(1,0,1))  # ARIMA(1,1,1)

fix_6 <- Arima(RealEXR_d1logts, order=c(2,0,1))  # ARIMA(2,1,1)
```

```
fix_1

fix_2

fix_3

fix_4

fix_5

fix_6
```

```
tsdiag(fix_1)

tsdiag(fix_2)

tsdiag(fix_3)

tsdiag(fix_4)

tsdiag(fix_5)

tsdiag(fix_6)
```

```
# Store all models in a list
models <- list(
  `ARIMA(1,0,0)` = fix_1,
  `ARIMA(2,0,0)` = fix_2,
  `ARIMA(0,0,1)` = fix_3,
  `ARIMA(0,0,2)` = fix_4,
  `ARIMA(1,0,1)` = fix_5,
  `ARIMA(2,0,1)` = fix_6
)

# Function to extract diagnostics from each model
diagnostics <- lapply(models, function(model) {

  # Coefficients
  coefs <- coef(model)
  coefs_se <- sqrt(diag(vcov(model)))  # Standard Errors
  t_values <- coefs / coefs_se  # t-values

  # Ljung-Box test for residuals at lags 4, 8, 12
  Q4 <- Box.test(residuals(model), lag = 4, type = "Ljung-Box")
  Q8 <- Box.test(residuals(model), lag = 8, type = "Ljung-Box")
  Q12 <- Box.test(residuals(model), lag = 12, type = "Ljung-Box")

  return(list(
```

```
    coefficients = coefs,
    t_values = t_values,
    Q4_stat = Q4$statistic, Q4_p_value = Q4$p.value,
    Q8_stat = Q8$statistic, Q8_p_value = Q8$p.value,
    Q12_stat = Q12$statistic, Q12_p_value = Q12$p.value
  ))
})

# Print the results
for (i in 1:length(diagnostics)) {
  cat("\nModel:", names(diagnostics)[i], "\n")

  # Coefficients and t-values
  cat("Coefficients:\n")
  print(diagnostics[[i]]$coefficients)

  cat("t-values:\n")
  print(diagnostics[[i]]$t_values)

  # Ljung-Box Q-test results at lags 4, 8, and 12
  cat("\nLjung-Box Q-test Results:\n")
  cat("Lag 4: Q-statistic =", diagnostics[[i]]$Q4_stat, ", p-value =",
diagnostics[[i]]$Q4_p_value, "\n")
  cat("Lag 8: Q-statistic =", diagnostics[[i]]$Q8_stat, ", p-value =",
diagnostics[[i]]$Q8_p_value, "\n")
```

```
  cat("Lag 12: Q-statistic =", diagnostics[[i]]$Q12_stat, ", p-value
=", diagnostics[[i]]$Q12_p_value, "\n")


  cat("\n----------------------------\n")

}
```

```
Model: ARIMA(1,0,0)
Coefficients:
          ar1     intercept
 0.2258971870 -0.0005625395
t-values:
       ar1  intercept
 2.4705155 -0.2493568

Ljung-Box Q-test Results:
Lag 4: Q-statistic = 3.689307 , p-value = 0.4496831
Lag 8: Q-statistic = 7.56778 , p-value = 0.4767864
Lag 12: Q-statistic = 12.52603 , p-value = 0.4044101

----------------------------
```

```
Model: ARIMA(2,0,0)
Coefficients:
          ar1           ar2     intercept
 0.2675090676 -0.1722709171 -0.0006166743
t-values:
       ar1        ar2  intercept
 2.8774109 -1.8611861 -0.3230193

Ljung-Box Q-test Results:
Lag 4: Q-statistic = 0.687867 , p-value = 0.9528167
Lag 8: Q-statistic = 4.86546 , p-value = 0.7718545
Lag 12: Q-statistic = 8.40065 , p-value = 0.7530898

----------------------------
```

```
Model: ARIMA(0,0,1)
Coefficients:
          mal     intercept
 0.3096486935 -0.0005635017
t-values:
       mal  intercept
 3.1937299 -0.2488696

Ljung-Box Q-test Results:
Lag 4: Q-statistic = 1.524884 , p-value = 0.8222245
Lag 8: Q-statistic = 5.697266 , p-value = 0.6811002
Lag 12: Q-statistic = 10.13075 , p-value = 0.6044913
```

```
Model: ARIMA(0,0,2)
Coefficients:
          mal            ma2     intercept
 0.2687385009 -0.0880224044 -0.0006129258
t-values:
       mal        ma2  intercept
 2.8500873 -1.0184577 -0.3010018

Ljung-Box Q-test Results:
Lag 4: Q-statistic = 0.4411697 , p-value = 0.9789696
Lag 8: Q-statistic = 4.988152 , p-value = 0.7588417
Lag 12: Q-statistic = 9.132924 , p-value = 0.691537
```

```
Model: ARIMA(1,0,1)
Coefficients:
         arl          mal    intercept
-0.190534395  0.473307206 -0.000594908
t-values:
       arl        mal  intercept
-0.8678437  2.5368965 -0.2786390

Ljung-Box Q-test Results:
Lag 4: Q-statistic = 0.7109504 , p-value = 0.9499717
Lag 8: Q-statistic = 5.17441 , p-value = 0.7387827
Lag 12: Q-statistic = 9.516554 , p-value = 0.6582907
```

```
Model: ARIMA(2,0,1)
Coefficients:
          arl           ar2           mal     intercept
 0.0141621797 -0.1180088538  0.2613355413 -0.0006202591
t-values:
        arl         ar2         mal   intercept
 0.03745737 -0.86549347  0.70282255 -0.31504094

Ljung-Box Q-test Results:
Lag 4: Q-statistic = 0.2756437 , p-value = 0.9913317
Lag 8: Q-statistic = 4.8672 , p-value = 0.7716711
Lag 12: Q-statistic = 8.720099 , p-value = 0.7266335

----------------------------
```

```
library(forecast)

# Forecast 12 periods ahead (can change to any number)
forecast_horizon <- 12

# Generate forecast
forecast_result <- forecast(model_3, h = forecast_horizon)

# Print forecast values
print(forecast_result)
```

# Plot the forecast with confidence intervals
plot(forecast_result, main = "Forecast using ARIMA(0,1,1)")

```
fix_11 <- Arima(RealExchangeRate_logts, order=c(1,1,0))  #
ARIMA(1,1,0)

fix_22 <- Arima(RealExchangeRate_logts, order=c(2,1,0))  #
ARIMA(2,1,0)

fix_33 <- Arima(RealExchangeRate_logts, order=c(0,1,1))  #
ARIMA(0,1,1)

fix_44 <- Arima(RealExchangeRate_logts, order=c(0,1,2))  #
ARIMA(0,1,2)

fix_55 <- Arima(RealExchangeRate_logts, order=c(1,1,1))  #
ARIMA(1,1,1)

fix_66 <- Arima(RealExchangeRate_logts, order=c(2,1,1))  #
ARIMA(2,1,1)



fix_11

fix_22

fix_33

fix_44

fix_55

fix_66
```