

自分で触ってよくわかる

データの読み込みの話：

ファイルを解析環境に読み込む

神戸市立医療センター中央市民病院
臨床研究推進センター

宮越 千智

今回の学習目標

- ✓「整然とした」データシートを作成できる
- ✓ クラウド上の解析環境にデータファイルをアップロードできる
- ✓ データファイルを読み込んで、内容を確認できる

復習:

整然としたデータセットとは

- 1つの観察単位が、1つの行に収められている
- 1つの変数が、1つの列に収められている
- 1つの値が、1つのセル(=マス)に収められている

ID	年齢	性別	身長	体重	...	転帰
1						
2						
...						
100						

復習：解析用データセット作成の注意点

人が見やすいかよりも機械が間違えずに読み込めるか

- 半角英数のみ使用の方が無難
- 見出し行は1行のみに
- 空欄にはNAを入れる方がよい
- 自由欄に書いたコメントはデータとして使えない(今のところ)

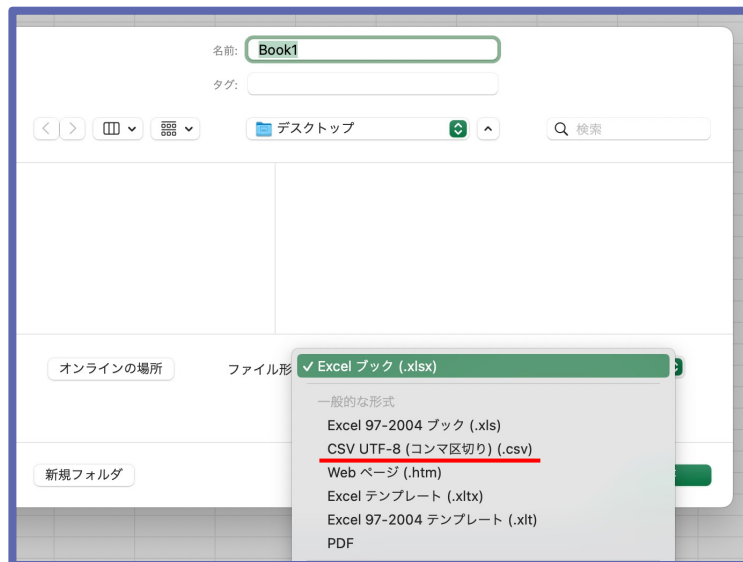
ID	入院時データ			...		コメント
	年齢	性別	身長/体重	...	転帰	
1	65	M	172/65	...	生存	研究参加に同意あり
2	75	F	155/58	...		途中でフォロー途切れた
...
100	80	F	161/45	...	死亡	...

復習：準備するデータファイル

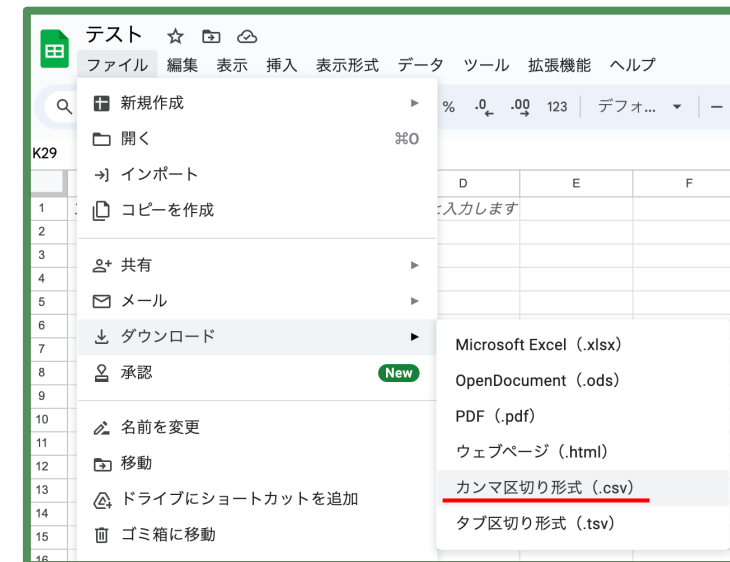
基本はcsvファイル

- csvファイルとは：
カンマで区切られた値(comma-separated values)が収められたテキストファイル
- スプレッドシートでcsv形式を選択して保存することで作成できる

Microsoft Excelの場合：



Googleスプレッドシートの場合：



復習：準備するデータファイル

半角英数のみ使用する方法が無難

- 全角文字はエンコード方法の違いで文字化けするリスクがある
(半角文字は大丈夫)

A20	▼		<i>fx</i>
	A	B	
1	あいうえお		
2	abcde		
3			
4			
5			



	A	B
1	縋ゅ > 縋 ∴ 縋	
2	abcde	
3		
4		
5		

復習：準備するデータファイル 使用を避けた方がよい記号・文字

計算記号として使われるもの	+ - * / % ^
プログラム言語にとって意味があるもの	\$.(ピリオド) ,(カンマ) : ; “ ” ‘ ’ #
ブール値として使われるもの	T, True, TRUE F, False, FALSE
存在が分かりにくいもの	(スペース)

安心して使える記号は「_(アンダースコア)」だけ

復習：準備するデータファイル 変数名の付け方

- 簡潔で、中身が分かりやすい名前をつける
 - ✓ 例: age2(2値化?バージョンが2?) → age_cat(カテゴリー化した年齢)
- 接頭辞、接尾辞を工夫すると一括操作がしやすい
 - ✓ 例: date_adm, date_event(date_を使って日付を表す変数を一括で選択できる)
- 2値変数はどちらに1を割り当てているかが分かれると便利
 - ✓ 例: sex(男女どちらが1?) → female(女性=1と想像しやすい)
- 統一感があると読みやすい
 - ✓ キャメルケース(dateAdmなど):最初は小文字で、あとの単語の先頭は大文字にする
 - ✓ スネークケース(date_adm):アンダースコアで単語を区切る

復習：準備するデータファイル 変数定義書を用意しよう

- データセットに含まれている変数について分かりやすくまとめた一覧表

番号	データセット名	変数名	タイプ	入力形式	内容	備考
1	DATA2024	age	数値		年齢(歳)	
2	DATA2024	age_cat	数値	1/2/3/4/5	年齢(カテゴリー化)	1: 50歳未満, 2: 50代, ...
3	DATA2024	gender	文字	M/F	性別	M: 男性, F: 女性

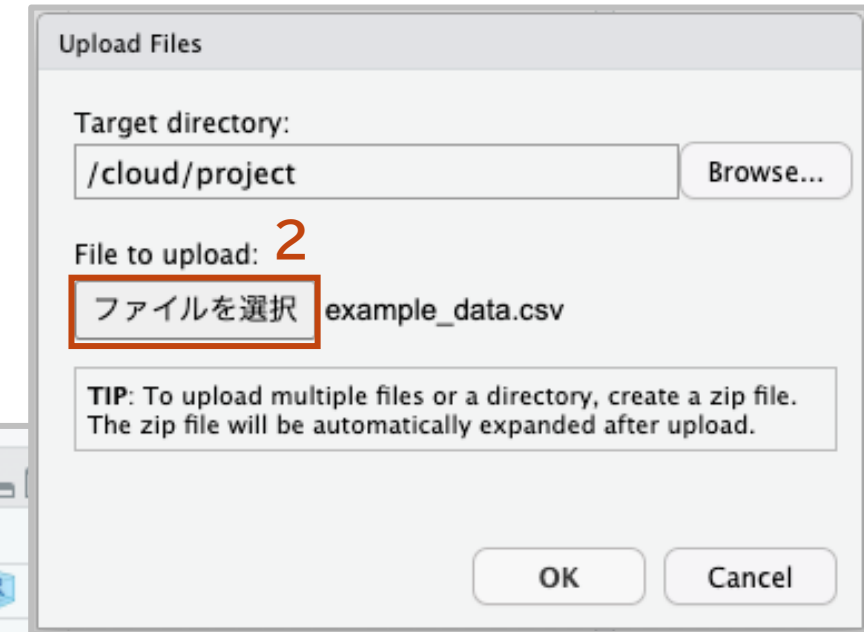
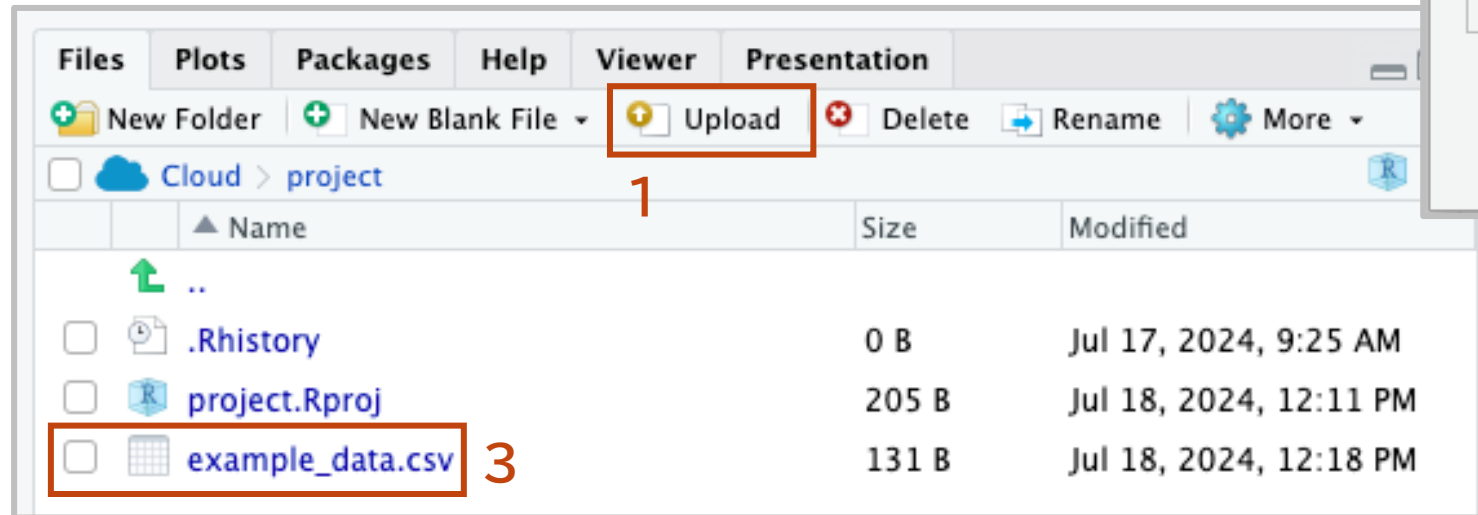
課題3：データシート作成

- 以下の条件にあう「整然とした」データシートを作成してください
 - ✓ 変数：ID、性別、年齢、BMI
 - ✓ 欠測のないデータを10人分
- 半角英数のみを使用したものと、全角漢字を含んだものの2通りを作成して、それぞれ名前を付けてcsv形式で保存してください
 - ✓ 半角英数のみのファイル → example_data.csv
 - ✓ 漢字を含んだファイル → 練習用データ.csv

Rを使いたい人 

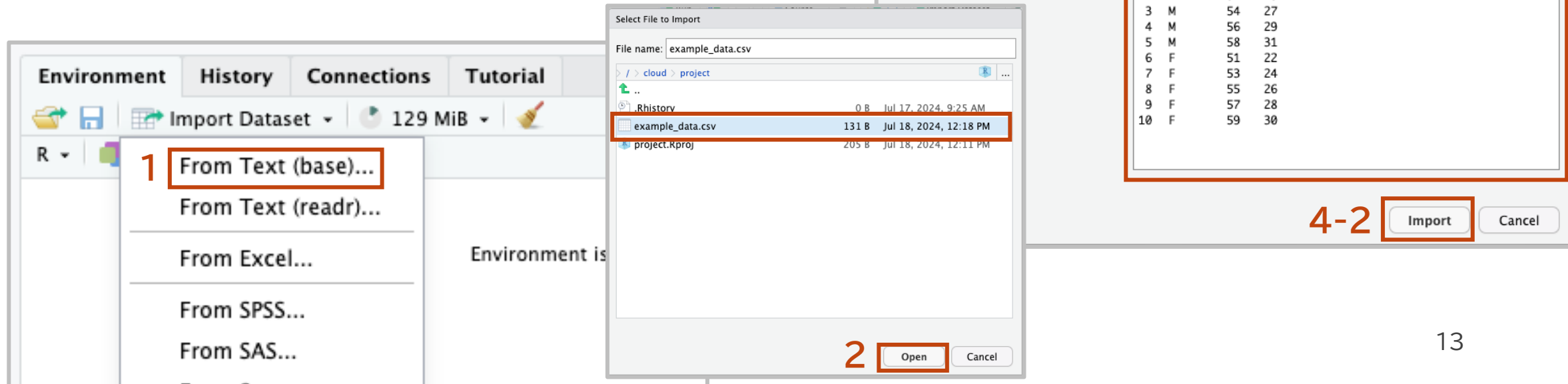
Rを使いたい人: データファイルをアップロードする

1. 右下の [Files] タブで [Upload] を押す
2. [ファイルを選択] を押して
アップロードしたいファイルを選択する
3. 目的のファイルがアップロードされたことを確認する



Rを使いたい人: データファイルを読み込む(ボタン操作編)

1. 右上の [Environment] タブで [Import Dataset] に進み、[From Text (base)…] を押す
2. 目的のファイルを選択して [Open] を押す
3. [Data Frame] で想定どおりに認識されていることを確認する
4. R内で用いるデータ名を付けて [Import] を押す



Environment History Connections Tutorial

Import Dataset 129 MiB

1 From Text (base)...

From Text (readr)...

From Excel...

From SPSS...

From SAS...

Environment is

Select File to Import

File name: example_data.csv

> / > cloud > project

..

.Rhistory 0 B Jul 17, 2024, 9:25 AM

example_data.csv 131 B Jul 18, 2024, 12:18 PM

project.Rproj 205 B Jul 18, 2024, 12:11 PM

2 Open Cancel

Import Dataset

Name data 4-1

Encoding Automatic

Heading Yes No

Row names Automatic

Separator Comma

Decimal Period

Quote Double (")

Comment None

na.strings NA

☐ Strings as factors

Input File

ID, gender, age, BMI

1, M, 50, 23

2, M, 52, 25

3, M, 54, 27

4, M, 56, 29

5, M, 58, 31

6, F, 51, 22

7, F, 53, 24

8, F, 55, 26

9, F, 57, 28

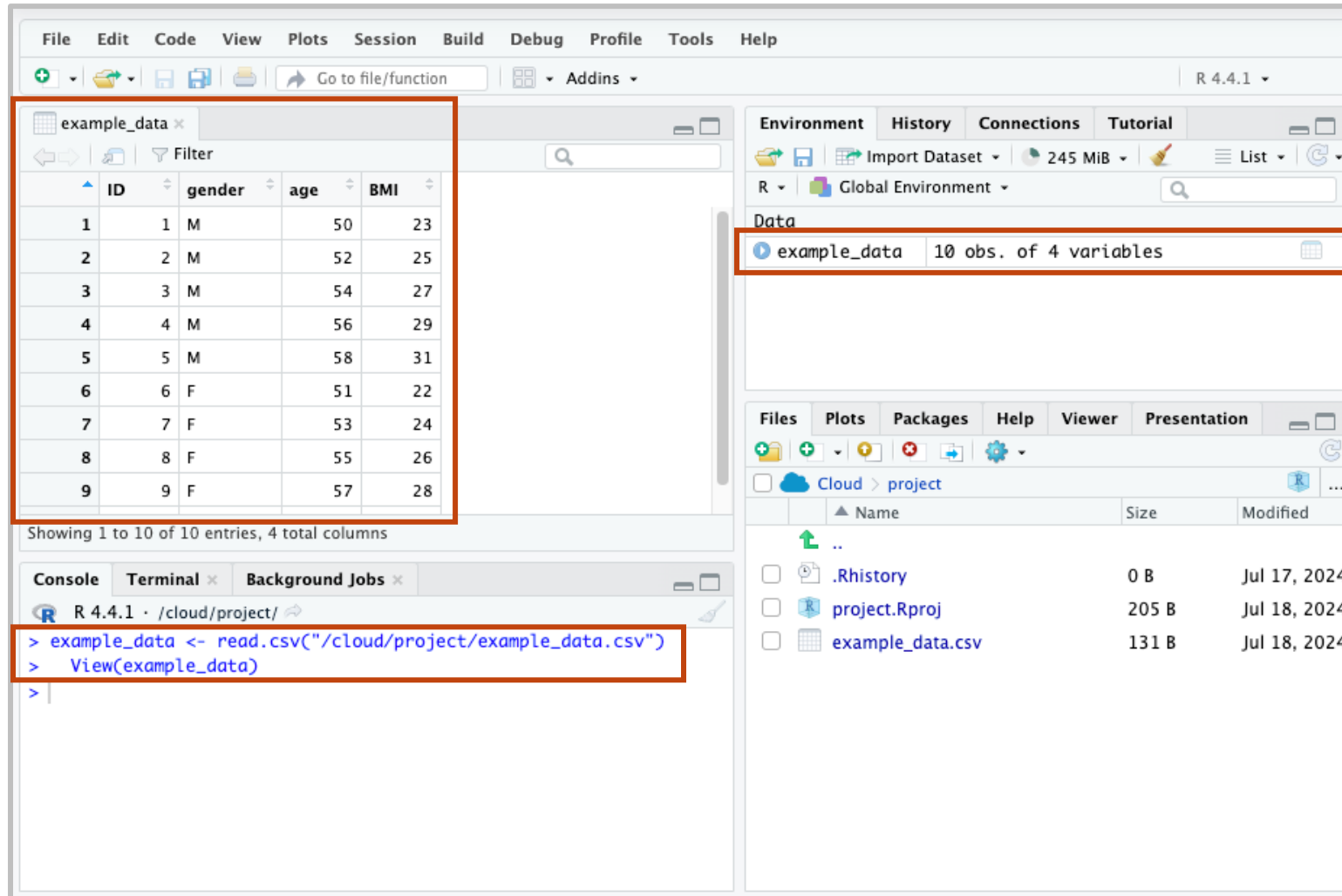
10, F, 59, 30

3 Data Frame

ID	gender	age	BMI
1	M	50	23
2	M	52	25
3	M	54	27
4	M	56	29
5	M	58	31
6	F	51	22
7	F	53	24
8	F	55	26
9	F	57	28
10	F	59	30

4-2 Import Cancel

Rを使いたい人: 上手く読み込まれていると...



The screenshot shows the RStudio interface with the following components:

- Environment pane:** Shows the variable `example_data` with 10 observations and 4 variables.
- Data viewer:** Displays a table with 10 rows and 4 columns: ID, gender, age, and BMI.
- Console:** Shows the commands used to load the data: `example_data <- read.csv("/cloud/project/example_data.csv")` and `View(example_data)`.
- Files pane:** Shows the project files, including `example_data.csv`.

ID	gender	age	BMI
1	M	50	23
2	M	52	25
3	M	54	27
4	M	56	29
5	M	58	31
6	F	51	22
7	F	53	24
8	F	55	26
9	F	57	28

```
> example_data <- read.csv("/cloud/project/example_data.csv")
> View(example_data)
```

Rを使いたい人:

データファイルを読み込む(コマンド操作編)

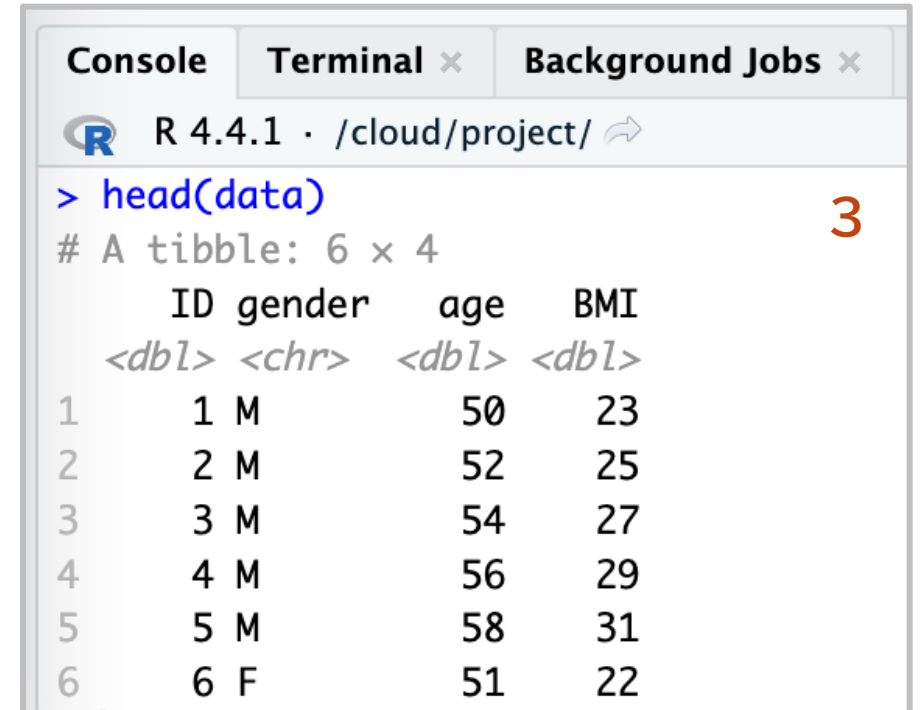
1. tidyverseパッケージをインストールしておく
(標準パッケージにもデータ読み込み関数は用意されていますが、tidyverseに慣れておきましょう)
2. read_csv()関数でデータを読み込む

```
data <- read_csv("example_data.csv")
```

- ✓ dataはR内で使用したい名前
- ✓ example_data.csvは読み込みたいデータファイルの名前
- ✓ コピペをして上手くいかないときは、
二重引用符が “ ” になっているからかもしれません。
半角英数で手入力すれば " が出てくるとおもいます。

3. head()関数でデータの一部を確認する


```
head(data)
```



The screenshot shows an R console window with the following content:

```
Console Terminal x Background Jobs x
R 4.4.1 · /cloud/project/
> head(data)
# A tibble: 6 × 4
  ID gender age BMI
  <dbl> <chr> <dbl> <dbl>
1     1 M     50   23
2     2 M     52   25
3     3 M     54   27
4     4 M     56   29
5     5 M     58   31
6     6 F     51   22
```

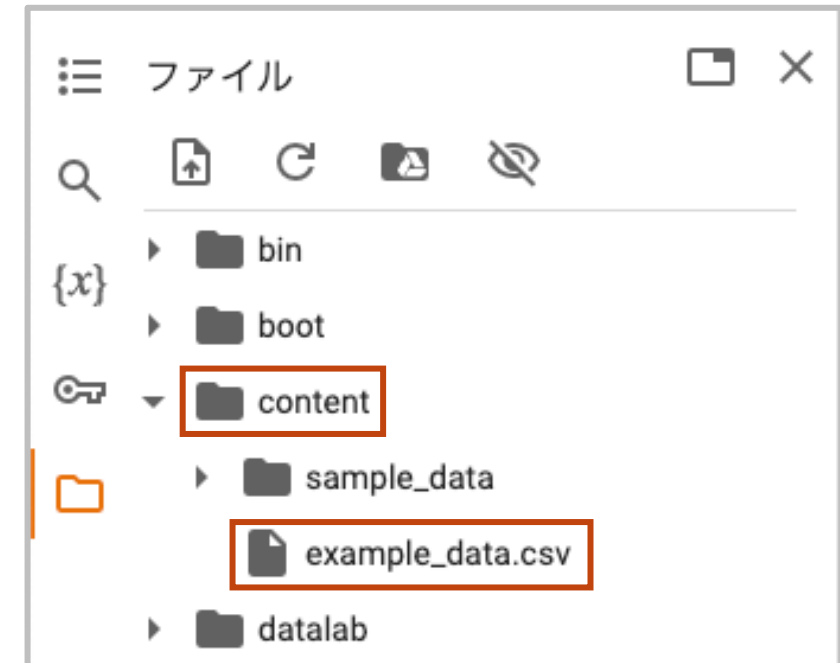
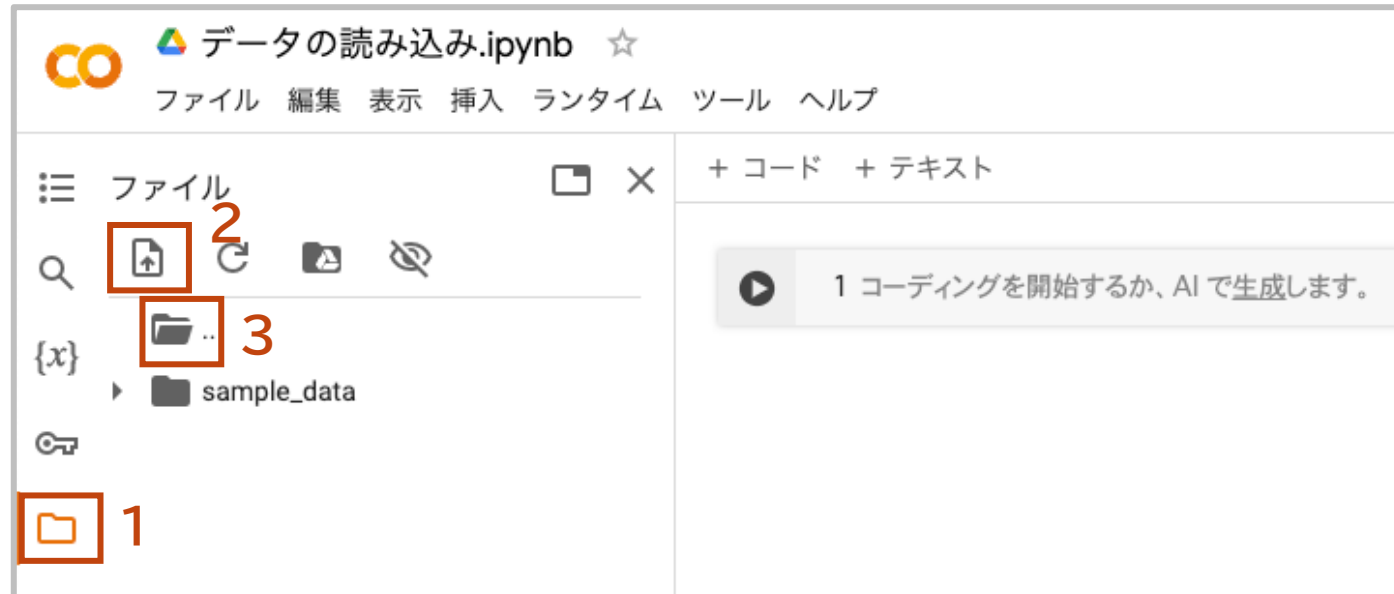
A red number '3' is visible on the right side of the console output.

Pythonを使いたい人 



Pythonを使いたい人: データファイルをアップロードする(自分のPCから)

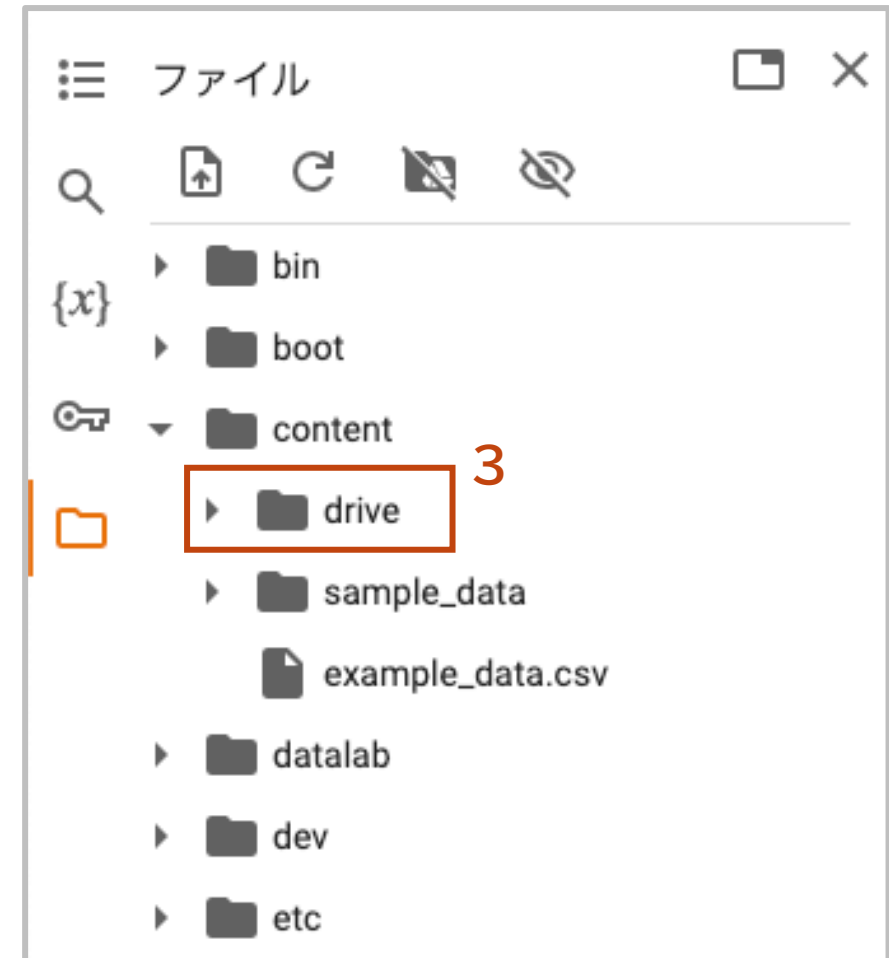
1. 左のサイドバーでファイルボタンを押す
2. アップロードボタンを押して目的のファイルを選択する
(元々sample_dataというフォルダがある。自分がアップロードしたファイルではないことに注意)
3. デフォルトでは/contentというディレクトリ(=ファイルの住所)にアップロードされるので、
「1つ上の階層に行く」ボタンを押して、contentフォルダを確認してみる





Pythonを使いたい人: データファイルをアップロードする(Googleドライブから)

1. 左のサイドバーでファイルメニューを開き、Googleドライブボタンを押す
2. Googleドライブへのアクセスを許可する
3. /contentの下に/driveが追加されたことを確認する





Pythonを使いたい人: データファイルを読み込む

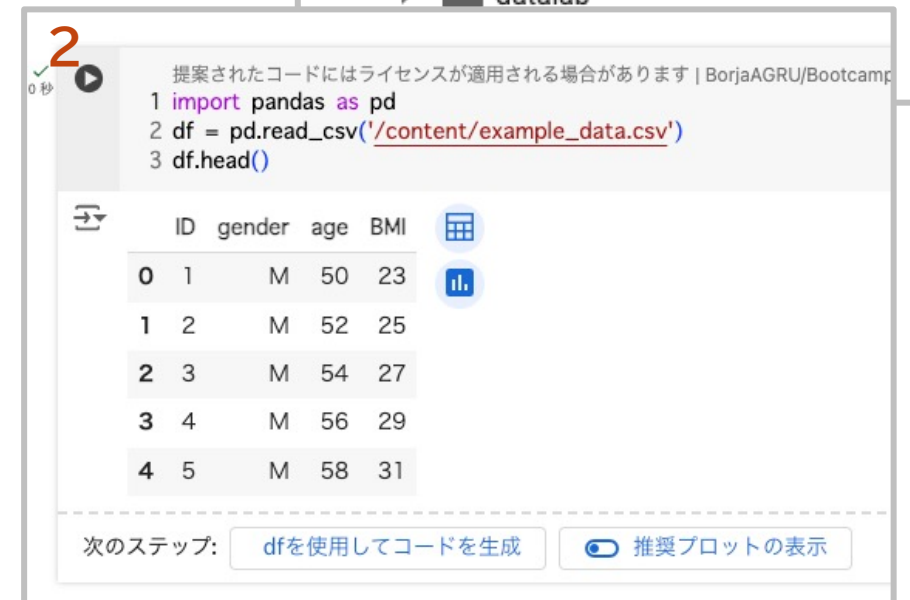
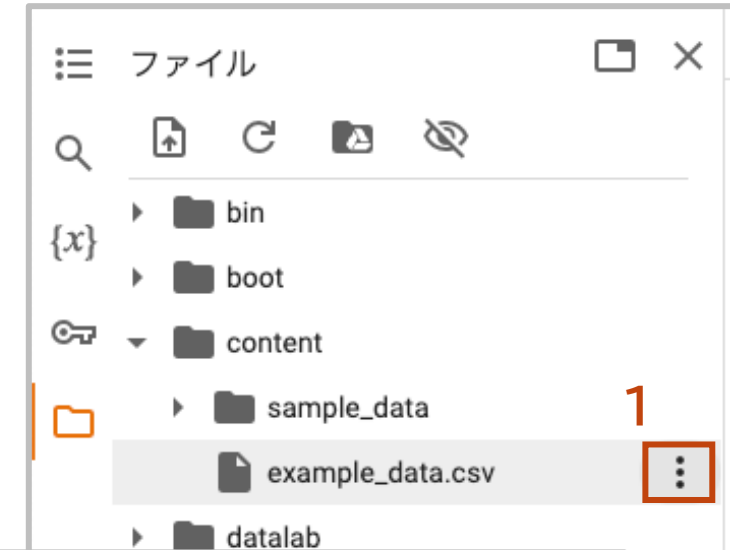
1. 取り込みたいデータファイルの右にポインタを合わせて
[3点マーク] を押す
2. [パスをコピー] を押す
3. `pd.read_csv()`の引数にパスを貼り付けて、コードを実行する。

```
import pandas as pd  
df = pd.read_csv('/content/example_data.csv')
```

- ✓ `df`はPython内で使用したい名前
- ✓ `/content/サンプルデータ.csv`はコピーしたパス

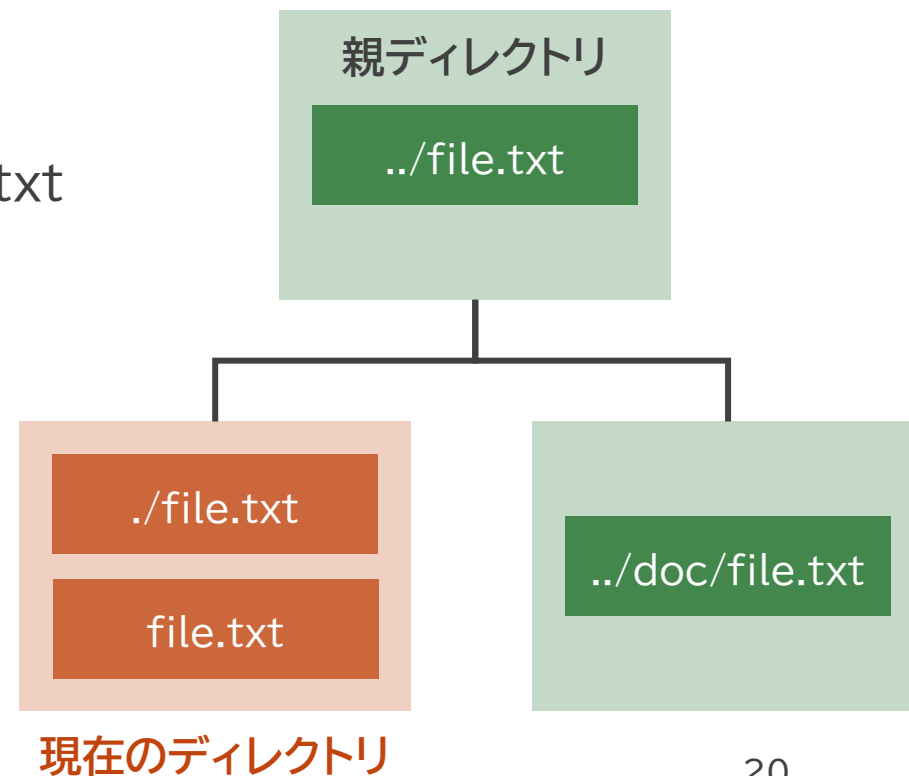
4. `.head()`でデータの一部を確認する

```
df.head()
```



パスとは

- ファイルの存在する場所(=ディレクトリ)を表した「住所」
- **絶対パス**: 最上位(=ルート)からのパスを完全に記載したもの
 - ✓ 例えるなら「兵庫県神戸市中央区〇〇町〇-〇-〇」
 - ✓ Macなら /home/user/documents/file.txt など
 - ✓ WindowsならC:¥Users¥User¥Documents¥file.txt
 - * ドライブレターから始める
 - * ¥は英字キーボードではバックslash\になる
- **相対パス**: 現在位置を基準にしたパス
 - ✓ 例えるなら「私の家の2軒隣の向かいの家」
 - ✓ 現在のディレクトリはピリオド(.)で表す(省略可)
 - ✓ 親ディレクトリはピリオド2つ(..)で表す



課題4：日本語ファイル読み込み

- 課題3で作成した漢字を含むcsvファイル(練習用データ.csv)を読み込んでみましょう
- 文字化けする場合は、エンコーディングを指定してみてください
 - ✓ R(tidyverse): `read_csv("ファイル名.csv", locale = locale(encoding = "エンコード名"))`
 - ✓ Python(pandas): `pd.read_csv('ファイル名.csv', encoding = 'エンコード名')`
 - ✓ エンコード名: UTF-8, CP932(←Windowで作成した場合はこれを試す)

今回のまとめ

- ✓ まずは「整然とした」データシートを作ることが心がけましょう
- ✓ データシートは半角英数のみ使う方がトラブルが少なくすみます
- ✓ 読み込んだら中身を確認するクセをつけましょう