Applied Time Series Analysis using Stata

Kedar Kulkarni

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This document provides an applied introduction to core topics in time series econometrics, beginning with the foundational ideas of statistical independence and identical distribution (i.i.d.) and showing, through simulations in Stata, how real-world economic data often violate these assumptions. We then move to time series decomposition, illustrating how trend, seasonality, and cyclical movements can be separated and understood before any formal modeling. Building on these components, the notes introduce ARIMA models, highlighting how autoregressive and moving-average dynamics help capture persistence and noise in economic series.

The material then addresses stationarity tests, explaining why stable statistical properties are essential for valid inference and forecasting, and how common tests diagnose unit roots. Next, we examine structural breaks, showing how shifts in underlying data-generating processes can distort estimates and alter long-run behavior. The final chapters introduce vector autoregressions (VARs) and vector error correction models (VECMs), which allow students to study multivariate dynamics, feedback relationships, and long-run equilibria between economic variables.

Overall, this is an applied resource built around Stata code, simulations, and real-world datasets to help students learn by doing. For deeper theoretical treatment, readers should consult established time series textbooks such as Levendis (2018) and Enders (2009).

Chapter 2

Independent and Identical Distributions and Its Violations

Introduction

This chapter walks through a set of Stata commands used to simulate and visualize the difference between data that is independent and identically distributed (i.i.d.), and data that violates one or both of these assumptions.

2.1 Simulating i.i.d. Data

```
clear
set obs 100
gen id = _n

// Identically distributed data: normal(0,1)
gen y_identical = rnormal(0,1)
```

Listing 2.1: Generating i.i.d. Data

We first create 100 observations and a simple ID variable. The variable *y-identical* is drawn from a standard normal distribution, and each observation is independent and identically distributed.

2.2 Simulating Non-Identically Distributed Data

```
gen y_nonidentical = .
replace y_nonidentical = rnormal(0,1) if id <= 50
replace y_nonidentical = rnormal(5,2) if id > 50
```

Listing 2.2: Generating Non-Identically Distributed Data

Here, we break the identical distribution assumption. The first 50 observations are drawn from $N(0, 1)$, and the next 50 from $N(5, 2)$. The variance and mean change halfway through the dataset.

2.3 Simulating Identically Distributed but Non-Independent Data

```
gen y_notindependent=.
gen epsilon = rnormal(0,1)
replace y_notindependent = epsilon[1] in 1
forvalues i = 2/100 {
replace epsilon = rnormal(0,1) in i'
    replace y_notindependent = 0.7 * y_notindependent[=i'-1'] + epsilon[i'] in `i'
}
```

Listing 2.3: Generating AR(1) Process

This code creates an autoregressive process (AR(1)) where each value of `y` depends on the previous value, introducing autocorrelation (i.e., lack of independence) but keeping the error term identically distributed.

2.4 Plotting the Data

```
// Plotting Independent and Identical Distribution //
twoway (line y_identical id, lcolor(blue) mcolor(blue) ///
        msymbol(o) lpattern(solid)), ///
        yline(0, lpattern(dash) lcolor(gs10)) ///
        title("Identically Distributed Data (N(0,1))") ///
        legend(off)

// Plotting Independent but Non Identical Distribution //
twoway (line y_nonidentical id, lcolor(red) mcolor(red) ///
        msymbol(o) lpattern(solid)), ///
        yline(0 5, lpattern(dash) lcolor(gs10)) ///
        title("Not Identically Distributed Data (N(0,1) then N(5,2))") ///
        legend(off)

// Plotting identically distributed but non independent Distribution //
twoway (line y_notindependent id, lcolor(blue)), ///
       title("Identically Distributed but Not Independent") ///
       ytitle("y (AR(1))") xtitle("Time") ///
       legend(off)
```

Listing 2.4: Plotting the Three Series

Each plot helps visually distinguish how the assumptions affect the data: flat random noise, level shifts, and smooth autocorrelated data.

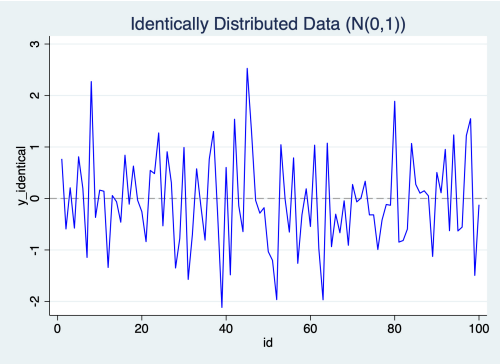

Figure 2.1: Identically Distributed Data

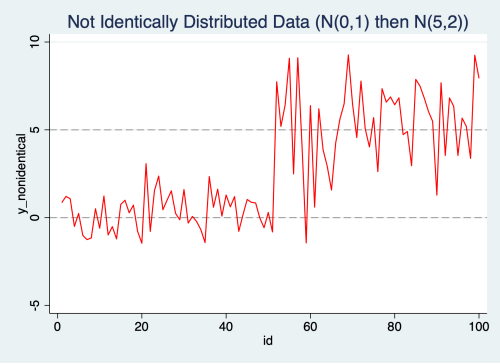

Figure 2.2: Not Identically Distributed but Independent Data

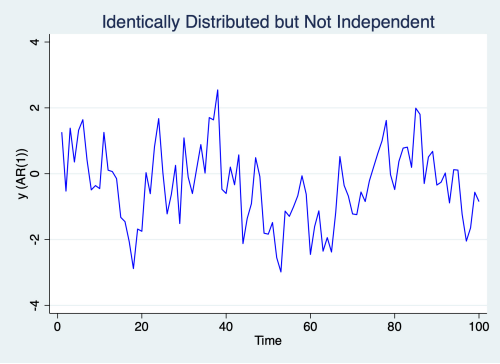

Figure 2.3: Identically Distributed but Not Independent Data

2.5 Non-Independent and Non-Identically Distributed Data

```
clear
set obs 100
gen id = _n
gen trend = id / 10
gen sigma = 0.5 + id / 100

gen y = .
gen epsilon = rnormal(0, sigma)
replace y = epsilon[1] + trend[1] in 1
forvalues i = 2/100 {
replace epsilon = rnormal(0, sigma[i']) in i'
replace y = 0.8 * y[=i'-1'] + trend[i'] + epsilon[i'] in `i'
}
```

Listing 2.5: Combining Time Trend, Changing Variance, and AR(1)

This more complex simulation includes three components: (1) a time trend, (2) increasing variance over time (heteroskedasticity), and (3) serial correlation (via AR(1) structure). This violates both independence and identical distribution.

```
// Generate non independent and non identically distributed //
twoway (line y id, lcolor(navy) msymbol(circle)), ///
       title("Non-Independent and Non-Identically Distributed Data") ///
       ytitle("Simulated y") xtitle("Time") ///
       legend(off)
```

Listing 2.6: Plotting the Final Series

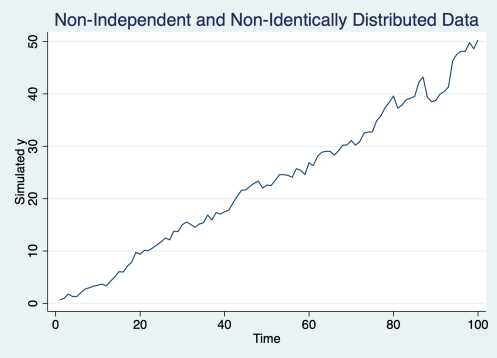

Figure 2.4: Non-Independent and Non-Identically Distributed Data

2.6 Conclusion

These simulations demonstrate the importance of the assumptions behind many statistical techniques. i.i.d. assumptions often do not hold in time series data, and this has consequences for estimation and inference.

Chapter 3

Decomposing Time Series

This Chapter walks through a Stata program that constructs a synthetic time series consisting of a linear trend, sinusoidal seasonality, and AR(1) irregular component. It then forecasts 10 future periods and plots the observed vs forecasted values.

3.1 Simulating Trend, Heteroskedasticity, and AR(1)

```
clear all
set obs 120
gen t = _n
```

Listing 3.1: Generate Observations

We begin by generating 120 time periods and assigning a time index $t = 1, \ldots, 120$.

```
gen T = 1 + 0.1 * t
gen S = 1.6 * sin(_pi * t / 6)
```

Listing 3.2: Generate Trend and Seasonal Data

The trend grows linearly. The seasonal component has a periodic structure with a 12-month cycle (since $\pi t/6$ gives a sine period of 2π every 12 months).

```
gen epsilon = rnormal(0,1)
gen I = .
replace I = epsilon in 1
forvalues i = 2/120 {
    replace I = 0.7 * I[_n-1] + epsilon in `i'
}
```

Listing 3.3: Generate AR(1) Component

We simulate an AR(1) process where each value depends on its lag with coefficient 0.7 plus a white noise error.

```
gen X = T + S + I
```

Listing 3.4: Generate Final Series

The final observed series X combines trend, seasonality, and irregular components.

3.2 Forecast Future Periods (t = 121 to 130)

```
set obs 130
replace t = _n
```

Listing 3.5: Extend Dataset

We add 10 additional time periods.

```
replace T = 1 + 0.1 * t if missing(T)
replace S = 1.6 * sin(_pi * t / 6) if missing(S)
```

Listing 3.6: Forecast Trend and Seasonality

We continue the deterministic trend and seasonal components using the same functional forms.

```
gen epsilon_f = rnormal(0,1)
replace I = . if missing(I)
replace I = 0.7 * I[_n-1] + epsilon_f in 121
replace I = 0.7 * I[_n-1] + epsilon_f in 122
...
replace I = 0.7 * I[_n-1] + epsilon_f in 130
```

Listing 3.7: Forecast AR(1) Component

We generate forecasts for the irregular component using the same AR(1) structure, re-using a single random error *epsilonf* (a simplification for illustration).

```
replace X = T + S + I if missing(X)
```

Listing 3.8: Construct Forecasted X

3.3 Time Index and Plotting

```
gen time = tm(2015m1) + _n - 1
format time %tm
tsset time, monthly
```

Listing 3.9: Define Time Series

We define a proper monthly time series index starting from January 2015.

```
gen type = "Observed"
replace type = "Forecasted" if t > 120
```

Listing 3.10: Observed vs Forecasted

```
tsline X, title("Time Series with 10-Step Forecast") ///
       ytitle("X") xtitle("Time") ///
       legend(off) lcolor(blue) ///
       note("Forecast from t = 121 to 130")
```

Listing 3.11: Plot Full Series with Forecast

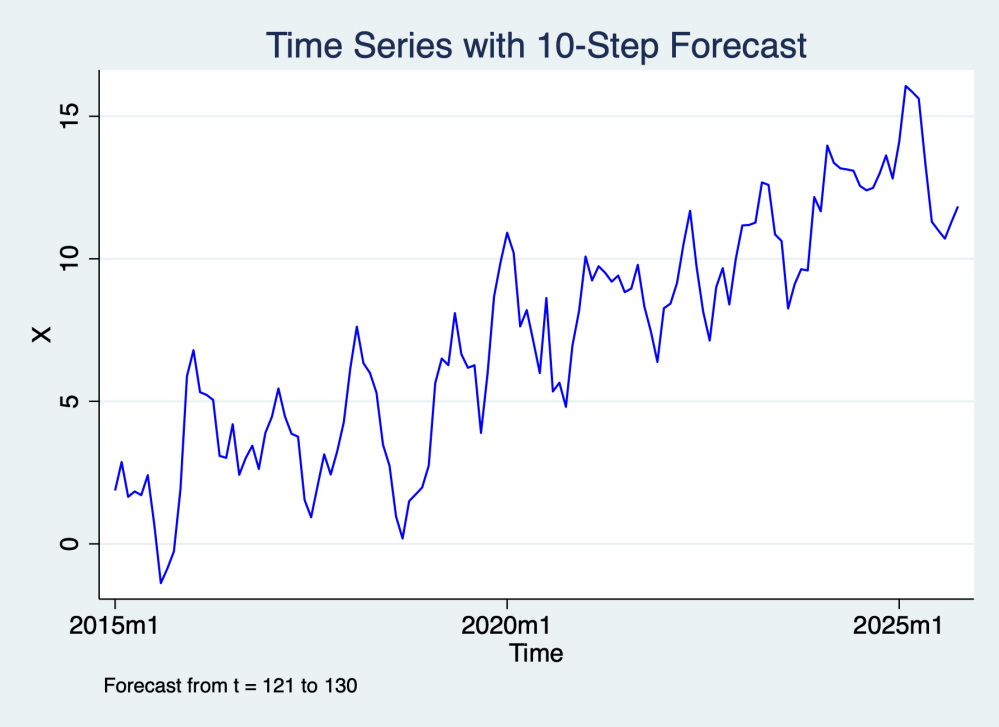

Figure 3.1: Time Series with 10-Step Forecast

```
twoway ///
    (line X time if type == "Observed", lcolor(black) lpattern(solid)) ///
    (line X time if type == "Forecasted", lcolor(red) lpattern(dash)), ///
    title("Observed vs Forecasted Values") ///
    ytitle("X") xtitle("Time") ///
    legend(order(1 "Observed" 2 "Forecasted")) ///
    xline(780, lpattern(dot) lcolor(gs8)) ///
    note("Forecasted values shown in red (t = 121 to 130)")
```

Listing 3.12: Overlay Observed vs Forecast

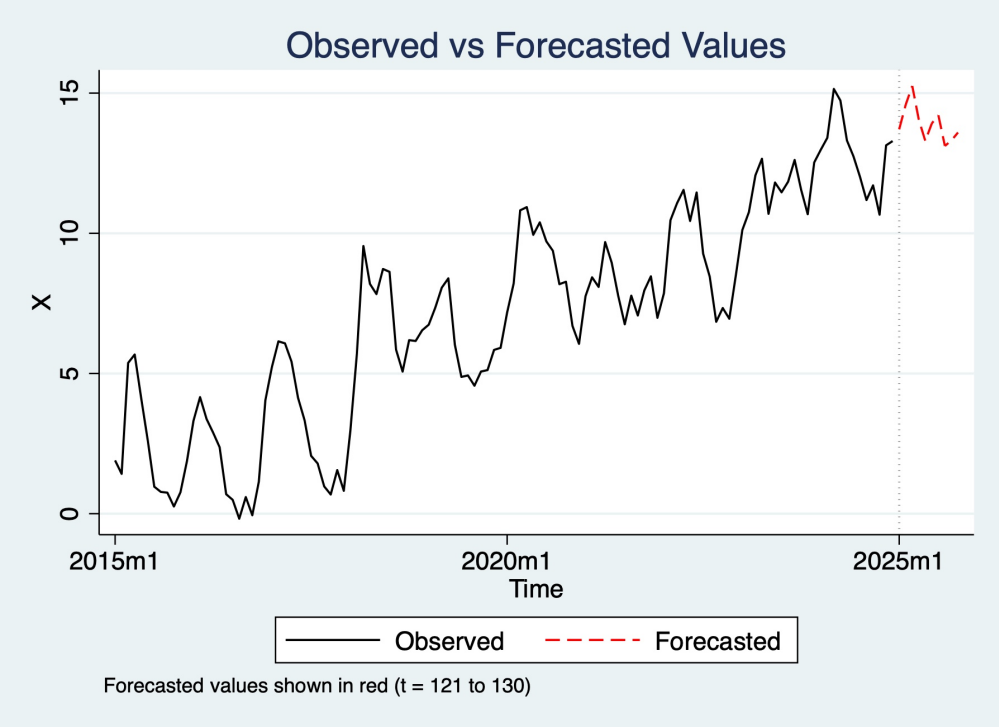

Figure 3.2: Observed vs Forecasted Values

Chapter 4

ARMA

4.1 Understanding the Mathematical Properties of Stationarity

In time series analysis, a process is said to be stationary if its statistical properties such as mean, variance, and autocovariance remain constant over time. One fundamental property of a weakly stationary process is that its expectation does not depend on time. Here, we illustrate two cases:

1. A series with a constant expectation over time.
2. A series where the expectation changes over time (non-stationary).

```
clear all
set obs 100
gen t = _n

* --- Case 1: Constant expectation over time ---
set seed 123
gen y_const = 50 + rnormal(0,5)   // Mean = 50, only random noise

* --- Case 2: Changing expectation over time ---
gen y_trend = 50 + 0.3*t + rnormal(0,5)  // Mean increases by 0.3 per time
    step

* --- Plot both series ---
twoway ///
    (line y_const t, lcolor(blue) lpattern(solid) ///
           yline(50, lpattern(dash) lcolor(red)) ///
           title("Constant Expectation Over Time")) ///
    , name(fig1, replace)

twoway ///
    (line y_trend t, lcolor(green) lpattern(solid) ///
           lwidth(medthick) ///
           title("Changing Expectation Over Time")) ///
    , name(fig2, replace)

graph combine fig1 fig2, col(2) title("Expectation Over Time: Constant vs
    Changing")
```

Listing 4.1: Understanding mathematical properties of Stationarity in STATA

4.2 Simulating Stationarity in Stata

Below we illustrate three cases in Stata:

- Stationary process with constant mean and variance
- Nonstationary process with a trend in mean
- Nonstationary process with changing variance.

We will use the *listings* package to show Stata code and explain each step.

(a) Stationary Process: AR(1) with $|\phi| < 1$

```
clear
set obs 200                  // Create 200 time periods
gen time = _n                // Time index
gen e = rnormal(0,1)         // Random shocks with mean 0, variance 1
gen x = .                    // Empty variable for the series

replace x = e in 1           // Initialize with first shock
forvalues t = 2/200 {
    replace x = 0.6 * x[_n-1] + e in `t'
}
tsset time                   // Declare data as time series
tsline x                     // Plot the series
```

Listing 4.2: Simulating a stationary AR(1) process in Stata

Explanation:

1. We create 200 periods and a time index.
2. Random shocks e_t are drawn from a normal distribution with mean 0 and variance 1.
3. The process $x_t = 0.6x_{t-1} + e_t$ satisfies the stationarity conditions because $|\phi| = 0.6 < 1$.

(b) Nonstationary Process: Random Walk

```
clear
set obs 200
gen time = _n
gen e = rnormal(0,1)
gen x = .

replace x = e in 1
forvalues t = 2/200 {
    replace x = x[_n-1] + e in `t'
}
tsset time
tsline x
```

Listing 4.3: Simulating a nonstationary random walk

Explanation:

- Here $x_t = x_{t-1} + e_t$ has a constant variance in the shocks, but the variance of x_t itself grows with t.
- This violates the constant variance condition of stationarity.

(c) Nonstationary Process: Changing Variance (Heteroskedasticity)

```
clear
set obs 200
gen time = _n
gen e = rnormal(0, time/50)  // Variance increases over time
gen x = e
tsset time
tsline x
```

Listing 4.4: Simulating a process with changing variance

Explanation:

- The shocks have variance proportional to t, so early values are small and later values are large in magnitude.
- This violates the constant variance condition.

With this understanding of stationarity and its violations, we can now proceed to study autoregressive (AR) processes, starting with the AR(1) model.

4.3 Estimating the AR(1) Model

4.3.1 Example 1: Monthly Sales

We begin with the dataset `sales.csv`. The following Stata code imports the data, encodes the month variable as an integer for time series use, and generates a time series plot.

```
***** Sales Dataset ******

import delimited "sales.csv", varnames(1)
encode month, gen(t)   // convert month variable from string to integer //
tsset t                // define time series //
tsline sales           // plot time series
```

Listing 4.5: Time Series Plot for Sales Data

The resulting plot suggests that the sales series is highly persistent and likely non-stationary, as there is no clear tendency for the process to revert to a constant mean.

Next, we estimate the AR(1) model. Since the model is specified as

$$y_t = \phi y_{t-1} + e_t,$$

we omit the constant term in estimation.

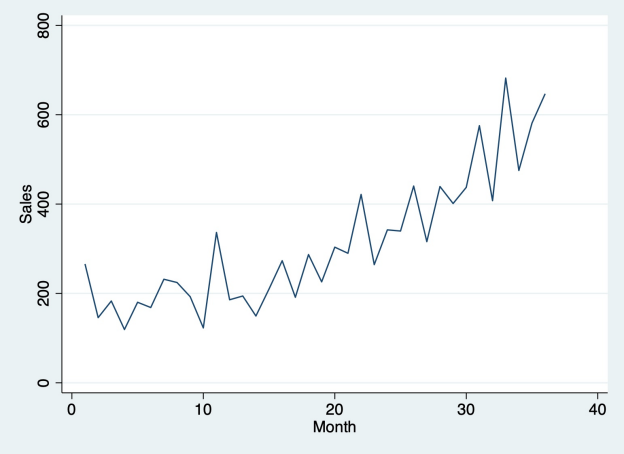

Figure 4.1: Time Series Plot of Monthly Sales

```
*** AR(1) model ***

regress sales l1.sales, nocons   // OLS regression without constant //
arima sales, ar(1) nocons        // AR(1) model estimated via maximum
   likelihood //
```

Listing 4.6: Estimating AR(1) Model for Sales in Stata

Table 4.1: Sales - AR(1) Model Estimates

	(OLS) Sales	(MLE) Sales
Lag of Sales	0.992***	0.966***
	(0.056)	(0.044)
σ		107.635***
		(13.510)
Observations	35	36

Standard errors in parentheses

* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

Interpretation of Sales Model Results

- The OLS regression (`regress`) estimates $\hat{\phi} = 0.991$.
- The maximum likelihood estimation via `arima` gives $\hat{\phi} = 0.966$.

Both estimates are close to one, which confirms strong persistence in the series.

- The OLS estimate (0.991) is very close to unity, suggesting the series is almost a random walk.

- The ARIMA estimate (0.966) is slightly smaller, implying that when properly accounting for the time series structure and likelihood, the process is still persistent but perhaps less extreme than OLS suggests.

Since we have imposed the no-constant restriction, the process is centered around zero. A ϕ value close to one indicates very slow mean reversion, which is consistent with the non-stationary appearance of the time series plot.

4.3.2 Example 2: Annual GDP Series

We now turn to macroeconomic data. The dataset `India GDP.csv` contains annual GDP observations. The following Stata code loads the data and estimates an AR(1) model:

```
******* GDP Dataset ******

clear all
import delimited "India GDP.csv", varnames(1)

tsset year // declare data as time series

****** AR(1) model ****
regress gdp l1.gdp, nocons
arima gdp, ar(1) nocons
```

Listing 4.7: Estimating AR(1) Model for GDP in Stata

Interpretation of GDP Model Results

The AR(1) coefficient estimates are:

- OLS regression (`regress`): $\hat{\phi} = 1.05$
- ARIMA maximum likelihood (`arima`): $\hat{\phi} = 0.99$

Table 4.2: GDP - AR(1) Model Estimates

	(OLS) GDP	(MLE) GDP
Lag of GDP	1.058***	0.999***
	(0.005)	(0.001)
σ		81.630***
		(5.506)
Observations	63	64

Standard errors in parentheses

* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

- The OLS estimate (1.05) is slightly greater than one, which implies an explosive process. In finite samples, OLS often overestimates persistence, especially when the true process is close to a unit root.

- The ARIMA estimate (0.99) is much closer to one but still slightly below, suggesting that GDP follows a near-unit-root process. This is consistent with the common finding in macroeconomics that GDP is highly persistent and may be modeled as a difference-stationary series.

- The difference between OLS and ARIMA again reflects the fact that maximum likelihood methods are better suited for near-unit-root processes and tend to shrink the coefficient slightly compared to OLS.

To better understand the dynamics of the estimated AR(1) process, we generate an impulse response function (IRF). In Stata, this is done using the `irf` commands:

```
*** Impulse Response Function ***

irf create AR1, replace step(10) set(ar1gdp)
irf graph irf
irf drop AR1
```

Listing 4.8: Impulse Response Function for an AR(1) Model in Stata

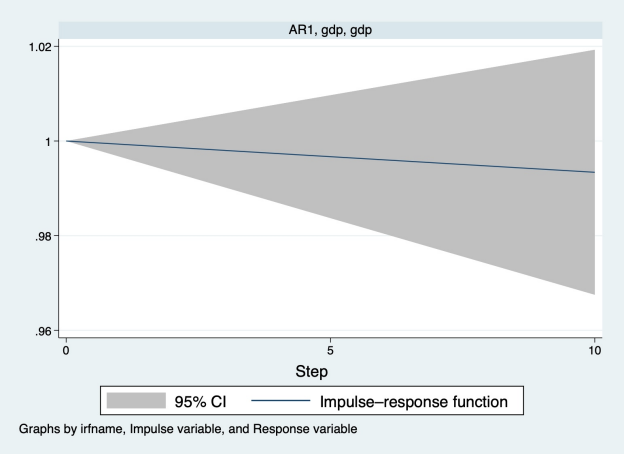

Figure 4.2: Impulse Response Function - GDP AR(1)

Interpretation of the Impulse Response Function

The IRF traces the effect of a one-period shock to GDP over the subsequent 10 periods:

- Because $\hat{\phi}$ is very close to one (0.99 under ARIMA), the shock has a highly persistent effect on GDP. The response decays only very slowly, indicating that the process is nearly non-stationary.

- In fact, if ϕ were exactly one, the shock would be permanent and the IRF would remain flat at the initial impact.

- The OLS estimate ($\hat{\phi} = 1.05$) would imply an explosive IRF, where the effect of a shock actually grows over time. However, the ARIMA estimate suggests near-unit-root persistence without true explosiveness.

4.3.3 Example 3: Growth Rates

Since GDP levels appear non-stationary, we now transform the data into growth rates. Both simple percentage change and log-difference growth rates are commonly used. The Stata code is:

```
** Growth rate **
gen growthgdp = (gdp - l.gdp) * 100 / (l.gdp)

** Growth rate using logarithms **
gen lngdp = ln(gdp)
gen growthrate = (lngdp - l.lngdp) * 100

twoway line gdp year
twoway line lngdp year

** Graph growth rate of GDP over time **
twoway line growthrate year

** Time series declaration and plot **
tsline growthrate
```

Listing 4.9: Growth Rate Plot in Stata

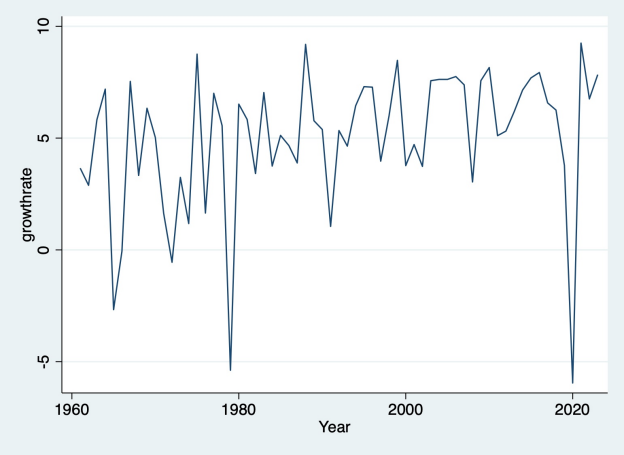

Figure 4.3: Time Series Plot of Growth Rates

- The plots of GDP in levels (`gdp`, `lngdp`) show clear upward trends and strong persistence, indicating non-stationarity.
- In contrast, the plot of GDP growth (`growthrate`) shows fluctuations around a relatively constant mean, with no obvious long-term trend.
- This suggests that growth rates are much closer to being stationary, unlike the GDP levels or sales data examined earlier.
- Intuitively, this happens because differencing the data (i.e., using $y_t - y_{t-1}$ or $\Delta \ln y_t$) removes the long-term trend. This is a form of *detrending*, and it often produces stationary series from non-stationary ones.

Finally, we estimate the AR(1) model using GDP growth rates. The Stata code is:

```
// AR(1) Model //
regress growthrate l.growthrate, r nocons
arima growthrate, ar(1) nocons
```

Listing 4.10: Growth Rates - AR(1) Estimation

Interpretation of Growth Rate Results

Table 4.3: Growth Rates - AR(1) Model Estimates

	OLS Growth Rate	MLE Growth Rate
Lag of Growth Rate	0.730***	0.722***
	(0.078)	(0.137)
σ		4.089***
		(0.378)
Observations	62	63

Standard errors in parentheses

* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

The estimated coefficients are:

- OLS regression (`regress`): $\hat{\phi} = 0.729$
- ARIMA maximum likelihood (`arima`): $\hat{\phi} = 0.722$
- Both estimates are less than one, which indicates a stationary process. Shocks to GDP growth dissipate over time rather than persisting indefinitely.
- The close similarity of OLS (0.729) and ARIMA (0.722) estimates reinforces the stability of the result, in contrast with GDP levels where OLS and ARIMA differed more sharply.
- This aligns with the visual evidence from the growth-rate plot: GDP growth oscillates around a stable mean, unlike GDP levels or sales data which exhibited strong persistence.
- Economically, this means that while the GDP level itself is dominated by long-term trends, short-term fluctuations in growth revert toward a steady average growth rate.

Interpretation of the Impulse Response Function

We next generate the impulse response function for the AR(1) model estimated on GDP growth rates. The Stata commands are:

```
irf create AR2, replace step(10) set(ar1growth)
irf graph irf
irf drop AR2
```

Listing 4.11: Impulse Response Function - Growth Rates

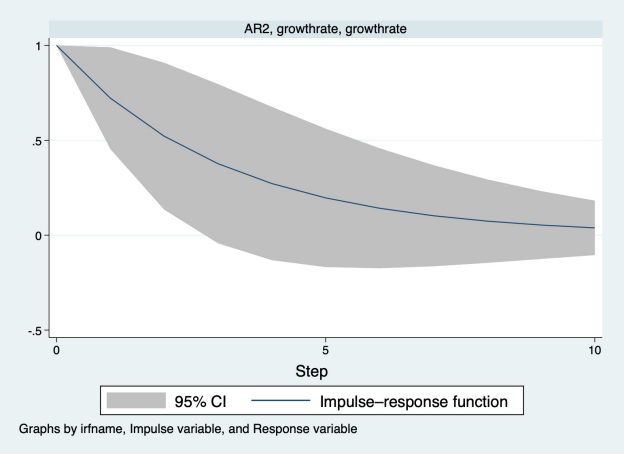

Figure 4.4: Impulse Response Function - Growth Rates AR(1)

The resulting IRF plot is shown in Figure 4.4.

- The IRF shows that a one-period shock to GDP growth has a strong immediate effect, but the impact decays fairly quickly over the next 10 periods.
- This contrasts sharply with the GDP level series, where shocks persisted almost indefinitely due to the near-unit-root behavior.
- The decay pattern here is consistent with our estimated $\phi \approx 0.72$:
  - Each period, the effect of the shock is reduced to roughly 72% of its previous value.
  - By around 5–6 periods, most of the shock's effect has dissipated.
- The shaded region indicates the 95% confidence interval, confirming that the IRF is statistically distinguishable from zero for a few periods, but not indefinitely.

Forecasting AR(1)

Finally, we use the estimated AR(1) model to generate forecasts of GDP growth rates. The following code extends the series by 5 periods, predicts values, and plots actual versus forecasted growth rates:

```
tsappend, add(5)
predict growthrate_predict

twoway ///
    (line growthrate year, lcolor(blue) lwidth(medium) lpattern(solid) ///
        legend(label(1 "Actual Growth Rate"))) ///
    (line growthrate_predict year, lcolor(red) lwidth(medium) lpattern(dash)
       ///
        legend(label(2 "Forecasted Growth Rate"))), ///
    title("GDP Growth Rate: Actual vs Forecast") ///
    ytitle("Growth Rate (%)") ///
    xtitle("Year") ///
    legend(order(1 2) ring(0) pos(11) col(1))
```

Listing 4.12: Forecasting Growth Rates

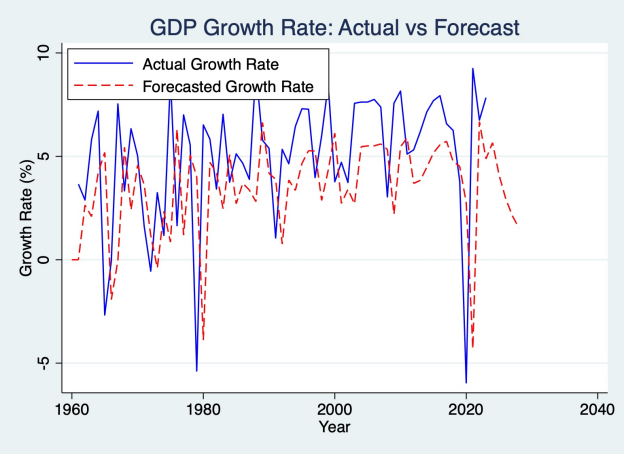

Figure 4.5: Forecasting Growth Rates from AR(1) Model

Figure 4.5 compares the actual GDP growth rate (blue solid line) with the forecasted growth rate from the AR(1) model (red dashed line).

- The forecasts follow the general mean-reverting nature of the growth rate process: instead of drifting upward or downward indefinitely (as would happen with GDP levels), the predictions remain bounded around a central tendency of roughly 4–6%.
- The forecasted path is smoother than the actual series. This is typical of AR(1) models: they capture persistence but not large short-run fluctuations. As a result, extreme spikes and crashes in actual GDP growth (e.g., the sharp dip around 2020) are not fully reproduced.
- In the forecast horizon (beyond the last observed year), the model projects growth gradually converging toward its long-run mean. The predicted values stay positive and hover close to historical averages.
- This highlights both the strength and limitation of simple AR(1) models:
  - Strength: they capture persistence and provide stable forecasts.
  - Limitation: they cannot anticipate structural breaks, crises, or unusually large volatility.

4.4 Estimating the AR(p) Model

So far, we have worked with an AR(1) model, where the current growth rate depends only on its immediate lag. We now generalize to an autoregressive process of order p, where:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t$$

The Stata code for estimating an AR(3) model is:

```
// AR(p) Model where p = 3 //
regress growthrate l.growthrate l2.growthrate l3.growthrate, r nocons
arima growthrate, ar(1/3) nocons
```

Listing 4.13: Estimating the AR(p) Model

Table 4.4: AR(3) Model Estimates

	OLS growthrate	MLE growthrate
L.growthrate	0.309**	0.309***
	(0.143)	(0.113)
L2.growthrate	0.242*	0.240**
	(0.139)	(0.109)
L3.growthrate	0.360**	0.348***
	(0.154)	(0.102)
σ		3.514***
		(0.294)
Observations	60	63

Standard errors in parentheses

* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

Interpretation

- In an AR(3) specification, the growth rate is explained by its past three values.
- This allows the model to capture richer dynamics — e.g., cyclical patterns or delayed effects of shocks — that the AR(1) model cannot.
- In practice, one chooses p based on model selection criteria such as AIC, BIC, or by examining autocorrelation (ACF) and partial autocorrelation (PACF) plots.
- If higher-order lags are insignificant, the model will effectively reduce to a simpler AR(1). If multiple lags are significant, it implies that shocks take several periods to fully dissipate.

Next Steps

Later, we will compare different AR(p) models formally using information criteria and diagnostic checks. This will help us decide whether the AR(1) is sufficient for GDP growth or whether higher-order dynamics are present.

Forecasting

We now generate forecasts from the AR(p) model. The following Stata code extends the sample by 10 periods, predicts values, and plots actual versus forecasted growth rates:

```
// Forecasting where p = 3 //
drop growthrate_predict
tsappend, add(10)
predict growthrate_predict

twoway ///
    (line growthrate year, lcolor(blue) lwidth(medium) lpattern(solid) ///
        legend(label(1 "Actual Growth Rate"))) ///
```

```
    (line growthrate_predict year, lcolor(red) lwidth(medium) lpattern(dash)
       ///
        legend(label(2 "Forecasted Growth Rate"))), ///
    title("GDP Growth Rate: Actual vs Forecast") ///
    ytitle("Growth Rate (%)") ///
    xtitle("Year") ///
    legend(order(1 2) ring(0) pos(11) col(1))
```

Listing 4.14: Forecasting the AR(p) Model

The figure below shows the actual GDP growth rate (solid blue line) and the forecasted growth rate from an AR(p) model (dashed red line).

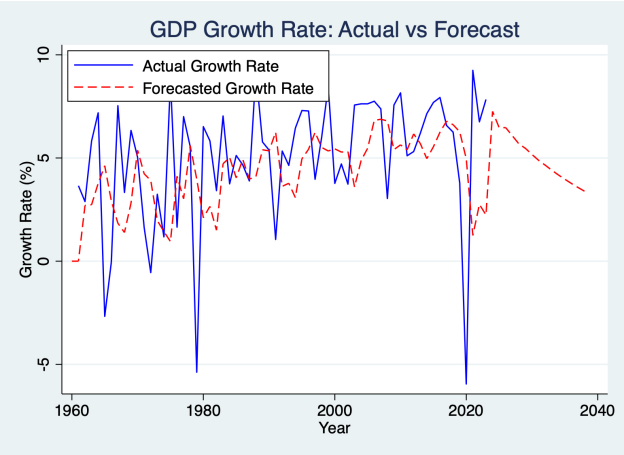

Figure 4.6: Forecasting AR(3) Model for Growth Rates

- The AR(p) model tracks the historical data reasonably well, capturing broad persistence and mean-reversion in growth rates.
- In the out-of-sample period, the forecast converges toward a long-run mean of around 3–5%, reflecting the model's dynamics.
- Forecasts are smoother than actual data — sharp booms or crises (e.g., 2020) are not replicated. This is because the AR model relies only on past values and does not incorporate external shocks.
- Hence, AR forecasts provide a useful *baseline projection*, but real-world events and structural breaks can cause deviations.

4.5 Moving Average Model: MA(1)

```
// MA(1) Model //
arima growthrate, ma(1) nocons
```

Listing 4.15: Estimating MA(1) Model

Model specification:

$$y_t = u_t + \beta u_{t-1}, \quad u_t \sim \text{WN}(0, \sigma^2)$$

Estimation results:

$$\hat{\beta} = 0.535, \quad \hat{\sigma} = 4.84$$

Interpretation:

- The MA(1) coefficient $\hat{\beta} = 0.535$ implies that a one-unit shock to growth in period $t-1$ carries over positively into the current period. Specifically, a unit increase in the previous innovation u_{t-1} raises y_t by approximately 0.535 units.
- The innovation standard deviation $\hat{\sigma} = 4.84$ measures the typical size of unpredictable fluctuations in growth. On average, growth deviates from its predicted value by about 4.84 percentage points due to random shocks.

Impulse response function (IRF):

```
// Impulse Response Function //
irf create ma1_irf, replace
irf graph irf, irf(ma1_irf)
```

Listing 4.16: IRF MA(1) Model

$$IRF(0) = 1, \quad IRF(1) = 0.535, \quad IRF(h) = 0 \text{ for } h \geq 2$$

Thus, shocks to growth have a strong immediate impact, a smaller but positive spillover into the next period, and then die out completely after one lag.

Forecasting:

- One-step ahead forecast: $\hat{y}_{T+1|T} = 0.535\, u_T$
- Two-steps ahead forecast: $\hat{y}_{T+2|T} = 0$
- Beyond two steps: $\hat{y}_{T+h|T} = 0$ for $h \geq 2$

Forecasts revert quickly to the mean (zero, since no constant is included), but forecast error variance increases with horizon.

4.6 Moving Average Model: MA(p)

```
// MA(1) Model //
arima growthrate, ma(1/3) nocons
```

Listing 4.17: Estimating MA(p) Model with p

Model specification:

$$y_t = u_t + \beta_1 u_{t-1} + \beta_2 u_{t-2} + \beta_3 u_{t-3}, \quad u_t \sim \text{WN}(0, \sigma^2)$$

Estimation results:

$$\hat{\beta}_1 = 0.495, \quad \hat{\beta}_2 = 0.498, \quad \hat{\beta}_3 = 0.558$$

Interpretation:

- The MA(3) model implies that shocks to growth have effects that persist for up to three periods.
- A one-unit shock at time t directly increases y_t by 1 unit, carries over as an increase of 0.495 units in y_{t+1}, 0.498 units in y_{t+2}, and 0.558 units in y_{t+3}.
- Beyond three periods, the effect of the shock disappears completely.

Impulse Response Function (IRF):

```
// Impulse Response Function //
irf create ma3_irf, replace
irf graph irf, irf(ma3_irf)
```

Listing 4.18: IRF MA(1/3) Model

$$IRF(0) = 1, \quad IRF(1) = 0.495, \quad IRF(2) = 0.498, \quad IRF(3) = 0.558, \quad IRF(h) = 0 \text{ for } h \geq 4$$

Thus, growth rates show short-run persistence: shocks die out after three periods, but the magnitudes (0.495, 0.498, 0.558) suggest that shocks have economically meaningful spillovers before fading out.

4.7 Model Selection

Constructing GDP Growth Rates

In time series analysis, raw GDP levels are often non-stationary. To study business cycle fluctuations, it is common to work with *growth rates* instead of levels. We define two types of growth rates using quarterly Indian GDP data:

- **Quarter-on-Quarter (QoQ) growth:** Measures how GDP changes relative to the immediately preceding quarter.
$$g_t^{qoq} = 100 \times \frac{GDP_t - GDP_{t-1}}{GDP_{t-1}}$$
This is useful for short-term dynamics, but can be noisy due to seasonal effects.
- **Year-on-Year (YoY) growth:** Measures how GDP in a given quarter compares with the same quarter in the previous year.
$$g_t^{yoy} = 100 \times \frac{GDP_t - GDP_{t-4}}{GDP_{t-4}}$$
This smooths seasonal patterns, making it more suitable for ARMA modeling.

The following code computes both growth measures:

```
*************************************************************
* Construct growth rates
*************************************************************
gen g_qoq = 100*(gdp - L.gdp)/L.gdp
gen g_yoy = 100*(gdp - L4.gdp)/L4.gdp
label var g_qoq "Quarter-on-quarter GDP growth (%)"
label var g_yoy "Year-on-year GDP growth (%)"
* Plot both series
tsline g_qoq, title("India: QoQ GDP growth (%)") ytitle("%") legend(off)
tsline g_yoy, title("India: YoY GDP growth (%)") ytitle("%") legend(off)
```

Listing 4.19: Constructing Growth Rates

Stationarity and Choice of Growth Series

Figure 4.7: GDP Growth - YoY and QoQ (%)

Figure 4.7 plots the QoQ and YoY growth series. Visual inspection indicates that both series are stationary. Since our primary interest lies in modeling annual growth patterns, we focus on the YoY series, g_t^{yoy}.

Autocorrelation and Partial Autocorrelation

A natural first step in ARMA modeling is to examine the *autocorrelation function (ACF)* and *partial autocorrelation function (PACF)*:

- The **ACF** measures how g_t^{yoy} is correlated with its own lags. A sharp cutoff in the ACF helps identify the order of the MA component (q).
- The **PACF** isolates the direct correlation between g_t^{yoy} and its lags, after controlling for shorter lags. A sharp cutoff in the PACF helps identify the AR order (p).

```
** Empirical ACF & PACF (YoY growth)
ac   g_yoy, lags(20)
pac g_yoy, lags(20)
corrgram g_yoy, lags(12)
```

Listing 4.20: ACF and PACF

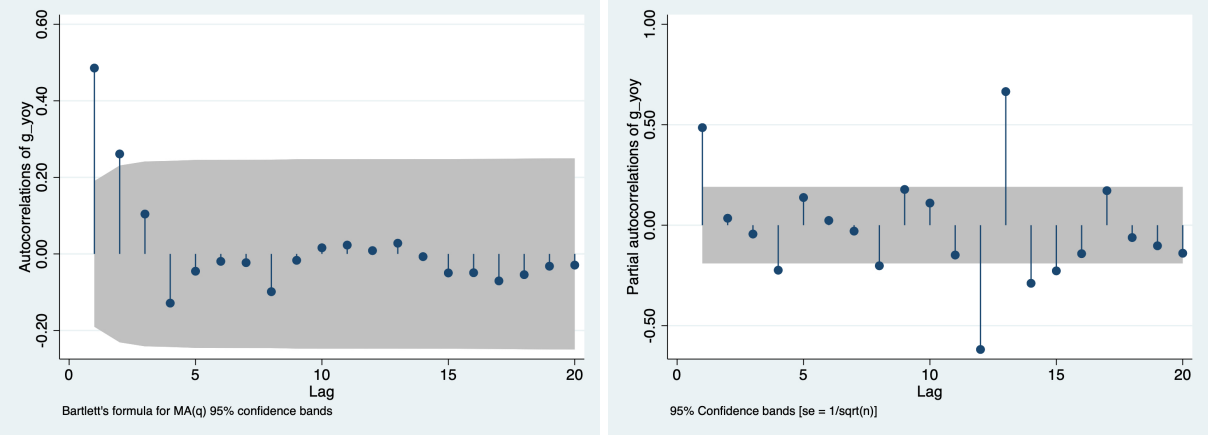

Figure 4.8: ACF and PACF plots for YoY GDP growth

The plots (Figure 4.8) show that the ACF cuts off around lag $q = 2$, while the PACF cuts off around lag $p = 1$. Based on this diagnostic, we narrow our attention to the following candidate models:

$$\text{ARMA}(0,0), \quad \text{ARMA}(0,1), \quad \text{ARMA}(1,0), \quad \text{ARMA}(1,1), \quad \text{ARMA}(1,2), \quad \text{ARMA}(0,2).$$

The next step involves choosing among these using information criteria.

```
** Model Selection - AIC and BIC
quietly arima g_yoy, arima(0,0,0)
estat ic
est store arima00

quietly arima g_yoy, arima(0,0,1)
estat ic
est store arima01

quietly arima g_yoy, arima(0,0,2)
estat ic
est store arima02

quietly arima g_yoy, arima(1,0,0)
estat ic
est store arima10

quietly arima g_yoy, arima(2,0,0)
estat ic
est store arima20

quietly arima g_yoy, arima(1,0,1)
estat ic
est store arima11

quietly arima g_yoy, arima(1,0,2)
estat ic
est store arima12
```

```
quietly arima g_yoy, arima(2,0,1)
estat ic
est store arima21

quietly arima g_yoy, arima(2,0,2)
estat ic
est store arima22

* Side-by-side comparison
est table arima00 arima01 arima02 arima10 arima11 arima12 arima20 arima21
   arima22 , stats(aic bic ll) star
```

Listing 4.21: AIC and BIC Selection Criteria

To formally choose among the candidate models, we estimate each specification and compare them using the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Both criteria reward goodness-of-fit while penalizing model complexity, with BIC penalizing complexity more heavily. Based on the AIC and BIC criteria, the **AR(1)** model is selected as the preferred specification. Although the ARMA(2,2) model shows slightly lower AIC and BIC values, the coefficient on the first lag is statistically insignificant. Since this contradicts our economic intuition that past growth should matter for current growth, we reject the ARMA(2,2) in favor of the more parsimonious and interpretable AR(1) model.

Residual Diagnostics

Having selected the AR(1) model as our preferred specification, we now examine whether the residuals satisfy the usual assumptions of white noise. This involves testing for autocorrelation, inspecting the residual correlogram, and checking normality.

```
************************************************************
* 5) Residual diagnostics for the preferred model
************************************************************
arima g_yoy, arima(1,0,0)

* Obtain residuals
predict ehat, resid

* Portmanteau (Ljung-Box) test for autocorrelation
wntestq ehat, lags(12)

* Visual inspection of residual autocorrelation
corrgram ehat, lags(12)

* Normality checks (optional)
kdensity ehat, normal
sktest ehat
```

Listing 4.22: Residual diagnostics for AR(1) model

The Portmanteau (Ljung–Box) test fails to reject the null hypothesis of no residual autocorrelation ($p > 0.05$), suggesting that the AR(1) residuals behave like white noise. The correlogram of residuals confirms that most autocorrelations lie within the confidence bands. Finally, the residual distribution is approximately normal, supporting the adequacy of the AR(1) specification.

Chapter 5

Unit Root Tests

5.1 Non-Stationary Processes

In time series analysis, a process is said to be *stationary* if its statistical properties—such as mean, variance, and autocovariance—do not change over time. Many real-world economic and financial series, however, do not satisfy this requirement. Instead, they exhibit persistent movements, long-term drifts, or evolving variability. Such series are referred to as *non-stationary processes*.

Non-stationarity can arise for different reasons, and understanding the source of non-stationarity is essential for model selection, forecasting, and statistical inference. Non-stationary processes can broadly be classified into those driven by deterministic components (such as linear trends) and those driven by stochastic components (such as cumulative shocks). The distinction between deterministic and stochastic trends is fundamental: while deterministic trends can be removed through detrending, stochastic trends typically require differencing to achieve stationarity.

To illustrate the main forms of non-stationary behavior encountered in practice, we consider four canonical examples:

- **Random Walk:** A purely stochastic trend generated by the accumulation of random shocks.
- **Random Walk with Drift:** A random walk augmented with a constant drift term, introducing a systematic upward or downward movement.
- **Deterministic Trend Process:** A series driven by a known, time-dependent deterministic component, such as a linear trend.
- **Random Walk with Drift and Deterministic Trend:** A process combining both stochastic and deterministic trends, resulting in the most strongly non-stationary behavior among the four.

These processes differ not only in their visual appearance but also in their statistical properties and the techniques required to transform them into stationary series. The following figures illustrate each type of non-stationary behavior.

5.1.1 Random Walk

A random walk is defined as:

$$y_t = y_{t-1} + \varepsilon_t,$$

where ε_t is white noise. The process has no long-run mean and its variance increases without bound, making it strongly non-stationary. Even though increments are stationary, the level of the process is not.

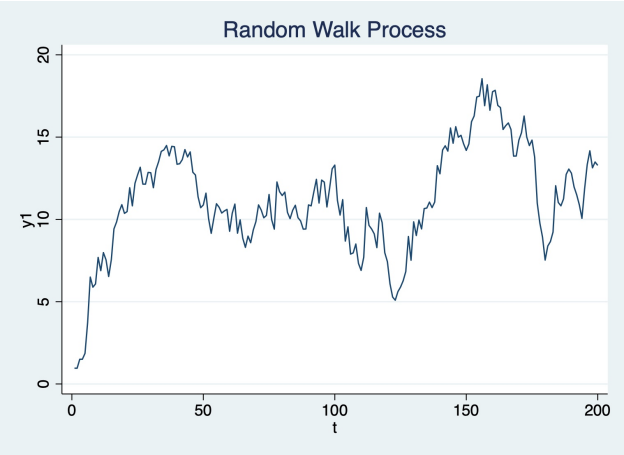

Figure 5.1: Random Walk Process

5.1.2 Random Walk with Drift

A random walk with drift takes the form:

$$y_t = \mu + y_{t-1} + \varepsilon_t,$$

where μ is a constant drift term. The series exhibits both a stochastic trend (random walk part) and a deterministic upward trend due to the drift.

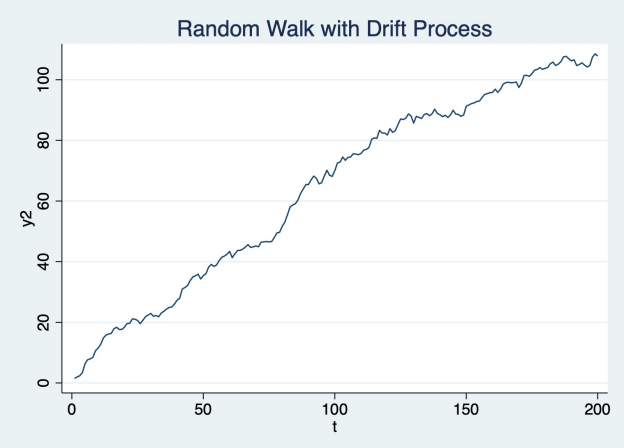

Figure 5.2: Random Walk with Drift

5.1.3 Deterministic Trend

A deterministic trend model is:

$$y_t = \beta_0 + \beta_1 t + \varepsilon_t.$$

The series increases (or decreases) predictably over time. Although the trend makes the series non-stationary, detrending can restore stationarity.

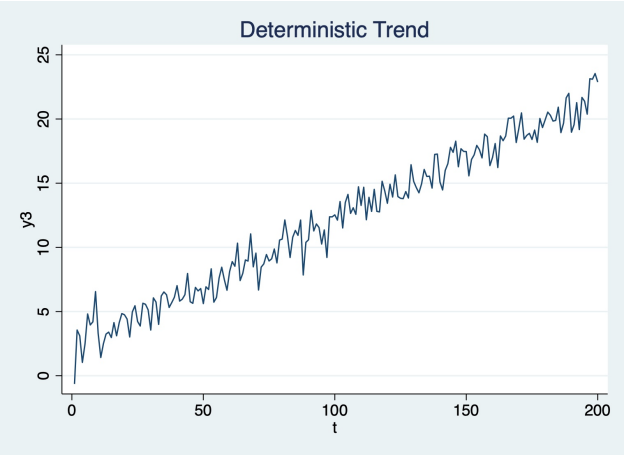

Figure 5.3: Deterministic Trend Process

5.1.4 Random Walk with Drift and Deterministic Trend

This process combines both stochastic and deterministic trend components:

$$y_t = \beta_0 + \beta_1 t + y_{t-1} + \varepsilon_t.$$

It is highly non-stationary since the random walk part produces an ever-increasing variance while the deterministic trend adds additional growth.

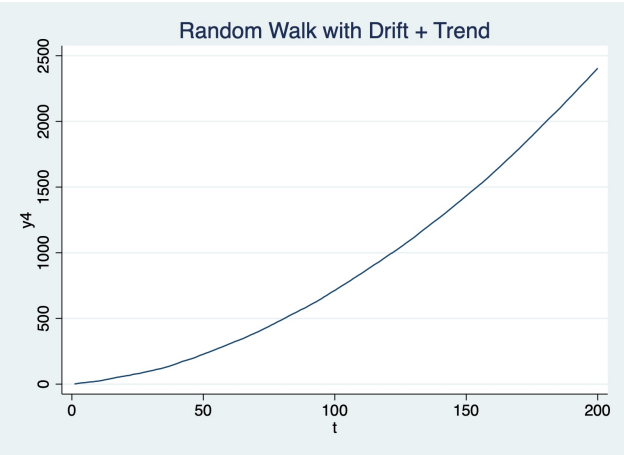

Figure 5.4: Random Walk with Drift and Trend

```
*****************************************************
* Simulating Types of Non-Stationary Processes
* with Differencing
*****************************************************

clear all
```

```
set obs 200                  // number of observations
gen t = _n                   // time index
tsset t
set seed 12345               // reproducibility

*****************************************************
* 1. Random Walk
* y_t = y_{t-1} + e_t
*****************************************************
gen e1 = rnormal(0,1)
gen y1 = .
replace y1 = e1 in 1
forvalues i = 2/200 {
    replace y1 = y1[_n-1] + e1 in `i'
}
gen dy1 = D.y1               // first difference

tsset t
tsline y1,  ///
       title("Random Walk Process")
*****************************************************
* 2. Random Walk with Drift
* y_t = alpha + y_{t-1} + e_t
*****************************************************
gen alpha = 0.5
gen e2 = rnormal(0,1)
gen y2 = .
replace y2 = e2 in 1
forvalues i = 2/200 {
    replace y2 = alpha + y2[_n-1] + e2 in `i'
}
gen dy2 = D.y2               // first difference
tsset t
tsline y2,  ///
       title("Random Walk with Drift Process")
*****************************************************
* 3. Deterministic Trend Process
* y_t = a + beta*t + e_t
*****************************************************
gen a = 2
gen beta  = 0.1
gen e3 = rnormal(0,1)
gen y3 = a + beta*t + e3
gen dy3 = D.y3               // first difference
tsset t
tsline y3, title("Deterministic Trend")
*****************************************************
* 4. Random Walk with Drift + Deterministic Trend
* y_t = alpha + beta*t + y_{t-1} + e_t
*****************************************************
gen e4 = rnormal(0,1)
gen y4 = .
replace y4 = e4 in 1
forvalues i = 2/200 {
```

```
    replace y4 = 2 + 0.1*t + y4[_n-1] + e4 in `i'
}
gen dy4  = D.y4             // first difference
gen d2y4 = D.dy4            // second difference
tsset t
tsline y4, title("Random Walk with Drift + Trend")
```

Listing 5.1: Simulating Non-Stationary Series

In the earlier chapters, we have estimated and analyzed ARMA models under the assumption of stationarity — that is, the statistical properties of the series (mean, variance, autocorrelation) remain constant over time. However, in real-world applications, most economic and financial time series — such as GDP, interest rates, unemployment rates, and prices — often exhibit trend components or other forms of non-stationarity. Therefore, before proceeding to any further modeling, it is essential to formally test whether a series is stationary or not.

To illustrate this process, we now turn to the Apple stock price dataset, examining the closing price, its log transformation, and the corresponding returns.

```
import delimited "Apple Stock.csv", varnames(1)

* Convert date to proper Stata date format
gen date2 = date(date, "DMY")
format date2 %td

* Extract components for convenience
gen day = day(date2)
gen month = month(date2)
gen year = year(date2)

* Create a time index
egen time = group(year month day)
label variable time "Time"

* Declare time variable
tsset time

* Plot the closing price
twoway line close time, ///
    title("Apple Stock Closing Price") ///
    ytitle("Price (USD)") xtitle("Time")

* Log transformation
gen lnprice = ln(close)
tsline lnprice, ///
    title("Apple Stock Log Closing Price") ///
    ytitle("ln(Price)")

* Compute log returns
gen lnprice_lag = lnprice[_n-1]
gen ret = lnprice - lnprice_lag
gen ret_pct = 100 * ret
label var ret "log return (ln P_t - ln P_{t-1})"
label var ret_pct "Daily returns (%)"
```

```
* Plot log price and returns
tsline lnprice, title("Apple Stock Log Closing Price") saving(lnprice_graph,
    replace)
tsline ret_pct, title("Apple Stock Daily Returns (%)") saving(ret_pct_graph,
    replace)

* Combine the two graphs
graph combine lnprice_graph.gph ret_pct_graph.gph, ///
    title("Apple Stock: Price and Returns") ///
    rows(2) ///
    note("Data source: Apple Stock CSV")
```

Listing 5.2: Visually Detecting Stationarity

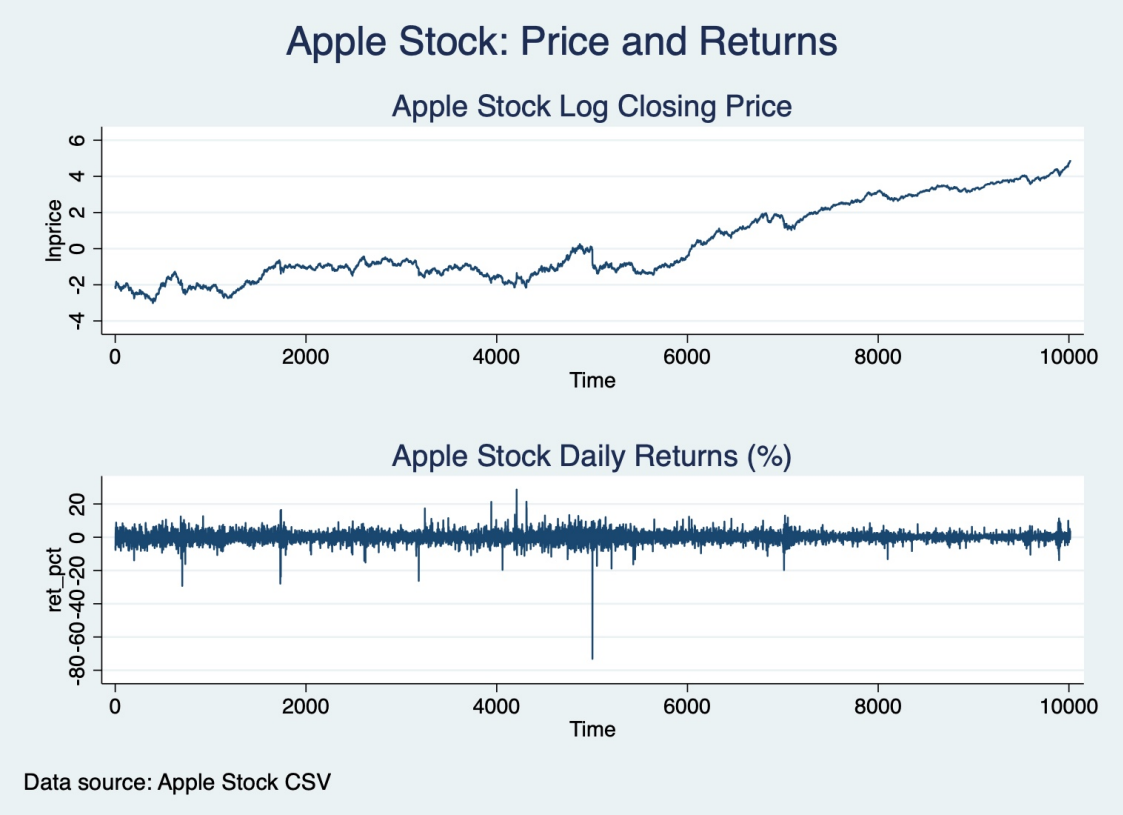

Figure 5.5: Apple Stock - Closing Price and Returns

It is evident from the graphs that the closing price series is non-stationary, while the returns series appears stationary. The price series displays a clear trend, whereas the returns fluctuate randomly around a constant mean.

5.2 Augmented Dickey - Fuller Test

To formally test these visual impressions, we employ the Augmented Dickey–Fuller (ADF) test. Recall that the ADF test is based on the following regression equation:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta y_{t-i} + \varepsilon_t$$

where:

- y_t is the series under investigation,
- $\Delta y_t = y_t - y_{t-1}$ denotes the first difference,
- t is a deterministic time trend (included if appropriate),
- p is the number of lagged difference terms included to account for serial correlation, and
- ε_t is a white-noise error term.

The hypotheses of the test are as follows:

$$\begin{aligned} H_0 &: \gamma = 0 \qquad \text{(the series has a unit root, i.e., it is non-stationary)} \\ H_1 &: \gamma < 0 \qquad \text{(the series is stationary)} \end{aligned}$$

If the estimated γ is significantly negative (i.e., the test statistic is less than the critical value), we reject the null hypothesis and conclude that the series is stationary. Otherwise, we fail to reject the null, implying the presence of a unit root and hence non-stationarity.

Before running the ADF test, we must determine the appropriate number of lags (p) to include in the regression. Choosing the correct lag length is important to ensure that the residuals of the test equation are free from autocorrelation. If too few lags are used, autocorrelation may bias the test statistic; if too many are included, the test may lose power.

There are two common approaches to selecting the lag length:

1. **Visual inspection:** Examine the *autocorrelation function (ACF)* and *partial autocorrelation function (PACF)* plots of the series. These graphs can provide a preliminary sense of how many lags are needed to capture the serial dependence in the data.
2. **Information criteria:** Use statistical criteria such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC) to formally select the lag order that minimizes the chosen criterion.

In `Stata`, the optimal number of lags for the ADF test can be identified using the `varsoc` command, which computes the AIC, HQIC, and BIC for different lag lengths. The lag with the lowest BIC value is typically preferred, as it imposes a stronger penalty on overfitting.

```
******* Determine optimal lag length using information criteria *******

varsoc lnprice
```

Listing 5.3: Selecting optimal lag length in Stata

Table 5.1: Lag-order selection criteria for `lnprice`

Lag	LL	LR	df	p-value	FPE	AIC	HQIC	SBIC
0	-21624.3	-	-	-	4.4036	4.3203	4.3205	4.3210
1	21198.5	85645.0	1	0.000	0.0008	-4.2346	-4.2341	**-4.2332***
2	21200.0	3.001	1	0.083	0.0008	-4.2347	-4.2340	-4.2326
3	21202.0	4.011	1	0.045	0.0008	-4.2349	-4.2340	-4.2321
4	21206.4	**8.910***	1	0.003	**0.0008**	**-4.2356***	**-4.2344***	-4.2320

Once the appropriate lag length has been determined, the ADF test can be conducted on both the log price and returns series to formally assess whether each is stationary or contains a unit root. Based on the information criteria output from the `varsoc` command, we determine the optimal lag length for the ADF regression. The selection is guided by the Bayesian Information Criterion (SBIC/BIC), which penalizes model complexity more heavily and helps avoid overfitting.

We conduct the Augmented Dickey–Fuller (ADF) test under three different model specifications: (1) with trend, (2) with drift, and (3) without constant or trend. Each specification corresponds to a slightly different version of the ADF regression model.

Table 5.2: ADF Test Results for `lnprice` (Lag = 1)

Specification	Model Components	ADF Statistic	1% CV	5% CV	10% CV	p-value
With trend	Constant, trend	-1.697	-3.960	-3.410	-3.120	0.7523
With drift	Constant only	0.606	-2.327	-1.645	-1.282	0.7276
No constant	None	0.858	-2.580	-1.950	-1.620	-

1. **ADF with trend:**

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta y_{t-i} + \varepsilon_t$$

Hypotheses:

$$H_0 : \gamma = 0 \quad \text{(unit root with trend)} \quad \text{vs.} \quad H_1 : \gamma < 0 \quad \text{(trend-stationary)}$$

2. **ADF with drift (no trend):**

$$\Delta y_t = \alpha + \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta y_{t-i} + \varepsilon_t$$

Hypotheses:

$$H_0 : \gamma = 0 \quad \text{(unit root with drift)} \quad \text{vs.} \quad H_1 : \gamma < 0 \quad \text{(stationary around a mean)}$$

3. **ADF without constant or trend:**

$$\Delta y_t = \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta y_{t-i} + \varepsilon_t$$

Hypotheses:

$$H_0 : \gamma = 0 \quad \text{(random walk without drift)} \quad \text{vs.} \quad H_1 : \gamma < 0 \quad \text{(stationary)}$$

Across all three model specifications, the ADF test statistic is greater (less negative) than the corresponding critical values at the 1%, 5%, and 10% significance levels. Hence, we fail to reject the null hypothesis of a unit root in each case. These results confirm that the log of the Apple stock price (`lnprice`) is non-stationary, consistent with the visual evidence of a trending series.

Chapter 6

Structural Breaks

Macroeconomic relationships often shift over time as policy regimes, institutions, and external shocks change, creating structural breaks that alter underlying parameters. Ignoring these breaks can lead to incorrect inferences—for example, fiscal or monetary effects estimated under one regime may not apply in another. Detecting such shifts is therefore crucial for policy evaluation, model stability, and accurate forecasting. In this course, we will use the Chow test for breaks at known points and the Bai–Perron test for detecting breaks at unknown points.

Figure 6.1 illustrates this idea using a simulated GDP growth series generated from an AR(1) process with a structural break in 1971. Before the break, GDP growth follows a regime with a lower mean and variance; after 1971, the mean level of growth increases and the volatility becomes noticeably higher. Ignoring such a break would incorrectly assume a single set of parameters governs the entire sample period.

```
clear all
set obs 100
gen year = 1921 + _n - 1     // simulate 1921-2020
set seed 12345

* Define break year
local break = 1971

* Create error term
gen e = rnormal(0, 1)

* Initialize GDP growth series
gen gdp_growth = .

* Simulate AR(1) process with a break in mean and variance
replace gdp_growth = 3 + e[1]   if year == 1921

forvalues i = 2/100 {
    if `i' < `=`break' - 1920' {
        * Pre-break regime
        replace gdp_growth = 3 + 0.5*gdp_growth[`=`i'-1'] + rnormal(0,1) in `i
            '
    }
    else {
        * Post-break regime (higher mean and variance)
```

```
        replace gdp_growth = 6 + 0.5*gdp_growth[`=`i'-1'] + rnormal(0,1.5) in
            `i'
    }
}

* Generate dummy for post-break period
gen post = year >= `break'

* Plot the series with a vertical line at the break
twoway ///
    (line gdp_growth year, lcolor(gs8) lwidth(medthick)) ///
    , ///
    title("Simulated GDP Growth with Structural Break in `break'") ///
    ytitle("GDP Growth (%)") xtitle("Year") ///
    xline(`break', lcolor(red) lpattern(dash) lwidth(thick)) ///
    note("Vertical red dashed line = Structural break in `break'") ///
    legend(off)
```

Listing 6.1: Simulating GDP Growth with Structural Breaks

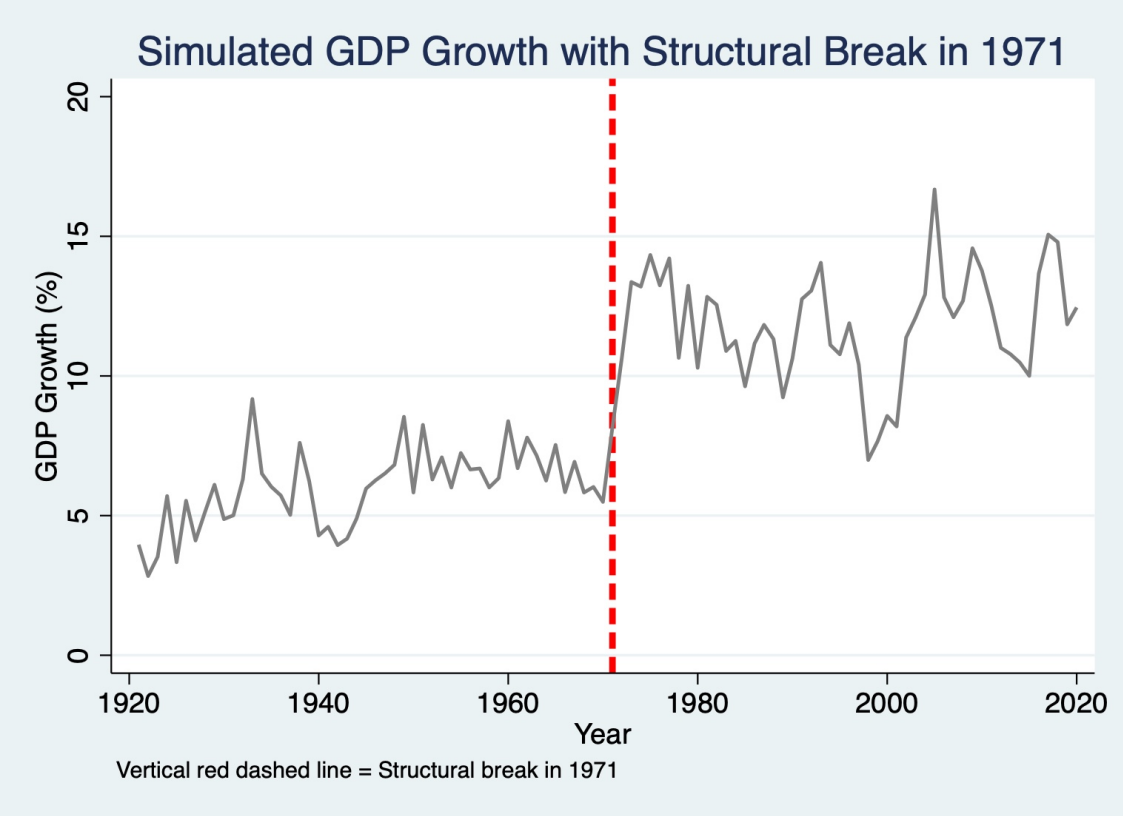

Figure 6.1: Simulated GDP Growth with a Structural Break in 1971.

6.1 Structural Breaks with *known* breakpoint

The Chow test is a classical econometric test for the presence of a structural break at a *known* breakpoint. It is based on comparing the fit of two separate regressions—before and after the suspected break—with that of a pooled regression estimated over the full sample.

Consider a linear model:

$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t, \quad t = 1, \ldots, T \tag{6.1}$$

Suppose there is a known breakpoint at observation T_b. The Chow test examines whether the parameters before and after T_b are equal. Specifically, we estimate the model separately for the two regimes:

Regime 1 (Pre-break):

$$y_t = \beta_0^{(1)} + \beta_1^{(1)} x_t + \varepsilon_t^{(1)}, \qquad t = 1, \ldots, T_b. \tag{6.2}$$

Regime 2 (Post-break):

$$y_t = \beta_0^{(2)} + \beta_1^{(2)} x_t + \varepsilon_t^{(2)}, \qquad t = T_b + 1, \ldots, T. \tag{6.3}$$

$$H_0 : \beta_0^{(1)} = \beta_0^{(2)}, \quad \beta_1^{(1)} = \beta_1^{(2)} \tag{6.4}$$

In the Stata implementation of the Chow test, the following regression is run:

$$y_t = \alpha + \beta_1 t + \beta_2 (t \times post_t) + \beta_3 post_t + \varepsilon_t,$$

A structural break at the known breakpoint is equivalent to a change in either the intercept or the slope after the break. This corresponds to testing the restrictions

$$H_0 : \ \beta_2 = 0 \quad \text{and} \quad \beta_3 = 0.$$

Under the null hypothesis H_0, the post-break regression has the same intercept and slope as the pre-break regression, implying no structural break. The alternative hypothesis is

$$H_A : \ \beta_2 \neq 0 \quad \text{or} \quad \beta_3 \neq 0,$$

which indicates that at least one parameter changes after the breakpoint, and therefore a structural break is present.

```
gen time=_n
gen post=1 if year>1970 // 1971 is the year of breakpoint
replace post=0 if year<=1970
regress gdp_growth time c.post#c.time post // chow test regression
test c.post#c.time post // F-test
```

Listing 6.2: Chow Test for known Structural Breaks

Under the null hypothesis of parameter stability, the restricted (pooled) model is valid. The test statistic is:

$$F = \frac{(SSR_P - (SSR_1 + SSR_2))/k}{(SSR_1 + SSR_2)/(T_1 + T_2 - 2k)} \tag{6.5}$$

where:

- SSR_P is the sum of squared residuals from the pooled regression,
- SSR_1 and SSR_2 are the residual sums of squares from the subsample regressions before and after T_b,
- k is the number of estimated parameters.

Under H_0, $F - stat$ follows an $F \sim (k, T_1 + T_2 - 2k)$ distribution. If $F - stat$ exceeds the critical value, we reject the null hypothesis and conclude that a structural break exists at T_b.

	Pooled	**Regime 1 (1921-1970)**	**Regime 2 (1972-2020)**
t	0.0954	0.0483	0.0171
	(0.0068)	(0.0117)	(0.0198)
Constant	4.0696	4.7493	10.5011
	(0.3961)	(0.3428)	(1.5225)
N	100	50	50
SSR	378.7078	68.4316	196.0753

Table 6.1: Pooled and Regime-Specific Regressions for GDP Growth

$$F = \frac{(SSR_P - (SSR_1 + SSR_2))/k}{(SSR_1 + SSR_2)/(T_1 + T_2 - 2k)}$$

$$SSR_P = 378.7078, \qquad SSR_1 = 68.4316, \qquad SSR_2 = 196.0753,$$

$$T_1 = 50, \qquad T_2 = 50, \qquad k = 2.$$

$$SSR_1 + SSR_2 = 68.4316 + 196.0753 = 264.5069,$$

$$\text{Numerator} = \frac{378.7078 - 264.5069}{2} = \frac{114.2009}{2} = 57.10045,$$

$$\text{Denominator} = \frac{264.5069}{50 + 50 - 4} = \frac{264.5069}{96} = 2.75528,$$

$$F = \frac{57.10045}{2.75528} \approx 20.72.$$

Since calculated F-statistic from the chow test is greater than the critical F-value, we can reject the null hypothesis and conclude the presence of structural break at the year 1971.

6.2 Structural Breaks with *unknown* breakpoint

The data provided contain annual observations on India's GDP at 2011–12 constant prices and GDP per capita at 2011–12 constant prices for the period 1950–2024. Using these two series, we

examine whether there are any significant structural breaks in the growth pattern of the Indian economy over this period. We begin by applying the **Bai–Perron multiple breakpoint test** to each series separately as described below:

- Identify the number and approximate location of structural breakpoints in the GDP (2011–12 base year) series.
- Identify the number and approximate location of structural breakpoints in the GDP per capita (2011–12 base year) series.
- Report the estimated break years and briefly interpret their economic relevance in the context of major policy changes or historical events in India.

Model Specification. Let y_t denote the natural logarithm of GDP (or GDP per capita) at time t, and let t be a deterministic time trend. The regression model allowing for multiple structural breaks in both intercept and slope is specified as:

$$y_t = \mu_j + \beta_j t + \varepsilon_t, \quad t = T_{j-1} + 1, \ldots, T_j, \quad j = 1, 2, \ldots, m+1, \tag{6.6}$$

where μ_j and β_j are regime-specific intercepts and slopes, respectively, and ε_t is a zero-mean disturbance term. The breakpoints $T_1, T_2, \ldots, T_m$ are endogenously determined.

Hypotheses. The Bai–Perron methodology tests the null hypothesis of no structural breaks against the alternative of m breaks:

$$\begin{aligned} &H_0 : \mu_1 = \mu_2 = \cdots = \mu_{m+1}, \quad \beta_1 = \beta_2 = \cdots = \beta_{m+1}, \\ &H_1 : \exists\, j \text{ such that } (\mu_j, \beta_j) \neq (\mu_{j+1}, \beta_{j+1}), \end{aligned}$$

In other words, the null states that there is no structural break and the alternate states presence of at least one regime change in the intercept and/or slope over the sample period.

Testing Procedure. We employ the sequential F-test procedure proposed by Bai and Perron (1998, 2003) to determine the optimal number of structural breaks. The algorithm tests the null hypothesis of no break ($m = 0$) against the alternative of one or more breaks, and proceeds sequentially to higher numbers of breaks until the null is not rejected. The test is implemented with a trimming parameter of 0.20, implying that each regime must contain at least 20% of the total sample observations.

Maximum Number of Breaks and Regimes. Given that the sample spans $T = 75$ annual observations (1950–2024), a 20% trimming requirement implies that each regime must contain at least

$$0.20 \times 75 = 15 \text{ observations (years)}.$$

Hence, the maximum feasible number of breaks is 4 which corresponds to a maximum of 5 regimes.

Testing results. Table 6.2 reports the sequential $F(m|m-1)$ statistics and the corresponding Bai–Perron critical values at the 1%, 5% and 10% significance levels.

Table 6.2: Sequential Test for Multiple Structural Breaks in log(GDP)

Test	Statistic	1% Crit. Value	5% Crit. Value	10% Crit. Value
$F(1\|0)$	20.68	11.94	8.22	6.72
$F(2\|1)$	18.45	13.61	9.71	8.13
$F(3\|2)$	12.70	14.31	10.66	9.07
$F(4\|3)$	36.42	14.80	11.34	9.66

Decision rule. The null hypothesis of no structural break ($H_0 : m = 0$) is rejected in favour of at least one break, since $F(1|0) = 20.68 > 11.94$ (1% critical value). Testing sequentially, $F(2|1) = 18.45$ also exceeds the 1% critical value, indicating a second break. At the 1% significance level, the sequential F-tests fails to reject the null of two breaks against three. Hence, the Bai–Perron sequential procedure identifies two significant structural breaks in the log-GDP series.

Number of structural breaks. Based on these results:

- At the 1% significance level, the sequential test detects a maximum of two structural breaks.
- At the 5% significance level, up to four breaks are statistically significant.

Given the trade-off between statistical significance and model parsimony, we proceed with the specification allowing for between two and four breakpoints, corresponding to three to five distinct growth regimes.

The estimate break points are **1964-65** and **2005-06** if 2 break points are considered and **1964-65**, **1978-79** and **2005-06** if 3 break points are considered.

Table 6.3: Estimated Growth Regimes in India's Real GDP (2011–12 Constant Prices)

Regime	**Period**	**Duration (Years)**	**Avg. g (%)**	**Avg. Per Capita g (%)**
1	1950–1964	15	4.08	2.04
2	1965–2005	41	4.67	2.50
3	2006–2024	19	6.48	5.10

```
clear all
import delimited "IndiaGDP-Allbaseyears.csv", clear
foreach var of varlist gdp*{
        gen ln‘var’=ln(‘var’)
}

encode year, gen(time)
tsset time

tsline lngdp1112

**** trimming 20% i.e. each segment should be atleast 20% of total T or ~ 15
   years *****

xtbreak lngdp1112 time
xtbreak test lngdp1112 time, hypothesis(1) trim(0.20)
```

```
xtbreak test lngdp1112 time, breaks(2) hypothesis(1) trim(0.20)
xtbreak test lngdp1112 time, breaks(3) hypothesis(1) trim(0.20)

xtbreak lngdppc1112 time
xtbreak test lngdppc1112 time, hypothesis(1) trim(0.20)
xtbreak test lngdppc1112 time, breaks(2) hypothesis(1) trim(0.20)
xtbreak test lngdppc1112 time, breaks(3) hypothesis(1) trim(0.20)

**** trimming 15% i.e. each segment should be atleast 15% of total T or ~ 11
    years *****

xtbreak lngdp1112 time
xtbreak test lngdp1112 time, hypothesis(1) trim(0.15)
xtbreak test lngdp1112 time, breaks(2) hypothesis(1) trim(0.15)
xtbreak test lngdp1112 time, breaks(3) hypothesis(1) trim(0.15)

xtbreak lngdppc1112 time
xtbreak test lngdppc1112 time, hypothesis(1) trim(0.15)
xtbreak test lngdppc1112 time, breaks(2) hypothesis(1) trim(0.15)
xtbreak test lngdppc1112 time, breaks(3) hypothesis(1) trim(0.15)
```

Listing 6.3: Simulating GDP Growth with Structural Breaks

Chapter 7

Vector Autoregressions

7.1 Inflation and Unemployment

The relationship between inflation and unemployment has long been a central topic in macroeconomics, commonly captured by the concept of the *Phillips Curve*. Originally proposed by A.W. Phillips (1958), the Phillips Curve describes an inverse relationship between the rate of inflation and the rate of unemployment: when unemployment is low, inflation tends to rise, and when unemployment is high, inflation tends to fall. This relationship suggests a short-run trade-off faced by policymakers between stabilizing prices and promoting employment. Over time, however, economists have debated the stability of this trade-off, particularly after the experience of stagflation in the 1970s, which challenged the notion of a consistent inverse relationship. Modern interpretations often incorporate expectations, leading to the expectations-augmented Phillips Curve, which emphasizes that the long-run relationship between inflation and unemployment may be neutral once inflation expectations adjust.

```
** unit root test **
dfuller unrate, drift lags(1) // reject null; unrate is stationary around a
   constant mean //
dfuller unrate, noconstant lags(1)
dfuller unrate, trend lags(1)

dfuller inflation, drift lags(1) // reject null; inflation is stationary
   around a constant mean //
dfuller inflation, noconstant lags(1)
dfuller inflation, trend lags(1)


*** var lag selection **

varsoc unrate inflation // optimal lag length is 3 using AIC //
```

Listing 7.1: Stationarity Tests and Lag Selection

7.1.1 Unit Root Test

- **Unrate**

Table 7.1: Augmented Dickey–Fuller test results

Variable	Model	ADF stat	1% crit	5% crit	p-value
Unrate	No constant	-0.851	-2.592	-1.950	–
	Constant (drift)	-2.802	-2.350	-1.654	0.0029
	Trend	-2.740	-4.019	-3.442	0.2197
Inflation	No constant	-1.174	-2.592	-1.950	–
	Constant (drift)	-2.301	-2.350	-1.655	0.0113
	Trend	-2.435	-4.020	-3.442	0.3614

  - No constant: ADF = -0.851, 5% critical = -1.950.
    Since $-0.851 > -1.950$, do **not** reject H_0 at 5% (nonstationary).
  - Constant (drift): ADF = -2.802, 5% critical = -1.654.
    Since $-2.802 < -1.654$, **reject** H_0 at 5% (stationary around a constant). (reported p-value = 0.0029)
  - Trend: ADF = -2.740, 5% critical = -3.442.
    Since $-2.740 > -3.442$, do **not** reject H_0 at 5% (not trend-stationary).

- **Inflation**
  - No constant: ADF = -1.174, 5% critical = -1.950.
    Since $-1.174 > -1.950$, do **not** reject H_0 at 5% (nonstationary).
  - Constant (drift): ADF = -2.301, 5% critical = -1.655.
    Since $-2.301 < -1.655$, **reject** H_0 at 5% (stationary around a constant). (reported p-value = 0.0113)
  - Trend: ADF = -2.435, 5% critical = -3.442.
    Since $-2.435 > -3.442$, do **not** reject H_0 at 5% (not trend-stationary).

Both *inflation* and *unemployment* are stationary around a constant.

7.1.2 Lag Order Selection Criteria

To determine the appropriate lag length for the VAR model including the variables *unrate* and *inflation*, the lag-order selection statistics were computed using the `varsoc` command in Stata. The sample covers the period from 1961Q2 to 2000Q4 with 159 observations. The results are presented in Table 7.2.

Table 7.2: VAR Lag Order Selection Criteria

Lag	LL	LR	df	p-value	FPE	AIC	HQIC	SBIC
0	-438.414				0.872882	5.5398	5.5555	5.5784
1	-35.406	806.02	4	0.000	0.005771	0.5208	0.5679	0.6366
2	3.858	78.528	4	0.000	0.003704	0.0773	0.1556	**0.2703**
3	10.5538	**13.392**	4	0.010	**0.003580**	**0.0433**	**0.1531**	0.3136
4	13.0203	4.9331	4	0.294	0.003651	0.0626	0.2037	0.4101

The information criteria (Final Prediction Error, Akaike Information Criterion, and Hannan–Quinn Information Criterion) all reach their minimum values at lag 3. Although the Schwarz Bayesian

Information Criterion (SBIC) selects a shorter lag length of 2, the majority of the criteria—particularly AIC, HQIC, and FPE—suggest a lag order of three.

Based on the Akaike Information Criterion (AIC) and supporting evidence from FPE and HQIC, the optimal lag length for the VAR model is 3.

7.1.3 Phillips Curve Estimation - VAR(3)

The bivariate VAR(3) in scalar equations is

$$\begin{aligned}\text{unrate}_t = \alpha_1 &+ \beta_{11,1}\,\text{unrate}_{t-1} + \beta_{11,2}\,\text{unrate}_{t-2} + \beta_{11,3}\,\text{unrate}_{t-3} \\ &+ \beta_{12,1}\,\text{inflation}_{t-1} + \beta_{12,2}\,\text{inflation}_{t-2} + \beta_{12,3}\,\text{inflation}_{t-3} + e_{1t}, \qquad (7.1)\end{aligned}$$

$$\begin{aligned}\text{inflation}_t = \alpha_2 &+ \beta_{21,1}\,\text{unrate}_{t-1} + \beta_{21,2}\,\text{unrate}_{t-2} + \beta_{21,3}\,\text{unrate}_{t-3} \\ &+ \beta_{22,1}\,\text{inflation}_{t-1} + \beta_{22,2}\,\text{inflation}_{t-2} + \beta_{22,3}\,\text{inflation}_{t-3} + e_{2t}, \qquad (7.2)\end{aligned}$$

where $\{e_t\} = (e_{1t}, e_{2t})'$ is the reduced-form error vector.

7.1.4 Vector Autoregression (VAR(3)) Estimation Results

A bivariate Vector Autoregression (VAR) model with three lags was estimated for *unrate* and *inflation* over the sample period 1961Q1–2000Q4. The model was estimated using 160 quarterly observations. The overall fit statistics suggest that the VAR(3) provides a good fit to the data.

Table 7.3: Vector Autoregression Results: Unemployment Rate and Inflation (1961Q1–2000Q4)

	Coefficient	**Std. Err.**	**z**	**P>z**	**[95% Conf. Interval]**
Equation: unrate					
L1.unrate	1.567205	0.076961	20.36	0.000	[1.416364, 1.718046]
L2.unrate	-0.691751	0.134117	-5.16	0.000	[-0.954616, -0.428886]
L3.unrate	0.078435	0.075944	1.03	0.302	[-0.070413, 0.227284]
L1.inflation	0.153515	0.073620	2.09	0.037	[0.009223, 0.297807]
L2.inflation	-0.121067	0.089903	-1.35	0.178	[-0.297273, 0.055139]
L3.inflation	0.117145	0.077036	1.52	0.128	[-0.033843, 0.268134]
cons	0.121848	0.079681	1.53	0.126	[-0.034325, 0.278020]
Equation: inflation					
L1.unrate	-0.295985	0.080101	-3.70	0.000	[-0.452980, -0.138991]
L2.unrate	0.368962	0.139588	2.64	0.008	[0.095374, 0.642549]
L3.unrate	-0.115111	0.079042	-1.46	0.145	[-0.270031, 0.039809]
L1.inflation	0.676385	0.076623	8.83	0.000	[0.526207, 0.826563]
L2.inflation	0.120046	0.093570	1.28	0.200	[-0.063348, 0.303440]
L3.inflation	0.199405	0.080179	2.49	0.013	[0.042258, 0.356553]
cons	0.255973	0.082932	3.09	0.002	[0.093430, 0.418516]

- The coefficients on the first and second lags of both *unrate* and *inflation* are statistically significant in several cases, indicating that past values of these variables have important effects on their current values. Both series display clear evidence of persistence.
- In the **unrate equation**, the first lag of unemployment (`L1.unrate`) is large and highly significant (1.567, $p < 0.01$), confirming strong persistence in the unemployment rate. The

second lag (`L2.unrate`) is negative and significant (-0.692, $p < 0.01$), suggesting partial mean reversion after accounting for the first lag. Among the inflation lags, only the first lag (`L1.inflation`) is marginally significant (0.154, $p = 0.037$), indicating that higher inflation in the previous quarter slightly increases the current unemployment rate. The other inflation lags are not statistically significant.

- In the **inflation equation**, inflation shows strong own dynamics. The first lag of inflation (`L1.inflation`) is positive and highly significant (0.676, $p < 0.01$), and the third lag (`L3.inflation`) is also positive and significant (0.199, $p = 0.013$), pointing to persistence in inflation over multiple quarters. Regarding unemployment, the first lag (`L1.unrate`) has a negative and significant coefficient (-0.296, $p < 0.01$)), meaning that higher unemployment in the previous quarter tends to lower current inflation—consistent with a short-run Phillips curve effect. The second lag (`L2.unrate`) is positive and significant (0.369, $p < 0.01$), suggesting that the disinflationary impact of unemployment may reverse after about half a year.
- Overall, both the unemployment rate and inflation exhibit strong inertia. The results show a dynamic interaction between the two variables: inflation affects unemployment positively with a short lag, while unemployment tends to reduce inflation in the near term. These patterns are consistent with a lagged Phillips curve relationship where real and nominal dynamics adjust over time.

7.1.5 Granger Causality

In the VAR(3) framework:

- **Unemployment (unrate) Granger-causes Inflation** if the past values of unrate help predict inflation, beyond the information contained in past values of inflation.
- **Inflation Granger-causes Unemployment** if the past values of inflation help predict unrate, beyond the information contained in past values of unrate.

The Granger causality tests in Table 7.4 indicate statistically significant *bidirectional predictive relationships* between inflation and unemployment. The null hypothesis that inflation does not Granger-cause unemployment is strongly rejected ($\chi^2 = 23.48$, $p < 0.01$), implying that past values of inflation contain significant information for predicting the unemployment rate. Similarly, the null hypothesis that unemployment does not Granger-cause inflation is also rejected ($\chi^2 = 19.03$, $p < 0.01$), suggesting that lagged unemployment contributes to forecasting inflation dynamics. Overall, these results point to a two-way Granger causal relationship between inflation and unemployment over the sample period 1961Q1–2000Q4, consistent with the dynamic interactions implied by the estimated VAR(3) model.

```
**** Estimate var (3) *******
var unrate inflation, lags(1/3)
varstable

**** granger causality test ***
vargranger

**** impulse response function *****
irf create myirf, step(12) set(myirf_results) replace
```

```
irf graph oirf, impulse(unrate) response(inflation)
irf graph oirf, impulse(unrate) response(unrate)
irf graph oirf, impulse(inflation) response(unrate)
irf graph oirf, impulse(inflation) response(inflation)

**** forecast error variance decomposition ****
irf table fevd, impulse(unrate inflation) response(unrate inflation)
```

Listing 7.2: VAR Estimation

Table 7.4: Granger Causality Wald Tests: Inflation and Unemployment

Equation	**Excluded Variable**	**Chi-Square**	**df**	**p-value**
Unemployment Rate	Inflation	23.483	1	0.000
Unemployment Rate	All	23.483	1	0.000
Inflation	Unemployment Rate	19.032	1	0.006
Inflation	All	19.032	1	0.006

Notes: Null hypothesis — the excluded variable does not Granger-cause the dependent variable. Rejection of the null (small p-value) indicates predictive causality in the Granger sense. Results are based on the VAR(3) model estimated over 1961Q1–2000Q4.

7.1.6 Impulse Response Functions

We begin with the response of the unemployment rate to inflation shocks. The response starts close to zero, indicating that inflation shocks have little immediate impact on unemployment. Over subsequent quarters, unemployment rate rises steadily, reaching around 0.3–0.35 after roughly 10–12 quarters. Rather than oscillating, the increase is smooth and persistent, suggesting that inflation shocks have a lasting positive effect on unemployment.

The VAR results do not support the traditional short-run Phillips curve, which predicts that higher inflation reduces unemployment. Instead, the findings point to a reverse or long-run Phillips curve effect, where inflation shocks are followed by rising unemployment. This outcome may reflect policy tightening in response to inflation, upward shifts in inflation expectations, or cost-push shocks that raise prices while depressing output and employment. Overall, the results suggest that inflationary pressures ultimately lead to weaker labor market conditions rather than lower unemployment.adjustment frictions in wages and prices.

Next, the IRF for the response of inflation to an unemployment rate shock provides clear evidence consistent with the Phillips curve. A positive shock to unemployment leads to a noticeable and persistent decline in inflation, indicating that higher unemployment exerts downward pressure on prices. The response is negative from the outset and remains below zero throughout the forecast horizon, showing little sign of returning to baseline. This suggests that unemployment shocks have a lasting disinflationary effect rather than a short-lived one. Overall, the IRF implies a stable and persistent inverse relationship between unemployment and inflation, in line with the traditional Phillips curve intuition.

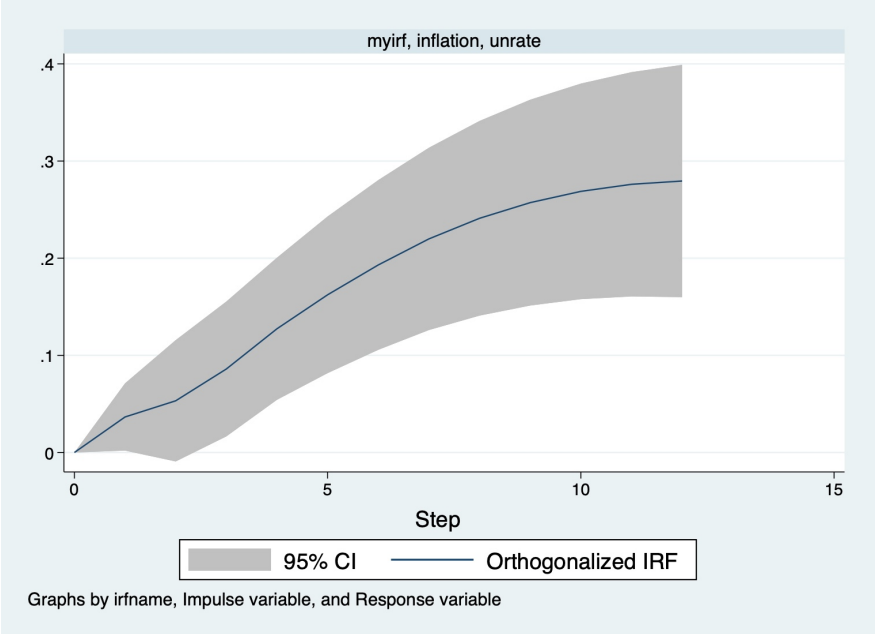

Figure 7.1: Impulse: *inflation*; Response: *unrate*

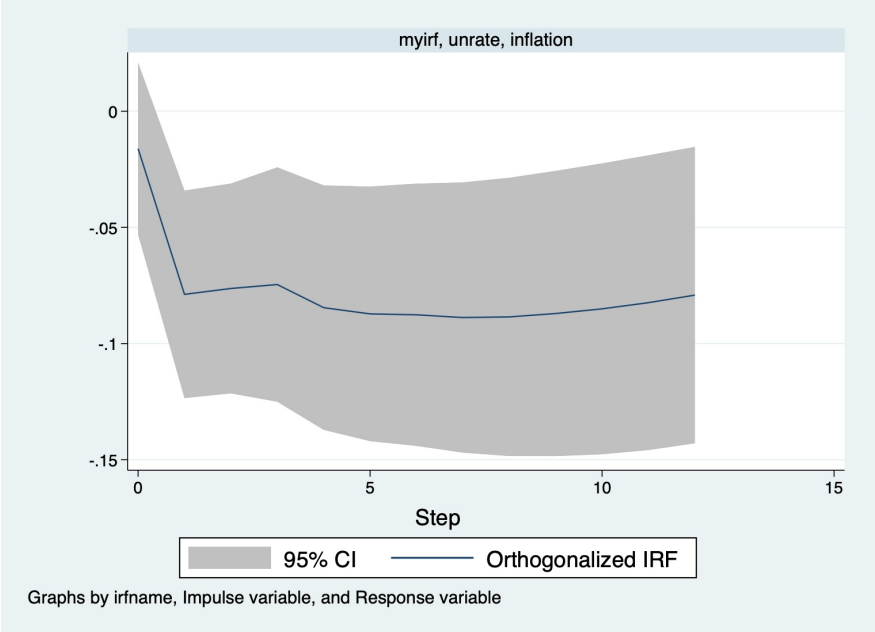

Figure 7.2: Impulse: *unrate*; Response: *inflation*

7.1.7 Forecast Error Variance Decomposition (FEVD)

The forecast error variance decomposition (FEVD) measures the proportion of the forecast variance of each endogenous variable that can be attributed to shocks in each variable within the system. Table 7.5 reports the FEVD for *unrate* and *inflation* at selected forecast horizons (steps).

Table 7.5: Forecast Error Variance Decomposition (FEVD)

Step	Response	Shock from unrate	Shock from inf	Lower (95%)	Upper (95%)
0	unrate	1.000	0.000	–	–
	inflation	0.000	1.000	–	–
1–3	unrate	0.988	0.111	-0.002	0.223
	inflation	0.012	0.889	0.777	1.001
4–6	unrate	0.926	0.173	0.006	0.339
	inflation	0.074	0.827	0.661	0.995
7–9	unrate	0.798	0.224	0.019	0.429
	inflation	0.202	0.776	0.571	0.981
10–12	unrate	0.658	0.264	0.030	0.498
	inflation	0.342	0.736	0.502	0.970

Interpretation

- At short horizons (1–3 quarters), the forecast variance of **unrate** is almost entirely explained by its own shocks (around 99% or more), with inflation contributing less than 1% to its variability.
- As the horizon lengthens, the role of inflation shocks in explaining unemployment variability rises steadily. By 6 quarters, inflation shocks account for roughly 17% of the forecast variance, reaching about 34% by the 12th quarter. This pattern indicates that inflation shocks exert an increasingly important and persistent influence on unemployment over time.
- For **inflation**, own shocks remain the dominant source of variation at all horizons—about 99% in the first quarter and still around 74% by the 12th quarter. However, the share of variance explained by unemployment shocks grows gradually from nearly zero initially to roughly 26% at longer horizons, reflecting a moderate feedback from labor market conditions to inflation dynamics.
- Overall, the FEVD results suggest that both series are primarily driven by their own innovations in the short run, but cross-variable effects become more pronounced as the forecast horizon increases. This reinforces the dynamic interaction between inflation and unemployment consistent with the Phillips curve mechanism—where each variable increasingly influences the other over time.

Chapter 8

Vector Error Correction Model

8.1 Long-Run Relationships and the Quantity Theory of Money

Figure 8.1 plots the logarithms of the money stock (lnm2sl), the price level (lncpiaucsl), and real output (lngdpc1) from 1960 onward. All three series exhibit strong and persistent upward trends over time. This pattern is consistent with the core prediction of the Quantity Theory of Money (QTM), summarized by the identity

$$M_t V_t = P_t Y_t, \tag{8.1}$$

which in logarithms becomes

$$\ln M_t = \ln P_t + \ln Y_t - \ln V_t. \tag{8.2}$$

If velocity is constant or at least stationary in the long run, the QTM implies that the long-run movements in money, prices, and output should be tightly linked. The similar trending behavior of the three series in the figure suggests that they may share a common stochastic trend. In other words, although the variables may deviate from one another in the short run, they should not drift apart indefinitely. This provides a natural motivation for testing for cointegration among them.

Because each of the variables appears to be integrated of order one, estimating long-run relationships in levels would ordinarily lead to spurious regressions. However, the QTM provides a theoretical basis for expecting a stable equilibrium condition of the form

$$\ln M_t - \ln P_t - \ln Y_t = c + \varepsilon_t, \tag{8.3}$$

where ε_t is stationary if the variables are cointegrated. Finding evidence of cointegration would therefore be consistent with the idea that velocity does not wander without bound and that the money market and real economy are connected through a long-run equilibrium condition.

Given cointegration, a Vector Error Correction Model (VECM) is the appropriate framework for modelling the joint dynamics of money, prices, and output. The VECM incorporates short-run adjustments while enforcing the long-run cointegrating relationship implied by the QTM. The error-correction term captures deviations from the long-run monetary equilibrium, and the adjustment coefficients measure how each variable responds when the equilibrium is temporarily violated.

This framework allows us to analyse both the long-run monetary relationship suggested by theory and the short-run dynamics that bring the system back to equilibrium.

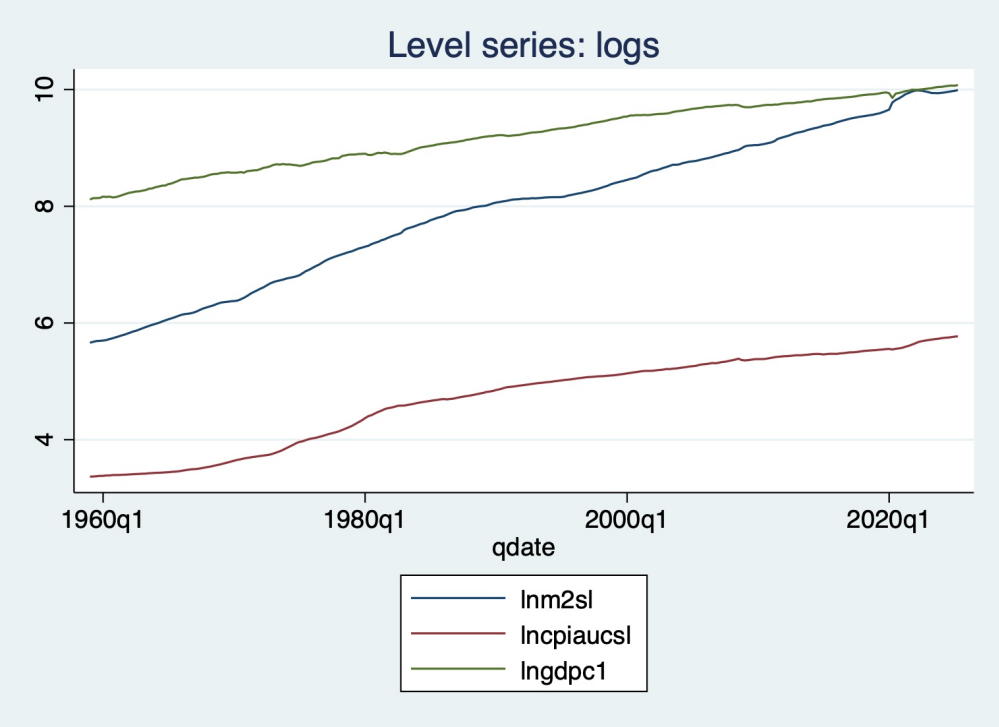

Figure 8.1: Trend in M2, CPI and GDP Per Capita

```
// Preliminary analysis - line plot //

tsline lnm2 lncpi lngdp, legend(cols(1)) title("Level series: logs") // seems
    non stationary //

* ---------------------------
* b) Stationarity testing: ADF tests (levels and first differences)
*    - Test each variable in levels (with and without trend)
*    - If nonstationary in levels but stationary in first differences => I(1)
* ---------------------------

varsoc lnm2 lncpi lngdp // choose 2 lags //

** unit root test **
dfuller lnm2, drift lags(2)
dfuller lnm2, noconstant lags(2)
dfuller lnm2, trend lags(2)

dfuller lncpi, drift lags(2)
dfuller lncpi, noconstant lags(2)
dfuller lncpi, trend lags(2)

dfuller lngdp, drift lags(2)
dfuller lngdp, noconstant lags(2)
dfuller lngdp, trend lags(2)
```

```
// Based on dfuller test, cannot reject null , so non-stationary//

** unit root test **
dfuller d.lnm2, drift lags(2)
dfuller d.lnm2, noconstant lags(2)
dfuller d.lnm2, trend lags(2)

dfuller d.lncpi, drift lags(2)
dfuller d.lncpi, noconstant lags(2)
dfuller d.lncpi, trend lags(2)

dfuller d.lngdp, drift lags(2)
dfuller d.lngdp, noconstant lags(2)
dfuller d.lngdp, trend lags(2)

// Based on dfuller test, reject null for trend and drift so all three are I
   (1) //


*** Johansen cointegration test - choose rank ***

vecrank lnm2 lncpi lngdp, trend(constant) lags(1) max // 1 rank significant //
vecrank lnm2 lncpi lngdp, trend(constant) lags(2) max // 0 rank //
vecrank lnm2 lncpi lngdp, trend(constant) lags(3) max // 0 rank //
vecrank lnm2 lncpi lngdp, trend(constant) lags(4) max // 0 rank //


vec lnm2 lncpi lngdp, rank(1) lags(1) trend(constant)
```

Listing 8.1: VECM

8.2 Estimation - Quantity Theory of Money

$$\Delta \mathbf{y}_t = \alpha \beta' \mathbf{y}_{t-1} + \Gamma_1 \Delta \mathbf{y}_{t-1} + \mathbf{c} + \varepsilon_t,$$

where

$$\mathbf{y}_t = \begin{bmatrix} \ln M2_t \\ \ln CPI_t \\ \ln GDP_t \end{bmatrix}.$$

Cointegrating Relation

Johansen normalization (normalized on $\ln M2$) gives the cointegrating vector

$$\beta' \mathbf{y}_t = \ln M2_t - 0.973 \ln CPI_t - 1.277 \ln GDP_t + 9.706.$$

Thus the error–correction term is

$$EC_{t-1} = \ln M2_{t-1} - 0.973 \ln CPI_{t-1} - 1.277 \ln GDP_{t-1} + 9.706.$$

Error–Correction Model

Money Equation:

$$\Delta \ln M2_t = 0.00953\, EC_{t-1} + 0.00366.$$

Inflation Equation:

$$\Delta \ln CPI_t = 0.00772\, EC_{t-1} - 0.00121.$$

GDP Equation:

$$\Delta \ln GDP_t = 0.00796\, EC_{t-1} - 0.00320.$$

(Only the error–correction coefficients are shown because of one lag)

Cointegration Interpretation

The trace test indicates one cointegrating vector linking money, prices, and output. The estimated long-run relation is

$$\ln M2_t = 0.973\, \ln CPI_t + 1.277\, \ln GDP_t - 9.706.$$

Higher price levels or higher real output require higher money balances to maintain long-run equilibrium. Departures from this equilibrium are captured by the error–correction term.

Adjustment Toward Long-Run Equilibrium

The speed-of-adjustment coefficients are:

$$\alpha_{\text{M2}} = 0.00953 \qquad (p = 0.001)$$

$$\alpha_{\text{CPI}} = 0.00772 \qquad (p = 0.000)$$

$$\alpha_{\text{GDP}} = 0.00796 \qquad (p = 0.003)$$

All three adjustment coefficients are positive and statistically significant.

Interpretation:
All variables respond to disequilibrium in the long-run money–price–output relationship:

- When money is "too high" relative to CPI and GDP, all three variables adjust upward in the following period.
- The system restores equilibrium jointly; no variable is weakly exogenous.
- Adjustment speeds are small (0.7–1.0%), meaning equilibrium correction is gradual.

Economic Meaning

- The long-run relationship is consistent with a money-demand interpretation: higher prices and higher output require higher money balances.
- All three variables adjust significantly to the long-run error–correction term.
- Money, inflation, and output are jointly endogenous and influence each other in restoring long-run balance.