

演習授業中の質問対応について

Zoom ミーティング

表示

ミーティング チャット

演習授業中の質問をチューターの先生方が対応させていただきます。

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

ここにメッセージは誰に表示されますか？

宛先: **全員** v

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 2

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



演習4

Python基礎 モジュール、パッケージ、ライブラリ

統合教育機構 曹 日丹

医療とAI・ビッグデータ入門

演習2-7の構成

Python基礎を学びましょう

演習2

Pythonの変数とデータの型

演習3

プログラミング基礎

演習4 12/7 10:40-11:35

モジュール、パッケージ、ライブラリ

Pythonを使ってみましょう

演習5 12/7 11:35-12:20

患者の歯に関する~~病院のリアルデータ~~の説明

架空データ

演習6

データクレンジングに必要なライブラリ（Pandas）の応用

演習7

データクレンジングとデータの可視化

モジュール Pythonのコード（**変数、関数、クラス**など）を含むファイルです

パッケージ

ライブラリ

モジュール Pythonのコード（**変数、関数、クラス**など）を含むファイルです

パッケージ 複数のモジュールをグループ化します。

モジュール

モジュール

モジュール

モジュール

モジュール

ライブラリ

Python基礎 モジュール、パッケージ、ライブラリ

モジュール Pythonのコード（**変数、関数、クラス**など）を含むファイルです

パッケージ 複数のモジュールをグループ化します。

モジュール

モジュール

モジュール

モジュール

モジュール

ライブラリ 複数のモジュールやパッケージで構成されています。

モジュール

モジュール Pythonのコード（**変数、関数、クラス**など）を含むファイルです。

拡張子： .py

モジュールを作ってみましょう

年末年始挨拶の言葉をモジュールにしたいです。

Merry Christmas!

関数

モジュール

Happy New Year!

関数

モジュールを作ってみましょう

年末年始挨拶の言葉をモジュールにしたいです。

コード：

```
def dec():  
    return "Merry Christmas!"
```

関数

コード：

```
def jan():  
    return "Happy New Year!"
```

関数

モジュール

モジュールを作ってみましょう

年末年始挨拶の言葉をモジュールにしたいです。

コード：

コマンド

ファイル名

```
%%writefile greetingM.py
```

```
def dec():  
    return "Merry Christmas!"
```

```
def jan():  
    return "Happy New Year!"
```

%%writefileとは：

新しいモジュールを、名前を指定して作成するコマンドです。

モジュール

モジュールを作ってみましょう

年末年始挨拶の言葉をモジュールにしたいです。

```
%%writefile greetingM.py
```

```
def dec():  
    return "Merry Christmas!"
```

```
def jan():  
    return "Happy New Year!"
```

 Writing greetingM.py



モジュールを作ってみましょう

年末年始挨拶の言葉をモジュールにしたいです。

関数を呼び出します

コード：

```
import greetingM
```

モジュール名：greetingM

モジュールを作ってみましょう

年末年始挨拶の言葉をモジュールにしたいです。

関数を呼び出します

モジュール名.関数名 ()

コード：

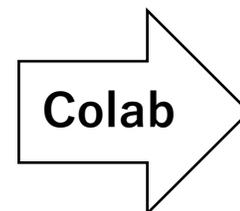
```
print(greetingM.dec())
```

➡ 'Merry Christmas!' greetingのdecを呼び出しました。

コード：

```
print(greetingM.jan())
```

➡ 'Happy New Year!' greetingのjanを呼び出しました。



検索google colab Colaboratory へようこそ - Colaboratory - Google


Colaboratory へようこそ

ファイル
編集
表示

☰ 目次

🔍 はじめに

{x} 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

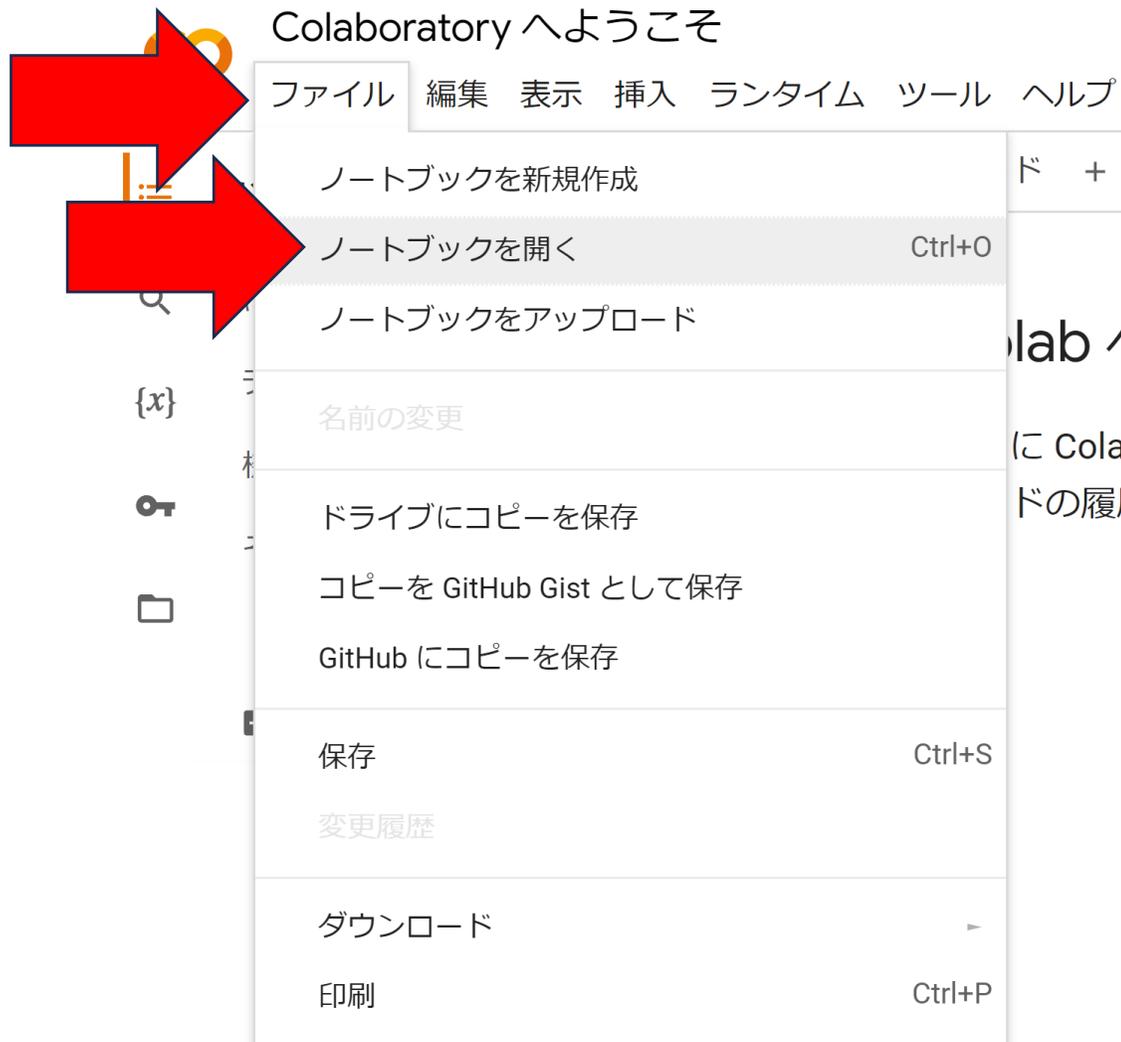
ノートブックを開く

🔄

	タイトル	最終閲覧 ▲	最初に開いた日 時 ▼	
例 >				
最近 >				
Google ドライブ >	 演習2コード	9:43	10月25日	 
GitHub >	 Colaboratory へようこそ	9:21	2022年12月23日	
アップロード >	 演習3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	 

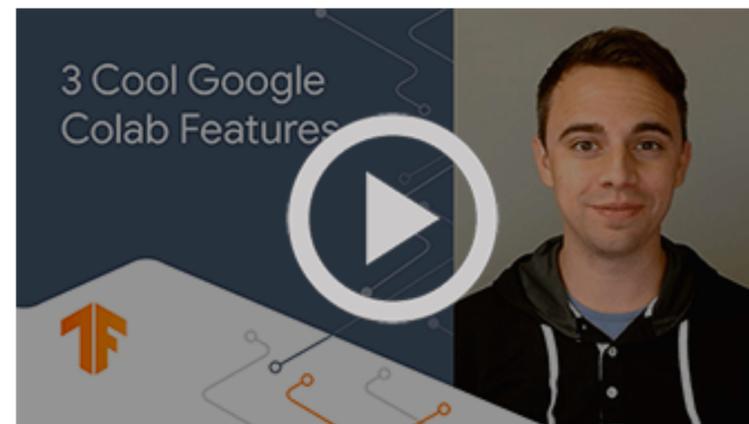
+ ノートブックを新規作成
キャンセル

検索google colab Colaboratory へようこそ - Colaboratory - Google



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

検索google colab Colaboratory へようこそ - Colaboratory - Google

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 ▼		
演習4コード.ipynb					
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

検索google colab Colaboratory へようこそ - Colaboratory - Google

ノートブックを開く

例 >

最近 >

Google ドラ
イブ >

GitHub >

アップロード >



参照

または、ここにファイルをドラッグしてください

演習4コード.ipynb

演習授業中の質問対応について

Zoom ミーティング

表示

ミーティング チャット

演習授業中の質問をチューターの先生方が対応させていただきます。

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

ここにメッセージは誰に表示されますか？

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



モジュール Pythonのコード（**変数、関数、クラス**など）を含むファイルです

パッケージ 複数のモジュールから構成されています。

モジュール

モジュール

モジュール

モジュール

モジュール

パッケージを作ってみましょう

パッケージ

パッケージは階層構造を持っています。

📁 giftP

📄 __init__.py

📄 greetingM.py モジュール1: greetingM.py

📄 presentM.py モジュール2: presentM .py

パッケージ

パッケージは階層構造を持っています。

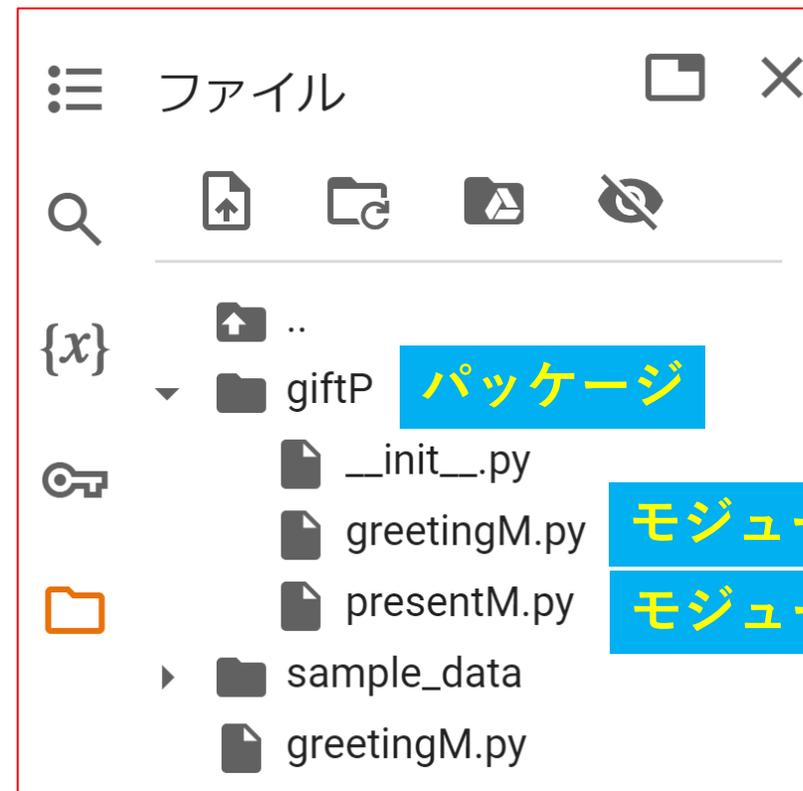
```
└─ giftP
   ├── __init__.py
   ├── greetingM.py
   └── presentM.py
```

コード：

```
!mkdir giftP          #ディレクトリを作成します
!touch giftP/__init__.py # 初期化ファイル
!touch giftP/greetingM.py # モジュール1
!touch giftP/presentM.py # モジュール2
```

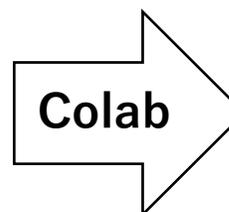
パッケージ

パッケージは階層構造を持っています。



コード：パッケージを作成します。

```
!mkdir giftP          #ディレクトリを作成します
!touch giftP/__init__.py # 初期化ファイル
!touch giftP/greetingM.py # モジュール1
!touch giftP/presentM.py # モジュール2
```



パッケージ

階層構造を持っています。

- folder giftP
 - file __init__.py
 - file greetingM.py
 - file presentM.py

コマンド

```
%%writefile
```

コマンド

```
%%writefile
```

パッケージ

階層構造を持っています。

📁 giftP

- 📄 __init__.py
- 📄 greetingM.py
- 📄 presentM.py

パッケージ名/モジュール名.py

```
%%writefile giftP/greetingM.py
```

パッケージ名/モジュール名.py

```
%%writefile giftP/presentM.py
```

パッケージ

階層構造を持っています。

folder giftP

file __init__.py

file greetingM.py

file presentM.py

モジュール内容

モジュール内容

コード :

```
%%writefile giftP/greetingM.py
def dec():
    return "Merry Christmas!"
def jan():
    return "Happy New Year!"
```

コード :

```
%%writefile giftP/presentM.py
def dec():
    return "a strawberry cake!"
def jan():
    return "otoshidama!"
```

パッケージ



```
from パッケージ名 import モジュール名
```

パッケージのモジュール1をインポートします

コード：

```
from giftP import greetingM
```

パッケージのモジュール2をインポートします

コード：

```
from giftP import presentM
```

パッケージ

階層構造を持っています。

```
└─ giftP
   ├── __init__.py
   ├── greetingM.py
   └── presentM.py
```

コード：

モジュールを呼び出します

```
from giftP import greetingM
```

```
    print(greetingM.dec())
```

```
⇒ 'Merry Christmas!'
```

```
    print(greetingM.jan())
```

```
⇒ 'Happy New Year!'
```

パッケージ

階層構造を持っています。



コード：

モジュールを呼び出します

```
from giftP import greetingM
```

```
print(greetingM.dec())
```

```
↳ 'Merry Christmas!'
```

```
print(greetingM.jan())
```

```
↳ 'Happy New Year!'
```

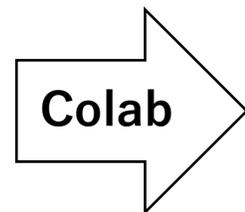
```
from giftP import presentM
```

```
print(presentM.dec())
```

```
↳ 'a strawberry cake'
```

```
print(presentM.jan())
```

```
↳ 'an otoshidama'
```



既に存在しているモジュール

Pythonには多くの標準モジュールがあります。
一般的なモジュールの例：

- **math** - 数学的な関数や定数を提供します。
- **datetime** - 日付と時刻を操作するためと関数などを提供します。

既存しているモジュール math

モジュールをインポートします。

コード：

```
import math
```

コード：

```
print(math.pi)
```

mathの属性

円周率 π の値を取得します。

コード：

```
print(math.e)
```

mathの属性

自然対数の底を取得します。

既存しているモジュール math

モジュールをインポートします。

コード：

```
import math
```

コード：

```
print(math.pi)
```

mathの属性

円周率 π の値を取得します。

→ 3.141592653589793

コード：

```
print(math.e)
```

mathの属性

自然対数の底を取得します。

→ 2.718281828459045

Colab

既存しているモジュール math

mathの関数： `math.sqrt()`

平方根を計算します。 `math.sqrt(引数)`

コード：

```
print(math.sqrt(16))
```

既存しているモジュール math

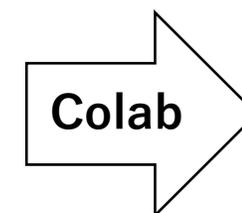
mathの関数： `math.sqrt()`

平方根を計算します。 `math.sqrt(引数)`

コード：

```
print(math.sqrt(16))
```

 4.0



既存しているモジュール math

mathの関数： $\log(x, \text{base})$

底が base である x の対数を計算します。

$\log(x, \text{base})$

コード：

```
x = 1000.0
base = 10.0

print(math.log(x, base))
```

既存しているモジュール math

mathの関数： $\log(x, \text{base})$

底が base である x の対数を計算します。 $\log(x, \text{base})$

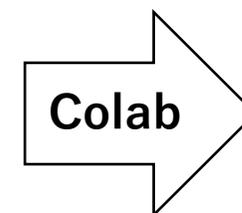
コード：

```
x = 1000.0
base = 10.0

print(math.log(x, base))
```

 2.9999999999999996

計算結果は 2.99999999999999962.9999999999999996 となりました。これは理論値である 33 に非常に近い値です。このわずかな差異は、コンピュータが浮動小数点数を扱う際に生じる誤差によるものです。



Pythonは豊富なライブラリを持っています。
いくつかの主要なPythonライブラリとその概要を示しています。

NumPy:

数値計算ライブラリ。
効率的な数値計算のための多次元配列と、それを操作するツールを提供しています。

Pandas:

データ分析ライブラリ。
DataFrameという強力なデータ構造を用いて、データの操作や分析を行うことができます。

Matplotlib:

データの可視化ライブラリ。
グラフの作成・描画ができます。

Scikit-learn:

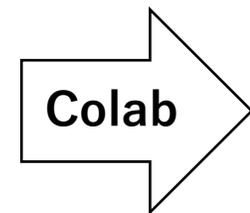
機械学習ライブラリ。
分類、回帰、クラスタリングなど、様々な機械学習アルゴリズムの実装が含まれています。

Numpy-ライブラリ

numpyをインポートする必要があります。

```
import numpy as np
```

npはnumpyの略称です。



numpyのarray関数

1次元配列の作成

1 2 3 4 5

2次元配列の作成

**1 2 3
4 5 6
7 8 9
10 11 12**

4列、3行

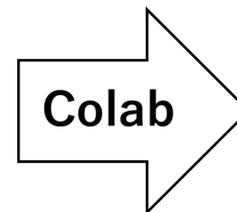
numpyのarray関数

```
np.array(リスト)  
print(np.array(リスト))
```

1次元配列の作成：リストから作成します

コード：

```
list1 = [1, 2, 3, 4, 5]  
  
print(list1)  
print(np.array([1, 2, 3, 4, 5]))
```



numpyのarray関数

```
np.array(リスト)  
print(np.array(リスト))
```

1次元配列の作成：リストから作成します

コード：

```
list1 = [1, 2, 3, 4, 5]  
  
print(list1)  
print(np.array([1, 2, 3, 4, 5]))
```

⇒ [1, 2, 3, 4, 5]
[1 2 3 4 5]

numpyのarray関数

```
np.array(リスト)  
print(np.array(リスト))
```

2次元配列(行列)の作成： リストから作成します

コード：

```
list2 = [[1, 2, 3], [4, 5, 6, ], [7, 8, 9], [10, 11, 12]]  
         [リスト1, リスト2, リスト3, リスト4]  
         [1行目, 2行目, 3行目, 4行目]
```

```
print(list2)  
print(np.array([[1, 2, 3], [4, 5, 6, ], [7, 8, 9], [10, 11, 12]]))
```

numpyのarray関数

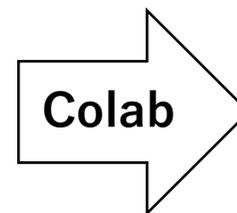
```
np.array(リスト)  
print(np.array(リスト))
```

2次元配列(行列)の作成： リストから作成します

コード：

```
list2 = [[1,2,3], [4,5,6,], [7,8,9], [10,11,12]]  
print(list1)  
print(np.array([[1,2,3], [4,5,6,], [7,8,9], [10,11,12]]))
```

```
→ [[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]  
[[ 1  2  3]  
 [ 4  5  6]  
 [ 7  8  9]  
 [10 11 12]]
```



属性

配列名.shape 形状を返します

配列名.size 総要素数を返します

配列名.ndim 次元数を返します

```
print(arr.shape) # 形状を表示
print(arr.size) # 総要素数を表示
print(arr.ndim) # 次元数を表示
```

コード:

```
arr1 = np.array([1, 2, 3, 4, 5])

print(arr1)
print(arr1.shape) # 形状を表示
print(arr1.size) # 総要素数を表示
print(arr1.ndim) # 次元数を表示
```

属性

配列名.shape 形状を返します

配列名.size 総要素数を返します

配列名.ndim 次元数を返します

```
print(arr.shape) # 形状を表示
print(arr.size) # 総要素数を表示
print(arr.ndim) # 次元数を表示
```

コード:

```
arr1 = np.array([1, 2, 3, 4, 5])
```

```
print(arr1)
```

```
print(arr1.shape) # 形状を表示
```

```
print(arr1.size) # 総要素数を表示
```

```
print(arr1.ndim) # 次元数を表示
```



[1 2 3 4 5]

(5,)

5

1

属性

配列名.shape 形状を返します

配列名.size 総要素数を返します

配列名.ndim 次元数を返します

```
print(arr.shape) # 形状を表示
print(arr.size) # 総要素数を表示
print(arr.ndim) # 次元数を表示
```

コード：

```
arr1 = np.array([1, 2, 3, 4, 5])
print(arr1)
print(arr1.shape) # 形状を表示
print(arr1.size) # 総要素数を表示
print(arr1.ndim) # 次元数を表示
```



```
[1 2 3 4 5]
(5, )
5
1
```

NumPyなどの配列操作ライブラリでの次元の表現方法に基づいています。

形状が (5,) であるということは、配列が5つの要素を持つ一次元配列であることを意味します。これは、2次元以上の配列とは異なり、行や列の概念が適用されないため、ただ一つの数値で要素数を示しています。

属性

配列名.shape 形状を返します

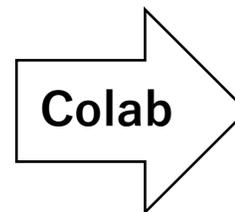
配列名.size 総要素数を返します

配列名.ndim 次元数を返します

コード：

```
arr2 = np.array([[1, 2, 3], [4, 5, 6]])

print(arr2)
print(arr2.shape)      # 形状を表示
print(arr2.size)      # 総要素数を表示
print(arr2.ndim)      # 次元数を表示
```



属性

配列名.shape 形状を返します

配列名.size 総要素数を返します

配列名.ndim 次元数を返します

コード：

```
arr2 = np.array([[1, 2, 3], [4, 5, 6]])  
  
print(arr2)  
print(arr2.shape)      # 形状を表示  
print(arr2.size)      # 総要素数を表示  
print(arr2.ndim)      # 次元数を表示
```

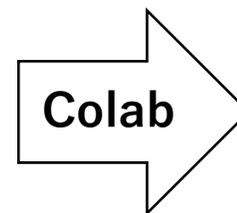


```
[[1 2 3]  
 [4 5 6]]
```

```
(2, 3)
```

```
6
```

```
2
```



配列操作メソッド 配列名.reshape(): 配列の形状を変更します。

1次元配列を2次元配列に変換します。

配列名.reshape(行の数、列の数)

2次元配列を1次元配列に変換します。

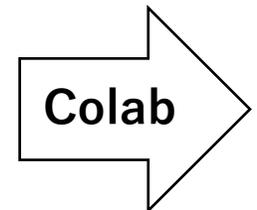
配列名.reshape(-1)

1次元配列を2次元配列に変換します。

コード:

```
arr2 = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])  
  
print(arr2)  
print(arr2.reshape(3, 3))
```

```
→ [1 2 3 4 5 6 7 8 9]  
   [[1 2 3]  
    [4 5 6]  
    [7 8 9]]
```

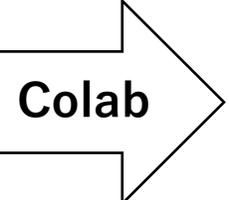


2次元配列を1次元配列に変換します。

コード:

```
arr3 = np.array([[1, 2], [3, 4], [5, 6], [7, 8]])  
  
print(arr3)  
print(arr3.reshape(-1))
```

```
→ [[1 2]  
    [3 4]  
    [5 6]  
    [7 8]]  
[1 2 3 4 5 6 7 8]
```

A white arrow pointing to the right with the word "Colab" written inside in black text.

統計的計算メソッド:

.sum(): 配列内の要素の合計を計算します。

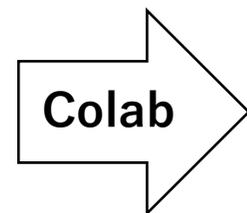
.mean(): 配列の平均値を計算します。

```
arr3 = np.array([[1, 2], [3, 4], [5, 6], [7, 8]])
```

コード:

```
print(arr3.sum())
```

```
print(arr3.mean())
```



統計的計算メソッド:

.sum(): 配列内の要素の合計を計算します。

.mean(): 配列の平均値を計算します。

```
arr3 = np.array([[1, 2], [3, 4], [5, 6], [7, 8]])
```

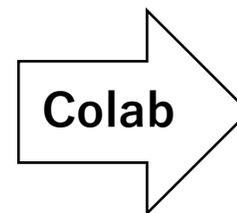
コード:

```
print(arr3.sum())
```

→ 36

```
print(arr3.mean())
```

→ 4.5



WebClassで課題を提出してください、締め切りは12月21日23:59までです。

課題1：

Numpyライブラリをインポートするコードを書いてください。

Numpyの略称は「np」とします。

課題2：

print関数を使って、下記2つのリストから2次元配列を作成するコードを書いてください。

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

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

演習授業中の質問対応について

Zoom ミーティング

表示

ミーティング チャット

演習授業中の質問をチューターの先生方が対応させていただきます。

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

ここにメッセージは誰に表示されますか？

宛先: **全員** v

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 2

画面共有

リアクション

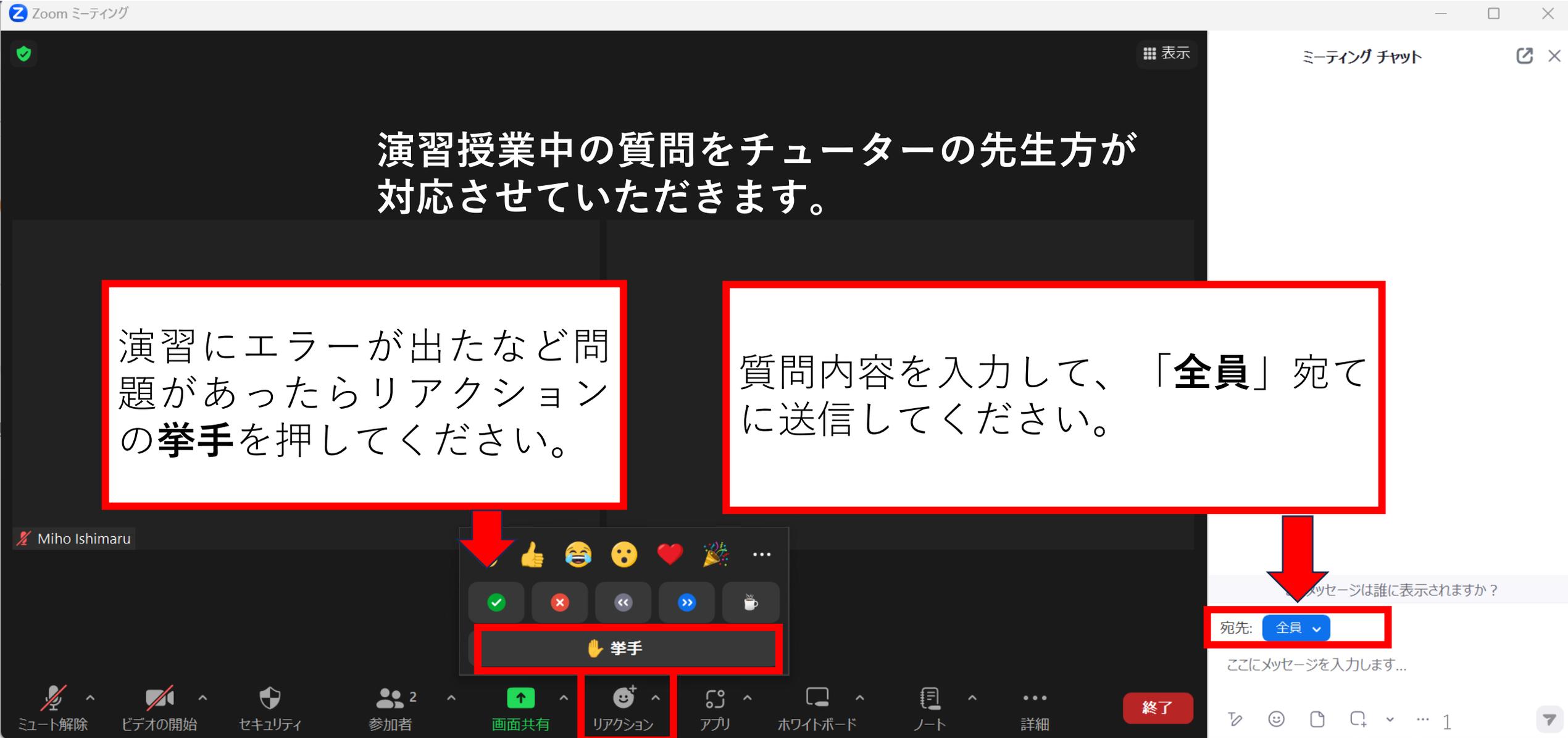
アプリ

ホワイトボード

ノート

詳細

終了



演習5

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

統合教育機構 曹 日丹

医療とAI・ビッグデータ入門

演習2-7の構成

Python基礎を学びましょう

演習2

Pythonの変数とデータの型

演習3

プログラミング基礎

演習4 12/7 10:40-11:35

モジュール、パッケージ、ライブラリ

Pythonを使ってみましょう

演習5 12/7 11:35-12:20

患者の歯に関する病院のリアルデータの説明

架空データ

演習6

データクレンジングに必要なライブラリ（Pandas）の応用

演習7

データクレンジングとデータの可視化

Pythonは豊富なライブラリを持っています。
いくつかの主要なPythonライブラリとその概要を示しています。

NumPy:

数値計算ライブラリ。

Pandas:

データ分析ライブラリ。

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。

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。

Matplotlib:

データの可視化ライブラリ。

Scikit-learn:

機械学習ライブラリ。

Pandas-ライブラリ

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。

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。

任意の列名を付けることができます。 **列名**

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

	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

Pandas-ライブラリ データフレーム 操作の例

サイズの変更が可能：列の追加や削除ができます。

データの操作と分析：集計など様々なデータ操作ができます。

データの読み込みと書き出し：

さまざまなファイル形式（CSV、Excel）で、データを読み込んだり、データを書き出したりできます。

The diagram shows a table representing a DataFrame. The columns are labeled 'ID', '名前', '年齢', and 'BMI'. The rows are indexed from 0 to 3. A red bracket above the table spans the columns from 'ID' to 'BMI', with the label '列名' (Column Name) above it. A red bracket to the left of the table spans the rows from index 0 to 3, with the label 'インデックス' (Index) to its left.

	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

Pandas-ライブラリ

Pandas-ライブラリのデータフレーム機能を使ってみましょう。
Pandasライブラリをインポートする必要があります。

コード01

```
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。)

検索google colab Colaboratory へようこそ - Colaboratory - Google

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 >

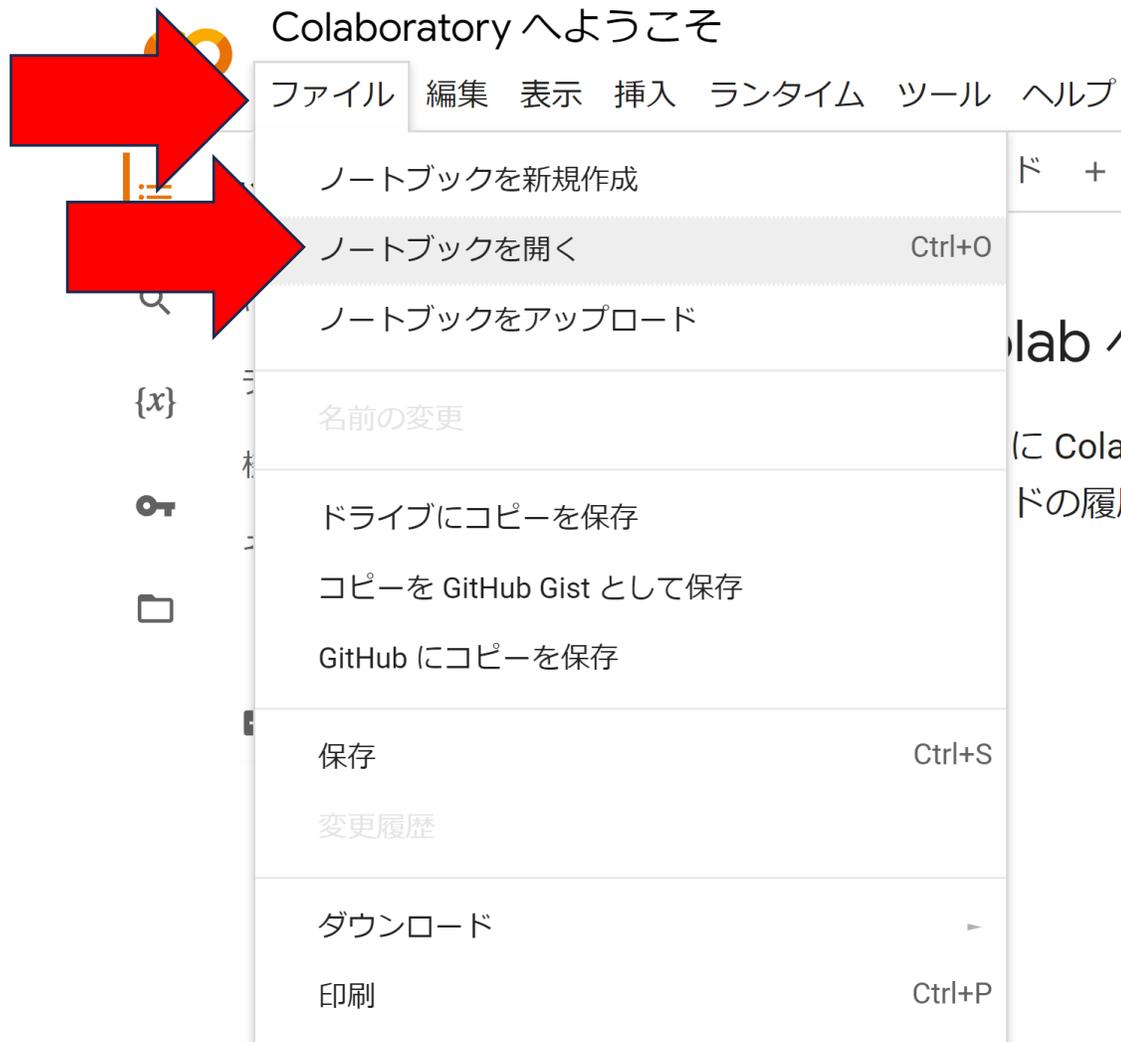
検索: ノートブックを検索

タイトル	最終閲覧	最初に開いた日時	
演習2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3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

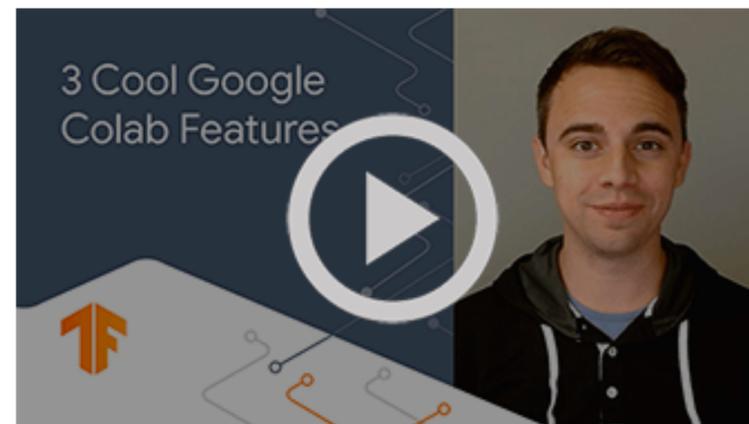
キャンセル

検索google colab Colaboratory へようこそ - Colaboratory - Google



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

検索google colab Colaboratory へようこそ - Colaboratory - Google

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 ▼		
演習5コード.ipynb					
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

検索google colab Colaboratory へようこそ - Colaboratory - Google

ノートブックを開く

例 >

最近 >

Google ドラ
イブ >

GitHub >

アップロード >



参照

または、ここにファイルをドラッグしてください

演習5コード.ipynb

演習授業中の質問対応について

Zoom ミーティング

表示

ミーティング チャット

演習授業中の質問をチューターの先生方が対応させていただきます。

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

ここにメッセージは誰に表示されますか？

宛先: **全員** v

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 2

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

13

架空データ

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。

絞り込み条件：

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：

初診日：2018/1/1-2022/12/31

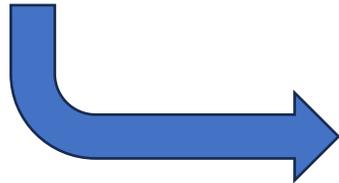
口腔内全ての永久歯

歯の状態：残存歯、喪失歯

性別

Anonymized id 匿名化ID

birthmonth 生年月



リアルワールドデータを元にして、一部を無作為にシャッフルして、**架空データ**を作成し、練習用データとしました。

架空データ

初診日

初診日：
患者さん初めて病院
で受診しに来た日付。

	A	B	C	D	E	F
1	tooth name ▼	tooth record ▼	gender ▼	anonymized id ▼	birthmonth ▼	yearmonth ▼
2	A1	残存歯	男	pt_1	Jan-43	2020/10
3	A2	残存歯	男	pt_1	Jan-43	2020/10
4	A3	残存歯	男	pt_1	Jan-43	2020/10
5	A4	残存歯	男	pt_1	Jan-43	2020/10
6	A5	残存歯	男	pt_1	Jan-43	2020/10
7	A6	残存歯	男	pt_1	Jan-43	2020/10
8	A7	残存歯	男	pt_1	Jan-43	2020/10
9	A8	残存歯	男	pt_1	Jan-43	2020/10
10	B1	残存歯	男	pt_1	Jan-43	2020/10
11	B2	喪失歯	男	pt_1	Jan-43	2020/10
12	B3	残存歯	男	pt_1	Jan-43	2020/10
13	B4	残存歯	男	pt_1	Jan-43	2020/10
14	B5	残存歯	男	pt_1	Jan-43	2020/10
15	B6	残存歯	男	pt_1	Jan-43	2020/10
16	B7	残存歯	男	pt_1	Jan-43	2020/10
17	B8	残存歯	男	pt_1	Jan-43	2020/10
18	C1	残存歯	男	pt_1	Jan-43	2020/10
19	C2	残存歯	男	pt_1	Jan-43	2020/10
20	C3	残存歯	男	pt_1	Jan-43	2020/10
21	C4	喪失歯	男	pt_1	Jan-43	2020/10
22	C5	喪失歯	男	pt_1	Jan-43	2020/10
23	C6	残存歯	男	pt_1	Jan-43	2020/10

架空データ

生年月

	A	B	C	D	E	F
1	tooth name ▼	tooth record ▼	gender ▼	anonymized id ▼	birthmonth ▼	yearmonth ▼
2	A1	残存歯	男	pt_1	Jan-43	1943年1月生まれ
3	A2	残存歯	男	pt_1	Jan-43	2020/10
4	A3	残存歯	男	pt_1	Jan-43	2020/10
5	A4	残存歯	男	pt_1	Jan-43	2020/10
6	A5	残存歯	男	pt_1	Jan-43	2020/10
7	A6	残存歯	男	pt_1	Jan-43	2020/10
8	A7	残存歯	男	pt_1	Jan-43	2020/10
9	A8	残存歯	男	pt_1	Jan-43	2020/10
10	B1	残存歯	男	pt_1	Jan-43	2020/10
11	B2	喪失歯	男	pt_1	Jan-43	2020/10
12	B3	残存歯	男	pt_1	Jan-43	2020/10
13	B4	残存歯	男	pt_1	Jan-43	2020/10
14	B5	残存歯	男	pt_1	Jan-43	2020/10
15	B6	残存歯	男	pt_1	Jan-43	2020/10
16	B7	残存歯	男	pt_1	Jan-43	2020/10
17	B8	残存歯	男	pt_1	Jan-43	2020/10
18	C1	残存歯	男	pt_1	Jan-43	2020/10
19	C2	残存歯	男	pt_1	Jan-43	2020/10
20	C3	残存歯	男	pt_1	Jan-43	2020/10
21	C4	喪失歯	男	pt_1	Jan-43	2020/10
22	C5	喪失歯	男	pt_1	Jan-43	2020/10
23	C6	残存歯	男	pt_1	Jan-43	2020/10

架空データ

患者ID（匿名化済）

	A	B	C	D	E	F
1	tooth name ▼	tooth record ▼	gender ▼	anonymized id ▼	birthmonth ▼	yearmonth ▼
2	A1	残存歯	男	pt_1	ptはpatient（患者）の略称	
3	A2	残存歯	男	pt_1		
4	A3	残存歯	男	pt_1	Jan-43	2020/10
5	A4	残存歯	男	pt_1	Jan-43	2020/10
6	A5	残存歯	男	pt_1	Jan-43	2020/10
7	A6	残存歯	男	pt_1	Jan-43	2020/10
8	A7	残存歯	男	pt_1	Jan-43	2020/10
9	A8	残存歯	男	pt_1	Jan-43	2020/10
10	B1	残存歯	男	pt_1	Jan-43	2020/10
11	B2	喪失歯	男	pt_1	Jan-43	2020/10
12	B3	残存歯	男	pt_1	Jan-43	2020/10
13	B4	残存歯	男	pt_1	Jan-43	2020/10
14	B5	残存歯	男	pt_1	Jan-43	2020/10
15	B6	残存歯	男	pt_1	Jan-43	2020/10
16	B7	残存歯	男	pt_1	Jan-43	2020/10
17	B8	残存歯	男	pt_1	Jan-43	2020/10
18	C1	残存歯	男	pt_1	Jan-43	2020/10
19	C2	残存歯	男	pt_1	Jan-43	2020/10
20	C3	残存歯	男	pt_1	Jan-43	2020/10
21	C4	喪失歯	男	pt_1	Jan-43	2020/10
22	C5	喪失歯	男	pt_1	Jan-43	2020/10
23	C6	残存歯	男	pt_1	Jan-43	2020/10

架空データ

歯の状態

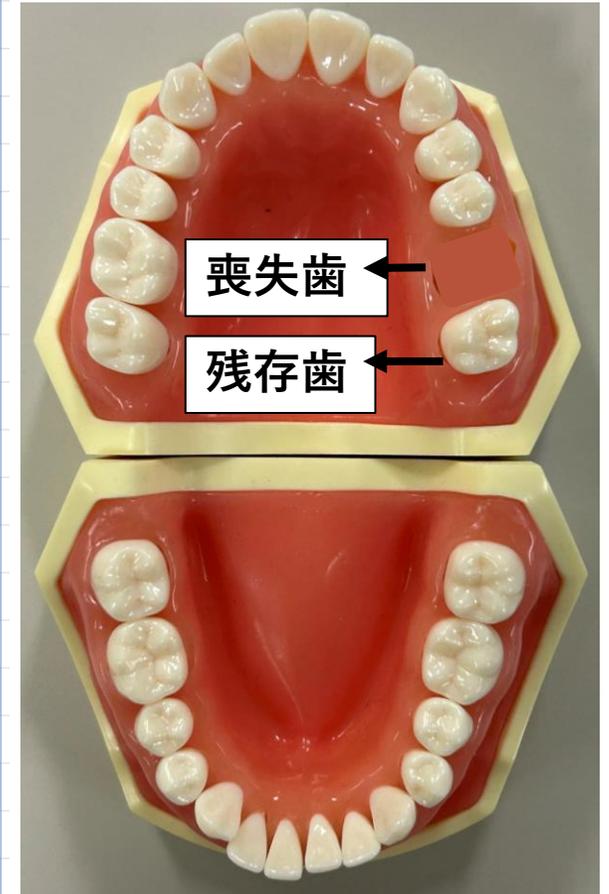
	A	B	C	D	E	F
1	tooth name ▼	tooth record ▼	gender ▼	anonymized id ▼	birthmonth ▼	yearmonth ▼
2	A1	残存歯	男	pt_1	Jan-43	2020/10
3	A2	残存歯	男	pt_1	Jan-43	2020/10
4	A3	残存歯	男	pt_1	Jan-43	2020/10
5	A4	残存歯	男	pt_1	Jan-43	2020/10
6	A5	残存歯	男	pt_1	Jan-43	2020/10
7	A6	残存歯	男	pt_1	Jan-43	2020/10
8	A7	残存歯	男	pt_1	Jan-43	2020/10
9	A8	残存歯	男	pt_1	Jan-43	2020/10
10	B1	残存歯	男	pt_1	Jan-43	2020/10
11	B2	喪失歯	男	pt_1	Jan-43	2020/10
12	B3	残存歯	男	pt_1	Jan-43	2020/10
13	B4	残存歯	男	pt_1	Jan-43	2020/10
14	B5	残存歯	男	pt_1	Jan-43	2020/10
15	B6	残存歯	男	pt_1	Jan-43	2020/10
16	B7	残存歯	男	pt_1	Jan-43	2020/10
17	B8	残存歯	男	pt_1	Jan-43	2020/10
18	C1	残存歯	男	pt_1	Jan-43	2020/10
19	C2	残存歯	男	pt_1	Jan-43	2020/10
20	C3	残存歯	男	pt_1	Jan-43	2020/10
21	C4	喪失歯	男	pt_1	Jan-43	2020/10
22	C5	喪失歯	男	pt_1	Jan-43	2020/10
23	C6	残存歯	男	pt_1	Jan-43	2020/10

残存歯：

口腔内に存在している歯

喪失歯：

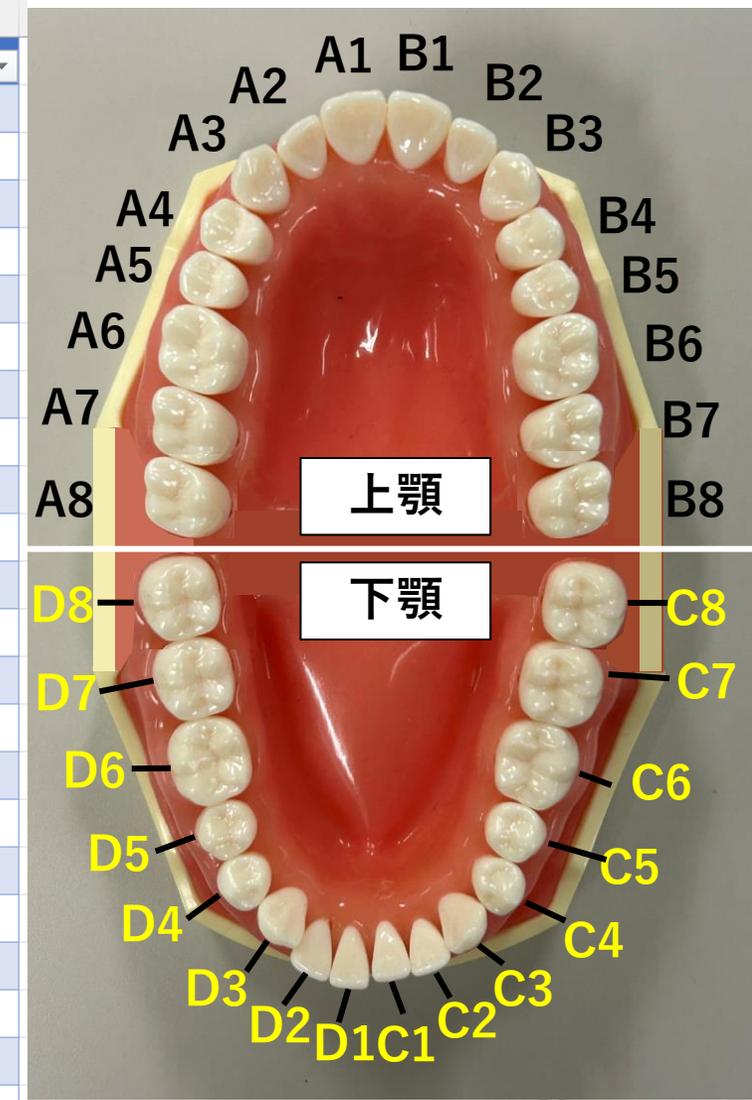
抜歯などによって欠損している歯



架空データ

歯の部位

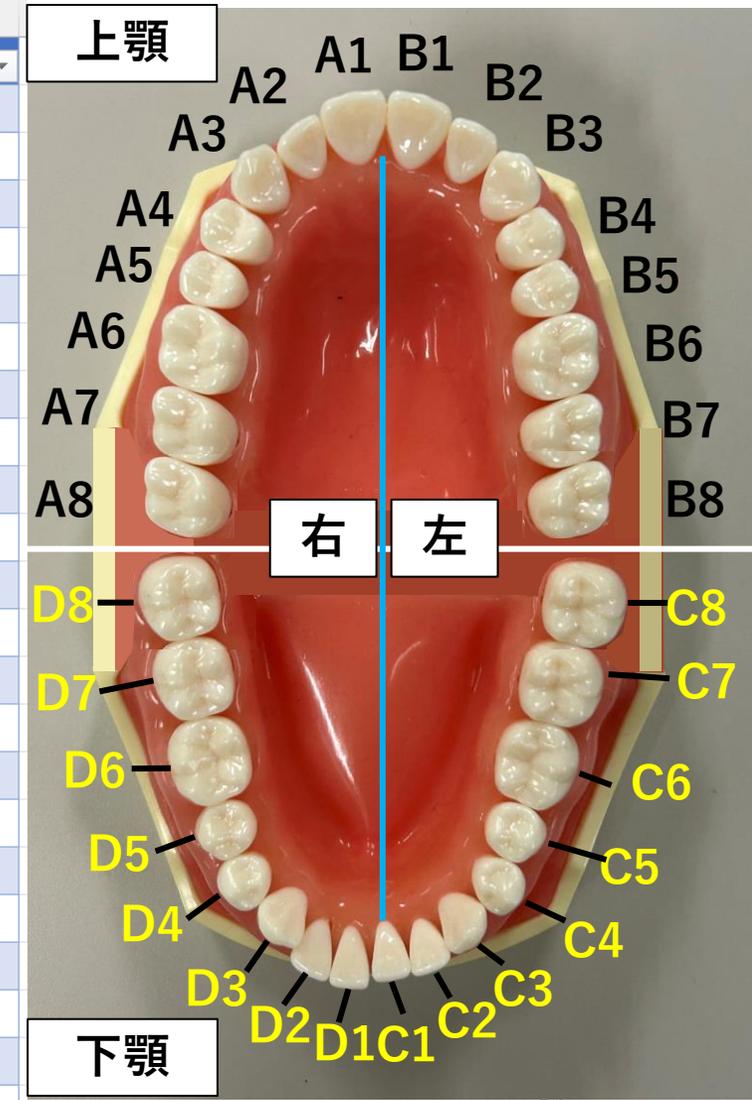
	A	B	C	D	E	F
1	tooth name	tooth record	gender	anonymized id	birthmonth	yearmonth
2	A1	残存歯	男	pt_1	Jan-43	2020/10
3	A2	残存歯	男	pt_1	Jan-43	2020/10
4	A3	残存歯	男	pt_1	Jan-43	2020/10
5	A4	残存歯	男	pt_1	Jan-43	2020/10
6	A5	残存歯	男	pt_1	Jan-43	2020/10
7	A6	残存歯	男	pt_1	Jan-43	2020/10
8	A7	残存歯	男	pt_1	Jan-43	2020/10
9	A8	残存歯	男	pt_1	Jan-43	2020/10
10	B1	残存歯	男	pt_1	Jan-43	2020/10
11	B2	喪失歯	男	pt_1	Jan-43	2020/10
12	B3	残存歯	男	pt_1	Jan-43	2020/10
13	B4	残存歯	男	pt_1	Jan-43	2020/10
14	B5	残存歯	男	pt_1	Jan-43	2020/10
15	B6	残存歯	男	pt_1	Jan-43	2020/10
16	B7	残存歯	男	pt_1	Jan-43	2020/10
17	B8	残存歯	男	pt_1	Jan-43	2020/10
18	C1	残存歯	男	pt_1	Jan-43	2020/10
19	C2	残存歯	男	pt_1	Jan-43	2020/10
20	C3	残存歯	男	pt_1	Jan-43	2020/10
21	C4	喪失歯	男	pt_1	Jan-43	2020/10
22	C5	喪失歯	男	pt_1	Jan-43	2020/10
23	C6	残存歯	男	pt_1	Jan-43	2020/10



架空データ

歯の部位

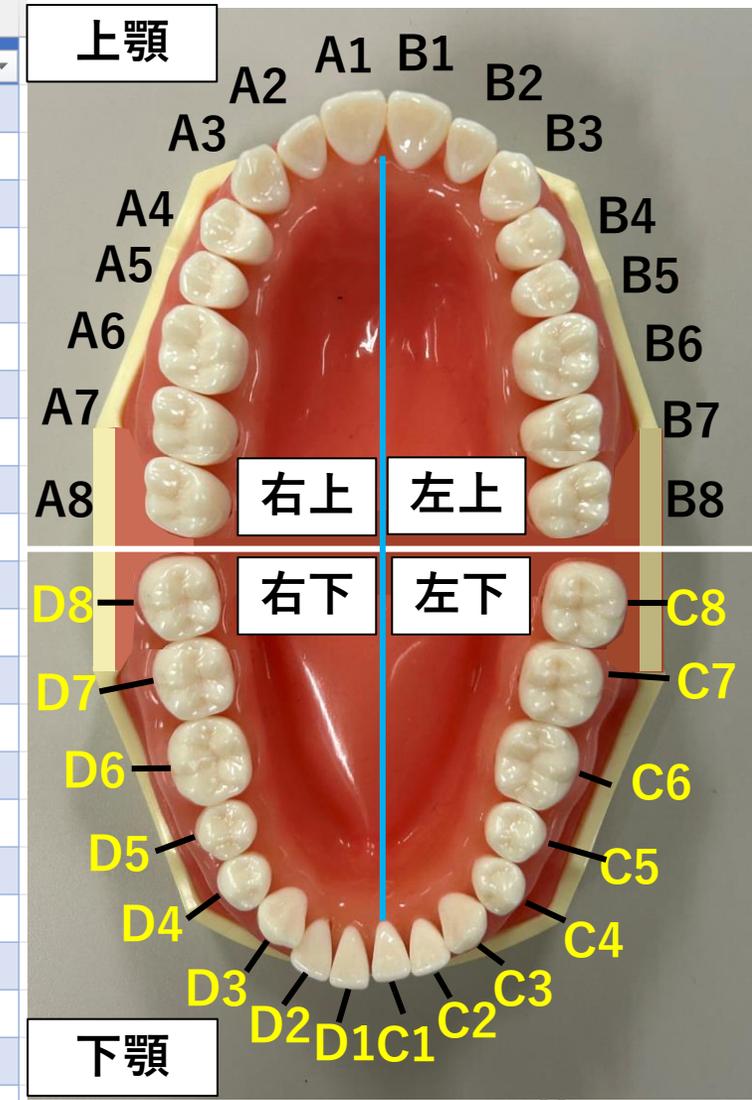
	A	B	C	D	E	F
1	tooth name	tooth record	gender	anonymized id	birthmonth	yearmonth
2	A1	残存歯	男	pt_1	Jan-43	2020/10
3	A2	残存歯	男	pt_1	Jan-43	2020/10
4	A3	残存歯	男	pt_1	Jan-43	2020/10
5	A4	残存歯	男	pt_1	Jan-43	2020/10
6	A5	残存歯	男	pt_1	Jan-43	2020/10
7	A6	残存歯	男	pt_1	Jan-43	2020/10
8	A7	残存歯	男	pt_1	Jan-43	2020/10
9	A8	残存歯	男	pt_1	Jan-43	2020/10
10	B1	残存歯	男	pt_1	Jan-43	2020/10
11	B2	喪失歯	男	pt_1	Jan-43	2020/10
12	B3	残存歯	男	pt_1	Jan-43	2020/10
13	B4	残存歯	男	pt_1	Jan-43	2020/10
14	B5	残存歯	男	pt_1	Jan-43	2020/10
15	B6	残存歯	男	pt_1	Jan-43	2020/10
16	B7	残存歯	男	pt_1	Jan-43	2020/10
17	B8	残存歯	男	pt_1	Jan-43	2020/10
18	C1	残存歯	男	pt_1	Jan-43	2020/10
19	C2	残存歯	男	pt_1	Jan-43	2020/10
20	C3	残存歯	男	pt_1	Jan-43	2020/10
21	C4	喪失歯	男	pt_1	Jan-43	2020/10
22	C5	喪失歯	男	pt_1	Jan-43	2020/10
23	C6	残存歯	男	pt_1	Jan-43	2020/10



架空データ

歯の部位

	A	B	C	D	E	F
1	tooth name	tooth record	gender	anonymized id	birthmonth	yearmonth
2	A1	残存歯	男	pt_1	Jan-43	2020/10
3	A2	残存歯	男	pt_1	Jan-43	2020/10
4	A3	残存歯	男	pt_1	Jan-43	2020/10
5	A4	残存歯	男	pt_1	Jan-43	2020/10
6	A5	残存歯	男	pt_1	Jan-43	2020/10
7	A6	残存歯	男	pt_1	Jan-43	2020/10
8	A7	残存歯	男	pt_1	Jan-43	2020/10
9	A8	残存歯	男	pt_1	Jan-43	2020/10
10	B1	残存歯	男	pt_1	Jan-43	2020/10
11	B2	喪失歯	男	pt_1	Jan-43	2020/10
12	B3	残存歯	男	pt_1	Jan-43	2020/10
13	B4	残存歯	男	pt_1	Jan-43	2020/10
14	B5	残存歯	男	pt_1	Jan-43	2020/10
15	B6	残存歯	男	pt_1	Jan-43	2020/10
16	B7	残存歯	男	pt_1	Jan-43	2020/10
17	B8	残存歯	男	pt_1	Jan-43	2020/10
18	C1	残存歯	男	pt_1	Jan-43	2020/10
19	C2	残存歯	男	pt_1	Jan-43	2020/10
20	C3	残存歯	男	pt_1	Jan-43	2020/10
21	C4	喪失歯	男	pt_1	Jan-43	2020/10
22	C5	喪失歯	男	pt_1	Jan-43	2020/10
23	C6	残存歯	男	pt_1	Jan-43	2020/10



架空データ

歯の部位

歯の状態

B6

喪失歯

B7

残存歯



架空データ

C2	残存歯	女	pt_1657	Dec-82	2022/12
C3	喪失歯	女	pt_1657	Dec-82	2022/12
C4	残存歯	女	pt_1657	Dec-82	2022/12
C5	喪失歯	女	pt_1657	Dec-82	2022/12
C6	残存歯	女	pt_1657	Dec-82	2022/12
C7	喪失歯	女	pt_1657	Dec-82	2022/12
C8	残存歯	女	pt_1657	Dec-82	2022/12
D1	残存歯	女	pt_1657	Dec-82	2022/12
D2	残存歯	女	pt_1657	Dec-82	2022/12
D3	残存歯	女	pt_1657	Dec-82	2022/12
D4	残存歯	女	pt_1657	Dec-82	2022/12
D5	残存歯	女	pt_1657	Dec-82	2022/12
D6	喪失歯	女	pt_1657	Dec-82	2022/12
D7	残存歯	女	pt_1657	Dec-82	2022/12
D8	残存歯	女	pt_1657	Dec-82	2022/12

多数の患者情報が入っています。

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。

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。

Pythonを使用してデータクレンジングを行う必要性

7-10行

A6	残存歯	男	pt_1	Jan-43	20-Oct
A7	残存歯	男	pt_1	Jan-43	20-Oct
A8	残存歯	男	pt_1	Jan-43	20-Oct
B1	残存歯	男	pt_1	Jan-43	20-Oct

患者1

同じ患者が歯毎に複数行に記録されています。
何人いますか？
→人数の集計が必要です。

45-48行

B4	残存歯	男	pt_2	Jan-43	22-Dec
B5	残存歯	男	pt_2	Jan-43	22-Dec
B6	喪失歯	男	pt_2	Jan-43	22-Dec
B7	残存歯	男	pt_2	Jan-43	22-Dec

患者2

3987-3990行

C2	残存歯	女	pt_125	Nov-46	18-May
C3	喪失歯	女	pt_125	Nov-46	18-May
C4	残存歯	女	pt_125	Nov-46	18-May
C5	残存歯	女	pt_125	Nov-46	18-May

患者3

45200-45302行

B7	喪失歯	女	pt_1413	Jan-75	21-Dec
B8	残存歯	女	pt_1413	Jan-75	21-Dec
C1	残存歯	女	pt_1413	Jan-75	21-Dec
C2	残存歯	女	pt_1413	Jan-75	21-Dec

患者4

Pythonを使用してデータクレンジングを行う必要性

7-10行

A6	残存歯	男	pt_1	Jan-43	20-Oct
A7	残存歯	男	pt_1	Jan-43	20-Oct
A8	残存歯	男	pt_1	Jan-43	20-Oct
B1	残存歯	男	pt_1	Jan-43	20-Oct

患者1

患者が何人いますか？
→人数の集計が必要です。

45-48行

B4	残存歯	歯あり
B5	残存歯	歯あり
B6	喪失歯	歯なし
B7	残存歯	歯あり

一人の患者さんが複数の歯の情報を持っています。
→歯の数の集計が必要です。

患者2

3987-3990行

C2	残存歯	女	pt_125	Nov-46	18-May
C3	喪失歯	女	pt_125	Nov-46	18-May
C4	残存歯	女	pt_125	Nov-46	18-May
C5	残存歯	女	pt_125	Nov-46	18-May

患者3

45200-45302行

B7	喪失歯	女	pt_1413	Jan-75	21-Dec
B8	残存歯	女	pt_1413	Jan-75	21-Dec
C1	残存歯	女	pt_1413	Jan-75	21-Dec
C2	残存歯	女	pt_1413	Jan-75	21-Dec

患者4

Pythonを使用してデータクレンジングを行う必要性

7-10行

ひとりの患者さんが複数の歯の情報を持っています。
→歯の数を集計しましょう。

	残存歯	男	pt_1	Jan-43
A6	残存歯	男	pt_1	Jan-43
A7	残存歯	男	pt_1	Jan-43
			pt_1	Jan-43
			pt_1	Jan-43

1943年1月
1943年1月
1943年1月
1943年1月

患者1

患者さんは何人いますか？
→人数を数えてみましょう

B4	残存歯	歯あり
B5	残存歯	歯あり
B6	喪失歯	歯なし
B7	残存歯	歯あり

	pt_2	Jan-43
pt_2	Jan-43	

1943年1月
1943年1月
1943年1月
1943年1月

患者2

患者さんは何歳でしょうか？
→年齢を算出してみましょう。

3987-3990行

	残存歯	女	pt_125	Nov-46
C2	残存歯	女	pt_125	Nov-46
C3	喪失歯	女	pt_125	Nov-46
C4	残存歯	女	pt_125	Nov-46
C5	残存歯	女	pt_125	Nov-46

1946年11月
1946年11月
1946年11月
1946年11月

患者3

45200-45302行

B7	喪失歯	女	pt_1413	Jan-75
B8	残存歯	女	pt_1413	Jan-75
C1	残存歯	女	pt_1413	Jan-75
C2	残存歯	女	pt_1413	Jan-75

1975年1月
1975年1月
1975年1月
1975年1月

患者4

演習最終目標：

患者人数

患者年齢

歯の本数

患者01
患者01
患者01
患者01

2018/01
2018/01
2018/01
2018/01

49歳
49歳
49歳
49歳

歯なし
歯なし
歯あり
歯あり

患者01

2018/01/15

49歳

歯なし
歯あり

○本
○本

患者02
患者02
患者02
患者02

2020/09
2020/09
2020/09
2020/09

68歳
68歳
68歳
68歳

歯あり
歯あり
歯なし
歯あり

患者02

2020/09/15

68歳

歯なし
歯あり

○本
○本

患者03
患者03
患者03
患者03

2022/12
2022/12
2022/12
2022/12

79歳
79歳
79歳
79歳

歯なし
歯なし
歯あり
歯あり

患者03

2022/12/22

79歳

歯なし
歯あり

○本
○本

患者04
患者04
患者04
患者04

2021/06
2021/06
2021/06
2021/06

43歳
43歳
43歳
43歳

歯なし
歯なし
歯なし
歯なし

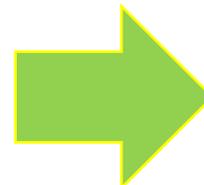
患者04

2021/06/22

43歳

歯なし
歯あり

○本
○本



...

患者n
患者n
患者n
患者n

20○/○
20○/○
20○/○
20○/○

○歳
○歳
○歳
○歳

歯なし
歯あり
歯あり
歯あり

患者n

20○/○/○

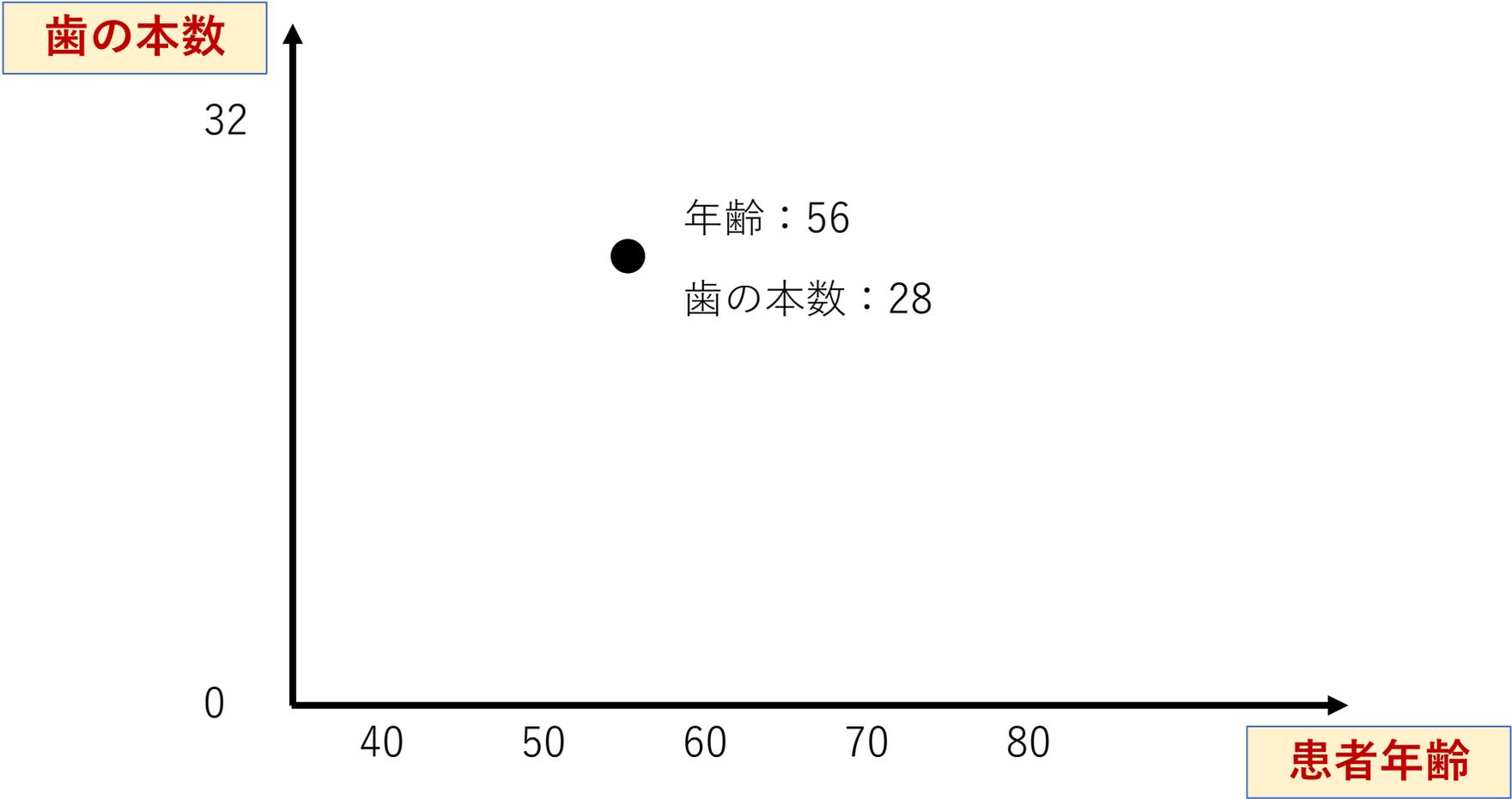
○歳

歯なし
歯あり

○本
○本

演習最終目標：データの可視化

plot散布図を作成します。



演習5～演習7:

・ Pythonを使用してデータクレンジングを行う

■ ライブラリのインポート:

Pandasライブラリをインポートします。

■ データの読み込み:

データは既にCSVの形式で抽出してあります。

■ データフレームの基本的な情報を確認します:

Pandasを使用してデータの形状、カラムなどを把握します。

■ データの集計および処理:

データフレームから歯の本数および患者さんの人数、年齢を集計します。

演習5：データフレーム読み込み

手順：Google Driveにファイル（ `dwhdata.csv` ）をアップロードします

グーグルドライブ

で検索します。

Google

https://drive.google.com › drive › my-drive

Google ドライブ: ログイン

でログインします。

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

🌐

+ 新規

マイドライブ

種類

ユーザー

最終更新

候補リスト

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

演習5：データフレーム読み込み

手順：Google Driveにファイル（ `dwhdata.csv` ）をアップロードします

The screenshot shows the Google Drive web interface. On the left sidebar, the '+ 新規' (New) button is highlighted with a red box and a red arrow. Below it, the 'マイドライブ' (My Drive) section is visible. In the main content area, the 'ドライブ' (Drive) search bar is at the top. Below it, the 'マイドライブ' (My Drive) section is shown. A red arrow points to the 'ファイルのアップロード' (File upload) option in the 'マイドライブ' section. To the right, a file named 'dwhdata.csv' is shown with a red box around it. The file icon is a blue document with a white page symbol.

演習5: データフレーム読み込み

手順: Google Driveをマウントします (ドライブのデータに直接アクセスできるようにする)
Google Colaboratoryでcsvファイルを読み込みます



```
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**

Google にログイン

アカウントの選択

[「Google Drive for desktop」に移動](#)

曹日丹
ridancaoemdv@gmail.com

別のアカウントを使用

続行するにあたり、Google はあなたの名前、メールアドレス、言語設定、プロフィール写真を Google Drive for desktop と共有します。このアプリを使用する前に、Google Drive for desktop の [プライバシー ポリシー](#) と [利用規約](#) をご確認ください。

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** 34

Drive already mounted at /content/drive

演習5：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

```
read_csv() csvファイルを読み込みます
```

```
DataFrame名= pd.read_csv("ファイルパス")
```

コード：

```
Klist = pd.read_csv("/content/drive/My Drive/dwhdata.csv", encoding = "utf-8")
```

```
encoding = "utf-8"
```

ファイル操作において、特定の文字エンコーディング形式を指定します。

```
print(Klist)
```

演習5：データフレーム読み込み

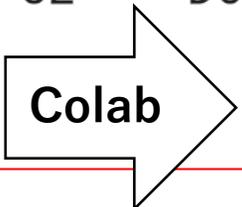
手順：Google Colaboratoryでcsvファイルを読み込みます

```
Klist = pd.read_csv("/content/drive/My Drive/dwhdata.csv", encoding = "utf-8")  
print(Klist)
```



	tooth name	tooth record	gender	anonymized id	birthmonth	yearmonth
0	A1	残存歯	男	pt_1	Jan-43	Oct-20
1	A2	残存歯	男	pt_1	Jan-43	Oct-20
2	A3	残存歯	男	pt_1	Jan-43	Oct-20
3	A4	残存歯	男	pt_1	Jan-43	Oct-20
4	A5	残存歯	男	pt_1	Jan-43	Oct-20
...
53019	D4	残存歯	女	pt_1657	Dec-82	Dec-22
53020	D5	残存歯	女	pt_1657	Dec-82	Dec-22
53021	D6	喪失歯	女	pt_1657	Dec-82	Dec-22
53022	D7	残存歯	女	pt_1657	Dec-82	Dec-22
53023	D8	残存歯	女	pt_1657	Dec-82	Dec-22

[53024 rows x 6 columns]



架空データ

データフレームのshape（行×列）とsize（データ数）を求める

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

コード:

```
print(Klist.shape)
```

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

コード:

```
print(Klist.size)
```

架空データ

データフレームのshape（行×列）とsize（データ数）を求める

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

コード:

```
print(Klist.shape)
```

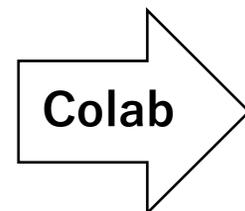
```
(53024, 6)
```

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

コード:

```
print(Klist.size)
```

```
318144
```



架空データ

特定の列を抽出することができます。

DataFrame名 [抽出する列名]

コード： Klistデータフレームから、birthmonthという名前の列を抽出します。

```
print(Klist["birthmonth"])
```

架空データ

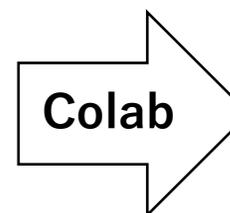
特定の列を抽出することができます。

DataFrame名 [抽出する列名]

コード： `Klist`データフレームから、`birthmonth`という名前の列を抽出します。

```
print(Klist["birthmonth"])
```

```
0      Jan-43
1      Jan-43
2      Jan-43
3      Jan-43
4      Jan-43
...
53019  Dec-82
53020  Dec-82
53021  Dec-82
53022  Dec-82
53023  Dec-82
Name: birthmonth, Length: 53024,
```



架空データ

インデックス番号を指定して、特定の行を抽出できます。

DataFrame名.**iloc**[インデックス番号：インデックス番号]
スライスを使って行を選択できます。

※：**iloc**はinteger locationの略号で、整数によるインデックス番号で位置を指定します

コード：

```
print(Klist.iloc[10000:20000])
```

架空データ

インデックス番号を指定して、特定の行を抽出できます。

DataFrame名.**iloc**[インデックス番号：インデックス番号]

コード：

スライスを使って行を選択できます。

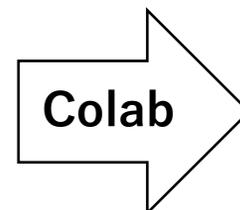
```
print(Klist.iloc[10000:20000])
```

```

↳   tooth name  tooth record  gender  anonymized  id  birthmonth  yearmonth
10000  C1        残存歯      男      pt_313      Sep-50    May-21
10001  C2        残存歯      男      pt_313      Sep-50    May-21
10002  C3        残存歯      男      pt_313      Sep-50    May-21
10003  C4        残存歯      男      pt_313      Sep-50    May-21
10004  C5        残存歯      男      pt_313      Sep-50    May-21
...
19995  D4        残存歯      男      pt_625      Jul-58    Jan-20
19996  D5        残存歯      男      pt_625      Jul-58    Jan-20
19997  D6        残存歯      男      pt_625      Jul-58    Jan-20
19998  D7        残存歯      男      pt_625      Jul-58    Jan-20
19999  D8        残存歯      男      pt_625      Jul-58    Jan-20

```

[10000 rows x 6 columns]



架空データ

特定の条件に基づいて行を選択、抽出することができます。

```
DataFrame名 [条件]
```

```
print(DataFrame名 [条件])
```

⇒ 条件に当てはまるデータフレームの一部を返します。

架空データ

特定の条件に基づいて行を選択、抽出することができます。

例：IDがpt_1の患者データを抽出したい。

DataFrame名 [条件]

患者ID列： `Klist["anonymized id"]`

IDがpt_1の患者という条件： `Klist["anonymized id"] == "pt_1"`

IDがpt_1の患者データを抽出します： `Klist[Klist["anonymized id"] == "pt_1"]`

架空データ

特定の条件に基づいて行を選択、抽出することができます。

例：IDがpt_1の患者データを抽出したい。

コード：

```
print(Klist[Klist["anonymized id"] == "pt_1"])
```

Klistデータフレーム

anonymized id列

The diagram shows a large blue rectangle representing a data frame. A vertical yellow column is highlighted in the center, representing the 'anonymized id' column. Red arrows point from the labels 'Klistデータフレーム' and 'anonymized id列' to the blue area and the yellow column respectively. The yellow column contains the following text from top to bottom: pt_555, pt_1, pt_1, pt_1, pt_1000, and pt_1002.

pt_555
pt_1
pt_1
pt_1
pt_1000
pt_1002

架空データ

特定の条件に基づいて行を選択、抽出することができます。

例：IDがpt_1の患者データを抽出したい。

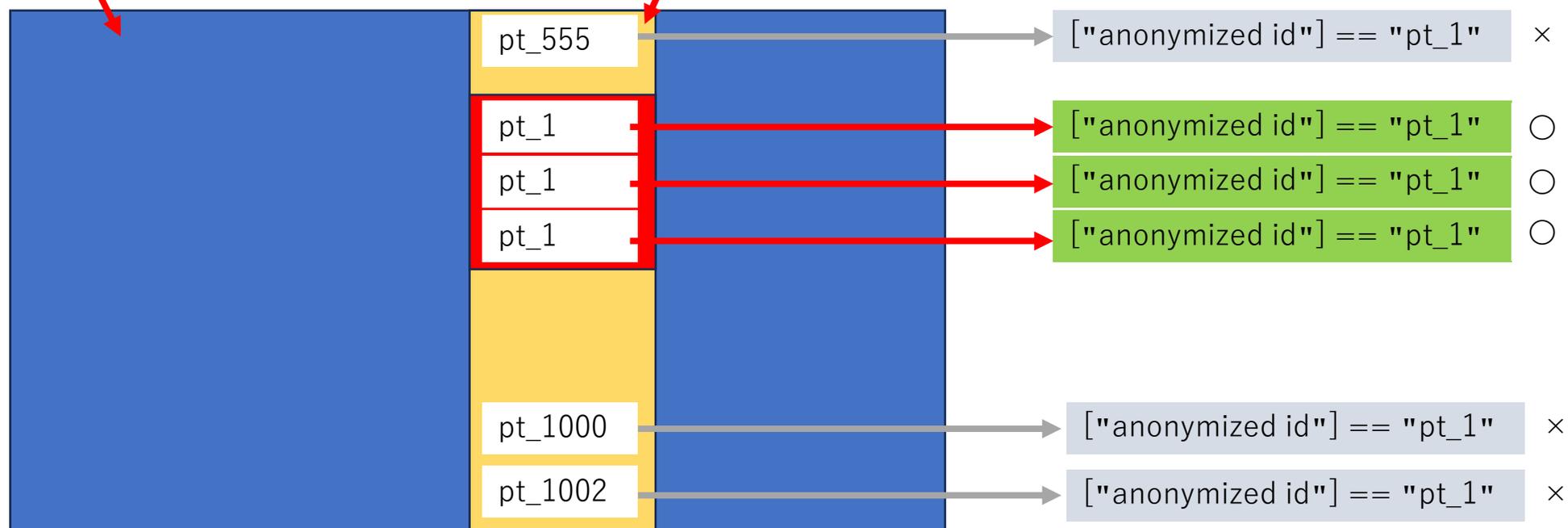
コード：

```
print(Klist[Klist["anonymized id"] == "pt_1"])
```

Klistデータフレーム

anonymized id列

条件



架空データ

特定の条件に基づいて行を選択、抽出することができます。

例：IDがpt_1の患者データを抽出したい。

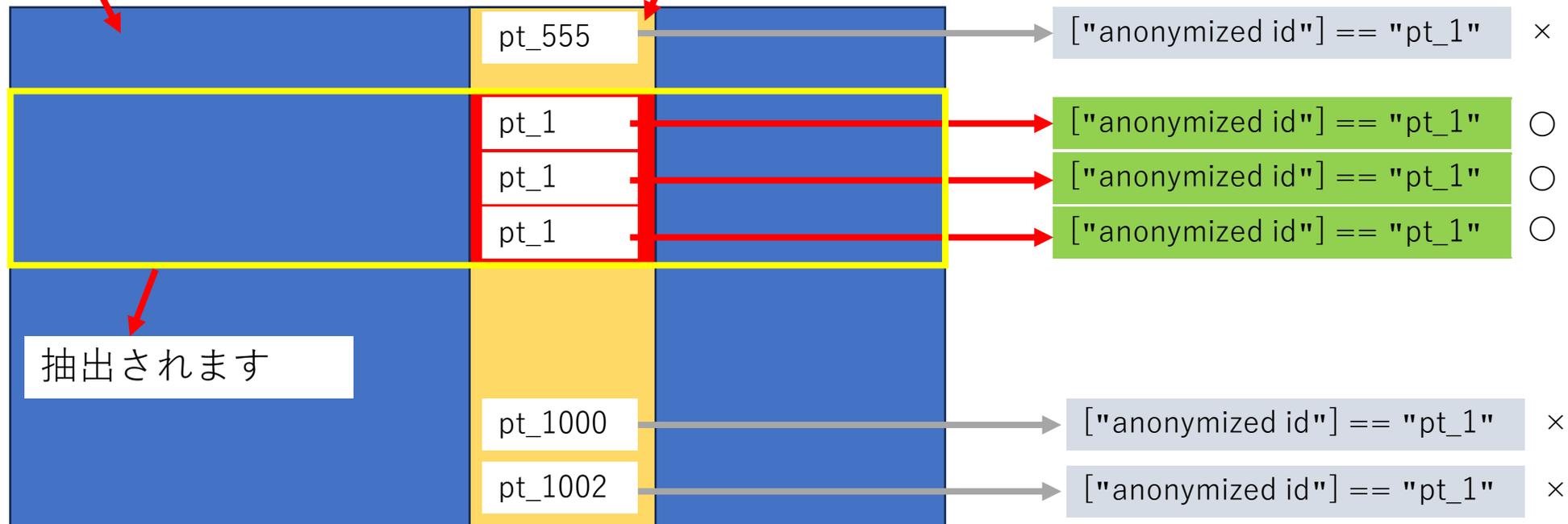
コード：

```
print(Klist[Klist["anonymized id"] == "pt_1"])
```

Klistデータフレーム

anonymized id列

条件



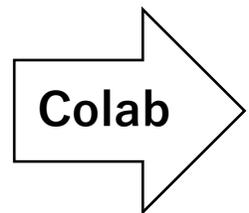
架空データ

特定の条件に基づいて行を選択、抽出することができます。

例：IDがpt_1の患者データを抽出したい。

```
print(Klist[Klist["anonymized id"] == "pt_1"])
```

	tooth name	tooth record	gender	anonymized id	birthmonth	yearmonth
0	A1	残存歯	男	pt_1	Jan-43	Oct-20
1	A2	残存歯	男	pt_1	Jan-43	Oct-20
2	A3	残存歯	男	pt_1	Jan-43	Oct-20
3	A4	残存歯	男	pt_1	Jan-43	Oct-20
4	A5	残存歯	男	pt_1	Jan-43	Oct-20
5	A6	残存歯	男	pt_1	Jan-43	Oct-20
6	A7	残存歯	男	pt_1	Jan-43	Oct-20
7	A8	残存歯	男	pt_1	Jan-43	Oct-20
8	B1	残存歯	男	pt_1	Jan-43	Oct-20
9	B2	喪失歯	男	pt_1	Jan-43	Oct-20
10	B3	残存歯	男	pt_1	Jan-43	Oct-20
11	B4	残存歯	男	pt_1	Jan-43	Oct-20
12	B5	残存歯	男	pt_1	Jan-43	Oct-20
13	B6	残存歯	男	pt_1	Jan-43	Oct-20
14	B7	残存歯	男	pt_1	Jan-43	Oct-20
15	B8	残存歯	男	pt_1	Jan-43	Oct-20
16	C1	残存歯	男	pt_1	Jan-43	Oct-20
17	C2	残存歯	男	pt_1	Jan-43	Oct-20
18	C3	残存歯	男	pt_1	Jan-43	Oct-20
19	C4	喪失歯	男	pt_1	Jan-43	Oct-20
20	C5	喪失歯	男	pt_1	Jan-43	Oct-20
21	C6	残存歯	男	pt_1	Jan-43	Oct-20
22	C7	残存歯	男	pt_1	Jan-43	Oct-20
23	C8	残存歯	男	pt_1	Jan-43	Oct-20
24	D1	残存歯	男	pt_1	Jan-43	Oct-20
25	D2	喪失歯	男	pt_1	Jan-43	Oct-20
26	D3	残存歯	男	pt_1	Jan-43	Oct-20
27	D4	残存歯	男	pt_1	Jan-43	Oct-20
28	D5	残存歯	男	pt_1	Jan-43	Oct-20
29	D6	残存歯	男	pt_1	Jan-43	Oct-20
30	D7	喪失歯	男	pt_1	Jan-43	Oct-20
31	D8	残存歯	男	pt_1	Jan-43	Oct-20



架空データ

WebClassで課題を提出してください、締め切りは12月21日23:59までです。

課題1：

Pandasライブラリをインポートするコードを書いてください。

Pandasの略称は「pd」とします。

課題2：

print関数を使って、データフレームklistの歯の状態という列を抽出するコードを書いてください。

歯の状態の列名はtooth recordです。

演習授業中の質問対応について

The image shows a Zoom meeting interface. The main window displays a presentation slide with the following text:

演習授業中の質問をチューターの先生方が対応させていただきます。

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。

The interface includes a Zoom toolbar at the bottom with icons for Mute, Video, Security, Participants, Screen Share, Reactions, Apps, Whiteboard, Notes, and Details. The Reactions icon is highlighted with a red box. The chat window on the right shows a dropdown menu for recipients, with "全員" (Everyone) selected and highlighted with a red box. A red arrow points from the "挙手" button in the toolbar to the "全員" selection in the chat window.

演習授業中の質問をチューターの先生方が対応させていただきます。

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。

挙手

宛先: 全員

演習6

データクレンジングに必要なライブラリ (Pandas) の応用

統合教育機構 曹 日丹

医療とAI・ビッグデータ入門

演習2-7の構成

Python基礎を学びましょう

演習2

Pythonの変数とデータの型

演習3

プログラミング基礎

演習4

モジュール、パッケージ、ライブラリ

Pythonを使ってみましょう

演習5

患者の歯に関する~~病院のリアルデータ~~の説明

架空データ

演習6 12/21 10:40-11:35

データクレンジングに必要なライブラリ（Pandas）の応用

演習7 12/21 11:35-12:20

データクレンジングとデータの可視化

Pythonは豊富なライブラリを持っています。
いくつかの主要なPythonライブラリとその概要を示しています。

NumPy:

数値計算ライブラリ。

Pandas:

データ分析ライブラリ。
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。
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。

Matplotlib:

データの可視化ライブラリ。

Scikit-learn:

機械学習ライブラリ。

手順1: Google Driveにファイル ([dwhdata1221.csv](#)) をアップロードします

dwhdata1221.csv

前回演習に使ってcsvファイルと同じ架空データですが、
日付の表記を使いやすい形式に変換したものです。

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

手順1: Google Driveにファイル ([dwhdata1221.csv](#)) をアップロードします

➡ **グーグルドライブ** で検索します。



➡ **Google ドライブ: ログイン** でログインします。

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。

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

手順1: Google Driveにファイル ([dwhdata1221.csv](#)) をアップロードします



The screenshot shows the Google Drive web interface. On the left sidebar, the '+ 新規' (New) button is highlighted with a red box and a red arrow pointing to it. Below it are navigation options like 'マイドライブ' (My Drive), 'パソコン' (Computer), and '共有アイテム' (Shared items). The main area shows 'マイドライブ' (My Drive) with filters for '種類' (Type), 'ユーザー' (User), and '最終更新' (Last updated). Below the filters is a '候補リスト' (Candidate list) showing three recent files: '演習5コード.ipynb', '演習5dataframe.ipynb', and '演習4コード.ipynb'. At the bottom, there is a table listing folders: '.ipynb_checkpoints', 'Colab Notebooks', and 'images', all owned by '自分' (Self).

ドライブ

ドライブで検索

+ 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

15 GB 中 7.91 GB を使用

保存容量を増やす

マイドライブ

種類 ユーザー 最終更新

候補リスト

演習5コード.ipynb

```
import pandas as pd

person = {"名前":["太郎"],"年齢":[30],"住所":["文京区"]}

print(person)

#辞書からデータフレームを作成します。
kdata = {"ID":[10001, 10002, 10003, 10004], "名前":["花
```

数秒前に編集したファイル

演習5dataframe.ipynb

```
import pandas as pd

person = {"name":"John","age":30, "city":"New York"}

data1 = {"ID":[10001, "名前":["花子"], "年齢":[58]}
df1 = pd.DataFrame(data1)
print(df1)

#辞書からデータフレームを作成します。
data = {"ID":[10001, 10002, 10003, 10004], "名前":["花子",
df = pd.DataFrame(data)
```

今日開いたファイル

演習4コード.ipynb

今日変更を加えたファイル

名前	↑	オーナー
.ipynb_checkpoints		自分
Colab Notebooks		自分
images		自分

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

手順1: Google Driveにファイル (`dwldata1221.csv`) をアップロードします

The screenshot shows the Google Drive web interface. On the left sidebar, the '+ 新規' (New) button is highlighted with a red box and a red arrow pointing to it. Below it, the 'マイドライブ' (My Drive) option is selected. The main content area shows the 'マイドライブ' view with a search bar and a list of options. The 'ファイルのアップロード' (Upload files) option is highlighted with a red box and a red arrow. To the right of this option, the filename 'dwldata1221.csv' is displayed in a yellow box. The interface also shows other options like '新しいフォルダ' (New folder), 'フォルダのアップロード' (Upload folders), and various Google Workspace applications like Google Docs, Sheets, Slides, and Forms.

演習6 データクレンジングに必要なライブラリ (Pandas) の応用



ドライブ

ドライブで検索



+ 新規

ホーム

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

15 GB 中 8 GB を使用

保存容量を増やす

検索結果

種類

ユーザー

最終更新

場所

タイトルのみ

ToDo

名前	オーナー	最終更新
演習6コード.ipynb	自分	2023/12/19
演習7コード	自分	2023/12/19
dataframe準備資料.ipynb	自分	2023/12/13
演習6コード準備内容.ipynb	自分	2023/12/15
dwldata1221.csv	自分	2023/12/15
乱数作業記録202311301646.ipynb	自分	2023/12/15
データの保存.ipynb	自分	2023/12/11
演習3コード.ipynb	自分	2023/12/10
演習2コード.ipynb	自分	2023/11/15
演習5コード.ipynb ★	自分	2023/12/07



[タイトルなし]

検索google colab

Colaboratory へようこそ - Colaboratory - Google

The screenshot shows the Google Colaboratory interface. On the left is a sidebar with a table of contents. The main area displays a search for notebooks. A table lists the search results, with the first row '演習6コード' highlighted. At the bottom, there is a button to create a new notebook and a 'キャンセル' (Cancel) button.

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 >

検索: ノートブックを検索

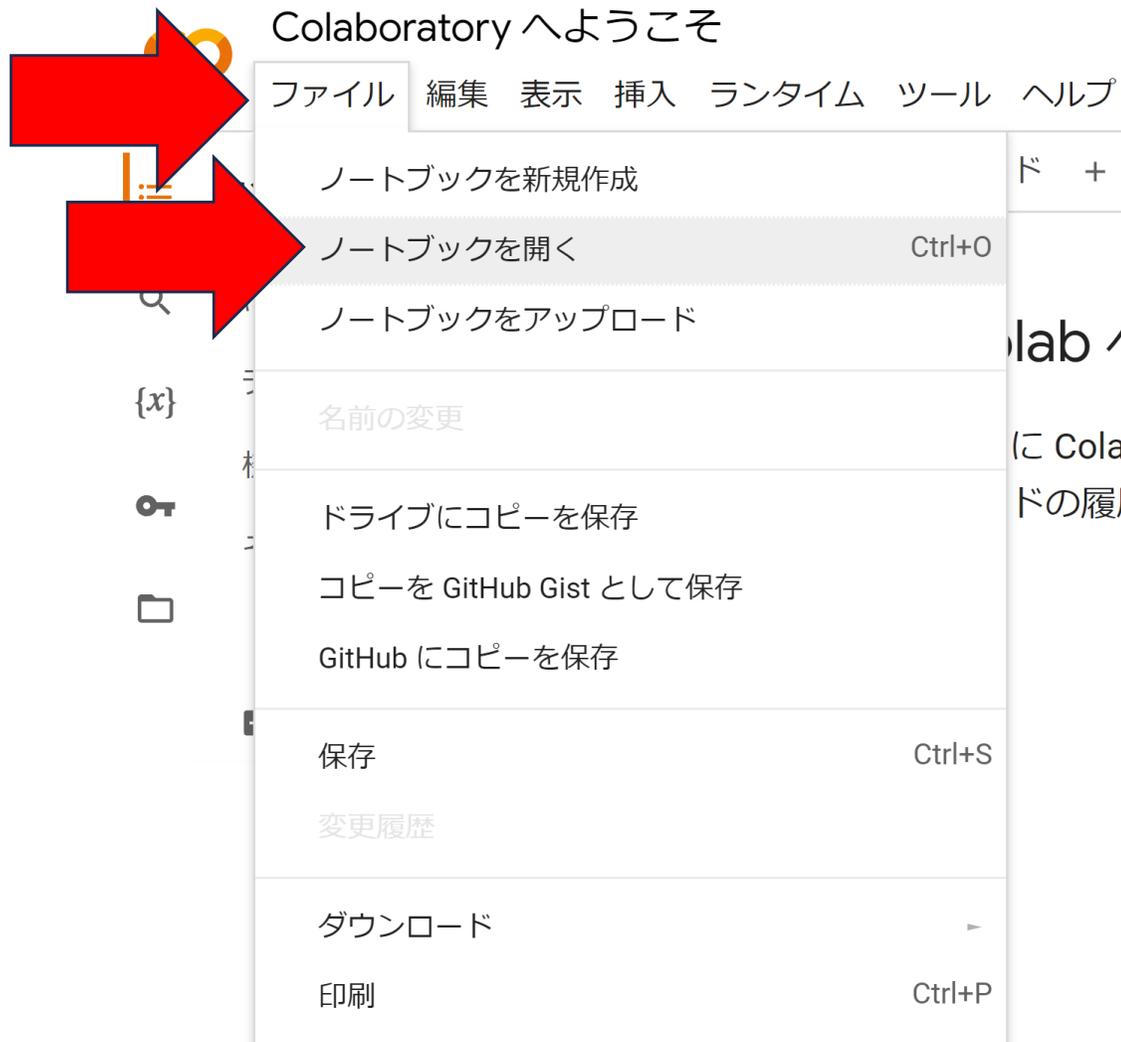
タイトル	最終閲覧	最初に開いた日時	
演習6コード	9:43	10月25日	
Colaboratory へようこそ	9:21	2022年12月23日	
演習7コード	9:21	9:21	
演習4コード	11月2日	11月1日	
演習5コード	11月1日	10月13日	

+ ノートブックを新規作成

キャンセル

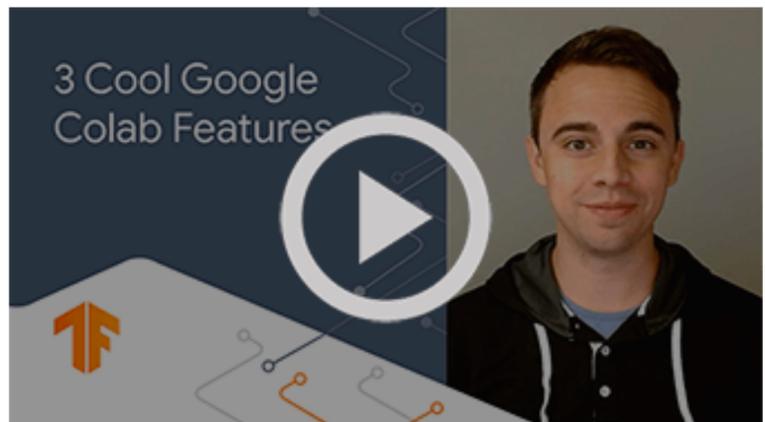
検索google colab

Colaboratory へようこそ - Colaboratory - Google



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

検索google colab

Colaboratory へようこそ - Colaboratory - Google

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 ▼		
演習6コード.ipynb					
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

検索google colab Colaboratory へようこそ - Colaboratory - Google

ノートブックを開く

例 >

最近 >

Google ドラ
イブ >

GitHub >

アップロード >



参照

または、ここにファイルをドラッグしてください

演習6コード.ipynb

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

手順: Google Driveをマウントします (ドライブのデータに直接アクセスできるようにします)
Google Colaboratoryでcsvファイルを読み込みます

▶ `import pandas as pd`

▶ `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 にログイン



アカウントの選択

「Google Drive for desktop」に移動

 曹日丹
ridancaoemdv@gmail.com

 別のアカウントを使用

続行するにあたり、Google はあなたの名前、メールアドレス、言語設定、プロフィール写真を Google Drive for desktop と共有します。このアプリを使用する前に、Google Drive for desktop の [プライバシー ポリシー](#) と [利用規約](#) をご確認ください。

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 14

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

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

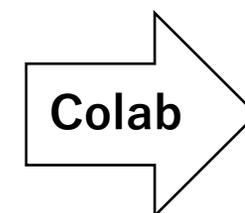


```
Klist = pd.read_csv("/content/drive/My Drive/dwhdata1221.csv", encoding = "utf-8")  
print(Klist)
```



```
tooth name  tooth record  gender      id  birthmonth  yearmonth  
0          A1      残存歯      男      pt_1      1943-01      2020-10  
1          A2      喪失歯      男      pt_1      1943-01      2020-10  
2          A3      残存歯      男      pt_1      1943-01      2020-10  
3          A4      残存歯      男      pt_1      1943-01      2020-10  
4          A5      残存歯      男      pt_1      1943-01      2020-10  
...      ...      ...      ...      ...      ...      ...  
53019     D4      喪失歯      女      pt_1657     1982-12     2022-12  
53020     D5      残存歯      女      pt_1657     1982-12     2022-12  
53021     D6      残存歯      女      pt_1657     1982-12     2022-12  
53022     D7      喪失歯      女      pt_1657     1982-12     2022-12  
53023     D8      残存歯      女      pt_1657     1982-12     2022-12
```

[53024 rows x 6 columns]



演習授業中の質問対応について

Zoom ミーティング

表示

ミーティング チャット

演習授業中の質問をチューターの先生方が対応させていただきます。

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

ここにメッセージは誰に表示されますか？

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

👍 🤔 🙄 ❤️ 🎉 ...

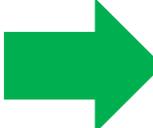
👍 ❌ ⏪ ⏩ ☕

👋 挙手

🗨️ 📄 🔄 ... 16

Pythonを使用してデータクレンジングを行う

- データの読み込み:
CSVファイルをGoogleドライブにアップロード、GoogleColabで読み込みました。
- ライブラリのインポート:
Pandasライブラリをインポートしました。



■データの集計および処理:
データフレームから患者さんの人数および歯の本数、年齢を集計します。

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

患者人数を集計しましょう。

Klistデータフレーム

同じ患者が歯毎に複数行に記録されています。

			ID		
A1	残存歯	男	pt_1	1943-01	2020-10
A2	喪失歯	男	pt_1	1943-01	2020-10
A3	残存歯	男	pt_1	1943-01	2020-10
A4	残存歯	男	pt_1	1943-01	2020-10
A5	残存歯	男	pt_1	1943-01	2020-10
...					
D6	喪失歯	女	pt_1423	1975-05	2018-05
D7	残存歯	女	pt_1423	1975-05	2018-05
D8	喪失歯	女	pt_1423	1975-05	2018-05
A1	残存歯	男	pt_1424	1975-05	2021-03
...					
D5	残存歯	男	pt_576	1957-04	2020-11
D6	残存歯	男	pt_576	1957-04	2020-11
D7	残存歯	男	pt_576	1957-04	2020-11
D8	残存歯	男	pt_576	1957-04	2020-11
A1	残存歯	男	pt_577	1957-04	2022-10
...					

患者を区別する方法

患者ID：病院や診療所で患者に割り当てられるユニークな番号。

同じIDが複数回重複しているため、データセット内の**ユニークな値**を特定し、それらを数える必要があります。

ユニークとは、重複のない、唯一の、一意の意味です。

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

患者人数を集計しましょう。

Klistデータフレーム

同じ患者が歯毎に複数行に記録されています。

			ID	生年月	初診月
A1	残存歯	男	pt_1	1943-01	2020-10
A2	喪失歯	男	pt_1	1943-01	2020-10
A3	残存歯	男	pt_1	1943-01	2020-10
A4	残存歯	男	pt_1	1943-01	2020-10
A5	残存歯	男	pt_1	1943-01	2020-10
...					
D6	喪失歯	女	pt_1423	1975-05	2018-05
D7	残存歯	女	pt_1423	1975-05	2018-05
D8	喪失歯	女	pt_1423	1975-05	2018-05
A1	残存歯	男	pt_1424	1975-05	2021-03
...					
D5	残存歯	男	pt_576	1957-04	2020-11
D6	残存歯	男	pt_576	1957-04	2020-11
D7	残存歯	男	pt_576	1957-04	2020-11
D8	残存歯	男	pt_576	1957-04	2020-11
A1	残存歯	男	pt_577	1957-04	2022-10
...					

生年月 他の患者と同じになる可能性があります。

初診月 他の患者と同じになる可能性があります。

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

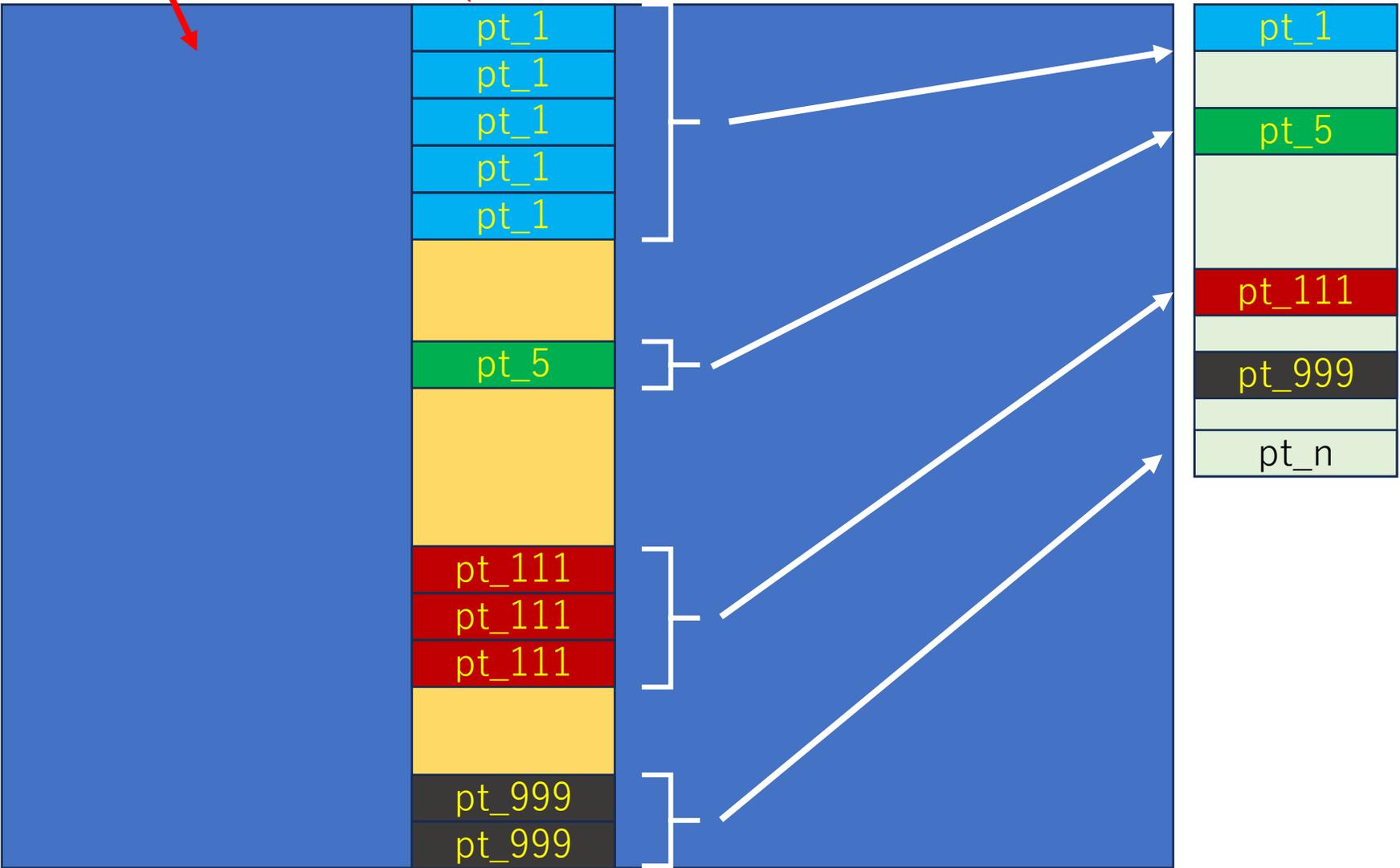
患者人数を集計しましょう。

.unique()

ID列内の**ユニークな値**を集めます。

Klistデータフレーム

id列



ユニークな値を数えます。

.nunique()

n = ●●

患者人数を集計しましょう。

unique()とnunique()メソッド

unique()

データフレーム内の特定の列のユニークな値を返します。

特定の列名.unique()

nunique()

データフレーム内の特定の列のユニークな値の数をカウントします。

特定の列名.nunique()

患者人数を集計しましょう。

unique()とnunique()メソッド

特定の列を抽出します。DataFrame名["列名"]

コード

```
print(Klist["id"])
```

unique()特定の列のユニークな値を返します。

特定の列名.unique()

コード

```
print(Klist["id"].unique())
```

nunique()特定の列のユニークな値の数をカウントします。

特定の列名.nunique()

コード

```
print(Klist["id"].nunique())
```

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

患者人数を集計しましょう。

特定の列を抽出します。 `print(Klist["id"])`

```
0      pt_1
1      pt_1
2      pt_1
3      pt_1
4      pt_1
...
53019  pt_1657
53020  pt_1657
53021  pt_1657
53022  pt_1657
53023  pt_1657
```

複数回重複している同じID

複数回重複している同じID

`unique()` `print(Klist["id"].unique())`

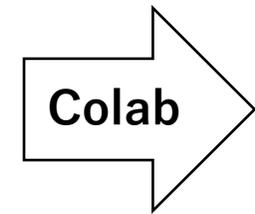
```
['pt_1' 'pt_2' 'pt_3' ... 'pt_1655' 'pt_1656' 'pt_1657']
```

→ numpy配列

ユニックなID

`nunique()` `print(Klist["id"].nunique())`

```
1657
```



Pythonを使用してデータクレンジングを行う

☑ ■データの読み込み:
CSVファイルをGoogleドライブにアップロード、GoogleColabで読み込みました。

☑ ■ライブラリのインポート:
Pandasライブラリをインポートしました。

☑ ■データの集計:
患者の人数を集計します。1657人

➡ 歯の本数を集計します。

年齢を集計します。

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

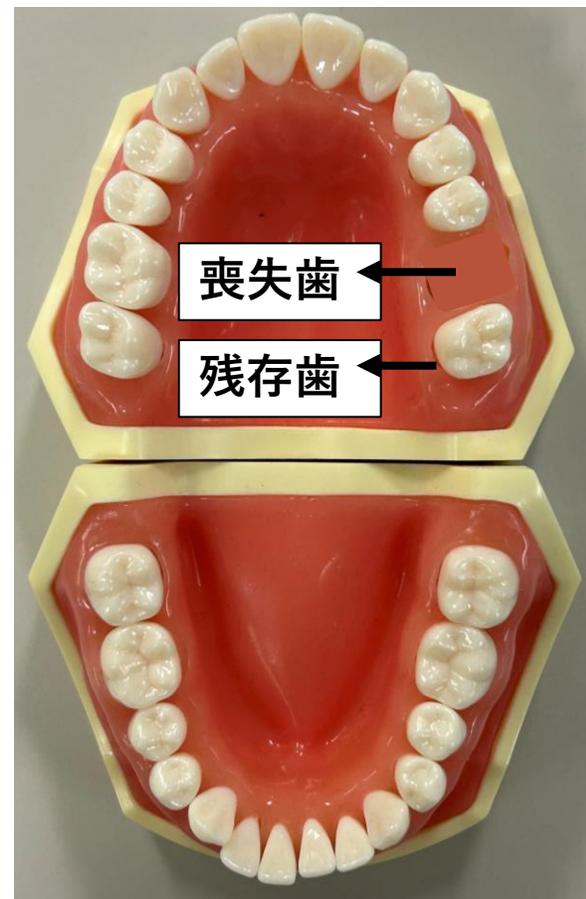
tooth record

Klistデータフレーム

一人の患者さんが複数の歯の情報を持っています：
どうやって歯の数を数えますか？

A1	残存歯	男	pt_1	1943-01	2020-10
A2	喪失歯	男	pt_1	1943-01	2020-10
A3	残存歯	男	pt_1	1943-01	2020-10
A4	残存歯	男	pt_1	1943-01	2020-10
A5	残存歯	男	pt_1	1943-01	2020-10
...					
D6	喪失歯	女	pt_1423	1975-05	2018-05
D7	残存歯	女	pt_1423	1975-05	2018-05
D8	喪失歯	女	pt_1423	1975-05	2018-05
A1	残存歯	男	pt_1424	1975-05	2021-03
...					
D5	残存歯	男	pt_576	1957-04	2020-11
D6	残存歯	男	pt_576	1957-04	2020-11
D7	残存歯	男	pt_576	1957-04	2020-11
D8	残存歯	男	pt_576	1957-04	2020-11
A1	残存歯	男	pt_577	1957-04	2022-10
...					

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



演習6 データクレンジングに必要なライブラリ (Pandas) の応用

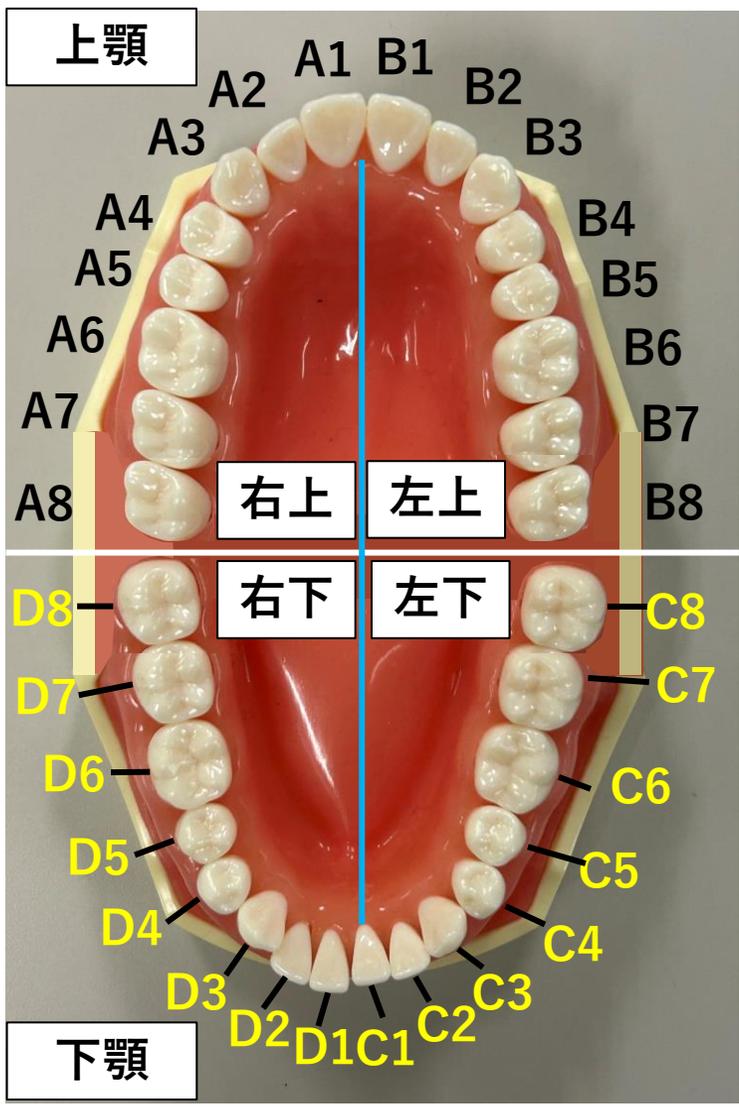
歯の数を集計しましょう。

tooth name

Klistデータフレーム

A1	残存歯	男	pt_1	1943-01	2020-10
A2	喪失歯	男	pt_1	1943-01	2020-10
A3	残存歯	男	pt_1	1943-01	2020-10
A4	残存歯	男	pt_1	1943-01	2020-10
A5	残存歯	男	pt_1	1943-01	2020-10
. . .					
D6	喪失歯	女	pt_1423	1975-05	2018-05
D7	残存歯	女	pt_1423	1975-05	2018-05
D8	喪失歯	女	pt_1423	1975-05	2018-05
A1	残存歯	男	pt_1424	1975-05	2021-03
. . .					
D5	残存歯	男	pt_576	1957-04	2020-11
D6	残存歯	男	pt_576	1957-04	2020-11
D7	残存歯	男	pt_576	1957-04	2020-11
D8	残存歯	男	pt_576	1957-04	2020-11
A1	残存歯	男	pt_577	1957-04	2022-10
. . .					

一人の患者さんが複数の歯の情報を持っています: どうやって歯の数を数えますか?

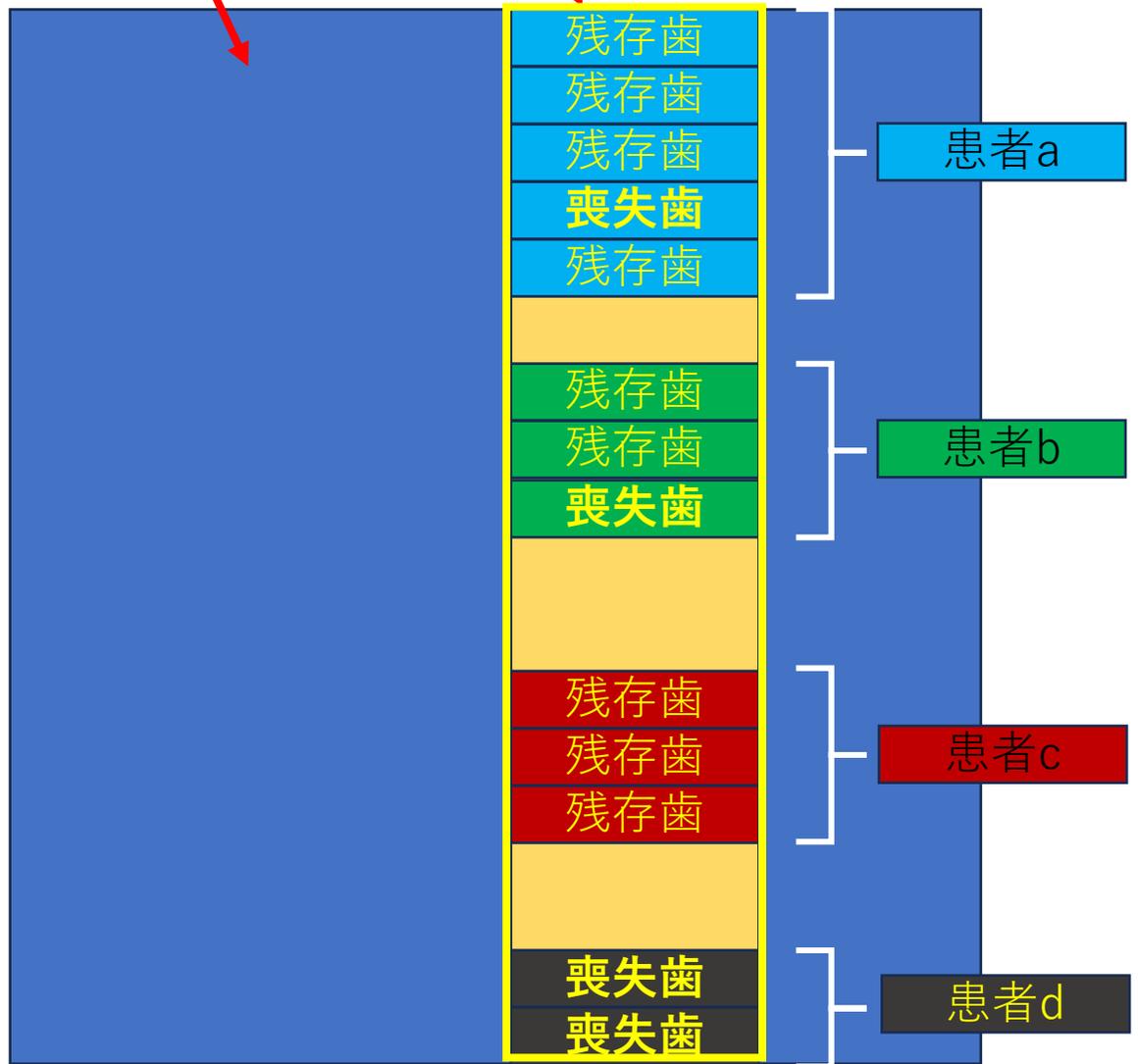


演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

Klistデータフレーム

tooth record列



演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

方法

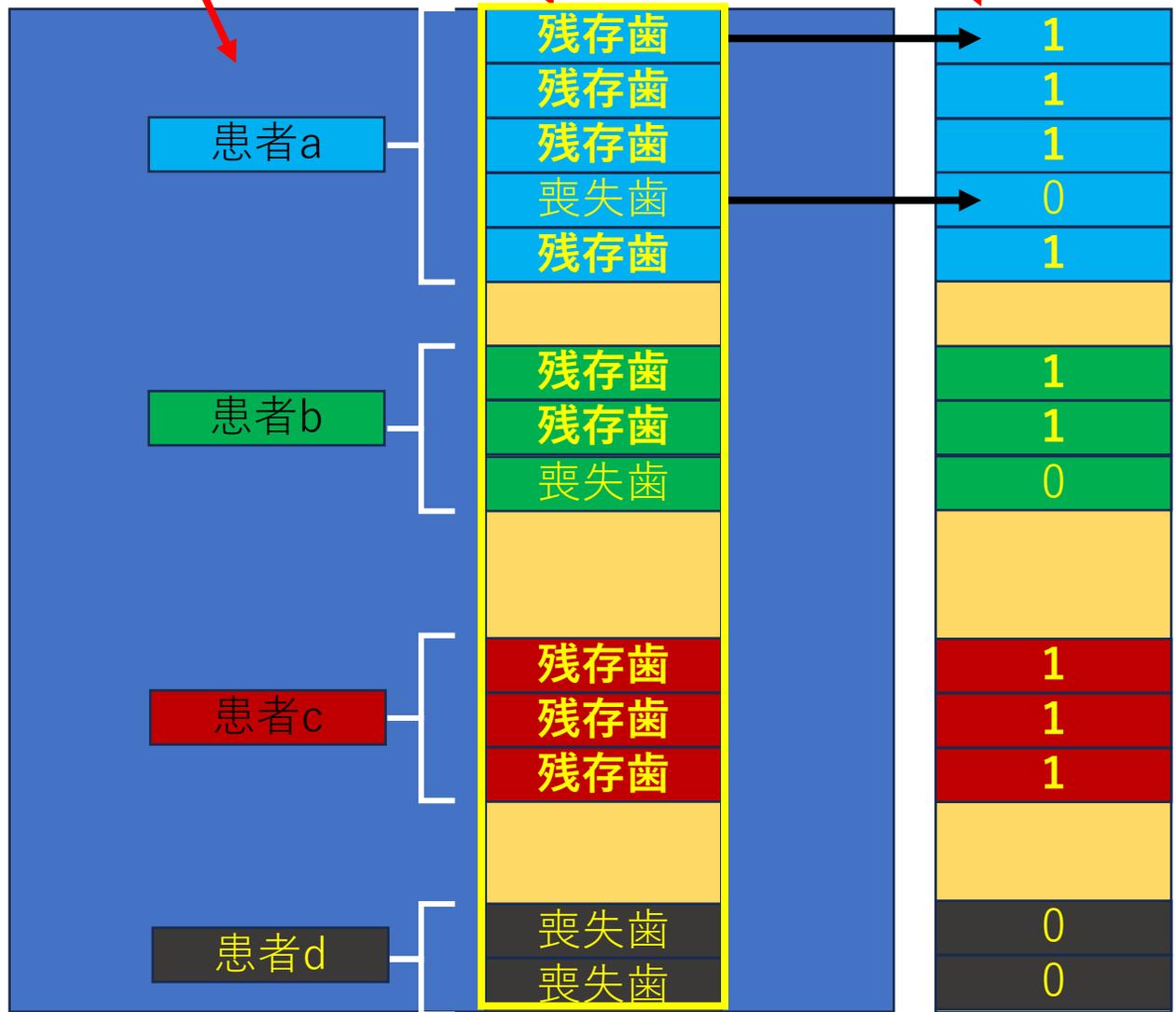
列を追加して歯の数を集計します。

Klistデータフレーム

tooth record列

tooth exist列

足し算で集計します



歯本数a= 1 + 1 + 1 + 0 + 1 ...

歯本数b= 1 + 1 + 0 ...

歯本数c= 1 + 1 + 1 ...

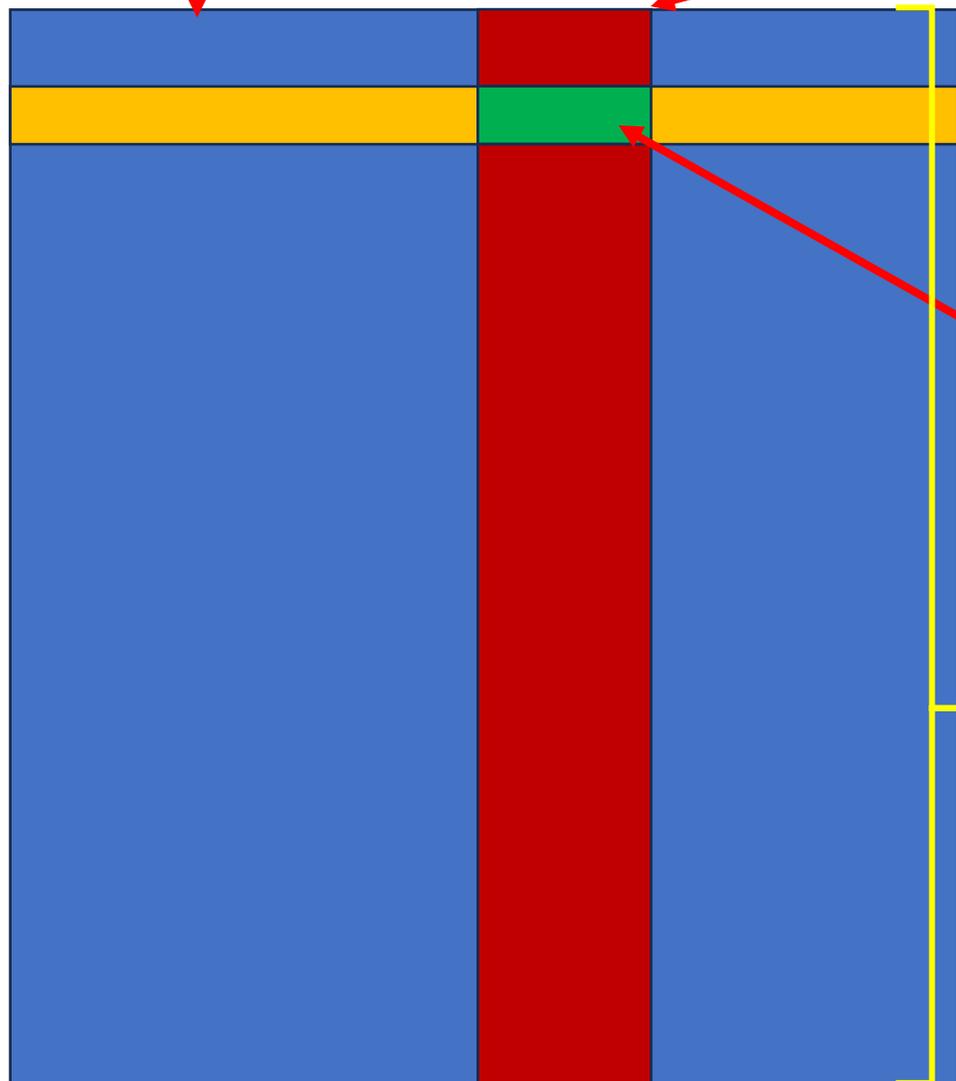
歯本数d= 0 + 0 ...

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

データフレームKlistに列を追加して歯の数を集計します。

Klistデータフレーム



特定の列にアクセスします

```
print(Klist["列名"])
```

特定の要素にアクセスします

```
print(Klist["列名"][行のインデックス番号])
```

データフレーム行の数を数えます

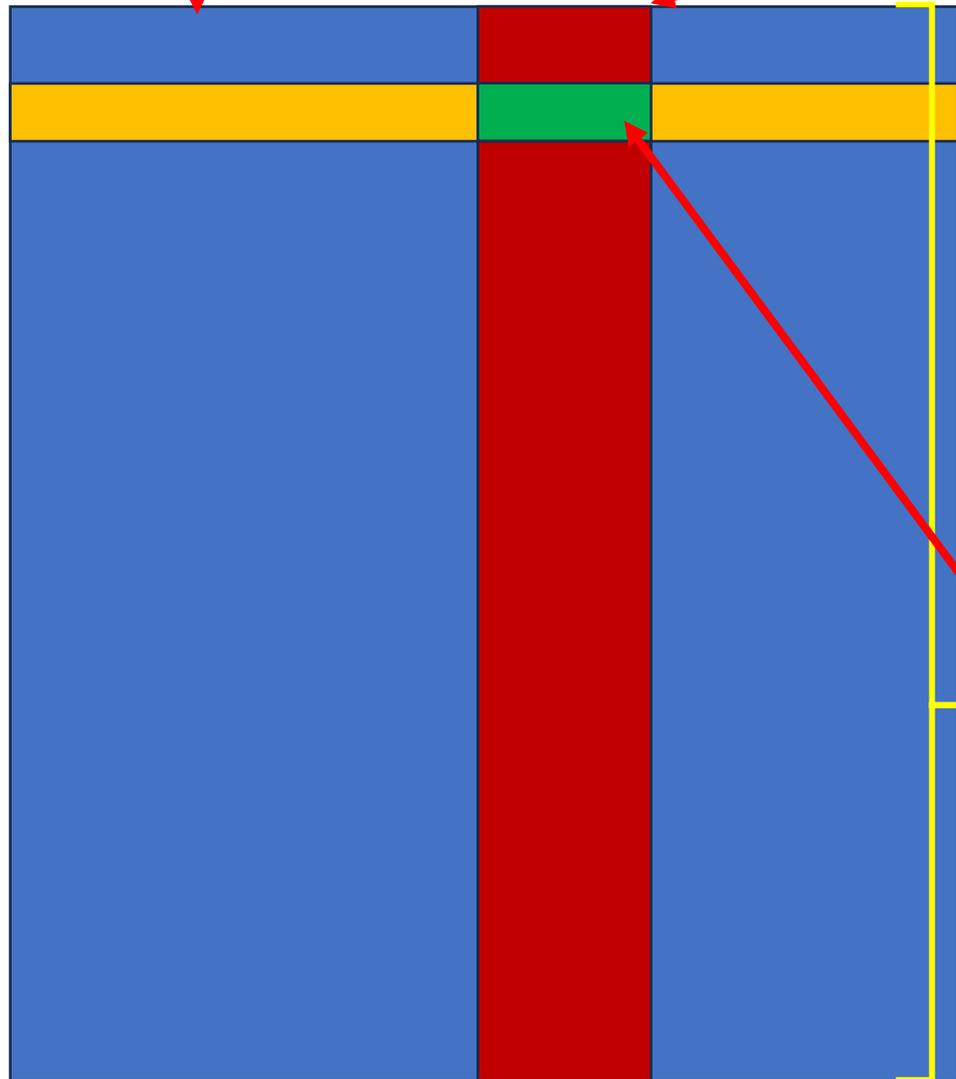
```
print(len(Klist))
```

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

データフレームKlistに列を追加して歯の数を集計します。

Klistデータフレーム



特定の列にアクセスします

```
print(Klist["tooth record"])
```

特定の列の要素にアクセスします

```
print(Klist["tooth record"][0])
```

データフレーム行の数を数えます

```
print(len(Klist))
```


演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

データフレームKlistに列を追加して歯の数を集計します。

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	

```
Klist["tooth exist"] = None  
print(Klist)
```

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

データフレームKlistに列を追加して歯の数を集計します。

tooth record列

tooth exist列

```
Klist["tooth exist"] = None  
print(Klist)
```

	tooth name	tooth record	gender	id	birthmonth	yearmonth	tooth exist
0	A1	残存歯	男	pt_1	1943-01	2020-10	None
1	A2	喪失歯	男	pt_1	1943-01	2020-10	None
2	A3	残存歯	男	pt_1	1943-01	2020-10	None
3	A4	残存歯	男	pt_1	1943-01	2020-10	None
4	A5	残存歯	男	pt_1	1943-01	2020-10	None
...
53019	D4	喪失歯	女	pt_1657	1982-12	2022-12	None
53020	D5	残存歯	女	pt_1657	1982-12	2022-12	None
53021	D6	残存歯	女	pt_1657	1982-12	2022-12	None
53022	D7	喪失歯	女	pt_1657	1982-12	2022-12	None
53023	D8	残存歯	女	pt_1657	1982-12	2022-12	None

[53024 rows x 7 columns]

Colab

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

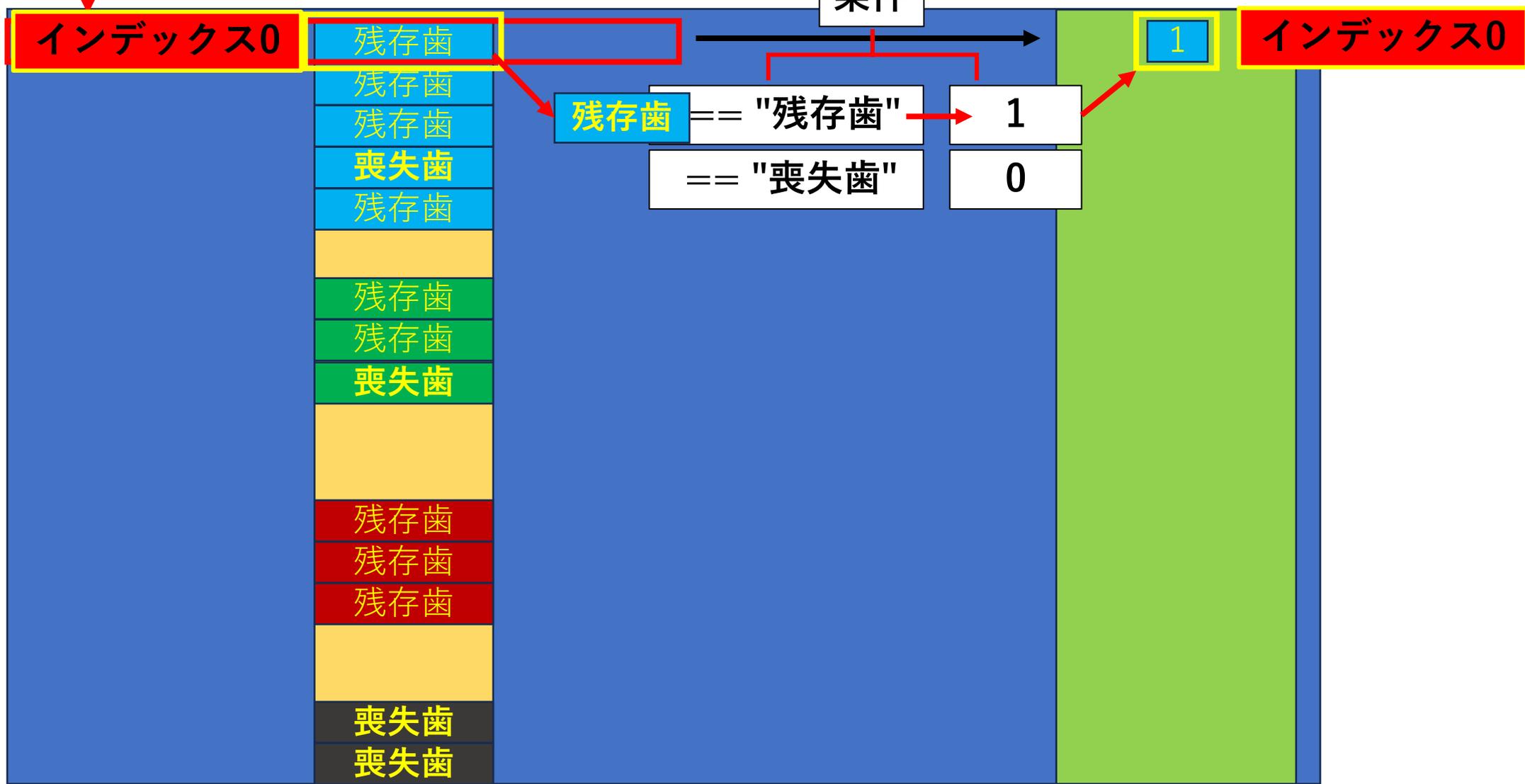
データフレームKlistに列を追加して歯の数を集計します。

Klistデータフレーム

tooth record列

条件

tooth exist列



演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

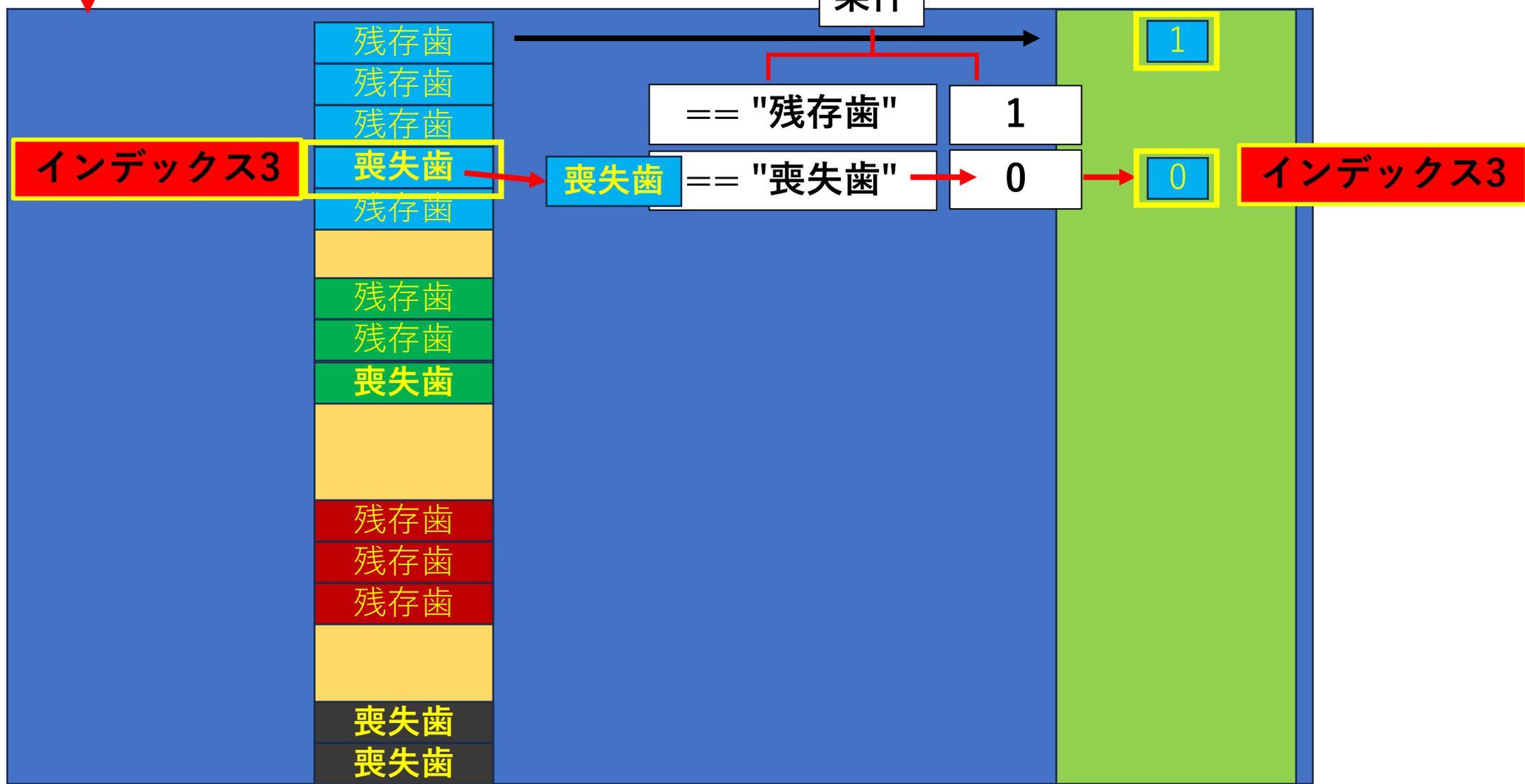
データフレームKlistに列を追加して歯の数を集計します。

Klistデータフレーム

tooth record列

条件

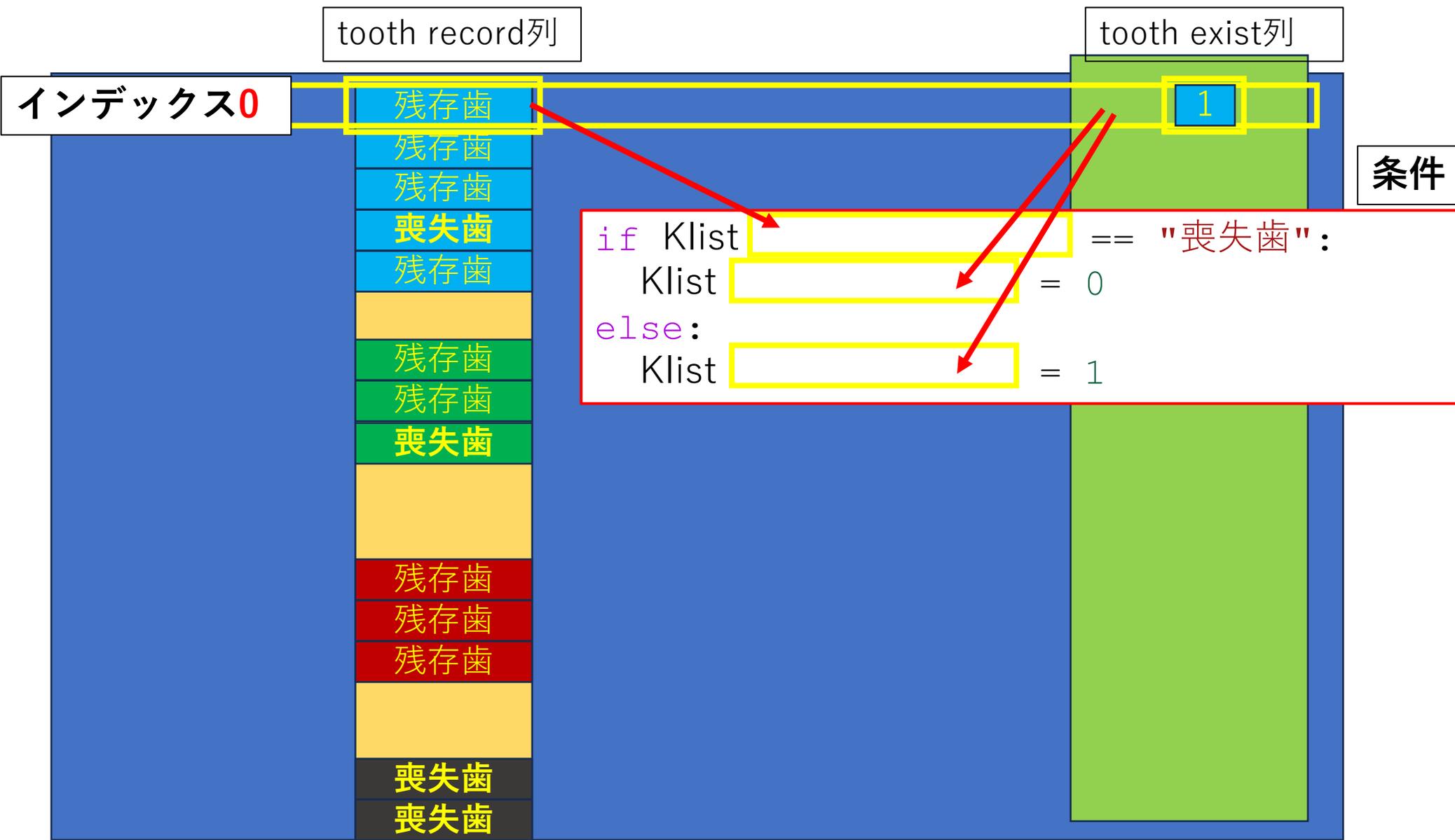
tooth exist列



演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

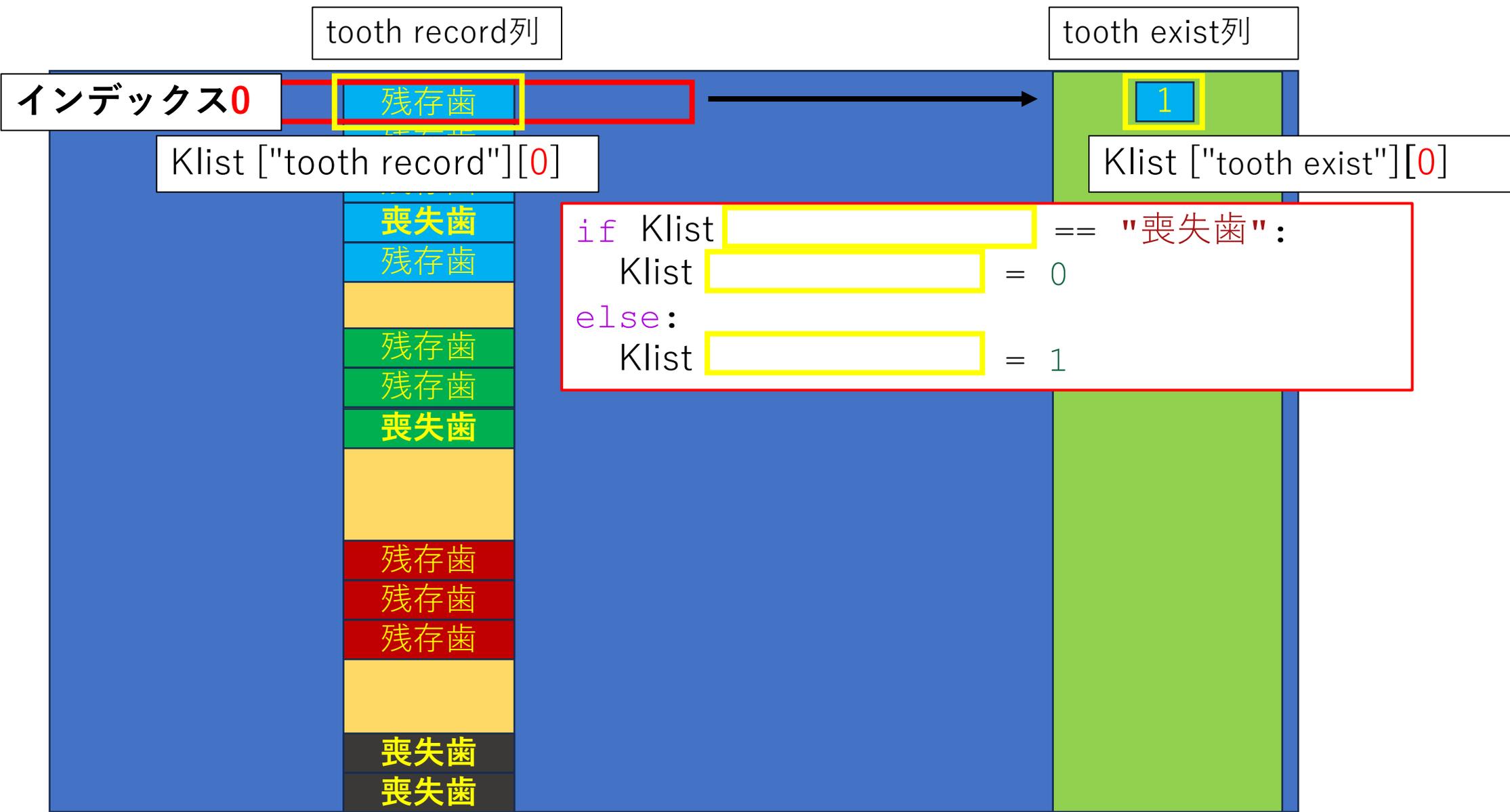
データフレームKlistに列を追加して歯の数を集計します。



演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

データフレームKlistに列を追加して歯の数を集計します。



演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

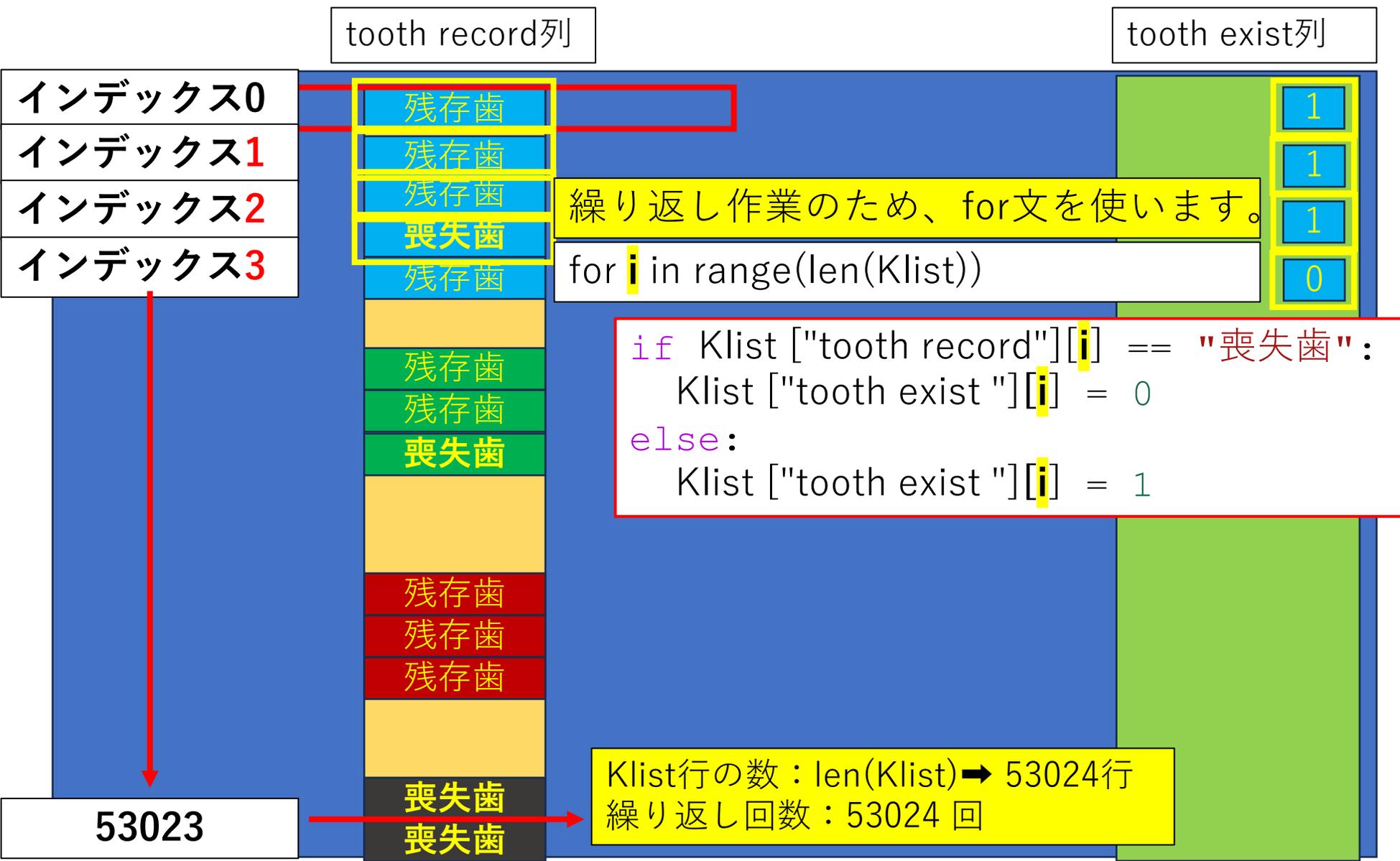
データフレームKlistに列を追加して歯の数を集計します。



演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

データフレームKlistに列を追加して歯の数を集計します。



演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

データフレームKlistに列を追加して歯の数を集計します。

tooth record列

tooth exist列

インデックス0	tooth record列	tooth exist列
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	

```
for i in range(len(Klist))
```

```
if Klist ["tooth record"][i] == "喪失歯":  
    Klist ["tooth exist "][i] = 0  
else:  
    Klist ["tooth exist "][i] = 1
```

```
print(Klist)
```

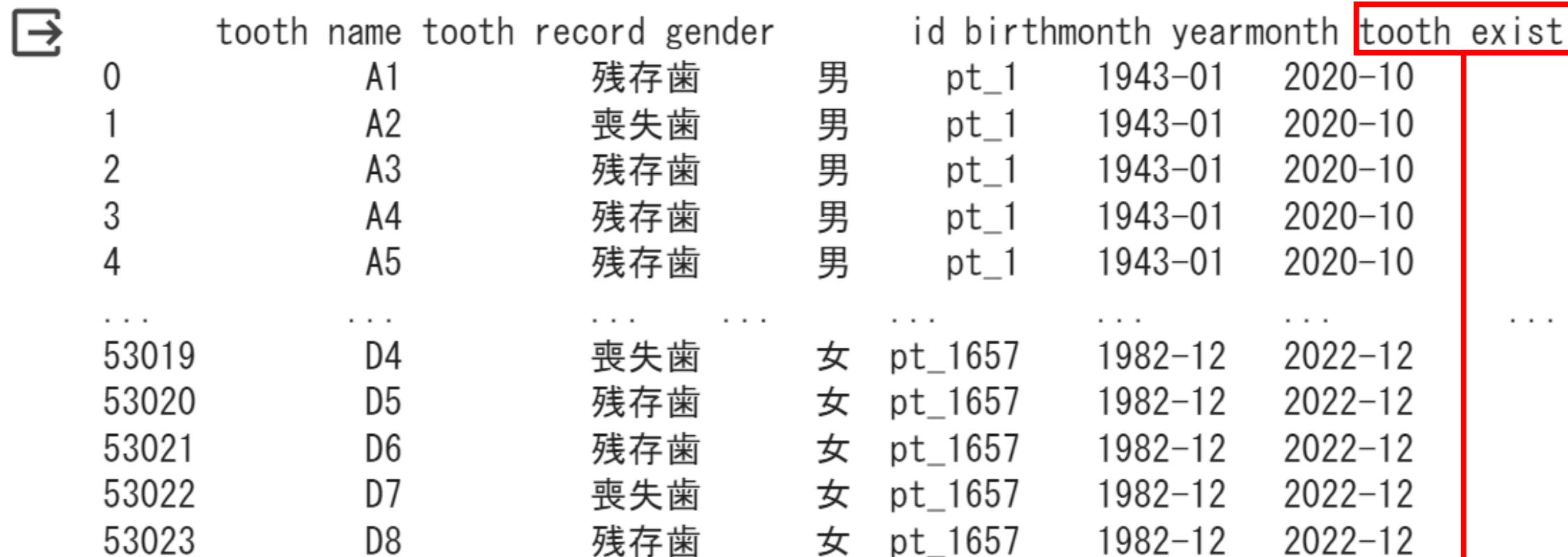
53023

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

データフレームKlistに列を追加して歯の数を集計します。

コード `print(Klist)`



```
tooth name  tooth record  gender  id  birthmonth  yearmonth  tooth exist
0          A1      残存歯      男    pt_1    1943-01    2020-10          1
1          A2      喪失歯      男    pt_1    1943-01    2020-10          0
2          A3      残存歯      男    pt_1    1943-01    2020-10          1
3          A4      残存歯      男    pt_1    1943-01    2020-10          1
4          A5      残存歯      男    pt_1    1943-01    2020-10          1
...         ...         ...         ...         ...         ...         ...
53019      D4      喪失歯      女    pt_1657    1982-12    2022-12          0
53020      D5      残存歯      女    pt_1657    1982-12    2022-12          1
53021      D6      残存歯      女    pt_1657    1982-12    2022-12          1
53022      D7      喪失歯      女    pt_1657    1982-12    2022-12          0
53023      D8      残存歯      女    pt_1657    1982-12    2022-12          1
```

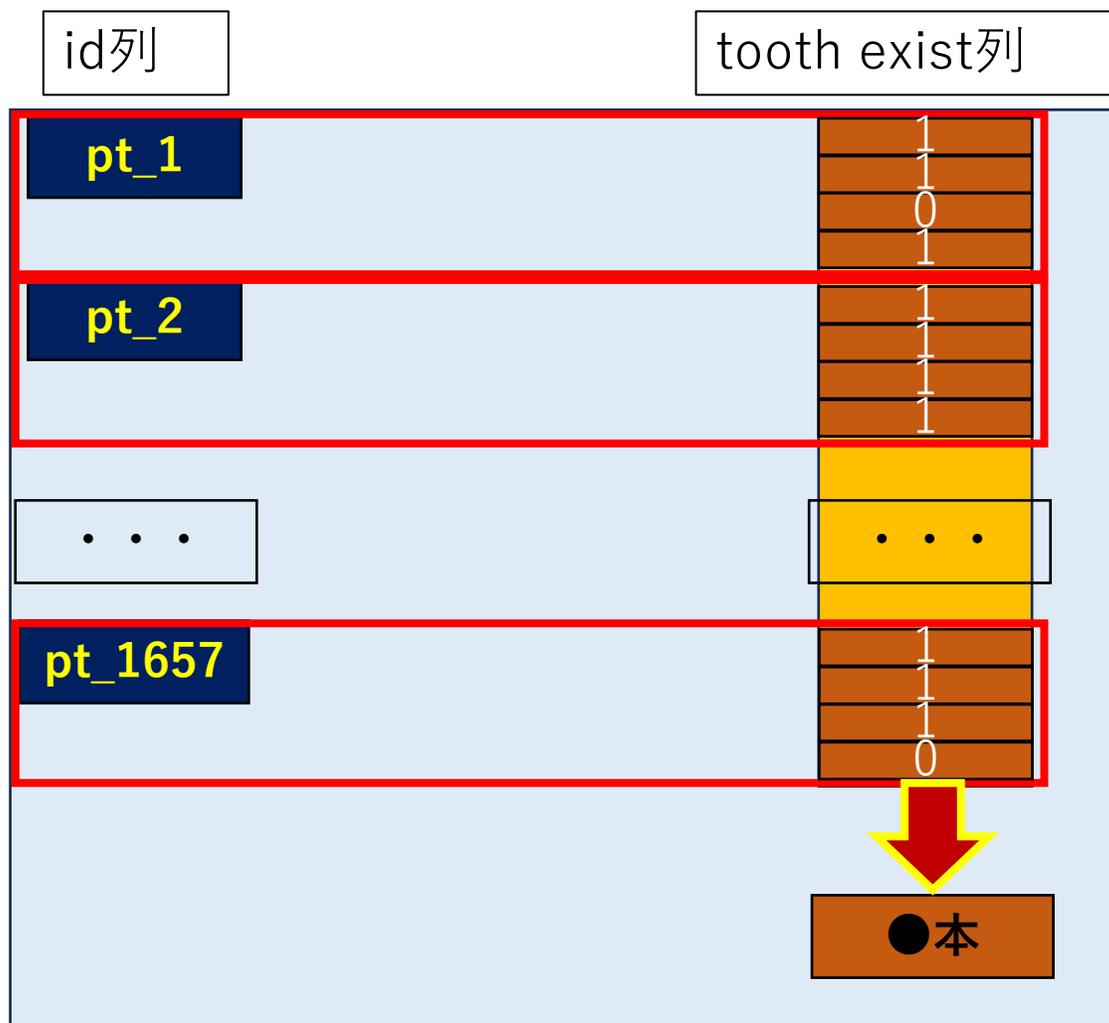
[53024 rows x 7 columns]

Colab

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

データフレームKlistに列を追加して歯の数を集計 (合計) します。



患者ID列指定して患者ごとにデータをグループ化し、それぞれのグループに対して歯の本数を集計したいです。



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。



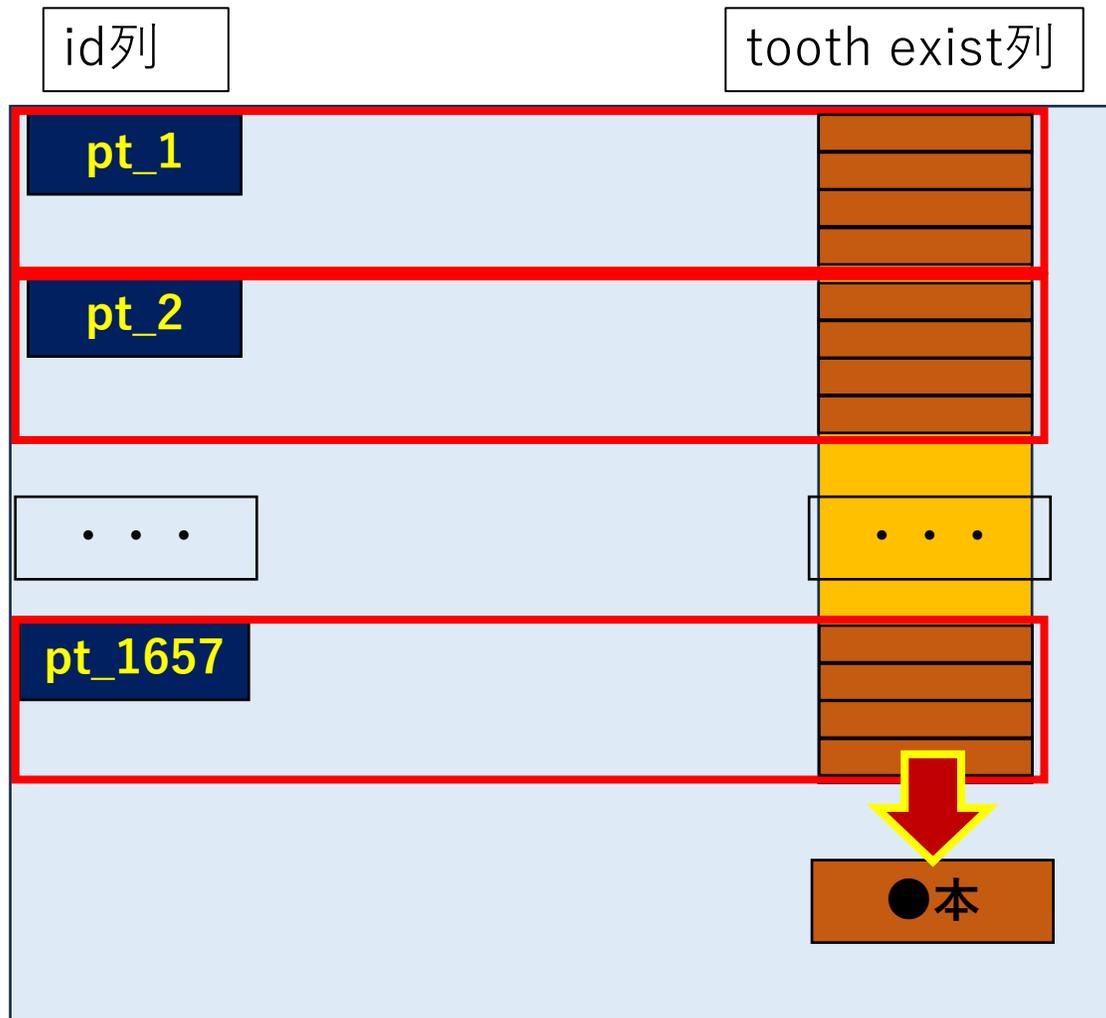
列Aでグループ化し、列Bを集計します。
DataFrame名.groupby("A")["B"].**sum()**

演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

データフレームKlistに列を追加して歯の数を集計 (合計) します。

コード `print(Klist.groupby("id")["tooth exist"].sum())`

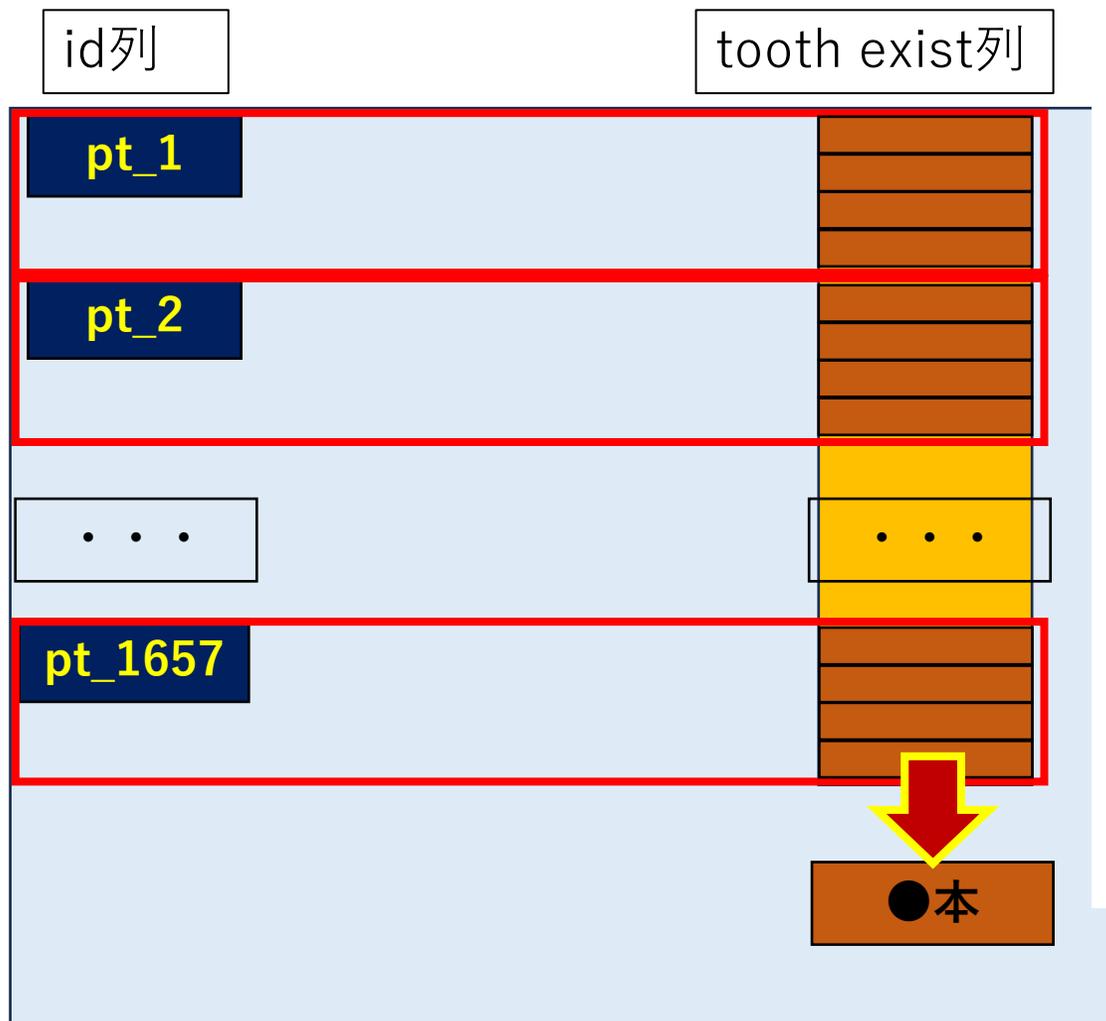


演習6 データクレンジングに必要なライブラリ (Pandas) の応用

歯の数を集計しましょう。

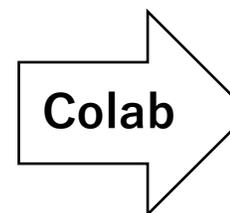
データフレームKlistに列を追加して歯の数を集計 (合計) します。

コード `print(Klist.groupby("id")["tooth exist"].sum())`



患者ID	歯の本数
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,



WebClassで課題を提出してください、締め切りは**1月11日23:59**までです。

課題1： `print()`関数を使って、性別列 (`gender`) のインデックス番号0行のデータを抽出して出力するコードを1行で書いてください。

データフレーム名： `Klist`

性別列の列名： `"gender"`

インデックス番号： `0`