

# 医療系データサイエンス教育ワークショップ2023

医療とAI・ビッグデータ入門演習内容紹介（45min/コマ×6コマ）

2024.2.22

本スライドは、自由にお使いください。  
使用した場合は、このQRコードからアンケート  
に回答をお願いします。



統合教育機構 曹 日丹

## 医療とAI・ビッグデータ入門演習内容紹介（45min/コマ×6コマ）

変数とリストの概念

プログラミング基礎：for文とif文

Pandasライブラリ

Python基礎

架空データ

患者の歯に関する病院の~~リアルワールドデータ~~の説明

（演習）データクレンジングによる歯の本数の集計

（演習）データクレンジングによる患者年齢の集計および可視化

データクレンジング

## 患者の歯に関する病院の~~リアルワールドデータ~~の説明

### 架空データ

東京医科歯科大学**歯科診療部門**初診患者さんのリアルワールドデータを抽出しました。

絞込条件：

初診日：2018/1/1-2022/12/31

生年月日：1943/1/1-1982/12/31（2023年1月1日時点：40歳以上、81歳未満）

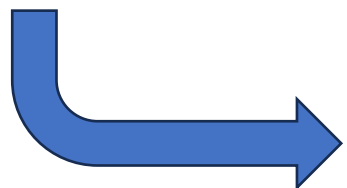
抽出した内容：

id：匿名化ID

歯式と歯の状態：残存歯、喪失歯

性別：男、女

birthmonth：生年月、初診日：2018/1/1-2022/12/31



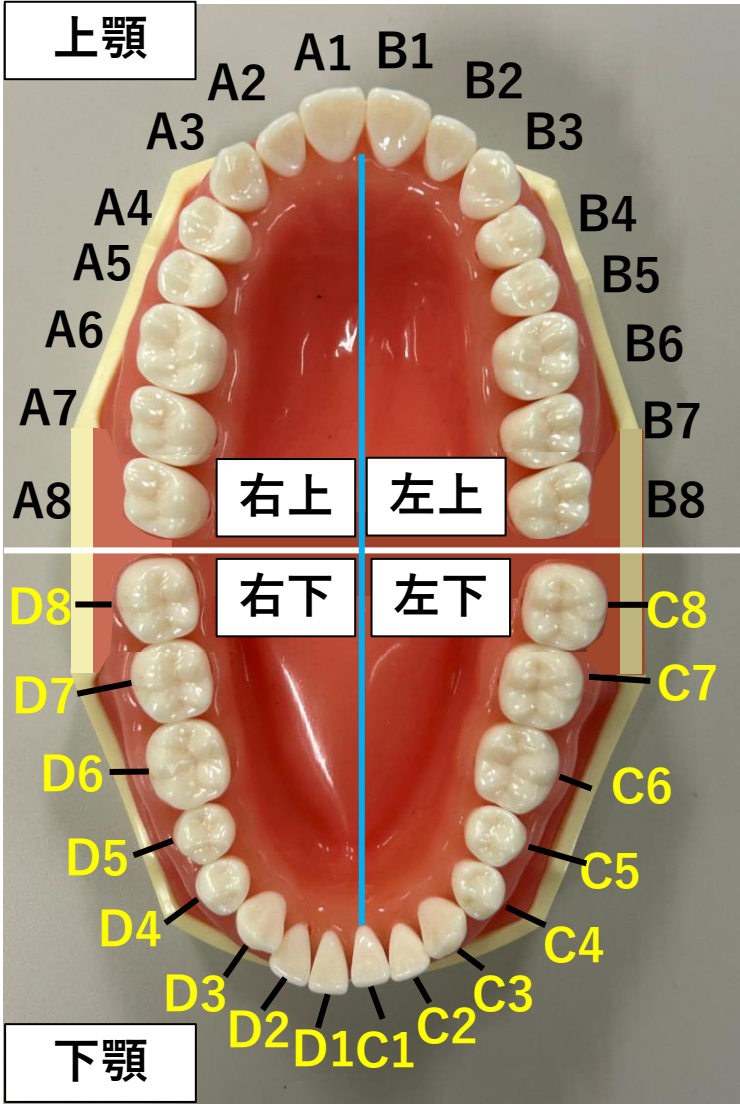
一部を無作為にシャッフルして、初診時年齢を算出して、**架空データ**を作成し、練習用データとしました。

患者の歯に関する病院のリアルワールドデータの説明

EXCEL  
ファイル

架空データ

	A	B	C	D	E
1	id	tooth name	tooth record	gender	age
2	pt_1	A1	残存歯	男	77
3	pt_1	A2	喪失歯	男	77
4	pt_1	A3	残存歯	男	77
5	pt_1	A4	残存歯	男	77
6	pt_1	A5	残存歯	男	77
7	pt_1	A6	残存歯	男	77
8	pt_1	A7	喪失歯	男	77
9	pt_1	A8	残存歯	男	77
10	pt_1	B1	残存歯	男	77
11	pt_1	B2	喪失歯	男	77
12	pt_1	B3	残存歯	男	77
13	pt_1	B4	残存歯	男	77
14	pt_1	B5	喪失歯	男	77
15	pt_1	B6	残存歯	男	77
16	pt_1	B7	残存歯	男	77
17	pt_1	B8	残存歯	男	77
18	pt_1	C1	残存歯	男	77
19	pt_1	C2	残存歯	男	77
20	pt_1	C3	喪失歯	男	77



患者の歯に関する病院のリアルワールドデータの説明

EXCEL  
ファイル

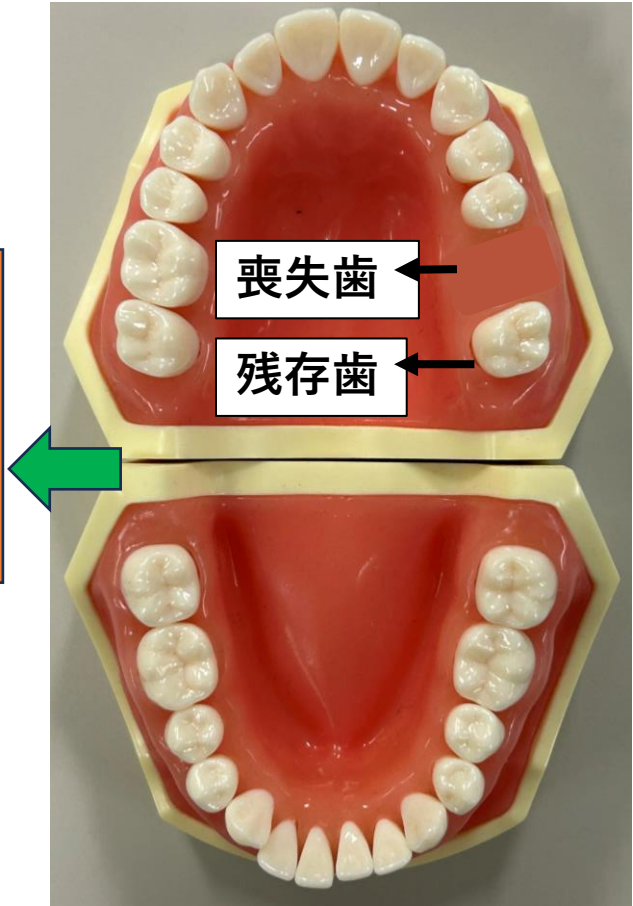
架空データ

	A	B	C	D	E
1	id	tooth name	tooth record	gender	age
2	pt_1	A1	残存歯	男	77
3	pt_1	A2	喪失歯	男	77
4	pt_1	A3	残存歯	男	77
5	pt_1	A4	残存歯	男	77
6	pt_1	A5	残存歯	男	77
7	pt_1	A6	残存歯	男	77
8	pt_1	A7	喪失歯	男	77
9	pt_1	A8	残存歯	男	77
10	pt_1	B1	残存歯	男	77
11	pt_1	B2	喪失歯	男	77
12	pt_1	B3	残存歯	男	77
13	pt_1	B4	残存歯	男	77
14	pt_1	B5	喪失歯	男	77
15	pt_1	B6	残存歯	男	77
16	pt_1	B7	残存歯	男	77
17	pt_1	B8	残存歯	男	77
18	pt_1	C1	残存歯	男	77
19	pt_1	C2	残存歯	男	77
20	pt_1	C3	喪失歯	男	77

歯の部位	歯の状態
B6	喪失歯
B7	残存歯

**残存歯：**  
口腔内に存在している歯

**喪失歯：**  
抜歯などによって欠損している歯



患者の歯に関する病院のリアルワールドデータの説明

EXCEL  
ファイル

架空データ

1	id	tooth name	tooth record	gender	age
2	pt_1	A1	残存歯	男	77
3	pt_1	A2	喪失歯	男	77
4	pt_1	A3	残存歯	男	77
5	pt_1	A4	残存歯	男	77
6	pt_1	A5	残存歯	男	77
...					
11419	pt_357	D2	残存歯	女	69
11420	pt_357	D3	残存歯	女	69
11421	pt_357	D4	喪失歯	女	69
11422	pt_357	D5	残存歯	女	69
11423	pt_357	D6	残存歯	女	69
...					
50636	pt_1583	B3	残存歯	女	40
50637	pt_1583	B4	残存歯	女	40
50638	pt_1583	B5	残存歯	女	40
50639	pt_1583	B6	残存歯	女	40
50640	pt_1583	B7	残存歯	女	40

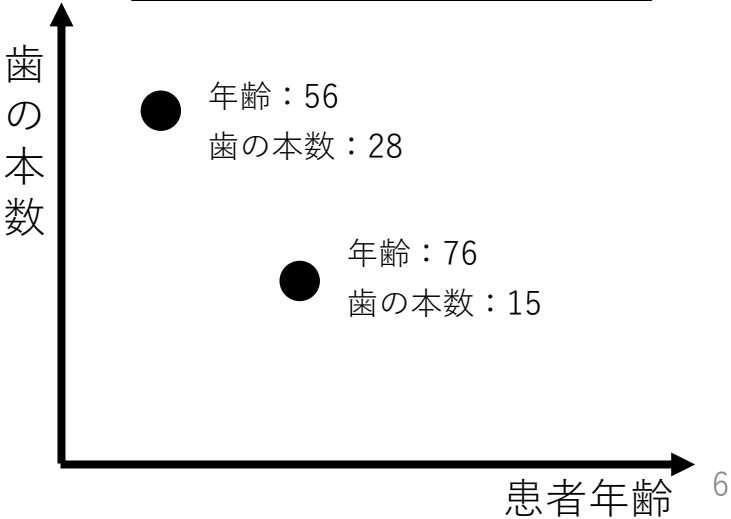
データクレンジング

演習の目標

ID	歯の本数	初診時年齢
pt_1	○本	77歳
...		
pt_357	○本	69歳
...		
pt_1583	○本	40歳

データを可視化する

散布図：年齢と歯の本数

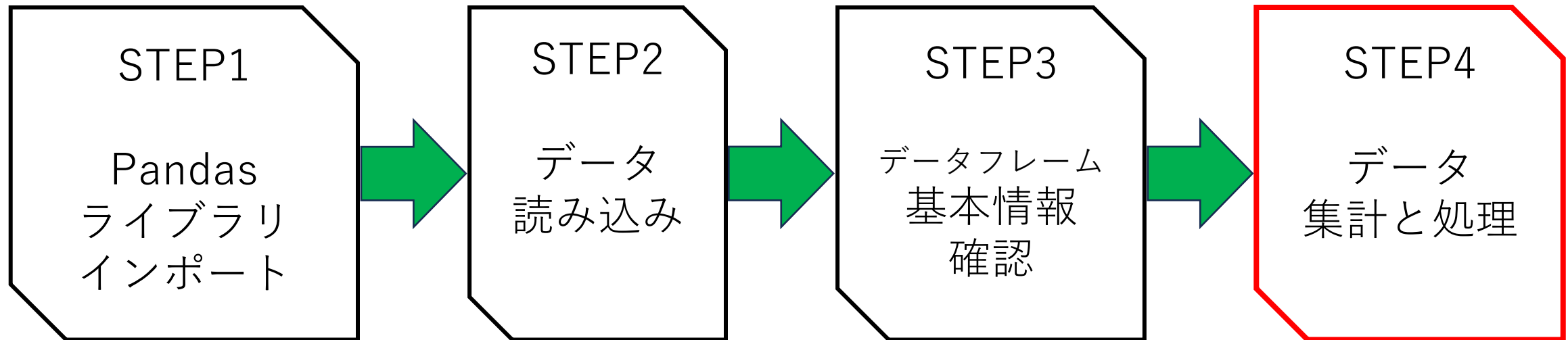


# Pythonを使用してデータクレンジングを行う必要性

データクレンジングは臨床研究や病院データ管理などの医療領域で非常に重要な手法です。

データクレンジングとは、データセットの特定の問題を修正し、データを分析や機械学習モデルのトレーニングに適した形式に整えるプロセスです。

## Pythonを使用してデータクレンジングを行うSTEP



# Pythonを使用してデータクレンジングを行うSTEP

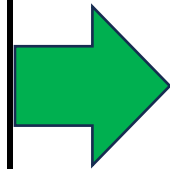
## STEP0

Python基礎  
変数、リスト  
データの型  
ライブラリ



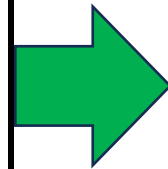
## STEP1

Pandas  
ライブラリ  
インポート



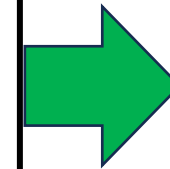
## STEP2

データ  
読み込み



## STEP3

データフレーム  
基本情報



## STEP4

データ  
集計と処理



**変数**とは、データや値を一時的に保持するための名前付きの「箱」のようなものです。  
この変数を使用することで、プログラム中でその値を繰り返し利用することができます。

### コード01

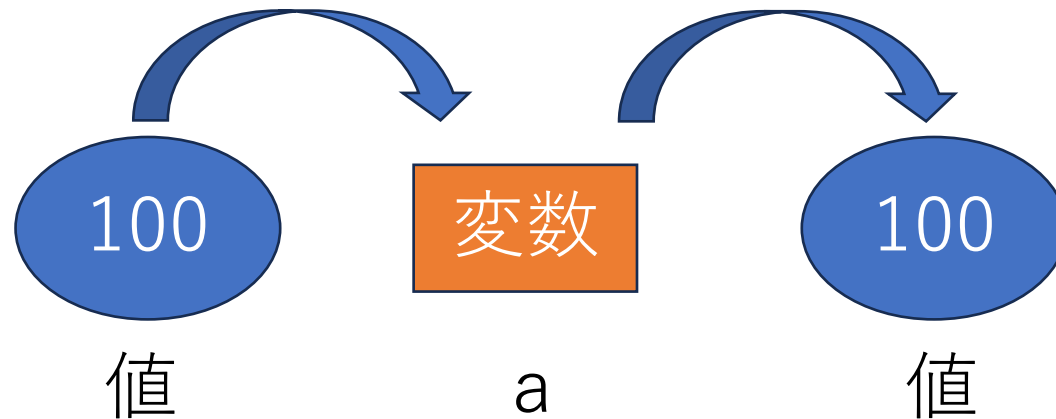
プログラミング言語  
で表現します

変数に値を代入

```
a = 100
```

変数の値を出力

```
print(a)
```



## STEP0


Python基礎  
変数、リスト  
データの型  
ライブラリ

コード01

```
a = 100  
print(a)
```

Google Colab画面

コード01

実行するには横の  か、shift + enterを押します。



```
a = 100  
print(a)
```

セル

実行ボタン

コード



100

実行結果

Colab

<http://colab.research.google.com>

# 検索google colab Colaboratory へようこそ

CO

Colaboratory へようこそ

ファイル 編集 表示

目次

はじめに

データサイエンス

機械学習

その他のリソース

使用例

セクション

ノートブックを開く

例

最近

Google ドライブ

GitHub

アップロード

ノートブックを検索

タイトル	最終閲覧	最初に開いた日時	
演習6コード	9:43	10月25日	
Colaboratory へようこそ	9:21	2022年12月23日	
演習7コード	9:21	9:21	
演習4コード	11月2日	11月1日	
演習5コード	11月1日	10月13日	

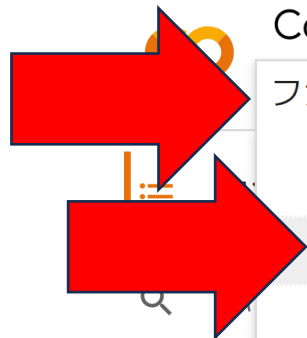
+ ノートブックを新規作成

キャンセル

# 検索google colab

<http://colab.research.google.com>

## Colaboratory へようこそ



Colaboratory へようこそ

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

ノートブックを新規作成

ノートブックを開く

Ctrl+O

ノートブックをアップロード

名前の変更

ドライブにコピーを保存

コピーを GitHub Gist として保存

GitHub にコピーを保存

保存

Ctrl+S

変更履歴

ダウンロード

印刷

Ctrl+P

ド + テキスト

ドライブにコピー

Colab へようこそ

に Colab をよくご存じの場合は、この動画でインタラクティブなラ  
ドの履歴表示、コマンドパレットについてご覧ください。



Colab とは

<http://colab.research.google.com>

## 検索google colab Colaboratory へようこそ

ノートブックを開く

例 >

最近 >

**Google ドライブ >**

GitHub >

アップロード >

タイトル	所有者	最終閲覧 ▲	最終更新 ▼		
<b>第1回コード.ipynb</b>					
演習準備資料.ipynb	曹日丹	11月1日	11月1日		
演習1116確認.ipynb のコピー	曹日丹	10月31日	10月27日		
2023入門dataframe.ipynb	曹日丹	10月31日	10月27日		
演習1116確認.ipynb	曹日丹	10月25日	10月25日		

+ ノートブックを新規作成

キャンセル

<http://colab.research.google.com>

検索google colab Colaboratory へようこそ

ノートブックを開く

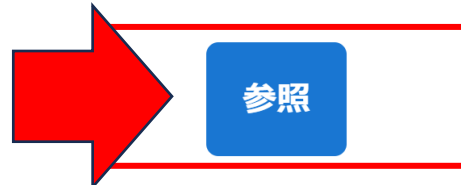
例 >

最近 >

Google ドラ  
イブ >

GitHub >

アップロード >



または、ここにファイルをドラッグしてください

第1回コード.ipynb

Colab

# ワークショップ中の質問をファシリテーターが対応させていただきます。

Zoom ミーティング

表示 ミーティング チャット

演習にエラーが出たなど問題があったらリアクションの**挙手**を押してください。

質問内容を入力して、「**全員**」宛てに送信してください。

Miho Ishimaru

👍 🙏 😂 😮 ❤️ 🎉 ...

👍 挙手

宛先: 全員

ここにメッセージを入力します...

ミュート解除 ビデオの開始 セキュリティ 参加者 画面共有 リアクション アプリ ホワイトボード ノート 詳細 終了

🗑️ 😊 📎 🔄 15

## コード02

```
a = 5
```

整数

```
b = 2.22
```

小数

```
c = 'Hello, World!'
```

文字

```
d = [1, 2, 3, 4, 5]
```

リスト

変数エクスペローラで型を確認できます

## STEP0

Python基礎  
変数、リスト  
データの型  
ライブラリ

Colab



## コード03



```
a= 'hello world'  
print(a)
```

文字

```
a= [2, 22]  
print(a)
```

リスト



```
hello world  
[2, 22]
```

1回目に実行した結果：文字

2回目に実行した結果：リスト

## STEP0

Python基礎  
変数、リスト  
データの型  
ライブラリ

Colab

## STEP0

Python基礎  
変数、リスト  
データの型  
ライブラリ

リストとは、**順序**が付けられた要素の集まりです。  
インデックスによって位置を指定して、リストの要素にアクセスできます。

## コード04

```
fruits = ['apple', 'banana', 'cherry', 'melon']
```

インデックス番号	[0]	[1]	[2]	[3]
----------	-----	-----	-----	-----

## STEP0

Python基礎  
変数、リスト  
データの型  
ライブラリ

## コード04

```
fruits = ['apple', 'banana', 'cherry', 'melon']
```

	↓	↓	↓	↓
インデックス番号	[0]	[1]	[2]	[3]

インデックス番号を使用して要素にアクセスする

```
print(fruits[0])
```

範囲を指定して要素にアクセスする

```
print(fruits[1:3])
```

## コード04

```
fruits = ['apple', 'banana', 'cherry', 'melon']
```

インデックス番号

fruits[0]

fruits[1]

fruits[2]

fruits[3]

```
print(fruits[0])
```

[→] apple

```
print(fruits[1:3])
```

[→] ['banana', 'cherry']

## STEP0

Python基礎  
変数、リスト  
データの型  
ライブラリ

Colab

Pythonライブラリは、データ分析、機械学習、データ可視化など、さまざまな高度な機能を提供します。

### NumPy :

Pythonで高速な数値計算を行うためのライブラリ。

**Pandas:**データクレンジングには、Pandasライブラリを使用します。

Pythonでデータ分析を容易にするライブラリで、構造化データの操作や分析を高速かつ効率的に行うためのデータ構造（DataFrame）と関数が提供されています。

### Matplotlib:

Pythonでグラフやチャートを描画するためのライブラリ。

### Scikit-learn:

Pythonで機械学習を行うためのライブラリで、分類、回帰、クラスタリングなどの機械学習アルゴリズムを簡単に利用できるように設計されています。

Pandasライブラリは、データ分析と操作のための強力なツールを提供しています。  
また、データフレーム(DataFrame)というデータ構造を用いて分析できるようになります。

### データフレームとは

表形式のデータを扱うための主要なツールです。

データフレーム

二次元のデータ構造：エクセルのような行と列から成る表形式のデータ構造です。

軸： 各行をインデックス番号で特定します。  
各列をその列名で特定します。

STEP0

Python基礎

変数、リスト

データの型

ライブラリ

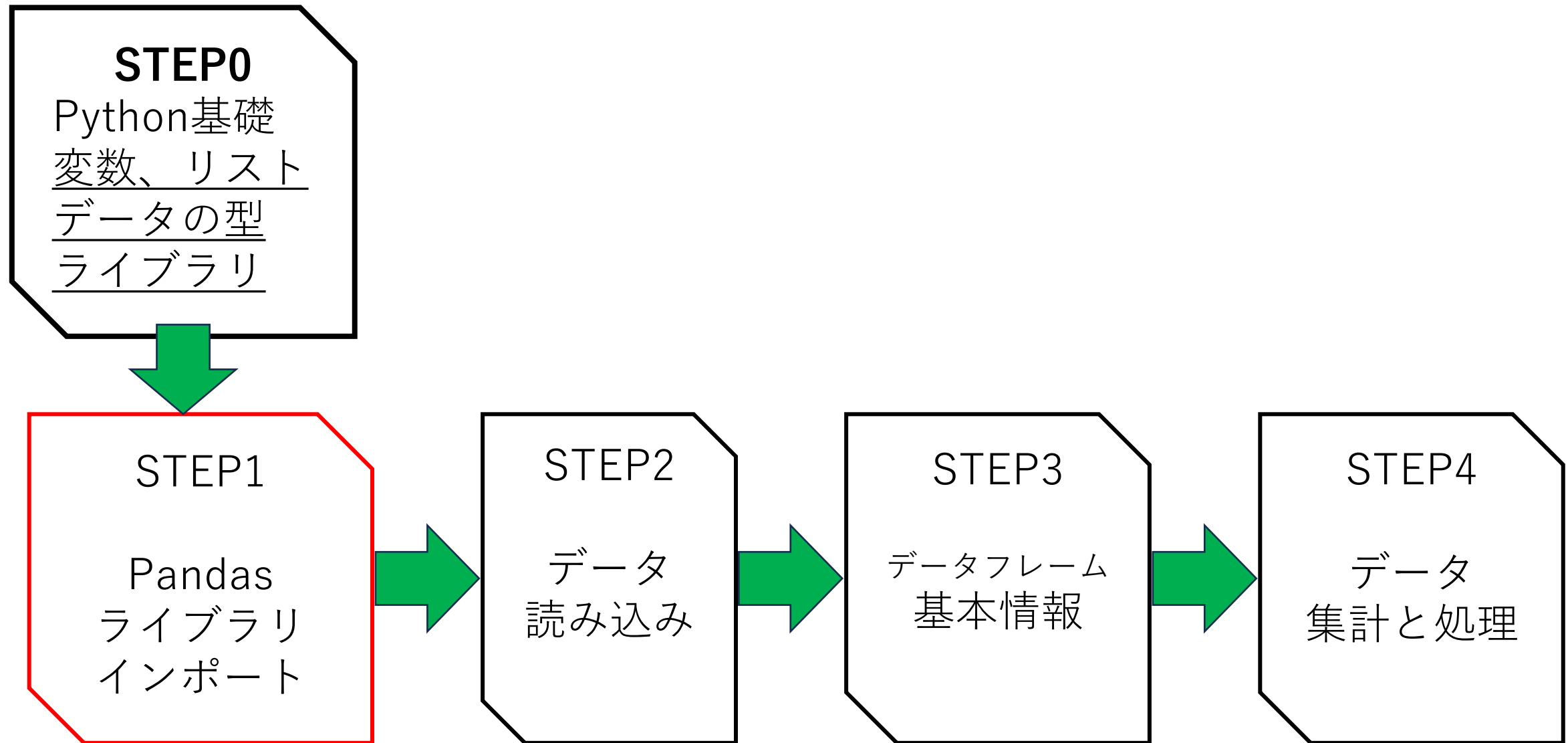
列名

	ID	名前	年齢	BMI
0	10001	花子	58	26.2
1	10002	太郎	61	20.5
2	10003	良子	70	23.4
3	10004	三郎	92	18.2

インデックス番号

リストと同様に0  
～n-1のインデッ  
クス番号が付与さ  
れています

# Pythonを使用してデータクレンジングを行うSTEP





## STEP1

Pandas  
ライブラリ  
インポート

Pandas-ライブラリのデータフレーム機能を使うため、  
Pandasライブラリをインポートする必要があります。

### コード05

```
import pandas as pd
```

Pandasというライブラリを**pd**という略称でインポートします。  
(pdでなくても良いですが、多くの人がpdを使っています。)

## STEP2

データ  
読み込み

<https://drive.google.com/>

Googleアカウントでログインします。



Google

<https://drive.google.com> > drive > my-drive

➡ <https://drive.google.com/>

Google ドライブには、Google アカウント（個人ユーザー向け）または Google Workspace アカウント（ビジネスユーザー向け）でアクセスできます。



ドライブ

🔍 ドライブで検索



マイドライブ ▾

種類 ▾

ユーザー ▾

最終更新 ▾

候補リスト

➡ + 新規

📁 マイドライブ

📁 パソコン

👤 共有アイテム

🕒 最近使用したアイテム

☆ スター付き

🔗 演習5コード.ipynb

演習5コード 12月7日

🔗 演習5dataframe.ipynb

```
import pandas as pd
person = {"name": "John", "age": 30, "city": "New York"}
```

🔗 演習4コード.ipynb

## STEP2

データ  
読み込み

Google Driveにファイル（ dwhdata20240222.csv ）をアップロードします



<https://drive.google.com/>

Googleアカウントでログインします。

+ 新規

マイドライブ

パソコン

共有アイテム

最近使用したアイテム

スター付き

スパム

ゴミ箱

保存容量



ドライブ



ド



新しいフォルダ



ファイルのアップロード

dwhdata20240222.csv



フォルダのアップロード



Google ドキュメント



Google スプレッドシート



Google スライド



Google フォーム

その他

STEP2

コード06

データ  
読み込み

```
from google.colab import drive  
drive.mount('/content/drive')
```

このノートブックに **Google** ドライブのファイルへのアクセスを許可しますか？

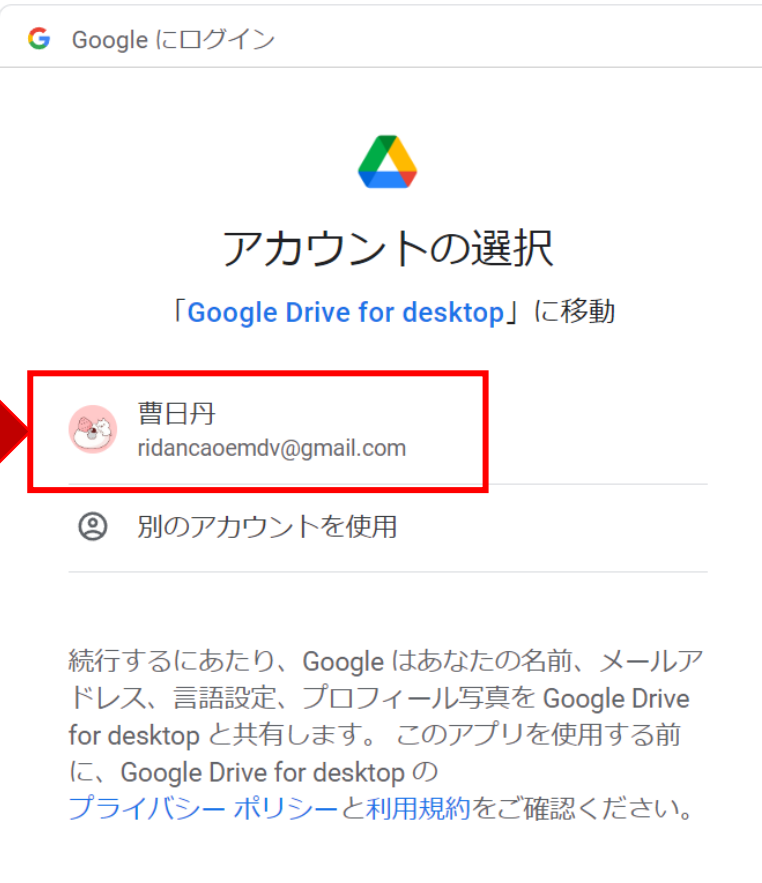
このノートブックは Google ドライブ ファイルへのアクセスをリクエストしています。Google ドライブへのアクセスを許可すると、ノートブックで実行されたコードに対し、Google ドライブ内のファイルの変更を許可することになります。このアクセスを許可する前に、ノートブック コードをご確認ください。

スキップ

Google ドライブに接続



**Drive already mounted** at /content/drive



Google Drive for desktop の [プライバシー ポリシー](#) と [利用規約](#) をご覧ください。

キャンセル

許可

28

## STEP2

データ  
読み込み

Google Colaboratoryでcsvファイルを読み込みます

Pandasライブラリの関数：`read_csv()`

DataFrame名= `pd.read_csv('ファイルパス')`

## コード07

```
klist = pd.read_csv('/content/drive/My Drive/dwhdata20240222.csv', encoding = 'utf-8')  
print(klist)
```


`encoding = 'utf-8'`

は開く際に文字エンコーディングを指定して文字化けを防ぐために入れています。

# データフレーム読み込み

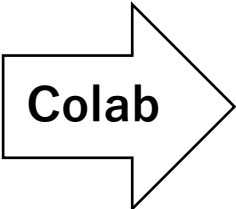
手順：Google Colaboratoryでcsvファイルを読み込みます

```
klist = pd.read_csv('/content/drive/My Drive/dwhdata20240222.csv', encoding = 'utf-8')
print(klist)
```

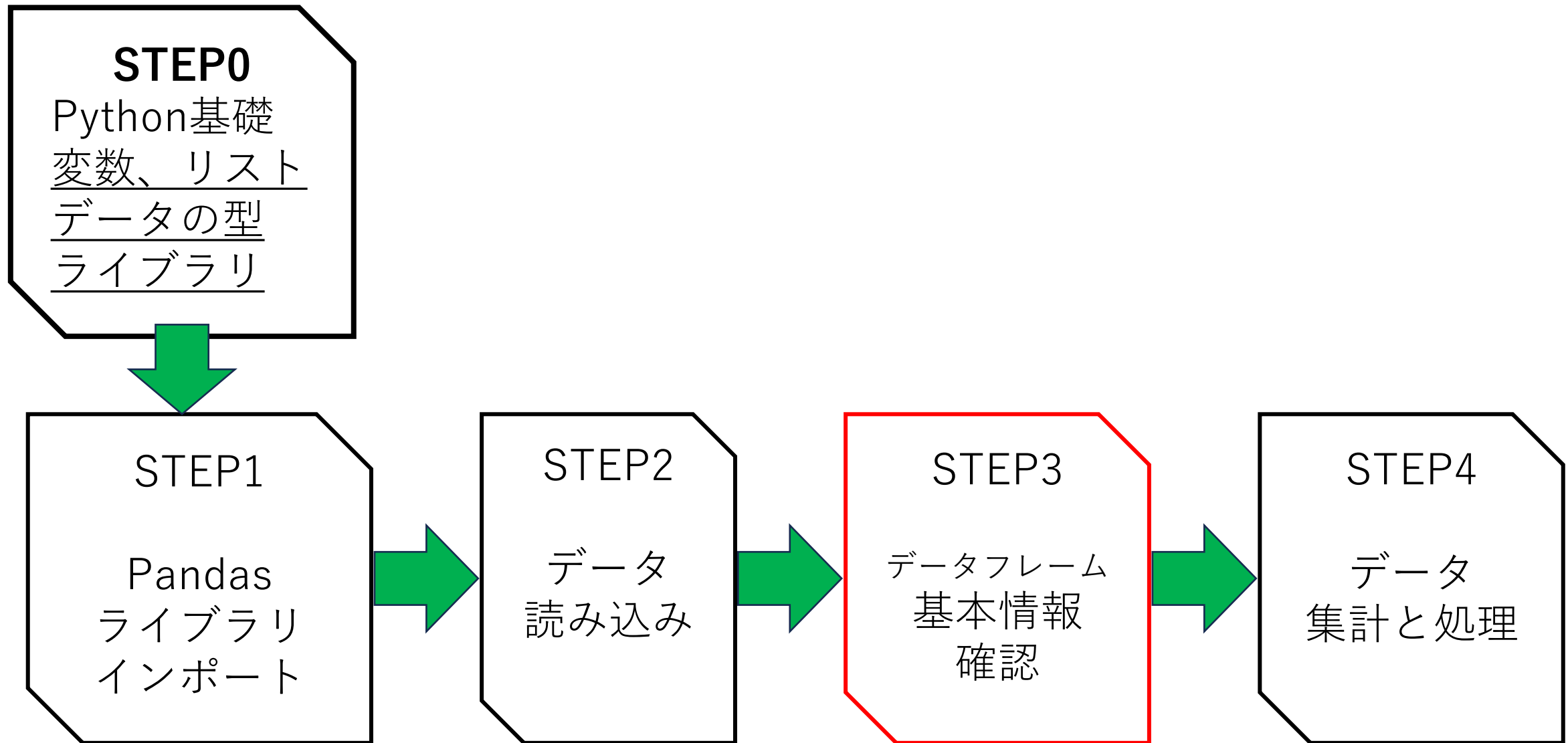


	id	tooth	name	tooth	record	gender	age
0	pt_1		A1		残存歯	男	77
1	pt_1		A2		喪失歯	男	77
2	pt_1		A3		残存歯	男	77
3	pt_1		A4		残存歯	男	77
4	pt_1		A5		残存歯	男	77
...	...		...		...	...	...
53019	pt_1657		D4		喪失歯	女	40
53020	pt_1657		D5		残存歯	女	40
53021	pt_1657		D6		残存歯	女	40
53022	pt_1657		D7		喪失歯	女	40
53023	pt_1657		D8		残存歯	女	40

[53024 rows x 5 columns]



# Pythonを使用してデータクレンジングを行うSTEP



### STEP3 データフレーム 基本情報

データフレームのshape（行×列）とsize（データ数）を求める

**コード08：** DataFrame名.shape: DataFrameの形状（行と列の数）を返します。

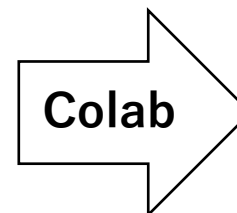
```
print(klist.shape)
```

**(53024, 5)**

**コード09：** DataFrame名.size: DataFrame内の要素（セル）の総数を返します。

```
print(klist.size)
```

**265120**





### STEP3

データフレーム  
基本情報

データフレーム内の特定の要素にアクセスする。

tooth record列

klistデータフレーム

Pandasの**locメソッド** (locは、"location"の略です)  
データフレーム内の特定の位置にアクセスできる。

`klist.loc[行のインデックス番号, '列名']`

コード10-1

```
print(klist.loc[0, 'tooth record'])
```

コード10-2

「:」を使用してtooth record列全ての範囲を取得します。  
`print(klist.loc[:, 'tooth record'])`

### STEP3

データフレーム  
基本情報

データフレーム内の特定の要素にアクセスする。

tooth record列

klistデータフレーム

Pandasの**locメソッド** (locは、"location"の略です)  
データフレーム内の特定の位置にアクセスできる。

コード10-1

```
print(klist.loc[0, 'tooth record'])
```

⇒ 残存歯

コード10-2

```
print(klist.loc[:, 'tooth record'])
```

⇒

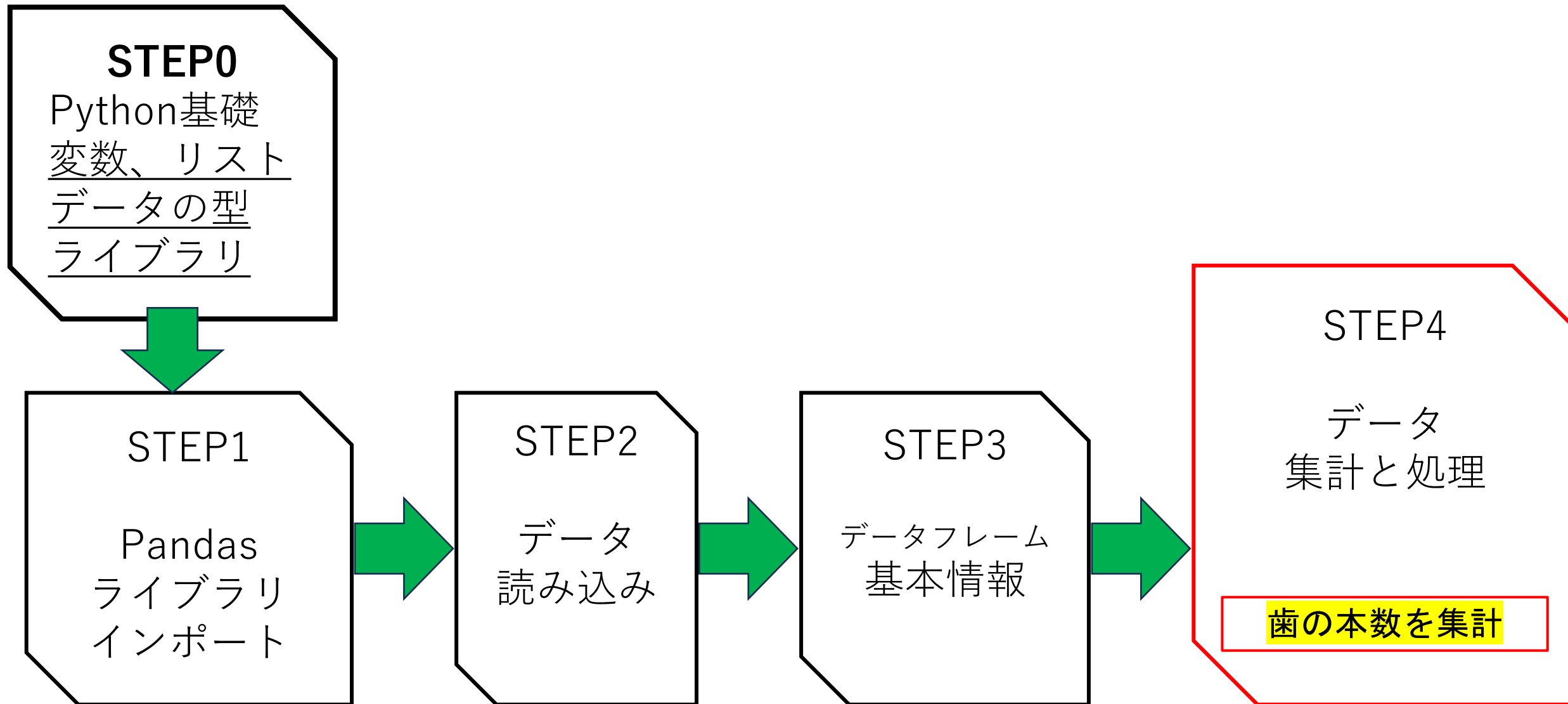
0	残存歯
1	喪失歯
2	残存歯
3	残存歯
4	残存歯

...

53019	喪失歯
53020	残存歯
53021	残存歯
53022	喪失歯
53023	残存歯

Colab

# Pythonを使用してデータクレンジングを行うSTEP



# 歯の本数を集計

## klistデータフレーム

tooth record列

tooth exist列

患者a	残存歯
	残存歯
	残存歯
	喪失歯
	残存歯
患者b	
	残存歯
	残存歯
	喪失歯
患者c	
	残存歯
	残存歯
	残存歯
患者d	喪失歯
	喪失歯

残存歯  
残存歯  
残存歯  
喪失歯  
残存歯

残存歯  
残存歯  
喪失歯

残存歯  
残存歯  
残存歯

喪失歯  
喪失歯

1

1

1

0

1

1

1

0

1

1

1

0

0

## 方法

残存歯と喪失歯を数字1と0に変換します。

患者ごとに足し算で本数を集計します。

歯本数a=

1

+

1

+

1

+

0

+

1

...

歯本数b=

1

+

1

+

0

...

歯本数c=

1

+

1

+

1

...

歯本数d=

0

+

0

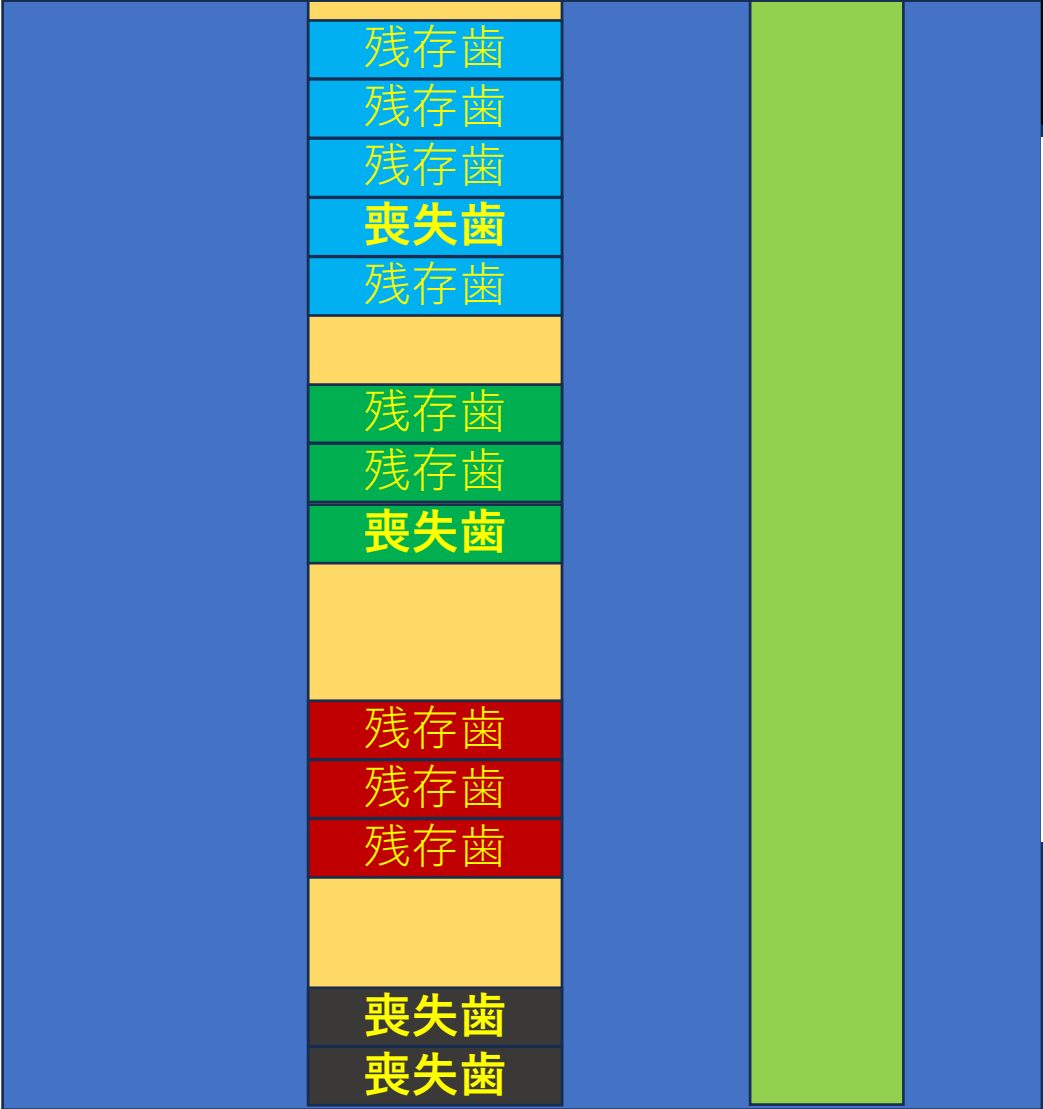
...

歯の本数を集計

klistデータフレーム

tooth record列

tooth exist列



残存歯
残存歯
残存歯
喪失歯
残存歯
残存歯
残存歯
喪失歯
残存歯
残存歯
残存歯
喪失歯
喪失歯

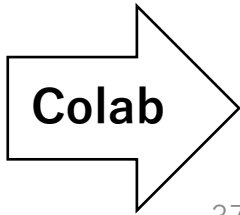
# コード11-1

列を追加する

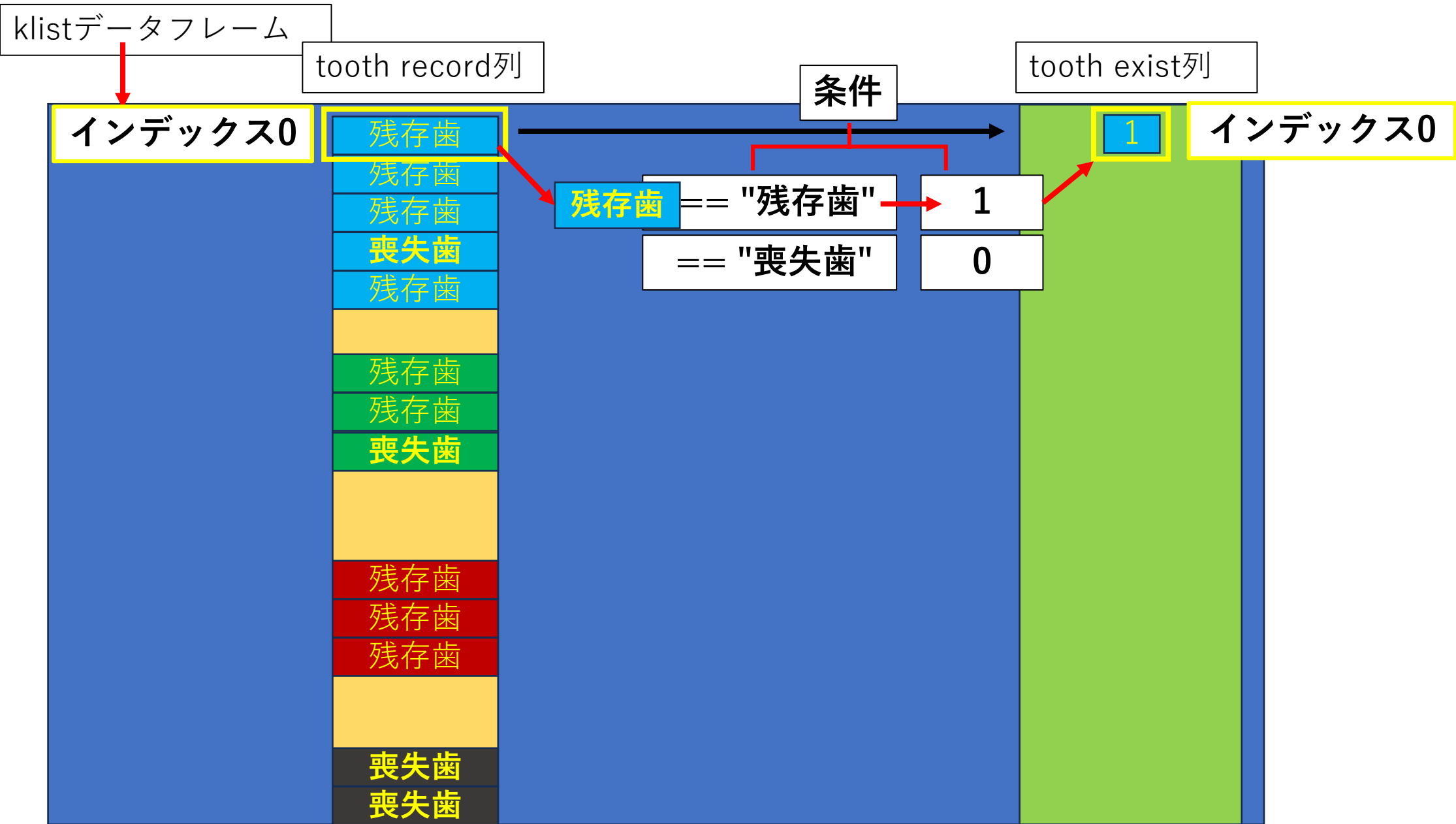
```
klist['tooth exist'] = None  
print(klist)
```

	id	tooth name	tooth record	gender	age	tooth exist
0	pt_1	A1	残存歯	男	77	None
1	pt_1	A2	喪失歯	男	77	None
2	pt_1	A3	残存歯	男	77	None
3	pt_1	A4	残存歯	男	77	None
4	pt_1	A5	残存歯	男	77	None
...	...	...	...	...	...	...
53019	pt_1657	D4	喪失歯	女	40	None
53020	pt_1657	D5	残存歯	女	40	None
53021	pt_1657	D6	残存歯	女	40	None
53022	pt_1657	D7	喪失歯	女	40	None
53023	pt_1657	D8	残存歯	女	40	None

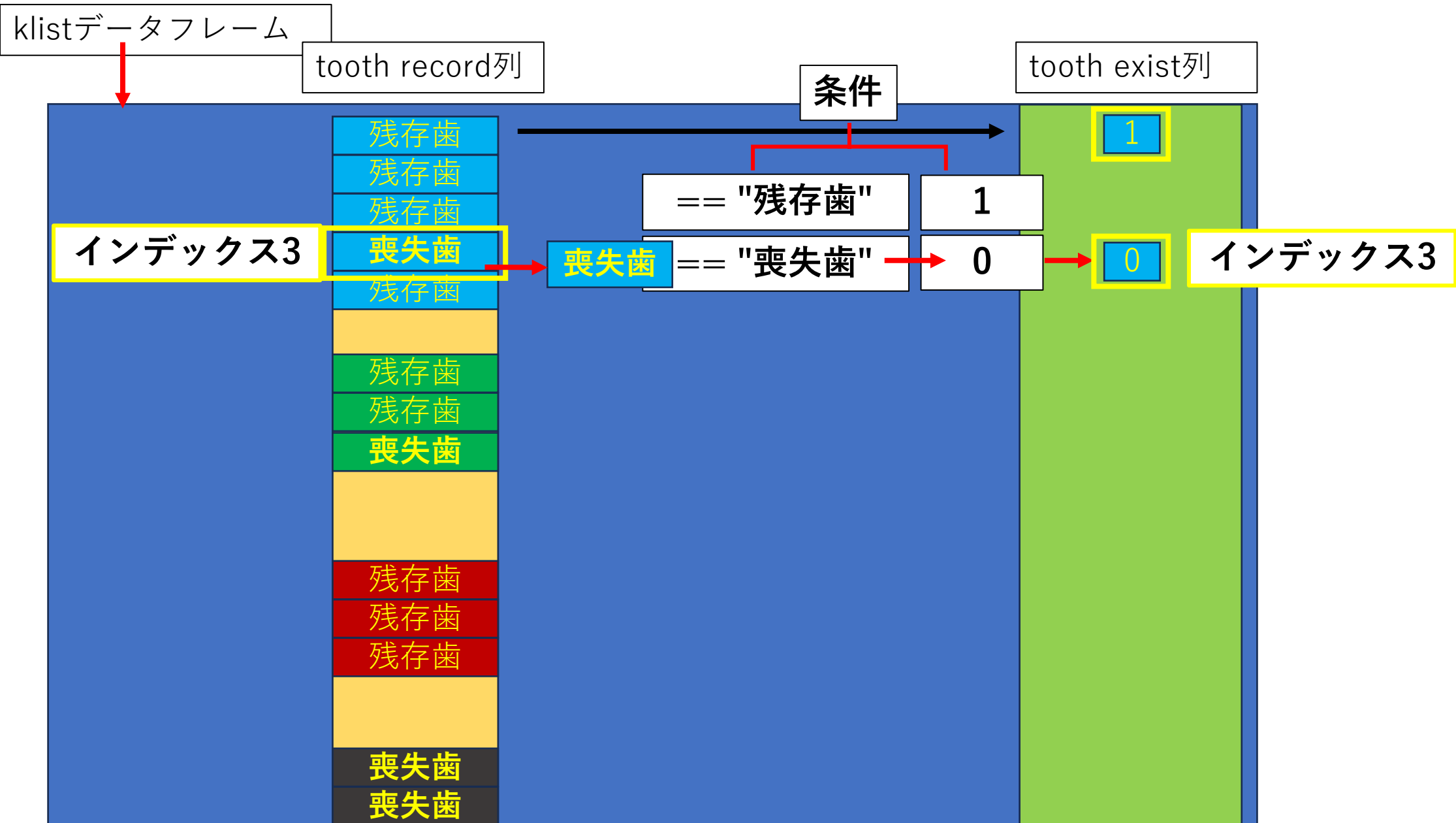
[53024 rows x 6 columns]



# 歯の本数を集計



# 歯の本数を集計



### if

日本語での**if文**に相当する部分は「**もし～なら**」や「**もし**条件が成り立つ**場合は**」などのように言えます。

if **条件**:

条件が**真**の場合に実行されるコード

else:

条件が**偽**の場合に実行されるコード

**真**(True)か**偽**(False)

条件が真(True)か偽(False)かを判定するためには、**比較演算子**や**論理演算子**を使用します。

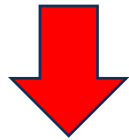


# 歯の本数を集計



**for** 日本語での**for文**に相当する部分は「**～ごとに繰り返す**」などのように言えます。

for **変数** in **要素の集まり**:  
変数ごとに繰り返す繰り返し実行するコード



for **行インデックス番号** in **(0,1,2...53023)** :

```
if klist.loc[インデックス番号, 'tooth record'] == '残存歯':  
    klist.loc[インデックス番号, 'tooth exist'] = 1  
else:  
    klist.loc[インデックス番号, 'tooth exist'] = 0
```

歯の本数を集計

tooth record		tooth exist	
インデックス0	残存歯	1	
1	残存歯	1	
2	残存歯	1	
3	喪失歯	0	
	残存歯	1	
	残存歯	1	
	残存歯	1	
	喪失歯	0	
	残存歯		
	残存歯		
	残存歯		
	喪失歯		
	喪失歯		

コード11-2

```
for i in range(53024)
```

```
    if klist.loc[i, 'tooth record'] == '残存歯':  
        klist.loc[i, 'tooth exist'] = 1  
    else:  
        klist.loc[i, 'tooth exist'] = 0
```

コード11-3

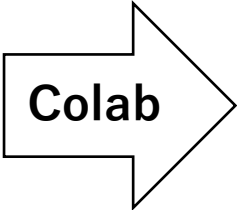
```
print(klist)
```

コード11-3 print(klist)

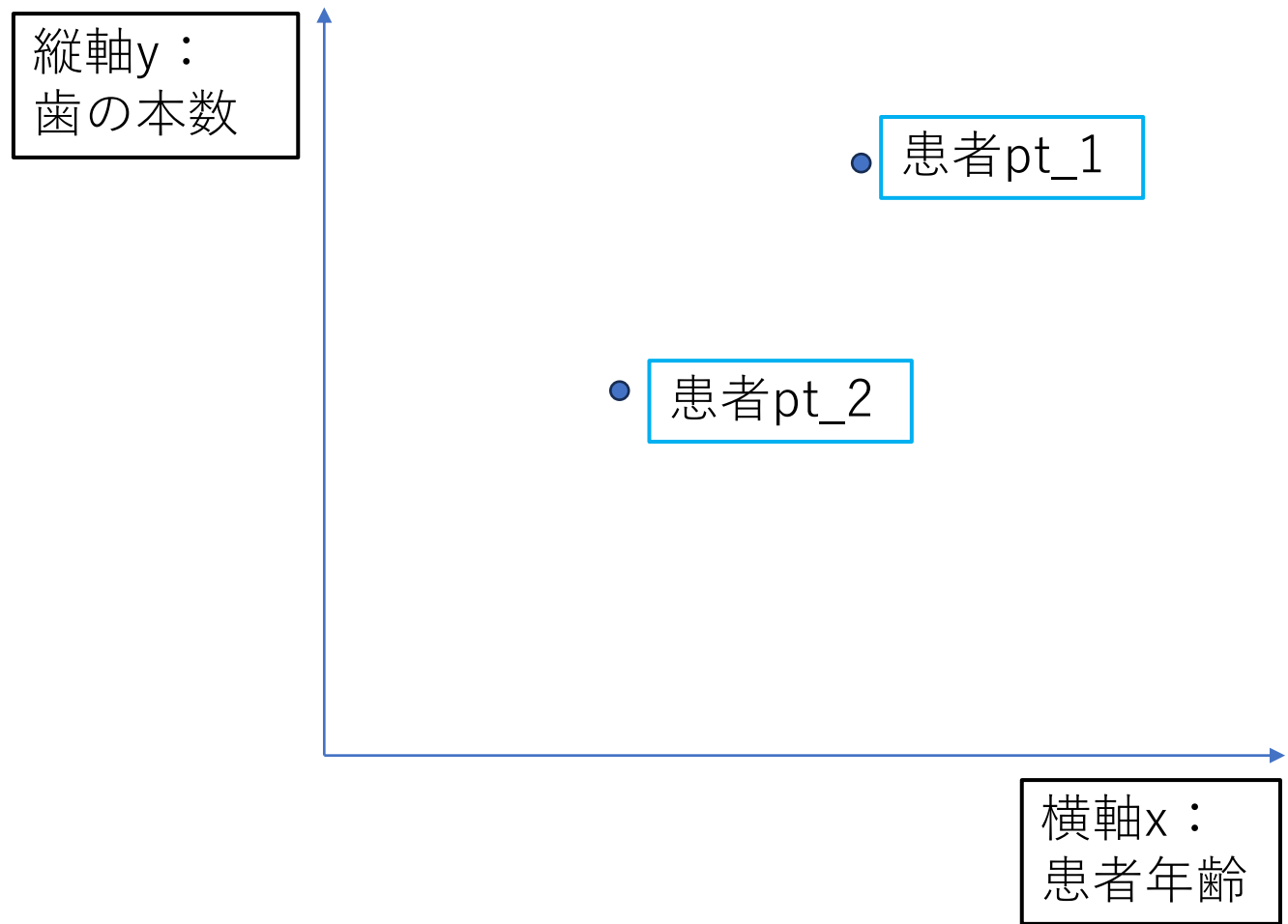
↗

	id	tooth	name	tooth	record	gender	age	tooth	exist
0	pt_1		A1	残存歯		男	77		1
1	pt_1		A2	喪失歯		男	77		0
2	pt_1		A3	残存歯		男	77		1
3	pt_1		A4	残存歯		男	77		1
4	pt_1		A5	残存歯		男	77		1
...	...		...	...		...		...	
53019	pt_1657		D4	喪失歯		女	40		0
53020	pt_1657		D5	残存歯		女	40		1
53021	pt_1657		D6	残存歯		女	40		1
53022	pt_1657		D7	喪失歯		女	40		0
53023	pt_1657		D8	残存歯		女	40		1

[53024 rows x 6 columns]



患者年齢と歯の本数の関係



	横軸x	縦軸y
患者pt_1	● 歳	● 本
患者pt_2	● 歳	● 本

id列	tooth exist列	age列
pt_1	1	77
	1	77
	0	77
	1	77
pt_2	1	65
	1	65
	1	65
	1	65
...		
pt_1657	1	40
	1	40
	1	40
	0	40

●本

●歳

Pandasの**groupbyメソッド**は、列を指定してデータをグループ化し、それぞれのグループに対して集計できます。

列Aでグループ化し、列Bを集計します。  
`klist.groupby('A')['B'].集計方法()`

横軸x

年齢の集計

A:id列

B:age列

集計方法：平均値

縦軸y

歯の本数の集計

A:id列

B:tooth exist列

集計方法：合計

id列	tooth exist列	age列
pt_1	1	77
	1	77
	0	77
	1	77
pt_2	1	65
	1	65
	1	65
	1	65
...		
pt_1657	1	40
	1	40
	1	40
	0	40

●本

●歳

Pandasの**groupbyメソッド**は、列を指定してデータをグループ化し、それぞれのグループに対して集計できます。

列Aでグループ化し、列Bを集計します。  
`klist.groupby('A')['B'].集計方法()`

横軸x      コード13-1

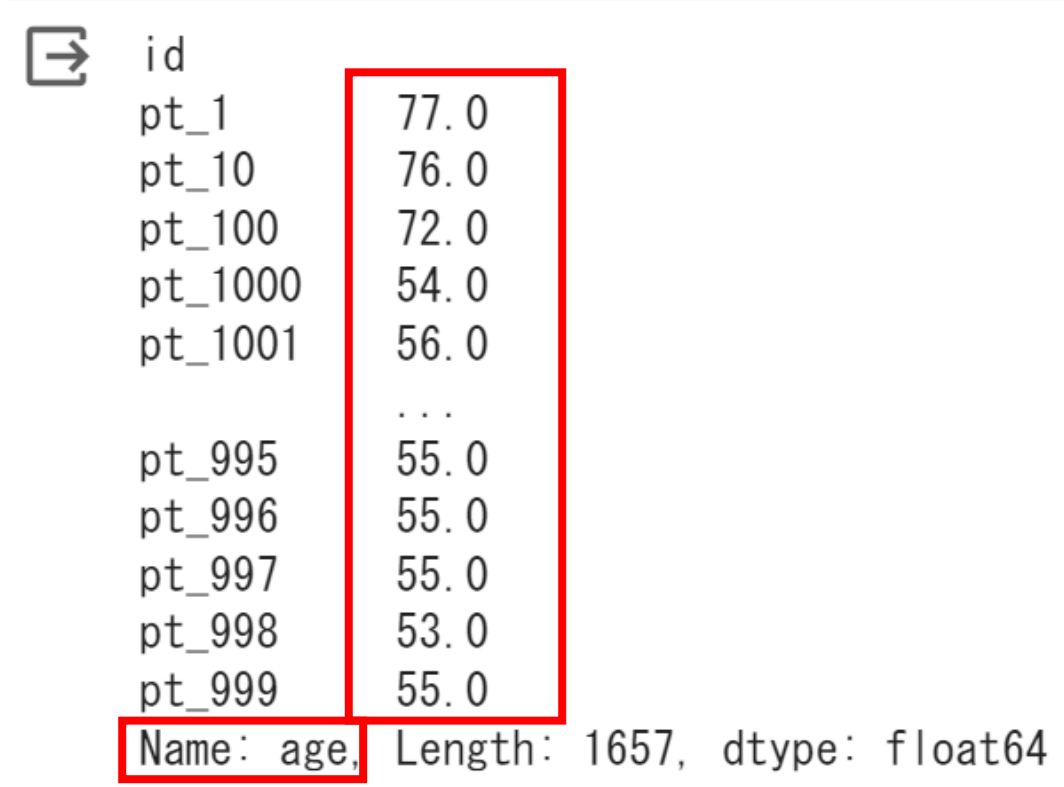
```
x = klist.groupby('id')['age'].mean()  
print(x)
```

縦軸y      コード13-2

```
y = klist.groupby('id')['tooth exist'].sum()  
print(y)
```

	横軸x	縦軸y
患者pt_1	●歳	●本
患者pt_2	●歳	●本

```
横軸x  コード13-1
x = klist.groupby('id')['age'].mean()
print(x)
```



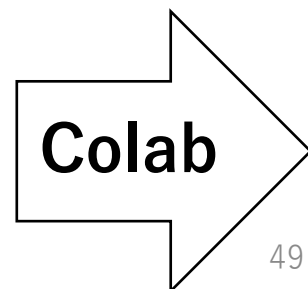


	横軸x	縦軸y
患者pt_1	●歳	●本
患者pt_2	●歳	●本

縦軸y    コード13-2

```
y = klist.groupby('id')['tooth exist'].sum()  
print(y)
```

```
id  
pt_1      24  
pt_10     25  
pt_100    26  
pt_1000   25  
pt_1001   29  
..  
pt_995    30  
pt_996    24  
pt_997    24  
pt_998    26  
pt_999    24  
Name: tooth exist, Length: 1657, dtype: object
```



## データクレンジングとデータの可視化

### 散布図を作成します

コード13-3

```
import numpy as np
x = np.array(x)
y = np.array(y)
```

コード13-4

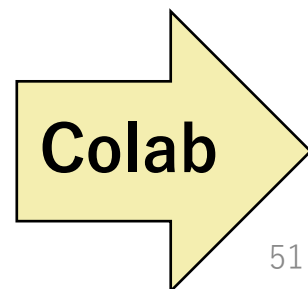
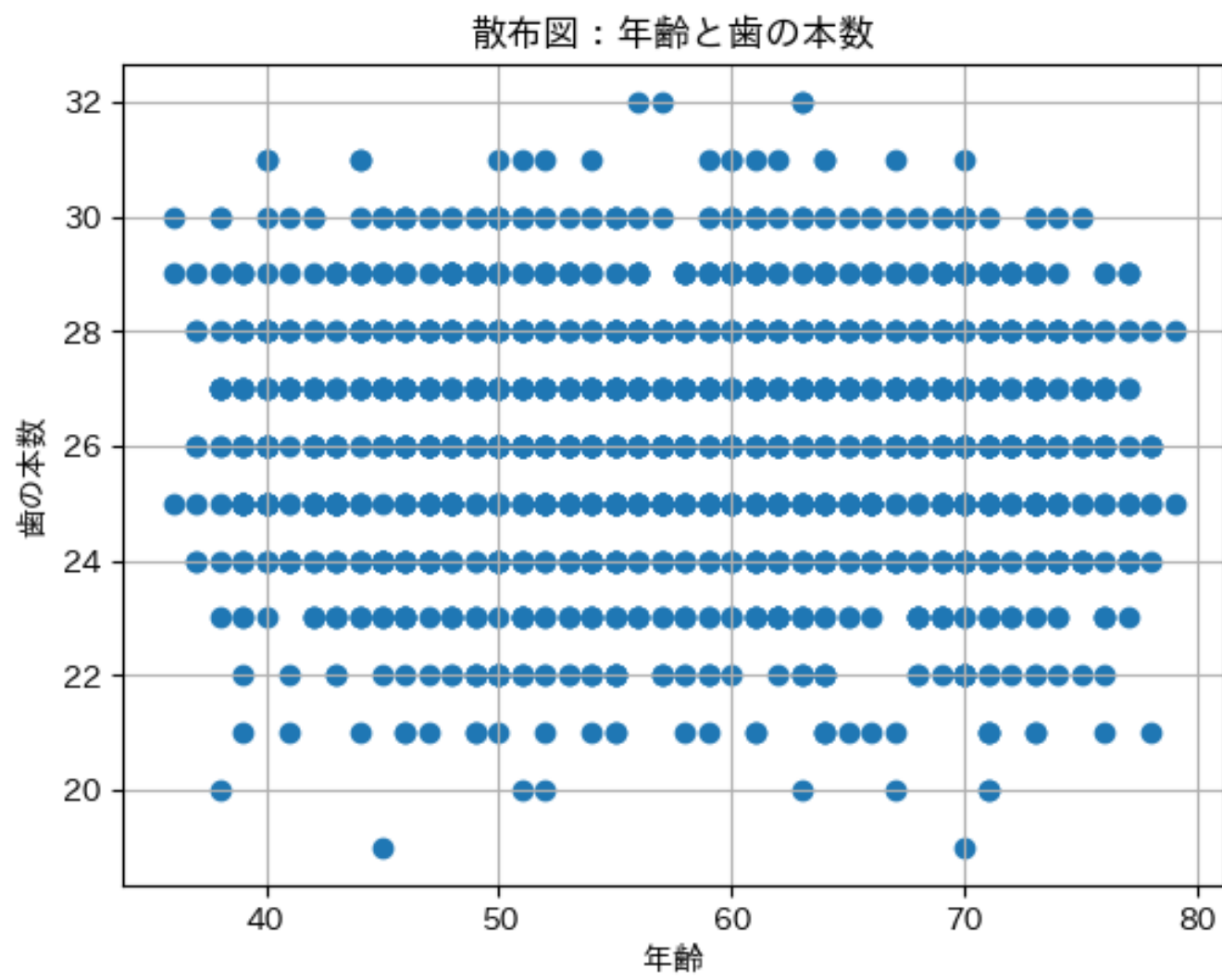
matplotlibライブラリと日本語入力方法をインポートします。

```
import matplotlib.pyplot as plt
!pip install japanize-matplotlib
import japanize_matplotlib
```

コード13-5

matplotlibライブラリで散布図を作成します。

```
plt.figure()
plt.title('散布図：年齢と歯の本数')
plt.xlabel('年齢')
plt.ylabel('歯の本数')
plt.grid()
plt.scatter(x, y)
plt.show()
```



# WS2回目の事前準備のご案内

WS2回目では画像データを使うため、事前準備のご協力をお願いします

① **webclass**にログインし、[IL2300919 医療系データサイエンス教育ワークショップ 2023年度版](#) のコースにアクセスしてください。「2回目の事前準備」をクリックしてください。

WebClass  
IL2300919 医療系データサイエンス教育ワークショップ 2023年度版

教材 マイレポート 成績 出席 その他 コース 管理者に戻る ログアウト

学生モード 解除

お知らせがあります。

タイムライン

Zoom情報 詳細  
資料

抄録  
New 医療系データサイエンス教育ワークショップ 抄録 詳細  
資料

事前課題  
事前課題 詳細  
資料

事前課題提出 詳細  
資料

New 2回目の事前準備 詳細  
資料 利用回数 1

Top

# 事前準備

## ② 第1章 「WS2回目\_事前準備」 に内容が書かれていますので、この通りに準備をお願いします



教材 資料を閉じる

> IL2300919 医療系データサイエンス教育ワークショップ 2023年度版  
2回目の事前準備

石丸 美穂 さんがログイン中

前のページ 次のページ

目次を隠す 資料を閉じる

第1節 2回目の事前準備について	1	
第1章 WS2回目_事前準備	2	<a href="#">添付資料</a>
第1節 Images_TMDU_2023.zip	3	<a href="#">添付資料</a>
第2節 WS2回目_事前準備.ipynb	4	<a href="#">添付資料</a>

表示に問題があるときは [別ウインドウ](#) で開いてください。

1 / 13 自動ズーム

### 事前準備

2回目では画像データを使うため、webclassから画像ファイル (image\_TMDU\_2023.zip)をダウンロードしてください。

ダウンロードしたzipファイルを展開(解凍)せずに、そのまま自分の Google Driveにアップロードしてください。

ご清聴をありがとうございます。

アンケートのご協力をお願いいたします。



Webclass QRコード

## IL2300919 医療系データサイエンス教育ワークショップ 2023年度版

### 第1回演習後アンケート

2月22日演習後に回答お願いいたします。

### 全体演習後アンケート

最終回までにご回答をお願いいたします。