

INTRODUÇÃO AO PYTHON PARA ANÁLISE DE DADOS FINANCEIROS



Trading com Dados

2ª Edição, agosto 2020

Sumário

Sobre nós e porque resolvemos criar esse e-book	3
Como acompanhar o conteúdo desse e-book?	4
O que é o mercado financeiro?.....	6
Por que aplicar a ciência de dados ao mercado financeiro?	6
E que dados conseguiremos analisar?	7
Desafios de lidar com dados de mercado financeiro	8
Como superar esse desafio?	9
Duas bibliotecas do R e do Python que vão nos ajudar bastante	10
1. Conceitos macroeconômicos	11
1.1 Inflação	13
1.2 Produto Interno Bruto (PIB)	14
1.3 Juros.....	15
1.3.1 Como Juros e Inflação se relacionam?	16
1.4 Moeda.....	18
1.5 Outros indicadores macroeconômicos que conseguimos obter	19
1.5.1 Ouro	19
1.5.2 Certificado de Depósito Interbancário (CDI).....	20
2. Mercado de Ações.....	21
2.1 Como essas variáveis se relacionam com o preço das ações?	26
Agradecimentos.....	28
Anexo A – Tutorial de instalação do Anaconda	29
1. Instruções iniciais.....	29
2. Configurações de instalação.....	30
3. Utilização do Jupyter	31
Anexo B – Tutorial de instalação do Quandl.....	33

Sobre o autor

Engenheiro Mecânico pela UFRN, mestre em Engenharia Industrial pela Northeastern University e MBA em Investimentos e Private Banking pelo Ibmecc, com passagem pelo Itaú e hoje é especialista em Data Science na XP Inc, onde atua especialmente em casos de Prevenção à Fraude e Compliance. Também ensina Data Science na FIA Business School e na Xpeed School.



Quem somos nós?

Somos uma iniciativa que nasceu com o propósito de disseminar os conhecimentos de ciência de dados e mercado financeiro. Assim, atuamos como uma ponte entre esses dois mundos, trazendo ao mercado financeiro o que há de mais avançado em machine learning e estatística, e levando à academia o pragmatismo e objetividade do mercado financeiro.

Porque resolvemos criar esse e-book?

Esse e-book surgiu justamente como resultado de workshops que desenvolvemos mostrando como o Python pode ser usado para analisar os dados do mercado financeiro. Percebemos que a programação aliada a conceitos simples de mercado financeiro fazia uma diferença enorme no aprendizado das pessoas, e por isso tivemos a iniciativa de consolidar isso num só e-book.

Quem é o público alvo desse e-book?

O e-book foi criado pensando nas pessoas que estão iniciando no mundo da programação ou Python, e querem entender como essas ferramentas podem ser usadas para obter e explorar dados de mercado financeiro.

Como acompanhar o conteúdo desse e-book?

Nesse e-book explicamos conceitos de mercado financeiro e macroeconomia de uma forma direta, leve, fácil de entender, e mostramos como a programação e a ciência de dados podem ser usadas para auxiliar o investidor na sua tomada de decisão.

Por isso, ao longo do livro, você vai ver os conceitos básicos sendo acompanhados de uma imagem mostrando como obter os dados correspondentes usando programação em Python.

Assim, se você quiser focar apenas nos conceitos, pode pular a seção que ensina a usar programação para fazer análise. Mas se você se interessa por essa metodologia, fique à vontade para explorar nossos tutoriais e testar as análises por conta própria.

Se você nunca programou em Python ou não possui familiaridade com a linguagem, recomendamos que veja o **Anexo A** desse e-book, contendo um tutorial de como instalar o Anaconda, a ferramenta mais difundida para programar em Python. Se preferir, criamos também um tutorial em vídeo que você pode acessar **[clcando aqui](#)**.

Mas se você já possui alguns conhecimentos em Python mas quer apenas aprender como usar o Quandl (ferramenta para conseguir acesso a dados financeiros no Python), recomendamos que você veja o Anexo B, contendo um tutorial para usar essa biblioteca do Python.

Não hesite em entrar em contato conosco durante o seu processo de aprendizado. Queremos ouvir dos leitores o que está legal e o que precisa ser melhorado. Ficamos felizes em receber dúvidas e feedbacks!

Para entrar em contato, [envie um e-mail](#) ou cadastre-se em nosso [canal do Telegram](#) para receber as novidades em primeira mão! Uma ótima leitura e até breve!



Equipe Trading com Dados

Nossas redes sociais:

Instagram – [instagram.com/tradingcomdados](https://www.instagram.com/tradingcomdados)

Twitter – twitter.com/tradingcomdados

Facebook – [facebook.com/somotradingcomdados](https://www.facebook.com/somotradingcomdados)

YouTube – [Trading com Dados](#)

Nosso site:

<https://www.tradingcomdados.com.br/>

O que é o mercado financeiro?

Mercado financeiro é, por definição, um ambiente de **compra e venda de ativos financeiros**, sejam eles valores mobiliários (ações, opções, títulos), câmbio (moedas estrangeiras) e mercadorias (ouro, produtos agrícolas).

Nessas negociações, estão envolvidas diversas instituições, que facilitam o encontro entre agentes e regulam e fiscalizam as transações.

No mercado financeiro, o investidor é aquele que dispõe de dinheiro sobrando e que deseja multiplicá-lo.

Os caminhos para isso são diversos, mas partem da mesma premissa: a verba é destinada a uma aplicação que oferece valorização de acordo com diretrizes acordadas entre as partes.

Por que aplicar a ciência de dados ao mercado financeiro?

Antes de começar nossas análises de dados de mercado financeiro, é essencial responder à pergunta acima. Isso porque acreditamos que uma análise só gera insights precisos e com algum impacto real, se estiverem conectadas a um propósito bem definido.

Ao explorar dados financeiros, passamos a entender melhor o comportamento dos nossos investimentos e dos ativos que estão disponíveis. Entendemos quais fatores podem estar por trás das quedas e subidas de algumas ações, como a inflação se relaciona com a renda fixa, etc. Dessa forma, conseguimos ter autonomia para criar as próprias análises para não depender de relatórios ou

análises pagas, e basear nossa própria tomada de decisão em dados.

Ao analisar dados de compra e venda de ativos, rentabilidade de ações, correlações entre papéis, dentre outros, conseguimos encontrar oportunidades que outras pessoas não veem. Portanto, se nosso objetivo é de melhorar nossos investimentos, conseguimos isso ao analisar métricas específicas, como retorno e risco, por exemplo.

Por fim, ao aprender as técnicas que apresentamos aqui, você se torna capaz de usar novas técnicas de programação aliadas a conhecimento de negócio que pode ser útil numa movimentação no mercado de trabalho!

E que dados conseguiremos analisar?

Basicamente, qualquer dado que esteja disponível online e que você consiga transformar em informação de valor no mercado financeiro. De forma mais detalhada:

- Indicadores macroeconômicos, como inflação, juros, PIB, taxa de desemprego; dados de negociação da bolsa, como preço de abertura, fechamento, máximas, mínimas;
- Informações sobre o desempenho econômico das empresas, como Receita, EBITDA, ROE, Lucro líquido;
- Indicadores baseados em análise técnica, como médias móveis, bandas de Bollinger, além de outras figuras gráficas;
- Sentimentos de notícias, redes sociais, sites de órgãos reguladores, usando técnicas como Processamento de Linguagem Natural, Text Mining, etc.

Desafios de lidar com dados de mercado financeiro

Depois de descobrir onde os dados estão, basta agora fazer download para obter todos eles, correto? Infelizmente não. Lidar com dados financeiros é complicado não apenas pela dificuldade de consegui-los, mas também de tratá-los e deixá-los de maneira que seja adequada para fazer nossos modelos.

Alguns desses dados não estão disponíveis gratuitamente na internet, como os dados contábeis consolidados de empresas ou mesmo dados de negociação em tempo real. É necessário pagar valores consideráveis em provedores especializados para consegui-los. Além disso, alguns outros dados, como os de textos em notícias ou redes sociais, são desestruturados, desorganizados, o que torna difícil o seu acesso e o posterior tratamento.

Por fim, temos que reconhecer ainda uma outra dificuldade do ponto de vista técnico. Mesmo que tenhamos os dados de negociação, eles são difíceis de serem modelados. Isso porque eles violam algumas premissas estatísticas que poderia impossibilitar nossos modelos – não são estacionários nem independentes (não se preocupe em saber o que isso quer dizer agora, vamos detalhar isso em outros materiais). Além disso, os fenômenos são difíceis de capturar do ponto de vista de modelagem, ou seja, exigem um volume de dados externos que torna praticamente impossível a análise pela pessoa física.

Como superar esse desafio?

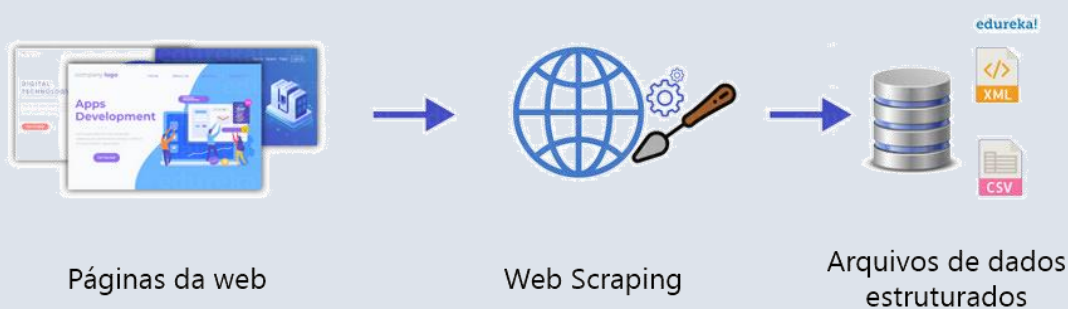
Dado todo esse cenário de dificuldades, você deve estar se perguntando:

Desse jeito eu não vou conseguir analisar dados de mercado financeiro, certo?

Errado!

Conseguimos sim, através de algumas ferramentas, contornar esses obstáculos e fazer ótimas análises para ajudar a tomada de decisão do investidor.

Com relação aos dados difíceis de serem obtidos, como textos de notícias e redes sociais como comentamos no post anterior, podemos criar nossos próprios algoritmos de “web scraping” para capturar essas informações de diferentes sites, inclusive dados fundamentalistas e contábeis de diferentes empresas.



“Web Scraping” é uma técnica que nos permite obter dados de sites da internet em um formato estruturado, apropriado para análises

Usando linguagens de programação com Python e R, podemos usar essas técnicas de capturas de dados para importar dados de negociação da bolsa diretamente para suas análises. Uma outra maneira de fazer isso é usar as chamadas **bibliotecas** dessas linguagens que permitem a obtenção direta desses dados sem nem precisar abrir qualquer site. Neste e-book vamos destacar as bibliotecas BatchGetSymbols do R e a Quandl do Python, que vão ser explicadas na próxima seção.

Duas bibliotecas do R e do Python que vão nos ajudar bastante

Como falamos na seção anterior, existem duas bibliotecas, uma na linguagem R e a outra no Python, que nos ajudam muito na hora de obter dados de mercado financeiro:



BatchGetSymbols

- Criada pelo brasileiro Marcelo Perlin
- Permite fácil acesso a dados das principais bolsas do mundo
- Não requer cadastro, basta chamar direto através do código



Quandl

- Criada pela empresa americana Quandl
- Além de permitir acesso a dados de bolsas, possui também dados macroeconômicos (como inflação e juros) de vários países
- Requer cadastro na plataforma da Quandl, que fornece um token de acesso

As duas bibliotecas conseguem obter virtualmente dados de qualquer ação brasileira, e de outros países também. A `BatchGetSymbols` não requer cadastro, basta apenas chamá-la diretamente da interface do próprio R.

Já a `Quandl`, exige um cadastro prévio na plataforma da própria `Quandl` onde você deve obter uma chave de acesso. Instruções de como fazer isso estão no **Anexo B** deste e-book.

Com a biblioteca `Quandl` você consegue além dos dados de ações, obter também dados macroeconômicos como juros, PIB, inflação, dólar, que serão objetos de análise nesse livro, e daí a razão por usarmos ela aqui nas nossas análises. Acreditamos que a biblioteca `Quandl` é uma ótima forma de começar a explorar esse mundo dos dados financeiros!

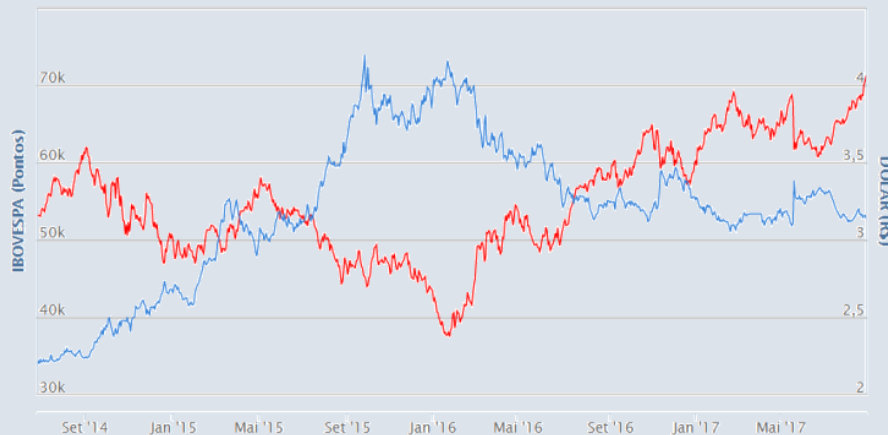
1. Conceitos macroeconômicos

Neste livro faremos uma abordagem introdutória sobre o assunto macroeconomia. Não abordaremos todos os tópicos desse assunto, mas apenas aqueles que acreditamos que terão um impacto nas nossas análises mais prioritárias. Nosso foco aqui é te dar uma noção do que você consegue fazer com o Python (principalmente usando a `Quandl`) e familiarizar com alguns conceitos de macroeconomia. Depois de apresentar todos os conceitos e como eles se relacionam, vamos mostrar alguns códigos onde você vai ter a possibilidade de desenvolver análises para si mesmo, de forma autônoma.

A macroeconomia é o estudo da economia como um todo, utilizando indicadores agregados que medem a produção de bens e serviços, o nível geral de emprego e o nível de preços.

Alguns exemplos de como indicadores macro podem estar correlacionados. No gráfico a seguir, você vê o IBOV em vermelho e o preço do Dólar em azul.

Vamos explicar depois o conceito de IBOV, mas por hora você só precisa saber que é um índice que representa o desempenho da bolsa de valores. Observe que eles têm um comportamento inverso. Quando o dólar cai, a bolsa sobe.



1.1 Inflação

Vamos começar nosso estudo com o conceito de inflação. É um conceito fundamental, que está tão presente em nossas vidas que não poderia ser deixado de fora em nenhuma análise.

A inflação é um fenômeno macroeconômico, dinâmico e de natureza monetária, caracterizado por uma elevação apreciável e persistente do nível geral de preços, ou seja, um aumento contínuo no nível geral de preços da economia. A taxa de inflação é a taxa na qual o nível de preços sobe. Da mesma forma, a deflação é uma redução contínua no nível de preços, ou uma inflação negativa.

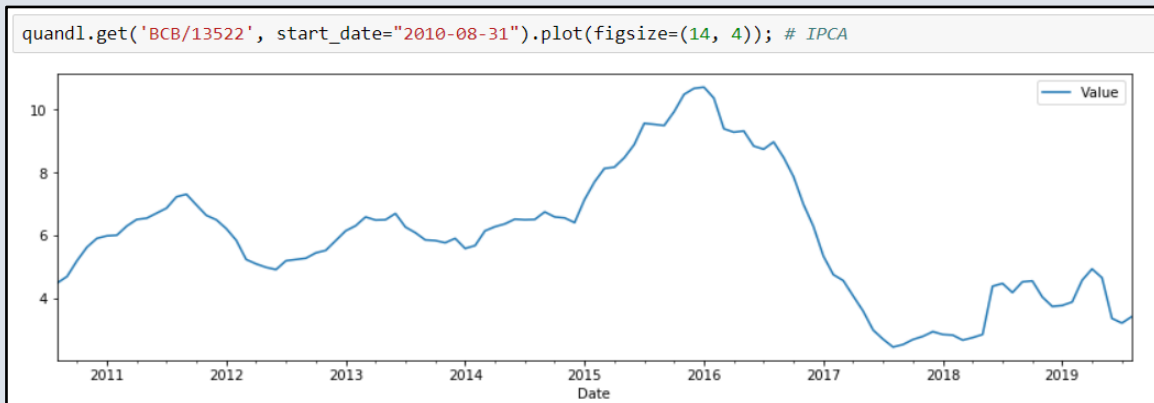
Segundo o Banco Central do Brasil,

“No Brasil, não há um índice oficial para inflação de períodos passados. A inflação é medida por meio de diversos índices, divulgados por várias instituições, tais como o Instituto Brasileiro de Geografia e Estatística (IBGE), a Fundação Getúlio Vargas (FGV), a Fundação Instituto de Pesquisas Econômicas (FIPE).”

Portanto, o Brasil não possui um indicador oficial de inflação, sendo os principais, o IPCA, o IGPM e o INPC. Já que o IPCA é o mais difundido, vamos analisa-lo e obter sua variação desde 2010 usando o Quandl.

Para obtê-la com a Quandl repare que usamos o código “BCB/13522”. Isso porque esses dados estão no repositório do Banco Central do Brasil, daí o “BCB” e tem código de referência 13522. Perceba que essa mesma estrutura de código será usada nas requisições posteriores.

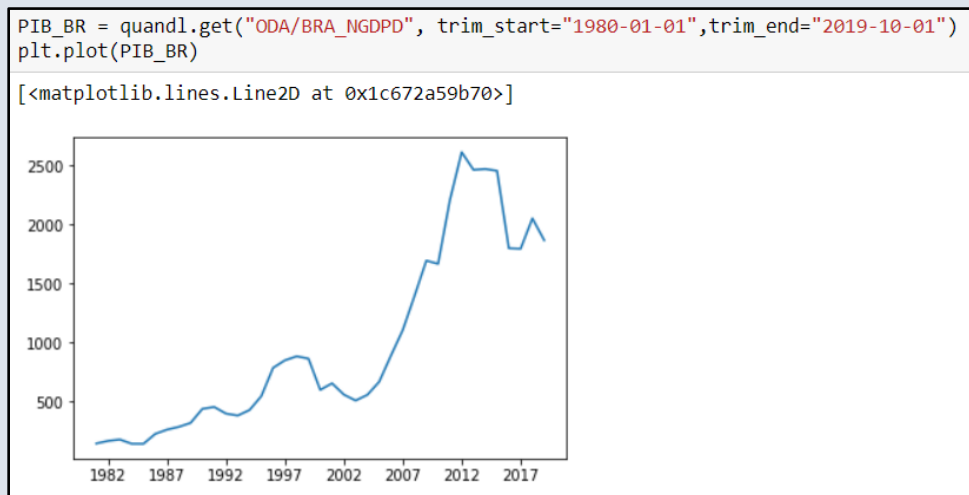
O gráfico juntamente com o código usado para obtê-lo é mostrado a seguir:



Vemos que a inflação no Brasil atingiu o nível máximo para o período no final de 2015 e em seguida iniciou uma queda forte, até ter mais ou menos estabilizado até o presente momento. Vamos agora para um outro conceito importante, o PIB.

1.2 Produto Interno Bruto (PIB)

PIB é o valor final (soma em valores monetários) de todos os bens e serviços produzidos pelas empresas que estão dentro das fronteiras de um território, em um dado período. Também conseguimos obter os dados históricos de PIB com a Quandl. O código para conseguir isso é mostrado a seguir, juntamente com o gráfico. Aqui, obtivemos os dados desde o início da década de 80:

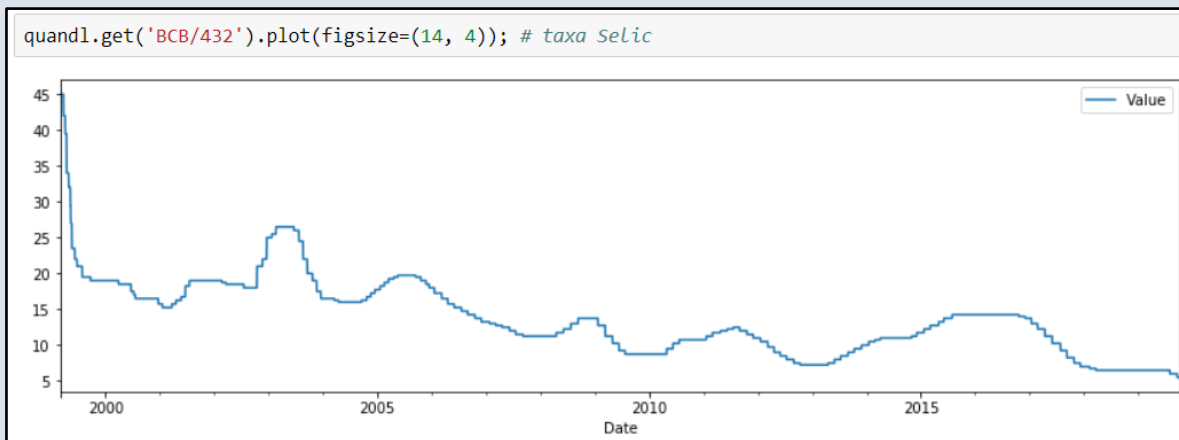


Vemos que o PIB brasileiro teve um crescimento acentuado do início dos anos 2000 até o início dos anos 2010, quando entrou em estagnação e depois teve uma queda forte.

1.3 Juros

Juros nada mais é que um rendimento (prêmio) pago para aquele que deixa de usufruir do valor monetário no momento para recebê-lo de volta no futuro. Ou seja, juros é o valor do dinheiro no tempo. Em outras palavras, juros é aquilo que valoriza o dinheiro ao longo do tempo quando você, por exemplo, aplica tal dinheiro em algum tipo de instrumento financeiro (poupança, empréstimos, etc.). No Brasil, a taxa SELIC, determinada pelo Banco Central, é utilizada como referência para o mercado na determinação dos juros das suas próprias operações.

Para obter os valores da SELIC desde o final da década de 1990 usando a biblioteca Quandl, devemos usar o código abaixo. Acompanhe o gráfico resultante:



Existe uma tendência clara da redução da taxa de juros básica (SELIC) do Brasil ao longo do tempo, com algumas reviravoltas ao longo do caminho. Em 2015, por exemplo, a SELIC alcançou patamares superiores a 12%, chegando a alcançar 14,25% no final de 2015. Do final de 2016 até o presente momento a taxa vem caindo consistentemente.

1.3.1 Como Juros e Inflação se relacionam?

Já sabemos como obter dados de inflação (IPCA) e juros (SELIC) do Brasil. E se quisermos comparar os dois ao longo do tempo para tirar algumas conclusões sobre o comportamento da economia brasileira nos últimos anos? A figura abaixo traz o código que usamos para plotar o IPCA e a SELIC no mesmo gráfico. Se você ainda não tem familiaridade com o Python, pode focar apenas no gráfico resultante!


```

fig, ax1 = plt.subplots()

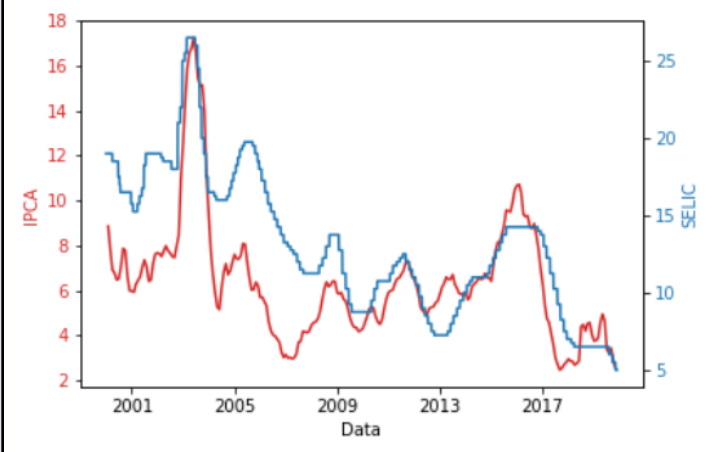
color = 'tab:red'
ax1.set_xlabel('Data')
ax1.set_ylabel('IPCA', color=color)
ax1.plot(ipca, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'tab:blue'
ax2.set_ylabel('SELIC', color=color)
ax2.plot(selic, color=color)
ax2.tick_params(axis='y', labelcolor=color)

plt.show()

```



Veja como a SELIC se aproxima da inflação! E não é à toa. A taxa básica de juros é determinada muitas vezes de forma reativa à inflação, de forma a controla-la. Quando a inflação sobe muito, aumentar a SELIC é uma forma de segurar o acesso ao crédito, que por consequência, tende a segurar o aumento geral nos preços.

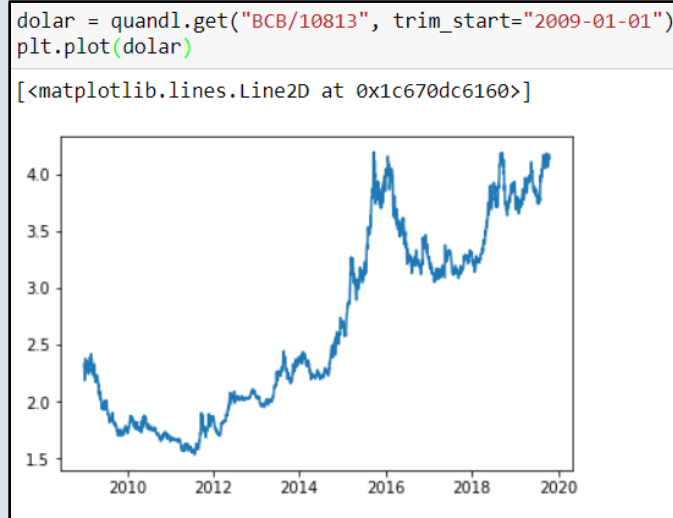
1.4 Moeda

É o meio pelo qual são efetuadas transações financeiras. Moedas precisam ter três características:

- Meio de troca: ou seja, é utilizada como meio de pagamento para executar transações financeiras;
- Unidade de conta: quando ela é usada como referência para atribuir valor a mercadorias;
- Reserva de valor: capacidade da moeda de reter para si o valor ao longo do tempo, mantendo seu poder de compra ou sendo capaz de medir a riqueza.

Podemos considerar como *moeda* o papel moeda que temos no bolso, os depósitos à vista em nossas contas correntes no banco, nossas reservas na poupança, ou aplicadas em títulos públicos, CDBs, etc.

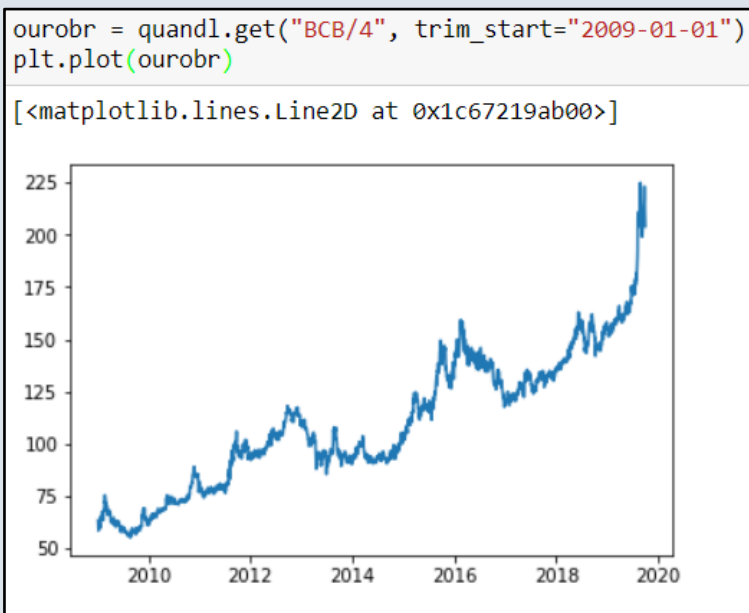
Para demonstrar o uso da biblioteca Quandl, vamos obter aqui os dados da moeda dólar americano com a sua cotação em reais, desde o ano de 2009:



1.5 Outros indicadores macroeconômicos que conseguimos obter

Conseguimos obter uma série de outros indicadores com a Quandl, e achamos interessante mostrar aqui outros dois, que também podem ser relevantes para análises:

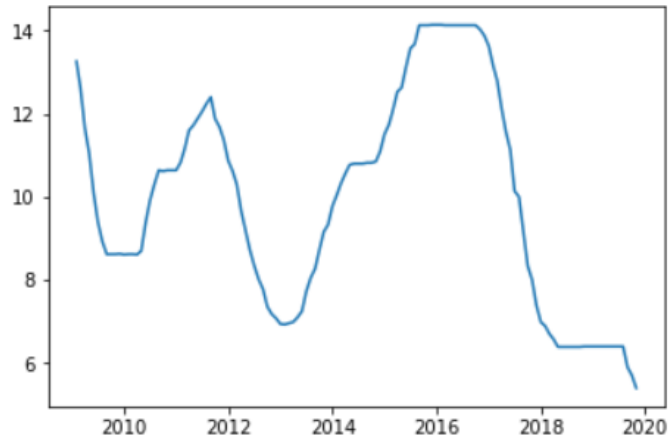
1.5.1 Ouro



1.5.2 Certificado de Depósito Interbancário (CDI)

```
cdi = quandl.get("BCB/4392", trim_start="2009-01-01")  
plt.plot(cdi)
```

[<matplotlib.lines.Line2D at 0x1c672af0898>]



1.5.3 Petróleo WTI

Observe que também podemos extrair dados de commodities usando a Quandl!

```
wti_crude = quandl.get("EIA/PET_RWTC_D", start_date = "2010-12-01", end_date = "2020-04-01")  
plt.plot(wti_crude)  
plt.show()
```



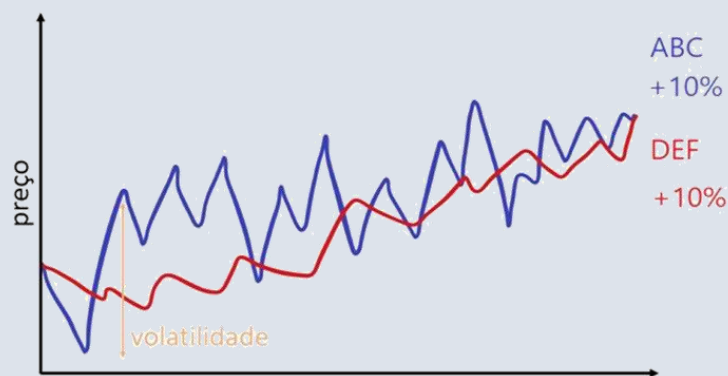
2. Mercado de Ações

O mercado de ações é onde se negocia títulos, estes por sua vez dão direito a uma parte da propriedade de uma companhia.

Esse mercado é importante, pois ele serve como viabilizador para levantamento de recursos pela companhia, e isso é feito através da emissão de ações.

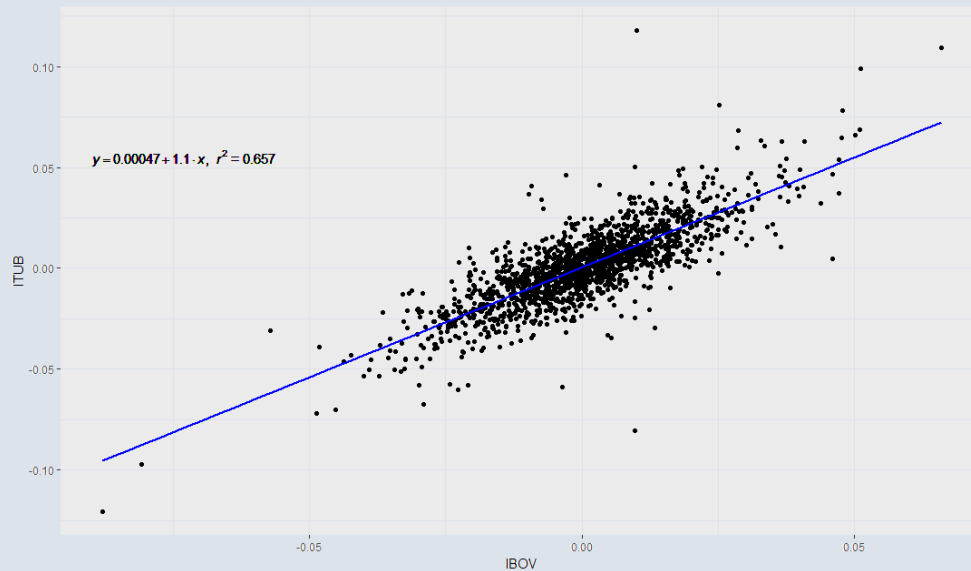
Para entender melhor o que influencia o preço de uma ação e tentar prevê-lo, é importante acima de tudo entender conceitos que são relacionados a elas, como retorno e volatilidade.

Imagine que temos duas ações, ABC e DEF, como mostradas no gráfico à esquerda abaixo. Veja que ao final do período, as duas ações renderam os mesmos 10%, mas a ação ABC enfrentou um percurso muito mais turbulento, oscilou muito mais que a ação DEF. Ou seja, o retorno das duas ações foi o mesmo, mas a ação ABC foi muito mais volátil e apresentou um risco maior.



Uma vez que já sabemos o que são retorno e volatilidade, podemos partir para análises mais complexas. Imagine que pegássemos os retornos das ações do Itaú (ITUB4) e os retornos do IBOV ao longo de um período e plotássemos essas duas informações em um

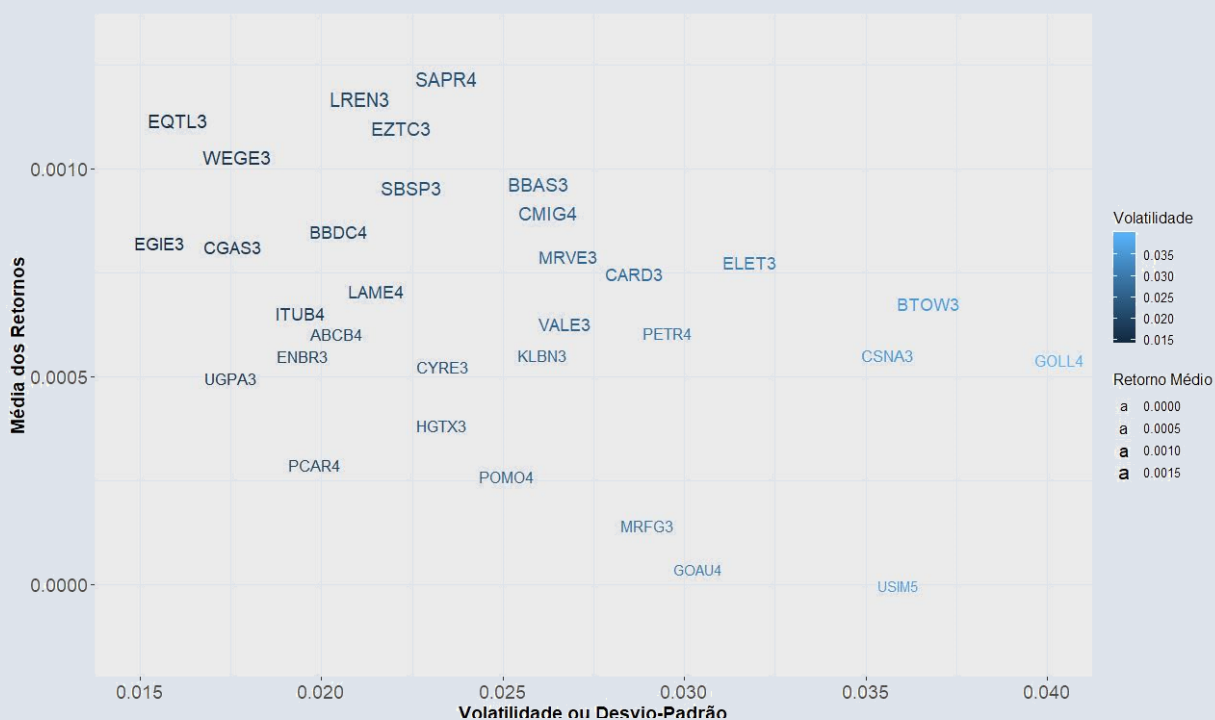
gráfico? O resultado seria semelhante ao do gráfico a seguir, feito usando o R.



Perceba que os retornos de ITUB4 e IBOV são bem próximos. Isso é, nos dias que IBOV subiu, as ações do Itaú provavelmente subiram, e vice-versa. Isso é um indício que elas estão diretamente relacionadas. Além de uma alta correlação, note também que os patamares de retorno são próximos, ou seja, quando Itaú sobe muito, IBOV sobe muito, quando Itaú sobe pouco, IBOV sobe pouco. Isso acontece porque ITUB4 é dita uma ação **reativa** ao IBOV. Quanto mais reativa for uma ação, mais intensa é a sua oscilação frente à do IBOV naquele mesmo período.

Vamos agora introduzir o conceito de volatilidade na nossa análise. Nas análises de ações, normalmente analisamos a volatilidade como sendo o **desvio-padrão** dos retornos daquela ação. Imagine

que conseguíssemos o retorno e a volatilidade de várias ações e fizéssemos um gráfico com elas. O resultado é mostrado a seguir.



Quanto mais à esquerda no gráfico estiver a ação, menor a sua volatilidade, ou seja, menores são as suas oscilações ao longo do tempo. Quanto mais acima elas estiverem, maiores foram os retornos nesse mesmo período. O que isso quer dizer? Que ações como EQTL3, WEGE3, EGIE3, LREN3, por exemplo, possuíram altos patamares de retorno e baixos patamares de volatilidade (ou de risco) para o período em que foram analisadas. Obviamente, essa análise depende muito do período em que foi feita. Se você fizesse em intervalos maiores ou menores de tempo poderia obter resultados diferentes.

Esse gráfico também foi feito usando a linguagem R, e se você tem curiosidade de saber como obtê-lo, pode checar o artigo no nosso site explicando o passo a passo. [Basta clicar aqui!](#)

A forma mais fácil de obter dados de ações com o Python é através da biblioteca **pandas_datareader**.

Acompanhe a demonstração:

```
from pandas_datareader import data

ibov = data.DataReader('^BVSP', data_source = 'yahoo')

ibov.head()
```

Date	High	Low	Open	Close	Volume	Adj Close
2015-04-20	54385.0	53510.0	53965.0	53761.0	2688000.0	53761.0
2015-04-22	54848.0	53762.0	53762.0	54617.0	3700600.0	54617.0
2015-04-23	55782.0	54064.0	54616.0	55685.0	5535100.0	55685.0
2015-04-24	56965.0	55687.0	55687.0	56594.0	4982000.0	56594.0
2015-04-27	56962.0	55515.0	56554.0	55535.0	3854300.0	55535.0

```
ibov = data.DataReader('^BVSP', data_source = 'yahoo', start = '2010-01-01')
ibov.rename(columns = {"Adj Close": "IBOV"}, inplace = True)

ibov
```

Date	High	Low	Open	Close	Volume	IBOV
2010-01-04	70081.0	68587.0	68587.0	70045.0	1655400.0	70045.0
2010-01-05	70595.0	69928.0	70046.0	70240.0	1984200.0	70240.0
2010-01-06	70937.0	70016.0	70237.0	70729.0	2243600.0	70729.0
2010-01-07	70723.0	70045.0	70723.0	70451.0	1555000.0	70451.0
2010-01-08	70766.0	70158.0	70455.0	70263.0	1634400.0	70263.0

Veja que em poucas linhas de código fomos capazes de capturar as cotações do IBOV.

O processo é parecido caso queiramos buscar dados de ações:

```
petr = data.DataReader('PETR4.SA', data_source = 'yahoo', start = '2000-01-01')
```

petr						
Date	High	Low	Open	Close	Volume	Adj Close
2000-01-03	5.875000	5.875000	5.875000	5.875000	3.538944e+10	4.050402
2000-01-04	5.550000	5.550000	5.550000	5.550000	2.886144e+10	3.826338
2000-01-05	5.494000	5.494000	5.494000	5.494000	4.303360e+10	3.787730
2000-01-06	5.475000	5.475000	5.475000	5.475000	3.405568e+10	3.774631
2000-01-07	5.500000	5.500000	5.500000	5.500000	2.091264e+10	3.791866

E se quiséssemos plotar dois índices no mesmo gráfico? Veja abaixo como mostrar as cotações de IBOV e S&P 500 no mesmo plot:

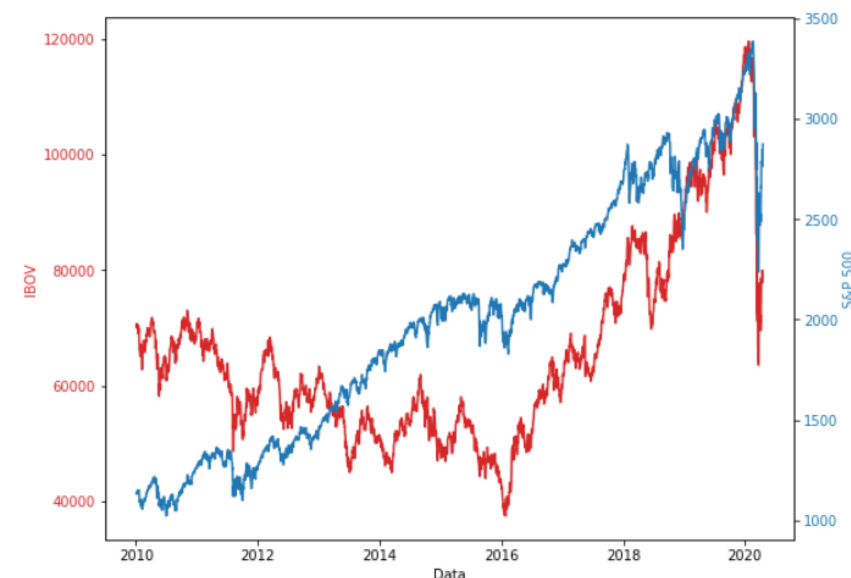
```
fig, ax1 = plt.subplots(figsize=(9,7))

color = 'tab:red'
ax1.set_xlabel('Data')
ax1.set_ylabel('IBOV', color=color)
ax1.plot(ibov, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx() # Configurar um outro eixo vertical que compartilha o mesmo eixo X

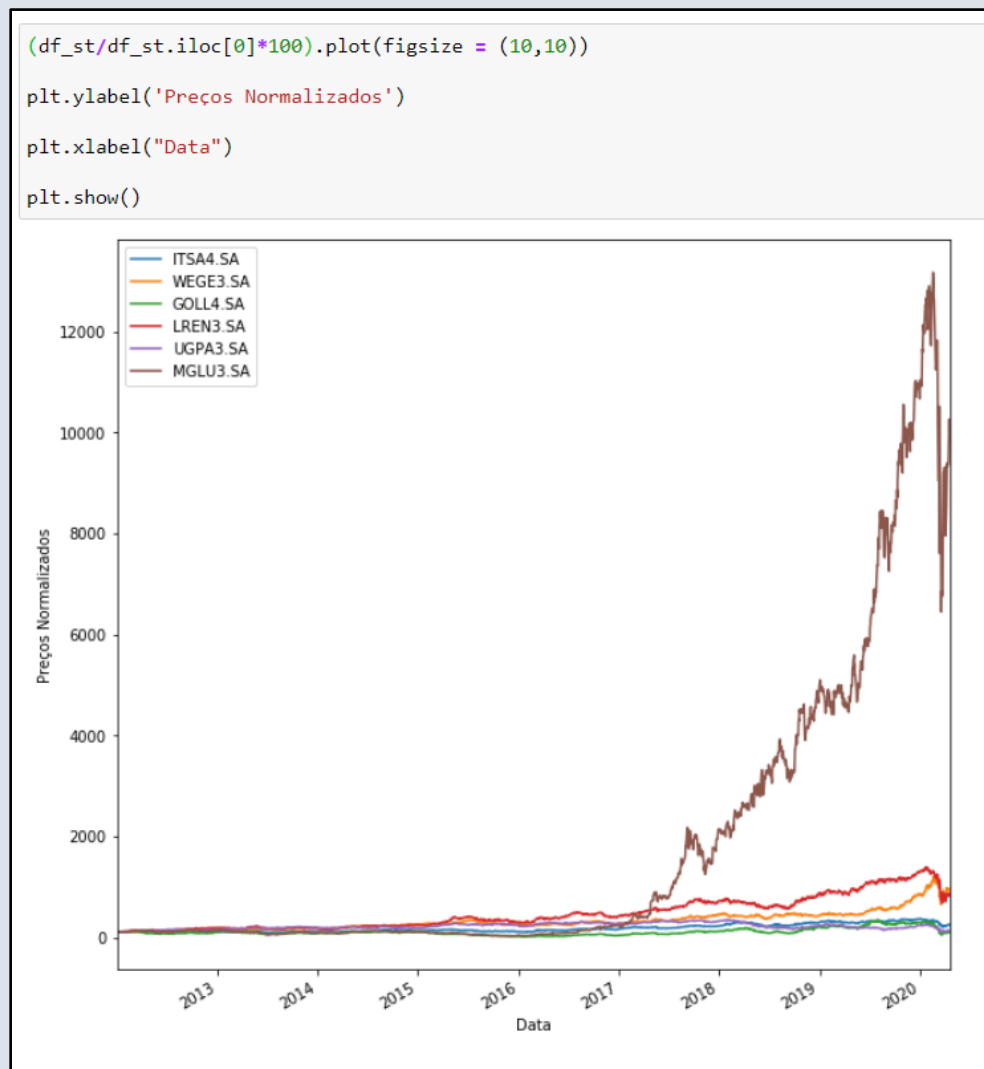
color = 'tab:blue'
ax2.set_ylabel('S&P 500', color=color)
ax2.plot(sp500, color=color)
ax2.tick_params(axis='y', labelcolor=color)

plt.show()
```



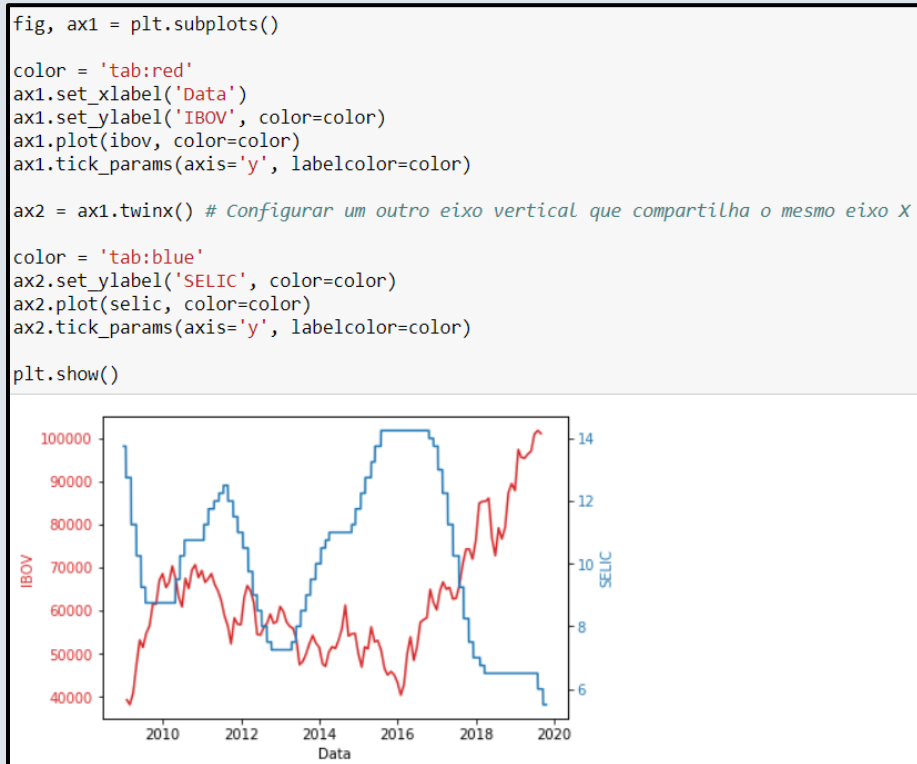
Interessante notar que esses dois indicadores passaram a maior parte do período tendo pequena ou quase nenhuma correlação. No entanto, num cenário de alta volatilidade e risco nos mercados globais, em meados de março de 2020 por causa da pandemia, esses índices passaram a ter correlações elevadas, caindo bruscamente. Essa tendência de aumento da correlação repentina de ativos ao redor do mundo se observa normalmente em cenários de fenômenos atípicos, como crises, por exemplo.

Para terminar nossa demonstração com ações, veja como fica nosso plot de vários papéis quando normalizamos seu preço frente a uma mesma data de início:



2.1 Como essas variáveis se relacionam com o preço das ações?

Imagine que por alguma razão queremos determinar se há uma correlação entre o IBOV e a taxa básica de juros do Brasil, a SELIC. Com algumas linhas de código conseguimos ver num só gráfico como os dois indicadores variaram!



O que acha de tentar por conta própria fazer um gráfico desses usando alguma outra ação de sua preferência? Nossa sugestão é começar com ações que são reativas ao IBOV, como explicamos anteriormente! O resultado deve ficar parecido com esse que obtivemos.

Agradecimentos

Gostaríamos de agradecer aos alunos de todas as universidades onde apresentamos esse nosso workshop e que nos deram as bases para escrever o nosso primeiro e-book, em especial o pessoal da Unicamp, do IBMEC, do IME-USP e da FMU. Um grande abraço!

Agradecemos também a sua leitura e o interesse nesse tópico que é uma verdadeira paixão para todos nós na Trading com Dados. Ficaríamos muito felizes se você pudesse nos enviar sua opinião! Gostou do que leu? Acha que faltou algum conteúdo? Não deixe de falar com a gente! [Mande um e-mail](#) ou converse com a gente em nossas redes sociais!



Equipe Trading com Dados

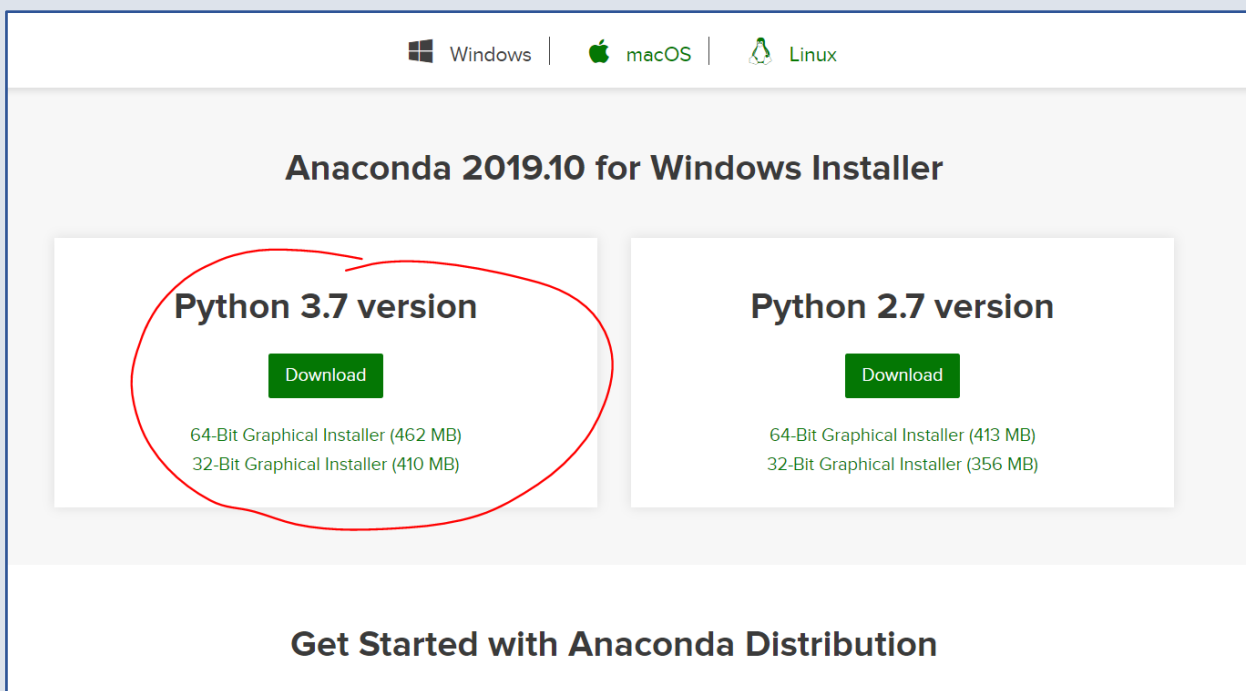
Anexo A – Tutorial de instalação do Anaconda

1. Instruções iniciais

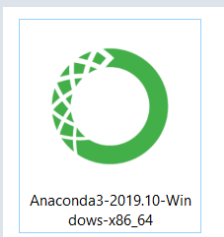
Acesse o link abaixo:

<https://www.anaconda.com/distribution/#download-section>

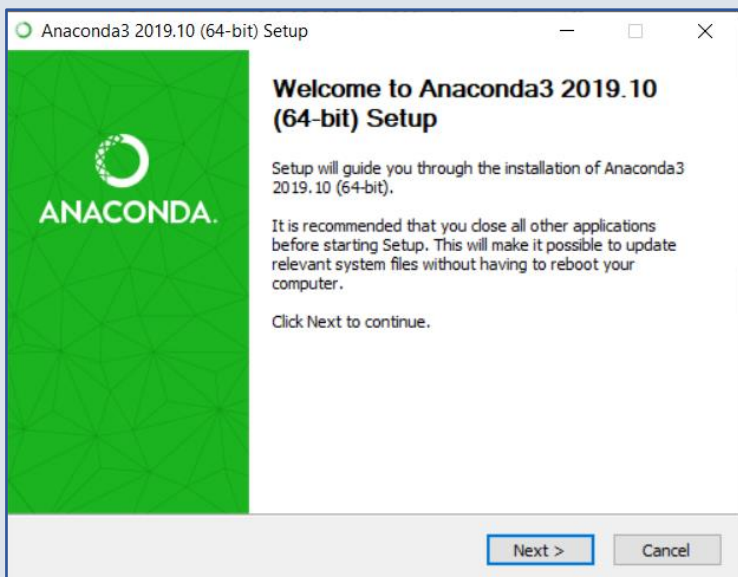
Selecione a versão 3.7:



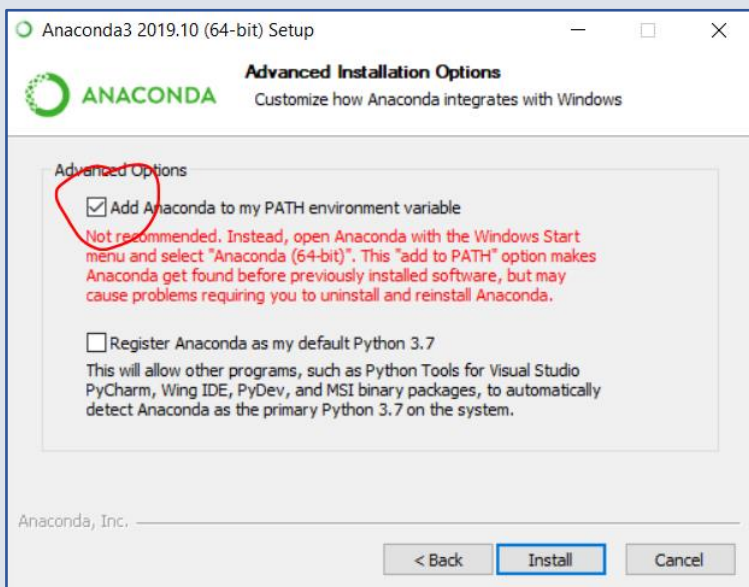
Clique no ícone para iniciar a instalação



2. Configurações de instalação



Avance pelas etapas até ser perguntado sobre adicionar ou não o Anaconda ao PATH. Marque essa opção, como mostrado na imagem abaixo:



3. Utilização do Jupyter

Após a instalação, você verá o seguinte ícone na sua área de trabalho:

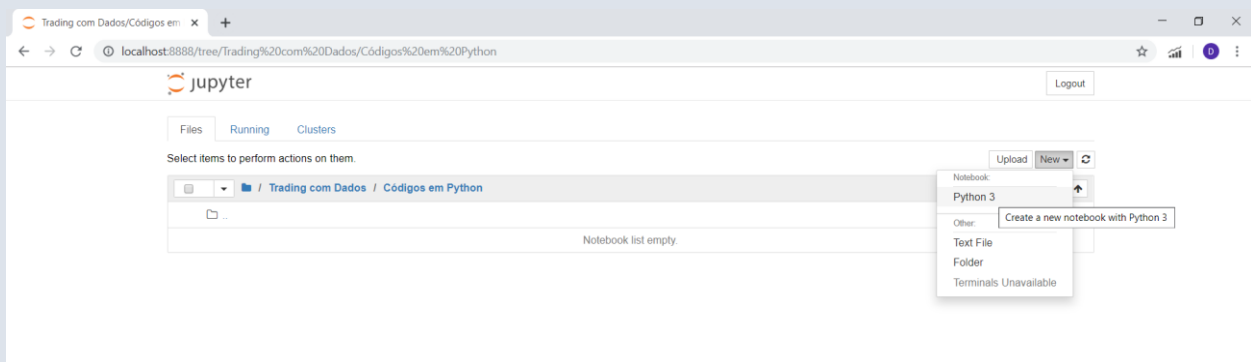


Caso a tela com o Jupyter não seja iniciada automaticamente no seu navegador, copie o código abaixo e cole no navegador de sua preferência:

```
Select C:\ProgramData\Anaconda3\python.exe
[I 01:45:20.085 NotebookApp] JupyterLab alpha preview extension loaded from C:\ProgramData\Anaconda3\lib\site-packages\jupyterlab
JupyterLab v0.27.0
Known labextensions:
[I 01:45:20.149 NotebookApp] Running the core application with no additional extensions or settings
[I 01:45:20.359 NotebookApp] Serving notebooks from local directory: C:\Users\vinici\
[I 01:45:20.359 NotebookApp] 0 active kernels
[I 01:45:20.359 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=bd21ccce403aee57f48fb84af12cb1b2fe918448ae6c203c
[I 01:45:20.359 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 01:45:20.380 NotebookApp]

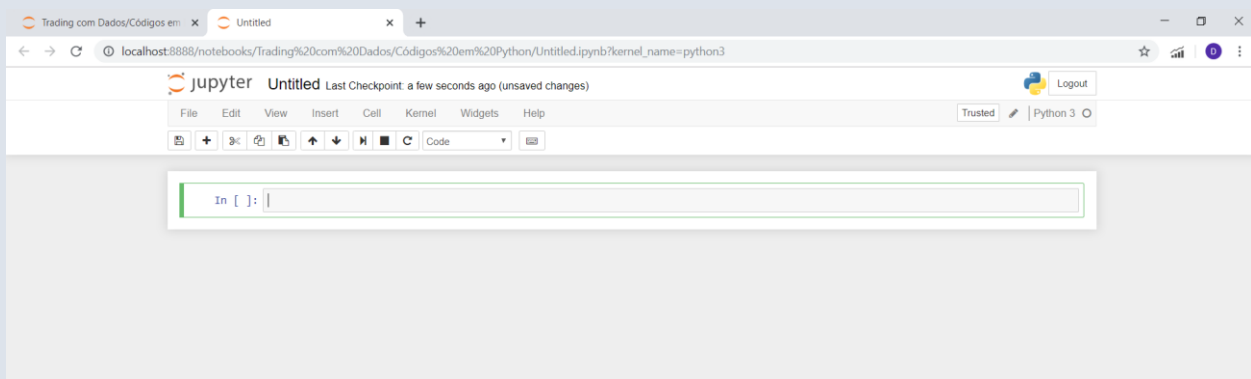
Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=bd21ccce403aee57f48fb84af12cb1b2fe918448ae6c203c
[I 01:45:20.745 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

Você será levado a uma tela semelhante a essa:

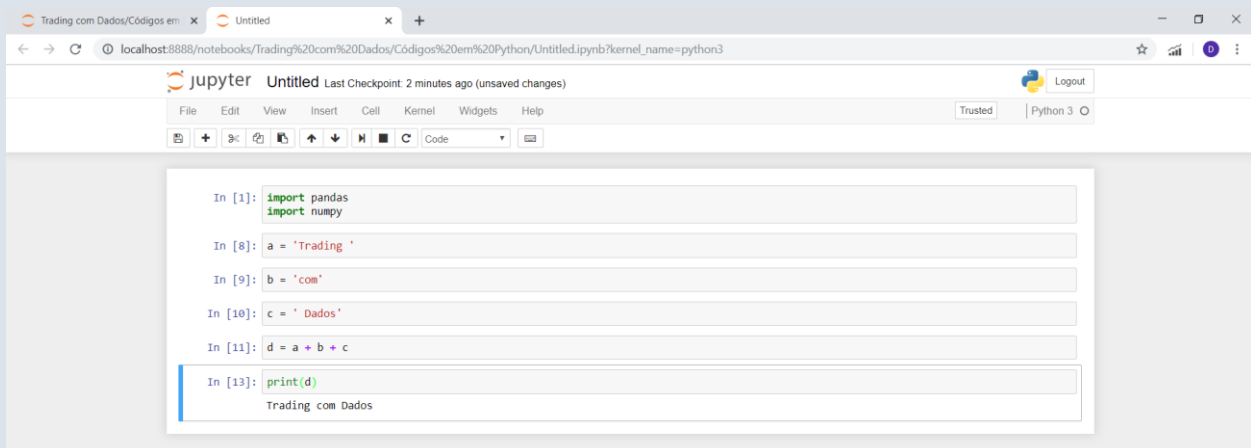


Você pode navegar pelas pastas no console até encontrar uma pasta onde deseja criar os seus códigos. Quando estiver na pasta desejada, clique no botão “New” no canto superior direito e em seguida “Python 3” para criar seu primeiro notebook.

Pronto! Agora você já pode programar em Python e testar!



Comece testando algumas bibliotecas básicas e funções simples e vá aos poucos avançando em comandos mais complexos.



Anexo B – Tutorial de instalação do Quandl

O Quandl é uma das plataformas de dados financeiros mais utilizadas no mundo. Por meio de uma única API gratuita você consegue obter milhões de conjuntos de dados financeiros e econômicos de centenas de editores.

a. Instalação

Acesse o link abaixo para as instruções de como baixar a biblioteca Quandl no Python

<https://docs.quandl.com/docs/python-installation>

INSTALLATION & AUTHENTICATION

TABLE OF CONTENTS

INSTALLATION

AUTHENTICATION

INSTALLATION

You can download the Quandl Python package from [PyPI](#) or from [GitHub](#). Follow the installation instructions below.

NOTE: Installation of the Quandl Python package varies depending on your system.

On most systems, the following commands will initiate installation:

Python

```
pip install quandl
import quandl
```

On some systems, you may need this command instead:

Python

```
pip3 install quandl
import quandl
```

Additionally, you can find detailed installation instructions for Python modules here: [Python 3.x](#) and [Python 2.7x](#).

AUTHENTICATION

The Quandl Python module is free but you must have a Quandl API key in order to download data. To get your own API key, you will need to create a free [Quandl account](#) and set your API key.

After importing the Quandl module, you can set your API key with the following command:

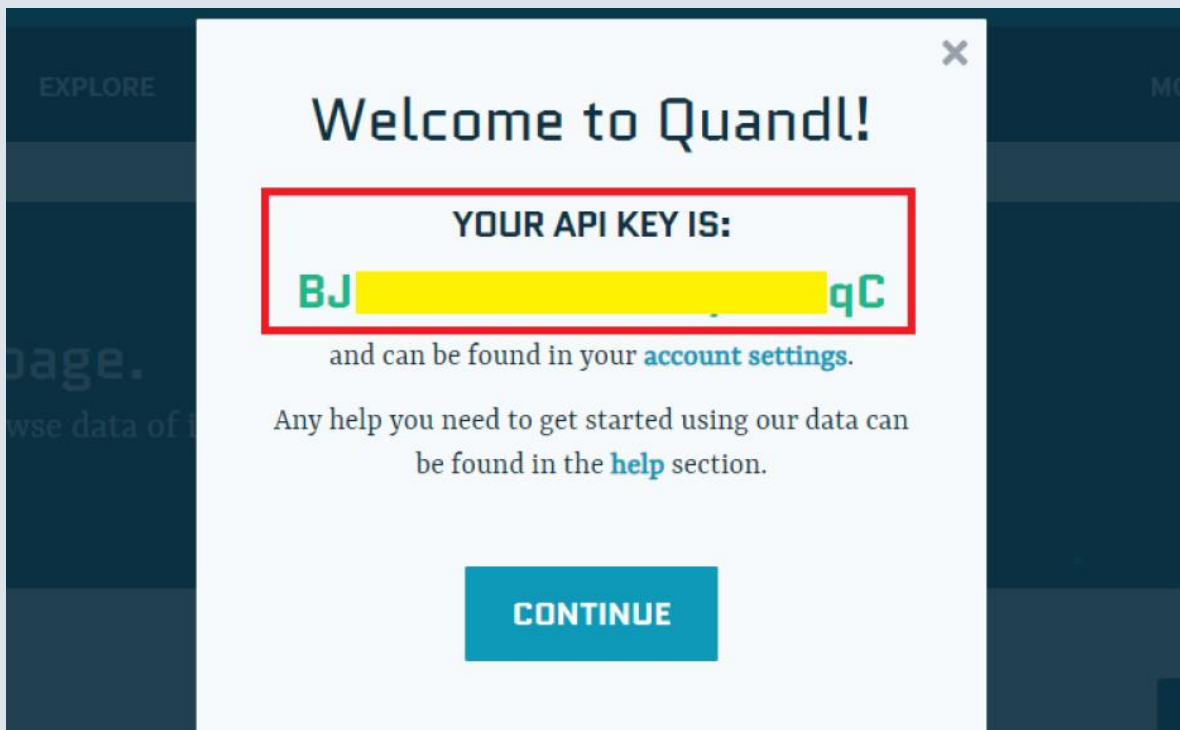
```
quandl.ApiConfig.api_key = "YOURAPIKEY"
```

b. Cadastro para obtenção de API_KEY

Para obter os dados teremos que fazer um cadastro simples, ao final obteremos uma API_KEY que será usada para fazer as requisições. Para isso, acesse o seguinte link:

<https://www.quandl.com/sign-up-modal?defaultModal=showSignUp>

Obs: Após o cadastro é necessário fazer o processo de confirmação de e-mail.



c. Obtendo os dados

Para importar a biblioteca e adicionar a chave da API, siga os seguintes passos com o Python:

```
In [10]: import quandl
          quandl.ApiConfig.api_key = "*****"
```

Onde você precisa substituir os asteriscos com a sua API_KEY.

Para obter os dados de "Time Series", siga a documentação abaixo:

<https://docs.quandl.com/docs/python-time-series>

Um exemplo: vamos criar uma chamada que recebe o WTI Crude Oil Price, que tem como código Quandl EIA/PET_RWTC_D do Departamento de Energia dos EUA:

```
In [18]: data = quandl.get("EIA/PET_RWTC_D")
```

```
In [19]: data.head()
```

Out[19]:

	Value
Date	
1986-01-02	25.56
1986-01-03	26.00
1986-01-06	26.53
1986-01-07	25.85
1986-01-08	25.87

Obtendo os dados de "Tables"

A API de tabelas é limitada a 10.000 linhas por chamada. No entanto, ao usar a biblioteca Python, anexar o argumento `paginate=True` estenderá o limite para 1.000.000 de linhas. Como tal, recomendamos o uso `paginate=True` para todas as chamadas. Observe que alguns conjuntos de dados podem retornar mais dados do que o Python permite. Se isso ocorrer, será necessário filtrar ainda mais sua chamada para baixar menos dados, conforme descrito nos exemplos abaixo. Ou você pode considerar usar o exportador de dados de Quandl.

Para a documentação completa, acesse:

<https://docs.quandl.com/docs/python-tables#section-example-1>

Alguns exemplos:

In [24]: *### Filtrar Linhas*

```
data = quandl.get_table('MER/F1', compnumber="39102", paginate=True)
data.head(5)
```

Out[24]:

	compnumber	reportid	mapcode	amount	reportdate	reporttype	auditorstatus	currency	consc
--	------------	----------	---------	--------	------------	------------	---------------	----------	-------

0	39102	1851368960	-9087	10.446851	2010-12-31	A	N	EUR	
1	39102	1851369024	-9087	10.446851	2010-12-31	Q4	N	EUR	
2	39102	1885110592	-1465	0.642525	2012-12-31	Q4	N	EUR	
3	39102	1885110592	-5370	8.596581	2012-12-31	Q4	N	EUR	
4	39102	1851368960	-1012	3.392546	2010-12-31	A	N	EUR	

5 rows × 32 columns

In [26]: *### Filtrar Colunas*

```
data = quandl.get_table('MER/F1', qopts={"columns": "compnumber"}, paginate=True)
data.head()
```

Out[26]:

	compnumber
0	2438
1	2438
2	2438
3	2438
4	2438

```
In [27]: ### Filtrar Linhas e Colunas  
  
data = quandl.get_table('MER/F1', compnumber="39102", qopts={"columns": "reportdate"}, paginate=True)  
data.head()
```

Out[27]:

	reportdate
0	2010-12-31
1	2010-12-31
2	2012-12-31
3	2012-12-31
4	2010-12-31

É isso aí!

Se tiver qualquer dúvida sobre qualquer uma dessas etapas, não hesite em entrar em contato com a gente!

Happy Coding!



Equipe Trading com Dados