World_Population_UM

December 9, 2024

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
import plotly.graph_objects as go
from plotly.offline import iplot, plot
from plotly.subplots import make_subplots
import plotly.subplots as sp
import warnings
warnings.filterwarnings('ignore')
```

[ ]:
```
colors = ["#b1e7cd","#854442","#000000","#fff4e6","#3c2f2f",
           "#be9b7b ","#512E5F","#45B39D","#AAB7B8 ","#20B2AA",
           "#FF69B4","#00CED1","#FF7F50","#7FFF00","#DA70D6"]

color_2 = px.colors.sequential.RdBu
```

0.1 Importing Data From Drive

[ ]:
```
!pip install gdown
```

```
Requirement already satisfied: gdown in /usr/local/lib/python3.10/dist-packages
(5.2.0)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-
packages (from gdown) (4.12.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from gdown) (3.16.1)
Requirement already satisfied: requests[socks] in
/usr/local/lib/python3.10/dist-packages (from gdown) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from gdown) (4.66.6)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-
packages (from beautifulsoup4->gdown) (2.6)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (3.4.0)
```

```
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests[socks]->gdown) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown)
(2024.8.30)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in
/usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (1.7.1)
```

0.1.1 Gdown

[ ]:
```
import gdown

file_id = "1wvbMZDv3yrxpKS2y2gWIPsyqgTiw_fVA"
url = f"https://drive.google.com/uc?id={file_id}"
output = "world_pop.ext"
gdown.download(url, output, quiet=False)
```

```
Downloading…
From: https://drive.google.com/uc?id=1wvbMZDv3yrxpKS2y2gWIPsyqgTiw_fVA
To: /content/world_pop.ext
100%|      | 29.2k/29.2k [00:00<00:00, 27.5MB/s]
```

[ ]:
```
'world_pop.ext'
```

[ ]:
```
import pandas as pd
data = pd.read_csv("world_pop.ext")
```

[ ]:
```
data.head()
```

[ ]:
```
   Rank CCA3 Country/Territory           Capital Continent  2022 Population  \
0    36  AFG       Afghanistan             Kabul      Asia         41128771
1   138  ALB           Albania            Tirana    Europe          2842321
2    34  DZA           Algeria           Algiers    Africa         44903225
3   213  ASM    American Samoa         Pago Pago   Oceania            44273
4   203  AND           Andorra  Andorra la Vella    Europe            79824

   2020 Population  2015 Population  2010 Population  2000 Population  \
0         38972230         33753499         28189672         19542982
1          2866849          2882481          2913399          3182021
2         43451666         39543154         35856344         30774621
3            46189            51368            54849            58230
4            77700            71746            71519            66097

   1990 Population  1980 Population  1970 Population  Area (km²)  \
0         10694796         12486631         10752971      652230
1          3295066          2941651          2324731       28748
```

```
2         25518074         18739378         13795915     2381741
3            47818            32886            27075         199
4            53569            35611            19860         468

   Density (per km²)  Growth Rate  World Population Percentage
0            63.0587       1.0257                         0.52
1            98.8702       0.9957                         0.04
2            18.8531       1.0164                         0.56
3           222.4774       0.9831                         0.00
4           170.5641       1.0100                         0.00
```

```
[ ]: data.shape
```

```
[ ]: (234, 17)
```

0.1.2 Direct link Refrence

```
[ ]: file_id = "1wvbMZDv3yrxpKS2y2gWIPsyqgTiw_fVA"
     url = f"https://drive.google.com/uc?id={file_id}"
     df = pd.read_csv(url)
     df.head()
```

```
[ ]:    Rank CCA3 Country/Territory           Capital Continent  2022 Population  \
     0    36  AFG       Afghanistan             Kabul      Asia         41128771
     1   138  ALB           Albania            Tirana    Europe          2842321
     2    34  DZA           Algeria           Algiers    Africa         44903225
     3   213  ASM    American Samoa         Pago Pago   Oceania            44273
     4   203  AND           Andorra  Andorra la Vella    Europe            79824

        2020 Population  2015 Population  2010 Population  2000 Population  \
     0         38972230         33753499         28189672         19542982
     1          2866849          2882481          2913399          3182021
     2         43451666         39543154         35856344         30774621
     3            46189            51368            54849            58230
     4            77700            71746            71519            66097

        1990 Population  1980 Population  1970 Population  Area (km²)  \
     0         10694796         12486631         10752971      652230
     1          3295066          2941651          2324731       28748
     2         25518074         18739378         13795915     2381741
     3            47818            32886            27075         199
     4            53569            35611            19860         468

        Density (per km²)  Growth Rate  World Population Percentage
     0            63.0587       1.0257                         0.52
     1            98.8702       0.9957                         0.04
     2            18.8531       1.0164                         0.56
```

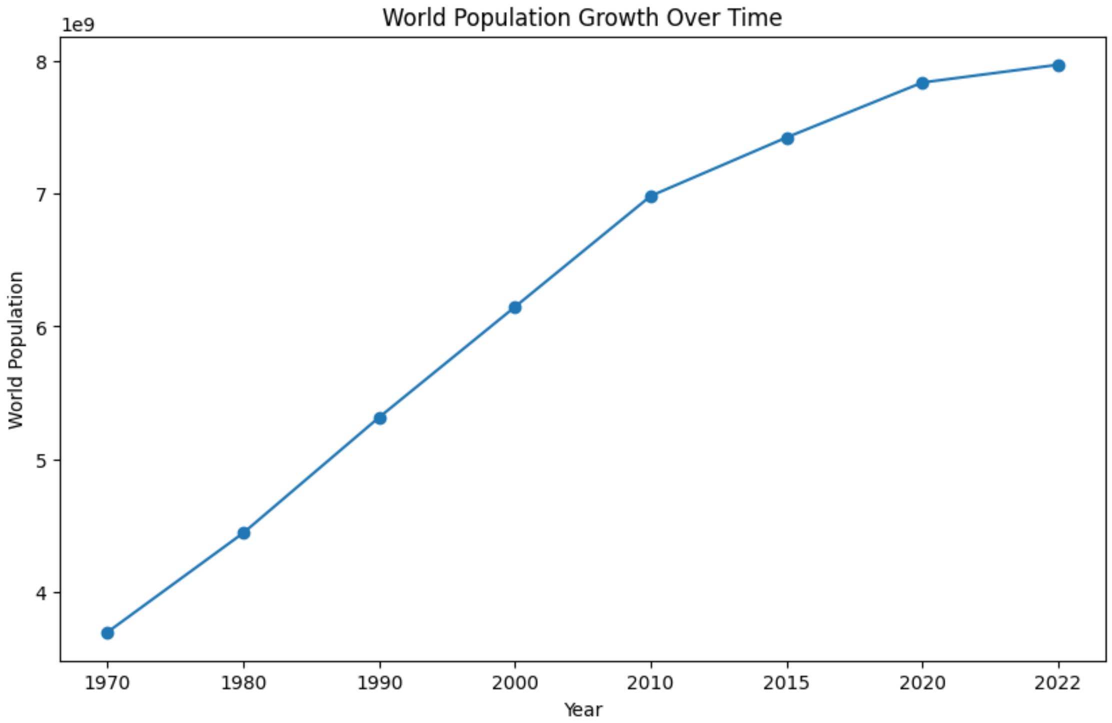
```
3            222.4774       0.9831                         0.00
4            170.5641       1.0100                         0.00
```

```
[ ]: from google.colab import sheets
     sheet = sheets.InteractiveSheet(df=df)
```

```
[ ]: df
```

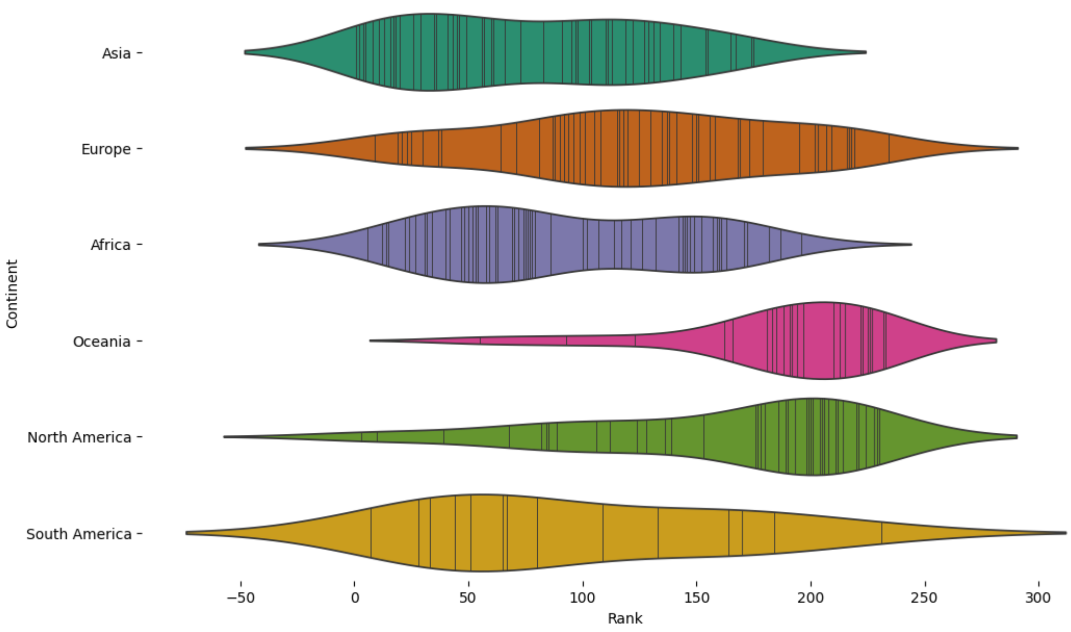
```
[ ]:      Rank CCA3  Country/Territory           Capital Continent  \
     0      36  AFG        Afghanistan             Kabul      Asia
     1     138  ALB            Albania            Tirana    Europe
     2      34  DZA            Algeria           Algiers    Africa
     3     213  ASM     American Samoa         Pago Pago   Oceania
     4     203  AND            Andorra  Andorra la Vella    Europe
     ..    …  …                …               …       …
     229   226  WLF  Wallis and Futuna          Mata-Utu   Oceania
     230   172  ESH     Western Sahara          El Aaiún    Africa
     231    46  YEM              Yemen             Sanaa      Asia
     232    63  ZMB             Zambia            Lusaka    Africa
     233    74  ZWE           Zimbabwe            Harare    Africa

          2022 Population  2020 Population  2015 Population  2010 Population  \
     0           41128771         38972230         33753499         28189672
     1            2842321          2866849          2882481          2913399
     2           44903225         43451666         39543154         35856344
     3              44273            46189            51368            54849
     4              79824            77700            71746            71519
     ..               …              …              …              …
     229            11572            11655            12182            13142
     230           575986           556048           491824           413296
     231         33696614         32284046         28516545         24743946
     232         20017675         18927715         16248230         13792086
     233         16320537         15669666         14154937         12839771

          2000 Population  1990 Population  1980 Population  1970 Population  \
     0           19542982         10694796         12486631         10752971
     1            3182021          3295066          2941651          2324731
     2           30774621         25518074         18739378         13795915
     3              58230            47818            32886            27075
     4              66097            53569            35611            19860
     ..               …              …              …              …
     229            14723            13454            11315             9377
     230           270375           178529           116775            76371
     231         18628700         13375121          9204938          6843607
     232          9891136          7686401          5720438          4281671
     233         11834676         10113893          7049926          5202918
          Area (km²)  Density (per km²)  Growth Rate  World Population Percentage
     0        652230            63.0587       1.0257                         0.52
     1         28748            98.8702       0.9957                         0.04
     2       2381741            18.8531       1.0164                         0.56
     3           199           222.4774       0.9831                         0.00
     4           468           170.5641       1.0100                         0.00
     ..          …                  …            …                          …
     229         142            81.4930       0.9953                         0.00
     230      266000             2.1654       1.0184                         0.01
     231      527968            63.8232       1.0217                         0.42
     232      752612            26.5976       1.0280                         0.25
     233      390757            41.7665       1.0204                         0.20

     [234 rows x 17 columns]
```

```
[ ]: # @title World Population Growth Over Time

     import matplotlib.pyplot as plt
     import pandas as pd

     # Assuming your DataFrame is named 'df'

     years = ['1970', '1980', '1990', '2000', '2010', '2015', '2020', '2022']
     populations = [df[f'{year} Population'].sum() for year in years]

     plt.figure(figsize=(10, 6))
     plt.plot(years, populations, marker='o')
     plt.xlabel('Year')
     plt.ylabel('World Population')
     _ = plt.title('World Population Growth Over Time')
```



[ ]:

```
# @title Continent vs Rank

from matplotlib import pyplot as plt
import seaborn as sns
figsize = (12, 1.2 * len(df['Continent'].unique()))
plt.figure(figsize=figsize)
sns.violinplot(df, x='Rank', y='Continent', inner='stick', palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```



```
[ ]: print(f'the number of rows is : {df.shape[0]} \nthe number of columns is : {df.
       ↪shape[1]} '.upper() )
```

```
THE NUMBER OF ROWS IS : 234
THE NUMBER OF COLUMNS IS : 17
```

1 Data Summarization

```
[ ]: 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
```

```
[ ]: df.duplicated().any()
```

```
[ ]: False
```

```
[ ]: 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
```

```
[ ]: # Creating a DataFrame to understand the Data

     pd.DataFrame({
         'counts': df.shape[0],
         'nulls': df.isnull().sum(),
         'nulls%': df.isnull().mean() * 100,
         'cardinality': df.nunique(),
         'dtypes': df.dtypes
     })
```

```
[ ]:                              counts  nulls  nulls%  cardinality   dtypes
     Rank                            234      0     0.0          234    int64
     CCA3                            234      0     0.0          234   object
```

```
Country/Territory              234     0     0.0          234   object
Capital                        234     0     0.0          234   object
Continent                      234     0     0.0            6   object
2022 Population                234     0     0.0          234    int64
2020 Population                234     0     0.0          234    int64
2015 Population                234     0     0.0          234    int64
2010 Population                234     0     0.0          234    int64
2000 Population                234     0     0.0          234    int64
1990 Population                234     0     0.0          234    int64
1980 Population                234     0     0.0          234    int64
1970 Population                234     0     0.0          234    int64
Area (km²)                     234     0     0.0          233    int64
Density (per km²)              234     0     0.0          234  float64
Growth Rate                    234     0     0.0          180  float64
World Population Percentage    234     0     0.0           70  float64
```

[ ]:

```
df.describe(include='int64')
```

[ ]:

```
             Rank  2022 Population  2020 Population  2015 Population  \
count  234.000000     2.340000e+02     2.340000e+02     2.340000e+02
mean   117.500000     3.407441e+07     3.350107e+07     3.172996e+07
std     67.694165     1.367664e+08     1.355899e+08     1.304050e+08
min      1.000000     5.100000e+02     5.200000e+02     5.640000e+02
25%     59.250000     4.197385e+05     4.152845e+05     4.046760e+05
50%    117.500000     5.559944e+06     5.493074e+06     5.307400e+06
75%    175.750000     2.247650e+07     2.144798e+07     1.973085e+07
max    234.000000     1.425887e+09     1.424930e+09     1.393715e+09

       2010 Population  2000 Population  1990 Population  1980 Population  \
count     2.340000e+02     2.340000e+02     2.340000e+02     2.340000e+02
mean      2.984524e+07     2.626947e+07     2.271022e+07     1.898462e+07
std       1.242185e+08     1.116982e+08     9.783217e+07     8.178519e+07
min       5.960000e+02     6.510000e+02     7.000000e+02     7.330000e+02
25%       3.931490e+05     3.272420e+05     2.641158e+05     2.296142e+05
50%       4.942770e+06     4.292907e+06     3.825410e+06     3.141146e+06
75%       1.915957e+07     1.576230e+07     1.186923e+07     9.826054e+06
max       1.348191e+09     1.264099e+09     1.153704e+09     9.823725e+08

       1970 Population    Area (km²)
count     2.340000e+02  2.340000e+02
mean      1.578691e+07  5.814494e+05
std       6.779509e+07  1.761841e+06
min       7.520000e+02  1.000000e+00
25%       1.559970e+05  2.650000e+03
50%       2.604830e+06  8.119950e+04
75%       8.817329e+06  4.304258e+05
max       8.225344e+08  1.709824e+07
```

1.0.1 Column Insights:

1. Rank

- **`mean:`** 117.5, **`std:`** 67.69 → The average rank is in the middle of the range (117.5), with a wide spread, suggesting evenly distributed rankings.
- **`min:`** 1, **`max:`** 234 → The ranks span from 1 to 234, representing all countries in the dataset.

2. Populations (`2022`, `2020`, `2015`, etc.)

- **`mean:`** Around 34 million in 2022, gradually decreasing in earlier years.
  This suggests a global increase in population over time.
- **`std:`** High standard deviations (e.g., ~136.77 million in 2022) indicate significant variation in country populations, with a few highly populous countries (like India and China) skewing the distribution.
- **`min:`** 510 (2022 Population) → Indicates that some countries have extremely small populations (likely small islands or territories).
- **`max:`** ~1.425 billion (2022 Population) → The most populous country (likely China or India) is over 1,000 times larger than the smallest.

3. `Area (km²)`

- **`mean:`** ~581,449 km^2 → The average country size is about 581,000 square kilometers.
- **`std:`** ~1.76 million km^2 → Large variation in country sizes, with a few massive countries skewing the average.
- **`min:`** 1 km^2 → Likely represents very small territories or city-states like Monaco.
- **`max:`** ~17.1 million km^2 → Represents the largest country by area (likely Russia).

1.0.2 Percentiles (25%, 50%, 75%)

Percentiles help understand the distribution of values:

Population Columns:

- **25% (First Quartile):** ~420,000 in 2022 → 25% of countries have populations below this number, suggesting many small-population countries.
- **50% (Median):** ~5.56 million in 2022 → The middle country has a population of ~5.56 million.
- **75% (Third Quartile):** ~22.47 million in 2022 → 75% of countries have populations below this value.

Area (km^2):

- **25% (First Quartile):** 2,650 km^2 → Small countries like island nations.
- **50% (Median):** ~81,200 km^2 → Half of the countries are smaller than this size.
- **75% (Third Quartile):** ~430,425 km^2 → Most countries are significantly smaller than the largest.

1.0.3 Key Observations:

1. **Population Trends:**
   - Populations show a steady increase over time, with current (2022) numbers being the largest.
   - A few countries dominate population size, creating a skewed distribution (high `std`).
2. **Area (km²):**
   - There's an extreme range in country sizes, with most countries being much smaller than the largest ones.
   - Median size is far below the mean, highlighting the influence of a few massive countries.
3. **Disparities in Data:**
   - Both population and area exhibit high variability, driven by the presence of both small (e.g., city-states, islands) and large countries (e.g., Russia, China).
4. **Small Populations and Areas:**
   - Many countries or territories have small populations and areas, contributing to the lower percentile values.
5. **Country Representation:**
   - The dataset includes a diverse set of countries, from very small to very large in both population and size, making it representative of global disparities.

1.0.4 Suggestions for Analysis:

- Explore relationships between population and area to identify trends (e.g., population density).
- Analyze population growth trends over decades to identify regions or countries with rapid growth.
- Investigate outliers (countries with extremely small or large populations/areas) for special cases like city-states or sparsely populated countries.

Let me know if you'd like help exploring or visualizing these insights further!

[ ]:

```
df.describe(include='object')
```

[ ]:

```
        CCA3 Country/Territory Capital Continent
count    234               234     234       234
unique   234               234     234         6
top      AFG       Afghanistan   Kabul    Africa
freq       1                 1       1        57
```

1.0.5 Metrics Explained:

1. **`count`:**
   - Number of non-null entries in each column.
   - In your data, all columns have a count of 234, meaning there are no missing values in these columns.
2. **`unique`:**
   - Number of unique values in each column.

    - High cardinality (equal to the count) in `CCA3`, `Country/Territory`, and `Capital`, indicating all values in these columns are unique.
    - Low cardinality (6 unique values) in `Continent`, suggesting repetition of continent names across countries.
3. **`top`:**
    - The most frequently occurring value in each column (the mode).
    - `top` provides a sample of the most common entry but does not indicate how representative it is without considering `freq`.
4. **`freq`:**
    - Frequency of the `top` value in the column.
    - Shows how many times the most common value appears.

1.0.6 Column Insights:

1. `CCA3`:

- **count = 234, unique = 234**: Every country has a unique 3-letter code. No duplicates or repetitions.
- **top = "AFG", freq = 1**: The code for Afghanistan appears only once, indicating each country has a unique identifier.

2. `Country/Territory`:

- **count = 234, unique = 234**: Each row represents a unique country or territory. No duplicates.
- **top = "Afghanistan", freq = 1**: Afghanistan is listed once, like every other country.

3. `Capital`:

- **count = 234, unique = 234**: Each country has a unique capital city name (no repeats).
- **top = "Kabul", freq = 1**: Kabul appears once, which is expected given the unique capitals.

4. `Continent`:

- **count = 234, unique = 6**: Only 6 unique continent names (e.g., Africa, Asia, etc.) are represented across the 234 countries.
- **top = "Africa", freq = 57**: Africa is the most common continent, with 57 countries from the dataset belonging to it.

1.0.7 Summary of the Data:

1. **`CCA3`, `Country/Territory`, and `Capital`**:
    - These columns represent unique identifiers or properties of each country.
    - No duplicates, meaning each country, its code, and its capital are distinct.
2. **`Continent`**:
    - Represents a grouping or categorization.

   - Some continents have a higher representation (e.g., Africa with 57 countries), while others have fewer.
3. **Key Takeaways:**
   - The dataset appears clean with no missing values in these object columns.
   - `CCA3` and `Country/Territory` are likely identifiers, and `Continent` is a low-cardinality column useful for grouping or aggregation.

If you'd like, we can explore how these insights might help in analysis or modeling!

[ ]:
```
num_cols = list(df.select_dtypes(include=np.number).columns)
cat_cols = list(df.select_dtypes(include='object').columns)
```

2 Data Visualization EDA

[ ]:
```
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')
```

[ ]:
```
countries_by_continent = df['Continent'].value_counts().reset_index()
```

[ ]:
```
# Create the bar chart
fig = px.bar(
countries_by_continent,
x='Continent',
y='count',
color='Continent',
text='count',
title='Number of Countries by Continent',
color_discrete_sequence=[colors]*len(countries_by_continent),
labels={'count': 'Number of Countries', 'Continent': 'Continent'},
template='plotly_dark'
)

# Customize the layout
fig.update_layout(
xaxis_title='Continents',
yaxis_title='Number of Countries',
plot_bgcolor='rgba(0,0,0,0)', # Set the background color to transparent
font_family='Arial', # Set font family
title_font_size=20 # Set title font size
)
```

```

# Show the plot
fig.show()
```

[ ]:
```
# Melt the DataFrame to have a long format
df_melted = df.melt(id_vars=['Continent'],

            value_vars = ['2022 Population', '2020 Population', '2015␣
 ↪Population', '2010 Population',
                        '2000 Population', '1990 Population', '1980␣
 ↪Population', '1970 Population'],
            var_name = 'Year', value_name = 'Population'
            )
```

[ ]:
```
df_melted
```

[ ]:
```
     Continent             Year  Population
0         Asia  2022 Population    41128771
1       Europe  2022 Population     2842321
2       Africa  2022 Population    44903225
3      Oceania  2022 Population       44273
4       Europe  2022 Population       79824
…          …                …          …
1867   Oceania  1970 Population        9377
1868    Africa  1970 Population       76371
1869      Asia  1970 Population     6843607
1870    Africa  1970 Population     4281671
1871    Africa  1970 Population     5202918

[1872 rows x 3 columns]
```

[ ]:
```
# Convert 'Year' to a more suitable format
df_melted['Year'] = df_melted['Year'].str.split().str[0].astype(int)
```

[ ]:
```
# Aggregate population by continent and year
population_by_continent = df_melted.groupby(['Continent', 'Year']).sum().
 ↪reset_index()
```

[ ]:
```

```

[ ]:
```
fig = px.line(population_by_continent, x='Year', y='Population',␣
 ↪color='Continent',

title='Population Trends by Continent Over Time',
labels={'Population': 'Population', 'Year': 'Year'},
color_discrete_sequence=colors)

```

```
fig.update_layout(

template='plotly_dark',
xaxis_title='Year',
yaxis_title='Population',
font_family='Arial',
title_font_size=20,
)

fig.update_traces(line=dict(width=3))

fig.show()
```

[ ]:

```
land_by_country = df.groupby('Country/Territory')['Area (km²)'].sum().
  ↪sort_values(ascending=False)
most_land = land_by_country.head(5)
least_land = land_by_country.tail(5)
```

[ ]:

```
most_land
```

[ ]:

```
Country/Territory
Russia           17098242
Canada            9984670
China             9706961
United States     9372610
Brazil            8515767
Name: Area (km²), dtype: int64
```

[ ]:

```
# Create subplots
fig = sp.make_subplots(rows=1, cols=2, subplot_titles=("Countries with Most␣
  ↪Land", "Countries with Least Land"))

# Plot countries with the most land
fig.add_trace(go.Bar(x=most_land.index, y=most_land.values, name='Most Land',
marker_color=colors[0]), row=1, col=1, )

# Plot countries with the least land
fig.add_trace(go.Bar(x=least_land.index, y=least_land.values, name='Least Land',
marker_color=colors[1]), row=1, col=2)

# fig.update_traces(
#     text=most_land.values,
#     texttemplate='%{text:.2s}',
#     textposition='outside',
#     hovertemplate='<b>Country: %{x}</b><br>Area (km²): %{y}<extra></extra>'
# )
```

```
fig.update_layout(
title_text="Geographical Distribution of Land Area by Country",
showlegend=False,
template='plotly_dark'
)

fig.update_yaxes(title_text="Area (km2)", row=1, col=1)
fig.update_yaxes(title_text="Area (km2)", row=1, col=2)

fig.show()
```

[ ]:

```

```

[ ]:

```
# GET THE SUM OF RANKS TO GET RELATION BETWEEN RANKS AND CONTINENT

con = df.groupby('Continent').sum()

fig =  ( px.bar(con,x=con.index,y='Rank'
                ,template='plotly_dark'
                ,color_discrete_sequence=['#20B2AA']
                ,title = '<b>Sum Of Rank Of Each Continent</b>')
       )

fig.update_traces(
    text=con['Rank'],
    texttemplate='%{text:.2s}',
    textposition='outside',
    hovertemplate='<b>Continent: %{x}</b><br>Population: %{y}<extra></extra>'
)

fig.update_layout(bargap=0.7)
iplot(fig)
```

[ ]:

```
col = df.iloc[:,5:13].columns # EXTRACTING YEARS COLOUMN

for year in col :

   fig = px.bar(con,x=con.index,y=year
                ,template='plotly_dark'
                ,title=f'Total Population For {year}')
   fig.update_traces(
    text=con[year],
    texttemplate='%{text:.2s}',
    textposition='outside',
    hovertemplate='<b>Continent: %{x}</b><br>Population: %{y}<extra></extra>'
   )
   fig.update_layout(bargap=0.5)
```

```
    iplot(fig)
```

Asia has highest population from 1972 to 2022 GROWTH RATE FOR CONTINENTS AND COUNTRIES

```
[ ]: con.columns
```

```
[ ]: Index(['Rank', 'CCA3', 'Country/Territory', 'Capital', '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')
```

```
[ ]: iplot(

     px.pie(con,names=con.index,values='Growth Rate',hole=0.3
            ,title='Growth Rate for each continent',template='plotly_dark')
     )
```

Africa has higher growth rate

```
[ ]: top_5_country = df.nlargest(5,'Growth Rate') #
     top = top_5_country['Country/Territory']

     iplot(
         px.pie(top_5_country, names = top, values = 'Growth Rate',
                title = 'Growth rate of top 5 countries', hole = 0.3, template =␣
      ↪'plotly_dark', color_discrete_sequence = colors)
     )
```

Although the percentages are very close, Moldova has the highest percentage.

DISTRIBUTION OF AREA AND DENSITY ACROSS CONTINENTS AND COUNTRIES

```
[ ]: iplot(
     px.bar(con,x=con.index,y='Area (km²)',template='plotly_dark'
            ,color_discrete_sequence=[colors[8]]
            ,title='Distribution of Area (km²) Across Continents',).
      ↪update_traces(texttemplate='%{y}',textposition='outside')
     )
```

```
[ ]: iplot(
     px.bar(con,x=con.index,y='Density (per km²)',template='plotly_dark'
            ,color_discrete_sequence=[colors[8]]
            ,title='Distribution of Density (per km²)) Across Continents',).
      ↪update_traces(texttemplate='%{y}',textposition='outside')
     )
```

Asia has the largest area and density per km2

[ ]:
```
top10_den = df.nlargest(5,'Density (per km²)')
top_den=top10_den['Country/Territory']

iplot(
   px.pie(top10_den,names=top_den,values='Density (per km²)'
         ,template='plotly_dark',color_discrete_sequence=colors,hole=0.7
          ,title= 'Top 5 Countries with Highest Population Density '
         ).update_traces(textposition='inside',textinfo='percent+label')
)
```

[ ]:
```
fig= px.bar(top10_den,y=top_den,x='Density (per km²)'
        ,template='plotly_dark',color_discrete_sequence=[colors[8]]
            ,title= 'Top 5 Countries with Highest Population Density')
fig.update_traces(texttemplate='%{x}',textposition='outside')
fig.update_layout(bargap=0.7)

iplot(fig)
```

MACAU Has the Highest Population Density

POPULATION GROWTH OVER TIME

[ ]:
```
iplot(

   px.bar(con,x=con.index,y='World Population Percentage',template='plotly_dark'
       ,color_discrete_sequence=[colors[1]],
         title='World Population Percentage for each Continents ').
 ↪update_traces(texttemplate='%{y}',textposition='outside')

)
```

[ ]:
```
top_countries_per_continent = pd.DataFrame()

for continent, group in df.groupby('Continent'):
   top_countries = group.nlargest(5,'2022 Population')
   top_countries_per_continent = pd.
 ↪concat([top_countries_per_continent,top_countries])

fig = px.bar(top_countries_per_continent,x='Country/Territory',y='World␣
 ↪Population Percentage',
            title='Top 5 Countries by World Population Percentage in 2022',
            labels={'World Population Percentage': 'Percentage'},
            facet_col='Continent',
            facet_col_wrap=2,
            color='Continent',
            template="plotly_dark")
```

```
iplot(fig)
```

[ ]:
```
top_countries_per_continent
```

[ ]:
```
     Rank CCA3 Country/Territory           Capital      Continent  \
149     6  NGA           Nigeria             Abuja         Africa
63     12  ETH          Ethiopia       Addis Ababa         Africa
57     14  EGY             Egypt             Cairo         Africa
55     15  COD          DR Congo          Kinshasa         Africa
205    22  TZA          Tanzania            Dodoma         Africa
41      1  CHN             China           Beijing           Asia
92      2  IND             India         New Delhi           Asia
93      4  IDN         Indonesia           Jakarta           Asia
156     5  PAK          Pakistan         Islamabad           Asia
16      8  BGD        Bangladesh             Dhaka           Asia
171     9  RUS            Russia            Moscow         Europe
74     19  DEU           Germany            Berlin         Europe
220    21  GBR    United Kingdom            London         Europe
68     23  FRA            France             Paris         Europe
99     25  ITA             Italy              Rome         Europe
221     3  USA     United States  Washington, D.C.  North America
131    10  MEX            Mexico       Mexico City  North America
35     39  CAN            Canada            Ottawa  North America
82     68  GTM         Guatemala    Guatemala City  North America
87     82  HTI             Haiti    Port-au-Prince  North America
11     55  AUS         Australia          Canberra        Oceania
160    93  PNG  Papua New Guinea      Port Moresby        Oceania
146   123  NZL       New Zealand        Wellington        Oceania
66    162  FJI              Fiji              Suva        Oceania
191   166  SLB   Solomon Islands           Honiara        Oceania
27      7  BRA            Brazil          Brasilia  South America
42     28  COL          Colombia            Bogota  South America
8      33  ARG         Argentina      Buenos Aires  South America
162    44  PER              Peru              Lima  South America
227    51  VEN         Venezuela           Caracas  South America

     2022 Population  2020 Population  2015 Population  2010 Population  \
149        218541212        208327405        183995785        160952853
63         123379924        117190911        102471895         89237791
57         110990103        107465134         97723799         87252413
55          99010212         92853164         78656904         66391257
205         65497748         61704518         52542823         45110527
41        1425887337       1424929781       1393715448       1348191368
92        1417173173       1396387127       1322866505       1240613620
93         275501339        271857970        259091970        244016173
156        235824862        227196741        210969298        194454498
```

```
16         171186372        167420951        157830000        148391139
171        144713314        145617329        144668389        143242599
74          83369843         83328988         82073226         81325090
220         67508936         67059474         65224364         62760039
68          64626628         64480053         63809769         62444567
99          59037474         59500579         60232906         59822450
221        338289857        335942003        324607776        311182845
131        127504125        125998302        120149897        112532401
35          38454327         37888705         35732126         33963412
82          17843908         17362718         16001107         14543121
87          11584996         11306801         10563757          9842880
11          26177413         25670051         23820236         22019168
160         10142619          9749640          8682174          7583269
146          5185288          5061133          4590590          4346338
66            929766           920422           917200           905169
191           724273           691191           612660           540394
27         215313498        213196304        205188205        196353492
42          51874024         50930662         47119728         44816108
8           45510318         45036032         43257065         41100123
162         34049588         33304756         30711863         29229572
227         28301696         28490453         30529716         28715022

     2000 Population  1990 Population  1980 Population  1970 Population  \
149        122851984         95214257         72951439         55569264
63          67031867         47878073         34945469         28308246
57          71371371         57214630         43748556         34781986
55          48616317         35987541         26708686         20151733
205         34463704         26206012         19297659         13618192
41        1264099069       1153704252        982372466        822534450
92        1059633675        870452165        696828385        557501301
93         214072421        182159874        148177096        115228394
156        154369924        115414069         80624057         59290872
16         129193327        107147651         83929765         67541860
171        146844839        148005704        138257420        130093010
74          81551677         79370196         77786703         78294583
220         58850043         57210442         56326328         55650166
68          58665453         56412897         53713830         50523586
99          56966397         56756561         56329482         53324036
221        282398554        248083732        223140018        200328340
131         97873442         81720428         67705186         50289306
35          30683313         27657204         24511510         21434577
82          11735894          9084780          6987767          5453208
87           8360225          6925331          5646676          4680812
11          19017963         17048003         14706322         12595034
160          5508297          3864972          3104788          2489059
146          3855266          3397389          3147168          2824061
66            832509           780430           644582           527634
```

```
191         429978         324171         233668         172833
27       175873720      150706446      122288383       96369875
42        39215135       32601393       26176195       20905254
8         37070774       32637657       28024803       23842803
162       26654439       22109099       17492406       13562371
227       24427729       19750579       15210443       11355475

     Area (km²)  Density (per km²)  Growth Rate  World Population Percentage
149      923768           236.5759       1.0241                         2.74
63      1104300           111.7268       1.0257                         1.55
57      1002450           110.7188       1.0158                         1.39
55      2344858            42.2244       1.0325                         1.24
205      945087            69.3034       1.0300                         0.82
41      9706961           146.8933       1.0000                        17.88
92      3287590           431.0675       1.0068                        17.77
93      1904569           144.6529       1.0064                         3.45
156      881912           267.4018       1.0191                         2.96
16       147570          1160.0350       1.0108                         2.15
171    17098242             8.4636       0.9973                         1.81
74       357114           233.4544       0.9995                         1.05
220      242900           277.9289       1.0034                         0.85
68       551695           117.1419       1.0015                         0.81
99       301336           195.9191       0.9966                         0.74
221     9372610            36.0935       1.0038                         4.24
131     1964375            64.9082       1.0063                         1.60
35      9984670             3.8513       1.0078                         0.48
82       108889           163.8725       1.0134                         0.22
87        27750           417.4773       1.0120                         0.15
11      7692024             3.4032       1.0099                         0.33
160      462840            21.9139       1.0194                         0.13
146      270467            19.1716       1.0108                         0.07
66        18272            50.8847       1.0056                         0.01
191       28896            25.0648       1.0232                         0.01
27      8515767            25.2841       1.0046                         2.70
42      1141748            45.4339       1.0069                         0.65
8       2780400            16.3683       1.0052                         0.57
162     1285216            26.4933       1.0099                         0.43
227      916445            30.8820       1.0036                         0.35
```

Asia has the Highest percentage

POPULATION GROWTH OVER TIME

```
[ ]: df.iloc[:,5:13].columns
```

```
[ ]: Index(['2022 Population', '2020 Population', '2015 Population',
            '2010 Population', '2000 Population', '1990 Population',
            '1980 Population', '1970 Population'],
      dtype='object')
```

[ ]:

```
df.iloc[:,5:13].sum().sort_values()
```

[ ]:

```
1970 Population    3694136661
1980 Population    4442400371
1990 Population    5314191665
2000 Population    6147055703
2010 Population    6983784998
2015 Population    7424809761
2020 Population    7839250603
2022 Population    7973413042
dtype: int64
```

[ ]:

```
trd= df.iloc[:,5:13].sum().sort_values()

iplot(

px.line(trd,x=trd.index,y=trd.values,template='plotly_dark', markers=True,
       color_discrete_sequence=[colors[9]]
       ,title='total trend population for (1970 => 2020)').
 ↪update_traces(textposition='top center')
)
```

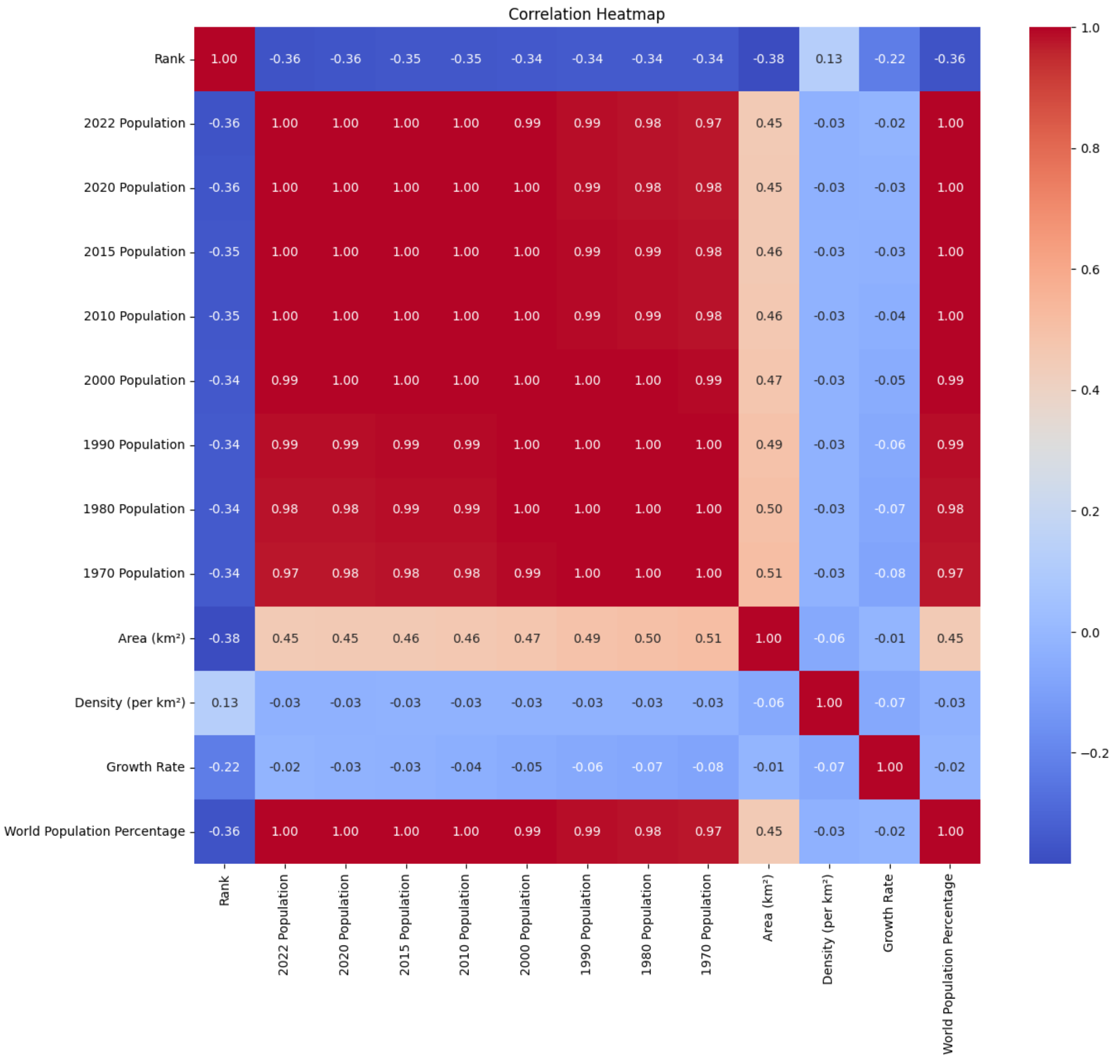
Conclusion

1. Africa has a higher growth rate than other continents.
2. Asia has the highest population from 1970 until 2022.
3. Asia has the largest area and density per km2
4. Asia has the Highest percentage of World Population
5. North America has the hightest ranks Between the Continents
6. Moldova has the highest percentage of Growth Rate between the countries
7. MACAU Has the Highest Population Density between the countries

3 Model Evaluation

[ ]:

```
numeric_df = df.select_dtypes(include=[np.number])
plt.figure(figsize=(14, 12))
sns.heatmap(numeric_df.corr(), annot=True, fmt='.2f', cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



```
[ ]: from sklearn.model_selection import train_test_split
```

```
[ ]: 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')
```

```
[ ]: print(len(df.columns))
```

```
17
```

[ ]:
```
df.iloc[:,5:13].columns
```

[ ]:
```
Index(['2022 Population', '2020 Population', '2015 Population',
       '2010 Population', '2000 Population', '1990 Population',
       '1980 Population', '1970 Population'],
      dtype='object')
```

[ ]:
```
df1 = df.copy()
df2 = df.copy()
df3 = df.copy()
```

[ ]:
```
# Prepare the data
X = df[['1970 Population', '1980 Population', '1990 Population', '2000␣
 ↪Population', '2010 Population', '2015 Population', '2020 Population']]
y = df['2022 Population']
```

[ ]:
```
x = df.iloc[:,6:13]
y = df.iloc[:,5]
```

[ ]:
```
# Split the data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
 ↪random_state=42)

print("x_train shape:", x_train.shape)
print("X_test shape:", x_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
x_train shape: (187, 7)
X_test shape: (47, 7)
y_train shape: (187,)
y_test shape: (47,)
```

[ ]:
```
from sklearn.metrics import *
```

[ ]:
```
def eval_model(model,mname):

    model.fit(x_train,y_train)
    y_pred = model.predict(x_test)

    # Error Evaluation
    mae =  mean_absolute_error(y_test,y_pred)
    mse =  mean_squared_error(y_test,y_pred)
    rmse = np.sqrt(mse)
    r2 =   r2_score(y_test,y_pred)
    print('MAE',mae)
    print('MSE',mse)
    print('RMSE',rmse)
    print('R2_Score',r2)

    # Train Acc
    train_acc = model.score(x_train,y_train)   # Train Acc
    test_acc = model.score(x_test,y_test)      # Test Acc
    # cm = confusion_matrix(y_test,ypred)
    # crep = classification_report(y_test,ypred) # for categorical data these␣
↪are used
    # print(cm)
    # print(crep)
    res_df = pd.DataFrame({
                            'Train_Acc':train_acc,
                            'Test_Acc':test_acc,
                            'MAE':mae,
                            'MSE':mse,
                            'RMSE':rmse,
                            'R2_Score':r2
                            },index = [mname])
    return res_df
```

3.1 Linear Regression

[ ]:
```
from sklearn.linear_model import LinearRegression

lr1 = LinearRegression()

res_df1 = eval_model(lr1,'Linear Regression')
res_df1
```

```
MAE 224905.0539170114
MSE 377461288652.7521
RMSE 614378.7827169425
R2_Score 0.9998845013551759
```

[ ]:
```
                   Train_Acc  Test_Acc            MAE           MSE  \
Linear Regression   0.999996  0.999885  224905.053917  3.774613e+11

                            RMSE  R2_Score
Linear Regression  614378.782717  0.999885
```

3.2 KNN Neighbours

[ ]:
```
from sklearn.neighbors import KNeighborsRegressor

knn = KNeighborsRegressor()
```

```
res_df2 = eval_model(knn,'KNN')
res_df2
```

```
MAE 4449574.5234042555
MSE 208261006459841.2
RMSE 14431251.03585414
R2_Score 0.9362746201030088
```

```
[ ]:      Train_Acc  Test_Acc           MAE           MSE          RMSE  R2_Score
     KNN     0.7572  0.936275  4.449575e+06  2.082610e+14  1.443125e+07  0.936275
```

3.3 Decision Tree

```
[ ]: from sklearn.tree import DecisionTreeRegressor

     dt = DecisionTreeRegressor()

     res_df3 = eval_model(dt,'Decision Tree')
     res_df3
```

```
MAE 4093555.063829787
MSE 151687721673780.7
RMSE 12316156.936064946
R2_Score 0.9535853693704553
```

```
[ ]:                Train_Acc  Test_Acc           MAE           MSE          RMSE  \
     Decision Tree        1.0  0.953585  4.093555e+06  1.516877e+14  1.231616e+07

                    R2_Score
     Decision Tree  0.953585
```

3.4 Random Forest

```
[ ]: from sklearn.ensemble import RandomForestRegressor

     rf = RandomForestRegressor()

     res_df4 = eval_model(rf,'Random Forest')
     res_df4
```

```
MAE 3079582.504255319
MSE 90751132805542.69
RMSE 9526338.898314646
R2_Score 0.9722312375589587
```

[ ]:

```
                Train_Acc  Test_Acc           MAE           MSE          RMSE  \
Random Forest    0.991636  0.972231  3.079583e+06  9.075113e+13  9.526339e+06

               R2_Score
Random Forest  0.972231
```

3.5 Result DataFrame

[ ]:

```
res_df = pd.concat([res_df1,res_df2,res_df3,res_df4]).
 ↪sort_values('R2_Score',ascending=False)
res_df
```

[ ]:

```
                   Train_Acc  Test_Acc           MAE           MSE  \
Linear Regression   0.999996  0.999885  2.249051e+05  3.774613e+11
Random Forest       0.991636  0.972231  3.079583e+06  9.075113e+13
Decision Tree       1.000000  0.953585  4.093555e+06  1.516877e+14
KNN                 0.757200  0.936275  4.449575e+06  2.082610e+14

                           RMSE  R2_Score
Linear Regression  6.143788e+05  0.999885
Random Forest      9.526339e+06  0.972231
Decision Tree      1.231616e+07  0.953585
KNN                1.443125e+07  0.936275
```

The result DataFrame shows that Linear Regression Model has best R2 score for our Regression problem which is about 99%

[ ]:

```

```