ALI Exam 2019

In [ ]:

```
# Student number: 253904
```

In [68]:

```
# Import all necessary libraries here
import sympy as sp
from sympy import *
init_printing()
from IPython.display import display, Latex, HTML
import numpy as np
import pandas as pd
```

Feel free to add cells if you need to. The easiest way to convert to pdf is to save this notebook as .html (File-->Download as-->HTML) and then convert/print this html file to pdf.

Assignment 1 (12%)

In [ ]:

```
# a)
#No. Two matrices are row equivalent only if there is a sequence of
#row operations that can transform one of the matrices into the other
```

In [ ]:

```
# b)
#true, since having no free variables makes it an inversible matrix,
#meaning it will have only one unique solution
```

In [ ]:

```
# c)
#No, because it A has more columns than rows, meaning it will
#have a free variable and will not have only one solution
```

In [ ]:

```
# d)
#It will NOT be diagonalizable, because A must have n distinct eignvalues, and here it
 does not
```

Assignment 2 (8%)

In [142]:

```
A = Matrix([[4,8,-2], [-6,2,10], [-2,6,6]])
# a)
```

In [143]:

```
# b)
A.rref()
#Since A has a free variable, the columns are dependant,
#therefore {v1,v2,v3} is a linearly dependant set
```

Out[143]:

$$\left(\begin{bmatrix}1 & 0 & -\frac{3}{2}\\ 0 & 1 & \frac{1}{2}\\ 0 & 0 & 0\end{bmatrix}, \quad (0, \quad 1)\right)$$

Assignment 3 (25%)

In [149]:

```
# a)
a,x, b, c = symbols('a x b c')
B = Matrix([[a-x, b, c], [1, -x, 0], [0,1,-x]])
solve(B.det(), a)
```

Out[149]:

$$\left[-\frac{b}{x} - \frac{c}{x^2} + x\right]$$

In [ ]:

In [145]:

```
# b)
l = symbols('l')
C = Matrix([[2-l, 4], [3,2-l]])
display(Latex("The determinant is 0 when l = $${}$$".format((latex(solve(C.det()+0, l
))))))
```

The determinant is 0 when l =

$$[2 + 2\sqrt{3}, \quad -2\sqrt{3} + 2]$$

In [146]:

```
# c)
D = Matrix([[2,0,0,1], [0,1,0,0], [1,6,2,0], [1,1,-2,3]])

# We go through the first row as it has the most 0s
D[0,0]*D.cofactor(0,0)+D[0,3]*D.cofactor(0,3)
```

Out[146]:

$$8$$

In [156]:

```
# d)
A = Matrix([[1,1,3], [1,2,4], [1,3,a]])
B = Matrix([2,3,b])

L,U, perm = A.row_join(B).LUdecomposition()
display(U)

# one solution - when a != 5
# many solutions - when a = 5 and b = 0
# no solutions = when a = 5 and b != 0

display(Latex("The determinant is 0 when l = $${}$$".format((latex(solve(A.det()+0, a
))))))
```

$$\begin{bmatrix}1 & 1 & 3 & 2\\0 & 1 & 1 & 1\\0 & 0 & a-5 & b-4\end{bmatrix}$$

The determinant is 0 when l =

$$\left[5\right]$$

Assignment 4 (10%)

In [ ]:

Assignment 5 (15%)

In [125]:

```
A = Matrix([[3, -1, -2], [2,0,-2], [2,-1,-1]])
display(A)

# a)
#Since A has n distinct eigenvalues, it is diagonalizable
```

$$\begin{bmatrix}3 & -1 & -2\\2 & 0 & -2\\2 & -1 & -1\end{bmatrix}$$

In [124]:

```
# b)
vecs = A.eigenvects()

#Construct P from the eigenvects
v1 = vecs[0][2][0]
v2 = vecs[1][2][0]
v3 = vecs[1][2][1]

X = v1.row_join(v2).row_join(v3)
Xinv = X**-1

D = Xinv*A*X

display(Latex("X = $${}$$".format((latex(X)))))
display(Latex("D = $${}$$".format((latex(D)))))

display(Latex("Checking if correct $${}$$".format((latex(X*D*Xinv)))))
```

X =

$$\begin{bmatrix} 1 & \frac{1}{2} & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

D =

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Checking if correct

$$\begin{bmatrix} 3 & -1 & -2 \\ 2 & 0 & -2 \\ 2 & -1 & -1 \end{bmatrix}$$

In [66]:

```
# c)

display(Latex("$$A^n = X*D^n*X^-1$$"))
```

$$A^n = X * D^n * X^-1$$

Assignment 6 (20%)

In [157]:

```
# a)
df = pd.read_excel('Smoking_and_Cancer.xlsx')
df.head

x = df['Cigarettes sold']
y = df['lung cancer']

# Linear model
X1 = Matrix([ones(len(x),1)]).row_join(Matrix(x))
XtX = X1.T*X1
Xty = X1.T*Matrix(y)
Mat, _ = XtX.row_join(Xty).rref()
B1 = Mat[:,-1]
display(Latex("$$y_1(x) = {}+{}x$$".format(round(B1[0],2), round(B1[1], 4))))

#Quadratic model
X2 = Matrix([ones(len(x),1)]).row_join(Matrix(x)).row_join(Matrix(x**2))
XtX = X2.T*X2
Xty = X2.T*Matrix(y)
Mat, _ = XtX.row_join(Xty).rref()
B2 = Mat[:,-1]
display(Latex("$$y_2(x) = {}+{}x+{}x^2$$"
              .format(round(B2[0],2), round(B2[1], 4), round(B2[2], 4))))
```

$$y_1(x) = 6.47 + 0.5291x$$

$$y_2(x) = -5.89 + 1.4732x + -0.0171x^2$$

In [158]:

```
# b)

display(Latex(
    "Error of linear model: {}, error of quadratic model: {}. The quadratic model is be
st fit for this data"
    .format(round((Matrix(y)-X1*B1).norm(), 2), round((Matrix(y)-X2*B2).norm(), 2))))
```

Error of linear model: 19.87, error of quadratic model: 19.01. The quadratic model is best fit for this data

In [159]:

```
# c)
display(Latex("$$y_2(50) = {}$$".format(round(B2[0]+B2[1]*50+B2[2]*50**2, 2
))))
```

$$y_2(50) = 24.9$$

Assignment 7 (10%)

In [114]:

```
A = Matrix([[2,5,4], [6,3,0], [6,3,0], [2,5,4]])
AtA = A.T*A
vecs = AtA.eigenvects()
vecs
```

Out[114]:

$$\left[\left(0, \quad 1, \quad \left[\left[\begin{matrix}\frac{1}{2}\\-1\\1\end{matrix}\right]\right]\right), \quad \left(36, \quad 1, \quad \left[\left[\begin{matrix}-1\\\frac{1}{2}\\1\end{matrix}\right]\right]\right), \quad \left(144, \quad 1, \quad \left[\left[\begin{matrix}2\\2\\1\end{matrix}\right]\right]\right.\right.$$

In [101]:

```
s1 = sqrt(vecs[2][0])
s2 = sqrt(vecs[1][0])
s3 = sqrt(vecs[0][0])

v1 = vecs[2][2][0].normalized()
v2 = vecs[1][2][0].normalized()
v3 = vecs[0][2][0].normalized()

u1 = (s1**-1)*A*v1
u2 = (s2**-1)*A*v2
U = u1.row_join(u2)

V = v1.row_join(v2).row_join(v3)
Vt = V.T
S = diag(s1,s2).row_join(zeros(2,1))

display(U*S*Vt)
display(A)
```

$$\left[\begin{matrix}2 & 5 & 4\\6 & 3 & 0\\6 & 3 & 0\\2 & 5 & 4\end{matrix}\right]$$

$$\left[\begin{matrix}2 & 5 & 4\\6 & 3 & 0\\6 & 3 & 0\\2 & 5 & 4\end{matrix}\right]$$

In [ ]:

```
```