Simulation-based forecasting in the financial industry

Mahdis Rahmani

Topics:

- An introduction to sequential data and time series
- Time series notation and time series assumptions
- Modeling (in order to be able to describe and predict the financial market)
- Some key Concepts
- Plots and tests
- A real world case study
- Limitations of mathematical models
- ML as a substitute?

What is sequential data?

Sequential data refers to a type of data that is ordered or arranged in a specific sequence or pattern.

Whenever the points in the dataset are dependent on the other points in the dataset the data is said to be Sequential data.

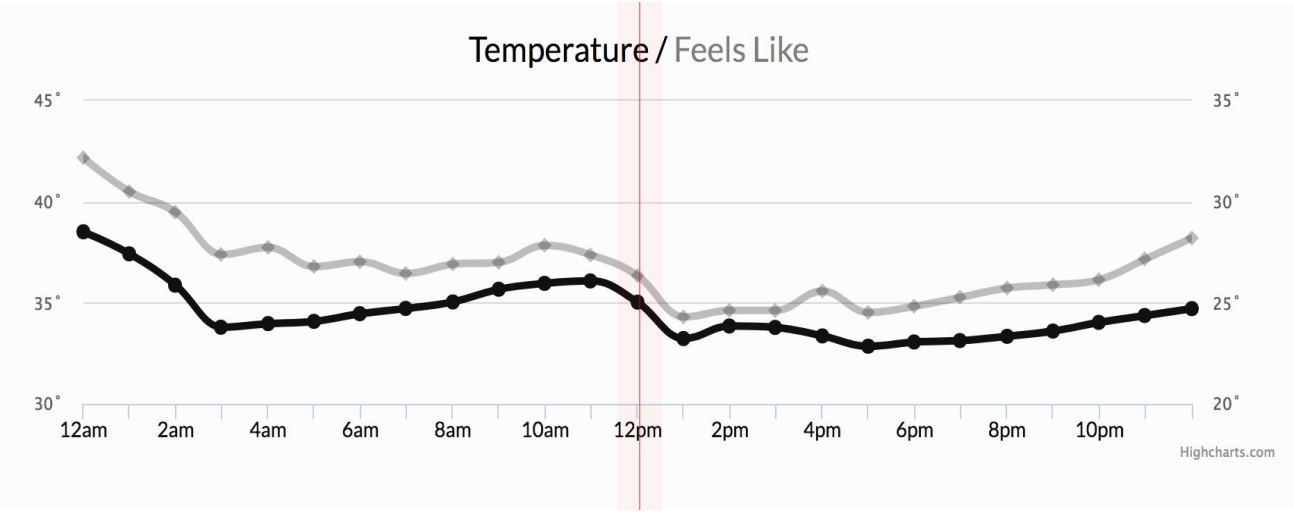
Sequential data includes text streams, audio clips, video clips, time-series data and etc.

DNA is considered a sequential data because it contains information that is arranged in a specific order that can be read and interpreted

Time-series

A serie of data points,
with time order.

Examples:
Weather forecasting,
Stock markets, …



Time series peculiarities

- Time Intervals: intervals must be equal so that we can gain meaningful insights, otherwise we have to make intervals equal:

  +: intervals are monthly but we need an annual report

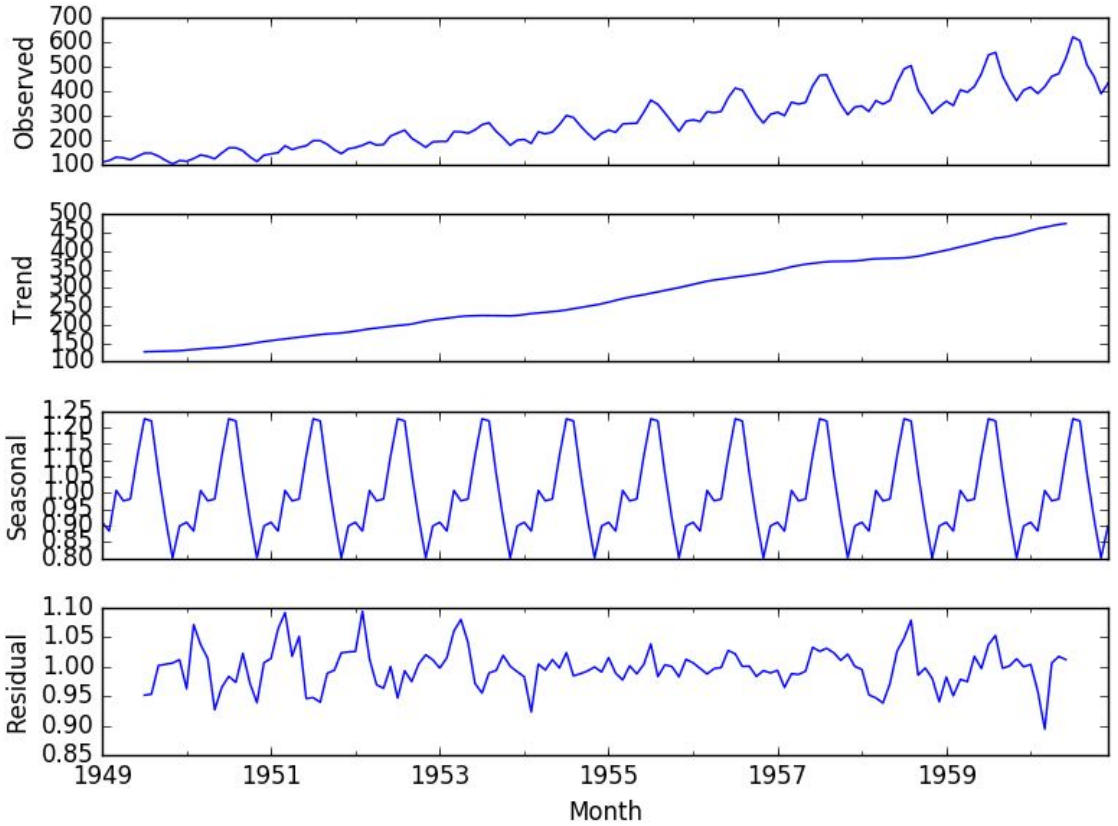
  +: intervals are annually but some years are missing

  Leads to dealing with missing values: bfiling - ffilling - mean - constant

Some Key Concepts:

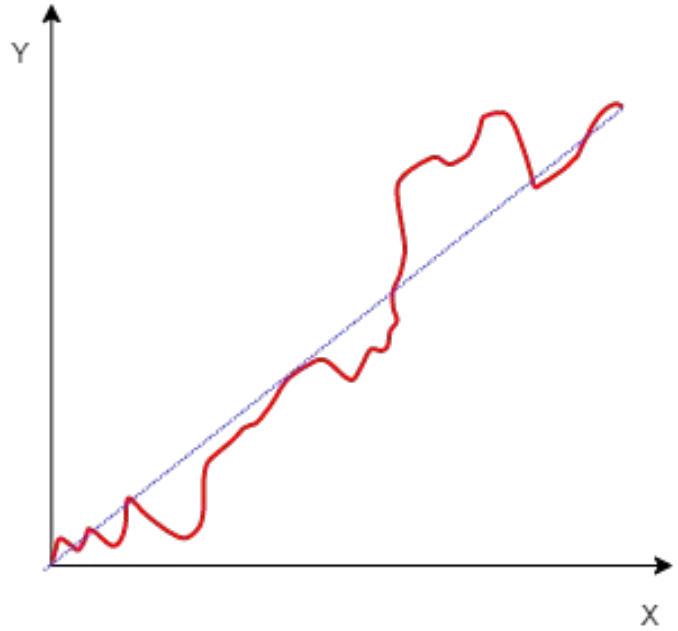
- Trend
- Seasonality
- White noise
- Stationarity
- Random Walk

Visualization of key concepts:



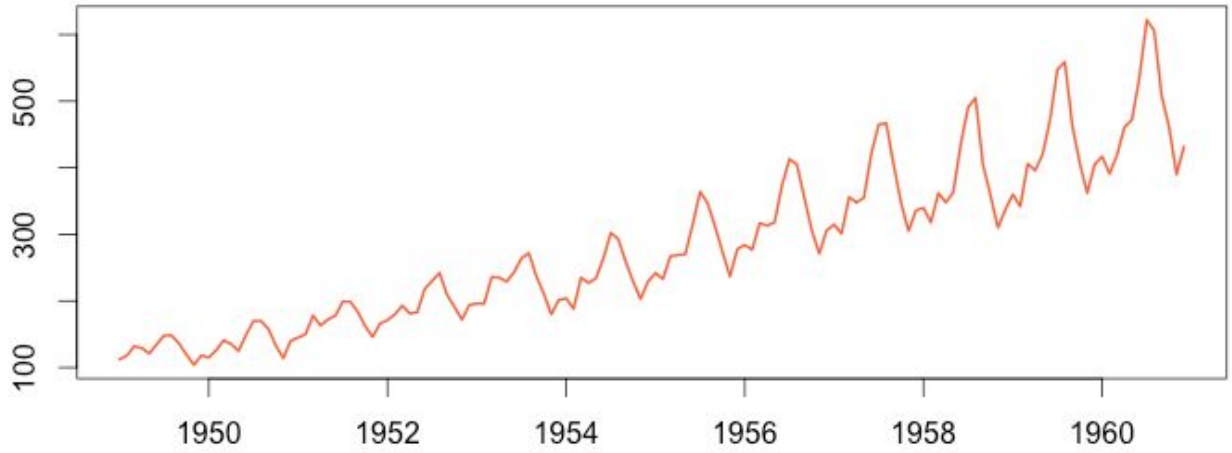
Trend

The trend represents the long-term change in the level of a time series. This change can be either upward (increase in level) or downward (decrease in level). If the change is systematic in one direction, then the trend is monotonic.

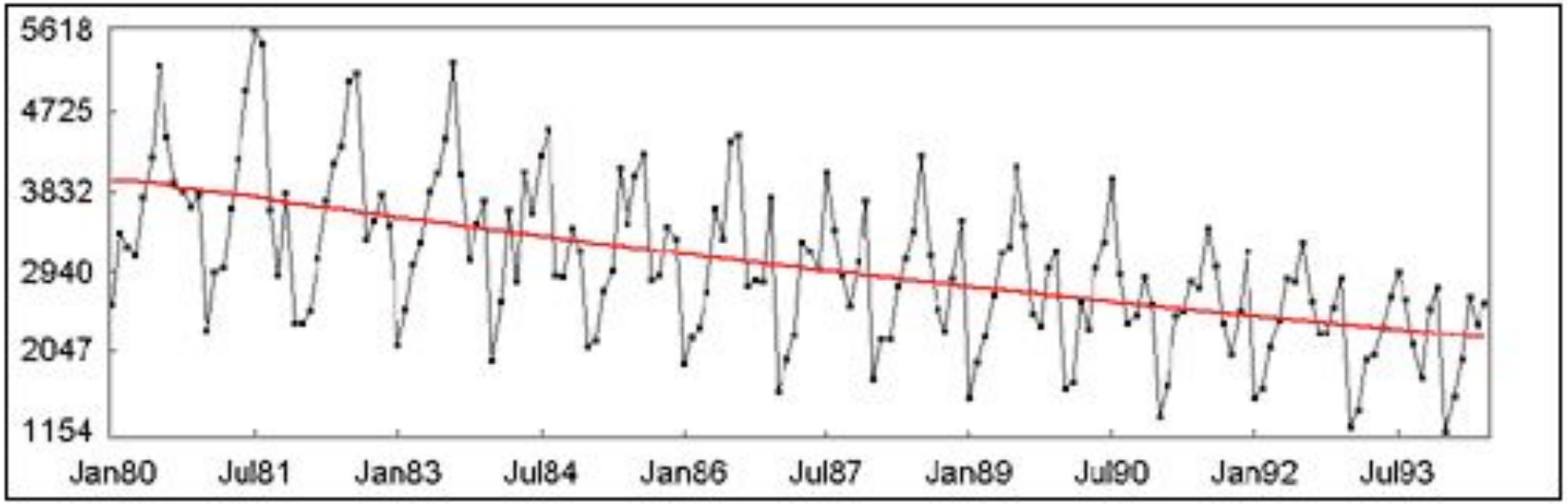


Seasonality

Seasonality is a characteristic of a time series in which the data experiences regular and predictable changes that recur every calendar year. Any predictable fluctuation or pattern that recurs or repeats over a one-year period is said to be seasonal.

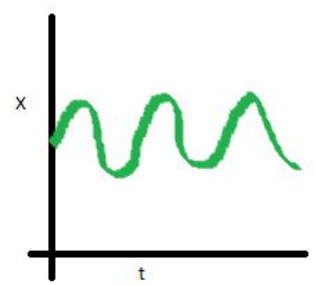


Trend + Seasonality

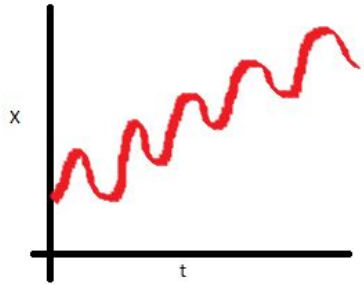


Stationary Series

Constant Mean

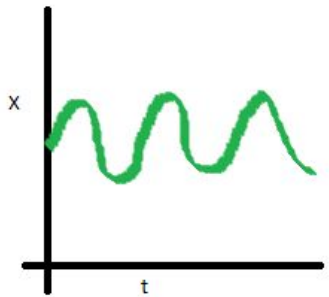


Stationary series



Non-Stationary series

Constant Variance

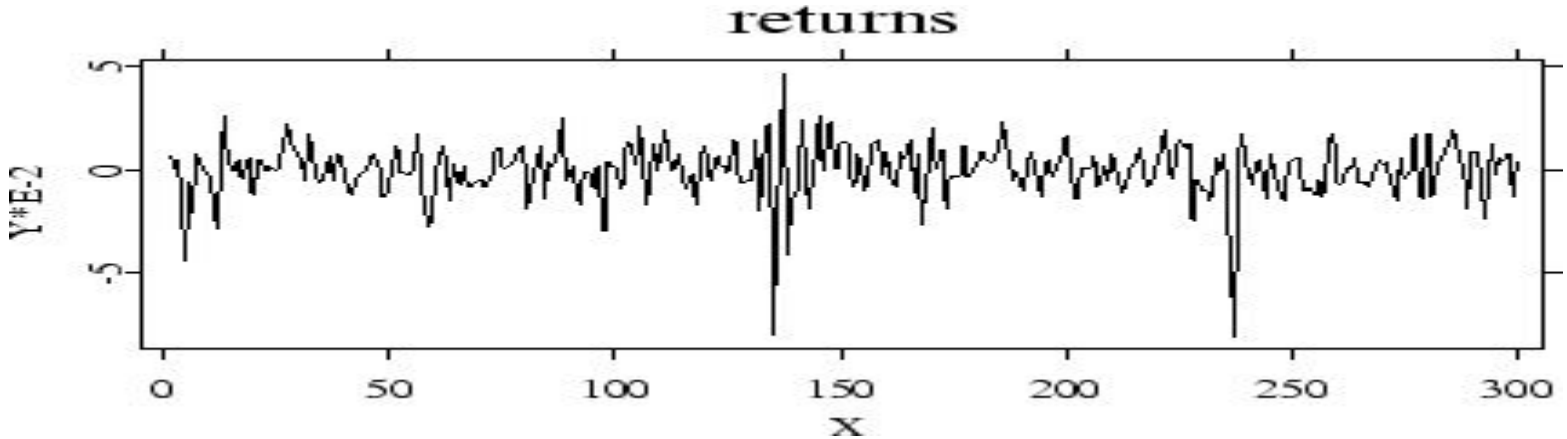


Stationary series

Non-Stationary series

Volatility

Volatility is a statistical measure of the dispersion of data around its mean over a certain period of time. It's calculated as the standard deviation multiplied by the square root of the number of periods of time, T. In finance, it represents this dispersion of market prices, on an annualized basis.



Random Walk

A random walk is one in which future steps or directions cannot be predicted on the basis of past history. When the term is applied to the stock market, it means that short-run changes in stock prices are unpredictable.

The process used to generate the series forces dependence from one-time step to the next. This dependence provides some consistency from step-to-step rather than the large jumps that a series of independent, random numbers provides.

It is this dependency that gives the process its name as a “random walk” or a “drunkard’s walk”.

Random Walk

A simple model of a random walk is as follows:

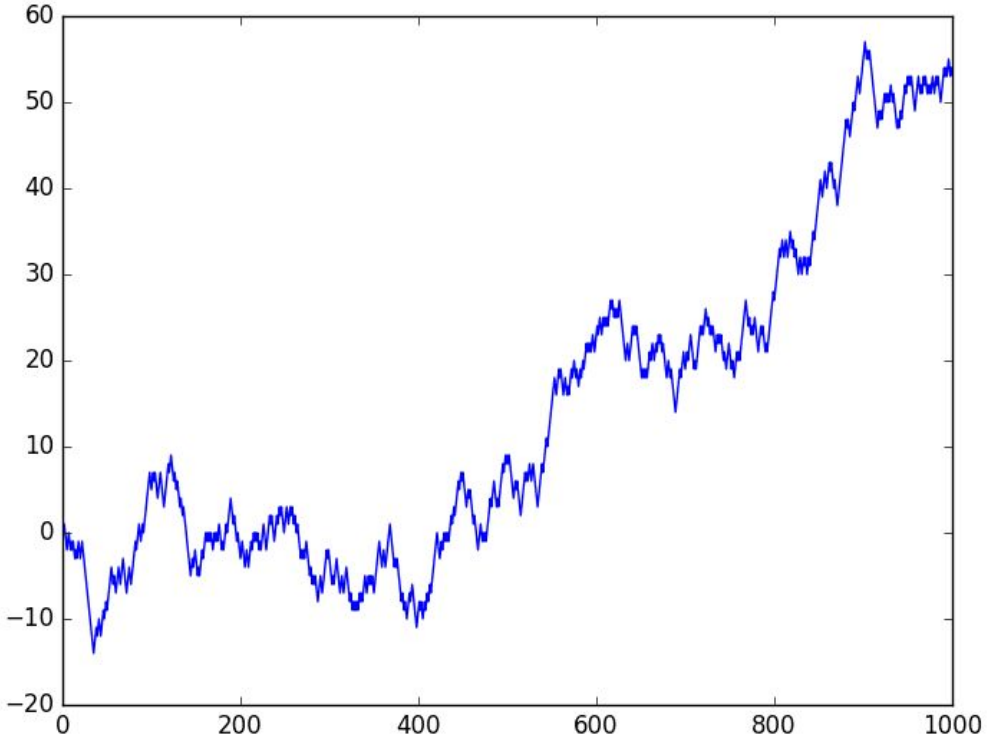
1. Start with a random number of either -1 or 1.
2. Randomly select a -1 or 1 and add it to the observation from the previous time step.
3. Repeat step 2 for as long as you like.

More succinctly, we can describe this process as:

$$X(t) = X(t\text{-}1) + Er(t)$$

A demo

```
1  from random import seed
2  from random import random
3  from matplotlib import pyplot
4  seed(1)
5  random_walk = list()
6  random_walk.append(-1 if random() < 0.5 else 1)
7  for i in range(1, 1000):
8   movement = -1 if random() < 0.5 else 1
9   value = random_walk[i-1] + movement
10  random_walk.append(value)
11 pyplot.plot(random_walk)
12 pyplot.show()
```



Example of Real World Data

Importing the Data

```
In [2]: raw_csv_data = pd.read_csv("Index2018.csv")
```

```
In [3]: df_comp = raw_csv_data.copy()
```

Examining the Data

```
In [4]: df_comp.head()
```

Out[4]:

	date	spx	dax	ftse	nikkei
0	07/01/1994	469.90	2224.95	3445.98	18124.01
1	10/01/1994	475.27	2225.00	3440.58	18443.44
2	11/01/1994	474.13	2228.10	3413.77	18485.25
3	12/01/1994	474.17	2182.06	3372.02	18793.88
4	13/01/1994	472.47	2142.37	3360.01	18577.26

(Normalizing)

SPX500, DAX 40, FTSE100, NIKKEI 225

The DAX is a stock market index consisting of the 40 major German blue chip companies trading on the Frankfurt Stock Exchange. It is a total return index. Prices are taken from the Xetra trading venue.

Foundation: 1 July 1988

The Financial Times Stock Exchange 100 Index, also called the **FTSE** 100 Index, FTSE 100, FTSE, or, informally, the "Footsie", is a share index of the 100 companies listed on the London Stock Exchange with the highest market capitalisation.

Foundation: 1984

Nikkei 225 and TEDPIX (372)

The Nikkei 225, or the Nikkei Stock Average, more commonly called the Nikkei or the Nikkei index, is a stock market index for the Tokyo Stock Exchange. It has been calculated daily by the Nihon Keizai Shimbun newspaper since 1950.

Foundation: 7 September 1950; 72 years ago

Weighting method: Price-weighted index

The TEDPIX is a major stock market index which tracks the performance of the major companies listed on the Tehran Stock Exchange.

S&P 500

The Standard and Poor's 500, or simply the **S&P 500**, is a stock market index tracking the stock performance of 500 of the largest companies listed on stock exchanges in the United States. It is one of the most commonly followed equity indices.

(503 companies now)

Foundation: March 4, 1957; 66 years ago

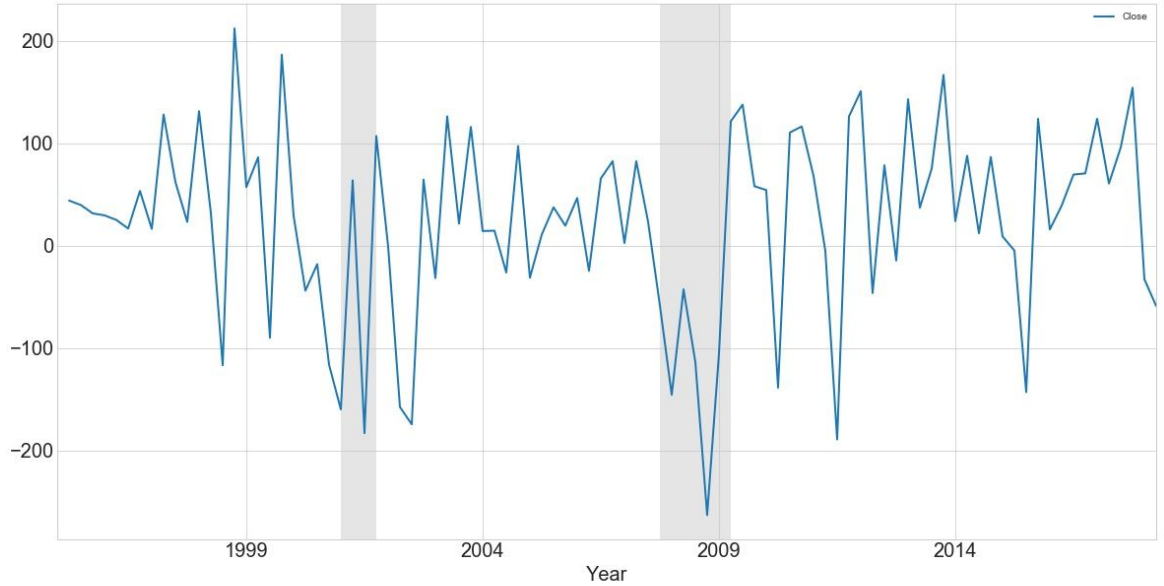
It’s a benchmark.

https://www.slickcharts.com/sp500

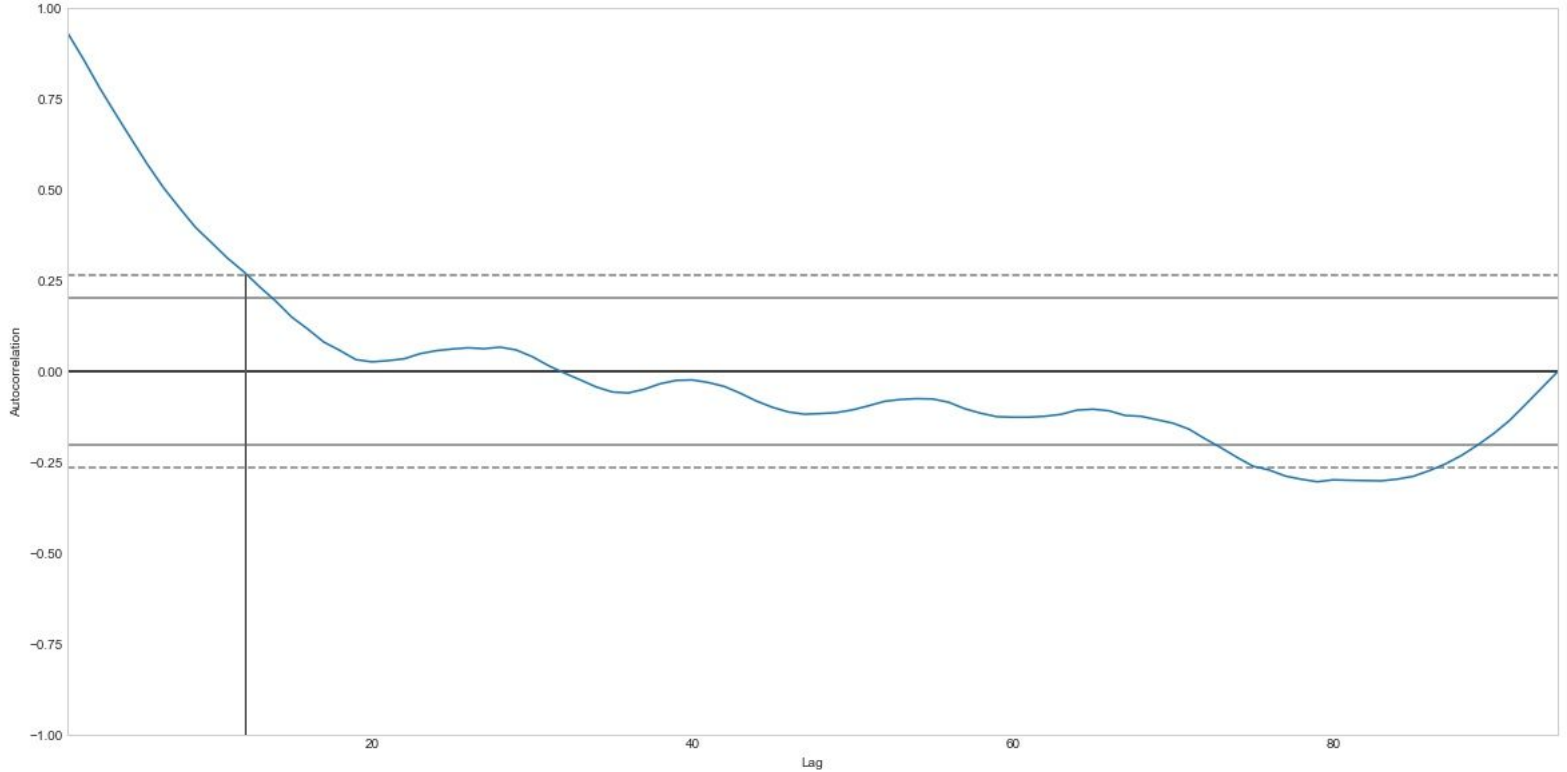
S&P 500 Daily Stock Prices

S&P 500 Daily Stock Prices - Trend

S&P 500 Daily Stock Prices - Stationarity



S&P 500 Daily Stock Prices - Auto-Correlation



https://www.slickcharts.com/sp500
https://tradingeconomics.com/iran/stock-market
https://finance.yahoo.com/quote/%5EN225/

Modeling

1- AR(n) : AutoRegressive

2- MA : Moving averages

3- ARMA : Autoregressive moving averages

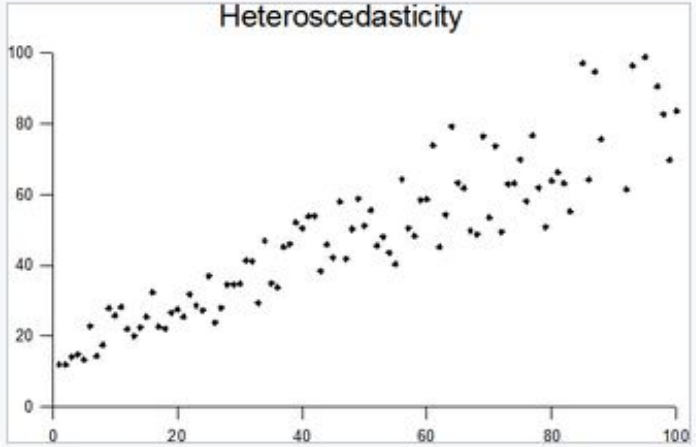
4 - ARIMA : Autoregressive integrated moving averages

5 - ARIMAX / SARIMAX : (Seasonal) Autoregressive Integrated Moving Averages with Exogenous variables

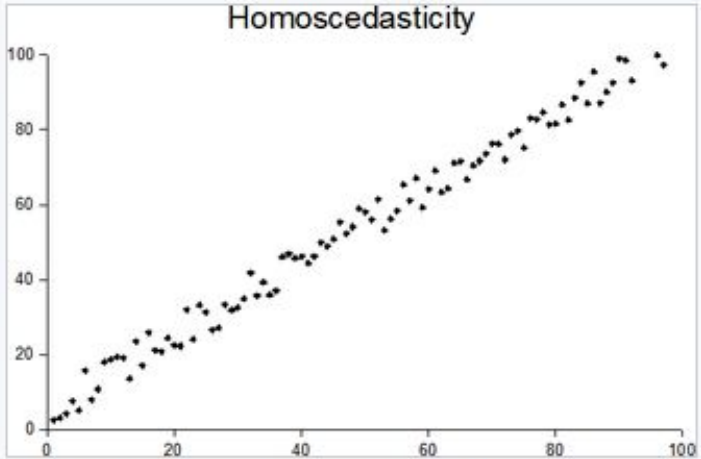
6 - ARCH: autoregressive conditionally heteroscedastic

7 - GARCH: Generalized AutoRegressive Conditional Heteroskedasticity

Homoscedasticity and heteroscedasticity

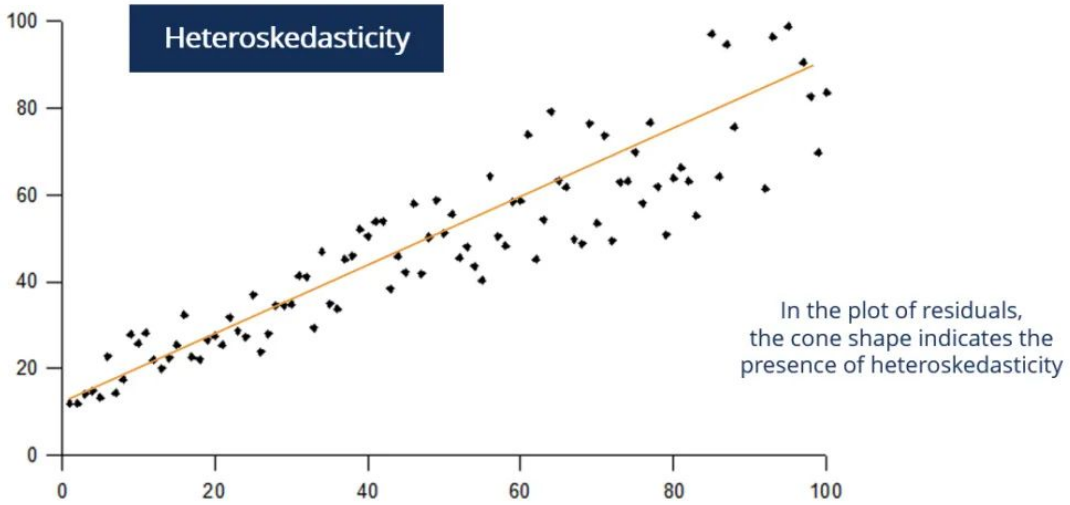


Plot with random data showing heteroscedasticity: The variance of the *y*-values of the dots increase with increasing values of *x*.



Plot with random data showing homoscedasticity: at each value of *x*, the *y*-value of the dots has about the same variance.

In statistics, heteroskedasticity (or heteroscedasticity) happens when the standard deviations of a predicted variable, monitored over different values of an independent variable or as related to prior time periods, are non-constant.



AR(n):

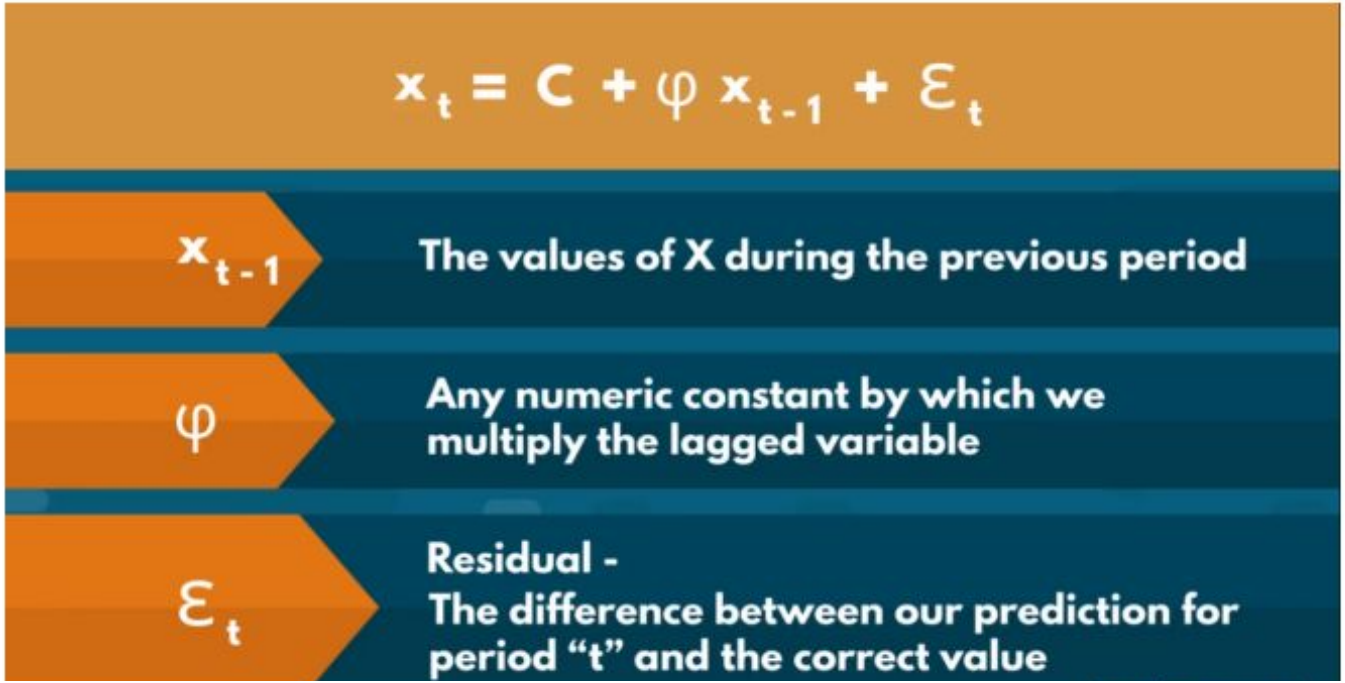
The name of the model comes from autoregression. This means that the model uses values of the same variable (auto) to estimate the current one (regression).

We rely on autoregressive models when there is clear autocorrelation within the data. The term

(autocorrelation) suggests that the variable is correlated with itself. More precisely, values from consecutive periods are related.

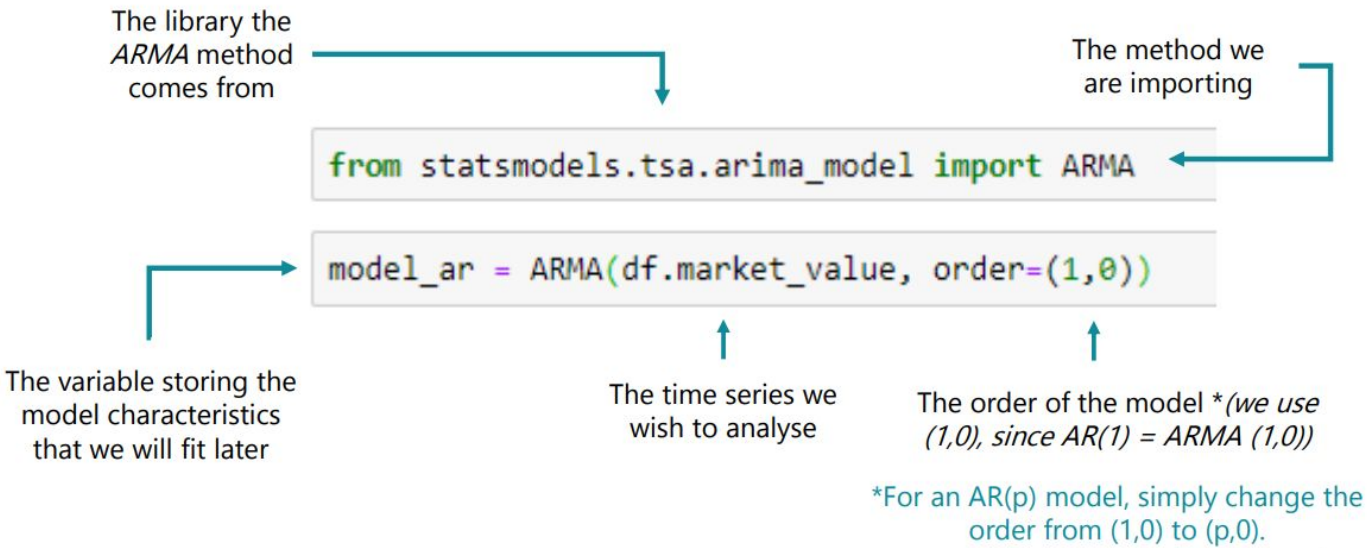
Since time series assumes that patterns found in the past translate to the future, if autocorrelation is present in the data, we need to use some form of AR model to capture this relationship if we wish to make good estimates.

Mathematical Notation:



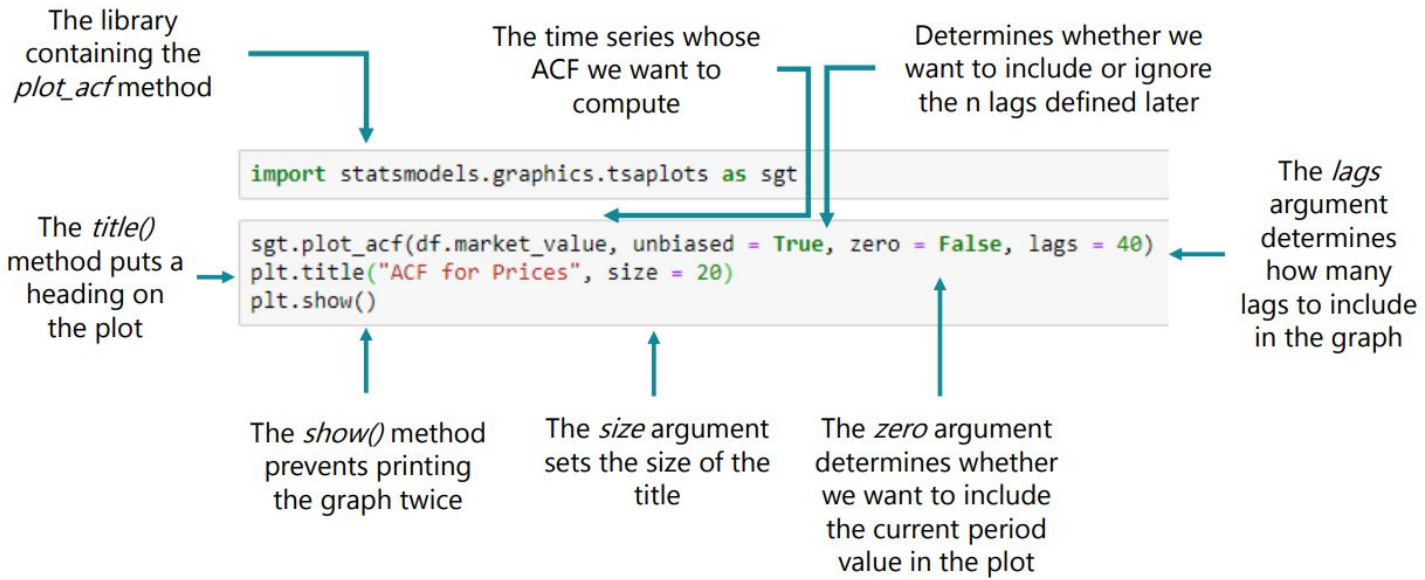
The AR Model

Implementation of the Simple Model in Python:



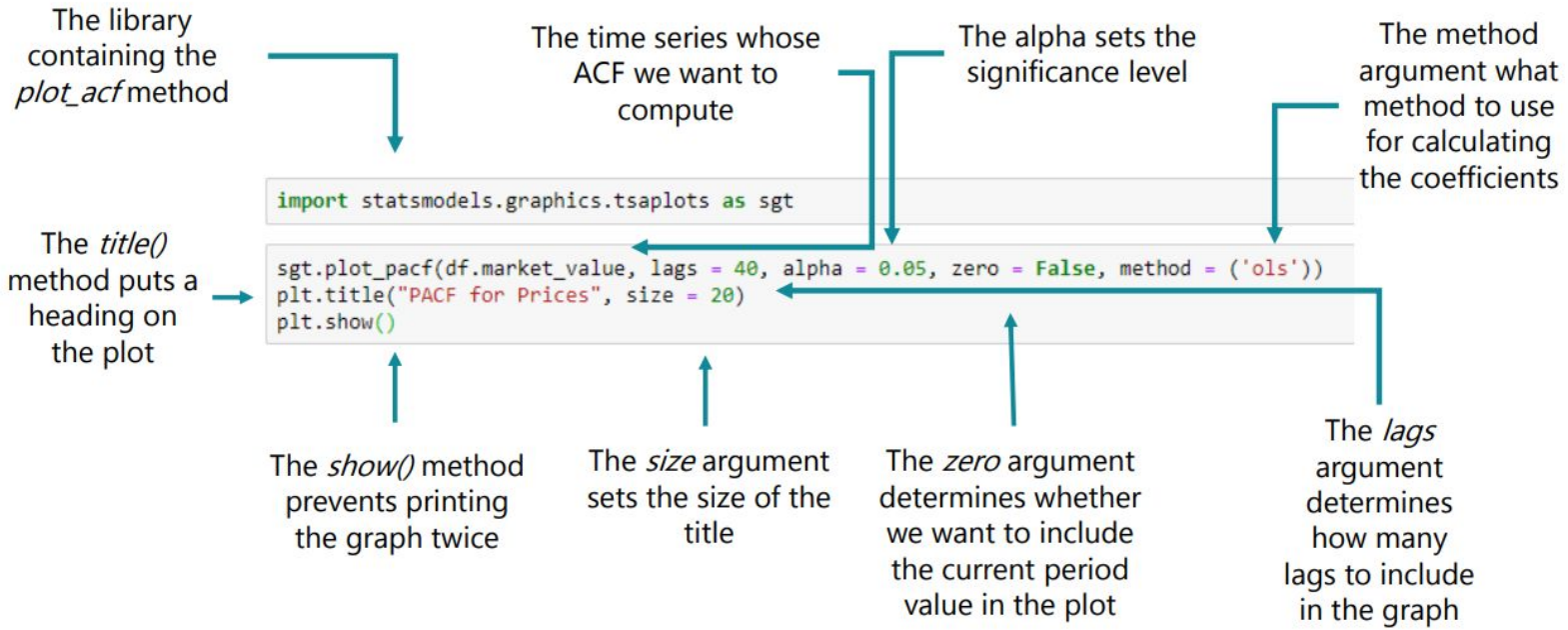
The ACF Method

Calling the ACF Method



The PACF Method

Calling the PACF Method



Moving Average (MA)

The name of the model comes from the moving averages of fixed period intervals as we cruise move through the data set.

This implies that there are other factors apart from the previous values of the variable that need to be accounted for. Therefore, if we know how far off our predictions were last time, then we have a better chance of estimating the prices better this time.

This is why, these models incorporate past residuals (also known as error terms) to help us improve our estimations. These make sure our model handles unexpected shocks well, which is

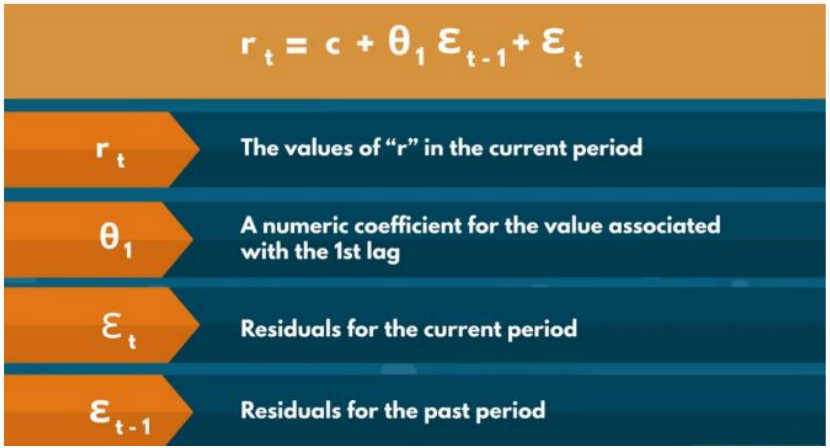
why it’s also known as a smoothing model.

The MA Model

Full Name:

The **M**oving **A**verage Model

Mathematical Notation:



Description:

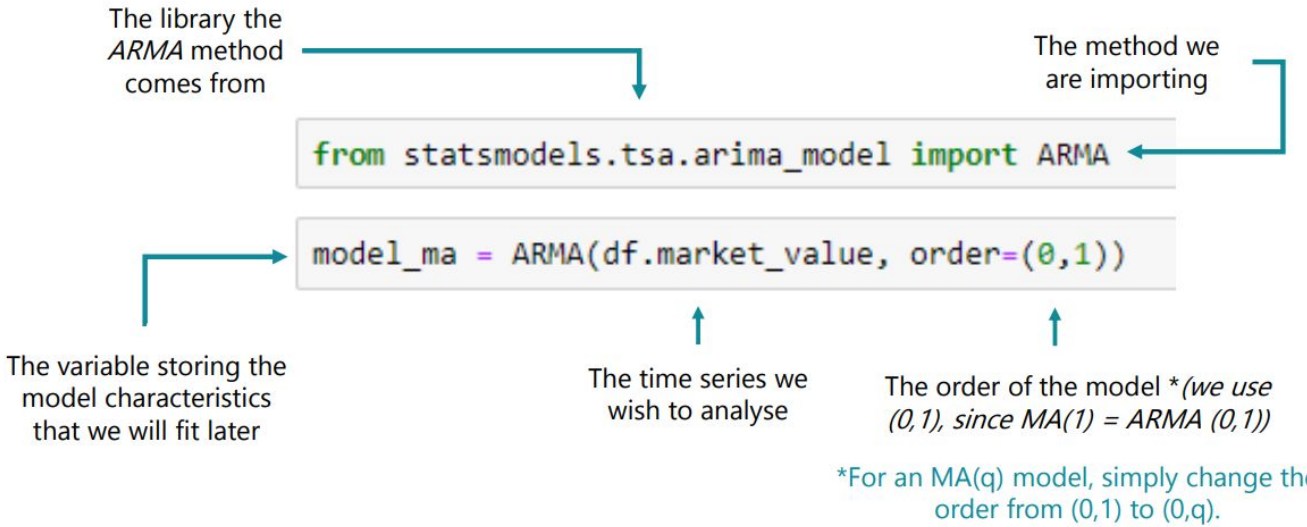
The name of the model comes from the moving averages of fixed period intervals as we cruise move through the data set.

This implies that there are other factors apart from the previous values of the variable that need to be accounted for. Therefore, if we know how far off our predictions were last time, then we have a better chance of estimating the prices better this time.

This is why, these models incorporate past residuals (also known as error terms) to help us improve our estimations. These make sure our model handles unexpected shocks well, which is why it's also known as a smoothing model.

The MA Model

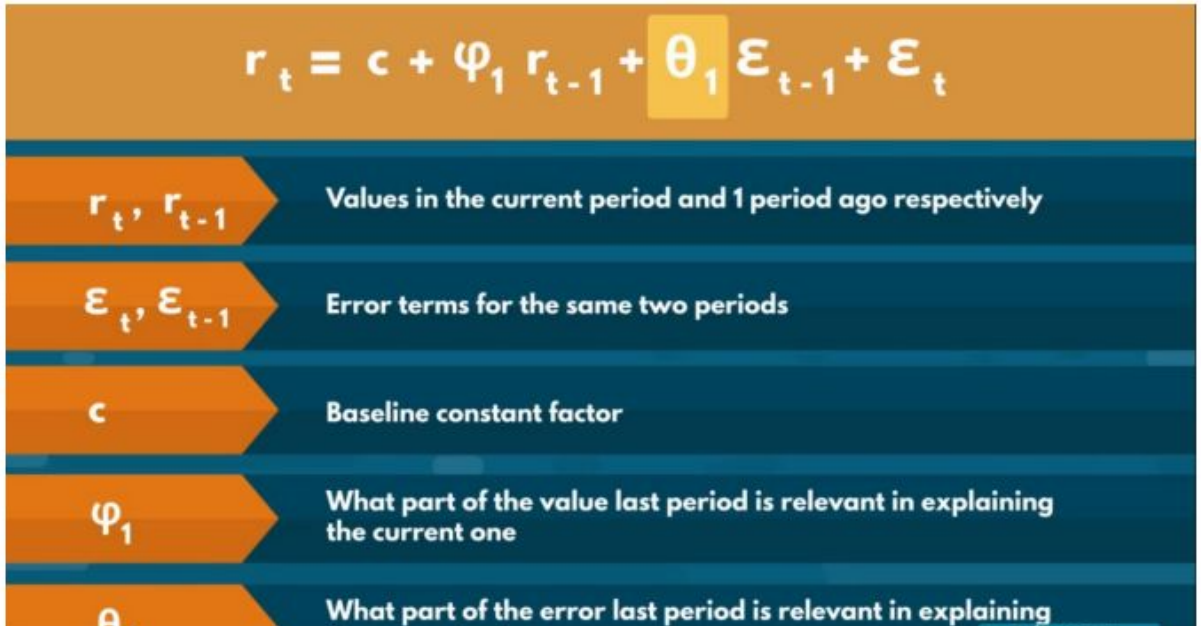
Implementation of the Simple Model in Python:



Full Name:

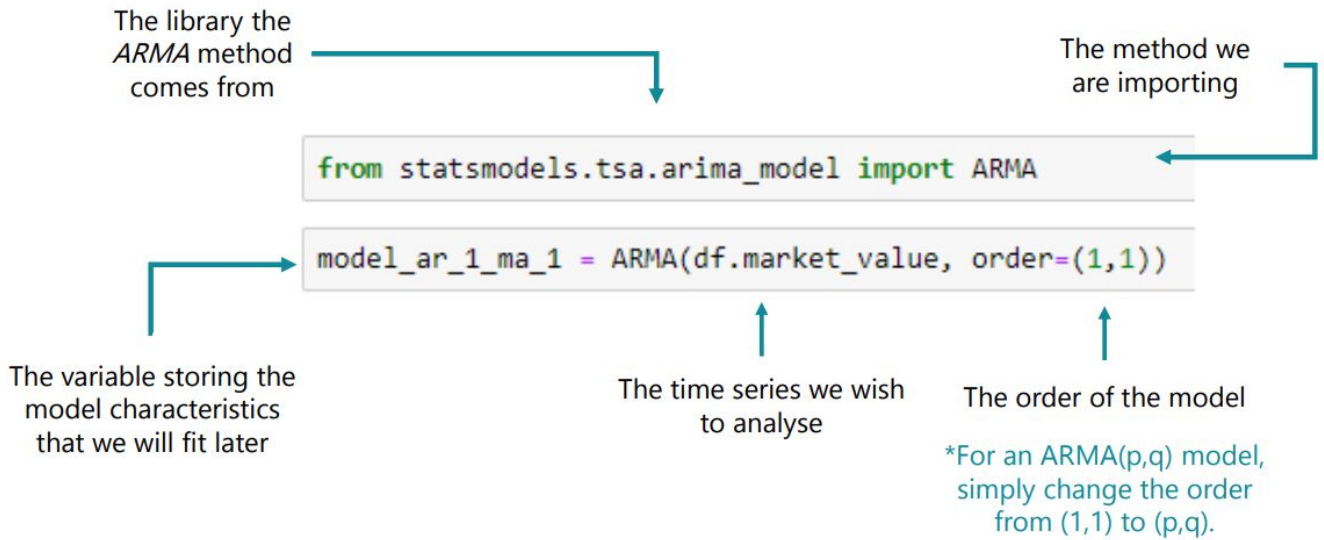
The **A**uto**r**egressive **M**oving **A**verage Model

Mathematical Notation:



The ARMA Model

Implementation of the Simple Model in Python:

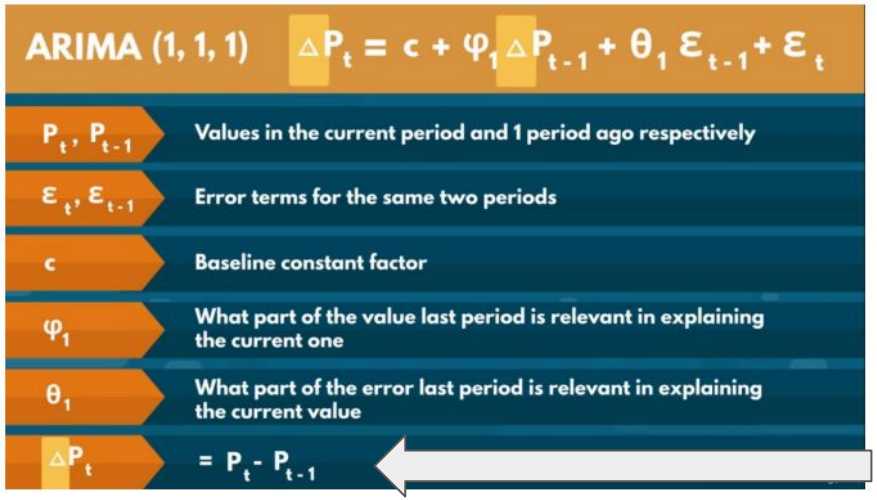


The ARIMA Model

Full Name:

The **A**uto**r**egressive **I**ntegrated **M**oving **A**verage Model

Mathematical Notation:



Description:

The ARIMA is just an integrated version of the ARMA model. What that means is, we simply integrate the data (however many times is needed) to get a stationary set.

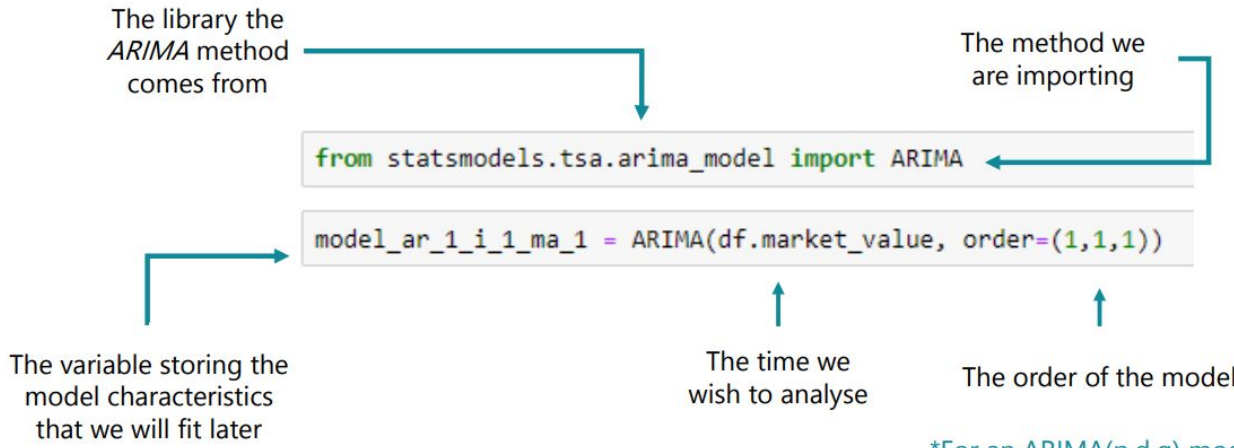
Then, we fit a normal ARMA model like we already learned to.

An ARIMA model with 0 degrees of integration is simply an ARMA model, and so any ARIMA (p, 0, q) model is equivalent to an ARMA (p,q).

The order of Integration (d) tells us exactly how many times we need to compute the non-seasonal differences between the values to reach stationarity and including more is discouraged (due to data attrition and

The ARIMA Model

Implementation of the Simple Model in Python:



The ARMAX Model

Full Name:

The **A**uto**r**egressive **M**oving **A**verage e**X**ogenous Model

Mathematical Notation:

$$P_t = c + \beta X + \varphi_1 P_{t-1} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

P_t, P_{t-1}	Values in the current period and 1 period ago respectively
ε_t, ε_{t-1}	Error terms for the same two periods
c	Baseline constant factor
φ_1	What part of the value last period is relevant in explaining the current one
θ_1	What part of the error last period is relevant in explaining the current value
X	Exogenous variable
β	Coefficient for the exogenous variable

Description:

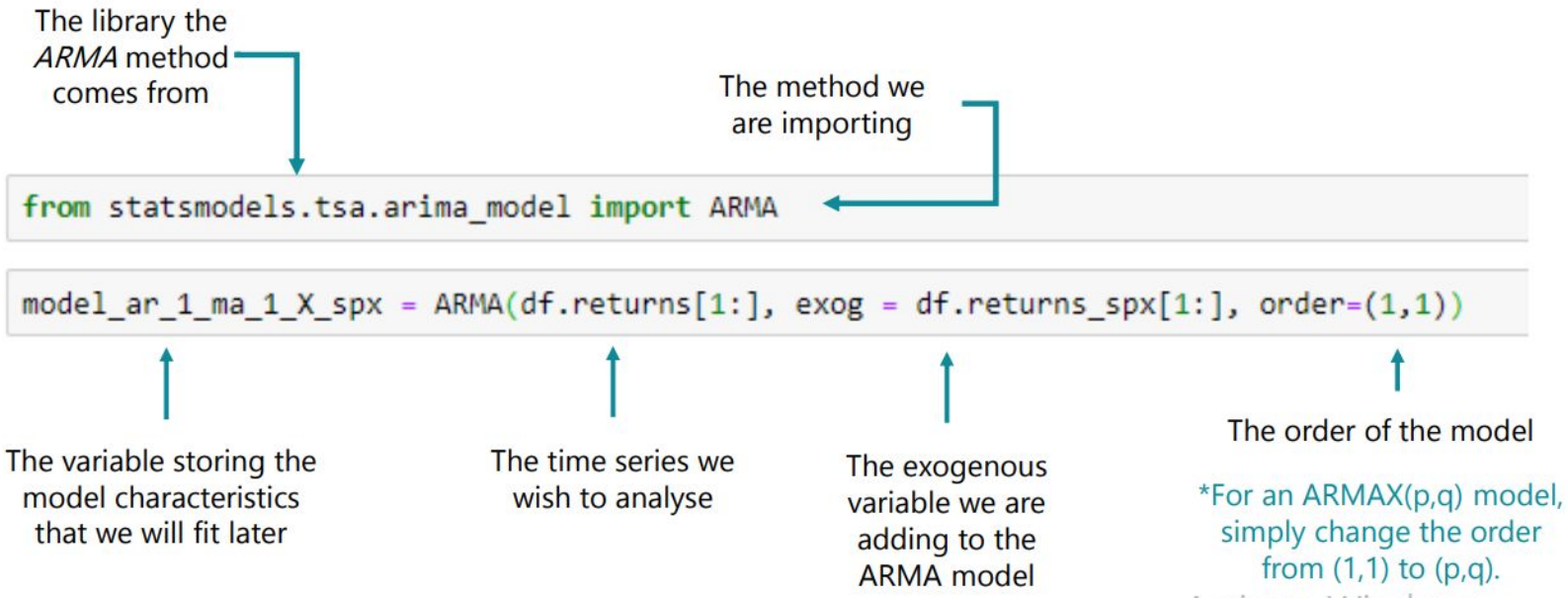
The ARMAX is an extension of the ARIMA model, which incorporates other **exogenous** variables.

These variables can be pretty much anything that can have an affect on the values we are trying to estimate. The only requirement is that we have data available for every time period we are interested in. Thus, we often rely on other time series as the exogenous components in the regression.

These models are great, when a big part of the change period to period cannot be explained by past values and past errors alone, so including other relevant values might be of great help (like the prices for an index of a market of a neighbouring country).

The ARMAX Model

Implementation of the Simple Model in Python:

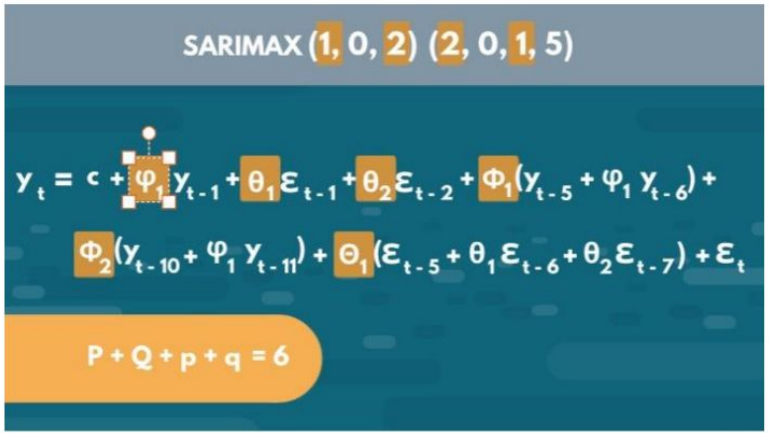


The SARIMAX Model

Full Name:

The **S**easonal **A**uto**r**egressive **I**ntegrated **M**oving **A**verage e**X**ogenous Model

Mathematical Notation:



Short Description:

The SARIMAX is the **seasonal** equivalent of the ARIMAX model. Of course, there exist seasonal versions of the other models as well (SARMA, SARIMA, SARMAX, etc.).

Seasonal models help capture patterns which aren't ever-present but appear periodically. For example, the amount of flights leaving an international hub like JFK Airport in NYC are far larger in December compared to October.

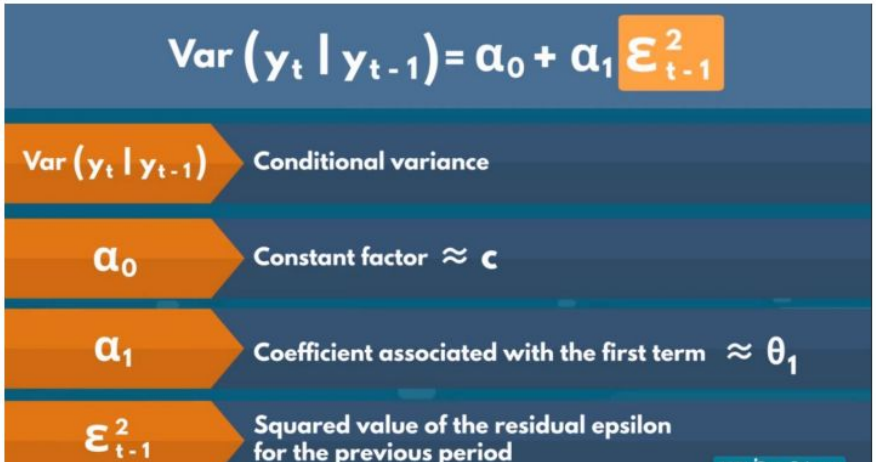
That is mainly due to the festive period for many countries in December. Thus, October is far less busy. Therefore, we need a way to account for this expected influx of demand in December and we can do so by checking the values in December of the previous year.

The ARCH Model

Full Name:

The **A**uto**r**egressive **C**onditional **H**eteroskedasticity Model

Mathematical Notation:



Description:

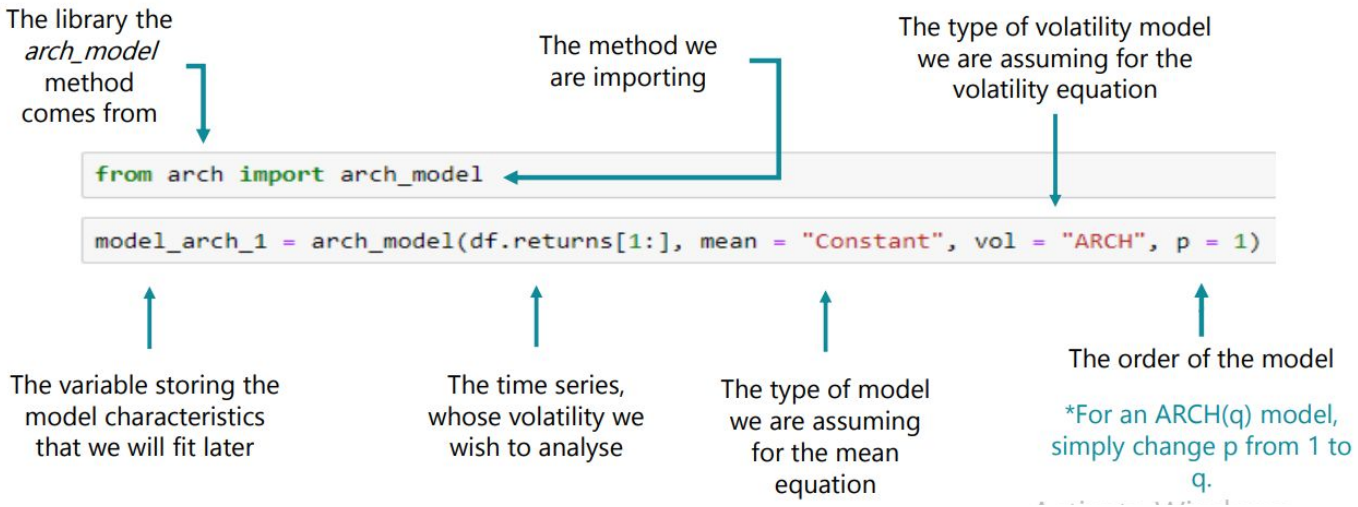
Unlike the previous models, the ARCH measures volatility of the results, rather than the results themselves. Thus, the purpose of it is entirely different and focused on predicting turbulence in the data, regardless of whether it's an increase or decrease in the values.

As you can see on the left side of the equation, the endogenous variable is the variance, rather than the time series variable.

Thus, this is only the variance equation of the model. The simplest ARCH model assumes a 0 or constant mean, so this is the only equation we are interested in.

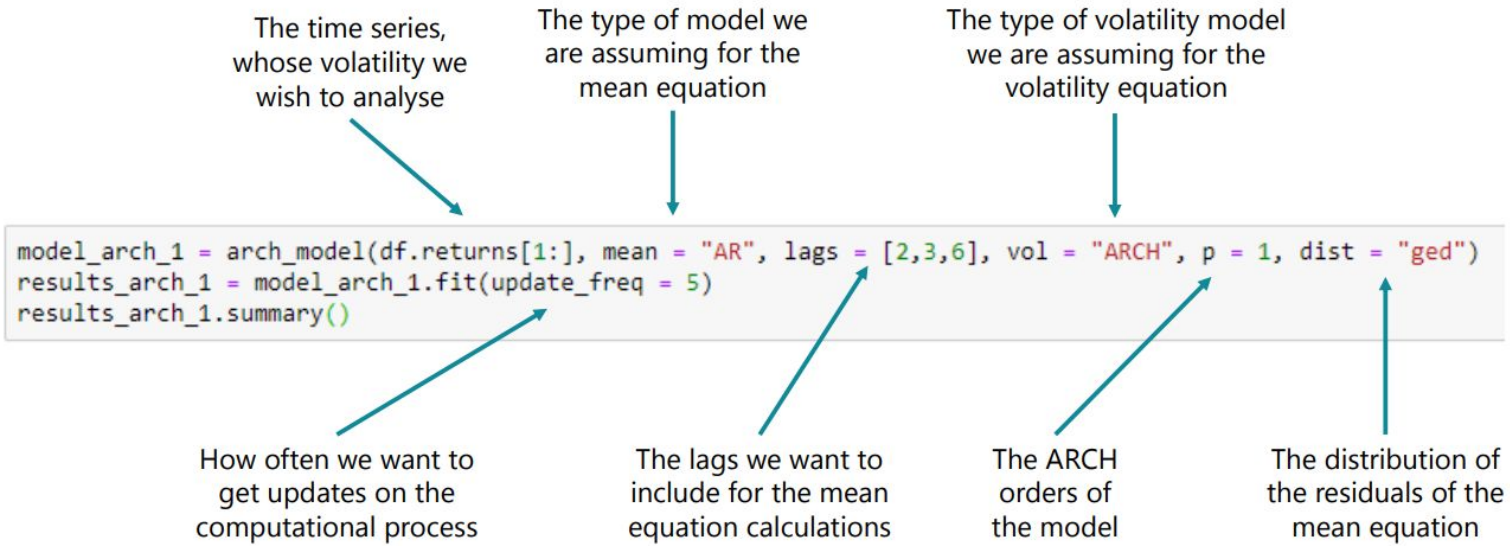
The ARCH Model

Implementation of the Simple Model in Python:



The arch_model Method

Calling the arch_model Method

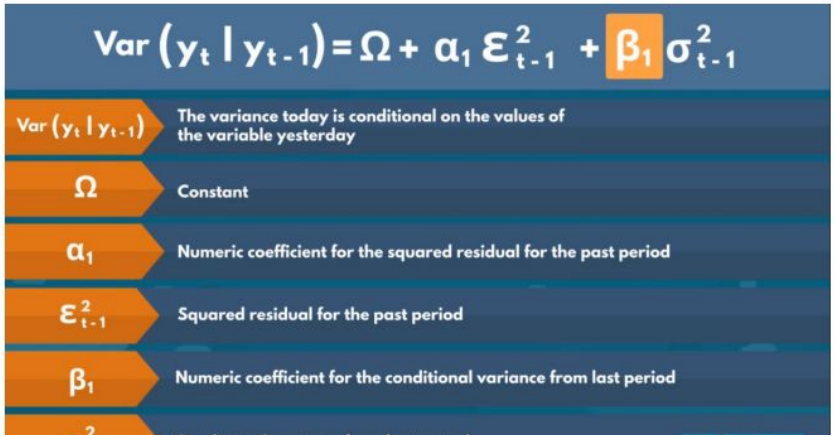


The GARCH Model

Full Name:

The **G**eneralized **A**uto**r**egressive **C**onditional **H**eteroskedasticity Model

Mathematical Notation:



Description:

As the name suggests, the GARCH is just the generalized version of the ARCH model.

This generalization is expressed in including past variances as well as past squared residuals to estimate current (and subsequent) variances.

The generalization comes from the fact that including a single past variance would (in theory) contain in itself the explanatory power of all other previous squared error terms.

It serves as a sort of ARMA equivalent to the ARCH, where we're including both past values and past errors (albeit squared).

The GARCH Model

Implementation of the Simple Model in Python: