Final Part 1 - Scatterplots and Regressions

March 15, 2021

0.1 Tracking Water Bill Debt by Demographics across California

0.1.1 Load Libraries and Data

First, I will load the libraries

[1]:
```
import pandas as pd
import plotly.express as px
import statsmodels.api as sm
import numpy as np
```

Now, I am going to upload the water bill data

[2]:
```
acs = pd.read_csv('../Data/Updated Bill Data 2_22.csv')
```

0.1.2 Explore the Data

Now that my data are uploaded, I need to get a sense of how they look.

[3]:
```
acs.shape
```

[3]:
```
(1073, 52)
```

This dataset has 1073 entries and 52 columns - it's big!

I also want to get a sense of the data itsef: missing data, data type, etc.

[4]:
```
acs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1073 entries, 0 to 1072
Data columns (total 52 columns):
 #   Column                                                  Non-Null Count
Dtype
---  ------                                                  --------------
-----
 0   Zip Codes                                               1073 non-null
object
 1   Count of Zip Code                                       1072 non-null
float64
 2   Sum of Less than $100                                   934 non-null
```

```
float64
 3   Sum of $100-$200                                          965 non-null
float64
 4   Sum of $200-$300                                          965 non-null
float64
 5   Sum of $300-$400                                          954 non-null
float64
 6   Sum of $400-$500                                          941 non-null
float64
 7   Sum of $500-$600                                          917 non-null
float64
 8   Sum of $600-$700                                          914 non-null
float64
 9   Sum of $700-$800                                          906 non-null
float64
 10  Sum of $800-$900                                          895 non-null
float64
 11  Sum of $900-$1000                                         894 non-null
float64
 12  Sum of More than $1000                                    936 non-null
float64
 13  Sum of Total number of delinquent residential accounts    1063 non-null
float64
 14  pop                                                       1030 non-null
float64
 15  nhw                                                       1030 non-null
float64
 16  black                                                     1030 non-null
float64
 17  hisp                                                      1030 non-null
float64
 18  asian                                                     1030 non-null
float64
 19  noncitizen                                                1030 non-null
float64
 20  immigrants                                                1030 non-null
float64
 21  ohu                                                       1030 non-null
float64
 22  lep_hh                                                    1030 non-null
float64
 23  dpov                                                      1030 non-null
float64
 24  npov                                                      1030 non-null
float64
 25  mhhi                                                      1030 non-null
float64
 26  overcrowded                                               1030 non-null
```

```
float64
 27  no_veh_hh                                                 1030 non-null
float64
 28  w_broadband                                               1030 non-null
float64
 29  pop_19_64                                                 1030 non-null
float64
 30  uninsured_19_64                                           1030 non-null
float64
 31  pct_nhw                                                   1030 non-null
float64
 32  pct_black                                                 1030 non-null
float64
 33  pct_hisp                                                  1030 non-null
float64
 34  pct_asian                                                 1030 non-null
float64
 35  pct_noncitizen                                            1030 non-null
float64
 36  pct_immigrants                                            1030 non-null
float64
 37  pct_lep_hh                                                1029 non-null
float64
 38  pct_povt                                                  1029 non-null
float64
 39  pct_overcrowded                                           1029 non-null
float64
 40  pct_no_veh_hh                                             1029 non-null
float64
 41  pct_broadband                                             1029 non-null
float64
 42  pct_no_broadband                                          1029 non-null
float64
 43  pct_uninsured_19_64                                       1030 non-null
float64
 44  aggveh                                                    1030 non-null
float64
 45  pct_no_hins                                               1030 non-null
float64
 46  veh_person                                                1030 non-null
float64
 47  Total Population in Occupied Housing Units: Renter Occupied  1030 non-null
float64
 48  Owner Occupied Pop                                        1030 non-null
float64
 49  % Renter Pop                                              1030 non-null
float64
 50  % Owner Pop                                               1030 non-null
```

```
float64
 51  Households                                              1031 non-null
float64
dtypes: float64(51), object(1)
memory usage: 436.0+ KB
```

So here we can see that there are fewer datapoints here than total zip codes in California. There are over 1,700 zips in the state. This can largely be attributed to the fact that the water bill debt data was conducted via a survey distribted by the California State Water Resources Control Board. Survey responses have their limitations in that they are completed on a voluntary basis. It is worth keeping in mind as I continue with my analysis that this is not a complete dataset of all zip codes in the state. Hopefullly, this dataset is complete enough though to draw some conclusions about water bill debt trends and demographics.

Note that all the data types are floats as well - so I shouldn't have problems conducting quantative analyses.

First thing before I get started is I need to check if there are NaN values.

[5]:
```
acs.tail()
```

[5]:

	Zip Codes	Count of Zip Code	Sum of Less than $100	\
1068	90033-2053	1.0	0.00	
1069	92780, 92705	1.0	277.00	
1070	95608 & 95628	1.0	227.00	
1071	(blank)	NaN	68.00	
1072	Grand Total	1475.0	336500.16	

	Sum of $100-$200	Sum of $200-$300	Sum of $300-$400	Sum of $400-$500	\
1068	1.00	0.00	0.00	0.00	
1069	290.00	166.00	50.00	19.00	
1070	285.00	69.00	24.00	5.00	
1071	103.00	66.00	46.00	16.00	
1072	306712.68	157566.72	101010.96	69191.64	

	Sum of $500-$600	Sum of $600-$700	Sum of $700-$800	…	\
1068	0.0	0.00	0.00	…	
1069	10.0	8.00	6.00	…	
1070	3.0	1.00	1.00	…	
1071	11.0	9.00	1.00	…	
1072	48782.2	39242.76	31735.32	…	

	pct_no_broadband	pct_uninsured_19_64	aggveh	pct_no_hins	veh_person	\
1068	NaN	NaN	NaN	NaN	NaN	
1069	NaN	NaN	NaN	NaN	NaN	
1070	NaN	NaN	NaN	NaN	NaN	
1071	NaN	NaN	NaN	NaN	NaN	
1072	NaN	NaN	NaN	NaN	NaN	

```
      Total Population in Occupied Housing Units: Renter Occupied  \
1068                                                        NaN
1069                                                        NaN
1070                                                        NaN
1071                                                        NaN
1072                                                        NaN

      Owner Occupied Pop  % Renter Pop  % Owner Pop  Households
1068                 NaN           NaN          NaN         NaN
1069                 NaN           NaN          NaN         NaN
1070                 NaN           NaN          NaN         NaN
1071                 NaN           NaN          NaN         NaN
1072                 NaN           NaN          NaN         NaN

[5 rows x 52 columns]
```

I want to make sure these "NaN values do not interfere with my analysis and regressions, so I will remove them.

[6]:
```
acs = acs.dropna()
```

Now that the "NaN" values are removed, I can start manipulating and cleaning my data.

0.1.3 Cleaning the Data

Next, I will only keep the columns of interest, including debt-related columns, as well as racial/ethnic factors, and household and income related factors. I want to see if any trends exist in the data.

[7]:
```
refined_columns = ['Zip Codes',
 'Sum of Less than $100',
 'Sum of $100-$200',
 'Sum of $200-$300',
 'Sum of $300-$400',
 'Sum of $400-$500',
 'Sum of $500-$600',
 'Sum of $600-$700',
 'Sum of $700-$800',
 'Sum of $800-$900',
 'Sum of $900-$1000',
 'Sum of More than $1000',
 'Sum of Total number of delinquent residential accounts',
 'pop',
 'mhhi',
 'pct_nhw',
 'pct_black',
 'pct_hisp',
 'pct_asian',
```

```
 'pct_povt',
 'pct_overcrowded',
 'pct_no_veh_hh',
 'pct_broadband',
 'pct_no_broadband',
 'pct_uninsured_19_64',
 'pct_noncitizen',
 'pct_immigrants',
 'pct_lep_hh',
 'pct_no_hins',
 '% Renter Pop',
  '% Owner Pop']
```

[8]:
```
acs = acs[refined_columns]
```

Now I have just saved the new data frame and will check my work.

[9]:
```
pd.set_option('display.max_columns', None)


acs.head()
```

[9]:

	Zip Codes	Sum of Less than $100	Sum of $100-$200	Sum of $200-$300	\
0	90001	5726.0	4937.0	1582.0	
1	90002	3130.0	2152.0	1052.0	
2	90003	1829.0	1880.0	1562.0	
3	90004	2199.0	1659.0	1119.0	
4	90005	1712.0	1107.0	598.0	

	Sum of $300-$400	Sum of $400-$500	Sum of $500-$600	Sum of $600-$700	\
0	684.0	395.0	283.0	220.0	
1	708.0	493.0	385.0	343.0	
2	1169.0	941.0	782.0	673.0	
3	735.0	502.0	387.0	279.0	
4	401.0	281.0	175.0	146.0	

	Sum of $700-$800	Sum of $800-$900	Sum of $900-$1000	\
0	137.0	132.0	97.0	
1	281.0	231.0	201.0	
2	513.0	482.0	401.0	
3	223.0	206.0	164.0	
4	95.0	95.0	58.0	

	Sum of More than $1000	\
0	709.0	
1	1779.0	
2	3437.0	

```
3                  1116.0
4                   426.0

   Sum of Total number of delinquent residential accounts      pop     mhhi  \
0                                            14902.0       58975.0  38521.0
1                                            10755.0       53111.0  35410.0
2                                            13669.0       72741.0  37226.0
3                                             8589.0       61586.0  48754.0
4                                             5094.0       39479.0  35149.0

    pct_nhw  pct_black  pct_hisp  pct_asian  pct_povt  pct_overcrowded  \
0  0.007003   0.088648  0.900144   0.002187  0.287524         0.137676
1  0.004199   0.194950  0.784640   0.006063  0.328603         0.065481
2  0.005389   0.221828  0.770542   0.003533  0.306597         0.102820
3  0.175689   0.038856  0.511139   0.251437  0.180601         0.128579
4  0.076496   0.061374  0.492338   0.350794  0.280593         0.205510

   pct_no_veh_hh  pct_broadband  pct_no_broadband  pct_uninsured_19_64  \
0       0.117191       0.700832          0.299168             0.243428
1       0.145994       0.620101          0.379899             0.255720
2       0.165236       0.679045          0.320955             0.261466
3       0.171544       0.777980          0.222020             0.247983
4       0.300450       0.698029          0.301971             0.328389

   pct_noncitizen  pct_immigrants  pct_lep_hh  pct_no_hins  % Renter Pop  \
0        0.289936        0.407630    0.208180     0.168425      0.647257
1        0.254542        0.350737    0.173383     0.177007      0.621999
2        0.273505        0.375442    0.165178     0.192422      0.692044
3        0.298006        0.495437    0.257157     0.195940      0.803900
4        0.388865        0.591352    0.413514     0.256415      0.912764

   % Owner Pop
0     0.352743
1     0.378001
2     0.307956
3     0.196100
4     0.087236
```

Just the cell columns if interest have saved.

0.1.4 Normalize the Data

I need the percentage of the deliquent population each debt "bucket" represents.

I also want to know what percent of the population in each zip code has water bill debt, broadly. To do this, I need to create new columns for each of these columns as a percentage of the deliquent population and total population, respectively.

Further I want to convert all of the % values from decimals to percents so they are clearer on my

outputs, maps, etc.

[10]:
```
list(acs)
```

[10]:
```
['Zip Codes',
 'Sum of Less than $100',
 'Sum of $100-$200',
 'Sum of $200-$300',
 'Sum of $300-$400',
 'Sum of $400-$500',
 'Sum of $500-$600',
 'Sum of $600-$700',
 'Sum of $700-$800',
 'Sum of $800-$900',
 'Sum of $900-$1000',
 'Sum of More than $1000',
 'Sum of Total number of delinquent residential accounts',
 'pop',
 'mhhi',
 'pct_nhw',
 'pct_black',
 'pct_hisp',
 'pct_asian',
 'pct_povt',
 'pct_overcrowded',
 'pct_no_veh_hh',
 'pct_broadband',
 'pct_no_broadband',
 'pct_uninsured_19_64',
 'pct_noncitizen',
 'pct_immigrants',
 'pct_lep_hh',
 'pct_no_hins',
 '% Renter Pop',
 '% Owner Pop']
```

[11]:
```
acs['Percent Delinquent'] = acs['Sum of Total number of delinquent residential␣
↪accounts']/acs['pop']*100


pct_debt_buckets = ['Percent Less than $100', 'Percent $100-$200', 'Percent␣
↪$200-$300' ,
                    'Percent $300-$400', 'Percent $400-$500', 'Percent $500-$600',␣
↪'Percent $600-$700',
                    'Percent $700-$800', 'Percent $800-$900', 'Percent $900-$1000',␣
↪'Percent More than $1000']
```

```
debt_buckets =  ['Sum of Less than $100', 'Sum of $100-$200', 'Sum of␣
↪$200-$300' ,
                'Sum of $300-$400', 'Sum of $400-$500', 'Sum of $500-$600',␣
↪'Sum of $600-$700',
                'Sum of $700-$800', 'Sum of $800-$900', 'Sum of $900-$1000',␣
↪'Sum of More than $1000']

sum_total = 'Sum of Total number of delinquent residential accounts'

demographics = ['pct_nhw','pct_black',␣
↪'pct_hisp','pct_asian','pct_povt','pct_overcrowded','pct_no_veh_hh',
                'pct_broadband','pct_no_broadband',␣
↪'pct_uninsured_19_64','pct_noncitizen','pct_immigrants',
                'pct_lep_hh','pct_no_hins','% Renter Pop','% Owner Pop']

for pct, debt in zip(pct_debt_buckets, debt_buckets):
    acs[pct] = acs[debt] / acs[sum_total]*100

for dem in demographics:
    acs[dem] = acs[dem]*100
```

To simplify my code, I divided up much of the repetitive calculations into loops. My standardized buckets were listed together and looped, and I did the same for my demographic factors to multiply them by 100 for standardized percents in my data.

[12]:
```
pd.set_option('display.max_columns', None)

acs.head()
```

[12]:

	Zip Codes	Sum of Less than $100	Sum of $100-$200	Sum of $200-$300	\
0	90001	5726.0	4937.0	1582.0	
1	90002	3130.0	2152.0	1052.0	
2	90003	1829.0	1880.0	1562.0	
3	90004	2199.0	1659.0	1119.0	
4	90005	1712.0	1107.0	598.0	

	Sum of $300-$400	Sum of $400-$500	Sum of $500-$600	Sum of $600-$700	\
0	684.0	395.0	283.0	220.0	
1	708.0	493.0	385.0	343.0	
2	1169.0	941.0	782.0	673.0	
3	735.0	502.0	387.0	279.0	
4	401.0	281.0	175.0	146.0	

	Sum of $700-$800	Sum of $800-$900	Sum of $900-$1000	\
0	137.0	132.0	97.0	
1	281.0	231.0	201.0	
2	513.0	482.0	401.0	

```
3            223.0            206.0             164.0
4             95.0             95.0              58.0

   Sum of More than $1000  \
0                   709.0
1                  1779.0
2                  3437.0
3                  1116.0
4                   426.0

   Sum of Total number of delinquent residential accounts      pop     mhhi  \
0                                              14902.0       58975.0  38521.0
1                                              10755.0       53111.0  35410.0
2                                              13669.0       72741.0  37226.0
3                                               8589.0       61586.0  48754.0
4                                               5094.0       39479.0  35149.0

     pct_nhw  pct_black   pct_hisp  pct_asian   pct_povt  pct_overcrowded  \
0   0.700297   8.864773  90.014413   0.218737  28.752380        13.767644
1   0.419875  19.495020  78.463972   0.606277  32.860252         6.548088
2   0.538898  22.182813  77.054206   0.353308  30.659739        10.282011
3  17.568928   3.885623  51.113890  25.143701  18.060146        12.857858
4   7.649637   6.137440  49.233770  35.079409  28.059292        20.551028

   pct_no_veh_hh  pct_broadband  pct_no_broadband  pct_uninsured_19_64  \
0      11.719146      70.083243         29.916757            24.342814
1      14.599402      62.010074         37.989926            25.571985
2      16.523618      67.904478         32.095522            26.146569
3      17.154431      77.798006         22.201994            24.798293
4      30.045007      69.802944         30.197056            32.838852

   pct_noncitizen  pct_immigrants  pct_lep_hh  pct_no_hins  % Renter Pop  \
0       28.993641       40.763035   20.817951    16.842480     64.725731
1       25.454237       35.073714   17.338265    17.700665     62.199921
2       27.350463       37.544164   16.517779    19.242243     69.204438
3       29.800604       49.543727   25.715716    19.594013     80.390024
4       38.886497       59.135236   41.351417    25.641480     91.276375

   % Owner Pop  Percent Delinquent  Percent Less than $100  Percent $100-$200  \
0    35.274269           25.268334               38.424373          33.129781
1    37.800079           20.250042               29.102743          20.009298
2    30.795562           18.791328               13.380642          13.753749
3    19.609976           13.946351               25.602515          19.315403
4     8.723625           12.903062               33.608166          21.731449

   Percent $200-$300  Percent $300-$400  Percent $400-$500  Percent $500-$600  \
0          10.616025           4.589988           2.650651           1.899074
```

```
1          9.781497          6.582985          4.583914          3.579730
2         11.427317          8.552198          6.884191          5.720974
3         13.028292          8.557457          5.844685          4.505763
4         11.739301          7.872006          5.516294          3.435414

   Percent $600-$700  Percent $700-$800  Percent $800-$900  \
0           1.476312           0.919340           0.885787
1           3.189214           2.612738           2.147838
2           4.923550           3.753018           3.526227
3           3.248341           2.596344           2.398417
4           2.866117           1.864939           1.864939

   Percent $900-$1000  Percent More than $1000
0            0.650919                 4.757751
1            1.868898                16.541144
2            2.933645                25.144488
3            1.909419                12.993364
4            1.138594                 8.362780
```

Above you'll see a quick check of my data. Next, I only wanted to keep columns of interest.

[13]:
```
columns_drop = [
 'Sum of $100-$200',
 'Sum of $200-$300',
 'Sum of $300-$400',
 'Sum of $400-$500',
 'Sum of $600-$700',
 'Sum of $700-$800',
 'Sum of $800-$900',
 'Sum of $900-$1000',]

acs = acs.drop(columns_drop, axis = 1)
```

The code I ran was a little shorter than listing the columns in their entirety. Then I checked my work.

[14]:
```
acs.head()
```

[14]:
```
   Zip Codes  Sum of Less than $100  Sum of $500-$600  Sum of More than $1000  \
0      90001                 5726.0             283.0                   709.0
1      90002                 3130.0             385.0                  1779.0
2      90003                 1829.0             782.0                  3437.0
3      90004                 2199.0             387.0                  1116.0
4      90005                 1712.0             175.0                   426.0

   Sum of Total number of delinquent residential accounts      pop     mhhi  \
0                                            14902.0       58975.0  38521.0
1                                            10755.0       53111.0  35410.0
```

```
2                                          13669.0       72741.0  37226.0
3                                           8589.0       61586.0  48754.0
4                                           5094.0       39479.0  35149.0

     pct_nhw  pct_black   pct_hisp  pct_asian   pct_povt  pct_overcrowded  \
0   0.700297   8.864773  90.014413   0.218737  28.752380        13.767644
1   0.419875  19.495020  78.463972   0.606277  32.860252         6.548088
2   0.538898  22.182813  77.054206   0.353308  30.659739        10.282011
3  17.568928   3.885623  51.113890  25.143701  18.060146        12.857858
4   7.649637   6.137440  49.233770  35.079409  28.059292        20.551028

   pct_no_veh_hh  pct_broadband  pct_no_broadband  pct_uninsured_19_64  \
0      11.719146      70.083243         29.916757            24.342814
1      14.599402      62.010074         37.989926            25.571985
2      16.523618      67.904478         32.095522            26.146569
3      17.154431      77.798006         22.201994            24.798293
4      30.045007      69.802944         30.197056            32.838852

   pct_noncitizen  pct_immigrants  pct_lep_hh  pct_no_hins  % Renter Pop  \
0       28.993641       40.763035   20.817951    16.842480     64.725731
1       25.454237       35.073714   17.338265    17.700665     62.199921
2       27.350463       37.544164   16.517779    19.242243     69.204438
3       29.800604       49.543727   25.715716    19.594013     80.390024
4       38.886497       59.135236   41.351417    25.641480     91.276375

   % Owner Pop  Percent Delinquent  Percent Less than $100  Percent $100-$200  \
0    35.274269           25.268334               38.424373          33.129781
1    37.800079           20.250042               29.102743          20.009298
2    30.795562           18.791328               13.380642          13.753749
3    19.609976           13.946351               25.602515          19.315403
4     8.723625           12.903062               33.608166          21.731449

   Percent $200-$300  Percent $300-$400  Percent $400-$500  Percent $500-$600  \
0          10.616025           4.589988           2.650651           1.899074
1           9.781497           6.582985           4.583914           3.579730
2          11.427317           8.552198           6.884191           5.720974
3          13.028292           8.557457           5.844685           4.505763
4          11.739301           7.872006           5.516294           3.435414

   Percent $600-$700  Percent $700-$800  Percent $800-$900  \
0           1.476312           0.919340           0.885787
1           3.189214           2.612738           2.147838
2           4.923550           3.753018           3.526227
3           3.248341           2.596344           2.398417
4           2.866117           1.864939           1.864939

   Percent $900-$1000  Percent More than $1000
```

```
0            0.650919                4.757751
1            1.868898               16.541144
2            2.933645               25.144488
3            1.909419               12.993364
4            1.138594                8.362780
```

[15]:
```
acs.tail()
```

[15]:
```
     Zip Codes  Sum of Less than $100  Sum of $500-$600  \
1055     96097                   33.0               0.0
1056     96101                   18.0              10.0
1062     96143                   14.0               6.0
1064     96150                  141.0             187.0
1065     96161                  372.0              21.0

      Sum of More than $1000  \
1055                     0.0
1056                     1.0
1062                     4.0
1064                   279.0
1065                     5.0

      Sum of Total number of delinquent residential accounts      pop  \
1055                                              118.0        9647.0
1056                                              125.0        5500.0
1062                                               84.0        3391.0
1064                                             2339.0       29357.0
1065                                              710.0       18333.0

         mhhi    pct_nhw  pct_black   pct_hisp  pct_asian   pct_povt  \
1055  42332.0  75.132165   1.326837  11.060433   1.461594  21.168572
1056  42097.0  82.090909   2.545455  11.709091   0.363636  19.028723
1062  47946.0  72.780891   0.000000  25.272781   0.000000  11.058685
1064  56321.0  68.811527   1.055966  23.013251   4.605375  11.916492
1065  93971.0  79.474172   0.556374  17.078492   1.581847   7.224833

      pct_overcrowded  pct_no_veh_hh  pct_broadband  pct_no_broadband  \
1055         1.943095       9.160305      73.814481         26.185519
1056         0.564236       5.512153      67.708333         32.291667
1062         0.000000       7.379310      80.068966         19.931034
1064         0.624133       7.229542      82.862344         17.137656
1065         1.414346       1.948333      93.808630          6.191370

      pct_uninsured_19_64  pct_noncitizen  pct_immigrants  pct_lep_hh  \
1055             8.188182        1.119519        3.265264    0.370113
1056            11.286863        2.909091        3.290909    2.647569
1062            19.564340       11.176644       14.921852    9.931034
```

```
1064           11.990227        8.560139       15.229758    4.542302
1065           12.619445        7.980145       10.876561    2.063790

      pct_no_hins  % Renter Pop  % Owner Pop  Percent Delinquent  \
1055     6.471383     36.467296    63.532704            1.223178
1056    13.249389     24.181818    75.818182            2.272727
1062    15.010321     47.242701    52.757299            2.477145
1064     9.379629     47.467384    52.532616            7.967435
1065    10.063819     22.625866    77.374134            3.872798

      Percent Less than $100  Percent $100-$200  Percent $200-$300  \
1055               27.966102          43.220339          15.254237
1056               14.400000          28.800000          24.000000
1062               16.666667          23.809524          10.714286
1064                6.028217          12.099188          25.651988
1065               52.394366          20.422535           9.577465

      Percent $300-$400  Percent $400-$500  Percent $500-$600  \
1055           7.627119           1.694915           0.000000
1056           6.400000           8.000000           8.000000
1062          10.714286          15.476190           7.142857
1064          22.146216           4.189825           7.994870
1065           4.929577           4.084507           2.957746

      Percent $600-$700  Percent $700-$800  Percent $800-$900  \
1055           1.694915           1.694915           0.847458
1056           4.800000           4.000000           0.000000
1062           4.761905           4.761905           0.000000
1064           3.548525           2.180419           1.710133
1065           2.394366           2.253521           0.000000

      Percent $900-$1000  Percent More than $1000
1055            0.000000                 0.000000
1056            0.800000                 0.800000
1062            1.190476                 4.761905
1064            2.522445                11.928174
1065            0.281690                 0.704225
```

I want to makes sure that after I've done these manupulations, that no more "NaN" values have cropped up, so I will remove them again, just in case.

[16]:
```
acs = acs.dropna()
```

I also want to make sure that no outliers reamin in my data. Based on a quick examination, I know that some values under "Percent Delinquent" and "Percent $400-$500 are above 100%. I will remove those.

[17]:
```
acs.drop(acs[acs['Percent Delinquent'] > 100].index, inplace = True)
acs.drop(acs[acs['Percent $400-$500'] > 100].index, inplace = True)
```

0.1.5 Summary Statistics

Now that my data is cleaned and normalized, I want to get a better sense of how it looks. I'll calculate some summary stats.

[18]:
```
Columns = ['pop','mhhi',
 'pct_nhw',
 'pct_black',
 'pct_hisp',
 'pct_asian',
 'pct_povt',
 'pct_overcrowded',
 'pct_no_veh_hh',
 'pct_broadband',
 'pct_no_broadband',
 'pct_uninsured_19_64',
 'pct_noncitizen',
 'pct_immigrants',
 'pct_lep_hh',
 'pct_no_hins',
 '% Renter Pop',
 '% Owner Pop',
 'Percent Delinquent',
 'Percent Less than $100',
 'Percent $100-$200',
 'Percent $200-$300',
 'Percent $300-$400',
 'Percent $400-$500',
 'Percent $500-$600',
 'Percent $600-$700',
 'Percent $700-$800',
 'Percent $800-$900',
 'Percent $900-$1000',
 'Percent More than $1000',
 'Sum of Less than $100',
 'Sum of $500-$600',
 'Sum of More than $1000',
 'Sum of Total number of delinquent residential accounts']


acs[Columns].describe()
```

[18]:

	pop	mhhi	pct_nhw	pct_black	pct_hisp	\
count	814.000000	814.000000	814.000000	814.000000	814.000000	

mean	34639.326781	79473.549140	43.067328	5.721547	33.827124
std	21618.452248	35941.207614	24.737833	8.119158	23.634024
min	56.000000	0.000000	0.419875	0.000000	0.000000
25%	19097.250000	53144.500000	21.586625	1.350253	14.308287
50%	32747.500000	73715.500000	44.357903	2.945989	26.709933
75%	47342.750000	98129.750000	64.062929	6.736171	50.258905
max	109414.000000	250001.000000	100.000000	80.307852	99.329502

	pct_asian	pct_povt	pct_overcrowded	pct_no_veh_hh	pct_broadband	\
count	814.000000	814.000000	814.000000	814.000000	814.000000	
mean	13.741078	14.178987	2.850151	7.495615	83.770609	
std	14.003297	9.117166	3.427620	7.957927	9.971648	
min	0.000000	0.000000	0.000000	0.000000	22.365989	
25%	3.971298	7.386855	0.760171	3.293007	79.384406	
50%	8.850978	11.719024	1.766863	5.335666	86.066575	
75%	17.924562	19.268553	3.661918	8.475880	91.031074	
max	71.248287	54.656265	29.792731	70.841312	100.000000	

	pct_no_broadband	pct_uninsured_19_64	pct_noncitizen	pct_immigrants	\
count	814.000000	814.000000	814.000000	814.000000	
mean	16.229391	10.356382	12.260887	25.572354	
std	9.971648	6.413342	7.826706	12.510187	
min	0.000000	0.000000	0.000000	0.000000	
25%	8.968926	5.573501	6.328193	16.020169	
50%	13.933425	8.994690	10.493989	23.628526	
75%	20.615594	13.809660	16.733234	34.184872	
max	77.634011	42.296651	51.867816	64.147483	

	pct_lep_hh	pct_no_hins	% Renter Pop	% Owner Pop	Percent Delinquent	\
count	814.000000	814.000000	814.000000	814.000000	814.000000	
mean	9.013155	8.012078	42.807498	57.192502	4.412848	
std	8.168846	4.789368	18.083785	18.083785	6.229630	
min	0.000000	0.000000	0.000000	0.355433	0.001400	
25%	3.462600	4.452807	29.646169	45.413031	0.755817	
50%	6.619639	6.993525	40.670793	59.329207	2.081049	
75%	12.249508	10.639786	54.586969	70.353831	5.302158	
max	58.410351	32.174823	99.644567	100.000000	82.509182	

	Percent Less than $100	Percent $100-$200	Percent $200-$300	\
count	814.000000	814.000000	814.000000	
mean	24.688410	22.877392	13.206015	
std	18.885616	13.689951	8.182374	
min	0.000000	0.000000	0.000000	
25%	11.702278	15.153655	9.469726	
50%	22.246260	21.038445	12.203970	
75%	34.534090	29.541641	15.620484	
max	100.000000	100.000000	100.000000	

	Percent $300-$400	Percent $400-$500	Percent $500-$600	\
count	814.000000	814.000000	814.000000	
mean	8.832737	6.175884	4.359706	
std	6.663979	5.847302	3.895716	
min	0.000000	0.000000	0.000000	
25%	5.409841	3.009828	2.110479	
50%	7.858945	5.464828	3.902276	
75%	10.131585	7.162102	5.394924	
max	100.000000	71.929825	50.000000	

	Percent $600-$700	Percent $700-$800	Percent $800-$900	\
count	814.000000	814.000000	814.000000	
mean	3.331521	2.413291	1.899418	
std	3.217239	2.422826	2.217892	
min	0.000000	0.000000	0.000000	
25%	1.431659	0.786854	0.507186	
50%	2.991916	2.073094	1.633098	
75%	4.251593	3.291232	2.742685	
max	33.333333	33.333333	40.000000	

	Percent $900-$1000	Percent More than $1000	Sum of Less than $100	\
count	814.000000	814.000000	814.000000	
mean	1.472669	9.923643	392.614447	
std	1.740038	12.943903	744.953156	
min	0.000000	0.000000	0.000000	
25%	0.264734	1.541620	16.000000	
50%	1.148687	5.252938	114.500000	
75%	2.170264	15.461090	460.750000	
max	22.222222	100.000000	7660.000000	

	Sum of $500-$600	Sum of More than $1000	\
count	814.000000	814.000000	
mean	57.334398	180.045209	
std	102.942018	446.905172	
min	0.000000	0.000000	
25%	5.000000	5.000000	
50%	17.000000	22.000000	
75%	55.750000	86.750000	
max	782.000000	3502.000000	

	Sum of Total number of delinquent residential accounts
count	814.000000
mean	1525.185504
std	2525.183650
min	1.000000
25%	147.250000

```
50%                                521.000000
75%                               1676.250000
max                              24261.000000
```

This is a lot of information in one place, but it can help me benchmark averages for debt and dempographics across the state. Thus, in my further analysis of mapping demographis, I have this table as a baseline to see which Zip Codes are above, below the mean and median, and which Zips are towards the lower and upper bounds of the distribution. These stats will help guide my mapping as I determine Zip codes that have demographic factors that might put them at a higher risk of increased deliquent bill debt. Below I plot some of these relationships to better visualize them.

0.1.6 Regresssions

Next, I want to get a better sense of the relationships in my data. I am going to run some robust linear regresstions and determine which demographic factors are associated with % Delinquent Accounts and which are associated with high levels of debt (Over %1,000)

First, I regressed all the demographic factors on "Percent Delinquent"

[19]:
```
X = acs[["pop","mhhi",␣
↪"pct_nhw","pct_black","pct_hisp","pct_asian","pct_povt","pct_overcrowded","pct_no_veh_hh",
"pct_broadband",␣
↪"pct_no_broadband","pct_uninsured_19_64","pct_noncitizen","pct_immigrants","pct_lep_hh",
"pct_no_hins","% Renter Pop", "% Owner Pop", 'Percent Less than $100',␣
↪'Percent $500-$600', 'Percent More than $1000']]
y = acs["Percent Delinquent"]

X = sm.add_constant(X)

model = sm.RLM(y, X).fit()
predictions = model.predict(X)


model.summary()
```

[19]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                    Robust linear Model Regression Results
==============================================================================
Dep. Variable:     Percent Delinquent   No. Observations:                  814
Model:                            RLM   Df Residuals:                      794
Method:                          IRLS   Df Model:                           19
Norm:                          HuberT
Scale Est.:                       mad
Cov Type:                          H1
Date:                Mon, 15 Mar 2021
Time:                        18:48:40
```

```
No. Iterations:                    50
==============================================================================
===========
                               coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
-----------
const                   -2.243e+06   1.41e+06     -1.591      0.112      -5e+06
5.19e+05
pop                     -6.592e-06   7.01e-06     -0.940      0.347   -2.03e-05
7.15e-06
mhhi                    -1.018e-05   6.52e-06     -1.562      0.118    -2.3e-05
2.6e-06
pct_nhw                     0.0925      0.063      1.469      0.142      -0.031
0.216
pct_black                   0.1326      0.066      2.005      0.045       0.003
0.262
pct_hisp                    0.0420      0.061      0.692      0.489      -0.077
0.161
pct_asian                   0.0159      0.065      0.243      0.808      -0.112
0.144
pct_povt                    0.0230      0.026      0.874      0.382      -0.029
0.074
pct_overcrowded             0.2884      0.066      4.381      0.000       0.159
0.417
pct_no_veh_hh              -0.0516      0.025     -2.032      0.042      -0.101
-0.002
pct_broadband            1.27e+08   8.08e+07      1.573      0.116   -3.13e+07
2.85e+08
pct_no_broadband         1.27e+08   8.08e+07      1.573      0.116   -3.13e+07
2.85e+08
pct_uninsured_19_64         0.1721      0.148      1.167      0.243      -0.117
0.461
pct_noncitizen             -0.2039      0.046     -4.412      0.000      -0.294
-0.113
pct_immigrants              0.1598      0.032      5.013      0.000       0.097
0.222
pct_lep_hh                 -0.0293      0.036     -0.816      0.414      -0.100
0.041
pct_no_hins                -0.0154      0.177     -0.087      0.931      -0.363
0.332
% Renter Pop            -1.27e+08   8.07e+07     -1.573      0.116   -2.85e+08
3.13e+07
% Owner Pop             -1.27e+08   8.07e+07     -1.573      0.116   -2.85e+08
3.13e+07
Percent Less than $100      0.0142      0.008      1.741      0.082      -0.002
0.030
```

```
Percent $500-$600          -0.1430      0.037     -3.857      0.000      -0.216
-0.070
Percent More than $1000     0.0634      0.011      5.647      0.000       0.041
0.085
==============================================================================
==========

If the model instance has been used for another fit with different fit
parameters, then the fit options might not be the correct ones anymore .
"""
```

Based on the output, we can see there are 5 Demographic characteristics that have a small, albeit statistically significant relationship with Percent Delinquent: * Percent Black (.1326) * Percent Overcrowded (.2884) * Percent No Vehicle Per Household (-.0516) * Percent Noncitizen (-.2039) * Percent Immigrants (.1598)

For No Vehicle Per Household and Percent Noncitizen, it is a bit strange that their coefficients are negative, and a bit counterintuitive that as the percent of delinquent increases per zip code, that that these factors decrease. I immagine there is some outside variable that is not included in this model that is having some impact on these values.

Maybe households without cars just live in more walkable/bikeable areas and lack of vehicle is not a very good proxy measurement for financial instability. Or, more broadly, perhaps vehicle ownership is not a good measure of debt, as there are many factors that contribute to any level of water bill debt which can span location, income, household type, etc. In the case of percent noncitizen, perhaps it is much harder to track or get data on citizen-status, as these folks might be less likely to trust or give information to government/related entities, this could skew the data.

For the sake of my analysis, I will use, in my risk indicators for percent delinquent: Percent Black, Percent Overcrowed, and Percent Immigrants, as I am not sure if the Vehicle and Citizen-status variables have some other confounding variable impacting their coefficient.

Next, I want to see which demographic factors are associated with extremely high levels of debt.

[20]:
```
X = acs[["pop","mhhi",␣
 ↪"pct_nhw","pct_black","pct_hisp","pct_asian","pct_povt","pct_overcrowded","pct_no_veh_hh",
 "pct_broadband",␣
 ↪"pct_no_broadband","pct_uninsured_19_64","pct_noncitizen","pct_immigrants","pct_lep_hh",
 "pct_no_hins","% Renter Pop", "% Owner Pop",'Percent Less than $100', 'Percent␣
 ↪$500-$600', 'Percent Delinquent']]
y = acs['Percent More than $1000']

X = sm.add_constant(X)
model = sm.RLM(y, X).fit()
predictions = model.predict(X)


model.summary()
```

[20]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                        Robust linear Model Regression Results
==============================================================================
===
Dep. Variable:     Percent More than $1000   No. Observations:
814
Model:                                RLM   Df Residuals:
794
Method:                              IRLS   Df Model:
19
Norm:                             HuberT
Scale Est.:                          mad
Cov Type:                             H1
Date:                   Mon, 15 Mar 2021
Time:                           18:48:41
No. Iterations:                        50
==============================================================================
==========
                              coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
----------
const                  -1.292e+06   3.06e+06     -0.423      0.673   -7.29e+06
4.7e+06
pop                     9.895e-06   1.52e-05      0.651      0.515   -1.99e-05
3.97e-05
mhhi                    4.51e-05    1.4e-05       3.224      0.001    1.77e-05
7.25e-05
pct_nhw                   0.2899      0.136       2.138      0.033       0.024
0.556
pct_black                 0.2181      0.143       1.527      0.127      -0.062
0.498
pct_hisp                  0.1924      0.131       1.470      0.142      -0.064
0.449
pct_asian                 0.0081      0.142       0.057      0.954      -0.269
0.286
pct_povt                  0.1574      0.056       2.814      0.005       0.048
0.267
pct_overcrowded          -0.1169      0.143      -0.818      0.414      -0.397
0.163
pct_no_veh_hh             0.2445      0.055       4.461      0.000       0.137
0.352
pct_broadband          7.417e+07   1.75e+08       0.424      0.672   -2.69e+08
4.17e+08
pct_no_broadband       7.417e+07   1.75e+08       0.424      0.672   -2.69e+08
4.17e+08
```

```
pct_uninsured_19_64         0.0786      0.319      0.246      0.805      -0.547
0.704
pct_noncitizen             -0.2361      0.101     -2.346      0.019      -0.433
-0.039
pct_immigrants              0.5437      0.068      8.041      0.000       0.411
0.676
pct_lep_hh                 -0.3181      0.077     -4.132      0.000      -0.469
-0.167
pct_no_hins                -0.1030      0.384     -0.268      0.789      -0.855
0.649
% Renter Pop           -7.415e+07   1.75e+08     -0.424      0.672   -4.17e+08
2.69e+08
% Owner Pop            -7.415e+07   1.75e+08     -0.424      0.672   -4.17e+08
2.69e+08
Percent Less than $100     -0.1582      0.016     -9.682      0.000      -0.190
-0.126
Percent $500-$600           0.2504      0.081      3.087      0.002       0.091
0.409
Percent Delinquent          0.2443      0.046      5.291      0.000       0.154
0.335
==============================================================================
==========

If the model instance has been used for another fit with different fit
parameters, then the fit options might not be the correct ones anymore .
"""
```

Above, you can see that seven demographic factors had a statiscally significant relationship with % More than $1,000 in water bill debt: * Median Household Income (4.51) * Percent Nonhispanic White (0.2899) * Percent Poverty (.1574) * Percent No Vehicle Per Household (0.2445) * Percent Noncitizen (-0.2361) * Percent Immigrants (0.5437) * Percent Limited English Proficiency Per Household (-0.3181)

Median household income has the strongest relationship of any demographic characteristics in either of my models. This is especially confusing, given that the relationship is positive, and you would expect household income to decrease as high debt increases. I'll take a look at the scatterplot to get a better sense of what's happening.

For percent nonhispanic white, I feel like this is a poor indicator of water bill debt, and is probably combined with other factors not included in this model - such as rual households, for example.

Percent poverty also has a small, but statistically significant impact on high levels of debt, which aligns with the logic I discussed under Median household income.

No vehicle per household cropped up again in this model, but instead has a positive relationship, which indicates a stronger relationship between high levels of debt and limited vehicle access.

Percent noncitizen again has a negative relationship, which again leads me to believe there is a confounding factor or this information might be unrelaiable due to its difficulty in obtaining.

Percent Immigrants has a small but statistically significant relationship, which suggests that immigration status might have a small impact on high degrees of bill debt, which might be due to a host of factors related to immigrant marginalization and distrimination.

Percent Limited English proficiency also has a negative coeficient, and this is probably also due to the unreliablilty of data and difficulty in collecting this information due to language barriers. While I think this probably is a strong indicator, more research would have to be done to include it in this model.

The risk indicators I will include for high levels of debt are: Median household income, percent poverty, percent no vehicle per household, and percent immigrants.

Next are some scatterplots to help with the visualization of each statistically-significant variable, but first, I want to better understand why the output for Median Household Income is so strange.

[21]:
```
acs[acs['Percent More than $1000'] == 100]
```

[21]:

	Zip Codes	Sum of Less than $100	Sum of $500-$600	\
349	92243	0.0	0.0	
459	92617	0.0	0.0	
475	92672	0.0	0.0	
714	94158	0.0	0.0	
984	95652	0.0	0.0	

	Sum of More than $1000	\
349	1.0	
459	1.0	
475	2.0	
714	1.0	
984	1.0	

	Sum of Total number of delinquent residential accounts	pop	\
349	1.0	50484.0	
459	1.0	17086.0	
475	2.0	34110.0	
714	1.0	7291.0	
984	1.0	667.0	

	mhhi	pct_nhw	pct_black	pct_hisp	pct_asian	pct_povt	\
349	46648.0	10.666746	2.107598	84.502020	2.331432	24.510826	
459	39135.0	35.309610	1.984081	24.007960	34.226852	45.026226	
475	81347.0	72.773380	0.384052	20.319554	2.905306	7.807095	
714	153077.0	39.665341	5.102181	15.333973	39.349883	13.248337	
984	51250.0	47.526237	18.140930	18.440780	9.145427	46.176912	

	pct_overcrowded	pct_no_veh_hh	pct_broadband	pct_no_broadband	\
349	3.893395	7.338741	75.635380	24.364620	
459	1.428213	16.988224	59.483839	40.516161	
475	2.199578	3.373155	88.025299	11.974701	

```
714         3.856243      33.709339      81.820567         18.179433
984         0.000000       6.936416      89.595376         10.404624

     pct_uninsured_19_64  pct_noncitizen  pct_immigrants  pct_lep_hh  \
349            11.753320       15.852547       31.017748   17.497103
459             3.831386       20.385111       28.005385    6.940616
475             8.192338        6.508355       12.937555    3.232607
714             3.992619       19.736662       39.857358    8.210913
984             6.410256        5.697151       12.593703    4.046243

     pct_no_hins  % Renter Pop  % Owner Pop  Percent Delinquent  \
349     9.493283     41.985580    58.014420            0.001981
459     3.868664     48.835304    51.164696            0.005853
475     6.768410     42.961008    57.038992            0.005863
714     3.730627     80.235907    19.764093            0.013716
984     3.828484     92.803598     7.196402            0.149925

     Percent Less than $100  Percent $100-$200  Percent $200-$300  \
349                     0.0                0.0                0.0
459                     0.0                0.0                0.0
475                     0.0                0.0                0.0
714                     0.0                0.0                0.0
984                     0.0                0.0                0.0

     Percent $300-$400  Percent $400-$500  Percent $500-$600  \
349                0.0                0.0                0.0
459                0.0                0.0                0.0
475                0.0                0.0                0.0
714                0.0                0.0                0.0
984                0.0                0.0                0.0

     Percent $600-$700  Percent $700-$800  Percent $800-$900  \
349                0.0                0.0                0.0
459                0.0                0.0                0.0
475                0.0                0.0                0.0
714                0.0                0.0                0.0
984                0.0                0.0                0.0

     Percent $900-$1000  Percent More than $1000
349                 0.0                   100.0
459                 0.0                   100.0
475                 0.0                   100.0
714                 0.0                   100.0
984                 0.0                   100.0
```

It seems there are, in fact, 5 zip codes with moderate median household incomes where 100% of the debt is more than $1000. It looks like this is not a typo, becuase no other debt buckets for

these zip codes have values. Given this somewhat strange phenomenon, I think I will leave Median Household Income out of my risk indicators. After some messing around with these 5 zip codes and removing them from the data, I still noticed the same trendline, meaning there is some other factor confounding this result. Because of this strange phenomenon, and because I do not think increased median household income, broadly, is a good indicator of high bill debt, I am going to make note of this for future analysis, but leave it out of my risk indicators.

0.1.7 Scatterplots

First, I will plot the statistically-significant variables associated with Percent Delinquency

[22]:
```
fig1 = px.scatter(acs,
          x ='pct_black',
          y = 'Percent Delinquent', trendline = "ols")

fig1
```

[23]:
```
fig1.write_html('../Final/pctblack_delin.html')
```

There is a very slight trend here, but not a very obvious one. However, we know from the regressions that it is statistically significant.

[24]:
```
fig2 = px.scatter(acs,
          x ='pct_immigrants',
          y = 'Percent Delinquent' , trendline= "ols")

fig2
```

[25]:
```
fig2.write_html('../Final/pctimmigrant_delin.html')
```

The trendline here is nearly horizontal, but the coefficient in this regresision was also rather small.

[26]:
```
fig3 = px.scatter(acs,
          x ='pct_overcrowded',
          y = 'Percent Delinquent' , trendline= "ols")

fig3
```

[27]:
```
fig3.write_html('../Final/pctovercrowd_delin.html')
```

The trendline here is slightly positive - there is a clear upward trend.

Now, moving on to scatterplots that depict the relationship between demographic variables of interest and extreme levels of debt.

[28]:
```
px.scatter(acs,
          x ='mhhi',
          y = 'Percent More than $1000', trendline = "ols")
```

Median household income is the strongest predictor of extreme levels of debt, and this is also evidenced by the pretty strong positive trendline above. Which, as previously discussed, is a bit counter-intuitive. Increased levels of debt should not increase as median household income increases.

[29]:
```
fig4 = px.scatter(acs,
          x ='pct_povt',
          y = 'Percent More than $1000', trendline = "ols")

fig4
```

[30]:
```
fig4.write_html('../Final/pctpovt_highdebt.html')
```

There is a very, very slight postitive trend here, which is in line with the regression output.

[31]:
```
fig5 = px.scatter(acs,
          x ='pct_immigrants',
          y = 'Percent More than $1000', trendline = "ols")

fig5
```

[32]:
```
fig5.write_html('../Final/pctimmigrant_highdebt.html')
```

Above, we see a slight positive trend, which again, is in line with the regression output.

[33]:
```
fig6 = px.scatter(acs,
          x ='pct_no_veh_hh',
          y = 'Percent More than $1000', trendline = "ols")

fig6
```

[34]:
```
fig6.write_html('../Final/pctveh_highdebt.html')
```

There is a very strong, positive trend here, which shows that high levels of debt are more strongly associated with hosouseholds that do not have vehicles. This is in contrast to the first regression I ran, which was just general debt percentage, which did not show any strong, positive relationship with vehicle ownership.

0.1.8 Conclusion from regressions and scatterplots

My final takeaway from the regressions and scatterplots is that there is no one silver bullet that fully explains household water debt. There are many facotrs that slightly contribute, but the landscape of water debt is vast and can range from less than $100 to over $1000 - this means there are many different household and demographic factors that may contribute. Looking at higher levels of debt does offer a bit more clarity, with median household income having the strongest relationship, but again, I think my research shows that many other facts contrbiute slighly, albeit in a statistically significant way, to these higher levels of debt.

[ ]:
```

```