Intro to pandas

Daniel García Hernández

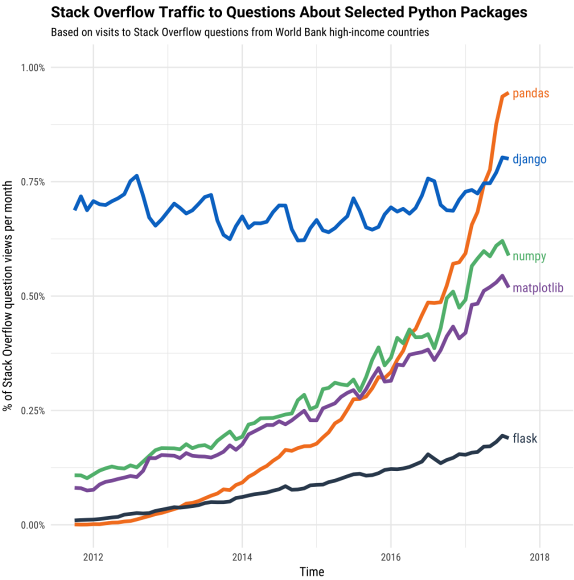
Master Big Data

February 20, 2020

: Overview

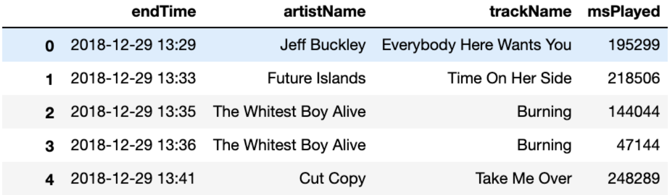
1. Why to learn pandas
2. Introduction to pandas
3. Series and DataFrames
4. Slicing, filtering, mapping
5. Real life uses of Pandas

Why to learn pandas: Evolution of pandas



Introduction to pandas: Description and elements

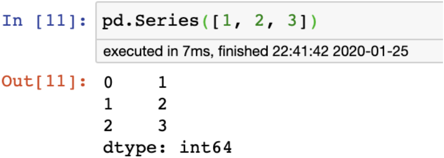
- Pandas is a Python library containing tools for data analysis
- NumPy under the hood
- Its main component is the series: 1D data
- Aggregated series conform a dataframe: 2D data



	endTime	artistName	trackName	msPlayed
0	2018-12-29 13:29	Jeff Buckley	Everybody Here Wants You	195299
1	2018-12-29 13:33	Future Islands	Time On Her Side	218506
2	2018-12-29 13:35	The Whitest Boy Alive	Burning	144044
3	2018-12-29 13:36	The Whitest Boy Alive	Burning	47144
4	2018-12-29 13:41	Cut Copy	Take Me Over	248289

Series and DataFrames: Elements in a Series object

- `pandas.Series`
- Series contain 1D in an array-like data structure
- Data contained in Series is assigned a label (index)
- Can be created from lists, NumPy arrays, dictionaries
- Can contain integers, floats, strings, booleans, dates,...



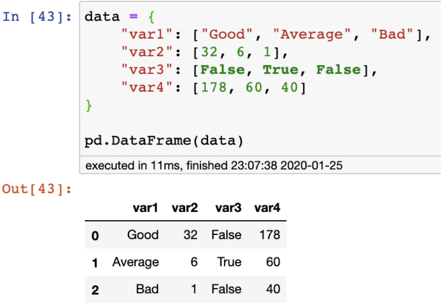
```
In [11]: pd.Series([1, 2, 3])
```

executed in 7ms, finished 22:41:42 2020-01-25

```
Out[11]: 0    1
         1    2
         2    3
         dtype: int64
```

Series and DataFrames: Elements in a DataFrame object (1)

- `pandas.DataFrame`
- DataFrames (df) are containers of Series, and with them we can store, treat and process tabular data
- Data contained in a df can be accessed by its coordinates (row, column)
- The index of a df is similar to a Series index



```
In [43]: data = {
              "var1": ["Good", "Average", "Bad"],
              "var2": [32, 6, 1],
              "var3": [False, True, False],
              "var4": [178, 60, 40]
         }

         pd.DataFrame(data)
```

executed in 11ms, finished 23:07:38 2020-01-25

Out[43]:

	var1	var2	var3	var4
0	Good	32	False	178
1	Average	6	True	60
2	Bad	1	False	40

Series and DataFrames: Elements in a DataFrame object (2)

- Even though rows and columns are the names for the coordinates within a dataframe, there are other denominations
  - Rows, observations, `axis=0`
  - Columns, variables, features, `axis=1`
- Columns accesible by using the `columns` property of a df
- Index accesible by using the `index`

Slicing, filtering, mapping: Slicing

- Slice a Series using `series.loc[start:end]`
- Slice a Dataframe
  - Using `df.loc[index_value, column_name]`
  - Using `df.iloc[ri:rf, ci:cf]`

Slicing, filtering, mapping: Filter

- Filter a Series using `series[condition]`
- Filter a Dataframe using `df[condition]`

`condition` must be so it returns a mask of boolean values

Slicing, filtering, mapping: Map

`map()` allows us to pass a function to every element of a series

- `series.map(function)`
- We can define the function using `def` or we can embrace the power of `lambda` functions

For dataframes, we can still use`map()` for a single column: `df[column].map(function)`

Or use `df.apply(function, axis)` in order to pass a function to every element in the specified axis (0 for rows, 1 for columns)

Real life uses of Pandas: BiciMAD dataset

Let's practice with `pandas` and the `bicimad.csv` dataset.

This dataset was obtained from Madrid's open data website:
https://datos.madrid.es/portal/site/egob/

The End