

<b>FT 013</b>
<b>Curso:</b> UFCD 10793
<b>UFCD/Módulo/Temática:</b> UFCD 10793 - Fundamentos de Python
<b>Ação:</b> 10793_05/N
<b>Formador/a:</b> Sandra Liliana Meira de Oliveira
<b>Data:</b>
<b>Nome do Formando/a:</b>

Utilizando os ficheiros fornecidos, vamos reproduzir os exemplos da apresentação. Devem colocar os ficheiros .csv na pasta onde cria as scripts python.

Começam por abrir os ficheiros e analisar a informação nos mesmos

```
#import pandas
import pandas as pd

#ler ficheiro cast.csv para dataframe casts
casts = pd.read_csv('cast.csv', index_col=None)

#dados estatisticos do conjunto
print(casts.describe())

#imprimir as cinco primeiras linhas do dataframe casts
print(casts.head())

#ler ficheiro titles.csv para dataframe titles
titles = pd.read_csv('titles.csv', index_col=None)

#imprimir as cinco últimas linhas do dataframe titles
print(titles.tail())

#alterar numero de linhas e colunas a visualizar sempre que se imprime todo o
dataframe
pd.set_option('display.max_rows', 10, 'display.max_columns', 10)

#imprimir dataframe titles
print(titles)

#atribuir à variável tamanho o numero total d linhas do dataframe titles
tamanho=len(titles)

#imprimir tamanho
```

```
print(tamanho)

#imprimir primeiras três linhas do dataframe titles
print(titles.head(3))

#obter serie t a partir do dataframe titles selecionando a coluna title
t = titles['title']

#imprimir tipo de dados de t
print(type(t))

#Obter informação geral do dataframe casts
casts.info()

#imprimir linha de indice 0 no dataframe titles
print(titles.iloc[0])

#obter dataframe after85 filtrando o dataframe titles selecionando os registos cujo
ano é superior a 1985
after85 = titles[titles['year'] > 1985]
print(after85.head())

t = titles
#obter dataframe movies90 filtrando o dataframe t selecionando os registos cujo ano é
superior e igual a 1990 e inferior a 2000
movies90 = t[ (t['year']>=1990) & (t['year']<2000)]
print(movies90.head())

#obter dataframe macbeth a partir do dataframe t cujo valor na coluna title é igual a
Macbeth
macbeth = t[ t['title'] == 'Macbeth']
print(macbeth.head())

#obter dataframe macbeth a partir do dataframe t cujo valor na coluna title é igual a
Macbeth e ordenar por index
macbeth = t[ t['title'] == 'Macbeth'].sort_index()
print(macbeth.head())

#obter dataframe macbeth a partir do dataframe t cujo valor na coluna title é igual a
Macbeth e ordenar pelos values
macbeth = t[ t['title'] == 'Macbeth'].sort_values('year')
print(macbeth.head())

#imprimir linhas 3 e 4 do dataframe casts
print(casts.loc[3:4])

c = casts
```

```
#devolve true em cada elemento da linha correspondente à coluna n se o valor for nulo
e false caso contrário
print(c['n'].isnull().head())

#devolve false em cada elemento da linha correspondente à coluna n se o valor for
nulo e true caso contrário
print(c['n'].notnull().head())

#devolve as primeiras linhas com valor nulo na coluna n
print(c[c['n'].isnull()].head(3))

#substitui os valores nulos de n por 'NA'
c_fill = c[c['n'].isnull()].fillna('NA')
print(c_fill.head(2))

t = titles
#imprime as linhas do dataframe cujo valor afetos à coluna title é igual a 'Maa'
print(t[t['title'] == 'Maa'])

#imprime as linhas do dataframe cujp valor afetos à coluna title é inicia com a 'Maa'
print(t[t['title'].str.startswith("Maa ")]).head(3))

#conta os valores não nulos presentes na coluna year
print(t['year'].value_counts().head())

c = casts

#obtem o dataframe cf a partir de c filtrando os dados que verifiquem a condição do
autor ser 'Aaron Abrams'
cf = c[c['name'] == 'Aaron Abrams']

#Agrupar os dados do dataframe anterior pela coluna year
print(cf.groupby(['year']).size().head())

#group by com múltiplas colunas
gbmultiplecolumns = cf.groupby(['year', 'title']).size()
gbmultiplecolumns.head()

#leitura dos dados do ficheiro release_dates.csv para o dataframe release
release = pd.read_csv('release_dates.csv', index_col=None)
print(release.head())

c_amelia = casts[ casts['title'] == 'Amelia' ]
print(c_amelia.head())
print(release [ release['title'] == 'Amelia' ].head())
```

```
#imprimir as primeiras cinco linhas do merge (fusão) c_amelia com o dataframe  
release  
print(c_amelia.merge(release).head())
```