Aula 11 - Introdução a manipulações de dados com python - Pandas

Profa. Gabrielly Queiroz

Introdução

Limpeza e transformação de dados são processos fundamentais para preparar dados brutos, tornando-os consistentes e adequados para análise e modelagem.

Importância:

- Melhora a qualidade dos dados.
- Reduz ruídos e inconsistências.
- Garante resultados mais precisos e confiáveis na análise.

https://colab.research.google.com/drive/1xnzf0DlgEi5BV8PL6oGc5FWOHfpH9gbM?usp=sharing

Pandas - Python

- O **Pandas** é uma biblioteca de código aberto do Python amplamente utilizada para análise de dados e manipulação de estruturas de dados.
- Trabalhar com dados tabulares (como planilhas) e séries temporais.


```
import pandas
```

```
import pandas as pd
```

```
import pandas
data = pandas.DataFrame({"A": [1, 2, 3]})
```

Etapas da Limpeza de Dados

Identificação de Problemas

- **Exemplo de Conjunto de Dados Problemático**

```
import pandas as pd

# Criando um exemplo de DataFrame
data = {
    'Nome': ['Ana', 'Bruno', 'Carlos', None, 'Ana'],
    'Idade': [23, None, 30, 25, 23],
    'Salário': [4000, 5000, None, 3500, 4000],
    'Cidade': ['São Paulo', 'São Paulo', 'Rio', 'Belo
Horizonte', 'São Paulo']
}
df = pd.DataFrame(data)
print("Dados Originais:")
print(df)
```

Tratamento de Valores Ausentes

Identificar Valores Ausentes

```
print("\nVerificando valores ausentes:")

print(df.isnull())

print("\nQuantidade de valores ausentes por coluna:")

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

Soluções

Remover linhas/colunas com valores ausentes.

Preencher valores ausentes com média, mediana ou valores específicos.

```
# Removendo linhas com valores ausentes

df_cleaned = df.dropna()

print("\nDados após remover linhas com valores ausentes:")

print(df_cleaned)
```

```
# Preenchendo valores ausentes na coluna 'Idade' com a média

df['Idade'] = df['Idade'].fillna(df['Idade'].mean())

print("\nDados após preencher valores ausentes em 'Idade':")

print(df)
```

Remoção de Duplicatas

```
# Verificando duplicatas
print("\nLinhas duplicadas:")
print(df.duplicated())

# Removendo duplicatas
df = df.drop_duplicates()
print("\nDados após remover duplicatas:")
print(df)
```

Correção de Inconsistências

```
# Padronizando os nomes das cidades

# Converte todos os nomes para letras minúsculas (str.lower())

df['Cidade'] = df['Cidade'].str.lower()


# Exibindo os dados padronizados

print("\nDados com padronização de cidades:")

print(df)
```

str.strip(): Remove espaços no início e no final da string.
str.lower(): Converte todas as letras para minúsculas.
str.upper(): Converte todas as letras para maiúsculas.
str.title(): Converte a primeira letra de cada palavra para maiúscula.
str.replace(old, new): Substitui ocorrências de um valor antigo (**old**) por um novo (**new**).
str.startswith(prefix): Verifica se a string começa com um prefixo específico.
str.endswith(suffix): Verifica se a string termina com um sufixo específico.
str.split(delimiter): Divide a string em partes com base em um delimitador.
str.normalize('NFKD'): Remove acentos e normaliza caracteres Unicode.

Abrir Arquivo Externo com Pandas

```
import pandas as pd

# Ler o arquivo Excel (substitua 'arquivo.xlsx' pelo nome do arquivo)

df2 = pd.read_excel('arquivo.xlsx')

# Exibir os primeiros dados

print(df2.head())
```

Transformação de Dados

```
# **1. Criando uma nova coluna com base em cálculos**

# Criar uma coluna "Salário Anual" multiplicando o salário mensal por 12

df['Salário Anual'] = df['Salário'] * 12

# **2. Filtragem de dados**

# Filtrar as linhas onde a idade é maior que 25

df_filtrado = df[df['Idade'] > 25]

# **3. Reordenando colunas**

# Alterar a ordem das colunas para: Nome, Cidade, Idade, Salário, Salário Anual

df = df[['Nome', 'Cidade', 'Idade', 'Salário', 'Salário Anual']]

# Exibindo os resultados

print("\nDataFrame após criar 'Salário Anual':")

print(df)

print("\nDataFrame filtrado (Idade > 25):")

print(df_filtrado)
```

Exercício

Você recebeu os dados de funcionários de uma empresa em formato de dicionário. Seu objetivo é usar o **Pandas** para realizar limpeza, transformação e análise desses dados.

Crie um DataFrame com os dados dos funcionários.
Calcule o salário anual:

- Adicione uma nova coluna chamada Salário Anual, que será o valor da coluna Salário multiplicado por 12.

Filtre os funcionários:

- Crie um novo DataFrame contendo apenas os funcionários com idade maior que 30.

Reorganize as colunas:

- Reordene o DataFrame para que as colunas fiquem na seguinte ordem: Nome, Departamento, Idade, Salário, Salário Anual.

Obtenha estatísticas dos salários:

- Exiba a média, o maior e o menor salário mensal.

Exiba os resultados:

- Mostre o DataFrame original com a nova coluna Salário Anual.
- Mostre o DataFrame filtrado.