Ｒ講習会

１．Ｒのインストール（日本語版）

文章としては URL の入力の形でかかれていますが，このファイルで該当ＵＲＬをダブルクリックすると同等のことができます．

(ア) Internet Explorer を起動する．

(イ) ftp://ftp.u-aizu.ac.jp/pub/lang/R/CRAN/bin/windows/base/を上のＵＲＬの欄に打ち込み，キーボードの Enter を押す．

(ウ) R-2.3.0-win32.exe をそのままデスクトップにドラッグする．[1]

(エ) ダウンロードが始まるのでそのまま待つ．

(オ) ダウンロードが終わったら，デスクトップ上の R-2.3.0-win32.exe のファイル（Internet Explorer のウインドウの裏に隠れていることが多いのでそのウインドを最小化すると見つけやすい）をダブルクリックして，でてきたウインドウの「Next>」ボタンをクリック，次のウインドウの「I accept the Agreement」ラジオボタンをクリックして「Next>」ボタンをクリック，次のウインドウは「Next>」がでたらそれをクリックし，次の「Select Components」のウインドウで「Version for East Asian languages」がでてくるまで，スクロールバー動かす．そして，そこのチェックボックスをクリックし，「Next>」をクリック．以降のウインドウでは，「Next>」がでたらそれをクリックすることを繰り返す．

（参考 URL:
http://www.okada.jp.org/RWiki/index.php?R%20%A4%CE%A5%A4%A5%F3%A5%B9%A5%C8%A1%BC%A5%EB ）

(カ) 日本語化します．http://r.nakama.ne.jp/R-2.1.1/config/win32/ を Internet Explorer の URL の欄に入力し，キーボードの Enter をおす．Rconsole と Rdevga のファイルをデスクトップにドラッグすると，ダウンロードが行われる．ただし，設定によっては，デスクトップにインターネットリンクしかできない場合がある．それを確認するには，デスクトップ上の当該ファイルを右クリックしてプロパティを選ぶ．そこに，ファイルの種別が書いてあるので，それがインターネットリンクであったならば，次の手順をとってダウンロードする．Rconsole と Rdevga のファイルを右クリックして，「対象をファイルに保存」(IE)「リンク先を名前をつけて保存」（Netscape,Mozilla,Firefox）にマウスポインタを持って行った後左クリックする．

---

[1] Ｒは日々バグフィックスと改良が行われているので，ヴァージョン番号が上がる可能性がある．その場合は，R-X.X.X-win32.exe の X.X.X を最新のヴァージョン番号に読み替えること．この形のファイル名のものが正式リリース版である．また，R-X.X.XRC-win32.exe(リリースキャンディデート<Release Candidate>)［正式リリース直前のかなり安定した状態のソフト］は，最新バージョンの正式リリース版がなければダウンロードしてよい．

それを使って，デスクトップ上にダウンロードする．ダウンロード後，それらを”?:\Program Files\R\rw2011\etc”に上書きコピー．（？の部分はそれぞれのコンピュータで異なります）ここのあと，必要ならば拡張子”.txt”を，ファイルを右クリックして「名前の変更」を選択して，変更する．

(キ)デスクトップ上にできたR2.1.1のアイコンをダブルクリック．無事，日本語化したRが起動します．どこまでが日本語化されたかに関しては，http://www.okada.jp.org/RWiki/index.php?R%A4%C8%C6%FC%CB%DC%B8%EC を参照のこと．

２．データの入力

２．０　はじめの注意

・ Rは大文字と小文字を区別する．

・ read.tableとRead.tableはことなる．エラーになったときは，スペルミスとともに大文字・小文字の間違いをチェック．

・ ヘルプの活用

とにかくヘルプを活用する力がコンピュータ操作の習得には重要．ヘルプさえ使えれば，最小限のことを頭に入れておくだけで十分．

help(名前)

名前のコマンドについてヘルプを検索．

コマンドではない，例えば，繰り返しを制御するforについて調べたいときは，help(“for”)と“ ”でくくる必要がある．

help("lm", try.all.packages=TRUE)

全てのパッケージについて，lmの意味を検索する．今のところロードされていないパッケージについても探してくれ，探した結果を表示してくれる．もし，すでにロードされているパッケージにある場合は，直ちにそのヘルプを表示してくれる．

help(“lm”, package=stats)

前記のコマンドでそのコマンドを含むパッケージが発見できたなら，そのパッケージ名を指定してヘルプを出すことができる．これは，そのパッケージがロードされていなくても可能．

help.search(“名前”)

名前を含むコマンド，組み込みデータを探してくれる．望みのものを発見したら前記のコマンドを打つ．

help.search("名前", package="パッケージ名")

特定のパッケージに限定して名前を持つコマンド，組み込みデータをさがす．

library()

現在インストールされているパッケージ名一覧

help(package="stats")

特定のパッケージに含まれるコマンド，組み込みデータの一覧を得る．

help.start()

標準のドキュメントをブラウザーインタフェースで出力しれくれる．トップページから，Search Engine & Keywords をクリックすると，検索エンジンに飛べる．上記のコマンドベースの検索は実は，ブラウザーインタフェースで全部できる．

・ 組み込みデータの活用

data()

現在ロードされているパッケージ内のすべての組み込みデータの一覧と説明

data(AirPassengers)

実際にパッケージからデータをロードする．

?AirPassengers

データの説明を出す．データの参照は，AirPassengers でできる．例えば，x<- AirPassengers など．

data(package = .packages(all.available = TRUE))

全てのインストールされているパッケージに関して使用できる組み込みデータを検索．

data(SP500, package="MASS")

特定のパッケージを指定して組み込みデータをロードする．

help(SP500, package=”MASS”)

その説明を求める．

２．１　概要

基本的にはテキストファイルを読むことができる.ポイントはデータフレームを利用すると言うこと．データとしては以下のようなテキストファイルを作ればよい．

	nikkei225	sp500
Aug-90	25978.37	322.56
Sep-90	20983.5	306.05
Oct-90	25194.1	304
Nov-90	22454.63	322.22
Dec-90	23848.71	330.22
Jan-91	23293.14	343.93
Feb-91	26409.22	367.07
Mar-91	26292.04	375.22

Apr-91	26111.25	375.34
May-91	25789.62	389.83
Jun-91	23290.96	371.16
Jul-91	24120.75	387.81
Aug-91	22335.87	395.43

[以下省略]

このファイルの形式について説明しよう．例にとったのは，『計量ファイナンス分析の基礎』の中の日経２２５とＳ＆Ｐ５００の値の列である．

列ごとに各変数のデータが縦に並ぶ．第１行目に各列の変数名が入っている．これを列ラベルと呼ぶ．これは，日本語でもよい．例えば，

	日経２２５	Ｓ＆Ｐ５００
Aug-90	25978.37	322.56

と全角の文字で指定してもいい．その場合も，日本語化されたＲでは変数名としてＲのコマンド内で指定することができる．ただ，日本語変数名は打ち込みづらいので，このように半角英数字で書くのが楽かもしれない．一方，日本語変数名の場合，データの見直しやＲのコマンド，プログラムが何をやっているか，後で解析するときにやりやすい．

では，第１行目の例のデータでは年月を示す列は何を示しているのだろうか．これは，データを与える対象の名前を指定すると考えればよい．この名前を行ラベルと呼ぶ．最初の行のデータは，９０年８月のそれぞれのインデクスの値であるが，Ｒではそのデータを与える対象が Aug-80 という名前で参照できるのである．第一行がデータを与える対象の名前を示していることを示すために，第１列の第１行目は空白を入れてある．

もし，この第１行，第１列も埋めてしまった，すなわち，以下のようなファイルを読み込ませた場合，何が起きるだろうか．

date	nikkei225	sp5000
Aug-90	25978.37	322.56

この場合，各行で与えられるデータを示す対象の名前は，Ｒが自動的に番号を振ることによって付ける．例えば，第１行のデータに対しては，１という対象名，行ラベルを与える．つまり，行番号がそのまま対象の名前になるのである．そして，第１列はデータを示していることとなり，その変数名は date となる．

いずれの形式にせよそのようなファイルが入手できる場合は次の手順に進む．そうでない場合はデータを加工してＲに読み込ませなければならない．

２．２　Excel によるデータの作成，加工とＲへのデータ渡し

このようなデータを作るために，Excel でデータを操作して，「名前を付けて保存」する．その時に，「テキスト（タブ区切り）」を下のメニューリストから選択して，保存する．かりにそのファイル名を ” exer1.txt ” とする．書き出したフォルダ場所を覚えておく．（実習ではデ

スクトップ上に「R」という名前のフォルダを作り，そこに書き出す）

実習では，僕の大学のＨＰから株価のデータの入ったエクセルファイルを読み込んできて，上記の操作をする．(URL は http://ramsey.econ.osaka-cu.ac.jp/~Nakagawa/data/exer1.xls)

２．３　Ｒ関係のデータのありかの指定

R をデスクトップ上の「R 2.1.0」のアイコンをダブルクリックして起動．メニューバーのファイルメニューを選んで，その中のディレクトリの変更を選ぶ．そして，Browse ボタンを押して，書き出したフォルダの場所を指定する．（実習ではデスクトップ上の「R」を指定する）

２．４　データをＲに読み込ませる．

以下では，各コマンドに番号を付ける．これは，実習の際，コマンドを参照しやすくするためである．従って，実際にＲに打ち込むコマンドは[番号]を取り除いた部分である．また，行の終わりの方の#以降はそのコマンドの説明であるが，ここも打ち込む必要はない．

```
[1] x<-read.table("exer1.txt")    # データフレームの読み出し．データ名が１列目
```

これは，２．１でのべた最初のタイプのデータフレームの読み込みに使用する．X というオブジェクト，つまり，データの入れ物に代入される．この代入が<-という２文字合わせた一種の矢印記号で表されているのである．代入には->も許されている．つまり，

```
read.table("exer1.txt")->x
```

でも許される．

これは，この場合は，行ラベルといわれるデータを与える対象名がテキストファイルに入っているのである．もし，用意したデータが第２のタイプの場合，すなわち，行ラベルが指定しされていない場合は，

```
x<-read.table("exer1.txt",header=TRUE)    # header=TRUE を忘れないこと
```

とする．このときは，R が自動的に行ラベルを１，２，３と行番号に従って振ってくれる．なお，Ｒの規則として関数の引数の名前（値ではない）には省略が許される．だから，header と打ち込むのが面倒くさい場合，

```
x<-read.table("exer1.txt",h=TRUE)
```

としてもよい．ただし，関数名にはこのような省略は許されないので，

```
x<-read.t("exer1.txt",h=TRUE)
```

としてはだめである．また，TRUE は T と省略しても良い．従って，

```
x<-read.table("exer1.txt",h=T)
```

でもいける．しかし，これは，関数引数名の省略とは全く別のメカニズムで許されているので，注意を要する．なお，これらの短縮入力は１回限りの処理には楽だが，後から何をやったかを解析する場合困るので，極力使用しないのが望ましい．

なお，以下関数の細かい内容を知るためには，

help(コマンド名)

help(“コマンド名”)

によって，ヘルプファイルを呼び出す．ただし，英語なので，日本語訳をみるという方法もある．
http://www.is.titech.ac.jp/%7Emase/mase/html.jp/html.jp.20020513.tar からダウンロードしてみる．

２．５　データフレームの参照

データフレームオブジェクトに代入されたデータの中身をみよう．

```
[2] x                              # データの中身をみる
```

これで

```
          nikkei225   sp500
Aug-90    25978.37 322.56
Sep-90    20983.50 306.05
Oct-90    25194.10 304.00
Nov-90    22454.63 322.22
          [以下省略]
```

と出力される．オブジェクト x に何かが入っているのはわかるが，いったい何が入っているのかみてみよう．

```
[3] x$nikkei225                    # データフレームの列参照
```

結果の出力は以下の通りである．

```
 [1] 25978.37 20983.50 25194.10 22454.63 23848.71 23293.14 26409.22 26292.04
 [9] 26111.25 25789.62 23290.96 24120.75 22335.87 23916.44 25222.28 22687.35
                          [以下省略]
```

これは，日経２２５平均株価の列が参照されている．なお，データフレーム内の参照についても，省略が可能である．

x$n

あるいは，

x$nikkei

でもいい．このデータがどうなっているか，みてみよう．

```
[4] x$nikkei225[3]                 # 列はベクトルとなっている
```

この結果，

[1] 25194.1

と出力される．この[1]というのは，ベクトルの第１要素という意味である．従って，結果は一つの要素しかないベクトルである．前のコマンドに対する[1]や[9]という出力のラベルは，第何要素かを示しているのである．なぜ結果ベクトルの１要素がさらに１要素だけのベクトルになっているのか不思議に思うかもしれない．これは，Ｒの場合，何次元もの配列（array）が許されていて，２次元の配列を行列（matrix），１次元の配列をベクトル（vector）としているためである．したがって，このベクトルのベクトルと考えると，２次元の配列＝行列ができ，さらにそのベクトルと考えると３次元配列となるというように再帰的に定義していけるのである．
そのことは以下のコマンドを打つとわかる．

```
[5] x[3,1]                    # 行列としても参照できる
```

その出力結果は，

[1] 25194.1

で，先ほどの[4]のコマンド，x$nikkei225[3]と同じ結果になる．つまり，x[行番号，列番号]をとして参照する要素を指定する．ただし，注意すべきなのは，１行目は，テキストファイル内の第１行目ではなく，第２行目になる．つまり，変数名を入れた行はとばして，本当のデータの１番目が１行目として参照されるのである．また，第１列目も，テキストファイル内の第１列目，すなわち，データを与える対象名を指定する列をとばして，次の本当のデータの列を１列目とみなす．x[3,1]は，Oct-90 のデータで日経２２５のデータを参照する．なお，この結果をほかのオブジェクトに代入することができる．その場合，

```
y<-x[3,1]
```

とする．基本的には，出力されたものが代入されると考えて良い．これによって，出力結果を他のコマンドで再利用できるである．
x の列だけ，および，行だけを参照してみよう．

```
[6] x[,2]                     # SP500 の列ベクトル
```

出力は以下の通りで，Ｓ＆Ｐ５００平均株価のデータがベクトルとして与えられている．

```
 [1] 322.56 306.05 304.00 322.22 330.22 343.93 367.07 375.22 375.34 389.83
[11] 371.16 387.81 395.43 387.86 392.45 375.22 417.09 408.78 412.70 403.69
```

[以下略]

次は，行を参照する．

```
[7] x[10,]                    # 行を参照．
```

その結果は，以下の通りである．

```
                              nikkei225  sp500
May-91  25789.62 389.83
```

ここから，わかるのは行もまたデータフレームになっていることである．その証明として，以下のコマンドを打ってみよう．

```
[8] x[10,]$nikkei225
[9] x[10,][1,]
```

行列としての参照において，列の参照を数字以外で指定することもできる．

```
[10] x[,”nikkei225”]          # データフレームの威力は変数名で参照できる
```

これで日経２２５の列を参照できる．その効果は，x$nikkei225 と同じである．データフレームの威力は，変数名による参照だけではない．データを与える対象の名前でも参照できる．そのために，行ラベルを付けるのである．第１列で指定した行ラベルを活用してデータを参照する．

```
[11] x[“May-91”,]             # 何番目のデータだけではなく，行ラベルでも参照可
```

これで，９１年５月のデータがデータフレームとして参照できる．当然，行，列の参照の両方をラベルでも参照できる．

```
[12] x[“May-91”, “nikkei225”] # ９１年５月の日経２２５．一種のデータベース．
```

さらに，日経２２５平均株価が２５０００円を超えた月を表示させてみよう．行の選択の条件を行を指定する部分に指定することでできる．

```
[13] x[x$nikkei225 > 25000,]  # 最後の,]を忘れないように
```

結果は，

```
                              nikkei225  sp500
Aug-90  25978.37 322.56
```

```
Oct-90  25194.10 304.00
Feb-91  26409.22 367.07
Mar-91  26292.04 375.22
Apr-91  26111.25 375.34
May-91  25789.62 389.83
Oct-91  25222.28 392.45
```

である．なお，負の数―０．３より小さいデータを選ぼうとして“x$nikkei225<-0.3”とやってしまうと x の中の nikkei225 の列全てに 0,3 が代入されてしまう．だから，必ず不等号と数字の間にはスペースを入れるようにしよう．

さて，入力ミスを行った場合や今まで実行させたのと似たコマンドを入力する場合に，今までのコマンドを入力行に呼び出したいことがある.その時は,上矢印キーと下矢印キーを利用する．また，２．３で指定したフォルダの.Rhistory ファイル内に前回Ｒを起動してから終了させるまでの間に入力されたコマンドの一覧が保存されている．それを利用することもできる．

２．６　記述統計

まず，いちいちデータフレームの名前を指定しないでいいように，名前を探す範囲を指定しておく．具体的にはデータフレーム x 内の変数名は x$をいちいち指定しなくても参照できるようにする．

```
[14] attach(x)                    # データフレームのカラム名だけで参照する．
```

これで，オブジェクト名に加えて x の各列名も「nikkei225」という名前を探す範囲に含まれる．その結果，nikkei225 という名前が x$nikkei225 と同等に使用できるのである．
注意点は，attach(x)を実行する前には x$n で参照できたが，実行してしまったら，

```
n
```

とやっても日経２２５のデータを参照できなくなるということである．なぜかは，ややこしい理由なので，強いて知る必要はない．知りたい人は，文書の中の「言語マニュアル」を参照してほしい．むしろ，みだりに変数名を省略してはいけないという教訓をくみ取ってほしい．

まず，散布図を書いてみよう．

```
[15] plot(sp500,nikke225)              # 散布図
```

あるいはこれでもいい．

[16] plot(x)　　# これでもできる

なお,二つの変数しかないデータフレームではなく,３つ以上変数を持っている場合については,僕のＨＰから（http://ramsey.econ.osaka-cu.ac.jp/~Nakagawa/data/divorce.xls）のデータをデスクトップ上の“Ｒ”というフォルダに書き出す．その後，それを開いて，必要な箇所だけに編集して，「名前を付けて保存」するが，そこで，メニューリストから「テキスト（タブ区切り）」を選択して，同じ場所に書き出す．その後は，

[17]　devorce <- read.table(“devorce.txt”)

[18]　plot(devorce)

を実行する．あるいは，

[19]　plot(read.table(“devorce.txt”))

でもよい．

株価そのものの相関を考えても意味がなさそうなので，収益率同士の相関を見てみる．そのためには，データフレームの中のデータを収益率に変える必要がある．そのために，

[20] dlognikkei<-diff(log(nikkei225))　　連続（複利）収益率の計算

というコマンドを実行する．日経２２５のデータを収益率に経関するために，log()という関数を実行して,日経２２５の log 値を計算し,diff()という関数で,その結果の差分をとっている．結果として日経２２５の複利収益率を得ることができる．これを，S&P５００のデータに対しても実行すればよいのだが,それらをまとめてできないものか？実は以下のコマンドでできるのである．

[21] dlogx<-apply(log(x), 2, diff)

　　# 一つ一つのカラムを操作するのは面倒なのでまとめて

さらにこれを

実際うまくいっているかどうか確認するためには，

[22] dlogx　　# うまくいっているようにみえる．

を実行させればよい．先ほどの[15]を説明しよう．

まず，関数は行列や配列を引数にとると，それ全体をまとめて処理するものと，行または列単位で処理するもの，そして，要素単位で処理するものに分かれる．log()をとるという操作は，もともとスカラーに対してそのログをスカラーという形で返す関数である．従って，要素単位で

処理する関数ということになる．この場合，行列や配列を引数にとると，行ラベル，列ラベルをのぞいたその全要素について log をとったものを要素とする行列，配列が返される．従って，log(x)はすべての株価データの log を計算する．それに apply(,2,diff)という操作をしているのであるが，apply はどういうことをしているのか？そのためには，diff 関数の性質を考える必要がある．diff 関数は，行，または，列のデータを数列とみなして，その差分を計算するから，第２の行，または，列単位の処理をする関数である．apply 関数の，第３引数に指定できるのは，このタイプの関数である．いまやりたいのは，log(x)の結果の全ての列に対して，diff 関数を作用させると，連続収益率の値が入った配列が得られるのであるから，このような指定をする．つまり，apply(対象となる行列・配列・データフレーム名，行を対象にするなら１・列を対象にするなら２，関数名)と指定するのである．この結果はどのように参照できるのかみてみよう．

```
[23] dlogx$nikkei                    # しかし，データフレームではない．
```

これは，エラーになる．よって，データフレームではない．要注意である．

```
[24] dlogx[,1]                       # 行列だよ
```

これはエラーにならない．従って，行列である．試しに，

```
[25] is.matrix(dlogx)
```

とうつと，行列であるので TRUE との答えが返ってくる．dlogx を表示させたときの第１列目の日付はどうやって参照できるのであろうか？

```
[26] dlogx[,0]                       # ０列目にはデータフレームのデータ名が入る
```

ここからわかるように，0 列目が日付になっている．さらに，

```
[27] rownames(dlogx)
```

とやると日付が参照できる．また，列の名前，つまり，変数名も

```
[28] colnames(dlogx)
[29] dlogx[0,]
```

でみえる．ここから，データフレームと同様な dlogx$nikkei の参照はできないけれども，

```
[30] dlogx[,2]                  # ２列目は SP500
[31] dlogx[,"nikkei225"]        # カラム名で参照もできる
[32] dlogx["Jan-97",1]          # ということはデータ名でも参照できる．
```

などのデータフレームの参照で活用したコマンドは同様に使えるのである．
　しかし，データフレームに変換しておくと便利なので，

```
[33] dlogx<-data.frame(dlogx)
```

としよう．これは，行列をデータフレームに変換するコマンドである．
　そこで，収益率同士の散布図を書いてみよう．

```
[34] plot(dlogx)                # 散布図．行列の名前を指定すれば自動的にやってくれる
```

ついでに，

```
[35]    pairs(dlogx, panel=panel.smooth)
```

とすると，散布図に移動平均で書いた平滑化回帰線も書き入れてくれる．
　次は，平均値，中央値，最大値，最小値を求めることである．

```
[36] attach(dlogx)                      # カラム名のみで参照
```

ここで気になるのが,さっきattach(x)でnikkei225がx$nikkei225を表すようにしていたのだが，このattach(dlogx)で何を参照することになるのかである．答えは，直近のattach()で宣言されている方が先に検索にヒットするので，dlogx$nikkei225が参照される．

```
[37] mean(nikkei225)                    # 日経の月次収益率の平均
[38] median(nikkei225)                  # 中央値，対応する日付，つまり行ラベルも表示
[39] max(nikkei225)                     # 最大値
[40] max(sp500)                         # 最大値
[41] min(nikkei225)                     # 最小値
```

これらは，各列にコマンドを入力しようというものである．ついでに，最大値，最小値がいつだったのかを知りたいとすると，

```
[42] rownames(dlogx)[nikkei225 == max(nikkei225)]
[43] rownames(dlogx)[nikkei225 == min(nikkei225)]
```

とすればよい．==は，等しいという比較のための演算子である．ここまでは，各列の平均を一つ一つ入力して計算しようとしていたが，各列の平均値をまとめて出したいとする．まず，

```
[42] min(dlogx)                    # 各列の最小値ではなく，データ全体の最小値
[43] mean(dlogx)                   # これはデータフレームの場合各列に対して計算
                                   # もし行列なら全てのデータに対する平均
```

とやってみるが，実は行列全体の平均，最小値を求めているだけであることに気づく．平均は，列単位でも，行列全体でも求めることができる．引数として指定されたもの中で一番大きいくくりで，その中での平均，最大，最小を求めてしまう，単位の処理を行う関数applyを使用するのがよい．

```
[44] apply(dlogx, 2, min)          # そのときは列データそれぞれに対して関数を操作
```

平均，最大，最小，中央値はこれでできるが，不偏分散，不偏標準偏差，不偏共分散，相関係数はどうだろうか．こちらは考えてくれる．

```
[45] var(nikkei225)        # 不偏分散
[46] var(dlogx)            # 行列にvarをとると列データを各変数とみたときの共分散
[47] sd(nikkei225)         # 日経の不偏標準偏差
[48] sd(dlogx)                     # 行列に対するsdは各列の標準偏差
[49] cov(nikkei225, sp500)         # 不偏共分散
[50] cov(dlogx)                    # var(dlogx)と同じ
[51] cor(nikkei225, sp500)         # 相関係数
[52] cor(dlogx)                    # 列データの相関係数の組み合わせ行列
```

次は，ヒストグラムの作成である．まず単純にデータ数，つまり，頻度をとったヒストグラムを書いてみる．

```
[53] hist(nikkei225)               # ヒストグラムを作る
```

頻度ではなく密度（つまり相対度数／階級幅）で表示したい場合は，

```
[54] hist(nikkei225,freq=FALSE)
```

とすればよい．

しかし，ここから分布の連続した確率密度を推定したいと思うこともある．その場合，

```
[55] plot(density(nikkei225))
```

とする．

さて，グラフを表示するウインドウは次々と切り替わってしまう．二つのグラフの比較をしたい場合どうしたらいいだろうか．前のグラフを，ファイルに書き出すか，クリップボードに納めるしかない．ファイルに書き出す場合，グラフを右クリックし，「メタファイルに書き出し」か「ポストスクリプトファイルに書き出し」を選び適当なところに書き出す．どちらを選ぶかは，書き出したファイルをダブルクリックして見えるかどうかに依存する．見える方を選択すればよい．どちらも見えない場合は，「ポストスクリプトファイルに書き出し」を選んでおいて，後で，Ghostscript 8.51 + GSview 4.7をダウンロードしてインストールする．

（ftp://akagi.ms.u-tokyo.ac.jp/pub/TeX/win32-gs/gs851w32full.zip，
ftp://ftp.riken.go.jp/pub/CTAN/support/ghostscript/ghostgum/gsv47w32.exe）

密度推定とヒストグラムを重ね書きする場合は以下のようにする．

```
[56] par(usr=c(-0.3,0.2,0,7))
[57] hist(nikkei225, freq=FALSE, xlim=c(-0.3,0.2), ylim=c(0,7))
[58] par(new=TRUE)
[59] plot(density(nikkei225),xlim=c(-0.3,0.2), ylim=c(0,7),xlab="", ylab="", main="")
```

これらを，一度気に入力する場合は，;で区切って入力すればよい．

さて,[56]はグラフの範囲を横軸は-0.3から0.2,縦軸は0から7までだと宣言している.これで，各グラフはこの範囲外にはみ出しても,この範囲に入る部分だけがかかれることになる.そして，[57]でヒストグラムを書いているが，xlimで横軸の範囲を-0.3から0,2まで，ylimで縦軸の範囲を0から7に指定してその範囲内のみを書かせている．[58]は重ね書きを指定して，[59]で推定密度関数を書いている．xlab=”” ,ylab=”” はそれぞれx軸，y軸のラベルを書かないと言うことを指定している．重ね書きなので，書かないのがいいのである．もし書く場合は，xlab=”横軸”などのように書きたい文字列を指定する．main=”” はグラフのメインの表題も書かないことを示している．表題を入れる場合は，xlab=”題”などと直接文字列を指定する．これらのラベルや表題の指定は，他のhistなどのグラフ関数に共通なので積極的に利用したい．次に，S&P500の収益率の分布もみよう．

```
[46] hist(sp500,freq=F,xlim=c(-0.1,0.2), ylim=c(0,15))
[47] par(new=TRUE)
[48] plot(density(sp500),xlim=c(-0.1,0.2), ylim=c(0,15),xlab="", ylab="", main="")
```

最後に，月次収益率の分布を正規分布と比較してみよう．

```
[49] library(stats)
[50] ec<-ecdf(dlogx[,"nikkei225"])        # 経験分布関数
[51] plot(ec)                             # グラフ
```

経験分布関数とは，（横軸座標，縦軸座標）として$\left(t, \frac{\text{t以下のデータ数}}{\text{標本数}}\right)$をプロットしたものである．しかし，これでは正規分布との違いがよくわからないない．そのために，Q-Qプロットといわれるものを描く．$\Phi(t)$を標準正規分布の確率分布関数とする．よって，Xが正規分布に従い，その平均がμ，標準偏差がσとすれば，

$$\Pr\left(\frac{X-\mu}{\sigma} \le t\right) = \Phi(t)$$

である．よって，

$$\Pr(X \le \mu + t\sigma) = \Phi(t)$$

である．標準正規分布のqパーセンタイルをz_qとすると，

$$\Pr(X \le \mu + z_q\sigma) = \Phi(z_q) = q$$

となり，$\Phi(z_q) = q$より$z_q = \Phi^{-1}(q)$となるので，$\Pr(X \le \mu + \sigma\Phi^{-1}(q)) = q$．

つまり，XのqパーセンタイルX_qは$\mu + \sigma\Phi^{-1}(q)$である．従って，$(X_q, \Phi^{-1}(q))$をプロットすると線形になるはずである．qは$\frac{\text{データ値以下のデータ数}}{\text{標本数}}$でわかるので，結局，データが正規分布に従っていれば，$\left(\text{データ値}, \Phi^{-1}\left(\frac{\text{データ値以下のデータ数}}{\text{標本数}}\right)\right)$の全てのデータ値に対するプロットは直線に近くなるはずである．これをQ-Qプロットと呼ぶ．これを描いてみよう．（なお，Rでは縦軸と横軸が逆になっている．従って，

$\left(\Phi^{-1}\left(\frac{\text{データ値以下のデータ数}}{\text{標本数}}\right), \text{データ値}\right)$をプロットする．条規解説通りの図を得たい場合は，qqnorm(nikkei225, datax=T)とする）

```
[52] qqnorm(nikkei225)    # Q-Qプロット
[53] qqline(nikkei225)    # 正規分布ならQ-Qプロットは直線．参考線を書く
[54] qqnorm(sp500])       # SP500
[55] qqline(sp500)

[56] q()
```

「作業スペースを保存しますか」に対して OK を答えよう．次から，作業スペースを保存した.RData というファイルをダブルクリックすることで作業を再開することができる．このとき，今まで入力した全てのコマンドの効果が及んだ状態で再開することができる．

[課題]

（１） devorce のデータについて，最大，最小の離婚率の県をもとめよ．また，離婚率が中央値に相当するデータを求めよ

（２） HP 上の日経２２５，TOPIX,，円ドルデータをダウンロードして，それぞれの収益率の分布が正規分布かどうか Q-Q プロットを利用して判定せよ．

２．７　回帰分析

定数項ありの回帰を行う．

```
[2.7.1] fm <- lm(nikkei225~sp500, data=dlogx)              # 回帰分析をする
```

最後 data=dlogx は，どのデータフレームを使うかを示している．注意点は，回帰式の指定の仕方である．この場合は，被説明変数が nikkei225，説明変数が sp500 であると指定している．その間の~が，被説明変数と説明変数の区切りである．複数の説明変数を指定する方法は，

被説明変数~説明変数+説明変数

である．また，定数項がない場合は

被説明変数~説明変数+説明変数-1

と指定するか，

被説明変数~説明変数+説明変数+0

と指定するかである．fm に回帰分析の結果が格納される．これを再度関数の引数として与えた

り，様々な計算を行うことで，結果の加工が行われる．散布図と回帰直線を重ね書きする．その次に，回帰の結果を出力する．

```
[2.7.2] plot(nikkei225~sp500, data=dlogx)          # 散布図
[2.7.3] abline(fm)                                 # 散布図に推定された回帰直線を書く
[2.7.4] summary(fm)                                # 回帰分析の結果の出力
```

出力結果は，以下のようになっている．

```
Call:
lm(formula = nikkei225 ~ sp500, data = dlogx)

Residuals:
     Min       1Q   Median       3Q      Max
-0.16464 -0.03918 -0.00697  0.04264  0.20006
```

（ここは，残差の５数要約といわれるものである．残差の，最大値，第１四分位数，中央値，第３四分位数，最大値を表示している．）

```
Coefficients:
             Estimate Std. Error   t value Pr(>|t|)
(Intercept) -0.012535   0.008728  -1.436   0.1551
sp500        0.691947   0.273158   2.533   0.0134 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

（Estimate の列は推定値，その次の列は推定値の標準誤差，次の列が t 値，最後の列は p 値である．）

```
Residual standard error: 0.07139 on 75 degrees of freedom
Multiple R-Squared: 0.07881,    Adjusted R-squared: 0.06653
F-statistic: 6.417 on 1 and 75 DF,  p-value: 0.01339
```

（ここには分散分析の結果が出力される．第１行は残差標準偏差，第２行目は決定係数と，自由度修正済み決定係数，第３行目は定数項以外の係数が全て０であるというキム仮説に対する F 検定の検定統計量値とｐ値である．）

次に，回帰分析の診断プロットを書く．

```
[2.7.5] opar<-par(mfrow=c(2,2), oma=c(0,0,1.1, 0)) # 4 面に次のプロットを次々と書く
```

opar に以前のグラフの書き方が待避される．診断図を書いた後に，この opar で図の描き方を元に戻す．

```
[2.7.6] plot(fm)                                    # 回帰の診断図
```

それぞれの図についての説明は後ほど行う．（クック距離 0.5 以上は異質なデータと考えられる．）

```
[2.7.7] par(opar)                                   # ４面プロットから元の１面プロットに戻す．
```

次に，定数なしの回帰を行う．

```
[2.7.8] fm2 <- lm(nikkei225~sp500-1, data=dlogx)
```

または

```
      fm2 <- lm(nikkei225~sp500+0, data=dlogx)
```

と指定する．次は，散布図と回帰直線，そして，結果のサマリーをだすのである．そして，診断結果を出す．

```
[2.7.9] plot(nikkei225~sp500, data=dlogx)
[2.7.10] abline(fm2)                                # 散布図に推定された回帰直線を書く
[2.7.11] summary(fm2)                               # 回帰分析の結果の出力
[2.7.12] opar<-par(mfrow=c(2,2), oma=c(0,0,1.1, 0))         # 4 面に次のプロットを次々と書く
[2.7.13] plot(fm2)                                          # 回帰の診断図
[2.7.14] par(opar)                                  # ４面プロットから元の１面プロットに戻す
```

次に，係数の信頼区間を計算しよう．これは，ややこしい．

```
[2.7.15] sm<-summary(fm)
[2.7.16] sm$coef                            # 係数の推定値，標準誤差，ｔ値，ｐ値の行列
[2.7.17] sm$coef[,1]+qnorm(0.975)*sm$coef[,2]       # ９５%信頼区間上限
         qnorm(パーセント，平均，分散)でパーセント点が求まる
```

点 x での密度関数値は dnorm(x,平均，分散)

点 x での分布関数値は pnorm(x,平均，分散)

n 個の正規分布の乱数を発生　　rnorm(n,平均，分散)

ただし，標準正規分布は引数の 0,1 の部分が省略可能

もし誤差項が正規分布ならば

sm1$coef[,1]+qt(0.975,fm1$df)*sm1$coef[,2]

[2.7.18] sm$coef[,1]-qnorm(0.975)*sm$coef[,2]　　# ９５%信頼区間下限

もし誤差項が正規分布ならば

sm1$coef[,1]-qt(0.975,fm1$df)*sm1$coef[,2]

回帰直線の信頼区間と予測信頼区間

```
[2.7.19] dlognikkei<-dlogx$nikkei225
[2.7.20] dlogsp<-dlogx$sp500
[2.7.21] fm1<-lm(dlognikkei~dlogsp)
[2.7.22] new<-data.frame(dlogsp=seq(-0.06,0.12,0.001))
[2.7.23] dp<-predict(fm1,new,interval="confidence", level=0.95)
[2.7.24] dc<-predict(fm1,new,interval="prediction", level=0.95)
[2.7.25] matplot(new$dlogsp, cbind(dp,dc), lty=c(1,2,2,3,3), type="l")
```

重回帰モデルとモデル選択

```
[2.7.26]  attach(devorce)
[2.7.27]  fm3<-lm(離婚率~ .,data=devorce)

[2.7.28]  summary(fm3)
```

ステップは AIC に基づくモデル選択を行う

```
[2.7.29]  stepm3<-step(fm3)
[2.7.30]  summary(stepm3)
[2.7.31]  fm4<-lm(離婚率~ .^2,data=devorce)
[2.7.32]  summary(fm4)
[2.7.33]  stepm4<-step(fm4)
[2.7.34]  summary(stepm4)
```

回帰診断

残差プロット

分散が不均一か

```
[2.7.35]  plot.lm(stepm4, which=1)
```

```
[2.7.36]  abline(h=0)
```

Q-Q プロット
残差の正規性

```
[2.7.37]  plot.lm(stepm4, which=2)
[2.7.38]  qqnorm(resid(stepm4))
[2.7.39]  qqline(resid(stepm4))
```

梃子比(leverage)
どれだけ回帰直線をその観測値の散布図上に近づけたかをしめす．0.2 以上危険，0.5 以上異質

```
[2.7.40]  xx<-model.matrix(stepm4)
[2.7.41]  lev<-hat(xx)
[2.7.42]  cx<-2*sum(lev)/dim(devorce)[1]
[2.7.43]  cx
[2.7.44]  names(lev)<-rownames(devorce)
[2.7.45]  lev[lev>cx]
```

スチューデント化残差
２より大，または，－２より小が多い場合，ある特定のデータや特定の当てはめ値に偏ってこのようなデータが多かった場合危険

```
[2.7.46]  stdres<-rstudent(stepm4)
[2.7.47]  plot(stdres)
[2.7.48]  abline(h=c(2,-2))
[2.7.49]  names(stdres)<-rownames(devorce)
[2.7.50]  stdres[stdres< -2 | stdres> 2]
```

S-L プロット
スチューデント化残差の絶対値の平方根，横軸は当てはめ値にした，警戒は 0.177,1.497
小さい方もとるのは，そのようなデータは回帰直線をその観測値に引きつけているから．

```
[2.7.51]  sl<-sqrt(abs(stdres))
[2.7.52]  plot(sl)
[2.7.53]  abline(h=c(0.177, 1.497))
[2.7.54]  sl[sl< 0.177 |  sl > 1.497]
[2.7.55]  plot.lm(stepm4, which=3)
```

クックの距離（0.5 以上が影響力多め，1.0 以上影響力がおおい）

```
[2.7.56]  plot.lm(stepm4, which=4)
```

課題

（１） 日経２２５の収益率を被説明変数として，円ドルレートの収益率，出来高を説明変数とする回帰モデルを推定せよ.さらに,step()関数を使ってAICによるモデル選択を行え．なお，診断プロットをみながら，回帰の妥当性を議論せよ．

（２） 離婚率の県別データをもとに，離婚率を被説明変数とした様々な回帰モデルを比較して，離婚率の決定要因を検討せよ．

３．パッケージの活用

Ｒは様々なパッケージが存在し，いろいろな統計処理を容易にしている．一方で，パッケージの質はまちまちであり，その結果の信頼性は必ずしも高くない．しかし，さしあたりの計算のために注意しながら使うには十分であろう．

３．１　パッケージの一覧の取得

英語でいいならば，http://cran.r-project.org/src/contrib/PACKAGES.html に現在使用可能なパッケージの全て説明がある．この中の一部のパッケージについては，日本語の説明が，
http://www.okada.jp.org/RWiki/index.php?CRAN%A5%D1%A5%C3%A5%B1%A1%BC%A5%B8%A5%EA%A5%B9%A5%C8
にある．

３．２　パッケージのインストールの仕方

３．２．１　ネットワーク接続が可能な環境の場合

３．２．１．１　ウィンドウインターフェースを使いたい人

RGui のメニューの「パッケージ」を左クリックすると，メニューリストがでてくるがその中の「パッケージのインストール」を左クリックで選択する．すると，パッケージの一覧が出てくる.この中から，インストールしたいパッケージを左クリックして，下の「ＯＫ」ボタンを押す．（このときパッケージ相互の依存関係も考えてインストールしてくれる）

ただし，最初に上記手順をやるときには，Ｒのパッケージをどのミラーサイトからとってくるかを聞かれるので，まず，日本のサイト，例えば，Japan(Tsukuba)を選ぶ．しかし，計量経済学関係のパッケージがない場合が多いので，インストールできなかったらUS(CA1)に切り替える..

３．２．１．２　コマンドインタフェースを使いたい人

[3.2.1] install.packages(“パッケージ名”,dependencies=T)

でも同様のことができる．この場合も，初めてにインストールという操作をする場合は，Ｒのパッケージをどのミラーサイトからとってくるかを聞かれるので，まず，日本のサイト，そこになかったら，US(CA1)に切り替える．最後の dependencies=T は，パッケージ相互の依存関係も考え

て,そのパッケージを作動させるのに必要な全てのパッケージもついでにインストールせよと言う指示である．

３．２．２　ネットワークに接続できず，パッケージの入ったｚｉｐファイルのみが入手できている場合．

　経済学部のＣＡ教室のように外部にアクセスできない場合やあまり速いネットワーク環境がない場合は，別途入手したパッケージのファイルをインストールすることになる．

３．２．２．１　ファイルの入手の仕方

http://cran.r-project.org/src/contrib/PACKAGES.html からアクセスするのがよいだろう．例えば，

UNF	Tools for creating universal numeric fingerprints for data.
urca	Unit root and cointegration tests for time series data
urn	Urn：Sampling Without Replacement
uroot	Unit root tests and graphics for seasonal time series.
UsingR	Data sets for the text "Using R for Introductory Statistics"

となっているので，”**urca”というパッケージを入手するなら，**urca のリンクをクリックする．すると，

urca: Unit root and cointegration tests for time series data

Unit root and cointegration tests encountered in applied econometric analysis are implemented.

Version:	0.8-2
Depends:	R (>= 2.0.0)
Imports:	nlme, methods, graphics, stats
Date:	2005-04-24
Author:	Bernhard Pfaff
Maintainer:	Bernhard Pfaff
License:	GPL version 2 or newer
URL:	http://www.r-project.org

Downloads:

Package source: urca_0.8-2.tar.gz

Windows binary: urca_0.8-2.zip

Reference manual: urca.pdf

というページが出る．この”Windows binary”の urca_0.8-2.zip を左クリックすればよい．

３．２．２．２　インストールの仕方

その名前を変更して，urca.zip とし（一般的には，_とバージョン番号を取り除くと考えればよい），これを現在の作業ディレクトリに入れる．作業ディレクトリの場所は，

[3.2.2] getwd()

で得られる．作業ディレクトリの変更は，２．３　R関係のデータのありかの指定にも述べたメニューの「ファイル」のメニューリストの中の「ディレクトリの変更」で指定するか，setwd(“ファイルのパス名”)で行う．

[3.2.3] install.packages("urca.zip", CRAN=NULL, destdir=getwd())

によってインストールする．

３．３　パッケージの利用の仕方

パッケージをインストールしたからと言ってすぐに使えるわけではない．パッケージをメモリにロードする作業が必要である．なぜ，インストールとロードが別の手順になっているかを説明しよう．各パッケージはその動作に大きなメモリを必要とするため，インストールしたパッケージを全てロードするとコンピュータの動作を遅くしてしまう（ひどいときにはフリーズさせることさえある）．このような事態を避けるために，必要なときに必要なパッケージだけをロードすることができるようにしてある．

パッケージのロードは

[3.2.4] library(urca)

のように library(パッケージ名)で行う．

ここで注意する点がある．コマンドインタフェースで install.packages()を実行したときに，dependencies=T を付け忘れると，パッケージが単独でインストールされ，その実行に必要な他のパッケージがインストールされていない場合がある．このときは，library()を実行すると，エラーメッセージがでてきて，ロードできない．このときは，install.packages()を dependencies=T をつけて実行するか，メニューから「パッケージのインストール」を選ぶことで対応する．

３．４　パッケージの更新

パッケージは日々更新されている．従って，ある程度の間隔でパッケージの更新を行う必要がある．

update.packages()を実行することで，新しいパッケージがある場合は知らせてくれる．更新したい場合，y で答え，そうでない場合，N で答えればよい．これを実行する場合，メニューの「パッケージ」のメニューリストから「CRAN ミラーサイトの設定」を選んで「Japan(Tsukuba)」などを選んでから，実行しよう．日本語化されているパッケージはこれで最新になるはずである．その後，「CRAN ミラーサイトの設定」で「US(CA1)」に切り替えて，update.packages()を行う．上の出力結果と対照しながら日本語化されたパッケージをアップデートしないように慎重に y を打っていく．

４．リンク

http://www.okada.jp.org/RWiki/　　Ｒの百科事典

http://cse.naro.affrc.go.jp/takezawa/funao.html

舟尾暢男さん作成の講義風のＲ解説ページ．現在本にもなっている．ＰＤＦ版と html 版あり．

http://buran.u-gakugei.ac.jp/~mori/LEARN/R/

東工大間瀬先生の翻訳したＲの文書の日本語訳

http://stat.sm.u-tokai.ac.jp/~yama/R/ 東海大学山本先生のページ

http://aoki2.si.gunma-u.ac.jp/R/　　群馬大学青木先生のページ

http://phi.ypu.jp/swtips/R.html　　群馬大学中澤先生のページ

http://genome.ag.saga-u.ac.jp/R/　　佐賀大学和田先生のページ

http://www.itp.phys.ethz.ch/econophysics/R/

Rmetrics のページ．金融時系列分析のパッケージ”fSeries”，”fSeries”などの開発者のページ．この中にＲを使った金融計量経済学の実習用のテキストがある．

５．文書

http://buran.u-gakugei.ac.jp/~mori/LEARN/R/R-lang.jp.v110.pdf

言語マニュアル．R をプログラム言語としてみた場合の諸注意がある．他の言語を知っている人は疑問を持ったときにこれを参照するとすっきりする可能性がある

６．本

間瀬他（２００４）『工学のためのデータサイエンス入門』数理工学社

船尾暢男（２００５）『The R tips:データ解析環境 R の基本技・グラフィックス活用集』九天社

岡田昌史（２００４）『The R Book:データ解析環境Ｒの活用事例集』九天社