

# 統計解析フリーソフト 入門

～ インストール方法から簡単な計算方法, グラフの出力方法まで ～

## 【目次】

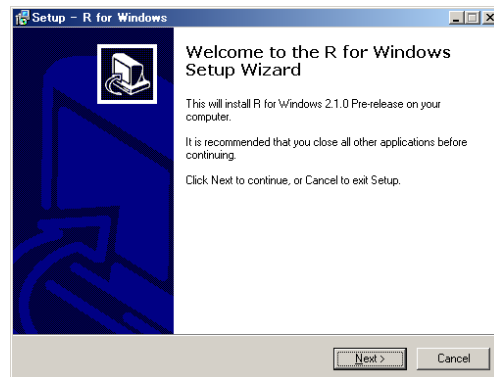
|      |                           |    |
|------|---------------------------|----|
| 1章   | インストール方法                  | 2  |
| 1.1節 | Windows版Rの場合              | 2  |
| 1.2節 | MacOS X (10.2, 10.3)版Rの場合 | 3  |
| 2章   | 簡単な計算方法                   | 4  |
| 2.1節 | Rによる計算方法                  | 4  |
| 2.2節 | 電卓としての使い方                 | 5  |
| 3章   | グラフ出力                     | 6  |
| 3.1節 | データの準備                    | 6  |
| 3.2節 | グラフの出力                    | 6  |
| 3.3節 | グラフの保存方法(簡便法)とプログラムの保存    | 9  |
| 3.4節 | グラフ集                      | 10 |
| 3.5節 | R Editorの保存方法             | 12 |
| 3.6節 | (参考)統計解析入門                | 13 |
| 4章   | 落穂ひろい                     | 15 |
| 4.1節 | Rによる関数定義の方法               | 15 |
| 4.2節 | その他                       | 17 |

# 1章 インストール方法

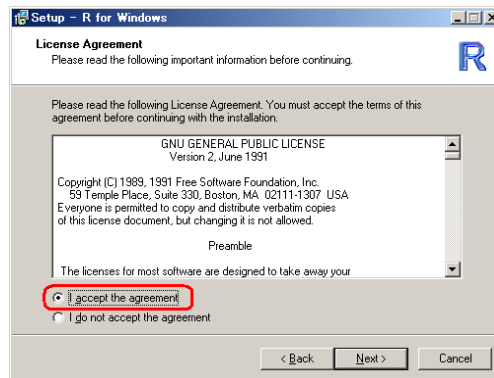
## 1.1節 Windows 版 R の場合

<http://r.nakama.ne.jp/Ri18n/> には, Windows 版 R-2.1.0 の国際版 (開発中) が, <http://r.nakama.ne.jp/> には Windows 版 R-2.0.1 を中間 栄治さんが日本語化した R (安定版) があります. 以下, R-2.1.0 の国際版 R のインストール方法を紹介します.

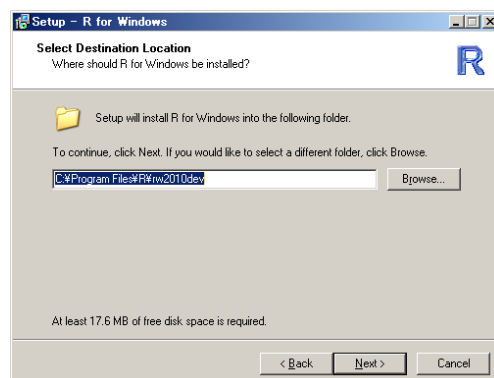
(1) <http://r.nakama.ne.jp/Ri18n/>にあるファイル「rw2010dev.exe」をダブルクリックすると, R のインストーラーが起動します. すると, 右図のようなウィンドウが表われます. ボタン[Next>]をクリックするとインストール作業が始まります.



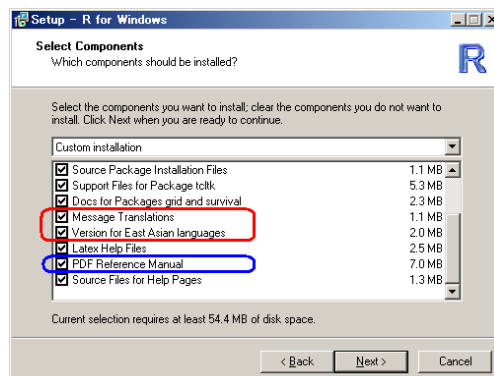
(2) すると右図のような画面が出ます. ライセンス条項を読んだ後, チェック項目 [I accept the agreement] をチェックして, [Next>] をクリックします.



(3) 次に, R をインストールするフォルダを指定します. 通常はそのまま [Next>] をクリックします.



(4) クリックすると右図のような画面が出ます. ここで, 必ず「Message Translations」と「Version For East Asian Languages」をチェックしてください (一覧の下の方にあります). 「PDF Reference Manual」はお好みでチェックしてください. あとは数回 [Next>] をクリックします.



しばらくするとインストールが終了します. [Finish] をクリックして下さい.

## 1.2節 MacOS X (10.2, 10.3) 版 R の場合

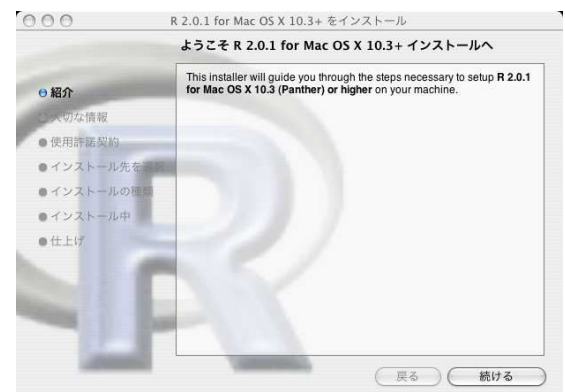
参考までに, MacOS X 版 R のインストール方法を紹介します.

(1) <ftp://ftp.u-aizu.ac.jp/pub/lang/R/CRAN/bin/macosx/>

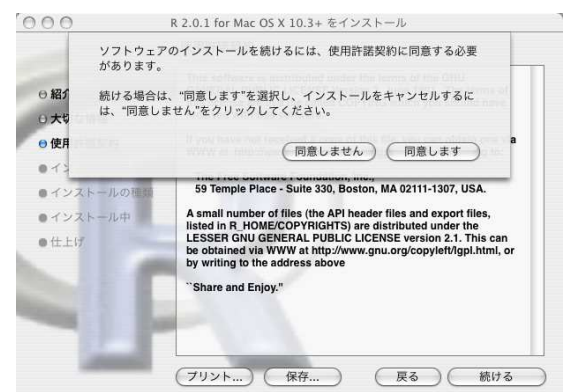
にあるファイル「R-2.0.1.dmg」をダウンロードします. ダウンロードしたファイル「R-2.0.1.dmg」ダブルクリックすると, R のディスクイメージがマウントされます. マウントされた R-2.0.1 の中から R.\*\*\*\*.mpkg (お使いの OS のバージョン用のパッケージ) をダブルクリックしてインストーラーを起動します.



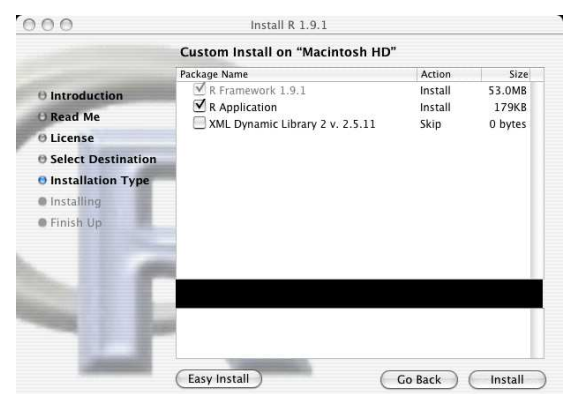
(2) すると, 右図のようなウィンドウが表われます. ボタン[続ける] (または[Continue]) をクリックし続けるとインストール作業が進みます.



(3) ただし, 右図のような使用許諾契約に同意する必要があります. あとはボタン[続ける] (または[Continue]) をクリックし続ければインストールが終了します.



(4) MacOS 10.2 を使っている場合は, XML Dynamic Library 2 を追加するよう, インストールをカスタマイズする必要があります. XML Dynamic Library 2 のチェックを追加して, [Install] をクリックするとインストールが始まり, しばらくするとインストールが終了します.



R を使う場合は, アプリケーションフォルダに [R] というアプリケーションが出来ておりますので, これをダブルクリックすることで R が起動します.

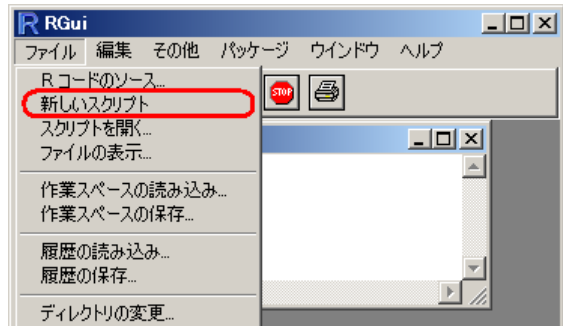
## 2章 簡単な計算方法

ここでは、Windows 版 R を使って、R による簡単な計算方法を紹介します。

### 2.1節 R による計算方法

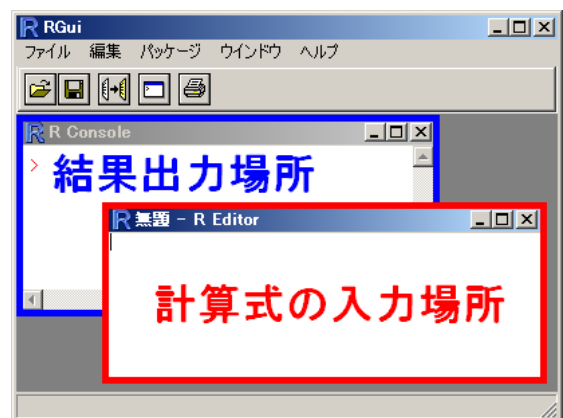
まず、R への入力方法と計算実行の方法を説明します。

(1) まずは R を起動します。すると、「R Console」というウィンドウが表れます。次に、メニューの [ファイル] → [新しいスクリプト] を選択します。

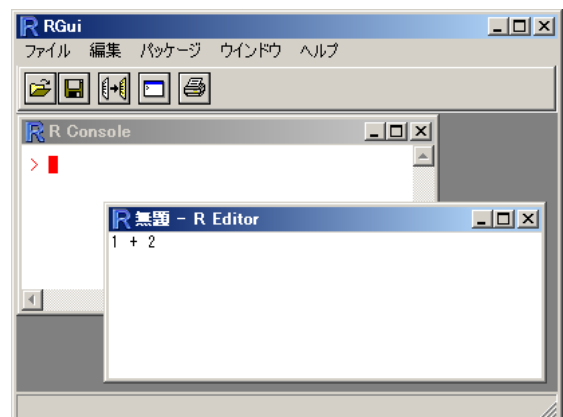


(2) すると、「無題 - R Editor」という名前のウィンドウが表れます。これら 2つのウィンドウは

- R Console : 結果出力ウィンドウ
  - R Editor : 計算式の入力ウィンドウ
- という働きをします。R では、「R Editor」に計算式を入力し、計算を実行すると結果が「R Console」に出力される、という流れになっています<sup>1</sup>。



(3) 試しに「1+2」という簡単な計算を行ってみましょう。まず、「R Editor」に計算式「1+2」を入力します。



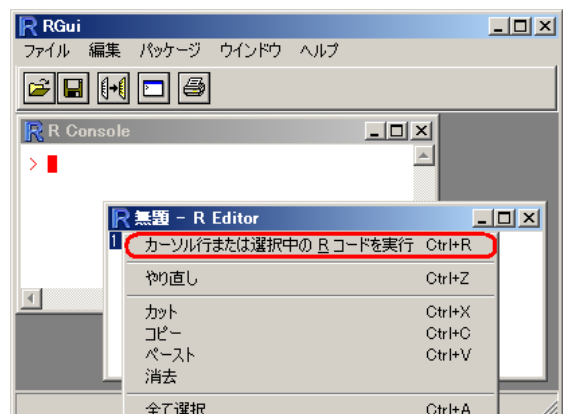
(4) 計算式「1+2」を入力した後、計算式を実行する場合は、計算したい式をマウスで範囲選択した後、右クリックから [カーソル行または選択中の R コードを実行] を選択します。

(参考) ショートカットキーがお好きな方は、計算したい式をマウスで範囲選択した後、

Windows : [Ctrl]+[R]

Mac OS X : [りんごマーク]+[Enter]

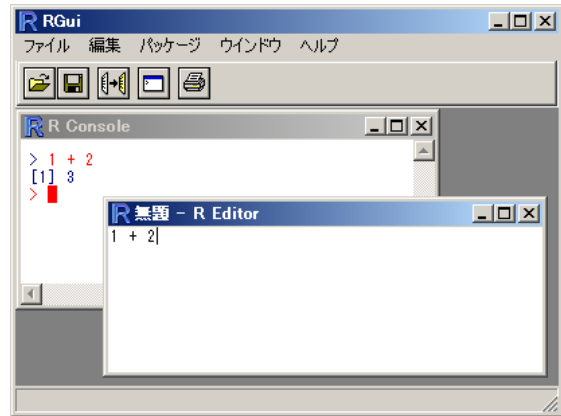
を押しても同様のことが行えます。



<sup>1</sup> 「R Console」に計算式を直接入力しても計算することができます。

(5) すると、「1+2」の計算結果「3」が「R Console」に出力されました。赤字が入力式、青字が計算結果となっています。

計算式は複数行あってもかまいません。1つの計算式が複数行にまたがっている場合も、2つ以上の計算式を同時に計算したい場合も、マウスで範囲選択した後、右クリックから「カーソル行または選択中の R コードを実行」を選択すれば計算することが出来ます。



## 2.2節 電卓としての使い方

それでは、Rでどのような計算が行えるかを見てみましょう。

(1) Rでは、以下で紹介する演算子の他に、カッコ（と）を使うことが出来ます。

|    |    |    |    |    |
|----|----|----|----|----|
| +  | -  | *  | /  | ^  |
| 加算 | 減算 | 乗算 | 除算 | 累乗 |

これらの演算子を使うことで、簡単な計算を気軽に行うことが出来ます。

(2) 数学関数を使った計算も出来ます。以下の表は、Rで使える数学関数の抜粋です。

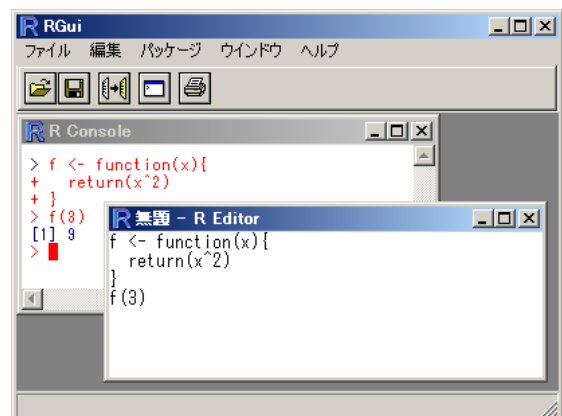
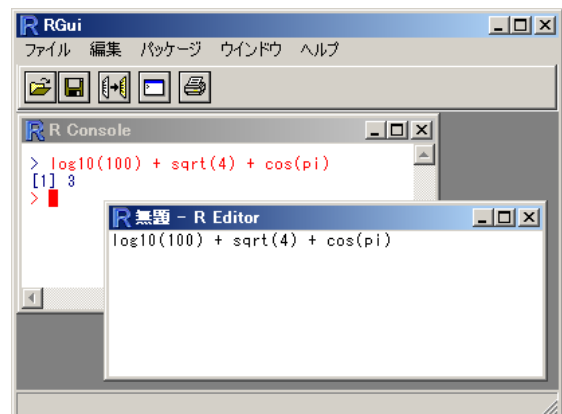
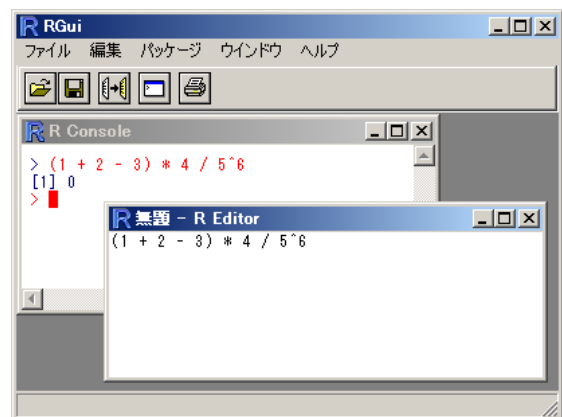
|         |          |          |                    |                     |
|---------|----------|----------|--------------------|---------------------|
| sin(x)  | cos(x)   | tan(x)   | log(x)             | log10(x)            |
| sinx    | cosx     | tanx     | log <sub>e</sub> x | log <sub>10</sub> x |
| sinh(x) | cosh(x)  | tanh(x)  | exp(x)             | sqrt(x)             |
| sinhx   | coshx    | tanhx    | e <sup>x</sup>     | x <sup>1/2</sup>    |
| abs(x)  | trunc(x) | round(x) | floor(x)           | ceiling(x)          |
| 絶対値     | 整数部分     | 丸め       | 切り捨て               | 切り上げ                |

右の計算では円周率 pi を使っています。

(3) 自分で関数を定義することも出来ます。試しに、関数 f(x)=x<sup>2</sup>を定義してから、f(3)を計算します。まず関数の定義は

```
f <- function(x) {
  return(x^2)
}
```

とします。右図では f(3) = 3<sup>2</sup> = 9 を計算しています<sup>2</sup>。



<sup>2</sup> 第4章「落穂ひろい」に、関数定義方法のもう少し詳しい解説を載せております。

## 3章 グラフ出力

Windows 版 R を使ったグラフの出力方法を紹介します。まず、グラフを出力する方法の手順を示します。

- (1) グラフに出力する「データ」または「関数」を用意します。
- (2) グラフ出力用関数に、「データ」または「関数」を入れて実行します。
- (3) 必要に応じてグラフをファイルに保存します。

### 3.1節 データの準備

ここでは、データの数が少ない場合のデータの準備方法を紹介します。

| ID | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
|----|----|----|----|----|----|----|----|----|----|
| 体重 | 60 | 59 | 63 | 72 | 66 | 68 | 63 | 65 | 64 |

以下では、変数 ID に番号のデータを、変数 weight に体重のデータを入力しています。データを入力する場合は関数 c() を使います。

```
ID <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
weight <- c(60, 55, 63, 72, 66, 68, 69, 65, 64)
```

ID や weight にデータを入力した後、データの中身を確認する場合は変数名を入力します。

```
weight
```

```
[1] 60 55 63 72 66 68 69 65 64
```

### 3.2節 グラフの出力

まずは、データ weight1 と weight2 を用いてグラフを描く方法を紹介します。

#### ① データのグラフ

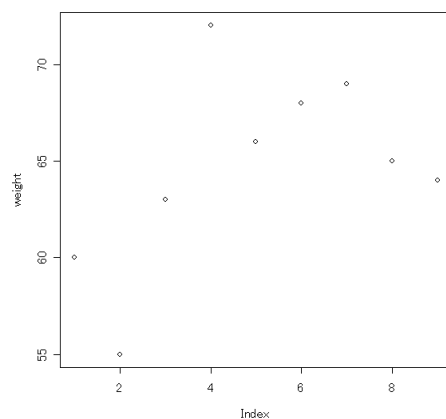
(1) データの散布図を描く場合は

```
plot(weight)
```

と入力します。x 軸は自動的に 1, 2, ... となり、y 軸はデータ weight の値となります。

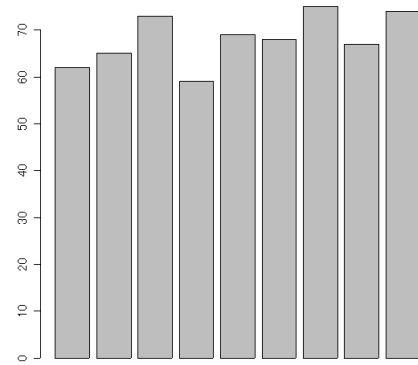
x 軸を明示的に指定する場合、例えば x 軸にデータ ID を指定する場合は以下のようにします。

```
plot(ID, weight)
```



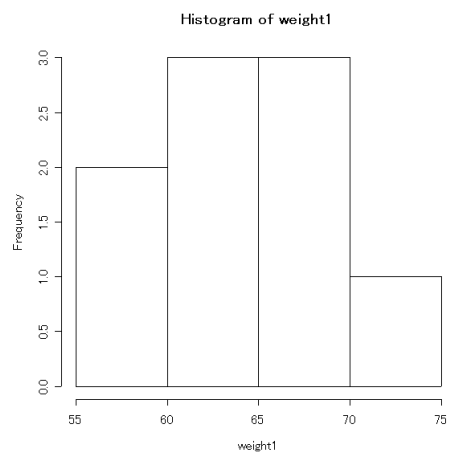
(2) 棒グラフを描く場合は関数 `barplot()` を使います.

```
barplot(weight2)
```



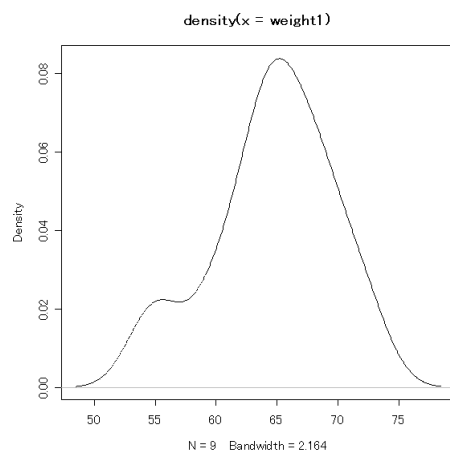
(3) ヒストグラムを描く場合は関数 `hist()` を使います.

```
hist(weight1)
```



(4) ヒストグラムのような棒ではなく、曲線でデータの密度を描く場合は関数 `plot()` と関数 `density()` を組み合わせます.

```
plot(density( weight1 ))
```

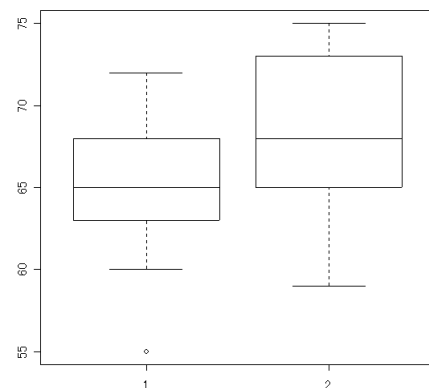


(5) 箱ひげ図を描く場合は、関数 `boxplot()` を使います.

```
boxplot(weight1)
```

データ `weight1` とデータ `weight2` の箱ひげ図を1つのグラフに描く場合は、データの名前を並べるだけでOKです.

```
boxplot(weight1, weight2)
```



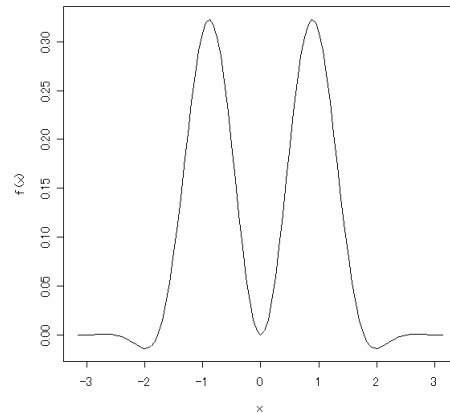
次に、関数のグラフを描く方法を紹介します。

## ② 関数のグラフ

- (6) 数学関数のグラフを描く場合は、関数 `curve()` を使います。例えば、 $f(x) = \sin(x^2) \cdot \exp(-x^2)$  のグラフを描く場合は以下のようにします。

```
f <- function(x) { sin(x^2)*exp(-x^2) }  
curve(f, -pi, pi)
```

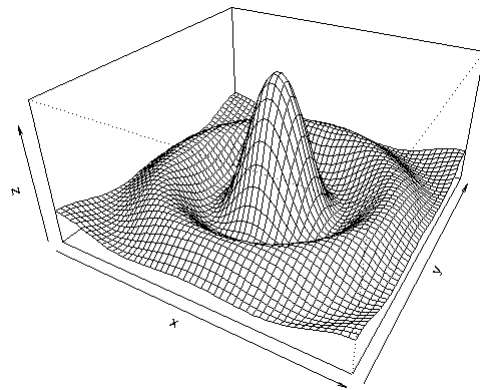
`curve(関数, x 軸の下限, x 軸の上限)` という書式になっている点に注意して下さい。



- (7) 3次元関数を描く場合は関数 `persp()` を使いますが、多少の手間がかかります。以下では

$f(x,y) = \frac{10\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$  のグラフを描いて

います。



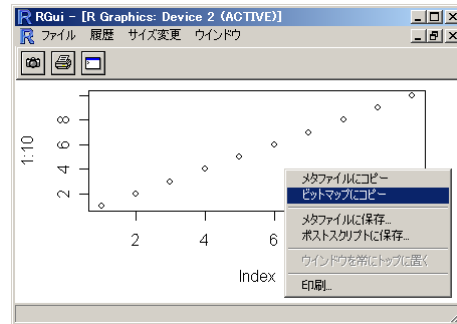
```
f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r } # 関数の定義  
x <- seq(-10, 10, length= 50) # x 軸の刻み幅  
y <- x # y 軸の刻み幅  
z <- outer(x, y, f) # z 軸の値  
persp(x, y, z, theta = 30, phi = 30, expand = 0.5)
```



### 3.3節 グラフの保存方法（簡便法）とプログラムの保存

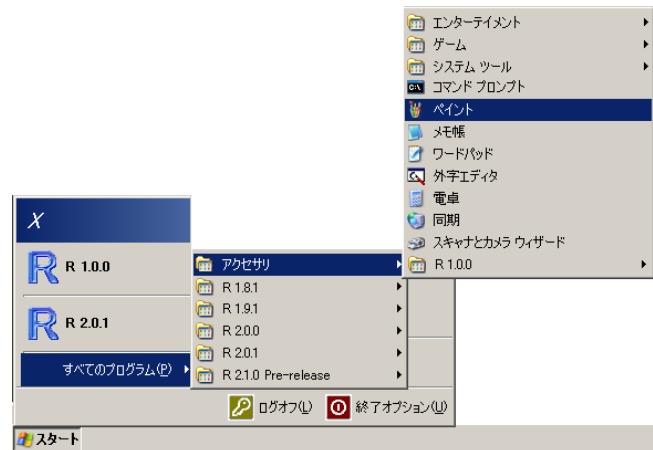
グラフの簡単な保存方法を紹介します。

(1) まず、描いたグラフのウィンドウを右クリックした後、メニューから[ビットマップにコピー]を選択します。これで、図がクリップボードにコピーされます（メモ帳やWORDなどで文字を「コピー」した後と同じ状態です）。

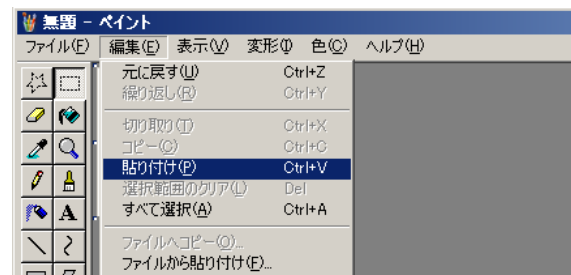


(2) 次に、適当なお絵かきソフトを起動します。筆者は貧乏なので Windows の「ペイント」を使います。

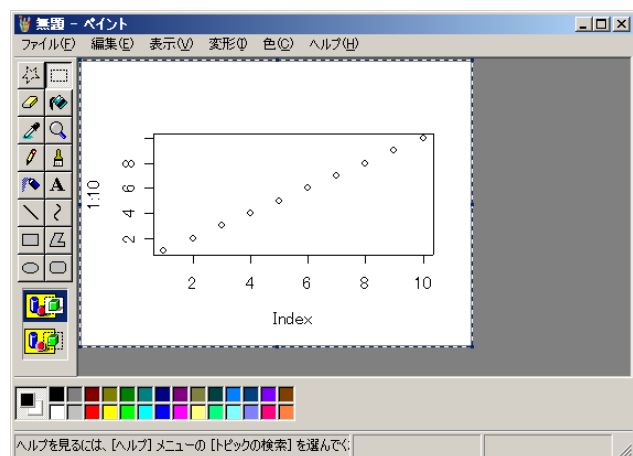
[スタート] メニューから [すべてのプログラム] → [アクセサリ] → [ペイント] を選択します。



(3) 「ペイント」を起動した後、メニューの [編集] から [貼り付け] を選択します。ショートカットがお好きな方は、[Ctrl]+[V]でも同様の作業が行えます。



(4) 見事にグラフが貼り付けられました！あとはメニューの [ファイル] から [上書き保存] を選択すれば、グラフをファイルに保存することが出来ます。



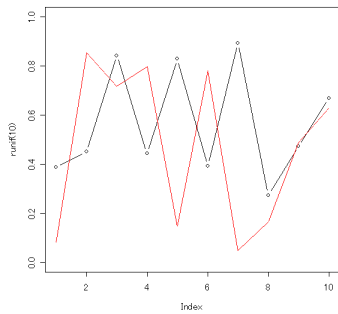
(参考1) 以上の方法は、グラフをお手軽に保存する方法です。Rのコマンド上でグラフをファイルに保存する方法は「The R Tips」を立ち読みして下さい。

(参考2) Windows の「ペイント」の詳しい使い方は、以下を参照下さい。

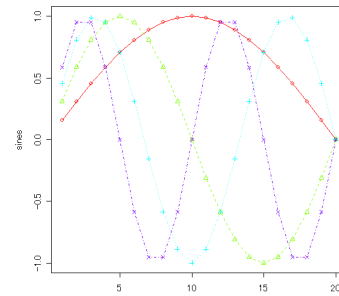
ペイント入門 : <http://www.geocities.co.jp/HeartLand-Icho/9090/win/paint/>

### 3.4節 グラフ集

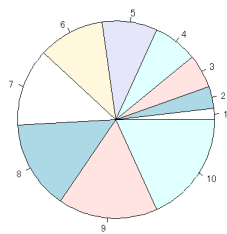
Rではこんなグラフが描けますよ, という紹介をします.



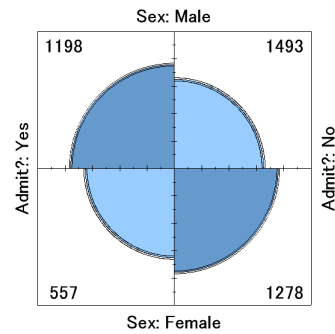
折れ線グラフ : `plot()`



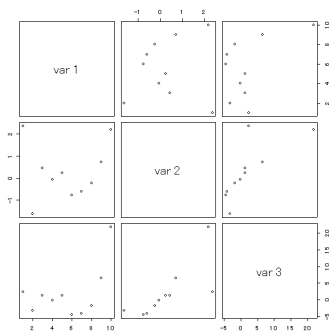
複数のグラフ : `matplot()`



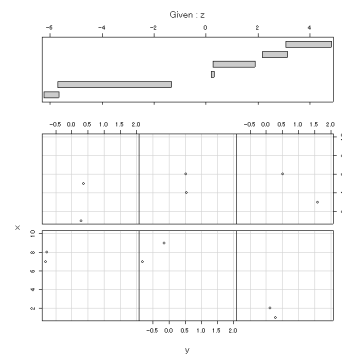
円グラフ : `pie()`



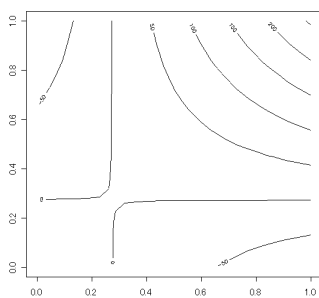
分割表データ : `fourfoldplot()`



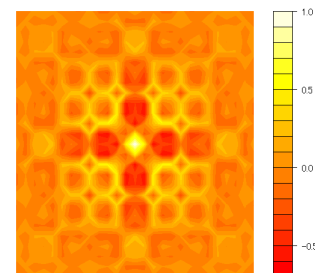
多変量データの図示 (1) : `pairs()`



多変量データの図示 (2) : `coplot()`

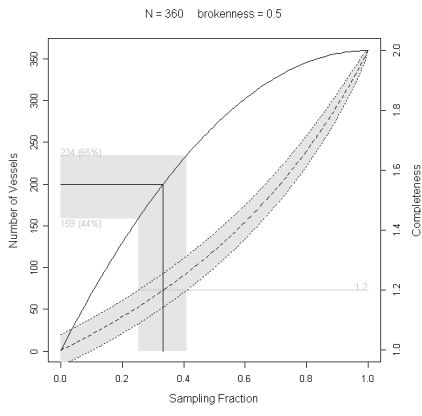


等高線図 (1) : `contour()`

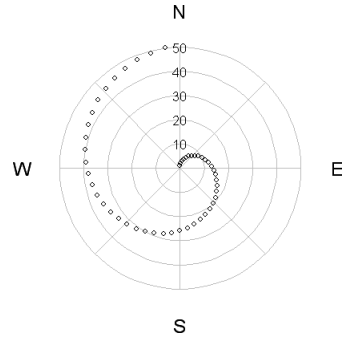


等高線図 (2) : `filled.contour()`

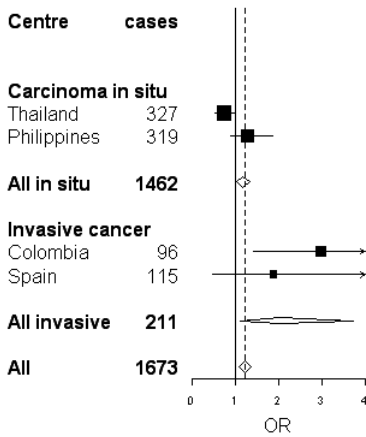
Paul Murrell's Home Page : <http://www.stat.auckland.ac.nz/~paul/> より引用.



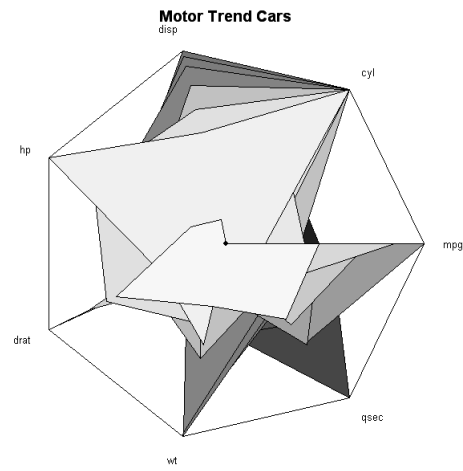
plotPars()



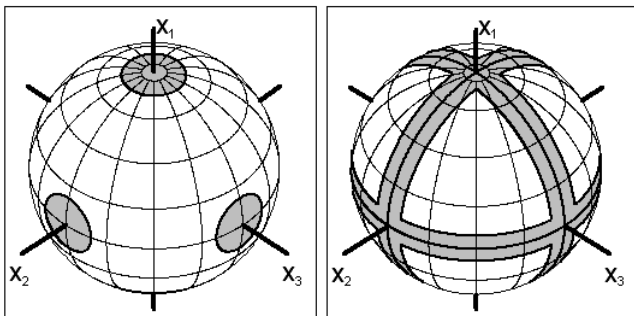
polar.plot()



An Example of "GRID" Package



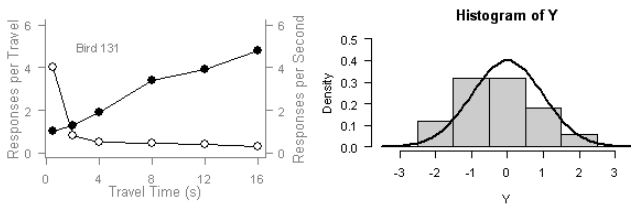
stars()



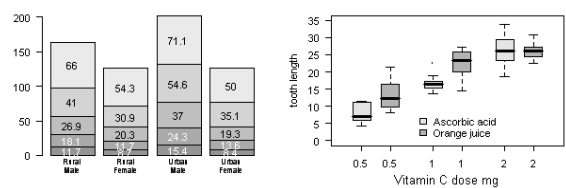
Lots of coordinate transformations being done "by hand"



An example of a one-off image drawn using the grid system.



simple example(1)

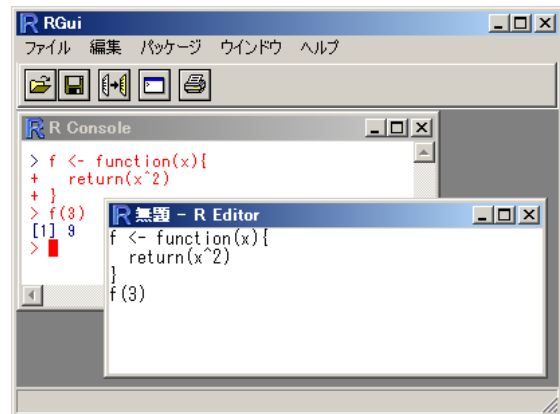


simple example(2)

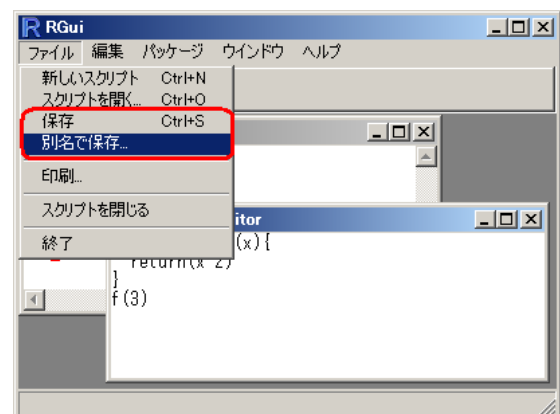
### 3.5節 R Editor の保存方法

R の計算式が書かれた「R Editor」をファイル（スクリプトファイル）へ保存する方法を紹介します。

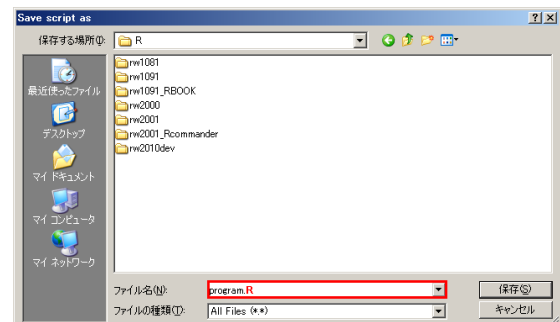
(1) まず、「R Editor」をマウスでクリックします  
(ウインドウが青くなります)。



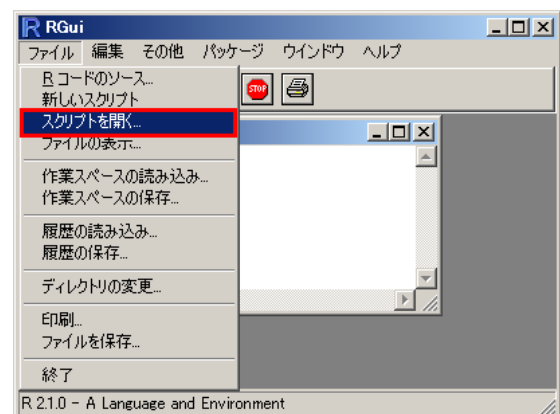
(2) 次に、[ファイル]の[保存]または[別名で保存]を選択します。



(3) すると、保存するファイルの名前を入力する画面（ダイアログ）が表示されますので、  
(ファイル名) **.R** または (ファイル名) **.r** を入力して[保存]をクリックします。右図では「program.R」という名前で保存しています。



(4) 保存したスクリプトファイルを開く場合は、  
[ファイル]の[スクリプトを開く]を選択します。  
すると、スクリプトを保存する際と同じダイアログが出ますので、保存してあったファイルを選択します。



## 3.6節 (参考) 統計解析入門

### ① データの準備

以下の2標本の体重データについて統計解析を行います。

|     |    |    |    |    |    |    |    |    |    |
|-----|----|----|----|----|----|----|----|----|----|
| 体重1 | 60 | 59 | 63 | 72 | 66 | 68 | 63 | 65 | 64 |
| 体重2 | 62 | 65 | 73 | 59 | 69 | 68 | 75 | 67 | 74 |

それぞれのデータの名前を weight1, weight2 として、関数 c() でデータを読み込みます。

```
> weight1 <- c(60, 55, 63, 72, 66, 68, 69, 65, 64)
> weight2 <- c(62, 65, 73, 59, 69, 68, 75, 67, 74)
```

### ② データの要約統計量

データを読み込ませた後は、データの要約統計量などを求めることができます。以下の表で、データ x, y について要約統計量を求める場合の書式を示します<sup>3</sup>。

| cor(x, y) | fivenum(x)  | IQR(x)   | max(x) | mean(x) | median(x)         |
|-----------|-------------|----------|--------|---------|-------------------|
| 相関係数      | 5数要約        | 4分位偏差    | 最大値    | 平均      | 中央値               |
| min(x)    | quantile(x) | range(x) | sd(x)  | sum(x)  | var(x), var(x, y) |
| 最小値       | quantile点   | 範囲       | 標準偏差   | 総和      | 不偏分散, 不偏共分散       |

以下にデータ weight1, weight2 についての要約統計量を求める例を示します。

```
> mean(weight2) # weight2 の平均
[1] 68
> var(weight1, weight2) # weight1 と weight2 の不偏共分散
[1] 1.25
```

### ③ 一標本 t 検定

データ weight1 について「母平均が 65 であるか否か」の一標本 t 検定を行う場合は関数 t.test() を使います。

```
> t.test(weight1, mu=65)
One Sample t-test
data: weight1 # データ名: weight1
t = -0.198, df = 8, p-value = 0.848 # t 値: -0.198, 自由度: 8, p 値: 0.848
alternative hypothesis: true mean is not equal to 65 # 対立仮説: 母平均は 65 ではない
95 percent confidence interval: # 95%信頼区間: [60.78508 68.54825]
60.78508 68.54825 #
sample estimates: # 標本推定:
mean of x # x(=weight)の標本平均 = 64.66667
64.66667 #
```

<sup>3</sup> var() は不偏分散を求める関数であり、標本分散を求める関数ではないことに注意して下さい。

#### ④ 二標本 t 検定

データ weight1, weight2 について「2 標本の平均の差が 0 であるか否か」の二標本 t 検定を行う場合は関数 `t.test()` を使います。等分散を仮定しない場合 (Welch の検定を行う場合) は, `var.equal=F` とします。

```
> t.test(weight1, weight2, var.equal=T)
      Two Sample t-test
data:  weight1 and weight2          # データ名 : weight1, weight2
t = -1.3453, df = 16, p-value = 0.1973  # t 値:-1.3453, 自由度:16, p 値:0.1973
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:      # 対立仮説 : 2 標本の平均の差は 0 でない
-8.585776  1.919110                 # 95%信頼区間 : [-8.585776  1.919110]
sample estimates:                   # 標本推定 :
mean of x mean of y                #  x(=weight) の標本平均 = 64.66667
 64.66667  68.00000                 #  y(=weight2) の標本平均 = 68.00000
```

#### ⑤ マン・ホイットニーの U 検定

データ weight1, weight2 について「2 標本の平均の差が 0 であるか否か」のノンパラメトリック検定 (マン・ホイットニーの U 検定) を行う場合は関数 `wilcox.test()` を使います。

```
> wilcox.test(weight1, weight2)
      Wilcoxon rank sum test with continuity correction
data:  weight1 and weight2          # データ名 : weight1, weight2
W = 26.5, p-value = 0.2325          # W 値 : 26.5, p 値:0.2325
alternative hypothesis: true mu is not equal to 0  # 対立仮説 : 母平均 mu は 0 ではない
```

#### ⑥ 等分散性の検定 (F 検定)

データ weight1, weight2 の分散が等しいか否かの検定は関数 `var.test()` を用います。

```
> var.test(weight1, weight2)
      F test to compare two variances
data:  weight1 and weight2          # データ名 : weight1, weight2
F = 0.8571, num df = 8, denom df = 8, p-value = 0.8327 # F 値:0.8571, 自由度:8, p 値:0.8327
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:      # 対立仮説 : 真の分散比は 1 ではない
0.1933437 3.7999370                 # 95%信頼区間 : [0.1933437 3.7999370]
sample estimates:  ratio of variances # 標本推定 : 分散比 = 0.8571429
                                0.8571429
```

#### ⑦ 検定関数一覧 (抜粋)

最後に, R に用意されている検定関数を紹介します。

|                              |                             |                             |                            |                            |
|------------------------------|-----------------------------|-----------------------------|----------------------------|----------------------------|
| <code>bartlett.test()</code> | <code>binom.test()</code>   | <code>chisq.test()</code>   | <code>cor.test()</code>    | <code>fisher.test()</code> |
| Bartlett 検定                  | 二項検定                        | $\chi^2$ 検定                 | 相関係数の検定                    | Fisher の正確検定               |
| <code>friedman.test()</code> | <code>kruskal.test()</code> | <code>mcnemar.test()</code> | <code>oneway.test()</code> | <code>prop.test()</code>   |
| Friedman 検定                  | Kruskal-Wallis 検定           | McNemar 検定                  | 一元配置分散分析                   | 比率の同一性検定                   |

## 4章 落穂ひろい

最後に、書き忘れそうになった事項を紹介します。

### 4.1節 Rによる関数定義の方法

Rには特別な機能を果たす命令を関数という形で呼び出すことができます。例えば関数 `date()` を実行すると、現在の日付が出力されます。

```
> date()
[1] "Wed Mar 09 09:15:12 2005"
```

対数を計算する関数 `log(x)` は「 $x$  の値の対数を計算する関数」ですが、関数に  $x$  の値を指定する場合には引数という形で関数に与えます。

```
> log(10)
[1] 2.302585
```

引数に式や関数を指定すれば、それを評価した値が引数として使われます。また、引数が 2 個以上ある場合はコンマ ( , ) で区切って並べます。

```
> log(10, base=10)           # 底を指定する引数 base に 10 を指定して対数を計算
[1] 1
```

自分で関数を定義する手順は以下の通りです。

- (1) 関数名を決めます。
- (2) 入力する変数の個数と種類を指定します。
- (3) 計算手順を 1 行ずつ記述し、最後に関数 `return()` で計算結果を出力する。

関数 `date()` のような、入力値が無い関数の関数定義の例を示します。以下の関数 `myfunc01()` は、実行すると 10 を出力します。

```
> myfunc01 <- function() {           #
+   return(10)                       # 関数定義部分
+ }                                   #

> myfunc01()                         # 関数を実行する
[1] 10
```

入力値が有る関数の関数定義の例を示します。以下の関数 `myfunc02(x)` は、実行すると入力した値を出力します。

```
> myfunc02 <- function(x) {           #
+   return(x)                         # 関数定義部分
+ }                                     #

> myfunc02(2)                          # 関数を実行する
[1] 2
```

入力値は 2 個以上あっても構いません。例として、2 つの数値を入れると入力した数値の積が結果として返ってくる関数 `mytimes(x, y)` を定義します。

```
> mytimes <- function(x, y) {         #
+   return(x*y)                       # 関数定義部分
+ }                                     #

> mytimes(2, 5)                        # 関数を実行する
[1] 10
```

関数の中では行を複数書いても構いません。例えば、数値を入力すると数値の絶対値が結果として返ってくる関数 `myabs(x)` を定義します。

```
> myabs <- function(x) {
+   y <- abs(x)
+   return(y)
+ }
```

以下に練習問題を 2 問ほど載せます。

- (1) 数値を入れると入力した数値が 3 乗されたものが結果として返ってくる関数 `mytriple(x)` を定義して下さい。
- (2) 2 つの数値  $x, y$  を入れると、 $x$  の  $y$  乗が結果として返ってくる関数 `mypower(x, y)` を定義して下さい。

```
> mytriple(2)                          # 2 の 3 乗を計算する
[1] 8
> mypower(2, 4)                        # 2 の 4 乗を計算する
[1] 16
```



## 4.2節 その他

### ① Mac OS X 10.3 をお使いの方へ

配布しました CD-ROM は、Mac OS X 10.3 では読み取ることが出来ない場合があります。以下に、R 本体とパッケージのダウンロード先をお示し致します。

**R 2.0.1.dmg (Mac OS X 用 R 本体) :**

<http://cran.md.tsukuba.ac.jp/bin/macosx/R-2.0.1.dmg>

**パッケージのダウンロード先 :**

<http://cran.md.tsukuba.ac.jp/src/contrib/PACKAGES.html>

### ② R 関係のホームページ

**The R Project :** <http://www.r-project.org/>

R の総本山です。R 本体が置かれているサイト集 (CRAN) や、R に関するドキュメントなどが置かれています。

**中間 栄治さんのホームページ :** <http://r.nakama.ne.jp/>

Windows 版と Vine / Linux 版の日本語化された R のバイナリやソースが置かれています。

**間瀬 茂 先生のホームページ :** <http://www.is.titech.ac.jp/~mase/R.html>

東京工業大学の間瀬先生による『R Tips』や、マニュアルの日本語訳、ヘルプの日本語訳が置かれています。

**R による統計処理 :** <http://aoki2.si.gunma-u.ac.jp/R/>

群馬大学の青木 繁伸 先生のホームページです。膨大な数の統計解析用プログラム、そこに付随する明快な統計手法の解説、分かりやすい例が載っています。Mac 版 R のインストール方法などもあります。

**関西大学・荒木 孝治先生のホームページ :** <http://www.ec.kansai-u.ac.jp/user/arakit/>

R の GUI パッケージ「Rcmdr0.9-18」のメニューおよび本体が日本語化されたものが公開されています。

**滋賀大学・熊澤 吉起先生の「R のページ」:** <http://www.biwako.shiga-u.ac.jp/sensei/kumazawa/R/index.html>

R の Install 方法とデータセットの紹介、例題や Tips が載っています。

**RjpWiki :** <http://www.okada.jp.org/RWiki/index.php?RjpWiki>

東京大学の岡田 昌史 先生が運営されているサイトです。R に関する日本語による情報交換を目的とした Wiki で、R に関する情報が日々集められています。

**R・HTML マニュアル検索 :** <http://www.okada.jp.org/cgi-bin/Rsearch/namazu.cgi/>

岡田 先生が運営されているサイトで、R の html マニュアルを検索することが出来ます。

**統計解析フリーソフト R - Tips :** <http://cse.naro.affrc.go.jp/takezawa/r-tips/r2.html>

昔、私が R についてまとめたホームページです。間違いだらけなので、只今改訂作業中です…。

### ③ R について書かれた書籍 (一部)

**『工学のためのデータサイエンス入門』 間瀬 茂 他 著 (数理工学社)**

R で統計解析をやる際のバイブルです。推定と検定、回帰分析 (+分散分析)、非線形回帰、シミュレーションについて書かれており、解説と R ソースでの例が絶妙なバランスで紹介されています。

**『The R Book』 岡田 昌史 他 著 (九天社)**

豪華執筆陣による R のアプリケーションが書かれています (データ入出力、クロス表とログリニア分析、時系列解析、生存時間解析、画像処理、地理情報システム GIS、ゲノムデータ解析…、他多数)。

**『The R Tips』 舟尾 暢男 著 (九天社)**

ウェブ版「R-Tips」が間違いだらけだったので、反省の意味も込めて執筆したつもりでしたが、またまた間違いだらけに仕上がってしまいました…。一応、R 入門者の方用の説明が載っています。

サポート頁はこちら : <http://cwoweb2.bai.ne.jp/~jgb11101/files/bbs/support.html>