Fuentes de datos

January 31, 2018

1 Fuentes de datos

Update del virtualenv:

```
conda install pandas openpyxl xlrd beautifulsoup4 scrapy lxml html5lib h5py
```

1.1 Pandas y DataFrames http://pandas.pydata.org/

pandas:

- Librería Open Source para análisis de datos
- Python
- Equivalente a data.frame y Dplyr de R-lang

pandas is an open source Python library for data analysis. Python has always been great for prepping and munging data, but it's never been great for analysis - you'd usually end up using R or loading it into a database and using SQL (or worse, Excel). pandas makes Python great for analysis.

Estructuras de datos: - Series - DataFrames

1.2 `import pandas as pd`

```
In [1]: import pandas as pd
```

```
In [ ]: # Series
        s = pd.Series([7, 'Heisenberg', 3.14, -1789710578, 'Happy Eating!'])
        s
```

```
In [ ]: # DataFrame
        data = {'year': [2010, 2011, 2012, 2011, 2012, 2010, 2011, 2012],
                'team': ['Bears', 'Bears', 'Bears', 'Packers', 'Packers', 'Lions', 'Lions', 'L
                'wins': [11, 8, 10, 15, 11, 6, 10, 4],
                'losses': [5, 8, 6, 1, 5, 10, 6, 12]}
        football = pd.DataFrame(data, columns=['year', 'team', 'wins', 'losses'])
        football
```

1.3 Pandas IO Tools

- read_csv / CSV file
- read_excel / MS Excel
- read_hdf / HDF5-Pytables
- read_sql / SQL Query
- read_json / JSON file-JSON dump
- read_msgpack (experimental) / Binary serialization
- read_html / HTML Tables
- read_gbq (experimental) / Google BigQuery
- read_stata / STATA
- read_sas / SAS
- read_clipboard
- read_pickle / pickle — Python object serialization
- to_csv
- to_excel
- to_hdf
- to_sql
- to_json
- to_msgpack (experimental)
- to_html
- to_gbq (experimental)
- to_stata
- to_clipboard
- to_pickle

Working with Excel Files in Python - http://www.python-excel.org/ - `openpyxl`: - A Python library to read/write Excel 2010 xlsx/xlsm files - https://openpyxl.readthedocs.io/en/default/ - `xldr` - Library for developers to extract data from Microsoft Excel (tm) spreadsheet files - http://xlrd.readthedocs.io/en/latest/

```
In [ ]: # reading from CSV
        csv_dataframe = pd.read_csv('data/FL_insurance_sample.csv')
```

In [ ]:
```
# reading from CSV
csv_dataframe_500000_rows = pd.read_csv('data/Spreadsheet-500000-rows.csv', encoding =
```

In [ ]:
```
len(csv_dataframe_500000_rows)
```

In [ ]:
```
csv_dataframe_500000_rows.head()
```

In [ ]:
```
# Reading data from Excel
# pd.read_excel('data/FL_insurance_sample.xlsx', 'Sheet1', index_col=None, na_values=[
```

In [ ]:
```
# Reading JSON
pd.read_json('https://api.github.com/repos/pydata/pandas/issues?per_page=5')
```

In [ ]:
```
# # Reading HDF5
# import numpy as np
# import h5py

# pd.read_hdf('foo.h5','df')
```

In [ ]:
```
# Read from a SQL Query


# Set up connection parameters
db_connection = psycopg2.connect(host="",user="", password="", dbname="")

# Connect to the database server
db_cursor = db_connection.cursor()

# Specify a query with SQL
sql_query = "SELECT * from XX"

# Execute SQL query and save it to a DataFrame
sql_dataframe = pandas.read_sql_query(sql_query, db_connection)

db_connection.close()
```

In [ ]:
```
# Read from a SQL Query
import sqlite3

# Set up connection parameters
db_connection =  sqlite3.connect('data/veekun-pokedex.sqlite')

# pokemon = pd.read_sql_query('SELECT * FROM  pokemon WHERE pokemon.height > 4 and pok

pokemon = pd.read_sql_query('SELECT * FROM  pokemon', db_connection)
```

In [ ]:
```
len(pokemon)
```

2 Web Scrapping con PANDAS

In [2]:
```
import pandas as pd  # Cargar modulo


# Asignar el resultado del metodo read_html a densidad_paises
# Parametro header=0 para establecer el header de la tabla
# Consultar: http://pandas.pydata.org/pandas-docs/stable/gotchas.html#html-gotchas
densidad_paises = pd.read_html('https://simple.wikipedia.org/wiki/List_of_countries_by_

# Observar los primeros 10 registros
densidad_paises[0][:10]
```

Out[2]:

	Rank	Country / dependent territory	Population	Date last updated	Area (km2)	Area (mi2)	Density (km2)	Density (mi2)	Notes
0	1	Macau (China)	541200	February 28, 2017	29.2	11.3	18534	48003	[1]
1	2	Monaco	33000	2013	1.95	0.75	16923	43830	[2][3]
2	3	Singapore	5076700	2000	710.2	274.2	7148	18513	[4]
3	4	Hong Kong (China)	7264100	1998	1104	426	6349	16444	[5]
4	5	Gibraltar (UK)	31000	2018	6.8	2.6	4559	11808	[2]
5	6	Vatican City	826	2009	0.44	0.17	1877	4861	[2][6]
6	7	Bahrain	1234596	2010	750	290	1646	4263	[10]
7	8	Malta	417617	January 1, 2011	316	122	1322	3424	Eurostat estimate
8	9	Bermuda (UK)	65000	2009	53	20	1226	3175	[2]
9	10	Sint Maarten (Netherlands)	37429	January 1, 2010	34	13	1101	2852	[2]

In [3]:
```
# Que tipo de objeto tenemos?
type(densidad_paises)
```

Out[3]:
```
list
```

In [4]:
```
# Como lo convertimos en un dataframe?

densidad_paises_dataframe = pd.DataFrame(densidad_paises[0])
```

In [5]:
```
# Que tipo de objeto tenemos?
type(densidad_paises_dataframe)
```

Out[5]:
```
pandas.core.frame.DataFrame
```

```
In [6]: densidad_paises_dataframe.head()
```

Out[6]:

	Rank	Country / dependent territory	Population	Date last updated	Area (km2)	Area (mi2)	Density (km2)	Density (mi2)	Notes
0	1	Macau (China)	541200	February 28, 2017	29.2	11.3	18534	48003	[1]
1	2	Monaco	33000	2013	1.95	0.75	16923	43830	[2][3]
2	3	Singapore	5076700	2000	710.2	274.2	7148	18513	[4]
3	4	Hong Kong (China)	7264100	1998	1104	426	6349	16444	[5]
4	5	Gibraltar (UK)	31000	2018	6.8	2.6	4559	11808	[2]

```
In [ ]: densidad_paises_dataframe.keys()

In [11]: densidad_paises_clean = densidad_paises_dataframe.copy()

In [7]: # http://pandas.pydata.org/pandas-docs/stable/dsintro.html#column-selection-addition-d
        densidad_paises_clean.pop('Unnamed: 1')


        ---------------------------------------------------------------------------

        NameError                                 Traceback (most recent call last)

        <ipython-input-7-337dad73f678> in <module>()
          1 # http://pandas.pydata.org/pandas-docs/stable/dsintro.html#column-selection-additio
    ----> 2 densidad_paises_clean.pop('Unnamed: 1')


        NameError: name 'densidad_paises_clean' is not defined


In [ ]: # Verificar
        densidad_paises_clean

In [12]: #
         densidad_paises_clean.pop('Area (mi2)')

Out[12]: 0      11.3
         1      0.75
         2     274.2
         3       426
         4       2.6
         5      0.17
         6       290
```

7	122
8	20
9	13
10	115
11	56980
12	45
13	30
14	20.5
15	13890
16	790
17	170
18	75
19	2320
20	24
21	144
22	38432
23	8.1
24	3427
25	171
26	4036
27	8.1
28	10
29	16033
...	
212	148709
213	239285
214	489000
215	830000
216	478841
217	132000
218	188500
219	119500
220	424164
221	496000
222	6601668
223	100
224	240535
225	1052100
226	103347
227	679360
228	83000
229	3855100
230	224610
231	395960
232	63250
233	40000
234	2966200
235	318261

```
236         35000
237        103000
238        603909
239          4700
240        840000
241    Area (mi2)
Name: Area (mi2), Length: 242, dtype: object
```

In [ ]:
```
densidad_paises_clean.pop('Density (/mi2)')
```

In [ ]:
```
# for cycle
column_list = []
for element in column_list:
    dataframe.pop(element)
```

In [8]:
```
densidad_paises_clean.head()
```

```
---------------------------------------------------------------------------

NameError                                 Traceback (most recent call last)

<ipython-input-8-e59015c82099> in <module>()
----> 1 densidad_paises_clean.head()


NameError: name 'densidad_paises_clean' is not defined
```

In [ ]:
```
del densidad_paises_clean['Notes']
```

In [9]:
```
densidad_paises_clean.head()
```

```
---------------------------------------------------------------------------

NameError                                 Traceback (most recent call last)

<ipython-input-9-e59015c82099> in <module>()
----> 1 densidad_paises_clean.head()


NameError: name 'densidad_paises_clean' is not defined
```

In [10]:
```
densidad_paises_clean.rename(columns={'Population': 'Pop'}, inplace=True)
```

```
---------------------------------------------------------------------------
```

```
NameError                                 Traceback (most recent call last)

<ipython-input-10-993e4b66b44b> in <module>()
----> 1 densidad_paises_clean.rename(columns={'Population': 'Pop'}, inplace=True)


NameError: name 'densidad_paises_clean' is not defined
```

```
In [ ]: densidad_paises_clean.rename(columns={'Area (km2)': 'Area'}, inplace=True)
```

```
In [ ]: densidad_paises_clean.rename(columns={'Density (/km2)': 'Density'}, inplace=True)
```

2.0.1 Exportar ?