# MAP 2112 – Introdução à Lógica de Programação e Modelagem Computacional

**1º Semestre - 2023**

**Prof. Dr. Luis Carlos de Castro Santos**

lsantos@ime.usp.br


# pandas (software)

Artigo Discussão

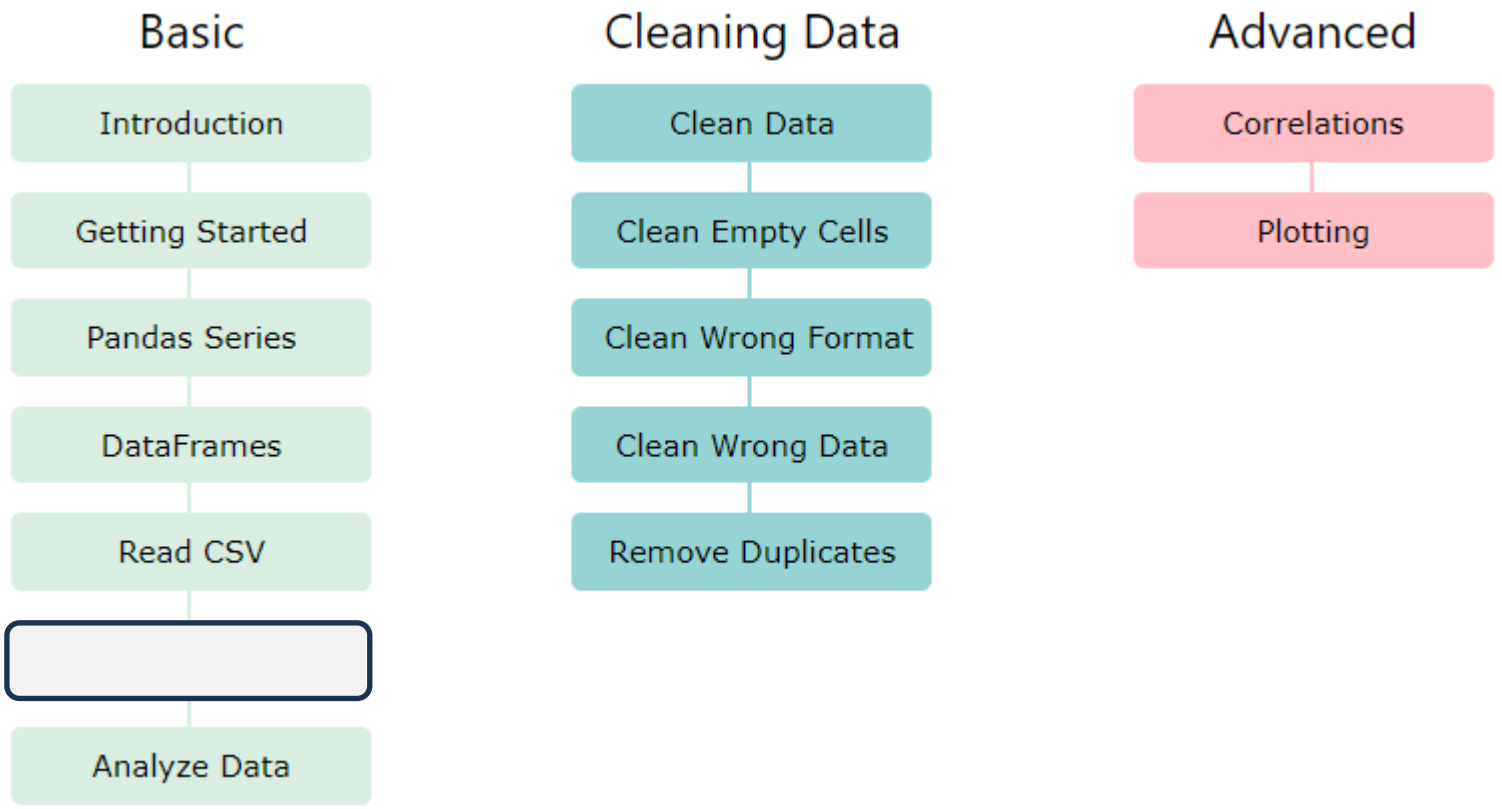
Origem: Wikipédia, a enciclopédia livre.

Em programação de computadores, **pandas** é uma biblioteca de software criada para a linguagem Python para manipulação e análise de dados. Em particular, oferece estruturas e operações para manipular tabelas numéricas e séries temporais. É software livre sob a licensa licença BSD.[2] O nome é derivado do termo inglês "panel data"(dados em painel), um termo usado em estatística e econometria para conjunto de dados que incluem várias unidades amostrais (indivíduos, empresas, etc) acompanhadas ao longo do tempo.[3]

Vamos seguir o roteiro do tutorial disponível em:

https://www.w3schools.com/python/pandas/default.asp

O roteiro será adaptado para um subconjunto do material



https://pandas.pydata.org/

pandas About us Getting started Documentation Community Contribute

# pandas

**pandas** is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,
built on top of the Python programming language.

Install pandas now!

## Getting started

- Install pandas
- Getting started

## Documentation

- User guide
- API reference
- Contributing to pandas
- Release notes

## Community

- About pandas
- Ask a question
- Ecosystem

## With the support of:

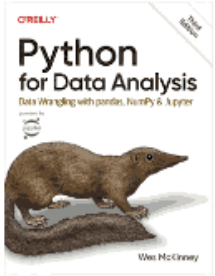
NUMFOCUS TWO SIGMA VOLTRON DATA Coiled Quansight Labs NVIDIA

## Latest version: 2.0.2

- What's new in 2.0.2
- Release date: May 28, 2023
- Documentation (web)
- Download source code

## Follow us

## Get the book




## Previous versions

- 2.0.1 (Apr 24, 2023) changelog | docs | code
- 2.0.0 (Apr 03, 2023) changelog | docs | code
- 1.5.3 (Jan 19, 2023) changelog | docs | code
- 1.5.2 (Nov 22, 2022) changelog | docs | code

Pode não ser necessário instalar – próximo slide

# PRIMEIROS PASSOS



pandas_01.py

```
1   import pandas
2
3   mydataset = {
4     'cars': ["BMW", "Volvo", "Ford"],
5     'passings': [3, 7, 2]
6   }
7
8   myvar = pandas.DataFrame(mydataset)
9
10  print(myvar)
11
```

Pandas é um módulo (incluindo métodos e objetos)

Um dicionário foi criado com os dados

O método “Dataframe” transforma o objeto dicionário num objeto dataframe.

```
In [1]: runfile('C:/Users/win/Dropbox/USP/20
pandas/pandas_01.py', wdir='C:/Users/win/Dro
MAP2112_2023/scripts/pandas')
    cars  passings
0    BMW         3
1  Volvo         7
2   Ford         2
```

O objeto dataframe é uma tabela de dados, cada coluna é chamada de série.

Se esse script rodar sua instalação está ok

# SÉRIES

pandas_02.py

```
1  import pandas as pd
2
3  a = [1, 7, 2]
4
5  myvar = pd.Series(a)
6
7  print(myvar)
8
9  print(myvar[0])
```

Uma prática em python é usar acrônimos para encurtar a escrita dos métodos.

Uma lista foi criada com os dados

O método “Series” transforma o objeto lista num objeto series.

```
In [2]: runfile('C:/Users/win/Dropbox/USP/
pandas/pandas_02.py', wdir='C:/Users/win/D
MAP2112_2023/scripts/pandas')
0    1
1    7
2    2
dtype: int64
1
```

A serie é criada com os dados sendo as linhas indexadas a partir do 0.

```
In [3]: type(myvar)
Out[3]: pandas.core.series.Series
```

myvar é objeto serie associado ao pandas.

# SÉRIES (II)

pandas_04.py

```
1  import pandas as pd
2
3  a = [1, 7, 2]
4
5  myvar = pd.Series(a, index = ["x", "y", "z"])
6
7  print(myvar)
8
9  print(myvar["y"])
```

A opção “index” do método “Series” define os rótulos das linhas.

```
In [4]: runfile('C:/Users/win/Dropbox
pandas/pandas_04.py', wdir='C:/Users/
MAP2112_2023/scripts/pandas')
x    1
y    7
z    2
dtype: int64
7
```

Com a série está indexada por strings o valor pode ser localizado usando a string.

# SÉRIES (III)

pandas_05.py

```
1  import pandas as pd
2
3  calories = {"day1": 420, "day2": 380, "day3": 390}
4
5  myvar = pd.Series(calories, index = ["day1", "day2"])
6
7  print(myvar)
```

Um dicionário foi criado com os dados

A opção “index” do método “Series” no caso de dicionários selecionas as chaves que serão usadas como rótulos das linhas.

```
In [8]: runfile('C:/Users/win/Drop
pandas/pandas_05.py', wdir='C:/Use
MAP2112_2023/scripts/pandas')
day1    420
day2    380
dtype: int64
```

# DATAFRAMES

pandas_06.py

```
1   import pandas as pd
2
3   data = {
4     "calories": [420, 380, 390],
5     "duration": [50, 40, 45]
6   }
7
8   myvar = pd.DataFrame(data)
9
10  print(myvar)
```

Um dicionário foi criado com os dados

O método “DataFrame” aplicado a dicionários selecionas as chaves como rótulo das colunas.

```
In [9]: runfile('C:/Users/win/Dropbox/
pandas/pandas_06.py', wdir='C:/Users/
MAP2112_2023/scripts/pandas')
   calories  duration
0       420        50
1       380        40
2       390        45
```

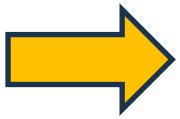
Um dataframe é uma estrutura tabular de linhas e colunas.

# DATAFRAMES (II)



pandas_07.py

```python
import pandas as pd

data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}

#Carregando os dados num DataFrame:
df = pd.DataFrame(data)

print(df)
print()

#Extraindo uma serie por índice:
print(df.loc[0])
print()
#Extraindo um sub-dataframe por índice:
print(df.loc[[0, 1]])
```



```
In [12]: runfile('C:/Users/win/Drc
pandas/pandas_07.py', wdir='C:/Use
MAP2112_2023/scripts/pandas')
   calories  duration
0       420        50
1       380        40
2       390        45

calories    420
duration     50
Name: 0, dtype: int64

   calories  duration
0       420        50
1       380        40
```

O uso de colchetes [] extrai um dataframe

# DATAFRAMES (II)



pandas_08.py

```
1   import pandas as pd
2
3   data = {
4     "calories": [420, 380, 390],
5     "duration": [50, 40, 45]
6   }
7
8   df = pd.DataFrame(data, index = ["day1", "day2", "day3"])
9
10  print(df)
11  print()
12
13  #Extraindo uma série por nome:
14  print(df.loc["day2"])
15  print()
16
17  #Extraindo uma sub-dataframe por nome:
18  print(df.loc[["day2"]])
```

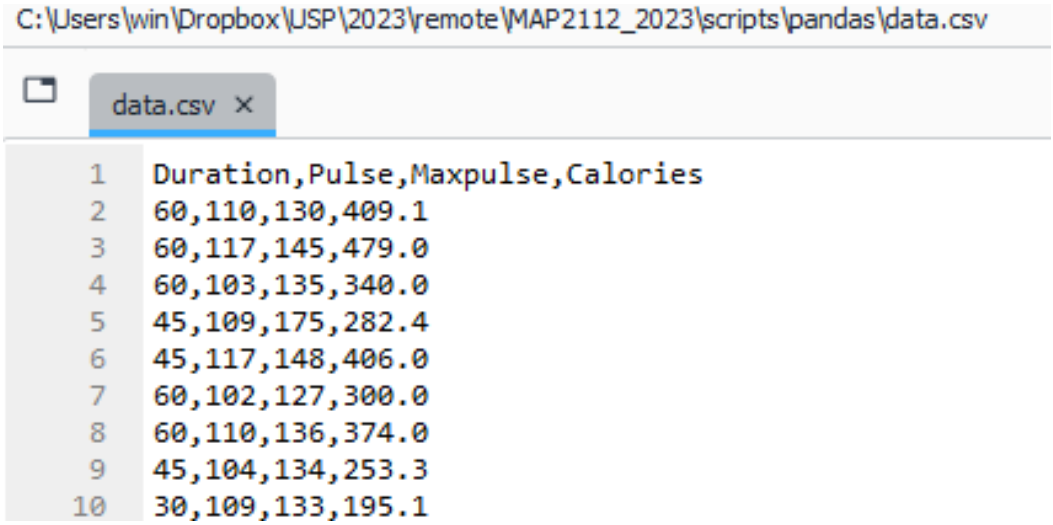
Os índices das colunas são incluídos pela opção “index”.

```
In [17]: runfile('C:/Users/win/Dropbox/U
pandas/pandas_08.py', wdir='C:/Users/win
MAP2112_2023/scripts/pandas')
      calories  duration
day1       420        50
day2       380        40
day3       390        45

calories    380
duration     40
Name: day2, dtype: int64

      calories  duration
day2       380        40
```

O uso de colchetes [] extrai um dataframe

# LEITURA DE ARQUIVOS

Um formato amplamente utilizado para armazenar dados é o CSV (“comma separated values), literalmente valores separados por vírgula. Esse formato é uma opção de armazenamento comum do Excel.




Apresentando um arquivo de exemplo data.csv aberto no spyder.

Esse mesmo arquivo estará disponível no e-disciplinas.

# LEITURA DE ARQUIVOS (II)



pandas_09.py

```
1  import pandas as pd
2
3  df = pd.read_csv('data.csv')
4
5  print(df.to_string())
```

O método “read_csv” lê o arquivo, e se o caminho não for declarado ele procura o arquivo onde o script python está.

O objeto df será um dataframe onde as colunas correspondem aos valores que estavam entre vírgulas.

O método “to_string” imprime o conteúdo do dataframe como string.

A saída será o conteúdo completo, se o dataframe for grande é de pouca utilidade apresentar tudo.

```
In [18]: runfile('C:/Users/win/Dropbox/USP/2023/remote/MAP2112_2023/
pandas/pandas_09.py', wdir='C:/Users/win/Dropbox/USP/2023/remote/
MAP2112_2023/scripts/pandas')
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
5          60    102       127     300.0
6          60    110       136     374.0
7          45    104       134     253.3
8          30    109       133     195.1
9          60     98       124     269.0
10         60    103       147     329.3
11         60    100       120     250.7
12         60    106       128     345.3
13         60    104       132     379.3
⋮
```

# LEITURA DE ARQUIVOS (III)



pandas_10.py

```
1  import pandas as pd
2
3  df = pd.read_csv('data.csv')
4
5  print(df)
6
```

```
In [19]: runfile('C:/Users/win/Dropbox/USP/2023/remote
pandas/pandas_10.py', wdir='C:/Users/win/Dropbox/USP/2
MAP2112_2023/scripts/pandas')
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
..        ...    ...       ...       ...
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4

[169 rows x 4 columns]


In [20]: print(pd.options.display.max_rows)
60
```

A saída da impressão direta apresenta o início (5 linhas) e o final (5 linhas) do dataframe e um resumo da estrutura.

A opção default de máximo no. de linhas é 60. Se o dataframe for menor que 60 a impressa direta apresenta o dataframe completo, se for maior faz a quebra das extremidades. Essa opção pode ser modificada.

# ANALISANDO DATAFRAMES



pandas_11.py

```
1   import pandas as pd
2
3   df = pd.read_csv('data.csv')
4
5   print(df.head())
6   print()
7
8   print(df.tail())
9   print()
10
11  print(df.info())
```

O default de head e tail são 5 linhas

```
In [22]: runfile('C:/Users/win/Dropbox/USP/
pandas/pandas_11.py', wdir='C:/Users/win/Dr
MAP2112_2023/scripts/pandas')
   Duration  Pulse  Maxpulse  Calories
0        60    110       130     409.1
1        60    117       145     479.0
2        60    103       135     340.0
3        45    109       175     282.4
4        45    117       148     406.0

     Duration  Pulse  Maxpulse  Calories
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Duration  169 non-null    int64
 1   Pulse     169 non-null    int64
 2   Maxpulse  169 non-null    int64
 3   Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 5.4 KB
None
```

O método info apresenta um resumo das informações do dataframe

A análise dessas informações indica (ou não) a necessidade de limpeza dos dados

# LIMPEZA DOS DADOS

O conjunto de dados podem conter algumas anomalias que atrapalham a avaliação da informação.

Alguns exemplos:

- Dados ausentes
- Dados em formatos inadequados
- Dados inválidos
- Dados duplicados

As técnicas mais comuns de lidar com essas anomalias serão apresentadas usando um arquivo exemplo.

# LIMPEZA DOS DADOS (II)



data2.csv

```
,Duration,Date,Pulse,Maxpulse,Calories
0,60,"2020/12/01",110,130,409.1
1,60,"2020/12/02",117,145,479.0
2,60,"2020/12/03",103,135,340.0
3,45,"2020/12/04",109,175,282.4
4,45,"2020/12/05",117,148,406.0
5,60,"2020/12/06",102,127,300.0
6,60,"2020/12/07",110,136,374.0
7,450,"2020/12/08",104,134,253.3
8,30,"2020/12/09",109,133,195.1
9,60,"2020/12/10",98,124,269.0
10,60,"2020/12/11",103,147,329.3
11,60,"2020/12/12",100,120,250.7
12,60,"2020/12/12",100,120,250.7
13,60,"2020/12/13",106,128,345.3
14,60,"2020/12/14",104,132,379.3
15,60,"2020/12/15",98,123,275.0
16,60,"2020/12/16",98,120,215.2
17,60,"2020/12/17",100,120,300.0
18,45,"2020/12/18",90,112,NaN
19,60,"2020/12/19",103,123,323.0
20,45,"2020/12/20",97,125,243.0
21,60,"2020/12/21",108,131,364.2
22,45,NaN,100,119,282.0
23,60,"2020/12/23",130,101,300.0
24,45,"2020/12/24",105,132,246.0
25,60,"2020/12/25",102,126,334.5
26,60,2020/12/26,100,120,250.0
27,60,"2020/12/27",92,118,241.0
28,60,"2020/12/28",103,132,NaN
29,60,"2020/12/29",100,132,280.0
30,60,"2020/12/30",102,129,380.3
31,60,"2020/12/31",92,115,243.0
```

O arquivo tem vários problemas:

- Alguns NaN (“Not a Number”)
- Uma data no formato incorreto
- Um valor que parece incoerente
- Uma linha duplicada

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  32 non-null     int64
 1   Duration    32 non-null     int64
 2   Date        31 non-null     object
 3   Pulse       32 non-null     int64
 4   Maxpulse    32 non-null     int64
 5   Calories    30 non-null     float64
dtypes: float64(1), int64(4), object(1)
memory usage: 1.6+ KB
None
```

# LIMPEZA DOS DADOS (III) – DADOS FALTANTES



data2.csv × pandas_13.py ×

```
1  import pandas as pd
2
3  df = pd.read_csv('data2.csv')
4
5  new_df = df.dropna()
6
7  print(new_df.to_string())
```

O método *dropna* remove as linhas que contém NA (ou NaN)

| | Unnamed: 0 | Duration | Date | Pulse | Maxpulse | Calories |
|---|---|---|---|---|---|---|
| 0 | 0 | 60 | 2020/12/01 | 110 | 130 | 409.1 |
| 1 | 1 | 60 | 2020/12/02 | 117 | 145 | 479.0 |
| 2 | 2 | 60 | 2020/12/03 | 103 | 135 | 340.0 |
| 3 | 3 | 45 | 2020/12/04 | 109 | 175 | 282.4 |
| 4 | 4 | 45 | 2020/12/05 | 117 | 148 | 406.0 |
| 5 | 5 | 60 | 2020/12/06 | 102 | 127 | 300.0 |
| 6 | 6 | 60 | 2020/12/07 | 110 | 136 | 374.0 |
| 7 | 7 | 450 | 2020/12/08 | 104 | 134 | 253.3 |
| 8 | 8 | 30 | 2020/12/09 | 109 | 133 | 195.1 |
| 9 | 9 | 60 | 2020/12/10 | 98 | 124 | 269.0 |
| 10 | 10 | 60 | 2020/12/11 | 103 | 147 | 329.3 |
| 11 | 11 | 60 | 2020/12/12 | 100 | 120 | 250.7 |
| 12 | 12 | 60 | 2020/12/12 | 100 | 120 | 250.7 |
| 13 | 13 | 60 | 2020/12/13 | 106 | 128 | 345.3 |
| 14 | 14 | 60 | 2020/12/14 | 104 | 132 | 379.3 |
| 15 | 15 | 60 | 2020/12/15 | 98 | 123 | 275.0 |
| 16 | 16 | 60 | 2020/12/16 | 98 | 120 | 215.2 |
| 17 | 17 | 60 | 2020/12/17 | 100 | 120 | 300.0 |
| 19 | 19 | 60 | 2020/12/19 | 103 | 123 | 323.0 |
| 20 | 20 | 45 | 2020/12/20 | 97 | 125 | 243.0 |
| 21 | 21 | 60 | 2020/12/21 | 108 | 131 | 364.2 |
| 23 | 23 | 60 | 2020/12/23 | 130 | 101 | 300.0 |
| 24 | 24 | 45 | 2020/12/24 | 105 | 132 | 246.0 |
| 25 | 25 | 60 | 2020/12/25 | 102 | 126 | 334.5 |
| 26 | 26 | 60 | 2020/12/26 | 100 | 120 | 250.0 |
| 27 | 27 | 60 | 2020/12/27 | 92 | 118 | 241.0 |
| 29 | 29 | 60 | 2020/12/29 | 100 | 132 | 280.0 |
| 30 | 30 | 60 | 2020/12/30 | 102 | 129 | 380.3 |
| 31 | 31 | 60 | 2020/12/31 | 92 | 115 | 243.0 |

# LIMPEZA DOS DADOS (IV) – DADOS FALTANTES



data2.csv × pandas_14.py ×

```
1  import pandas as pd
2
3  df = pd.read_csv('data2.csv')
4
5  df.dropna(inplace = True)
6
7  print(df.to_string())
```

O método *dropna* tem uma opção *inplace* que permite a remoção das linhas que contém NA (ou NaN) sem a necessidade de criação de cópia do dataframe.

| | Unnamed: 0 | Duration | Date | Pulse | Maxpulse | Calories |
|---|---|---|---|---|---|---|
| 0 | 0 | 60 | 2020/12/01 | 110 | 130 | 409.1 |
| 1 | 1 | 60 | 2020/12/02 | 117 | 145 | 479.0 |
| 2 | 2 | 60 | 2020/12/03 | 103 | 135 | 340.0 |
| 3 | 3 | 45 | 2020/12/04 | 109 | 175 | 282.4 |
| 4 | 4 | 45 | 2020/12/05 | 117 | 148 | 406.0 |
| 5 | 5 | 60 | 2020/12/06 | 102 | 127 | 300.0 |
| 6 | 6 | 60 | 2020/12/07 | 110 | 136 | 374.0 |
| 7 | 7 | 450 | 2020/12/08 | 104 | 134 | 253.3 |
| 8 | 8 | 30 | 2020/12/09 | 109 | 133 | 195.1 |
| 9 | 9 | 60 | 2020/12/10 | 98 | 124 | 269.0 |
| 10 | 10 | 60 | 2020/12/11 | 103 | 147 | 329.3 |
| 11 | 11 | 60 | 2020/12/12 | 100 | 120 | 250.7 |
| 12 | 12 | 60 | 2020/12/12 | 100 | 120 | 250.7 |
| 13 | 13 | 60 | 2020/12/13 | 106 | 128 | 345.3 |
| 14 | 14 | 60 | 2020/12/14 | 104 | 132 | 379.3 |
| 15 | 15 | 60 | 2020/12/15 | 98 | 123 | 275.0 |
| 16 | 16 | 60 | 2020/12/16 | 98 | 120 | 215.2 |
| 17 | 17 | 60 | 2020/12/17 | 100 | 120 | 300.0 |
| 19 | 19 | 60 | 2020/12/19 | 103 | 123 | 323.0 |
| 20 | 20 | 45 | 2020/12/20 | 97 | 125 | 243.0 |
| 21 | 21 | 60 | 2020/12/21 | 108 | 131 | 364.2 |
| 23 | 23 | 60 | 2020/12/23 | 130 | 101 | 300.0 |
| 24 | 24 | 45 | 2020/12/24 | 105 | 132 | 246.0 |
| 25 | 25 | 60 | 2020/12/25 | 102 | 126 | 334.5 |
| 26 | 26 | 60 | 2020/12/26 | 100 | 120 | 250.0 |
| 27 | 27 | 60 | 2020/12/27 | 92 | 118 | 241.0 |
| 29 | 29 | 60 | 2020/12/29 | 100 | 132 | 280.0 |
| 30 | 30 | 60 | 2020/12/30 | 102 | 129 | 380.3 |
| 31 | 31 | 60 | 2020/12/31 | 92 | 115 | 243.0 |

# LIMPEZA DOS DADOS (V) – DADOS FALTANTES



Tem-se a opção de substituir os valores faltantes por valores definidos pelo usuário, ou mesmo uma função dos dados restantes, quando se deseja preservar informação das colunas que contém dados válidos.

```python
import pandas as pd

df = pd.read_csv('data2.csv')

df.fillna(130, inplace = True)
```

```python
import pandas as pd

df = pd.read_csv('data.csv')

df["Calories"].fillna(130, inplace = True)
```

O método *fillna* inclui o valor especificado em todos os lugares onde NA (ou NaN) aparecem.

É possível definir sobre qual coluna o método *fillna* se aplica

Esse uso considera que o usuário está tomando uma decisão consciente de alterar os dados do dataframe

# LIMPEZA DOS DADOS (VI) – DADOS FALTANTES

Uma possibilidade de limpeza de dados é substituir o resultado faltante por alguma das propriedades estatísticas dos dados restantes.

Algumas possibilidades são:

Média: Soma dos valores divididas pela quantidade

Mediana: O valor central, aquele em que metade das observações são maiores, e metade são menores.

Moda: O valor que aparece com maior frequência

# LIMPEZA DOS DADOS (VII) – DADOS FALTANTES



data2.csv × pandas_16.py ×

```
1   import pandas as pd
2
3   df = pd.read_csv('data2.csv')
4
5   x = df["Calories"].mean()
6
7   df["Calories"].fillna(x, inplace = True)
8
9   print(df.to_string())
10
11  print(x)
```

O valor da média da coluna "Calories" é calculado, e usando com o método *fillna* como valor especificado em onde NA (ou NaN) aparece.

As outras funções podem ser chamadas usando:

```
x = df["Calories"].median()

x = df["Calories"].mode()[0]
```

```
In [35]: runfile('C:/Users/win/Dropbox/USP/2023/remote/MAP2112_2(
pandas/pandas_16.py', wdir='C:/Users/win/Dropbox/USP/2023/remote,
MAP2112_2023/scripts/pandas')
    Unnamed: 0  Duration        Date  Pulse  Maxpulse  Calories
0            0        60  2020/12/01    110       130    409.10
1            1        60  2020/12/02    117       145    479.00
2            2        60  2020/12/03    103       135    340.00
3            3        45  2020/12/04    109       175    282.40
4            4        45  2020/12/05    117       148    406.00
5            5        60  2020/12/06    102       127    300.00
6            6        60  2020/12/07    110       136    374.00
7            7       450  2020/12/08    104       134    253.30
8            8        30  2020/12/09    109       133    195.10
9            9        60  2020/12/10     98       124    269.00
10          10        60  2020/12/11    103       147    329.30
11          11        60  2020/12/12    100       120    250.70
12          12        60  2020/12/12    100       120    250.70
13          13        60  2020/12/13    106       128    345.30
14          14        60  2020/12/14    104       132    379.30
15          15        60  2020/12/15     98       123    275.00
16          16        60  2020/12/16     98       120    215.20
17          17        60  2020/12/17    100       120    300.00
18          18        45  2020/12/18     90       112    304.68
19          19        60  2020/12/19    103       123    323.00
20          20        45  2020/12/20     97       125    243.00
21          21        60  2020/12/21    108       131    364.20
22          22        45         NaN    100       119    282.00
23          23        60  2020/12/23    130       101    300.00
24          24        45  2020/12/24    105       132    246.00
25          25        60  2020/12/25    102       126    334.50
26          26        60  2020/12/26    100       120    250.00
27          27        60  2020/12/27     92       118    241.00
28          28        60  2020/12/28    103       132    304.68
29          29        60  2020/12/29    100       132    280.00
30          30        60  2020/12/30    102       129    380.30
31          31        60  2020/12/31     92       115    243.00
304.68
```

média

# LIMPEZA DOS DADOS (VIII) - FORMATOS



data2.csv

```
,Duration,Date,Pulse,Maxpulse,Calories
0,60,"2020/12/01",110,130,409.1
1,60,"2020/12/02",117,145,479.0
2,60,"2020/12/03",103,135,340.0
3,45,"2020/12/04",109,175,282.4
4,45,"2020/12/05",117,148,406.0
5,60,"2020/12/06",102,127,300.0
6,60,"2020/12/07",110,136,374.0
7,450,"2020/12/08",104,134,253.3
8,30,"2020/12/09",109,133,195.1
9,60,"2020/12/10",98,124,269.0
10,60,"2020/12/11",103,147,329.3
11,60,"2020/12/12",100,120,250.7
12,60,"2020/12/12",100,120,250.7
13,60,"2020/12/13",106,128,345.3
14,60,"2020/12/14",104,132,379.3
15,60,"2020/12/15",98,123,275.0
16,60,"2020/12/16",98,120,215.2
17,60,"2020/12/17",100,120,300.0
18,45,"2020/12/18",90,112,NaN
19,60,"2020/12/19",103,123,323.0
20,45,"2020/12/20",97,125,243.0
21,60,"2020/12/21",108,131,364.2
22,45,NaN,100,119,282.0
23,60,"2020/12/23",130,101,300.0
24,45,"2020/12/24",105,132,246.0
25,60,"2020/12/25",102,126,334.5
26,60,2020/12/26,100,120,250.0
27,60,"2020/12/27",92,118,241.0
28,60,"2020/12/28",103,132,NaN
29,60,"2020/12/29",100,132,280.0
30,60,"2020/12/30",102,129,380.3
31,60,"2020/12/31",92,115,243.0
```

No arquivo original existem algum valores com formatos errados na coluna “Date”.

Linha 22 - Um valor está faltando (NaN)

Linha 26 – O valor não está entre aspas

Entre as opções disponíveis, já vimos como remover as linhas. Uma outra possibilidade é converter todas as células da coluna ao mesmo formato.

# LIMPEZA DOS DADOS (IX) - FORMATOS



data2.csv × pandas_17.py ×

```
1  import pandas as pd
2
3  df = pd.read_csv('data2.csv')
4
5  df['Date'] = pd.to_datetime(df['Date'])
6
7  print(df.to_string())
```

O método “to_datetime” é aplicado na coluna “Date”.

A linha 26 foi corrigida mas o dado faltante na linha 22 ainda precisa ser tratado, seja pela remoção ou por preenchimento (perceba que é uma sequência diária de datas).

```
In [38]: runfile('C:/Users/win/Dropbox/USP/2023/remote/MAP2112_
pandas/pandas_17.py', wdir='C:/Users/win/Dropbox/USP/2023/remot
MAP2112_2023/scripts/pandas')
    Unnamed: 0  Duration       Date  Pulse  Maxpulse  Calories
0            0        60 2020-12-01    110       130     409.1
1            1        60 2020-12-02    117       145     479.0
2            2        60 2020-12-03    103       135     340.0
3            3        45 2020-12-04    109       175     282.4
4            4        45 2020-12-05    117       148     406.0
5            5        60 2020-12-06    102       127     300.0
6            6        60 2020-12-07    110       136     374.0
7            7       450 2020-12-08    104       134     253.3
8            8        30 2020-12-09    109       133     195.1
9            9        60 2020-12-10     98       124     269.0
10          10        60 2020-12-11    103       147     329.3
11          11        60 2020-12-12    100       120     250.7
12          12        60 2020-12-12    100       120     250.7
13          13        60 2020-12-13    106       128     345.3
14          14        60 2020-12-14    104       132     379.3
15          15        60 2020-12-15     98       123     275.0
16          16        60 2020-12-16     98       120     215.2
17          17        60 2020-12-17    100       120     300.0
18          18        45 2020-12-18     90       112       NaN
19          19        60 2020-12-19    103       123     323.0
20          20        45 2020-12-20     97       125     243.0
21          21        60 2020-12-21    108       131     364.2
22          22        45        NaT    100       119     282.0
23          23        60 2020-12-23    130       101     300.0
24          24        45 2020-12-24    105       132     246.0
25          25        60 2020-12-25    102       126     334.5
26          26        60 2020-12-26    100       120     250.0
27          27        60 2020-12-27     92       118     241.0
28          28        60 2020-12-28    103       132       NaN
29          29        60 2020-12-29    100       132     280.0
30          30        60 2020-12-30    102       129     380.3
31          31        60 2020-12-31     92       115     243.0
```

# LIMPEZA DOS DADOS (X) - FORMATOS



data2.csv × pandas_18.py ×

```
1  import pandas as pd
2
3  df = pd.read_csv('data2.csv')
4
5  df['Date'] = pd.to_datetime(df['Date'])
6
7  df["Date"].fillna("2020-12-22", inplace = True)
8
9  print(df.to_string())
```

Usando o método “fillna” visto anteriormente podemos inserir diretamente a data é aplicado na coluna “Date”.

Data sequencial inserida

```
In [43]: runfile('C:/Users/win/Dropbox/USP/2023/remote/MAP2112_2
pandas/pandas_18.py', wdir='C:/Users/win/Dropbox/USP/2023/remote
MAP2112_2023/scripts/pandas')
    Unnamed: 0  Duration       Date  Pulse  Maxpulse  Calories
0            0        60 2020-12-01    110       130     409.1
1            1        60 2020-12-02    117       145     479.0
2            2        60 2020-12-03    103       135     340.0
3            3        45 2020-12-04    109       175     282.4
4            4        45 2020-12-05    117       148     406.0
5            5        60 2020-12-06    102       127     300.0
6            6        60 2020-12-07    110       136     374.0
7            7       450 2020-12-08    104       134     253.3
8            8        30 2020-12-09    109       133     195.1
9            9        60 2020-12-10     98       124     269.0
10          10        60 2020-12-11    103       147     329.3
11          11        60 2020-12-12    100       120     250.7
12          12        60 2020-12-12    100       120     250.7
13          13        60 2020-12-13    106       128     345.3
14          14        60 2020-12-14    104       132     379.3
15          15        60 2020-12-15     98       123     275.0
16          16        60 2020-12-16     98       120     215.2
17          17        60 2020-12-17    100       120     300.0
18          18        45 2020-12-18     90       112       NaN
19          19        60 2020-12-19    103       123     323.0
20          20        45 2020-12-20     97       125     243.0
21          21        60 2020-12-21    108       131     364.2
22          22        45 2020-12-22    100       119     282.0
23          23        60 2020-12-23    130       101     300.0
24          24        45 2020-12-24    105       132     246.0
25          25        60 2020-12-25    102       126     334.5
26          26        60 2020-12-26    100       120     250.0
27          27        60 2020-12-27     92       118     241.0
28          28        60 2020-12-28    103       132       NaN
29          29        60 2020-12-29    100       132     280.0
30          30        60 2020-12-30    102       129     380.3
31          31        60 2020-12-31     92       115     243.0
```

# LIMPEZA DOS DADOS (XI) - FORMATOS



data2.csv × pandas_19.py ×

```
1  import pandas as pd
2
3  df = pd.read_csv('data2.csv')
4
5  df['Date'] = pd.to_datetime(df['Date'])
6
7  df.dropna(subset=['Date'], inplace = True)
8
9  print(df.to_string())
```

O método “dropna” é aplicado na coluna “Date” usando a opção subset.

A linha 22 foi removida.

```
In [42]: runfile('C:/Users/win/Dropbox/USP/2023/remote/MAP2112_2
pandas/pandas_19.py', wdir='C:/Users/win/Dropbox/USP/2023/remote
MAP2112_2023/scripts/pandas')
    Unnamed: 0  Duration       Date  Pulse  Maxpulse  Calories
0            0        60 2020-12-01    110       130     409.1
1            1        60 2020-12-02    117       145     479.0
2            2        60 2020-12-03    103       135     340.0
3            3        45 2020-12-04    109       175     282.4
4            4        45 2020-12-05    117       148     406.0
5            5        60 2020-12-06    102       127     300.0
6            6        60 2020-12-07    110       136     374.0
7            7       450 2020-12-08    104       134     253.3
8            8        30 2020-12-09    109       133     195.1
9            9        60 2020-12-10     98       124     269.0
10          10        60 2020-12-11    103       147     329.3
11          11        60 2020-12-12    100       120     250.7
12          12        60 2020-12-12    100       120     250.7
13          13        60 2020-12-13    106       128     345.3
14          14        60 2020-12-14    104       132     379.3
15          15        60 2020-12-15     98       123     275.0
16          16        60 2020-12-16     98       120     215.2
17          17        60 2020-12-17    100       120     300.0
18          18        45 2020-12-18     90       112       NaN
19          19        60 2020-12-19    103       123     323.0
20          20        45 2020-12-20     97       125     243.0
21          21        60 2020-12-21    108       131     364.2
23          23        60 2020-12-23    130       101     300.0
24          24        45 2020-12-24    105       132     246.0
25          25        60 2020-12-25    102       126     334.5
26          26        60 2020-12-26    100       120     250.0
27          27        60 2020-12-27     92       118     241.0
28          28        60 2020-12-28    103       132       NaN
29          29        60 2020-12-29    100       132     280.0
30          30        60 2020-12-30    102       129     380.3
31          31        60 2020-12-31     92       115     243.0
```

# LIMPEZA DOS DADOS (XII) – DATA INCORRETOS/FILTROS



data2.csv

```
,Duration,Date,Pulse,Maxpulse,Calories
0,60,"2020/12/01",110,130,409.1
1,60,"2020/12/02",117,145,479.0
2,60,"2020/12/03",103,135,340.0
3,45,"2020/12/04",109,175,282.4
4,45,"2020/12/05",117,148,406.0
5,60,"2020/12/06",102,127,300.0
6,60,"2020/12/07",110,136,374.0
7,450,"2020/12/08",104,134,253.3
8,30,"2020/12/09",109,133,195.1
9,60,"2020/12/10",98,124,269.0
10,60,"2020/12/11",103,147,329.3
11,60,"2020/12/12",100,120,250.7
12,60,"2020/12/12",100,120,250.7
13,60,"2020/12/13",106,128,345.3
14,60,"2020/12/14",104,132,379.3
15,60,"2020/12/15",98,123,275.0
16,60,"2020/12/16",98,120,215.2
17,60,"2020/12/17",100,120,300.0
18,45,"2020/12/18",90,112,NaN
19,60,"2020/12/19",103,123,323.0
20,45,"2020/12/20",97,125,243.0
21,60,"2020/12/21",108,131,364.2
22,45,NaN,100,119,282.0
23,60,"2020/12/23",130,101,300.0
24,45,"2020/12/24",105,132,246.0
25,60,"2020/12/25",102,126,334.5
26,60,2020/12/26,100,120,250.0
27,60,"2020/12/27",92,118,241.0
28,60,"2020/12/28",103,132,NaN
29,60,"2020/12/29",100,132,280.0
30,60,"2020/12/30",102,129,380.3
31,60,"2020/12/31",92,115,243.0
```

No arquivo original existe um valores inconsistente na coluna “Duration” na linha 7.

O valor máximo da coluna é 60 e outro valor que aparece é 45. O valor 450 pode ser um erro de digitação, que pode ser corrigido por substituição direta.

# LIMPEZA DOS DADOS (XIII) – DATA INCORRETOS/FILTROS



data2.csv × pandas_20.py ×

```
1  import pandas as pd
2
3  df = pd.read_csv('data2.csv')
4
5  df.loc[7, 'Duration'] = 45
6
7  print(df.to_string())
```

O método “loc” insere o valor desejado na linha identificada por índice e pelo coluna por é aplicado na coluna “Date” usando a opção subset.

A linha 7 foi corrigida.

```
In [44]: runfile('C:/Users/win/Dropbox/USP/2023/remote/MAP2112_20
pandas/pandas_20.py', wdir='C:/Users/win/Dropbox/USP/2023/remote/
MAP2112_2023/scripts/pandas')
    Unnamed: 0  Duration        Date  Pulse  Maxpulse  Calories
0            0        60  2020/12/01    110       130     409.1
1            1        60  2020/12/02    117       145     479.0
2            2        60  2020/12/03    103       135     340.0
3            3        45  2020/12/04    109       175     282.4
4            4        45  2020/12/05    117       148     406.0
5            5        60  2020/12/06    102       127     300.0
6            6        60  2020/12/07    110       136     374.0
7            7        45  2020/12/08    104       134     253.3
8            8        30  2020/12/09    109       133     195.1
9            9        60  2020/12/10     98       124     269.0
10          10        60  2020/12/11    103       147     329.3
11          11        60  2020/12/12    100       120     250.7
12          12        60  2020/12/12    100       120     250.7
13          13        60  2020/12/13    106       128     345.3
14          14        60  2020/12/14    104       132     379.3
15          15        60  2020/12/15     98       123     275.0
16          16        60  2020/12/16     98       120     215.2
17          17        60  2020/12/17    100       120     300.0
18          18        45  2020/12/18     90       112       NaN
19          19        60  2020/12/19    103       123     323.0
20          20        45  2020/12/20     97       125     243.0
21          21        60  2020/12/21    108       131     364.2
22          22        45         NaN    100       119     282.0
23          23        60  2020/12/23    130       101     300.0
24          24        45  2020/12/24    105       132     246.0
25          25        60  2020/12/25    102       126     334.5
26          26        60  2020/12/26    100       120     250.0
27          27        60  2020/12/27     92       118     241.0
28          28        60  2020/12/28    103       132       NaN
29          29        60  2020/12/29    100       132     280.0
30          30        60  2020/12/30    102       129     380.3
31          31        60  2020/12/31     92       115     243.0
```

# LIMPEZA DOS DADOS (XIV) – DATA INCORRETOS/FILTROS



data2.csv × pandas_21.py ×

```
1  import pandas as pd
2
3  df = pd.read_csv('data2.csv')
4
5  for x in df.index:
6    if df.loc[x, "Duration"] > 45:
7      df.loc[x, "Duration"] = 45
8
9  print(df.to_string())
```

Usando o método "*loc*" incluído num loop de *for* pode-se construir um filtro para ajustar os valores de acordo com alguma lógica.

O método *index* extrai o valor da linhas.

O *if* ajusta o valor, no exemplo, para um valor máximo.

```
In [46]: runfile('C:/Users/win/Dropbox/USP/2023/remote/MAP2112_20
pandas/pandas_21.py', wdir='C:/Users/win/Dropbox/USP/2023/remote/
MAP2112_2023/scripts/pandas')
    Unnamed: 0  Duration        Date  Pulse  Maxpulse  Calories
0            0        45  2020/12/01    110       130     409.1
1            1        45  2020/12/02    117       145     479.0
2            2        45  2020/12/03    103       135     340.0
3            3        45  2020/12/04    109       175     282.4
4            4        45  2020/12/05    117       148     406.0
5            5        45  2020/12/06    102       127     300.0
6            6        45  2020/12/07    110       136     374.0
7            7        45  2020/12/08    104       134     253.3
8            8        30  2020/12/09    109       133     195.1
9            9        45  2020/12/10     98       124     269.0
10          10        45  2020/12/11    103       147     329.3
11          11        45  2020/12/12    100       120     250.7
12          12        45  2020/12/12    100       120     250.7
13          13        45  2020/12/13    106       128     345.3
14          14        45  2020/12/14    104       132     379.3
15          15        45  2020/12/15     98       123     275.0
16          16        45  2020/12/16     98       120     215.2
17          17        45  2020/12/17    100       120     300.0
18          18        45  2020/12/18     90       112       NaN
19          19        45  2020/12/19    103       123     323.0
20          20        45  2020/12/20     97       125     243.0
21          21        45  2020/12/21    108       131     364.2
22          22        45         NaN    100       119     282.0
23          23        45  2020/12/23    130       101     300.0
24          24        45  2020/12/24    105       132     246.0
25          25        45  2020/12/25    102       126     334.5
26          26        45  2020/12/26    100       120     250.0
27          27        45  2020/12/27     92       118     241.0
28          28        45  2020/12/28    103       132       NaN
29          29        45  2020/12/29    100       132     280.0
30          30        45  2020/12/30    102       129     380.3
31          31        45  2020/12/31     92       115     243.0
```

# LIMPEZA DOS DADOS (XV) – DATA INCORRETOS/FILTROS



data2.csv × pandas_22.py ×

```
1  import pandas as pd
2
3  df = pd.read_csv('data2.csv')
4
5  for x in df.index:
6    if df.loc[x, "Duration"] > 120:
7      df.drop(x, inplace = True)
8
9  print(df.to_string())
```

Tem-se a opção de deletar linhas baseadas no filtro usando o método drop.

```
In [47]: runfile('C:/Users/win/Dropbox/USP/2023/remote/MAP2112_20
pandas/pandas_22.py', wdir='C:/Users/win/Dropbox/USP/2023/remote/
MAP2112_2023/scripts/pandas')
    Unnamed: 0  Duration        Date  Pulse  Maxpulse  Calories
0            0        60  2020/12/01    110       130     409.1
1            1        60  2020/12/02    117       145     479.0
2            2        60  2020/12/03    103       135     340.0
3            3        45  2020/12/04    109       175     282.4
4            4        45  2020/12/05    117       148     406.0
5            5        60  2020/12/06    102       127     300.0
6            6        60  2020/12/07    110       136     374.0
8            8        30  2020/12/09    109       133     195.1
9            9        60  2020/12/10     98       124     269.0
10          10        60  2020/12/11    103       147     329.3
11          11        60  2020/12/12    100       120     250.7
12          12        60  2020/12/12    100       120     250.7
13          13        60  2020/12/13    106       128     345.3
14          14        60  2020/12/14    104       132     379.3
15          15        60  2020/12/15     98       123     275.0
16          16        60  2020/12/16     98       120     215.2
17          17        60  2020/12/17    100       120     300.0
18          18        45  2020/12/18     90       112       NaN
19          19        60  2020/12/19    103       123     323.0
20          20        45  2020/12/20     97       125     243.0
21          21        60  2020/12/21    108       131     364.2
22          22        45         NaN    100       119     282.0
23          23        60  2020/12/23    130       101     300.0
24          24        45  2020/12/24    105       132     246.0
25          25        60  2020/12/25    102       126     334.5
26          26        60  2020/12/26    100       120     250.0
27          27        60  2020/12/27     92       118     241.0
28          28        60  2020/12/28    103       132       NaN
29          29        60  2020/12/29    100       132     280.0
30          30        60  2020/12/30    102       129     380.3
31          31        60  2020/12/31     92       115     243.0
```

# LIMPEZA DOS DADOS (XVI) – DADOS DUPLICADOS



data2.csv

```
,Duration,Date,Pulse,Maxpulse,Calories
0,60,"2020/12/01",110,130,409.1
1,60,"2020/12/02",117,145,479.0
2,60,"2020/12/03",103,135,340.0
3,45,"2020/12/04",109,175,282.4
4,45,"2020/12/05",117,148,406.0
5,60,"2020/12/06",102,127,300.0
6,60,"2020/12/07",110,136,374.0
7,450,"2020/12/08",104,134,253.3
8,30,"2020/12/09",109,133,195.1
9,60,"2020/12/10",98,124,269.0
10,60,"2020/12/11",103,147,329.3
11,60,"2020/12/12",100,120,250.7
12,60,"2020/12/12",100,120,250.7
13,60,"2020/12/13",106,128,345.3
14,60,"2020/12/14",104,132,379.3
15,60,"2020/12/15",98,123,275.0
16,60,"2020/12/16",98,120,215.2
17,60,"2020/12/17",100,120,300.0
18,45,"2020/12/18",90,112,NaN
19,60,"2020/12/19",103,123,323.0
20,45,"2020/12/20",97,125,243.0
21,60,"2020/12/21",108,131,364.2
22,45,NaN,100,119,282.0
23,60,"2020/12/23",130,101,300.0
24,45,"2020/12/24",105,132,246.0
25,60,"2020/12/25",102,126,334.5
26,60,2020/12/26,100,120,250.0
27,60,"2020/12/27",92,118,241.0
28,60,"2020/12/28",103,132,NaN
29,60,"2020/12/29",100,132,280.0
30,60,"2020/12/30",102,129,380.3
31,60,"2020/12/31",92,115,243.0
```

No arquivo original existem linhas duplicadas, 11 e 12.

Por algum motivo, relacionado a forma em que criei o arquivos o método *duplicate* não funcionou.

A linha pode ser removida diretamente usando o método drop

# ANÁLISE DE DADOS – CORRELAÇÕES

Uma forma preliminar de extrair uma relação entre os dados é analisar a correlação entre as variáveis. Tomando como exemplo o arquivo data.csv.

data3.csv × pandas_24.py ×

```
1  import pandas as pd
2
3  df = pd.read_csv('data.csv')
4
5  print(df)
6
7  print(df.info())
8
9  print(df.corr())
```

O método “corr” extrai a matriz de correlação

```
In [65]: runfile('C:/Users/win/Dropbox/USP/2023/remot
pandas/pandas_24.py', wdir='C:/Users/win/Dropbox/USP/
MAP2112_2023/scripts/pandas')
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
..        ...    ...       ...       ...
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4

[169 rows x 4 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Duration  169 non-null    int64
 1   Pulse     169 non-null    int64
 2   Maxpulse  169 non-null    int64
 3   Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 5.4 KB
None
             Duration     Pulse  Maxpulse  Calories
Duration     1.000000 -0.155408  0.009403  0.922717
Pulse       -0.155408  1.000000  0.786535  0.025121
Maxpulse     0.009403  0.786535  1.000000  0.203813
Calories     0.922717  0.025121  0.203813  1.000000
```

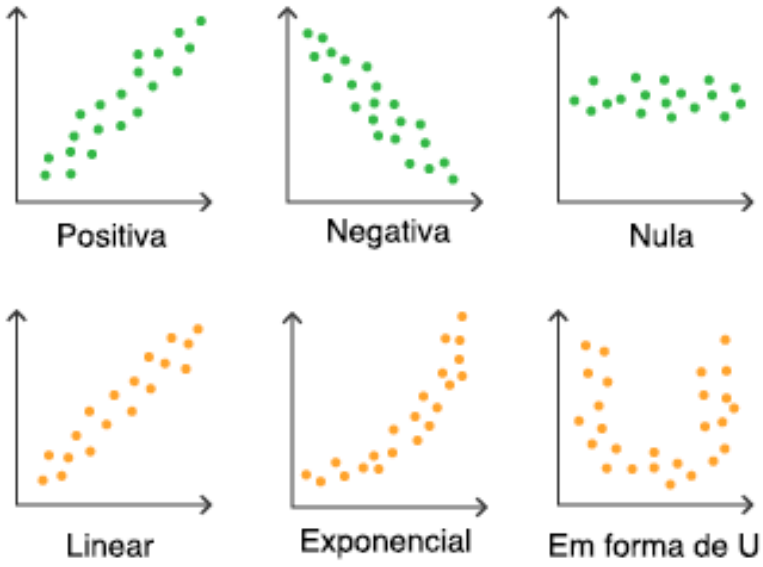
Os coeficientes da matriz são calculados usando essa relação:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \qquad \textbf{(Eq.3)}$$

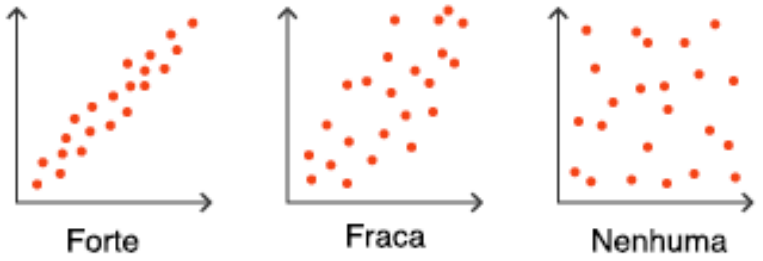
# ANÁLISE DE DADOS (III) – CORRELAÇÕES

Considerando apenas duas variáveis.

**Tipos de correlação:**




**Força da correlação:**




O coeficiente de correlação varia entre -1 e 1.

O valor de referência para a força de correlação é o coeficiente de Pearson maior que 0.6 (em valor absoluto).

# ANÁLISE DE DADOS (IV) – CORRELAÇÕES

| | Duration | Pulse | Maxpulse | Calories |
|---|---|---|---|---|
| Duration | 1.000000 | -0.155408 | 0.009403 | 0.922717 |
| Pulse | -0.155408 | 1.000000 | 0.786535 | 0.025121 |
| Maxpulse | 0.009403 | 0.786535 | 1.000000 | 0.203813 |
| Calories | 0.922717 | 0.025121 | 0.203813 | 1.000000 |

A matriz indica o valor de coeficientes de correlação para as variáveis duas a duas:

- A diagonal indica que a correlação das variáveis consigo mesmas é perfeita (o que é esperado).
- A correlação entre calorias e duração do exercícios é alta, maior do 0.9, o que é esperado, quanto maior a duração do exercício é esperada uma maior queima calórica.
- Também existe uma boa correlação entre a pulsação média e a máxima (0.78).
- Os outros valores são menores que 0.6.

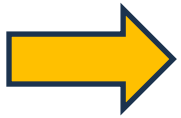
# VISUALIZAÇÃO DOS DADOS - GRÁFICOS

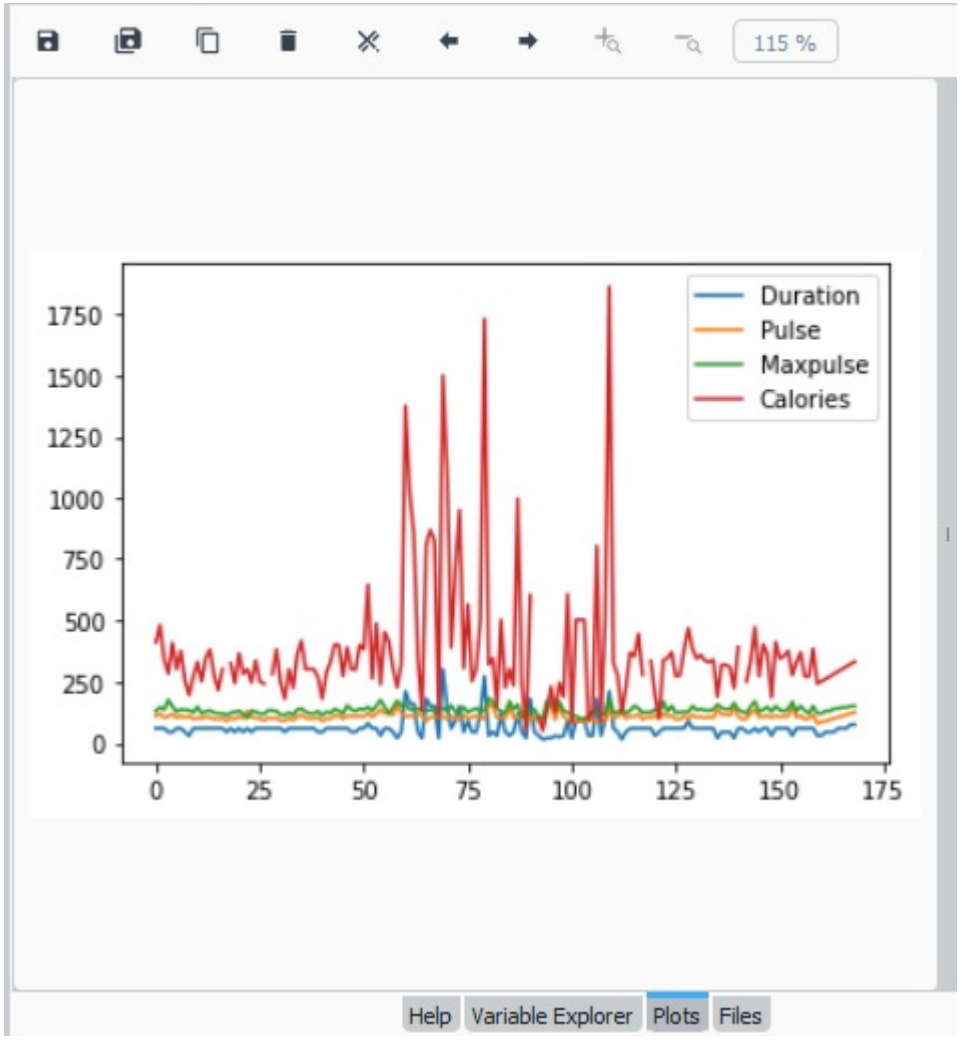


O módulo mais popular de visualização é o matplotlib. Ele deve ser importado para habilitar os métodos de plotagem. O mais simples é o pyplot.

data3.csv × pandas_25.py ×

```
1  import pandas as pd
2
3  import matplotlib.pyplot as plt
4
5  df = pd.read_csv('data.csv')
6
7  df.plot()
8
9  plt.show()
```






O método “plot” gera um gráfico default sobre o dataframe.

O método show exibe o gráfico.

Se nenhuma opção é selecionada o aspecto é o do lado, todas as colunas na vertical sendo o índice da linha horizontal. A legenda é gerada.

# VISUALIZAÇÃO DOS DADOS (II) - GRÁFICOS



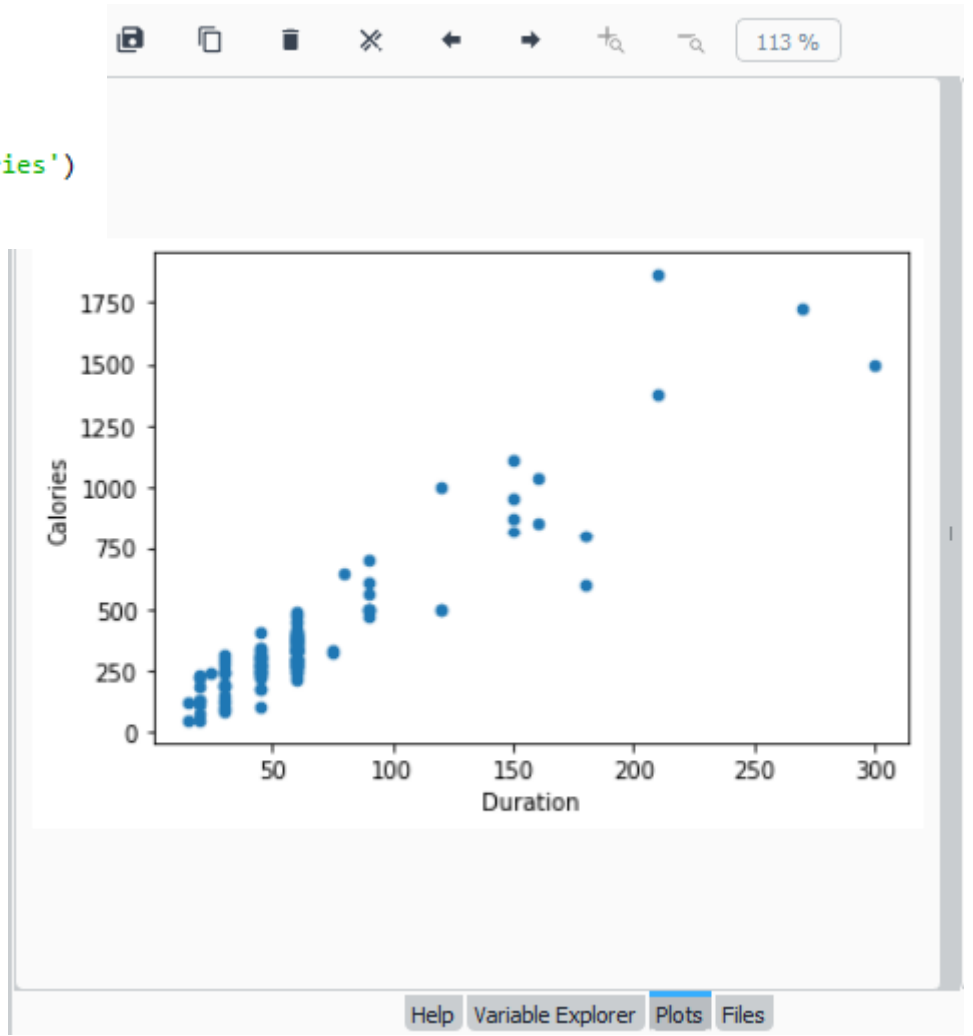
Um dos primeiros gráficos para melhor entender os dados, e observar graficamente os valores de correlação é o gráfico de dispersão (scatter plot)

data3.csv × pandas_26.py ×

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  df = pd.read_csv('data.csv')
5
6  df.plot(kind = 'scatter', x = 'Duration', y = 'Calories')
7
8  plt.show()
```

A opção “kind” no método “plot” permite escolher o tipo de gráfico. Os rótulos dos eixos são definidos pelas opções x e y.

O aspecto da correlação positiva é visível nos dados



# VISUALIZAÇÃO DOS DADOS (III) - GRÁFICOS

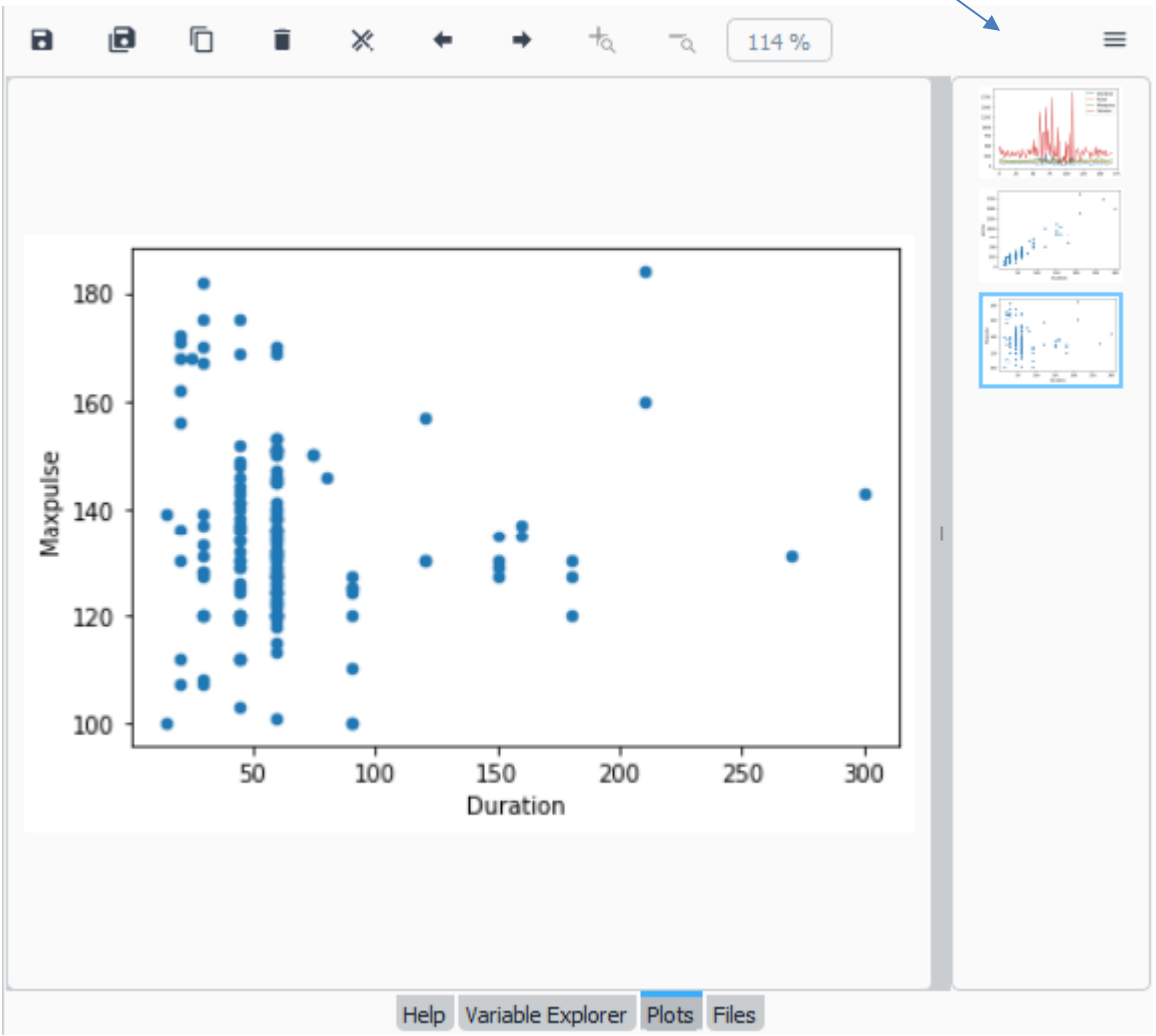


data3.csv × pandas_27.py ×

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  df = pd.read_csv('data.csv')
5
6  df.plot(kind = 'scatter', x = 'Duration', y = 'Maxpulse')
7
8  plt.show()
```

Os gráficos gerados vão se acumulando na lateral do painel

A baixa correlação entre "Duration" e "Maxpulse" é visível nos dados

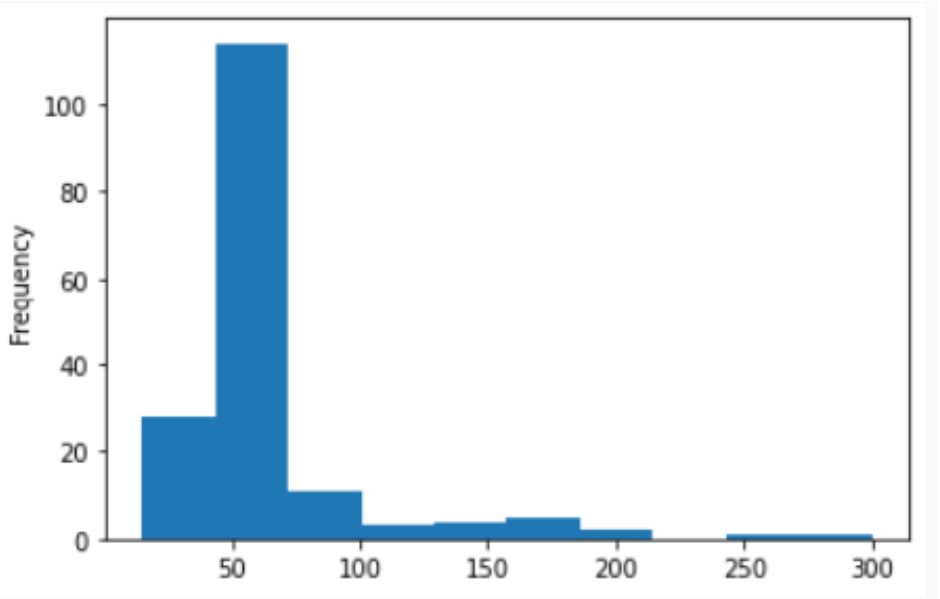


# VISUALIZAÇÃO DOS DADOS (IV) - GRÁFICOS

Um outro gráficos que pode ser relevante, para melhor entender os dados, é o gráfico de frequência - histograma (histogram). Os histogramas indicam quantos valores existem dentro de faixas discretas de dados.

data3.csv × pandas_28.py ×

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  df = pd.read_csv('data.csv')
5
6  df["Duration"].plot(kind = 'hist')
7
8  plt.show()
```




A opção “kind” no método “plot” foi modificada para “hist”.

O gráfico apresenta quantas leituras existem para os intervalos definidos.

Percebe-se, visualmente, que a maior quantidade treinos ocorreu entre 45 e 60 minutos.

# REGRESSÃO LINEAR

Para aplicar o modelo de regressão linear, vamos seguir o roteiro do tutorial disponível em:

https://www.w3schools.com/python/python_ml_linear_regression.asp

Ao invés de seguir o exemplo da referência relativo a regressão linear iremos utilizar os dataframe da seção sobre os métodos do Pandas.

A análise das informações do data.csv revela que existem dados faltantes na coluna "Calories".

```
In [70]: runfile('C:/Users/win/Dropbox/USP/2023/
pandas/pandas_29.py', wdir='C:/Users/win/Dropbo
MAP2112_2023/scripts/pandas')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Duration  169 non-null    int64
 1   Pulse     169 non-null    int64
 2   Maxpulse  169 non-null    int64
 3   Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 5.4 KB
None
```

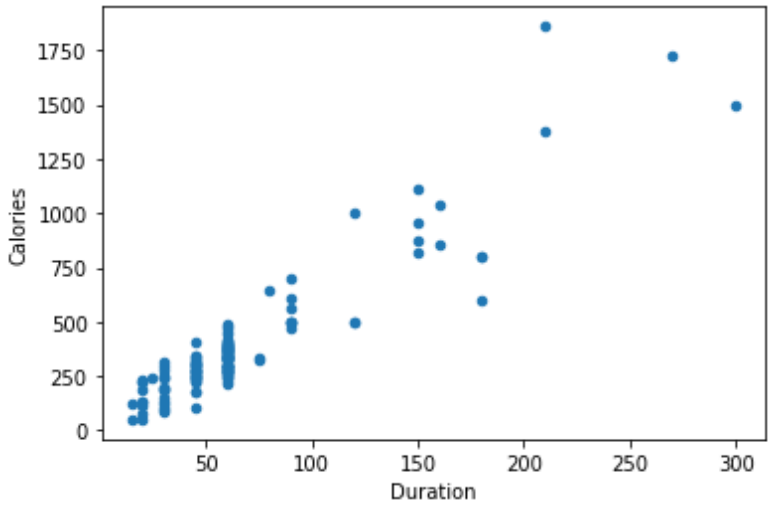
# REGRESSÃO LINEAR (II)

Promovendo a limpeza do arquivo removendo as linhas com dados faltantes e plotando o gráfico de dispersão (scatter plot).

data3.csv × pandas_29.py ×

```
1   import pandas as pd
2   import matplotlib.pyplot as plt
3
4   df = pd.read_csv('data.csv')
5
6   df.dropna(subset=['Calories'], inplace = True)
7
8   print(df.info())
9
10  df.plot(kind = 'scatter', x = 'Duration', y = 'Calories')
11
12  plt.show()
```

```
In [74]: runfile('C:/Users/win/Dropbox/USP
pandas/pandas_29.py', wdir='C:/Users/win/D
MAP2112_2023/scripts/pandas')
<class 'pandas.core.frame.DataFrame'>
Index: 164 entries, 0 to 168
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Duration  164 non-null    int64
 1   Pulse     164 non-null    int64
 2   Maxpulse  164 non-null    int64
 3   Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 6.4 KB
None
```

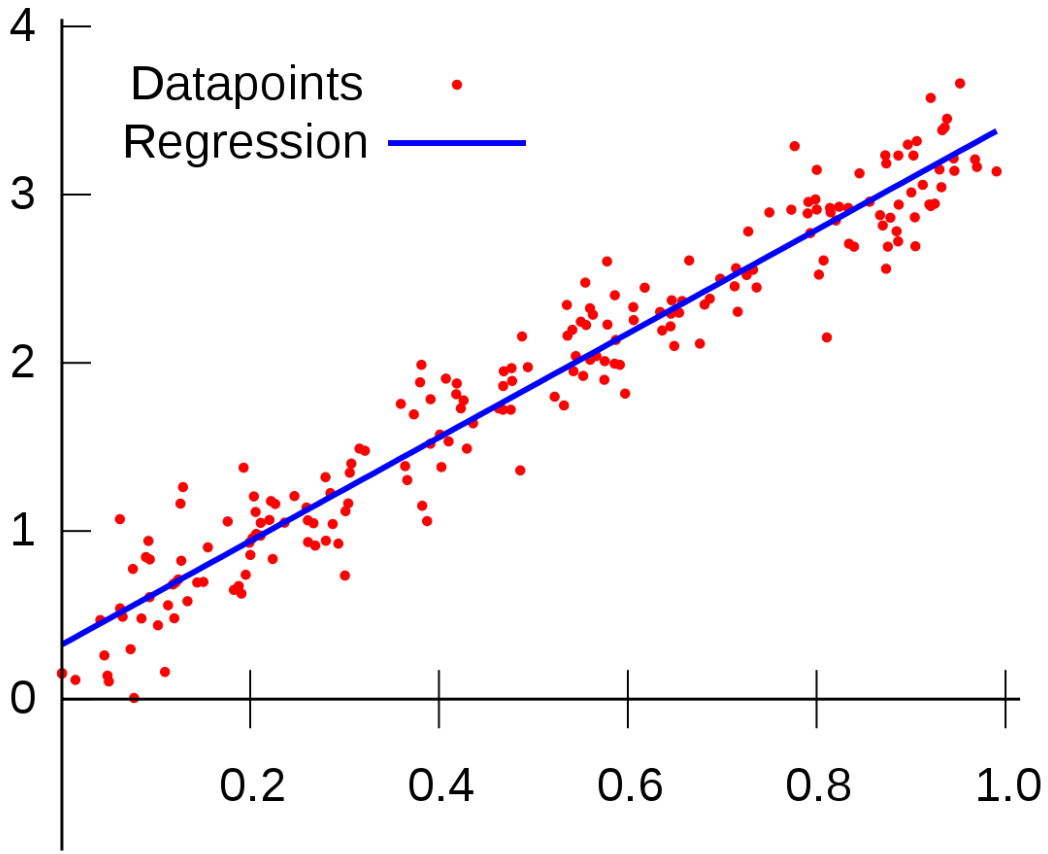


# REGRESSÃO LINEAR (III)

Uma regressão linear é um ajuste polinomial onde os coeficientes $\beta_0$ e $\beta_1$ são calculados pelo método dos mínimos quadrados.

$$Y = \beta_0 + \beta_1 X + e$$

Variável resposta | Intecepto | Coeficiente angular | Variável explicativa | Erro




Existe um módulo de métodos estatísticos que pode ser usado calcular os coeficientes da regressão.

# REGRESSÃO LINEAR (IV)



data3.csv × pandas_30.py ×

```
1   import pandas as pd
2   import matplotlib.pyplot as plt
3   from scipy import stats
4
5   df = pd.read_csv('data.csv')
6
7   df.dropna(subset=['Calories'], inplace = True)
8
9   x = df["Duration"]
10  y = df["Calories"]
11
12  slope, intercept, r, p, std_err = stats.linregress(x, y)
13
14  def myfunc(x):
15    return slope * x + intercept
16
17  mymodel = list(map(myfunc, x))
18
19  df.plot(kind = 'scatter', x = 'Duration', y = 'Calories')
20  plt.plot(x, mymodel)
21
22  plt.show()
23
24  print("slope: ",slope," intercept: ",intercept)
25  print("corr. coef. r: ",r)
26
```

O módulo scipy contém métodos de estatística (stats)

As colunas selecionadas foram armazenadas nas variáveis x e y para legilibilidade

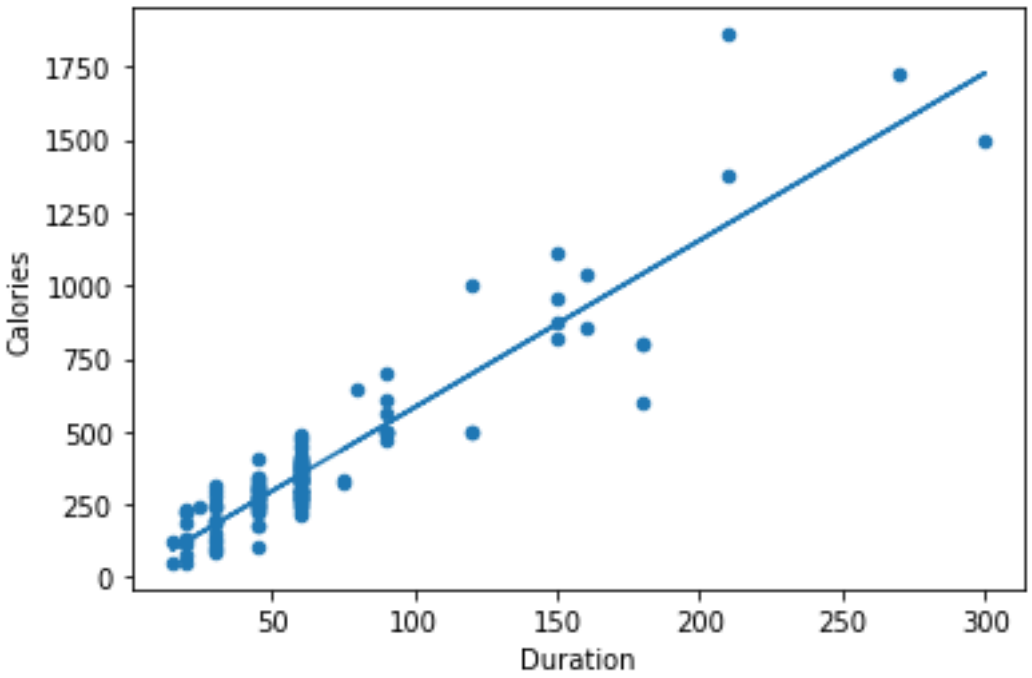
O método *linregress* extrai os coeficientes e o valor r de correlação (outros parâmetros foram mantidos por completude)

Uma função foi criada para armazenar o modelo linear

O modelo é usado para incluir a reta no gráfico.

# REGRESSÃO LINEAR (V)

Os gráficos podem ser supostos chamando o método plt.show() após todas as curvas serem criadas.




A saída do script com os valores dos coeficientes do ajuste e de correlação

```
In [77]: runfile('C:/Users/win/Dropbox/USP/2023/remote/MAP2112
pandas/pandas_30.py', wdir='C:/Users/win/Dropbox/USP/2023/remo
MAP2112_2023/scripts/pandas')
slope:  5.730938660462624  intercept:  8.171495682519492
corr. coef. r:  0.9227166783472468
```

O valor do coeficientes de correlação indica um bom modelo.

# REGRESSÃO LINEAR (VI)

Uma vez que o modelo foi construído (os coeficientes foram encontrados) ele pode ser usado para fazer previsões. Por exemplo:

Qual a perda calórica esperada ao treinar 90 minutos ?

Qual a duração de treino para perder 1000 calorias ?

```
print("perda 90 min: ",slope*90+intercept)

print("duração 100 cal: ", (1000-intercept)/slope)
```

```
perda 90 min:  523.9559751241557
duração 100 cal:  173.06562904957286
```

# Fim Aula 08