Name: Aragya Goyal
Student ID: 4541458
D Value = 58

Question 1:

a. My ML problem is using a ground-facing camera feed on an autonomous fixed-wing plane to determine/identify safe landing zones in case of a system failure. This problem would be considered a classification problem since the ML algorithm would determine whether a patch of area is safe or not for landing.
b. I would use the colors in the image to try and look for plains of grass, water bodies, etc. I could also use the image to try and determine the relative texture of the area a pixel is located such as by taking a pixel and comparing it with its neighbor and determining whether it is a big change or a small change in greyscale intensity. Finally, I could use edge detections on images and then calculate edge densities as a feature to feed into my ML model.
c. The output of my model would be classifying areas of the image into regions that are safe (1) or unsafe (0).
d. I would use Kaggle in order to efficiently find any open source datasets that exist on the internet. I was able to find this dataset from a quick search which looks relatively promising: https://www.kaggle.com/code/aletbm/aerial-image-for-semantic-segmentation. This dataset also comes with prelabeled data which would be nice for training purposes.
e. This problem would be challenging because when given an image, you cannot just say whether the whole image is safe to land or not and rather have to identify parts of the image where we can identify safe landing spots. Secondly, while this is a classification problem, there is a high likelihood that we are not able to create classes for everything that the camera might see which means that some areas may be uncategorized.
f. This problem interests me for my interest in robotics and autonomous systems. I have especially been interested in aerial systems as I have not worked with them before so far and would like to do so in the future.

Name: Aragya Goyal
Student ID: 4541458
D Value = 58

Question 2:
Text Response #1: The x plot does look gaussian. The z plot does look uniform. I would estimate x’s mean visually to be approximately 2.7. The true mean of x is approx. 2.58 which means that it is relatively close.

Text Response #2: The looping time was approx. 2.65 seconds whereas the vectorized operation was approximately 0.001 seconds. This means that the vectorized operation was faster by a factor of 2650 times which is very significant. Vectorized operations are a lot faster than loops because they remove a lot of the overhead. When looping, the computer has to increment the loop counter, check the loop condition, perform the action, and repeat until the loop is done. However, when working with vectorized operations, libraries like the NumPy library optimize these operations so that they effectively use hardware and also access memory more efficiently. The combination of these efforts allow for vectorized operations to be a lot faster than loops.

Text Response #3: The count from run #1 was 332,453, while the count from run #2 was 334,305. These values differ because the data used to generate the vector z comes from a random process, which results in a varying number of elements falling within the range 0 to 0.8 for each run. If the vector size were increased to 10 million, the overall count would increase and the relative difference between runs would become smaller, but the counts would still not be identical across multiple runs.

Name: Aragya Goyal
Student ID: 4541458
D Value = 58

Question 3:
The solution from the code was the following: [x, y, z] = [32.21212121, -4.45454545, -7.92424242]

HW #1

$\boxed{Q3}$ $[x, y, z] = [32.212121, -4.454545, -7.924242]$

$D = 58$

① $2x + 5y - 2z = 58$

$2(32.21) + 5(-4.45) - 2(-7.92) = 58$

$64.42 - 22.25 + 15.84 = 58$

$\underbrace{58.01}_{\text{Rounding error}} \approx 58$

Rounding error → Works ✓

② $2x + 6y + 4z = 6$

$2(32.21) + 6(-4.45) + 4(-7.92) = 6$

$64.42 - 26.7 - 31.68 = 6$

$6.04 = 6$ → ✓

③ $6x + 8y + 18z = 15$

$6(32.21) + 8(-4.45) + 18(-7.92) = 15$

$193.26 - 35.6 - 142.56 = 15$

$15.1 = 15$ → ✓

Name: Aragya Goyal
Student ID: 4541458
D Value = 58

Question 4:
Text Response #1: I would not get the same split. This is because of the function that I used in order to shuffle the rows. Each time the code is run, the row indices are randomized in a different pattern which means that my split will also be different each time.

Text Response #2: In order to make the split reproducible, I set a seed before I generated the shuffle for the row indices. Reproducibility is important in machine learning so that results can be trusted and verified. Without reproducibility, we would not be able to determine whether a result was actually correct implementation or potential data quirks/external influences.