NumPyと数学復習

前回の課題の例（複数のSQL)

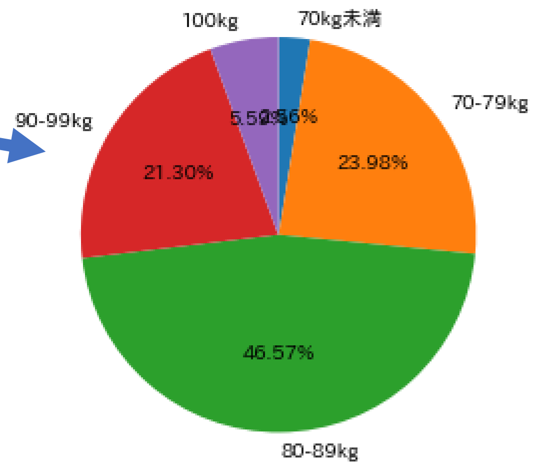
```
sql=["select count(*) from player where 体重<70",
"select count(*) from player where 体重>=70 and 体重<=79",
"select count(*) from player where 体重>=80 and 体重<=89",
"select count(*) from player where 体重>=90 and 体重<=99",
"select count(*) from player where 体重>=100"]
w=[]
for s in sql:
    for row in c.execute(s):
        s1=row[0]
        w.append(s1)
conn.close()
labels = [                          # グラフ要素のラベル
    '70kg未満', '70-79kg', '80-89kg', '90-99kg', '100kg'
]
plt.pie(x=w,                        # グラフ要素の値を設定
        labels=labels,              # グラフ要素のラベルを設定
        autopct='%.2f%%',           # 構成割合として小数点以下2桁までをプロット
        startangle=90,              # 90度（真上）の位置から開始
        counterclock=False          # 時計回りにする
       )
plt.axis('equal')                   # グラフを真円にする
plt.show()
```

```
["select count(*) from player where 体重<70",
"select count(*) from player where 体重>=70 and 体重<=79",
"select count(*) from player where 体重>=80 and 体重<=89",
"select count(*) from player where 体重>=90 and 体重<=99",
"select count(*) from player where 体重>=100"]
```

w
23

w
23
34

w
23
34
45



課題

- 身長から体重への円グラフに書き換えてください。

目次

- numpy復習
- 平均・標準偏差・分散
- 微分・積分・偏微分
- 行列
- 直線の式と空間方程式やベクトル
- 確率（ベイズの定理・二項分布・など）

スラッシング

- d=np.array([1,2,3,4,5,6,7]);
print(d[1:5])?

解答

0 1 2 3 4 5 6

([1 ,2, 3, 4, 5, 6, 7)

1から 5-1=4

始点　終点　間隔

- d[0:5:2]
- d[::-1]

- d[::-1] 反対に表示される

- d[0:5:2]

0 1 2 3 4 5 6

([**1 ,2, 3, 4, 5,** 6, 7)

ndarray(多次元行列)

a=np.array([[1,2,3],[4,5,6]])

(1)a.T

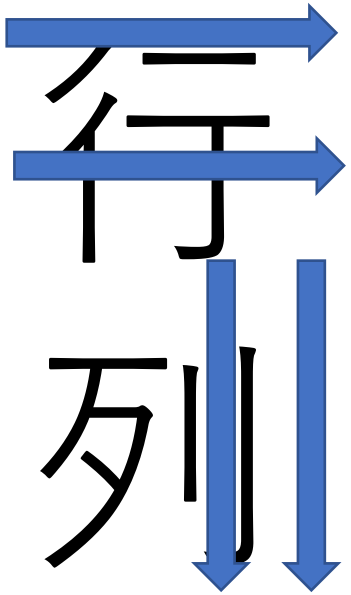
(2)a.shape

1	2	3
4	5	6

行→横

列→縦

どちらが縦か横かの覚え方



a.T　　転置

a.shape 型を表示

a.ndim　次元

```
>>> a.shape
(2, 3)
>>> a.T
array([[1, 4],
       [2, 5],
       [3, 6]])
>>> a.ndim
2
```

多次元のスライシング

問題
a=np.arange(10)で以下の出力する数字を書いてください
a[1:5]
a[2:8:2]
a[::-1]
a[:3]
a[4:]
a[:3],a[3:]
a[::2]
a[:]

解答

reshape

- 型を変える（行と列）

```
b=np.arange(20).reshape(4,5)
>>> b
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19]])
```

演習

- ４ × ５ の行列をすべて３の成分にしてください

```
>>> b=np.repeat(3,20);
>>> b.reshape(4,5)
array([[3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3],
       [3, 3, 3, 3, 3]])
```

スラッシング

- 問題

(1)b[1:3,2:4]

(2)b[:2,1:]

(3)b[::2,:]

(4)b[:,::2]

(5)b[:,::-1]

3 次元配列

c=np.zeros((3,4,5))

```
>>> c=np.zeros((3,4,5))
>>> c
```

5

4

```
array([[[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]],

       [[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]],

       [[0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]]])
```

3

問題

c[:,1:2,3:]=1は１になる部分は？

c[:,1:2,3:]=1

```
array([[[0., 0., 0., 0., 0.],
        [0., 0., 0., 1., 1.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]],

       [[0., 0., 0., 0., 0.],
        [0., 0., 0., 1., 1.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]],

       [[0., 0., 0., 0., 0.],
        [0., 0., 0., 1., 1.],
        [0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.]]])
```

:はすべて
*1:2*は1から２－１＝１まで
*3:*は３からすべて

Numpy数学関数

np.average

```
a=np.array([1,2,3,4,5,6,7,8])
print(np.average(a))
b=a.reshape(2,4)
print(b)
print(np.average(b,axis=0))
print(np.average(b,axis=1))
```

axis=1
[2.5 6.5]

axis=0
[3. 4. 5. 6.]

```
[[1 2 3 4]
 [5 6 7 8]]
```

平均だけ見ていていいのか（P学院般教の数学レポート）

- 毎年３倍なら４年で平均は？
- 距離９０ｋｍを行きは時速９０ｋｍ帰りは時速４５ｋｍ
平均の時速は
- 社長は年収1億、社員は年収５００万、３００万
会社の平均年収は

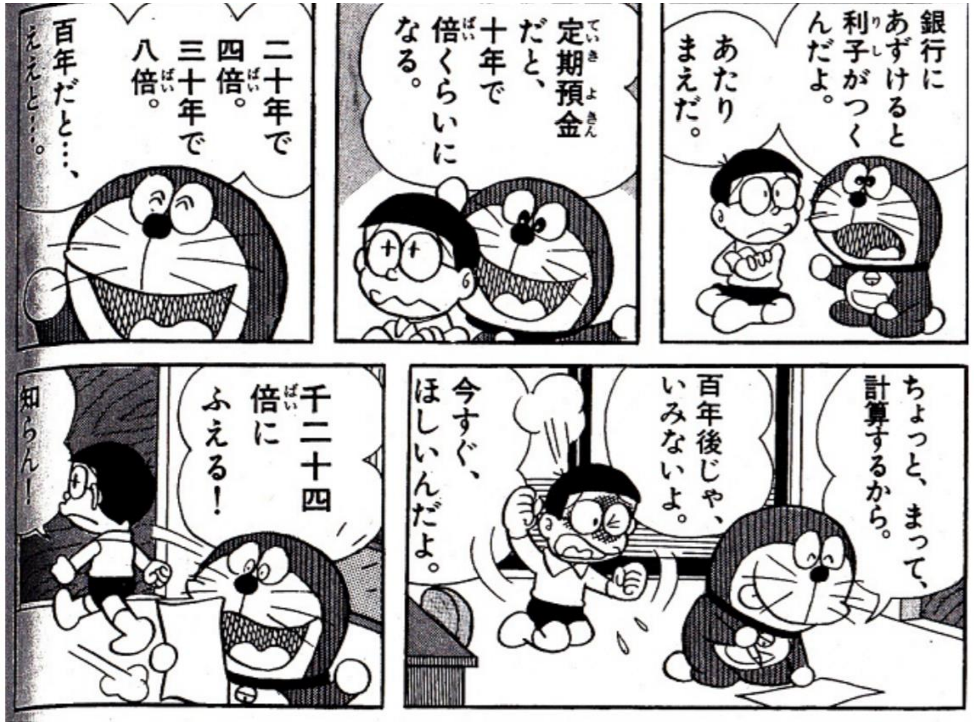
誤答例

（１）３倍

（２）９０＋４５＝１３５　１３５÷２＝６７．５

（３）(１億＋500万＋300万)÷３

課題 金利は？(100年後には1024倍)



金利が何％のときぐらいで1024倍？（risoku3.py）

単純条件として税金なしで金利はずっと一定で複利とします

pythonでシュミレートしてください。

ヒント　（risoku3.py）

```
principal=100000#元本といて計算
for year in range(0,100):#100年
    principal=principal * ( 1.0 + interest )
```

pythonは数式処理もできます

多項式の展開（中3）

```
>>> from sympy import *
>>> x = Symbol('x')
>>> y = Symbol('y')
>>> expr = (x + y)**2
>>> >>> expand(expr)
x**2 + 2*x*y + y**2
```

方程式を解く(中３)

```
>>> expr = x**2+4*x+4
>>> solve(expr, x)
[-2]
```

連立方程式を解く（中学２年）

```
>>> x, y = symbols('x y ')
>>> eq1=x + y-4
>>> eq2=2*x+3*y-6
>>> solve([eq1,eq2], [x,y])
{x: 6, y: -2}
```

式に値を代入する（中３）

```
>>> f = x**2 + 3*x + 2
>>> f1 = f.subs([(x, 1)])
>>> f1
6
```

微分（高３）

```
>>> x,y = symbols('x y’)
>>> f = x**2 + 2/x+sin(x)
>>> diff(f,x)
2*x + cos(x) - 2/x**2
```

積分（高３）

```
>>> expr = cos(x) * ln(y) + 2/y
>>> integrate(expr, x)
2*x/y + log(y)*sin(x)
```

例題

$y = x^2 + 4x$を微分したときｘ＝１のときの値を求めてみてください

解答

```
from sympy import *
x,y=symbols('x y')
f = x**2 + 4*x
f1=diff(f,x)
print(f1)
df = f1.subs([(x, 1)])
print(df)
```

標準偏差と分散と偏差値

$$標準偏差 = \sqrt{分散}$$

$$平均値：\bar{x} = \frac{1}{n}(x_1 + x_2 + \cdot \cdot \cdot \cdot x_n)$$

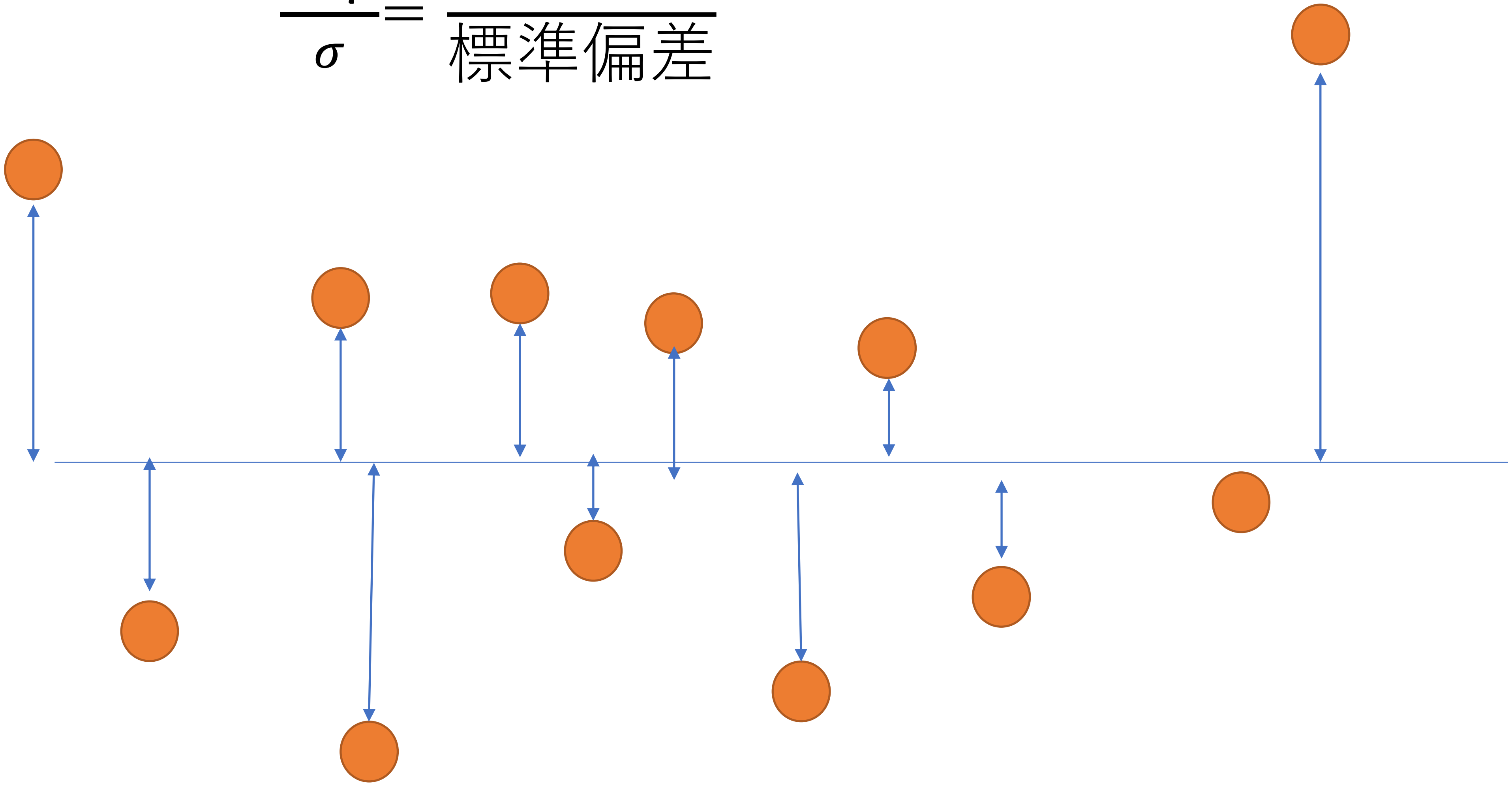
$$偏差 = x_i - \bar{x}$$

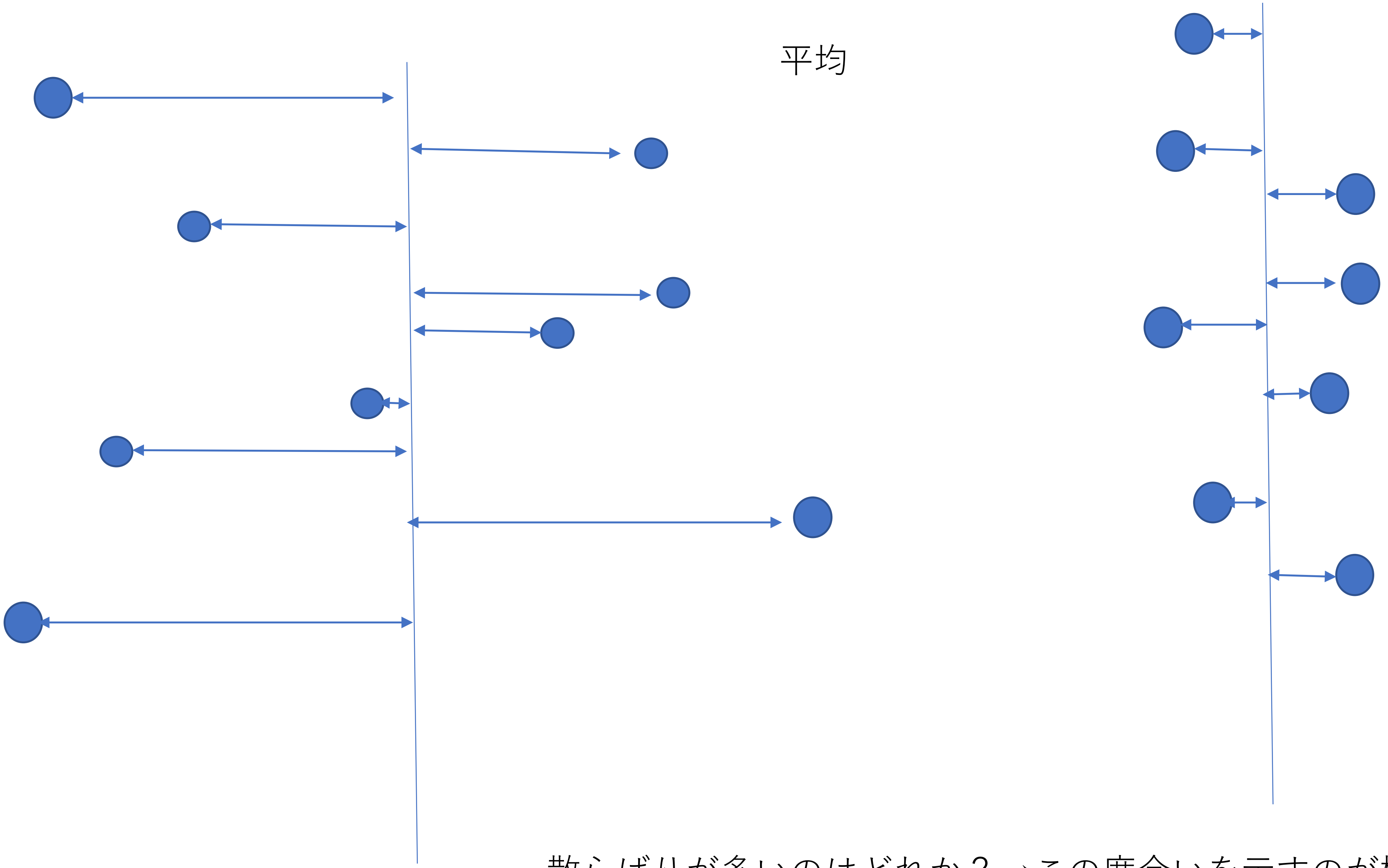
$$分散：\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$標準偏差：\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

散らばりを表す

$$\frac{x-\mu}{\sigma} = \frac{x-\text{平均}}{\text{標準偏差}}$$





散らばりが多いのはどれか？→この度合いを示すのが標準偏差

標準偏差・分散の公式

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2$$

$$\Downarrow$$

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} x_i^2 - \mu^2$$

標準偏差は分散の平方根$\sqrt{\sigma^2}$

標準偏差std,分散var

	国語	社会	数学	理科	英語
90-100	4	0	8	2	2
80-89	7	17	19	10	13
70-79	18	13	17	6	12
60-69	28	12	23	13	14
50-59	36	17	16	14	26
40-49	27	17	18	20	21
30-39	10	22	17	24	16
0-29	7	39	19	48	33

標準偏差std,分散var

標準偏差

```
>>> s=np.std([4,7,18,28,36,27,10,7])
>>> s
11.18523021667413
>>> s=np.var([4,7,18,28,36,27,10,7])
>>> s
125.109375
```

課題定義で求めてください

偏差値

- 平均が50、標準偏差が10の正規分布は偏差値を表す曲線
- 一般に言われているところは

「平均点だと偏差値が50」、「偏差値が70の学校はかなり難しい」

$$\frac{x-\mu}{\sigma}\times 10+50=\frac{x-\text{平均}}{\text{標準偏差}}\times 10+50$$

問題

mathscore=[8,19,17,23,16,18,17,19]

(順番に階級値は90-100,80-89,70-79,60-69,50-59,40-49,30-39,0-29)

でこの学校の標準偏差を求めよ（階級値は真ん中の数を点数とする。９０から１００までは95が８人とする）

また２９点の偏差値を求めなさい

90-100　→95点が８人

80-89,　→85点が19人

70-79,　→75点が17人

60-69,　→65点が23人

50-59,　→55点が16人

40-49,　→45点が18人

30-39,　→35点が17人

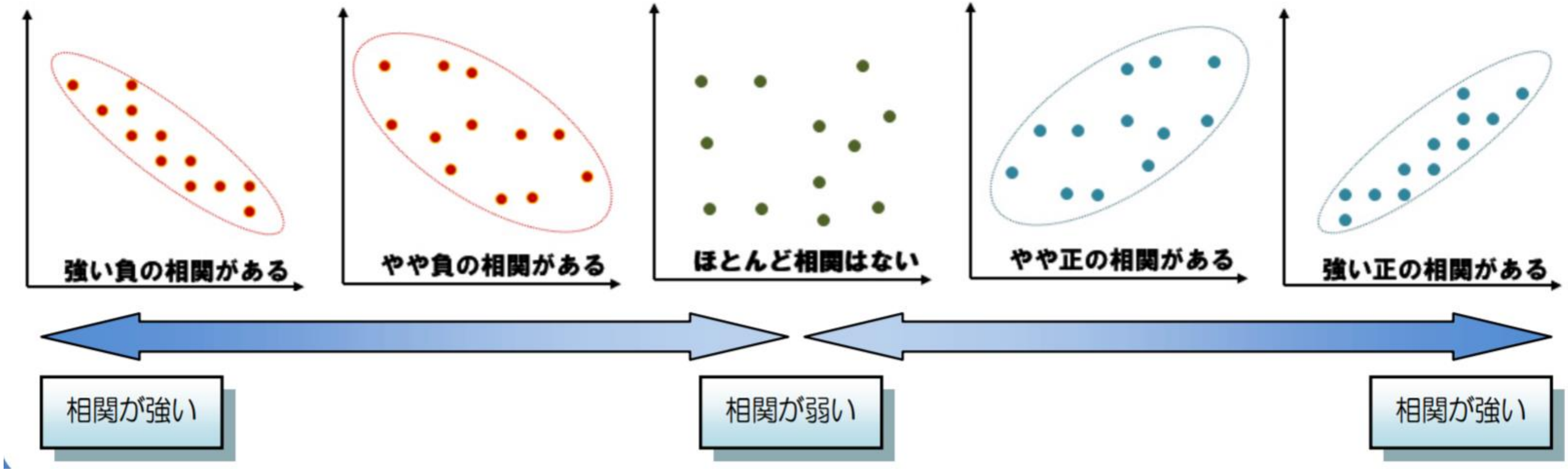
0-29　→15点が19人

相関係数

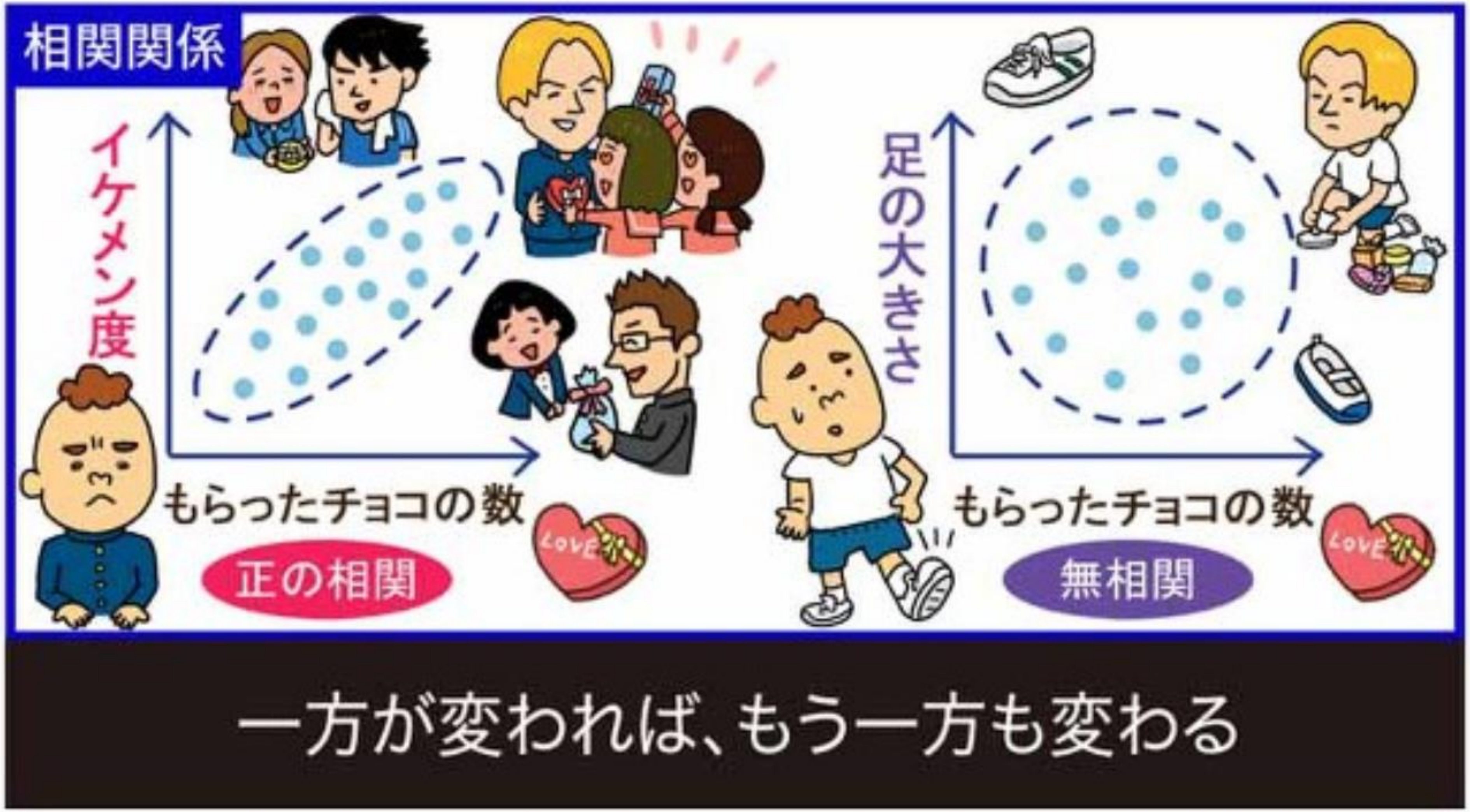
$$r = \frac{\displaystyle\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\displaystyle\sum_{j=1}^{n}(x_j - \overline{x})^2}\sqrt{\displaystyle\sum_{k=1}^{n}(y_k - \overline{y})^2}}$$

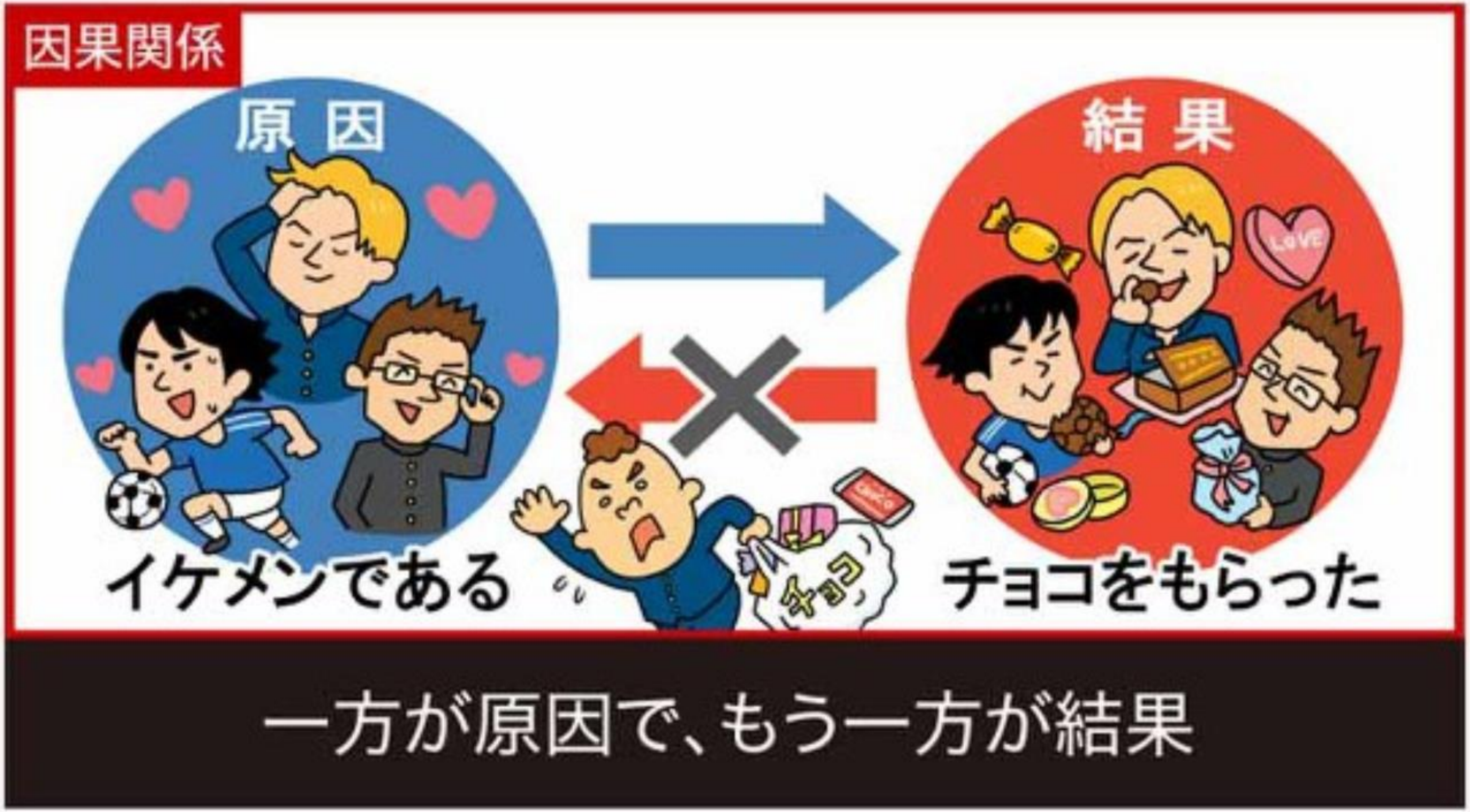
相関係数

```
#相関係数
import numpy
japanese = [5, 73, 29, 63, 68, 28, 45, 78, 70, 93]
math = [11, 82, 25, 61, 66, 27, 42, 88, 71, 84]
correlation = numpy.corrcoef(japanese, math)
print(correlation[0,1])
```


強い負の相関がある
やや負の相関がある
ほとんど相関はない
やや正の相関がある
強い正の相関がある
相関が強い
相関が弱い
相関が強い

相関と因果


相関関係
イケメン度
もらったチョコの数
正の相関
足の大きさ
もらったチョコの数
無相関
LOVE
LOVE
一方が変われば、もう一方も変わる


因果関係
原因
結果
イケメンである
チョコをもらった
一方が原因で、もう一方が結果

練習 11） 次のような２つの変量x, yからなるデータがある。これらについてxとyの間に相関があるかどうかを調べよ。また、相関がある場合には、正か負のどちらの相関であるかをいえ。

(1)

x	3.5	2.6	5.2	2.5	3.9	6.5	3.3	6.0	4.4	3.5
y	129	128	152	120	143	168	131	177	130	129

(2)

x	15	33	18	25	45	33	38	40	32	15
y	180	143	172	160	142	146	155	128	175	180

(3)

x	29	34	25	20	40	24	37	33	44	29
y	11	8	9	13	16	8	10	15	7	11

課題

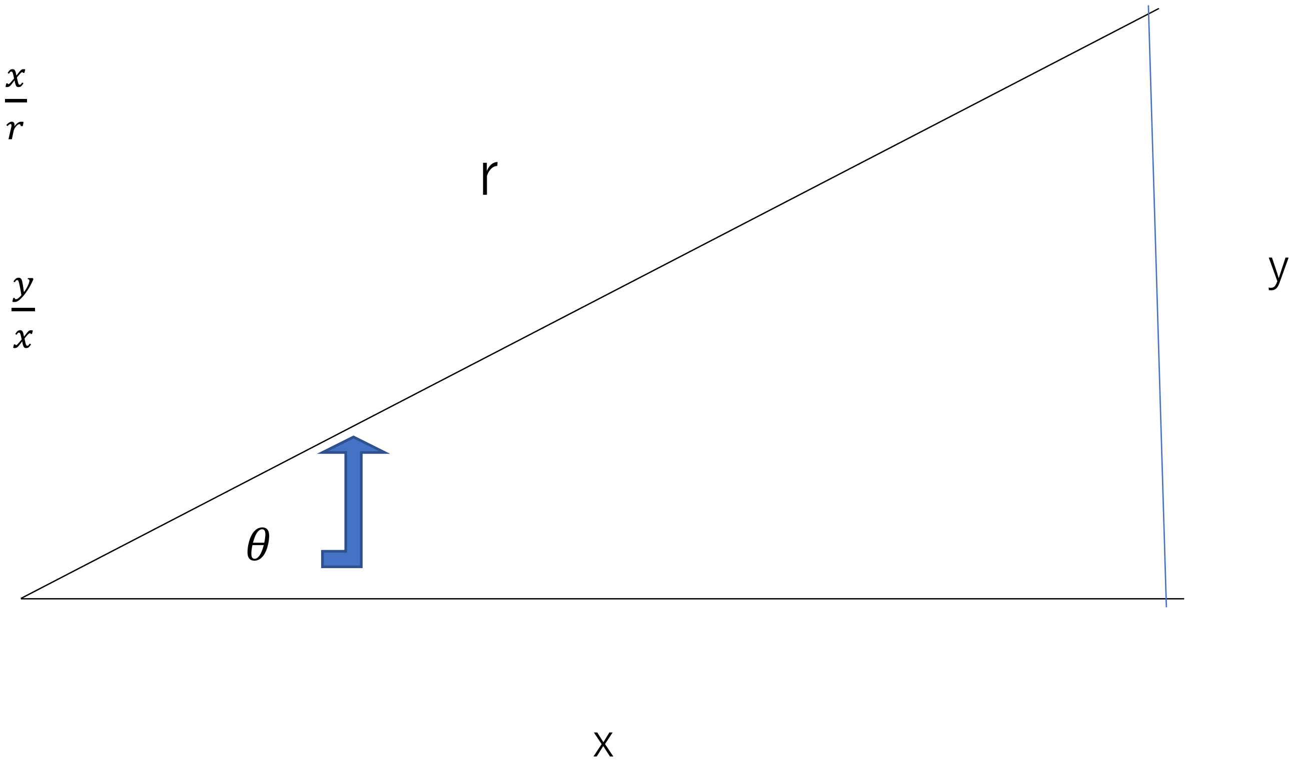
相関係数をnumpy.corrcoefを使わず定義式でコーディングしてみてください。（相関係数のpython.txt）

三角関数

$\sin\theta = \frac{x}{r}$

$\cos\theta = \frac{x}{r}$

$\tan\theta = \frac{y}{x}$



三角関数

弧度法

$\pi/180=1$度とする

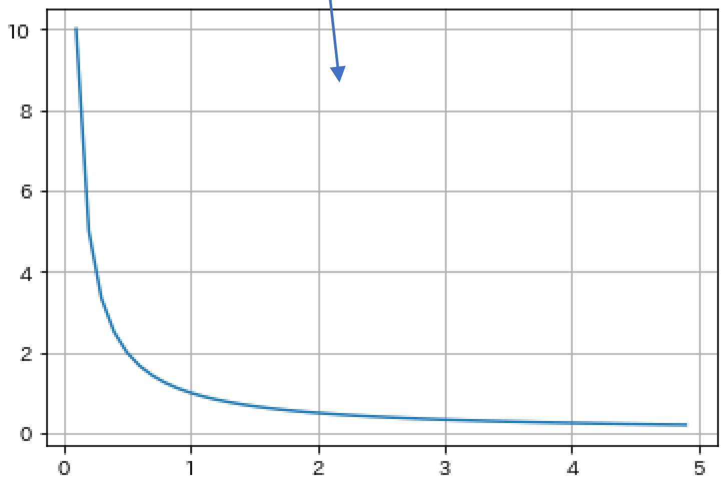
例

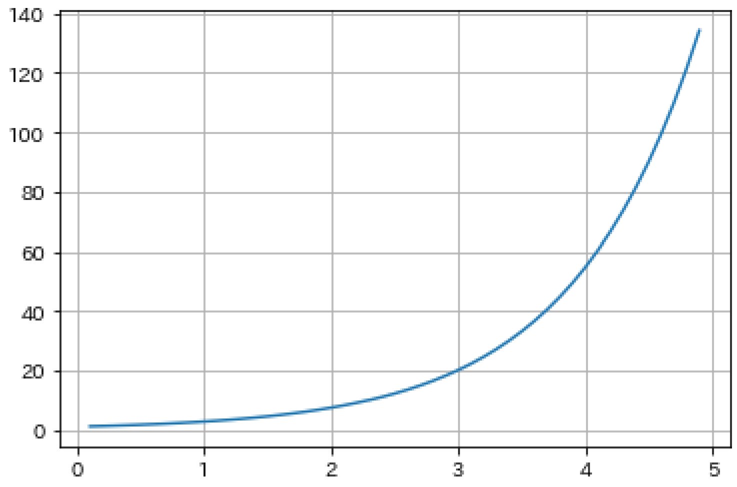
$\pi/180\times 30=\pi/3$

$\sin 60=\sin\pi/3$と表現します

指数関数(発散と収束をすると？)

$\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^{n} = e$(値は収束する)= 2.718281828459045235 36





課題

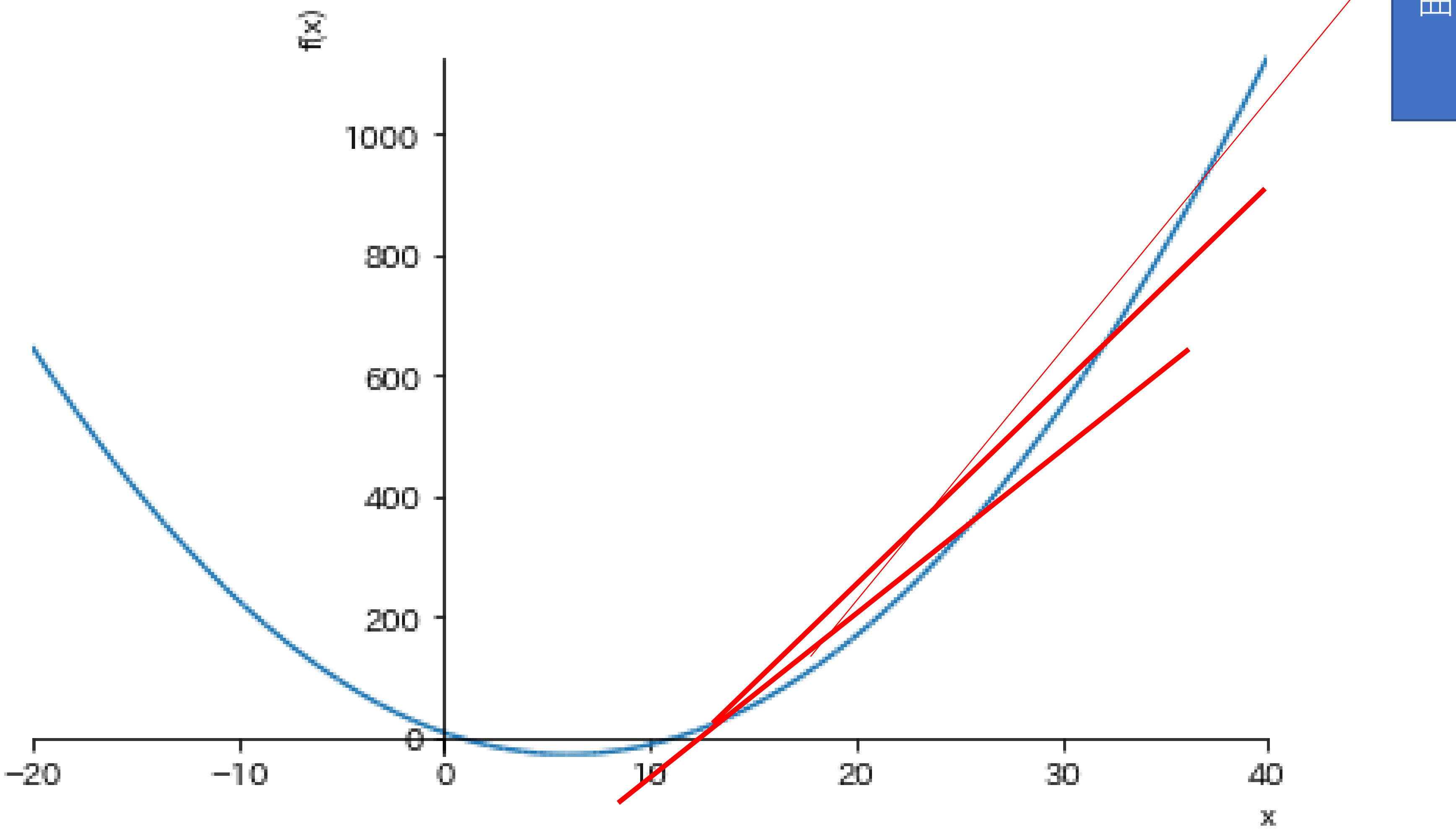
- eが収束するのpythonで確かめてください

```
const_neipia=2.718281828459045235360
def neipia(x,n):
    y=1+1/x
    y1=np.power(y, n)
    return y1
w=[]

for x in range(1,10000):
    y=neipia(x,x)
    error=const_neipia-y
    print(x,'*****',y,"error=",error)
    w.append(y)
plt.ylim(2.69,2.72)
plt.plot(w)
```

微分・積分・偏微分

- 微分の定義



曲線のその点での傾きを求めること

微分係数の求め方

$$f'(a) = \lim_{b \to a} \frac{f(b) - f(a)}{b - a}$$

または

$$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$$

微分係数の求め方　f’(x)= y ’ = dy/dx=$(x^2)'$

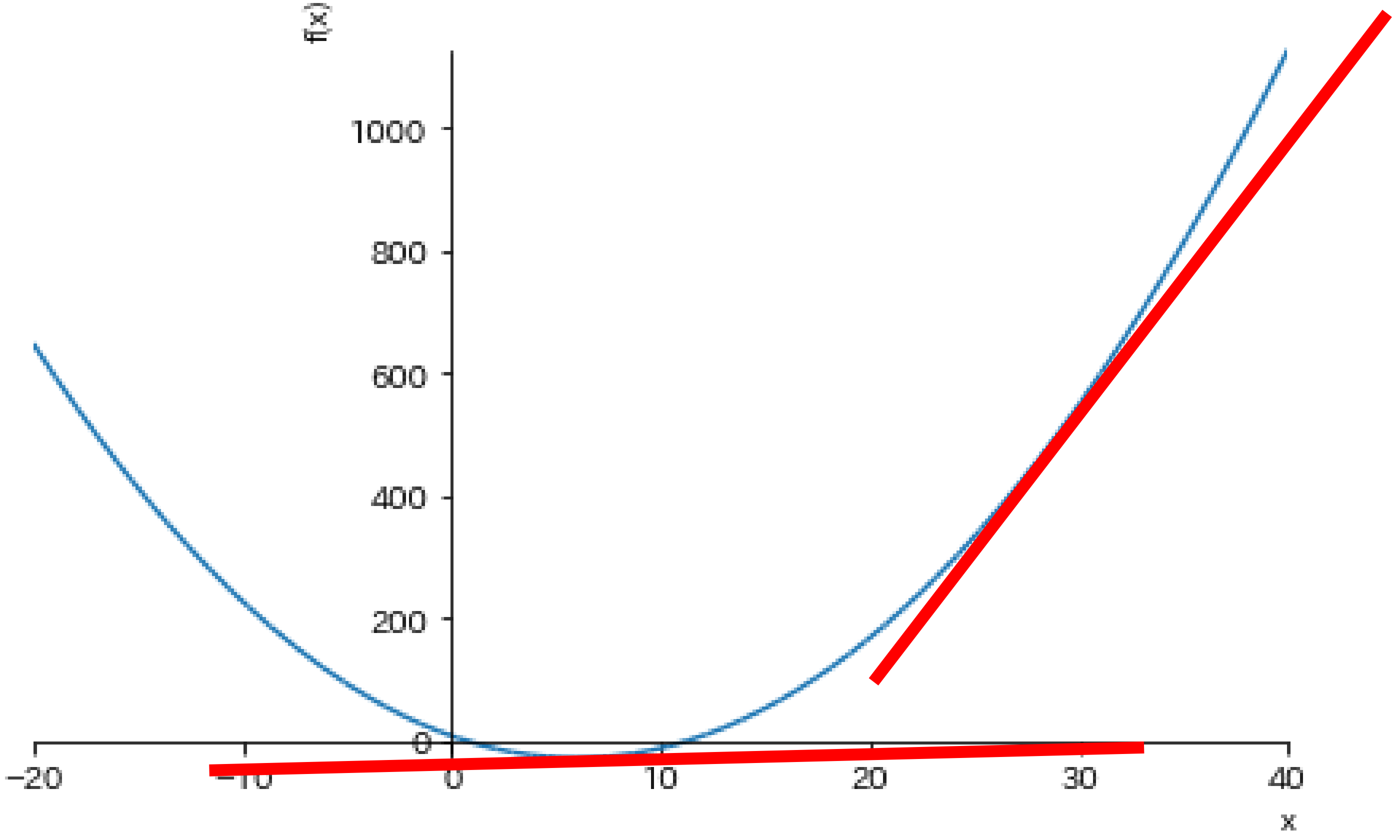
公式　$y' = nx^{n-1}$

$$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$$

ｙ ＝x^2の関数のｘ＝２ ｙ＝４の接線の方程式は？

傾きが０ということは？

- **傾き０を極値（最大値、最小値）**



つまり傾き０がｙの値の最大値または最小値→極値という

積分

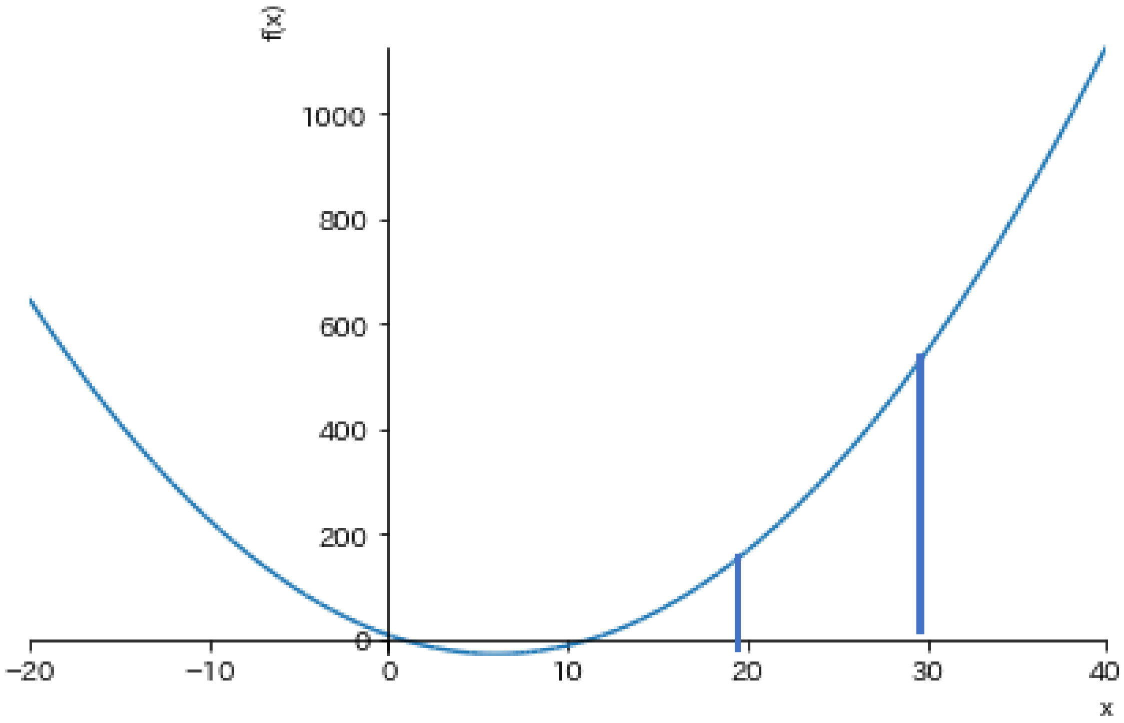
(1)

微分→積分

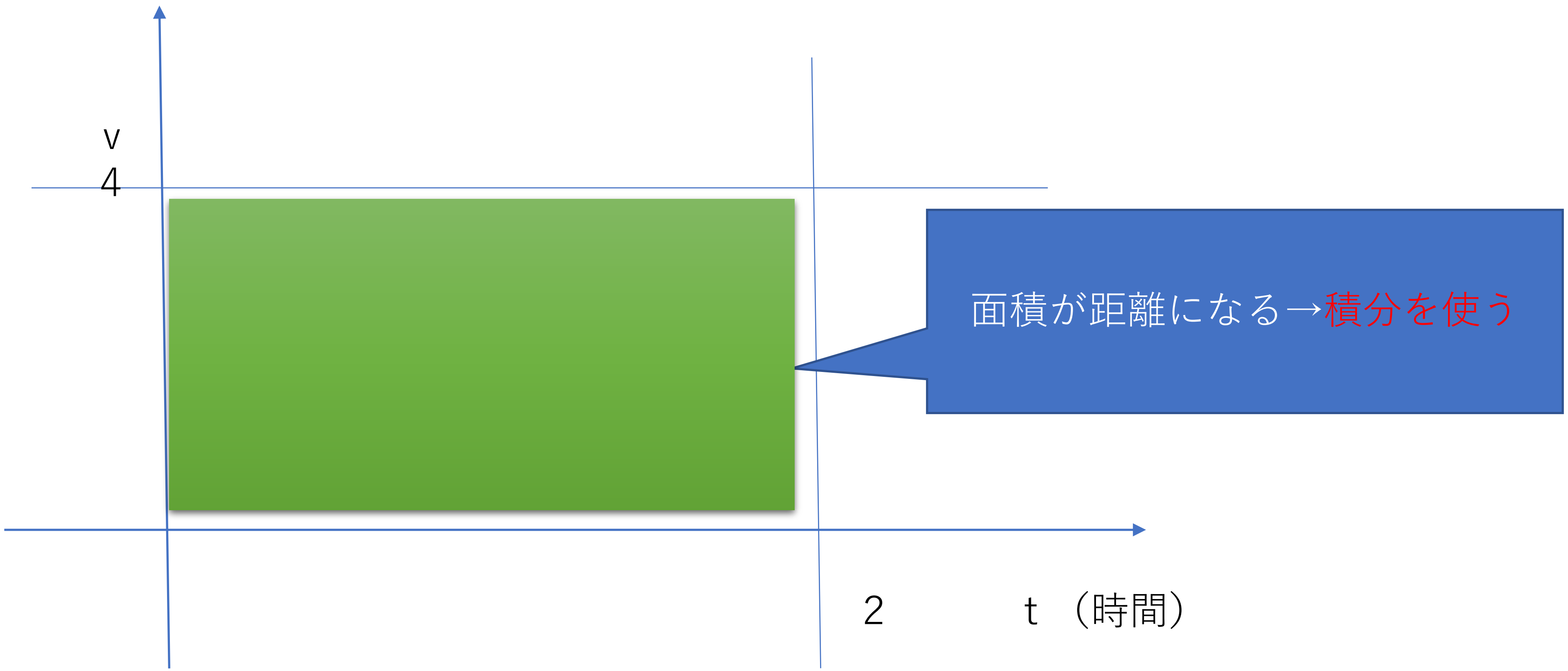
積分→微分　微分したものを元に戻すこと

$$\int x dx = \frac{x^{n+1}}{n+1} + c$$

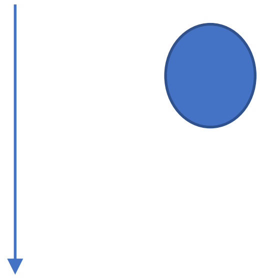
積分はx軸と囲まれた面積の関数といえる



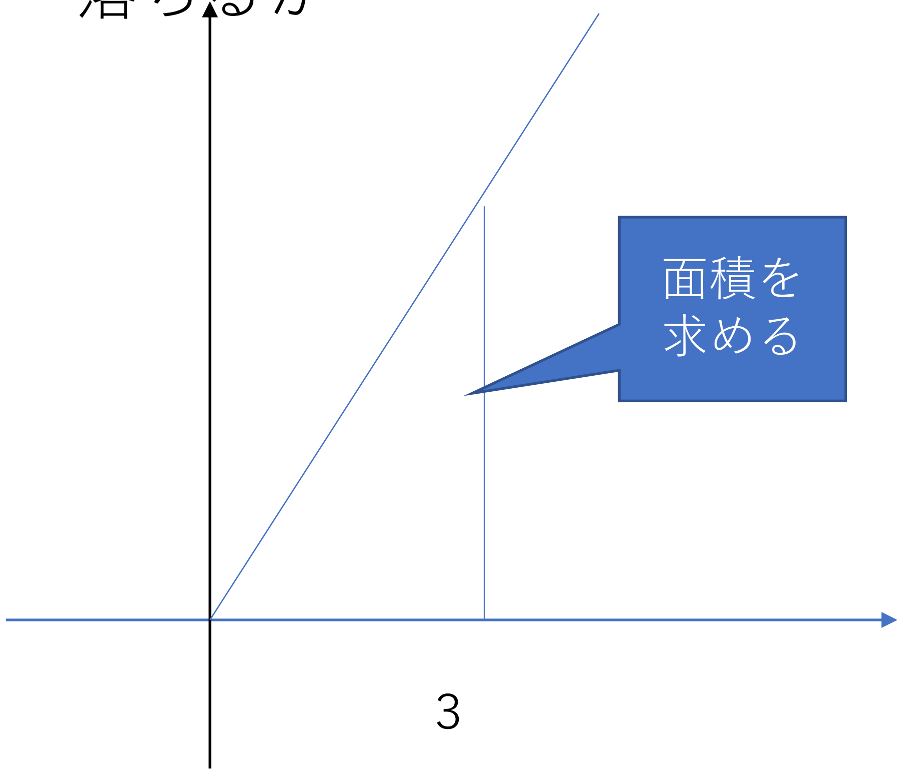
物理の世界では
s=vt(距離=速さ×時間)
時速４ｋｍで２時間歩いたときの距離



例　　物の自由落下



ｖ＝１０×ｔで表せる。３秒後にはどれだけ物は落ちるか



直線の式・平面の式

y=ax+bは直線

z=ax+by+cは？

点を多数集めたものが直線

直線をたくさん集めたものが平面

平面をたくさん集めたもの？

0次元　点　１次元　直線　　２次元　面

３次元　立体　４次元？

- y=ax+bと移行したcx+dy+e=0の違い

　　→直線は(c,d)と直線の傾きは９０度すなわち直交する

例

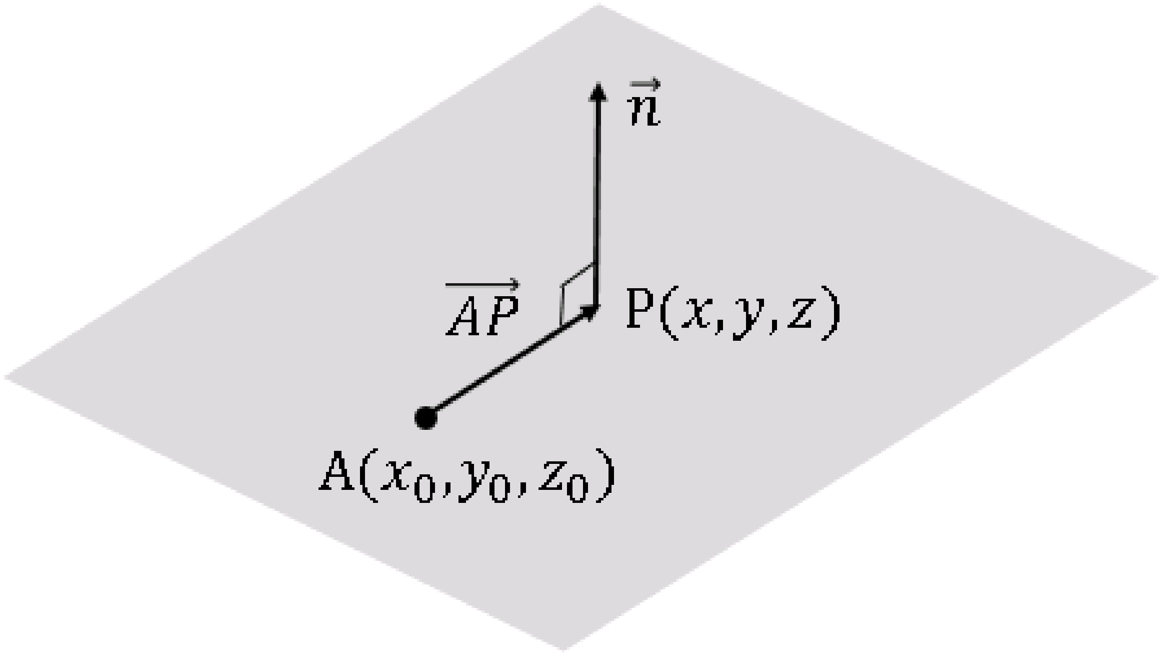
　y＝2x＋3と2x-y-3=0

　傾きは2　(2,-1)の傾きは-1/2

　2×-1/2=-1は直交している

平面では？

同じでax+by+c=0は（a,b,c)で平面とで直交



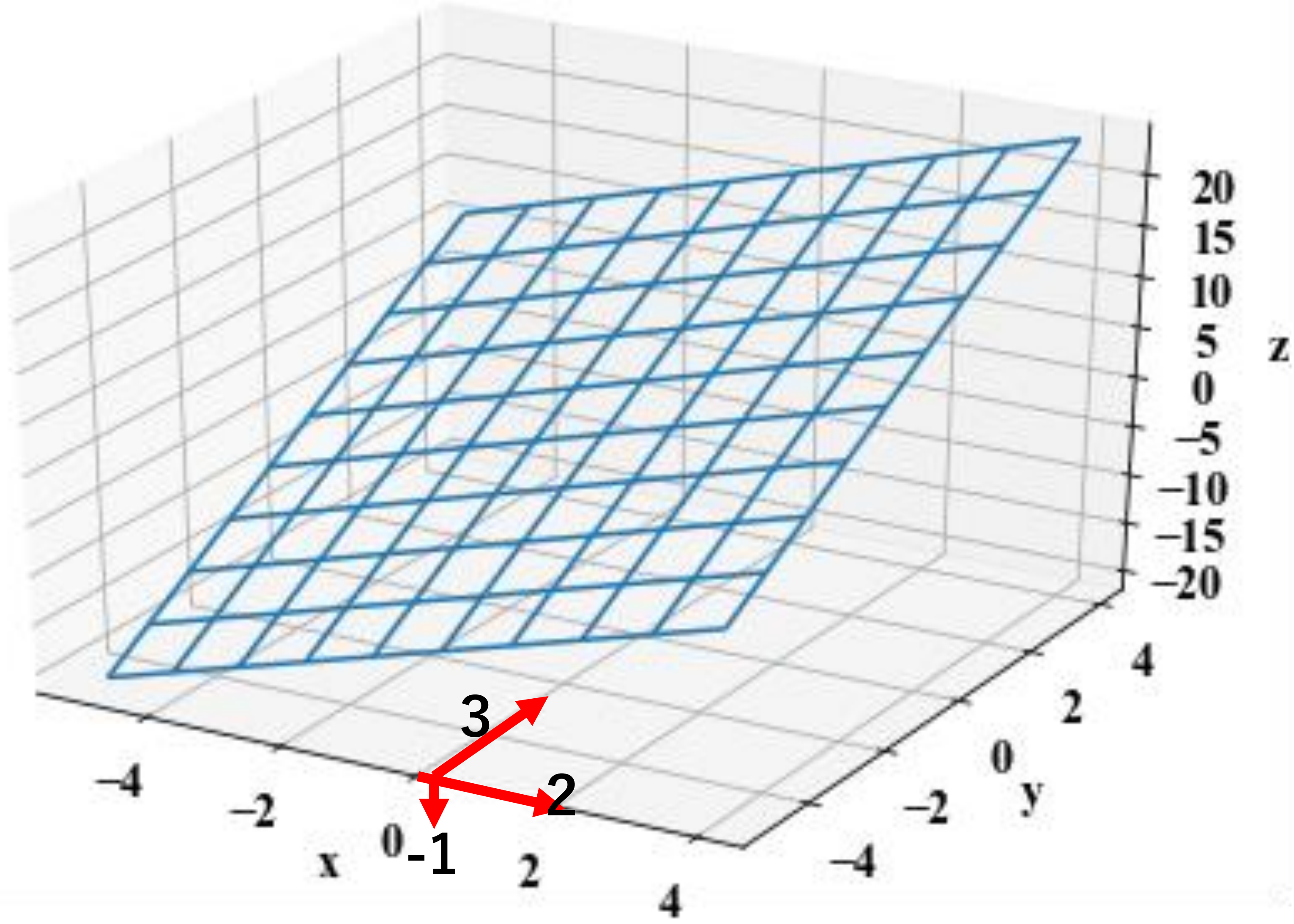
ベクトルでは内積が０が直交条件

課題

$z=2x+3y+4$のグラフを描いてください

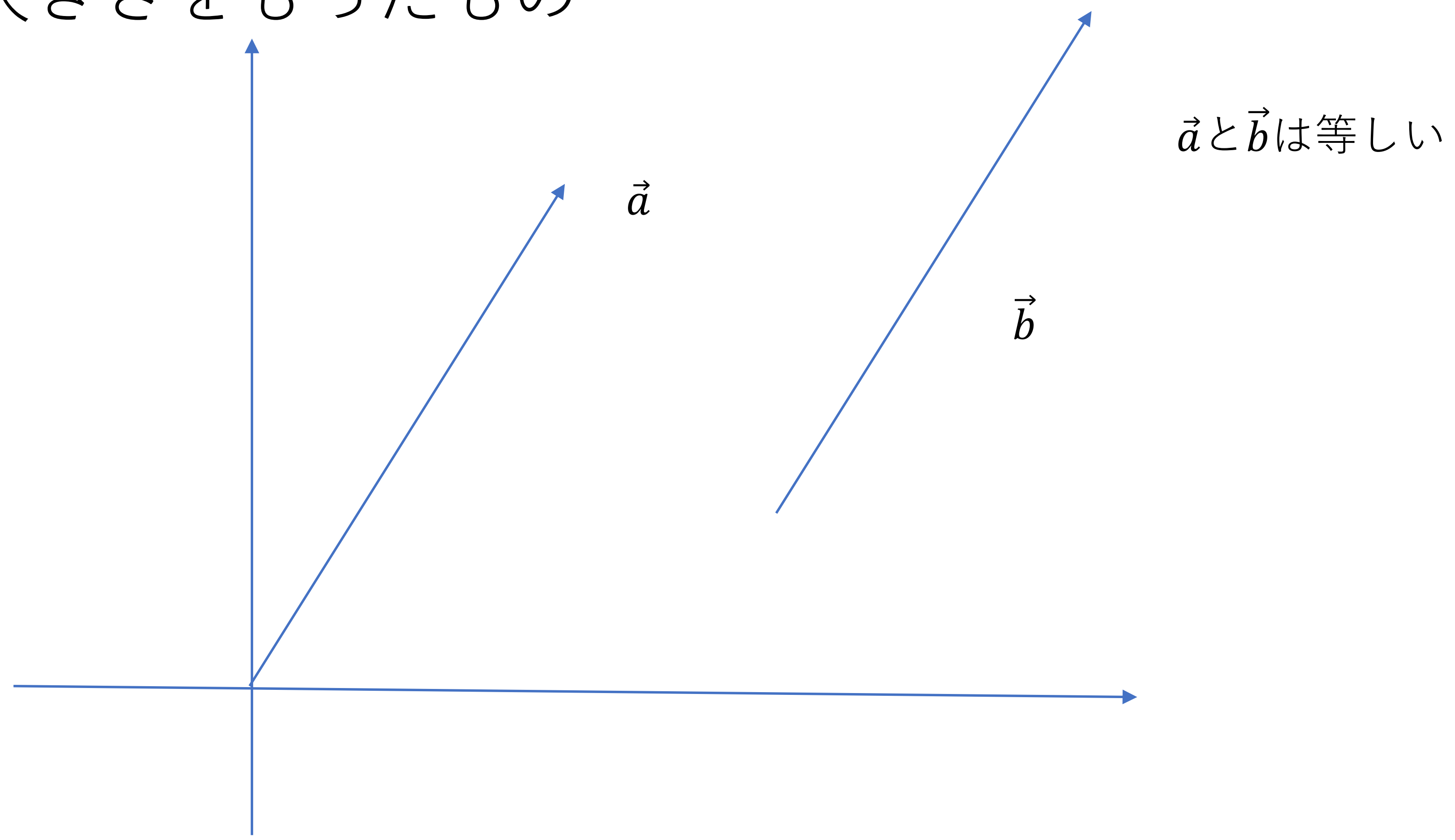
この平面の法線のベクトルを考えてください。

$(2x+3y+4-z=0$ 。法線ベクトル $(2,3,-1)$


20
15
10
5
0
−5
−10
−15
−20
z
−4
−2
0
2
4
x
−4
−2
0
2
4
y
3
2
-1

ベクトル

- 向きと大きさをもったもの



ベクトル　内積

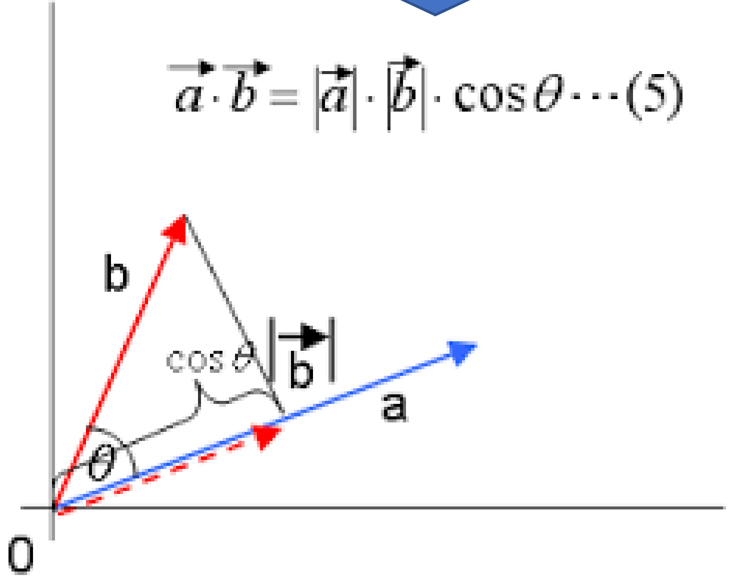
2つのベクトル $\overrightarrow{a} = (a_1, a_2)$, $\overrightarrow{b} = (b_1, b_2)$ に対して

$$\boxed{\overrightarrow{a} \cdot \overrightarrow{b} = a_1 b_1 + a_2 b_2}$$

$\overrightarrow{a}$ と $\overrightarrow{b}$ の内積

内積が０は２つのベクトルが垂直である

$$\overrightarrow{a} \cdot \overrightarrow{b} = |\overrightarrow{a}| \cdot |\overrightarrow{b}| \cdot \cos\theta \cdots (5)$$



行列の積

2列 = 2行

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax+cy \\ bx+dy \end{bmatrix}$$

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \times \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} = \begin{pmatrix} a_1b_1 + a_2b_3 & a_1b_2 + a_2b_4 \\ a_3b_1 + a_4b_3 & a_3b_2 + a_4b_4 \end{pmatrix}$$

【例】

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \times \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 \times 5 + 2 \times 7 & 1 \times 6 + 2 \times 8 \\ 3 \times 5 + 4 \times 7 & 3 \times 6 + 4 \times 8 \end{pmatrix}$$
$$= \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$$

```
>>> a = np.array([[1, 2], [3,4]])
>>> b = np.array([[5, 6], [7,8]])
>>> c=np.dot(a,b)
>>> c
array([[19, 22],
       [43, 50]])
>>>
```

行列の和

$$
和\quad \begin{array}{c} \text{行列A} \\ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \end{array} + \begin{array}{c} \text{行列B} \\ \begin{pmatrix} w & x \\ y & z \end{pmatrix} \end{array} = \begin{pmatrix} a+w & b+x \\ c+y & d+z \end{pmatrix}
$$

$$
差\quad \begin{array}{c} \text{行列A} \\ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \end{array} - \begin{array}{c} \text{行列B} \\ \begin{pmatrix} w & x \\ y & z \end{pmatrix} \end{array} = \begin{pmatrix} a-w & b-x \\ c-y & d-z \end{pmatrix}
$$

```
>>> c=a+b
>>> c
array([[ 6,  8],
       [10, 12]])
>>>
```

$$
\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} + \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}
$$

行列のルール

必ず同じ値

- (3,2)×(2,3)×(3,5)=(3,5)になる

単位行列（対角が１で他は０の行列）

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

どのような行列をEにかけても変わらない

逆行列

行列 A に対して逆行列 A^{-1} とは

$$A \cdot A^{-1} = A^{-1} \cdot A = E$$

逆行列を求めるnp.linalg.inv

x+y=5
2x+y=8

$$\begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 5 \\ 8 \end{pmatrix}$$

```
>>> a=np.array([[1,1],[2,1]])
>>> np.linalg.inv(a)
array([[-1.,  1.],
       [ 2., -1.]])
>>> inv=np.linalg.inv(a)
>>> b=np.array([[5],[8]])
>>> np.dot(inv,b)
array([[3.],
       [2.]])
>>>
```

$$A = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}$$

Aの逆行列= $\begin{pmatrix} -1 & 1 \\ 2 & -1 \end{pmatrix}$

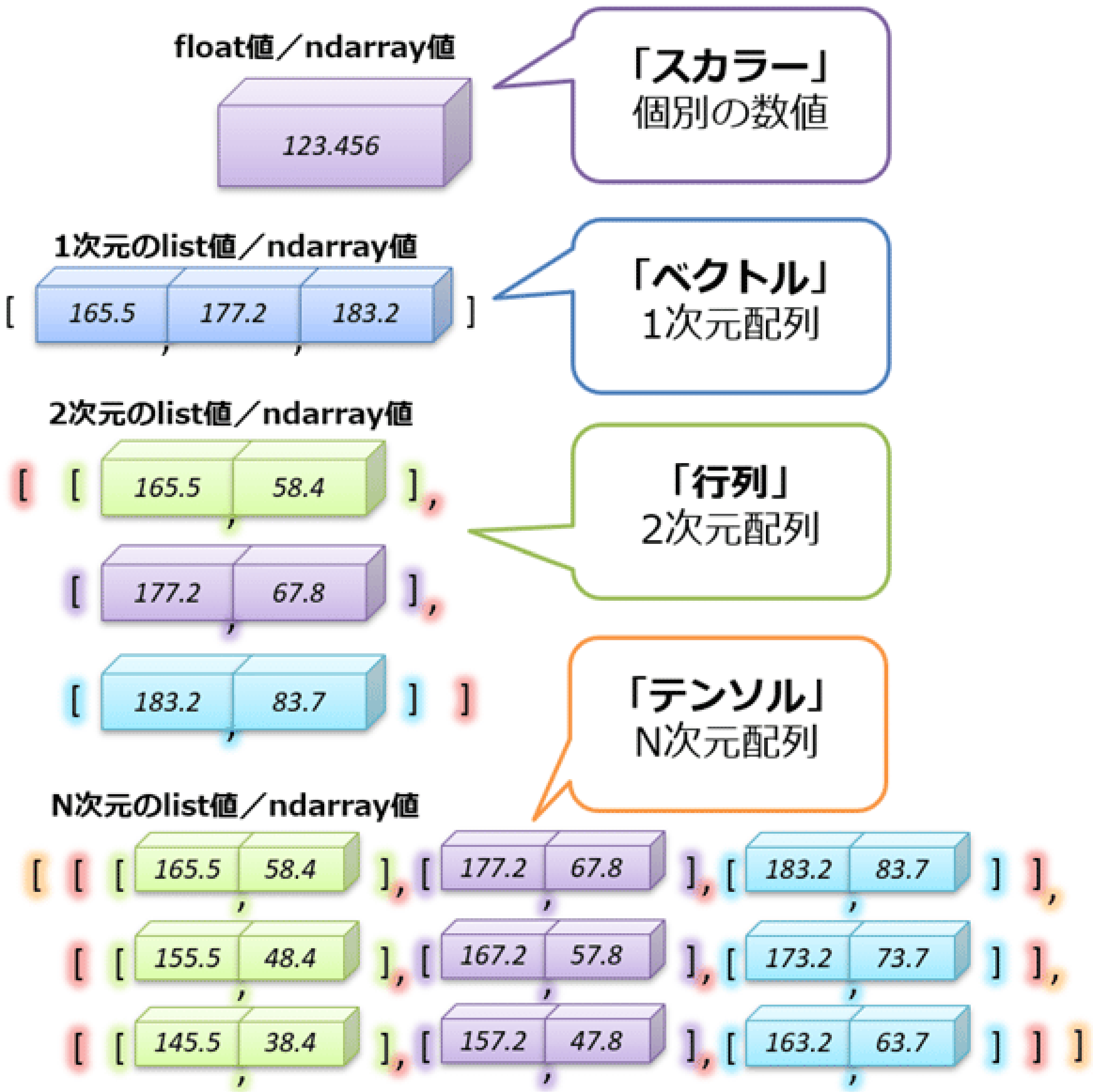
連立方程式を行列で表し解いてみてください

$$5x - 4y + 6z = 8$$
$$7x - 6y + 10z = 14$$
$$4x + 9y + 7z = 74$$

$$\begin{pmatrix} 5 & -4 & 6 \\ 7 & -6 & -10 \\ 4 & 9 & 7 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 8 \\ 14 \\ 74 \end{pmatrix}$$

解答

```
>>> a=np.array([[5,-4,6],[7,-
6,10],[4,9,7]])
>>> np.linalg.inv(a)
array([[ 1.29411765, -0.80392157,
0.03921569],
       [ 0.08823529, -0.10784314,
0.07843137],
       [-0.85294118,  0.59803922,
0.01960784]])
>>> inv=np.linalg.inv(a)
>>> b=np.array([[8],[14],[74]]
... )
>>> np.dot(inv,b)
array([[2.],
       [5.],
       [3.]])
```


float値／ndarray値
123.456
「スカラー」
個別の数値
1次元のlist値／ndarray値
[165.5, 177.2, 183.2]
「ベクトル」
1次元配列
2次元のlist値／ndarray値
[[165.5, 58.4],
[177.2, 67.8],
[183.2, 83.7]]
「行列」
2次元配列
「テンソル」
N次元配列
N次元のlist値／ndarray値
[[[165.5, 58.4], [177.2, 67.8], [183.2, 83.7]],
[[155.5, 48.4], [167.2, 57.8], [173.2, 73.7]],
[[145.5, 38.4], [157.2, 47.8], [163.2, 63.7]]]

ベイスの定理・順列・組み合わせ・重複順列

ベイズの定理

$P(A|B)P(B)=P(B|A)P(A)$

問題

病気にかかる率=0.01%としたとき

偽陰性

	陽性	陰性
病人	98%	2%
健康	20%	80%

偽陽性

病気にかかっている人に検査すると９８％で正しく診断される
健康な人に検査すると２０％で陽性と診断される

病気で陽性である確率

- 0.0001×0.98/(0.0001×0.98+0.9999×0.2)=0.0004・・

P(病気|陽) ×P(陽) =P (陽|病) ×P(病)

⬇

P(病気|陽) = P (陽|病) ×P(病) / P(陽)

98%陽性 ← コロナ 0.0001%

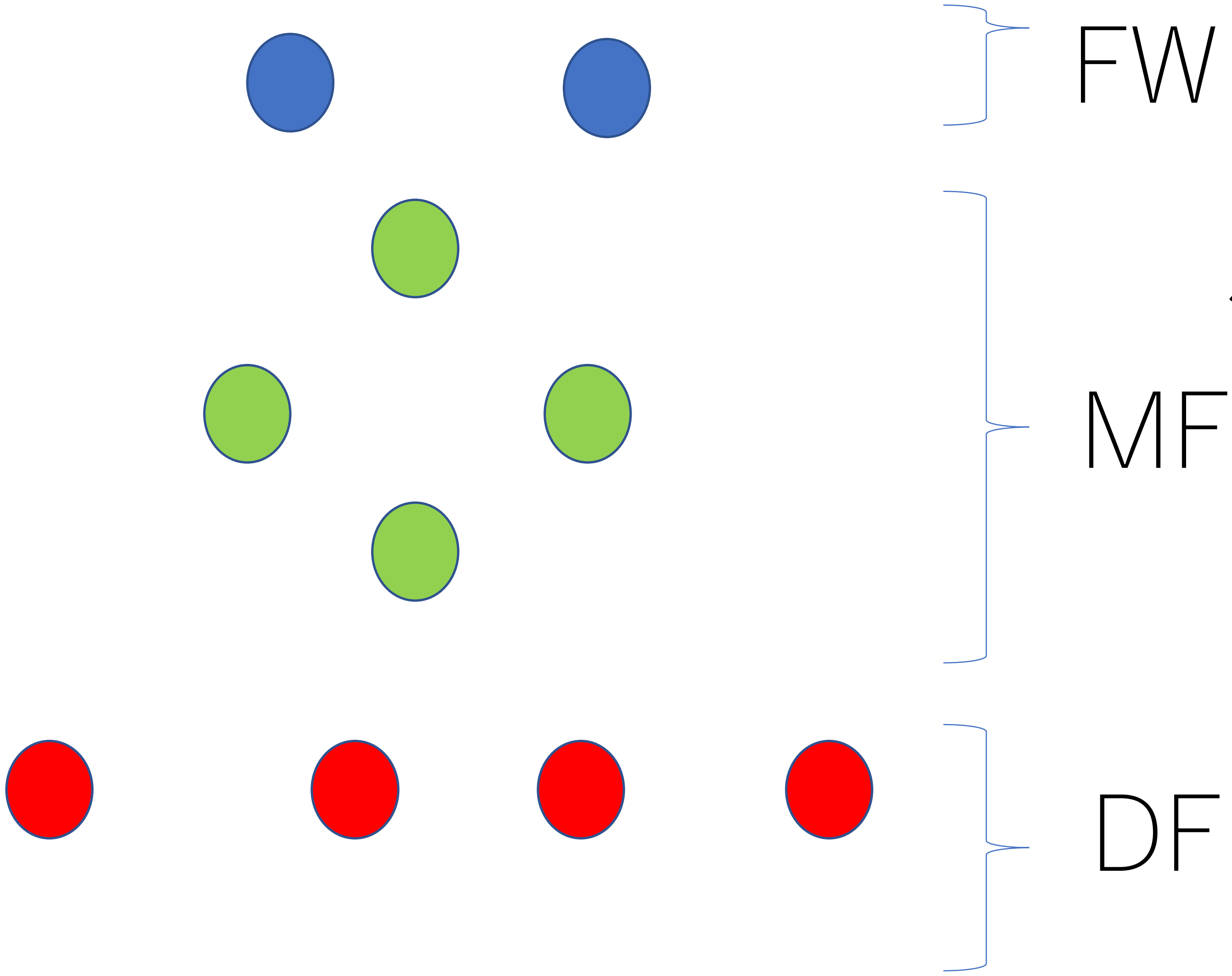
コロナでない 0.9999% → 20%陽性

組み合わせと順列

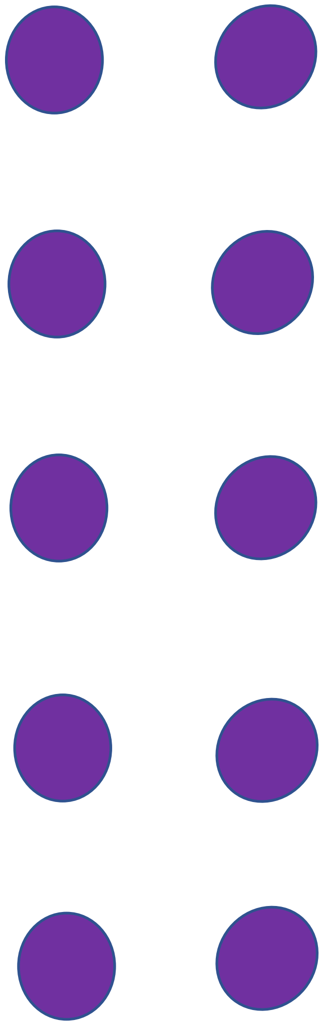
AKB４８の並び方→順列(Permintation)

BTSの中から３人選ぶ通り数→組み合わせ(Combination)

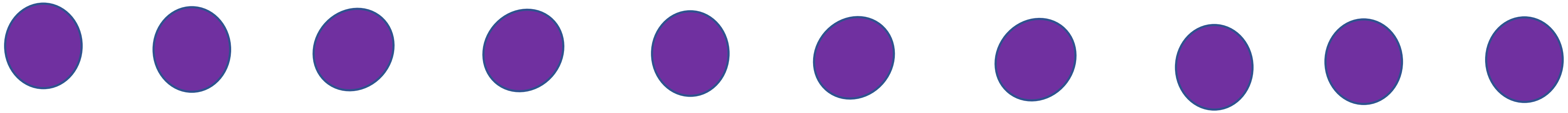
サッカー（２－４－４）



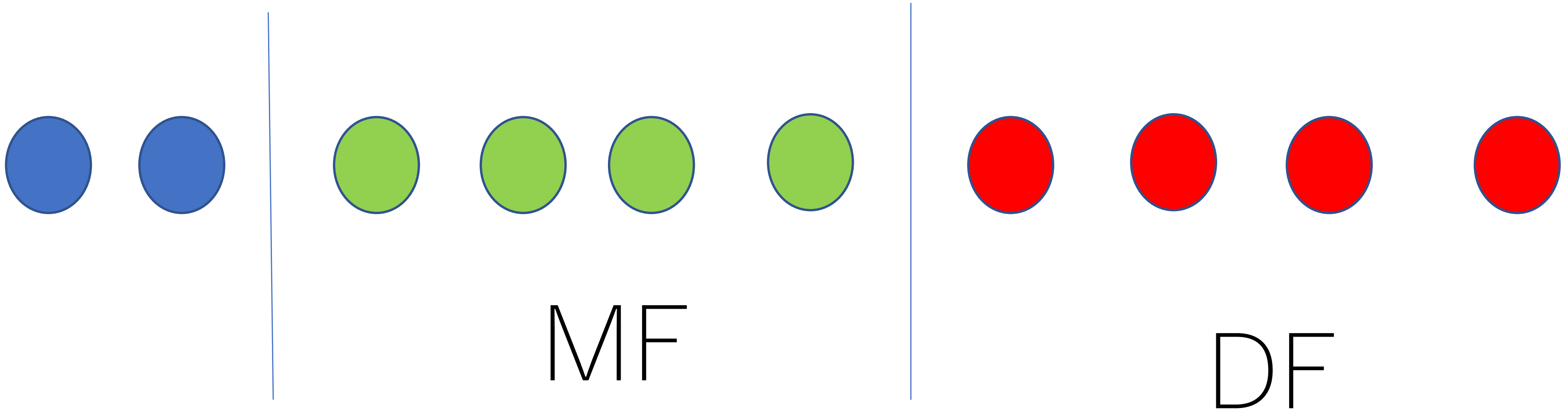
10人を配置するのは何通り？



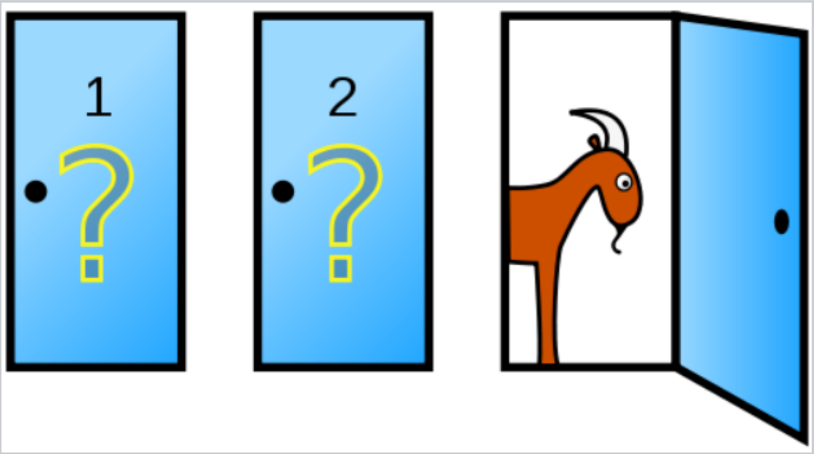
重複順列



$$\frac{10!}{2!\times 4!\times 4!}$$



モンティーホール問題



モンティ・ホール問題
閉まった3つのドアのうち、当たりは1つ。例示のように1つのドアが外れとわかった場合、直感的には残り2枚の当たりの確率はそれぞれ1/2になるように思える。