Disciplina:

Python

Professor: Nelson Júnior

Pandas

- Estrutura e métodos para manipulação de dados em formato tabular.
- Implementa a infraestrutura de `data.frames` do R.
- Fornece operações para:
  - Leitura e escrita.
  - Ordenação e arranjo de disposição.
  - Seleção, fatiamento e filtragem.
  - Transformação e agregração.
  - Junções.
- http://pandas.pydata.org/.
- Primeira versão em Janeiro de 2008 por Wes McKinney.
- Escrita em Python, Cython e C.

Detalhes

- Pandas é a biblioteca para manipulação de dados no Python.
- Implementa a estrutura de *Data Frame* e métodos que atuam sobre ele.
- Contém todas as operações usuais disponíveis em instruções SQL.
- Além destas, dispõe de funções estatísticas e recursos de visualização.
- Em termos de performance computacional, aproxima do `data.table` do R e empata com o `tidyverse`.

Operações típicas de manipulação

1. Importar e/ou acessar dados.
2. Ordenar os registros da tabela.
3. Selecionar e fatiar nos índices/eixos.
4. Filtrar registros por predicado.
5. Renomear os índices/eixos.
6. Modificar a disposição do conteúdo.
7. Modificar/transformar o conteúdo.
8. Aplicar funções/calcular medidas resumo.
9. Agregar por categorias e aplicar.
10. Concatenar tabelas.
11. Juntar ou conciliar tabelas.

Series

Uma Series é como um array unidimensional, uma lista de valores.

Toda Series possui um índice, o index, que dá rótulos a cada elemento da lista.

```
'''
Created on 31 de mai de 2022

@author: Nelson
'''


import pandas as pd


notas = pd.Series([2,7,5,10,6])
print(notas)
```

Criamos uma Series notas, o index desta Series é a coluna à esquerda, que vai de 0 a 4 neste caso, que o pandas criou automaticamente, já que não especificamos uma lista de rótulos.

Podemos aqui verificar os atributos da nossa Series, comecemos pelos valores e o índice, os dois atributos fundamentais nesta estrutura:

```
notas.values
array([ 2,  7,  5, 10,  6])
notas.index
RangeIndex(start=0, stop=5, step=1)
```

PUC Minas

```
'''
Created on 31 de mai de 2022

@author: Nelson
'''


import pandas as pd
from pandas.core.indexes.range import RangeIndex


notas = pd.Series([2,7,5,10,6])
print(notas)

print(notas.values)
notas.index
RangeIndex(start=0, stop=5, step=1)

notas = pd.Series([2,7,5,10,6], index=["Wilfred", "Abbie", "Harry", "Julia", "Carrie"])
print(notas)

print("Média:", notas.mean())
print("Desvio padrão:", notas.std())
```

PUC Minas

EXERCÍCIOS

Crie Series com Função Salários utilizando a biblioteca pandas e faça:

a) Exiba a series preenchidas com no mínimo 10 elementos
b) Crie um index
c) Exiba o Index com sua media

DataFrame

Já um DataFrame é uma estrutura bidimensional de dados, como uma planilha.

```
'''
Created on 31 de mai de 2022

@author: Nelson
'''


import pandas as pd

df = pd.DataFrame({'Aluno' : ["Wilfred", "Abbie", "Harry", "Julia", "Carrie"],
                   'Faltas' : [3,4,2,1,4],
                   'Prova' : [2,7,5,10,6],
                   'Seminário': [8.5,7.5,9.0,7.5,8.0]})

print(df)
```

DataFrame

É possível acessar a lista de colunas de forma bem intuitiva.
Os nomes das colunas podem ser usadas pra acessar seus valores:

```
'''
Created on 31 de mai de 2022

@author: Nelson
'''


import pandas as pd

df = pd.DataFrame({'Aluno' : ["Wilfred", "Abbie", "Harry", "Julia", "Carrie"],
                   'Faltas' : [3,4,2,1,4],
                   'Prova' : [2,7,5,10,6],
                   'Seminário': [8.5,7.5,9.0,7.5,8.0]})

print(df)
print(df.columns)
print(df["Seminário"])
print(df.describe())
print(df.sort_values(by="Seminário"))

```

PUC Minas

EXERCÍCIOS

Crie um dataframe Pandas usando uma lista de Produtos com seu valores e quantitativo em Estoque

a) Exiba o dataFrame preenchido com no mínimo 50 itens
b) Exiba a media dos valores do produto
c) Exiba a quantidade de produtos no dataFrame
d) Ordene o DataFrame

Inserir dados SQL em um dataframe Pandas no Python.

- ✓ As linhas e colunas de dados contidos no dataframe podem ser usadas para explorar ainda mais os dados.

Pré-requisitos

- MySQL Windows ou Linux.
- Pandas : pip install pandas
- *MySQL Connector/Python :* pip install mysql-connector-python

Vantagens

Visualização de dados: Como o Pandas engloba algumas funcionalidades da biblioteca *matplotlib*, ela permite que os usuários criem visualizações simplificadas dos dados.

Tratamento flexível : Possibilita de forma simplificada a visualização dados que estamos trabalhando pode apresentar;

Combinações e operações relacionais: O Pandas disponibiliza métodos para facilitar a combinação de conjuntos de dados.

```
import mysql.connector
from banco.Professor import Professor


con = mysql.connector.connect(host='localhost', database='puc', user='root', password='')
if con.is_connected():
    db_info = con.get_server_info()
    print("Conectado ao servidor MySQL versão ", db_info)
    cursor = con.cursor()


def fecha_conexao():
    if con.is_connected():
        cursor.close()
        con.close()
        print("Conexão ao MySQL foi encerrada")



def incluir_professor(professor):
    url = "INSERT INTO professor (id, nome_professor, curso, turma) VALUES ( " + str(professor.id) + ",' " + professor.nome + "', '" + professor.curso + "', '" + professor.turma + "'" + " )"
    cursor.execute(url)
    con.commit()


  professor = Professor()
  professor.nome = "Girafalis"
  professor.turma = "202201"
  professor.curso = "PUC"
  professor.id = 2
  incluir_professor(professor)
  fecha_conexao()
```

```
'''
Created on 10 de jun. de 2022


@author: nelsonjunior
'''



class Professor:


    def __init__(self):
        self.id = 0
        self.nome = ""
        self.curso = ""
        self.turma = ""

    def __str__(self):
        return {'Nome': self.nome,
                'Curso': self.faltas,
                'Turma': self.notas}
```

PUC Minas

```
'''
Created on 14 de jun. de 2022


@author: nelsonjunior
'''
import pandas as pd
import mysql.connector


conn = mysql.connector.connect(host='localhost', database='puc', user='root', password='')


sql_query = pd.read_sql_query ('''
                               SELECT
                               *
                               FROM professor
                               ''', conn)


df = pd.DataFrame(sql_query, columns = ['nome_professor', 'turma', 'curso'])
file_name = 'Exemplo Pandas SQl.xlsx'
df.to_excel(file_name)
print("ok")
```

JSON

Significa JavaScript Object Notation.

- Baseado no formato de objetos em JavaScript
- Técnica de codificação para representar dados estruturados.
- Amplamente utilizado atualmente, especialmente para compartilhar dados entre servidores e aplicativos da web.
- Um arquivo CSV é usado para armazenar dados em um formato tabular, como planilhas do Excel.

```
import pandas as pd

def converte_csv_json(nomeArquivo):
    df = pd.read_csv (r'ArquivoCSV.csv')
    df.to_json (r'ArquivoJson.json')


def le_arquivo(nomeArquivo):
    arquivo = open(nomeArquivo, "r")
    tamanho = arquivo.readlines()
    cont = 0
    dados = []
    while(len(tamanho) > cont):
        linha = tamanho[cont]
        print(linha)
        dados.append(linha)
        cont += 1

    arquivo.close()
    return linha

def grava_arquivo(nomeArquivo, dados):
    arquivo = open(nomeArquivo + ".txt", "w+")

    for dado in dados:
        arquivo.write(dado)
        print(dado)

    arquivo.close()

nomeArquivo = "ArquivoCSV.csv"
dados = le_arquivo(nomeArquivo)
converte_csv_json("")
```

PUC Minas

EXERCÍCIOS