Lecture 15: Exploratory data analysis (EDA)

Textbook: chapter 10

Dr. Huiping Cao

Example dataset – House data

- The housing dataset contains information about houses in the California.
- 8 **Explanatory variables**
  - MedInc median income in block group
  - HouseAge median house age in block group
  - AveRooms average number of rooms per household
  - AveBedrms average number of bedrooms per household
  - Population block group population
  - AveOccup average number of household members
  - Latitude block group latitude
  - Longitude block group longitude,
- **Response variable**: target variable, MEDV (median value of owner-occupied homes)

Read in house data

```
from sklearn.datasets import fetch_california_housing
import numpy as np
import pandas as pd

CA_housing = fetch_california_housing()
print('CA_housing feature names:', CA_housing.feature_names)

df_data = pd.DataFrame(CA_housing.data[:1000], columns = CA_housing.feature_names)
df_target = pd.DataFrame(CA_housing.target[:1000], columns=['MEDV'])
df = pd.concat([df_data, df_target],axis=1)
print(df.info)
```

```
CA_housing feature names: ['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'Population', 'AveOccup',
'Latitude', 'Longitude’]


…
[1000 rows x 9 columns]
```

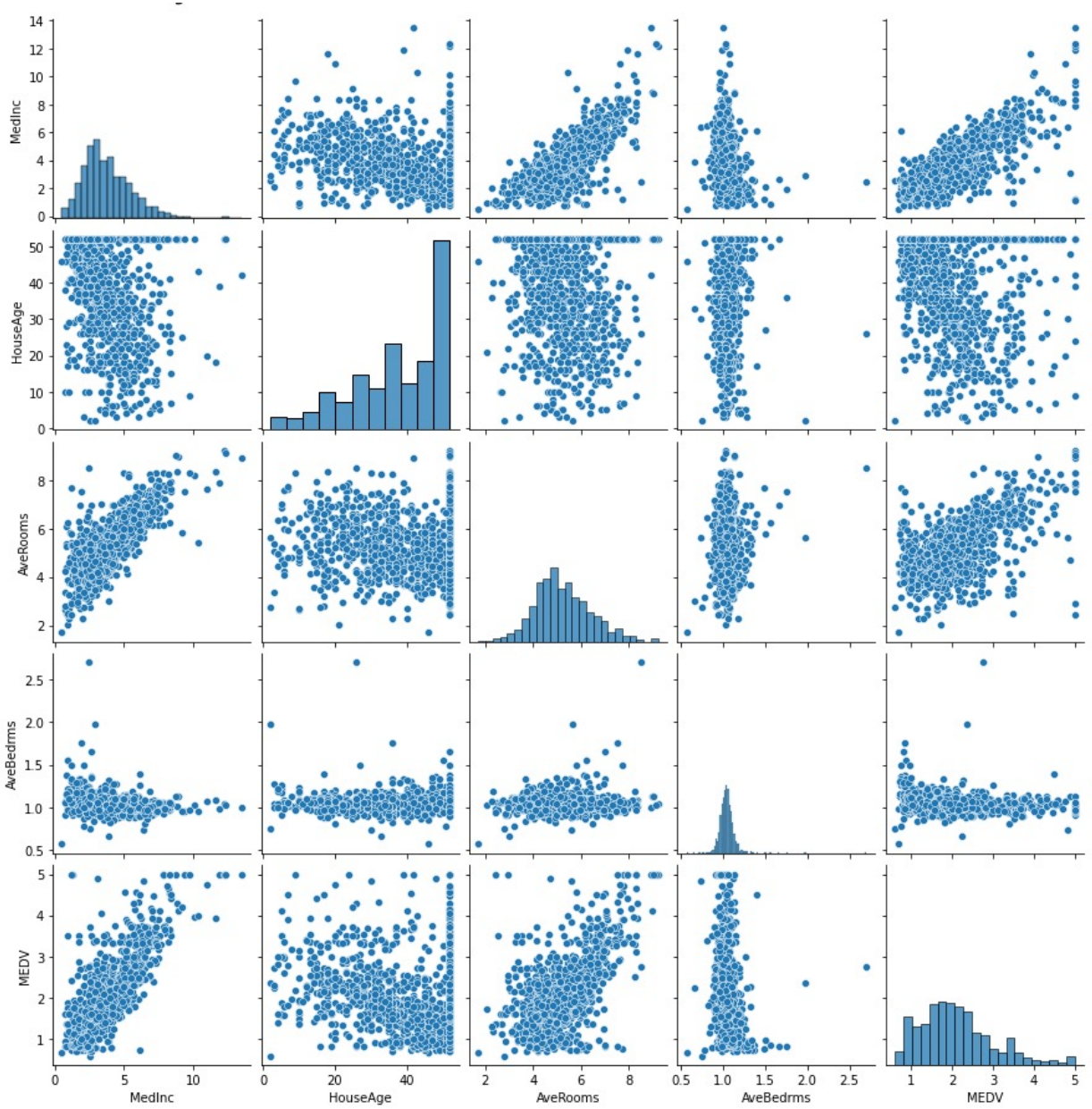
Exploratory data analysis (EDA)

- **Exploratory data analysis (EDA)** can help us get a basic idea about the data (e.g., patterns, anomalies, or relationships).
  - E.g., detect the presence of outliers, the distribution of the data, and the relationships between features.
- **Seaborn library** (http://seaborn.pydata.org/) is a Python library for drawing statistical plots based on Matplotlib.

Scatterplot matrix

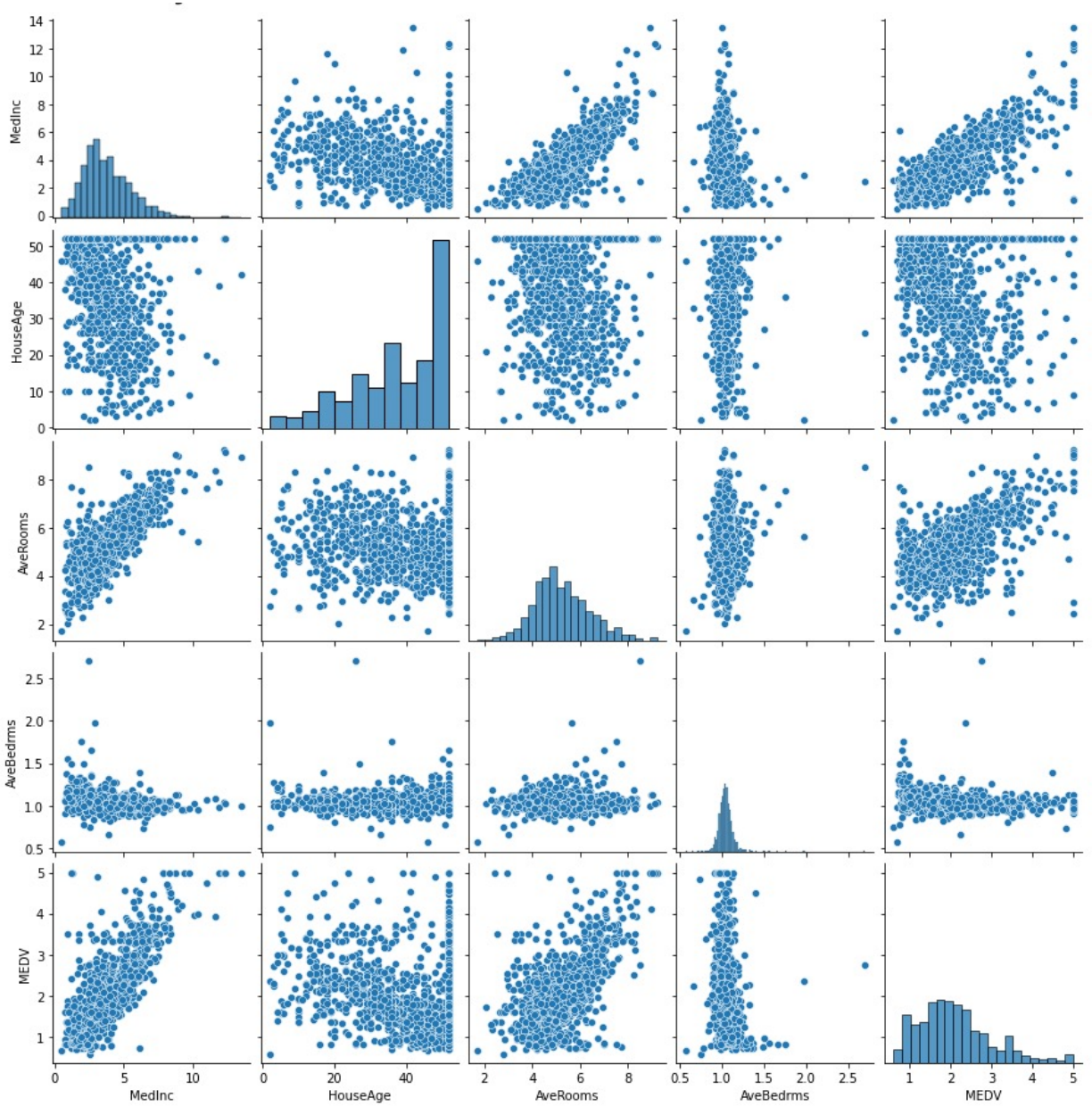
- Visualize the pair-wise correlations between the different features in one place.

```
import matplotlib.pyplot as plt
import seaborn as sns

cols = ['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'MEDV']
sns.pairplot(df[cols],size=2.5)

plt.tight_layout()
plt.show()
```



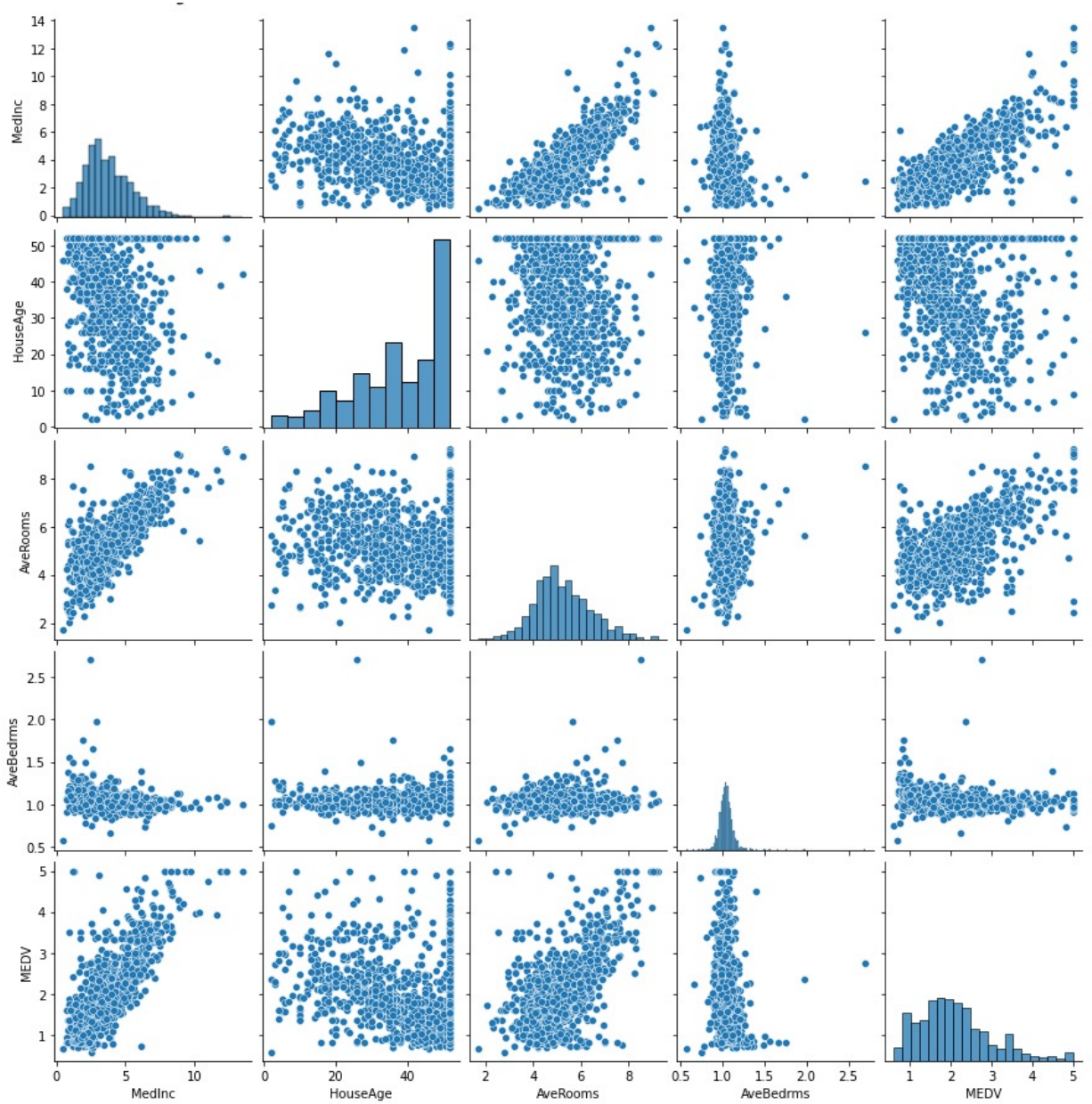
Pairplot function

- By default, this function creates a grid of Axes such that each variable in data will be shared in the **y- axis** across a single row and in the **x-axis** across a single column.
- The **diagonal Axes** are treated differently. It draws a plot to show the **univariate distribution** of the data for the variable in that column.
- It is also possible to show a **subset of variables or plot different variables** on the rows and columns.



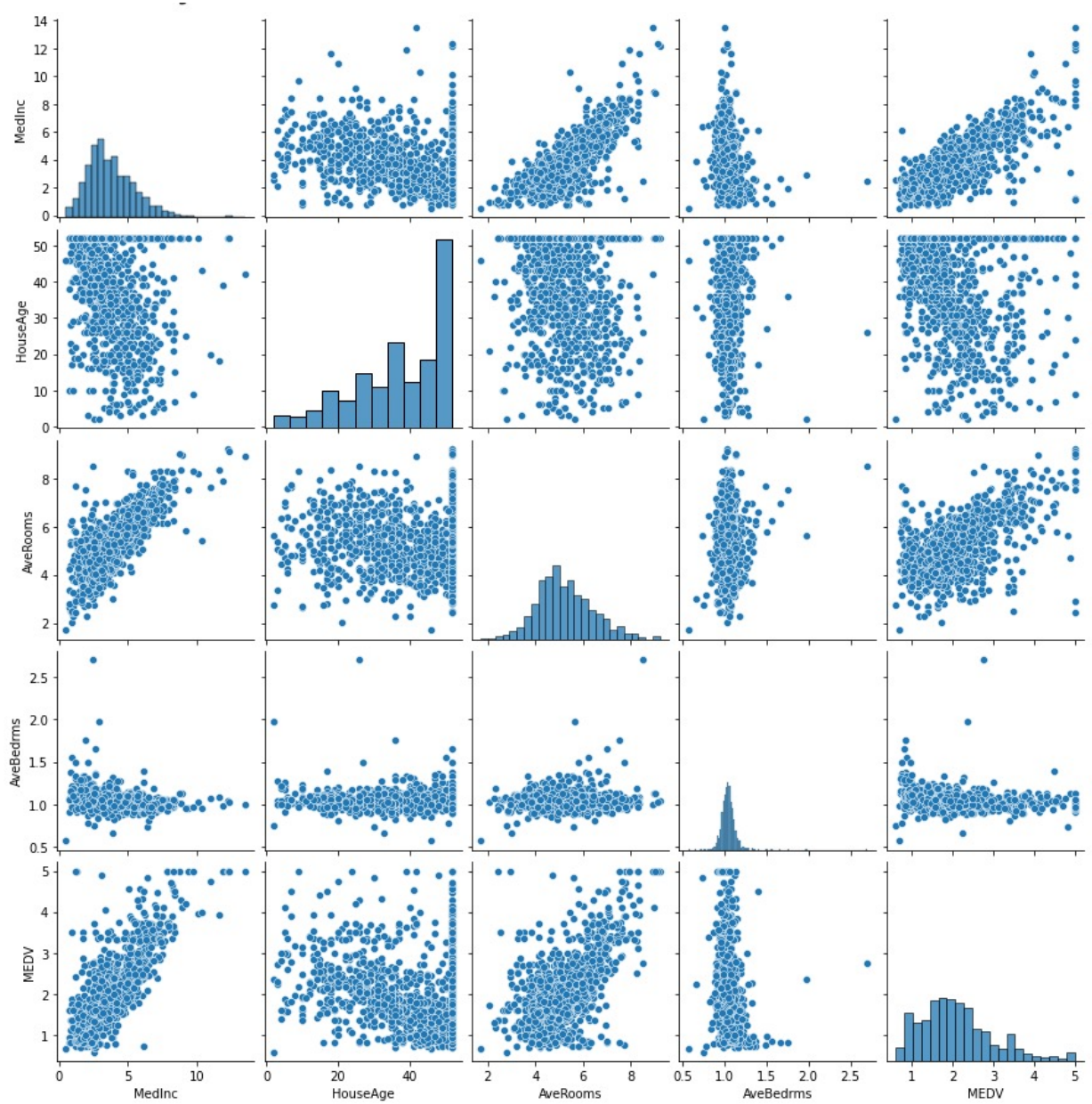
Pairplot function

- The pair plot allows us to see both (i) the **distribution of single variables** (diagonal Axes) and (ii) the **relationships between two variables**
- House price seems to be normally distributed but contains several outliers.
- **Note**: Linear regression does not require that the explanatory or target variables are normally distributed.



Pairplot

- **Observations**:
  There is a linear relationship between MedInc and MEDV (house prices).



Correlation matrix

- Correlation matrix is a square matrix containing the **Pearson product-moment correlation coefficient** (which is abbreviated as **Pearson's r**). It represents the linear dependence between pairs of features.
- Given a dataset with *n* instances and *m* features. Its correlation matrix is an $m \times m$ matrix.

Coefficient of features *x* and *y*

- Coefficient of feature *x* and *y*

$$r_{xy} = \frac{\sum_{i=1}^{n}[\,(x^{(i)} - \mu_x)(y^{(i)} - \mu_y)\,]}{\sqrt{\sum_{i=1}^{n}(x^{(i)} - \mu_x)^2} \cdot \sqrt{\sum_{i=1}^{n}(y^{(i)} - \mu_y)^2}} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

- $\mu_x(\mu_y)$: sample mean of feature *x* (*y*)
- σ_{xy}: covariance between features *x* and *y*
- $\sigma_x(\sigma_y)$: the standard deviation of feature *x* (*y*)

Property of Pearson correlation coefficients

- **Correlation coefficients** are in the range of [-1, 1].
  - if *r* = 1: perfect positive correlation.
    if *r* = 0: no correlation.
    if *r* = −1: perfect negative correlation.
- **Correlation matrix is a rescaled version of the covariance matrix**: correlation matrix is identical to a covariance matrix computed from standardized features.
  - The linear correlation coefficient of two features r_{xy} equals to the covariance σ'_{xy} between their standardized features *x'* and *y'*.

Calculate correlation matrix

- For our response variable, the largest correlation is with MedInc (-0.77).
  - From the scatterplot matrix, we can see clearly that there is no linear relationship between LSTAT and the target variable.
- The correlation with AveRooms is the 2nd highest (0.58).

```
import numpy as np

cols = ['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'MEDV']
correlation_coefficient = np.corrcoef(df[cols].values.T)
print(correlation_coefficient)
```

```
[[ 1.         -0.26741281  0.77162956 -0.23703851  0.76659388]
 [-0.26741281  1.         -0.0978515   0.00423074 -0.13832933]
 [ 0.77162956 -0.0978515   1.          0.07405811  0.58120513]
 [-0.23703851  0.00423074  0.07405811  1.         -0.16955287]
 [ 0.76659388 -0.13832933  0.58120513 -0.16955287  1.        ]]
```

Draw correlation matrix

```
sns.heatmap(correlation_coefficient, annot=True,
        yticklabels = cols, xticklabels=cols)
plt.show()
```

- More intuitive to see the relationships.
- To fit a linear regression model, we are interested in using features that have a high correlation with our target variable.