

# 生存時間解析

第2回 Rによるハンズオン

2019/10/25版

統計数理研究所

長島 健悟

<https://nshi.jp/>

# 目的：Rを使ったハンズオン

- Rの基本的な使用方法
  - R言語の学習のポイントとデータ構造
  - 基本関数によるデータハンドリング
- RStudioとR Markdown
- Rによる生存時間解析

# What is R?

- R is a language and environment for statistical computing and graphics. ... R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

<https://www.r-project.org/about.html> より

# R (2006年)



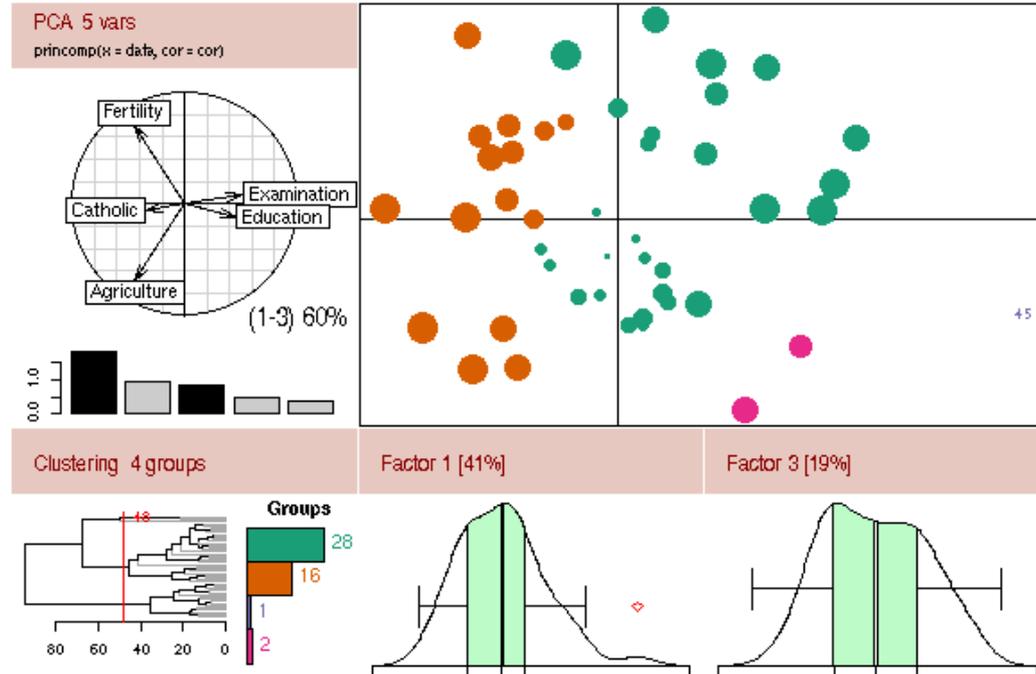
About R  
[What is R?](#)  
[Contributors](#)  
[Screenshots](#)  
[What's new?](#)

Download  
[CRAN](#)

R Project  
[Foundation](#)  
[Members & Donors](#)  
[Mailing Lists](#)  
[Bug Tracking](#)  
[Developer Page](#)  
[Conferences](#)  
[Search](#)

Documentation  
[Manuals](#)  
FAQs  
[Newsletter](#)  
[Wiki](#)

## The R Project for Statistical Computing



### News:

- [DSC 2007](#), the 5th workshop on Directions in Statistical Computing, February 15-16, 2007, Auckland, New Zealand.
- Sources of [R 2.4.0 alpha and beta releases](#) (daily snapshot).
- [R News 6/3](#) has been published on 2006-08-24.
- [R version 2.3.1](#) has been released on 2006-06-01.

# R (2019年)



[\[Home\]](#)

## Download

[CRAN](#)

## R Project

[About R](#)

[Logo](#)

[Contributors](#)

[What's New?](#)

[Reporting Bugs](#)

[Conferences](#)

[Search](#)

[Get Involved: Mailing Lists](#)

[Developer Pages](#)

[R Blog](#)

## R Foundation

[Foundation](#)

# The R Project for Statistical Computing

## Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

## News

- [R version 3.6.1 \(Action of the Toes\)](#) has been released on 2019-07-05.
- useR! 2020 will take place in St. Louis, Missouri, USA.
- [R version 3.5.3 \(Great Truth\)](#) has been released on 2019-03-11.
- The R Foundation Conference Committee has released a [call for proposals](#) to host useR! 2020 in North America.
- You can now support the R Foundation with a renewable subscription as a [supporting member](#)
- The R Foundation has been awarded the Personality/Organization of the year 2018 award by the professional association of German market and social researchers.

# Conference (useR!)



**useR! 2019** @UseR2019\_Conf · Aug 26, 2019



Some facts about useR! 2019 participants (1/9):



**useR! 2019**  
@UseR2019\_Conf

1178 participants, including 182 guests / sponsors / scholars with a free registration. Facts are available for paid registrations only (2/9)

♡ 6 11:59 PM - Aug 26, 2019



**useR! 2019**  
@UseR2019\_Conf

151 oral presentations, 71 flash presentations and 94 posters (among 467 submitted abstracts) (8/9)

♡ 6 12:00 AM - Aug 27, 2019



 [See useR! 2019's other Tweets](#)



# The R Journal



## Navigation

[Current Issue](#)  
[Accepted articles](#)  
[Archive](#)  
[R News](#)  
[News and Notes](#)  
[Submissions](#)  
[Reviews and Proofreading](#)  
[Editorial Board](#)

## Subscribe

[RSS Feed](#) 

ISSN: 2073-4859

## The R Journal

*The R Journal* is the open access, refereed journal of the [R project](#) for statistical computing. It features short to medium length articles covering topics that should be of interest to users or developers of R. *The R Journal* intends to reach a wide audience and have a thorough review process. Papers are expected to be reasonably short, clearly written, not too technical, and of course focused on R. Authors of refereed articles should take care to:

- put their contribution in context, in particular discuss related R functions or packages;
- explain the motivation for their contribution;
- provide code examples that are reproducible.

Following revision of the content description of *The R Journal*, from January 2017 submitted articles may include:

### Reviews and proposals:

surveying and discussing challenges and opportunities of potential importance for the broader R community, including proposals and proof-of-concept implementations.

### Comparisons and benchmarking:

of implementations in base-R and contributed packages with each other, and where relevant with implementations in other software systems.

### Applications:

demonstrating how new or existing techniques can be applied in an area of current interest using R, providing a fresh view of such analyses in R that is of benefit beyond the specific application.

### Add-on packages:

short introductions to contributed R packages that are already available on CRAN or Bioconductor, and going beyond package vignettes in aiming to provide broader context and to attract a wider readership than package users. Authors need to make a strong case for such introductions, based for example on novelty in implementation and use of R, or the introduction of new data structures representing general architectures that invite re-use.

# Rの特徴

- 関数型プログラミング言語
  - 基本的文法は必修
- 複雑な構造 (関数もデータも)
  - ベクトル処理 (Rはデータ処理言語)
  - リスト (異構造のデータをまとめて保持), データフレーム (表形式データ)
  - データ読み込み + 変数を指定して解析 :  
だけなら簡単...

# 例

- 自作の関数が必要になるケースが...

```
plus <- function(a, b) {  
  a + b  
}  
plus(2, 3)  
## [1] 5
```

よくある

# RStudio



DOWNLOAD SUPPORT COMMUNITY

Products ▾ Resources ▾ Pricing About ▾ Blogs ▾

## RStudio IDE Features

RStudio is the premier integrated development environment for R. It is available in open source and commercial editions on the desktop (Windows, Mac, and Linux) and from a web browser to a Linux server running RStudio Server or RStudio Server Pro.



### An IDE that was built just for R

- Syntax highlighting, code completion, and smart indentation
- Execute R code directly from the source editor
- Quickly jump to function definitions



### Bring your workflow together

- Integrated R help and documentation
- Easily manage multiple working directories using projects
- Workspace browser and data viewer



### Powerful authoring & Debugging

- Interactive debugger to diagnose and fix errors quickly
- Extensive package development tools
- Authoring with Sweave and R Markdown

# RStudio

- Rの統合開発環境
  - エディタ (コード補完), R Markdown
  - オブジェクト表示, コード実行履歴
  - ヘルプ表示, パッケージ管理
  - プロジェクト管理
  - パッケージ開発補助

**Rのコーディングが快適に**

# Rによる生存時間解析

# RStudioの画面構成

The image shows the RStudio interface with three blue callout boxes highlighting key components:

- エディタ (Editor):** Located in the top-left pane, it displays R code for a function named `plus`. The code is:

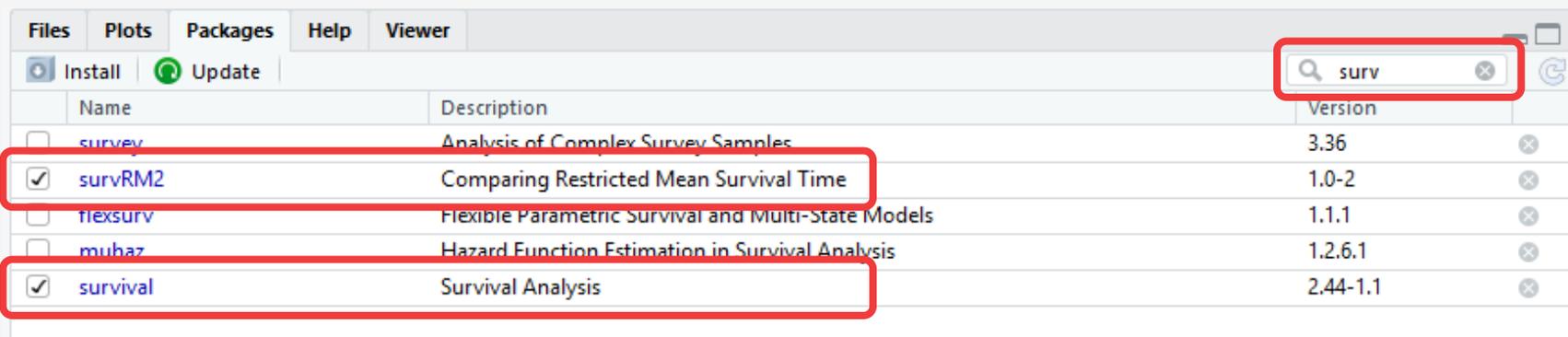
```
1 plus <- function(a, b) {  
2   a + b  
3 }  
4 plus(2, 3)  
5
```
- オブジェクト等 (Objects, etc.):** Located in the top-right pane, titled "Environment". It shows the "Global Environment" with a table structure for variables.
- パッケージ等 (Packages, etc.):** Located in the bottom-right pane, titled "Packages". It shows a list of installed and available packages in the "User Library".
- コンソール (Console):** Located in the bottom-left pane, it shows the R startup message and the prompt `>`.

Name	Type	Length	Size	Value
------	------	--------	------	-------

Name	Description	Version
<input type="checkbox"/> abind	Combine Multidimensional Arrays	1.4-5
<input type="checkbox"/> acepack	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
<input type="checkbox"/>	Multilevel/Hierarchical Models	
<input type="checkbox"/> askpass	Safe Password Entry for R, Git, and SSH	1.1
<input type="checkbox"/> assertthat	Easy Pre and Post Assertions	0.2.1
<input type="checkbox"/> backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.4
<input type="checkbox"/> base64enc	Tools for base64 encoding	0.1-3

# パッケージの準備

- 右下画面でsurvivalとsurvRM2を探してチェック (虫眼鏡は検索)

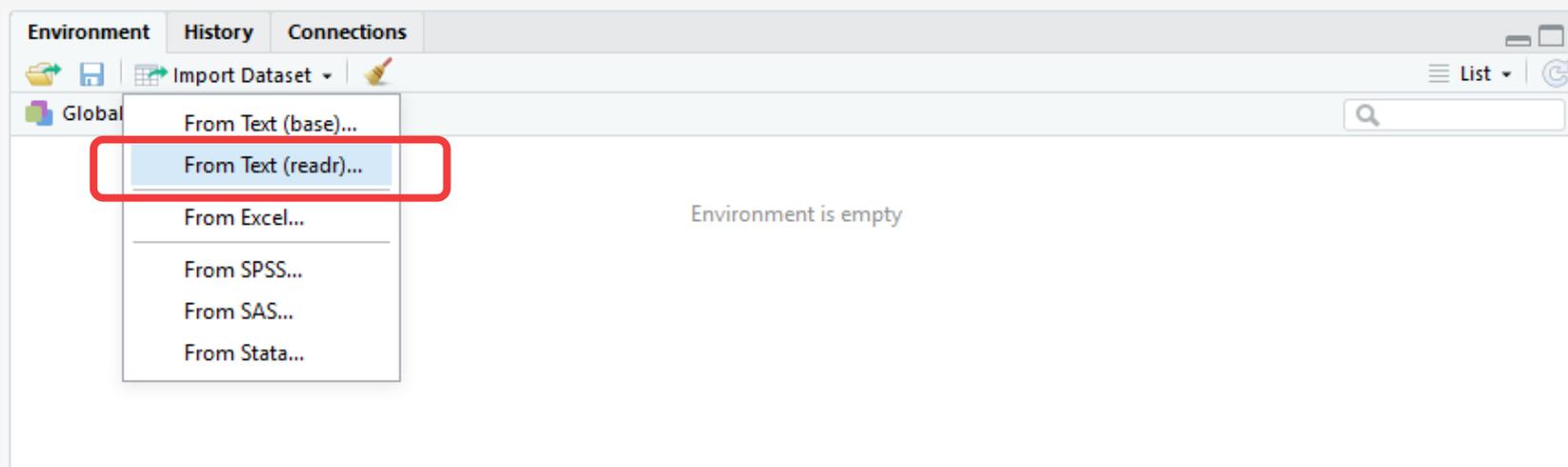


The screenshot shows the RStudio Packages window with a search bar containing 'surv'. The search results are as follows:

Name	Description	Version
<input type="checkbox"/> survey	Analysis of Complex Survey Samples	3.36
<input checked="" type="checkbox"/> survRM2	Comparing Restricted Mean Survival Time	1.0-2
<input type="checkbox"/> flexsurv	Flexible Parametric Survival and Multi-State Models	1.1.1
<input type="checkbox"/> muhaz	Hazard Function Estimation in Survival Analysis	1.2.6.1
<input checked="" type="checkbox"/> survival	Survival Analysis	2.44-1.1

# データの読み込み (1)

- 右上画面から Import Dataset → From Text (readr)...



- Excel, SAS, SPSS, Stataのデータにも対応

# データ読み込み (2)

- File/Url: <https://nshi.jp/dat.csv> → Update → Import

Import Text Data

File/Url:

Data Preview:

x (double)	u (double)	event (double)	t (double)	edf (double)	km (double)	expna (double)	Fy (double)
2.501758605	0.90523437	1	0.003994946	0.01	0.9900000	0.9900498	0.9960130
0.246758883	0.93216550	1	0.006510342	0.02	0.9800000	0.9800997	0.9935108
0.006581957	2.29677471	1	0.006581957	0.03	0.9700000	0.9701495	0.9934397
1.742746090	0.01539260	1	0.007866976	0.04	0.9600000	0.9601993	0.9921639

Code Preview:

```
library(readr)
dat <- read_csv("https://nshi.jp/dat.csv")
view(dat)
```

# データ読み込み (3)

event	x	t	u	edf	km	expna	Fy	trt	ntrt	
1	1	0.003994946	2.501758605	0.90523437	0.01	0.990000000	0.99004983	0.9960130	test	1
2	1	0.006510342	0.246758883	0.93216550	0.02	0.980000000	0.98009967	0.9935108	test	1
3	1	0.006581957	0.006581957	2.29677471	0.03	0.970000000	0.97014950	0.9934397	standard	0
4	1	0.007866976	1.742746090	0.01539260	0.04	0.960000000	0.96019934	0.9921639	standard	0
5	1	0.009091824	0.387182584	0.26488490	0.05	0.950000000	0.95024918	0.9909494	test	1
6	0	0.012717100	0.089949671	0.93352384	0.05	0.950000000	0.95024918	0.9873634	standard	0
7	1	0.014613740	0.824081515	0.48513873	0.06	0.93989362	0.94019372	0.9854925	test	1
8	0	0.015392597	0.202617901	0.27782263	0.06	0.93989362	0.94019372	0.9847253	test	1
9	1	0.017095131	0.838040319	0.08890984	0.07	0.92967738	0.93002957	0.9830502	test	1
10	1	0.023167162	0.760430301	0.64370236	0.08	0.91946115	0.91986541	0.9770991	test	1
11	0	0.023712310	1.880076678	0.66940532	0.08	0.91946115	0.91986541	0.9765666	test	1
12	1	0.029013252	1.596105418	0.46546444	0.09	0.90913012	0.90958769	0.9714036	standard	0
13	1	0.031725529	1.658662384	0.36623081	0.10	0.89879910	0.89930998	0.9687724	test	1
14	1	0.038379962	3.052458100	0.96312247	0.11	0.88846807	0.88903226	0.9623472	standard	0
15	0	0.040102330	1.750680133	0.53571094	0.11	0.88846807	0.88903226	0.9606911	test	1
16	1	0.084437639	0.031725529	0.05261699	0.12	0.87801551	0.87863434	0.9190290	standard	0
17	1	0.088539510	0.876960574	0.46460377	0.13	0.86756294	0.86823642	0.9152669	standard	0
18	0	0.088909842	0.014613740	0.92381014	0.13	0.86756294	0.86823642	0.9149281	test	1
19	0	0.089513152	1.835064027	1.30984019	0.13	0.86756294	0.86823642	0.9143762	test	1
20	1	0.089949671	0.519341271	2.02151392	0.14	0.85685229	0.85758335	0.9139772	test	1

Showing 1 to 20 of 100 entries

```
library(readr)
```

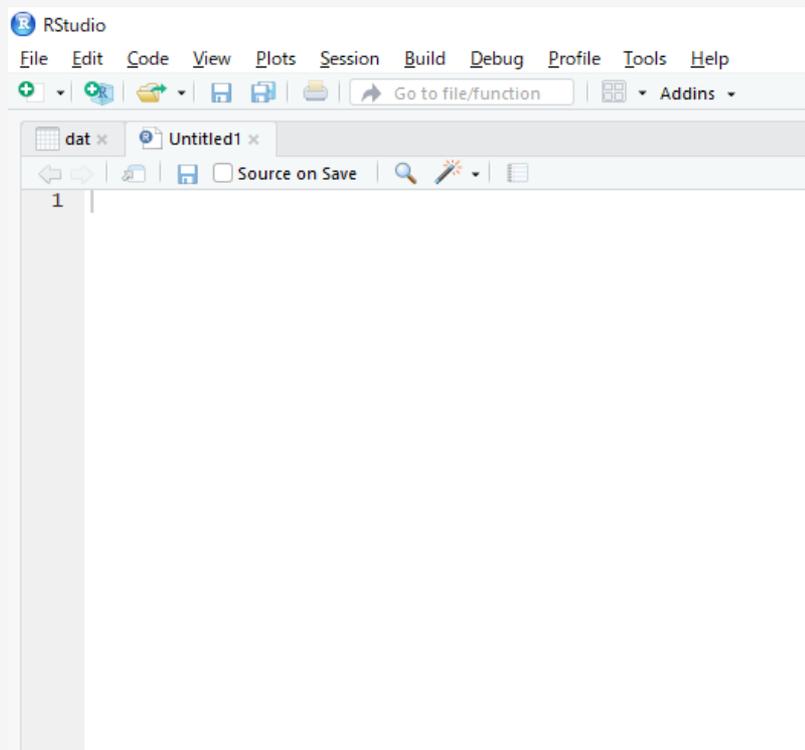
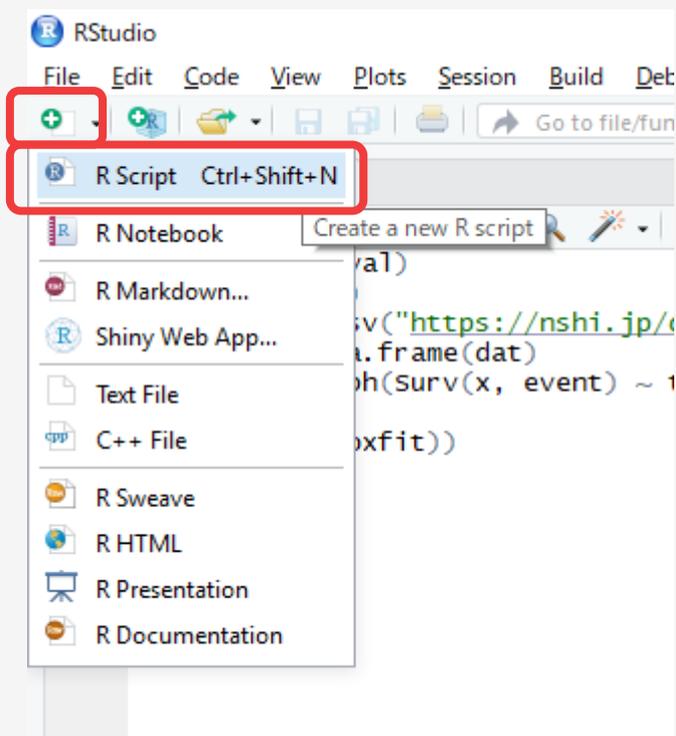
```
dat <- read_csv("https://nshi.jp/dat.csv")
```

# 仮想データ

- event: イベントあり = 1, なし = 0
- x:  $x = \min(t, u)$
- t: イベント時間
- u: 打切り時間
- edf: 経験分布関数
- km: プールしたKaplan–Meier推定量
- expna: プールしたNelson–Aalen推定量の変換推定量
- Fy: 真の分布関数値
- trt: 治療群 ("test", "standard")
- ntrt: 治療群 ("1" [test], "0" [standard])

# エディタを開く

- 左上の+アイコン → R Script



# 補足：Rの構文

- 代入 "`<-`"
  - 矢印の先に, 矢印の元のオブジェクトを代入
- いろいろな関数 "`xxx(arguments)`"
  - 関数名(引数)

```
dat <- as.data.frame(dat)
```

- `as.data.frame`関数に`dat`を指定して実行した結果を`dat`に格納

# 補足：コード補完

- コード補完機能：コードを途中まで入力  
→ Tabキー
- 候補, パッケージ名, 簡易ヘルプが表示

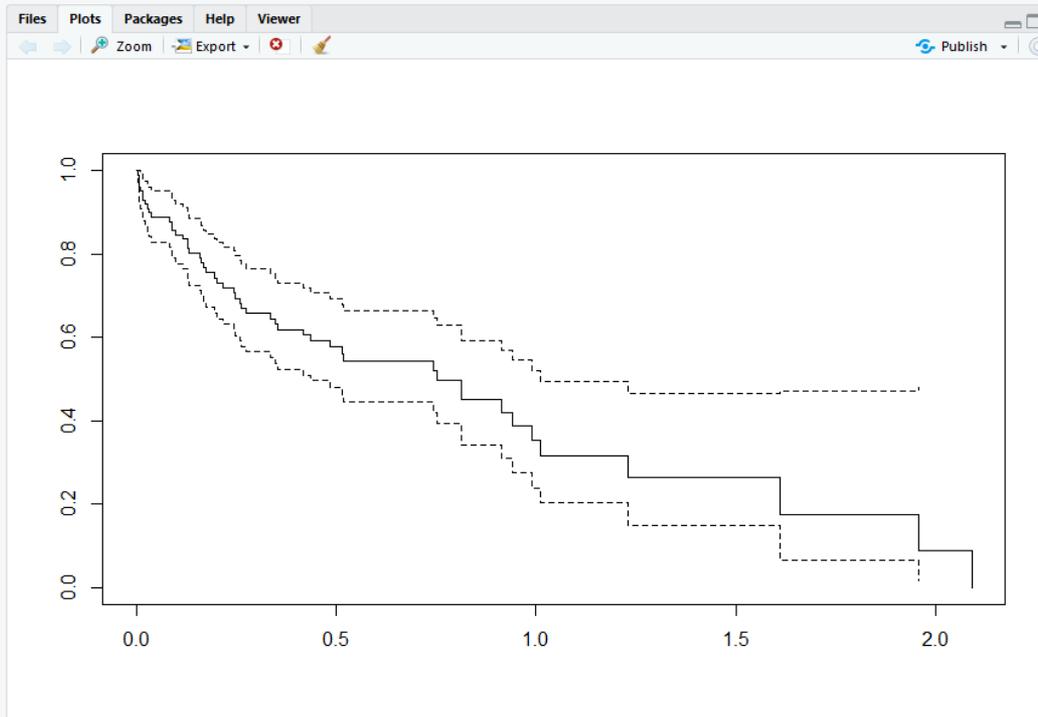
```
8  
9 pl|
```

◆ plclust	{stats}	^
◆ plnorm	{stats}	
◆ plogis	{stats}	
◆ plot	{graphics}	
◆ plot.default	{graphics}	
◆ plot.design	{graphics}	
◆ plot.ecdf	{stats}	
◆ plot.function	{graphics}	▼

```
plclust(tree, hang = 0.1, unit = FALSE, level = FALSE,  
hmin = 0, square = TRUE, labels = NULL, plot. = TRUE,  
axes = TRUE, frame.plot = FALSE, ann = TRUE, main =  
"", sub = NULL, xlab = NULL, ylab = "Height")  
  
These functions are provided for compatibility with older versions of R  
only, and may be defunct as soon as the next release.  
  
Press F1 for additional help
```

# プールした生存関数を描画 (KM)

```
dat <- as.data.frame(dat)
sfit <- survfit(Surv(x, event) ~ 1,
  data = dat)
plot(sfit)
```



# 補足：データが読み込めない場合

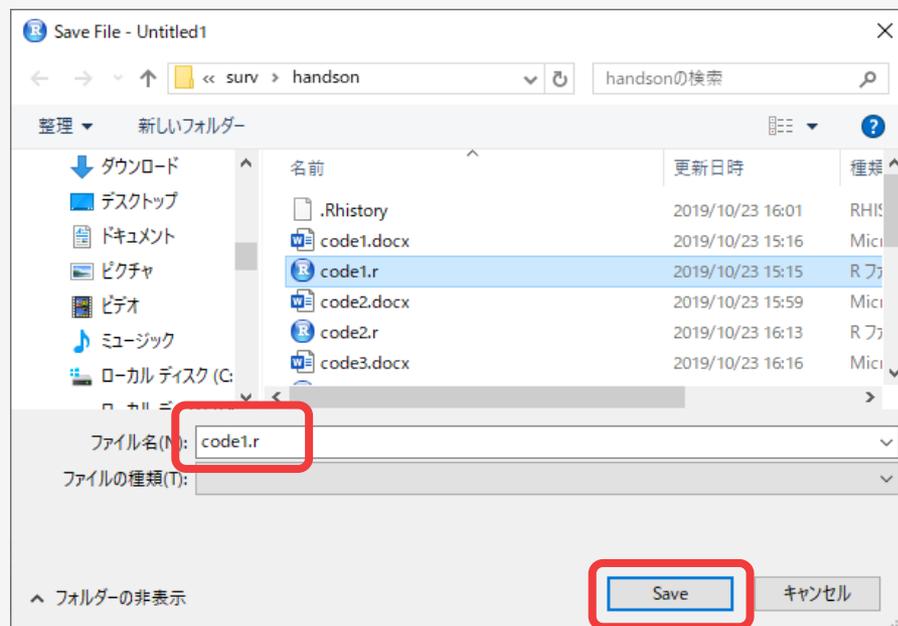
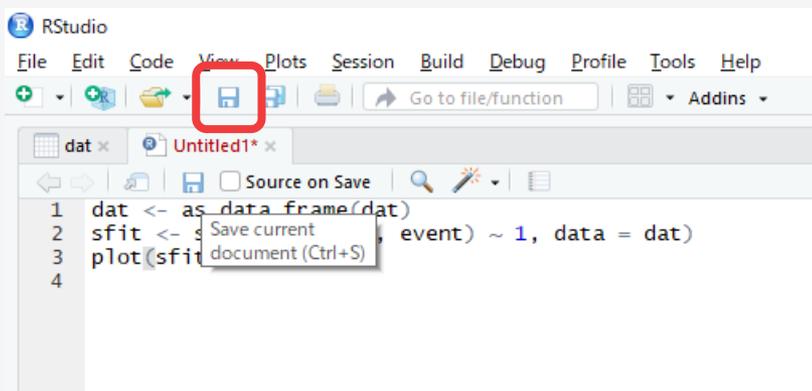
- readrが使えない場合は以下に差し替え

```
# library(readr)
# dat <-
read_csv("https://nshi.jp/dat.csv")
load(url("https://nshi.jp/dat.RData"))
```

- R言語では大文字と小文字は区別されますのでご注意ください
- # 以降がコメントとして扱われます (実行されない)

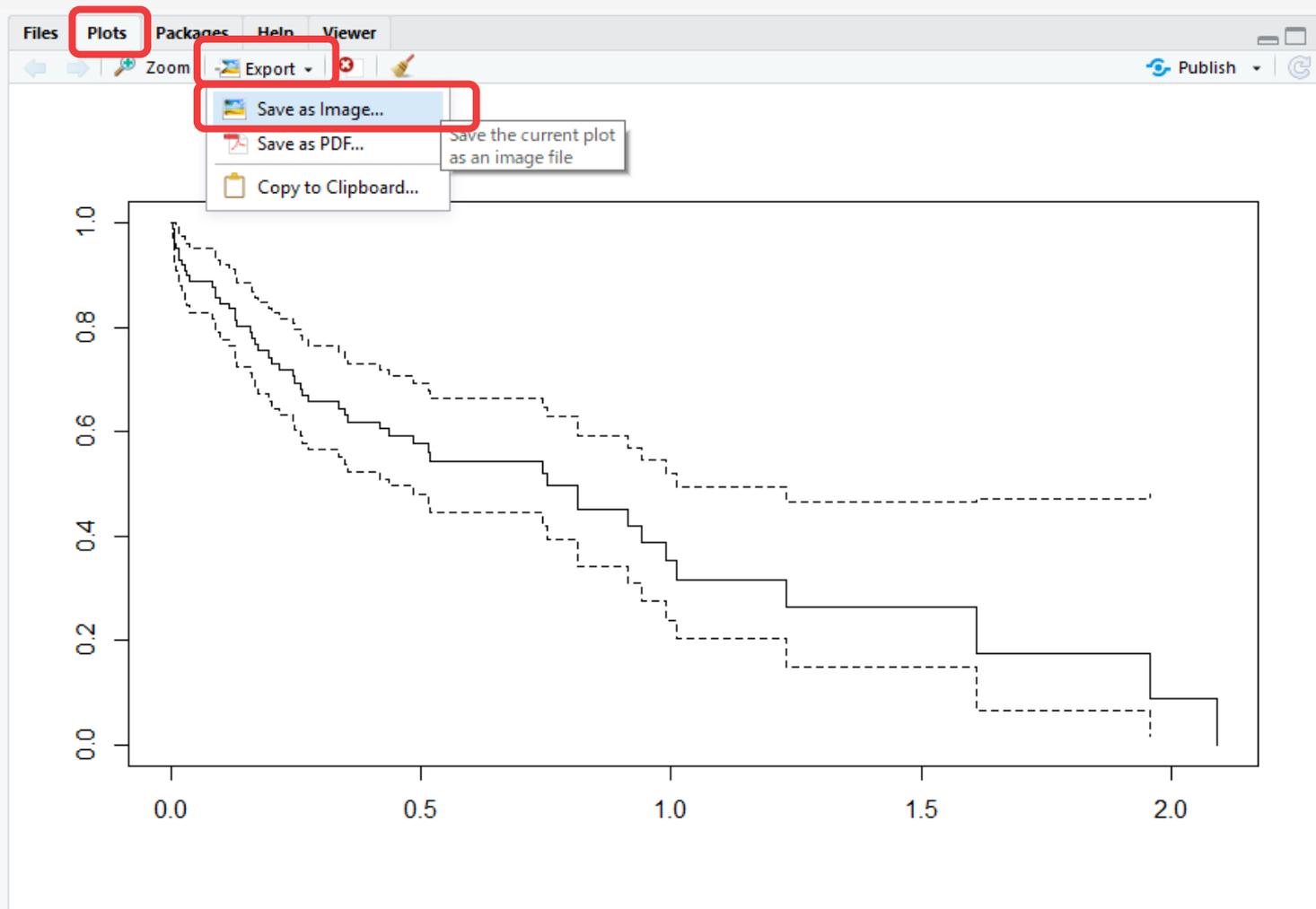
# スクリプトの保存

- 左上のフロッピーディスクアイコン → 好きな名前を付けてSave



# 画像ファイルの保存

- 右下のPlots画面のExport → 適宜選択



# 群間比較 (log-rank test)

- survdiff関数

```
survdiff(Surv(x, event) ~ trt, data = dat)
```

```
## Call:
```

```
## survdiff(formula = Surv(x, event) ~ trt, data = dat)
```

```
##
```

```
##           N Observed Expected (O-E)^2/E (O-E)^2/V
```

```
## trt=standard 42         25     17.7      2.97      4.94
```

```
## trt=test     58         26     33.3      1.58      4.94
```

```
##
```

```
## Chisq= 4.9  on 1 degrees of freedom, p= 0.03
```

# Cox regression

- coxph関数

```
coxfit <- coxph(Surv(x, event) ~ trt, data = dat)
coxfit

## Call:
## coxph(formula = Surv(x, event) ~ trt, data = dat)
##
##              coef exp(coef) se(coef)      z      p
## trttest -0.6568      0.5185   0.3005 -2.185 0.0289
##
## Likelihood ratio test=4.8  on 1 df, p=0.0285
## n= 100, number of events= 51

exp(confint(coxfit))

##              2.5 %      97.5 %
## trttest 0.2877159 0.9345059
```

# RMST

- rmst2関数

```
rfit <- rmst2(dat$x, dat$event, dat$ntrt)
rfit
```

```
##
```

```
## The truncation time, tau, was not specified. Thus, the default tau (the minimum of the largest observed time on each of the two groups) 0.991 is used.
```

```
##
```

```
## Restricted Mean Survival Time (RMST) by arm
```

```
##           Est.      se lower .95 upper .95
```

```
## RMST (arm=1) 0.658 0.057      0.546      0.769
```

```
## RMST (arm=0) 0.537 0.061      0.418      0.655
```

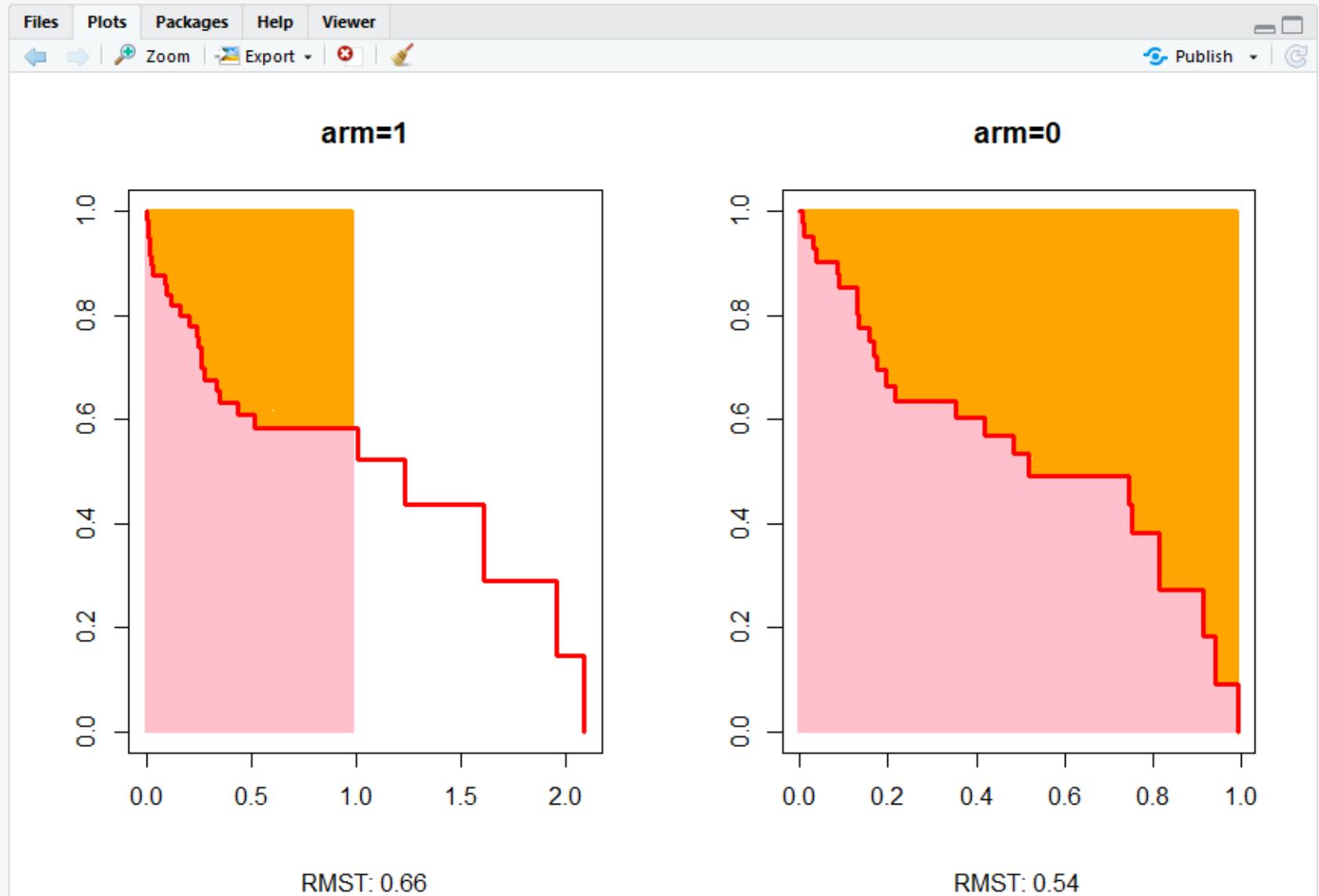
# RMST

- rmst2関数

```
## Restricted Mean Time Lost (RMTL) by arm
##           Est.      se lower .95 upper .95
## RMTL (arm=1) 0.333 0.057      0.221      0.445
## RMTL (arm=0) 0.454 0.061      0.335      0.573
##
##
## Between-group contrast
##           Est. lower .95 upper .95      p
## RMST (arm=1)-(arm=0) 0.121      -0.042      0.284 0.146
## RMST (arm=1)/(arm=0) 1.225      0.927      1.620 0.153
## RMTL (arm=1)/(arm=0) 0.734      0.480      1.122 0.153

plot(rfit)
```

# RMST



# 演習

- 解析スクリプトを一つのファイルにまとめて保存してみましよう
- パッケージの読み込み (survival, survRM2, readr)
- データの読み込み
  - データの変換 (readrの場合のみ)
- 各種解析

# 完成例

- <https://nshi.jp/codesurv.r>

```
# Load Library & data
library(survival)
library(survRM2)
library(readr)
dat <- read_csv("https://nshi.jp/dat.csv")
dat <- as.data.frame(dat)

# KM plot
sfit <- survfit(Surv(x, event) ~ 1, data = dat)
plot(sfit)

# Log-rank test
survdif(Surv(x, event) ~ trt, data = dat)

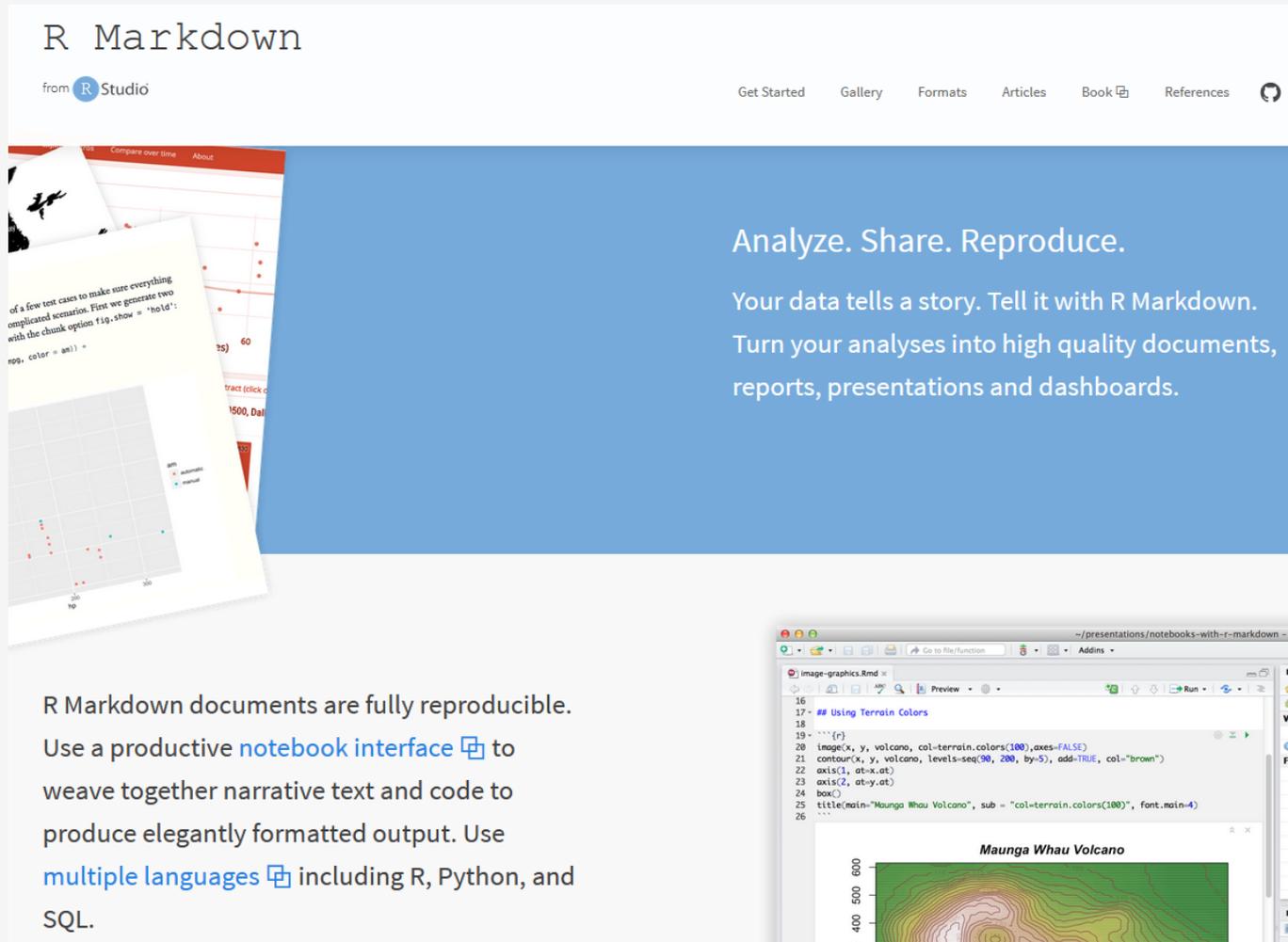
# Cox regression
coxfit <- coxph(Surv(x, event) ~ trt, data = dat)
coxfit
exp(confint(coxfit))

# RMST
rfit <- rmst2(dat$x, dat$event, dat$trt)
rfit
plot(rfit)
```

# R Markdown

# R Markdown

- Markdown記法 + Rコード向けの拡張 + α



The image shows a composite of three elements related to R Markdown. At the top is the R Markdown website header, featuring the title 'R Markdown', the R Studio logo, and navigation links for 'Get Started', 'Gallery', 'Formats', 'Articles', 'Book', and 'References'. Below this is a large blue banner with the text 'Analyze. Share. Reproduce.' and 'Your data tells a story. Tell it with R Markdown. Turn your analyses into high quality documents, reports, presentations and dashboards.' To the left of the banner is a notebook preview showing a scatter plot and some text. At the bottom right is a screenshot of the R Studio interface showing a notebook with R code for generating a contour plot of Maunga Whau Volcano and the resulting plot.

R Markdown documents are fully reproducible. Use a productive [notebook interface](#) to weave together narrative text and code to produce elegantly formatted output. Use [multiple languages](#) including R, Python, and SQL.

```
16  
17- ## Using Terrain Colors  
18  
19- ```{r}  
20 image(x, y, volcano, col=terrain.colors(100),axes=FALSE)  
21 contour(x, y, volcano, levels=seq(30, 200, by=5), add=TRUE, col="brown")  
22 axis(1, at=x.at)  
23 axis(2, at=y.at)  
24 box()  
25 title(main="Maunga Whau Volcano", sub = "col=terrain.colors(100)", font.main=4)  
26
```

Maunga Whau Volcano

# Markdown記法

- 文書に構造や修飾を指定して記述するマークアップ言語の一種

# R Markdown

### 使用例

> テキストファイルを構造と装飾付きの文書に変換。

- Nelson--Aalen推定量は $\hat{\Lambda}(t)$ です。
- リスク集合のサイズは $\bar{Y}(t)$ です。
- Kaplan--Meier推定量は $\hat{S}(t)$ です。
- 別名Product Limit Estimator

```
```\r\na <- 1\r\nb <- 2\r\na + b\r\n```
```

## R Markdown

### 使用例

テキストファイルを構造と装飾付きの文書に変換。

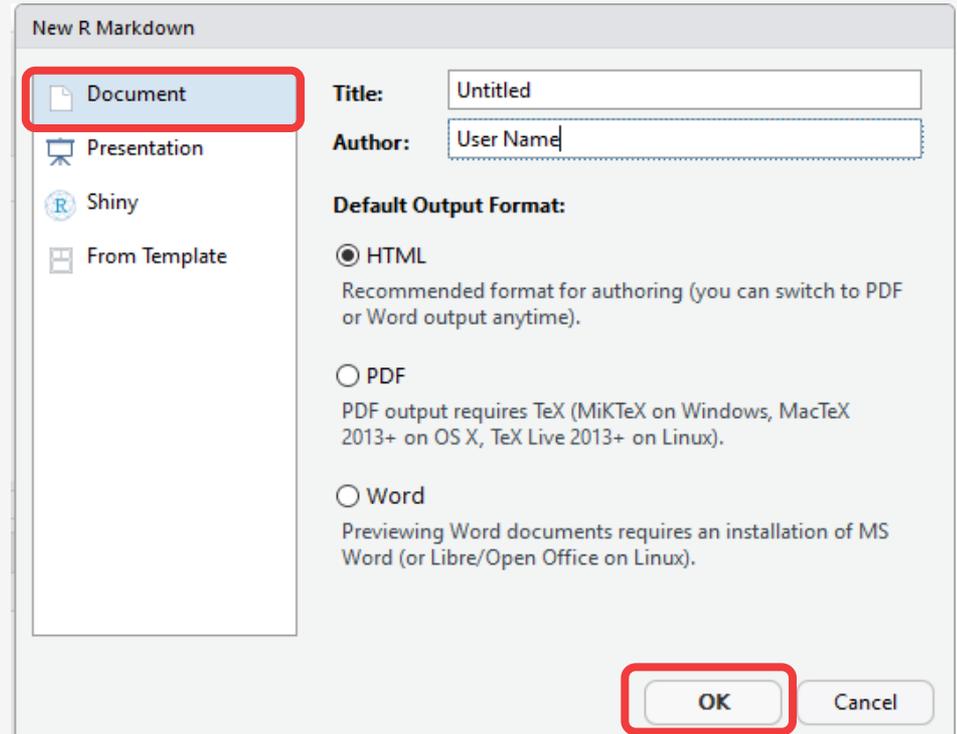
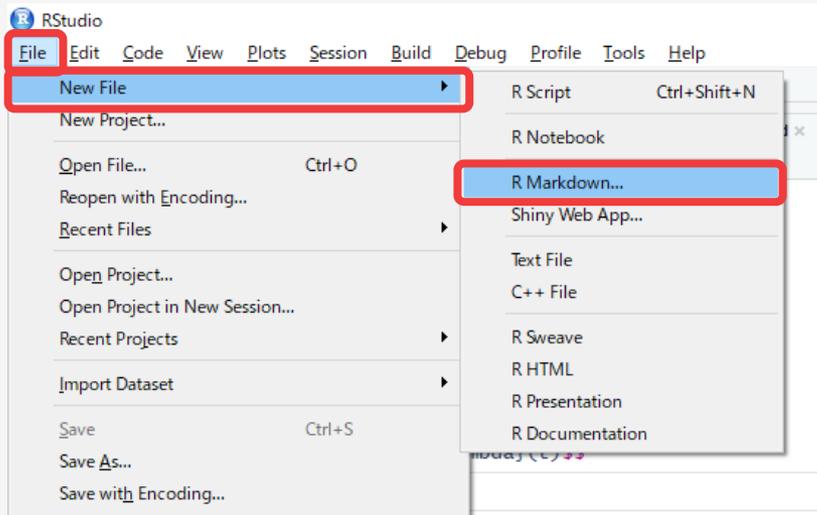
- Nelson--Aalen推定量は $\hat{\Lambda}(t)$ です。
  - リスク集合のサイズは $\bar{Y}(t)$ です。
- Kaplan--Meier推定量は $\hat{S}(t)$ です。
  - 別名Product Limit Estimator

```
a <- 1\r\nb <- 2\r\na + b
```

```
## [1] 3
```

# R Markdownファイルの新規作成

- 左上のFile → New File → R Markdown...
- Document → 各種指定 → OK



PDFやdocxも可  
スライドも作成可

# R Markdownファイルの新規作成

- Knit → Save File後に文書ファイルを生成



```
1 ---
2 title: "untitled"
3 author: "User Name"
4 date: "2019年10月24日"
5 output: html_document
6 ---
7 |
8 {r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 {r cars}
11 summary(cars)
12 ## R Markdown
13 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS word documents.
14 For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
15 when you click the knit button a document will be generated that includes both content as well as the output of any
16 embedded R code chunks within the document. You can embed an R code chunk like this:
17 {r cars}
18 summary(cars)
19 ## Including Plots
20 You can also embed plots, for example:
21 {r pressure, echo=FALSE}
22 plot(pressure)
23 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot
```

# 実行結果

rmd0.html Open in Browser Find Publish

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
## Min.   : 4.0    Min.   :  2.00
## 1st Qu.:12.0    1st Qu.: 26.00
## Median :15.0    Median : 36.00
## Mean   :15.4    Mean   : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
## Max.   :25.0    Max.   :120.00
```

## Including Plots

You can also embed plots, for example:



speed	dist	pressure
4.0	2.00	360
12.0	26.00	570
15.4	42.98	800

# 見出し

# 見出し(大)

## 見出し(中)

### 見出し(小)

見出しは6段階まで。

rmd2.html Open in Browser Find

Publish

見出し(大)

見出し(中)

見出し(小)

見出しは6段階まで。

# 数式

- インライン数式とブロック数式

### 数式

Nelson--Aalen推定量 :  $\hat{\Lambda}(t)$

\$\$

$\hat{\Lambda}(t) = \sum_{j \mid X_j \leq t, \delta_j = 1} \frac{D_j}{\bar{Y}(X_j)}$ .

\$\$

数式

Nelson--Aalen推定量:  $\hat{\Lambda}(t)$

$$\hat{\Lambda}(t) = \sum_{j \mid X_j \leq t, \delta_j = 1} \frac{D_j}{\bar{Y}(X_j)}$$

# リスト

- スペース2個以上でネスト
- 番号付きリストは1. 2. 3. ...

## ### リスト

- 1. item 1
- 2. item 2
- item 3
  - item 3-1
  - item 3-1-1

## リスト

- 1. item 1
- 2. item 2
- item 3
  - item 3-1
    - item 3-1-1

# コードブロック

- ````{r}```で囲む (オプションが色々ある)

## ### Rのコード記述

```
````{r}
a <- 1
b <- 2
a + b
````
```

### Rのコード記述

```
a <- 1
b <- 2
a + b
```

```
## [1] 3
```

# リンク

- [表示文字列](リンク先URLなど)
- ^ [footnote]

## リンク

[R Markdown](<https://rmarkdown.rstudio.com/>)

^ [footnote]

## リンク

R Markdown <sup>1</sup>

---

1. footnote↔

# テーブル

- 左, 中央, 右揃え |:--- | :---: | ---:

## ## テーブル

```
id	event	$T$
1	1	1
2	0	2
3	1	3
```

## テーブル

| id | event | <i>T</i> |
|----|-------|----------|
| 1  | 1     | 1        |
| 2  | 0     | 2        |
| 3  | 1     | 3        |

# 装飾

- 斜体, 太字, 下・上付き添字, 文中のコード, 水平線

## ## 装飾

\*強調\*, \*\*強調\*\*, 添字~添字~, 添字^添字^, `文中のコード`

水平線

\*\*\*

## 装飾

強調, 強調, 添字<sub>添字</sub>, 添字<sup>添字</sup>, 文中のコード

水平線

---

# 引用

- >をつけるだけ

## ### 引用

> "An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem."

>

> --- John Tukey

### 引用

"An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem."

— John Tukey

# 演習

- 「Rによる生存時間解析」の所で示したコードをR Markdownレポートとしてまとめてみましょう
- 各コードブロックにちょっとした説明を入れる
- 実行結果が順番に表示されるようなR Markdownファイルを作成する

# 完成例

- <https://nshi.jp/codesurv.Rmd>



The screenshot shows a web browser displaying a R Markdown document titled "Survival Analysis Using R". The document content includes:

## Survival Analysis Using R

2019-10-24

### パッケージとデータのロード

```
# load library & data
library(survival)
library(survRM2)
library(readr)
dat <- read_csv("https://nshi.jp/dat.csv")
```

```
## Parsed with column specification:
## cols(
##   event = col_double(),
##   x = col_double(),
##   t = col_double(),
##   u = col_double(),
##   edf = col_double(),
##   km = col_double(),
##   expna = col_double(),
##   Fy = col_double(),
##   trt = col_character(),
##   ntrt = col_double()
## )
```

```
dat <- as.data.frame(dat)
```

### Kaplan-Meierプロットの描画

```
# KM plot
sfit <- survfit(Surv(x, event) ~ 1, data = dat)
plot(sfit)
```

# 補足

- 日本語PDFファイルはフォント指定しないと文字化け

---

title: "Survival Analysis Using R"

date: "`r Sys.Date()`"

output:

pdf\_document:

latex\_engine: xelatex

CJKmainfont: MS Gothic

---

Survival Analysis Using R

2019-10-24

パッケージとデータのロード

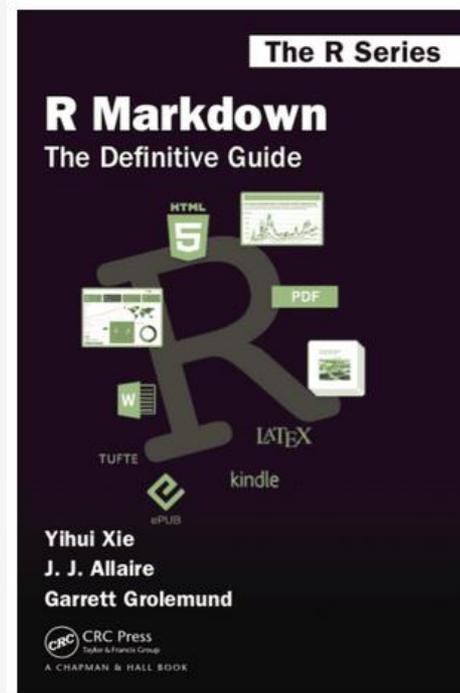
```
# load library & data
library(survival)
library(survRM2)
library(readr)
```

# R Markdown: The Definitive Guide

- 解説書のオンラインバージョンがFreeで公開されています
- <https://bookdown.org/yihui/rmarkdown/>

## R Markdown: The Definitive Guide

*Yihui Xie, J. J. Allaire, Garrett Golemund*



# R Markdown補足

- LaTeXとの親和性が高く (数式, BibTeX等), 論文原稿なども作成可

## 2.1. Random-effects model

Let us consider the case of combining information from a series of  $K$  comparative studies by the random-effects model (Cochran 1937; DerSimonian and Laird 1986). The random-effects model can be defined as follows:

$$\begin{aligned} Y_k &= \theta_k + \epsilon_k, \\ \theta_k &= \mu + u_k, \end{aligned}$$

where  $Y_k$  is a random variable of an effect size estimate from the  $k$ th study,  $\theta_k$  is the true treatment effect in the  $k$ th study,  $\epsilon_k$  is a random error within a study,  $\mu$  is an unknown parameter of the overall mean effect,  $u_k$  is a random variable reflecting study-specific deviation from the overall mean effect, and  $\epsilon_k$  and  $u_k$  are assumed to be mutually independent and normally distributed, which means that  $\epsilon_k \sim N(0, \sigma_k^2)$ ,  $u_k \sim N(0, \tau^2)$ ,  $\sigma_k^2 > 0$ ,  $\tau^2 > 0$ . The parameters of the within-study variances,  $\sigma_k^2$ , are known and replaced by their efficient estimates (Biggerstaff and Jackson 2008). The parameter of the heterogeneity variance is an unknown parameter that should be estimated.

The overall mean effect,  $\mu$ , can be estimated as  $\hat{\mu} = (\sum_{k=1}^K \hat{w}_k Y_k) / (\sum_{k=1}^K \hat{w}_k)$ , where  $\hat{w}_k = (\sigma_k + \hat{\tau})^{-1}$ , and  $\hat{\tau}$  is an estimator of the heterogeneity variance that has been proposed by a number of researchers [Sidik and Jonkman (2007); Veroniki et al. (2016); Petropoulou and Mavridis (2017); Langan et al. (2018); see also Appendix A.1]. Similarly, many methods to estimate a confidence interval of the overall mean effect,  $\mu$ , have also been proposed [Brockwell and Giorganni (2008); Marín-Martínez (2008); Veroniki et al. (2018); see also Appendix

## 2.2. Higgins-Thompson-Spiegelhalter prediction

Higgins, Thompson, and Spiegelhalter (2009) proposed a simple prediction interval expressed as

$$\left[ \hat{\mu} - t_{K-2}^{\alpha} \sqrt{\hat{\tau}_{DL}^2 + \widehat{\text{Var}}[\hat{\mu}]}, \hat{\mu} + t_{K-2}^{\alpha} \sqrt{\hat{\tau}_{DL}^2 + \widehat{\text{Var}}[\hat{\mu}]} \right]$$

where  $\widehat{\text{Var}}[\hat{\mu}] = 1 / (\sum_{k=1}^K \hat{w}_k)$  is an approximate variance estimator of  $\mu$ ,  $t_{K-2}^{\alpha}$  is the  $100(1 - \alpha/2)$  percentile of the  $t$  distribution with  $K - 2$  degrees of freedom,  $\hat{\tau}_{DL}^2$  is the DerSimonian-Laird estimator of the heterogeneity variance (DerSimonian and Laird 1986) that is defined as  $\hat{\tau}_{DL}^2 = \max\{0, \{Q - (K - 1)\} / \{S_1 + S_2 / S_1\}\}$ ,  $Q = \sum_{k=1}^K v_k (Y_k - \bar{Y})^2$ ,  $S_r = \sum_{k=1}^K v_k^r$ ,  $v_k = (\sigma_k^2)^{-1}$ , and  $\bar{Y} = (\sum_{k=1}^K v_k Y_k) / (\sum_{k=1}^K v_k)$ . This prediction interval is essentially based on the

もはや何でもできる  
のでは . . .

Rの基本知識と基本関数によるデータ  
ハンドリング

# Rの基本的な使用方法

# Rの特徴

- 関数型プログラミング言語
  - 基本的文法は必修
- 複雑な構造 (関数もデータも)
  - ベクトル処理 (Rはデータ処理言語)
  - リスト (別構造のオブジェクトの一括保持), データフレーム (表形式データ)
- データ読み込み + 変数を指定して解析 :  
だけなら先程のように簡単...

# Rの分かりにくい点

- 多彩なデータ
  - データ型：実数, 整数, 文字列, ...
  - データ構造：ベクトル型, 行列型, ...

柔軟なプログラミングができる

一方で学習コストが高い

- 多彩なContributed Package

# データ型

| データ型 | 説明                        |
|------|---------------------------|
| 論理値  | TRUE or FALSE             |
| 整数   | ..., -2, -1, 0, 1, 2, ... |
| 実数   | 1.23456789...             |
| 複素数  | $1+1i$                    |
| 文字列  | "text"                    |

- 5つのデータ型が定義されている
- 下に行くほど優先される

# 型チェック

- `is.logical()`
- `is.integer()`
- `is.numeric()`
- `is.complex()`
- `is.character()`
- `mode()`

```
int <- 1L
txt <- "text"
is.integer(int)
## [1] TRUE
is.integer(txt)
## [1] FALSE
is.character(int)
## [1] FALSE
is.character(txt)
## [1] TRUE
mode(txt)
## [1] "character"
```

# 型変換

- `as.logical()`
- `as.integer()`
- `as.numeric()`
- `as.complex()`
- `as.character()`

```
lgc <- TRUE
int <- 1L
num <- 1.5
cmp <- 1+2i
txt <- "string"
as.integer(lgc)
## [1] 1
as.integer(num)
## [1] 1
as.numeric(txt)
## Warning: 強制変換により
NA が生成されました
## [1] NA
as.complex(num)
## [1] 1.5+0i
```

# 自動型変換

- 優先順に自動的に型が揃えられる
- 定義されていない演算はエラー

```
lgc <- TRUE
num <- 1.5
txt <- "text"
lgc + num
## [1] 2.5
mode(lgc + num)
## [1] "numeric"
paste0(lgc, num, txt)
## [1] "TRUE1.5text"
mode(paste(lgc, num, txt))
## [1] "character"
num + txt
Error in num + txt : non-numeric argument to binary operator
```

# 注意点

- 自動変換が行われることがある点には少し注意
- SAS(ほぼ数値・文字のみ)等ではあまり意識しないポイントなので覚えておくと良い

# データ構造

| データ構造          | 説明                                  |
|----------------|-------------------------------------|
| scalar         | スカラー                                |
| vector         | ベクトル                                |
| matrix         | 行列                                  |
| data.frame     | 行列の拡張, 列ごとにデータ型が違って良い, SASデータセットと同じ |
| array          | 配列                                  |
| list           | 任意の種類オブジェクトを名前付きで含む                 |
| factor         | 因子型, 数値+コーディング                      |
| ordered.factor | 順位付きの因子型                            |

# vector型

- 関数c()で作ることができる
- 異なるデータ型の混合は強制変換

```
c(1, 2, 3.4, -5.6) # numeric
## [1] 1.0 2.0 3.4 -5.6
c("one", "two", "three") # character
## [1] "one" "two" "three"
c(TRUE, FALSE, TRUE, FALSE) # Logical
## [1] TRUE FALSE TRUE FALSE
c(1L, "true") # character!!!
## [1] "1" "true"
```

# matrix型

- ベクトル型を引数に持つ関数matrix()に、行数と列数を指定する
- 行名と列名をつけることができる (Helpも適宜参照：?matrix)

```
matrix(c(1, 2, 3.4, -5.6), nrow = 2, ncol = 2)
```

```
##      [,1] [,2]  
## [1,]    1  3.4  
## [2,]    2 -5.6
```

```
matrix(c(1, 2, "three", "four"), nrow = 2, ncol = 2)
```

```
##      [,1] [,2]  
## [1,] "1"  "three"  
## [2,] "2"  "four"
```

# data.frame型

- matrix型と違い, 列ごとに違うデータ型が指定できる
- 列数が揃っていないとエラー

```
data.frame(  
  x = c(1, 2, 3),  
  y = c("one", "two", "three"),  
  z = c(TRUE, TRUE, TRUE)  
)
```

```
##      x      y      z  
## 1 1    one TRUE  
## 2 2    two TRUE  
## 3 3   three TRUE
```

# data.frameの列へのアクセス

- data.frame\$nameによりアクセス可

```
df <- data.frame(id = 1:4, z = 0)
df$id
## [1] 1 2 3 4
df$z
## [1] 0 0 0 0
```

# 要素の抽出

- vector型などのデータでは要素の一部を抽出できる
- 何番目の要素かを[]で囲んで指定
- matrixやdata.frameでは[行, 列]で指定して抽出できる

```
vc <- c(1, 2, 3, 4)
mt <- matrix(vc, ncol = 2)
df <- data.frame(mt)
df

##      X1 X2
## 1    1  3
## 2    2  4

vc[3]

## [1] 3

mt[2, 1]

## [1] 2

df[2, 1]

## [1] 2
```

# 要素の抽出

- 抽出の範囲指定が可  
a:bでaからbまでの  
整数列
- マイナス指定すると  
その要素を除くこと  
を意味

```
vc <- c(1, 2, 3, 4)
mt <- matrix(vc, ncol = 2)
df <- data.frame(mt)
vc[3:4]

## [1] 3 4

mt[1:2, 1]

## [1] 1 2

df[1:2, -1]

## [1] 3 4
```

# オブジェクトの結合

- 関数cbind()は列方向にオブジェクトを結合
- 関数rbind()は行方向にオブジェクトを結合
- vector, matrix, data.frameの結合にたまに使う

```
vc <- c(1, 2, 3, 4)
mt <- matrix(vc, ncol = 2)
cbind(vc, vc, vc)

##          vc vc vc
## [1,]    1  1  1
## [2,]    2  2  2
## [3,]    3  3  3
## [4,]    4  4  4

rbind(mt, mt)

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

# 便利な関数

- length: 要素数
- dim: 次元数
- str: 構造チェック
- class: データ構造の種類を表示
- names: 名前(列名など)表示

```
vc <- c(1, 2, 3, 4)
mt <- matrix(vc, ncol = 2)
df <- data.frame(mt)
length(vc)
## [1] 4
dim(df)
## [1] 2 2
str(df)
## 'data.frame': 2 obs. of 2 variables:
## $ X1: num 1 2
## $ X2: num 3 4
class(df)
## [1] "data.frame"
names(df)
## [1] "X1" "X2"
```

# array型

- 3次元以上の配列
- あまり使わない

```
ar <- array(1:6, c(2, 2, 2))
ar

##      , , 1
##
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
##
##      , , 2
##
##      [,1] [,2]
## [1,]    5    1
## [2,]    6    2

ar[2, 1, 2]

## [1] 6
```

# list型

- サイズ・型・種類が異なるオブジェクトもまとめて格納できる
- listの中にlistを入れることもできる
- ほぼ何でも入る

```
list(  
  a = c(1, 2),  
  b = matrix(1:4, ncol = 2),  
  c = list(d = 1, e = data.frame(f = 1))  
)  
## $a  
## [1] 1 2  
##  
## $b  
##      [,1] [,2]  
## [1,]    1    3  
## [2,]    2    4  
##  
## $c  
## $c$d  
## [1] 1  
##  
## $c$e  
##    f  
## 1 1
```

# listへのアクセス

- list\$nameまたはlist[["name"]]によりアクセス
- 取り出したlist内のオブジェクトの各要素にアクセスする場合は続けて[,]等で記述

```
lst <- list(
  a = c(1, 2),
  b = matrix(1:4, ncol = 2),
  c = list(d = 1, e = data.frame
(f = 1))
)
lst$a
## [1] 1 2
lst[["a"]]
## [1] 1 2
lst$c$e[1,1]
## [1] 1
class(lst$c$e)
## [1] "data.frame"
```

# factor型

- 名義尺度 (orderedは順序) データの定義
- 一部の関数 (例えばsummary()) でfactor型専用の挙動が準備されている

```
blood <- c(
  rep("A", 3), rep("B", 4),
  rep("AB", 5), rep("O", 6)
)
blood <- factor(blood)
summary(blood)

##  A AB  B  O
##  3  5  4  6
```

- repは指定した回数の繰り返し

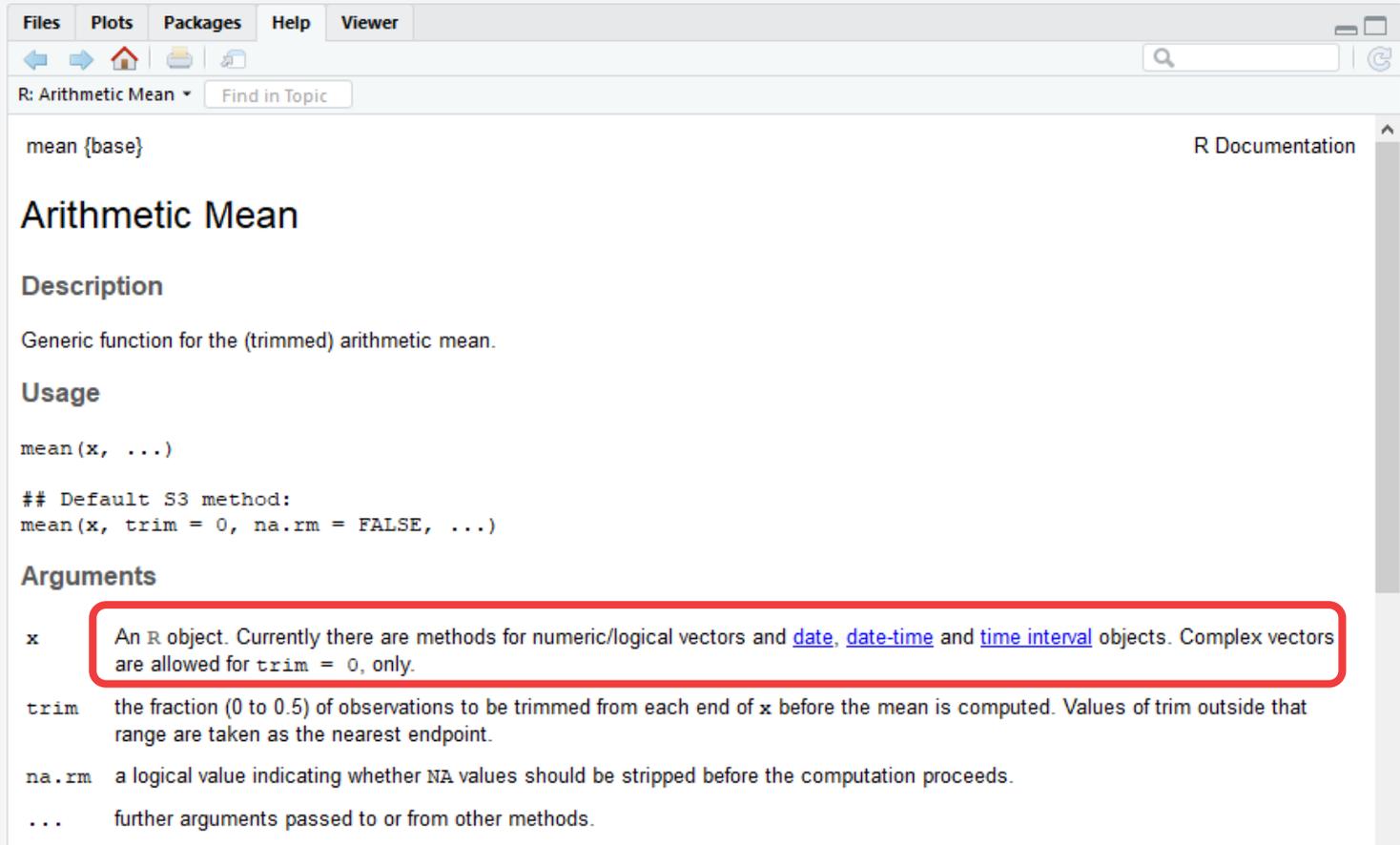
# 注意点

- データ構造は柔軟で便利だが, みんなが好き勝手に使う
- 構造毎の挙動の違いを知っておくと良い
- どのデータ型, データ構造に対応しているかは関数によって異なっている
- 関数の実行結果が, どのデータ構造で返ってくるかも, それぞれ異なっている

基本的にはヘルプを読むしかない...

# ヘルプの表示

?mean



The screenshot shows the R help window for the `mean` function. The window title is "R: Arithmetic Mean" and it includes a search bar and navigation icons. The content is structured as follows:

- mean {base}** (R Documentation)
- Arithmetic Mean**
- Description**  
Generic function for the (trimmed) arithmetic mean.
- Usage**  
`mean(x, ...)`  
## Default S3 method:  
`mean(x, trim = 0, na.rm = FALSE, ...)`
- Arguments**
  - x**: An R object. Currently there are methods for numeric/logical vectors and [date](#), [date-time](#) and [time interval](#) objects. Complex vectors are allowed for `trim = 0`, only.
  - trim**: the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of `trim` outside that range are taken as the nearest endpoint.
  - na.rm**: a logical value indicating whether NA values should be stripped before the computation proceeds.
  - ...**: further arguments passed to or from other methods.

# ヘルプ問題

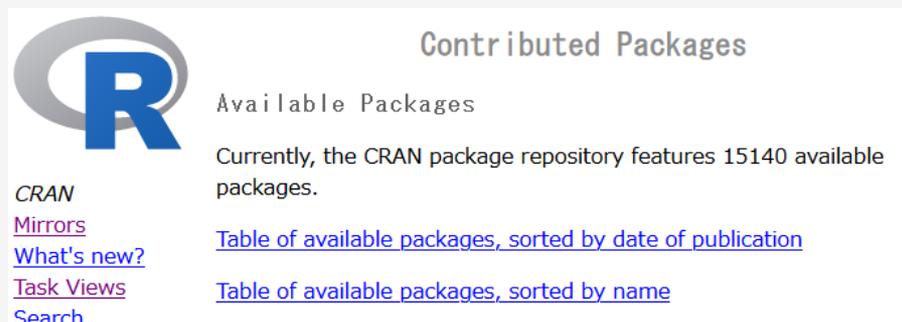
- ユーザによるContributed Packageでは詳細がない場合も...

ソースコードを読む...

```
library(survival)
Surv
## function (time, time2, event, type = c("right", "left",
## "interval",
## "counting", "interval2", "mstate"), origin = 0)
## {
##   if (missing(time))
##     stop("Must have a time argument")
```

# Contributed Packages

- ユーザがボランティアで作成, 登録したパッケージ : Contributed Packages
- 論文出版時に公開 : 最新手法が使える
- ボランティア : 質のばらつき, ヘルプが簡素, サポートは作者の気持ち次第
- 15140個... : CRAN Task Views



The screenshot shows the CRAN website's 'Contributed Packages' section. On the left is the CRAN logo (a blue 'R' inside a grey circle) and a list of links: 'CRAN', 'Mirrors', 'What's new?', 'Task Views', and 'Search'. On the right, the text reads 'Contributed Packages' followed by 'Available Packages' and 'Currently, the CRAN package repository features 15140 available packages.' Below this are two blue links: 'Table of available packages, sorted by date of publication' and 'Table of available packages, sorted by name'.

# ヘルプファイルが簡素な理由

- 昔：開発ツールがない
  - 手作業で複雑な仕様のファイル群をたくさん作る必要
- 今：RStudioと開発用ツールによってかなり改善
  - 忙しい中で開発ツールを学習するのもやっぱり大変

# CRAN Task Views

- CRAN Task Views : 特定のトピックに関連するパッケージリスト (頻繁に更新)

## CRAN Task View: Meta-Analysis

**Maintainer:** Michael Dewey

**Contact:** lists at dewey.myzen.co.uk

**Version:** 2019-09-17

**URL:** <https://CRAN.R-project.org/view=MetaAnalysis>

This task view covers packages which include facilities for meta-analysis of summary statistics from primary studies. The task view does not consider the meta-analysis of individual participant data (IPD) which can be handled by any of the standard linear modelling functions but does include some packages which offer special facilities for IPD.

The standard meta-analysis model is a form of weighted least squares and so any of the wide range of R packages providing weighted least squares would in principle be able to fit the model. The advantage of using a specialised package is that (a) it takes care of the small tweaks necessary (b) it provides a range of ancillary functions for displaying and investigating the model. Where the model is referred to below it is this model which is meant.

Where summary statistics are not available a meta-analysis of significance levels is possible. This is not completely unconnected with the problem of adjustment for multiple comparisons but the packages below which offer this, chiefly in the context of genetic data, also offer additional functionality.

## pimeta: Prediction Intervals for Random-Effects Meta-Analysis

An implementation of prediction intervals for random-effects meta-analysis: Higgins et al. (2009) <[doi:10.1111/j.1467-985X.2008.00552.x](https://doi.org/10.1111/j.1467-985X.2008.00552.x)>, Partlett and Riley (2017) <[doi:10.1002/sim.7140](https://doi.org/10.1002/sim.7140)>, and Nagashima et al. (2019) <[doi:10.1177/0962280218773520](https://doi.org/10.1177/0962280218773520)>, <[arXiv:1804.01054](https://arxiv.org/abs/1804.01054)>.

Version: 1.1.3

Depends: R (≥ 3.2.0)

Imports: stats, [Rcpp](#) (≥ 1.0.2), [ggplot2](#) (≥ 3.2.1), [scales](#) (≥ 1.0.0), utils

LinkingTo: [Rcpp](#), [RcppEigen](#)

Suggests: [knitr](#), [rmarkdown](#)

Published: 2019-09-17

Author: Kengo Nagashima  [aut, cre], Hisashi Noma [aut], Toshi A. Furukawa [aut]

Maintainer: Kengo Nagashima <nshi1201 at gmail.com>

License: [GPL-3](#)

NeedsCompilation: yes

Citation: [pimeta citation info](#)

Materials: [README](#) [NEWS](#)

In views: [MetaAnalysis](#)

CRAN checks: [pimeta results](#)

# 基本関数でのデータハンドリング

- 統計解析の際はdata.frame型の構造のデータが一番よく使われる
  - 縦結合
  - 横結合 (マッチマージ)
  - 変数の追加
  - survivalとsurvRM2パッケージで使うデータ構造を調べてみよう

# 仮想データ

```
set.seed(1234)
df1 <- data.frame(id = 1:4, t = rexp(4))
df2 <- data.frame(id = 5:10, t = rexp(6))
df3 <- data.frame(id = 1:10, a = runif(10))
```

```
df1
##      id          t
## 1     1 2.501758605
## 2     2 0.246758883
## 3     3 0.006581957
## 4     4 1.742746090
```

```
df2
##      id          t
## 1     5 0.38718258
## 2     6 0.08994967
## 3     7 0.82408151
## 4     8 0.20261790
## 5     9 0.83804032
## 6    10 0.76043030
```

```
df3
##      id          a
## 1     1 0.18672279
## 2     2 0.23222591
## 3     3 0.31661245
## 4     4 0.30269337
## 5     5 0.15904600
## 6     6 0.03999592
## 7     7 0.21879954
## 8     8 0.81059855
## 9     9 0.52569755
## 10    10 0.91465817
```

# 縦結合

- 関数 `rbind()`

```
df12 <- rbind(df1, df2)
```

```
df12
```

```
##      id      t
## 1     1 2.501758605
## 2     2 0.246758883
## 3     3 0.006581957
## 4     4 1.742746090
## 5     5 0.387182584
## 6     6 0.089949671
## 7     7 0.824081515
## 8     8 0.202617901
## 9     9 0.838040319
## 10    10 0.760430301
```

# 縦結合の注意点

- 列数が違ったり, 名前が違っているとエラー

```
set.seed(1234)
```

```
df1 <- data.frame(id = 1:4, t = rexp(4), c = rexp(4))
```

```
df2 <- data.frame(id = 5:10, t = rexp(6))
```

Error in rbind(deparse.level, ...) : 引数の列の数が一致しません

```
set.seed(1234)
```

```
df1 <- data.frame(id = 1:4, t = rexp(4))
```

```
df2 <- data.frame(id = 5:10, b = rexp(6))
```

Error in match.names(clabs, names(xi)) : 名前が以前の名前に一致しません

# 横結合 (マッチマージ)

- 関数merge()

```
merge(df12, df3, by = "id")
```

```
##      id          t          a
## 1     1 2.501758605 0.18672279
## 2     2 0.246758883 0.23222591
## 3     3 0.006581957 0.31661245
## 4     4 1.742746090 0.30269337
## 5     5 0.387182584 0.15904600
## 6     6 0.089949671 0.03999592
## 7     7 0.824081515 0.21879954
## 8     8 0.202617901 0.81059855
## 9     9 0.838040319 0.52569755
## 10    10 0.760430301 0.91465817
```

# 横結合 (マッチマージ) の注意点

- デフォルトではマージ成功行のみが返ってくる

```
set.seed(1234)
df1 <- data.frame(id = 1:4, t = rexp(4))
df2 <- data.frame(id = 5:10, t = rexp(6))
df3 <- data.frame(id = 1:10, a = runif(10))
df12 <- rbind(df1, df2)
merge(df12[1:5,], df3, by = "id")
```

```
##      id          t          a
## 1    1 2.501758605 0.1867228
## 2    2 0.246758883 0.2322259
## 3    3 0.006581957 0.3166125
## 4    4 1.742746090 0.3026934
## 5    5 0.387182584 0.1590460
```

# 横結合 (マッチマージ) の注意点

- 全例出力の場合：all = TRUEオプション

```
merge(df12[1:5,], df3, by = "id", all = TRUE)
```

```
##      id          t          a
## 1     1 2.501758605 0.18672279
## 2     2 0.246758883 0.23222591
## 3     3 0.006581957 0.31661245
## 4     4 1.742746090 0.30269337
## 5     5 0.387182584 0.15904600
## 6     6           NA 0.03999592
## 7     7           NA 0.21879954
## 8     8           NA 0.81059855
## 9     9           NA 0.52569755
## 10    10          NA 0.91465817
```

# 変数の追加

- 新たな変数を\$で定義して <- で代入

```
df12 <- rbind(df1, df2)
df123 <- merge(df12, df3, by = "id", all = TRUE)
df123$c <- df123$t*df123$a
df123
```

| ##    | id | t           | a          | c           |
|-------|----|-------------|------------|-------------|
| ## 1  | 1  | 2.501758605 | 0.18672279 | 0.467135346 |
| ## 2  | 2  | 0.246758883 | 0.23222591 | 0.057303806 |
| ## 3  | 3  | 0.006581957 | 0.31661245 | 0.002083929 |
| ## 4  | 4  | 1.742746090 | 0.30269337 | 0.527517688 |
| ## 5  | 5  | 0.387182584 | 0.15904600 | 0.061579842 |
| ## 6  | 6  | 0.089949671 | 0.03999592 | 0.003597620 |
| ## 7  | 7  | 0.824081515 | 0.21879954 | 0.180308657 |
| ## 8  | 8  | 0.202617901 | 0.81059855 | 0.164241777 |
| ## 9  | 9  | 0.838040319 | 0.52569755 | 0.440555740 |
| ## 10 | 10 | 0.760430301 | 0.91465817 | 0.695533784 |

# 関数survfit()の例

- survfit(), survdiff(), coxph()は同じ構造
- それぞれ同じ長さのベクトル型のオブジェクトを適切に指定して解析

```
survfit(Surv(dat$x, dat$event) ~ dat$trt)
```

```
## Call: survfit(formula = Surv(dat$x, dat$event) ~ dat$trt)
```

```
##
```

```
##           n events median 0.95LCL 0.95UCL
```

```
## dat$trt=standard 42      25  0.519   0.215      NA
```

```
## dat$trt=test     58      26  1.230   0.435      NA
```

# 関数survfit()の例

- formula+data.frame形式も使うことができる

```
survfit(Surv(x, event) ~ trt, data = dat)
```

```
## Call: survfit(formula = Surv(x, event) ~ trt, data = dat)
```

```
##
```

```
##           n events median 0.95LCL 0.95UCL
```

```
## trt=standard 42      25  0.519   0.215     NA
```

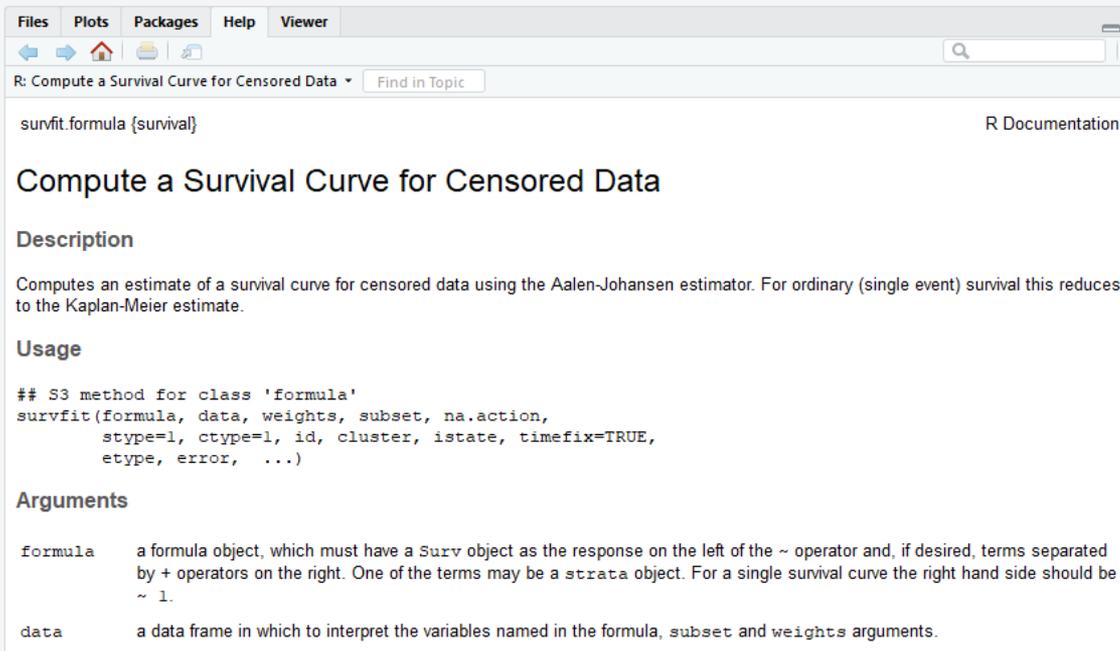
```
## trt=test      58      26  1.230   0.435     NA
```

- formulaは回帰モデル式の記述にも使用
  - factor変換, 交互作用, 変数変換, 対比, スプライン, 層別, 変量効果, オフセット等

# 関数survfit()の例

- ヘルプファイルにかなり細かい情報が書いてある (survivalパッケージはSystem Library)

?survfit.formula



The screenshot shows the R help window for the `survfit.formula` function. The window title is "R: Compute a Survival Curve for Censored Data". The content includes the following sections:

- survfit.formula {survival}** (R Documentation)
- Compute a Survival Curve for Censored Data**
- Description**  
Computes an estimate of a survival curve for censored data using the Aalen-Johansen estimator. For ordinary (single event) survival this reduces to the Kaplan-Meier estimate.
- Usage**  
## S3 method for class 'formula'  
`survfit(formula, data, weights, subset, na.action, stype=1, ctype=1, id, cluster, istate, timefix=TRUE, etype, error, ...)`
- Arguments**
  - `formula`: a formula object, which must have a `Surv` object as the response on the left of the `~` operator and, if desired, terms separated by `+` operators on the right. One of the terms may be a `strata` object. For a single survival curve the right hand side should be `~ 1`.
  - `data`: a data frame in which to interpret the variables named in the formula, `subset` and `weights` arguments.

# 関数rmst2()の例

- こちらはベクトル形式のみ
- 治療群は1=test, 0=controlで指定する

```
rfit <- rmst2(dat$x, dat$event, dat$ntrt)
```

```
rfit
```

```
##
```

```
## The truncation time, tau, was not specified. Thus, the default tau (the minimum of the largest observed time on each of the two groups) 0.991 is used.
```

```
##
```

```
## Restricted Mean Survival Time (RMST) by arm
```

```
##           Est.      se lower .95 upper .95
```

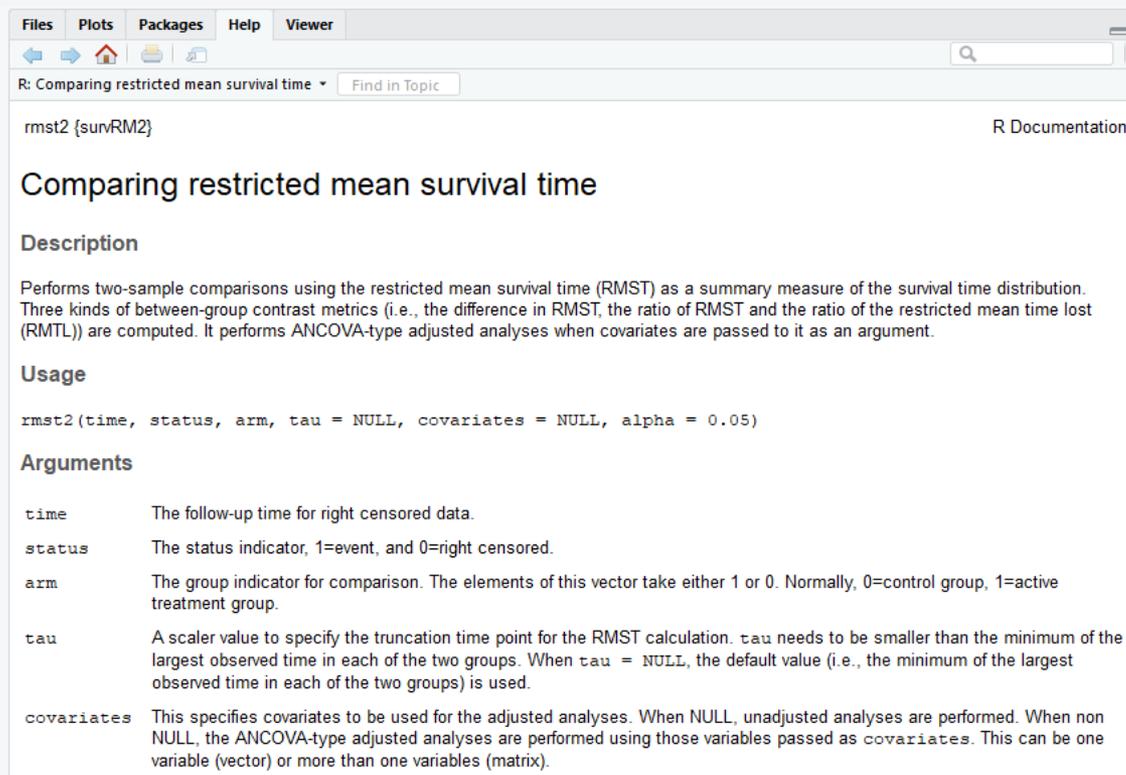
```
## RMST (arm=1) 0.658 0.057      0.546      0.769
```

```
## RMST (arm=0) 0.537 0.061      0.418      0.655
```

# 関数rmst2()の例

- こちらもヘルプファイルを見ると色々と書いてある

?rmst2



The screenshot shows the R help window for the `rmst2` function. The title bar indicates the topic is "R: Comparing restricted mean survival time". The main content includes the package name `rmst2 {survRM2}`, the title "Comparing restricted mean survival time", a description of the function's purpose, its usage, and a list of arguments with their descriptions.

**rmst2 {survRM2}** R Documentation

## Comparing restricted mean survival time

### Description

Performs two-sample comparisons using the restricted mean survival time (RMST) as a summary measure of the survival time distribution. Three kinds of between-group contrast metrics (i.e., the difference in RMST, the ratio of RMST and the ratio of the restricted mean time lost (RMTL)) are computed. It performs ANCOVA-type adjusted analyses when covariates are passed to it as an argument.

### Usage

```
rmst2(time, status, arm, tau = NULL, covariates = NULL, alpha = 0.05)
```

### Arguments

|                         |                                                                                                                                                                                                                                                                                                                                           |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>time</code>       | The follow-up time for right censored data.                                                                                                                                                                                                                                                                                               |
| <code>status</code>     | The status indicator, 1=event, and 0=right censored.                                                                                                                                                                                                                                                                                      |
| <code>arm</code>        | The group indicator for comparison. The elements of this vector take either 1 or 0. Normally, 0=control group, 1=active treatment group.                                                                                                                                                                                                  |
| <code>tau</code>        | A scaler value to specify the truncation time point for the RMST calculation. <code>tau</code> needs to be smaller than the minimum of the largest observed time in each of the two groups. When <code>tau = NULL</code> , the default value (i.e., the minimum of the largest observed time in each of the two groups) is used.          |
| <code>covariates</code> | This specifies covariates to be used for the adjusted analyses. When <code>NULL</code> , unadjusted analyses are performed. When non <code>NULL</code> , the ANCOVA-type adjusted analyses are performed using those variables passed as <code>covariates</code> . This can be one variable (vector) or more than one variables (matrix). |

# クラス：その他データ関連の話題

- Rのオブジェクト指向プログラミングは複数の方式 (S3, S4, RC) が提供
- 簡単なためS3クラスとS3メソッドがよく使われる
- クラスごとの処理の切り替え
  - 関数f(クラスA)と関数f(クラスB)で違う結果や

# クラスのチェック

- 関数class()

```
library(survival)
data(leukemia)
sv <- Surv(leukemia$time, leukemia$status)
class(leukemia)

## [1] "data.frame"

class(sv)

## [1] "Surv"

print(leukemia)

print.data.frame(leukemia)

##      time status          x
## 1      9      1 Maintained
## 2     13      1 Maintained
...
```

# クラスのチェック

```
print(sv)
```

```
## [1] 9 13 13+ 18 23 28+ 31 34 45+ 48 161+ 5  
5 8  
## [15] 8 12 16+ 23 27 30 33 43 45
```

```
getAnywhere(print.Surv)
```

```
## A single object matching 'print.Surv' was found  
## It was found in the following places  
## registered S3 method for print from namespace survival  
## namespace:survival  
## with value  
##  
## function (x, quote = FALSE, ...)  
## {  
## invisible(print(as.character.Surv(x), quote = quote, ...))  
## }  
## <bytecode: 0x000000001ed525f0>  
## <environment: namespace:survival>
```

# まとめ

- Rによる生存時間解析
- RStudioとR Markdown
- Rの基本的な使用方法
- R言語の学習のポイントとデータ構造
- 基本関数によるデータハンドリング

**複雑なデータ構造の理解＋ヘルプ**

**実際のところプログラミング言語の  
基本を学ぶと全貌がよく見えてくる**