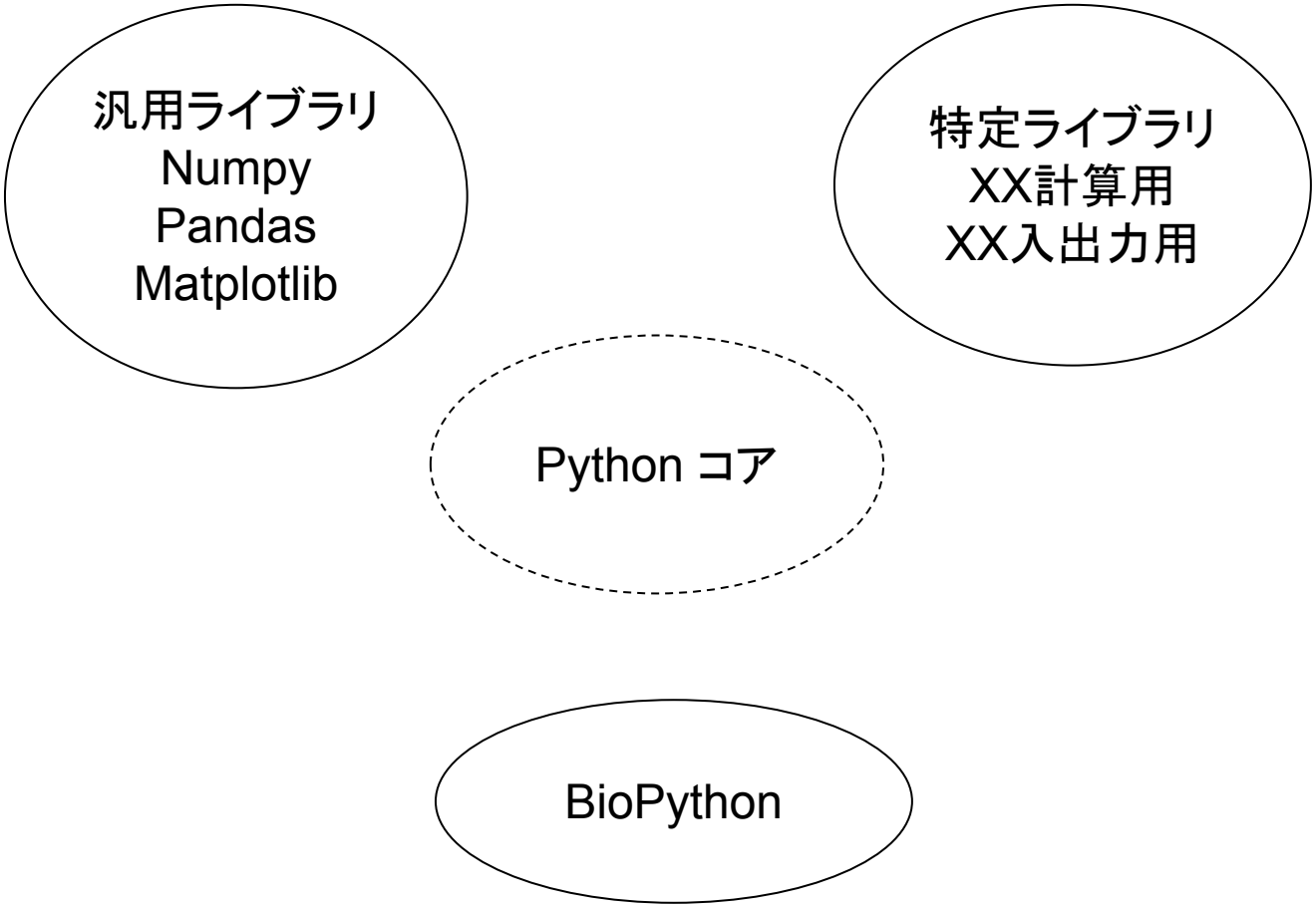
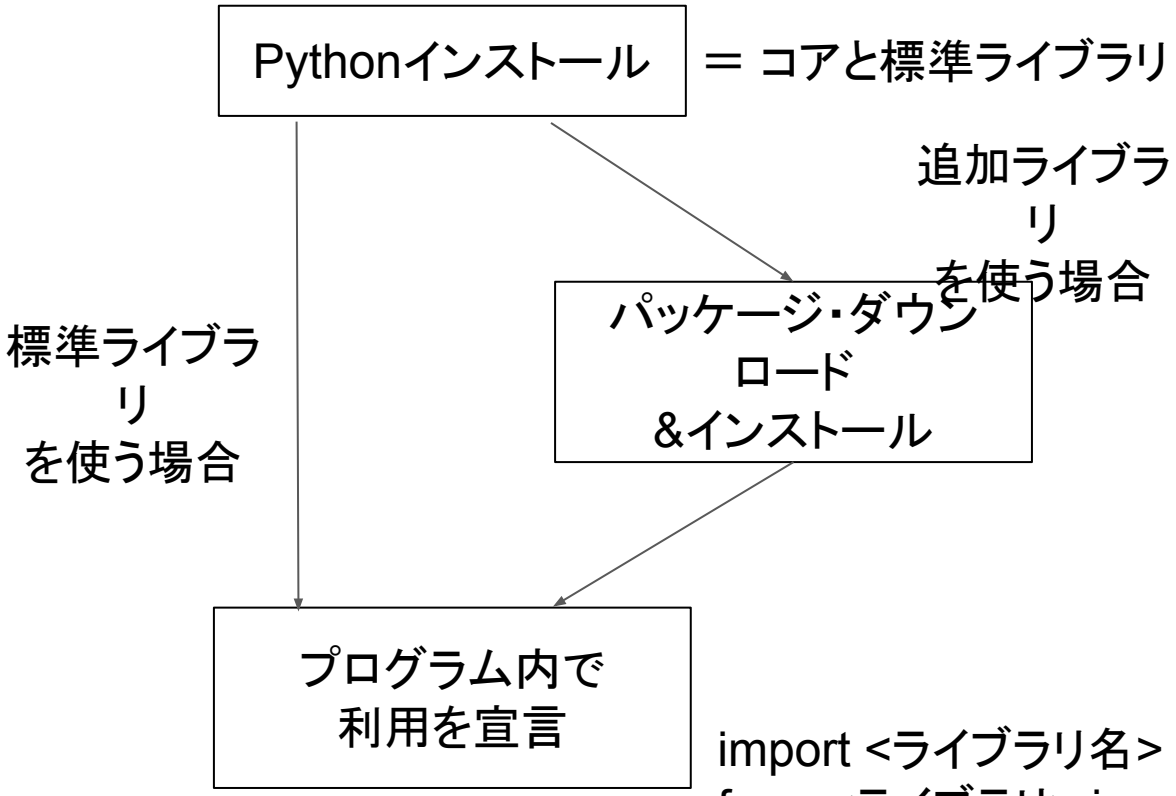
pandasを使ってみる

2019-08-19　　山内長承

Pythonを使ってみる



ライブラリを使うお作法



pandasを使ってみる

	pos	freq
0	2	0.1
1	7	0.2
2	13	0.4

- https://pandas.pydata.org/pandas-docs/stable/
- （主に）Excelの表のような形でデータを扱う
  - 「データ処理」向き
  - DataFrame = 行・列
  - Series = 行だけ（1次元）
  - 要素ごと、行単位、列単位の操作が用意されている
    - df['pos'] ⇒'pos'列の切出し　・df*2 ⇒要素ごとに2倍
    - df.sum() ⇒列ごとの総和　　・df.sum(axis=1) ⇒行ごとの総和
  - 裸のPythonやnumpyなどと混ぜて使える

試してみる

- pandasが未だダウンロードされていなければ、ダウンロードする

```

```

- Jupyter Notebookを起動する
- pandasの利用を宣言する（import、別名 pd を付ける）

```

```

- データフレームを作る～元データはリスト形式で与える
  - ２次元のリスト：　[[2,0.1],　[7,　0.2],　[13,　0.4]]
  - データフレームを作る（インスタンス化）

    pd.DataFrame(<二次元のリスト>)

    pd.DataFrame(<二次元のリスト> ,　columns = <列の名前のリスト>)

```
                                                , columns = ['pos', 'freq'] )
```

	pos	freq
0	2	0.1
1	7	0.2
2	13	0.4

試してみる（続き）

	pos	freq
0	2	0.1
1	7	0.2
2	13	0.4

- データフレームを作る（続き）
  - pd.DataFrame(<二次元のリスト> , columns = <列の名前のリスト>)
  - 変数 df に代入する形にする

    df = pd.DataFrame(… …)

- データフレームの内容をprintで確認する
  - print(df)　　画面に表示する

```
   pos  freq
0    2   0.1
1    7   0.2
2   13   0.4
```

- 全体を run する

演算を試してみる

- 列の切出し：　df['pos']、　複数列：　df[['pos','freq']]

	pos	freq
0	2	0.1
1	7	0.2
2	13	0.4

- 行の切出し：　２種類が考えられる
  - df[0:1]　指定された範囲［0:1］を切出す ⇒ １行のデータフレーム
  - df.loc[0]　指定された位置の行を切り出す ⇒ Series(1次元vector)
- 定数を掛ける：　df*2 ⇒

	pos	freq
0	4	0.2
1	14	0.4
2	26	0.8

- 列ごと・行ごとの計算：　たとえば sum
  - df.sum() ⇒

pos	22.0
freq	0.7

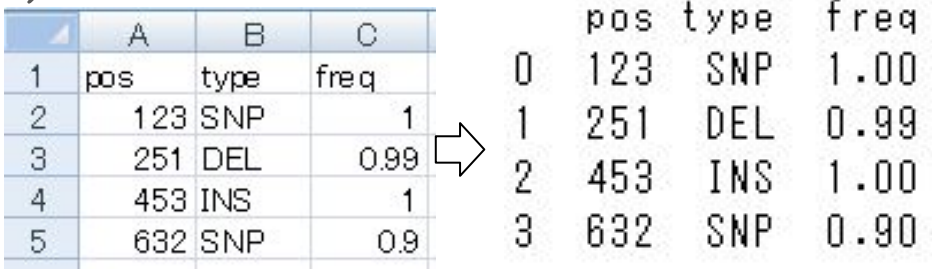
  - df.sum(axis=1) ⇒

0	2.1
1	7.2
2	13.4

その他⇒https://pandas.pydata.org/pandas-docs/stable/reference/frame.html#computations-descriptive-stats

（脱線）pandasでExcelやCSVからの読込み・書出し

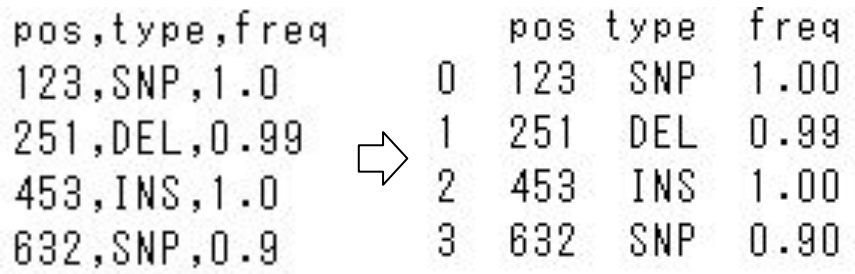
- df = pd.read_excel('ファイル名')



	A	B	C
1	pos	type	freq
2	123	SNP	1
3	251	DEL	0.99
4	453	INS	1
5	632	SNP	0.9

	pos	type	freq
0	123	SNP	1.00
1	251	DEL	0.99
2	453	INS	1.00
3	632	SNP	0.90

- df = pd.read_csv('ファイル名')



```
pos,type,freq
123,SNP,1.0
251,DEL,0.99
453,INS,1.0
632,SNP,0.9
```

	pos	type	freq
0	123	SNP	1.00
1	251	DEL	0.99
2	453	INS	1.00
3	632	SNP	0.90

- df.to_excel('ファイル名')
- df.to_csv('ファイル名')

DataFrameの切出し

- SNPだけを切出す

```
dfx = df[ df['type']=='SNP' ]
print(dfx)
```

```
   pos type  freq
0  123  SNP  1.00
1  251  DEL  0.99
2  453  INS  1.00
3  632  SNP  0.90
```

⇨

```
   pos type  freq
0  123  SNP   1.0
3  632  SNP   0.9
```

- INSとDELを切出す

```
dfx = df[df['type'].isin('INS', DEL')]' ]
print(dfx)
```

```
   pos type  freq
0  123  SNP  1.00
1  251  DEL  0.99
2  453  INS  1.00
3  632  SNP  0.90
```

⇨

```
   pos type  freq
1  251  DEL  0.99
2  453  INS  1.00
```

- freq<=0.95 を切出す

```
dfx = df[df['freq']<=0.95]
print(dfx)
dfy = df[(df['freq']>=0.95)&
         df['pos']<=400)]
```

```
   pos type  freq
0  123  SNP  1.00
1  251  DEL  0.99
2  453  INS  1.00
3  632  SNP  0.90
```

⇨

```
   pos type  freq
3  632  SNP   0.9
```

おまけ

- Webブラウザで https://pepper.is.sci.toho-u.ac.jp
  ⇒　授業のページ「岸本研勉強会」
  ⇒　AP012030_CDS.xlsxをダウンロード
- df = read_excel('AP012030_CDS.xlsx') で読込み
- prin(df)で中を見たいが、全部表示すると長すぎるので、
  print(df.head()) としてみる。　先頭の5行を表示
  もしくはprint(df.head(20))　なら20行表示
- 必要に応じて、行を切り出して表示する。たとえば、
  print(df[['pos', 'len','gene', 'product']].head(10))

	pos	len	gene	product
0	189	66	['thrL']	['thr operon leader peptide']
1	337	2463	['thrA']	['bifunctional aspartokinase I/homeserine dehy...
2	2801	933	['thrB']	['homoserine kinase']
3	3734	1287	['thrC']	['threonine synthase']
4	5234	297	['yaaX']	['predicted protein']