

2022年度全国高等学校情報教育研究会全国大会  
R2-6

# 表計算・プログラミング両面による 統計分析指導実践省察

佼成学園中学校・高等学校非常勤講師  
布村 寛



EXCEL



PYTHON



筆者サイト  
strnun mountain view

<http://strnun.fool.jp>



授業用  
資料

[http://strnun.fool.jp/pov-ray\\_strnun/Python\\_1.pdf](http://strnun.fool.jp/pov-ray_strnun/Python_1.pdf)



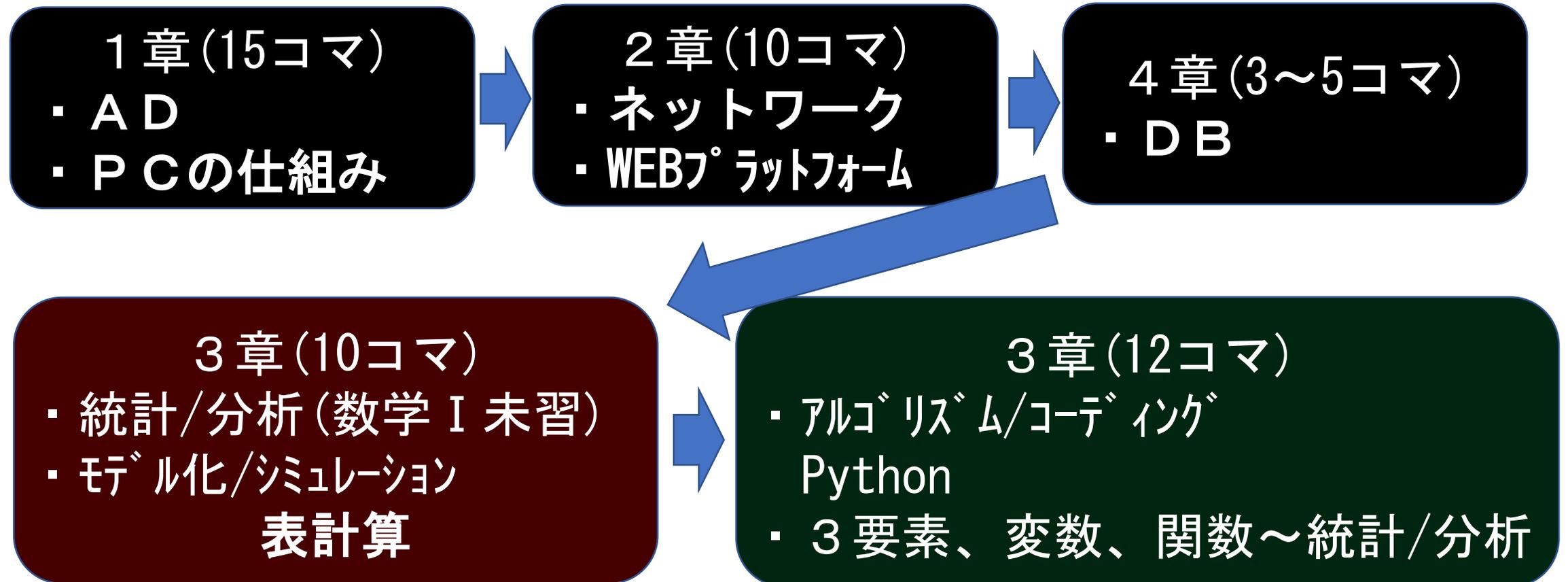
当資料

[https://strnun.fool.jp/pov-ray\\_strnun/2022zenkojyoken\\_statistics.pdf](https://strnun.fool.jp/pov-ray_strnun/2022zenkojyoken_statistics.pdf)

## 前任校昨年度までの科目配置(各2単位)と授業計画

1年必履「情報の科学」教科書：実教出版・・・276人

3年選択「社会と情報」教科書：実教出版・・・数人～20人



## 過去プログラミング言語指導と環境

- Ray tracing CG (Pov-Ray)
- Flash\_Action Script (Parafra)
- JavaScript (Text Editor)
- VBA (VB Editor)

- 機械学習への進展
- 大阪大会発表(統計授業実践)への助言
- 教員対象研修 Python主流

- 指導環境
- 教室リース仕様改変 → コスト問題
- Google Colab (Colaboratory) 選択

## 年間計画修正

- 表計算
- Python (Colab)  
両面で統計



# Python文法指導例(統計展開前)

分岐

```
n = input('what is your name?')
h = input('How tall are you in meters?')
w = input('How much is your weight in kilograms?')
bmi=float(w)/float(h)**2
if bmi>=25:
    print(n, 'さん、BMIは', bmi, 'over weight')
elif bmi <18.5:
    print(n, 'さん、BMIは', bmi, 'under weight')
else:
    print(n, 'さん、BMIは', bmi, 'healthy weight')
```

The Python logo, consisting of two interlocking snakes, one blue and one yellow, with the word "PYTHON" written in white capital letters across the middle.

# Python文法指導例（統計展開前段階） 乱数、おみくじ

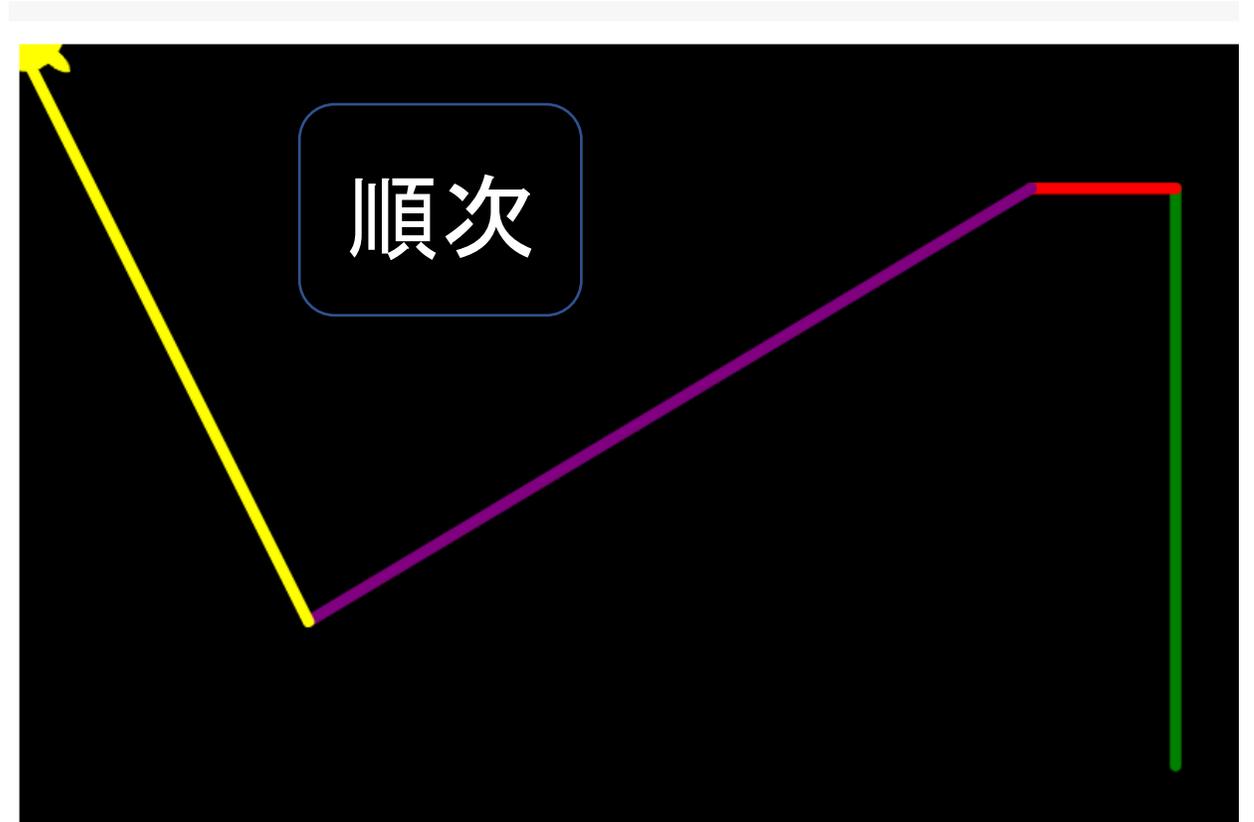
```
import random
fortune=['excellent', 'luck', 'good', 'unknown',
'bad luck', 'terrible']
print(random.choice(fortune))
```

# Python文法指導例（統計展開前段階） 図形描画、座標



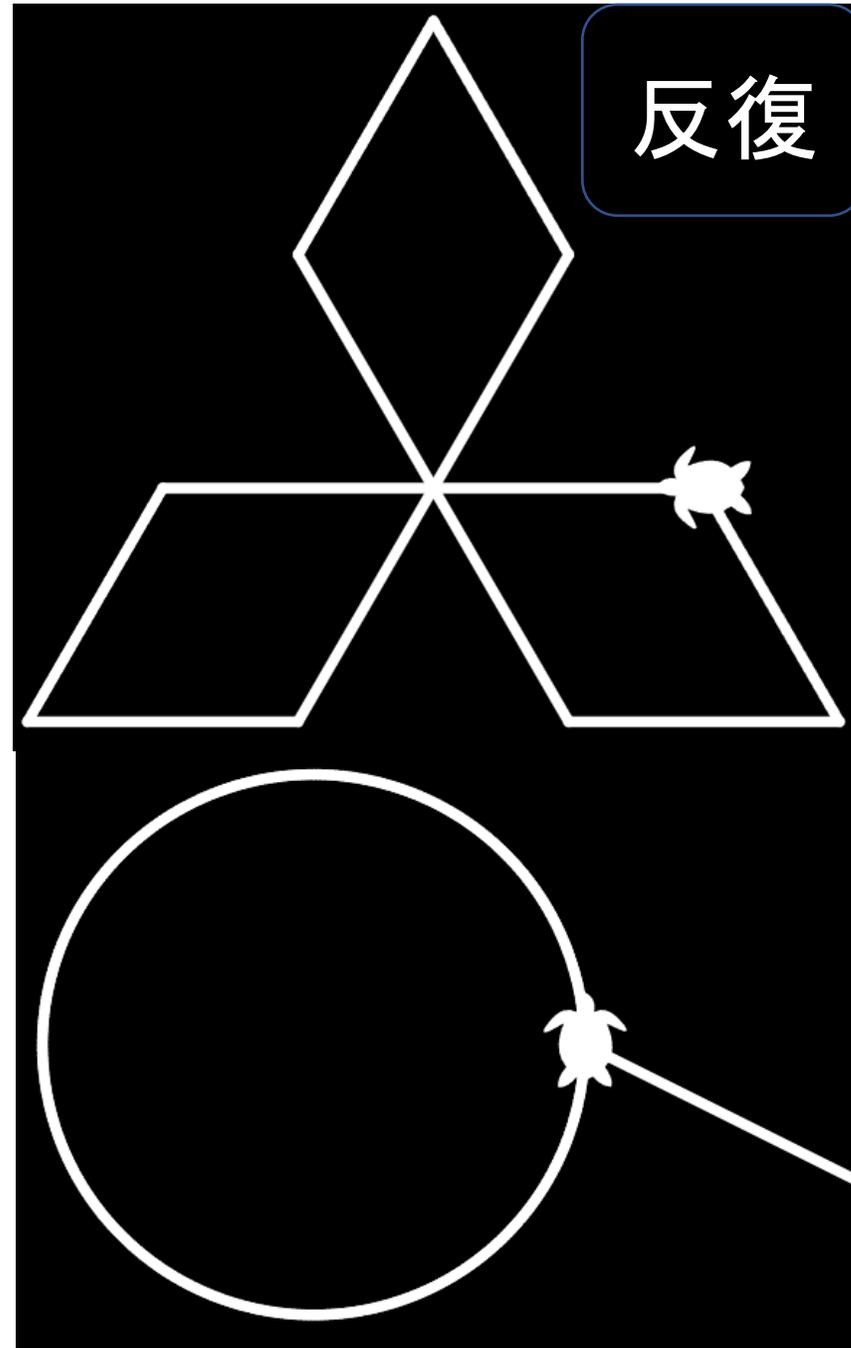
```
!pip3 install ColabTurtle
from ColabTurtle.Turtle import *

initializeTurtle()
pencolor('green')
forward(200)
pencolor('red')
left(90)
forward(50)
pencolor('purple')
goto(100,200)
pencolor('yellow')
goto(0,0)
```



```
initializeTurtle()  
left(90)  
for i in range(3):  
    forward(200)  
    left(60)  
    forward(100)  
    left(120)  
    forward(100)  
    left(60)
```

```
initializeTurtle()  
for a in range(360):  
    t = math.radians(a)  
    x = 100*(math.cos(t))  
  
    y = 100*(math.sin(t))  
  
    goto(x+200,y+200)
```

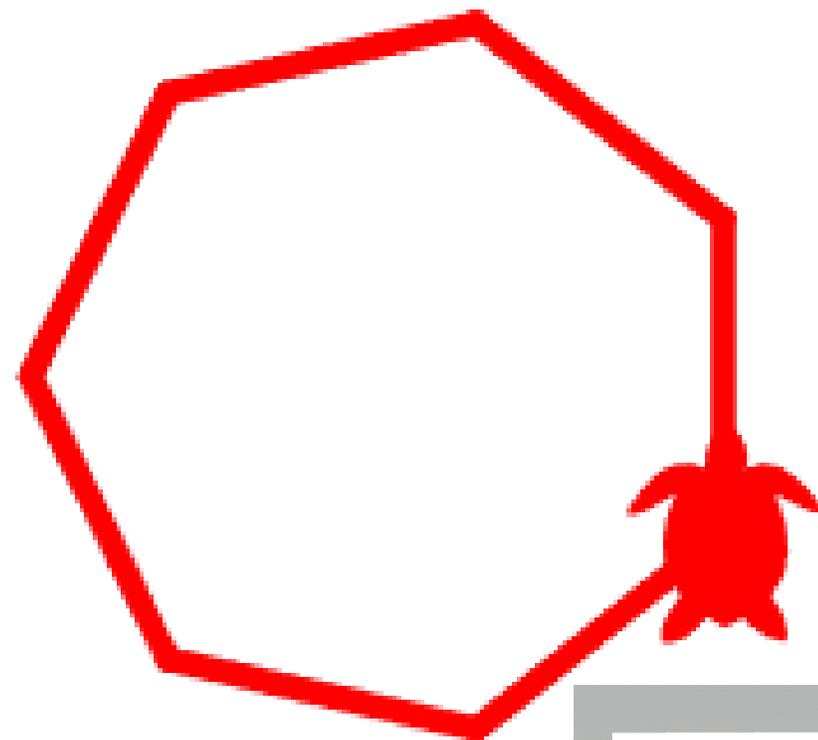


PYTHON

# 反復

冗長：HTMLとの類似性

```
initializeTurtle()  
bgcolor('white')  
color('red')  
n=7  
for i in range(n):  
    forward(50)  
    left(180 - (180 * (n-2) / n))
```



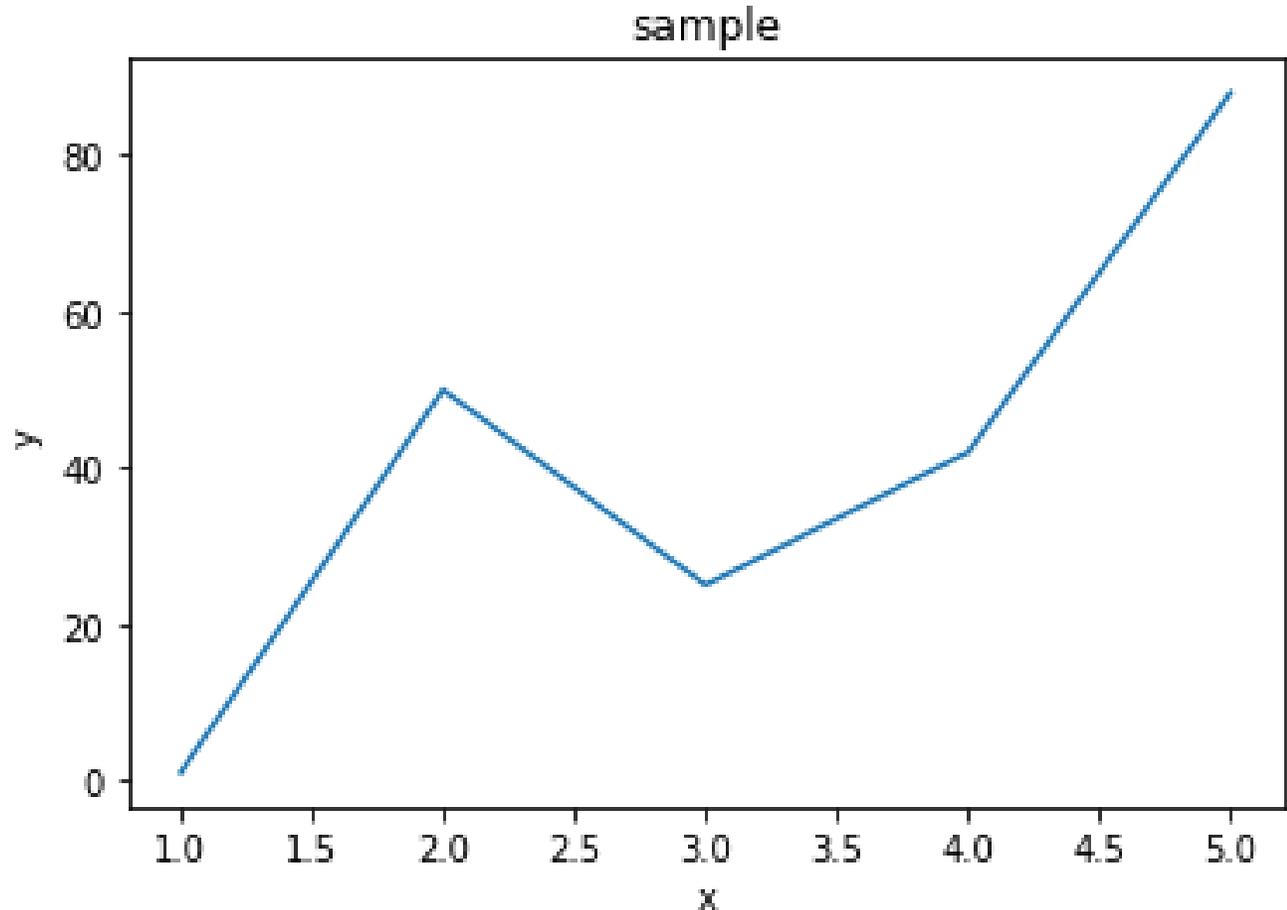
PYTHON



# Python文法指導例(統計展開前) グラフ描画

```
import matplotlib.pyplot as plt
y=[1, 50, 25, 42, 88]
x=[1, 2, 3, 4, 5]
plt.plot(x, y)
plt.title("sample")
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

ライブラリ活用の概念  
例：グラフ描画  
→表計算と同等のツール



## ガウス分布 1万個コイントス1万回実験

	A	B	C	D	E	F	G	H
1	コイン10,000個を投げて表なら1, 裏なら0として確立1/2の乱数をB列に発生させる。これを10,000回試行した結果をF列に記録し、ヒストグラムを作成する。右下のスイッチを押して若干時間経過後に処理が終了したらF2~F10001の範囲でヒストグラムを作成する。							
2	coin1	1			trial1	4988		
3	coin2	1			trial2	4997		
4	coin3	1			trial3	4989		
5	coin4	0			trial4	5037		

**10,000回  
試行スイッチ**

**EXCEL**

Col : 2

各セルの式やマクロを理解させやすい

coin1	1
coin2	1
coin3	0
coin4	1
coin5	0

coin9999	1
coin10000	1
sum	4972

**=RANDBETWEEN(0, 1)**

```
Sub Macro1 ()  
  Dim x As Integer  
  For x = 2 To 10001  
    Cells(x, 6) = Cells(10002, 2)  
  Next x  
End Sub
```

マクロ4行入力  
VBEで記述させると  
+15~20分  
仕込みでも十分

**=SUM(B2:B10001)**

Col : 6

trial1	4988
trial2	4997
trial3	4989
trial4	5037

Col : 6

trial9999	5087
trial10000	5054
max	5175
min	4795
med	5001
ave	5000
stdev	50

統計値算出式：既習

=MAX (F3:F10002)

=MIN (F3:F10002)

=MEDIAN (F3:F10002)

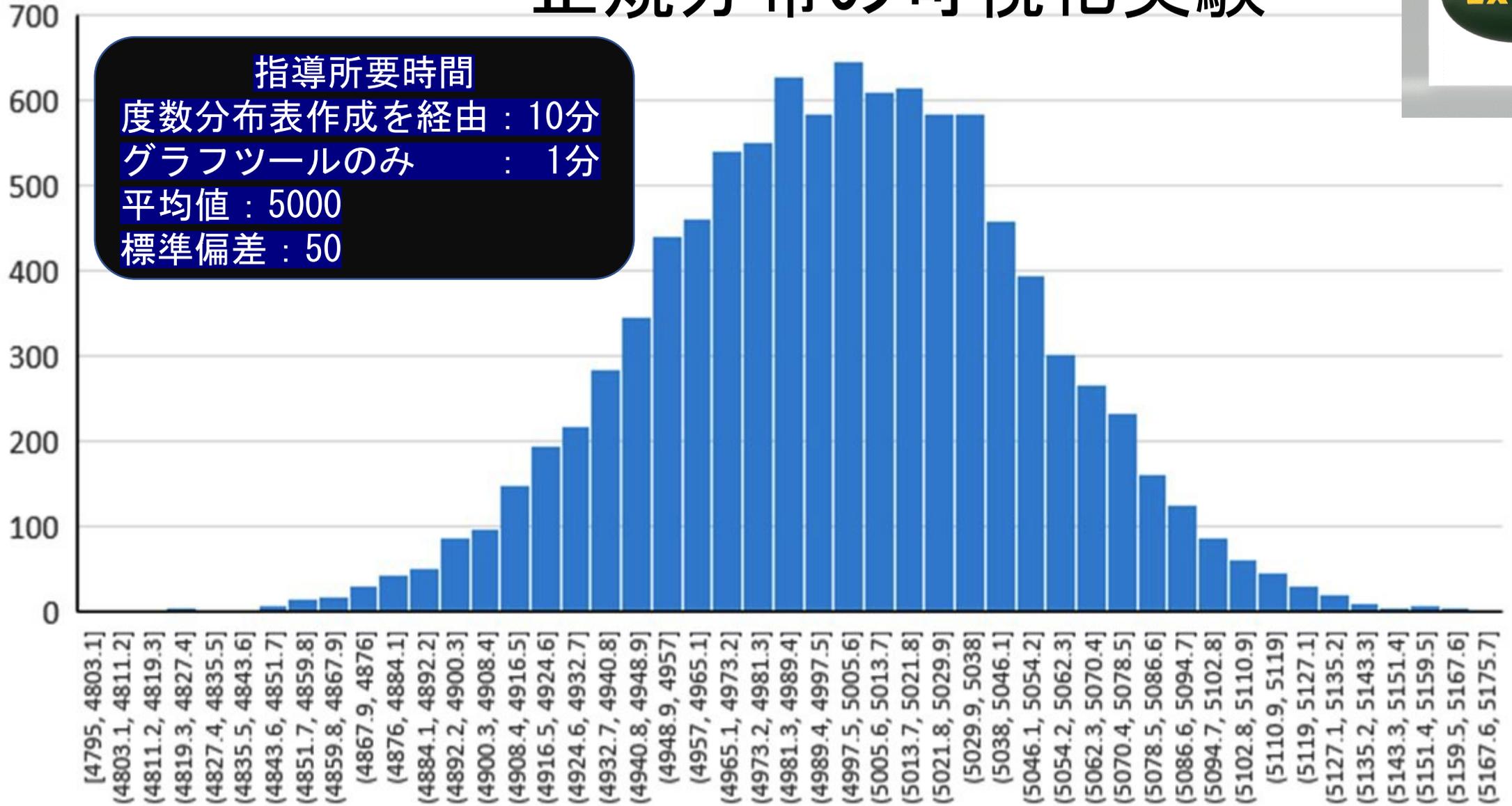
=AVERAGE (F3:F10002)

=STDEV. P (F3:F10002)

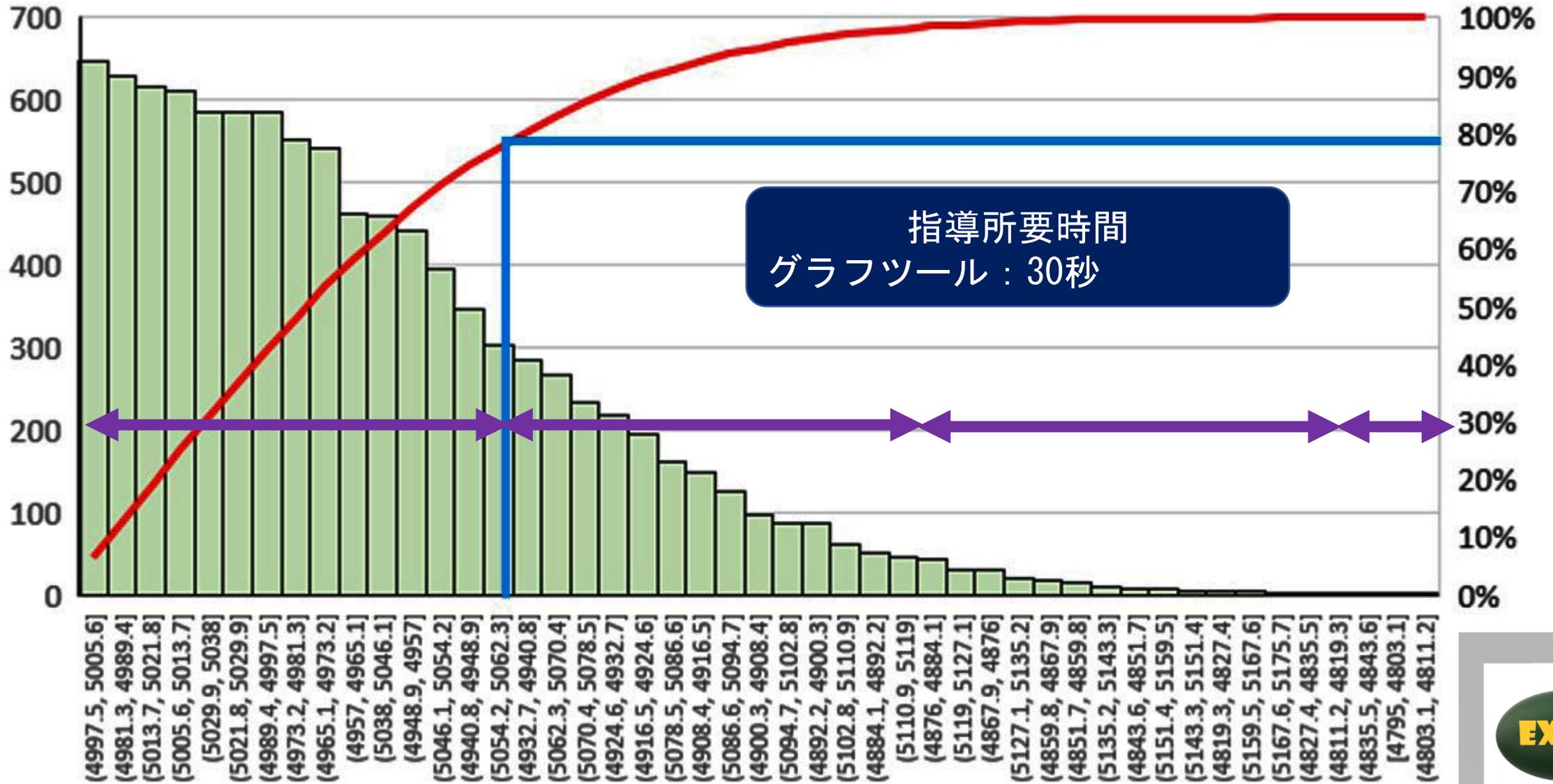
# 正規分布の可視化実験



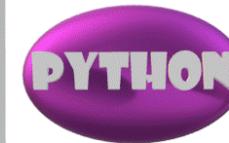
指導所要時間  
度数分布表作成を經由 : 10分  
グラフツールのみ : 1分  
平均値 : 5000  
標準偏差 : 50



# パレート



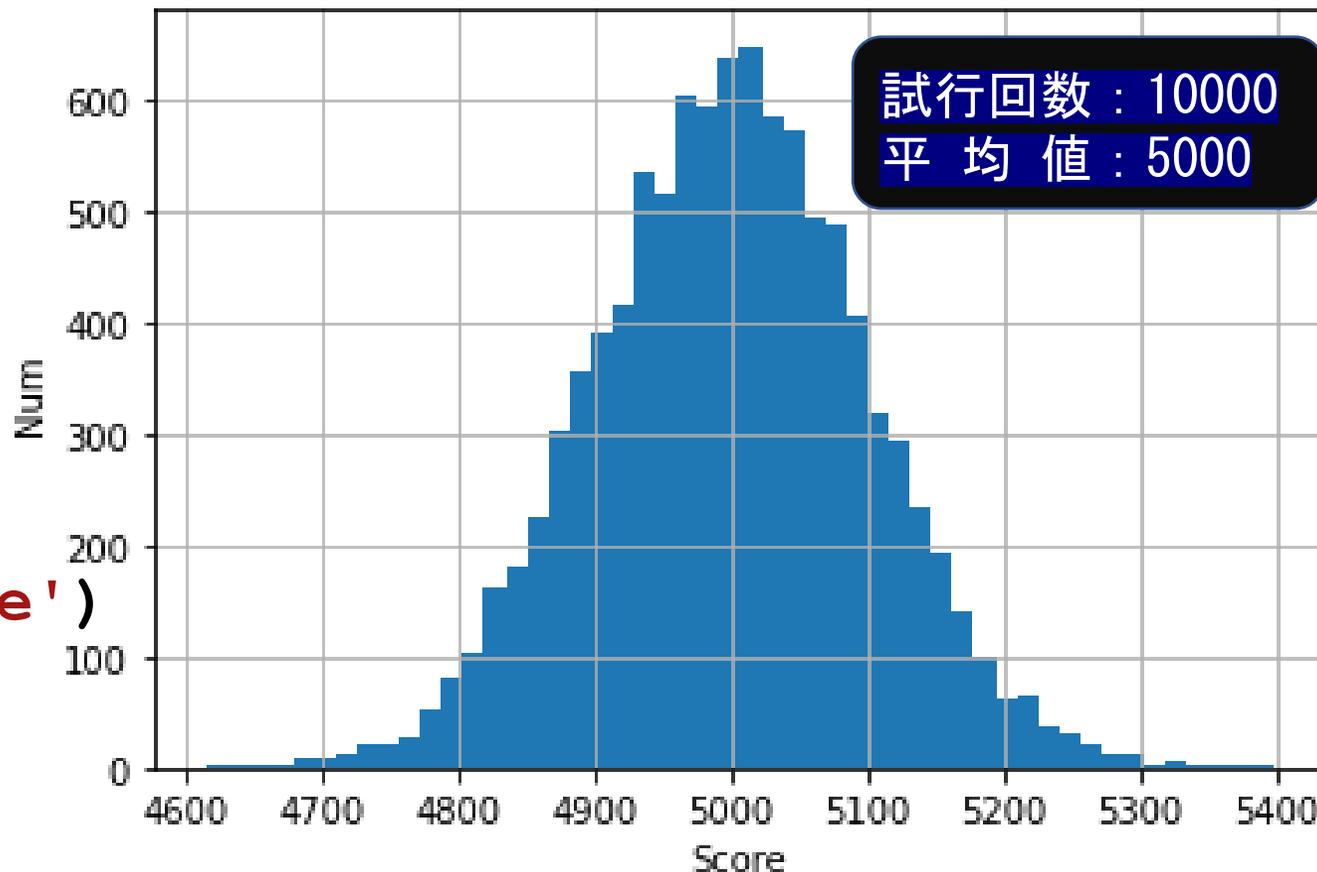
# 正規分布の可視化実験



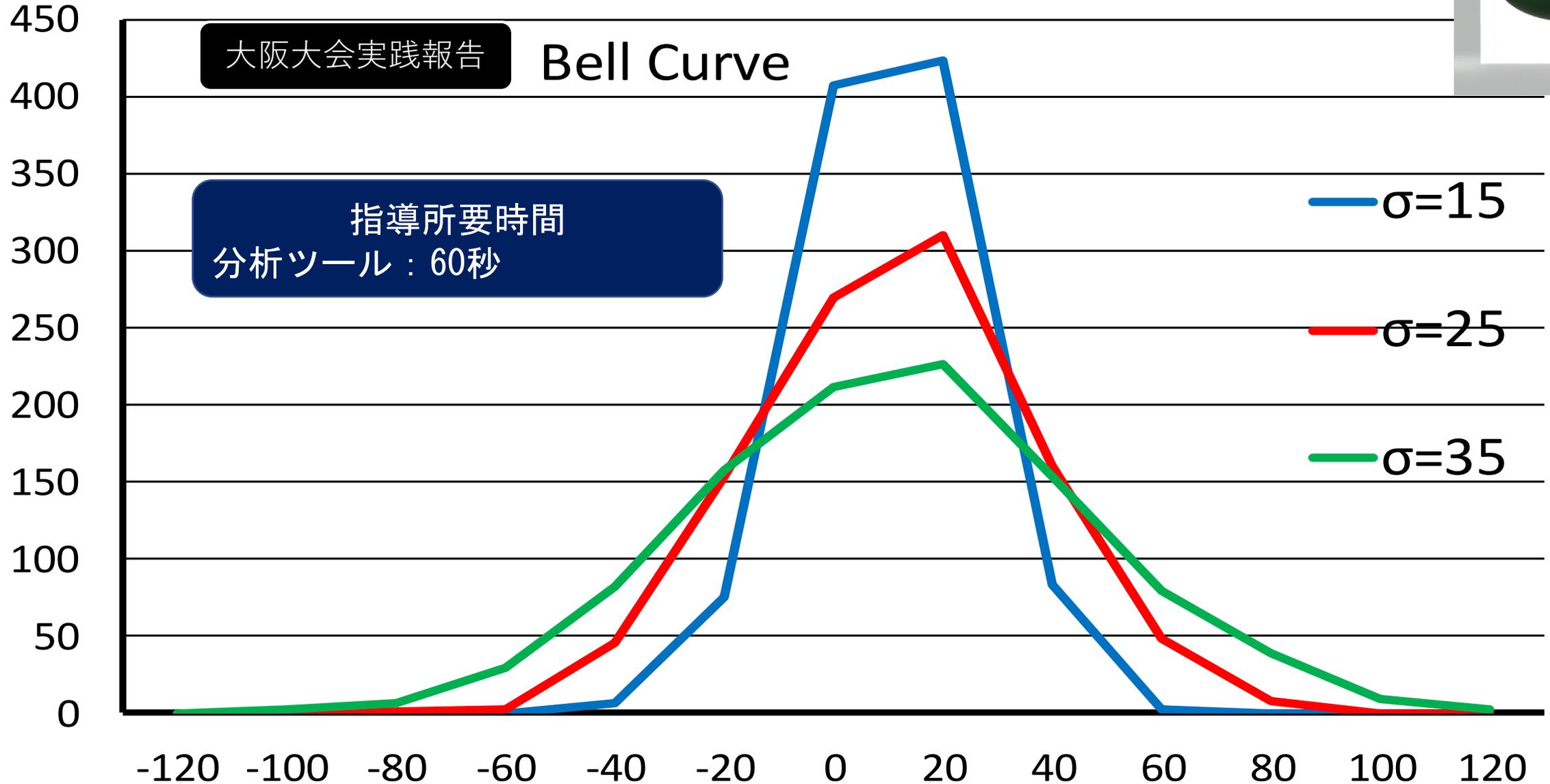
```
import numpy as np
import matplotlib.pyplot as plt
r = np.random.normal(
    loc = 5000,
    scale = 100,
    size = 10000,
)
fig = plt.figure()
ax = fig.add_subplot(1,1,1)
ax.hist(r, bins=50)
ax.set_title('Gaussian curve')
ax.set_xlabel('Score')
ax.set_ylabel('Num')
plt.grid(True)
fig.show()
```

グラフ描画 matplotlib  
配列処理 numpy

Gaussian curve



# Excel分析ツール標準偏差とベルカーブ実験

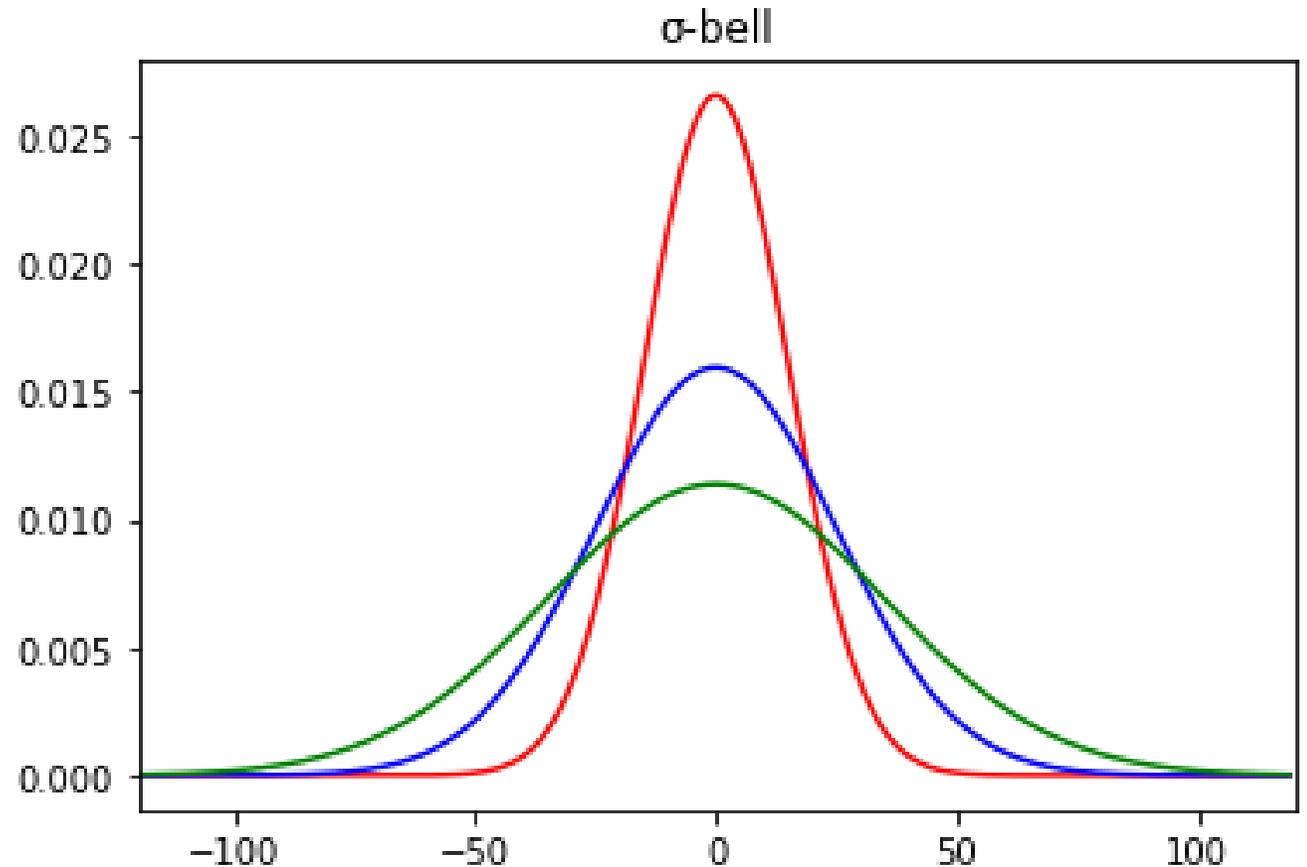


# Pythonによる標準偏差とベルカーブ実験



```
import matplotlib.pyplot as plt
from scipy.stats import norm
x = np.arange(-120, 120, 1)
y1 = norm.pdf(x, 0, 15)
y2 = norm.pdf(x, 0, 25)
y3 = norm.pdf(x, 0, 35)
plt.plot(x, y1, color="red")
plt.plot(x, y2, color="blue")
plt.plot(x, y3, color="green")
plt.title("σ-bell")
plt.xlim(-120, 120)
```

指導所要時間  
タイピング：生徒各自



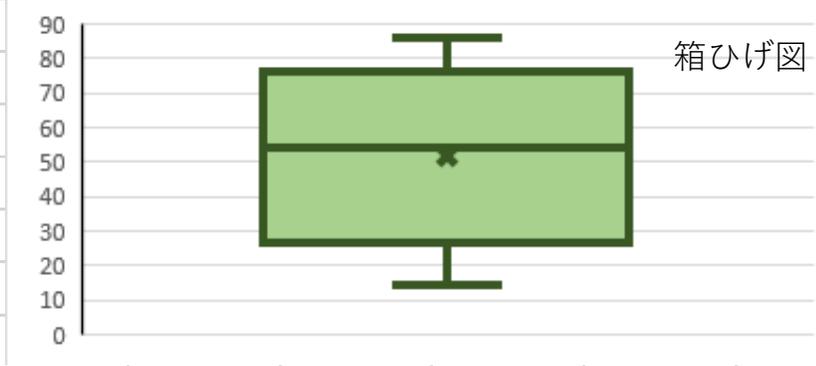
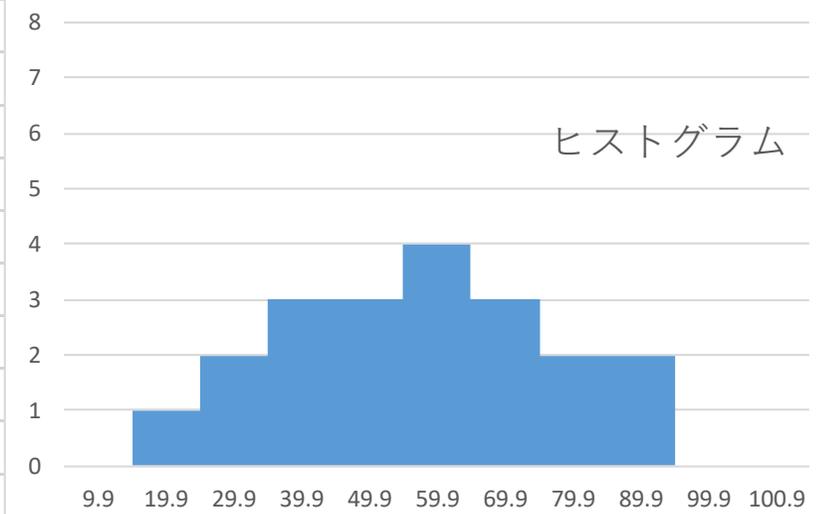
# 数学 I 統計 準拠実験

集団A								集団B							
母集団A	ポイント	偏差	(偏差) <sup>2</sup>	(偏差) <sup>2</sup> 平均値	(偏差) <sup>2</sup> 平均値の 平方根	偏差値	順位	母集団B	ポイント	偏差	(偏差) <sup>2</sup>	(偏差) <sup>2</sup> 平均値	(偏差) <sup>2</sup> 平均値の 平方根	偏差値	順位
A1	42	-11.0	121.0			44.4	14	B1	45	-8.0	64.0			43.7	15
A2	25	-28.0	784.0			35.6	19	B2	54	1.0	1.0			50.8	10
A3	36	-17.0	289.0			41.3	17	B3	72	19.0	361.0			65.0	2
A4	38	-15.0	225.0			42.3	16	B4	33	-20.0	400.0			34.2	19
A5	55	2.0	4.0			51.0	10	B5	64	11.0	121.0			58.7	5
A6	14	-39.0	1521.0			30.0	20	B6	36	-17.0	289.0			36.6	18
A7	65	12.0	144.0			56.2	6	B7	42	-11.0	121.0			41.3	16
A8	67	14.0	196.0			57.2	5	B8	56	3.0	9.0			52.4	7
A9	78	25.0	625.0			62.8	3	B9	51	-2.0	4.0			48.4	13
A10	63	10.0	100.0			55.1	7	B10	65	12.0	144.0			59.5	4
A11	39	-14.0	196.0			42.8	15	B11	66	13.0	169.0			60.3	3
A12	59	6.0	36.0			53.1	8	B12	55	2.0	4.0			51.6	9
A13	57	4.0	16.0			52.1	9	B13	49	-4.0	16.0			46.8	14
A14	86	33.0	1089.0			66.9	1	B14	28	-25.0	625.0			30.2	20
A15	53	0.0	0.0			50.0	11	B15	61	8.0	64.0			56.3	6
A16	75	22.0	484.0			61.3	4	B16	42	-11.0	121.0			41.3	16
A17	48	-5.0	25.0			47.4	12	B17	52	-1.0	1.0			49.2	12
A18	45	-8.0	64.0			45.9	13	B18	54	1.0	1.0			50.8	10
A19	86	33.0	1089.0			66.9	1	B19	56	3.0	9.0			52.4	7
A20	29	-24.0	576.0	379.2	19.5	37.7	18	B20	79	26.0	676.0	160.0	12.6	70.6	1
合計	1060	0.0	7584.0			50.0		合計	1060	0.0	3200.0			50.0	
個数	20		379.2					個数	20		160.0				
最小値	14.0							最小値	28.0						
最大値	86.0							最大値	79.0						
中央値	54.0							中央値	54.0						
平均値	53.0		379.2		19.5			平均値	53.0		160.0		12.6		
最頻値	86.0							最頻値	54.0						
分散	379.2							分散	160.0						
標準偏差	19.5							標準偏差	12.6						

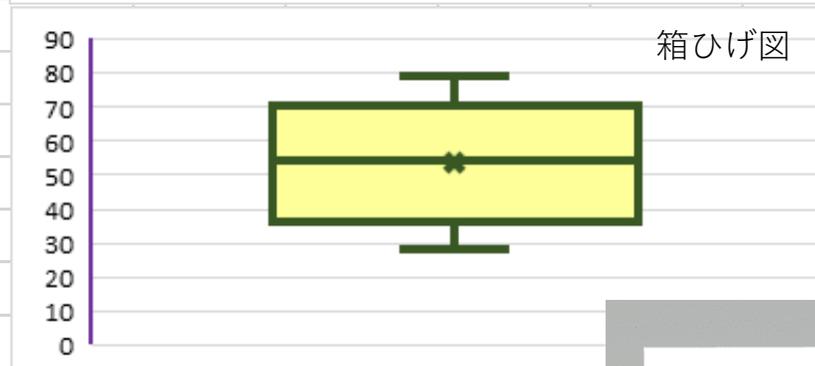
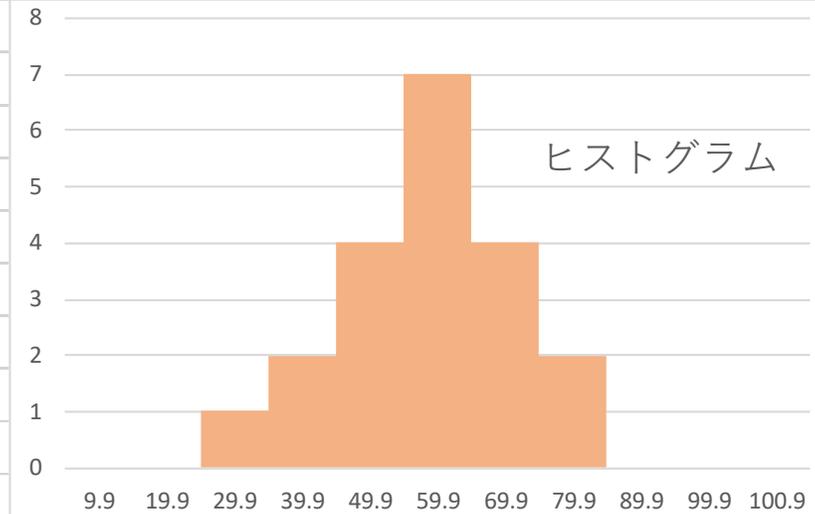


度数分布⇒ヒストグラム  
四分位点⇒箱ひげ図

区間・階級	頻度
9.9	0
19.9	1
29.9	2
39.9	3
49.9	3
59.9	4
69.9	3
79.9	2
89.9	2
99.9	0
100.9	0
最小値	14
第1四分位点	38.75
中央値	54
第3四分位点	65.5
最大値	86



区間・階級	頻度
9.9	0
19.9	0
29.9	1
39.9	2
49.9	4
59.9	7
69.9	4
79.9	2
89.9	0
99.9	0
100.9	0
最小値	28
第1四分位点	44.25
中央値	54
第3四分位点	61.75
最大値	79



数学 I の指導に準拠しやすい



A18	45	=B22-B\$30	=C22^2	$\sigma^2$ =AVERAGE(D5:D24)	$\sigma$ =SQRT(E5)	=(10*(B22-\$B\$30)/\$B\$33)+50 =(10*(B23-\$B\$30)/\$B\$33)+50 =(10*(B24-\$B\$30)/\$B\$33)+50 =AVERAGE(G5:G24)	=RANK.EQ(B22,B\$5:B\$24) =RANK.EQ(B23,B\$5:B\$24) =RANK.EQ(B24,B\$5:B\$24)
A19	86	=B23-B\$30	=C23^2				
A20	29	=B24-B\$30	=C24^2				
合計		=SUM(B5:B24)	=SUM(C5:C24)	=SUM(D5:D24)			
個数		=COUNT(B5:B24)		=AVERAGE(D5:D24)			
最小値		=MIN(B5:B24)					
最大値		=MAX(B5:B24)					
中央値		=MEDIAN(B5:B24)					
平均値		=AVERAGE(B5:B24)		=AVERAGE(D5:D24)			
最頻値		=MODE.SNGL(B5:B24)			=SQRT(B32)		
分散		=VAR.P(B5:B24)					
標準偏差		=STDEV.P(B5:B24)					

SS偏差値

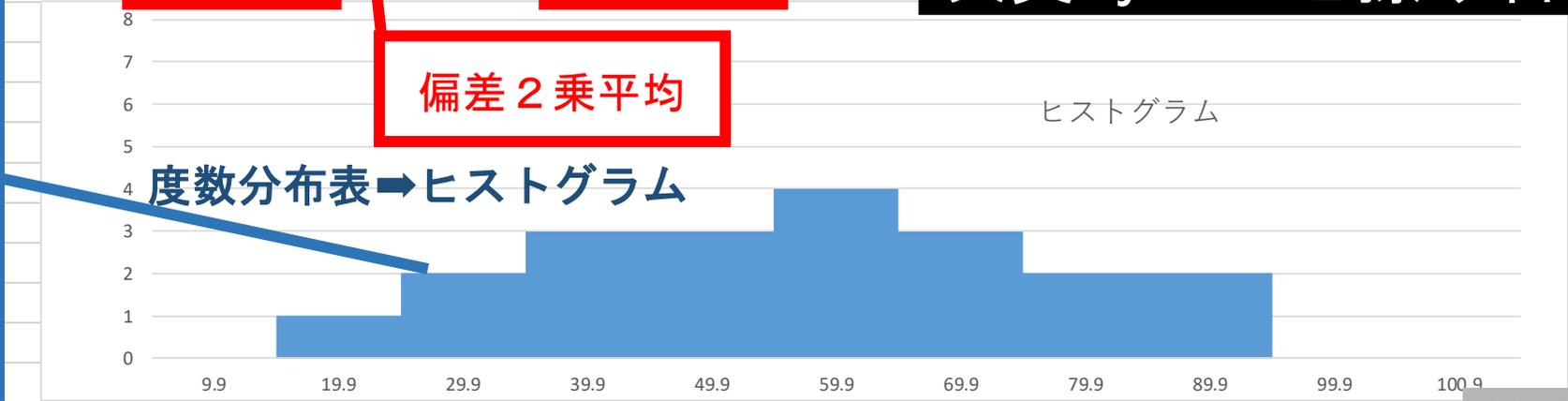
数式理解  
 数学 I 統計未習 → 不可避  
 次頁Pythonと擦り合わせ

区間・階級	頻度
9.9	=FREQUENCY(B5:B24,A35:A45)
19.9	=FREQUENCY(B5:B24,A35:A45)
29.9	=FREQUENCY(B5:B24,A35:A45)
39.9	=FREQUENCY(B5:B24,A35:A45)
49.9	=FREQUENCY(B5:B24,A35:A45)
59.9	=FREQUENCY(B5:B24,A35:A45)
69.9	=FREQUENCY(B5:B24,A35:A45)
79.9	=FREQUENCY(B5:B24,A35:A45)
89.9	=FREQUENCY(B5:B24,A35:A45)
99.9	=FREQUENCY(B5:B24,A35:A45)
100.9	=FREQUENCY(B5:B24,A35:A45)

偏差

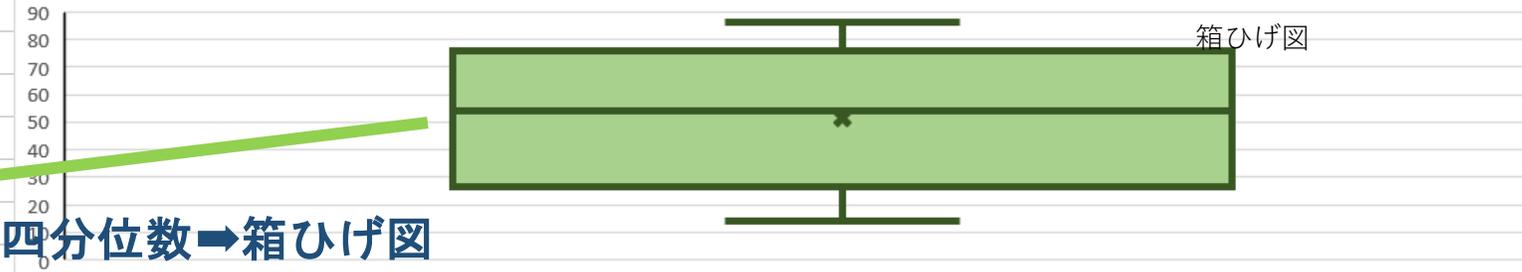
偏差 2 乗

偏差 2 乗平均



度数分布表 → ヒストグラム

最小値	=QUARTILE(B\$5:B\$24,0)
第1四分位点	=QUARTILE(B\$5:B\$24,1)
中央値	=QUARTILE(B\$5:B\$24,2)
第3四分位点	=QUARTILE(B\$5:B\$24,3)
最大値	=QUARTILE(B\$5:B\$24,4)



四分位数 → 箱ひげ図



```
import statistics
a=[42,25,36,38,55,14,65,67,78,63,39
,59,57,86,53,75,48,45,86,29]
b=[45,54,72,33,64,36,42,56,51,65,66
,55,49,28,61,42,52,54,56,79]
```

```
print(len(a))           → 20
print(sum(a))           → 1060
print(min(a))           → 14
print(max(a))           → 86
print(statistics.pvariance(a)) → 379.2
print(statistics.pstdev(a)) → 19.473058311420935
print(statistics.median(a)) → 54.0
print(len(b))           → 20
print(sum(b))           → 1060
print(min(b))           → 28
print(max(b))           → 79
print(statistics.pvariance(b)) → 160
print(statistics.pstdev(b)) → 12.649110640673518
print(statistics.median(b)) → 54.0
```

前頁Excelと擦り合わせ  
数学 I 統計既習：必要なし  
→ 立式の根拠理解必須

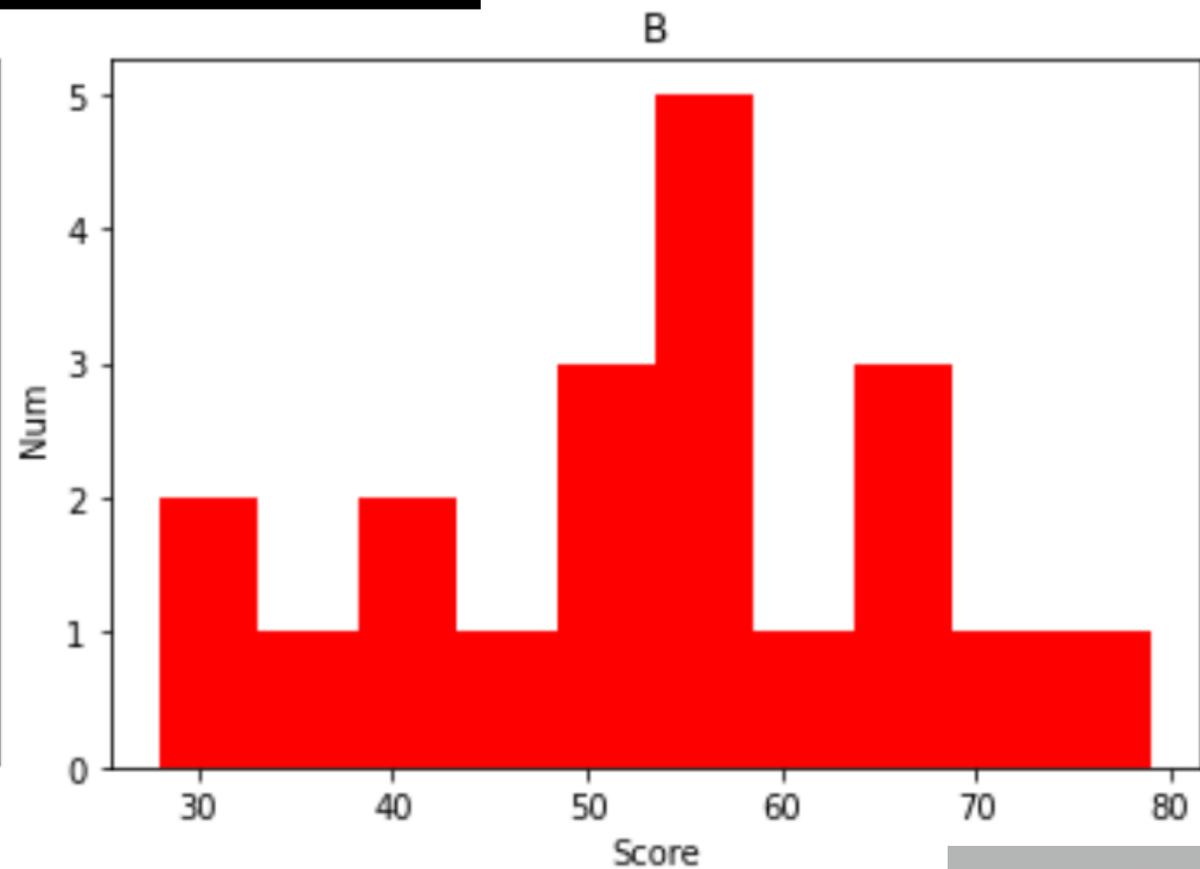
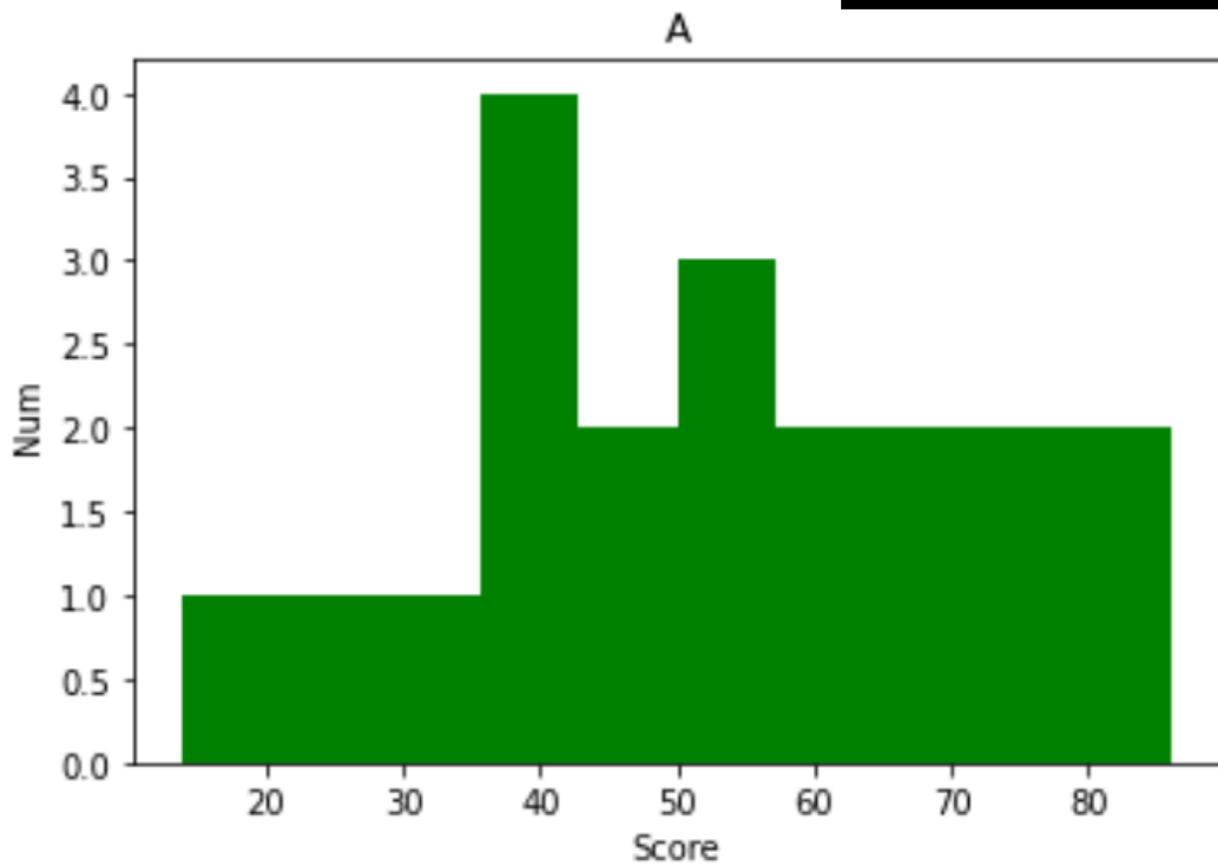
The Python logo, consisting of two interlocking snakes, one blue and one yellow, with the word "PYTHON" written in white capital letters across the center.

```
import matplotlib.pyplot as plt
a = [42, 25, 36, 38, 55, 14, 65, 67, 78, 63, 39, 59, 57, 86, 53, 75, 48, 45, 86, 29]
b = [45, 54, 72, 33, 64, 36, 42, 56, 51, 65, 66, 55, 49, 28, 61, 42, 52, 54, 56, 79]
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.hist(a, bins=10, color='green')
ax.set_title('A')
ax.set_xlabel('Score')
ax.set_ylabel('Num')
fig = plt.figure()
bx = fig.add_subplot(1, 1, 1)
bx.hist(b, bins=10, color='red')
bx.set_title('B')
bx.set_xlabel('Score')
bx.set_ylabel('Num')
fig.show()
```

配列⇒表計算の場合と同値  
☞箱ひげ図描画

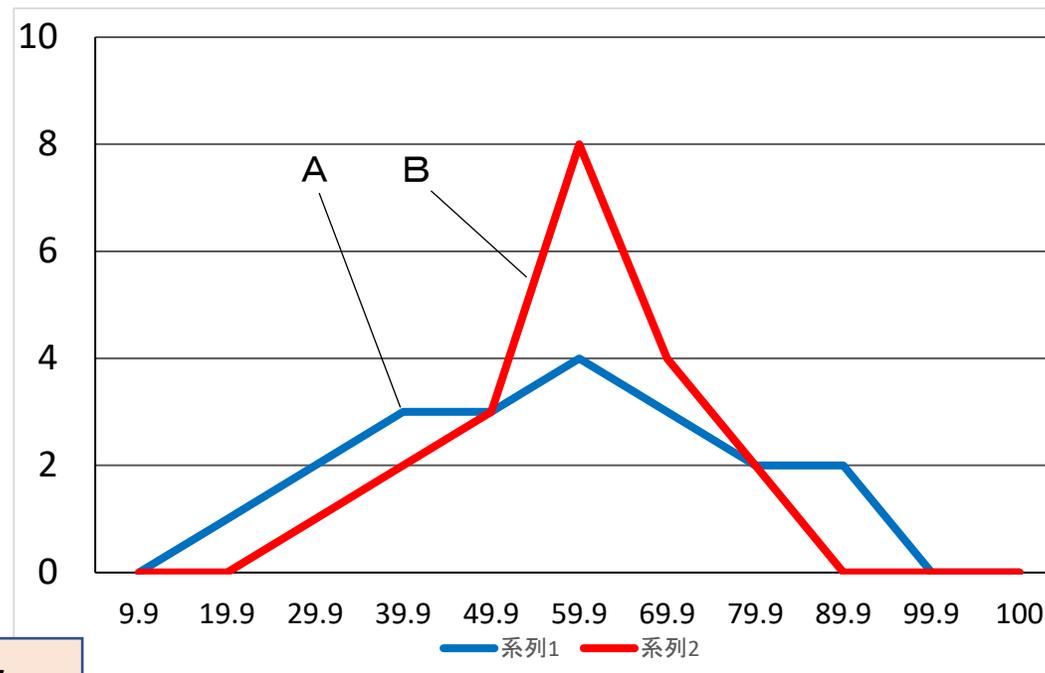


# ヒストグラム描画



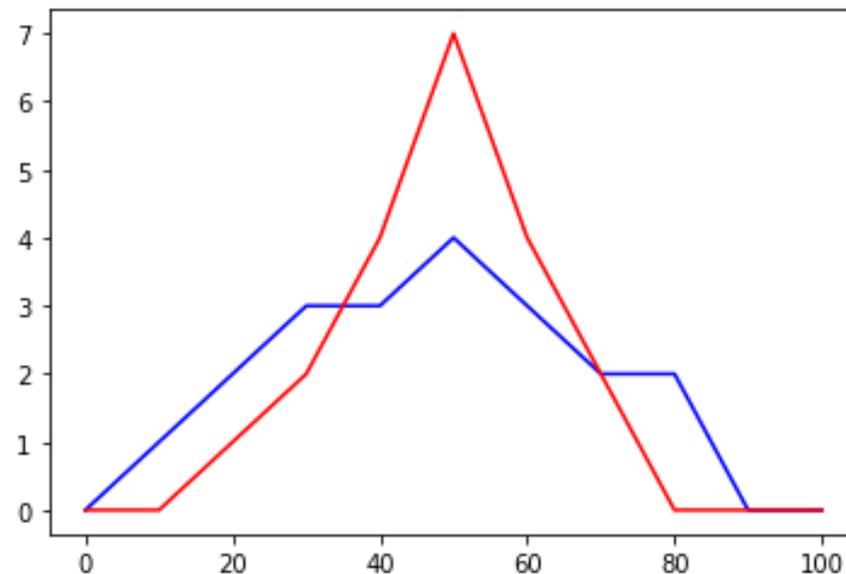
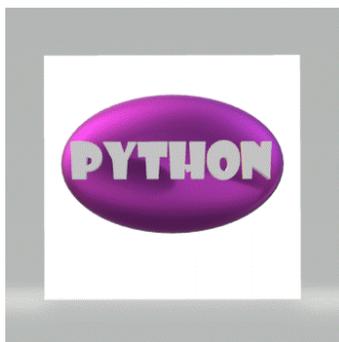
PYTHON

階級上限	A頻度	B頻度
9.9	0	10
19.9	1	0
29.9	2	1
39.9	3	0
49.9	3	0
59.9	4	1
69.9	3	0
79.9	2	0
89.9	2	0
99.9	0	0
100	0	0



ツールとして大差なし

```
import matplotlib.pyplot as plt
x=[0,10,20,30,40,50,60,70,80,90,100]
a=[0,1,2,3,3,4,3,2,2,0,0]
a=[0,1,2,3,3,4,3,2,2,0,0]
b=[0,0,1,2,4,7,4,2,0,0,0]
plt.plot(x,a,c='blue')
plt.plot(x,b,c='red')
plt.show()
```

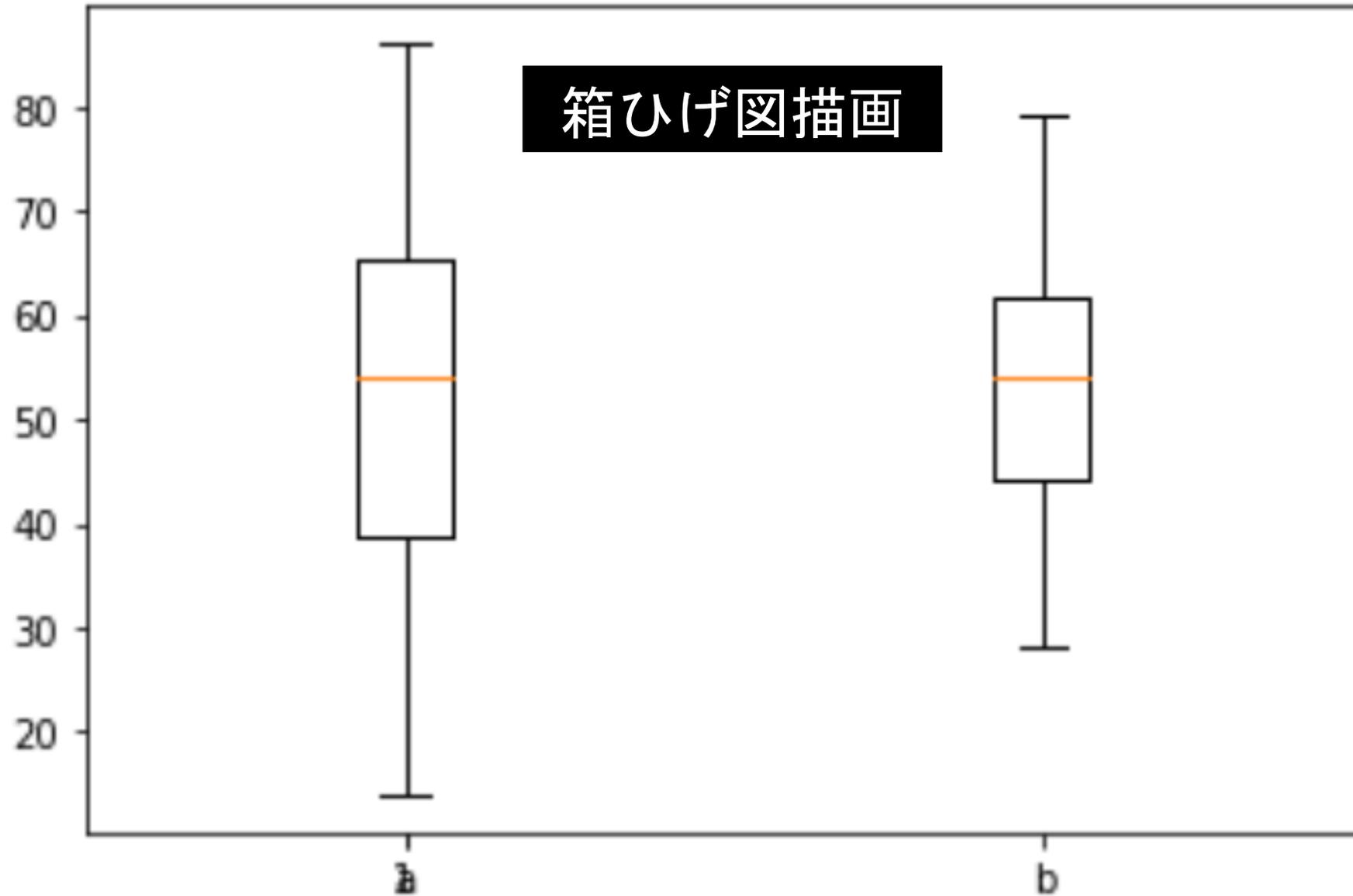


```
import matplotlib.pyplot as plt
a= [42, 25, 36, 38, 55, 14, 65, 67, 78, 63, 39, 59, 57, 86, 53, 75,
48, 45, 86, 29]
b= [45, 54, 72, 33, 64, 36, 42, 56, 51, 65, 66, 55, 49, 28, 61, 42,
52, 54, 56, 79]
points = (a,b)
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
bp = ax.boxplot(points)
ax.set_xticklabels(['a', 'b'])
ax.boxplot(a)
plt.show()
ax.boxplot(b)
plt.show()
```

配列 → 表計算の場合と同値

👉 箱ひげ図描画

The Python logo, consisting of two interlocking snakes, one blue and one yellow, with the word "PYTHON" written in white capital letters across the center.



# 表計算による人口動態(総理府データ)～回帰直線(曲線)とR<sup>2</sup>値実験

大阪大会実践報告

年度	人口(万人)
1920	5,596
1925	5,974
1930	6,445
1935	6,925
1940	7,193
1945	7,215
1950	8,411

年度	人口(万人)
1955	9,008
1960	9,430
1965	9,921
1970	10,467
1975	11,194
1980	11,706
1985	12,105

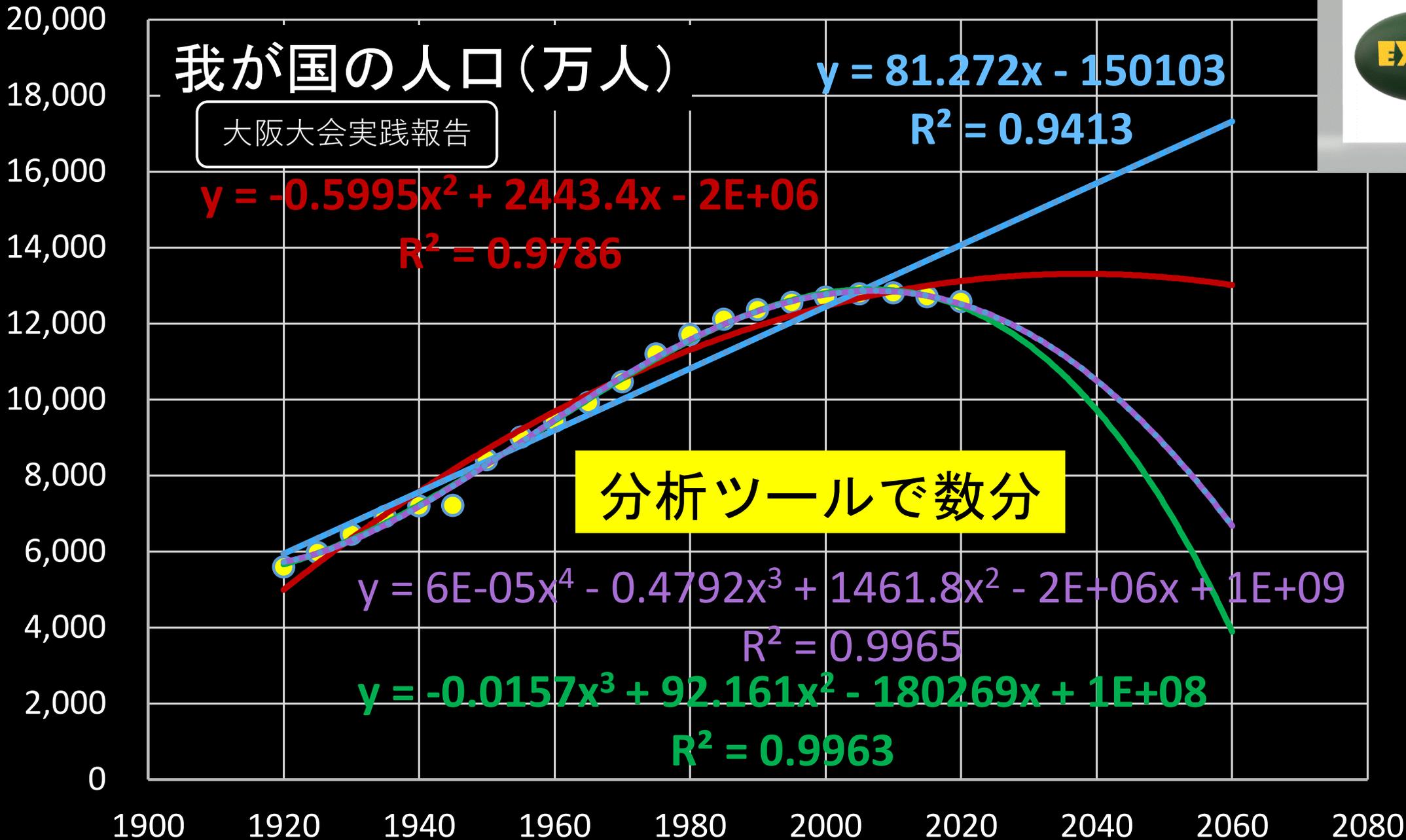
年度	人口(万人)
1990	12,361
1995	12,557
2000	12,693
2005	12,777
2010	12,806
2015	12,709
2020	12,577

The logo consists of the word "EXCEL" in a bold, yellow, sans-serif font, centered within a dark green oval shape. This oval is set against a white square background, which is itself within a grey rectangular frame.

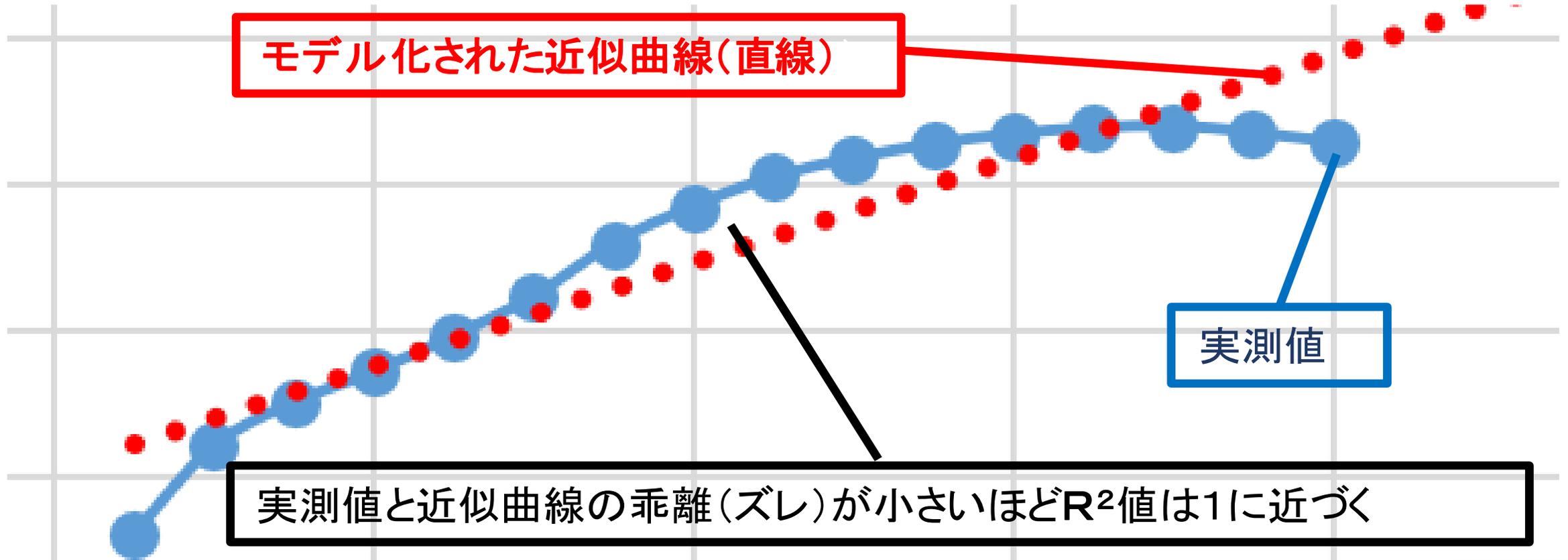


# 我が国の人口(万人)

大阪大会実践報告



# R<sup>2</sup>値と「もっともらしさ」

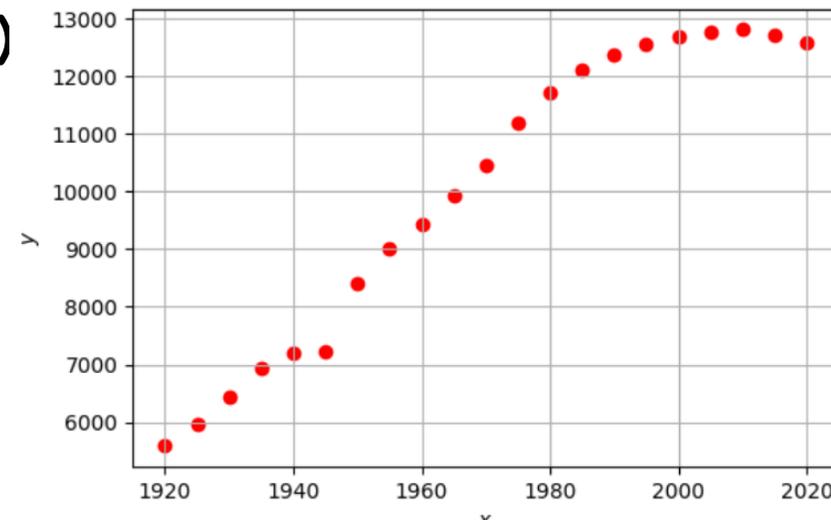


# 人口動態（総理府データ） 回帰直線（曲線）と $R^2$ 値実験



```
import numpy as np
import matplotlib.pyplot as plt
X = np.array([ 1920, 1925, 1930, 1935, 1940, 1945, 1950, 1955, 1960, 19
65, 1970, 1975, 1980, 1985, 1990, 1995, 2000, 2005, 2010, 2015, 2020])
Y = np.array([5596, 5974, 6445, 6925, 7193, 7215, 8411.4, 9007.6, 9430
.1, 9920.9, 10466.5, 11193.9, 11706, 12104.8, 12361.1, 12557, 12692.5,
12776.7, 12805.7, 12709.4, 12577])
fig, ax = plt.subplots(dpi=100)
ax.scatter(X, Y, marker='o', color='red')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
ax.grid()
```

👉 散布図描画



今年度展開予定

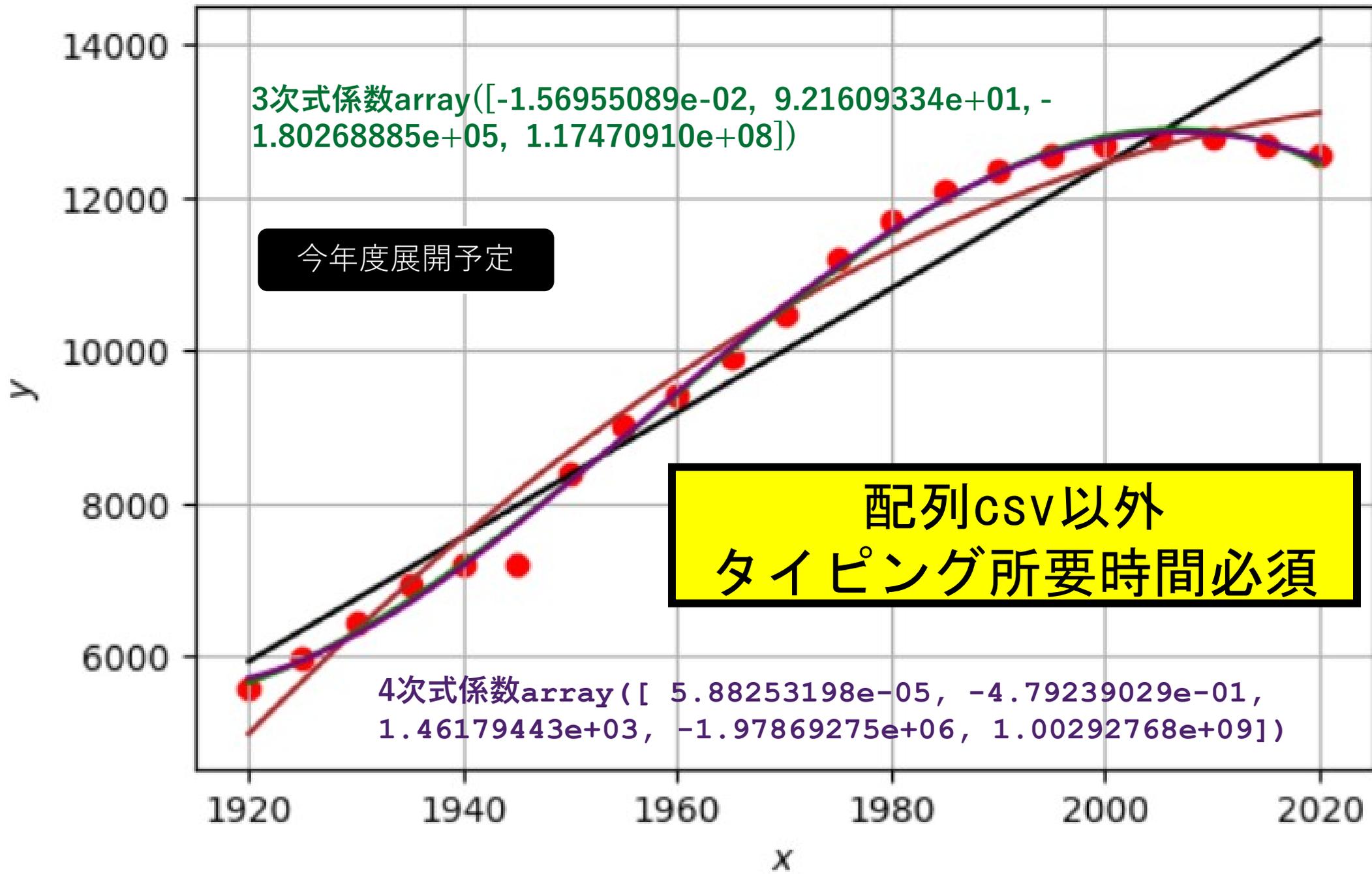


```
Degree1 = np.polyfit(X, Y, 1)
Degree1
Degree2 = np.polyfit(X, Y, 2)
Degree2
Degree3 = np.polyfit(X, Y, 3)
Degree3
Degree4 = np.polyfit(X, Y, 4)
Degree4
```

```
x = np.linspace(1920, 2020, 1000)
fig, ax = plt.subplots(dpi=100)
ax.scatter(X, Y, marker='o', color='red')
ax.plot(x, np.polyval(Degree1, x), 'black')
ax.plot(x, np.polyval(Degree2, x), 'brown')
ax.plot(x, np.polyval(Degree3, x), 'green')
ax.plot(x, np.polyval(Degree4, x), 'purple')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
ax.grid()
```

Matplotlib plt.subplots()  
Scatter → 散布図  
numpy.polyval → 多項式

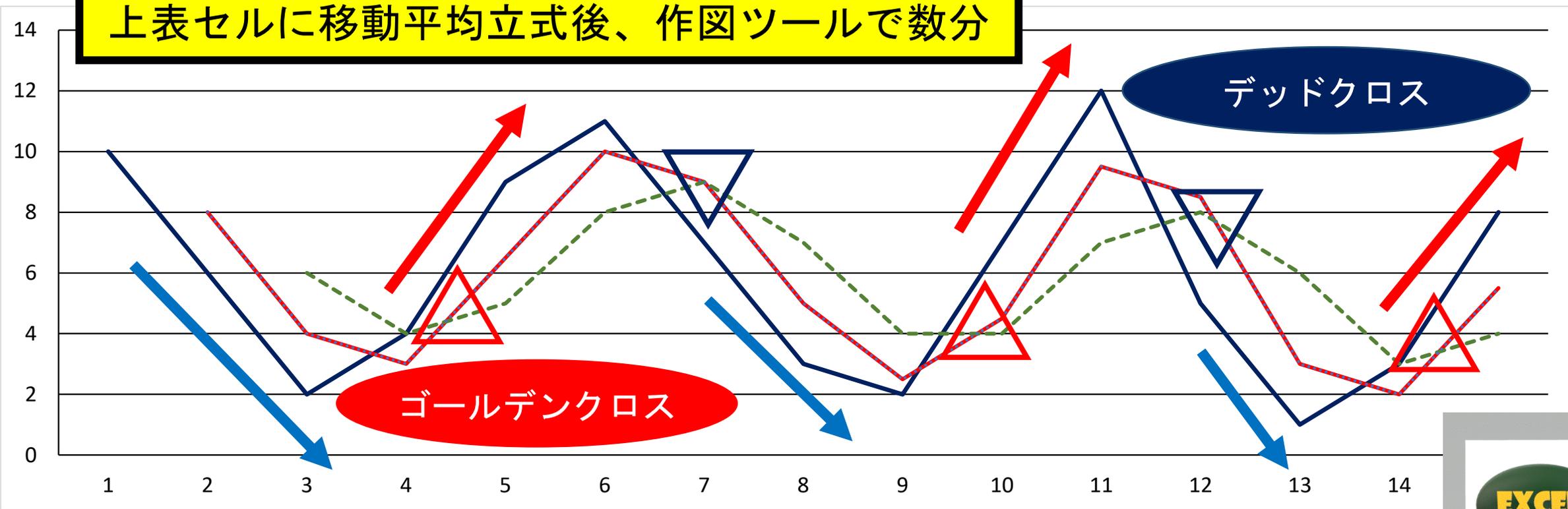
今年度展開予定



# 表計算による移動平均：平坦化とトレンド

時間(日)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
値	10	6	2	4	9	11	7	3	2	7	12	5	1	3	8
2日平均	-	8	4	3	6.5	10	9	5	2.5	4.5	9.5	8.5	3	2	5.5
3日平均	-	-	6	4	5	8	9	7	4	4	7	8	6	3	4

上表セルに移動平均立式後、作図ツールで数分



# Python による移動平均：平坦化とトレンド

```
import pandas as pd
import matplotlib.pyplot as plt
data = {"date": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14],
        "value": [10, 6, 2, 4, 9, 11, 7, 3, 2, 7, 12, 5, 1, 3, 8]}
df = pd.DataFrame(data)
df["2-day movin'ave"] = df["value"].rolling(2).mean().round(1)
df["3-day movin'ave"] = df["value"].rolling(3).mean().round(1)
print(df)
plt.plot(df["date"], df["value"], label="daily")
plt.plot(df["date"], df["2-day movin'ave"], "g-.", label="MA(2)")
plt.plot(df["date"], df["3-day movin'ave"], "r--", label="MA(3)")
plt.xticks(rotation=90)
plt.xlabel("date")
plt.ylabel("quantity")
plt.legend()
plt.show()
```

配列 → 表計算の場合と同値

rolling () → 移動平均

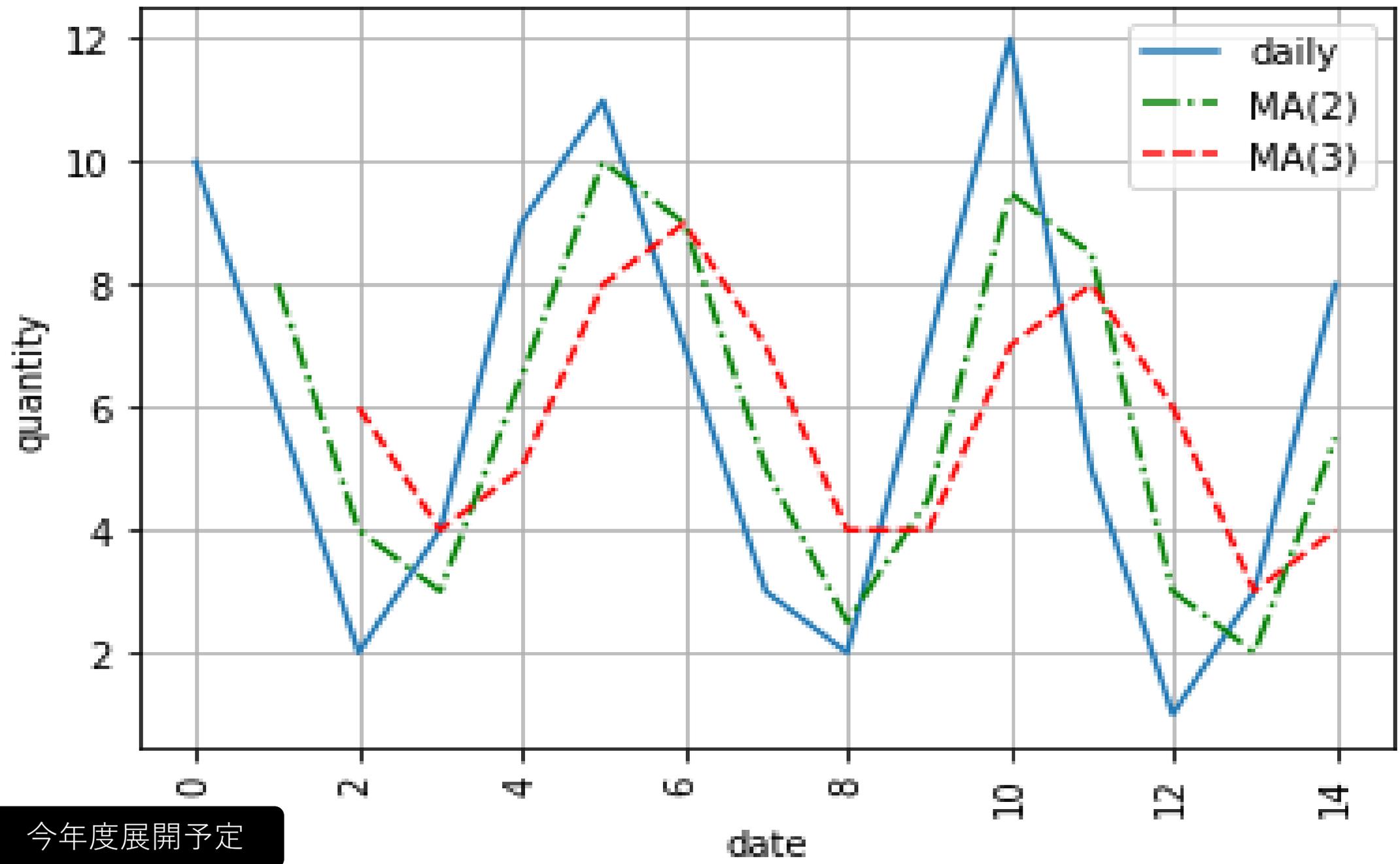
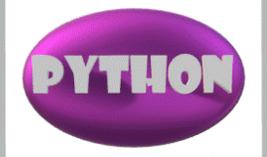
今年度展開予定





date	value	2-day movin' ave	3-day movin' ave
0	10	NaN	NaN
1	6	8.0	NaN
2	2	4.0	6.0
3	4	3.0	4.0
4	9	6.5	5.0
5	11	10.0	8.0
6	7	9.0	9.0
7	3	5.0	7.0
8	2	2.5	4.0
9	7	4.5	4.0
10	12	9.5	7.0
11	5	8.5	8.0
12	1	3.0	6.0
13	3	2.0	3.0
14	8	5.5	4.0

今年度展開予定



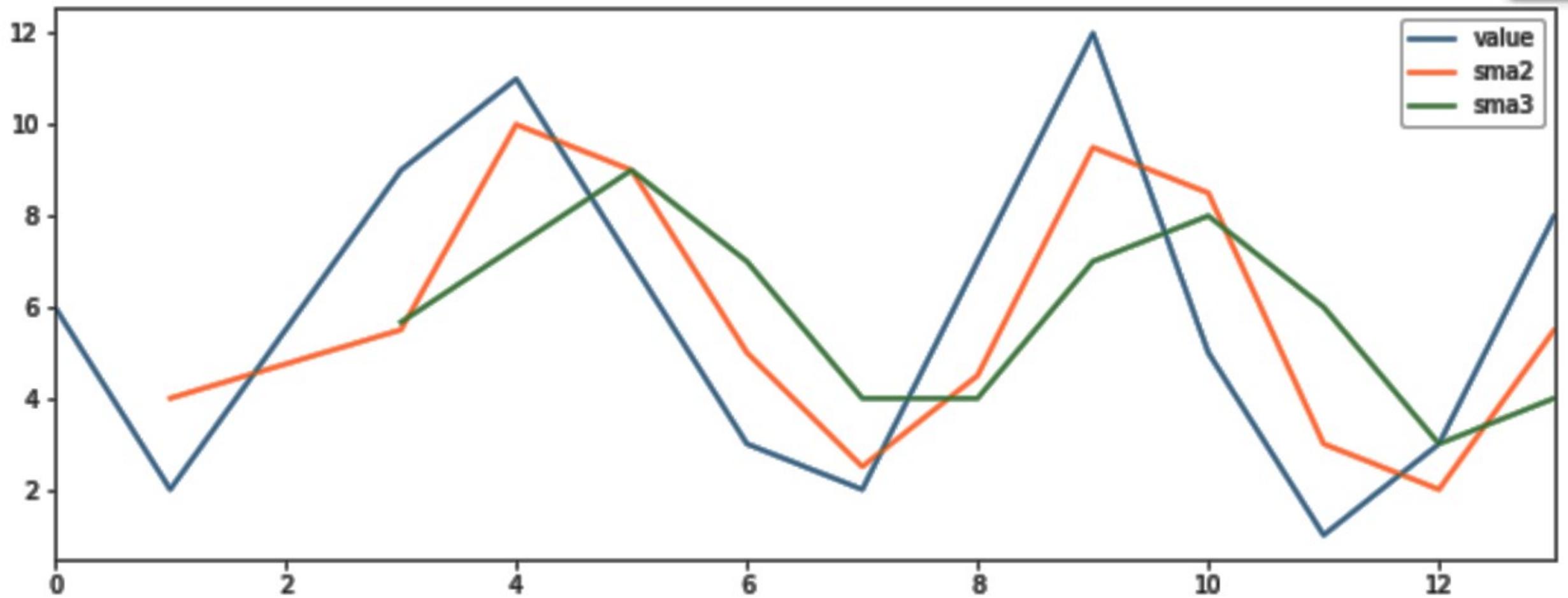
今年度展開予定

WEBからデータのインポート  
→例は筆者サーバからcsvダウンロード

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
CSV_URL = "http://strnun.fool.jp/povray_strnun/movin'ave2022data.csv"
df = pd.read_csv(CSV_URL, names=['date', 'value'], header=1);
df = df.dropna()
df['ma2'] = df.value.rolling(2).mean()
df['ma3'] = df.value.rolling(3).mean()
fig, axes = plt.subplots(2,1,figsize=(12, 10))
df[['value', 'ma2', 'ma3']].plot(ax=axes[0], xlim=[0, len(df)])
plt.show()
```

今年度展開予定

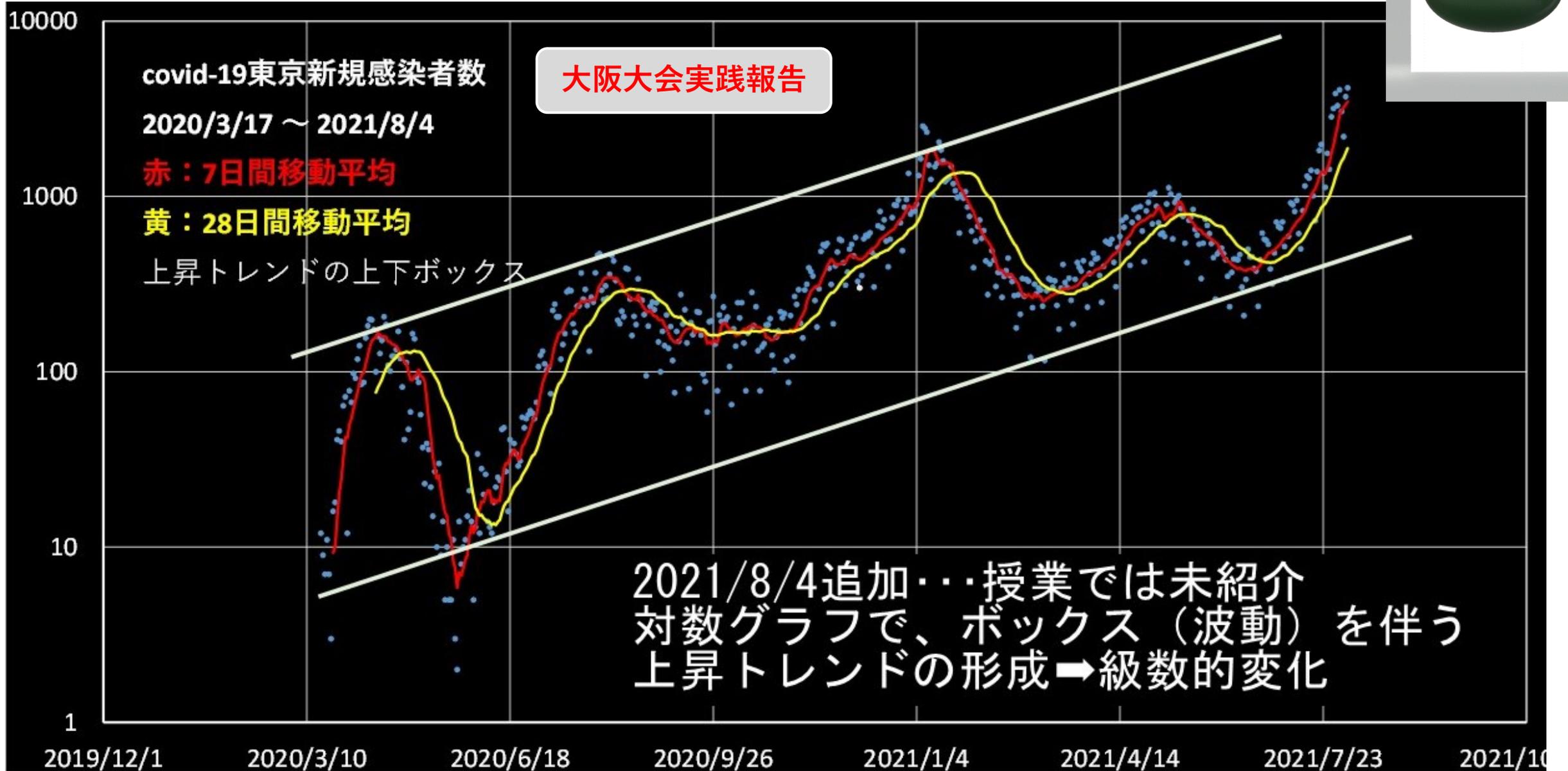
PYTHON



今年度展開予定



# 表計算による東京都covid19新規感染者数移動平均分析

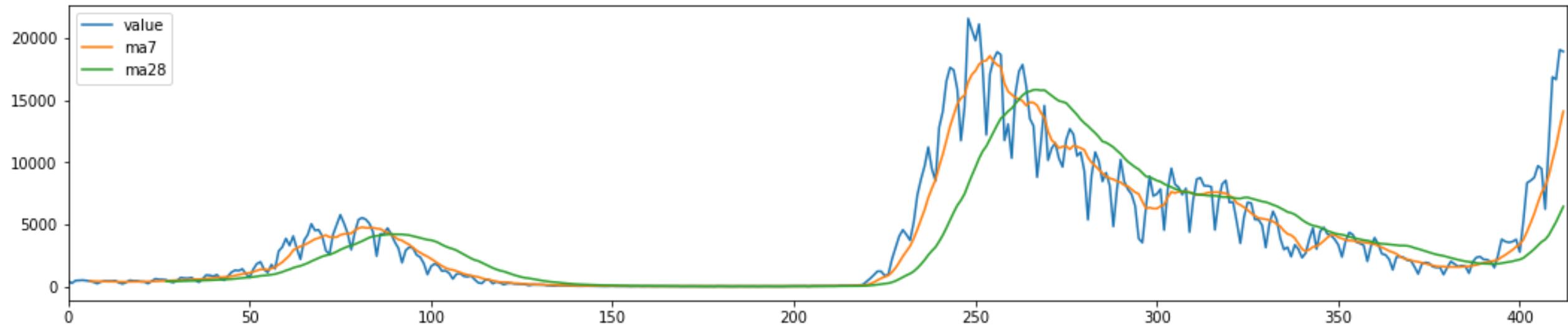


# Pythonによる東京都covid19新規感染者数移動平均分析：

今年度展開予定

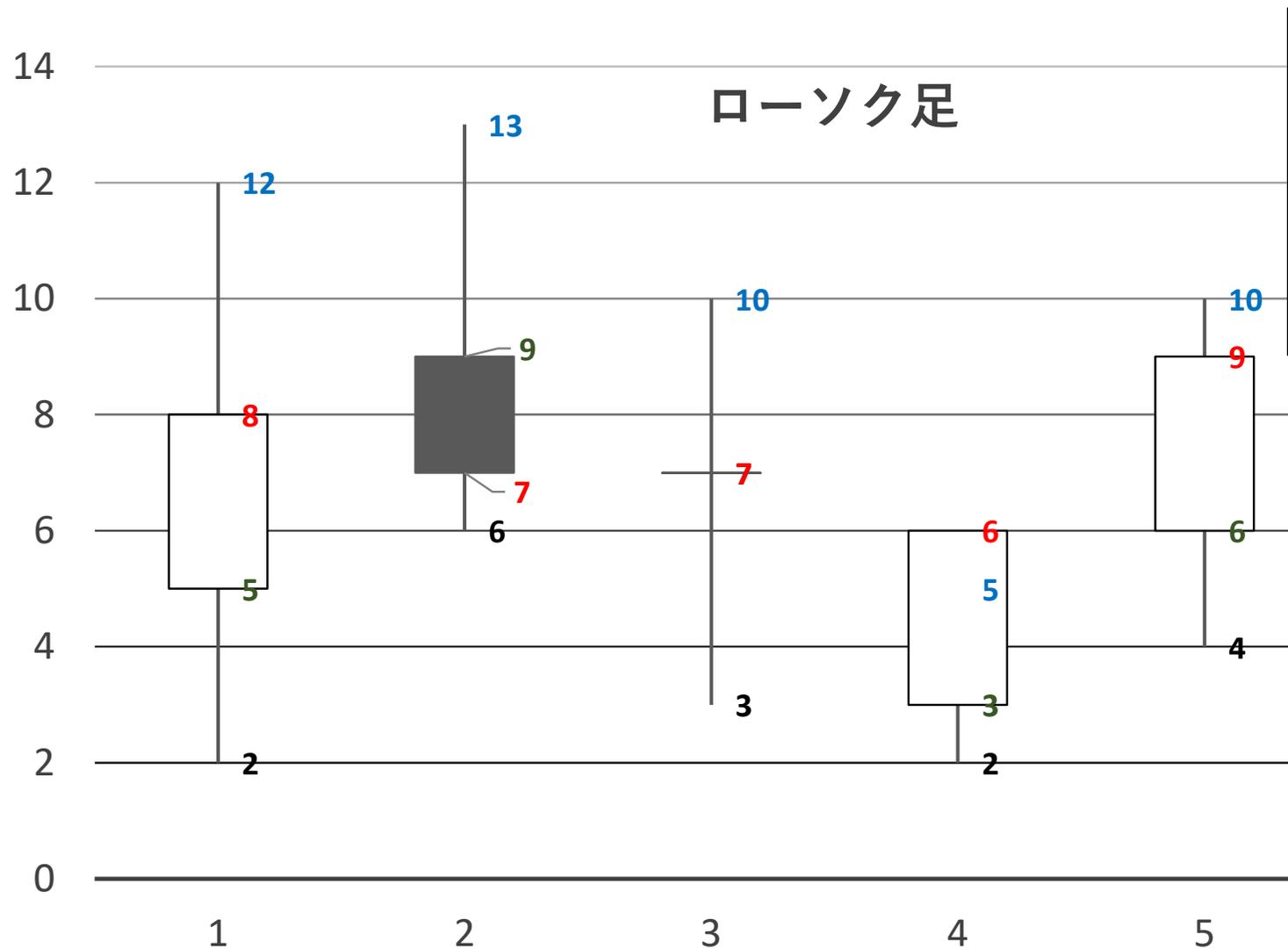


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
CSV_URL="http://strnun.fool.jp/pov-ray_strnun//covid19tkyvalue.csv"
df = pd.read_csv(CSV_URL, names=['date', 'value'], header=500);
df['ma7'] = df.value.rolling(7).mean()
df['ma28'] = df.value.rolling(28).mean()
fig, axes = plt.subplots(2, 1, figsize=(15, 10))
df[['date', 'value', 'ma7', 'ma28']].plot(ax=axes[0], xlim=[0, len(df)])
plt.show()
```



# チャートグラフ（ローソク足）と予測基本

大阪大会実践報告



	始値	最高値	最低値	終値
1	5	12	2	8
2	9	13	6	7
3	7	10	3	7
4	3	5	2	6
5	6	10	4	9



# 表計算による県別Covid-19新規感染者数チャートと予測実験

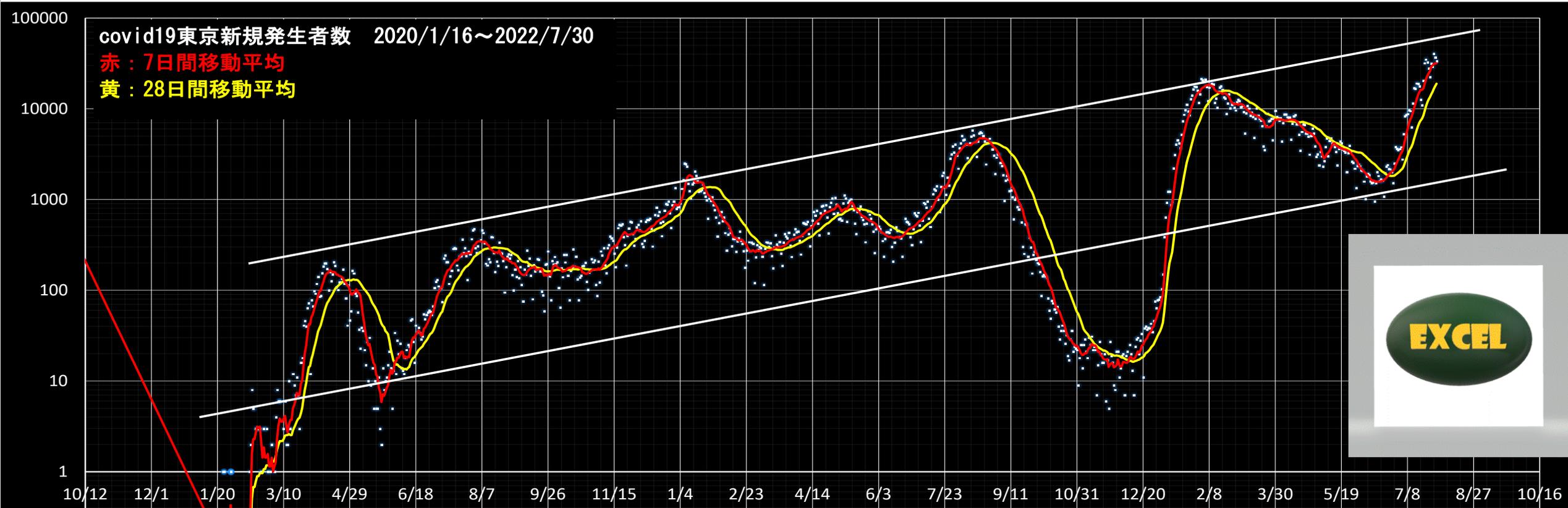
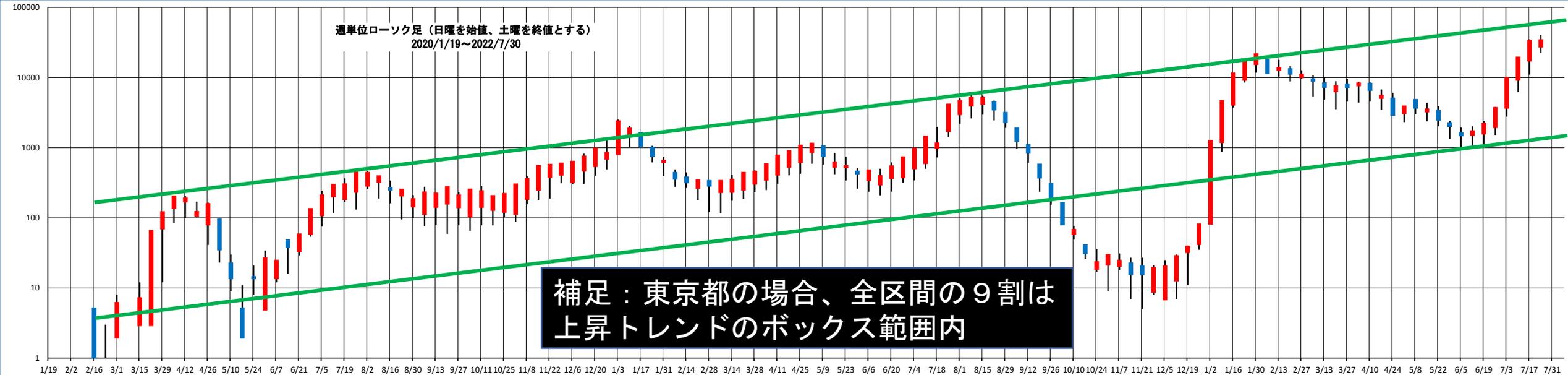


- ①日曜～土曜の7日間を区間とし、日曜を始値、土曜を終値とする。  
週内の最大値、最小値を併せて4要素のテーブルを関数で作成させる。
- ②表計算ソフトウェアの株価グラフで描画。

大阪大会実践報告

週番号	70	71	72	73	74	75	76	77	78
週開始日	2021/5/16	2021/5/23	2021/5/30	2021/6/6	2021/6/13	2021/6/20	2021/6/27	2021/7/4	2021/7/11
日	542	535	448	351	304	376	386	518	614
月	419	340	260	235	209	236	317	342	502
火	732	542	471	369	337	435	476	593	830
水	766	743	487	440	501	619	714	920	1149
木	843	684	508	439	452	570	673	896	1308
金	649	614	472	435	453	562	660	822	1271
土	602	539	436	467	388	534	716	950	1410
週開始	542	535	448	351	304	376	386	518	614
週最大	843	743	508	467	501	619	716	950	1410
週最小	419	340	260	235	209	236	317	342	502
週末	602	539	436	467	388	534	716	950	1410





# PythonによるCovid19東京都新規感染者数チャート分析：

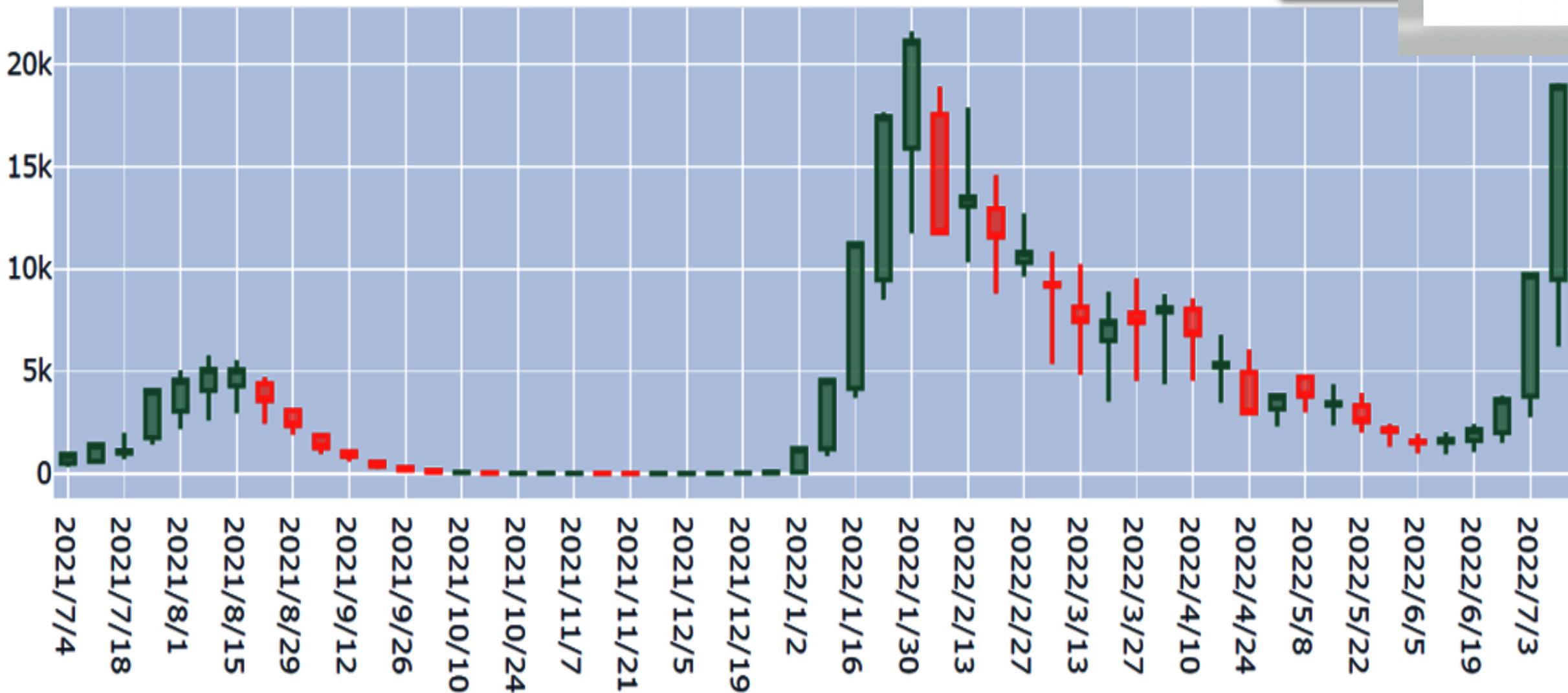
今年度展開予定

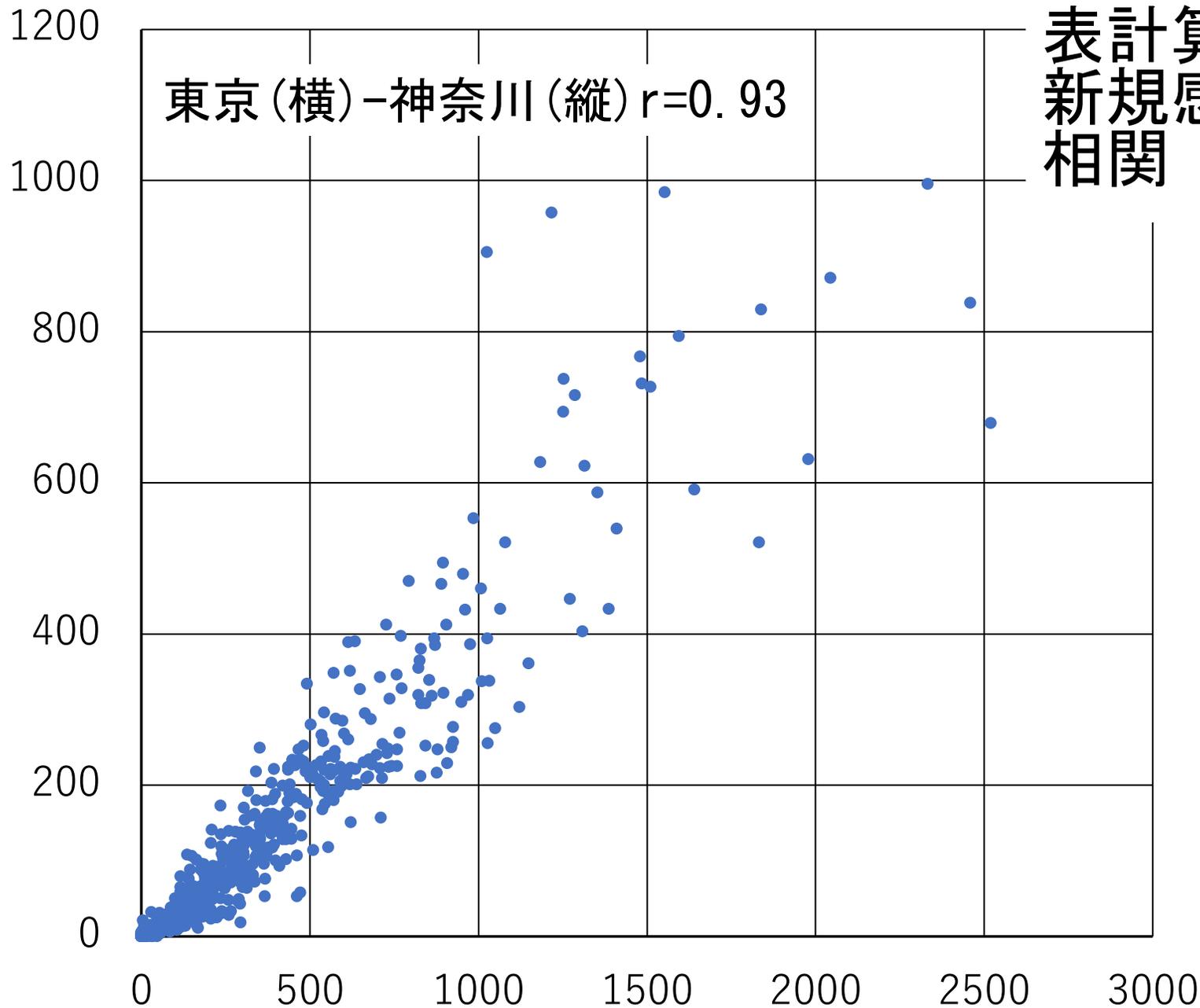
```
import plotly
import plotly.graph_objs as gr
from plotly.offline import init_notebook_mode, iplot
import pandas as pd
from datetime import datetime
df = pd.read_csv('http://strnun.fool.jp/pov-
ray_strnun/covid19tky.csv')
fig = gr.Figure(data=[gr.Candlestick(x=df['DATE'],
    open=df['OP'],
    high=df['MAX'],
    low=df['MIN'],
    close=df['CL'])])

fig.show()
```

The Python logo, consisting of two interlocking snakes, one blue and one yellow, with the word "PYTHON" written in white capital letters across the center.

# PythonによるCovid19東京都新規感染者数チャート分析(今年度)





# 表計算によるCovid19 新規感染者数の東京-神奈川 相関

大阪大会実践報告

データ 2 行 (2 列)  
選択  
作図ツールで 1 分





# PythonによるCovid19新規 感染者数東京-神奈川相関

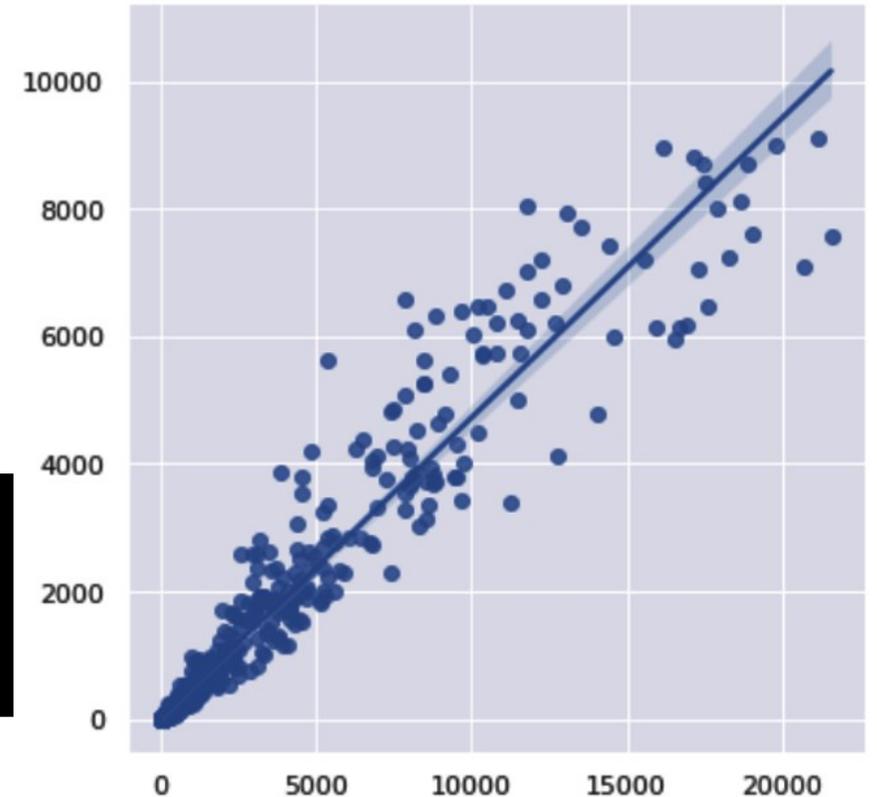
```
import numpy as np
import seaborn as sns
sns.set()
tky = np.array([67, 98, 92, 118, 141, ..., 16878, 16662, 19059])
kan = np.array([19, 25, 31, 21, 27, ..., 6193, 6155, 7603])
correlation = np.corrcoef(tky, kan)
sns.jointplot(tky, kan, kind="reg")
print(correlation[0, 1])
```

0.9672444710977218

↑ 相関係数

散布図 ⇒

今年度展開予定





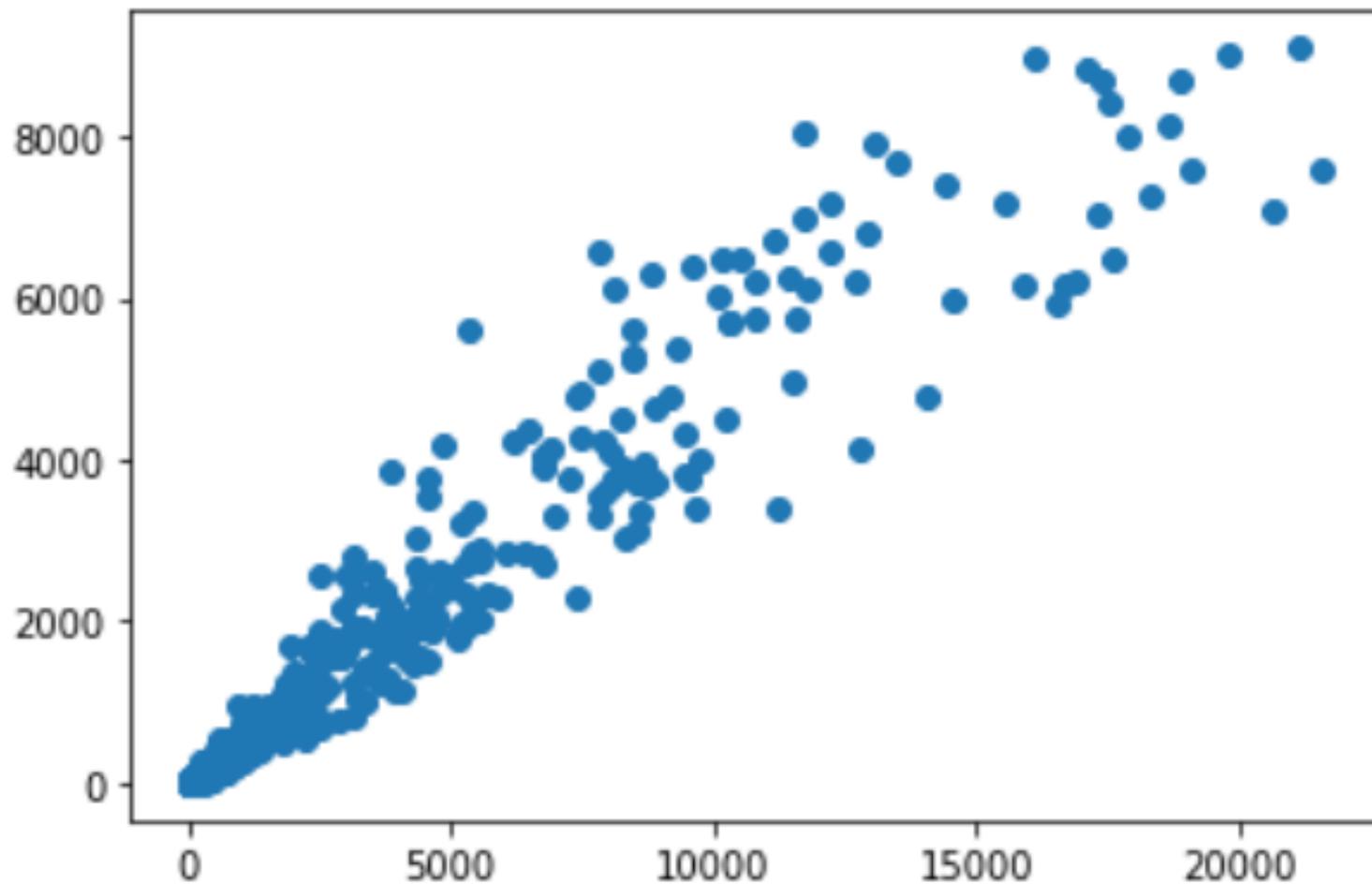
WEBからデータのインポート  
→例は筆者サーバからcsvダウンロード

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
CSV_URL="http://strnun.fool.jp/
pov-ray_strnun/covid19tky-kan.csv"
df = pd.read_csv(CSV_URL,
names=['date', 'tky', 'kan'], header=0);
df.head(836)
```

今年度展開予定

	date	tky	kan
<b>0</b>	2020/4/1	67	19
<b>1</b>	2020/4/2	98	25
<b>2</b>	2020/4/3	92	31
<b>3</b>	2020/4/4	118	21
<b>4</b>	2020/4/5	141	27
...	...	...	...
<b>831</b>	2022/7/11	6231	4230
<b>832</b>	2022/7/12	11511	4991
<b>833</b>	2022/7/13	16878	6193
<b>834</b>	2022/7/14	16662	6155
<b>835</b>	2022/7/15	19059	7603

836 rows × 3 columns



PYTHON

# 省察 1 「表計算 (Excel) による統計分析」

- ・ 旧課程 1 学年生徒対象 → 数学 I で統計未習前提で考察
- ・ 確率分布、統計値概念 → 表計算に利 理由は…
- ①セルに数式／関数を直接入力 → 数学 I 統計単元の数式理解  
教科連携で先行補完
- ②授業展開の速度 → 数式と値入力以外 G U I 操作表  
コーディング時のタイプミスがない
- ③グラフ描画/分析ツール → G U I 操作…表計算に利  
pythonライブラリ活用…ツール
- ・ Excel (P C アプリ) 弱点 → 限定的な学習機会

## 省察 2 「プログラミング (python) による統計分析」

- ・ Python → 作法指導必須
- ・ クラウド環境でシームレスな学習可 → 課題全員提出
- ・ タイピング能力、関心意欲 → 到達度の差
- ・ 教室での授業一斉展開、クラウド活用の個別学習
- ・ 統計学習への進展：数学 I 未習 → 選択科目実施が妥当  
： 数学 I 既習 → 旧課程限定で妥当
- ・ モジュール/ライブラリ活用：エラー対応 → 小論文指導並みの困難
- ・ 昨年度、該当領域をクリアした生徒：270 余名のうち 2 名

## 省察 2 「プログラミング (python) による統計分析」

- ② 今年度 3 学年生徒対象 : **数学 I 既習** ~ Python 基軸を予定  
**入試対策不要**
- ③ 統計、DB : 入試予想問題範囲外 (学習者の立場では無駄)
  - ➔ **情報 I では受験指導の面から妥当性なし**
  - 受験指向 ➔ ソート・サーチなど
  - ➔ 人材教育上のニーズ対応 : 学校設定科目等
- ④ 教員対象の研修 : 機械学習 / AI 活用
  - ➔ 授業担当者として課題 : 指導範囲  
(文法作法からの進展先として研究余地)
  - ➔ **受験対策ニーズ「錬成・実践・発展情報」**

## 省察 3 コーディング指導18年間の課題

コーディング指導のあり方（約15年前の議論）

- ① タイピング…間違え探し
- ② サンプルコード提供～パラメータ改変
- ③ **タイルスク립ティング指導流行**（約10年前）
- ④ 言語指導へ回帰（約5年前 新指対応）

Pythonでは①②折衷とし、

モジュール、ライブラリ活用有無で分ける

- ・ 活用無し⇒①
- ・ 活用有り⇒部分的に②
- ・ CSVデータなどと共に構文を筆者サーバからDL