Transcrição do curso de **Introdução à Data Science**

Boas vindas ao nosso curso de **Introdução à Data Science** com Python.

Aqui, carregaremos um conjunto de dados do mundo real, oriundo do [MovieLens](https://movielens.org/), que trabalha com filmes e notas.

Exploraremos estas informações via **Exploratory Data Analysis** (que em tradução livre seria algo como "análise de dados de maneira exploratória"), utilizando diversas bibliotecas do Python.

Estas ferramentas poderiam ser de outras linguagens sem nenhum problema. Optamos pelo Python que, atualmente, está sendo muito bem utilizado no mercado para este tipo de trabalho.

Usaremos ferramentas como **NumPy**, **Pandas**, **Seaborn** para visualização, **Matplotlib**, ou o próprio Pandas com funções de plotagem próprias, para explorar os dados sobre os quais faremos queries, pesquisas de informações sobre tendências de tais dados, como a Tendência Central, a média ou mediana.

Entenderemos a distribuição das notas dadas aos filmes por meio de histogramas ou boxplots, com os quais faremos comparações. Assim, conseguiremos resumir informações de diversos filmes, como as línguas mais populares contidas no conjunto de dados coletados, por meio de visualizações de gráficos mais interessantes e difíceis de serem compreendidos de início.

Aqui, teremos uma introdução ao assunto, à medida em que exploramos três arquivos de dados diferentes — dois do GroupLens (MovieLens) e um do [TMDb](https://www.themoviedb.org/) (The Movie Database), que contém informações de diversos filmes.

Falaremos sobre a parte de **visualização de dados**, lidaremos com boas e más práticas, e com questões que poderão ser aprofundadas ou complementadas em outros cursos, de acordo com o seu interesse pessoal.

Talvez você queira seguir estudando Estatística, por exemplo, para fazer testes estatísticos, ou para **Data Visualization**, para criar dashboards e apresentações gráficas, exibindo visualizações que fazem sentido e contam uma história. Ou então aprofundar mais no Pandas, e em como trabalhar com os seus dados.

Além de modelagens, regressões, entre outros temas, há diversas áreas que podem ser continuações deste curso. Vamos começar?

**Preparando ambiente**

# Olá

Meu nome é Guilherme Silveira, e neste curso vamos analisar e explorar alguns dados de forma descritiva, utilizando o [Pandas](http://pandas.pydata.org/pandas-docs/stable/), e visual, com o [Seaborn](http://seaborn.pydata.org/introduction.html).

[Neste link](https://github.com/alura-cursos/introducao-a-data-science/archive/aula0.zip) você encontra o material com os dados que iremos analisar neste curso.

### Ambiente de análise

Usaremos uma ferramenta do Google chamada [Colaboratory](https://colab.research.google.com/notebooks/welcome.ipynb), que é similar ao [Jupyter Notebook](https://jupyter.org/) e não requer configuração para ser usada.

### Colaboratory

Para usar este ambiente, é necessário ter uma conta **gmail**, pois todo notebook ficará armazenado no Drive. Caso não tenha um gmail, faça o seu clicando neste [link](https://accounts.google.com/signup/v2/webcreateaccount?flowName=GlifWebSignIn&flowEntry=SignUp).

### Informações importantes sobre o Colaboratory

* O código do seu notebook é executado em uma máquina virtual dedicada à sua conta. As máquinas virtuais são recicladas após um determinado tempo ocioso, ou caso a janela seja fechada;
* Para restaurar seu notebook, talvez seja necessário refazer o upload do arquivo .csv e executar as opções "Runtime" e "Restart and run all...".

### Posso usar outro ambiente Jupyter Notebook para realizar este curso?

Sim, sem problemas. Caso queira utilizar o [Anaconda](https://www.anaconda.com/download), visto no curso de [Pandas](https://cursos.alura.com.br/course/introducao-python-pandas) também é possível.

#### Caso queria utilizar Anaconda

Temos um vídeo ensinando a instalação do [Python no Windows](https://cursos.alura.com.br/course/python-3-introducao-a-nova-versao-da-linguagem/task/22687), e em [outras plataformas](https://cursos.alura.com.br/course/python-3-introducao-a-nova-versao-da-linguagem/task/22688), com o detalhe de que no caso trata-se da versão 3.6, e a instalação do [Anaconda](https://cursos.alura.com.br/course/introducao-python-pandas/task/40979).

### Vamos começar?

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](http://jupyter.org).

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](http://www.kaggle.com), por exemplo, permite a criação dos *notebooks* e contém vários dados a serem analisados, os chamados *datasets*. No nosso caso, usaremos o [Colaboratory](http://colab.research.google.com), 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

Por fim, poderemos substituir variáveis:

idade = 38

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](http://grouplens.org/datasets/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](http://bit.ly/ml-latest-small-sample).

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](https://pandas.pydata.org/). Solicitaremos sua importação, e então a leitura do arquivo CSV:

import pandas as pd

pd.read\_csv("ratings.csv")

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()

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"]

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](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html) (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']

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](http://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.html), 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()

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()

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.