Traits of Time Series Data

Martin Burger
R PROGRAMMING TUTOR

www.r-tutorials.com

Statistical Background

Important terminology to know

Time series vectors and lags

Visualizing time series data

Identifying time series characteristics with plots

Stationarity

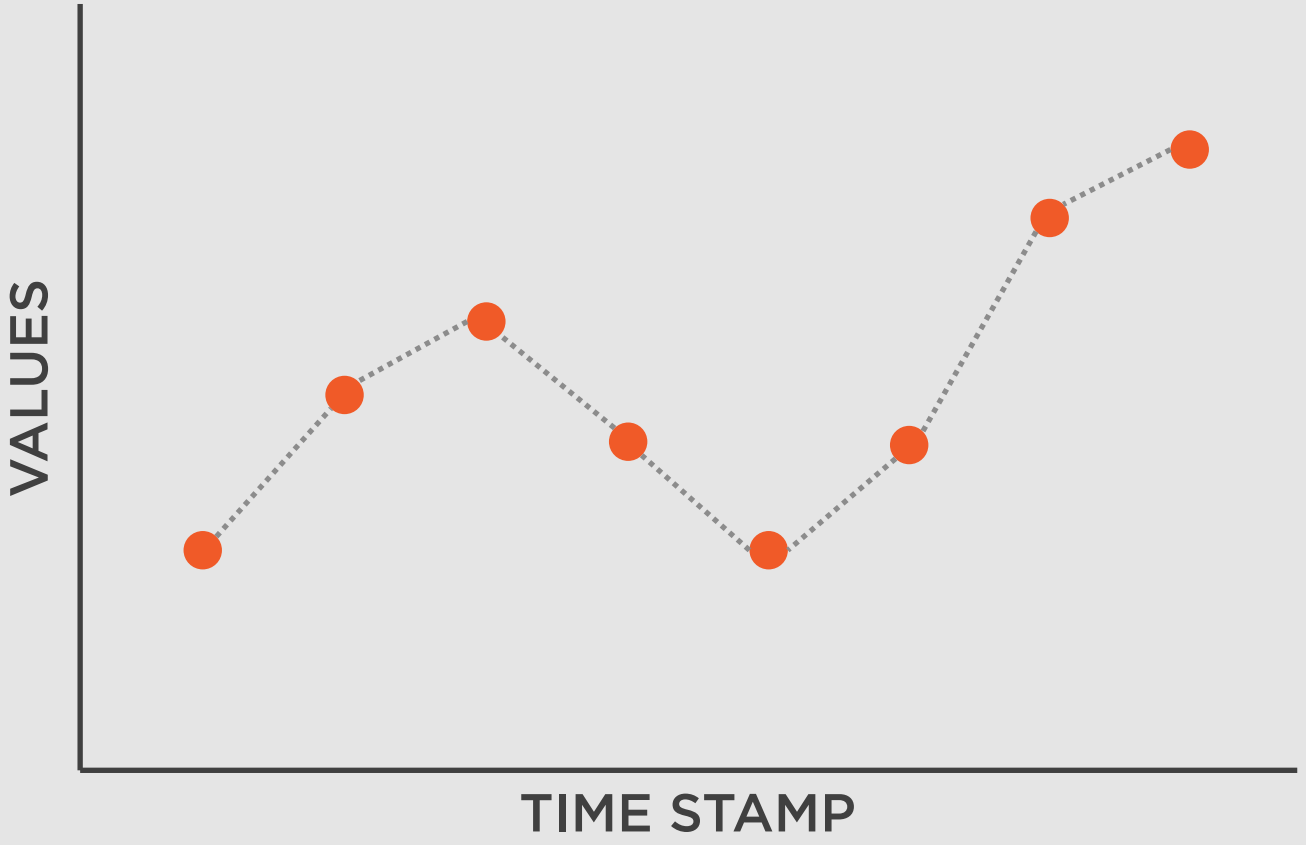
- Mean and variance
- Differencing

Autocorrelation

What are the basic characteristics of a time series?

What makes the difference between a vector and a univariate time series?

The Time Stamp



Time Series or Vector?

Time Series

The time stamp specifies a successive order for the values

Vector

A unique ID does not necessarily provide a specific order to the data

Meaningful and Non-meaningful Orders

May June July August September

Choose time stamps
whenever possible.

Converting Vectors to Time Series

Function `ts()`

- Attaches a time stamp to a vectors
- Converts the class to ‘ts’
- Use it to build time series from scratch

`Library(xts)`

- Importing time series data into R

What Is a Lag?

Y_t: An observation of the time series

Y_{114} = 3396 (The last observation of 'lynx')

Lag of 1 = $Y_t - Y_{t-1} = Y_{114} - Y_{113}$ = 3396–2657

Lag of 2 = $Y_t - Y_{t-2} = Y_{114} - Y_{112}$ = 3396 – 1590

Lag: a gap between two or more observations

General Summary Statistics of Time Series

Univariate Time Series

One variable attached to a time stamp

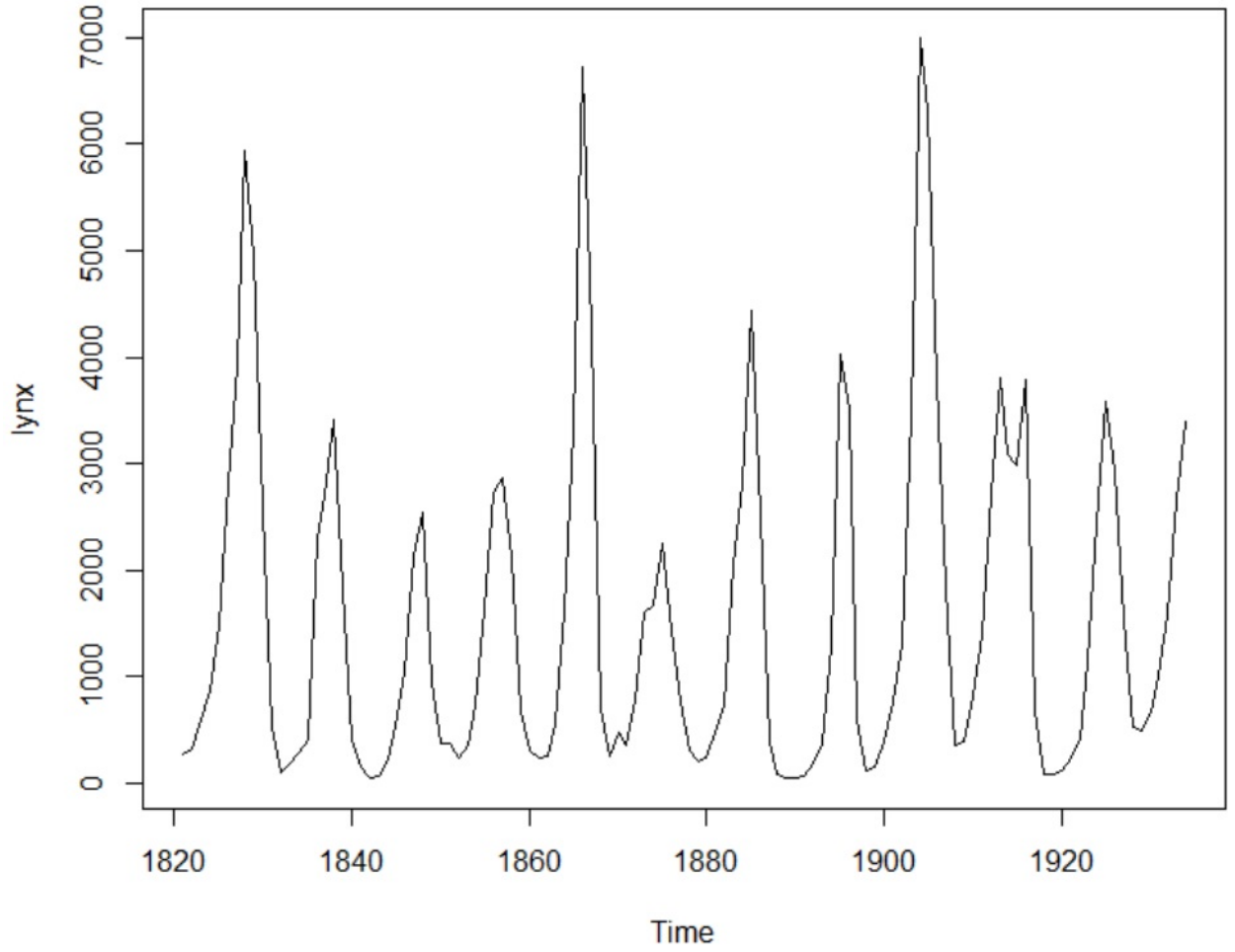
Multivariate Time Series

Two or more variables attached to a time stamp

Short, cyclical peaks

High mean

Lower median



```
quantile(
    lynx,
    prob =
        seq(0, 1, length=11),
    type = 5)
```

- Extracting the deciles
- Use the function quantile()

- Data to process
- Numeric vector of probabilities
- Sequence of 11 values from 0 to 1
- Specifies the deciles

Basic Functions for Univariate Time Series

mean()

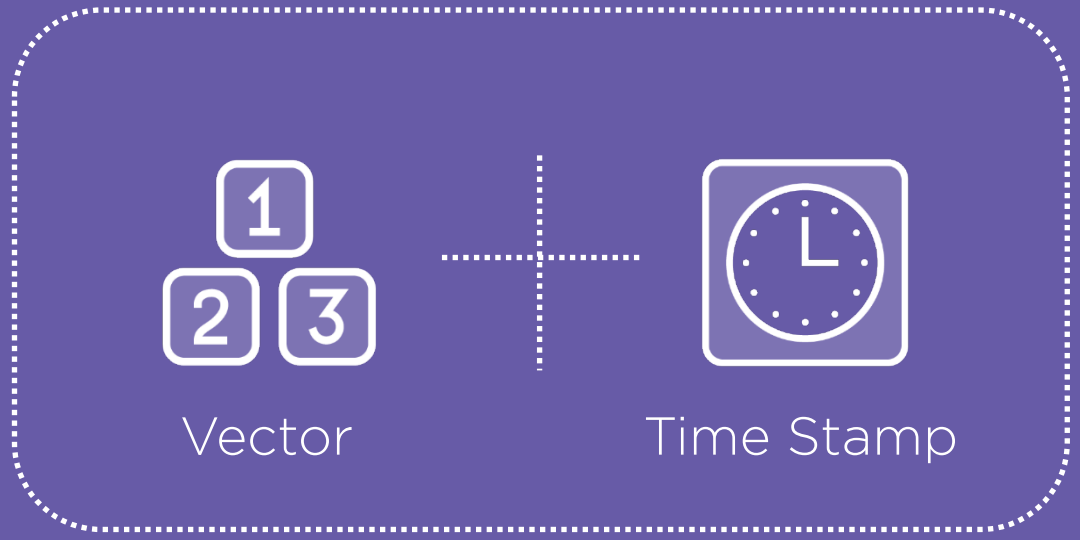
median()

plot()

sort()

quantile()

From Vector to Time Series



Class ‘ts’

Time Series Data

```
mytimeseries =
    ts(
        data = mydata,
        start = c(1956, 3),
        frequency = 4)
```

- Changing the class and attaching the time stamp
- Time series object to be created
- Function `ts()` converts `'mydata'` to 'ts' class
- Data to be converted
- Start point of the time stamp
- Concatenate: 1956 Q3
- Assigning four values per year to get quarterly data
- Describe equally spaced time interval patterns with the arguments frequency and start
- `Class = 'mts'` for multivariate time series

Datasets and their statistical character

Successive order by time component

Visualizing reveals the characteristics

Visualizing helps to find the appropriate analytical method

Patterns to identify:

- Trend
- Seasonality
- Mean
- Variance
- Stationarity

Main Statistical Characters of Univariate Data

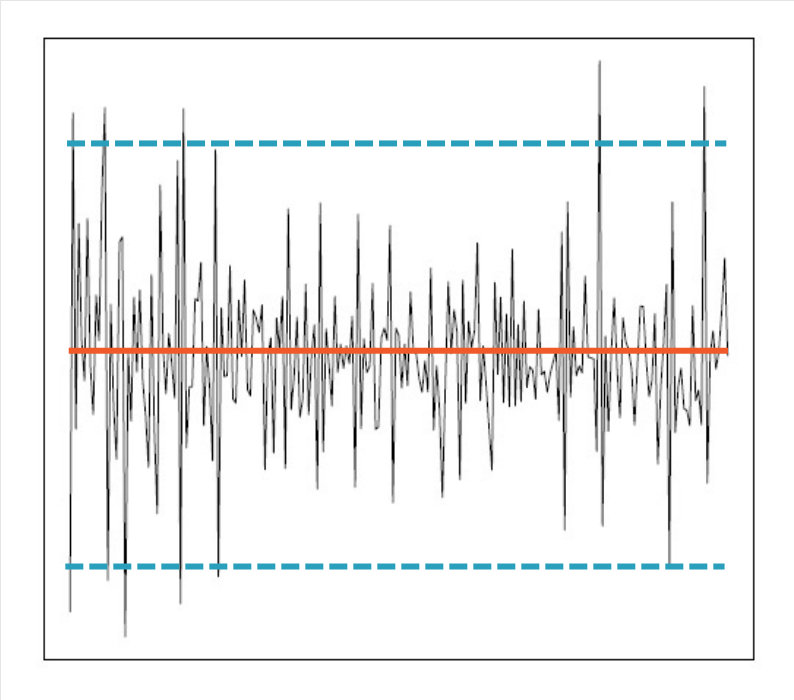
Trend
Dataset moving towards a direction

Seasonality
Repeated pattern over a fixed interval

Mean
The average of the dataset

Variance
Indicator of variability

Stationarity
Constant mean and variance



Time series of random normally distributed data

No trend

Constant mean

Constant variance

Stationarity is present

Transformation and differencing is not needed

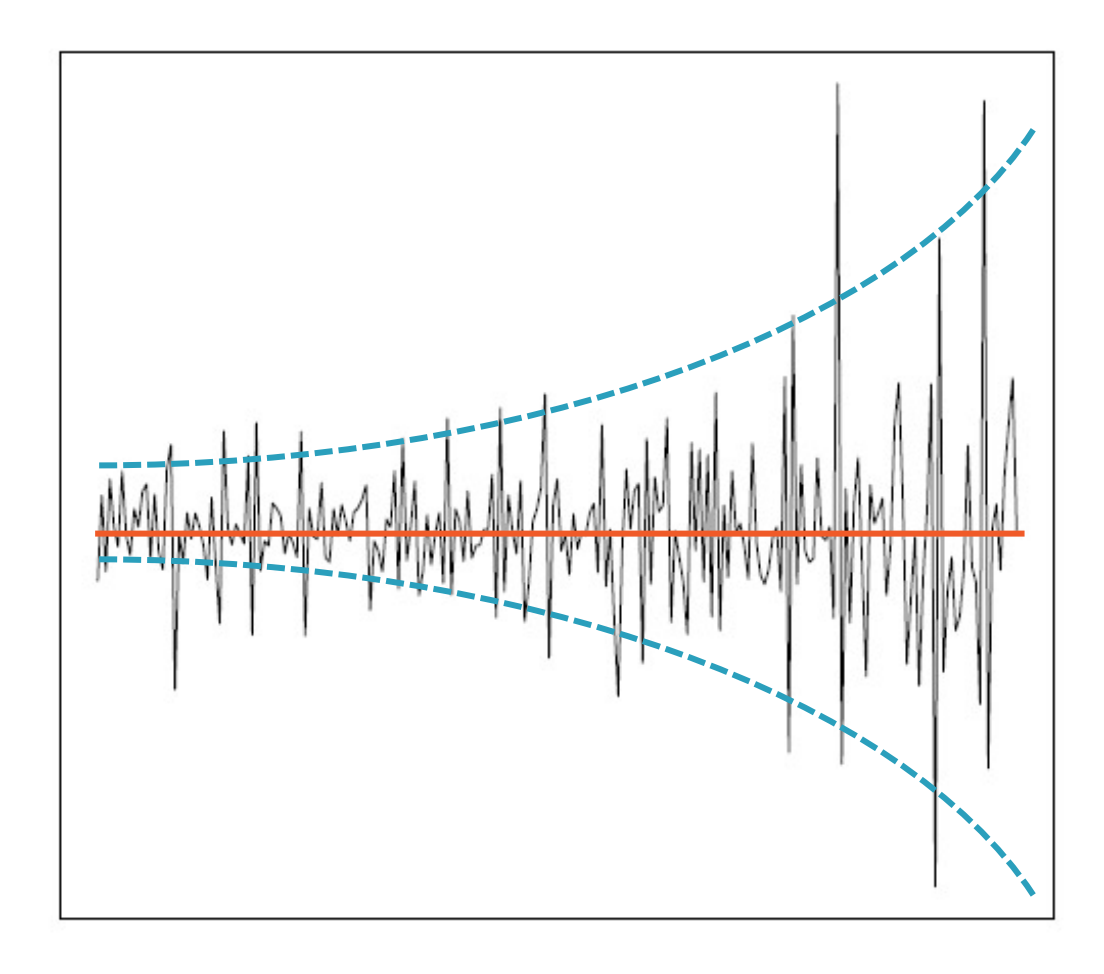
Constant mean

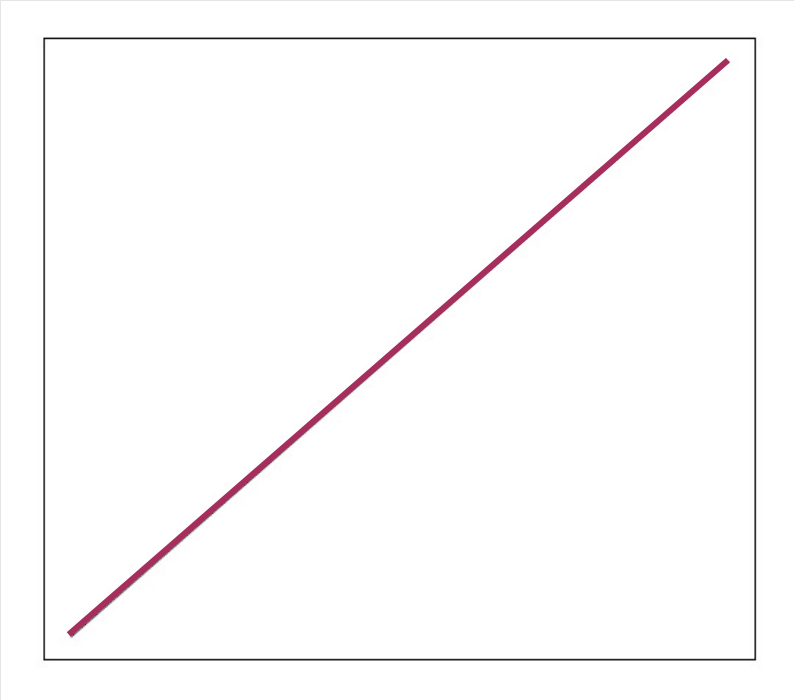
Changing variance

Heteroscedastic dataset

Non-stationary

Preprocessing prior analysis





Time series with a clear trend

Increasing mean

Non-stationary dataset

Preprocessing prior modeling

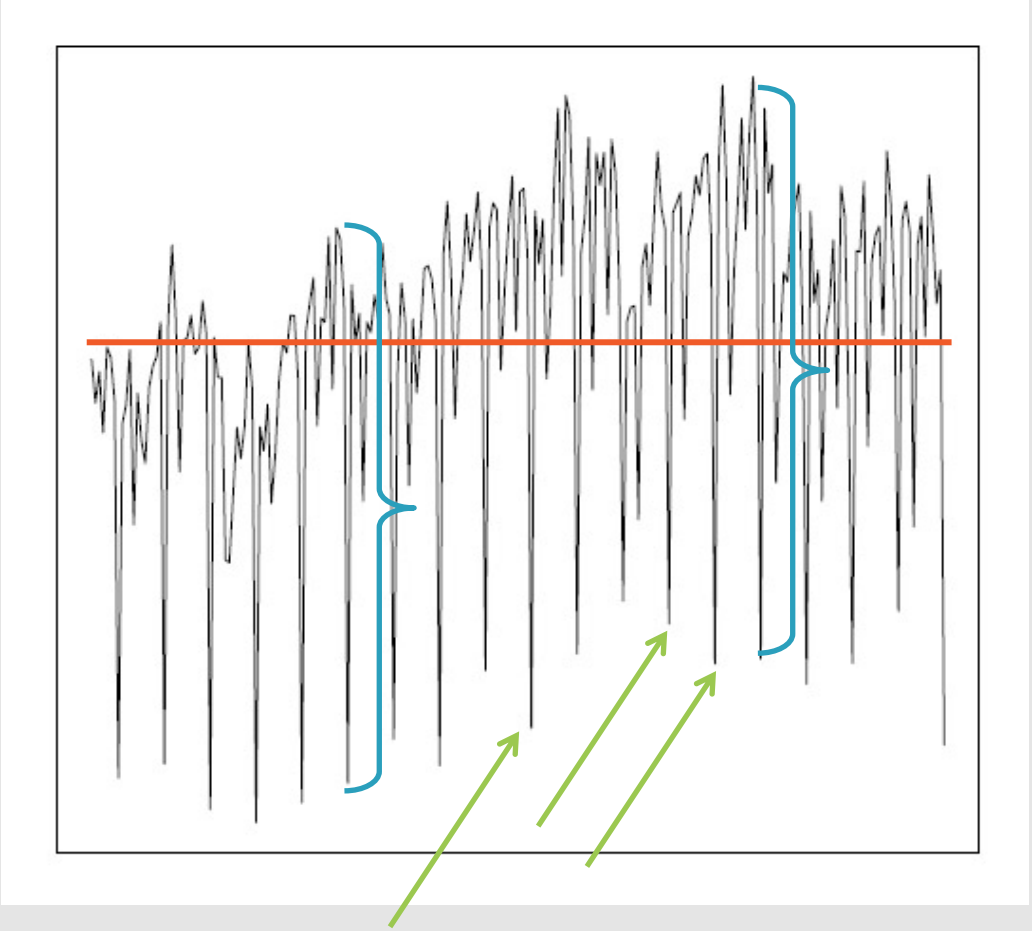
Seasonality

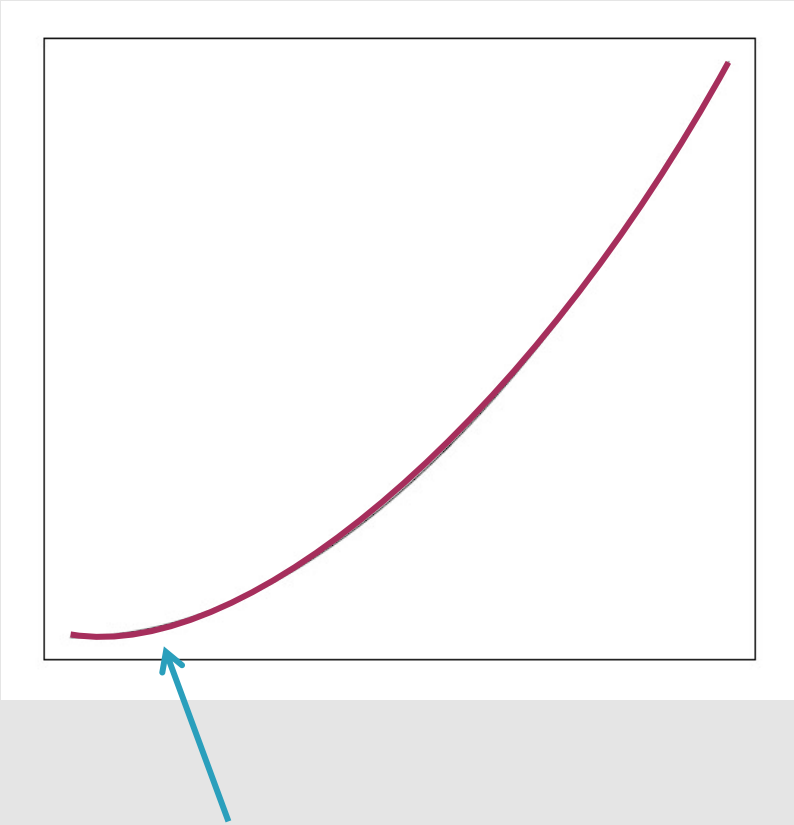
Constant mean

Constant variance

Non-stationary

Autocorrelation is present





Trend with exponential curve

Changing mean

Changing variance

Non-stationary

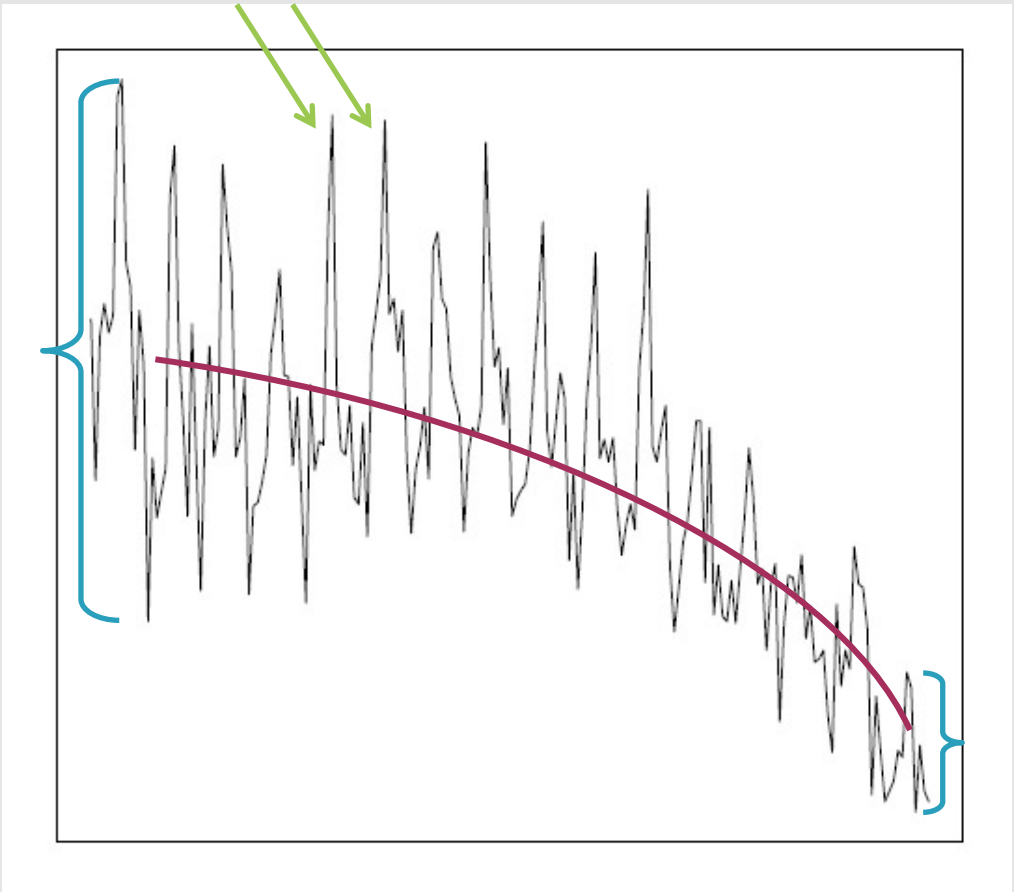
Transformation is required

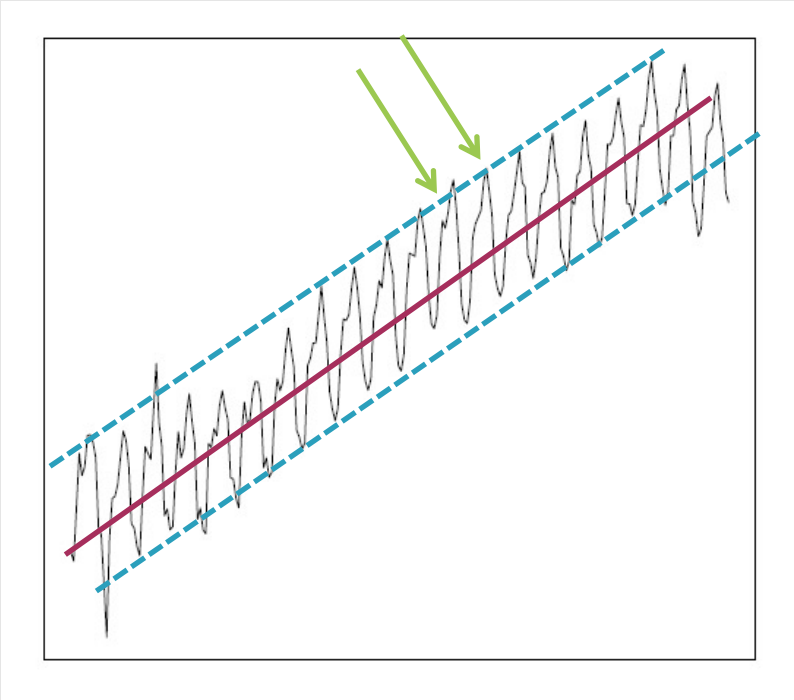
Seasonality

Changing variance

Trend

Non-stationary





Seasonality

Constant variance

Clear trend

Non-stationary

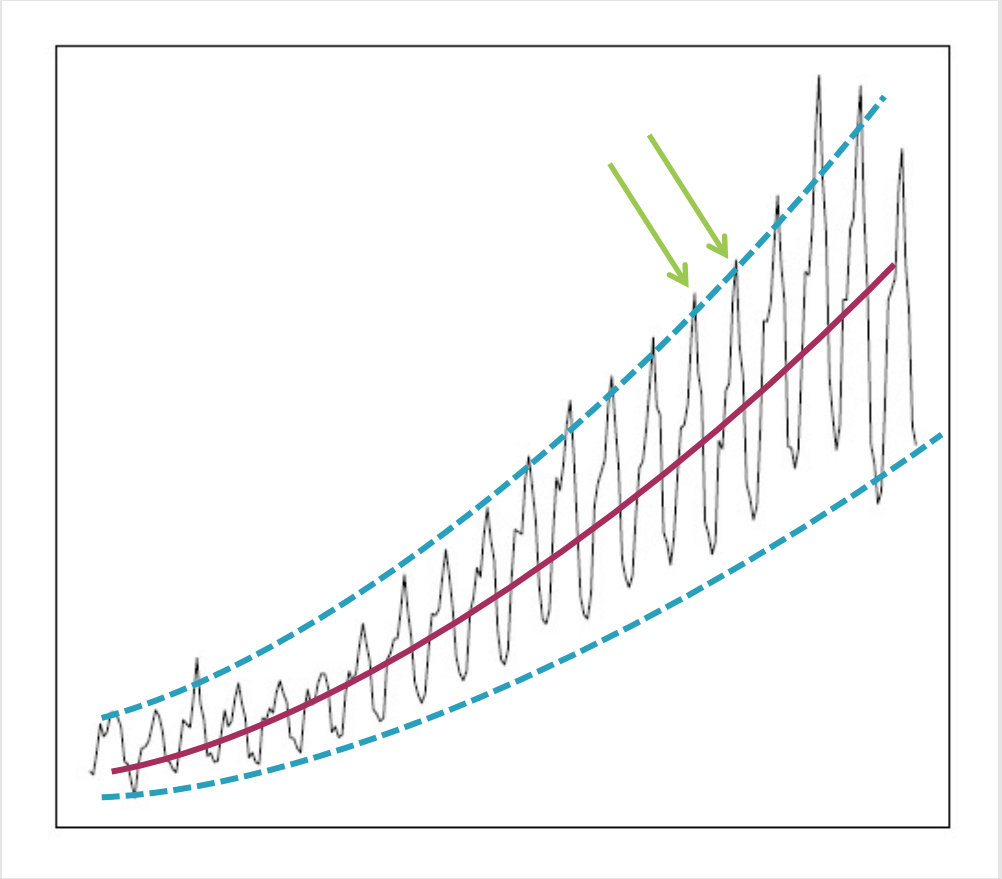
Autocorrelation

Exponential seasonality

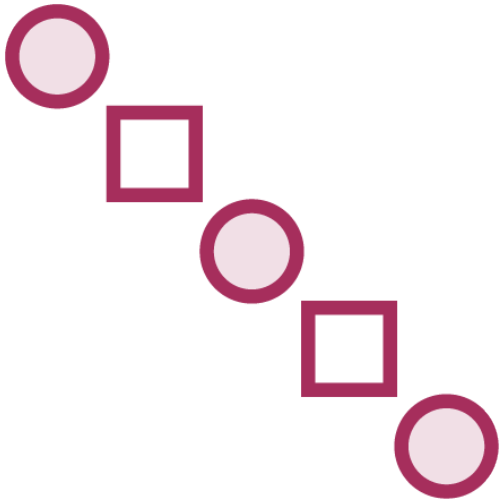
Exponential trend

Changing variance

Non-stationary



Classification of Time Series Data



Patterns

Learn and recognize the most frequent ones

Tools and methods

Transform the data before further analysis

Visualizing Time Series Data

Makes presentation easy

Supports understanding of patterns

Demo

- Function plot() of R Base
- Line graphs
- Basics of plotting time series data
- No advanced packages (besides forecast)

Useful Functions

`plot()`

`decompose()`

`plot.ts()`

`ggseasonplot()`

Autocorrelation in time series

Previous observations influence the later ones

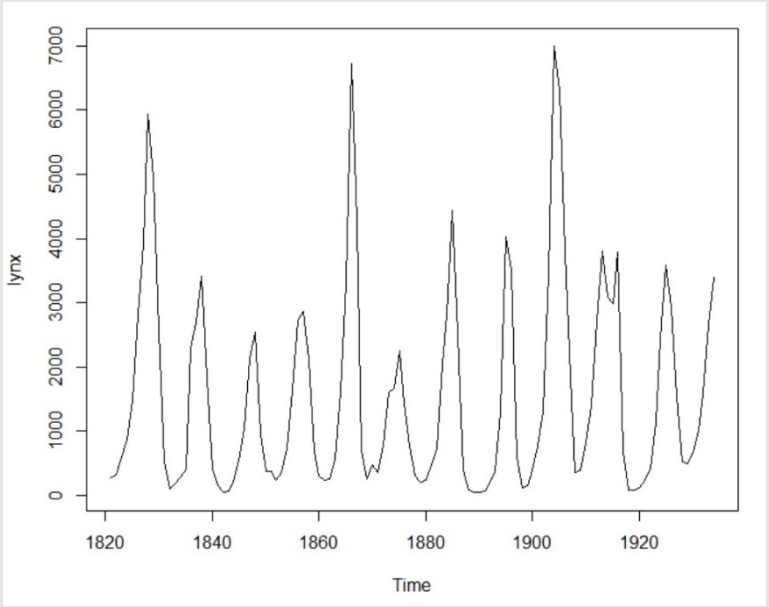
- Correlation between the lags

R tools for identifying autocorrelation

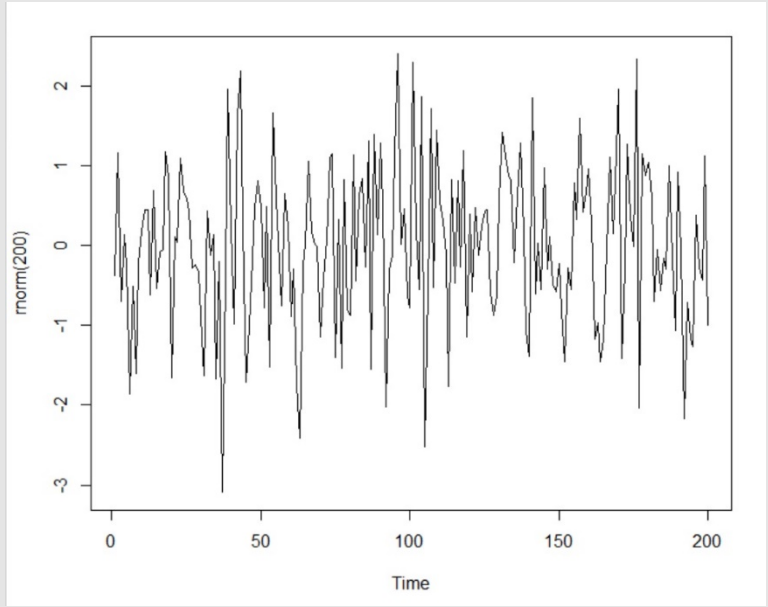
- Autocorrelation is often obvious to see

Classic dataset: Lynx trappings

Autocorrelation Pattern



‘Lynx’ dataset
with autocorrelation present



Random dataset with no
autocorrelation

Identifying Model Parameters

Function acf()

Shows the autocorrelation

Function pacf()

Shows the partial autocorrelation

Two Definitions to Remember

Autocorrelation

The correlation coefficient between lags of the time series

Partial autocorrelation

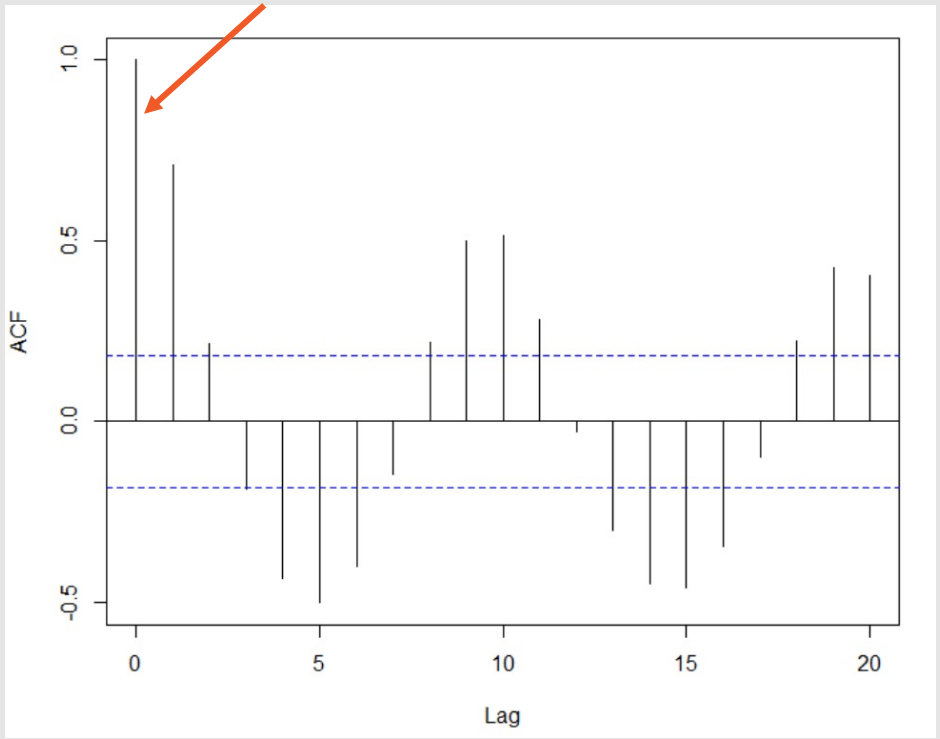
The correlation coefficient adjusted for shorter lags

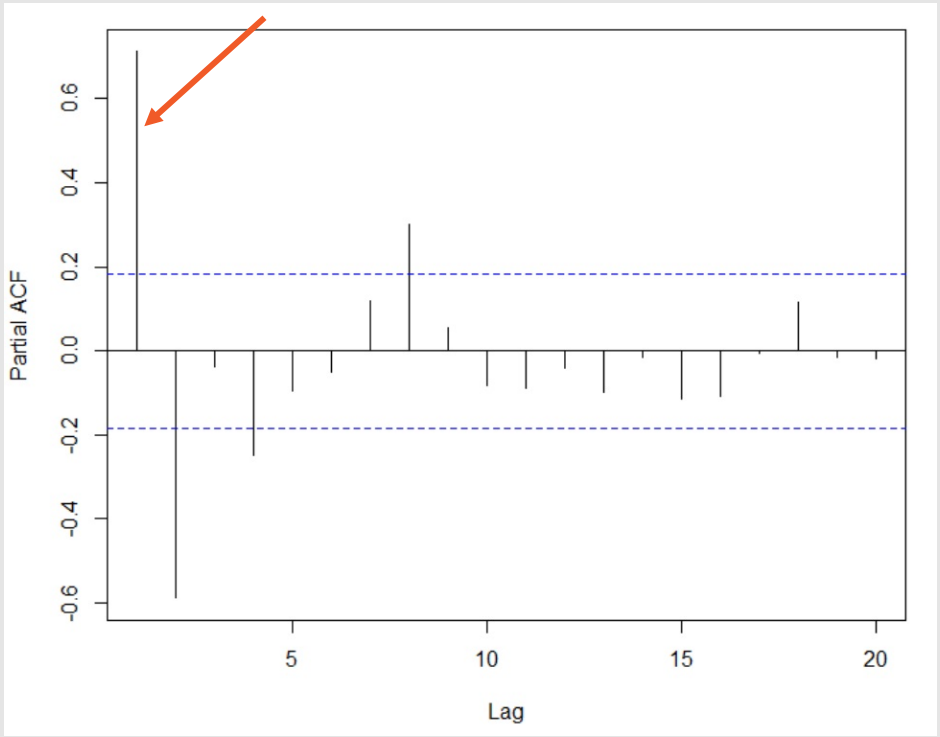
```
acf(
    lynx,
    lag.max = 20)


pacf(
    lynx,
    lag.max = 20)
```

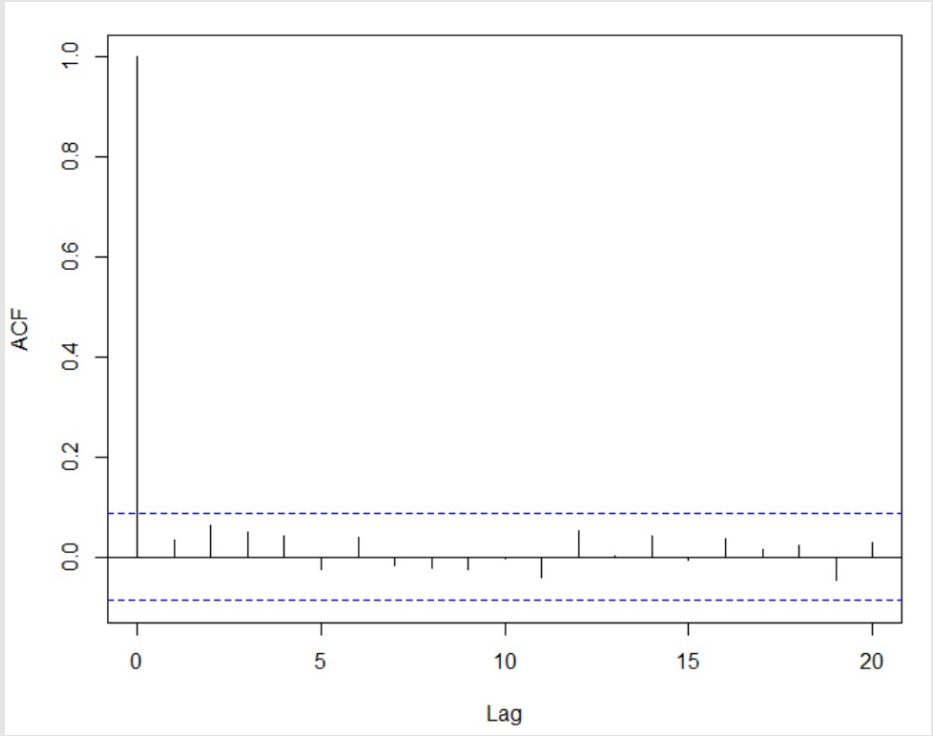
- Testing for autocorrelation and partial autocorrelation
- Use the functions `acf()` and `pacf()`
- Dataset of lynx trappings
- Maximum number of lags
- `'Plot = FALSE'` suppresses the plot and returns the coefficient values only

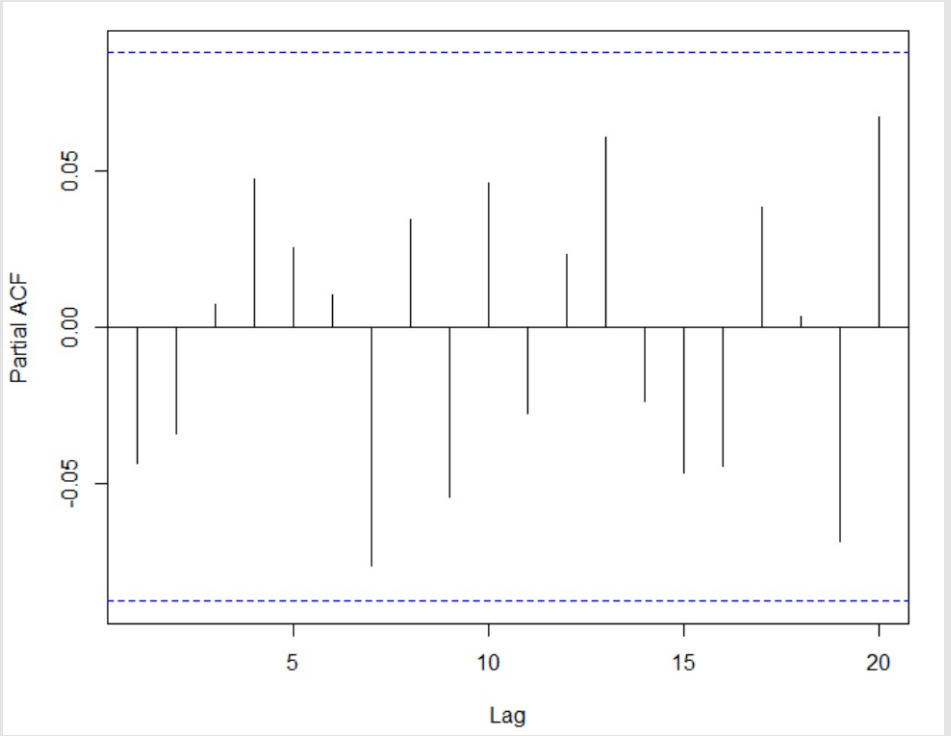
ACF and PACF of ‘Lynx’



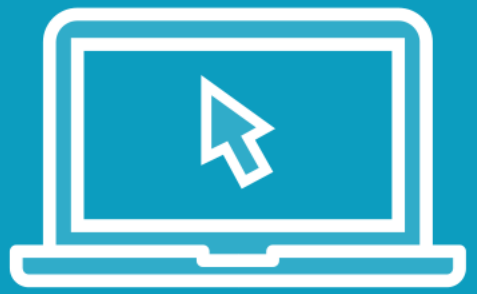


ACF and PACF of ‘rnorm’





Demo



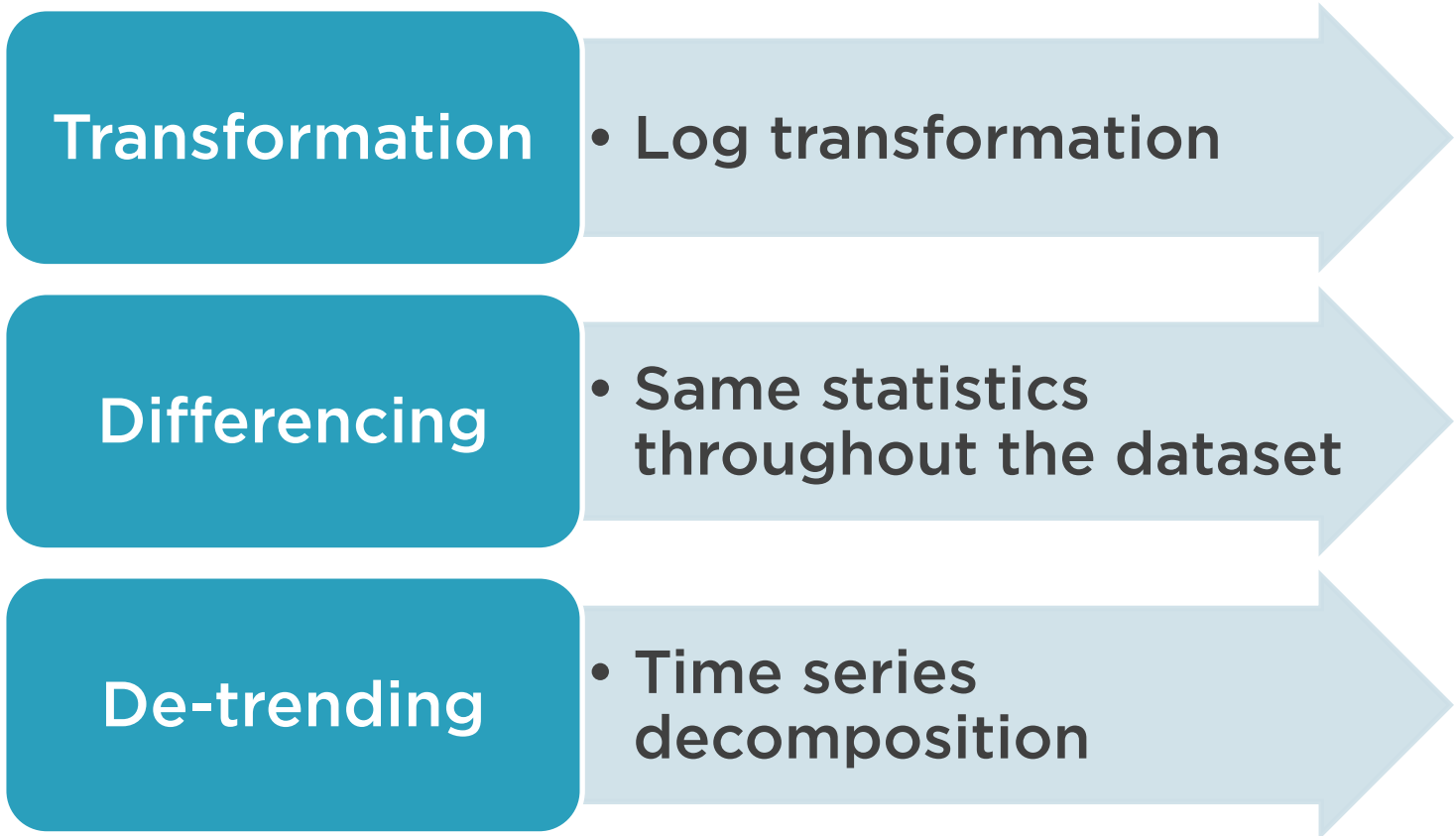
Testing for stationarity

Has the data the same statistical properties throughout the time series?

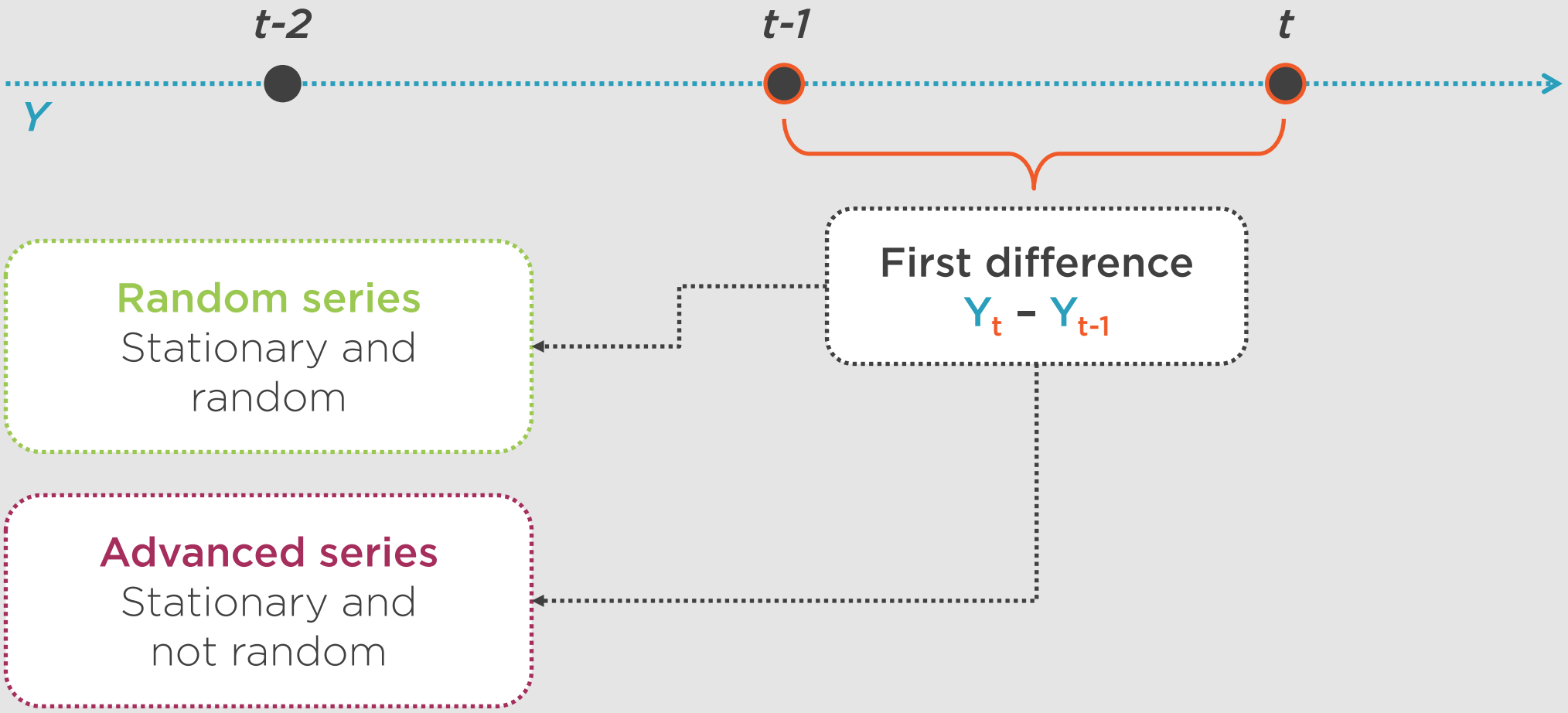
Stats:

- Constant mean,
- Constant variance,
- No autocorrelation

What to Do with Non-stationary Data?



Differencing on the Lag of One



Unit Root Tests

Unit root tests check if a dataset is stationary and if differencing is required

The Augmented Dickey-Fuller test removes autocorrelation and tests for stationarity

Statistical Background of Time Series Data

Function ts()

Data visualization

Stationarity

Patterns

Differencing

Autocorrelation