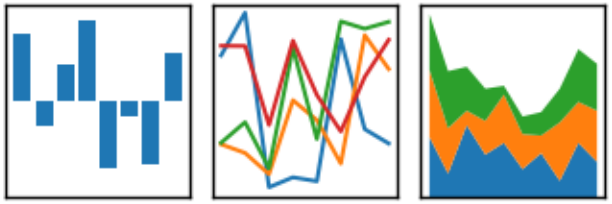
LISBON
SCHOOL OF
ECONOMICS &
MANAGEMENT
UNIVERSIDADE DE LISBOA

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



CARLOS J. COSTA

Pandas

- https://pandas.pydata.org/
- Biblioteca open source,
- Licenciado sobre BSD
- Alto desempenho
- Facil de utilizar
- Inclui ferramentas de estruturas de dados e análise de dados

Pandas

- **DataFrame**
  - É uma estrutura etiquetada
  - Tem colunas com tipos de dados potencialmente diferentes
  - Parecido com folha de cáclulo ou tabela SQL
  - É o objeto mais utilizado pelao pandas
  - Para além dos dados pode passer ainda as colunas (etiquetas das colunas) e indices (ediquetas das linas)

Pandas

- Criar dataframe a partir de dicionário

```
import pandas as pd
    d = {'col1': [1,2,1,3,1,2], 'col2': [1,2,3,4,5,6]}
    df = pd.DataFrame(data=d)
    df.count()
    df['col1'].value_counts()
    df['col1'][1]=5
```

Pandas

- Copiar coluna

  coluna1=df['col1’]

  coluna1[2]=99

- Qual o resultado em coluna1 e df?

  nova_coluna1 = coluna1.copy()

  nova_coluna1[2]=9999

- E agora?

Pandas

- No collaboratoy:

  from google.colab import files

  files.upload()

- Depois no final

  files.download(‘nome do ficheiro')

Pandas

- No computador, colocar o ficheiro de dados na mesma pasta onde está gravado o notebook

Pandas

- Importar pandas

```
import pandas as pd
df = pd.read_csv(‘factbook.csv')
```

Pandas

- Analisar informação

```
df.head()  #cinco linhas
df.info()
df.describe()
df.columns
```

Pandas

- DataFrame.loc
- Acede a um grupo de linhas e colunas por etiqueta (s) ou uma matriz booleana.
- loc [] é principalmente baseado em etiquetas, mas também pode ser usado com uma matriz booleana.

Pandas

- DataFrame.at
  - Acede a um único valor para um par de etiquetas de linha/coluna.
- DataFrame. iloc
  - acede a um grupo de linhas e colunas por posição inteira (s).
- DataFrame. Xs
  - Retorna uma seção transversal (linha (s) ou coluna (s)) da série/DataFrame.
- Série. Loc
  - Acede a um grupo de valores utilizando etiquetas.

Pandas

- Celulas:

  df.iloc[195][0]

- Linhas:

  df.iloc[[195][0]]

- Colunas:

  df.loc[:,'GDPpercapita']

Pandas

- Tipos de dados

df.dtypes

- Se o resultado for object há a necessidade de converter uma coluna complete com etiqueta espefica para numérico

df.loc[:,'GDPpercapita']=**pd.to_numeric(df['GDPpercapita'], errors='coerce')**

pd.to_numeric(argmento, errors)

Pode ser lista, tuplo, array, série 1D

Errors pode ser ignore, raise ou coerce. Este último converte em NAN

Pandas

- Obviamente, se for necessário também nas outras variáveis:
- Pode-se entretanto

```
df.loc[:,'GDPpercapita']=pd.to_numeric(df['GDPpercapita'], errors='coerce')
df.loc[:,'Military_percent_GDP']=pd.to_numeric(df['Military_percent_GDP'], errors='coerce')
df.loc[:,'Unemployment rate(%)']=pd.to_numeric(df['Unemployment rate(%)'], errors='coerce')
```

- Pode-se ver o resultado:

```
df.dtypes
```

Pandas

- Criar uma nova dataframe

  YX = df[['GDPpercapita','Military_percent_GDP','Unemployment rate(%)']]

- e

  YX.dtypes

- Tudo numericos claro…

Pandas

- Eliminar *missing values* de toda a matriz

```
YX=YX.dropna()
```

- Criar o X e Y:

```
Y = YX[['GDPpercapita']]
X = YX[['Military_percent_GDP','Unemployment rate(%)']]
```

Pandas

- Para criar uma coluna correspondente a internet per capita é necessário fazer simplesmente:

```
df['internetpercapita']=df['Internet users']/df['Population’]
```

Bibliografia

- https://pandas.pydata.org/
- https://pandas.pydata.org/pandas-docs/stable/getting_started/10min.html
- https://scikit-learn.org/
- https://scikit-learn.org/stable/index.html
- https://www.statsmodels.org/stable/index.html