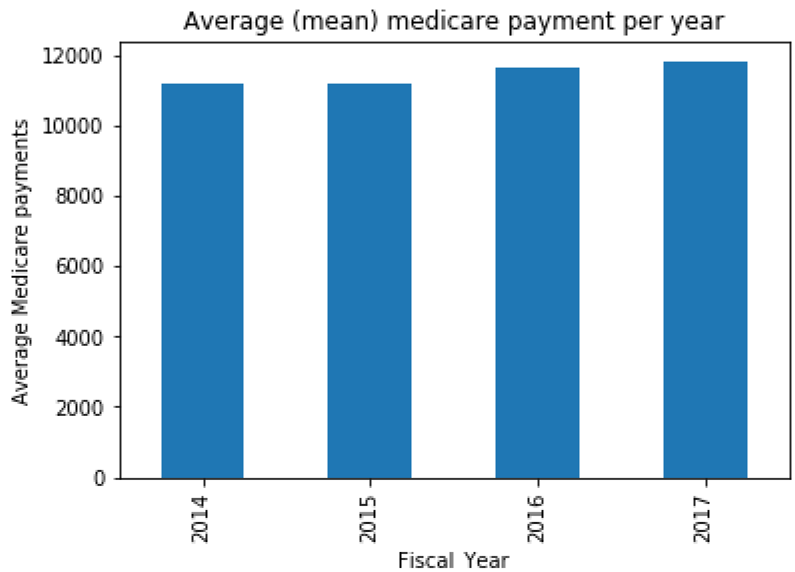
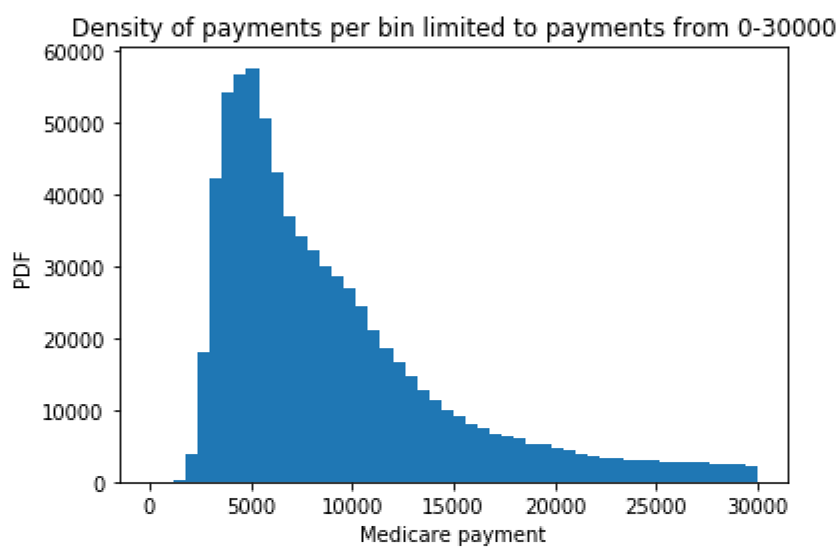
Statistical Analysis

I looked at the mean and median medicare payment per year. The values themselves were different but both graphs looked very similar as below:



I checked to see if the data is in a normal distribution. It is rather noticeably right tailed, which means that my calculated mean and median will read high. I will need to take this into account for my analysis.



To determine if there was a statistically significant change, I completed bootstrap permutation hypotheses testing (detailed below findings) for 2014-2015, 2015-2016, and 2016-2017. I do realize that bootstrapping assumes a normal distribution, which my data is not.

Each test had a null hypothesis that the years were the same and an alternative hypothesis that they were not the same. Results were as follows:

- 2014-2015: not statistically significant, cannot reject the null hypothesis
- 2015-2016: statistically significant, can reject the null
    - This was also the biggest difference with a change of ~$440
    - Likely the only one that could have practical significance
- 2016-2017: statically significant, can reject the null
    - May not be practically significant, only a difference of ~$160

To complete the bootstrap testing, I utilized the following method (screenshots of actual code below description - this will also be available in a jupyter notebook):

- Made a function to call bootstrap replicates by first creating an empty numpy array.
    - Then for every value in the range of the length of the data, I used np.random.choice to pull random values from that data and appended the (in this case) mean of that sample to the numpy array
    - The function returns the array, now with bootstrap replicates.
- Then I created a pandas series for each individual year and calculated their individual means as well as the difference of means.
- I also created a series for each group and calculated those means.
- I shifted the arrarys to make all means equal, drew bootstrap replicates, and calculated a p-value.

I also completed an ANOVA to provide a double check of my data. The p-values calculated with this were consistent with my conclusions from the bootstrapping testing.

In [14]:
```python
def draw_bs_reps(data, func, size=1):
    bs_replicates = np.empty(size)

    for i in range(size):
        sample = np.random.choice(data, len(data))
        bs_replicates[i] = func(sample)

    return bs_replicates
```

I created

- a pandas series for each individual year
- calculated the individual means
- a series for each grouping (2014-2015, 2015-2016, 2016-2017)
- calculated the combined means of my groups
- calculated the difference of means

In [15]:
```python
#individual series and means
all_2014 = df1['Average Medicare Payments']
mean_2014 = np.mean(df1['Average Medicare Payments'])

all_2015 = df2['Average Medicare Payments']
mean_2015 = np.mean(df2['Average Medicare Payments'])

all_2016 = df3['Average Medicare Payments']
mean_2016 = np.mean(df3['Average Medicare Payments'])

all_2017 = df4['Average Medicare Payments']
mean_2017 = np.mean(df4['Average Medicare Payments'])

#combined groups and means
comb_1415 = pd.concat([all_2014, all_2015])
comb_mean_1415 = np.mean(comb_1415)

comb_1516 = pd.concat([all_2015, all_2016])
comb_mean_1516 = np.mean(comb_1516)

comb_1617 = pd.concat([all_2016, all_2017])
comb_mean_1617 = np.mean(comb_1617)

#differences of means
diff_means_1415 = np.mean(all_2014) - np.mean(all_2015)
diff_means_1516 = np.mean(all_2015) - np.mean(all_2016)
diff_means_1617 = np.mean(all_2016) - np.mean(all_2017)
```

I then shifted the arrays to make all means equal

In [16]:
```
#shifts for 2014-2015
shift_14 = all_2014 - mean_2014 + comb_mean_1415
shift_15_1 = all_2015 - mean_2015 + comb_mean_1415

#shifts for 2015-2016
shift_15_2 = all_2015 - mean_2015 + comb_mean_1516
shift_16_1 = all_2016 - mean_2016 + comb_mean_1516

#shifts for 2016-2017
shift_16_2 = all_2016 - mean_2016 + comb_mean_1617
shift_17 = all_2017 - mean_2017 + comb_mean_1617
```

Followed by calculating bootstrap replicates

In [17]:
```
#bootstrap reps for 2014-2015
reps_14 = draw_bs_reps(shift_14, np.mean, size=1000)
reps_15_1 = draw_bs_reps(shift_15_1, np.mean, size=1000)

#difference of means
diff_1415 = reps_14 - reps_15_1

#calculating p value
p_1415 = np.sum((diff_1415) >= diff_means_1415) / len(diff_1415)

print('p-value for 2014-2015 =', p_1415)
```

```
p-value for 2014-2015 = 0.464
```

For the above, I cannot reject the null hypothesis (that 2014 and 2015 are the same). This leads me to believe that the difference is not statistically significant and indeed, when I look at the two means, they are incredibly similar, only $5 different

In [19]:
```
#bootstrap reps for 2015-2016
reps_15_2 = draw_bs_reps(shift_15_2, np.mean, size=1000)
reps_16_1 = draw_bs_reps(shift_16_1, np.mean, size=1000)

#difference of means
diff_1516 = reps_15_2 - reps_16_1

#calculating p value
p_1516 = np.sum((diff_1516) <= diff_means_1516) / len(diff_1516)

print('p-value for 2015-2016 =', p_1516)
```

```
p-value for 2015-2016 = 0.0
```

In [20]:
```
#bootstrap reps for 2016-2017
reps_16_2 = draw_bs_reps(shift_16_2, np.mean, size=1000)
reps_17 = draw_bs_reps(shift_17, np.mean, size=1000)

#difference of means
diff_1617 = reps_16_2 - reps_17

#calculating p-value
p_1617 = np.sum((diff_1617) <= diff_means_1617) / len(diff_1617)

print('p-value for 2016-2017 =', p_1617)
```

```
p-value for 2016-2017 = 0.0
```

For the 2015-2016 and 2016-2017 comparisons, we can reject the null hypothesis and state that there is a statistically significant difference.

In terms of practical significance, things become a little more grey. We see that the difference form 2015 to 2016 is just over $440 more on average, but only about $160 more from 2016 to 2017.