Aula 02: DataFrame, NumPy e Quartis

A Base da Análise: Estatística

A partir de agora, ter uma boa base matemática e estatística se torna indispensável para a análise de dados. Você já usou anteriormente medidas de posicionamento como Quartis e Mediana?

- **Mediana:** É o valor central de um conjunto de dados. Se o conjunto tiver um número par de elementos, a mediana é a média dos dois valores centrais.
- **Quartil:** Indica a posição de um valor em um conjunto de dados, dividindo-o em subconjuntos.

Vamos usar a biblioteca NumPy para calcular os quartis e a mediana:

```
import numpy as np

# Dados de exemplo
dados = np.array([12, 15, 17, 20, 22, 25, 28, 30, 35, 40])

# Calcular quartis
q1 = np.percentile(dados, 25) # O 1º quartil representa 25% dos dados
q2 = np.percentile(dados, 50) # A Mediana é o mesmo que o quartil de 50%
q3 = np.percentile(dados, 75) # O 3º quartil representa 75% dos dados

# Exibir os resultados
print(f"Primeiro quartil (Q1): {q1}")
print(f"Segundo quartil (Q2, Mediana): {q2}")
print(f"Terceiro quartil (Q3): {q3}")
```

Agora, vamos aplicar esses conceitos partindo do case iniciado na aula anterior:

```
import pandas as pd

# Carregar a planilha 'Transacoes' para um DataFrame
df_transacoes = pd.read_excel('base_invest.xlsx', sheet_name='Transacoes')

# Exibir as primeiras 5 linhas para verificar os dados
print(df_transacoes.head())
```

1. Identifique variáveis:
   Neste DataFrame, a coluna operacao é uma variável **qualitativa** (compra, venda), e a coluna preco é uma variável **quantitativa** (medida).

2. Calcule os quartis:

```
# Calcule o Q1, Q2 (Mediana) e Q3 para a coluna 'preco'
q1_preco = df_transacoes['preco'].quantile(0.25)
q2_preco = df_transacoes['preco'].quantile(0.50)
q3_preco = df_transacoes['preco'].quantile(0.75)

print(f"Preço Q1: {q1_preco}")
print(f"Preço Mediana (Q2): {q2_preco}")
print(f"Preço Q3: {q3_preco}")
```

3. Debata os dados:
   Com base nos resultados, vamos refletir sobre algumas informações. Por exemplo:
   "O que o valor da mediana nos diz sobre o preço das transações?"

Estatísticas Descritivas

O DataFrame tem um método incrível chamado `.describe()` que gera um resumo estatístico completo para as colunas numéricas de uma só vez:

```
print(df_transacoes.describe())
```

Seleção de Dados: Loc, Iloc e Query

Aprender a selecionar dados de forma precisa é crucial:

- `.loc`: Para selecionar dados usando rótulos (nomes de colunas ou índices)
- `.iloc`: Para selecionar dados usando a posição numérica
- `.query`: Para filtrar dados com expressões mais complexas

```
# Selecione a linha com índice 2 (iloc)
print(df_transacoes.iloc[2])

# Selecione a linha com id_participante igual a 101 (query)
print(df_transacoes.query("id_participante == 101"))
```

Breve Introdução a Gráficos com `.plot()`

Para criar visualizações simples, o Pandas se utiliza do método `.plot()` que é associado com a MathPlotLib:

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Contar a frequência de cada tipo de operação
contagem_operacao = df_transacoes['operacao'].value_counts()

# Criar um gráfico de barras
contagem_operacao.plot(kind='bar', title='Tipos de Operação')

# Mostrar o gráfico
plt.show()
```

Onde Encontrar Mais Dados?

Bases de dados gratuitas podem ser encontradas em sites como o **Kaggle**. Ele é uma plataforma popular para ciência de dados e oferece uma grande variedade de conjuntos de dados em formato CSV.