Forecasting with Autocorrelation
ARIMA Models

Nagiza F. Samatova, samatova@csc.ncsu.edu
Professor, Department of Computer Science
North Carolina State University

Senior Scientist, Computer Science & Mathematics Division
Oak Ridge National Laboratory

Basic Notation

Symbol	Definition
$\boldsymbol{t = 1, 2, 3, \ldots,}$	An index for the time period of interest; e.g., for a *daily* time period, $t = 1$ means day 1, $t = 2$ means day 2, etc.
$\boldsymbol{y_1, y_2, \ldots, y_T}$	A series of T values measure over T time periods; e.g., for the annual average stock price, y_1 denotes the price for year 1, y_2 denotes the price for year 2, etc.
$\boldsymbol{F_t}$ or $\boldsymbol{\hat{y}_t}$	The forecast value for time period t
$\boldsymbol{F_{t+k}}$ or $\boldsymbol{\widehat{y_{t+k}}}$	The k-step-ahead forecast when forecasting time is t; e.g., F_{t+1} is the forecast for time period $(t + 1)$ made during the time period t
$\boldsymbol{e_t = y_t - F_t}$	The forecast error for time period t

TS Parts: Systematic vs Non-systematic

TS Part	Definition	Detection	How to deal w/
Level	Average value of ts		
Trend	Long-term increase decrease in the data	lag.plot	De-trend via lag-1 differencing
Seasonality	Variations occurring during known periods of the year (monthly, quarterly, holidays)	lag.plot, Acf plots	De-seasonalize via lag-k differencing
Cycles	Other oscillating patterns about the trend (e.g., business or economic conditions)		
Auto-correlation	Correlation between neighboring points in ts	Acf, lag.plot	
Noise	Residuals after level, trend, seasonality, and cycles are removed	Normality tests	

Additive and Multiplicative TS Components

- A time series with additive components can be modeled as:

$$y_t = Level + Trend/Cycles + Seasonality + Noise$$

- A time series with multiplicative components is modeled as:

$$y_t = Level \times Trend/Cycles \times Seasonality \times Noise$$

- Forecasting methods attempt to isolate the systematic part and quantify the noise level.
  - The systematic part is used for generating point forecasts
  - The level of noise helps assess the uncertainty associated with the point forecasts

Global vs Local Patterns in TS Data

- **Global patterns** are the patterns in the TS data that extend throughout the TS period used for training and forecasting:
  - Trends: persistent upward or persistent downward trend with a non-changing slope

- **Local patterns** change over-time in a hard-to-predict manner

TS Data Analysis Methods

TS Data Analysis & Forecasting

Data-Driven

Data-driven methods are used when model assumptions are likely to be violated, or when the structure of time series changes over time.

- **Baseline**: average, naive, seasonal naive, drift
- **Differencing**
- **Smoothing**: moving average, exponential smoothing

Model-based

Training data is used to estimate model parameters, and then the model with these parameters is used to generate forecasts.

- **ARIMA**
- **Linear Regression**
- **Logistic Regression**
- **Neural Networks**

Data-driven vs. Model-Based Methods

- **Model-based methods are generally preferable for forecasting series with global patterns; they use all the data to estimate the global patterns.**
  - For a local pattern, a model would require specifying how and when the patterns change, which is usually impractical and often unknown.
  - Model-based methods such as neural networks, regression trees, etc. are also used for TS forecasting for **incorporating external information into forecasts**.

- **Data-driven methods are preferable for forecasting series with local patterns. Such methods "learn" patterns from the data, and their memory length can be set to best adapt to the rate of change in the series.**
  - Patterns that change quickly warrant a "short memory," whereas patterns that change slowly warrant a "long memory"

TS Data Analysis Methods

TS Data Analysis & Forecasting

Data-Driven

Data-driven methods are used when model assumptions are likely to be violated, or when the structure of time series changes over time.

- **Baseline**: average, naive, seasonal naive, drift
- **Differencing**
- **Smoothing**: moving average, exponential smoothing

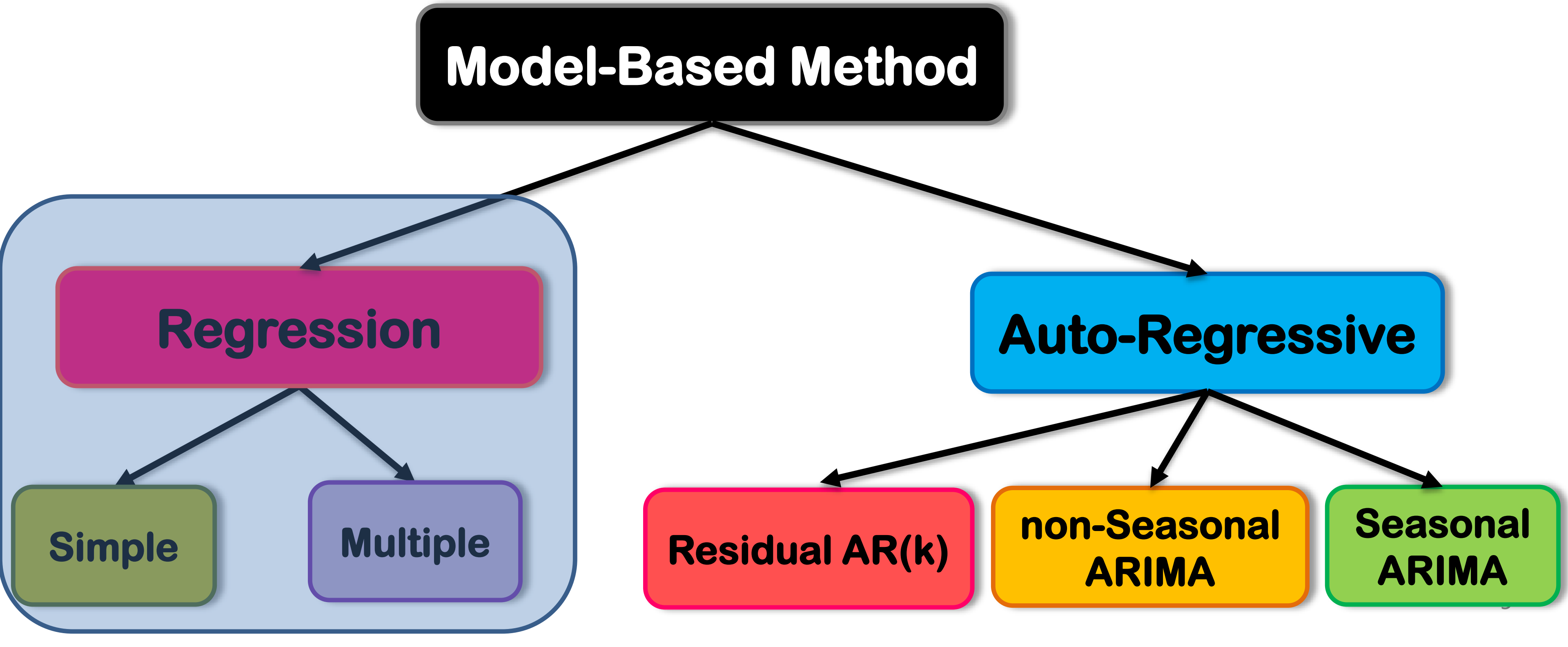
Model-based

Training data is used to estimate model parameters, and then the model with these parameters is used to generate forecasts.

- **ARIMA**
- **Linear Regression**
- **Logistic Regression**
- **Neural Networks**

Forecasting with Autocorrelation

ARIMA: AUTO-REGRESSIVE INTEGRATED MOVING AVERAGE



Autocorrelation & Auto-Regressive Models

- **When we use linear regression for time series forecasting, we are able to account for patterns such as trend and seasonality.**
- **However, ordinary regression models do NOT account for dependence between observations, which in cross-sectional data is assumed to be absent.**
- **Yet, in the time series context, *observations in neighboring periods tend to be correlated*.**
  - Such correlation, or autocorrelation, is informative and can help in improving forecasts.
- **How to best utilize autocorrelation information for improving forecasting?**

I. Improved Forecast = Forecast Series + Forecast Errors

- To improve forecasts by integrating autocorrelation: to construct a second-level forecasting model for the residuals
  1. Generate k-step-ahead forecast of the series F_{t+k} using a forecasting method
  2. Generate k-step-ahead forecast of the forecast error (e_{t+k}), using an autoregressive (AR) model.
  3. Improve the initial k-step-ahead forecast of the series by adjusting it according to its forecasted error:

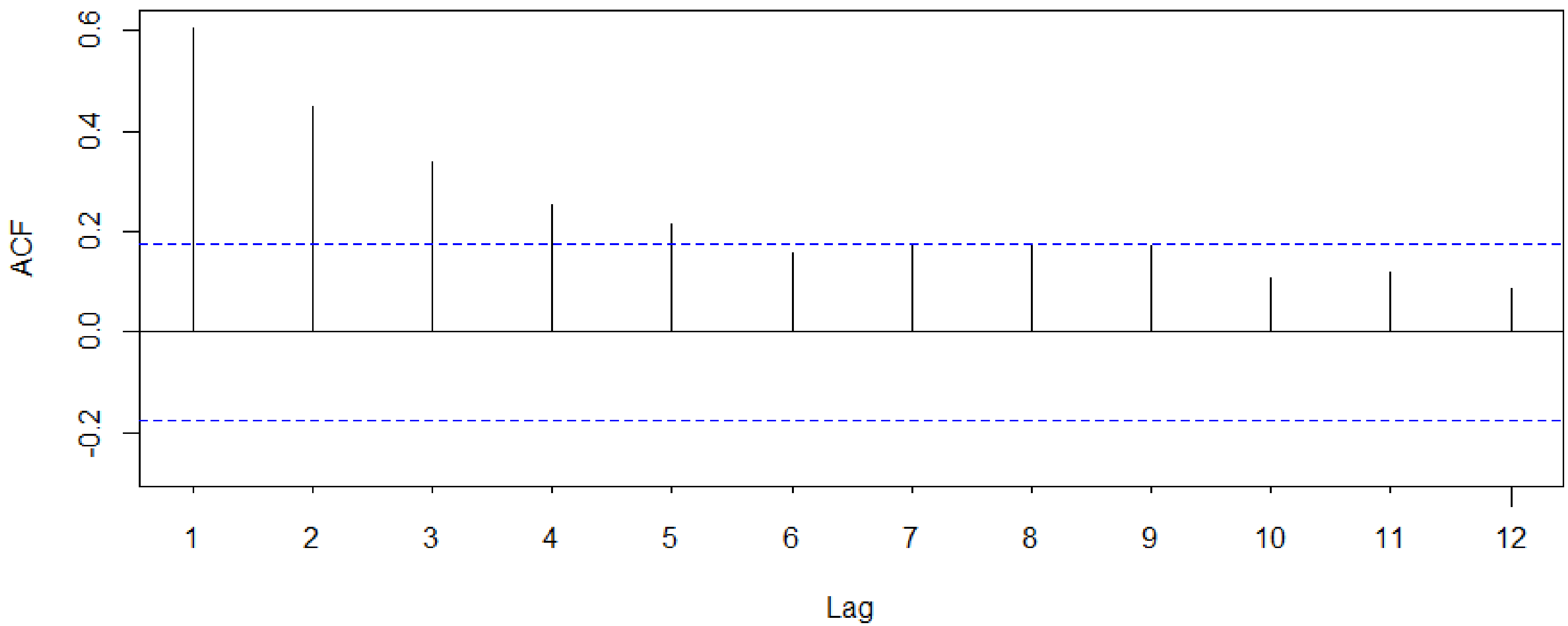
$$\boxed{F^*_{t+k} = F_{t+k} + e_{t+k}}$$

Modeling Residuals: Fitting AR(k) model

- **To fit an autoregressive (AR) model to the series of residuals,**
  1. **Examine the autocorrelations of the residual series:**
     - E.g., plot $e_t - e_{t-1}$, does it look linear? This will indicate a strong lag-1 autocorrelation (use Acf() function in R)
  2. **Choose the order of the AR model according to the lags in which autocorrelation appears.**
     - e.g. if autocorrelation exists at lag-1 and higher, it is sufficient to fit an AR(1) model of the form
       - $\boldsymbol{e_t = \beta_0 + \beta_1 e_{t-1} + \epsilon_t}$
       - where e_t denotes the residual (or **forecast error**) at time t

Example: Autocorrelation of Residuals

- **Although the autocorrelations appear large from lags 1 to 10 or so, it's likely that an AR(1) would capture all these relationships.**



- **The reason is that if neighboring values are correlated, the relationship can propagate to values that are two periods away, then three periods away, and so forth.**

II. Auto-Regressive Integrated Moving-Average

- **Among regression-type models that account for autocorrelation are autoregressive (AR) models, or the more general ARIMA models**
  - **ARIMA: Auto-Regressive Integrated Moving-Average**

$$\boxed{ARIMA\,(p,d,q) = AR(p) + I(d) + MA(q)}$$

Autoregressive of order p **Differencing of order d** **Moving Average of order q**

$$\boxed{\begin{aligned} y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_p y_{t-p} + \\ + e_t + \alpha_1 e_{t-1} + \alpha_2 e_{t-2} + \cdots + \alpha_q e_{t-q} \end{aligned}}$$

- Predictors include both **lagged values of y_t** and **lagged errors e_t**.
- Maximum Likelihood Estimation (MLE) is used to estimate the unknown parameters $\alpha's$ and $\beta's$.

$$AR(p):\; y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_p y_{t-p}$$

$$\mathrm{MA}(q):\; y_t = \alpha_0 + e_t + \alpha_1 e_{t-1} + \alpha_2 e_{t-2} + \cdots + \alpha_q e_{t-p}$$

Example: ARIMA (0,0,3) = MA(3)

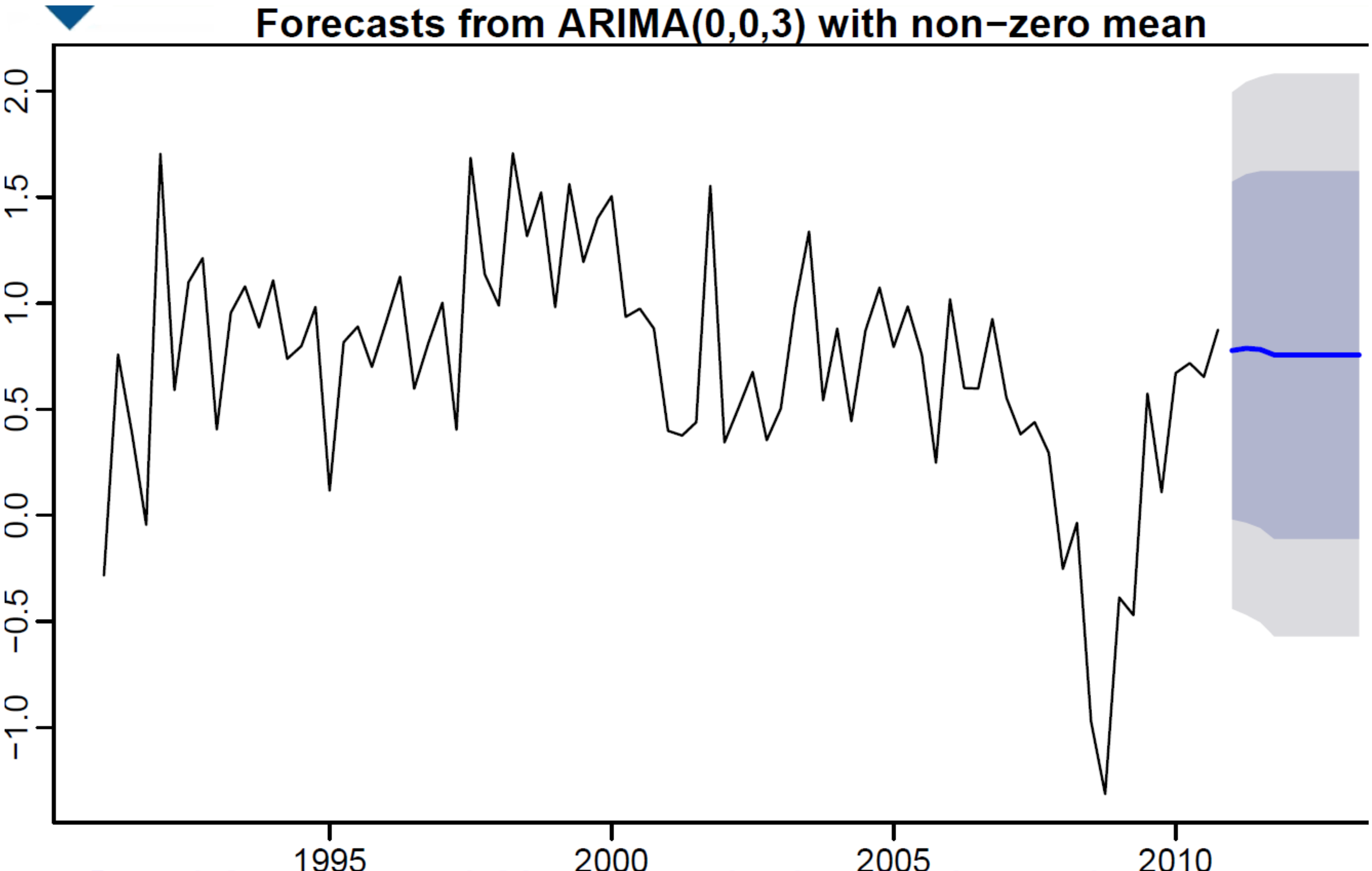
```
> fit <- auto.arima(usconsumption[,1], seasonal=FALSE)

ARIMA(0,0,3) with non-zero mean
Coefficients:
         ma1     ma2     ma3  intercept
      0.2542  0.2260  0.2695     0.7562
s.e.  0.0767  0.0779  0.0692     0.0844

sigma^2 estimated as 0.3856:  log likelihood=-154.73
AIC=319.46   AICc=319.84   BIC=334.96
```

ARIMA(0,0,3) or MA(3) model:

$$y_t = 0.756 + e_t + 0.254e_{t-1} + 0.226e_{t-2} + 0.269e_{t-3},$$

where e_t is white noise with standard deviation $0.62 = \sqrt{0.3856}$.

Example: Forecast with ARIMA (0,0,3)




```
plot(forecast(fit,h=10),include=80)
```

Parameters of the ARIMA Model

$$\boldsymbol{ARIMA\ (p, d, q) = AR(p) + I(d) + MA(q)}$$

Autoregressive of order p **Differencing of order d** **Moving Average of order q**

- Forecast variance and d
  - The higher the value of d, the more rapidly the prediction intervals increase in size.
  - For d = 0, the long-term forecast standard deviation will go to the standard deviation of the historical data.
- Cyclic behavior
  - For cyclic forecasts, p > 2 and some restrictions on coefficients are required.

How to Select Orders for ARIMA Parameters?

$$\boxed{ARIMA\,(p,d,q) = AR(p) + I(d) + MA(q)}$$

- Select d via unit root tests
- Select p, q via minimizing AICc

Steps in auto.arima() function in R:

Step 1: Select current model (with smallest AIC) from:
ARIMA$(2, d, 2)$
ARIMA$(0, d, 0)$
ARIMA$(1, d, 0)$
ARIMA$(0, d, 1)$

Step 2: Consider variations of current model:
- vary one of p, q, from current model by ± 1
- p, q both vary from current model by ± 1
- Include/exclude c from current model

Model with lowest AICc becomes current model.

Repeat Step 2 until no lower AICc can be found.

ARIMA Modeling Process

I. Plot the ts data:
- Identify unusual observations
- Understand patterns

II. Stabilize the variance:
Box-Cox transformations

III. Stabilize the mean & Check Stationarity:
Difference the data to stabilize the mean
till ts appears stationary
Use unit root tests to verify stationarity

IV. Find "best" ARIMA model:
Use auto.arima()

V.a Check residuals ≡ white noise:
Plot ACF of the residuals and
Do Portmanteau Test of residuals

V.b If residuals ≠ white noise:
Plot ACF of the differenced data
to find candidate models to fit.
Use AIC/BIC to choose a better model
Check V.a.

VI. Calculate forecasts

Example: Seasonally Adjusted Electrical Equipment

```
> fit <- auto.arima(eeadj)
> summary(fit)
Series: eeadj
ARIMA(3,1,1)

Coefficients:
         ar1     ar2     ar3      ma1
      0.0519  0.1191  0.3730  -0.4542
s.e.  0.1840  0.0888  0.0679   0.1993

sigma^2 estimated as 9.532:  log likelihood=-484.08
AIC=978.17   AICc=978.49   BIC=994.4
```

IV. Find "best" ARIMA model:
Use auto.arima()

```
Acf(residuals(fit))
Box.test(residuals(fit), lag=24,
   fitdf=4, type="Ljung")
```

V.a Check residuals ≡ white noise:
Plot ACF of the residuals and
Do Portmanteau Test of residuals

```
plot(forecast(fit))
```

VI. Calculate forecasts

Summary: ARMA & ARIMA (auto.arima(ts))

- **Among regression-type models that account for autocorrelation are autoregressive (AR) models, or the more general ARIMA models**
  - **ARIMA: Auto-Regressive Integrated Moving-Average**

$$\boxed{ARIMA\,(p, d, q) = AR(p) + I(d) + MA(q)}$$

Autoregressive of order p

Differencing of order d

Moving Average of order q

- Predictors include both **lagged values of y_t** and **lagged errors, e_t**.
- ARMA models can be used for a huge range of **stationary time series**.
- They model the **short-term dynamics**.
- An **ARMA** model applied to ***differenced*** data is an **ARIMA** model.

$$\boxed{\begin{aligned} y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_p y_{t-p} + \\ + e_t + \alpha_1 e_{t-1} + \alpha_2 e_{t-2} + \cdots + \alpha_q e_{t-q} \end{aligned}}$$

Acknowledgements

- Books
  - Free and online (otexts.com/fpp): Forecasting Principles & Practice by R. Hyndman, G. Athanasopoulos ← Excellent Book!!!
  - Practical Time Series Forecasting with R: A Hand-on Guide by Shmueli & Lichtendahl
- Packages
  - R: fpp (install.packages ("fpp", dependencies=TRUE))