

# Problem4

February 4, 2020

```
[3]: import numpy as np

# Create matrix of 10,000 samples from 2-D Gaussian distribution
mean = [-5, 5]
cov = [[20, .8], [.8, 30]]
n = 10000
x = np.random.multivariate_normal(mean, cov, n)
```

```
[4]: # Calculate mean matrix by taking mean of each column (mean matrix composed of
      ↳ mean from each column/variable)
sum1 = 0
sum2 = 0

for element in x:
    sum1 += element[0]
    sum2 += element[1]

ave1 = sum1 / n
ave2 = sum2 / n

meanMatrix = [ave1, ave2]
print("Mean Matrix:", meanMatrix)
```

Mean Matrix: [-4.955734757204614, 5.014607839704126]

```
[6]: # Calculate covariance matrix (main diagonals are variance of X and Y and other
      ↳ spots are covariance of X and Y)
varX = 0
varY = 0
covXY = 0

for element in x:
    varX += np.square((element[0] - meanMatrix[0]))
    varY += np.square((element[1] - meanMatrix[1]))
    covXY += ((element[0] - meanMatrix[0]) * (element[1] - meanMatrix[1]))

varX /= n
```

```
varY /= n
covXY /= n

covMatrix = [[varX, covXY], [covXY, varY]]
print("Covariance Matrix: ", covMatrix)
```

```
Covariance Matrix:  [[19.999386178942345, 0.7607910376502169],
[0.7607910376502169, 29.28712649905788]]
```

```
[ ]:
```