Problem Statement 1

Q) Suppose that you want to investigate the process of tea preparation. (You may choose your own style of tea, e.g., regular, masala, kadak etc.)

- What would you use as the response variable?
- List all of the potential sources of variability that impact the response.

ANSWER:

- Response variable is that variable which we want to quantify as output. So in this case we can use a number of response variables to quantify our output:

1. Smell
2. Taste
3. Colour and many more

However among these criteria's taste and smell are subjective to bias. So for them to behave as a response variable properly we need to have some sort of averaging method. However in this case we can consider colour of the tea to be quantifiable, thus it can act as a proper response variable.

- Potential sources of variability in the process of tea preparation is that:

1. Temperature of the water.
2. Ratio of milk and water added.
3. Amount of tea leaves given.
4. Amount of sugar added.

Problem Statement 2

Q) Create a script in python to get maximum, minimum, and average values from the following data elements.

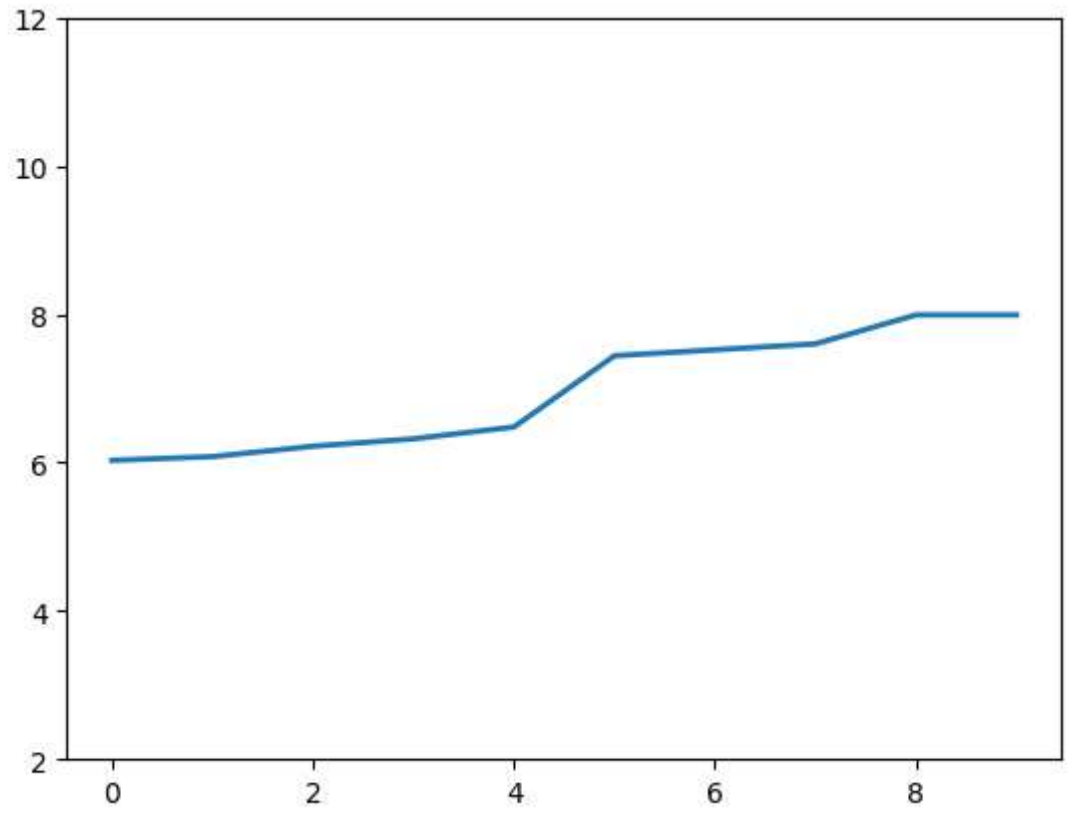
- Datapoints : [6.08 6.22 7.99 7.44 6.48 7.99 6.32 7.60 6.03 7.52]
- Estimate median, compare it with the average value and comment on the result.

In [15]:
```
import matplotlib.pyplot as plt
import statistics
li=[6.08,6.22,7.99,7.44,6.48,7.99,6.32,7.60,6.03,7.52]
li.sort()
#print(li)
plt.plot(li,linewidth=2)
plt.ylim(2,12)
plt.show()
print("MEDIAN =",statistics.median(li))
print("MEAN=",statistics.mean(li))
print("MAXIMUM VALUE=",max(li))
print("MINIMUM VALUE=",min(li))

'''
COMMENTS ON THE RESULT:

From the output it can be seen that here we get Mean and the median of the dataset are equal. Therefore from the datas
et we can conclude
that since median is the center of dataset and equal to mean so the points are evenly distributed and equally spaced.

'''
```




```
MEDIAN = 6.960000000000001
MEAN= 6.967000000000005
MAXIMUM VALUE= 7.99
MINIMUM VALUE= 6.03
```

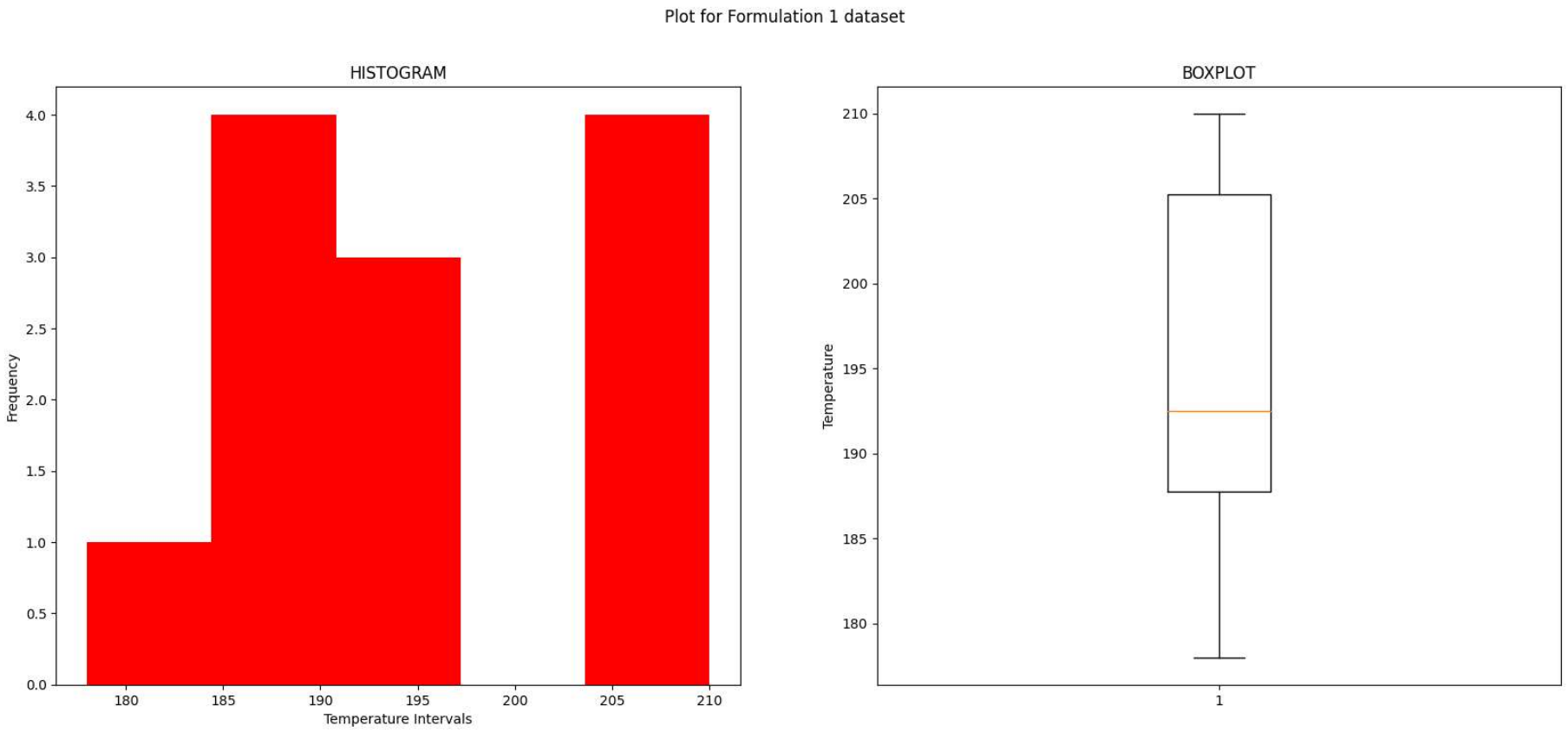
Out[15]:
```
'\nCOMMENTS ON THE RESULT:\n\nFrom the output it can be seen that here we get Mean and the median of the dataset are
equal. Therefore from the dataset we can conclude \nthat since median is the center of dataset and equal to mean so t
he points are evenly distributed and equally spaced.\n\n'
```

Problem Statement 3

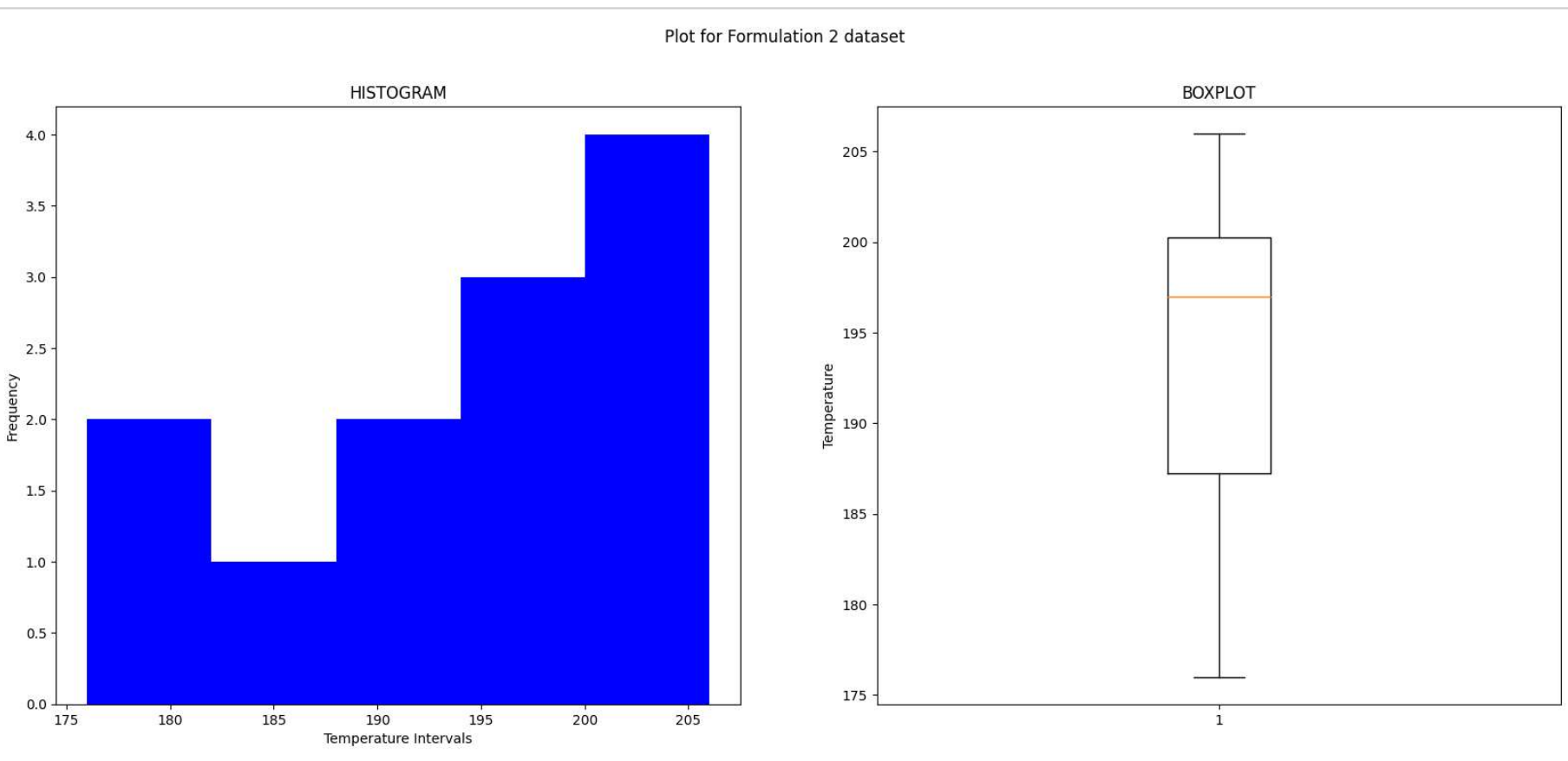
Q) The deflection temperature under load for two different formulations of ABS plastic pipe is being studied. Two samples of 12 observations each are prepared using each formulation, and the deflection temperatures (in °F) are reported below:

- Dataset_1 : [206,193,192,188,207,210,205,185,194,187,189,178]
- Dataset_2 : [177,176,198,197,185,188,206,200,189,201,197,203]

In [16]:
```
# Represent the datasets in Histogram plot and box and whisker plot.
# Plot for datset 1
Dataset_1=[206,193,192,188,207,210,205,185,194,187,189,178]
plt.figure(figsize=(20,8))
plt.suptitle("Plot for Formulation 1 dataset")
plt.subplot(1,    2,       1)
plt.title("HISTOGRAM")
plt.hist(Dataset_1,bins=5,color='Red') # Number of bins taken=5
plt.ylabel("Frequency")
plt.xlabel("Temperature Intervals")
# create the second panel and set current axis
plt.subplot(1,2, 2)
plt.title("BOXPLOT")
plt.boxplot(Dataset_1)
plt.ylabel("Temperature")
plt.show()
```



In [17]:
```
# Plot for datset 1
Dataset_2=[177,176,198,197,185,188,206,200,189,201,197,203]
plt.figure(figsize=(20,8))
plt.suptitle("Plot for Formulation 2 dataset")
plt.subplot(1,    2,       1)
plt.title("HISTOGRAM")
plt.hist(Dataset_2,bins=5,color='b')   # Number of bins taken=5
plt.ylabel("Frequency")
plt.xlabel("Temperature Intervals")
# create the second panel and set current axis
plt.subplot(1,2, 2)
plt.title("BOXPLOT")
plt.ylabel("Temperature")
plt.boxplot(Dataset_2)
plt.show()
```

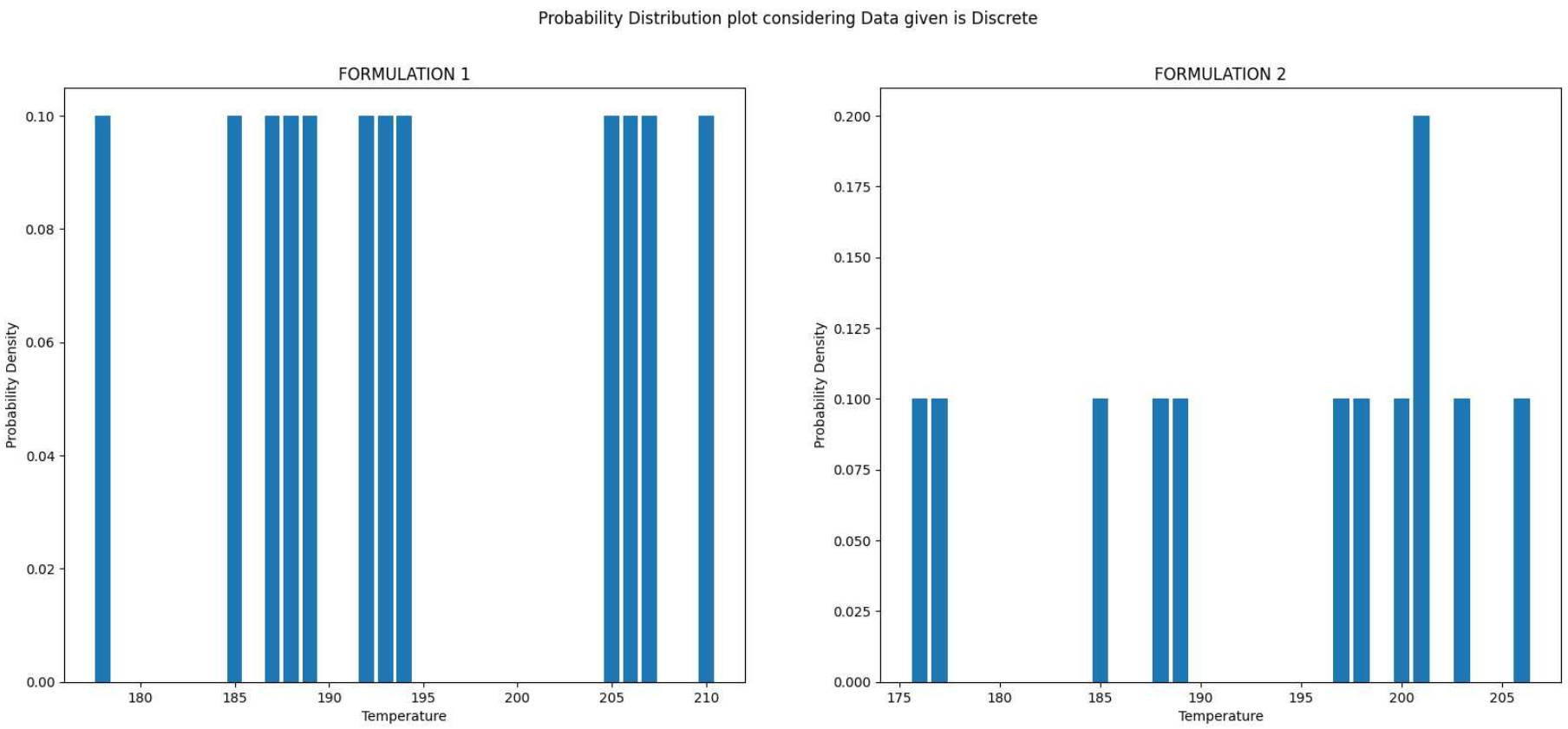


For the given data sets, plot the probability distribution functions.

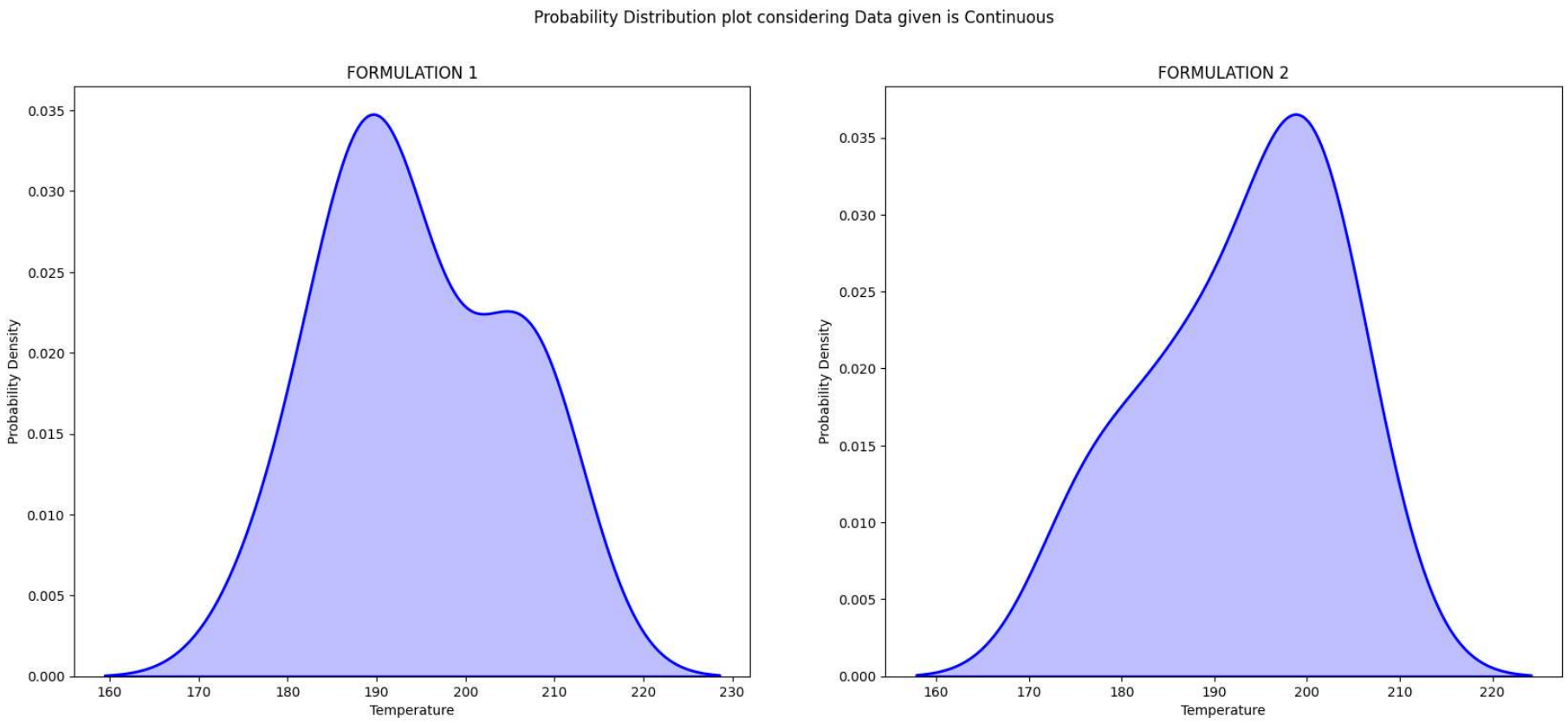
- data is discrete
- data is continuous

In [18]:
```
# If data is discrete
def frequencies(values): # Function to store frequencies
    frequencies = {} # Create dictionary
    for v in values:
        if v in frequencies:
            frequencies[v] += 1
        else:
            frequencies[v] = 1
    return frequencies

def probabilities(data, freqs): #Convert frequencies into probabilities
    probs = []
    for k,v in freqs.items():
        probs.append(round(v/len(data),1))
    return probs
# For Formulation_1
freq1 = frequencies(Dataset_1)
prob1 = probabilities(Dataset_1, freq1)
# For formulation_2
freq2 = frequencies(Dataset_2)
prob2 = probabilities(Dataset_2, freq2)
#Plotting
plt.figure(figsize=(20,8))
plt.suptitle("Probability Distribution plot considering Data given is Discrete")
plt.subplot(1,    2,       1)
plt.title("FORMULATION 1")
plt.bar(list(set(Dataset_1)),prob1)
plt.ylabel("Probability Density")
plt.xlabel("Temperature")
# create the second panel and set current axis
plt.subplot(1,2, 2)
plt.title("FORMULATION 2")
plt.bar(list(set(Dataset_2)),prob2)
plt.ylabel("Probability Density")
plt.xlabel("Temperature")
plt.show()
```



In [19]:
```
# If data is continuous
import seaborn as sns
#sns.displot(Dataset_1,kde=True)
plt.figure(figsize=(20,8))
plt.suptitle("Probability Distribution plot considering Data given is Continuous")
plt.subplot(1,    2,       1)
plt.title("FORMULATION 1")
sns.kdeplot(Dataset_1,fill=True, color='b',linewidth=2)
#sns.displot(Dataset_1)
plt.ylabel("Probability Density")
plt.xlabel("Temperature")
# create the second panel and set current axis
plt.subplot(1,2, 2)
plt.title("FORMULATION 2")
sns.kdeplot(Dataset_2,fill=True, color='b',linewidth=2)
plt.ylabel("Probability Density")
plt.xlabel("Temperature")
plt.show()
```



Q) Compare the mean, variance and expected values for the two data sets. Comment if the distribution is ‘Normal’ or not.

In [20]:
```
# Considering datasets as discrete
print("MEDIAN FOR FORMULATION_1 = ",statistics.median(Dataset_1),"\t\t\tMEDIAN FOR FORMULATION_2 = ",statistics.mean(D
ataset_2))
print("MEAN FOR FORMULATION_1 = ",statistics.mean(Dataset_1),"\t\t\tMEAN FOR FORMULATION_2 = ",statistics.mean(Dataset
_2))
print("VARIANCE FOR FORMULATION_1 = ",statistics.variance(Dataset_1),"\tVARIANCE FOR FORMULATION_2 = ",statistics.vari
ance(Dataset_2))
print("MODE FOR FORMULATION_1 = ",statistics.mode(Dataset_1),"\t\t\t\tMODE FOR FORMULATION_2 = ",statistics.mode(Datas
et_2))

'''
Now to find the expected Values of the two datasets. Here the datsets are discrete so the expected values of the datas
ets will be equal
to the mean of the datasets.

'''
```

```
MEDIAN FOR FORMULATION_1 =  192.5                    MEDIAN FOR FORMULATION_2 =  193.08333333333334
MEAN FOR FORMULATION_1 =  194.5                      MEAN FOR FORMULATION_2 =  193.08333333333334
VARIANCE FOR FORMULATION_1 =  103.54545454545455     VARIANCE FOR FORMULATION_2 =  98.99242424242425
MODE FOR FORMULATION_1 =  206                        MODE FOR FORMULATION_2 =  197
```

Out[20]:
```
'\nNow to find the expected Values of the two datasets. Here the datsets are discrete so the expected values of the d
atasets will be equal\nto the mean of the datasets.\n\n'
```

Q) Are the distribution Normal ?

A) For a normal distribution to be possible the following condition has to be satisfied i.e. [MEAN=MEDIAN=MODE]. In the dataset Formulation_1 we have Mean not equal to median. so the data set distribution is not Normal.

For the dataset Formulation_2 we have Mean=Median. However the mode is different in this case. So the dataset cannot be a Normal distribution.