Tecnológico
de Monterrey

Time Series Analysis for Data Science

CS5056 Data Analytics

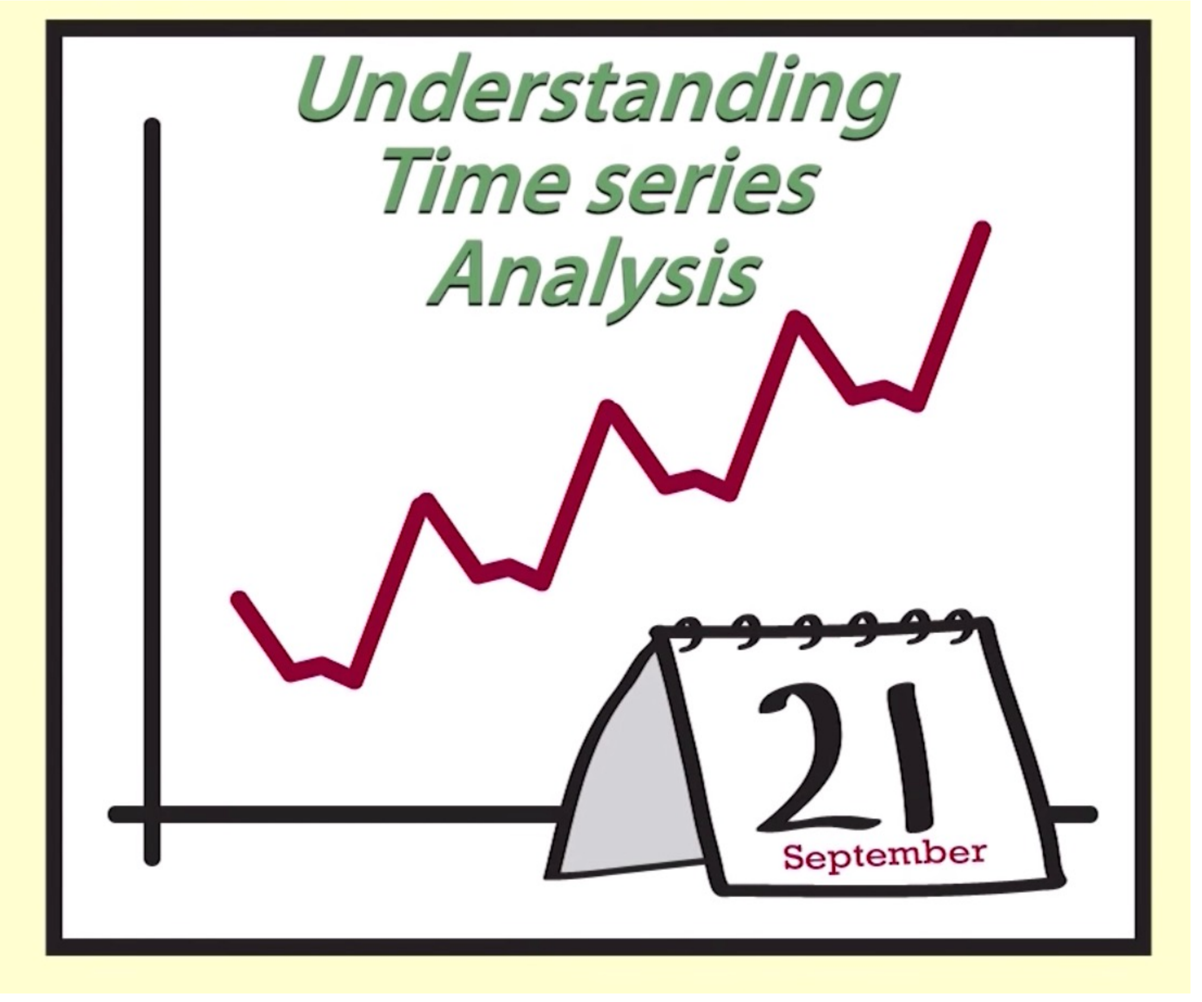
Francisco J. Cantú, Héctor Ceballos
Tecnológico de Monterrey
March 24, 2021

Februrary-June, 2021

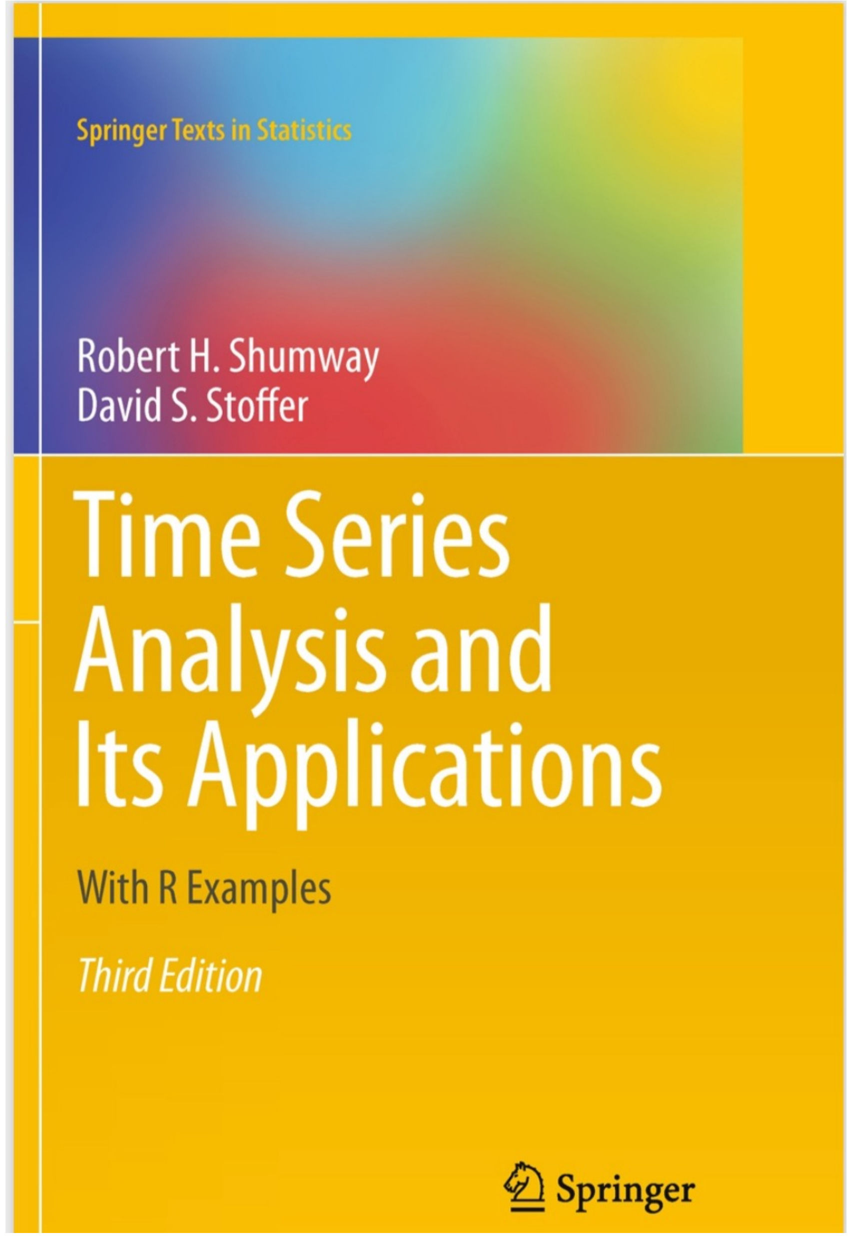
Agenda

CS5056 Data Analytics
March 24, 2021, Class 7/16

- An Overview of Time Series Analysis (TSA)
- Hands-on Exercise: TSA in R
- Hans-on Exercise: TSA in Python
- Ideas for your FPA

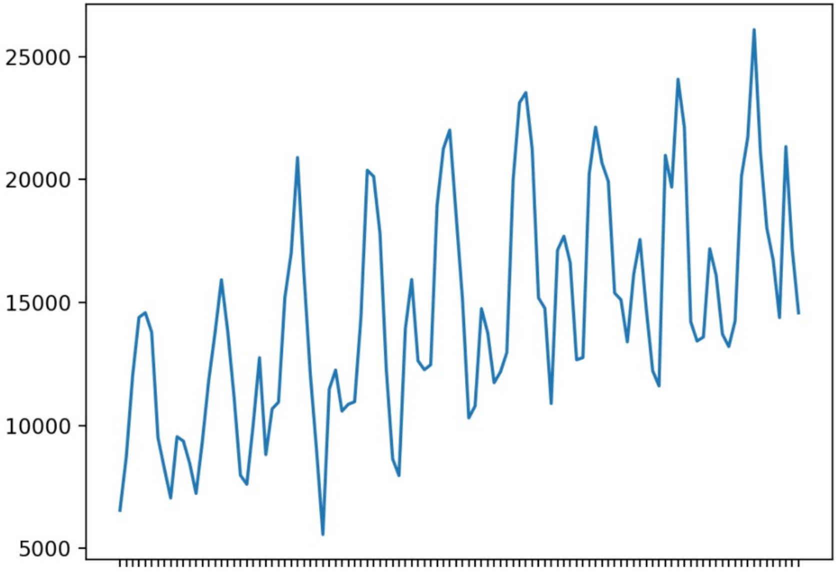


https://www.youtube.com/watch?v=GUq_tO2BjaU


Springer Texts in Statistics
Robert H. Shumway
David S. Stoffer
Time Series Analysis and Its Applications
With R Examples
Third Edition
Springer

Time Series Analysis

- A time series is a series of data points indexed in time order.
- It is a sequence of discrete-time data taken at successive equally spaced points in time.



Time Series Modeling for Data Science

- Thus, Time Series involves working on time-based data to derive hidden insights to make informed decision making
- Time may be in years, days, hours, minutes units of same length

Examples of TSA

- Heights of ocean tides
- Counts of sunspots
- The daily closing value of the Dow Jones stock market


I'VE FINALLY FOUND IT... AFTER 15 YEARS
Robotatertotcomics
THE SCROLL OF TRUTH!
It's impossible to predict stock prices
NYEHHH

TSA and Panel Data

- A time series is one type of panel data
- Panel data is the general class, a multidimensional data set, whereas a time series data set is a one-dimensional panel (as is a cross-sectional dataset)
- A data set may exhibit characteristics of both panel data and time series data.

TSA and Panel Data

- One way to tell is to ask **what makes one data record unique** from the other records
- If the answer is the time data field, then this is a time series data set candidate
- If determining a unique record requires a time data field and an additional identifier which is unrelated to time (student ID, stock symbol, country code), then it is panel data candidate
- If the differentiation lies on the non-time identifier, then the data set is a cross-sectional data set candidate.

Examples of Time Series

Time Series Examples

1. Kings of England age of death

2. Covid-19 confirmed cases

3. Births per month in New York city

4. Rain fall in London

5. Hem diameter of women’s skirts

6. Sales at a souvenir shop

Download the R script 1-tsa-examples.R

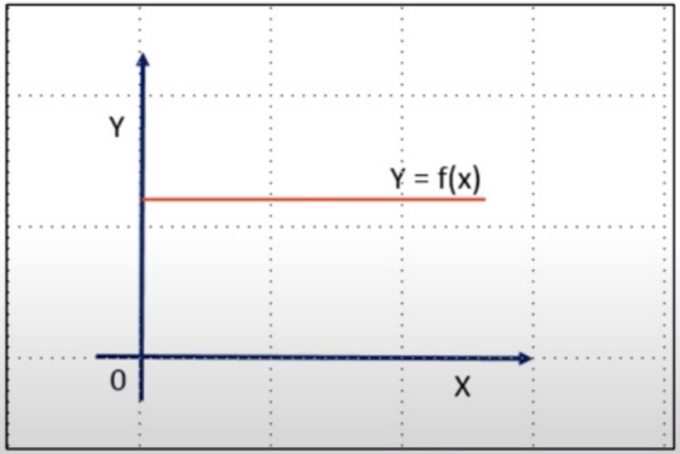
Some of the examples are taken from the Booklet A Little Book of R For Time Series Release 0.2, by Avril Coghlan, September 10. 2018

https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html

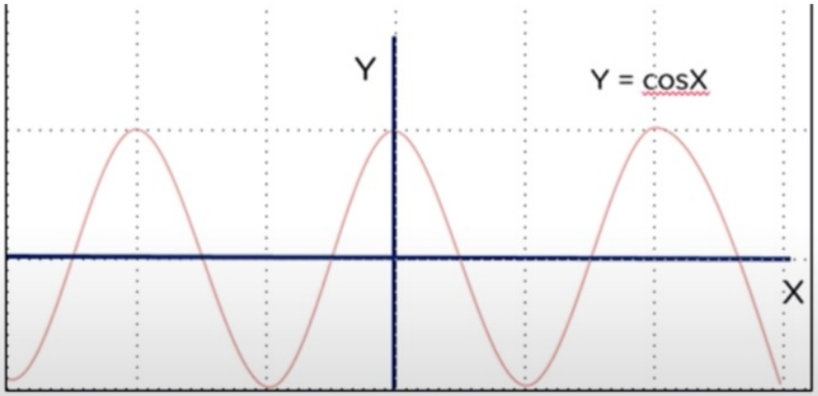
When NOT to use Time Series Analysis?

There are various conditions where you should not use Time Series:

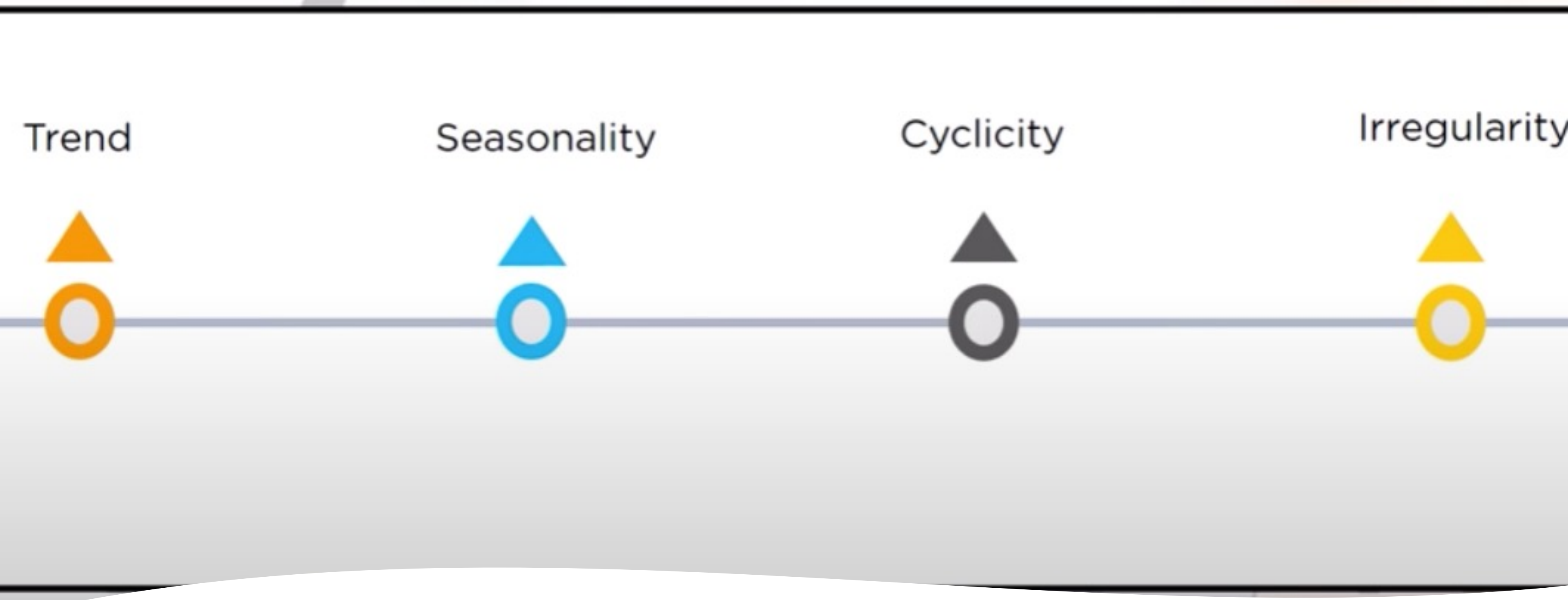
1. When the values are constant over a period of time:



2. When values can be represented by known functions like cosx, sinx etc:



Dissecting a Time Series



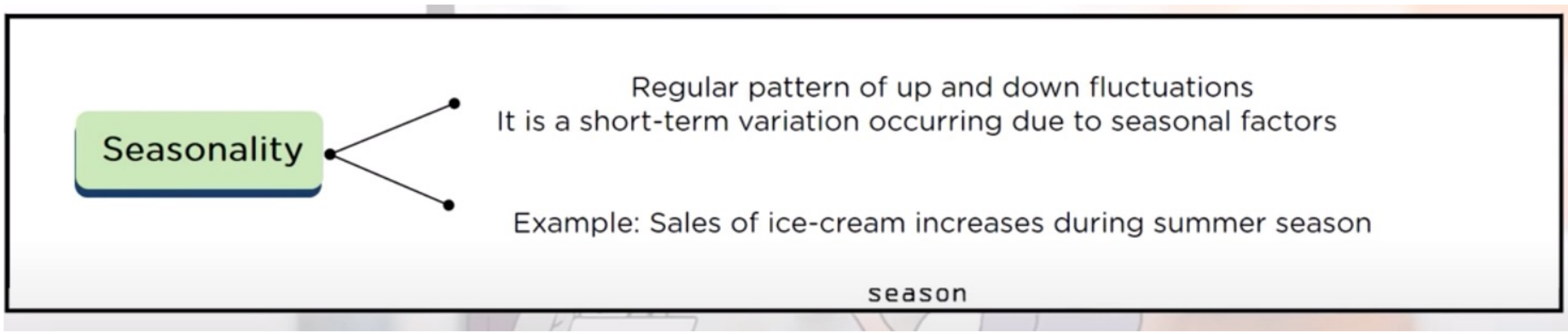
Time Series Components

- Trend
- Seasonality
- Cyclicity
- Irregularity or Randomness

https://www.youtube.com/watch?v=gj4L2isnOf8

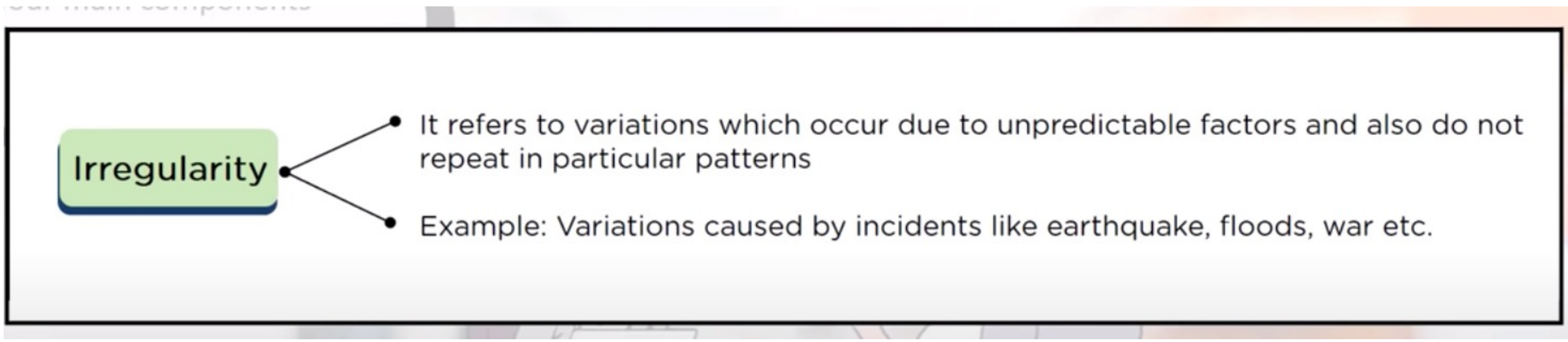
Trend

- Trend is the increase or decrease in the series over a period of time, it persists over a long period of time
- Example: Population growth over the years can be seen as an upward trend


Seasonality
Regular pattern of up and down fluctuations
It is a short-term variation occurring due to seasonal factors
Example: Sales of ice-cream increases during summer season
season

Cyclicity

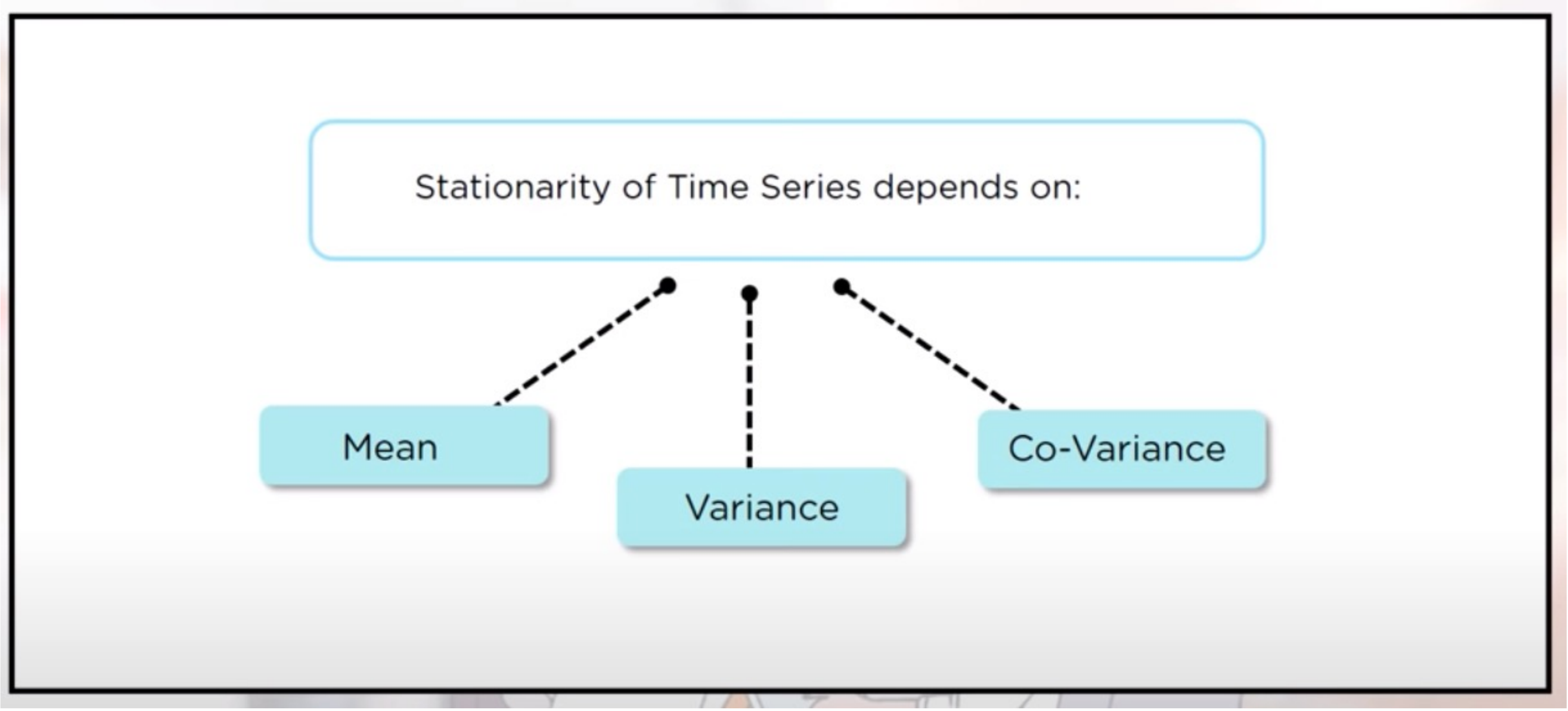
- It is a medium-term variation caused by circumstances, which repeat in irregular intervals
- Example: 5 years of economic growth, followed by 2 years of economic recession, followed be 7 years of economic growth followed by 1 year of economic recession


Irregularity
It refers to variations which occur due to unpredictable factors and also do not repeat in particular patterns
Example: Variations caused by incidents like earthquake, floods, war etc.

Non-Stationarity

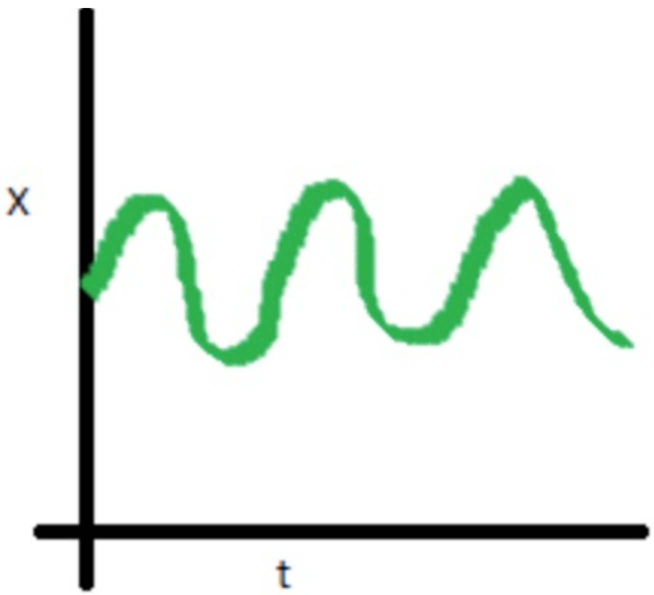
- A Time Series that has Trends, Seasonality, Cyclicity, or Irregularity is **Non-Stationary**
- What distinguishes a Stationary time series?

Stationarity

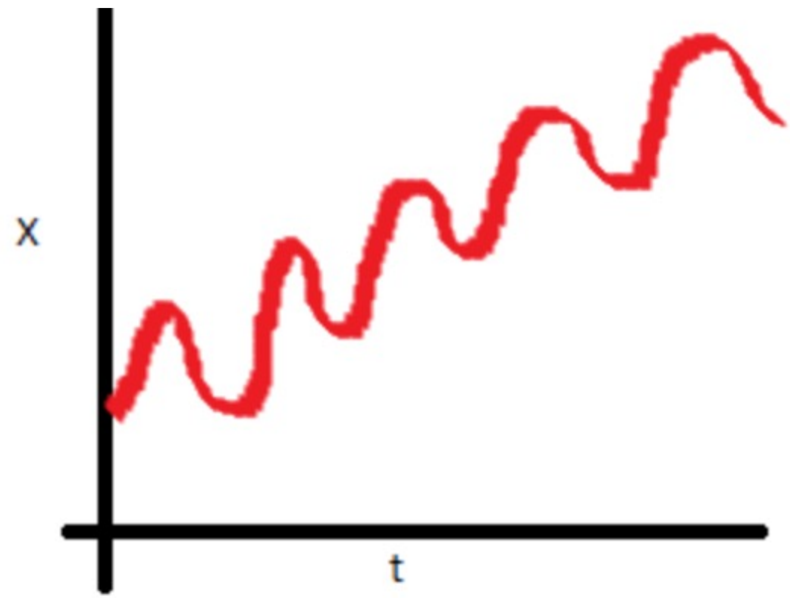


Mean

- The mean of the series should not be a function of time rather should be a constant.
- The image below has the left-hand graph satisfying the condition whereas the graph in red has a time dependent mean.



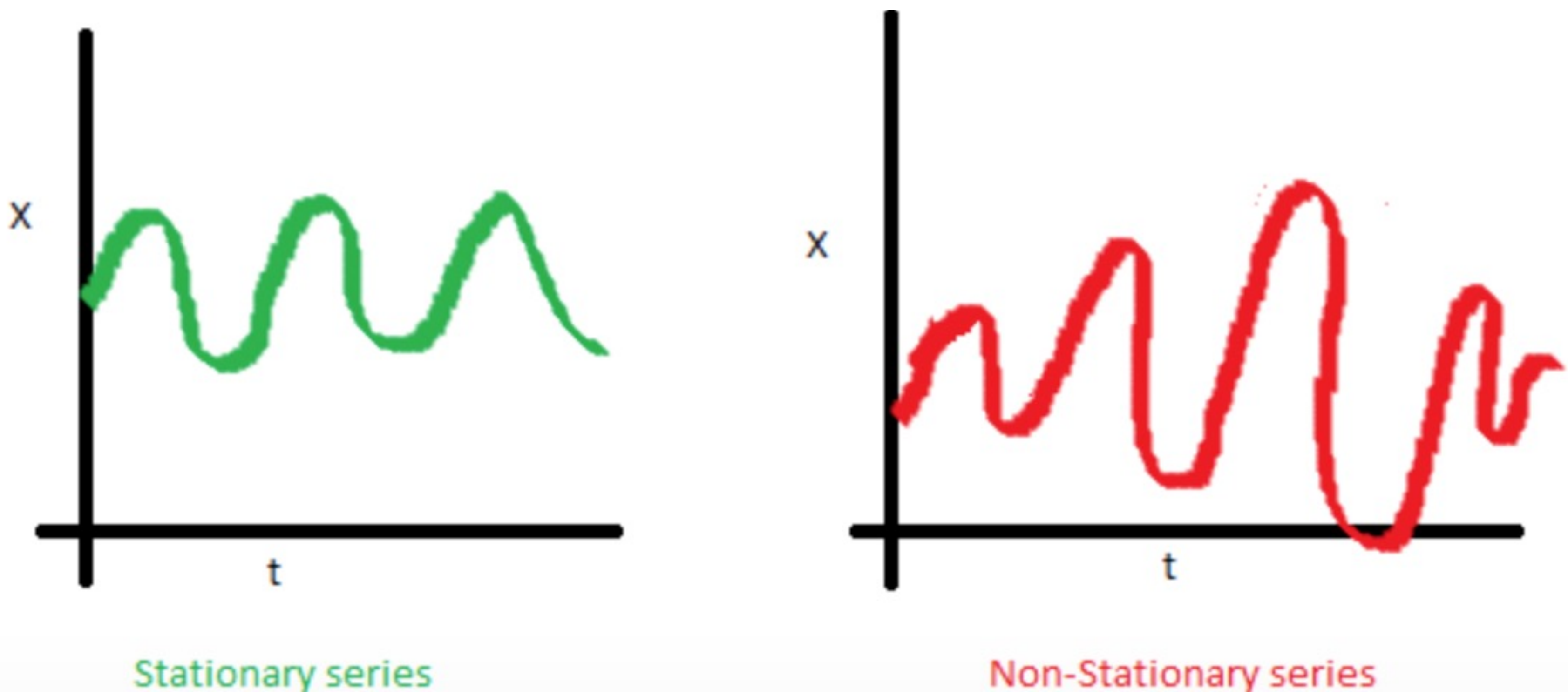
Stationary series



Non-Stationary series

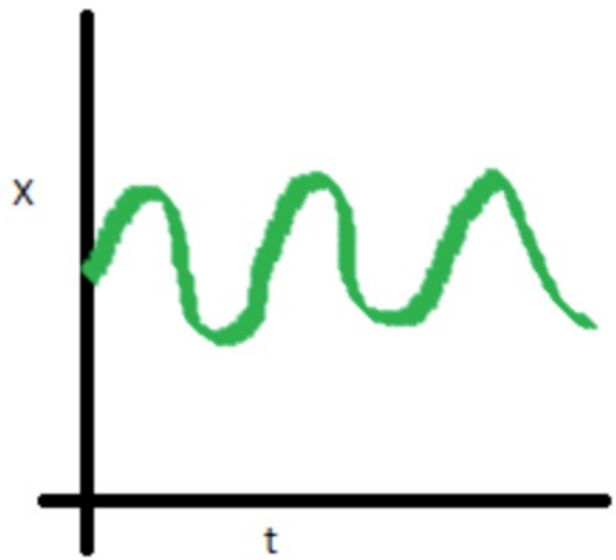
Variance

- The variance of the series should not a be a function of time. This property is known as homoscedasticity.
- Following graph depicts what is and what is not a stationary series. (Notice the varying spread of distribution in the right-hand graph)

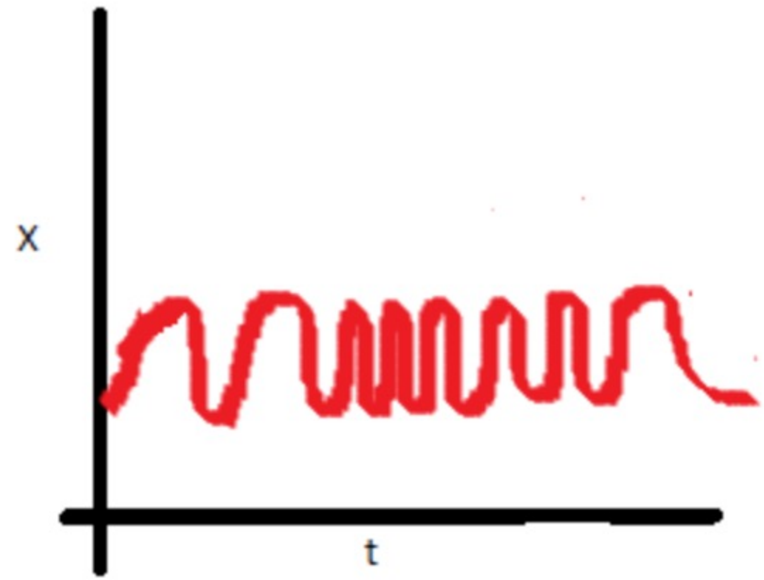


Covariance

- The covariance of the i-th term and the (i + m)-th term should not be a function of time.
- In the following graph, you will notice the spread becomes closer as the time increases. Hence, the covariance is not constant with time for the 'red series'.



Stationary series



Non-Stationary series

Examples of Time Series Decomposition

Examples of Time Series Decomposition

1. Kings of England age of death

2. Covid-19 confirmed cases

3. Births per month in New York city

4. Rain fall in London

5. Hem diameter of women's skirts

6. Sales at a souvenir shop

Download the R script 2-tsa-examples.R

Some of the examples are taken from the Booklet A Little Book of R For Time Series Release 0.2, by Avril Coghlan, September 10. 2018

https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html

Prediction with TSA

To apply TSA, the series should be **Stationary** so as to be able to make prediction

Prediction with TSA

How do we do it?

First, decompose the series into its components

Then, make seasonality adjustments

And finally, **smooth** the required components

Seasonality Adjustment

- Consists in subtracting the Seasonality component from the whole time series
- Download the R script 3-tsa-seasonality-adjust.R

Prediction using Time Series Analysis

Smoothing

Additive or Multiplicative Time Series Models

- A data model is additive if the effects of individual factors are differentiated and added together to model the data.
- A data model is multiplicative if it assumes that as the data increase, so does the seasonal pattern. Most time series plots exhibit such a pattern
- In a multiplicative model, the trend and seasonal components are multiplied and then added to the error component.

Smoothing

- As we have seen, Inherent in the collection of data taken over time is some form of random variation
- **Smoothing** is a method for reducing or canceling out the effect due to random variation
- When properly applied, reveals more clearly the underlying trend, seasonal and cyclic components.

Types of Smoothing

There are distinct groups of smoothing methods

- Moving Average (MA)
- Exponential Smoothing (ES)
- Auto Regressive Methods (AR)
- Combination of AR and MA: ARMA, ARIMA

Moving Average (MA)

- Moving Average is a method for smoothing a time series
- Provides a way to reduce noise and smooth a time series and is a very basic forecasting technique
- It is used for modeling univariate time series models:

$$X_t = \mu + A_t - \theta_1 A_{t-1} - \theta_2 A_{t-2} - \cdots - \theta_q A_{t-q}$$

- X_t is the time series, μ is the mean of the series, A_{t-i} are white noise terms, and $\theta_1, \ldots, \theta_q$ are the parameters of the model.
- The value of q is called the order of the MA model
- Simple Moving Average (SMA) calculates the arithmetic mean of the series over the past q observations.

TTR Library

SMA calculates the arithmetic mean of the series over the past n observations.

EMA calculates an exponentially-weighted mean, giving more weight to recent observations.

WMA is similar to an EMA, but with linear weighting if the length of wts is equal to n. If the length

of wts is equal to the length of x, the WMA will use the values of wts as weights.

DEMA is calculated as: DEMA = (1 + v) * EMA(x,n) - EMA(EMA(x,n),n) * v (with the corresponding

wilder and ratio arguments).

EVWMA uses volume to define the period of the MA.

ZLEMA is similar to an EMA, as it gives more weight to recent observations, but attempts to remove

lag by subtracting data prior to (n-1)/2 periods (default) to minimize the cumulative effect.

VWMA and VWAP calculate the volume-weighted moving average price.

VMA calculate a variable-length moving average based on the absolute value of w. Higher (lower)

values of w will cause VMA to react faster (slower).

HMA a WMA of the difference of two other WMAs, making it very responsive.

ALMA inspired by Gaussian filters. Tends to put less weight on most recent observations, reducing

tendency to overshoot.

Exponential Smoothing

Simple Exponential Smoothing

- It is a technique used for data that **has no trend or seasonal pattern**.
- The SES is the simplest among all the exponential smoothing techniques
- We know that in any type of exponential smoothing we weigh the recent values or observations more heavily rather than the old values or observations
- The weight of each and every parameter is always determined by a **smoothing parameter** or **alpha.** The value of alpha lies between 0 and 1.
- In practice, if alpha is between 0.1 and 0.2 then SES will perform quite well.
- When alpha is closer to 0 then it is considered as slow learning since the algorithm is giving more weight to the historical data.
- If the value of alpha is closer to 1 then it is referred to as fast learning since the algorithm is giving the recent observations more weight
- Hence, we can say that the recent changes in the data will be leaving a greater impact on the forecasting

Holt's Exponential Smoothing Method

In SES we have to remove the long-term trends to improve the model

But in **Holt's Method**, we can apply exponential smoothing while we are capturing trends in the data

This is a technique that works with data **having a trend but no seasonality**

In order to make predictions on the data, the Holt's Method uses two **smoothing parameters, alpha**, **and beta**, which correspond to the level components and trend components.

Holt-Winter’s Exponential Smoothing Method

- It is used for data **with both seasonal patterns and trends**
- This method can be implemented either by using Additive structure or by using the Multiplicative structure depending on the data set
- The Additive structure or model is used when the seasonal pattern of data has the same magnitude or is consistent throughout, while the Multiplicative structure or model is used if the magnitude of the seasonal pattern of the data increases over time
- It uses **three smoothing parameters,- alpha, beta, and gamma**

Evaluation of Autocorrelation and Residuals

We can investigate whether the predictive model can be improved upon by checking whether the in-sample forecast errors show non-zero autocorrelations at lags 1-20, by making a correlogram and carrying out the Ljung-Box test:

- acf(rainseriesforecasts2$residuals, lag.max = NULL, type = c("correlation", "covariance", "partial"), plot = TRUE, na.action = na.pass, demean = TRUE)
- plot.ts(souvenirtimeseriesforecasts2$residuals)
- plotForecastErrors(souvenirtimeseriesforecasts2$residuals)

Hands-on Exercise with Moving Averages and Exponential Smoothing

- Download R script 4-tsa-exp-smooth.R

ARIMA Smoothing

Autoregressive Models (AR)

Exponential smoothing methods are useful for making forecasts and make no assumptions about the correlations between successive values of the time series.

However, if you want to make prediction intervals for forecasts made using exponential smoothing methods, the prediction intervals require that the forecast errors are uncorrelated and are normally distributed with mean zero and constant variance.

While exponential smoothing methods do not make any assumptions about correlations between successive values of the time series, in some cases you can make a better predictive model by taking correlations in the data into account.

Autoregressive Models (AR)

An Autoregressive model is a linear regression of the current value of the series against one or more prior values of the series.

The value of p is called the order of the AR model.

$X_t=\delta+\varphi_1 X_{t-1}+\varphi_2 X_{t-2}+\cdots+\varphi_p X_{t-p}+A_t$

X_t is the time series, A_t is white noise, and $\delta=(1-\sum_{i=1}^{p}\varphi i)\mu$, with μ denoting the process mean

ARIMA Models

- **Autoregressive Integrated Moving Average (ARIMA)** models include an explicit statistical model for the irregular component of a time series, that allows for non-zero autocorrelations in the irregular component.
- ARIMA models are defined for stationary time series.
- Therefore, if you start-off with a non-stationary time series, you will first need to **'difference'** the time series until you obtain a stationary time series.
- If you have to difference the time series **d** times to obtain a stationary series, then you have an ARIMA(p,**d**,q) model, where **d** is the order of differencing used, **p** is the window size of AR, and **q** is the window size of MA

ARIMA Models

- If your time series is stationary, or if you have transformed it to a stationary time series by differencing d times, the next step is to select the appropriate ARIMA model, which means finding the values of most appropriate values of p and q for an ARIMA(p,d,q) model

- To do this, you usually need to examine the correlogram and partial correlogram of the stationary time series

- To plot a correlogram and partial correlogram, we can use the "acf()" and "pacf()" functions in R, respectively

- To get the actual values of the autocorrelations and partial autocorrelations, we set "plot=FALSE" in the "acf()" and "pacf()" functions.

Skirts Example with ARIMA

For example, the time series of the annual diameter of women's skirts at the hem, from 1866 to 1911 is not stationary in mean, as the level changes a lot over time

You can difference a time series using the "diff()" function in R. :

```
# Forecasts using ARIMA
> skirtsseriesdiff1 <- diff(skirtsseries, differences=1)
> plot.ts(skirtsseriesdiff1)
> skirtsseriesdiff2 <- diff(skirtsseries, differences=2)
> plot.ts(skirtsseriesdiff2)
```

Hands-on Exercises with ARIMA using R

- Download the R script 5-tsa-ARIMA.R

ARIMA Models

- The time series of second differences (above) does appear to be stationary in mean and variance, as the level of the series stays roughly constant over time, and the variance of the series appears roughly constant over time
- Thus, it appears that we need to difference the time series of the diameter of skirts twice in order to achieve a stationary series.
- In general, if you need to difference your original time series data d times in order to obtain a stationary time series, this means that you can use an ARIMA(p,d,q) model for your time series, where d is the order of differencing used
- For example, for the time series of the diameter of women's skirts, we had to difference the time series twice, and so the order of differencing (d) is 2
- This means that you can use an **ARIMA(p,2,q)** model for your time series. The next step is to figure out the values of p and q for the ARIMA model.

Kings Example with ARIMA

Coming back to the time series of the age of death of the successive kings of England

Forecasts using ARIMA

```
> kingtimeseriesdiff1 <- diff(kingstimeseries, differences=1)
> plot.ts(kingtimeseriesdiff1)>
```

ARIMA Models

- The time series of first differences appears to be stationary in mean and variance, and so an ARIMA(p,1,q) model is probably appropriate for the time series of the age of death of the kings of England.
- By taking the time series of first differences, we have removed the trend component of the time series of the ages at death of the kings and are left with an irregular component.
- We can now examine whether there are correlations between successive terms of this irregular component; if so, this could help us to make a predictive model for the ages at death of the kings.

Kings Example with ARIMA

For example, to plot the correlogram for lags 1-20 of the once differenced time series of the ages at death of the kings of England, and to get the values of the autocorrelations, we type:

- #Forecasts using ARIMA

```
> acf(kingtimeseriesdiff1, lag.max=20)
> acf(kingtimeseriesdiff1, lag.max=20,
plot=FALSE)
```

Kings Example with ARIMA

To plot the partial correlogram for lags 1-20 for the once differenced time series of the ages at death of the English kings, and get the values of the partial autocorrelations, we use the “pacf()” function, by typing:

```
> pacf(kingtimeseriesdiff1, lag.max=20)
> pacf(kingtimeseriesdiff1, lag.max=20, plot=FALSE)
```

ARMA Model

- The partial correlogram shows that the partial autocorrelations at lags 1, 2 and 3 exceed the significance bounds, are negative, and are slowly decreasing in magnitude with increasing lag (lag 1: -0.360, lag 2: -0.335, lag 3:-0.321). The partial autocorrelations tail off to zero after lag 3.
- Since the correlogram is zero after lag 1, and the partial correlogram tails off to zero after lag 3, this means that the following ARMA (autoregressive moving average) models are possible for the time series of first differences:
- an ARMA(3,0) model, that is, an autoregressive model of order p=3, since the partial autocorrelogram is zero after lag 3, and the autocorrelogram tails off to zero (although perhaps too abruptly for this model to be appro- priate)
- An ARMA(0,1) model, that is, a moving average model of order q=1, since the autocorrelogram is zero after lag 1 and the partial autocorrelogram tails off to zero
- an ARMA(p,q) model, that is, a mixed model with p and q greater than 0, since the autocorrelogram and partial correlogram tail off to zero (although the correlogram probably tails off to zero too abruptly for this model to be appropriate)

ARMA Model

We use the principle of parsimony to decide which model is best: that is, we assume that the model with the fewest parameters is best.

The ARMA(3,0) model has 3 parameters, the ARMA(0,1) model has 1 parameter, and the ARMA(p,q) model has at least 2 parameters. Therefore, the ARMA(0,1) model is taken as the best model.

ARMA Model

An ARMA(0,1) model is a moving average model of order 1, or MA(1) model. This model can be written as: X_t - mu = Z_t - (theta * Z_t-1), where X_t is the stationary time series we are studying (the first differenced series of ages at death of English kings), mu is the mean of time series X_t, Z_t is white noise with mean zero and constant variance, and theta is a parameter that can be estimated.

A MA (moving average) model is used to model a time series that shows short-term dependencies between successive observations. Intuitively, it makes good sense that a MA model can be used to describe the irregular component in the time series of ages at death of English kings, as we might expect the age at death of a particular English king to have some effect on the ages at death of the next king or two, but not much effect on the ages at death of kings that reign much longer after that.

Auto ARIMA

The auto.arima() function can be used to find the appropriate ARIMA model

> auto.arima(kings)

The output says an appropriate model is ARIMA(0,1,1).

Since an ARMA(0,1) model (with p=0, q=1) is taken to be the best candidate model for the time series of first differences of the ages at death of English kings, then the original time series of the ages of death can be modelled using an ARIMA(0,1,1) model (with p=0, d=1, q=1, where d is the order of differencing required).

Kings Example with ARIMA

We continue with time series of the ages at death of the kings of England.

You can specify the values of p, d and q in the ARIMA model by using the “order” argument of the “arima()” function in R.

To fit an ARIMA(p,d,q) model to this time series, we type:

```
# Forecasts using ARIMA
> kingstimeseriesarima <- arima(kingstimeseries, order=c(0,1,1))
> kingstimeseriesarima
```

Confidence Level

- As mentioned above, if we are fitting an ARIMA(0,1,1) model to our time series, it means we are fitting an an ARMA(0,1) model to the time series of first differences.
- An ARMA(0,1) model can be written X_t - mu = Z_t - (theta * Z_t-1), where theta is a parameter to be estimated.
- From the output of the "arima()" R function (above), the estimated value of theta (given as 'ma1' in the R output) is -0.7218 in the case of the ARIMA(0,1,1) model fitted to the time series of ages at death of kings.
- You can specify the confidence level for prediction intervals in forecast.Arima() by using the "level" argument. For example, to get a 99.5% prediction interval, we would type "forecast.Arima(kingstimeseriesarima, h=5, level=c(99.5))".

Prediction with the Kings Example using ARIMA

We can then use the ARIMA model to make forecasts for future values of the time series, using the "forecast()" function in the "forecast" R package:

```
# Forecasts using ARIMA

> kingstimeseriesforecasts <-
forecast.Arima(kingstimeseriesarima, h=5)
> kingstimeseriesforecasts
> plot.forecast(kingstimeseriesforecasts)
```

As in the case of exponential smoothing models, it is a good idea to investigate whether the forecast errors of an ARIMA model are normally distributed with mean zero and constant variance, and whether the are correlations be- tween successive forecast errors.

Prediction with the Kings Example using ARIMA

For example, we can make a correlogram of the forecast errors for our ARIMA(0,1,1) model for the ages at death of kings, and perform the Ljung-Box test for lags 1-20, by typing:

#Forecasts using ARIMA

```
> acf(kingstimeseriesforecasts$residuals, lag.max=20)
> Box.test(kingstimeseriesforecasts$residuals, lag=20, type="Ljung-Box")
```

Since the correlogram shows that none of the sample autocorrelations for lags 1-20 exceed the significance bounds, and the p-value for the Ljung-Box test is 0.9, we can conclude that there is very little evidence for non-zero autocorrelations in the forecast errors at lags 1-20.

Forecast Errors

To investigate whether the forecast errors are normally distributed with mean zero and constant variance, we can make a time plot and histogram (with overlaid normal curve) of the forecast errors:

#Forecasts using ARIMA

```
> plot.ts(kingstimeseriesforecasts$residuals)
```

```
> plotForecastErrors(kingstimeseriesforecasts$residuals)
```

Revising Model Conditions

The time plot of the in-sample forecast errors shows that the variance of the forecast errors seems to be roughly constant over time (though perhaps there is slightly higher variance for the second half of the time series).

The histogram of the time series shows that the forecast errors are roughly normally distributed, and the mean seems to be close to zero

Therefore, it is plausible that the forecast errors are normally distributed with mean zero and constant variance.

Since successive forecast errors do not seem to be correlated, and the forecast errors seem to be normally distributed with mean zero and constant variance, the ARIMA(0,1,1) does seem to provide an adequate predictive model for the ages at death of English kings.

Hands-on Exercises with Time Series Analysis using Python

Download the Python script tsa1.py

Download the Python script tsa2.py

Francisco J. Cantú-Ortiz, PhD

Professor of Computer Science and Artificial Intelligence
Tecnológico de Monterrey
Enago-Academy Advisor for Strategic Alliances

E-mail: fcantu@itesm.mx, fjcantor@gmail.com
Cel: +52 81 1050 8294, SNI-2 CVU: 9804
Personal Page: http://semtech.mty.itesm.mx/fcantu/
Facebook: fcantu; Twitter: @fjcantor; Skype: fjcantor
Orcid: 0000-0002-2015-0562
Scopus ID:6701563520
Researcher ID: B-8457-2009
https://www.researchgate.net/profile/Francisco_Cantu-Ortiz
https://scholar.google.com.mx/citations?hl=es&user=45-uuK4AAAAJ
https://itesm.academia.edu/FranciscoJavierCantuOrtiz
Ave. Eugenio Garza Sada No. 2501, Monterrey N.L., C.P. 64849, México