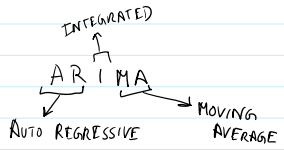
ARIMA

July 31, 2025 12:10 PM



(1) Step 1

AR I MA

Integrated — Create a stationary series (Mean & Variance constant over time)

Its important to ensure that
∵ these models assume that mean & variance doesn't change over time
☆ Note covariance could change with lag/stride

To make the series stationary, we do

Differencing (d)

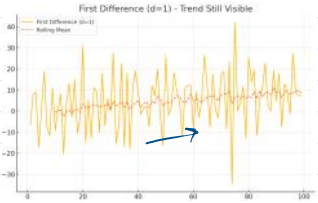
Linear trends: $d=1$, $y'_t = y_t - y_{t-1}$ OR $y'_t = y_t - y_{t-s}$ (Seasonality)

Quadratic OR polynomial trends: $d = 0 > 1$, $\nabla^d y_t = (1-B)^d y_t$ — Backshift Op. $By_t = y_{t-1}$

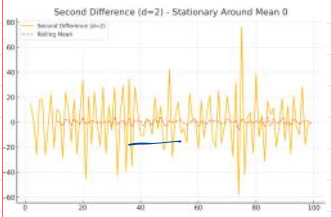
e.g. $d=2$

$$y''_t = (1-B)^2 y_t = y_t + B^2 y_t - 2By_t$$
$$= y_t + B.y_{t-1} - 2y_{t-1}$$
$$= y_t + y_{t-2} - 2y_{t-1}$$

$d=1$, $y'_t = y_t - y_{t-1}$



$d=2$, $y''_t = y_t - 2y_{t-1} + y_{t-2}$




```
window = 10  # rolling window size

rolling_mean_original = ts.rolling(window).mean()
rolling_mean_diff1 = diff1.rolling(window).mean()
rolling_mean_diff2 = diff2.rolling(window).mean()

plt.figure(figsize=(10,6))
plt.plot(ts, label="Original Series (Quadratic Trend)")
plt.plot(rolling_mean_original, label="Rolling Mean", linestyle="--")
plt.legend()
plt.title("Original Series (Non-Stationary)")
plt.show()

plt.figure(figsize=(10,6))
plt.plot(diff1, label="First Difference (d=1)")
plt.plot(rolling_mean_diff1, label="Rolling Mean", linestyle="--")
plt.legend()
plt.title("First Difference (d=1) - Trend Still Visible")
plt.show()

plt.figure(figsize=(10,6))
plt.plot(diff2, label="Second Difference (d=2)")
plt.plot(rolling_mean_diff2, label="Rolling Mean", linestyle="--")
plt.legend()
plt.title("Second Difference (d=2) - Stationary Around Mean 0")
plt.show()
```

(2) Step 2

AR I MA

Auto Regressive (p)

AR Models - Based on past values, look back recursively, Congruence

Use $y_t, y_{t-1}, y_{t-2} \cdots y_{t-p}$ to predict y_t

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_3 y_{t-3} + \cdots + \phi_p y_{t-p} + e^t$$

- y_t: Target
- c: Intercept
- $\phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p}$: Auto Regressive coeff.
  - Captures the dependence on specific Lag
  - essentially weight
  - $\phi_1 = 0 \Rightarrow$ No correlation
  - $\phi_1 = 1 \Rightarrow$ Non stationary Random walk
  - $\phi_1 < 0 \Rightarrow$ Oscillation
  - $|\phi_1| < 1 \Rightarrow$ Stationary & value indicates corr.
- e^t: Error
  - $\bar{e}_t = 0$
  - $Var(e_t) = \sigma^2 = \text{const.}$

$\Rightarrow$ As an operator,

$$\phi(B)\, y_t = c + e^t$$

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 \cdots - \phi_p B^p$$

e.g. For an AR(1) series $p=1$, $y_t = c + \phi_1 y_{t-1} + e^t$

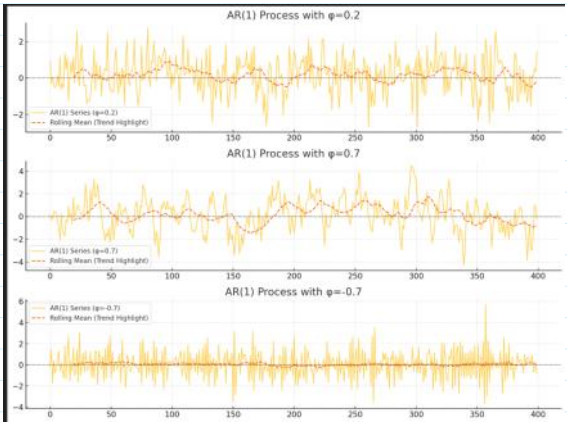
```
n = 400
phi_values = [0.2, 0.7, -0.7]
window = 20

fig, axes = plt.subplots(len(phi_values), 1, figsize=(12, 9))

for i, phi in enumerate(phi_values):
    series = simulate_ar([phi], n=n)
    rolling_mean = pd.Series(series).rolling(window=window).mean()

    axes[i].plot(series, alpha=0.6, label=f"AR(1) Series (ϕ={phi})")
    axes[i].plot(rolling_mean, linestyle="--", label="Rolling Mean
(Trend Highlight)")
    axes[i].axhline(0, color="black", linewidth=1, linestyle=":")
    axes[i].legend()
    axes[i].set_title(f"AR(1) Process with ϕ=[phi]")

plt.tight_layout()
plt.show()
```



$\phi_1 = 0.2$

Fluctuations around mean, but quick series more or less around mean (for stationary $\phi_1 = 0$)

$\phi_1 = 0.7$

Fluctuations last longer, mean slowly drifts ($\phi_1 = 1 \Rightarrow$ non stationary)

$\phi_1 = -0.7$

Too many oscillations around the mean sharp fluctuations ($\phi_1 < 0 \Rightarrow$ oscillations)

(3) Step 3

ARIMA

Moving Average (q)

Uses errors in the past to predict future

Also called as dependence on past shocks.

Short memory

$$y_t = C + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}$$

$\theta_1 > 0$ +ve errors

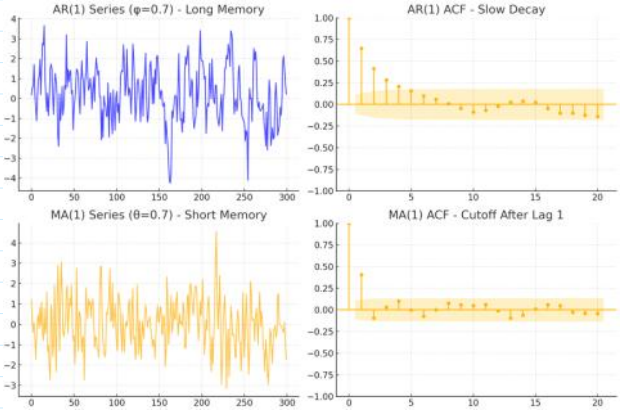
$\theta_1 < 0$ oscillating errors

$\Rightarrow$ As an operator,

$$y_t = C + \Theta(B)\epsilon_t$$

$$\Theta B = 1 + \theta_1 B + \theta_2 B^2 + \dots \theta_q B^q$$

eg Comparing an AR(1) & MA(1) series




```
from statsmodels.graphics.tsaplots import plot_acf

series_ar = simulate_ar([0.7], n=300)
series_ma = simulate_ma([0.7], n=300)

fig, axes = plt.subplots(2, 2, figsize=(12, 8))

axes[0,0].plot(series_ar, color="blue", alpha=0.7)
axes[0,0].set_title("AR(1) Series (ϕ=0.7) - Long Memory")

axes[1,0].plot(series_ma, color="orange", alpha=0.7)
axes[1,0].set_title("MA(1) Series (θ=0.7) - Short Memory")

plot_acf(series_ar, lags=20, ax=axes[0,1])
axes[0,1].set_title("AR(1) ACF - Slow Decay")

plot_acf(series_ma, lags=20, ax=axes[1,1])
axes[1,1].set_title("MA(1) ACF - Cutoff After Lag 1")

plt.tight_layout()
plt.show()
```

AR

Long memory

ACF is higher for multiple lags (1, 2, 3, 4, 5)

Any shocks will last longer

MA

Short memory

Only immediate lags show correlation (1)

Shocks don't last very long.

★ Where ACF is AutoCorrelationFunction

$$\rho_k = \frac{Cov(y_t, y_{t-k})}{Var(y_t)} \quad \text{eg} \quad \text{ACF at lag 1} = \frac{Cov(y_t, y_{t-1})}{Var(y_t)}$$

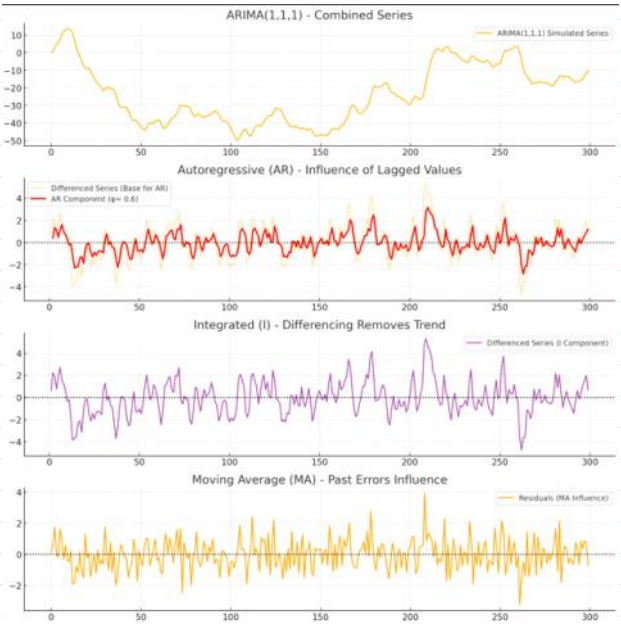
(4) Step 4

ARIMA AR + I + MA

$\Rightarrow$

$$\phi(B)\,(1-B)^d\,y_t = C + \Theta(B)\,\epsilon_t$$

- $\phi(B)$ — AR: Accounting for past lags (weighted feedback)
- $(1-B)^d$ — I: Making series stationary
- $\Theta(B)\,\epsilon_t$ — MA: Accounting for past errors




```
np.random.seed(42)
n = 300
phi = 0.6   # AR(1) coefficient
theta = 0.5  # MA(1) coefficient
c = 0
e = np.random.normal(0, 1, n+1)  # white noise
y = np.zeros(n+1)
for t in range(1, n+1):
    y[t] = y[t-1] + c + phi * (y[t-1] - y[t-2] if t>1 else 0) + e[t] + theta * e[t-1]

series_arima = pd.Series(y[1:])

diff_series = series_arima.diff().dropna()

from statsmodels.tsa.arima.model import ARIMA
model = ARIMA(series_arima, order=(1,1,1)).fit()
fitted = model.fittedvalues
residuals = model.resid

ar_component = phi * diff_series.shift(1)
fig, axes = plt.subplots(4, 1, figsize=(12, 12))

axes[0].plot(series_arima, label="ARIMA(1,1,1) Simulated Series")
axes[0].set_title("ARIMA(1,1,1) - Combined Series")
axes[0].legend()

axes[1].plot(diff_series, alpha=0.4, label="Differenced Series (Base for AR)")
axes[1].plot(ar_component, color="red", label=f"AR Component (ϕ= {phi})",
linewidth=2)
axes[1].axhline(0, color="black", linestyle=":")
axes[1].set_title("Autoregressive (AR) - Influence of Lagged Values")
axes[1].legend()

axes[2].plot(diff_series, color="purple", alpha=0.7, label="Differenced Series (I
Component)")
axes[2].axhline(0, color="black", linestyle=":")
axes[2].set_title("Integrated (I) - Differencing Removes Trend")
axes[2].legend()

axes[3].plot(residuals, color="orange", label="Residuals (MA Influence)")
axes[3].axhline(0, color="black", linestyle=":")
axes[3].set_title("Moving Average (MA) - Past Errors Influence")
axes[3].legend()

plt.tight_layout()
plt.show()
```