Exploratory Data Analysis(EDA)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df = pd.read_csv("/content/world_population.csv")
df
```

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population	1980 Population
0	36	AFG	Afghanistan	Kabul	Asia	41128771.00	38972230.00	33753499.00	28189672.00	19542982.00	10694796.00	12486631.00
1	138	ALB	Albania	Tirana	Europe	2842321.00	2866849.00	2882481.00	2913399.00	3182021.00	3295066.00	2941651.00
2	34	DZA	Algeria	Algiers	Africa	44903225.00	43451666.00	39543154.00	35856344.00	30774621.00	25518074.00	18739378.00
3	213	ASM	American Samoa	Pago Pago	Oceania	44273.00	46189.00	51368.00	54849.00	58230.00	47818.00	32886.00
4	203	AND	Andorra	Andorra la Vella	Europe	79824.00	77700.00	71746.00	71519.00	66097.00	53569.00	35611.00
...	...	...	...	...	...	...	...	...	...	...	...	...
229	226	WLF	Wallis and Futuna	Mata-Utu	Oceania	11572.00	11655.00	12182.00	13142.00	14723.00	13454.00	11315.00
230	172	ESH	Western Sahara	El Aaiún	Africa	575986.00	556048.00	491824.00	413296.00	270375.00	178529.00	116775.00
231	46	YEM	Yemen	Sanaa	Asia	33696614.00	32284046.00	28516545.00	24743946.00	18628700.00	13375121.00	9204938.00
232	63	ZMB	Zambia	Lusaka	Africa	20017675.00	18927715.00	NaN	13792086.00	9891136.00	7686401.00	5720438.00
233	74	ZWE	Zimbabwe	Harare	Africa	16320537.00	15669666.00	14154937.00	12839771.00	11834676.00	10113893.00	7049926.00

234 rows × 17 columns

Next steps: Generate code with df View recommended plots

```
# Sets pandas display option to format all floating-point numbers with 2 decimal places
pd.set_option('display.float_format', lambda x: '%.2f' % x)
```

```
df.info()  # Displays concise summary of DataFrame including index, column names, non-null values, and memory usage
```

```
<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
```

```
df.describe() # Generates descriptive statistics of DataFrame columns including count, mean, std, min, 25%, 50%, 75%, max
```

	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

```
df.isnull().sum() # Checks for missing values in DataFrame columns
```

```
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
```

```
df.nunique() # Returns number of unique values in each column
```

```
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
```

```
df.sort_values(by="World Population Percentage", ascending=False).head(10) # Sorts DataFrame by a specified column in ascending order
```

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	Popu
41	1	CHN	China	Beijing	Asia	1425887337.00	1424929781.00	139371
92	2	IND	India	New Delhi	Asia	1417173173.00	1396387127.00	132286
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	25909
156	5	PAK	Pakistan	Islamabad	Asia	235824862.00	227196741.00	21096
149	6	NGA	Nigeria	Abuja	Africa	218541212.00	208327405.00	18399
27	7	BRA	Brazil	Brasilia	South America	215313498.00	213196304.00	20518
16	8	BGD	Bangladesh	Dhaka	Asia	171186372.00	167420951.00	15783
171	9	RUS	Russia	Moscow	Europe	144713314.00	145617329.00	14466
131	10	MEX	Mexico	Mexico City	North America	127504125.00	125998302.00	12014

```
df['2022 Population'].fillna(df['2022 Population'].mean(), inplace=True)
df['2020 Population'].fillna(df['2020 Population'].mean(), inplace=True)
df['2015 Population'].fillna(df['2015 Population'].mean(), inplace=True)
df['2010 Population'].fillna(df['2010 Population'].mean(), inplace=True)
df['2000 Population'].fillna(df['2000 Population'].mean(), inplace=True)
df['1990 Population'].fillna(df['1990 Population'].mean(), inplace=True)
df['1980 Population'].fillna(df['1980 Population'].mean(), inplace=True)
df['1970 Population'].fillna(df['1970 Population'].mean(), inplace=True)
df['Area (km²)'].fillna(df['Area (km²)'].mean(), inplace=True)
df['Density (per km²)'].fillna(df['Density (per km²)'].mean(), inplace=True)
df['Growth Rate'].fillna(df['Growth Rate'].mean(), inplace=True)
df['World Population Percentage'].fillna(df['World Population Percentage'].mean(), inplace=True)
```

```
df.isnull().sum() # Checks for missing values in DataFrame columns after filling missing values with mean
```

```
Rank                          0
CCA3                          0
Country                       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
```

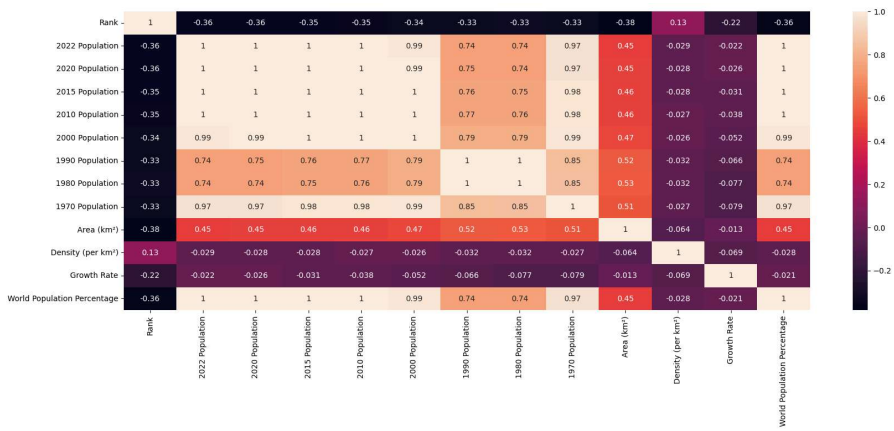
```
# Calculate correlation matrix for numeric columns only, excluding non-numeric data
numeric_df = df.select_dtypes(include=['float64', 'int64'])
numeric_df.corr()
```

	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	199 Populatio
Rank	1.00	-0.36	-0.36	-0.35	-0.35	-0.34	-0.3
2022 Population	-0.36	1.00	1.00	1.00	1.00	0.99	0.7
2020 Population	-0.36	1.00	1.00	1.00	1.00	0.99	0.7
2015 Population	-0.35	1.00	1.00	1.00	1.00	1.00	0.7
2010 Population	-0.35	1.00	1.00	1.00	1.00	1.00	0.7
2000 Population	-0.34	0.99	0.99	1.00	1.00	1.00	0.7
1990 Population	-0.33	0.74	0.75	0.76	0.77	0.79	1.0
1980 Population	-0.33	0.74	0.74	0.75	0.76	0.79	1.0
1970 Population	-0.33	0.97	0.97	0.98	0.98	0.99	0.8
Area (km²)	-0.38	0.45	0.45	0.46	0.46	0.47	0.5
Density (per km²)	0.13	-0.03	-0.03	-0.03	-0.03	-0.03	-0.0
Growth Rate	-0.22	-0.02	-0.03	-0.03	-0.04	-0.05	-0.0
World Population Percentage	-0.36	1.00	1.00	1.00	1.00	0.99	0.7

```
sns.heatmap(numeric_df.corr(), annot = True) # Displays correlation heatmap using seaborn library

plt.rcParams['figure.figsize'] = (20,7) # Sets the size of the heatmap

plt.show() # Displays the heatmap
```



```
df
```

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	2015 Population	F
0	36	AFG	Afghanistan	Kabul	Asia	41128771.00	38972230.00	33753499.00	2
1	138	ALB	Albania	Tirana	Europe	2842321.00	2866849.00	2882481.00	
2	34	DZA	Algeria	Algiers	Africa	44903225.00	43451666.00	39543154.00	3
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	2
232	63	ZMB	Zambia	Lusaka	Africa	20017675.00	18927715.00	32066004.16	1
233	74	ZWE	Zimbabwe	Harare	Africa	16320537.00	15669666.00	14154937.00	1

234 rows × 17 columns

Next steps: Generate code with df View recommended plots

```
# First, let's create a list of population columns
population_columns = ['1970 Population', '1980 Population', '1990 Population', '2000 Population',
                      '2010 Population', '2015 Population', '2020 Population', '2022 Population']
```

```
df1 = df.groupby('Continent')[population_columns].mean().sort_values(by="2022 Population",ascending=False)
df1
```

Continent	1970 Population	1980 Population	1990 Population	2000 Population	2010 Population	2015 Population
Asia	42720942.69	39318515.39	47467614.08	76281607.92	85558713.67	89165003.64
South America	13781939.71	17270643.29	21224743.93	25146217.73	27038021.89	29509599.71
Africa	6893467.34	8721064.49	11516499.60	14813140.14	19496721.90	21980035.18
Europe	13118479.82	14283319.73	14967014.34	15058141.91	14712278.68	15368225.12
North America	7885865.15	9207334.03	10531660.62	12151739.60	13568016.28	14259596.25

Next steps: Generate code with df1 View recommended plots

```
df[df['Continent'].str.contains('Oceania')]
```

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	20 Populati
3	213	ASM	American Samoa	Pago Pago	Oceania	44273.00	46189.00	51368.
11	55	AUS	Australia	Canberra	Oceania	26177413.00	25670051.00	23820236.
44	223	COK	Cook Islands	Avarua	Oceania	17011.00	17029.00	17695.
66	162	FJI	Fiji	Suva	Oceania	929766.00	920422.00	917200.
70	183	PYF	French Polynesia	Papeete	Oceania	306279.00	301920.00	291787.
81	191	GUM	Guam	Hagåtña	Oceania	171774.00	169231.00	167978.
107	192	KIR	Kiribati	Tarawa	Oceania	131232.00	126463.00	116707.
126	215	MHL	Marshall Islands	Majuro	Oceania	41569.00	43413.00	49410.
132	194	FSM	Micronesia	Palikir	Oceania	114164.00	112106.00	109462.
142	225	NRU	Nauru	Yaren	Oceania	12668.00	12315.00	11185.
145	185	NCL	New Caledonia	Nouméa	Oceania	289950.00	286403.00	283032.
146	123	NZL	New Zealand	Wellington	Oceania	5185288.00	5061133.00	4590590.
150	232	NIU	Niue	Alofi	Oceania	1934.00	1942.00	1847.
153	210	NFK	Northern Mariana Islands	Saipan	Oceania	49551.00	49587.00	51514.
157	222	PLW	Palau	Ngerulmud	Oceania	34632250.88	17972.00	17794.
160	93	PNG	Papua New Guinea	Port Moresby	Oceania	10142619.00	9749640.00	8682174.
179	188	WSM	Samoa	Apia	Oceania	222382.00	214929.00	203571.
191	166	SLB	Solomon Islands	Honiara	Oceania	724273.00	691191.00	612660.
209	233	TKL	Tokelau	Nukunonu	Oceania	1871.00	1827.00	1454.
210	197	TON	Tonga	Nukuʻalofa	Oceania	106858.00	105254.00	106122.
216	227	TUV	Tuvalu	Funafuti	Oceania	11312.00	11069.00	10877.
225	181	VUT	Vanuatu	Port-Vila	Oceania	326740.00	311685.00	276438.
229	226	WLF	Wallis and Futuna	Mata-Utu	Oceania	11572.00	11655.00	12182.

```
df.columns
```

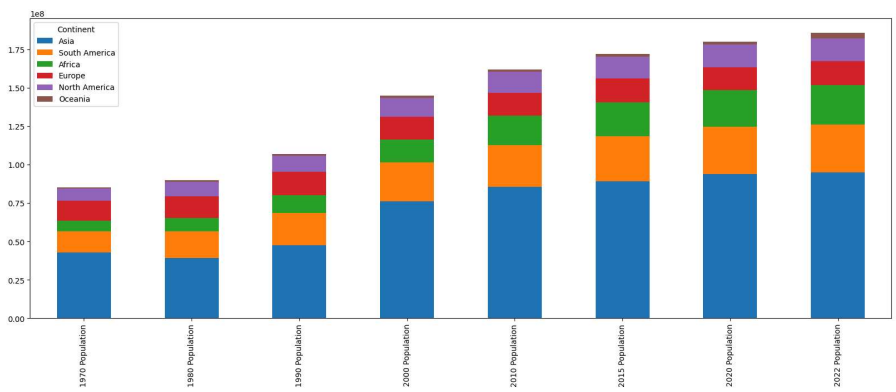
```
Index(['Rank', 'CCA3', 'Country', 'Capital', 'Continent', '2022 Population',
       '2020 Population', '2015 Population', '2010 Population',
       '2000 Population', '1990 Population', '1980 Population',
       '1970 Population', 'Area (km²)', 'Density (per km²)', 'Growth Rate',
       'World Population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

```
df2 = df1.transpose()
df2
```

Continent	Asia	South America	Africa	Europe	North America	Oceania
1970 Population	42720942.69	13781939.71	6893467.34	13118479.82	7885865.15	846968.26
1980 Population	39318515.39	17270643.29	8721064.49	14283319.73	9207334.03	996532.17
1990 Population	47467614.08	21224743.93	11516499.60	14967014.34	10531660.62	1162774.87
2000 Population	76281607.92	25146217.73	14813140.14	15058141.91	12151739.60	1357512.09
2010 Population	85558713.67	27038021.80	19496721.90	14712278.68	13568016.28	1613163.65

Next steps: Generate code with df2 View recommended plots

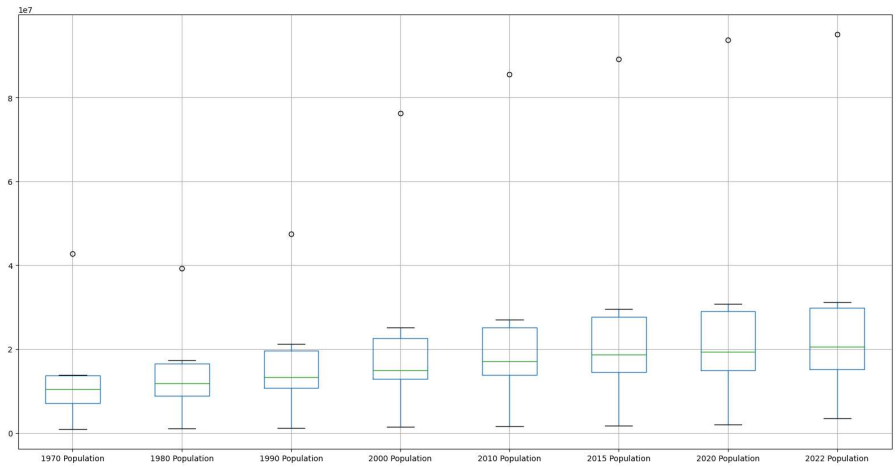
```
df2.plot(kind='bar', stacked=True)
plt.show()
```




```
df2.plot()
plt.show()
```

```
df1.boxplot(figsize=(20,10))
```

```
<Axes: >
```




```
df.boxplot(figsize=(20,10))
```

```
<Axes: >
```