In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

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

In [3]:
```
data.head()
```

Out[3]:

	Acres	FamilyIncome	FamilyType	NumBedrooms	NumChildren	NumPeople	NumRo
0	10-Jan	150	Married	4	1	3	9
1	10-Jan	180	Female Head	3	2	4	6
2	10-Jan	280	Female Head	4	0	2	8
3	10-Jan	330	Female Head	2	1	2	4
4	10-Jan	330	Male Head	3	1	2	5

In [4]:
```
# data.info()
# data.describe()
```

In [5]:
```
# data.isnull().sum()
```

In [6]:
```
x = data.iloc[:, 2:]
y = data.iloc[:, 1:2]
```

In [7]:
```
encoded_x = pd.get_dummies(x,drop_first = True)
```

In [8]:
```
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)
```

In [9]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [10]:
```
from sklearn.decomposition import PCA
pca = PCA(n_components = None)
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
```

21 PCA components would explain 70% of the variance. So, n_components = 20

In [11]:
```
pca = PCA(n_components = 21)
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
```

In [12]:
```
explained_variance.sum()
```

Out[12]:
```
0.70782659642493506
```

In [13]:
```
from sklearn.linear_model import LinearRegression
regression = LinearRegression()
lm = regression.fit(X_train,Y_train)
```

In [14]:
```
y_pred = regression.predict(X_test)
```

In [15]:
```
print(lm.intercept_)
print(lm.coef_)
```

```
[ 110896.03566718]
[[ 25740.49144811   6826.07957203 -14733.98755492   6646.11675194
   -4935.35405931  10827.15269498  -1862.35518609  -3925.32898871
   -2716.33715587   2475.43489519   1623.68203507   1465.18796052
     300.53492013   -565.93740687  -1008.7609598    1314.24719896
    2488.7288067   -1935.96478188   2705.99290428   1044.20067516
    2087.79900133]]
```

In [16]:
```
import sklearn
```

In [17]:
```
#we will take out the R^2 value
sklearn.metrics.r2_score(Y_test, y_pred)
```

Out[17]:
```
0.30358744933024473
```

In [18]:
```
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 [19]:
```
X = X_train
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[19]:

OLS Regression Results

Dep. Variable:	FamilyIncome	**R-squared:**	0.291
Model:	OLS	**Adj. R-squared:**	0.291
Method:	Least Squares	**F-statistic:**	355.8
Date:	Sat, 14 Apr 2018	**Prob (F-statistic):**	0.00
Time:	00:44:04	**Log-Likelihood:**	-2.3241e+05
No. Observations:	18196	**AIC:**	4.649e+05
Df Residuals:	18174	**BIC:**	4.650e+05
Df Model:	21		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	1.109e+05	632.424	175.351	0.000	1.1e+05	1.12e+05
x1	2.574e+04	355.196	72.468	0.000	2.5e+04	2.64e+04
x2	6826.0796	399.064	17.105	0.000	6043.877	7608.282
x3	-1.473e+04	462.964	-31.825	0.000	-1.56e+04	-1.38e+04
x4	6646.1168	474.137	14.017	0.000	5716.764	7575.470
x5	-4935.3541	497.710	-9.916	0.000	-5910.913	-3959.795
x6	1.083e+04	515.938	20.985	0.000	9815.866	1.18e+04
x7	-1862.3552	550.660	-3.382	0.001	-2941.700	-783.010
x8	-3925.3290	568.021	-6.911	0.000	-5038.704	-2811.954
x9	-2716.3372	575.780	-4.718	0.000	-3844.920	-1587.754
x10	2475.4349	588.094	4.209	0.000	1322.714	3628.155
x11	1623.6820	593.402	2.736	0.006	460.558	2786.806
x12	1465.1880	596.525	2.456	0.014	295.943	2634.433
x13	300.5349	600.576	0.500	0.617	-876.651	1477.721
x14	-565.9374	605.755	-0.934	0.350	-1753.275	621.400
x15	-1008.7610	614.263	-1.642	0.101	-2212.775	195.253
x16	1314.2472	618.231	2.126	0.034	102.456	2526.038
x17	2488.7288	621.639	4.003	0.000	1270.257	3707.201
x18	-1935.9648	625.307	-3.096	0.002	-3161.626	-710.303
x19	2705.9929	626.278	4.321	0.000	1478.429	3933.556
x20	1044.2007	629.683	1.658	0.097	-190.038	2278.440

x21	2087.7990	631.170	3.308	0.001	850.647	3324.951

Omnibus:	13122.618	**Durbin-Watson:**	2.005
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	334621.308
Skew:	3.189	**Prob(JB):**	0.00
Kurtosis:	23.017	**Cond. No.**	1.78