
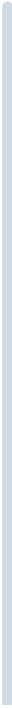


```
!pip install pandas
!pip install polars
!pip install dask
!pip install pyarrow
```



-----	14.2/35.0	MB	2.2	MB/s	eta	0:00:10
-----	14.7/35.0	MB	2.3	MB/s	eta	0:00:10
-----	15.5/35.0	MB	2.3	MB/s	eta	0:00:09
-----	16.0/35.0	MB	2.3	MB/s	eta	0:00:09
-----	16.8/35.0	MB	2.3	MB/s	eta	0:00:08
-----	17.3/35.0	MB	2.3	MB/s	eta	0:00:08
-----	17.8/35.0	MB	2.3	MB/s	eta	0:00:08
-----	18.4/35.0	MB	2.4	MB/s	eta	0:00:08
-----	19.1/35.0	MB	2.4	MB/s	eta	0:00:07
-----	19.7/35.0	MB	2.4	MB/s	eta	0:00:07
-----	20.4/35.0	MB	2.4	MB/s	eta	0:00:07
-----	21.0/35.0	MB	2.4	MB/s	eta	0:00:06



```
[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
```

```
# 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

```
# Dask com o df de aula - gapminder
```

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

```
df_dask = dd.from_pandas(df, npartitions=2)
print("Primeiras linhas no Dask:")
display(df_dask.head().round({
    'lifeExp': 2,
    'gdpPercap': 2
})))
```

```
media_populacao = df_dask['pop'].mean()
print("\nMédia da populacao:", f"{media_populacao.compute():.0f}".replace(",", "x").replace(".", ",").replace("x", "."))
```

↗ Primeiras linhas no Dask:

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.80	8425333	779.45
1	Afghanistan	Asia	1957	30.33	9240934	820.85
2	Afghanistan	Asia	1962	32.00	10267083	853.10
3	Afghanistan	Asia	1967	34.02	11537966	836.20
4	Afghanistan	Asia	1972	36.09	13079460	739.98

Média da populacao: 29.601.212

```
# Polars para o df da aula - gapminder

import polars as pl

# Criar um DataFrame diretamente com Polars
df_polars = pl.DataFrame(df)

print("DataFrame em Polars (primeiras linhas):")
display(df_polars.head().select(
    pl.col('country'),
    pl.col('continent'),
    pl.col('year'),
    pl.col('lifeExp').round(2),
    pl.col('pop').cast(pl.Int64),
    pl.col('gdpPercap').round(2)
))

# Format population mean without scientific notation
populacao = df_polars.select(pl.col('pop').mean()).item()
print("\nMédia da populacao:", f"{populacao:,.0f}".replace(",","x").replace(".",",").replace("x", "."))
```



DataFrame em Polars (primeiras linhas):

shape: (5, 6)

country	continent	year	lifeExp	pop	gdpPercap
str	str	i64	f64	i64	f64
"Afghanistan"	"Asia"	1952	28.8	8425333	779.45
"Afghanistan"	"Asia"	1957	30.33	9240934	820.85
"Afghanistan"	"Asia"	1962	32.0	10267083	853.1
"Afghanistan"	"Asia"	1967	34.02	11537966	836.2
"Afghanistan"	"Asia"	1972	36.09	13079460	739.98

Média da populacao: 29.601.212