

```
In [1]: # 機械学習 p. 48~57

# 2.3 データを手に入れる
# データのダウンロード並びに表示とテストセット

import os
import tarfile
import urllib
```

```
In [2]: # 定数定義

DOWNLOAD_ROOT = "https://raw.githubusercontent.com/ageron/handson-ml2/master/"
HOUSING_PATH = os.path.join("datasets", "housing")
HOUSING_URL = DOWNLOAD_ROOT + "datasets/housing/housing.tgz"
```

```
In [3]: # tgzファイルの読み込み

def fetch_housing_data(housing_url=HOUSING_URL, housing_path=HOUSING_PATH):
    os.makedirs(housing_path, exist_ok=True)
    tgz_path = os.path.join(housing_path, "housing.tgz")
    urllib.request.urlretrieve(housing_url, tgz_path)
    housing_tgz = tarfile.open(tgz_path)
    housing_tgz.extractall(path=housing_path)
    housing_tgz.close()
```

```
In [4]: # CSVファイルの読み込み

import pandas as pd

def load_housing_data(housing_path=HOUSING_PATH):
    csv_path = os.path.join(housing_path, "housing.csv")
    return pd.read_csv(csv_path)
```

```
In [5]: # Main Program

fetch_housing_data()

housing = load_housing_data()
housing.head() # .tail()
```

```
# longitude latitude housing_median_age total_rooms total_bedrooms households median_income median_house_value ocean_proximity
# 経度 緯度 築年数の中央値 部屋数 寝室数 世帯数 収入の中央値 住宅価格の中央値 海との位置関係
```

```
Out[5]:
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY

```
In [6]: # データについての情報
housing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   longitude              20640 non-null  float64
1   latitude               20640 non-null  float64
2   housing_median_age     20640 non-null  float64
3   total_rooms            20640 non-null  float64
4   total_bedrooms         20433 non-null  float64
5   population             20640 non-null  float64
6   households              20640 non-null  float64
7   median_income          20640 non-null  float64
8   median_house_value     20640 non-null  float64
9   ocean_proximity        20640 non-null  object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

```
In [7]: # データの種類や個数の表示
housing["ocean_proximity"].value_counts()
```

```
Out[7]: <1H OCEAN      9136
INLAND         6551
NEAR OCEAN     2658
NEAR BAY       2290
ISLAND          5
Name: ocean_proximity, dtype: int64
```

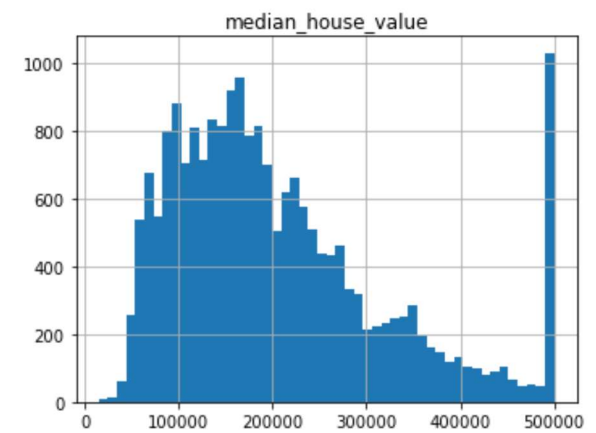
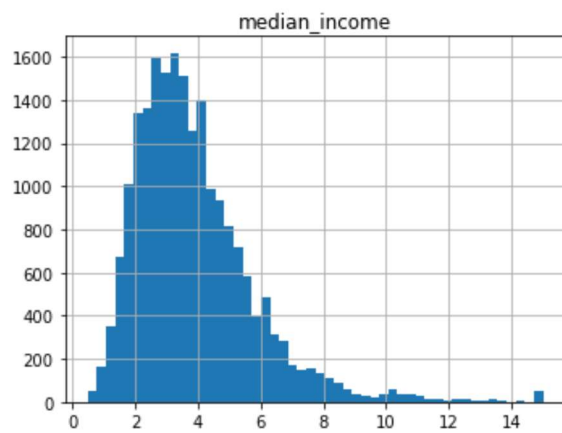
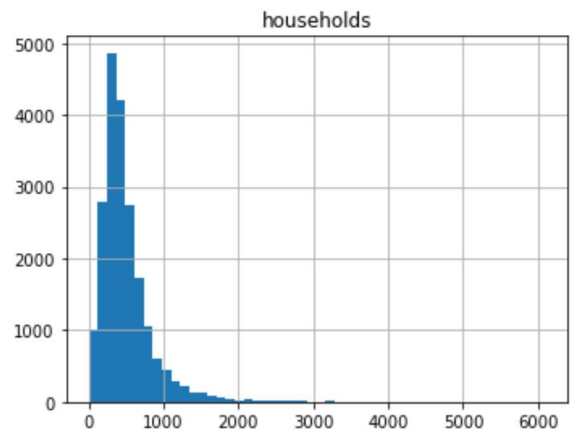
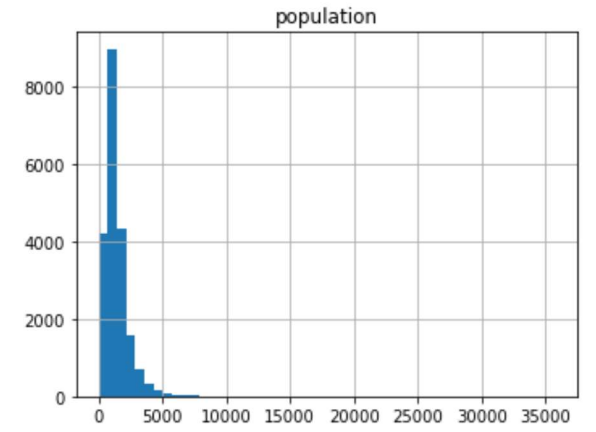
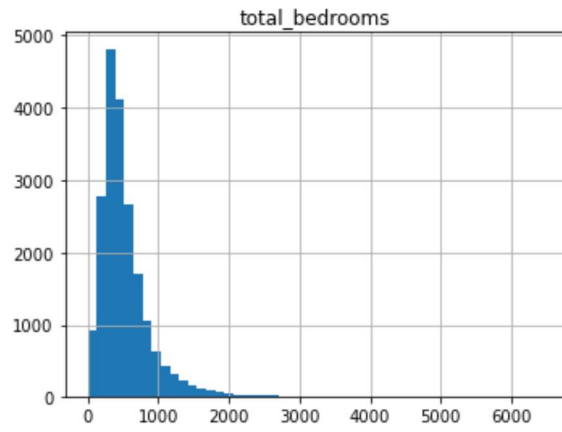
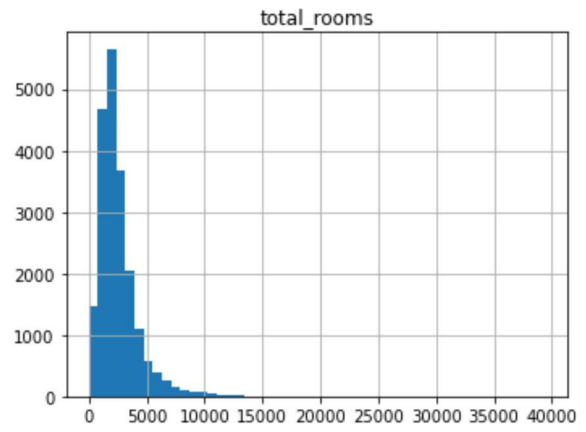
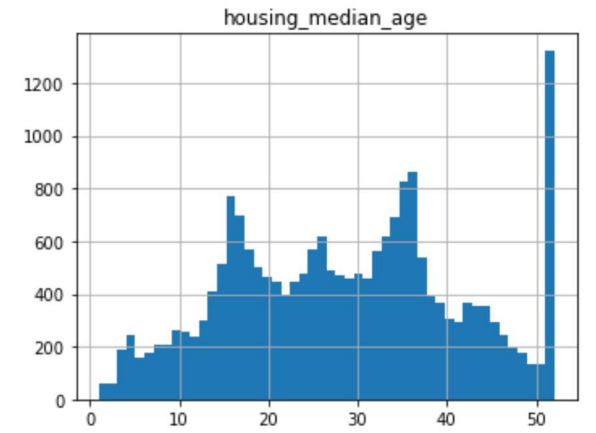
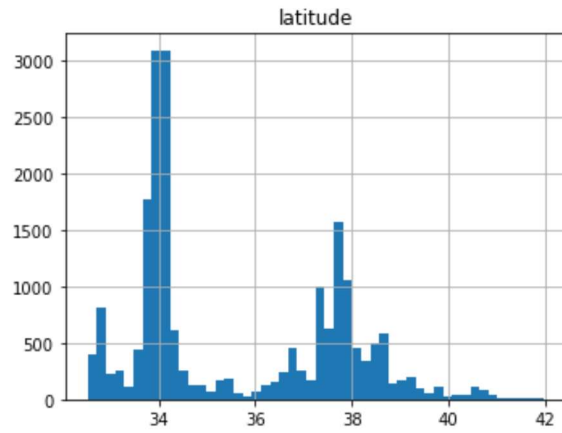
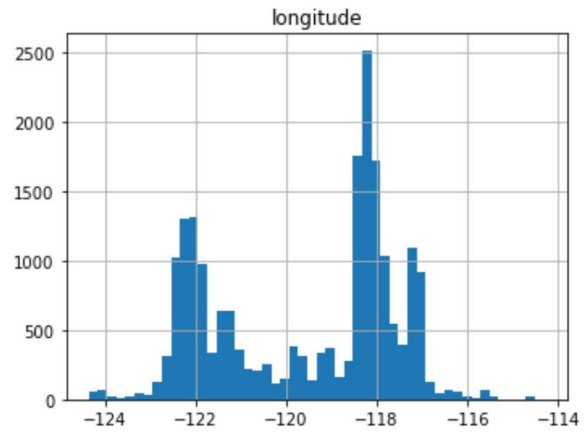
```
In [8]: # 数値情報の集計情報の表示
housing.describe()
```

Out [8]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671	206855.816909
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822	115395.615874
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400	119600.000000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800	179700.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250	264725.000000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100	500001.000000

```
In [9]: # 数値属性のヒストグラムの表示
import matplotlib.pyplot as plt
housing.hist(bins=50, figsize=(20, 15))
plt.show
```

```
Out [9]: <function matplotlib.pyplot.show(close=None, block=None)>
```



```
In [10]: # テストセット作成用関数

import numpy as np

def split_train_test(data, test_ratio):
    np.random.seed(42) # 乱数の種の設定
    shuffled_indices = np.random.permutation(len(data))
    test_set_size = int(len(data) * test_ratio)
    test_indices = shuffled_indices[:test_set_size]
    train_indices = shuffled_indices[test_set_size:]
    return data.iloc[train_indices], data.iloc[test_indices]
```

```
In [11]: # テストセットの作成

train_set, test_set = split_train_test(housing, 0.2)
len(train_set), len(test_set)
```

Out[11]: (16512, 4128)

```
In [12]: test_set.head()
```

```
Out[12]:
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proxim
20046	-119.01	36.06	25.0	1505.0	NaN	1392.0	359.0	1.6812	47700.0	INLA
3024	-119.46	35.14	30.0	2943.0	NaN	1565.0	584.0	2.5313	45800.0	INLA
15663	-122.44	37.80	52.0	3830.0	NaN	1310.0	963.0	3.4801	500001.0	NEAR
20484	-118.72	34.28	17.0	3051.0	NaN	1705.0	495.0	5.7376	218600.0	<1H OC
9814	-121.93	36.62	34.0	2351.0	NaN	1063.0	428.0	3.7250	278000.0	NEAR OC

```
In [13]: # テストセットの作成の工夫
# scikit-learn の関数によるサブセット分割

from sklearn.model_selection import train_test_split

train_set, test_set = train_test_split(housing, test_size=0.2, random_state=42)
test_set.head()
```

```
Out[13]:
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proxim
--	-----------	----------	--------------------	-------------	----------------	------------	------------	---------------	--------------------	--------------

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proxim
20046	-119.01	36.06	25.0	1505.0	NaN	1392.0	359.0	1.6812	47700.0	INLA
3024	-119.46	35.14	30.0	2943.0	NaN	1565.0	584.0	2.5313	45800.0	INLA
15663	-122.44	37.80	52.0	3830.0	NaN	1310.0	963.0	3.4801	500001.0	NEAR
20484	-118.72	34.28	17.0	3051.0	NaN	1705.0	495.0	5.7376	218600.0	<1H OC
9814	-121.93	36.62	34.0	2351.0	NaN	1063.0	428.0	3.7250	278000.0	NEAR OC

```
In [14]: # 層化抽出法 (stratified sampling)
# 収入カテゴリ項目の追加

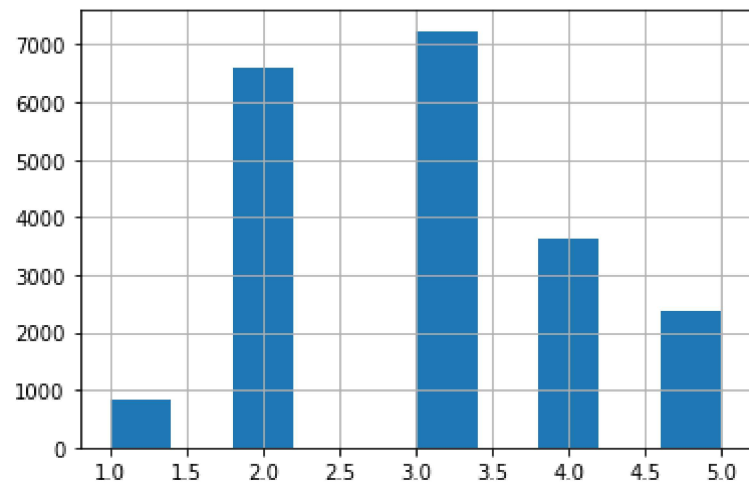
housing["income_cat"] = pd.cut(housing["median_income"], bins=[0.0, 1.5, 3.0, 4.5, 6.0, np.inf], labels=[1, 2, 3, 4, 5])
housing.head()
```

```
Out[14]:
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY

```
In [15]: housing["income_cat"].hist()
```

```
Out[15]: <AxesSubplot:>
```



```
In [16]: # 収入カテゴリによる層化抽出

from sklearn.model_selection import StratifiedShuffleSplit

split = StratifiedShuffleSplit(n_splits=1, test_size=0.2, random_state=42)

for train_index, test_index in split.split(housing, housing["income_cat"]):
    strat_train_set = housing.loc[train_index]
    strat_test_set = housing.loc[test_index]

strat_test_set["income_cat"].value_counts() / len(strat_test_set)
```

```
Out[16]: 3    0.350533
         2    0.318798
         4    0.176357
         5    0.114583
         1    0.039729
Name: income_cat, dtype: float64
```

```
In [17]: # 無作為抽出

train_set, test_set = train_test_split(housing, test_size=0.2, random_state=42)
test_set["income_cat"].value_counts() / len(test_set)
```

```
Out[17]: 3    0.358527
         2    0.324370
         4    0.167393
         5    0.109496
         1    0.040213
Name: income_cat, dtype: float64
```

In [18]: # 全体

```
housing["income_cat"].value_counts() / len(housing)
```

```
Out[18]: 3    0.350581
2    0.318847
4    0.176308
5    0.114438
1    0.039826
Name: income_cat, dtype: float64
```

In [19]: # income_cat 属性の削除

```
for set_ in (strat_train_set, strat_test_set):
    set_.drop("income_cat", axis=1, inplace=True)
```

In [20]: strat_test_set.head()

```
Out[20]:
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proxim
5241	-118.39	34.12	29.0	6447.0	1012.0	2184.0	960.0	8.2816	500001.0	<1H OC
10970	-117.86	33.77	39.0	4159.0	655.0	1669.0	651.0	4.6111	240300.0	<1H OC
20351	-119.05	34.21	27.0	4357.0	926.0	2110.0	876.0	3.0119	218200.0	<1H OC
6568	-118.15	34.20	52.0	1786.0	306.0	1018.0	322.0	4.1518	182100.0	INLA
13285	-117.68	34.07	32.0	1775.0	314.0	1067.0	302.0	4.0375	121300.0	INLA