In [1]: import pandas as pd import numpy as np import seaborn as sns import matplotlib.pyplot as plt %matplotlib inline import cufflinks as cf from plotly.offline import download plotlyjs, init notebook mode, plot, iplot init notebook mode(connected = True) cf.go offline() In []: In [3]: us housing = pd.read csv("USAHousing.csv") In [10]: us housing Out[10]: Avg. Area Number Avg. Area Avg. Area Avg. Area Number of Area Address **Price** Income **House Age** of Rooms **Bedrooms Population** 208 Michael Ferry Apt. 0 79545.45857 5.682861 7.009188 23086.80050 1.059034e+06 674\nLaurabury, NE 3701... 188 Johnson Views Suite 40173.07217 1.505891e+06 **1** 79248.64245 6.002900 6.730821 3.09 079\nLake Kathleen, CA... 9127 Elizabeth Stravenue\nDanieltown, WI **2** 61287.06718 5.865890 8.512727 5.13 36882.15940 1.058988e+06 06482... USS Barnett\nFPO AP 44820 63345.24005 34310.24283 1.260617e+06 7.188236 5.586729 USNS Raymond\nFPO AE 59982.19723 5.040555 7.839388 26354.10947 6.309435e+05 09386 USNS Williams\nFPO AP 60567.94414 6.137356 22837.36103 1.060194e+06 4995 7.830362 30153-7653 PSC 9258, Box 8489\nAPO AA 78491.27543 6.999135 6.576763 25616.11549 1.482618e+06 4996 4.02 42991-3352 4215 Tracy Garden Suite 4997 63390.68689 7.250591 4.805081 33266.14549 1.030730e+06 076\nJoshualand, VA 01... 4998 68001.33124 5.534388 7.130144 42625.62016 1.198657e+06 USS Wallace\nFPO AE 73316 37778 George Ridges Apt. 65510.58180 6.792336 4999 5.992305 4.07 46501.28380 1.298950e+06 509\nEast Holly, NV 2... 5000 rows × 7 columns In [11]: us housing.count() Out[11]: Avg. Area Income 5000 Avg. Area House Age 5000 Avg. Area Number of Rooms 5000 Avg. Area Number of Bedrooms 5000 Area Population 5000 Price 5000 Address 5000 dtype: int64 In [67]: us housing.tail() Out[67]: Avg. Area Avg. Area Number Avg. Area Avg. Area Number of Area **Price Address** House Age Income of Rooms **Bedrooms Population** USNS Williams\nFPO AP 30153-3.46 22837.36103 1060193.786 **4995** 60567.94414 7.830362 6.137356 PSC 9258, Box 8489\nAPO AA 78491.27543 6.576763 4.02 25616.11549 1482617.729 4996 6.999135 42991-3352 4215 Tracy Garden Suite 63390.68689 4.805081 1030729.583 4997 7.250591 2.13 33266.14549 076\nJoshualand, VA 01... 7.130144 1198656.872 4998 68001.33124 5.534388 42625.62016 USS Wallace\nFPO AE 73316 37778 George Ridges Apt. 5.992305 4.07 46501.28380 4999 65510.58180 6.792336 1298950.480 509\nEast Holly, NV 2... In [73]: us housing['Avg. Area Number of Bedrooms'].nunique() Out[73]: 255 In [6]: sns.heatmap(us_housing.isnull()) Out[6]: <matplotlib.axes. subplots.AxesSubplot at 0x1b715d47070> - 0.100 239 478 717 956 1195 1434 1673 1912 2151 2390 2629 2868 3107 3346 3585 - 0.075 0.050 0.025 0.000 -0.025 -0.050 3824 4063 4302 -0.075 4541 4780 -0.100Address Avg. Area Income Price Avg. Area House Age Avg. Area Number of Rooms Avg. Area Number of Bedrooms Area Population In [8]: us housing.columns Out[8]: Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'], dtype='object') In [44]: us housing.iloc[465] Out[44]: Avg. Area Income 90592.5 7.70013 Avg. Area House Age Avg. Area Number of Rooms 9.7088 Avg. Area Number of Bedrooms 5.19 37223.9 Area Population Price 2.46907e+06 USNS Vargas\nFPO AE 56319-6904 Address Name: 465, dtype: object print(us housing['Price'].max()) In [46]: 2469065.594 print(us housing[us housing.Price ==us housing.Price.min(axis =0)]) In [50]: Avg. Area Income Avg. Area House Age Avg. Area Number of Rooms 3212 47320.65721 3.558054 7.006987 Avg. Area Number of Bedrooms Area Population 3212 15776.6186 15938.65792 Address 3212 91410 Megan Camp Suite 360\nLaurafort, OH 15735 In [51]: print(us_housing[us_housing.Price == us_housing.Price.max()]) Avg. Area Income Avg. Area House Age Avg. Area Number of Rooms 465 90592.46961 Avg. Area Number of Bedrooms Area Population Price \ 465 5.19 37223.87617 2469065.594 Address USNS Vargas\nFPO AE 56319-6904 In [60]: | sns.jointplot(us_housing['Avg. Area Number of Rooms'], us_housing['Price'], kind='scatter', color = 'red') Out[60]: <seaborn.axisgrid.JointGrid at 0x1b7192113a0> 2.5 2.0 1.5 Price 1.0 0.5 0.0 11 10 Avg. Area Number of Rooms In [62]: sns.pairplot(us housing) Out[62]: <seaborn.axisgrid.PairGrid at 0x1b71b336d60> 100000 Area Income 80000 60000 40000 20000 Avg. Area House Age Avg. Area Number Area Number of Bedrooms 60000 40000 Ag 20000 2.5 2.0 e 1.5 1.0 0.5 0.0 25000 50000 75000 100000 Avg. Area Income 20000 40000 60000 5.0 7.5 10.0 i Price Avg. Area House Age Area Population Avg. Area Number of Rooms Avg. Area Number of Bedrooms In [64]: us_housing.corr(method = 'pearson') Out[64]: Avg. Area House Avg. Area Avg. Area Number of Avg. Area Number of **Area Price** Income **Population** Age Rooms **Bedrooms** 0.639734 -0.002007 -0.011032 0.019788 -0.016234 Avg. Area Income 1.000000 -0.009428 -0.018743 Avg. Area House Age -0.002007 1.000000 0.006149 0.452543 Avg. Area Number of 0.002040 0.335664 -0.011032 -0.009428 1.000000 0.462695 **Rooms** Avg. Area Number of 0.019788 0.006149 0.462695 1.000000 -0.022168 0.171071 **Bedrooms** -0.016234 -0.022168 **Area Population** -0.018743 0.002040 1.000000 0.408556 **Price** 0.639734 0.452543 0.335664 0.171071 0.408556 1.000000 In [66]: sns.heatmap(us_housing.corr(method = 'pearson')) Out[66]: <matplotlib.axes._subplots.AxesSubplot at 0x1b71dd049a0> - 1.0 Avg. Area Income - 0.8 Avg. Area House Age 0.6 Avg. Area Number of Rooms Avg. Area Number of Bedrooms 0.4

Area Population 0.2 Price Price Area Population Avg. Area House Age of Rooms Bedrooms Avg. Area Number In [90]: us_housing(us_housing.) Out[90]:

0 79545.45857 5.682861 7.009188 79248.64245 1 6.002900 6.730821 8.512727 61287.06718 5.865890 63345.24005 3 7.188236 5.586729 59982.19723 5.040555 7.839388 ••• 4995 60567.94414 7.830362 6.137356 4996 78491.27543 6.999135 6.576763 4997 63390.68689 4.805081 7.250591 68001.33124 4998 5.534388 7.130144 4999 65510.58180 5.992305 6.792336 5000 rows × 6 columns In [124]: X = us housing.drop('Price', axis= 1).columns In [127]: x.columns 'Avg. Area Number of Bedrooms', 'Area Population'], dtype='object') In []:

Price

23086.80050 1.059034e+06

40173.07217 1.505891e+06

36882.15940 1.058988e+06

34310.24283 1.260617e+06

26354.10947 6.309435e+05

22837.36103 1.060194e+06

25616.11549 1.482618e+06

33266.14549 1.030730e+06

42625.62016 1.198657e+06

46501.28380 1.298950e+06

4.09

3.09

5.13

3.26

4.23

... 3.46

4.02

2.13

5.44

4.07

from sklearn.linear_model import LinearRegression

Avg. Area Income Avg. Area House Age Avg. Area Number of Rooms Avg. Area Number of Bedrooms Area Population Out[127]: Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', In [95]: y = us_housing['Price'] In [86]: from sklearn.model_selection import train test split In [89]: reg = LinearRegression() In [133]: reg.fit(us_housing[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Avg. Area Number of Bedrooms', 'Area Population']],us_housing.Price)

Out[133]: LinearRegression() In [135]: us housing.iloc[46] Out[135]: Avg. Area Income 7.394685e+04 Avq. Area House Age 4.863154e+00 Avg. Area Number of Rooms 7.537182e+00 6.350000e+00 Avg. Area Number of Bedrooms Area Population 3.526113e+04 1.109588e+06

Out[142]: -2637299.0333314524

Out[146]: 0.9165644843776913

In [146]: r_score

In []:

In [144]: r_score = reg.score(X_test, y_test)

Name: 46, dtype: float64 In [136]: reg.predict([[50000,5,3,5,73734]]) Out[136]: array([760835.76798376]) In [140]: reg.coef_ Out[140]: array([2.15780494e+01, 1.65637027e+05, 1.20659949e+05, 1.65113905e+03, 1.52007439e+01]) In [142]: reg.intercept_