Object-oriented programming: Web Import

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.

Preparação do ambiente - pip install wget==3.2

```python
import wget

wget.download(url='link', out='./dados.zip')
```

https://archive.ics.uci.edu/ml/machine-learning-databases/00312/dow_jones_index.zip

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

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

```python
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`.

Pandas - pip install pandas==1.1.5

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

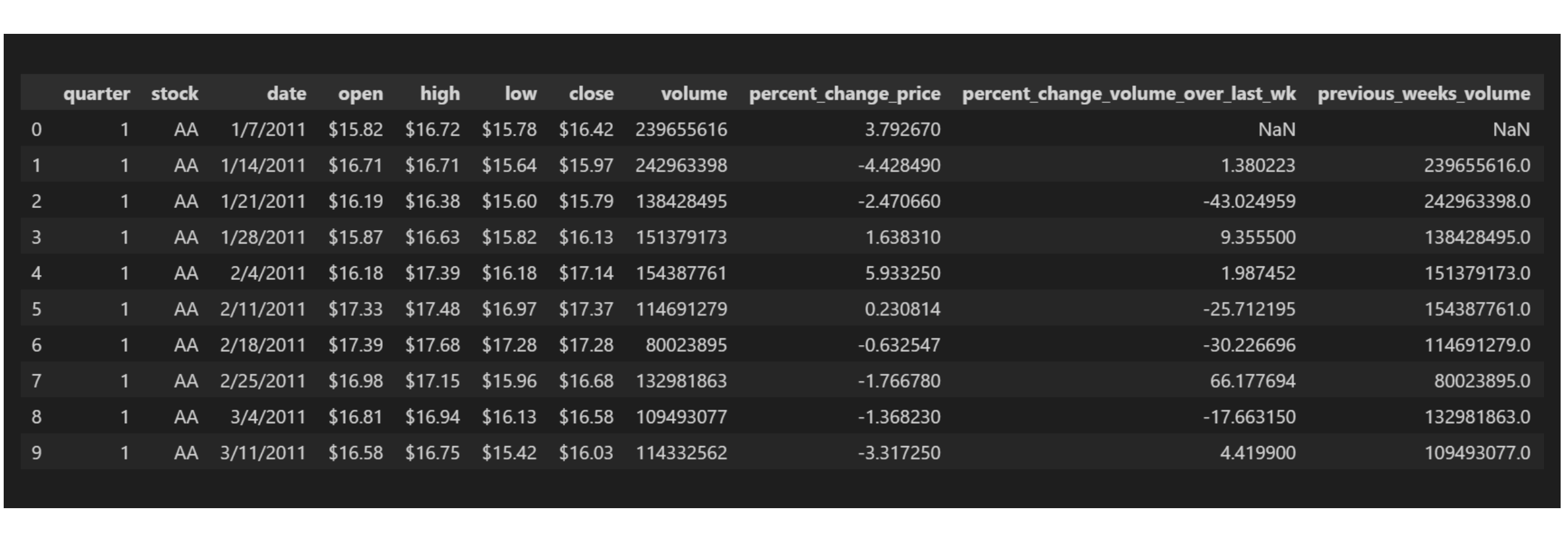
Vamos importar o pacote com o apelido (alias) `pd`

Estamos prontos para ler o arquivo.

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:

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



	quarter	stock	date	open	high	low	close	volume	percent_change_price	percent_change_volume_over_last_wk	previous_weeks_volume
0	1	AA	1/7/2011	$15.82	$16.72	$15.78	$16.42	239655616	3.792670	NaN	NaN
1	1	AA	1/14/2011	$16.71	$16.71	$15.64	$15.97	242963398	-4.428490	1.380223	239655616.0
2	1	AA	1/21/2011	$16.19	$16.38	$15.60	$15.79	138428495	-2.470660	-43.024959	242963398.0
3	1	AA	1/28/2011	$15.87	$16.63	$15.82	$16.13	151379173	1.638310	9.355500	138428495.0
4	1	AA	2/4/2011	$16.18	$17.39	$16.18	$17.14	154387761	5.933250	1.987452	151379173.0
5	1	AA	2/11/2011	$17.33	$17.48	$16.97	$17.37	114691279	0.230814	-25.712195	154387761.0
6	1	AA	2/18/2011	$17.39	$17.68	$17.28	$17.28	80023895	-0.632547	-30.226696	114691279.0
7	1	AA	2/25/2011	$16.98	$17.15	$15.96	$16.68	132981863	-1.766780	66.177694	80023895.0
8	1	AA	3/4/2011	$16.81	$16.94	$16.13	$16.58	109493077	-1.368230	-17.663150	132981863.0
9	1	AA	3/11/2011	$16.58	$16.75	$15.42	$16.03	114332562	-3.317250	4.419900	109493077.0

- Visualizando o nome das colunas:

```python
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']
```

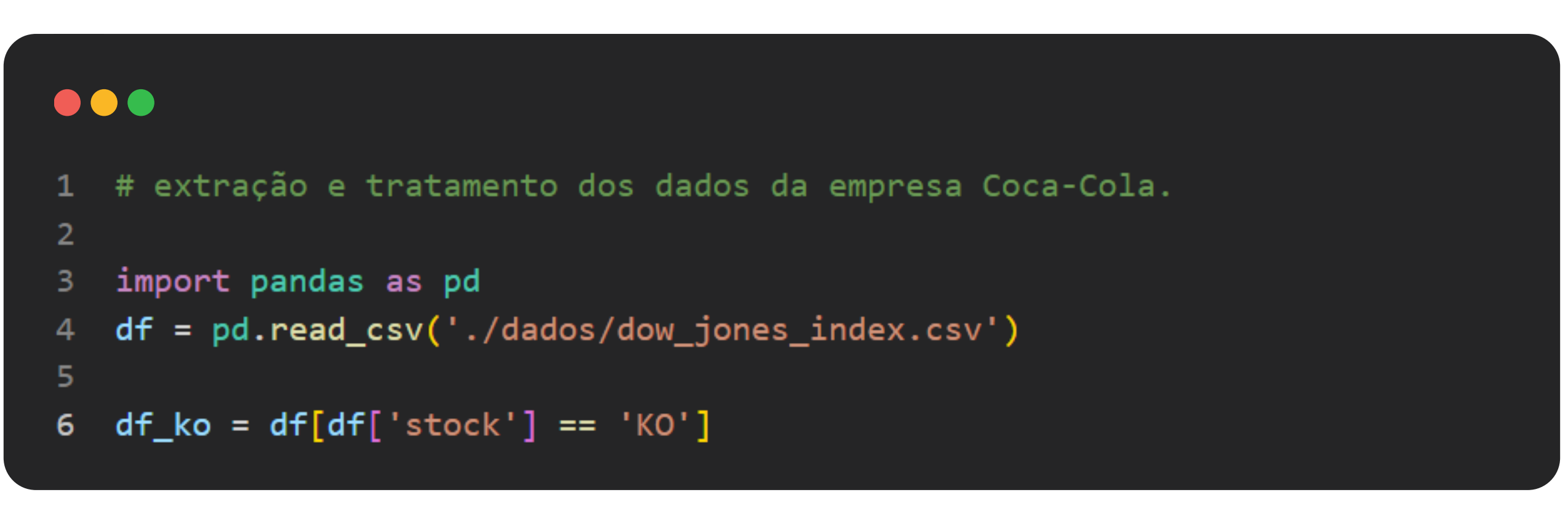
- Verificando o número de linhas e colunas:

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

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

Extrair e tratar os dados da empresa Coca-Cola

(`stock` column igual a `KO`)



```python
# 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']
```

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

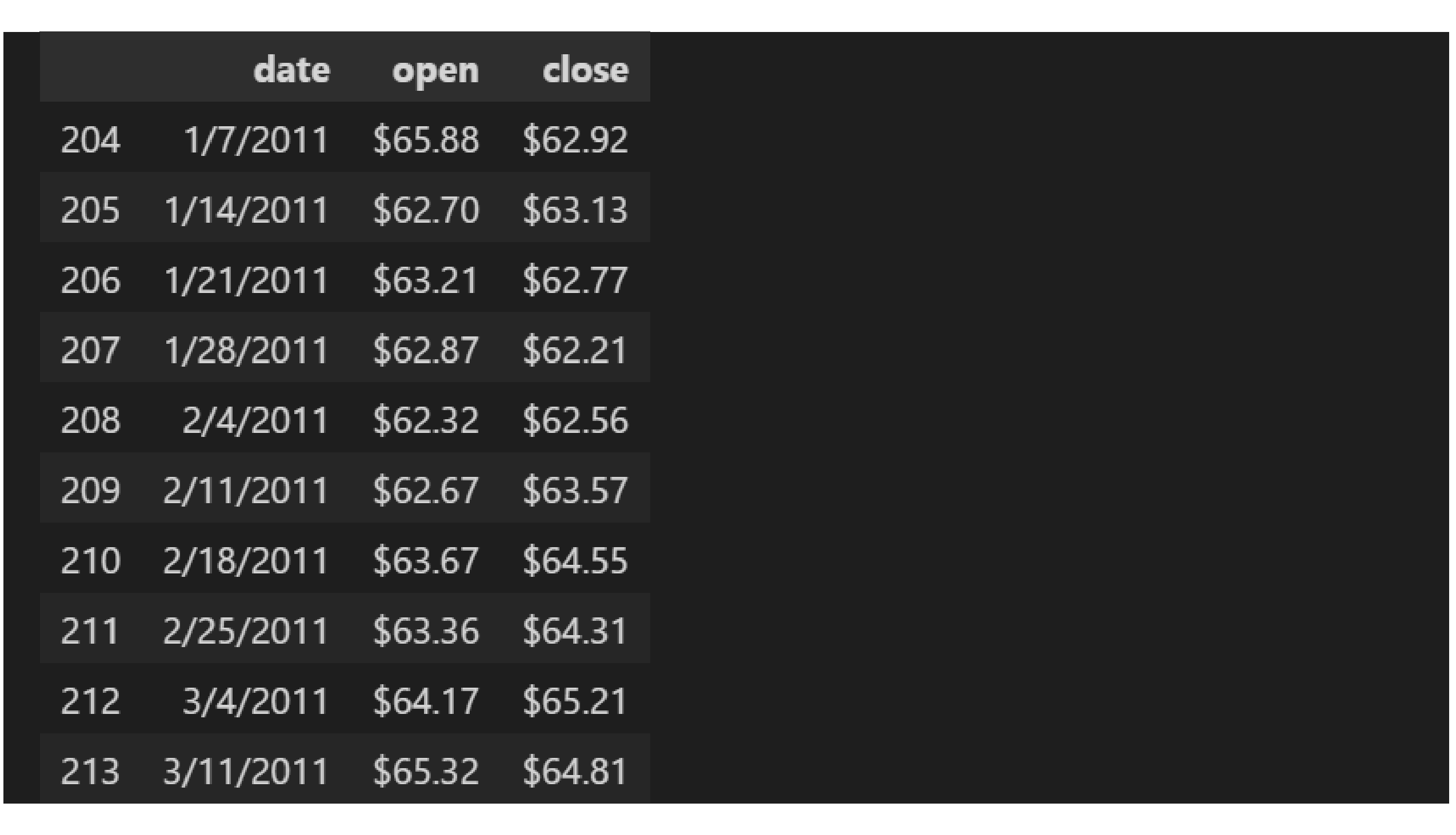
Vamos selecionar os valores de abertura, fechamento, máximo e mínimo das ações da empresa Coca-Cola, listado na Dow Jones como KO:

- Selecionando apenas as colunas de data e valores de ações

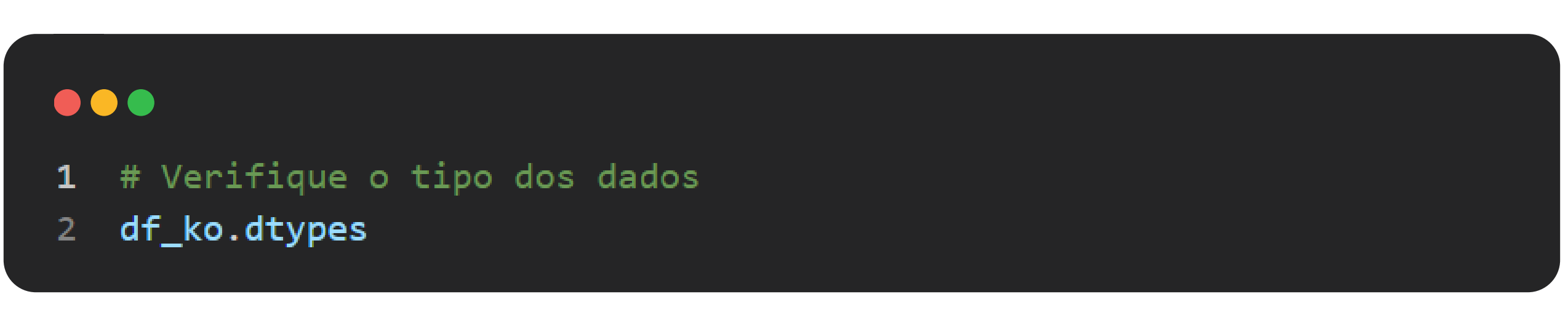
```python
df_ko = df_ko[['date', 'open', 'close']]
```

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

```python
# Visualize 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



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

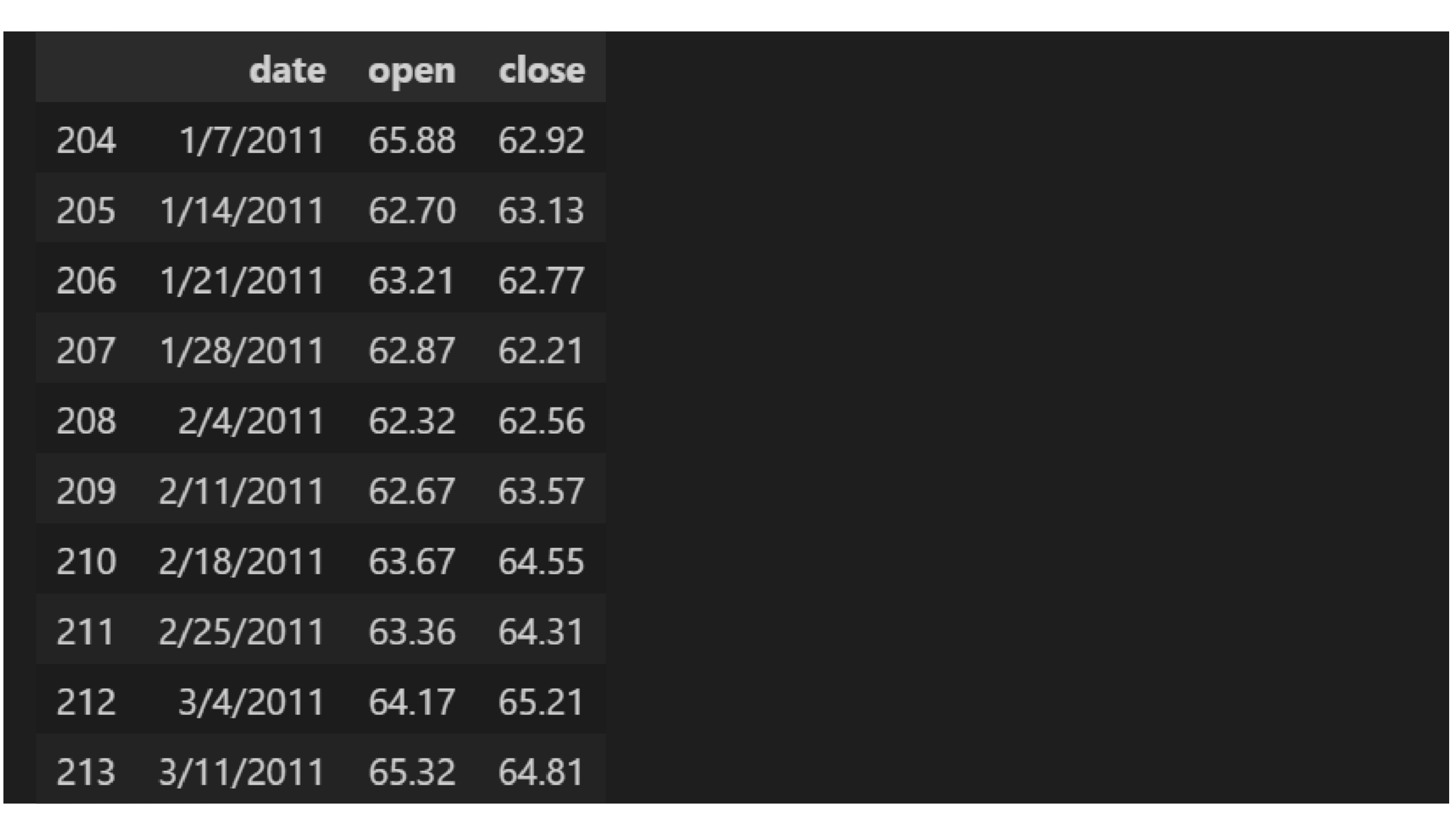
```
date     object
open     object
close    object
dtype: object
```

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`.

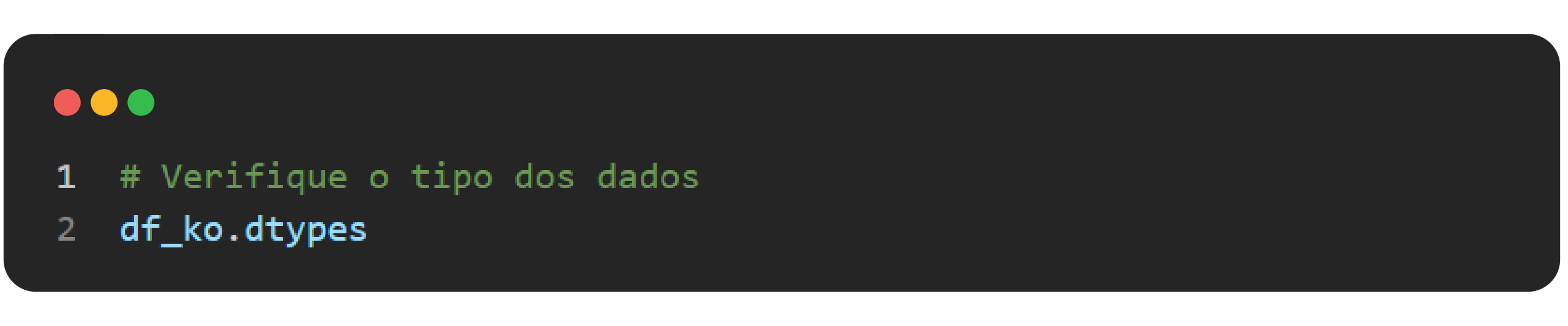
```python
for col in ['open', 'close']:
  df_ko[col] = df_ko[col].apply(lambda value: float(value.split(sep='$')[-1]))
```

- Verifique novamente os dados e seus tipos.

```python
# Visualize 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



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

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

Excelente, agora podemos explorar os dados visualmente.

SEABORN - pip install seaborn==0.11.1

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

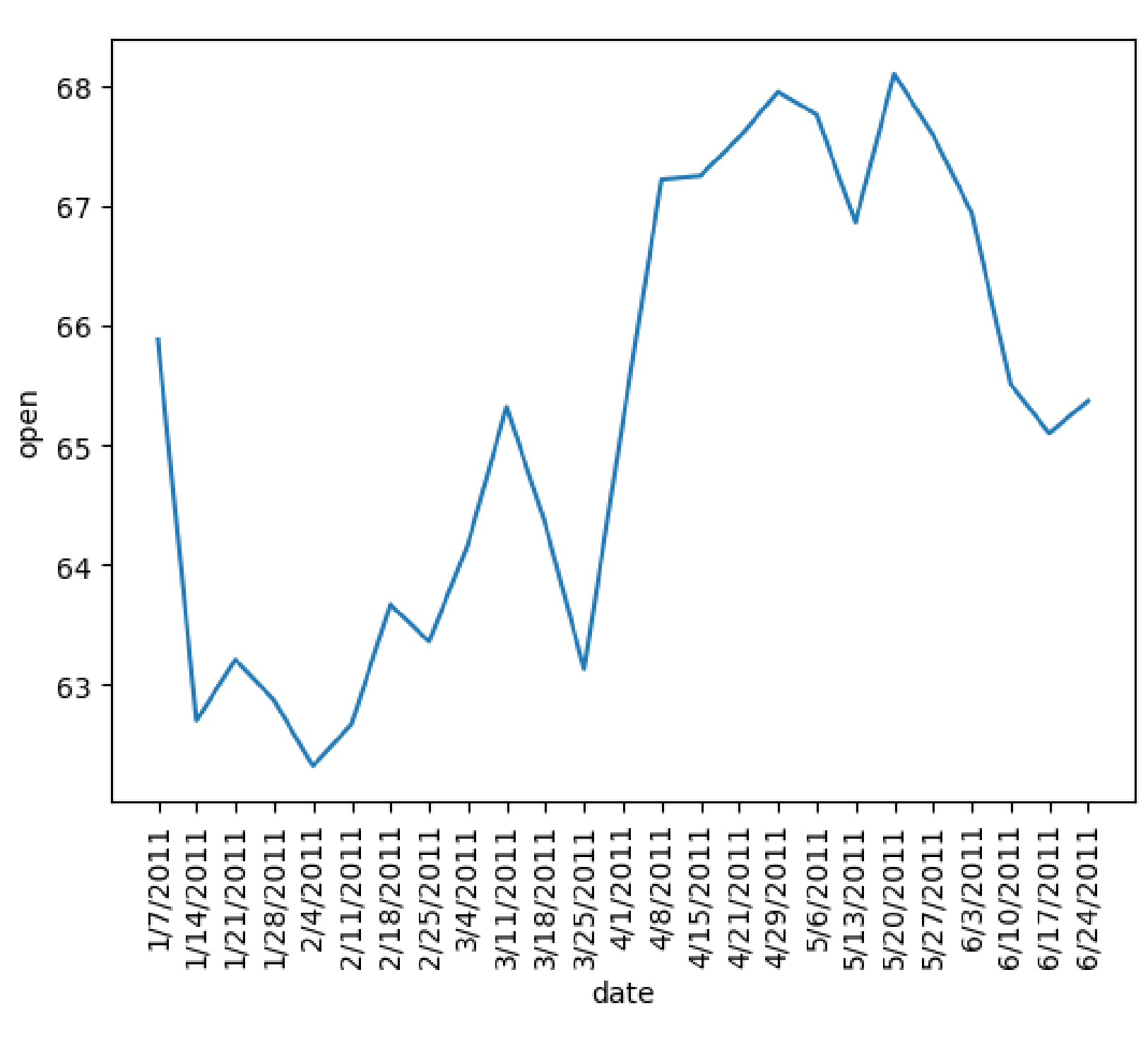
- Vamos importar o pacote com o apelido (alias) `sns`.

```python
import seaborn as sns
```

Gráfico da empresa Coca-Cola e a imagem com o nome `coca-cola.png`.

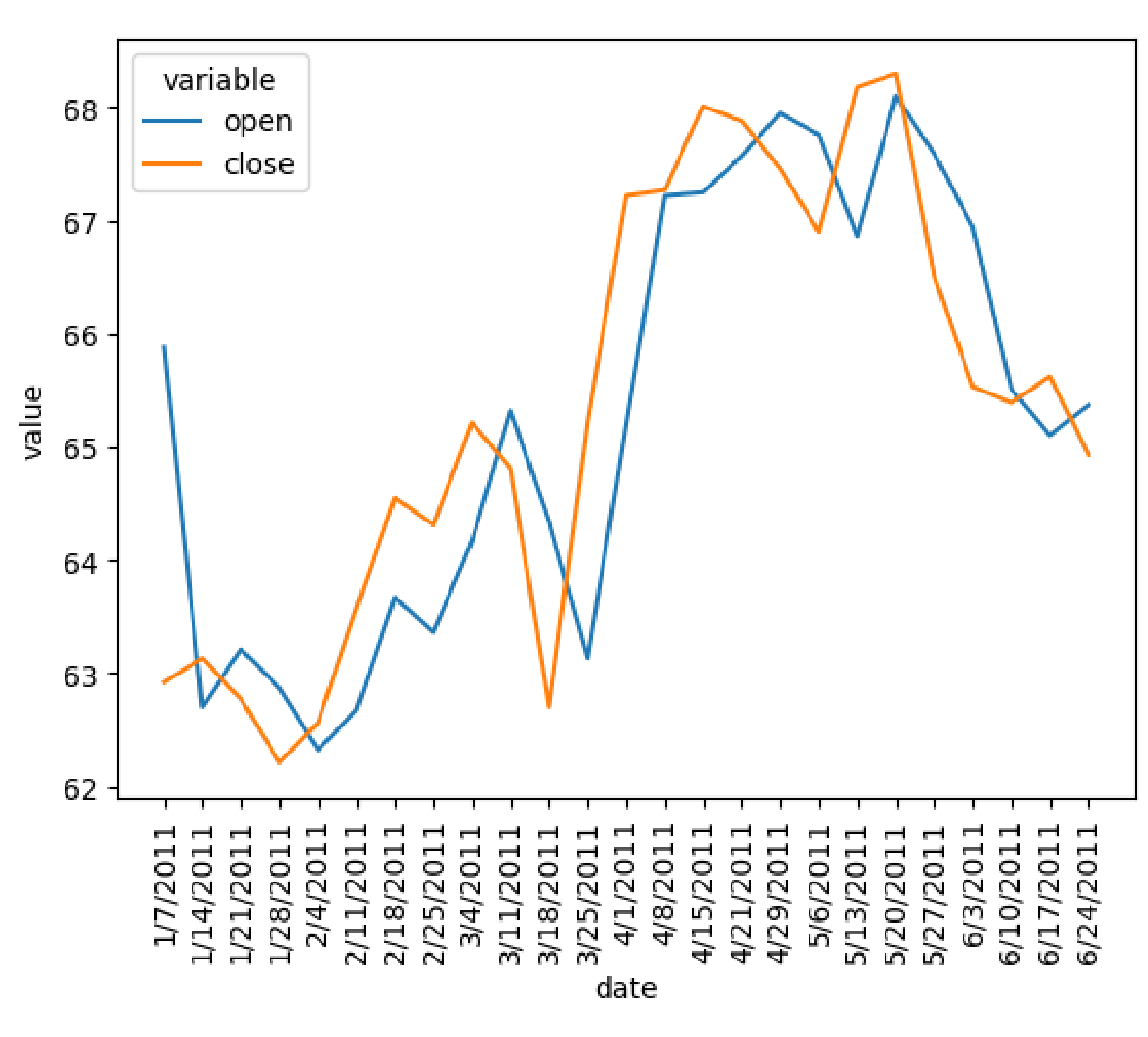
```python
plot.figure.savefig("./coca-cola.png")
```

```python
# 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)
```



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

```python
# 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)
```



By Enzo Schitini