

HW2_GraphPopusData

February 10, 2026

Week 2

```
[2]: from palette import palette    # this is just a 'easy-on-the-eyes' color
      ↪ palette i made

import numpy as np
import matplotlib.pyplot as plt

data = np.genfromtxt(fname='popus.txt', dtype=int, delimiter='\t')    # Show me
      ↪ the goods
```

Plot using OOP:

Axes should have labels

Ticks scaled well

Non-default colors & line sizes

```
[9]: years = [item[0] for item in data]    # extracts years
      values = [item[1] for item in data]    # extracts population

fig, ax = plt.subplots(figsize=(16,10)) # I like big graphs

fig.set_facecolor(color=palette['black'])    # white hurts my eyes
ax.set_facecolor(color=palette['gray'])

ax.set_xlim(min(years), max(years))    # --/ setting hard limits
ax.set_ylim(150000000, 350000000)    # --/

ax.plot(years, values, color='k',
        marker='x', markersize=3.5,
        markerfacecolor='purple', markeredgecolor='purple')

ax.fill_between(years, values, color=palette['dark_blue'])    # filling the
      ↪ space under the projected line
ax.set_title('Population Over Time', fontsize=18)

# making evenly spaced ticks
start = 140000000
```

```

end = int(max(values) // 10000000 + 1) * 10000000

ax.set_xticks(years)      # a tick for each year
ax.set_yticks(np.arange(start, end + 1, 10000000)) # 8 evenly spaced ticks

for i, label in enumerate(ax.get_xticklabels()):      # these 'for' loops
    ↪help make 'Major' ticks every 5 years and 'Minor' ticks in between
    label.set_rotation(90)
    if i % 5 == 0:
        label.set_fontsize(10)
        label.set_fontweight('bold')
    else:
        label.set_fontsize(8)

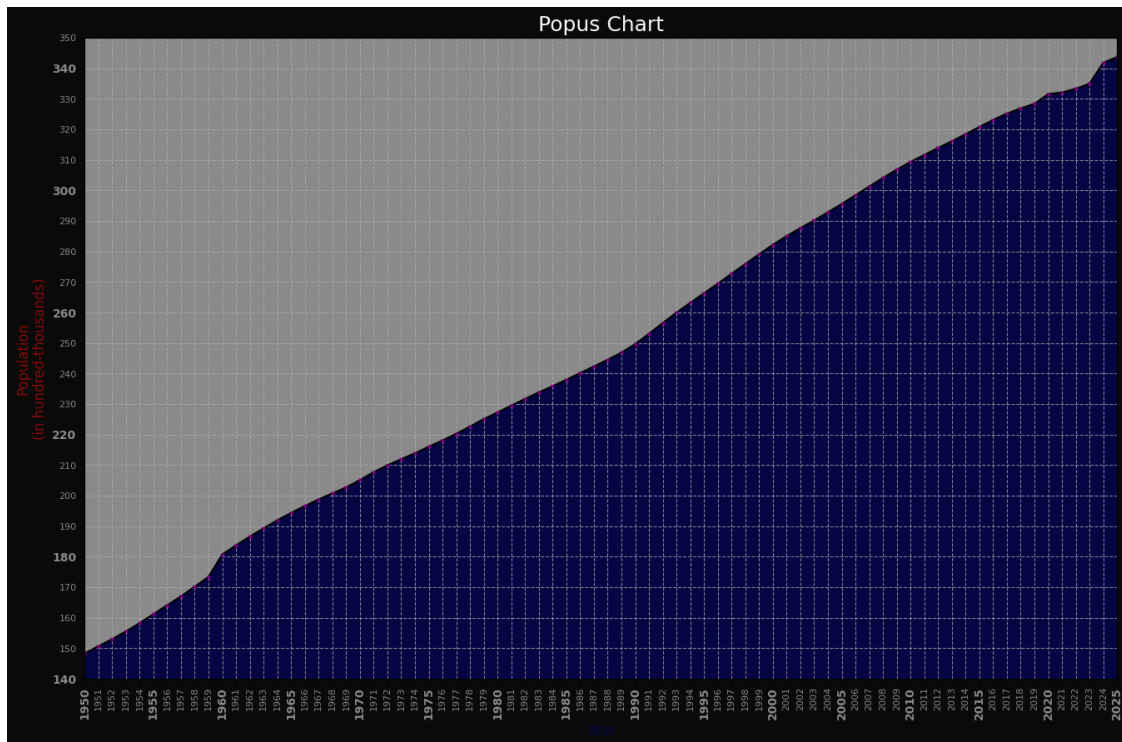
for i, label in enumerate(ax.get_yticklabels()):
    if i % 4 == 0:
        label.set_fontsize(10)
        label.set_fontweight('bold')
    else:
        label.set_fontsize(8)

ax.set_title('Popus Chart', fontsize=18, color=palette['white'])
ax.set_xlabel('Year', fontsize=12, color=palette['dark_blue'], loc='center')
ax.set_xticklabels(years, color=palette['gray'])
ax.set_ylabel('Population\n(in hundred-thousands)', fontsize=12,
    ↪color=palette['red'], loc='center')
ax.set_yticklabels([f"{int(y/1000000)}" for y in ax.get_yticks()],
    ↪color=palette['gray'])

ax.grid(True, linestyle='--', alpha=0.75)

plt.show()

```



Plot 2 – (year to year):

Bar graph

NumPy function calculates differences between arrays

Axis should have labels

Ticks scaled well

Non-default colors/lines sizes

```
[24]: new_vals = np.diff(values)
      new_yrs = years[1:]

      figure, ax2 = plt.subplots(figsize=(16,10))
      figure.set_facecolor(color=palette['black'])
      ax2.set_facecolor(color=palette['gray'])

      bars = ax2.bar(new_yrs, -new_vals, color=palette['dark_blue'],
                    ↪edgecolor=palette['red'])

      for bar, value in zip(bars, -new_vals):
          ax2.text(
              bar.get_x() + bar.get_width() / 2,
              bar.get_height() / 2,
```

```

        f"{int(value)}",
        ha='center',
        va='center',
        color=palette['white'],
        fontsize=8,
        rotation=90
    )

ax2.set_title('Population Year to Year', fontsize=18, color=palette['gray'])
ax2.set_xlabel('Year', fontsize=12, color=palette['blue'])
ax2.set_ylabel('Population\n(in millions)', fontsize=12, color=palette['red'])

y_ticks = ax2.get_yticks()
ax2.set_xticks(new_yrs)
ax2.set_yticks(y_ticks)

for i, label in enumerate(ax2.get_xticklabels()):
    label.set_rotation(90)
    label.set_color(palette['gray'])
    if i % 5 == 0:
        label.set_fontsize(10)
        label.set_fontweight('bold')
    else:
        label.set_fontsize(6)

ax2.set_xticklabels([str(year) for year in new_yrs], color=palette['gray'])
ax2.set_yticklabels([f"{int(np.abs(y))/1000000}" for y in y_ticks],
                    color=palette['gray'])

ax2.grid(True, linestyle='--', color=palette['gray'], alpha=0.5)
plt.tight_layout()

```

