Lecture 10, November 10, 2025

Exam Preparation

Marta Boczon

Department of Economics

Copnehagen Business School

mbo.eco@cbs.dk

In [52]:
```
#install.packages("quantmod")
#install.packages("fredr")
#install.packages("ggfortify")
#install.packages('urca')
#install.packages("tseries")
#install.packages("forecast")
#install.packages("dynlm")
#install.packages("stargazer")
#install.packages("pracma")
#install.packages("dLagM")
#install.packages("gets")
#install.packages("car")
#install.packages("lmtest")
#install.packages("vars")
#install.packages("tseries")
#install.packages("strucchange")
#install.packages("graphics")
#install.packages("grDevices")
#install.packages("tsDyn")
options(warn=-1)
```

Roadmap

Today we will practice the material we have covered so far. There are **eight exercises** that touch on all the main topics and models discussed in previous sessions. You will have about **15 minutes** to solve each exercise, after which I will go through the solution before we move on to the next one.

Except for the first exercise, the remaining exercises (2–8) should be completed one by one. You are welcome to work **in groups, pairs, or individually** — whichever you prefer. Feel free to **ask questions at any point** if something is unclear or if you would like to discuss your reasoning.

By the end of today's session, you should feel more confident applying the methods we learned — from ARIMA and ADL models to VAR and cointegration analysis — and be able to interpret model results in both theoretical and empirical contexts.

Exercise 1

Provide 1–3 sentence answers to the following questions.

1. Define weak stationarity.

2. Discuss the difference between weak and strict stationarity.

3. Explain the difference between stationary and trend-stationary data.

4. Discuss the shape of the ACF and PACF of an AR(3) process.

5. Discuss the shape of the ACF and PACF of an MA(5) process.

6. Discuss how information criteria can be used to determine the best model.

7. Discuss the iterative procedure of eliminating insignificant coefficients.

8. Discuss the recursive pseudo-out-of-sample forecasting procedure.

9. Discuss the difference between AR and MA models.
   v

10. Discuss the difference between ADL and VAR models.

11. Explain the concept of Granger causality.

12. Explain the difference between DF and ADF tests.

13. Discuss the difference between the Engle–Granger and Phillips–Ouliaris tests for cointegration.

14. Explain what it means for a series to be integrated of order 1.

15. Explain the concept of cointegration.

16. Explain the difference between the AIC and BIC information criteria.

17. Describe the concept of unit roots in time series.

18. Define an impulse response function in the context of VAR models.

19. Explain the Box–Jenkins methodology.

20. Explain how cointegration differs from correlation.

21. Explain the difference between a VAR model and a VECM.

22. Why might one choose a VAR model over an ADL model?

23. Why is it necessary to check for stationarity in the variables before estimating a VAR model?

24. How can cointegration be used in economic policy analysis?

25. How do you interpret a large positive error correction term in a VECM?

Solution to Exercise 1

1. **Define weak stationarity.**
   Weak stationarity, or second-order stationarity, means that a time series has a constant mean, a constant variance, and an autocovariance that depends only on the lag, not on time.

2. **Discuss the difference between weak and strict stationarity.**
   Strict stationarity requires that the entire distribution of a series is time-invariant, while weak stationarity only requires constancy of the mean, variance, and autocovariance function. Therefore, strict stationarity is a stronger condition than weak stationarity.

3. **Explain the difference between stationary and trend-stationary data.**
   Stationary data fluctuate around a constant mean without trends, whereas trend-stationary data have a deterministic trend (e.g., linear) that can be removed to make the series stationary.

4. **Discuss the shape of the ACF and PACF of an AR(3) process.**
   For an AR(3) process, the PACF cuts off after lag 3, while the ACF decays gradually over time. The specific shape of the ACF depends on the model coefficients.
5. **Discuss the shape of the ACF and PACF of an MA(5) process.**
   In an MA(5) process, the ACF cuts off after lag 5, while the PACF decays gradually without a clear cutoff, reflecting the finite moving average structure.

6. **Discuss how information criteria can be used to determine the best model.**
   Information criteria like AIC and BIC balance model fit and complexity, with lower values indicating better models. Generally, AIC favors more complex models, while BIC penalizes complexity more heavily.

7. **Discuss the iterative procedure of eliminating insignificant coefficients.**
   Start with a full model, then iteratively remove statistically insignificant coefficients based on p-values until only significant terms remain, refining the model’s simplicity without losing predictive accuracy.

8. **Discuss the recursive pseudo-out-of-sample forecasting procedure.**
   This involves generating forecasts for data points excluded from estimation, computing forecast errors, and comparing models based on their out-of-sample performance over multiple iterations.

9. **Discuss the difference between AR and MA models.**
   AR models express the current value as a function of past values, while MA models represent it as a function of past shocks. AR models have a gradually decaying PACF, whereas MA models have a sharp ACF cutoff.

10. **Discuss the difference between ADL and VAR models.**
    ADL models include external predictors in a univariate framework, whereas VAR models are multivariate, modeling each variable as a function of its own lags and the lags of all other variables.

11. **Explain the concept of Granger causality.**
    Granger causality tests whether past values of one variable contain information that helps predict another variable beyond what is explained by its own past values.

12. **Explain the difference between DF and ADF tests.**
    The DF test checks for a unit root without lagged differenced terms, while the ADF test includes them to control for autocorrelation, providing a more robust test for unit roots.

13. **Discuss the difference between the Engle–Granger and Phillips–Ouliaris tests for cointegration.**
    The Engle–Granger test is a two-step procedure sensitive to the choice of dependent variable, while the Phillips–Ouliaris test avoids this issue by testing the residuals from a cointegrating regression without assuming a particular dependent variable.

14. **Explain what it means for a series to be integrated of order 1.**
    A series is integrated of order 1 if it becomes stationary after first differencing, meaning it has a unit root in levels.

15. **Explain the concept of cointegration.**
    Cointegration occurs when two or more non-stationary series share a common stochastic trend, so that a linear combination of them is stationary, indicating a long-run equilibrium relationship.

16. **Explain the difference between AIC and BIC information criteria.**
    Both AIC and BIC balance model fit with complexity, but BIC imposes a stronger penalty on model complexity and tends to select more parsimonious models, especially in large samples.

17. **Describe the concept of unit roots in time series.**
    A unit root indicates a non-stationary process where shocks have permanent effects, meaning the series does not revert to a stable mean and often requires differencing to achieve stationarity.

18. **Define an impulse response function in the context of VAR models.**
    An impulse response function shows how a one-time shock to one variable affects the current and future values of all variables in a VAR model, illustrating dynamic interactions over time.

19. **Explain the Box–Jenkins methodology.**
    The Box–Jenkins methodology is an iterative approach for identifying, estimating, and diagnosing ARIMA models, involving steps such as model identification, parameter estimation, and residual diagnostics.

20. **Explain how cointegration differs from correlation.**
    Cointegration reflects a long-term equilibrium relationship among non-stationary series, while correlation only measures short-term comovement at a given point in time.

21. **Explain the difference between a VAR model and a VECM.**
    A VAR model is applied to stationary data, while a VECM (Vector Error Correction Model) is used when series are cointegrated, separating short-run dynamics from long-run equilibrium adjustments.

22. **Why might one choose a VAR model over an ADL model?**
    A VAR model is preferred when analyzing multiple interdependent time series, as it allows for feedback and mutual interactions among variables, unlike the single-equation ADL model.

23. **Why is it necessary to check for stationarity in the variables before estimating a VAR model?**
    Stationarity ensures that relationships between variables remain stable over time. Non-stationary data can lead to spurious results, so differencing or a VECM formulation is needed if variables are cointegrated.

24. **How can cointegration be used in economic policy analysis?**
    Cointegration helps identify long-term equilibrium relationships among economic variables, guiding policymakers by revealing stable links that persist despite short-term fluctuations.

25. **How do you interpret a large positive error correction term in a VECM?**
    A large positive error correction term indicates a strong adjustment toward the long-run equilibrium, meaning short-term deviations are corrected rapidly.

Exchange Rates

Mexican Peso-US Dollar Exchange Rate Mexican Pesos to U.S. Dollar Spot Exchange Rate (DEXMXUS)

- **Frequency:** Daily
- **Units:** Mexican Pesos to One U.S. Dollar
- **Seasonal adjustment:** Not seasonally adjusted
- **Data availability:** 1993-11-08 through 2025-10-31

Indian Rupee-US Dollar Exchange rate Indian Rupees to U.S. Dollar Spot Exchange Rate (DEXINUS)

- **Frequency:** Daily
- **Units:**Indian Rupee to One U.S. Dollar,
- **Seasonal adjustment:** Not seasonally adjusted
- **Data availability:** 1973-01-02 through 2025-10-31

In [53]:

```
imp1 = read.csv("DEXMXUS.csv")   # rate data from CSV into a data frame
imp2 = read.csv("DEXINUS.csv")    # rate data from CSV into a data frame

head(imp1)  # quickly preview the first 6 rows of TB3MS.csv
head(imp2)  # quickly preview the first 6 rows of GS10.csv

tail(imp1)  # see the last 6 rows
tail(imp2)  # see the last 6 rows
```

A data.frame: 6 × 2

	observation_date	DEXMXUS
	<chr>	<dbl>
1	1993-11-08	3.152
2	1993-11-09	3.240
3	1993-11-10	3.240
4	1993-11-11	NA
5	1993-11-12	3.240
6	1993-11-15	3.215

A data.frame: 6 × 2

	observation_date	DEXINUS
	<chr>	<dbl>
1	1973-01-02	8.02
2	1973-01-03	8.02
3	1973-01-04	8.00
4	1973-01-05	8.01
5	1973-01-08	8.00
6	1973-01-09	8.00

A data.frame: 6 × 2

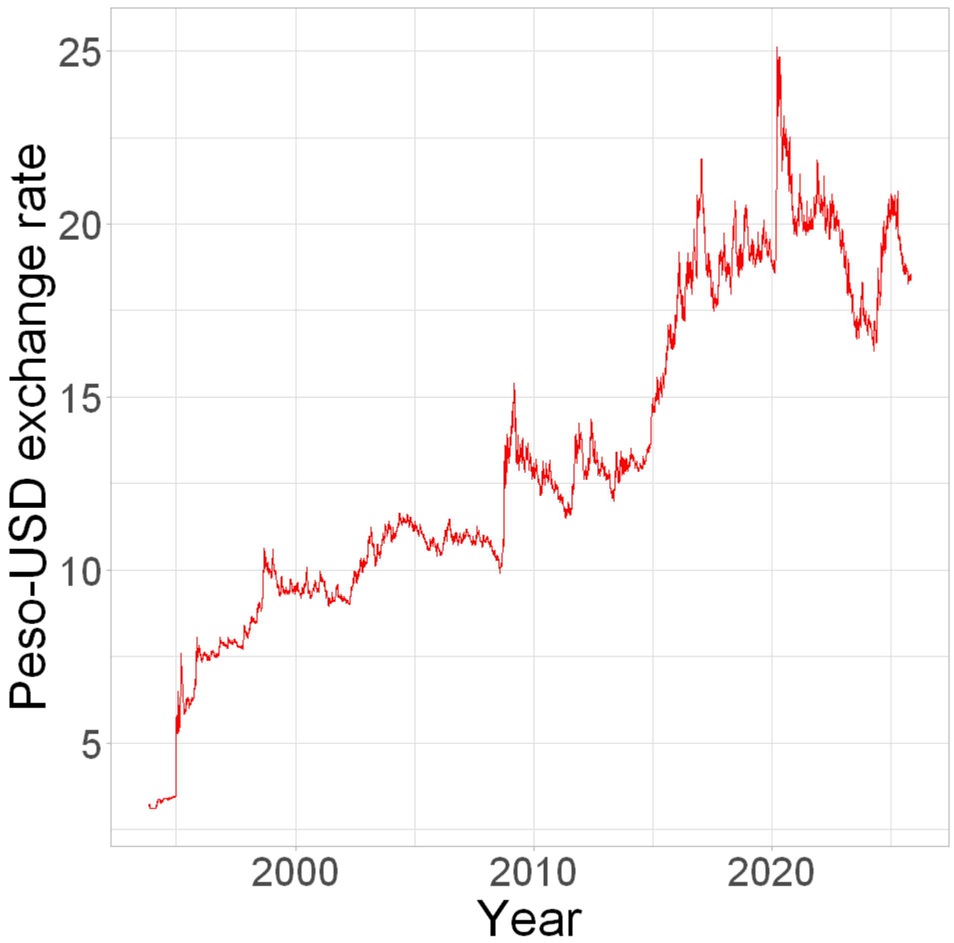
	observation_date	DEXMXUS
	<chr>	<dbl>
8340	2025-10-24	18.3930
8341	2025-10-27	18.3920
8342	2025-10-28	18.4133
8343	2025-10-29	18.4194
8344	2025-10-30	18.5430
8345	2025-10-31	18.5465

A data.frame: 6 × 2

	observation_date	DEXINUS
	<chr>	<dbl>
13779	2025-10-24	87.83
13780	2025-10-27	88.24
13781	2025-10-28	88.21
13782	2025-10-29	88.20
13783	2025-10-30	88.64
13784	2025-10-31	88.75

In [54]:
```
# Load required libraries
library(xts)        # Date-indexed time series for irregular trading calendars

# Create an xts series from raw data (imp1: Date in col 1, price/rate in col 2)
# Why xts (vs ts): markets skip weekends/holidays; xts preserves real dates and missing days,
# aligns assets by timestamps, and supports finance-friendly ops (apply.weekly, rollapply).

d1   = as.Date(imp1[, 1])                 # trading dates
mxus = xts(imp1[, 2], d1)                 # date-indexed series
colnames(mxus) = "MXUS"
mxus = na.omit(mxus)                      # drop any missing observation
```

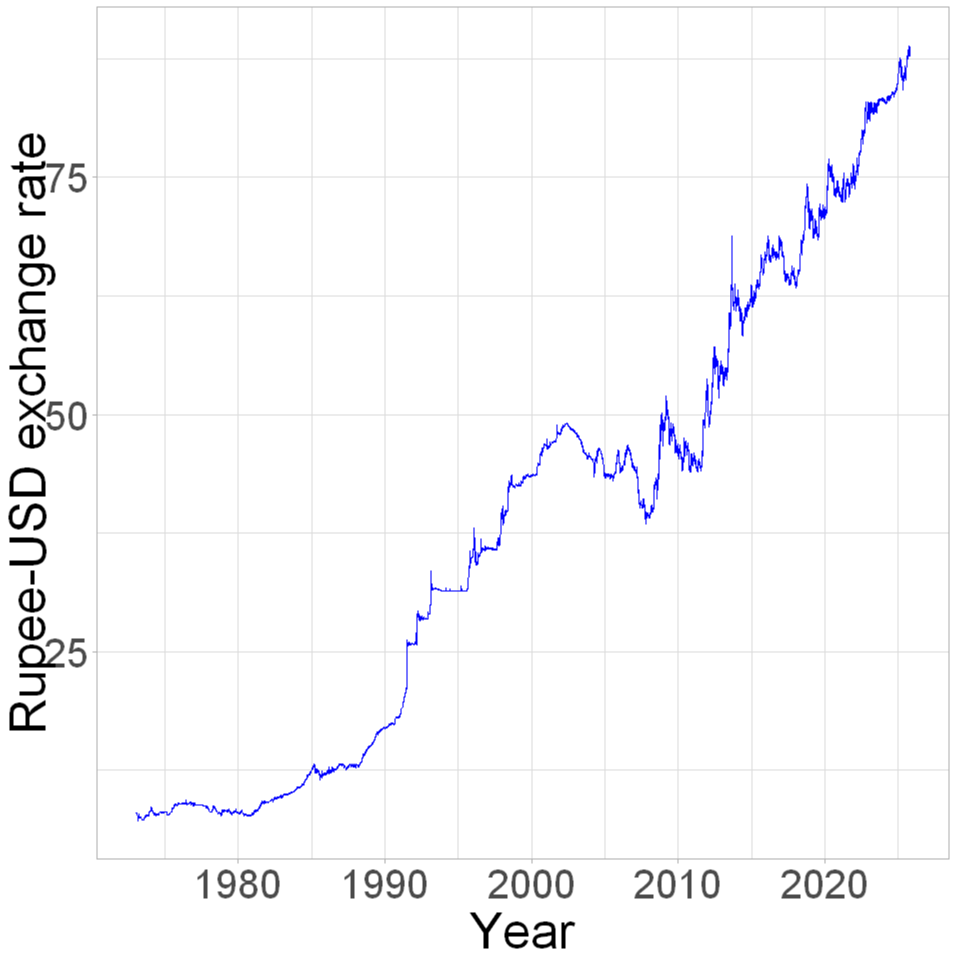
In [55]:
```
library(ggplot2)
library(ggfortify)
options(repr.plot.width=8, repr.plot.height=8)
fig = autoplot(mxus, colour = 'red')
fig = fig + theme(aspect.ratio=1) +
theme_light() +
theme(aspect.ratio=1) +
theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +
theme(text=element_text(size=30)) +
labs(x = "Year") +
labs(y = "Peso-USD exchange rate")
fig
```



In [56]:
```
# Load required libraries
library(xts)        # Date-indexed time series for irregular trading calendars

# Create an xts series from raw data (imp1: Date in col 1, price/rate in col 2)
# Why xts (vs ts): markets skip weekends/holidays; xts preserves real dates and missing days,
# aligns assets by timestamps, and supports finance-friendly ops (apply.weekly, rollapply).

d2   = as.Date(imp2[, 1])                 # trading dates
inus = xts(imp2[, 2], d2)                 # date-indexed series
colnames(inus) = "INUS"
inus = na.omit(inus)                      # drop any missing observation
```

In [57]:
```
library(ggplot2)
library(ggfortify)
options(repr.plot.width=8, repr.plot.height=8)
fig = autoplot(inus, colour = 'blue')
fig = fig + theme(aspect.ratio=1) +
theme_light() +
theme(aspect.ratio=1) +
theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +
```

```
theme(text=element_text(size=30)) +
labs(x = "Year") +
labs(y = "Rupee-USD exchange rate")
fig
```



Important Tips

Working with `xts` vs `ts` objects

There are several differences when working with `xts` objects as opposed to `ts` objects:

1. At any stage of the analysis, missing values need to be removed, for example using `na.omit()`.
2. To construct a joint time series object, instead of using `ts.intersect()`, use:
   `data = merge(x, y, join = 'inner')`
3. When estimating a model using `dynlm`, the data needs to be specified as a zoo object:
   `model = dynlm(y ~ L(x, 1) + L(y, 1), data = as.zoo(data))`

Exercise 2

1. Explain the role of deterministic components in the ADF test for a unit root.

2. Is the Mexican Peso–US Dollar exchange rate integrated of order 2, integrated of order 1, trend-stationary, or stationary in levels?

3. Is the Indian Rupee–US Dollar exchange rate integrated of order 2, integrated of order 1, trend-stationary, or stationary in levels?

Solution to Exercise 2

1. Deterministic components in the ADF test—such as a constant (drift) or a time trend—account for predictable patterns in the data that are not due to stochastic trends. Including them ensures that the test correctly distinguishes between a genuine unit root and non-stationarity caused by deterministic factors. Choosing the appropriate specification is crucial, as omitting relevant components can bias the test results.

2. **Mexican Peso–US Dollar exchange rate:** Integrated of order 1, $I(1)$.

3. **Indian Rupee–US Dollar exchange rate:** Integrated of order 1, $I(1)$.

In [58]:
```
library(urca)
summary(ur.df(mxus, type='trend', lags=30, selectlags="AIC"))
```

```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression trend


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)

Residuals:
     Min       1Q   Median       3Q      Max
-1.23187 -0.04467 -0.00590  0.03687  1.34506

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.792e-02  5.087e-03   3.523 0.000429 ***
z.lag.1      -2.584e-03  8.044e-04  -3.212 0.001323 **
tt            4.519e-06  1.643e-06   2.750 0.005976 **
z.diff.lag1  -1.871e-03  1.121e-02  -0.167 0.867417
z.diff.lag2   1.101e-02  1.120e-02   0.983 0.325491
z.diff.lag3  -1.536e-02  1.120e-02  -1.371 0.170279
z.diff.lag4  -1.667e-02  1.120e-02  -1.489 0.136470
z.diff.lag5   2.387e-02  1.119e-02   2.133 0.032967 *
z.diff.lag6  -3.474e-02  1.120e-02  -3.103 0.001921 **
z.diff.lag7   4.322e-02  1.120e-02   3.859 0.000115 ***
z.diff.lag8   2.295e-02  1.121e-02   2.047 0.040689 *
z.diff.lag9   3.292e-02  1.121e-02   2.935 0.003340 **
z.diff.lag10 -3.876e-02  1.122e-02  -3.456 0.000552 ***
z.diff.lag11  1.946e-02  1.122e-02   1.734 0.082919 .
z.diff.lag12  1.769e-02  1.122e-02   1.577 0.114891
z.diff.lag13 -4.411e-04  1.122e-02  -0.039 0.968639
z.diff.lag14 -1.339e-02  1.122e-02  -1.193 0.232836
z.diff.lag15  7.522e-03  1.122e-02   0.670 0.502647
z.diff.lag16 -3.037e-02  1.122e-02  -2.707 0.006810 **
z.diff.lag17  2.476e-02  1.122e-02   2.206 0.027404 *
z.diff.lag18 -1.915e-02  1.122e-02  -1.707 0.087802 .
z.diff.lag19 -4.438e-03  1.121e-02  -0.396 0.692335
z.diff.lag20  8.178e-03  1.121e-02   0.729 0.465820
z.diff.lag21 -1.990e-02  1.121e-02  -1.776 0.075723 .
z.diff.lag22 -1.024e-03  1.120e-02  -0.091 0.927178
z.diff.lag23 -2.383e-02  1.120e-02  -2.129 0.033321 *
z.diff.lag24  3.409e-02  1.120e-02   3.044 0.002343 **
z.diff.lag25  1.993e-02  1.120e-02   1.779 0.075253 .
z.diff.lag26 -3.518e-02  1.120e-02  -3.140 0.001698 **
z.diff.lag27  2.166e-02  1.121e-02   1.932 0.053443 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1128 on 7958 degrees of freedom
Multiple R-squared:  0.01597,	Adjusted R-squared:  0.01238
F-statistic: 4.452 on 29 and 7958 DF,  p-value: 1.886e-14


Value of test-statistic is: -3.212 4.4479 5.5068

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -3.96 -3.41 -3.12
phi2  6.09  4.68  4.03
phi3  8.27  6.25  5.34
```

Interpretation of the Unit Root Test for the Mexican Peso–US Dollar Exchange Rate

Key results: τ_3 = -3.212; φ_2 = 4.4479; φ_3 = 5.5068

- **φ_2 (constant) and φ_3 (constant + trend):** Both below their 5% critical values ⇒ **cannot reject** that the constant and trend are zero.

Conclusion: The test should be **rerun without the deterministic terms** (set `type = "none"`).

In [59]:
```
library(urca)
summary(ur.df(mxus, type='none', lags=30, selectlags="AIC"))
```

```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.23144 -0.04323 -0.00223  0.03865  1.33906 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
z.lag.1       7.332e-05  9.017e-05   0.813 0.416176    
z.diff.lag1  -2.960e-03  1.121e-02  -0.264 0.791688    
z.diff.lag2   9.963e-03  1.120e-02   0.890 0.373758    
z.diff.lag3  -1.645e-02  1.120e-02  -1.469 0.142007    
z.diff.lag4  -1.777e-02  1.119e-02  -1.587 0.112483    
z.diff.lag5   2.283e-02  1.119e-02   2.040 0.041404 *  
z.diff.lag6  -3.582e-02  1.120e-02  -3.199 0.001383 ** 
z.diff.lag7   4.223e-02  1.120e-02   3.770 0.000164 ***
z.diff.lag8   2.188e-02  1.121e-02   1.952 0.051013 .  
z.diff.lag9   3.182e-02  1.121e-02   2.837 0.004559 ** 
z.diff.lag10 -3.989e-02  1.122e-02  -3.556 0.000379 ***
z.diff.lag11  1.837e-02  1.122e-02   1.637 0.101609    
z.diff.lag12  1.661e-02  1.122e-02   1.481 0.138773    
z.diff.lag13 -1.553e-03  1.122e-02  -0.138 0.889898    
z.diff.lag14 -1.448e-02  1.122e-02  -1.290 0.196921    
z.diff.lag15  6.449e-03  1.122e-02   0.575 0.565514    
z.diff.lag16 -3.147e-02  1.122e-02  -2.805 0.005044 ** 
z.diff.lag17  2.369e-02  1.122e-02   2.111 0.034829 *  
z.diff.lag18 -2.022e-02  1.122e-02  -1.802 0.071520 .  
z.diff.lag19 -5.505e-03  1.121e-02  -0.491 0.623518    
z.diff.lag20  7.100e-03  1.121e-02   0.633 0.526602    
z.diff.lag21 -2.104e-02  1.120e-02  -1.878 0.060379 .  
z.diff.lag22 -2.091e-03  1.120e-02  -0.187 0.851868    
z.diff.lag23 -2.492e-02  1.120e-02  -2.226 0.026044 *  
z.diff.lag24  3.306e-02  1.120e-02   2.952 0.003166 ** 
z.diff.lag25  1.887e-02  1.120e-02   1.684 0.092128 .  
z.diff.lag26 -3.629e-02  1.120e-02  -3.239 0.001205 ** 
z.diff.lag27  2.060e-02  1.121e-02   1.837 0.066225 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1129 on 7960 degrees of freedom
Multiple R-squared:  0.01468,	Adjusted R-squared:  0.01122 
F-statistic: 4.236 on 28 and 7960 DF,  p-value: 5.253e-13


Value of test-statistic is: 0.8131 

Critical values for test statistics: 
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Interpretation of the Unit Root Test for the Mexican Peso–US Dollar Exchange Rate

Key results: τ_1 = 0.8131

- **τ_1 (unit root):** Much higher than the 5% critical value (−1.95) ⇒ **cannot reject** the null of a unit root.
- **Deterministic components:** The model has no deterministic terms (type = "none"), so the test directly examines stationarity in levels.

Conclusion: At the 5% significance level, we **cannot reject** the null hypothesis of a unit root. The series remains **non-stationary in levels** and should be **differenced once** before further analysis.

In [60]:

```
dmxus = diff(mxus)
dmxus = na.omit(dmxus)

library(urca)
summary(ur.df(dmxus, type='none', lags=30, selectlags="AIC"))
```

```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.23105 -0.04210 -0.00150  0.03953  1.34056 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
z.lag.1      -0.9928616  0.0575631 -17.248   <2e-16 ***
z.diff.lag1  -0.0099392  0.0564309  -0.176   0.8602    
z.diff.lag2   0.0001763  0.0553568   0.003   0.9975    
z.diff.lag3  -0.0161151  0.0542182  -0.297   0.7663    
z.diff.lag4  -0.0337237  0.0529756  -0.637   0.5244    
z.diff.lag5  -0.0107341  0.0517653  -0.207   0.8357    
z.diff.lag6  -0.0463972  0.0505296  -0.918   0.3585    
z.diff.lag7  -0.0040108  0.0493182  -0.081   0.9352    
z.diff.lag8   0.0180217  0.0480548   0.375   0.7077    
z.diff.lag9   0.0499887  0.0467722   1.069   0.2852    
z.diff.lag10  0.0102478  0.0455105   0.225   0.8218    
z.diff.lag11  0.0287743  0.0441428   0.652   0.5145    
z.diff.lag12  0.0455308  0.0428223   1.063   0.2877    
z.diff.lag13  0.0441242  0.0414423   1.065   0.2870    
z.diff.lag14  0.0297914  0.0400576   0.744   0.4571    
z.diff.lag15  0.0363845  0.0386259   0.942   0.3462    
z.diff.lag16  0.0050579  0.0370863   0.136   0.8915    
z.diff.lag17  0.0288995  0.0354136   0.816   0.4145    
z.diff.lag18  0.0088232  0.0338015   0.261   0.7941    
z.diff.lag19  0.0034682  0.0319866   0.108   0.9137    
z.diff.lag20  0.0107237  0.0299713   0.358   0.7205    
z.diff.lag21 -0.0101612  0.0276138  -0.368   0.7129    
z.diff.lag22 -0.0120969  0.0252195  -0.480   0.6315    
z.diff.lag23 -0.0368603  0.0224419  -1.642   0.1005    
z.diff.lag24 -0.0036444  0.0193879  -0.188   0.8509    
z.diff.lag25  0.0153814  0.0158763   0.969   0.3327    
z.diff.lag26 -0.0207530  0.0112110  -1.851   0.0642 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1129 on 7960 degrees of freedom
Multiple R-squared:  0.5116,    Adjusted R-squared:   0.51 
F-statistic: 308.8 on 27 and 7960 DF,  p-value: < 2.2e-16


Value of test-statistic is: -17.2482 

Critical values for test statistics: 
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Interpretation of the Unit Root Test for the Mexican Peso–US Dollar Exchange Rate

Key results: τ_1 = -17.2482

- **τ_1 (unit root):** Far below the 1% critical value (−2.58) ⇒ **reject** the null hypothesis of a unit root.
- **Deterministic components:** The model includes no deterministic terms (type = "none"), so the rejection indicates stationarity of the first differece.

Conclusion: At the 1% significance level, we **reject** the null hypothesis of a second unit root. The series is **integrated of order 1/strong>.**

In [61]:
```
library(urca)
summary(ur.df(inus, type='trend', lags=30, selectlags="AIC"))
```

```
################################################
# Augmented Dickey-Fuller Test Unit Root Test #
################################################

Test regression trend


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0609 -0.0409 -0.0023  0.0332  3.2165

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.105e-05  3.256e-03   0.010 0.992393
z.lag.1      -5.875e-04  3.003e-04  -1.956 0.050447 .
tt            4.316e-06  1.893e-06   2.279 0.022656 *
z.diff.lag1  -5.290e-02  8.699e-03  -6.082 1.22e-09 ***
z.diff.lag2  -3.078e-02  8.711e-03  -3.534 0.000411 ***
z.diff.lag3   1.464e-02  8.714e-03   1.680 0.093059 .
z.diff.lag4   2.877e-02  8.715e-03   3.301 0.000967 ***
z.diff.lag5   2.938e-02  8.720e-03   3.369 0.000756 ***
z.diff.lag6  -4.167e-02  8.722e-03  -4.777 1.80e-06 ***
z.diff.lag7   3.298e-02  8.730e-03   3.778 0.000159 ***
z.diff.lag8  -2.101e-03  8.735e-03  -0.240 0.809955
z.diff.lag9   2.814e-03  8.735e-03   0.322 0.747306
z.diff.lag10  4.163e-04  8.730e-03   0.048 0.961967
z.diff.lag11  1.736e-02  8.723e-03   1.990 0.046577 *
z.diff.lag12  6.978e-03  8.721e-03   0.800 0.423642
z.diff.lag13  8.467e-03  8.724e-03   0.970 0.331845
z.diff.lag14  1.446e-02  8.724e-03   1.658 0.097369 .
z.diff.lag15 -3.889e-03  8.721e-03  -0.446 0.655614
z.diff.lag16 -4.025e-02  8.708e-03  -4.622 3.84e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1806 on 13193 degrees of freedom
Multiple R-squared:  0.01141,	Adjusted R-squared:  0.01006
F-statistic:  8.46 on 18 and 13193 DF,  p-value: < 2.2e-16


Value of test-statistic is: -1.9563 7.4103 3.3556

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -3.96 -3.41 -3.12
phi2  6.09  4.68  4.03
phi3  8.27  6.25  5.34
```

Interpretation of the Unit Root Test for the Indian Rupee–US Dollar Exchange Rate

Key results: τ_3 = -1.9563; φ_2 = 7.4103; φ_3 = 3.3556

- **φ_2 (test for constant) and φ_3 (test for constant + trend):** φ_2 exceeds its 5% critical value (4.68), but φ_3 is below 6.25 ⇒ **cannot reject** that the deterministic trend is zero, though the constant may be relevant.

Conclusion: The test should be **rerun with only a constant** (set `type = "drift"`).

In [62]:
```
library(urca)
summary(ur.df(inus, type='drift', lags=30, selectlags="AIC"))
```

```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression drift


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0581 -0.0408 -0.0043  0.0331  3.2151 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   3.191e-03  2.947e-03   1.083 0.279015    
z.lag.1       8.058e-05  6.547e-05   1.231 0.218426    
z.diff.lag1  -5.317e-02  8.700e-03  -6.111 1.02e-09 ***
z.diff.lag2  -3.101e-02  8.712e-03  -3.559 0.000373 ***
z.diff.lag3   1.442e-02  8.715e-03   1.655 0.098041 .  
z.diff.lag4   2.854e-02  8.716e-03   3.275 0.001061 ** 
z.diff.lag5   2.915e-02  8.721e-03   3.342 0.000833 ***
z.diff.lag6  -4.192e-02  8.723e-03  -4.805 1.56e-06 ***
z.diff.lag7   3.275e-02  8.731e-03   3.751 0.000177 ***
z.diff.lag8  -2.349e-03  8.736e-03  -0.269 0.788037    
z.diff.lag9   2.568e-03  8.736e-03   0.294 0.768801    
z.diff.lag10  1.596e-04  8.731e-03   0.018 0.985417    
z.diff.lag11  1.712e-02  8.723e-03   1.962 0.049761 *  
z.diff.lag12  6.717e-03  8.722e-03   0.770 0.441225    
z.diff.lag13  8.185e-03  8.725e-03   0.938 0.348178    
z.diff.lag14  1.417e-02  8.724e-03   1.625 0.104245    
z.diff.lag15 -4.177e-03  8.721e-03  -0.479 0.631966    
z.diff.lag16 -4.052e-02  8.709e-03  -4.653 3.31e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1806 on 13194 degrees of freedom
Multiple R-squared:  0.01102,	Adjusted R-squared:  0.009747 
F-statistic: 8.649 on 17 and 13194 DF,  p-value: < 2.2e-16


Value of test-statistic is: 1.2308 8.5147 

Critical values for test statistics: 
      1pct  5pct 10pct
tau2 -3.43 -2.86 -2.57
phi1  6.43  4.59  3.78
```

Interpretation of the Unit Root Test for the Indian Rupee–US Dollar Exchange Rate

Key results: τ_2 = 1.2308; φ_1 = 8.5147

- **τ_2 (t-statistic for the drift model):** The value 1.2308 is much higher (less negative) than all critical values (−3.43, −2.86, −2.57) ⇒ **fail to reject** the null hypothesis of a unit root.
- **φ_1 (test for the constant term):** The value 8.5147 exceeds its 1% critical value (6.43) ⇒ **reject** the null hypothesis that the constant is zero.

Conclusion: At the 5% significance level, we **cannot reject** the presence of a unit root. The series is therefore **non-stationary with a constant drift and needs to be tested for the second unit root.**.

In [63]:

```
dinus = diff(inus)
dinus = na.omit(dinus)

library(urca)
summary(ur.df(dinus, type='drift', lags=30, selectlags="AIC"))
```

```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression drift


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0567 -0.0408 -0.0060  0.0326  3.2143 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)   0.006249   0.001586   3.939 8.21e-05 ***
z.lag.1      -1.016242   0.035531 -28.602  < 2e-16 ***
z.diff.lag1  -0.036740   0.034500  -1.065 0.286931    
z.diff.lag2  -0.067575   0.033449  -2.020 0.043380 *  
z.diff.lag3  -0.052961   0.032336  -1.638 0.101483    
z.diff.lag4  -0.024224   0.031160  -0.777 0.436933    
z.diff.lag5   0.005114   0.029920   0.171 0.864296    
z.diff.lag6  -0.036612   0.028576  -1.281 0.200137    
z.diff.lag7  -0.003668   0.027165  -0.135 0.892592    
z.diff.lag8  -0.005824   0.025667  -0.227 0.820499    
z.diff.lag9  -0.003062   0.024080  -0.127 0.898800    
z.diff.lag10 -0.002703   0.022264  -0.121 0.903373    
z.diff.lag11  0.014604   0.020449   0.714 0.475136    
z.diff.lag12  0.021518   0.018330   1.174 0.240464    
z.diff.lag13  0.029912   0.015783   1.895 0.058093 .  
z.diff.lag14  0.044298   0.012647   3.503 0.000462 ***
z.diff.lag15  0.040325   0.008708   4.631 3.68e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1806 on 13194 degrees of freedom
Multiple R-squared:  0.5303,    Adjusted R-squared:  0.5297 
F-statistic: 931.1 on 16 and 13194 DF,  p-value: < 2.2e-16


Value of test-statistic is: -28.6018 409.0337 

Critical values for test statistics: 
      1pct  5pct 10pct
tau2 -3.43 -2.86 -2.57
phi1  6.43  4.59  3.78
```

Interpretation of the Unit Root Test for the Indian Rupee–US Dollar Exchange Rate

Key results: τ_2 = -28.6018; φ_1 = 409.0337

- **τ_2 (t-statistic for the drift model):** Far below the 1% critical value (−3.43) ⇒ **reject** the null hypothesis of a unit root.
- **φ_1 (test for the constant term):** Much greater than its 1% critical value (6.43) ⇒ **reject** the null hypothesis that the constant is zero.

Conclusion: At the 1% significance level, both tests strongly **reject** the null hypothesis of a unit root and the absence of a constant term. The differenced series is therefore **stationary with a nonzero mean (drift)**.

Exercise 3

1. Explain what a subset ARIMA model is and what its advantages and disavdentages are compared to a standard ARIMA model.
2. Find the best subset ARIMA model for the Mexican Peso–US Dollar exchange rate. Consider up to 4 lags of AR and up to 4 lags of MA. Evaluate both the AIC and BIC information criteria.
3. How many models are investigated? How many models would have been investigated if we considered up to 5 lags?

Solution to Exercise 3

1. A subset ARIMA model is a version of the ARIMA model in which only selected autoregressive or moving average lags are included, rather than all lags up to a given order. This approach makes the model more parsimonious and focuses on the lags that significantly improve forecasting accuracy. The main drawback is that identifying the optimal subset ARIMA model can be computationally intensive.
2. As shown below, the **AIC** suggests a model with the first four AR components and the second and fourth MA components. In contrast, the **BIC** suggests a model with the first, third, and fourth AR components and the first and fourth MA components.

3. With 4 lags, we consider a total of **256** models. If we were to consider 5 lags, we would evaluate a total of **1,024** models.

In [64]:
```
data = merge(dmxus, dinus, join = "inner")
colnames(data) = c("dmxus", "dinus")
```

In [65]:
```
################################################
## Exhaustive ARIMA Subset Search with AIC/BIC
################################################
p = 4                      # number of AR coefficients considered (AR(1) .. AR(4))
q = 4                      # number of MA coefficients considered (MA(1) .. MA(4))
pq = p + q                 # total number of non-seasonal AR and MA parameters

# Build a data frame of all 2^(p+q) combinations of {0, NA} for AR1..AR4 and MA1..MA4.
# Each row is a pattern telling arima() which coefficients are fixed (0) vs free (NA).
indvar = do.call(expand.grid, replicate(pq, c(0, NA), simplify = FALSE))

## Give column names to aid interpretation (AR first, then MA)
colnames(indvar) = c("ar1","ar2","ar3","ar4","ma1","ma2","ma3","ma4")

res_list = list()          # container to store per-model results (AIC/BIC and names)

iter = 0  # Counter for results (row counter in res_list)

# Loop over every 0/NA pattern (i.e., every model specification)
for (i in 1:nrow(indvar)) {
        iter = iter + 1
        # Identify which AR/MA parameters are FREE under this pattern:
        # NA entries indicate parameters to estimate.
        active_idx  = which(is.na(indvar[i, ]))  # indices of NA among ar*/ma* slots

        # Map those indices for reporting.
        active_names = colnames(indvar)[active_idx]

        # Fit ARIMA(4,0,4) with this fixed pattern.
        # 'fixed' must contain AR1..AR4, MA1..MA4, and then the mean.
        # Appending NA at the end frees the intercept/mean in every spec.
        model = try(suppressWarnings(arima(data[1:nrow(data), 1], order = c(p,0,q), fixed =c(indvar[i,], NA))), silent = TRUE)

        # If estimation fails (e.g., non-invertible MA, singular information matrix),
        # skip this specification and continue to the next pattern.
        if (inherits(model, "try-error")) next

        # Save a compact summary row:
        # - 'regressors' lists the FREE (estimated) AR/MA parameter names for this pattern
        # - AIC/BIC are the information criteria for model comparison
        res_list[[iter]] = data.frame(
         regressors = paste(active_names, collapse = " + "),
         AIC        = AIC(model),
         BIC        = BIC(model)
        )
}

## ---- 8) Bind results into a single data frame -----------------------------
# Stack all rows from res_list into a single data frame
results = do.call(rbind, res_list)

## Quick look at the first and last few rows of the results table
#head(results)
#tail(results)

## ---- 9) Identify the best models by AIC and BIC ---------------------------
# Find the row indices of the minimum AIC and BIC
best_by_AIC = results[which.min(results$AIC), ]
best_by_BIC = results[which.min(results$BIC), ]

## Display the best specs under each criterion
best_by_AIC
best_by_BIC
```

A data.frame: 1 × 3

	regressors	AIC	BIC
	<chr>	<dbl>	<dbl>
176	ar1 + ar2 + ar3 + ar4 + ma2 + ma4	-12206.27	-12150.36

A data.frame: 1 × 3

	regressors	AIC	BIC
	<chr>	<dbl>	<dbl>
158	ar1 + ar3 + ar4 + ma1 + ma4	-12205.2	-12156.27

Exercise 4

1. Find the best subset ADL model for the Mexican Peso–US Dollar exchange rate. Consider up to four lags of the AR term and up to four lags of the Indian Rupee–US Dollar exchange rate. Evaluate both the AIC and BIC information criteria.

2. Does your result suggest that the Indian Rupee–US Dollar exchange rate is Granger-causal for the Mexican Peso–US Dollar exchange rate? Explain why.

Solution to Exercise 4

1. According to the **AIC** information criterion, the best model includes the first lag of the Mexican Peso–US Dollar exchange rate and the first two lags of the Indian Rupee–US Dollar exchange rate. According to the **BIC** information criterion, the best model includes only the first lag of the Indian Rupee–US Dollar exchange rate.

2. These results suggest that the Indian Rupee–US Dollar exchange rate is **Granger-causal** for the Mexican Peso–US Dollar exchange rate.

In [66]:
```
##############################################
## Exhaustive ADL Subset Search with AIC/BIC
##############################################

## ---- 0) Libraries ------------------------------------------------------
## 'gets' provides the arx() function for regression with optional diagnostics.
library(gets)

## ---- 1) Create lagged variables ----------------------------------------
## Assume 'data' is a bivariate ts object:
##   - column 1: y (e.g., GDP growth)
##   - column 2: x (e.g., ΔHours worked)
## We build up to 4 lags for each series.

y0 = data[, 1]                # y_t (dependent variable at time t)

y1 = lag(data[, 1], -1)       # y_{t-1}
y2 = lag(data[, 1], -2)       # y_{t-2}
y3 = lag(data[, 1], -3)       # y_{t-3}
y4 = lag(data[, 1], -4)       # y_{t-4}

x1 = lag(data[, 2], -1)       # x_{t-1}
x2 = lag(data[, 2], -2)       # x_{t-2}
x3 = lag(data[, 2], -3)       # x_{t-3}
x4 = lag(data[, 2], -4)       # x_{t-4}

## ---- 2) Align sample (common time span) ---------------------------------
## ts.intersect keeps only the time points where all included series are observed.
## This ensures comparable samples across all model specifications.
tempdata = merge(
  y0,
  y1, y2, y3, y4,
  x1, x2, x3, x4, join = "inner"
)

## ---- 3) Define dependent and regressor matrices --------------------------
## depvar: y_t aligned to the intersection sample.
## indvar: matrix of candidate regressors (lags of y and x).
depvar = tempdata[, 1]
indvar = tempdata[, 2:ncol(tempdata)]
## Give column names to aid interpretation
colnames(indvar) = c("y1","y2","y3","y4","x1","x2","x3","x4")

## ---- 4) Initialize containers for results -------------------------------
## We'll accumulate rows (one per model) in a list, then rbind at the end.
res_list = list()

iter = 1
model = arx(depvar, mc = TRUE)
res_list[[iter]] = data.frame(
      regressors = paste(""),
      AIC        = AIC(model),
      BIC        = BIC(model)
    )

## ---- 5) Define the model space -----------------------------------------
## We will try ALL non-empty subsets of the candidate regressors:
## sizes i = 1, 2, ..., kmax.
## WARNING: This is 2^kmax - 1 models (grows quickly with kmax).
kmax = ncol(indvar)   # total number of available regressors
k = 1:kmax            # index set for regressors

## ---- 6) Exhaustive subset loop -----------------------------------------
## Outer loop over subset size i
for (i in 1:kmax) {

  ## All combinations (columns) of size i from the set {1, ..., kmax}.
  ## combn returns a matrix with i rows and choose(kmax, i) columns.
  ksets = combn(k, i)

  ## Inner loop over each specific combination of regressors of size i
```

```r
  for (j in 1:ncol(ksets)) {
    iter = iter + 1

    ## Indices of regressors to include in this model
    regressors = ksets[, j]

    ## Fit the ADL variant via arx():
    ## - depvar is the dependent variable (y_t)
    ## - mxreg supplies *exogenous* regressors (here: lags of y and x treated as "xreg")
    ## - mc = TRUE includes an intercept
    model = try(suppressWarnings(arx(depvar, mxreg = indvar[, regressors], mc = TRUE)), silent = TRUE)

    # If estimation fails
    # skip this specification and continue to the next pattern.
    if (inherits(model, "try-error")) next

    ## ---- 7) Store information criteria and regressor labels ----------------
    res_list[[iter]] = data.frame(
      regressors = paste(colnames(indvar)[regressors], collapse = " + "),
      AIC        = AIC(model),
      BIC        = BIC(model)
    )
  }
}

## ---- 8) Bind results into a single data frame ----------------------------
results = do.call(rbind, res_list)

## Quick look at the first and last few rows of the results table
#head(results)
#tail(results)

## ---- 9) Identify the best models by AIC and BIC --------------------------
best_by_AIC = results[which.min(results$AIC), ]
best_by_BIC = results[which.min(results$BIC), ]

## Display the best specs under each criterion
best_by_AIC
best_by_BIC
```

A data.frame: 1 × 3

	regressors	AIC	BIC
	<chr>	<dbl>	<dbl>
53	y1 + x1 + x2	-12173.79	-12145.84

A data.frame: 1 × 3

	regressors	AIC	BIC
	<chr>	<dbl>	<dbl>
6	x1	-12170.54	-12156.57

Exercise 5

1. Which model — the subset ARIMA or the ADL — forecasts the Mexican Peso–US Dollar exchange rate more accurately in one-day-ahead predictions? To answer this question, focus on the subset ARIMA and ADL models suggested by the BIC criterion and conduct a pseudo–out-of-sample forecasting exercise. Use the RMSFE (Root Mean Squared Forecast Error) as the evaluation metric.

2. Based on the model that performs better in the one-day-ahead forecast, produce one-step ahead forecast value and its 95% confidence interval for 11-04-2025.

3. Explain the meaning of the forecast confidence interval.

Solution to Exercise 5

1. According to the **BIC**, the optimal subset ARIMA model is:

$$y_t = \alpha_0 y_{t-1} + \alpha_1 y_{t-3} + \alpha_2 y_{t-4} + \alpha_3 e_{t-1} + \alpha_4 e_{t-4} + e_t$$

   The **RMSFE** over the 25 most recent periods is equal to **0.05761**.

   According to the **BIC**, the optimal ADL model is:

$$y_t = \alpha_0 + \alpha_1 x_{t-1} + e_t$$

The **RMSFE** over the 25 most recent periods is equal to **0.05685**.

Therefore, the ADL model outperforms the subset-ARIMA in pseudo–out-of-sample one-step-ahead forecasting.

2. The one-step-ahead forecast for **2025-11-03** is **0.00158**, which implies an increase of about 0.002 Mexican pesos per 1 US dollar. The 95% confidence interval is **[−0.221, 0.224]**, covering both a possible depreciation of 0.22 pesos and an appreciation of 0.22 pesos.

3. The 95% confidence interval means that if the forecast for 2025-11-03 were repeated 100 times, in about 95 of those cases the actual change in the exchange rate would fall within the interval between −0.23 and 0.22 pesos. In a multiverse interpretation, if November 3, 2025, were to occur 100 times under the same model and economic conditions but with different random shocks, we would expect the actual change in the exchange rate to fall within this 95% forecast interval in about 95 of those universes.

In [67]:
```
# https://www.rdocumentation.org/packages/forecast/versions/8.16/topics/forecast
library(forecast)

# Length of sample
T = nrow(data)

# Container for results
results = data.frame()

# Initial estimation window
Tstar = 7992

# Number of iterations
Tmax = T - Tstar

cat("Our data has", T, "observations.\n")
cat("We will keep the first", Tstar, "observations fixed for stability (initial estimation window).\n")
cat("From there, we add one observation at a time and forecast one step ahead at each iteration.\n")
cat("We will produce", Tmax, "out-of-sample 1-step-ahead forecasts, ending at observation", T, ".\n\n")


for (i in 1:Tmax) {
  # Expand window estimation: use data up to time Tstar + i - 1
  model = arima(data[1:(Tstar + i - 1), 1], order = c(4, 0, 4), fixed = c(NA, 0, NA, NA, NA, 0, 0, NA, NA))
  # One-step-ahead forecast with a single 95% PI
  f = forecast::forecast(model, h = 1, level = 95)
  # Extract components (coerce to numeric to avoid ts attributes in arithmetic)
  f.mean = as.numeric(f$mean[1])              # point forecast y_{T+1|T}
  f.lower95 = as.numeric(f$lower[1])          # 95% lower bound
  f.upper95 = as.numeric(f$upper[1])          # 95% upper bound
  obs.val = as.numeric(dmxus[Tstar + i])       # actual y_{T+1}

  # Absolute forecast error
  f.error = abs(f.mean - obs.val)

  # Row to append
  temp = c(f.mean, f.lower95, f.upper95, obs.val, f.error)
  results = rbind(results, temp)
}

# Name the columns
names(results) = c("Mean forecast", "Lower bound 95%", "Upper bound 95%", "Actual value", "Absolute error")

# Quick peek at the first 6 interations
#head(results)

# Quick peek at the last 6 iterations
#tail(results)

# Save ARMA rolling results
results.ARMA = results

# Root Mean Squared Forecast Error (RMSFE)
rmsfe.ARMA = sqrt(mean((results.ARMA[, "Mean forecast"] - results.ARMA[, "Actual value"])^2))

cat("The RMSFE is", round(rmsfe.ARMA, 5))
```

```
Our data has 8017 observations.
We will keep the first 7992 observations fixed for stability (initial estimation window).
From there, we add one observation at a time and forecast one step ahead at each iteration.
We will produce 25 out-of-sample 1-step-ahead forecasts, ending at observation 8017 .

The RMSFE is 0.05761
```

In [68]:
```
library(dynlm)

# Length of sample
T = nrow(data)

# Container for results
results = data.frame()

# Initial estimation window
```

```r
Tstar = 7992

# Number of iterations
Tmax = T - Tstar

cat("Our data has", T, "observations.\n")
cat("We will keep the first", Tstar, "observations fixed for stability (initial estimation window).\n")
cat("From there, we add one observation at a time and forecast one step ahead at each iteration.\n")
cat("We will produce", Tmax, "out-of-sample 1-step-ahead forecasts, ending at observation", T, ".\n\n")

for (i in 1: (T - Tstar)) {
    model = dynlm(dmxus ~  L(dinus, 1), as.zoo(data[1: (Tstar + i - 1), ]))
    newdata = data.frame("dinus" = data[(Tstar + i - 1), "dinus"])

    # One-step-ahead forecast with a single 95% PI
    f = predict(model, newdata = newdata, interval = "prediction", level = 0.95)

    # Extract components (coerce to numeric to avoid ts attributes in arithmetic)
    f.mean = as.numeric(f[1,1])
    f.lower95 = as.numeric(f[1,2])
    f.upper95 = as.numeric(f[1,3])

    # Realized value at time (Tstar + i) to compare against the forecast
    obs.val = as.numeric(data[(Tstar + i), "dmxus"])

    # Absolute forecast error
    f.error = abs(f.mean - obs.val)

    # Row to append
    temp = c(f.mean, f.lower95, f.upper95, obs.val, f.error)
    results = rbind(results, temp)
}
names(results)=c("Mean forecast", "Lower bound 95%", "Upper bound 95%", "Actual value", "Absolute error")

# Quick peek at the first 6 iterations
#head(results)

# Quick peek at the last 6 iterations
#tail(results)

# Save ADL rolling results
results.ADL = results

# Root Mean Squared Forecast Error (RMSFE)
rmsfe.ADL = sqrt(mean((results.ADL[, "Mean forecast"] - results.ADL[, "Actual value"])^2))

cat("The RMSFE is", round(rmsfe.ADL, 5))
```

```
Our data has 8017 observations.
We will keep the first 7992 observations fixed for stability (initial estimation window).
From there, we add one observation at a time and forecast one step ahead at each iteration.
We will produce 25 out-of-sample 1-step-ahead forecasts, ending at observation 8017 .

The RMSFE is 0.05685
```

In [45]:
```r
temp = dynlm(dmxus ~  L(dinus, 1), as.zoo(data))
predict(temp, newdata = data.frame("dinus" = data[nrow(data), "dinus"]), interval = "prediction", level = 0.95)
```

A matrix: 1 × 3 of type dbl

	fit	lwr	upr
2025-10-31	0.00158491	-0.2205728	0.2237426

Exercise 6

1. Use a reduced-form VAR model with the number of lags suggested by the BIC to produce a one-step-ahead forecast of the Mexican Peso–US Dollar exchange rate.

2. Compare this forecast to the optimal univariate forecast you found earlier. How do the two forecasts differ?

3. Describe how you would design an exercise to determine which model performs better.

Solution to Exercise 6

1. The **BIC** information criterion suggests a VAR(1) model, i.e., a VAR model with one lag. The one-step-ahead forecast from the VAR(1) model is **0.00168** with a 95% confidence interval of **[−0.2204, 0.2238]**. The model predicts that the exchange rate will increase by approximately **0.002 pesos**.

2. The forecasts are very similar because both the ADL and the VAR(1) models capture essentially the same dynamics. The information criteria selected a VAR with only one lag, meaning that each variable in the system depends primarily on its own lag and the lag of the other variable—exactly what the ADL structure already models. In other words, the ADL for the peso already incorporates the relevant

information from the other currency's lag, so moving to a VAR(1) adds little new information. As a result, the forecasts from the ADL and VAR(1) models are nearly identical

3. Based on a single forecast, we cannot determine which model produces more reliable predictions over a longer period of time. We could design a loop to generate **pseudo–out-of-sample forecasts**, similar to the procedure used earlier for the ARIMA and ADL models, and then compare their forecast accuracy (e.g., using RMSFE).

In [36]:
```
library(vars)
# Selects optimal VAR lag (up to 8) based on information criteria.

out = VARselect(data, lag.max = 8, type = "const")
out$selection
```

AIC(n): 8 **HQ(n):** 7 **SC(n):** 1 **FPE(n):** 8

In [37]:
```
VAR1 = VAR(data, p = 1, type = "const")

# Generate forecasts from the VAR model
prd = predict(
  VAR1,   # the estimated VAR object
  n.ahead = 1, # number of steps ahead to forecast (here: 4 periods)
  ci = 0.95    # confidence interval level (95%)
)

# Display the forecast results
print(prd)
```

```
$dmxus
                  fcst      lower     upper        CI
dmxus.fcst 0.001688055 -0.2204265 0.2238026 0.2221146

$dinus
                   fcst      lower     upper        CI
dinus.fcst 0.0009491483 -0.4309164 0.4328147 0.4318656
```

Exercise 7

1. Based on what we have learned so far about the Mexican Peso–US Dollar exchange rates, can they be cointegrated

2. Use the Phillips–Ouliaris test to determine whether there is a cointegrating relationship between the two exchange rates.

Note: We will not use the Engle–Granger test because with very large samples (like ours), the ADF test on residuals can become **too sensitive**. It may incorrectly suggest cointegration when there is only a tiny amount of mean reversion that is not economically meaningful. The Phillips–Ouliaris test is **more robust** to this issue, as it adjusts for serial correlation and long-run variance nonparametrically, making it more reliable in large datasets.

The ADF test basically says — if it finds a *unit root*, then the series most likely **has a unit root**. But if it says there is *no unit root* and the sample is very large, this result can be misleading due to a **power or sensitivity issue**: the test might detect tiny deviations from a unit root that are statistically significant but not meaningful in practice.

Solution to Exercise 7

1. Yes, they can be, since both exchange rates are integrated of the same order. Because both series are I(1), it is appropriate to test for a possible cointegrating relationship between them.

2. The **Phillips–Ouliaris test**, which tests the null hypothesis of *no cointegration*, indicates that there is **no cointegrating relationship** between the two exchange rates.

In [38]:
```
# Phillips--Ouliaris Cointegration Test
library(urca)
# summary(ca.po(data, demean = c("none", "constant", "trend"),
#       lag = c("short", "long"), type = c("Pu", "Pz"), tol = NULL))
# lag = "long" — series exhibit strong autocorrelation
# lag = "short" — series exhibit weak autocorrelation

rowdata = merge(mxus, inus, join = "inner")
colnames(rowdata) = c("mxus", "inus")

summary(ca.po(rowdata, lag = "long", demean = "trend", type = "Pz"))
```

```
########################################
# Phillips and Ouliaris Unit Root Test #
########################################

Test of type Pz
detrending of series with constant and linear trend

Response mxus :

Call:
lm(formula = mxus ~ zr + trd)

Residuals:
     Min       1Q   Median       3Q      Max
-1.25030 -0.04475 -0.00592  0.03703  1.34303

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept) 1.474e-02  7.825e-03    1.883   0.0597 .
zrmxus      9.974e-01  8.265e-04 1206.799   <2e-16 ***
zrinus      1.129e-04  2.538e-04    0.445   0.6564
trd         3.860e-06  1.989e-06    1.941   0.0523 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1133 on 8013 degrees of freedom
Multiple R-squared:  0.9994,    Adjusted R-squared:  0.9994
F-statistic: 4.672e+06 on 3 and 8013 DF,  p-value: < 2.2e-16


Response inus :

Call:
lm(formula = inus ~ zr + trd)

Residuals:
     Min       1Q   Median       3Q      Max
-2.25605 -0.07627 -0.00371  0.06874  2.55253

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)
(Intercept)  2.886e-02  1.526e-02    1.891   0.0586 .
zrmxus      -1.518e-03  1.612e-03   -0.942   0.3462
zrinus       9.994e-01  4.948e-04 2019.875   <2e-16 ***
trd          7.845e-06  3.878e-06    2.023   0.0431 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2208 on 8013 degrees of freedom
Multiple R-squared:  0.9998,    Adjusted R-squared:  0.9998
F-statistic: 1.35e+07 on 3 and 8013 DF,  p-value: < 2.2e-16



Value of test-statistic is: 24.007

Critical values of Pz are:
                  10pct    5pct     1pct
critical values 71.9586 81.3812 102.0167
```

Exercise 8

1. Test for the ARCH effect in the Mexican Peso-US Dollar exchange rate.

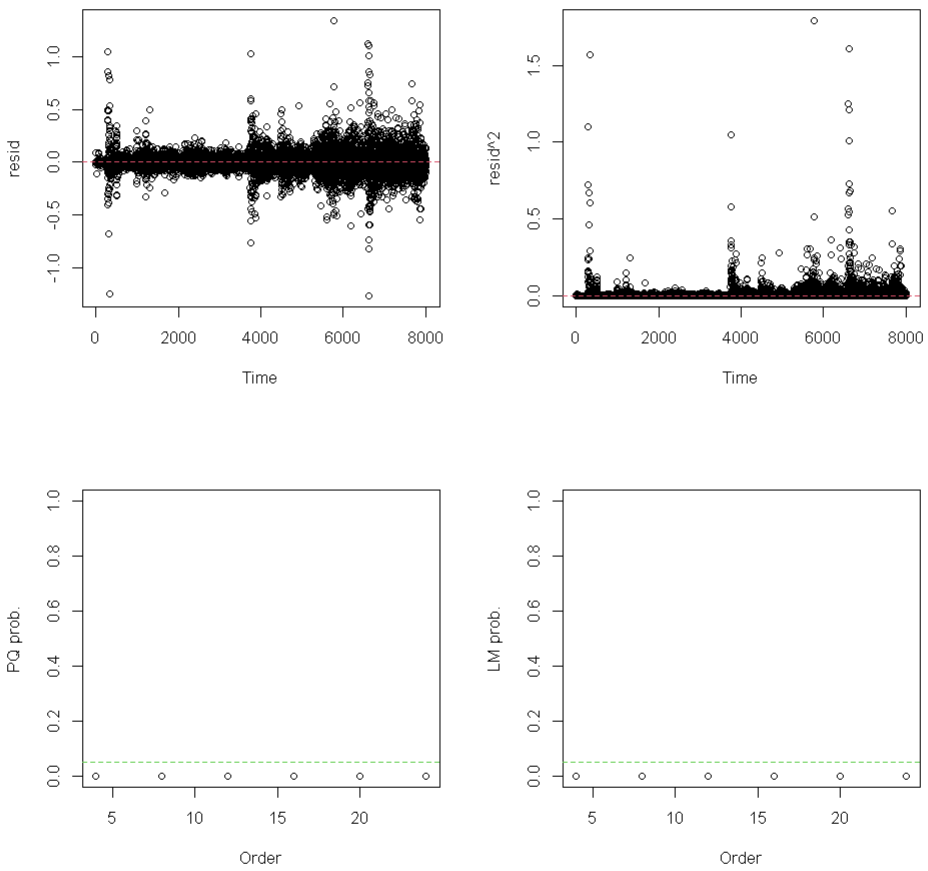
Solution to Exercise 8

In [47]:
```
model.dmxus = auto.arima(dmxus, max.p = 20, max.q = 20)
model.dmxus
```

```
Series: dmxus
ARIMA(0,0,0) with non-zero mean

Coefficients:
        mean
      0.0019
s.e.  0.0013

sigma^2 = 0.01284:  log likelihood = 6083.81
AIC=-12163.63   AICc=-12163.63   BIC=-12149.65
```

In [50]:
```
library(aTSA)
est = arima(dmxus, order=c(0,0,0))
arch.test(est)
```

```
ARCH heteroscedasticity test for residuals
alternative: heteroscedastic

Portmanteau-Q test:
     order   PQ p.value
[1,]     4 3266       0
[2,]     8 4459       0
[3,]    12 5357       0
[4,]    16 6016       0
[5,]    20 6274       0
[6,]    24 6761       0
Lagrange-Multiplier test:
     order   LM p.value
[1,]     4 5863       0
[2,]     8 2780       0
[3,]    12 1781       0
[4,]    16 1310       0
[5,]    20 1043       0
[6,]    24  849       0
```



Interpretation of the ARCH-LM Test

Procedure: The ARCH-LM test evaluates whether the residuals from a mean equation (in this case, an ARIMA(0,0,0) with mean) exhibit conditional heteroskedasticity—that is, whether their variance depends on past squared residuals. The test was performed using the `arch.test()` function from the **aTSA** package, which reports both a Portmanteau-Q test and Engle's Lagrange Multiplier (LM) test across different lag orders. The null hypothesis is that there are no ARCH effects (homoskedastic residuals).

Results: The output shows extremely small p-values (reported as 0 due to rounding) for both the Portmanteau-Q and LM statistics across all lag orders. This strongly rejects the null hypothesis, indicating the presence of ARCH effects in the residuals. In other words, the variance of `dmxus` is not constant over time, and periods of high and low volatility tend to cluster.

Interpretation: The presence of ARCH effects means that while the ARIMA model adequately captures the mean dynamics of `dmxus`, it fails to model the time-varying volatility. This finding suggests that extending the model to include a conditional variance component, such as a GARCH(1,1) specification, would provide a better representation of the data. The same conclusion would be supported visually if the ACF of squared residuals shows slow decay or significant autocorrelation at multiple lags—both typical signs of volatility clustering.

Conclusion: The ARCH-LM results confirm strong evidence of conditional heteroskedasticity. Therefore, modeling the volatility explicitly (for example, with a GARCH-type model) is necessary to improve forecast accuracy and obtain valid inference regarding uncertainty and confidence intervals.

In [ ]: