**Data Science**

**Curso I – Primeiros Passos:**

Link da documentação das bibliotecas que utilizaremos nesse curso:

- Pandas: <https://pandas.pydata.org/pandas-docs/stable/>;

- Seaborn: <http://seaborn.pydata.org/introduction.html>.

Vou colocar todo arquivo escrito na pasta da aula, mas aqui está o link do drive, de qualquer forma: <https://drive.google.com/drive/folders/1YV0fPKe-_f5beekL2cBUVSOsZ_2G6uaS>.

1. **Aula 1 – Data Science – Dados e Visualizações:**
   1. Podemos sim alterar o nome das colunas passando uma lista com os novos nomes a partir do atributo colunas do nosso df:

notas.columns = ['usuarioId', 'filmeId', 'nota', 'momento']

* 1. O pandas possui visualização e construção de dados a partir do método plot().
     1. O plot recebe o parâmetro kind=’’, onde podemos escolher o tipo de gráfico, como um hist de histograma, por exemplo. Se colocar somente plot() ele cria um plot normal onde não vemos nada:

notas.nota.plot(kind='hist')



notas.nota.plot()



* + - 1. O hist é um tipo de gráfico que conta uma história, mostrando no eixo x as notas que um filme recebeu e no y a quantidade de notas. Basicamente o inverso do plot.
    1. S.median(): retorna a mediana de uma series, mostrando exatamente qual o valor que dividi um conjunto de dados no meio:

notas.nota.median()



* + 1. Importamos a seaborn e fizemos um boxplot para visualizar melhor os gráficos:

import seaborn as sns

sns.boxplot(notas.nota)



* + - 1. A mediana é o traço no meio do box. Quartil 25 é o final esquerdo do box e 75 o final direito. As linhas com traço na ponta dos lados esquerdo e direito são, respectivamente os limites inferior e superior, ou seja, da mediana para as laterais do box temos 25% cada lado, os li e ls são outros 25%, somando 50% no box e 50% fora.
      2. Com esse gráfico podemos concluir que metade das pessoas, ou seja, o box inteiro, deram notas entre 3 e 4.
  1. O que aprendemos:
     1. Importar pandas que é uma biblioteca de análise de dados;
     2. Ler dados CSV;
     3. Fazer upload de arquivo para o Google Colab;
     4. Renomear colunas;
     5. Contar dados;
     6. Melhorar a visualização das informações.

1. **Aula 2 – Análise Exploratória:**
   1. Podemos usar queries para procurar coisas específicas nos nossos dados e selecionar a coluna que queremos, bem como a análise estatística depois dela:

notas.query('filmeId == 1').nota.mean()



* + 1. Dissemos que queremos a média das notas de todo filme que tenha o id == 1 dentro do nosso df de notas.
    2. Se quisermos tirar a média de todas as notas para cada id de filme teria que fazer isso infinitamente, mas, graças ao pandas, podemos agrupar todos os filmes pelo seu id com o df.groupby(‘coluna’) e tirar a média:

notas.groupby('filmeId').mean()



* + - 1. Contudo queremos só a coluna de notas, então podemos selecionar somente ela. O retorno será uma series. Para transformar em df basta colocar um to\_frame(). Como já vimos em outros cursos, podemos colocar tudo numa mesma linha para facilitar, ficando assim:

notas.groupby('filmeId').nota.mean().to\_frame()



* + 1. Após atribuir a uma variável, podemos fazer um histograma para ver a distribuição da média das notas:

medias\_por\_filme = notas.groupby('filmeId').nota.mean().to\_frame()

medias\_por\_filme.plot(kind='hist')



* + 1. Ou então pedir um sns.boxplot(medias\_por\_filme), porém, aprendi que ele não aceita frames, apenas series, nesse caso removi o s.to\_frame() da linha acima, rodei novamente e ele aceitou fazer o gráfico:

medias\_por\_filme = notas.groupby('filmeId').nota.mean()

sns.boxplot(medias\_por\_filme)



* + - 1. Geralmente boxplots são exibidos na vertical e não horizontal. Para alterar isso basta passar nosso s como y:

sns.boxplot(y=medias\_por\_filme)



* + 1. Para fazer um histograma no seaborn a nomenclatura muda, se tornando sns.distplot(), ou seja, gráfico de distribuição. Sua visualização é diferente também:

sns.distplot(medias\_por\_filme)



* + - 1. Podemos notar que esse histograma é diferente do que o pandas fez, mas isso ocorre de acordo com a distribuição de barras que cada um determina fazer.
      2. O pandas fez 10 separações/caixinhas/barras para distribuir as notas, já o seaborn fez MUITO mais, por isso a exibição fica diferente, mas isso não altera o resultado do gráfico, uma vez que os valores são um só.
      3. Entretanto, se quisermos podemos escolher quantas caixinhas o seaborn vai exibir, isto é, quantos bins=n queremos que ele faça para exibir o gráfico, passando esse como segundo parâmetro/argumento na hora de fazer o plot:

sns.distplot(medias\_por\_filme, bins=10)



* + - 1. Note que ao colocar 10 a distribuição fica exatamente igual ao do pandas.
      2. Precisamos tomar cuidado pois dependendo do número que escolhermos podemos passar diferentes mensagens com nosso gráfico, desse modo podendo passar uma mensagem errônea para frente ou até mesmo uma mentira. Precisamos ter cuidado com isso!
      3. Na prática o próprio seaborn já faz uns cálculos para determinar qual seria a quantidade de bins necessárias para cada gráfico.
      4. Além do hist. Ele também faz uma curva para mostrar mais suavemente como ficam essas notas de uma outra forma.
    1. Tanto o pandas quanto o seaborn utilizam uma biblioteca bem baixo nível para produzir esses gráficos. A biblioteca que ambos usam é a matplotlib.pyplot, sendo importada como plt. Biblioteca essa que também possui seu próprio histograma:

import matplotlib.pyplot as plt

plt.hist(medias\_por\_filme)



* + - 1. Essa lib possui diversos métodos, parâmetros que podem ser passados para alterar e modificar nossos gráficos de diversos modos que não vamos ver nesse curso, mas sim no data visualization mais pra frente.
    1. Voltando um pouco nas anotações vimos que o boxplot do seaborn ficou achatado quando mudamos para a horizontal. Para alterar isso podemos utilizar a biblioteca plt, uma vez que ela é quem está sendo rodada por baixo dos panos pelas outras libs que estamos utilizando. Para configurar o tamanho da imagem do gráfico utilizamos o plt.figure(figsize=(x,y)), podendo ser totalmente personalizado:

plt.figure(figsize=(5,8))

sns.boxplot(y=medias\_por\_filme)

