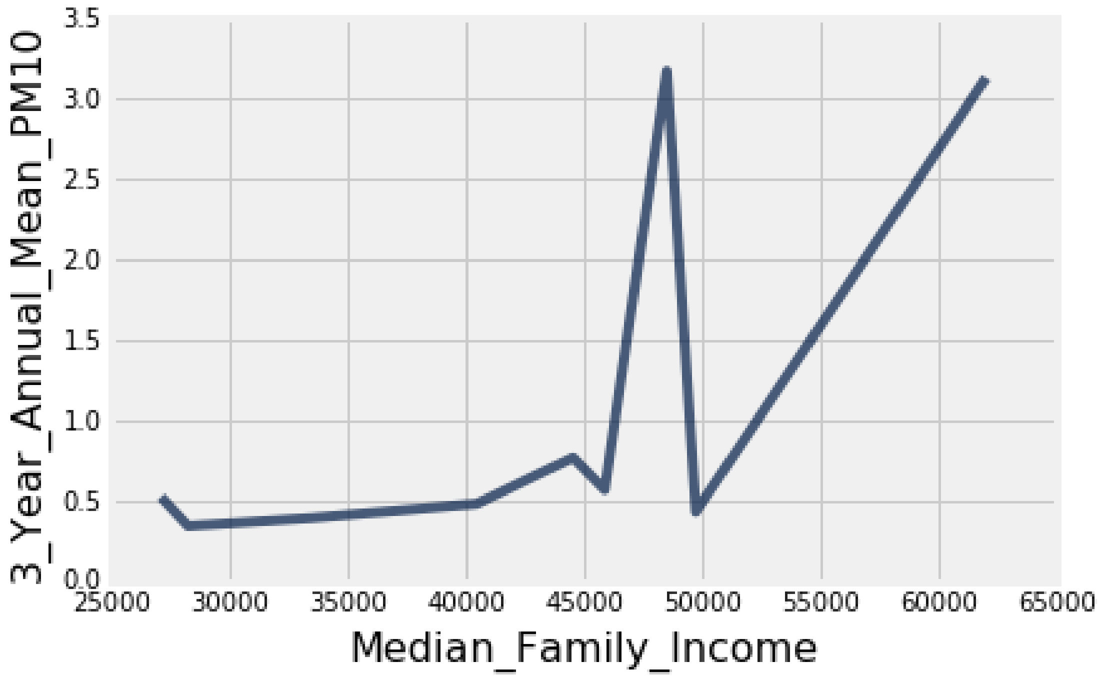
9 Counties in Maryland (MD) -Dongchan Yang (Jaden)

March 3, 2017

1 Median Family Income and 3 Year annual PM10 Graph

In [67]:

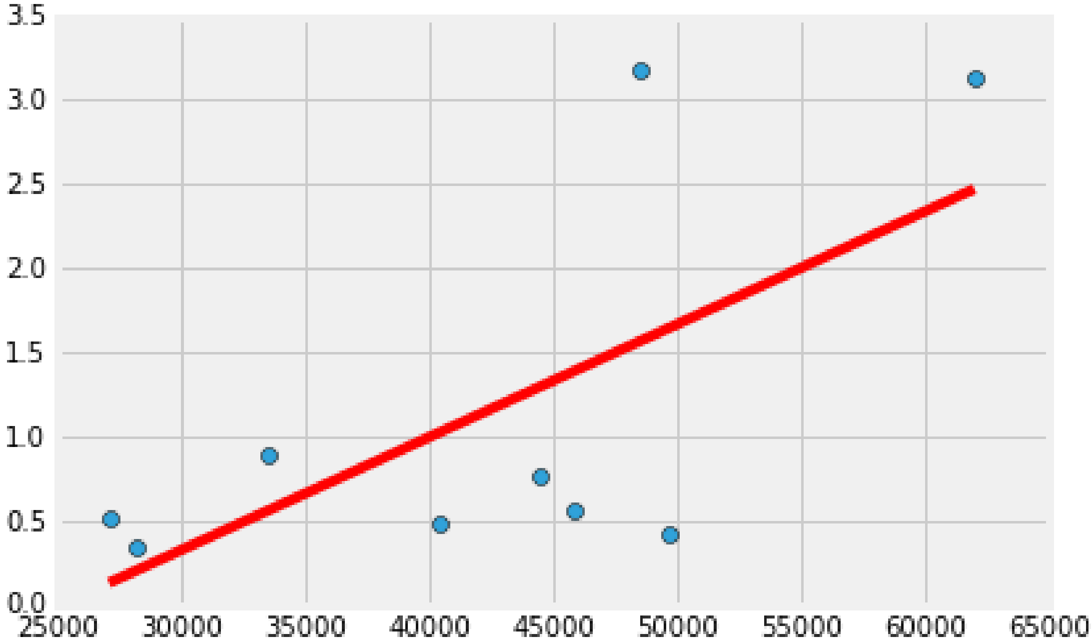


1.0.1 Based on 3 year annual mean PM10 and median family income data in 9 counties, I created this graph to find the relationship between income and higher pollution (=higher PM10). Interestingly, higher income tends to be higher PM10 average. However, plots were ambiguous so I created a regression model.

2 Regression Model of Income and PM10

In [27]:

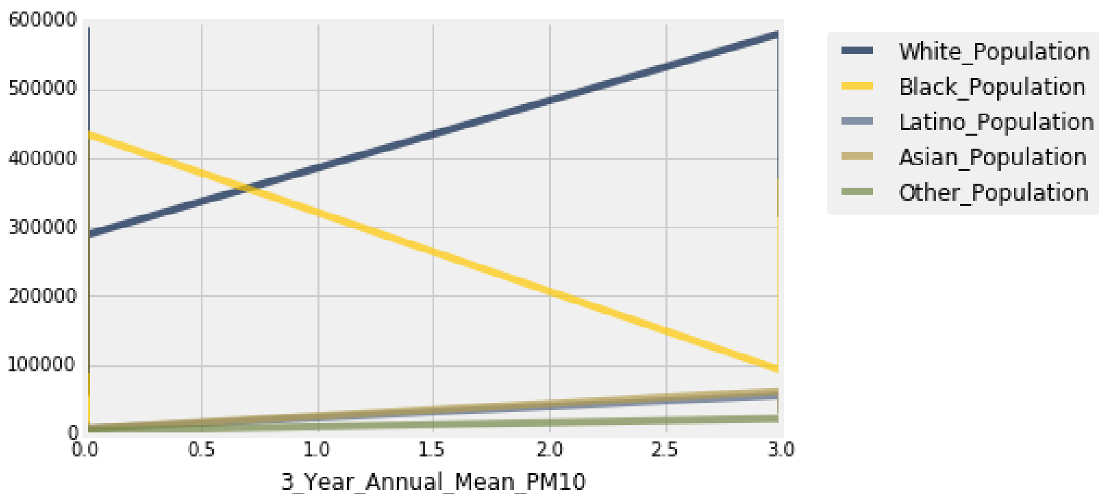
```
Out[27]: [<matplotlib.lines.Line2D at 0x7fb2dcdb59b0>]
```



2.0.2 In this model, x-axis is median family income and y-axis is 3 year annual mean PM10 (log). Each blue dot represents relations between income and PM10 in each county. Redline, which is a regression line, is sloping up with 0.654622180564 r-value. It means as income increases, PM10 is more likley to increase. So, 1 and 2 graphs show that family income and annual average PM10 are interrelated in Maryland, and higher family income county is more likely to have a higher PM10.

3 Race and 3 Year Annual Mean PM10 Graph (extra credit)

```
In [62]:
```



3.0.3 In this graph, x-axis is 3 year annual mean PM10, and y-axis is a population size. On my PM10 excel file, I converted race percentage into the actual population size by race and each county, and I created a graph between PM10 and population size by race. In Maryland, there are not many Asain, Latino, and other races compared to Black and White so I did not take Latino, Asian, and other race groups into account when I interpret this graph.

3.0.4 Interestingly, this graph shows that counties where a majority of population is White and less Black people tend to have a higher PM10 level. I could not find the relationship between higher PM10 level and non-white race group in Maryland.

3.1 In conclusion, counties where people have higher median family house income and more White population are more likely to have a higher annual average PM10 level.

4 ==

5 Below is the Original Work on Python

6 ==

```
In [2]: from datascience import Table
        %matplotlib inline
        import matplotlib.pyplot as plt
        import numpy as np
        plt.style.use('fivethirtyeight')

        import pandas as pd

        import datascience as ds
        ds.__version__
```

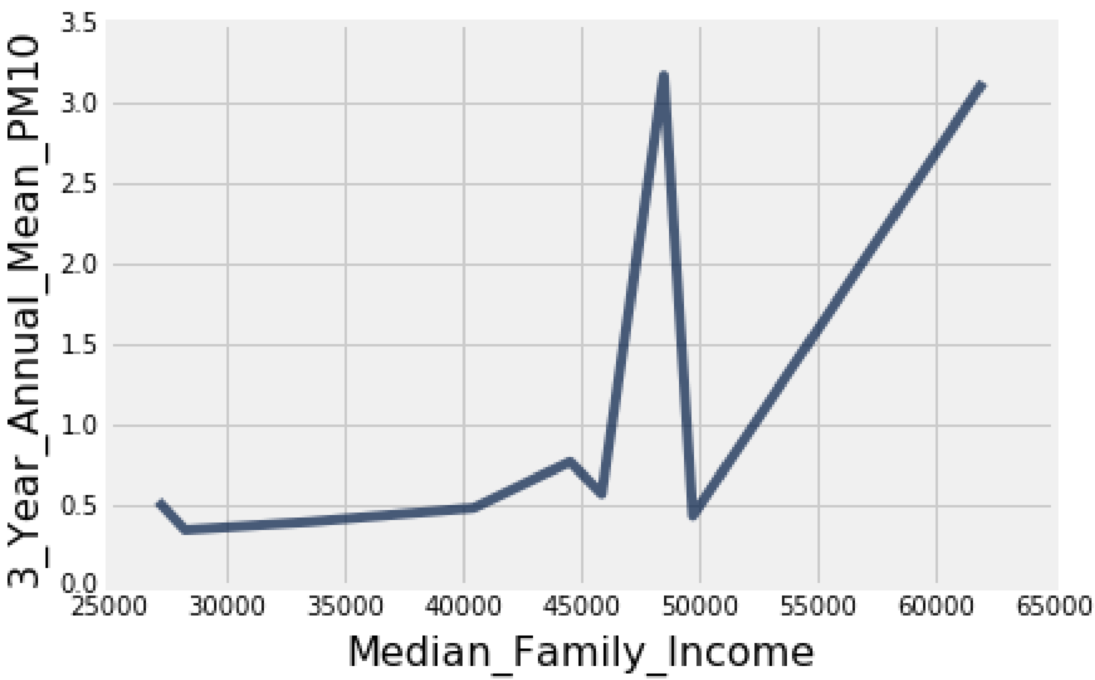
```
Out[2]: '0.8.2'
```

```
In [58]: Maryland_Income = Table(['Median_Family_Income'
                                  , '3_Year_Annual_Mean_PM10']).with_rows([[27069,
                                                                           ,[45847, 0.
                                                                           ,[61988, 3.
                                                                           ,[33449, 0.
                                                                           ,[28217, 0.

         Maryland_Income
```

```
Out[58]: Median_Family_Income | 3_Year_Annual_Mean_PM10
         27069                | 0.518967
         40420                | 0.476607
         45847                | 0.564388
```

```
49706                   | 0.427164
61988                   | 3.13114
44502                   | 0.764268
33449                   | 0.389505
48471                   | 3.17388
28217                   | 0.33875
```

```
In [64]: Maryland_Income.select(["Median_Family_Income"
                                , "3_Year_Annual_Mean_PM10"])
                                .plot("Median_Family_Income"
                              , "3_Year_Annual_Mean_PM10")
```

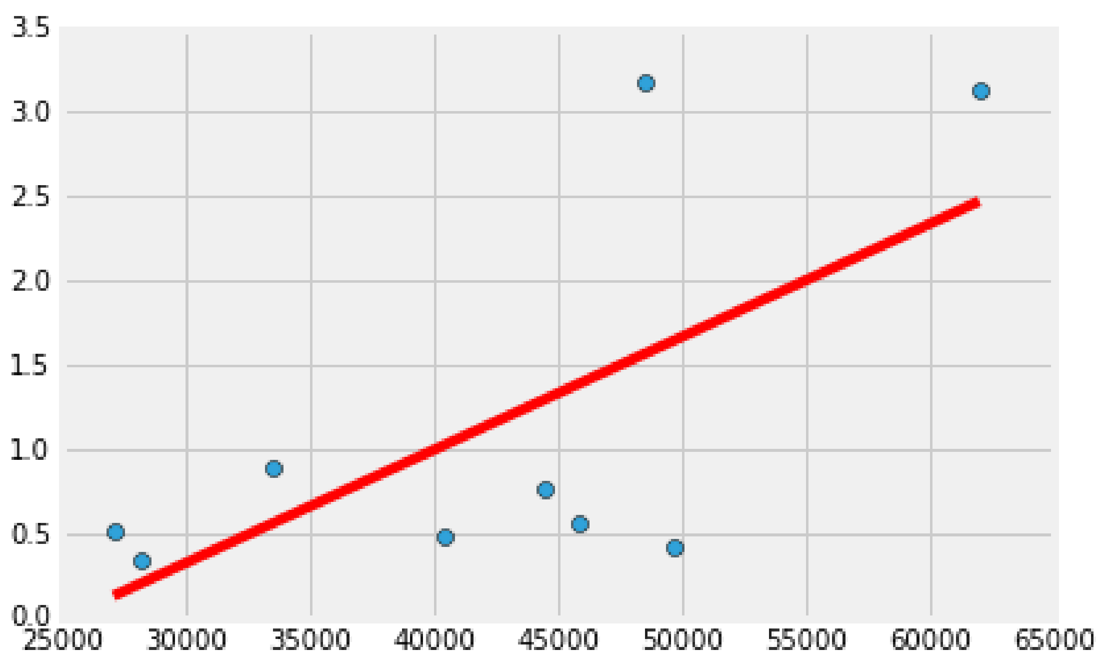



```
In [15]: from numpy import *
```

```
In [16]: x = array([27069, 28217, 33449, 40420, 44502, 45847
                   , 48471, 49706, 61988])
         y = array([0.518967, 0.33875, 0.89505, 0.476607
                   , 0.764268, 0.564388
                   , 3.17388, 0.427164, 3.13114])
```

```
In [17]: print(x)
         print(y)

[27069 28217 33449 40420 44502 45847 48471 49706 61988]
[ 0.518967  0.33875   0.89505   0.476607  0.764268  0.564388  3.17388
  0.427164  3.13114 ]
```

```
In [18]: from scipy.interpolate import *

In [19]: p1 = polyfit(x,y,1)

In [20]: print(p1)   ## find Slope and Intercept

[  6.70918161e-05  -1.68694097e+00]


In [21]: from matplotlib.pyplot import *

In [22]: %matplotlib inline

In [23]: p2 = polyfit(x,y,2)
         p3 = polyfit(x,y,3)

In [24]: print(p1)
         print(p2) ## coefficietn
         print(p3)

[  6.70918161e-05  -1.68694097e+00]
[  2.72372079e-09  -1.67156556e-04   3.04208868e+00]
[  1.10921351e-15   2.57449976e-09  -1.60754007e-04   2.95506627e+00]


In [65]: plot(x,y,'o')
         plot(x,polyval(p1,x),'r-')

Out[65]: [<matplotlib.lines.Line2D at 0x7fb2dc7a2748>]
```



```
In [29]: from scipy.stats import *
         slope,intercept,r_value,p_value,std_err = linregress(x,y)
         print(r_value)
```

```
0.654622180564
```

```
In [31]: print(slope)
```

```
6.70918160754e-05
```

```
In [61]: Maryland_Race = Table(['Total_Population','White_Population'
         , 'Black_Population', 'Latino_Population'
         , 'Asian_Population', 'Other_Population'
         , '3_Year_Annual_Mean_PM10']).with_rows([[74946, 72955, 1535
                                                   , 319, 321, 135, 0.518967]
                                               ,[71347, 67450, 3240
                                                 , 635, 310, 347, 0.476607]
                                               ,[150208, 139909, 8010
                                                 , 1713, 1510, 779, 0.5643881]
                                               ,[427239, 365953, 50525
                                                 , 6815, 7675, 3086, 0.4271639]
                                               ,[757027, 580635, 92267
                                                 , 55684, 61981, 22144, 3.1311369]
                                               ,[692134, 587898, 85451
                                                 , 8131, 15544, 3241, 0.7642682]
                                               ,[74339, 56755, 16573
                                                 , 610, 671, 340, 0.3895048]
                                               ,[729268, 314616, 369791
                                                 , 29983, 28255, 16606, 3.1738785]
                                               ,[736014, 287753, 435768
                                                 , 7602, 7942, 4551, 0.3387501]])
         Maryland_Race
```

```
Out[61]: Total_Population | White_Population | Black_Population | Latino_Populatio
         74946            | 72955            | 1535             | 319
         71347            | 67450            | 3240             | 635
         150208           | 139909           | 8010             | 1713
         427239           | 365953           | 50525            | 6815
         757027           | 580635           | 92267            | 55684
         692134           | 587898           | 85451            | 8131
         74339            | 56755            | 16573            | 610
         729268           | 314616           | 369791           | 29983
         736014           | 287753           | 435768           | 7602
```

```
In [66]: type(Maryland_Race.column("3_Year_Annual_Mean_PM10")[1])
         Maryland_Race.apply(np.int, "3_Year_Annual_Mean_PM10")
         Maryland_Race["3_Year_Annual_Mean_PM10"] = Maryland_Race.apply(np.int
                                                                        , "3_Year_A
         type(Maryland_Race.column("3_Year_Annual_Mean_PM10")[1])
         Maryland_Race.select(["3_Year_Annual_Mean_PM10", "White_Population"
                              , "Black_Population", "Latino_Population"
                              , "Asian_Population"
                              , "Other_Population"]).plot("3_Year_Annual_Mean_PM10
                              , ["White_Population"
                              , "Black_Population", "Latino_Population"
                              , "Asian_Population", "Other_Population"])
```