超入門！

すべての
医療従事者のための

RStudioで
はじめる
医療統計

笹渕 裕介
大野 幸子
橋本 洋平
石丸 美穂 著

サンプルデータでらくらくマスター

Kinpodo

第6章 データの概要の確認

ポイント

- glimpse() でデータ全体を俯瞰する
- summary(), table() で各変数の分布を確認する
- ‘tableone’ パッケージで表を作成する

本章で必要なパッケージ ● tidyverse ● tableone

：「さぁデータクリーニングも終わったぞ。ついに解析だ！　さっそく回帰分析だ！」

：「早い、早いよ、A くん。まず集めたデータの平均、中央値、分散などの分布や頻度などを確認してみよう」

1 データの俯瞰と要約

データを手に入れると上の A くんのようにすぐ解析をしたくなるものです。

しかし、データ解析を始める前（さらに言うならデータクリーニング前）に、まず自分が扱っているデータについて

- どのような変数がデータに含まれるのか？（年齢、性別など）
- 変数はいくつあるのか？（＝列数）
- サンプル数はいくつか？（＝行数）
- 各変数の型は？
- 各変数の値の平均、分散は？

などを大まかに把握しておく必要があります。すべての変数の情報をコンパクトに出力する `glimpse()` でデータを俯瞰します。

```
glimpse(data)
```

data	変数の一覧を表示するデータフレーム

```
df <- read_csv("R_book_data.csv")
df %>%
   glimpse()
```

```
Observations: 500
Variables: 18
$ id              <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...
$ Year            <dbl> 2010, 2012, 2012, 2011, 2013, 2010, 2014, 2011, ...
$ Admday          <chr> "2010/10/24", "2012/9/24", "2012/12/9", "2011/9/...
$ Discday         <chr> "2010/11/5", "2012/10/3", "2012/12/14", "2011/9/...
$ New_Treatment   <dbl> 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, ...
$ Age             <dbl> 62, 82, 75, 78, 78, 68, 72, 71, 80, 72, 75, 63, ...
$ Sex             <dbl> 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, ...
$ Height          <dbl> 167, 156, 155, 153, 154, 157, 168, 154, 165, 152...
$ Weight          <dbl> 75.8, 57.0, 61.2, 49.5, 52.5, 61.1, 64.3, 47.3, ...
$ DM              <dbl> 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, ...
$ Stroke          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ MI              <dbl> 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ...
$ Severity        <dbl> 2, 3, 2, 2, 8, 3, 1, 7, 11, 2, 6, 1, 2, 1, 1, 2,...
$ Death           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, ...
$ LOS             <dbl> 13, 10, 6, 11, 15, 9, 19, 22, 15, 10, 7, 10, 13,...
$ Treatment_3cat  <dbl> 2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 2, 2, 1, 3, 3, 1, ...
$ pre1            <dbl> -4.3483485, 5.5934687, 6.9424774, 1.6532087, -0....
$ pre2            <dbl> -2.95081859, 0.44854366, 1.62122564, 0.28893993,..
```

1）Observations は観測数（行の数）を、Variables は変数の数（列の数）を表します。df は 500 行×18 列のデータということがわかります。各変数は、データ型（第 13 章参照）と最初の数個の値が出力されています。

連続変数の要約を行うには `summary()` が便利です。最小値、最大値、平均値、中央地、四分位点がまとめて表示されます。

```
summary(x)
```

`x` 要約する変数

```
summary(df$Age)
```

```
##  Min. 1st Qu.  Median   Mean 3rd Qu.    Max.
##  54.0  70.0    75.0  74.9  79.0  91.0
```

カテゴリ変数の頻度を見たいときには、`table()` が便利です。

```
table(x)
```

`x` 集計する変数

```
table(df$New_Treatment)
```

```
##
##   0   1
## 321 179
```

`New_Treatment` 列には 0 が 321 個、1 が 179 個あることがわかります。

各カテゴリーの割合を見たいときは、さらに `prop.table()` を使います。

```
prop.table(table(df$New_Treatment))
```

```
##
##     0     1
## 0.642 0.358
```

以上より、`New_Treatment` 列には 0 が 321 個（64.2%）、1 が 179 個（35.80%）あることがわかります。

2 tableone パッケージ

`tableone` パッケージの `CreateTableOne()` を用いることで、**臨床研究論文の Table 1 にあたる患者の背景情報を簡単に要約すること**ができます。

サンプルデータの New_Treatment が行われた群（New_Treatment=1）と行われなかった群（New_Treatment=0）の背景情報を比較してみます。

```
CreateTableOne(vars, strata, factorVars, data)
```

の形で使います。

vars	テーブルに含める変数名（列名）
strata	層別化（グループ化）する変数名
factorVars	vars の変数名のうち、カテゴリー変数（離散変数）
data	データフレーム名

New_Treatment の 0 と 1 の 2 群の比較を行う場合、strata="New_Treatment" とします。テーブルに含めたい変数は "Age"，"Sex"，"Height"，"Weight"，"Severity"，"DM" とします。これらをすべて vars に指定します。このうち、Sex は 1 か 2、DM は 1 か 0 のいずれかの値をとるカテゴリー変数なので、factorVars には "Sex"，"DM" の 2 変数を指定します。

```
library(tableone)
tbl_1 <- CreateTableOne(vars = c("Age", "Sex", "Height", "Weight",
"Severity", "DM"), strata = "New_Treatment", factorVars = c("Sex", "DM"), data
= df)

tbl_1
```

```
                     Stratified by New_Treatment
                      0              1           p      test
  n                     321            179
  Age (mean (SD))     72.58 (5.15)   79.06 (5.00)  <0.001 1)
  Sex = 2 (%)          198 (61.7)     106 (59.2)    0.656
  Height (mean (SD))  155.52 (6.40)  155.12 (6.17)   0.498
  Weight (mean (SD))   55.26 (8.12)   54.61 (8.00)   0.392
  Severity (mean (SD))  3.38 (2.77)    6.54 (3.23)  <0.001
  DM = 1 (%)            25 ( 7.8)      48 (26.8)    <0.001
```

論文の表 1 にそのまま使える結果がこれだけで作成できてしまいます。

1） 年齢は、New_Treatment が 0 の群では平均 72.6 歳、標準偏差 5.1 歳であり New_Treatment が 1 の群では平均 79.1 歳、標準偏差 5.0 歳です。そして p<0.001 なので有意に 1 の群の年齢が高いことがわかります。

tbl_1 をファイルとして出力したいときは、tbl_1 オブジェクトを `print()` したうえで、`wirte.csv()` で csv ファイルへ出力します。

```
write.csv(x, file)
```

x	csv ファイルとして出力したいオブジェクト（データフレームなど）
file	ファイル名

```
tbl_1 %>%
  print() %>%
  write.csv(file = "tableone.csv")
```

```
                    Stratified by New_Treatment
                     0              1              p      test
n                      321            179
Age (mean (SD))      72.58 (5.15)   79.06 (5.00)  <0.001
Sex = 2 (%)            198 (61.7)     106 (59.2)   0.656
Height (mean (SD))  155.52 (6.40)  155.12 (6.17)   0.498
Weight (mean (SD))   55.26 (8.12)   54.61 (8.00)   0.392
Severity (mean (SD))  3.38 (2.77)    6.54 (3.23)  <0.001
DM = 1 (%)             25 ( 7.8)      48 (26.8)  <0.001
```


すると、tableone.csv というファイルがプロジェクトフォルダの中に作成されます（プロジェクトフォルダは第 5 章を参照）。これで論文投稿や学会発表にそのままの形で用いることができ、大変便利です。

	A	B	C	D	E
1		0	1	p	test
2	n	321	179		
3	Age (mean (SI	72.58 (5.15)	79.06 (5.00)	<0.001	
4	Sex = 2 (%)	198 (61.7)	106 (59.2)	0.656	
5	Height (mean	155.52 (6.40)	155.12 (6.17)	0.498	
6	Weight (mean	55.26 (8.12)	54.61 (8.00)	0.392	
7	Severity (mea	3.38 (2.77)	6.54 (3.23)	<0.001	
8	DM = 1 (%)	25 (7.8)	48 (26.8)	<0.001	

図6-1 出力された tableone.csv

図 6-1 の通り、`CreateTableOne()` では、デフォルトで連続変数には *t* 検定、離散変数にはカイ 2 乗検定が使用されます（ウィルコクソン検定や、フィッシャー正確検定については第 8、9 章参照）。

第8章 2群間の比較

ポイント

- t.test()、wilcox.test() で連続変数の検定を行う
- fisher.test()、chisq.test() でカテゴリー変数の検定を行う

本章で必要なパッケージ ● tidyverse ● tableone

：「平均値やグラフを見ると、新治療を行った群では入院期間が長くて、死亡は少なさそうだな。統計的に差があると言うためには、検定する必要があるんだよな」

2群のデータを比較するには検定を行います。2群の値に本当は差がない場合、「今のデータが示している差が偶然起こる確率はどのくらいか？」を表すのが p 値です。医学研究では慣習的に p 値 <0.05 を有意とします。さまざまな検定方法がありますが、本書では医学研究において最も一般的に用いられる方法を紹介します。

1 統計手法の選択

表 8-1 の統計手法の選択に従い、2群比較を行う際の検定について説明します。

表8-1　統計手法の選択

	カテゴリー変数	連続変数			生存時間
可視化	棒グラフ	ヒストグラム・箱ひげ図			カプランマイヤー曲線
分布の記述	度数分布 分割表	平均・分散・標準偏差 中央値・四分位範囲			
単純な群比較	フィッシャー正確検定 カイ二乗検定	2群	正規分布	*t* 検定	ログランク検定
			非正規分布	ウィルコクソン順位和検定	
		3群以上	正規分布	一元配置分散分析	
			非正規分布	クラスカルウォリス検定	
多変量回帰	ロジスティック回帰	重回帰			コックス回帰

2　連続変数の比較

最初にデータの分布を確認します。

正規分布に近い分布であれば *t* 検定を行います。一方、サンプル数が小さい場合や、正規分布から大きく外れている場合は後述するウィルコクソン順位和検定を行います（第6章「2 .tableone パッケージ」、第7章「3 . ヒストグラム」参照）。

① *t* 検定

t 検定を行うには `t.test()` を利用します。また、`tableone` パッケージの `CreateTableOne()` を利用して行うことも可能です。`CreateTableOne()` を利用する場合はいったん `CreateTableOne()` の結果をオブジェクトに格納してから、`print()` で結果を表示します。

t.test() を使った *t* 検定

```
t.test(x ~ group, data = data)
```

x	検定を行う連続変数
group	群分け変数
data	データフレーム

CreateTableOne() を利用した *t* 検定

```
tbl_ttest <- CreateTableOne(vars, strata, data)
print(tbl_ttest)
```

vars	検定を行う連続変数
strata	群分け変数
data	データフレーム

実際にサンプルデータを使って検定を行っていきましょう。

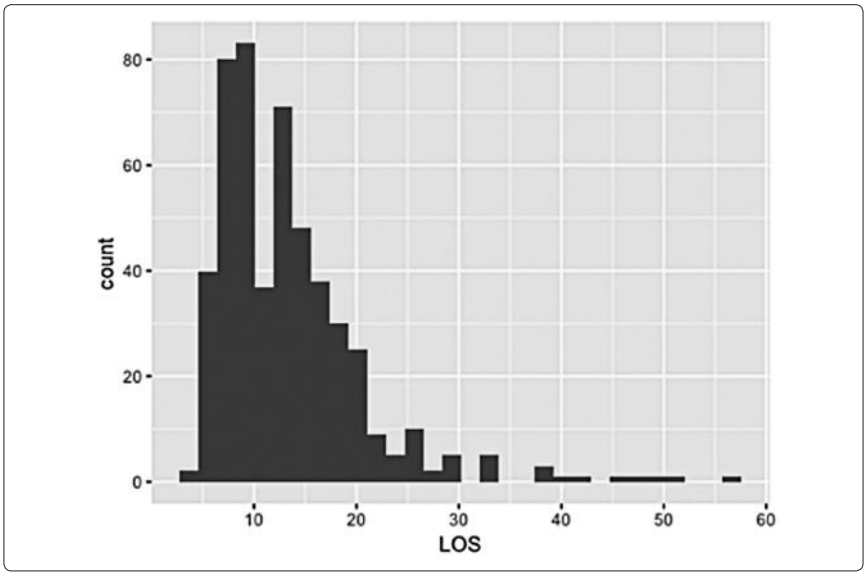
今回は New_Treatment の有無によって入院期間の平均が異なるかどうかを検定します。

```
library(tidyverse)
library(tableone)

# sample データの読み込み
df <- read_csv("R_book_data.csv")

# 分布の確認
g_dist_2g <- ggplot(data = df, aes(x = LOS)) +
  geom_histogram()

g_dist_2g
```



ヒストグラムを見ると右に裾を引いた形をしています。サンプル数が小さい場合や、正規分布から大きく外れている場合は後述のウィルコクソン順位和検定を行います。LOS は正規分布とは言えませんが、ここでは便宜上、*t* 検定もウィルコクソン検定も LOS を使用します。

```
# t.test() を使った t 検定
t.test(LOS ~ New_Treatment, data = df)
```

```
 Welch Two Sample t-test 1)

data:  LOS by New_Treatment
t = -4.9642, df = 303.78, p-value = 1.152e-06 2)
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: 3)
 -4.836858 -2.090760
sample estimates:
mean in group 0 mean in group 1 4)
       12.13396        15.59777
```

1）*t* 検定を行ったことを示しています。

2）結果は p-value < 1.152e-06 でした（R ではしばしば e-06 のような表記が登場します。これは 10 の -6 乗を表します。**1.152e-06 は 1.152×10 の -6 乗です。e+6 と表記された場合は 10 の 6 乗を表します。e とありますが、自然対数の底ではありません**）。2 群間に統計学的な有意差を認めました。

3）平均値の差の 95% 信頼区間は -4.8〜-2.1 であることが示されています。

4）グループごとの集計を見ると、平均入院期間は対照群 12.1 日、New_Treatment 群は 15.6 日であり、統計学的に有意に New_Treatment 群の入院期間が長いことがわかります。

```
# CreateTableOne() を使った t 検定

tbl_ttest <- CreateTableOne(vars = "LOS",
                            strata = "New_Treatment",
                            data = df)
print(tbl_ttest)
```

```
                  Stratified by New_Treatment
                   0            1            p      test
  n                  321          179
  LOS (mean (sd)) 12.13 (6.37) 15.60 (8.03) <0.001
```

5)

5） `CreateTableOne()` を利用した場合、表形式で各群の平均入院日数と標準偏差、*p* 値が出力されます。*p* 値が 0.001 よりも小さい場合、*p* < 0.001 と表示されます。

②ウィルコクソン順位和検定

ウィルコクソン順位和検定は 2 群のすべてのデータを合わせて昇順（または降順）に並べて順位をつけます。この順位の和が 2 群間で異なるかを検定します（**表 8-1**）。サンプル数が小さい場合や、正規分布から大きく外れている場合はウィルコクソン順位和検定を行います。

```
wilcox.test(x ~ group, data)
```

x	検定を行う連続変数
group	群分け変数
data	データフレーム

```
# wilcox.test() を使ったウィルコクソン順位和検定
wilcox.test(LOS ~ New_Treatment, data=df)
```

```
        Wilcoxon rank sum test 1) with continuity correction
data:  LOS by New_Treatment
W = 20304, p-value = 5.022e-08 2)
alternative hypothesis: true location shift is not equal to 0
```

1）ウィルコクソン順位和検定を行っていることを示しています。
2）p-value = 5.022e-08(5.022×10^{-8}) ですので統計学的有意差を認めました。

グループ集計の結果と合わせて、統計学的に有意に New_Treatment 群の入院期間が長いことがわかります。

CreateTableOne() を利用したウィルコクソン順位和検定

```
tbl_wilcox <- CreateTableOne(vars, strata, data)
```

vars	検定を行う連続変数
strata	群分け変数
data	データフレーム

```
print(tbl_wilcox, nonnormal)
```

tbl_wilcox	CreateTableOne() の結果
nonnormal	ウィルコクソン順位和検定を行う変数名

```
# CreateTableOne() を使ったウィルコクソン順位和検定
tbl_wilcox <- CreateTableOne(vars = "LOS",
                             strata = "New_Treatment",
                             data = df)
print(tbl_wilcox, nonnormal = "LOS")
```

```
                  Stratified by New_Treatment
                   0                 1                 p      test
n                    321               179
LOS (median [IQR]) 11.00 [8.00, 15.00] 13.00 [10.00, 19.00] <0.001 nonnorm
```
1)

1）群ごとの集計結果から、入院期間の中央値は対照群 11 日、New_Treatment 群は 13 日であり ([ ] は 4 分位範囲を表示しています)、p 値が 0.001 よりも小さい場合、$p<0.001$ と表示されます。また、test は nonnorm とあり、ウィルコクソン順位和検定を行ったことを示しています。

3 カテゴリー変数の比較

2群の比率が異なるかを検定するには、通常はフィッシャー正確検定を選択すれば問題ありません。データ量が大きく、フィッシャー正確検定が難しい場合にはカイ二乗検定を行いましょう。

①フィッシャー正確検定

fisher.test() を使用したフィッシャー正確検定

```
fisher.test(x, y)
```

x	変数1（検定を行うカテゴリー変数）
y	変数2（群分け変数）

```
# fisher.test() を使用したフィッシャー正確検定
fisher.test(df$Death, df$New_Treatment)
```

```
        Fisher's Exact Test for Count Data                    1)

data:  df$Death and df$New_Treatment                          2)
p-value = 0.0388                                              3)
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:                               4)
 0.2736981 0.9887565
sample estimates:
odds ratio                                                    4)
 0.5326503
```

1） フィッシャー正確検定を行ったことを示します。
2） 検定を行った2変数を示しています。
3） p-value = 0.0388 であり、統計学的に有意差を認めました。
4） オッズ比の95%信頼区間と点推定値を示しています。1をまたいでいないので、有意に New_Treatment 群の死亡オッズが低いことがわかります。

CreateTableOne() を利用したフィッシャー正確検定

```
tbl_fisher <- CreateTableOne(vars, strata, factorVars, data)
```

vars	検定を行う変数
strata	群分け変数
factorVars	アウトカム（カテゴリー変数）
data	データフレーム

```
print(tbl_fisher, exact)
```

tbl_fisher	CreateTableOne() 関数の結果
exact	変数名を指定することでフィッシャー正確検定を行う。省略するとカイ二乗検定を行う

```
# CreateTableOne() を利用したフィッシャー正確検定
tbl_fisher < -CreateTableOne(vars = "Death", strata = "New_Treatment",
factorVars = "Death", data =df)
print(tbl_fisher, exact = "Death")
```

```
            Stratified by New_Treatment
                  0           1           p      test
n               321           179
Death = 1 (%)  50 (15.6)   16 (8.9)   0.039 exact
```

1)

1）群ごとの死亡数と割合を示しています。対照群は 321 人中 50 人（15.6%）、New_Treatment 群では 179 人中 16 人（8.9%）死亡していることがわかります。フィッシャー正確検定を行ったことを示す exact が表示されます。*p* 値は 0.039 であり、統計学的に有意に New_Treatment 群の死亡割合が低いという結果でした。