Exploratory Data Analysis (EDA) with Python

Thuan L Nguyen, PhD

1. Data Set: housing_boston.csv

For this project:
--) We will investigate the Boston House Price dataset.
--) Each record in the database describes a Boston suburb or town.
--) The data was drawn from the Boston Standard Metropolitan Statistical Area (SMSA) in 1970.

The attributes are defined as follows:

1. CRIM: per capita crime rate by town
2. ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS: proportion of non-retail business acres per town
4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. NOX: nitric oxides concentration (parts per 10 million)
6. RM: average number of rooms per dwelling
7. AGE: proportion of owner-occupied units built prior to 1940
8. DIS: weighted distances to five Boston employment centers
9. RAD: index of accessibility to radial highways
10. TAX: full-value property-tax rate per 10,000 dollars
11. PTRATIO: pupil-teacher ratio by town
12. B: 1000(Bk - 0:63)^2 where Bk is the proportion of blacks by town
13. LSTAT: % lower status of the population
14. MEDV: Median value of owner-occupied homes in 1000 dollars

For this lecture: we use a **subset of the original dataset**.
The sub-dataset only contains the following variables:

1. RM: average number of rooms per dwelling
2. AGE: proportion of owner-occupied units built prior to 1940
3. DIS: weighted distances to five Boston employment centers
4. RAD: index of accessibility to radial highways
5. PTRATIO: pupil-teacher ratio by town
6. MEDV: Median value of owner-occupied homes in 1000 dollars

```
1
```

```
1
```

```
1
```

```
# 2. Load Data
```

2.1 Import Python Libraries and Modules

In [20]:

```
# Import Python libraries: NumPy and Pandas
import pandas as pd
import numpy as np

# Import libraries & modules for data visualization
from pandas.plotting import scatter_matrix
from matplotlib import pyplot

# Import the module for converting float to currency
import locale
```

2.2 Load Dataset

In [21]:

```
# Set locale: Currency in US Dollar
locale.setlocale( locale.LC_ALL, 'English_United States.1252' )

# Specify what and where is the data file
filename = 'DATA/housing_boston.csv'

# Specify the fields with their names
names = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRA

# Load the data into a Pandas DataFrame
df = pd.read_csv(filename, names=names)

# VIP NOTES: [[ ... ]]
# Extract a sub-dataset from the original one --> dataframe: df2
df2 = df[['RM', 'AGE', 'DIS', 'RAD', 'PTRATIO', 'MEDV']]
```

3. Preprocess Dataset

3.1 Clean Data: Find & Mark Missing Values

NOTES:

The following columns cannot contain 0 (zero) values.
i.e., 0 (zero) values are invalid in these columns.

- RM: average number of rooms per dwelling
- PTRATIO: pupil-teacher ratio by town

- MEDV: Median value of owner-occupied homes in 1000 dollars

If they exist, we need to mark them as missing value or numpy.NaN

In [22]:

```
# mark zero values as missing or NaN
df[['RM', 'PTRATIO', 'MEDV']] = df[['RM', 'PTRATIO', 'MEDV']].replace(0, np.NaN)

# count the number of NaN values in each column
print(df.isnull().sum())
```

```
RIM       0
N         0
NDUS      0
HAS       0
OX        0
M         0
GE        0
IS        0
AD        0
AX        0
TRATIO    0
          0
STAT      0
EDV       0
type: int64
```

NOTES:

So, there is no invalid 0 (zero) value in any column of the original dataframe.
We don't have to clean the dataset.

4. Perform Exploratory Data Analysis on Dataset

4.1 Get Shape

In [23]:

```
# Get the dimensions or Shape of the dataset
# i.e. number of records/rows x number of variables/columns

print(df2.shape)
```

```
(452, 6)
```

In [ ]:

```

```

```

```

4.2 Get Data Types

In [24]:

```
# Get the data types of all the variables/attributes of the data set
# The results shows

print(df2.dtypes)
```

```
M          float64
GE         float64
IS         float64
AD           int64
TRATIO     float64
EDV        float64
type: object
```

4.3 Have a Sneak Peek of Data

In [25]:

```
# Get several records/rows at the top of the dataset
# Get the first five records

print(df2.head(5))
```

```
      RM   AGE     DIS  RAD  PTRATIO  MEDV
0  6.575  65.2  4.0900    1     15.3  24.0
1  6.421  78.9  4.9671    2     17.8  21.6
2  7.185  61.1  4.9671    2     17.8  34.7
3  6.998  45.8  6.0622    3     18.7  33.4
4  7.147  54.2  6.0622    3     18.7  36.2
```

4.4 Get Statistics Summary

In [26]:

```
# Get the summary statistics of the numeric variables/attributes of the dataset

print(df2.describe())
```

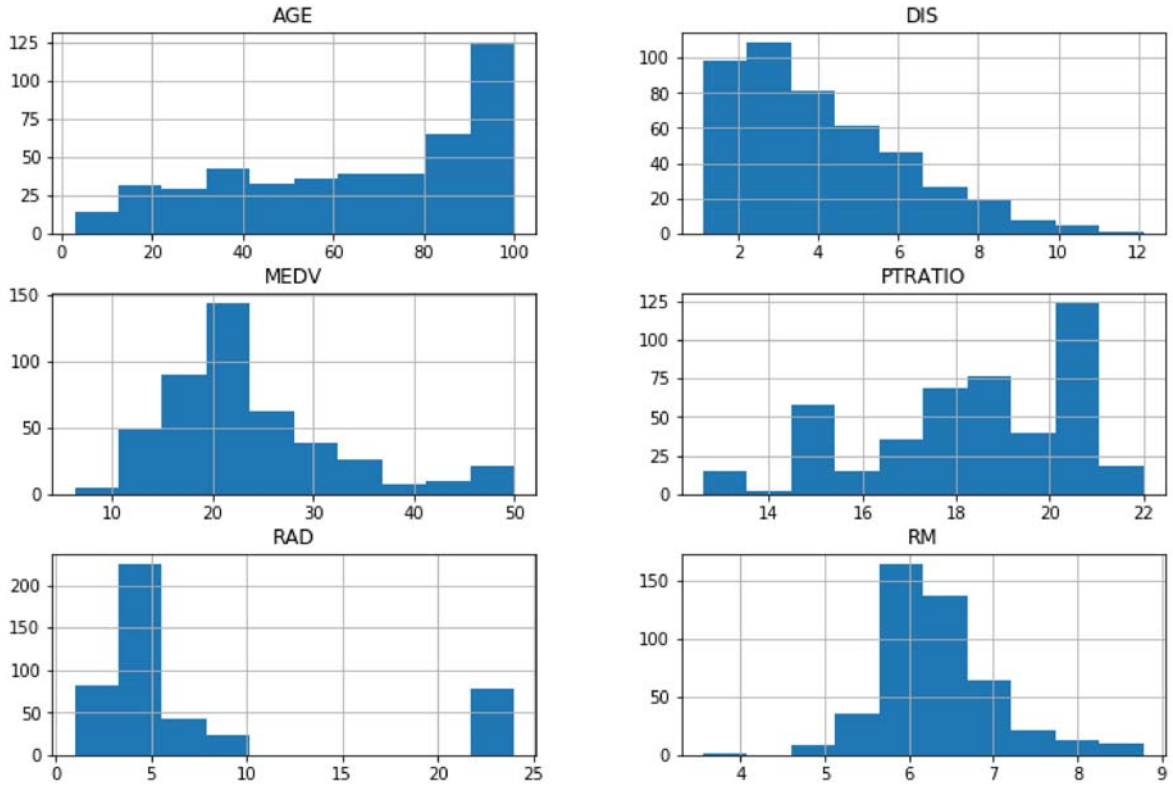
	RM	AGE	DIS	RAD	PTRATIO	MED
ount	452.000000	452.000000	452.000000	452.000000	452.000000	452.00000
ean	6.343538	65.557965	4.043570	7.823009	18.247124	23.75044
td	0.666808	28.127025	2.090492	7.543494	2.200064	8.80860
in	3.561000	2.900000	1.129600	1.000000	12.600000	6.30000
5%	5.926750	40.950000	2.354750	4.000000	16.800000	18.50000
0%	6.229000	71.800000	3.550400	5.000000	18.600000	21.95000
5%	6.635000	91.625000	5.401100	7.000000	20.200000	26.60000
ax	8.780000	100.000000	12.126500	24.000000	22.000000	50.00000

4.5 Visualize Data: Histograms

In [27]:

```
# Plot histrogram for each numeric variable/sttribute of the dataset
# VIP NOTES: The first variable ID is also plotted. However the plot should be ignored

df2.hist(figsize=(12, 8))
pyplot.show()
```

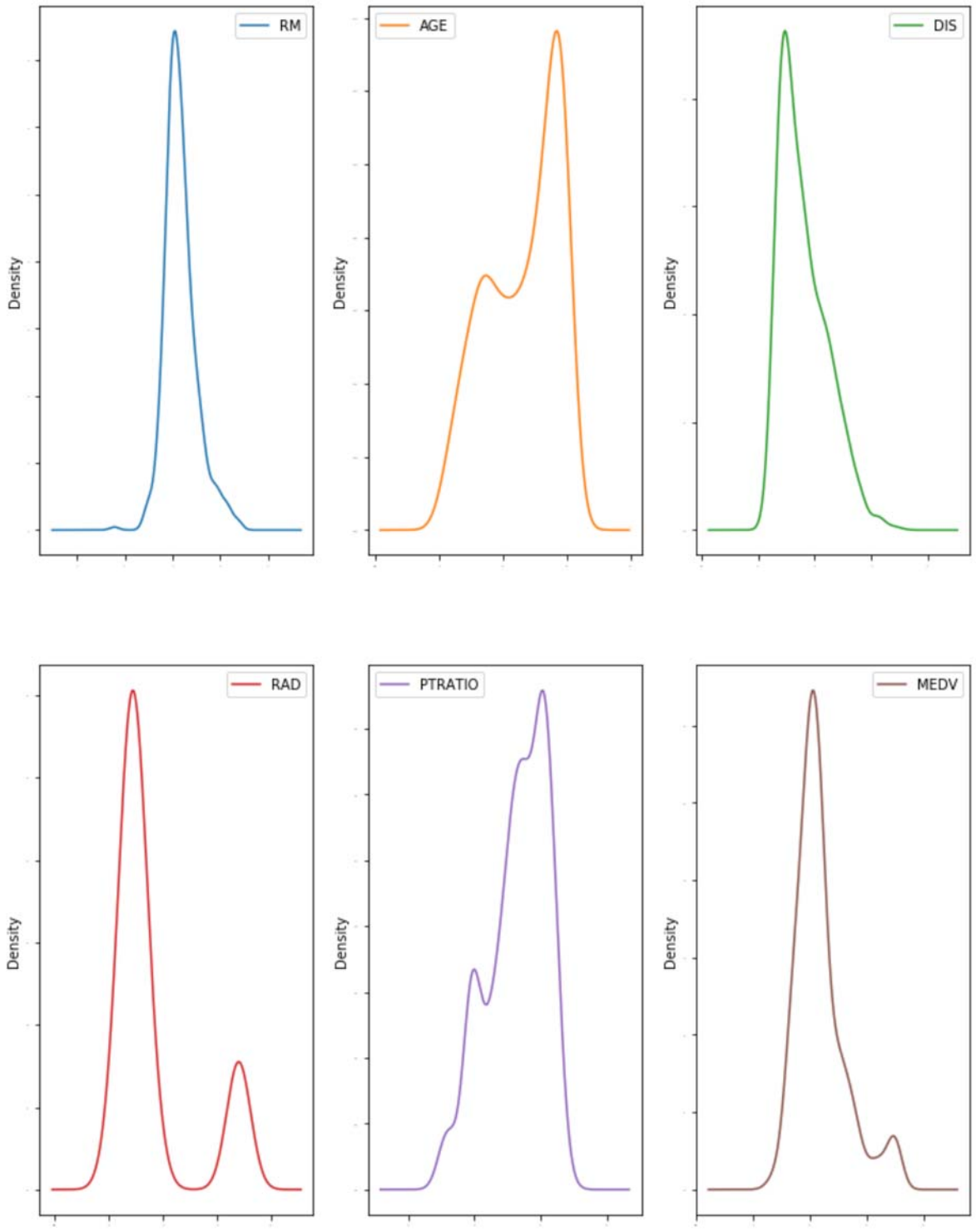


4.6 Visualize Data: Density Plot

In [28]:

```
# Density plots
# IMPORTANT NOTES: 5 numeric variables --> at least 5 plots --> layout (2, 3): 2 rows,

df2.plot(kind='density', subplots=True, layout=(2, 3), sharex=False, legend=True, font
pyplot.show()
```

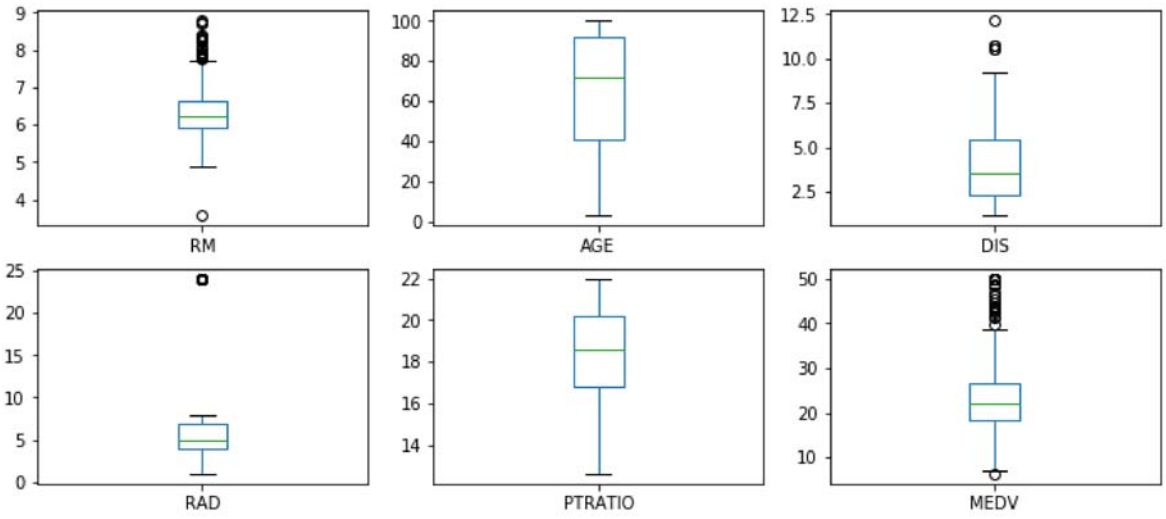



```
1
```

4.7 Visualize Data: Boxplot

In [29]:

```
df2.plot(kind='box', subplots=True, layout=(3,3), sharex=False, sharey=False, figsize=
pyplot.show()
```



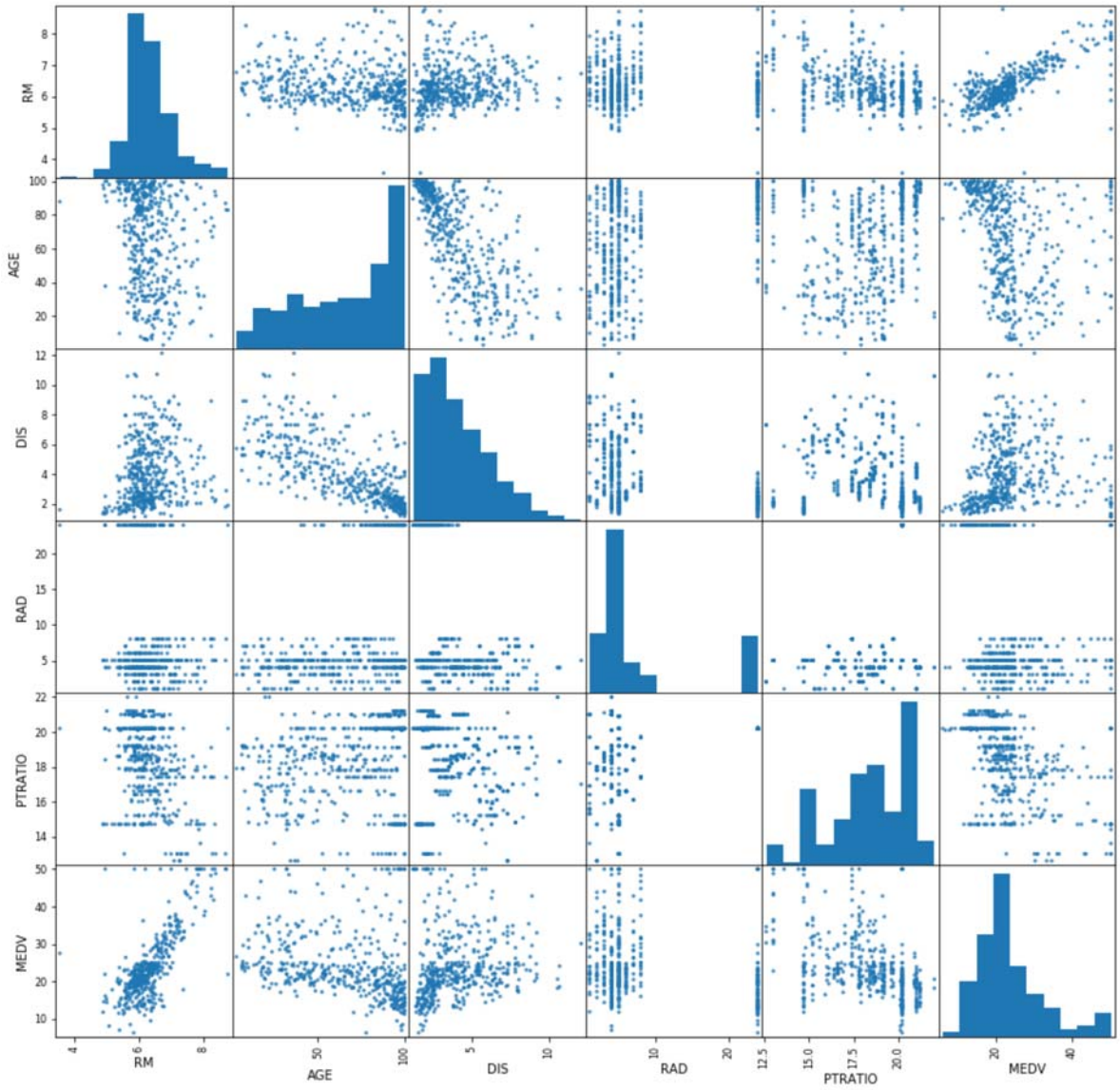
4.8: Visualize Data (Multivariate): Scatter Plot

In [30]:

```
# scatter plot matrix

scatter_matrix(df2, alpha=0.8, figsize=(15, 15))

pyplot.show()
```



In [ ]:

```

```