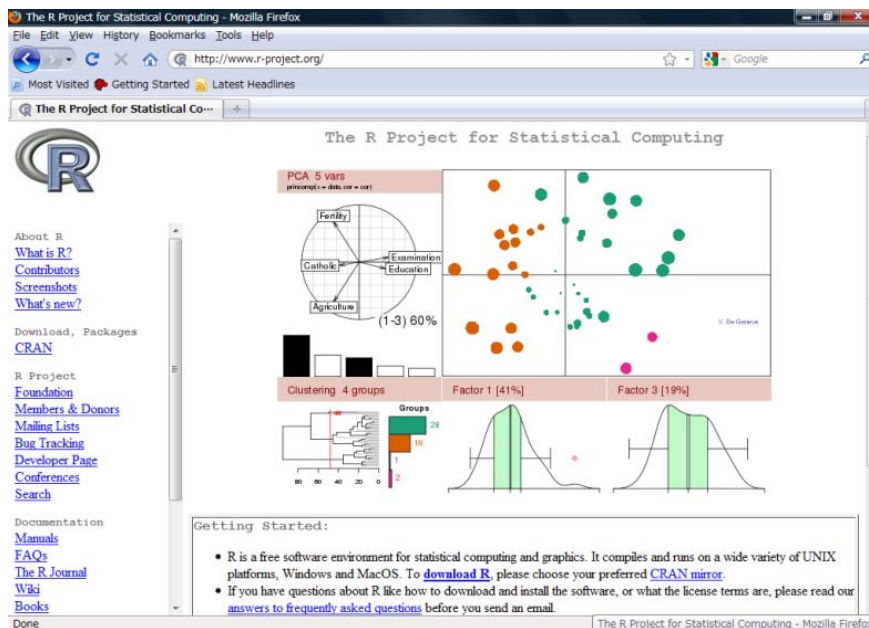


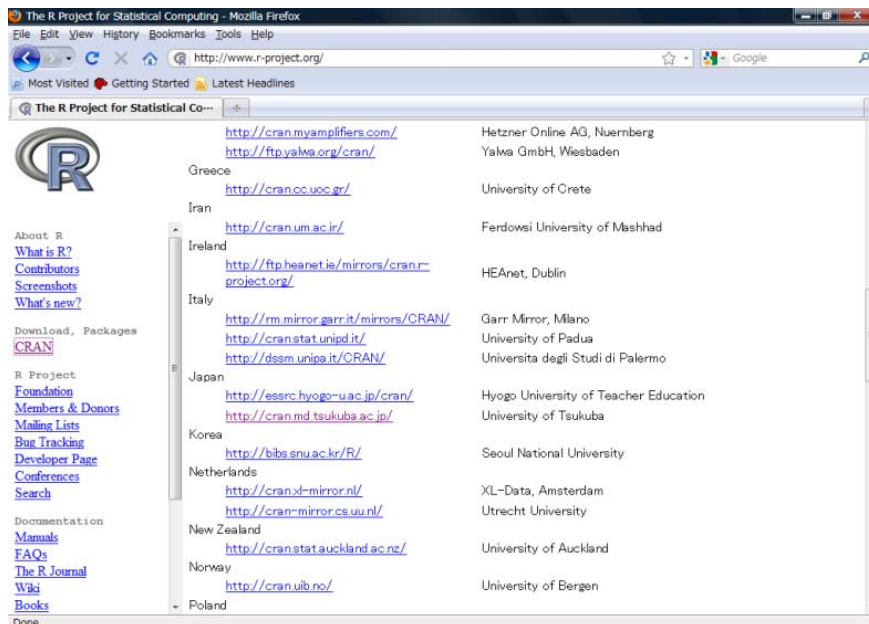
Rのインストール

【注意】 管理者または管理者権限を持ったユーザーとしてパソコンを起動（ログイン）している必要がある。

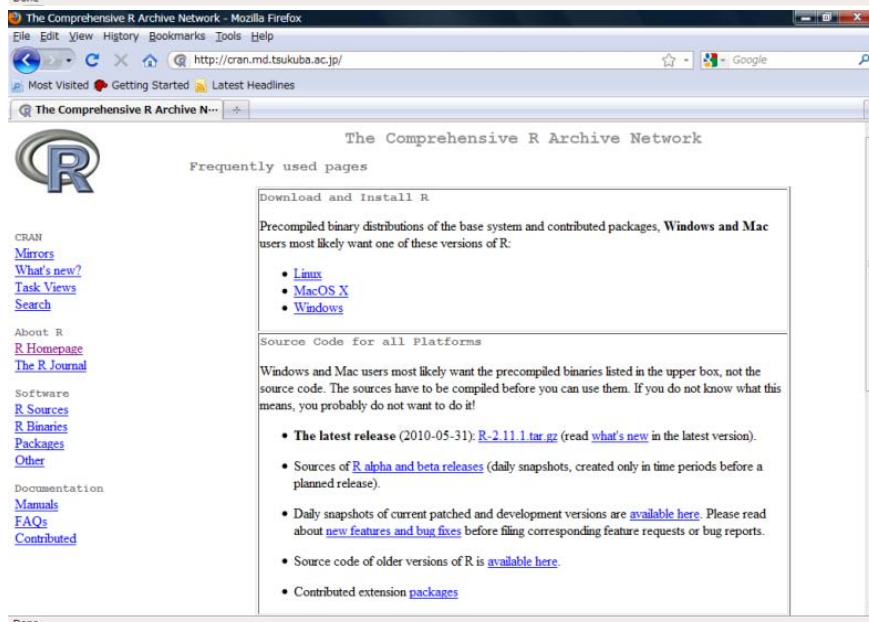


<http://www.r-project.org/>
を開く。

CRAN をクリックして、最寄りのミラーサイトを選ぶ。



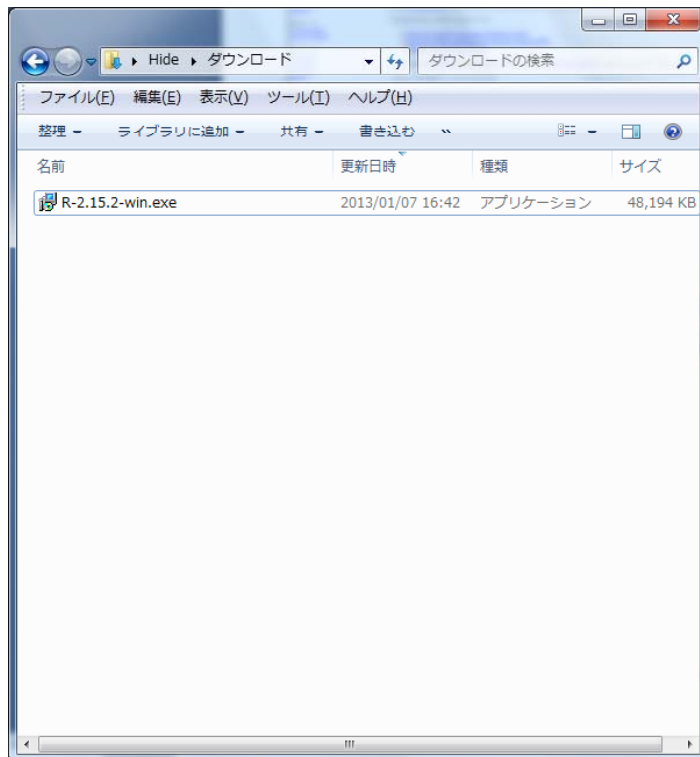
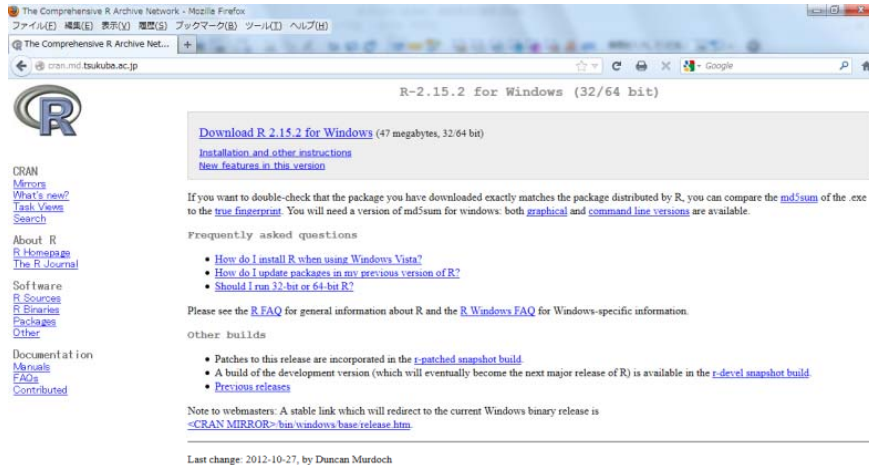
Japan の Tsukuba からダウンロードするのが分かりやすい。



Download and Instal R で、ウインドウズ、マック、リナックス のうち、自分の使っているコンピュータにあうものを選ぶ。

次の画面で base と contrib が出てくるので、baseを選ぶ。

一番上のDownload R... を選ぶ.

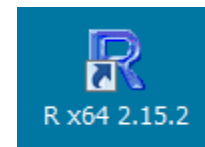


ダウンロードしたファイルが、「ダウンロード」というフォルダか、「デスクトップ」か、どこかにあるはずなので、探し出す。

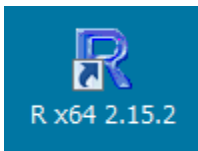
「R-○○.exe」をダブルクリックして、インストールする。○○のところは、RのバージョンやOSなどにより異なる。

32bitマシンか 64bit マシンかを聞かれたら、自分のあてはまるほうを選ぶ。

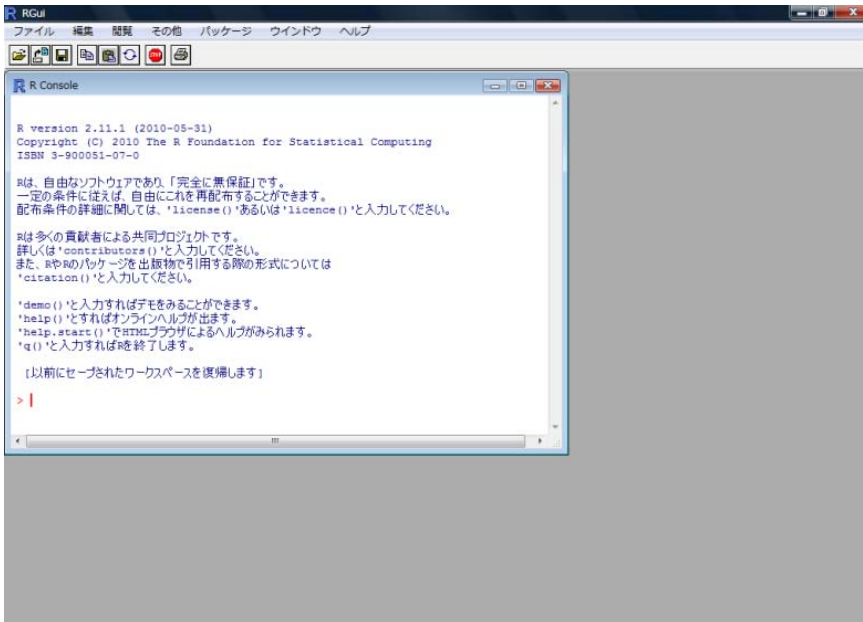
インストールができれば、画面上にこんなアイコンができています。



Rの起動と終了

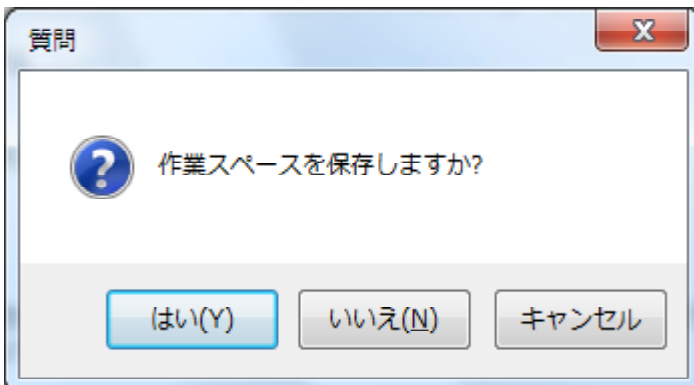


画面上にある左にあるようなアイコンをクリックすると、Rが起動する。

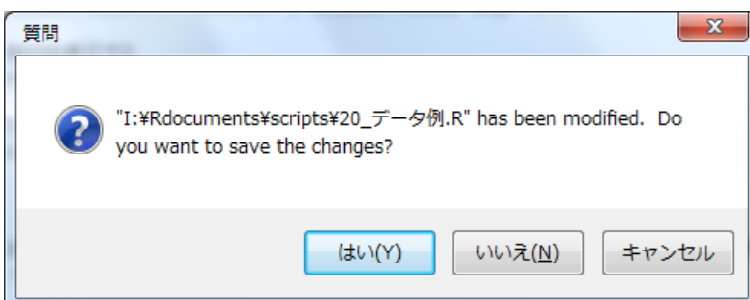


Rが起動すると左図の画面が開き、Rが起動する。

右上の×を推すと、Rが閉じる。



Rを閉じるとき、作業内容を保存するかどうか聞いてくるので、「はい」か「いいえ」かを答える。どちらがいいかは本人次第である。



最新のスクリプトが保存されていないときは、最新のスクリプトを保存するか聞いてくるので、「はい」か「いいえ」かを答える。「はい」とした場合には、スクリプトの保存が行われる。

その後、改めてRを終了する。

パッケージのインストール

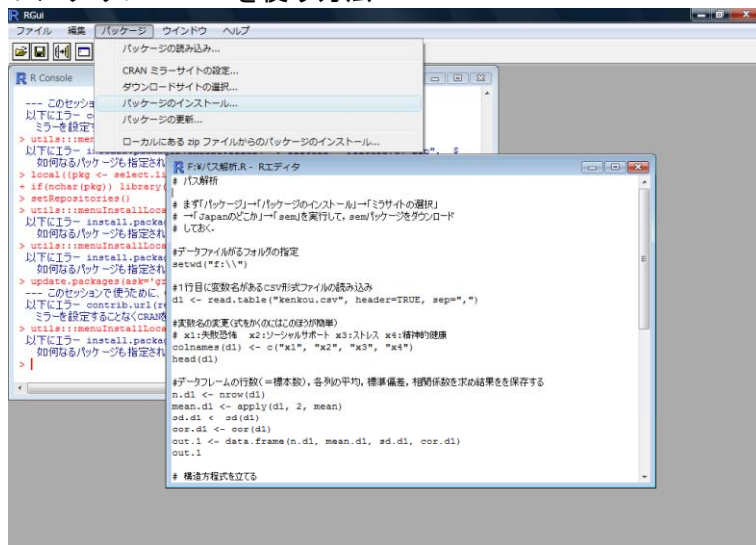
Rには、本体の他にパッケージというものがあり、パッケージに含まれる関数を使うためには、まずパッケージのインストールをする必要がある。ただし、パッケージにも、MASSのようにRをインストールしたときに同時にコンピュータにインストールされるものと、psychやsemのようにRをインストールしたときにはコンピュータにインストールされないものがある。以下は、後者のパッケージのインストールの説明である。

パッケージのインストールは、原則として当該コンピュータで一度だけやればよいが、Rのバージョンアップをしたり、システムの復元をしたりすると、過去のパッケージがインストールされていない状態に戻ることがあるので、その場合は改めてインストールする。

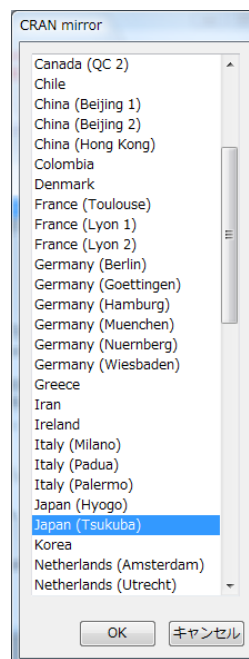
パッケージに含まれる各コマンドについては、Rをインストールしたフォルダの中にそのパッケージ名のフォルダができていれば、その中にパッケージのマニュアルがあるので参照するとよい。

【注意】 管理者または管理者権限を持ったユーザーとしてパソコンを起動（ログイン）している必要がある。

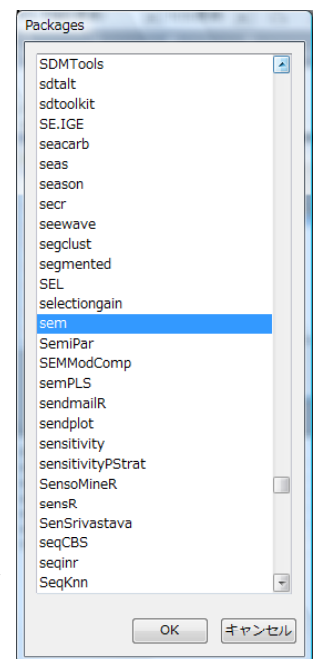
プルダウンメニューを使う方法



(1) 「パッケージ」→「パッケージのインストール」とする。



(2) ミラーサイトを設定する画面が出てくるので、最寄りのミラーサイトを選択する。JapanのTsukubaがわかりやすい。



(3) パッケージのリストが出てくるので、必要なパッケージ (psych や sem など) を選ぶ。

これでインストールできる。

Rのコマンドを使う方法

(1) Rのコンソール画面上に以下を入力し実行する
`install.packages("パッケージ名")`

パッケージ名のところは、psych や sem など、自分が必要とするパッケージの名前を書く。

(2) ミラーサイトを設定する画面が出てくるので、最寄りのミラーサイトを選択する。
JapanのTsukuba がわかりやすい。

これでインストールできる。

パッケージの読み込み・取り外し

パッケージは、Rをインストールしたときに同時にインストールされるものでも、後からインストールしたものでも、読み込みをしてからでないと使用できない。

読み込み

`library(パッケージ名)`

上記スクリプトを書いて実行すると、そのパッケージが読み込まれる。

パッケージ名を"で囲わない。

読み込み後なら、ヘルプを使って、そのパッケージに含まれる関数の使い方を読むことができる。

取り外し

`detach("package:パッケージ名")`

異なるパッケージが同一の関数名を使用しているときに、その2つのパッケージを読み込むと、どちらか一方のパッケージの関数が使えなくなる。これを避けるために、不要なほうのパッケージを取り外す。

一度取り外したパッケージでも、`library()` でまた読み込むことができる（再インストールする必要はない）。

```
# パッケージの読み込み・取り外し

# sem パッケージの読み込み

library(sem)

# lavaan パッケージの読み込み

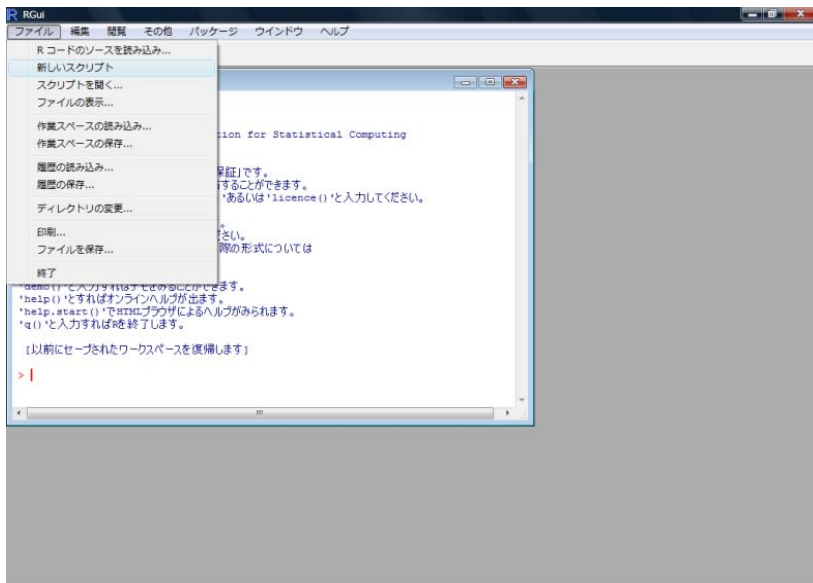
library(lavaan)

# どちらにも sem というコマンドがあるので、
# このあと sem パッケージのほうを使うためには、
# lavaan パッケージを取り外す。

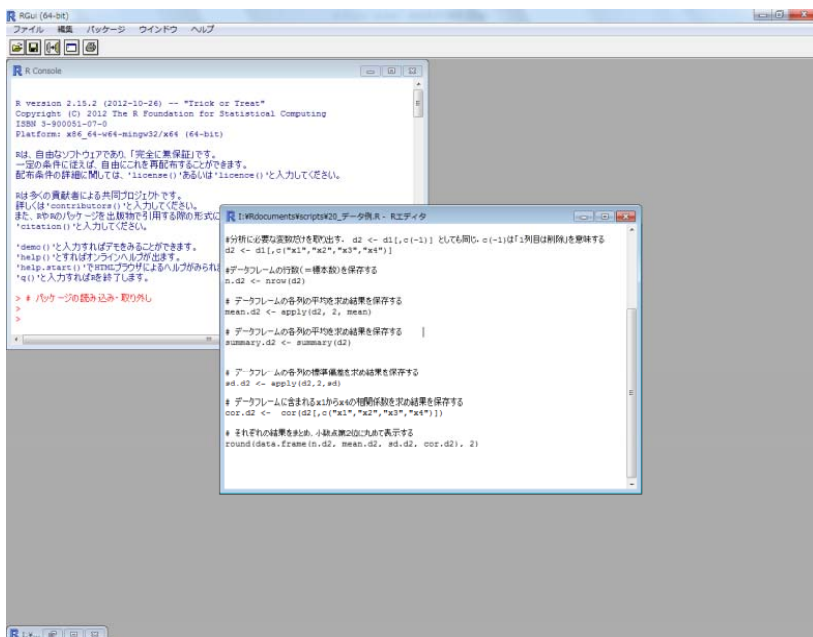
detach("package:lavaan")
```


スクリプトの作成・一般的注意

Rは、スクリプトと言われるプログラムを書いてそれを実行する、という使い方をします。
スクリプトは、Rエディタで開き、実行する範囲を指定して実行する。
実行結果は、コンソール（Console）画面や、グラフィクス（Graphics）画面に出力される。



スクリプトを最初を書くときは、Rを起動し、「ファイル」→「新しいスクリプト」を選択する。



スクリプトを書く画面（Rエディタ）が表示されるので、ここにスクリプト（プログラム）を書く。

スクリプト作成の一般的注意

スクリプトを書くときに「>」「+」「1:」などを左端につけてはいけない。

数式の途中で改行するときは、演算記号（+*/ など）を行末に書いた後に改行する。改行された行の先頭が + - * / などになってはいけない。

スクリプトにおいて、「#」以降のその行は、無視されるので、コメントなどを書くのに利用する。

生成物（オブジェクト）がある場合、それを表示するためには、そのオブジェクトを生成するスクリプトの部分で()で囲うと、生成結果がコンソール画面に表示される。

困ったときの対処

R が動かない、動くけどエラーが出て結果が出力されないときは、以下の可能性を検討する。

ユーザー名やディレクトリ名に全角文字や半角カタカナがある

コンピュータのユーザー名やディレクトリ名に、日本語フォントが入っていると、R が動かないことがある。

パッケージが正しくインストールされていない

パッケージをインストールするとき、R が自動的に指定するフォルダの不具合で、パッケージがインストールできないことがある（多くの場合、`c:\Program Files\R\R. バージョン番号\library`）。対応としては、他のフォルダに入ってしまったパッケージファイルをコピーするか、フォルダのプロパティを書き込み可能に変更することが考えられる。

スクリプトが間違っている

見た目は正しそう似見えても、スクリプトにミスがあることはよくある。

全角記号の混入

`x1` を `x 1` など

全角スペースの混入

`dm[<- d1[d1$sex=="男",]` を `dm[<- d1[d1$sex=="男",]` ただし は半角スペース、 は全角スペースなど

スペルミス

1（いち）と1（エル）を間違える	<code>x1</code> を <code>xl</code>
大文字と小文字を間違える	<code>Total</code> を <code>total</code>
複数形の <code>s</code> をつけ忘れる	<code>colnames</code> を <code>colname</code>
その他のスペルミス など	

ダブルクォーテーション忘れ

変数名や文字を指定するときに `"` で囲うのを忘れる `"x1"` を `x1`, `"a"` を `a` など

カンマ忘れ

関数のオプションを複数指定するときに要素間の `,` を忘れる
`table(d1$x1, d1$x2)` を `table(d1$x1 d1$x2)` など

カッコ忘れ

関数のオプション内での計算 `xlim=c((xmin - .5), (xmax + .5))` を `xlim=c(xmin - .5, xmax + .5)`
 純粋なカッコ忘れ `round(data.frame(n.d1, mean.d1), 2)` を `round(data.frame(n.d1, mean.d1, 2)`
 など

記号の違い

条件式で `==` とすべきところを `=` としている `d1[d1$sex=="f",]` を `d1[d1$sex="f",]` など

有効な対処法

ユーザー名、ディレクトリ名、変数名、データ値等に、全角記号や半角カタカナを用いない

アルファベットで書いたユーザを追加して、R を使うときはそのユーザー名でコンピュータにログインする。
 変数名などはローマ字にしておくとう分かりやすい

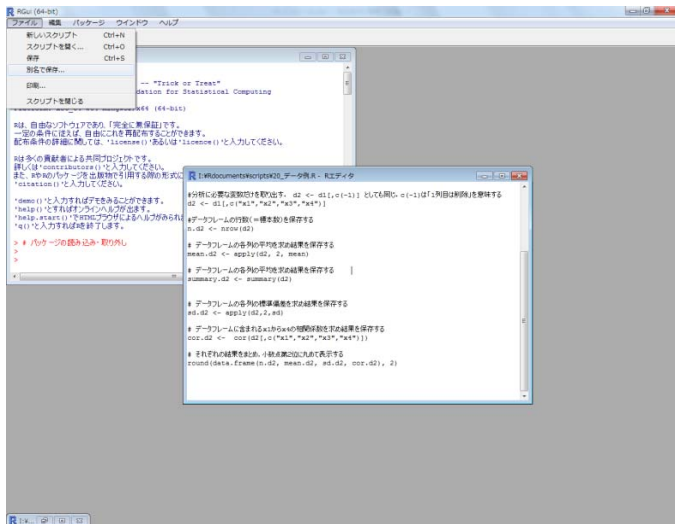
ヘルプを読む

関数のヘルプを見て、スクリプトが正しいかどうか、よく確認する。

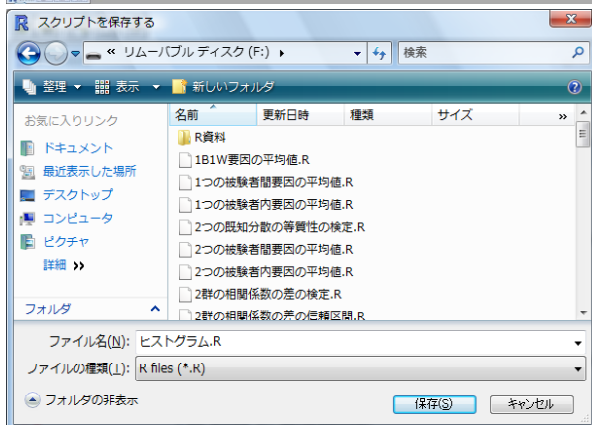
エラーが出る部分を削除して、書き直す

コピペせず、該当箇所をすべて書き直す。
 修正するより書き直した方が速いことも多い
 全角スペース、全角カンマなどは、スクリプト画面ではそれとわかりづらい。

スクリプトの保存・読み込み

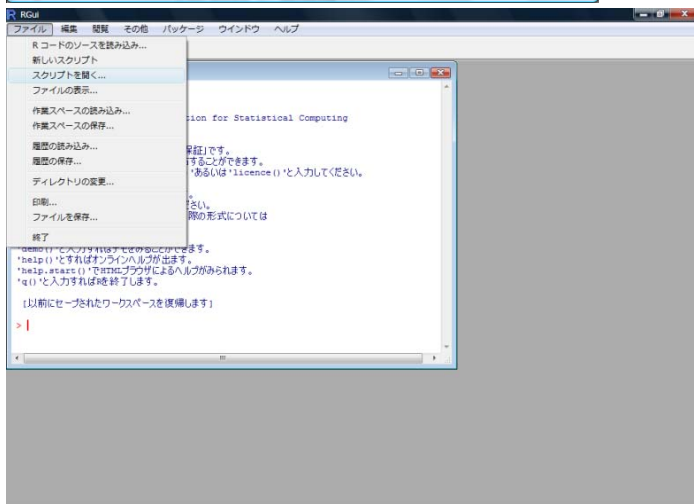


スクリプト（プログラム）を書いたら「ファイル」→「別名で保存」とする。
（和文フォントがファイルやディレクトリ名にあるとき「保存」を使うと変なファイル名になる。）

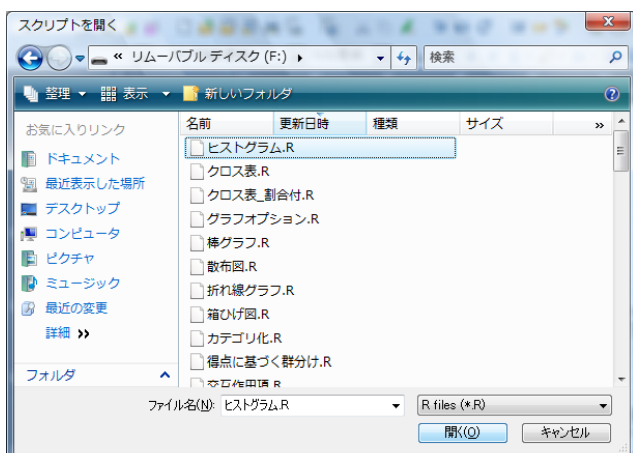


ドライブ、フォルダを移動し、ファイル名のところに「ファイル名.R」を入力する。

ファイル名は任意。「.R」という拡張子を書くのを忘れないようにする。（自動的に付けてくれない）



次にそのスクリプトを使うときは、Rを起動し、「ファイル」→「スクリプトを開く」とする。



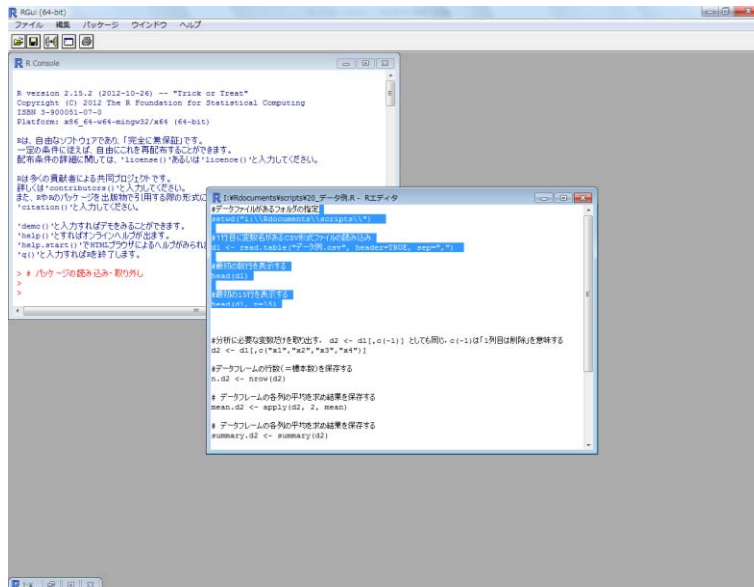
ドライブ、フォルダを移動し、目的のファイルを選択して、「開く」を押す。

読み込んだスクリプトを再編集して上書き保存するときも、「別名で保存」とした上で、同名ファイルを選択して保存するほうが間違いがない。

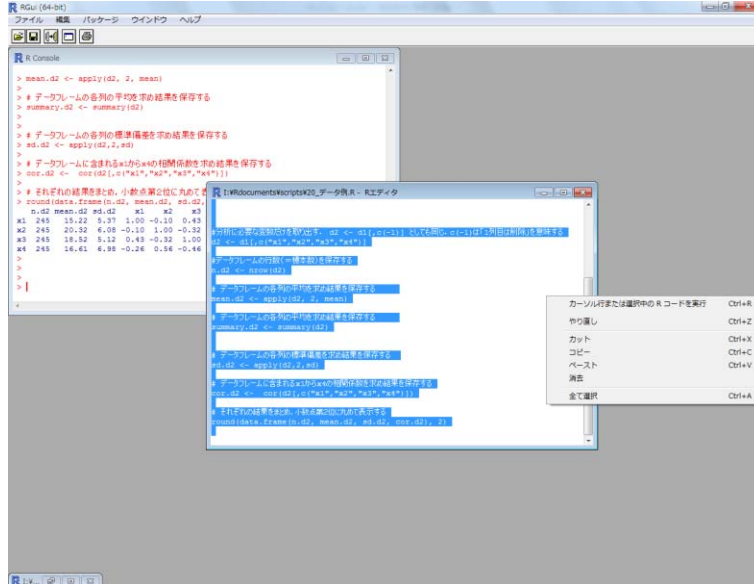
同名保存する場合、「既に存在します。上書きしますか?」と訊いてくるので、良ければ「OK」を押す。

ファイル名やディレクトリ名に和文フォントがなければ、「保存」で上書き保存できる。

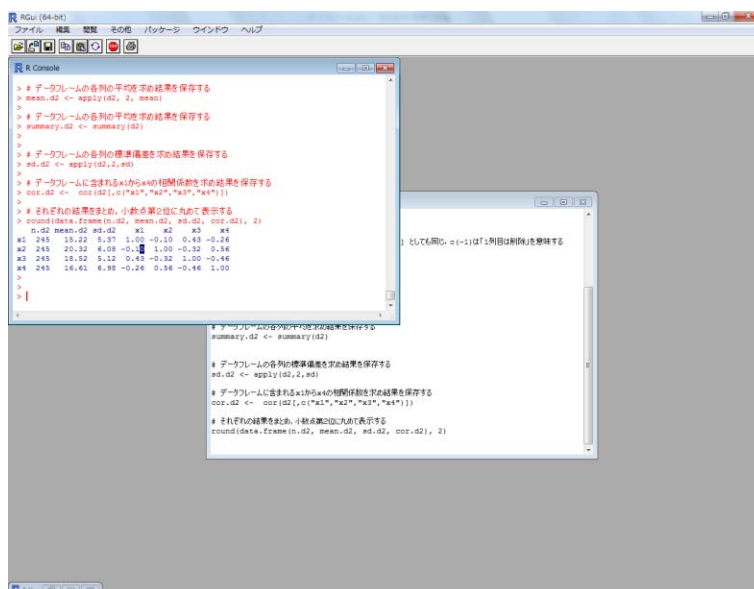
スクリプトの実行



スクリプトの実行したい部分を選択する。スクリプト全体を選択するのであれば、スクリプトエディタ上で右クリックし、「全て選択」を選ぶ。



スクリプトエディタ上で右クリックし、「カーソル行または選択中のRコードを実行」を選ぶ。



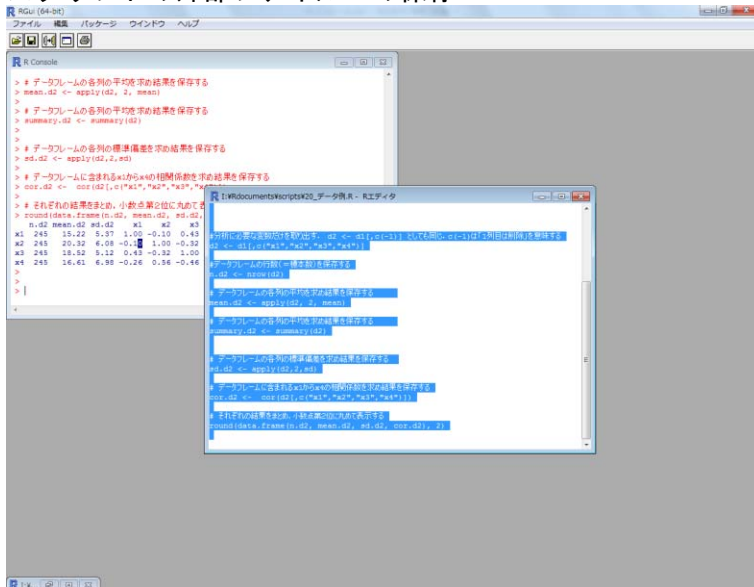
スクリプトが実行され、結果が表示される。

スクリプトとともに、コンソール画面に結果が表示される。

実行されたスクリプトは、コンソール画面上において、「>」が左端について表示される。

図は、グラフィクス画面に出力される。

スクリプトの外部ファイルへの保存



スクリプトの実行したい部分を選択する。スクリプト全体を選択するのであれば、スクリプトエディタ上で右クリックし「全て選択」を選ぶ。

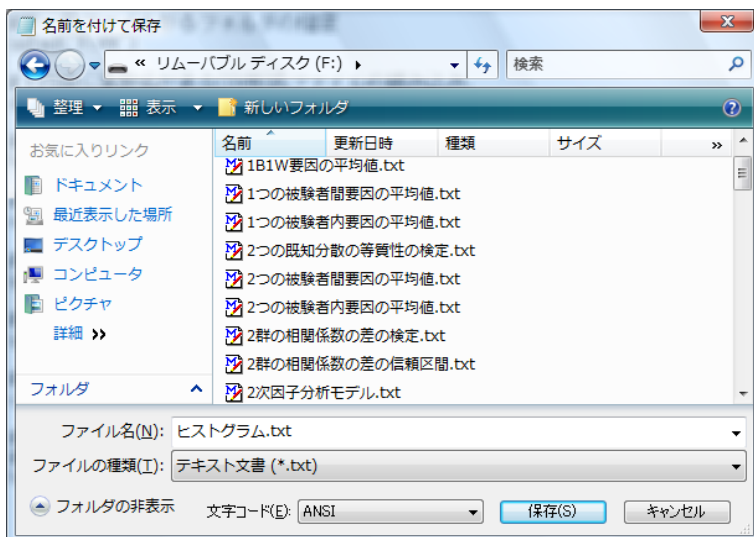
スクリプトエディタ上で右クリックし「コピー」を選ぶか、「CTRL+C」キーを押す。

```
#最初の 15 行を表示する
head(d1, n=15)

#分析に必要な変数だけを取り出す。 d2<-d1[c(1)] としても同じ。 c(1)は
#分析に必要な変数だけを取り出す。 d2<-d1[c(1)] としても同じ。 c(1)は
#データフレームの行数（＝標本数）を保存する
n.d2 <- nrow(d2)

# データフレームの各列の平均を求め結果を保存する
mean.d2 <- apply(d2, 2, mean)
```

メモ帳などを起動し、スクリプトをペースト（貼り付け）する。（wordでも構わない）

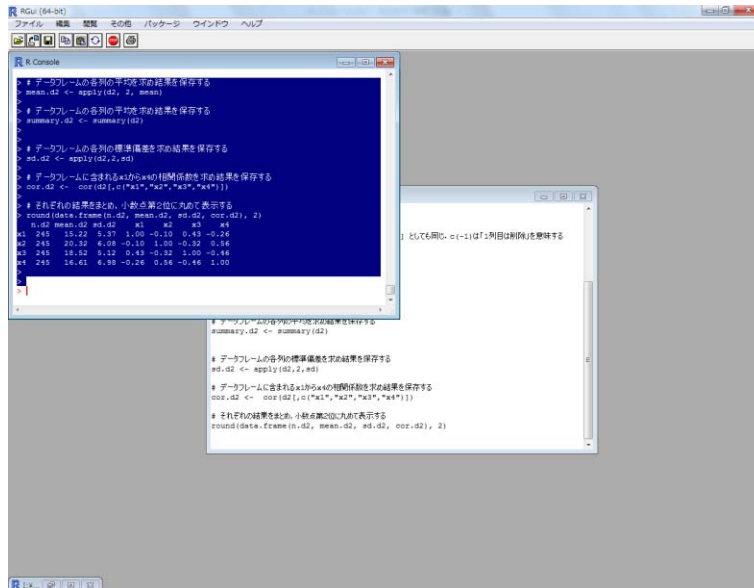


ドライブ、フォルダを移動し、「ファイル」→「名前をつけて保存」を選んで、ファイル名のところに、任意の名前を書いて「保存」する。

出力結果の外部保存

コピー・ペーストを使う方法と、sink関数を使う方法とがある。

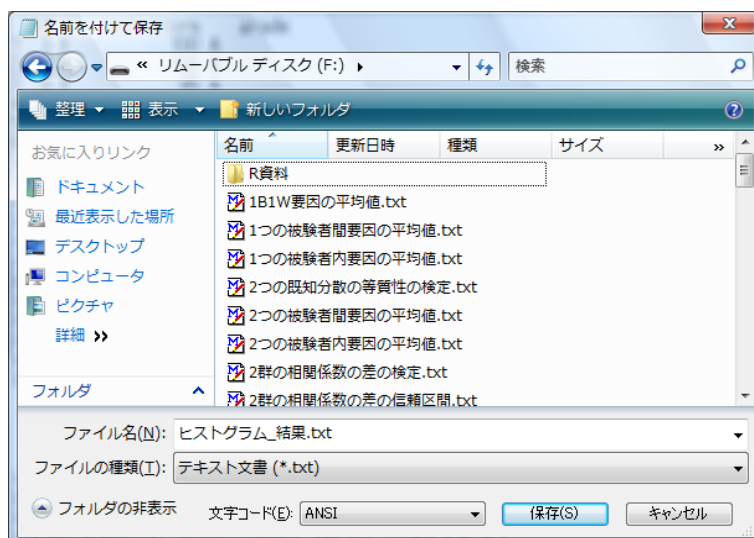
コピー・ペーストを使う方法



Rコンソール画面の該当箇所を範囲選択して、「編集」→「コピー」とするか、「CTRL+C」キーを押す。

```
> # データフレームの各列の標準偏差を求め結果を保存する
> sd.d2 <- apply(d2,2,sd)
>
> # データフレームに含まれるx1からx4の相関係数を求め結果を保存する
> cor.d2 <- cor(d2[,c("x1","x2","x3","x4")])
>
> # それぞれの結果をまとめ、小数点第2位に丸めて表示する
> round(data.frame(n.d2, mean.d2, sd.d2, cor.d2), 2)
  n.d2 mean.d2 sd.d2   x1   x2   x3   x4
x1  245   15.22  5.37  1.00 -0.10  0.43 -0.26
x2  245   20.32  6.08 -0.10  1.00 -0.32  0.56
x3  245   18.52  5.12  0.43 -0.32  1.00 -0.46
x4  245   16.61  6.98 -0.26  0.56 -0.46  1.00
```

メモ帳、Wordなど、適当なソフト（エディタ）を起動し、ペースト（貼り付け）する。



ドライブ、フォルダを移動し、「ファイル」→「名前をつけて保存」を選んで、ファイル名のところに、任意の名前を書いて「保存」する。

sink関数を使う方法

```
sink("ファイル名.txt")
Rスクリプト
sink()
```

結果のみが保存され、スクリプト部分は保存されない。
 コンソール画面には、結果は表示されない。
 Append=T オプションをつけないと、ファイルは上書きされる。
 ファイルが開いていると保存されない。

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> dl <- read.table("度数分布_データ.csv", header=TRUE, sep=",")
>
>
> # sink で結果をファイルに出力するようにする
> sink("sinkout1.txt")
>
> # コンソール画面には結果は表示されない
>
> # 最初の数行を表示
> head(dl)
>
> # 標本サイズ
> (n.score <- nrow(dl))
>
> # 最小値, 中央値, 平均値, 最大値, 四分位など
> (summary.score <- summary(dl$score))
>
> # 標準偏差
> (sd.score <- sd(dl$score))
>
> # 結果をまとめて出力
> c(n.score, summary.score, sd.score)
>
>
> # ファイルを閉じる
> sink()
>
```

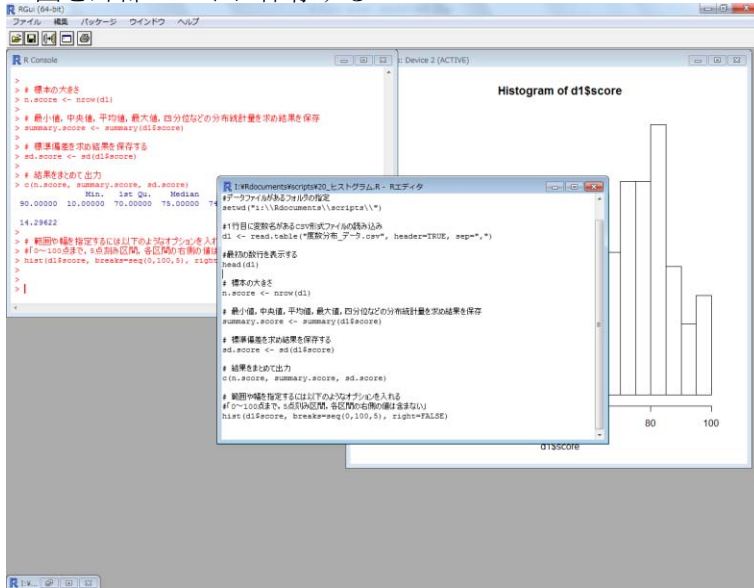
	A	B	C	D
1	id	sex	score	grade
2	1	f	85	A
3	2	f	100	A
4	3	m	80	A
5	4	f	80	A
6	5	f	65	C
7	6	m	80	A
8	7	m	75	B
9	8	m	100	A
10	9	f	65	C
11	10	m	55	D
12	11	f	90	A
13	12	m	80	A
14	13	f	65	C
15	14	f	75	B
16	15	m	70	B
17	16	f	70	B
18	17	f	75	B
19	18	m	65	C
20	19	m	50	D
21	20	f	95	A

Wordなどを使って、出力結果を確認する

```
| id      sex score  grade↵
1  1 f      85  A    ↵
2  2 f     100  A    ↵
3  3 m      80  A    ↵
4  4 f      80  A    ↵
5  5 f      65  C    ↵
6  6 m      80  A    ↵
[1] 90↵
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. ↵
 10.00  70.00   75.00   74.67  80.00  100.00 ↵
[1] 14.29622↵
      Min.    1st Qu.    Median    Mean   3rd Qu.    Max.
90.00000 10.00000  70.00000  75.00000  74.67000  80.00000 100.00000 14.29622
```

図の保存・PDFファイルに出力

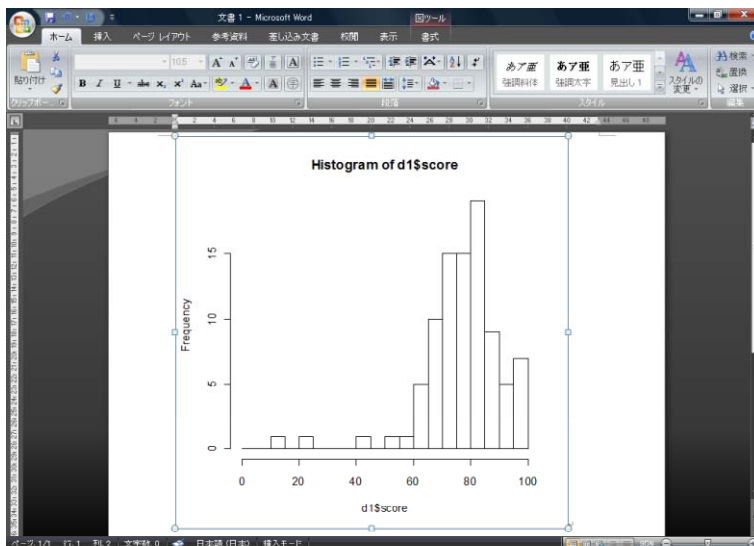
図を外部ソフトに保存する.



グラフィックス画面上で右クリックし、「メタファイルにコピー」を選択する.

もしくは、図上で左クリックし、「CTRL」+「C」キーを押す.

貼り付け先によっては、「メタファイルにコピー」だと背景が透明になり、「CTRL」+「C」だと背景が白になる.



Wordなどを起動し、ペースト（貼り付け）し、保存する.

ファイルに直接出力する方法

図の出力先をBMP, JPEG, PNG, TIFF, PDF形式ファイルにする `bmp`, `jpeg`, `png`, `tiff`, `pdf` 関数もある. 図を作成する関数の前にこれらの関数を置いておくと、指定された形式のファイルに図が出力ができる.

PDFファイルに出力する例

```
pdf("ファイル名.pdf", paper="a4", width=8, height=16, family="Japan1")
layout(matrix(c(1, 2, 3, 4, 5, 6), 3, 2, byrow=TRUE))
for(i in 1:6){
  plot(dx[, i], data=dx)
}
layout(1)
dev.off()
```

オブジェクトの確認・消去

Rでは、計算や分析の結果をメモリ上に“オブジェクト”として再利用することができる。

オブジェクトを生成するには、たとえば「`r1 <- c(1, 2, 3)`」とすると、`r1`という名前で、`1, 2, 3` を要素に持つベクトルが生成される。

例えば、スクリプトにミスがあるのに、同名の古いオブジェクトが残っていると、過去の結果がいつまでも使われて、ミスに気がつかないことがある。

そこで、いまメモリ上にあるオブジェクト名を確認したり、オブジェクトを消去したりする必要がある。新しく分析を始めるときは、いったんメモリ上のオブジェクトを消去しておいたほうがよい。

メモリ上にあるオブジェクト名を確認

```
ls()
```

メモリ上にあるすべてのオブジェクトを消去

```
rm(list=ls())
```

```
> r1 <- c(1, 2, 3)
```

```
> r1
```

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

```
> ls()
```

```
[1] "r1"
```

```
> rm(list=ls())
```

```
> ls()
```

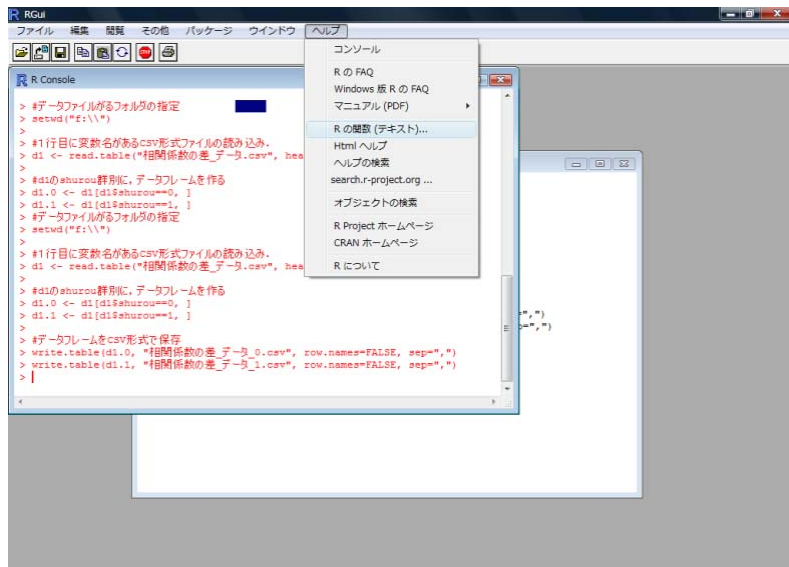
```
character(0)
```

```
> r1
```

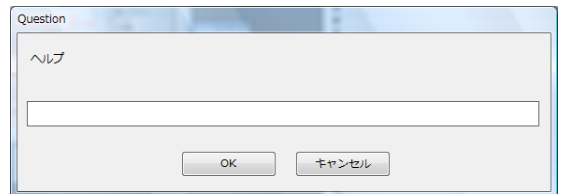
```
エラー： オブジェクト 'r1' がありません
```


ヘルプの使い方

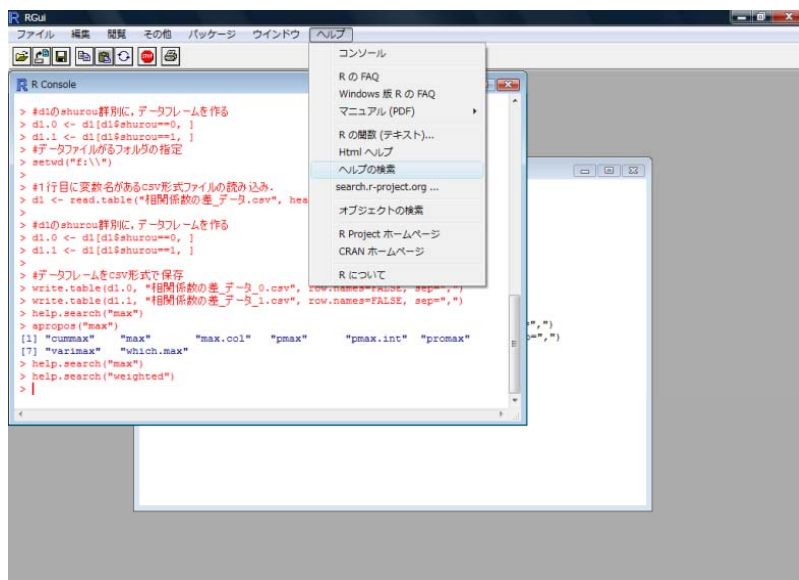
使うコマンド(関数名)のスペルが分かっている場合



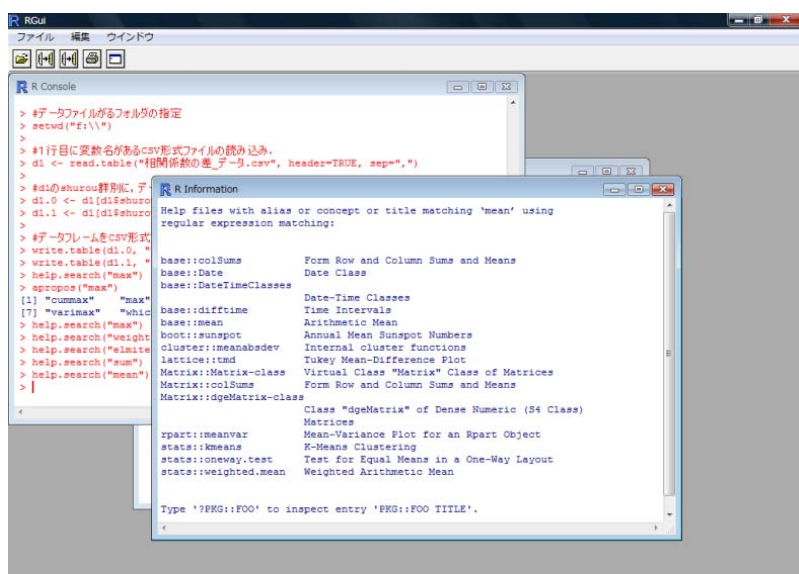
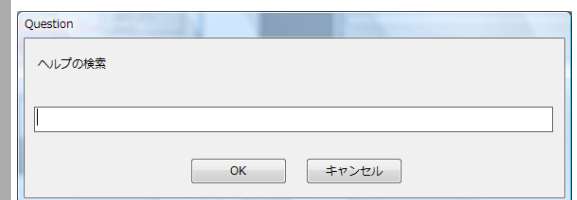
「ヘルプ」→「Rの関数 (テキスト)」とすると、次の入力画面が表示されるので、コマンドのスペルを入力して使い方を調べればよい。



使うべきコマンド(関数名)がよくわからない場合: キーワード検索

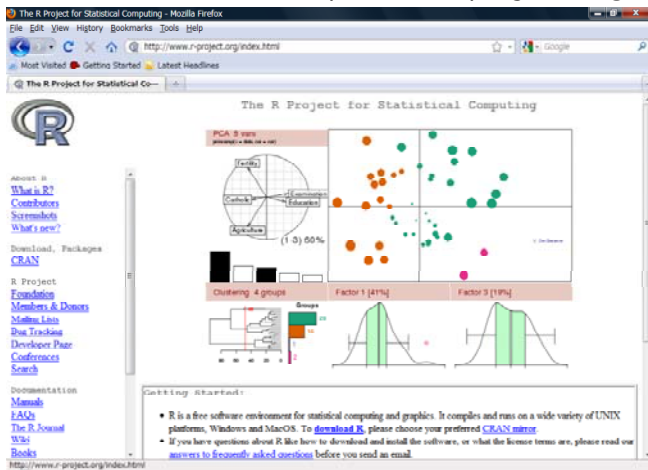


「ヘルプ」→「ヘルプの検索」とすると、次の入力画面が表示されるので、キーワードを入力する。



たとえば、「mean」というキーワードを入力すると、「mean」を含むヘルプ文書の一覧を表示してくれるので、その中からコマンド(関数名)を選んで、上の方法で関数のヘルプを見る。

Rの情報源

公式ホームページ <http://www.r-project.org/>

プログラムやパッケージ（オプションで作られた分析コマンド）をダウンロードしてくる時に使う。Japanだったら、Tsukubaのミラーサイトを使うのがわかりやすい。

日本語のRの情報集積ホームページ <http://www.okada.jp.org/RWiki/?RjpWiki>

新しいパッケージや、FAQなどがあり、何か調べたりするのに便利。掲示板もある。

他にもいろいろあります。