AnimatedPopPyramid

February 12, 2021

1 POPULATION DATAFRAME FUNCTION

1.0.1 return a pandas df indexed by (region or country) & year – Double Indexed

1.0.2 columns will be counts of people in Different Age/Sex groups

The World Bank maintains a large set of "World Development Indicators" (WDI), including information on population.

- API for WDI is available at https://datahelpdesk.worldbank.org/knowledgebase/articles/889392-about-the-indicators-api-documentation
- A `python` module that uses the API is `wbdata`, written by Oliver Sherouse.
- Available at http://github.com/OliverSherouse/wbdata.
- Documented at https://wbdata.readthedocs.io.

We walk through the process of getting data from the WDI into a `pandas` DataFrame.

The `wbdata` module has several key functions we'll want to use:

- **`search_countries()`:** Returns code for different countries or regions.
- **`get_source()`:** Gives list of different data sources that can be accessed using the module; returns a numeric key;
- **`get_indicator()`:** Given a source, this returns a list of available variables (indicators).
- **`get_dataframe()`:** Given a source and a list of indicators, this returns a dataframe populated with the requested data for whatever

Begin by importing the module:

[1]:
```
## If import fails with "ModuleNotFoundError"
## uncomment below & try again
!pip install wbdata

import wbdata
```

```
Requirement already satisfied: wbdata in /opt/conda/lib/python3.8/site-packages
(0.3.0)
Requirement already satisfied: appdirs<2.0,>=1.4 in
/opt/conda/lib/python3.8/site-packages (from wbdata) (1.4.4)
Requirement already satisfied: tabulate>=0.8.5 in /opt/conda/lib/python3.8/site-
packages (from wbdata) (0.8.7)
```

```
Requirement already satisfied: requests>=2.0 in /opt/conda/lib/python3.8/site-
packages (from wbdata) (2.25.1)
Requirement already satisfied: decorator>=4.0 in /opt/conda/lib/python3.8/site-
packages (from wbdata) (4.4.2)
Requirement already satisfied: chardet<5,>=3.0.2 in
/opt/conda/lib/python3.8/site-packages (from requests>=2.0->wbdata) (3.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.8/site-packages (from requests>=2.0->wbdata) (1.25.7)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.8/site-
packages (from requests>=2.0->wbdata) (2.8)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.8/site-packages (from requests>=2.0->wbdata) (2019.11.28)
```

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

[3]:
```
# Give variable for clarity
variable_labels = {"SP.POP.TOTL":"Total Population",
                   "SP.POP.TOTL.FE.IN":"Female Population",
                   "SP.POP.TOTL.MA.IN":"Male Population",

                   "SP.POP.0014.FE.IN":"Females (0-14)",
                   "SP.POP.0014.MA.IN":"Males (0-14)",

                   "SP.POP.1564.FE.IN":"Females (15-64)",
                   "SP.POP.1564.MA.IN":"Males (15-64)",

                   "SP.POP.65UP.FE.IN":"Females (65+)",
                   "SP.POP.65UP.MA.IN":"Males (65+)",

                   "SP.POP.0004.FE":"Females (0-4)",
                   "SP.POP.0004.MA":"Males (0-4)",

                   "SP.POP.0509.FE":"Females (5-9)",
                   "SP.POP.0509.MA":"Males (5-9)",

                   "SP.POP.1014.FE":"Females (10-14)",
                   "SP.POP.1014.MA":"Males (10-14)",

                   "SP.POP.1519.FE":"Females (15-19)",
                   "SP.POP.1519.MA":"Males (15-19)",

                   "SP.POP.2024.FE":"Females (20-24)",
                   "SP.POP.2024.MA":"Males (20-24)",

                   "SP.POP.2529.FE":"Females (25-29)",
                   "SP.POP.2529.MA":"Males (25-29)",
```

```
                 "SP.POP.3034.FE":"Females (30-34)",
                 "SP.POP.3034.MA":"Males (30-34)",

                 "SP.POP.3539.FE":"Females (35-39)",
                 "SP.POP.3539.MA":"Males (35-39)",

                 "SP.POP.4044.FE":"Females (40-44)",
                 "SP.POP.4044.MA":"Males (40-44)",

                 "SP.POP.4549.FE":"Females (45-49)",
                 "SP.POP.4549.MA":"Males (45-49)",

                 "SP.POP.5054.FE":"Females (50-54)",
                 "SP.POP.5054.MA":"Males (50-54)",

                 "SP.POP.5559.FE":"Females (55-59)",
                 "SP.POP.5559.MA":"Males (55-59)",

                 "SP.POP.6064.FE":"Females (60-64)",
                 "SP.POP.6064.MA":"Males (60-64)",

                 "SP.POP.6569.FE":"Females (65-69)",
                 "SP.POP.6569.MA":"Males (65-69)",

                 "SP.POP.7074.FE":"Females (70-74)",
                 "SP.POP.7074.MA":"Males (70-74)",

                 "SP.POP.7579.FE":"Females (75-79)",
                 "SP.POP.7579.MA":"Males (75-79)",

                 "SP.POP.80UP.FE":"Females (80+)",
                 "SP.POP.80UP.MA":"Males (80+)"

                }
world = wbdata.get_dataframe(variable_labels, country="")
world.head()
```

[3]:

country	date	Total Population	Female Population	Male Population	Females (0-14)	Males (0-14)	Females (15-64)	\
Afghanistan	1960	8996973.0	4347397.0	4649576.0				
	1961	9169410.0	4439158.0	4730252.0				
	1962	9351441.0	4535392.0	4816049.0				
	1963	9543205.0	4636172.0	4907033.0				
	1964	9744781.0	4741531.0	5003250.0				

```
country     date
Afghanistan 1960       1889085.0     1902314.0        2342557.0
            1961       1938162.0     1954539.0        2382132.0
            1962       1980414.0     2006726.0        2433192.0
            1963       2019727.0     2059859.0        2492014.0
            1964       2061674.0     2114431.0        2553216.0

                    Males (15-64)  Females (65+)  Males (65+)  Females (0-4)  \
country     date
Afghanistan 1960      2611254.0       115755.0     136008.0       760938.0
            1961      2637080.0       118864.0     138633.0       795378.0
            1962      2668877.0       121786.0     140446.0       818678.0
            1963      2705711.0       124431.0     141463.0       834934.0
            1964      2747068.0       126641.0     141751.0       850992.0

                    …  Females (60-64)  Males (60-64)  Females (65-69)  \
country     date  …
Afghanistan 1960  …          81378.0        93053.0          56624.0
            1961  …          82261.0        93359.0          57393.0
            1962  …          83487.0        94140.0          58326.0
            1963  …          85012.0        95186.0          59399.0
            1964  …          86747.0        96187.0          60569.0

                    Males (65-69)  Females (70-74)  Males (70-74)  \
country     date
Afghanistan 1960        67283.0          34655.0        40748.0
            1961        67330.0          35363.0        41520.0
            1962        67197.0          36055.0        42186.0
            1963        67052.0          36757.0        42704.0
            1964        67117.0          37461.0        42987.0

                    Females (75-79)  Males (75-79)  Females (80+)  Males (80+)
country     date
Afghanistan 1960          16990.0        19683.0         7486.0       8294.0
            1961          17750.0        20500.0         8358.0       9283.0
            1962          18364.0        21050.0         9041.0      10013.0
            1963          18819.0        21331.0         9456.0      10376.0
            1964          19098.0        21378.0         9513.0      10269.0

[5 rows x 43 columns]
```

[4]:
```
world['Total Population']
```

[4]:
```
country      date
Afghanistan  1960     8996973.0
             1961     9169410.0
             1962     9351441.0
```

```
              1963     9543205.0
              1964     9744781.0
                          …
Zimbabwe      2016    14030390.0
              2017    14236745.0
              2018    14439018.0
              2019    14645468.0
              2020           NaN
Name: Total Population, Length: 16104, dtype: float64
```

[5]:
```
world.loc["China", "1970"]
```

[5]:
```
Total Population     818315000.0
Female Population    398315770.0
Male Population      419999230.0
Females (0-14)       161012345.0
Males (0-14)         169646014.0
Females (15-64)      219622356.0
Males (15-64)        237363620.0
Females (65+)         17681069.0
Males (65+)           12989596.0
Females (0-4)         65584375.0
Males (0-4)           68624211.0
Females (5-9)         52485537.0
Males (5-9)           55526358.0
Females (10-14)       42942434.0
Males (10-14)         45495445.0
Females (15-19)       44782029.0
Males (15-19)         47319220.0
Females (20-24)       31775811.0
Males (20-24)         33714434.0
Females (25-29)       24134106.0
Males (25-29)         27033426.0
Females (30-34)       22745491.0
Males (30-34)         26094031.0
Females (35-39)       21498047.0
Males (35-39)         24307928.0
Females (40-44)       18429006.0
Males (40-44)         21171912.0
Females (45-49)       16535607.0
Males (45-49)         17999527.0
Females (50-54)       15644010.0
Males (50-54)         16761329.0
Females (55-59)       13422847.0
Males (55-59)         13448064.0
Females (60-64)       10655401.0
Males (60-64)          9513748.0
```

```
Females (65-69)        8213500.0
Males (65-69)          6568845.0
Females (70-74)        5042650.0
Males (70-74)          3775367.0
Females (75-79)        2883120.0
Males (75-79)          1887193.0
Females (80+)          1541799.0
Males (80+)             758191.0
Name: (China, 1970), dtype: float64
```

[6]:
```
world.iloc[2]
```

[6]:
```
Total Population     9351441.0
Female Population    4535392.0
Male Population      4816049.0
Females (0-14)       1980414.0
Males (0-14)         2006726.0
Females (15-64)      2433192.0
Males (15-64)        2668877.0
Females (65+)         121786.0
Males (65+)           140446.0
Females (0-4)         818678.0
Males (0-4)           831163.0
Females (5-9)         611717.0
Males (5-9)           637537.0
Females (10-14)       550019.0
Males (10-14)         538026.0
Females (15-19)       477029.0
Males (15-19)         485476.0
Females (20-24)       394896.0
Males (20-24)         429297.0
Females (25-29)       337367.0
Males (25-29)         369574.0
Females (30-34)       285151.0
Males (30-34)         324628.0
Females (35-39)       239468.0
Males (35-39)         277057.0
Females (40-44)       200415.0
Males (40-44)         228662.0
Females (45-49)       167294.0
Males (45-49)         187610.0
Females (50-54)       137894.0
Males (50-54)         150506.0
Females (55-59)       110191.0
Males (55-59)         121928.0
Females (60-64)        83487.0
Males (60-64)          94140.0
```

```
Females (65-69)        58326.0
Males (65-69)          67197.0
Females (70-74)        36055.0
Males (70-74)          42186.0
Females (75-79)        18364.0
Males (75-79)          21050.0
Females (80+)           9041.0
Males (80+)            10013.0
Name: (Afghanistan, 1962), dtype: float64
```

[7]:
```
world.loc['Afghanistan', '1960']
```

[7]:
```
Total Population     8996973.0
Female Population    4347397.0
Male Population      4649576.0
Females (0-14)       1889085.0
Males (0-14)         1902314.0
Females (15-64)      2342557.0
Males (15-64)        2611254.0
Females (65+)         115755.0
Males (65+)           136008.0
Females (0-4)         760938.0
Males (0-4)           780471.0
Females (5-9)         583953.0
Males (5-9)           598721.0
Females (10-14)       544194.0
Males (10-14)         523122.0
Females (15-19)       447872.0
Males (15-19)         478075.0
Females (20-24)       382542.0
Males (20-24)         415970.0
Females (25-29)       325664.0
Males (25-29)         361150.0
Females (30-34)       275401.0
Males (30-34)         319216.0
Females (35-39)       231431.0
Males (35-39)         267882.0
Females (40-44)       194047.0
Males (40-44)         222625.0
Females (45-49)       162562.0
Males (45-49)         183174.0
Females (50-54)       134240.0
Males (50-54)         147801.0
Females (55-59)       107418.0
Males (55-59)         122306.0
Females (60-64)        81378.0
Males (60-64)          93053.0
```

```
Females (65-69)        56624.0
Males (65-69)          67283.0
Females (70-74)        34655.0
Males (70-74)          40748.0
Females (75-79)        16990.0
Males (75-79)          19683.0
Females (80+)           7486.0
Males (80+)             8294.0
Name: (Afghanistan, 1960), dtype: float64
```

[8]:
```
import pandas as pd
example =pd.DataFrame({'Age':␣
↪['0-4','5-9','10-14','15-19','20-24','25-29','30-34','35-39','40-44','45-49','50-54','55-59
                    'Male': [-49228000, -61283000, -64391000, -52437000,␣
↪-42955000, -44667000, -31570000, -23887000, -22390000, -20971000, -17685000,␣
↪-15450000, -13932000, -11020000, -7611000, -4653000, -1952000, -625000,␣
↪-116000, -14000, -1000],
                    'Female': [52367000, 64959000, 67161000, 55388000,␣
↪45448000, 47129000, 33436000, 26710000, 25627000, 23612000, 20075000,␣
↪16368000, 14220000, 10125000, 5984000, 3131000, 1151000, 312000, 49000,␣
↪4000, 0]})
example.head()
```

[8]:

	Age	Male	Female
0	0-4	-49228000	52367000
1	5-9	-61283000	64959000
2	10-14	-64391000	67161000
3	15-19	-52437000	55388000
4	20-24	-42955000	45448000

[9]:
```
trial = pd.DataFrame({'Category': world.loc["China", "1970"].index,
                      'Population': world.loc["China", "1970"]})
trial.filter(regex = 'Female').values
```

[9]:
```
array([], shape=(43, 0), dtype=float64)
```

[10]:
```
male_df = pd.DataFrame({'Category' : world.loc["China", "1970"].filter(regex =␣
↪'Male').index,
                        'Population' : world.loc["China", "1970"].filter(regex =␣
↪'Male').values})
male_df = male_df.loc[1:]
female_df = pd.DataFrame({'Category' : world.loc["China", "1970"].filter(regex␣
↪= 'Female').index,
                        'Population' : world.loc["China", "1970"].filter(regex =␣
↪'Female').values})
female_df = female_df.loc[1:]
```

```
#combine male and female dataframes
combined_df = pd.DataFrame({'Age' : ['0-14', '15-64', '65+', '0-4', '5-9',␣
↪'10-14', '15-19', '20-24', '25-29', '30-34', '35-39', '40-44', '45-49',␣
↪'50-54', '55-59', '60-64', '65-69', '70-74', '75-79', '80+'],
                            'Male' : male_df['Population'] * -1,
                            'Female': female_df['Population']})
combined_df = combined_df.loc[4:]
combined_df.index = np.arange(1, len(combined_df) + 1)
combined_df
```

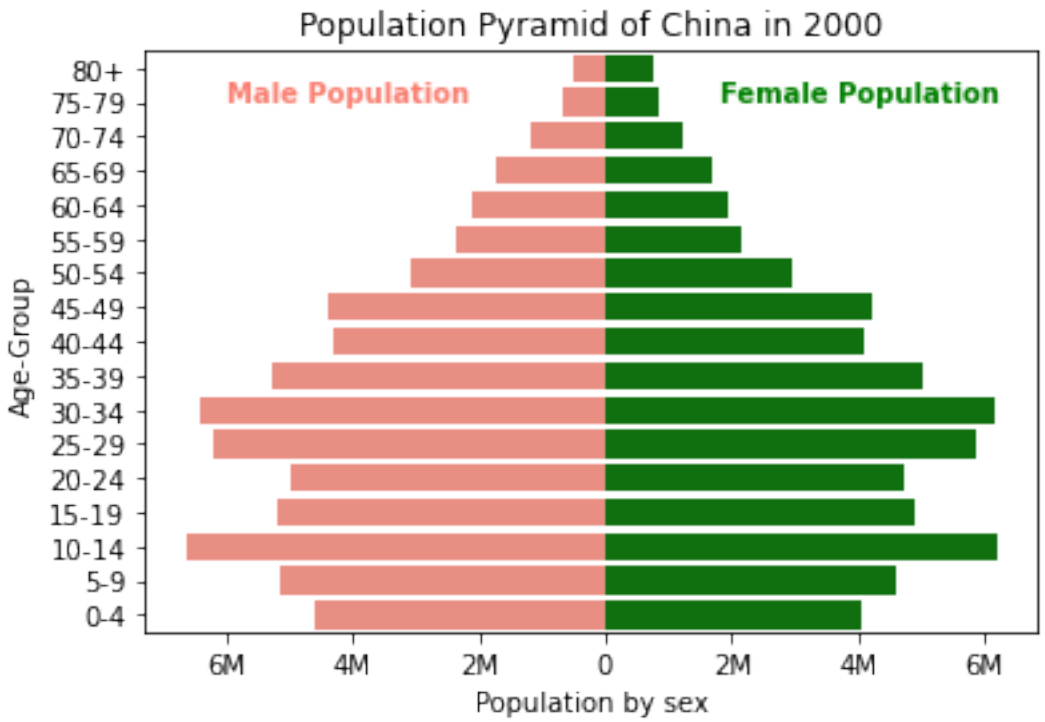
[10]:

	Age	Male	Female
1	0-4	-68624211.0	65584375.0
2	5-9	-55526358.0	52485537.0
3	10-14	-45495445.0	42942434.0
4	15-19	-47319220.0	44782029.0
5	20-24	-33714434.0	31775811.0
6	25-29	-27033426.0	24134106.0
7	30-34	-26094031.0	22745491.0
8	35-39	-24307928.0	21498047.0
9	40-44	-21171912.0	18429006.0
10	45-49	-17999527.0	16535607.0
11	50-54	-16761329.0	15644010.0
12	55-59	-13448064.0	13422847.0
13	60-64	-9513748.0	10655401.0
14	65-69	-6568845.0	8213500.0
15	70-74	-3775367.0	5042650.0
16	75-79	-1887193.0	2883120.0
17	80+	-758191.0	1541799.0

[341]:
```
def population_pyramid(country, date):
    #create male and female dataframes with info we want
    male_df = pd.DataFrame({'Category' : world.loc[country, date].filter(regex␣
↪= 'Male').index,
                        'Population' : world.loc[country, date].filter(regex =␣
↪'Male').values})
    male_df = male_df.loc[1:]
    female_df = pd.DataFrame({'Category' : world.loc[country, date].
↪filter(regex = 'Female').index,
                        'Population' : world.loc[country, date].filter(regex =␣
↪'Female').values})
    female_df = female_df.loc[1:]

    #combine male and female dataframes
    combined_df = pd.DataFrame({'Age' : ['0-14', '15-64', '65+', '0-4', '5-9',␣
↪'10-14', '15-19', '20-24', '25-29', '30-34', '35-39', '40-44', '45-49',␣
↪'50-54', '55-59', '60-64', '65-69', '70-74', '75-79', '80+'],
```

```
                            'Male' : male_df['Population'] * -1,
                            'Female': female_df['Population']})
    combined_df = combined_df.loc[4:]
    combined_df.index = np.arange(1, len(combined_df) + 1)

    #make a population pyramid using sns
    AgeClass =␣
↪['80+','75-79','70-74','65-69','60-64','55-59','50-54','45-49','40-44','35-39','30-34','25-
    labels = ['8M', '6M', '4M', '2M', '0', '2M', '4M', '6M', '8M']

    bar_plot = sns.barplot(x='Male', y='Age', data=combined_df, order=AgeClass,␣
↪color='Salmon', lw=0)
    bar_plot = sns.barplot(x='Female', y='Age', data=combined_df,␣
↪order=AgeClass, color='Green', lw=0)

    bar_plot.set(xlabel="Population by sex", ylabel="Age-Group")
    bar_plot.set_xticklabels(labels)
    bar_plot.text(18000000 , 1, "Female Population", color = "Green", weight =␣
↪"semibold")
    bar_plot.text(-60000000 ,1, "Male Population", color = "Salmon", weight =␣
↪"semibold")
    plt.title("Population Pyramid of %s in %d" % (country, int(date)))
    return bar_plot

population_pyramid("China", "2000")
```

[341]: <AxesSubplot:title={'center':'Population Pyramid of China in 2000'}, xlabel='Population by sex', ylabel='Age-Group'>



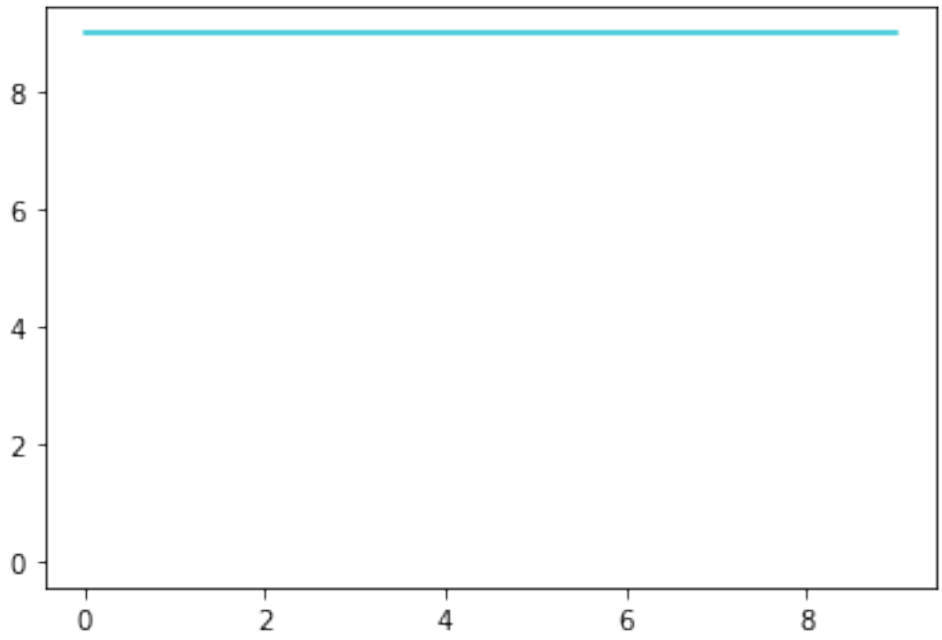
1.0.3 Here's my best try yet

[22]:
```
!pip install celluloid
```

```
Requirement already satisfied: celluloid in /opt/conda/lib/python3.8/site-
packages (0.2.0)
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.8/site-
packages (from celluloid) (3.3.3)
Requirement already satisfied: python-dateutil>=2.1 in
/opt/conda/lib/python3.8/site-packages (from matplotlib->celluloid) (2.8.1)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.8/site-
packages (from matplotlib->celluloid) (8.1.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
/opt/conda/lib/python3.8/site-packages (from matplotlib->celluloid) (2.4.7)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.8/site-packages (from matplotlib->celluloid) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.8/site-
packages (from matplotlib->celluloid) (0.10.0)
Requirement already satisfied: numpy>=1.15 in /opt/conda/lib/python3.8/site-
packages (from matplotlib->celluloid) (1.19.5)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.8/site-
packages (from python-dateutil>=2.1->matplotlib->celluloid) (1.15.0)
```

[188]:
```
from matplotlib import pyplot as plt
from celluloid import Camera
from IPython.display import HTML
def animate():
    fig = plt.figure()
    camera = Camera(fig)
    for i in range(10):
        plt.plot([i] * 10)
        camera.snap()
    animation = camera.animate()
    return HTML(animation.to_html5_video())


#HTML(animation.to_html5_video())
animate()
```

[188]:
```
<IPython.core.display.HTML object>
```



2 ANIMATED POP PYRAMID USING PLOTLY

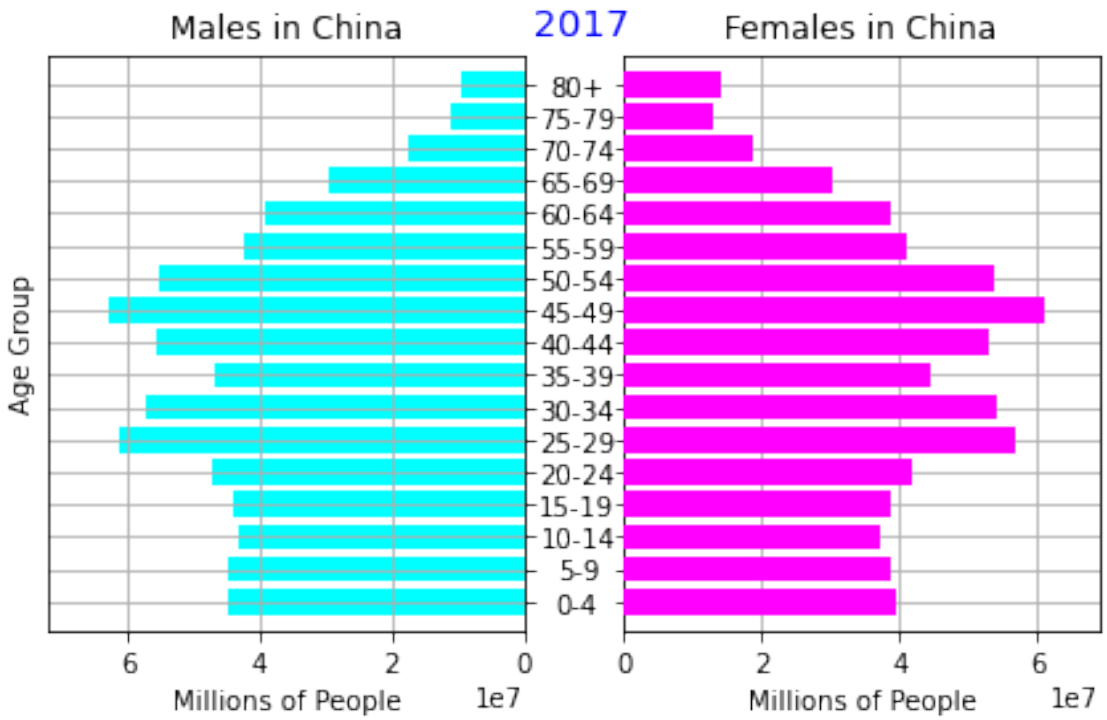
[343]:
```
from matplotlib import pyplot as plt
from celluloid import Camera
import plotly.express as px

import warnings
```

```
warnings.filterwarnings(action = 'ignore')

def anim_pop(country, start, end):


    fig, axes = plt.subplots(ncols=2, sharey=True)
    camera = Camera(fig)

    axes[0].invert_xaxis()

    i = 0
    while start != end:
        date = str(start)
        #create male and female dataframes with info we want

        male_df = pd.DataFrame({'Category' : world.loc[country, date].
↪filter(regex = 'Male').index,
                                'Population' : world.loc[country, date].
↪filter(regex = 'Male').values})
        male_df = male_df.loc[1:]
        female_df = pd.DataFrame({'Category' : world.loc[country, date].
↪filter(regex = 'Female').index,
                                  'Population' : world.loc[country, date].
↪filter(regex = 'Female').values})
        female_df = female_df.loc[1:]

        #combine male and female dataframes
        combined_df = pd.DataFrame({'Age' : ['0-14', '15-64', '65+', '0-4',␣
↪'5-9', '10-14', '15-19', '20-24', '25-29', '30-34', '35-39', '40-44',␣
↪'45-49', '50-54', '55-59', '60-64', '65-69', '70-74', '75-79', '80+'],
                                    'Male' : male_df['Population'] * -1,
                                    'Female': female_df['Population']})
        combined_df = combined_df.loc[4:]
        combined_df.index = np.arange(1, len(combined_df) + 1)

        #make a population pyramid using sns
        #AgeClass =␣
↪['80+','75-79','70-74','65-69','60-64','55-59','50-54','45-49','40-44','35-39','30-34','25-
        #labels = ['8M', '6M', '4M', '2M', '0', '2M', '4M', '6M', '8M']
        cd = combined_df.reset_index()
        index = cd["index"]
        Male = cd["Male"] * - 1
        Female = cd["Female"]
        ages = cd["Age"]
```

```
        axes[0].barh(index, Male, color = "cyan")
        axes[0].set(title='Males in %s' % (country))
        axes[1].barh(index, Female, align='center', color='magenta', zorder=10)
        axes[1].set(title='Females in %s' % (country))


        axes[0].set(yticks=y, yticklabels=[])
        for yloc, age in zip(y, ages):
            axes[0].annotate(age, (0.52, yloc + 0.5), xycoords=('figure␣
↪fraction', 'data'),
                         ha='center', va = "bottom")
        axes[0].yaxis.tick_right()
        axes[0].annotate(start, (0.48, 18.5), xycoords = ("figure fraction",␣
↪"data"), size = 'x-large', color = "Blue")

        for ax in axes.flat:
            ax.margins(0.03)
            ax.grid(True)

        fig.tight_layout()
        fig.subplots_adjust(wspace=0.21)
        axes[0].set_xlabel("Millions of People")
        axes[1].set_xlabel("Millions of People")
        axes[0].set_ylabel("Age Group")

        start += 1

        camera.snap()
    animation = camera.animate()
    return HTML(animation.to_html5_video())
```

[344]:
```
anim_pop("China", 1960, 2018)
```

[344]:
```
<IPython.core.display.HTML object>
```



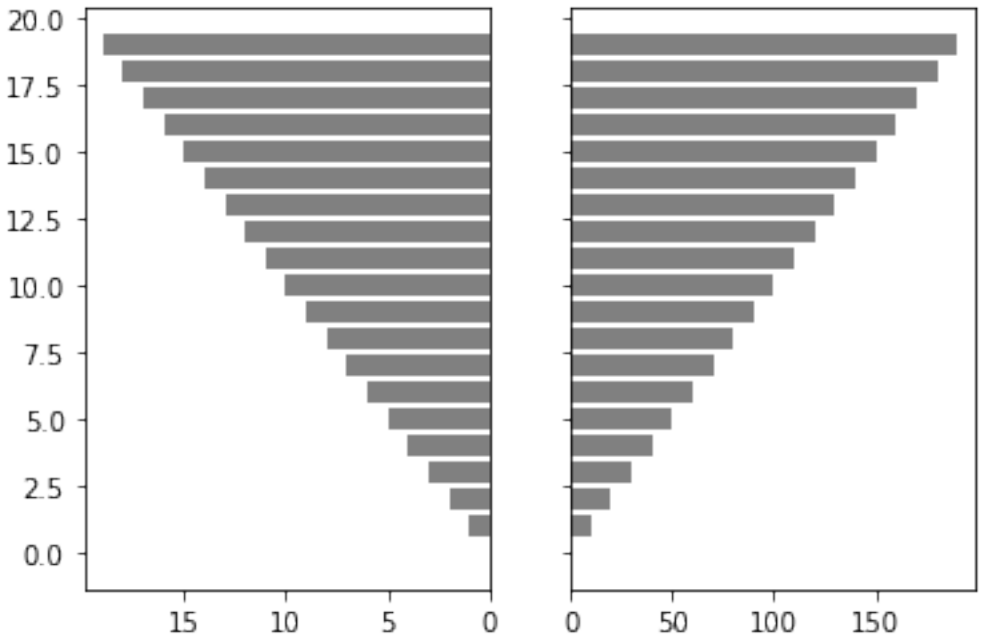
[ ]:

[ ]:

[ ]:

[ ]:

[136]:
```
y = range(20)
x1 = range(20)
x2 = range(0, 200, 10)

fig, axes = plt.subplots(ncols=2, sharey=True)
axes[0].barh(y, x1, align='center', color='gray')
axes[1].barh(y, x2, align='center', color='gray')
axes[0].invert_xaxis()
plt.show()
```



[334]:
```
from matplotlib import pyplot as plt
fig, axes = plt.subplots(ncols=2, sharey=True)
axes[0].barh(index, Male)
axes[0].set(title='Males in population')
axes[1].barh(index, Female, align='center', color='pink', zorder=10)
axes[1].set(title='Females in population')

axes[0].invert_xaxis()
axes[0].set(yticks=y, yticklabels=[])
for yloc, age in zip(y, ages):
    axes[0].annotate(age, (0.52, yloc + 0.5), xycoords=('figure fraction',␣
 ↪'data'),
                     ha='center', va = "bottom")
axes[0].yaxis.tick_right()
axes[0].annotate("1970", (0.46, 18.5), xycoords = ("figure fraction", "data"),␣
 ↪size = 'x-large', color = "Blue")
axes[0].set_xlabel("Millions of People")
axes[1].set_xlabel("Millions of People")
axes[0].set_ylabel("Age Group")

for ax in axes.flat:
    ax.margins(0.03)
    ax.grid(True)

fig.tight_layout()
```

```
fig.subplots_adjust(wspace=0.21)
fig.show()
```