My Own Model

August 29, 2018

In [103]:
```
import math
import warnings

from IPython.display import display
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn import linear_model
import statsmodels.formula.api as smf

# Display preferences.
%matplotlib inline
pd.options.display.float_format = '{:.3f}'.format

# Suppress annoying harmless error.
warnings.filterwarnings(
    action="ignore",
    module="scipy",
    message="^internal gelsd"
)
```

I started out by limiting the data to towns with a population of 5000. The large cities contained some outliers that made the regression silly.

I then experimented with different values, but I tried to be careful by not using any of the values included in Property crimes as features. By looking at the log of Property Crimes instead of Property Crimes directly, and using Violent Crime, Population and Population squared as features, I got an R2 of 0.53, an a normal residual curve.

In [208]:
```
# Acquire, load, and preview the data.
data = pd.read_csv('crime_low_pop.csv')
display(data.head())

# New feature
data['LPop'] = np.log10(data['Population'] + 1)
data['LPopSq'] = np.log10(data['Population'] + 1)
data['LProp'] = np.log10(data['Property\ncrime'] + 1)
```

```
data['SRProp'] = data['Property\ncrime'] ** 0.5
data['InvProp'] = 1 / data['SRProp']
data['VCpPerson'] = data['Violent\ncrime'] / data['Population']

# X_all = data[['Population', 'Violent\ncrime', 'Population_sq', 'LPop', 'LPopSq', 'LP
# Correlation Matrix
X_all = data[['Population', 'Violent\ncrime', 'Population_sq', 'LPop', 'LPopSq', 'LPro
correlation_matrix = X_all.corr()
display(correlation_matrix)
display(data.head())

# Instantiate and fit our model.
regr = linear_model.LinearRegression()
Y = data['LProp'].values.reshape(-1, 1)
X = data[['Violent\ncrime', 'Population', 'Population_sq']]
regr.fit(X, Y)

# Inspect the results.
print('\nCoefficients: \n', regr.coef_)
print('\nIntercept: \n', regr.intercept_)
print('\nR-squared:')
print(regr.score(X, Y))
```

	Unnamed: 0	City	Population	Violent\ncrime	\
0	0	Adams Village	1861	0	
1	1	Addison Town and Village	2577	3	
2	2	Akron Village	2846	3	
3	5	Alfred Village	4089	5	
4	6	Allegany Village	1781	3	

	Murder and\nnonnegligent\nmanslaughter	Rape\n(revised\ndefinition)1	\
0	0	nan	
1	0	nan	
2	0	nan	
3	0	nan	
4	0	nan	

	Rape\n(legacy\ndefinition)2	Robbery	Aggravated\nassault	Property\ncrime	\
0	0	0	0	12	
1	0	0	3	24	
2	0	0	3	16	
3	0	3	2	46	
4	0	0	3	10	

	Burglary	Larceny-\ntheft	Motor\nvehicle\ntheft	Arson3	Population_sq	\
0	2	10	0	0.000	3463321	
1	3	20	1	0.000	6640929	
2	1	15	0	0.000	8099716	

```
3        10               36                     0   0.000       16719921
4         0               10                     0   0.000        3171961
```

```
   Murder_cat  Robbery_cat
0           0            0
1           0            0
2           0            0
3           0            1
4           0            0
```

```
                Population  Violent\ncrime  Population_sq  LPop  LPopSq  \
Population           1.000           0.490          0.978 0.962   0.962
Violent\ncrime       0.490           1.000          0.499 0.448   0.448
Population_sq        0.978           0.499          1.000 0.886   0.886
LPop                 0.962           0.448          0.886 1.000   1.000
LPopSq               0.962           0.448          0.886 1.000   1.000
LProp                0.640           0.547          0.585 0.668   0.668
SRProp               0.635           0.650          0.608 0.628   0.628
Property\ncrime      0.553           0.675          0.551 0.520   0.520
```

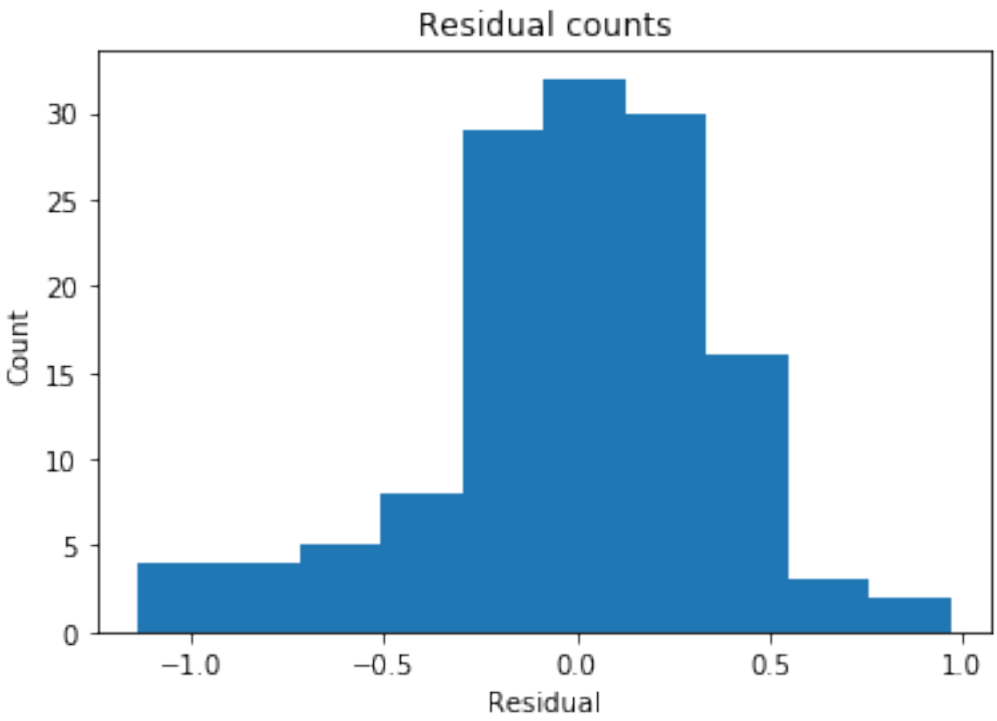
```
                LProp  SRProp  Property\ncrime
Population      0.640   0.635            0.553
Violent\ncrime  0.547   0.650            0.675
Population_sq   0.585   0.608            0.551
LPop            0.668   0.628            0.520
LPopSq          0.668   0.628            0.520
LProp           1.000   0.938            0.784
SRProp          0.938   1.000            0.948
Property\ncrime 0.784   0.948            1.000
```

```
   Unnamed: 0                     City  Population  Violent\ncrime  \
0           0             Adams Village        1861               0
1           1  Addison Town and Village        2577               3
2           2             Akron Village        2846               3
3           5            Alfred Village        4089               5
4           6          Allegany Village        1781               3
```

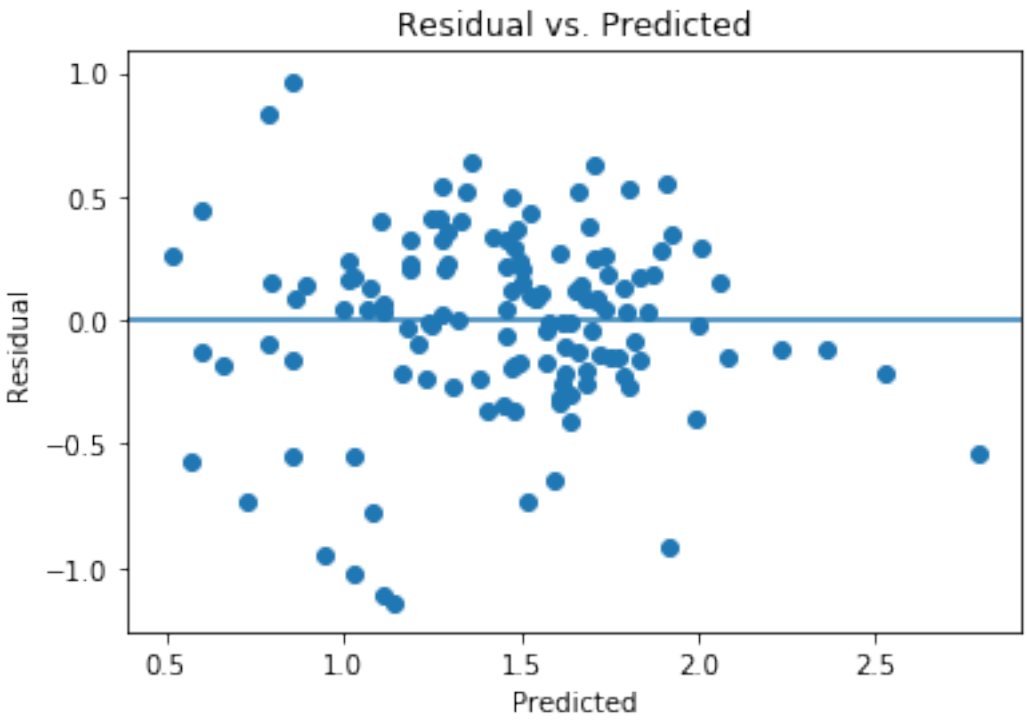
```
   Murder and\nnonnegligent\nmanslaughter  Rape\n(revised\ndefinition)1  \
0                                       0                           nan
1                                       0                           nan
2                                       0                           nan
3                                       0                           nan
4                                       0                           nan
```

```
   Rape\n(legacy\ndefinition)2  Robbery  Aggravated\nassault  Property\ncrime  \
0                            0        0                    0               12
```

```
1                          0        0                   3               24
2                          0        0                   3               16
3                          0        3                   2               46
4                          0        0                   3               10

     ...      Arson3  Population_sq  Murder_cat  Robbery_cat  LPop  LPopSq  \
0    ...       0.000        3463321           0            0 3.270   3.270
1    ...       0.000        6640929           0            0 3.411   3.411
2    ...       0.000        8099716           0            0 3.454   3.454
3    ...       0.000       16719921           0            1 3.612   3.612
4    ...       0.000        3171961           0            0 3.251   3.251

   LProp  SRProp  InvProp  VCpPerson
0  1.114   3.464    0.289      0.000
1  1.398   4.899    0.204      0.001
2  1.230   4.000    0.250      0.001
3  1.672   6.782    0.147      0.001
4  1.041   3.162    0.316      0.002

[5 rows x 23 columns]



Coefficients:
 [[ 4.39314224e-02  7.44158728e-04 -9.44920651e-08]]

Intercept:
 [0.1518167]

R-squared:
0.5302550894759066


In [209]: # data[data['City']=='Seneca Falls Town']

In [210]: # Extract predicted values.
          predicted = regr.predict(X).ravel()
          # actual = data['Property\ncrime']
          actual = data['LProp']
          # Calculate the error, also called the residual.
          residual = actual - predicted

          # This looks a bit concerning.
          plt.hist(residual)
          plt.title('Residual counts')
          plt.xlabel('Residual')
          plt.ylabel('Count')
          plt.show()
```




```
In [211]: plt.scatter(predicted, residual)
          plt.xlabel('Predicted')
          plt.ylabel('Residual')
          plt.axhline(y=0)
          plt.title('Residual vs. Predicted')
          plt.show()
```




```
In [212]: correlation_matrix = X.corr()
          display(correlation_matrix)
```

	Violent\ncrime	Population	Population_sq
Violent\ncrime	1.000	0.490	0.499
Population	0.490	1.000	0.978
Population_sq	0.499	0.978	1.000

```
In [213]: data.describe()
```

Out[213]:

	Unnamed: 0	Population	Violent\ncrime	\
count	133.000	133.000	133.000	
mean	166.203	2553.023	3.030	
std	106.400	1101.040	4.127	
min	0.000	526.000	0.000	
25%	70.000	1754.000	0.000	
50%	165.000	2412.000	2.000	
75%	260.000	3457.000	4.000	
max	344.000	4982.000	27.000	

	Murder and\nnonnegligent\nmanslaughter	Rape\n(revised\ndefinition)1	\
count	133.000	0.000	
mean	0.000	nan	

```
std                                    0.000                          nan
min                                    0.000                          nan
25%                                    0.000                          nan
50%                                    0.000                          nan
75%                                    0.000                          nan
max                                    0.000                          nan

       Rape\n(legacy\ndefinition)2  Robbery  Aggravated\nassault  \
count                      133.000  133.000              133.000
mean                         0.256    0.398                2.376
std                          0.785    0.788                3.267
min                          0.000    0.000                0.000
25%                          0.000    0.000                0.000
50%                          0.000    0.000                1.000
75%                          0.000    1.000                3.000
max                          5.000    4.000               19.000

       Property\ncrime  Burglary    ...      Arson3  Population_sq  \
count          133.000   133.000    ...     133.000        133.000
mean            49.609     8.053    ...       0.053    7721098.195
std             52.932     9.151    ...       0.224    6085018.697
min              0.000     0.000    ...       0.000     276676.000
25%             14.000     2.000    ...       0.000    3076516.000
50%             36.000     6.000    ...       0.000    5817744.000
75%             62.000    11.000    ...       0.000   11950849.000
max            292.000    49.000    ...       1.000   24820324.000

       Murder_cat  Robbery_cat    LPop  LPopSq   LProp  SRProp  InvProp  \
count     133.000      133.000 133.000 133.000 133.000 133.000  133.000
mean        0.000        0.271   3.360   3.360   1.455   6.145      inf
std         0.000        0.446   0.214   0.214   0.541   3.455      nan
min         0.000        0.000   2.722   2.722   0.000   0.000    0.059
25%         0.000        0.000   3.244   3.244   1.176   3.742    0.127
50%         0.000        0.000   3.383   3.383   1.568   6.000    0.167
75%         0.000        1.000   3.539   3.539   1.799   7.874    0.267
max         0.000        1.000   3.697   3.697   2.467  17.088      inf

       VCpPerson
count    133.000
mean       0.001
std        0.001
min        0.000
25%        0.000
50%        0.001
75%        0.002
max        0.008

[8 rows x 22 columns]
```

In [114]:
```
# Declare that you want to make a scatterplot matrix.
g = sns.PairGrid(X_all, diag_sharey=False)
# Scatterplot.
g.map_upper(plt.scatter, alpha=.5)
# Fit line summarizing the linear relationship of the two variables.
g.map_lower(sns.regplot, scatter_kws=dict(alpha=0))
# Give information about the univariate distributions of the variables.
g.map_diag(sns.kdeplot, lw=3)
plt.show()
```