What's a python module?

- Python modules are pre-built chunks of code that allow you (and the programming community!) to easily re-use common code
- There are thousands of packages!
- If you have a computational need, someone has probably already written a module that does most of the work for you!

numpy – python module that adds significant mathematic functionality to python

scipy – add-on to numpy that adds more scientific functionality

matplotlib – charts, charts, and more charts!

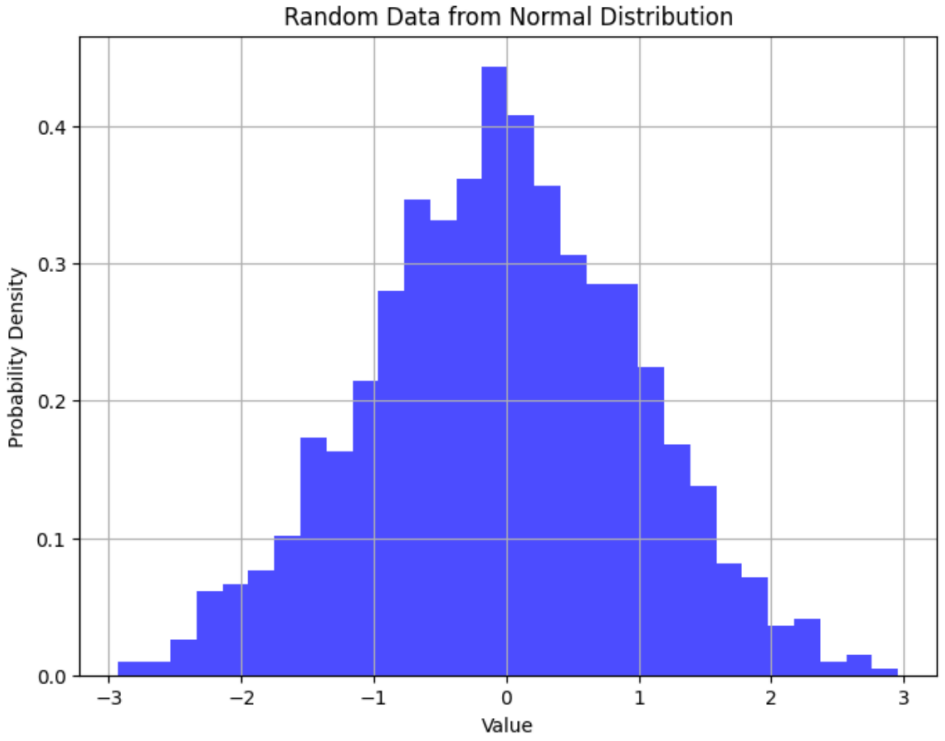
A Good Programmer is a Lazy Programmer

[1]:
```python
import numpy as np

# Parameters for the normal distribution
mean = 0      # Mean (center) of the distribution
std_dev = 1   # Standard deviation (spread or width) of the distribution
num_samples = 1000  # Number of samples to generate
```

```
[-8.05733011e-01 -4.29666753e-01  6.17691108e-01  4.23619392e-01
 -1.95801802e+00  5.09553791e-01  8.90686247e-01  2.35525981e+00
 -7.38639855e-01 -7.42922315e-01  7.31900250e-01  7.15959076e-01
  1.82481731e+00  9.96634458e-01  4.49518812e-01 -6.51242741e-01
 -5.75160408e-01 -1.90064209e-01 -9.60935665e-01 -1.15771868e+00
 -1.23360452e-01 -1.31442877e+00  3.85512162e-01 -6.40959085e-01
 -3.00165043e-01  4.95349298e-01  7.11888333e-01  8.94422379e-01
 -1.37443750e+00 -1.07846809e+00  3.74886317e-01 -3.47583538e-01
  9.87675590e-01  8.55212979e-01  3.61609433e-01 -2.31140170e+00
  8.53603709e-01 -2.00961702e+00 -1.43271239e+00 -1.33937295e+00
 -6.88571267e-01 -1.56989813e+00  7.11544446e-01 -7.18212371e-02
  1.55042101e-01  1.25411254e+00 -2.29935819e-01 -1.00645857e+00
 -1.57881151e-01 -1.09223444e+00 -8.23289815e-01  1.36600150e+00
 -7.93250140e-01  1.60073378e-02  9.58522882e-01  1.71888130e+00
 -1.12930159e+00 -3.22088561e-02 -1.39594821e-01 -2.04061943e+00
  2.56493484e+00 -2.25124906e+00 -1.59615779e+00  2.36227642e-01
  1.33032125e-02  4.75034296e-01 -1.23711091e+00  7.65410161e-01
 -3.85901405e-01 -1.65035637e-01  1.23045077e+00  1.80660378e+00
 -2.08037010e-01  6.00378704e-01  4.93477551e-01  6.26194464e-01
 -5.68070210e-01 -1.59618522e+00 -5.06686339e-01  1.95115103e-01
 -1.04269041e+00 -1.19371512e+00 -8.60496596e-01  1.39248721e+00
  1.26553455e+00 -4.23043986e-01 -1.35063231e+00  1.97297341e+00
 -2.16144466e+00 -1.72786853e+00  3.01535816e-01 -9.44680527e-01
```

How do we interpret/visualize this data?

```
[4]: import matplotlib.pyplot as plt

     # Plot the data using Matplotlib
     plt.figure(figsize=(8, 6))
     plt.hist(data, bins=30, density=True, color='blue', alpha=0.7)
     plt.title('Random Data from Normal Distribution')
     plt.xlabel('Value')
     plt.ylabel('Probability Density')
     plt.grid(True)
     plt.show()
```

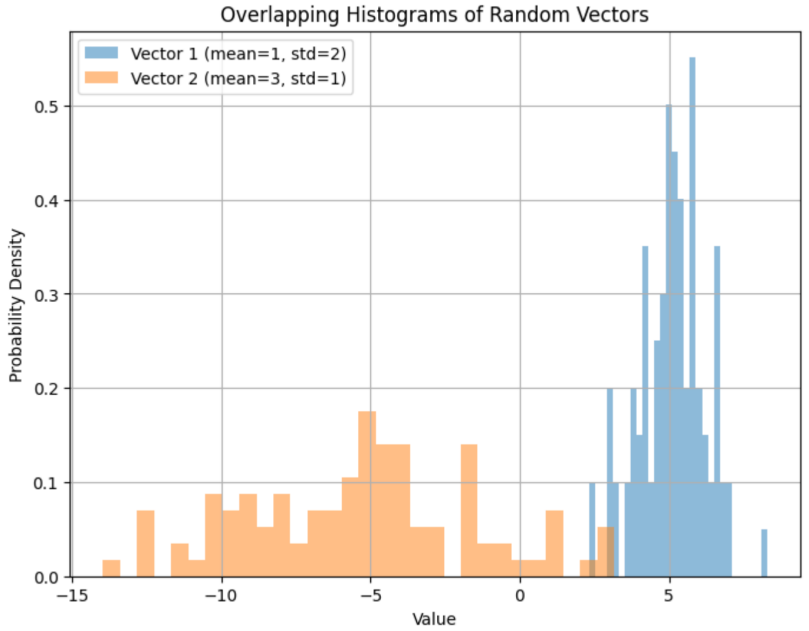


- **Try changing the mean and std_dev variables!**
- **Try changing the color ‘blue’!**
- **Change the xlabel!**

https://matplotlib.org/stable/gallery/index

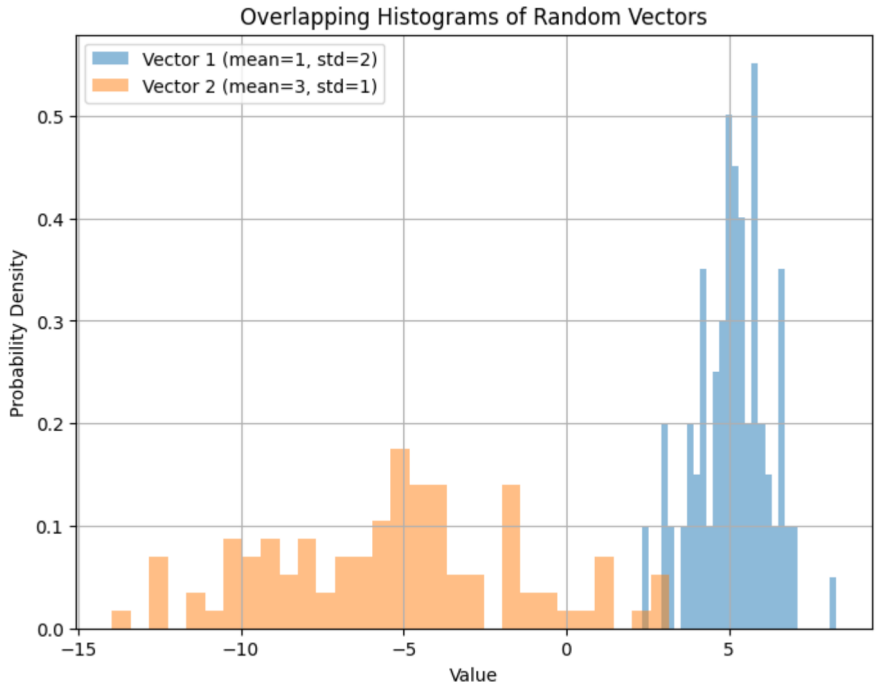
A little more realistic example

[13]:
```python
# Parameters for the normal distributions
mean1, std1 = 5, 1
mean2, std2 = -5, 4
num_samples = 100

# Generate random data vectors from the normal distributions
vector1 = np.random.normal(mean1, std1, num_samples)
vector2 = np.random.normal(mean2, std2, num_samples)

# Plot overlapping histograms
plt.figure(figsize=(8, 6))
plt.hist(vector1, bins=30, density=True, alpha=0.5, label='Vector 1 (mean=1, std=2)')
plt.hist(vector2, bins=30, density=True, alpha=0.5, label='Vector 2 (mean=3, std=1)')
plt.title('Overlapping Histograms of Random Vectors')
plt.xlabel('Value')
plt.ylabel('Probability Density')
plt.legend()
plt.grid(True)
plt.show()
```



What do you see?

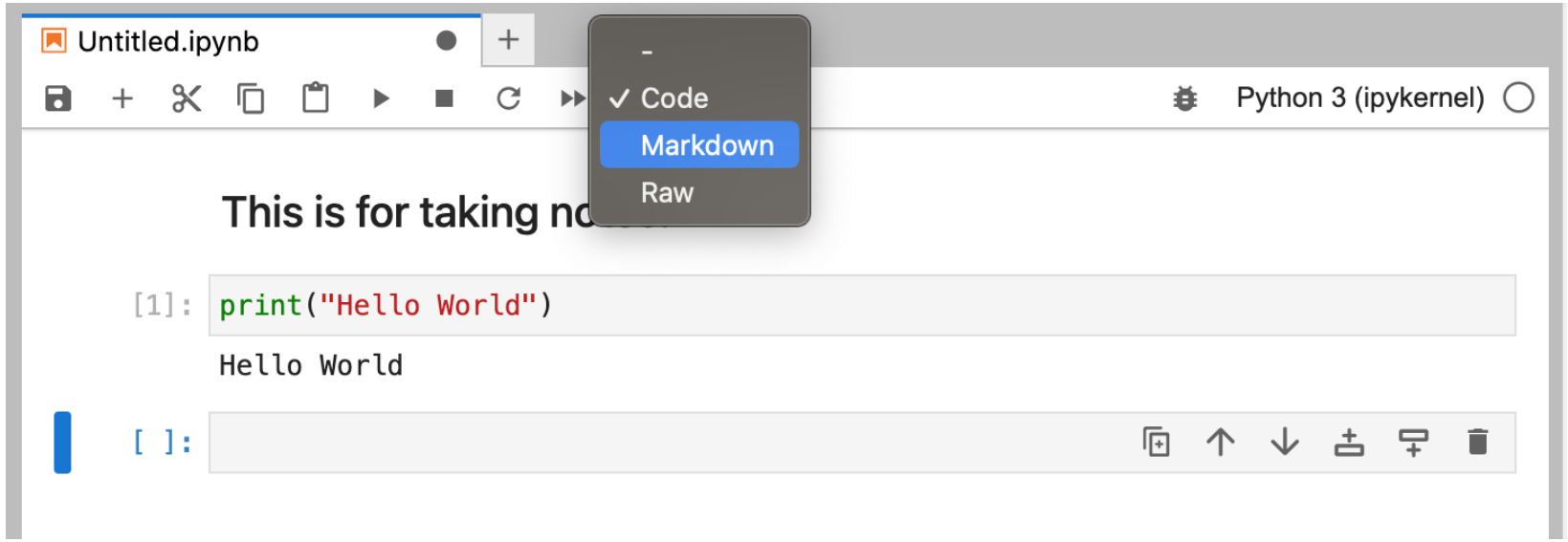
I performed an experiment and want to know if these two populations are significantly different….

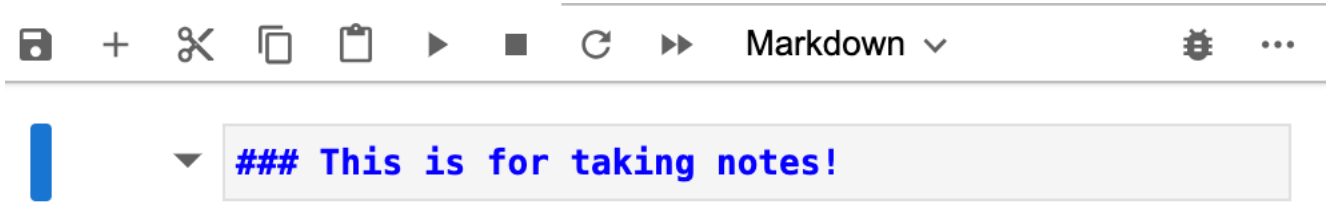


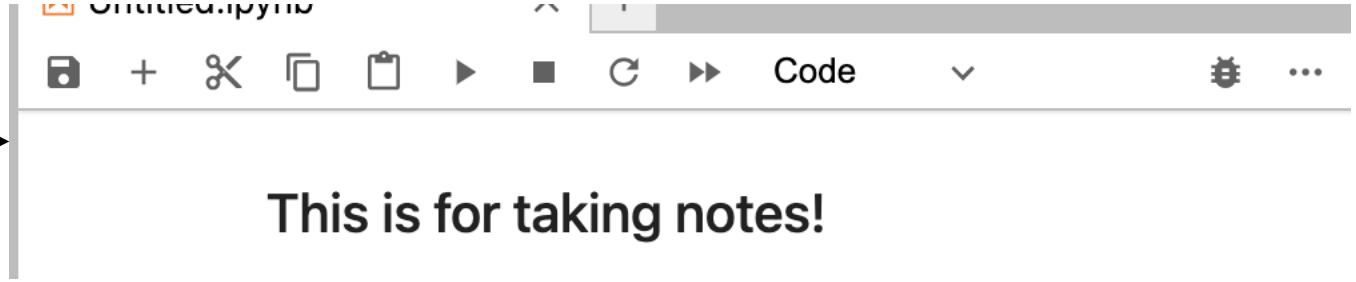
[15]:
```python
from scipy import stats

# Calculate t-test statistic
t_statistic, p_value = stats.ttest_ind(vector1, vector2)

print(f"T-Test Statistic: {t_statistic}")
print(f"P-Value: {p_value}")

if p_value < 0.05:
    print("The vectors likely come from different distributions.")
else:
    print("The vectors likely come from similar distributions.")
```

```
T-Test Statistic: 26.132811734308312
P-Value: 4.252753930108446e-66
The vectors likely come from different distributions.
```

Markdown mode lets you take notes and format them

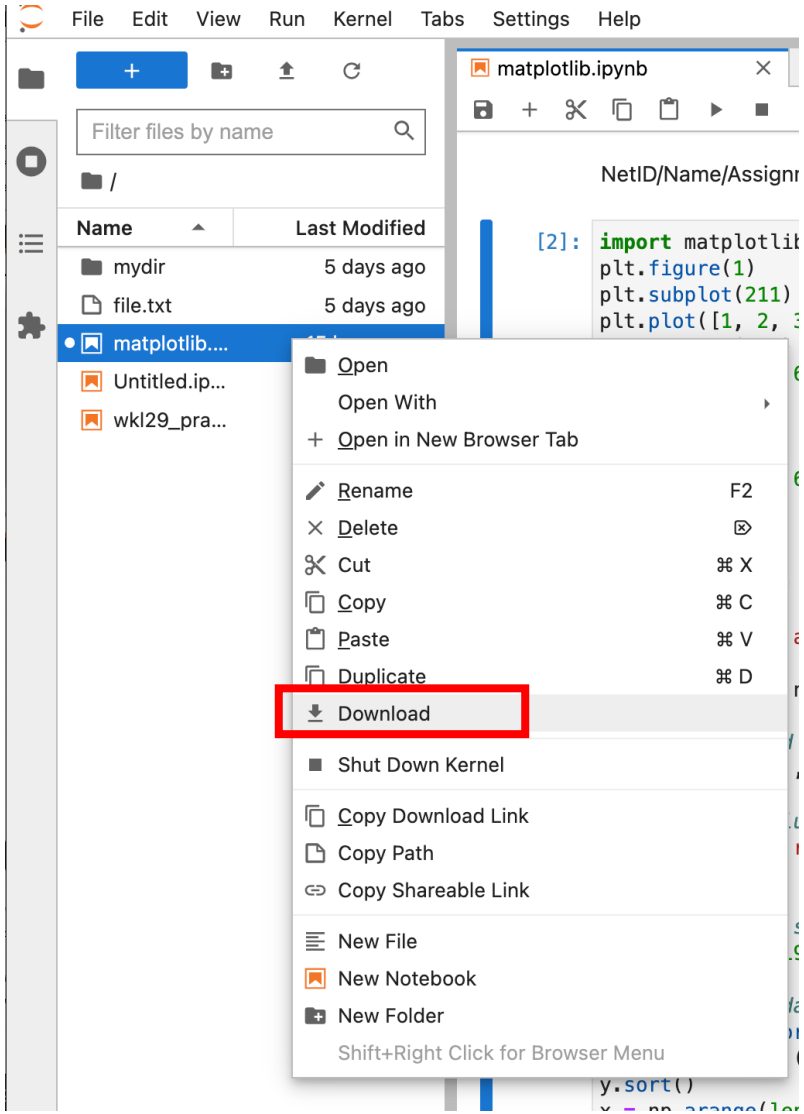




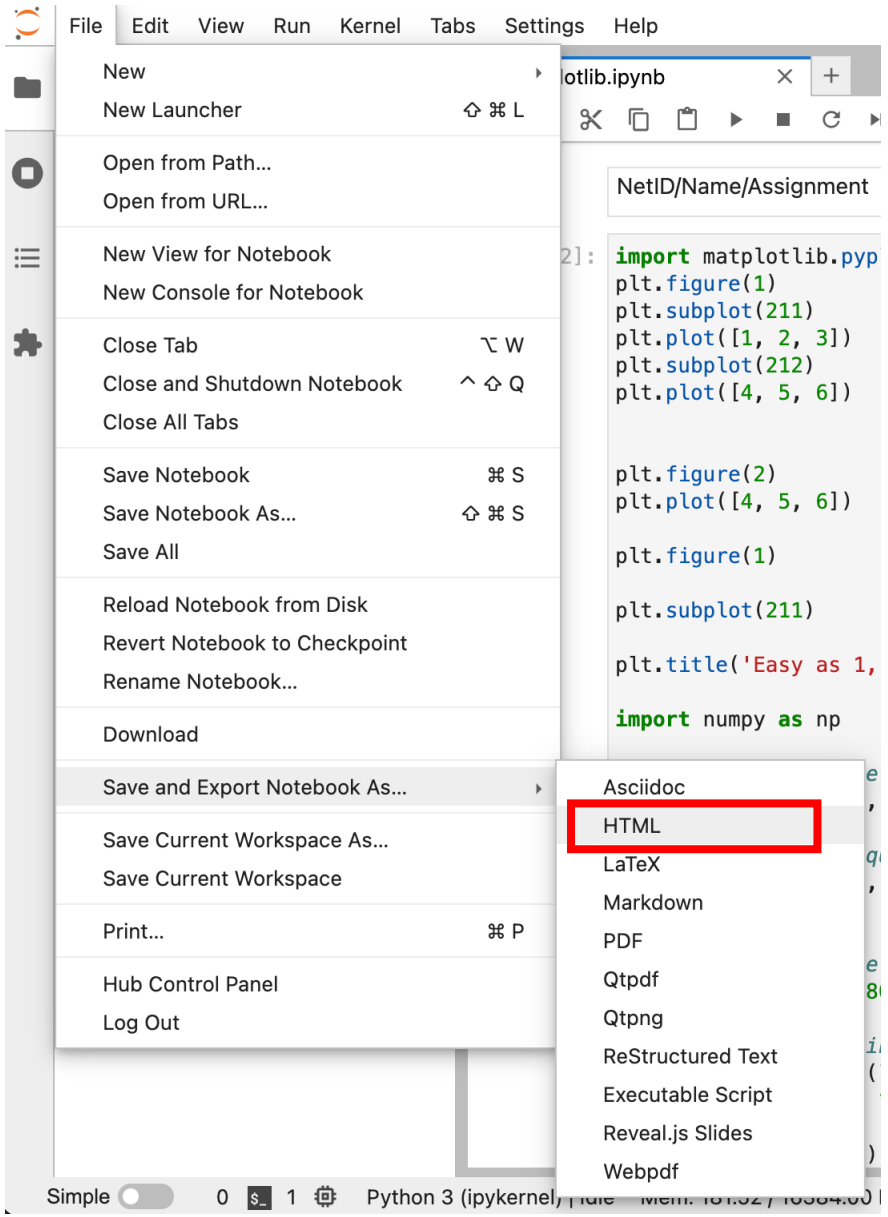


https://www.markdownguide.org/basic-syntax/

How to download *.ipynb



How to export as HTML



Practical #1:

Due: Monday 8/28 6pm
Upload to Canvas

File to send:
netID_practical1.ipynb
netID_practical1.html

1. Write a loop that calculates the sum of every odd number from 1 to 10 million

2. Generate a 2 numpy vectors of length 1,000 that randomly sample from the normal distribution
   - Vector 1: mean = 1, std = 2
   - Vector 2: mean = 3, std = 4

3. Show both vectors in an overlapping histogram

4. Calculate the t-test statistic to determine if the vectors are from unique distributions