HW 1

Kaahan Motwani

Question 1

A **The minimum possible top score is 5.**
Since the median of the data set is 5, 5 must be a part of the data set, and thus the minimum possible score cannot be any less than 5. Further, we are given that there are 5 students and that the median and mean are 5 and 4, respectively. Thus, we need to show that there exists a set of 5 students' scores, where the median is 5, the mean is 4, and the top score is 5. One such set that follows these constraints is: 5, 5, 5, 3, 2. The median is clearly 5, and the mean can be calculated to be $20/5 = 4$, and the highest score is **5**.

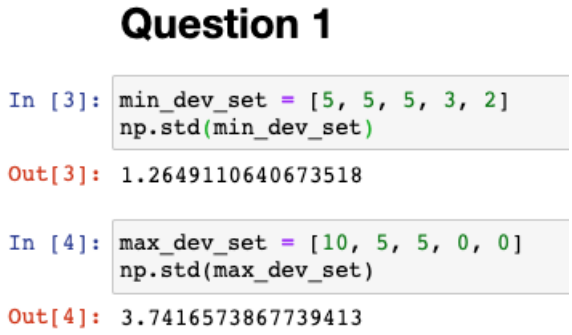
B **The maximum possible top score is 10.**
We are given that there are 5 students and that the median and mean are 5 and 4, respectively. We need to find a set where the median is 5, the mean is 4, and maximize the top possible score. Further, since the median of the data set is 5, 5 must be a part of the data set. Since we know the mean = 4 and there are 5 students, the sum of all scores should be 4 x 5 = 20. One such set that follows these constraints is: 10, 5, 5, 0, 0. The median is clearly 5, and the mean can again be calculated to be $20/5 = 4$, and the highest score that comes out is **10**.

C **The minimum possible standard deviation for this data set is 1.26**
The standard deviation numerically describes the extent to which the data items are close to the mean (defined by the square root of the squared distances from the mean of the data points, divided by the number of elements). Again, we are given that there are 5 students and that the median and mean are 5 and 4, respectively. We need to find a set where the data points' combined distances from the mean squared are minimized. We once again know that since 5 is the median, it must be in the middle of the data set. In order to minimize distance of the points from the mean (4), let us say that there are two more data points that are 5, which maintains the median of the set as 5. For the final (smallest) two data points, they must maintain a mean of 4; in order to do so, the total sum must be 5 x 4 = 20, but since the first three data points are 5, the last two numbers must sum to 5 to keep the mean of 4. To keep these two data points' squared differences from the mean minimized, let us choose 3 and 2 for the final data points. The combined squared distances from the

mean are $1^2 + 2^2 = 5$. If, for example, the last two points we chose were 4 and 1, we would have combined squared distanced from the mean of $3^2 + 0^2 = 9$; this is why we want to avoid values with greater distance from the mean, because they are much larger when squared. Hence, the final data set for this scenario is 5, 5, 5, 3, 2. Using standard deviation calculations in Python code, we can see from the image below that the minimum possible standard deviation is **1.26**.

D **The maximum possible standard deviation for this data set is 3.74**
The standard deviation numerically describes the extent to which the data items are close to the mean (defined by the square root of the squared distances from the mean of the data points, divided by the number of elements). Again, we are given that there are 5 students and that the median and mean are 5 and 4, respectively. We need to find a set where the data points' combined distances from the mean squared are maximized. We once again know that since 5 is the median, it must be in the middle of the data set. In order to maximize distance of the points from the mean (4), let us say that there are two data points that are 0, which allows the median to remain as 5 and maximize the distance from the mean (4). For the other (largest) two data points, they must maintain a mean of 4; in order to do so, the total sum must be 5 x 4 = 20, but since the first three data points are 5, 0, 0, the last two numbers must sum to 15 to keep the mean of 4. To keep these two data points' squared differences from the mean maximized, let us choose 10 and 5 for the final data points. The combined squared distances from the mean are $6^2 + 1^2 = 37$. If, for example, the last two points we chose were 8 and 7, we would have combined squared distanced from the mean of $4^2 + 2^2 = 20$; this is why we want to avoid values with a smaller combined squared distance from the mean, because they are much smaller when squared and summed. Hence, the final data set for this scenario is 10, 5, 5, 0, 0. Using standard deviation calculations in Python code, we can see from the image below that the minimum possible standard deviation is **3.74**.

Figure 1: These are the calculations of minimum and maximum possible standard deviations for this problem.



Question 2

A To represent a data point in standard coordinates, we use the formula

$$\hat{x} = \frac{(x - \mu)}{\sigma}$$

where $\hat{\boldsymbol{x}}$ is for a data point in a set that is in standard coordinates. In order to find the mean of this set, we must sum all points in this data set. Since we already know how to convert a point to standard coordinates, we will calculate the summation of all points using the formula above. The numerator in the above formula is

$$(x - \mu)$$

We know from a proven definition in the textbook that the signed difference between x and μ for all data points sums to a total of 0. Thus, since the numerator of our formula for mean of the standard set is 0, the mean is also **0**.

Next, let us observe the variance of the data set with standard coordinates. To calculate variance for a data set, we use the formula

$$\frac{(\hat{x} - \hat{\mu})^2}{N}$$

where N is the number of elements in the data set. From the previous proof, we already proved that $\mu = 0$, so our formula becomes $(\hat{x})^2$. Further, we already know that $\hat{x} = \frac{(x-\mu)}{\sigma}$ to represent a data point in standard coordinates, so we can substitute and our formula becomes the sum of

$$\frac{1}{N} \cdot \frac{(x - \mu)^2}{\sigma^2}$$

for all elements of data. Next, the standard deviation, σ, relates to the variance because variance = σ^2. Thus, using the formula for variance, $\frac{(x-\mu)^2}{N} = \sigma^2$, and therefore, $(x - \mu)^2 = \sigma^2 \cdot N$. Plugging this into the above equation, we get

$$\frac{1}{N} \cdot \frac{\sigma^2 \cdot N}{\sigma^2} = 1$$

. Finally, since our variance for a standard coordinates data set = 1 and variance = σ^2, we can say that the standard deviation, $\sigma = \sqrt{1} = \mathbf{1}$.

B **In this case, the data is right-skewed.** This is because the median of the data set is less than the mean of the data set. Since the mean is more affected by outliers than the median, the mean wavers more towards outliers (which are on the right tail of the distribution when right-skewed). Thus, since the mean = 0 and median = -1, the mean is greater than the median, which suggests that the data in this case is right-skewed.

Question 3

A **There are no outliers in the data for the Cost and Megawatts (MWatts) columns, but there are two outliers, 70.92 and 71.08, for the Date column.** In order to find this out, I used the definition of outliers; I calculated the interquartile range (IQR) for the data set and checked if any data points were more than 1.5 x IQR below or above the 25th and 75th percentiles, respectively. This can be seen in the image below with the provided Python code.

Figure 2: These are the calculations of IQR and checking for any outliers within the dataset for this problem.

Question 3 ¶

In [51]:
```
data = arff.loadarff('NuclearPlants.arff')
df = pd.DataFrame(data[0])
```

In [52]:
```
# Computing IQR for Cost
Q1 = df['Cost'].quantile(0.25)
Q3 = df['Cost'].quantile(0.75)
IQR = Q3 - Q1
for num in df['Cost']:
    if num > (Q3 + (1.5 * IQR)) or (num < Q1 - (1.5 * IQR)):
        print(num)
```

In [53]:
```
# Computing IQR for MWatts
Q1 = df['MWatts'].quantile(0.25)
Q3 = df['MWatts'].quantile(0.75)
IQR = Q3 - Q1
for num in df['MWatts']:
    if num > (Q3 + (1.5 * IQR)) or (num < Q1 - (1.5 * IQR)):
        print(num)
```

In [54]:
```
# Computing IQR for Date
Q1 = df['Date'].quantile(0.25)
Q3 = df['Date'].quantile(0.75)
IQR = Q3 - Q1
for num in df['Date']:
    if num > (Q3 + (1.5 * IQR)) or (num < Q1 - (1.5 * IQR)):
        print(num)
```

```
70.92
71.08
```

B **The mean cost of a power plant is $461.56. The standard deviation for the cost of a power plant is $170.12.** This can be seen in **Figure 3** below and is done using the mean() and std() properties of a Python Pandas DataFrame.

Figure 3: These are the calculations for the mean and standard deviation for the cost of a power plant.

```
In [33]: df["Cost"].mean()
Out[33]: 461.5603124999999

In [9]: df["Cost"].std()
Out[9]: 170.12066957318333
```

C **The mean cost per megawatt of a power plant is $0.57. The standard deviation for the cost per megawatt of a power plant is $0.19.** This can be seen in **Figure 4** below and is done using the mean() and std() properties of a Python Pandas DataFrame.

Figure 4: These are the calculations for the mean and standard deviation for the cost per megawatt of a power plant.

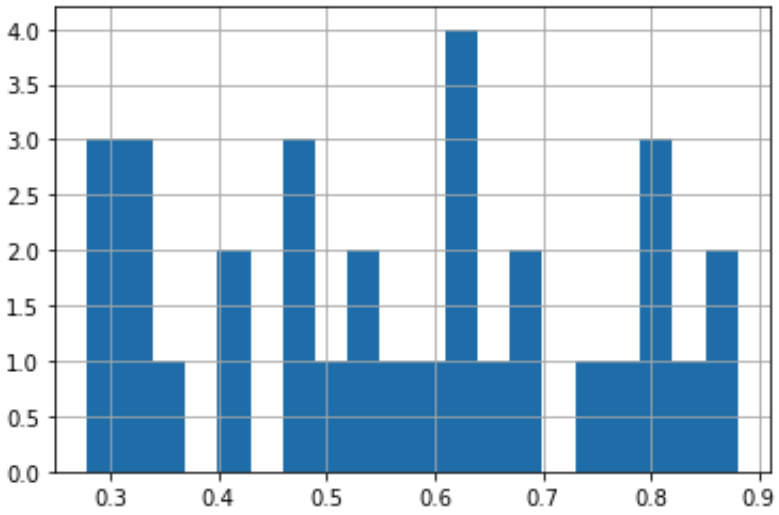
```
In [34]: (df["Cost"]/df["MWatts"]).mean()
Out[34]: 0.569735439641486

In [10]: (df["Cost"]/df["MWatts"]).std()
Out[10]: 0.1871243018960576
```

D **On the next page is the histogram for the cost per megawatt. The histogram is not skewed in either direction.** The reason it is not skewed is because by looking at the histogram, it can be concluded that the data is approximately uniformly distributed and symmetrical, and thus there is not significant enough skew to say that it is left or right skewed.

```
In [31]: df['Cost/MW'] = df["Cost"]/df["MWatts"]
```

```
In [33]: hist = (df['Cost/MW']).hist(bins=20)
```



Question 4

For this question, I created class-conditional histograms using Pandas DataFrames, which are using matplotlib in the background to create the class-conditional histograms for calories and sodium content. We can see the different bins from the legend and different colors for Beef, Meat, and Poultry. The histograms are visible on the next page in **Figure 5**.

Histogram 1 (Calories) For the first histogram, which depicts the number of calories for the three types of hot dogs, there are some trends and statistics that can be observed. First looking at the poultry (black bars), it can be seen that the data appears to be bimodal, with two clear peaks at approximately 100 calories and 145 calories. Further, it clearly has the lowest mean and median relative to the beef and meat histograms. Next, looking at the meat (blue bars), the meat clearly has the greatest mode of any of the three hot dog types, at approximately 138 calories. The meat data also seems to be skewed slightly to the left, since more of the data is on the right of the histogram and the left tail appears to be longer. This tells us that the mean is likely less than the median for the meat. Lastly, observing the beef data (orange bars), the data appears to be relatively symmetric, with not enough skew in either direction to definitively claim that it is left or right skewed.

Histogram 2 (Sodium) For the second histogram, which depicts the amount of sodium for the three types of hot dogs, there are some trends and statistics that can be observed. First looking at the poultry (black bars), it can be seen that the data appears to be bimodal, with two clear peaks at approximately 370 and 530. Next, looking at the meat (blue bars), it also appears to be bimodal, with two clear peaks at approximately 380 and 510. The meat also clearly has the lowest data point of any of the three hot dog types, as evidenced by the blue bar well below 200 on the left. Lastly, observing the beef data (orange bars), the data appears to be right-skewed , with much of the data on the left side, and the right tail much longer. This tells us that the mean is greater than the median since it will be more affected by outliers on the right (greater) side of the data.

Figure 5: The two class-conditional histograms.

Question 4

In [50]:
```
df = pd.read_csv('hotdogs.csv', header=None)
df.columns = ["Type", "Calories", "Sodium", "CUnused"]
del df['CUnused']
```

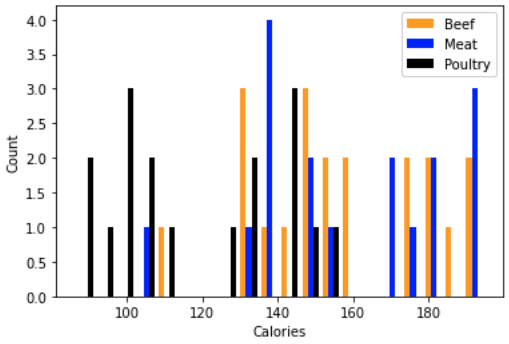
In [26]:
```
# Class-conditional histogram

n_bins = 20

tmpdf1 = df[df["Type"]=="Beef"]
tmpdf2 = df[df["Type"]=="Meat"]
tmpdf3 = df[df["Type"]=="Poultry"]


colors = ['orange', 'blue', 'black']
labels = ['Beef','Meat','Poultry']
x_multi = [tmpdf1["Calories"],tmpdf2["Calories"],tmpdf3["Calories"]]
plt.hist(x_multi, n_bins, histtype='bar',color=colors,label=labels)
plt.legend(prop={'size': 10})
plt.xlabel("Calories")
plt.ylabel("Count")
```

Out[26]:
```
Text(0, 0.5, 'Count')
```

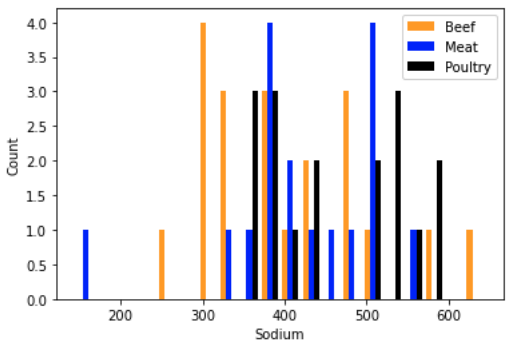


In [27]:
```
n_bins = 20

tmpdf1 = df[df["Type"]=="Beef"]
tmpdf2 = df[df["Type"]=="Meat"]
tmpdf3 = df[df["Type"]=="Poultry"]


colors = ['orange', 'blue', 'black']
labels = ['Beef','Meat','Poultry']
x_multi = [tmpdf1["Sodium"],tmpdf2["Sodium"],tmpdf3["Sodium"]]
plt.hist(x_multi, n_bins, histtype='bar',color=colors,label=labels)
plt.legend(prop={'size': 10})
plt.xlabel("Sodium")
plt.ylabel("Count")
```

Out[27]:
```
Text(0, 0.5, 'Count')
```



Question 5

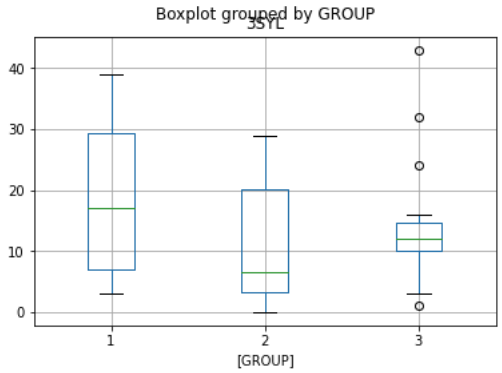
A Plotted below is the box plot that compares the number of three or more syllable words for the ads in magazines in these three groups; outliers are also shown in the box plots, if they exist. In this box plot, it can be observed that the median of 3 syllable words is highest in Group 1, but with Group 1 also having the largest interquartile range. Group 2 has the lowest median of the three groups, while Group 3 has by far the smallest IQR for the number of 3 syllable words. Group 3 also has 4 outliers (1.5 x IQR rule) in its data.

Question 5

In [49]:
```
df = pd.read_csv('mag_ads.csv', header=None)
df.columns = ["Words", "SEN", "3SYL", "MAG", 'GROUP']
```

In [47]:
```
df.boxplot(column=['3SYL'], by=['GROUP']) # 3 syllable words grouped by GROUP
```

Out[47]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa7b117ea10>
```



B Plotted below is the box plot that compares the number of sentences appearing in the ads in magazines in these three groups; outliers are also shown in the box plots, if they exist. In this box plot, it can be observed that the median number of sentences is highest in Group 1, but very similar across all groups. However, Group 2 has the largest interquartile range, while Group 3 has by far the smallest IQR. Group 3 also has 3 outliers (1.5 x IQR rule) in its data for the number of sentences.

```
In [48]: df.boxplot(column=['SEN'], by=['GROUP']) # Number of sentences grouped by GROUP
Out[48]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa7b12a6fd0>
```