

✓ Entrega 1 - Big Data

Emilly Mickeli Depine da Silva 23025480


Renan Teixeira Pinheiro 23025274

Gustavo Henrique Santos Araujo 23025397

Fernando José dos Santos 23025299

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 import csv
```

```
1 # Integrando o Colab com o meu Drive para possibilitar a leitura dos arquivos de qualquer dispositivo
2 caminho_dados = '/content/drive/MyDrive/BaseDadosUber/'
3
4 # Lendo os arquivos e definindo o separador padrão como ponto e vírgula, para não confundir as vírgulas com separadores
5 product_df = pd.read_csv(caminho_dados + 'product.csv', sep=';')
6 ride_v2_df = pd.read_csv(caminho_dados + 'ride_v2.csv', sep=';')
7 rideaddress_v1_df = pd.read_csv(caminho_dados + 'rideaddress_v1.csv', sep=';')
8 rideestimative_v3_df = pd.read_csv(caminho_dados + 'rideestimative_v3.csv', sep=';')
```

 <ipython-input-7-5de2d7ce2e12>:12: DtypeWarning: Columns (7,8) have mixed types. Specify dtype option on import or set low_memory=False

```
1 # Somando quantas entradas cada coluna tem
2 print("\nDescrição do product.csv:")
3 print(product_df.describe().loc[['count']])
4
5 print("\nDescrição do ride_v2.csv:")
6 print(ride_v2_df.describe().loc[['count']])
7
8 print("\nDescrição do rideaddress_v1.csv:")
9 print(rideaddress_v1_df.describe().loc[['count']])
10
11 print("\nDescrição do rideestimative_v3.csv:")
12 print(rideestimative_v3_df.describe().loc[['count']])
```



Descrição do product.csv:

	ProviderID	CategoryID
count	237.0	237.0

Descrição do ride_v2.csv:

	RideID	RideStatusID	CompanyID	ProviderID	price	CategoryID	\
count	500000.0	500000.0	500000.0	228157.0	500000.0	24714.0	

	TotalUsers	RideDriverLocationID	ScheduledRide
count	500000.0	14864.0	500000.0

Descrição do rideaddress_v1.csv:

	RideAddressID	RideAddressTypeID	RideID
count	1000000.0	1000000.0	1000000.0

Descrição do rideestimative_v3.csv:

	RideEstimativeID	RideID	WaitingTime	Price	Selected	\
count	2000000.0	2000000.0	2000000.0	2000000.0	2000000.0	

	RideReasonSelectedEstimativeID	Fee
count	234021.0	2000000.0

```
1 # Informações sobre os tipos de dados e valores não nulos
2 print("\nInformações sobre o product.csv:")
3 print(product_df.info())
4
5 print("\nInformações sobre o ride_v2.csv:")
6 print(ride_v2_df.info())
7
8 print("\nInformações sobre o rideaddress_v1.csv:")
9 print(rideaddress_v1_df.info())
10
11 print("\nInformações sobre o rideestimative_v3.csv:")
12 print(rideestimative_v3_df.info())
```

```

13
14 # Contagem de valores ausentes por coluna
15 print("\nValores ausentes em product.csv:")
16 print(product_df.isnull().sum())
17
18 print("\nValores ausentes em ride_v2.csv:")
19 print(ride_v2_df.isnull().sum())
20
21 print("\nValores ausentes em rideaddress_v1.csv:")
22 print(rideaddress_v1_df.isnull().sum())
23
24 print("\nValores ausentes em rideestimative_v3.csv:")
25 print(rideestimative_v3_df.isnull().sum())

```



Informações sobre o product.csv:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 237 entries, 0 to 236

Data columns (total 4 columns):

#	Column	Non-Null Count	Dtype
0	ProductID	237 non-null	object
1	ProviderID	237 non-null	int64
2	CategoryID	237 non-null	int64
3	Description	237 non-null	object

dtypes: int64(2), object(2)

memory usage: 7.5+ KB

None

Informações sobre o ride_v2.csv:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 500000 entries, 0 to 499999

Data columns (total 15 columns):

#	Column	Non-Null Count	Dtype
0	RideID	500000 non-null	int64
1	UserID	500000 non-null	object
2	Schedule	500000 non-null	object
3	Create	500000 non-null	object
4	RideStatusID	500000 non-null	int64
5	CompanyID	500000 non-null	int64
6	ProviderID	228157 non-null	float64
7	RideProviderID	21440 non-null	object
8	price	500000 non-null	float64
9	Updated	500000 non-null	object
10	CategoryID	24714 non-null	float64
11	TotalUsers	500000 non-null	int64
12	Car	14944 non-null	object
13	RideDriverLocationID	14864 non-null	float64
14	ScheduledRide	500000 non-null	int64

dtypes: float64(4), int64(5), object(6)

memory usage: 57.2+ MB

None

Informações sobre o rideaddress_v1.csv:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1000000 entries, 0 to 999999

Data columns (total 11 columns):

#	Column	Non-Null Count	Dtype
0	RideAddressID	1000000 non-null	int64
1	Address	1000000 non-null	object
2	Street	998103 non-null	object
3	Number	784370 non-null	object
4	Neighborhood	410032 non-null	object
5	City	617798 non-null	object
6	State	768544 non-null	object
7	Lat	1000000 non-null	object
8	Lng	1000000 non-null	object
9	RideAddressTypeID	1000000 non-null	int64
10	RideID	1000000 non-null	int64

dtypes: int64(3), object(8)

```

1 # Verificação de valores duplicados
2 print("\nValores duplicados em product.csv:")
3 print(product_df.duplicated().sum())
4
5 print("\nValores duplicados em ride_v2.csv:")
6 print(ride_v2_df.duplicated().sum())
7
8 print("\nValores duplicados em rideaddress_v1.csv:")
9 print(rideaddress_v1_df.duplicated().sum())
10
11 print("\nValores duplicados em rideestimative_v3.csv:")
12 print(rideestimative_v3_df.duplicated().sum())
13
14 # Verificação de valores únicos em colunas importantes (ex: id)

```

```

15 print("\nValores únicos em 'product_id' de product.csv:")
16 print(product_df['ProductID'].nunique())
17
18 print("\nValores únicos em 'ride_id' de ride_v2.csv:")
19 print(ride_v2_df['RideID'].nunique())
20
21 # Produto mais usado em product.csv
22 produto_mais_usado_product = product_df['ProductID'].mode()[0]
23 contagem_produto_mais_usado_product = product_df['ProductID'].value_counts()[produto_mais_usado_product]
24
25 # CompanyID mais usada em ride_v2
26 company_mais_usada = ride_v2_df['CompanyID'].mode()[0]
27 contagem_company_mais_usada = ride_v2_df['CompanyID'].value_counts()[company_mais_usada]
28 print(f"\nA CompanyID mais usada em ride_v2 é '{company_mais_usada}', usada {contagem_company_mais_usada} vezes.")
29
30 #Contagem de preços zero em ride_v2
31 contagem_precos_zero = (ride_v2_df['price'] == 0).sum()
32 print(f"\nExistem {contagem_precos_zero} corridas com preço zero em ride_v2.")
33
34 # Produto mais usado em rideestimative_v3
35 produto_mais_usado = rideestimative_v3_df['ProductID'].mode()[0]
36 contagem_produto_mais_usado = rideestimative_v3_df['ProductID'].value_counts()[produto_mais_usado]
37 print(f"\nO ProductID mais usado em rideestimative_v3 é '{produto_mais_usado}', usado {contagem_produto_mais_usado} vezes.")
38
39 #Cidade mais frequente em rideaddress_v1
40 cidade_mais_frequente = rideaddress_v1_df['City'].mode()[0]
41 contagem_cidade_mais_frequente = rideaddress_v1_df['City'].value_counts()[cidade_mais_frequente]
42 print(f"\nA cidade mais frequente em rideaddress_v1 é '{cidade_mais_frequente}', que aparece {contagem_cidade_mais_frequente} vezes.

```



```

Valores duplicados em product.csv:
0

Valores duplicados em ride_v2.csv:
0

Valores duplicados em rideaddress_v1.csv:
0

Valores duplicados em rideestimative_v3.csv:
0

Valores únicos em 'product_id' de product.csv:
236

Valores únicos em 'ride_id' de ride_v2.csv:
500000

A CompanyID mais usada em ride_v2 é '40', usada 144654 vezes.

Existem 26630 corridas com preço zero em ride_v2.

O ProductID mais usado em rideestimative_v3 é 'UberX', usado 235734 vezes.

A cidade mais frequente em rideaddress_v1 é 'São Paulo', que aparece 127724 vezes.

```

Outras conclusões e inferências

✓ Planilha ride_v2.csv

- Existem 500.000 registros de corridas.
- Os RideIDs seguem uma faixa numérica específica, o que é esperado para identificadores únicos.
- As CompanyIDs variam amplamente (de 1 a 292), sugerindo que várias empresas estão envolvidas.
- A contagem de ProviderID é menor que a contagem total de corridas, o que significa que nem todas as corridas têm um ProviderID associado. Isso pode indicar valores ausentes ou corridas que não se aplicam a um provedor específico.
- Os preços das corridas variam muito (de 0 a 15254.80), com um desvio padrão alto.
- Existem 26630 corridas com preço 0, o que sugere a quantidade de corridas canceladas ou reembolsadas
- A maioria das corridas tem apenas um usuário (a mediana é 1), mas existem corridas com até 4 usuários
- A grande maioria das corridas não é agendada (a mediana é 0)

Planilha rideaddress_v1.csv:

- Existem 1.000.000 registros de endereços de corrida.
- RideAddressTypeID varia entre 1 e 2, provavelmente indicando tipos de endereços (origem e destino, por exemplo).

- Os RideID dessa tabela, correspondem aos RideID da tabela ride_v2.
- Os Id's de ambas as colunas Id, seguem uma distribuição uniforme.

Planilha rideestimative_v3.csv

- Existem 2.000.000 estimativas de corrida.
- Os RideID dessa tabela, correspondem aos RideID da tabela ride_v2
- O tempo de espera médio é de aproximadamente 6 minutos, com uma variação relativamente pequena.