

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

ここにメッセージを入力します...

宛先: 全員

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7

データクレンジングとデータの可視化

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

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

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



The screenshot displays the Google Colaboratory web interface. On the left, a sidebar contains icons for file management and code execution. The main menu bar includes 'ファイル' (File), '編集' (Edit), '表示' (View), '挿入' (Insert), 'ランタイム' (Runtime), 'ツール' (Tools), and 'ヘルプ' (Help). The 'ファイル' menu is open, showing options such as 'ノートブックを新規作成' (Create new notebook), 'ノートブックを開く' (Open notebook) with a keyboard shortcut 'Ctrl+O', 'ノートブックをアップロード' (Upload notebook), '名前の変更' (Rename), 'ドライブにコピーを保存' (Save copy to Drive), 'コピーを GitHub Gist として保存' (Save copy as GitHub Gist), 'GitHub にコピーを保存' (Save copy to GitHub), '保存' (Save) with 'Ctrl+S', '変更履歴' (History), 'ダウンロード' (Download), and '印刷' (Print) with 'Ctrl+P'. Two large red arrows point to the 'ファイル' menu and the 'ノートブックを開く' option. To the right, a video thumbnail titled '3 Cool Google Colab Features' is shown, featuring a man's face and a play button icon.

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

ド + テキスト | ドライブにコピー

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

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。

3 Cool Google Colab Features

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 >

🔍 ノートブックを検索

タイトル	所有者	最終閲覧 ▲	最終更新 ▼	
演習7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 >



または、ここにファイルをドラッグしてください

演習7コード.ipynb

Colab

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

ここにメッセージを入力します...

宛先: 全員

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

・ 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

✓ ■データの集計:
歯の本数を集計します。

 ☐ 患者初診時の年齢

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列 birthmonth列 yearmonth列

	pt_1		1943-01		2020-10
	pt_1		1943-01		2020-10
	pt_1		1943-01		2020-10
	pt_1		1943-01		2020-10
			生年月		初診月
	pt_2		1982-12		2022-12
	pt_2		1982-12		2022-12

初診時の年齢

●歳
●歳
●歳
●歳
●歳
●歳

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列 birthmonth列 yearmonth列

	pt_1		1943-01		2020-10
	pt_1		1943-01		2020-10
	pt_1		1943-01		2020-10
	pt_1		1943-01		2020-10
			生年月		初診月
	pt_2		1982-12		2022-12
	pt_2		1982-12		2022-12

誕生日の年月と初診日の年月を比較します。

条件式で年齢を計算します

```
if 初診月 >= 誕生月:
    age = 初診日の年 - 生まれた年
else:
    age = 初診日の年 - 生まれた年 - 1
```

初診時の年齢

変数：age

●歳
●歳
●歳
●歳
●歳

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列	birthmonth列	yearmonth列
pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
	生年月	初診月
pt_2	1982-12	2022-12
pt_2	1982-12	2022-12

誕生日の年月と初診の年月を比較します。

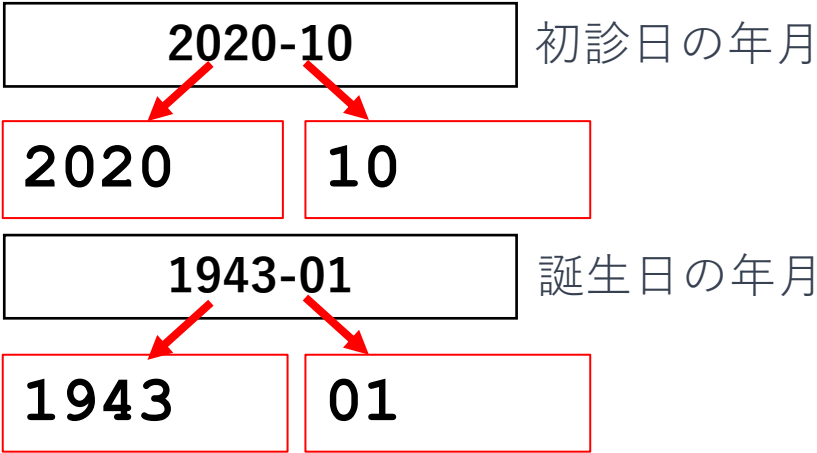
条件式で年齢を計算します

if 初診月 >= 誕生日:

age = 初診の年 - 生まれた年

else:

age = 初診の年 - 生まれた年 - 1



初診時の年齢	age
	●歳
	●歳
	●歳
	●歳
	●歳
	●歳

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列	birthmonth列	yearmonth列
pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
	生年月	初診月
pt_2	1982-12	2022-12
pt_2	1982-12	2022-12

年と月を分解します

split() メソッド
文字列を特定の区切り文字で分割してリストにすることができます。

文字列：1943-01
区切り文字：" - "

コード書き方：

文字列.**split**(" - ")

↓
["1943", "01"]

↓
文字列.**split**(" - ")[0] → ["1943"]
文字列.**split**(" - ")[1] → ["01"]

演習7データクレンジングとデータの可視化

患者年齢を求めましょう


Klistデータフレーム

id列

birthmonth列

yearmonth列

年と月を分解します

	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
		生年月	初診月
	pt_2	1982-12	2022-12
	pt_2	1982-12	2022-12

インデックス番号0

文字列：
生年月：Klist["birthmonth"][0]

文字列：
初診月：Klist["yearmonth"][0]

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列

birthmonth列

yearmonth列

年と月を分解します

	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
		生年月	初診月
	pt_2	1982-12	2022-12
	pt_2	1982-12	2022-12

インデックス番号0

生年月：Klist["birthmonth"][0]
Klist["birthmonth"][0].split("-")
1943-01 → ["1943", "01"]

初診月：Klist["yearmonth"][0]
Klist["yearmonth"][0].split("-")
2020-10 → ["2020", "10"]

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列		birthmonth列	yearmonth列
pt_1		1943-01	2020-10
		1943-01	2020-10
		1943-01	2020-10
		1943-01	2020-10
		生年月	初診月
pt_2		1982-12	2022-12
		1982-12	2022-12

年と月を分解します

インデックス番号0

["1943", "01"]

Klist["birthmonth"][0].split("-")[0]

1943

Klist["birthmonth"][0].split("-")[1]

01

["2020", "10"]

Klist["yearmonth"][0].split("-")[0]

2020

Klist["yearmonth"][0].split("-")[1]

10

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

	id列	birthmonth列	yearmonth列
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
		生年月	初診月
	pt_2	1982-12	2022-12
	pt_2	1982-12	2022-12

年と月を分解します

コード

```
print(Klist["birthmonth"][0].split("-")[0])
1943
print(Klist["birthmonth"][0].split("-")[1])
01
print(Klist["yearmonth"][0].split("-")[0])
2020
print(Klist["yearmonth"][0].split("-")[1])
10
```

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

	id列	birthmonth列	yearmonth列
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
		生年月	初診月
	pt_2	1982-12	2022-12
	pt_2	1982-12	2022-12

年と月を分解します

コード

```
print(Klist["birthmonth"][0].split("-")[0])
1943
print(Klist["birthmonth"][0].split("-")[1])
01
print(Klist["yearmonth"][0].split("-")[0])
2020
print(Klist["yearmonth"][0].split("-")[1])
10
```

文字列リストの要素であるため、型は文字です。比較演算子で比較するため、型を文字から整数に変換する必要があります。

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

	id列	birthmonth列	yearmonth列
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
		生年月	初診月
	pt_2	1982-12	2022-12
	pt_2	1982-12	2022-12

年と月をの型を整数に変換します

int() 関数は文字列の値を、整数に変換します。

コードの書き方：int(文字列)

コード 文字列 ➡ 整数

```
print(int(Klist["birthmonth"][0].split("-")[0]))
1943

print(int(Klist["birthmonth"][0].split("-")[1]))
01

print(int(Klist["yearmonth"][0].split("-")[0]))
2020

print(int(Klist["yearmonth"][0].split("-")[1]))
10
```

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

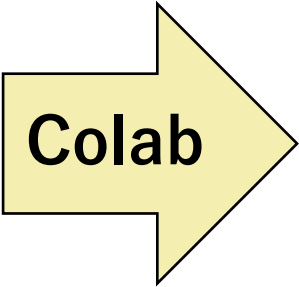
id列		birthmonth列	yearmonth列
pt_1	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
		生年月	初診月
pt_2	pt_2	1982-12	2022-12
	pt_2	1982-12	2022-12
		インデックス番号53023	

年と月をの型を整数に変換します

int() 関数は文字列の値を、整数に変換します。

コードの書き方：int(文字列)

コード 文字列 ➡ 整数



```
print(int(Klist["birthmonth"][53023].split("-")[0]))
1982
print(int(Klist["birthmonth"][53023].split("-")[1]))
12
print(int(Klist["yearmonth"][53023].split("-")[0]))
2022
print(int(Klist["yearmonth"][53023].split("-")[1]))
12
```

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列 birthmonth列 yearmonth列

誕生日の年月と初診の年月を比較します。

初診時の年齢

pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
	生年月	初診月
pt_2	1982-12	2022-12
pt_2	1982-12	2022-12

インデックス番号0



インデックス番号0

●歳
●歳
●歳
●歳

●歳
●歳

コード：

条件式で年齢を計算します
if 初診月 >= 誕生日：
 age =初診の年-生まれた年
else：
 age =初診の年-生まれた年-1

条件式で年齢を計算します = ●歳
if visitmonth >= birthmonth：
 age = visityear - birthyear
else：
 age = visityear - birthyear -1

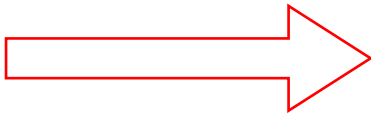
演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

	id列	birthmonth列	yearmonth列
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
		生年月	初診月
	pt_2	1982-12	2022-12
	pt_2	1982-12	2022-12

インデックス番号0



インデックス番号0

コード：

```
birthyear = int(Klist["birthmonth"][0].split("-")[0])
birthmonth = int(Klist["birthmonth"][0].split("-")[1])
visityear = int(Klist["yearmonth"][0].split("-")[0])
visitmonth = int(Klist["yearmonth"][0].split("-")[1])
```

```
if visitmonth >= birthmonth:
    age = visityear - birthyear
else:
    age = visityear = birthyear - 1
```

print(age) = ●歳

初診時の年齢
●歳
●歳
●歳
●歳
●歳
●歳

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

	id列	birthmonth列	yearmonth列
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
	pt_1	1943-01	2020-10
		生年月	初診月
	pt_2	1982-12	2022-12
	pt_2	1982-12	2022-12

インデックス番号0

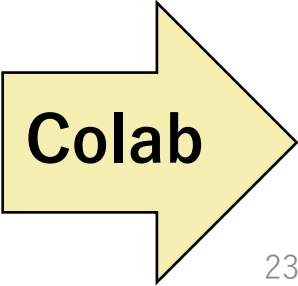
コード：

```
条件式で年齢を計算します
if 10 >= 01:
    age =2020-1943
else:
    age =初診の年-生まれた年-1
```

= 77歳

初診時の年齢

77歳
●歳
●歳
●歳
●歳
●歳



演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列 birthmonth列 yearmonth列

pt_1	1943-01	2020-10	インデックス番号0
------	---------	---------	-----------

pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
pt_1	1943-01	2020-10
	生年月	初診月
pt_2	1982-12	2022-12
pt_2	1982-12	2022-12

コード：

条件式で年齢を計算します
if 初診月 >= 誕生月：
 age =初診の年-生まれた年
else：
 age =初診の年-生まれた年-1

初診時の年齢：age

77歳
●歳
●歳
●歳
●歳
●歳

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

コード書き方：

新しいageという列を追加

初診時の年齢：age

id列		birthmonth列		yearmonth列	
pt_1	pt_1	1943-01	2020-10	インデックス番号0	
	pt_1	1943-01	2020-10	インデックス番号1	
	pt_1	1943-01	2020-10	インデックス番号2	
	pt_1	1943-01	2020-10	インデックス番号3	
		生年月	初診月		
pt_2	pt_2	1982-12	2022-12		
	pt_2	1982-12	2022-12		

for文：繰り返す作業

if文：年齢計算作業

ageという列に値を代入

77歳
●歳
●歳
●歳
●歳
●歳

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列	birthmonth	yearmonth	age
pt_1	1943-01	2020-10	None
pt_1	1943-01	2020-10	None
pt_1	1943-01	2020-10	None
pt_1	1943-01	2020-10	None
	生年月	初診月	
pt_2	1982-12	2022-12	None
pt_2	1982-12	2022-12	None

コード：

新しいageという列を追加

```
Klist["age"] = None
print(Klist)
```

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列	birthmonth	yearmonth	age
pt_1	1943-01	2020-10	None
pt_1	1943-01	2020-10	None
pt_1	1943-01	2020-10	None
pt_1	1943-01	2020-10	None
	生年月	初診月	
pt_2	1982-12	2022-12	None
pt_2	1982-12	2022-12	None

コード：

新しいageという列を追加

```
Klist["age"] = None
print(Klist)
```

for文：繰り返す作業

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

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列	birthmonth	yearmonth	age
pt_1	1943-01	2020-10	None
pt_1	1943-01	2020-10	None
pt_1	1943-01	2020-10	None
pt_1	1943-01	2020-10	None
	生年月	初診月	
pt_2	1982-12	2022-12	None
pt_2	1982-12	2022-12	None

コード：

新しいageという列を追加

```
Klist["age"] = None
print(Klist)
```

for文：繰り返す作業

```
for i in range(len(Klist)):
    birthmonth = int(Klist['birthmonth'][i].split('-')[1])
    birthyear = int(Klist['birthmonth'][i].split('-')[0])
    visitmonth = int(Klist["yearmonth"][i].split('-')[1])
    visityear = int(Klist["yearmonth"][i].split('-')[0])
    if visitmonth >= birthmonth:
        age = visityear - birthyear
    else:
        age = visityear - birthyear-1
```

演習7データクレンジングとデータの可視化

患者年齢を求めましょう

Klistデータフレーム

id列	birthmonth	yearmonth	age
pt_1	1943-01	2020-10	●歳
pt_1	1943-01	2020-10	●歳
pt_1	1943-01	2020-10	●歳
pt_1	1943-01	2020-10	●歳
	生年月	初診月	
pt_2	1982-12	2022-12	●歳
pt_2	1982-12	2022-12	●歳

コード：

新しいageという列を追加

```
Klist["age"] = None
print(Klist)
```

for文：繰り返す作業

```
for i in range(len(Klist)):
    birthmonth = int(Klist['birthmonth'][i].split('-')[1])
    birthyear = int(Klist['birthmonth'][i].split('-')[0])
    visitmonth = int(Klist["yearmonth"][i].split('-')[1])
    visityear = int(Klist["yearmonth"][i].split('-')[0])

    if visitmonth >= birthmonth:
        age = visityear - birthyear
    else:
        age = visityear - birthyear-1
```

ageという列に値を代入

```
Klist["age"][i]=age
```

演習7データクレンジングとデータの可視化

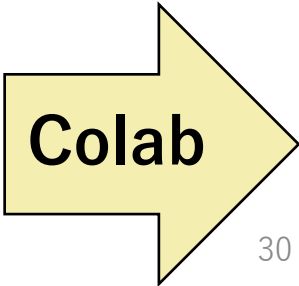
患者年齢を求めましょう

```
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		

	tooth	exist	age
0		1	77
1		0	77
2		1	77
3		1	77
4		1	77
...
53019		0	40
53020		1	40
53021		1	40
53022		0	40
53023		1	40

[53024 rows x 8 columns]



演習7 データクレンジングとデータの可視化

散布図を作しましょう

患者年齢と歯の本数の関係

縦軸y：
歯の本数

● 患者pt_n

● 患者pt_m

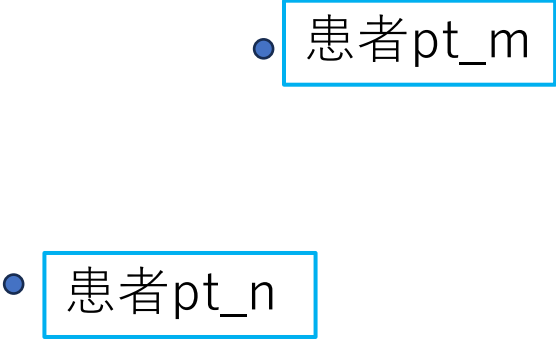
横軸x：患者年齢

演習7データクレンジングとデータの可視化

散布図を作しましょう

患者年齢と歯の本数の関係

縦軸y：
歯の本数



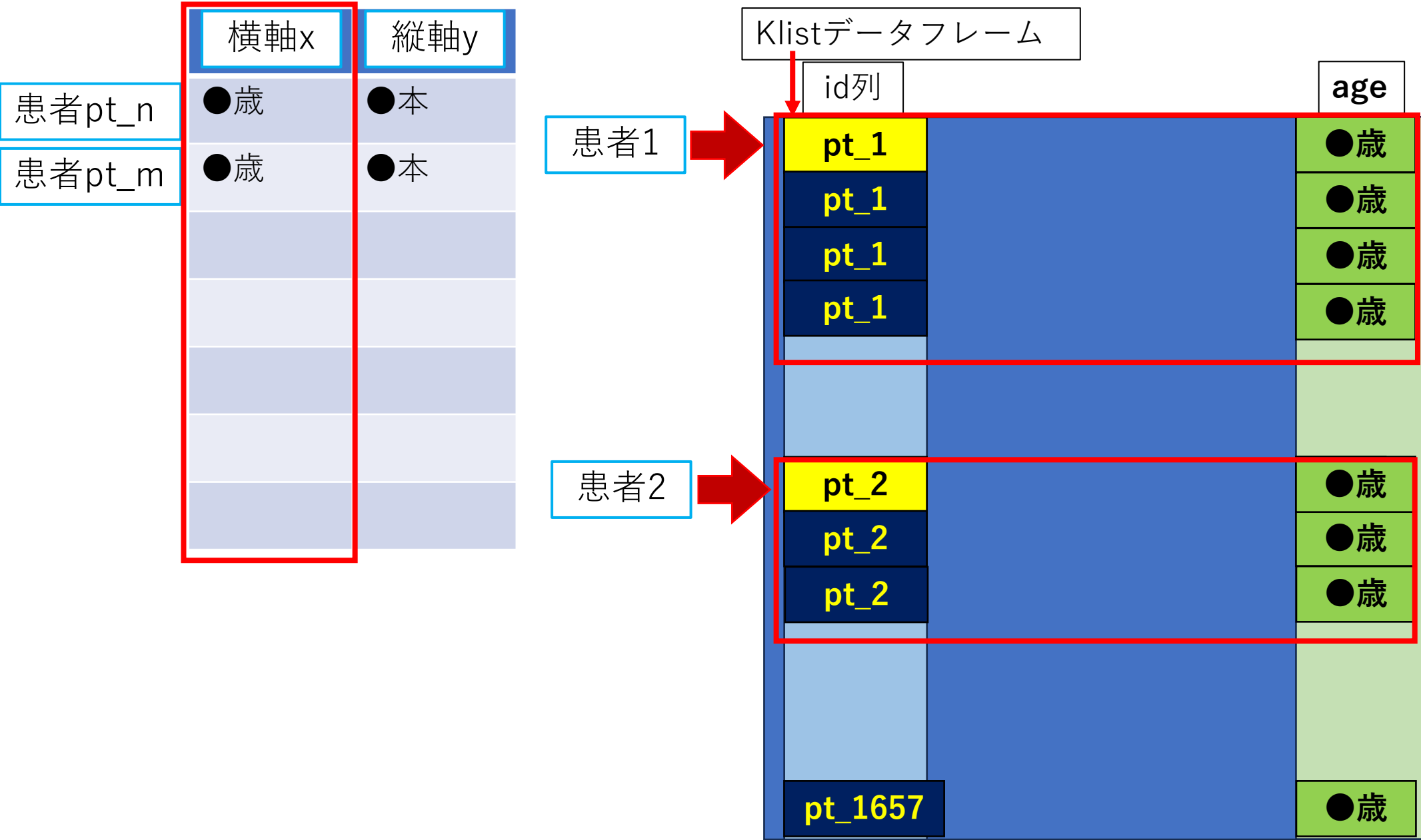
横軸x：患者年齢

	横軸x	縦軸y
患者pt_n	● 歳	● 本
患者pt_m	● 歳	● 本
	numpy 配列	numpy 配列

演習7データクレンジングとデータの可視化

散布図を作しましょう

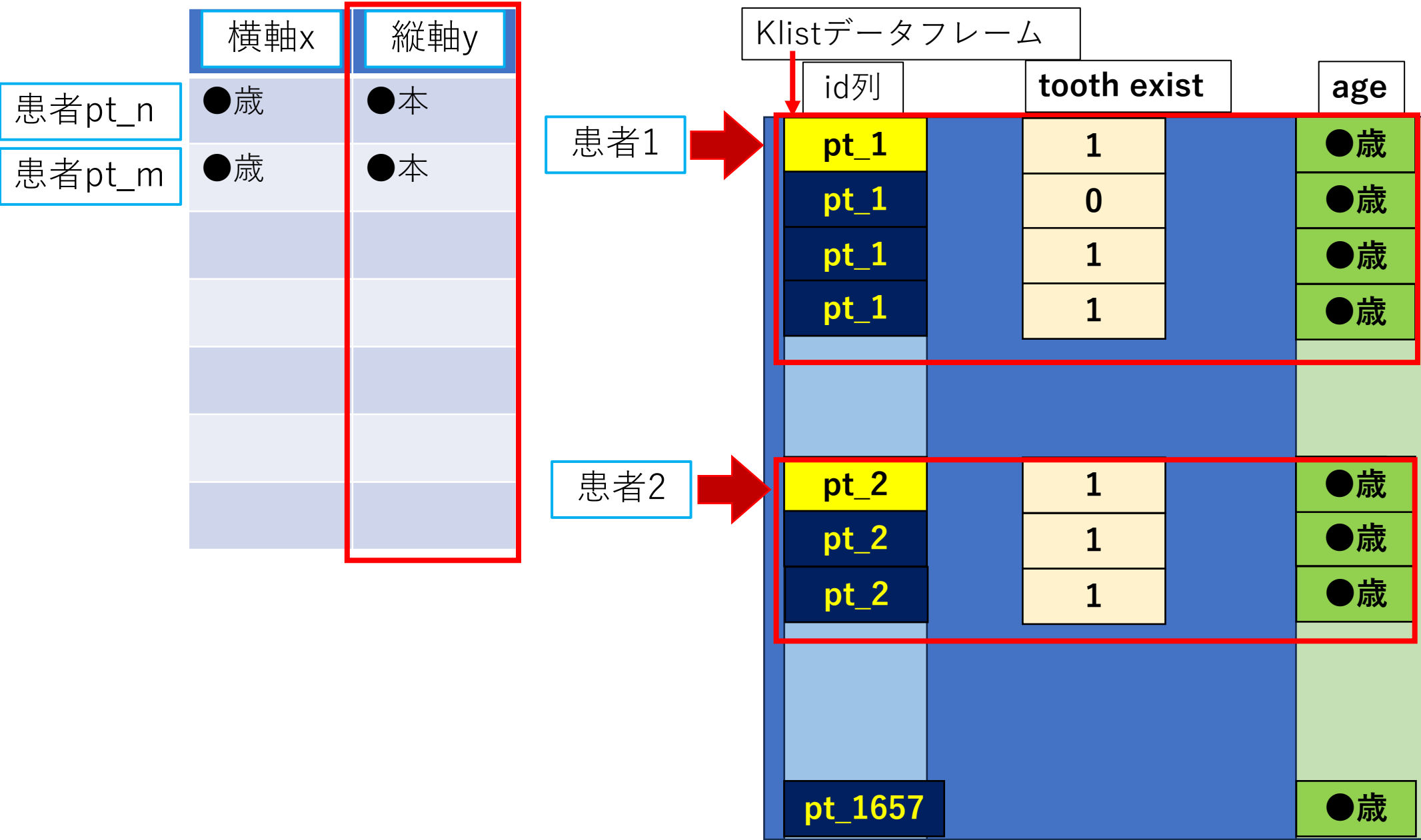
患者年齢と歯の本数の関係



演習7データクレンジングとデータの可視化

散布図を作しましょう

患者年齢と歯の本数の関係



演習7データクレンジングとデータの可視化

散布図を作しましょう

患者年齢と歯の本数の関係

	横軸x	縦軸y
患者pt_n	●歳	●本
患者pt_m	●歳	●本
	numpy 配列	numpy 配列

横軸x

コード

```
x = Klist.groupby("id")["age"].mean()  
print(x)
```

縦軸y

コード

```
y = Klist.groupby("id")["tooth exist"].sum()  
print(y)
```

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"].**mean()**

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7データクレンジングとデータの可視化


散布図を作しましょう

患者年齢と歯の本数の関係

	横軸x	縦軸y
患者pt_n	●歳	●本
患者pt_m	●歳	●本
	numpy 配列	numpy 配列

横軸x コード

```
x = Klist.groupby("id")["age"].mean()  
print(x)
```



id	
pt_1	77.0
pt_10	76.0
pt_100	72.0
pt_1000	54.0
pt_1001	56.0
...	
pt_995	55.0
pt_996	55.0
pt_997	55.0
pt_998	53.0
pt_999	55.0

Name: age, Length: 1657, dtype: float64

演習7データクレンジングとデータの可視化

散布図を作しましょう

患者年齢と歯の本数の関係

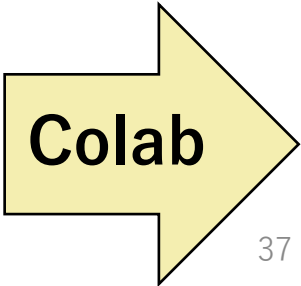
	横軸x	縦軸y
患者pt_n	●歳	●本
患者pt_m	●歳	●本
	numpy 配列	numpy 配列

縦軸y コード

```
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



演習7データクレンジングとデータの可視化

散布図を作しましょう

患者年齢と歯の本数の関係

	横軸x	縦軸y	コード
患者pt_n	●歳	●本	<pre>import numpy as np x = np.array(x) y = np.array(y)</pre>
患者pt_m	●歳	●本	
	numpy 配列	numpy 配列	

演習7 データクレンジングとデータの可視化

散布図を作しましょう

患者年齢と歯の本数の関係

コード

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
```

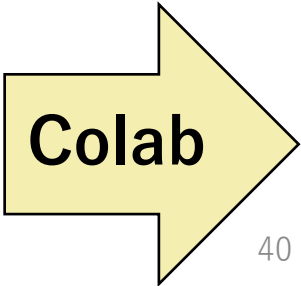
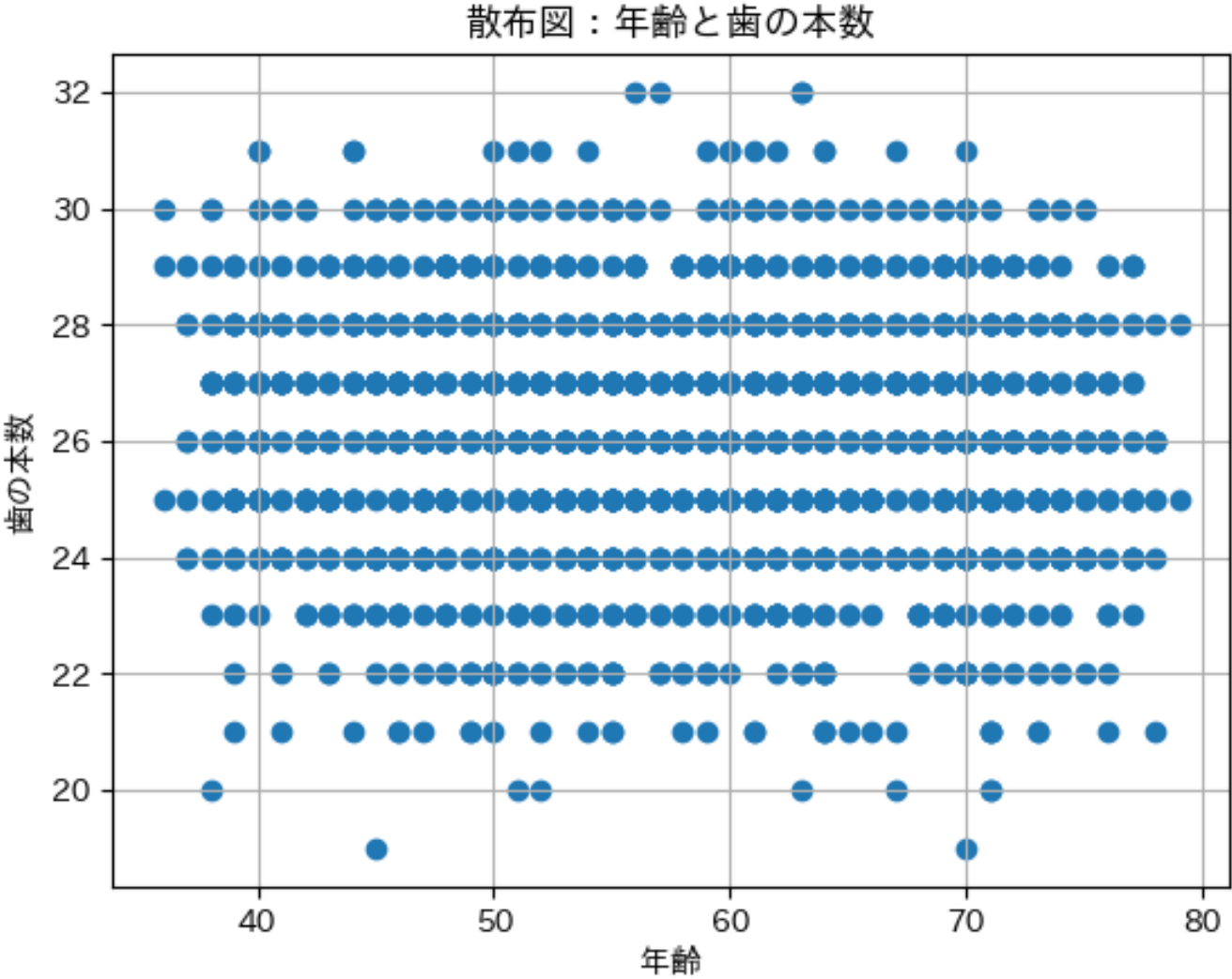
matplotlib ライブラリで散布図を作成します。

```
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()
```

演習7 データクレンジングとデータの可視化

散布図を作しましょう

患者年齢と歯の本数の関係



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、コードを1行に書いてください。

Klistデータフレームのid列のインデックス番号1行の文字データ（pt_1）をsplit()メソッドを使って、アルファベットと数字の部分を分解し、また数字の部分を整数に変換してください。

データフレーム名： Klist

列名： "id"

インデックス番号： 1

文字データ： "pt_1"

区切り文字： _

課題2： matplotlibのpyplotモジュールをインポートするコードを書いてください。略称は、pltとします。