```
#importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#view the dataset
df=pd.read_csv('/content/world_population.csv')
df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 234,\n  \"fields\": [\
n    {\n      \"column\": \"Rank\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 67,\n        \"min\": 1,\n
\"max\": 234,\n        \"num_unique_values\": 234,\n
\"samples\": [\n          184,\n          20,\n          218\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"CCA3\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 234,\n        \"samples\": [\n
\"GUF\",\n          \"THA\",\n          \"SMR\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Country/Territory\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 234,\n        \"samples\": [\n
\"French Guiana\",\n          \"Thailand\",\n          \"San Marino\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Capital\",\n      \"properties\": {\n        \"dtype\": \"string\",\
n        \"num_unique_values\": 234,\n        \"samples\": [\n
\"Cayenne\",\n          \"Bangkok\",\n          \"San Marino\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Continent\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 6,\n        \"samples\": [\n
\"Asia\",\n          \"Europe\",\n          \"South America\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"2022 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
136766424,\n        \"min\": 510,\n        \"max\": 1425887337,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
304557,\n          71697030,\n          33660\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2020 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
135589876,\n        \"min\": 520,\n        \"max\": 1424929781,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
290969,\n          71475664,\n          34007\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2015 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
```

```
130404992,\n        \"min\": 564,\n        \"max\": 1393715448,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
257026,\n          70294397,\n          33570\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2010 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
124218487,\n        \"min\": 596,\n        \"max\": 1348191368,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
228453,\n          68270489,\n          31608\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2000 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
111698206,\n        \"min\": 651,\n        \"max\": 1264099069,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
164351,\n          63066603,\n          26823\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"1990 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
97832173,\n        \"min\": 700,\n        \"max\": 1153704252,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
113931,\n          55228410,\n          23132\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"1980 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
81785186,\n        \"min\": 733,\n        \"max\": 982372466,\n
\"num_unique_values\": 234,\n        \"samples\": [\n          66825,\
n          45737753,\n          21346\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"1970 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
67795091,\n        \"min\": 752,\n        \"max\": 822534450,\n
\"num_unique_values\": 234,\n        \"samples\": [\n          46484,\
n          35791728,\n          18169\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Area (km\\u00b2)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1761840,\n        \"min\": 1,\n        \"max\": 17098242,\n
\"num_unique_values\": 233,\n        \"samples\": [\n
245857,\n          241550,\n          752612\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Density (per km\\u00b2)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2066.1219036046123,\n        \"min\": 0.0261,\n        \"max\":
23172.2667,\n        \"num_unique_values\": 234,\n        \"samples\":
[\n          3.6459,\n          139.7276,\n          551.8033\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Growth Rate\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.01338498453399876,\n        \"min\": 0.912,\n        \"max\":
```

```
1.0691,\n        \"num_unique_values\": 180,\n        \"samples\": [\n
1.0038,\n          1.0005,\n          1.0232\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"World Population Percentage\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.7149767768102147,\n        \"min\": 0.0,\n        \"max\": 17.88,\n
\"num_unique_values\": 70,\n        \"samples\": [\n          0.48,\n
0.52,\n          0.47\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


INSPECT THE DATASET

```
from google.colab import drive
drive.mount('/content/drive')

#view top rows of a dataset
df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 234,\n  \"fields\": [\
n    {\n      \"column\": \"Rank\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 67,\n        \"min\": 1,\n
\"max\": 234,\n        \"num_unique_values\": 234,\n
\"samples\": [\n          184,\n          20,\n          218\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"CCA3\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 234,\n        \"samples\": [\n
\"GUF\",\n          \"THA\",\n          \"SMR\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Country/Territory\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 234,\n        \"samples\": [\n
\"French Guiana\",\n          \"Thailand\",\n          \"San Marino\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Capital\",\n      \"properties\": {\n        \"dtype\": \"string\",\
n        \"num_unique_values\": 234,\n        \"samples\": [\n
\"Cayenne\",\n          \"Bangkok\",\n          \"San Marino\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Continent\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 6,\n        \"samples\": [\n
\"Asia\",\n          \"Europe\",\n          \"South America\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"2022 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
136766424,\n        \"min\": 510,\n        \"max\": 1425887337,\n
```

```
\"num_unique_values\": 234,\n        \"samples\": [\n
304557,\n          71697030,\n          33660\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2020 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
135589876,\n        \"min\": 520,\n        \"max\": 1424929781,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
290969,\n          71475664,\n          34007\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2015 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
130404992,\n        \"min\": 564,\n        \"max\": 1393715448,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
257026,\n          70294397,\n          33570\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2010 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
124218487,\n        \"min\": 596,\n        \"max\": 1348191368,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
228453,\n          68270489,\n          31608\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2000 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
111698206,\n        \"min\": 651,\n        \"max\": 1264099069,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
164351,\n          63066603,\n          26823\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"1990 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
97832173,\n        \"min\": 700,\n        \"max\": 1153704252,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
113931,\n          55228410,\n          23132\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"1980 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
81785186,\n        \"min\": 733,\n        \"max\": 982372466,\n
\"num_unique_values\": 234,\n        \"samples\": [\n          66825,\
n          45737753,\n          21346\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"1970 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
67795091,\n        \"min\": 752,\n        \"max\": 822534450,\n
\"num_unique_values\": 234,\n        \"samples\": [\n          46484,\
n          35791728,\n          18169\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Area (km\\u00b2)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1761840,\n        \"min\": 1,\n        \"max\": 17098242,\n
\"num_unique_values\": 233,\n        \"samples\": [\n
```

```
245857,\n          241550,\n          752612\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Density (per km\\u00b2)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2066.1219036046123,\n        \"min\": 0.0261,\n        \"max\":
23172.2667,\n        \"num_unique_values\": 234,\n        \"samples\":
[\n          3.6459,\n          139.7276,\n          551.8033\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Growth Rate\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.01338498453399876,\n        \"min\": 0.912,\n        \"max\":
1.0691,\n        \"num_unique_values\": 180,\n        \"samples\": [\n
1.0038,\n          1.0005,\n          1.0232\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"World Population Percentage\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.7149767768102147,\n        \"min\": 0.0,\n        \"max\": 17.88,\n
\"num_unique_values\": 70,\n        \"samples\": [\n          0.48,\n
0.52,\n          0.47\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


```
#view the top 20 rows in dataset
df.head(20)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 234,\n  \"fields\": [\
n    {\n      \"column\": \"Rank\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 67,\n        \"min\": 1,\n
\"max\": 234,\n        \"num_unique_values\": 234,\n
\"samples\": [\n          184,\n          20,\n          218\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"CCA3\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 234,\n        \"samples\": [\n
\"GUF\",\n          \"THA\",\n          \"SMR\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Country/Territory\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 234,\n        \"samples\": [\n
\"French Guiana\",\n          \"Thailand\",\n          \"San Marino\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Capital\",\n      \"properties\": {\n        \"dtype\": \"string\",\
n        \"num_unique_values\": 234,\n        \"samples\": [\n
\"Cayenne\",\n          \"Bangkok\",\n          \"San Marino\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Continent\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 6,\n        \"samples\": [\n
\"Asia\",\n          \"Europe\",\n          \"South America\"\n
```

```
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"2022 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
136766424,\n        \"min\": 510,\n        \"max\": 1425887337,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
304557,\n          71697030,\n          33660\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2020 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
135589876,\n        \"min\": 520,\n        \"max\": 1424929781,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
290969,\n          71475664,\n          34007\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2015 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
130404992,\n        \"min\": 564,\n        \"max\": 1393715448,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
257026,\n          70294397,\n          33570\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2010 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
124218487,\n        \"min\": 596,\n        \"max\": 1348191368,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
228453,\n          68270489,\n          31608\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2000 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
111698206,\n        \"min\": 651,\n        \"max\": 1264099069,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
164351,\n          63066603,\n          26823\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"1990 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
97832173,\n        \"min\": 700,\n        \"max\": 1153704252,\n
\"num_unique_values\": 234,\n        \"samples\": [\n
113931,\n          55228410,\n          23132\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"1980 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
81785186,\n        \"min\": 733,\n        \"max\": 982372466,\n
\"num_unique_values\": 234,\n        \"samples\": [\n          66825,\
n          45737753,\n          21346\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"1970 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
67795091,\n        \"min\": 752,\n        \"max\": 822534450,\n
\"num_unique_values\": 234,\n        \"samples\": [\n          46484,\
n          35791728,\n          18169\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

```
n    },\n    {\n      \"column\": \"Area (km\\u00b2)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1761840,\n        \"min\": 1,\n        \"max\": 17098242,\n
\"num_unique_values\": 233,\n        \"samples\": [\n
245857,\n          241550,\n          752612\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Density (per km\\u00b2)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2066.1219036046123,\n        \"min\": 0.0261,\n        \"max\":
23172.2667,\n        \"num_unique_values\": 234,\n        \"samples\":
[\n          3.6459,\n          139.7276,\n          551.8033\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Growth Rate\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.01338498453399876,\n        \"min\": 0.912,\n        \"max\":
1.0691,\n        \"num_unique_values\": 180,\n        \"samples\": [\n
1.0038,\n          1.0005,\n          1.0232\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"World Population Percentage\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.7149767768102147,\n        \"min\": 0.0,\n        \"max\": 17.88,\n
\"num_unique_values\": 70,\n        \"samples\": [\n          0.48,\n
0.52,\n          0.47\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


```
#view the bottom rows in dataset
df.tail()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 5,\n  \"fields\": [\n
{\n      \"column\": \"Rank\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 78,\n        \"min\": 46,\n
\"max\": 226,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          172,\n          74,\n          46\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"CCA3\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 5,\n
\"samples\": [\n          \"ESH\",\n          \"ZWE\",\n
\"YEM\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Country/Territory\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          \"Western Sahara\",\n          \"Zimbabwe\",\n
\"Yemen\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Capital\",\n      \"properties\": {\n        \"dtype\": \"string\",\
n        \"num_unique_values\": 5,\n        \"samples\": [\n
\"El Aai\\u00fan\",\n          \"Harare\",\n          \"Sanaa\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Continent\",\n
```

```
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n
\"Oceania\",\n          \"Africa\",\n          \"Asia\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2022 Population\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
14189403,\n        \"min\": 11572,\n        \"max\": 33696614,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          575986,\n
16320537,\n          33696614\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"2020 Population\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 13569163,\n        \"min\":
11655,\n        \"max\": 32284046,\n        \"num_unique_values\": 5,\
n        \"samples\": [\n          556048,\n          15669666,\n
32284046\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2015 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 11953733,\n        \"min\": 12182,\n
\"max\": 28516545,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          491824,\n          14154937,\n
28516545\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2010 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 10378111,\n        \"min\": 13142,\n
\"max\": 24743946,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          413296,\n          12839771,\n
24743946\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2000 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 7979353,\n        \"min\": 14723,\n
\"max\": 18628700,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          270375,\n          11834676,\n
18628700\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"1990 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 5989882,\n        \"min\": 13454,\n
\"max\": 13375121,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          178529,\n          10113893,\n
13375121\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"1980 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 4167059,\n        \"min\": 11315,\n
\"max\": 9204938,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          116775,\n          7049926,\n
9204938\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"1970 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3096789,\n        \"min\": 9377,\n
\"max\": 6843607,\n        \"num_unique_values\": 5,\n
```

\"samples\": [\n   76371,\n   5202918,\n
6843607\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Area (km\\u00b2)\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 281894,\n   \"min\": 142,\n
\"max\": 752612,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   266000,\n   390757,\n
527968\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Density (per km\\u00b2)\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 31.037068445296182,\n
\"min\": 2.1654,\n   \"max\": 81.493,\n
\"num_unique_values\": 5,\n   \"samples\": [\n   2.1654,\n
41.7665,\n   63.8232\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"Growth Rate\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.012522100462781818,\n
\"min\": 0.9953,\n   \"max\": 1.028,\n
\"num_unique_values\": 5,\n   \"samples\": [\n   1.0184,\n
1.0204,\n   1.0217\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"World Population Percentage\",\n   \"properties\": {\
n   \"dtype\": \"number\",\n   \"std\":
0.17615334229017626,\n   \"min\": 0.0,\n   \"max\": 0.42,\n
\"num_unique_values\": 5,\n   \"samples\": [\n   0.01,\n
0.2,\n   0.42\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
#view the bottom 20 rows in dataset
df.tail(20)
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 20,\n \"fields\": [\n
{\n   \"column\": \"Rank\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 81,\n   \"min\": 3,\n
\"max\": 234,\n   \"num_unique_values\": 20,\n
\"samples\": [\n   111,\n   46,\n   226\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"CCA3\",\n   \"properties\": {\n   \"dtype\": \"string\",\n
\"num_unique_values\": 20,\n   \"samples\": [\n
\"TKM\",\n   \"YEM\",\n   \"WLF\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Country/Territory\",\n
\"properties\": {\n   \"dtype\": \"string\",\n
\"num_unique_values\": 20,\n   \"samples\": [\n
\"Turkmenistan\",\n   \"Yemen\",\n   \"Wallis and
Futuna\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Capital\",\n   \"properties\": {\n   \"dtype\": \"string\",\
n   \"num_unique_values\": 20,\n   \"samples\": [\n

```
\"Ashgabat\",\n          \"Sanaa\",\n          \"Mata-Utu\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Continent\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 6,\n        \"samples\":
[\n          \"Asia\",\n          \"North America\",\n
\"South America\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2022 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 75636061,\n        \"min\": 510,\n
\"max\": 338289857,\n        \"num_unique_values\": 20,\n
\"samples\": [\n          6430770,\n          33696614,\n
11572\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2020 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 75101881,\n        \"min\": 520,\n
\"max\": 335942003,\n        \"num_unique_values\": 20,\n
\"samples\": [\n          6250438,\n          32284046,\n
11655\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2015 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 72573044,\n        \"min\": 564,\n
\"max\": 324607776,\n        \"num_unique_values\": 20,\n
\"samples\": [\n          5766431,\n          28516545,\n
12182\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2010 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 69619609,\n        \"min\": 596,\n
\"max\": 311182845,\n        \"num_unique_values\": 20,\n
\"samples\": [\n          5267970,\n          24743946,\n
13142\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2000 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 63489395,\n        \"min\": 651,\n
\"max\": 282398554,\n        \"num_unique_values\": 20,\n
\"samples\": [\n          4569132,\n          18628700,\n
14723\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"1990 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 56110232,\n        \"min\": 700,\n
\"max\": 248083732,\n        \"num_unique_values\": 20,\n
\"samples\": [\n          3720278,\n          13375121,\n
13454\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"1980 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 50636444,\n        \"min\": 733,\n
\"max\": 223140018,\n        \"num_unique_values\": 20,\n
\"samples\": [\n          2862903,\n          9204938,\n
11315\n        ],\n        \"semantic_type\": \"\",\n
```

\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"1970 Population\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 45697438,\n   \"min\": 752,\n
\"max\": 200328340,\n   \"num_unique_values\": 20,\n
\"samples\": [\n   2201432,\n   6843607,\n
9377\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Area (km\\u00b2)\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 2049183,\n   \"min\": 1,\n
\"max\": 9372610,\n   \"num_unique_values\": 20,\n
\"samples\": [\n   488100,\n   527968,\n   142\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"Density (per km\\u00b2)\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
149.82293501550984,\n   \"min\": 2.1654,\n   \"max\":
510.0,\n   \"num_unique_values\": 20,\n   \"samples\": [\n
13.1751,\n   63.8232,\n   81.493\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Growth Rate\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.024569649120301587,\n   \"min\": 0.912,\n   \"max\":
1.0304,\n   \"num_unique_values\": 20,\n   \"samples\": [\n
1.014,\n   1.0217,\n   0.9953\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"World Population Percentage\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.9482975049501801,\n   \"min\": 0.0,\n   \"max\": 4.24,\n
\"num_unique_values\": 15,\n   \"samples\": [\n   0.35,\n
0.01,\n   0.08\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
#list of columns of a dataset
df.columns
```

```
Index(['Rank', 'CCA3', 'Country/Territory', 'Capital', 'Continent',
       '2022 Population', '2020 Population', '2015 Population',
       '2010 Population', '2000 Population', '1990 Population',
       '1980 Population', '1970 Population', 'Area (km²)', 'Density
(per km²)',
       'Growth Rate', 'World Population Percentage'],
      dtype='object')
```

```
#datatypes in dataset
df.dtypes
```

```
Rank                             int64
CCA3                            object
Country/Territory               object
Capital                         object
Continent                       object
```

```
2022 Population                    int64
2020 Population                    int64
2015 Population                    int64
2010 Population                    int64
2000 Population                    int64
1990 Population                    int64
1980 Population                    int64
1970 Population                    int64
Area (km²)                         int64
Density (per km²)                float64
Growth Rate                      float64
World Population Percentage      float64
dtype: object
```

```
# in this step see the how many columns and rows are there in our dataset
df.shape
```

```
(234, 17)
```

```
# summary of the dataset.in this dataset there is no null values
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 234 entries, 0 to 233
Data columns (total 17 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Rank                            234 non-null    int64  
 1   CCA3                            234 non-null    object 
 2   Country/Territory              234 non-null    object 
 3   Capital                         234 non-null    object 
 4   Continent                       234 non-null    object 
 5   2022 Population                 234 non-null    int64  
 6   2020 Population                 234 non-null    int64  
 7   2015 Population                 234 non-null    int64  
 8   2010 Population                 234 non-null    int64  
 9   2000 Population                 234 non-null    int64  
 10  1990 Population                 234 non-null    int64  
 11  1980 Population                 234 non-null    int64  
 12  1970 Population                 234 non-null    int64  
 13  Area (km²)                      234 non-null    int64  
 14  Density (per km²)               234 non-null    float64
 15  Growth Rate                     234 non-null    float64
 16  World Population Percentage     234 non-null    float64
dtypes: float64(3), int64(10), object(4)
memory usage: 31.2+ KB
```

```
# statistical summary of dataset
df.describe()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"Rank\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 84.03016654153295,\n
\"min\": 1.0,\n        \"max\": 234.0,\n        \"num_unique_values\":
6,\n        \"samples\": [\n          234.0,\n          117.5,\n
175.75\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2022 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 496185992.11852425,\n        \"min\":
234.0,\n        \"max\": 1425887337.0,\n        \"num_unique_values\":
8,\n        \"samples\": [\n          34074414.70940171,\n
5559944.5,\n          234.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"2020 Population\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 495957072.5169234,\n
\"min\": 234.0,\n        \"max\": 1424929781.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
33501070.952991452,\n          5493074.5,\n          234.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2015 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 485255639.91421485,\n        \"min\":
234.0,\n        \"max\": 1393715448.0,\n        \"num_unique_values\":
8,\n        \"samples\": [\n          31729956.243589744,\n
5307400.0,\n          234.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"2010 Population\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 469492336.38133293,\n
\"min\": 234.0,\n        \"max\": 1348191368.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
29845235.034188036,\n          4942770.5,\n          234.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2000 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 440528819.92598873,\n        \"min\":
234.0,\n        \"max\": 1264099069.0,\n        \"num_unique_values\":
8,\n        \"samples\": [\n          26269468.816239316,\n
4292907.0,\n          234.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"1990 Population\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 402349692.0246431,\n
\"min\": 234.0,\n        \"max\": 1153704252.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
22710220.790598292,\n          3825409.5,\n          234.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"1980 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 342671474.72215766,\n        \"min\":
234.0,\n        \"max\": 982372466.0,\n        \"num_unique_values\":
8,\n        \"samples\": [\n          18984616.970085472,\n
```

3141145.5,\n   234.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n \"column\": \"1970 Population\",\n   \"properties\": {\n \"dtype\": \"number\",\n   \"std\": 286910196.77459174,\n \"min\": 234.0,\n   \"max\": 822534450.0,\n \"num_unique_values\": 8,\n   \"samples\": [\n 15786908.807692308,\n   2604830.0,\n   234.0\n   ],\n   \"semantic_type\": \"\",\n \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Area (km\\u00b2)\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 5930592.102984848,\n   \"min\": 1.0,\n   \"max\": 17098242.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   581449.3846153846,\n 81199.5,\n   234.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n \"column\": \"Density (per km\\u00b2)\",\n   \"properties\": {\n \"dtype\": \"number\",\n   \"std\": 8063.201536842123,\n \"min\": 0.0261,\n   \"max\": 23172.2667,\n \"num_unique_values\": 8,\n   \"samples\": [\n 452.1270435897435,\n   95.34675,\n   234.0\n   ],\n \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Growth Rate\",\n \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 82.42763613310915,\n   \"min\": 0.01338498453399876,\n \"max\": 234.0,\n   \"num_unique_values\": 8,\n \"samples\": [\n   1.0095773504273504,\n   1.0079,\n 234.0\n   ],\n   \"semantic_type\": \"\",\n \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"World Population Percentage\",\n   \"properties\": {\n \"dtype\": \"number\",\n   \"std\": 81.9321359359815,\n \"min\": 0.0,\n   \"max\": 234.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   0.42705128205128207,\n 0.07,\n   234.0\n   ],\n   \"semantic_type\": \"\",\n \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
# we want access the single column we can use this
df['Capital']
```

```
0                  Kabul
1                 Tirana
2                Algiers
3              Pago Pago
4       Andorra la Vella
              ...
229            Mata-Utu
230            El Aaiún
231               Sanaa
232              Lusaka
233              Harare
Name: Capital, Length: 234, dtype: object
```

```
# we want access the more than one column we can use this
df[['Capital','CCA3']]
```


```
{"summary":"{\n  \"name\": \"df[['Capital','CCA3']]\",\n  \"rows\":
234,\n  \"fields\": [\n    {\n      \"column\": \"Capital\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 234,\n        \"samples\": [\n
\"Cayenne\",\n          \"Bangkok\",\n          \"San Marino\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"CCA3\",\n      \"properties\":
{\n        \"dtype\": \"string\",\n        \"num_unique_values\":
234,\n        \"samples\": [\n          \"GUF\",\n          \"THA\",\n
\"SMR\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```


```
# to access data from row with its position ,we can use this icon
df.iloc[[2,4,6,8]]
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 4,\n  \"fields\": [\n
{\n      \"column\": \"Rank\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 104,\n        \"min\": 33,\n
\"max\": 224,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          203,\n          33,\n          34\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"CCA3\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 4,\n
\"samples\": [\n          \"AND\",\n          \"ARG\",\n
\"DZA\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Country/Territory\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"Andorra\",\n          \"Argentina\",\n
\"Algeria\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Capital\",\n      \"properties\": {\n        \"dtype\": \"string\",\
n        \"num_unique_values\": 4,\n        \"samples\": [\n
\"Andorra la Vella\",\n          \"Buenos Aires\",\n
\"Algiers\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Continent\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"Europe\",\n          \"South America\",\n
\"Africa\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2022 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 26073712,\n        \"min\": 15857,\n
\"max\": 45510318,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          79824,\n          45510318,\n
44903225\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

```
\"2020 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 25525478,\n        \"min\": 15585,\n
\"max\": 45036032,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          77700,\n          45036032,\n
43451666\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2015 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 23925561,\n        \"min\": 14525,\n
\"max\": 43257065,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          71746,\n          43257065,\n
39543154\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2010 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 22294003,\n        \"min\": 13172,\n
\"max\": 41100123,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          71519,\n          41100123,\n
35856344\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2000 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 19731161,\n        \"min\": 11047,\n
\"max\": 37070774,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          66097,\n          37070774,\n
30774621\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"1990 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 17020272,\n        \"min\": 8316,\n
\"max\": 32637657,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          53569,\n          32637657,\n
25518074\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"1980 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 14010074,\n        \"min\": 6560,\n
\"max\": 28024803,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          35611,\n          28024803,\n
18739378\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"1970 Population\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 11606700,\n        \"min\": 6283,\n
\"max\": 23842803,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          19860,\n          23842803,\n
13795915\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Area (km\\u00b2)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1498882,\n        \"min\": 91,\n
\"max\": 2780400,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          468,\n          2780400,\n          2381741\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Density (per km\\u00b2)\",\n      \"properties\": {\n
```

```
\"dtype\": \"number\",\n        \"std\": 89.39093527084648,\n
\"min\": 16.3683,\n        \"max\": 174.2527,\n
\"num_unique_values\": 4,\n        \"samples\": [\n
170.5641,\n          16.3683,\n          18.8531\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Growth Rate\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.004991659710623951,\n        \"min\": 1.0052,\n        \"max\":
1.0164,\n        \"num_unique_values\": 4,\n        \"samples\": [\n
1.01,\n          1.0052,\n          1.0164\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"World Population Percentage\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.3262284475639732,\n        \"min\": 0.0,\n        \"max\": 0.57,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          0.56,\n
0.0,\n          0.57\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```


CHECKING FOR MISSING VALUES

```
#In this dataset there is no null values
df.isnull()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 234,\n  \"fields\": [\
n    {\n      \"column\": \"Rank\",\n      \"properties\": {\n
\"dtype\": \"boolean\",\n        \"num_unique_values\": 1,\n
\"samples\": [\n          false\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"CCA3\",\n      \"properties\": {\n
\"dtype\": \"boolean\",\n        \"num_unique_values\": 1,\n
\"samples\": [\n          false\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Country/Territory\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Capital\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Continent\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"2022 Population\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
```

```
}\n    },\n    {\n      \"column\": \"2020 Population\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"2015 Population\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"2010 Population\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"2000 Population\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"1990 Population\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"1980 Population\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"1970 Population\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Area (km\\u00b2)\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Density (per km\\u00b2)\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Growth Rate\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"World Population Percentage\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          false\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```

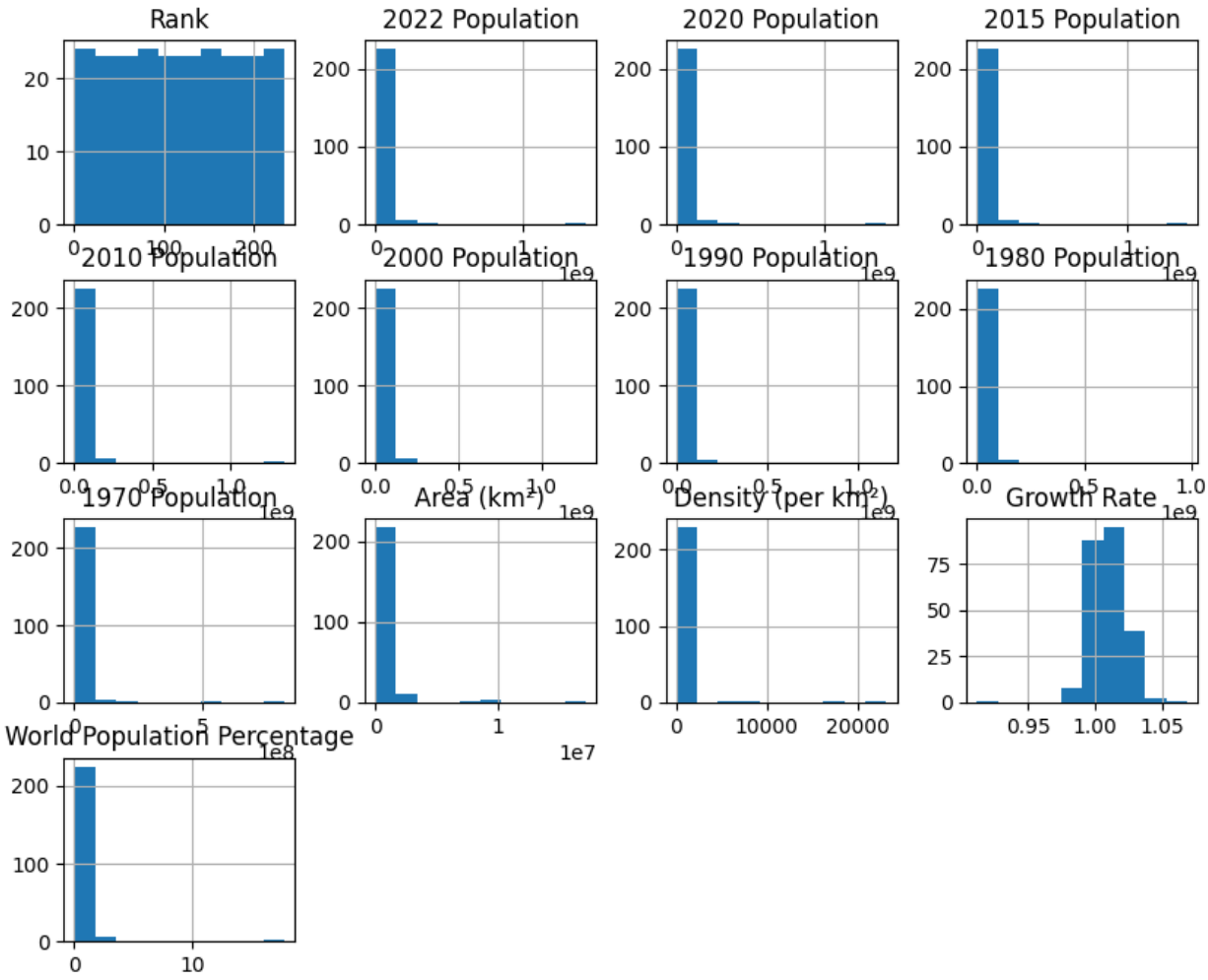

```
# to know the count of the null values in each column
df.isnull().sum()
```

```
Rank                                0
CCA3                                0
Country/Territory                   0
Capital                             0
Continent                           0
2022 Population                     0
2020 Population                     0
2015 Population                     0
2010 Population                     0
2000 Population                     0
1990 Population                     0
1980 Population                     0
1970 Population                     0
Area (km²)                          0
Density (per km²)                   0
Growth Rate                         0
World Population Percentage         0
dtype: int64
```

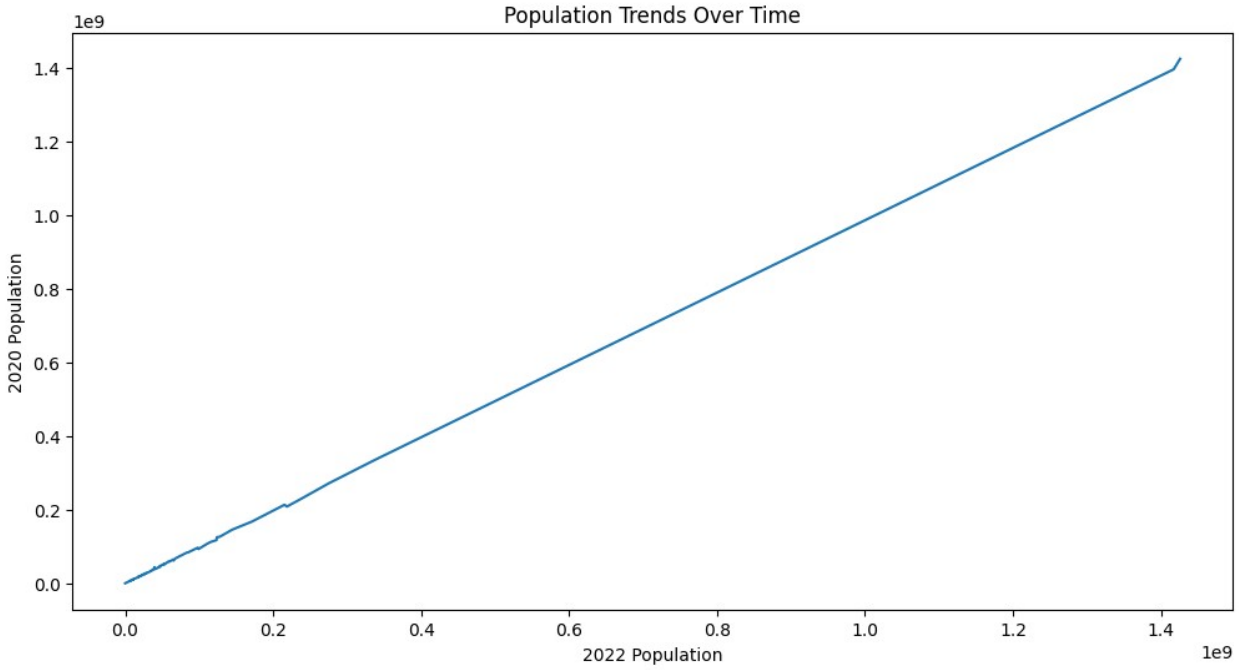
*in this dataset there is no null values.that's why not applying the dropna method,dropna(axis=1)and also fillna method also.

DATA VISUALIZATION

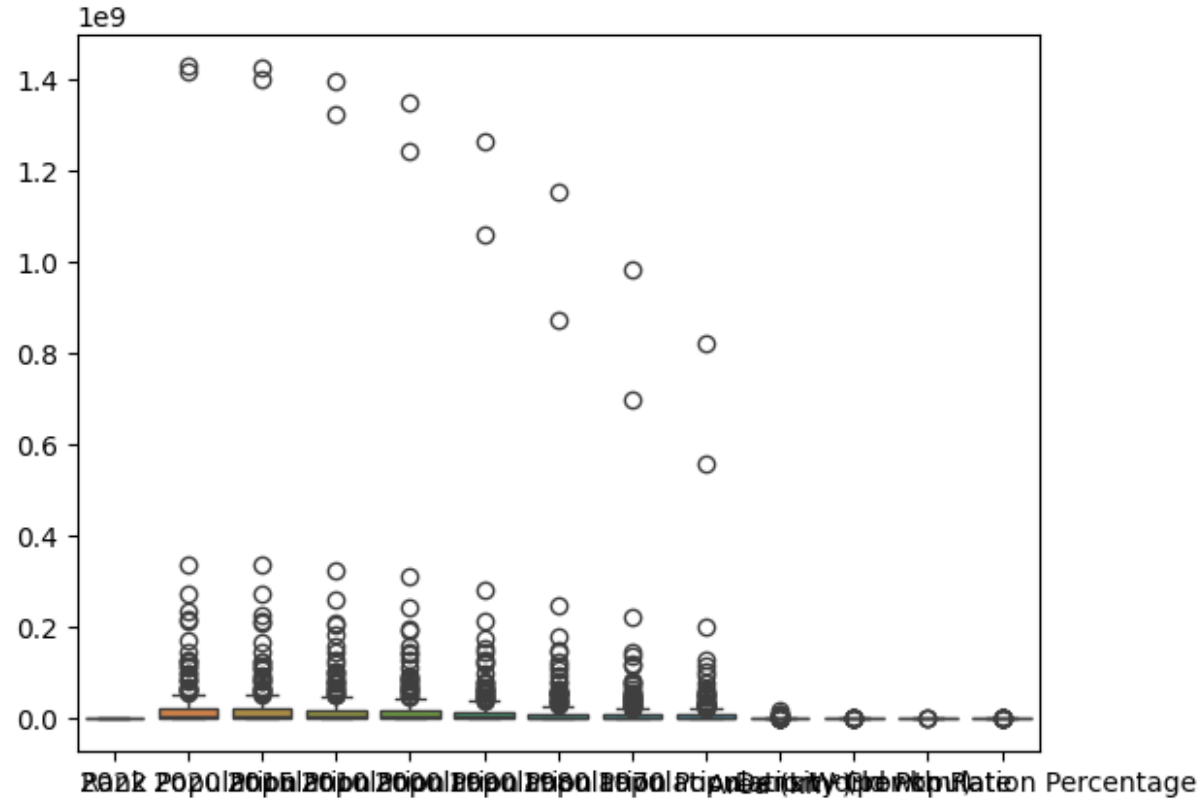
```
#Histogram
df.hist(figsize=(10, 8))
plt.show()
```



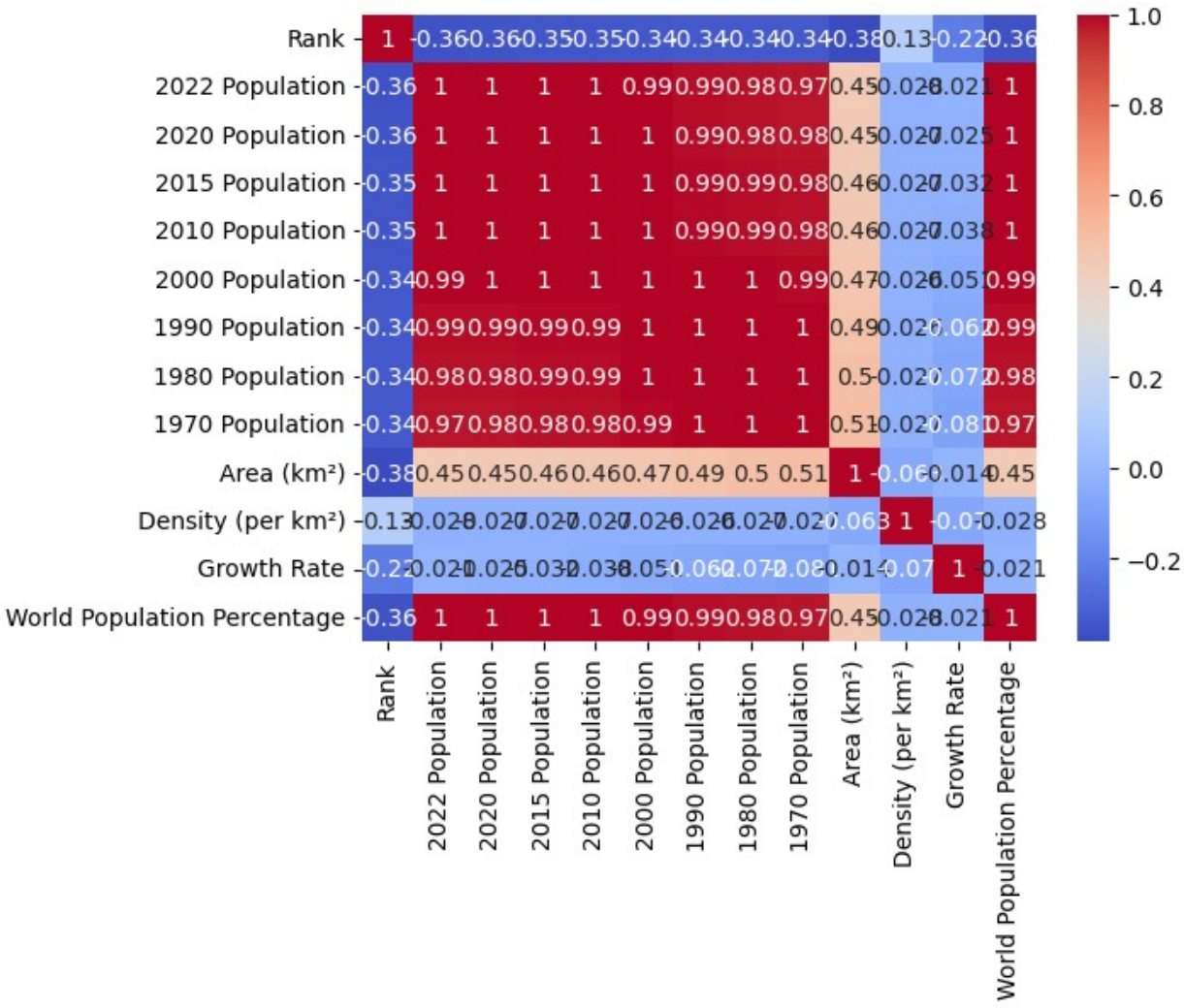

```
# Visualizing population trends over time
plt.figure(figsize=(12,6))
sns.lineplot(x=df["2022 Population"], y=df["2020 Population"],
legend=False)
plt.title("Population Trends Over Time")
plt.xlabel("2022 Population")
plt.ylabel("2020 Population")
plt.show()
```



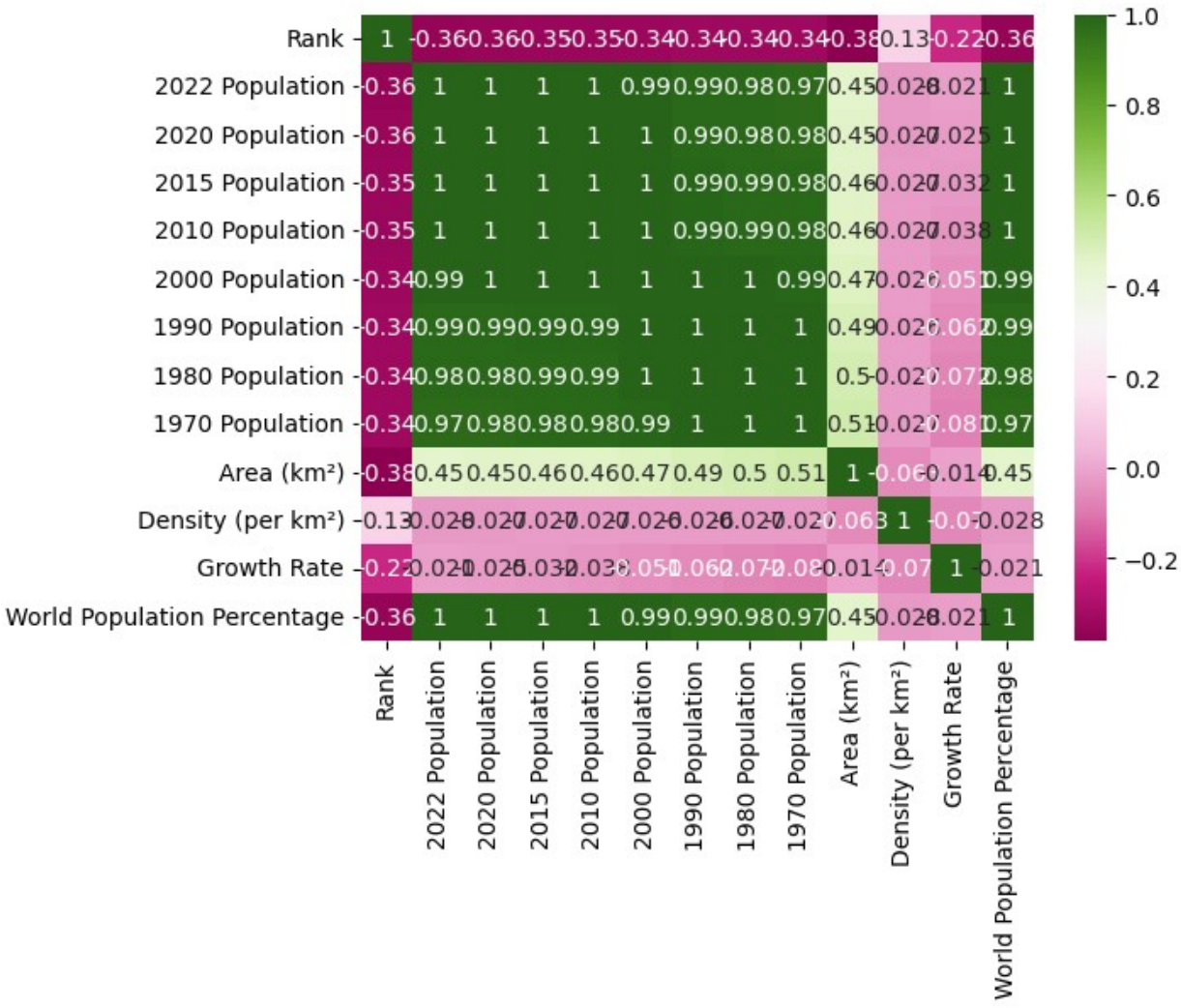

```
# Box Plot
sns.boxplot(data=df)
plt.show()
```



```
#Correlation HeatMap
numeric_df = df.select_dtypes(include=np.number)
corr=numeric_df.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.show()
```

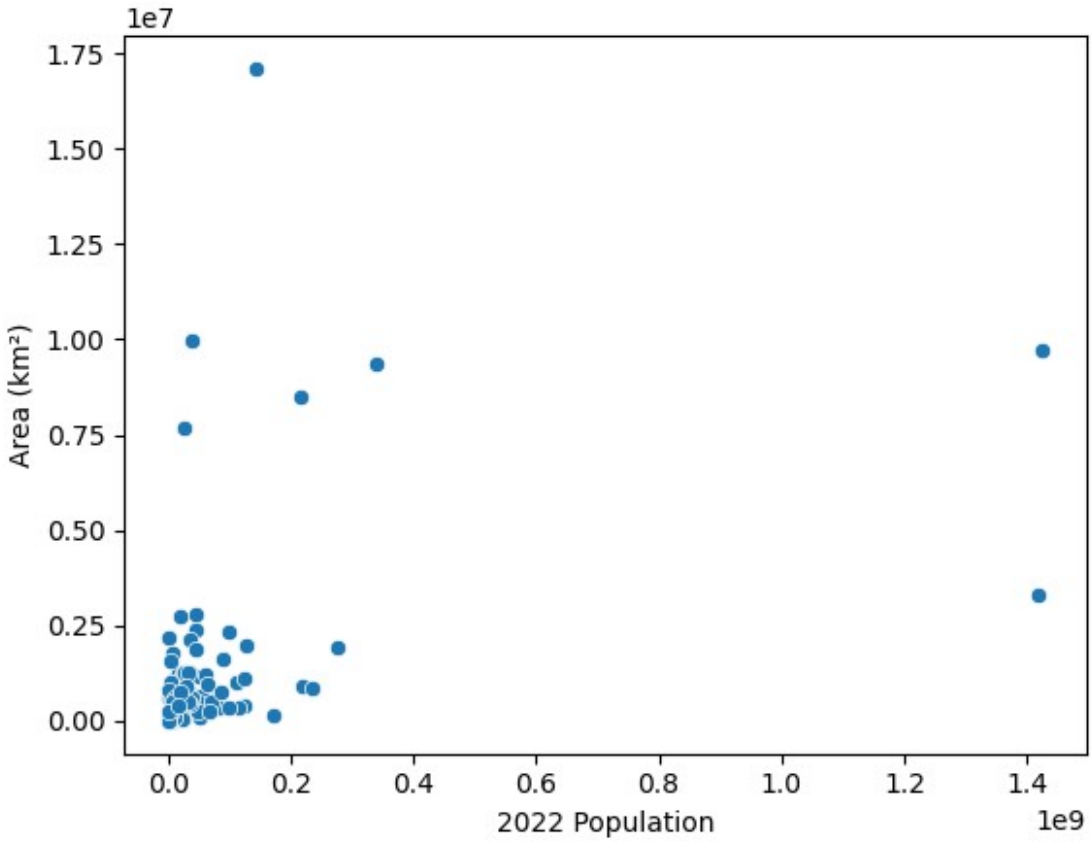



```
#another type of heatmap
numeric_df = df.select_dtypes(include=np.number)
corr=numeric_df.corr()
sns.heatmap(corr, annot=True, cmap='PiYG')
plt.show()
```

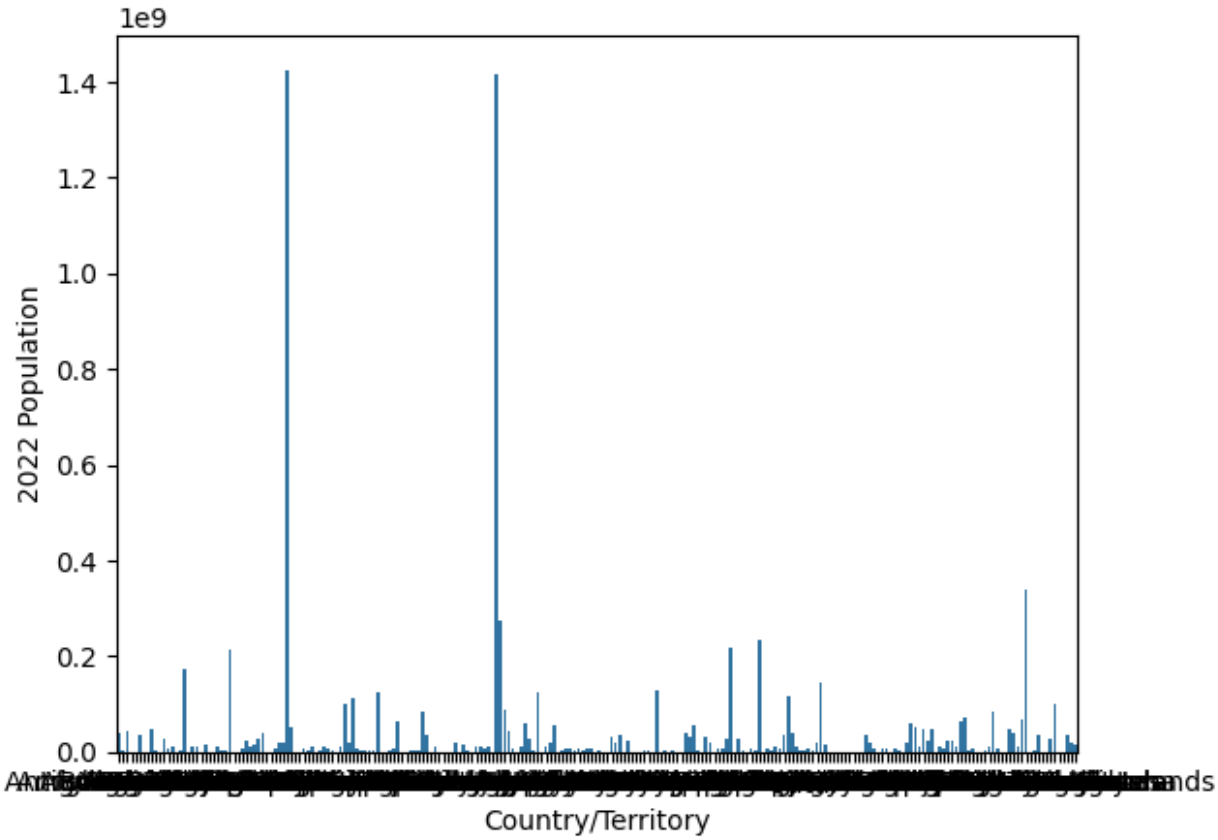


Analyze Relationships

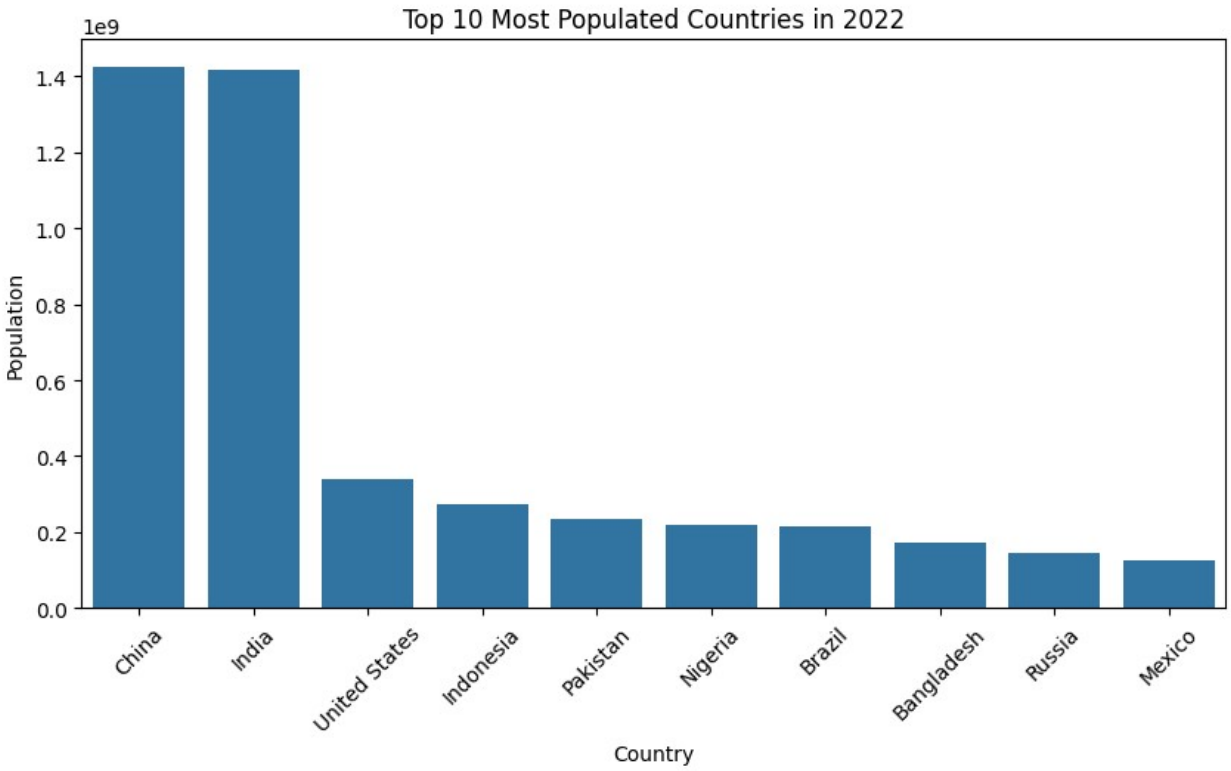
```
#Scatter Plot for numerical variables
sns.scatterplot(x='2022 Population', y='Area (km²)', data=df)
plt.show()
```




```
#Bar Chart for Categorical Variables
sns.barplot(x='Country/Territory', y='2022 Population', data=df)
plt.show()
```




```
latest_year_column = '2022 Population'
latest_year = int(latest_year_column[:4])
top_countries = df.sort_values(by=latest_year_column,
ascending=False).head(10) # Sort by latest population
plt.figure(figsize=(10, 5))
sns.barplot(x=top_countries["Country/Territory"],
y=top_countries[latest_year_column]) # Use 'Country/Territory' column
for x-axis
plt.xticks(rotation=45)
plt.title(f"Top 10 Most Populated Countries in {latest_year}")
plt.xlabel("Country")
plt.ylabel("Population")
plt.show()
```




```
#Handle Missing or incorrect Dta
df.dropna(inplace=True)  # To drop rows with missing values
print(df.fillna('2022 Population', inplace=True))# Replace missing
values with a specific value
```

```
None
```

```
#Check for Duplicates
print(df.duplicated().sum())
df = df.drop_duplicates()
```

```
0
```

#MODELS

```
#NORMALIZATION
from sklearn.model_selection import train_test_split
X = df[['2022 Population', 'Area (km²)']]  # Replace with relevant
features
y = df['World Population Percentage']  # Replace with the target
variable
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
print(X_train)
print(X_test)
```

```
print(y_train)
print(y_test)
```

```
     2022 Population  Area (km²)
117          2750055       65300
155          4576298      309500
148         26207977     1267000
158          5250072        6220
231         33696614      527968
..               ...         ...
106         54027487      580367
14            409984       13943
92        1417173173     3287590
179           222382        2842
102        123951692      377930

[187 rows x 2 columns]
     2022 Population  Area (km²)
69            304557       83534
206         71697030      513120
180            33660          61
9            2780469       29743
127           367507        1128
109          6630623      199951
196         47558630      505992
93         275501339     1904569
213         85341241      783562
15           1472233         765
224         34627652      447400
55          99010212     2344858
227         28301696      916445
30           6781953      110879
201          8740472       41284
147          6948392      130373
19          11655930       30528
112          5489739       10452
225           326740       12189
24          12224110     1098581
208          8848699       56785
10            106445         180
95          44496122      438317
144         17564014       41850
220         67508936      242900
25           3233526       51209
86            808726      214969
183         17316449      196722
84          13859341      245857
18           9534954      207600
146          5185288      270467
66            929766       18272
```

```
45           5180829       51100
139         32969517      801590
125           533286         316
170         19659267      238391
164         39857145      312679
16         171186372      147570
156        235824862      881912
177             5862         242
161          6780744      406752
203         23893394       36193
124         22593590     1240192
60           3684032      117600
199           618040      163820
229            11572         142
82          17843908      108889
117     0.03
155     0.06
148     0.33
158     0.07
231     0.42
       ...
106     0.68
14      0.01
92     17.77
179     0.00
102     1.55
Name: World Population Percentage, Length: 187, dtype: float64
69     0.00
206    0.90
180    0.00
9      0.03
127    0.00
109    0.08
196    0.60
93     3.45
213    1.07
15     0.02
224    0.43
55     1.24
227    0.35
30     0.09
201    0.11
147    0.09
19     0.15
112    0.07
225    0.00
24     0.15
208    0.11
10     0.00
```

```
95     0.56
144    0.22
220    0.85
25     0.04
86     0.01
183    0.22
84     0.17
18     0.12
146    0.07
66     0.01
45     0.06
139    0.41
125    0.01
170    0.25
164    0.50
16     2.15
156    2.96
177    0.00
161    0.09
203    0.30
124    0.28
60     0.05
199    0.01
229    0.00
82     0.22
Name: World Population Percentage, dtype: float64
```

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

# Load the dataset
file_path = "world_population.csv"  # Update this if needed
df = pd.read_csv(file_path)

# Create a binary target variable based on the median population in
2022
median_population = df["2022 Population"].median()
df["High Population"] = (df["2022 Population"] >
median_population).astype(int)

# Select features for the model
features = ["2020 Population", "2015 Population", "2010 Population",
"2000 Population", "Density (per km²)", "Growth Rate"]
X = df[features]  # Use multiple features instead of just "2022
Population"
y = df["High Population"]
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)  # Now correctly 2D
X_test_scaled = scaler.transform(X_test)

# Train a logistic regression model
model = LogisticRegression()
model.fit(X_train_scaled, y_train)

# Make predictions
y_pred = model.predict(X_test_scaled)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy*100:.2f}%")
print("Classification Report:\n", report)
```

```
Accuracy: 82.98%
Classification Report:
              precision    recall  f1-score   support

           0       0.69      1.00      0.82        18
           1       1.00      0.72      0.84        29

    accuracy                           0.83        47
   macro avg       0.85      0.86      0.83        47
weighted avg       0.88      0.83      0.83        47
```

```
# Decision Tree Regression for Population Prediction
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score

# Select meaningful features
features = ["2020 Population", "2015 Population", "2010 Population",
"2000 Population"]
target = "2022 Population"

# Define X and y
X = df[features]
y = df[target]

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
```

```
test_size=0.2, random_state=42)

# Train Decision Tree Regressor
model = DecisionTreeRegressor()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Calculate evaluation metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)  # This acts like "accuracy" for
regression

print(f"Decision Tree MAE: {mae:.2f}")
print(f"Decision Tree MSE: {mse:.2f}")
print(f"Decision Tree R² Score (Accuracy): {r2:.2%}")  # Convert to
percentage


Decision Tree MAE: 3869298.74
Decision Tree MSE: 128808475931880.23
Decision Tree R² Score (Accuracy): 96.06%

# Random Forest Classifier for Population Category Prediction
# Creating population categories (Low, Medium, High)
# Use '2022 Population' instead of 'Population'
import pandas as pd # This import is likely already present in your
notebook, but it's included here for completeness
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier # Import the
RandomForestClassifier
from sklearn.metrics import accuracy_score
df['Population_Category'] = pd.qcut(df['2022 Population'], q=3,
labels=[0, 1, 2])  # 3 categories

X = df[['2022 Population']]  # If 'Year' column exists, otherwise
replace with a relevant feature column
y = df['Population_Category']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

rf_model = RandomForestClassifier(n_estimators=100, random_state=42) #
Now you can use RandomForestClassifier
rf_model.fit(X_train, y_train)

rf_predictions = rf_model.predict(X_test)
accuracy = accuracy_score(y_test, rf_predictions)
print(f"Random Forest Classifier Accuracy: {accuracy*100:.2f}")
```

```
Random Forest Classifier Accuracy: 100.00
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error,
mean_squared_error

# Load dataset
file_path = "world_population.csv"  # Update with correct path
df = pd.read_csv(file_path)

# Define features and target
features = ["2020 Population", "2015 Population", "2010 Population",
"2000 Population"]
target = "2022 Population"

X = df[features]
y = df[target]

# Handle non-numeric values
X = X.apply(pd.to_numeric, errors='coerce')  # Convert non-numeric to
NaN
y = pd.to_numeric(y, errors='coerce')

# Handle missing values
X.fillna(X.mean(), inplace=True)  # Replace NaNs with column means
y.fillna(y.mean(), inplace=True)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Standardize features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train Linear Regression model
model = LinearRegression()
model.fit(X_train_scaled, y_train)

# Make predictions
y_pred = model.predict(X_test_scaled)

# Evaluate model
r2 = r2_score(y_test, y_pred)  # Accuracy for regression
mae = mean_absolute_error(y_test, y_pred)  # Average error
mse = mean_squared_error(y_test, y_pred)  # Squared error
```

```python
print(f"Linear Regression R² Score (Accuracy): {r2:.2%}")  # Convert to percentage
print(f"Mean Absolute Error: {mae:.2f}")
print(f"Mean Squared Error: {mse:.2f}")
```

```
Linear Regression R² Score (Accuracy): 99.99%
Mean Absolute Error: 221989.47
Mean Squared Error: 340522093422.65
```