

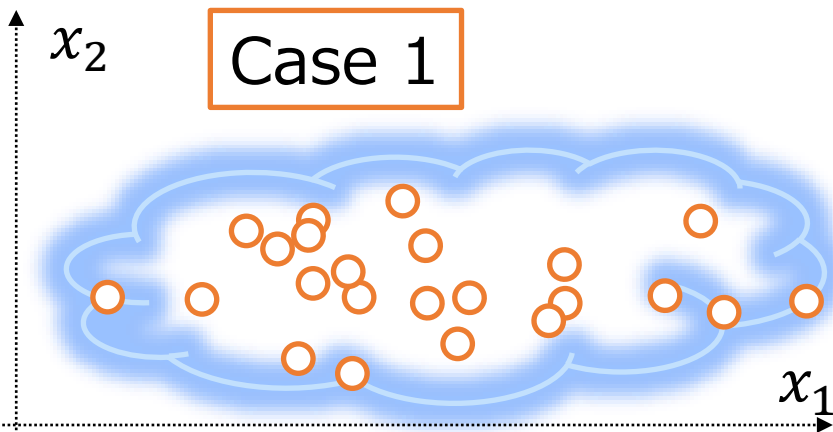
「相関分析と回帰分析」

九州大学 大学院システム情報科学研究所
情報知能工学部門
データサイエンス実践特別講座
末廣大貴, Diego Thomas, 正井克俊

相関分析

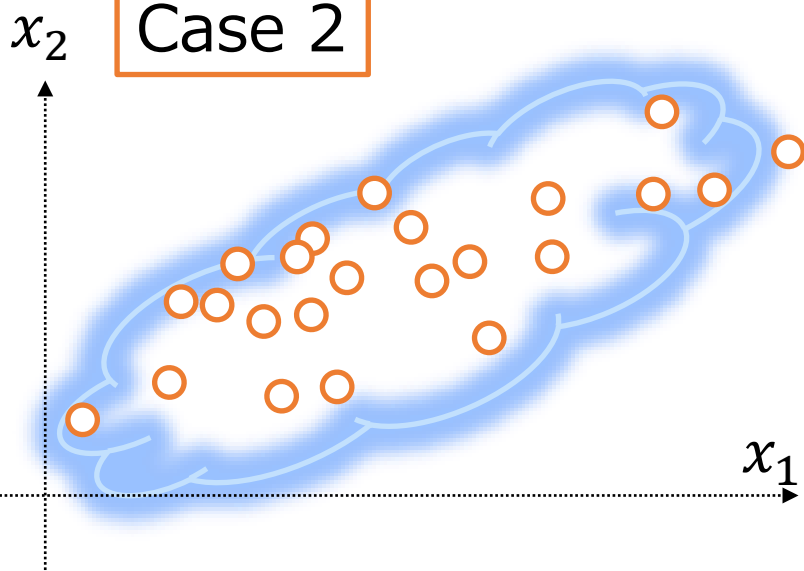
データの広がり方(=分散)に潜む関係～**相関**

Case 1

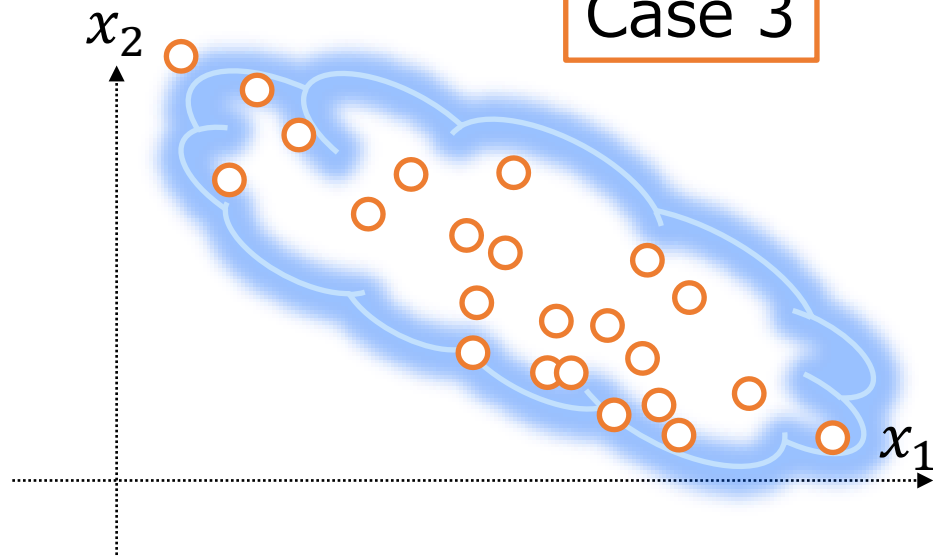


身長と体重は？
身長と数学の点数は？
身長とバレーボール攻撃失敗率は？

Case 2



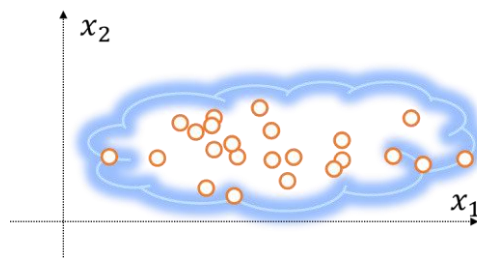
Case 3



データの広がり方(=分散)に潜む関係～相関

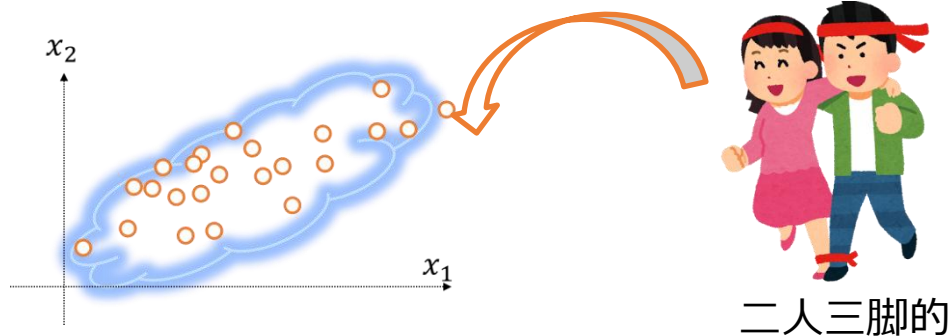
● Case 1: 無相関

- $x_1 \rightarrow$ 大, $x_2 \rightarrow$ 特段の傾向無し
- 要するに, x_1 と x_2 は無関係
- 身長と数学の点数



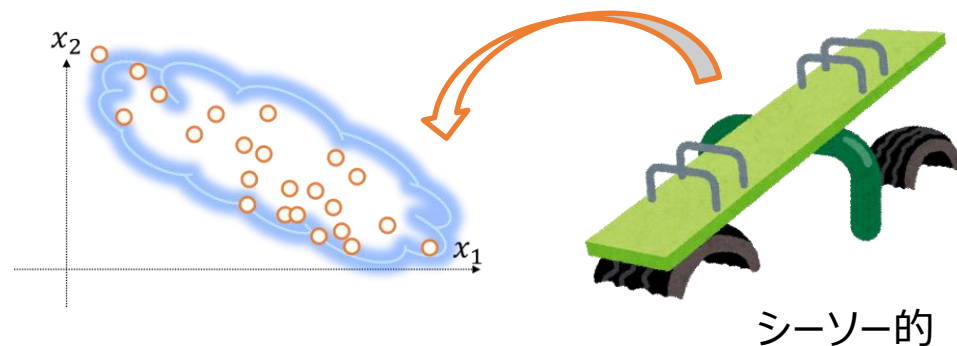
● Case 2: 正の相関

- $x_1 \rightarrow$ 大, $x_2 \rightarrow$ 大
- 身長と体重



● Case 3: 負の相関

- $x_1 \rightarrow$ 大, $x_2 \rightarrow$ 小
- 身長とバレーボール攻撃失敗率



相関係数 ρ ～相関の定量化 (1/3)

- 簡単のために x_1 も x_2 も平均ゼロとする
 - = x_1 と x_2 を平均0になるように分布をシフトしただけ
- この時, 相関を(簡単な言葉で)定義すると...
 - 相関の強さを表す指標 ($-1 \leq \rho \leq 1$)

$$\rho = \frac{(x_1 \times x_2) \text{の平均値}}{\sqrt{x_1 \text{の分散} \cdot x_2 \text{の分散}}}$$

分子が大事

分母は正規化の役目
(標準化)

相関

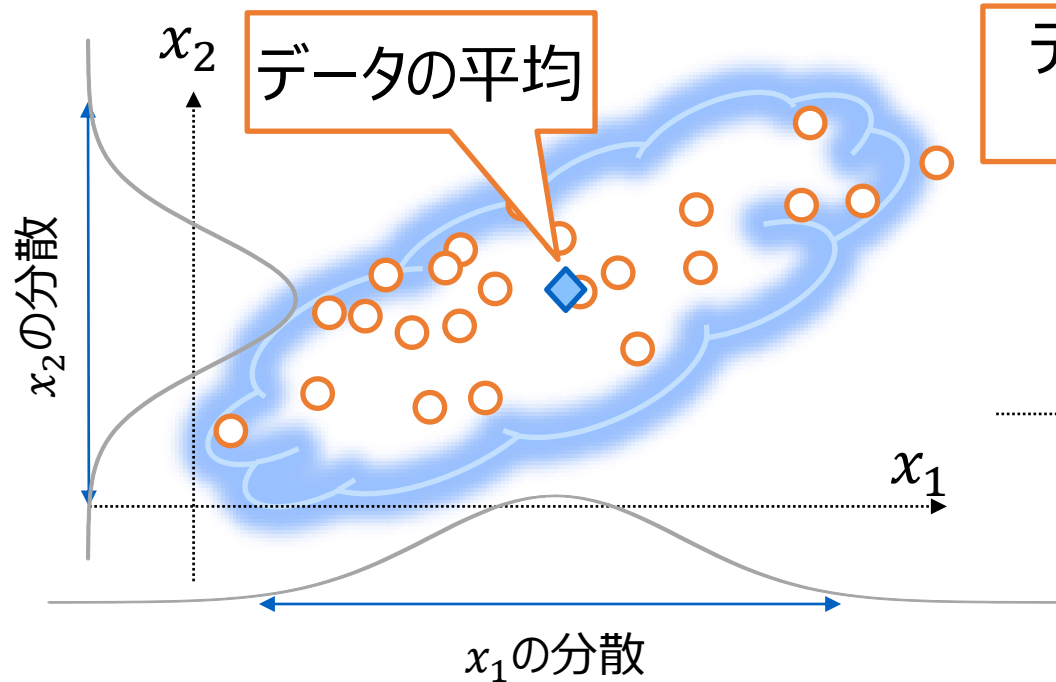
x_1 の平均

x_2 の平均

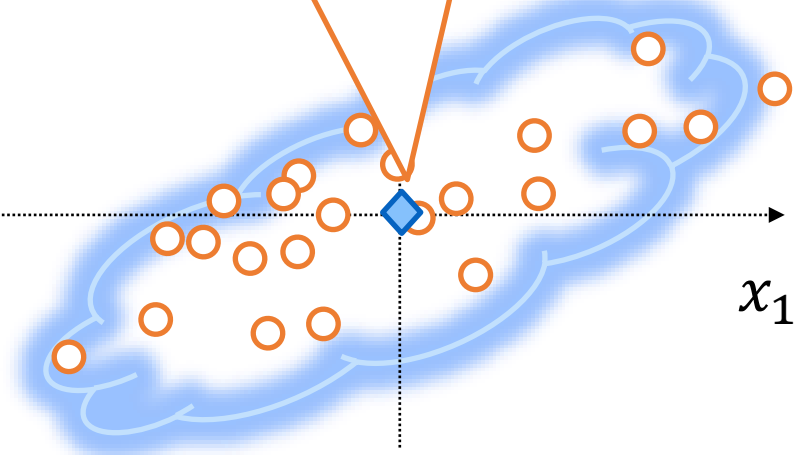
共分散

$$\rho = \frac{(x_1 - \bar{x}_1) \times (x_2 - \bar{x}_2) \text{の平均値}}{\sqrt{x_1 \text{の分散} \cdot x_2 \text{の分散}}}$$

$$\rho = \frac{(x_1 \times x_2) \text{の平均値}}{\sqrt{x_1 \text{の分散} \cdot x_2 \text{の分散}}}$$



データの平均が原点になるように移動



相関

分子部分

$$(x_1 \times x_2)$$

$$(+) \times (+) = (+)$$

$$(-) \times (-) = (+)$$

$$(+) \times (-) = (-)$$

$$(-) \times (+) = (-)$$

- 相関係数：高い

- (+) のデータ数 > (-) のデータ数

⇒ 平均は高くなる

- 相関係数：低い

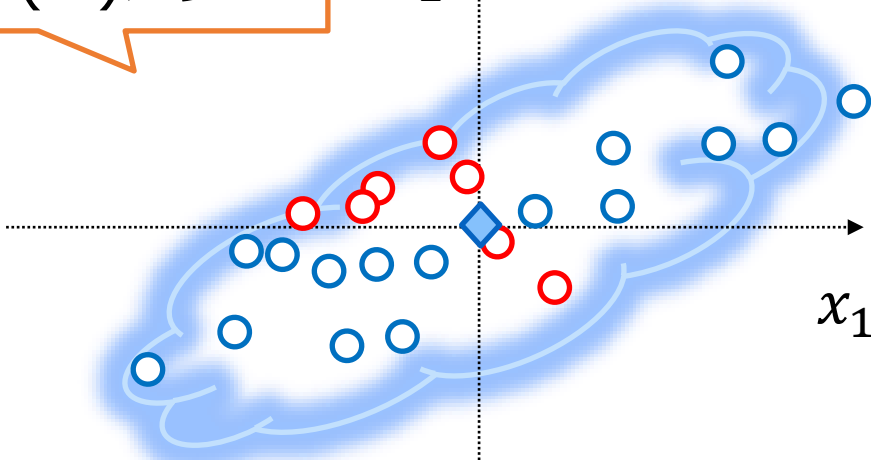
- (+) のデータ数 と (-,-) のデータ数 に差がない

- ランダム（無相関）ほど、どの領域に入るかがばらける

⇒ 平均は0に近づく

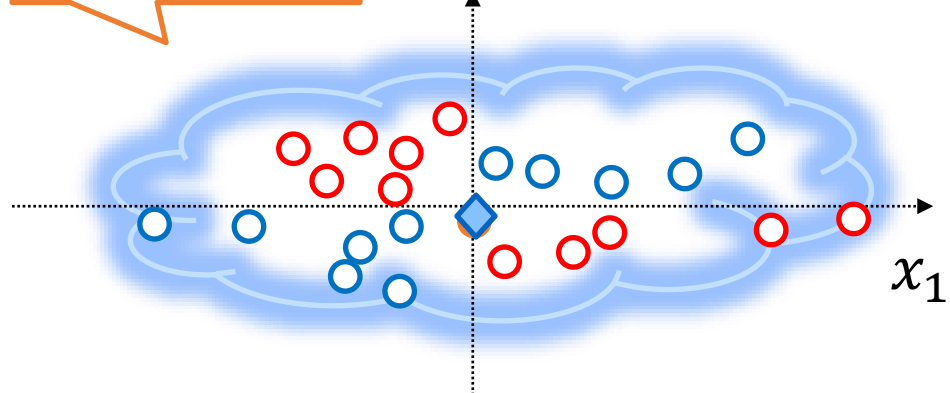
(+)が多い

x_2



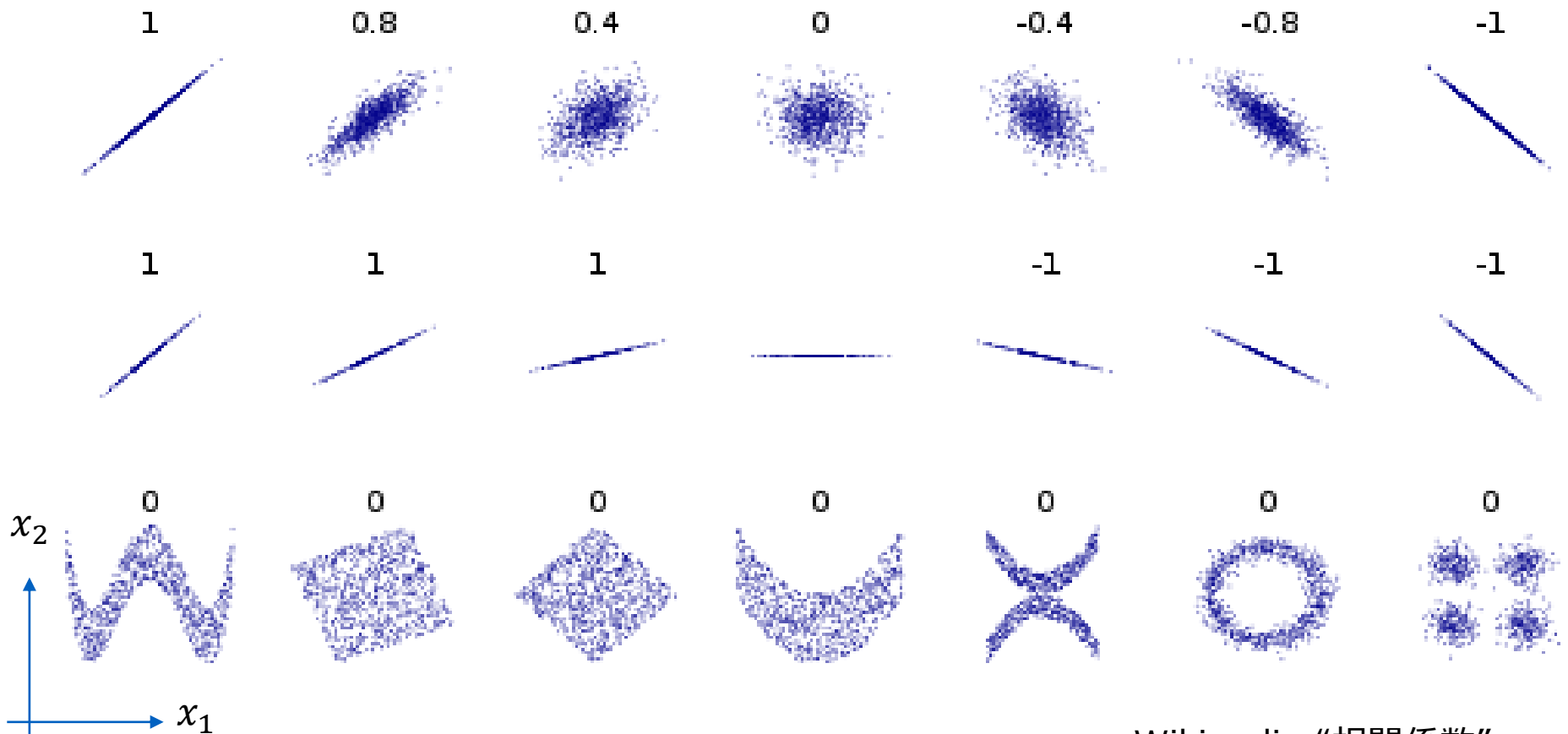
似たような数

x_2



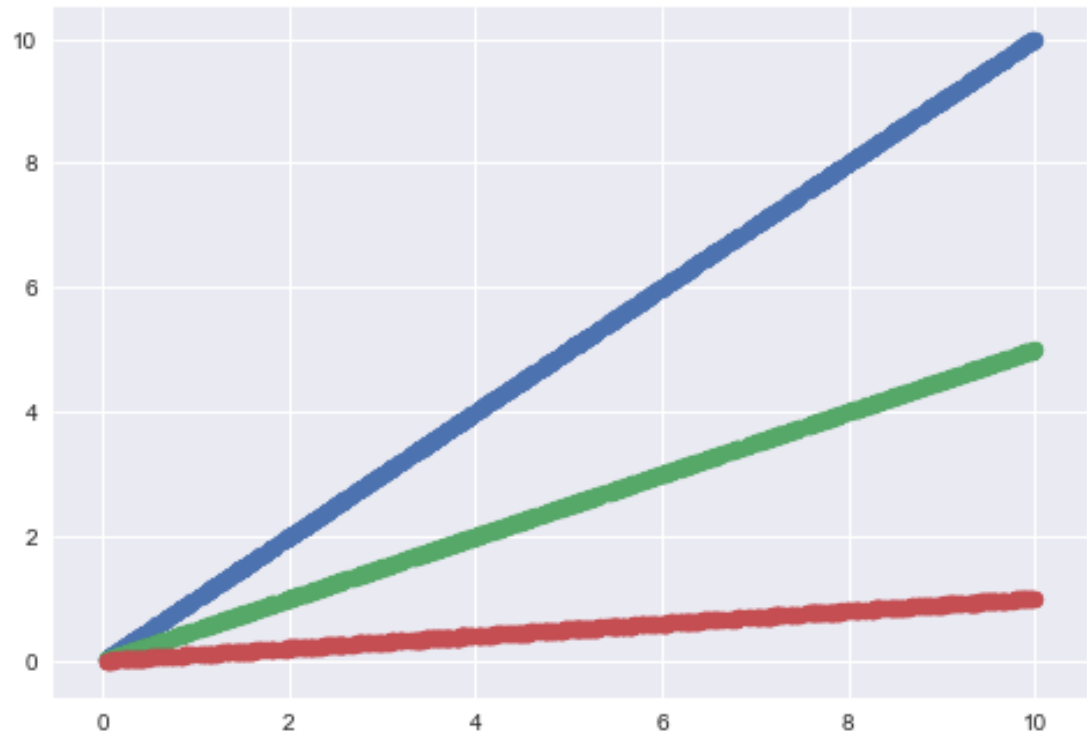
相関係数 ρ ～相関の定量化 (3/3)

- 相関係数 ρ がわかると、分布の形をある程度想像できる



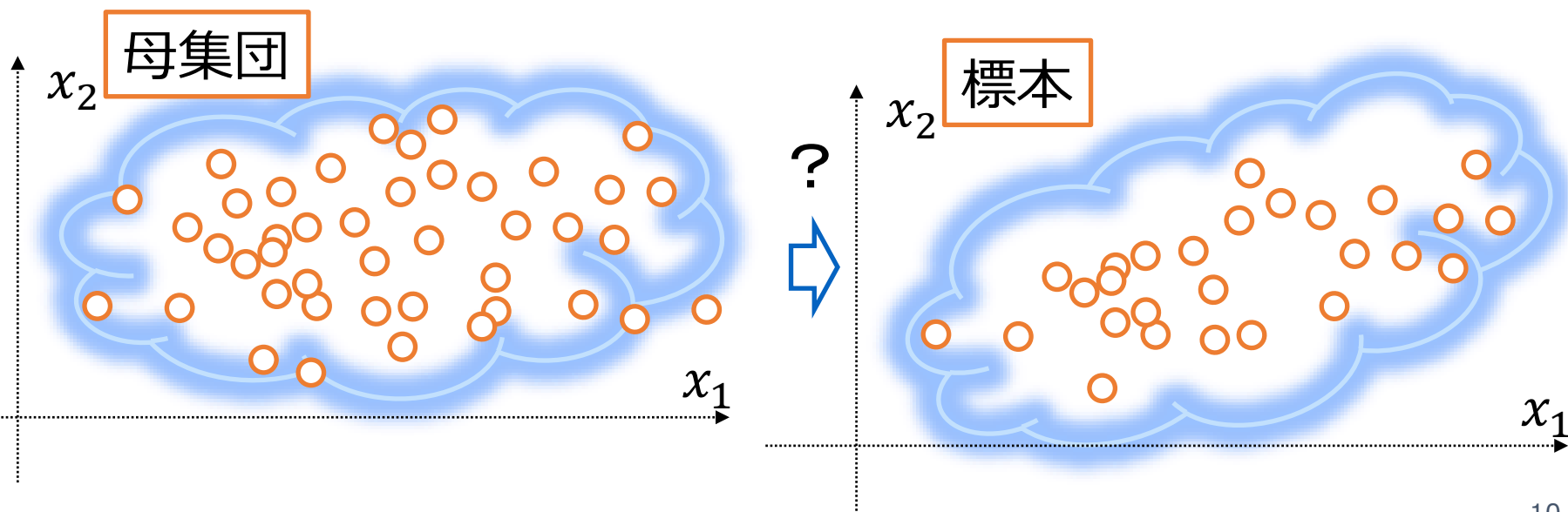
注意

- 相関は傾きは関係ない
 - 下記の3つの分布は、全て相関 1
- 外れ値に弱い



相関係数の検定

- 相関係数自体は、計算できるようになった
- 求めた相関係数は尤もらしいか？
- 母集団は無相関だけどたまたま標本時に相関が出てしまうサンプリングをしてしまったかも？ ⇒ 検定



相関係数の検定

- 相関係数が有意かどうかの検定
- 帰無仮説：母集団の相関係数が 0
- 相関係数の検定でもt分布を用いた検定ができます
(難解なので説明は省略)

相関分析

- XとYに相関があるか、ないかは以下で検定可能
- `stats.pearsonr(X,Y)` : 相関検定
- `stats.spearmanr(X, Y)` : 順位相関検定
1:思わない, 2:あまり思わない, 3:どちらでもない,
4:まあそう思う, 5:そう思う
などのアンケートが該当します

分布の可視化

- “height_weight.csv”を読み込んで、身長と体重の相関を見てみる
- まずは、散布図でデータの分布をみる

```
HW_data = pd.read_csv('./height_weight.csv')  
plt.scatter(HW_data['Ht'], HW_data['Wt']) # 分布をみる
```



相関係数

- “height_weight.csv”を読み込んで、身長と体重の相関を試みる
- まずは、散布図でデータの分布をみる

```
r,p = scipy.stats.pearsonr(HW_data['Ht'],HW_data['Wt'])  
print(r)  
print(p)
```

```
相関係数 r  0.7809062893282465  
p値        9.638647211605826e-43
```

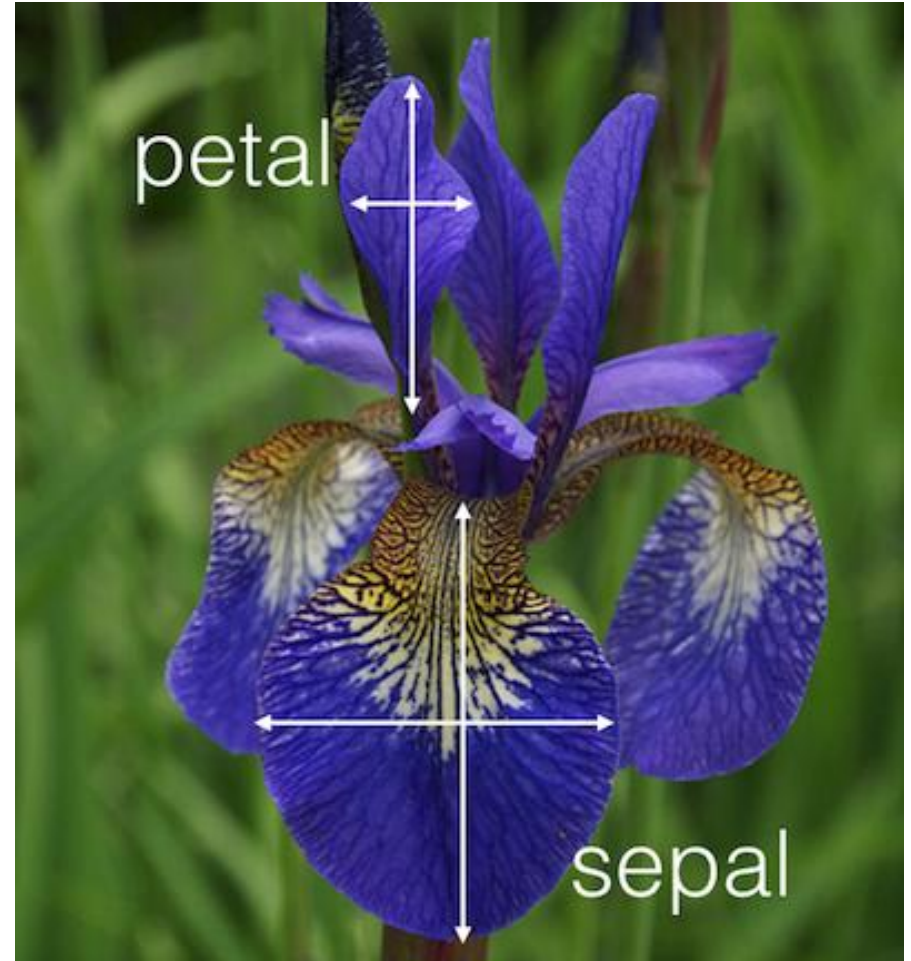
p値が0.01より小さいので、有意水準1%で相関係数は有意である

注意点

- 相関係数が高いからと言って有意差があるとは限らない
 - p値は相関係数と必ずしも関係がない
 - サンプルサイズにも大きく依存する（データ数が少なくても相関係数が大きくなることはある）
- 有意差があるからと言って相関係数が高いとは限らない
 - 相関の検定で有意差がある \Leftrightarrow 相関がないとは言えない
- 相関のあるなし（検定）と相関の度合い（相関係数）は別々に考えましょう

アイリスデータ

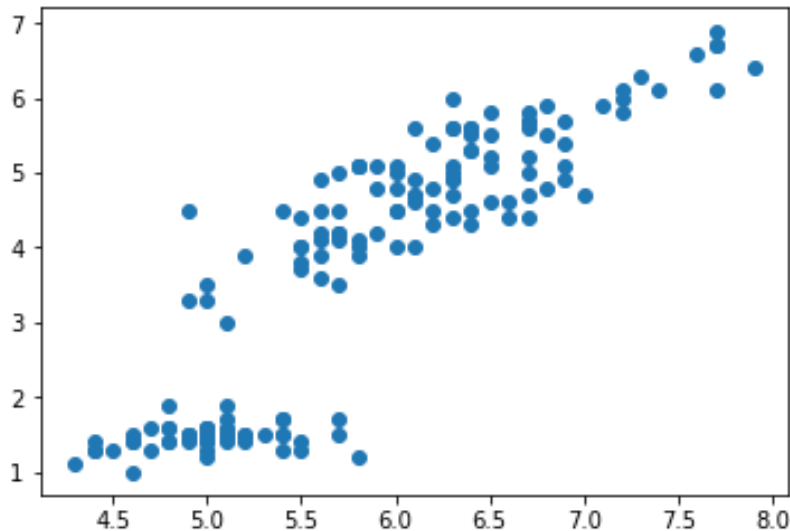
- アイリスデータ
- 3クラス
 - Iris-setosa
 - Iris-versicolor
 - Iris-virginica
- 指標
 - sepal length
 - sepal width
 - petal length
 - petal width



相関：練習

- sepal length と petal length に相関はあるか？

```
plt.scatter(iris.iloc[:,0], iris.iloc[:,2]) # 分布を見る  
r,p = scipy.stats.pearsonr(iris.iloc[:,0], iris.iloc[:,2])
```



相関係数 r 0.871282940672
p値 2.73506593965e-47

p値が0.01より小さく、有意水準1%で
相関係数は有意である

相関マップ：練習

- 項目の組み合わせで相関係数を求め、「相関マップ」で可視化してみる

```
# 相関マップ
```

```
R = np.corrcoef([iris.iloc[:,0], iris.iloc[:,1],  
                iris.iloc[:,2], iris.iloc[:,3]])
```

```
In [606]: R
```

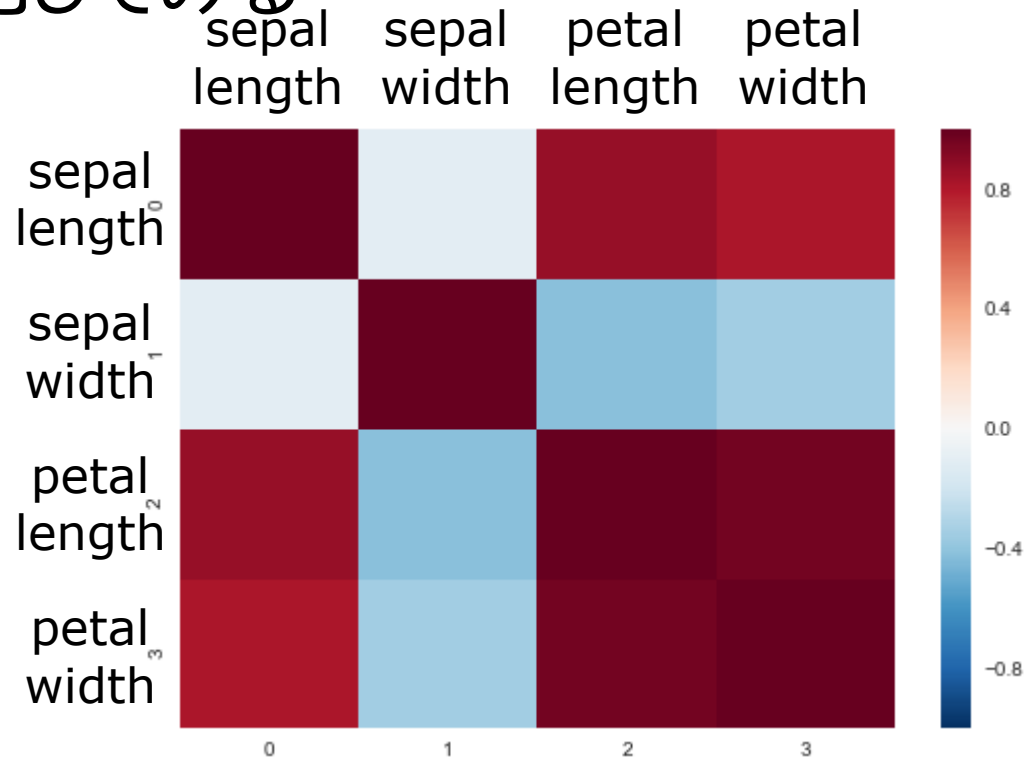
```
Out[606]:
```

```
[[ 1.          -0.10936925  0.87175416  0.81795363]  
 [-0.10936925  1.          -0.4205161  -0.35654409]  
 [ 0.87175416 -0.4205161   1.          0.9627571 ]  
 [ 0.81795363 -0.35654409  0.9627571   1.          ]]
```

相関マップ：練習

- 項目の組み合わせで相関係数を求め、「相関マップ」で可視化してみる

```
# 相関行列のヒートマップを描く
import seaborn as sns
sns.heatmap(R)
# グラフを表示する
plt.show()
```



練習1: 相関・相関マップ

- sepal width と petal width に相関があるか調べよう
- ヒント : `stats.pearsonr(X,Y)`

- 相関マップ中のどこに該当するか、値は一致するかを確認しよう

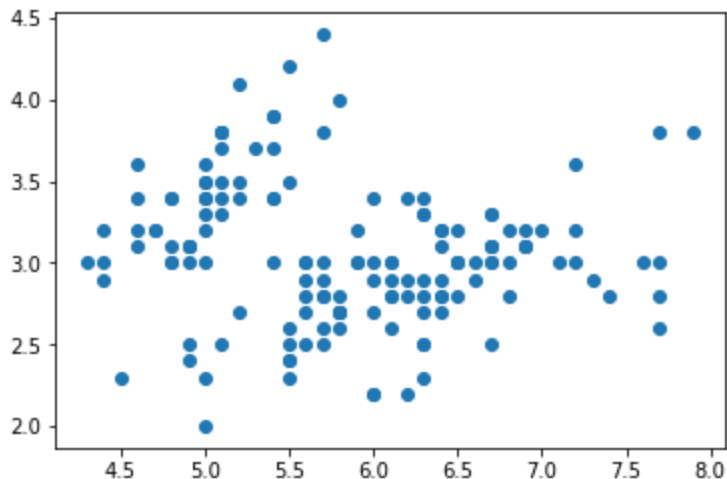
相関：練習

- sepal length と sepal width の相関を見よう！

```
# iris data 読み込み  
iris=pd.read_csv('./iris.csv')
```

```
# 分布を見よう！  
plt.scatter(iris.iloc[:,0], iris.iloc[:,1])
```

```
# sepal length と sepal width の相関係数 r と有意確率 p を求める  
r,p = scipy.stats.pearsonr(iris.iloc[:,0], iris.iloc[:,1])
```

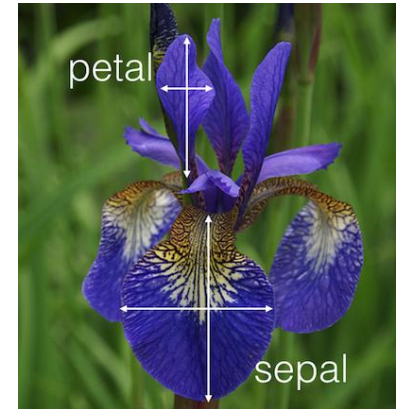
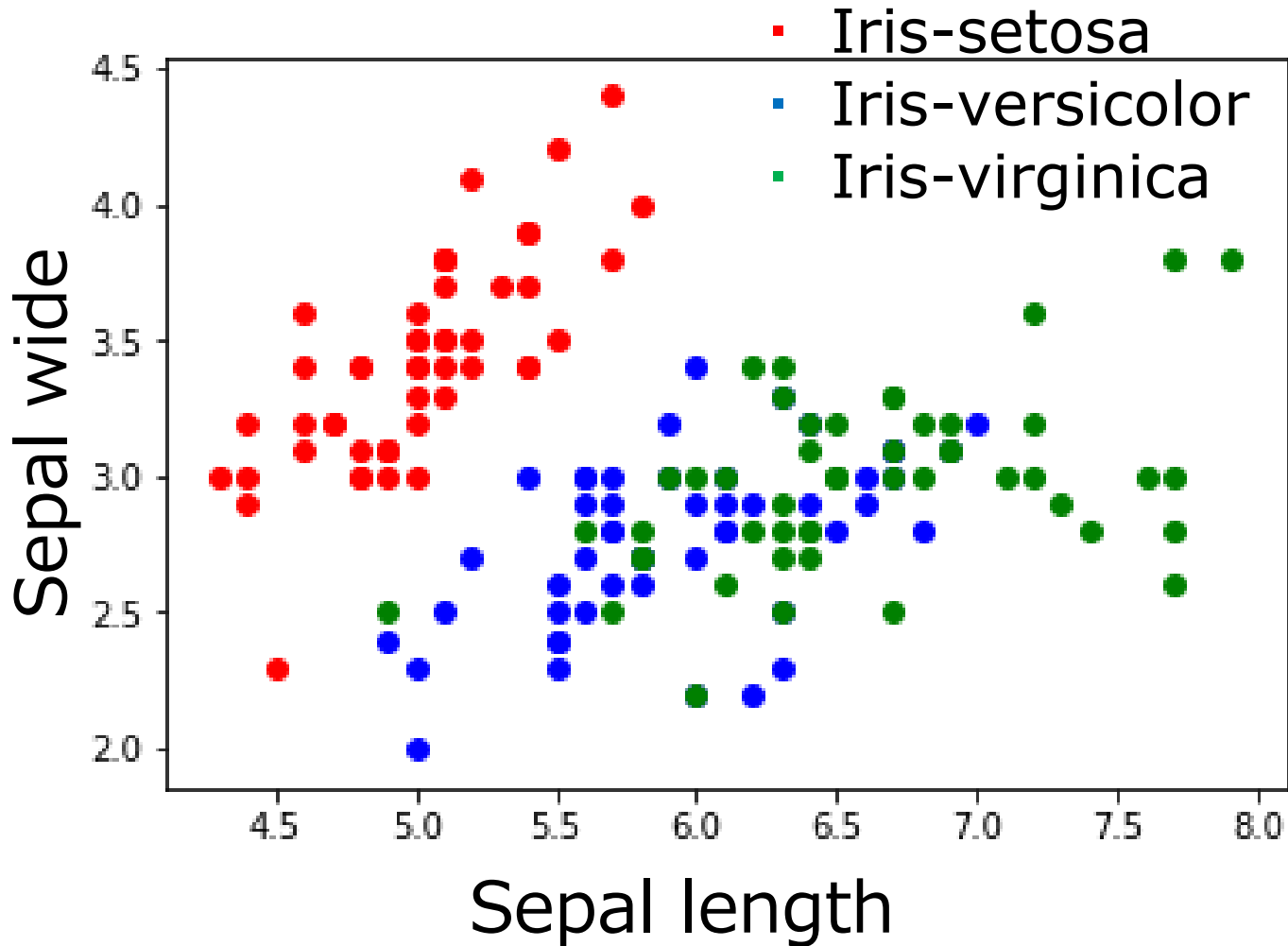


相関係数 r p 値
(-0.10378414795681816, 0.20782011930226704)

p 値が約0.2で、有意水準5%では
帰無仮説は棄却できない。

複数のクラスが混在しているから？

アイリスデータ解析



アイリスデータ解析：クラスごとに分けて分析

- クラスごとにデータに分けて、sepal length と sepal width の相関をみる

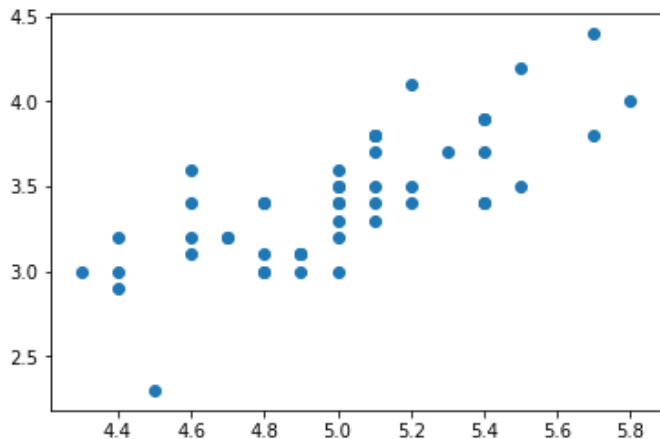
classごとにデータを分ける

```
irisSetosa = iris[iris.iloc[:,4]=='Iris-setosa']  
irisVersicolor = iris[iris.iloc[:,4]=='Iris-versicolor']  
irisVirginica = iris[iris.iloc[:,4]=='Iris-virginica']
```

アイリスデータ解析：相関（1）

- クラス「Iris-setosa」のデータだけに注目して、sepal length と sepal width の相関を見る

```
# setosa
plt.scatter(irisSetosa.iloc[:,0], irisSetosa.iloc[:,1]) # 分布を見る
r,p = scipy.stats.pearsonr(irisSetosa.iloc[:,0], irisSetosa.iloc[:,1])
print(r)
print(p)
```



相関係数 r 0.746498014199

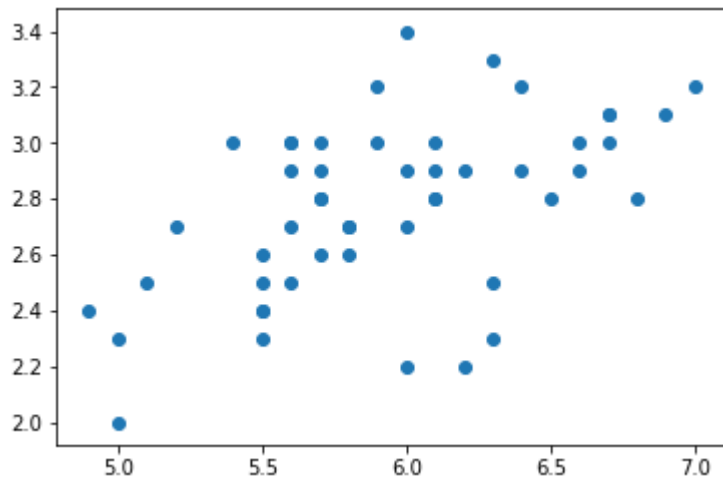
p値 7.38375131936e-10

p値が0.01より小さく、有意水準1%で相関係数は有意である

アイリスデータ解析：相関（2）

- クラス「Iris-versicolor」のデータだけに注目して、sepal length と sepal width の相関を見る

```
# versicolor
plt.scatter(irisVersicolor.iloc[:,0], irisVersicolor.iloc[:,1]) # 分布を見る
r,p = scipy.stats.pearsonr(irisVersicolor.iloc[:,0], irisVersicolor.iloc[:,1])
print(r)
print(p)
```



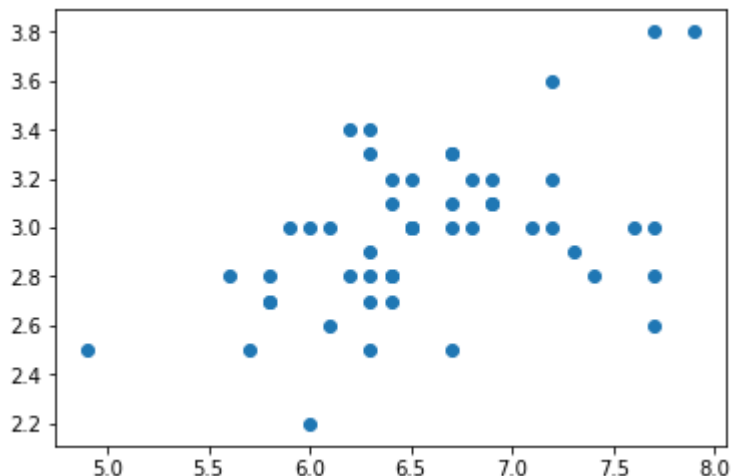
相関係数 r **0.5259107172828246**
p値 **8.771860011973842e-05**

p値が0.01より小さく、有意水準1%で相関係数は有意である

アイリスデータ解析：相関（3）

- クラス「Iris-virginica」のデータだけに注目して、sepal length と sepal width の相関を見る

```
# virginica
plt.scatter(irisVirginica.iloc[:,0], irisVirginica.iloc[:,1]) # 分布を見る
r,p = scipy.stats.pearsonr(irisVirginica.iloc[:,0], irisVirginica.iloc[:,1])
print(r)
print(p)
```



相関係数 r 0.457227816394

p値 0.000843462472371

p値が0.01より小さく、有意水準1%で相関係数は有意である

練習 2 : 相関

- Class ごとにデータを分類して、
`petal length` と `petal width` の相関を見よう！

```
# classごとにデータを分類
```

```
irisSetosa = iris[iris.iloc[:,4]=='Iris-setosa']
```

```
irisVersicolor = iris[iris.iloc[:,4]=='Iris-versicolor']
```

```
irisVirginica = iris[iris.iloc[:,4]=='Iris-virginica']
```

```
# classごとに petal length と petal width の相関係数 r と  
有意確率 p を求める
```

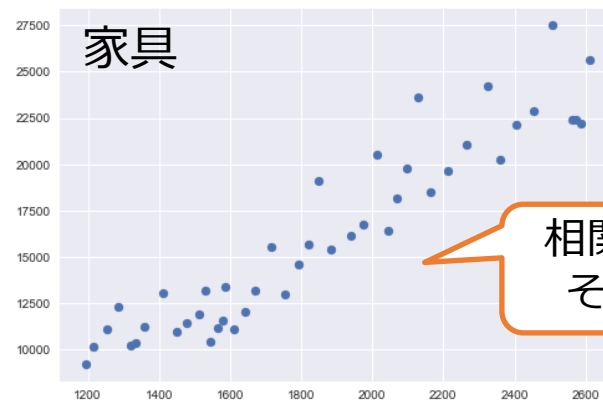
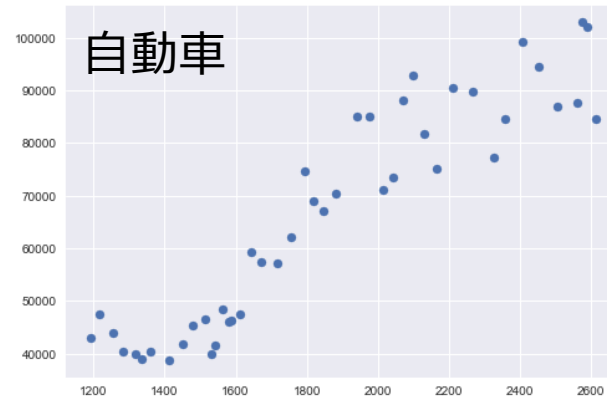
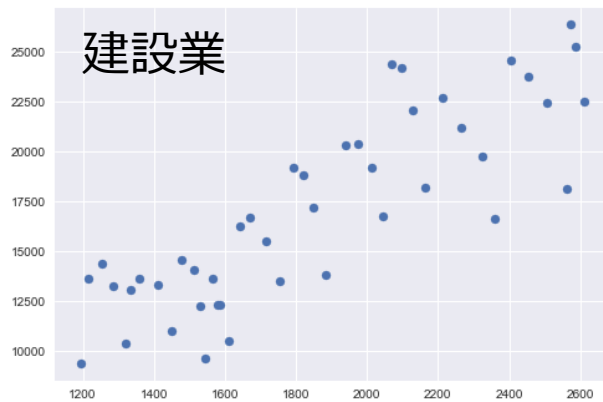
小売売上高のデータ

- “predicti.csv”：非農業の雇用、給料、給料支出額と4種類の小売業の11年間分の4半期販売データ

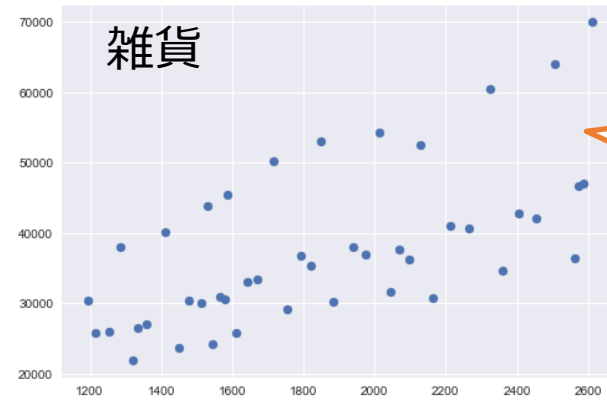
No	変数名	変数ラベル	説明	型
1	TIME	TIME	1979年の第1四半期から1989年の第4四半期までの四半期	id
2	WASA	WASA	国民所得と給料支出額 (10億\$)	numerical
3	EMPL	EMPL	非農業の企業の給料支払簿上の従業員数 (1,000)	numerical
4	BLDG	BLDG	建築資材の販売額 (100万\$)	numerical
5	AUTO	AUTO	自動車の販売額 (100万\$)	numerical
6	FURN	FURN	家具と家の装具の販売額 (100万\$)	numerical
7	GMER	GMER	雑貨の販売額 (100万\$)	numerical

小売売上高：分布可視化

- まずは、仮説を考えて、その項目の分布を見る
- 国民所得と各小売業の販売額は相関がありそう



相関高
そう



他より若干
ばらつきが

小売売上高：ヒートマップ

- まずは、相関行列のヒートマップを見て、どの項目同士で相関が高いか見てみよう！

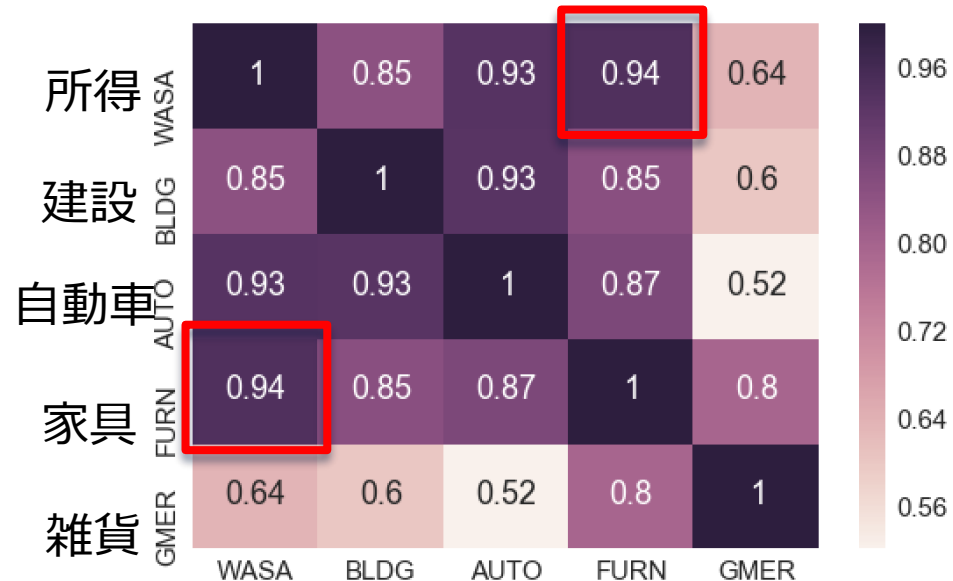
```
# 相関マップ
R = np.corrcoef([predicti['WASA'],
                 predicti['BLDG'], predicti['AUTO'],
                 predicti['FURN'],predicti['GMER']])
# 相関行列のヒートマップを描く
import seaborn as sns
sns.set(font_scale=1.5)
cr = ['WASA', 'BLDG', 'AUTO', 'FURN', 'GMER']
sns.heatmap(R, annot=True,xticklabels=cr,yticklabels=cr)
# グラフを表示する
plt.show()
```

小売売上高：最大相関

- 一番、相関が高いのは？
- 変数が多いときは以下のようなコードを書けば簡単に見つかる

```
# 最も相関が高い組み合わせを見つける
maxval = 0
maxind = [-1,-1]
for i in range(R.shape[0]):
    for j in range(R.shape[1]):
        if i == j:
            continue
        val = R[i,j]
        if val > maxval:
            maxval = val
            maxind = [i,j]

print maxval
print maxind
```



0.941942943693
[3, 0]

「所得」と「家具の売り上げ」の相関が高い

小売売上高：有意差

- 一番、高い相関を持つ項目で、相関検定もおこなって、有意性を確かめてみよう！

```
# 相関係数とp値を算出
```

「所得」

「家具の売り上げ」

```
r,p = scipy.stats.pearsonr(predicti['WASA'], predicti['FURN'])  
print(r)  
print(p)
```

相関係数 : 0.941942943693

p値 : 1.60835257649e-21

p値が0.01より小さく、有意水準1%で相関係数は有意である

回歸分析

回帰分析とは？

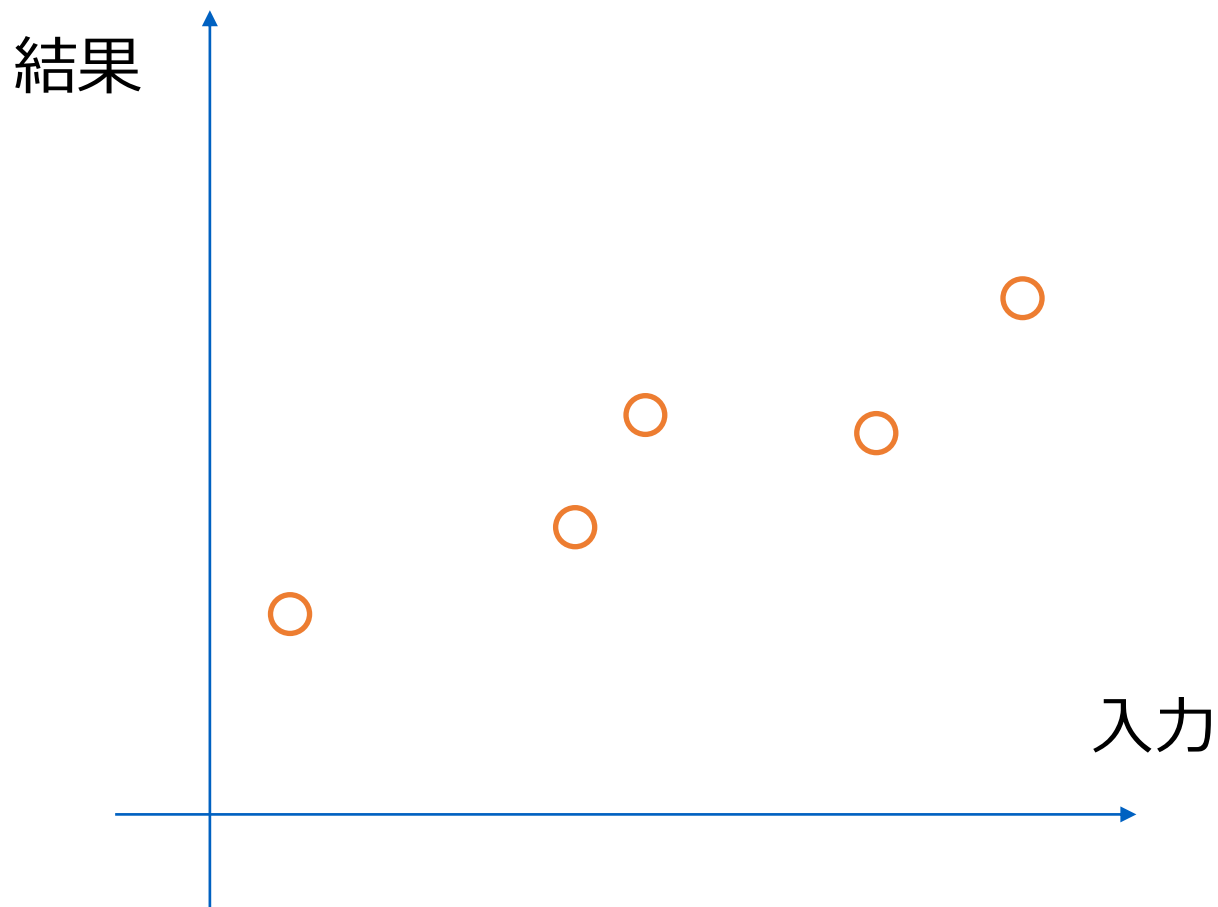
- 「入力に対する結果を推定する」手法



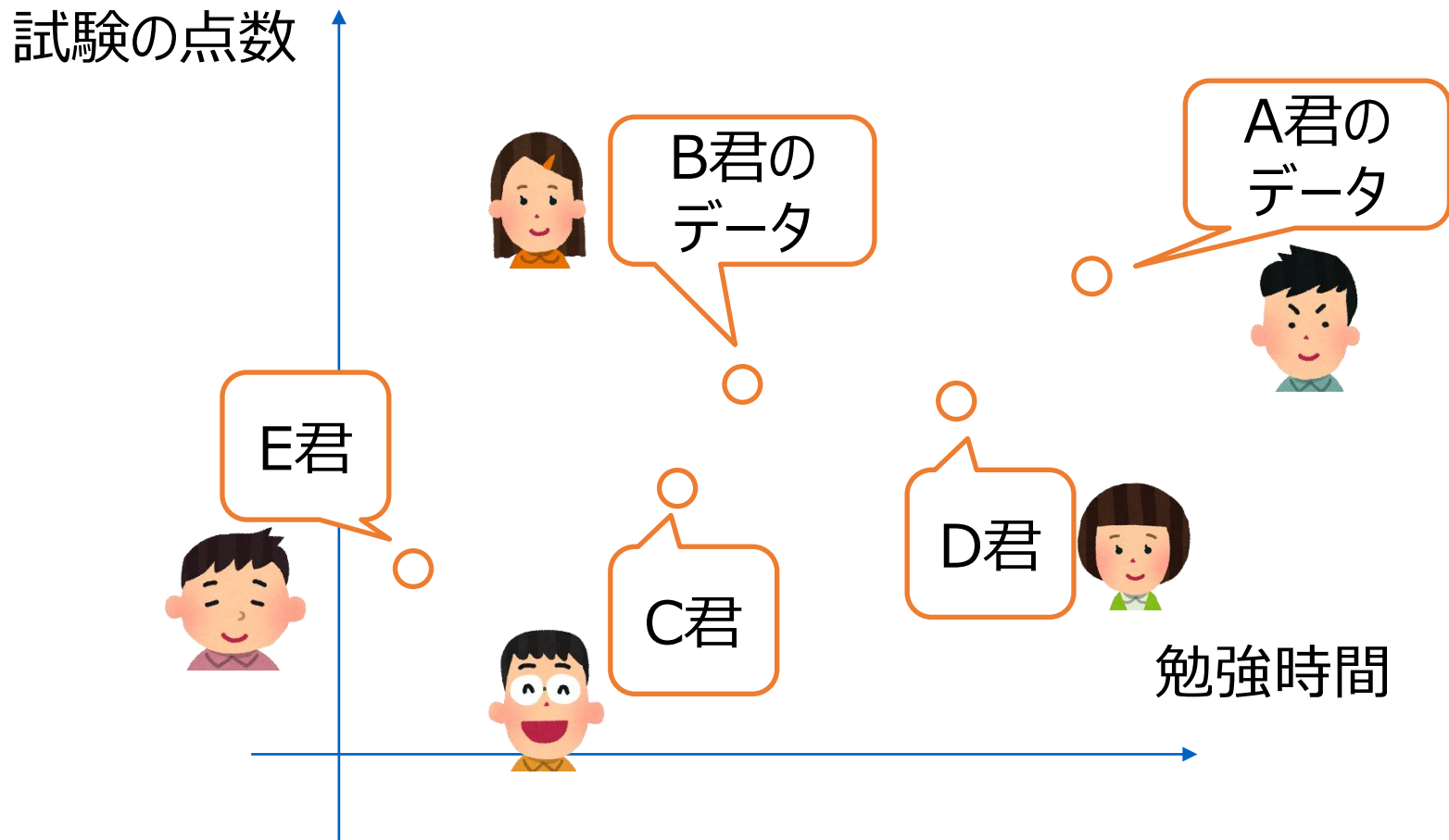
- 例

入力(原因)	結果
勉強時間	試験の点数
食事量	糖尿病になる確率
(身長, 体重)	バスケットボールの得点
画像	その画像がリンゴの写真である確率

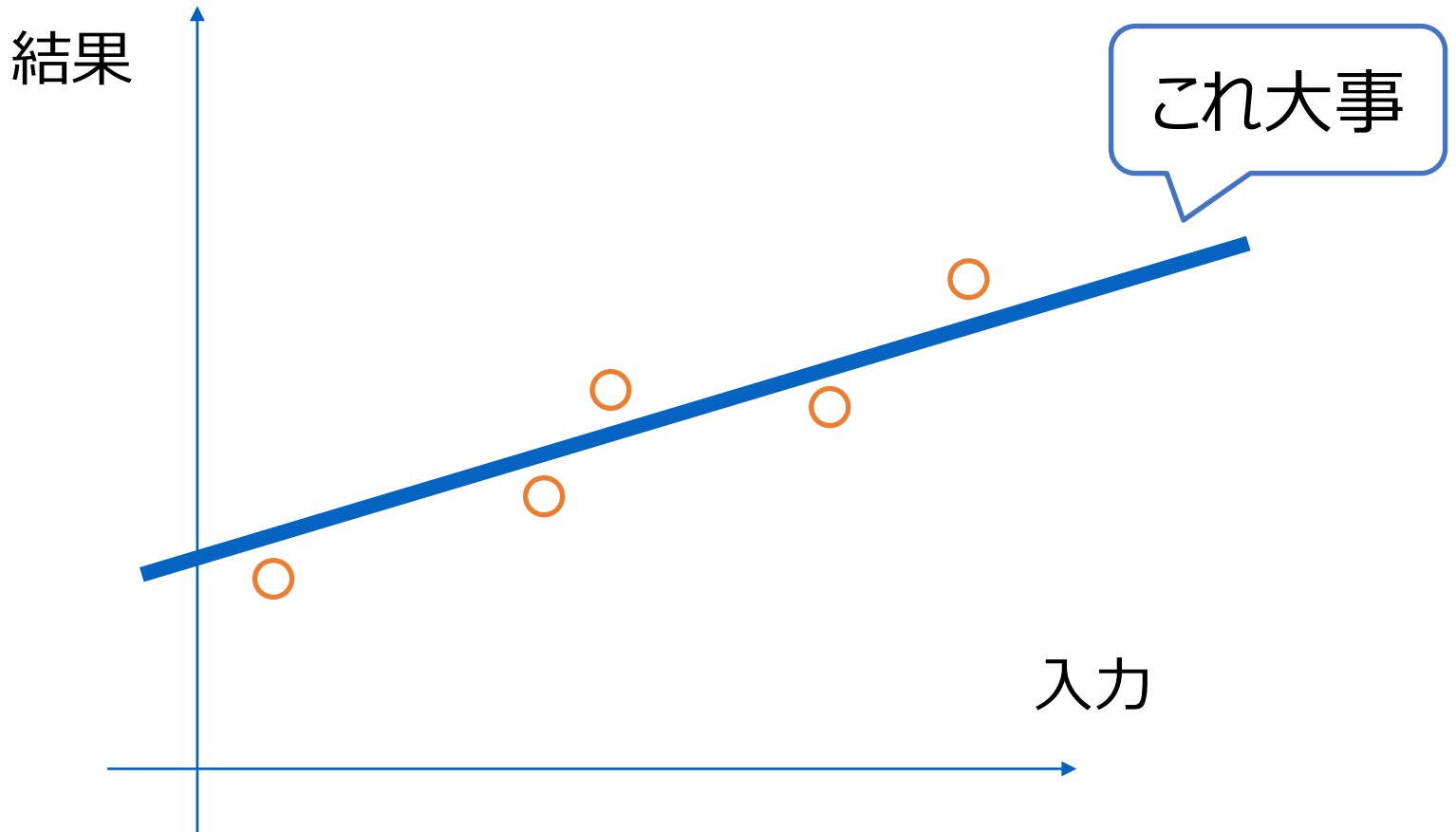
回帰 (ステップ1/3) : データ収集



回帰（ステップ1/3）： データ収集の例

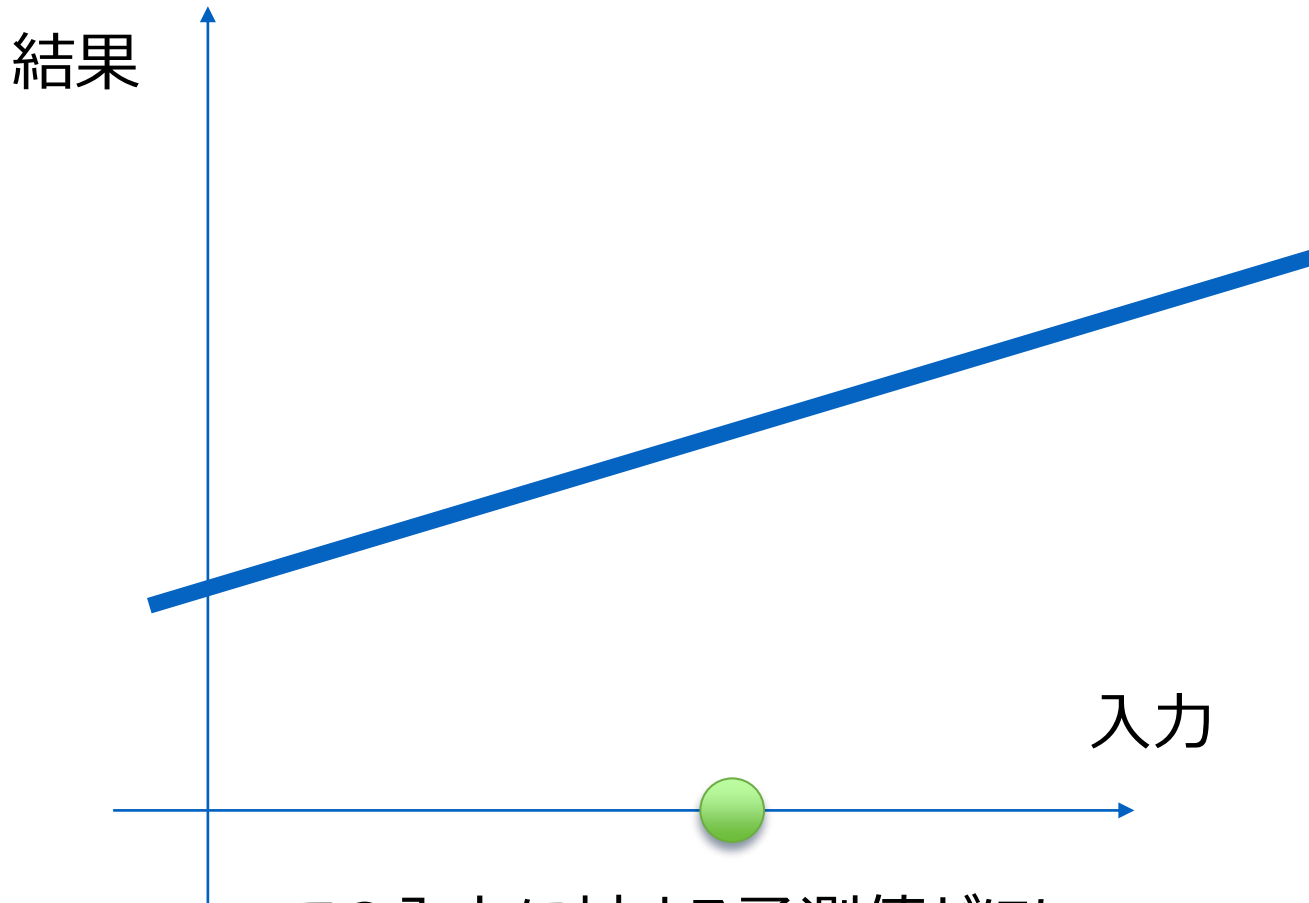


回帰（ステップ2/3）： モデルあてはめ



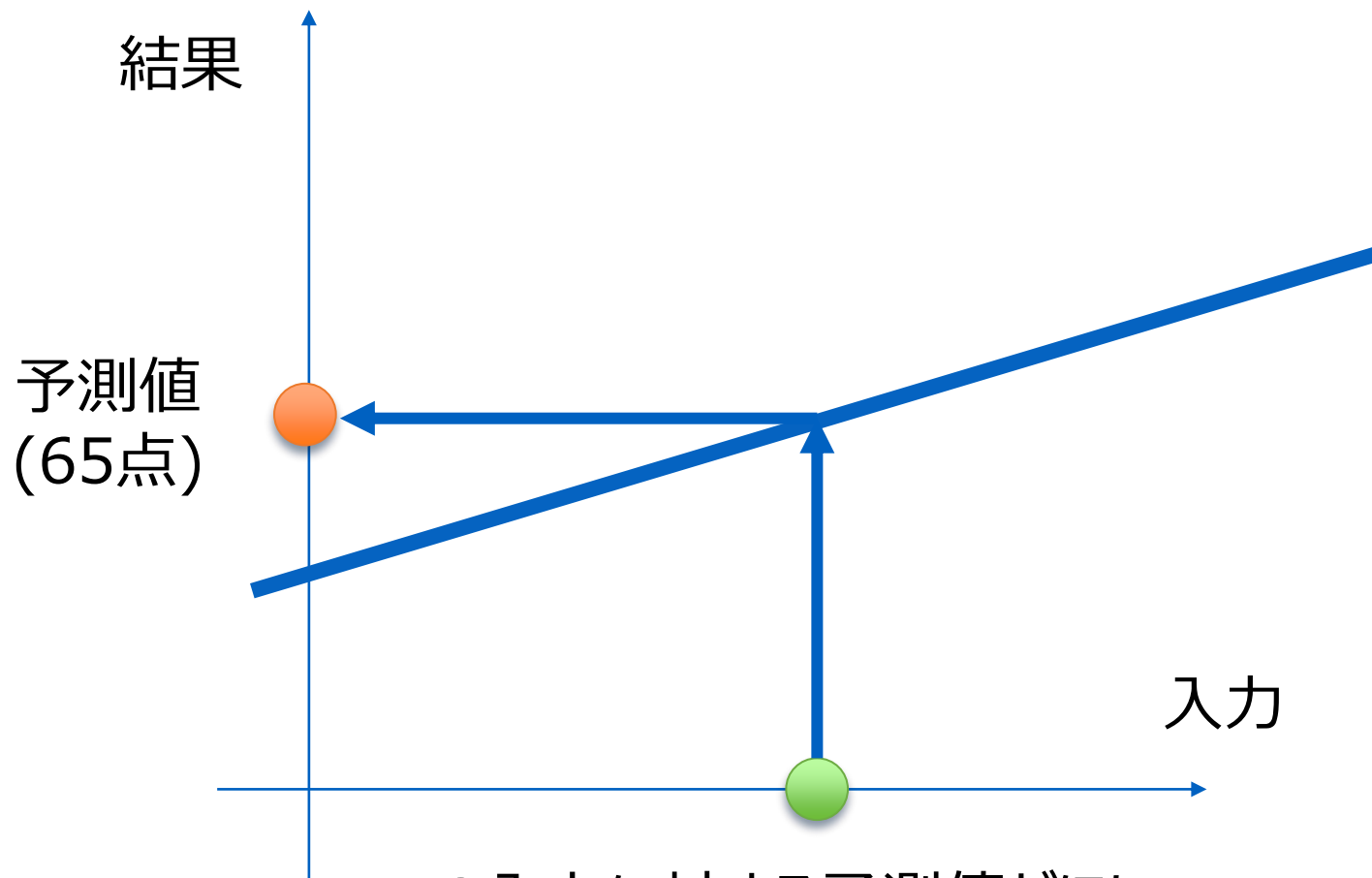
モデル = 「条件 or 入力」と「結果」間に成り立つと予想される関係。
上記は「線形モデル」

回帰 (ステップ3/3) : 予測



この入力に対する予測値がほしい
(ex. 勉強3時間だと何点取れそう?)

回帰 (ステップ3/3) : 予測

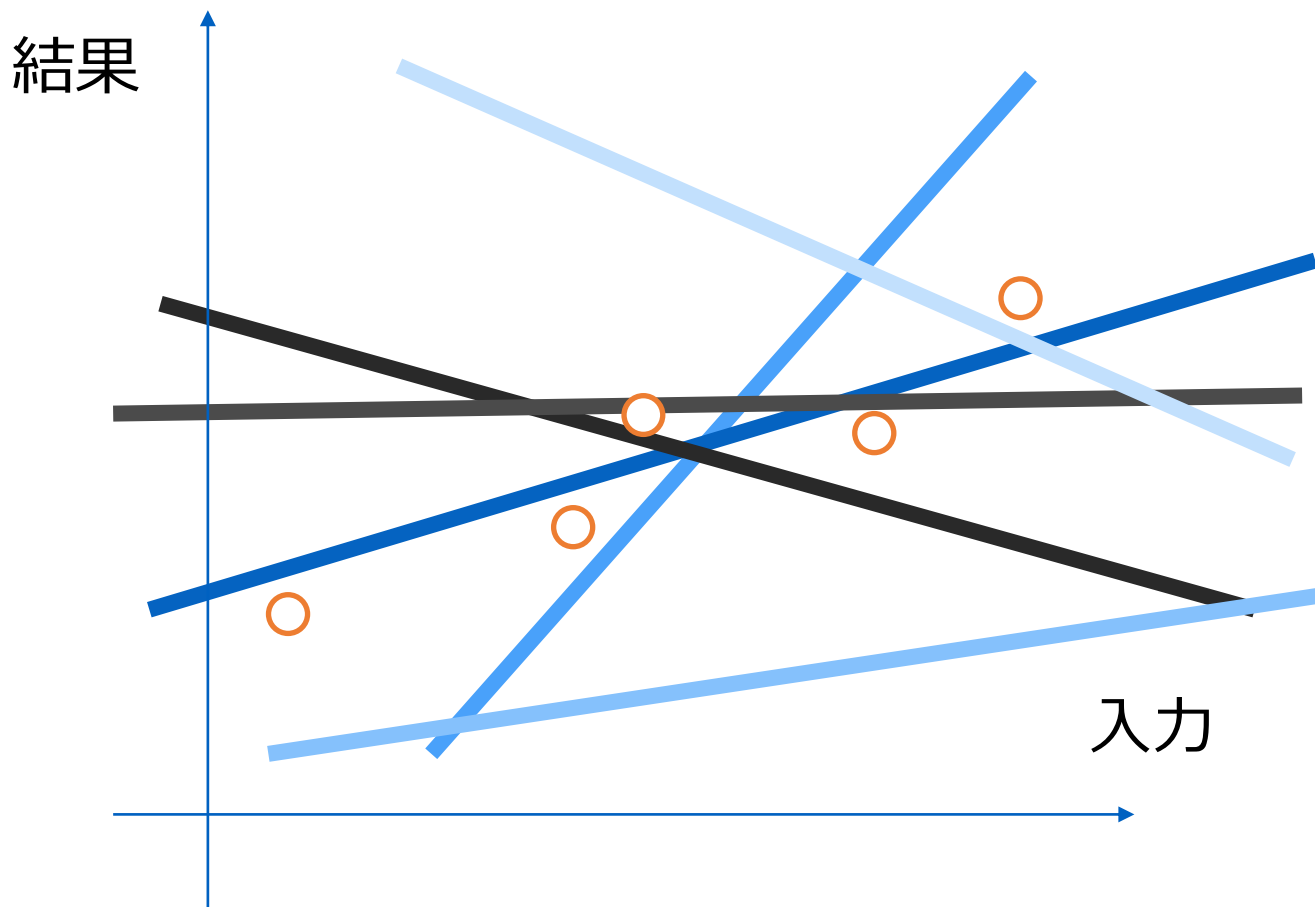


この入力に対する予測値がほしい
(ex. 勉強3時間だと何点取れそう?)

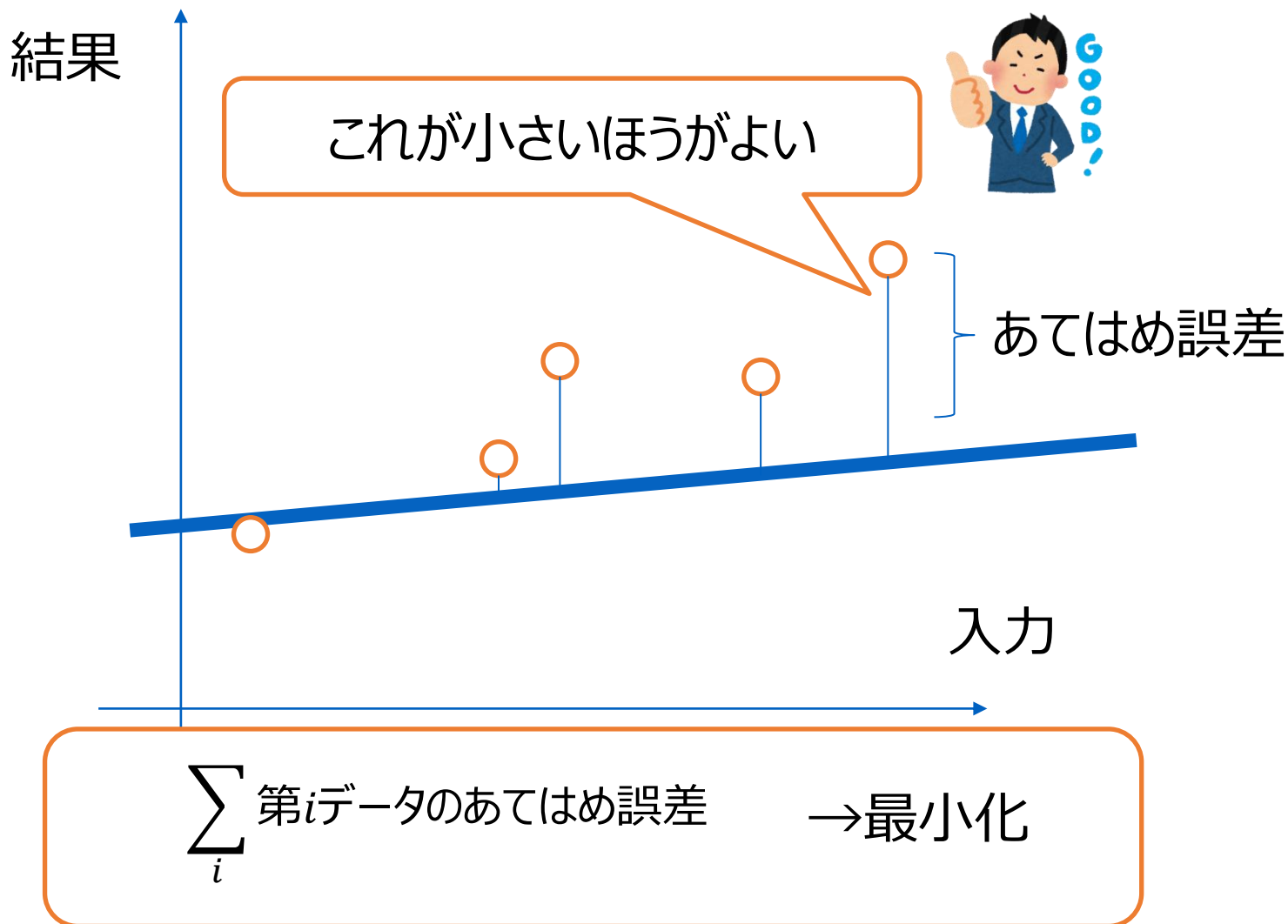
回帰分析とは

- いくつかの（入力，結果）の事例に基づいて，それらの関係をモデル化することで，新たな入力に対して，その結果を予測する方法
 - 結果の原因となっている要因を知ることができます
- 専門的な用語としては
 - 入力→説明変数
 - 結果→目的変数，外的変数などと呼びます

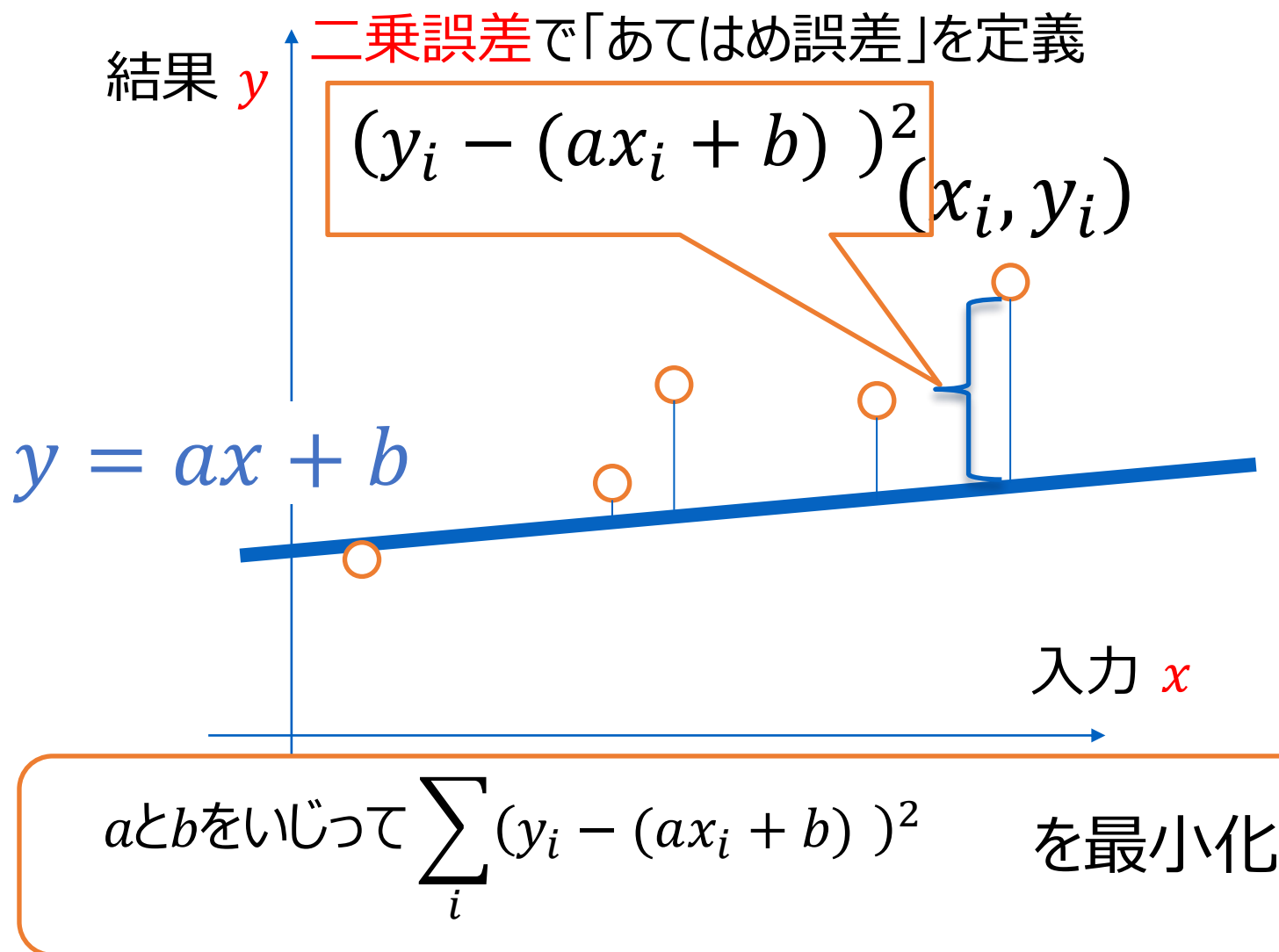
どういう「あてはめ結果」が望ましい？ (1/3)



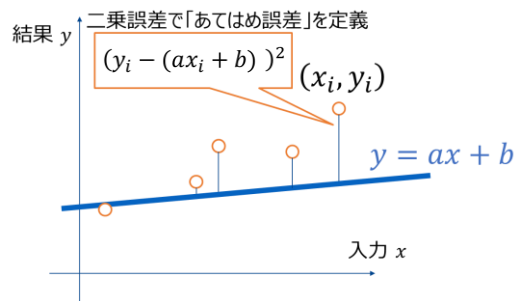
どういふ「あてはめ結果」が望ましい？ (2/3)



どういふ「あてはめ結果」が望ましい？ (3/3)



これを「最小二乗法」と呼ぶ



二乗誤差を最小にしたいから
最小二乗法

a と b をいじって $\sum_i (y_i - (ax_i + b))^2$ を最小化

どうやって??



習ったかもしれませんが、
難しいことは python のパッケージがやってくれます！

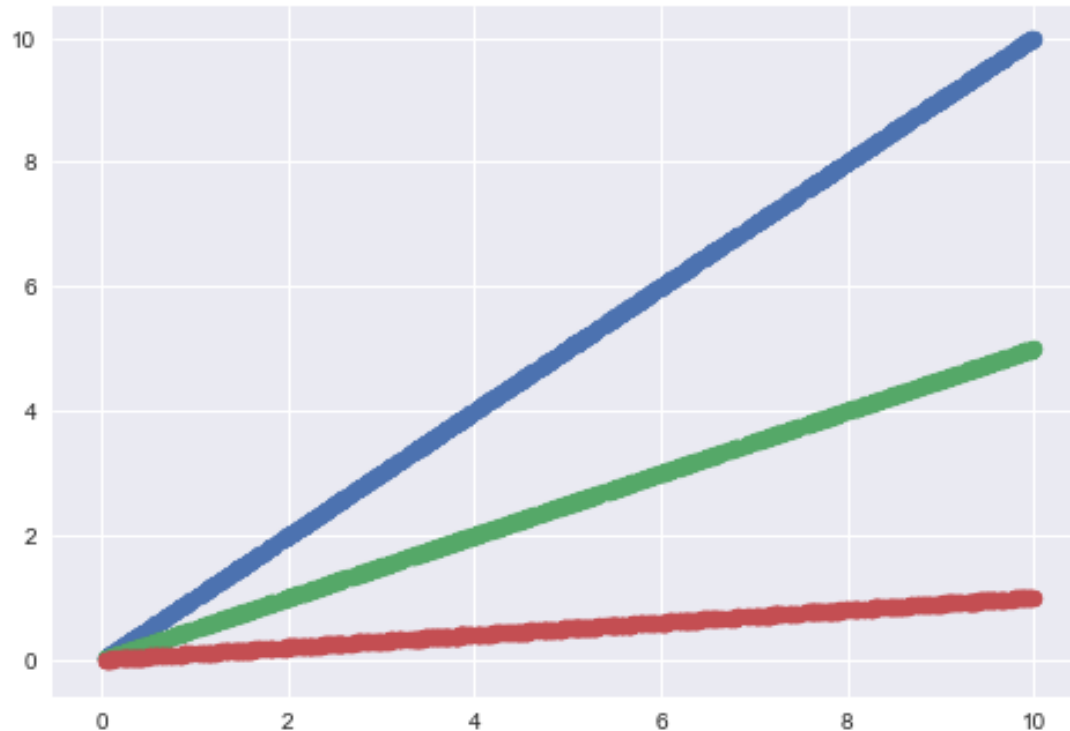
難しいことは一回忘れて・・・

回帰分析の目的 (1)

- 回帰分析の目的は大きく2つ
- 予測
 - 目的：将来どのようになるかを予測したい！
 - 例：あるコンビニでの、過去365日分の天候データ（天気，気温，湿度）から，アイスの売上を回帰
→ 明日の天気でのアイスの売上を予測
- 原因分析
 - 目的：どのような要因で事象が起こっているか知りたい！
 - 例：アイスの売上に関わっているものは何か？

相関分析との違い

- 相関は傾きは関係ない
- 下記の3つの分布は、全て相関 1
- 回帰直線上にデータがまとまっているか？



単回帰

- 一つの説明変数Xから一つの変数Yを回帰
- 例：身長から体重を回帰

回帰

```
from statsmodels import api as sm
import matplotlib.pyplot as plt
```

```
ais = pd.read_csv('./ais.csv')
X = ais.Ht          ← 「Ht」を説明変数にセット
X = sm.add_constant(X) ← 定数項を付けます（後述）
Y = ais.Wt
model=sm.OLS(Y,X)
result = model.fit()
result.summary()
```

※ ais.csv は今までの身長体重データの

フルバージョン：<http://www.statsci.org/data/oz/ais.html>

OLS: Ordinary Least Squares

const 補足

- $y = ax + b$
- $x = x_1, x_2 = 1$ として2変数回帰と考える
- $y = a_1x_1 + a_2x_2 = a_1x_1 + a_2$
- もともとのデータ（説明変数の数 n ）に新しい説明変数（ただし全部1）を追加すれば、定数項のついている回帰は $n+1$ 変数の回帰として解ける

単回帰の結果

決定係数（後述）：
0~1 で1に近い
ほど良い

OLS Regression Results

```
Dep. Variable:          Wt      R-squared:                0.610
Model:                  OLS      Adj. R-squared:           0.608
Method:                  Least Squares      F-statistic:              312.6
Date:                    Tue, 20 Jun 2017      Prob (F-statistic):       9.64e-43
Time:                    00:26:52      Log-Likelihood:           -723.08
No. Observations:        202      AIC:                      1450.
Df Residuals:            200      BIC:                      1457.
Df Model:                 1
Covariance Type:         nonrobust
```

coef (変数に対応した係数)=0という仮説の棄却

説明変数に効果があるか？のp値

< 0.05
で効果あり！

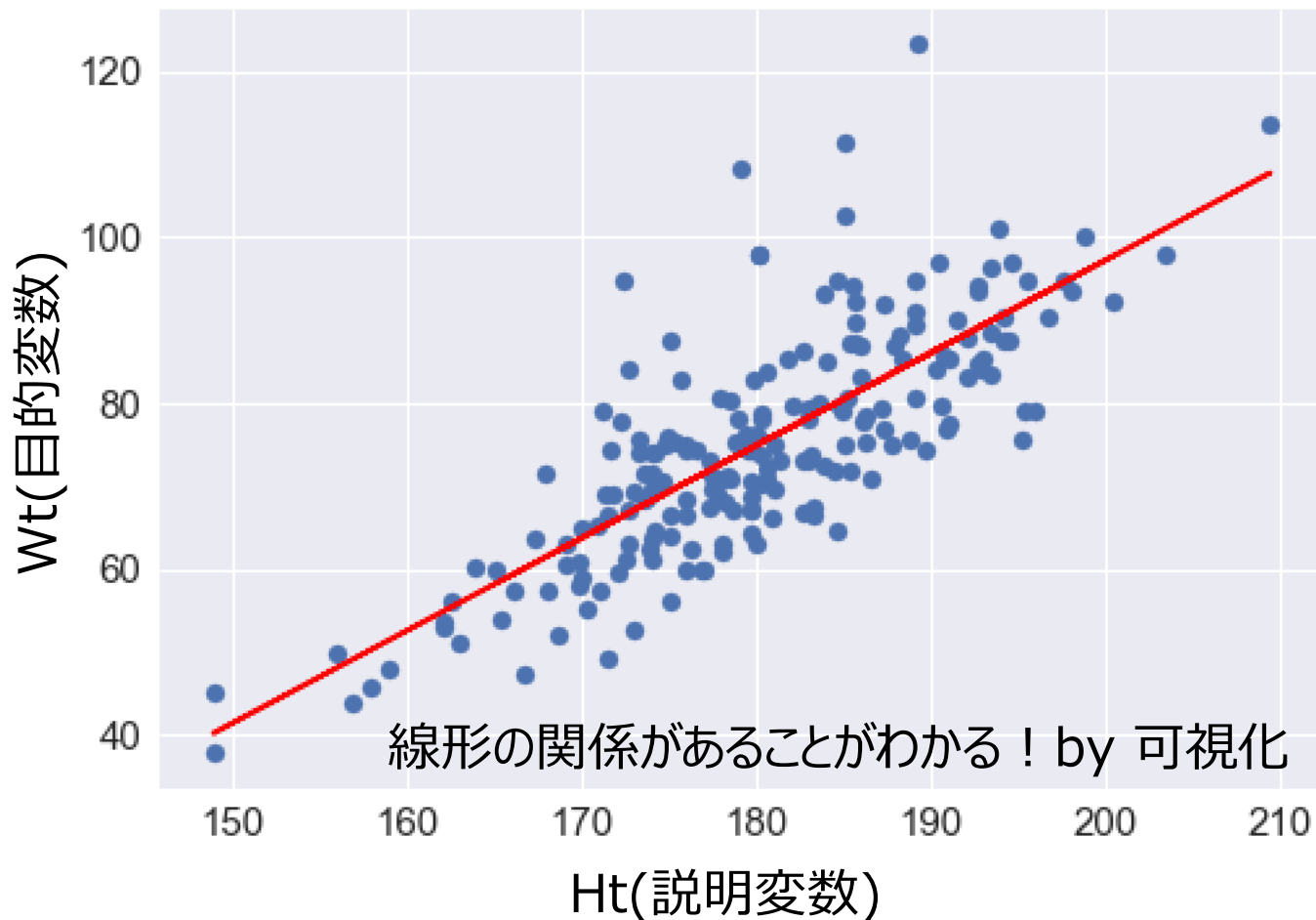
	coef	std err	t	P> t	[95.0% Conf. Int.]	
const	-126.1890	11.397	-11.073	0.000	-148.662	-103.716
Ht	1.1171	0.063	17.680	0.000	0.993	1.242

```
Omnibus:                57.269      Durbin-Watson:            1.580
Prob(Omnibus):           0.000      Jarque-Bera (JB):         136.811
Skew:                    1.266      Prob(JB):                 1.96e-30
Kurtosis:                6.137      Cond. No.                 3.35e+03
```

「あてはめ（予測）」式：
 $Wt(\text{目的変数}) = 1.1171 \times Ht(\text{説明変数}) - 126.1890$
線形の関係があることがわかる！ by 式

可視化

```
plt.scatter(x=ais.Ht,y=ais.Wt) and plt.plot(ais.Ht,  
model.fit().predict(X),c="red")
```

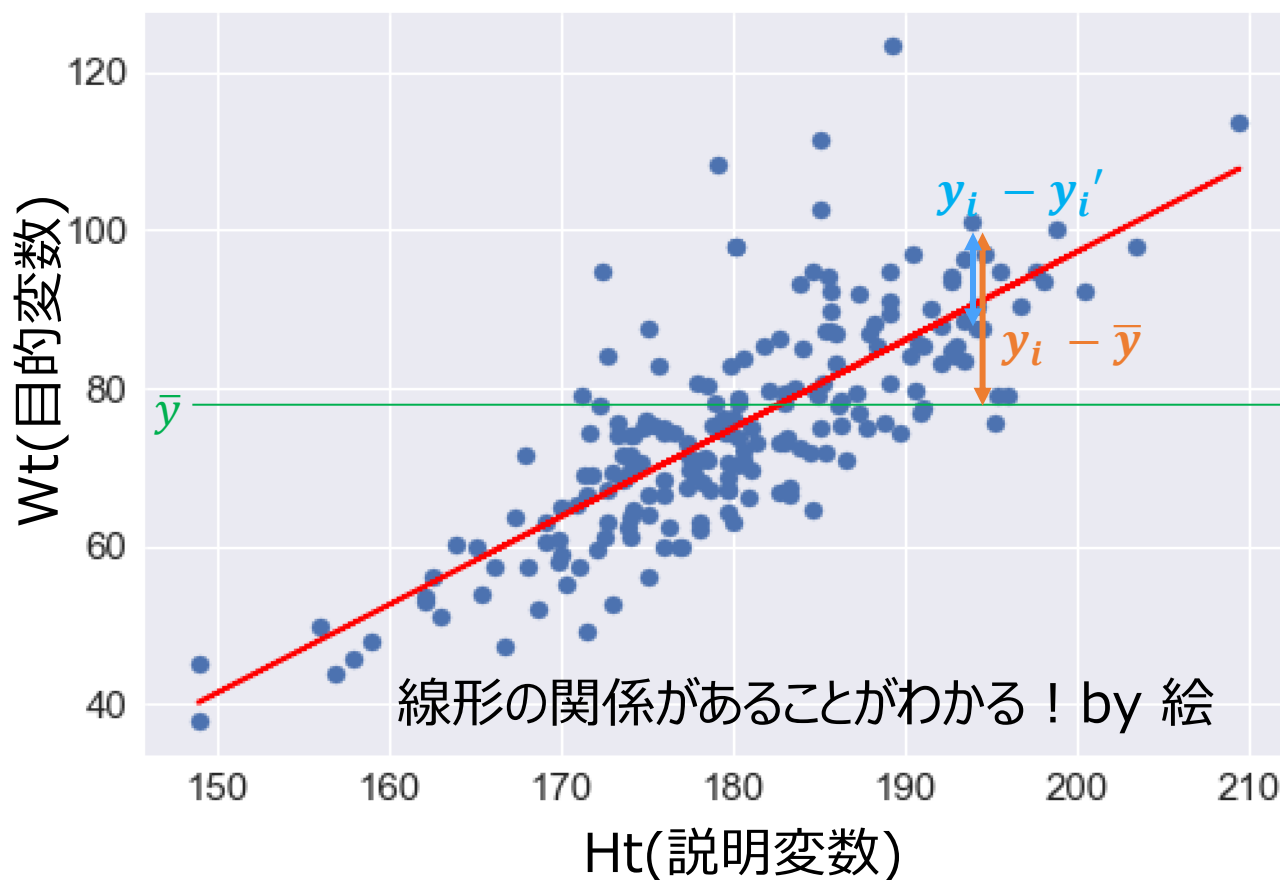


決定係数

```
plt.scatter(x=ais.Ht,y=ais.Wt) and plt.plot(ais.Ht,  
model.fit().predict(X),c="red")
```

$$\text{決定係数: } R^2 = 1 - \frac{\sum_{i=1}^n (y_i - y_i')^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

y_i : 標本値
 y_i' : 予測値
 \bar{y} : 標本平均値



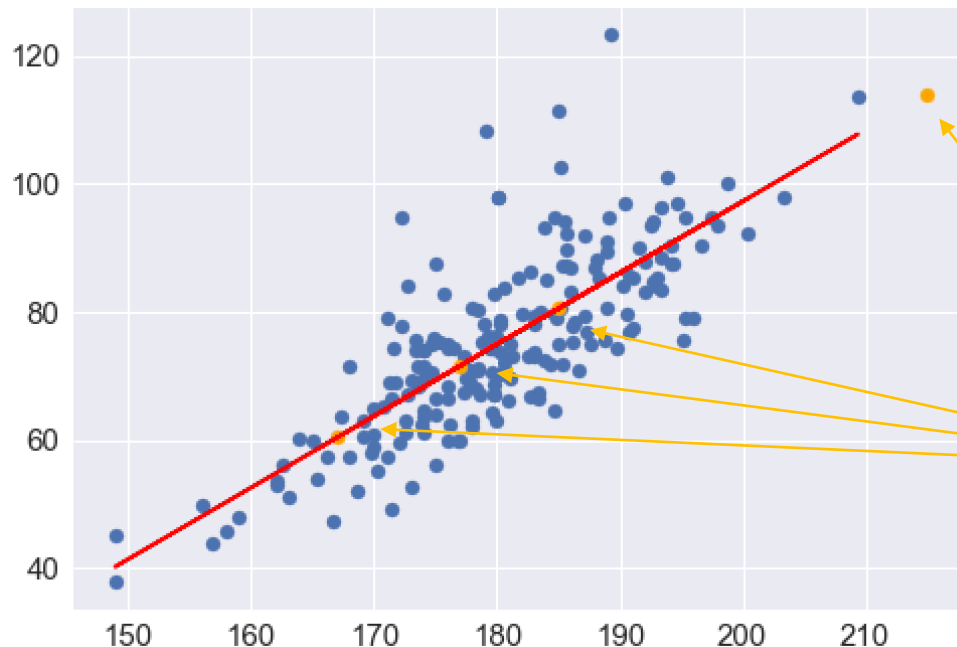
「線形関数」で
の予測がどの程度
うまくいっているか
の指標

未知データでの予測

- 得られた回帰式を用いて、身長が [185, 167, 215, 177] の人たちの体重を予測したい

未知データ

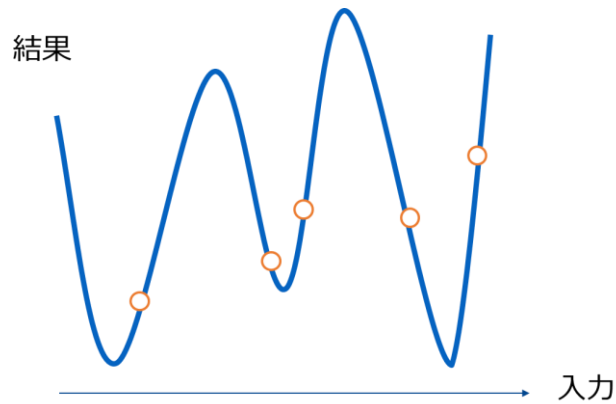
```
unknown = [185, 167, 215, 177]
X2 = sm.add_constant(pd.DataFrame(unknown))
Wt2 = model.fit().predict(X2)
plt.scatter(x=ais.Ht,y=ais.Wt) and plt.plot(ais.Ht,
model.fit().predict(X),c="red") and plt.scatter(x=unknown,y=Wt2, c='orange')
```



予測値

より複雑（線形でない）モデル： オーバーフィッティングと汎化能力

高次多項式モデル

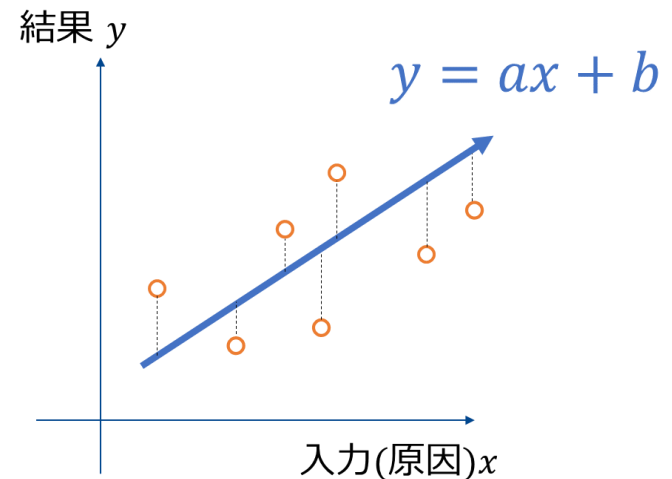


$$y = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

(良) 事前に与えられたデータに
対する誤差小さい

(悪) 大量に事前データがないと
回帰結果に汎化能力がない恐れ
(予測がうまくできない。
オーバーフィッティング)

線形モデル



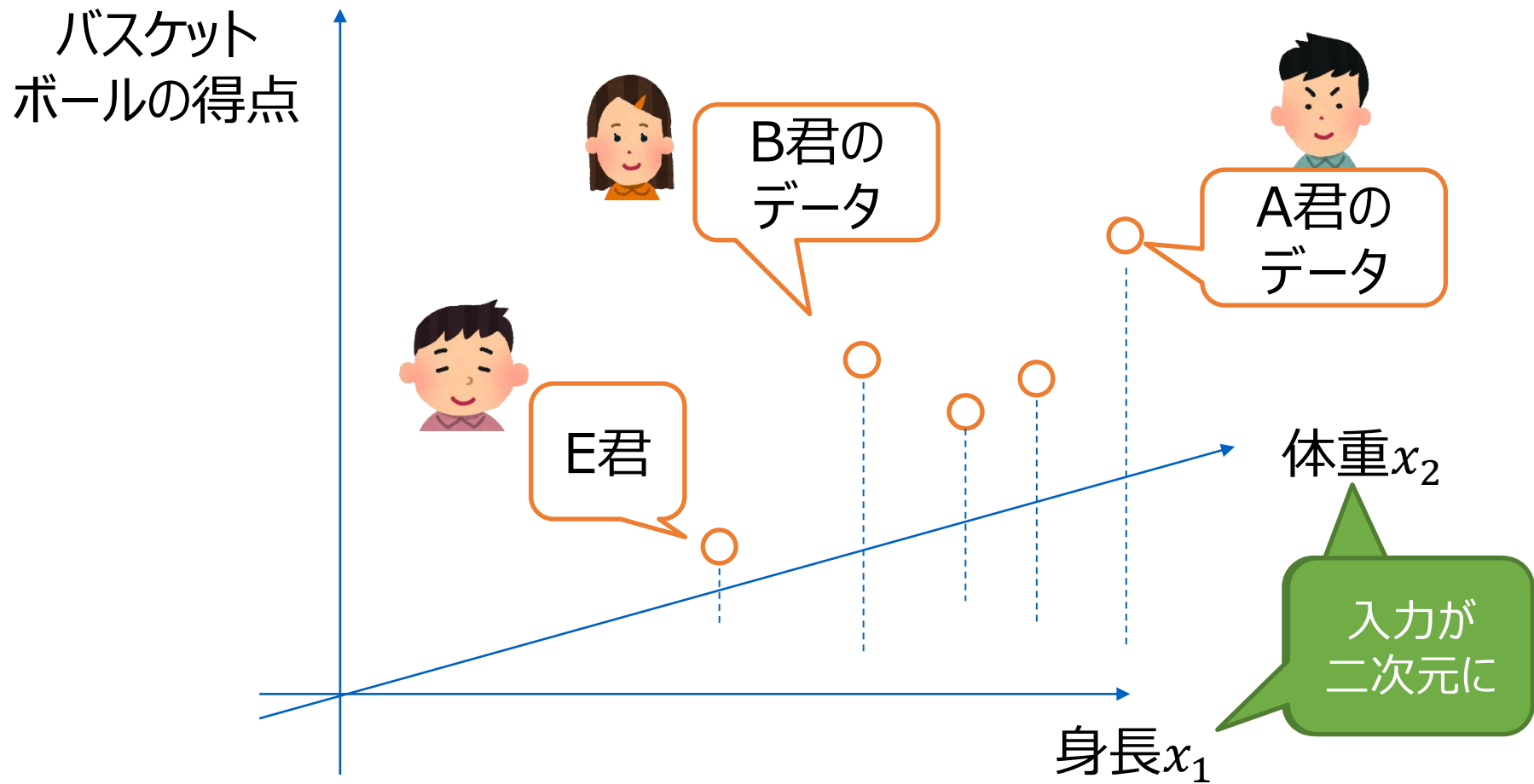
(悪) 事前に与えられたデータに
対する誤差大

(良) 事前データ数が少なくても
ひどいオーバーフィッティングは少ない

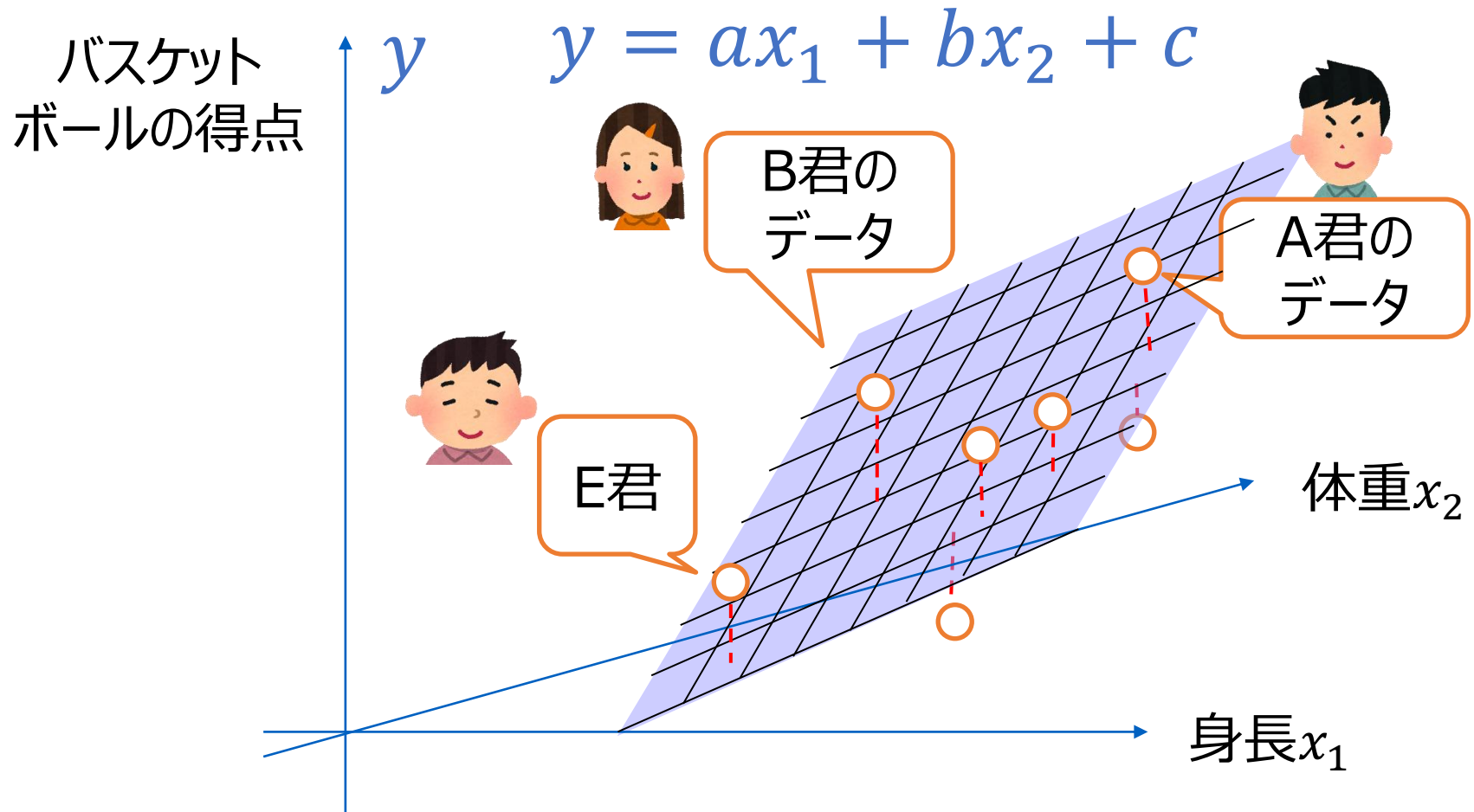
練習 3 : 単回帰

- ais.csvを読み込んで、回帰分析を用いて、体重から身長を回帰してみよう

重回帰による予測（ステップ1/3）： データ収集



重回帰による予測（ステップ2/3）： モデルあてはめ



→ やはり最小二乗法になります

各データから平面へ伸ばした
誤差の二乗を最小化

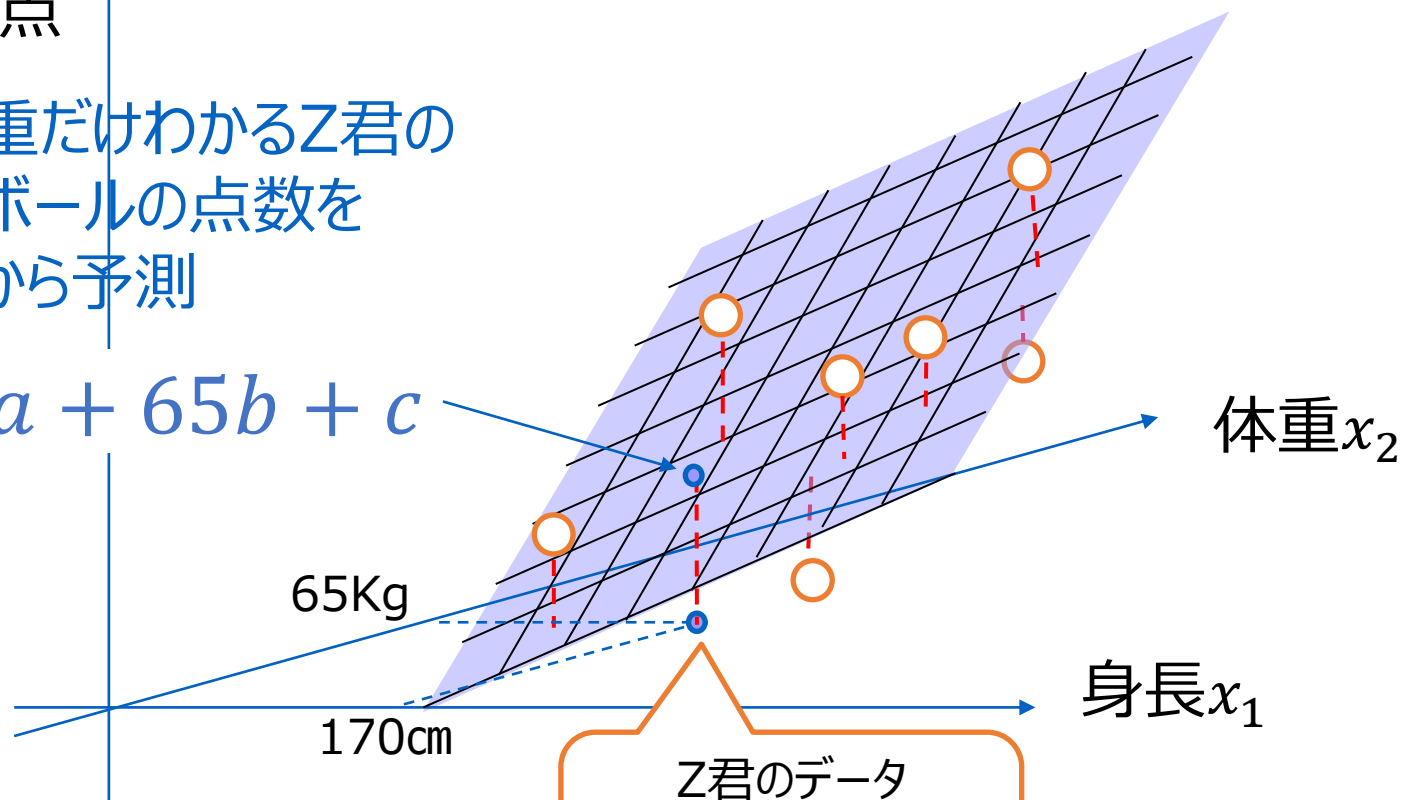
重回帰による予測（ステップ 3 / 3）： 予測

バスケット
ボールの得点

$$y = ax_1 + bx_2 + c$$

→ 身長と体重だけわかるZ君の
バスケットボールの点数を
求めた式から予測

$$y = 170a + 65b + c$$



Z君のデータ
身長：170cm、
体重：65Kg

重回帰

- 複数の説明変数Xから目的変数Yを回帰
- 例1：Ht, Wt からLBM(Lean Body Mass, 脂肪を除いた重量)

例1：身長と体重からLBMを予測

```
X = ais[['Ht', 'Wt']]
X = sm.add_constant(X)
Y = ais.LBM
model=sm.OLS(Y,X)
result = model.fit()
result.summary()
```

重回帰の結果

OLS Regression Results

```
=====
Dep. Variable:          LBM      R-squared:                0.881
Model:                  OLS      Adj. R-squared:           0.880
Method:                 Least Squares      F-statistic:              737.1
Date:                   Tue, 20 Jun 2017    Prob (F-statistic):       9.85e-93
Time:                   01:12:47          Log-Likelihood:           -590.29
No. Observations:      202          AIC:                      1187.
Df Residuals:          199          BIC:                      1196.
Df Model:               2
Covariance Type:      nonrobust
=====
```

	coef	std err	t	P> t	[95.0% Conf. Int.]	
const	-36.6586	7.519	-4.875	0.000	-51.486	-21.831
Ht	0.2587	0.053	4.922	0.000	0.155	0.362
Wt	0.7325	0.037	19.941	0.000	0.660	0.805

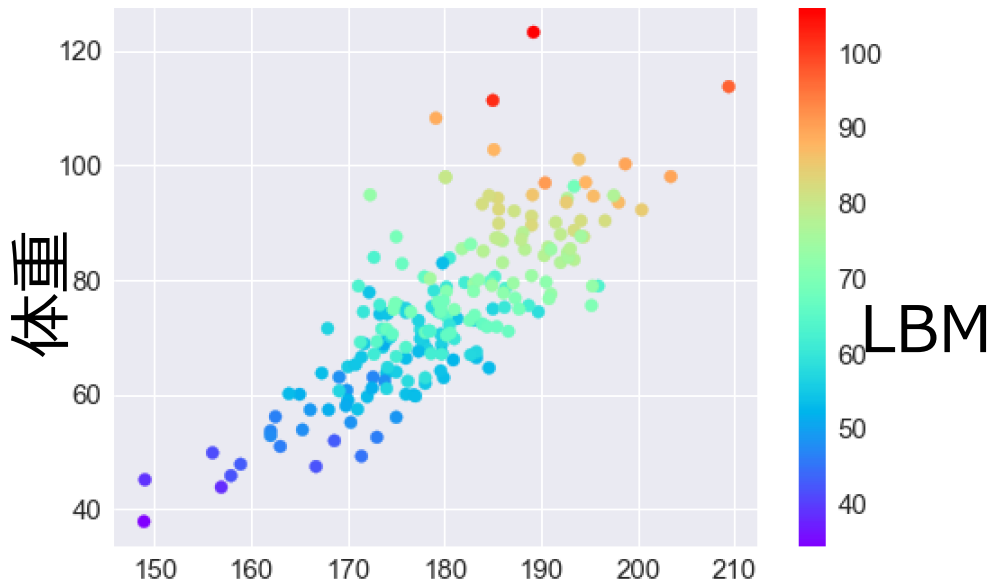
```
=====
Omnibus:                22.720      Durbin-Watson:            0.692
Prob(Omnibus):          0.000      Jarque-Bera (JB):         26.767
Skew:                   -0.862     Prob(JB):                  1.54e-06
Kurtosis:                3.458      Cond. No.                  4.61e+03
=====
```


$$\text{LBM(目的変数)} = 0.2587 \times \text{Ht} + 0.7325 \times \text{Wt} - 36.6586$$

可視化

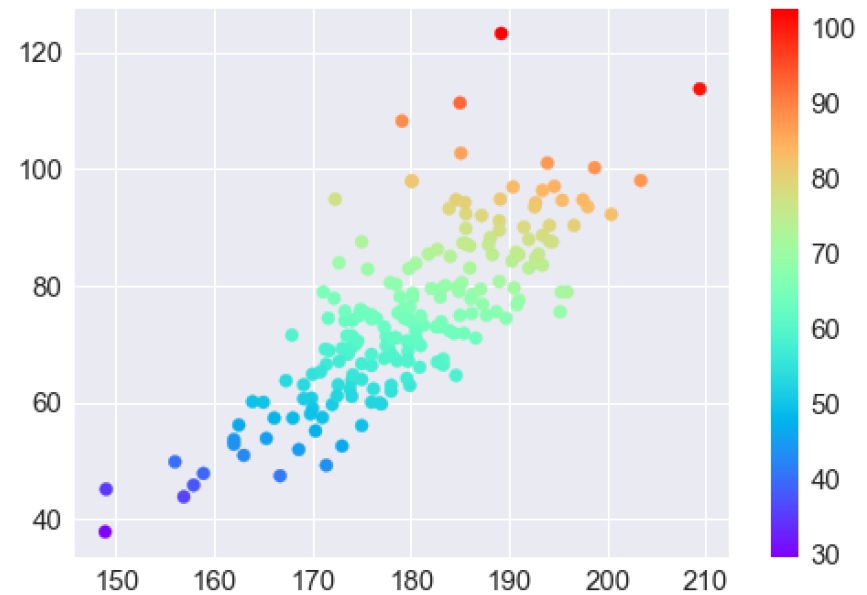
```
plt.scatter(x=X.Ht, y=X.Wt, c=Y,  
           cmap='rainbow') and plt.colorbar()
```

色は実際のLBMに対応



```
plt.scatter(x=X.Ht, y=X.Wt, c=model.fit().predict(X),  
           cmap='rainbow') and plt.colorbar()
```

色はLBMの推定値に基づく



身長
目的変数の値（色分け）を2変数で
うまく予測できていることがわかる！

重回帰

- 例2：ワインの品質データ
 - winequality-red.csv
 - アルコール度数やpH値等， 11個の説明変数
 - 3～8 までの品質スコア

ワインの品質データ

```
from statsmodels import api as sm
import matplotlib.pyplot as plt
wine_data = pd.read_csv("winequality-red.csv", sep=";")
X = wine_data.drop("quality", axis=1) ←「quality」以外を説明変数に
X = sm.add_constant(X)
Y = wine_data['quality'] ←「quality」を目的変数に
model = sm.OLS(Y, X) ← 回帰！
result = model.fit()
result.summary()
```

回帰結果

OLS Regression Results

```
=====
Dep. Variable:          quality    R-squared:                0.361
Model:                 OLS        Adj. R-squared:           0.356
Method:                Least Squares  F-statistic:              81.35
Date:                  Tue, 20 Jun 2017  Prob (F-statistic):       1.79e-145
Time:                  01:04:29     Log-Likelihood:          -1569.1
No. Observations:     1599        AIC:                     3162.
Df Residuals:         1587        BIC:                     3227.
Df Model:              11
Covariance Type:      nonrobust
=====
```

決定係数:0~1 で1に近いほど良い

各説明変数に効果があるか? のp値

	coef	std err	t	P> t	[95.0% Conf. Int.]
const	21.9652	21.195	1.036	0.300	-19.607 63.538
fixed acidity	0.0250	0.026	0.963	0.336	-0.026 0.076
volatile acidity	-1.0836	0.121	-8.948	0.000	-1.321 -0.846
citric acid	-0.1826	0.147	-1.240	0.215	-0.471 0.106
residual sugar	0.0163	0.015	1.089	0.276	-0.013 0.046
chlorides	-1.8742	0.419	-4.470	0.000	-2.697 -1.052
free sulfur dioxide	0.0044	0.002	2.009	0.045	0.000 0.009
total sulfur dioxide	-0.0033	0.001	-4.480	0.000	-0.005 -0.002
density	-17.8812	21.633	-0.827	0.409	-60.314 24.551
pH	-0.4137	0.192	-2.159	0.031	-0.789 -0.038
sulphates	0.9163	0.114	8.014	0.000	0.692 1.141
alcohol	0.2762	0.026	10.429	0.000	0.224 0.328

< 0.05
で効果あり!

```
=====
Omnibus:                27.376    Durbin-Watson:           1.757
Prob(Omnibus):          0.000    Jarque-Bera (JB):        40.965
Skew:                   -0.168   Prob(JB):                 1.27e-09
Kurtosis:                3.708    Cond. No.:                1.13e+05
=====
```

quality(目的変数) = 0.0250 × fixed acidity + ...

説明変数が大量になっても大丈夫!

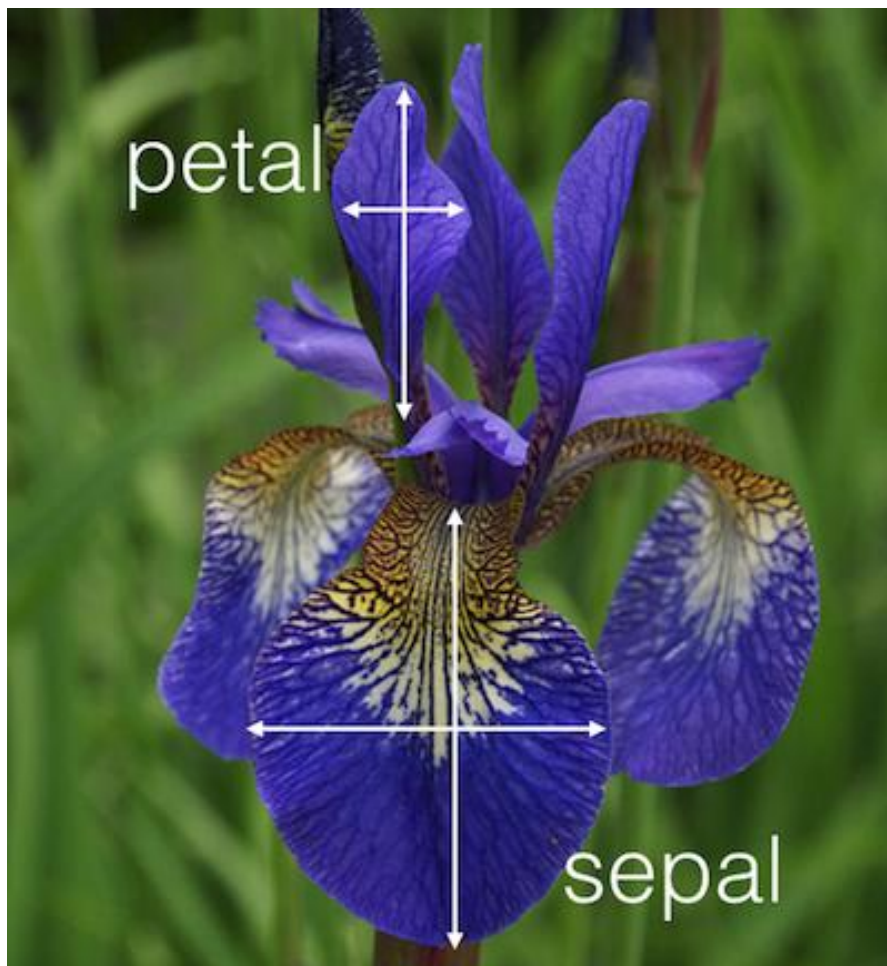
練習 4 : 重回帰

- `ais.csv`を読み込んで、回帰分析を用いて、身長とLBMから体重を予測してみよう
- ヒント
`X = ais[['Ht', 'LBM']]`

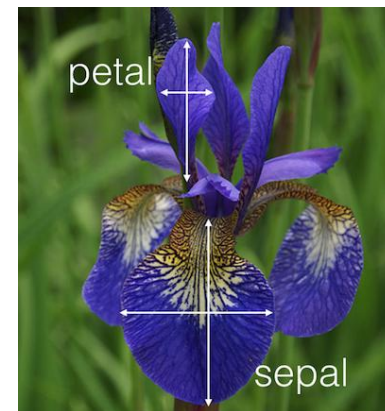
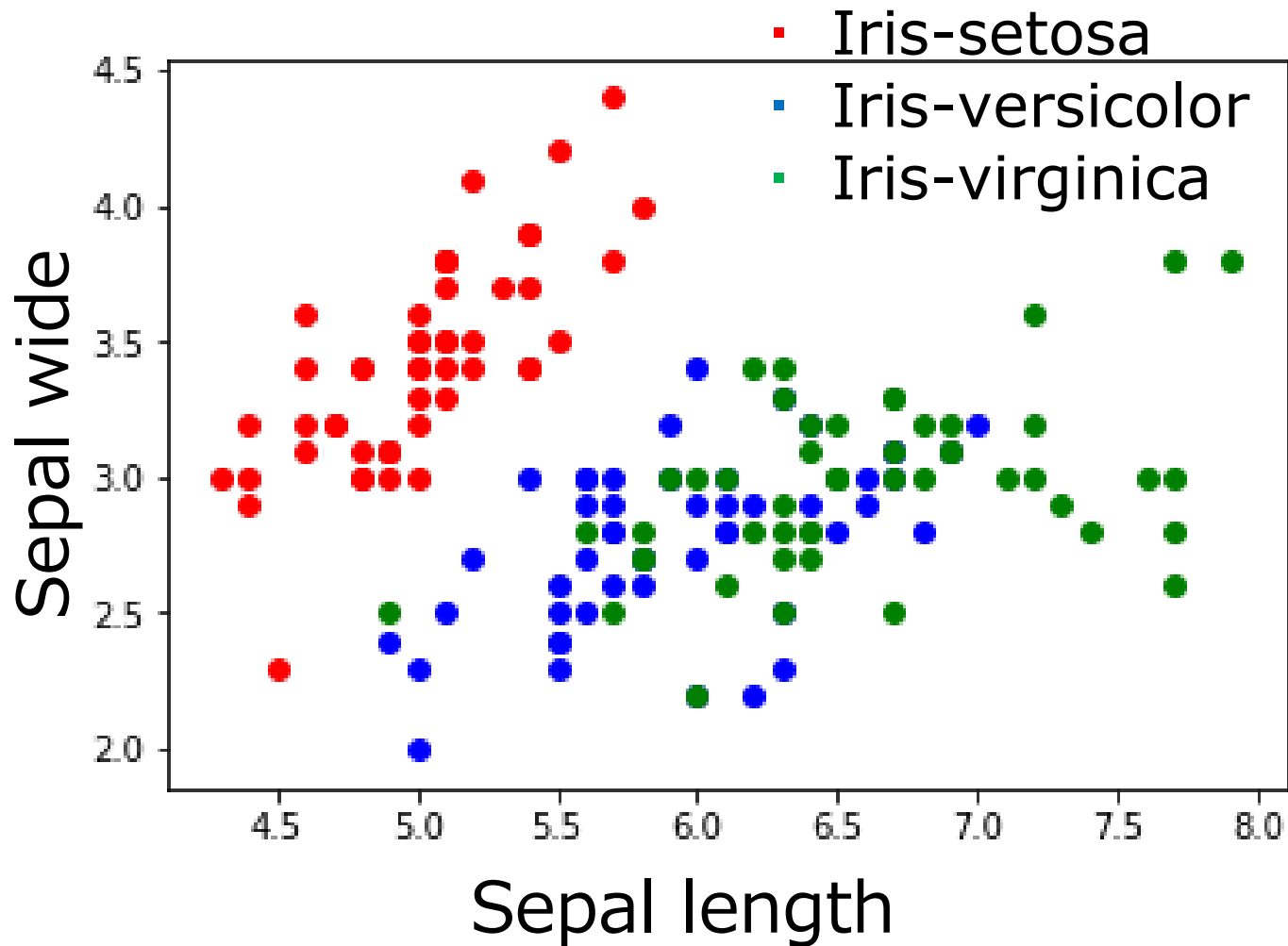
演習

演習 1 : アイリスデータ

- アイリスデータ
- 3クラス
 - Iris-setosa
 - Iris-versicolor
 - Iris-virginica
- 指標
 - sepal length
 - sepal width
 - petal length
 - petal width



演習 1 : 3つのクラス



演習 1 – 1 : 相関マップ

- 3 つクラスに分けて、クラスごとに相関マップを作れ

- ヒント

classごとにデータを分ける

```
irisSetosa = iris[iris.iloc[:,4]=='Iris-setosa']  
irisVersicolor = iris[iris.iloc[:,4]=='Iris-versicolor']  
irisVirginica = iris[iris.iloc[:,4]=='Iris-virginica']
```

演習 1 – 2 : 単回帰

- 3 つクラスに分けて、クラスごとにsepal-lengthを説明変数, petal-lengthを目的変数とした回帰分析を行い、データ分布及び回帰直線を可視化。
- 回帰分析結果のsummaryを読み取り、回帰直線の傾き及び決定係数を答えよ。
- 3つのクラスの中で、決定係数が高い順に答え、分布と直線とのFit具合を比べよ。

- ヒント

```
# classごとにデータを分ける
```

```
irisSetosa = iris[iris.iloc[:,4]=='Iris-setosa']  
irisVersicolor = iris[iris.iloc[:,4]=='Iris-versicolor']  
irisVirginica = iris[iris.iloc[:,4]=='Iris-virginica']
```

演習 2 : 小売売上高

- “predicti.csv” : 非農業の雇用、給料、給料支出額と4種類の小売業の11年間分の4半期販売データ

No	変数名	変数ラベル	説明	型
1	TIME	TIME	1979年の第1四半期から1989年の第4四半期までの四半期	id
2	WASA	WASA	国民所得と給料支出額 (10億\$)	numerical
3	EMPL	EMPL	非農業の企業の給料支払簿上の従業員数 (1,000)	numerical
4	BLDG	BLDG	建築資材の販売額 (100万\$)	numerical
5	AUTO	AUTO	自動車の販売額 (100万\$)	numerical
6	FURN	FURN	家具と家の装具の販売額 (100万\$)	numerical
7	GMER	GMER	雑貨の販売額 (100万\$)	numerical

演習 2-1 : 散布図

- 「従業員数-建築資材」、「従業員数-自動車」、「従業員数-家具」、「従業員数-雑貨」それぞれで散布図を作れ
- ヒント
 - `import matplotlib.pyplot as plt`
 - `plt.scatter`

演習 2 – 2 : 相関分析

- 「従業員数－建築資材」、「従業員数－自動車」、「従業員数－家具」、「従業員数－雑貨」それぞれで相関分析せよ
- 相関マップを作れ
- 最も相関の高い変数のペアを見つけよ

演習 2 – 3 : 回帰分析

- 「従業員数－建築資材」、「従業員数－自動車」、「従業員数－家具」、「従業員数－雑貨」それぞれで回帰分析してみよう！まずは、それぞれ可視化してみよう！
- ヒント：
 - `model = sm.OLS(Y, X)`
 - `result = model.fit()`

演習 2 - 4 : 回帰分析

- 「従業員数-建築資材、自動車、家具、雑貨」で重回帰分析してみよう！
- ヒント：
 - `model = sm.OLS(Y, X)`
 - `result = model.fit()`

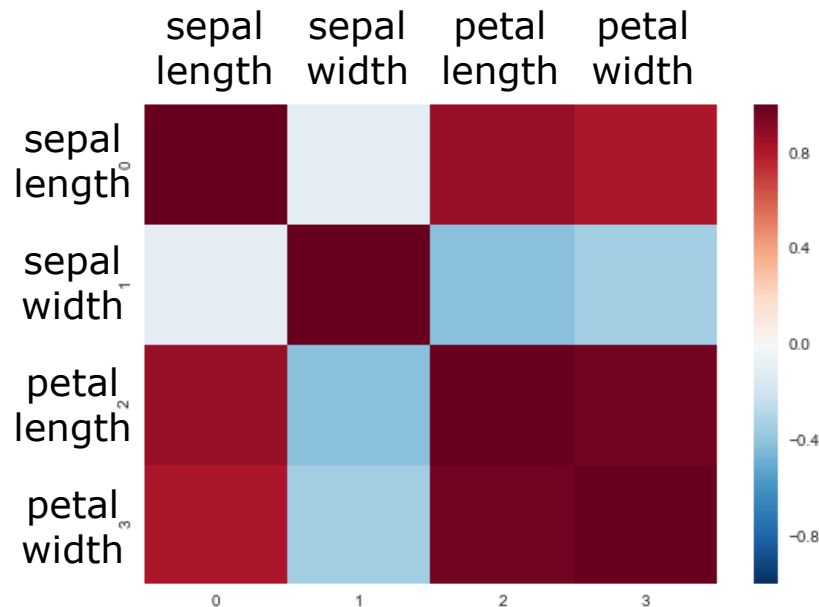
Advance

早く終わった人はやってみよう

演習 3 : 相関マップ

- 演習 1 の全ての項目の組み合わせで、相関の検定を行い、最も相関が高い項目の組み合わせを見つけるプログラムを書け！
ただし、対角成分（同じ項目の組み合わせ）は除く

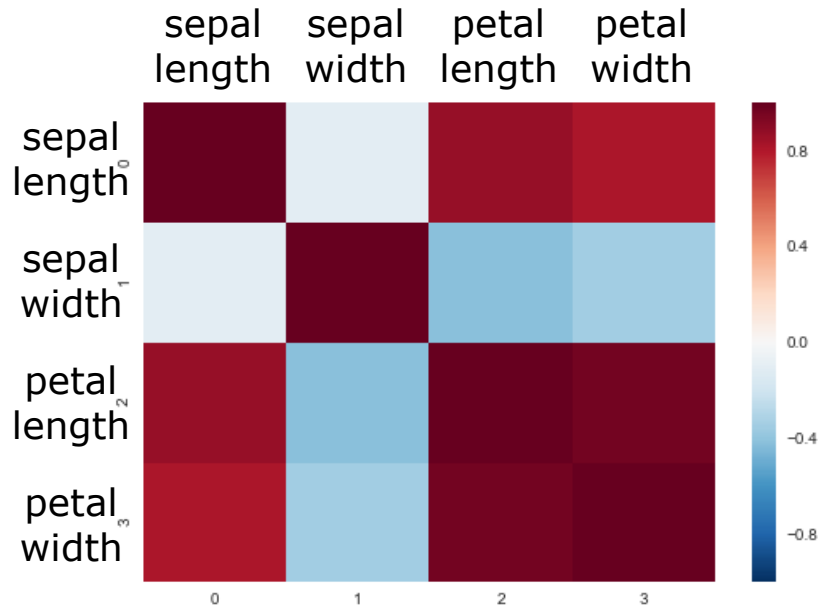
相関係数マップR



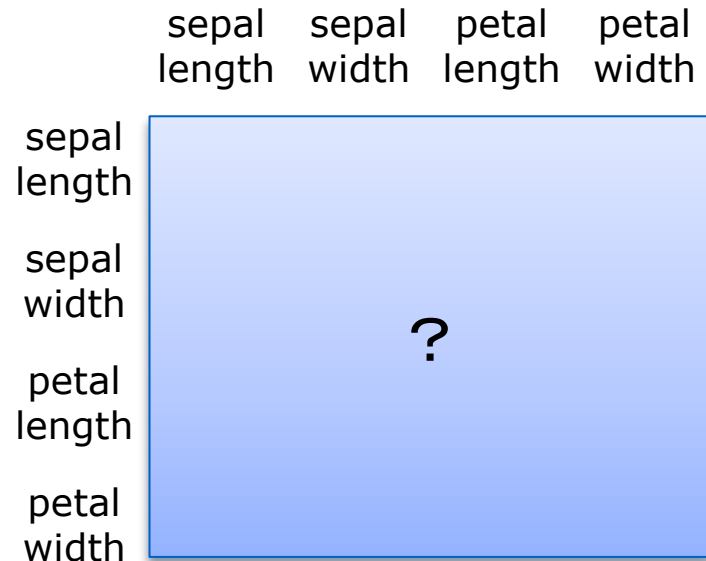
演習 4 : 相関マップ

- 演習 1 の全ての項目の組み合わせで、相関の検定を行い、そのp値のマップを作ってみよう！

相関係数マップR



P値マップP



演習 5

- boston.csv (変数は以下) を用いて,
 1. CRIM: per capita crime rate by town
 2. ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
 3. INDUS: proportion of non-retail business acres per town
 4. CHAS: Charles River dummy variable (=1 if tract bounds river; 0 otherwise)
 5. NOX: nitric oxides concentration (parts per 10 million)
 6. RM: average number of rooms per dwelling
 7. AGE: proportion of owner-occupied units built prior to 1940
 8. DIS: weighted distances to five Boston employment centres
 9. RAD: index of accessibility to radial highways
 10. TAX: full-value property-tax rate per \$10,000
 11. PT: pupil-teacher ratio by town
 12. B: $1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by town
 13. LSTAT: % lower status of the population
 14. MV: Median value of owner-occupied homes in \$1000's (要は家賃)
- 重回帰分析を行い, 2乗誤差と, 家賃に係り性の強い説明変数は何か挙げよ.