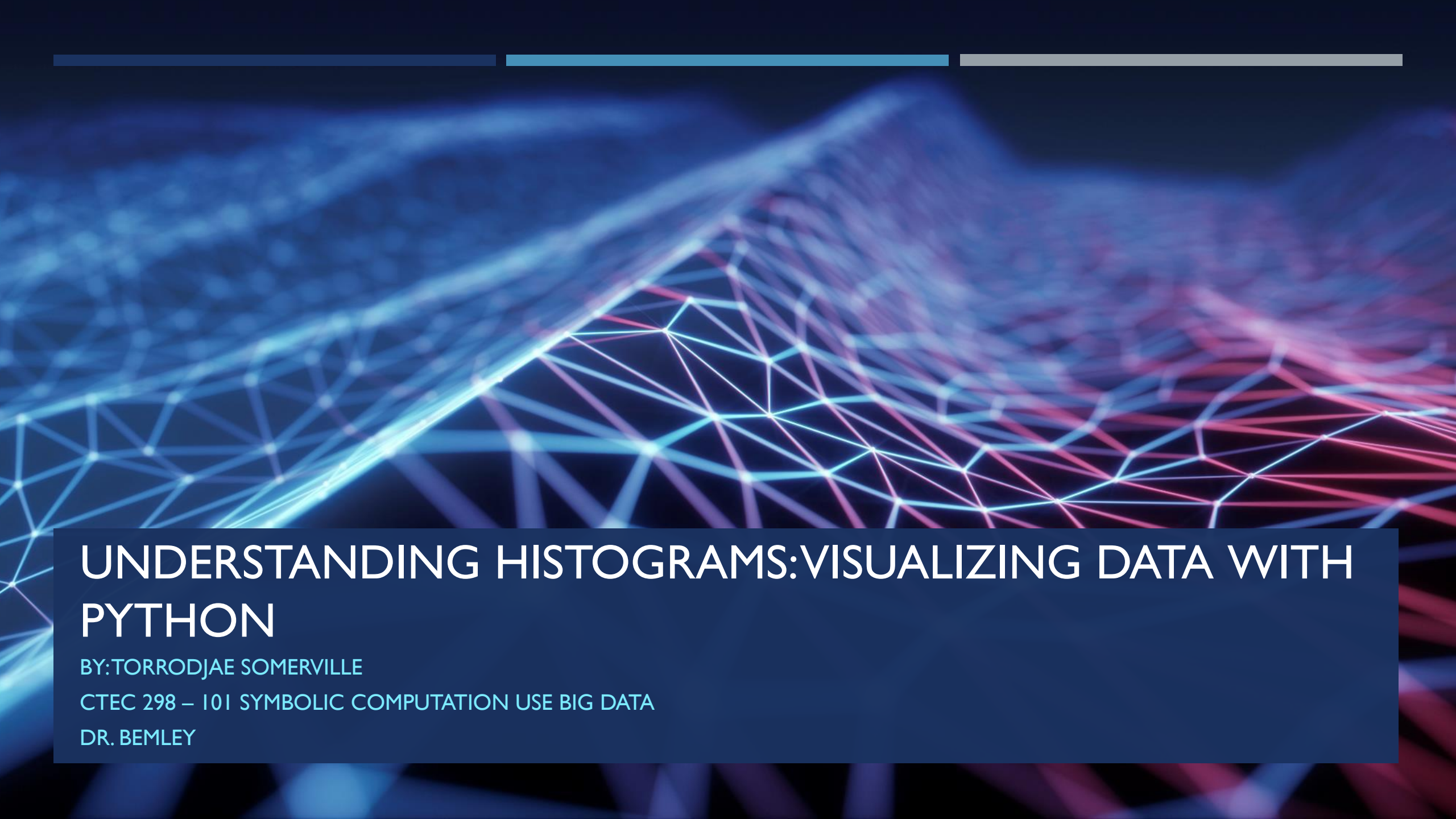


UNDERSTANDING HISTOGRAMS: VISUALIZING DATA WITH PYTHON

BY: TORRODJAE SOMERVILLE

CTEC 298 – 101 SYMBOLIC COMPUTATION USE BIG DATA

DR. BEMLEY

WHAT IS A HISTOGRAM?

- In simple terms, a histogram is a **graph that shows how often different values appear in your data**.
- It groups numbers into ranges called **"bins"**.
- The height of each bar shows the **frequency** which is just a fancy word for the **count or how many data points** fall into that specific range.
- 🚫 **It's NOT a Bar Graph: The Key Difference**
  - **A Bar Graph** compares *different, separate categories* (e.g., sales of Apples vs. Oranges vs. Bananas). The bars have space between them.
  - **A Histogram** shows the distribution of *one continuous variable* (e.g., the ages of everyone in a room). The bars represent ranges of a number line (like 0-9, 10-19) and they **touch each other** to show the data is continuous.
  - **The main question it answers is: "Where are my values clustering, and how are they spread out?"**

THE HISTORY: WHY PEARSON CREATED THE HISTOGRAM

- **The Problem (pre-1895):** Scientists like Pearson had tons of measurement data (like skull sizes, plant heights) but no good way to *see the patterns* in continuous data.
- **The Innovation:** In 1895, **Karl Pearson** (Statistician) needed to analyze evolutionary data and invented the histogram as a visual tool to:
  - Spot underlying patterns in continuous measurements
  - Compare distributions between different groups
  - Test his theories about heredity and evolution
- **The Legacy:** It revolutionized how we work with data by letting us **visually analyze distributions** rather than just looking at numbers.
- **My Connection:** Today, we're using Pearson's same invention to analyze matplotlib population ages, exactly the kind of continuous data he designed it for.

THE BEST TIMES TO USE A HISTOGRAM:

- You'd use a histogram anytime you have a bunch of numbers and you want to quickly understand their story.
- **For example, you can use it to:**
  - See the **frequency** of different age groups in a population (like we will do!).
  - Check if test scores are normally distributed (that classic bell curve).
  - Find the most common price range for products sold in a store.
  - Spot weird outliers, like a single, super-expensive item.
- **It's your go-to for the first step in analyzing any numerical dataset.**

BUILDING ONE WITH MATPLOTLIB (THE CODE):

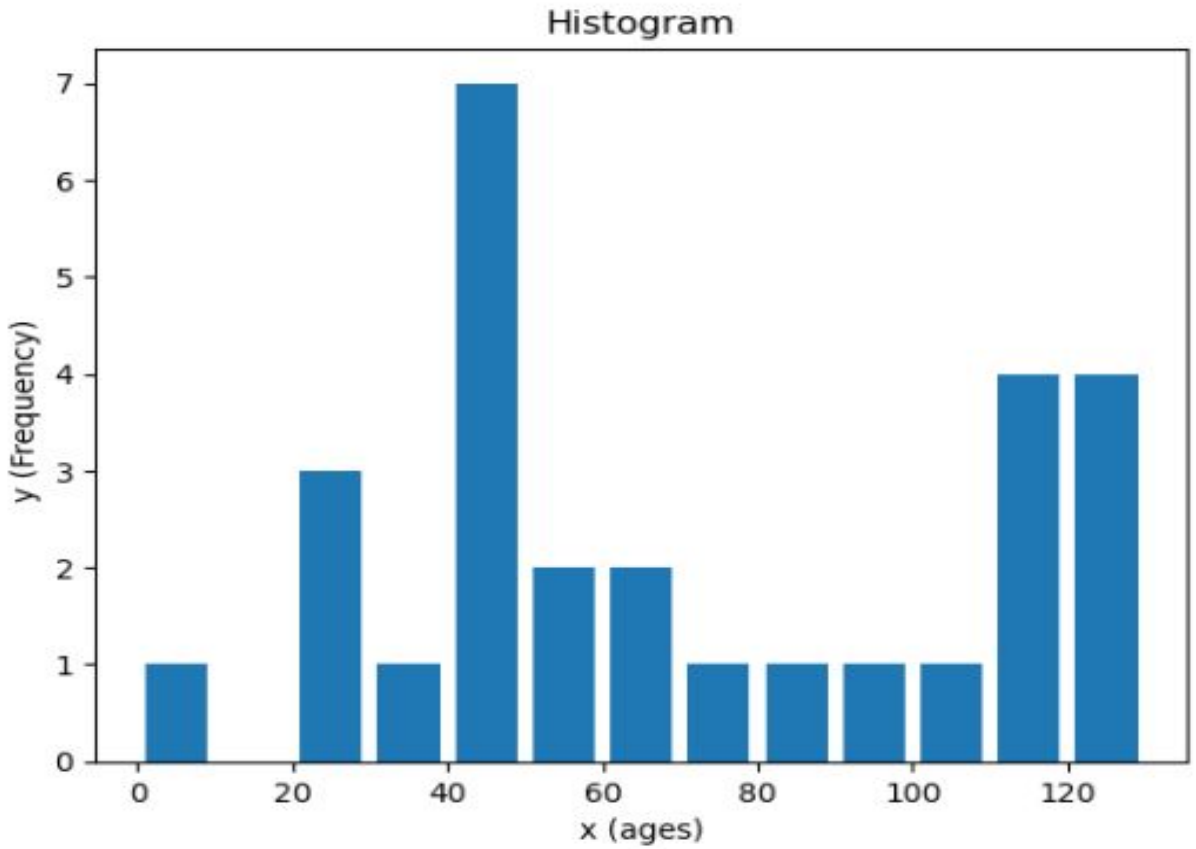
```
#importing matplotlib for visualization
import matplotlib.pyplot as plt

#Definine Dataset
population_ages = [22,55,62,45,21,22,34,42,42,4,99,102,110,120,121,122,130,111,115,112,80,75,65,54,44,43,42,48]

#Define the Bins (Age Ranges)
bins = [0,10,20,30,40,50,60,70,80,90,100,110,120,130]

#Creating visual
plt.hist(population_ages, bins, histtype='bar', rwidth=0.8)

#Adding Title and Labels
plt.xlabel('x (ages)')
plt.ylabel('y (Frequency) ')
plt.title('Histogram')

#Display plot
plt.show()
```

BREAKING DOWN THE CODE:

Here's what the main parts do:

- **population_ages**: This is just our raw list of everyone's age.
- **bins:** This is the most important part. It's how we create the groups.
  - Our bins [0,10,20,...] create groups for ages 0-9, 10-19, 20-29, and so on.
  - The computer counts how many ages fall into each group.
- **rwidth=0.8:** This makes the bars 80% wide so they don't touch each other too much, which makes it easier to read.
- **plt.ylabel(...):** Notice I put **"(Frequency)"** in the label. This reminds everyone that the y-axis is showing the **count** of people in each bin.

DATA VISUALIZED:



UNDERSTANDING THE GRAPH'S STORY:

Now, let's understand what we're seeing. Look at the height of the bars that **frequency** tells us everything.

- **The Biggest Group:** The tallest bar is for people in their 40s (the 40-50 bin). This has the highest **frequency**, meaning it's the most common age group in our sample.
- **Other Common Ages:** We also see a good amount of people in their 20s and 50s.
- **The Young and the Old:**
  - There's a small **frequency** in the 0-10 bin, meaning few children.
  - But look at the right side! We have several people over 100 years old, which is pretty cool.
- **A Quiet Generation:** The bar for the 70-80 age group is very short, meaning a low **frequency**. Maybe something happened that affected that generation.
- **So, by looking at the frequency in each bin, we get a full picture of our population's age structure.**

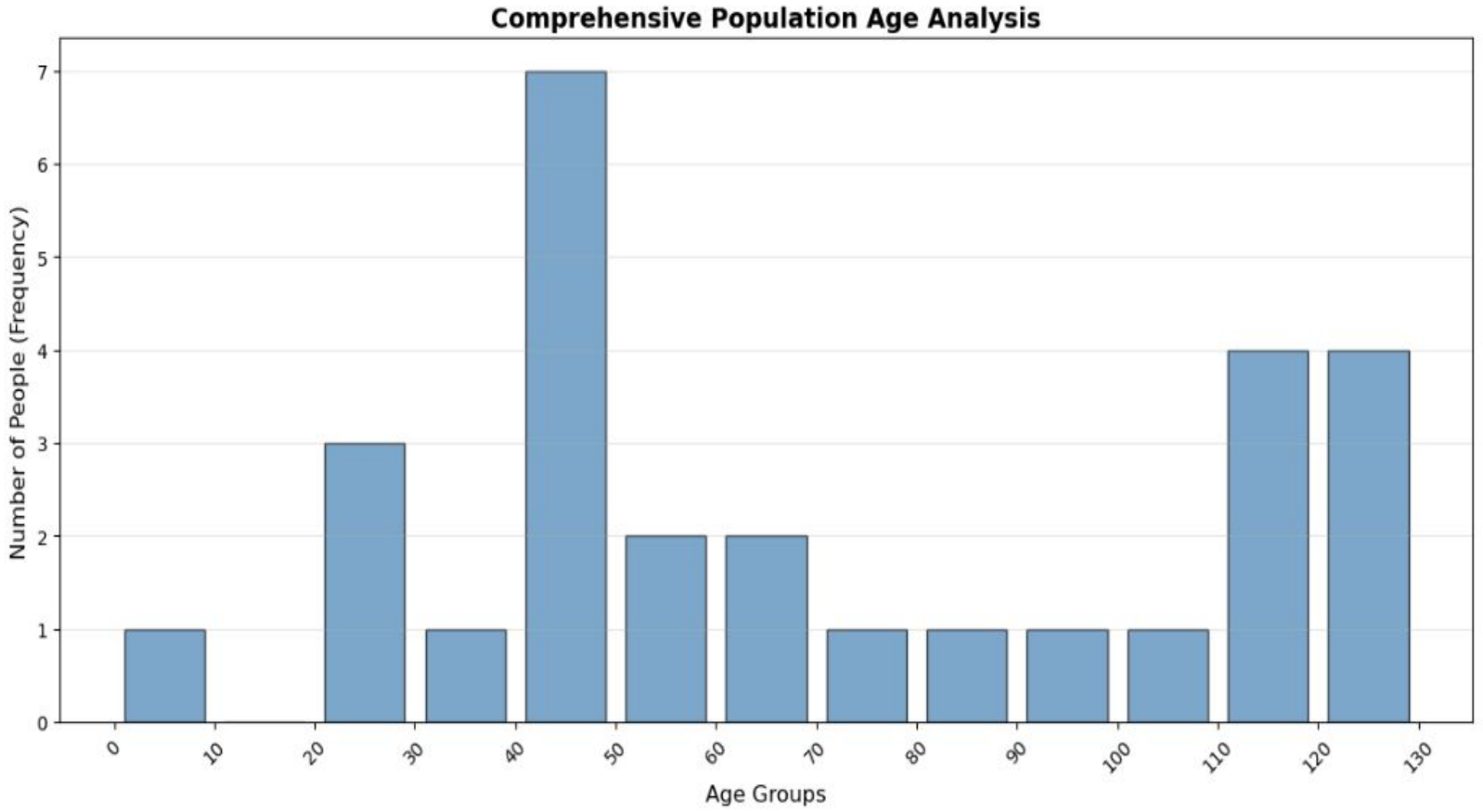
MAKING IT LOOK A LITTLE MORE APPEALING:

```
#importing matplotlib for visualization
import matplotlib.pyplot as plt

#Defining Dataset
population_ages = [22,55,62,45,21,22,34,42,42,4,99,102,110,120,121,122,130,111,115,112,80,75,65,54,44,43,42,48]

#Define the Bins (Age Ranges)
bins = [0,10,20,30,40,50,60,70,80,90,100,110,120,130]

#Creating professional visual
plt.figure(figsize=(12, 6))  # Makes the whole graph wider

plt.hist(population_ages, bins, histtype='bar', rwidth=0.8,
         color='steelblue', edgecolor='black', alpha=0.7) # Nicer colors with outlines

#Adding Title and Labels with better formatting
plt.xlabel('Age Groups', fontsize=12)
plt.ylabel('Number of People (Frequency)', fontsize=12)
plt.title('Comprehensive Population Age Analysis', fontsize=14, fontweight='bold')

#Adding professional touches
plt.grid(axis='y', alpha=0.3) # Adds light horizontal lines to help read the frequency
plt.xticks(bins, rotation=45) # Puts a label at every bin and tilts them

plt.tight_layout() # Fixes spacing so no labels get cut off

#Display plot
plt.show()
```

A CLEANER PROFESSION VIEW:

Why these changes help:

- **figsize=(12,6):** Gives us more space so it's not squished.
- **color, edgecolor:** The blue color with a black outline makes each bar's **frequency** easier to see.
- **grid(axis='y'):** Those horizontal lines help your eye follow the height of a bar over to the y-axis to read the exact **frequency**.
- **xticks(rotation=45):** Prevents the age group labels from overlapping.



SUMMARY AND KEY DETAILS:

- **Historical Foundation:** Histograms were invented by **Karl Pearson in 1895**, making them a foundational tool in statistics for over a century.
- **The Core Concept:** A histogram visualizes the **distribution** of numerical data by grouping values into **bins** and using bar heights to show **frequency** (the count in each bin).
- **The Key to Building It:** In Matplotlib, the plt.hist() function is powered by careful **bin selection**, which determines how we see the data's story.
- **The Power of Interpretation:** By analyzing the **shape, center, and spread**, we can understand our data's story. Our population analysis showed a complex structure with a strong middle-aged base.
- **Professional Presentation:** Customizing elements like colors, labels, and layout transforms a basic plot into an effective, communication-ready visualization.

REFERENCES

- Cappelleri, J. C., & Darlington, R. B. (2019). Histograms and the shape of distributions. In *Encyclopedia of Research Design*. SAGE Publications.
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55
- Matplotlib Development Team. (2021). Matplotlib: Visualization with Python. https://matplotlib.org/stable/contents.html
- Pearson, K. (1895). Contributions to the mathematical theory of evolution. II. Skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London. A, 186*, 343–414.
- VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data*. O'Reilly Media.
- Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021
- Wikipedia contributors. (2024, October 28). Histogram. In *Wikipedia, The Free Encyclopedia*. Retrieved November 18, 2025, from https://en.wikipedia.org/w/index.php?title=Histogram&oldid=1192341232

THANK YOU