

```

# python: 3.5.2
# encoding: utf-8
# numpy: 1.14.1

import numpy as np
import matplotlib.pyplot as plt

def load_data(filename):
    """载入数据。"""
    xys = []
    with open(filename, 'r') as f:
        for line in f:
            xys.append(map(float, line.strip().split()))
    xs, ys = zip(*xys)
    return np.asarray(xs), np.asarray(ys)

def evaluate(ys, ys_pred):
    """评估模型。"""
    std = np.sqrt(np.mean(np.abs(ys - ys_pred) ** 2))
    return std

def main(x_train, y_train):
    """训练模型，并返回从 x 到 y 的映射。"""

    # 使用线性回归训练模型，根据训练集计算最优化参数
    ## 请补全此处代码，替换以下示例
    phi0 = np.expand_dims(np.ones_like(x_train), axis=1)
    phi1 = np.expand_dims(x_train, axis=1)
    phi = np.concatenate([phi0, phi1], axis=1)
    w = np.dot(np.linalg.pinv(phi), y_train)

    # 返回从 x 到 y 的映射函数 y=f(x)
    # 注意：函数 f(x)的变量只有 x，参数 w 应作为内部变量
    def f(x):
        ## 请补全此处代码，替换以下示例
        phi0 = np.expand_dims(np.ones_like(x), axis=1)
        phi1 = np.expand_dims(x, axis=1)
        phi = np.concatenate([phi0, phi1], axis=1)
        y = np.dot(phi, w)
        return y
    pass

```

```
return f
```

```
# 程序主入口（建议不要改动以下函数的接口）
```

```
if __name__ == '__main__':
```

```
    train_file = 'train.txt'
```

```
    test_file = 'test.txt'
```

```
    # 载入数据
```

```
    x_train, y_train = load_data(train_file)
```

```
    x_test, y_test = load_data(test_file)
```

```
    print(x_train.shape)
```

```
    print(x_test.shape)
```

```
    # 使用线性回归训练模型，返回一个函数 f()使得  $y = f(x)$ 
```

```
    f = main(x_train, y_train)
```

```
    # 计算预测的输出值
```

```
    y_test_pred = f(x_test)
```

```
    # 使用测试集评估模型
```

```
    std = evaluate(y_test, y_test_pred)
```

```
    print('预测值与真实值的标准差: {:.1f}'.format(std))
```

```
    # 显示结果
```

```
    plt.plot(x_train, y_train, 'ro', markersize=3)
```

```
    plt.plot(x_test, y_test, 'k')
```

```
    plt.plot(x_test, y_test_pred)
```

```
    plt.xlabel('x')
```

```
    plt.ylabel('y')
```

```
    plt.title('Linear Regression')
```

```
    plt.legend(['train', 'test', 'pred'])
```

```
    plt.show()
```

Linear Regression

