In [1]:
```
                                                    ###### Homework 3 - Part 3 ###########
```

In [2]:
```
#Note the following:

#1. Each question is worth 1 point
#2. Write your name and student ID below
#3. Remember to save your file and upload to Canvas
```

In [3]:
```
#Student Name: Julia Troni
```

In [4]:
```
#Student ID: 109280095
```

In [ ]:
```

```

In [5]:
```
#RUN THIS CELL FIRST TO ANSWER QUESTIONS 1 - 5

import numpy as np
from scipy import linalg
from numpy import array
from numpy import linalg
from numpy.linalg import norm
from numpy.linalg import qr
from numpy.linalg import eig
from math import inf

# define matrix A
A = np.array([[2, 1, 1],
 [1, 3, 2],
 [1, 0, 0]])

#define vector b
b = np.array([4, 5, 6])

#define matrix K
K = np.array([ [2., 1., 1.], [1., 3., 2.], [1., 0., 0]])
```

In [6]:
```
#Question 1

#Suppose you are given the matrix and vector above as a system of linear equations (SLE

#Write Python code to find vector x

#Write your answer below

x = linalg.solve(A, b)
x
```

Out[6]:
```
array([  6.,  15., -23.])
```

In [7]:
```
##another way to solve Question 1 using QR decomposition
Q,R=linalg.qr(A)
y=np.dot(Q.T,b)
x= linalg.solve(R,y)
x
```

Out[7]:
```
array([  6.,  15., -23.])
```

In [ ]:
```

```

In [8]:
```
#Question 2

#Print the vector x you calculated in Question 1

#Write your answer below


x = linalg.solve(A, b)
x
```

Out[8]:
```
array([  6.,  15., -23.])
```

In [ ]:
```

```

In [9]:
```
#Question 3


#Write Python code to find the Euclidean norm of vector x in Question 2


#Write your answer below

norm(x,2)
```

Out[9]:
```
28.106938645110393
```

In [ ]:
```

```

In [10]:
```
#Question 4

#Write Python code to perform the following:

#1. Decompose matrix K by QR decomposition
#2. Obtain the eigenvalues (values) and eigenvectors(vectors) of matrix K
#3. print Q

#Write your answer below

Q, R = linalg.qr(K) # QR decomposition with qr function
```

Out[10]:
```
array([[-0.81649658,  0.27602622, -0.50709255],
       [-0.40824829, -0.89708523,  0.16903085],
       [-0.40824829,  0.34503278,  0.84515425]])
```

In [11]:
```
#Question 5


#Write Python code to compute the dot product of values and vectors you obtained in Que

#Write your answer below

np.dot(Q,R)
```

Out[11]:
```
array([[ 2.00000000e+00,  1.00000000e+00,  1.00000000e+00],
       [ 1.00000000e+00,  3.00000000e+00,  2.00000000e+00],
       [ 1.00000000e+00,  1.09812324e-16, -2.08416416e-17]])
```

In [12]:
```
#RUN THIS CELL FIRST TO ANSWER QUESTIONS 6 - 7

#singular-value decomposition
from numpy import array
from scipy.linalg import svd

# define a matrix Z as follows
Z = array([
[1, 2],
[3, 4],
[5, 6],
[12,-3]])

print(Z)
```

```
[[ 1  2]
 [ 3  4]
 [ 5  6]
 [12 -3]]
```

In [13]:
```
#Question 6

#1. Write Python code to find the singular value decomposition of matrix Z defined abov
#2. Print matrix U


#Write your answer below

U, s, V = svd(Z)

U
```

Out[13]:
```
array([[-0.08484342, -0.2398591 , -0.38212411, -0.88839764],
       [-0.24413269, -0.47104012, -0.71362907,  0.45744288],
       [-0.40342195, -0.70222114,  0.58614032, -0.02399435],
       [-0.8777537 ,  0.47694266, -0.03397419, -0.03032994]])
```

In [ ]:
```

```

In [14]:
```
#Question 7

#Write Python code to find the scalar multiplication of matrix U by the max norm of vec

#Write your answer below


# calculate norm
maxnorm = norm(x, inf)
U*maxnorm
```

Out[14]:
```
array([[ -1.95139876,  -5.5167592 ,  -8.78885448, -20.43314581],
       [ -5.61505182, -10.83392269, -16.41346856,  10.52118634],
       [ -9.27870489, -16.15108617,  13.48122735,  -0.55186994],
       [-20.18833503,  10.96968107,  -0.78140638,  -0.69758863]])
```

In [ ]:
```

```

In [15]:
```
#RUN THIS CELL FIRST TO ANSWER QUESTIONS 8 - 10

import numpy as np
from scipy import linalg
from numpy import array
from numpy import linalg
from numpy.linalg import norm
from numpy.linalg import qr
from numpy.linalg import eig
from math import inf

# define matrix M
M = np.array([[12, -1, 41,34],
 [2, 3, -5, 17],
 [4, 6, -7,33],
 [3,5,2,1]])

#define matrix D
D = np.array([ [2, 1, 12, -7], [10, -32, 2, 4], [11, 20, 0, 5]])

#define vector f
f = np.array([14, 4, -6])

#define vector h
h = np.array([1, 4, 2])

#define vector o
o = np.array([1, 4, 2])
```

In [16]:
```
#Question 8


#Write Python code to find the dot product of matrix D and matrix M

#Write your answer below
```

Out[16]:
```
array([[  53,   38,  -21,  474],
       [  76,  -74,  564, -134],
       [ 187,   74,  361,  719]])
```

In [ ]:

In [17]:
```
#Question 9

#Suppose you were asked to confirm if the difference between the Euclidean norm of vect
# of vector h is a positive real number, what Python code would you write to show this?


#Write your answer below

#euclidean norm f
fnorm= norm(f,2)
# max norm of h
hmaxnorm = norm(h, inf)
diff= fnorm-hmaxnorm

if (diff > 0):
    print("Yes the difference is a positive real number: ", diff)
else:
    print("NO not positive real number")
```

```
Yes the difference is a positive real number:  11.748015748023622
```

In [ ]:

In [18]:
```
#Question 10

#What is the value of your result of your answer in Question 9 rounded to two decimal p

#Write your answer below

round(diff,2)
```

Out[18]:
```
11.75
```

In [ ]:

In [19]:
```
#RUN THIS CELL FIRST To ANSWER QUESTIONS 11 - 15

import numpy as np
import pandas as pd

#Obtain dataset
df = pd.read_csv("https://vincentarelbundock.github.io/Rdatasets/csv/AER/CigarettesB.cs
X = df.iloc[:,:-1]
y = df.iloc[:,-1]

#Use random splits as validation strategy for model evaluation
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, shuffle=True,
                                    test_size=0.05, random_state=42)

#import linear regression; obtain instance of linear regression
from sklearn.linear_model import LinearRegression
lr = LinearRegression()

#Fit linear regression model
lr.fit(X_train, y_train)

#Use linear regression model to predict target using test dataset
y_pred = lr.predict(X_test)

#import some linear regression performance metrics
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print("RMSE: ", np.round(rmse, 2))
```

```
RMSE:  0.2
```

In [20]:

```
#Question 11

#What is the value of the Mean Squared Error (MSE) from the output of RMSE above rounde


#Write your answer below

print("MSE: ", np.round(mse, 2))
```

```
MSE:  0.04
```

In [ ]:

In [21]:

```
#Question 12


#1. Record the RMSE value from the running the cell above
print("RMSE before was:  0.14 ")
print("MSE before was:  0.02 ")

#2. Change the value of test_size from 0.20 to 0.05
#3. Run the cell again with test_size = 0.05 and record the new RMSE value
print("New RMSE with test_size = 0.05 is : ", np.round(rmse, 2))
print("New MSE with test_size = 0.05 is : ", np.round(mse, 2))
#4. Compare the new RMSE with the original RMSE


#What can you say about the trend of RMSE values for both models?


#Write your answer below

'''When I changed the test size from 0.2 to 0.05, the new RMSE increased from 0.14 to 0
```

```
RMSE before was:  0.14
MSE before was:  0.02
New RMSE with test_size = 0.05 is :  0.2
New MSE with test_size = 0.05 is :  0.04
```

Out[21]:
```
'When I changed the test size from 0.2 to 0.05, the new RMSE increased from 0.14 to 0.2.
Also the MSE increased from 0.02 to 0.04'
```

In [22]:
```
#Question 13

#Consider your answer to Question 12. What can you say about the impact of decreasing t
# on how reliable your model is?

#Write your answer below



'''The first model with the test size of 0.2 appears to be more reliable than the model
```

Out[22]:
```
'The first model with the test size of 0.2 appears to be more reliable than the model wi
th the test size of 0.05, as it had lower RMSE and MSE values.'
```

In [ ]:

In [23]:
```
#Question 14


#How would you justify your answer in Question 13?

#Write your answer below
```

MSE measures the average squared difference between the predicted and actual values of the dependent variable. RMSE is simply the square root which converts the units back to the origiinal units of the output variable and is therefore more interpretable. So, both RMSE and MSE provides an idea of the magnitude of error

Smaller RMSE and MSE indicate better performance of the model.

Thus, the model with the test size of 0.2 appears to be more reliable than the model with the test size of 0.05, as it had lower RMSE and MSE values.

In [ ]:

In [24]:
```
#Question 15

#How would you fix the issue you raised/identified in your answer in Question 14 above?



#Write your answer below
```

There are several ways to improve the reliability.

1. Hyperparameter tuning: I can experiment with different hyperparameters like regularization strength or learning rate,
2. Address outliers and anomalies: since they can can significantly affect the model's performance, it's important to carefully analyze and address these data points.

In [25]:
```
##### End of Homework 3: Part 3 ####
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: