```
!pip install pandas
```

```
Requirement already satisfied: pandas in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (2.2.3)
Requirement already satisfied: numpy>=1.26.0 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from pandas) (2.2.4)
Requirement already satisfied: python-dateutil>=2.8.2 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)

[notice] A new release of pip is available: 24.3.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
import pandas as pd

# Séries
usa_data = pd.Series([13.33, 14.02, 14.25, 15.01], index=["2000","2001","2002","2003"])
print(usa_data)
```

```
2000    13.33
2001    14.02
2002    14.25
2003    15.01
dtype: float64
```

```
# # Séries
india_data = pd.Series([9.02, 9.01, 8.84, 9.84], index=["2000","2001","2002","2003"])
print(india_data)
```

```
2000    9.02
2001    9.01
2002    8.84
2003    9.84
dtype: float64
```

```
# Dataframe
df = pd.DataFrame({"USA": usa_data, "India": india_data})
print(df)
```

```
        USA  India
2000  13.33   9.02
2001  14.02   9.01
2002  14.25   8.84
2003  15.01   9.84
```

```
# Carregando o dataset ( conjunto de dados)
df = pd.read_csv(r"gapminder.tsv", sep="\t")
print(df.head())
print("\n")
print(type(df))
print("\n")
print(df.shape)
print("\n")
```

```
print(df.columns)
print("\n")
print(df.dtypes)
print("\n")
print(df.info())
```

```
       country continent  year  lifeExp       pop   gdpPercap
0  Afghanistan      Asia  1952   28.801   8425333  779.445314
1  Afghanistan      Asia  1957   30.332   9240934  820.853030
2  Afghanistan      Asia  1962   31.997  10267083  853.100710
3  Afghanistan      Asia  1967   34.020  11537966  836.197138
4  Afghanistan      Asia  1972   36.088  13079460  739.981106


<class 'pandas.core.frame.DataFrame'>


(1704, 6)


Index(['country', 'continent', 'year', 'lifeExp', 'pop', 'gdpPercap'], dtype='object')


country       object
continent     object
year           int64
lifeExp      float64
pop            int64
gdpPercap    float64
dtype: object


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1704 entries, 0 to 1703
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   country    1704 non-null   object
 1   continent  1704 non-null   object
 2   year       1704 non-null   int64
 3   lifeExp    1704 non-null   float64
 4   pop        1704 non-null   int64
 5   gdpPercap  1704 non-null   float64
dtypes: float64(2), int64(2), object(2)
memory usage: 80.0+ KB
None
```

```
# Observando linhas e colunas: colunas
country_df = df ["country"]
print(country_df.tail())
```

```
1699    Zimbabwe
1700    Zimbabwe
1701    Zimbabwe
1702    Zimbabwe
1703    Zimbabwe
Name: country, dtype: object
```

```
subset = df [["country", "continent", "year"]]
print(subset.head())
```

```
       country continent  year
0  Afghanistan      Asia  1952
1  Afghanistan      Asia  1957
2  Afghanistan      Asia  1962
3  Afghanistan      Asia  1967
4  Afghanistan      Asia  1972
```

```
# Subconjunto de linhas
display(df.loc[0])
print("\n")
display(df.loc [99])
print("\n")

try:
    display(df.loc[-1])
except:
    print("Não existe a posição -1")
```

```
country      Afghanistan
continent           Asia
year                1952
lifeExp           28.801
pop              8425333
gdpPercap     779.445314
Name: 0, dtype: object


country      Bangladesh
continent          Asia
year               1967
lifeExp          43.453
pop            62821884
gdpPercap    721.186086
Name: 99, dtype: object


Não existe a posição -1
```

```
number_of_rows = df.shape[0]
last_row_index = number_of_rows - 1
display(df.loc [last_row_index])
print("\n")
display(df.tail(n=1))
```

```
country        Zimbabwe
continent        Africa
year               2007
lifeExp          43.487
pop            12311143
gdpPercap    469.709298
Name: 1703, dtype: object
```

	country	continent	year	lifeExp	pop	gdpPercap
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298

```
display(df.loc[[0,99,999]])
```

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
99	Bangladesh	Asia	1967	43.453	62821884	721.186086
999	Mongolia	Asia	1967	51.253	1149500	1226.041130

```
display(df.iloc[1])
print("\n")
display(df.iloc[99])
print("\n")
display(df.iloc[-1])
print("\n")
display(df.iloc[[0,99,999]])
```

```
country      Afghanistan
continent           Asia
year                1957
lifeExp           30.332
pop              9240934
gdpPercap      820.85303
Name: 1, dtype: object


country      Bangladesh
continent          Asia
year               1967
lifeExp          43.453
pop            62821884
gdpPercap    721.186086
Name: 99, dtype: object


country        Zimbabwe
continent        Africa
year               2007
lifeExp          43.487
pop            12311143
gdpPercap    469.709298
Name: 1703, dtype: object
```

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
99	Bangladesh	Asia	1967	43.453	62821884	721.186086
999	Mongolia	Asia	1967	51.253	1149500	1226.041130

```
# Combinando
subset = df.loc[:, ["year", "pop"]]
display(subset.head())
```

	year	pop
0	1952	8425333
1	1957	9240934
2	1962	10267083
3	1967	11537966
4	1972	13079460

```
subset = df.iloc[:, [2,4,-1]]
display(subset.head())
```

	year	pop	gdpPercap
0	1952	8425333	779.445314
1	1957	9240934	820.853030
2	1962	10267083	853.100710
3	1967	11537966	836.197138
4	1972	13079460	739.981106

```
# Subconjunto de várias linhas e colunas
display(df.iloc[[0,99,999], [0,3,5]])
display(df.loc[[0,99,999], ["country","lifeExp", "gdpPercap"]])
```

	country	lifeExp	gdpPercap
0	Afghanistan	28.801	779.445314
99	Bangladesh	43.453	721.186086
999	Mongolia	51.253	1226.041130

	country	lifeExp	gdpPercap
0	Afghanistan	28.801	779.445314
99	Bangladesh	43.453	721.186086
999	Mongolia	51.253	1226.041130

```
# Obtendo colunas por intervalo
small_range = list(range(5))
display(small_range)
subset = df.iloc[:, small_range]
display(subset.head())

small_range = list(range(3, 6))
display(small_range)
subset = df.iloc[:, small_range]
display(subset.head())
```

```
[0, 1, 2, 3, 4]
```

	country	continent	year	lifeExp	pop
0	Afghanistan	Asia	1952	28.801	8425333
1	Afghanistan	Asia	1957	30.332	9240934
2	Afghanistan	Asia	1962	31.997	10267083
3	Afghanistan	Asia	1967	34.020	11537966
4	Afghanistan	Asia	1972	36.088	13079460

```
[3, 4, 5]
```

	lifeExp	pop	gdpPercap
0	28.801	8425333	779.445314
1	30.332	9240934	820.853030
2	31.997	10267083	853.100710
3	34.020	11537966	836.197138
4	36.088	13079460	739.981106

O que acontece se for esptipulado um valor fora da faixa?

Resp.: Causa um erro de "IndexError: positional indexers are out-of-bounds"

```
small_range = list(range(3, 7))
subset = df.iloc[:, small_range]
display(subset.head())
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[16], line 2
      1 small_range = list(range(3, 7))
----> 2 subset = df.iloc[:, small_range]
      3 display(subset.head())

File d:\Bruno\FATEC CDN\TecnicasProgramacaoCienciaDados\FatecVenv\Lib\site-packages\pandas\core\indexing.py:1184, in _LocationIndexer.__getitem__(self, key)
   1182     if self._is_scalar_access(key):
   1183         return self.obj._get_value(*key, takeable=self._takeable)
-> 1184     return self._getitem_tuple(key)
   1185 else:
   1186     # we by definition only have the 0th axis
   1187     axis = self.axis or 0

File d:\Bruno\FATEC CDN\TecnicasProgramacaoCienciaDados\FatecVenv\Lib\site-packages\pandas\core\indexing.py:1690, in _iLocIndexer._getitem_tuple(self, tup)
   1689 def _getitem_tuple(self, tup: tuple):
-> 1690     tup = self._validate_tuple_indexer(tup)
   1691     with suppress(IndexingError):
   1692         return self._getitem_lowerdim(tup)

File d:\Bruno\FATEC CDN\TecnicasProgramacaoCienciaDados\FatecVenv\Lib\site-packages\pandas\core\indexing.py:966, in _LocationIndexer._validate_tuple_indexer(self, key)
    964 for i, k in enumerate(key):
    965     try:
--> 966         self._validate_key(k, i)
    967     except ValueError as err:
    968         raise ValueError(
    969             "Location based indexing can only have "
    970             f"[{self._valid_types}] types"
    971         ) from err

File d:\Bruno\FATEC CDN\TecnicasProgramacaoCienciaDados\FatecVenv\Lib\site-packages\pandas\core\indexing.py:1612, in _iLocIndexer._validate_key(self, key, axis)
   1610     # check that the key does not exceed the maximum size of the index
   1611     if len(arr) and (arr.max() >= len_axis or arr.min() < -len_axis):
-> 1612         raise IndexError("positional indexers are out-of-bounds")
   1613 else:
   1614     raise ValueError(f"Can only index by location with a [{self._valid_types}]")

IndexError: positional indexers are out-of-bounds
```

```
# Fatiando colunas
small_range = list(range(3))
subset = df.iloc[:,small_range]
display(subset.head())
subset = df.iloc[:,:3]
display(subset.head())
small_range = list(range(3,6))
subset = df.iloc[:,small_range]
display(subset.head())
subset = df.iloc[:,3:6]
display(subset.head())
small_range = list(range(0,6,2))
subset = df.iloc[:,small_range]
display(subset.head())
```

	country	continent	year
0	Afghanistan	Asia	1952
1	Afghanistan	Asia	1957
2	Afghanistan	Asia	1962
3	Afghanistan	Asia	1967
4	Afghanistan	Asia	1972

	country	continent	year
0	Afghanistan	Asia	1952
1	Afghanistan	Asia	1957
2	Afghanistan	Asia	1962
3	Afghanistan	Asia	1967
4	Afghanistan	Asia	1972

	lifeExp	pop	gdpPercap
0	28.801	8425333	779.445314
1	30.332	9240934	820.853030
2	31.997	10267083	853.100710
3	34.020	11537966	836.197138
4	36.088	13079460	739.981106

	lifeExp	pop	gdpPercap
0	28.801	8425333	779.445314
1	30.332	9240934	820.853030
2	31.997	10267083	853.100710
3	34.020	11537966	836.197138
4	36.088	13079460	739.981106

	country	year	pop
0	Afghanistan	1952	8425333
1	Afghanistan	1957	9240934
2	Afghanistan	1962	10267083
3	Afghanistan	1967	11537966
4	Afghanistan	1972	13079460

```
# A) df.iloc[:,0:6:]
subset = df.iloc[:,0:6:]
display(subset.head())
```

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
4	Afghanistan	Asia	1972	36.088	13079460	739.981106

```
# b) df.iloc[:,0::2]
subset = df.iloc[:,0::2]
display(subset.head())
```

	country	year	pop
0	Afghanistan	1952	8425333
1	Afghanistan	1957	9240934
2	Afghanistan	1962	10267083
3	Afghanistan	1967	11537966
4	Afghanistan	1972	13079460

```
# c) df.iloc[:,:6:2]
subset = df.iloc[:,:6:2]
display(subset.head())
```

	country	year	pop
0	Afghanistan	1952	8425333
1	Afghanistan	1957	9240934
2	Afghanistan	1962	10267083
3	Afghanistan	1967	11537966
4	Afghanistan	1972	13079460

```
# d) df.iloc[:,::2]
subset = df.iloc[:,::2]
display(subset.head())
```

	country	year	pop
0	Afghanistan	1952	8425333
1	Afghanistan	1957	9240934
2	Afghanistan	1962	10267083
3	Afghanistan	1967	11537966
4	Afghanistan	1972	13079460

```
# e) df.iloc[:,::]
subset = df.iloc[:,::]
display(subset.head())
```

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
4	Afghanistan	Asia	1972	36.088	13079460	739.981106

O que acontecerá se usar o método de fatiamento com dois-pontos, mas deixar de especificar um valor? Por exemplo, qual será o resultado obtido nos casos abaixo?

A) df.iloc[:,0:6:] Início: 0 Fim: 6 Passo: 1 (padrão) Resultado: Seleciona todas as linhas (:) e as colunas de índice 0 a 5 (não inclui o índice 6).

b) df.iloc[:,0::2] Início: 0 Fim: Não especificado, assume o total de colunas. Passo: 2 Resultado: Seleciona todas as linhas (:) e as colunas de índice 0, 2, 4, etc., até o final.

c) df.iloc[:,:6:2] Início: Não especificado, assume 0. Fim: 6 Passo: 2 Resultado: Seleciona todas as linhas (:) e as colunas de índice 0, 2, 4 (até o índice 5, sem incluir 6).

d) df.iloc[:,::2] Início: Não especificado, assume 0. Fim: Não especificado, assume o total de colunas. Passo: 2 Resultado: Seleciona todas as linhas (:) e as colunas de índice 0, 2, 4, etc., até o final.

e) df.iloc[:,::] Início: Não especificado, assume 0. Fim: Não especificado, assume o total de colunas. Passo: Não especificado, assume 1. Resultado: Seleciona todas as linhas (:) e todas as colunas (:).

```
!pip install dask
!pip install pyarrow
```

```
Requirement already satisfied: dask in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (2025.4.1)
Requirement already satisfied: click>=8.1 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from dask) (8.1.8)
Requirement already satisfied: cloudpickle>=3.0.0 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from dask) (3.1.1)
Requirement already satisfied: fsspec>=2021.09.0 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from dask) (2025.3.2)
Requirement already satisfied: packaging>=20.0 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from dask) (24.2)
Requirement already satisfied: partd>=1.4.0 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from dask) (1.4.2)
Requirement already satisfied: pyyaml>=5.3.1 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from dask) (6.0.2)
```

```
Requirement already satisfied: toolz>=0.10.0 in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from dask) (1.0.0)
Requirement already satisfied: colorama in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from click>=8.1->dask) (0.4.6)
Requirement already satisfied: locket in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (from partd>=1.4.0->dask) (1.0.0)

[notice] A new release of pip is available: 24.3.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
Requirement already satisfied: pyarrow in d:\bruno\fatec cdn\tecnicasprogramacaocienciadados\fatecvenv\lib\site-packages (20.0.0)

[notice] A new release of pip is available: 24.3.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
df2 = pd.read_csv(r"gapminder.tsv", sep="\t", usecols=[0,2])
display(df2.head())
```

	country	year
0	Afghanistan	1952
1	Afghanistan	1957
2	Afghanistan	1962
3	Afghanistan	1967
4	Afghanistan	1972

```
chunks = pd.read_csv(r"gapminder.tsv", sep="\t", chunksize=100)
total_rows = 0
for chunk in chunks:
    total_rows += len(chunk)
print("Total rows: ", total_rows)
```

```
Total rows:  1704
```

```
# Dask
import pandas as pd
import dask.dataframe as dd

dados = {
 'nome': ['Ana', 'Bruno', 'Carla', 'Daniel'],
 'idade': [25, 30, 22, 40],
 'salario': [3500, 4200, 3000, 5000]
}

df_pandas = pd.DataFrame(dados)
```