week-4

Python implementation of linear regression

```
import numpy as np
import matplotlib.pyplot as plt

def estimate_coef(x,y):
    n = np.size(x)
    m_x = np.mean(x)
    m_y = np.mean(y)
    SS_xy = np.sum(y*x) - n*m_y*m_x
    SS_xx = np.sum(x*x) - n*m_x*m_x
    b_1 = SS_xy / SS_xx
    b_0 = m_y - b_1*m_x
    return (b_0, b_1)
def plot_regression_line(x, y, b):
    plt.scatter(x, y, color = "m", marker = "o", s = 30)
    y_pred = b[0] + b[1]*x
    plt.plot(x, y_pred, color = "g")
    plt.xlabel('x')
    plt.ylabel('y')
def main():
    x = np.array([0,1,2,3,4,5,6,7,8,9])
    y = np.array([1,3,2,5,7,8,9,10,12,7])
    b = estimate_coeff(x,y)
print("Estimated coefficients: \nb_0 = {} \nb_1 = {}",
                format(b))
```

Output:

(b_0, b_1) = (1.2 3 b 3 . . . . , 1.16969 . . .)

multiple regression line

```
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model, metrics

data_url = "url"
raw_df = pd.read_csv(data_url, sep="\s+", skiprows=2,
                     header=None)

x = np.concatenate([raw_df.values[::2, :], raw_df.
                    values[1::2, :2]])
y = raw_df.values[1::2, 2]

x_train, x_test, y_train, y_test = train_test_split
    (x, y, test_size = 0.4, random_state = 1)

reg = linear_model.LinearRegression()
reg.fit(x_train, y_train)
print("coefficients =", reg.coef_)
print("variance score = {}".format(reg.score
                                   (x_test, y_test)))
plt.style.use('fivethirtyeight')
plt.scatter(reg.predict(x_train), reg.predict(x_train)
            - y_train, color = 'green', s=10, label = 'train data')
plt.scatter(reg.predict(x_test), reg.predict(x_test)
            - y_test, color = 'blue', s=10, label='test data')
plt.hlines(y=0, xmin=0, xmax=50, linewidth=2)
plt.legend(loc='upper right')
plt.title('Residual core')
plt.show()
```