data-cleaning

August 19, 2024

1 Limpeza de dados

[108]:
```
import pandas as pd
import numpy as np
```

[109]:
```
df = pd.read_csv('fake_missing_data.csv')
# reordenando os valores
df.sort_values(by=['Data'], inplace=True)
df.head(5)
```

[109]:

	Unnamed: 0	Data	Produto	Categoria	Quantidade	Preço
155	155	2023-01-02	Bread roll	bread	8.0	1.5
43	43	2023-01-03	Pie slice	slice	10.0	4.0
111	111	2023-01-03	Pie slice	slice	4.0	4.0
5	5	2023-01-08	Veggie sandwich	sandwich	3.0	2.8
193	193	2023-01-13	Veggie sandwich	sandwich	1.0	2.8

A primeira coluna (ID) está sem nome! Vamos renomea-la

[110]:
```
df.columns
```

[110]:
```
Index(['Unnamed: 0', 'Data', 'Produto', 'Categoria', 'Quantidade', 'Preço'],
dtype='object')
```

[111]:
```
# inplace=True aplica a transformação no objeto 'df'
df.rename(columns={'Unnamed: 0': 'ID'}, inplace=True)
df.head(5)
```

[111]:

	ID	Data	Produto	Categoria	Quantidade	Preço
155	155	2023-01-02	Bread roll	bread	8.0	1.5
43	43	2023-01-03	Pie slice	slice	10.0	4.0
111	111	2023-01-03	Pie slice	slice	4.0	4.0
5	5	2023-01-08	Veggie sandwich	sandwich	3.0	2.8
193	193	2023-01-13	Veggie sandwich	sandwich	1.0	2.8

[112]:
```
# como nao coloquei inplace=True, tenho que salvar a variavel nela mesmo
df = df.drop_duplicates()
df.shape # para ver se tem duplicatas. se nao tiver, o formato deve ser (200, 6)
```

[112]: (200, 6)

[113]:
```
# mostrando linhas com valores faltantes
dfnull = df[df.isnull().any(axis=1)]
display(dfnull)
```

	ID	Data	Produto	Categoria	Quantidade	Preço
105	105	2023-04-01	Veggie sandwich	sandwich	3.0	NaN
187	187	2023-04-11	Veggie sandwich	NaN	7.0	2.8
156	156	2023-08-13	NaN	ready-made	3.0	2.0
128	128	2023-09-26	Veggie sandwich	sandwich	NaN	2.8

Parece que as colunas com IDs 105, 187, 156 e 128 tem valores faltantes. As estratégias para preenchimento estão no arquivo README.md, mas vou detalha-las aqui também em comentários no código

[114]:
```
dfveg = df[df['Produto'] == 'Veggie sandwich']
dfveg.describe() # para ver o preço
```

[114]:

	ID	Quantidade	Preço
count	27.000000	26.000000	2.600000e+01
mean	95.000000	4.576923	2.800000e+00
std	60.851648	2.802471	4.528839e-16
min	5.000000	1.000000	2.800000e+00
25%	49.000000	2.250000	2.800000e+00
50%	79.000000	4.500000	2.800000e+00
75%	148.000000	6.750000	2.800000e+00
max	194.000000	10.000000	2.800000e+00

[115]:
```
# como o sanduiche sempre tem o mesmo preco, so coloco ele no lugar
df.loc[df['ID'] == 105, ['Preço']] = 2.8

# checando se funcionou ...
dfnull = df[df.isnull().any(axis=1)]
display(dfnull)
```

	ID	Data	Produto	Categoria	Quantidade	Preço
187	187	2023-04-11	Veggie sandwich	NaN	7.0	2.8
156	156	2023-08-13	NaN	ready-made	3.0	2.0
128	128	2023-09-26	Veggie sandwich	sandwich	NaN	2.8

[116]:
```
# vendo as estatisticas para preencher a quantidade
print(f"Mediana: {dfveg['Quantidade'].median()}")
print(f"Media: {dfveg['Quantidade'].mean()}")
print(f"Moda: {dfveg['Quantidade'].mode()}")
```

```
Mediana: 4.5
Media: 4.576923076923077
```

```
Moda: 0    3.0
Name: Quantidade, dtype: float64
```

[117]:
```
from math import trunc

# colocando a categoria
df.loc[df['ID'] == 187, ['Categoria']] = 'sandwich'
# na quantidade, decidi usar a media truncada
# ja que a moda e a mediana sao menores
# NOTA: poderia ter sido feito como os outros
df['Quantidade'] = df['Quantidade'].fillna(trunc(df['Quantidade'].mean()))

dfnull = df[df.isnull().any(axis=1)]
display(dfnull)
```

	ID	Data	Produto	Categoria	Quantidade	Preço
156	156	2023-08-13	NaN	ready-made	3.0	2.0

[118]:
```
# para esse ultimo vemos qual ready-made tem preco = 2.0
dfrm = df[df['Categoria'] == 'ready-made']
dfrm.head()
```

[118]:

	ID	Data	Produto	Categoria	Quantidade	Preço
154	154	2023-01-19	Mints	ready-made	3.0	1.0
66	66	2023-01-19	Chocolate bar	ready-made	3.0	2.0
57	57	2023-02-03	Chocolate bar	ready-made	8.0	2.0
118	118	2023-02-10	Chocolate bar	ready-made	2.0	2.0
82	82	2023-02-19	Mints	ready-made	6.0	1.0

[121]:
```
df.loc[df['ID'] == 156, ['Produto']] = 'Chocolate bar'

# nao deve ter valores null agora
dfnull = df[df.isnull().any(axis=1)]
display(dfnull)
```

```
Empty DataFrame
Columns: [ID, Data, Produto, Categoria, Quantidade, Preço]
Index: []
```

Salvando o dataset limpo e ordenado

[122]:
```
df.to_csv('data_clean.csv')
```