

自分で触ってよくわかる

データの要約の話：

データの概要をつかむ

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

- ✓ 練習に使えるサンプルデータの読み込み方法を理解する
- ✓ 一連の操作を1つのコードにまとめて書く方法を知る

復習： 機械の視点でみた変数の型

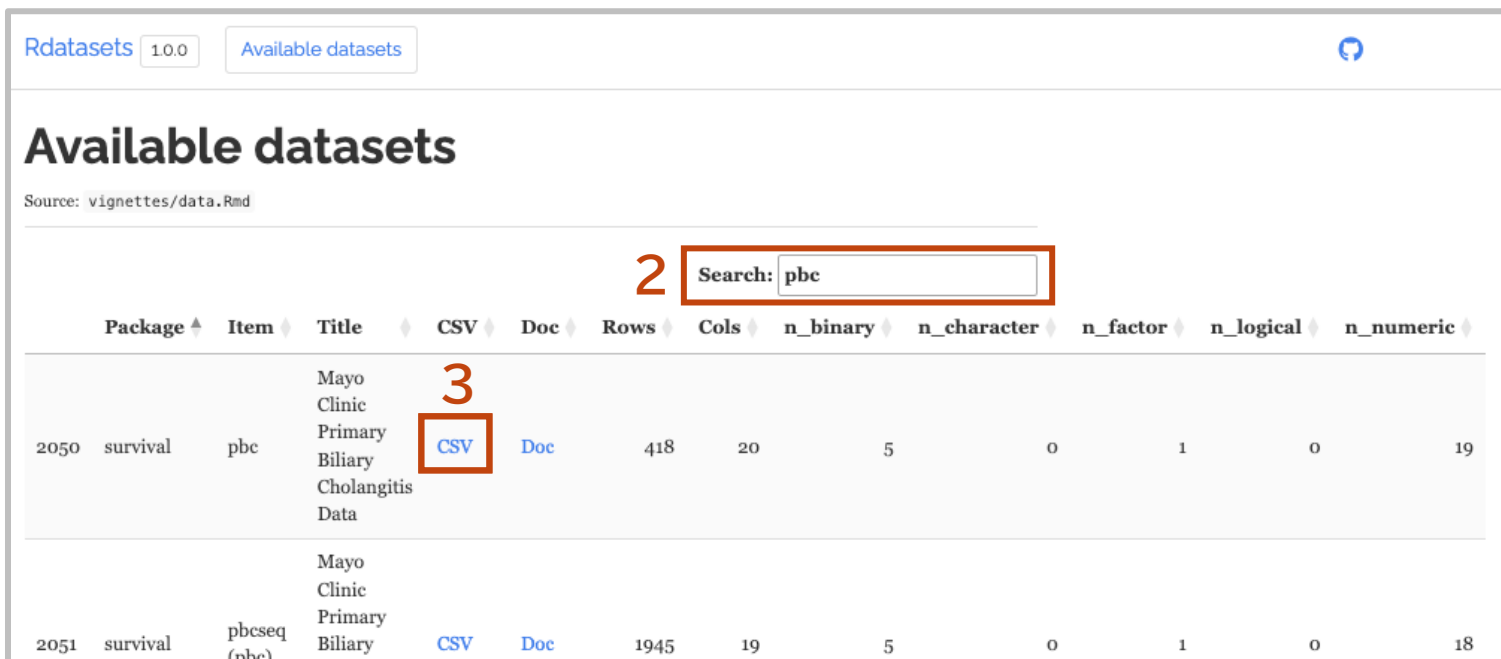
変数の型	説明	例
文字列型	文字の並びを値にとる 数字を文字列として認識させることもある	M, F, “1”
整数型	整数のみ	-1, 0, 1
浮動小数点数型 (実数型)	小数もOK	1.0, 3.14
ブール型	真か偽かどちらかの値をとる 論理式の真偽を示すために使う	True, False
日付・時間型	日付や時間を表す 期間の計算が可能	2024-02-25 02:50:15

復習： 変数の分布を数値で示す方法

変数の種類	示し方	指標		対応する グラフ
質的変数	水準ごとに度数と割合を示す	度数、割合		棒グラフ 円グラフ
量的変数	いくつかの区分に分けて 度数と割合を示す	度数、割合		ヒストグラム
		中心位置	平均値 中央値 最頻値	箱ヒゲ図
	要約値で示す	散らばり具合	分散・標準偏差 四分位範囲 範囲	

用意されている練習用データセットを使う

- Rの豊富なサンプルデータ集(2000種類以上)
 - ✓ <https://vincentarelbundock.github.io/Rdatasets/articles/data.html>
 - ✓ Pythonにもサンプルデータはあるが、Rの方が豊富
- 今回はsurvivalパッケージのpbcデータを使う
 1. 上記のリンクにアクセス
 2. データセット名で検索
 3. csvファイルをダウンロード



The screenshot shows the Rdatasets website interface. At the top, there's a header with 'Rdatasets 1.0.0' and a link to 'Available datasets'. Below this is a search bar with the text 'Search: pbc' and a red box around it, labeled with a red '2'. The main content area is titled 'Available datasets' and lists datasets. The first dataset listed is 'pbc' from the 'survival' package, with a red box around the 'CSV' link in the 'CSV' column, labeled with a red '3'. The table has columns for Package, Item, Title, CSV, Doc, Rows, Cols, n_binary, n_character, n_factor, n_logical, and n_numeric.

Package	Item	Title	CSV	Doc	Rows	Cols	n_binary	n_character	n_factor	n_logical	n_numeric
2050	survival	pbc Mayo Clinic Primary Biliary Cholangitis Data	CSV	Doc	418	20	5	0	1	0	19
2051	survival	pbcseq (pbc) Mayo Clinic Primary Biliary	CSV	Doc	1945	19	5	0	1	0	18

Rを使いたい人 

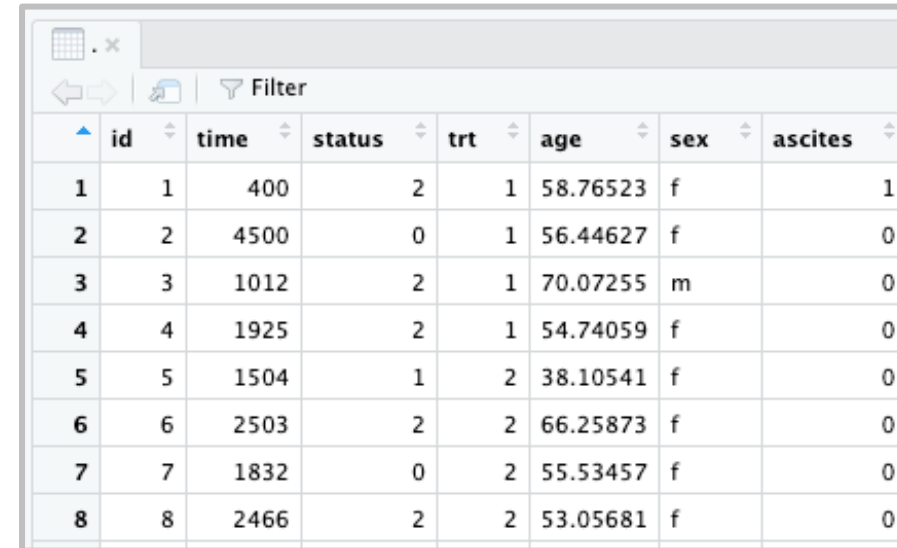
Rを使いたい人: サンプルデータを読み込んで確認する

1. survialパッケージを読み込む
(tidyverseパッケージも読み込んでおく)
2. data()関数でデータセットを読み込む

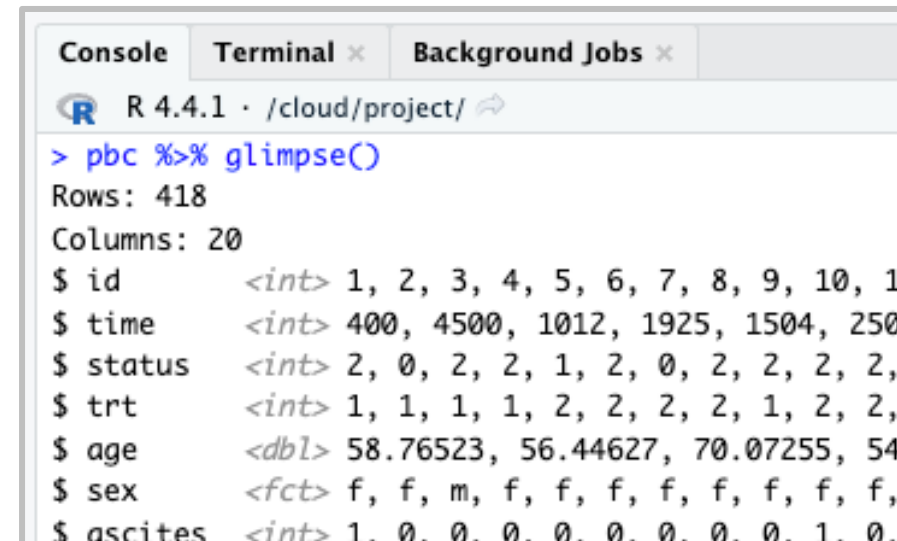
```
data(pbc)
```

3. 下のコードを実行して、データの全体像をつかんでおく

```
pbc %>% View()      #データセットを別タブで表示  
pbc %>% glimpse()   #変数一覧を表示
```



	id	time	status	trt	age	sex	ascites
1	1	400	2	1	58.76523	f	1
2	2	4500	0	1	56.44627	f	0
3	3	1012	2	1	70.07255	m	0
4	4	1925	2	1	54.74059	f	0
5	5	1504	1	2	38.10541	f	0
6	6	2503	2	2	66.25873	f	0
7	7	1832	0	2	55.53457	f	0
8	8	2466	2	2	53.05681	f	0

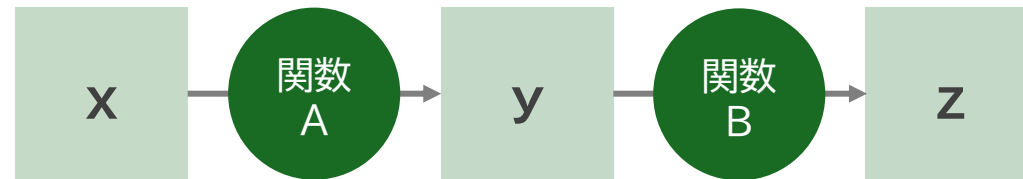


```
R 4.4.1 · /cloud/project/  
> pbc %>% glimpse()  
Rows: 418  
Columns: 20  
$ id      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1  
$ time    <int> 400, 4500, 1012, 1925, 1504, 250  
$ status  <int> 2, 0, 2, 2, 1, 2, 0, 2, 2, 2, 2  
$ trt     <int> 1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2  
$ age     <dbl> 58.76523, 56.44627, 70.07255, 54  
$ sex     <fct> f, f, m, f, f, f, f, f, f, f, f  
$ ascites <int> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0
```

Rを使いたい人:

%>% (パイプ記号) で関数に引数を渡す

- tidyverseパッケージを読み込むとパイプ記号が使える
(R4.1.0以降だと標準機能として |> というパイプ記号が使えますが、見慣れている %>% を使っていきます)
- 関数の戻り値を、次の関数に第1引数として渡すときに便利
- 例: xを関数Aに渡して得られる結果を、関数Bに渡した結果が欲しい



- ✓ 通常書き方だと、必要のない中間産物(y)にも名前をつけないといけない

```
y <- A(x)
z <- B(y)
```


- ✓ パイプ記号を使うと中間産物が発生しないので読みやすい

```
z <- A(x) %>% B()
```


Rを使いたい人:

データの概要を確認するための関数

説明	関数	パッケージ	備考
全体をそのまま表示	<code>View(data)</code>	標準	別タブで表示
全体を要約して表示	<code>skim(data)</code>	skimr	欠測割合や要約値を一括表示
	<code>summary(data)</code>	標準	要約値を一括表示
一部を表示	<code>head(data)</code>	標準	冒頭の一部を表示
	<code>slice_head(data)</code>	tidyverse	冒頭の一部を表示
	<code>slice_tail(data)</code>	tidyverse	末尾の一部を表示
	<code>slice_sample(data)</code>	tidyverse	ランダムに抽出して表示
	<code>slice_max(data, 変数名)</code>	tidyverse	指定した変数について降順に一部表示
	<code>slice_min(data, 変数名)</code>	tidyverse	指定した変数について昇順に一部表示
変数一覧を表示	<code>str(data)</code>	標準	変数名と型を一覧表示
	<code>glimpse(data)</code>	tidyverse	変数名と型を一覧表示

Pythonを使いたい人 



Pythonを使いたい人: サンプルデータを読み込んで確認する

1. 使いたいサンプルデータのパッケージ名とデータセット名をメモしておく
2. 以下のように、statsmodelsパッケージのget_rdataset()を使う

```
import statsmodels.api as sm
dataset = sm.datasets.get_rdataset("データセット名", "パッケージ名")
df = dataset.data
```

✓ datasetには、データ本体(.data)のほか、データセットのタイトル(.title)やデータセットに関する説明(._doc_)も含まれているので、.dataという属性のみ取り出してdfと名前をつけた

3. 下のコードを実行して、データの全体像をつかんでおく

```
print(df)
```

	id	time	status	trt	age	sex	ascites	hepato	spiders	edema
0	1	400	2	1.0	58.765229	f	1.0	1.0	1.0	1.0
1	2	4500	0	1.0	56.446270	f	0.0	1.0	1.0	0.0
2	3	1012	2	1.0	70.072553	m	0.0	0.0	0.0	0.5
3	4	1925	2	1.0	54.740589	f	0.0	1.0	1.0	0.5
4	5	1504	1	2.0	38.105407	f	0.0	1.0	1.0	0.0
...
413	414	681	2	NaN	67.000684	f	NaN	NaN	NaN	0.0
414	415	1103	0	NaN	39.000684	f	NaN	NaN	NaN	0.0
415	416	1055	0	NaN	56.999316	f	NaN	NaN	NaN	0.0
416	417	601	0	NaN	58.001360	f	NaN	NaN	NaN	0.0



Pythonを使いたい人: データの概要を確認するためのメソッド

説明	関数	備考
全体を要約して表示	<code>df.describe()</code>	要約値を一括表示
欠測数を表示	<code>df.isnull().sum()</code>	(次スライド:メソッドチェーン参照)
カテゴリー毎の度数を表示	<code>df['変数名'].value_counts()</code>	
一部を表示	<code>df.head()</code>	冒頭の一部を表示
	<code>df.tail()</code>	末尾の一部を表示
変数一覧を表示	<code>df.info()</code>	変数名と型を一覧表示
	<code>df.dtypes</code>	変数名と型を一覧表示

Pythonを使いたい人:

メソッドをつなげて一気に処理する(メソッドチェーン)

- オブジェクト名の後に「.メソッド名」を足すことで、オブジェクトに対して操作ができる
- メソッドを適用した結果に対して、次のメソッドを適用したいときは、そのままメソッドを数珠状に続けて書くことができる
- 例: dfにメソッドAを適用し、その結果にメソッドBを適用したい



- ✓ 通常書き方だと、必要のない中間産物(df2)にも名前をつけないといけない

```
df1 = df.a  
df2 = df1.b
```

- ✓ メソッドチェーンを使うと中間産物が発生しないので読みやすい

```
df2 = df.a.b
```

課題5：データ確認

- Rのsurvivalパッケージにあるpbcデータについて、治療別にデータを要約してみましょう
 - ✓ 治療方法を表す変数: trt
 - ✓ R(tidyverse): group_by(変数名)を挟んでからskim()を実行
 - ✓ Python(pandas): .groupby('変数名')を挟んでから.describe()を実行

今回のまとめ

- ✓ Rではパイプ記号、Pythonではメソッドチェーンを使って、一連の操作を1つのコードにまとめて書くことができます
- ✓ 複数行にまたがる場合は以下の点に注意してください

R(tidyverse)

```
result <- df %>%  
  fun_a() %>%  
  fub_b()
```

行末にパイプ記号がくるようにする

Python

```
result = (df  
          .method_a  
          .method_b  
)
```

全体を()でくくる