

2025年8月29-31日
遺伝統計学・夏の学校 講義実習資料

統計解析ソフトウェアR入門

東京大学大学院医学系研究科 遺伝情報学
大阪大学大学院医学系研究科 遺伝統計学
理化学研究所生命医科学研究センター システム遺伝学チーム

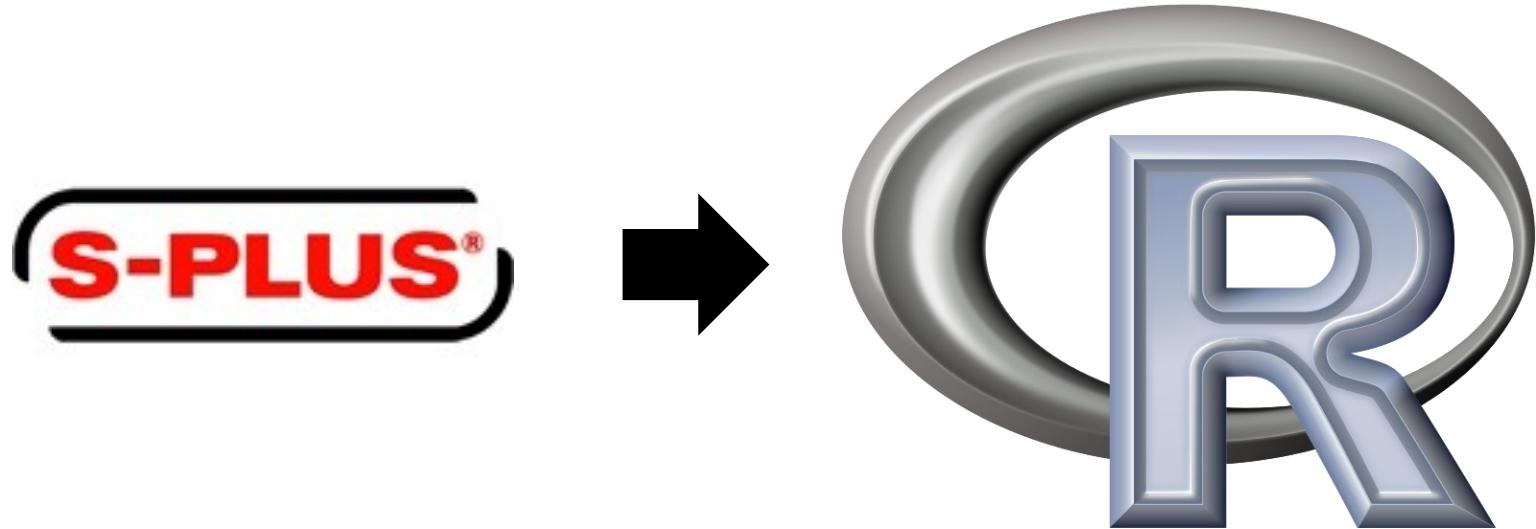
<https://genome.m.u-tokyo.ac.jp/index.html>

統計解析ソフトウェアR入門

- ① 統計解析ソフトウェアRについて
- ② Rのインストール方法
- ③ 数値計算、変数(ベクトル・テーブル)の扱い
- ④ Rの関数
- ⑤ if文とfor文
- ⑥ ファイルの読み書き
- ⑦ 統計検定
- ⑧ グラフの描画

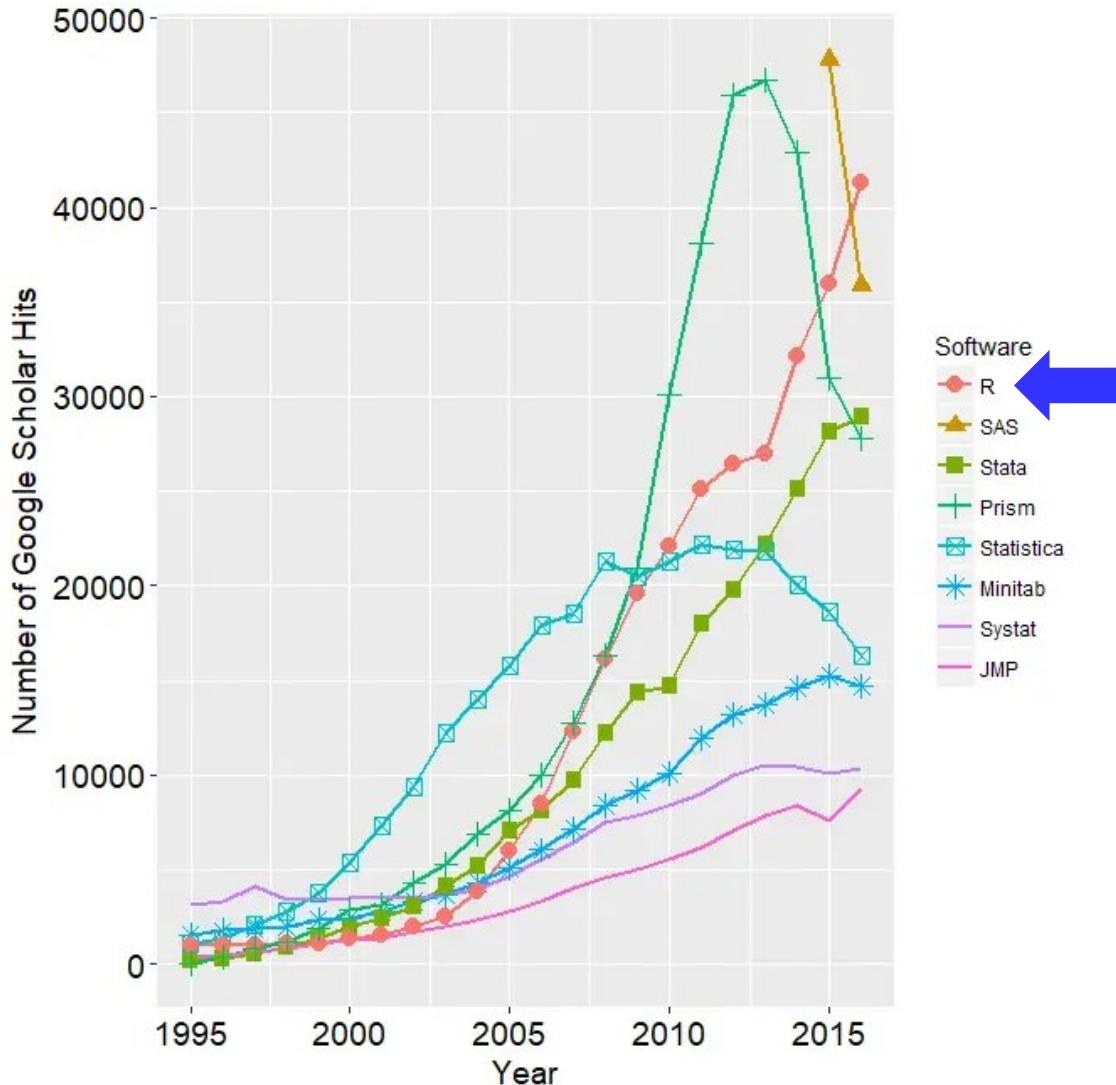
本講義資料は、Windows PC上で
C:¥SummerSchoolにフォルダを配置すること
を想定しています。

① 統計解析ソフトウェアRについて



- ・”R”は、プログラム言語の一種で、統計解析機能を備えています。
- ・元々は、有償の統計プログラム言語S-PLUSから派生しました。
- ・無償のオープンソースのソフトウェアであり、統計解析機能やデータ可視化機能に優れていることから、世界中で広く使用されています。

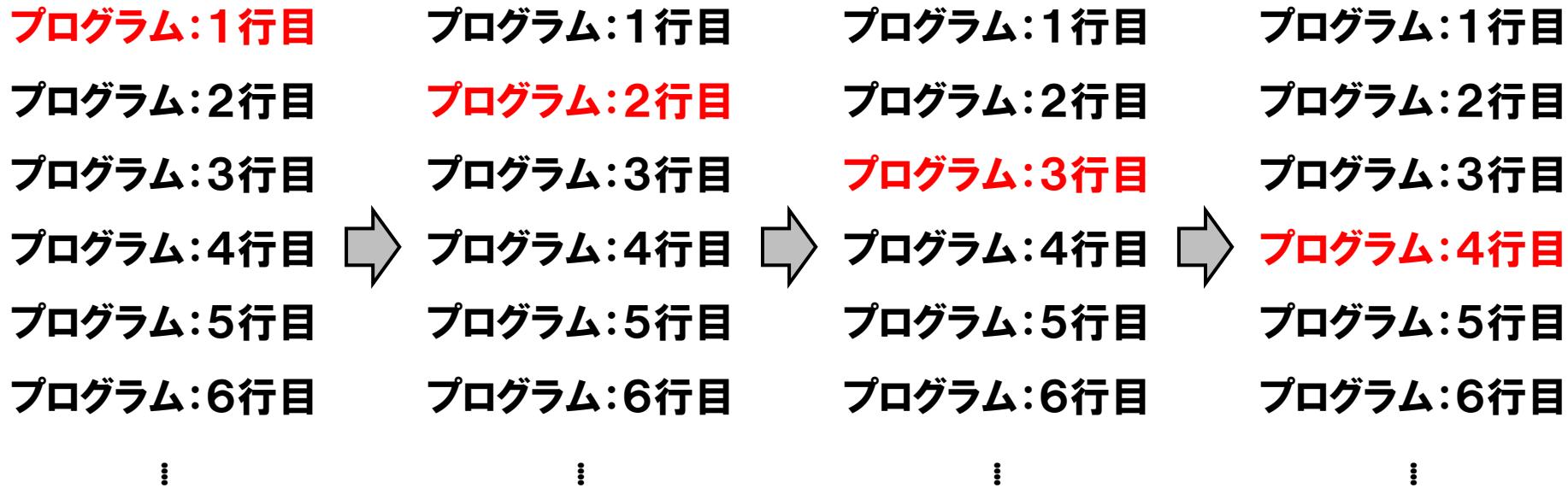
① 統計解析ソフトウェアRについて



- ・RやS-PLUS以外にも、統計解析ソフトウェアは多数存在します。
- ・以前はSASやSPSSが有名でしたが、最近はRにシェアが移っています。

① 統計解析ソフトウェアRについて

インタプリタ型のプログラム言語の実行形式



- ・Rは、ソースコードを一行一行実施していく**インタプリタ型**のプログラム言語に分類され、初步的なプログラム入門言語としても適しています。
- ・最近のバージョンでは、大容量データをプログラム上で扱うメモリ機能が充実し、ゲノムのような**ビッグデータ解析**にも**適用可能**になりました。⁵

① 統計解析ソフトウェアRについて

Google

r解説



すべて

動画

画像

ニュース

ショッピング

もっと見る

ツール

約 799,000,000 件 (0.46 秒)

<https://ai-trend.jp/r-beginner> r-beginner ▾

R言語入門 | AVILEN AI Trend

2020/05/20 — また、Rによる代表的な統計解析を自作関数例を交えて紹介しています。... Rでベクトルと呼ばれるリストの扱い方や、行列の操作の基礎を解説してい...

ベクトル行列操作・Rのインストールの仕方...・プログラムの実行方法・便利なコマンド集

<https://www.amazon.co.jp> R言語徹底解説-Hadley-Wi... ▾

R言語徹底解説 単行本 – 2016/2/10 - アマゾン

AmazonでHadley Wickham, 基広, 石田, 太祐, 市川, 慎一, 高柳, 真太朗, 福島のR言語徹底解説。

アマゾンならポイント還元本が多数。Hadley Wickham, 基広, 石田, 太祐, ...

★★★★★ 評価: 3.7 · 9 件のレビュー · ¥5,940

<https://udemy.benesse.co.jp> データ分析 ▾

R言語とは？統計解析をするならPythonとどちらがおすすめ？

2022/03/04 — 機械学習は、膨大なデータからルールやパターンを発見して、分析する手法を意味します。例えは、これを利用することで「ペットの表情から病気を予測する...

<https://and-engineer.com> YPePSBIAACEAeEkt ▾

R言語入門！概要や基礎、正規分布のグラフの作成方法 ...

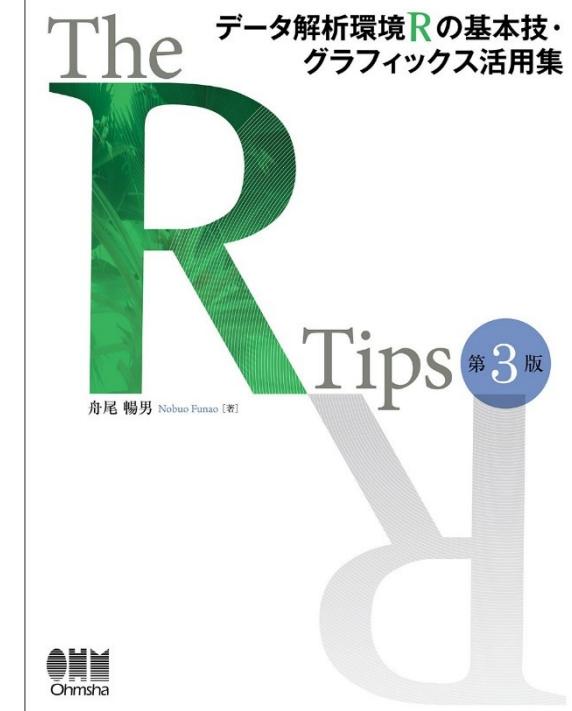
R言語は統計解析に強みをもっているプログラミング言語です。... 記事では、R言語の概要やPythonとの違い、R言語の基礎、正規分布のグラフの作成方法について解説します。

1-1. R言語とは · 1-2. R言語とPythonの違い · 2-1. ベクトル · 2-2. R言語は実数

<https://persol-tech-s.co.jp> hatalabo · it_engineer ▾

R言語とは？主な特徴5つやできること・始め方をわかり ...

2021/09/29 — R言語は統計解析や機械学習に使用される開発言語です。... R言語の特徴や利用用途、具体的なエンジニア職種について解説していきます。



・Rを取り扱ったwebサイトや、入門本も大変充実しています。

① 統計解析ソフトウェアRについて

RStudio is becoming Posit.
Learn more at posit.co

Level up your data science training with RStudio Academy

Customized training An emphasis on skills Practice-based Mentor-led feedback

(<https://www.rstudio.com/>)

・Rはコマンド(命令)をコンソール(画面)に打ち込んで使いますが、マウスクリックを併用するGUI機能等が追加された、“RStudio”といった派生ソフトウェアも充実しています(本講義では説明しません)。

統計解析ソフトウェアR入門

- ① 統計解析ソフトウェアRについて
- ② Rのインストール方法
- ③ 数値計算、変数(ベクトル・テーブル)の扱い
- ④ Rの関数
- ⑤ if文とfor文
- ⑥ ファイルの読み書き
- ⑦ 統計検定
- ⑧ グラフの描画

② Rのインストール方法



[CRAN](#)
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

[About R](#)
[R Homepage](#)
[The R Journal](#)

[Software](#)
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

[Documentation](#)
[Manuals](#)
[FAQ](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows** and **Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)



R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

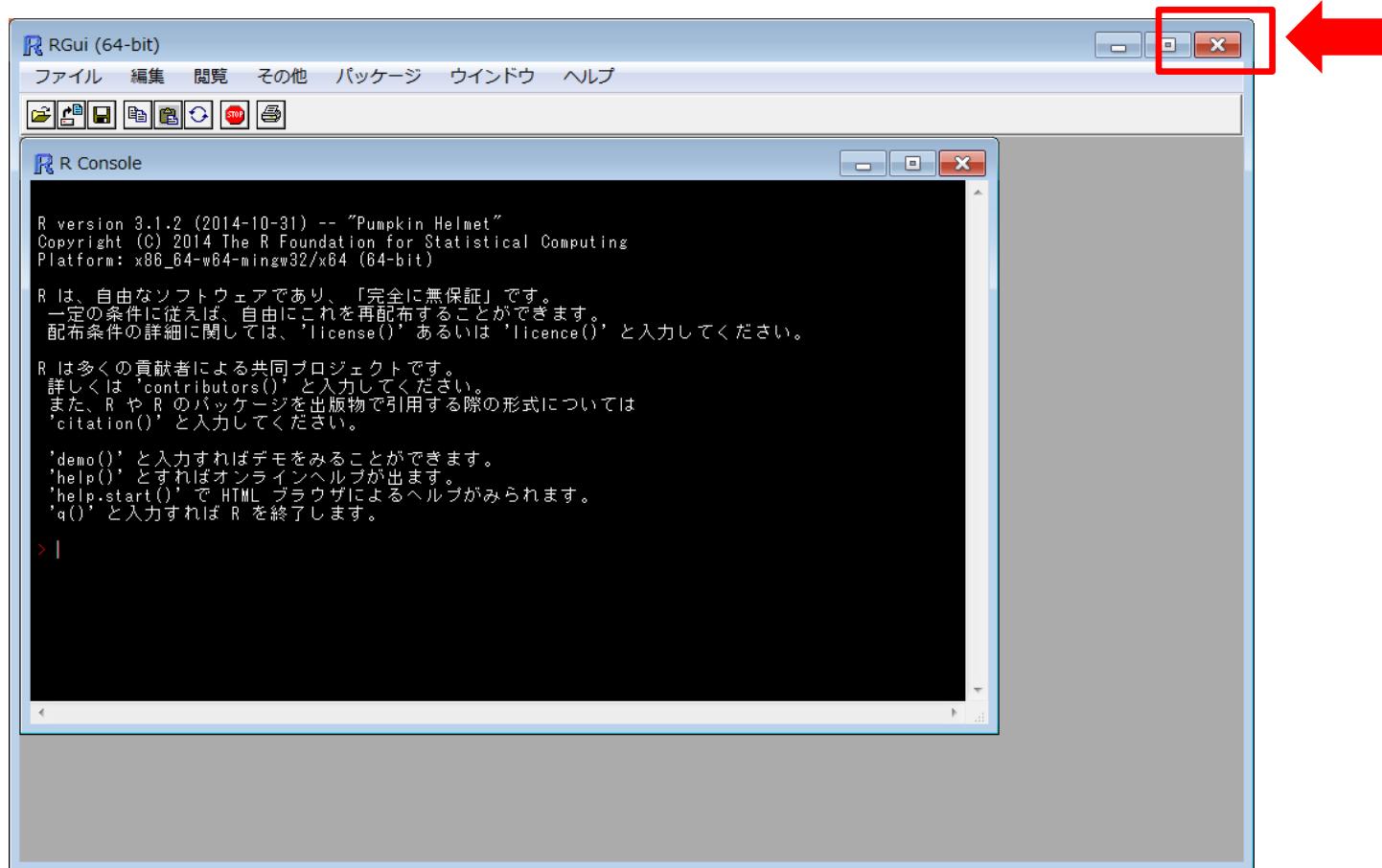
Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (Tuesday 2016-06-21, Bug in Your Hair) [R-3.3.1.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).

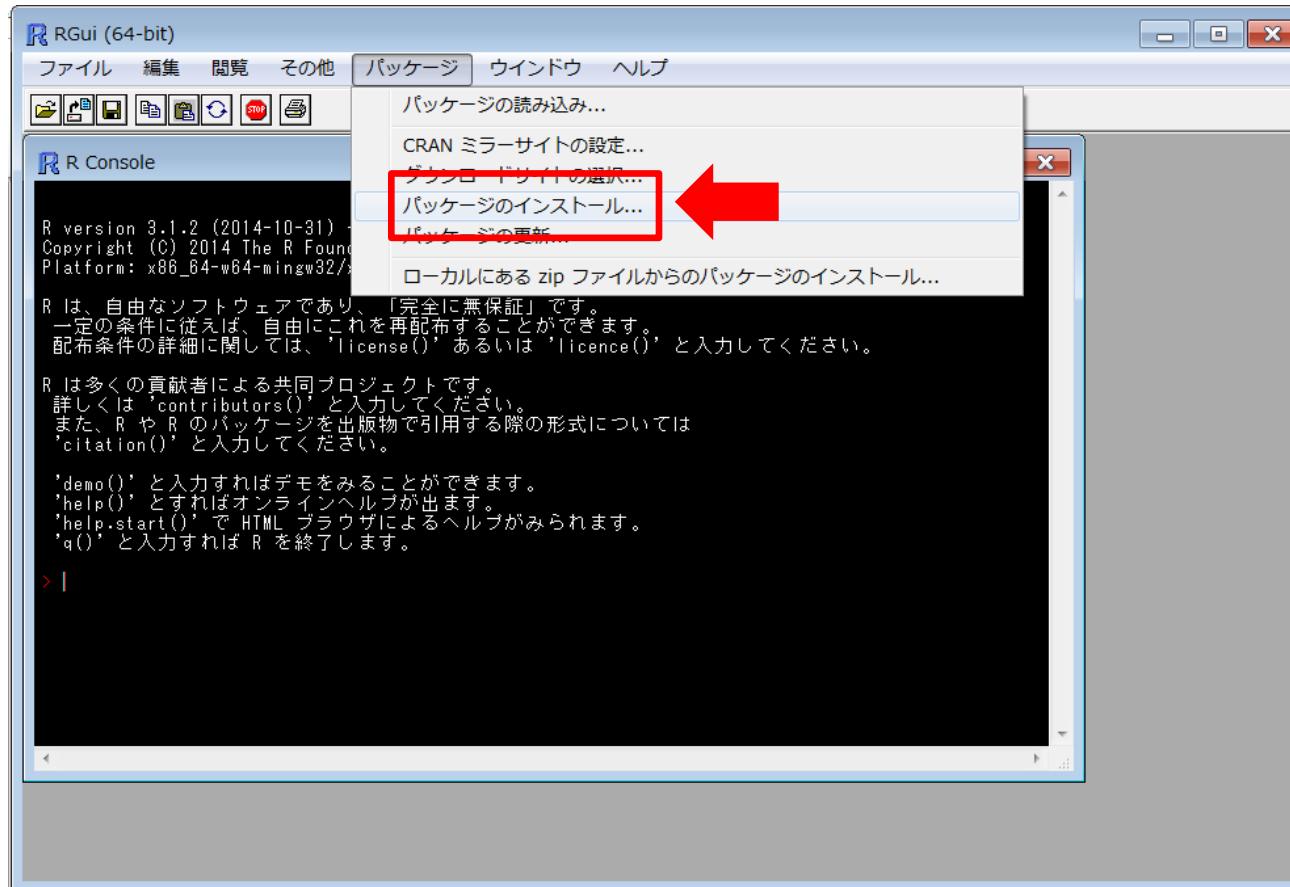
- Rは、CRAN(Comprehensive R Archive Network)のWeb サイト(<http://cran.r-project.org/>)から無償で入手できます。
- OS(Windows/Mac/Linux)やbit数(64/32bit)を選んでダウンロードし、ダブルクリックでインストールできます。

② Rのインストール方法



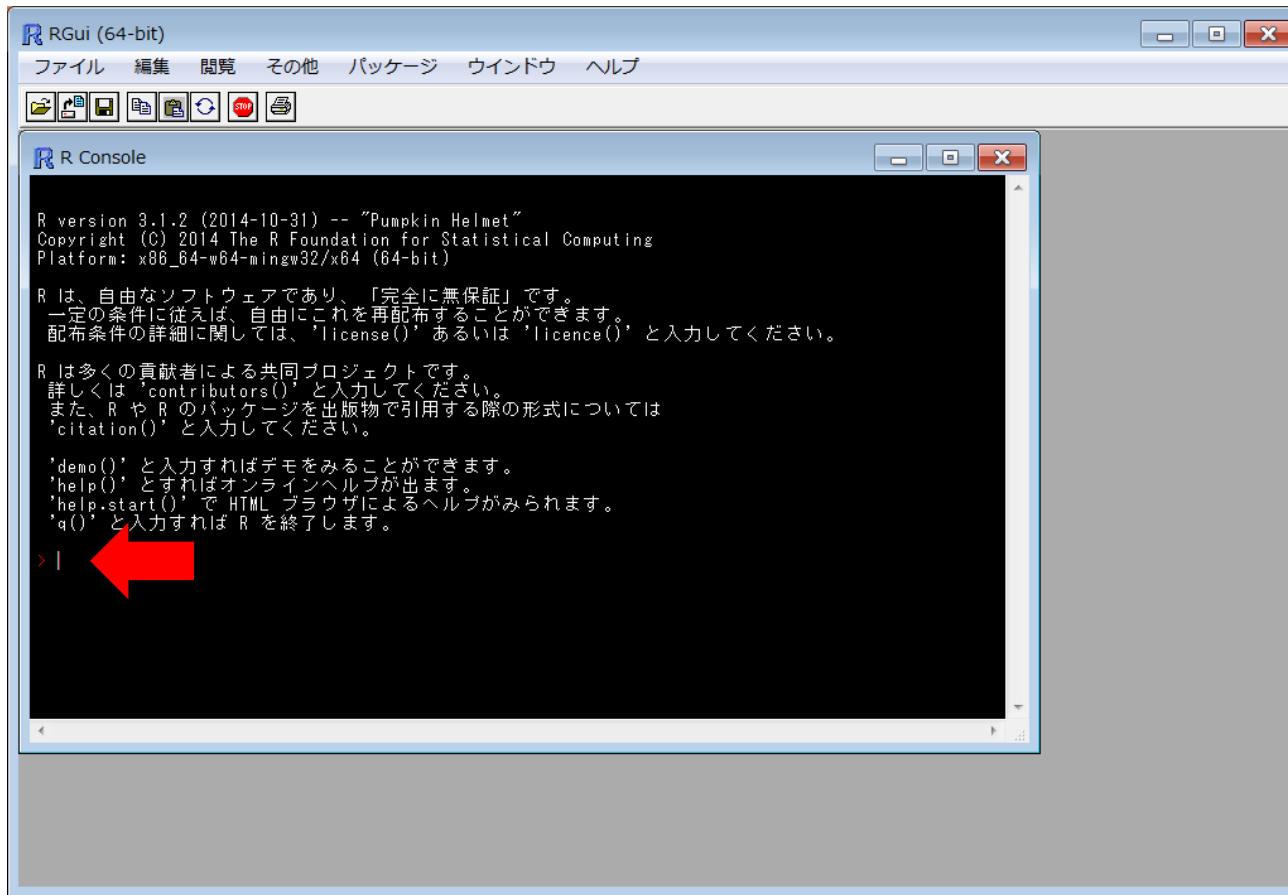
- ・インストール後、画面上のアイコンをクリックすると、Rが起動します。
- ・「ファイル」→「終了」か、右上の×をクリックで終了できます。
- ・終了するとき、「作業スペースを保存しますか？」と聞かれますが、(継続して解析を実施したい場合を除いて)「いいえ」で問題ありません。

② Rのインストール方法



- ・Rはオープンソースなので、最初に実装された機能以外にも、有志で開発されたパッケージ(追加解析機能)が公開されています。
- ・「パッケージ」→「パッケージのインストール」で、ダウンロード用のCRANミラーサイトを選択後、希望パッケージを選択するとインストール可能です。

② Rのインストール方法



・それでは、入力画面に色々な命令を入力して、Rを使ってみましょう。

統計解析ソフトウェアR入門

- ① 統計解析ソフトウェアRについて
- ② Rのインストール方法
- ③ 数値計算、変数(ベクトル・テーブル)の扱い
- ④ Rの関数
- ⑤ if文とfor文
- ⑥ ファイルの読み書き
- ⑦ 統計検定
- ⑧ グラフの描画

③ 数値計算、変数(ベクトル・テーブル)の扱い

```
> 10
[1] 10
> 10+3      ← 足し算
[1] 13
> 10-3      ← 引き算
[1] 7
> 10*3      ← かけ算
[1] 30
> 10**3     ← 累乗
[1] 1000
> 10^3       ← 累乗
[1] 1000
> 10/3       ← わり算
[1] 3.333333
> 10%%3      ← わり算(剰余)
[1] 1
```

・Rの画面上に、数字や数式を打ち込むと、電卓として機能します。

③ 数値計算、変数(ベクトル・テーブル)の扱い

```
> x <- 1
```

```
> x
```

```
[1] 1
```

← “<-”を使うと、変数(x)に値を代入できます。

```
> x = 2
```

← “=”でも代入できます。値は上書きされます。

```
> x
```

```
[1] 2
```

```
> x <- "hensu"
```

← 数値以外の型、文字列(character)も代入できます。

```
> x
```

```
[1] "hensu"
```

```
> x <- TRUE
```

← 数値以外の型、論理値(boolean: True or FALSE)も代入できます。

```
> x
```

```
[1] TRUE
```

・“<-”や“=”を使うと、変数に数値を代入することができます。

③ 数値計算、変数(ベクトル・テーブル)の扱い

```
> x <- c(1,2,3,4,5) ← "c ()" で数値をくくると、1つのベクトルとしてまとめることができます。
```

```
[1] 1 2 3 4 5
```

```
> x+1 ← ベクトルに対しては、数値と同じ感覚で計算ができます。
```

```
[1] 2 3 4 5 6
```

```
> x*2
```

```
[1] 2 4 6 8 10
```

```
> c(x,x) ← 数値だけでなく、変数やベクトルそのものも、"c ()" でくくることができます。変数の型を明示的に定義しなくても使用することができます。
```

```
[1] 1 2 3 4 5 1 2 3 4 5
```

- ・"c ()" で複数の値や変数をベクトルとしてまとめることができます。
- ・ベクトルに対しては、変数と同じ感覚で計算や代入ができます。

③ 数値計算、変数(ベクトル・テーブル)の扱い

> x <- c(6,7,8,9,10) ← ベクトルの直後に”[]”を付けると、ベクトルの内部で、
[] の中に示された順番(場所)の値を取り出すことができます。

> x[c(2,3,4)] ← [] の中の順番は、数値だけでなくベクトルを使って指定することも出来ます。

> 2:4 ← “2:4”というのは、”2から4まで”という意味で、c (2,3,4)と同じになります。

> x[2:4]
[1] 7 8 9

- ・ベクトルを構成する各々の値を、**要素**といいます。
- ・ベクトルの直後に”[]”を付けると、ベクトルの内部で、[] の中に示された順番(場所)の要素を取り出すことができます。

③ 数値計算、変数(ベクトル・テーブル)の扱い

```
> x <- c(6,7,8,9,10)
```

```
> x <= 7
```

```
[1] TRUE TRUE FALSE FALSE FALSE
```

```
> x[x<=7]
```

```
[1] 6 7
```

← []の中には、各順番に対応する論理値(条件式)を入れることもできます。

```
> x[x<=7 | x>=9]
```

```
[1] 6 7 9 10
```

← []の中に入れる条件式は、複数組み合わせることもできます。"|"は"OR"で、"+">が"AND"です。

```
> x[x<=7 & x>=9]
```

```
numeric(0)
```

← 条件に合致する順番が存在しないので、結果が得られませんでした。

- ・ベクトルの要素を取り出す [] には、条件式を入れることも出来ます。
- ・ベクトルの定義や計算が簡単で、色々な要素の扱い方ができるのが、R の特徴です。統計計算ではベクトルを頻繁に使いますので、便利です。

③ 数値計算、変数(ベクトル・テーブル)の扱い

ベクトルの内部構成

順番	1	2	3	4	5
要素	6	7	8	9	10



・Rで扱うベクトルの要素の順番は、1から始まります。

(実際には、Pythonのように、0から始まるプログラムの方が、多数派です。)

③ 数値計算、変数(ベクトル・テーブル)の扱い

↓ `matrix()` という関数を使って、4から9まで並んだ数字のベクトル(4:9)を、縦2行(`nrow=2`)、横3列(`ncol=3`)のテーブル(行列)に、横並び優先(`byrow=T`)で代入しています。

```
> x <- matrix(4:9, nrow=2, ncol=3, byrow=T)
```

```
> x
```

	[,1]	[,2]	[,3]	列の順番を示しています。
[1,]	4	5	6	
[2,]	7	8	9	行列の内容を示しています。
↑				行の順番を示しています。

- 要素が1次元に並んだのがベクトルとすると、2次元に配置されたのがテーブル(行列)です。
- 3次元に並んだ場合はリストと呼ばれます。(本講義では説明しません)

③ 数値計算、変数(ベクトル・テーブル)の扱い

```
> matrix(4:9, nrow=2, ncol=3, byrow=F)
```

```
[,1] [,2] [,3]
```

↑ `matrix()` という関数の条件を変えると、色々なテーブルを作ることができます。

```
[1,] 4 6 8  
[2,] 5 7 9
```

```
> matrix(4:9, nrow=3, ncol=2, byrow=T)
```

```
[,1] [,2]
```

```
[1,] 4 5  
[2,] 6 7  
[3,] 8 9
```

```
> matrix(4:8, nrow=2, ncol=3, byrow=T)
```

```
[,1] [,2] [,3]
```

↓ テーブルに格納する要素が1つ足りなかったので、エラーが出てしました。

```
[1,] 4 5 6  
[2,] 7 8 4
```

警告メッセージ: In `matrix(4:8, nrow = 2, ncol = 3, byrow = T)` :
データ長 [5] が行数 [2] を整数で割った、もしくは掛けた値ではありません

③ 数値計算、変数(ベクトル・テーブル)の扱い

```
> x<-matrix(4:9, nrow=2, ncol=3, byrow=T)
```

```
> x
```

```
 [,1] [,2] [,3]
```

```
[1,] 4 5 6
```

```
[2,] 7 8 9
```

```
> x[1,2] ← 1行目の2列目、というふうに、テーブルの要素にアクセスする際は、行・列それぞれの場所を指定するのが一般的です。
```

```
> x[1,2:3]
```

```
[1] 5 6
```

```
> x[1:2,2:3] ← 複数の行や複数の列を指定することも可能です。
```

```
 [,1] [,2]
```

```
[1,] 5 6
```

```
[2,] 8 9
```

・テーブルの要素も、ベクトルの時と同じく、様々な取り出しができます。

③ 数値計算、変数(ベクトル・テーブル)の扱い

```
> x<-matrix(4:9, nrow=2, ncol=3, byrow=T)
```

```
> x
```

```
 [,1] [,2] [,3]
```

```
[1,] 4 5 6
```

```
[2,] 7 8 9
```

```
> x[1,] ← 行の順番のみ指定した場合、全ての列が取り出し対象になります。
```

```
[1] 4 5 6
```

```
> x[,1] ← 列の順番のみ指定した場合、全ての行が取り出し対象になります。
```

```
[1] 4 7
```

・テーブルの要素も、ベクトルの時と同じく、様々な取り出しができます。

③ 数値計算、変数(ベクトル・テーブル)の扱い

```
> x<-matrix(4:9, nrow=2, ncol=3, byrow=T)
```

```
> x
```

```
 [,1] [,2] [,3]
```

```
[1,] 4 5 6
```

```
[2,] 7 8 9
```

```
> x+1
```

```
 [,1] [,2] [,3]
```

```
[1,] 5 6 7
```

```
[2,] 8 9 10
```

```
> x*2
```

```
 [,1] [,2] [,3]
```

```
[1,] 8 10 12
```

```
[2,] 14 16 18
```

・テーブルの要素も、ベクトルの時と同じく、様々な計算ができます。

統計解析ソフトウェアR入門

- ① 統計解析ソフトウェアRについて
- ② Rのインストール方法
- ③ 数値計算、変数(ベクトル・テーブル)の扱い
- ④ Rの関数
- ⑤ if文とfor文
- ⑥ ファイルの読み書き
- ⑦ 統計検定
- ⑧ グラフの描画

④ Rの関数

```
> x <- c(1,2,3,4,5,6,7,8,9,10)
```

```
> mean(x)
```

```
[1] 5.5
```

```
> sum(x)
```

```
[1] 55
```

```
> var(x)
```

```
[1] 9.166667
```

・ベクトルの構成要素の情報を集計する関数

関数	機能
mean (x)	xの構成要素の平均値を求める。
sum (x)	xの構成要素の和を求める。
var (x)	xの構成要素の分散を求める。
sd (x)	xの構成要素の標準偏差を求める。
max (x)	xの構成要素の最大値を求める。
min (x)	xの構成要素の最小値を求める。
median (x)	xの構成要素の中央値を求める。

- ・Rには、入力値に対して特定の計算を行い結果を出力する、**関数**という機能があります。
- ・基本的な数値計算や統計量計算、ファイル読み書き、グラフ描画など、様々な関数が予め備わっています。

④ Rの関数

> log(10) ← 引数(ひきすう)
[1] 2.302585

> log(10, base=10) ← オプションを指定
[1] 1

・数値計算を行う関数

関数	機能
log (x)	xの自然対数値を求める。オプション "base"で底を指定できる。
log10 (x)	xの対数値を求める(底は10)。
log2 (x)	xの対数値を求める(底は2)。
exp (x)	xの指数を求める。
abs (x)	xの絶対値を求める。
cos (x)	xの三角関数値を求める
sin (x)	xの三角関数値を求める

- ・関数に入力する値を引数(ひきすう)といいます。
- ・各関数に実装されたオプションを指定することで、機能を拡張できます。

④ Rの関数

・ベクトルや行列を取り扱う関数

関数	機能
numeric (x)	0がx個連續したベクトルを作成する。
rep (x, y)	xがy個連續したベクトルを作成する。
seq (x,y,z)	xからyまで、zずつ値が増えていくベクトルを作成する。
unique (x)	xの構成要素から要素の重複を除いたベクトルを作成する。
runif (x)	0-1の間で一様分布に従うx個の乱数で構成されるベクトルを作成する。
rnorm (x)	正規分布に従うx個の乱数で構成されるベクトルを作成する。
matrix (x)	テーブルを作成する。
prcomp (x)	テーブルxの主成分分析を行う。

※引数の詳細は、
一部省略しています。

・文字列を取り扱う関数

関数	機能
paste (x,y)	文字列xとyと結合する。
nchar (x)	文字列xの文字数を求める。
substr (x)	文字列xから特定の文字を切り出す。

・Rには、色々な関数が実装されています。少しずつ覚えていきましょう。

④ Rの関数

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.

Value

If trim is zero (the default), the arithmetic mean of the values in x is computed, as a numeric or complex vector of length one. If x is not logical (coerced to numeric), numeric (including integer) or complex, NA_real_ is returned with a warning.

・入力画面で”?mean”と、関数名に”?”を付けて入力すると、該当関数の説明画面を呼び出すことができます。オプションの確認に便利です。

④ Rの関数

```
> Calc <- function (x, y, z) {  
+ mini <- min(c(x, y, z));  
+ medi <- median(c(x, y, z));  
+ maxi <- max(c(x, y, z));  
+ answer <- mini*medi^maxi;  
+ return (answer);  
+ }  
>  
> Calc(6,2,4);  
[1] 8192  
>  
> 2*4^6  
[1] 8192
```

※ファイル”R_Command_1.txt”を開いて、内容をRの入力画面にコピー & ペーストしてください

引数として入力された3つの値から、
・最小値×中央値^{最大値}
を計算するオリジナル関数”Calc”を定義しました。

・自分のオリジナル関数を定義して、解析に使うことも出来ます。

統計解析ソフトウェアR入門

- ① 統計解析ソフトウェアRについて
- ② Rのインストール方法
- ③ 数値計算、変数(ベクトル・テーブル)の扱い
- ④ Rの関数
- ⑤ if文とfor文
- ⑥ ファイルの読み書き
- ⑦ 統計検定
- ⑧ グラフの描画

⑤ if文とfor文

if (条件式) {実行内容}

for (繰り返し条件) {繰り返し実行内容}

- **if文(条件分岐式)とfor文(繰り返し実行命令)**は、プログラムを実行させる際に重要な機能です。

⑤ if文とfor文

○:Rでのif文

```
if (i>=10 & i<=20) {  
  print(i);  
}
```

※変数iの値が10以上かつ20以下の
場合に、表示する条件式です。

○:Javaでのif文

```
if (i>=10 && i<=20) {  
  System.out.println(i);  
}
```

○:Pythonでのif文

```
if (i>=10 and i<=20):  
  print(i)
```

・**if文**は様々なプログラムに実装されていて、Rでも(やや簡単に)実行可能
です。

⑤ if文とfor文

○:Rでのfor文

```
for (i in 1:100) {  
  print(i);  
}
```

※変数iの値を1つずつ増やしながら表示する動作を、100回繰り返しています。

○:Javaでのfor文

```
for (int i=0;i<100;i++) {  
  System.out.println(i+1);  
}
```

○:Pythonでのfor文

```
for i in xrange(100):  
  print(i)
```

・**for文**は様々なプログラムに実装されていて、Rでも(やや簡単に)実行可能ですが、ただし、あまり速くありません。

⑤ if文とfor文

```
> for (i in 1:100) {  
+ if (i>=10 & i<=15 | i%%17==0) {  
+ print(i);  
+ }  
+ }  
[1] 10
```

※ファイル”R_Command_2.txt”を開いて、内容をRの入力画面にコピー & ペーストしてください

一般的に、ANDとORでは、ANDが優先されます。

```
[1] 11  
[1] 12  
[1] 13  
[1] 14  
[1] 15  
[1] 17  
[1] 34  
[1] 51  
[1] 68  
[1] 85
```

演算子	意味
<code>==</code>	等しい
<code>!=</code>	等しくない
<code><</code>	～より小さい
<code>></code>	～より大きい
<code><=</code>	～以上
<code>>=</code>	～以下
<code>&</code>	AND
<code> </code>	OR

・if文とfor文を組み合わせて、プログラムを実行してみましょう。

⑤ if文とfor文

```
> x <- 1:10;  
>  
> for (i in 1:10) {  
+ print(x[i]^2)  
+ }  
[1] 2  
[1] 4  
:  
[1] 18  
[1] 20  
>  
> print(x^2);  
[1] 2 4 6 8 10 12 14 16 18 20  
>
```

実質的に同じ結果が得られます。

・他のプログラムと比較して、Rのfor文実行速度は遅めですが、ベクトルを有効に使うと、for文を使わないプログラムにすることも可能です。³⁶

⑤ if文とfor文

関数	用途	入力	出力
apply	配列や行列の行・列ごとの集計	配列/行列	ベクトル/配列
lapply	リスト要素ごとに関数を適用	リスト/ベクトル	リスト
sapply	lapply のラッパー関数(出力をベクトルや行列)	リスト/ベクトル	ベクトル/行列
vapply	sapply の型安全版(出力型を事前指定)	リスト/ベクトル	指定した型
mapply	複数の引数を並列に関数へ適用	複数ベクトル等	ベクトル/配列/リスト
tapply	ベクトルを因子でグループ化して集計	ベクトル + 因子	ベクトル/配列/リスト

- ・Rでは、**for文を使わずに行列・リストデータの一括処理**を行う関数として、**apply, tapply, lapply**といった、**apply系関数**が実装されています。
- ・使いこなすが難しいですが便利な関数なので、是非調べてみてください。

統計解析ソフトウェアR入門

- ① 統計解析ソフトウェアRについて
- ② Rのインストール方法
- ③ 数値計算、変数(ベクトル・テーブル)の扱い
- ④ Rの関数
- ⑤ if文とfor文
- ⑥ ファイルの読み書き
- ⑦ 統計検定
- ⑧ グラフの描画

⑥ ファイルの読み書き

入力データ



プログラム

```
#!/bin/sh
set -ef

if test -n "$KSH_VERSION"; then
    puts() {
        print -r -- "$*"
    }
    else
    puts() {
        printf '%s\n' "$*"
    }
fi

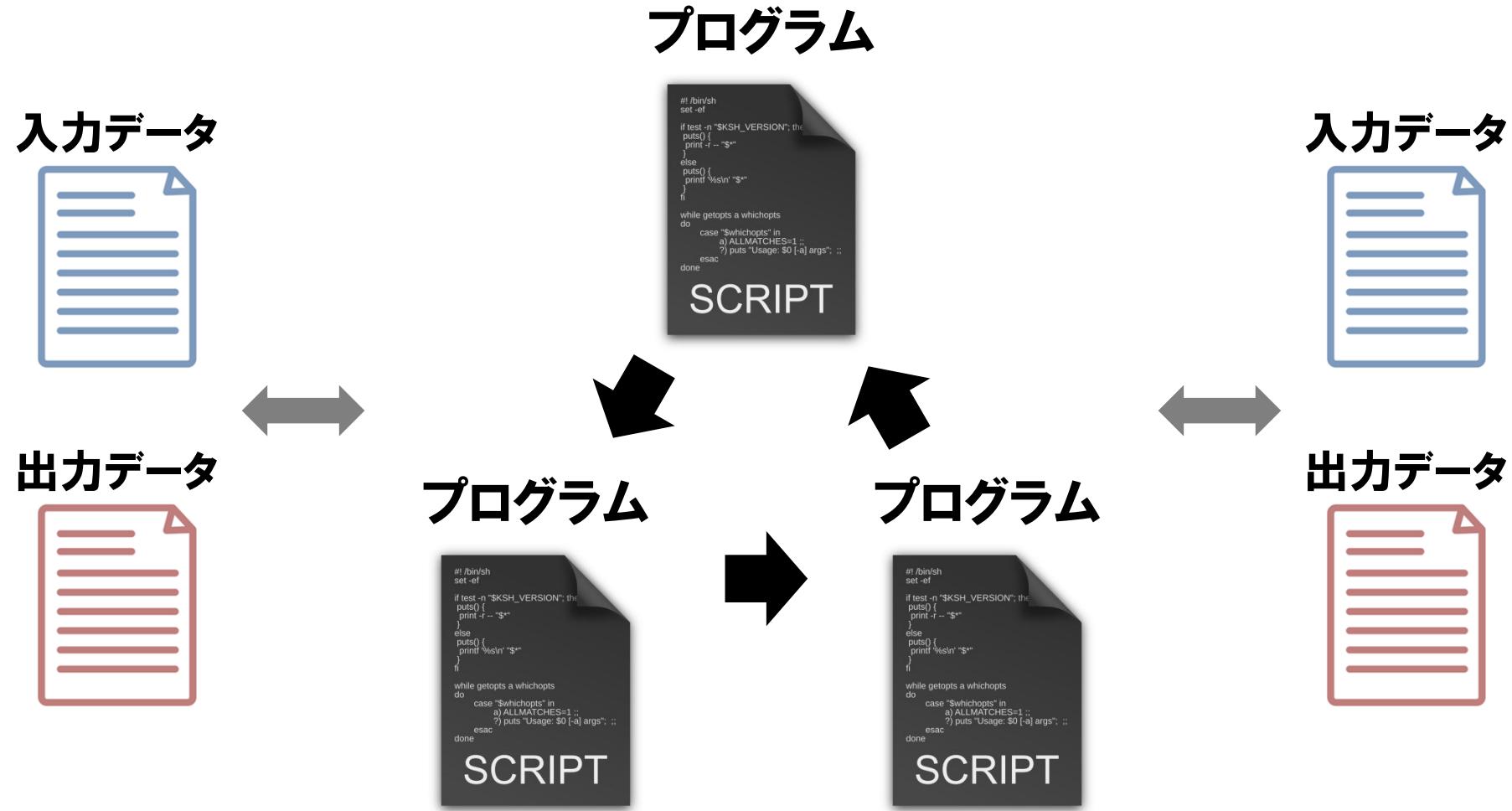
while getopts a: whichopts
do
    case "$whichopts" in
        a) ALLMATCHES=1 ;;
        ?) puts "Usage: $0 [-a] args"; ;;
    esac
done
```

出力データ



- ・データ解析のプログラムは、データの入力→解析→出力と、一方方向に進むことが多いです。
- ・データ解析のプログラムにおいて、ファイルに書き込まれたデータの読み込みと、計算結果のファイルへの書き出しは、重要な機能です。

⑥ ファイルの読み書き



・一方で、市販ソフトウェア等、一般的なソフトウェアでは、プログラムが
継続的に実施されている途中に、副次的にデータ入出力が発生します。
(本講義では説明しません)

⑥ ファイルの読み書き



Diagram illustrating the data flow: Input Data (document icon) → Program (SCRIPT) → Output Data (document icon).

Screenshot of a text editor showing a tab-delimited file "GeneExpression_example.txt". The file contains gene expression data with 11 rows and 10 columns. A red box highlights the tab character between columns 4 and 5, and a red arrow points from this box to the word "タブ(“\t”)" in the explanatory text below.

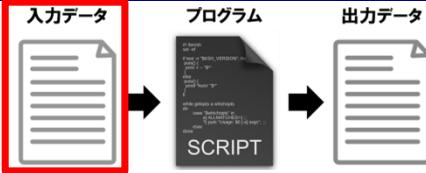
	0	10	20	30	40	50	60	70	80	
1	Gene	> HG01334	> HG00375	> NA12775	> NA20802	> NA12815	> HG00272	> NA20759	> HG00328	> HG00128
2	RP11-1094M14.9	> 10.5745390920804	> 4.62031160828663	> 5.53753871111284	> 5.23187					
3	RP11-16E12.2	> 216.499774043119	> 580.619158774687	> 1001.1869989692	> 383.6705288					
4	ATR	> 2449.95374122831	> 1992.89440704097	> 2027.84667600952	> 1445.74031089282					
5	RP11-181G12.2	> 7.79176564679607	> 5.39036354300107	> 7.1988003244467	> 7.84780627					
6	TCOF1	> 2757.72848427675	> 1940.53087548038	> 2135.27492700511	> 2721.44481923794					
7	NSRP1	> 648.386212751245	> 813.174843058447	> 861.087269578047	> 695.838822733698					
8	AF001548.5	> 50.0899220151176	> 8.47057128185882	> 7.75255419555798	> 14.82363406					
9	OPA1	> 2790.00865624205	> 2032.1670557114	> 2124.19984958289	> 1834.64271056604					
10	RP1-68018.2	> 3.33932813434117	> 6.93046741242995	> 6.64504645333541	> 1.743956949					
11	RHER	> 71	> 616220749011	> 608	> 341028424406	> 575	> 350272084625	> 682	> 759145614644	

※ファイル"GeneExpression_example.txt"

タブ(“\t”)で区切られています。 を、テキストエディタで開いています。

- ・読み込むデータが決まった形式でファイルに書かれていることが重要。
- ・データを、**区切り文字**(タブ、スペース、コンマ)で仕切られた形で、**テキスト形式**(アスキーコード)を使って作成するとわかりやすいです。

⑥ ファイルの読み書き



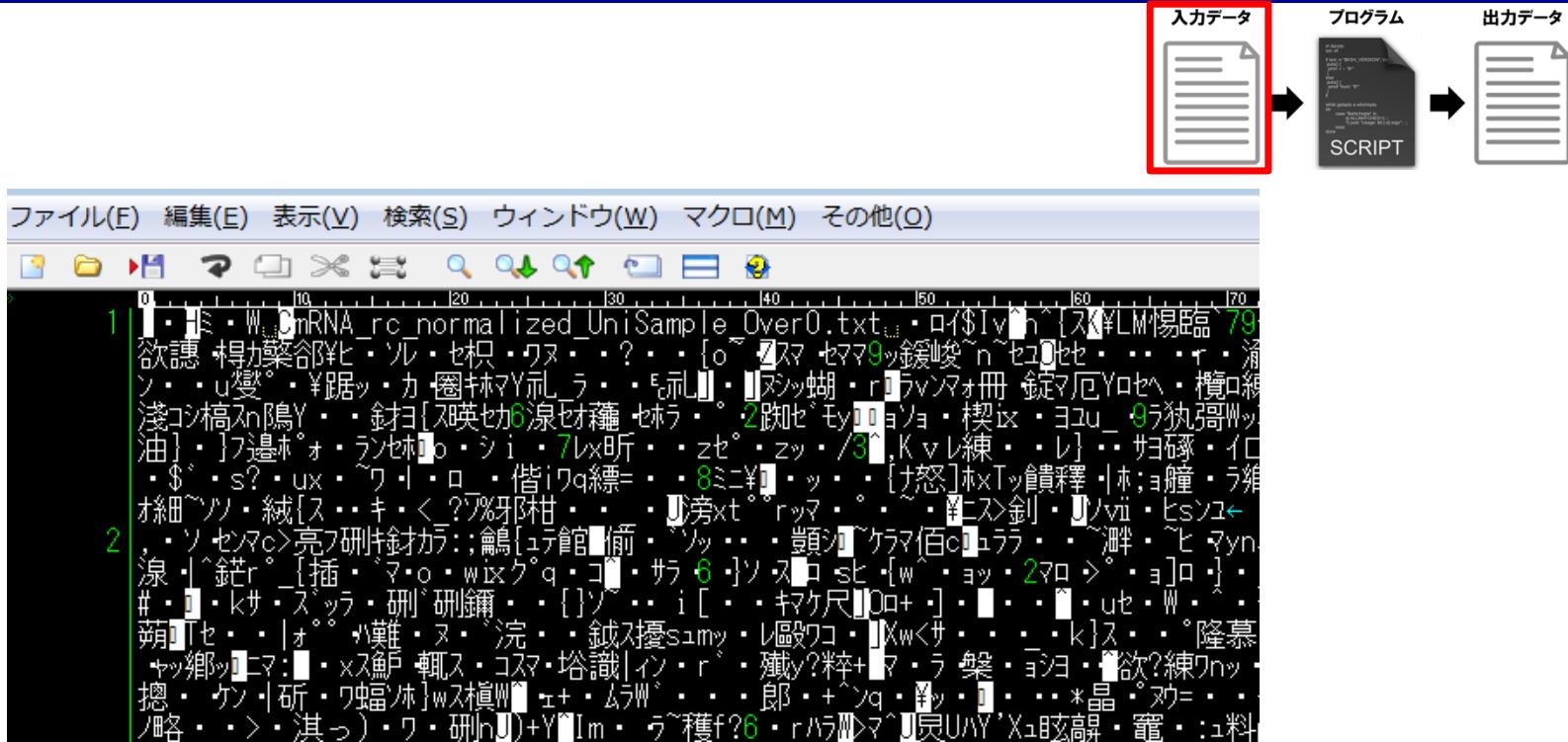
The screenshot shows a Microsoft Excel spreadsheet with data from a text file. The data consists of 11 rows of gene expression values. The first column is labeled 'Gene' and the second column is labeled 'Gene'. The data is as follows:

	Gene	HG01334	HG00375	NA12775	NA20802	NA12815	HG00272	NA20759	HG00328	HG00128
1	RP11-1094M14.9	10.574539	4.6203116	5.5375387	5.2318708	2.1915785	4.9575508	4.9049613	4.4309949	4.943907
2	RP11-16E12.2	216.49977	580.61916	1001.187	383.67053	1016.8924	392.47277	433.59858	463.98847	384.63596
3	ATRX	2449.9537	1992.8944	2027.8467	1445.7403	1373.3892	1878.9117	2481.9104	1986.9847	1463.3965
4	RP11-181G12.2	7.7917656	5.3903635	7.1988003	7.8478063	2.1915785	15.698911	4.9049613	5.0639942	0
5	TCOF1	2757.7285	1940.5309	2135.2749	2721.4448	2691.2584	2247.423	2679.0899	2871.2847	2696.4069
6	NSRP1	648.38621	813.17484	861.08727	695.83882	412.74728	846.91492	1058.4906	803.90907	712.91139
7	AF001548.5	50.089922	8.4705713	7.7525542	14.823634	108.11787	30.571563	7.8479381	26.585969	50.427851
8	OPA1	2790.0087	2032.1671	2124.1998	1834.6427	1583.7807	2303.6086	2785.037	2322.4743	2047.7663
9	RP1-68D18.2	3.3393281	6.9304674	6.6450465	1.7439569	15.341049	9.9151015	4.9049613	3.7979956	6.9214698
10	RHEB	714.61622	608.34103	575.35027	682.75915	617.29461	699.01466	728.87725	590.58832	575.47078
11										

※ファイル"GeneExpression_example.txt"のデータを、エクセルに貼り付けています。

・区切り文字(タブ、スペース、コンマ)で仕切られたテキストファイルは、エクセル等の表計算ソフトに貼り付けると、内容の確認が行いやすいです。

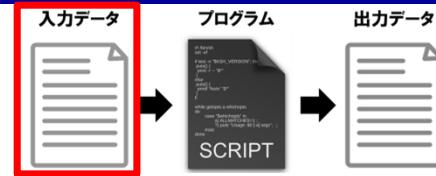
⑥ ファイルの読み書き



※バイナリ形式のファイルを開いても、人間には読めません。

- ・テキストファイル形式は便利ですが、ファイルの容量が大きくなります。
 - ・コンピューターにしか読めない**バイナリ形式**でファイルを作成すると、ファイル容量を小さくし、ハードディスクを節約できます。
 - ・Rでは、**gz形式**で圧縮されたファイルも読み込むことができます。

⑥ ファイルの読み書き



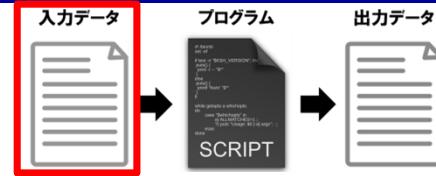
・ファイルの読み書きを取り扱う関数

関数	機能
setwd (x)	作業ディレクトリをxに変更する。
getwd ()	現在の作業ディレクトリを取得する。
scan (x)	テキストファイルxの内容をベクトルとして読み込む。
read.table (x)	テキストファイルxの内容をテーブルとして読み込む。
write (x)	テキストファイルxにデータを書き込む。
write.table (x)	テキストファイルxにテーブル形式で書き込む。

※引数の詳細は、
一部省略しています。

- ・Rには、**行列形式のデータを読みこむ関数**が実装されています。
- ・予め、**ファイルが置かれたディレクトリ**(コンピューター上の場所)を指定しておく必要があります。

⑥ ファイルの読み書き



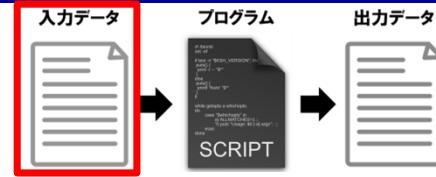
```
> setwd("C:/SummerSchool/Rsoftware");
> Data <- read.table(file="GeneExpression_example.txt", header=T, sep="\t");
> Data;
      Gene   HG01334   HG00375   NA12775
1 RP11-1094M14.9  10.574539  4.620312  5.537539
2 RP11-16E12.2  216.499774  580.619159 1001.186999
3 ATRX  2449.953741 1992.894407 2027.846676
4 RP11-181G12.2  7.791766  5.390364  7.198800
5 TCOF1 2757.728484 1940.530875 2135.274927
```

(以下省略)

※ファイル”R_Command_3.txt”を開いて、内容をRの入力画面にコピー＆ペーストしてください

・ディレクトリ”C:\SummerSchool\Rsoftware”に置かれたファイル”GeneExpression_example.txt”を読み込んでみましょう。

⑥ ファイルの読み書き



```
setwd("C:/SummerSchool/Rsoftware");
```

作業ディレクトリの指定

(Windows上のディレクトリ区切り文字は"¥"ですが、R上での区切り文字は"/"です。)

(WSLでRを動かしている場合、`setwd (" /mnt/c/SummerSchool/Rsoftware/ ")`;としてください)

```
Data <- read.table(file="GeneExpression_example.txt", header=T, sep="¥t");
```

ファイル名の指定

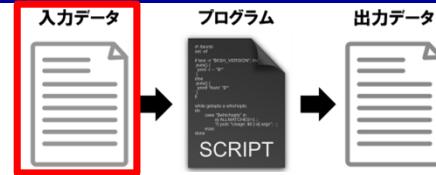
↑
区切り文字の指定
(タブ、"¥t")

第一行目はデータではなく、ヘッダー
(列の名前)として読み込むことを指定

- ・ディレクトリ"C:¥SummerSchool¥Rsoftware"に置かれたファイル"GeneExpression_example.txt"を読み込んでみましょう。

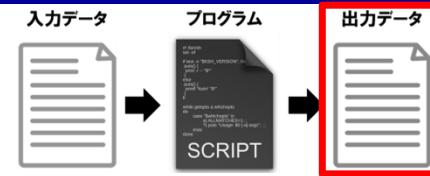
⑥ ファイルの読み書き

```
> Data[1,1];  
[1] RP11-1094M14.9  
> Data[,2];  
[1] 10.574539 216.499774 2449.953741 (以下省略)  
> sum(Data[,2]);  
[1] 9648.989  
> median(Data[,2]);  
[1] 432.443  
> mean(Data[,2]);  
[1] 964.8989  
> sd(Data[,2]);  
[1] 1204.705  
> var(Data[,2]);  
[1] 1451315
```



・読み込んだデータは、通常のテーブルとして要素の取り出しや関数を使った計算ができます。

⑥ ファイルの読み書き



書き出すデータの指定

```
> write.table(Data[,1:5], file="GeneExpression_example_out.txt",  
sep="\t", quote=F, row.names=F, col.names=T);
```

書き出すファイルの指定

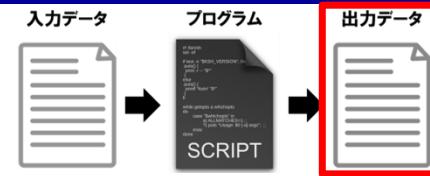
細かいオプションの指定
(本講義では説明しません)

区切り文字の指定
(タブ、"\t")

※読み込んだデータの最初の5列を、ファイル
"GeneExpression_example_out.txt"として書き出しています。

•Rには、変数をファイルに書き出す関数が実装されています。

⑥ ファイルの読み書き



ファイル(E) 編集(E) 表示(V) 検索(S) ウィンドウ(W) マクロ(M) その他(O)

□ フォルダ □ ファイル □ リスト □ フィルタ □ 検索 □ ダウンロード □ アップロード □ フォルダ内 □ ファイル内 □ フィルタ内 □ フィルタ外

1	Gene	HG01334	HG00375	NA12775	NA20802
2	RP11-1094M14.9	10.5745390920804	4.62031160828663	5.53753871111284	5.2318
3	RP11-16E12.2	216.499774043119	580.619158774687	1001.1869989692	383.670521
4	ATRX	2449.95374122831	1992.89440704097	2027.84667600952	1445.7403108928
5	RP11-181G12.2	7.79176564679607	5.39036354300107	7.1988003244467	7.84780621
6	TCOF1	2757.72848427675	1940.53087548038	2135.27492700511	2721.4448192379
7	NSRP1	648.386212751245	813.174843058447	861.087269578047	695.838822733698
8	AF001548.5	50.0899220151176	8.47057128185882	7.75255419555798	14.8236340
9	OPA1	2790.00865624205	2032.1670557114	2124.19984958289	1834.64271056604
10	RP1-68D18.2	3.33932813434117	6.93046741242995	6.64504645333541	1.7439569
11	RHEB	714.616220749011	608.341028424406	575.350272084625	682.75914561464

※読み込んだデータの最初の5列を、ファイル”GeneExpression_example_out.txt”として書き出しています。

・データがファイルに正しく書き込まれたか、確認してみましょう。

統計解析ソフトウェアR入門

- ① 統計解析ソフトウェアRについて
- ② Rのインストール方法
- ③ 数値計算、変数(ベクトル・テーブル)の扱い
- ④ Rの関数
- ⑤ if文とfor文
- ⑥ ファイルの読み書き
- ⑦ 統計検定
- ⑧ グラフの描画

⑦ 統計検定

・統計検定に関する関数

関数	機能
<code>t.test (x)</code>	1標本(x)のt検定
<code>t.test (x, y)</code>	2標本(x,y)のt検定
<code>var.test (x, y)</code>	2標本(x,y)の等分散の検定
<code>cor.test (x, y)</code>	2標本(x,y)の相関関係の検定
<code>lm (x~y)</code>	従属変数(x)に対する独立変数(y)の単回帰
<code>lm (x~y+z)</code>	従属変数(x)に対する独立変数(y,z)の重回帰
<code>glm (x~y, family="binomial")</code>	従属変数(x)に対する独立変数(y)のロジスティック回帰
<code>chisq.test (x, df=y)</code>	テーブル(x)に対する自由度(y)のカイ二乗検定
<code>fisher.test (x)</code>	テーブル(x)に対するフィッシャー正確確率検定
<code>qchisq (x, df=y)</code>	P値(x)に対する自由度(y)のカイ二乗値
<code>pchisq (x, df=y)</code>	カイ二乗値(x)に対する自由度(y)のP値

- ・Rには、解析に必要な多数の統計検定が関数として実装されています。
- ・研究論文においても、有償の統計ソフトではなく、無償のRに実装された統計検定を使うことが(概ね)認められています。

⑦ 統計検定



とりあえずRの統計検定関数を実行したら、こんな結果が出たのですが、あってますか？



Rの関数を実行することと、統計解析を行うことは、全然違う話です。

- ・Rの統計検定関数を実行することと、正しく統計解析を行うことは、別の次元の話であることに、注意して下さい。
- ・正しい統計解析には、正しい統計の知識に基づき、適切な解析データの取得、帰無仮説と検定方法の選択、検定結果の解釈が必要です。



Rの関数の実行結果は、信じても大丈夫ですか？



Rはフリーソフトで、動作の正確性の100%保証はされていません。SAS等の有償ソフトでの確認も重要です。

- ・Rは無償のフリーソフトウェアで、多数の人により広く開発されています。
- ・しかし、統計検定の関数が絶対に正確であるとは保証されていません。
- ・使用頻度の低い関数や統計検定を用いる場合、他の有償ソフトウェアでのダブルチェックなど、正しい結果を追求する姿勢は重要です。

⑦ 統計検定



何とかして有意な結果(0.05以下のP値)を出したいんですけど…



目的と結果が逆になっていますよ！！

- Rの統計検定関数を実行する前に、研究デザインに沿った帰無仮説と検定の種類を選択してください。
- 解析結果を見てから帰無仮説や検定の種類を選び直すのは×です。

⑦ 統計検定

・t検定に関するオプション例

関数	機能
<code>t.test (x, y)</code>	対応のない2標本(x,y)のt検定
<code>t.test (x, y, paired=T)</code>	対応のある2標本(x,y)のt検定
<code>t.test (x, y, paired=F, var.equal=T)</code>	対応のない2標本(x,y)のt検定を等分散を仮定して実行

・相関係数に関するオプション例

関数	機能
<code>cor.test (x, y, method="pearson")</code>	2標本(x,y)の相関関係の検定 (ピアソンの相関係数)
<code>cor.test (x, y, method="spearman")</code>	2標本(x,y)の相関関係の検定 (スピアマンの順位相関係数)

- ・Rの統計検定関数は、オプションによって異なる検定を行います。
- ・Rの統計検定関数を正しく使いこなすには、**オプションを適切に指定する必要があります。**

⑦ 統計検定

```
> x <- c(0.61, 0.65, 0.20, 0.60, 0.30, 0.06, 0.00, 0.34, 0.69, 0.91, 1.00, 0.10, 0.63,  
0.72, 0.83, 0.02, 0.60, 0.40, 0.19, 0.91);  
> y <- c(0.79, 0.71, 0.94, 0.79, 0.98, 0.13, 0.94, 0.64, 0.74, 0.86, 0.70, 1.05, 0.87,  
0.43, 0.89, 0.93, 0.72, 0.77, 0.56, 0.24);  
> t.test(x, y);
```

Welch Two Sample t-test

data: x and y

t = -2.7521, df = 35.228, p-value = 0.009295

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-0.42742282 -0.06457718

sample estimates:

mean of x mean of y

0.488 0.734

※ファイル”R_Command_4.txt”を開いて、内容をRの入力画面にコピー＆ペーストしてください

・t検定を実施してみましょう。

```
> t.test(x, y, paired=T);
```

Paired t-test

data: x and y

t = -2.5422, df = 19, p-value = 0.01989

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-0.44853612 -0.04346388

sample estimates:

mean of the differences

-0.246

・t検定を実施してみましょう。

```
> t.test(x, y, paired=F, var.equal=T);
```

Two Sample t-test

data: x and y

t = -2.7521, df = 38, p-value = 0.009026

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-0.42695412 -0.06504588

sample estimates:

mean of x mean of y

0.488 0.734

・t検定を実施してみましょう。

⑦ 統計検定

```
> x <- c(0.45, 0.12, 0.89, 0.41, 0.38, 0.14, 0.48, 0.32, 0.47, 0.51);  
> y <- c(0.52, 0.15, 1.05, 0.17, 0.21, 0.05, 0.71, 0.19, 0.70, 0.68);  
>  
> cor.test(x, y, method="pearson");
```

Pearson's product-moment correlation

data: x and y

t = 5.4643, df = 8, **p-value = 0.0005985**

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.5862713 0.9734135

sample estimates:

cor

0.8880793

※ファイル”R_Command_5.txt”を開いて、内容をRの入力画面にコピー＆ペーストしてください

・相関関係の検定を実施してみましょう。

⑦ 統計検定

```
> cor.test(x, y, method="spearman");
```

Spearman's rank correlation rho

data: x and y

S = 14, **p-value = 0.0004667**

alternative hypothesis: true rho is not equal to 0

sample estimates:

rho

0.9151515

・相関関係の検定を実施してみましょう。

⑦ 統計検定

```
> x <- c(0.30, 0.57, 0.68, 0.60, 0.85, 0.77, 0.42, 0.60, 0.40, 0.54, 0.80, 0.71, 0.74, 0.43, 0.52, 0.75, 0.70, 0.86, 0.47, 0.38);  
> y <- c(0.28, 0.77, 0.46, 0.36, 0.84, 0.84, 0.29, 0.36, 0.16, 0.73, 1.03, 0.85, 0.57, 0.44, 0.23, 0.54, 0.60, 0.89, 0.75, 0.34);  
> z <- c(0.15, 0.89, 0.67, 0.52, 0.85, 1.10, 0.15, 0.57, 0.18, 0.63, 0.96, 0.65, 0.59, 0.50, 0.30, 0.76, 0.38, 0.71, 0.93, 0.16);  
> x2 <- c(0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1);  
> summary(lm(x~y));
```

Call:

```
lm(formula = x ~ y)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.22079	-0.11576	0.01177	0.09530	0.15796

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.33811	0.06639	5.092	7.6e-05 ***
y	0.47025	0.10720	4.387	0.000356 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.12 on 18 degrees of freedom

Multiple R-squared: 0.5167, Adjusted R-squared: 0.4899

F-statistic: 19.24 on 1 and 18 DF, p-value: 0.0003557

・回帰分析を実施してみましょう。

※ファイル”R_Command_6.txt”を開いて、内容をRの入力画面にコピー＆ペーストしてください

⑦ 統計検定

```
> summary(lm(x~y+z));
```

Call:

```
lm(formula = x ~ y + z)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.252348	-0.090598	-0.000703	0.097370	0.135834

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.32971	0.06667	4.945	0.000123 ***
y	0.29834	0.19508	1.529	0.144581
z	0.18159	0.17240	1.053	0.306942

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1197 on 17 degrees of freedom

Multiple R-squared: 0.5463, Adjusted R-squared: 0.4929

F-statistic: 10.24 on 2 and 17 DF, p-value: 0.001209

・回帰分析を実施してみましょう。

⑦ 統計検定

```
> summary(glm(x2~y, family="binomial"));
```

Call:

```
glm(formula = x2 ~ y, family = "binomial")
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.40410	-1.03603	-0.01962	1.15622	1.45668

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.072	1.154	-0.929	0.353
y	1.893	1.873	1.011	0.312

(Dispersion parameter for binomial family taken to be 1)

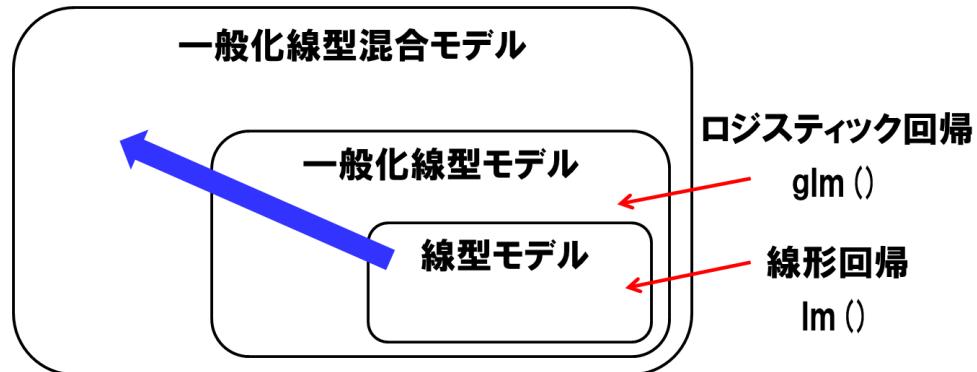
Null deviance: 27.726 on 19 degrees of freedom

Residual deviance: 26.655 on 18 degrees of freedom

AIC: 30.655

Number of Fisher Scoring iterations: 4

・回帰分析を実施してみましょう。



⑦ 統計検定

```
> x <- matrix(c(200,150,170,180), ncol=2);  
>  
> x;  
[1] [2]  
[1,] 200 170  
[2,] 150 180  
>  
> chisq.test(x, correct=F);
```

※オプション”correct=F”をつけないと、イエイツの補正がかかってしまう点に注意！

Pearson's Chi-squared test

```
data: x  
X-squared = 5.1597, df = 1, p-value = 0.02312
```

※ファイル”R_Command_7.txt”を開いて、内容をRの入力画面にコピー＆ペーストしてください

・分割表に対する検定を実施してみましょう。

⑦ 統計検定

```
> fisher.test(x);
```

Fisher's Exact Test for Count Data

data: x

p-value = 0.02804

alternative hypothesis: true odds ratio is not equal to 1

95 percent confidence interval:

1.036471 1.923133

sample estimates:

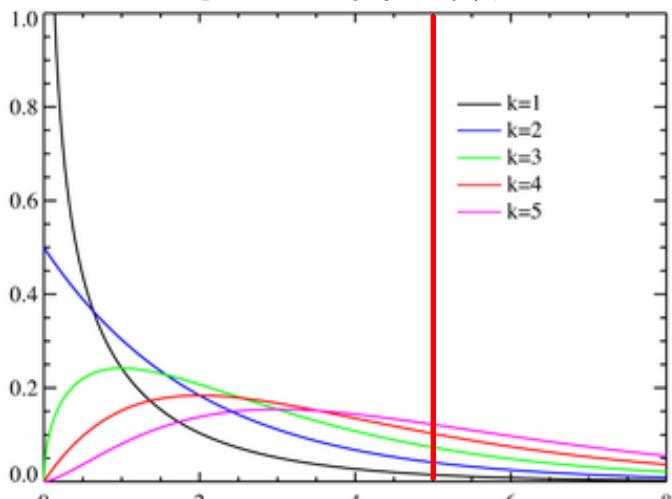
odds ratio

1.411037

・分割表に対する検定を実施してみましょう。

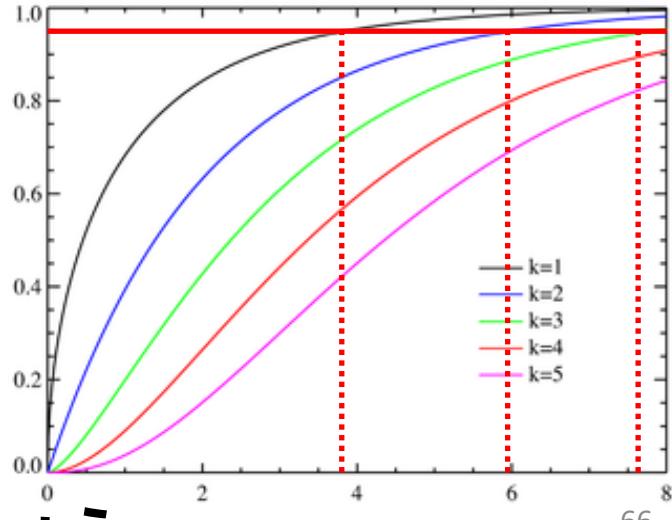
⑦ 統計検定

カイ二乗分布 確率密度関数



積分

累積分布関数



```
> chi.val <- 5.0;  
> pchisq(chi.val, df=1, lower.tail=F);  
[1] 0.02534732  
> pchisq(chi.val, df=2, lower.tail=F);  
[1] 0.082085  
> pchisq(chi.val, df=3, lower.tail=F);  
[1] 0.1717971  
>  
> p.val <- 0.05;  
> qchisq(p.val, df=1, lower.tail=F);  
[1] 3.841459  
> qchisq(p.val, df=2, lower.tail=F);  
[1] 5.991465  
> qchisq(p.val, df=3, lower.tail=F);  
[1] 7.814728
```

・対応するP値とカイ二乗値を計算してみましょう。

統計解析ソフトウェアR入門

- ① 統計解析ソフトウェアRについて
- ② Rのインストール方法
- ③ 数値計算、変数(ベクトル・テーブル)の扱い
- ④ Rの関数
- ⑤ if文とfor文
- ⑥ ファイルの読み書き
- ⑦ 統計検定
- ⑧ グラフの描画

⑧ グラフの描画



データ解析の実施

入力データ



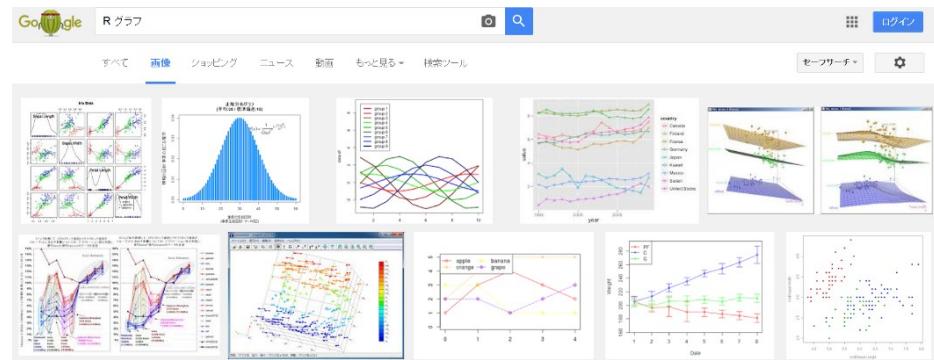
プログラム



出力データ



解析データの可視化



- Rは、解析データのグラフ描画機能が充実しています。
- 研究論文の図を作成する際に、Rでグラフが描けると大変便利です。

⑧ グラフの描画

・グラフを描く関数

関数	機能
plot ()	グラフ(散布図etc.)を描く関数
hist ()	ヒストグラムを描く関数
truehist ()	ヒストグラムを描く関数*
boxplot ()	箱ひげ図を描く関数
barplot ()	棒グラフを描く関数
persp ()	3次元グラフを描く関数
scatterplot3d ()	3次元グラフを描く関数**
ggplot2 ()	多彩な機能を持つグラフ描画関数***

*:MASSライブラリーの呼び出しが必要

**:scatterplot3dライブラリーの呼び出しが必要

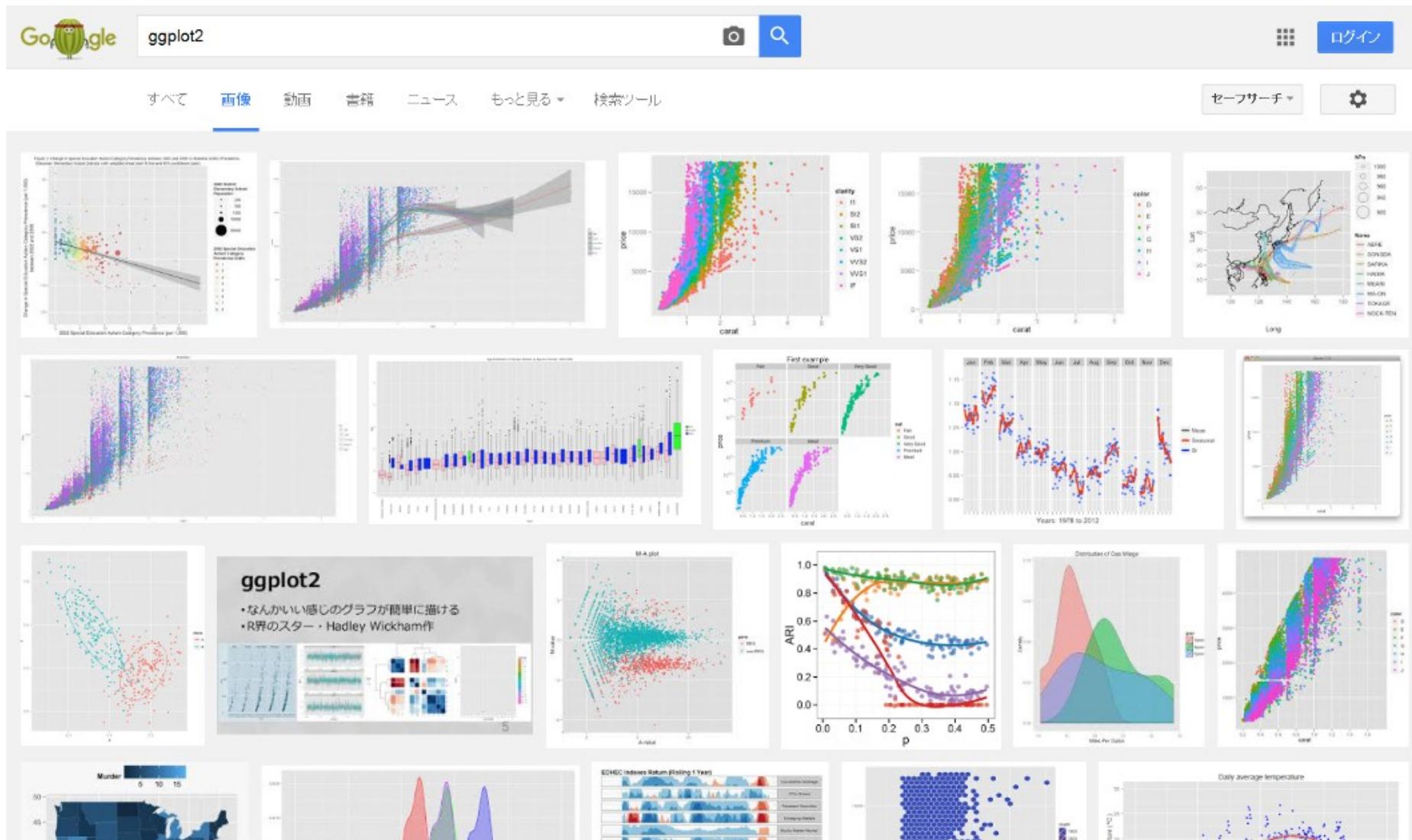
***:ggplot2ライブラリーの呼び出しが必要

・グラフをファイルに出力する関数

関数	機能
png ()	グラフ描画用pngファイルを開く関数
pdf ()	グラフ描画用pdfファイルを開く関数
jpeg ()	グラフ描画用jpegファイルを開く関数
dev.off ()	グラフ描画用ファイルを閉じる関数

・Rには、様々なグラフ描画関数が実装されています。

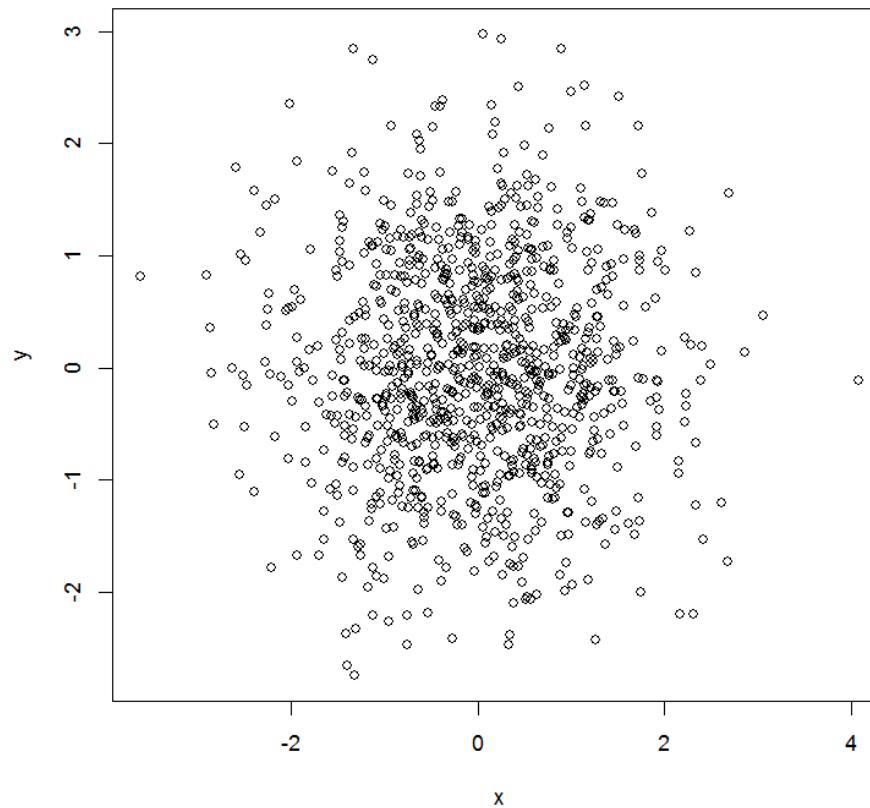
⑧ グラフの描画



・最近では、綺麗な図が描ける関数ggplot2が人気です。
(本講義では説明しません)

⑧ グラフの描画

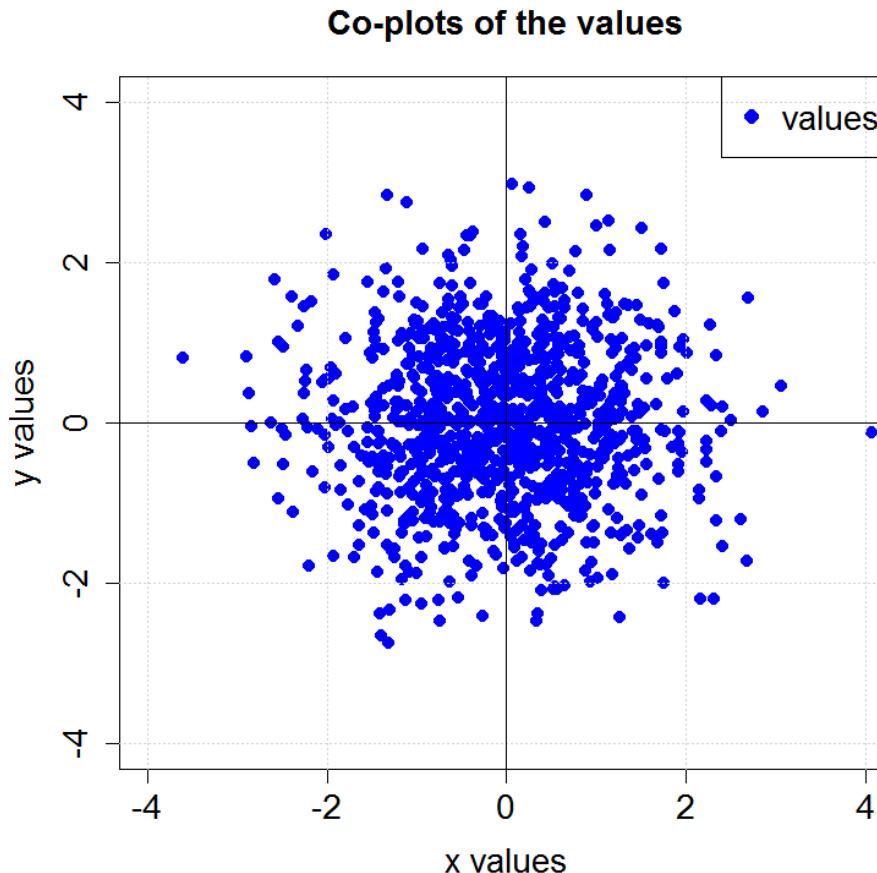
```
> x <- rnorm(1000);  
> y <- rnorm(1000);  
>  
> plot(x, y);
```



・関数plotで、散布図を描くことが出来ます。

⑧ グラフの描画

```
> plot(x, y, col=4, pch=16, xlab="x values", ylab="y values", xlim=c(-4, 4),  
ylim=c(-4, 4), main="Co-plots of the values", cex.lab=1.5, cex.main=1.5,  
cex.axis=1.5, cex=1.2);grid();abline(h=0);abline(v=0);legend("topright", col=4,  
pch=16, legend="values", cex=1.5);
```



※ファイル
"R_Command_8.txt"を
開いて、内容をRの入
力画面にコピー & ペー
ストしてください

・関数plotのオプションを使うことで、綺麗な散布図を描くことが出来ます。

⑧ グラフの描画

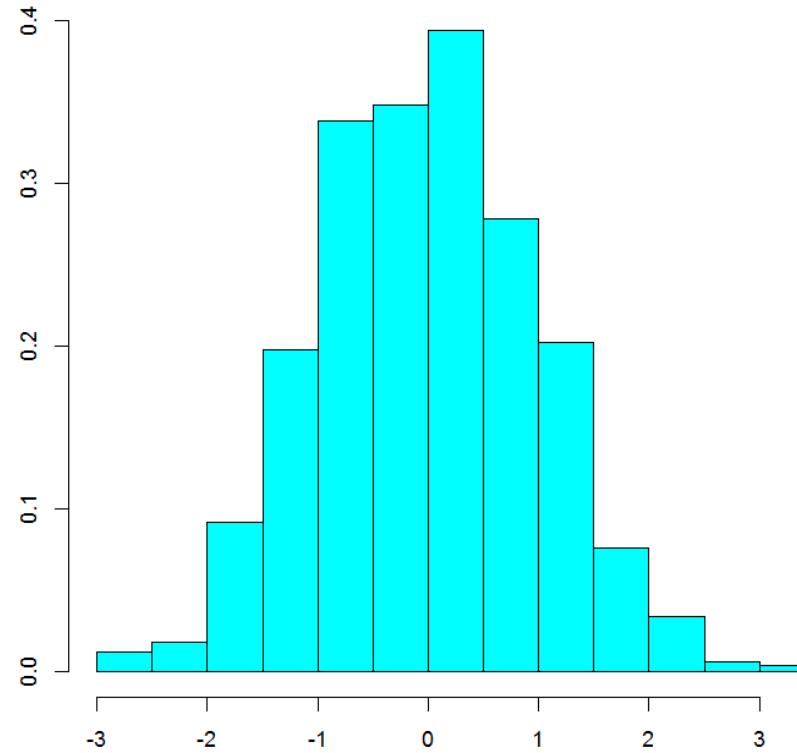
・グラフを描く関数

オプション	機能
col	プロットの色を指定(1:黒、2:赤、3:緑、4:青)
pch	プロットの形を指定(15:■、16:●、17:▲、18:◆)
xlab	x軸の名前を指定
ylab	y軸の名前を指定
xlim	x軸の範囲を指定
ylim	y軸の範囲を指定
main	グラフの名前を指定
cex.lab	x/y軸の名前の表示の大きさを指定
cex.main	グラフの名前の表示の大きさを指定
cex.axis	x/y軸の表示の大きさを指定
cex	プロットの大きさを指定
grid()	グラフに格子を追加表示
abline()	グラフに直線を追加表示
legend	グラフに凡例を追加表示

・関数plotのオプションを使うことで、綺麗な散布図を描くことが出来ます。

⑧ グラフの描画

```
> library(MASS); ← MASSライブラリーの呼び出し  
>  
> x <- rnorm(1000);  
>  
> truehist(x);
```



- ・関数truehistで、ヒストグラムを描くことが出来ます。
- ・MASSライブラリーの呼び出しが必要です。

⑧ グラフの描画

```
> setwd("C:/SummerSchool/Rsoftware");
> library(MASS);
>
> x <- rnorm(1000);
>
> png(file="ExamplePlot.png", width = 500, height = 500);
> truehist(x);
> dev.off();
null device 1

>
> pdf(file="ExamplePlot.pdf");
> truehist(x);
> dev.off();
null device 1
```

※ファイル”R_Command_9.txt”を開いて、内容をRの入力画面にコピー＆ペーストしてください

・Rで描いたグラフは、画像ファイル(pdf, png, jpeg, etc.)で出力可能です。

終わりに

- ・Rの使い方を、駆け足でなぞってみました。
- ・Rを自分で使う際に必要な情報を、一通り網羅してみました。
- ・関数や統計解析の細かい情報については、各自調べてみて下さい。
- ・Rは統計解析だけでなく、プログラミング入門や、データの可視化ツールとしても有用なことがわかったと思います。
- ・是非とも自分で色々と試してみて、使いこなせるようになって下さい。