

# Dataset Overview and Making a Pivot Table

October 3, 2024

```
[2]: import pandas as pd
```

```
[5]: df_population_raw = pd.read_csv('population_total.csv')
```

```
[6]: df_population_raw
```

```
[6]:
```

	country	year	population
0	China	2020.0	1.439324e+09
1	China	2019.0	1.433784e+09
2	China	2018.0	1.427648e+09
3	China	2017.0	1.421022e+09
4	China	2016.0	1.414049e+09
...	...	...	...
4180	United States	1965.0	1.997337e+08
4181	United States	1960.0	1.867206e+08
4182	United States	1955.0	1.716853e+08
4183	India	1960.0	4.505477e+08
4184	India	1955.0	4.098806e+08

[4185 rows x 3 columns]

```
[8]: # dropping null values
df_population_raw.dropna(inplace=True)
```

```
[10]: # making a pivot table
df_pivot = df_population_raw.pivot(index='year', columns='country',
↪values='population')
```

```
[ ]: #selecting some countries
```

```
[12]: df_pivot[['United States', 'India', 'China', 'Indonesia', 'Brazil']]
```

```
[13]: df_pivot
```

```
[13]:
```

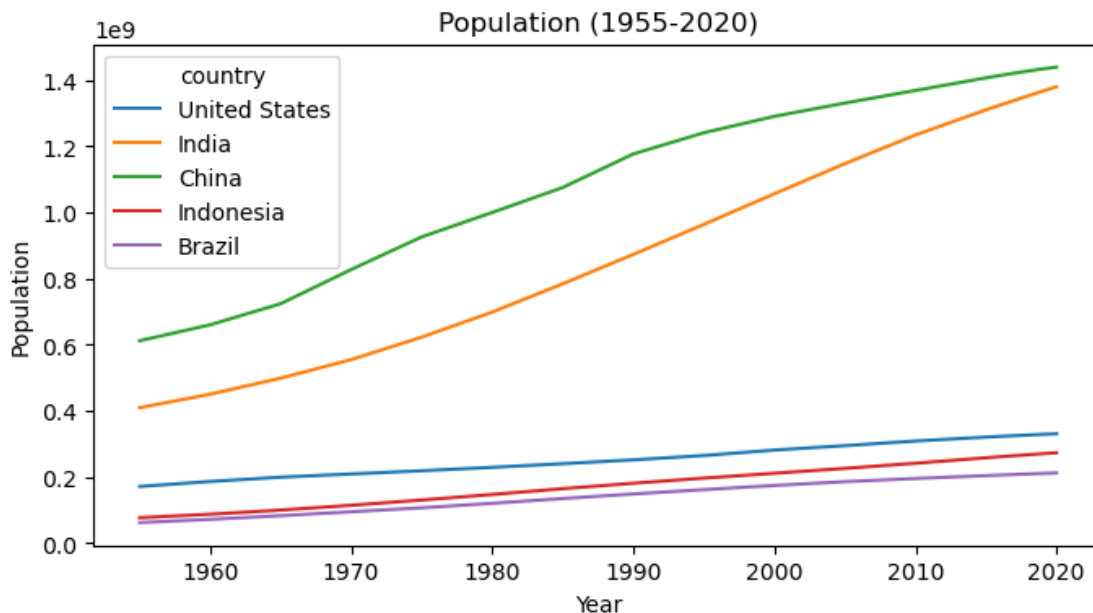
country	United States	India	China	Indonesia	Brazil
year					
1955.0	171685336.0	4.098806e+08	6.122416e+08	77273425.0	62533919.0
1960.0	186720571.0	4.505477e+08	6.604081e+08	87751068.0	72179226.0

1965.0	199733676.0	4.991233e+08	7.242190e+08	100267062.0	83373530.0
1970.0	209513341.0	5.551898e+08	8.276014e+08	114793178.0	95113265.0
1975.0	219081251.0	6.231029e+08	9.262409e+08	130680727.0	107216205.0
1980.0	229476354.0	6.989528e+08	1.000089e+09	147447836.0	120694009.0
1985.0	240499825.0	7.843600e+08	1.075589e+09	164982451.0	135274080.0
1990.0	252120309.0	8.732778e+08	1.176884e+09	181413402.0	149003223.0
1995.0	265163745.0	9.639226e+08	1.240921e+09	196934260.0	162019896.0
2000.0	281710909.0	1.056576e+09	1.290551e+09	211513823.0	174790340.0
2005.0	294993511.0	1.147610e+09	1.330776e+09	226289470.0	186127103.0
2010.0	309011475.0	1.234281e+09	1.368811e+09	241834215.0	195713635.0
2015.0	320878310.0	1.310152e+09	1.406848e+09	258383256.0	204471769.0
2016.0	323015995.0	1.324517e+09	1.414049e+09	261556381.0	206163053.0
2017.0	325084756.0	1.338677e+09	1.421022e+09	264650963.0	207833823.0
2018.0	327096265.0	1.352642e+09	1.427648e+09	267670543.0	209469323.0
2019.0	329064917.0	1.366418e+09	1.433784e+09	270625568.0	211049527.0
2020.0	331002651.0	1.380004e+09	1.439324e+09	273523615.0	212559417.0

```
[62]: # importing matplotlib
import matplotlib.pyplot as plt
```

```
[63]: # Data Visualisation with Pandas(Lineplot)
df_pivot.plot(kind='line', xlabel='Year', ylabel='Population',
               title='Population (1955-2020)',
               figsize=(8,4))

# save plot
plt.savefig('kunle.png')
plt.show()
```



```
[65]: df_pivot.to_excel('pivot_table.xlsx')
```

```
[2]: import pandas as pd
```

```
[3]: df_population_raw = pd.read_csv('population_total.csv')
```

```
[5]: df_population_raw
```

```
[5]:
```

	country	year	population
0	China	2020.0	1.439324e+09
1	China	2019.0	1.433784e+09
2	China	2018.0	1.427648e+09
3	China	2017.0	1.421022e+09
4	China	2016.0	1.414049e+09
...	...	...	...
4180	United States	1965.0	1.997337e+08
4181	United States	1960.0	1.867206e+08
4182	United States	1955.0	1.716853e+08
4183	India	1960.0	4.505477e+08
4184	India	1955.0	4.098806e+08

[4185 rows x 3 columns]

```
[12]: df_population_raw.dropna(inplace=True)
```

```
[14]: df_pivot = df_population_raw.pivot(index='year', columns='country',  
↪values='population')
```

```
[15]: df_pivot = df_pivot[['United States', 'India', 'China', 'Indonesia', 'Brazil']]
```

```
[16]: df_pivot
```

```
[16]:
```

country	United States	India	China	Indonesia	Brazil
year					
1955.0	171685336.0	4.098806e+08	6.122416e+08	77273425.0	62533919.0
1960.0	186720571.0	4.505477e+08	6.604081e+08	87751068.0	72179226.0
1965.0	199733676.0	4.991233e+08	7.242190e+08	100267062.0	83373530.0
1970.0	209513341.0	5.551898e+08	8.276014e+08	114793178.0	95113265.0
1975.0	219081251.0	6.231029e+08	9.262409e+08	130680727.0	107216205.0
1980.0	229476354.0	6.989528e+08	1.000089e+09	147447836.0	120694009.0
1985.0	240499825.0	7.843600e+08	1.075589e+09	164982451.0	135274080.0
1990.0	252120309.0	8.732778e+08	1.176884e+09	181413402.0	149003223.0
1995.0	265163745.0	9.639226e+08	1.240921e+09	196934260.0	162019896.0
2000.0	281710909.0	1.056576e+09	1.290551e+09	211513823.0	174790340.0
2005.0	294993511.0	1.147610e+09	1.330776e+09	226289470.0	186127103.0

2010.0	309011475.0	1.234281e+09	1.368811e+09	241834215.0	195713635.0
2015.0	320878310.0	1.310152e+09	1.406848e+09	258383256.0	204471769.0
2016.0	323015995.0	1.324517e+09	1.414049e+09	261556381.0	206163053.0
2017.0	325084756.0	1.338677e+09	1.421022e+09	264650963.0	207833823.0
2018.0	327096265.0	1.352642e+09	1.427648e+09	267670543.0	209469323.0
2019.0	329064917.0	1.366418e+09	1.433784e+09	270625568.0	211049527.0
2020.0	331002651.0	1.380004e+09	1.439324e+09	273523615.0	212559417.0

```
[17]: # Selecting to show index
df_pivot.index
```

```
[17]: Index([1955.0, 1960.0, 1965.0, 1970.0, 1975.0, 1980.0, 1985.0, 1990.0, 1995.0,
        2000.0, 2005.0, 2010.0, 2015.0, 2016.0, 2017.0, 2018.0, 2019.0, 2020.0],
        dtype='float64', name='year')
```

```
[19]: ## selecting index year 2020 from DF
df_pivot_2020 = df_pivot[df_pivot.index.isin([2020])]
```

```
[20]: # showing new DF
df_pivot_2020
```

```
[20]: country  United States      India      China  Indonesia      Brazil
year
2020.0      331002651.0  1.380004e+09  1.439324e+09  273523615.0  212559417.0
```

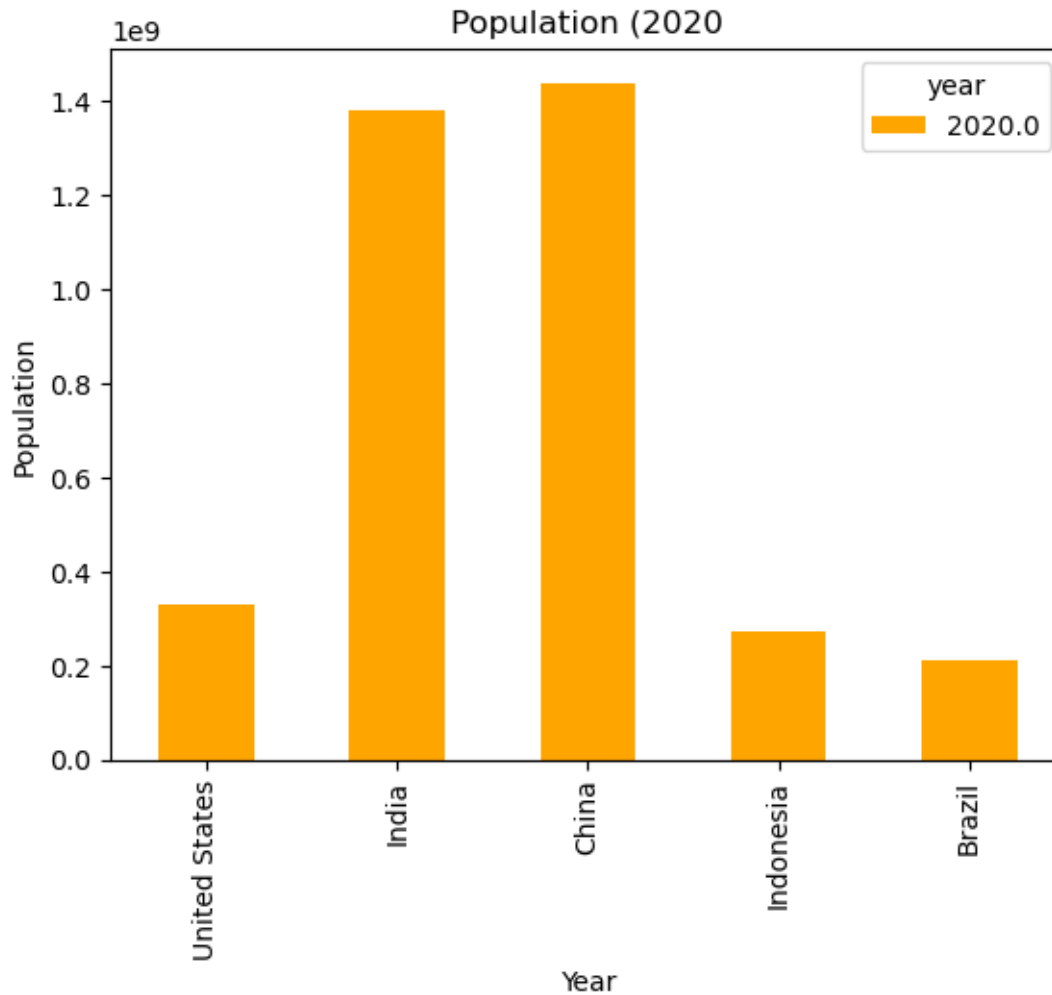
```
[23]: # transpose DF from rows to columns or viceversa
df_pivot_2020.T
```

```
[23]: year      2020.0
country
United States  3.310027e+08
India          1.380004e+09
China          1.439324e+09
Indonesia      2.735236e+08
Brazil         2.125594e+08
```

```
[24]: df_pivot_2020 = df_pivot_2020.T
```

```
[30]: # making barplot
df_pivot_2020.plot(kind='bar', color='orange',
                  xlabel='Year', ylabel='Population',
                  title='Population (2020)')
```

```
[30]: <Axes: title={'center': 'Population (2020)', xlabel='Year', ylabel='Population'>
```



```
[31]: # barplot grouped by "n" variables( selecting some years)

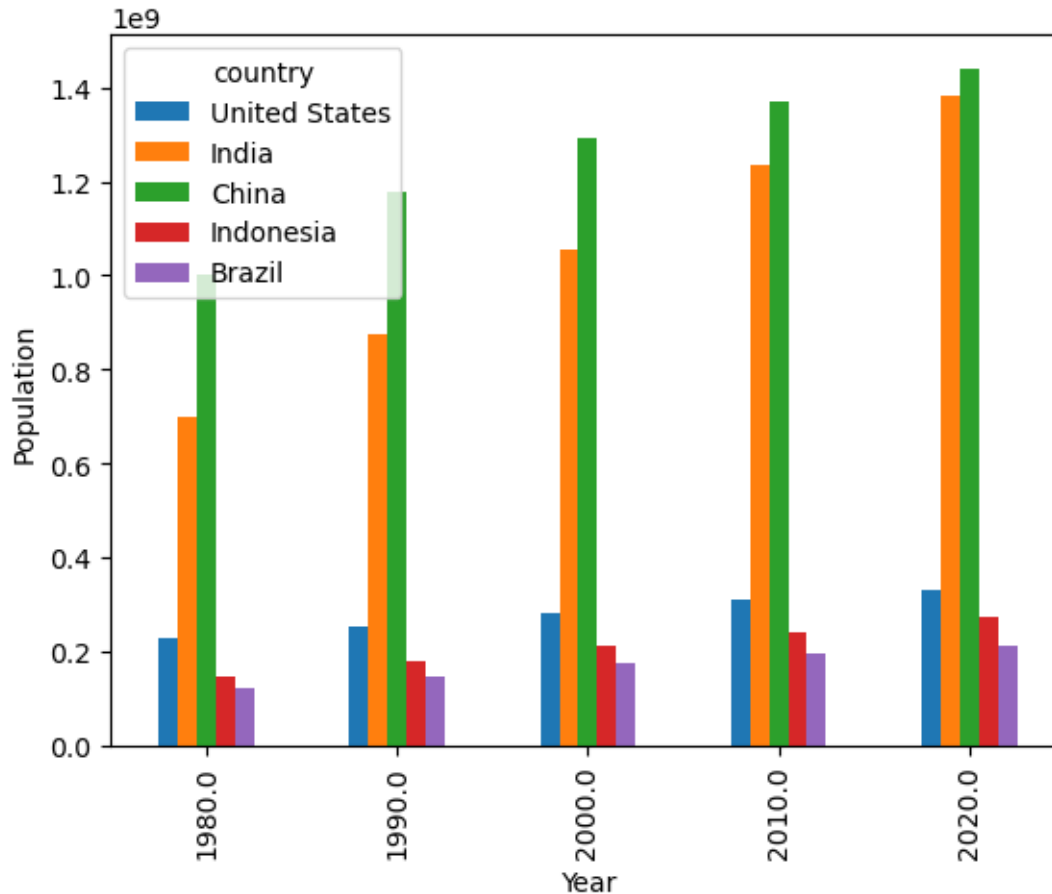
df_pivot_years = df_pivot_2020 = df_pivot[df_pivot.index.isin([1980, 1990, 2000, 2010, 2020])]
```

```
[32]: df_pivot_years
```

```
[32]: country  United States      India      China  Indonesia      Brazil
year
1980.0      229476354.0  6.989528e+08  1.000089e+09  147447836.0  120694009.0
1990.0      252120309.0  8.732778e+08  1.176884e+09  181413402.0  149003223.0
2000.0      281710909.0  1.056576e+09  1.290551e+09  211513823.0  174790340.0
2010.0      309011475.0  1.234281e+09  1.368811e+09  241834215.0  195713635.0
2020.0      331002651.0  1.380004e+09  1.439324e+09  273523615.0  212559417.0
```

```
[41]: # making a grouped barplot from the years 1980, 1990, 2000, 2010, and 2020.
df_pivot_years.plot(kind='bar', xlabel='Year', ylabel='Population')
```

```
[41]: <Axes: xlabel='Year', ylabel='Population'>
```



```
[46]: df_pivot_2020 == df_pivot[df_pivot.index.isin([2020])]
```

```
[46]: country  United States  India  China  Indonesia  Brazil
year
2020.0          True    True   True         True    True
```

```
[43]: df_pivot_2020
```

```
[43]: country  United States          India          China  Indonesia  Brazil
year
2020.0    331002651.0  1.380004e+09  1.439324e+09  273523615.0  212559417.0
```

```
[51]: df_pivot_2020.T.rename(columns={2020:'2020'}, inplace=True)
```

```
[53]: df_pivot_2020
```

```
[53]: country  United States      India      China  Indonesia      Brazil
year
2020.0      331002651.0  1.380004e+09  1.439324e+09  273523615.0  212559417.0
```

```
[55]: df_pivot_2020 = df_pivot_2020.T
```

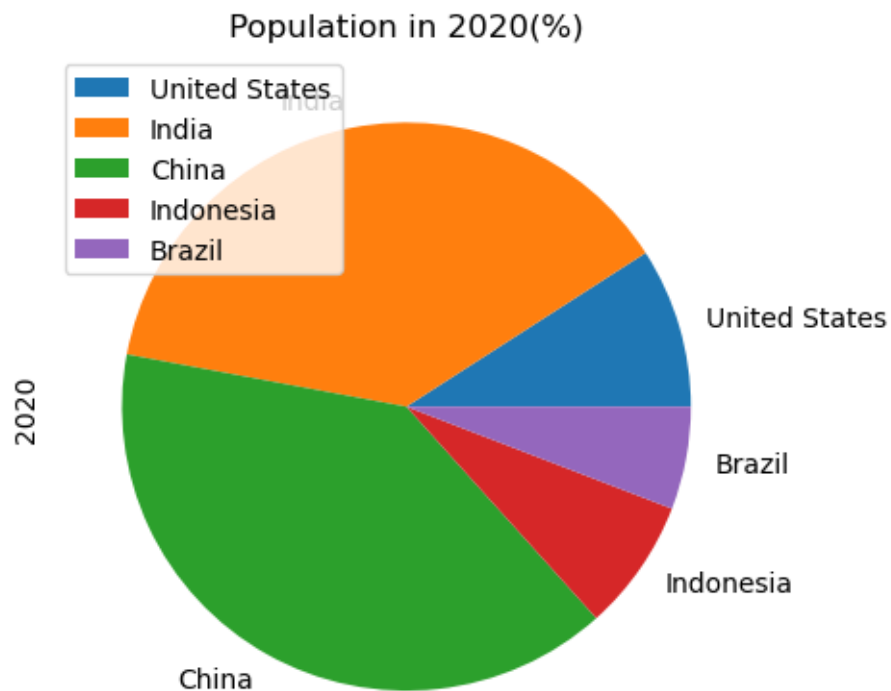
```
[56]: df_pivot_2020
```

```
[56]: year          2020.0
country
United States  3.310027e+08
India          1.380004e+09
China          1.439324e+09
Indonesia      2.735236e+08
Brazil         2.125594e+08
```

```
[58]: # Piechart( changing column name to make a piechart)
df_pivot_2020.rename(columns={2020:'2020'}, inplace=True)
```

```
[61]: df_pivot_2020.plot(kind='pie', y='2020' , title='Population in 2020(%))')
```

```
[61]: <Axes: title={'center': 'Population in 2020(%))'}, ylabel='2020'>
```



[ ]:	
[ ]:	
[ ]:	
[ ]:	
[ ]:	
[ ]:	
[ ]:	
[ ]:	
[ ]:	