

***INTELIGÊNCIA ARTIFICIAL***

**UA02 / LABORATÓRIO # 1**

1-) Enunciado para Exercício Prático de IA (Análise Exploratória de Dados):

Você está trabalhando com um conjunto de dados sobre pinguins e precisa visualizar a relação entre o comprimento das nadadeiras e o comprimento do bico, agrupando os dados por espécie. Seu objetivo é criar uma visualização que mostre essa relação de forma clara, permitindo identificar padrões ou diferenças entre as espécies.

**Tarefa:**

1.) Escreva um código em Python que:

- Carregue o conjunto de dados "penguins" da biblioteca Seaborn.

- Crie um gráfico de dispersão (scatter plot) combinado com histogramas (joint plot) que mostre a relação entre o comprimento das nadadeiras (`flipper\_length\_mm`) e o comprimento do bico (`bill\_length\_mm`), colorindo os pontos de acordo com a espécie (`species`).

2.) Explique por que você escolheu esse código e essa visualização.

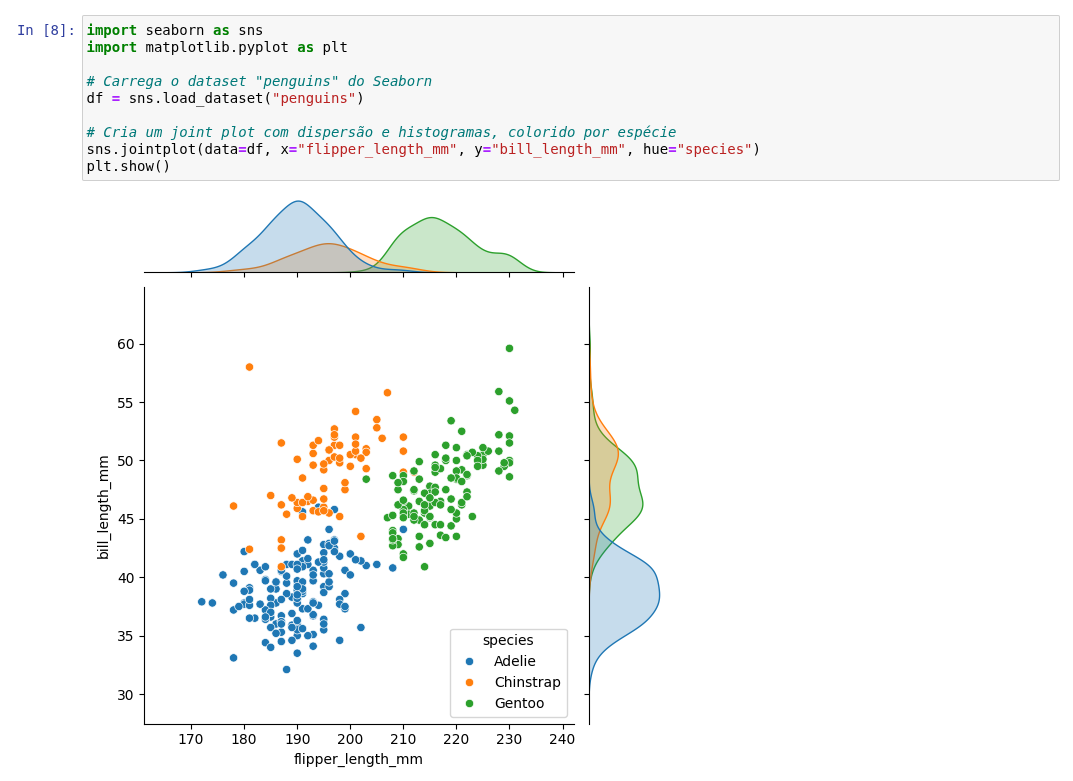
**Código para a Tarefa:**

import seaborn as sns  
import matplotlib.pyplot as plt

# Carrega o dataset "penguins" do Seaborn  
df = sns.load\_dataset("penguins")

# Cria um joint plot com dispersão e histogramas, colorido por espécie  
sns.jointplot(data=df, x="flipper\_length\_mm", y="bill\_length\_mm", hue="species")

plt.show()



**Explicação do Código:**

1.) Bibliotecas utilizadas:

- `seaborn` (sns): Biblioteca de visualização estatística que facilita a criação de gráficos atraentes e informativos.

- `matplotlib.pyplot` (plt): Biblioteca para visualização de dados em Python, usada aqui para exibir o gráfico.

2.) Carregamento do dataset (`sns.load\_dataset`):

- O dataset "penguins" é um conjunto de dados embutido no Seaborn, contendo informações sobre pinguins (como comprimento do bico, nadadeiras, espécie, etc.). É um dataset comum para praticar visualização e análise exploratória.

3.) Visualização com `sns.jointplot`:

- O `jointplot` é usado porque ele combina um gráfico de dispersão (scatter plot) com histogramas nas margens, permitindo visualizar:

- A relação entre duas variáveis contínuas (`flipper\_length\_mm` e `bill\_length\_mm`) no scatter plot.

- A distribuição individual de cada variável nos histogramas.

- O parâmetro `hue="species"` destaca a espécie dos pinguins usando cores diferentes, o que ajuda a identificar padrões ou agrupamentos por espécie.

4.) Por que essa visualização?

- O joint plot é ideal para explorar relações entre duas variáveis e suas distribuições simultaneamente.

- A coloração por espécie (`hue`) permite responder perguntas como:

- "Existe uma diferença no comprimento do bico ou das nadadeiras entre as espécies?"

- "Há correlação entre essas medidas para todas as espécies ou apenas algumas?"

5.) Alternativas consideradas:

- Um simples scatter plot (sem histogramas) mostraria apenas a relação entre as variáveis, mas perderia as distribuições marginais.

- Um pair plot (para todas as variáveis) seria excessivo, pois o foco aqui é apenas em duas variáveis.

2-) Enunciado para Exercício Prático de IA (Análise Exploratória de Dados):

Você recebeu um conjunto de dados sobre carros (`mpg`) que contém informações como consumo de combustível, cilindrada, ano de fabricação, entre outros. Seu objetivo é analisar a relação entre a potência do motor ("horsepower") e o consumo de combustível em milhas por galão ("mpg"), verificando se há diferenças significativas entre os países de origem ("origin").

**Tarefa:**

1.) Escreva um código em Python que:

- Carregue o dataset `mpg` da biblioteca Seaborn.

- Crie uma visualização que mostre a relação entre "horsepower" (eixo X) e "mpg" (eixo Y), diferenciando os carros por país de origem ("origin") usando cores.

- Inclua uma linha de regressão (linear fit) para cada grupo, facilitando a interpretação da tendência.

2.) Explique por que essa visualização é adequada para essa análise.

**Código Esperado (Resposta):**

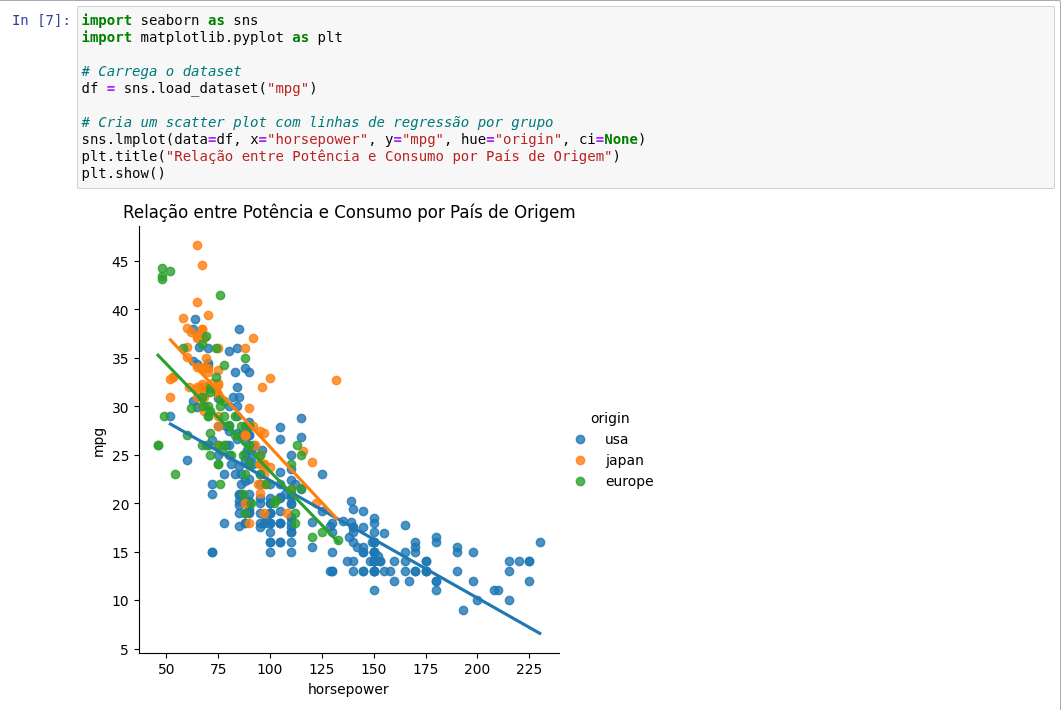
import seaborn as sns  
import matplotlib.pyplot as plt

# Carrega o dataset  
df = sns.load\_dataset("mpg")

# Cria um scatter plot com linhas de regressão por grupo  
sns.lmplot(data=df, x="horsepower", y="mpg", hue="origin", ci=None)

plt.title("Relação entre Potência e Consumo por País de Origem")

plt.show()



**Explicação da Solução:**

1.) Bibliotecas e Dados:

- `seaborn` facilita a criação de gráficos estatísticos.

- O dataset `mpg` já está disponível no Seaborn e contém as variáveis necessárias.

2.) Escolha do Gráfico (`sns.lmplot`):

- Scatter plot + Linha de regressão: Mostra a relação entre duas variáveis contínuas ("horsepower" e "mpg") e a tendência linear (útil para identificar correlação).

- `hue="origin"`: Diferencia os países de origem por cor, permitindo comparar se a relação potência/consumo varia por região.

- `ci=None`: Remove o intervalo de confiança para deixar o gráfico mais limpo (opcional).

3.) Por que essa visualização?

- Responde perguntas como:

- "Carros de qual origem tendem a ser mais eficientes (maior mpg) para uma mesma potência?"

- "Há uma correlação negativa forte entre potência e consumo (ex: carros mais potentes tendem a consumir mais)?"

- Alternativas rejeitadas:

- Um `jointplot` mostraria distribuições marginais, mas sem focar na comparação por grupo.

- Um simples `scatter plot` (sem regressão) não destacaria as tendências.

3-) Enunciado para Exercício Prático de IA (Análise de Dados Categóricos):

Você está analisando um conjunto de dados sobre táxis em Nova York (`taxis`), que inclui informações como valor da corrida, distância percorrida, tipo de pagamento e número de passageiros. Seu objetivo é entender como o valor médio da corrida (`fare`) varia de acordo com o número de passageiros (`passengers`) e se há diferenças significativas entre os tipos de pagamento (`payment`).

**Tarefa:**

1.) Escreva um código em Python que:

- Carregue o dataset `taxis` da biblioteca Seaborn.

- Crie uma visualização que mostre a distribuição do valor da corrida (`fare`) para cada número de passageiros (`passengers`), separando os dados por tipo de pagamento (`payment`).

- Utilize um gráfico que permita comparar facilmente as medianas e a dispersão dos valores.

2.) Explique por que essa visualização é adequada para essa análise.

**Código Esperado (Resposta):**

import seaborn as sns

import matplotlib.pyplot as plt

# Carrega o dataset  
df = sns.load\_dataset("taxis")

# Cria um boxplot para comparar a distribuição de 'fare' por 'passengers' e 'payment'

sns.boxplot(data=df, x="passengers", y="fare", hue="payment")

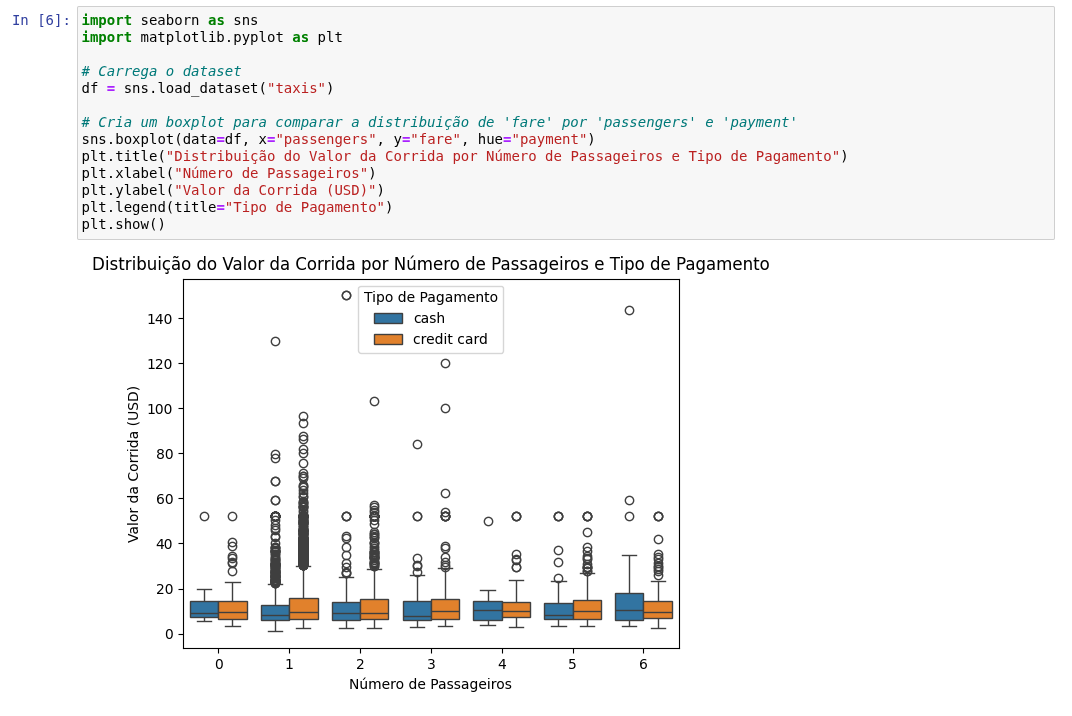
plt.title("Distribuição do Valor da Corrida por Número de Passageiros e Tipo de Pagamento")

plt.xlabel("Número de Passageiros")

plt.ylabel("Valor da Corrida (USD)")

plt.legend(title="Tipo de Pagamento")

plt.show()



**Explicação da Solução:**

1.) Bibliotecas e Dados:

- `seaborn` é usado para criar visualizações estatísticas de forma eficiente.

- O dataset `taxis` já está disponível no Seaborn e contém as variáveis necessárias.

2.) Escolha do Gráfico (`sns.boxplot`):

- Boxplot: Ideal para comparar distribuições, mostrando mediana, quartis e outliers.

- `x="passengers"`: Define o número de passageiros como categoria no eixo X.

- `y="fare"`: Analisa a distribuição do valor da corrida no eixo Y.

- `hue="payment"`: Diferencia os tipos de pagamento por cor, permitindo comparação adicional.

3.) Por que essa visualização?

- Responde perguntas como:

- "O valor médio da corrida aumenta com o número de passageiros?”

- "Há diferenças no valor pago entre cartão de crédito e dinheiro?"

- "Existem outliers (corridas muito caras) em algum grupo?"

- Alternativas rejeitadas:

- Um `violinplot` mostraria a densidade, mas pode ser menos intuitivo para comparar medianas.

- Um `barplot` mostraria apenas a média, ocultando a dispersão dos dados.