

```
#Fiz o import do pandas!  
import pandas as pd
```

```
# Ler os arquivos CSV  
train_df = pd.read_csv('/content/test.csv')  
test_df = pd.read_csv('/content/test.csv')  
sample_submission_df = pd.read_csv('/content/Sample_Submission.csv')
```

```
# Aqui vou Exibir as primeiras linhas de cada arquivo para entender os dados  
print("Train Data:")  
print(train_df.head())
```

```
print("\nTest Data:")  
print(test_df.head())
```

```
print("\nSample Submission:")  
print(sample_submission_df.head())
```

```
# Aqui eu optei por fazer uma verificação mais profunda pois  
# pode haver strings vazias ou valores "estranhos" em colunas do tipo object.  
print(train_df.isin(['', 'NA', 'N/A', 'na']).sum())
```

```
# Aqui eu Converti as colunas numéricas que estavam como string.  
train_df['Delivery_person_Age'] = pd.to_numeric(train_df['Delivery_person_Age'],  
errors='coerce')  
train_df['Delivery_person_Ratings'] =  
pd.to_numeric(train_df['Delivery_person_Ratings'], errors='coerce')
```

```
#Após perceber que tinha valores nulos. Optei por preenchê-los em vez de remover.  
Então preenchi os valores nulos utilizando a mediana nas colunas afetadas.  
train_df['Delivery_person_Age'].fillna(train_df['Delivery_person_Age'].median(),  
inplace=True)  
train_df['Delivery_person_Ratings'].fillna(train_df['Delivery_person_Ratings'].median(  
, inplace=True)
```

```
#após fazer esse tratamento eu realizei outra verificação para conferir se ainda tinha  
valores nulos ou não.  
print(train_df.isnull().sum())
```

```
# Aqui estou mostrando as estatísticas descritivas para colunas numéricas  
print(train_df.describe())
```

```
# Aqui estou tarzendo a Frequência de valores nas colunas categóricas  
print(train_df['Type_of_order'].value_counts())
```

```
print(train_df['Weatherconditions'].value_counts())
```

```
# Aqui mostro o grafico da Distribuição da Idade dos Entregadores
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(10, 6))
```

```
sns.histplot(train_df['Delivery_person_Age'], bins=20, kde=True)
```

```
plt.title('Distribuição da Idade dos Entregadores')
```

```
plt.xlabel('Idade')
```

```
plt.ylabel('Frequência')
```

```
plt.show()
```

```
# Aqui apresento a Distribuição das Avaliações dos Entregadores
```

```
plt.figure(figsize=(10, 6))
```

```
sns.histplot(train_df['Delivery_person_Ratings'], bins=20, kde=True)
```

```
plt.title('Distribuição das Avaliações dos Entregadores')
```

```
plt.xlabel('Avaliações')
```

```
plt.ylabel('Frequência')
```

```
plt.show()
```

```
# Aqui optei por Exibir as colunas do DataFrame
```

```
print(train_df.columns)
```

```
# Filtrando apenas as colunas numéricas
```

```
numeric_df = train_df.select_dtypes(include=['float64', 'int64'])
```

```
# Matriz de correlação
```

```
correlation_matrix = numeric_df.corr()
```

```
# Visualizar a matriz de correlação
```

```
plt.figure(figsize=(10, 8))
```

```
sns.heatmap(correlation_matrix, annot=True, fmt='.2f', cmap='coolwarm')
```

```
plt.title('Matriz de Correlação')
```

```
plt.show()
```

```
#Aqui optei por utilizar a matriz de correlação em forma de um gráfico de calor pois  
achei mais prático e interessante a ser mostrado.
```

```
# Criar um gráfico de dispersão para visualizar a relação entre a idade dos  
entregadores
```

```
# e suas avaliações. Esse gráfico ajuda a identificar padrões ou correlações entre
```

```
# essas variáveis, o que pode ser relevante para entender como a experiência  
(idade)
```

# pode afetar a qualidade do serviço (avaliações).

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Delivery_person_Age', y='Delivery_person_Ratings',
data=train_df)
plt.title('Idade dos Entregadores vs. Avaliações')
plt.xlabel('Idade dos Entregadores')
plt.ylabel('Avaliações dos Entregadores')
plt.show()
```

# Analisando colunas categóricas

```
# Contar a frequência de valores na coluna 'Type_of_order'
# Essa contagem ajuda a entender quais tipos de pedidos são mais comuns,
# o que pode influenciar a estratégia de negócios e operações.
print(train_df['Type_of_order'].value_counts())
```

```
# Contar a frequência de condições climáticas na coluna 'Weatherconditions'
# Compreender as condições climáticas durante as entregas pode ser importante
# para avaliar como elas impactam o desempenho dos entregadores e o tempo de
entrega.
print(train_df['Weatherconditions'].value_counts())
```