escola britânica de artes criativas & tecnologia

Módulo 07 | Python: Programação Orientada a Objetos

Caderno de **Exercícios**
Professor André Perez

Tópicos

1. from / import / as;
2. Módulo;
3. Pacote;
4. Baixando pacotes.

Exercícios

0. Preparação do ambiente

Neste exercício vamos utilizar a base de dados de ações da bolsa de valores dos EUA, a Dow Jones. Os dados estão disponíveis para *download* neste link. Vamos utilizar o pacote `wget` para fazer o *download* dos dados.

- Instalando o pacote `wget` na versão 3.2.

```
pip install wget==3.2
```

- Fazendo o download dos dados no arquivo compactado `dados.zip`.

```
import wget

wget.download(url='https://archive.ics.uci.edu/ml/machine-learning-databases/00312/dow_jones_index.zip', out='./dados.zip')
```

```
'./dados (1).zip'
```

- Descompactando os `dados` na pasta dados com o pacote nativo `zipfile`.

```
import zipfile

with zipfile.ZipFile('./dados.zip', 'r') as fp:
  fp.extractall('./dados')
```

Verifique a pasta dados criada, ela deve conter dois arquivos:

- **dow_jones_index.data**: um arquivo com os dados;
- **dow_jones_index.names**: um arquivo com a descrição completa dos dados.

É possível observar que o arquivo de dados é um arquivo separado por virgulas, o famoso `csv`. Vamos renomear o arquivo de dados para que ele tenha a extensão `csv` com o pacote nativo `os`.

- Renomeando o arquivo com o pacote nativo `os`.

```
import os

os.rename('./dados/dow_jones_index.data', './dados/dow_jones_index.csv')
```

```
---------------------------------------------------------------------------
FileExistsError                           Traceback (most recent call last)
c:\Users\Soldado\Downloads\Python\Modulos\Exercícios\Python_M7_exercise.ipynb Cella 16 line 3
      <a href='vscode-notebook-cell:/c%3A/Users/Soldado/Downloads/Python/Modulos/Exerc%C3%ADcios/Python_M7_exercise.ipynb#Y426sZmls
----> <a href='vscode-notebook-cell:/c%3A/Users/Soldado/Downloads/Python/Modulos/Exerc%C3%ADcios/Python_M7_exercise.ipynb#Y426sZmls

FileExistsError: [WinError 183] Impossibile creare un file, se il file esiste già: './dados/dow_jones_index.data' -> './dados/dow_j
```

RICERCA SU STACK OVERFLOW

Pronto! Abra o arquivo e o Google Colab irá apresentar uma visualização bem legal dos dados.

---

1. Pandas

Para processar os dados, vamos utilizar o pacote `pandas` na versão `1.1.5`. A documentação completa por ser encontrada neste link

```
!pip install pandas==1.1.5
```

Vamos importar o pacote com o apelido (alias) `pd`.

```
import pandas as pd
```

Estamos prontos para ler o arquivo.

```
df = pd.read_csv('./dados/dow_jones_index.csv')
```

O pandas trabalha com o conceito de dataframe, uma estrutura de dados com muitos métodos e atributos que aceleram o processamento de dados. Alguns exemplos:

- Visualizando as `n` primeiras linhas:

```
df.head(n=10)
```

	quarter	stock	date	open	high	low	close	volume	percent_change_price	percent_change_volume_over_last_wk	pre
0	1	AA	1/7/2011	$15.82	$16.72	$15.78	$16.42	239655616	3.792670	NaN	
1	1	AA	1/14/2011	$16.71	$16.71	$15.64	$15.97	242963398	-4.428490	1.380223	
2	1	AA	1/21/2011	$16.19	$16.38	$15.60	$15.79	138428495	-2.470660	-43.024959	
3	1	AA	1/28/2011	$15.87	$16.63	$15.82	$16.13	151379173	1.638310	9.355500	
4	1	AA	2/4/2011	$16.18	$17.39	$16.18	$17.14	154387761	5.933250	1.987452	
5	1	AA	2/11/2011	$17.33	$17.48	$16.97	$17.37	114691279	0.230814	-25.712195	
6	1	AA	2/18/2011	$17.39	$17.68	$17.28	$17.28	80023895	-0.632547	-30.226696	
7	1	AA	2/25/2011	$16.98	$17.15	$15.96	$16.68	132981863	-1.766780	66.177694	
8	1	AA	3/4/2011	$16.81	$16.94	$16.13	$16.58	109493077	-1.368230	-17.663150	
9	1	AA	3/11/2011	$16.58	$16.75	$15.42	$16.03	114332562	-3.317250	4.419900	

- Visualizando o nome das colunas:

```
df.columns.to_list()
```

```
['quarter',
 'stock',
 'date',
 'open',
 'high',
 'low',
 'close',
 'volume',
 'percent_change_price',
 'percent_change_volume_over_last_wk',
 'previous_weeks_volume',
 'next_weeks_open',
 'next_weeks_close',
 'percent_change_next_weeks_price',
 'days_to_next_dividend',
 'percent_return_next_dividend']
```

- Verificando o número de linhas e colunas.

```
linhas, colunas = df.shape
print(f'Número de linhas: {linhas}')
print(f'Número de colunas: {colunas}')
```

```
Número de linhas: 750
Número de colunas: 16
```

Vamos selecionar os valores de abertura, fechamento, máximo e mínimo das ações do McDonalds, listado na Dow Jones como MCD:

- Selecionando as linha do dataframe original `df` em que a coluna `stock` é igual a `MCD`.

```
df_mcd = df[df['stock'] == 'MCD']
```

- Selecionando apenas as colunas de data e valores de ações.

```
df_mcd = df_mcd[['date', 'open', 'high', 'low', 'close']]
```

Excelente, o problema é que as colunas com os valores possuem o carater `$` e são do tipo texto (`object` no `pandas`).

```
df_mcd.head(n=10)
```

	date	open	high	low	close
216	1/7/2011	$77.10	$77.59	$73.59	$74.37
217	1/14/2011	$74.25	$74.49	$72.46	$74.06
218	1/21/2011	$74.65	$75.75	$74.31	$75.01
219	1/28/2011	$74.25	$75.85	$73.05	$73.28
220	2/4/2011	$73.80	$74.50	$73.08	$74.05
221	2/11/2011	$74.13	$76.32	$73.30	$76.14
222	2/18/2011	$76.07	$76.45	$75.70	$76.13
223	2/25/2011	$75.95	$76.45	$74.42	$74.44
224	3/4/2011	$74.51	$76.63	$73.64	$76.03
225	3/11/2011	$76.38	$77.25	$74.97	$76.73

```
df_mcd.dtypes
```

```
date     object
open     object
high     object
low      object
close    object
dtype: object
```

Vamos limpar as colunas com o método `apply`, que permite a aplicação de uma função anônima (`lambda`) qualquer. A função `lambda` remove o caracter **$** e faz a conversão do tipo de `str` para `float`.

```
for col in ['open', 'high', 'low', 'close']:
  df_mcd[col] = df_mcd[col].apply(lambda value: float(value.split(sep='$')[-1]))
```

Verifique novamente os dados e seus tipos.

```
df_mcd.head(n=10)
```

	date	open	high	low	close
216	1/7/2011	77.10	77.59	73.59	74.37
217	1/14/2011	74.25	74.49	72.46	74.06
218	1/21/2011	74.65	75.75	74.31	75.01
219	1/28/2011	74.25	75.85	73.05	73.28
220	2/4/2011	73.80	74.50	73.08	74.05
221	2/11/2011	74.13	76.32	73.30	76.14
222	2/18/2011	76.07	76.45	75.70	76.13
223	2/25/2011	75.95	76.45	74.42	74.44
224	3/4/2011	74.51	76.63	73.64	76.03
225	3/11/2011	76.38	77.25	74.97	76.73

```
df_mcd.dtypes
```

```
date      object
open     float64
high     float64
low      float64
close    float64
dtype: object
```

Excelente, agora podemos explorar os dados visualmente.

Agora é a sua vez! Conduza o mesmo processo para extrair e tratar os dados da empresa Coca-Cola (`stock` column igual a `KO`).

- Selecionando as linha do dataframe original `df` em que a coluna `stock` é igual a `KO`.

```
# extração e tratamento dos dados da empresa Coca-Cola.

import pandas as pd
df = pd.read_csv('./dados/dow_jones_index.csv')

df_ko = df[df['stock'] == 'KO']
```

Vamos selecionar os valores de abertura, fechamento, máximo e mínimo das ações da empresa Coca-Cola, listado na Dow Jones como KO:

- Selecionando apenas as colunas de data e valores de ações.

```
df_ko = df_ko[['date', 'open', 'close']]
```

Excelente, o problema é que as colunas com os valores possuem o carater `$` e são do tipo texto (`object` no `pandas`).

```
# Visualize os dados do dataframe
df_ko.head(n=10)
```

```
              date    open   close
```

```
# Verifique o tipo dos dados
df_ko.dtypes
```

```
     date     object
     open     object
     close    object
     dtype: object
```

Vamos limpar as colunas com o método `apply`, que permite a aplicação de uma função anônima (`lambda`) qualquer. A função `lambda` remove o caracter **$** e faz a conversão do tipo de `str` para `float`.

```
     211  2/25/2011  $63.36  $64.31
```

```
for col in ['open', 'close']:
  df_ko[col] = df_ko[col].apply(lambda value: float(value.split(sep='$')[-1]))
```

```
     213  3/11/2011  $65.32  $64.81
```

Verifique novamente os dados e seus tipos.

```
# Visualize novamente os dados do dataframe
df_ko.head(n=10)
```

	date	open	close
204	1/7/2011	65.88	62.92
205	1/14/2011	62.70	63.13
206	1/21/2011	63.21	62.77
207	1/28/2011	62.87	62.21
208	2/4/2011	62.32	62.56
209	2/11/2011	62.67	63.57
210	2/18/2011	63.67	64.55
211	2/25/2011	63.36	64.31
212	3/4/2011	64.17	65.21
213	3/11/2011	65.32	64.81

```
# Verifique novamente o tipo dos dados
df_ko.dtypes
```

```
     date      object
     open     float64
     close    float64
     dtype: object
```

Excelente, agora podemos explorar os dados visualmente.

---

2. Seaborn

Para visualizar os dados, vamos utilizar o pacote `seaborn` na versão `0.11.1`. A documentação completa por ser encontrada neste link

```
!pip install seaborn==0.11.1
```

Vamos importar o pacote com o apelido (alias) `sns`.

```
import seaborn as sns
```

Vamos visualizar os valores de abertura das ações ao longo do tempo.

```
plot = sns.lineplot(x="date", y="open", data=df_mcd)
_ = plot.set_xticklabels(labels=df_mcd['date'], rotation=90)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
c:\Users\Soldado\Downloads\Python\Modulos\Exercícios\Python_M7_exercise.ipynb Cella 69 line 1
----> <a href='vscode-notebook-cell:/c%3A/Users/Soldado/Downloads/Python/Modulos/Exerc%C3%ADcios/Python_M7_exercise.ipynb#Y461sZmls
      <a href='vscode-notebook-cell:/c%3A/Users/Soldado/Downloads/Python/Modulos/Exerc%C3%ADcios/Python_M7_exercise.ipynb#Y461sZmls

NameError: name 'df_mcd' is not defined
```

RICERCA SU STACK OVERFLOW

Vamos também visualizar os valores de fechamento das ações ao longo do tempo.

```
plot = sns.lineplot(x="date", y="close", data=df_mcd)
_ = plot.set_xticklabels(labels=df_mcd['date'], rotation=90)
```

Para facilitar a comparação, vamo visualizar os quatro valores no mesmo gráfico.

```
plot = sns.lineplot(x="date", y="value", hue='variable', data=pd.melt(df_mcd, ['date']))
_ = plot.set_xticklabels(labels=df_mcd['date'], rotation=90)
```

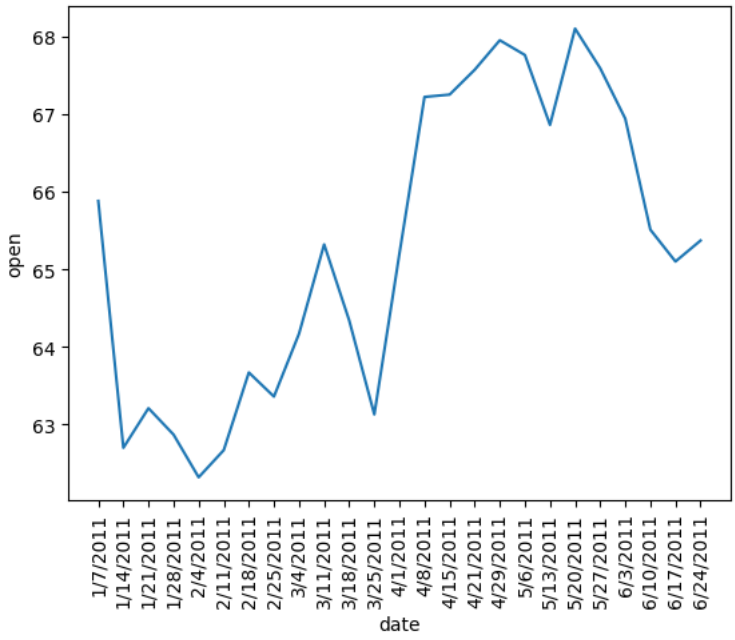
Para finalizar, vamos salvar o gráfico numa figura.

```
plot.figure.savefig("./coca-cola.png")
```

Agora é a sua vez, faça o gráfico acima para a empresa Coca-Cola e salve a imagem com o nome `ko.png`.

```
# visualização dos dados da Coca-Cola.
plot = sns.lineplot(x="date", y="open", data=df_ko)
_ = plot.set_xticklabels(labels=df_ko['date'], rotation=90)
```

```
c:\Users\Soldado\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_core.py:1218: FutureWarning: is_categorical_dty
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\Soldado\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_core.py:1218: FutureWarning: is_categorical_dty
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\Soldado\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_core.py:1218: FutureWarning: is_categorical_dty
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\Soldado\AppData\Local\Temp\ipykernel_12884\437458.py:3: UserWarning: set_ticklabels() should only be used with a fixed num
  _ = plot.set_xticklabels(labels=df_ko['date'], rotation=90)
```



Vamos visualizar os valores de abertura das ações ao longo do tempo.

```
plot = sns.lineplot(x="date", y="open", data=df_ko)
_ = plot.set_xticklabels(labels=df_ko['date'], rotation=90)
```

Vamos também visualizar os valores de fechamento das ações ao longo do tempo.

```
plot = sns.lineplot(x="date", y="close", data=df_ko)
_ = plot.set_xticklabels(labels=df_ko['date'], rotation=90)
```

Para facilitar a comparação, vamo visualizar os quatro valores no mesmo gráfico.

```
plot = sns.lineplot(x="date", y="value", hue='variable', data=pd.melt(df_ko, ['date']))
_ = plot.set_xticklabels(labels=df_ko['date'], rotation=90)
```

Para finalizar, vamos salvar o gráfico numa figura.

```
plot.figure.savefig("./coca-cola.png")
```

Analise as duas imagens e escreva pelo menos um *insight* que você consegue extrair dos dados. Fique a vontade para escrever quantos *insights* você quiser.

Obs: *Insights* são observações sobre o que você percebe/entende/interpreta em suas análises. No caso deste exercício, você vai analisar os dados dos gráficos da empresa McDonalds e da empresa Cola-Cola e notar o que os dados gerados podem ser interessante, que tipo de interpretação o comportamento dos dados estão te trazendo.

Insight #1: ...

O gráfico apresenta duas linhas que representam os valores de fechamento (linha laranja) e abertura (linha azul) das ações de uma determinada empresa no período de tempo.

Em geral, o valor de fechamento é superior ao valor de abertura, o que pode indicar que o preço das ações tende a aumentar ao longo do dia de negociação. No entanto, existem alguns pontos onde as duas linhas se cruzam.

Insight #2: ...

Podemos observar que o intervalo de valores no eixo y está entre 62 e 68. Isto sugere que, apesar das flutuações diárias, o preço das ações permaneceu relativamente estável ao longo do período representado em o gráfico.

Além disso, não parece haver uma clara tendência ascendente ou descendente durante o período considerado. Isso pode indicar que a empresa teve um desempenho estável nesse período.
