Import libraries

In [20]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
```

In [21]:
```
df= pd.read_csv(r'/Users/nikhillohar/Downloads/world_population.csv')
df
```

Out[21]:

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	Popu
0	36	AFG	Afghanistan	Kabul	Asia	41128771.00	38972230.00	33753
1	138	ALB	Albania	Tirana	Europe	2842321.00	2866849.00	2882
2	34	DZA	Algeria	Algiers	Africa	44903225.00	43451666.00	39543
3	213	ASM	American Samoa	Pago Pago	Oceania	44273.00	46189.00	51
4	203	AND	Andorra	Andorra la Vella	Europe	79824.00	77700.00	71
...	...	...	...	...	...	...	...	
229	226	WLF	Wallis and Futuna	Mata-Utu	Oceania	11572.00	11655.00	12
230	172	ESH	Western Sahara	El Aaiún	Africa	575986.00	556048.00	491
231	46	YEM	Yemen	Sanaa	Asia	33696614.00	32284046.00	28516
232	63	ZMB	Zambia	Lusaka	Africa	20017675.00	18927715.00	
233	74	ZWE	Zimbabwe	Harare	Africa	16320537.00	15669666.00	14154

234 rows × 17 columns

In [22]:
```
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
```

Based on the result we can see that our data set has 17 distinct columns. Also, this shows data type for each column and non null count for each column

In [23]:
```
df.describe()
```

Out[23]:

	Rank	2022 Population	2020 Population	2015 Population	2010 Population	Pop
count	234.00	230.00	233.00	230.00	227.00	
mean	117.50	34632250.88	33600710.95	32066004.16	30270164.48	26840
std	67.69	137889172.44	135873196.61	131507146.34	126074183.54	113352
min	1.00	510.00	520.00	564.00	596.00	
25%	59.25	419738.50	406471.00	394295.00	382726.50	329
50%	117.50	5762857.00	5456681.00	5244415.00	4889741.00	4491
75%	175.75	22653719.00	21522626.00	19730853.75	16825852.50	15625
max	234.00	1425887337.00	1424929781.00	1393715448.00	1348191368.00	1264099

In [24]:
```
pd.set_option('display.float_format',lambda x:'%.2f' % x)
```

In [25]:
```
df.describe()
```

Out[25]:

	Rank	2022 Population	2020 Population	2015 Population	2010 Population	Pop
count	234.00	230.00	233.00	230.00	227.00	
mean	117.50	34632250.88	33600710.95	32066004.16	30270164.48	26840
std	67.69	137889172.44	135873196.61	131507146.34	126074183.54	113352
min	1.00	510.00	520.00	564.00	596.00	
25%	59.25	419738.50	406471.00	394295.00	382726.50	329
50%	117.50	5762857.00	5456681.00	5244415.00	4889741.00	4491
75%	175.75	22653719.00	21522626.00	19730853.75	16825852.50	15625
max	234.00	1425887337.00	1424929781.00	1393715448.00	1348191368.00	1264099

In [26]:
```
df.isnull().sum()
```

Out[26]:
```
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 [27]:
```
df.nunique()
```

Out[27]:
```
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
```

This will show us how many unique values it contains

In [28]:
```
df.sort_values(by='2022 Population').head()
```

Out[28]:

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	201 Populatio
226	234	VAT	Vatican City	Vatican City	Europe	510.00	520.00	564.0
209	233	TKL	Tokelau	Nukunonu	Oceania	1871.00	1827.00	1454.0
150	232	NIU	Niue	Alofi	Oceania	1934.00	1942.00	1847.0
64	231	FLK	Falkland Islands	Stanley	South America	3780.00	3747.00	3408.0
137	230	MSR	Montserrat	Brades	North America	4390.00	4500.00	5059.0

This will give us sorted list of high population

Now we will find the correlation between the columns

In [29]:
```
df.corr(method='pearson',numeric_only=True)
```

Out[29]:

	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	19 Populat
Rank	1.00	-0.36	-0.36	-0.35	-0.35	-0.34	-0
2022 Population	-0.36	1.00	1.00	1.00	1.00	0.99	0
2020 Population	-0.36	1.00	1.00	1.00	1.00	1.00	0
2015 Population	-0.35	1.00	1.00	1.00	1.00	1.00	0
2010 Population	-0.35	1.00	1.00	1.00	1.00	1.00	1
2000 Population	-0.34	0.99	1.00	1.00	1.00	1.00	1
1990 Population	-0.33	0.99	0.99	0.99	1.00	1.00	1
1980 Population	-0.33	0.99	0.99	0.99	0.99	1.00	1
1970 Population	-0.34	0.97	0.98	0.98	0.98	0.99	1
Area (km²)	-0.38	0.45	0.45	0.46	0.46	0.47	0
Density (per km²)	0.13	-0.03	-0.03	-0.03	-0.03	-0.03	-0
Growth Rate	-0.22	-0.02	-0.03	-0.03	-0.04	-0.05	-0
World Population Percentage	-0.36	1.00	1.00	1.00	1.00	0.99	0

This will compare each column with every other column in the dataframe.

Now we will look at the continents to see how each continent.

In [30]:
```
df.groupby(by='Continent').mean(numeric_only=True).sort_values('2022 Populat
```

Out[30]:

Continent	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population
Asia	77.56	96327387.31	94955134.37	89165003.64	89087770.00	80580835.1
South America	97.57	31201186.29	30823574.50	29509599.71	26789395.54	25015888.69
Africa	92.16	25455879.68	23871435.26	21419703.57	18898197.31	14598365.95
Europe	124.50	15055371.82	14915843.92	15027454.12	14712278.68	14817685.7
North America	160.93	15007403.40	14855914.82	14259596.25	13568016.28	12151739.60
Oceania	188.52	2046386.32	1910148.96	1756664.48	1613163.65	1357512.09

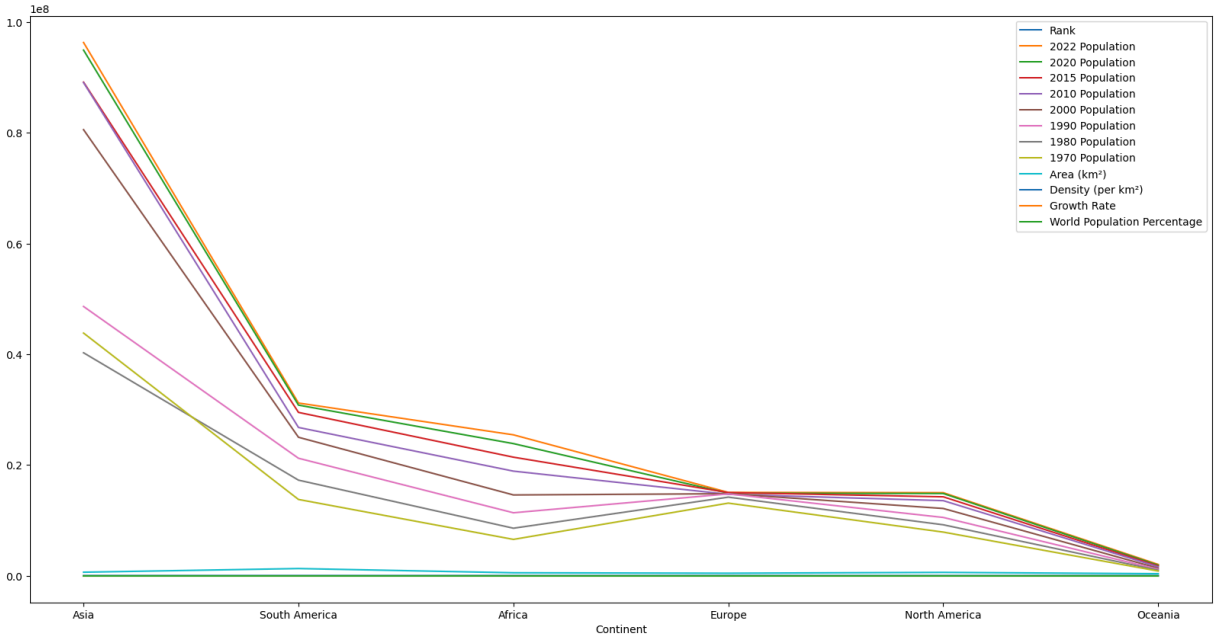
In [31]:
```
sns.heatmap(data=df.corr(method='pearson',numeric_only=True), annot=True,squ
plt.show()
```

Based on this heatmap, we can see that the yearly population is highly correlated. On the other hand rank is negatively correlated which we expected.

From the result we can see that the Asia has the highest density and each of the countries make up about 1% of the world population. Lets plot a graph to visualize this.

In [32]:
```
df2=df.groupby(by='Continent').mean(numeric_only=True).sort_values('2022 Pop
df2.plot(figsize=(20,10))
```

Out[32]: <Axes: xlabel='Continent'>



We have unwanted columns in the dataframe that does not make sense in this graph; so we will plot a graph with years on the X-axis.

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

Out[33]:
```
Index(['Rank', 'CCA3', 'Country', 'Capital', 'Continent', '2022 Populatio
n',
       '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 [34]:
```
df2=df.groupby(by='Continent')[['1970 Population',
       '1980 Population', '1990 Population', '2000 Population',
       '2010 Population', '2015 Population', '2020 Population',
       '2022 Population']].mean(numeric_only=True).sort_values('2022 Populat
df2
```

Out[34]:

Continent	1970 Population	1980 Population	1990 Population	2000 Population	2010 Population	Pop
Asia	43839877.83	40278333.33	48639995.33	80580835.11	89087770.00	89165
South America	13781939.71	17270643.29	21224743.93	25015888.69	26789395.54	2950
Africa	6567175.27	8586031.98	11376964.52	14598365.95	18898197.31	2141
Europe	13118479.82	14200004.52	14785203.94	14817685.71	14712278.68	1502
North America	7885865.15	9207334.03	10531660.62	12151739.60	13568016.28	1425
Oceania	846968.26	996532.17	1162774.87	1357512.09	1613163.65	1756

In [35]:
```
df3=df2.transpose()
```

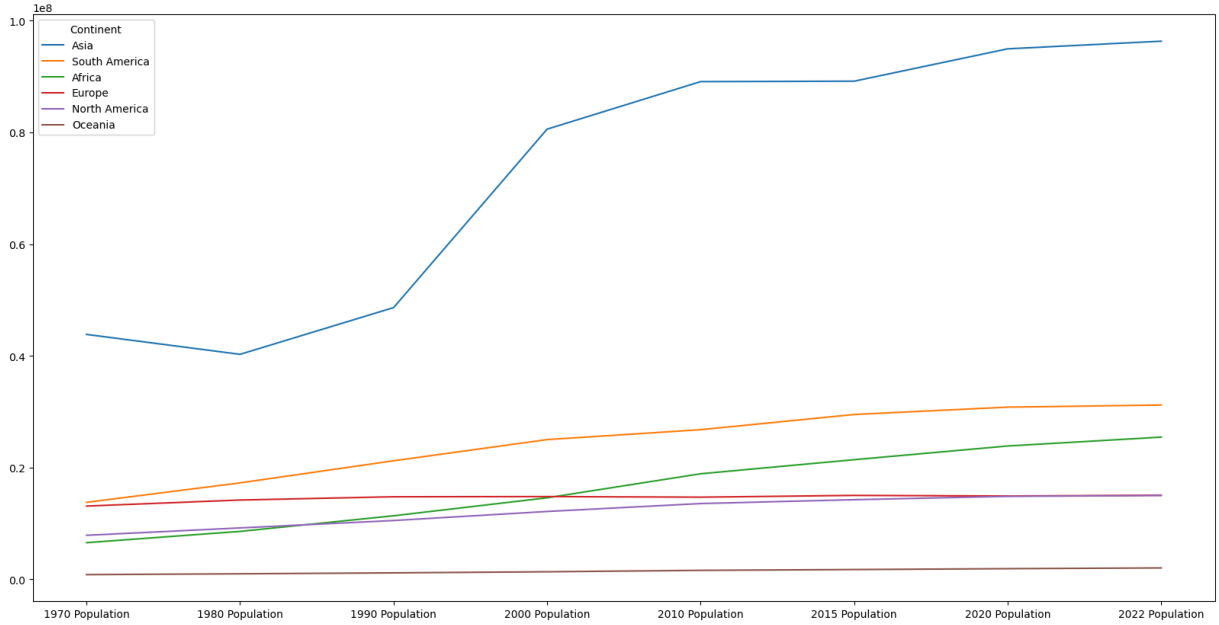
In [36]:
```
df3
```

Out[36]:

Continent	Asia	South America	Africa	Europe	North America	O
1970 Population	43839877.83	13781939.71	6567175.27	13118479.82	7885865.15	846
1980 Population	40278333.33	17270643.29	8586031.98	14200004.52	9207334.03	996
1990 Population	48639995.33	21224743.93	11376964.52	14785203.94	10531660.62	1162
2000 Population	80580835.11	25015888.69	14598365.95	14817685.71	12151739.60	1357
2010 Population	89087770.00	26789395.54	18898197.31	14712278.68	13568016.28	1613
2015 Population	89165003.64	29509599.71	21419703.57	15027454.12	14259596.25	1756
2020 Population	94955134.37	30823574.50	23871435.26	14915843.92	14855914.82	1910
2022 Population	96327387.31	31201186.29	25455879.68	15055371.82	15007403.40	2046

In [37]:
```
df3.plot(figsize=(20,10))
```

Out[37]:
```
<Axes: >
```

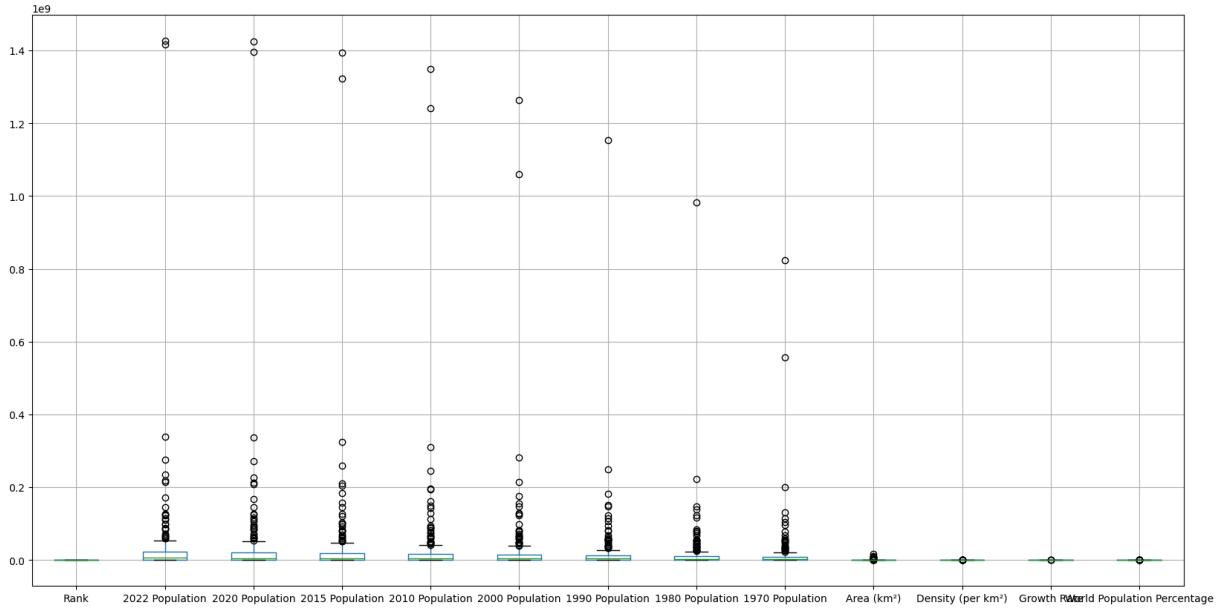


After converting the data, we can visualize that the population of south america and africa also growing at constant rate.

Let's plot a box plot to visualize the distribution

In [38]:
```
df.boxplot(figsize=(20,10))
```

Out[38]: <Axes: >



Based on the box plot we can see that our dataset contains many outliers. This is due to the fact that lot of countries has different area as well as different population density.

This is a demo on EDA. It is an intial impression of the dataset and trying to understand the attributes and the data. This can be performed in multiple ways and it entirely depends on the analyst.