General Assembly

STUDY GUIDE

# PRINCIPLES OF DATA VISUALIZATION (PYTHON)

## Key Terms and Definitions

» **Key Attributes of Visualization:** There are many attributes involved in visualization, but the three that are most immediately impactful are *position, color* and *size*.

» **Color Map:** The set of colors being used in a particular visualization. There are three types:

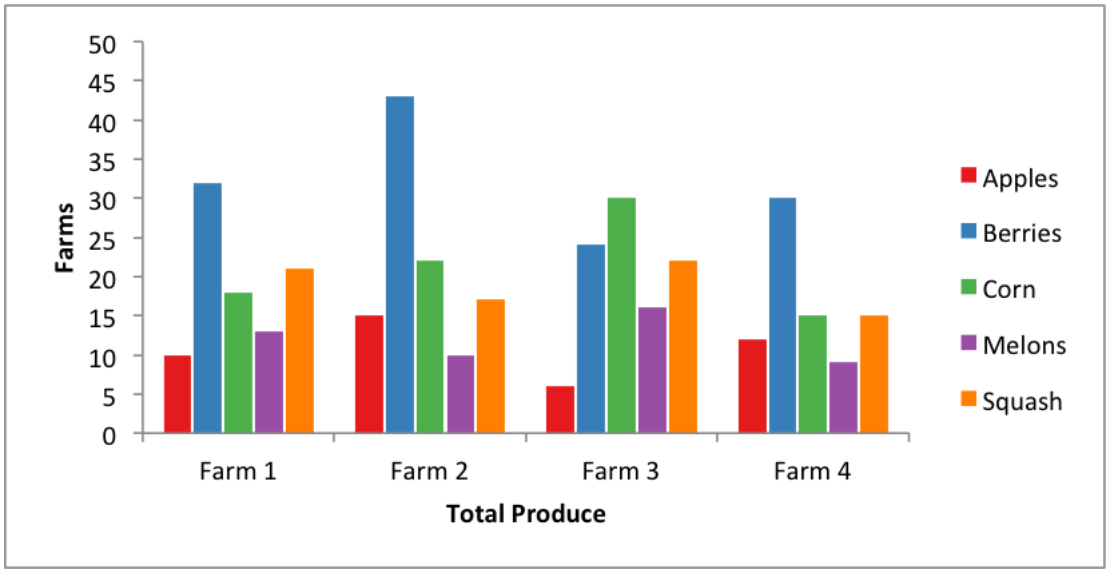
» Sequential: Best for values ordered from low to high.

» Divergent: Best for values that have a critical midpoint, such as an average or zero.

» Categorical: Best for values that fall into distinct groups (often for qualitative data).

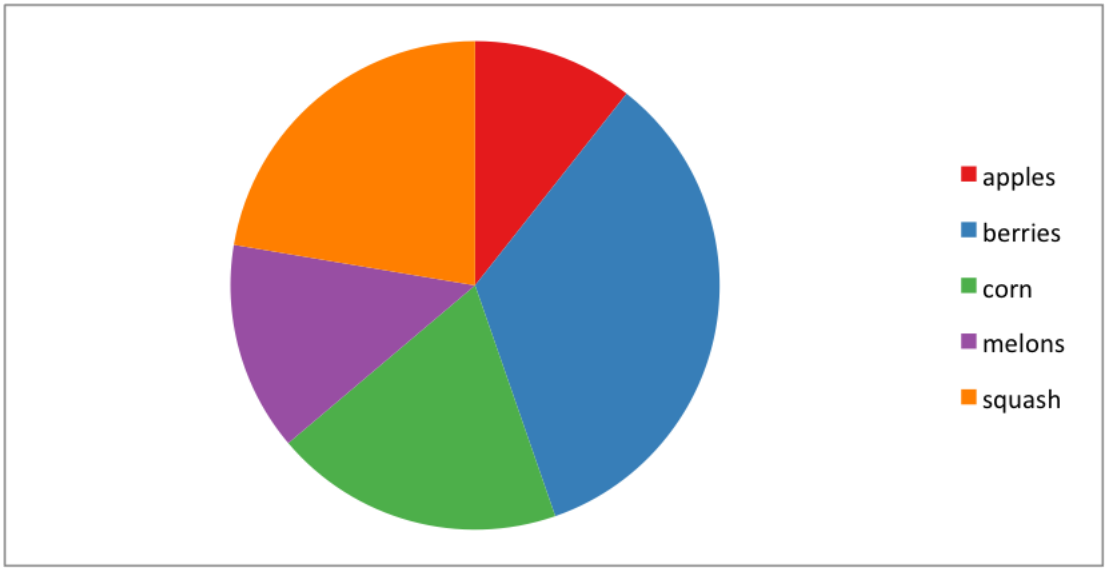
**Chart Types:**

» **Bar Chart**: One of the most common ways of visualizing data. It illustrates a summary statistic (e.g., count, sum, etc.) for each data point in a set of categories with a bar of a specific height. Bars have a space between them to indicate that they are distinct. They are best for numerical data that is split into categories.

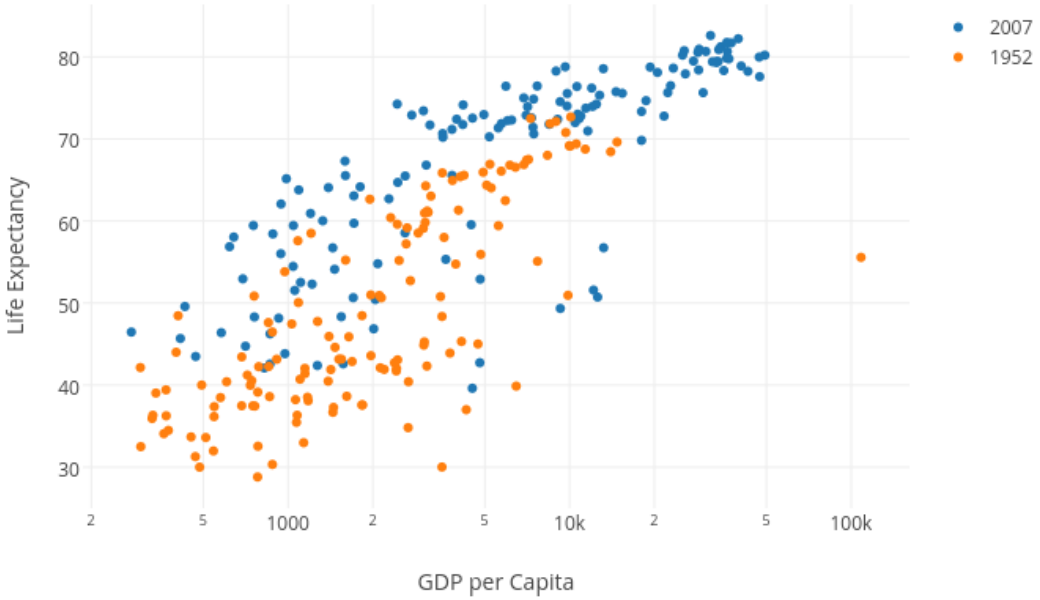



» **Pie Chart**: Shows the proportion of the whole that is represented by each category. They are commonly

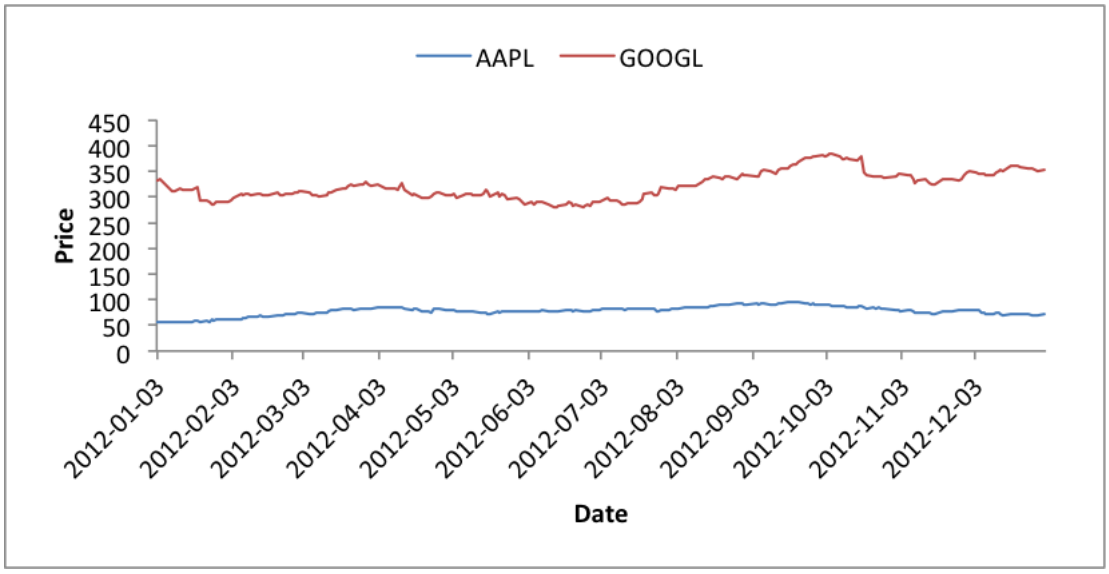
misused and can be hard to interpret. They should only be used when data is split into categories and you are showing relative proportions or percentages.



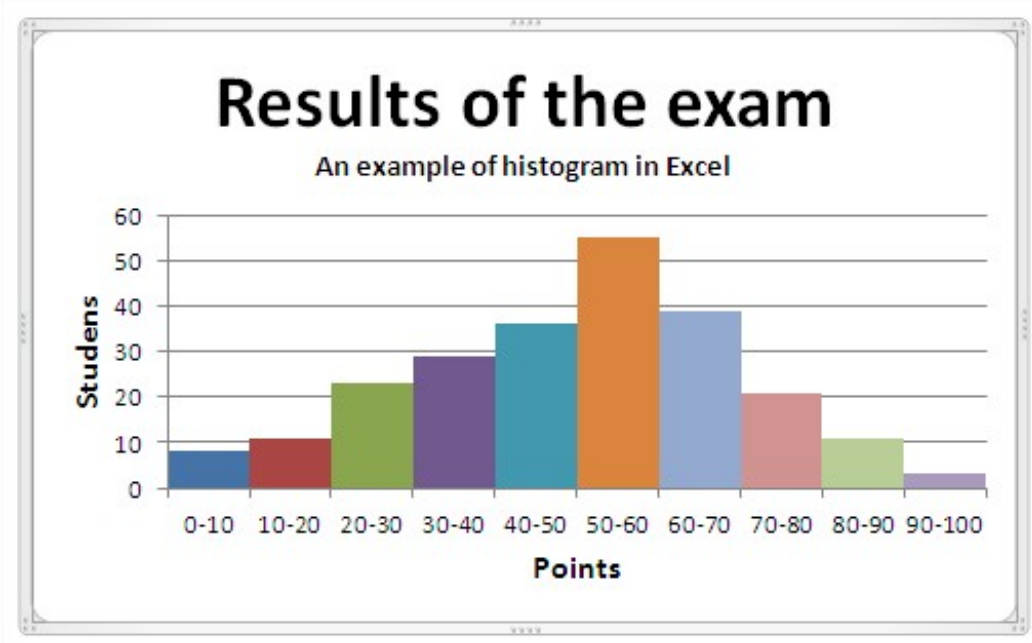

- » **Scatterplot**: Helpful for giving you a sense of trends, concentrations, and outliers. They plot data along two numerical axes. Best for comparing two-dimensional data (or two specific attributes of a data set) and the data lies along a continuous, numerical scale.




- » **Line Graph**: Used to indicate trends, most often ones that occur over time. The relationship between each data point matters and you should want to see increases and decreases between them. They should be used to show a progression or change over time, and the data represents a continuous trend, such as a steady increase in the independent variable.




- **Histogram**: Useful for exploring the distribution of your data, especially when checking for normal distribution (like a bell curve). They are similar to bar charts, but the x axis of a histogram represents a continuous variable rather than distinct categories. Unlike bar charts, you should not leave a gap between the bars of a histogram, because you're showing a continuous and connected data set. Use to show the distribution of a set of data along a continuous scale.




- **Python Plotting Libraries (Main)**:
  - **Matplotlib:** The core visualization library in Python.
  - **Seaborn:** A plotting library created on top of Matplotlib that is intended to make complex visualizations more immediately appealing. Seaborn is not a library for plotting basic chart types. It is used to create statistically based chart types and to quickly define chart styles.
  - **Pandas:** The Python library for organizing, cleaning, and otherwise manipulating your data, which a different approach to visualization. Instead of creating the plot and then inserting the data, it creates the graph directly from the data itself.

- **Python Plotting Libraries (Secondary)**:

- **Plotly:** A popular library for dynamic, interactive data visualizations. It is primarily used through an online interface.
- **Bokeh:** A library that's great for creating visualizations for the web, as its outputs can take the form of JSON, HTML, or other web applications. Additionally, the visualizations in Bokeh — like Plotly — are interactive and dynamic.
- **Pygal:** Another library for interactive visualizations. The main benefit of Pygal is the ability to output its plots as SVG files, which is important for designers.
- **ggplot:** Another library built on Matplotlib. It's less customizable than others, but its main value is its ability to recreate a similar library in R (ggplot2) for use within Python.

» **Exploratory Data Analysis (EDA):** The process of investigating data visually as a primary step in understanding the data set.

# Guiding Questions

1. Why are pie charts so frequently misused?
2. When would you use a histogram instead of a bar chart? How can you tell the difference?
3. Why should you include multiple chart types in a presentation or the EDA process?
4. With so many visualization libraries available, what are some distinguishing features that would help you choose which to use?
5. What are some of the key visualization attributes that make a chart immediately understandable for the human brain?

# Additional Resources

1. Another Great Lesson on Data Visualization.
2. Does Your Data Say What You Think It Says?
3. Tableau's Guide to Choosing a Chart.
4. Why Color Matters.
5. Choosing the Right Color Palette.
6. Guide to Color Types.
7. A Guide to Some of Python's Visualization Tools

# Technical Resources:

1. Install Anaconda and Verify Installation
2. Jupyter Notebook Practice