科目名：プログラミング技法 II　　担当教員名：新田　直子
学籍番号：01Z18032　　名前：トゥチェク・トム

2019 年 06 月 06 日

平均，分散，標準偏差、場所間の距離、線形回帰

課題２：平均，分散，標準偏差

N（＝25）個のランダムな、－100 以上 100 未満の値でリストを作った。平均は、合計を要素数で割ったものである。分散は、個々の値と平均との違い二乗、その合計をまた要素数で割ったものである。標準偏差は、分散の平方根となる。

それらの方法でできた結果をモジュール statistics の関数の結果と一緒に出力した。平均と分散には少し違う結果があっても、あまり有意はないと思う。標準偏差の有意な違いの原因は、平方根を計算する関数である。

課題２：実行結果表

自分で作った関数	モジュール statistics の関数
Mean:　15.332881970289058	15.332881970289058
Variance:　3320.9053586176756	3320.905358617675
Standard deviation:　57.627300000789496	58.81561370271286
Mean:　9.354598797483284	9.354598797483286
Variance:　3961.221435390986	3961.221435390985
Standard deviation:　62.9382000009658	64.23606719384583
Mean:　-4.300679735644522	-4.300679735644522
Variance:　2124.5622577029017	2124.5622577029017
Standard deviation:　46.092900000406594	47.04344465605372

課題 3: 場所間の距離

距離を計算する関数は、式の通り、モジュール math を使用しながら実装した。各地のデータをファイルから読み込み、辞書に変換した。まず、辞書を全部もラジアンに変換したが、それは無駄なので、後で選択された地点だけを変換した。その地点は名前の入力で選択される。入力された名前の一つも辞書にはない場合、エラーメッセージを出力する。両方の名前がある場合、距離を作った関数で計算し、出力する。

課題３：実行結果表

入力 1	入力 2	出力
Osaka	Tokyo	distance: 403.60244096985537 km
Toono	Nagoya	distance: 580.1441848848292 km
Suita	Toyonaka	distance: 4.840757283316017 km
Hokkaido	Okinawa	distance: 2208.1834787207986 km
Vienna		Error: Could not find Vienna

課題 4 : 線形回帰

課題 4-1

データを読み込み、男性のデータをリストへ、zip と random.sample で、N＝100 個をランダムに抽出した。身長、体重をそれぞれのリストに変換し戻った。それらのリストで、プロットを出力する。

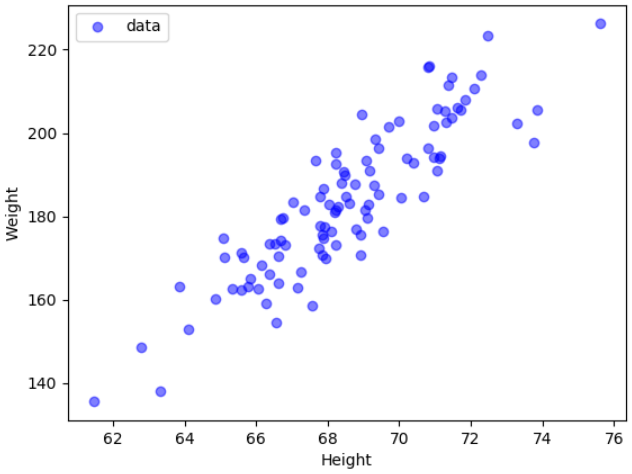

図 1　課題 4-1 の結果

課題 4-2

標準化する関数を作成した。個々の値マイナス平均、標準偏差で割ると、標準化されたデータを得る。出力されたプロットは大体、前回と同じ見えるが、値が（0,1）の正規分布に従う。

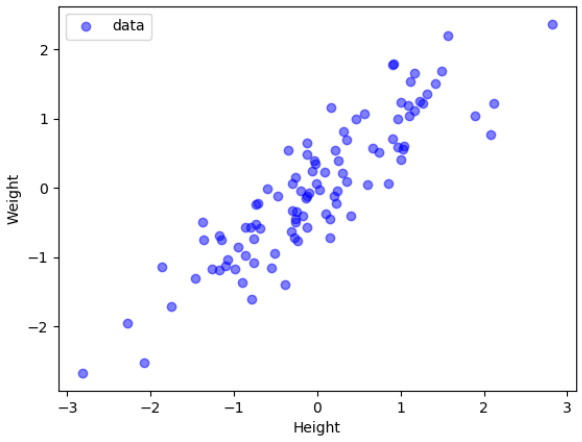

図 2　課題 4-2 の結果

課題 4-3

MSE の関数を式の通り実装した。

a	b	出力
2	2	5.794538806970911
0	0	0.9900000000000003
1	1	1.251836840620225
10	10	183.17735672606767
-10	-10	216.93241480351992

課題 4-4

df/da の関数は fda という、df/db の関数は fdb というになった。それで a と b の更新を、限界を超えること（max_iterations = 1000）または、更新は十分低くなること（precision = 0.005）まで繰り返す。そうすると、関数から a、b と MSE のデータを得る。それぞれ、グラフに示す。

初期値を変化すると、MSE の最初の値が強く変える。それは図 3 と図 4 に示したデータからわかる。また、学習率は繰り返しの数に関係がある。学習率は高くすると、もっと早く終わるが、結果の正確さが低くなる。

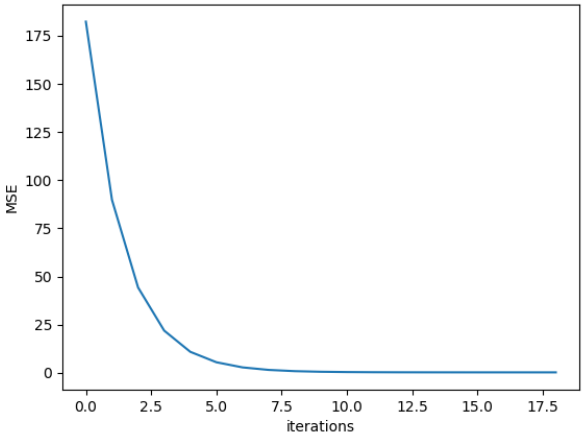

図 3　初期値は 10、学習率は 0.3 の MSE グラフ

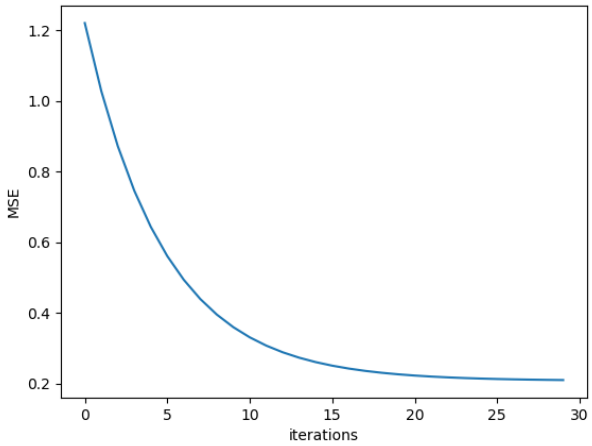

図 4　初期値は 1、学習率は 0.1 の MSE グラフ

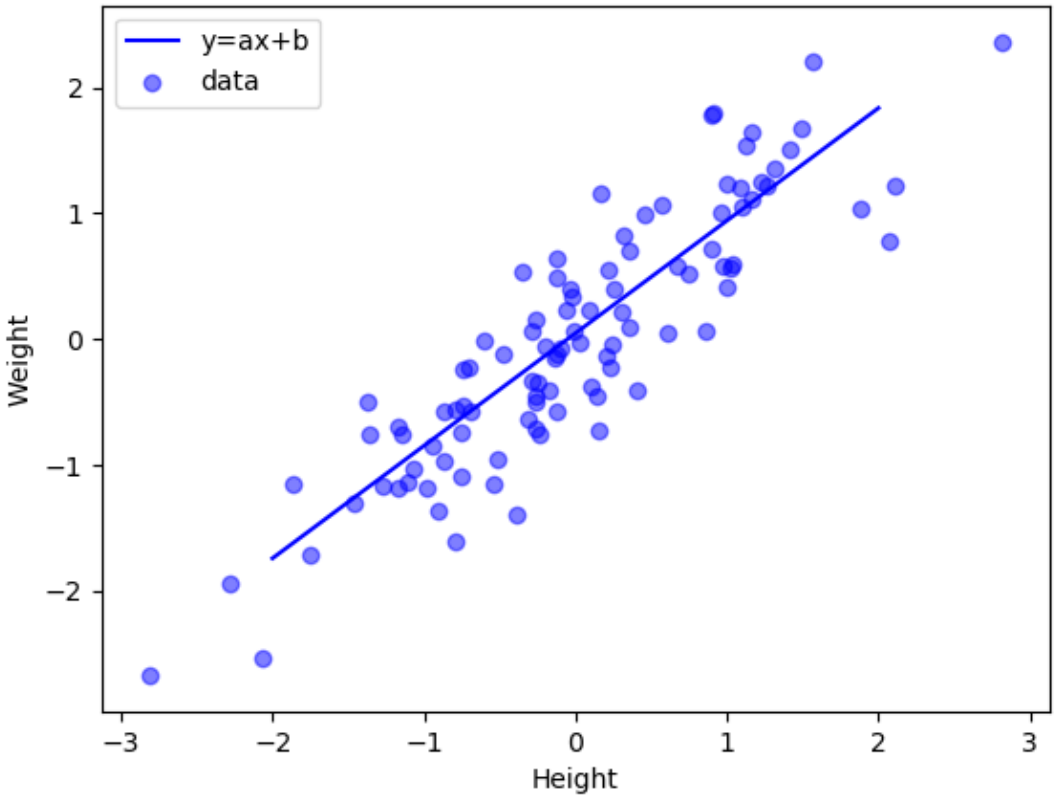

図 5　データとともに a、b を示すグラフ

課題 4-5

この課題は私にとって一番難しかった。データを Numpy の行列に変換するのは、どうするかわからなかったが、先生が手伝ってくださったおかげで、なんとかできた。それで、更新の関数を変化して、課題 4－4 と同じような結果が得る。

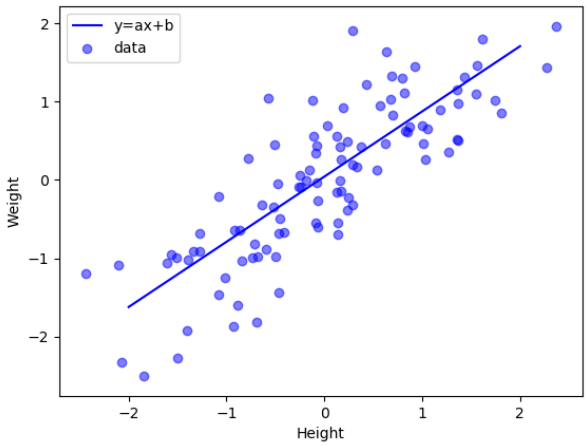

図 6　課題 4－5 の結果

課題 4-6

簡単な公式で、前回と同じような結果を得る。。

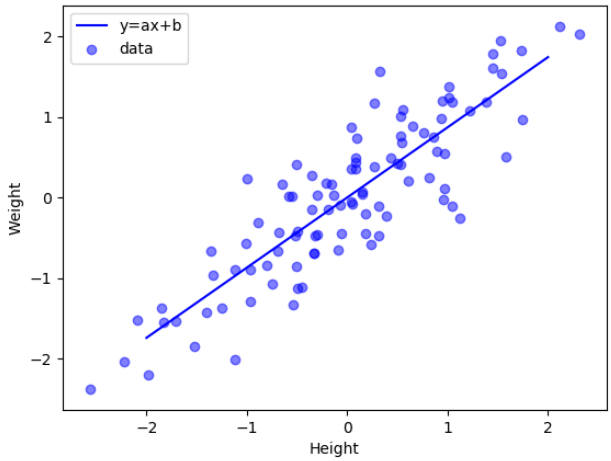

図 7　課題 4－6 の結果

課題 4-7

np.polyfit でまた、同じような結果を得る。polyfit の関数には、普通の身長・体重の配列が必要だ。動作検証は下記、4－8 のところでする。

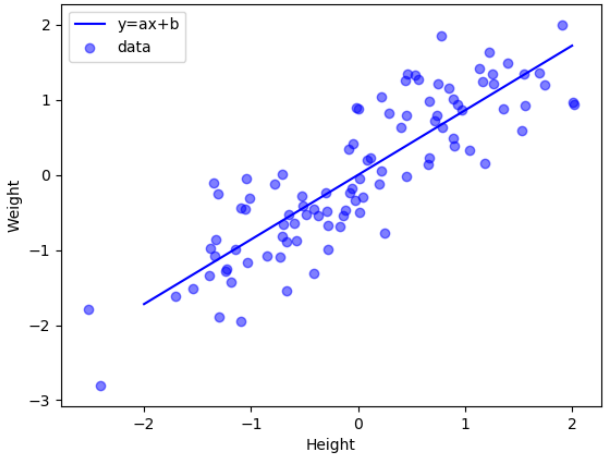

図 8　課題 4－7 の結果

課題 4-8

それぞれの関数の結果を一緒に示すと（図 9）、小さな違いが見えるが、それはあまり有意ではないと思う。それは、標準化後だ。標準化前、課題 4－4 と課題 4－5 の関数は無限に関する問題が起こるので、結果が出られない。それらのアルゴリズムには、標準化された値が必要であるようだ。

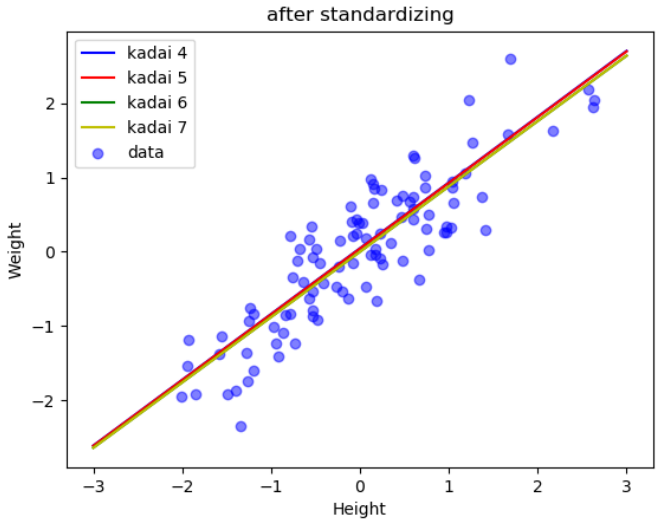

図 9　それぞれの関数の結果の比較（標準化後）

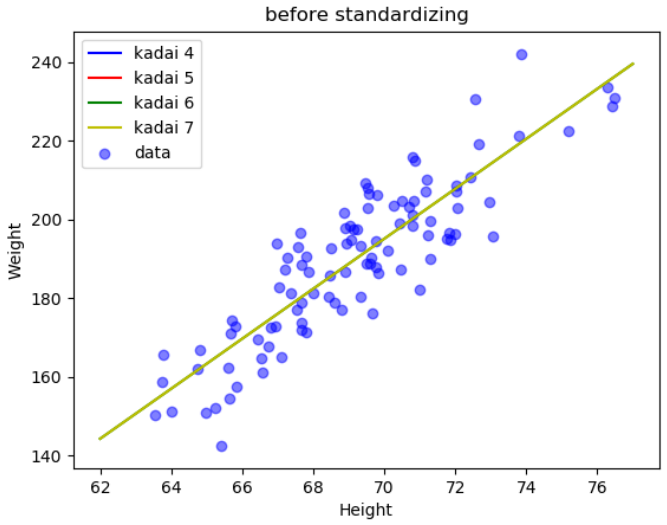

図 10　それぞれの関数の結果の比較（標準化前）