Senac

Fecomércio Sesc

Data Science – Princípios e Técnicas

Maio

2025

Data Science

Onde me encontrar:

https://www.linkedin.com/in/marco-mialaret-junior/

e

https://github.com/MatmJr

Vamos Gerar um relatório com o nosso ETL

Data Science

Nas aulas passadas aprendemos a trabalhar com o pandas, fizemos uma ETL com dados do banco central e alguns conceitos de Estatística.

Data Science

Hoje vamos organizar as ideias e montar um relatório.

A análise exploratória

Data Science

Vamos carregar o nosso conjunto de dados, execule o ETL na sua máquina e carregue os dados que estão no SQLite:

```
import pandas as pd
import sqlite3

con = sqlite3.connect("src/datasets/etlbcb.db")
query = "select * from meios_pagamentos_tri"
df = pd.read_sql(query, con)
con.close()
```

Data Science

Como obtemos as informações gerais sobre os dados?

```
df.info()
```

Data Science

Procurando dados faltantes:

```
df.isnull().sum()
```

Data Science

Note que data trimestre está como object(string), mas seria mais interessante estar como Date. Vamos comprovar isso.

```
type(df['datatrimestre'][0])
```

Data Science

Convertendo para date.

```
df['datatrimestre'] = pd.to_datetime(df['datatrimestre'])
df['datatrimestre'] = df['datatrimestre'].dt.date
```

Data Science

Ajustando as colunas que estão em milhões

```
monetary_cols = [
    'valorPix', 'valorTED', 'valorTEC', 'valorCheque', 'valorBoleto',
    'valorDOC', 'valorCartaoCredito', 'valorCartaoDebito', 'valorCartaoPrePago',
    'valorTransIntrabancaria', 'valorConvenios', 'valorDebitoDireto', 'valorSaques'
]
```

Data Science

```
for col in monetary_cols:
    df[col] = df[col] * 1_000_000
    df[col] = df[col].round(2)
```

Data Science

Ajustando as colunas que estão em milhares

```
thousands_cols = [
    'quantidadePix', 'quantidadeTED', 'quantidadeTEC', 'quantidadeCheque',
    'quantidadeBoleto', 'quantidadeDOC', 'quantidadeCartaoCredito',
    'quantidadeCartaoDebito', 'quantidadeCartaoPrePago',
    'quantidadeTransIntrabancaria', 'quantidadeConvenios',
    'quantidadeDebitoDireto', 'quantidadeSaques'
]
```

Data Science

```
for col in thousands_cols:
    df[col] = df[col] * 1_000
    df[col] = df[col].round(2)
```

Estatísticas Descritivas

Data Science

Nosso conjunto de dados está com muitas colunas, vamos nos concentrar em um tema específico para continuar com a análise.

Vamos nos concentrar nas séries de Pix e cartões.

Data Science

```
date_col = ['datatrimestre']

pix_cols = ['valorPix', 'quantidadePix']

cartao_cols = [
    'valorCartaoCredito', 'quantidadeCartaoCredito',
    'valorCartaoDebito',  'quantidadeCartaoDebito',
    'valorCartaoPrePago', 'quantidadeCartaoPrePago'
]
```

Data Science

```
selected_cols = date_col + pix_cols + cartao_cols
df_pix_cards = df[selected_cols].copy()

df_pix_cards.head()
```

Data Science

As estatísticas do nosso conjunto de dados

```
df_pix_cards.describe()
```

Data Science

As médias:

Nas aulas passadas encontramos o valor:

```
soma1 = df_pix_cards['valorPix'].sum()
soma2 = df_pix_cards['quantidadePix'].sum()

soma1/soma2
```

✓ 0.0s

```
np.float64(430.713035665615)
```

Data Science

Por outro lado:

```
(df_pix_cards['valorPix']/df_pix_cards['quantidadePix']).mean()
```

✓ 0.0s Python

```
np.float64(491.13828791552487)
```

Data Science

A diferença existe porque, na segunda abordagem, você está dando peso igual a cada trimestre, independentemente de quantas transações ocorreram naquele período. Já na primeira, cada transação — seja no trimestre que for — “conta” igualmente, o que faz o trimestre com mais transações influenciar mais o resultado final.

Data Science

Use a média geral quando quiser o valor médio de toda a amostra, e a média das médias trimestrais quando quiser tratar cada período (trimestre) com igual importância.

Data Science

Variância e Desvio-Padrão:

```
x = df_pix_cards['valorPix'] / df_pix_cards['quantidadePix']
print("Média: ", x.mean())
print("Variância: ", x.var(ddof=0))
print("Desvio-padrão: ", x.std(ddof=0))
```

✓ 0.0s

```
Média:  491.13828791552487
Variância:  14819.807350927598
Desvio-padrão:  121.73663109733076
```

Visualizações

Data Science

Para complementar o processo de análise de dados vamos gerar alguns gráficos, para isso usaremos a biblioteca matplotlib.

Data Science

Para complementar o processo de análise de dados vamos gerar alguns gráficos, para isso usaremos a biblioteca matplotlib.

```
(.venv) E:\Python\AulaETL\etlBCB>pip install matplotlib
```

Data Science

Histograma

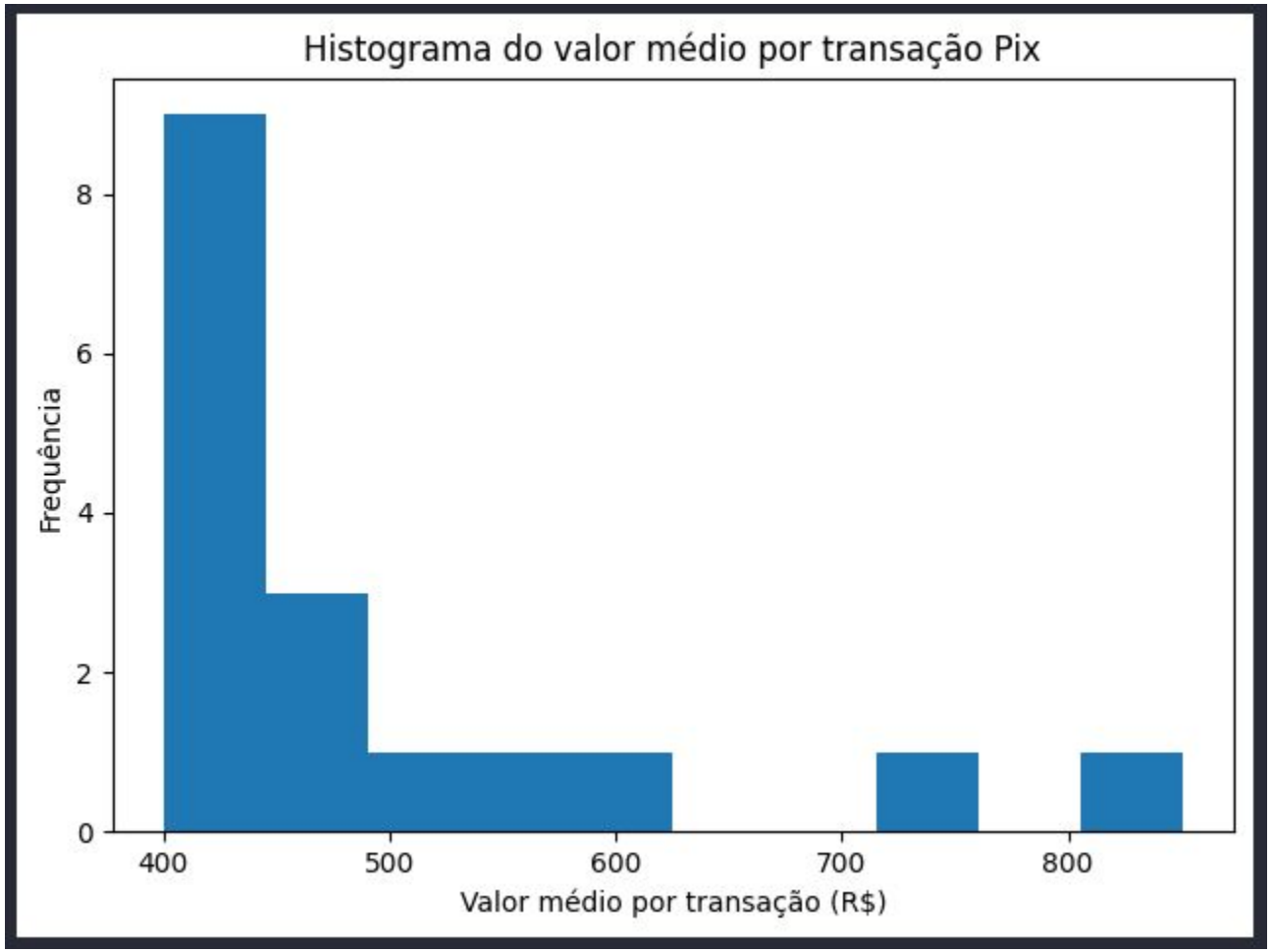
Mostra a distribuição de frequências de uma variável numérica dividida em “bins” (intervalos). Cada barra indica quantos valores caem em cada faixa. Use quando quiser entender a forma geral dos dados (simetria, sesgo, picôs múltiplos) e identificar aglomerados ou lacunas.

Data Science

```
import matplotlib.pyplot as plt

# ——— Histograma ———
plt.figure()
plt.hist(x)
plt.title('Histograma do valor médio por transação Pix')
plt.xlabel('Valor médio por transação (R$)')
plt.ylabel('Frequência')
plt.tight_layout()
plt.show()
```

Data Science



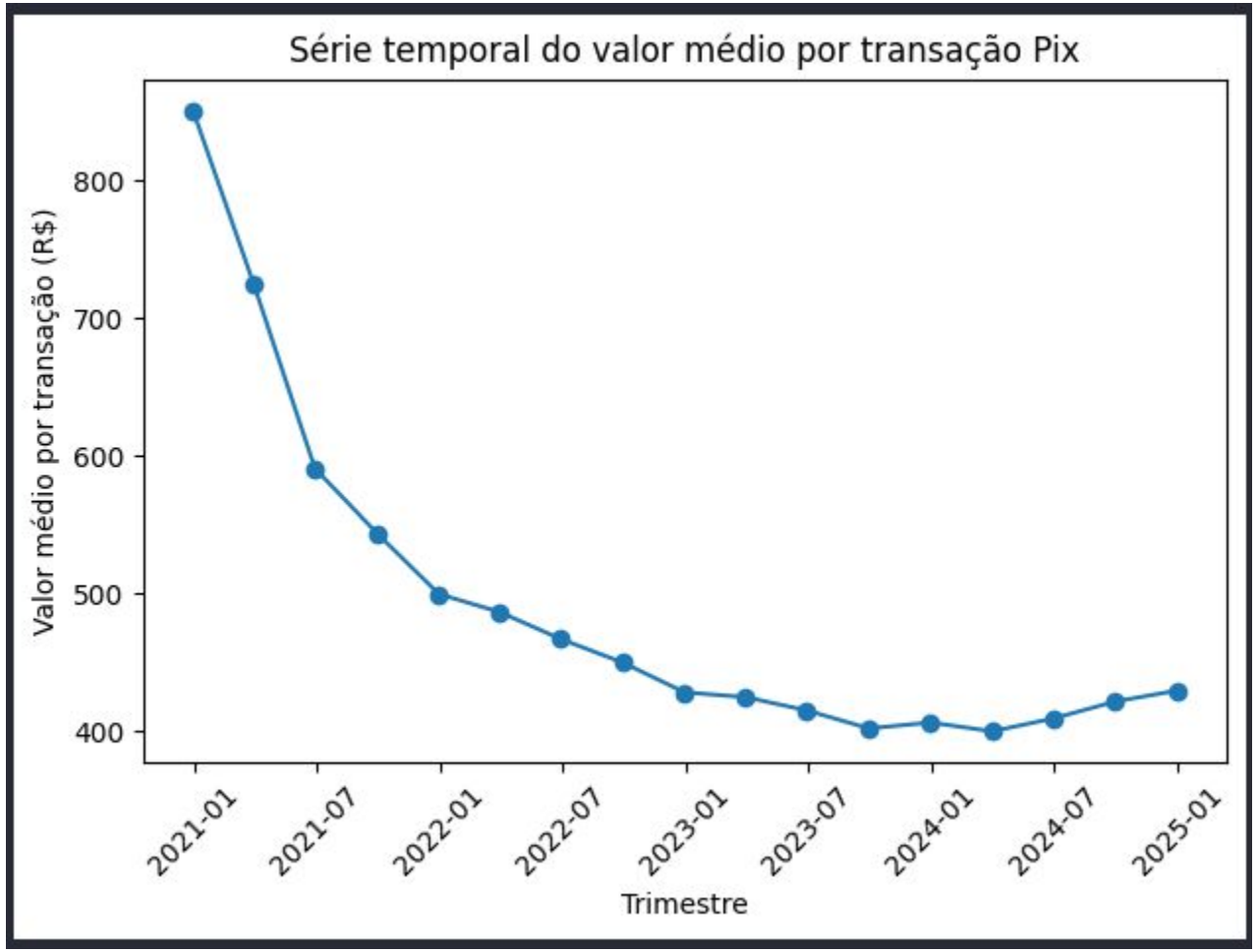
Data Science

Série temporal (line plot)
Plota um valor ao longo de um eixo temporal conectado por linhas. Ideal para acompanhar tendências, ciclos sazonais e rupturas em séries de tempo (ex.: evolução do ticket médio de Pix trimestre a trimestre). Ajuda a visualizar crescimento, quedas e padrões periódicos.

Data Science

```
# ——— Série temporal ———
plt.figure()
plt.plot(df['datatrimestre'], x, marker='o')
plt.title('Série temporal do valor médio por transação Pix')
plt.xlabel('Trimestre')
plt.ylabel('Valor médio por transação (R$)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Data Science



Data Science

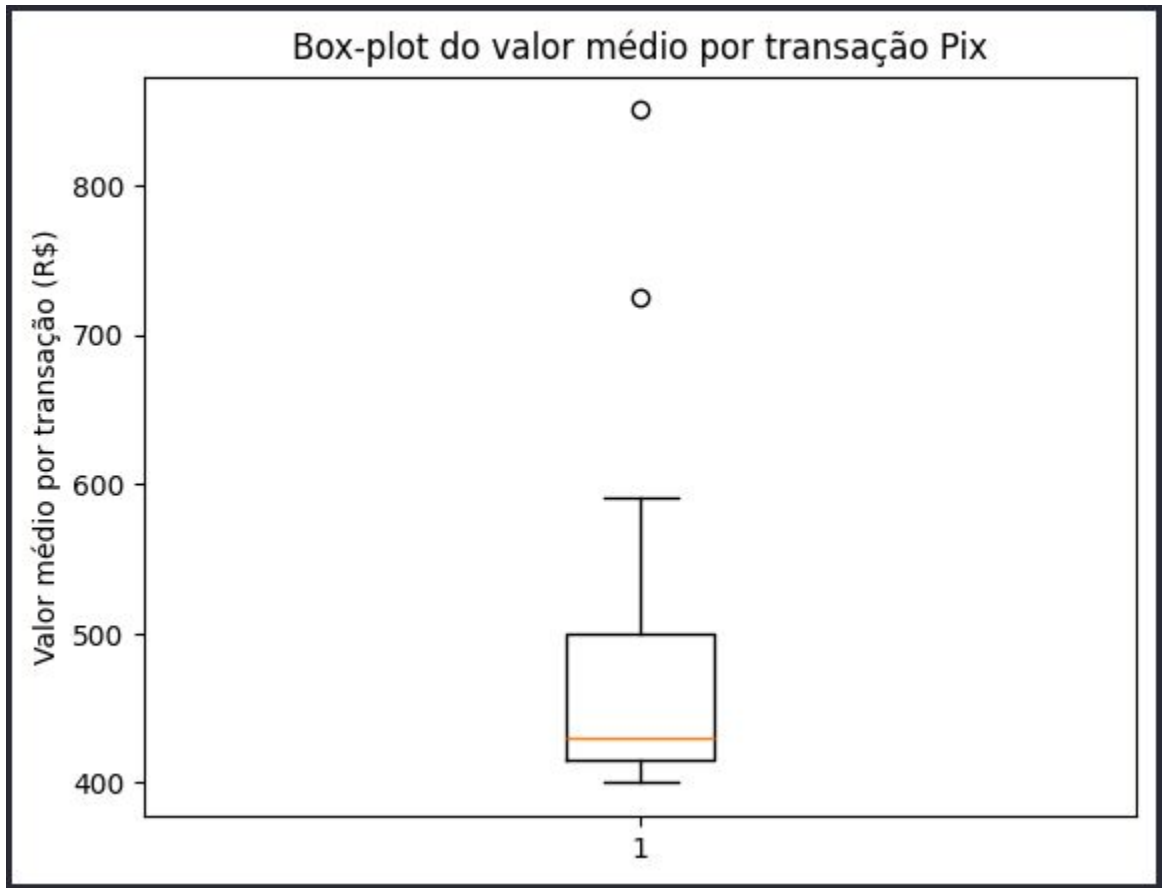
Box-plot

Representa a mediana, quartis e potenciais outliers de um conjunto de observações. A “caixa” vai do primeiro ao terceiro quartil, com uma linha na mediana, e “bigodes” indicando o alcance dos dados sem outliers. Use para comparar dispersão e simetria entre grupos (por exemplo, diferentes canais de pagamento) e para detectar valores atípicos.

Data Science

```
x_clean = (df_pix_cards['valorPix'] / df_pix_cards['quantidadePix']).dropna()
# Box-plot
plt.figure()
plt.boxplot(x_clean)
plt.title('Box-plot do valor médio por transação Pix')
plt.ylabel('Valor médio por transação (R$)')
plt.show()
```

Data Science



Dúvidas?

Marco Mialaret, MSc

Telefone:
81 98160 7018

E-mail:
marco.junior@pe.senac.br