**Olá, estudante!**

Imagine que médicos estão comparando a letalidade de dengue, zika e Chikungunya, de acordo com a idade. Para isso, eles obtiveram dados de muitos pacientes que faleceram de cada doença e passaram a fazer comparações. Qual faixa etária foi mais comum em cada doença? Qual foi a mais incomum? Outros fatores, como hábito de fumar, podem estar relacionados?

Ao analisar distribuições de dados, muitas vezes, é fundamental analisar e comparar como os valores estão distribuídos em relação a diferentes categorias presentes nos dados. A partir daí, é possível obter uma maior compreensão de cada categoria e dos padrões nos dados de forma geral. Por isso, a Visualização de Dados é uma ferramenta poderosa nesse processo.

É com base nisso, que, nesta aula, você aprenderá diferentes técnicas para representar múltiplas distribuições de dados categóricos, utilizando as bibliotecas pandas, matplotlib e seaborn da linguagem Python.

**Objetivos**

* Representar distribuições categóricas com gráfico de dispersão categórico;
* Interpretar e criar gráficos de caixa para representar distribuições categóricas;
* Interpretar e criar gráficos de violino para representar distribuições categóricas.

## Tópico 1 – Gráficos de dispersão categóricos

**OBJETIVOS**

* Aprender a criar gráficos de dispersão categóricos, utilizando a biblioteca seaborn;
* Aprender a criar gráficos de dispersão categóricos através da função stripplot();
* Aprender a criar gráficos de enxame através da função swarmplot().

Você já deve saber que analisar como os valores de uma variável quantitativa estão distribuídos fornece uma importante compreensão sobre os dados. Para visualizar a distribuição de uma única variável, o histograma é uma ferramenta muito útil. Porém, quando existem variáveis categóricas, pode ser muito importante descobrir como os valores se distribuem em relação a cada categoria. Com isso, os histogramas podem ser usados para representar cada categoria separadamente, mas isso dificulta a comparação entre distribuições, que muitas vezes nos interessarão. Portanto, existem diversas técnicas que permitem visualizar e comparar distribuições categóricas em uma única visualização.

Neste tópico, você aprenderá a utilizar uma dessas técnicas, conhecida como gráfico de dispersão categórico, e suas variações, utilizando a biblioteca seaborn.

Para ilustrar os conceitos estudados nesta aula, você irá explorar um conjunto de dados reais sobre os passageiros que estiveram a bordo do famoso navio afundado: Titanic.

Os exemplos serão executados no já conhecido ambiente Jupyter Notebooks e necessitam das bibliotecas pandas, matplotlib e seaborn devidamente instaladas. Em primeiro lugar, deve-se criar um notebook e importar e configurar as bibliotecas que serão utilizadas. Confira o código a seguir:

1 import pandas as pd

2 import matplotlib.pyplot as plt

3 import seaborn as sns

4 %matplotlib inline

5 sns.set(style="darkgrid")

Download do código sem numeração no link a seguir: [Download código](https://leadfortaleza.com.br/ead2pcd/conteudo/tmp/myopenolat_1_102915070709558/aula/code/codet1p2s4.txt)

Nas linhas 1, 2 e 3, são importadas as bibliotecas, atribuindo a elas os respectivos pseudônimos, que são utilizados por padrão. O comando na linha 4, chamado palavra mágica, é específico do Jupyter Notebook, e apenas se refere às imagens geradas pelo matplotlib, exibindo-as na página do Notebook. A linha 5 é opcional e configura o estilo dos gráficos gerados com a biblioteca seaborn. Pronto! A partir disso, você já pode carregar os dados que iremos utilizar e começar a explorá-los. Vamos lá!

Esse conjunto de dados é muito famoso no mundo de análise de dados e pode ser carregado diretamente através do seaborn, sem a necessidade de baixar os dados para a máquina. Confira, no código a seguir, como carregar os dados em um DataFrame de pandas, através da função load\_dataset do seaborn, e exibir as 5 primeiras linhas da tabela de dados, disponível logo após o código.

1 titanic = sns.load\_dataset("titanic")

2 titanic.head()

|  | **survived** | **pclass** | **sex** | **age** | **sibsp** | **parch** | **fare** | **embarked** | **class** | **who** | **adult\_male** | **deck** | **embark\_town** | **alive** | **alone** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 3 | male | 22.0 | 1 | 0 | 7.2500 | S | Third | man | TRUE | NaN | Southampton | no | FALSE |
| 1 | 1 | 1 | female | 38.0 | 1 | 0 | 712.833 | C | First | woman | FALSE | C | Cherbourg | yes | FALSE |
| 2 | 1 | 3 | female | 26.0 | 0 | 0 | 7.9250 | S | Third | woman | FALSE | NaN | Southampton | yes | TRUE |
| 3 | 1 | 1 | female | 35.0 | 1 | 0 | 531.000 | S | First | woman | FALSE | C | Southampton | yes | FALSE |
| 4 | 0 | 3 | male | 35.0 | 0 | 0 | 8.0500 | S | Third | man | TRUE | NaN | Southampton | no | TRUE |

A função load\_dataset() pode carregar diversos conjuntos de dados fornecidos pelo seaborn. No caso, passando o valor ‘titanic’, o conjunto de dados dos passageiros do Titanic foram carregados em um DataFrame e passados para a variável titanic. A função head() exibe as 5 primeiras linhas do DataFrame. Note que esse conjunto de dados possui muitas variáveis, tanto categóricas como quantitativas. Pois bem, agora, você vai prosseguir concentrando a nossa análise na idade (age), classe da passagem (class), sexo (sex) e se o passageiro sobreviveu ou não ao naufrágio (survived).

Para isso, vamos começar pelas idades dos passageiros em cada classe. Suponha que você queira saber se os passageiros eram mais jovens ou mais velhos na primeira, segunda e terceira classe. Ao invés de criar um histograma para cada classe, você pode usar apenas um gráfico de dispersão categórico, pois este separa as categorias em um eixo; e o outro usa para mapear valores de uma variável quantitativa, representando cada dado como um ponto, cuja posição indica o valor da variável quantitativa. Esse gráfico pode ser criado através da função strimplot(). Confira, no exemplo a seguir, como usar strimplot() para visualizar as distribuições de idade em cada classe:

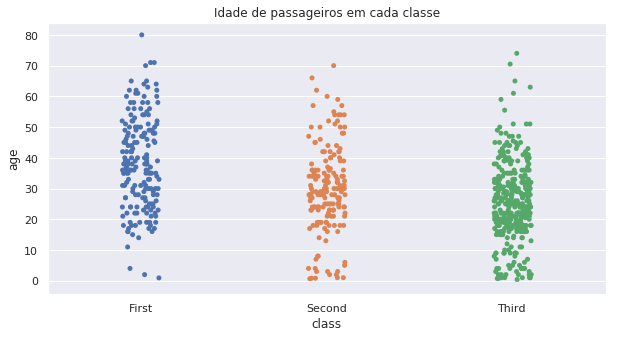
1 plt.figure(figsize=(10,5))

2 plt.title('Idade de passageiros em cada classe')

3 sns.stripplot(x="class", y="age", data=titanic)

4 plt.show()

Download do código sem numeração no link a seguir: [Download código](https://leadfortaleza.com.br/ead2pcd/conteudo/tmp/myopenolat_1_102915070709558/aula/code/codet1p4s6.txt)

Figura 1 – Idade de passageiros em cada classe

As linhas 1 e 2 apenas utilizam matplotlib para configurar a visualização criada com seaborn. Na linha 3, é chamada a função stripplot(), passando a coluna ‘class’ para o eixo x; a coluna ‘age’ para o eixo y; e indicando que os dados pertencem ao DataFrame titanic. Perceba que a classe está mapeada no eixo x, e a idade no eixo y (poderia também ser ao contrário, mapeando a idade no eixo horizontal, dependendo da sua preferência). Assim, cada passageiro é representado por um ponto, e quanto mais alta sua posição vertical, maior a idade do passageiro.

Note que já é possível ter uma noção comparativa entre as classes. Você pode observar, por exemplo, que a primeira classe possuía menos passageiros com menos de 20 anos. Porém, como existem muitos passageiros, ocorre muita sobreposição de pontos, e não se pode ter uma ideia objetiva, principalmente na terceira classe. Uma solução para isso é uma variação chamada gráfico de enxame. Confira a seguir.

É possível representar mais um atributo categórico na visualização, utilizando o parâmetro hue, que irá mapear o atributo para a cor dos pontos. Observe, a seguir, o código e o resultado de como representar o sexo fazendo o parâmetro hue receber a coluna ‘sex’.

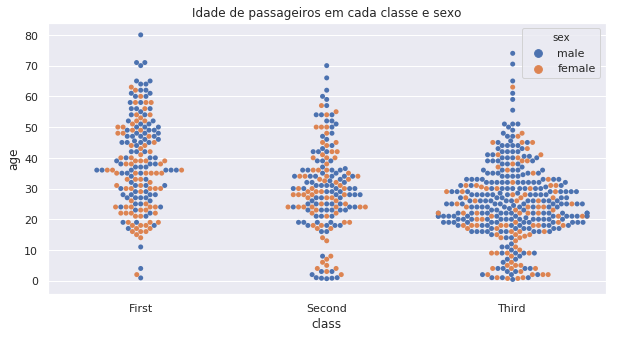
1 plt.figure(figsize=(10,5))

2 plt.title('Idade de passageiros em cada classe e sexo')

3 sns.swarmplot(x="class", y="age", hue = 'sex', data=titanic)

4 plt.show()

Download do código sem numeração no link a seguir: [Download código](https://leadfortaleza.com.br/ead2pcd/conteudo/tmp/myopenolat_1_102915070709558/aula/code/codet1p6s8.txt)

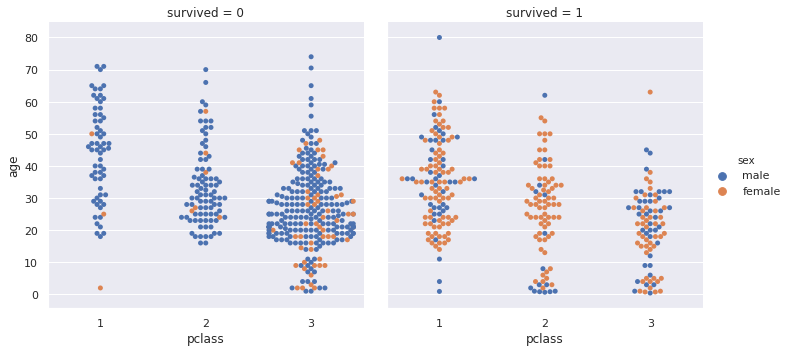
Figura 3 – Idade e sexo de passageiros em cada classe em gráfico de enxame

Agora, os pontos azuis são passageiros masculinos, e os pontos laranjas são passageiras femininas. Assim, novos padrões surgiram e você tem uma noção mais ampla sobre os passageiros! Por exemplo, a maioria dos passageiros da terceira classe eram do sexo masculino. Você notou outros padrões?

Apesar de interessante, representar uma variável pela cor em um gráfico de enxame fornece apenas uma noção geral sobre quantidades de elementos de cada classe, mas não é eficaz em relação a sua distribuição, já que os pontos estão misturados. Mas vamos prosseguir.

Uma outra variável importante é se os passageiros sobreviveram ou não ao naufrágio. Para representar essa variável, você poderia mapeá-la para a cor, mas aí perderia a informação sobre os sexos. Então, uma alternativa para visualizar as duas variáveis, simultaneamente, é criar uma grade de visualizações, usando a função catplot(). Essa é uma função mais genérica, que pode ser usada para criar diferentes tipos de visualização, incluindo os gráficos de dispersão categóricos e de enxame. Porém, ela é baseada na classe FacetGrid do seaborn, que permite criar grades de gráficos. No entanto, não é possível alterar as configurações visuais da visualização através das funções do matplotlib, devido ao FacetGrid ter suas próprias configurações visuais. Confira, no exemplo a seguir, como usar a função catplot() para criar um gráfico de enxame que representa as distribuições de idade por classe, gênero e se o passageiro sobreviveu ou não.

1 sns.catplot(x="pclass", y="age", hue = 'sex', col = 'survived', kind = 'swarm', data=titanic)

Figura 4 – Idade, sexo, classe de passageiros que sobreviveram ou não: gráfico de enxame em grade

Note que apenas dois novos parâmetros foram utilizados: col, que indica as colunas da grade; e kind, que indica o tipo de visualização. Nesse caso, foi passada a coluna ‘survived’ para o parâmetro col, indicando que, em uma coluna, deve ter somente os passageiros que não sobreviveram (0); e, na outra, somente os que sobreviveram (1). Para o parâmetro kind, foi passado o valor ‘swarm’, indicando que a visualização será igual ao swarmplot().

Observe os padrões que surgiram! Do lado direito, pode-se notar que a maioria dos sobreviventes são do sexo feminino. Portanto, verifica-se que muitas crianças sobreviveram também, indicando que a regra de mulheres e crianças, primeiro, foi seguida em parte. Nota-se também que houveram muitos mais sobreviventes na primeira classe em relação à terceira.

O gráfico de enxame é útil para criar uma noção sobre a distribuição dos dados, mas ele possui limitações. Uma delas é a já mencionada falta de precisão ao se representar uma variável categórica pela cor. Outra limitação é que esse tipo de gráfico não se comporta bem para uma grande quantidade de dados, pois, conforme os dados aumentam, ele necessita de mais espaço para não sobrepor os pontos.

Um tipo de gráfico mais compacto que soluciona ambos os problemas é o gráfico de caixa, que será discutido no próximo tópico.