統計解析フリーソフト R 入門

～ R によるデータの扱い方 ～

【 目 次 】

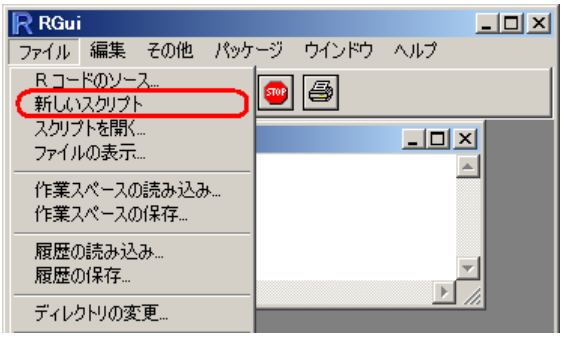
0章 前回のおさらい

ここでは，Windows 版 R を使って，R による簡単な計算を行う方法を復習します．

0.1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．

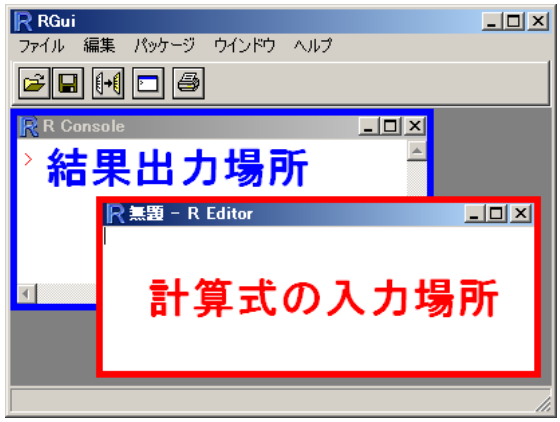
(1) まずは R を起動します．すると，「R Console」というウインドウが表れます．次に，メニューの［ファイル］→［新しいスクリプト］を選択します．



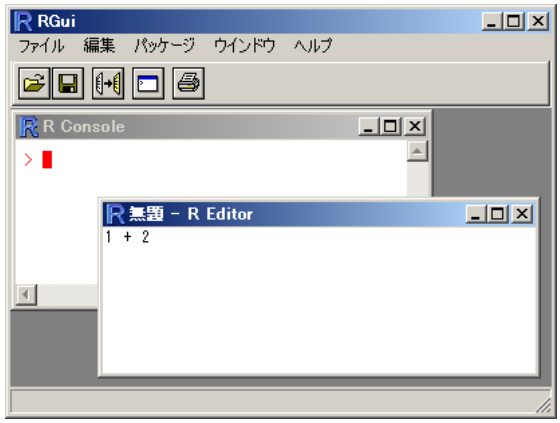
(2) すると，「無題 － R Editor」という名前のウインドウが表れます．これら 2 つのウインドウは

- **R Console** ： 結果出力ウインドウ
- **R Editor** ： 計算式の入力ウインドウ

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[1]．

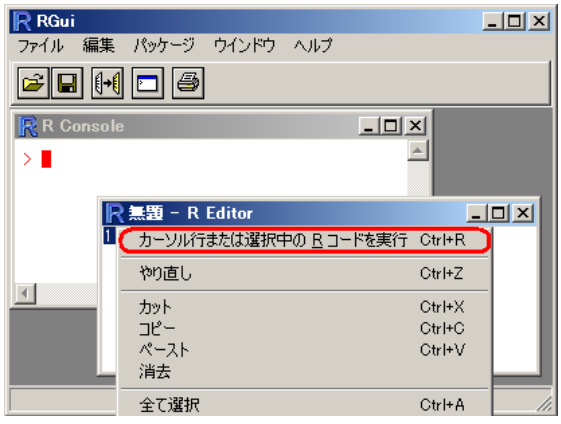


(3) 試しに「1+2」という簡単な計算を行ってみましょう．まず，「R Editor」に計算式「1+2」を入力します．



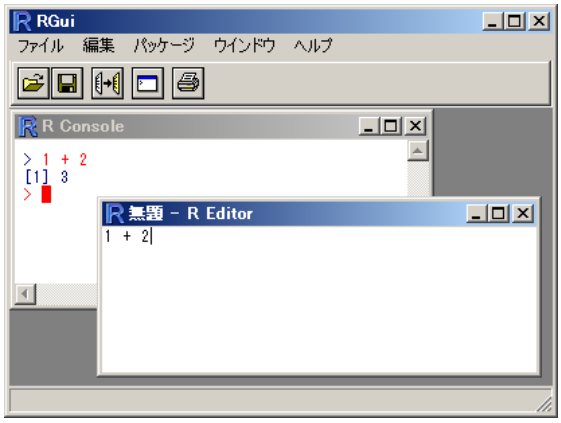
(4) 計算式「1+2」を入力した後，計算式を実行する場合は，計算したい式をマウスで範囲選択した後，右クリックから［カーソル行または選択中の R コードを実行］を選択します．

（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，［Ctrl］+［R］を押しても同様のことが行えます．
Mac 版 R ならば［リンゴ］＋［Enter］を押します．



1 「R Console」に計算式を接入力しても計算することが出来ます．

(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．

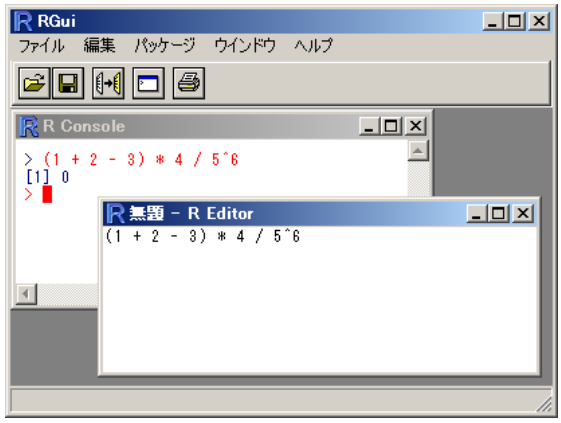


0.2節　電卓としての使い方

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．

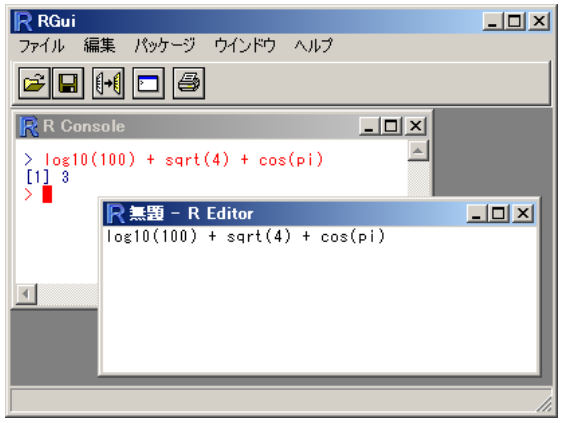
+	−	*	／	^
加算	減算	乗算	除算	累乗



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)
sin**x**	cos**x**	tan**x**	$\log_e$**x**	$\log_{10}$**x**
sinh(x)	cosh(x)	tanh(x)	exp(x)	sqrt(x)
sinh**x**	cosh**x**	tanh**x**	$e^{\mathbf{x}}$	$\mathbf{x}^{1/2}$
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

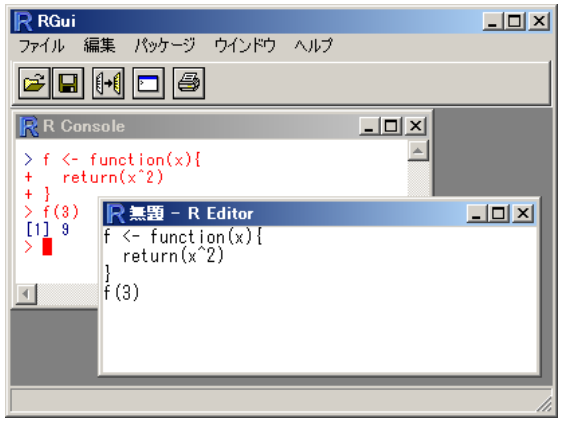


右の計算では円周率 pi を使っています．

(3) 自分で関数を定義することも出来ます．試しに，関数 $f(x)=x^2$ 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)
}
```

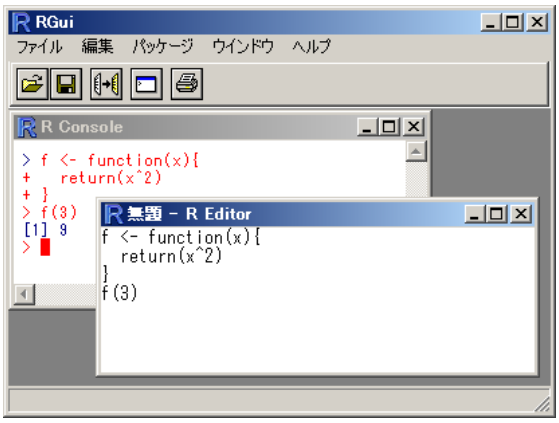
とします．右図では $f(3) = 3^2 = 9$ を計算しています．



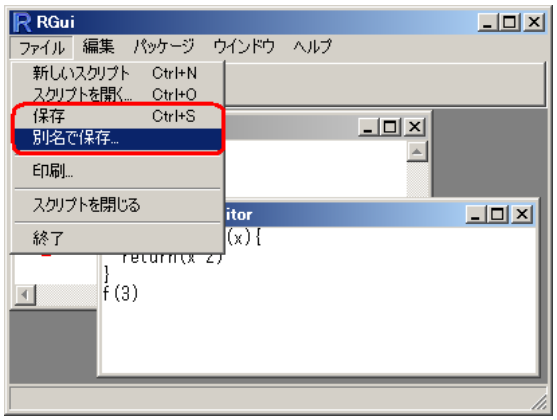
0.3節　R Editor の保存方法

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.

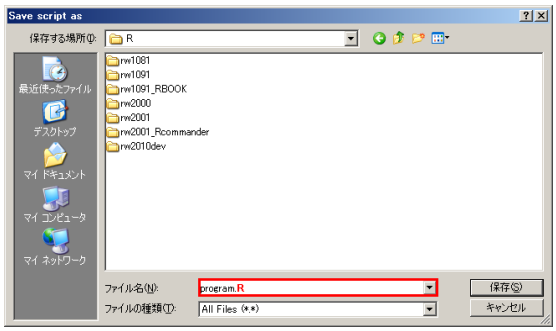
(1) まず,「R Editor」をマウスでクリックします（ウインドウが青くなります）.



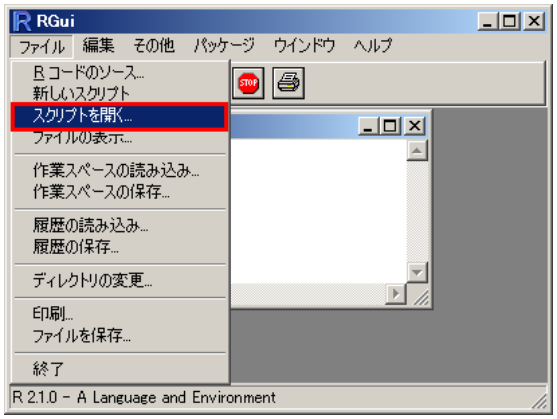
(2) 次に，[ファイル]の[保存]または[別名で保存]を選択します.



(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.



1章 ベクトルとデータの扱い方

1.1節　ベクトル（一行のデータ）を作成する

「ベクトル」と云われると，何だか難しいイメージを持ってしまいますが，R の「ベクトル」とは単に「数値や文字を一つにまとめたもの」のことです．

① 手入力でベクトルを作成する

以下の 2 標本の体重データについて考えてみます．

x	60	59	63	72	66	68	63	65	64
y	62	65	73	59	69	68	75	67	74

それぞれのデータの名前を x，y として，**関数 c(データ 1, データ 2, ・・・)** でデータを読み込みます．

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

これで x と y のそれぞれにデータが読み込まれました．入っているデータを確かめる際は，データ名をそのまま入力します．以下に x と入力した際の結果出力画面を示します．

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

前述の数学関数にデータ名を放り込むと，各要素それぞれに数学関数が適用されます．

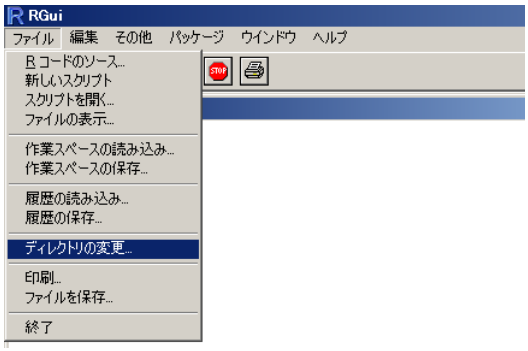
```
> sqrt(x)
[1] 7.745967 7.416198 7.937254 8.485281 8.124038 8.246211 8.306624 8.062258 8.000000
```

② ファイルからベクトルを作成する（作業ディレクトリの変更）

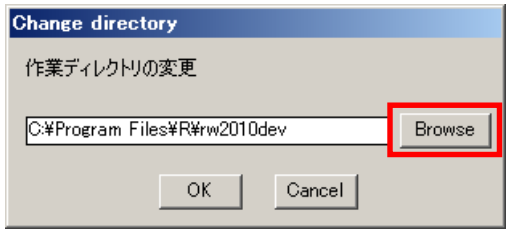
【Windows 版 R の場合】

(1) ファイルからデータを読み込む場合，作業ディレクトリを，データがあるディレクトリ（フォルダ）に変更する必要があります．

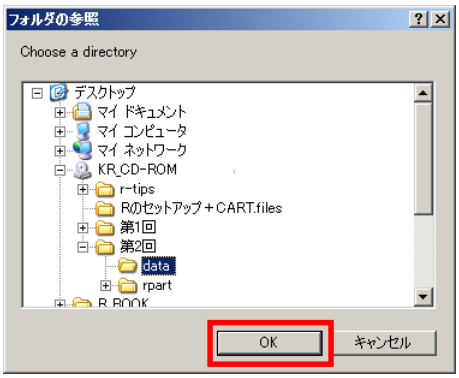
メニューの[ファイル]→[ディレクトリの変更]を選択します．



(2) [Change directory]という名前のウインドウが出ますので，[Browse]をクリックします．



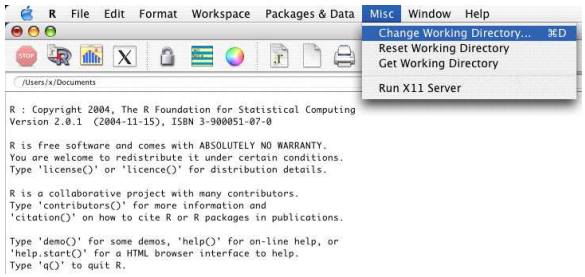
(3) すると，[フォルダの参照]という名前のウインドウが出ます．ここで，データがあるディレクトリ（フォルダ）を指定して，[OK]をクリックします．



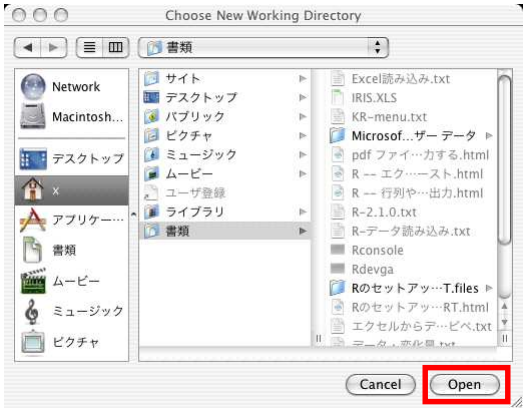
(4) これで，作業ディレクトリを指定することが出来ましたので，[OK]をクリックします．

【Mac OS X 版 R の場合】

(1) Mac OS X 版 R の場合でも，ファイルからデータを読み込む場合は，作業ディレクトリをデータがあるディレクトリ（フォルダ）に変更する必要があります．まず，メニューから
[Misc]→[Change Working Directory…]
を選択します．



(2) すると，[Choose New Working Directory]という名前のウインドウが出ます．ここで，データがあるディレクトリ（フォルダ）を指定して，[OK]をクリックします．



【コマンドでディレクトリを変更する場合】

R のコマンドで作業ディレクトリを変更する場合，Windows 版と Mac OS X 版のどちらでも使える **関数 setwd("ファイルのパス")** を使います[2]．

```
> setwd("E:/第 2 回/data")
```

現在の作業ディレクトリを表示する場合は関数 **getwd()** を使います．

```
> getwd()
[1] "E:/第 2 回/data"
```

[2] 【Windows 版 R を使われている方】¥マークは使わない方が良いでしょう．もし¥マークを使う場合は，¥¥を重ねてください（例：setwd("E:¥¥第 2 回¥¥data")）．

③ ファイルからベクトルを作成する（データの読み込み）

(1) 以下のようなテキストファイル vector1.txt があったとします.

```
60 59 63 72 66 68 63 65 64
```

このようなデータをベクトルとして読み込む場合は，関数 **scan("ファイルのパス")** を用います.

```
> x <- scan("vector1.txt")
Read 9 items
```

上では x という名前の変数にデータを代入しました．正しくデータが入力されたか確認します.

```
> x
[1] 60 59 63 72 66 68 63 65 64
```

(2) 1 行目に変数名が入っているようなテキストファイル vector2.txt があったとします.

```
x
60 59 63 72 66 68 63 65 64
```

関数 **scan("ファイルのパス", skip=読み飛ばす行数)** を用いることで，読み飛ばす行を指定することが出来ます．以下では 1 行目を読み飛ばして 2 行目から読み込んでいます.

```
> x <- scan("vector2.txt", skip=1)
Read 9 items
> x
[1] 60 59 63 72 66 68 63 65 64
```

(2) データがコンマで区切られているようなテキストファイル vector2.txt があったとします.

```
60,59,63,72,66,68,63,65,64
```

関数 **scan("ファイルのパス", sep="区切り文字")** を用いることで，データとデータの間の文字を指定することが出来ます．以下では区切り文字を , としてデータを読み込んでいます.

```
> x <- scan("vector3.txt", sep=",")
Read 9 items
> x
[1] 60 59 63 72 66 68 63 65 64
```

④ ベクトル（データ）の要約統計量

ベクトル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出来ます．以下の表で，データ x，y について要約統計量を求めるための関数一覧を示します[3]．

sum()	mean()	median()	var()	cor()	max()
総和	平均	中央値	不偏分散	相関係数	最大値
range()	cumsum()	prod()	diff()	rev()	min()
範囲	累積和	総積	前進差分	要素を逆順	最小値
pmax()	pmin()	sort()	rank()	order()	
列最大値	列最小値	昇順整列	各要素の順位	各要素の元の位置	

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(y)              # y の平均
[1] 68
```

⑤ 検定

R に用意されている検定関数を紹介します．

t.test()		wilcox.test()		var.test()
1 標本 t 検定＆2 標本 t 検定		マン・ホイットニーの U 検定		F 検定
bartlett.test()	binom.test()	chisq.test()	cor.test()	fisher.test()
Bartlett 検定	二項検定	χ^2 検定	相関係数の検定	Fisher の正確検定
friedman.test()	kruskal.test()	mcnemar.test()	oneway.test()	prop.test()
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

データ x 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(x, mu=65)
        One Sample t-test
data:  x                                          # データ名：x
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                                         #
```

[3] 「The R Tips」（舟尾 暢男 著，九天社）より引用．

データ x, y 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.

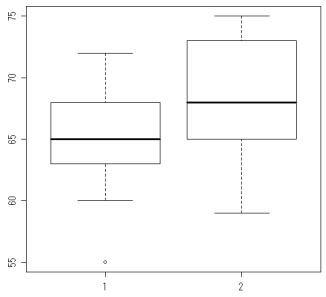
```
> t.test(x, y, var.equal=T)
        Two Sample t-test
data:  x and y                                          # データ名：x, y
t = -1.3453, df = 16, p-value = 0.1973                  # t 値:-1.3453, 自由度:16, p 値:0.1973
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence 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
```

⑥ グラフ

グラフ出力に関する詳しい説明は第 1 回の資料をご参照下さい.

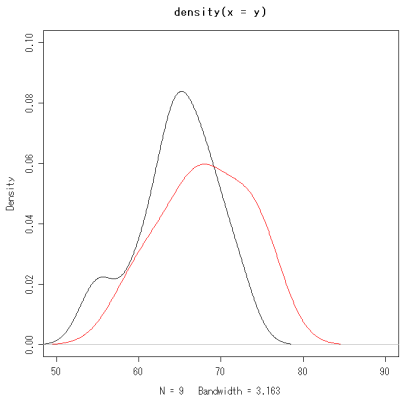
(1) x と y の箱ひげ図を描いてみます.

```
> boxplot(x, y)
```



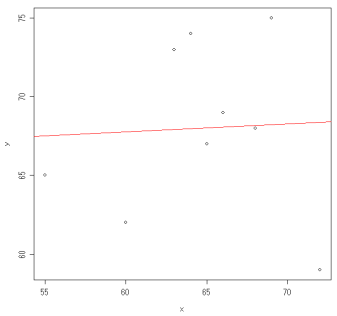
(2) x と y の密度推定のグラフを重ね描きしてみます. グラフを重ね描きする場合は, プロットとプロットの間に **par(new=T)** とします.

```
> x <- c(60, 55, 63, 72, 66, 68, 69, 65, 64)
> y <- c(62, 65, 73, 59, 69, 68, 75, 67, 74)
> plot(density(x),xlim=c(50,90),ylim=c(0,0.1),ann=F)
> par(new=T)
> plot(density(y),xlim=c(50,90),ylim=c(0,0.1),col="red")
```



(3) 回帰直線を描くことも出来ます.

```
> plot(x, y)
> result <- lm(y ~ x)
> abline(result, col="red")
```



1.2節　複数のデータ（データフレーム）の読み込み方法

R では，複数のベクトルデータを 1 つにまとめて扱うことが出来ます．1 つにまとめたデータのことを「データフレーム」といいます．データフレームには以下のような特長があります．

- 1 つのベクトルデータを列（縦）ベクトルとして，横に並べます．
- 各ベクトルにはラベル（データの名前）が付きます．この名前でデータを取り出します．
- 数値ベクトルと文字型ベクトル等，異なった型のベクトルをまとめることが出来ます．

① 手入力でベクトルを作成する

(1) それぞれのデータの名前を x，y として，
関数 **data.frame(ラベル 1=変数名 1, ・・・)**
でデータを読み込みます．

```
> x <- c(60,59,63,72,66,68,63,65,64)
> y <- c(62,65,73,59,69,68,75,67,74)
> mydata <- data.frame(X=x, Y=y)
```

これで mydata という名前の変数にデータが読み込まれました．入っているデータを確かめる際は，データ名をそのまま入力します．右に mydata と入力した際の結果出力画面を示します．

```
> mydata
   X  Y
1 60 62
2 59 65
3 63 73
4 72 59
5 66 69
6 68 68
7 63 75
8 65 67
9 64 74
```

(2) データを取り出すときは　**データフレーム名＄ラベル名**　で，データを取り出します．

```
> mydata$X
[1] 60 59 63 72 66 68 63 65 64
```

取り出したデータは，ベクトルと同じように扱うことが出来，ベクトルの要約統計量を求めたり，検定やプロットを行うことが出来ます．

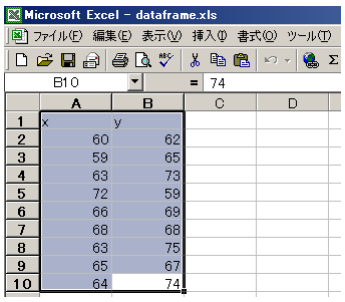
```
> mean(mydata$X)
[1] 64.44444
```

② ファイルからデータフレームを作成する（EXCEL シートからコピー＆ペースト）

EXCEL ファイルからデータをコピー＆ペーストして R にデータを読み込ませる方法を紹介します．

【Windows 版 R の場合】

(1) まず，EXCEL データを開き，作成したい部分のデータをコピーします．このとき，変数名も一緒にコピーします．



(2) 次に，R に戻って以下を入力します．

```
> mydata <- read.delim("clipboard")
```

これでデータが mydata に読み込まれます．

もし，変数名を一緒にコピーしない場合は，以下のように入力します．

```
> mydata <- read.delim("clipboard",header=F)
```

右に示すとおり，適当な名前が振られます．

```
> mydata
  V1 V2
1 60 62
2 59 65
3 63 73
4 72 59
5 66 69
6 68 68
7 63 75
8 65 67
9 64 74
```

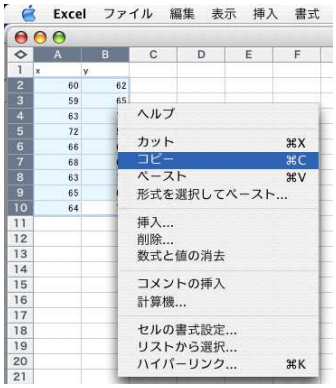
【Mac OS X 版 R の場合】

(1) まず，以下の関数を定義します．

```
excel.mac <- function(...) {
  args <- c(...)
  temp <- matrix(scan(""), byrow=TRUE, ncol=length(args))
  data <- data.frame(temp)
  colnames(data) <- args
  return(data)
}
```

(2) 次に，作成したい部分のデータを**変数名は除いて**コピーします．

(3) 関数 excel.mac("変数名 1","変数名 2",・・・) を実行します．すると，以下のような画面になります．



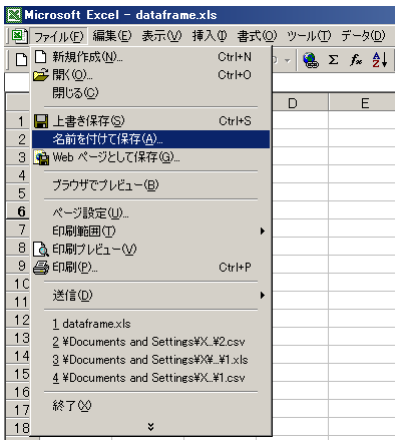

```
> excel.mac("X","Y")
1:
```

(4) 画面にデータをペーストします．データが読みとられた後，[Enter]キーを 2 回ほど押すとデータ入力が終了します．

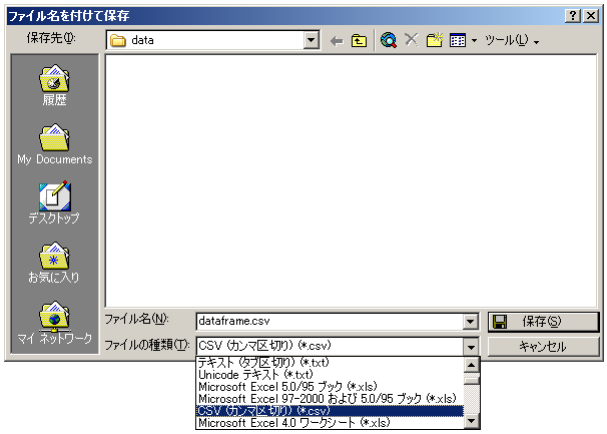
③ ファイルからデータフレームを作成する（EXCEL ファイル→テキストファイル→R）

EXCEL ファイルからデータをコピー＆ペーストして R にデータを読み込ませる方法を紹介します．
（以下では Windows 版での方法を紹介しますが，Mac OS X 版もほぼ同様です）

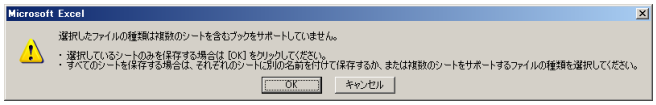
(1) まず，EXCEL ファイルを開き，読み込みたいデータが含まれているシートを表示します．次に，メニューの[ファイル]→[名前を付けて保存]を選択します．

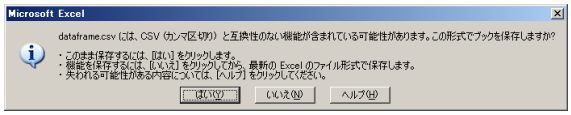


(2) 保存する際，[ファイルの種類]を
　csv（カンマ区切り）（*.csv）
とします．



(3) 2 つほど警告ウインドウが出ますが，気にせず[OK]をクリックします．



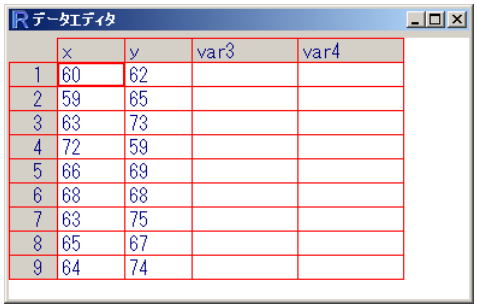


(4) 最後に，関数 read.csv("ファイル名") で読み込みます．

```
> mydata <- read.csv("dataframe.csv")
```

(5) おまけですが，関数 edit() でデータをセル形式で表示・編集することが出来ます．閲覧・編集が終了したら，ウインドウの[×]をクリックします．

```
> edit(mydata)
```



	x	y	var3	var4
1	60	62		
2	59	65		
3	63	73		
4	72	59		
5	66	69		
6	68	68		
7	63	75		
8	65	67		
9	64	74		

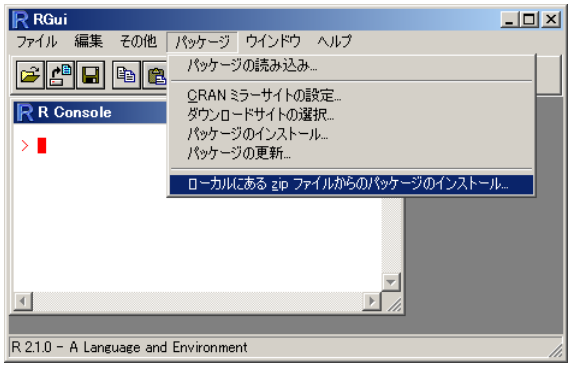
2章 CART・回帰樹

パッケージのインストール例，読み込んだデータの使用例として，R で CART（回帰樹）を実行します[4].

2.1節　rpart のインストール方法

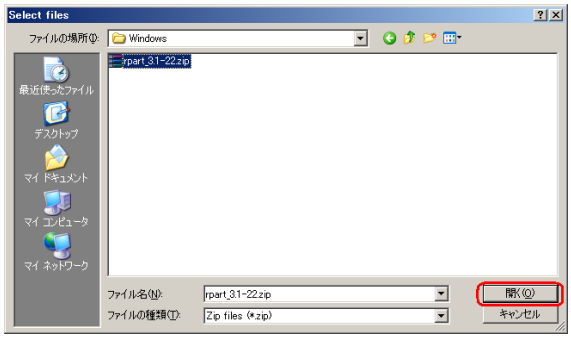
(1) まず，以下のサイトから CART 用パッケージ「rpart」の .zip ファイルをダウンロードします.

http://cran.md.tsukuba.ac.jp/src/contrib/Descriptions/rpart.html



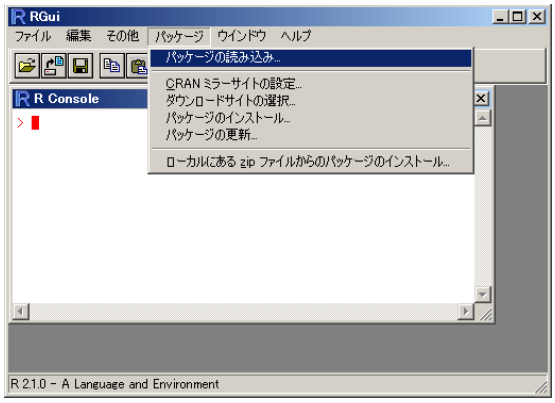
次に，［パッケージ］の［ローカルにある zip ファイルのパッケージのインストール］をクリックします[5].

(2) 先程ダウンロードしたファイル「rpart_3.1-xx.zip」をマウスで選択し，「開く」をクリックします．これでパッケージのインストールは完了です.



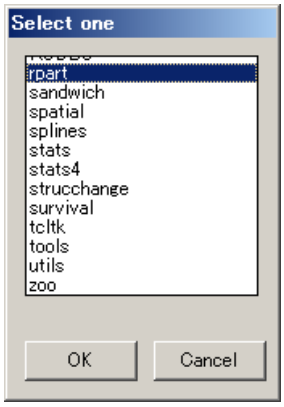
(2') 自動的にインストールすることも出来ます.

```
install.packages("rpart")
```



(3) rpart を開く場合は，［パッケージ］の［パッケージを読み込み］をクリックします.

(4) すると，どのパッケージを読み込むかを選ぶ画面（ダイアログ）が出ますので，下の方にある「rpart」を選択します．これで rpart パッケージが読み込まれました.



rpart パッケージにどのような関数が用意されているかを確かめる際は以下のように入力します.

```
help(package="rpart")
```

[4] Windows 版 R 用のパッケージです.
[5] 「パッケージの読み込み」を選択して，メニューから "rpart" を選択していただいても構いません.

2.2節　関数 rpart() の使用方法

(1) フィッシャーのアヤメの分類データ「iris.xls」で CART を行ってみましょう．アヤメのがくの長さ（Sepal.Length），がくの幅（Sepal.Width），花弁の長さ（Petal.Length），花弁の幅（Petal.Width）を説明変数として，フィッシャーは，アヤメの種類（Species）を判別しようとしました．まず，「iris.xls」を開き，データをコピーした後，R から次のように入力します．

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
・・・	・・・	・・・	・・・	・・・

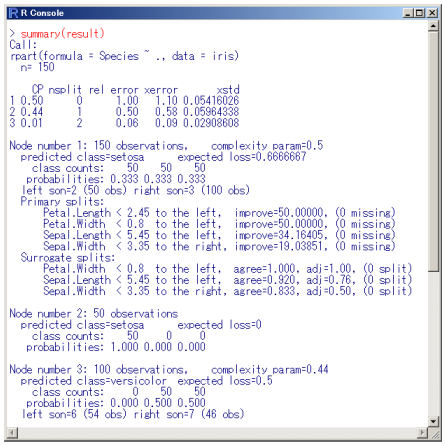
```
data <- read.delim("clipboard")
```

(2)変数 data にデータが格納されました．それではいきなり R で解析しましょう．R で CART を行うには関数 rpart() を使います．

```
result <- rpart(Species ~ ., data=data)
```

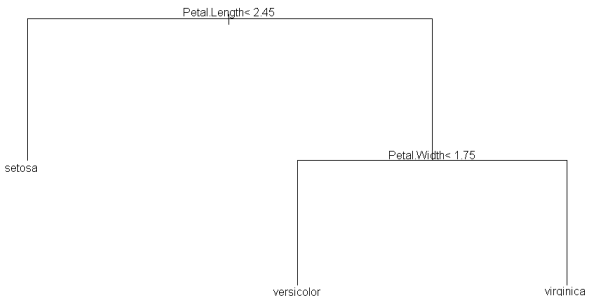
結果の要約を見る際は，以下のように入力します．

```
summary(result)
```



(3) 出力結果が英語であるせいか，あまり要約されていない気がします・・・．よく分からないので，分類結果を図で表示してみましょう．

```
plot(result)
text(result)
```



出力図は以下のようになります（プロット図を大きくしないと文字が欠けます・・・）．結果は

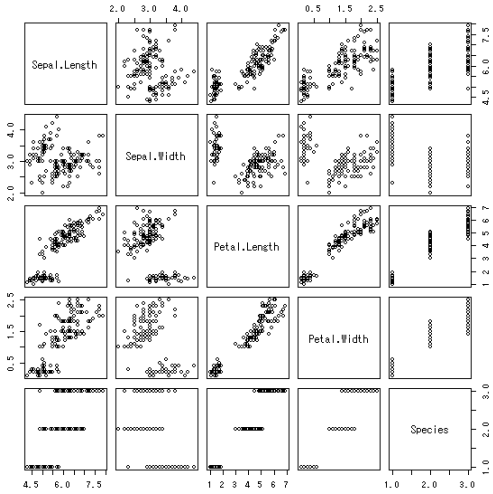
- 花弁の長さ（Petal.Length）が 2.45 未満ならば setosa
- 花弁の長さ(Petal.Length)が 2.45 以上で花弁の幅(Petal.Width)が 1.75 未満ならば versicolor
- 花弁の長さ(Petal.Length)が 2.45 以上で花弁の幅(Petal.Width)が 1.75 以上ならば virginica

であることを表しています．この木を見てから，先程出力された要約を読むと，何かが分かった気になれます．この回帰ツリーは単純明快な木になっていますが，もしも生成された木が複雑すぎて（枝が多すぎて）解釈が難しい場合は，剪定（pruning）する関数 prune() を用いて，木の枝を減らすことも出来ます．

2.3節　おまけ

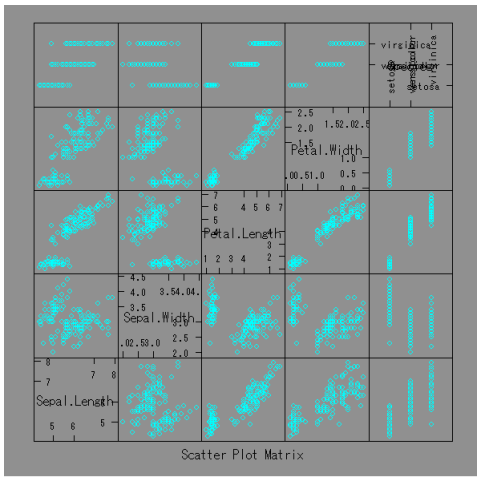
(1) 木だけでは寂しいので，データ iris の変数間のプロットをしてみましょう.

```
pairs(data)
```



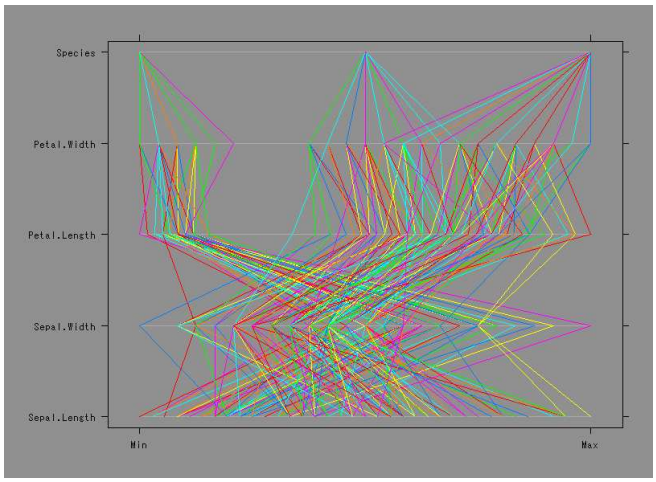
(2) 似たような図をもう一枚.

```
library(lattice)
splom(iris)
```



(3) こんなのもあります.

```
library(lattice)
parallel(iris)
```



（参考文献）

『R による統計処理』（群馬大学・青木先生）：http://aoki2.si.gunma-u.ac.jp/R/

『統計処理ソフトウェア R についての Tips』（群馬大学・中澤先生）：
http://phi.med.gunma-u.ac.jp/swtips/R.html

『The R Tips』 舟尾 暢男 著（九天社）：
http://cwoweb2.bai.ne.jp/~jgb11101/files/bbs/support.html