Connected to base (Python 3.12.5)

In [ ]:
```
from main import (
    load_dataset_pl,
    pl_describe,
    get_mean,
    get_median,
    get_std,
    save_to_md,
    create_boxplot,
)

import polars as pl
import pandas as pd
from pyinstrument import Profiler
```

In [ ]:
```
data = "https://raw.githubusercontent.com/anlane611/datasets/main/population.csv"
dataframe = load_dataset_pl(data)

print(dataframe)
```

```
shape: (100_000, 3)
```

Y	X1	X2
f64	i64	i64
21.97361	4	1
12.387638	3	1
12.665114	3	1
16.753335	1	1
22.435229	2	1
…	…	…
23.310289	2	1
20.406937	4	1
25.335073	2	1
29.479947	4	1
16.47385	2	1

In [ ]:
```
# Define loading dataset and decribe functions using pandas:
def load_dataset_pd(dataset):
    df_pd = pd.read_csv(dataset)
    return df_pd


def describe_pd(df):
    df_description = df.describe()
    return df_description
```

In [ ]:
```
# Use Profiler to compare pandas and polars
with Profiler(interval=0.1) as profiler:
    check_pl = data
    df = load_dataset_pl(check_pl)
    print(df.shape)
    print(pl_describe(df))
    print(df["Y"].mean)
    print(df["Y"].median)

profiler.print()

with Profiler(interval=0.1) as profiler:
    check_pd = data
    df = load_dataset_pd(check_pd)
    print(df.shape)
    print(describe_pd(df))
    print(df["Y"].mean)
    print(df["Y"].mean)

profiler.print()
```

```
(100000, 3)
shape: (9, 4)
```

statistic	Y	X1	X2
str	f64	f64	f64
count	100000.0	100000.0	100000.0
null_count	0.0	0.0	0.0
mean	19.975793	3.004	0.99193
std	5.004965	1.379131	0.08947
min	-3.058222	1.0	0.0
25%	16.590563	2.0	1.0
50%	19.971087	3.0	1.0
75%	23.351626	4.0	1.0
max	45.856084	5.0	1.0

```
<bound method Series.mean of shape: (100_000,)
Series: 'Y' [f64]
[
        21.97361
        12.387638
        12.665114
        16.753335
        22.435229
        …
        23.310289
        20.406937
        25.335073
        29.479947
        16.47385
]>
<bound method Series.median of shape: (100_000,)
Series: 'Y' [f64]
[
        21.97361
        12.387638
        12.665114
        16.753335
        22.435229
        …
        23.310289
        20.406937
        25.335073
        29.479947
        16.47385
]>

 _     ._    __/__   _ _  _  _ _/_   Recorded: 17:36:53  Samples:  2
/_//_/// /_\ / //_// / //_'/ //     Duration: 0.350     CPU time: 0.031
/   _/                      v4.7.3

Profile at <ipython-input-4-3d746cf6ed75>:2

0.260 <module>  <ipython-input-4-3d746cf6ed75>:1
└─ 0.260 load_dataset_pl  main.py:6
      └─ 0.260 wrapper  polars/_utils/deprecation.py:86
            [22 frames hidden]  polars, urllib, http, socket, ssl
               0.160 _SSLSocket.read  <built-in>
               0.100 _SSLSocket.read  <built-in>


(100000, 3)
                Y              X1             X2
count  100000.000000  100000.000000  100000.00000
mean       19.975793       3.004000       0.99193
std         5.004965       1.379131       0.08947
min        -3.058220       1.000000       0.00000
25%        16.590524       2.000000       1.00000
50%        19.971020       3.000000       1.00000
75%        23.351637       4.000000       1.00000
max        45.856084       5.000000       1.00000
<bound method Series.mean of 0        21.973610
1         12.387638
2         12.665114
3         16.753335
4         22.435229
           ...
99995     23.310289
99996     20.406937
99997     25.335073
99998     29.479947
99999     16.473850
Name: Y, Length: 100000, dtype: float64>
<bound method Series.mean of 0        21.973610
1         12.387638
2         12.665114
3         16.753335
4         22.435229
           ...
99995     23.310289
99996     20.406937
99997     25.335073
99998     29.479947
99999     16.473850
Name: Y, Length: 100000, dtype: float64>

 _     ._    __/__   _ _  _  _ _/_   Recorded: 17:36:53  Samples:  2
/_//_/// /_\ / //_// / //_'/ //     Duration: 0.263     CPU time: 0.042
/   _/                      v4.7.3

Profile at <ipython-input-4-3d746cf6ed75>:12

0.203 <module>  <ipython-input-4-3d746cf6ed75>:1
└─ 0.203 load_dataset_pd  <ipython-input-3-04637061a9a0>:2
   └─ 0.203 read_csv  pandas/io/parsers/readers.py:868
         [10 frames hidden]  pandas, http, socket, ssl
            0.203 _SSLSocket.read  <built-in>
```

In [ ]:
```
# Print descriptive statistics
print(pl_describe(dataframe))
print(get_mean(dataframe, "Y"))
print(get_median(dataframe, "Y"))
print(get_std(dataframe, "Y"))
```

```
shape: (9, 4)
```

statistic	Y	X1	X2
str	f64	f64	f64
count	100000.0	100000.0	100000.0
null_count	0.0	0.0	0.0
mean	19.975793	3.004	0.99193
std	5.004965	1.379131	0.08947
min	-3.058222	1.0	0.0
25%	16.590563	2.0	1.0
50%	19.971087	3.0	1.0
75%	23.351626	4.0	1.0
max	45.856084	5.0	1.0

```
19.97579252039033
19.971020000166825
5.0049645594229216
```

In [ ]:
```
# Define test functions
def test_mean():
    """Test the get_mean function"""
    assert round(get_mean(dataframe, "Y"), 3) == 19.976


def test_median():
    """Test the get_median function"""
    assert round(get_median(dataframe, "Y"), 3) == 19.971


def test_std():
    """Test the get_std function"""
    assert round(get_std(dataframe, "Y"), 3) == 5.005
```

In [ ]:
```
if __name__ == "__main__":
    test_mean()
    test_median()
    test_std()
    create_boxplot(dataframe["Y"], "boxplot.png")
    mean_y = get_mean(dataframe, "Y")
    median_y = get_median(dataframe, "Y")
    std_y = get_std(dataframe, "Y")
    save_to_md(mean_y, median_y, std_y)
```