In [1]:

```
### Put your NAME and EID here: Kushal Shah, KHS722
```

# Problem Set 0: Programming

In this first assignment, we want you to get familiar with jupyter notebooks as well as common Python packages that will be used in this class. If you need any help, refer to the documentation hints for the problems.

Also make sure you have the following packages installed for Python3:

- numpy
- matplotlib

In [3]:

```
# imports needed
import numpy as np
import matplotlib.pyplot as plt

# setting seed, DO NOT modify
np.random.seed(10)
```

## Problem 1

For each of the following distributions, complete both **part A** and **part B**.

1. Uniform: $[0, 1]$
2. Normal: $\mu = 0;\ \sigma^2 = 1$
3. Exponential: $\lambda = 2$

### Part A.

Generate a sample of **500 points** .

Useful modules:

```
- numpy.random (.uniform, .normal, .exponential)
```

In [5]:

```
uniform=np.random.uniform(0,1,500)
normal=np.random.normal(0,1,500)
exponential=np.random.exponential(2,500)
```

## Part B.

Plot separate histograms for each (with **bin size 10**).

Make sure to **title each** with the respective distribution (uniform, normal, exponential).

Useful modules:

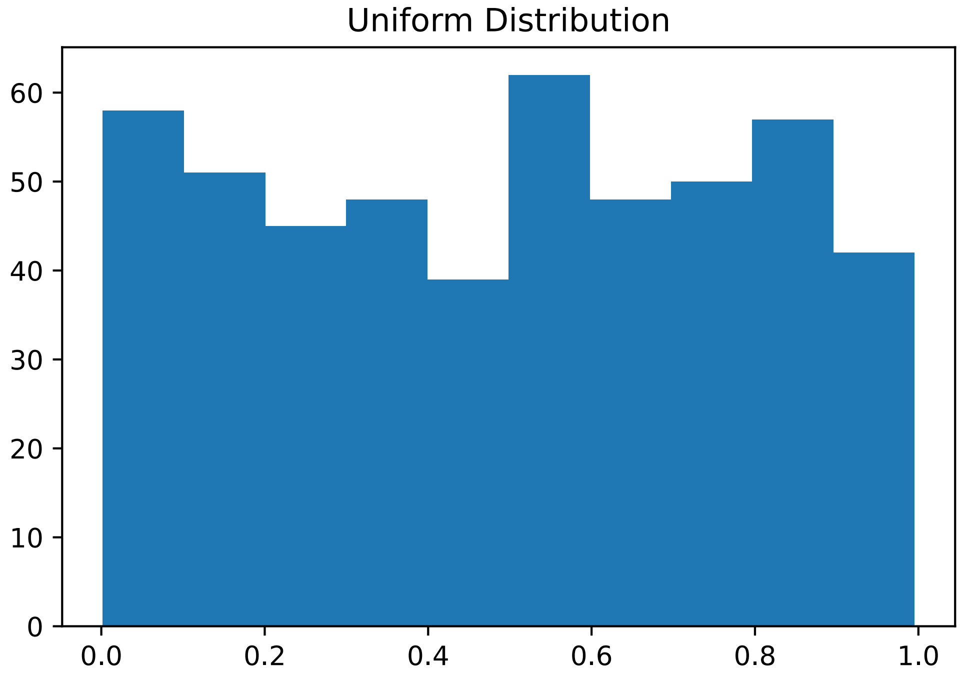
```
      - matplotlib.pyplot.hist
```

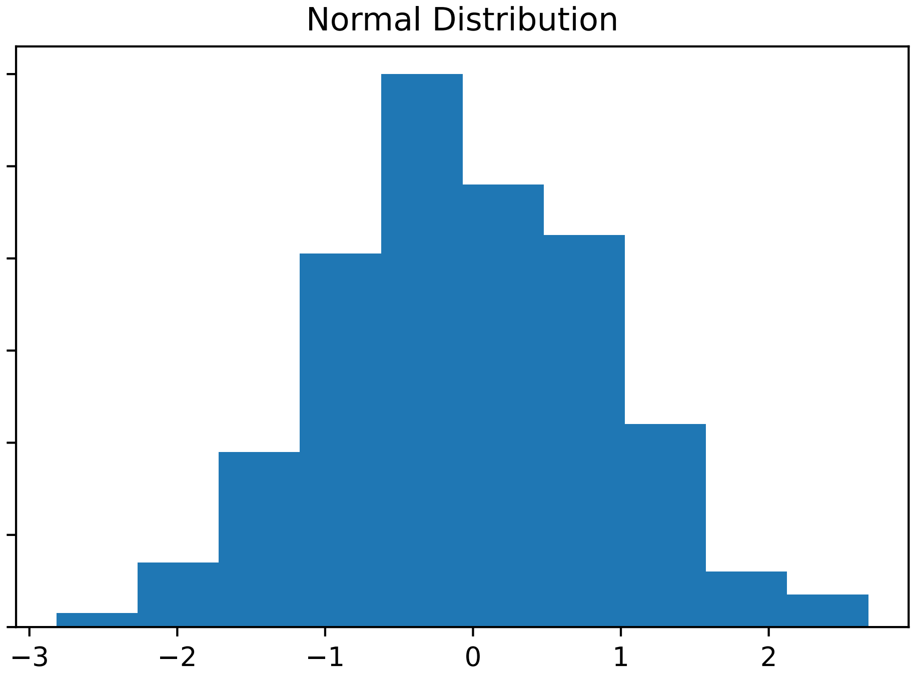
In [6]:

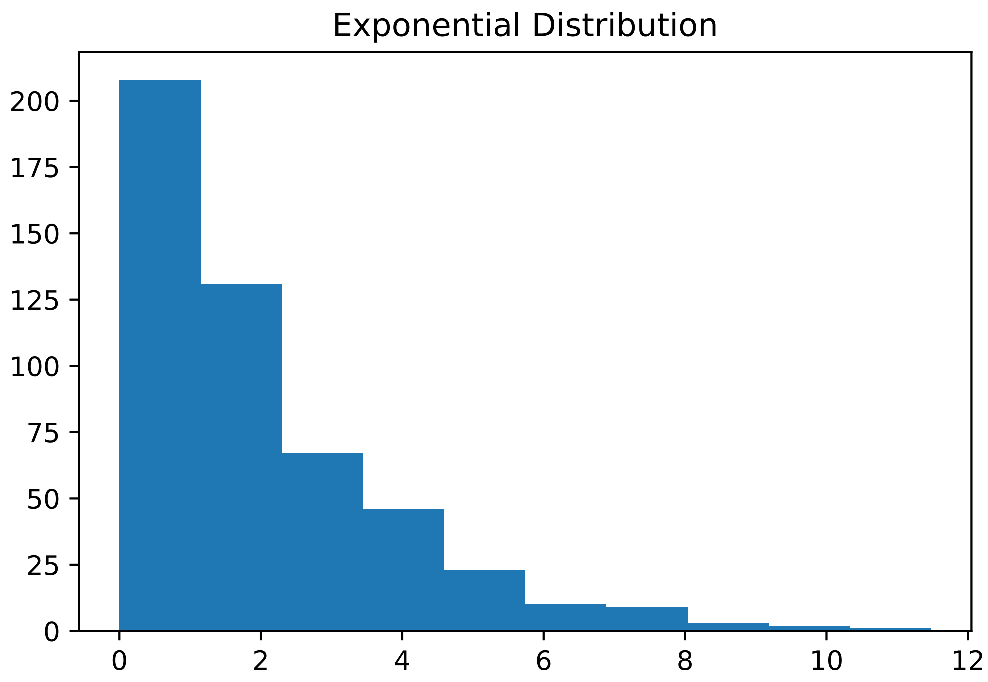
```
# code here
plt.hist(uniform,10)
plt.title("Uniform Distribution")
plt.show()

plt.hist(normal,10)
plt.title("Normal Distribution")
plt.show()

plt.hist(exponential,10)
plt.title("Exponential Distribution")
plt.show()
```


Uniform Distribution


Normal Distribution



# Problem 2

## Part A.

We will now explore a simple linear regression problem. Your first task will be to:

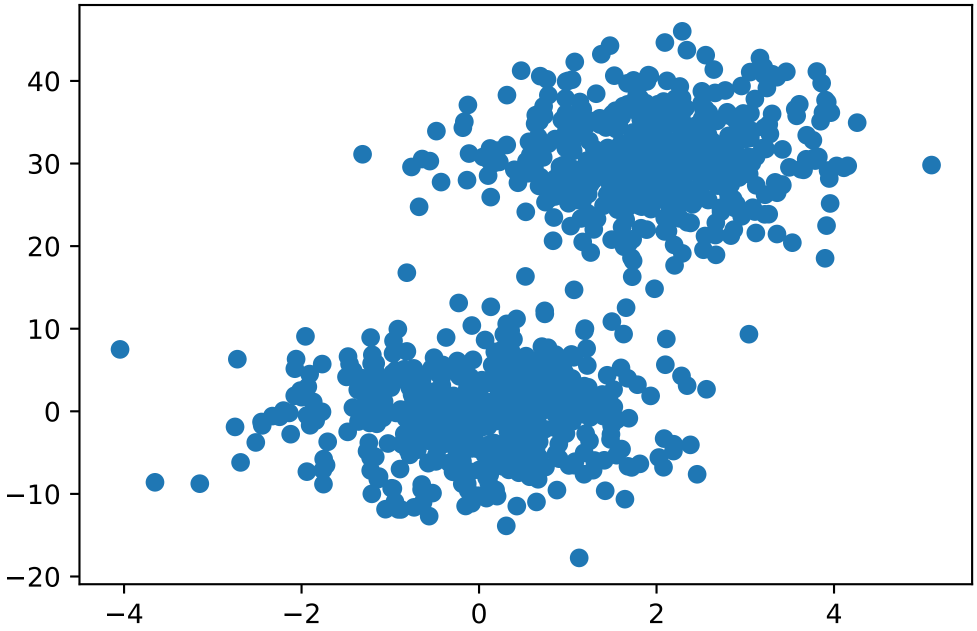
- Find the **ps0.data** file and load it in using numpy.
  - The array in this file should have **shape** of **(1000,101)**.
  - The **y-value** to be regressed to is the **final column**, and the **features** to be used will be found in the **first 100**.
- Plot the data points using the **2nd feature** (for the x-axis) and **y-value**. Make sure it is a **scatter plot.**
  - For the purposes of this assignment, we will start counting from **1.**
  - e.g. If **X** stores your features with shape (1000,100), then **X[0]** will be the **1st feature**.

Useful modules:

```
- numpy.load
- matplotlib.pyplot.scatter
```

In [32]:

```
# code here
data=np.load("ps0.data")
y=data[:,100]
x=np.delete(data, 100, axis=1)
plt.scatter(x[:,1],y)
plt.show()
```



## Part B.

Now, our goal is to solve the linear regression. Recall that if $X$ stores our features and we are trying to predict $y$, then we want to solve for $\beta$ using **least-squares**:

$$\min_{\beta} \sum_{i} ||y - X\beta||^2$$

Because $X$ has more data points than features, the following provides our solution:

$$\beta = (X^T X)^{-1} X^T y$$

- Code this up using numpy, to find the $\beta$ for each feature.
- If you explore $\beta$, you might notice that most elements are very small. Thus, find the **top 10 features** and their respective **values in** $\beta$. **Sort by decreasing value**.
  - The top 10 values should be determined by their magnitudes.
  - You can just print out the indeces of these features as well as their values.

Useful modules:

```
- numpy.linalg.inv
- numpy.transpose
- numpy.argsort
```

In [56]:

```
# code here

XT=np.transpose(x)
invProd=np.linalg.inv(np.matmul(XT,x))
beta=np.matmul(np.matmul(invProd,XT),y)
absBeta=np.abs(beta)
absBeta.sort()
for i in range(10):
    print(str(np.where(beta == absBeta[-i-1])[0])+":"+str(absBeta[-i-1]))
```

```
[52]:2.4945407930767733
[83]:2.1276631573921003
[48]:2.001950882742575
[8]:1.996255003763831
[30]:1.9370281599843318
[44]:1.5343148670430504
[78]:1.471915391701621
[69]:0.8366922337607553
[32]:0.8303539007240266
[1]:0.42840263044733107
```

# Turn in Instructions

Once you have completed Problems 1 and 2, please submit (for this part of the assignment):

- This .ipynb file.
- A PDF version of this file. To do this:
    1. Go to File -> Download as -> HTML
    2. Open the HTML and Print, and change the **destination** to **PDF**.