
APOSTILA

# Análise de dados com Pandas

**AULA 3**

Empowerdata

# Seja bem-vindo(a) à imersão Semana do Python na prática

É um prazer ter você aqui com a gente nessa imersão!

Essa apostila tem como objetivo trazer de uma forma simples e direta todo o conteúdo que foi passado na aula ao vivo.

Aqui na Empowerdata, acreditamos fortemente no aprendizado por projetos e por isso, cada dia da imersão traz um novo projeto a ser desenvolvido.

Lembre-se sempre na nossa metodologia ORA:

**OBSERVAR:** durante a aula ao vivo, apenas acompanhe o que está sendo feito e anote as explicações.

**REPETIR:** depois, refaça o projeto com o material em mãos e caso tenha alguma dificuldade, reveja a aula. Todas as aulas ficam gravadas e disponíveis no Youtube durante a semana da imersão.

**APLICAR:** agora que você observou e repetiu, é hora de aplicar o que aprendeu. Crie novos projetos com o conhecimento adquirido.

Empowerdata © Todos os direitos reservados

## O que vamos aprender?

- Usar a biblioteca de Análise de Dados mais utilizada no mundo: **Pandas**
- Carregar dados de um arquivo **Excel**
- Fazer análise exploratória dos dados
- Gerar estatísticas
- Gerar gráficos interativos com a biblioteca **Plotly**
- Enviar e-mail de forma automática

## Projeto da aula

Você foi contratado(a) como um(a) **Analista de Dados** por uma rede de lojas de **Açaí** e deverá **criar análises sobre o negócio** utilizando uma **base de dados em um arquivo Excel**, que foi extraída do sistema de vendas da rede.

## Bibliotecas que iremos utilizar

### Instalando as bibliotecas Pandas, OpenpyXL e Plotly Express

**TERMINAL**

```
pip install pandas
```

```
pip install openpyxl
```

```
pip install plotly nbformat
```

Empowerdata © Todos os direitos reservados

# Carregando dados do arquivo Excel

**ENTRADA**

```
import pandas as pd

dados = pd.read_excel("vendas.xlsx")
```

# Análise Exploratória

## Verificando as primeiras e últimas linhas

**ENTRADA**

```
dados.head()
```

**SAÍDA**

| | id_pedido | data | loja | cidade | estado | regiao | tamanho | local_consumo | preco | forma_pagamento | ano_mes |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | PED1994 | 2020-01-01 | Loja 4 | Santos | São Paulo | Sudeste | 300ml | Consumo no local | 5 | Dinheiro | 2020-01 |
| 1 | PED2246 | 2020-01-01 | Loja 6 | Florianópolis | Santa Catarina | Sul | 500ml | Consumo no local | 11 | Débito | 2020-01 |
| 2 | PED3876 | 2020-01-01 | Loja 3 | Rio de Janeiro | Rio de Janeiro | Sudeste | 300ml | Delivery | 7 | Crédito | 2020-01 |
| 3 | PED4352 | 2020-01-01 | Loja 1 | Fortaleza | Ceará | Nordeste | 1000ml | Consumo no local | 7 | Débito | 2020-01 |
| 4 | PED8633 | 2020-01-01 | Loja 5 | São Paulo | São Paulo | Sudeste | 200ml | Delivery | 9 | Crédito | 2020-01 |

## Verificando as últimas linhas

**ENTRADA**

```
dados.tail()
```

Empowerdata © Todos os direitos reservados

**SAÍDA**

| | id_pedido | data | loja | cidade | estado | regiao | tamanho | local_consumo | preco | forma_pagamento | ano_mes |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 69995 | PED67084 | 2022-12-31 | Loja 6 | Florianópolis | Santa Catarina | Sul | 500ml | Consumo no local | 11 | Crédito | 2022-12 |
| 69996 | PED67857 | 2022-12-31 | Loja 3 | Rio de Janeiro | Rio de Janeiro | Sudeste | 200ml | Consumo no local | 7 | Pix | 2022-12 |
| 69997 | PED69171 | 2022-12-31 | Loja 4 | Santos | São Paulo | Sudeste | 500ml | Consumo no local | 5 | Dinheiro | 2022-12 |
| 69998 | PED69229 | 2022-12-31 | Loja 4 | Santos | São Paulo | Sudeste | 300ml | Consumo no local | 9 | Pix | 2022-12 |
| 69999 | PED69356 | 2022-12-31 | Loja 1 | Fortaleza | Ceará | Nordeste | 300ml | Delivery | 9 | Pix | 2022-12 |

## Quantidade de linhas e colunas

**ENTRADA**

```
dados.shape
```

**SAÍDA**

```
(70000, 11)
```

## Informações sobre as colunas

**ENTRADA**

```
dados.info()
```

**SAÍDA**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70000 entries, 0 to 69999
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id_pedido        70000 non-null  object
 1   data             70000 non-null  datetime64[ns]
 2   loja             70000 non-null  object
 3   cidade           70000 non-null  object
 4   estado           70000 non-null  object
 5   regiao           70000 non-null  object
 6   tamanho          70000 non-null  object
 7   local_consumo    70000 non-null  object
 8   preco            70000 non-null  int64
 9   forma_pagamento  70000 non-null  object
 10  ano_mes          70000 non-null  object
dtypes: datetime64[ns](1), int64(1), object(9)
memory usage: 5.9+ MB
```

Empowerdata © Todos os direitos reservados

## Gerando as estatísticas

Agora, vamos gerar algumas estatísticas importantes sobre a coluna **preço**.

**ENTRADA**

```
dados.describe()
```

**SAÍDA**

| | preco |
|---|---|
| count | 70000.000000 |
| mean | 8.355200 |
| std | 2.653061 |
| min | 5.000000 |
| 25% | 7.000000 |
| 50% | 7.000000 |
| 75% | 11.000000 |
| max | 13.000000 |

## Obtendo os valores únicos de uma coluna

**ENTRADA**

```
dados["loja"].unique()
```

**SAÍDA**

```
array(['Loja 4', 'Loja 6', 'Loja 3', 'Loja 1', 'Loja 5', 'Loja 2'],
dtype=object)
```

Empowerdata © Todos os direitos reservados

## Contagem de valores

**ENTRADA**

```
dados["loja"].value_counts()
```

**SAÍDA**

```
loja
Loja 4    13483
Loja 6    13075
Loja 1    12344
Loja 5    12177
Loja 3    10603
Loja 2     8318
Name: count, dtype: int64
```

## Agrupando dados

O método **groupby()** realiza o agrupamento de dados por determinada coluna. Sempre que utilizamos esse método, precisamos definir a **função de agregação** que será aplicada nos dados.

Exemplo: **sum()** para soma e **mean()** para média.

**ENTRADA**

```
dados.groupby("loja").preco.sum()
```

**SAÍDA**

```
loja
Loja 1    103162
Loja 2     69592
Loja 3     88357
Loja 4    112379
Loja 5    102189
Loja 6    109185
Name: preco, dtype: int64
```

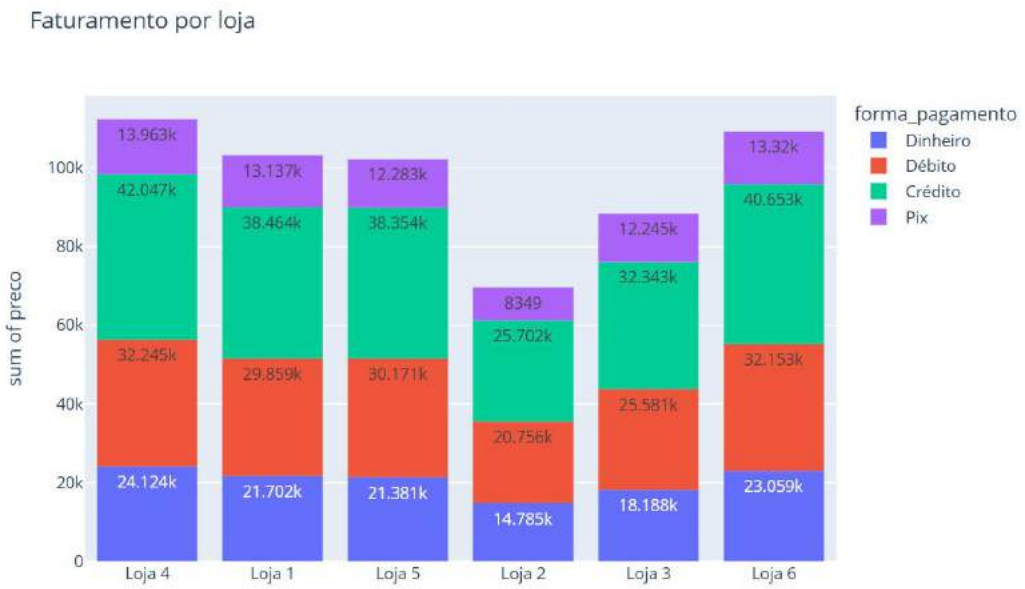
Empowerdata © Todos os direitos reservados

## Gráficos interativos

**ENTRADA**

```
grafico = px.histogram(dados, x="loja",
                                    y="preco",
                                    color="forma_pagamento",
                                    text_auto=True,
                                    title="Faturamento por loja")
grafico.show()
```

**SAÍDA**




## Listas e o comando for

**ENTRADA**

```
nomes = [“Maria”, “João”, “José”]
for nome in nomes:
   print(nome)
```

**SAÍDA**

```
Maria
João
José
```

Empowerdata © Todos os direitos reservados


## ACOMPANHE MAIS CONTEÚDOS EM

**Instagram @empowerdata**

**Instagram @empowerpython**

**Canal no Youtube**

Empowerdata