In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
data = pd.read_csv('acs_ny.csv')
```

From Model 4, we found some variables we had to drop because of multicollinearity and not enough correlation between independent and dependent variables.

Acres, YearBuilt, Language, HeatingFuel, NumChildren, NumBedrooms are the variables we will drop here as well.

In [3]:
```
data = data.drop(['Acres', 'YearBuilt', 'Language', 'HeatingFuel', 'NumChildre
n', 'NumBedrooms'],axis = 1)
```

In [4]:
```
y = data.iloc[:, 0:1]
x = data.iloc[:, 1:]
```

In [5]:
```
encoded_x = pd.get_dummies(x, drop_first = True) #to get rid of dummy variable
 trap
```

In [6]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(encoded_x, y, test_size =
0.2, random_state = 0)
```

I am going to use PCA first without standarizing the variables.

Then, I am going to standardize all the variables.

In [7]:
```
from sklearn.decomposition import PCA
pca = PCA(n_components = None)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
```

In [8]:
```
explained_variance
```

Out[8]:
```
array([  7.53279076e-01,   2.42492061e-01,   4.22498903e-03,
         2.20628111e-06,   8.18926315e-07,   4.33826270e-07,
         1.89213459e-07,   7.81120429e-08,   6.83415401e-08,
         3.41041033e-08,   2.25054130e-08,   1.43686654e-08,
         6.07455041e-09,   2.83155563e-09])
```

The first two principle components would explain almost 99.5% of the variance. Awesome. Let us use n_components = 2 next.

In [9]:
```
pca = PCA(n_components = 2)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
```

In [10]:
```
explained_variance.sum()
```

Out[10]:
```
0.99577113638526049
```

In [11]:
```
import statsmodels.api as sm
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\statsmodels\compat\pandas.py:56: FutureWarning: The pandas.core.datet
ools module is deprecated and will be removed in a future version. Please use
the pandas.tseries module instead.
  from pandas.core import datetools
```

In [13]:
```
X = X_train_pca
Y = Y_train
## fit a OLS model with intercept on TV and Radio
X = sm.add_constant(X)
est = sm.OLS(Y, X).fit()

est.summary()
```

Out[13]:

OLS Regression Results

Dep. Variable:	FamilyIncome	**R-squared:**	0.258
Model:	OLS	**Adj. R-squared:**	0.258
Method:	Least Squares	**F-statistic:**	3163.
Date:	Sat, 14 Apr 2018	**Prob (F-statistic):**	0.00
Time:	13:10:38	**Log-Likelihood:**	-2.3282e+05
No. Observations:	18196	**AIC:**	4.657e+05
Df Residuals:	18193	**BIC:**	4.657e+05
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	1.109e+05	646.796	171.454	0.000	1.1e+05	1.12e+05
x1	38.8084	0.488	79.530	0.000	37.852	39.765
x2	0.4086	0.860	0.475	0.635	-1.277	2.094

Omnibus:	12122.225	**Durbin-Watson:**	2.005
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	270385.674
Skew:	2.881	**Prob(JB):**	0.00
Kurtosis:	20.984	**Cond. No.**	1.33e+03

Not really any improvement. We will standardize all variables now.

In [14]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train_std = sc.fit_transform(X_train)
X_test_std = sc.transform(X_test)
```

In [16]:
```
from sklearn.decomposition import PCA
pca = PCA(n_components = None)
X_train_pca = pca.fit_transform(X_train_std)
X_test_pca = pca.transform(X_test_std)
explained_variance = pca.explained_variance_ratio_
```

In [17]:
```
explained_variance
```

Out[17]:
```
array([ 0.20053618,  0.13138799,  0.11152171,  0.09922881,  0.07979838,
        0.06797892,  0.05669482,  0.05135995,  0.04793473,  0.04269118,
        0.03816415,  0.03258892,  0.03063314,  0.00948112])
```

We will use n_components = 8.

In [18]:
```
pca = PCA(n_components = 8)
X_train_pca = pca.fit_transform(X_train_std)
X_test_pca = pca.transform(X_test_std)
explained_variance = pca.explained_variance_ratio_
```

In [19]:
```
explained_variance.sum()
```

Out[19]:
```
0.79850675137283367
```

In [20]:
```
import statsmodels.api as sm
```

In [21]:
```
X = X_train_pca
Y = Y_train
## fit a OLS model with intercept on TV and Radio
X = sm.add_constant(X)
est = sm.OLS(Y, X).fit()

est.summary()
```

Out[21]:

OLS Regression Results

Dep. Variable:	FamilyIncome	**R-squared:**	0.292
Model:	OLS	**Adj. R-squared:**	0.292
Method:	Least Squares	**F-statistic:**	937.6
Date:	Sat, 14 Apr 2018	**Prob (F-statistic):**	0.00
Time:	13:15:06	**Log-Likelihood:**	-2.3240e+05
No. Observations:	18196	**AIC:**	4.648e+05
Df Residuals:	18187	**BIC:**	4.649e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	1.109e+05	631.903	175.495	0.000	1.1e+05	1.12e+05
x1	-2.723e+04	377.129	-72.214	0.000	-2.8e+04	-2.65e+04
x2	2.056e+04	465.917	44.129	0.000	1.96e+04	2.15e+04
x3	3082.4431	505.716	6.095	0.000	2091.192	4073.694
x4	-8769.1216	536.127	-16.356	0.000	-9819.981	-7718.262
x5	-342.7102	597.846	-0.573	0.566	-1514.545	829.125
x6	-3287.2481	647.738	-5.075	0.000	-4556.876	-2017.621
x7	95.1233	709.275	0.134	0.893	-1295.123	1485.370
x8	2051.6037	745.202	2.753	0.006	590.937	3512.271

Omnibus:	13014.910	**Durbin-Watson:**	2.004
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	326688.145
Skew:	3.156	**Prob(JB):**	0.00
Kurtosis:	22.775	**Cond. No.**	1.98

Improvement after standarization but not the best model.

We are going to use Multiple Factor Analysis for the next calculation because MFA works best with both categorical and continuous variables.

In [24]:
```
from sklearn.decomposition import FactorAnalysis
```

In [33]:
```
fa = FactorAnalysis()
X_train_fa = fa.fit_transform(X_train)
X_test_fa = fa.transform(X_test)
```

In [39]:
```
pd.DataFrame(fa.components_)
```

Out[39]:

	0	1	2	3	4	5	6	
0	0.212486	0.619490	0.090751	0.080650	1065.088576	40.789294	787.918578	-0.01
1	-0.166071	0.170240	0.070700	-0.035539	-447.300638	1.789355	604.556959	-0.00
2	0.253265	0.218095	0.103842	0.050979	-2.315289	99.214674	-2.006687	-0.00
3	0.257755	1.984586	0.247091	0.133713	-0.000427	-0.005467	-0.000956	-0.01
4	-0.069756	0.301680	-0.406775	-0.290256	-0.000021	0.000165	0.000196	0.047
5	0.896366	-0.091501	0.110402	0.270713	-0.000202	-0.002562	0.000268	-0.00
6	-0.152551	0.059518	0.077531	0.099551	-0.000031	0.000240	-0.000120	-0.06
7	-0.038673	-0.002759	0.130854	0.098987	0.000005	-0.000069	-0.000033	0.061
8	0.000000	-0.000000	-0.000000	-0.000000	0.000000	-0.000000	-0.000000	0.000
9	-0.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	0.000000	-0.00
10	-0.000000	-0.000000	0.000000	-0.000000	0.000000	0.000000	-0.000000	-0.00
11	0.000000	-0.000000	0.000000	-0.000000	0.000000	-0.000000	0.000000	0.000
12	-0.000000	0.000000	-0.000000	0.000000	-0.000000	0.000000	0.000000	0.000
13	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	0.000000	-0.00

In [40]:
```
# import statsmodels.api as sm
```

In [41]:
```
X = X_train_fa
Y = Y_train
## fit a OLS model with intercept on TV and Radio
X = sm.add_constant(X)
est = sm.OLS(Y, X).fit()

est.summary()
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\statsmodels\regression\linear_model.py:1471: RuntimeWarning: divide b
y zero encountered in double_scalars
  return np.sqrt(eigvals[0]/eigvals[-1])
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\statsmodels\base\model.py:1036: RuntimeWarning: invalid value encount
ered in true_divide
  return self.params / self.bse
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\scipy\stats\_distn_infrastructure.py:879: RuntimeWarning: invalid val
ue encountered in greater
  return (self.a < x) & (x < self.b)
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\scipy\stats\_distn_infrastructure.py:879: RuntimeWarning: invalid val
ue encountered in less
  return (self.a < x) & (x < self.b)
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\scipy\stats\_distn_infrastructure.py:1818: RuntimeWarning: invalid va
lue encountered in less_equal
  cond2 = cond0 & (x <= self.a)
```

Out[41]:

OLS Regression Results

Dep. Variable:	FamilyIncome	**R-squared:**	0.322
Model:	OLS	**Adj. R-squared:**	0.322
Method:	Least Squares	**F-statistic:**	1080.
Date:	Sat, 14 Apr 2018	**Prob (F-statistic):**	0.00
Time:	14:15:40	**Log-Likelihood:**	-2.3200e+05
No. Observations:	18196	**AIC:**	4.640e+05
Df Residuals:	18187	**BIC:**	4.641e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	1.109e+05	618.309	179.354	0.000	1.1e+05	1.12e+05
x1	5.144e+04	618.309	83.194	0.000	5.02e+04	5.27e+04
x2	307.2691	618.309	0.497	0.619	-904.676	1519.214
x3	6263.8383	618.340	10.130	0.000	5051.833	7475.844
x4	1.788e+04	686.057	26.061	0.000	1.65e+04	1.92e+04
x5	-1.727e+04	830.771	-20.784	0.000	-1.89e+04	-1.56e+04
x6	374.6020	851.201	0.440	0.660	-1293.832	2043.036
x7	2.087e+04	972.628	21.455	0.000	1.9e+04	2.28e+04
x8	1.075e+04	1572.779	6.835	0.000	7666.428	1.38e+04
x9	0	0	nan	nan	0	0
x10	0	0	nan	nan	0	0
x11	0	0	nan	nan	0	0
x12	0	0	nan	nan	0	0
x13	0	0	nan	nan	0	0
x14	0	0	nan	nan	0	0

Omnibus:	13047.199	**Durbin-Watson:**	2.002
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	351119.557
Skew:	3.140	**Prob(JB):**	0.00
Kurtosis:	23.584	**Cond. No.**	inf

Now, we will use Factor Analysis on standardized values.

In [42]:
```
fa = FactorAnalysis()
X_train_fa = fa.fit_transform(X_train_std)
X_test_fa = fa.transform(X_test_std)
```

In [43]:
```
X = X_train_fa
Y = Y_train
## fit a OLS model with intercept on TV and Radio
X = sm.add_constant(X)
est = sm.OLS(Y, X).fit()

est.summary()
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\statsmodels\regression\linear_model.py:1471: RuntimeWarning: divide b
y zero encountered in double_scalars
  return np.sqrt(eigvals[0]/eigvals[-1])
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\statsmodels\base\model.py:1036: RuntimeWarning: invalid value encount
ered in true_divide
  return self.params / self.bse
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\scipy\stats\_distn_infrastructure.py:879: RuntimeWarning: invalid val
ue encountered in greater
  return (self.a < x) & (x < self.b)
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\scipy\stats\_distn_infrastructure.py:879: RuntimeWarning: invalid val
ue encountered in less
  return (self.a < x) & (x < self.b)
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\scipy\stats\_distn_infrastructure.py:1818: RuntimeWarning: invalid va
lue encountered in less_equal
  cond2 = cond0 & (x <= self.a)
```

Out[43]:

OLS Regression Results

Dep. Variable:	FamilyIncome	**R-squared:**	0.297
Model:	OLS	**Adj. R-squared:**	0.296
Method:	Least Squares	**F-statistic:**	1279.
Date:	Sat, 14 Apr 2018	**Prob (F-statistic):**	0.00
Time:	14:18:05	**Log-Likelihood:**	-2.3234e+05
No. Observations:	18196	**AIC:**	4.647e+05
Df Residuals:	18189	**BIC:**	4.647e+05
Df Model:	6		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	1.109e+05	629.759	176.093	0.000	1.1e+05	1.12e+05
x1	-4.243e+04	722.704	-58.707	0.000	-4.38e+04	-4.1e+04
x2	4.82e+04	777.258	62.016	0.000	4.67e+04	4.97e+04
x3	1.188e+04	869.471	13.660	0.000	1.02e+04	1.36e+04
x4	-1.021e+04	926.211	-11.024	0.000	-1.2e+04	-8394.796
x5	-1358.2937	1112.100	-1.221	0.222	-3538.114	821.527
x6	-1.685e+04	1989.893	-8.470	0.000	-2.08e+04	-1.3e+04
x7	0	0	nan	nan	0	0
x8	0	0	nan	nan	0	0
x9	0	0	nan	nan	0	0
x10	0	0	nan	nan	0	0
x11	0	0	nan	nan	0	0
x12	0	0	nan	nan	0	0
x13	0	0	nan	nan	0	0
x14	0	0	nan	nan	0	0

Omnibus:	13026.187	**Durbin-Watson:**	2.005
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	328576.647
Skew:	3.158	**Prob(JB):**	0.00
Kurtosis:	22.837	**Cond. No.**	inf

Not so good results again. It is not worth it to test our predictions.