

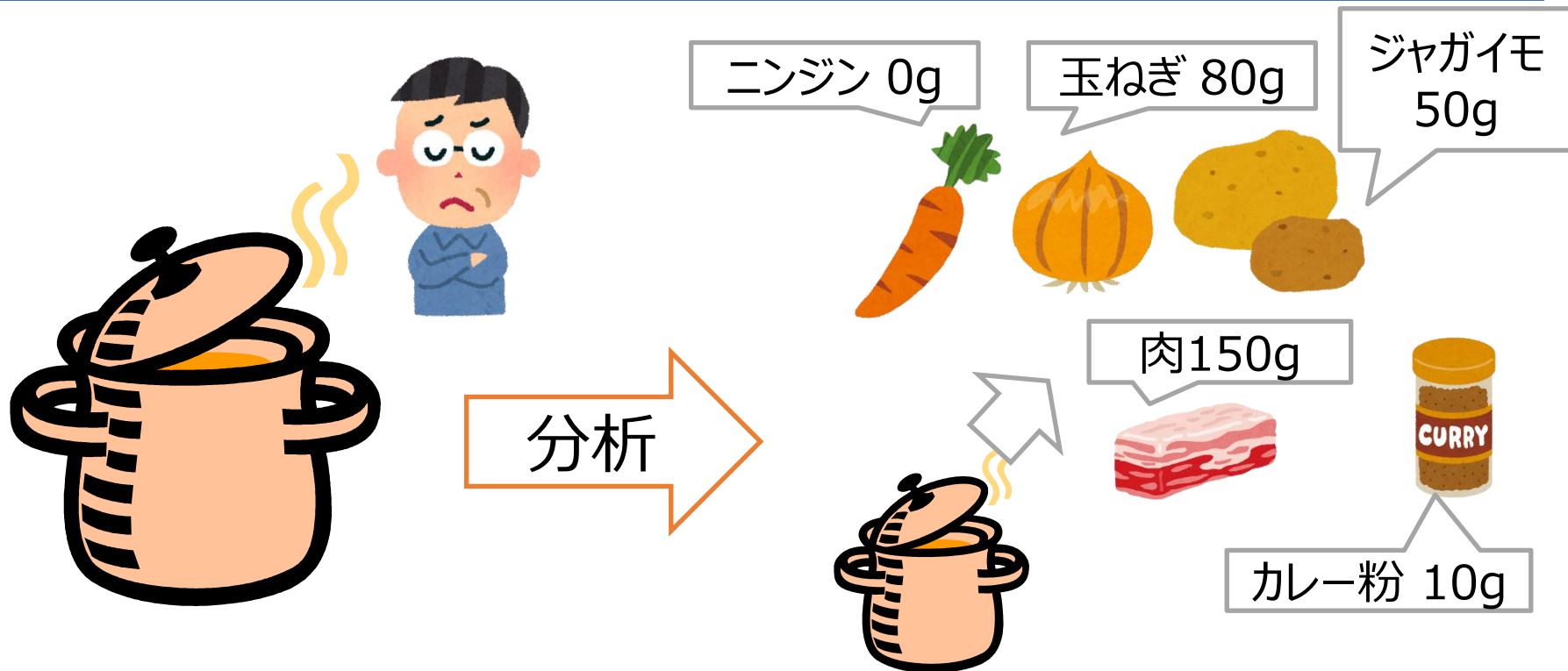
# 「主成分分析(PCA)」

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

# 分析とは？

ベクトルの分解と合成

# 分析の例： カレーにジャガイモは何グラム入ってる？



色々なものが混ざっているので  
パッと見ただけでは  
どんなカレーかわからない

何がどれくらい混ざっているか  
わかったら、どんなカレーか  
クリアになる！



# 参考：「温泉の成分表」

## これがあるからどんな温泉かがわかる！



### 5. 試料 1 kg 中の成分:分量および組成

#### (イ) 陽イオン

成分	ミリグラム (mg)	ミリバール (mval)	ミリバール% (mval%)
ナトリウムイオン (Na <sup>+</sup> )	3692	160.6	71.82
カリウムイオン (K <sup>+</sup> )	201.5	5.15	2.30
マグネシウムイオン (Mg <sup>2+</sup> )	62.7	5.16	2.31
カルシウムイオン (Ca <sup>2+</sup> )	1056	52.69	23.56
鉄(II)イオン(フェロイオン) (Fe <sup>2+</sup> )	0.7	0.03	0.01
陽イオン 計	5013	223.6	100

#### (ロ) 陰イオン

成分	ミリグラム (mg)	ミリバール (mval)	ミリバール% (mval%)
ふっ化物イオン (F <sup>-</sup> )	1.9	0.10	0.05
塩化物イオン (Cl <sup>-</sup> )	7634	215.3	98.99
臭化物イオン (Br <sup>-</sup> )	8.4	0.11	0.05
よう化物イオン (I <sup>-</sup> )	8.8	0.07	0.03
硫化水素イオン (HS <sup>-</sup> )	7.7	0.23	0.11
硫酸イオン (SO <sub>4</sub> <sup>2-</sup> )	18.0	0.37	0.17
炭酸水素イオン (HCO <sub>3</sub> <sup>-</sup> )	80.1	1.31	0.60
炭酸イオン (CO <sub>3</sub> <sup>2-</sup> )	0.2	0.01	0.00
陰イオン 計	7759	217.5	100

#### (ハ) 遊離成分

##### 非解離成分

成分	ミリグラム (mg)	ミリモル (mmol)
メタけい酸 (H <sub>2</sub> SiO <sub>3</sub> )	72.3	0.93
メタほう酸 (HBO <sub>2</sub> )	67.0	1.53
非解離成分 計	139.3	2.46

溶存物質(ガス性のものを除く): 12.91 g/kg

##### 溶存ガス成分

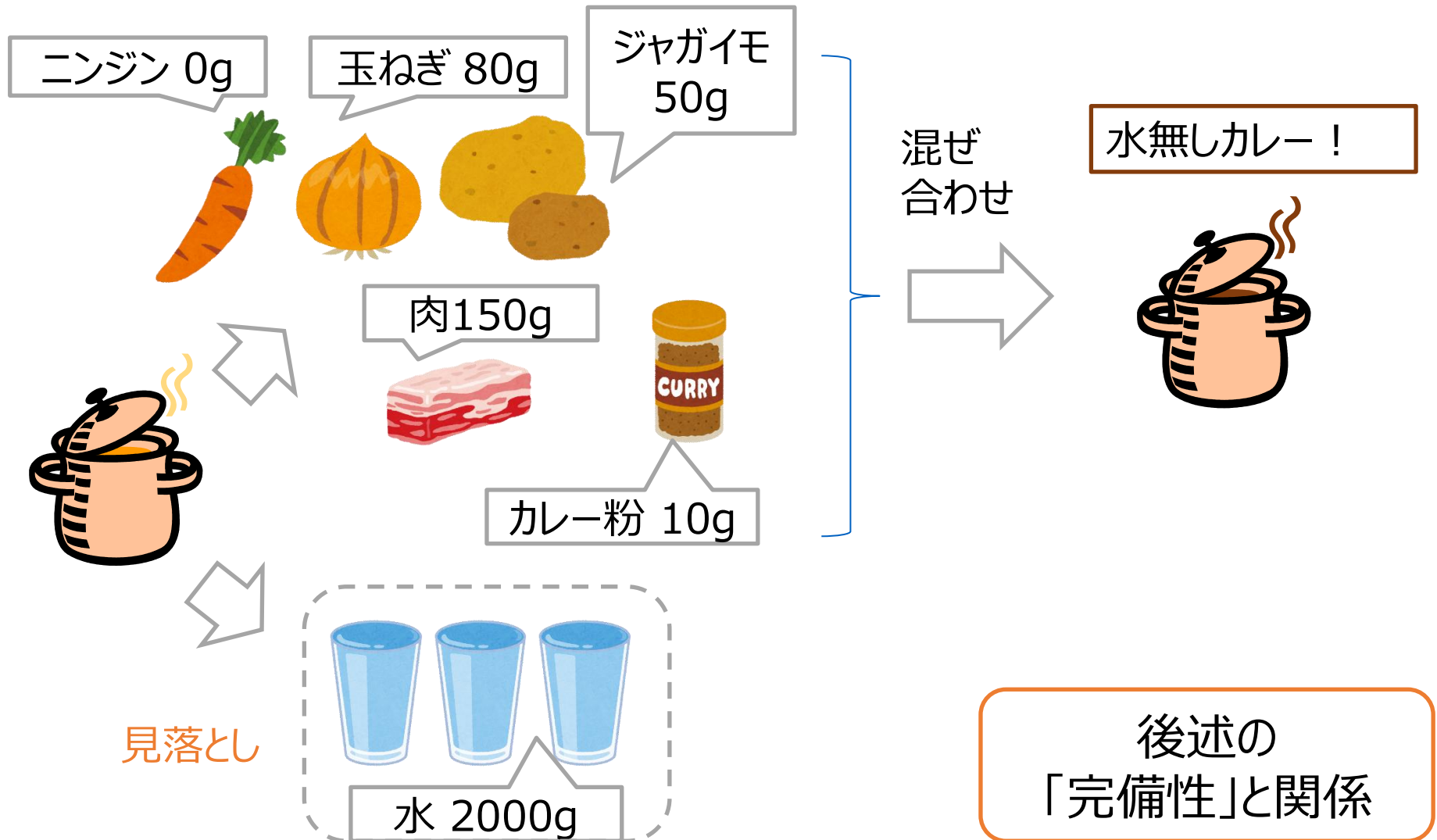
成分	ミリグラム (mg)	ミリモル (mmol)
遊離二酸化炭素 (CO <sub>2</sub> ) (遊離炭酸)	4.8	0.11
遊離硫化水素 (H <sub>2</sub> S)	2.2	0.06
溶存ガス成分 計	7.0	0.17

成分総計: 12.92 g/kg



# 分析の際にケアすべきポイント(1/4)

## 基本的に見落としはNG



# 分析の際にケアすべきポイント(2/4) 分析項目に重複がないほうが良いだろう

ニンジン 0g



玉ねぎ 80g



ジャガイモ  
50g

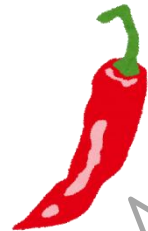


肉150g



カレー粉 10g

実はカレー粉の中に入っている成分



唐辛子 2g



クミン 1g

水 2000g

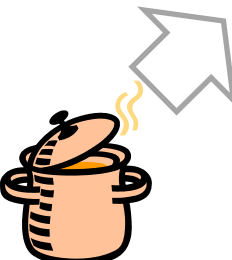


混ぜ  
合わせ

やたらスパシーな  
カレー！



後述の  
「直交性」と関係



# 分析の際にケアすべきポイント(3/4) 分析する単位は統一したほうがよいだろう

ニンジン 0本

玉ねぎ 80g

ジャガイモ  
30 cm<sup>3</sup>

肉 0.15 kg

水 3カップ

比べにくい!

CURRY

カレー粉 10000 mg

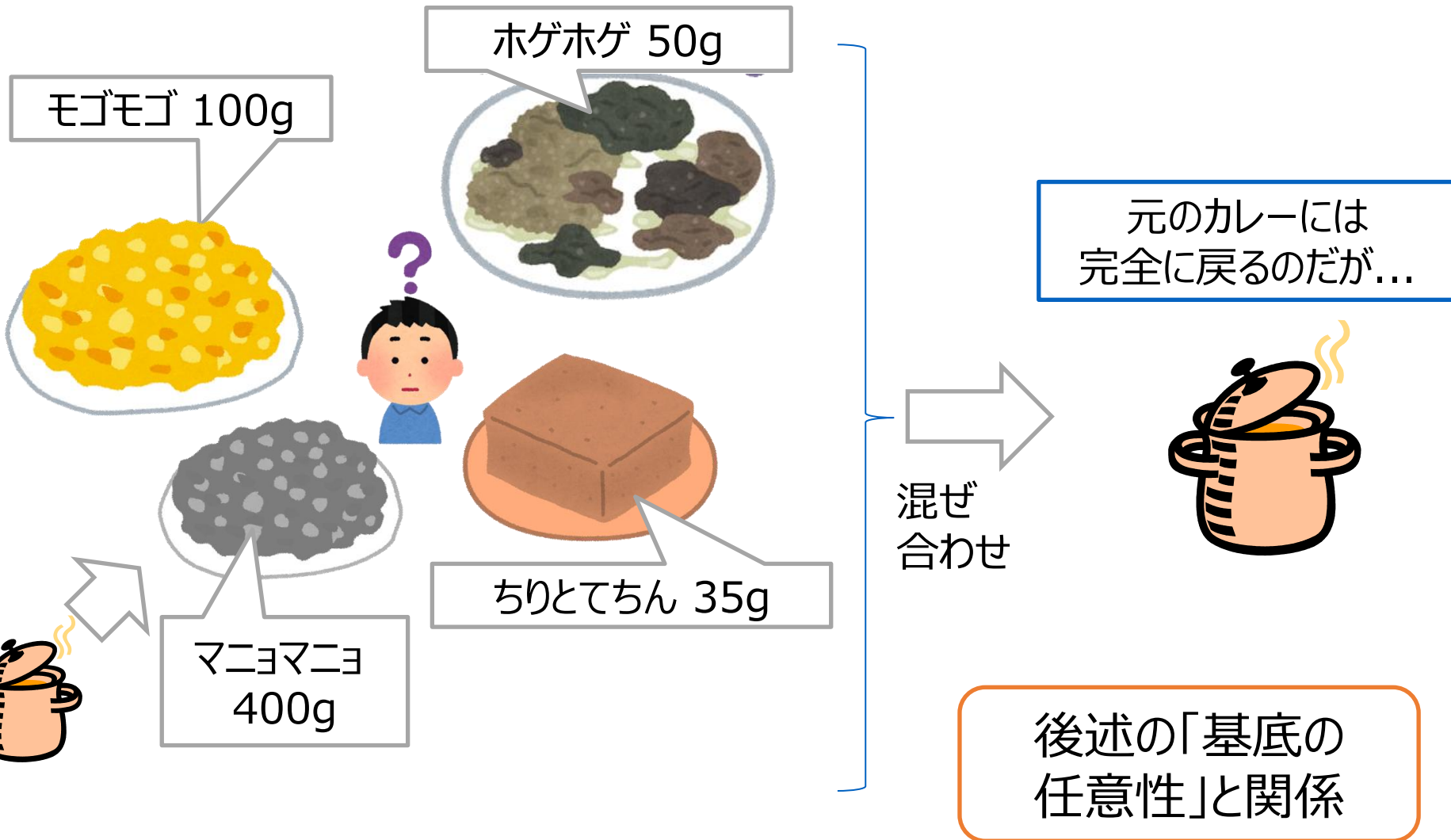
元のカレーには  
完全に戻るのだが...

混ぜ  
合わせ

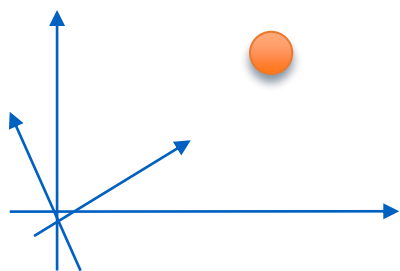
後述の  
「正規性」と関係

# 分析の際にケアすべきポイント(4/4)

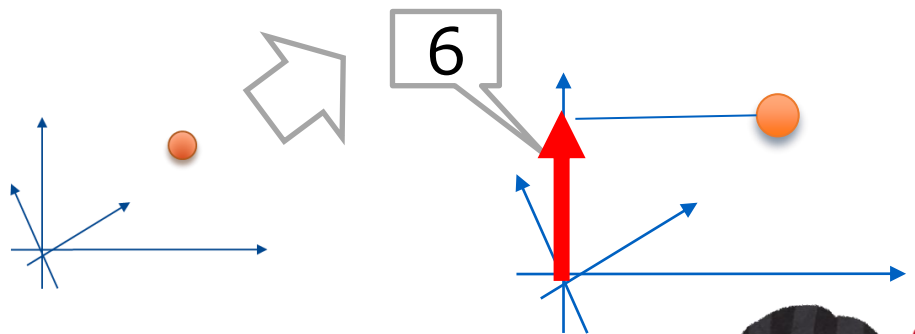
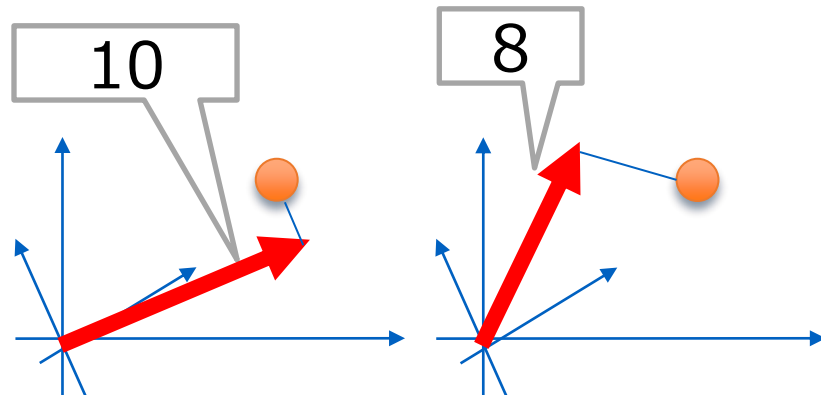
## 解釈容易な成分に分解したほうがよい



# これからの話： カレーからベクトルへ



分析



色々なものが混ざっているので  
パッと見ただけでは  
どんな高次元ベクトルかわからない

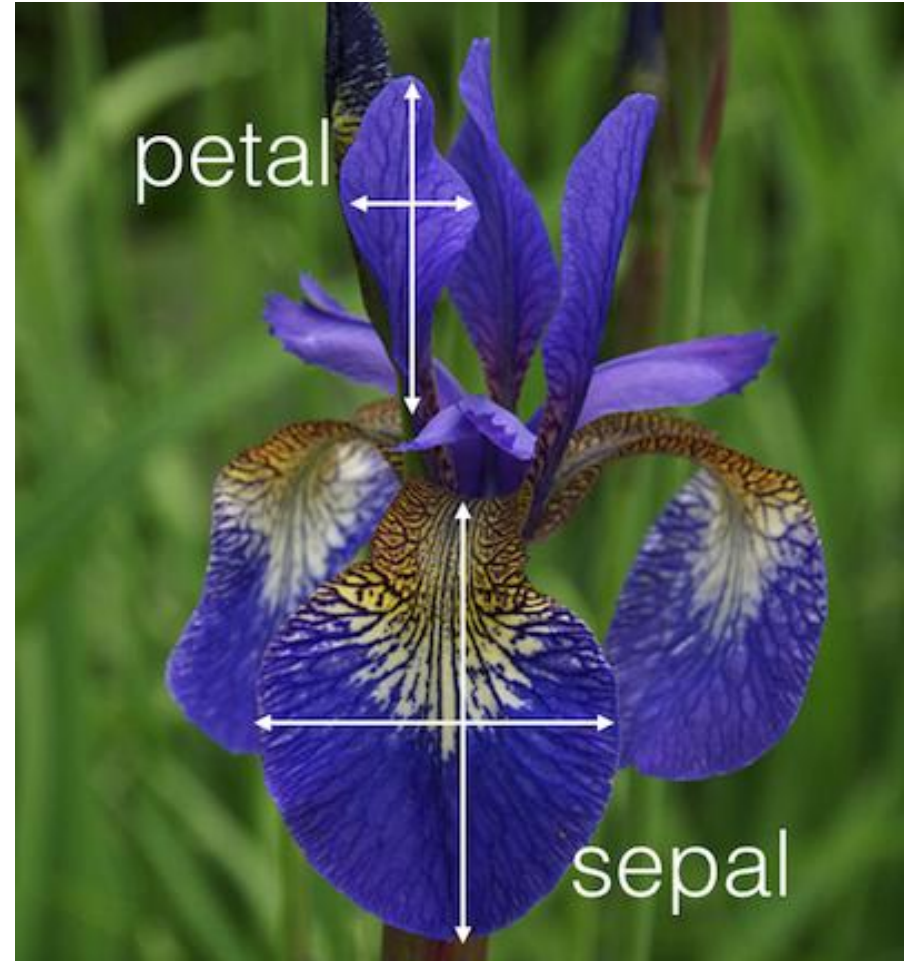
何がどれくらい混ざっているか  
わかったら、**どんな高次元  
ベクトル**がクリアになる！



# 主成分分析

# 早速ですが、具体例

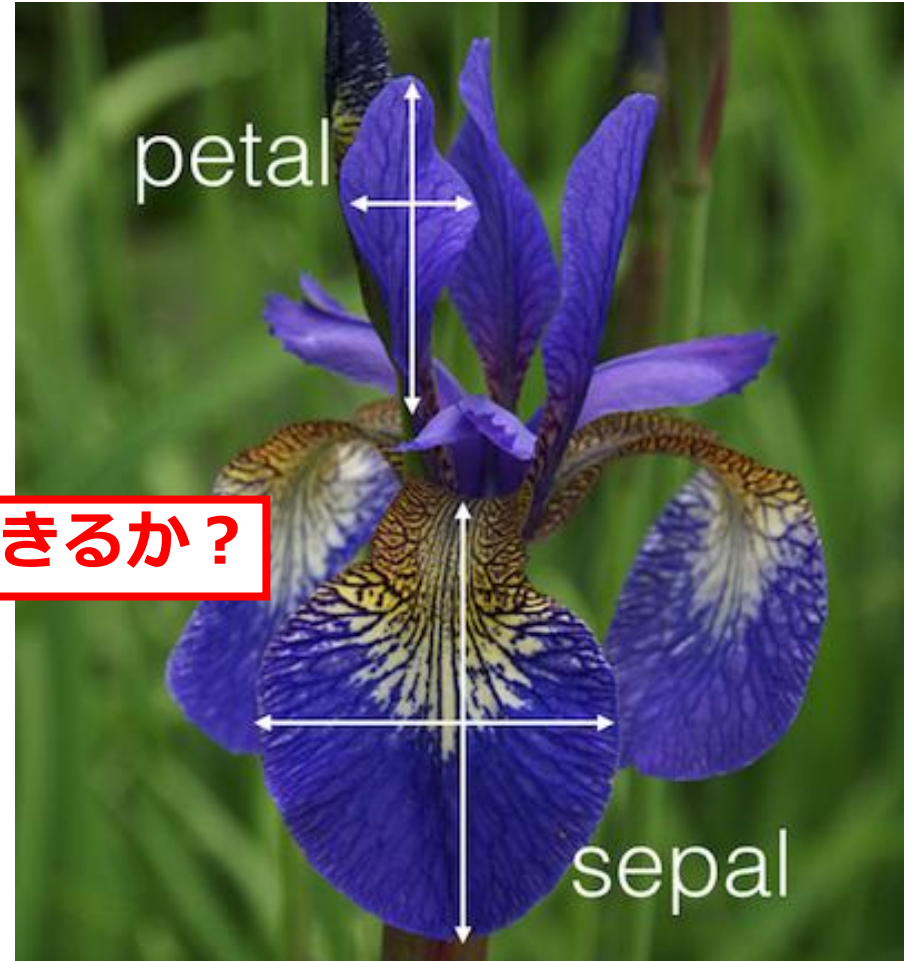
- アイリスデータ
- 3クラス
  - Iris-setosa
  - Iris-versicolor
  - Iris-virginica
- 指標（4次元）
  - sepal length
  - sepal width
  - petal length
  - petal width



# 早速ですが、具体例

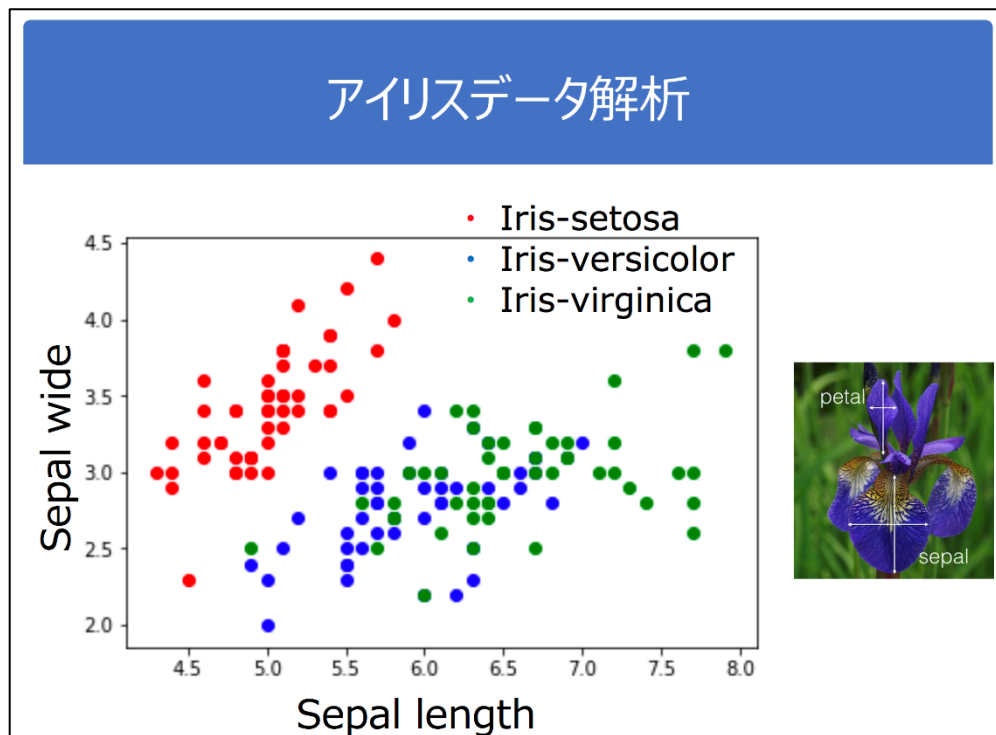
- アイリスデータ
- 3クラス
  - Iris-setosa
  - Iris-versicolor
  - Iris-virginica
- 指標（4次元）
  - sepal length
  - sepal width
  - petal length
  - petal width

→2次元にできるか？



# 好きな2次元を「選ぶ」ことも可能だが・・・

例：アイリスデータの二次元プロット



**データを的確に表現できる2次元を「作る」→ PCAの出番**

※2次元でなくても、好きなd次元でよいです

# Python で PCA

```
from __future__ import division
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.decomposition import PCA

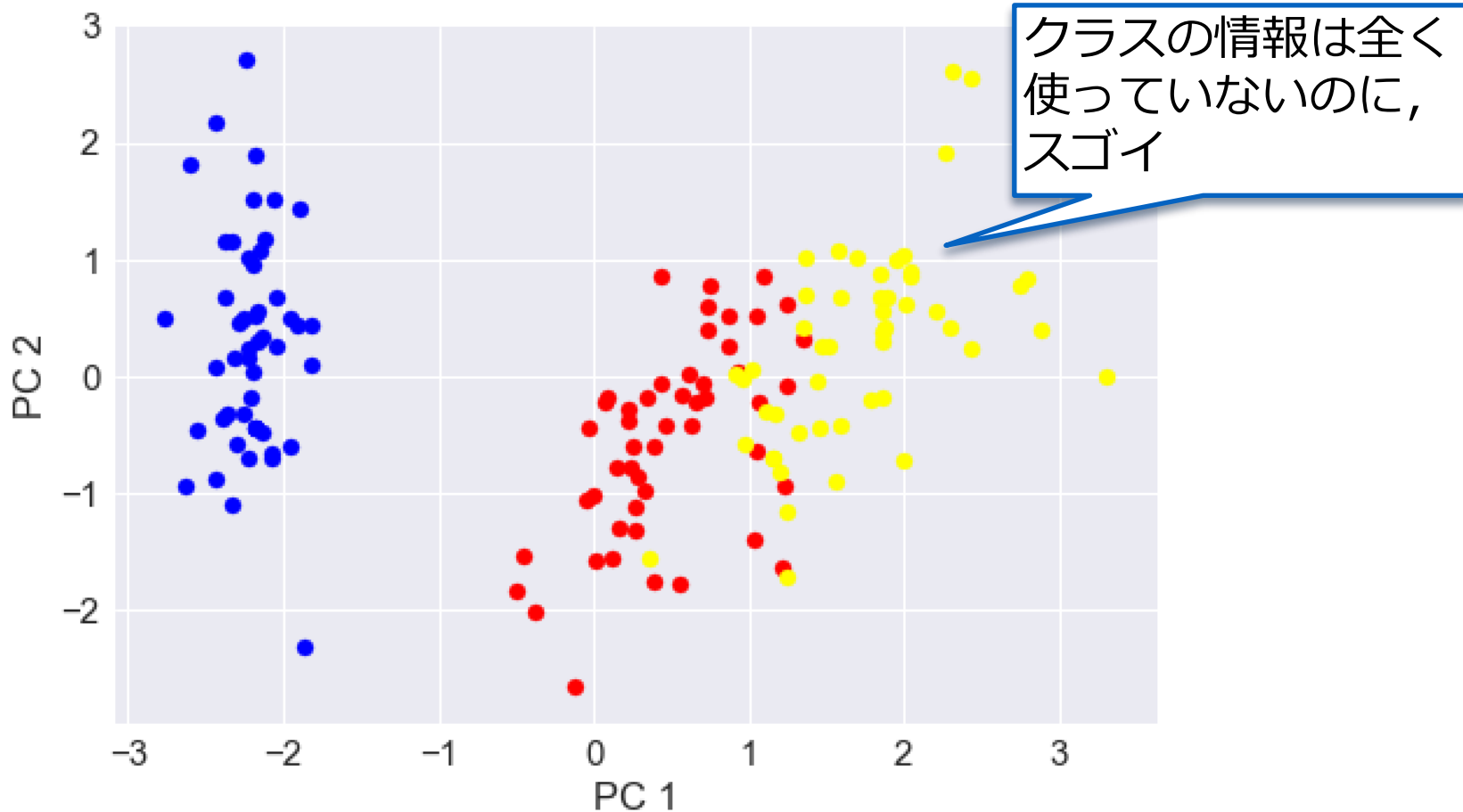
iris = pd.read_csv('./iris.csv') ←データの読み込み & 4次元をXに
X = iris.drop('class', axis=1)
Y = iris['class']

X_scaled = (X-X.mean())/X.std() ←データの正規化（後述）

pca = PCA(n_components=2) ←PCAのパラメータをセット：（2次元）
pca.fit(X_scaled) ←PCAで2次元にするルール作り
reduced_iris = pca.fit_transform(X_scaled) ←2次元にしたXをゲット
colors = ['blue', 'red', 'Yellow'] ←プロット
uniqueY = pd.unique(Y)
for i in range(len(uniqueY)):
    Yi = uniqueY[i]
    color = colors[i]
    plt.scatter(reduced_iris[np.where(Y == Yi), 0],
                reduced_iris[np.where(Y == Yi), 1],c=color)

plt.xlabel('PC 1')
plt.ylabel('PC 2')
```

# プロットしてみると. . .



横軸：第一主成分，縦軸：第二主成分

# データを「的確に」表現できているか？

- 各主成分の「寄与率」

```
print(pca.explained_variance_ratio_)  
[0.72770452 0.23030523]
```

→ 第一主成分で 72.7%, 第二主成分と合わせて 95.8% を表現できている！（データの広がり具合 = 分散を95.8%説明できている）

- 各主成分の「中身」

```
print(pca.components_)  
[[ 0.52237162 -0.26335492 0.58125401 0.56561105]  
 [ 0.37231836 0.92555649 0.02109478 0.06541577]]
```

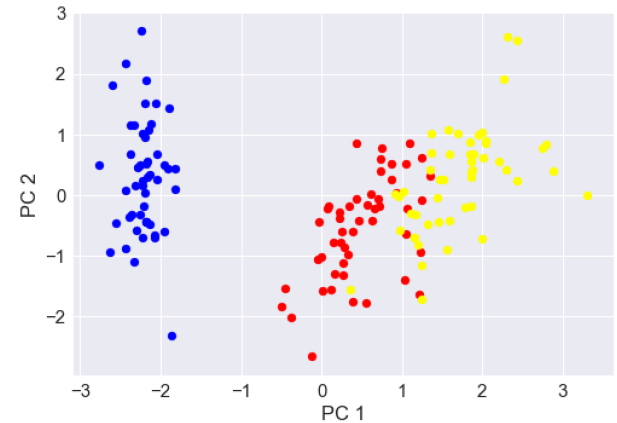
→ 第一主成分は

$0.522\dots \times (\text{sepal length}) - 0.263\dots \times (\text{sepal width})$   
 $+ 0.581\dots \times (\text{petal length}) + 0.565\dots \times (\text{petal width})$

で構成されている。各係数を主成分負荷量とよぶ

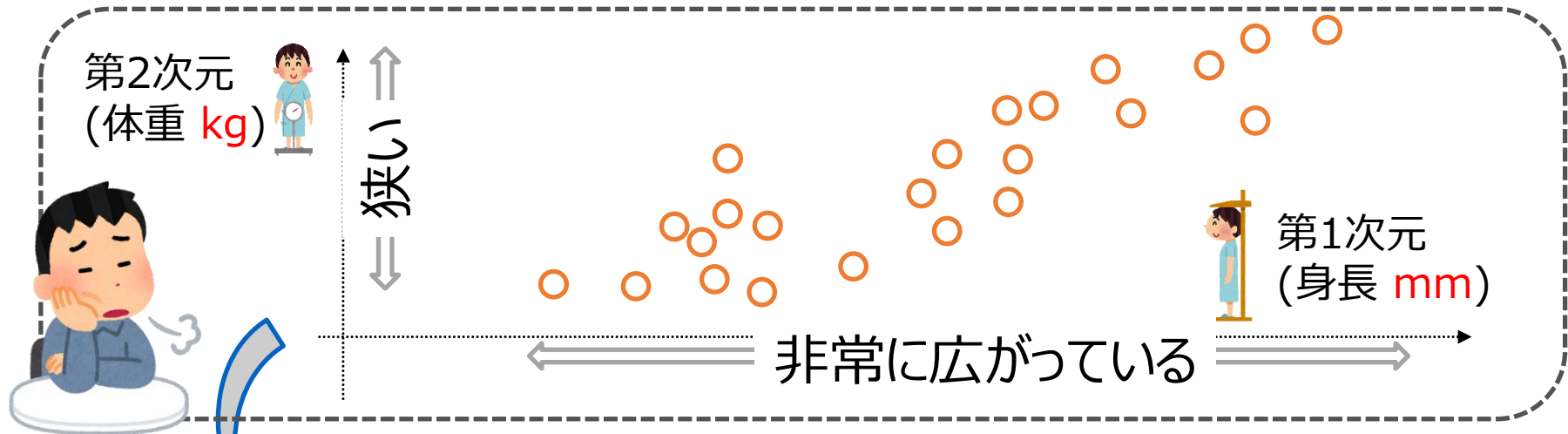
# Features of PCA

- Compact representation of the entire  $d$ -dimensional vector distribution with  $d \sim (< \text{dimensional number } d)$  principal components.
  - In this sense, it is similar to the representative vector of clustering
- Sometimes  $d \sim \ll d$ 
  - If you want to do it, you can make  $d \sim$  small as much as you want.
  - However, if you do too much, you will not be able to grasp the structure of the distribution well.
- It can also be regarded as the principal component direction = major variation from the mean

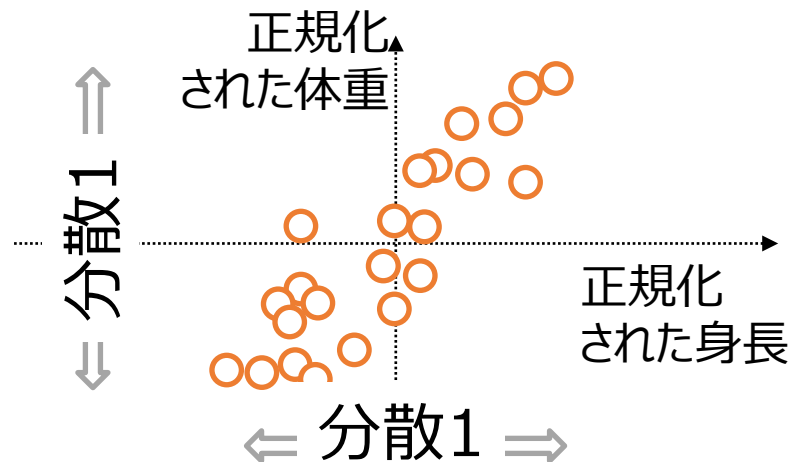


# Normalization: What to do first in many data analyses, not just PCA

- "Normalize" the **mean to zero** and the **variance to 1**!



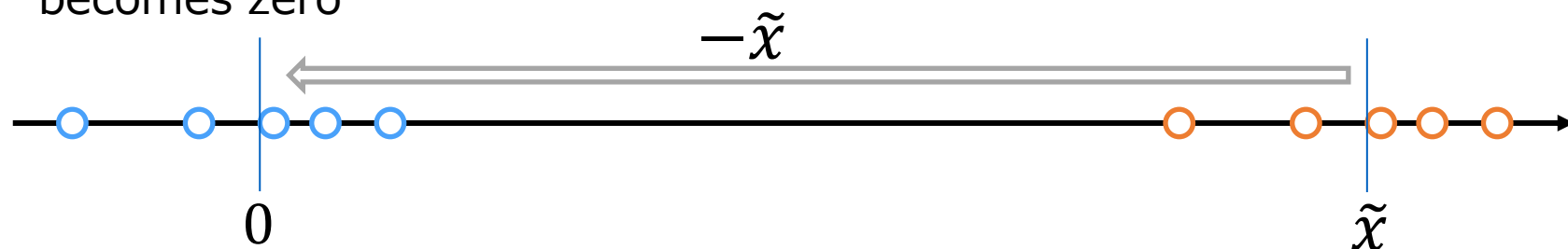
Stretch and contract each axis to achieve the same degree of "spread"



# How to make the mean zero, the variance 1?

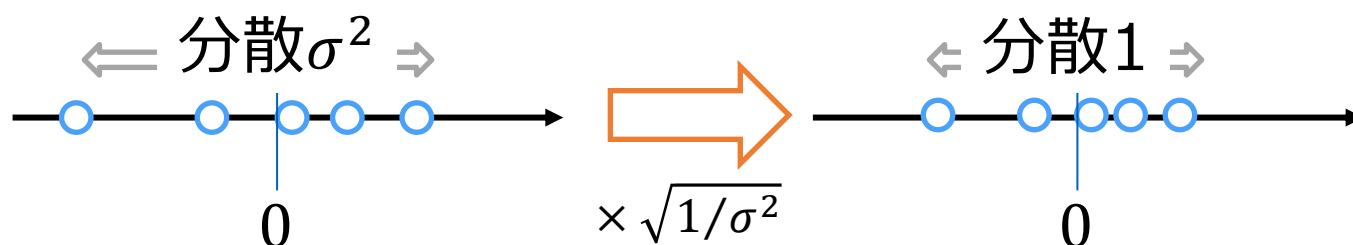
- First, set the average to zero

- If you subtract the current average from all the values, the average becomes zero



- Next, set the variance to 1!

- I said before, "If you change the value from  $x_i$  to  $\alpha x_i$ , the variance will be  $\alpha^2 \sigma^2$ "
- If I want  $\alpha^2 \sigma^2 = 1$ , then multiply  $x_i$  by  $\alpha = \sqrt{1/\sigma^2}$  is OK!



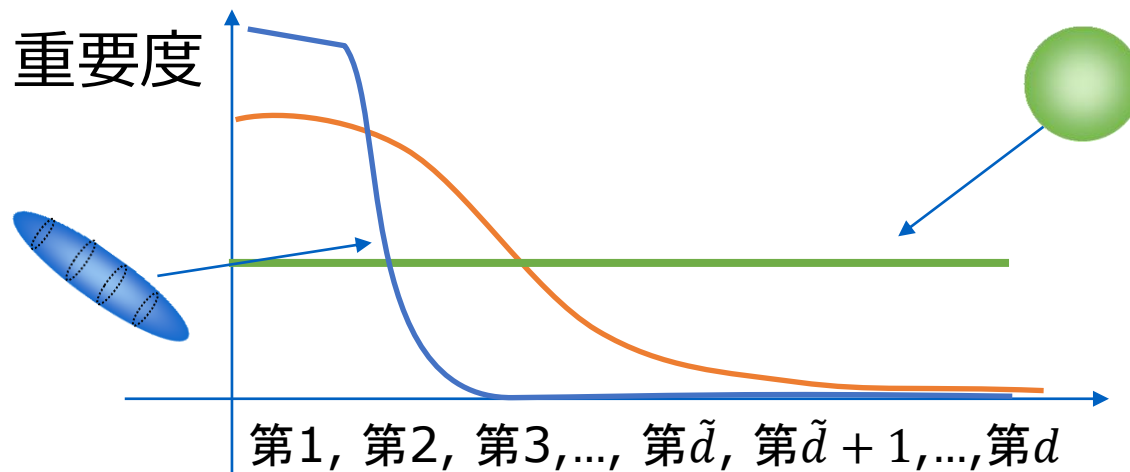
# 練習1：

- Irisデータセットの “petal width” を正規化せよ
- 正規化したデータと、正規化前のデータをプロットせよ

主成分分析でわかること

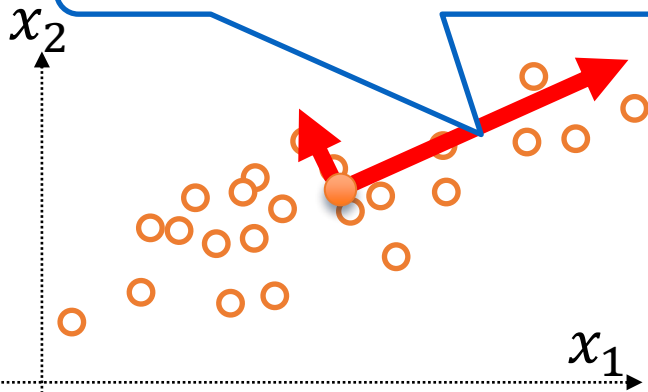
# The importance of each principal component (such as "cumulative contribution rate" or "eigenvalue")

- It is widely spread → high importance
- 1st, 2nd, 3rd... and the importance decreases
- However, there are different ways to go down, and that's very important!
  - As you get used to it, you can imagine the shape of the distribution in this "descending curve"

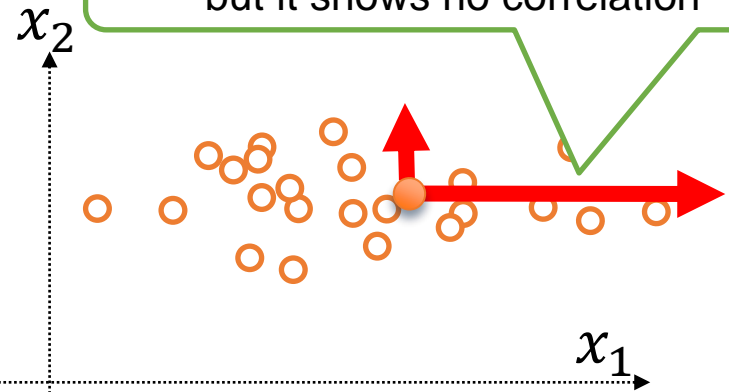


As you get used to it, you'll find out: **Correlation** between the two factors (more on that later)

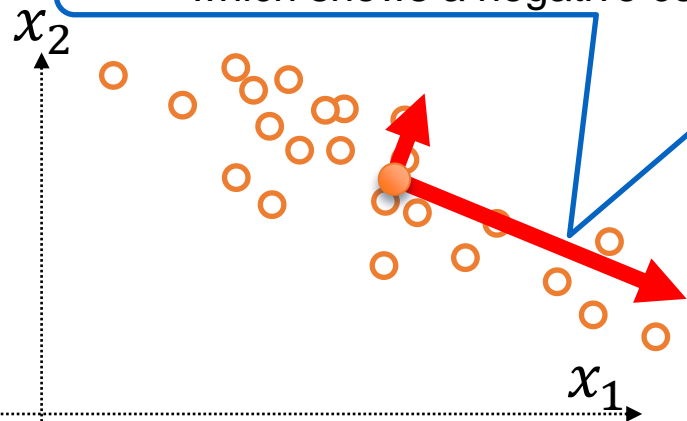
There is a very spread direction (= bias), which shows a positive correlation



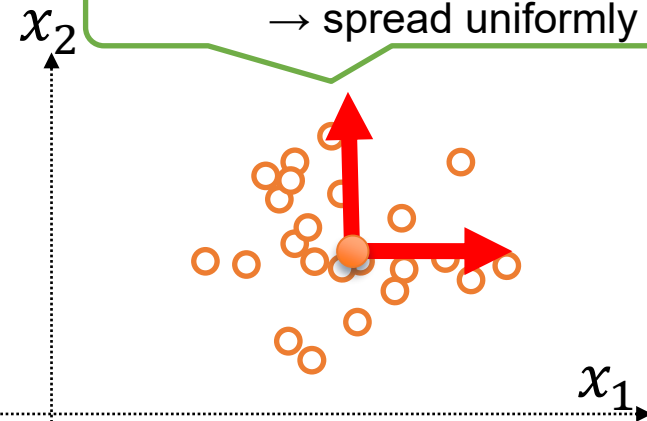
There is a very widespread direction, but it shows no correlation



There is a very widespread direction (= biased), which shows a negative correlation

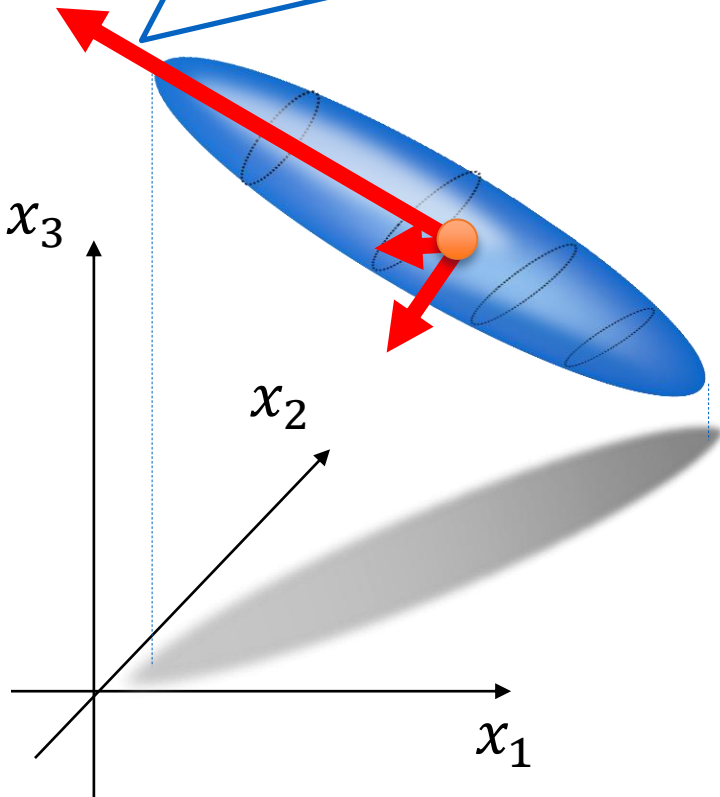


Spherical distribution → uncorrelated  
→ spread uniformly

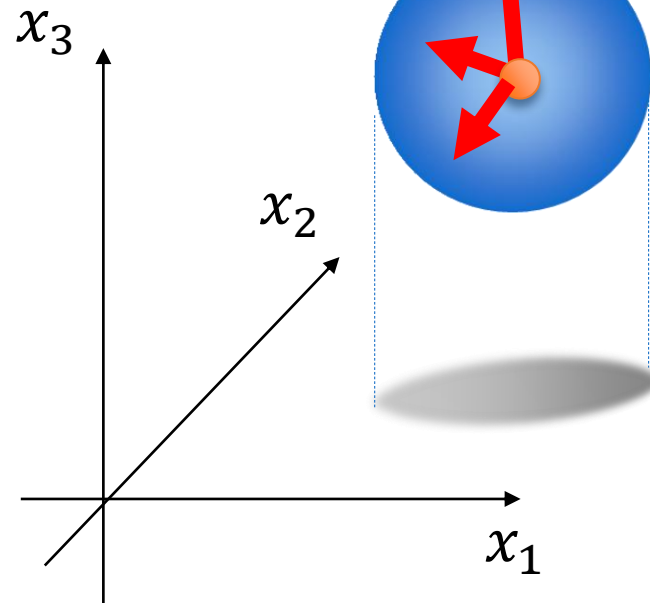


# As you get used to it, you'll find out: Correlation between the two factors (more on that later)

非常に広がっている方向があり(=偏りがあり),  
 $x_1 \rightarrow$ 大,  $x_2 \rightarrow$ 大,  $x_3 \rightarrow$ 小  
( $x_1 \rightarrow$ 小,  $x_2 \rightarrow$ 小,  $x_3 \rightarrow$ 大, と等価)

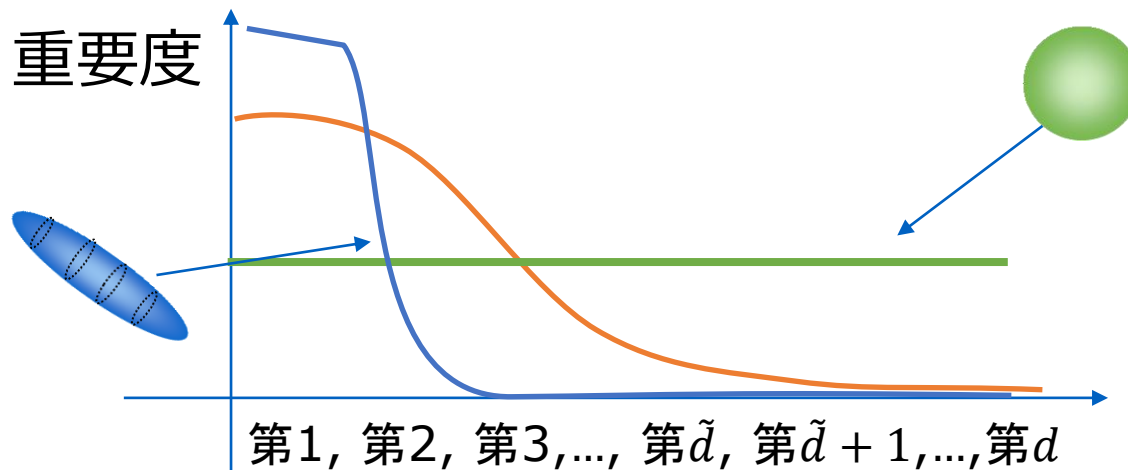


広がり一様  $\rightarrow$  球状分布  
 $\rightarrow$  無相関



## 練習 2 :

- Irisデータに対してPCA ( $n=4$ ) を行え
- 各次元の重要度を大きい順にプロットせよ



# 主成分を求める実際の方法

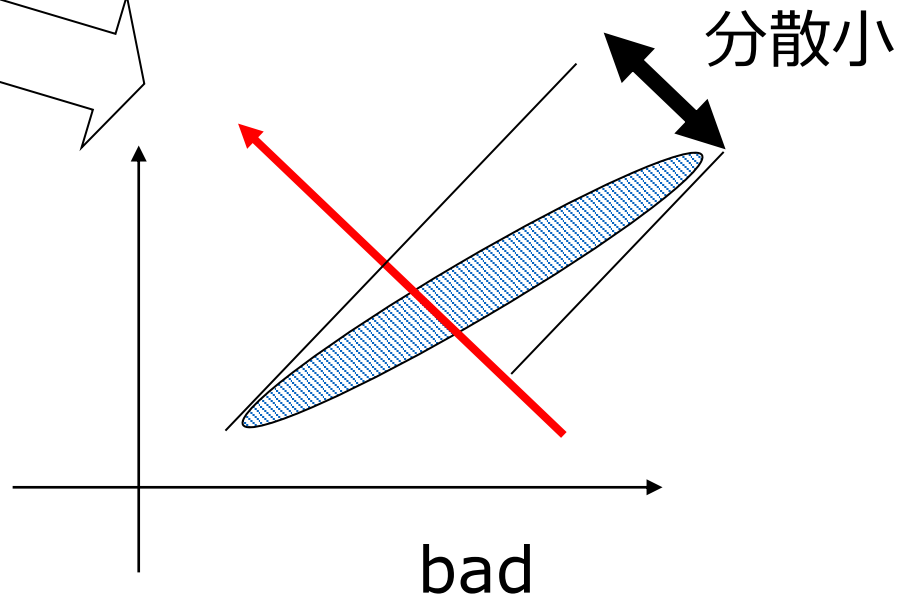
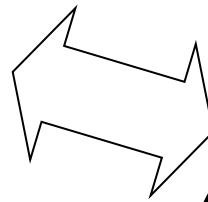
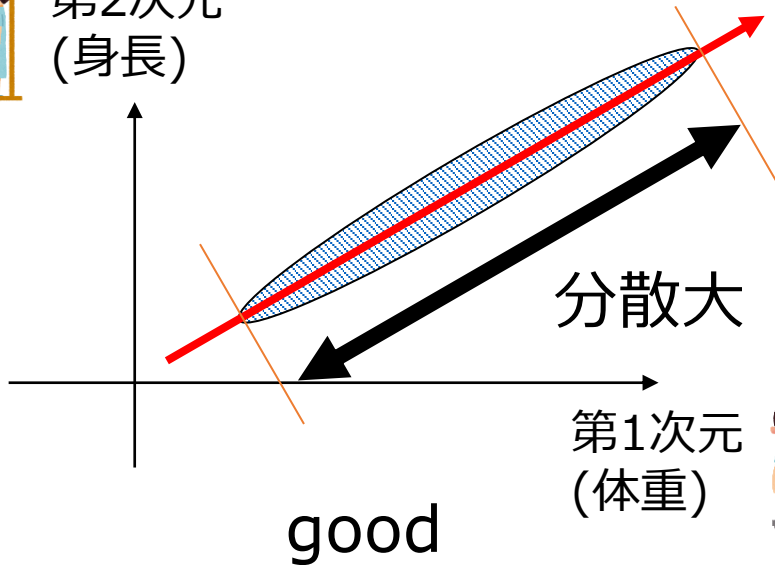
詳細略

# Two criteria for finding the direction of spread (1): The maximum dispersion criterion

- I want to find the main component in the direction of increasing the dispersion as much as possible.

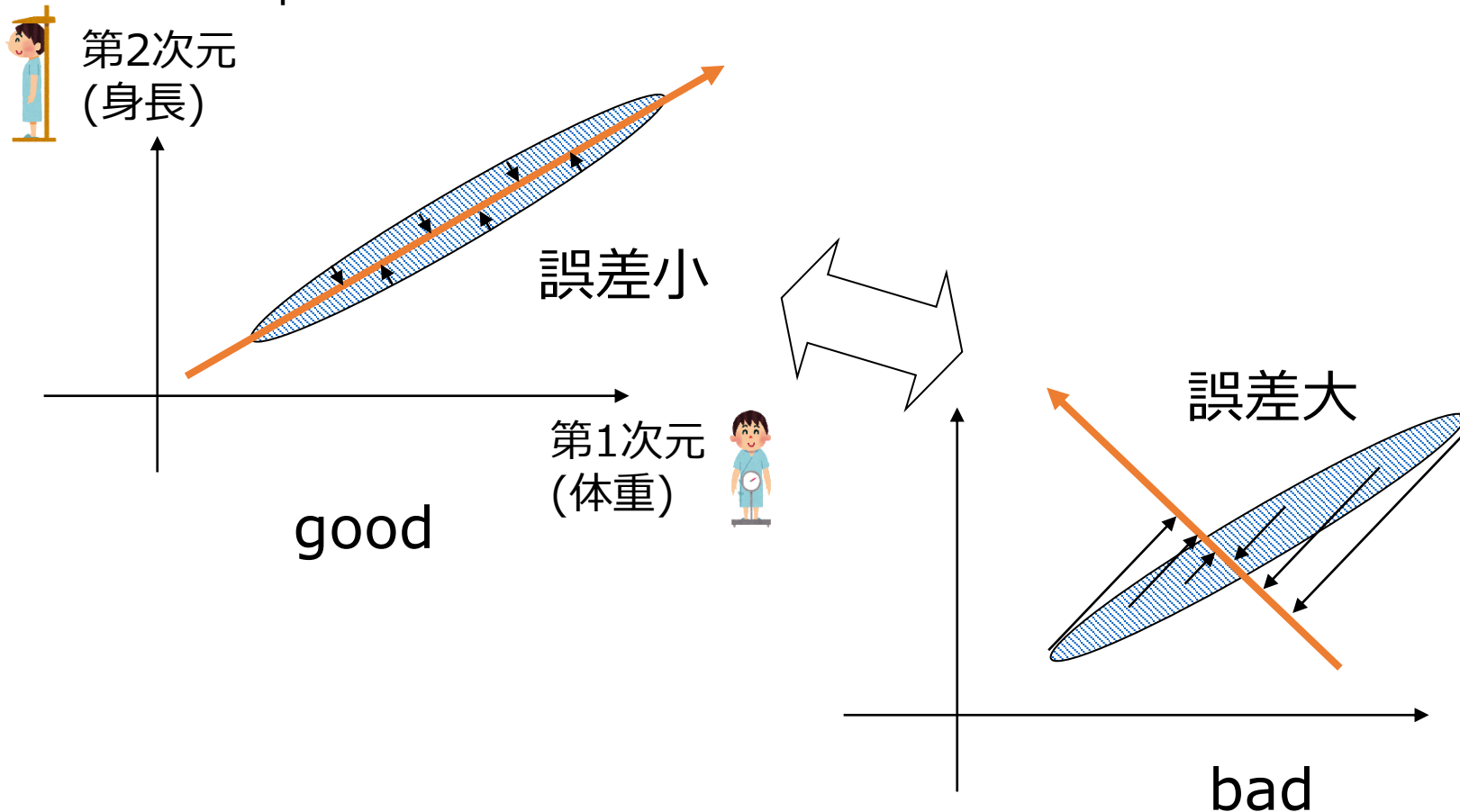


第2次元  
(身長)



# Two criteria for finding the direction of spreading(2): the least squares error criterion

- I want to find the main component in the direction where the error is as small as possible



# Reference: Principal component analysis solution

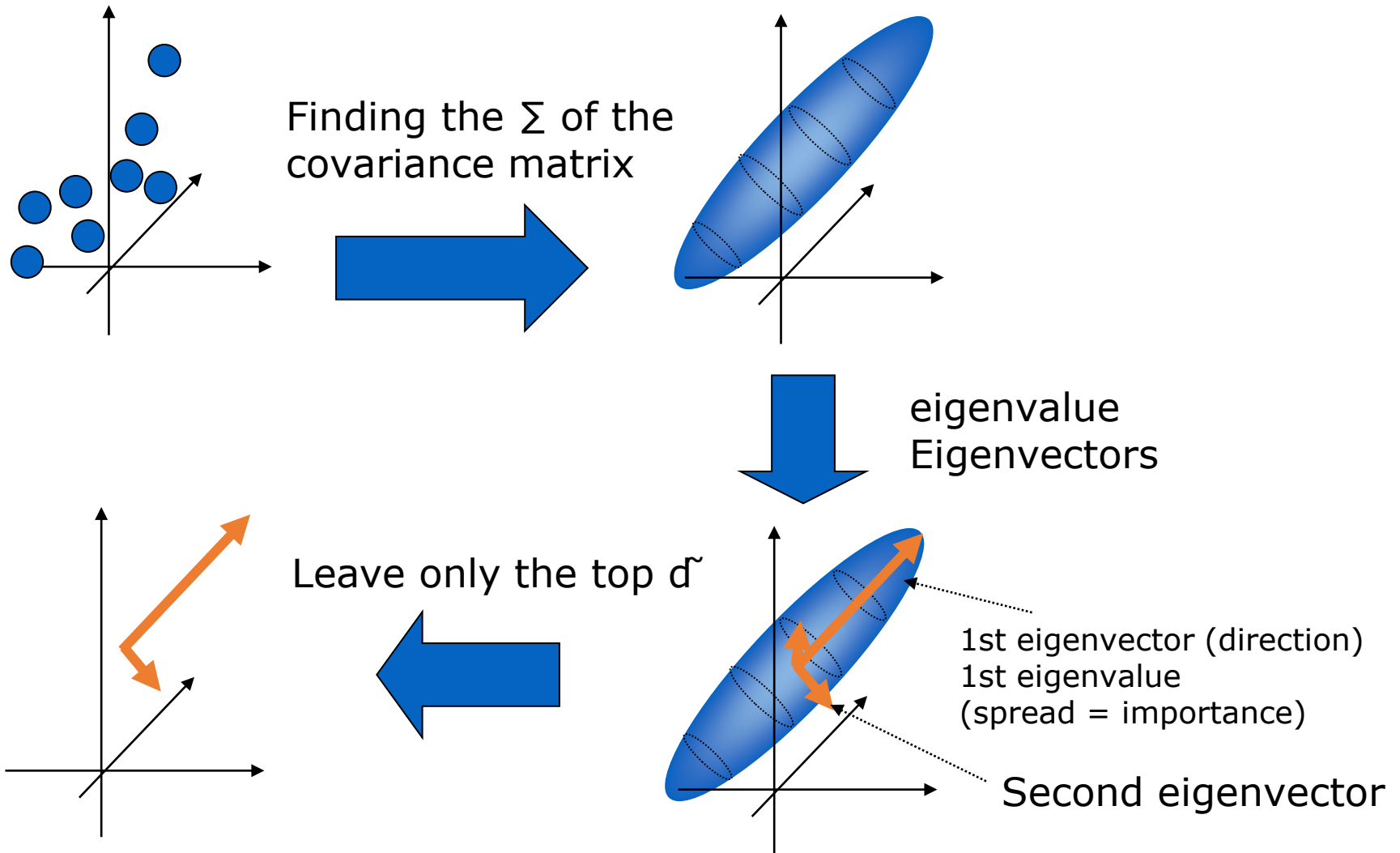
- Dispersion and minimum error, both criteria can be solved as follows
  - Exactly the same main components are required
- Finish in 3 steps
  1. Finding the covariance matrix  $Cov$  from a set of data (each  $d$ -dimensional vector).
  2. Finding the eigenvalues and eigenvectors of  $Cov$
  3. From the large eigenvalue,  $d \sim$  eigenvectors are the main components ( $d \sim$  is determined appropriately)



# 主成分分析の解法（詳細）

完全に覚える必要はないです  
必要なときの参考資料として. . .

# Illustrate the atmosphere of the solution



# Covariance

- Equation (covariance of vector X and vector Y)

$$\text{cov}(X, Y) = \frac{\sum_{i \in [0:n-1]} (X_i - \text{mean}(X))(Y_i - \text{mean}(Y))}{n}$$

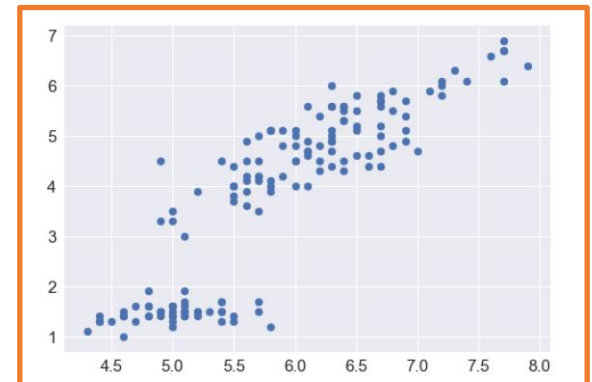
- Meaning of covariance
  - Signs are the most important:
    - When it is positive ( $>0$ )... When the  $X_i$  is large  $Y_i$  it tends to be large.
    - When negative ( $<0$ )... When the  $X_i$  is large  $Y_i$  it tends to be small.
    - When 0 ( $=0$ )...  $X_i$  No tendency to "large"  $Y_i$
    - In fact, we learned last time that each variable is divided by the standard deviation. It will be a correlation coefficient!

# Specific examples of covariance

- If you look for the covariance matrix of sepal\_length and petal\_length in IRIS data...

```
sepal_length = np.array(iris['sepal length'])
petal_length = np.array(iris['petal length'])
CC = np.cov(sepal_length, petal_length)
print(CC)
plt.scatter(sepal_length, petal_length)
```

```
[[0.68569351 1.27368233]
 [1.27368233 3.11317942]]
```



$\text{cov}(\text{sepal\_length}, \text{petal\_length}) > 0$ :  
sepal\_length が大きいとき, petal\_length も大きい傾向にある

# Eigenvalues and eigenvectors

- When  $Ax = \lambda x$  holds,  $x$  is the eigenvector of the matrix  $A$

- $\lambda$  is the eigenvalue of  $A$

- Example: Matrix  $A = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}$

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 11 \\ 5 \end{bmatrix}$$

固有ベクトルでない

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \end{bmatrix} = 4 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

固有ベクトル

ちなみに・・・

A matrix is a pair of vectors

$$A = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}$$

Aはベクトル(2,3)とベクトル(2,1)の組

# How to find eigenvalues and eigenvectors

- You can calculate it as follows

```
# 固有値と固有ベクトル
A = np.matrix([[2,3], [2,1]])

from numpy import linalg as LA
(eigenvalues, eigenvectors) = LA.eig(A)
```

固有値      固有ベクトル

```
print(eigenvalues)
```

```
[ 4. -1.]
```

```
print(eigenvectors)
```

```
[[ 0.83205029 -0.70710678]
 [ 0.5547002  0.70710678]]
```

行列 A:

$$A = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}$$

An eigenvector is not necessarily unique  
When sorted in order of highest eigenvalue, the first principal component is the largest

# Eigenvalues and eigenvectors (confirmation)

- If you look at it to see if it really fits...

`e_vec1 = eigenvectors[:,0]` 第一固有ベクトル

`e_val1 = eigenvalues[0]` 第一固有値

```
print(np.dot(A, e_vec1))
```

```
[[3.32820118]  
 [2.21880078]]
```

Inner product of matrices and vectors

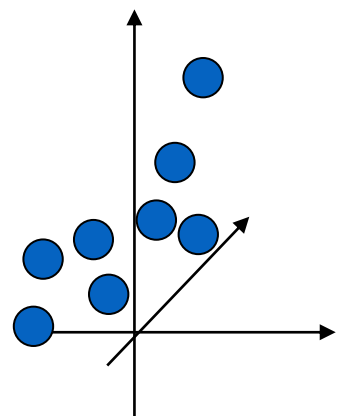
```
print(e_val1*e_vec1)
```

```
[[3.32820118]  
 [2.21880078]]
```

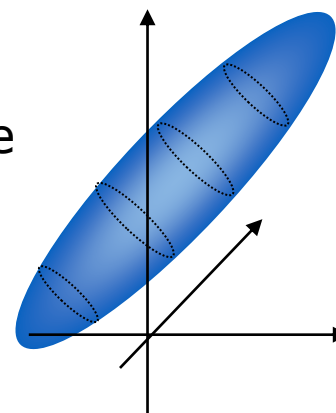
Meets the conditions of the eigenvector

$$A \times e_1 = \lambda \times e_1$$

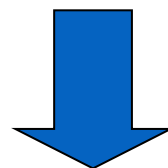
# Illustrate the atmosphere of the solution (reposted)



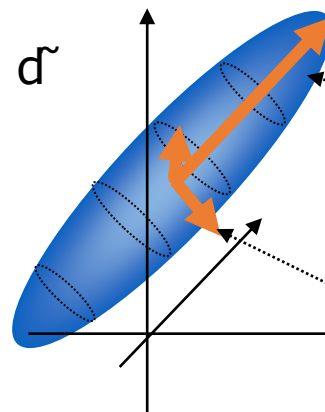
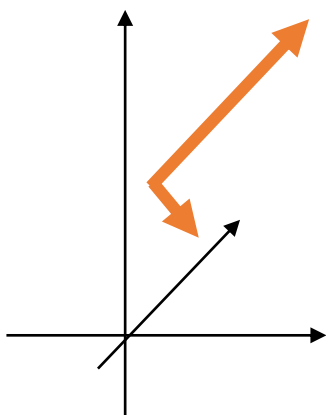
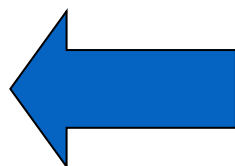
Finding the  $\Sigma$  of the covariance matrix



eigenvalue  
Eigenvectors



Leave only the top  $d$



1st eigenvector (direction)  
1st eigenvalue  
(spread = importance)

Second eigenvector

# Try it with Iris data (no normalization)

- Let's visualize the distribution of Sepal length and Petal length and the primary principal components. (Normalize only the average value for clarity)

```
# sepal_length と petal_length の共分散行列の固有値
# Read iris dataset
iris = pd.read_csv('./iris.csv')
X = iris.drop(['sepal width', 'petal width', 'class'], axis=1)

# Normalize the data
#X_scaled = (X-X.mean())/X.std()
X_scaled = (X-X.mean())

# Transform input data into an np.array
Y = np.array(X_scaled)

# Compute the covariance matrix
cov_matrix = np.cov(Y.transpose())

# Compute the eigen values and eigen vectors
(l,V) = np.linalg.eig(cov_matrix)

print("Explained variance ratio: ", l/l.sum())
print("the 2 principal vectors: \n", V.transpose())
```

# Try it with Iris data (no normalization)

- Let's compare it to the PCA function

```
Explained variance ratio: [0.03686636 0.96313364]
the 2 principal vectors:
[[-0.91920275  0.39378459]
 [-0.39378459 -0.91920275]]
```

The order is the same if sorted in order of greatest

```
pca = PCA(n_components=2)
pca.fit(X_scaled)
print(pca.explained_variance_ratio_)
print(pca.components_)

[0.96313364 0.03686636]
[[ 0.39378459  0.91920275]
 [-0.91920275  0.39378459]]
```

# Try it with Iris data (no normalization)

## ● Let's visualize it

```
A = pca.components_
```

```
# 第一主成分
```

```
a = A[0][1]/A[0][0]
```

```
xx = [min(X_scaled['sepal length']) + 0.01*(max(X_scaled['sepal length'])  
-min(X_scaled['sepal length']))*i for i in range(101)]
```

```
yy = [xx[i]*a for i in range(len(xx))]
```

```
# 第二主成分
```

```
b = A[1][1]/A[1][0]
```

```
xx2 = [min(X_scaled['sepal length']) + 0.01*(max(X_scaled['sepal length'])  
-min(X_scaled['sepal length']))*i for i in range(101)]
```

```
yy2 = [xx2[i]*b for i in range(len(xx2))]
```

```
# 表示
```

```
plt.scatter(X_scaled['sepal length'],X_scaled['petal length'])
```

```
plt.scatter(xx,yy)
```

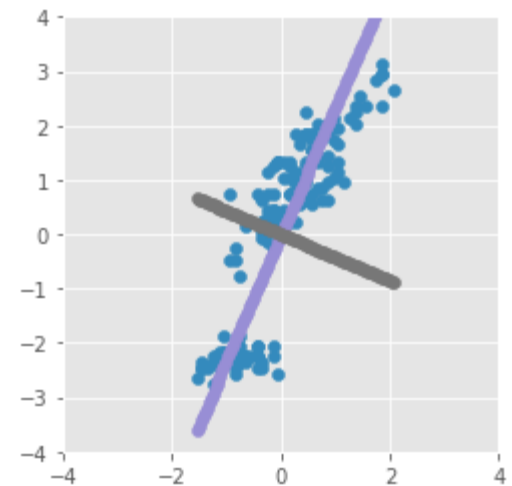
```
plt.scatter(xx2,yy2)
```

```
plt.axis('scaled')
```

```
plt.xlim([-4,4])
```

```
plt.ylim([-4,4])
```

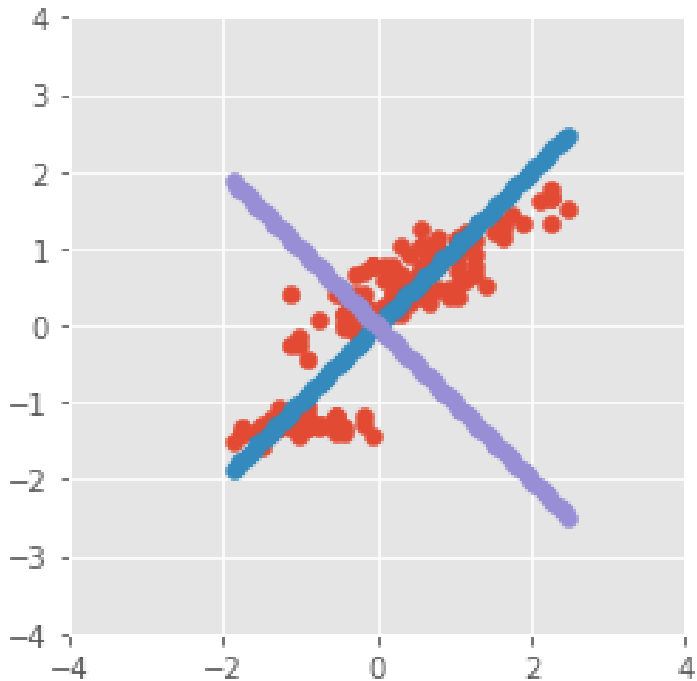
第二主成分



第一主成分

# Try it out with Iris data (with normalization)

傾き : 1



```
X_scaled = (X-X.mean())/X.std()
```

```
[ [ 0.93587708  0.06412292]
  [ 0.70710678  0.70710678]
  [-0.70710678  0.70710678]]
```

Analyze the contribution rate  
of the principal component

## 練習3：

- Irisデータセットを `np.array()` に変換
- `np.cov` を使って共分散行列を求める
- `np.linalg.eig` を使って固有値と固有ベクトルを求める

# 演習

# 演習1

- 次のデータを正規化しなさい（平均0, 分散1）
  - ais.csvの ('Ht', 'Wt', 'LBM', 'BMI')  
(`X = ais[['Ht', 'Wt', 'LBM', 'BMI']]`)
- 各データを主成分分析し, 第二主成分まででデータをプロットしなさい
  - 発展: 性別ごとに色を変えてみよう (P14を参考)
- 主成分の寄与率をプロットし、第何主成分までとれば、計90%の寄与率になるか調べよ

# 演習1

- Normalize the following data (mean 0, variance 1)
  - ais.csv ('Ht', 'Wt', 'LBM', 'BMI')
  - (X = ais[['Ht', 'Wt', 'LBM', 'BMI']])
- Each data is analyzed as a principal component, and up to the second principal component  
Plot the data
  - Development: Let's change the color for each gender (see page 14)
- Plot the contribution ratios of the principal components and determine up to which principal component is needed to reach a cumulative contribution ratio of 90%.

## 演習2

- 次のデータを正規化しなさい（平均0, 分散1）
  - wine\_data.csv（qualityは除く）
    - 注）「;」区切りになっている
- 各データを主成分分析し，第二主成分まででデータをプロットしなさい
- 主成分の寄与率をプロットし、第何主成分までとれば、計90%の寄与率になるか調べよ

# 演習2

- Normalize the following data (mean 0, variance 1)
  - wine\_data.csv (excluding quality)
    - Note) It is separated by 「;」
- Each data is analyzed as a principal component, and up to the second principal component. Plot the data
- Plot the contribution ratios of the principal components and determine up to which principal component is needed to reach a cumulative contribution ratio of 90%.

# 演習3

- 次のデータを正規化しなさい（平均0，分散1）
  - boston\_house.csv（MVは除く）
- 各データを主成分分析し，第二主成分まででデータをプロットしなさい
- 主成分の寄与率をプロットし，第何主成分までとれば，計80%の寄与率になるか調べよ

# 演習3

- Normalize the following data (mean 0, variance 1)
  - boston\_house.csv (excluding MV)
- Analyze each data as a principal component and plot the data up to the second principal component.
- Plot the contribution ratios of the principal components and determine up to which principal component is needed to reach a cumulative contribution ratio of 80%.

Advance

早く終わった人はやってみよう

# 演習4

- 次のデータを `pca` 関数を使わずに、共分散、固有値分解を行うことで主成分分析を行え
- `wine_data.csv` (qualityは除く)
  - 注) 「;」区切りになっている
- 各データを主成分分析し, 第二主成分まででデータをプロットしなさい

# 演習4

- Principal component analysis can be performed by performing covariance and eigenvalue decomposition of the following data without using the PCA function.
  - wine\_data.csv (excluding quality)
    - Note) It is separated by 「;」
- Each data is analyzed as a principal component, and up to the second principal component. Plot the data