

How can I create a population pyramid in Python?

Authored by
stats writer

April 16, 2024

RECOMMENDED CITATION

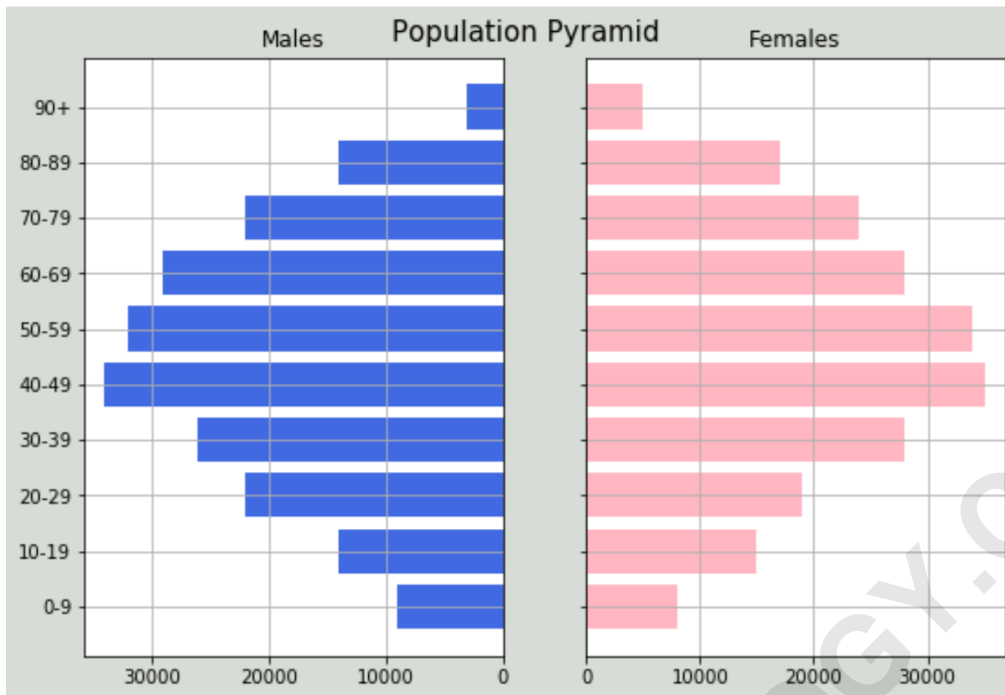
stats writer (2024). *How can I create a population pyramid in Python?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=136135>

Creating a population pyramid in Python involves using the programming language's built-in libraries and functions to visualize demographic data. This can be achieved by first importing the necessary libraries, such as Matplotlib or Pandas, and then using data manipulation techniques to organize the data into two separate groups based on age and gender. Finally, the data can be plotted in a pyramid-like shape, with age on the x-axis and population on the y-axis, to represent the distribution of a population across different age groups and genders. With proper coding techniques, a population pyramid can be easily created in Python to effectively display and analyze demographic patterns.

Create a Population Pyramid in Python

A population pyramid is a graph that shows the age and gender distribution of a given population. It's useful for understanding the composition of a population and the trend in population growth.

This tutorial explains how to create the following population pyramid in Python:



Population Pyramid in Python

Suppose we have the following dataset that displays the total population of males and females by age group for a given country:

```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

#create dataframe
df = pd.DataFrame({'Age': ,
'Male': ,
```

```
'Female': })
```

```
#view dataframe
```

```
df
```

```
Age Male Female
```

```
0 0-9 9000 8000
```

```
1 10-19 14000 15000
```

```
2 20-29 22000 19000
```

```
3 30-39 26000 28000
```

```
4 40-49 34000 35000
```

```
5 50-59 32000 34000
```

```
6 60-69 29000 28000
```

```
7 70-79 22000 24000
```

```
8 80-89 14000 17000
```

```
9 90+ 3000 5000
```

We can use the following code to create a population pyramid for the data:

```
#define x and y limits
```

```
y = range(0, len(df))
```

```
x_male = df
```

```
x_female = df
```

```
#define plot parameters
```

```
fig, axes = plt.subplots(ncols=2, sharey=True,  
figsize=(9, 6))
```

```
#specify background color and plot title
```

```
fig.patch.set_facecolor('xkcd:light grey')
```

```
plt.figtext(.5,.9,"Population Pyramid ", fontsize=15,  
ha='center')
```

```
#define male and female bars
```

```
axes.barh(y, x_male, align='center', color='royalblue')
```

```
axes.set(title='Males')
```

```
axes.barh(y, x_female, align='center', color='lightpink')
```

```
axes.set(title='Females')
```

```
#adjust grid parameters and specify labels for y-axis
```

```
axes.grid()
```

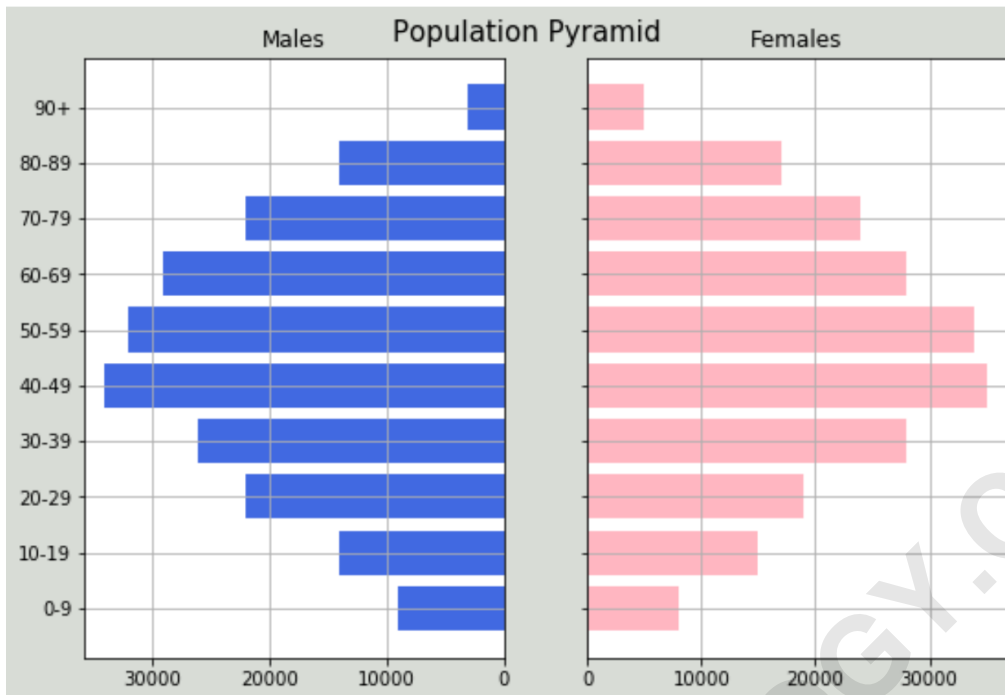
```
axes.set(yticks=y, yticklabels=df)
```

```
axes.invert_xaxis()
```

```
axes.grid()
```

```
#display plot
```

```
plt.show()
```



From the plot we can see that the distribution of males and females is fairly symmetrical, with most of the population falling in the middle-age brackets. By simply looking at this one plot, we can get a decent idea about the demographics of this particular country.

Note that you can adjust the colors of the plot background and the individual bars by specifying colors from the .

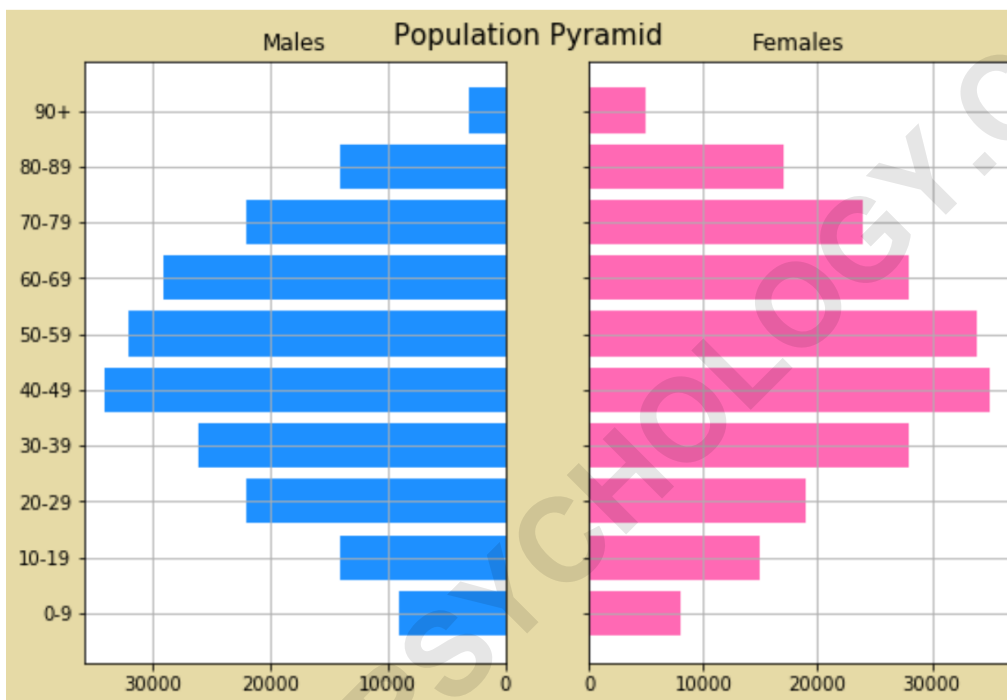
For example, we could specify 'hotpink' and 'dodgerblue' to be used with a 'beige' background:

```
fig.patch.set_facecolor('xkcd:beige')
```

```
axes.barh(y, x_male, align='center', color='dodgerblue')
```

```
axes.barh(y, x_female, align='center', color='hotpink')
```

```
plt.show()
```



Feel free to modify the color scheme based on what you think looks best.