Nosso primeiro passo consiste em decidirmos que ferramenta usar para escrevermos o código que analisará nossos dados. Existem diversas ferramentas e formas de trabalharmos, e optaremos por uma bastante conhecida quando se usa Python, o Jupyter.

Este projeto permite a criação de um caderno, em que fazemos anotações à medida em que exploramos os dados. Não existe uma maneira única de se trabalhar com ele, e podemos baixá-lo em nossa máquina. Além disso, é possível utilizar equivalentes do Jupyter, ou até mesmo o próprio Jupyter na web, na nuvem.

O site do Kaggle, por exemplo, permite a criação dos *notebooks* e contém vários dados a serem analisados, os chamados *datasets*. No nosso caso, usaremos o Colaboratory, do Google. Trata-se de uma versão dos *notebooks* no *cloud*, sem a necessidade de baixar ou instalar algo localmente.

O único requisito para usarmos o Google Colab é uma conta do Gmail. Feito o login, clicaremos em "New Python 3 Notebook" na parte inferior da caixa de diálogo. O novo *notebook* será criado, e o renomearemos de "Introdução à Data Science". Se clicarmos em "File", existem as opções de salvarmos uma cópia no Drive, no GitHub, ou na nuvem.

A célula que é exibida na tela é onde digitamos o código Python, como um simples `nome = "Guilherme"` . Ao clicarmos no botão de *play*, um círculo escuro com um triângulo logo à esquerda da célula, o código será rodado. A primeira vez costuma demorar um pouco mais, pois é preciso acessar o *cloud* e uma máquina especial (virtual, ou um contêiner), para que possamos rodar nosso código Python.

Feita a conexão, o código é rodado, mas já que uma expressão como esta não devolve nada, nada é impresso na tela. Criaremos uma célula clicando em "Code" com um quadrado e um símbolo de `+` , na parte superior e logo abaixo do menu de ferramentas principal. A primeira célula, então, já foi rodada, e nesta escreveremos `print(nome)` e pressionaremos o *play*. Teremos o retorno `Guilherme` , como esperado.

> As variáveis ficam em memória à medida em que executamos os códigos, e uma célula pode ter várias linhas de código.

Criaremos mais uma linha de código devolvendo o resultado da linha digitada anteriormente:

```
idade = 30
idade
```

COPIAR CÓDIGO

Por fim, poderemos substituir variáveis:

```
idade = 38
```

COPIAR CÓDIGO

A qualquer instante, é possível clicar em "Runtime" e reiniciar tudo que está sendo executado com "Restart runtime...". Quando clicarmos em "Yes", perderemos todas as variáveis em memória, isto é, tudo o que foi executado anteriormente. As saídas anteriores são mantidas na tela para o caso de querermos consultá-las, mesmo que não existam mais.

Com o atalho "Shift + Enter" conseguimos rodar as células, no entanto, se o fizermos após a reinicialização, obteremos um erro indicando que a variável `nome` não foi definida. Porém, se usarmos o mesmo atalho em cada uma das células anteriores e executarmos o código, deixaremos de ter esse erro.

No caso, removeremos todas as células clicando nos três pontos localizados do lado direito de cada uma delas, e em "Delete cell". Com isso, estamos prontos para começar a analisar os nossos dados!

Trabalharemos com um conjunto de dados real, a avaliação de diversos filmes por usuários da internet, do MovieLens. O site abriga variações destes dados, que podem ser baixados sob licença de uso. Existem versões de 20 milhões, 100 mil, 27 milhões de notas (*ratings*) para filmes, e por aí vai.

Neste curso optaremos pelo arquivo contendo 100 mil. Cada versão disponibilizada pelo site é atualizada periodicamente, então, se baixarmos uma delas hoje, provavelmente dali um tempo o mesmo arquivo terá notas e filmes diferentes. Isso porque trata-se de uma amostra aleatória para análise.

> A exata versão que usaremos neste curso pode ser baixada neste link.

Após o download e descompactação, usaremos inicialmente o arquivo `ratings.csv`, com as avaliações organizadas em uma tabela cujos cabeçalhos são: "userId", "movieId", "rating" e "timestamp", ou seja, usuário, filme avaliado, nota e o momento em que ela foi atribuída no site, respectivamente. Na nossa análise apenas as três primeiras colunas nos interessam.

Ao abrirmos o arquivo, notaremos que os números são separados por vírgulas, pois CSV remete a ***comma-separated values***. E é este o arquivo que queremos ler para analisar os dados.

No Python, existe uma biblioteca com um módulo feito para a leitura de arquivos neste formato, o Pandas. Solicitaremos sua importação, e então a leitura do arquivo CSV:

```
import pandas as pd

pd.read_csv("ratings.csv")
```

COPIAR CÓDIGO

Rodaremos o código com "Shift + Enter", mas nos depararemos com uma mensagem informando que o arquivo `.csv` não foi encontrado. Claro, pois ele se encontra na nossa máquina local, enquanto o código está sendo rodado no *cloud*. Caso você rode o código no Jupyter da sua máquina, basta que ele esteja no mesmo diretório, com o *path* adequado. Caso o arquivo esteja na nuvem do Google, como o subiremos?

Clicaremos na aba escura com um `>`, localizada na extrema esquerda da tela, em "Files" e "Upload". Será exibida uma mensagem indicando que os arquivos são deletados toda vez que zeramos a nossa *runtime*, após o qual o nosso arquivo é listado. Rodaremos tudo mais uma vez e, agora sim, o arquivo é lido e trazido com sucesso.

São muitas informações, portanto atribuiremos tudo isso a `notas` e, em vez de todas, pediremos para que apenas as cinco primeiras avaliações sejam exibidas, isto é, a "cabeça" (*head*) da lista de elementos:

```
import pandas as pd

notas = pd.read_csv("ratings.csv")
notas.head()
```

COPIAR CÓDIGO

Há diversas maneiras de saber quantas avaliações existem, e uma delas é pedir o formato da tabela, com `notas.shape`. Isto nos retornará a informação de que há `100836` avaliações e `4` colunas. O contador à esquerda, na tabela, será denominado **índice**, que não consideramos como sendo uma coluna.

Continuando, caso queiramos trabalhar com o português, e não inglês, alteraremos os nomes das nossas colunas com o atributo `columns`:

```
notas.columns = ["usuarioId", "filmeId", "nota", "momento"]
```

COPIAR CÓDIGO

Feita esta redefinição, solicitaremos a impressão dos dados com `notas.head()`, na mesma célula. Assim sendo, `notas` é um objeto do Pandas com várias colunas e `0` ou várias linhas, um tipo conhecido como ***Pandas DataFrame***, cuja documentação corrente (versão `0.24.1`) indica suas inúmeras possibilidades.

De maneira rápida, o que conseguimos analisar com o que temos até aqui?

A coluna "nota" contém os valores `4.0` e `5.0`, mas será que eles são únicos? Para consultarmos todos os valores desta coluna, digitaremos e rodaremos o seguinte código:

```
notas['nota']
```

COPIAR CÓDIGO

Já entendemos que são `100836` valores, e anteriormente os dados eram impressos de forma visualmente agradável, em uma tabela, por ser um *dataframe*. Agora que solicitamos uma única coluna, por padrão, ela será uma **série de números**, que chamamos de ***Pandas Series***. Trata-se de uma série de dados, e de acordo com sua documentação, ela também fornece uma grande quantidade de possibilidades.

Por exemplo, para sabermos quais são os valores colocados nesta coluna de maneira única, utilizamos `unique()`. Ao usarmos o código `notas['nota'].unique()`, e o rodarmos com "Shift + Enter", o retorno será:

```
array([4., 5., 3., 2., 1., 4.5, 3.5, 2.5, 0.5, 1.5])
```

As notas, portanto, variam de `0.5` a `5`, e a nota `0` não foi dada em nenhum momento. O Pandas serve para a leitura e escrita de um conjunto de dados de diversas maneiras, e também para extrair informações a partir destes dados.

Se quisermos saber quantas vezes uma nota específica aparece nesta coluna, poderemos usar:

```
notas['nota'].value_counts()
```

COPIAR CÓDIGO

Isso imprimirá duas colunas de valores, sendo a primeira com as notas e a segunda a quantidade de vezes que ela foi dada, ordenadas de forma decrescente (do maior para o menor). Para encontrarmos a média destas notas, utilizaremos:

```
notas['nota'].mean()
```

COPIAR CÓDIGO

O que trará o valor `3.501556983616962`. Existem outras medidas que serão vistas neste curso, mais ou menos relevantes dependendo do contexto. A seguir, continuaremos explorando tudo isso e mais.