プログラム

実行環境と用いた言語・ライブラリを以下の表 1 に示す。

表 1: プログラムの実行環境

OS	:	Microsoft Windows 10 Pro (64bit)
CPU	:	Intel(R) Core(TM) i5-4300U
RAM	:	4.00 GB
使用言語	:	Python3.6
可視化	:	matplotlib ライブラリ

Listings 1: assignment3.py

```
# -*- coding: utf-8 -*-


import numpy as np
import matplotlib.pyplot as plt


def generate_sample(x_min=-3., x_max=3., sample_size=10):
    x = np.linspace(x_min, x_max, num=sample_size)
    y = x + np.random.normal(loc=0., scale=.2, size=sample_size)
    y[-1] = y[-2] = y[1] = -4  # outliers
    return x, y


def model(x, theta):
    f = theta[0] + theta[1] * x
    return f


def turkey_loss(r, eta):
    rho = (1 - (1 - (r/eta)**2)**3)/6
    rho[np.abs(r) > eta] = 1/6
    loss = (1/2) * np.sum(rho)
    return loss


def compute_loss(x, y, theta, eta):
    y_pred = model(x, theta)
    r = y_pred - y
    loss = turkey_loss(r, eta)
    return loss


```

```
def calc_Phi(x, theta):
    n = x.shape[0]
    b = theta.shape[0]
    phi = np.zeros((n, b))
    phi[:, 0] = theta[0]
    phi[:, 1] = theta[1] * x
    return phi


def calc_W(r, eta):
    w = (1 - (r/eta)**2)**2
    w[np.abs(r) > eta] = 0
    w = np.diag(w)
    return w


def update(theta, phi, y, w):
    theta = np.linalg.inv(phi.T.dot(w).dot(phi)).dot(phi.T).dot(w).dot(y)
    return theta


def solve(x, y, theta_initial, eta, eps=1e-4, n=5, max_iter=100):
    diffs = []
    theta = theta_initial
    for i in range(max_iter):
        theta_old = theta.copy()
        phi = calc_Phi(x, theta)
        y_pred = model(x, theta)
        r = y_pred - y
        w = calc_W(r, eta)
        theta = update(theta, phi, y, w)
        diff = np.linalg.norm(theta - theta_old)
        if len(diffs) < n:
            diffs.append(diff)
        else:
            if (max(diffs) - min(diffs)) < eps:
                break
            diffs = diffs[1:] + [diff]
    n_iter = i + 1
    return theta, n_iter


def main():
    #np.random.seed(0)  # set the random seed for reproducibility

    # create sample
    x_min, x_max = -3, 3
```

```
81      sample_size = 50
82      x, y = generate_sample(x_min, x_max, sample_size)
83      # print(x.shape, y.shape)
84
85      # hyper parameter
86      eta = 1.5
87
88      # parameter
89      theta_init = np.random.rand(2)
90
91      # solve
92      theta, n_iter = solve(x, y, theta_init, eta, eps=1e-4, n=5, max_iter=100)
93
94      # calc loss
95      loss = compute_loss(x, y, theta, eta)
96
97      # result
98      print('eta: {}'.format(eta))
99      print('theta_init: {}'.format(theta_init))
100     print('theta: {}'.format(theta))
101     print('loss: {:.4f}'.format(loss))
102     print('n_iter: {}'.format(n_iter))
103
104     # plot
105     x_axis = np.linspace(x_min, x_max, 100)
106     plt.scatter(x, y)
107     plt.plot(x_axis, model(x_axis, theta))
108
        plt.savefig('../figures/assignment3_result_eta{}.png'.format(int(10*eta)))
109     plt.show()
110
111
112 if __name__ == '__main__':
113     main()
```