Python Pandas

- ```
  pip3 install pandas
  ```

- soporte para excel:

  ```
  pip3.9 install openpyxl # needed to load Excel .xlsx files
  pip3.9 install xlrd # needed to load Excel old .xls files)
  ```

- Estilos: https://docs.bokeh.org/en/latest/docs/user_guide/styling/plots.html#ug-styling-plots
- datos: https://pythonhow.com/media/data/supermarkets.json

varios

importar en dataframe

- `pandas.read_csv('data.txt',header=None)`
- `pandas.read_json('data.json')`

columnas

```
df = pandas.read_csv('data.txt',header=None)
```

- `header=None` para archivos sin nombres de columnas en la primera fila
- acceder columnas:

  ```
  df.columns
  df.columns[inicial:final]
  ```

  - pueden ser indices numéricos o nombres
- acceder indices:

  ```
  df.index
  ```

- asignar nombres de columna:

  ```
  df.columns = ["Col1","Col2","Col3"]
  ```

- asignar indice:

  ```
  df.set_index("nom_col"[,inplace=True, drop=False])
  ```

  - `inplace`: los cambios se aplican en el mismo objeto (variable), Por defecto, los cambios se deberán guardar en otra variable.
  - `drop`: Con False, no saca la columna de los datos para convertirla en índice
  ```

manipulación datos

Para acceder a datos en el dataframe es como un sistema de coordenadas

label based indexing

- loc[start_value_row;final_value_row,start_column_header:final_column_header]

```
df.loc["valor":"valor","columna":"columna"]
list(df.loc[:,"Columna"]) # devuelve como lista los valores de la columna, para todas las filas
```

- se puede indicar solo un valor en los rangos

position based indexing

iloc[start_value_row;final_value_row,start_column_header:final_column_header]

```
df.iloc[1:4,2:5]
```

- `iloc` se comporta como cualquier **slice** en python, el último valor no está incluido. Así, en el ejemplo cogería las filas 1,2,3 y las columnas 2,3,4

borrar columnas

- `drop(«fila»|«columna»)`
- los cambios no son `implace`

```
df.drop("filas",0)
df.drop(df.index[0:3],0) # elimina las filas 0,1,2
df.drop(df.columns[0:3],1) # elimina las columnas 1,2
```

añadir columnas

```
df.shape # tupla con número de filas y columnas
df.shape[0] # número de filas
df.shape[1] # número de columnas
```

```
df["new_column"]=df.shape[0]*["Valor"] # ha de coincidir con el número de filas en el índice
```

```
df["new_column"]=df.["columna"] + "," + "Valor"
```

Transposición, para añadir registro ¿?

```
df_T = df.T # función de transposición
df_T["nuevo_registro"] = ["valores_columnas"["valores_columnas",...]]
```

```
df = df_T.T # función de transposición
```

pandas + bokeh ejemplos

```
# segundo bokeh plot con pandas
from bokeh.plotting import figure
from bokeh.io import output_file, show
import pandas

df=pandas.read_csv("https://pythonizing.github.io/data/bachelors.csv")

x=df["Year"]
y=df["Engineering"]

output_file("Line.html")

f=figure()
f.line(x,y)

show(f)
```

```
import pandas

from bokeh.plotting import figure, output_file, show

df=pandas.read_excel("https://github.com/pythonizing/data/blob/master/verlegenhuken
.xlsx",sheet_name=0)
df["Temperature"]=df["Temperature"]/10
df["Pressure"]=df["Pressure"]/10

p=figure(width=500,height=400,tools='pan')

p.title.text="Temperature and Air Pressure"
p.title.text_color="Gray"
p.title.text_font="arial"
p.title.text_font_style="bold"
p.xaxis.minor_tick_line_color=None
p.yaxis.minor_tick_line_color=None
p.xaxis.axis_label="Temperature (°C)"
p.yaxis.axis_label="Pressure (hPa)"

# old versions:
# p.circle(df["Temperature"],df["Pressure"],size=0.5)

p.scatter(x=df["Temperature"],y=df["Pressure"],size=0.5)
output_file("Weather.html")
show(p)
```

- desde Jupyter Notebook problema leyendo fichero desde URL

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/development:python:pandas

Last update: **10/10/2024 16:14**