Quiz 1 - Questions 1 & 2

Amanjit Gill

In [44]:
```
from sympy import *
from sympy.integrals import *
```

Question 1

(a)

In [45]:
```
# define symbols
x1 = Symbol('x_1')
x2 = Symbol('x_2')
```

In [46]:
```
# define joint density
f = (1/(2*x2)) * exp(-(x2/4 + 2*x1/x2))
f
```

Out[46]:

$$\frac{e^{-\frac{2x_1}{x_2}-\frac{x_2}{4}}}{2x_2}$$

In [47]:
```
# integrate to get marginal density of x2

wrt = x1
lower = 0
upper = oo

f_x2 = integrate(f, (wrt, lower, upper), conds='none')
f_x2
```

Out[47]:

$$\frac{e^{-\frac{x_2}{4}}}{4}$$

(b)

In [48]:
```
# divide f by f_x2 to get conditional probability of x1|x2

f_x1_cond = simplify(f/f_x2)
f_x1_cond
```

Out[48]:

$$\frac{2e^{-\frac{2x_1}{x_2}}}{x_2}$$

(c)

In [49]:
```
# obtain g*(X1) by E(X1 | X2 = x2)

wrt = x1
lower = 0
upper = oo
```

```
g_star = integrate(x1*f_x1_cond, (wrt, lower, upper), conds='none')
g_star
```

Out[49]: $\frac{x_2}{2}$

(d)

In [76]:
```
# MSE

wrt = x1
lower = 0
upper = oo

integrate((x1 - x2/2)**2 * f_x1_cond, (wrt, lower, upper), conds='none')
```

Out[76]: $\frac{x_2^2}{4}$

Question 2

(a)

In [83]:
```
# characteristic polynomial

lam = Symbol('lambda')

Sigma = Matrix([[8, 3],[3, 9]])

char_poly = det(Sigma - lam*eye(2))
char_poly
```

Out[83]: $\lambda^2 - 17\lambda + 63$

(b)

In [117…
```
# eigenvalues

eigs = solve(char_poly, lam)
eigs
```

Out[117]:
```
[17/2 - sqrt(37)/2, sqrt(37)/2 + 17/2]
```

In [163…
```
# eigenvectors

Sigma.eigenvects()
```

Out[163]:
```
[(17/2 - sqrt(37)/2,
  1,
  [Matrix([
   [-sqrt(37)/6 - 1/6],
   [                1]])]),
 (sqrt(37)/2 + 17/2,
  1,
  [Matrix([
   [-1/6 + sqrt(37)/6],
   [                1]])])]
```

In [219…
```
# normalise eigenvector (17 - sqrt(37))/2
```

```
vect1 = Sigma.eigenvects()[0][2][0]
length = sqrt(vect1[0]**2 + vect1[1]**2)
vect1_norm = cancel(vect1/length)
vect1_norm
```

Out[219]:

$$\begin{bmatrix} \frac{-\sqrt{37}-1}{\sqrt{2\sqrt{37}+74}} \\ \frac{6}{\sqrt{2\sqrt{37}+74}} \end{bmatrix}$$

In [227…

```
vect1_norm.applyfunc(lambda x : round(x, 4))
```

Out[227]:

$$\begin{bmatrix} -0.763 \\ 0.6464 \end{bmatrix}$$

In [221…

```
# normalise eigenvector (17 + sqrt(37))/2

vect2 = Sigma.eigenvects()[1][2][0]
length = sqrt(vect2[0]**2 + vect2[1]**2)
vect2_norm = cancel(vect2/length)
vect2_norm
```

Out[221]:

$$\begin{bmatrix} \frac{-1+\sqrt{37}}{\sqrt{74-2\sqrt{37}}} \\ \frac{6}{\sqrt{74-2\sqrt{37}}} \end{bmatrix}$$

In [228…

```
vect2_norm.applyfunc(lambda x : round(x, 4))
```

Out[228]:

$$\begin{bmatrix} 0.6464 \\ 0.763 \end{bmatrix}$$

(c)

In [251…

```
# use eigenvalues and eigenvectors to compute X^(1/2)
eigval1 = Sigma.eigenvects()[0][0]
eigval2 = Sigma.eigenvects()[1][0]

(sqrt(eigval1)*vect1_norm*vect1_norm.T + sqrt(eigval2)*vect2_norm*vect2_norm.T).eva
```

Out[251]:

$$\begin{bmatrix} 2.77961 & 0.523229 \\ 0.523229 & 2.95402 \end{bmatrix}$$

(d)

In [254…

```
# use eigenvalues and eigenvectors to compute X^(1/2)

(eigval1**-0.5*vect1_norm*vect1_norm.T + eigval2**-0.5*vect2_norm*vect2_norm.T).eva
```

Out[254]:

$$\begin{bmatrix} 0.372171 & -0.0659206 \\ -0.0659206 & 0.350198 \end{bmatrix}$$