Using Advanced Time Series Models

Martin Burger
R PROGRAMMING TUTOR
www.r-tutorials.com

Advanced Models

Identifying patterns

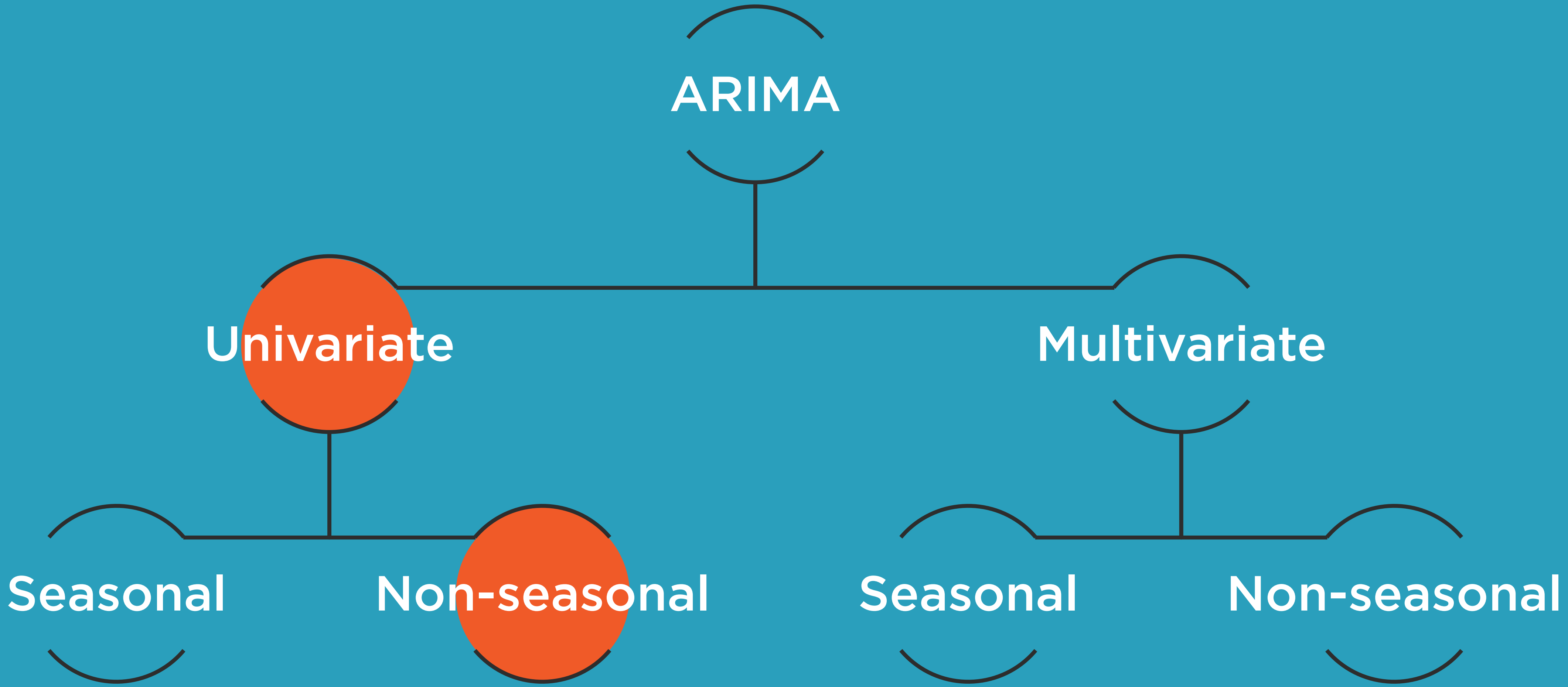
- Capturing patterns
- Putting info into equations

Library ‘forecast’

ARIMA model

Exponential smoothing

Summary


ARIMA
Univariate
Multivariate
Seasonal
Non-seasonal
Seasonal
Non-seasonal

Demo

Theory of univariate ARIMA models

Example of usage

A framework developed by George Box and Gwilym Jenkins

Autoregressive Integrated Moving Average

AR	• Autoregressive term – ‘p’
I	• Integration/ differencing – ‘d’
MA	• Moving average – ‘q’

Stationarity

ARIMA(p, d, q)

Non-stationary time series gets differenced (‘d’) before ‘p’ and ‘q’ get specified

ARMA(p, q)

With stationary time series the autoregressive (‘p’) and moving average (‘q’) terms get ordered without differencing

ARIMA Functions in R

arima()

- R Base
- Parameters need to be calculated with functions acf() and pacf()

auto.arima()

- Library(forecast)
- Calculates the parameters and does the differencing automatically

Variations of the Model

AR(1) – ARIMA(1, 0, 0)

Autoregressive model ('p' only)

MA(1) – ARIMA(0, 0, 1)

Moving average model ('q' only)

What Do the Parameters Do?

p

Summation of lags – AR

$Y_t = c + \varphi_1 * y_{t-1} + \varphi_2 * y_{t-2} + \cdots + \varphi_p * y_{t-p}$

d

Degree of differencing – I

q

Summation of forecast error terms – MA

$Y_t = c + \vartheta_1 * e_{t-1} + \vartheta_2 * e_{t-2} + \cdots + \vartheta_q * e_{t-q}$

How to Calculate an AR Model

Coefficients:

	ar1	mean
	1.1246	1547.3859
s.e.	0.0903	136.8501

$$Y_t = c + \varphi_1 * y_{t-1}$$

Coefficients:

	ar1	ar2	ar3	ar4	mean
	1.1246	-0.7174	0.2634	-0.2543	1547.3859
s.e.	0.0903	0.1367	0.1361	0.0897	136.8501

$$Y_t = c + \varphi_1 * y_{t-1} + \varphi_2 * y_{t-2} +$$
$$+ \varphi_3 * y_{t-3} + \varphi_4 * y_{t-4}$$

How to Calculate an ARMA Model

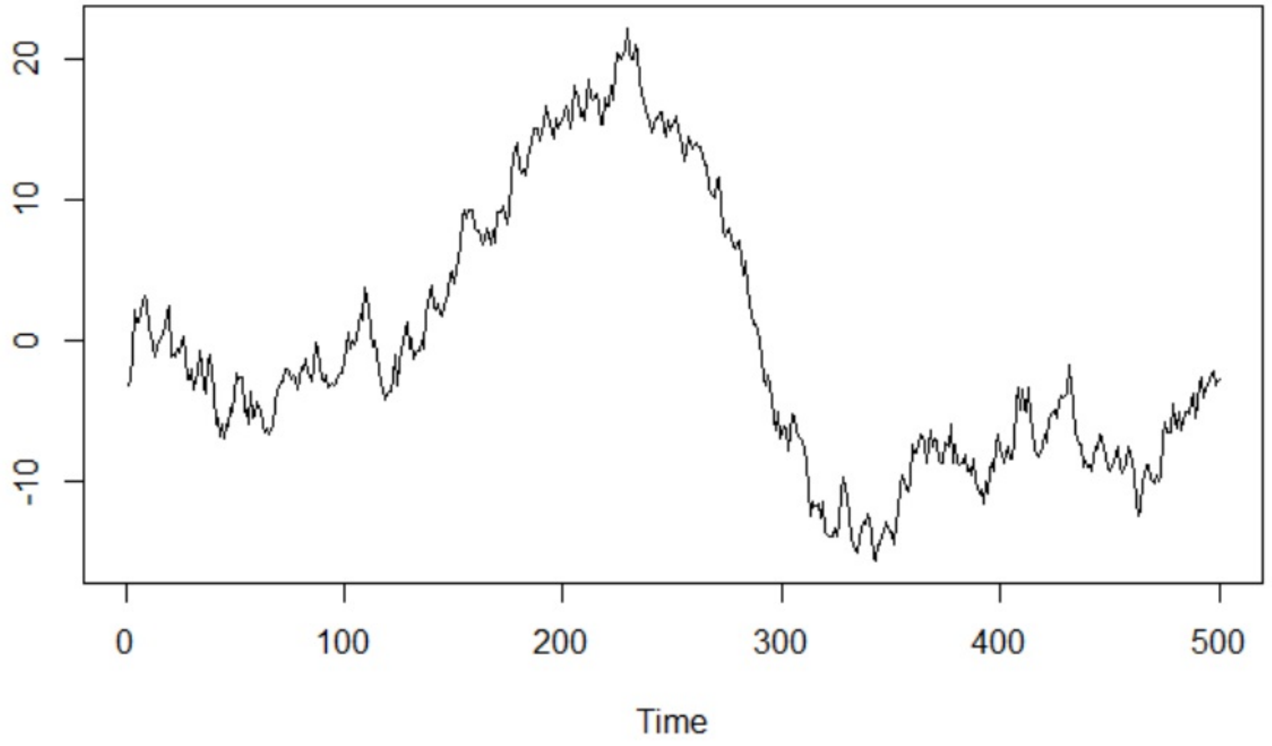
```
Coefficients:
          ar1      ma1       mean
       1.3421  -0.2027  1544.4039
s.e.   0.0984   0.1261   131.9242
```

$$Y_t = c + \varphi_1 * y_{t-1} + \vartheta_1 * e_{t-1}$$

ARIMA(0, 1, 0)

Drift: $c = Y_t - Y_{t-1}$

No drift: $Y_t = Y_{t-1}$



Demo

ARIMA models in practice

Dataset: ‘lynx’

Terms to be familiar with:

- Stationarity
- Autoregression

What to Expect from ‘lynx’?

p

Autocorrelation is clear

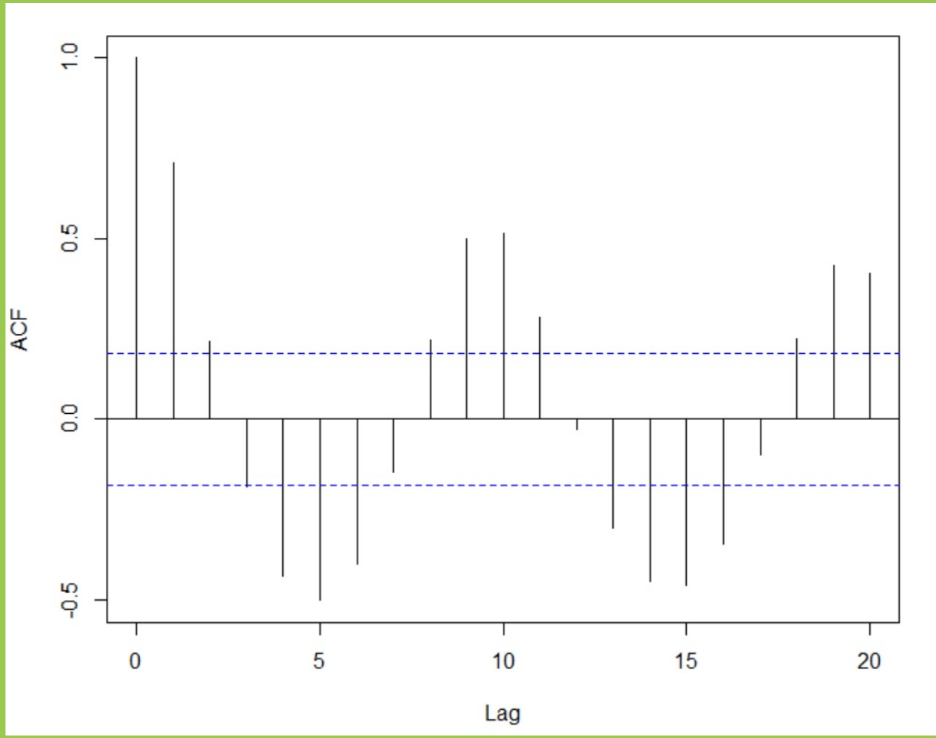
d

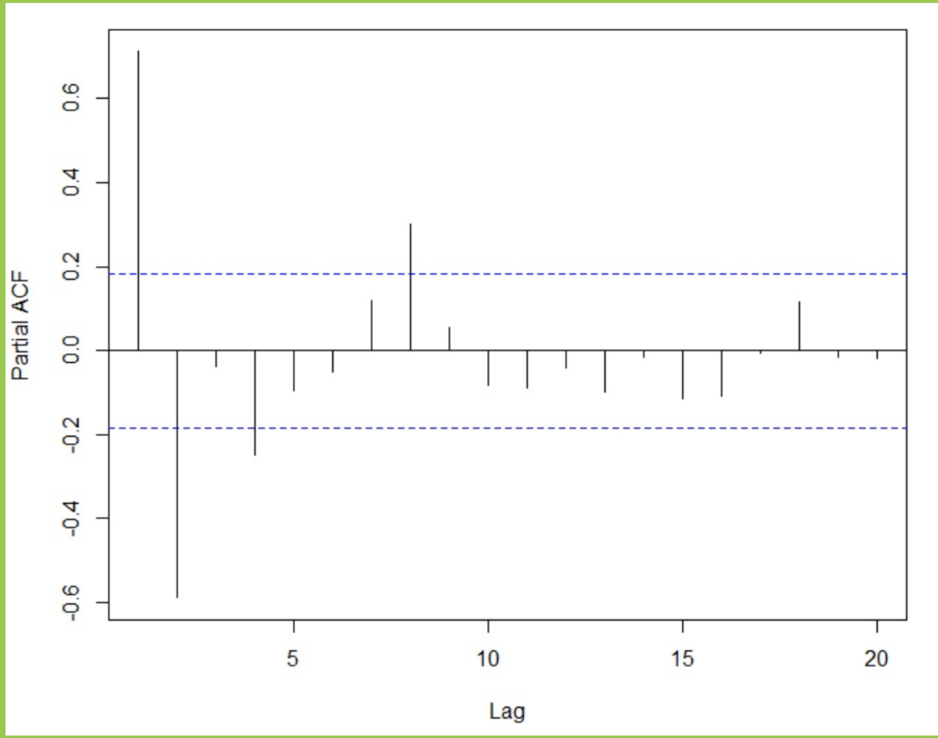
It might be stationary

q

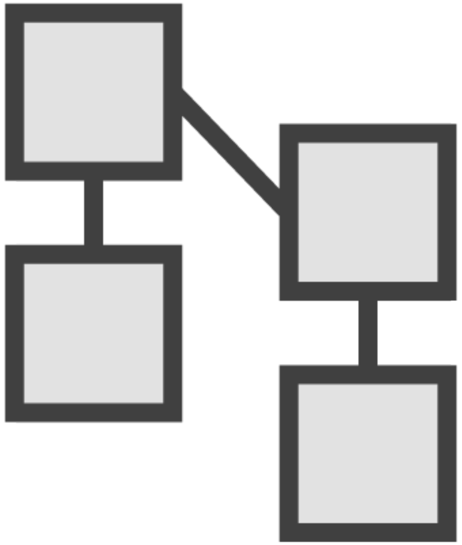
There might be forecasting errors

ACF and PACF Plots of ‘lynx’





How to Get the Best Model for the Data?



Several model options

Find the best suited one

- Smallest number of orders
- Smallest information criteria (AIC, AICc, BIC)
- Zero mean or non-zero mean

Calculating the ARIMA Model

```
Series: lynx
ARIMA(2,0,2) with non-zero mean

Coefficients:
         ar1      ar2      ma1      ma2       mean
      1.3421  -0.6738  -0.2027  -0.2564  1544.4039
s.e.  0.0984   0.0801   0.1261   0.1097   131.9242

sigma^2 estimated as 761965:  log likelihood=-932.08
AIC=1876.17   AICc=1876.95   BIC=1892.58
```

$$
\begin{aligned}
Y_t = {} & 1554.4 + 1.3421 * Y_{t-1} + \\
& + (-0.6738) * Y_{t-2} + \\
& + (-0.2027) * e_{t-1} + \\
& + (-0.2564) * e_{t-2}
\end{aligned}
$$

Choosing a model is up to the analyst

- Dataset
- Surrounding factors

Literature of your field

- Best practices
- Conventions

Demo

Exponential smoothing

Time series data modeling

R implementation

Parameters of Exponential Smoothing

Error

Additive or multiplicative (if $x \in R^+$)

Trend

Non-present, additive or multiplicative

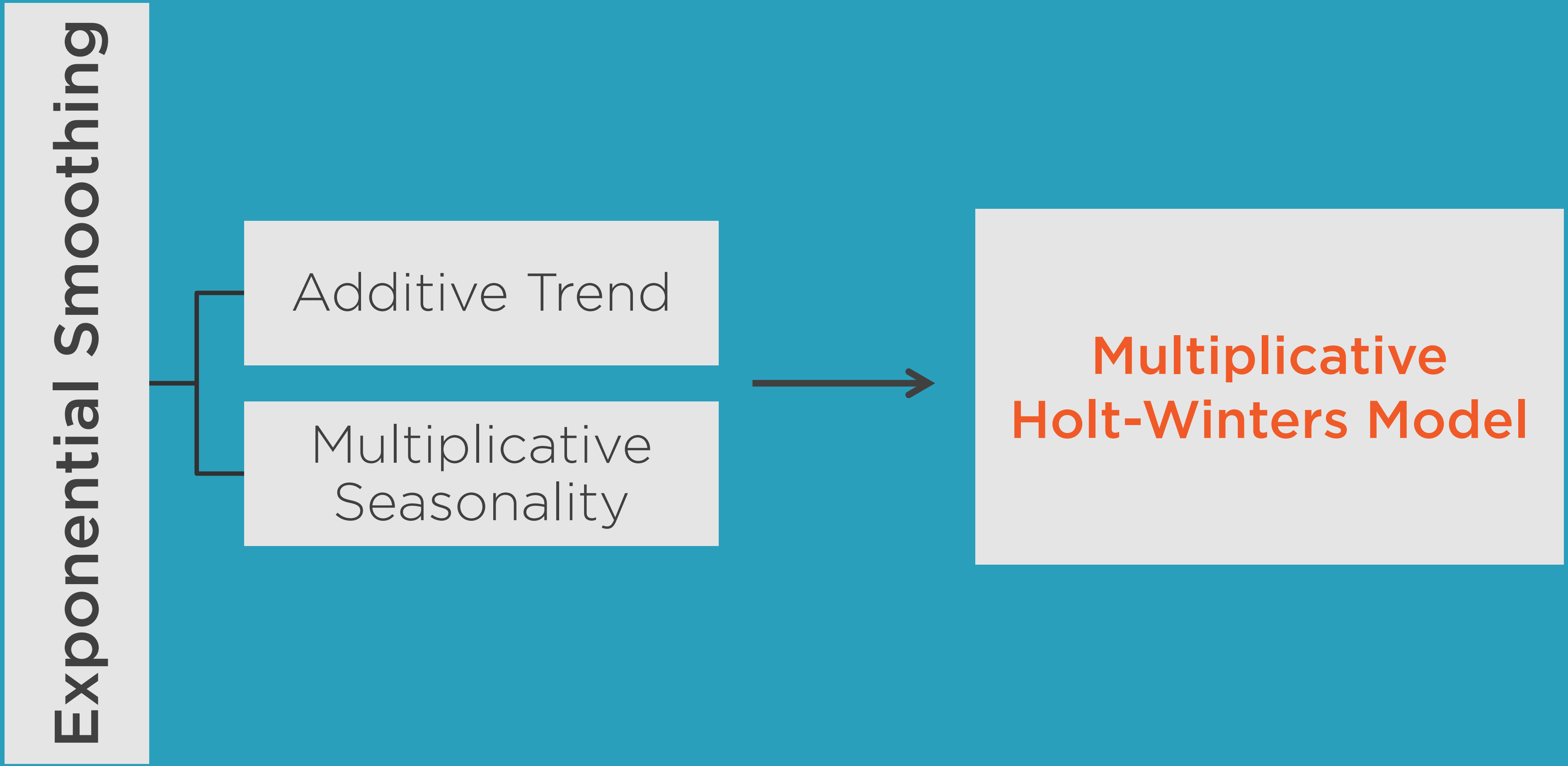
Seasonality

Non-present, additive or multiplicative

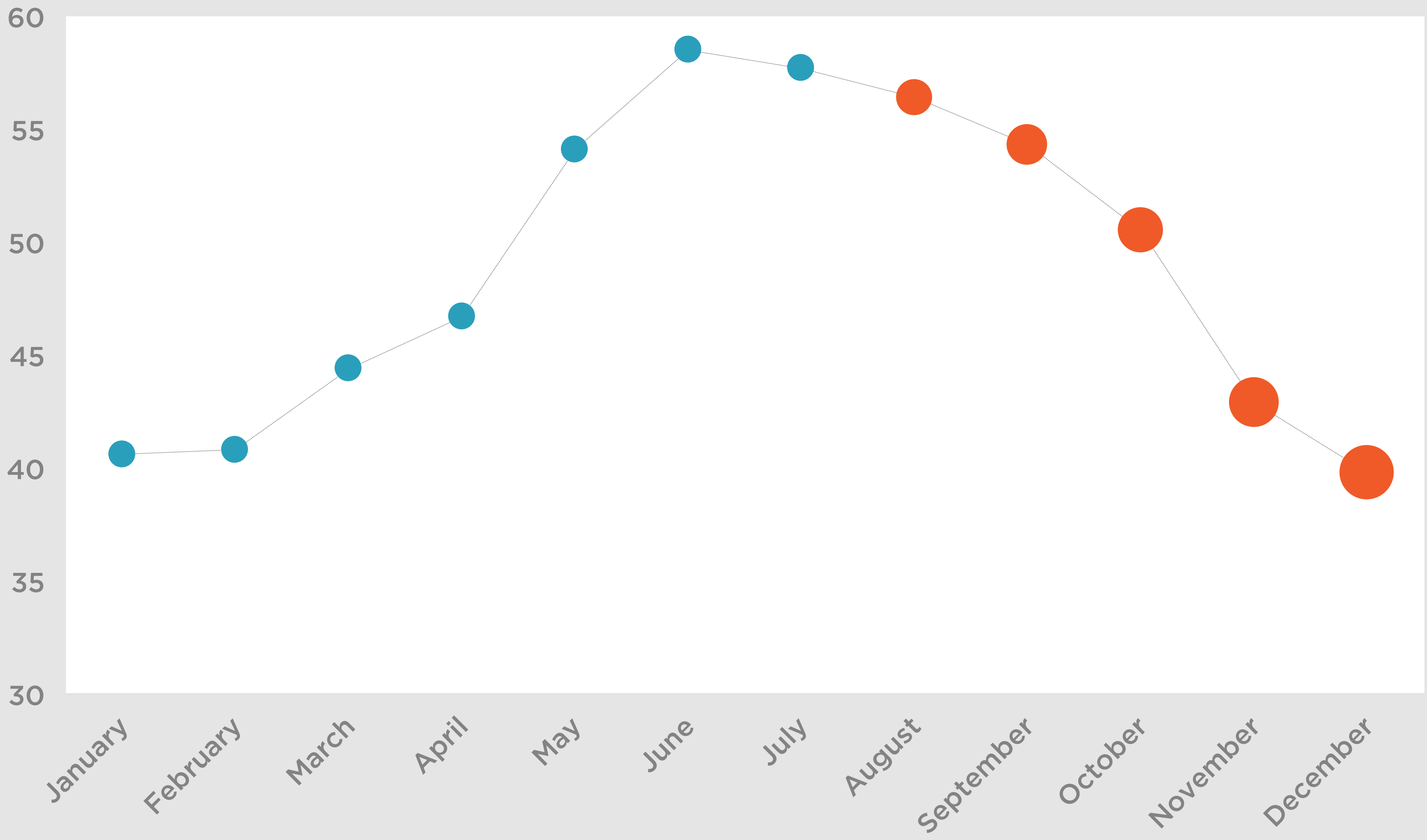
Parameter Operators

A	Summation of components
M	Multiplication of components
N	Components are omitted

Example Model



What Exponential Smoothing Does

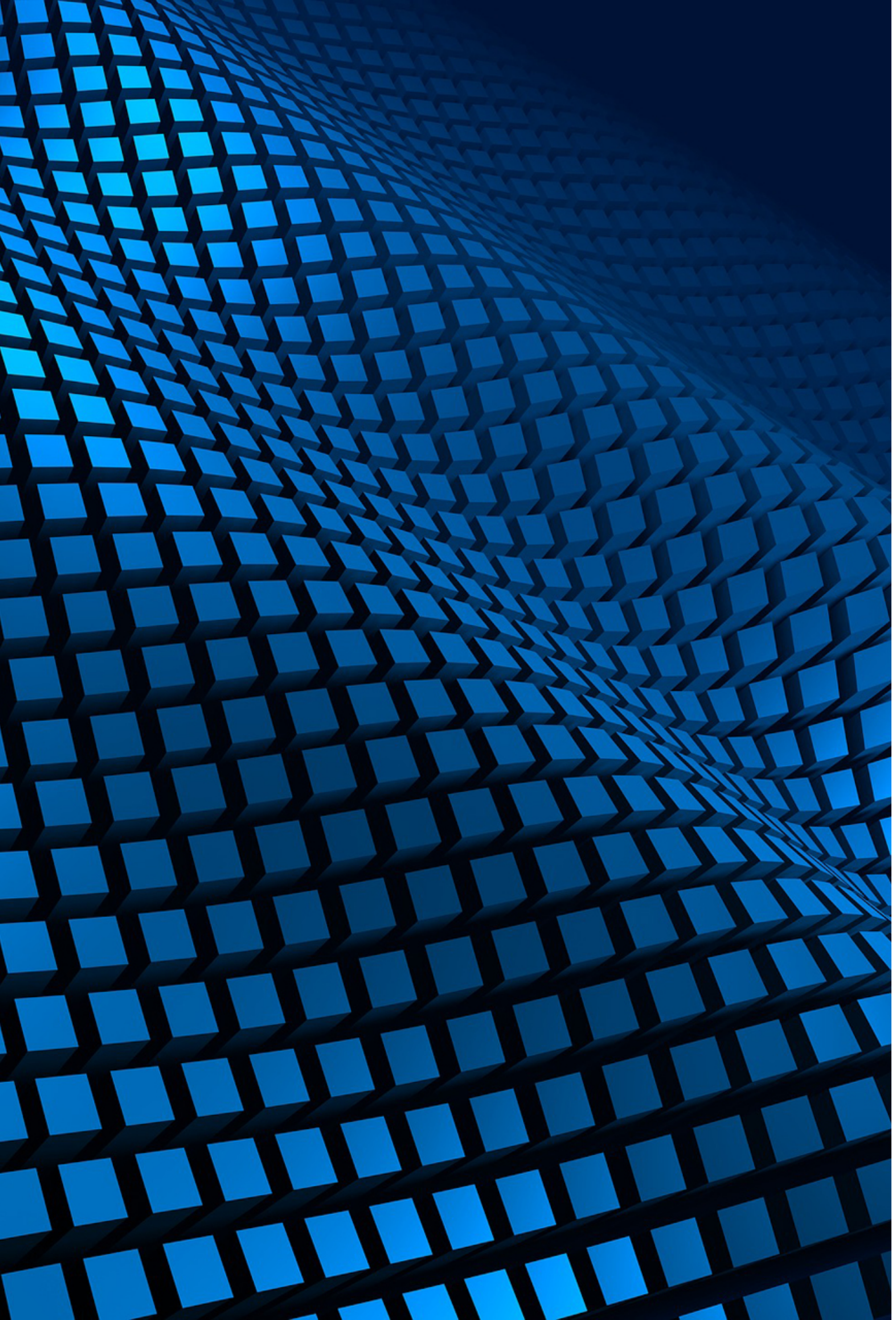


R Implementation

Library(forecast)

Function ets()

Automatically selects the optimal model for the data



Smoothing coefficients

Manage weighting

- Recent data → reactive model (~1)
- Whole data → smooth curves (~0)

Coefficients:

- α: Initial level
- β: Trend
- γ: Seasonality

Exponential Smoothing Functions in R

Function ses()
Simple exponential smoothing

Function holt()
Trend methods

Function hw()
Holt-Winters seasonal methods

Function ets()
Selects the optimal model

The 'model=' Argument of 'ets()'

Z

Auto selection

A

Additive model

M

Multiplicative model

N

Non-present (except error)

Further Arguments of ‘ets()’

‘alpha=’

‘beta=’

‘gamma=’

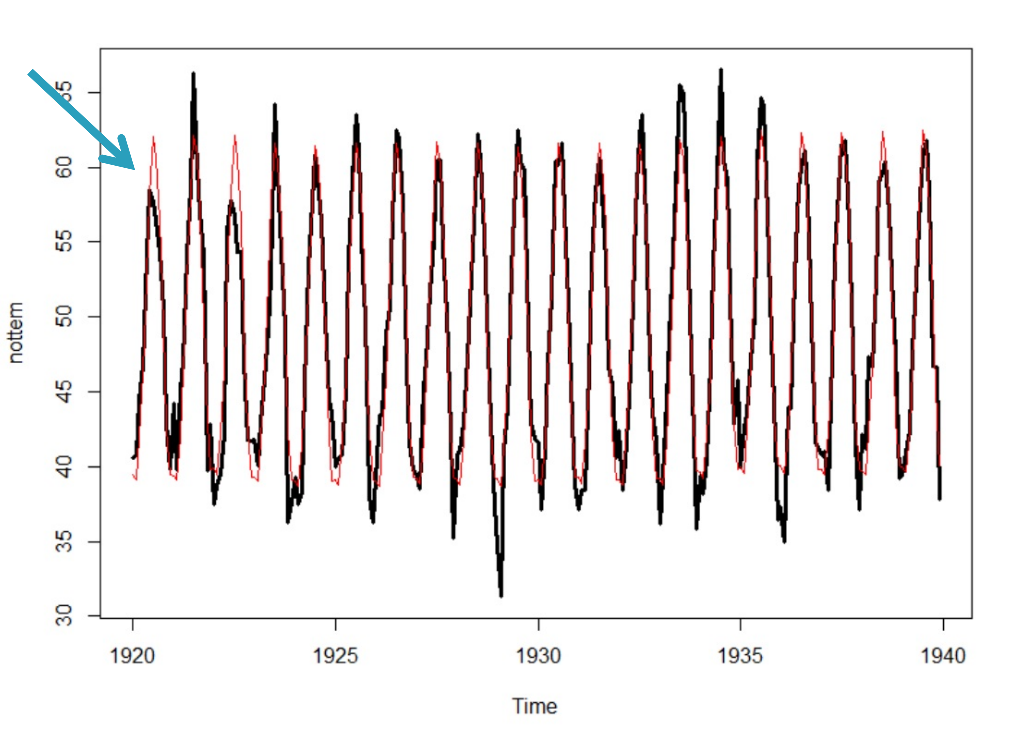
‘upper=’

‘lower=’

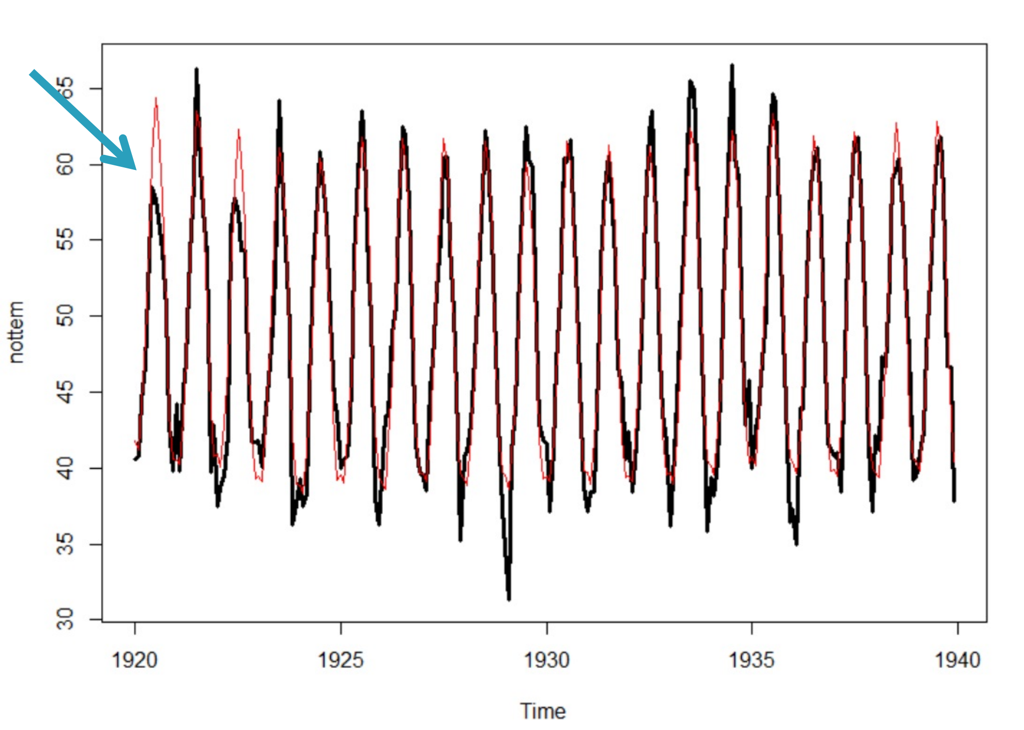
```r
plot(
    nottem, lwd = 3),
lines(
    etsmodel$fitted,
    col = “red”)
```

- Comparing the model to the original data
- Plots ‘nottem’
- Linewidth: 3px
- Adds an extra line to above plot
- Values: fitted values of ‘etsmodel’
- Line colour: red

Comparing the Models



Model 'ANA'



Model 'MNM'

R Implementation

Library(forecast)

Function ets()

Automatically selects the optimal model for the data

Share Your Thoughts with Us

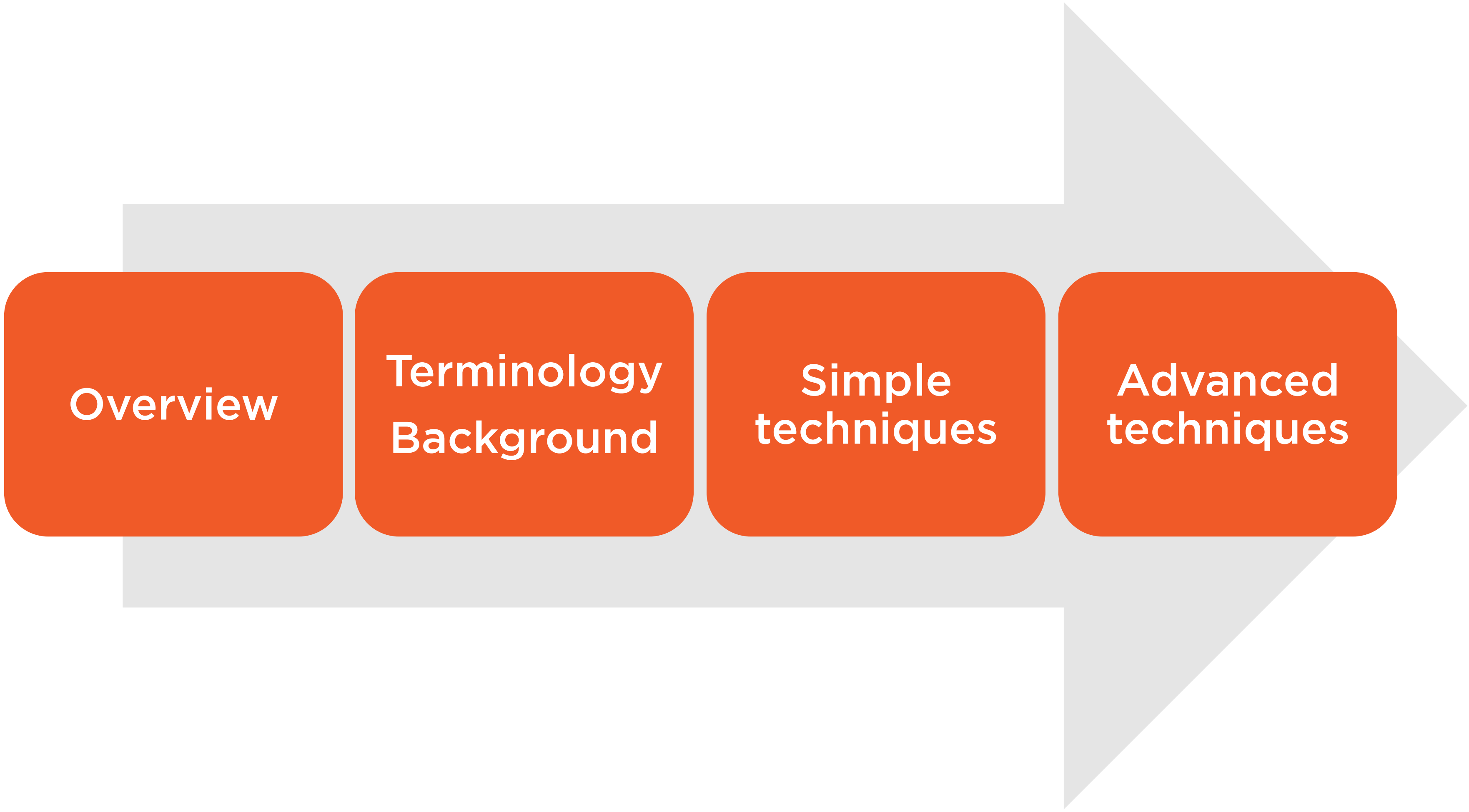
Questions

Use the discussion board to ask questions and leave comments

Rating

Please rate this course and give some feedback

Course Layout



Module: Traits of Time Series Data

Time Series Vectors (Lags)

Terminology
Stats Background

Time Series Patterns

Time Series Visualizations

Main Concepts

Autocorrelation – Correlation within the dataset

Stationarity – Constant mean and variance

Differencing – The difference between two consecutive observations

Module: Using Simple Time Series Models

Simple methods

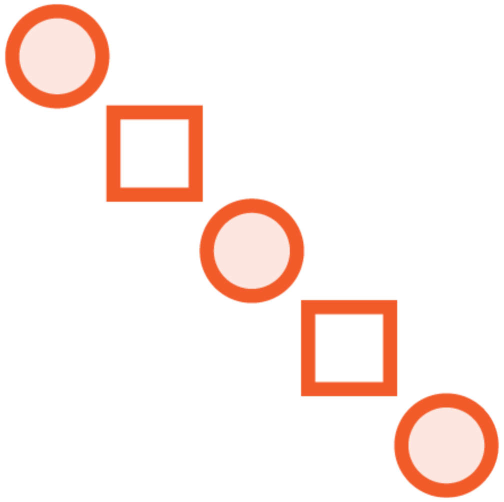
- Naïve
- Mean
- Drift

Model comparison with accuracy() and quality check with the residuals

Module: Using Advanced Time Series Models

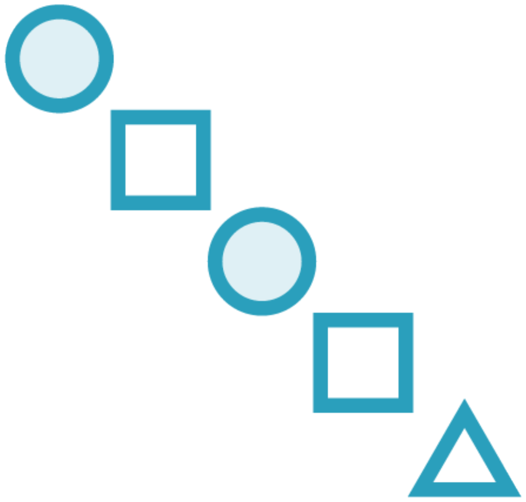
Forecast package					
ARIMA auto.arima()			**Exponential smoothing** ets()		
AR	I	MA	Error	Trend	Seasonality

Simple or Advanced Techniques?



Advanced Techniques

Pattern in the data



Simple Techniques

No pattern is present

Congratulations! You have finished this course

Implement these tools and techniques in your daily work

Don’t forget to update your portfolio/CV