Data Analytics

Analisis data adalah teknik menganalisis dan memeriksa serta membersihkan dan mentransformasikan data untuk mengambil maklumat yang berguna atau mencadangkan penyelesaian. Selain daripada itu, proses ini turut mampu membantu dalam membuat keputusan untuk perniagaan atau urusan lain. Dalam tutorial ini, anda akan belajar mengenai pelbagai jenis analisis data dan cara penggunaannya.

How is Data Analyzed

Analisis data penyelidikan kualitatif terdiri daripada metodologi statistik yang menganalisis iteratif data dan lebih mudah jika dibandingkan berdasarkan nilai. Selain itu, para saintis dan penganalisis data turut mencari corak dalam pelbagai pemerhatian dalam data.

Descriptive

Kira-kira 90% syarikat dan organisasi mengikuti pendekatan analisis data ini. Ini adalah pendekatan kuantitatif yang menguraikan ciri utama dalam pengumpulan data dan memberitahu kembali kepada penganalisis dengan satu jawapan penting - "Apa yang Terjadi?" dengan lebih jelas. Fokus utama menggunakan teknik analisis data ini adalah untuk mengetahui alasan dan sebab di sebalik kenaikan atau kejatuhan perniagaan yang mahal pada masa lalu dan penyelesaiannya untuk masa depan. Ia kebanyakannya digunakan dalam bidang BI (Business Intelligence) dan data mining.

Inferential

Teknik ini bertujuan untuk menguji teori mengenai tingkah laku dan sifat data apa pun bergantung pada beberapa sampel "subjek" atau "hasil" yang diambil dari pemerhatian. Ini mencerminkan matlamat model statistik namun ia masih bergantung terutamanya kepada populasi data dan juga skema pensampelan.

Predictive

Teknik analisis ini menambahkan rasa mengawal tuntutan masa depan dan mengakses risiko. Tambahan lagi, ia mengarahkan pelbagai hasil yang mungkin akan meningkatkan perniagaan utama, terutamanya berkaitan dengan "Apa yang harus dilakukan oleh prestasi perniagaan untuk mencapai tujuan tertentu?"

Jenis analisis ini menggunakan konsep maju seperti di bawah:

1. **Mengoptimumkan untuk mencapai hasil terbaik.**
2. **Pengoptimuman stokastik yang membantu dalam memahami cara mencapai dan mengenal pasti keraguan data untuk membuat keputusan yang lebih baik.**

In [ ]:

```
# Mulakan dengan mengimport modul yang diperlukan
import pandas as pd
import numpy as np
```

In [ ]:

```
# Untuk contoh kali ini, kita akan menggunakan data yang dikongsi oleh Google, iaitu cali
fornia housing price dataset
train = pd.read_csv("sample_data/california_housing_train.csv")
test = pd.read_csv("sample_data/california_housing_test.csv")

print(train.shape, test.shape)

data = pd.concat([train, test])

print(data.shape)
```

```
(17000, 9) (3000, 9)
(20000, 9)
```

In [ ]:

```
data.reset_index(drop=True, inplace=True)
```

Columns Description

About this data

1. **longitude: A measure of how far west a house is; a higher value is farther west**
2. **latitude: A measure of how far north a house is; a higher value is farther north**
3. **housingMedianAge: Median age of a house within a block; a lower number is a newer building**
4. **totalRooms: Total number of rooms within a block**
5. **totalBedrooms: Total number of bedrooms within a block**
6. **population: Total number of people residing within a block**
7. **households: Total number of households, a group of people residing within a home unit, for a block**
8. **medianIncome: Median income for households within a block of houses (measured in tens of thousands of US Dollars)**
9. **medianHouseValue: Median house value for households within a block (measured in US Dollars)**

Source : https://www.kaggle.com/camnugent/california-housing-prices

In [ ]:

```
# Mulakan dengan melihat contoh data yang terdapat dalam dataset
data.sample(10)
```

Out[ ]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	media
19434	-117.42	34.10	18.0	3977.0	809.0	2231.0	742.0	4.1399	
1380	-117.17	32.73	52.0	55.0	18.0	65.0	22.0	1.6591	
12426	-121.58	39.51	24.0	1865.0	372.0	1087.0	385.0	1.6389	
7363	-118.35	34.15	35.0	2245.0	393.0	783.0	402.0	4.1544	
2823	-117.72	33.63	15.0	1362.0	255.0	378.0	202.0	1.9000	
5667	-118.18	33.85	44.0	1890.0	465.0	1378.0	430.0	3.8819	
14504	-122.14	37.47	36.0	2081.0	412.0	1931.0	373.0	3.7917	
14846	-122.22	39.51	17.0	1201.0	268.0	555.0	277.0	2.1000	
56	-115.52	32.77	18.0	1715.0	337.0	1166.0	333.0	2.2417	
919	-117.08	32.63	30.0	2504.0	559.0	1827.0	490.0	2.6146	

In [ ]:

```
# Menganalisis bilangan baris kosong
data.isna().sum()
```

Out[ ]:

```
longitude             0
latitude              0
housing_median_age    0
total_rooms           0
total_bedrooms        0
population            0
households            0
median_income         0
median_house_value    0
dtype: int64
```

In [ ]:

```
# Lihat perihalan statistik
data.describe()
```

Out[ ]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_i
count	20000.000000	20000.000000	20000.000000	20000.000000	20000.000000	20000.000000	20000.000000	20000.
mean	-119.566172	35.626750	28.627750	2637.051550	537.991800	1425.557650	499.525450	3.
std	2.003609	2.136141	12.582229	2176.314757	420.631119	1131.048487	381.729517	1.
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.
25%	-121.790000	33.930000	18.000000	1451.000000	296.000000	788.000000	280.000000	2.
50%	-118.490000	34.250000	29.000000	2126.000000	434.000000	1166.000000	409.000000	3.
75%	-118.000000	37.710000	37.000000	3149.000000	647.000000	1724.000000	604.000000	4.
max	-114.310000	41.950000	52.000000	37937.000000	6445.000000	35682.000000	6082.000000	15.

In [ ]:

```
# Perhatikan bilangan nilai unik setiap lajur
data.nunique()
```

Out[ ]:

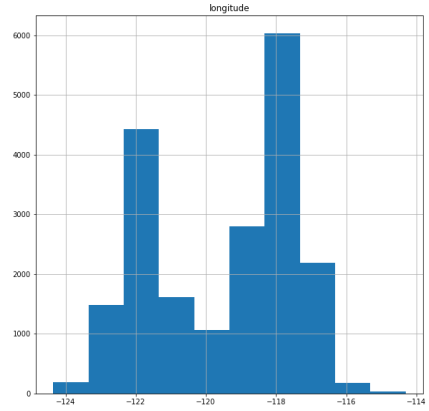
```
longitude                842
latitude                 856
housing_median_age        52
total_rooms             5861
total_bedrooms          1911
population              3857
households              1799
median_income          12632
median_house_value      3824
dtype: int64
```

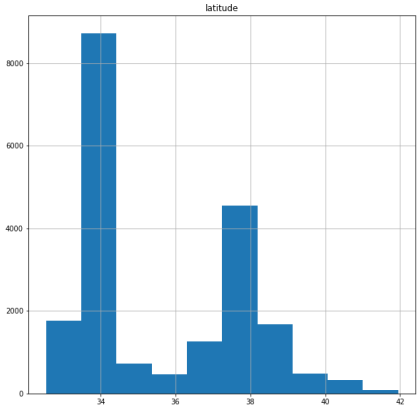
In [ ]:

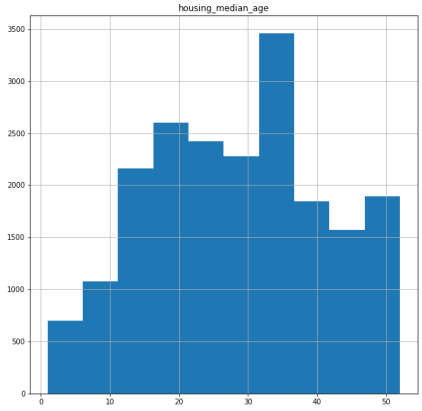
```
# Fasa Data Visualization
import seaborn as sns
import matplotlib.pyplot as plt

plt.rcParams['figure.figsize'] = (12,5)
```

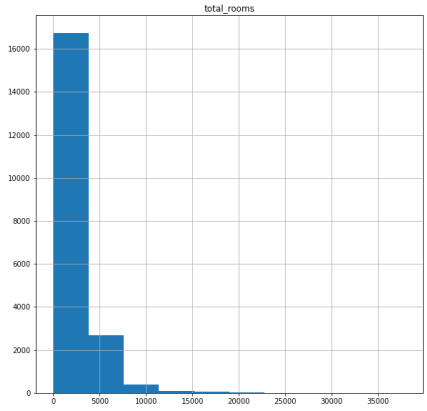
In [ ]:

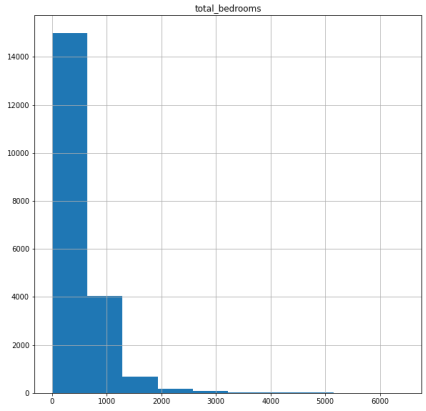
```
# (Data Distribution) Sebaran data adalah fungsi atau senarai yang menunjukkan semua kemu
ngkinan nilai (atau selang) data.
# Ini juga (dan ini penting) memberitahu anda berapa kerap setiap nilai berlaku.
data.hist(figsize = (35,35))
plt.show()
```

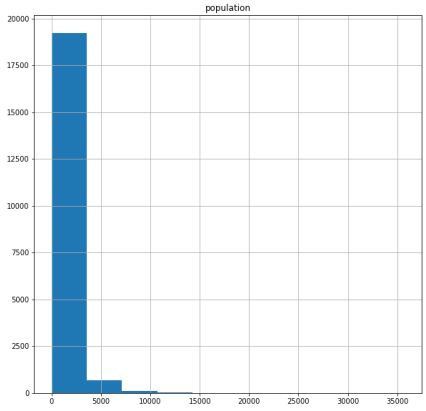


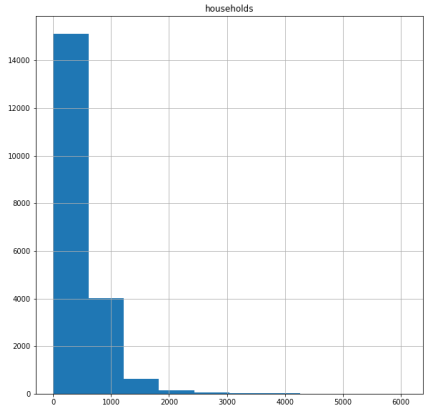


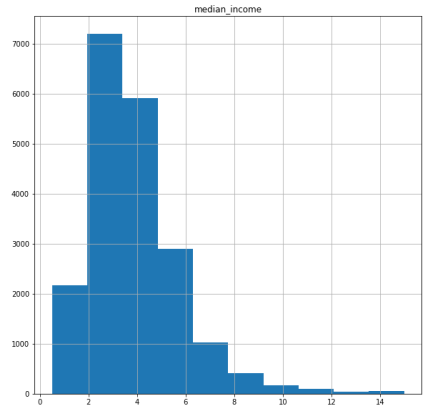


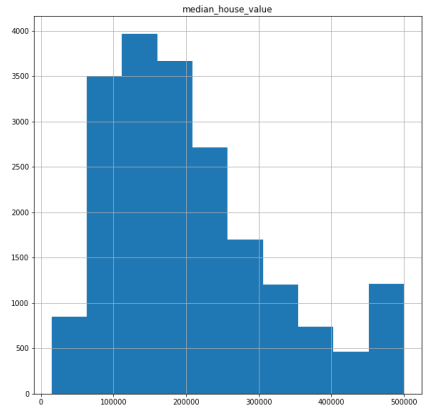












Skewness and Kurtios

Dalam statistik, skewness dan kurtosis adalah dua cara untuk mengukur bentuk taburan.

Skewness adalah ukuran asimetri sebaran. Nilai ini boleh menjadi positif atau negatif.

- **Kecenderungan negatif menunjukkan bahawa ekor berada di sebelah kiri taburan, yang memanjang ke arah nilai yang lebih negatif.**
- **Kecenderungan positif menunjukkan bahawa ekor berada di sebelah kanan taburan, yang memanjang ke arah nilai yang lebih positif.**
- **Nilai sifar menunjukkan bahawa tidak ada kecenderungan dalam pengedaran sama sekali, yang bermaksud pembahagiannya simetri dengan sempurna.**

Kurtosis adalah ukuran sama ada atau tidak pengedaran adalah berat atau ekor ringan berbanding dengan taburan normal.

- **Kurtosis taburan normal adalah 3.**
- **Sekiranya sebaran yang diberikan mempunyai kurtosis kurang dari 3, ia dikatakan playkurtic, yang bermaksud cenderung menghasilkan outliers yang lebih sedikit dan lebih ekstrem daripada taburan normal.**
- **Sekiranya sebaran yang diberikan mempunyai kurtosis lebih besar daripada 3, dikatakan leptokurtik, yang bermaksud cenderung menghasilkan lebih banyak penyimpangan daripada taburan normal.**

In [ ]:

```
from scipy.stats import skew, kurtosis

for col in data.columns:
  col_skew = skew(data[col].values, bias=False)
  col_kur = kurtosis(data[col].values, bias=False)
  print("Column {} : Skewness is {} and Kurtosis is {} \n".format(col, col_skew, col_kur)
)
```

```
Column longitude : Skewness is -0.30300836337488696 and Kurtosis is -1.3282411448196452

Column latitude : Skewness is 0.46997169442081455 and Kurtosis is -1.1140669613232959
```

```
Column housing_median_age : Skewness is 0.05793693694852265 and Kurtosis is -0.8017565137
14464

Column total_rooms : Skewness is 4.026410281276376 and Kurtosis is 29.879537902875825

Column total_bedrooms : Skewness is 3.4006385959718552 and Kurtosis is 20.945522400474776


Column population : Skewness is 4.944664922860473 and Kurtosis is 74.65497884969864

Column households : Skewness is 3.373107927134306 and Kurtosis is 21.41308046273746

Column median_income : Skewness is 1.6371835680973563 and Kurtosis is 4.882349071537418

Column median_house_value : Skewness is 0.9756130631038522 and Kurtosis is 0.317585376485
4204
```

In [ ]:

```
data_log = data.copy()
for col in data_log.columns:
  if col != 'longitude' and col != 'latitude' and col != 'median_house_value':
    data_log[col] = np.log(data_log[col].values) # Tukar semua lajur ke log

data_log.describe()
```

Out[ ]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_i
count	20000.000000	20000.000000	20000.000000	20000.000000	20000.000000	20000.000000	20000.000000	20000.
mean	-119.566172	35.626750	3.224614	7.629768	6.051902	7.024720	5.981448	1.
std	2.003609	2.136141	0.567976	0.749375	0.730655	0.737381	0.731154	0.
min	-124.350000	32.540000	0.000000	0.693147	0.000000	1.098612	0.000000	-0.
25%	-121.790000	33.930000	2.890372	7.280008	5.690359	6.669498	5.634790	0.
50%	-118.490000	34.250000	3.367296	7.661998	6.073045	7.061334	6.013715	1.
75%	-118.000000	37.710000	3.610918	8.054840	6.472346	7.452402	6.403574	1.
max	-114.310000	41.950000	3.951244	10.543682	8.771060	10.482402	8.713089	2.

In [ ]:

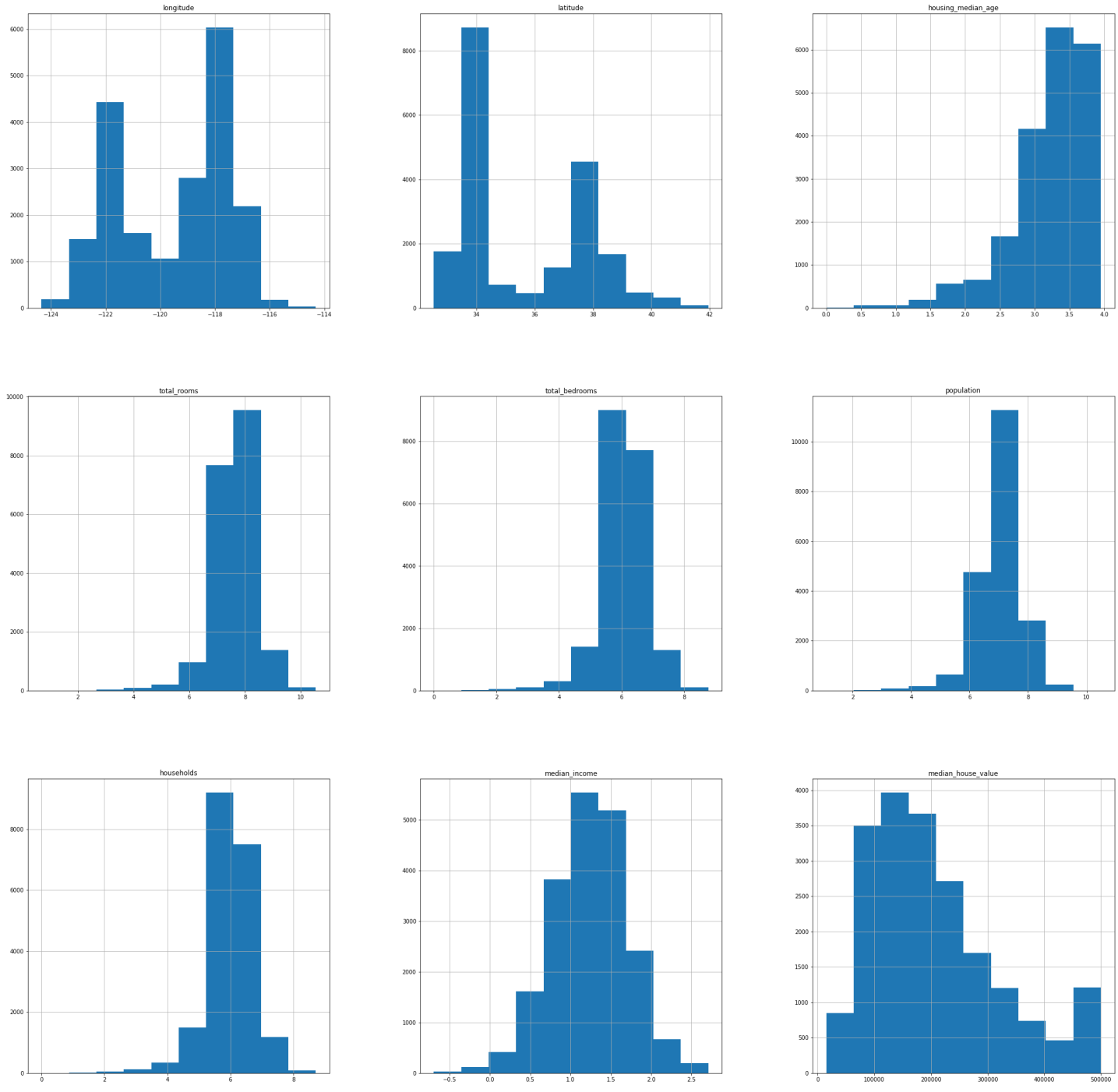
```
for col in data_log.columns:
  col_skew = skew(data_log[col].values, bias=False)
  col_kur = kurtosis(data_log[col].values, bias=False)
  print("Column {} : Skewness is {} and Kurtosis is {} \n".format(col, col_skew, col_kur)
)
```

```
Column longitude : Skewness is -0.30300836337488696 and Kurtosis is -1.3282411448196452

Column latitude : Skewness is 0.46997169442081455 and Kurtosis is -1.1140669613232959

Column housing_median_age : Skewness is -1.2866424476088916 and Kurtosis is 2.10313113045
80633

Column total_rooms : Skewness is -1.0816558991838037 and Kurtosis is 5.221676456379729

Column total_bedrooms : Skewness is -1.048313646985874 and Kurtosis is 5.052544712979573

Column population : Skewness is -1.0700100350218145 and Kurtosis is 4.707330361970095

Column households : Skewness is -1.113491669096607 and Kurtosis is 4.965678520650513

Column median_income : Skewness is -0.14777718989591349 and Kurtosis is 0.389690325520090
3
```

```
Column median_house_value : Skewness is 0.9756130631038522 and Kurtosis is 0.3175853764854204
```

In [ ]:

```
data_log.hist(figsize = (35,35))
plt.show()
```

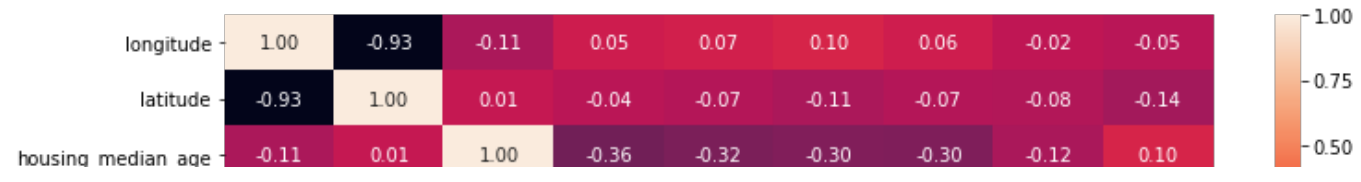


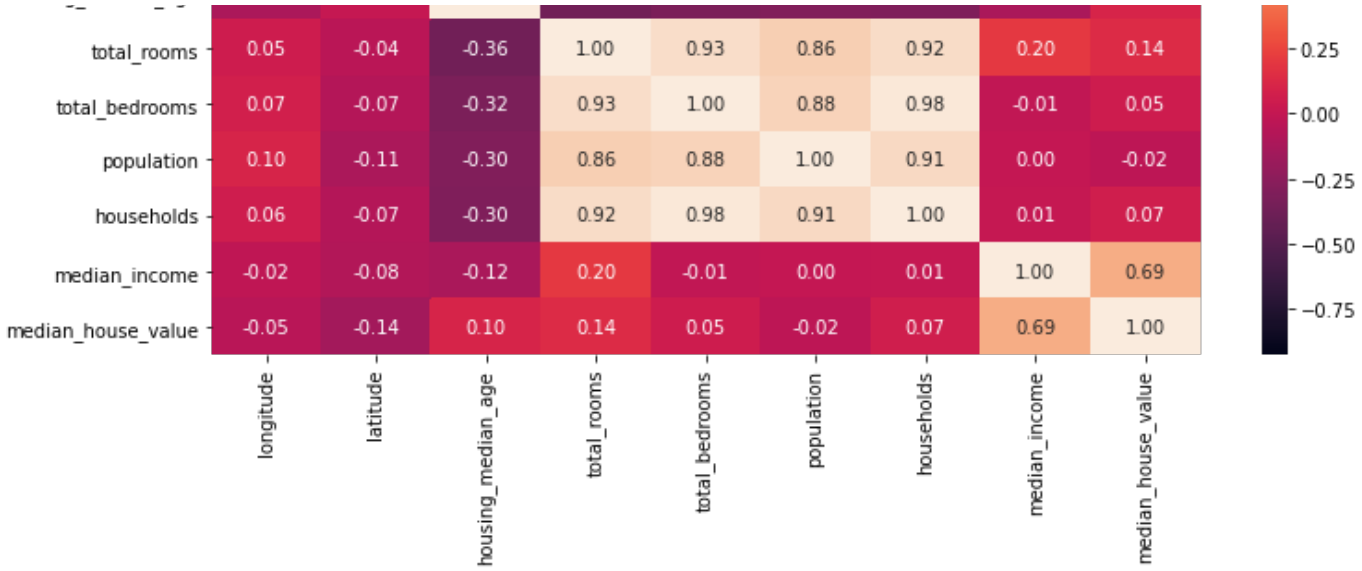
In [ ]:

```
sns.heatmap(data.corr(), annot=True, fmt='.2f')
# Matriks korelasi adalah jadual yang menunjukkan pekali korelasi antara pemboleh ubah.
# Setiap sel dalam jadual menunjukkan korelasi antara dua pemboleh ubah.
# Matriks korelasi digunakan untuk meringkaskan data, sebagai masukan ke analisis yang lebih maju, dan sebagai diagnostik untuk analisis lanjutan.
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f90d8e85dd0>
```



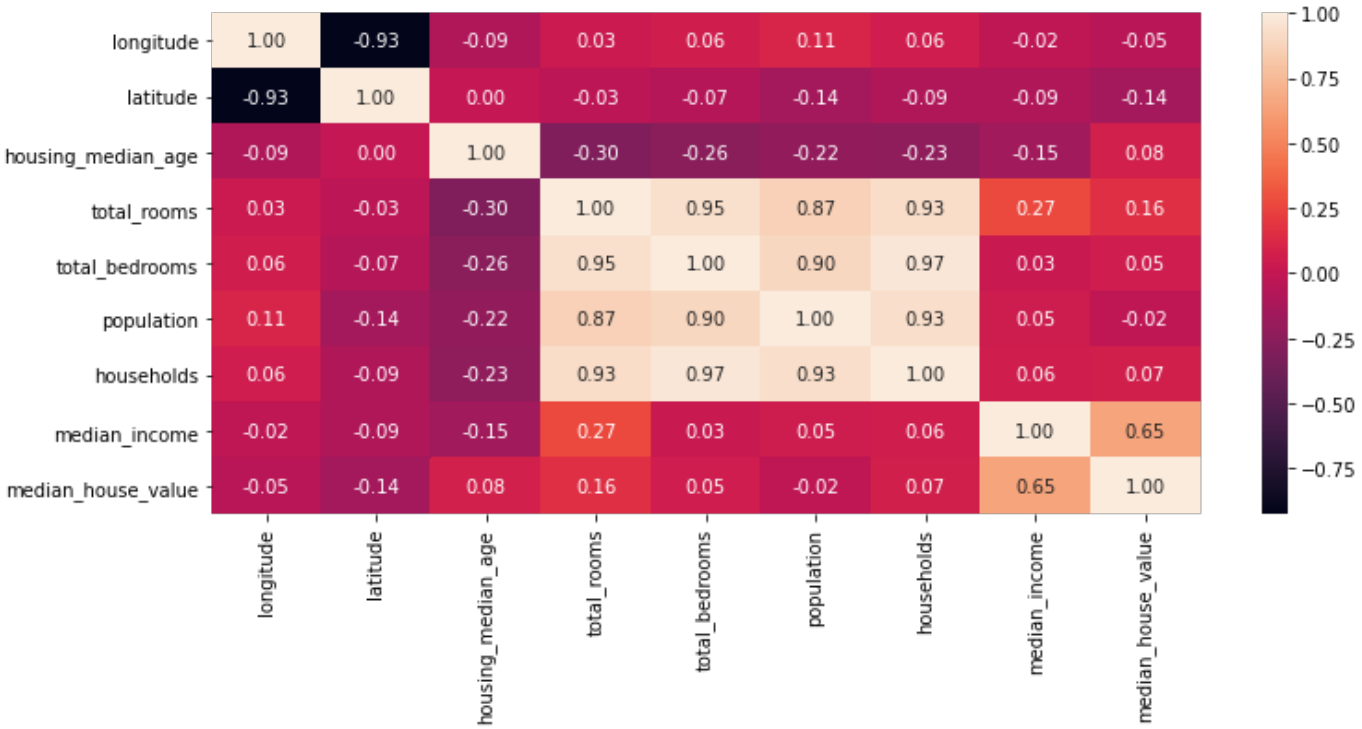


In [ ]:

```
sns.heatmap(data_log.corr(), annot=True, fmt='.2f')
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f90cec3cf90>
```

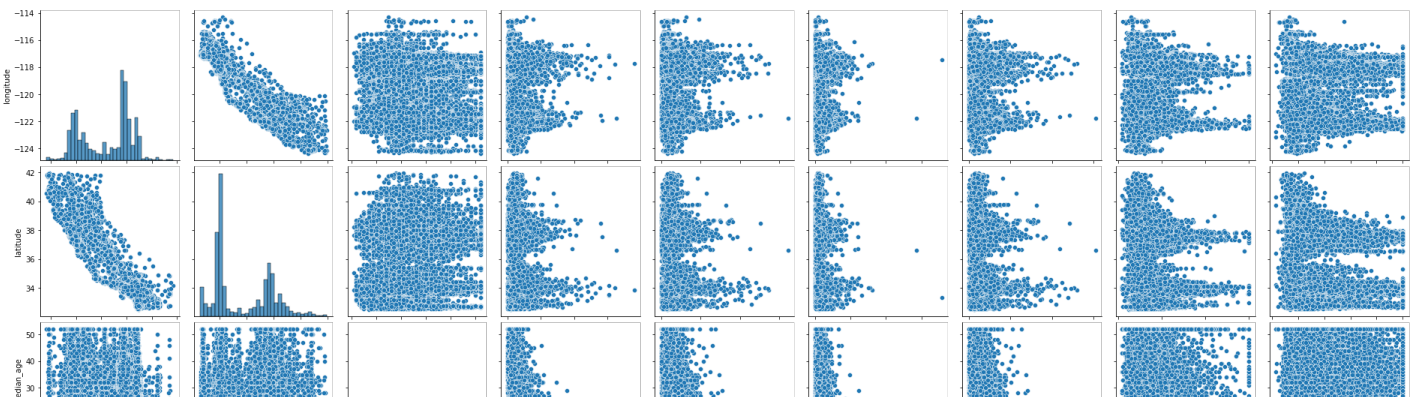


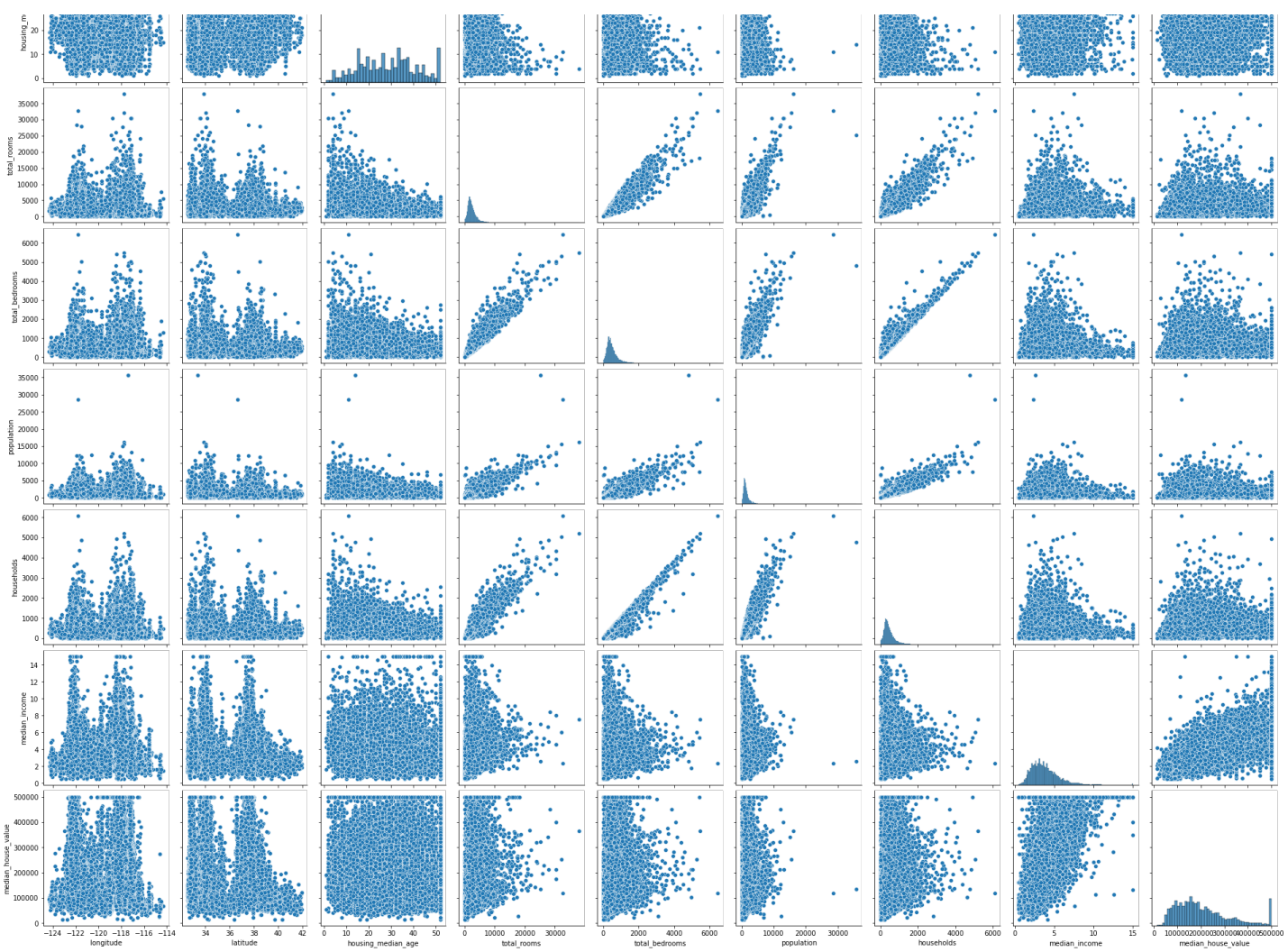
In [ ]:

```
sns.pairplot(data, height=3)
```

Out[ ]:

```
<seaborn.axisgrid.PairGrid at 0x7f90ceb31290>
```



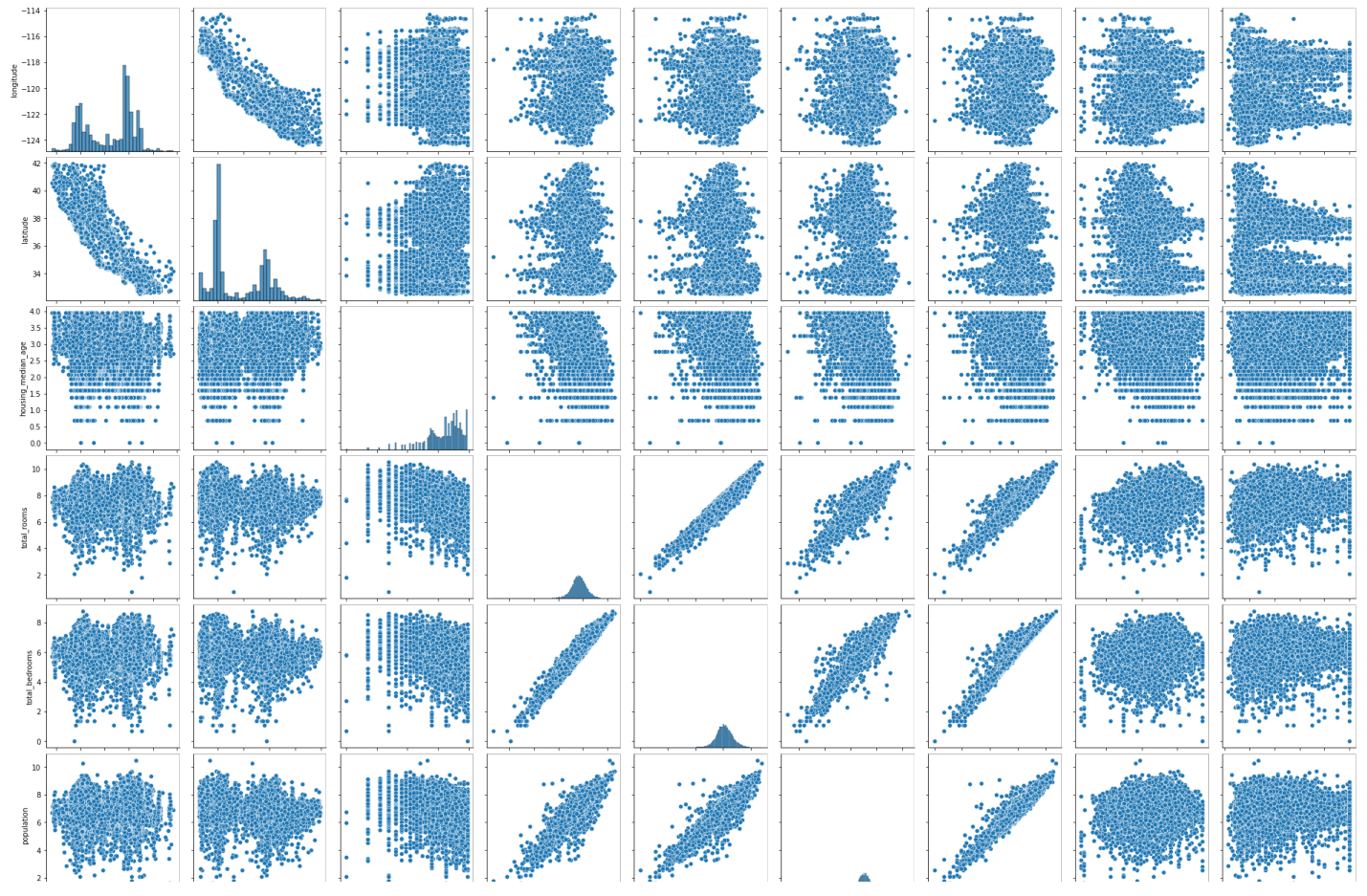


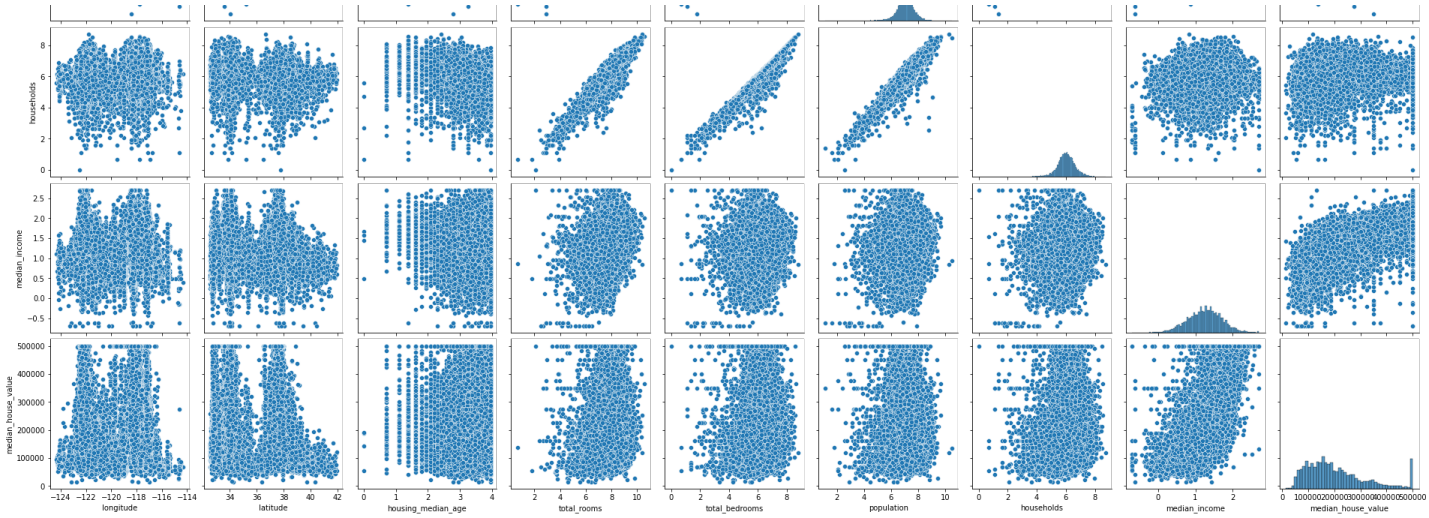
In [ ]:

```
sns.pairplot(data_log, height=3)
```

Out[ ]:

<seaborn.axisgrid.PairGrid at 0x7f90cbb16390>





Ordinary Least Squares Regression

Ordinary Least Squares Regression (OLS) adalah kaedah analisis statistik yang menganggarkan hubungan antara satu atau lebih pemboleh ubah bebas dan pemboleh ubah bersandar; kaedah mengira hubungan dengan meminimumkan jumlah petak dalam perbezaan antara nilai yang diperhatikan dan diramalkan dari pemboleh ubah bersandar yang dikonfigurasikan sebagai garis lurus.

`statsmodels` adalah modul Python yang menyediakan kelas dan fungsi untuk perkiraan banyak model statistik yang berbeza, serta untuk menjalankan ujian statistik, dan penerokaan data statistik. Senarai statistik hasil yang luas tersedia untuk setiap penganggar.

In [ ]:

```
import statsmodels.formula.api as smf
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: p
andas.util.testing is deprecated. Use the functions in the public API at pandas.testing i
nstead.
  import pandas.util.testing as tm
```

In [ ]:

```
reg1 = smf.ols(formula = 'median_house_value ~ median_income', data=data_log).fit()
print(reg1.summary())

# Signifincance level
# 0.05 = 5% (default)
# 0.01 = 1% (extreme lower)
# 0.1 = 10% (extreme higher)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     median_house_value   R-squared:                       0.426
Model:                            OLS   Adj. R-squared:                  0.426
Method:                 Least Squares   F-statistic:                 1.484e+04
Date:                Wed, 23 Jun 2021   Prob (F-statistic):               0.00
Time:                        07:16:56   Log-Likelihood:            -2.5598e+05
No. Observations:               20000   AIC:                         5.120e+05
Df Residuals:                   19998   BIC:                         5.120e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      7745.1829   1749.485      4.427      0.000     4316.048     1.12e+04
median_income  1.601e+05   1314.561    121.823      0.000     1.58e+05     1.63e+05
==============================================================================
Omnibus:                     3494.030   Durbin-Watson:                   1.016
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             6543.462
Skew:                           1.091   Prob(JB):                         0.00
Kurtosis:                       4.757   Cond. No.                         5.71
==============================================================================
```

```
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifie
d.
```

In [ ]:

```
reg2 = smf.ols(formula = 'median_house_value ~ median_income+ total_rooms+ population+ ho
useholds+ total_bedrooms', data=data_log).fit()
print(reg2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     median_house_value   R-squared:                       0.511
Model:                            OLS   Adj. R-squared:                  0.511
Method:                 Least Squares   F-statistic:                     4185.
Date:                Wed, 23 Jun 2021   Prob (F-statistic):               0.00
Time:                        07:16:56   Log-Likelihood:            -2.5437e+05
No. Observations:               20000   AIC:                         5.087e+05
Df Residuals:                   19994   BIC:                         5.088e+05
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept        2.78e+05   7276.972     38.206      0.000    2.64e+05    2.92e+05
median_income    2.045e+05   1796.048    113.839      0.000    2.01e+05    2.08e+05
total_rooms     -1.26e+05   3461.253    -36.407      0.000   -1.33e+05   -1.19e+05
population     -9.183e+04   2217.359    -41.413      0.000   -9.62e+04   -8.75e+04
households       1.086e+05   4336.706     25.041      0.000       1e+05    1.17e+05
total_bedrooms   1.043e+05   4802.166     21.730      0.000    9.49e+04    1.14e+05
==============================================================================
Omnibus:                     3259.776   Durbin-Watson:                   1.261
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7535.873
Skew:                           0.940   Prob(JB):                         0.00
Kurtosis:                       5.347   Cond. No.                         188.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifie
d.
```

In [ ]:

```
reg2 = smf.ols(formula = 'median_house_value ~ median_income+ total_rooms+ population+ ho
useholds+ total_bedrooms', data=data).fit()
print(reg2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     median_house_value   R-squared:                       0.532
Model:                            OLS   Adj. R-squared:                  0.531
Method:                 Least Squares   F-statistic:                     4538.
Date:                Wed, 23 Jun 2021   Prob (F-statistic):               0.00
Time:                        07:16:56   Log-Likelihood:            -2.5394e+05
No. Observations:               20000   AIC:                         5.079e+05
Df Residuals:                   19994   BIC:                         5.079e+05
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept        1.94e+04   1675.649     11.575      0.000    1.61e+04    2.27e+04
median_income    4.723e+04    362.581    130.266      0.000    4.65e+04    4.79e+04
total_rooms       -24.1866      0.879    -27.519      0.000     -25.909     -22.464
population        -36.9514      1.235    -29.916      0.000     -39.372     -34.530
households        154.2211      8.439     18.274      0.000     137.680     170.763
total_bedrooms     82.1918      7.861     10.456      0.000      66.784      97.600
==============================================================================
Omnibus:                     3961.290   Durbin-Watson:                   1.209
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            11058.245
Skew:                           1.054   Prob(JB):                         0.00
```

```
Skew:                          1.054   Prob(JB):                     0.00
Kurtosis:                      5.970   Cond. No.                  1.20e+04
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifie
d.
[2] The condition number is large, 1.2e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [ ]: