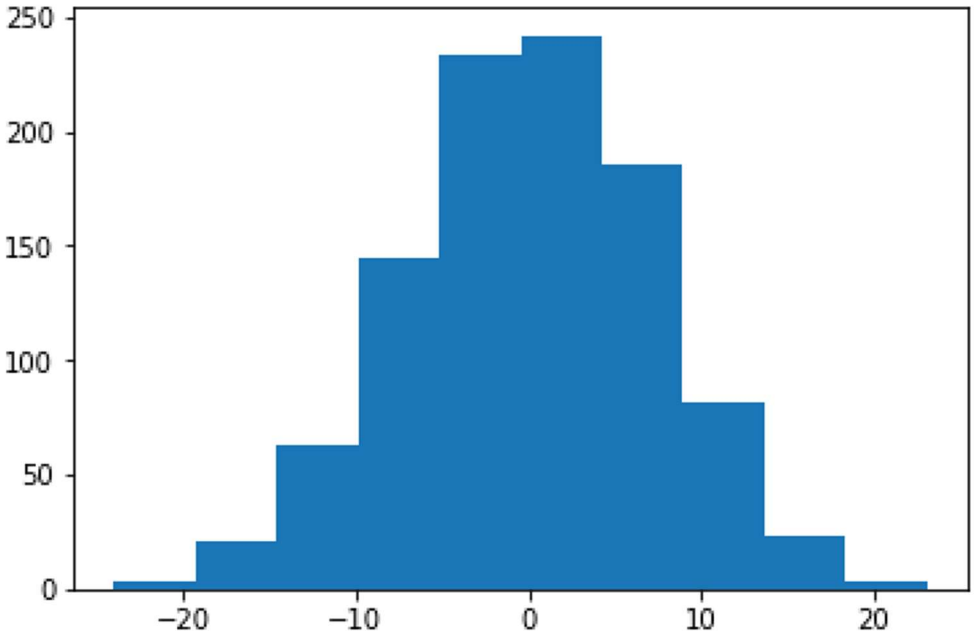
Lab 1 Programming

Caleb Johnson cdj2273
Aimun Khan aak2629

1.

```
import numpy
import matplotlib

array1 = numpy.random.normal(-10,5,1000)
array2 = numpy.random.normal(10,5,1000)
array3 = []

for x in range (0, len(array1)):
   array3.append(array1[x] + array2[x])

matplotlib.pyplot.hist(array3)
```



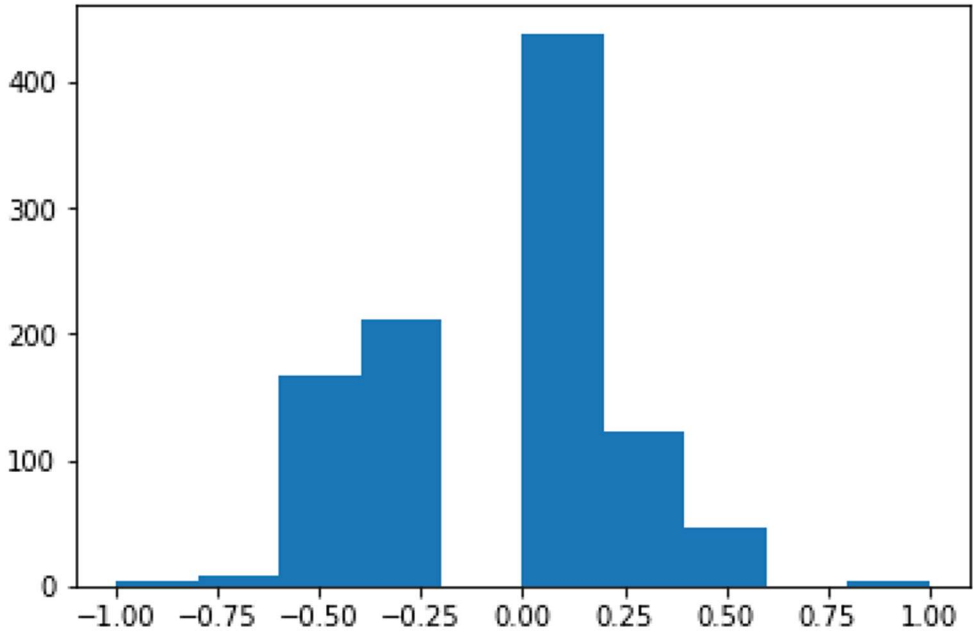
The mean of the sum approaches 0
The variance of the sum is the sum of the variances, i.e. 5 + 5 = 10

2.

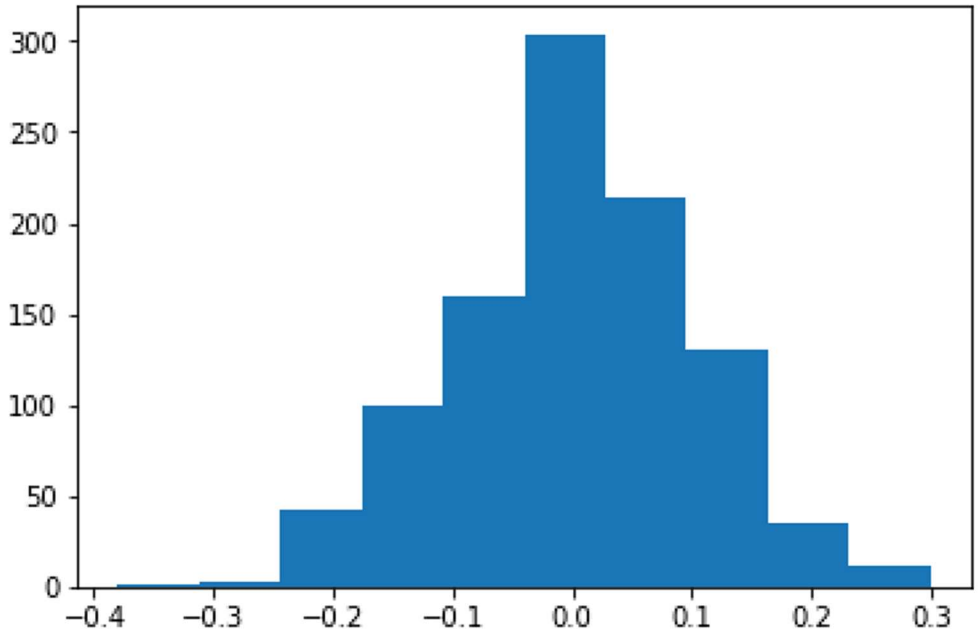
```
import numpy
import matplotlib

n = 1000
totalsum = []
```

```
for number in range(1000):
    sum = 0
    array = numpy.random.binomial(1, .5, n)
    for i in range(len(array)):
        if array[i] == 0:
            array[i] = -1
        sum += array[i]
    sum /= n
    totalsum.append(sum)

matplotlib.pyplot.hist(totalsum)
```

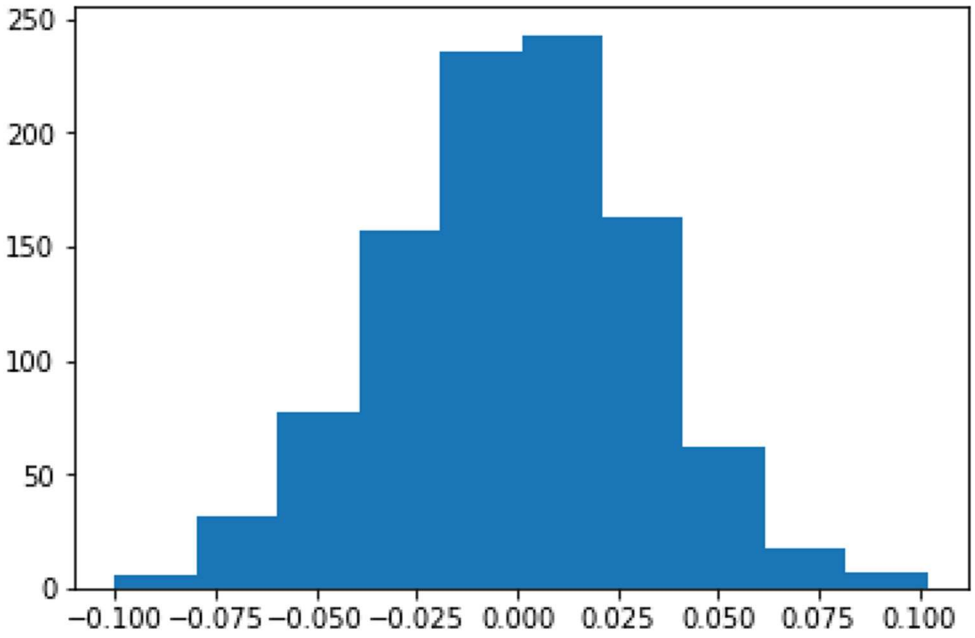
n=10



N=100



N=1000

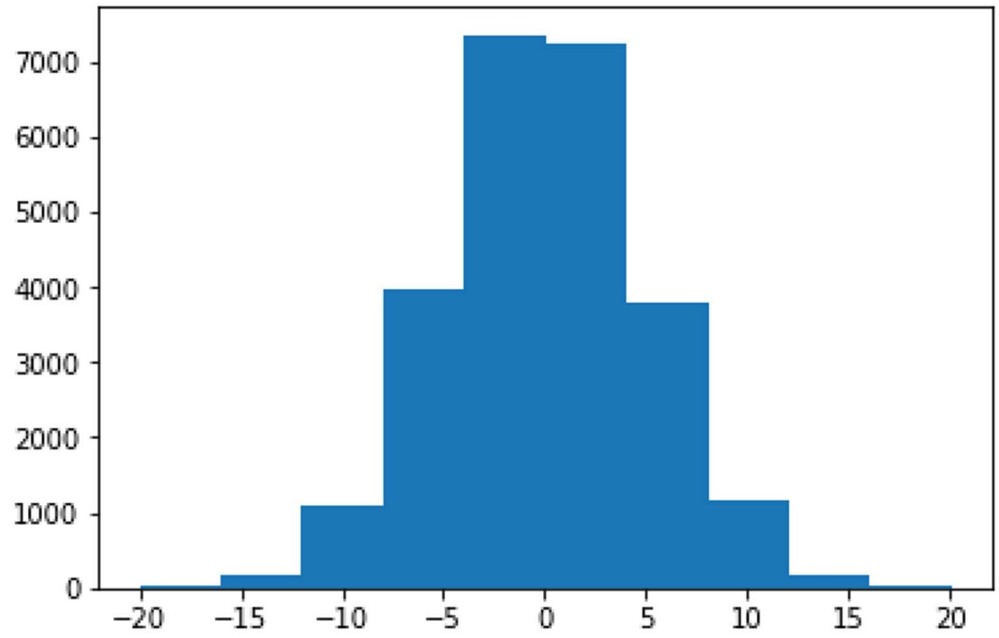


3.

```
import numpy
import matplotlib
import math

array = numpy.random.normal(0,5,25000)
```

```
sum = 0
for value in array:
   sum += value
mean = sum/len(array)

newarray = []
for value in array:
   newarray.append(value)

sum = 0
for i in range(len(array)):
   newarray[i] = (newarray[i] - mean) ** 2
   sum += newarray[i]
std = math.sqrt(sum/len(array))


print("Mean: " + str(mean))
print("Std:" + str(std))

matplotlib.pyplot.hist(array)
```

```
Mean: 0.011250622596999072
Std:4.965503365100472
```



4.

```
import numpy

mean = [-5, 5]
```

```
cov = [[20, .8], [.8, 30]]

array = numpy.random.multivariate_normal(mean, cov, 10000)

sum1 = 0
sum2 = 0
for value in array:
   sum1 += value[0]
   sum2 += value[1]

mean1 = sum1/10000
mean2 = sum2/10000

corrx = 0
corrxy = 0
corry = 0
for value in array:
   corrx += (value[0] - mean1) ** 2
   corrxy += ((value[0] - mean1) * (value[1] - mean2))
   corry += (value[1] - mean2) ** 2
corrx /= 9999
corrxy /= 9999
corry /= 9999

covariance = [[corrx, corrxy], [corrxy, corry]]

print (mean1)
print (mean2)
print (covariance)
```

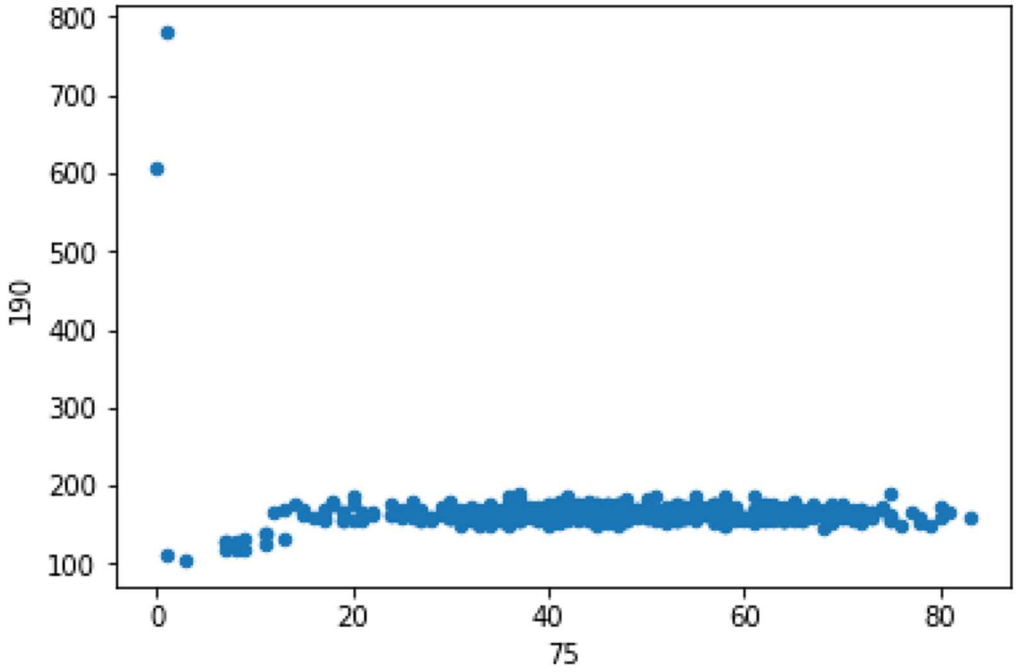
Mean: [-5.083717188375436, 5.039420631004392]
Covariance: [[19.715769965968118, 0.8180022059634413], [0.8180022059634413, 29.9631276343814]]

5.

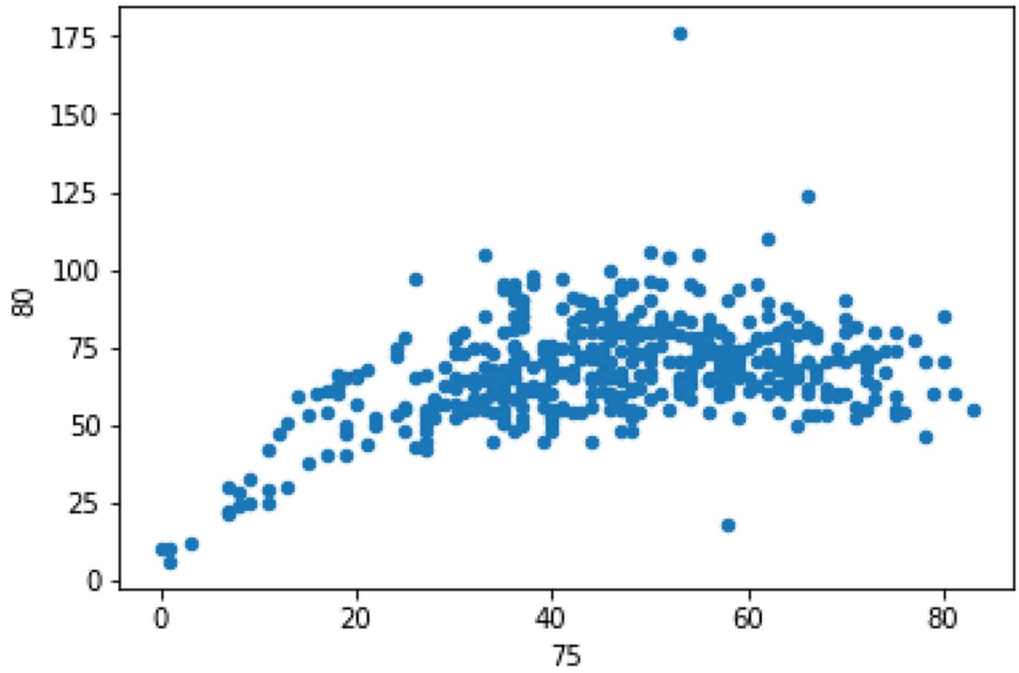
```
import pandas

patient_data = pandas.read_csv('PatientData.csv')

print(patient_data)
patient_data.info()
patient_data.plot.scatter(0,2)
patient_data.plot.scatter(0,3)
patient_data.fillna(patient_data.mean())
print(patient_data.corr().sort_values('8'))
```

a. 451 patients, 279 features
b. Age, sex, height, weight

patient_data.plot.scatter(0,2):



patient_data.plot.scatter(0,3):



c. patient_data.fillna(patient_data.mean())
d. print(patient_data.corr().sort_values('8'))
e. Columns FJ, E, N have the highest magnitude correlation coefficients

cdj2273 { CALEB JOHNSON
oak2629 { AIMUN KHAN
LAB# 1

1)

	X=0	X=1
Y=0	1/4	1/4
Y=1	1/6	1/3

a) $P(X=1) = 1/4 + 1/3 = \boxed{7/12}$
b) $P(X=1|Y=1) = \boxed{2/3}$
c) $Var(X) = E[X^2] - E[X]^2 = 7/12 - 49/144 = \boxed{35/144}$
d) $Var(X|Y=1) = E[X^2|Y=1] - E[X|Y=1]^2 = \frac{2}{3} - \frac{4}{9} = \boxed{\frac{2}{9}}$
e) $E[X^3+X^2+3Y^7|Y=1] = 10/3 + 3/3 = 13/3$

2) FOR POINTS $r_p = [3,3,3], [1,2,3], [0,0,1]$, FIND THEIR PROJECTION ONTO A PLANE GIVEN BY VECTORS $[1,1,1]$ AND $[1,0,0]$. CHANGE $[0,1,1]$ TO $[0,1,-1]$ TO MAKE ORTHOGONAL
- SET ORIGIN TO $[0,0,0]$. FIND ORTHOGONAL, NORMAL VECTOR $n = [0,1,-1]$
- LET $e_1 = \frac{1}{\sqrt{2}}[0,1,1]$, $e_2 = [1,0,0]$, $r_0 = [0,0,0]$
- OUR PROJECTIONS ONTO THE GIVEN PLANE ARE $= [t_1, t_2]$ WHERE $t_1 = dot(e_1, r_p - r_0)$, $t_2 = dot(e_2, r_p - r_0)$

FOR [3,3,3]
$[t_1, t_2] = [3,3]$

FOR [1,2,3]
$[t_1, t_2] = [5/\sqrt{2}, 1]$

FOR [0,0,1]
$[t_1, t_2] = [\sqrt{2}, 0]$

3)
THE PROBABILITY OF A NUMBER OF HEADS, h, IN AN EXPERIMENT WITH 100 BINOMIAL TRIALS IS REPRESENTED AS:

$$P(\#\text{ OF HEADS} = h) = \left(\frac{2}{3}\right)^h \left(\frac{1}{3}\right)^{(100-h)} \binom{100}{h}$$

- USING PYTHON TOWRITE A SHORT FUNCTION TO INCREMENTALLY ACCUMULATE THE PROBALITY OF #'S OF HEADS FROM 0 TO 50.

- $P(\text{HEADS} \leq 50) = \boxed{.00041934}$

PYTHON PSEUDOCODE

```
for x=50:100
    prob =
    totalprob += prob
answer = totalprob
```