Technique assignment 2

Cogs 109 Spring 2020

Er Lin
A16140839

In [49]:
```
# load matplotlib for plotting
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as stats
```

Section 1

Question 1

Two six-sided dice are rolled and the two numbers showing are added to produce a sum between 2 and 12

a. Create a plot showing each event on the X axis and P(X), the probability of each event, on the Y axis. Include axis labels.

In [50]:
```
x_axis = np.arange(2, 13)
x_axis
```

Out[50]:
```
array([ 2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

In [51]:
```
# array to count how many times each number from 2 to 12 appears.
count = np.zeros(11, int)
count
```

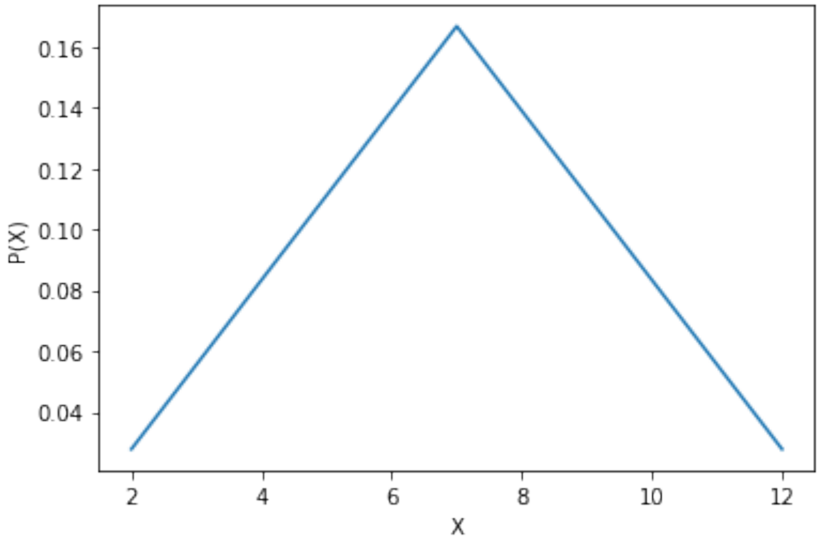
Out[51]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [52]:
```
# count how many times each sum occurs
for x in range(1, 7):
    for y in range(1, 7):
        count[x+y - 2] += 1
```

In [53]:
```
# get the probability
prob = count/36
```

In [54]:
```
# plot results
plt.plot(x_axis, prob)
plt.xlabel('X')
plt.ylabel('P(X)')
```

Out[54]: Text(0, 0.5, 'P(X)')



b. Does this distribution look more like a normal distribution or a uniform distribution? Why?

This looks more like a normal distribution we can roughly see the bell shaped curve with x = 7 being the center and having the highest probability and the events at both corners x = 2 and x = 12 with the least probability. Thus resembling a normal distribution.

c. What is P(X = 8 or 9)?

P(x=8 or x=9) = P(x=8) + P(x=9) = 0.25

In [55]:
```
prob[8-2] + prob[9-2]
```

Out[55]: 0.25

Question 2

For this problem, we will consider two normal (Gaussian) distributions: the first is characterized by μ=10, σ=3. The second is characterized by μ=10, σ=1. Hint: use the scipy.stats module and norm

a. Plot both distributions in the same graph, including axis labels and a legend to distinguish the two distributions.

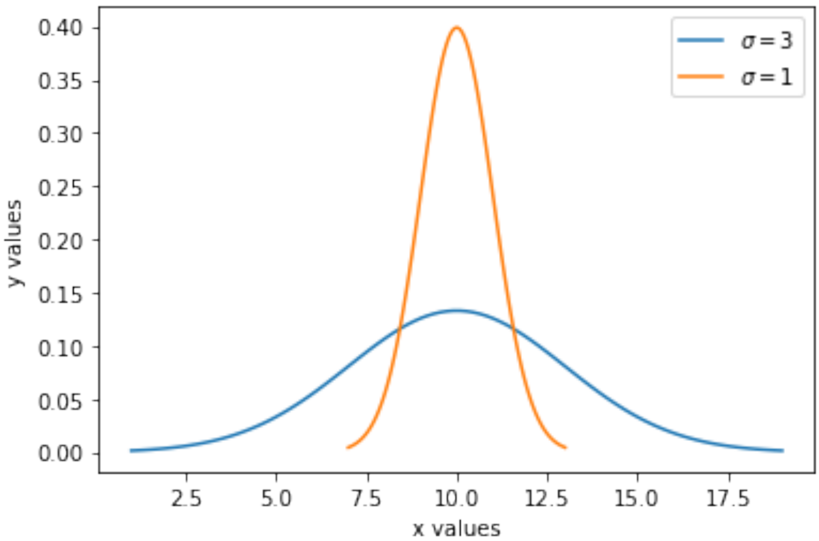
In [56]:
```
# initialize variables
mu = 10
sigma1 = 3
sigma2 = 1

# initialize x and y values for first distribution
x1 = np.linspace(mu - 3 * sigma1, mu + 3 * sigma1, 100)
y1 = stats.norm.pdf(x1, mu, sigma1)

# initialize x and y values for second distribution
x2 = np.linspace(mu - 3 * sigma2, mu + 3 * sigma2, 100)
y2 = stats.norm.pdf(x2, mu, sigma2)

# plot the values
plt.plot(x1, y1, label='$\sigma=3$')
plt.plot(x2, y2, label='$\sigma=1$')
plt.legend()# show legend with the given labels

# label the axis
plt.xlabel('x values')
plt.ylabel('y values')
plt.show()
```



b. Calculate p(x>11) for each distribution. Hint: the integral of the pdf is called a cdf.

In [57]:
```
# calculate the cdf for each distribution
# we subtract for one because we want to find greater than 11
cdf_1 = 1 - stats.norm(mu, sigma1).cdf(11)
cdf_2 = 1 - stats.norm(mu, sigma2).cdf(11)

print("CDF for σ=3: ",p_cdf_1)
print("CDF for σ=1: ",p_cdf_2)
```

```
CDF for σ=3:  0.36944134018176367
CDF for σ=1:  0.15865525393145707
```

c. Calculate p(x>11) for each distribution using the array(s) you created in Python. It should be close to your answer in 2b, but may not match exactly.

In [58]:
```
dist_1 = 0
dist_2 = 0

# calculate for the first distribution
for i in range(len(x1)-1):
    if x1[i] > 11.0:
        dist_1 += (x1[i+1] - x1[i]) * y1[i]

# calculate for the second distribution
for i in range(len(x2)-1):
    if x2[i] > 11.0:
        dist_2 += (x2[i+1] - x2[i]) * y2[i]

print("P(x>11) for σ=3:", dist_1)
print("P(x>11) for σ=1:",dist_2)
```

```
P(x>11) for σ=3: 0.3565557315499524
P(x>11) for σ=1: 0.14990861729440258
```

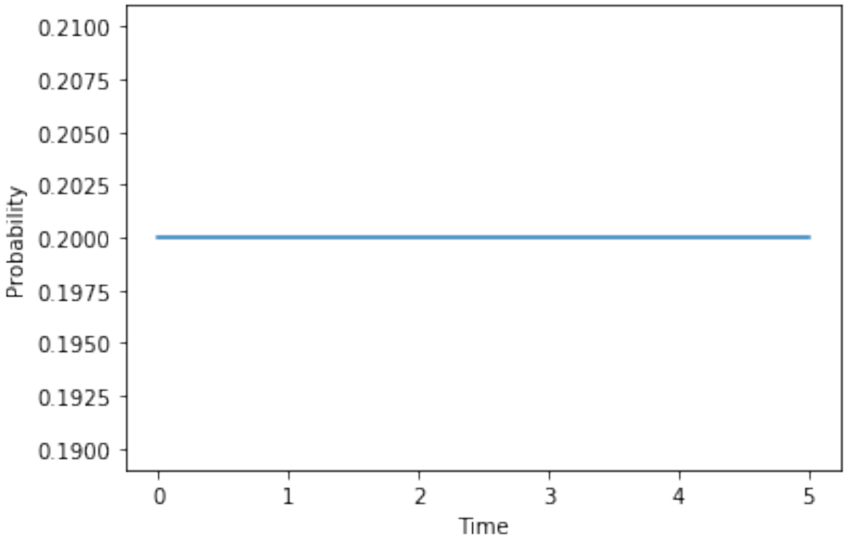
Question 3

Suppose the local news tells you to expect a small earthquake tonight between midnight and 5am. You assume that the earthquake is equally likely to occur at any time during those five hours.

a. Create a graph showing time on the X axis (x=0-5, minutes not necessary) and P(x) on the Y axis.

In [63]:
```
x = np.arange(0, 6)
y = np.repeat(0.2, 6)
plt.plot(x, y)
plt.xlabel('Time')
plt.ylabel('Probability')
```

Out[63]: Text(0, 0.5, 'Probability')



b. What is E[X]?

E[X] = 0.2 since it is equally likely to occur at any time

c. If you will be playing video games from midnight to 1:30am, what is the probability that the earthquake will happen during your gaming time?

(Total time playing) / (Time that earthquake can occur)

In [64]:
```
1.5 / 5
```

Out[64]: 0.3

Section 2

Question 4

Let's imagine flipping a penny 4 times.

a. What is the probability of getting HHHH?

We have 16 different permutations for flippign a penny 4 times.
HHHH is only one out of the 16, Therfore 1/16.
(1/2)^4 = 1/16 = 0.0625

In [67]:
```
1/16
```

Out[67]:
```
0.0625
```

b. What is the probability of getting HTHT?

This is the same as above: 1/16.
Same explanation.

c. If event E is getting H on the first throw and event F is getting H on the second throw, what is P(E intersection F)?

P(E intersection F) = 1/2 * 1/2 = 1/4

d. What is P(E union F)?

P(E union F) = P(E) + P(F) - P(E intersection F)
P(E union F) = 1/2 + 1/2 - 1/4

e. What is the probability of getting at least one H?

P(one H) = 1 - P(all H)
P(one H) = 1 - 1/16 = 15/16

Two events are independent if knowing something about one event does not change what you know about the other event (that is: P(AÇB) = P(A)P(B); P(A|B) = P(A); P(B|A) = P(B))

Question 5

Are the following pairs of events dependent or independent? A one-word answer is sufficient, but feel free to include an explanation if it helps you.

a. Independent

b. Dependent

c. Independent

d. Dependent

e. Independent

Question 6 | Contingency table

The contingency table shows how many students in each grade level of a high school are members of the school band or the school choir. Students can be in both. Please give the answer to each question as a decimal, percentage, or simplified fraction.

a. P(Band or Choir) = P(Band) + P(Choir) = 320/940 + 195/940 = 515/940 = 0.5479
b. P(Band | Freshman) = 100/225 = 4/9 = 0.4444
c. P(Junior | Neither) = 160/490 = 16/49 = 0.3265
b. P(Choir | Senior) = 60/235 = 12/47 = 0.2553

Question 7 | Bayes' rule

There is a disease that affects 0.5% of a population. A test for the disease gives a positive result for 99% of people who have the disease. It also gives a positive result for 1% of people who do not have the disease (false positive). If a randomly selected person is tested and receives a positive result, what is the probability that they have the disease?

P(dis | positive) = P(positive | dis) *P(dis) / P(positive)*
P(dis) = 0.005
P(positive | dis) = 0.99
P(positive) = P(positive | dis)P(dis) + P(positive | noDis)P(noDis)
P(positive) = 0.99 0.005 + 0.01 * 0.955 = 0.0145

P(dis | positive) = 0.99 * 0.005 / 0.0145 = 0.3414

Question 8 | Conditional probability

You have two coins in your pocket. One is a regular coin and the other is a weighted coin that has a 75% chance of landing heads up. You can’t tell the coins apart by inspecting them.

You take a coin out of your pocket and toss it. It lands heads up.

a. What is the probability that the coin is the fair coin?

P(isFair | H) = P(H | isFair) *P(isFair) / P(H)*
P(H | isFair) = 1/2
P(isFair) = 1/2
P(H) = P(H | isFair)P(isFair) + P(H | notFair)P(notFair)
P(H) = 1/2 1/2 + 3/4 * 1/2 = 5/8

P(isFair | H) = (1/2 * 1/2) / (5/8) = 2/5

b. How many times would you decide to flip the coin before you are pretty sure it’s the fair coin? Explain your answer.

2 times.
P(isFair | H) = 2/5
P(H | isFair) = 1/2
P(Heads in 'n' tosses) = (1/2)^n

(1/2)^n = 2/5
After calculations n = 1.32 which means that to make sure we should flip 2 times.

Question 9 | Extra Credit

In [69]:
```python
import pandas as pd
```

In [100]:
```python
df = pd.DataFrame({'Year': ['Freshman', 'Sophmore', 'Junior', 'Seni
or', 'Alumni', 'Total'],
                   'Java': [20, 50, 100, 250 , 80, 500],
                   'Python': [100, 250, 120, 60, 270, 800],
                   'C++': [60, 100, 200, 80, 100, 540]
                  }).set_index('Year')
```

In [101]:
```
df.assign(Total=df.apply(sum, axis=1))
```

Out[101]:

	Java	Python	C++	Total
Year				
Freshman	20	100	60	180
Sophmore	50	250	100	400
Junior	100	120	200	420
Senior	250	60	80	390
Alumni	80	270	100	450
Total	500	800	540	1840

The contingency table above shows the number of students in each year and the the count for their favorite programming language.

a. Probability tthat a freshman's favorite programming language is Python.
P(Python | Freshman) = 100 / 180 = 5/9

b. Probability someone's who's favorite programming language is C++, is an Senior.
P(Senior | C++) = 80/540 = 4/27 = 0.1481