escola britânica de artes criativas & tecnologia

Analista de Dados

Data Wrangling II

Módulo | Análise de Dados: Data Wrangling II

Caderno de Aula

Professor André Perez

Tópicos

1. Agregação e Ordenação;
2. Combinação;
3. Técnicas Avançadas.

Aulas

0. Estruturas de dados

- **Não estruturado**: texto, imagem, aúdio, etc.
- **Semi estruturado**: html, json, etc.
- **Estruturado**: tabelas, planilhas, etc.

1. Agregações e Ordenação

Arquivo CSV: github.csv

In [ ]:

```
%%writefile github.csv
ranking;project;language;stars;stars_today;forks
1;plow;go;1304;574;38
2;n8n;typescript;15668;280;1370
3;slides;go;3218;265;80
4;defi-developer-road-map;;636;247;49
5;pytorch-image-models;python;11065;101;1646
6;javascript-algorithms;javascript;110768;248;18331
7;paddleclas;python;1429;283;323
8;reddit_sentiment_trader;python;369;71;60
9;augly;python;2849;393;99
10;self-taught-guide-to-cloud-computing;;863;179;84
```

DataFrame: github_df

In [ ]:
```
import pandas as pd

github_df = pd.read_csv('github.csv', sep=';')
```

In [ ]:
```
github_df
```

1.1. Agregações

Uma agregação é o processo de resumir um conjunto de dados através de uma métrica agregada, como soma, média, máximo, mínimo, etc.

Exemplo: Agregação de dados com o método `describe`.

In [ ]:
```
describe_df = github_df[
    ['ranking', 'stars', 'stars_today', 'forks']
].describe().T # colunas numéricas
```

In [ ]:
```
describe_df
```

In [ ]:
```
describe_df.loc['stars', 'max']
```

Exemplo: Agregação com o método `agg`.

In [ ]:
```
sum_series = github_df[['stars', 'stars_today', 'forks']].agg('sum')
```

In [ ]:
```
sum_series
```

In [ ]:
```
sum_series.loc['stars']
```

Exemplo: Agregação com o método `agg` com múltiplas métricas.

In [ ]:
```
mean_max_df = github_df[['stars', 'stars_today', 'forks']].agg(
    ['mean', 'max']
)
```

In [ ]:
```
mean_max_df
```

In [ ]:
```
mean_max_df.loc['mean', 'stars']
```

Exemplo: Agregação por grupos com os métodos `groupby` e `agg`.

In [ ]:
```
grouped_sum_df = github_df[
    ['language', 'stars', 'stars_today', 'forks']
].groupby('language').agg('sum')
```

In [ ]:
```

```

```
grouped_sum_df
```

In [ ]:
```
grouped_sum_df.loc['python', 'stars']
```

Exemplo: Agregação por grupos com os métodos `groupby` e `agg` com multiplas métricas.

In [ ]:
```
grouped_count_sum_mean_std_df = github_df[
    ['language', 'stars', 'stars_today', 'forks']
].groupby('language').agg(['count', 'sum', 'mean', 'std'])
```

In [ ]:
```
grouped_count_sum_mean_std_df
```

In [ ]:
```
grouped_count_sum_mean_std_df.loc['python', 'stars']
```

In [ ]:
```
grouped_count_sum_mean_std_df.loc['python', 'stars'].loc['sum']
```

1.2. Ordenação

Uma ordenação é o processo de ordernar um conjunto de dados a partir de um conjunto de colunas e um critério (ascendente ou descendente).

Exemplo: Ordernação com uma coluna de referência através do método`sort_values` .

In [ ]:
```
github_df.sort_values(by=['stars'])
```

In [ ]:
```
github_df.sort_values(by=['stars'], ascending=False)
```

Exemplo: Ordernação com um conjunto de colunas de referência através do método `sort_values` .

In [ ]:
```
github_df.sort_values(by=['language', 'forks'], ascending=False)
```

Exemplo: Ordernação com a "coluna" de indices como referência através do método `sort_index` .

In [ ]:
```
github_df.sort_index()
```

In [ ]:
```
github_df.sort_index(ascending=False)
```

2. Combinação

2.1. Método `concat`

Combina dataframes baseado nas **linhas**, de maneira simples, **sem lógica de combinação**.

In [ ]:
```
primeiros_5 = github_df.query('ranking <= 5')
```

In [ ]:
```
primeiros_5
```

In [ ]:
```
ultimos_5 = github_df.query('ranking > 5')
```

In [ ]:
```
ultimos_5
```

- **Exemplo**: Concatenação com colunas iguais:

In [ ]:
```
pd.concat([primeiros_5, ultimos_5])
```

In [ ]:
```
pd.concat([ultimos_5, primeiros_5])
```

- **Exemplo**: Concatenação com colunas diferentes:

In [ ]:
```
pd.concat(
    [primeiros_5[['ranking', 'stars']], ultimos_5[['ranking', 'language']]]
)
```

2.2. Método `merge`

Combina dataframes baseado em **colunas**, com **lógica de combinação**.

In [ ]:
```
linguagem_df = pd.DataFrame({
    'language': ['c', 'go', 'python', 'javascript', 'typescript'],
    'creation_year': [1972, 2009, 1991, 1995, 2012],
    'paradigm': [
        'imperative',
        'imperative',
        'imperative, object-oriented',
        'imperative, object-oriented',
        'imperative, object-oriented'
    ]
})
```

In [ ]:
```
linguagem_df
```

- **Exemplo**: Combinação do tipo `inner`:

In [ ]:
```
pd.merge(left=github_df, right=linguagem_df, on='language', how='inner')
```

- **Exemplo**: Combinação do tipo `left` e `right`:

In [ ]:
```
pd.merge(left=github_df, right=linguagem_df, on='language', how='left')
```

In [ ]:
```
pd.merge(left=github_df, right=linguagem_df, on='language', how='right')
```

- **Exemplo**: Combinação do tipo `outer`:

In [ ]:
```
pd.merge(left=github_df, right=linguagem_df, on='language', how='outer')
```

3. Técnicas Avançadas

3.1. Gráficos

O Pandas possuí o método `plot` (documentação) para a geração de gráficos a partir de DataFrames. Por padrão, utiliza o pacote Python de geração de gráficos **MatplotLib** (documentação).

3.1.1 Pizza

Gráfico que relaciona uma variável **categórica** com uma variável **numérica**. Vamos utilizar o método `pie` (documentação) do pacote MatplotLib.

- **Exemplo**: Proporção das linguagens de programação no ranking:

In [ ]:
```
{'amount': len(github_df)*[1]}
```

In [ ]:
```
languages_df = pd.concat([
    github_df[['language']],
    pd.DataFrame({'amount': len(github_df)*[1]})
], axis=1)
```

In [ ]:
```
languages_df
```

In [ ]:
```
grouped_languages_df = languages_df.groupby('language').agg('sum')
```

In [ ]:
```
grouped_languages_df
```

In [ ]:
```
grouped_languages_df.plot.pie(y='amount', figsize=(11, 6))
```

3.1.2 Pontos

Gráfico que relaciona variáveis **numéricas**. Vamos utilizar o método `scatter` (documentação) do pacote MatplotLib.

- **Exemplo**: Relação entre o número de `stars` com o `forks`:

In [ ]:
```
github_df.plot.scatter(x='stars', y='forks')
```

In [ ]:
```
github_df.query('stars < 100000').plot.scatter(
    x='stars',
    y='forks',
    c='ranking',
    colormap='viridis'
)
```

3.2. Valores Nulos

Valores nulos em um DataFrame Pandas ocorrem quando o pacote não consegue interpretar o dado da fonte de dados, exemplos:

- Colunas categóricas com valores vazios (` `);
- Colunas numéricas com valores nulos (`None`);
- etc.

3.2.1 Identificação

- **Exemplo**: Identificaçao de linhas com algum valor nulo com os métodos `isnull` e `any` :

In [ ]:
```
github_df
```

In [ ]:
```
github_df.isnull()
```

In [ ]:
```
github_df.isnull().any()
```

In [ ]:
```
github_df.isnull().any().any()
```

In [ ]:
```
def has_null(df: pd.DataFrame) -> bool:
  return df.isnull().any().any()
```

In [ ]:
```
has_null(df=github_df)
```

3.2.2 Remoção

- **Exemplo**: Remoção de linhas com algum valor nulo com o método `dropna` :

In [ ]:
```
github_df.dropna()
```

In [ ]:
```
has_null(df=github_df.dropna())
```

3.2.3 Preenchimento

- **Exemplo**: Preenchimento valores nulos com o método `fillna` :

In [ ]:
```
github_df.fillna('')
```

In [ ]:
```
has_null(df=github_df.fillna(''))
```