

# Simple and Multiple Linear Regression

## (Djaouida Dahmani-Zaouche)

### Exercise 1

1. Consider the following instructions:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
x=pd.DataFrame([[i] for i in np.arange(1000)])
y=pd.DataFrame([2*i for i in np.arange(1000)])
#make noise
y=y+pd.DataFrame([np.random.rand(1) for i in
np.arange(1000)])
y=pd.DataFrame([2*i + np.random.rand(1,) for i in
np.arange(1000)])
print(x.shape)
print(y.shape)
lm.fit(x,y)
print(lm.coef_)
print(lm.intercept_)
x=lm.predict([[3],[1000]])
print('Predicted value of 3 is',x[0][0])
print('Predicted value of 1000 is',x[1][0])*
```

- What is the role of the LinearRegression package?
- What is the role of the fit function fit LinearRegression?
- Give the formula modelling here the hypothesis function  $h(x)$ .

- Explain line \*.
- Show the chart and conclude

```
import matplotlib.pyplot as plt
plt.plot(y,lm.predict(x),c='b', ls=':')
plt.xlabel("Real $Y_i$")
plt.ylabel("Estimated $\hat{Y}_i$")
plt.title("Real and Estimated Prices")
```

## Exercise 2

In this exercise, we use a data set available in [sklearn](#) Python module. We are going to import Boston data set into Ipython notebook and store it in a variable called *boston*. We will make analysis on the data of *Boston.data*.

Execute these instructions:

```
from sklearn.datasets import load_boston
boston= load_boston()
boston.keys()
```

1. What is the kind of *boston*.
2. Display *boston.target* and its shape.
3. Display *boston.data* and its shape.
4. Display *boston.feature*. Give a sense of each name (use `print(boston.DESCR)`).
5. Define a DataFrame *df* where its first rows as follows:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	Price
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3.0	222.0	18.7	394.12	5.21	28.7
6	0.08829	12.5	7.87	0.0	0.524	6.012	66.6	5.5605	5.0	311.0	15.2	395.60	12.43	22.9
7	0.14455	12.5	7.87	0.0	0.524	6.172	96.1	5.9505	5.0	311.0	15.2	396.90	19.15	27.1
8	0.21124	12.5	7.87	0.0	0.524	5.631	100.0	6.0821	5.0	311.0	15.2	386.63	29.93	16.5
9	0.17004	12.5	7.87	0.0	0.524	6.004	85.9	6.5921	5.0	311.0	15.2	386.71	17.10	18.9

**Figure 1**

Indication: use `boston.data` for the constructor `DataFrame`, `boston.feature_names` to name the columns and `boston.target` for the last column of `df`.

6. Display the number of rows and columns of `df`. Display the 3 first lines and the 4 last lines of `df`.

7. `from sklearn.linear_model import LinearRegression`

8. Create an object

```
lm = LinearRegression()
```

9. Execute these instructions:

```
print(boston.data)
```

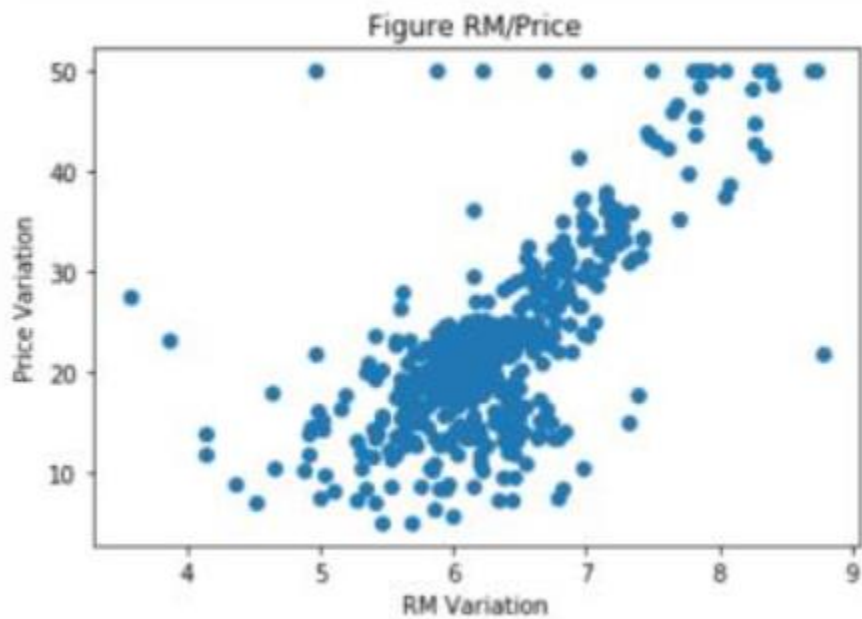
```
lm.fit(boston.data,boston.target )
```

Print the coefficients and intercept of the equation defined by `lm`. Give its literal expression

10. Write the python instructions to display the table of figure 2 and the chart of figure 3.

	Feature	Estimated coef
x0	CRIM	-0.107171
x1	ZN	0.046395
x2	INDUS	0.020860
x3	CHAS	2.688561
x4	NOX	-17.795759
x5	RM	3.804752
x6	AGE	0.000751
x7	DIS	-1.475759
x8	RAD	0.305655
x9	TAX	-0.012329
x10	PTRATIO	-0.953464
x11	B	0.009393
x12	LSTAT	-0.525467

Figure 2



**Figure 3**

11. Execute and interpret:
 

```
X=pd.DataFrame(botson.data)
lm.predict(X)
```
12. Write the python instructions which allow to build the chart of figure 4.



**Figure 4**

13. Reconsider questions 9..13. Split `boston.data` and `boston.target` into `x_train`, `x_test`, `y9_train`, `y_test`. For this question, use function `train_test_split` and parameter `test_size = 0.5` and import `train_test_split` from `sklearn.model_selection`.