PG08: DataFrame の操作

DataFrame は表計算ソフトで使われるスプレッドシートに似た形で表示されるので視覚的にわかり安く便利である。 スプレッドシートと同様に、DataFrame の各要素に様々な操作を行うことができる。 ここでは、最も基本的な事柄を説明しておこう。

8.1. データの読み込み (RioAthletes2016.csv)

かなり大きなデータである。これを例にして DataFrame の基本操作を示す。

In [1]:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

In [2]:

```
pd.read_csv('F:/2022_数理統計学概論/StatData/RioAthletes2016.csv')
```

Out[2]:

	This dataset includes the official statistics on the 11,538 athletes (6,333 men and 5,205 women) and 306 events at the 2016 Olympic Games in Rio de Janeiro.\nhttps://github.com/flother/rio2016	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	id	name	nationality	sex	date_of_birth
1	736041664	A Jesus Garcia	ESP	male	1969/10/17
2	532037425	A Lam Shin	KOR	female	1986/9/23
3	435962603	Aaron Brown	CAN	male	1992/5/27
4	521041435	Aaron Cook	MDA	male	1991/1/2
...	...	...	...	...	..
11534	265605954	Zurian Hechavarria	CUB	female	1995/8/10
11535	214461847	Zuzana Hejnova	CZE	female	1986/12/19
11536	88361042	di Xiao	CHN	male	1991/5/14
11537	900065925	le Quoc Toan Tran	VIE	male	1989/4/5
11538	711404576	le Roux Hamman	RSA	male	1992/1/6

11539 rows × 12 columns

まず、csv ファイルの1行目は表題であるから、これを削除して読み込むとよい。

In [3]:

```
Data = pd.read_csv('F:/2022_数理統計学概論/StatData/RioAthletes2016.csv', skiprows=1)
Data
```

Out[3]:

	id	name	nationality	sex	date_of_birth	height	weight	sport	gold	silver	b
0	736041664	A Jesus Garcia	ESP	male	1969/10/17	1.72	64.0	athletics	0	0	
1	532037425	A Lam Shin	KOR	female	1986/9/23	1.68	56.0	fencing	0	0	
2	435962603	Aaron Brown	CAN	male	1992/5/27	1.98	79.0	athletics	0	0	
3	521041435	Aaron Cook	MDA	male	1991/1/2	1.83	80.0	taekwondo	0	0	
4	33922579	Aaron Gate	NZL	male	1990/11/26	1.81	71.0	cycling	0	0	
...	...	...	...	...	...	...	...	...	...	...	...
11533	265605954	Zurian Hechavarria	CUB	female	1995/8/10	1.64	58.0	athletics	0	0	
		Zuzana									

8.2. 要素へのアクセス

個別の要素へのアクセス

iat[i,j] メソッド: 絶対番地（i行j列）を使う。i, j ともに0から始まる。

at[i,'カラム名'] メソッド：インデックス番号 i とカラム名を使う。

In [4]:

```
Data.at[2, 'sport']
```

Out[4]:

```
'athletics'
```

In [5]:

```
Data.iat[2, 7]
```

Out[5]:

```
'athletics'
```

個別要素の修正（上書き）は代入すればよい。

In [6]:

```
Data.at[2, 'sport'] = 'baseball'
Data.head()
```

Out[6]:

	id	name	nationality	sex	date_of_birth	height	weight	sport	gold	silver
0	736041664	A Jesus Garcia	ESP	male	1969/10/17	1.72	64.0	athletics	0	0
1	532037425	A Lam Shin	KOR	female	1986/9/23	1.68	56.0	fencing	0	0
2	435962603	Aaron Brown	CAN	male	1992/5/27	1.98	79.0	baseball	0	0
3	521041435	Aaron Cook	MDA	male	1991/1/2	1.83	80.0	taekwondo	0	0
4	33922579	Aaron Gate	NZL	male	1990/11/26	1.81	71.0	cycling	0	0

In [7]:

```
# 元に戻しておく
Data.at[2, 'sport'] = 'athletics'
Data.head()
```

Out[7]:

	id	name	nationality	sex	date_of_birth	height	weight	sport	gold	silver
0	736041664	A Jesus Garcia	ESP	male	1969/10/17	1.72	64.0	athletics	0	0
1	532037425	A Lam Shin	KOR	female	1986/9/23	1.68	56.0	fencing	0	0
2	435962603	Aaron Brown	CAN	male	1992/5/27	1.98	79.0	athletics	0	0
3	521041435	Aaron Cook	MDA	male	1991/1/2	1.83	80.0	taekwondo	0	0
4	33922579	Aaron Gate	NZL	male	1990/11/26	1.81	71.0	cycling	0	0

In [ ]:

```

```

8.3. 行へのアクセス

スライス記法を用いる。

Data[2:3] インデックス2から3の直前まで（よってインデックス2のみ）

Data[2:] インデックス2以降全部

loc[:10] 初めからインデックス10の直前まで

In [8]:

```
Data[2:3]
```

Out[8]:

	id	name	nationality	sex	date_of_birth	height	weight	sport	gold	silver	br
2	435962603	Aaron Brown	CAN	male	1992/5/27	1.98	79.0	athletics	0	0	

In [9]:

```
Data[2:]
```

Out[9]:

	id	name	nationality	sex	date_of_birth	height	weight	sport	gc
2	435962603	Aaron Brown	CAN	male	1992/5/27	1.98	79.0	athletics	
3	521041435	Aaron Cook	MDA	male	1991/1/2	1.83	80.0	taekwondo	
4	33922579	Aaron Gate	NZL	male	1990/11/26	1.81	71.0	cycling	
5	173071782	Aaron Royle	AUS	male	1990/1/26	1.80	67.0	triathlon	
6	266237702	Aaron Russell	USA	male	1993/6/4	2.05	98.0	volleyball	
...	...	...	...	...	...	...	...	...	
11533	265605954	Zurian Hechavarria	CUB	female	1995/8/10	1.64	58.0	athletics	
11534	214461847	Zuzana Hejnova	CZE	female	1986/12/19	1.73	63.0	athletics	
11535	88361042	di Xiao	CHN	male	1991/5/14	1.85	100.0	wrestling	
11536	900065925	le Quoc Toan Tran	VIE	male	1989/4/5	1.60	56.0	weightlifting	
11537	711404576	le Roux Hamman	RSA	male	1992/1/6	1.85	70.0	athletics	

11536 rows × 12 columns

In [10]:

```
Data[:10]
```

Out[10]:

	id	name	nationality	sex	date_of_birth	height	weight	sport	gold
0	736041664	A Jesus Garcia	ESP	male	1969/10/17	1.72	64.0	athletics	0
1	532037425	A Lam Shin	KOR	female	1986/9/23	1.68	56.0	fencing	0
2	435962603	Aaron Brown	CAN	male	1992/5/27	1.98	79.0	athletics	0
3	521041435	Aaron Cook	MDA	male	1991/1/2	1.83	80.0	taekwondo	0
4	33922579	Aaron Gate	NZL	male	1990/11/26	1.81	71.0	cycling	0
5	173071782	Aaron Royle	AUS	male	1990/1/26	1.80	67.0	triathlon	0
6	266237702	Aaron Russell	USA	male	1993/6/4	2.05	98.0	volleyball	0
7	382571888	Aaron Younger	AUS	male	1991/9/25	1.93	100.0	aquatics	0
8	87689776	Aauri Lorena Bokesa	ESP	female	1988/12/14	1.80	62.0	athletics	0
9	997877719	Ababel Yeshaneh	ETH	female	1991/7/22	1.65	54.0	athletics	0

8.4. 列へのアクセス

Data[['列名1', '列名2']] のように列挙する。

In [11]:

```
Data[['name']]
```

Out[11]:

	name
0	A Jesus Garcia
1	A Lam Shin
2	Aaron Brown
3	Aaron Cook
4	Aaron Gate
...	...
11533	Zurian Hechavarria
11534	Zuzana Hejnova
11535	di Xiao
11536	le Quoc Toan Tran
11537	le Roux Hamman

11538 rows × 1 columns

In [12]:

```
Data[['name', 'sport']]
```

Out[12]:

	name	sport
0	A Jesus Garcia	athletics
1	A Lam Shin	fencing
2	Aaron Brown	athletics
3	Aaron Cook	taekwondo
4	Aaron Gate	cycling
...	...	...
11533	Zurian Hechavarria	athletics
11534	Zuzana Hejnova	athletics
11535	di Xiao	wrestling
11536	le Quoc Toan Tran	weightlifting
11537	le Roux Hamman	athletics

11538 rows × 2 columns

8.5. 条件抽出

特定のカラムに注目して、条件に合う行だけを抽出する。

たとえば、sport カラムから athletics だけを抽出してみよう。抽出してできた新しい DataFrame は DataA と名付ける。

In [13]:

```
DataA = Data[Data['sport']=='athletics']
DataA
```

Out[13]:

	id	name	nationality	sex	date_of_birth	height	weight	sport	gold
0	736041664	A Jesus Garcia	ESP	male	1969/10/17	1.72	64.0	athletics	0
2	435962603	Aaron Brown	CAN	male	1992/5/27	1.98	79.0	athletics	0
8	87689776	Aauri Lorena Bokesa	ESP	female	1988/12/14	1.80	62.0	athletics	0
9	997877719	Ababel Yeshaneh	ETH	female	1991/7/22	1.65	54.0	athletics	0
10	343694681	Abadi Hadis	ETH	male	1997/11/6	1.70	63.0	athletics	0
...	...	...	...	...	...	...	...	...	...
11520	724419150	Zouhair Aouad	BRN	male	1989/4/7	1.75	69.0	athletics	0
11525	999437858	Zsofia Erdelyi	HUN	female	1987/12/10	1.64	53.0	athletics	0
11533	265605954	Zurian Hechavarria	CUB	female	1995/8/10	1.64	58.0	athletics	0
11534	214461847	Zuzana Hejnova	CZE	female	1986/12/19	1.73	63.0	athletics	0
11537	711404576	le Roux Hamman	RSA	male	1992/1/6	1.85	70.0	athletics	0

2363 rows × 12 columns

複数条件を設定することも可能。条件が「PかつQ」のときは P & Q と書く。

ちなみに、条件が「PまたはQ」「P or Q」のときは P|Q と書く。

In [14]:

```
DataAMJ = Data[(Data['sport']=='athletics') & (Data['sex']=='male') & (Data['nationality']=='JPN')]
DataAMJ.head(10)
```

Out[14]:

	id	name	nationality	sex	date_of_birth	height	weight	sport	gold	sil
242	327133724	Akihiko Nakamura	JPN	male	1990/10/23	1.80	73.0	athletics	0	
1208	95516264	Aska Cambridge	JPN	male	1993/5/31	1.79	74.0	athletics	0	
2246	102532767	Daichi Sawano	JPN	male	1980/9/16	1.83	74.0	athletics	0	
2249	212523902	Daigo Hasegawa	JPN	male	1990/2/27	1.73	58.0	athletics	0	
2255	46762119	Daisuke Matsunaga	JPN	male	1995/3/24	1.74	60.0	athletics	0	
2838	407225209	Eiki Takahashi	JPN	male	1992/11/19	1.76	58.0	athletics	0	
4119	361521049	Hiroki Ogita	JPN	male	1987/12/30	1.86	80.0	athletics	0	
4122	361129434	Hirooki Arai	JPN	male	1988/5/18	1.80	62.0	athletics	0	
4126	327247777	Hisanori Kitajima	JPN	male	1984/10/16	1.71	55.0	athletics	0	
4417	624395952	Isamu Fujisawa	JPN	male	1987/10/12	1.65	54.0	athletics	0	

8.6. 再インデックス付け

reset_index() メソッドを使う。

デフォルト：旧インデックスが新しいカラムとして追加される。

reset_index(drop=True)：旧インデックスが削除され、連番に置き換わる。

In [15]:

```
DataAMJ_new = DataAMJ.reset_index(drop=True)
DataAMJ_new.head()
```

Out[15]:

	id	name	nationality	sex	date_of_birth	height	weight	sport	gold	silver
0	327133724	Akihiko Nakamura	JPN	male	1990/10/23	1.80	73.0	athletics	0	0
1	95516264	Aska Cambridge	JPN	male	1993/5/31	1.79	74.0	athletics	0	1
2	102532767	Daichi Sawano	JPN	male	1980/9/16	1.83	74.0	athletics	0	0
3	212523902	Daigo Hasegawa	JPN	male	1990/2/27	1.73	58.0	athletics	0	0
4	46762119	Daisuke Matsunaga	JPN	male	1995/3/24	1.74	60.0	athletics	0	0

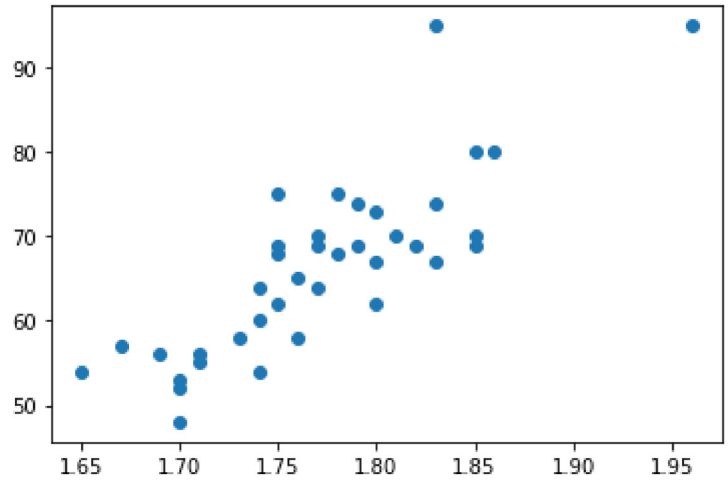
【例】日本の男子 Athletics 選手の身長と体重の散布図（描画の整形は要するが省略）

In [16]:

```
plt.scatter(DataAMJ_new['height'],DataAMJ_new['weight'])
```

Out[16]:

```
<matplotlib.collections.PathCollection at 0x196247112b0>
```



8.7. カラム（列）と行の削除

削除するカラム名を列挙する。

drop('カラム名', axis=1)　または　drop(['カラム名','カラム名'], axis=1)

上書きしたければ, オプションに inplace=True を追記する。

In [17]:

```
Data_small = Data.drop(['id','gold', 'silver', 'bronze', 'info'], axis=1)
Data_small
```

Out[17]:

	name	nationality	sex	date_of_birth	height	weight	sport
0	A Jesus Garcia	ESP	male	1969/10/17	1.72	64.0	athletics
1	A Lam Shin	KOR	female	1986/9/23	1.68	56.0	fencing
2	Aaron Brown	CAN	male	1992/5/27	1.98	79.0	athletics
3	Aaron Cook	MDA	male	1991/1/2	1.83	80.0	taekwondo
4	Aaron Gate	NZL	male	1990/11/26	1.81	71.0	cycling
...	...	...	...	...	...	...	...
11533	Zurian Hechavarria	CUB	female	1995/8/10	1.64	58.0	athletics
11534	Zuzana Hejnova	CZE	female	1986/12/19	1.73	63.0	athletics
11535	di Xiao	CHN	male	1991/5/14	1.85	100.0	wrestling
11536	le Quoc Toan Tran	VIE	male	1989/4/5	1.60	56.0	weightlifting
11537	le Roux Hamman	RSA	male	1992/1/6	1.85	70.0	athletics

11538 rows × 7 columns

行の削除は行番号を指定する。

Data.drop([i], axis=0) または Data.drop([i, j], axis=0) のように行番号を列挙する。 ただし、axis=0 は省略可。

In [18]:

```
Data.drop([1,3], axis=0)
```

Out[18]:

	id	name	nationality	sex	date_of_birth	height	weight	sport	gc
0	736041664	A Jesus Garcia	ESP	male	1969/10/17	1.72	64.0	athletics	
2	435962603	Aaron Brown	CAN	male	1992/5/27	1.98	79.0	athletics	
4	33922579	Aaron Gate	NZL	male	1990/11/26	1.81	71.0	cycling	
5	173071782	Aaron Royle	AUS	male	1990/1/26	1.80	67.0	triathlon	
6	266237702	Aaron Russell	USA	male	1993/6/4	2.05	98.0	volleyball	
...	...	...	...	...	...	...	...	...	
11533	265605954	Zurian Hechavarria	CUB	female	1995/8/10	1.64	58.0	athletics	
11534	214461847	Zuzana Hejnova	CZE	female	1986/12/19	1.73	63.0	athletics	
11535	88361042	di Xiao	CHN	male	1991/5/14	1.85	100.0	wrestling	
11536	900065925	le Quoc Toan Tran	VIE	male	1989/4/5	1.60	56.0	weightlifting	
11537	711404576	le Roux Hamman	RSA	male	1992/1/6	1.85	70.0	athletics	

11536 rows × 12 columns

8.8 カラムに現れるデータ値の収集

特定のカラムを用いてデータを分類することはしばしば必要になる。

たとえば、カラム nationality を見ると、異なった国名が出てくる。これらの国名のリストや出現度数などを集計しよう。

In [19]:

```
# カラムに含まれる異なる文字列をリストにする
ListNationality = np.unique(Data['nationality'])   # ソートされて出力される
ListNationality
```

Out[19]:

```
array(['AFG', 'ALB', 'ALG', 'AND', 'ANG', 'ANT', 'ARG', 'ARM', 'ARU',
       'ASA', 'AUS', 'AUT', 'AZE', 'BAH', 'BAN', 'BAR', 'BDI', 'BEL',
       'BEN', 'BER', 'BHU', 'BIH', 'BIZ', 'BLR', 'BOL', 'BOT', 'BRA',
       'BRN', 'BRU', 'BUL', 'BUR', 'CAF', 'CAM', 'CAN', 'CAY', 'CGO',
       'CHA', 'CHI', 'CHN', 'CIV', 'CMR', 'COD', 'COK', 'COL', 'COM',
       'CPV', 'CRC', 'CRO', 'CUB', 'CYP', 'CZE', 'DEN', 'DJI', 'DMA',
       'DOM', 'ECU', 'EGY', 'ERI', 'ESA', 'ESP', 'EST', 'ETH', 'FIJ',
       'FIN', 'FRA', 'FSM', 'GAB', 'GAM', 'GBR', 'GBS', 'GEO', 'GEQ',
       'GER', 'GHA', 'GRE', 'GRN', 'GUA', 'GUI', 'GUM', 'GUY', 'HAI',
       'HKG', 'HON', 'HUN', 'INA', 'IND', 'IOA', 'IRI', 'IRL', 'IRQ',
       'ISL', 'ISR', 'ISV', 'ITA', 'IVB', 'JAM', 'JOR', 'JPN', 'KAZ',
       'KEN', 'KGZ', 'KIR', 'KOR', 'KOS', 'KSA', 'LAO', 'LAT', 'LBA',
       'LBR', 'LCA', 'LES', 'LIB', 'LIE', 'LTU', 'LUX', 'MAD', 'MAR',
       'MAS', 'MAW', 'MDA', 'MDV', 'MEX', 'MGL', 'MHL', 'MKD', 'MLI',
       'MLT', 'MNE', 'MON', 'MOZ', 'MRI', 'MTN', 'MYA', 'NAM', 'NCA',
       'NED', 'NEP', 'NGR', 'NIG', 'NOR', 'NRU', 'NZL', 'OMA', 'PAK',
       'PAN', 'PAR', 'PER', 'PHI', 'PLE', 'PLW', 'PNG', 'POL', 'POR',
       'PRK', 'PUR', 'QAT', 'ROT', 'ROU', 'RSA', 'RUS', 'RWA', 'SAM',
       'SEN', 'SEY', 'SIN', 'SKN', 'SLE', 'SLO', 'SMR', 'SOL', 'SOM',
       'SRB', 'SRI', 'SSD', 'STP', 'SUD', 'SUI', 'SUR', 'SVK', 'SWE',
       'SWZ', 'SYR', 'TAN', 'TGA', 'THA', 'TJK', 'TKM', 'TLS', 'TOG',
       'TPE', 'TTO', 'TUN', 'TUR', 'TUV', 'UAE', 'UGA', 'UKR', 'URU',
       'USA', 'UZB', 'VAN', 'VEN', 'VIE', 'VIN', 'YEM', 'ZAM', 'ZIM'],
      dtype=object)
```

In [20]:

```
# カラムに含まれる異なる文字列ごとに、その出現頻度をリストする
Data['nationality'].value_counts()  # 出現頻度の大きいほうから自動的にソートされる
```

Out[20]:

```
USA    567
BRA    485
GER    441
AUS    431
FRA    410
      ...
SWZ      2
NRU      2
MTN      2
CHA      2
TUV      1
Name: nationality, Length: 207, dtype: int64
```

In [21]:

```
# 上の結果を DataFrame にする
NationalityFrequency = pd.DataFrame(Data['nationality'].value_counts())
NationalityFrequency
```

Out[21]:

	nationality
USA	567
BRA	485
GER	441
AUS	431
FRA	410
...	...
SWZ	2
NRU	2
MTN	2
CHA	2
TUV	1

207 rows × 1 columns

In [22]:

```
NationalityFrequency.rename(columns={'nationality':'frequency'}, inplace=True)
```

特定のカラムでソートするためには sort_values(by='カラム名', ascending=True) メソッドを使う。オプションは、

昇順：ascending=True

降順：ascending=False

In [23]:

```
NationalityFrequency.sort_values(by='frequency',ascending=True)
```

Out[23]:

	frequency
TUV	1
SOM	2
LBR	2
GEQ	2
BHU	2
...	...
FRA	410
AUS	431
GER	441
BRA	485
USA	567

207 rows × 1 columns

インデックスでソートするためには sort_index() メソッドを使う。

In [24]:

```
NationalityFrequency.sort_index()
```

Out[24]:

	frequency
AFG	3
ALB	6
ALG	68
AND	5
ANG	26
...	...
VIE	23
VIN	4
YEM	3
ZAM	7
ZIM	35

207 rows × 1 columns