Advanced Data Analysis

DATA 71200

Class 3

Schedule

2-Jun	**Inspecting Data**
6-Jun	Representing Data
7-Jun	Async: DataCamp
8-Jun	Evaluation Methods
9-Jun	Async: DataCamp

Terminology Review

- **labeled training set**
  - “a training set that contains the desired solution (a.k.a. a label) for each instance.”
- **regularization**
  - “constraining a model to make it simpler and reduce the risk of overfitting”
- **hyperparameter**
  - “amount of regularization to apply during learning”

Terminology Review

- **Training set**
  - data used to train the model
- **Testing set**
  - hold out data used to estimate the generalization error on new data
- **Validation set**
  - used to compare models
- **Cross-validation**
  - iteratively holding out a subset of the data and testing on the rest (typically 80/20)

More Terminology

- **Class**
  - "One of a set of enumerated target values for a label."
- **Classification**
  - "A type of machine learning model for distinguishing among two or more discrete classes."

https://developers.google.com/machine-learning/glossary

More Terminology

- **Samples**
  - Individual items
  - **Label**
    - “In supervised learning, the "answer" or "result" portion of an example”
  - **Feature**
    - “An input variable used in making predictions.”

https://developers.google.com/machine-learning/glossary

Machine Learning Pipeline

- **“However simple or complex the Machine Learning problem at hand may be, it will always contain the following steps:**
  - Data loading, preparation and splitting into the train and test partitions
  - Model selection and training ("fitting")
  - Model performance assessment”

“AI Fundamentals” course, DataCamp

Frame the Problem

- **“what exactly is the business (research) objective”**
  - “how does the company (researcher) expect to use and benefit from this model?”
  - will determine
    - “how you frame the problem”
    - “what algorithms you will select”
    - “what performance measure you will use to evaluate your model”
    - "how much effort you should spend tweaking it”

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O’Reilly Media, Inc.

Pipeline

- **“sequence of data processing components is called a data pipeline”**
  - “components typically run asynchronously” and “is fairly self-contained”
  - each component
    - “pulls in a large amount of data”
    - “processes it"
    - “spits out the result in another data store”
    - a later “component in the pipeline pulls this data and spits out its own output”

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O’Reilly Media, Inc.

Inspecting Data to Gain Insights

- **Data size and type**
- **Summary statistics**
- **Histograms**
- **Visualizing Geographic Data**

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```
In [6]: housing.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
longitude             20640 non-null float64
latitude              20640 non-null float64
housing_median_age    20640 non-null float64
total_rooms           20640 non-null float64
total_bedrooms        20433 non-null float64
population            20640 non-null float64
households            20640 non-null float64
median_income         20640 non-null float64
median_house_value    20640 non-null float64
ocean_proximity       20640 non-null object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

Figure 2-6. Housing info

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```
In [8]: housing.describe()
```

Out[8]:

	longitude	latitude	housing_median_age	total_rooms	total_bedr
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000
mean	-119.569704	35.631861	28.639486	2635.763081	537.87055
std	2.003532	2.135952	12.585558	2181.615252	421.38507
min	-124.350000	32.540000	1.000000	2.000000	1.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.00000
50%	-118.490000	34.260000	29.000000	2127.000000	435.00000
75%	-118.010000	37.710000	37.000000	3148.000000	647.00000
max	-114.310000	41.950000	52.000000	39320.000000	6445.0000

Figure 2-7. Summary of each numerical attribute

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```
%matplotlib inline   # only in a Jupyter notebook
import matplotlib.pyplot as plt
housing.hist(bins=50, figsize=(20,15))
plt.show()
```

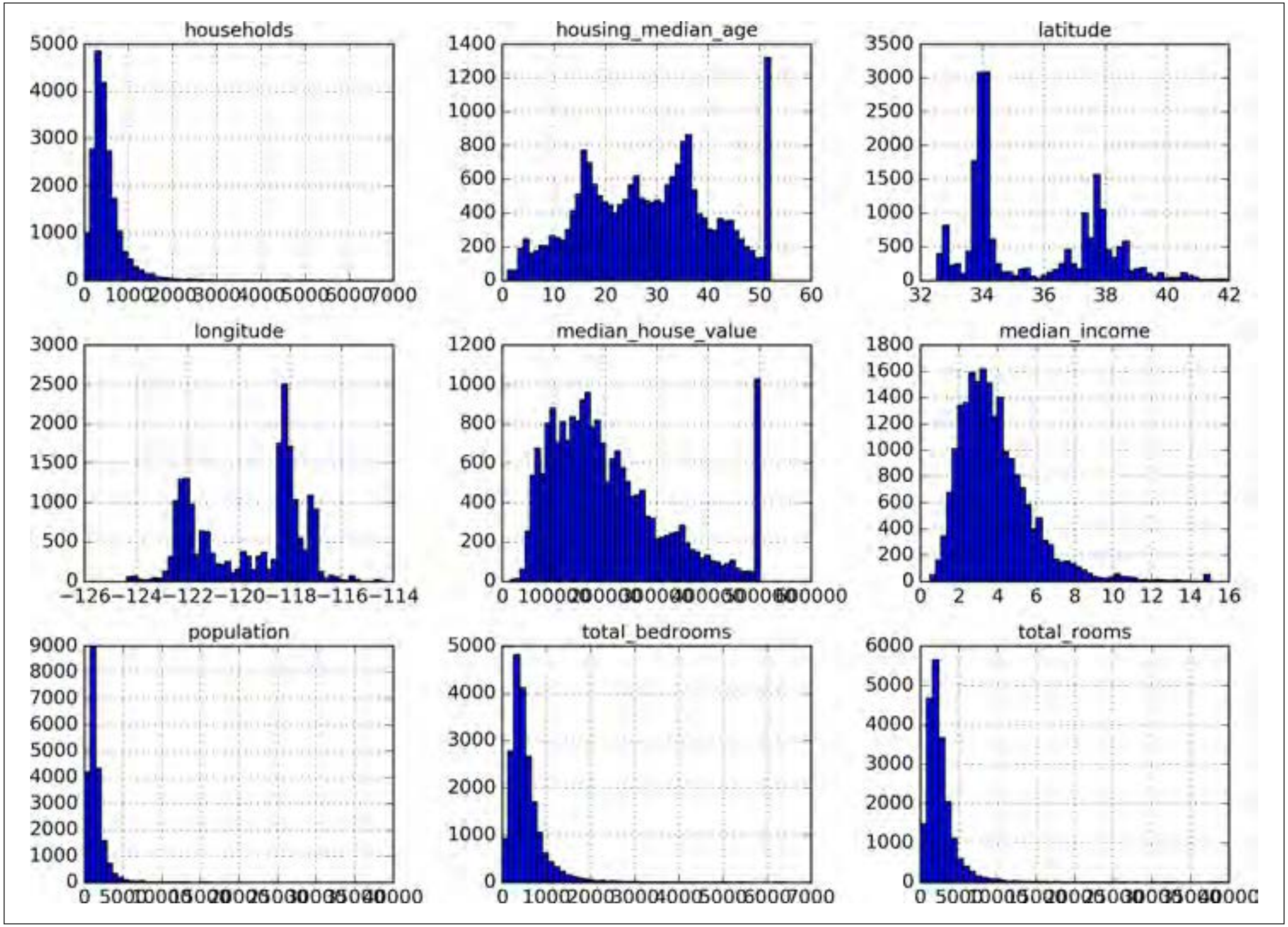


Figure 2-8. A histogram for each numerical attribute

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```python
%matplotlib inline   # only in a Jupyter notebook
import matplotlib.pyplot as plt
housing.hist(bins=50, figsize=(20,15))
plt.show()
```

First, the median income attribute does not look like it is expressed in US dollars (USD). After checking with the team that collected the data, you are told that the data has been scaled and capped at 15 (actually 15.0001) for higher median incomes, and at 0.5 (actually 0.4999) for lower median incomes. Working with preprocessed attributes is common in Machine Learning, and it is not necessarily a problem, but you should try to understand how the data was computed.

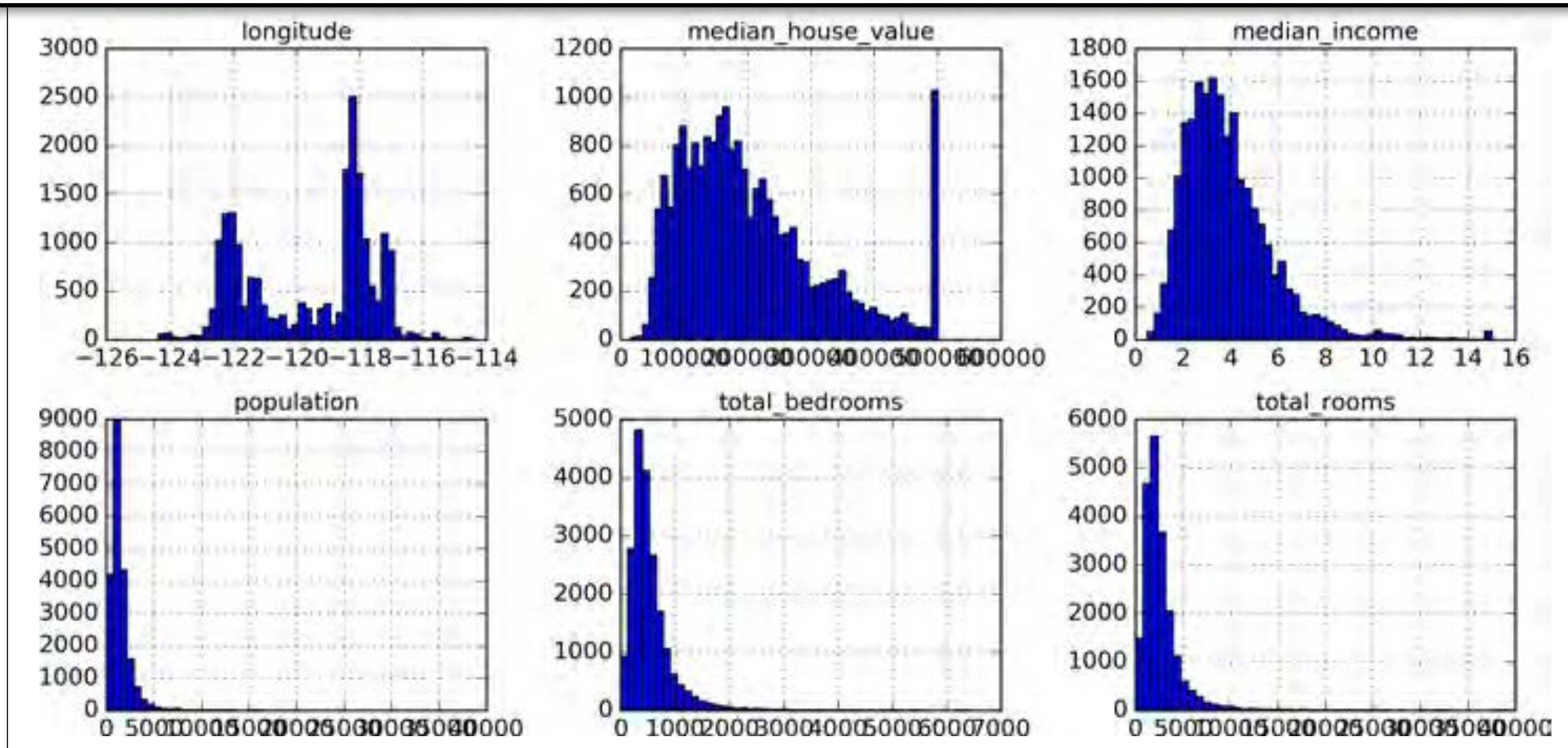
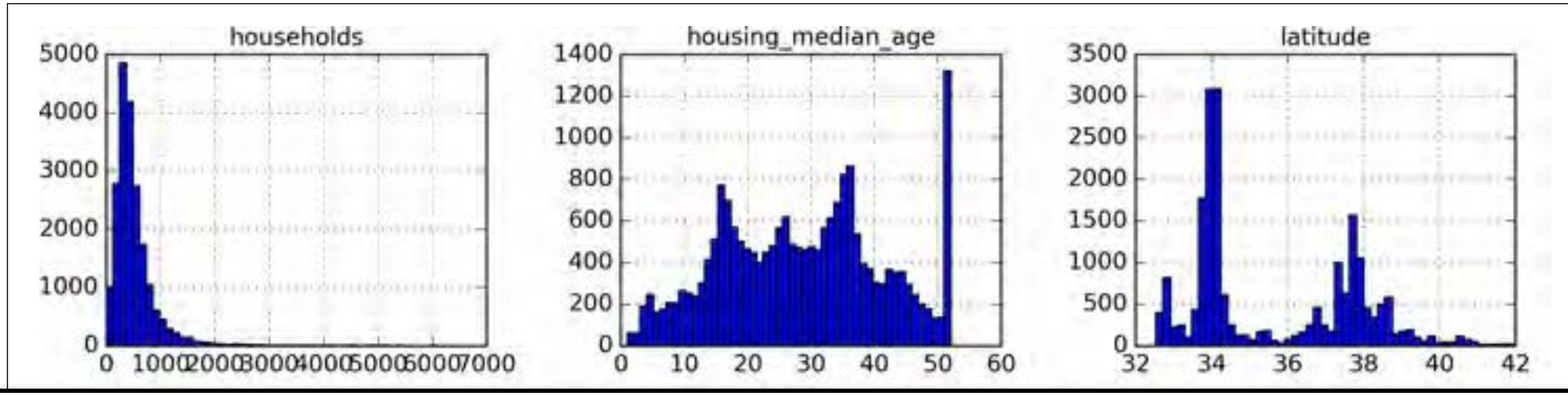


Figure 2-8. A histogram for each numerical attribute

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```python
%matplotlib inline   # only in a Jupyter notebook
import matplotlib.pyplot as plt
housing.hist(bins=50, figsize=(20,15))
plt.show()
```



The housing median age and the median house value were also capped. The latter may be a serious problem since it is your target attribute (your labels). Your Machine Learning algorithms may learn that prices never go beyond that limit. You need to check with your client team (the team that will use your system's output) to see if this is a problem or not. If they tell you that they need precise predictions even beyond $500,000, then you have mainly two options:

a. Collect proper labels for the districts whose labels were capped.

b. Remove those districts from the training set (and also from the test set, since your system should not be evaluated poorly if it predicts values beyond $500,000).

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

Finally, many histograms are *tail heavy*: they extend much farther to the right of the median than to the left. This may make it a bit harder for some Machine Learning algorithms to detect patterns. We will try transforming these attributes later on to have more bell-shaped distributions.

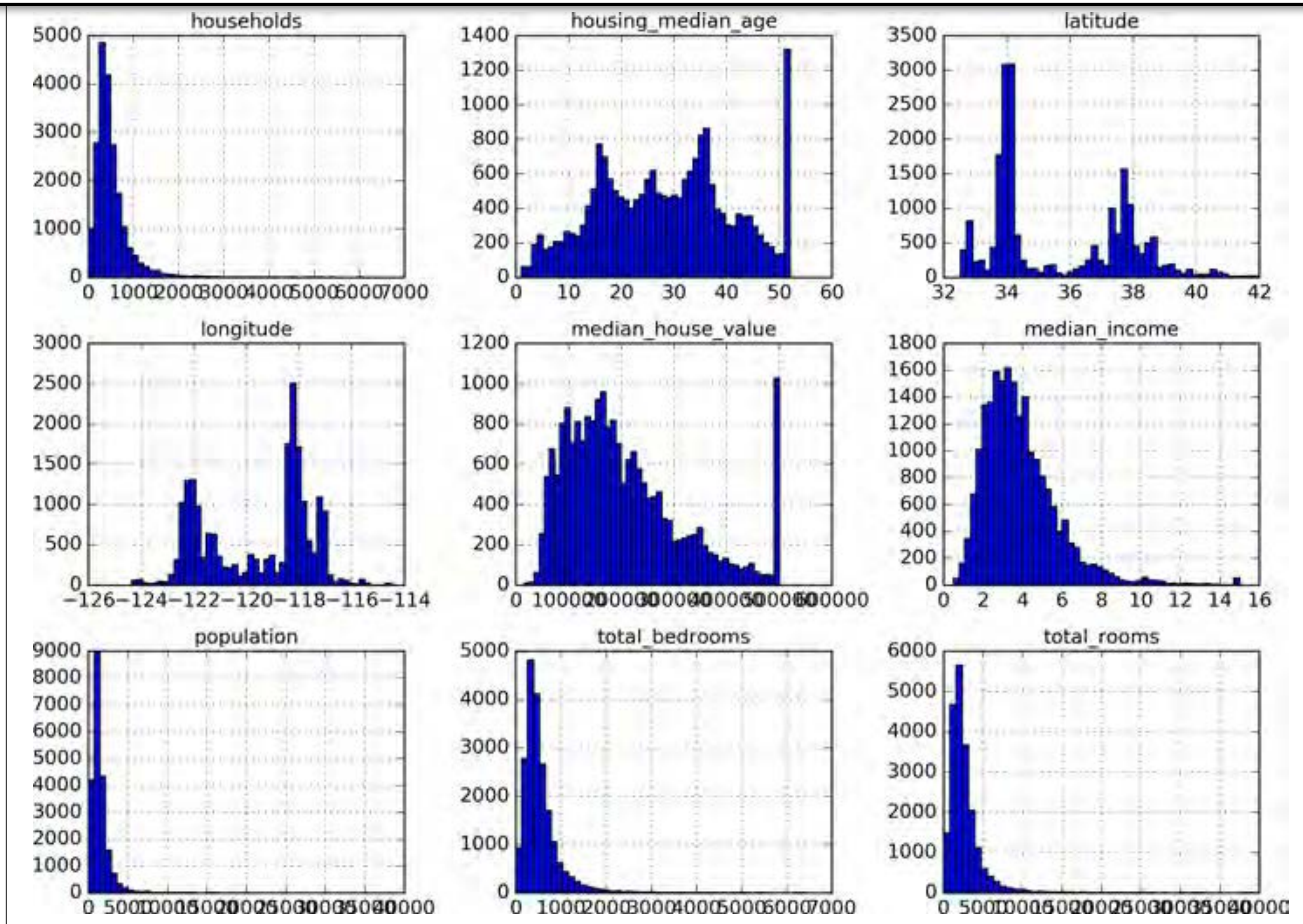


Figure 2-8. A histogram for each numerical attribute

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```
housing.plot(kind="scatter", x="longitude", y="latitude")
```

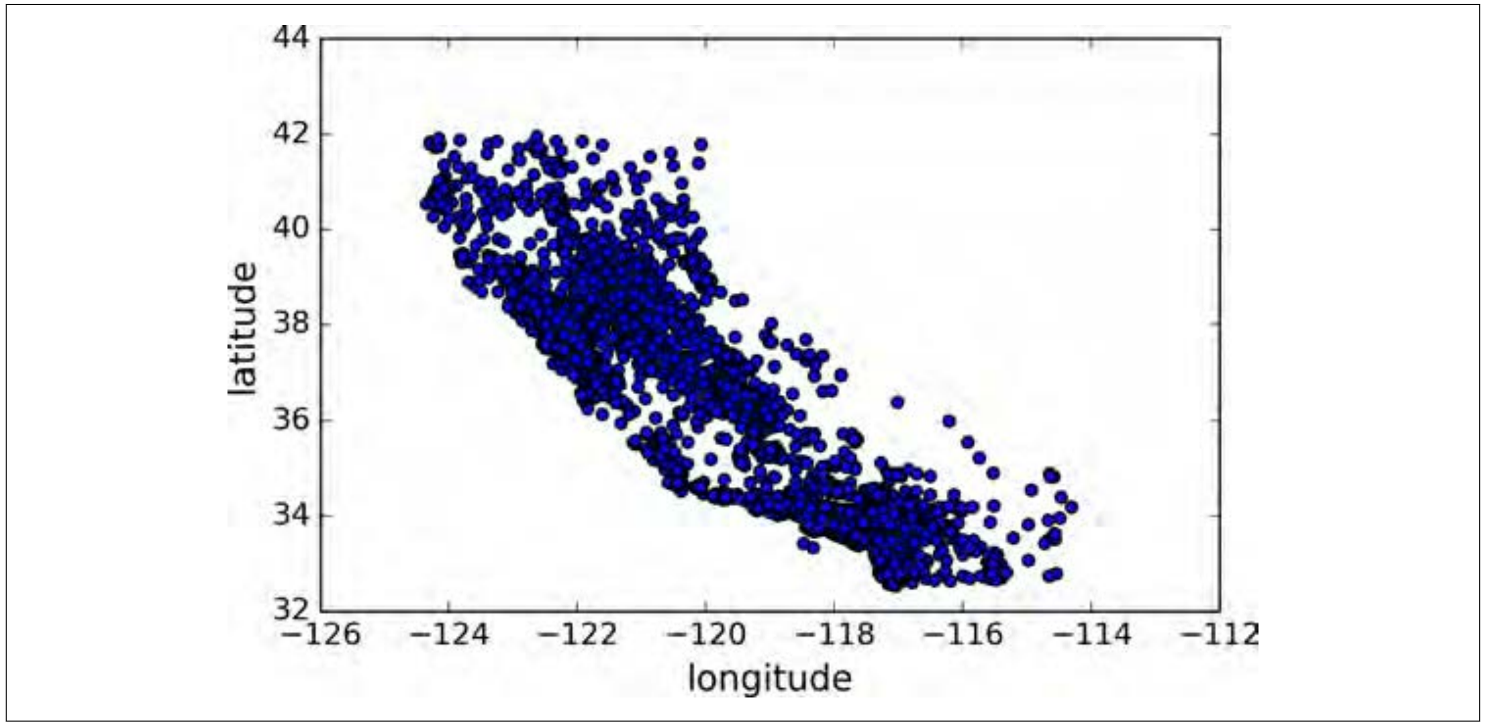


Figure 2-11. A geographical scatterplot of the data

```
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.1)
```

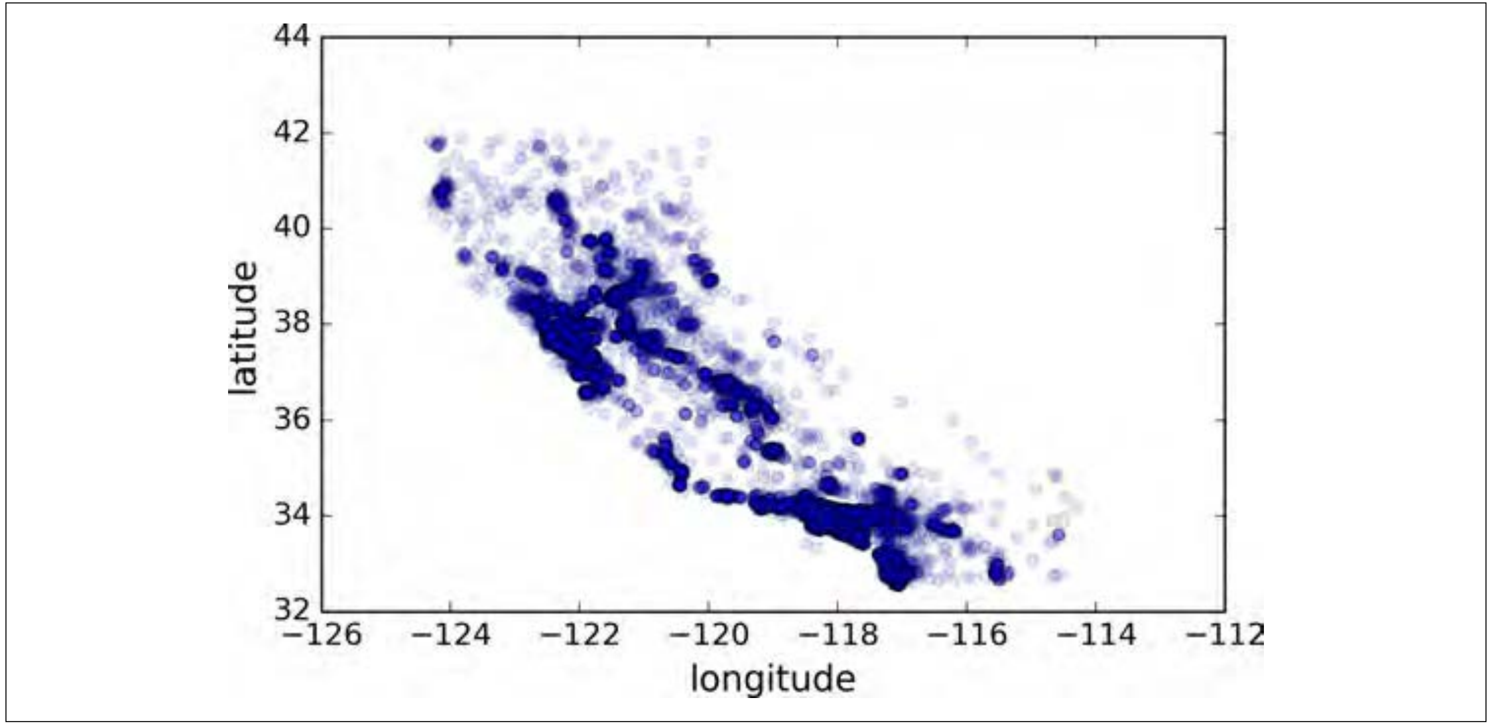


Figure 2-12. A better visualization highlighting high-density areas

Setting alpha to 0.1 emphasizes high density areas

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```python
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.4,
    s=housing["population"]/100, label="population",
    c="median_house_value", cmap=plt.get_cmap("jet"), colorbar=True,
)
plt.legend()
```

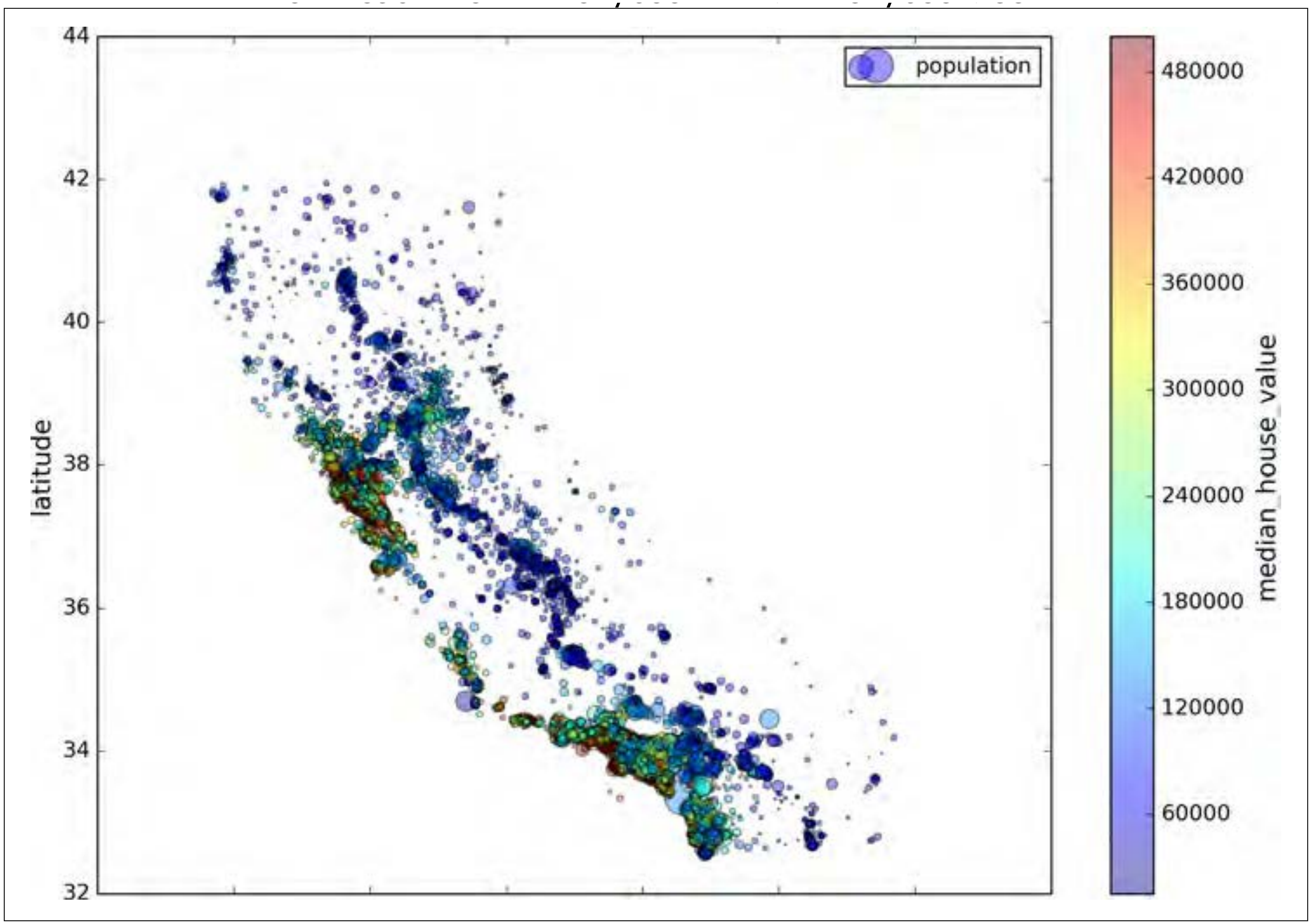


Figure 2-13. California housing prices

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

Look for Correlations

```python
corr_matrix = housing.corr()
```

```
>>> corr_matrix["median_house_value"].sort_values(ascending=False)
median_house_value    1.000000
median_income         0.687170
total_rooms           0.135231
housing_median_age    0.114220
households            0.064702
total_bedrooms        0.047865
population           -0.026699
longitude            -0.047279
latitude             -0.142826
Name: median_house_value, dtype: float64
```

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

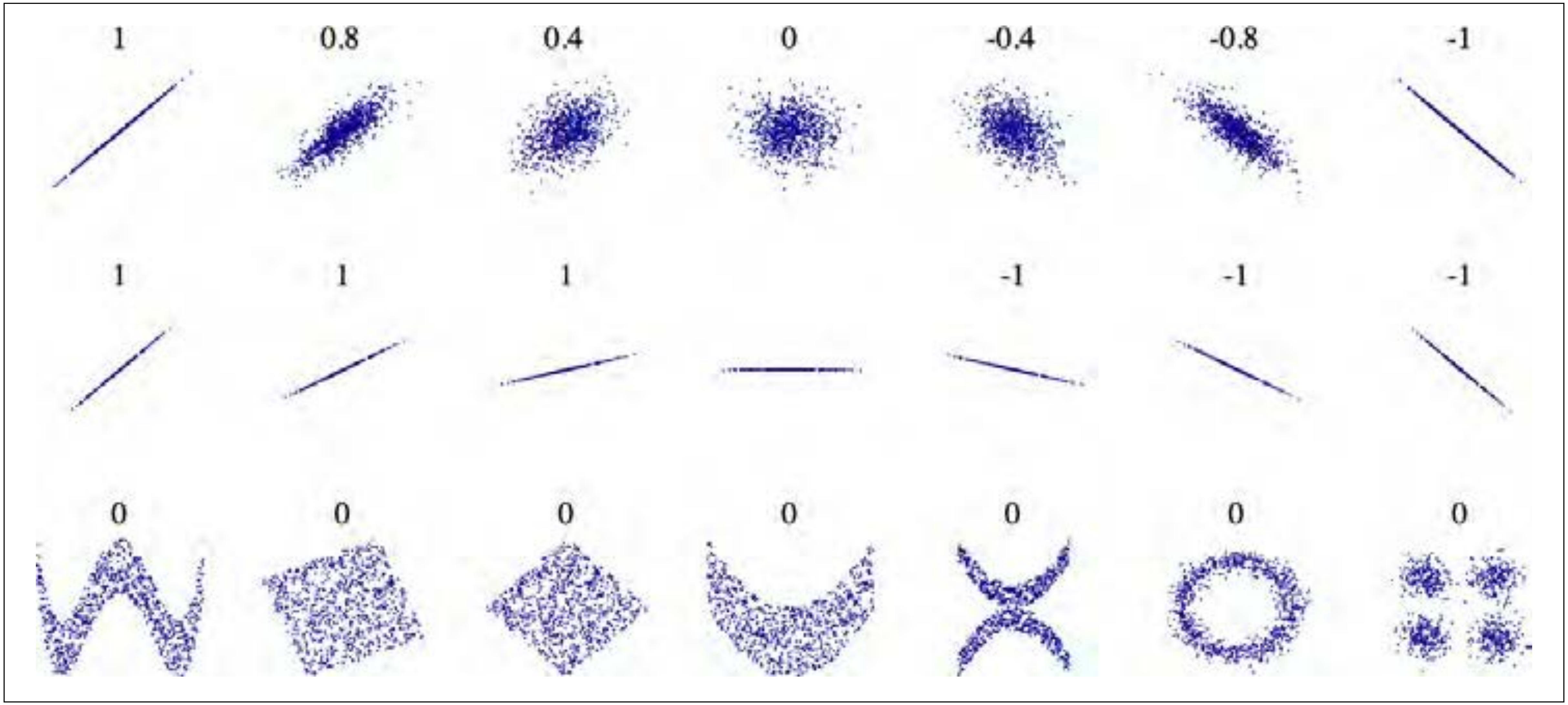


Figure 2-14. Standard correlation coefficient of various datasets (source: Wikipedia; public domain image)

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```python
from pandas.plotting import scatter_matrix

attributes = ["median_house_value", "median_income", "total_rooms",
              "housing_median_age"]
scatter_matrix(housing[attributes], figsize=(12, 8))
```

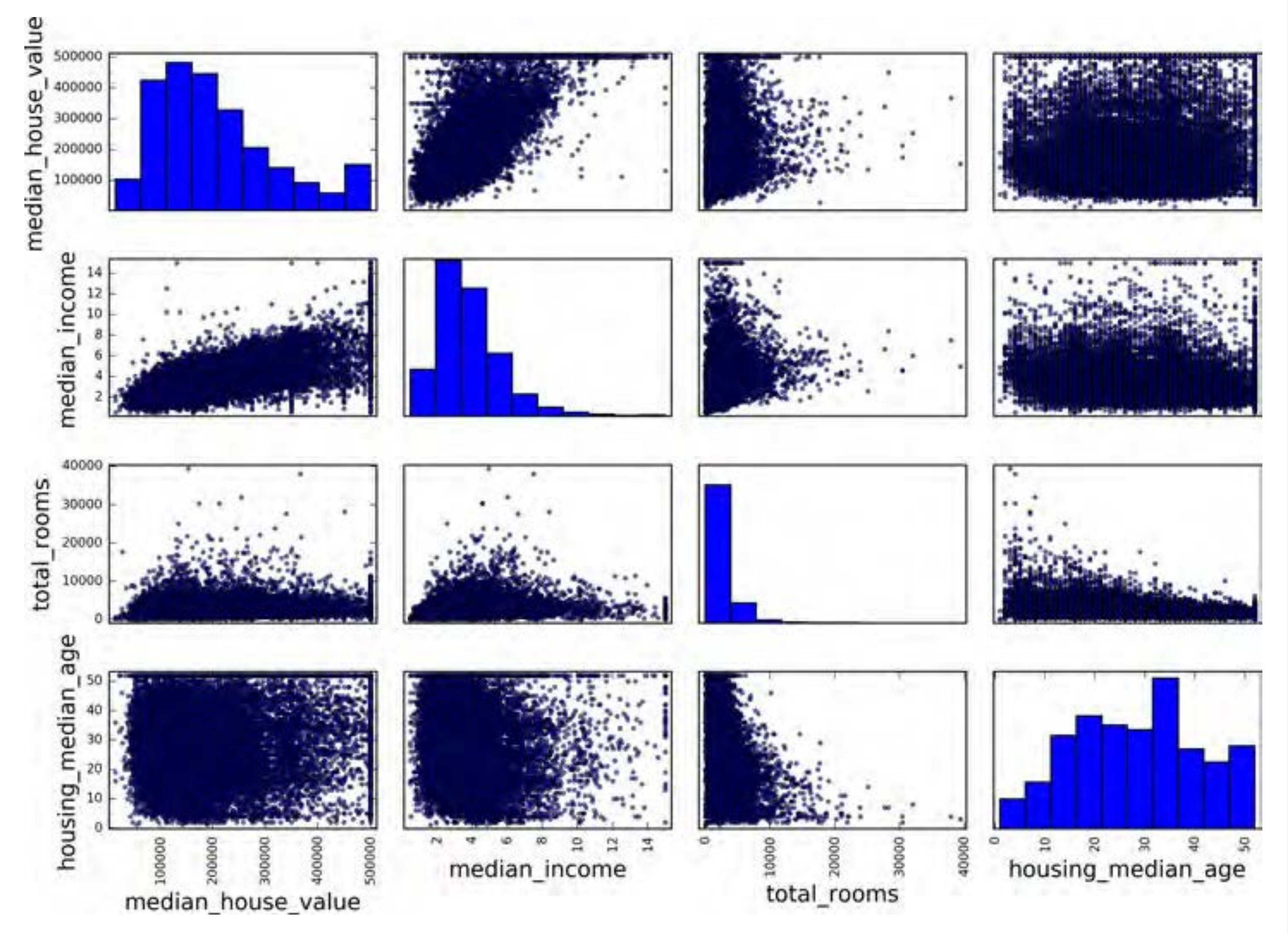


Figure 2-15. Scatter matrix

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```python
from pandas.plotting import scatter_matrix

attributes = ["median_house_value", "median_income", "total_rooms",
              "housing_median_age"]
scatter_matrix(housing[attributes], figsize=(12, 8))
```

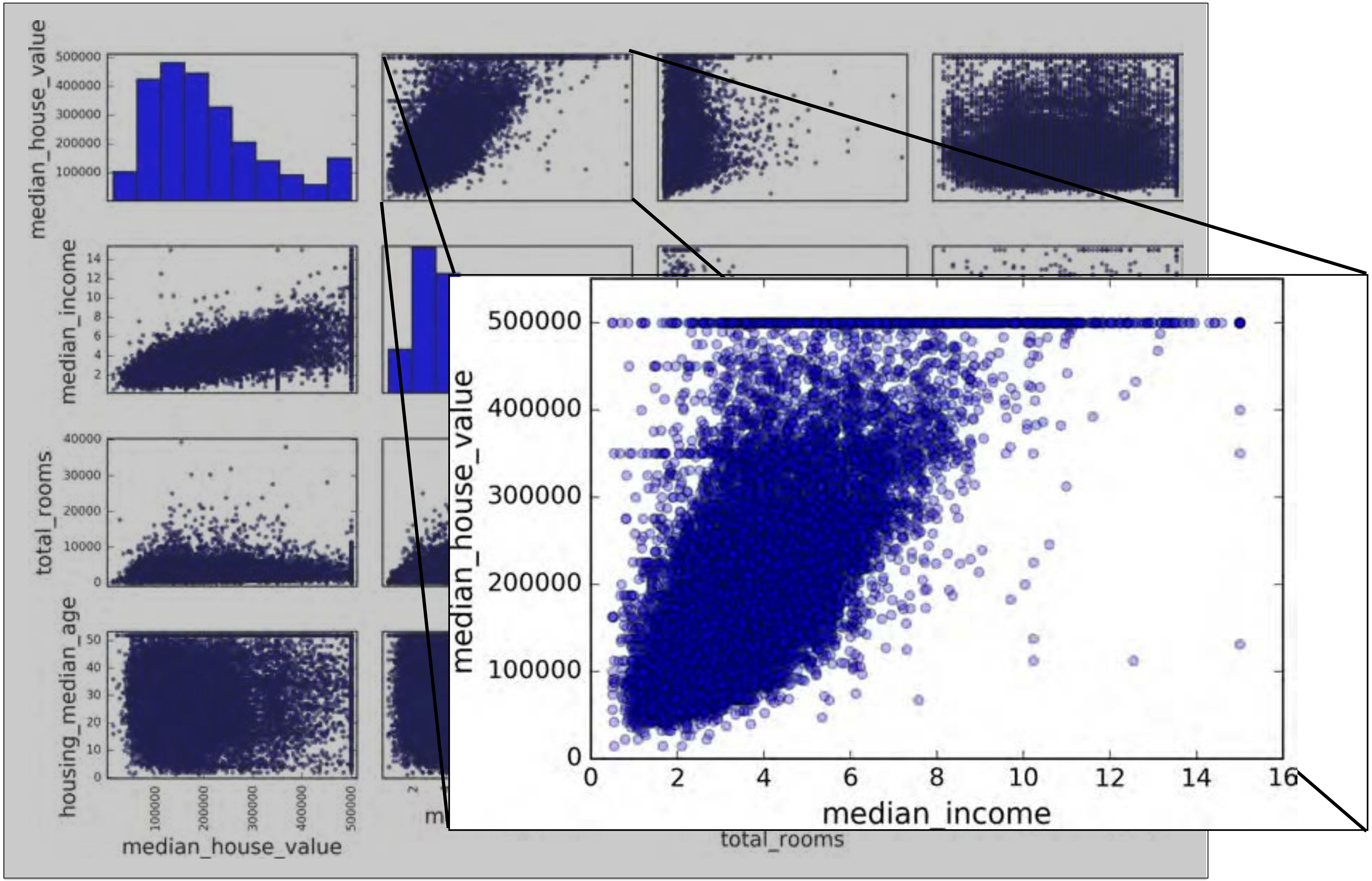


Figure 2-15. Scatter matrix

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

In-Class Activity

- **Replicate some of the plots from the California Housing Dataset with the Boston Housing Dataset included with scikit learn**
  - ```
    import pandas as pd
    from sklearn.datasets import load_boston
    boston = load_boston()
    boston_pd = pd.DataFrame(boston.data)
    ```
- **Be sure to add median value to the main dataframe**
  - `boston_pd.columns = boston.feature_names`
  - `boston_pd['MEDV'] = boston.target`
- **Generate a histogram for all of the data columns**
- **Generate a scatter matrix for the attributes "MEDV", "LSTAT", “RM”, "AGE"**

Also check out: https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html
    ```

Reading for Monday

- Ch 4: “Representing Data/Engineering Features” in Guido, Sarah and Andreas C. Muller. (2016). *Introduction to Machine Learning with Python*, O’Reilly Media, Inc. 213–55

Project 1

- **Due June 13**
- **Keep exploring potential datasets**
  - kaggle.com
  - archive.ics.uci.edu/ml/datasets.php
  - libguides.nypl.org/eresources
  - opendata.cityofnewyork.us/data/
- **The data set will need to be labeled as you are going to use it for both supervised and unsupervised learning tasks**

DataCamp for next week

- *Introduction to Python (If Needed)*
- AI Fundamentals
  - Introduction to AI
- Data Manipulation with pandas
  - Transforming Data
  - Aggregating Data
  - Slicing and Indexing
  - *Creating and Visualizing DataFrames (Optional)*
- *Writing Efficient Code with pandas (Optional)*