Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is widely used for plotting graphs and charts, and it integrates well with many other libraries and tools in the Python ecosystem.

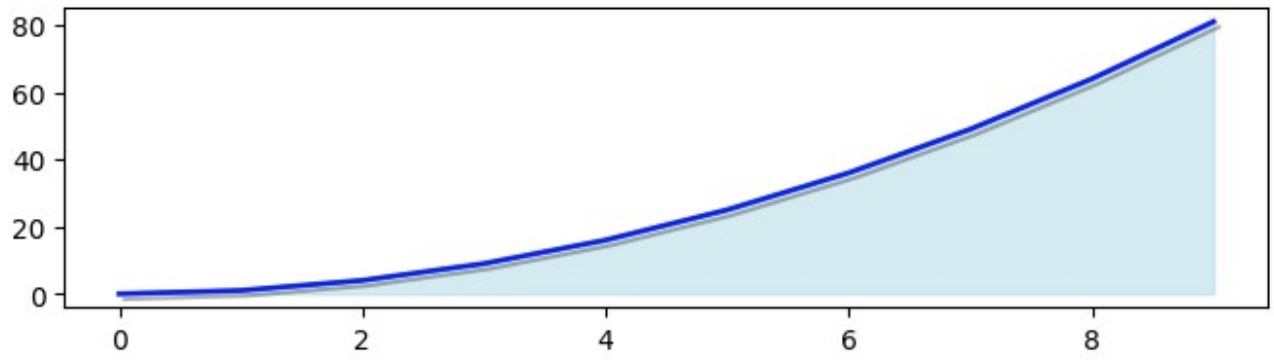
Key Features:

- **Versatile Plotting**: Supports a wide range of plots including `line`, `bar`, `scatter`, `histogram`, and more.
- **Customization**: Highly customizable with options to modify `colors`, `labels`, `scales`, and more.
- **Integration**: Works seamlessly with `NumPy`, `pandas`, and other scientific computing libraries.
- **Interactive Plots**: Capable of creating interactive plots that can be embedded in `Jupyter Notebooks` and web applications.
- **Publication Quality**: Generates high-quality plots suitable for publication.

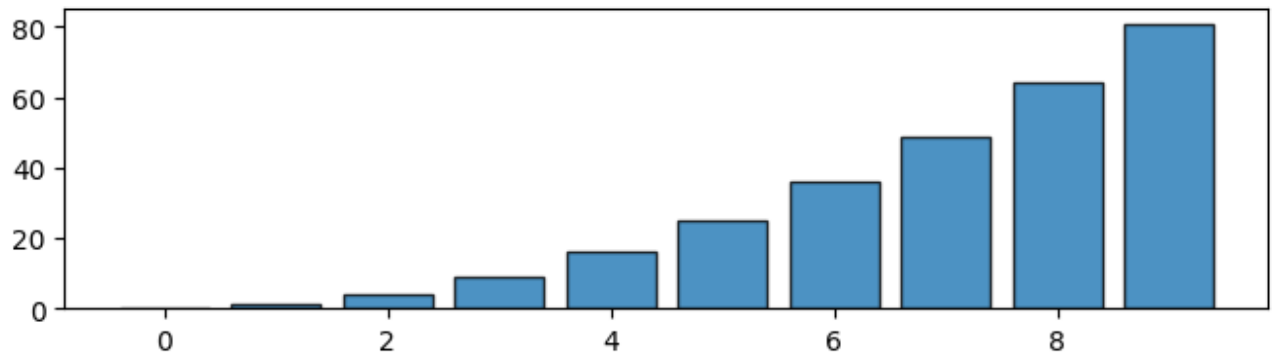
For more information, visit the Matplotlib documentation.

```
import matplotlib.pyplot as plt; import matplotlib.patheffects as pe

"1. Create a simple line plot with x = [0,1,2,3,4,5,6,7,8,9] and y = x^2."
x = [0,1,2,3,4,5,6,7,8,9]
y = [i**2 for i in x]; plt.figure(figsize=(8, 2))
plt.plot(x, y, path_effects=[pe.SimpleLineShadow(), pe.Normal()])
plt.plot(x, y, color='blue', linewidth=2, alpha=0.7, label='y = x²')
plt.fill_between(x, y, color='lightblue', alpha=0.5)
plt.show()
```

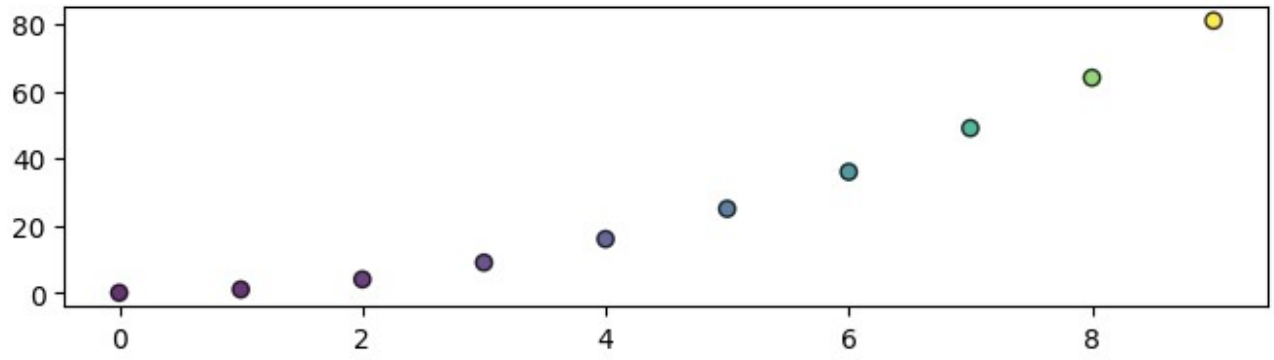



```
"2. Create a simple bar plot with x = [0,1,2,3,4,5,6,7,8,9] and y = x^2."
x = [0,1,2,3,4,5,6,7,8,9]
y = [i**2 for i in x]; plt.figure(figsize=(8, 2))
plt.bar(x, y, edgecolor='black', alpha=0.8)
plt.show()
```

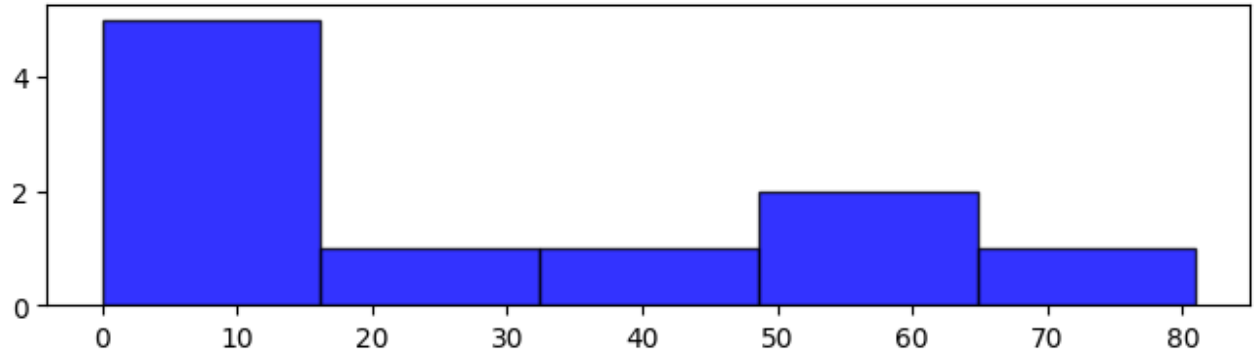



```
"3. Create a simple scatter plot with x = [0,1,2,3,4,5,6,7,8,9] and y = x^2."
x = [0,1,2,3,4,5,6,7,8,9]
```

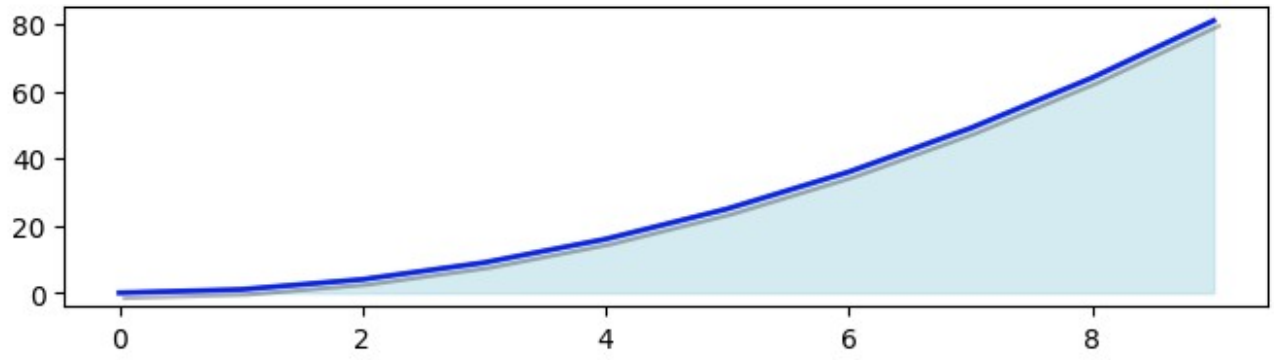
```
y = [i**2 for i in x]; plt.figure(figsize=(8, 2))
plt.scatter(x, y, c=y, cmap='viridis', edgecolors='black', alpha=0.8)
plt.show()
```




```
"4. Create a simple histogram plot with x = [0,1,2,3,4,5,6,7,8,9] and y = x^2."
import matplotlib.pyplot as plt; import matplotlib.patheffects as pe
x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
y = [i**2 for i in x]; plt.figure(figsize=(8, 2))
plt.hist(y, bins=5, color='blue', edgecolor='black', alpha=0.8)  # Use a single
color
plt.show()
```

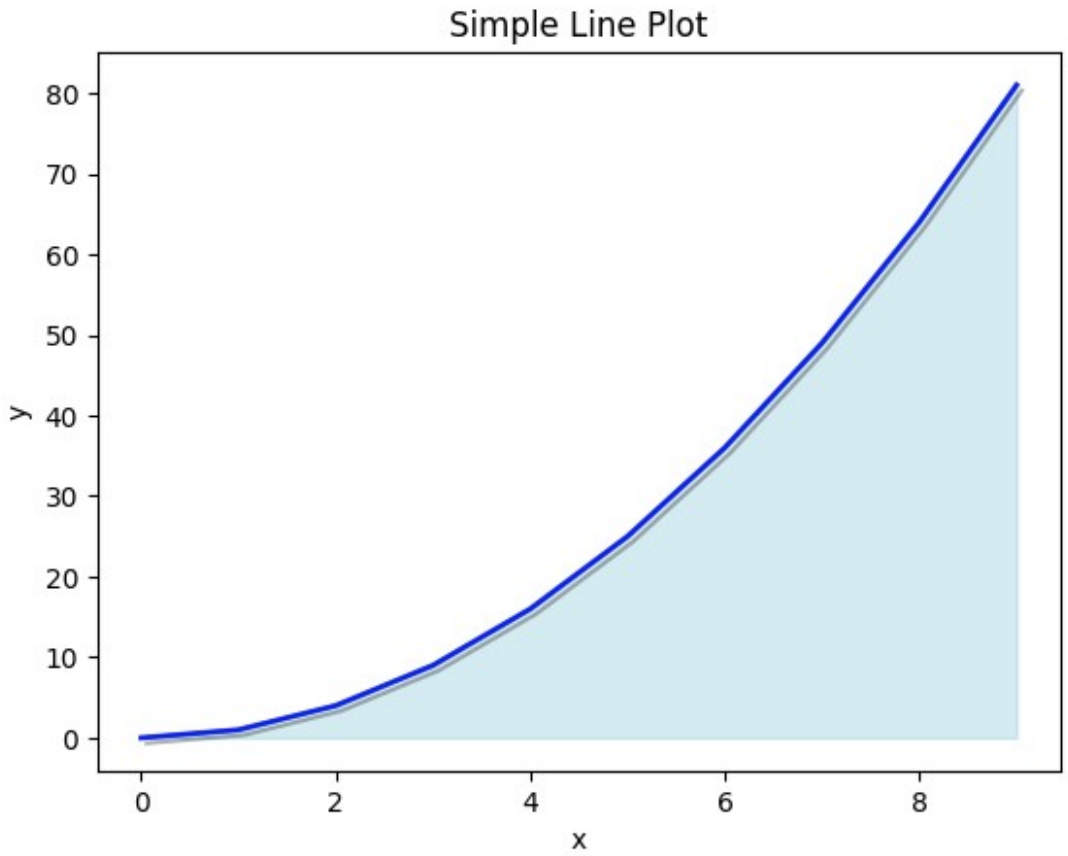



```
"5.  Create a simple plot with x = [0,1,2,3,4,5,6,7,8,9] and y = x². Save the
plot as a PNG image with the filename 'my_plot.png'."
x=range(10); y=[i**2 for i in x]; plt.figure(figsize=(8, 2))
plt.plot(x,y, path_effects=[pe.SimpleLineShadow(), pe.Normal()])
plt.plot(x, y, color='blue', linewidth=2, alpha=0.7, label='y = x²')
plt.fill_between(x, y, color='lightblue', alpha=0.5)
plt.savefig('my_plot.png'); plt.show()
```

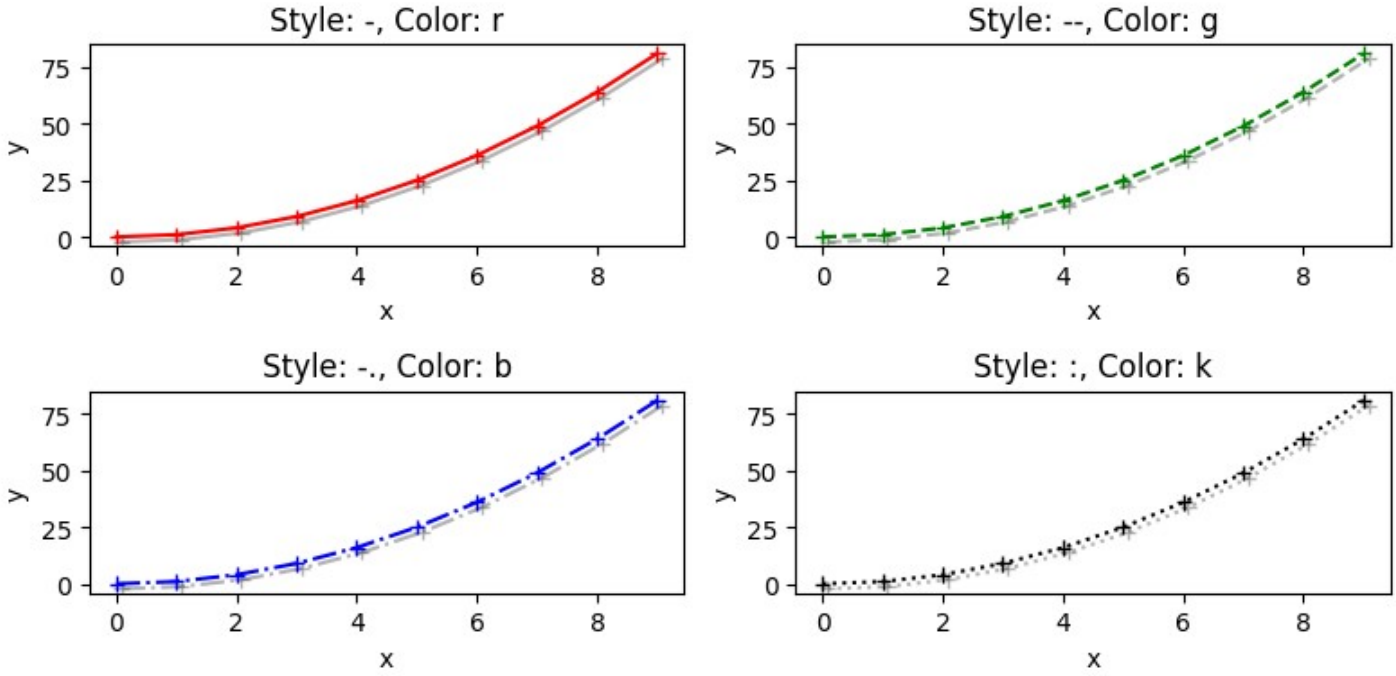



```
"6. Modify your previous plot to include the following: Adding Title and
Labels."
x=range(10); y=[i**2 for i in x];
```

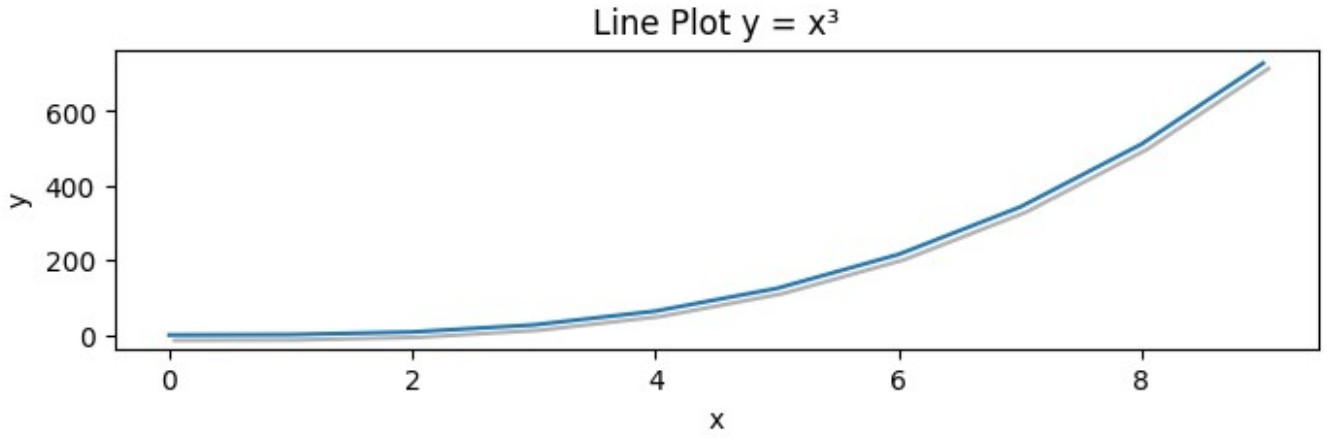
```
plt.plot(x,y, path_effects=[pe.SimpleLineShadow(), pe.Normal()])
plt.plot(x, y, color='blue', linewidth=2, alpha=0.7, label='y = x²')
plt.fill_between(x, y, color='lightblue', alpha=0.5)
plt.title('Simple Line Plot'); plt.xlabel('x'); plt.ylabel('y')
plt.show()
```



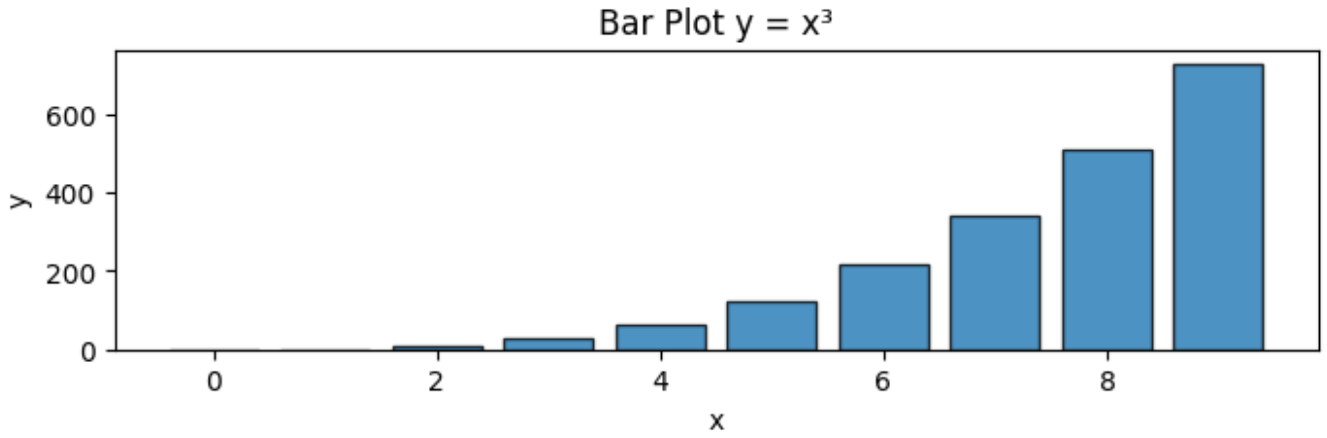

```
"7. Modify your previous plot to include the following: Adding Title and Labels. Customizing the Line Style, markers, color (4 types of line styles)."
x=range(10); y=[i**2 for i in x]
styles=['-','--','-.',':']; colors=['r','g','b','k']
fig, axs = plt.subplots(2, 2, figsize=(8, 4))
for ax, s, c in zip(axs.flatten(), styles, colors):
    ax.plot(x, y, linestyle=s, marker='+', color=c,
path_effects=[pe.SimpleLineShadow(), pe.Normal()])
    ax.set_title(f'Style: {s}, Color: {c}')
    ax.set_xlabel('x')
    ax.set_ylabel('y')
plt.tight_layout()
plt.show()
```



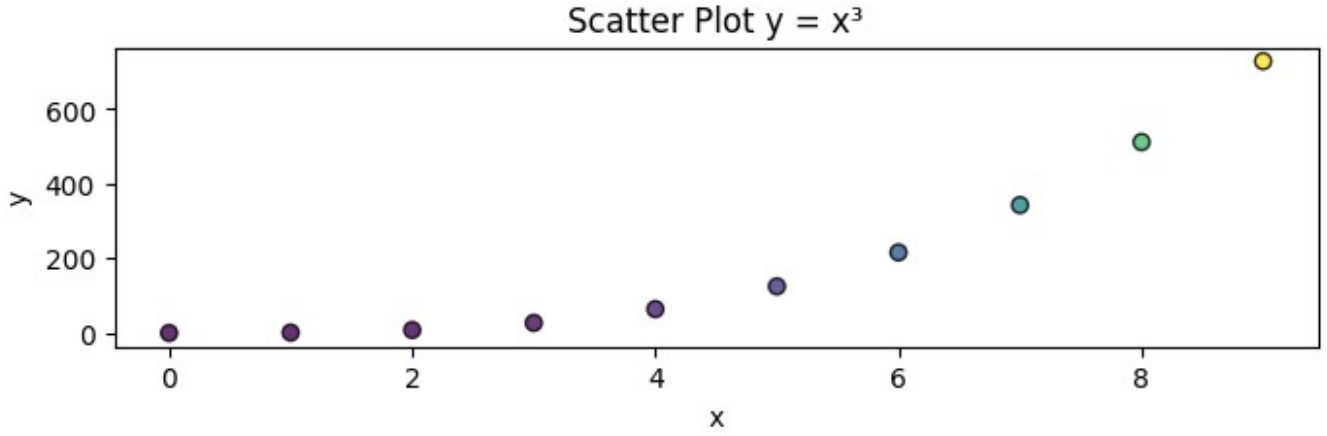

```
"8. Create a simple line plot with x = [0,1,2,3,4,5,6,7,8,9] and y = x³."
x=range(10); y=[i**3 for i in x]; plt.figure(figsize=(8, 2))
plt.plot(x,y, path_effects=[pe.SimpleLineShadow(), pe.Normal()])
plt.title('Line Plot y = x³'); plt.xlabel('x'); plt.ylabel('y')
plt.show()
```



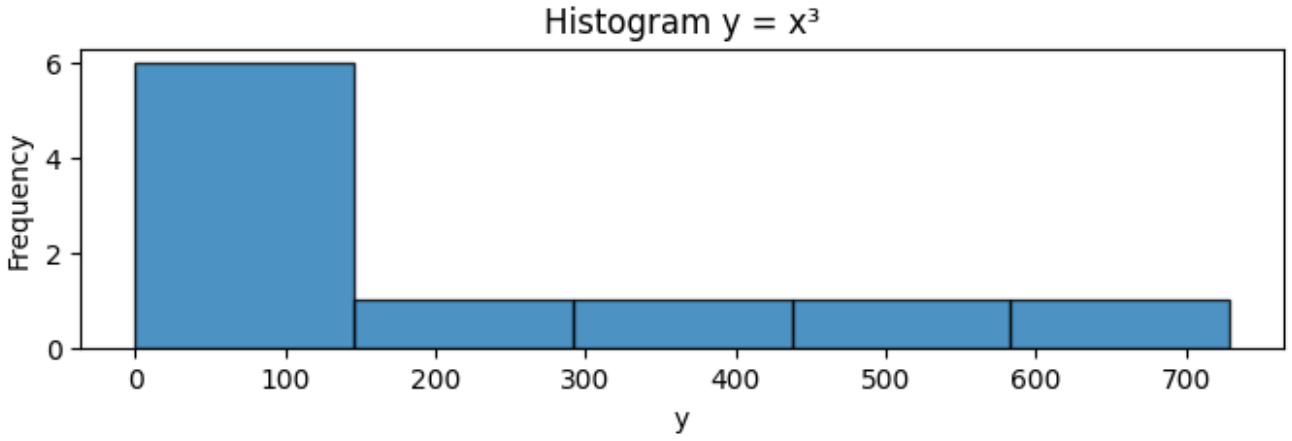

```
"9. Create a simple bar plot with x = [0,1,2,3,4,5,6,7,8,9] and y = x³."
x=range(10); y=[i**3 for i in x]; plt.figure(figsize=(8, 2))
plt.bar(x,y, edgecolor='black', alpha=0.8)
plt.title('Bar Plot y = x³'); plt.xlabel('x'); plt.ylabel('y')
plt.show()
```




```
"10. Create a simple scatter plot with x = [0,1,2,3,4,5,6,7,8,9] and y = x³."
x=range(10); y=[i**3 for i in x]; plt.figure(figsize=(8, 2))
plt.scatter(x,y, c=y, cmap='viridis', edgecolors='black', alpha=0.8)
plt.title('Scatter Plot y = x³'); plt.xlabel('x'); plt.ylabel('y')
plt.show()
```

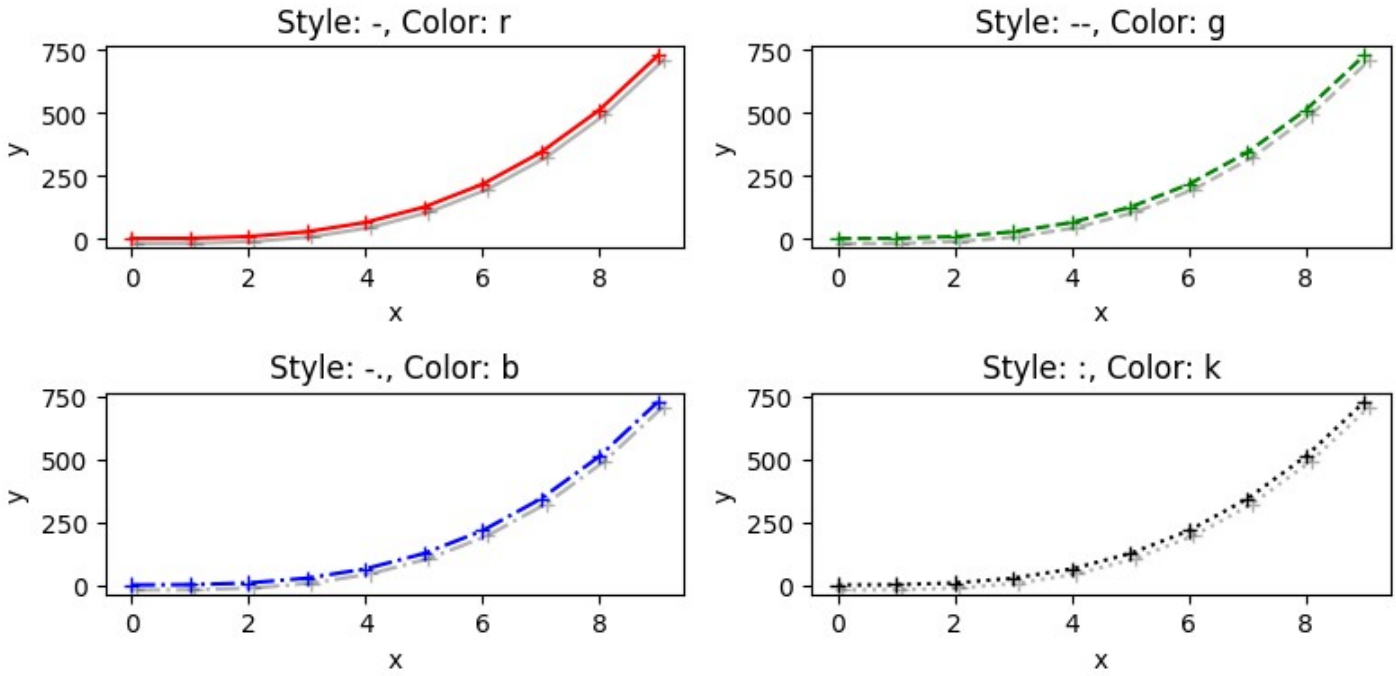



```
"11. Create a simple histogram plot with x = [0,1,2,3,4,5,6,7,8,9] and y = x³."
y=[i**3 for i in range(10)]; plt.figure(figsize=(8, 2))
plt.hist(y, bins=5, edgecolor='black', alpha=0.8)
plt.title('Histogram y = x³'); plt.xlabel('y'); plt.ylabel('Frequency')
plt.show()
```

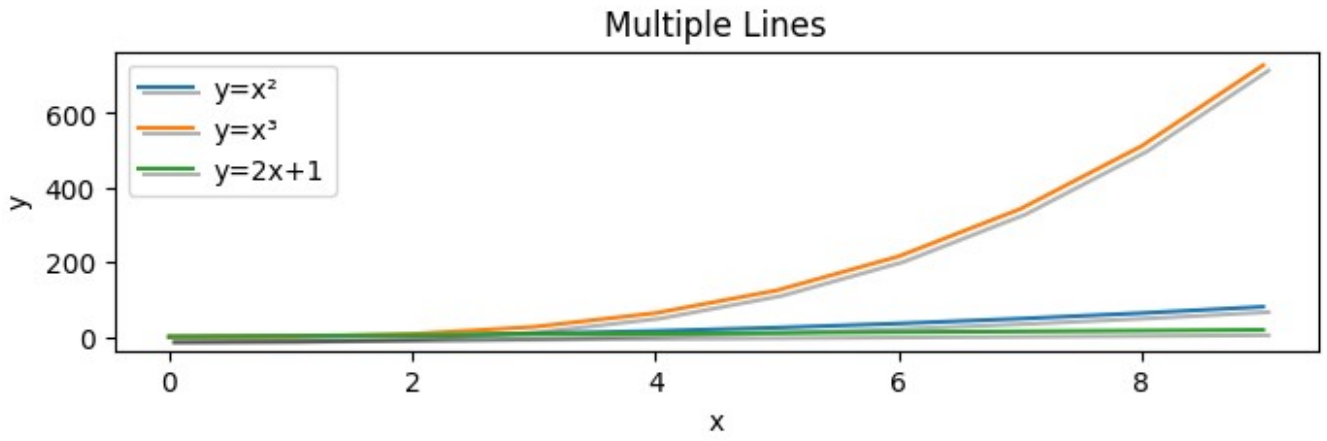



```
"12. Modify your previous plot(y = x³) to include the following: Adding Title
and Labels. Customizing the Line Style, markers, color (4 types of line
styles)."
x=range(10); y=[i**3 for i in x]
```

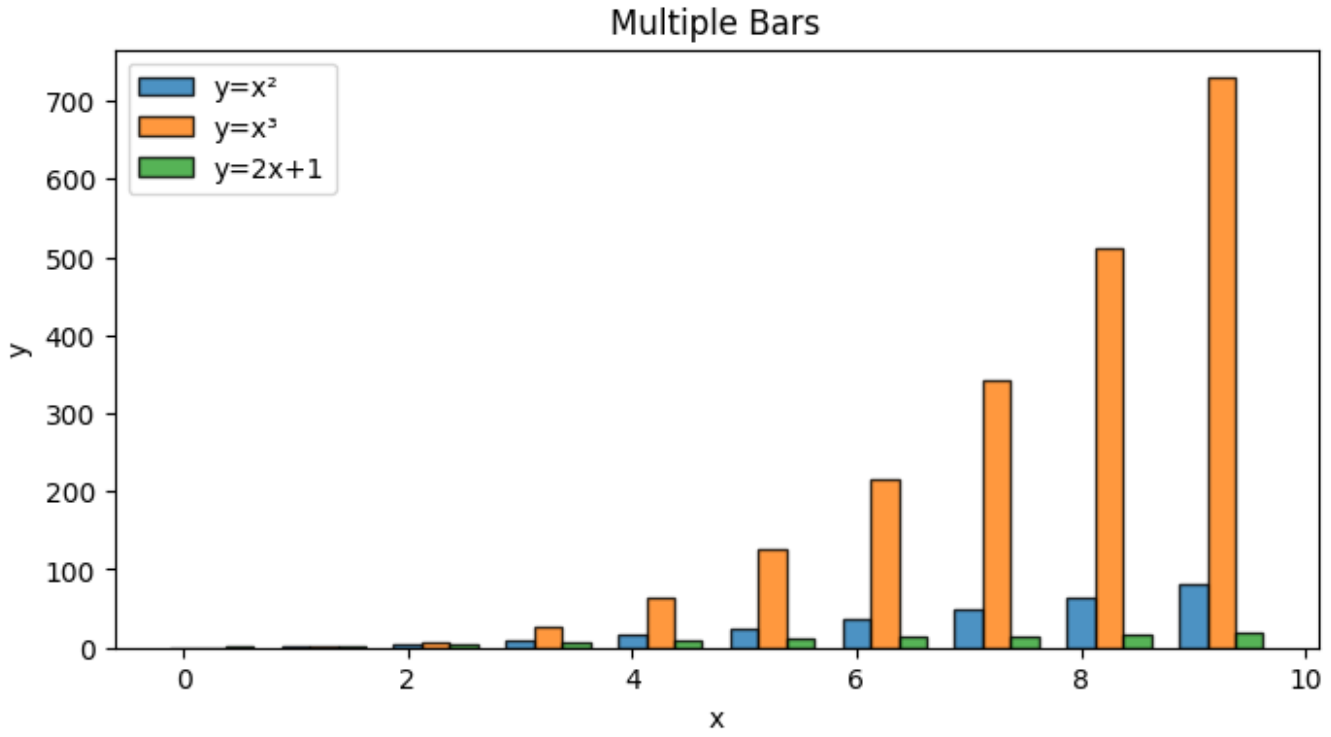
```
styles=['-','--','-.',':']; colors=['r','g','b','k']
fig, axs = plt.subplots(2, 2, figsize=(8, 4))
for ax, s, c in zip(axs.flatten(), styles, colors):
    ax.plot(x, y, linestyle=s, marker='+', color=c,
path_effects=[pe.SimpleLineShadow(), pe.Normal()])
    ax.set_title(f'Style: {s}, Color: {c}')
    ax.set_xlabel('x')
    ax.set_ylabel('y')
plt.tight_layout()
plt.show()
```



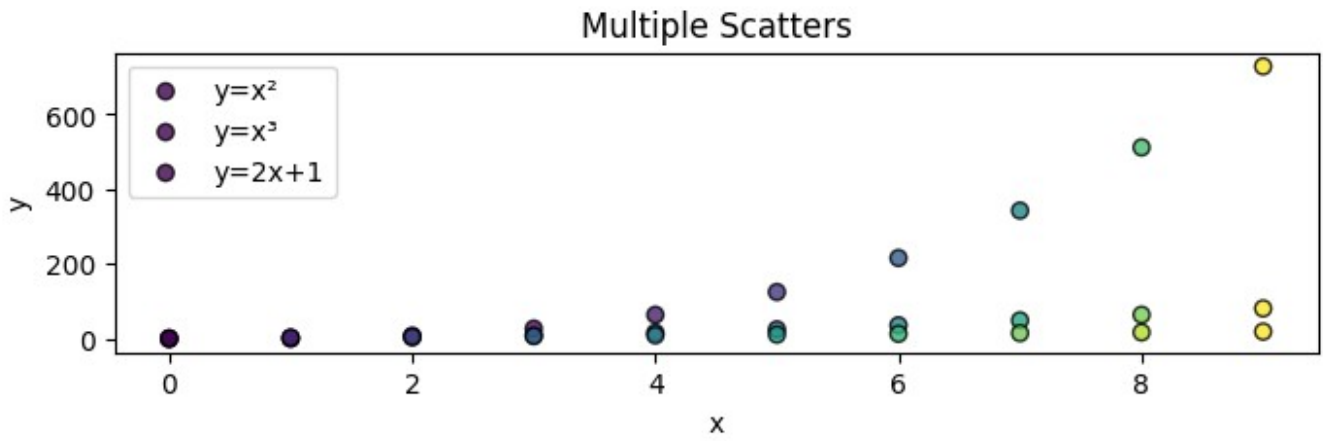

```
"13. Multiple Lines in One Plot y = x², y = x³, y = 2x + 1. Adding Title and Labels."
x=range(10); plt.figure(figsize=(8, 2))
plt.plot(x,[i**2 for i in x], label='y=x²', path_effects=[pe.SimpleLineShadow(),
pe.Normal()])
plt.plot(x,[i**3 for i in x], label='y=x³', path_effects=[pe.SimpleLineShadow(),
pe.Normal()])
plt.plot(x,[2*i+1 for i in x], label='y=2x+1',
path_effects=[pe.SimpleLineShadow(), pe.Normal()])
plt.title('Multiple Lines'); plt.xlabel('x'); plt.ylabel('y')
plt.legend(); plt.show()
```



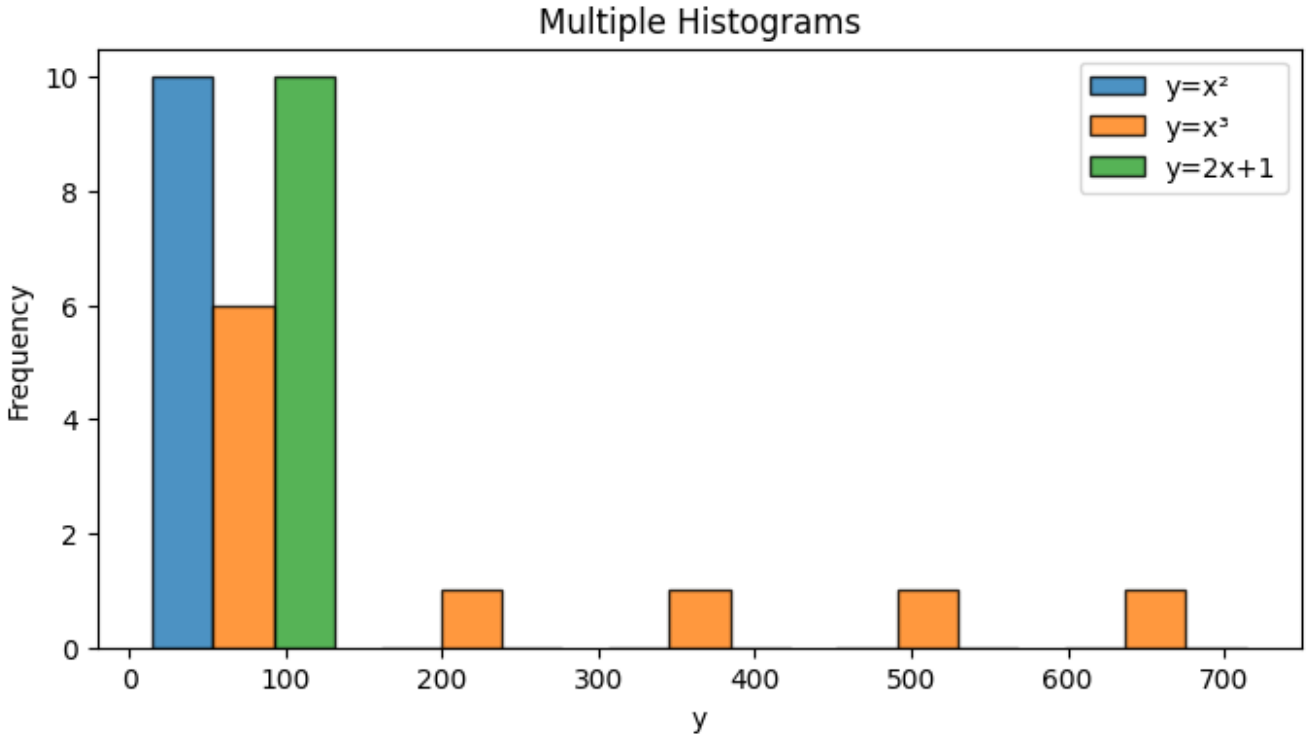
```
"14. Multiple bars in One Plot y = x², y = x³, y = 2x + 1. Adding Title and Labels."
import numpy as np; plt.figure(figsize=(8, 4))
x=np.arange(10); w=0.25
plt.bar(x, [i**2 for i in x], width=w, label='y=x²', edgecolor='black',
alpha=0.8)
plt.bar(x+w, [i**3 for i in x], width=w, label='y=x³', edgecolor='black',
alpha=0.8)
plt.bar(x+2*w, [2*i+1 for i in x], width=w, label='y=2x+1', edgecolor='black',
alpha=0.8)
plt.title('Multiple Bars'); plt.xlabel('x'); plt.ylabel('y')
plt.legend(); plt.show()
```




```
"15. Multiple scatters in One Plot y = x², y = x³, y = 2x + 1. Adding Title and Labels."
x=range(10); plt.figure(figsize=(8, 2))
plt.scatter(x,[i**2 for i in x], label='y=x²', c=[i**2 for i in x],
cmap='viridis', edgecolors='black', alpha=0.8)
plt.scatter(x,[i**3 for i in x], label='y=x³', c=[i**3 for i in x],
cmap='viridis', edgecolors='black', alpha=0.8)
plt.scatter(x,[2*i+1 for i in x], label='y=2x+1', c=[2*i+1 for i in x],
cmap='viridis', edgecolors='black', alpha=0.8)
plt.title('Multiple Scatters'); plt.xlabel('x'); plt.ylabel('y')
plt.legend(); plt.show()
```



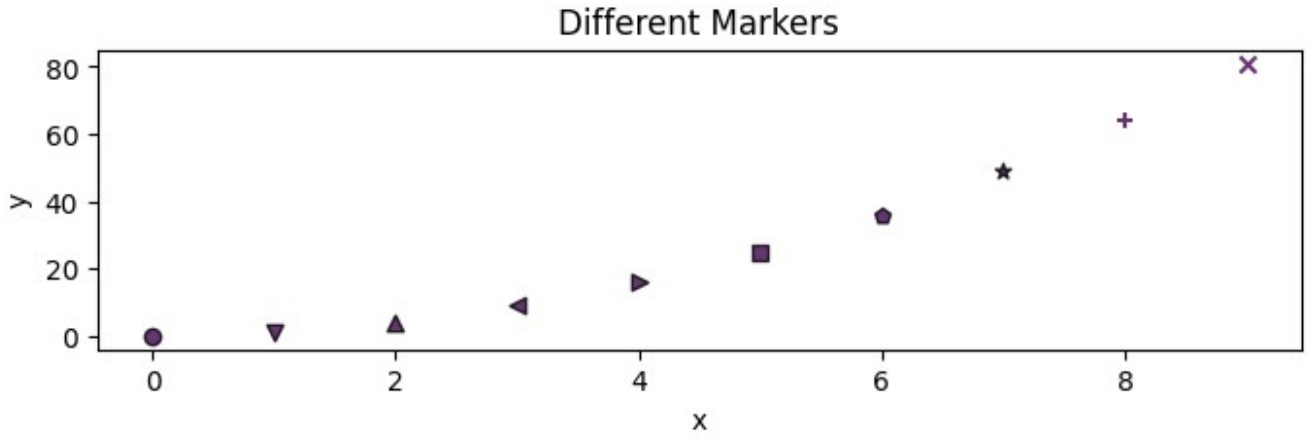

```
"16. Multiple histograms in One Plot y = x², y = x³, y = 2x + 1. Adding Title and Labels."
y1=[i**2 for i in range(10)]
y2=[i**3 for i in range(10)]
y3=[2*i+1 for i in range(10)]; plt.figure(figsize=(8, 4))
plt.hist([y1,y2,y3], bins=5, label=['y=x²','y=x³','y=2x+1'], edgecolor='black', alpha=0.8)
plt.title('Multiple Histograms'); plt.xlabel('y'); plt.ylabel('Frequency')
plt.legend(); plt.show()
```



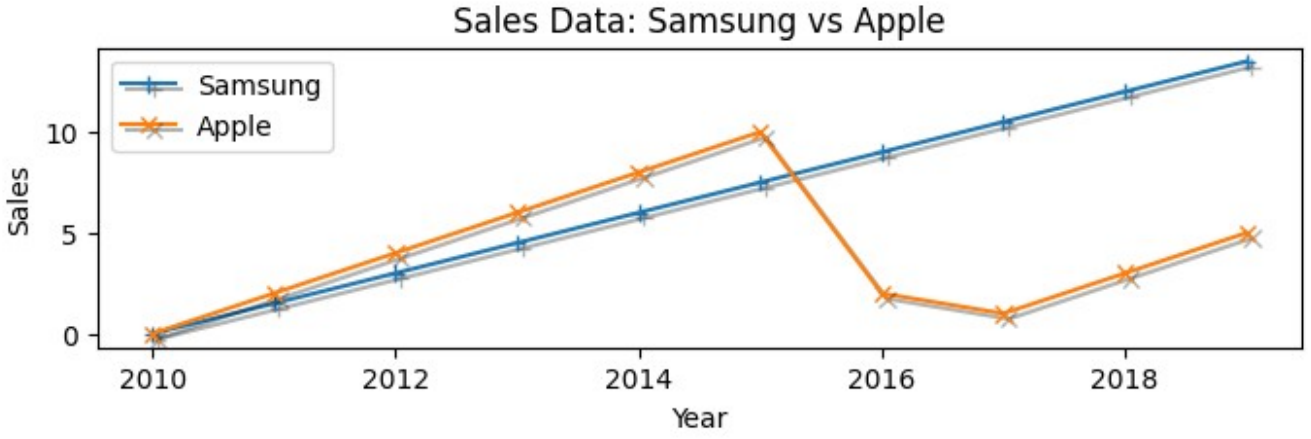

```
"17. A plot where each point has a different marker."
x=range(10); y=[i**2 for i in x]; plt.figure(figsize=(8, 2))
markers=['o','v','^','<','>','s','p','*','+','x']
for i in range(10):
    plt.scatter(x[i], y[i], marker=markers[i], c=[y[i]], cmap='viridis', edgecolors='black', alpha=0.8)
plt.title('Different Markers'); plt.xlabel('x'); plt.ylabel('y')
plt.show()
```

```
C:\Users\askt\AppData\Local\Temp\ipykernel_5124\3251998713.py:5: UserWarning: You passed a edgecolor/edgecolors ('black') for an unfilled marker ('+'). Matplotlib is ignoring the edgecolor in favor of the facecolor.  This behavior may change in the future.
```

```
  plt.scatter(x[i], y[i], marker=markers[i], c=[y[i]], cmap='viridis',
edgecolors='black', alpha=0.8)
C:\Users\askt\AppData\Local\Temp\ipykernel_5124\3251998713.py:5: UserWarning:
You passed a edgecolor/edgecolors ('black') for an unfilled marker ('x').
Matplotlib is ignoring the edgecolor in favor of the facecolor.  This behavior
may change in the future.
  plt.scatter(x[i], y[i], marker=markers[i], c=[y[i]], cmap='viridis',
edgecolors='black', alpha=0.8)
```

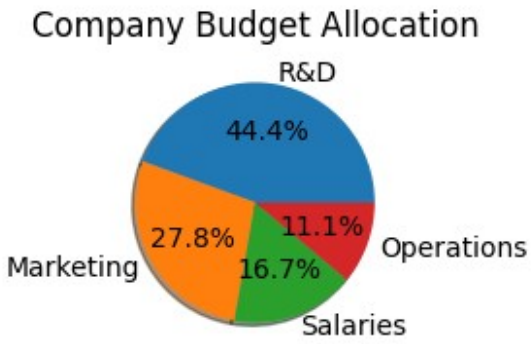



```
"18. A chart showing sales data for both (X is Samsung and Y is Apple) companies
side by side for last 10 years."
import matplotlib.pyplot as plt; import matplotlib.patheffects as pe
years = range(2010, 2020)
samsung = [i*1.5 for i in range(10)]
apple = [i*2 for i in range(5)] + [10, 2, 1, 3, 5]; plt.figure(figsize=(8, 2))
plt.plot(years, samsung, label='Samsung', marker='+',
path_effects=[pe.SimpleLineShadow(), pe.Normal()])
plt.plot(years, apple, label='Apple', marker='x',
path_effects=[pe.SimpleLineShadow(), pe.Normal()])
plt.title('Sales Data: Samsung vs Apple')
plt.xlabel('Year')
plt.ylabel('Sales')
plt.legend()
plt.show()
```




```
"19. A pie chart displaying the budget allocation across the categories, with a
title like 'Company Budget Allocation' and percentage labels for each slice."
labels=['R&D','Marketing','Salaries','Operations']
sizes=[40,25,15,10]; plt.figure(figsize=(2, 2))
```

```
plt.pie(sizes, labels=labels, autopct='%1.1f%%', shadow=True)
plt.title('Company Budget Allocation'); plt.show()
```




```
"20. Based on a survey of 100 people, create a pie chart showing their preferred programming languages."
import matplotlib.pyplot as plt; import matplotlib.patheffects as pe
langs=['Python', 'JavaScript', 'C--', 'Rust']; plt.figure(figsize=(3, 3))
sizes=[45, 25, 10, 18];
plt.pie(sizes, labels=langs, autopct='%1.1f%%', shadow=True)
plt.title('Preferred Programming Languages')
plt.show()
```