Mining Data from Time Series

INTRODUCTION

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

Introduction

Course orientation and expectations

Background of time series analysis and forecasting

Python for time series analysis

Course datasets explained

Data import and time series formatting

Managing Expectations

Course Requirements

Anaconda Distribution

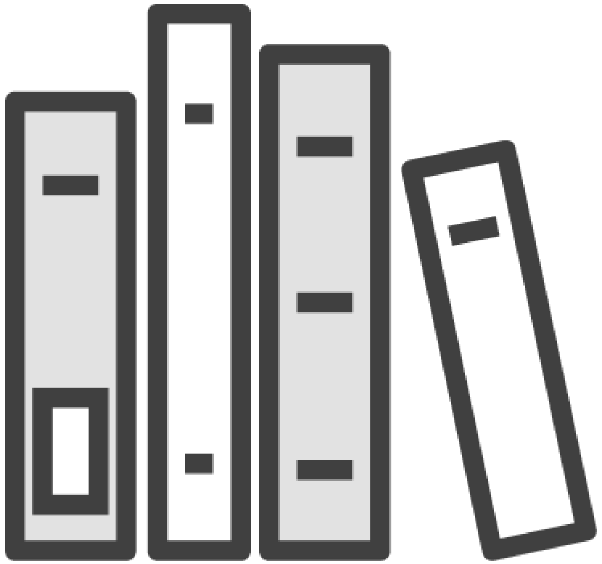
Includes Python 3, Jupyter Notebook and statistical modules

Python Skills

Installing modules, understanding of syntax, general orientation

Beginning Time Series Analysis and Forecasting with R

by Martin Burger



Models for univariate time series data

Theory and implementation in Python

The statistics of time series:

- Stationarity, autocorrelation, smoothers
- Data visualization techniques

Non-seasonal ARIMA models

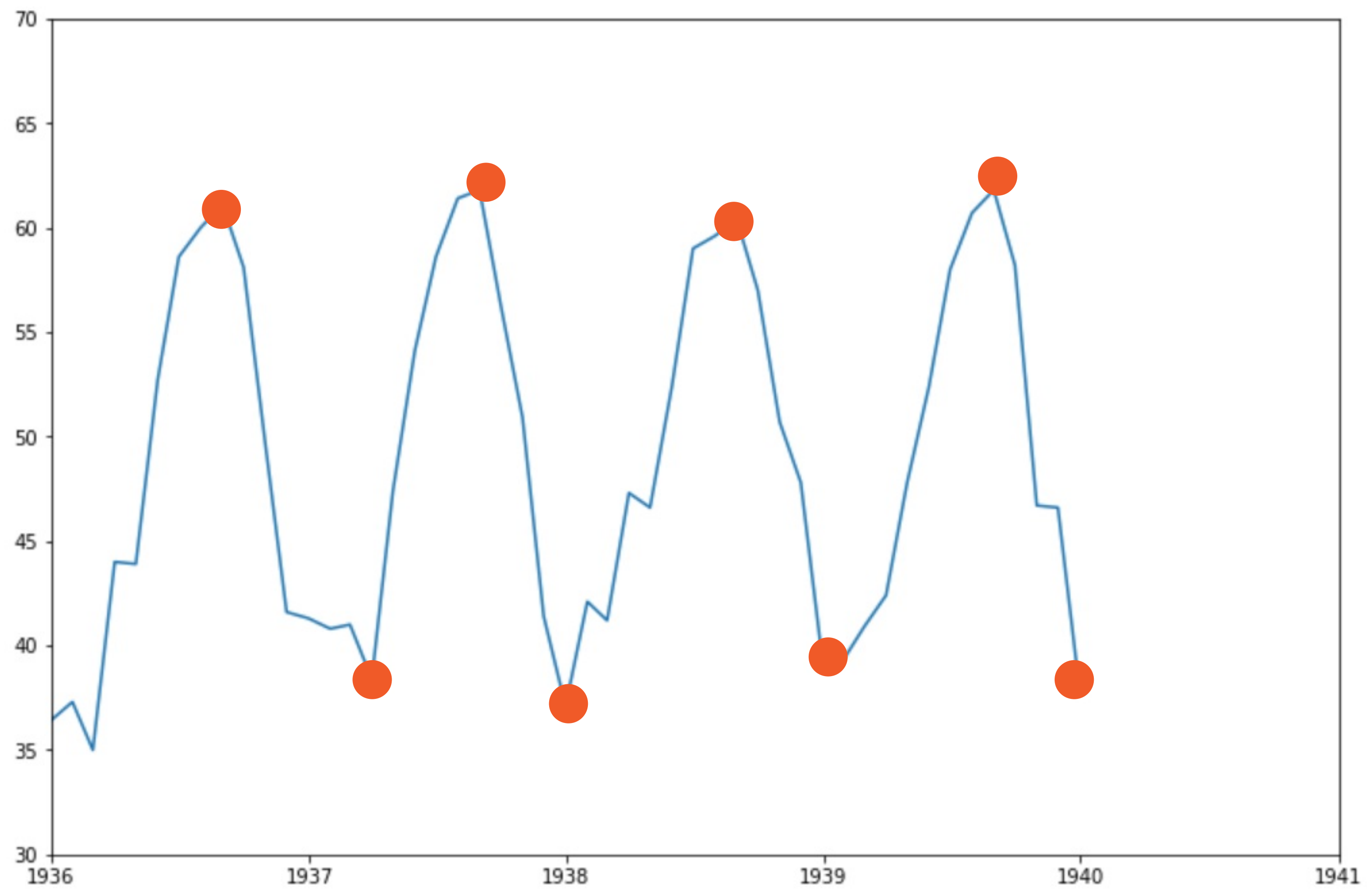
Seasonal ARIMA models and seasonal decomposition

Exponential smoothing

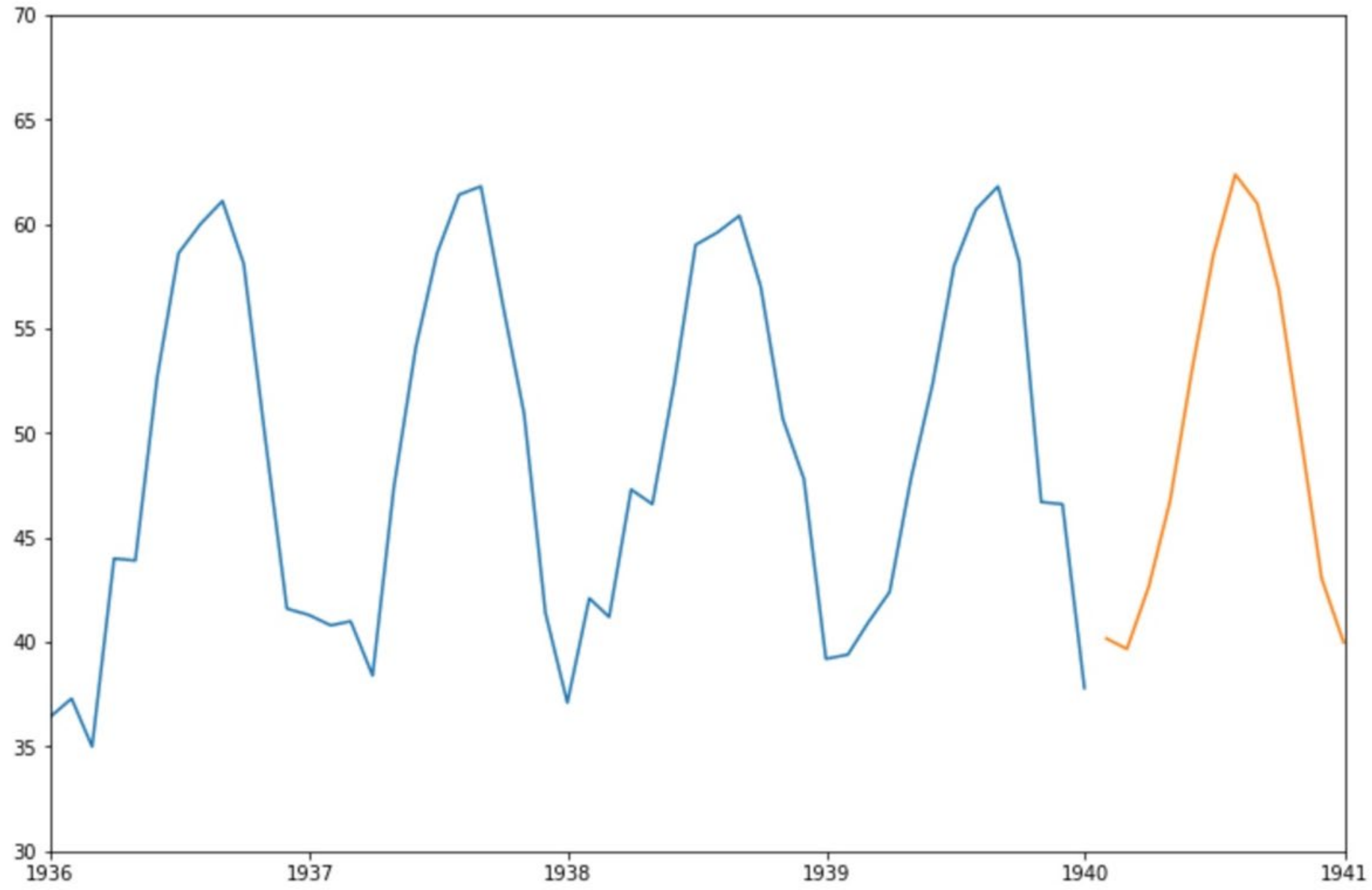
Advanced modeling techniques and tools

Time Series Analysis and Forecasting Basics

Extrapolating Patterns into the Future

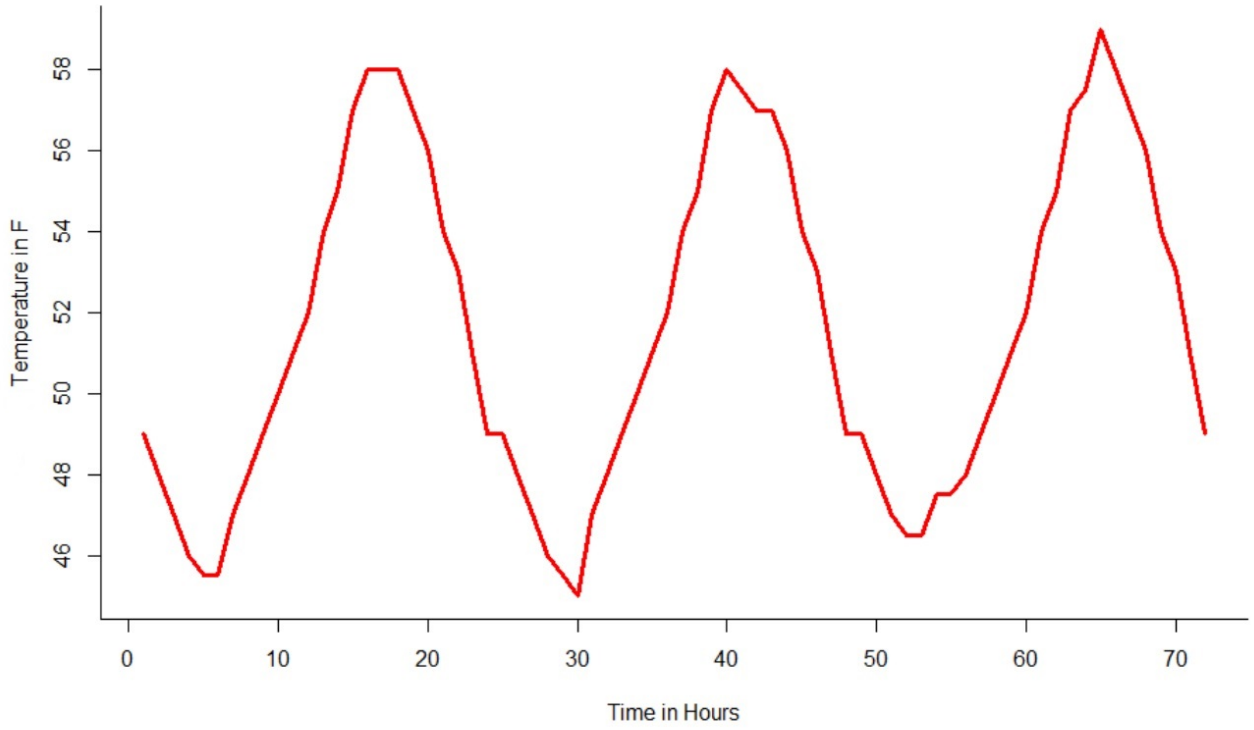


Extrapolating Patterns into the Future



Plausible forecasts require clean patterns

E.g: Seasonal cycles



Factors facilitating a successful forecast

The amount of data

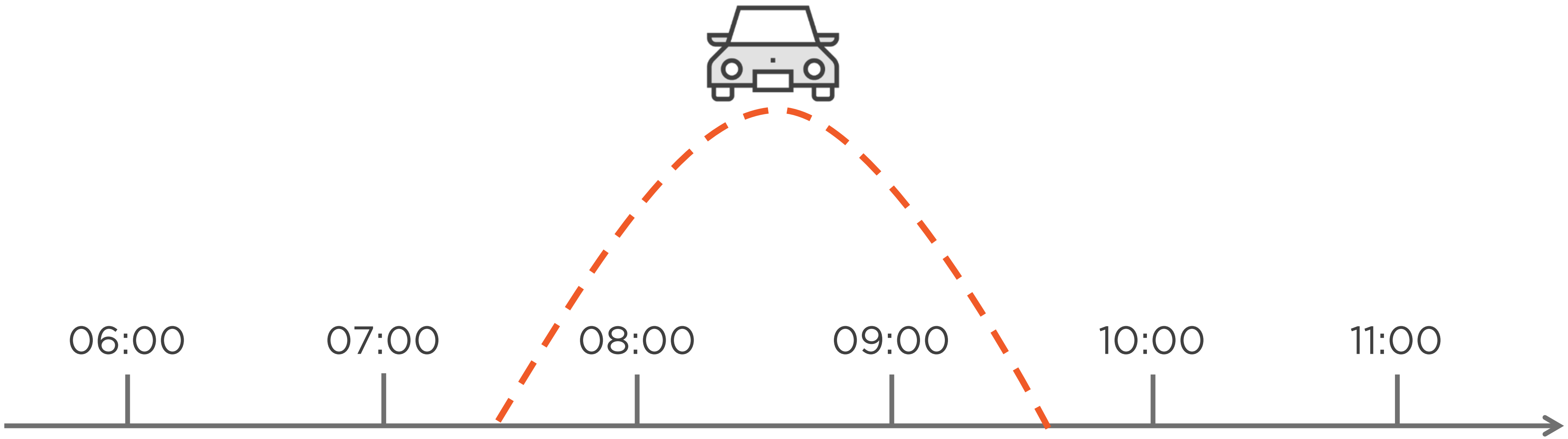
- Is there a minimum requirement?

Clarity of patterns: Regular intervals and distinct characteristics

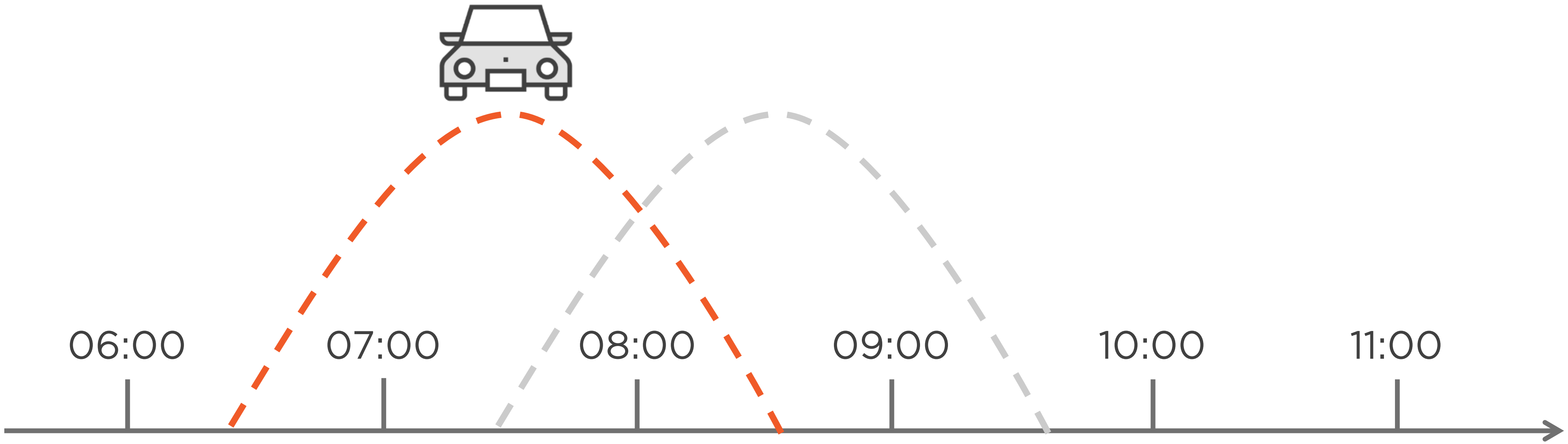
- Clear pattern: Temperature measurements
- High degree of randomness: Stock prices

The effect of the forecast on the time series

The Forecast Influences Future Data



The Forecast Influences Future Data



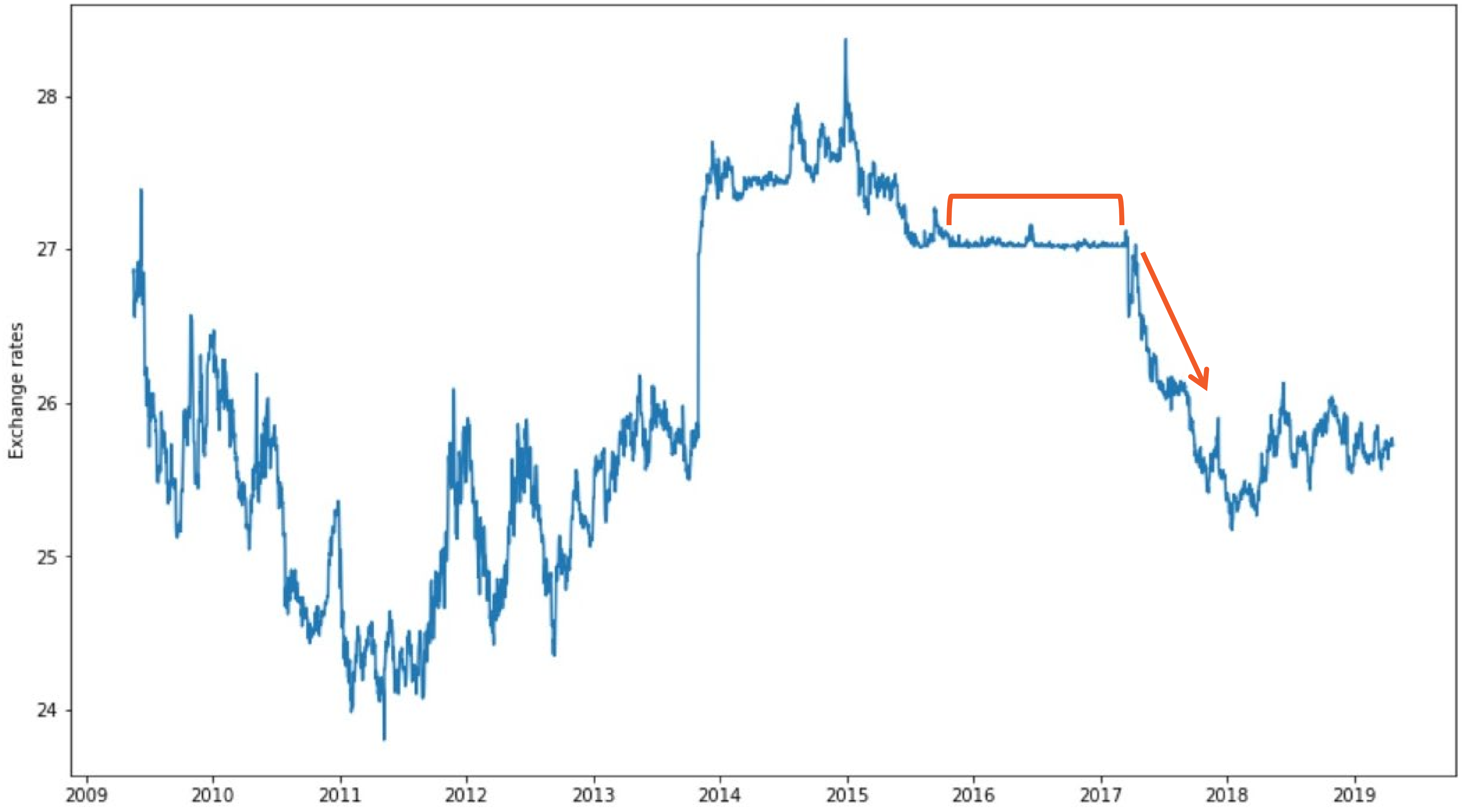
Scrutinizing the Reliability of Patterns

Distinguishing real patterns from randomness

Detecting changes in pattern over time

Exploring the background of the time series

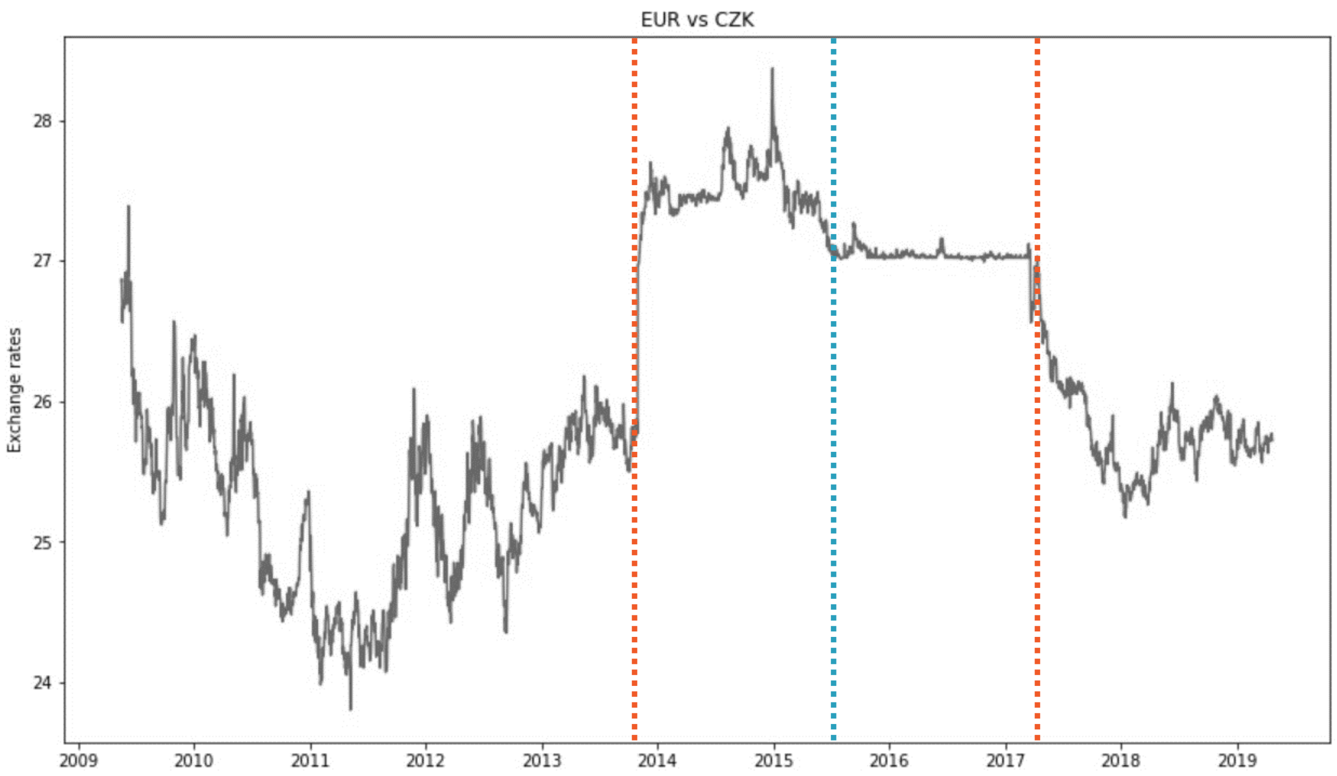
EUR-CZK Historical Exchange Rates



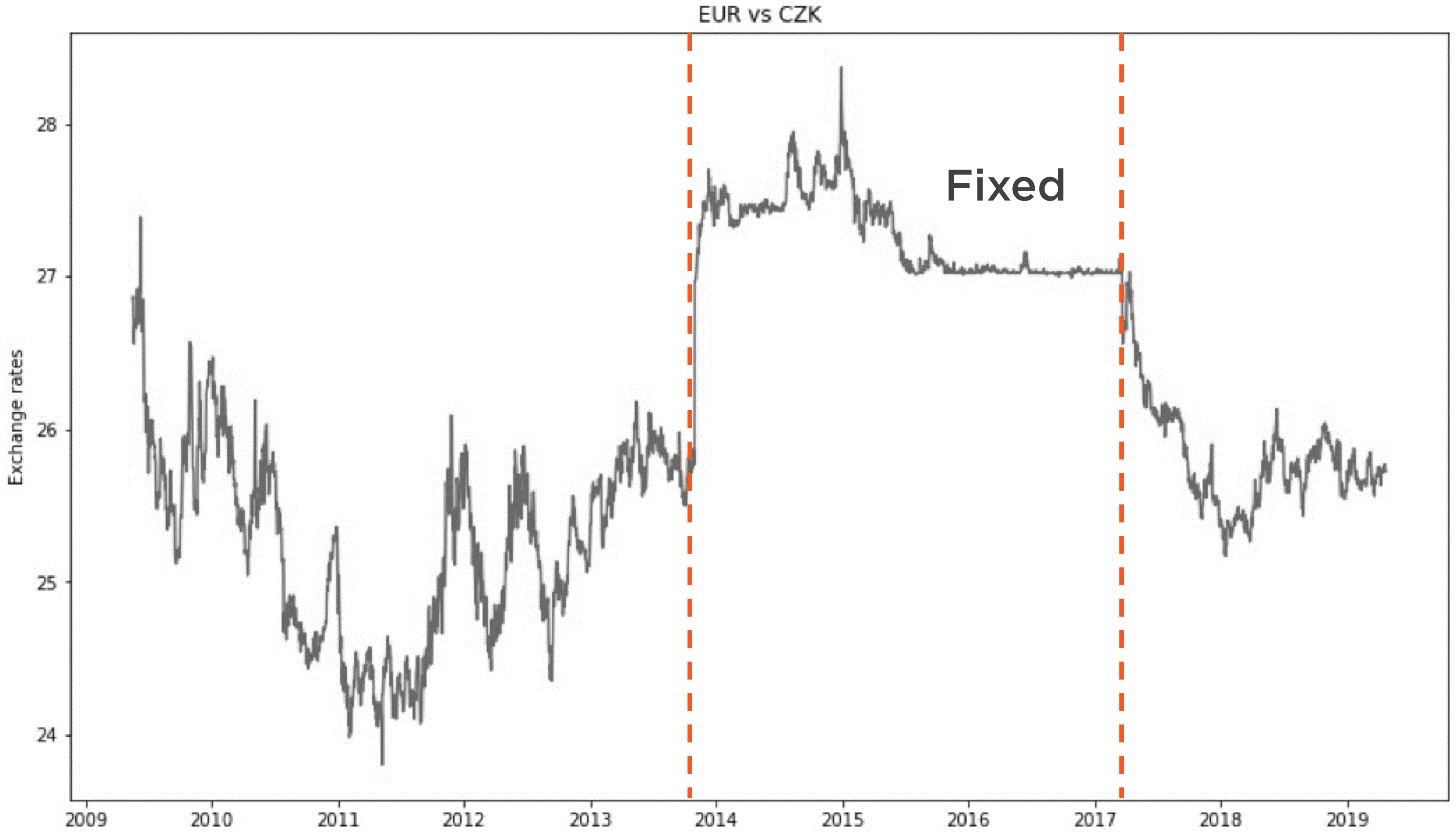
Stop of monetary pegging at Apr 2017

Start: End of 2013

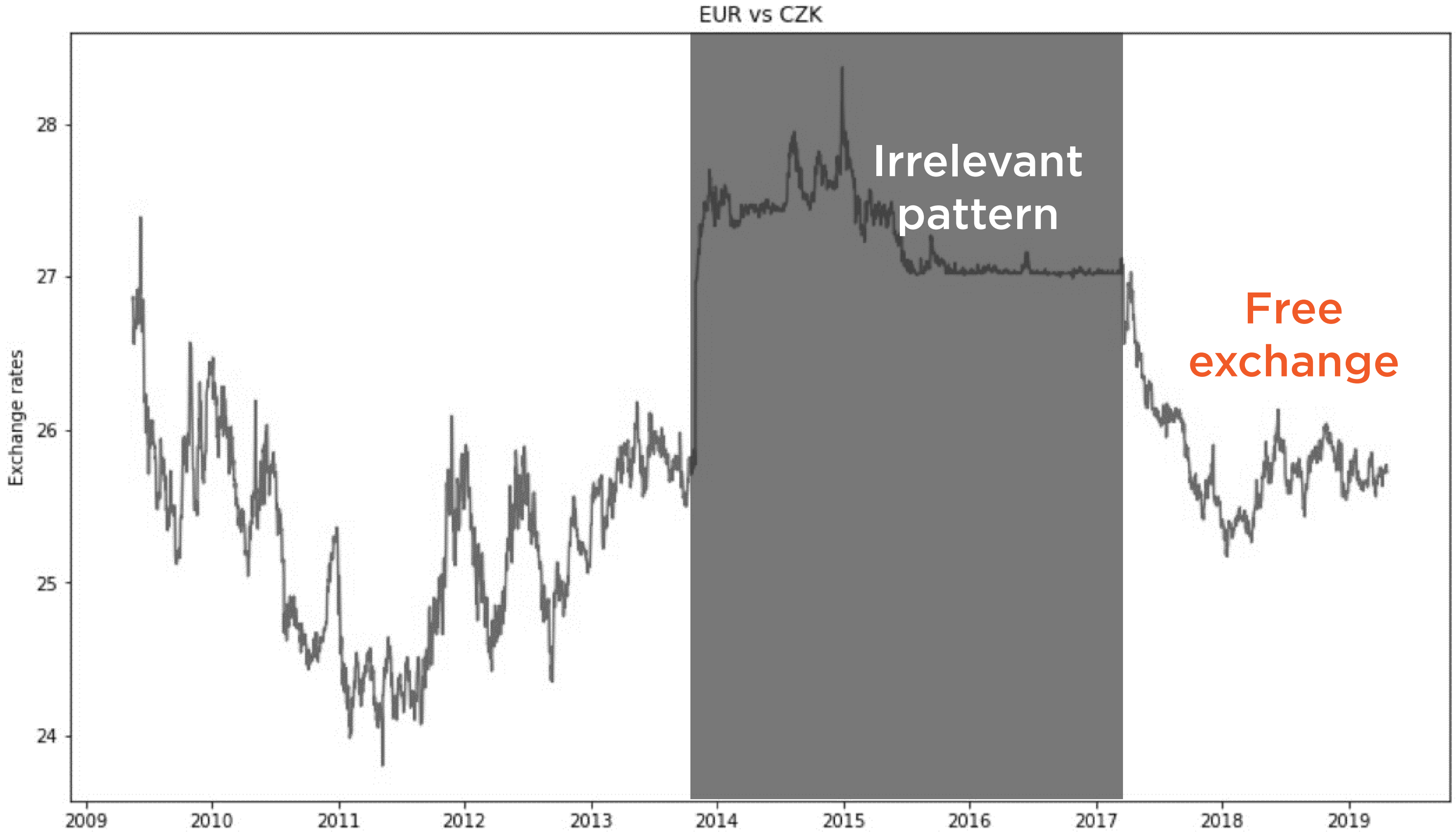
Full fruition from mid 2015



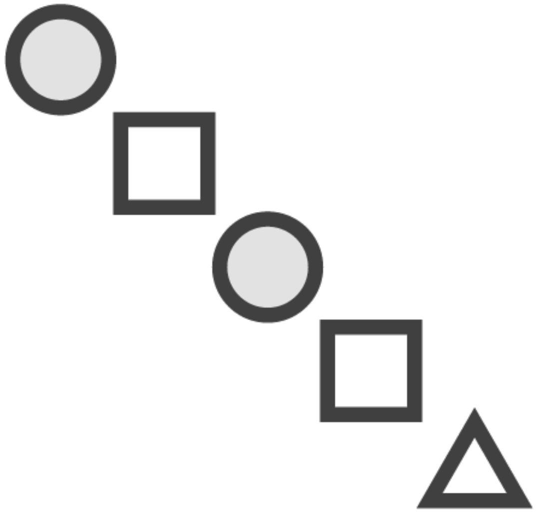
Change in Pattern



Change in Pattern



Reveal the Background Behind Patterns



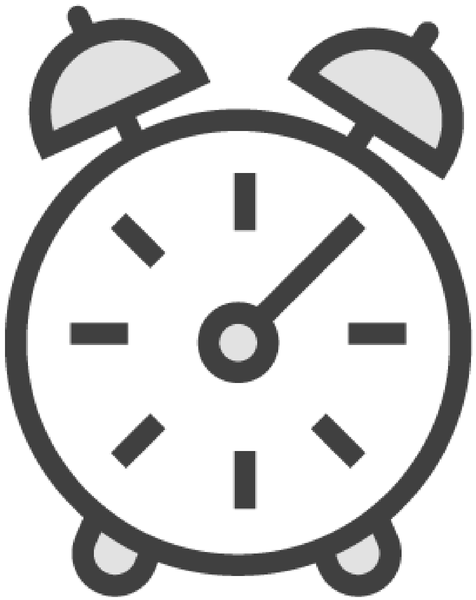
Predictions should be made on data collected under the same rules

Awareness of manufactured changes in patterns

Models should put more weight on most recent events (e.g. exponential smoothing)

The Time Frame of the Forecast

The Time Frame of the Forecast



The longer the forecast the less accurate it gets

Time frames: Short, medium and long term

The actual length of the time frame is relative

Deriving decisions based on different time frames

Short term: Business week

- Staff allocation and restocking based on high and low sales days

Medium term: Quarter year

- Hiring staff and increase of certain supplies

Long term: Years

- Strategic planning to increase sales capacities
- Example: Enlarging store space or opening of new stores

Run several forecasting methods till they give approximately the same results.

Forecast Preparation Steps for Best Results

Time Frame

Short, medium or long term

Key Metrics

Measures pointing towards the goal of the analysis

Insights from Experts

Gain perspective on the problem

User and Delivery

Autopilot or on-demand reports

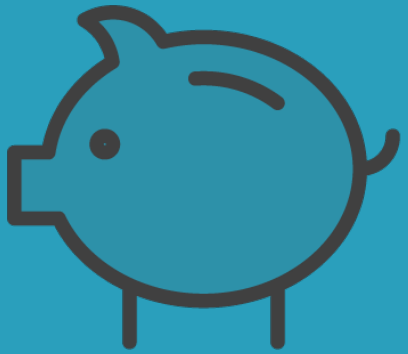
Python for Time Series Analysis

Reasons to Choose Python

Helpful user community

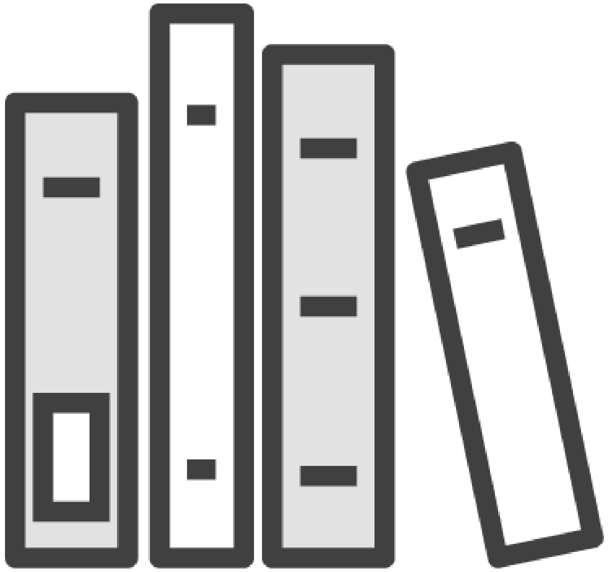
Good documentation

Multipurpose tool



Open source

Python Modules



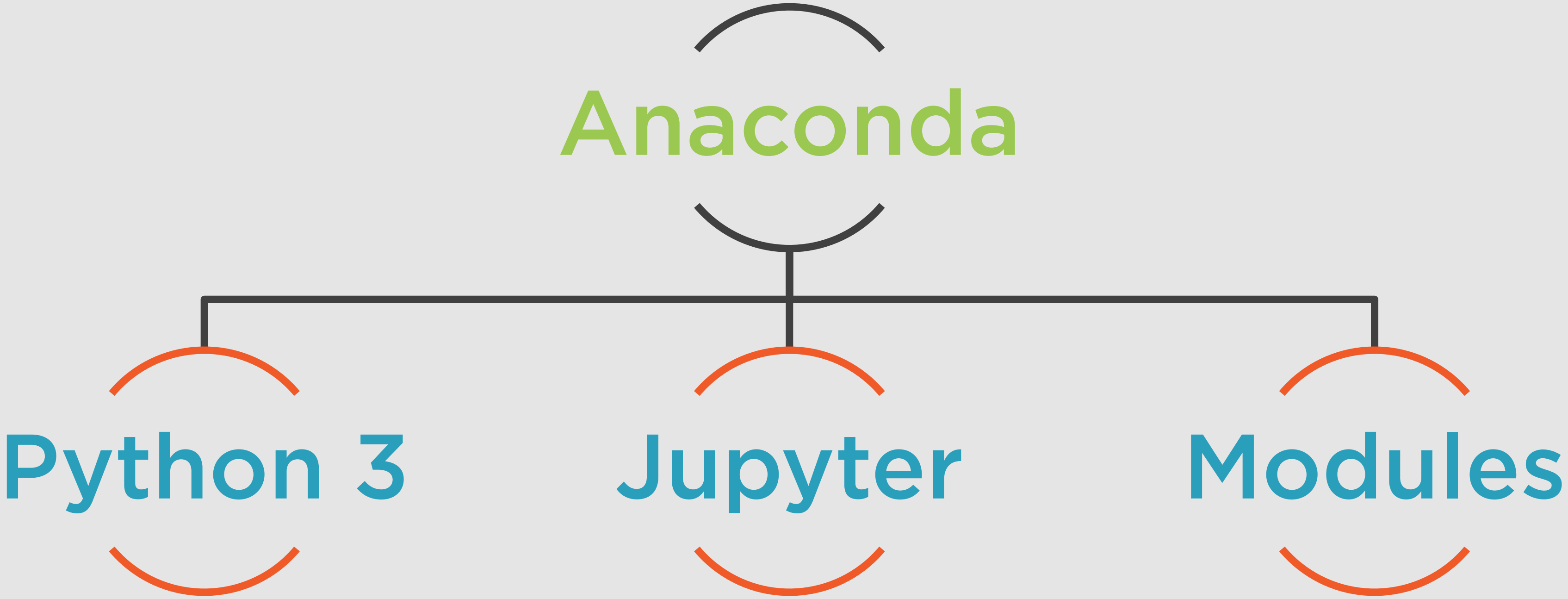
Statistical toolbox: StatsModels library

- TSA module for time series analysis

Third party modules: STLdecompose, pmdarima

Modules pandas, NumPy, Matplotlib

About the Environment



Do I Need R Besides Python?

R has better a documentation and more tools for time series analysis

Most analytical approaches are covered in Python

Integration of R in Python is possible (rpy2 module)

Datasets

Datasets

Lynx

Yearly lynx trappings in Canada

Nottem

Monthly temperature averages in Nottingham, UK

Dataset: LYNXdata.csv

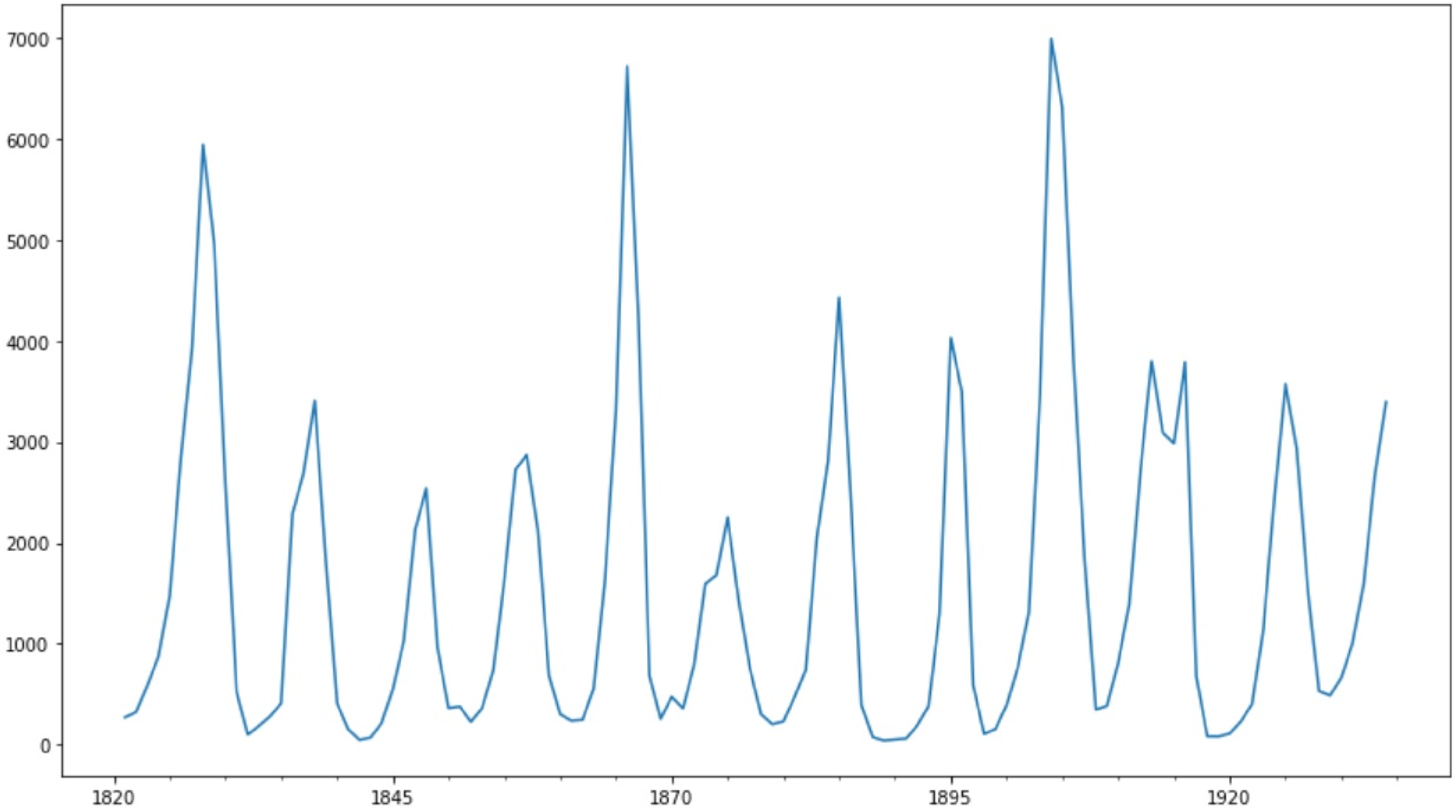
Yearly lynx trappings in Canada between 1821-1934

- Length: 144 observations

Seasonal pulses: Predator-prey relationship (autocorrelation)

- High population of predators decimates the prey population
- Food shortage causes a decrease in the predator population
- High number of trappings in one year means less lynx available in the following years

Seasonal Pulse in Lynx

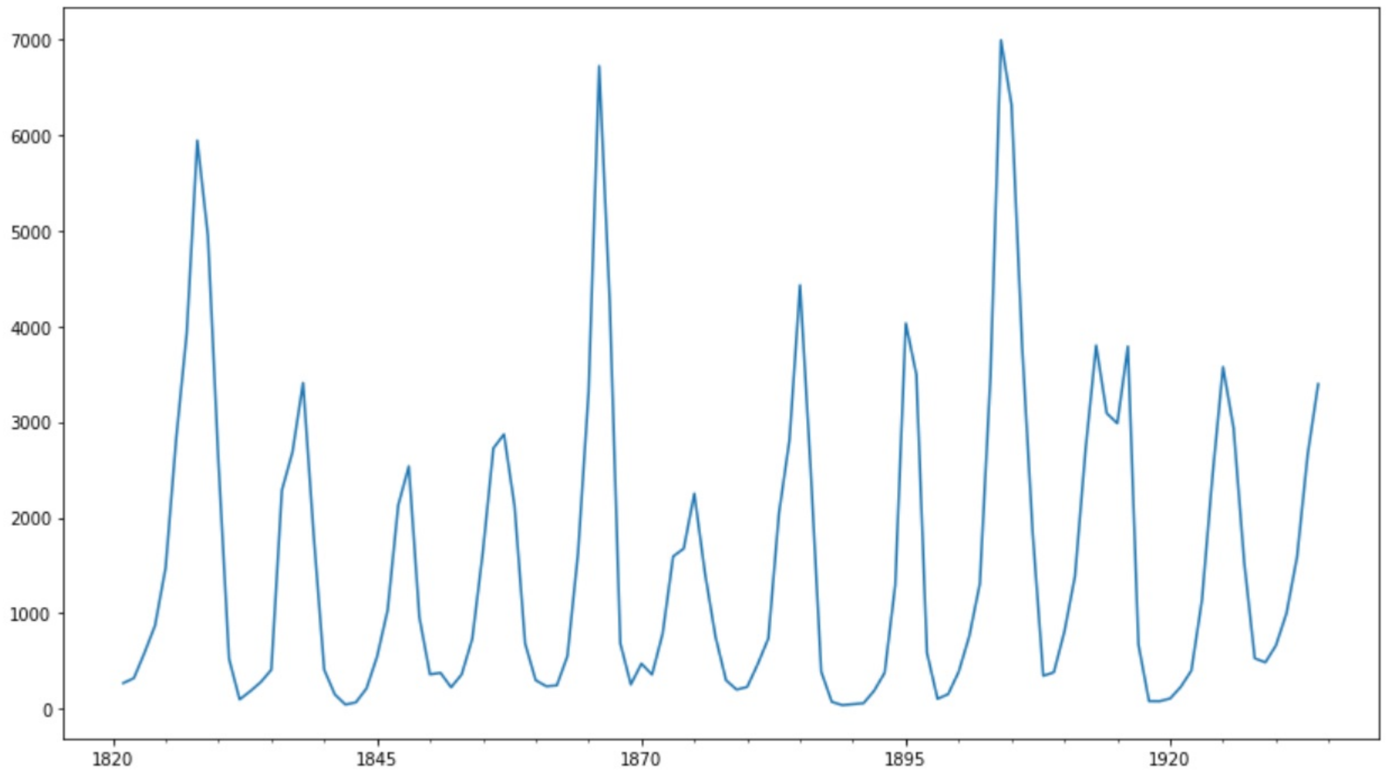


Autocorrelation

Constant mean

Constant variance

No trend



Dataset: nottem.csv

240 observations of monthly temperature averages

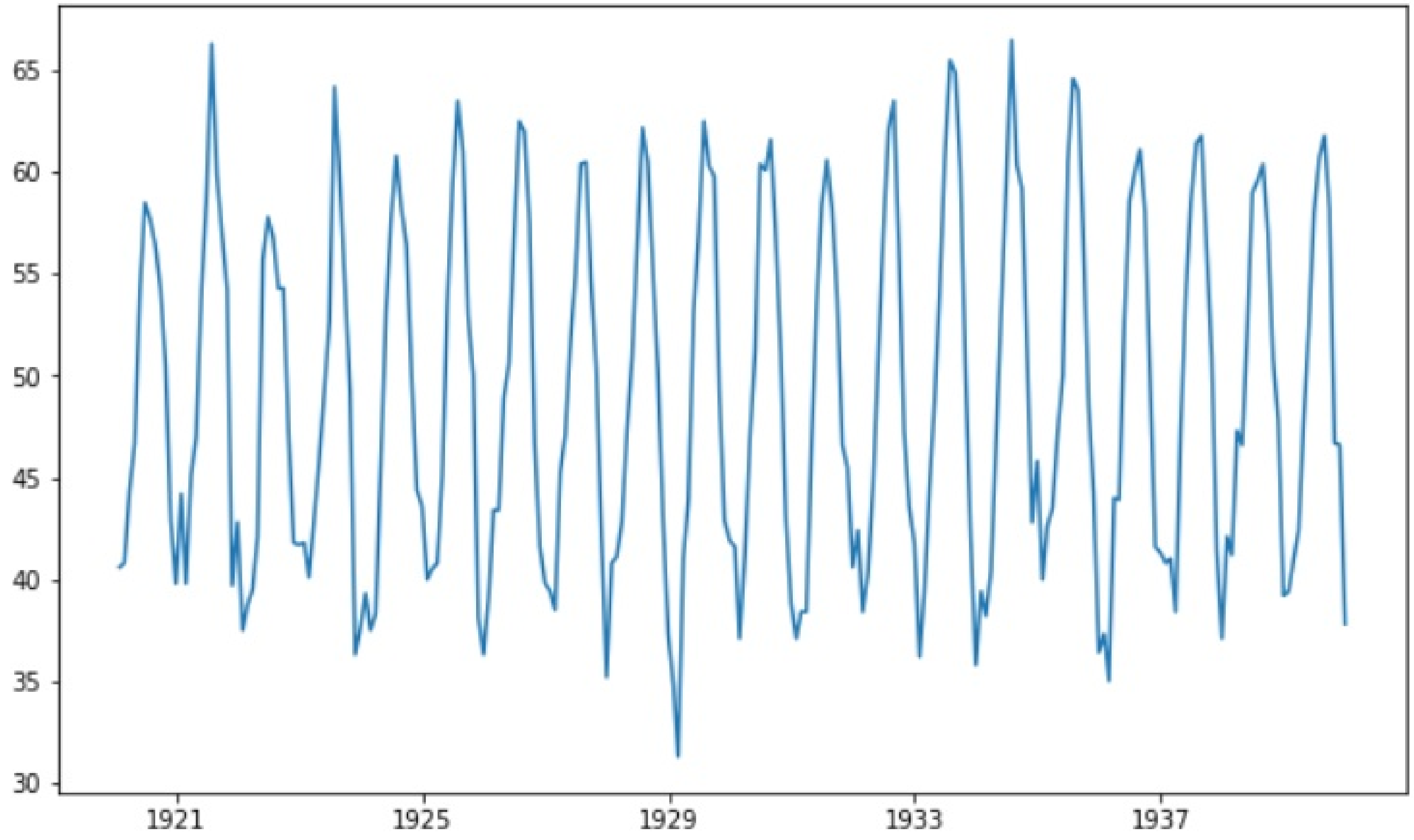
- Jan 1920 – Dec 1939

Seasonal frequency of 12

- Even number of periods, no offset

Clean dataset

Clear Seasonality with a Constant Mean



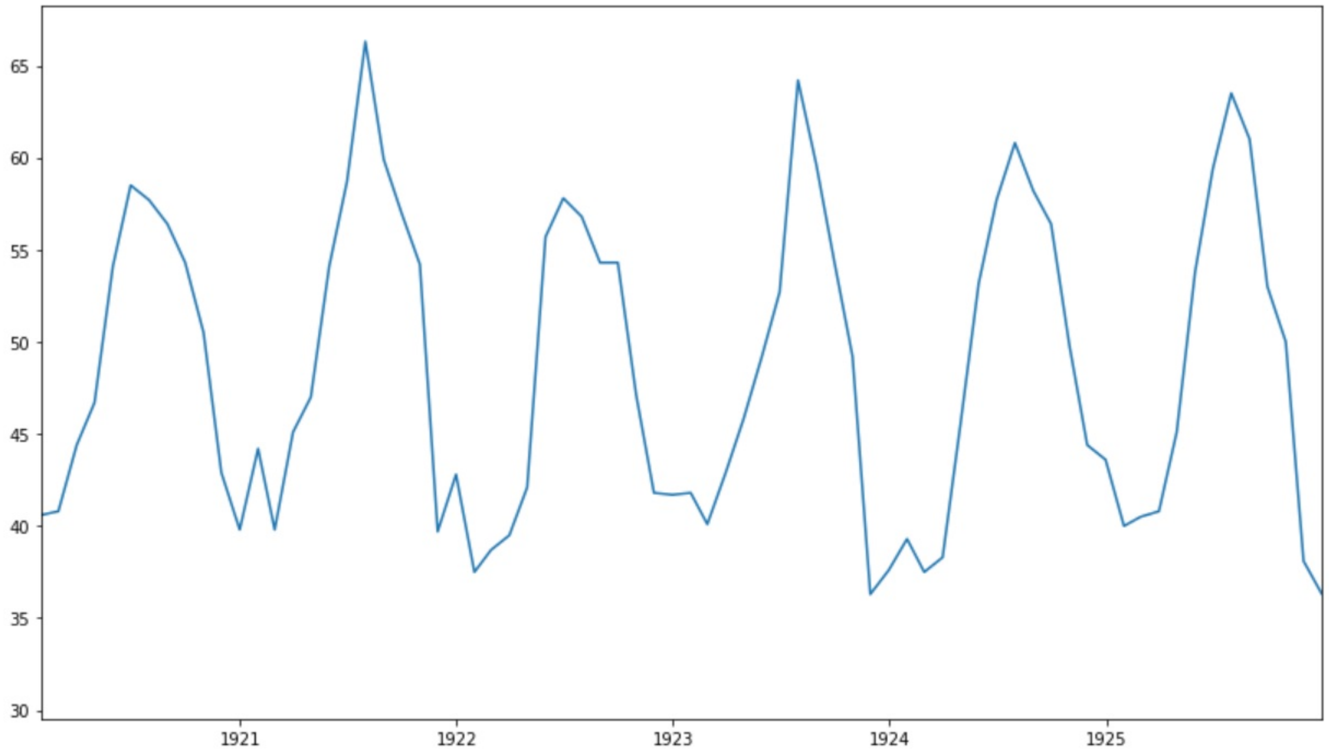
Time Series Vectors and Lags

Time Series of a Specific Order

Time series data has a specific order

Changing the order corrupts the patterns

Time stamp or index



```
pd.Series(data, index = pd.date_range())
```

Attaching the Time Stamp

Index generation with pd.date_range()

Proper date format with frequency

Common Frequency Indicators

‘D’

‘B’

‘H’

‘T’/ ‘min’

‘S’

Introduction

Course structure overview and general expectations

Concepts of time series analysis and forecasting

Relevant packages from Anaconda and third party modules

Importing and formatting the datasets lynx and nottem for later demos