

R で学ぶデータ解析とシミュレーション

～ R の概要と基本操作 ～



- フリーな統計処理ソフトである R の基本的な使用方法，自分で関数を作成する方法，簡単なシミュレーションを行なう方法，データ解析の基本的な手順を習得します【解説 演習の 2 ステップ】
- R によるデータ解析の入門講座であるため，なるべく統計の基礎知識を前提としないように講義を進めます
- 初等的な数学知識（微積分や行列代数の知識）および基本的なパソコンの知識があれば望ましいです
- 演習時にパソコンを使用します



- R の概要 R でどんなことが出来るか
- R の基本的な使用方法
- R でベクトル、行列の作成
- R でグラフ作成
- R で関数を使用する 関数の定義方法
- R でシミュレーション
- R でデータフレームの作成
- R でデータハンドリング
- R でグラフによるデータマイニング

1 時間目のメニュー



- R の概要
- R の基本【演習】
 - ☐ 起動
 - ☐ 簡単な計算
 - ☐ ヘルプの見方
 - ☐ 終了
- 変数とベクトル
 - ☐ 変数とは
 - ☐ ベクトルとは
 - ☐ 演習
- 参考

R 以外の統計解析用ツール



■ プログラム言語：C 言語や JAVA など

- 計算速度が速い
- プログラム作成が面倒（作成の手間，バグ取り）



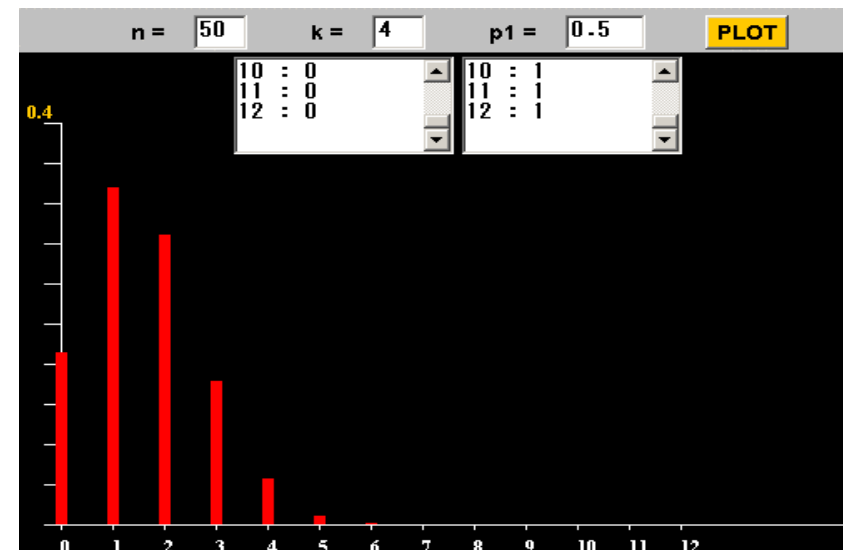
```
C:\ DOS
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>cd c:\

C:\>bcc32 test.c
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
test.c:
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland

C:\>test
t 値は -1.8608, 自由度は 18, p 値は 0.0794 です

C:\>
```

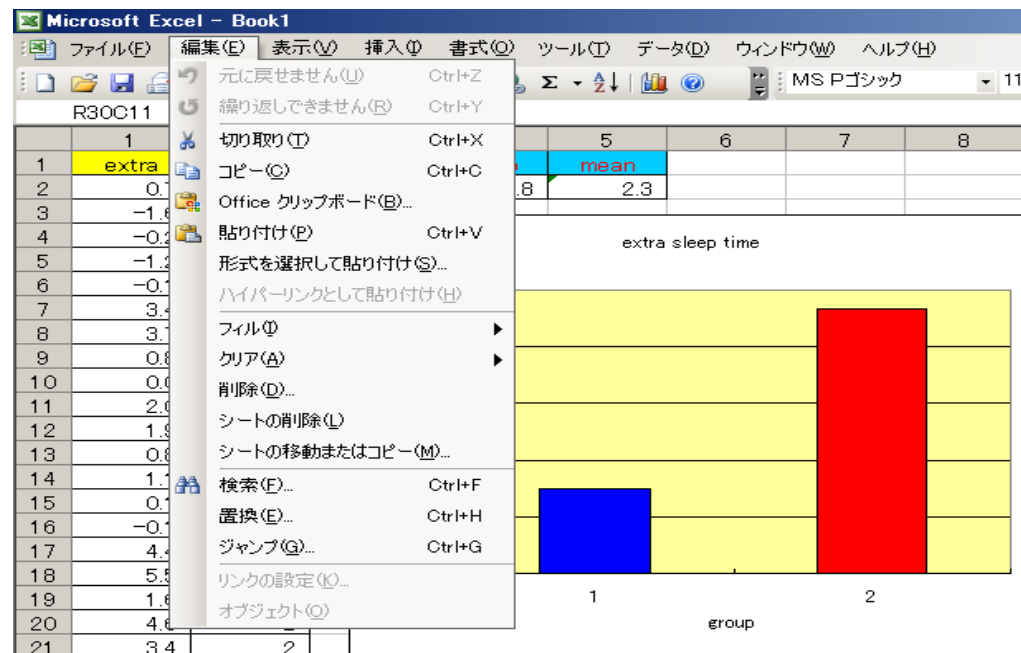


R 以外の統計解析用ツール



■ 表計算ソフト：EXCEL など

- データを GUI 上で操作することが可能
- 手軽にグラフィックス表示をすることが可能
- データの加工やプログラム作成には不向き

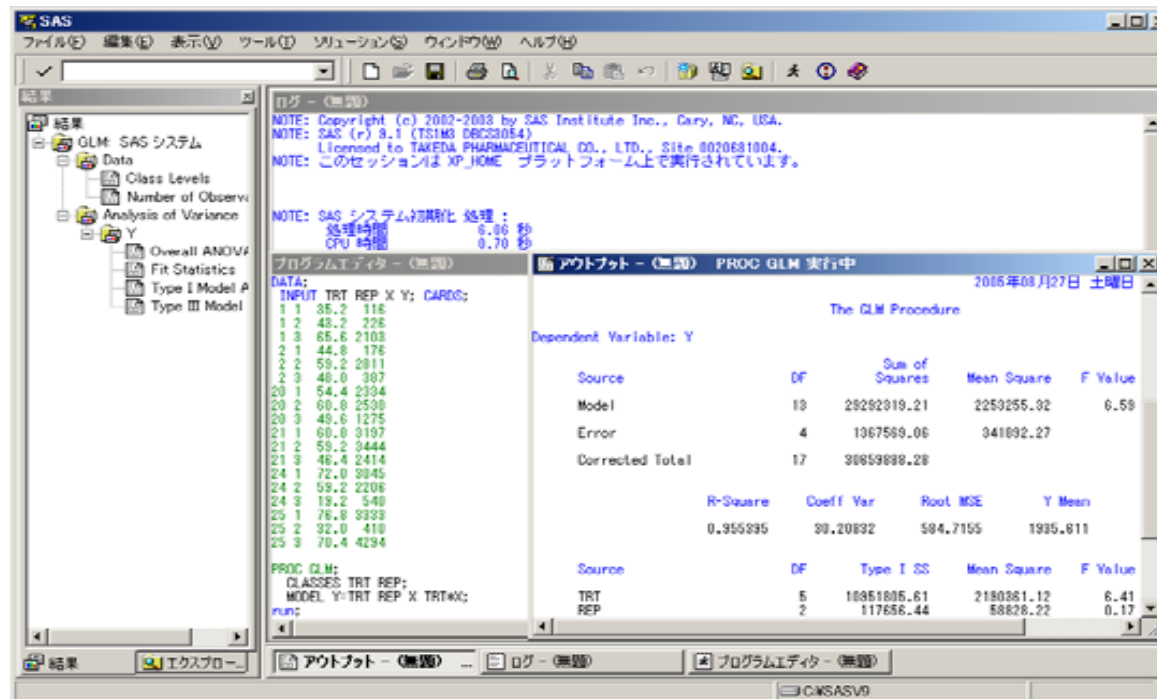


R 以外の統計解析用ツール



■ 統計解析ソフト：SAS や SPSS など

- データ操作に長け，グラフィックス表示が可能，プログラム作成が出来る
- 非常に高価（個人で買える値段ではない）



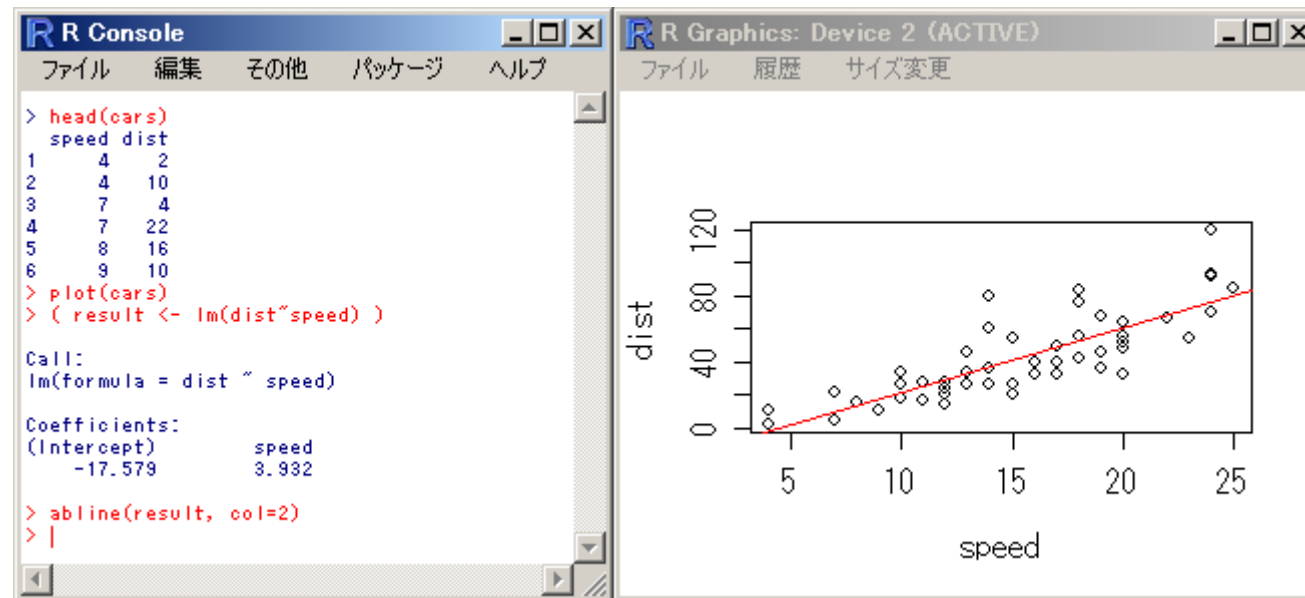
R の概要



■ オープンソース & フリーの統計解析用ソフト

【長所】

- 関数電卓，数値計算，プログラミング，統計解析，グラフィックスの機能があり，どの機能も充実している
- 機能拡張が容易に行える
- 使用人口が多いので，バグが少なく情報も豊富



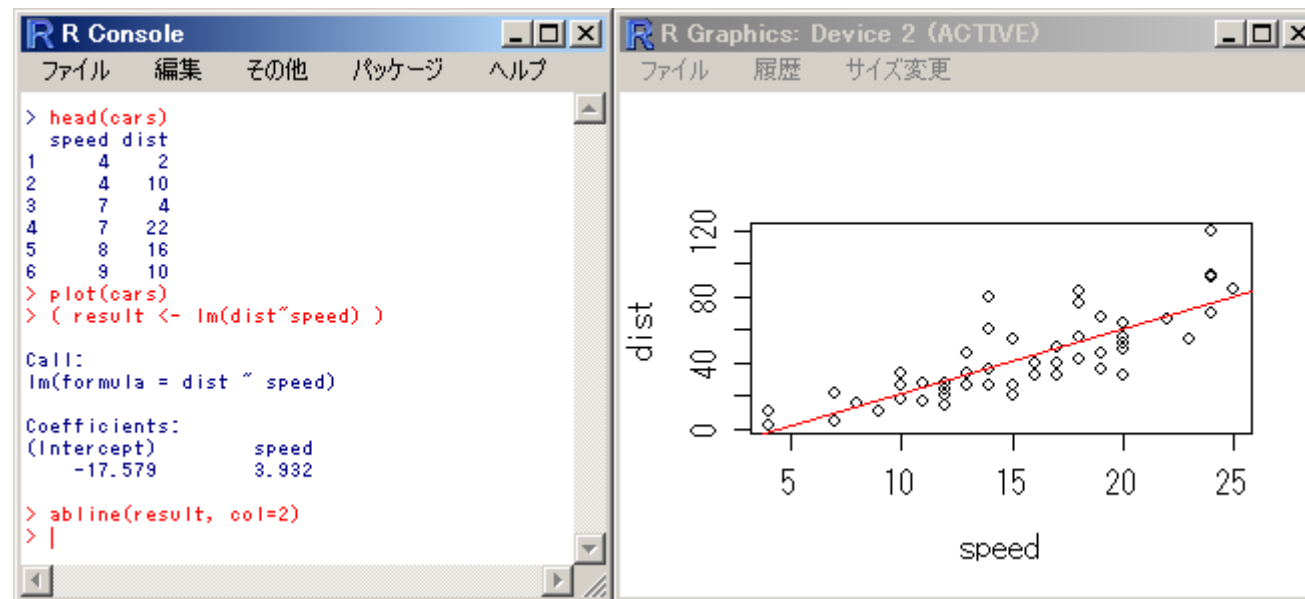
R の概要



■ オープンソース & フリーの統計解析用ソフト

【短所】

- EXCEL などの表計算ソフトに比べて GUI（マウス操作）の機能が劣っているので，R の命令をひとつひとつ覚えなければいけない... GUI 版 R（R Commander）のパッケージあり！
- 大規模なデータを扱う場合は多少骨が折れる



R の概要



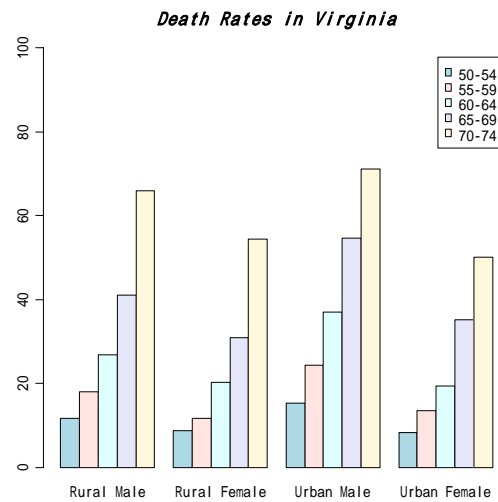
- ニュージーランドの Auckland 大学の講師 Ross Ihaka 氏と Robert Gentleman 氏（当時，現在は Fred Hutchinson Cancer Research Center）が開発した統計解析とグラフィックスのためのソフトウェア
- 当初は，ベル研究所（旧AT&T，現 Lucent Technology）で開発された S 言語と同様の言語仕様にするために改良に改良が重ねられた
- R はオープンソースである（ R を構築したプログラムを見ることが出来る ）ので世界中の人が開発に携わることが出来る
- R は英語のソフトだが，筑波大学の岡田 昌史先生がとりまとめられている翻訳チーム？（ RjpWikiという R コミュニティのメンバー）が日本語に翻訳している！

R で出来ること



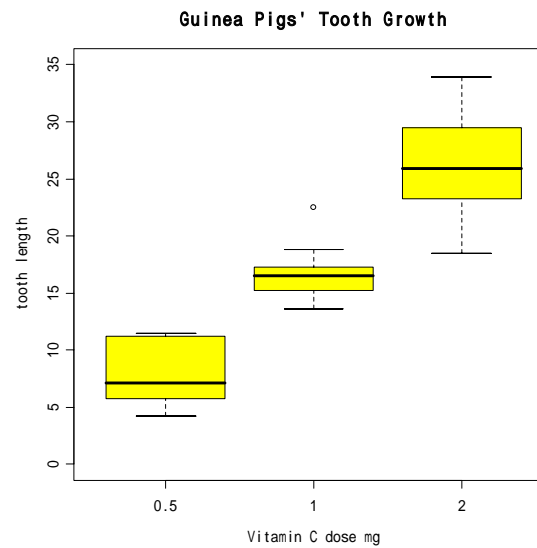
- 四則演算や数学関数を使った複雑な計算を実行する
- 数個～数万個のまとまったデータをベクトルと呼ばれる形で保存しデータの平均値や標準偏差などの要約統計量を求める
- データに関するグラフ（散布図，棒グラフ，円グラフ，数学関数のプロット，三次元立体図など）を作成してデータを視覚的に眺める
- 外部のファイル（テキストや EXCEL ファイル）を読み込む
- 多様な統計手法（各種検定手法，線形モデル，時系列解析，多変量解析，データマイニング，その他多数）を適用する
- プログラムの作成やシミュレーションを実行する
- 新しい統計手法や新しいグラフなどの追加機能が詰まっている「パッケージ」を無料で自分の環境に組み込む

グラフのカタログ



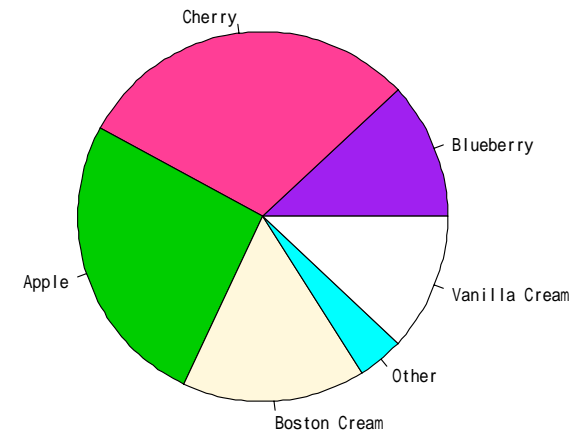
棒グラフ

`barplot(...)`



箱ひげ図

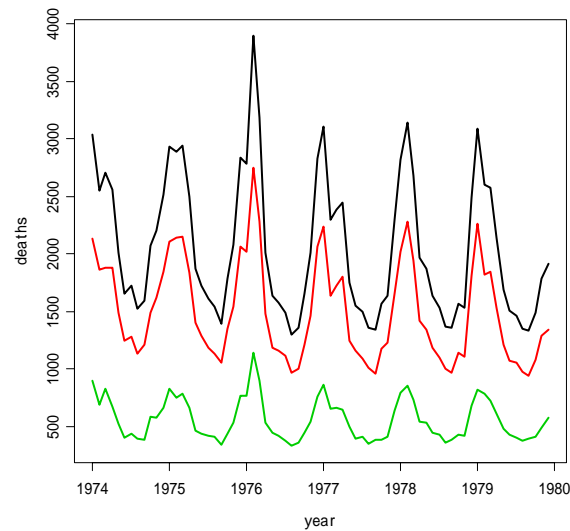
`boxplot(...)`



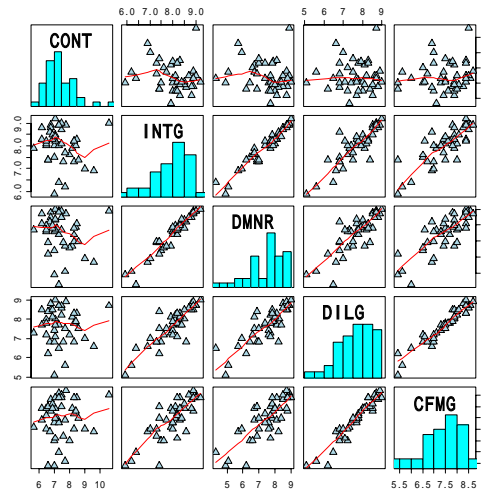
円グラフ

`pie(...)`

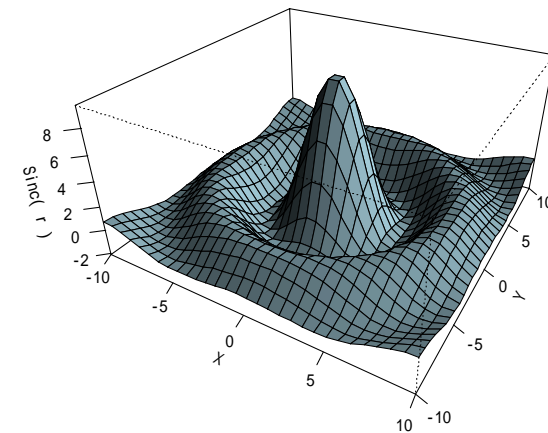
グラフのカタログ



時系列データのプロット
`plot(...)`

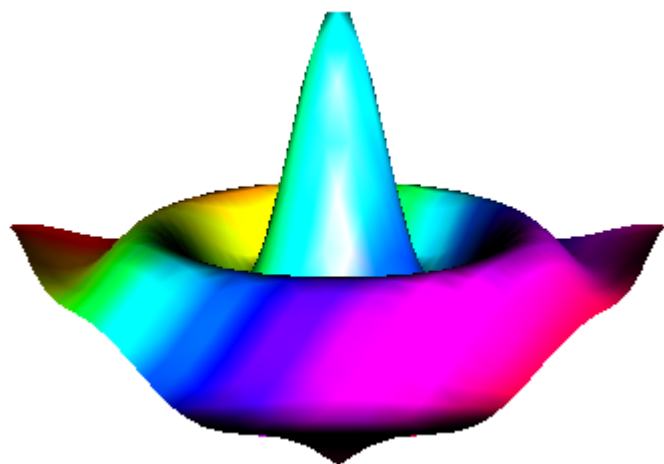


5変数データの散布図
`pairs(...)`



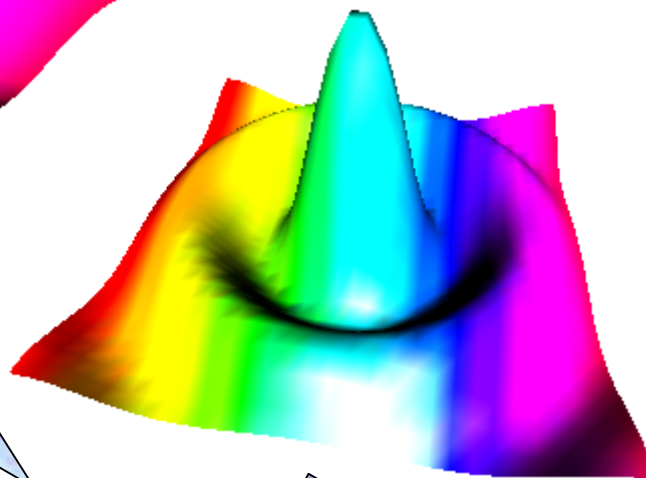
2変数関数のグラフ
`persp(...)`

グラフのカタログ

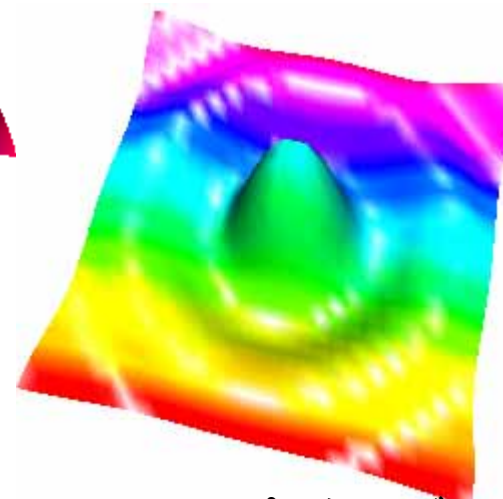


```
rgl.bg(color=c("white", "black"))
rgl.light(theta = 5, phi = 5)
x <- seq(-10, 10, length = 30)
y <- x
f <- function(x,y) {
  r <- sqrt(x^2+y^2); 10 * sin(r)/r
}
z <- outer(x, y, f)
z[is.na(z)] <- 1
ylim <- range(y)
ylen <- ylim[2] - ylim[1] + 1
colorlut <- rainbow(ylen)
col <- colorlut[ y-ylim[1]+1 ]
rgl.surface(x, y, z, color=col)
```

マウスでグリグリ

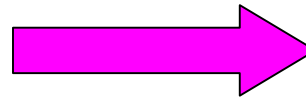
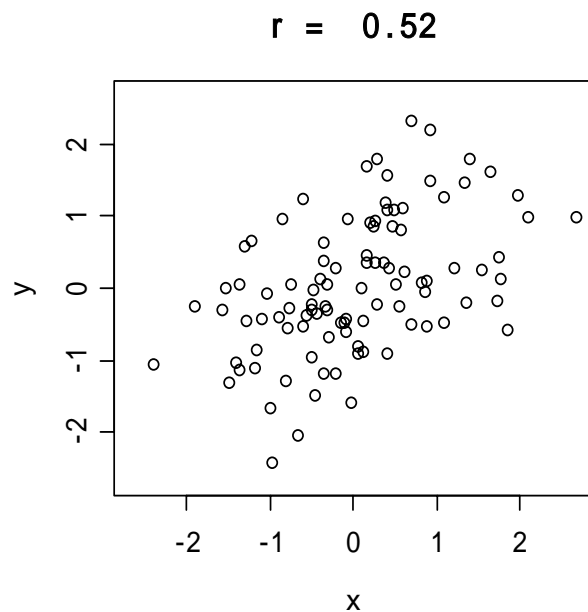


マウスでグリグリ

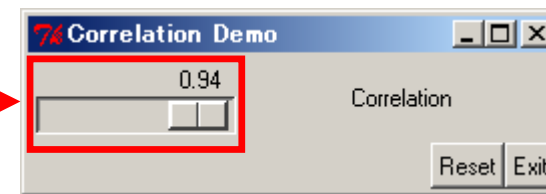
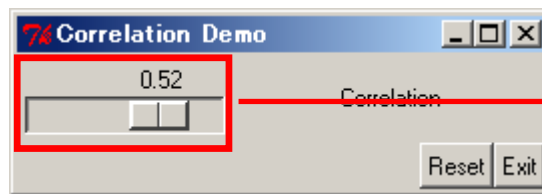
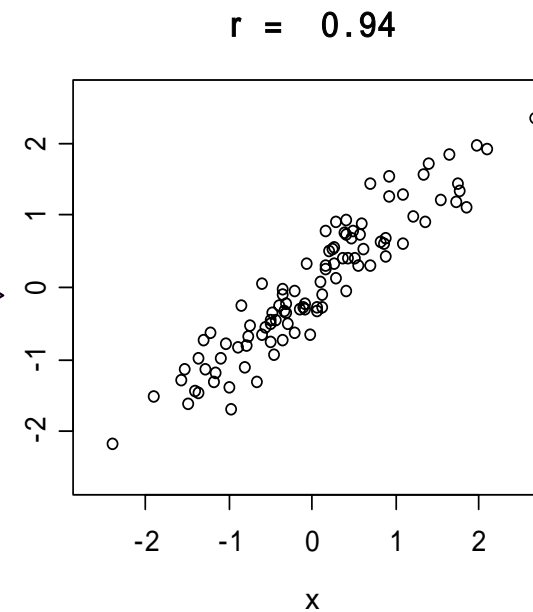


パッケージ rgl

グラフのカタログ



スライダー
動かすこと
図が変わる！





1 時間目のメニュー

- R の概要
- R の基本【演習】
 - 起動
 - 簡単な計算
 - ヘルプの見方
 - 終了
- 変数とベクトル
 - 変数とは
 - ベクトルとは
 - 演習
- 参考


【演習】



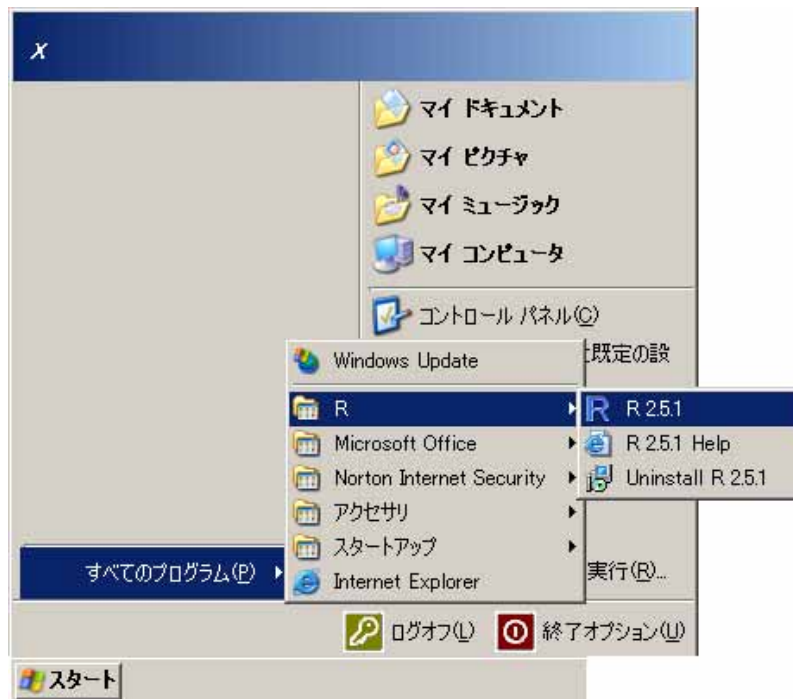
1. R を起動してください
2. 1+2 と入力して Enter キーを押してください
3. $\sqrt{2}$ は `sqrt(2)` で計算することが出来ます
 $\sqrt{2}$ を計算して下さい
4. 対数計算をするため `log(2)` を実行して下さい
5. 4. で `log(2)` を実行しましたが、底が 10 なのか e なのかよく分かりません・・・
`help(log)` で「log」に関するヘルプを見てください
6. 1. ~ 5. まで全て終了しましたら、R を終了してください



1. R の起動（以下の 2 通り）

- アイコン  をクリックする

- メニューから選択する



← クリック！

Windows Vista をお使いの方は，R のショートカットを右クリックした後
「管理者権限として実行」を選択して R を起動してください

2. 「1+2」の計算



```
R Console
ファイル 編集 その他 パッケージ ヘルプ

R version 2.5.1 (2007-06-27)
Copyright (C) 2007 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

Rはフリーソフトウェアであり、「完全に無保証」です。
一定の条件に従えば、自由にこれを再配布することができます。
配布条件の詳細に関しては、'license()'あるいは'licence()'と入力してください。

Rは多くの貢献者による共同プロジェクトです。
詳しくは'contributors()'と入力してください。
また、RやRのパッケージを出版物で引用する際の形式については
'citation()'と入力してください。

'demo()'と入力すればデモをみることができます。
'help()'とすればオンラインヘルプが出ます。
'help.start()'でHTMLブラウザによるヘルプがみられます。
'q()'と入力すればRを終了します。

> 1 + 2
[1] 3
>
```

計算式を入力して [Enter] キーを押す

計算結果が表示される

R では「 計算式を入力」「 結果の表示」という作業を繰り返す

3. ルート 2 の計算



```
> sqrt(2) ⏎  
[1] 1.414214
```

- [Tab] キーを押すことで補完機能が働きます

コラム

```
> sq Tab
```



途中まで入力して [Tab] キーを押すと残りのスペルを補完してくれる

```
> sqrt
```

【参考】R で用意されている演算子・数学関数



演算子

演算子	+	-	*	/	^
意 味	加算	減算	乗算	除算	累乗

数学関数

関 数	$\sin(x)$	$\cos(x)$	$\tan(x)$	$\log(x)$	$\log_{10}(x)$
意 味	$\sin x$	$\cos x$	$\tan x$	$\log_e x$	$\log_{10} x$
関 数	$\sinh(x)$	$\cosh(x)$	$\tanh(x)$	$\exp(x)$	$\text{sqrt}(x)$
意 味	$\sinh x$	$\cosh x$	$\tanh x$	e^x	ルート x
関 数	$\text{abs}(x)$	$\text{trunc}(x)$	$\text{round}(x)$	$\text{floor}(x)$	$\text{ceiling}(x)$
意 味	絶対値	整数部分	丸め	切り下げ	切り上げ

4. log2 の計算



```
> log(2) ↵  
[1] 0.6931472
```

- 記号 # を使ってコメントを付けることができます

コラム

```
> log(2) # この文章は無視されます ↵
```

- 記号 ; で区切ることで複数の数式を同時に実行出来ます

```
> log(2); log(3); log(4) ↵  
[1] 0.6931472 [1] 1.098612 [1] 1.386294
```

5. 関数 log() のヘルプを見る



```
> help(log) ↵
```

- 他にも以下のようなヘルプの検索方法があります

コラム

```
> help.search( " log " )    # log の機能が含まれる関数
> apropos("log")           # "log" という文字列が含まれる関数
```

- ☐ や ☐ で今までに実行したコマンドの履歴が迎れます

```
> ☐ (1つ前に実行したコマンド)が表示される
```

5. 関数 log() のヘルプを見る



log(base)

R Documentation

Logarithms and Exponentials

Description

log computes natural logarithms, log10 computes common (i.e., base 10) logarithms, and log2 computes binary (i.e., base 2) logarithms. The general form logb(x, base) computes logarithms with base base.

Usage

```
log(x, base = exp(1))
```

Arguments

x a numeric or complex vector.

base positive number. The base with respect to which logarithms are computed. Defaults to $e = \exp(1)$.

Details

log10 and log2 are only special cases, but will be computed more efficiently and accurately where supported by the OS.

Value

A vector of the same length as x containing the transformed values. log(0) gives -Inf (when available).

説明

関数の雛形

引数の説明

補足説明

返り値
(結果)

5. 関数 log() のヘルプを見る



```
log(base) R Documentation
```

Logarithms and Exponentials

S4 methods

exp is S4 generic and a member of the [Math](#) group generic.

Note

log and logb are the same thing in R, but logb is preferred if base is specified, for S-PLUS compatibility.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole. (for log, log10 and exp.)

See Also

[Trig](#), [sqrt](#), [Arithmetic](#).

Examples

```
log(exp(3))
log10(1e7) # = 7

x <- 10^-(1+2*1:9)
cbind(x, log(1+x), log1p(x), exp(x)-1, expm1(x))
```

← 無視

← 注意書き

← 参考文献

← 関連の関数

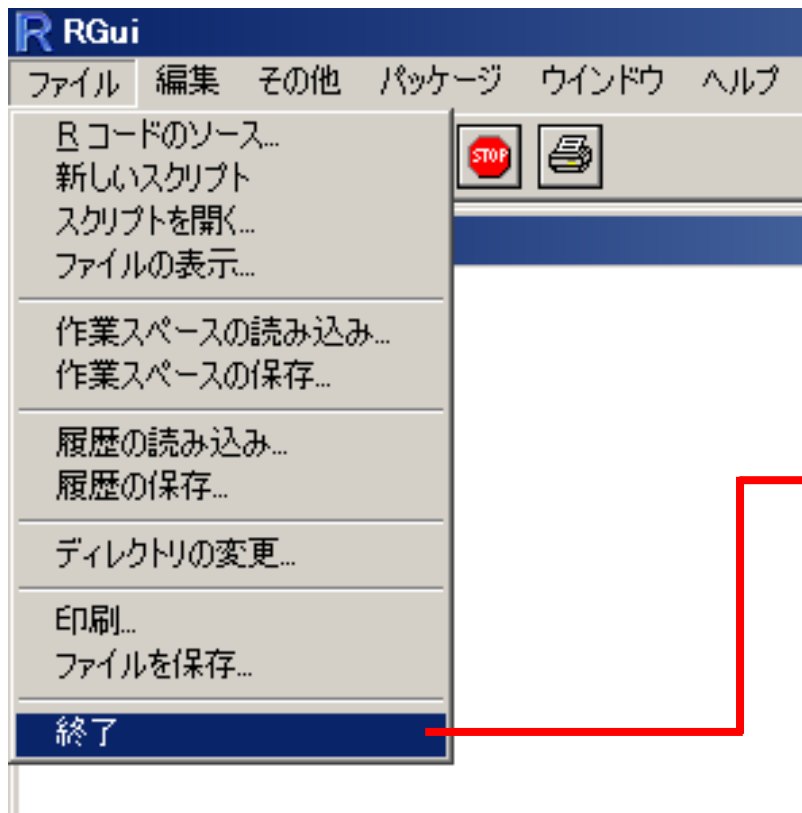
← 例

とりあえず を見た後に を実行すると良い？

6. R の終了



- 関数 `q()` または `quit()` を実行する
- R ウィンドウの右上の `[×]` 印をクリックする
- メニューから選択する



は い：これまでの作業内容を保存する
いいえ：次に使うときはまっさらな状態

【重要】機能が豊富な R を使いこなす近道は？



- 「メモ」「アンチョコ」「自分用のコマンド集」
を作ってください
- ヘルプの見方，分からないことが出てきたときの検索
の方法を身につけてください
- エラーが出ても気にしないでください
- 「とりあえず人様が書いたプログラムを実行」
「そのプログラムの一部を修正して実行」
という作業に慣れてください



1 時間目のメニュー

- R の概要
- R の基本【演習】
 - ☐ 起動
 - ☐ 簡単な計算
 - ☐ ヘルプの見方
 - ☐ 終了
- 変数とベクトル
 - ☐ 変数とは
 - ☐ ベクトルとは
 - ☐ 演習
- 参考

変数とは



- 「変数名 <- 計算式」で計算結果を保存することが出来る
(「変数」への「代入」という ; "<-" は代入記号)

```
> x <- 1 + 2 ↵
```

- 保存した値は変数名を入力することで再表示される

```
> x ↵  
[1] 3
```

- 丸括弧を使うと、値の保存と表示を同時に実行できる

```
> ( x <- 1 + 2 ) ↵  
[1] 3
```

変数の使用例



- ある 5 人の体重について，体重の合計値を求める

```
> x1 <- (55 + 60 + 65 + 70 + 75) ↵  
> x1 ↵  
[1] 325
```

- 同じ 5 人の体重について平均値を求める

```
> x2 <- (55 + 60 + 65 + 70 + 75) / 5 ↵  
> x2 ↵  
[1] 65
```

- 同じ 5 人の体重について，今度は分散を求め・・・
(同じデータなのだから，手間を省く方法はないの??)

ベクトルの作成〔関数 c() を使用した方法〕



- 1 つの変数に 1 つの値を代入することが出来たように ,
1 つの変数に複数の値を代入することも出来る

```
> x <- c(55, 60, 65, 70, 75) ↵
```

- 複数の値が代入された変数を「ベクトル」とよぶ
1 つの値しか代入されていない変数も「ベクトル」

```
> x ↵  
[1] 55 60 65 70 75
```

ベクトル用の関数



- R には「ベクトル」用の関数が多数用意されている

```
> sum(x) ↵  
[1] 325
```

- ベクトルに入っているデータ数（要素数）を計算する場合は関数 `length()` を使用する

```
> length(x) ↵  
[1] 5
```


【参考】R で用意されているベクトル用関数



ベクトル用関数


関 数	cor(x)	max(x)	mean(x)	median(x)	min(x)
意 味	相関係数	最大値	平均値	中央値	最小値
関 数	prod(x)	summary(x)	sd(x)	sum(x)	var(x)
意 味	総積	要約統計量	標準偏差	総和	不偏分散

コラム

- コマンド入力の途中で  を押してしまうと . . .

```
> log(2 
```

+

- の状態になっても気にせずに入力を続けてください
(コマンド入力を中断する場合は  を押してください)

ベクトル操作



- ベクトルの中の数値を取り出す場合は [] を使用する

```
> x[2]    # 変数 x の 2 番目の値を取り出す ␣  
[1] 60
```

- ベクトルとベクトルを結合する場合は以下の様にする

```
> c(x, 80) # c(ベクトル, ベクトル) ␣  
[1] 55 60 65 70 75 80
```

- ベクトルの中の数値を書き換える場合は以下の様にする

```
> x[2] <- 99 # 変数 x の 2 番目の値を 99 に修正 ␣  
> x ␣  
[1] 55 99 65 70 75
```

【演習】



1. R を起動して「0.8-4」の結果を変数 x に代入してください
2. 整数部分 `trunc(x)` と切り下げ `floor(x)` の違いは何でしょう？
3. 以下の 5 つのデータを変数 y に格納してください

1	2	3	4	5
---	---	---	---	---

4. 変数 y の値と変数 Y（大文字）の値を表示してください
5. 3. で作成した変数 y の平均と標準偏差を求めてください
6. 変数 y の後ろに，変数 x を結合し，結果を変数 z に格納してください
7. 6. で作成した変数 z の 6 番目の値を表示してください



1 時間目のメニュー

- R の概要
- R の基本【演習】
 - ☐ 起動
 - ☐ 簡単な計算
 - ☐ ヘルプの見方
 - ☐ 終了
- 変数とベクトル
 - ☐ 変数とは
 - ☐ ベクトルとは
 - ☐ 演習
- 参考〔統計量に関する補足，行列の作成〕

【参考】統計量に関する補足



■ **sum()** : 総和 データを全て足し合わせた値

■ **mean()** : 平均値

データの和をデータ数で割った値

□ データの真ん中を表す尺度のひとつ

□ 他のデータに比べて極端に大きい値や小さい値（外れ値）がデータの中にある場合は，平均値がその値に引っ張られてしまうという欠点あり

■ **median()** : 中央値

データを値の小さい順に並べたときに真ん中の順位にくるデータの値

□ データの真ん中を表す尺度のひとつ

□ データ数が偶数個の場合は，中央に位置する 2 個の値の平均値が中央値

【参考】統計量に関する補足



■ `var()` : 分散（不偏分散）

「データとデータの平均値との差」を2乗したものを全て足し合わせ，「データ数 - 1」で割った値

- データの平均値からの散らばり具合を表す指標
- 値が大きければデータはよく散らばっている
- 値が小さければデータが平均値の周りに集まっている

■ `sd()` : 標準偏差 分散の平方根（ルート）をとった値

- データの平均値からの散らばり具合を表す指標
- 標準偏差は分散と違って，元のデータと単位が揃っている

【参考】 平均値と中央値



```
> x <- c(1, 2, 3, 3, 3, 4, 4, 5, 5, 70) ␣  
> mean(x)    # ベクトル x について平均値を計算する ␣  
[1] 10  
> median(x)  # ベクトル x について中央値を計算する ␣  
[1] 3.5
```

- ベクトル x の中に 70 という外れ値があるため，平均値が大きくなる方向へ押し上げられてしまっている
10 よりも小さい値は全体の 90% を占める...
 - ベクトル x の中に 70 という外れ値があるにも関わらず，中央値は 3.5 となっている
 - 3.5 よりも小さい値は全体の丁度 50%
 - 中央値は外れ値に引っ張られてにくいという利点がある！
- ベクトル x のデータ数は 10 個なので，中央値は小さい方から 5 番目の値 (3) と 6 番目の値 (4) の平均となる



【参考】行列の作成方法

- 行列 $\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$ を作成する手順は以下のとおり
- 1. 行列の要素をベクトルで用意する
- 2. 関数 `matrix(ベクトル, 行数, 列数)` でベクトルから行列に変換する

```
> x <- c(1, 2, 3, 4, 5, 6) ↵
```

```
> A <- matrix(x, 2, 3) ↵
```

```
> A ↵
```

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


【参考】R で用意されている行列用演算子 & 関数



演算子

演算子	+	-	%*%	*
意 味	加算	減算	行列の積	対応する要素同士の積

数学関数

関数	機能
chol(A)	行列 A のコレスキー分解を行う
crossprod(A)	行列 A のクロス積を求める
eigen(A)	行列 A の固有値と固有ベクトルを求める
ginv(A)	行列 A の一般化逆行列を求める (MASS パッケージの呼び出しが必要)
qr(A)	行列 A の QR 分解を行う
svd(A)	行列 A の特異値分解を行う
solve(A)	行列 A の逆行列を求める
t(A)	行列 A の転置を行う

【参考】行列要素の取り出し



コマンド	機能
A[1,]	行列 A の 1 行目を取り出す
A[, 1]	行列 A の 1 列目を取り出す
A[1, 2]	行列 A の 1 行 2 列目の成分を 取り出す
A[c(1,2), 3]	行列 A の 1, 2 行 3 列目の成分を取り出す
A[c(1,2), c(2,3)]	行列 A の 1, 2 行目と 2, 3 列を取り出す

```
> A[1, ] ↵
```

```
[1] 1 3 5
```

```
> A[1, 2] ↵
```

```
[1] 3
```



【参考】行列計算の例

```
> x <- c(1,2,3,4) ␣
> A <- matrix(x, 2, 2) ␣
> A * A      # 対応する要素同士の積 ( 行列の積 ) ␣
      [,1] [,2]
[1,]    1    9
[2,]    4   16
> A %*% A    # 行列の積 ␣
      [,1] [,2]
[1,]    7   15
[2,]   10   22
> t(A)       # 行列 A の転置行列 ␣
      [,1] [,2]
[1,]    1    2
[2,]    3    4
```

演習（38枚目のスライド分）の回答例



```
> x <- 0.8-4 # 設問 1 の回答例
> trunc(x) ; floor(x) # 設問 2 の回答例（結果は省略）
[1] -3 -4 # マイナスの場合は注意！
> ( y <- c(1,2,3,4,5) ) # 設問 3 と 4 の回答例
[1] 1 2 3 4 5
> Y # R では大文字と小文字を区別する！
エラー： オブジェクト "Y" は存在しません
> mean(y) # 設問 5 の回答例（変数 y の平均値）
[1] 3
> sd(y) # 設問 5 の回答例（変数 y の標準偏差）
[1] 1.581139 # sd() は不偏標準偏差であることに注意！
> ( z <- c(y, x) ) # 設問 6 の回答例
[1] 1 2 3 4 5 6
> z[6] # 設問 7 の回答例
[1] 6
```



1 時間目にやったこと

- R の概要
- R の基本【演習】
 - 起動
 - 簡単な計算
 - ヘルプの見方
 - 終了
- 変数とベクトル
 - 変数とは
 - ベクトルとは
 - 演習
- 参考〔統計量に関する補足，行列の作成〕

終