

 Estácio	Universidade Estácio Campus Belém Curso de Desenvolvimento Full Stack Relatório da Missão Prática 3 - Mundo 5
Disciplina:	RPG 0033 - Tratando a Imensidão dos Dados
Nome:	Cleyton Isamu Muto
Turma:	2023.1

Missão Prática: “Tratando a Imensidão dos Dados”

Objetivos da Prática

- Descrever como ler um arquivo CSV usando a biblioteca Pandas (Python);
- Descrever como criar um subconjunto de dados a partir de um conjunto existente usando a biblioteca Pandas (Python);
- Descrever como configurar o número máximo de linhas a serem exibidas na visualização de um conjunto de dados usando a biblioteca Pandas (Python);
- Descrever como exibir as primeiras e últimas “N” linhas de um conjunto de dados usando a biblioteca Pandas (Python);
- Descrever como exibir informações gerais sobre as colunas, linhas e dados de um conjunto de dados usando a biblioteca Pandas (Python);

Foi instalada e utilizada a ferramenta Jupyter Lab e Jupyter Notebook, conforme a documentação: <https://jupyter.org/install>

A classe auxiliar que atende às micro atividades é denominada “microatividades.ipynb”

1. Microatividade 1: Descrever como ler um arquivo CSV usando a biblioteca Pandas (Python)
 - criar um novo arquivo.
 - importar a biblioteca pandas.
 - criar a variável para receber o conteúdo do arquivo CSV (picoweb.csv), passando como parâmetros o separador de colunas (;), a engine - com o valor 'python' e o *encoding* (UTF-8) relativo aos dados constantes no arquivo lido (esse último parâmetro pode ser opcional, dependendo do encoding existente).
 - atribuir os dados lidos do CSV à variável criada anteriormente; salvar as alterações; imprimir/exibir em tela os dados da variável.
2. Microatividade 2: Descrever como criar um subconjunto de dados a partir de um conjunto existente usando a biblioteca Pandas (Python)
 - no mesmo arquivo/script utilizado na microatividade 1, criar uma nova variável (subset)
 - atribuir, a essa nova variável (subset), um subconjunto de dados contendo apenas parte das colunas (recomenda-se a utilização de 3 colunas) disponíveis no conjunto de dados original;
 - salve as alterações realizadas;
 - imprimir / exibir em tela os dados da nova variável (que contém o subconjunto de dados).

3. Microatividade 3: Descrever como configurar o número máximo de linhas a serem exibidas na visualização de um conjunto de dados usando a biblioteca Pandas (Python)

- abrir o arquivo/script utilizado nas micro atividades anteriores;
- usando as opções de configuração da biblioteca pandas, defina um novo valor para a propriedade "max_rows", definindo o novo valor para 9999;
- salvar as alterações;
- imprimir na tela o conjunto de dados original (criado na microatividade 1)
- usando o método "to_string()".

4. Microatividade 4: Descrever como exibir as primeiras e últimas "N" linhas de um conjunto de dados usando a biblioteca Pandas (Python)

- abrir o arquivo/script utilizado nas micro atividades anteriores;
- imprimir na tela as apenas as primeiras 10 linhas do conjunto de dados original (criado na microatividade 1);
- imprimir na tela as apenas as últimas 10 linhas do conjunto de dados original (criado na microatividade 1).

5. Microatividade 5: Descrever como exibir informações gerais sobre as colunas, linhas e dados de um conjunto de dados usando a biblioteca Pandas (Python)

- abrir o arquivo/script utilizado nas micro atividades anteriores;
- tendo como base o conjunto de dados original:
 - imprimir as informações gerais sobre o conjunto – suas colunas, linhas e dados;
 - descobrir a partir do comando acima:
 - o total de linhas = 32 linhas (RangeIndex: 32 entries, 0 to 31)
 - o total de colunas = 6 colunas Data columns (total 6 columns);
 - a quantidade de dados nulos, caso existam = 1 data nula e 1 caloria nula;
 - o tipo de dado de cada coluna (int64, int64, object, int64, int64, float64);
 - a quantidade de memória utilizada pelo conjunto de dados (1.6+ KB).

6. Missão Prática | Tratando a imensidão dos dados

A classe principal que atende à Missão Prática é denominada "main.ipynb"

1. para essa atividade você deverá, obrigatoriamente, utilizar o conjunto de dados (fornecido anteriormente, na seção "Contextualização") composto pelas colunas ID; Duration; Date; Pulse; Maxpulse; Calories.
2. criar um novo arquivo / script (main.ipynb).
3. ler o conteúdo do CSV fornecido, atentando-se para a necessidade ou não de incluir parâmetros adicionais como os relativos ao separador dos dados, a engine e o encoding.
4. atribuir os dados lidos a uma variável (data).
5. verificar se os dados foram importados adequadamente:
 - a. imprimir as informações gerais sobre o conjunto de dados.
 - b. imprimir as primeiras e últimas N linhas do arquivo (N = 10).

6. criar uma nova variável (`data_copy`) e atribuir a ela uma cópia do conjunto de dados original (variável criada no passo 4).
7. nessa nova variável, contendo uma cópia dos dados:
 - a. substituir todos os valores nulos da coluna 'Calories' por 0;
 - b. imprimir o conjunto de dados para verificar se a mudança acima foi aplicada com sucesso;
8. ainda na nova variável:
 - a. substituir os valores nulos da coluna 'Date' por '1900/01/01'.
 - b. imprimir o conjunto de dados e conferir se a mudança foi aplicada com sucesso.
 - c. transformar os dados da coluna 'Date' em datetime usando o método 'to_datetime'.
9. Tendo seguido todas as instruções anteriores, ao executar o passo anterior você deverá ter encontrado um erro informando que o valor '1900/01/01' não corresponde ao formato '%Y/%m/%d'. Para resolver esse problema:
 - a. substituir, na coluna 'Date', o valor '1900/01/01' por 'NaN' **[professor(a), neste ponto há um erro no enunciado da atividade: o correto para datas é 'NaT' (Not a Time) e não 'NaN' (Not a Number)]**.
 - b. utilizando o método 'to_datetime', repita o passo de transformação dos dados da coluna 'Date' para datetime.
 - c. imprimir o conjunto de dados para verificar se as mudanças acima foram aplicadas com sucesso.
10. Nesse ponto, você deverá ter esbarrado em outro erro, informando agora que o valor "20201226" não corresponde ao formato "%Y/%m/%d". Você precisará, agora, na coluna 'Date', transformar especificamente esse valor, atualmente uma string, para o formato datetime. Para isso você deverá combinar os métodos 'replace' e 'to_datetime'.
11. Após o passo anterior, executar novamente a transformação de todos os dados da coluna 'Date' para o formato datetime (usando o `to_datetime`). Imprimir o conjunto de dados atual para verificar se todas as transformações foram executadas com sucesso.
12. Por fim, remover os registros contendo valores nulos. Nesse ponto, apenas a coluna 'Date' possui um registro que atende a essa premissa (linha 22). Logo, utilizá-la como base para realizar a transformação solicitada.
13. Imprimir o dataframe e verificar se todas as transformações foram executadas conforme solicitado nos passos anteriores.