In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [9]:
```
df = pd.read_csv(r"C:\Users\kallzz\Desktop\Data Analytics Stuff\Data Analyst - Boot Camp\Python - Ju
pd.set_option('display.max.rows', 10)
```

In [10]:
```
pd.set_option('display.float_format', lambda x: '%.2f' % x)
```

In [11]:
```
df
```

Out[11]:

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	Popul
0	36	AFG	Afghanistan	Kabul	Asia	41128771.00	38972230.00	33753499.00	28189672.00	19542982.00	106947
1	138	ALB	Albania	Tirana	Europe	2842321.00	2866849.00	2882481.00	2913399.00	3182021.00	32950
2	34	DZA	Algeria	Algiers	Africa	44903225.00	43451666.00	39543154.00	35856344.00	30774621.00	255180
3	213	ASM	American Samoa	Pago Pago	Oceania	44273.00	46189.00	51368.00	54849.00	58230.00	478
4	203	AND	Andorra	Andorra la Vella	Europe	79824.00	77700.00	71746.00	71519.00	66097.00	535
...	...	...	...	...	...	...	...	...	...	...	
229	226	WLF	Wallis and Futuna	Mata-Utu	Oceania	11572.00	11655.00	12182.00	13142.00	14723.00	134
230	172	ESH	Western Sahara	El Aaiún	Africa	575986.00	556048.00	491824.00	413296.00	270375.00	1785
231	46	YEM	Yemen	Sanaa	Asia	33696614.00	32284046.00	28516545.00	24743946.00	18628700.00	133751
232	63	ZMB	Zambia	Lusaka	Africa	20017675.00	18927715.00	NaN	13792086.00	9891136.00	76864
233	74	ZWE	Zimbabwe	Harare	Africa	16320537.00	15669666.00	14154937.00	12839771.00	11834676.00	101138

234 rows × 17 columns

In [12]:
```
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
```

In [13]:
```
df.describe()
```

Out[13]:

	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	1980 Population	
count	234.00	230.00	233.00	230.00	227.00	227.00	229.00	229.00	
mean	117.50	34632250.88	33600710.95	32066004.16	30270164.48	26840495.26	19330463.93	16282884.78	1
std	67.69	137889172.44	135873196.61	131507146.34	126074183.54	113352454.57	81309624.96	69345465.54	6
min	1.00	510.00	520.00	564.00	596.00	651.00	700.00	733.00	
25%	59.25	419738.50	406471.00	394295.00	382726.50	329470.00	261928.00	223752.00	
50%	117.50	5762857.00	5456681.00	5244415.00	4889741.00	4491202.00	3785847.00	3135123.00	
75%	175.75	22653719.00	21522626.00	19730853.75	16825852.50	15625467.00	11882762.00	9817257.00	
max	234.00	1425887337.00	1424929781.00	1393715448.00	1348191368.00	1264099069.00	1153704252.00	982372466.00	82

In [14]:
```
df.isnull().sum()
```

Out[14]:
```
Rank                           0
CCA3                           0
Country                        0
Capital                        0
Continent                      0
                              ..
1970 Population                4
Area (km²)                     2
Density (per km²)              4
Growth Rate                    2
World Population Percentage    0
Length: 17, dtype: int64
```

In [16]:
```
df.nunique()
```

Out[16]:
```
Rank                           234
CCA3                           234
Country                        234
Capital                        234
Continent                        6
                              ...
1970 Population                230
Area (km²)                     231
Density (per km²)              230
Growth Rate                    178
World Population Percentage     70
Length: 17, dtype: int64
```

In [19]:
```
df.sort_values('2022 Population', ascending = False).head()
```

Out[19]:

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	2015 Population	2010 Population	Popul
41	1	CHN	China	Beijing	Asia	1425887337.00	1424929781.00	1393715448.00	1348191368.00	126409906
92	2	IND	India	New Delhi	Asia	1417173173.00	1396387127.00	1322866505.00	1240613620.00	105963367
221	3	USA	United States	Washington, D.C.	North America	338289857.00	335942003.00	324607776.00	311182845.00	2823985
93	4	IDN	Indonesia	Jakarta	Asia	275501339.00	271857970.00	259091970.00	244016173.00	2140724
156	5	PAK	Pakistan	Islamabad	Asia	235824862.00	227196741.00	210969298.00	194454498.00	1543699

In [20]:
```
df.corr()
```

```
C:\Users\kallzz\AppData\Local\Temp\ipykernel_13860\1134722465.py:1: FutureWarning: The default valu
e of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. S
elect only valid columns or specify the value of numeric_only to silence this warning.
  df.corr()
```

Out[20]:

	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	1980 Population	1970 Population	Area (km²)	De
Rank	1.00	-0.36	-0.36	-0.35	-0.35	-0.34	-0.33	-0.33	-0.34	-0.38	
2022 Population	-0.36	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.97	0.45	
2020 Population	-0.36	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.98	0.45	
2015 Population	-0.35	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.98	0.46	
2010 Population	-0.35	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.98	0.46	
...	...	...	...	...	...	...	...	...	...	...	
1970 Population	-0.34	0.97	0.98	0.98	0.98	0.99	1.00	1.00	1.00	0.51	
Area (km²)	-0.38	0.45	0.45	0.46	0.46	0.47	0.52	0.53	0.51	1.00	
Density (per km²)	0.13	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.06	
Growth Rate	-0.22	-0.02	-0.03	-0.03	-0.04	-0.05	-0.07	-0.08	-0.08	-0.01	
World Population Percentage	-0.36	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.97	0.45	

13 rows × 13 columns

In [21]:
```
sns.heatmap(df.corr(), annot = True)
plt.rcParams['figure.figsize'] = (10,7)

plt.show()
```

```
C:\Users\kallzz\AppData\Local\Temp\ipykernel_13860\3346872259.py:1: FutureWarning: The default valu
e of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. S
elect only valid columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(df.corr(), annot = True)
```

In [22]:
```
df.groupby('Continent').mean()
```

```
C:\Users\kallzz\AppData\Local\Temp\ipykernel_13860\3700721160.py:1: FutureWarning: The default valu
e of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will de
fault to False. Either specify numeric_only or select only columns which should be valid for the fu
nction.
  df.groupby('Continent').mean()
```

Out[22]:

Continent	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	1980 Population	1970 Population
Africa	92.16	25455879.68	23871435.26	21419703.57	18898197.31	14598365.95	11376964.52	8586031.98	6567175.27
Asia	77.56	96327387.31	94955134.37	89165003.64	89087770.00	80580835.11	48639995.33	40278333.33	43839877.83
Europe	124.50	15055371.82	14915843.92	15027454.12	14712278.68	14817685.71	14785203.94	14200004.52	13118479.82
North America	160.93	15007403.40	14855914.82	14259596.25	13568016.28	12151739.60	10531660.62	9207334.03	7885865.15
Oceania	188.52	2046386.32	1910148.96	1756664.48	1613163.65	1357512.09	1162774.87	996532.17	846968.26
South America	97.57	31201186.29	30823574.50	29509599.71	26789395.54	25015888.69	21224743.93	17270643.29	13781939.71

In [24]:
```
df2 = df.groupby('Continent').mean().sort_values(by = '2022 Population', ascending = False)
df2
```

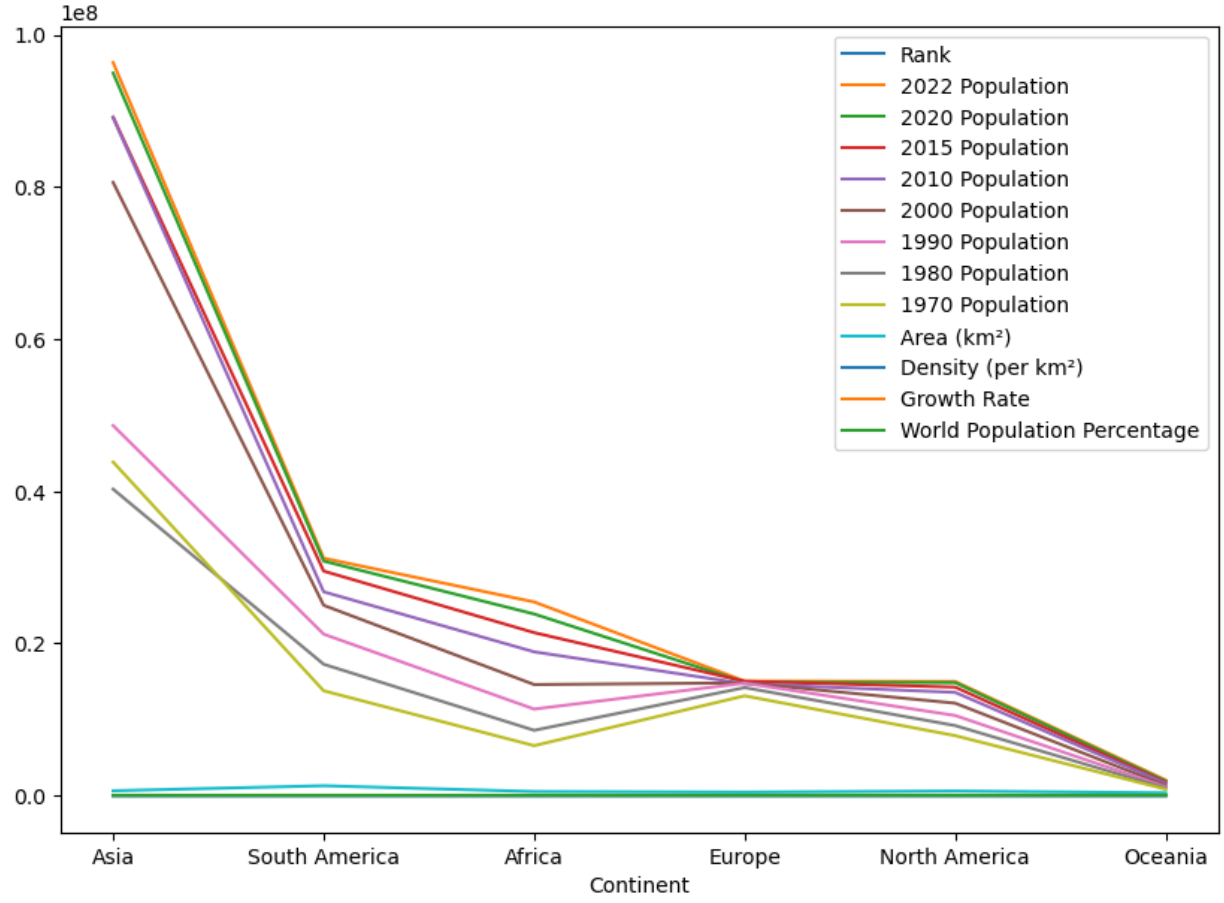
```
C:\Users\kallzz\AppData\Local\Temp\ipykernel_13860\242022662.py:1: FutureWarning: The default value
of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will defa
ult to False. Either specify numeric_only or select only columns which should be valid for the func
tion.
  df2 = df.groupby('Continent').mean().sort_values(by = '2022 Population', ascending = False)
```

Out[24]:

Continent	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	1980 Population	1970 Population
Asia	77.56	96327387.31	94955134.37	89165003.64	89087770.00	80580835.11	48639995.33	40278333.33	43839877.83
South America	97.57	31201186.29	30823574.50	29509599.71	26789395.54	25015888.69	21224743.93	17270643.29	13781939.71
Africa	92.16	25455879.68	23871435.26	21419703.57	18898197.31	14598365.95	11376964.52	8586031.98	6567175.27
Europe	124.50	15055371.82	14915843.92	15027454.12	14712278.68	14817685.71	14785203.94	14200004.52	13118479.82
North America	160.93	15007403.40	14855914.82	14259596.25	13568016.28	12151739.60	10531660.62	9207334.03	7885865.15
Oceania	188.52	2046386.32	1910148.96	1756664.48	1613163.65	1357512.09	1162774.87	996532.17	846968.26

In [25]:
```
# To view only the population trend at continent level
df2.plot()
```

Out[25]: <Axes: xlabel='Continent'>

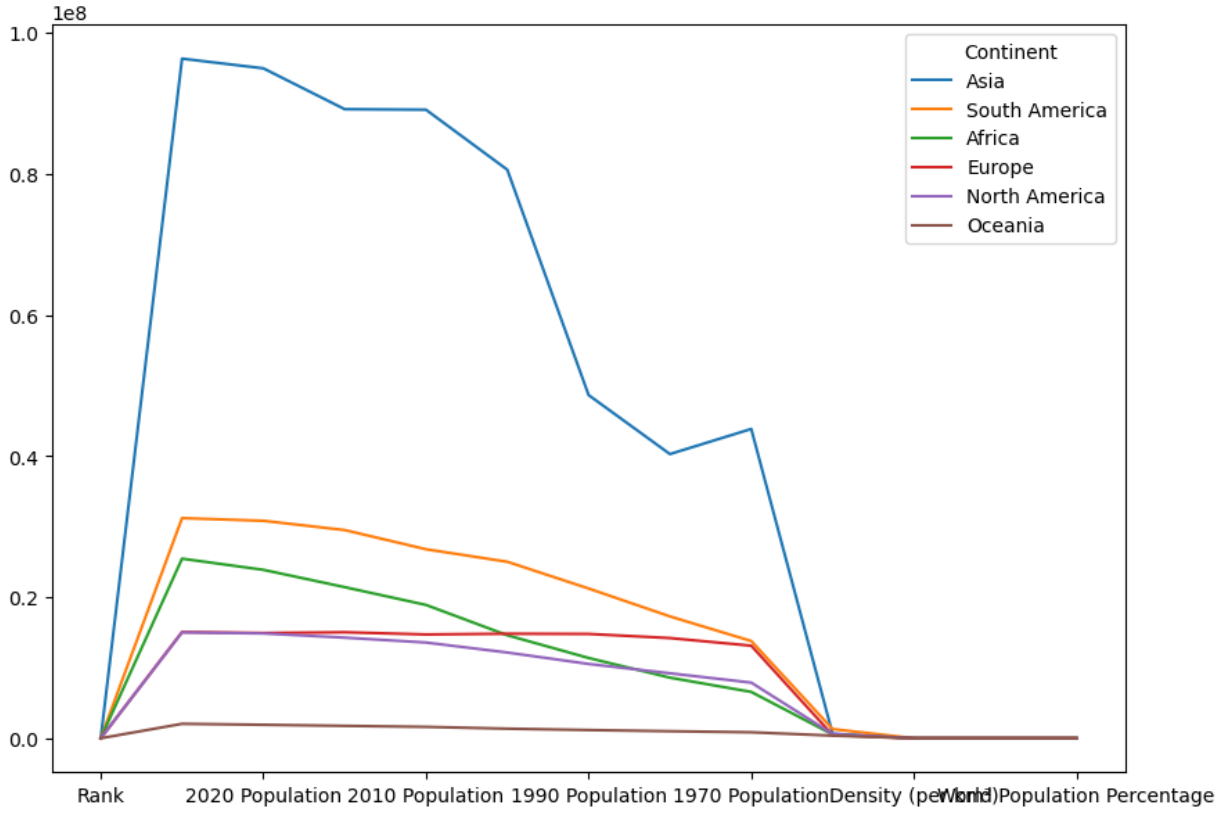


In [31]:
```
# to transpose
df2 = df2.transpose()
```

In [32]:
```
df2.plot()

# lines are dipping at the end becoz of unwanted columns in the analysis like pop density, average..
```

Out[32]: <Axes: >



In [33]:
```
df.columns
```

Out[33]:
```
Index(['Rank', 'CCA3', 'Country', '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')
```

In [36]:
```
df3 = df.groupby('Continent')[['2022 Population',
       '2020 Population', '2015 Population', '2010 Population',
       '2000 Population', '1990 Population', '1980 Population',
       '1970 Population']].mean().sort_values('2022 Population', ascending = False)
df3
```

Out[36]:

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

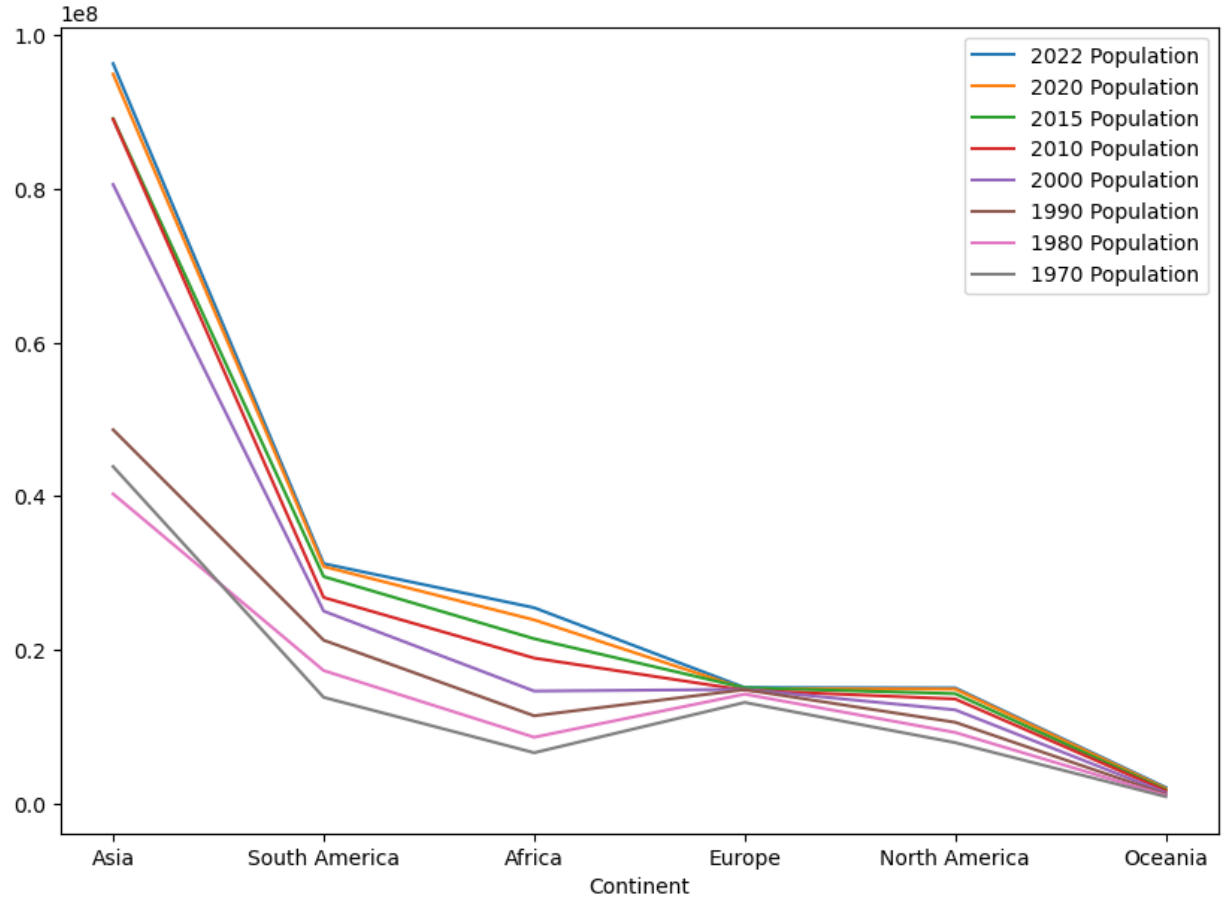
In [37]:
```
# another way to choose specific columns for analysis
df3 = df.groupby('Continent')[df.columns[5:13]].mean().sort_values('2022 Population', ascending = Fa
df3
```

Out[37]:

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

In [38]:
```
df3.plot()
```

Out[38]: <Axes: xlabel='Continent'>



In [41]:
```
# data plotted from 2022 --> 1970 which is reverse and not recommended to find out the trend. Change

df3 = df.groupby('Continent')[['1970 Population',
       '1980 Population', '1990 Population', '2000 Population',
       '2010 Population', '2015 Population', '2020 Population',
       '2022 Population']].mean().sort_values('2022 Population', ascending = False)
df3
```

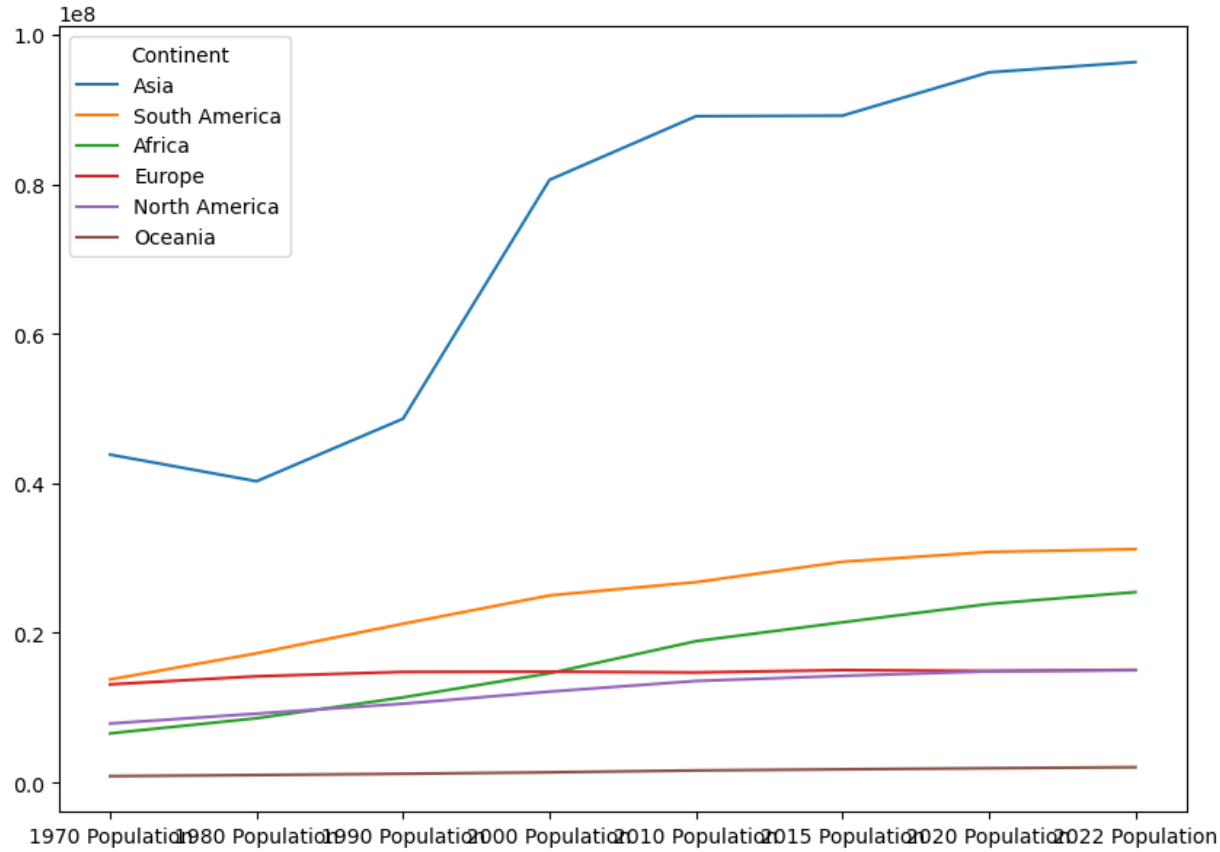
Out[41]:

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

In [43]:
```
df4 = df3.transpose()
```

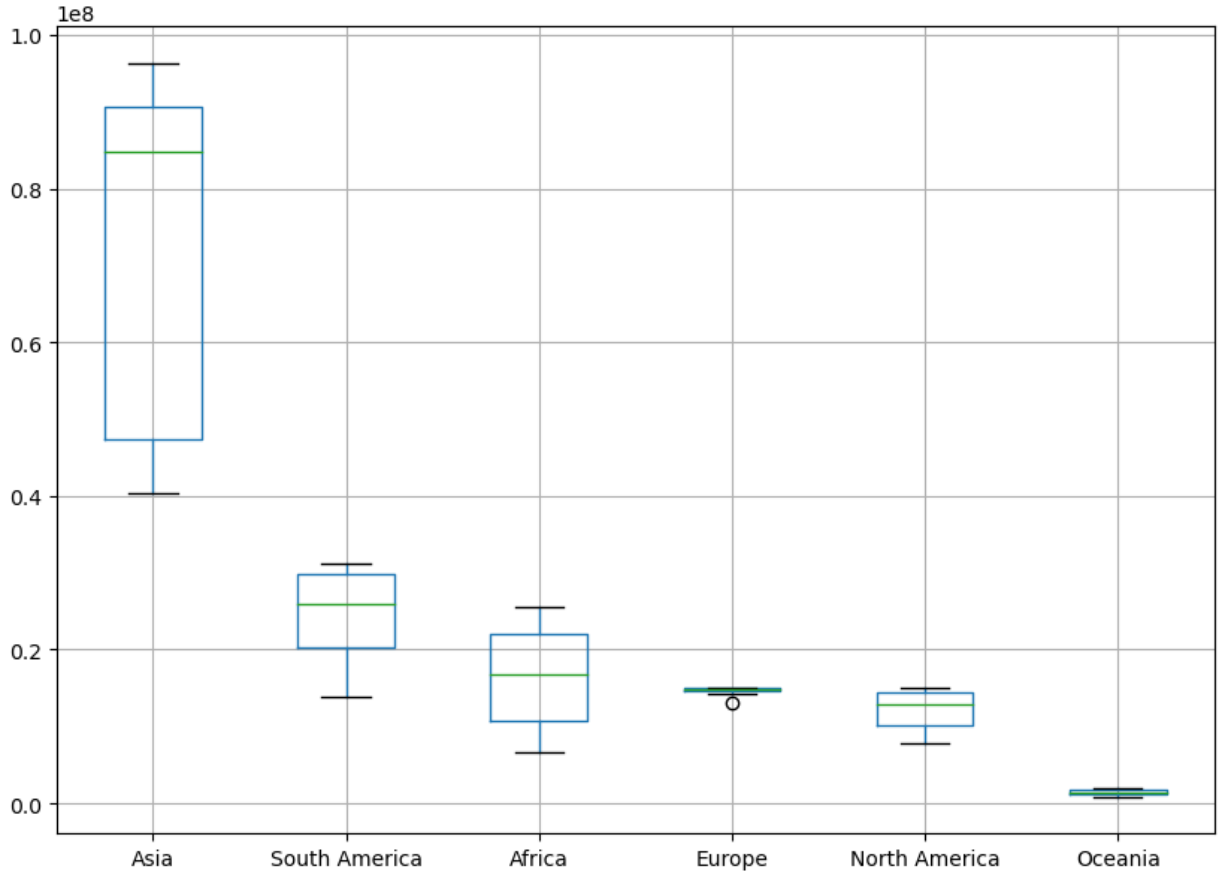
In [44]:
```
df4.plot()
```

Out[44]: <Axes: >



In [45]:
```
df4.boxplot()
```

Out[45]: <Axes: >



In [46]:
```
df.select_dtypes(include = 'number')
```

Out[46]:

	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	1980 Population	1970 Population	Area (
0	36	41128771.00	38972230.00	33753499.00	28189672.00	19542982.00	10694796.00	12486631.00	10752971.00	6522
1	138	2842321.00	2866849.00	2882481.00	2913399.00	3182021.00	3295066.00	2941651.00	2324731.00	287
2	34	44903225.00	43451666.00	39543154.00	35856344.00	30774621.00	25518074.00	18739378.00	13795915.00	2381
3	213	44273.00	46189.00	51368.00	54849.00	58230.00	47818.00	32886.00	27075.00	1
4	203	79824.00	77700.00	71746.00	71519.00	66097.00	53569.00	35611.00	19860.00	4
...	...	...	...	...	...	...	...	...	...	
229	226	11572.00	11655.00	12182.00	13142.00	14723.00	13454.00	11315.00	9377.00	1
230	172	575986.00	556048.00	491824.00	413296.00	270375.00	178529.00	116775.00	76371.00	2660
231	46	33696614.00	32284046.00	28516545.00	24743946.00	18628700.00	13375121.00	9204938.00	6843607.00	5279
232	63	20017675.00	18927715.00	NaN	13792086.00	9891136.00	7686401.00	5720438.00	4281671.00	7526
233	74	16320537.00	15669666.00	14154937.00	12839771.00	11834676.00	10113893.00	7049926.00	5202918.00	3907

234 rows × 13 columns

In [47]:
```
df.select_dtypes(include = 'object')
```

Out[47]:

	CCA3	Country	Capital	Continent
0	AFG	Afghanistan	Kabul	Asia
1	ALB	Albania	Tirana	Europe
2	DZA	Algeria	Algiers	Africa
3	ASM	American Samoa	Pago Pago	Oceania
4	AND	Andorra	Andorra la Vella	Europe
...	...	...	...	...
229	WLF	Wallis and Futuna	Mata-Utu	Oceania
230	ESH	Western Sahara	El Aaiún	Africa
231	YEM	Yemen	Sanaa	Asia
232	ZMB	Zambia	Lusaka	Africa
233	ZWE	Zimbabwe	Harare	Africa

234 rows × 4 columns

In [ ]:
```

```