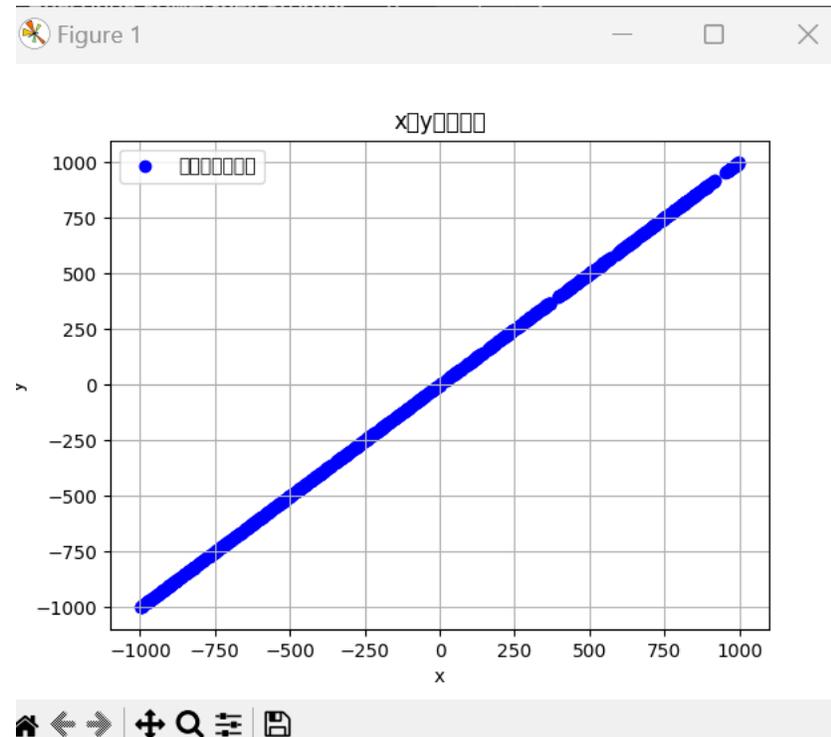


Pythonでグラフを表示しよう



この教材でできること

- Pythonのライブラリについて理解できる
- Pythonでグラフを描く

どんな教材？

PythonでExcelの数値データを獲得して、グラフ化する。そして外れ値を除外して新しくExcelのデータを生成する

目次

- ① コードを書く前の準備をしよう
- ② Excelのデータを獲得してグラフを描こう
- ③ Pythonでグラフの外れ値を除去して新しいファイルを生成しよう
- ④ 新しいファイルからグラフを表示させよう

目次

- ① コードを書く前の準備をしよう
- ② Excelのデータを獲得してグラフを描こう
- ③ Pythonでグラフの外れ値を除去して新しいファイルを生成しよう
- ④ 新しいファイルからグラフを表示させよう

① コードを書く前の準備をしよう

このページでは環境設定や使うファイル、ライブラリなどの準備を行う

① コードを書く前の準備をしよう

まずは環境設定を行おう！

```
(base) PS C:\Users\小西 翔 > cd python  
(base) PS C:\Users\小西 翔\python> cd python9  
(base) PS C:\Users\小西 翔\python\python9> conda create --name graph python=3.8
```



```
(base) PS C:\Users\小西 翔\python\python9> conda activate graph  
(graph) PS C:\Users\小西 翔\python\python9>
```

質問

次は実際にライブラリをインポートしてみよう

今回必要になりそうなことを下に書いているからそれを見てどのライブラリが必要か調べてみよう

- ・ Excelを使うライブラリ
- ・ グラフを描くライブラリ

※考えてから次のスライドに進んでね

答え

「pip install pandas openpyxl matplotlib scikit-learn」

このライブラリをインポートしてみよう

後に必要になりそうなライブラリが出てきたらその都度インポートしていこう！

※必要なライブラリが複数あるときは半角スペースでライブラリを複数指定すると1回でインポートできるよ

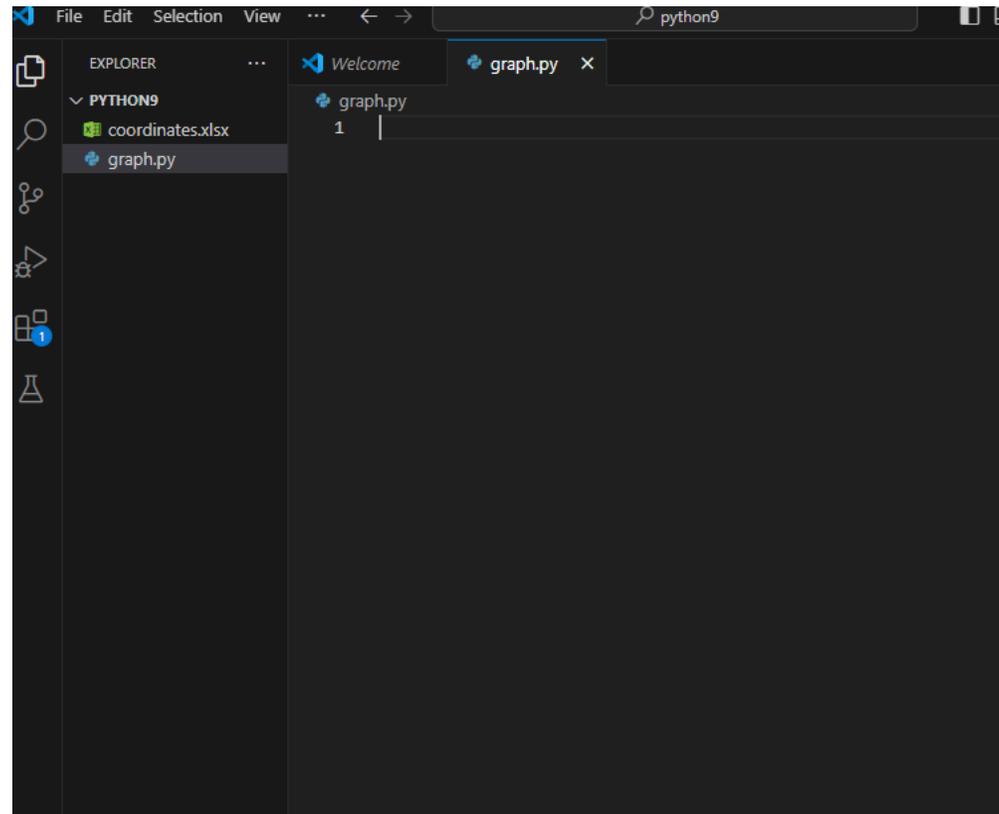
① コードを書く前の準備をしよう

今回使うExcelをダウンロードしよう

[coordinates.xlsx \(live.com\)](#)

① コードを書く前の準備をしよう

ここまで出来たらVscodeを開いてファイルを生成しよう



目次

- ① コードを書く前の準備をしよう
- ② Excelのデータを獲得してグラフを描こう
- ③ Pythonでグラフの外れ値を除去して新しいファイルを生成しよう
- ④ 新しいファイルからグラフを表示させよう

②Excelのデータを獲得してグラフを描こう

このページではExcelのデータ値をグラフにしてみるよ

②Excelのデータを獲得してグラフを描こう

まずは必要なライブラリ等をインポートしよう

```
graphipy > ...  
1  from tkinter import Tk  
2  from tkinter.filedialog import askopenfilename  
3
```

②Excelのデータを獲得してグラフを描こう

Excelファイルを選択して値を返すような関数を作ろう

```
3
4 # ファイルを選択する関数
5 def select_file():
6     Tk().withdraw() # GUIを表示しないように設定
7     filename = askopenfilename(title="Excelファイルを選択してください", filetypes=[("Excelファイル", "*.xlsx *.xls")])
8     return filename
```

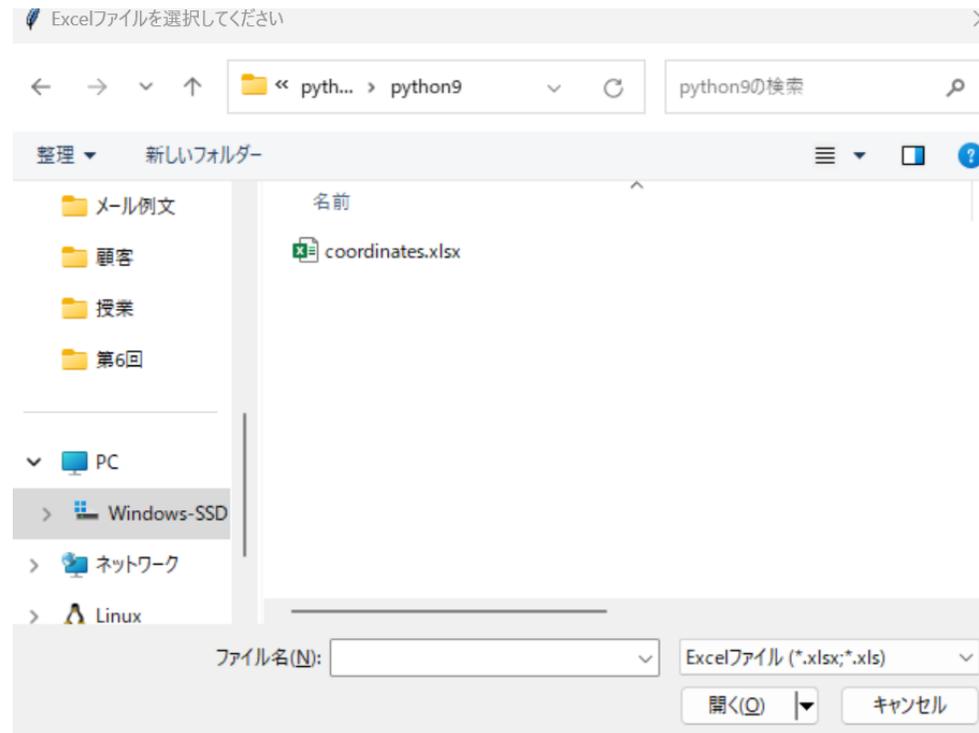
②Excelのデータを獲得してグラフを描こう

次にメイン関数を作ってファイルを選択できるようにしよう

```
9
10 def main():
11     # Excelファイルを選択
12     file_path = select_file()
13
14 if __name__ == "__main__":
15     main()
```

②Excelのデータを獲得してグラフを描こう

ここまでで実行したらダイアログが開くようになったかな？



②Excelのデータを獲得してグラフを描こう

ExcelのデータをPythonで使えるように変更するためのライブラリをインポートしよう

```
1 from tkinter import Tk
2 from tkinter.filedialog
3 import pandas as pd
4
```

②Excelのデータを獲得してグラフを描こう

ExcelのデータをPythonで使えるように変更しよう

```
11 def main():  
12     # Excelファイルを選択  
13     file_path = select_file()  
14  
15     # Excelファイルのデータを読み込む  
16     df = pd.read_excel(file_path)  
17
```

②Excelのデータを獲得してグラフを描こう

グラフを描けるようなライブラリをインポートしよう

```
2 from tkinter.filedialog import asko  
3 import pandas as pd  
4 import matplotlib.pyplot as plt
```

②Excelのデータを獲得してグラフを描こう

グラフを表示する関数を作ろう

今回は引数を3つ設定しているよ

```
11
12 # データをプロットする関数
13 def plot_data(df, x_column, y_column):
14     plt.scatter(df[x_column], df[y_column], label="データポイント", color="blue")
15     plt.xlabel(x_column)
16     plt.ylabel(y_column)
17     plt.title(f'{x_column}と{y_column}の散布図')
18     plt.legend()
19     plt.grid(True)
20     plt.show()
```

説明

1. `plt.scatter(df[x_column], df[y_column], label="データポイント", color="blue")`

- `plt.scatter()`: 散布図を作成
- `df[x_column]`: データフレーム `df` の `x` 軸に対応する列のデータを抽出
- `label="データポイント"`: 凡例に表示されるラベルを指定
- `color="blue"`: プロットされた点の色を青色に指定

説明

2. plt.xlabel(x_column)

- ・ plt.xlabel(): x軸のラベル（名前）を設定する関数

3. plt.title(f'{x_column}と{y_column}の散布図')

- ・ plt.title(): プロット全体のタイトルを設定する関数

説明

4. plt.legend()

- ・ `plt.legend()`: プロットに凡例を追加する関数。凡例は、プロットに含まれるデータのラベル（名前）を表示するために使用される。このコードでは、`label="データポイント"`で指定したラベルが凡例に表示される。

6. plt.grid(True)

- ・ `plt.grid(True)`: グリッド線（縦線と横線）をプロットに追加する関数

7. plt.show()

- ・ `plt.show()`: 作成したプロットを画面に表示する関数

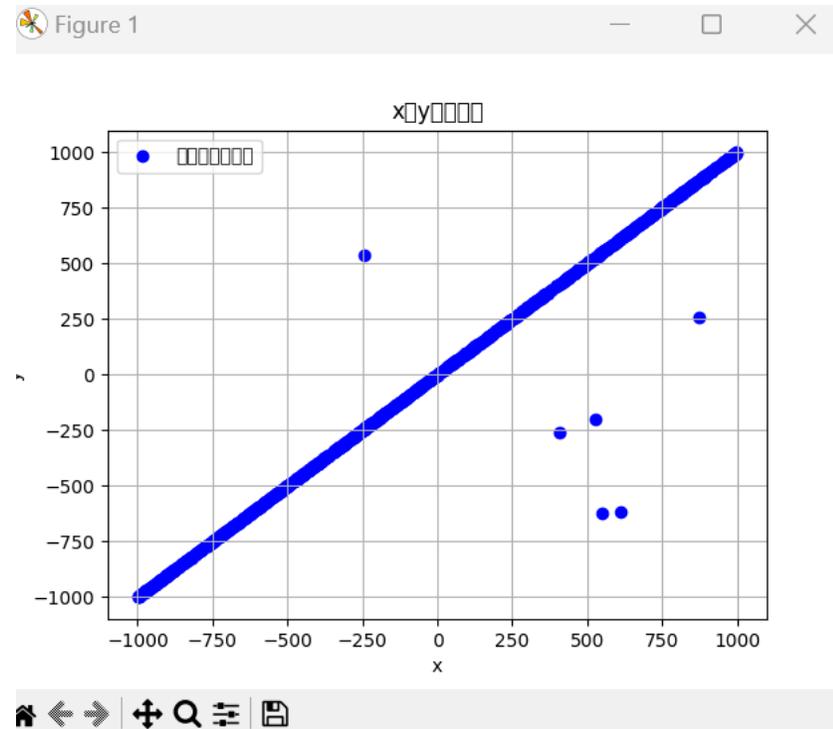
②Excelのデータを獲得してグラフを描こう

最後にメイン関数内でグラフの関数を呼び出してみよう

```
25
26     # Excelファイルのデータを読み込む
27     df = pd.read_excel(file_path)
28
29     # データをプロット
30     plot_data(df, 'x', 'y')
```

②Excelのデータを獲得してグラフを描こう

コードを実行してみてグラフが表示されるようになったかな？



目次

- ① コードを書く前の準備をしよう
- ② Excelのデータを獲得してグラフを描こう
- ③ Pythonでグラフの外れ値を除去して新しい
ファイルを生成しよう
- ④ 新しいファイルからグラフを表示させよう

③Pythonでグラフの外れ値を除去して新しいファイルを生成しよう

このページではExcelの外れ値をPythonで調べて除去し、新しくExcelのファイルとして生成してみよう

③Pythonでグラフの外れ値を除去して新しいファイルを生成しよう

dfのx,yの値を配列Xに代入しよう

```
29 # データをプロット
30 plot_data(df, 'x', 'y')
31
32 # x, y の値を取得
33 X = df[['x', 'y']].values
34
```

③Pythonでグラフの外れ値を除去して新しいファイルを生成しよう

外れ値を除去する関数を新しく作ろう！

```
20     plt.grid(True)
21     plt.show()
22
23     # 外れ値を調べ、除去する関数
24     def out_of_point(file_path, df, point):
25
26     def main():
27         # Excelファイルを選択
28         file_path = select_file()
29
```

③Pythonでグラフの外れ値を除去して新しいファイルを生成しよう

ここで新しいライブラリをインポートしよう

```
1  from tkinter import Tk
2  from tkinter.filedialog import askopenfilename
3  import pandas as pd
4  import matplotlib.pyplot as plt
5  from sklearn.cluster import DBSCAN
```

説明

```
from sklearn.cluster import DBSCAN
```

ノイズ除去を行うための機械学習のクラスタリング手法の一種

- ・データポイントの密度を基準にクラスタを形成します。密度が高い領域にデータポイントが集まっている場合、それらのポイントをクラスタとみなし、密度が低い場所にあるポイントはノイズ（異常値）として扱います。

③Pythonでグラフの外れ値を除去して新しいファイルを生成しよう

関数のコードを書いていこう！

まずは外れ値の検出をしよう

```
22
23 # 外れ値を調べ、除去する関数
24 def out_of_point(file_path, df, point):
25     # DBSCANで外れ値を検出
26     dbscan = DBSCAN(eps=10, min_samples=5)
27     labels = dbscan.fit_predict(point)
28     outliers = df[labels == -1]
```

③Pythonでグラフの外れ値を除去して新しいファイルを生成しよう

外れ値がある場合は外れ値の情報を出力するようにしよう

```
29
30 # 外れ値がある場合、それらの行を報告し、データから除外する
31 if not outliers.empty:
32     print("以下の行に外れ値があります。これらを除外します。")
33     for index in outliers.index:
34         print(f"{index + 2}行目: x={df.at[index, 'x']}, y={df.at[index, 'y']}")
35
36
```

③Pythonでグラフの外れ値を除去して新しいファイルを生成しよう

ファイルの除去が終わったら新しくファイルを保存しよう

```
35
36     # 外れ値を除外
37     df_cleaned = df[labels != -1]
38
39     # 新しいExcelファイルとして保存
40     new_file_path = file_path.replace('.xlsx', '_cleaned.xlsx')
41     df_cleaned.to_excel(new_file_path, index=False)
42     print(f"外れ値を除外した新しいファイルを保存しました: {new_file_path}")
43
```

③Pythonでグラフの外れ値を除去して新しいファイルを生成しよう

外れ値がなかったらelse文で外れ値がないことを表示させよう

```
30     # 外れ値がある場合、それらの行を報告し、データから除外
31     if not outliers.empty:
32         print("以下の行に外れ値があります。これらを除外します。")
33         for index in outliers.index:
34             print(f"{index + 2}行目: x={df.at[index, 'x']} y={df.at[index, 'y']}")
35
36         # 外れ値を除外
37         df_cleaned = df[labels != -1]
38
39         # 新しいExcelファイルとして保存
40         new_file_path = df.replace('.xlsx', '_cleaned.xlsx')
41         df_cleaned.to_excel(new_file_path, index=False)
42         print(f"外れ値を除外した新しいファイルを保存しました。")
43
44     else:
45         print("外れ値は見つかりませんでした。")
```

③Pythonでグラフの外れ値を除去して新しいファイルを生成しよう

関数が完成したらメイン関数で外れ値の除去の関数を呼び出そう！

```
53
54     # データをプロット
55     plot_data(df, 'x', 'y')
56
57     # x, y の値を取得
58     X = df[['x', 'y']].values
59
60     # データの除去
61     out_of_point(file_path, df, X)
62
```

③Pythonでグラフの外れ値を除去して新しいファイルを生成しよう

実行してみたらExcelのファイルが1つ増えることが確認できるかな？

 coordinates.xlsx	2024/08/25 12:42	Microsoft Excel ワーク...	26 KB
 graph.py	2024/09/07 14:48	Python ソース ファイル	3 KB
 sample.py	2024/09/04 0:25	Python ソース ファイル	2 KB



 coordinates.xlsx	2024/08/25 12:42	Microsoft Excel ワーク...	26 KB
 graph.py	2024/09/07 14:56	Python ソース ファイル	3 KB
 sample.py	2024/09/04 0:25	Python ソース ファイル	2 KB
 coordinates_cleaned.xlsx	2024/09/07 14:55	Microsoft Excel ワーク...	24 KB

質問

ターミナルを見て、何か気付くことはないかな？

※考えてから次のスライドに進んでね

```
C:\Users\小西 翔\.conda\envs\graph\lib\tkinter\__init__.py:814: UserWarning:
ng from current font.
    func(*args)
以下の行に外れ値があります。これらを除外します。
41行目: x=405.9469494315304, y=-261.2976850084441
88行目: x=585.3170307075493, y=585.3170307075493
114行目: x=875.002474886684, y=255.3668075289297
120行目: x=948.0629777778233, y=948.0629777778233
131行目: x=943.6248957207226, y=943.6248957207226
174行目: x=-243.05226274143, y=537.110123529993
226行目: x=931.8882472301559, y=931.8882472301559
310行目: x=611.4291197852149, y=-615.4567209395285
430行目: x=253.8992765229275, y=253.8992765229275
503行目: x=11.69958928568656, y=11.69958928568656
510行目: x=727.7634084651977, y=727.7634084651977
555行目: x=-984.0851249064857, y=-984.0851249064857
574行目: x=382.2595962034029, y=382.2595962034029
585行目: x=584.2758997558665, y=584.2758997558665
590行目: x=151.9756897911984, y=151.9756897911984
600行目: x=527.5381858724359, y=-201.0612508883492
666行目: x=548.8519248357686, y=-621.0486846780229
699行目: x=947.8318127494847, y=947.8318127494847
812行目: x=9.893291270290433, y=9.893291270290433
832行目: x=661.4528494541614, y=661.4528494541614
834行目: x=375.8838089754236, y=375.8838089754236
854行目: x=580.0684681837477, y=580.0684681837477
892行目: x=934.1572889113052, y=934.1572889113052
929行目: x=928.6769295903621, y=928.6769295903621
943行目: x=662.5985803980616, y=662.5985803980616
```

答え

合っているはずの値も外れ値として除去しているね！

```
C:\Users\小西 翔\.conda\envs\graph\lib\tkinter\__init__.py:814: UserWarning: Using font.
func(*args)
以下の行に外れ値があります。これらを除外します。
41行目: x=405.9469494315304, y=-261.2976850084441
88行目: x=585.3170307075493, y=585.3170307075493
114行目: x=875.002474886684, y=255.3668075289297
120行目: x=948.0629777778233, y=948.0629777778233
131行目: x=943.6248957207226, y=943.6248957207226
174行目: x=-243.05226274143, y=537.110123529993
226行目: x=931.8882472301559, y=931.8882472301559
310行目: x=611.4291197852149, y=-615.4567209395285
430行目: x=253.8992765229275, y=253.8992765229275
503行目: x=11.69958928568656, y=11.69958928568656
510行目: x=727.7634084651977, y=727.7634084651977
555行目: x=-984.0851249064857, y=-984.0851249064857
574行目: x=382.2595962034029, y=382.2595962034029
585行目: x=584.2758997558665, y=584.2758997558665
590行目: x=151.9756897911984, y=151.9756897911984
600行目: x=527.5381858724359, y=-201.0612508883492
666行目: x=540.0519240357606, y=-621.0406046700229
699行目: x=947.8318127494847, y=947.8318127494847
812行目: x=9.893291270290433, y=9.893291270290433
832行目: x=661.4528494541614, y=661.4528494541614
834行目: x=375.8838089754236, y=375.8838089754236
854行目: x=580.0684681837477, y=580.0684681837477
892行目: x=934.1572889113052, y=934.1572889113052
929行目: x=928.6769295903621, y=928.6769295903621
943行目: x=662.5985803980616, y=662.5985803980616
```

答え

理由として以下のことが考えられるよ！

- ・ 機械学習を行うためのモデルが少ない
- ・ 外れ値として判断する際の判断基準に間違いが生じる

今回はライブラリで機械学習をパソコンでさせているから間違いが起きるけど、ライブラリに頼らず自分で機械学習をさせるとうまくいく可能性があるよ！

目次

- ① コードを書く前の準備をしよう
- ② Excelのデータを獲得してグラフを描こう
- ③ Pythonでグラフの外れ値を除去して新しいファイルを生成しよう
- ④ 新しいファイルからグラフを表示させよう

④新しいファイルからグラフを表示させよう

このページでは生成したファイルがうまくいっているか再度グラフを表示してみるよ

④新しいファイルからグラフを表示させよう

外れ値除去の関数を少し変更しよう！

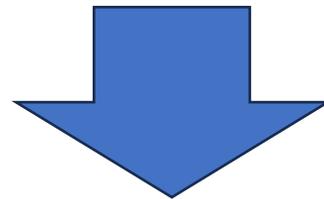
if文のところに返り値を付け加えよう

```
39     # 新しいExcelファイルとして保存
40     new_file_path = file_path.replace('.xlsx',
41     df_cleaned.to_excel(new_file_path, index=False)
42     print(f"外れ値を除外した新しいファイルを保存しました。")
43
44     return new_file_path
45
46 else:
47     print("外れ値は見つかりませんでした。")
48
49     return 0
```

④新しいファイルからグラフを表示させよう

メイン関数で戻り値を変数で受け取るように変更しよう

```
# データの除去  
out_of_point(file_path, df, X)
```



```
63  
64 # データの除去  
65 new_file_path = out_of_point(file_path, df, X)  
66
```

④新しいファイルからグラフを表示させよう

new_file_pathが0ではない場合(何かしらのファイルパスが入っている場合)にグラフの表示を行おう！

```
66
67     if new_file_path !=0:
68         new_df =pd.read_excel(new_file_path)
69         plot_data(new_df, 'x', 'y')
70
```


お疲れさまでした

テキストは終了です。
あとは自分なりにアレンジを付け加えていこう！