World Population Data Analysis Report

Only EDA

Author

Lahiru Sadakelum

July 3, 2024

Abstract

This report presents a detailed analysis of world population data since 1970. The primary objective is to understand world population trends and their implications. The analysis covers key metrics such as population growth rate, distribution, and geographic variations. We use data from reputable resources. We found various patterns through statistical and visualization methods. This report aims to provide some valuable information for researchers and the general public to better understand the world population

Introduction

Understanding global population dynamics is crucial for policymakers, researchers, and organizations to formulate informed strategies and policies. The world population has experienced significant growth and transformation, influenced by factors such as fertility rates, mortality rates, migration patterns, and socio-economic development. Analyzing these trends helps in anticipating future challenges and opportunities related to healthcare, education, employment, and environmental sustainability.

In this report, we utilize data from the United Nations World Population Prospects 2022 and other reputable sources to examine historical trends, regional disparities, and demographic projections. The analysis focuses on key metrics such as population size, growth rates, age structure, and geographical distribution, aiming to provide a comprehensive overview of global demographic patterns and their implications for the future.

Methodology

Data cleaning and EDA processes are below.

In [2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
df = pd.read_csv(r"C:\Users\lahir\Desktop\Python Jupiter\EDA\world_populatic
df
```

Out[3]:

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	2015 Population	Popu
0	36	AFG	Afghanistan	Kabul	Asia	41128771.0	38972230.0	33753499.0	28189
1	138	ALB	Albania	Tirana	Europe	2842321.0	2866849.0	2882481.0	2913
2	34	DZA	Algeria	Algiers	Africa	44903225.0	43451666.0	39543154.0	35856
3	213	ASM	American Samoa	Pago Pago	Oceania	44273.0	46189.0	51368.0	54
4	203	AND	Andorra	Andorra la Vella	Europe	79824.0	77700.0	71746.0	71
...	...	...	...	...	...	...	...	...	
229	226	WLF	Wallis and Futuna	Mata-Utu	Oceania	11572.0	11655.0	12182.0	13
230	172	ESH	Western Sahara	El Aaiún	Africa	575986.0	556048.0	491824.0	413
231	46	YEM	Yemen	Sanaa	Asia	33696614.0	32284046.0	28516545.0	24743
232	63	ZMB	Zambia	Lusaka	Africa	20017675.0	18927715.0	NaN	13792
233	74	ZWE	Zimbabwe	Harare	Africa	16320537.0	15669666.0	14154937.0	12839

234 rows × 17 columns

Check Data Frame

In [4]:
```
df.head()
```

Out[4]:

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	2015 Population	20 Populat
0	36	AFG	Afghanistan	Kabul	Asia	41128771.0	38972230.0	33753499.0	2818967
1	138	ALB	Albania	Tirana	Europe	2842321.0	2866849.0	2882481.0	291339
2	34	DZA	Algeria	Algiers	Africa	44903225.0	43451666.0	39543154.0	3585634
3	213	ASM	American Samoa	Pago Pago	Oceania	44273.0	46189.0	51368.0	5484
4	203	AND	Andorra	Andorra la Vella	Europe	79824.0	77700.0	71746.0	7151

Add Floating Point

In [5]:
```
pd.set_option("display.float_format", lambda x: "%.2f" %x);
```

In [6]:
```
df
```

Out[6]:

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	2015 Population	F
0	36	AFG	Afghanistan	Kabul	Asia	41128771.00	38972230.00	33753499.00	28
1	138	ALB	Albania	Tirana	Europe	2842321.00	2866849.00	2882481.00	2
2	34	DZA	Algeria	Algiers	Africa	44903225.00	43451666.00	39543154.00	35
3	213	ASM	American Samoa	Pago Pago	Oceania	44273.00	46189.00	51368.00	
4	203	AND	Andorra	Andorra la Vella	Europe	79824.00	77700.00	71746.00	
...	...	...	...	...	...	...	...	...	
229	226	WLF	Wallis and Futuna	Mata-Utu	Oceania	11572.00	11655.00	12182.00	
230	172	ESH	Western Sahara	El Aaiún	Africa	575986.00	556048.00	491824.00	
231	46	YEM	Yemen	Sanaa	Asia	33696614.00	32284046.00	28516545.00	24
232	63	ZMB	Zambia	Lusaka	Africa	20017675.00	18927715.00	NaN	13
233	74	ZWE	Zimbabwe	Harare	Africa	16320537.00	15669666.00	14154937.00	12

234 rows × 17 columns

In [7]:

```
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                      234 non-null    object 
 3   Capital                      234 non-null    object 
 4   Continent                    234 non-null    object 
 5   2022 Population              230 non-null    float64
 6   2020 Population              233 non-null    float64
 7   2015 Population              230 non-null    float64
 8   2010 Population              227 non-null    float64
 9   2000 Population              227 non-null    float64
 10  1990 Population              229 non-null    float64
 11  1980 Population              229 non-null    float64
 12  1970 Population              230 non-null    float64
 13  Area (km²)                   232 non-null    float64
 14  Density (per km²)            230 non-null    float64
 15  Growth Rate                  232 non-null    float64
 16  World Population Percentage  234 non-null    float64
dtypes: float64(12), int64(1), object(4)
memory usage: 31.2+ KB
```

Get Some Statistical Info

In [8]:

```
df.describe()
```

Out[8]:

	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	
count	234.00	230.00	233.00	230.00	227.00	227.00	
mean	117.50	34632250.88	33600710.95	32066004.16	30270164.48	26840495.26	
std	67.69	137889172.44	135873196.61	131507146.34	126074183.54	113352454.57	
min	1.00	510.00	520.00	564.00	596.00	651.00	
25%	59.25	419738.50	406471.00	394295.00	382726.50	329470.00	
50%	117.50	5762857.00	5456681.00	5244415.00	4889741.00	4491202.00	
75%	175.75	22653719.00	21522626.00	19730853.75	16825852.50	15625467.00	
max	234.00	1425887337.00	1424929781.00	1393715448.00	1348191368.00	1264099069.00	1

In [9]:
```
df.isnull()
```

Out[9]:

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	2015 Population	201 Populatio
0	False	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	.
229	False	False	False	False	False	False	False	False	Fals
230	False	False	False	False	False	False	False	False	Fals
231	False	False	False	False	False	False	False	False	Fals
232	False	False	False	False	False	False	False	True	Fals
233	False	False	False	False	False	False	False	False	Fals

234 rows × 17 columns

Count Null Values

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
Rank                           0
CCA3                           0
Country                        0
Capital                        0
Continent                      0
2022 Population                4
2020 Population                1
2015 Population                4
2010 Population                7
2000 Population                7
1990 Population                5
1980 Population                5
1970 Population                4
Area (km²)                     2
Density (per km²)              4
Growth Rate                    2
World Population Percentage    0
dtype: int64
```

In [11]:
```
df.nunique()
```

Out[11]:
```
Rank                            234
CCA3                            234
Country                         234
Capital                         234
Continent                         6
2022 Population                 230
2020 Population                 233
2015 Population                 230
2010 Population                 227
2000 Population                 227
1990 Population                 229
1980 Population                 229
1970 Population                 230
Area (km²)                      231
Density (per km²)               230
Growth Rate                     178
World Population Percentage      70
dtype: int64
```

Sorting Values

In [12]:
```
df.sort_values(by="World Population Percentage",ascending = False).head(10)
```

Out[12]:

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	Pop
41	1	CHN	China	Beijing	Asia	1425887337.00	1424929781.00	1393715
92	2	IND	India	New Delhi	Asia	1417173173.00	1396387127.00	1322866
221	3	USA	United States	Washington, D.C.	North America	338289857.00	335942003.00	324607
93	4	IDN	Indonesia	Jakarta	Asia	275501339.00	271857970.00	259091
156	5	PAK	Pakistan	Islamabad	Asia	235824862.00	227196741.00	210969
149	6	NGA	Nigeria	Abuja	Africa	218541212.00	208327405.00	183995
27	7	BRA	Brazil	Brasilia	South America	215313498.00	213196304.00	205188
16	8	BGD	Bangladesh	Dhaka	Asia	171186372.00	167420951.00	157830
171	9	RUS	Russia	Moscow	Europe	144713314.00	145617329.00	144668
131	10	MEX	Mexico	Mexico City	North America	127504125.00	125998302.00	120149

In [57]:
```
df.corr()
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Cell In[57], line 1
----> 1 df.corr()

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:10054, in DataFram
e.corr(self, method, min_periods, numeric_only)
  10052 cols = data.columns
  10053 idx = cols.copy()
> 10054 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  10056 if method == "pearson":
  10057     correl = libalgos.nancorr(mat, minp=min_periods)

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:1838, in DataFram
e.to_numpy(self, dtype, copy, na_value)
   1836 if dtype is not None:
   1837     dtype = np.dtype(dtype)
-> 1838 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_va
lue)
   1839 if result.dtype is not dtype:
   1840     result = np.array(result, dtype=dtype, copy=False)

File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1732,
in BlockManager.as_array(self, dtype, copy, na_value)
   1730         arr.flags.writeable = False
   1731 else:
-> 1732     arr = self._interleave(dtype=dtype, na_value=na_value)
   1733     # The underlying data was copied within _interleave, so no nee
d
   1734     # to further copy if copy=True or setting na_value
   1736 if na_value is not lib.no_default:

File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1794,
in BlockManager._interleave(self, dtype, na_value)
   1792     else:
   1793         arr = blk.get_values(dtype)
-> 1794     result[rl.indexer] = arr
   1795     itemmask[rl.indexer] = 1
   1797 if not itemmask.all():

ValueError: could not convert string to float: 'AFG'
```

Now I have an error. Because correlation can be calculated only for numeric values,

In [15]:
```
numeric_df = df.select_dtypes(include = [float,int])
```

In [16]:
```
numeric_df
```

Out[16]:

	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population
0	36	41128771.00	38972230.00	33753499.00	28189672.00	19542982.00	10694796.00
1	138	2842321.00	2866849.00	2882481.00	2913399.00	3182021.00	3295066.00
2	34	44903225.00	43451666.00	39543154.00	35856344.00	30774621.00	25518074.00
3	213	44273.00	46189.00	51368.00	54849.00	58230.00	47818.00
4	203	79824.00	77700.00	71746.00	71519.00	66097.00	53569.00
...	...	...	...	...	...	...	...
229	226	11572.00	11655.00	12182.00	13142.00	14723.00	13454.00
230	172	575986.00	556048.00	491824.00	413296.00	270375.00	178529.00
231	46	33696614.00	32284046.00	28516545.00	24743946.00	18628700.00	13375121.00
232	63	20017675.00	18927715.00	NaN	13792086.00	9891136.00	7686401.00
233	74	16320537.00	15669666.00	14154937.00	12839771.00	11834676.00	10113893.00

234 rows × 13 columns

In [17]:
```
numeric_df.corr()
```

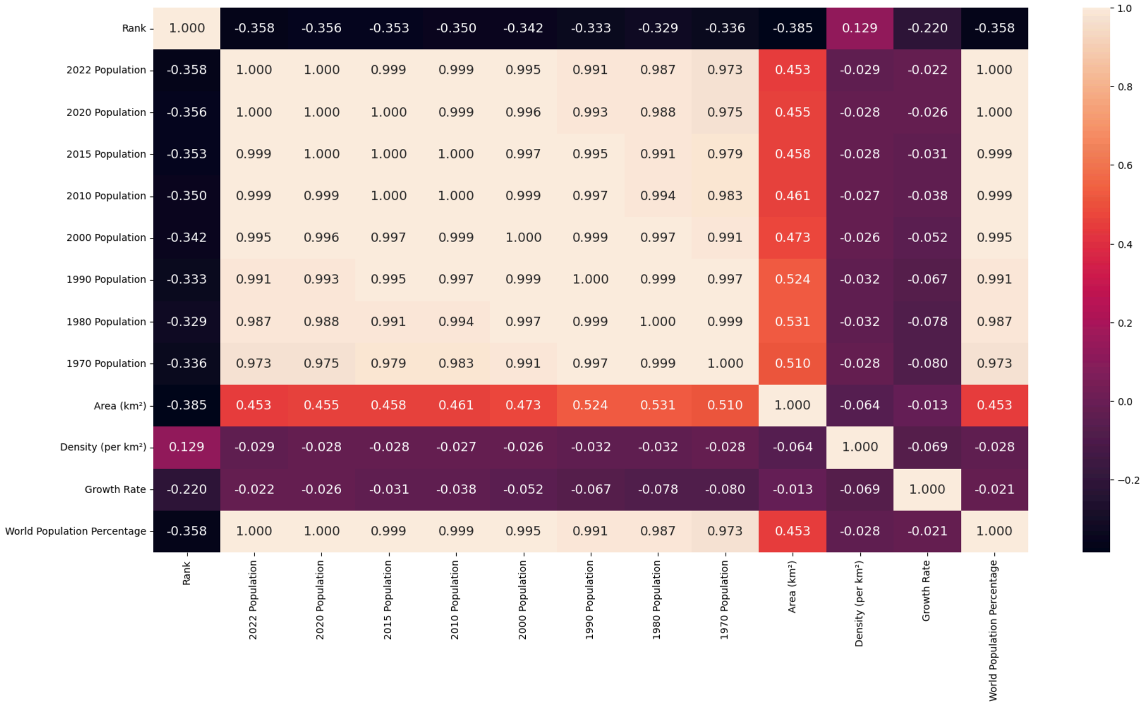
Out[17]:

	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	Pc
Rank	1.00	-0.36	-0.36	-0.35	-0.35	-0.34	-0.33	
2022 Population	-0.36	1.00	1.00	1.00	1.00	0.99	0.99	
2020 Population	-0.36	1.00	1.00	1.00	1.00	1.00	0.99	
2015 Population	-0.35	1.00	1.00	1.00	1.00	1.00	0.99	
2010 Population	-0.35	1.00	1.00	1.00	1.00	1.00	1.00	
2000 Population	-0.34	0.99	1.00	1.00	1.00	1.00	1.00	
1990 Population	-0.33	0.99	0.99	0.99	1.00	1.00	1.00	
1980 Population	-0.33	0.99	0.99	0.99	0.99	1.00	1.00	
1970 Population	-0.34	0.97	0.98	0.98	0.98	0.99	1.00	
Area (km²)	-0.38	0.45	0.45	0.46	0.46	0.47	0.52	
Density (per km²)	0.13	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	
Growth Rate	-0.22	-0.02	-0.03	-0.03	-0.04	-0.05	-0.07	
World Population Percentage	-0.36	1.00	1.00	1.00	1.00	0.99	0.99	

In [18]:
```
corr_matrix = numeric_df.corr()
```

In [34]:
```
plt.figure(figsize= (20,10))
sns.heatmap(corr_matrix,annot= True,fmt = '.3f',annot_kws = {"size":13})
plt.figure(figsize= (20,10))
```

Out[34]:
```
<Figure size 2000x1000 with 0 Axes>
```



	Rank	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
Rank	1.000	-0.358	-0.356	-0.353	-0.350	-0.342	-0.333	-0.329	-0.336	-0.385	0.129	-0.220	-0.358
2022 Population	-0.358	1.000	1.000	0.999	0.999	0.995	0.991	0.987	0.973	0.453	-0.029	-0.022	1.000
2020 Population	-0.356	1.000	1.000	1.000	0.999	0.996	0.993	0.988	0.975	0.455	-0.028	-0.026	1.000
2015 Population	-0.353	0.999	1.000	1.000	1.000	0.997	0.995	0.991	0.979	0.458	-0.028	-0.031	0.999
2010 Population	-0.350	0.999	0.999	1.000	1.000	0.999	0.997	0.994	0.983	0.461	-0.027	-0.038	0.999
2000 Population	-0.342	0.995	0.996	0.997	0.999	1.000	0.999	0.997	0.991	0.473	-0.026	-0.052	0.995
1990 Population	-0.333	0.991	0.993	0.995	0.997	0.999	1.000	0.999	0.997	0.524	-0.032	-0.067	0.991
1980 Population	-0.329	0.987	0.988	0.991	0.994	0.997	0.999	1.000	0.999	0.531	-0.032	-0.078	0.987
1970 Population	-0.336	0.973	0.975	0.979	0.983	0.991	0.997	0.999	1.000	0.510	-0.028	-0.080	0.973
Area (km²)	-0.385	0.453	0.455	0.458	0.461	0.473	0.524	0.531	0.510	1.000	-0.064	-0.013	0.453
Density (per km²)	0.129	-0.029	-0.028	-0.028	-0.027	-0.026	-0.032	-0.032	-0.028	-0.064	1.000	-0.069	-0.028
Growth Rate	-0.220	-0.022	-0.026	-0.031	-0.038	-0.052	-0.067	-0.078	-0.080	-0.013	-0.069	1.000	-0.021
World Population Percentage	-0.358	1.000	1.000	0.999	0.999	0.995	0.991	0.987	0.973	0.453	-0.028	-0.021	1.000

```
<Figure size 2000x1000 with 0 Axes>
```

In [52]:
```
df2 = df.groupby("Continent")[['1970 Population',
       '1980 Population', '1990 Population', '2000 Population',
       '2010 Population', '2015 Population', '2020 Population',
       '2022 Population']].mean().sort_values(by = '2022 Population',ascend:
df2
```

Out[52]:

	1970 Population	1980 Population	1990 Population	2000 Population	2010 Population	2015 Population	P
Continent							
Oceania	846968.26	996532.17	1162774.87	1357512.09	1613163.65	1756664.48	1
North America	7885865.15	9207334.03	10531660.62	12151739.60	13568016.28	14259596.25	14
Europe	13118479.82	14200004.52	14785203.94	14817685.71	14712278.68	15027454.12	14
Africa	6567175.27	8586031.98	11376964.52	14598365.95	18898197.31	21419703.57	23
South America	13781939.71	17270643.29	21224743.93	25015888.69	26789395.54	29509599.71	30
Asia	43839877.83	40278333.33	48639995.33	80580835.11	89087770.00	89165003.64	94

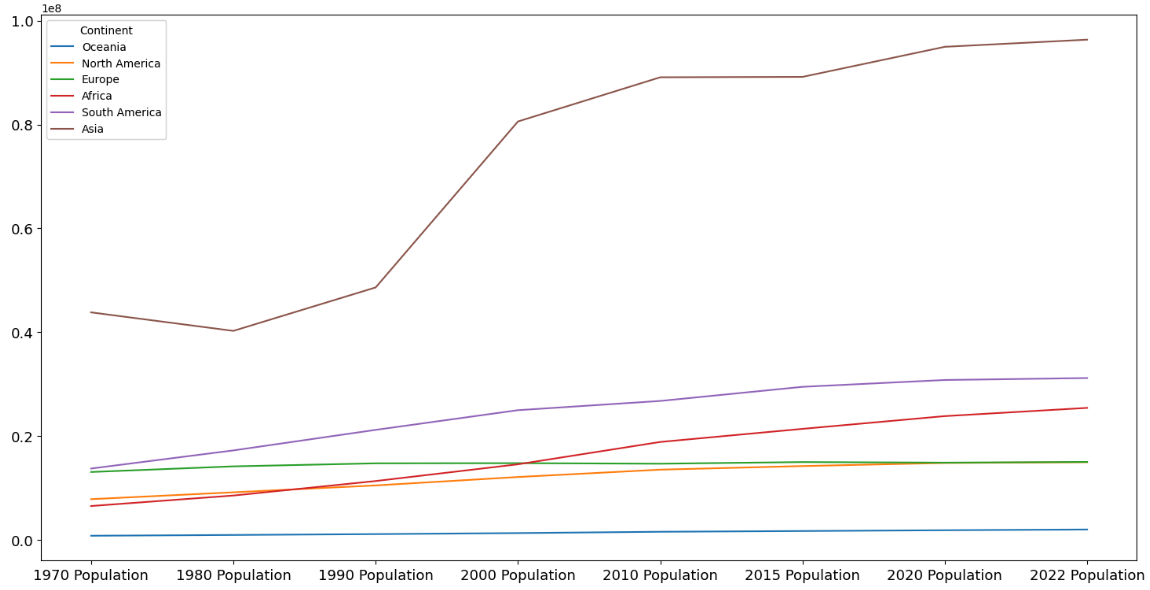
In [54]:
```
df3 = df2.transpose()
df3
```

Out[54]:

Continent	Oceania	North America	Europe	Africa	South America	Asia
1970 Population	846968.26	7885865.15	13118479.82	6567175.27	13781939.71	43839877.83
1980 Population	996532.17	9207334.03	14200004.52	8586031.98	17270643.29	40278333.33
1990 Population	1162774.87	10531660.62	14785203.94	11376964.52	21224743.93	48639995.33
2000 Population	1357512.09	12151739.60	14817685.71	14598365.95	25015888.69	80580835.11
2010 Population	1613163.65	13568016.28	14712278.68	18898197.31	26789395.54	89087770.00
2015 Population	1756664.48	14259596.25	15027454.12	21419703.57	29509599.71	89165003.64
2020 Population	1910148.96	14855914.82	14915843.92	23871435.26	30823574.50	94955134.37
2022 Population	2046386.32	15007403.40	15055371.82	25455879.68	31201186.29	96327387.31

In [75]:
```
df3.plot(figsize=(18,9),fontsize=13)
```

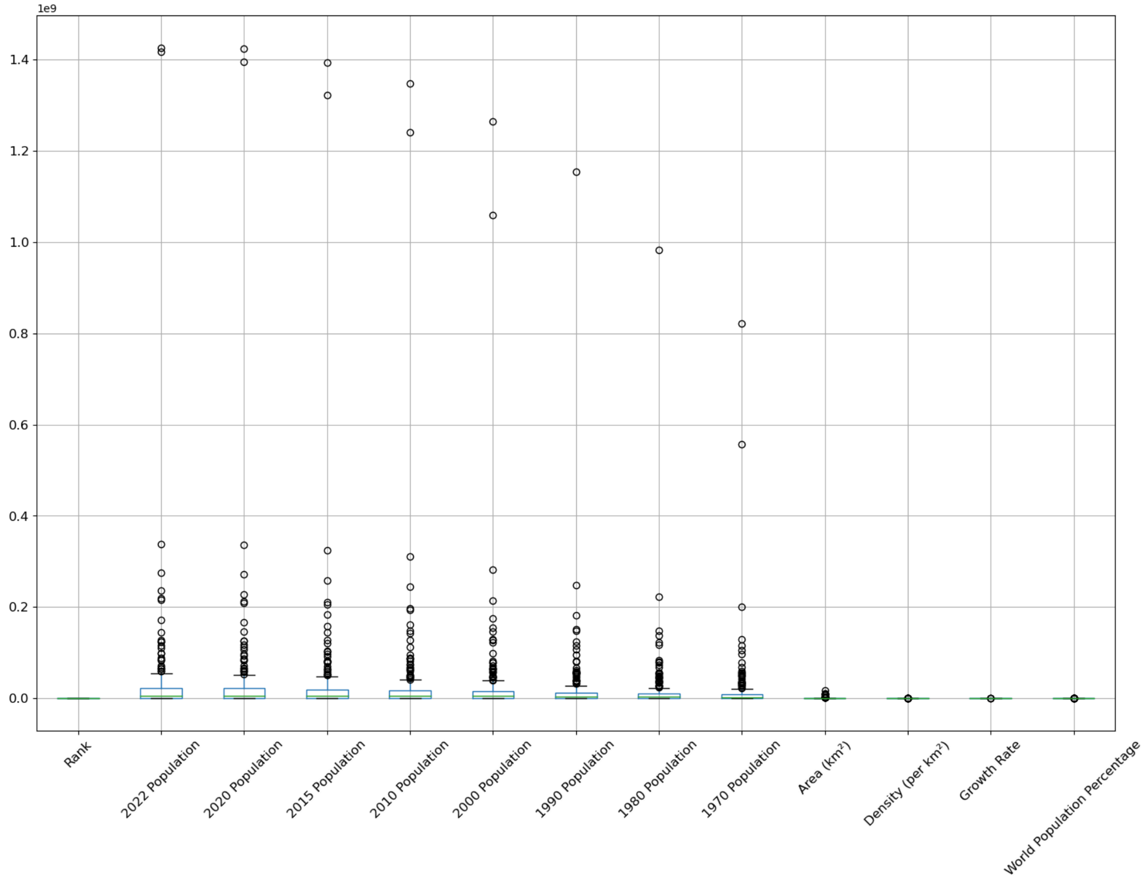
Out[75]:
```
<Axes: >
```



Check Outliers

```
In [90]: numeric_df.boxplot(figsize= (18,12),rot=45,fontsize = 12)
```

```
Out[90]: <Axes: >
```



Conclusion

In conclusion, The global population has grown steadily, with significant regional disparities. Asia remains the most populous continent, while Africa exhibits the highest growth rates. These trends underscore the need for region-specific policies addressing population management and resource allocation.