WEEK 3 NORMAL DISTRIBUTION LAB

Import all the relevant libraries and functions.

[2]

2s

```
import numpy as np
from scipy import stats
import math as mt
import random as rnd
```

The normal distribution CDF , to calculate the area up to 64.43.

[6]

```
from scipy.stats import norm

mean = 64.43
std_dev = 2.99

x = norm.cdf(64.43, mean, std_dev)

print(x) # prints 0.5
```

Getting a middle range probability using the CDF. The following code shows how to calculate the area between 62 and 66.

[7]

```
from scipy.stats import norm

mean = 64.43
std_dev = 2.99

x = norm.cdf(66, mean, std_dev) - norm.cdf(62, mean, std_dev)

print(x) # prints 0.4920450147062894
```

Using the inverse CDF (called ppf()) in Python. This is used to return corresponding x value of a given area. For example, find the weight that 95% of golden retrievers fall under.

[9]

```
from scipy.stats import norm
```

```
x = norm.ppf(.95, loc=64.43, scale=2.99)
print(x) # 69.3481123445849
```

The z_score() function will take an x-value and scale it in terms of standard deviations, given a mean and standard deviation. The z_to_x() function takes a Z-score and converts it back to an x-value. Studying the two functions, you can see their algebraic relationship, one solving for the Z-score and the other for the x-value. We then turn an x-value of 8.0 into a Z-score of
3.333 and then turn that Z-score back into an x-value.

---

[10]

```
def z_score(x, mean, std):
    return (x - mean) / std
def z_to_x(z, mean, std):
    return (z * std) + mean
mean = 140000
std_dev = 3000
x = 150000
# Convert to Z-score and then back to X
z = z_score(x, mean, std_dev)
back_to_x = z_to_x(z, mean, std_dev)
print("Z-Score: {}".format(z))  # Z-Score: 3.333
print("Back to X: {}".format(back_to_x))  # Back to X: 150000.0
```

Choose 3 questions from the class worksheet and solve them using the codes given.