# REPORT FOR RISK PROFILING

## Introduction

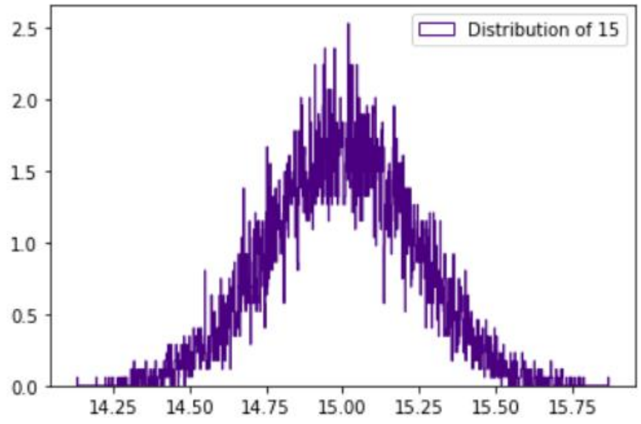
The purpose of this project is to simulate a scenario where temperature readings are being collected from multiple sensors and we want to analyze the risk of an error occurring in these readings. We also want to calculate the expected cost of such errors occurring, based on a cost function that we define.

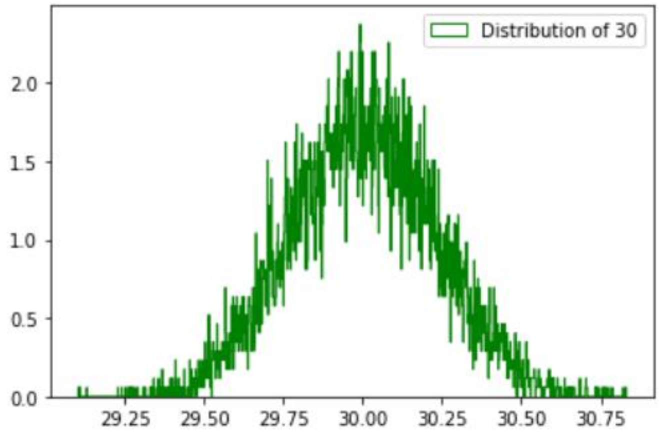
The code is implemented in Python and uses the NumPy and Matplotlib libraries for generating random numbers and creating histograms.

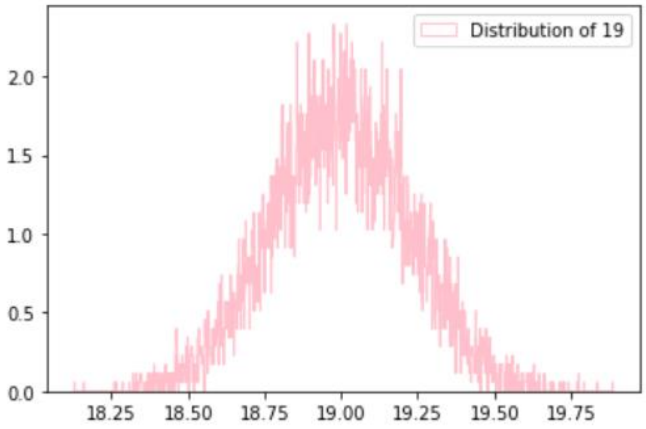
## Risk Assessment

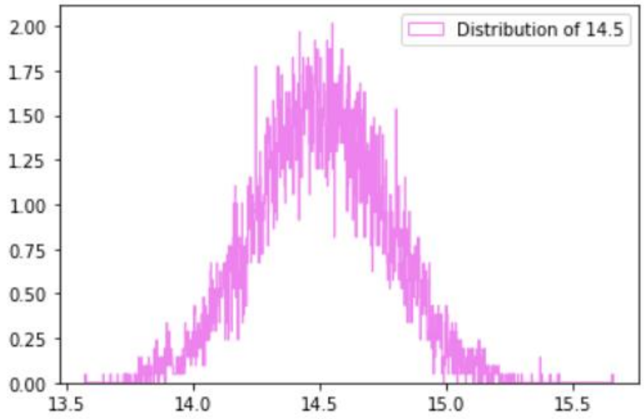
The first part of the code generates random temperature readings for 80 different sensors, with each sensor having a mean temperature and a standard deviation randomly chosen from a uniform distribution. The temperature readings are generated using a normal distribution with the mean and standard deviation values for each sensor.

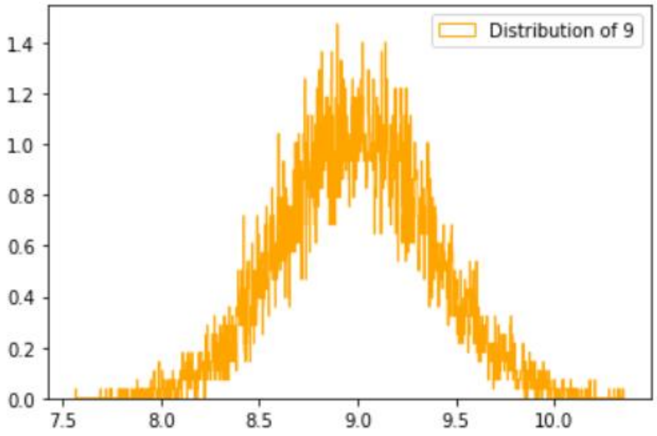
The code then plots histograms of the temperature readings for 5 different sensors, with the histograms coloured differently for each sensor.



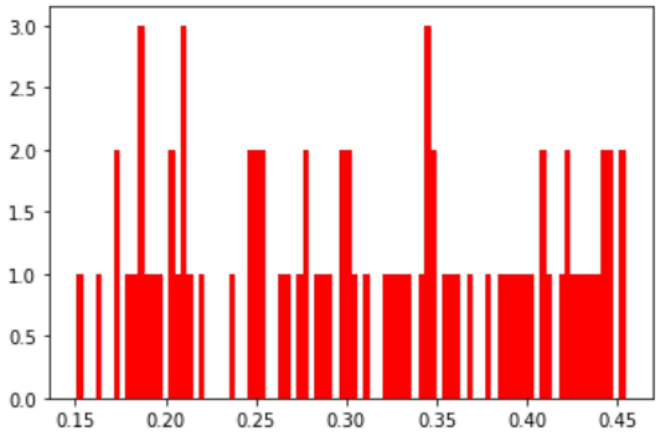








Next, the code calculates the probability of an error occurring for each sensor, defined as the proportion of temperature readings that fall outside a range of +/- 0.3 degrees from the mean temperature value for that sensor. The code then plots a histogram of the probability of error values for all 80 sensors.

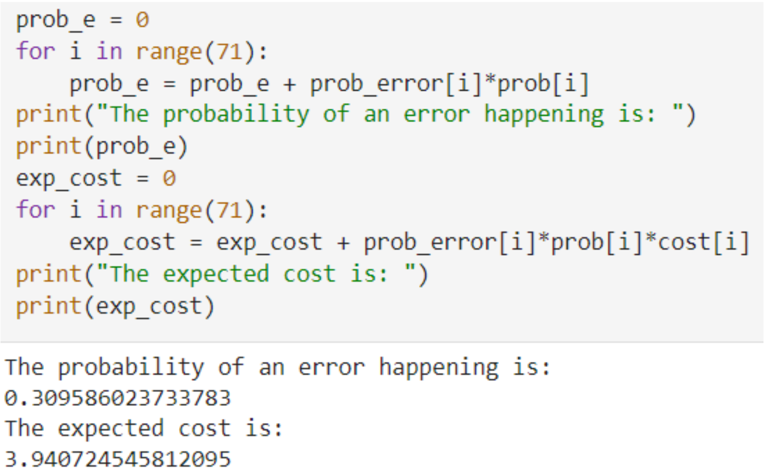


## Risk Analysis

The second part of the code defines a probability distribution for the true temperature values for each sensor. The probability distribution is based on the assumption that temperatures in the middle of the range (30 to 50 degrees) are more likely to occur, while temperatures at the extremes (below 5 or above 70 degrees) are less likely to occur. The probabilities for each sensor are randomly chosen from a uniform distribution.

The code then defines a cost function for each sensor, which is the cost of an error occurring for that sensor. The cost function is also based on a uniform distribution and varies depending on the true temperature value for each sensor.

Finally, the code calculates the probability of an error occurring for all 80 sensors, weighted by the probability of the true temperature values occurring. The code also calculates the expected cost of an error occurring, based on the probability of error and the cost function for each sensor.



```
prob_e = 0
for i in range(71):
    prob_e = prob_e + prob_error[i]*prob[i]
print("The probability of an error happening is: ")
print(prob_e)
exp_cost = 0
for i in range(71):
    exp_cost = exp_cost + prob_error[i]*prob[i]*cost[i]
print("The expected cost is: ")
print(exp_cost)
```

```
The probability of an error happening is: 
0.309586023733783
The expected cost is: 
3.940724545812095
```