pivots&graphs

April 19, 2024

[83]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

[7]:
```
#read and show data
df_gdp = pd.read_csv('/Users/xolo/Desktop/Python Data analysis/2.Python for␣
 ↪Excel Users/gdp.csv',encoding='latin1')
df_gdp
```

[7]:

	unid	wbid	country	year	SES	gdppc	yrseduc	popshare
0	4	AFG	Afghanistan	1970	3.474212	709.00000	NaN	0.003097
1	4	AFG	Afghanistan	1920	26.968016	731.75677	NaN	0.003245
2	4	AFG	Afghanistan	1990	1.269530	604.00000	NaN	0.002347
3	4	AFG	Afghanistan	1960	15.763076	739.00000	NaN	0.003039
4	4	AFG	Afghanistan	2000	2.061114	565.00000	NaN	0.003309
…	…	…	…	…	…	…	…	…
2081	716	ZWE	Zimbabwe	1940	52.746567	813.00000	NaN	0.001113
2082	716	ZWE	Zimbabwe	2010	27.091389	1388.97300	NaN	0.002074
2083	716	ZWE	Zimbabwe	1990	59.110970	2526.07230	NaN	0.002039
2084	716	ZWE	Zimbabwe	1900	54.722645	601.00000	NaN	0.000996
2085	716	ZWE	Zimbabwe	1960	53.811077	938.00000	NaN	0.001268

[2086 rows x 8 columns]

[10]:
```
df_gdp['country'].value_counts()
```

[10]:
```
country
Afghanistan    14
Mozambique     14
Mauritius      14
Malawi         14
Malaysia       14
               ..
Gambia         14
Greece         14
Guatemala      14
Guyana         14
Zimbabwe       14
Name: count, Length: 149, dtype: int64
```

[13]:
```
#Country gdp across the years
df_gdp.pivot(index='country',columns = 'year', values= 'gdppc').round(2)
```

[13]:
```
year            1880     1890     1900     1910     1920     1930     1940  \
country
Afghanistan   585.47   635.93   686.40   736.86   731.76   702.84   673.92
Albania       522.00   598.00   685.00   780.00   861.31   929.57   965.29
Algeria       819.19   923.37  1027.56  1131.74  1201.22  1255.81  1310.41
Angola        533.67   567.33   601.00   628.69   682.63   747.81   813.00
Argentina    1731.50  2152.00  2756.00  3822.00  3473.00  4080.00  4161.00
…                …        …        …        …        …        …        …
Venezuela     653.00   737.00   821.00   886.00  1173.00  3444.00  4045.00
Vietnam       556.63   608.26   659.88   711.51   713.95   695.30   676.65
Yemen         811.77   881.53   951.30  1021.07  1017.22   981.81   946.41
Zambia        533.67   567.33   601.00   628.69   682.63   747.81   813.00
Zimbabwe      533.67   567.33   601.00   628.69   682.63   747.81   813.00

year            1950    1960     1970     1980      1990      2000      2010
country
Afghanistan    645.0   739.0    709.0    690.0    604.00    565.00   1662.80
Albania       1001.0  1451.0   2004.0   2347.0   4557.41   5470.02   9927.18
Algeria       1365.0  2088.0   2249.0   3152.0  10238.88  10211.10  12898.30
Angola        1052.0  1253.0   1768.0    961.0   4806.43   3831.00   6492.18
Argentina     4987.0  5559.0   7302.0   8206.0  10833.50  14924.37  18794.27
…                …       …        …        …         …         …         …
Venezuela     7462.0  9646.0  10672.0  10139.0  14450.68  14417.10  16563.40
Vietnam        658.0   799.0    735.0    757.0   1501.14   2649.72   4486.26
Yemen          911.0   964.0   1230.0   2290.0   3353.79   3911.22   4481.55
Zambia         661.0   960.0   1073.0    911.0   2308.60   2092.26   3263.39
Zimbabwe       701.0   938.0   1282.0   1295.0   2526.07   2521.35   1388.97

[149 rows x 14 columns]
```

1 Pivot Tables

[15]:
```
#read excel file csv
df_supersales = pd.read_excel('/Users/xolo/Desktop/Python Data analysis/2.
 ↪Python for Excel Users/supermarket_sales.xlsx')
df_supersales
```

[15]:
```
      Invoice ID Branch       City Customer type  Gender  \
0    750-67-8428      A     Yangon        Member  Female
1    226-31-3081      C  Naypyitaw        Normal  Female
2    631-41-3108      A     Yangon        Normal    Male
3    123-19-1176      A     Yangon        Member    Male
4    373-73-7910      A     Yangon        Normal    Male
..           …    …        …             …      …
```

```
995  233-67-5758      C  Naypyitaw        Normal    Male
996  303-96-2227      B   Mandalay        Normal  Female
997  727-02-1313      A     Yangon        Member    Male
998  347-56-2442      A     Yangon        Normal    Male
999  849-09-3807      A     Yangon        Member  Female

              Product line  Unit price  Quantity   Tax 5%      Total  \
0         Health and beauty       74.69         7  26.1415   548.9715
1    Electronic accessories       15.28         5   3.8200    80.2200
2        Home and lifestyle       46.33         7  16.2155   340.5255
3         Health and beauty       58.22         8  23.2880   489.0480
4         Sports and travel       86.31         7  30.2085   634.3785
..                      …         …       …      …         …
995       Health and beauty       40.35         1   2.0175    42.3675
996      Home and lifestyle       97.38        10  48.6900  1022.4900
997      Food and beverages       31.84         1   1.5920    33.4320
998      Home and lifestyle       65.82         1   3.2910    69.1110
999     Fashion accessories       88.34         7  30.9190   649.2990

          Date      Time      Payment    cogs  gross margin percentage  \
0   2019-01-05  13:08:00      Ewallet  522.83                 4.761905
1   2019-03-08  10:29:00         Cash   76.40                 4.761905
2   2019-03-03  13:23:00  Credit card  324.31                 4.761905
3   2019-01-27  20:33:00      Ewallet  465.76                 4.761905
4   2019-02-08  10:37:00      Ewallet  604.17                 4.761905
..         …       …           …     …                      …
995 2019-01-29  13:46:00      Ewallet   40.35                 4.761905
996 2019-03-02  17:16:00      Ewallet  973.80                 4.761905
997 2019-02-09  13:22:00         Cash   31.84                 4.761905
998 2019-02-22  15:33:00         Cash   65.82                 4.761905
999 2019-02-18  13:28:00         Cash  618.38                 4.761905

     gross income  Rating
0         26.1415     9.1
1          3.8200     9.6
2         16.2155     7.4
3         23.2880     8.4
4         30.2085     5.3
..            …     …
995        2.0175     6.2
996       48.6900     4.4
997        1.5920     7.7
998        3.2910     4.1
999       30.9190     6.6

[1000 rows x 17 columns]
```

[23]:
```
df_supersales.dtypes
```

[23]:
```
Invoice ID                     object
Branch                         object
City                           object
Customer type                  object
Gender                         object
Product line                   object
Unit price                    float64
Quantity                        int64
Tax 5%                        float64
Total                         float64
Date                   datetime64[ns]
Time                           object
Payment                        object
cogs                          float64
gross margin percentage       float64
gross income                  float64
Rating                        float64
dtype: object
```

[28]:
```
#use pivot_tables function and aggragate data
df_supersales_nodate= df_supersales.drop(columns=['Date','Time'])
#df_supersales_nodate
df_supersales_nodate.pivot_table(index = "Gender", values=["Quantity","Total"],␣
 ↪aggfunc = "sum")
```

[28]:
```
        Quantity       Total
Gender
Female      2869  167882.925
Male        2641  155083.824
```

[31]:
```
#pivot to how much male and female spent in each category
df_supersales_nodate.pivot_table(index = "Gender",columns= "Product line",␣
 ↪values=["Quantity","Total"], aggfunc = "sum").round(2)
```

[31]:
```
                           Quantity                                         \
Product line Electronic accessories Fashion accessories Food and beverages
Gender
Female                           488                  530                514
Male                             483                  372                438

                                                                        \
Product line Health and beauty Home and lifestyle Sports and travel
Gender
Female                     343                 498                496
Male                       511                 413                424
```

```
                                Total                                         \
Product line Electronic accessories Fashion accessories Food and beverages
Gender
Female                       27102.02             30437.4           33170.92
Male                         27235.51             23868.5           22973.93


Product line Health and beauty Home and lifestyle Sports and travel
Gender
Female                 18560.99           30036.88          28574.72
Male                   30632.75           23825.04          26548.11
```

2 Data Visualisation

[34]:
```
#read csv file
df_population_raw = pd.read_csv('/Users/xolo/Desktop/Python Data analysis/2.
 ↪Python for Excel Users/population_total.csv')
df_population_raw
```

[34]:
```
            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]
```

2.1 Maving a Pivot Table

[38]:
```
#dropping null values
df_population_raw = df_population_raw.dropna()
df_population_raw
```

[38]:
```
            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

[4178 rows x 3 columns]
```

[49]:
```
# List of countries you want to include
countries_to_include = ['China', 'Zambia', 'India', 'Andorra', 'Yemen',␣
 ↪'Algeria']

# Filter the DataFrame to include only the specified countries
df_filtered = df_population_raw[df_population_raw['country'].
 ↪isin(countries_to_include)]

# Pivot the filtered DataFrame
df_pop_pivot = df_filtered.pivot(index='country', columns='year',␣
 ↪values='population')
```

[50]:
```
#select columns
df_pop_pivot
```

[50]:
```
year          1955.0       1960.0       1965.0       1970.0       1975.0  \
country
Algeria     9774283.0   11057863.0   12550885.0   14464985.0   16607707.0
Andorra        9232.0      13411.0      18549.0      24276.0      30705.0
China     612241554.0  660408056.0  724218968.0  827601394.0  926240885.0
India     409880595.0  450547679.0  499123324.0  555189792.0  623102897.0
Yemen       4965574.0    5315355.0    5727751.0    6193384.0    6784695.0
Zambia      2644976.0    3070776.0    3570464.0    4179067.0    4943283.0

year           1980.0        1985.0        1990.0        1995.0        2000.0  \
country
Algeria  1.922166e+07  2.243150e+07  2.575887e+07  2.875778e+07  3.104224e+07
Andorra  3.606700e+04  4.460000e+04  5.450900e+04  6.385000e+04  6.539000e+04
China    1.000089e+09  1.075589e+09  1.176884e+09  1.240921e+09  1.290551e+09
India    6.989528e+08  7.843600e+08  8.732778e+08  9.639226e+08  1.056576e+09
Yemen    7.941898e+06  9.572175e+06  1.170999e+07  1.491332e+07  1.740907e+07
Zambia   5.851825e+06  6.923149e+06  8.036845e+06  9.096607e+06  1.041594e+07

year           2005.0        2010.0        2015.0        2016.0        2017.0  \
country
Algeria  3.314972e+07  3.597746e+07  3.972802e+07  4.055139e+07  4.138919e+07
Andorra  7.886700e+04  8.444900e+04  7.801100e+04  7.729700e+04  7.700100e+04
```

```
China    1.330776e+09  1.368811e+09  1.406848e+09  1.414049e+09  1.421022e+09
India    1.147610e+09  1.234281e+09  1.310152e+09  1.324517e+09  1.338677e+09
Yemen    2.010741e+07  2.315486e+07  2.649789e+07  2.716821e+07  2.783482e+07
Zambia   1.185625e+07  1.360598e+07  1.587936e+07  1.636346e+07  1.685360e+07

year           2018.0        2019.0        2020.0
country
Algeria  4.222841e+07  4.305305e+07  4.385104e+07
Andorra  7.700600e+04  7.714200e+04           NaN
China    1.427648e+09  1.433784e+09  1.439324e+09
India    1.352642e+09  1.366418e+09  1.380004e+09
Yemen    2.849868e+07  2.916192e+07  2.982596e+07
Zambia   1.735171e+07  1.786103e+07  1.838396e+07
```

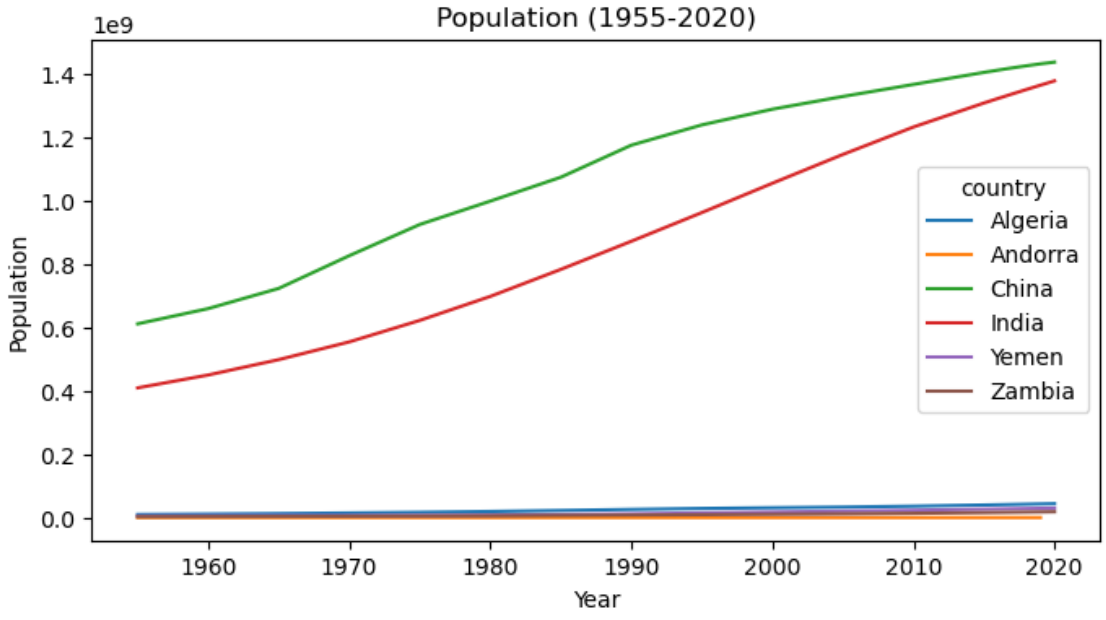
[53]:
```
df_pop_transposed = df_pop_pivot.transpose()
df_pop_transposed
```

[53]:
```
country     Algeria  Andorra         China         India       Yemen  \
year
1955.0    9774283.0   9232.0  6.122416e+08  4.098806e+08   4965574.0
1960.0   11057863.0  13411.0  6.604081e+08  4.505477e+08   5315355.0
1965.0   12550885.0  18549.0  7.242190e+08  4.991233e+08   5727751.0
1970.0   14464985.0  24276.0  8.276014e+08  5.551898e+08   6193384.0
1975.0   16607707.0  30705.0  9.262409e+08  6.231029e+08   6784695.0
1980.0   19221665.0  36067.0  1.000089e+09  6.989528e+08   7941898.0
1985.0   22431502.0  44600.0  1.075589e+09  7.843600e+08   9572175.0
1990.0   25758869.0  54509.0  1.176884e+09  8.732778e+08  11709993.0
1995.0   28757785.0  63850.0  1.240921e+09  9.639226e+08  14913315.0
2000.0   31042235.0  65390.0  1.290551e+09  1.056576e+09  17409072.0
2005.0   33149724.0  78867.0  1.330776e+09  1.147610e+09  20107409.0
2010.0   35977455.0  84449.0  1.368811e+09  1.234281e+09  23154855.0
2015.0   39728025.0  78011.0  1.406848e+09  1.310152e+09  26497889.0
2016.0   40551392.0  77297.0  1.414049e+09  1.324517e+09  27168208.0
2017.0   41389189.0  77001.0  1.421022e+09  1.338677e+09  27834819.0
2018.0   42228408.0  77006.0  1.427648e+09  1.352642e+09  28498683.0
2019.0   43053054.0  77142.0  1.433784e+09  1.366418e+09  29161922.0
2020.0   43851044.0      NaN  1.439324e+09  1.380004e+09  29825964.0

country      Zambia
year
1955.0    2644976.0
1960.0    3070776.0
1965.0    3570464.0
1970.0    4179067.0
1975.0    4943283.0
1980.0    5851825.0
1985.0    6923149.0
```

```
1990.0    8036845.0
1995.0    9096607.0
2000.0   10415944.0
2005.0   11856247.0
2010.0   13605984.0
2015.0   15879361.0
2016.0   16363458.0
2017.0   16853599.0
2018.0   17351708.0
2019.0   17861030.0
2020.0   18383955.0
```

2.1.1 1. Line Plot

[84]:
```
#Line plot
df_pop_transposed .plot(kind = 'line', xlabel = 'Year', ylabel = 'Population',␣
 ↪title = 'Population (1955-2020)', figsize= (8,4))

#Save plot
plt.savefig('line_plot.png')

#show plot
plt.show()
```

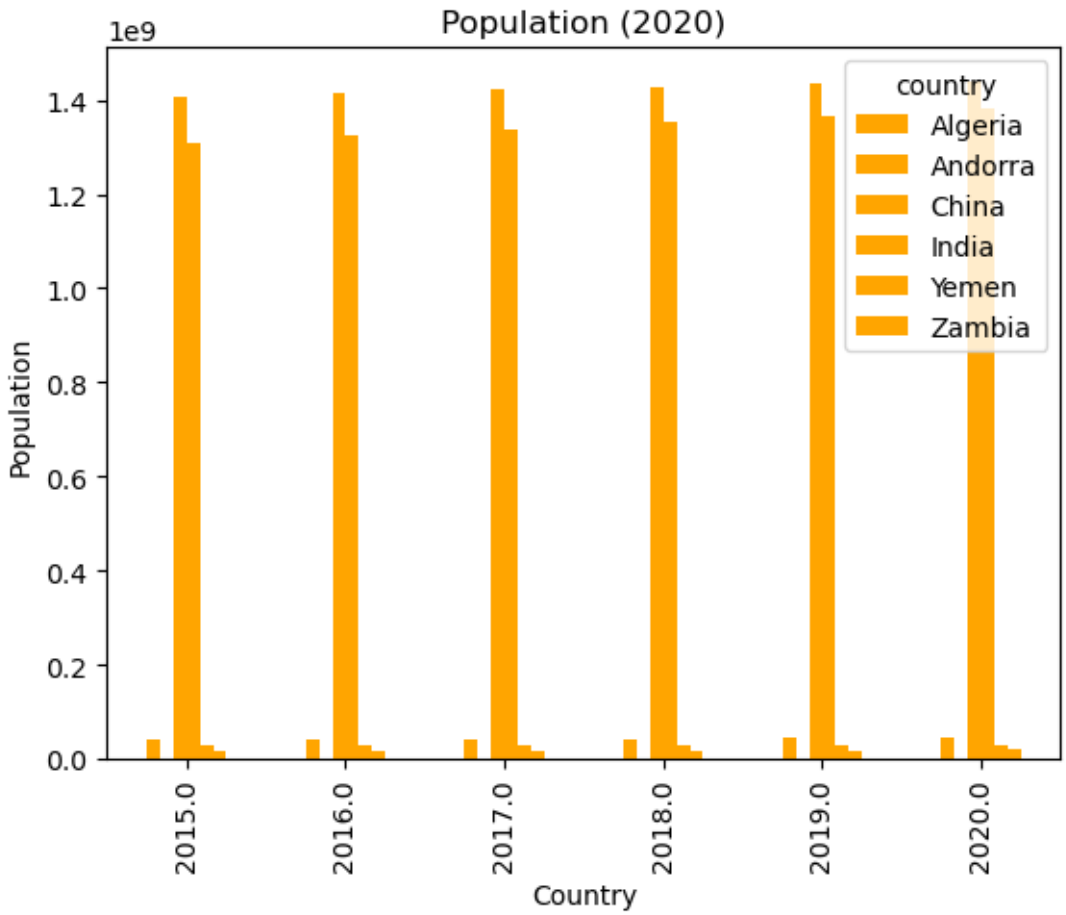


2.1.2 2.1 Bar plot

[66]:
```
#selecting a year (2020)
df_pop_2020 = df_pop_transposed[df_pop_transposed.index.isin([2020])]
df_pop_2020 = df_pop_2020.T
df_pop_2020
```

[66]:
```
year           2020.0
country
Algeria  4.385104e+07
Andorra           NaN
China    1.439324e+09
India    1.380004e+09
Yemen    2.982596e+07
Zambia   1.838396e+07
```

[85]:
```
#make a barplot
df_pop_2020.plot(kind = 'bar', xlabel = 'Country', ylabel = 'Population', title␣
 ↪= 'Population (2020)',color = 'orange')

#Save plot
plt.savefig('bar_plot.png')

#show plot
plt.show()
```



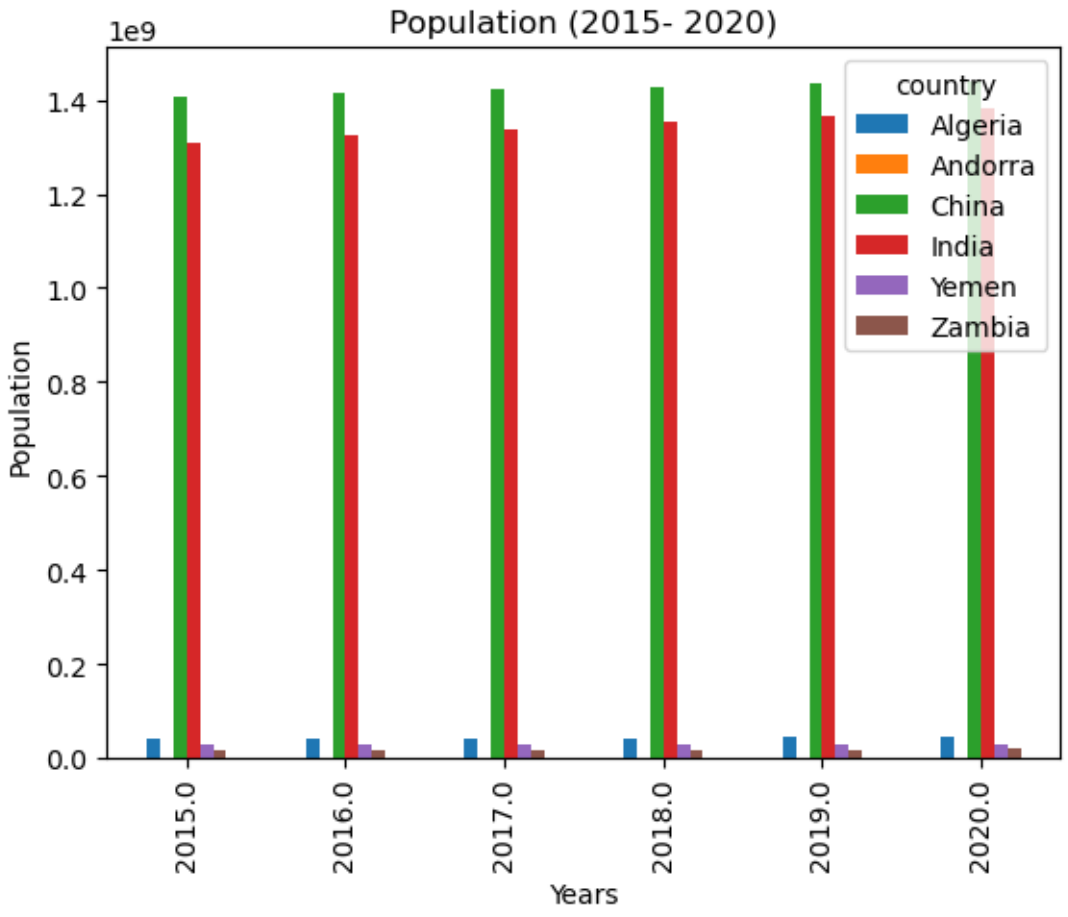
2.1.3 2.2 Bar Plot grouped by n variables

[70]:
```
#selecting a few years
df_pop_5yr = df_pop_transposed[df_pop_transposed.index.
 ↪isin([2015,2016,2017,2018,2019,2020])]
df_pop_5yr
```

[70]:
```
country     Algeria  Andorra         China         India       Yemen  \
year
2015.0   39728025.0  78011.0  1.406848e+09  1.310152e+09  26497889.0
2016.0   40551392.0  77297.0  1.414049e+09  1.324517e+09  27168208.0
2017.0   41389189.0  77001.0  1.421022e+09  1.338677e+09  27834819.0
2018.0   42228408.0  77006.0  1.427648e+09  1.352642e+09  28498683.0
2019.0   43053054.0  77142.0  1.433784e+09  1.366418e+09  29161922.0
2020.0   43851044.0      NaN  1.439324e+09  1.380004e+09  29825964.0

country      Zambia
year
```

```
2015.0   15879361.0
2016.0   16363458.0
2017.0   16853599.0
2018.0   17351708.0
2019.0   17861030.0
2020.0   18383955.0
```

[86]:
```
#making grouped barplot
df_pop_5yr.plot(kind = 'bar',xlabel = 'Years', ylabel = 'Population', title =␣
 ↪'Population (2015- 2020)')

#Save plot
plt.savefig('group_barplot.png')

#show plot
plt.show()
```

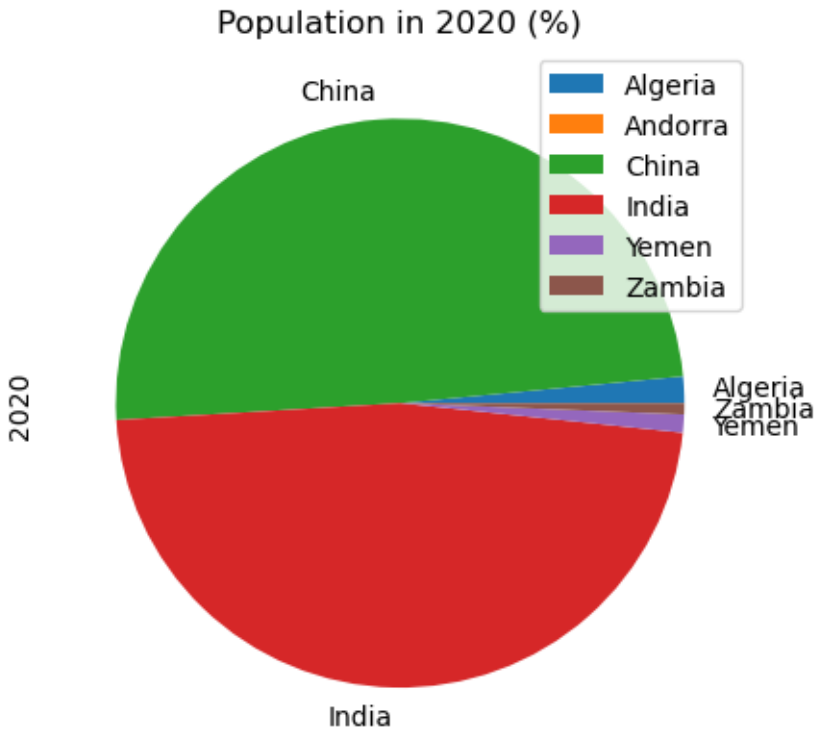


2.1.4 3.Pie Chart

[79]:
```
#changing column name
df_popu_2020 =df_pop_transposed[df_pop_transposed.index.isin([2020])]
df_popu_2020 = df_popu_2020.T
df_popu_n2020 =df_popu_2020.rename(columns= {2020:'2020'})
df_popu_n2020
```

[79]:
```
year             2020
country
Algeria  4.385104e+07
Andorra           NaN
China    1.439324e+09
India    1.380004e+09
Yemen    2.982596e+07
Zambia   1.838396e+07
```

[87]:
```
df_popu_n2020.plot(kind = 'pie',y ='2020', title = 'Population in 2020 (%)')

#Save plot
plt.savefig('pie_chart.png')

#show plot
plt.show()
```



[88]:
```
df_popu_2020.to_excel('country population 2020.xlsx')
```

[ ]:

[ ]: