Introdução à Ciência de Dados 2024.2

Aula 3 - Um pouco mais de prática!

Prof. Dr. Thiago Pereira da Silva

Agenda

01
Análise Exploratória de Dados

02
Vehicles data set

01
Análise Exploratória de Dados

Análise Exploratória de Dados

- A análise exploratória de dados (ou EDA - *Exploratory Data Analysis*) é uma etapa inicial e essencial no processo de análise de dados.

- Seu objetivo é examinar, resumir e compreender as características principais de um conjunto de dados antes de aplicar técnicas mais complexas, como modelagem estatística ou aprendizado de máquina.
    - Identificar **padrões**, **tendências**, **anomalias** e **relações** nos dados, bem como para **garantir que os dados estão prontos** para serem usados em análises posteriores.

- A EDA é uma **etapa investigativa** que combina **estatísticas descritivas** e **visualização de dados** para revelar informações importantes e preparar os dados para análises mais detalhadas.

Importância da EDA

- **Entendimento do contexto**: Permite conhecer o conjunto de dados e validar hipóteses iniciais.
  - **Exemplo**: *Se os dados representam vendas, é importante entender quais colunas indicam preços, datas, categorias de produtos, entre outros, para saber como analisá-los.*

- **Qualidade dos dados**: Identificar erros, inconsistências e dados ausentes.
  - **Exemplo**: *Um dataset com valores nulos em variáveis importantes (como "idade" ou "renda") pode levar a erros de previsão se não forem tratados.*

- **Padrões e Tendências**: Detectar padrões e tendências que podem não ser evidentes.
  - **Exemplo**: *Uma análise pode mostrar que as vendas aumentam significativamente em finais de semana, influenciando decisões de estoque.*

Importância da EDA

- **Detecção de Outliers e Anomalias -** Permitindo decisões sobre como lidar com outliers (excluir, corrigir ou modelar separadamente).
  - **Exemplo**: *Em um conjunto de dados sobre salários, um valor excessivamente alto pode ser um erro de digitação ou representar um dado válido, como um CEO. Decisões diferentes são tomadas dependendo do caso.*

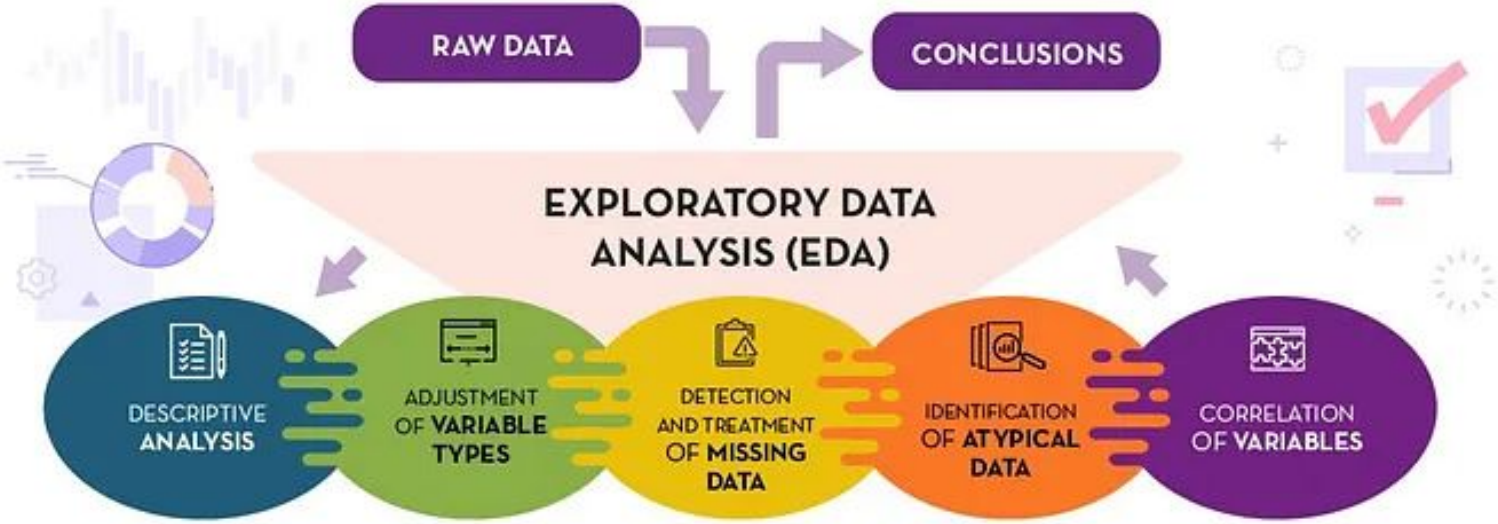
- **Guia para etapas posteriores**: Ajuda na escolha de modelos e técnicas adequadas.
  - **Exemplo**: *Uma análise pode mostrar que as vendas aumentam significativamente em finais de semana, influenciando decisões de estoque.*

Importância da EDA

- **Escolha de Modelos Adequados:** Entender a distribuição dos dados (normal, assimétrica, etc.) é essencial para escolher modelos ou testes estatísticos apropriados.
    - **Exemplo:** *Se os dados de vendas têm uma distribuição logarítmica, pode ser necessário aplicar uma transformação logarítmica antes de modelá-los.*

- **Validação de Hipóteses:** Formular e testar hipóteses iniciais sobre os dados.
    - **Exemplo:** *Uma empresa pode querer saber se o aumento no número de visitas ao site está relacionado ao aumento de vendas.*

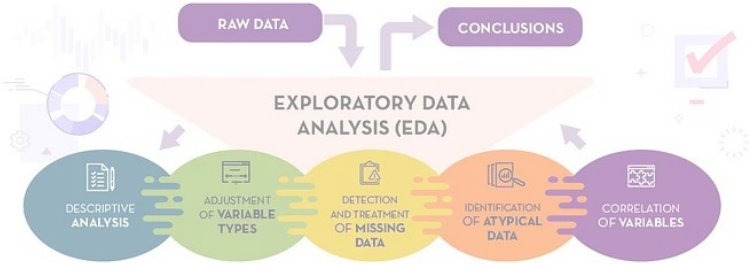
Principais etapas e técnicas da EDA



Principais etapas e técnicas da EDA

- **Entendimento do conjunto de dados e Estatísticas Descritivas**
  - Verificar o tamanho do conjunto de dados (número de linhas e colunas).
  - Identificar os tipos de variáveis (numéricas, categóricas, textuais, etc.).
  - Calcular estatísticas descritivas e distribuição das variáveis.
  - Visualização de dados usando gráficos.

- **Ajustar o tipo das variáveis**
  - Corrigir os tipos de dados (data types) de cada variável.
  - Avaliar as categorias presentes e suas frequências.

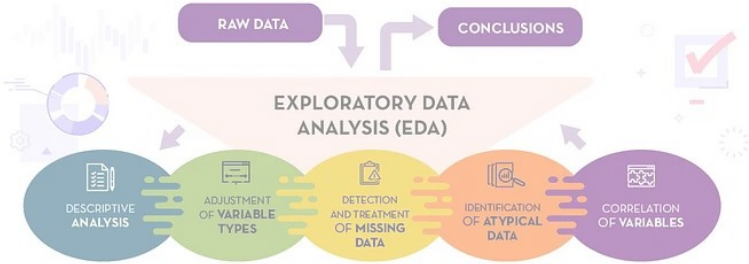


Principais etapas e técnicas da EDA

- **Detectar e tratar valores ausentes**
  - Identificar e tratar dados faltantes (missing data).
  - Corrigir inconsistências ou erros nos dados.

- **Identificação de outliers**
  - Detectar valores que estão fora do padrão esperado.

- **Análise de correlação**
  - Investigar relações entre variáveis numéricas (ex.: matriz de correlação).



EDA com Pandas

Entendimento do conjunto de dados

- **Tamanho do conjunto de dados:**
  - `df.shape  # Retorna uma tupla (n_linhas, n_colunas)`
- **Tipos de variáveis:**
  - `df.dtypes  # Mostra os tipos de cada coluna`
- **Visualização das primeiras linhas:**
  - `df.head()  # Mostra as primeiras 5 linhas do DataFrame`
- **Nomes das colunas:**
  - `df.columns  # Exibe os nomes das colunas`
- **Valores ausentes (missing values):**
  - `df.isnull().sum()  # Mostra a quantidade de valores ausentes por coluna`
- **Verificar Valores Únicos em uma Coluna**
  - `df['genero'].unique()`
- **Verificar a Quantidade de Valores Únicos por Coluna**
  - `df.nunique())`

EDA com Pandas

Estatísticas Descritivas

- **Medidas de Tendência Central:**
  - `df['coluna'].mean() # Média`
  - `df['coluna'].median() # Mediana`
  - `df['coluna'].mode()[0] # Moda`
- **Medidas de Dispersão:**
  - `df['coluna'].std() # Desvio Padrão`
  - `df['coluna'].var() # Variância`
  - `df['coluna'].max() - df['coluna'].min() # Amplitude`
- **Medidas de Forma:**
  - `df['coluna'].skew() # Assimetria (metade esquerda é uma imagem refletida da metade direita)`
  - `df['coluna'].kurt() # Curtose (distribuição mais achatada que gaussiana)`
- **Quartis e Boxplot:**
  - `df['coluna'].quantile([0.25, 0.5, 0.75]) # Quartis`
  - `IQR = df['coluna'].quantile(0.75) - df['coluna'].quantile(0.25) # IQR`
    - A interquartil (IQR) é a diferença entre Q3 e Q1, ajudando a identificar outliers.

EDA com Pandas

Estatísticas Descritivas (continuação)

- **Correlação:**
  - `df.corr() # Matriz de correlação`
- **Visualizações:**
  - `df['coluna'].hist() # Histograma`
  - `sns.scatterplot(x='col1', y='col2', data=df) # Gráfico de dispersão`
  - `sns.boxplot(x='coluna', data=df) # Boxplot`
- **Resumo estatístico das variáveis numéricas:**
  - `df.describe()`
- **Identificar Outliers:**

```
Q1 = df['idade'].quantile(0.25)
Q3 = df['idade'].quantile(0.75)
IQR = Q3 - Q1
# Identificando outliers
outliers = df[(df['idade'] < (Q1 - 1.5 * IQR)) | (df['idade'] > (Q3 +
1.5 * IQR))]
```

EDA com Pandas

Estatísticas Descritivas – Resumo

Etapa	Descrição	Exemplo em Python
1. Verificar Dimensão	Obter o número de linhas e colunas do DataFrame.	`df.shape`
2. Amostra dos Dados	Examinar as primeiras ou últimas linhas do conjunto de dados.	`df.head()` ou `df.tail()`
3. Nomes das Colunas	Identificar os nomes das colunas e possíveis erros de digitação.	`df.columns`
4. Tipos de Dados	Analisar o tipo de dado de cada coluna (numérico, categórico, etc.).	`df.dtypes`
5. Valores Ausentes	Identificar colunas com valores nulos (NaN).	`df.isnull().sum()`
6. Valores Únicos	Verificar valores únicos em colunas categóricas ou quantidade de categorias.	`df['coluna'].unique()` ou `df.nunique()`
7. Estatísticas	Resumo estatístico de colunas numéricas (média, mediana, desvio-padrão).	`df.describe()`

EDA com Pandas

Estatísticas Descritivas – Resumo

8. Detectar Outliers	Identificar valores atípicos com IQR ou boxplots.	`sns.boxplot(data=df['coluna'])`
9. Distribuição	Visualizar distribuição de variáveis numéricas.	`df['coluna'].hist()` ou `sns.distplot(df['coluna'])`
10. Correlações	Analisar relações entre variáveis numéricas.	`df.corr()` ou `sns.heatmap(df.corr(), annot=True)`
11. Consistência	Validar se os dados fazem sentido no contexto (ex.: idade negativa).	`df[(df['idade'] < 0)]` ou verificações específicas.
12. Duplicatas	Identificar e lidar com linhas duplicadas.	`df.duplicated().sum()` ou `df.drop_duplicates()`

EDA com Pandas

Ajustar o tipo das variáveis

- Converter variáveis numéricas para categóricas

```
# Criando um DataFrame
data = {'Produto_ID': [101, 102, 103, 101, 102],
        'Vendas': [200, 150, 300, 250, 180]}
df = pd.DataFrame(data)

# Verificando o tipo original
print(df.dtypes)
# Produto_ID    int64
# Vendas        int64

# Convertendo 'Produto_ID' para categórico
df['Produto_ID'] = df['Produto_ID'].astype('category')
print(df.dtypes)
# Produto_ID    category
# Vendas        int64
```

EDA com Pandas

Ajustar o tipo das variáveis

- Converter variáveis categóricas (strings) para tipo *category*

```
data = {'Gênero': ['Masculino', 'Feminino', 'Masculino', 'Feminino']}
df = pd.DataFrame(data)

# Verificando o tipo original
print(df.dtypes)
# Gênero    object

# Convertendo para categórico
df['Gênero'] = df['Gênero'].astype('category')
print(df.dtypes)
# Gênero    category
```

EDA com Pandas

Ajustar o tipo das variáveis

- **Converter variáveis categóricas para numéricas**

```
# Mapear categorias para números
df['Gênero_Codificado'] = df['Gênero'].cat.codes
print(df)

      Gênero  Gênero_Codificado
0  Masculino                  1
1   Feminino                  0
2  Masculino                  1
3   Feminino                  0
```

EDA com Pandas

Ajustar o tipo das variáveis

- Converter variáveis numéricas armazenadas como strings

```
data = {'Salário': ['2000', '3000', 'NaN', '4000']}
df = pd.DataFrame(data)

# Verificando o tipo original
print(df.dtypes)
# Salário    object

# Convertendo para numérico
df['Salário'] = pd.to_numeric(df['Salário'], errors='coerce')
print(df.dtypes)
# Salário    float64
```

EDA com Pandas

Ajustar o tipo das variáveis

- Converter variáveis de data e hora

```
data = {'Data': ['2024-01-01', '2024-02-01', '2024-03-01']}
df = pd.DataFrame(data)

print(df.dtypes)
# Data    object

# Convertendo para datetime
df['Data'] = pd.to_datetime(df['Data'])
print(df.dtypes)
# Data    datetime64[ns]

# Extrair o ano
df['Ano'] = df['Data'].dt.year

# Diferença entre datas
df['Dias_desde_início'] = (df['Data'] - df['Data'].min()).dt.days
print(df)
```

EDA com Pandas

Ajustar o tipo das variáveis

- Converter booleanos para numéricos

```
data = {'Aprovado': [True, False, True, True]}
df = pd.DataFrame(data)

# Convertendo booleanos para numéricos
df['Aprovado'] = df['Aprovado'].astype(int)
print(df)

   Aprovado
0         1
1         0
2         1
3         1
```

EDA com Pandas

Ajustar o tipo das variáveis

- Tratamento de colunas com múltiplos tipos

```
data = {'ID': [1, 2, 'três', 4]}
df = pd.DataFrame(data)

# Convertendo para numérico (tratando valores inválidos como NaN)
df['ID'] = pd.to_numeric(df['ID'], errors='coerce')
print(df)

     ID
0   1.0
1   2.0
2   NaN
3   4.0;
```

EDA com Pandas

Ajustar o tipo das variáveis – Resumo

Conversão	Método	Exemplo
Numérico → Categórico	`.astype('category')`	`df['coluna'] = df['coluna'].astype('category')`
Categórico → Numérico	`.cat.codes` ou `map()`	`df['coluna_codificada'] = df['coluna'].cat.codes`
String → Numérico	`pd.to_numeric(errors='coerce')`	`df['coluna'] = pd.to_numeric(df['coluna'], errors='coerce')`
String → Data/Tempo	`pd.to_datetime()`	`df['data'] = pd.to_datetime(df['data'])`
Booleano → Numérico	`.astype(int)`	`df['coluna'] = df['coluna'].astype(int)`

EDA com Pandas

Detectar e tratar valores ausentes

- Identificar valores ausentes

```
import pandas as pd

# Criando um DataFrame com valores ausentes
data = {'Nome': ['Ana', 'Bruno', 'Carlos', None],
        'Idade': [28, 35, None, 40],
        'Salário': [3000, None, 4000, 5000]}
df = pd.DataFrame(data)

# Verificar valores ausentes
print(df.isnull())  # True indica valor ausente

# Contar valores ausentes por coluna
print(df.isnull().sum())
```

EDA com Pandas

Detectar e tratar valores ausentes

- Remover valores ausentes
  - Remover linhas com valores ausentes

```
df_sem_nulos = df.dropna()
print(df_sem_nulos
```

  - Remover colunas com valores ausentes

```
df_sem_colunas_nulas = df.dropna(axis=1)
print(df_sem_colunas_nulas)
```

EDA com Pandas

Detectar e tratar valores ausentes

- Preencher valores ausentes
  - Preencher com um valor fixo

```
df['Idade'] = df['Idade'].fillna(30)  # Preenche com 30
print(df)
```

  - Preencher com a média

```
df['Idade'] = df['Idade'].fillna(df['Idade'].mean())
print(df)
```

  - Preencher com a mediana

```
df['Salário'] = df['Salário'].fillna(df['Salário'].median())
print(df)
```

  - Preencher com a moda

```
df['Nome'] = df['Nome'].fillna(df['Nome'].mode()[0])
print(df)
```

EDA com Pandas

Detectar e tratar valores ausentes

- **Interpolação de valores**
  - Para dados numéricos ou temporais, a interpolação preenche valores ausentes com base em dados adjacentes.

```
df['Idade'] = df['Idade'].interpolate()
print(df)
```

- **Tratamento personalizado**
  - Exemplo: Preencher valores ausentes com base em outra coluna

```
df['Salário'] = df['Salário'].fillna(df['Idade'] * 100)
# Supondo relação salário ~ idade
print(df)
```

EDA com Pandas

Detectar e tratar valores ausentes – Resumo

Método	Uso
`dropna()`	Remover linhas/colunas com valores ausentes.
`fillna(valor)`	Preencher com um valor específico, como média ou mediana.
`interpolate()`	Preencher valores com interpolação.
`isnull()` ou `notnull()`	Detectar valores nulos.

EDA com Pandas

Identificação de outliers

- **Métodos para Identificação de Outliers**
  - **Estatísticas Descritivas (Regra do IQR)**
    - Calcular os quartis Q1 (25º percentil) e Q3 (75º percentil).
    - Determinar o IQR = Q3 − Q1
    - Valores abaixo de Q1−1.5 × IQR ou acima de Q3 +1.5 × IQR são considerados outliers.
  - **Visualização com Boxplot**
  - **Análise de Z-Score**
    - O Z-Score mede a distância de um valor em relação à média, em unidades de desvio padrão. Valores com Z-Score acima de 3 ou abaixo de -3 são considerados outliers.
  - **Visualização com Histogramas**
    - Picos inesperados.

EDA com Pandas

Identificação de outliers

- **Métodos para Tratamento de Outliers**
  - **Remoção**

```
df_sem_outliers = df[(df['Idade'] >= limite_inferior) & (df['Idade'] <= limite_superior)]
```

  - **Transformações**

```
df['Salário_Log'] = df['Salário'].apply(lambda x: np.log(x))
```

  - **Substituição**

```
df['Idade'] = df['Idade'].apply(lambda x: limite_superior if x > limite_superior else x)
```

  - **Tratamento Específico**
    - Em alguns casos, os outliers são mantidos como estão, caso sejam relevantes (ex.: renda de bilionários em estudos financeiros).

EDA com Pandas

Identificação de outliers – Resumo

Método	Descrição	Exemplo
IQR	Baseado em limites calculados com quartis.	$\text{limite superior} = Q3 + 1.5 \times IQR$
Z-Score	Mede a distância em desvios-padrão da média.	$Z = \frac{(x-\mu)}{\sigma}$
Boxplot	Visualiza dispersão e possíveis outliers.	`sns.boxplot()`
Scatterplot	Identifica padrões em dados multidimensionais.	`sns.scatterplot()`

EDA com Pandas

Análise de correlação

- **Medição da Correlação**
  - **Coeficiente de Correlação de Pearson, Spearman ou Kendall**

```
correlacao = df.corr(method='pearson')
```

- **Visualizar Correlações**

```
sns.heatmap(correlacao, annot=True, cmap='coolwarm')
```

- **Considerações ao Usar Correlação**
  - Correlação não implica **causalidade**: Mesmo que duas variáveis estejam correlacionadas, isso não significa que uma causa a outra.
  - **Normalização**: Dados em escalas diferentes podem distorcer os resultados.
  - Relações não lineares: O coeficiente de Pearson não captura relações não lineares. Usar gráficos (como scatterplots) pode ajudar a identificar esses casos.

EDA com Pandas

Análise de correlação – RESUMO

Método	Uso
`corr()`	Calcula coeficientes de correlação.
`heatmap()`	Visualiza correlação em uma matriz.
`scatterplot()`	Analisa relações entre pares de variáveis.
Pearson	Relação linear.
Spearman	Relação monotônica.
Kendall	Associação de classificações.

02
Vehicles data set

Vehicles data set

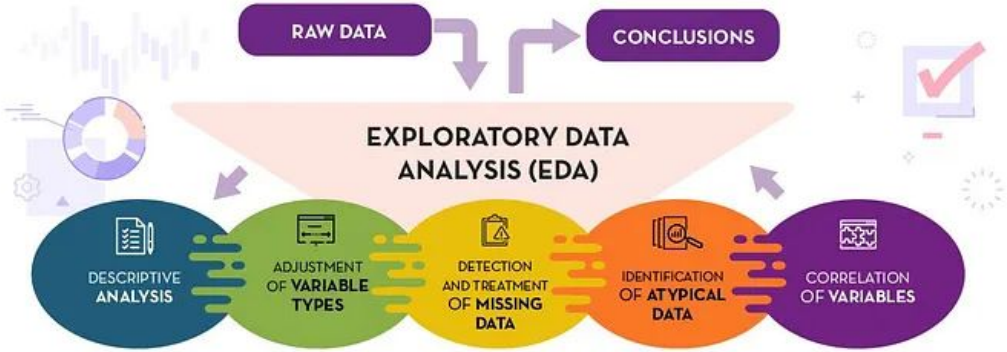
Descrição das Variáveis

- **data.set** - Dado de treino ou de test
- **total.cost** - Custo total do veículo
- **lot.sale.days** - Tempo de venda em dias
- **overage** - Venda após 90 dias
- **mileage** - Kilometragem
- **vehicle.type** - Tipo do veículo (economy, family.medium, family.large...)
- **domestic.import** - Fabricação, domestico ou importado
- **vehicle.age** - Idade do veículo
- **vehicle.age.group** - Grupo de idade do veículo
- **color.set** - Cor
- **makex** - Marca
- **state** - Estado do Carro (região)
- **make.model** - Modelo

Vehicles data set

Para fazer em sala de aula (parte 1)

- Faça a Análise Exploratória do Dataset. Utilize o roteiro apresentado na aula.



Vehicles data set

Para fazer em sala de aula (parte 2)

- Responda às seguintes questões:
  - **1. Vendas e Estoques**
    - Qual é o tempo médio em dias que os veículos passam no lote antes de serem vendidos (**lot.sale.days**)?
    - Veículos de que tipo (**vehicle.type**) têm o menor tempo de venda?
    - Como a quilometragem (**mileage**) influencia o tempo de venda (**lot.sale.days**)?
    - Existe alguma correlação entre o **vehicle.age** (idade do veículo) e o **total.cost** (custo total)?
  - **2. Preço e Custos**
    - Qual é o preço médio dos veículos vendidos, segmentado por **state** (estado)?
    - Veículos importados (**domestic.import**) são, em média, mais caros que os domésticos?
    - Existe alguma diferença significativa no **total.cost** de veículos com cores diferentes (**color.set**)?
    - Como o **total.cost** varia por **vehicle.age.group** (grupo de idade) e por tipo de veículo (vehicle.type)?

Vehicles data set

Para fazer em sala de aula (parte 2) continuação..

- Responda às seguintes questões:
  - **3. Tendências por Categoria**
    - Quais marcas (**makex**) e modelos (**make.model**) são mais vendidas em cada estado?
    - Quais faixas etárias de veículos (**vehicle.age.group**) são mais comuns entre os veículos vendidos?
    - Existe uma tendência sazonal nos **lot.sale.days**, com veículos sendo vendidos mais rapidamente em determinados meses ou estações?
    - Veículos mais antigos (**vehicle.age**) tendem a ser vendidos em períodos específicos do ano?
  - **4. Impacto de Características no Desempenho**
    - Como a quilometragem (**mileage**) afeta o **total.cost** dos veículos, considerando diferentes tipos de veículo (**vehicle.type)**?
    - Quais características (ex.: vehicle.type, vehicle.age.group, domestic.import) mais influenciam o custo total?

Vehicles data set

Para fazer em sala de aula (parte 2) continuação..

- Responda às seguintes questões:
  - **5. Identificação de Outliers e Anomalias**
    - Existem veículos que ficaram muito mais tempo no lote do que a média? Quais são suas características?
    - Algum estado ou marca apresenta veículos com preços significativamente fora da média?
  - **6. Estratégias de Marketing e Logística**
    - Qual é a cor mais popular (color.set) entre os veículos vendidos?
    - Veículos de quais marcas ou modelos têm melhor desempenho em cada estado?

Obrigado!

Alguma dúvida?

thiago.silva@ufmt.br