Aprendiendo Python para Análisis Estadísticos

Pandas

Juan C. Correa, Ph.D.

j.correa.n@gmail.com

https://correajc.com/

Objetivo de esta Charla

- Comprender los alcances conceptuales y operativos de la librería Pandas como herramienta útil para el análisis estadístico de datos.
- Introducir, a través de un tutorial, los elementos más básicos de la librería de Pandas

Agenda

1. Pandas: Breve Descripción
   - Series
   - Dataframes

Referencias

Breve Descrpipción

Pandas proporciona una variedad de funciones diseñadas para trabajar con datos estructurados de una manera rápida, fácil y expresiva (McKinney, 2012).

Es uno de los ingredientes críticos que permiten que Python sea un entorno de análisis de datos potente y productivo.

Los datos estructurados que se manejan con Pandas se implementan a través de sus dos caballos de fuerza más importantes: ***Series*** y ***Dataframes***.

En Python, un objeto ***Series*** es un vector unidimensional que contiene un conjunto de datos (de cualquier clase `NumPy`) junto a su conjunto de etiquetas de datos (las etiquetas van por fila).

```
(base) juan@pop-os:~$ python
Python 3.8.8 (default, Apr 13 2021, 19:58:26)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from pandas import Series, DataFrame
>>> import pandas as pd
>>> Datos = Series([4, 7, -5, 3])
>>> Datos
0    4
1    7
2   -5
3    3
dtype: int64
>>>
```

El objeto `Datos` tiene cuatro valores (4, 7, -5, 3) cada uno de los cuales tiene su propia etiqueta (0, 1, 2, 3).

```
(base) juan@pop-os:~$ python
Python 3.8.8 (default, Apr 13 2021, 19:58:26)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from pandas import Series, DataFrame
>>> import pandas as pd
>>> Datos = Series([4, 7, -5, 3])
>>> Datos
0    4
1    7
2   -5
3    3
dtype: int64
>>> Datos.index
RangeIndex(start=0, stop=4, step=1)
>>> Datos.values
array([ 4,  7, -5,  3])
>>>
```

Vamos a crear el objeto `Datos2` a partir de otro tipo de estructura de datos conocida como dict y que aquí llamamos `sdata`.

```
>>> sdata = {'Ohio': 35000, 'Texas': 71000, 'Oregon': 16000, 'Utah': 5000}
>>> Datos2 = Series(sdata)
>>> Datos2
Ohio      35000
Texas     71000
Oregon    16000
Utah       5000
dtype: int64
>>>
```

Entre las páginas 112 a 115 del libro de texto de McKinney (2012) se presentan otras características de los objetos `Series`.

Los objetos de tipo `Dataframes` son objetos tipo hoja de cálculo (filas por columnas) con etiquetas para filas y columnas.

In [13]:
```
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
'year': [2000, 2001, 2002, 2001, 2002],
'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
```

In [14]:
```
data
```

Out[14]:
```
{'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
 'year': [2000, 2001, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
```

In [16]:
```
frame = DataFrame(data)
frame
```

Out[16]:

	state	year	pop
0	Ohio	2000	1.5
1	Ohio	2001	1.7
2	Ohio	2002	3.6
3	Nevada	2001	2.4
4	Nevada	2002	2.9

Referencias I

McKinney, W. (2012). *Python for Data Analysis*. Cambridge: O'Reilly.