Análisis de datos con la librería Pandas de Python

```
>>> import pandas as pd
>>> data=[1,2,3,4]
>>> type(data)
<class 'list'>
>>> datos=pd.Series(data)
>>> datos
0    1
1    2
2    3
3    4
dtype: int64
>>> type(datos)
<class 'pandas.core.series.Series'>

   >>> datos.dtype
   dtype('int64')
```

```
>>> valores=[34.7,8.9,9.9,3.14]
>>> valores_serie=pd.Series(valores)
>>> valores_serie.dtype
dtype('float64')
```

Aritmética sobre Series

```
>>> nuevo=valores_serie*3
>>> nuevo
0    104.10
1     26.70
2     29.70
3      9.42
dtype: float64
>>> nuevo2=valores**2
Traceback (most recent call last):
  File "<pyshell#23>", line 1, in <module>
    nuevo2=valores**2
TypeError: unsupported operand type(s) for ** or pow(): 'list' and 'int'
>>> nuevo2=valores+3
Traceback (most recent call last):
  File "<pyshell#24>", line 1, in <module>
    nuevo2=valores+3
TypeError: can only concatenate list (not "int") to list
```

```
>>> import numpy as np
>>> vector=np.array([2,45.5,89.8])
>>> type(vector)
<class 'numpy.ndarray'>
>>> vector_series=pd.Series(vector)
>>> nuevo3=vector_series*2
>>> nuevo4=vector_series**2
>>> nuevo5=vector_series+3
```

```
>>> nuevo3
0      4.0
1     91.0
2    179.6
dtype: float64
>>> nuevo4
0       4.00
1    2070.25
2    8064.04
dtype: float64
>>> nuevo5
0     5.0
1    48.5
2    92.8
dtype: float64
>>> type(nuevo3)
<class 'pandas.core.series.Series'>
>>> nuevo3.dtype
dtype('float64')
```

Accediendo a elementos de la serie

```
>>> nuevo
0    104.10
1     26.70
2     29.70
3      9.42
dtype: float64
>>> nuevo[3]
9.42
```

```
>>> viejo=[23.5,56.7,89.7]
>>> viejo_series=pd.Series(viejo)
>>> viejo_series[2]
89.7
>>> type(viejo_series[2])
<class 'numpy.float64'>
>>> viejo=[2.5,5.7,8.7]
>>> viejo_series=pd.Series(viejo)
>>> viejo_series[2]
8.7
>>> type(viejo_series[2])
<class 'numpy.float64'>
```

```
>>> a=[1,2,3,4]
>>> type(a)
<class 'list'>
>>> a_series=pd.Series(a)
>>> a_series[1]
2
>>> type(a_series[1])
<class 'numpy.int64'>
>>> b=[1.6,2,3,4]
>>> type(b)
<class 'list'>
>>> b_series=pd.Series(b)
>>> b_series[0]
1.6
>>> type(b_series[1])
<class 'numpy.float64'>
```

```
>>> b_series.dtype
dtype('float64')
```

Creando una tabla (DataFrame)

```
>>> valores={'tensión':[1.2,4.6,6.7,9.8],'corriente':[0.3,0.7,0.8,1.3]}
>>> type(valores)
<class 'dict'>
>>> tabla=pd.DataFrame(valores)
>>> type(tabla)
<class 'pandas.core.frame.DataFrame'>
```

```
>>> tabla
   tensión  corriente
0      1.2        0.3
1      4.6        0.7
2      6.7        0.8
3      9.8        1.3
```

```
>>> tabla['corriente']
0    0.3
1    0.7
2    0.8
3    1.3
Name: corriente, dtype: float64
```

Aritmética sobre Tabla de datos (DataFrame)

```
>>> tabla
   tensión  corriente
0      1.2        0.3
1      4.6        0.7
2      6.7        0.8
3      9.8        1.3
```

```
>>> tabla**3
   tensión  corriente
0    1.728      0.027
1   97.336      0.343
2  300.763      0.512
3  941.192      2.197
```

```
>>> tabla*2
   tensión  corriente
0      2.4        0.6
1      9.2        1.4
2     13.4        1.6
3     19.6        2.6
```

```
>>> tabla+45
   tensión  corriente
0     46.2       45.3
1     49.6       45.7
2     51.7       45.8
3     54.8       46.3
```

Accediendo a los valores de Tabla de datos (DataFrame)

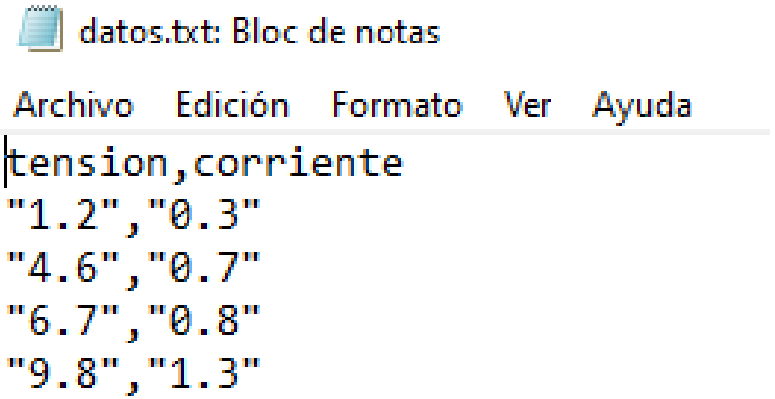
```
>>> tabla
   tensión  corriente
0      1.2        0.3
1      4.6        0.7
2      6.7        0.8
3      9.8        1.3
```

```
>>> tabla['corriente'][2]
0.8
>>> tabla['tensión'][1]
4.6
```

```
>>> tabla.loc[1]
tensión      4.6
corriente    0.7
Name: 1, dtype: float64

>>> tabla.loc[1][0]
4.6
```

Importando datos de un archivo CSV




```
>> tabladatos=pd.read_csv('datos.txt')
>>> tabladatos
   tension  corriente
0      1.2        0.3
1      4.6        0.7
2      6.7        0.8
3      9.8        1.3
```

```
>>> help(pd.read_csv)
```

Squeezed text (290 lines).

```
>>> tabladicy=pd.read_csv('dicyd.csv')
>>> tabladicy
          fecha  apertura  maximo  minimo  cierre  volumen  openint
0    2013-06-04      88.0   88.00   88.00   88.00     2000        0
1    2013-07-11      79.4   79.40   79.40   79.40   149000        0
2    2013-12-19     108.0  108.00  108.00  108.00   185000        0
3    2013-12-30     105.0  105.00  105.00  105.00     3000        0
4    2014-01-13      99.0   99.00   99.00   99.00     2000        0
...         ...       ...     ...     ...     ...      ...
1021 2020-04-21      58.5   59.00   57.00   57.75   519696        0
1022 2020-04-22      55.5   56.50   54.50   56.00   191496        0
1023 2020-04-23      56.0   56.00   53.75   55.90   497269        0
1024 2020-04-24      56.0   56.00   54.00   55.00   311882        0
1025 2020-04-27      54.0   55.85   53.80   55.00   471662        0

[1026 rows x 7 columns]
```

```
>>> columna_apertura=tabladicy['apertura'][:]
>>> type(columna_apertura)
<class 'pandas.core.series.Series'>
>>> columna_apertura.dtype
dtype('float64')
```

```
>>> import matplotlib.pyplot as plt
>>> plt.plot(columna_apertura)
[<matplotlib.lines.Line2D object at 0x000001748B9EFCC8>]
>>> plt.show()
```