

第5回 pandasライブラリで遊ぼう

- pandasライブラリ powerful Python data analysis toolkit
 - HP ⇒ <https://pandas.pydata.org/>
 - ドキュメント <https://pandas.pydata.org/pandas-docs/stable/>
 - チュートリアル 例「10 Minutes to pandas」
https://pandas.pydata.org/pandas-docs/stable/getting_started/tutorials.html
 - 何ができるか？ data analysis?
 - Excelと同じような表形式データが使える
 - 行単位・列単位の計算ができる
 - 行(列)の和、行(列)の平均、行(列)の分散、などなど
 - 集計機能（たとえば頻度を数える）
 - グラフが簡単に描ける
 - Excelやcsvとのやりとりが簡単にできる
- など

pandasのデータ

- DataFrame ~ Excelの表

月	月間アイスクリーム支出	平均気温
1	464	10.6
2	397	12.2
3	493	14.9
4	617	20.3
	以下略	

- Series ~ 1次元データ (ベクトル、リスト)

464	397	493	617
-----	-----	-----	-----

DataFrameを作るには

- データをリストとして与える

1	464	10.6
2	397	12.2
3	493	14.9

[[1, 464, 10.6], [2, 397, 12.2], [3, 493, 14.9]]



```
df=pd.DataFrame( [[1, 464, 10.6], [2, 397, 12.2], [3, 493, 14.9]] )
```



```
df=pd.DataFrame( [[1, 464, 10.6], [2, 397, 12.2], [3, 493, 14.9]],  
                 columns=['月', '月間アイスクリーム支出', '平均気温'])
```

- ファイルから読む CSVファイル、Excelファイル

```
df = pd.read_csv('icecream.csv', index_col=0)
```

```
df = pd.read_excel('icecream.xlsx')
```

CSVファイル、Excelファイル

- CSV comma-separated values コンマ区切り形式

```
月,月間アイスクリーム支出,平均気温  
1,464,10.6  
2,397,12.2  
3,493,14.9
```

icecream.csv

単純なのでいろいろなプログラム間の受け渡しに使われる

- Excel icecream.xlsx

	A	B	C	D
1	月	月間アイスクリーム支出	平均気温	
2	1	464	10.6	
3	2	397	12.2	
4	3	493	14.9	
5				

CSVを試してみよう

- CSVファイルをダウンロード
 - ブラウザで <https://bit.ly/32wqXQR> をアクセスし
icecream.csv を USBの"notebook"フォルダ中にセーブ
 - テキストエディタを用いて icecream.csv を開いてみる
- pandasのread_csvを使ってdfに読込む

```
df = pd.read_csv('icecream.csv', index_col=0)
print(df)
```

XLSXを試してみよう

- Excelのxlsxファイルをダウンロード
 - ブラウザで <https://bit.ly/33RTj8n> をアクセスし
icecream.xlsx を USBの"notebook"フォルダ中にセーブ
 - テキストエディタを用いて icecream.xlsx を開いてみる
⇒ 今度は(テキストエディタでは)読めないはず
- pandasのread_xlsxを使ってdf2に読込む

```
df2 = pd.read_xlsx('icecream.xlsx')
print(df2)
```

(前回の最後) pandasとmatplotlibを使う

icecream.xlsx データを使ってみる

```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
df = pd.DataFrame(
    [[165.0, 65.0], [170.0, 68.0 ], [167.5, 67.2]],
    columns=['身長', '体重'],
    index=['田中', '山田', '佐藤'])
print(df)
df.plot.scatter(x='身長', y='体重') ← 散布図を描く
plt.title('身長と体重')
plt.xlabel('身長')
plt.ylabel('体重')
plt.show()
```

← 画面に表示させる
おまじない

置換える
このころを

} 適宜修正
この辺も

試してみよう

irisデータのXLSXを試してみよう

- Excelのxlsxファイルをダウンロード
 - ブラウザで <https://bit.ly/2MzEDF6> をアクセスし
iris.xlsx を USBの"notebook"フォルダ中にセーブ
- pandasのread_xlsxを使ってdfに読込む

```
df = pd.read_xlsx('iris.xlsx')  
print(df.head())
```

← 先頭の5行だけprint

(前回の最後) pandasとmatplotlibを使う

iris.csv データを使ってみる

```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
df = pd.DataFrame(
    [[165.0, 65.0], [170.0, 68.0 ], [167.5, 67.2]],
    columns=['身長', '体重'],
    index=['田中', '山田', '佐藤'])
# print(df)
df.plot.scatter(x='身長', y='体重') ← 散布図を描く
plt.title('身長と体重')
plt.xlabel('身長')
plt.ylabel('体重')
plt.show()
```

← 画面に表示させる
おまじない

← 置き換える
このころを

} 適宜修正
この辺も

TAチェック

おまけ 時間があれば試してみよう①

irisの種類が3つあるので、それを区別する

```
%matplotlib inline
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
データを讀込んでdfに代入する (前頁と同じ)
```

```
.....
```

```
df0 = df[df.target==0]
```

```
df1 = df[df.target==1]
```

```
df2 = df[df.target==2]
```

```
ax = df0.plot.scatter(x='petal length', y='petal width', c='red')
```

```
df1.plot.scatter(x='petal length', y='petal width', c='blue', ax=ax)
```

```
df2.plot.scatter(x='petal length', y='petal width', c='green', ax=ax)
```

```
plt.title('iris')
```

```
plt.xlabel('花弁の長さ')
```

```
plt.ylabel('花弁の幅')
```

```
plt.show()
```

← 画面に表示させる
おまじない

}

試してみよう

おまけ 時間があれば試してみよう②

irisの種類が3つあるので、それを区別する

下記の部分に、グラフ上の点の形と、凡例用の説明文字列を追加する

```
ax = df0.plot.scatter(x='petal length', y='petal width', c='red',
                      marker='o', label='Setosa')
df1.plot.scatter(x='petal length', y='petal width', c='blue', ax=ax,
                 marker='x', label='Versicolour')
df2.plot.scatter(x='petal length', y='petal width', c='green', ax=ax,
                 marker='^', label='Virginica')
plt.legend()
```

後は同じ



試してみよう

- `c='red'` 色colorがred/blue/green/...
- `marker='o'` マーカーの形状がo/x/^/...
- `label='...'` 凡例legendの名前付けの文字列