プログラミング技法 2_課題 5

阪田征之助

2023 年 6 月 9 日

ソースコード

ソースコードの主要部を list1 に示す。

list 1　code

```
import pandas as pd
import matplotlib.pylab as plt
import numpy as np
import scipy

N = 100
epsilon = 0.0001
lr = 0.1

if __name__=="__main__":

    #csvファイルからの読み込み
    df = pd.read_csv("weight-height.csv")

    # 05_01
    #100個をランダムに取得
    df_smp = df.sample(N)
    x = df_smp["Weight"].tolist()
    y = df_smp["Height"].tolist()
    plt.figure(1).clf()
    plt.scatter(x, y)
    plt.xlabel('Height')
    plt.ylabel('Weight')

    # 05_02
    #標準化
    x = (x - np.mean(x))/np.sqrt(np.var(x))
    y = (y - np.mean(y))/np.sqrt(np.var(y))
    print('Averaged standardized w:\t{:.2f}+-{:.2f}'.format(np.mean(x), np.std(x))) #
        np.mean(w) = 0, np.std(w) = 1
    print('Averaged standardized h:\t{:.2f}+-{:.2f}'.format(np.mean(y), np.std(y))) #
        np.mean(h) = 0, np.std(h) = 1

    # 05_03
    # MSE
    a = np.cov([x,y])[0][1]/np.var(x)
    b = np.mean(y) - a * np.mean(x)
    mse = sum((y[i] - (b + a*x[i]))**2 for i in range(N))/N
    print('03:\tMSE: {:.2f}\ta: {:.2f}\tb: {:.2f}'.format(mse, a, b))

```

```
# 05_04
# 最急降下法
a = 0
b = 0
mse = sum((y[i] - (a + b*x[i]))**2 for i in range(N))/N
mse_list = [mse]
while 1:
    new_a = a - lr * sum(x[i]*((b + a*x[i])- y[i]) for i in range(N))/N
    new_b = b - lr * sum((b + a*x[i])- y[i] for i in range(N))/N
    a = new_a
    b = new_b
    new_mse = sum((y[i] - (b + a*x[i]))**2 for i in range(N))/N
    mse_list.append(new_mse)
    if (new_mse - mse)**2 < epsilon**2:
        break
    else:
        mse = new_mse

count = [i+1 for i in range(len(mse_list))]
print('04:\tMSE: {:.2f}\ta: {:.2f}\tb: {:.2f}'.format(new_mse, a, b))
plt.figure(2).clf()
plt.scatter(count,mse_list, label='04')

# 05_05
# 行列で最急降下法
#行列化
X = []
for _ in range(N):
    X.append([1,x[_]])
X = np.matrix(X)
Y = np.matrix([y]).T
w = np.matrix([0,0]).T

mse = sum((Y[i,0] - np.dot(X[i],w)[0,0])**2 for i in range(N))/N
mse_list = [mse]
while 1:
    w = w - lr * np.dot(X.T,(np.dot(X,w)-Y))/N
    new_mse = sum((Y[i,0] - np.dot(X[i],w)[0,0])**2 for i in range(N))/N
    mse_list.append(new_mse)
    if (new_mse - mse)**2 < epsilon**2:
        break
    else:
        mse = new_mse

count = [i+1 for i in range(len(mse_list))]
print('05:\tMSE: {:.2f}\ta: {:.2f}\tb: {:.2f}'.format(new_mse, w[1,0], w[0,0]))
```

```
85      plt.figure(2)
86      plt.scatter(count,mse_list, label='05')
87      plt.xlabel('Iterations')
88      plt.ylabel('MSE')
89      plt.legend()
90
91      # 05_06
92      # 直接求める
93      w = np.dot(np.linalg.inv(np.dot(X.T,X)),np.dot(X.T,Y))
94      mse = sum((Y[i,0] - np.dot(X[i],w)[0,0])**2 for i in range(N))/N
95      print('06:\tMSE: {:.2f}\ta: {:.2f}\tb: {:.2f}'.format(mse, w[1,0], w[0,0]))
96
97      # 05_07
98      # polyfitで求める
99      a, b =np.polyfit(x, y, 1)
100     mse = sum((y[i] - a*x[i] - b)**2 for i in range(N))/N
101     print('07:\tMSE: {:.2f}\ta: {:.2f}\tb: {:.2f}'.format(mse, a, b))
102
103     # 05_08
104     df_male = df[df["Gender"] == "Male"]
105     df_female = df[df["Gender"] == "Female"]
106
107     x_male = df_male["Weight"].tolist()
108     x_female = df_female["Weight"].tolist()
109     y_male = df_male["Height"].tolist()
110     y_female = df_female["Height"].tolist()
111
112     x_m_ave = np.mean(x_male)
113     x_m_var = np.sqrt(np.var(x_male))
114     y_m_ave = np.mean(y_male)
115     y_m_var = np.sqrt(np.var(y_male))
116     x_f_ave = np.mean(x_female)
117     x_f_var = np.sqrt(np.var(x_female))
118     y_f_ave = np.mean(y_female)
119     y_f_var = np.sqrt(np.var(y_female))
120
121     pvalue_x = scipy.stats.ttest_ind(x_male, x_female)[1]
122     pvalue_y = scipy.stats.ttest_ind(y_male, y_female)[1]
123
124     plt.figure(3).clf()
125     plt.subplot(121)
126     plt.bar([0, 1], [x_m_ave,x_f_ave], yerr=[x_m_var, x_f_var])
127     plt.ylabel('Height')
128     plt.xticks([0, 1], ['Male', 'Female'])
129
130     plt.subplot(122)
```

```
131         plt.bar([0, 1], [y_m_ave,y_f_ave], yerr=[y_m_var, y_f_var])
132         plt.ylabel('Weight')
133         plt.xticks([0, 1], ['Male', 'Female'])
134         plt.tight_layout()
135         plt.show()
136 
137         if pvalue_x > .95:
138             print('Female is higher than Male (p={:.3f})'.format(1 - pvalue_x))
139         elif pvalue_x < .05:
140             print('Male is higher than Female (p={:.3f})'.format(pvalue_x))
141         else:
142             print('There is no significant difference between Male and Female in height (p
                    ={:.3f})'.format(pvalue_x))
143 
144         if pvalue_y > .95:
145             print('Female is heavier than Male (p={:.3f})'.format(1 - pvalue_y))
146         elif pvalue_y < .05:
147             print('Male is heavier than Female (p={:.3f})'.format(pvalue_y))
148         else:
149             print('There is no significant difference between Male and Female in weight (p
                    ={:.3f})'.format(pvalue_y))
```

平均と分散を求める関数をそれぞれ作成し、mein 関数から呼び出す方式をとっている。また、モジュールを使用して計算するパートでは numpy を使用している。

出力結果

出力結果を list2 に示す。

list 2　output

```
1   Averaged standardized w: 0.00+-1.00
2   Averaged standardized h: -0.00+-1.00
3   03: MSE: 0.13 a: 0.94 b: -0.00
4   04: MSE: 0.13 a: 0.92 b: -0.00
5   05: MSE: 0.13 a: 0.92 b: -0.00
6   06: MSE: 0.13 a: 0.93 b: -0.00
7   07: MSE: 0.13 a: 0.93 b: -0.00
8   Male is higher than Female (p=0.000)
9   Male is heavier than Female (p=0.000)
```

作成した関数の出力とモジュールを使用したパートの出力がほぼ一致しているため、正しく計算できていることがわかる。

課題 1 で作成した散布図を図 1 に示す。

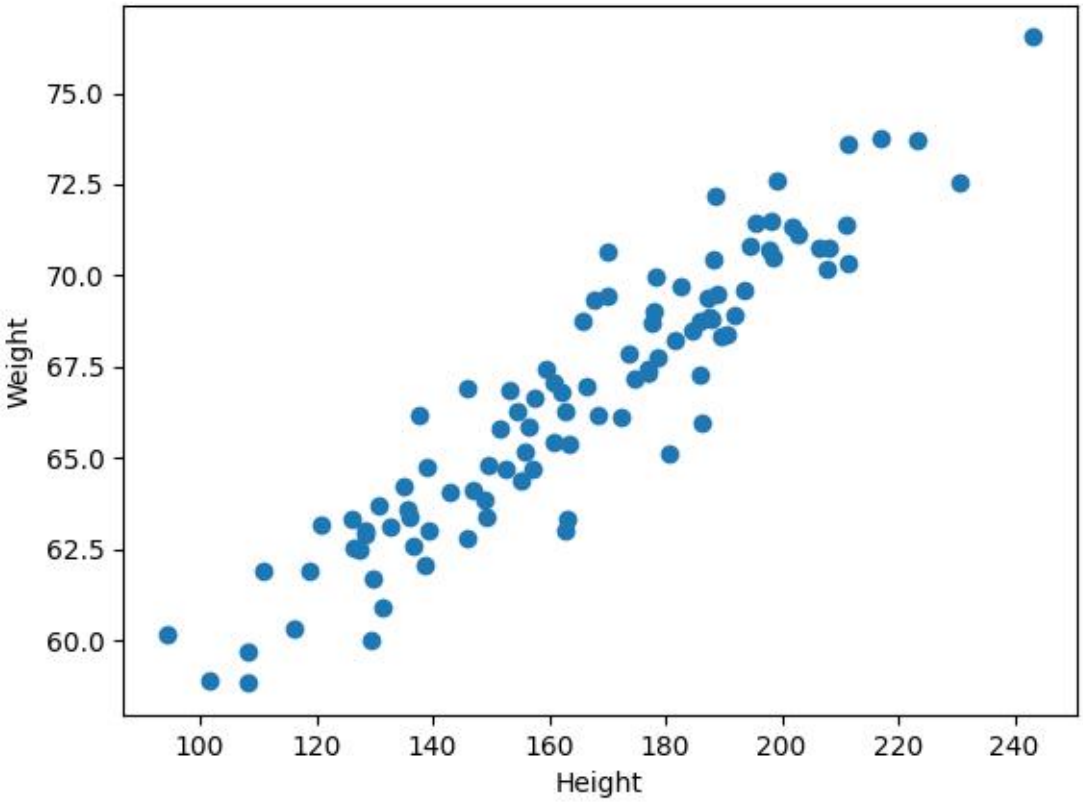

図 1 Figure1

課題 4,5 で作成した散布図を図 2 に示す。

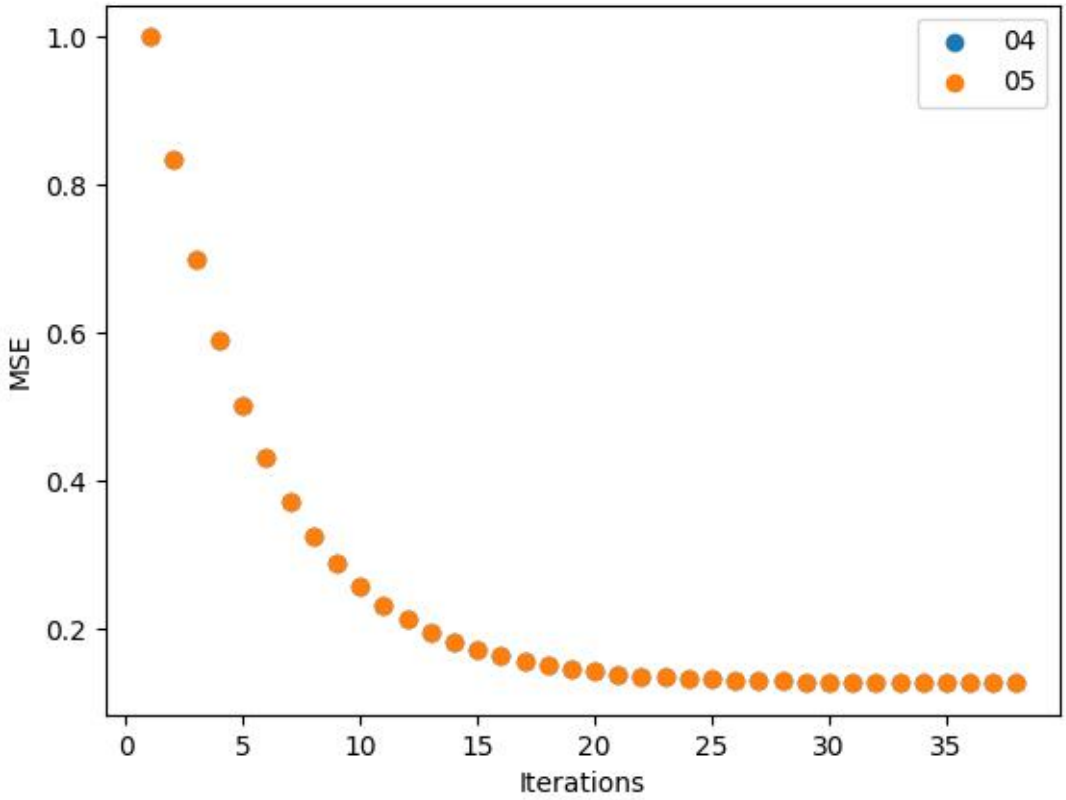

図 2 Figure2

課題 8 で作成した散布図を図 3 に示す。

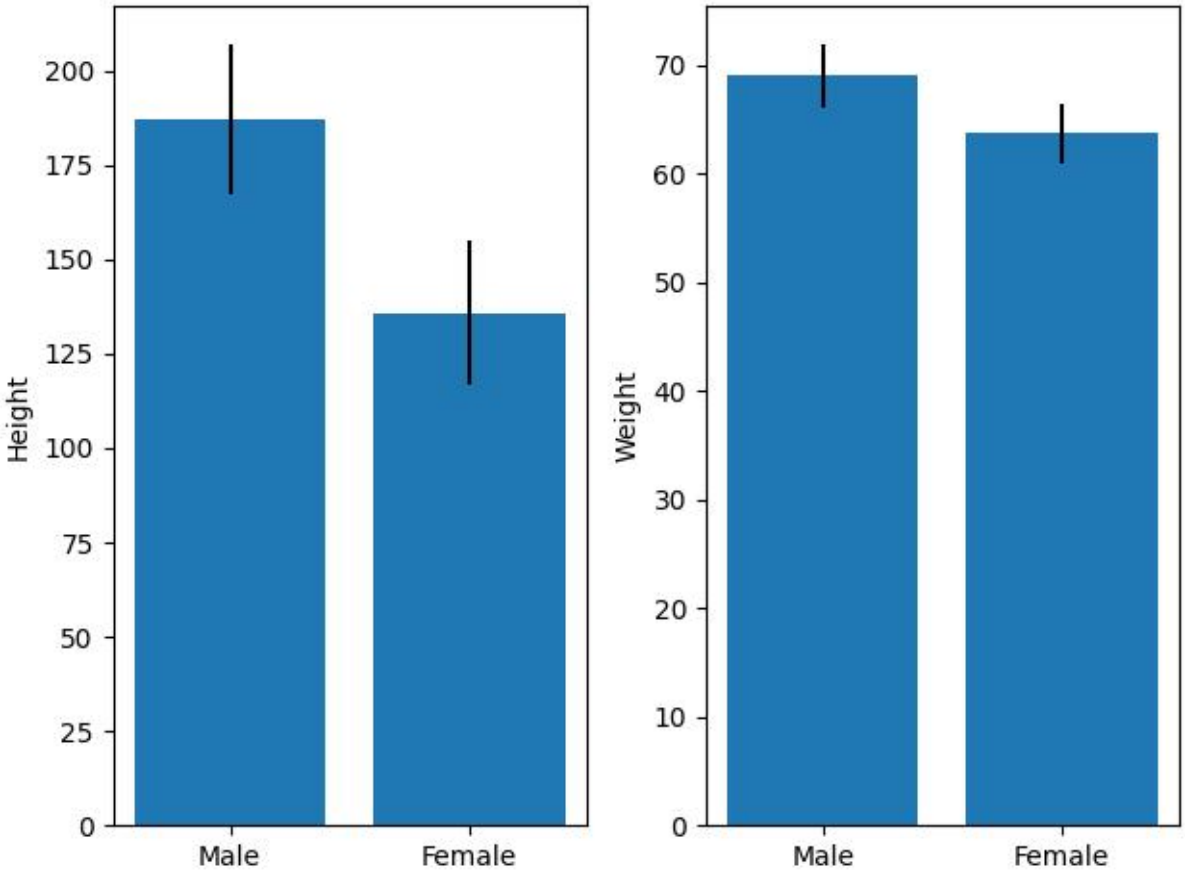

図 3 Figure3

工夫した箇所

今回課題ごとにプログラムを作成し、動作確認をしてから一つのプログラムに統合するという手段をとった。そのため、各プログラムでデバッグがしやすかった。しかし、モジュール化を行っていないため、すごく長いプログラムとなってしまった。