q7

February 23, 2024

1 Section 7

[ ]:
```
import matplotlib.pyplot as plt
import matplotlib.ticker as tick
import numpy as np
import scipy.stats as stats
```

[ ]:
```
N = 100
seed = 0
rng = np.random.Generator(np.random.PCG64(seed))
#Generating samples
xOne = rng.normal(3, 3, N)
xTwoBias = rng.normal(4, 2, N)
xTwo = 0.5*xOne + xTwoBias
```

1.1 Question 1

[ ]:
```
#Question 1
samples = np.dstack((xOne, xTwo))[0]
print("Shape of samples: ", np.shape(samples))
sampleMean = np.mean(samples, axis=0)
print("Mean of samples: ", sampleMean)

# print(xOne)
# print(xTwo)
#print('\n', samples)
```

```
Shape of samples:  (100, 2)
Mean of samples:  [3.24329008 5.52050421]
```

1.2 Question 2

[ ]:
```
#Question 2
cov = np.cov(samples.T)
print(cov)
```

```
[[8.41540931 4.51673469]
 [4.51673469 6.10325205]]
```

1.3 Question 3

```
[ ]: #Question 3
     eigenvalues, normalizedEigenvectors = np.linalg.eig(cov)
     print("Normalized eigenvector: ", normalizedEigenvectors)
```

```
Normalized eigenvector:  [[ 0.78992438 -0.61320427]
 [ 0.61320427  0.78992438]]
```

1.4 Question 4

```
[ ]: #Question 4
     eigenvectors = np.array(normalizedEigenvectors)
     eigenvectors[: ,0] = normalizedEigenvectors[:, 0] * eigenvalues[0]
     eigenvectors[: ,1] = normalizedEigenvectors[:, 1] * eigenvalues[1]

     print("Eigenvector with eigenvalue magnitude: \n", eigenvectors)
     print("Eigenvalue: \n", eigenvalues)

     startingPoint = np.array([sampleMean, sampleMean]).T
     print("SampleMean: \n", startingPoint)

     plt.scatter(xOne, xTwo)
     plt.quiver(startingPoint[0], startingPoint[1], eigenvectors[0],␣
      ↪eigenvectors[1], scale= 1, scale_units = 'xy', angles = 'xy', color = ['r',␣
      ↪'k'])
     plt.xlim(-15, 15)
     plt.ylim(-15, 15)
     plt.title("Q4 Eigenvector and Sample Points")
     plt.xlabel("X1")
     plt.ylabel("X2")
     plt.show()
```

```
Eigenvector with eigenvalue magnitude:
 [[ 9.41721797 -1.59248604]
 [ 7.31041905  2.05142661]]
Eigenvalue:
 [11.92167023  2.59699113]
SampleMean:
 [[3.24329008 3.24329008]
 [5.52050421 5.52050421]]
```

1.5 Question 5

```
[ ]: normSamples = samples - sampleMean
     #np.shape(normalizedEigenvectors.T)
     transposedNormEigenvec = normalizedEigenvectors.T
```

```python
for i in range(0, len(normSamples)):
    normSamples[i] = np.dot(transposedNormEigenvec, normSamples[i])

print(np.shape(normSamples.T))
normSamplesT = normSamples.T
plt.scatter(normSamplesT[0], normSamplesT[1])
plt.xlim(-15, 15)
plt.ylim(-15, 15)
plt.title("Q5 Normalized Sample Points")
plt.xlabel("Evec1")
plt.ylabel("Evec2")
plt.show()
```

```
(2, 100)
```

[ ]:

```

```