Part 1

I first installed pandas using pip installer and created dataframe for specific columns using the following command:

```
cols = [0,1,5,6,7,8,117]

df = pd.read_csv('LANGEVIN_DATA.txt',delimiter=' ',usecols =
cols, names = ['Time', 'Occupant Number', 'INDOOR Ambient
Temp.', 'INDOOR Relative Humidity', 'INDOOR Air Velocity',
'INDOOR Mean Radiant Temp.', 'Predicted Mean Vote'])
```

I simply used the following command to output the desired results as csv:

```
df.to_csv('processed_data.csv',index = False)
```

Part 2

For getting various statistics of the data like mean, median, mode and standard deviation I used respective pandas functions like mean(), median(), mode() and std() in the dataframe.

```
import extract
print extract.df.mean()
print extract.df.median()
print extract.df.mode().iloc[0]
print extract.df.std()
```

The values for mean, median, mode, standard deviation are as follows:
****Data Frame Mean****

Time 735262.500000
Occupant Number 12.500000
INDOOR Ambient Temp. 22.643550
INDOOR Relative Humidity 36.298591
INDOOR Air Velocity 0.027971
INDOOR Mean Radiant Temp. 22.631670
Predicted Mean Vote -0.473796
dtype: float64

****Data Frame Median****

Time 735262.499999
Occupant Number 12.500000
INDOOR Ambient Temp. 22.823889
INDOOR Relative Humidity 31.015500
INDOOR Air Velocity 0.026924
INDOOR Mean Radiant Temp. 22.806852
Predicted Mean Vote -0.442999
dtype: float64

****Data Frame Mode****

Time 735080.000000
Occupant Number 1.000000
INDOOR Ambient Temp. 23.063889
INDOOR Relative Humidity 31.015500
INDOOR Air Velocity 0.025908
INDOOR Mean Radiant Temp. 23.231392
Predicted Mean Vote -0.111416
Name: 0, dtype: float64

****Data Frame Standard Deviation****

Time 105.369493
Occupant Number 6.922191
INDOOR Ambient Temp. 1.816115
INDOOR Relative Humidity 13.820629
INDOOR Air Velocity 0.009337
INDOOR Mean Radiant Temp. 1.829641
Predicted Mean Vote 0.562146
dtype: float64

For comparing the runtime, I used the timeit module in python. Following is the code which I wrote to do it:

```
start_time = timeit.default_timer()
getStats()
printHistograms()
print "***Calculating the time***"
print(timeit.default_timer() - start_time)
```

As it can be seen, I am starting timer right before calling functions for 2a and I am calculating the time difference to calculate the time taken for execution.
I got the time for this one as 3.04429793358. I used similar logic in old program and got the time of 18.2330495871. Hence using pandas, the time taken for solving statistical problems got significantly reduced.

Also, I was able to generate the following histograms by using pandas “hist” function for different columns: `'Time', 'Occupant Number', 'INDOOR Ambient Temp.', 'INDOOR Relative Humidity', 'INDOOR Air Velocity', 'INDOOR Mean Radiant Temp.' and 'Predicted Mean Vote'.`

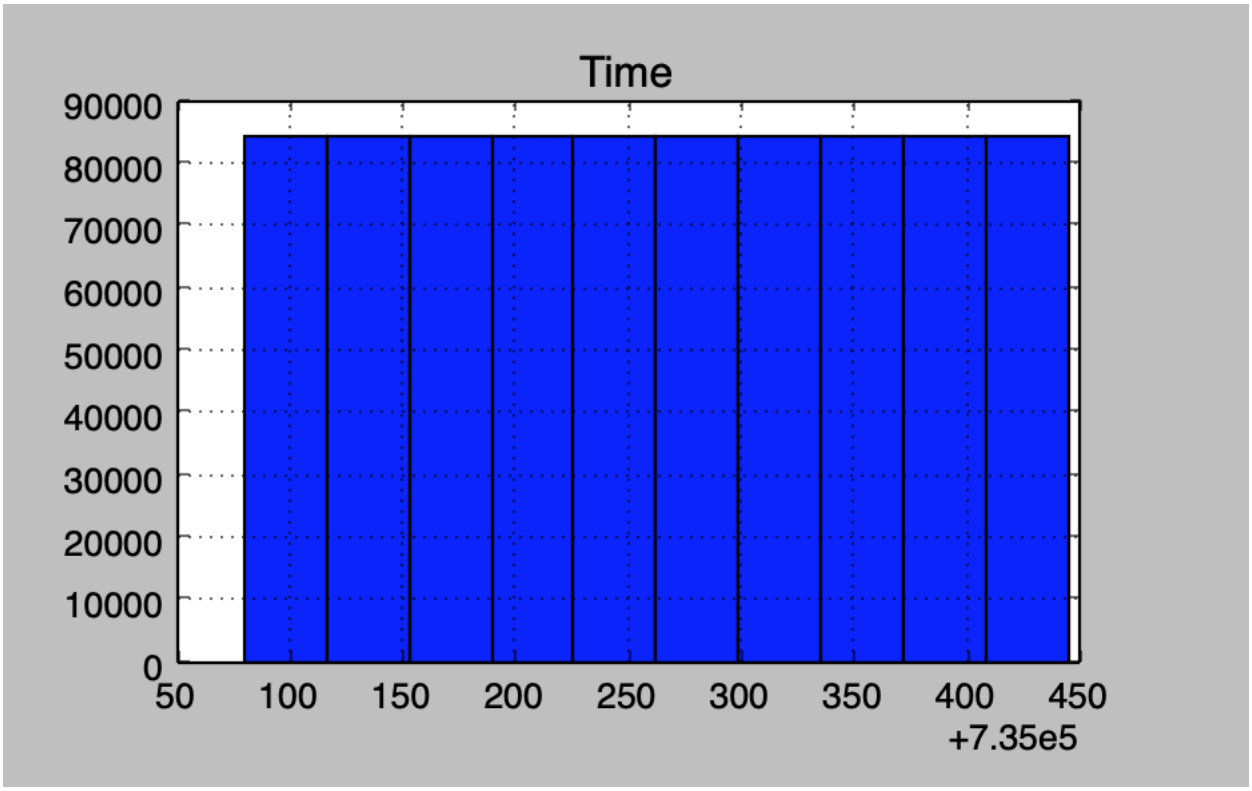

Figure 1: Time Histogram

While generating the histogram for the column named "time", I could see that there were equal number of value in each range.
So this means that the column "time is uniformly distributed"

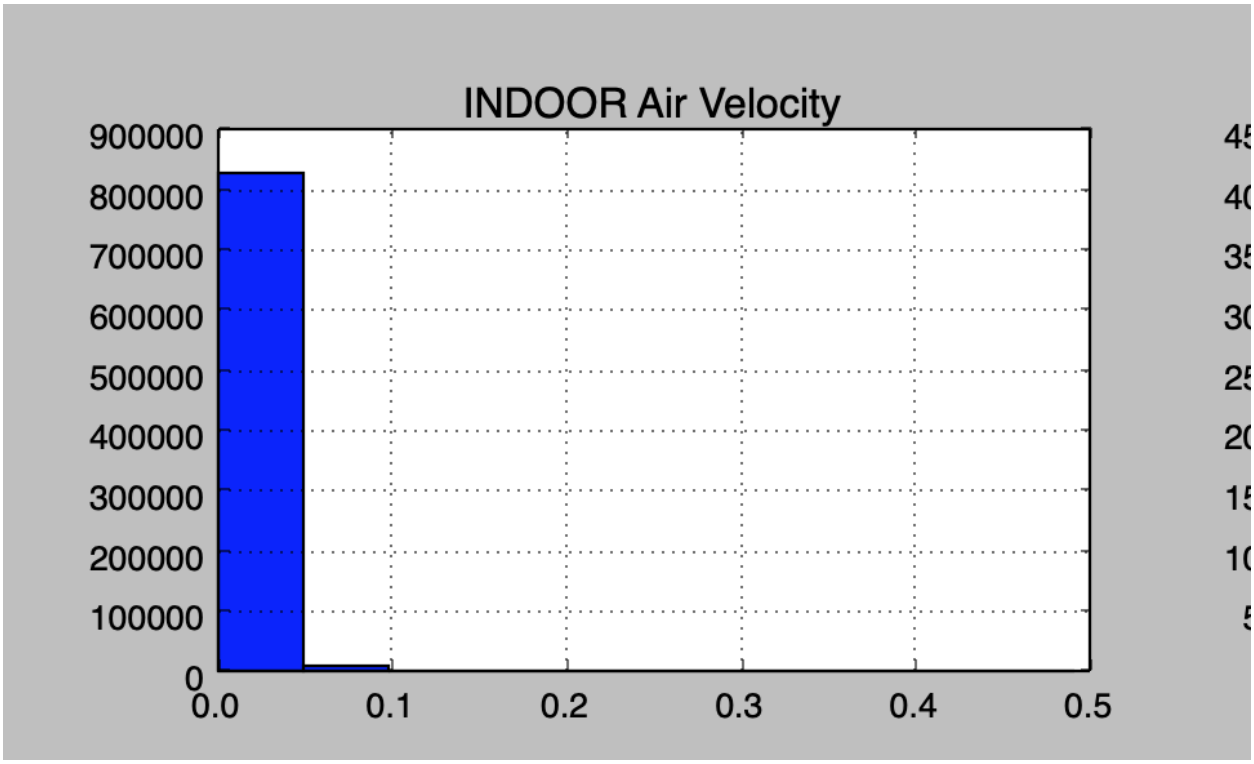

Figure 2: Indoor Air Velocity Histogram

The column indoor Air Velocity had numerous occurence of value 0.025908 as it can be seen in the histogram above. This can also be explained by the value of mode which came to be 0.025908.

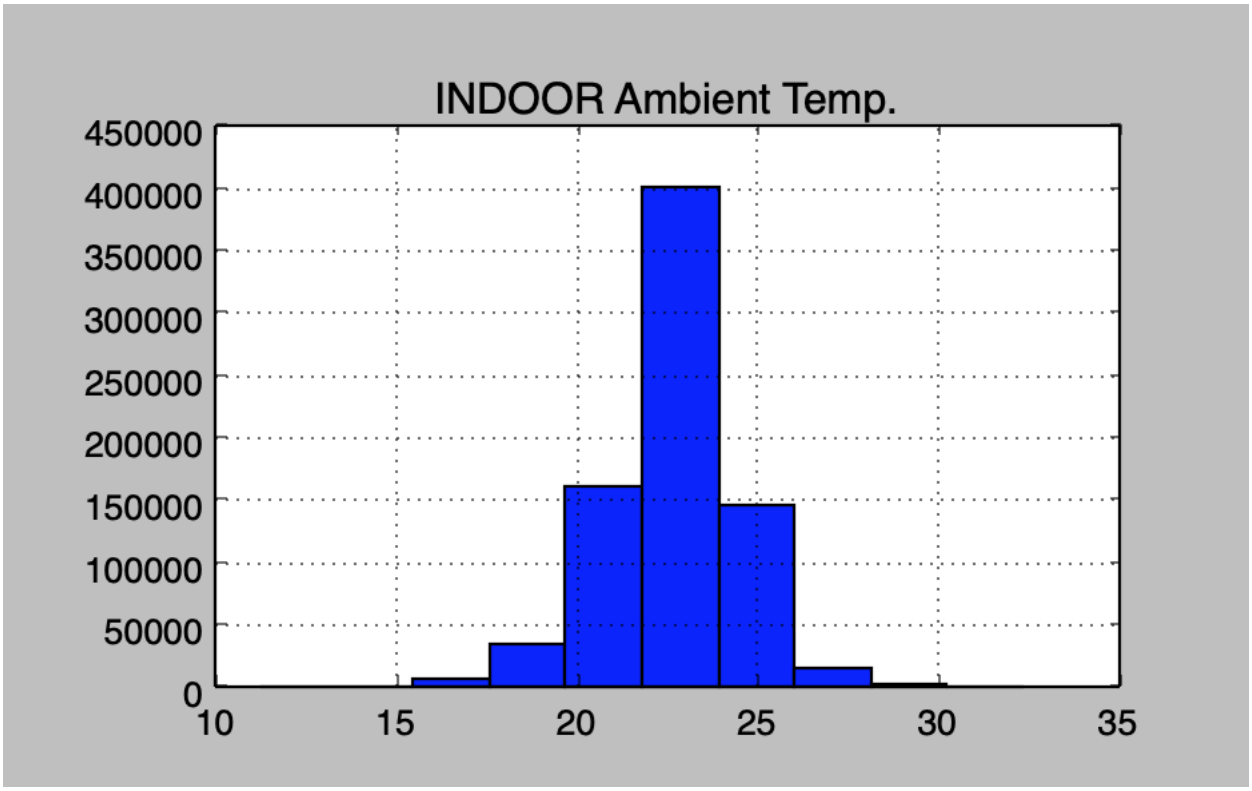

Figure 3: Indoor Ambient Temperature Histogram

The column indoor ambient temperature looked to be normally distributed which could be seen by the bell curved histogram above

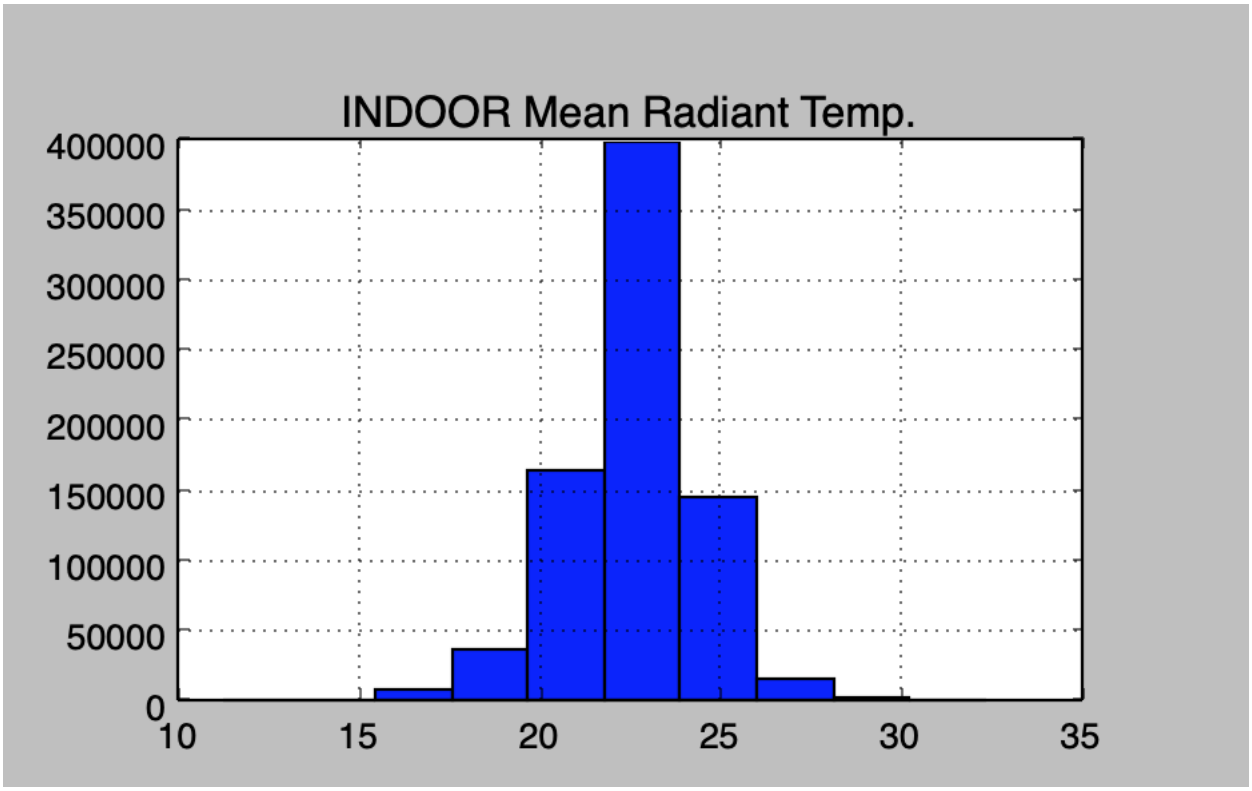

Figure 4: Indoor Mean Radiant Temperature Histogram

The column Indoor Mean Radiant Temperature also had normally distributed data which can be seen in the bell curve above.

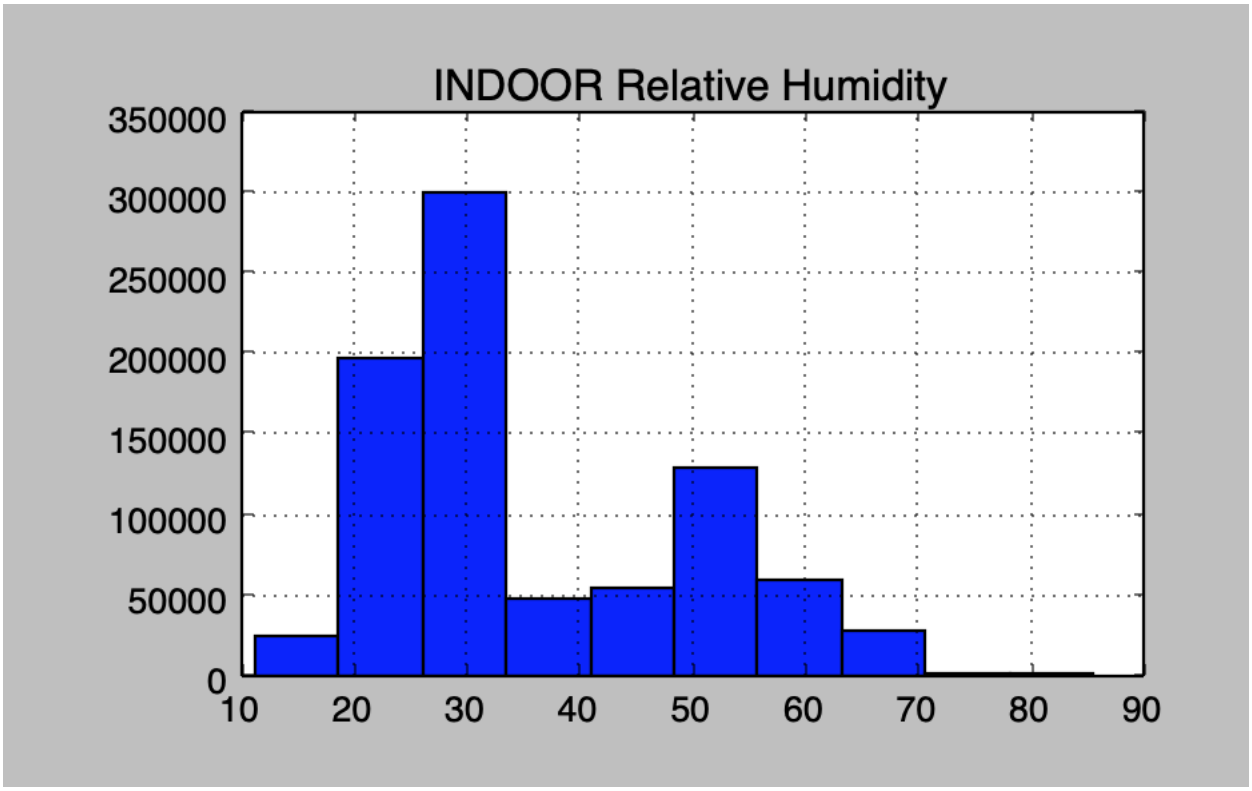

Figure 5: Indoor Relative Humidity Histogram

The column Indoor relative humidity looked to be denser in the region of 22-23, 30-32 and 54-55 which can be seen in the histogram.

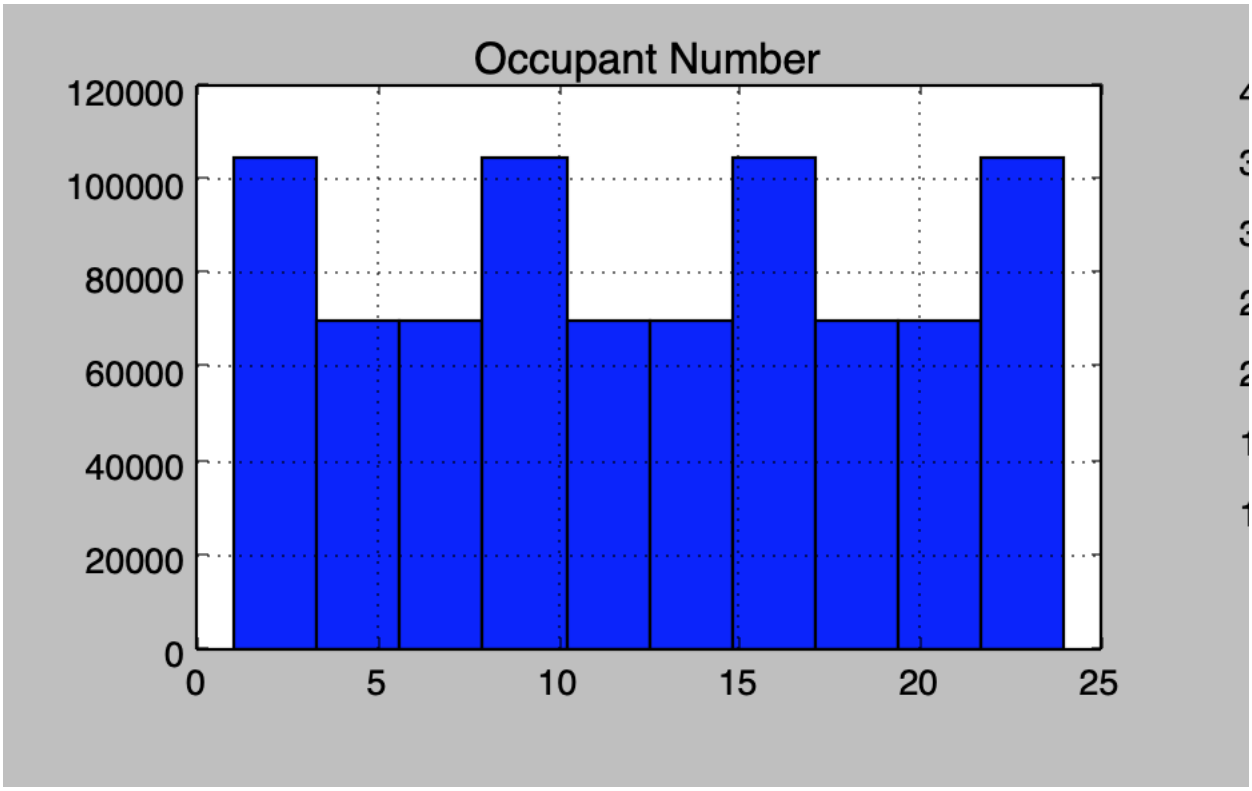

Figure 6: Occupant Number Histogram

While generating the histogram for the column named "occupant number", I could see that there were equal number of value in each range (roughly 17,000 in 1-2,2-3 all the way until 22-23 and almost double (about 35000) in the range 23-24).

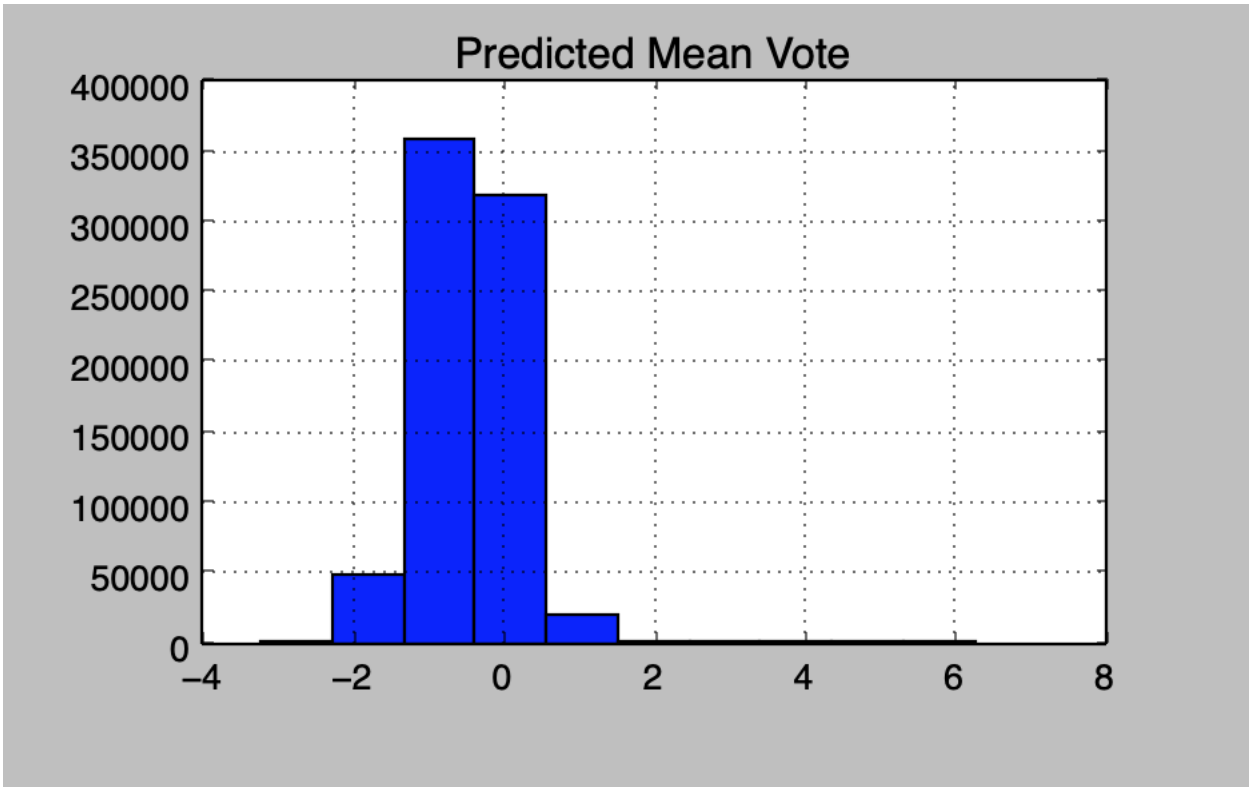

Figure 7: Predicted Mean Vote Histogram

The histogram shows that the distribution looks more like normal for this column.