q6

February 23, 2024

1 Appendix

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

[ ]:
```
np.random.seed(0)
```

[ ]:
```
def normalDistGen(mean, covar, axisRange, fineness):
    normal = stats.multivariate_normal(mean, covar)
    xUpperWindow = axisRange[0] + axisRange[2]
    yUpperWindow = axisRange[1] + axisRange[2]
    x = np.linspace(axisRange[0], xUpperWindow, fineness)
    y = np.linspace(axisRange[1], yUpperWindow, fineness)
    X, Y = np.meshgrid(x,y)
    pos = np.empty(X.shape + (2,))
    pos[:, :, 0] = X; pos[:, :, 1] = Y

    return (normal, X, Y, pos)
```

[ ]:
```
#Question 1
mean = np.array([1, 1])
covar = np.array([[1, 0], [0, 2]])
normal, X, Y, pos = normalDistGen(mean, covar, [-2, -2, 6], 500)

print(pos.shape)
#Plot Surface
# fig, ax = plt.subplots(subplot_kw={"projection": "3d"})
# surf = ax.plot_surface(X, Y, normal.pdf(pos), cmap='viridis',
#                        linewidth=0, antialiased=False)
# fig.colorbar(surf)

#Plot Contour
fig, ax = plt.subplots(1,1)
```

```
contour = ax.contourf(X, Y, normal.pdf(pos), levels = 10, cmap = 'viridis')
fig.colorbar(contour)

plt.title("Q1 Plot")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

```
(500, 500, 2)
```

[ ]:
```
#Question 2
mean = np.array([-1, 2])
covar = np.array([[2, 1], [1, 4]])
normal, X, Y, pos = normalDistGen(mean, covar, [-6, -3, 10], 500)

#Plot
# fig, ax = plt.subplots(1,1) #plt.subplots(subplot_kw={"projection": "3d"})
# surf = ax.plot_surface(X, Y, normal.pdf(pos), cmap='viridis',
#                        linewidth=0, antialiased=False)
# fig.colorbar(surf)

#Plot Contour
fig, ax = plt.subplots(1,1)
contour = ax.contourf(X, Y, normal.pdf(pos), levels = 10, cmap = 'viridis')
fig.colorbar(contour)

plt.title("Q2 Plot")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

[ ]:
```
#Question 3
meanOne = np.array([0, 2])
meanTwo = np.array([2, 0])
covar = np.array([[2, 1], [1, 1]])
normalOne, X, Y, pos = normalDistGen(meanOne, covar, [-4, -4, 10], 500)
normalTwo, X, Y, pos = normalDistGen(meanTwo, covar, [-4, -4, 10], 500)

#Plot
# fig, ax = plt.subplots(subplot_kw={"projection": "3d"})
# surf = ax.plot_surface(X, Y, normalOne.pdf(pos) - normalTwo.pdf(pos),␣
 ↪cmap='viridis',
#                        linewidth=0, antialiased=False)
# fig.colorbar(surf)

#Plot Contour
fig, ax = plt.subplots(1,1)
```

```
contour = ax.contourf(X, Y, normalOne.pdf(pos) - normalTwo.pdf(pos), levels =␣
 ↪10, cmap = 'viridis')
fig.colorbar(contour)

plt.title("Q3 Plot")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

[ ]:
```
#Question 4
meanOne = np.array([0, 2])
meanTwo = np.array([2, 0])
covarOne = np.array([[2, 1], [1, 1]])
covarTwo = np.array([[2, 1], [1, 4]])
normalOne, X, Y, pos = normalDistGen(meanOne, covarOne, [-4, -4, 10], 500)
normalTwo, X, Y, pos = normalDistGen(meanTwo, covarTwo, [-4, -4, 10], 500)

#Plot
# fig, ax = plt.subplots(subplot_kw={"projection": "3d"})
# surf = ax.plot_surface(X, Y, normalOne.pdf(pos) - normalTwo.pdf(pos),␣
 ↪cmap='viridis',
#                        linewidth=0, antialiased=False)
# fig.colorbar(surf)

#Plot Contour
fig, ax = plt.subplots(1,1)
locator = tick.MaxNLocator(prune='both',nbins=5)
contour = ax.contourf(X, Y, normalOne.pdf(pos) - normalTwo.pdf(pos), levels =␣
 ↪10)
fig.colorbar(contour)

plt.title("Q4 Plot")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

[ ]:
```
#Question 5
meanOne = np.array([1, 1])
meanTwo = np.array([-1, -1])
covarOne = np.array([[2, 0], [0, 1]])
covarTwo = np.array([[2, 1], [1, 2]])
normalOne, X, Y, pos = normalDistGen(meanOne, covarOne, [-7, -7, 12], 500)
normalTwo, X, Y, pos = normalDistGen(meanTwo, covarTwo, [-7, -7, 12], 500)

#Plot
# fig, ax = plt.subplots(subplot_kw={"projection": "3d"})
```

```
# surf = ax.plot_surface(X, Y, normalOne.pdf(pos) - normalTwo.pdf(pos),␣
 ↪cmap='viridis',
#                        linewidth=0, antialiased=False)
# fig.colorbar(surf)

#Plot Contour
fig, ax = plt.subplots(1,1)
locator = tick.MaxNLocator(prune='both',nbins=5)
contour = ax.contourf(X, Y, normalOne.pdf(pos) - normalTwo.pdf(pos), levels =␣
 ↪10)
fig.colorbar(contour)

plt.title("Q5 Plot")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

[ ]: