**Tópico 2 – Gráficos de Barra para Proporções**

**OBJETIVOS**

* Representar proporções com gráfico de barras;
* Criar e customizar gráficos de barras com matplotlib no Jupyter Notebook.

É comum, em conjuntos de dados, que o todo esteja dividido em muitos grupos. Por exemplo, considere um conjunto de dados demográficos sobre os países que fazem parte da ONU, que, atualmente, possui 193 países-membros. Certo, você deve está se perguntando: “sim, mas e aí?” E aí que isso significa que são muitas classes! Você já sabe que, nesses casos, gráficos de pizza não são adequados. Então, qual seria a alternativa? Uma abordagem muito utilizada, conhecida como gráficos de barras, que são ótimos para representar quantidades relacionadas à categorias e podem, igualmente, ser utilizados para representar proporções.

Neste tópico, você irá criar gráficos de barras com a biblioteca matplotlib, para visualizar proporções em um conjunto de dados real.

Em primeiro lugar, vamos desenvolver alguns exemplos. Para isso, iremos utilizar um conjunto de dados muito interessante sobre vendas de videogames. Ah, vale lembrar que esses dados estão disponibilizados, de forma aberta, em um arquivo CSV em uma plataforma chamada Kaggle, pois o arquivo possui informação de mais de 16 mil jogos, e o seu download pode ser feito pelo seguinte link: <https://raw.githubusercontent.com/cabrau/curso-analise-visualizacao-dados-python/master/vgsales.csv>

Por isso, é importante salvar o arquivo na mesma pasta em que se encontra o Jupyter Notebook, que você criou, para acompanhar este conteúdo com o nome de vgsales.csv.

Agora, abra e explore um pouco esses dados, utilizando o pandas. A seguir, confira o código. Depois, verifique o seu resultado em uma tabela com várias colunas: da esquerda para direita, os atributos representam a posição entre os mais vendidos, o nome do jogo, a plataforma, o ano de lançamento, o gênero, a empresa que produziu, as vendas na América do Norte, nos Estados Unidos, no Japão, em outros locais e o total de vendas global.

1 games = pd.read\_csv('vgsales.csv')

2 games.head()

|  | **Rank** | **Name** | **Platform** | **Year** | **Genre** | **Publisher** | **NA\_Sales** | **EU\_Sales** | **JP\_Sales** | **Other\_Sales** | **Global\_Sales** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | Wii Sports | Wii | 2006.0 | Sports | Nintendo | 41.49 | 29.02 | 3.77 | 8.46 | 82.74 |
| 1 | 2 | Super Mario Bros | NES | 1985.0 | Platform | Nintendo | 29.08 | 3.58 | 6.81 | 0.77 | 40.24 |
| 2 | 3 | Mario Kart Wii | Wii | 2008.0 | Racing | Nintendo | 15.85 | 12.88 | 3.79 | 3.31 | 35.82 |
| 3 | 4 | Wii Sports Resort | Wii | 2009.0 | Sports | Nintendo | 15.75 | 11.01 | 3.28 | 2.96 | 33.00 |
| 4 | 5 | Pokemon Red/Pokemon Blue | GB | 1996.0 | Role-Playing | Nintendo | 11.27 | 8.89 | 10.22 | 1.00 | 31.37 |

Na linha 1, os dados foram carregados em um DataFrame e atribuídos à variável games. Na linha 2, foram exibidas as colunas e as 5 primeiras linhas do DataFrame. Note que a tabela possui diversos atributos, categóricos e quantitativos. Nos exemplos seguintes, explore as proporções de vendas globais (atributo Global\_Sales), entre os gêneros de jogos (atributo Genre).

O primeiro passo a seguir é tentar visualizar a [proporção](https://leadfortaleza.com.br/ead/glossary/Propor%C3%A7%C3%A3o) de gêneros, utilizando os gráficos de pizza e de rosca. Para isso, será necessário agrupar os dados em torno dos gêneros, para calcular a soma que cada gênero teve de venda global no total. Feito isso, em seguida, deve-se utilizar o método group\_by() do pandas. Confira o código a seguir e acompanhe os dados na tabela.

1 generos = games.groupby(by='Genre').sum()

2 generos

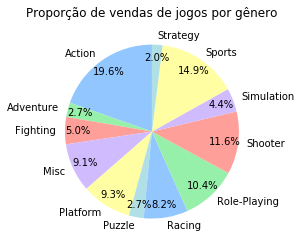
|  | **Rank** | **Year** | **NA\_Sales** | **EU\_Sales** | **JP\_Sales** | **Other\_Sales** | **Global\_Sales** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Genre** |  |  |  |  |  |  |  |
| **Action** | 26441383 | 6531731.0 | 877.83 | 525.00 | 159.95 | 187.38 | 1751.18 |
| **Adventure** | 14831165 | 2562375.0 | 105.80 | 64.13 | 52.07 | 16.81 | 239.04 |
| **Fighting** | 6484242 | 1675871.0 | 223.59 | 101.32 | 87.35 | 36.68 | 448.91 |
| **Misc** | 14889052 | 3432412.0 | 410.24 | 215.98 | 107.76 | 75.32 | 809.96 |
| **Platform** | 6137545 | 1755347.0 | 447.05 | 201.63 | 130.77 | 51.59 | 831.37 |
| **Platform** | 6137545 | 1755347.0 | 447.05 | 201.63 | 130.77 | 51.59 | 831.37 |
| **Puzzle** | 5603136 | 1144994.0 | 123.78 | 50.78 | 57.31 | 12.55 | 244.95 |
| **Racing** | 9943933 | 2457934.0 | 359.42 | 238.39 | 56.69 | 77.27 | 732.04 |
| **Role-Playing** | 12032228 | 2952379.0 | 327.28 | 188.06 | 352.31 | 59.61 | 927.37 |
| **Shooter** | 9653872 | 2571588.0 | 582.60 | 313.27 | 38.28 | 102.69 | 1037.37 |
| **Simulation** | 7478816 | 1707589.0 | 183.31 | 113.38 | 63.70 | 31.52 | 392.20 |
| **Sports** | 17419112 | 4620621.0 | 683.35 | 376.85 | 135.37 | 134.97 | 1330.93 |
| **Strategy** | 6858962 | 1345757.0 | 68.70 | 45.34 | 49.46 | 11.36 | 175.12 |

Perceba que foi criado um novo DataFrame, atribuído à variável generos, onde o index são os gêneros; e as colunas são o somatório de cada atributo em relação ao gênero.

Agora que manipulamos os dados para calcular o que desejávamos, podemos criar os gráficos de pizza como já fizemos. Confira no exemplo a seguir:

1 plt.pie(x=generos['Global\_Sales'], labels = generos.index, autopct='%1.1f%%',startangle=90, pctdistance=0.85)

2 plt.show()

Figura 6 – Gráfico de pizza de vendas de jogos por gênero

Note que é passado, ao parâmetro x, o valor dos somatórios das vendas globais através da coluna ‘Global\_Sales’ do DataFrame gêneros. Já o parâmetro labels recebe o index do DataFrame, onde estão armazenados os nomes de cada categoria. O que você achou do resultado? Parece muito poluído e é difícil perceber a informação visualmente? Por conta disso, o gráfico de pizza não é a visualização mais adequada para esses dados. Agora... será que o gráfico de rosca daria um resultado melhor? A partir do exemplo a seguir, confira como criá-lo:

1 plt.pie(x=generos['Global\_Sales'], labels = generos.index, autopct='%1.1f%%',startangle=90, pctdistance=0.90)

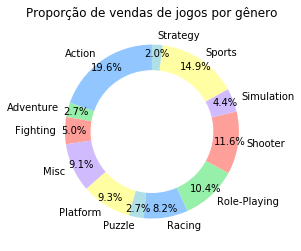
2 my\_circle=plt.Circle((0,0), 0.7, color='white')

3 p=plt.gcf()

4 p.gca().add\_artist(my\_circle)

5 plt.show()

Download do código sem numeração no link a seguir: [Download código](https://leadfortaleza.com.br/ead2pcd/conteudo/tmp/myopenolat_1_102915070652962/aula/code/codet2p4s14.txt)

Figura 7 – Gráfico de rosca de vendas de jogos por gênero

Embora seja um pouco mais leve, o resultado ainda é confuso e exige dedicação do leitor para compreender as informações que se deseja obter. Por esses motivos, deve-se tentar criar uma melhor visualização, utilizando o gráfico de barras.

O matplotlib não dispõe de uma função própria para utilizar gráficos de barras de porcentagens. Então, é necessário calcular vetores com as porcentagens desejadas e exibi-las, usando a função bar(). Também é interessante ordenar os dados, para facilitar a comparação entre eles. Por isso, realizar essas operações, sobre os dados, é muito simples com o pandas. Observe no código, a seguir, como manipular os dados e como criar a visualização com a função bar():

1 generos = generos.sort\_values(by='Global\_Sales',ascending=False)

2 porcentagens = generos['Global\_Sales']\*100.0/generos['Global\_Sales'].sum()

3 plt.style.use('seaborn-darkgrid')

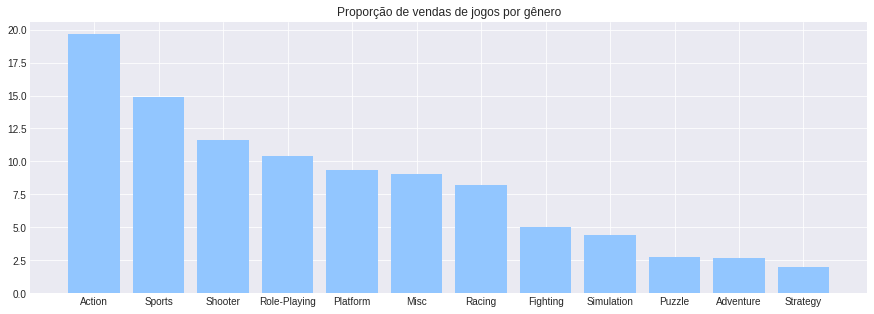
4 plt.figure(figsize=(15,5))

5 plt.bar(generos.index,porcentagens)

6 plt.title('Proporção de vendas de jogos por gênero')

7 plt.show()

Download do código sem numeração no link a seguir: [Download código](https://leadfortaleza.com.br/ead2pcd/conteudo/tmp/myopenolat_1_102915070652962/aula/code/codet2p5s15.txt)

Figura 8 – [Proporção](https://leadfortaleza.com.br/ead/glossary/Propor%C3%A7%C3%A3o) de vendas de jogos por gênero

Primeiro, analise linha a linha esse código. Na linha 1, o DataFrame é ordenado, utilizando o método sort\_values(). A ordem é da maior soma de vendas até a menor. Na linha 2, é calculado o vetor com as porcentagens, multiplicando cada elemento da coluna ‘Global\_Sales’ por 100, e dividindo pelo somatório total da mesma coluna. O resultado é uma sequência de porcentagens referentes a cada gênero. Com esses dados, basta utilizá-los na função bar(), que é feito na linha 5. Em seguida, as linhas 3, 4 e 6 são ajustes da estética da visualização.

O que você achou das visualizações? Perceba que os gráficos passam a mesma informação, mas dando ênfase a diferentes aspectos. Enquanto os gráficos de pizza enfatizam a relação parte de um todo, o gráfico de barras dá ênfase a comparação entre as proporções. Além disso, o gráfico de barras tem como vantagem ser uma visualização mais limpa e melhor distribuída, enquanto os gráficos de pizza concentram muita informação em um espaço pequeno. Ah, lembre-se de que a qualidade de uma visualização está relacionada também ao seu objetivo, ou seja, à informação que ela deve transmitir.

Agora... que conclusões, sobre os dados, você consegue tirar a partir de cada visualização? A seguir, compare os resultados dos códigos dos exemplos anteriores novamente.

Figura 9 – Comparando os resultados

E então? As vendas estão igualmente distribuídas entre as categorias? Algum gênero se destaca em relação aos demais? Note que existe uma grande diferença entre a [proporção](https://leadfortaleza.com.br/ead/glossary/Propor%C3%A7%C3%A3o) de vendas entre os mais vendidos e os menos vendidos.

A seguir, confira uma dica interessante no box Saiba Mais.

#### Ícone Saiba Mais

Além dos gráficos mostrados neste conteúdo, existem outras opções para visualizar proporções, como gráficos de barras empilhadas. Então, procure saber mais a respeito dessas outras opções, consultando as referências deste material!

Caro aluno, você chegou ao final da aula sobre visualizar proporções categóricas. Recapitulando um pouco sobre o que você aprendeu nesta aula, você aprendeu a utilizar as bibliotecas matplotlib e pandas para gerar gráficos de pizza, de rosca e de barras. Também exercitou critérios para gerar boas visualizações. Além de tudo isso, você deve ter percebido como é sempre fácil escolher a melhor opção possível. Porém, tendo em mente o objetivo e as qualidades de boas visualizações, você conseguirá representar bem a informação desejada, não é mesmo? Ah, também é útil testar as opções existentes, não se esqueça!

Espero que você tenha aprendido e experimentado bastante neste conteúdo. Até a próxima!

### Referências

* Matplotlib. <https://matplotlib.org/index.html>. Acessado pela última vez em Maio de 2020.
* Understanding Pie Charts. <https://eagereyes.org/techniques/pie-charts>. Acessado pela última vez em Maio de 2020.
* WILKE, Claus O. **Fundamentals of Data Visualization**. O’Reilly, 2019.