Data Science Module 4

Day 3 - Time Series Analysis

Today we are going to learn to...

1. Identify and work with time series data
2. Understand trend and seasonality and how to identify them
3. Compute rolling statistics
4. Understanding what makes a time series stationary
5. Use differences to remove trends in time series data (de-trending)

...and then we are done!

Before we start...

1. Make sure you are comfortable
2. Have water and maybe a strong coffee handy
3. If you need a break...take it!
4. If you need a stretch - please go ahead!
5. Please mute yourselves if you are not talking
6. Have your video on at all times

...and let’s get started!

Introduction to Time Series Analysis

Group discussion...

- What makes a dataset a time series?
- Time series at work - any examples?

What is a time series

- Series of data points **indexed (or listed) in time order**

- Data must be measured over time at **consistent interval**

- Time series are often represented as a set of observations that have a **time-bound relation,** which is represented as an index

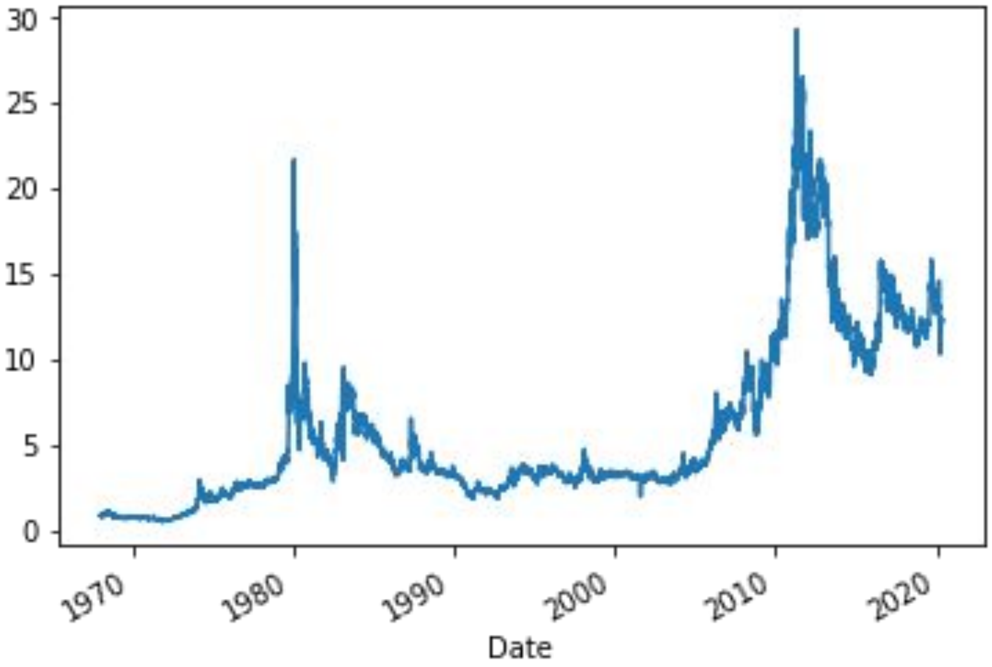
What are the goals of time series analysis?

There are two main goals of time series analysis:

- Identifying trends, cycles, and seasonal variances to aid in the forecasting of a future event

- Forecasting (predicting future values of the time series variable)

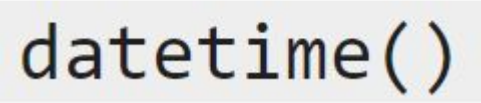
Time Series



Silver Pricing data by the LBMA

Working with Time Series

The DateTime Library



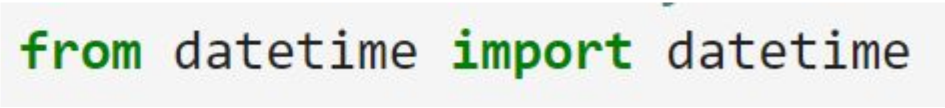
```
datetime()
```

- Dealing with data containing date and time can be tedious job to do
- In python we have a **datetime** library enables us to easily manipulate date and time by creating a **datetime object**

Key object classes

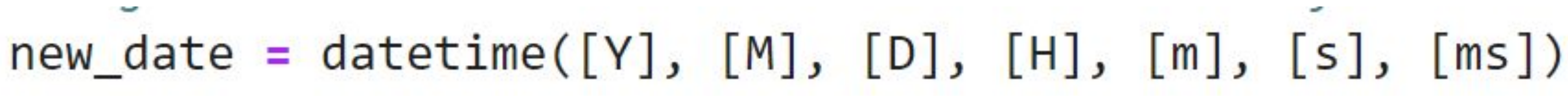
- **datetime.date :** It allows us to manipulate date without interfering time (month, day, year)
- **datetime.time :** It allows us to manipulate date without interfering date (hour, minute, second, microsecond)
- **datetime.datetime :** It allows us to manipulate the combination of date and time (month, day, year, hour, second, microsecond).
- **datetime.tzinfo :** An abstract class for dealing with time zones. These types of objects are immutable. For instance, to account for different time zones and/or daylight saving times.
- **datetime.timedelta :** It is the difference between two date, time or datetime instances

The DateTime Library - Creating a datetime object



```
from datetime import datetime
```

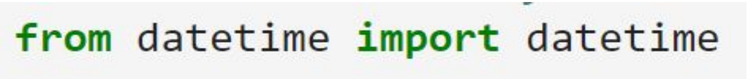
- To create a date object, call the datetime class and supply the necessary arguments



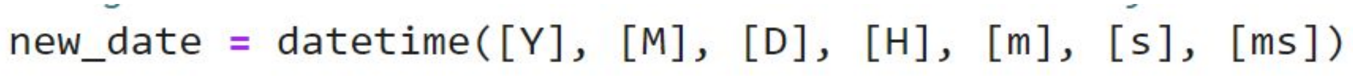
```
new_date = datetime([Y], [M], [D], [H], [m], [s], [ms])
```

The DateTime Library - Basic Operations

- To **create a date object**, call the datetime class and supply the necessary arguments

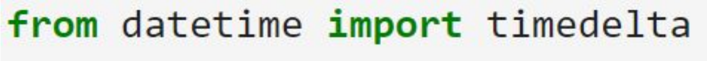


```
from datetime import datetime
```

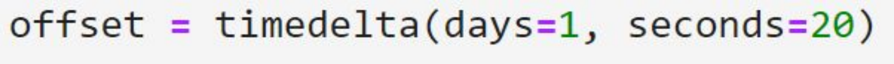


```
new_date = datetime([Y], [M], [D], [H], [m], [s], [ms])
```

- To **calculate time differences** (**create a timedelta object)**, call the timedelta class and supply the necessary arguments



```
from datetime import timedelta
```



```
offset = timedelta(days=1, seconds=20)
```

01_time_series- Apprentice

Preprocessing Time Series data with Pandas

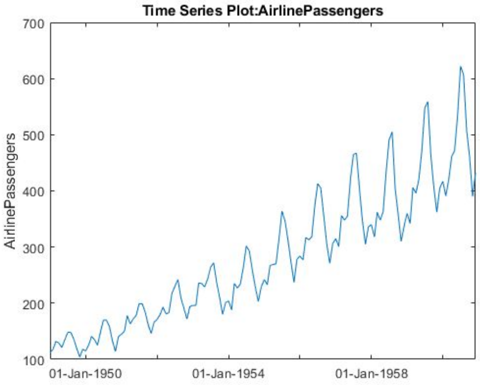
Key operations:

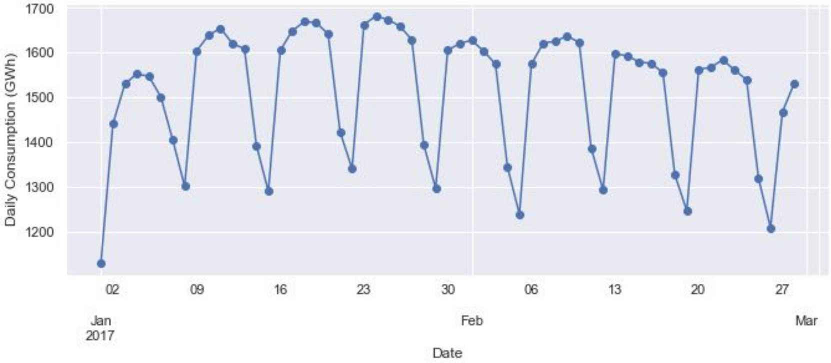
- *Converting date to datetime object using* ***pd.to_datetime(colum)***
- *Extracting different date parts from date object* ***using .dt attribute***
- *Creating Timestamps for filtering on Date* **pd.to_datetime()**
- *Setting a datetime column to index using* ***df.set_index('Date', inplace = True)***

01_time_series- Apprentice

Time Series Rolling Stats

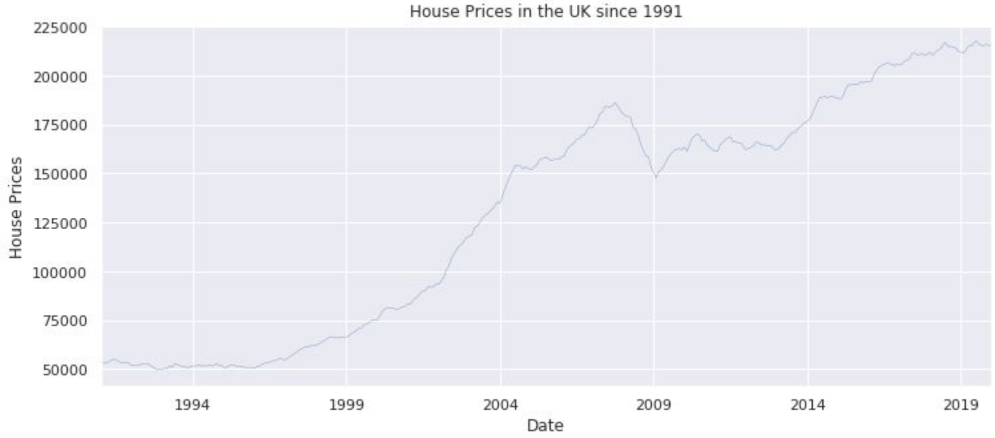
Key components of a time series





Key components of a time series

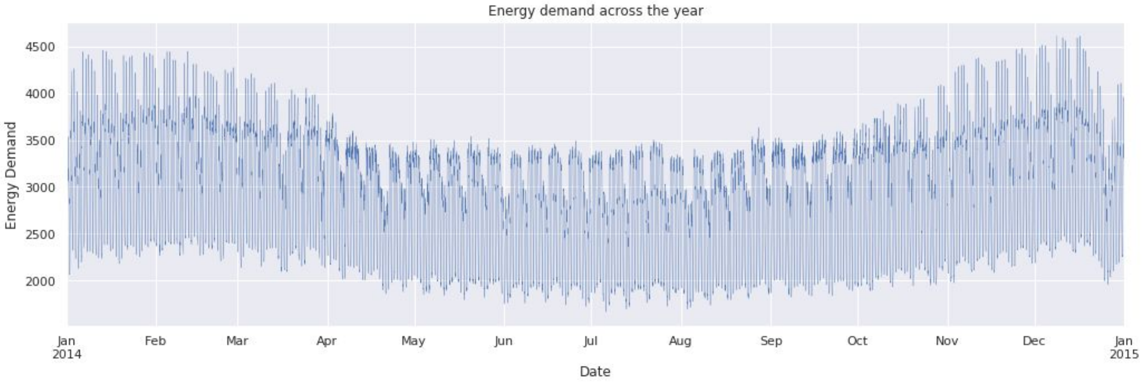
Trend: General direction in which something is developing or changing. A trend can be upward(uptrend) or downward(downtrend).



Key components of a time series

Seasonality: *Predictable pattern that recurs or repeats over regular intervals. Seasonality is often observed within a year or less.*

Can we think of example of data with seasonality?



The easiest way to visualize patterns is drawing them

02_rolling_statistics_Apprentice

Key rolling stats concepts

- ***Resample:*** used to summarize data by date or time.

- ***Rolling:*** *used to calculate stats of the last n values.*

- ***Expanding:*** used to calculate stats using all of the data points up to the current time as opposed to a moving (rolling) window

- ***Exponentially Weighted Windows:*** *used to calculate stats decreasing the contribution of time points that are further in the past*

Time Series Differencing and Stationarity

What makes a time series stationary?

Why does it matter?

What makes a time series stationary?

A time series is stationary when its statistical properties: the mean, variance, and autocorrelation (covered tomorrow) do not change over time.

Why does it matter?

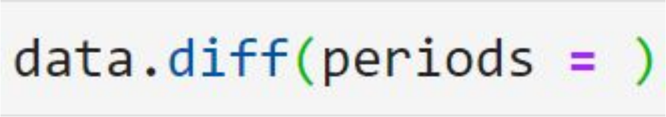
Most statistical forecasting methods assume the time series you are forecasting on to be stationary.

A stationarized series is relatively easy to predict: you simply predict that its statistical properties will be the same in the future as they have been in the past!

Time Series Differencing and Stationarity

- **Differencing is the most common way to make a series stationary**
- **This leads to removal of trends, resulting in a mean of zero across time**
- **Sometimes multiple differencing is carried out to achieve stationarity**

$$\Delta y_t = y_t - y_{t-1}$$

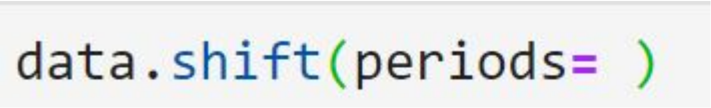


```
data.diff(periods = )
```

Where periods is the shift for calculating difference

Shifting and lagging time series data

- **Shifting or lagging time series data backward and forwards allows us to calculate percentage change from sample to sample or period to period**



```
data.shift(periods= )
```

02_rolling_statistics_Apprentice

Where periods is the shift for calculating difference