Leonardo moret de Oliveira

**Documentação do Código**

**1. Importação de Bibliotecas**

import pandas as pd

* **Descrição**: Importa a biblioteca pandas para manipulação e análise de dados.

**2. Carregar a Base de Dados**

carros = pd.read\_csv('base\_carros\_usados.csv')

* **Descrição**: Carrega o arquivo CSV base\_carros\_usados.csv para um DataFrame chamado carros.

**3. Exibir a Estrutura das Variáveis**

carros.info()

* **Descrição**: Exibe a estrutura das variáveis (colunas) no DataFrame, mostrando o nome das colunas, tipo de dados e valores não nulos.

**4. Renomear Colunas**

carros.rename(columns={

'car\_name': 'nome\_carro',

'brand': 'marca',

'model': 'modelo',

'new\_price': 'novo\_preco',

'min cost\_price': 'preco\_minimo',

'max\_cost\_price': 'preco\_maximo',

'vehicle age': 'idade\_veiculo',

'km\_driven': 'km\_rodados',

'seller\_type': 'tipo\_vendedor',

'fuel\_type': 'tipo\_combustivel',

'transmission\_type': 'tipo\_transmissao',

'mileage': 'milhas',

'engine': 'motor',

'max power': 'potencia\_maxima',

'seats': 'assentos',

'selling price': 'preco\_venda'

}, inplace=True)

* **Descrição**: Renomeia as colunas do DataFrame para o idioma português brasileiro, utilizando nomes sem espaços nem acentos.

**5. Criar Dicionário de Dados**

dicionario\_dados = [

{'Coluna': 'id', 'Tipo da variável': 'Quantitativa discreta', 'Descrição': 'Identificador único do carro'},

{'Coluna': 'nome\_carro', 'Tipo da variável': 'Qualitativa nominal', 'Descrição': 'Nome do carro'},

{'Coluna': 'marca', 'Tipo da variável': 'Qualitativa nominal', 'Descrição': 'Marca do carro'},

{'Coluna': 'modelo', 'Tipo da variável': 'Qualitativa nominal', 'Descrição': 'Modelo do carro'},

{'Coluna': 'preco\_novo', 'Tipo da variável': 'Quantitativa contínua', 'Descrição': 'Preço do carro novo'},

{'Coluna': 'preco\_custo\_minimo', 'Tipo da variável': 'Quantitativa contínua', 'Descrição': 'Preço de custo mínimo do carro'},

{'Coluna': 'preco\_custo\_maximo', 'Tipo da variável': 'Quantitativa contínua', 'Descrição': 'Preço de custo máximo do carro'},

{'Coluna': 'idade\_veiculo', 'Tipo da variável': 'Quantitativa discreta', 'Descrição': 'Idade do veículo em anos'},

{'Coluna': 'km\_rodados', 'Tipo da variável': 'Quantitativa contínua', 'Descrição': 'Quilometragem rodada pelo carro'},

{'Coluna': 'tipo\_vendedor', 'Tipo da variável': 'Qualitativa categórica', 'Descrição': 'Tipo de vendedor (particular, revenda)'},

{'Coluna': 'tipo\_combustivel', 'Tipo da variável': 'Qualitativa categórica', 'Descrição': 'Tipo de combustível utilizado pelo carro'},

{'Coluna': 'tipo\_transmissao', 'Tipo da variável': 'Qualitativa categórica', 'Descrição': 'Tipo de transmissão (manual, automática)'},

{'Coluna': 'quilometragem', 'Tipo da variável': 'Quantitativa contínua', 'Descrição': 'Consumo de combustível por quilômetro rodado'},

{'Coluna': 'motor', 'Tipo da variável': 'Quantitativa contínua', 'Descrição': 'Capacidade do motor em litros'},

{'Coluna': 'potencia\_maxima', 'Tipo da variável': 'Quantitativa contínua', 'Descrição': 'Potência máxima do carro em cavalos (HP)'},

{'Coluna': 'assentos', 'Tipo da variável': 'Quantitativa discreta', 'Descrição': 'Número de assentos no carro'},

{'Coluna': 'preco\_venda', 'Tipo da variável': 'Quantitativa contínua', 'Descrição': 'Preço de venda do carro em reais'}

]

dicionario\_df = pd.DataFrame(dicionario\_dados)

* **Descrição**: Cria uma tabela com o nome, tipo e descrição das colunas do DataFrame.

**6. Exibir as 5 Primeiras Linhas**

carros.head(5)

* **Descrição**: Exibe as 5 primeiras linhas do DataFrame.

**7. Exibir as 5 Últimas Linhas**

carros.tail(5)

* **Descrição**: Exibe as 5 últimas linhas do DataFrame.

**8. Verificar e Tratar Valores Faltantes**

carros.isnull().sum()

carros['marca'].value\_counts()

# Preenche valores faltantes na coluna 'novo\_preco' com 'Novo preco nao informado'

carros['novo\_preco'] = carros['novo\_preco'].fillna('Novo preco nao informado')

marcas = ["Maruti", "Hyundai", "Honda", "Mahindra", "Toyota", "Tata", "Ford", "Volkswagen",

"Renault", "Mercedes-Benz", "BMW", "Skoda", "Chevrolet", "Audi", "Nissan",

"Datsun", "Fiat", "Jaguar", "Land Rover", "Volvo", "Jeep", "Mitsubishi",

"Kia", "Porsche", "Mini", "MG", "Isuzu", "Lexus", "Force", "Bentley",

"Ambassador", "OpelCorsa", "ISUZU", "DC", "Maserati", "Daewoo",

"Premier", "Lamborghini", "Ferrari", "Mercedes-AMG", "Rolls-Royce", "Opel"]

# Calcular a média dos preços mínimos para cada marca

media\_preco\_minimo\_por\_marca = carros.groupby('marca')['preco\_minimo'].mean()

# Atualizar valores faltantes com a média correspondente

for marca in marcas:

if marca in media\_preco\_minimo\_por\_marca.index:

carros.loc[carros['marca'] == marca, 'preco\_minimo'] = carros.loc[carros['marca'] == marca, 'preco\_minimo'].fillna(media\_preco\_minimo\_por\_marca[marca])

media\_preco\_maximo\_por\_marca = carros.groupby('marca')['preco\_maximo'].mean()

# Atualizar valores faltantes com a média correspondente

for marca in marcas:

if marca in media\_preco\_maximo\_por\_marca.index:

carros.loc[carros['marca'] == marca, 'preco\_maximo'] = carros.loc[carros['marca'] == marca, 'preco\_maximo'].fillna(media\_preco\_maximo\_por\_marca[marca])

carros = carros.dropna()

carros.reset\_index(drop=True, inplace=True)

* **Descrição**: Verifica valores faltantes. Preenche valores na coluna novo\_preco com uma string informativa. Preenche valores faltantes nas colunas preco\_minimo e preco\_maximo com a média correspondente para cada marca. Remove linhas restantes com valores NaN e redefine o índice.

**9. Excluir Colunas Desnecessárias**

carros.drop(columns=['nome\_carro', 'novo\_preco', 'assentos', 'milhas', 'motor', 'preco\_maximo'], inplace=True)

* **Descrição**: Remove colunas que não são necessárias para a análise. As colunas removidas são:
  + nome\_carro: Nome do carro não é necessário para análise.
  + novo\_preco: Preço do carro novo não é relevante para a análise atual.
  + assentos, milhas, motor: Informações específicas que não são usadas para a análise.
  + preco\_maximo: Coluna removida por decisão do analista.

**10. Excluir Linhas com Marca 'Tata'**

indices\_tata = carros[carros['marca'] == 'Tata'].index

carros = carros.drop(indices\_tata)

carros.reset\_index(drop=True, inplace=True)

* **Descrição**: Remove todas as linhas onde a marca é 'Tata' e redefine o índice.

**11. Alterar Dados na Coluna 'tipo\_vendedor'**

carros['tipo\_vendedor'] = carros['tipo\_vendedor'].replace({

'Individual': 'particular',

'Trustmark Dealer': 'concessionária',

'Dealer': 'distribuidora'

})

* **Descrição**: Substitui valores na coluna tipo\_vendedor por categorias mais descritivas.

**12. Criar e Atualizar Novas Colunas**

carros['ano'] = 2024 - carros['idade\_veiculo']

carros['estado'] = None

carros.loc[carros['km\_rodados'] < 30000, 'estado'] = "novo"

carros.loc[(carros['km\_rodados'] >= 30000) & (carros['km\_rodados'] < 80000), 'estado'] = "moderadamente rodado"

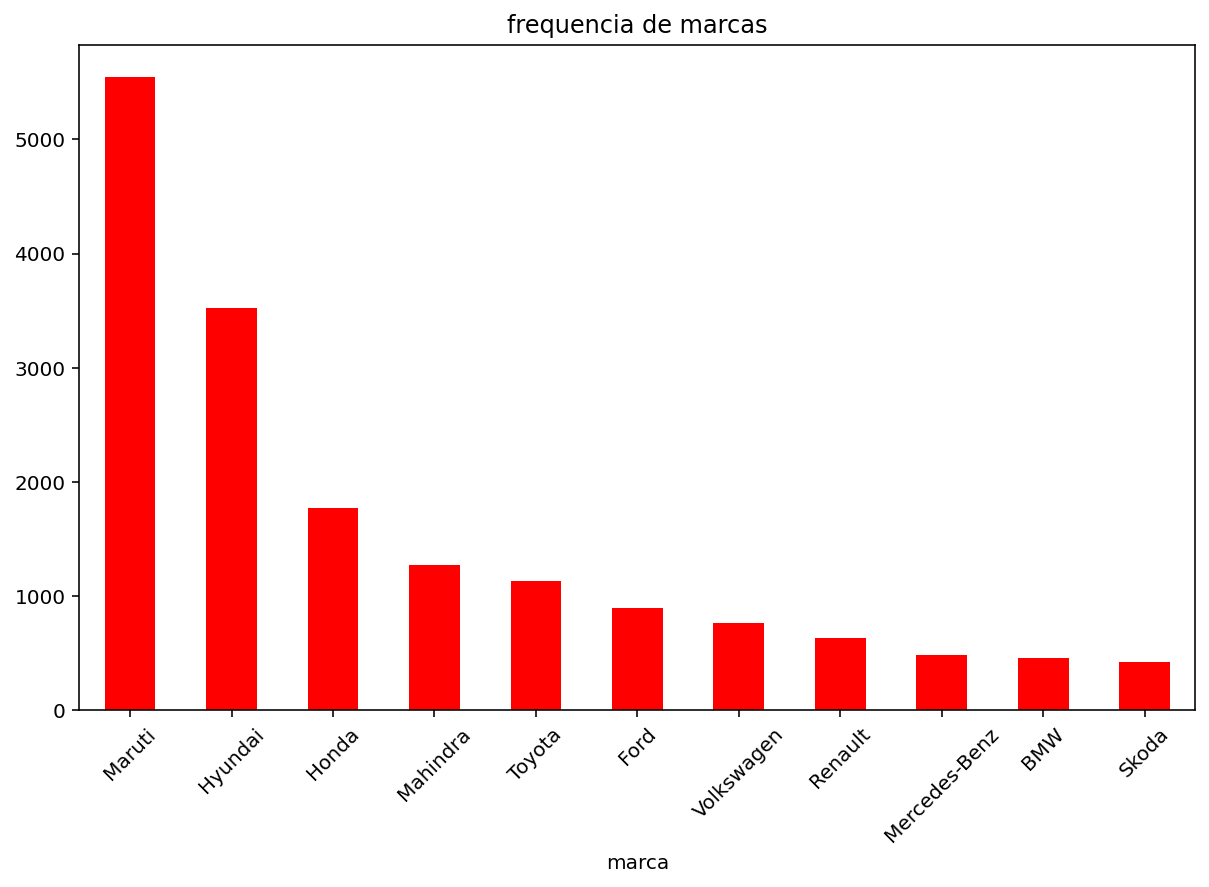
carros.loc[carros['km\_rodados'] >= 80000, 'estado'] = "rodado"

* **Descrição**: Adiciona as colunas ano (ano de fabricação do carro) e estado (condição do carro com base na quilometragem).

**13. Salvar o DataFrame Atualizado**

carros.to\_csv('carros\_usados\_atualizado.csv')

* **Descrição**: Salva o DataFrame atualizado em um arquivo CSV chamado carros\_usados\_atualizado.csv.



A green bar graph with white text

Description automatically generatedA pie chart with text on it

Description automatically generatedA graph of blue rectangular bars

Description automatically generated with medium confidenceA graph with orange bars

Description automatically generated with medium confidenceA graph with blue lines and white text

Description automatically generatedA green bar graph with black text

Description automatically generated

A graph with red lines

Description automatically generatedA white rectangular object with black text

Description automatically generatedA graph of numbers and lines

Description automatically generated with medium confidenceA graph with red lines

Description automatically generated