R Maniax

フリーの統計ソフト「R」を使いこなす本

後藤和智（後藤和智事務所 OffLine）

フリーの統計ソフト「R」を使いこなす本

後藤和智（後藤和智事務所 OffLine）

まえがき

32 冊目の同人誌となります、後藤和智です。

さて、以前弊サークルでは、「コミックマーケット 81」（2011 年冬コミ）において、『三訂版・市民のための統計解析』という同人誌を刊行しました。その同人誌は弊サークルでも最大の売上を記録し、さらに翌年の冬コミで出した『紅魔館の統計学なティータイム』は、その売上をさらに更新しました。

また統計学をの手法を用いた同人誌も確実に増えており、弊サークルが統計学で活動する前から統計学を用いたアダルトコンテンツなどのレビューを行っている、「エロマンガ統計」シリーズのサークル「でいひま」（奉祝、ニコニコ超会議 2 のニコニコ学会ポスター発表で大賞受賞！）の他、『ボカロ動画の統計分析』シリーズのサークル「intact」や、『声優統計』のサークル「日本声優統計学会」などといった興味深い研究も出てきています。

さらにジャーナリズム界隈でも、朝日新聞や毎日新聞、東京新聞などで「データジャーナリズム」が注目され、データや統計学に基づいた知見の重要性も見直されてきています。さらに、西内啓『統計学が最強の学問である』（ダイヤモンド社）がベストセラーになったり（この本は超おすすめです！あと 3 ヶ月早く発売されていたら紅魔統計本で紹介したのに！）、『週刊東洋経済』や『週刊エコノミスト』なども統計学の特集を組んだりしているなど、ビジネスでも統計学を重視する動きが広がっています。

そんな状況下で、弊サークルは、統計学を身近にするツールとして、フリーの統計ソフト「R」に注目してきました。無料で提供されている「R」は、無料であるにもかかわらず商用ソフトにも劣らないほどの機能を有しており、また元来の問題点であったインターフェイスの扱いづらさも、「R コマンダー」「R スタジオ」「Rz」などのパッケージの普及により解消されつつあります。

本書は、2012 年冬コミに完売宣言を行い、2013 年 5 月に Kindle 電子書籍として刊行し、シリーズに一区切りをつけた『三訂版・市民のための統計解析』の問題意識を受け継ぎ、よりわかりやすい、使いやすい解説書を志向しております。本書の特徴としては、それぞれの事項の解説を 2 ページないし 4 ページにまとめており、R のガイドブックとしての役割をより明確にするものになっていると考えております（自分で言うな）。

「理系と文系の融合」などが一部では言われていますが、少なくとも客観的にものを考えるツールとしての基礎的な統計学を理解せずに、徒にビッグデータだとかに飛びついてしまう様は滑稽なものでしかありません。様々なデータを実社会に役立てるためにも、今こそ統計学の基礎知識が必要となります。そして、統計学をうまく使うためのツールとしての「R」を、より多くの人に知ってほしいという思いで執筆しております。

なお本書は、Windows ユーザー向けに書かれております（筆者の動作環境は Windows8 です）。R はオープンソースのソフトウェアであり、Macintosh や Linux でも動かすことはできますが、その方法については、申し訳ありませんが本書では割愛いたします。また本書においては、R3.0.0 を使用しております。

目次

第１章
Ｒの導入と基礎

1.1 Ｒのインストールと下準備

まえがきでも書いた通り、本書で取り扱う統計ソフト「R」は、オープンソースで開発されているフリーソフトです。そのため、保障やサポートなどはありませんが、多くの人によってＲそのものやパッケージの開発が行われており、進歩や改善が進められています。ただその分、Ｒの更新ペースも早いということがありますが、定期的にアップデートしておくようにしましょう。

Ｒはその開発元である R-project のサイトから入手することができます。ただ、まずはＲを知っておくために、日本のＲの総本山である「RjpWiki」（http://www.okada.jp.org/RWiki/index.php?RjpWiki。図 1.1）にアクセスしましょう。RjpWiki には、Ｒの概要や歴史が掲載されているほか、操作方法などユーザー向けのコンテンツも充実しており、Ｒを使うようになったあとも何回もアクセスするサイトです。

とにかくＲを始めたい方や、あるいは RjpWiki をある程度読んだという方は、早速Ｒを入手しましょう。RjpWiki の「Ｒのインストール」のページに飛んで、「最新版はこちらから」の「こちら」をクリックすると、Ｒのサーバーである CRAN のサイトに行けるので、一番上の「Download R X.X.X for Windows」をクリックし、インストーラを入手します。その後はインストーラの指示に従ってインストールしてください。

図 1.1 RjpWiki（http://www.okada.jp.org/RWiki/index.php?RjpWiki、2013 年 6 月 1 日閲覧）

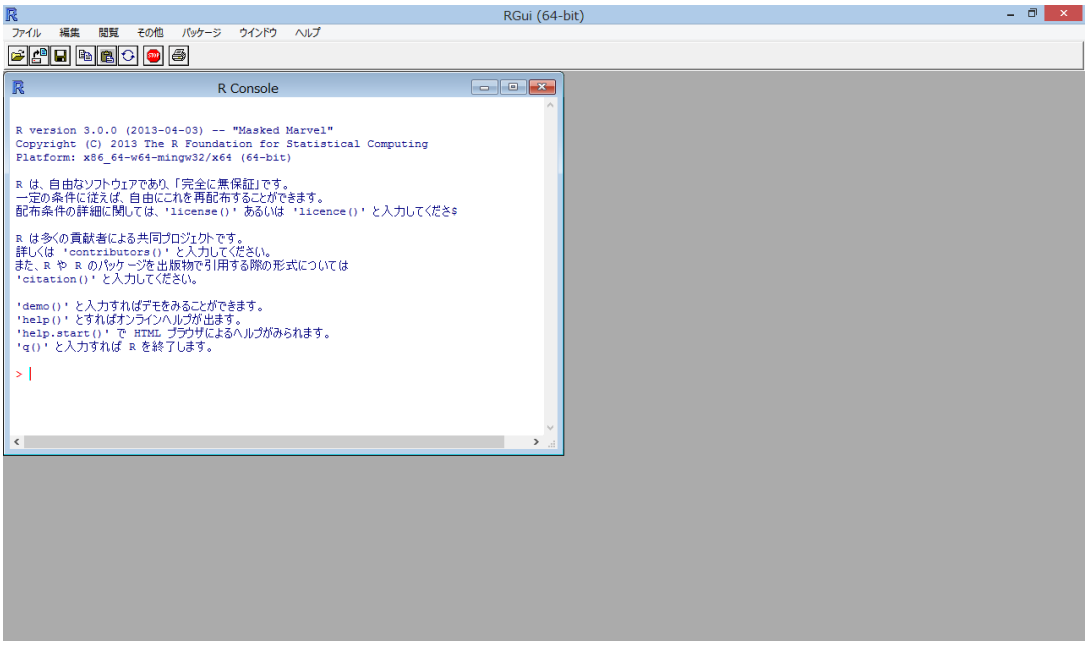

図 1.2 R の起動画面

R を起動すると、図 1.2 のような画面が現れます。ここで、文章が表示され、入力待ちになっている画面を「コンソール」と言い、これからの分析は主にこのコンソールで行われます。また、グラフなどを作成したときには、グラフを表示するためのフィールドがその都度表示されます。

R で分析を行う前に、R の使用環境を整えておきましょう。まず、コンソールのフォントなどは、「GUI プリファレンス」で変更することができます。「編集」メニューから「GUI プリファレンス …」を選ぶと、図 1.3 のような画面が現れるので、ここで好きなように調整しましょう。ちなみに筆者は、この画面の「Font」欄を「MS Gothic」（MS ゴシック）に変更して使うようにしています。

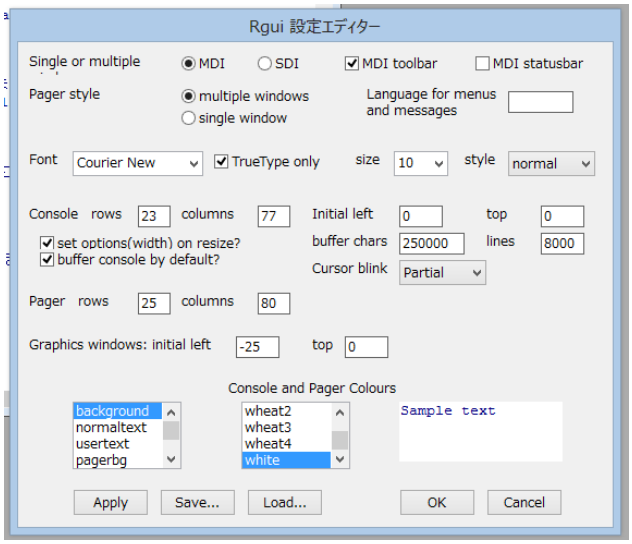

図 1.3 R の GUI プリファレンス

また、R で重要になってくるのは作業ディレクトリです。これから説明するデータの読み込みなどを行う場合も、読み込み対象となるデータが作業ディレクトリに入っていなければなりませんし、ファイルを出力する場合にも作業ディレクトリに出力されます。作業ディレクトリは、「ファイル」メニューの「ディレクトリの変更 …」で変更することができます。

また、R では、作業中のスペースをファイルとして保存することもできます。「ファイル」メニューの「作業スペースの保存 …」で保存できるほか、R を閉じる際にも作業スペースを保存するかの選択肢が現れます。特に R を閉じるときの選択肢で作業スペースを保存した場合は「.RData」というファイルが保存されます。保存した .RData 形式のファイルをエクスプローラでダブルクリックしたときは、それを開いたフォルダが作業ディレクトリになるので、いちいち作業ディレクトリを変更しないで済みます。

最後に強調しておきたいことがあります。それは「R は大文字と小文字を区別する」ということです。R においては、大文字と小文字が違うだけでも別物として扱われます。R を扱うときには、そのことには注意を払っておいてください。

1.2 パッケージのインストールと R コマンダー

R はフリーソフトでありながら、商用ソフトにも引けを取らない様々な分析を行うことができますが、R の機能をフル活用したいのであれば、パッケージというものをインストールする必要があります。パッケージとは、R の機能を強化するもので、一部の検定力分析や、多変量解析、空間統計学などといった高度な手法は、パッケージを使わないとできないものも多くあります。

パッケージのインストールは、「パッケージ」メニューの「パッケージのインストール ...」から行います。ただその前に、CRAN（R のサーバー）のミラーサイトと（どこを選んでもほとんど影響はないと思います）、ダウンロードするサイトの設定を行っておきましょう。これらの設定は、「パッケージ」メニューから行えます。CRAN ミラーサイトの設定は「CRAN ミラーサイトの設定 ...」、ダウンロードするサイトの設定は「ダウンロードサイトの選択 ...」を使います。ダウンロードサイトについては、もしわからなかったら「CRAN」「CRAN(extras)」「R-forge」の 3 つを選択するのがおすすめです（Ctrl キーを押しながらクリックすれば個別に選択することが可能です）。

CRAN のミラーサイトと使用するサイトを選んだら、図 1.4 のような画面が出るはずです。ここでインストールするパッケージを決めるわけですが、PC に余裕があれば、インストールできるパッケージすべてをインストールしてしまうことをおすすめします。全てのパッケージをインストールするには、まず一番上のパッケージが選択されていることを確認した後、スクロールバーを一番下まで下ろし、一番下のパッケージを Shift キーを押しながらクリックします。そして、全てのパッケージが青く反転されれば成功です。

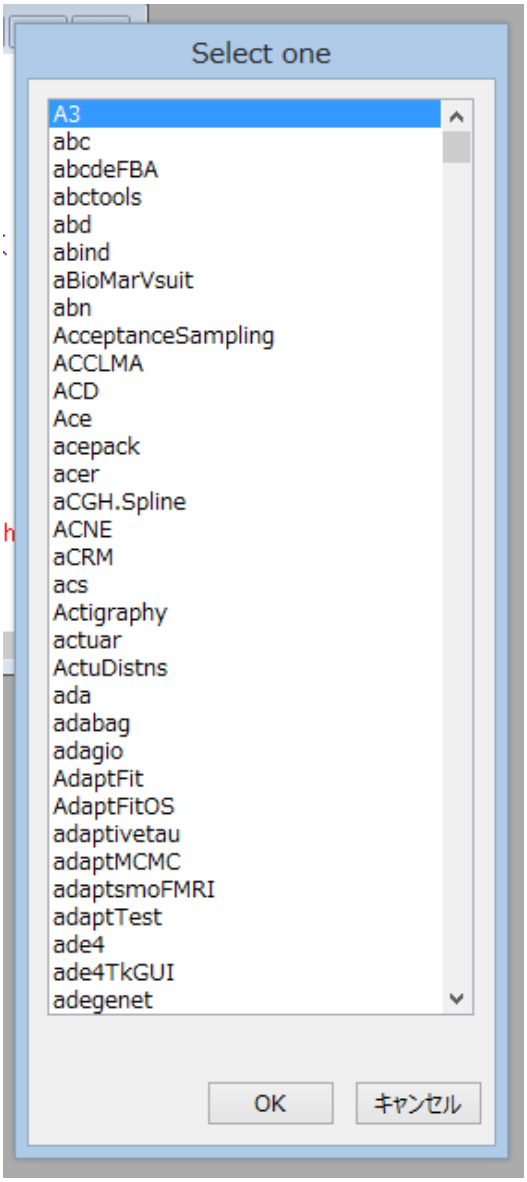

図 1-4 パッケージの選択画面

CRAN、CRAN(extras)、R-forge のパッケージをすべてインストールしようとすると、だいたい 2 ～ 3GB くらいの量になります。ブロードバンド環境下でも相当時間がかかりますので、なるべく時間と回線に余裕のあるときに行うことをおすすめします。睡眠時間中にやってしまうのが吉でしょう。

このようにしてインストールしたパッケージは、「パッケージ」メニューの「パッケージの読み込み」か、あるいはコマンド **library(パッケージ名)** で呼び出すことができます。ひとたびパッケージを呼び出せば、R を閉じるまでそのパッケージの中に入ったコマンドやデータを使うことができます。

なお、パッケージには、CRAN で提供されていないものもあります。そのようなパッケージは（R のパッケージは zip ファイル形式で提供されています）、パッケージをコンピュータに保存したあと、「パッケージ」メニューの「ローカルにある zip ファイルからのパッケージのインストール ...」で先ほど保存したパッケージを選択します。あとは、これで読み込んだパッケージも、CRAN から取得したパッケージと同じように呼び出すことができます。

さて本書で紹介する統計解析手法は、Rのコンソールのほか、代表的なRの入力支援ソフトである「R Commander」（Rコマンダー）というものも用います。Rコマンダーは、第一義的にはプログラミング言語である故インターフェイスの不親切さが問題視されていたR言語をより使いやすくするためのソフトの一つで、CRANでパッケージとして提供されています。

Rコマンダーのパッケージの名称は「Rcmdr」です（大文字・小文字注意！）。ここでは試しにRコマンダーを起動してみましょう。Rコマンダーを起動するには、次のように入力します。

```
> library(Rcmdr)
```

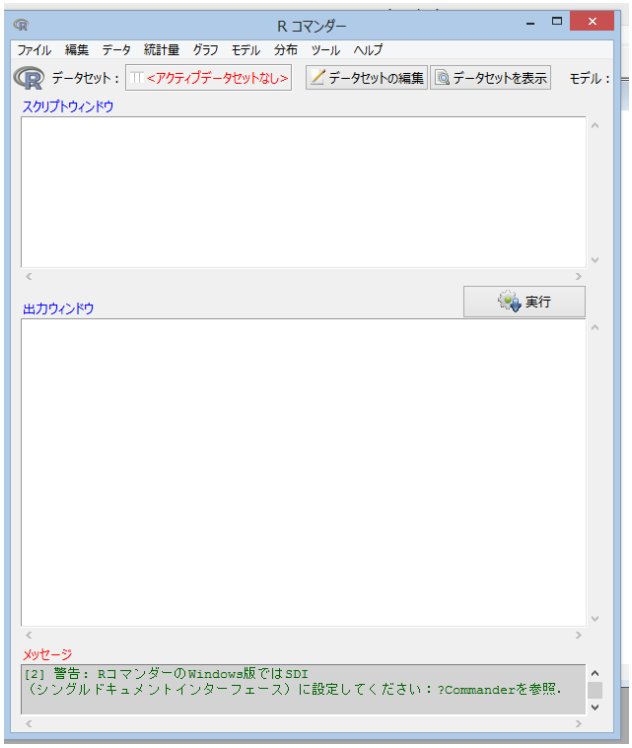

図1.5 Rコマンダー

これを入力すると、Rコマンダーに付随するいくつかのパッケージ（MASSなど）が同時に読み込まれたあとに、図1.5のようなウインドウが出ます。これがRコマンダーであり、Rコマンダーを使って作業をする際には、適宜この画面を見に行くことになります。

Rコマンダーには上部の「スクリプトウインドウ」と下部の「出力ウインドウ」があり、基本的にはスクリプトウインドウで入力されたコマンドの結果が、下の出力ウインドウに出るという形式がとられます。試しにスクリプトウインドウに「1+1」と入力してみると、次のような結果が出力ウインドウに出力されます。

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

このような四則演算はR本体のコンソールでも可能です。また、Rのコンソールや Rコマンダーのスクリプトウインドウにおいて、半角の＃を入力した場合、そのあとは註釈として無視されます。本書を含め多くのRの解説書では、コマンドを説明するときに半角の＃が使われますし、プログラミングを行うときにもメモとして半角の＃を使えば、何がやりたいのかわかりやすくなります。

```
> 1+1 # 足し算
[1] 2
> 2^5 # 累乗
[1] 32
> a # 文字を呼び出す（aには何も入っていない）
 エラー： オブジェクト 'a' がありません
> a <- 3 # 文字に数値を代入
> a # 文字を呼び出す
[1] 3
> b <- 7/5 # 文字には計算結果も入れられます
> b
[1] 1.4
```

1.3 データの形式と読み込み

Rを使う上で覚えておかなければならないのは、Rにおける、複数の数値を含むデータは、「ベクトル」と「行列」、そして「データフレーム」の3種類があるということです。

Rで言う「ベクトル」とは、数値を直線状に並べたもので、「行列」とは数値を格子状に並べたものを指します。また、「データフレーム」とは、数値を格子状に並べているという点では「行列」と変わらないかもしれませんが、「行列」がそれぞれの行（横の並び）と列（縦の並び）についてそれぞれ特に意味を持たないのに対し、「データフレーム」ではそれぞれの行はデータの「個票」としての、また列は「パラメータ」としての意味を持ちます。さらに、今後紹介していく分析用のコマンドは、「データフレーム」でないと使うこともできないものも多数存在し、さらにRコマンダーは「データフレーム」でないとデータセットとして設定できないので、特に「行列」と「データフレーム」の違いについては、Rを使う際には常に意識しておく必要があります。

Rでベクトルを作るには、**c(数値1, 数値2, 数値3,...)** といった具合に入力します。これを文字に代入することもできます。また行列を作るには、コマンド**matrix(データ ,ncol= 列数 ,nrow= 行数 ,byrow=F)** で作成します。行数、列数、byrowなどは必要に応じて省略可能（byrowを省略した場合はbyrow=F(FALSE)として処理される）です。作りたい行列に合わせてアレンジしましょう。

```
> c(1,2,3,4,5)
[1] 1 2 3 4 5
> a <- c(1,2,3,4,5)
> a
[1] 1 2 3 4 5
> rep(3,4)
[1] 3 3 3 3
> rep(3,4) # 繰り返し（ここでは3を4回繰り返す）
[1] 3 3 3 3
> 1:5 #1から5まで1刻み
[1] 1 2 3 4 5
> matrix(1:6,ncol=2) # 列数のみを指定する
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
> matrix(1:6,ncol=2,byrow=T) #byrow=Tを入力すると順番が変わる
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
> matrix(0,nrow=3,ncol=5) # 行数・列数を指定すれば同じ数値ばかりの行列も作れる
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    0    0    0
[2,]    0    0    0    0    0
[3,]    0    0    0    0    0
```

次に「行列」と「データフレーム」の違いについて説明します。行列もデータフレームも、どちらも数値を格子状に並べたものですが、データフレームは、先ほども述べたとおり、それぞれの列がデータの「個票」としての、またそれぞれの行がデータの「パラメータ」としての役割を持っています。

その違いを見ていくために、簡単な行列を作り、それをデータフレームに変換して、それぞれの値を呼び出してみます。行列をデータフレームに変換するには、**data.frame(行列)** と入力します。

```
> x <- matrix(1:6,ncol=2,byrow=T)
> x
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
> x <- data.frame(x) # 行列をデータフレームに変換
> x
  X1 X2
1  1  2
2  3  4
3  5  6
```

ここで注目してほしいのは、行列の場合とデータフレームの場合で、それぞれデータを呼び出したときです。行列のときは、それぞれの行と列に [,1] や [1,] などという行と列の番号が割り振られているのに対し、それをデータフレームに変換したときは、「X1」「X2」という名前がついていることがわかるはずです。

データフレームの場合は、x という文字にデータが入っているとき、**x$ 列の名前**というコマンドでその名前のついた列のデータを取り出すことができるようになります。また、行や列の名前は、それぞれ **rownames()**、**colnames()** というコマンドで変更することもできます。

```
> x <- matrix(1:6,ncol=2,byrow=T)
> x[3,2] # 行数と列数を指定してデータを呼び出す
[1] 6
> x[3,] # 行のみ、列のみを呼び出すことも可能（ここでは行のみ）
[1] 5 6
> x <- data.frame(x)
> x
  X1 X2
1  1  2
2  3  4
3  5  6
> x$X1 # 列の名前を指定して列の数値を呼び出す（データフレームのみ）
[1] 1 3 5
> colnames(x) <- c("データ1","データ2") # 列の名前を変更
> x
  データ1 データ2
1       1       2
2       3       4
3       5       6
> x$データ1 # 変更後の列の名前で呼び出す
[1] 1 3 5
```

また、R ではテキストデータや csv ファイルなどからデータを読み込むこともできます。ここでは最も頻繁に使うと思われる、csv ファイルからの読み方を説明しましょう。

まず、データを表計算ソフトで作り、それを csv ファイルとして保存します（図 1.6）。ここでは平成 23 年東京都統計年鑑における、京王電鉄井の頭線のそれぞれの乗車・降車人員の数を使ってみましょう。右のようなデータを Excel で作り、「keioinokashira.csv」という名前で作業フォルダに保存します。

	A	B	C	D	E
1	駅名	定期乗車	普通乗車	定期降車	普通降車
2	渋谷	34975	25778	34975	25890
3	神泉	647	1035	647	1263
4	駒場東大前	4569	2615	4569	2478
5	池ノ上	688	1052	688	965
6	下北沢	11926	10871	11926	11407
7	新代田	627	932	627	826
8	東松原	1956	1413	1956	1335
9	明大前	2771	3459	2771	3235
10	永福町	2810	2575	2810	2646
11	西永福	1702	1629	1702	1530
12	浜田山	2700	2403	2700	2416
13	高井戸	4866	2669	4866	2579
14	富士見ヶ丘	1170	1257	1170	1257
15	久我山	3781	2904	3781	2815
16	三鷹台	2540	1613	2540	1485
17	井の頭公園	560	641	560	632
18	吉祥寺	13917	11400	13917	11882
19					

図 1.6 Excel でデータを作成

csv ファイルからデータを読み取るときのコマンドは read.csv(" ファイル名 ") です。また、右のように最初の行に列の名前がある場合は「header=T」を、また行の名前がある列を指定する場合は「row.names= 名前のある行の場所」と括弧の中に追加します。ここでは 1 列目に行の名前があるため、「row.name=1」となります。読み込みは次のように行います。ただし注意しなければならないのは、read.csv コマンドでデータを読み込んだ場合、「行列」として読み込まれることです。そのため「データフレーム」でしか使えないコマンドを使いたい場合は、データフレームに変換する必要があります。

そのほかにも、write.csv(データ名 ," ファイル名 .csv") というコマンドを使えば、csv 形式で作業フォルダに保存することも可能です。

```
> dataset <- read.csv("keioinokashira.csv",header=T,row.names=1)
> dataset <- data.frame(dataset)
> dataset
           定期乗車 普通乗車 定期降車 普通降車
渋谷         34975    25778    34975    25890
神泉           647     1035      647     1263
駒場東大前     4569     2615     4569     2478
池ノ上         688     1052      688      965
下北沢       11926    10871    11926    11407
新代田         627      932      627      826
東松原        1956     1413     1956     1335
明大前        2771     3459     2771     3235
永福町        2810     2575     2810     2646
西永福        1702     1629     1702     1530
浜田山        2700     2403     2700     2416
高井戸        4866     2669     4866     2579
富士見ヶ丘     1170     1257     1170     1257
久我山        3781     2904     3781     2815
三鷹台        2540     1613     2540     1485
井の頭公園      560      641      560      632
吉祥寺       13917    11400    13917    11882
> write.csv(dataset,"keioinokashira2.csv") # データを csv 形式で書き出す
```

さらに、データフレームの中のデータも、「数値」と「因子」の 2 つに分けられます。「数値」とは量的な、「因子」とは質的なデータを指します。データを集計する過程においては、例えば性別が「男性が 1、女性が 2」という風に設定されているデータがあったり、あるいは年代などが数値で入っていることは少なくありません。しかし、明らかに質的なパラメータであるこれらのパラメータが数値として集計されると、いろいろと困難をきたします。

csv ファイルなどを読み込んでデータセットを作成したとき、その中のデータが「数値」か「因子」かについては、列ごとに判断されます。列に入っている全てのデータが数値のときは、その列のパラメータは「数値」として判定されます。しかし、その中に一つでも文字が入っていたら、それは「因子」となります。

また、データセット内の「数値」として保存されているデータを「因子」に置き換えるには、コマンド factor を使います。ここでは例として、先の京王井の頭線のデータに、停車する種別（急行が停車する場合は「急行」、各駅停車のみ停車する場合は「各停」）と、その駅を始発とする運用があるかどうか（井の頭線で始発となる駅は両端の渋谷と吉祥寺、中間駅では富士見ヶ丘のみ）をダミー変数（その条件に当てはまるものを 1、当てはまらないものを 0 とする変数）として収録したものを与えたデータを作成したので、それで説明しましょう。

```
> dataset2 <- read.csv("keioinokashira_dash.csv",row.names=1,header=T)
> dataset2 <- data.frame(dataset2)
> dataset2
           定期乗車 普通乗車 定期降車 普通降車 停車種別 始発ダミー
渋谷         34975    25778    34975    25890     急行         1
神泉           647     1035      647     1263     各停         0
駒場東大前     4569     2615     4569     2478     各停         0
```

```
（略）
> dataset2$始発ダミー <- factor(dataset2$始発ダミー) # 数値データを因子に置き換え
```

データは、R コマンダーを使うともっと簡単に読み込むことができます。R コマンダーの「データ」メニューの「データのインポート」を使えば、エクセルのファイルはもとより、エクセルのファイルをコピーしたクリップボードのほか、SPSS、SAS のファイルからもデータを読み取ることができます。研究者が公開しているデータには SPSS 形式で公開されているものも多いですが、それも R に適合する形式で読み取ることができるのです。

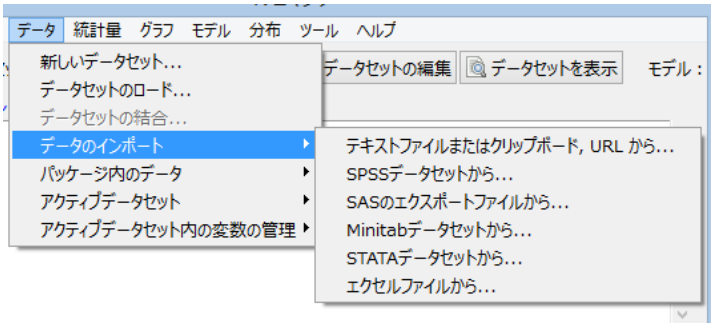

図 1.7 R コマンダーでのデータのインポート

この節の最後に、R コマンダーにおけるデータの扱い方について説明します。R コマンダーでは、各種の分析を行う際には、分析を行うデータをアクティブデータセットとして設定する必要があります。

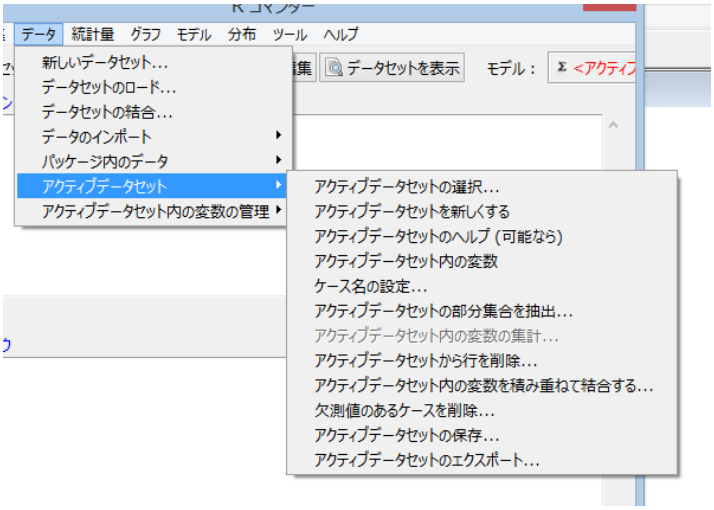

図 1.8 R コマンダーのアクティブデータセットメニュー

R コマンダーにアクティブデータセットとして設定できるデータは、データフレーム形式で保存されているもののみとなります。アクティブデータセットを設定するには、「データ」メニューの「アクティブデータセット」から、「アクティブデータセットの選択 ...」を選びます。そうすると、現在アクティブデータセットとして設定できるデータの一覧が表示されますので、その中からアクティブデータセットにしたいデータを選択します。

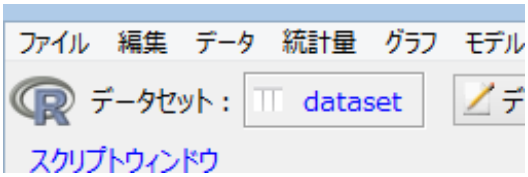

図 1.9 アクティブデータセットが正しく設定された

そして、R コマンダーの上部にある「データセット」という名前の横に、アクティブデータセットにしたいデータの名前が表示されたら成功です。R コマンダーでは、アクティブデータセットとして設定されたデータセットに対して分析やデータの要約が行われます。

第 2 章
統計学の基礎

2.1 平均と分散、標準偏差

データセット $\{x\}(= x_1, x_2, \ldots, x_n)$ について、

$$\text{平均 } E(X) = \bar{x} = \frac{1}{n}\sum_{k=1}^{n} x_k = \frac{x_1 + x_2 + \cdots + x_n}{n} \tag{1}$$

$$\text{分散 } V(X) = s_x^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \bar{x}))^2 \tag{2}$$

$$= E(X^2) - E(X)^2 \tag{3}$$

$$\text{不偏分散 } {s'}_x^2 = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \bar{x}))^2 = \frac{n}{n-1}s_x^2 \tag{4}$$

$$\text{標準偏差}\sigma(X) = s_x = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(x_k - \bar{x}))^2} \tag{5}$$

式 3 の証明

$$\begin{aligned} V(X) &= \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n}\sum_{i=1}^{n}\left(x_i^2 - 2\bar{x}x_i + \bar{x}^2\right) \\ &= \frac{1}{n}\sum_{i=1}^{n}x_i^2 - 2\bar{x} \times \frac{1}{n}\sum_{i=1}^{n}x_i + \frac{1}{n}\sum_{i=1}^{n}\bar{x}^2 \\ &= E(X^2) - 2E(X) \times E(X) + E(X)^2 = E(X^2) - E(X)^2 \end{aligned} \tag{6}$$

主な性質（a, b：定数）

$$E(aX + b) = aE(X) + b \tag{7}$$

$$V(aX + b) = a^2V(X) \tag{8}$$

平均と分散。標準偏差は、データの性質を示す統計量（要約統計量）のなかでももっとも基本的なものです。平均はデータ全体をならしたもので、分散はデータがどれだけばらついているかということを示す指標です。

データのばらつきを示す指標は、ここで紹介した分散のほか、平均絶対偏差（データと平均値の差の絶対値の平均をとったもの）もあります。

そのほか、分散は 2 次のモーメントと言いますが、3 次のモーメントを標準偏差の 3 乗で割ったものを歪度、4 次のモーメントを標準偏差の 4 乗で割ったものを尖度と言います。

Rでは平均はmean、分散はvar、標準偏差はsdで求めることができます。ただし、注意してほしいのは、Rでの分散を求めるコマンドで産出されるのは不偏分散であり、また標準偏差を求めるときは不偏分散の平方根が出ます。

```
> a <- c(1,2,3,4,5)
> mean(a) #平均を求める
[1] 3
> var(a) #不偏分散を求める
[1] 2.5
> sd(a) #標準偏差（不偏分散の正の平方根）を求める
[1] 1.581139
> (1+2+3+4+5)/5 #平均の検算
[1] 3
> ((1-3)^2+(2-3)^2+(3-3)^2+(4-3)^2+(5-3)^2)/4 #不偏分散の検算
[1] 2.5
```

またRでは、データフレーム形式のデータに対して、それぞれの列ごとのパラメータの平均値などを求めることも可能です。**summary(データフレーム)** というコマンドを使うと、数値で提供されているデータに対しては、平均などを、また因子で提供されているデータに対してはそれぞれの数を出すことができます。このようにしてデータの大まかな特徴を把握することも、Rでは可能なのです。

```
> dataset2 <- data.frame(read.csv("keioinokashira_dash.csv",header=T,row.names=1))
> dataset2$始発ダミー <- factor(dataset2$始発ダミー)
> summary(dataset2)
    定期乗車        普通乗車        定期降車        普通降車      停車種別   始発ダミー
 Min.   :  560   Min.   :  641   Min.   :  560   Min.   :  632   各停 :11   0:14
 1st Qu.: 1170   1st Qu.: 1257   1st Qu.: 1170   1st Qu.: 1263   急行 : 6   1: 3
 Median : 2700   Median : 2403   Median : 2700   Median : 2416
 Mean   : 5424   Mean   : 4367   Mean   : 5424   Mean   : 4391
 3rd Qu.: 4569   3rd Qu.: 2904   3rd Qu.: 4569   3rd Qu.: 2815
 Max.   :34975   Max.   :25778   Max.   :34975   Max.   :25890
```

summaryコマンドで要約統計量として表示されるのは次のものです。

Min.：最小値
1st Qu.：第1四分位点（データ全体のうち、下位25%にあたる点）
Median：中央値
Mean：平均
3rd Qu.：第3四分位点（データ全体の内、上位25%にあたる点）
Max..：最大値

なお、分散の値については、次の節で説明いたします。