Data Science: EN.553.436/636
Midterm 1

Name:
JHED:

Instructions

- You will have 75 minutes to complete this exam.
- The exam has 3 problems, each having multiple parts.
- You are allowed a single (front and back) **handwritten** note sheet.
  - You must turn in the note sheet along with your exam.
- No notes (other than your note sheet), books, calculators, phones, laptops, or any sort of electronic devices are allowed during the exam.
- Do not look turn the page before the exam begins. Doing so will result in a penalty.
- Good Luck!

Problem 1

In the following problem you will be given a few code snippets and you will have to infer what method was being implemented, circle some errors, and provide some additional comments and discussion.

The following code block was used to load in the data for this problem. The file *"midterm1Problem1.csv"* contains the heights (in feet) of 2974 adults. Some basic descriptors were printed to allow you to better understand the data.

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from scipy.stats import norm as gaussian
5
6 height = np.loadtxt("midterm1Problem1.csv", delimiter=",")
7 print(height.shape)
8 print(pd.DataFrame(height).describe()) # Gives overview of basic statistics of array
```

```
(2974,)
                0
count  2974.000000
mean      6.442041
std       0.295179
min       5.000000
25%       6.260000
50%       6.490000
75%       6.670000
max       7.000000
```

a)

In the designated answer box below provide a brief description of what method the following code block is implementing. Additionally, line 7 includes compound logic. Describe in detail what it is doing.

b)

There is an error in the implementation of the method of (a). Circle it within the code block and provide a brief description in the corresponding answer box.

```
1  density_domain = np.linspace(4, 9, 1000)
2  density_list = []
3
4  for h in [0.01, 0.1, 0.5]:
5      g_h = gaussian(0,h)
6
7      density_h = g_h.pdf((density_domain - height[:,np.newaxis])).mean(axis=1)
8
9      density_list.append(density_h)
10
11     print(h, density_h.shape)
```

```
0.01 (2974,)
0.1 (2974,)
0.5 (2974,)
```

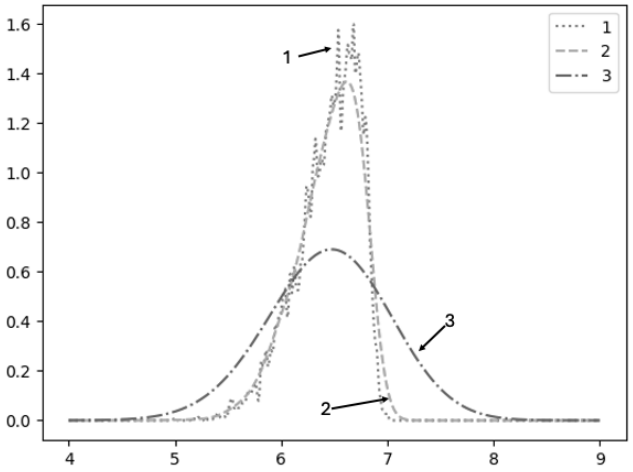
1.a) answer:

1.b) answer:

c)

The following code was used to plot the results from the previous code block. In the corresponding answer box match each h value to the line that it created. Provide a brief explanation of each choice.

```
plt.plot(density_domain, density_list[_], label="1")
plt.plot(density_domain, density_list[_], label="2")
plt.plot(density_domain, density_list[_], label="2")
plt.show()
```



1.c) answer

Line	h-value
1	
2	
3	

d)

What are the dangers of choosing an extremely large or small h-value?

1.d) answer

Problem 2

For Problem 2, parts (a,b,c) consider the following code block.

```
def myMidtermFunction(x, y):

    X = np.zeros((x.shape[0], 4))

    X[:,1] = x**2
    X[:,2] = x**3
    X[:,3] = np.sin(20*x)
    X[:,4] = np.cos(10*x)

    w = np.linalg.inv(X @ X) @ X.T @ y

    y_pred = w @ X

    return y_pred, w
```

a)

What method is being implemented?

2.a) answer:

b)

There are a total of 3 errors with the given implementation. Circle them within the code block and provide a brief explanation.

2.b) answer:

c)

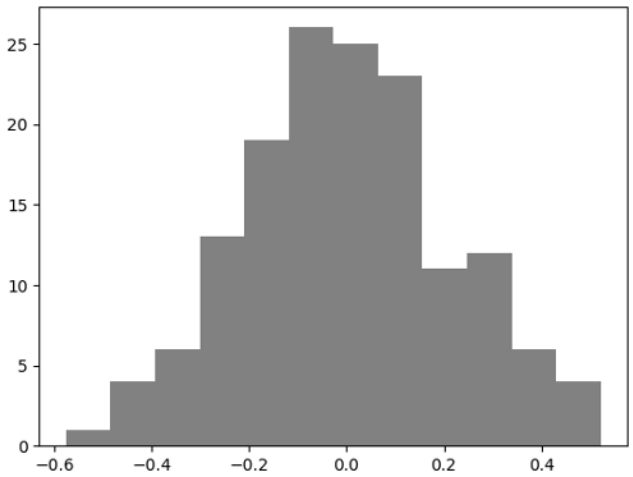
Provide the the equation that this code is attempting to predict. You can use β's as placeholder for the coefficients.

2.c) answer:

The following code was used to calculate and plot the residuals from the implementation on the last page.
NOTE: This is done under the assumption that the above code was implemented **correctly**, i.e. any errors that you identified in the previous part do **not** carry through.

```
y_pred, _ = myMidtermFunction(x,y)
res = y_pred - y
plt.hist(res, bins = 12)
plt.show()
print(np.mean(res), np.var(res))
```

```
-4.9353114188003625e-15 0.04530328695118832
```



d)

Given the provided histogram. Does this implementation satisfy the assumptions of the method that you specified for part (a)? Why or why not?

2.d) answer:

3)

The following code block was used to load in the data for this problem. The file *"midterm1Problem3.csv"* contains data with **8 features.**

```
X1 = np.genfromtxt("midterm1-PCA-data.csv", delimiter=",")
print(X1.shape)
```

```
(8, 442)
```

Consider the following code for the parts ( ).

```
X_ctd = X1 - X1.mean(axis = 0, keepdims = True)

C = X_ctd @ X_ctd.T / (X_ctd.shape[1] + 1)

E, V = np.linalg.eig(C)
```

a)

What is this code attempting to accomplish? (What method is being implemented?)

3.a) answer:

b)

There are 2 errors in the code. Circle them and provide a brief explanantion in the following box.

3.b) answer:

c)

Why is it necessary to center the data?

3.c) answer:

d)

What is the statistical meaning of the values in **E**? (Insufficient to just say they are the eigenvalues)

3.d) answer:

e)

Geometrically, how are the entries of **V** related to one another? Explain why this is important.

3.e) answer: