Question 1:

Why is it a good idea to standardize/normalize the predictor variables 2 and 3 and why are predictor variables 4 and 5 probably not very useful by themselves to predict median house values in a block?

This question deals with one of the key aspects we need to account for when working with real-world data. Indeed, before applying - in our case linear regression- we need to make sure our data is ready and do some refining and that includes checking whether it requires any standardization /normalization.

- **Part1: Why is it a good idea to standardize/normalize the predictor variables 2 and 3**

Standardizing predictor variables of the total number of rooms (2) and the number of bedrooms (3) is a good idea as we need to make sure that the housing blocks are comparable, thus, they need to be on the same scale. For better visualization of the idea, we can consider a house with 3 bedrooms and 4 bathrooms, bathrooms could also be considered as rooms, also the raw number of rooms is not sufficient to predict the house value on its own and thus the raw number of rooms or bedrooms is irrelevant in itself. Therefore, for optimal results when we do linear regression and be able to predict the housing value as accurately as possible, we need to normalize these 2 features. Indeed, normalized data helps the model to learn better and perform at its best.

- **Part 2: why are predictor variables 4 and 5 probably not very useful by themselves to predict median house values in a block**

This question falls back from the last question and is also part of the refining we need to ensure before running our model. Looking at the predictor variable population in block (4) and the number of households in the block (5), we see that these two features give us more information about the block rather than the value of the house itself. Therefore, utilizing them as predictors for our linear regression will not allow us to get a better sense of the house price or value. Indeed, they still should not be discarded, and combining them with the other features (as we will see in the next question) will make them useful to predict house values in a block.
To answer the question from an analytical and statistical point of view, we can confirm our intuition by looking at the correlation coefficient between each of the two features and the median house values.
Considering that this question is asking about predictors 4 and 5 being good/bad predictors, therefore, it is implicitly asking us do they constitute good ways to consider to predict a median house value. Given our data has all the features as well as the median house value, finding the

correlation between 4 and 5 individually with regards to the desired output will give us a good idea of whether they could be used for ensuring a good performance of our model. Therefore, we simply call the pre defined Python function corr() to the housing data and get our correlation coefficients for all the columns of the housing data. (Please refer to the submitted assignment for the full version of the code).

Indeed, we find results that match our initial intuition and approach to the question. As we can see in Figure 1 and as highlighted in light blue, the correlation between the population and the median house value is weak with -0.026882 correlation as well as a 0.064590 correlation between the number of households and the median house value. As a strong correlation is one of the key features of good predictors, it is clear then that variables 4 and 5 will not be very useful on their own to predict the median house price considering the low correlation.

Out[105]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
longitude	1.000000	-0.924478	-0.105823	0.048909	0.076686	0.108071	0.063146	-0.019615	-0.047466
latitude	-0.924478	1.000000	0.005737	-0.039245	-0.072550	-0.115290	-0.077765	-0.075146	-0.142673
housing_median_age	-0.105823	0.005737	1.000000	-0.364535	-0.325101	-0.298737	-0.306473	-0.111315	0.114146
total_rooms	0.048909	-0.039245	-0.364535	1.000000	0.929391	0.855103	0.918396	0.200133	0.135140
total_bedrooms	0.076686	-0.072550	-0.325101	0.929391	1.000000	0.876324	0.980167	-0.009643	0.047781
population	0.108071	-0.115290	-0.298737	0.855103	0.876324	1.000000	0.904639	0.002421	-0.026882
households	0.063146	-0.077765	-0.306473	0.918396	0.980167	0.904639	1.000000	0.010869	0.064590
median_income	-0.019615	-0.075146	-0.111315	0.200133	-0.009643	0.002421	0.010869	1.000000	0.687151
median_house_value	-0.047466	-0.142673	0.114146	0.135140	0.047781	-0.026882	0.064590	0.687151	1.000000

Fig. 1.

Question 2:

To meaningfully use predictor variables 2 (number of rooms) and 3 (number of bedrooms), you will need to standardize/normalize them. Using the data, is it better to normalize them by population (4) or number of households (5)?

This question involves working with 4 different cases;

a. either using the predictor of the number of rooms (2) and normalizing it by population (4), or

b. using the predictor of the number of rooms (2) and normalizing it by the number of households (5), or
c. using the predictor of the number of bedrooms (3) and normalizing it by the number of population (4), or
d. using the predictor of the number of rooms (2) and normalizing it by the number of households (5)

Following the logic and reasoning of the first question, this question focuses on the actual normalization of the predictor variables 2 and 3 by population (4) and the number of households (5) and then compares which normalization performs best through computing R^2 for each of the cases. Indeed, applying the linear regression for each of the normalization cases will allow us to clearly identify whether it is better to normalize predictors 2 and 3 by population or the number of households; the actual difference will be seen by computing the R^squared for each.

The approach taken to normalize in this question is dividing the predictor (either 2 or 3) by the population or number of households. Also, for the running we are using the fit function we defined in the SimpleLinearRegression class and pass in the variable predictor we are interested in. Additionally, for the R^squared the function r_squared() has been defined in the class SimpleLinearRegression and we just call it after the fitting with linear regression; simply lr.r_squared().

Lastly, for better visualization of the actual training performance beyond just the R squared value, we plot the data as well as the linear regression line. We then visually compare and contrast both to see how well they match and whether there are conclusions to be drawn; did the model perform well or not?

A. Using the predictor of the number of rooms (2) and normalizing it by population (4)

For this case we will normalize by updating the number of rooms to be the number of rooms (2) divided by the population (4) (as clearly shown in the code). As mentioned previously, in order to evaluate how good is this normalisation with regards to the following ones we will be doing, we need to get a certain value to compare; we are choosing the R^2 method (there are also other methods to evaluate the performance of models, indeed, we go with this one for this homework).

After running this case, the R-squared value we get is 0.03976822198867225.

Additionally, we also make a plot for this case (please refer to Figure 2 bellow); the plot shows that the model and the betas we got were not really that precise as we are not seeing a match between the linear regression line and the actual data.

```
In [51]: # plotting
         plt.scatter(X, y)
         plt.plot([min(X), max(X)], [min(y_pred), max(y_pred)], color='red')  # regression line
         plt.show()
         # SHOULD be able all points that there is and line
```

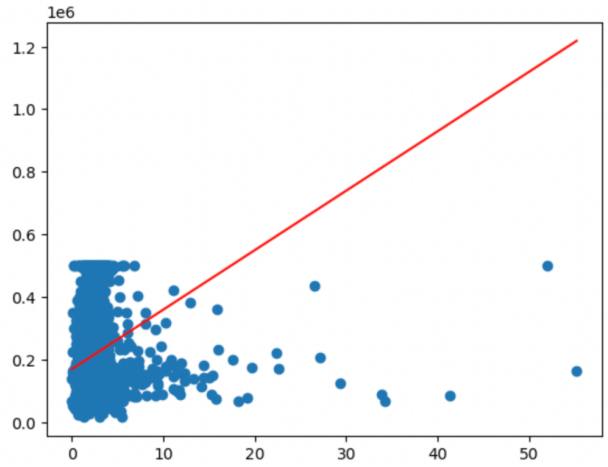


Fig. 2.

B. Using the predictor of the number of rooms (2) and normalizing it by the number of households (5)

For this case we will be updating the number of rooms to be the number of rooms (2) divided by the population (4) (as clearly shown in the code).

After running this case, the R-squared value we get is 0.021390582248860812 which is a bit lower compared to the previous case.

Additionally, we also make a plot for this case (please refer to Figure 3 below); the plot shows that the model and the betas we got were not really as precise as we are not seeing a clear match between the linear regression line and the actual data.

```
In [187]: # plotting (for question 2)
          plt.scatter(X, y)
          plt.plot([min(X), max(X)], [min(y_pred), max(y_pred)], color='red')  # regression line
          plt.show()
```

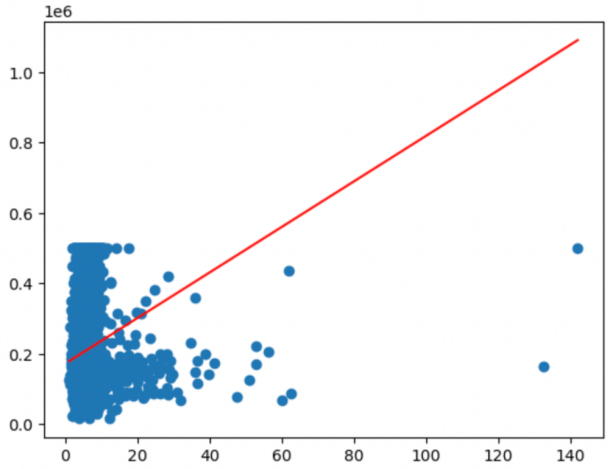


Fig.3.

C. Using the predictor of the number of bedrooms (3) and normalizing it by the number of population (4),

For this case we will be updating the number of rooms to be the number of bedrooms (3) divided by the population (4) (as clearly shown in the code).

After running this case, the R-squared value we get is nan.

Additionally, we also make a plot for this case (please refer to Figure 4 below); the plot shows that the model prediction was off in a sense to be not very predictive and following the distribution of the data. Indeed, it makes sense, considering that the R squared we got is nan or

not a value making this case definitely not a good way to predict the house value.

```
In [552]: # plotting (for question 2)
          plt.scatter(X, y)
          plt.plot([min(X), max(X)], [min(y_pred), max(y_pred)], color='red')  # regression line
          plt.show()
```

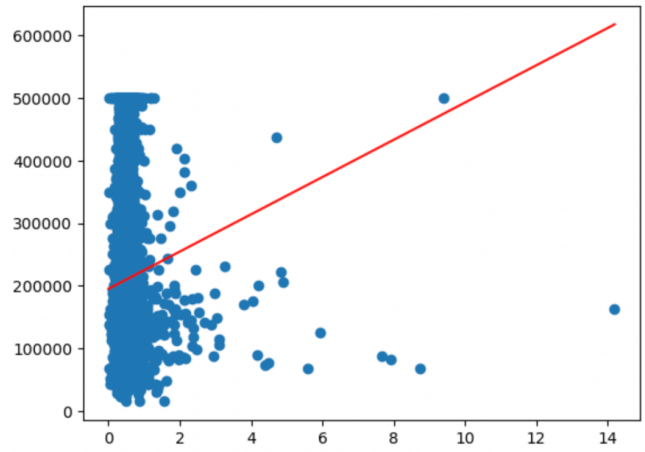


Fig. 4.

D. using the predictor of the number of bedrooms (3) and normalizing it by the number of households (5)

For this case we will be updating the number of rooms to be the number of bedrooms (3) divided by the population (5) (as clearly shown in the code).

After running this case, the R-squared value we get is nan.

Additionally, we also make a plot for this case (please refer to Figure 4 below); the plot shows that the model prediction was off in a sense to be not very predictive and following the distribution of the data. Indeed, it makes sense, considering that the R squared we got is nan or not a value making this case definitely not a good way to predict the house value.

```
In [676]: # plotting (for question 2)
          plt.scatter(X, y)
          plt.plot([min(X), max(X)], [min(y_pred), max(y_pred)], color='red')  # regression line
          plt.show()
```

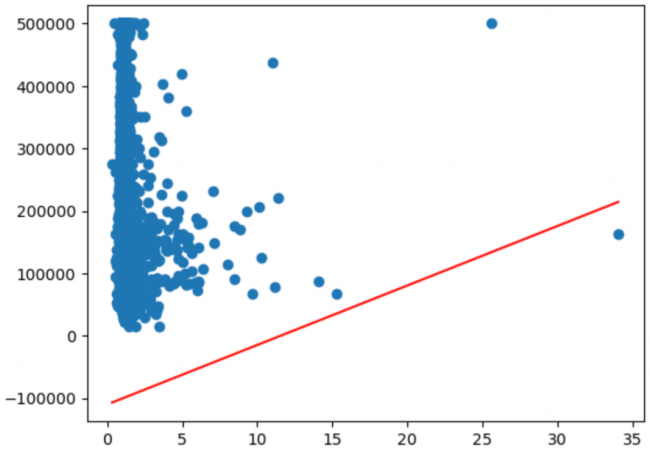


Fig. 5.

Conclusion:

The results found are very interesting as they show the complexity of the preprocessing and how it should be given a lot of attention before focusing on the linear regression model itself.

According to the results we got we can conclude that the best-performing model out of the 2 cases for the predictor variable 2 (normalizing by variable 4 or 5) is the normalization by population (4) as it has the highest R squared value of 0.03976822198867225.

Indeed, for the predictor variable of the number of bedrooms cases, the results were quite confusing as we got an r-squared that is not a number.

Question 3:

Which of the seven variables is most *and* least predictive of housing value, from a simple linear regression perspective? [Hints: a) Make sure to use the standardized/normalized variables from 2. above; b) Make sure to inspect the scatter plots and comment on a potential issue – would the best predictor be even more predictive if not for an unfortunate limitation of the data?]

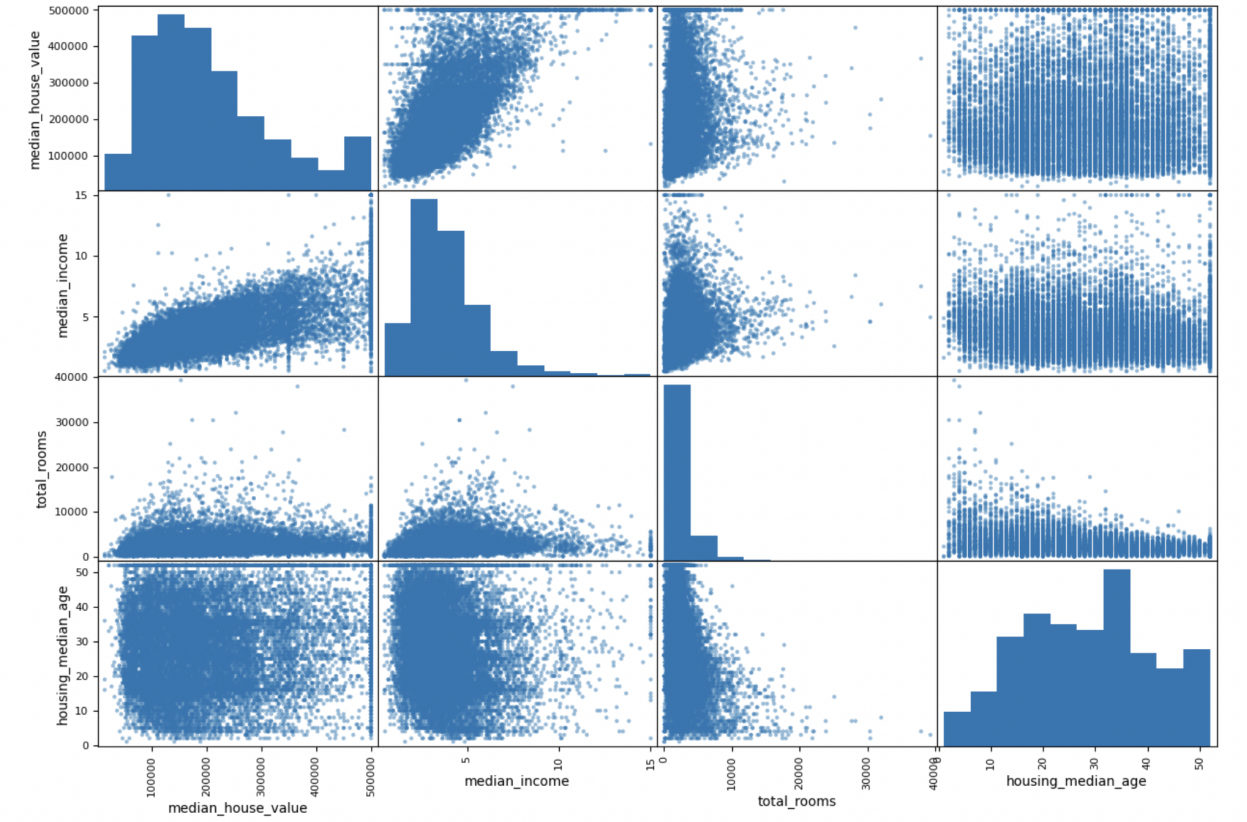
The approach for this question is running the model on each of the 7 variable predictors and comparing them by looking at their respective R-squared values the ones with the strongest R-squared will be the best predictive of housing value and the one with the weakest R will be the least predictive of housing value.

To get an idea about what we should expect, the approach we could implement is first looking at the correlation coefficients of each of these predictors with respect to the housing value. The lowest correlation coefficient will be for the least predictive feature and the highest correlation coefficient will be for the most predictive feature.

Out[320]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
longitude	1.000000	-0.924664	-0.108197	0.044568	0.069608	0.099773	0.055310	-0.015176	-0.045967
latitude	-0.924664	1.000000	0.011173	-0.036100	-0.066983	-0.108785	-0.071035	-0.079809	-0.144160
housing_median_age	-0.108197	0.011173	1.000000	-0.361262	-0.320451	-0.296244	-0.302916	-0.119034	0.105623
total_rooms	0.044568	-0.036100	-0.361262	1.000000	0.930380	0.857126	0.918484	0.198050	0.134153
total_bedrooms	0.069608	-0.066983	-0.320451	0.930380	1.000000	0.877747	0.979728	-0.007723	0.049686
population	0.099773	-0.108785	-0.296244	0.857126	0.877747	1.000000	0.907222	0.004834	-0.024650
households	0.055310	-0.071035	-0.302916	0.918484	0.979728	0.907222	1.000000	0.013033	0.065843
median_income	-0.015176	-0.079809	-0.119034	0.198050	-0.007723	0.004834	0.013033	1.000000	0.688075
median_house_value	-0.045967	-0.144160	0.105623	0.134153	0.049686	-0.024650	0.065843	0.688075	1.000000

Indeed, we find that the highest (absolute value) of correlation coefficients to median house



value is median income, making it the most predictive of housing value with a correlation of 0.688075. For the least predictive, we find -0.024650 as the smallest absolute value making population as the least predictive feature for the housing value. Theses results also go with the standard intuition as indeed the median income impacts a lot on the value of a house, indeed, the population does not really impact on whether the price of a house is small or big.

On the same note, as shown in the plot figure above we find a strong linear relationship in the plot of median income against the median house value, thus, goes with the results found by looking at the correlation coefficients.

Now, we can actually move to answer to the question with doing the training for each of the predictor variables and comparing the R^2.

1) Median age of the houses in the block: 0.013029298376515785
2) Total number of rooms in a given block: 0.03976822198867225.
3) Number of bedrooms in a given block: nan
4) Population in the block: 0.0007226421149140183
5) Number of households in the block: 0.004171891913933079
6) Median household income in the block: 0.47217589093803647
7) Proximity to the ocean: nan

Therefore, we can conclude that the most predictive feature is household income in the block and the least predictive is population; which confirms our initial findings.

Question 4:

Putting all predictors together in a multiple regression model – how well do these predictors taken together predict housing value? How does this full model compare to the model that just has the single best predictor from 3.?

This question deals with the Multivariate linear regression which code we can also find with the submission. In order to answer the question, we need to compare R squared values between what we found in the previous questions and the R squared we got for this regression.

The R^2 found by using all the predictors is 0.8023764076009697 which is way higher than all the previous R-squared values we found in the last question. Therefore, it is indeed better to use Multivariate linear regression to predict the housing value. Additionally, the results also make sense and go along with the general logic as the more predictors and variables you have to predict the value of a house, the better the prediction as in real life a value of a house does not only depend on one parameter.

Question 5:

Considering the relationship between the (standardized) variables 2 and 3, is there potentially a concern regarding collinearity? Is there a similar concern regarding variables 4 and 5, if you were to include them in the model?

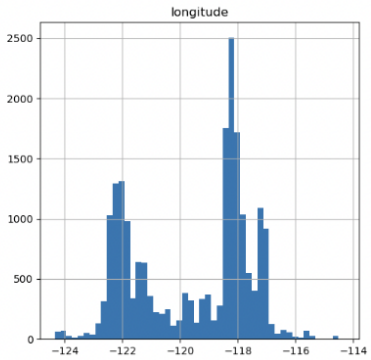
Indeed, there is an issue with colinearity as from a logical perspective the number of rooms nad the number of bedrooms are relatively the same feature in real life; you could count bedrooms

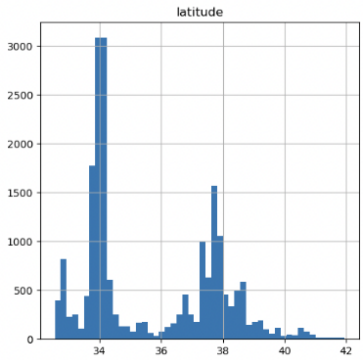
as rooms as well when you do your data acquisition. Thus, there is a lot of room of this aspect not to allow our model to perform to its best, therefore, we need to get rid of one of the two to make the model prediction better. Indeed, when we see the correlation table in our code and in the figure previously shown above, we see a very strong correlation between these two features. This means the regression coefficients will not uniquely determined. Therefore, it will hurt the interpretability of the model as then the regression coefficients are not unique and have influences from other features in our case having 2 features pretty much the same.

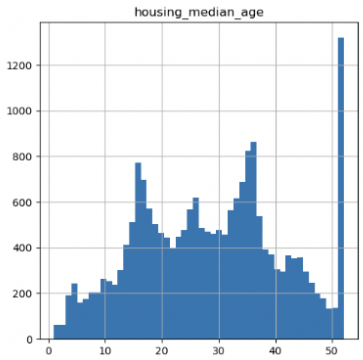
Extra credit:

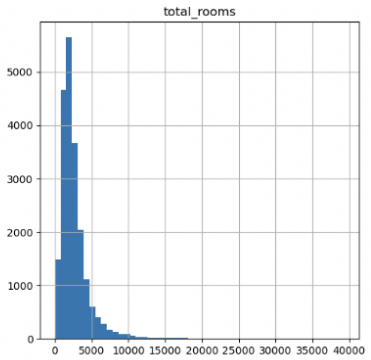
a. Does any of the variables (predictor or outcome) follow a distribution that can reasonably be described as a normal distribution?
b. Examine the distribution of the outcome variable. Are there any characteristics of this distribution that might limit the validity of the conclusions when answering the questions above? If so, please comment on this characteristic.

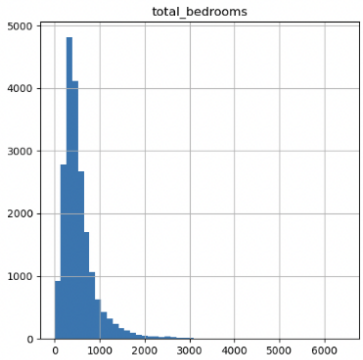
A. Indeed, looking at the plots bellow, we find that many of the features follow distributions that can be described as normal; for instance, the median income. Indeed, for the total rooms, bedrooms, population and households, it is nearly normal as we would have to do some cropping and remove some ranges of x that do not make the data normally distributed. We could also calculate the mean and std of each and see if it is close to 0 and 1 respectively to also help us see whether the distribution is normal.
B. Indeed, looking at the median house value in the plot below, we see that the data is not clearly normally distributed as it is quite uneven and most of the data is concentrated in the first half of the range. Another limitation is the chuck of data towards the end of the x range which does not follow the rest of the data and requires cropping. In fact, methods such as box plotting would allow us to make the data more normaly distributed.

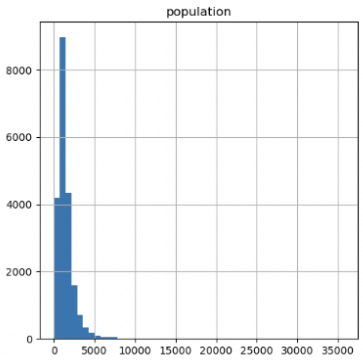

longitude
2500
2000
1500
1000
500
0
−124
−122
−120
−118
−116
−114

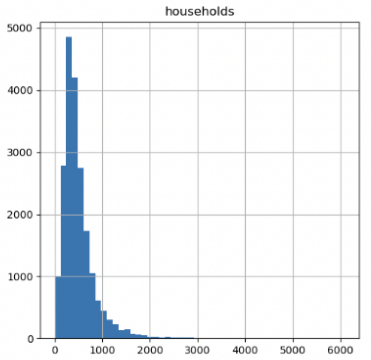

latitude
3000
2500
2000
1500
1000
500
0
34
36
38
40
42

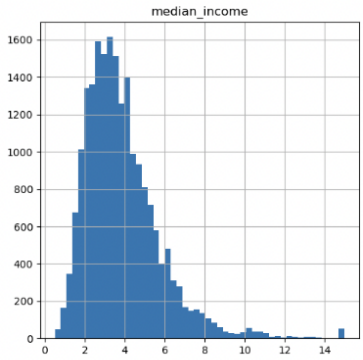

housing_median_age
1200
1000
800
600
400
200
0
0
10
20
30
40
50

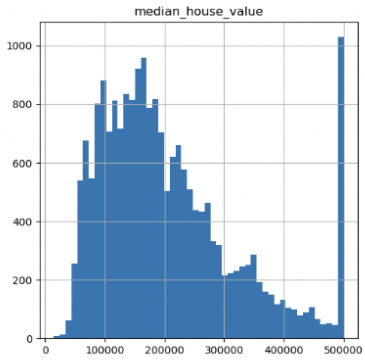

total_rooms
5000
4000
3000
2000
1000
0
0
5000
10000
15000
20000
25000
30000
35000
40000


total_bedrooms
5000
4000
3000
2000
1000
0
0
1000
2000
3000
4000
5000
6000


population
8000
6000
4000
2000
0
0
5000
10000
15000
20000
25000
30000
35000


households
5000
4000
3000
2000
1000
0
0
1000
2000
3000
4000
5000
6000


median_income
1600
1400
1200
1000
800
600
400
200
0
0
2
4
6
8
10
12
14


median_house_value
1000
800
600
400
200
0
0
100000
200000
300000
400000
500000