### Olá, estudante!

Ao trabalhar com dados categóricos, você deverá sempre analisar as quantidades associadas a grupos. Além do valor bruto da quantidade, frequentemente, é útil conhecer o quanto essas quantidades representam proporcionalmente em relação a um todo. Alguns exemplos disso são: a [proporção](https://leadfortaleza.com.br/ead/glossary/Propor%C3%A7%C3%A3o) entre homens e mulheres em um grupo de pessoas; a [proporção](https://leadfortaleza.com.br/ead/glossary/Propor%C3%A7%C3%A3o) de votos em cada partido político; e a [proporção](https://leadfortaleza.com.br/ead/glossary/Propor%C3%A7%C3%A3o) de vendas entre diferentes produtos, por exemplo. Sabendo disso, elas podem ser facilmente compreendidas através de boas visualizações.

A partir disso, nesta aula, você irá aprender os conceitos necessários para criar representações funcionais de proporções em dados categóricos. Além de aprender a utilizar as bibliotecas pandas e matplotlib, para gerar as visualizações mais comumente utilizadas na representação de proporções em dados categóricos.

**Objetivos**

* Conhecer e aplicar conceitos estéticos e funcionais para criar boas visualizações de proporções;
* Representar proporções em dados categóricos com gráficos de pizza;
* Representar proporções em dados categóricos com gráficos de barras.

**Tópico 1 – Gráficos de Pizza**

**OBJETIVOS**

* Conhecer conceitos estéticos e funcionais para criar boas visualizações de proporções;
* Aprender a criar e a customizar gráficos de pizza e de rosca com as bibliotecas matplotlib no Jupyter Notebook.

Relações do tipo parte de um todo são muito úteis para descrever dados, principalmente, através de **porcentagens**. Boas visualizações de dados tornam clara essas relações e auxiliam na compreensão dos dados. Porém, visualizar proporções nem sempre é uma tarefa simples e existem diferentes métodos que se adequam a cada situação. Uma das visualizações mais utilizadas para representar porcentagens são os famosos gráficos de pizza.

Neste tópico você aprenderá a criar gráficos de pizza com matplotlib para visualizar proporções, além de ter a noção de em que casos esses gráficos são melhor aplicados. Vamos lá?! Ótimo!

Você provavelmente já teve contato com gráficos de pizza antes, e, talvez, até já tenha criado alguns. Então, sabe-se que um gráfico de pizza é um círculo dividido em fatias, normalmente de cores diferentes, lembrando uma pizza. Em inglês, esse gráfico é chamado pie chart, ou gráfico de torta. Por isso, as proporções entre grupos que se deseja visualizar são mapeadas na área de cada fatia. Com isso, pela sua simplicidade e facilidade de entendimento, gráficos de pizza são amplamente utilizados. Por causa dessas funcionalidades, a biblioteca de visualização de dados matplotlib da linguagem Python permite criar e customizar gráficos de pizza de forma simples e rápida. Nos exemplos a seguir, você irá aprender como fazer isso. Ah, os exemplos utilizados neste conteúdo serão criados no ambiente Jupyter Notebooks e necessitam das bibliotecas pandas e matplotlib devidamente instaladas. Por isso, se você utiliza a distribuição Anaconda, que é fortemente recomendado, elas já vem instaladas por default. Então, vamos lá! Em primeiro lugar, é preciso importar e configurar as bibliotecas que serão utilizadas. Confira:

1 import pandas as pd

2 import matplotlib.pyplot as plt

3 %matplotlib inline

Nas linhas 1 e 2, são importadas as bibliotecas, atribuindo a elas os respectivos pseudônimos, que são utilizados por padrão. O comando na linha 3, que é conhecido como palavra mágica, é específico do Jupyter Notebook. Por isso, este comando apenas diz que imagens geradas pelo matplotlib devem ser exibidas na página do Notebook, entendeu? A seguir, confira melhor como você pode criar um gráfico de pizza.

A função para criar gráficos de pizza no matplotlib é pie(). Como outras funções do matplotlib, ela pode receber diversos argumentos para configurar sua aparência. Os parâmetros básicos da função são labels e x, para os quais são passados uma sequência de dados (lista de Python ou Series de pandas, por exemplo) com as categorias e uma sequência de dados com os valores associados às categorias, respectivamente. Confira, no exemplo de código a seguir, a criação de um gráfico de pizza feito para representar dados fictícios sobre pacientes de uma clínica veterinária. Em seguida, confira o resultado da aplicação do código.

1 especie = ['Cachorro', 'Gato', 'Pássaro']

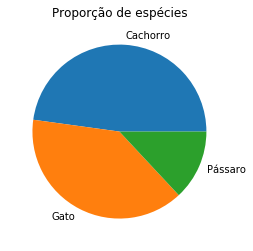
2 quantidade = [110,90,30]

3 plt.title('Proporção de espécies')

4 plt.pie(x=quantidade, labels = especie)

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/codet1p3s5.txt)

Figura 1 – Gráfico [Proporção](https://leadfortaleza.com.br/ead/glossary/Propor%C3%A7%C3%A3o) de espécies

Vamos analisar o código e a visualização? Nas linhas 1 e 2, são criadas duas listas: uma com as categorias de animais atendidos e outra com a quantidade de pacientes de cada categoria. Já a linha 3 configura o título da visualização. Em seguida, na linha 4, é chamada a função pie() do matplotlib, através do pseudônimo plt atribuído à biblioteca. Dessa forma, note que os parâmetros utilizados são x, que recebe os valores quantitativos associados às categorias; e labels, que recebe os nomes das categorias. Então, a função show() exibe a visualização na página.

A partir disso, a [função](https://leadfortaleza.com.br/ead/glossary/Funcao_Visualizacao_de_dados) pie() já calcula a porcentagem que cada quantidade representa de um todo e mapeia essa porcentagem na área das fatias. Nesse momento, perceba que o resultado já fornece uma boa noção da [proporção](https://leadfortaleza.com.br/ead/glossary/Propor%C3%A7%C3%A3o) entre cães, gatos e pássaros atendidos na clínica. Porém, essa noção não é tão funcional, pois um problema dos gráficos de pizza é que a percepção visual humana de áreas não é muito efetiva. Então, comparar áreas pode ser difícil. Por isso, buscando facilitar a compreensão, é útil mostrar na visualização os valores de porcentagens em cada fatia. Vamos verificar como isso pode ser feito?

O matplotlib é altamente customizável. É simples indicar as porcentagens na função pie() através do parâmetro autopct que recebe uma string formatada, indicando como as porcentagens devem ser exibidas. O estilo geral e cores dos gráficos do matplotlib também podem ser alterados, ao se escolher um estilo pré-definido, dentre os vários oferecidos pelo matplotlib. Agora, confira o exemplo de código a seguir e o seu resultado.

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

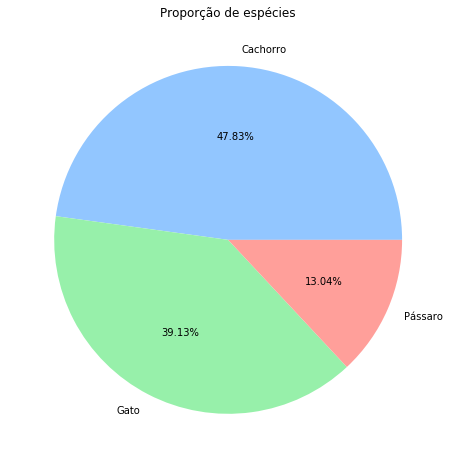
2 plt.style.use('seaborn-pastel')

3 plt.title('Proporção de espécies')

4 plt.pie(x=quantidade, labels = especie, autopct='%1.2f%%')

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/codet1p4s6.txt)

Figura 2 – Gráfico [Proporção](https://leadfortaleza.com.br/ead/glossary/Propor%C3%A7%C3%A3o) de espécies com porcentagens

A linha 1 configura o tamanho da figura em 8 por 8 polegadas. Na linha 2, é escolhido o estilo ‘seaborn-pastel', que utiliza cores mais suaves, através do método style.use(). Logo, uma lista completa dos estilos disponíveis pode ser encontrada neste link: <https://matplotlib.org/3.1.0/gallery/style_sheets/style_sheets_reference.html>. Experimente usar outros estilos, que tal?

Na linha 4, na função pie() foi passado, como argumento, à string '%1.1f%%'. Essa formatação indica que as porcentagens devem ser exibidas com uma casa decimal. Se você quiser exibir com duas casas decimais, por exemplo, pode utilizar '%1.2f%%’. Partindo dessas informações, saiba que a função pie() pode receber diversos argumentos para controlar o visual dos gráficos. Então, para uma lista completa, consulte a documentação da função e os exemplos neste link: <https://matplotlib.org/3.2.1/api/_as_gen/matplotlib.pyplot.pie.html>.

Nessa perspectiva, note que a visualização dá uma melhor noção sobre as proporções de cada espécie. Em seguida, vamos prosseguir com os gráficos.

Existe ainda uma variação do gráfico de pizza que é mais apreciada em algumas situações. Ele é chamado gráfico de rosca (em inglês, donut chart). Em essência, é um gráfico de pizza com área central removida. Por isso, acontece uma diferença de percepção, pois, ao invés de avaliarmos áreas de fatias, estamos avaliando ângulos, o que pode ser mais fácil de perceber, principalmente, quando há várias categorias. Agora, vamos ilustrar isso com dois exemplos, certo? Confira, no exemplo a seguir, um gráfico de pizza das raças de cães atendidos na clínica veterinária fictícia dos exemplos anteriores:

1 races = ['Poodle', 'Golden Retriver', 'Bulldog', 'Pastor Alemão', 'Labrador', 'Sem Raça Definida']

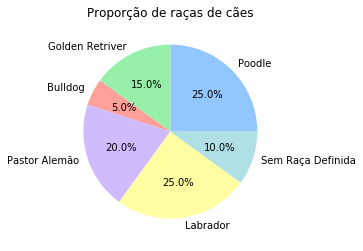
2 quant\_races = [25,15,5,20,25,10]

3 plt.title('Proporção de raças de cães')

4 plt.pie(x=quant\_races, labels = races, autopct='%1.1f%%')

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/codet1p5s7.txt)

Figura 3 – Gráfico [Proporção](https://leadfortaleza.com.br/ead/glossary/Propor%C3%A7%C3%A3o) de raças de cães

Nas linhas 1 e 2, são criadas duas listas, contendo as raças e as quantidades de pacientes de cada raça, respectivamente. Isso é o mesmo que foi feito no exemplo anterior. Já as linhas 3 a 5 configuram e criam o gráfico, assim como foi feito nos exemplos anteriores. Então, perceba que é difícil distinguir proporções similares visualmente através da área. Pois, se a porcentagem não fosse indicada, seria difícil comparar as proporções entre Labrador e Pastor Alemão, por exemplo, e isso não é o que se deseja. Ademais, saiba que boas visualizações devem tornar as informações claras através da percepção de quem vai interpretá-las. Por isso, o gráfico de rosca é uma boa opção de visualização para esses casos. Vamos conferir como eles podem ser criados?

O matplotlib não possui uma função própria para criar gráficos de rosca, mas eles podem ser criados fazendo um gráfico de pizza e desenhando um círculo branco no centro. Você pode conferir como isso é feito de acordo com o código e sua aplicação a seguir.

1 plt.pie(x=quant\_races, labels=races, autopct='%1.1f%%',startangle=90, pctdistance=0.85)

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

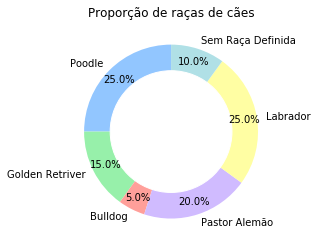
3 p=plt.gcf()

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

5 plt.title('Proporção de raças de cães')

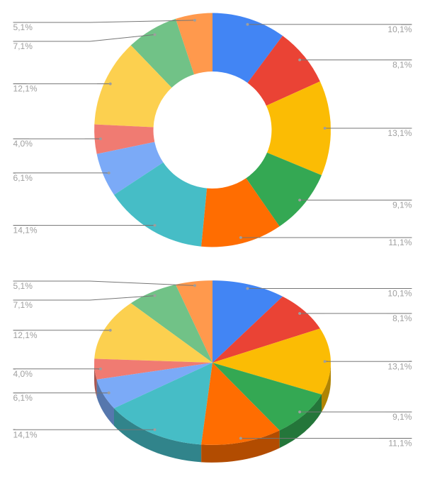
6 plt.show()

Download do código sem numeração no link a seguir: [Download código](https://leadfortaleza.com.br/ead2pcd/conteudo/tmp/myopenolat_1_102915070652962/aula/code/codet1p6s8.txt)

Figura 4 – Gráfico [Proporção](https://leadfortaleza.com.br/ead/glossary/Propor%C3%A7%C3%A3o) de raça e cães adaptado

Perceba que o matplotlib permite criar gráficos não relacionados a dados. Na linha 1, é criado o gráfico e configurado os seus parâmetros, assim como feito anteriormente. Devido a isso, note que, na chamada da função pie(), foram usados dois parâmetros não utilizados antes: startangle, que define o ângulo de início; e pctdistance, que define a distância do texto em relação ao centro do círculo. Nas linhas 2, 3 e 4, é criado um círculo branco que é sobreposto ao gráfico de pizza. Qual desses dois exemplos você achou melhor para perceber as proporções? O gráfico de rosca não necessariamente será sempre preferível ao gráfico de pizza. De fato, em visualização de dados, não existe um tipo de visualização que sempre funciona em toda ocasião, pois tanto o gráfico de pizza quanto o de rosca possuem fraquezas que limitam o seu uso em algumas situações. Então, agora, você vai conhecer melhor essas fraquezas, preparado? Ótimo!

Gráficos de pizza e de rosca são questionados entre especialistas em visualização de dados, gerando muitas discussões sobre sua funcionalidade. Embora possam ser muito úteis para focar na relação parte-todo, entre categorias, e sejam simples de entender; esses gráficos são tão propagados, que muitas vezes acabam sendo mal utilizados e não cumprem bem a sua [função](https://leadfortaleza.com.br/ead/glossary/Funcao_Visualizacao_de_dados). Dessa maneira, esses gráficos possuem duas limitações principais:

Figura 5 – Gráficos de rosca e de pizza inadequados

* Primeiro, sabe-se que é difícil, geralmente, para a percepção visual humana, distinguir áreas. Então, o gráfico de pizza dificulta a comparação entre proporções que possuem valores próximos. Em contrapartida, o gráfico de rosca ameniza isso, mas não cobre essa fraqueza completamente.
* Segundo, quando existe uma grande quantidade de classes, o gráfico pode ficar muito poluído e de difícil compreensão. Apesar das porcentagens estarem indicadas, é difícil perceber, visualmente, a diferença entre as classes. Assim, o ideal é que esse tipo de gráfico seja utilizado quando há poucas classes, no máximo cinco.

A seguir, confira mais uma dica sobre visualizações inadequadas!

Um outro problema comum na má utilização de gráficos de pizza, e em visualizações de dados em geral, é o uso desnecessário de 3D, que deforma as áreas das fatias, sem passar informação alguma, além de dificultar ainda mais a comparação de áreas. Então, concluindo, boas visualizações devem ser objetivas e transmitir a informação desejada claramente.

Por esses motivos, muitos especialistas da área de visualização de dados torcem o nariz para gráficos de pizza. Porém, quando esse gráfico é bem utilizado, ele pode gerar boas visualizações de proporções, pois é enfatizado a percepção de partes de um todo, além de ser visualmente atrativo e de fácil compreensão para o público leigo.

No tópico a seguir, você conhecerá outras opções para representar proporções em conjuntos de dados.