医学における R の利用入門

2006 年 7 月 18 日-19 日　中澤　港（群馬大学大学院医学系研究科）

E-mail: nminato@med.gunma-u.ac.jp

1　R について

R は，GNU General Public License Version 2 に従って[*1]配布されている，オープンソースで拡張性が高いデータ解析環境である。誰でも自由に無料で使うことができるのが最大の利点である。世界中の研究者が GIS を含む空間統計解析やゲノム解析などに至るまでさまざまな追加ライブラリを公開しているし[*2]，自分で新しい拡張関数を付け加えることもできる。

オープンソースということは，誰でもその気になれば計算の中身をインプリメンテーションレベルでチェックできることを意味する。これは，商用ソフトにはありえない利点である。商用ソフトでは，利用している計算式はわかっても，コードそのものは公開されないために，インプリメンテーションにバグがないかどうかは，サンプルデータ解析を実行してクロスチェックすることでしか確認できない場合が多い。R の場合は，世界中の人が使いながらコードチェックもしているので，計算の信頼性もかなり高い。

統計教育のツールとしては，多くの有用なサンプルデータが含まれているので，いちいちデータ入力をしなくても分析手法に合わせて適切なデータを利用できる点も利点といえる。Fisher の iris データのような有名なものは当然として，白血病患者の生存時間データとしてよく使われる Gehan のデータも，`MASS` ライブラリ（推奨ライブラリなので Windows 用バイナリ配布ファイルには含まれている）に `gehan` として入っている。

また，国際協力などの場面でもカウンターパートと共用できる。英文のみならず，仏文，西文などのマニュアルも公開されている。Windows だけでなく，Macintosh でも Linux でも FreeBSD でも動作するので，さまざまな環境で同じ統計解析を行なうことができる。R 以上に各国語対応している統計解析のフリーソフトウェアとして，米国 CDC が提供している EPIINFO があるが（ただし EPIINFO は Windows のみ），利用できる統計解析手法の種類は R の方がずっと多い。R には SPSS でさえ実装されていないような新しい分析手法が多く含まれている。結果の信頼性も高く，最近では多くの学術論文が統計解析に R を使っている[*3]。

R は S 言語にほぼ互換な関数型言語のインタープリタなので，結果はすべてオブジェクトであり，それをシンボルに付値して保存したり加工したりできる。さらに素晴らしいことに，そうやって実行したすべてのプロセスを，テキストファイルとして記録し，保存しておけるので，後になって，どういう分析をしたかをチェックすることができる。しかも，保存しておいたファイルは（例えば `test.R` というファイル名だとすると），`source("test.R")` とすると再実行できる。どんなに複雑な作業をしても，それを何度でも簡単に再現できるということである。逆に考えれば，適当なテキストエディタでプログラムとして R のコマンドを書き連ねておいたものを読み込ませれば，複雑な分析手続きでも 1 回の操作で終わらせることができる。R Commander（`Rcmdr`）というライブラリを使うと，メニュー形式で操作することもある程度可能になるが，その場合でも，ログはきちんと関数リストとして保存される。結果の導出過程を再現できることは学術論文において非常に重要なので，ログも一定の期間は保存しておくべきである。

美しい図を作るのも実に簡単で，しかもその図を `pdf` とか `postscript` とか `png` とか `jpeg` とか Windows 拡張メタファイルの形式 (emf) で保存でき（ただし emf で保存できるのは Windows 版の R のみ），他のソフトに容易に取り込める。例

[*1] 若干，LGPL (Lesser GNU General Public License) Version 2.1 に従うファイルも含まれている。

[*2] CRAN というサイト (`http://cran.r-project.org/`) に集積される仕組みもあるので，CRAN 内で検索すれば大抵の処理は見つけることができる。ちなみに，R バイナリに built-in なライブラリは，base, datasets, grDevices, graphics, grid, methods, splines, stats, stats4, tcltk, tools, utils であり，Recommended なライブラリで，将来全バイナリに built-in される予定なのは，KernSmooth, MASS, boot, class, cluster, foreign, lattice, mgcv, nlme, nnet, rpart, spatial, survival である（Windows 版バイナリには入っていてインストール済みだがロードはされていないので，使うときは `library(survival)` とか `require(survival)` のようにしてロードせねばならない）。`search()` でロード済みライブラリ一覧，`.packages(all.avail=T)` でインストール済み一覧が表示される。ロード済みのライブラリをアンロードするには `detach(package:survival)` などとする。

[*3] 例えば，2004 年 2 月には Nature にも R を使って分析された論文が掲載されている。Morris RJ, Lewis OT, Godfray HCJ: Experimental evidence for apparent competition in a tropical forest food web. Nature 428: 310-313, 2004.

えば win.metafile() 関数を使って emf 形式で保存すれば，Microsoft PowerPoint や OpenOffice.org の Draw などの中で，ベクトルグラフィックスとして再編集することが可能である。

バージョン 2 以降，国際化対応したので，日本の R ユーザ有志の手によってメッセージまで日本語化されたものも使えるようになった。2006 年 6 月 1 日に最新版 2.3.1 がリリースされ，会津大学[*4]や筑波大学[*5]など国内の CRAN ミラーサーバから入手することができる。

日本語による解説書があまりないという欠点も，2003 年 10 月に出版された拙著『R による統計解析の基礎』（ピアソン・エデュケーション）を皮切りに，間瀬ら (2004)，岡田 [編](2004) など相次いだ出版によって解消されたといえる。リファレンスマニュアル的に使える本として，舟尾 (2005)『The R Tips』という決定版が出版されたので，プロンプトからコマンドを打つスタイルで R を使われる方は，是非 1 冊『The R Tips』を入手しておき，手元に置いておくと役に立つと思う。ウェブ上の情報も，群馬大学社会情報学部の青木繁伸教授のサイト内の「R による統計処理」[*6]や岡田昌史氏による「RjpWiki」[*7]を始めとして充実したので，環境としては R を使う上で支障はなくなったといえよう。

2 R の使い方の基本

以下の解説は Windows 版による。基本的に Linux 版でも Mac OS X 版でも大差ないが，使えるデバイスなどが多少異なる。なお，本文中の\記号は¥の半角と同じものであることに注意されたい。

Rgui の起動は，デスクトップの R のアイコンをダブルクリックするだけでいい[*8]。ウィンドウが開き，作業ディレクトリの.Rprofile が実行され，保存された作業環境.RData が読まれて，

```
>
```

と表示されて入力待ちになる。この記号>をプロンプトと呼ぶ。R への対話的なコマンド入力は，基本的にプロンプトに対して行う（閉じ括弧を付け忘れたり命令や関数の途中で改行してしまった場合はプロンプトが継続行を意味する+となることに注意されたい）。また，以下の実行例の中では，プロンプトのない行が R の出力である。

入力した命令や関数は，「ファイル」メニューの「履歴の保存」で保存でき，後で「ファイル」の Source で呼び出せば再現できる。プロンプトに対して source(”プログラムファイル名”) としても同じことになる（但し，Windows ではファイルパス中，ディレクトリ（フォルダ）の区切りは/または\\で表すことに注意。できるだけ 1 つの作業ディレクトリを決めて作業することにする方が簡単である）。また，「上向き矢印キー」で既に入力したコマンドを呼び戻すことができる。

なお，R をインストールしたディレクトリの bin にパスを通しておけば，Windows 2000/XP のコマンドプロンプトで R と打っても，R を起動することができる。この場合は，コマンドプロンプトが R コンソールの代わりにシェルとして動作する。

2.1 最も基本の操作

R の終了 「ファイル」の「終了」を選ぶのでもいいし，ウィンドウ右上の×をクリックしてもいいのだが，通常は，

[*4] ftp://ftp.u-aizu.ac.jp/pub/lang/R/CRAN/

[*5] http://cran.md.tsukuba.ac.jp/

[*6] http://aoki2.si.gunma-u.ac.jp/R/

[*7] http://www.okada.jp.org/RWiki/

[*8] 前もって起動アイコンを右クリックしてプロパティを選択し，「作業フォルダ (S)」に作業ディレクトリを指定しておくとよい。環境変数 R_USER も同じ作業ディレクトリに指定するとよい（ただし，システムの環境変数または作業ディレクトリにテキストファイル.Renviron を置き，R_USER="c:/work"などと書いておくと，それが優先される）。また，企業ユーザなどで proxy を通さないと外部のネットワークと接続できない場合は，Windows のインターネットの設定できちんと proxy を設定した上で，起動アイコンのプロパティで，「起動コマンドのリンク先」末尾に--internet2 と付しておく。

```
> q()
```

とする。

付値 R でプロンプトに対して入力する内容はオブジェクトであり，「付値」という操作で，シンボルにオブジェクトを関連づけることができる。シンボル，即ち変数名は，だいたい自由に付けられ，宣言，型定義も不要である。コメントをつけることができるので，簡単な変数名にしておいて，その説明をコメントでつける方が便利な場合が多い[*9]。大文字と小文字は区別されるので，`X` と `x` は別物である。また，文字列型のオブジェクトを定義するには，半角のダブルクォート（"）またはシングルクォート（’）で括らねばならないことにも注意されたい。また，付値の際には，規定の関数名でさえもオーバーライドできてしまうものが多いが，NA（欠損値を示すシンボル）には付値できない。付値は，計算結果を任意の変数に保存することになるので「代入」と似ているが，`comment(x)<-"test"`のような関数への操作を考えると代入とは異なることがわかるだろう。関数定義の中での変数はローカルなスコープをもつので，通常の付値は関数外には影響しないが，関数外にも影響するようなグローバルな付値をすることもできる。

```
> x <- 7
> x
[1] 7
> comment(x) <- "テストデータ"
> x
[1] 7
> comment(x)
[1] "テストデータ"
> 6 -> x
> x
[1] 6
> comment(x)
NULL
> names(x) <- "テストラベル"
> x
テストラベル
  6
> x[1] <- 4
> x
テストラベル
  4
> z <<- 5 # これがグローバルな付値
```

関数ヘルプ R では html による充実したヘルプページを（インストールオプションによっては pdf のマニュアルも）参照できる（「ヘルプ」の「Html ヘルプ」など）ばかりでなく，いつでも関数ごとのヘルプを見ることができる。例として，t 検定を行う関数 `t.test()` のヘルプの見方は以下の通りである。

```
> help(t.test); # ?t.test でも同じこと
```

関数名を検索 関数の機能がわかっていて関数名がわからない場合は，ヘルプにその機能の名称一部が含まれる可能性があるので，わかる機能の特徴的な語を使って，それが含まれる関数名を検索することができる。例えば，フィッシャーを検索するには以下のようにすると，インストールされているすべてのライブラリの中から”Fisher”を含む関数がリストされたウィンドウが表示される（標準インストールでは `fisher.test(stats)` が表示されるので，Fihser の正

[*9] ただし，シンボルに付値しなおすとコメントも消えてしまうので注意。また，名前つき数値という変数の型もあり，説明は名前としてつけることもできるが，名前であっても付値し直すと消えてしまう。ただしこの場合は要素への付値であることを明示すれば名前は消えない。

確な検定を実行する関数名がそれだとわかる)。

```
> help.search("Fisher")
```

例示 R の関数の多くは，サンプル実行コードとともに提供されているので，用例を見ることができる。例えば，棒グラフを描く関数の使用例を表示するには以下のようにする。

```
> example(barplot)
```

ベクトル生成 R にもっとも簡単にベクトル型のオブジェクトを定義する方法は，関数 c() を使うことである。また整数の連番からなるベクトルは:で最初と最後の数字をつなげば定義できる。

```
> x <- c(2,7,11:19)
> x
[1] 2 7 11 12 13 14 15 16 17 18 19
```

関数定義 R では function() として任意の関数を定義することができる。定義の最後の値が，その関数の値となるので，複数の値をもたせたいときは最終行を list() にする。

```
> x <- 2
> z <- function(a) { x <<- x+a }
> print(z(5))
[1] 7
> x
[1] 7
> meansd <- function(X) {list(mean=mean(X),sd=sd(X))}
> meansd(c(1,5,8))
$mean
[1] 4.666667

$sd
[1] 3.511885
```

ライブラリ導入 CRAN に登録されているライブラリをダウンロードしてインストールするには，CRAN ミラーサイトの規定値を決めておくと便利である。筑波大ミラーを規定値にするには，

```
> options(repos="http://cran.md.tsukuba.ac.jp/")
```

とすればよいのだが，R を起動するたびに入力しなおすのは面倒なので，作業ディレクトリに.Rprofile という名前のテキストファイルを作り，この内容を書いておけばよい。あとは install.packages("**ライブラリ名**") で自動的にダウンロードとインストールができる。例として，カテゴリカルデータ解析のライブラリ vcd をインストールするには，インターネットに接続された環境で以下のようにする（dep=T は，dependency が TRUE という意味で，そのライブラリが依存するライブラリで未インストールのものがあれば，それも自動的にダウンロードしてインストールしてくれるオプションである)。

```
> install.packages("vcd",dep=T)
```

2.2 変数の型とその確認

R の変数の型には，factor（因子），character（文字列），ordered（順序），numeric（実数），integer（整数），logical（論理）など，構造をもたない型と，data.frame（データフレーム），list（リスト），ts（時系列型），matrix（行列），vector（ベクトル）など，構造をもつ型がある。ファイルから読み込んだデータは，通常，data.frame 型になる。なお，numeric は常に double（倍精度）である。single（単精度）も宣言できるが，内部的にすべての実数は倍精度扱いなので，C や Fortran の外部プログラムを取り込んでデータの受け渡しをする場合を除き意味がない。complex（複素数）という型もあるが，通常のデータ解析では使わないだろう。

強制的に因子型にする	`dat$C <- as.factor(dat$C)`
強制的に文字列型にする	`dat$S <- as.character(dat$S)`
強制的に順序型にする	`dat$I <- as.ordered(dat$I)`
強制的に実数型にする	`dat$X <- as.numeric(dat$X)`
型（構造があれば構造も）の表示	`str(dat)`
データフレーム内変数名一覧	`names(dat)`

3 データ入力の方法

3.1 データが少ないとき

データ数がごく少ない場合は，R のプログラム内で直接付値すればいい。例えば，身長 155 cm，160 cm，170 cm の 3 人のデータを入力するには，`c(155,160,170)` を用いる。`seq()` は等間隔の数値列を与え，`rep()` は同じ数値の繰り返しを与える。そこで，`mean(dat)` とすれば平均値が得られるし，`sd(dat)` とすれば不偏標準偏差が得られる。身長と体重など，2 つの属性値がある場合は，データフレームとして扱うと間違いが起こりにくい。155 cm，50 kg の A さん，160 cm，55 kg の B さん，170 cm，70kg の C さんがいる場合，`data.frame(ht=c(155,160,170),wt=c(50,55,70))` とすればデータフレームができる。データフレームの中の変数は`$`で変数名をつけて参照するか，データフレームを `attach()` しておいて変数名だけで参照する。`attach()` した場合，使い終わったら `detach()` するのを癖にした方がよい。

```
> dat <- c(155,160,170)
> dat2 <- seq(155,170,by=5)
> dat3 <- rep(160,3)
> mean(dat)
[1] 161.6667
> sd(dat)
[1] 7.637626
> dat4 <- data.frame(ht=c(155,160,170),wt=c(50,55,70))
> dat4$ht
[1] 155 160 170
> attach(dat4)
> wt
[1] 50 55 70
> detach(dat4)
```

クロス集計表を入力したい場合，行列として入力することは簡単である。例えば，次の表のようなデータがある場合を考えよう。

曝露の有無	疾病あり	疾病なし
曝露あり	20	10
曝露なし	12	18

これを dat という変数に付値するには，

```
> dat <-  matrix(c(20,12,10,18),nc=2)
> rownames(dat) <- c('曝露あり','曝露なし')
> colnames(dat) <- c('疾病あり','疾病なし')
```

とすればよい（計算するだけなら 2 行目と 3 行目は不要だが，行列ラベルをつけておく方が結果が見やすい）。matrix() は行列を定義する関数で，第 1 要素のベクトルを第 2 要素に従って並べてくれる。nc=2 は列数が 2 であることを意味し，この場合，第 1 要素のベクトルは，左上，左下，右上，右下の順に読まれる。その後，曝露と疾病が独立であるかどうかをカイ二乗検定するには，chisq.test(dat) とするだけでよい。

3.2 ある程度大きなデータを単発で入れる場合

ある程度大きなデータを入力するときは，プログラムに直接書くのは見通しが悪くなるので，データとプログラムは分離するのが普通である。同じ調査を繰り返しするとか，きわめて大きなデータであるとかでなければ，表計算ソフトで入力するのが手軽であろう[*10]。きわめて単純な例として，10 人の対象者についての身長と体重のデータが次の表のように得られているとする。

対象者 ID	身長 (cm)	体重 (kg)
1	170	70
2	172	80
3	166	72
4	170	75
5	174	55
6	199	92
7	168	80
8	183	78
9	177	87
10	185	100

この表は，表計算ソフト（Microsoft Excel や OpenOffice.org[*11]の calc など）で入力するとよい。一番上の行には変数名を入れる。日本語対応版なら漢字やカタカナ，ひらがなも使えるが，半角英数字（半角ピリオドも使える）にしておくのが無難である。ここでは，PID, HT, WT としよう（前述の通り大文字と小文字は区別されるので注意）。入力が終わったら，一旦，そのソフトの標準の形式で保存しておく（ハングアップしても困らないように）。

次に，この表をタブ区切りテキスト形式で保存する。Microsoft Excel の場合，メニューバーの「ファイル (F)」から「名前を付けて保存」を選び，現れるウィンドウの一番下の「ファイルの種類 (T)」のプルダウンメニューから「テキスト（タブ区切り）(*.txt)」を選ぶと，自動的にその上の行のファイル名の拡張子も xls から txt に変わるので，「保存 (S)」ボタンを押せば OK である。複数のシートを含むブックの保存をサポートした形式でないとかいった警告がでてくるが無視して「はい」を選んでよい。その直後に Excel を終了しようとすると，何も変更していないのに「保存しますか」と聞く警告ウィンドウがでるが，既に保存してあるので「いいえ」と答えていい（「はい」を選んでも同じ内容が上書きされるだけだが）。

あとは R で読み込めばいい。この例のように，複数の変数を含む変数名付きのデータを読み込むときは，データフレームに付値する。保存済みのデータが R の作業ディレクトリの sample.txt だとすれば，R のプロンプトに対して，dat <- read.delim("sample.txt") と打てば，データが dat というデータフレームに付値される[*12]。

例えば，このデータで身長と体重のピアソンの積率相関係数を算出し，母相関がゼロという帰無仮説で検定したいときは次のようにすればよい。

[*10] R にもデータエディタが付属していて de() で起動できるが，やや使いにくいので，通常の表計算ソフトを使う方が便利である。

[*11] http://ja.openoffice.org/を参照されたいが，Microsoft Office と互換性の高い，フリーなオフィスソフトである。

[*12] ただし，Windows 環境では，タブ区切りテキストファイルを作らずに，Excel や calc の上で範囲選択してコピーしておき，R で dat <- read.delim("clipboard") としても良い。

```
> attach(dat)
> cor.test(HT,WT)
> detach(dat)
```

3.3 欠損値について

ここで注意しなければならないのは，欠損値の取扱いである。一般に，統計処理をする対象のデータは，母集団から標本抽出したサンプルについてのものである。サンプルデータを統計解析して，母集団についての情報を得るためには，そのサンプルが正しく母集団を代表していることが何より大切である。質問紙調査の場合でも，実験研究の場合でも，欠損値（質問紙なら無回答，非該当，わからない，等，実験研究なら検出限界以下，サンプル量不足，測定失敗等）をどのように扱うかによって，サンプルの代表性が歪められてしまうことがある。欠損が少なければあまり気にしなくていいが，たとえば，健診の際の食生活質問等で，「甘いものが好きですか」に対して無回答の人は，好きだけれどもそれが健康に悪いと判断されるだろうから答えたくない可能性があり，その人たちを分析から除くと，甘いもの好きの人の割合が，全体よりも少なめに偏った対象の分析になってしまう。なるべく欠損が少なくなるような努力をすべきだけれども，どうしても欠損のままに残ってしまった場合は，結果を解釈する際に注意する。

欠損値のコードは，通常，無回答 (NA) と非該当と不十分な回答が区別できる形でコーディングするが，ソフトウェアの上で欠損値を欠損値として認識させるためのコードは，分析に使うソフトウェアによって異なっているので（欠損値を表すコードの方を変更することも可能），それに合わせておくのも 1 つの方法である。デフォルトの欠損値記号は，R なら `NA` である。Excel ではブランク（何も入力しない）にしておくと欠損値として扱われる，入力段階で欠損値をブランクにしておくと，「入力し忘れたのか欠損値なのかが区別できない」という問題を生じるので，入力段階では決まった記号を入力しておいた方が良い。

次に問題になるのが，欠損値を含むデータをどう扱うかである。結果を解釈する上で一番紛れのない方法は，「1 つでも無回答項目があったケースは分析対象から外す」ということである[*13]。その場合，統計ソフトに渡す前の段階で，そのケースのデータ全体（Excel 上の 1 行）を削除してしまうのが簡単である（もちろん，元データは別名で保存しておいて，コピー上で行削除）。もちろん R 上で操作することもできる。`dat` に欠損値を含むデータフレームが付値されているとして，欠損値が 1 つでも含まれているケースは除いたデータフレーム `datc` を作るには，次のようにする。

```
> datc <- subset(dat,complete.cases(dat),drop=T)
```

質問紙調査の場合，たとえば 100 人を調査対象としてサンプリングして，調査できた人がそのうち 80 人で，無回答項目があった人が 5 人いたとすると，回収率 (recovery rate) は 80%（80/100) となり，有効回収率 (effective recovery rate) が 75%（75/100）となる。調査の信頼性を示す上で，これらの情報を明記することは重要である。目安としては 80%程度は欲しい。

3.4 大量のデータあるいは継続的に何度も繰り返してとるデータの場合

Microsoft Access，Oracle，あるいは postgreSQL などのデータベースソフトを使い，入力用に設計したフォームから入力するのが一般的である。あるいは，データベースソフトはバックエンドにして，PHP4 や Apache httpd などを組み合わせ，ウェブアプリとするのが流行である[*14]。もっといえば，Tcl/Tk とデータベースアクセス用のライブラリを組み合わせ，R で直接入力システムを作ることも不可能ではない。「入門」の範囲を超えるので，ここでは説明しない。なお，`RODBC` ライブラリを使えば，R から直接 Dbase や Oracle，postgreSQL のデータベースを呼び出すことができる。

[*13] 非該当は外さない。

[*14] 作成途中だが `http://phi.med.gunma-u.ac.jp/swtips/webdb.html` が参考になるかもしれない。

4 R における層別の扱い方

4.1 データの例

対象者 ID	身長 (cm)	体重 (kg)	性別	年齢
1	170	70	M	54
2	162	50	F	34
3	166	72	M	62
4	170	75	M	41
5	164	55	F	37
6	159	62	F	55
7	168	80	F	67
8	183	78	M	47
9	157	47	F	49
10	185	100	M	45

10 人の対象者についての身長と体重のデータが上表のように得られているとする。この程度のデータ入力は通常なら表計算ソフトで行い，1 行目の変数名を PID, HT, WT, SEX, AGE とし，タブ区切りテキスト形式で，R の作業ディレクトリに sample.txt というファイル名で保存しておいて，

```
> dat <- read.delim("sample.txt")
```

と打って，データを dat というデータフレームに付値すればよい。あるいは，表計算ソフトは使わず，直接次のようにしてもよい。c() でデータを表の通りにコンマで区切って入力していくのだが，括弧を閉じない限りデータは継続しているとみなされる。matrix として定義するときに重要なのは，nc=5 でカラム数，つまり変数数が 5 であると示すことと，読む順序が 1 行ずつであることを byrow=T で示すことである。あとはその行列を as.data.frame() でデータフレームに変え，colnames() で変数名をつければいい。ただし，1 つでも文字が入っていると全体が Factor 扱いされてしまうのと，'M' などとクォーテーション付きで入力するのは煩雑なので，因子型のデータも数値で入力しておき，後で因子型に変換した方がいい。タブ区切りテキストファイルから read.delim() で入力すればその種の問題はない。

```
> dat <- as.data.frame(matrix(c(
+ 1, 170, 70, 1, 54,
+ 2, 162, 50, 2, 34,
+ 3, 166, 72, 1, 62,
+ 4, 170, 75, 1, 41,
+ 5, 164, 55, 2, 37,
+ 6, 159, 62, 2, 55,
+ 7, 168, 80, 2, 67,
+ 8, 183, 78, 1, 47,
+ 9, 157, 47, 2, 49,
+ 10, 185, 100, 1, 45),nc=5,byrow=T))
+ colnames(dat) <- c('PID','HT','WT','SEX','AGE')
> dat$SEX <- as.factor(dat$SEX)
> levels(dat$SEX) <- c('M','F')
```

また別の方法としては，変数ごとに縦に入力しておいて，それを data.frame() でまとめる手もある。ただ，それだとカラムが揃わないので入力ミスを見つけにくいという欠点があり，あまりお薦めしない。

4.2 データフレームの一部だけの解析をする

R では，変数名の後に [ ] で条件設定をすることで，変数の一部だけを分析することが可能である。例えば男性だけの身長の平均値を計算したければ，

```
> attach(dat)
> mean(HT[SEX=='M'])
[1] 174.8
> detach(dat)
```

とすればよい。しかし，同じ条件でたくさんの変数の一部だけについて複数の統計量を計算させたいとき，いちいち [dat$SEX=='M'] とつけるのは面倒だろう。そういう場合は関数定義すると便利である。

```
> cNms <- function(X,C) { list(N=NROW(X[C]),mean=mean(X[C]),sd=sd(X[C])) }
```

は，人数 N と平均 mean と不偏標準偏差 sd を返す関数を cNms() という名前で定義している。すると次から，

```
> attach(dat)
> cNms(HT,SEX=='M')
$N
[1] 5

$mean
[1] 174.8

$sd
[1] 8.58487
> detach(dat)
```

のようなことができる。条件設定は一致することを意味する==だけでなく，不等号も使えるし，is.na() などの関数も使えるので，40 歳以上だけについて身長の平均値を計算させるといったことも可能である。条件設定は，要するに論理型変数のベクトルを作っているだけなので，一時的に途中でシンボル名（変数）に付値することもできる。論理型変数の否定は “!” という記号でできる。

```
> attach(dat)
> overforty <- AGE>=40
> cNms(HT,overforty)
$N
[1] 8

$mean
[1] 169.75

$sd
[1] 10.02497
> detach(dat)
```

といったことも可能である。&（かつ）や|（または）を使ってこれらの条件を組み合わせることもできるので，40 歳以上

男性の人数と体重の平均と標準偏差を求めるには,

```
> attach(dat)
> males <- SEX=='M'
> cNms(WT,overforty&males)
$N
[1] 5

$mean
[1] 79

$sd
[1] 12.12436
> detach(dat)
```

とすればよい。なお，[ ] の中に数字を入れると，その順番のオブジェクトを参照することもできる。

4.3 層別の分析をする

R には，実は分類変数によって層別に任意の関数を適用する関数 tapply() が用意されている。例えば次のようなことができる。

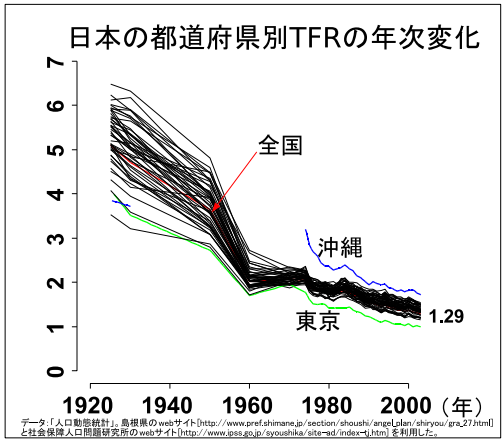
```
> attach(dat)
> tapply(HT,SEX,mean)
    M     F
174.8 162.0
> detach(dat)
```

5 図示の基本

データの図示の目的は大別して２つある。１つは見せるためであり，もう１つは考えるためである。もちろん，両者の機能を併せもつグラフも存在するが，重視すべきポイントが変わってくるので，一般には，この２つは別のグラフになる。

見せるためのグラフでも，プレゼンテーションやポスターに使うグラフと，投稿論文に載せるグラフは，一般に別物である。前者は，１つのグラフに１つのことだけを語らせる必要があり，とにかくわかりやすさが最大のポイントであるのに対して，後者は複数の内容を語らせることも可能である。これは，見る人が１枚のグラフを見るために使える時間からくる制約である。例えば，日本の都道府県別 TFR（合計出生率）の年次変化のグラフを示すのに，プレゼンテーションならば左図のようにした方が見やすいが，論文に載せる場合は右図のようにする方が良い。いずれの場合も統計ソフトだけで仕上げるのは（不可能ではないが）面倒だし，管理上も不都合なので，プレゼンテーションソフト[*15]か描画ソフト[*16]に貼り付けて仕上げるのが普通である。

[*15] PowerPoint のほかには，フリーソフトの OpenOffice.org に含まれている Impress というものが有名であり，概ね PowerPoint と互換である。
[*16] OpenOffice.org に含まれている Draw も使える。もし使える環境にあれば，Adobe の Illustrator がよい（高価だが）。



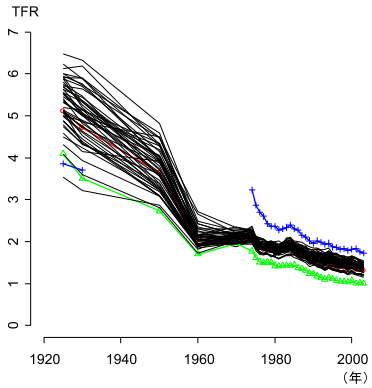


図2. 日本の都道府県別合計出生率(TFR)の年次変化
(△:東京, +:沖縄, マークなし:他道府県, ○:全国).
データは「人口動態統計」に基づく.

見せるためのグラフについて詳しく知りたい方は，山本義郎 (2005)『レポート・プレゼンに強くなるグラフの表現術』講談社現代新書 (ISBN4-06-149773-1) を一読されることをお勧めする。時間的な制約もあるので，この演習では考えるためのグラフに絞って説明する。考えるためのグラフに必要なのは，データの性質に忠実に作るということである。データの大局的性質を把握するために，ともかくたくさんのグラフを作って多角的に眺めてみよう。人間の視覚的認識能力は，パターン認識に関してはコンピュータより遥かに優れていると言われているから，それを生かさない手はない。統計解析は，いろいろな仮定をおいて理論構築されているので，ただソフトウェアの計算結果の数値だけを妄信してしまうのは危険である。図示されたものをみれば，直感的なチェックができるので，仮定を満たしていない統計手法を使ってしまう危険が避けられる場合が多い。つまり，

統計解析前に図示は必須

である。たくさんの図を作ったときは，ある程度まとめて管理できた方が便利だし，コメントもつけておく方が再利用するときに役に立つと思われるので，作った図はメタファイルとしてプレゼンテーションソフトまたは描画ソフトに貼り付けておくことをお勧めする。

では，具体的な図示の方法に入ろう。変数が表す尺度の種類によって，さまざまな図示の方法があるので，それをざっと示すことにする。

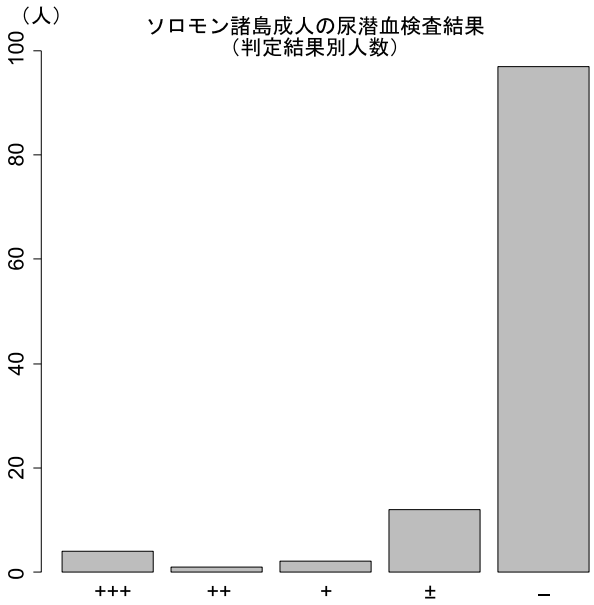
5.1 名義尺度または順序尺度をもつ変数の場合

因子型または順序型の変数についての作図は，カテゴリごとの度数を情報として使うことになる。そのため，作図関数に渡す値は一般にデータそのものではなく，その集計結果になる[*17]。もちろん，既に表の形になっている場合は，そのまま作図関数に渡すことができる。

度数分布図 値ごとの頻度を縦棒として，異なる値ごとに，この縦棒を横に並べた図である。離散変数の名前を X とすれば，R では `barplot(table(X))` で描画される。例えば，ソロモン諸島の M 村の成人について尿検査をして，潜血の結果が，＋＋＋ が 4 人，++ が 1 人，＋ が 2 人，± が 12 人，− が 97 人だったとしよう。これを度数分布図として棒グラフを作成するには，どうしたらいいだろうか。この例では，`table(X)` に当たる部分が既に与えられているので，下枠内のように，まずカテゴリ別度数を `c()` で与え，`names()` を使ってカテゴリに名前を付けてから，`barplot()` 関数で棒グラフを描画すればよい（以下の例では OpenOffice.org の Draw を使って加工済みのグラフを載せておく）。

[*17] `table()` 関数を使って度数分布を求め，その結果を作図関数に与えるのが普通である。

```
> ob <- c(4,1,2,12,97)
> names(ob) <- c("+++","++","+","±","-")
> barplot(ob,ylim=c(0,100),main="ソロモン諸島成人の尿潜血検査結果\n（判定結果別人数）")
```



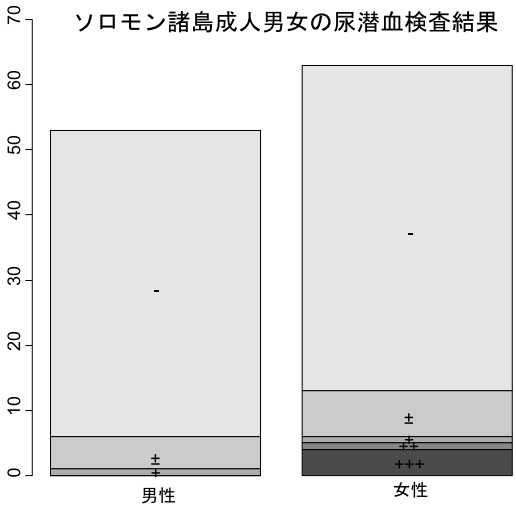
なお，合計で割って縦軸を割合にした方が見やすい場合もある。

積み上げ棒グラフ　値ごとの頻度の縦棒を積み上げた図である。上のデータで積み上げ棒グラフを描くには以下のようにする。最初の 2 行は変わらない。残りは，barplot() の引数を変えることと，カテゴリ名を適切な位置に書き込むために必要な部分である。

```
> ob <- c(4,1,2,12,97)
> names(ob) <- c("+++","++","+","±","-")
> ii <- barplot(matrix(ob,NROW(ob)),beside=F,ylim=c(0,120),
+ main="ソロモン諸島成人の尿潜血検査結果")
> oc <- ob
> for (i in 1:length(ob)) { oc[i] <- sum(ob[1:i])-ob[i]/2 }
> text(ii,oc,paste(names(ob)))
```

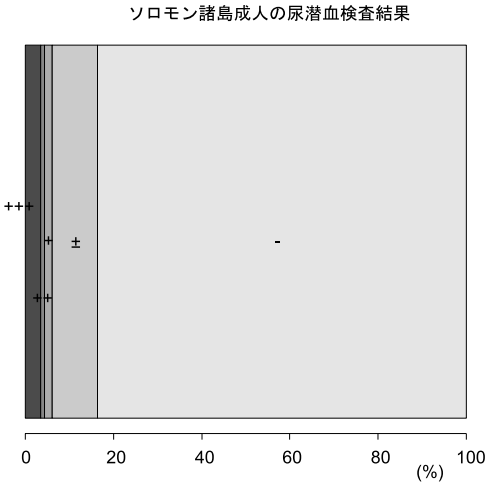
積み上げ棒グラフは単独で用いるよりも，複数の積み上げ棒グラフを並べて比較するのに向いている。例えば，上の結果を男女別に見ると，男性では＋＋＋と ++ が 0 人，＋が 1 人，± が 5 人，− が 47 人，女性では＋＋＋が 4 人，++ が 1 人，＋が 1 人，± が 7 人，− が 50 人だったとき，男女別々に積み上げ棒グラフを描いて並べると，内訳を男女で比較することができる。実行するための R のコードは以下の通りである。

```
> obm <- c(0,0,1,5,47)
> obf <- c(4,1,1,7,50)
> obx <- cbind(obm,obf)
> rownames(obx) <- c("+++","++","+","±","-")
> colnames(obx) <- c("男性","女性")
> ii <- barplot(obx,beside=F,ylim=c(0,70),main="ソロモン諸島成人男女の尿潜血検査結果")
> oc <- obx
> for (i in 1:length(obx[,1])) { oc[i,1] <- sum(obx[1:i,1])-obx[i,1]/2 }
> for (i in 1:length(obx[,2])) { oc[i,2] <- sum(obx[1:i,2])-obx[i,2]/2 }
> text(ii[1],oc[,1],paste(rownames(obx)))
> text(ii[2],oc[,2],paste(rownames(obx)))
```



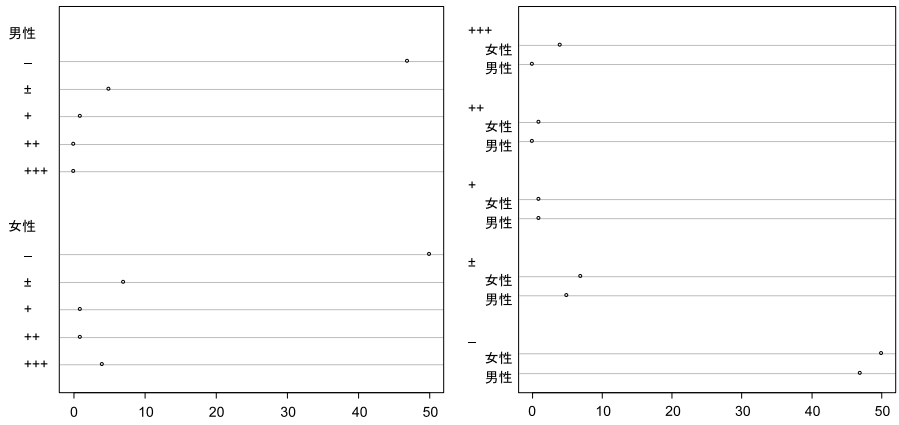
帯グラフ　横棒を全体を 100%として各カテゴリの割合にしたがって区切って塗り分けた図である。内訳を見るのに向いている。複数並べて構成比を比べたいときに効果を発揮する。ソロモン諸島成人の尿潜血検査結果について帯グラフを描くための R のコードは下枠内の通り。2 行目で人数を構成割合 (%) に変える計算をしているのと，4 行目の `horiz=T` が重要である。

```
> ob <- c(4,1,2,12,97)
> obp <- ob/sum(ob)*100
> names(obp) <- c("+++","++","+","±","-")
> ii <- barplot(matrix(obp,NROW(obp)),horiz=T,beside=F,xlim=c(0,100),
+  xlab="(%)",main="ソロモン諸島成人の尿潜血検査結果")
> oc <- obp
> for (i in 1:length(obp)) { oc[i] <- sum(obp[1:i])-obp[i]/2 }
> text(oc,ii,paste(names(obp)))
```



ドットチャート　棒グラフの棒を描く代わりに上端に点を打ったグラフである。複数のドットチャートを並列することもできる。基本的に `barplot()` の代わりに `dotchart()` を使えばよい。ソロモン諸島成人男女の尿潜血の例では，下枠内のコードを用いればよい。

```
> obm <- c(0,0,1,5,47)
> obf <- c(4,1,1,7,50)
> obx <- cbind(obm,obf)
> rownames(obx) <- c("+++","++","+","±","-")
> colnames(obx) <- c("男性","女性")
> dotchart(obx)
> dotchart(t(obx))
```



円グラフ　円全体を 100%として，各カテゴリの割合にしたがって中心から区切り線を引き，塗り分けた図である。構成比を見るには，帯グラフよりも直感的にわかりやすい場合も多い[*18]。ドーナツグラフでは 2 つの同心円にして，内側の円内を空白にする。R では `pie()` 関数を用いる[*19]。ソロモン諸島成人の尿潜血検査結果について円グラフを描かせる R のコードは下枠内の通りである。

```
> ob <- c(4,1,2,12,97)
> names(ob) <- c("+++","++","+","±","-")
> pie(ob)
```

5.2　連続変数の場合

ヒストグラム　変数値を適当に区切って度数分布を求め，分布の様子を見るものである。Excel ではツールのアドインの分析ツールに含まれているヒストグラム作成機能で区切りも与えてやらないと作成できず，非常に面倒だが，R では `hist()` 関数にデータベクトルを与えるだけである，棒グラフとの違いは，横軸（人口ピラミッド[*20]のように 90 度回転して縦軸になることもある）が，連続していることである（区切りに隙間があってはいけない）。基本的に，区切りにはアプリオリな意味はないので，分布の形を見やすくするとか，10 進法で切りのいい数字にするとかでよい。対数軸にする場合も同様である。さっきのデータで身長のヒストグラムを描かせるコードは下枠内の通りである。

[*18] ただし，認知心理学者の Cleveland は，その著書【Cleveland WS (1985) The elements of graphing data. Wadsworth, Monterey, CA, USA.】の p.264 で，「円グラフで示すことができるデータは，常にドットチャートでも示すことができる。このことは，共通した軸上の位置の判定が，正確度の低い角度の判定の代わりに使えることを意味する。」と，実験研究の結果から述べているし，R の help ファイルは，構成比を示すためにも円グラフよりもドットチャートや帯グラフあるいは積み上げ棒グラフを使うことを薦めているので，円グラフはむしろ「見せるためのグラフ」として使う際に価値が高いといえよう。

[*19] R-1.5 以前は `piechart()` 関数だったが置き換えられた

[*20] 詳しくは `http://phi.med.gunma-u.ac.jp/demography/makepyramid.html` を参照。

```
> attach(dat)
> hist(HT,main="身長のヒストグラム")
> detach(dat)
```

正規確率プロット　連続変数が正規分布しているかどうかを見るグラフである。正規分布に当てはまっていれば点が直線上に並ぶし，ずれていればどのようにずれているかを読み取ることができる。ヒストグラムに比べると，正規確率プロットから分布の様子を把握するには熟練を要するが，区切りの恣意性によって分布の様子が違って見える危険がないので，ヒストグラムと両方実施すべきである。ヒストグラムで示したのと同じデータについて正規確率プロットを描かせるには，下枠内を打てばよい。qqnorm() で正規確率プロットが描かれ，qqline() で，もしデータが正規分布していればこの直線上に点がプロットされるはず，という直線が描かれる。

```
> qqnorm(dat$HT,main="身長の正規確率プロット",ylab="身長 (cm)")
> qqline(dat$HT,lty=2)
```

幹葉表示　英語では stem and leaf plot と呼ぶ。大体の概数（整数区切りとか 5 の倍数とか 10 の倍数にすることが多い）を縦に並べて幹とし，それぞれの概数に相当する値の細かい部分を葉として横に並べて作成する図。R では stem() 関数を用いる。ただしテキスト出力画面に出力されるため，グラフィックとして扱うには少々工夫が必要である。ヒストグラムを 90 度回転して数字で作るようなものだが，各階級の内訳がわかる利点がある。身長の例では，下枠内のように打てばテキスト画面に幹葉表示が得られる[*21]。

```
> stem(dat$HT)
```

箱ヒゲ図　英語では box and whisker plot，または boxplot と呼ばれる。データを小さい方から順番に並べて，ちょうど真中にくる値を中央値 (median) といい，小さい方から 1/4 の位置の値を第 1 四分位 (first quartile)，大きいほうから 1/4 の位置の値を第 3 四分位 (third quartile) という。縦軸に変数値をとって，第 1 四分位を下に，第 3 四分位を上にした箱を書き，中央値の位置にも線を引いて，さらに第 1 四分位と第 3 四分位の差（四分位範囲）を 1.5 倍した線分をヒゲとして第 1 四分位の下と第 3 四分位の上に伸ばし，ヒゲの先より外れた値を外れ値として○をプロットした図である。カテゴリによって層別した箱ヒゲ図を横に並べて描くと，大体の分布の様子と外れ値の様子が同時に比較できるので便利である。R では boxplot() 関数を用いる。身長データだと下枠内を打てばよい。

```
> boxplot(dat$HT)
```

ストリップチャート　2 群間で平均値を比較する場合などに，群ごとに大まかに縦軸での位置を決め，横軸には各データ点の正確な値をプロットした図（群の数によって縦軸と横軸は入れ換えた方が見やすいこともある）。R では stripchart() 関数を用いる（縦軸と横軸を入れ換えるには，vert=T オプションをつける）。横に平均値と標準偏差の棒を付加することも多い。身長データを男女間で比較するためのコードの例を下枠内に示す。

```
> attach(dat)
> mHT <- tapply(HT,SEX,mean)
> sHT <- tapply(HT,SEX,sd)
> IS <- c(1,2)+0.15
> stripchart(HT~SEX,method="jitter",vert=T,ylab="身長 (cm)")
> points(IS,mHT,pch=18)
> arrows(IS,mHT-sHT,IS,mHT+sHT,code=3,angle=90,length=.1)
> detach(dat)
```

[*21] source("http://phi.med.gunma-u.ac.jp/swtips/gstem.R") としてから stem() の代わりに gstem() を使えば，図形としての出力が得られる。

散布図　2つの連続変数の関係を2次元の平面上の点として示した図である。R では plot() 関数を用いる。異なる群ごとに別々のプロットをしたい場合は plot() の pch オプションで塗り分けたり，points() 関数を使って重ね打ちしたりできる。点ごとに異なる情報を示したい場合は symbols() 関数を用いることができるし[*22]，複数の連続変数間の関係を調べるために，重ね描きしたい場合は matplot() 関数と matpoints() 関数を，別々のグラフとして並べて同時に示したい場合は pairs() 関数を用いることができる。データ点に文字列を付記したい場合は text() 関数が使えるし，マウスで選んだデータ点にだけ文字列を付記したい場合は identify() 関数が使える。もっとも基本的な使い方として，身長と体重の関係を男女別にマークを変えてプロットするなら，下枠内のようにする。

```
> plot(dat$HT,dat$WT,pch=paste(dat$SEX),xlab="身長 (cm)",ylab="体重 (kg)")
```

レーダーチャート　複数の連続変数を中心点から放射状に数直線としてとり，データ点をつないで表される図である。それら複数の変数によって特徴付けられる性質のバランスをみるのに役立つ。1つのケースについて1つのレーダーチャートができるので，他のケースと比較するには，並べて描画するか，重ね描きする。R では stars() 関数を用いるが，詳細は省略する。

5.3 その他のグラフ

以上説明した基本的なグラフ以外にも，maptools ライブラリと ESRI 社が公開しているデータを使えば，GIS のように統計情報によって地図を塗り分けることができるし[*23]，系統関係を示す樹状図（デンドログラム）は，クラスタ分析で描けるし，生存時間データについては生存関数を描けるなど，目的に応じて多様なグラフがある。

6 記述統計

記述統計とは，生データが含んでいる多くの情報を少ない数値に集約して示す方法である。つまり，分布の特徴をいくつかの数値で代表させようというわけである。このような値を，代表値と呼ぶ。たんに代表値という場合は分布の位置を指すことが多いが，ここではもう少し広い意味で用いる。

分布の特徴を代表させる値には，分布の位置を示す値と，分布の広がりを示す値がある。例えば，正規分布だったら，$N(\mu, \sigma^2)$ という形で表されるように，平均 μ，分散 σ^2 という2つの値によって分布が決まるわけだが，この場合，μ が分布の位置を決める情報で，σ^2 が分布の広がりを決める情報である。分布の位置を示す代表値は central tendency（中心傾向）と呼ばれ，分布の広がりを示す代表値は variability（ばらつき）と呼ばれる。

一般に，統計処理の対象になっているデータは，仮想的な母集団（言い換えると，その研究結果を適用可能と考えられる範囲でもある）からの標本（サンプル）であり，データから計算される代表値は，母集団での分布の位置や広がりを推定するために使われる。母集団での位置や広がりを示す値は母数（parameter）と呼ばれ，分布の位置を決める母数を位置母数 (location parameter)，分布の広がりを決める母数を尺度母数 (scale parameter) と呼ぶ。例えば不偏分散は尺度母数の一つである。

[*22] 使用例は，http://phi.med.gunma-u.ac.jp/medstat/semen.R を試されたい。
[*23] 詳細は http://phi.med.gunma-u.ac.jp/swtips/EpiMap.html を参照されたい。

6.1 1 変数記述統計のまとめ

度数分布表	`table(C)`
五数要約値［最小，Q1，中央値，Q3，最大］	`fivenum(X)`
サンプルサイズ	`NROW(X)` または `length(X)`
合計	`sum(X)`
最大値	`max(X)`
最小値	`min(X)`
平均（算術平均）	`mean(X)` （`sum(X)/length(X)` と同値）
幾何平均	`exp(mean(log(X)))` または `prod(X)^(1/NROW(X))`
調和平均	`1/(mean(1/X))`
中央値	`median(X)`
四分位範囲	`fivenum(X)[4]-fivenum(X)[2]`
四分位偏差	`(fivenum(X)[4]-fivenum(X)[2])/2`
不偏分散	`var(X)` （`sum((X-mean(X))^2)/(length(X)-1)` と同値）
不偏標準偏差	`sd(X)` （`sqrt(var(X))` と同値）

6.2 2 変数記述統計のまとめ

カテゴリ×カテゴリ＝クロス集計表	`table(C1,C2)`
量×量＝ピアソンの相関係数	`cor(X,Y)`
量×量＝スピアマンの順位相関係数	`cor(X,Y,method="spearman")`

6.3 練習のためのサンプルデータ

R にはさまざまなサンプルデータが含まれており，`try(data())` とすると一覧表示できる。その中の `ChickWeight` を使ってみることにする。これは，含まれるタンパク質が異なる 4 種類の試験食を与えて飼育した 50 羽の鶏の体重を 0 日目から 20 日目まで測定したデータである（何日分か欠損がある）。

Chick という順序型の変数に鶏の個体番号が入っており，Diet という因子型の変数に食事の種類を示す数字が入っており，Time という数値型の変数に何日目の体重かという値が入っており，weight という数値型の変数に体重が入っている。以下のようにすると，`X` という変数に，餌の種類を問わず，20 日目の鶏の体重がすべて入る。50 羽の鶏の体重データは，図示や記述統計の練習をするにはちょうどいいだろう。

```
> data(ChickWeight)
> attach(ChickWeight)
> X <- weight[Time==20]
> detach(ChickWeight)
```

7 検定

7.1 基本的な分布

R における確率分布は，分布名の前に d がつけば確率密度関数，p がつけば分布関数，q がつけば分位点関数，r がつけばその分布に従う乱数を発生させる関数となる。例えば自由度 15 の t 分布の 97.5%点を得る関数は，`qt(0.975,df=15)` となるし，平均 0，標準偏差 1 の正規分布に従う乱数を 100 個発生させる関数は，`rnorm(100,0,1)` となる。確率分布の名前

は，正規分布が norm，t 分布が t，F 分布が f，カイ二乗分布が chisq，ウィルコクソンの順位和統計量の分布が wilcox などである。有名な分布は用意されているので，名前がわからない場合は，help.search() で探してみればよい。分布の形をグラフでみるには，curve() という関数が便利である。たとえば，平均 10，分散 2 の正規分布の確率密度関数を区間 (0,20) で描くには，以下のようにする。

```
> curve(dnorm(x,10,2),0,20)
```

7.2 検定の考え方と第一種，第二種の過誤

検定とは，帰無仮説（一般には，差がない，という仮説）の元で得られた統計量を，既知の確率分布をもつ量と見た場合に，その値よりも外れた値が得られる確率（これを「有意確率」と呼ぶ）がどれほど小さいかを調べ，有意水準[*24]より小さければ，統計的に意味があることと捉え（統計的に有意である，という），帰無仮説がおかしいと判断して棄却する（つまり，「差がないとは言えない」と判断する）という意思決定を行うものである。

この意思決定が間違っていて，本当は帰無仮説が正しいのに，間違って帰無仮説を棄却してしまう確率は，有意水準と等しいので，その意味で，有意水準を第一種の過誤と（α エラーとも）呼ぶ。逆に，本当は帰無仮説が正しくないのに，その差を検出できず，有意でないと判断してしまう確率を，第二種の過誤と（β エラーとも）呼び，1 から第二種の過誤を引いた値が検出力になる。

注意しなければいけないのは，有意確率の小ささは，あくまで，帰無仮説のありえなさを示すだけであって，**差の大きさを意味するのではない**点である。

R で検定を行う関数の出力には，大抵の場合，分布関数を使って計算された有意確率が p.value として表示されている。

7.3 分布の正規性の検定

高度な統計解析をするときには，データが正規分布する母集団からのサンプルであるという仮定を置くことが多いが，それを実際に確認することは難しいので，一般には，分布の正規性の検定を行うことが多い。考案者の名前から Shapiro-Wilk の検定と呼ばれるものが代表的である。

Shapiro-Wilk の検定の原理をざっと説明すると，$Z_i = (X_i - \mu)/\sigma$ とおけば，Z_i が帰無仮説「X が正規分布にしたがう」の下で $N(0,1)$ からの標本の順序統計量となり，$c(i) = E[Z(i)], d_{ij} = Cov(Z(i), Z(j))$ が母数に無関係な定数となるので，「$X(1) < X(2) < ... < X(n)$ の $c(1), c(2), ...c(n)$ への回帰が線形である」を帰無仮説として，そのモデルの下で σ の最良線形不偏推定量 $\hat{(\sigma)} = \sum_{i=1}^{n} a_i X(i)$ と $S^2 = \sum_{i=1}^{n}(X_i - \bar{X})^2$ を用いて，$W = (k\hat{\sigma}^2)/S^2$ を検定統計量として検定するものである。k は $\sum_{i=1}^{n}(ka_i)^2 = 1$ より求められる。

数値型変数 X の分布が正規分布にフィットしているかどうかを検定するには，

```
> shapiro.test(X)
```

とすればよい。変数 X のデータ数（ベクトルの要素数，R のコードでは length(X)）は，3 から 5000 の間でなければならない。2 以下では分布を考える意味がなく，また，検定統計量 W の分布がモンテカルロシミュレーションによって得られたものであるため，あまりに大きなサンプルサイズについては値が与えられていない。

[*24] 分析者が決める一定の確率。当該研究分野の伝統に従うのが普通である。先行研究があればそれに従う。他に基準がなければ 5%か 1%にすることが多い。

7.4 1 変数の検定のまとめ

分布の正規性（シャピロ＝ウィルクの検定）	`shapiro.test(X)`
母平均の検定	`t.test(X,mu=`**母平均**`)`
母比率の検定	`binom.test(table(B)[2],length(B),p=`**母比率**`)`

7.5 平均値の差の検定における両側検定と片側検定

2 つの量的変数 X と Y の平均値の差の検定をする場合，それぞれの母平均を μ_X，μ_Y と書けば，その推定量は $\mu_X = \mathrm{mean}(X) = \sum X/n$ と $\mu_Y = \mathrm{mean}(Y) = \sum Y/n$ となる。

両側検定では，帰無仮説 H0 : $\mu_X = \mu_Y$ に対して対立仮説（帰無仮説が棄却された場合に採択される仮説）H1 : $\mu_X \neq \mu_Y$ である。H1 を書き直すと，「$\mu_X > \mu_Y$ または $\mu_X < \mu_Y$」ということである。つまり，t_0 を「平均値の差を標準誤差で割った値」として求めると，t_0 が負になる場合も正になる場合もあるので，有意水準 5%で検定して有意になる場合というのは，t_0 が負で t 分布の下側 2.5%点より小さい場合と，t_0 が正で t 分布の上側 2.5%点（つまり 97.5%点）より大きい場合の両方を含む。t 分布は原点について対称なので，結局両側検定の場合は，上述のように差の絶対値を分子にして，t_0 の t 分布の上側確率[*25]を 2 倍すれば有意確率が得られることになる。

片側検定は，先験的に X と Y の間に大小関係が仮定できる場合に行い，例えば，X の方が Y より小さくなっているかどうかを検定したい場合なら，帰無仮説 H0 : $\mu_X \geq \mu_Y$ に対して対立仮説 H1 : $\mu_X < \mu_Y$ となる。この場合は，t_0 が正になる場合だけ考えればよい。有意水準 5%で検定して有意になるのは，t_0 が t 分布の上側 5%点（つまり 95%点）より大きい場合である。なお，R で平均値の差の検定を行うための関数は `t.test()` であるが，片側検定をするときは対立仮説を `alt="greater"`とか `alt="less"`などオプションとして与えればよい。

7.6 独立 2 変数の平均値の差の検定

標本調査によって得られた独立した 2 つの量的変数 X と Y（サンプル数が各々 n_X と n_Y とする）について，平均値に差があるかどうかを検定することを考える[*26]。

母分散が既知で等しい V である場合　$z_0 = |E(X) - E(Y)|/\sqrt{V/n_X + V/n_Y}$ が標準正規分布に従うことを使って検定する[*27]。

母分散が未知の場合　調査データを分析する場合は母分散が既知であることはほとんどなく，こちらが普通である。手順は以下の通り。

1. F 検定（分散が等しいかどうか）：2 つの量的変数 `X` と `Y` の不偏分散 `SX<-var(X)` と `SY<-var(Y)` の大きい方を小さい方で（以下の説明では `SX>SY` だったとする）割った `F0<-SX/SY` が第 1 自由度 `DFX<-length(X)-1`，第 2 自由度 `DFY<-length(Y)-1` の F 分布に従うことを使って検定する。有意確率は `1-pf(F0,DFX,DFY)` で得られる。しかし，`F0` を手計算しなくても，`var.test(X,Y)` で等分散かどうかの検定が実行できる。また，1 つの量的変数 `X` と 1 つの群分け変数 `C` があって，`C` の 2 群間で `X` の分散が等しいかどうか検定するというスタイルでデータを入力してある場合は，`var.test(X~C)` とすればよい。
2. 分散に差があるか差がないかによって，平均値が等しいかどうかの検定法は異なる。分散に差があるときは，その事実をもって別の母集団からとられた標本であると判断し，平均値が等しいかどうかを検定する意味はないとする考え方もあるが，一般には Welch の方法を使うか，ノンパラメトリックな方法を使って検定する。

[*25] t 分布の確率密度関数を t_0 から無限大まで積分した値，即ち，t 分布の分布関数の t_0 のところの値を 1 から引いた値。R では `1-pt(t0,`**自由度**`)`。

[*26] 数学的な仕組みについては，1981 年に出た数学セミナー増刊の竹内啓・大橋靖雄『入門｜現代の数学 [11] 統計的推測－2 標本問題』日本評論社を参照されたい。

[*27] 分布がひどく歪んでいる場合には，Mann-Whitney の U 検定（Wilcoxon の順位和検定と数学的に同値）を行う。

分散に差がない場合　まず母分散 S を `S<-(DFX*SX+DFY*SY)/(DFX+DFY)` として推定する。

`t0<-abs(mean(X)-mean(Y))/sqrt(S/length(X)+S/length(Y))` が自由度 `DFX+DFY` の t 分布に従うことから，帰無仮説「X と Y の平均値には差がない」を検定すると，`(1-pt(t0,DFX+DFY))*2` が有意確率となる。

R では，`t.test(X,Y,var.equal=T)` とする。また，F 検定のところで触れた量的変数と群分け変数という入力の仕方の場合は，`t.test(X~C,var.equal=T)` とする。ただしこれだと両側検定なので，片側検定したい場合は，`t.test(X,Y,var.equal=T,alternative="less")` などとする（`alternative="less"`は対立仮説が `X<Y` という意味なので，帰無仮説が `X>=Y` であることを意味する)。

分散に差がある場合　Welch の方法を用いる。

$t_0 = |E(X) - E(Y)|/\sqrt{S_X/n_X + S_Y/n_Y}$ が自由度 ϕ の t 分布に従うことを使って検定する。但し，ϕ は下式による。

$$\phi = \frac{(S_X/n_X + S_Y/n_Y)^2}{\{(S_X/n_X)^2/(n_X-1) + (S_Y/n_Y)^2/(n_Y-1)\}}$$

R では，`t.test(X,Y,var.equal=F)` だが，`var.equal` の指定を省略した時は等分散でないと仮定して Welch の検定がなされるので省略して `t.test(X,Y)` でいい。量的変数と群分け変数という入力の仕方の場合は，`t.test(X~C)` とする。

Fisher のアヤメのデータで setosa 種と virginica 種の間で Sepal.Length（萼片の長さ）に差があるかを t 検定したい場合は，以下のようにする（ただし Fisher のアヤメのデータは versicolor も含め 3 種のデータがあるので，本来は 3 群間の比較をするデザインであることに注意)。setosa 種の平均値が 5.006，virginica 種の平均値が 6.588 で，p-value が 2.2×10^{-16} ということは有意確率がほとんどゼロなので「差はない」という帰無仮説は棄却される。

```
> data(iris)
> setosa.sl <- iris$Sepal.Length[iris$Species=="setosa"]
> virginica.sl <- iris$Sepal.Length[iris$Species=="virginica"]
> vareq <- var.test(setosa.sl,virginica.sl)$p.value >= 0.05
> t.test(setosa.sl,virginica.sl,var.equal=vareq)
        Welch Two Sample t-test

data:  setosa.sl and virginica.sl
t = -15.3862, df = 76.516, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.786760 -1.377240
sample estimates:
mean of x mean of y
    5.006     6.588
```

7.7　対応のある 2 標本の平均値の差の検定

アヤメの花は，種によって花弁の大きさと萼片の大きさの関係が違っている。setosa 種は萼片の方が花弁より圧倒的に大きいが，virginica 種ではそれほどの違いはないように見える。virginica 種 50 個体について Sepal.Length と Petal.Length（花弁の長さ）に差があるかどうかを検定したい場合，全体の平均に差があるかないかだけをみるのではなく，個体ごとの違いを見るほうが情報量が失われない。このような場合は，独立 2 標本の平均値の差の検定をするよりも，対応のある 2 標本として分析する方が切れ味がよい（差の検出力が高い)。対応のある 2 標本の差の検定は，paired-t 検定と呼ばれ，意味合いとしてはペア間の値の差を計算して値の差の母平均が 0 であるかどうかを調べることになる。R で対応のある変数 X と Y の paired-t 検定をするには，`t.test(X,Y,paired=T)` で実行できるし，それは `t.test(X-Y,mu=0)` と等価である。な

お，分布が歪んでいる場合や，分布が仮定できない場合の対応のある 2 標本の分布の位置の差があるかどうか検定するには，ウィルコクソンの符号順位検定を用いる。

以上の分析をするための R のコードは以下の通りである。対応のある t 検定でもウィルコクソンの符号順位検定でも p-value はほとんどゼロなので，virginica 種の萼片と花弁の長さに差がないという帰無仮説は棄却される。

```
> data(iris)
> virginica <- subset(iris,Species=="virginica",drop=T)
> attach(virginica)
> t.test(Sepal.Length,Petal.Length,paired=T)
        Paired t-test

data:  Sepal.Length and Petal.Length
t = 22.8981, df = 49, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.9450788 1.1269212
sample estimates:
mean of the differences
                  1.036

> wilcox.test(Sepal.Length,Petal.Length,paired=T)
        Wilcoxon signed rank test with continuity correction

data:  Sepal.Length and Petal.Length
V = 1275, p-value = 7.481e-10
alternative hypothesis: true mu is not equal to 0

> detach(virginica)
```

8 一元配置分散分析と多重比較

8.1 多群の比較の思想

3 群以上を比較するために，単純に 2 群間の差の検定を繰り返すことは誤りである。なぜなら，n 群から 2 群を抽出するやりかたは ${}_nC_2$ 通りあって，1 回あたりの第 1 種の過誤を 5%未満にしたとしても，3 群以上の比較全体として「少なくとも 1 組の差のある群がある」というと，全体としての第 1 種の過誤が 5%よりずっと大きくなってしまうからである。

この問題を解消するには，**多群間の比較という捉え方をやめて，群分け変数が注目している量の変数に与える効果があるかどうかという捉え方にする**のが一つの方法であり，具体的には，一元配置分散分析やクラスカル＝ウォリス (Kruskal-Wallis) の検定がこれに当たる。

別のアプローチとして，有意水準 5%の 2 群間の検定を繰り返すことによって全体としては大きくなってしまう**第 1 種の過誤を調整することによって，全体としての検定の有意水準を 5%に抑える**方法もある。このやり方は「多重比較」と呼ばれる。

これら 2 つのアプローチは別々に行うというよりも，段階を踏んで行うものと考えるのが一般的だが，永田，吉田 (1997) が指摘するように，段階を踏んで実行すると，ここにまた検定の多重性の問題が生じるので，両方はやるべきではない，という考え方にも一理ある【典拠：永田靖，吉田道弘『統計的多重比較法の基礎』，サイエンティスト社, 1997 年】。つまり，厳密に考えれば，群分け変数が量的変数に与える効果があるかどうかを調べたいのか，群間で量的変数に差があるかどうかを調べたいのかによって，これら 2 つのアプローチを使い分けるべきかもしれない。

段階を踏むとは，一元配置分散分析やクラスカル＝ウォリスの検定によって群間に何らかの差があると結論されてから，

初めてどの群とどの群の差があるのかを調べるために多重比較を使うという意味である。そのため，多重比較は *post hoc* な解析と呼ばれることがある。仮に多重比較で有意な結果が出たとしても，一元配置分散分析の結果が有意でなければ，偶然のばらつきの効果が群間の差よりも大きいということなので，特定群間の差に意味があると結論することはできない。

8.2 一元配置分散分析

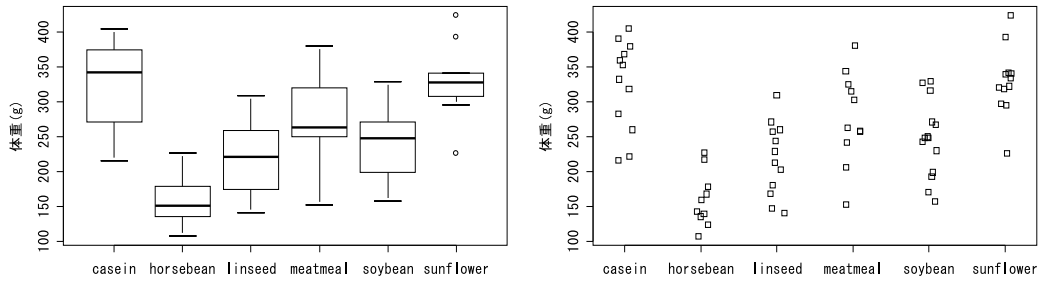
一元配置分散分析の思想は，データのばらつき（変動）を，群間の違いという意味のはっきりしているばらつき（群間変動）と，各データが群ごとの平均からどれくらいばらついているか（誤差）のすべての群についての合計（誤差変動）とに分解し，前者が後者よりもどれくらい大きいかを検討することによって，群分け変数がデータの変数に与える効果が誤差に比べて有意に大きいかどうかを調べるということである。帰無仮説は，「群分け変数がデータの変数に与える効果が誤差の効果に比べて大きくない」ということになる。言い換えると「すべての群の母平均が等しい」が帰無仮説である。

では，具体的に，R に含まれているデータ `chickwts` で説明しよう。これは，既に一部使ったが，71 羽の鶏を孵化直後にランダムに 6 群に分けて，それぞれ異なる餌を与え，6 週間後に何グラムになったかを示すデータである（R Console で `?chickwts` と入力してヘルプをみると，出典は，Anonymous (1948) *Biometrika*, 35: 214. である）。すべての値を下表に示す。

餌	その餌を食べて 6 週間育った鶏の体重 (g)
カゼイン (casein)	368, 390, 379, 260, 404, 318, 352, 359, 216, 222, 283, 332
ソラマメ (horsebean)	179, 160, 136, 227, 217, 168, 108, 124, 143, 140
アマニの種 (linseed)	309, 229, 181, 141, 260, 203, 148, 169, 213, 257, 244, 271
肉の配合餌 (meatmeal)	325, 257, 303, 315, 380, 153, 263, 242, 206, 344, 258
大豆 (soybean)	243, 230, 248, 327, 329, 250, 193, 271, 316, 267, 199, 171, 158, 248
ヒマワリの種 (sunflower)	423, 340, 392, 339, 341, 226, 320, 295, 334, 322, 297, 318

`chickwts` はデータフレームであり，体重を示す数値型変数 `weight` と，餌の種類を示す因子型変数 `feed` という形でデータが入っている。変数が 2 つで，オブザーベーションが 71 個という形になっている。餌の種類によって鶏の体重に差が出るかをみるためには，まずグラフ表示をしてみる。下枠内を打てば，層別箱ヒゲ図と並列ストリップチャートが横に並べて描かれるはずである[*28]。

```
> data(chickwts)
> attach(chickwts)
> layout(cbind(1,2))
> boxplot(weight~feed,ylab="体重 (g)")
> stripchart(weight~feed,vert=T,method="jitter",ylab="体重 (g)")
> detach(chickwts)
```



群間で体重に差がないという帰無仮説を検定するためには，`weight` という量的変数に対して，`feed` という群分け変数の

[*28] ここに示した一連のプロセスをまとめて `http://phi.med.gunma-u.ac.jp/medstat/it07-1.R` としてダウンロードできる。

効果をみる形で一元配置分散分析することになる。R Console に入力するコマンドは，summary(aov(weight~feed)) または anova(lm(weight~feed)) である。どちらでも同じ結果が下枠内の通りに得られる。後者は，一元配置分散分析が線形モデルの一種であることを利用している。

```
> attach(chickwts)
> summary(aov(weight~feed))
            Df Sum Sq Mean Sq F value    Pr(>F)
feed         5 231129   46226  15.365 5.936e-10 ***
Residuals   65 195556    3009
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> detach(chickwts)
```

このような結果の表を分散分析表という。右端の*の数は有意性を示す目安だが，有意確率そのもの（Pr(>F) の下の数字）に注目してみるほうがよい。

Sum Sq は，平方和 (sum of squares) の略である。

feed の Sum Sq の値 231129 は，餌の種類が異なる群ごとの平均値から総平均を引いて二乗した値を，餌の種類が異なる群ごとの鶏の個体数で重み付けした和である。群間変動または級間変動と呼ばれ，feed 間でのばらつきの程度を意味する。

Residuals の Sum Sq の値 195556 は各鶏の体重から，その鶏が属する餌群の鶏の平均体重を引いて二乗したものの総和であり，誤差変動と呼ばれ，餌群によらない（それ以外の要因がないとすれば偶然の）ばらつきの程度を意味する。

Mean Sq は平均平方和 (mean square) の略であり，平方和を自由度 (Df) で割ったものである。平均平方和は分散なので，feed の Mean Sq の値 46226 は群間分散または級間分散と呼ばれることがあり，Residuals の Mean Sq の値 3009 は誤差分散と呼ばれることがある。

F value は分散比と呼ばれ，群間分散の誤差分散に対する比である。この場合の分散比は第 1 自由度 5，第 2 自由度 65 の F 分布に従う。一般に，一元配置分散分析の場合は，対立仮説の下では $F > 1$ であることが期待されるため右片側検定すればよく，分散比がこの実現値よりも偶然大きくなる確率は，1-pf(15.365,5,65) で得られる。Pr(>F) の下の数字は，まさにその値を示すものである。この例では 5.936e-10 と[*29]，ほとんどゼロといえるくらい小さいので，feed の効果は 5%水準で有意であり，帰無仮説は棄却される。つまり，鶏の体重は，生後 6 週間に与えた餌の種類によって差があることになる。

ただし，一元配置分散分析は，各群が等しい母分散をもつ正規分布に従うことを仮定して行っているパラメトリックな分析法なので，データの分布がひどく歪んでいる場合は結果がおかしくなる場合がある。各群の母分散が等しいかどうかを調べる検定法として，バートレット (Bartlett) の検定と呼ばれる方法がある。R では bartlett.test(**量的変数~群分け変数**) で実行できる。帰無仮説「各群の母分散が等しい」が棄却された場合は，クラスカル＝ウォリスの検定（関数名は kruskal.test()）のようなノンパラメトリックな方法を使うことも考えられる。

この例では，bartlett.test(weight~feed) と入力して得られる結果の p-value をみると，0.66 であり，5%よりずっと大きいので帰無仮説は棄却されず，一元配置分散分析を実行しても問題ないことになる。以上の一元配置分散分析のプロセスをまとめると，下枠内のようになる。

[*29] 注：1-pf(15.365,5,65) の結果とは小数点以下第 3 位で 1 違うが，これは F value の丸め誤差による。

```
> attach(chickwts)
> print(res.bt <- bartlett.test(weight~feed))
        Bartlett test of homogeneity of variances

data:  weight by feed
Bartlett's K-squared = 3.2597, df = 5, p-value = 0.66

> ifelse(res.bt$p.value<0.05,
+ cat("不等分散！ Bartlett の検定で p=",res.bt$p.value,"\n"),
+ summary(aov(weight~feed)))
[[1]]
            Df Sum Sq Mean Sq F value    Pr(>F)
feed         5 231129   46226  15.365 5.936e-10 ***
Residuals   65 195556    3009
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> detach(chickwts)
```

8.3 多重比較

この鶏の体重の例では，一元配置分散分析の結果，餌群の効果が有意だったので，次に調べたいことは，具体的にどの餌とどの餌の間で差がでてくるかであろう。多重比較の方法についての詳細は「R による統計解析の基礎」などを参照されたいが，通常は R のデフォルトである Holm の方法で十分である。R で実行すると以下のようになり，カゼインとソラマメ，カゼインとアマニの種，カゼインと大豆，ソラマメと肉の配合餌，ソラマメと大豆，ソラマメとヒマワリの種，アマニの種とヒマワリの種，大豆とヒマワリの種の間は互いに有意差があるといえる。

```
> attach(chickwts)
> pairwise.t.test(weight,feed)
        Pairwise comparisons using t tests with pooled SD

data:  weight and feed

          casein  horsebean linseed meatmeal soybean
horsebean 2.9e-08 -         -       -        -
linseed   0.00016 0.09435   -       -        -
meatmeal  0.18227 9.0e-05   0.09435 -        -
soybean   0.00532 0.00298   0.51766 0.51766  -
sunflower 0.81249 1.2e-08   8.1e-05 0.13218  0.00298

P value adjustment method: holm
> detach(chickwts)
```

なお，前述の Fisher のアヤメのデータで，3 種間で花弁の長さ (Petal.Width) を比較することも，まったく同様の方法でできるので試されたい。

9 クロス集計と疫学

疫学研究では，2 つのカテゴリ変数の関係の分析をすることが多い。まずは 2 つのカテゴリ変数が独立である（つまり，関係がない）という帰無仮説を検定する方法について説明する。例えば，肺がんと判明した男性患者 100 人と，年齢が同じくらいの健康な男性 100 人を標本としてもってきて，それまで 10 年間にどれくらい喫煙をしたかという聞き取りを行うという「症例対照研究 (case control study)[*30]」を実施したとする。喫煙の程度を「一度も吸ったことがない」から「ずっとヘビースモーカーだった」まで何段階かのスコアを振れば，喫煙状況という変数と肺がんの有無という変数の組み合わせが得られる。そこで，それらが独立であるかどうか（関連がないかどうか）を検討することになる[*31]。

9.1 クロス集計とは？

カテゴリ変数のもつ統計的な情報は，カテゴリごとの度数だけである。そこで，2 つのカテゴリ変数の関係について検討したいときには，まず**それらの組み合わせの度数**を調べた表を作成する。これをクロス集計表と呼ぶ。例えば，下表の生データを考える。

人	1	2	3	4	5	6	7	8	9	10	11	12	13
性別	男	男	男	男	男	男	女	女	女	女	女	女	女
病気	有	有	有	無	無	無	有	有	有	有	無	無	無

この生データをもっとも簡単に R に入力し，性別と病気のクロス集計表を作るには次の枠内を打つ。下から 2 行目の `table()` という関数が，生のカテゴリ変数値の組み合わせをカウントしてクロス集計表を作ってくれ，最下行の `mosaicplot()` という関数が，それをグラフ表示してくれる。

```
> pid <- 1:13
> sex <- as.factor(c(rep(1,6),rep(2,7)))
> levels(sex) <- c("男","女")
> disease <- as.factor(c(1,1,1,2,2,2,1,1,1,1,2,2,2))
> levels(disease) <- c("有","無")
> print(ctab <- table(sex,disease))
    disease
sex  有 無
  男  3  3
  女  4  3
> mosaicplot(ctab,main="2 × 2 クロス集計表のモザイクプロット例")
```

とくに，2 つのカテゴリ変数が，この例のようにともに 2 値変数のとき，そのクロス集計は 2 × 2 クロス集計表（2 × 2 分割表）と呼ばれ，その統計的性質が良く調べられている。

クロス集計表ができてしまえば，独立性のカイ二乗検定（イェーツの連続性の補正済み）は `chisq.test(ctab)` でできる。

[*30] 患者対照研究ともいう。

[*31] ただし，むしろこのデザインは，肺がん患者は健康な人に比べて，どれくらい喫煙していた割合が高いか，を評価するためのデザインである。既に亡くなっている人が除かれてしまっているので，発生リスクは過小評価されるかもしれない。逆に，喫煙者と非喫煙者を 100 人ずつ集めて，その後の肺がん発生率を追跡調査する前向きのコホート研究 (cohort study) では，非喫煙群に比べて，喫煙者ではどれくらい肺がんの発生率が高いかを評価できる。「……に比べてどれくらい高いか」を示すためには，リスク比とかオッズ比のような「比」を用いるのが普通である。

例題

肺がんの患者 100 人に対して，1 人ずつ性・年齢が同じ健康な人を対照として 100 人選び[a]，それぞれについて過去の喫煙の有無を尋ねた結果，患者群では過去に喫煙を経験した人が 80 人，対照群では過去に喫煙を経験した人が 55 人だった。肺がんと喫煙は無関係といえるか？　独立性のカイ二乗検定をせよ。

[a] この操作をペアマッチサンプリングという。ただし，このような症例対照研究でマッチングをすると，却ってバイアスが生じる場合があるので注意されたい。

帰無仮説は，肺がんと喫煙が無関係（独立）ということである。クロス集計表を作ってみると，

	肺がん患者群	健康な対照群	合計
過去の喫煙経験あり	80	55	135
過去の喫煙経験なし	20	45	65
合計	100	100	200

となる。肺がんと喫煙が無関係という帰無仮説の下で期待される各カテゴリの人数は，

	肺がんあり	肺がんなし
喫煙あり	$135 \times 100/200 = 67.5$	$135 \times 100/200 = 67.5$
喫煙なし	$65 \times 100/200 = 32.5$	$65 \times 100/200 = 32.5$

となる。従って，連続性の補正を行なったカイ二乗統計量は，

$$\chi_c^2 = (80-68)^2/67.5 + (55-67)^2/67.5 + (20-32)^2/32.5 + (45-33)^2/32.5 = 13.128...$$

となり，自由度 1 のカイ二乗分布で検定すると `1-pchisq(13.128,1)` より有意確率は 0.00029... となり，有意水準 5%で帰無仮説は棄却される。つまり，肺がんの有無と過去の喫煙の有無は独立とはいえない。R では以下の通り。

```
> X <- matrix(c(80,20,55,45),nr=2)
> chisq.test(X)
        Pearson's Chi-squared test with Yates' continuity correction

data:  X
X-squared = 13.1282, df = 1, p-value = 0.0002909
```

この検定は，肺がん群と対照群の間で，過去の喫煙者の割合に差があるかどうかを検定することと数学的に同値である。下枠内を実行すれば，まったく同じ検定結果が得られる。

```
> smoker <- c(80,55)
> pop <- c(100,100)
> prop.test(smoker,pop)
```

カイ二乗検定や比率の差の検定は正規近似なので，サンプルサイズが小さいときは Fisher の正確な検定をした方がいい。既に `X` にクロス集計表が付値されているので，`fisher.test(X)` を実行すると，有意確率は `0.0002590` と得られ，有意水準 5%で「肺がんの有無と過去の喫煙の有無は独立」という帰無仮説は棄却される。なお，このように 2×2 クロス集計表を分析する場合は，`fisher.test()` 関数は，後述するオッズ比とその 95%信頼区間も同時に計算してくれる。

9.2 主な疫学指標

独立とはいえないなら，次に調べることは，どの程度の関連性があるのかということである。カテゴリ変数間の関連については，従来より疫学分野で多くの研究が蓄積されてきた。疫学研究では，研究デザインによって，得られる関連性の指標は異なることに注意しなければならない。その意味で，具体的な解析方法に入る前に，疫学の基礎知識が必要なのでまとめ

ておく。集団内の疾病の状況を表すためには，たんに患者数だけでは不十分である。まず，どのくらいの規模のどういう集団をどのくらいの期間観察したのか，という意味で，分母を定義することが必要である。具体的な指標としては，まず，以下の３つを区別する必要がある。

疫学指標の基礎知識 (1) 頻度の指標

有病割合 (prevalence)　有病率と呼ばれることもあるが，割合と呼ぶ方が紛れがない。一時点での人口に対する患者の割合で無次元である。一時点でのということを明示するには，point prevalence という。急性感染症で prevalence が高いなら患者が次々に発生していることを意味するが，慢性疾患の場合はそうとは限らない。行政施策として必要な医療資源や社会福祉資源の算定に役立つ。例：高血圧や高コレステロール血症は prevalence が高いので，対策がとられている。

累積罹患率 (cumulative incidence)　通常，たんにリスク (risk) といえば，この累積罹患率を指す。期首人口のうち観察期間中に病気になった人数の割合であり，無次元である。当然，観察期間が短ければ小さい値になるので，「20 年間のがんの発症リスク」のような表現になる。脱落者は分母から除外する（脱落分を正しく扱うためには生存時間解析という手法を用いる）。無作為割付けの介入研究でよく使われる指標である。

罹患率 (incidence rate)　発生率ともいう。個々の観察人年の総和で発生数を割った値。そのため，観察期間によらない値になる。次元は１／年。International Epidemiological Association の Last JM [Ed.]“A Dictionary of Epidemiology, 4th Ed.”(Oxford Univ. Press, 2001) に明記されているように，incidence は発生数である。感受性の人の中で新たに罹患する人が分子。再発を含む場合はそう明記する必要がある。意味としては，瞬時における病気へのかかりやすさ。つまり疾病罹患の危険度（ハザード）を示す。疾病発生状況と有病期間が安定していれば，平均有病期間＝有病割合／罹患率という関係が成り立つ。ランダム化臨床試験でよく使われる指標である。

さらに，オッズ (odds) という概念を押さえておく必要がある。オッズとは，**ある事象が起きる確率の起きない確率に対する比**である。一時点での非患者数に対する患者数の比を疾病オッズ (disease-odds) と呼ぶ。また，症例対照研究などで，過去に何らかの危険因子に曝露した人数の，曝露していない人数に対する比を曝露オッズ (exposure-odds) と呼ぶ。

頻度の指標を押さえた上で，何らかの危険因子への曝露があると，なかった場合に比べて何倍くらい病気に罹りやすさが上昇する**効果**をもつかといったことを推論することになる。効果の指標としては，以下の４つを区別しておこう。とくにリスク比やオッズ比はよく使われる指標である。

疫学指標の基礎知識 (2) 効果・関連性の指標

相対危険（Relative Risk）　以下の３つの総称。

- **リスク比** (risk ratio)　累積罹患率比（cumulative incidence rate ratio）ともいう。曝露群のリスクの非曝露群のリスクに対する比である。
- **罹患率比** (incidence rate ratio)　曝露群の罹患率の非曝露群の罹患率に対する比をいう。
- **死亡率比** (mortality rate ratio)　曝露群の死亡率の非曝露群の死亡率に対する比をいう。罹患率比と死亡率比を合わせて率比（rate ratio）という。

オッズ比 (odds ratio)　オッズの比。2 種類のオッズ比（コホート研究における累積罹患率のオッズ比と患者対照研究における曝露率のオッズ比）は数値としては一致する。オッズ比は比較的簡単に得られる値なので，率比の近似値として価値がある。

寄与危険 (attributable risk)　危険因子への曝露による発症増加を累積罹患率（リスク）または罹患率の差で表した値。つまり，累積罹患率差＝リスク差 (risk difference)，または罹患率差 (incidence rate difference) である。超過危険 (excess risk) ともいう。

寄与割合 (attributable proportion)　曝露群の罹患率のうちその曝露が原因となっている割合。つまり罹患率差を曝露群の罹患率で割った値になる。罹患率比から１を引いて罹患率比で割った値とも等しい。

要因の有無と病気の有無がまったく関係がなければ，リスク比もオッズ比も 1 になることが期待される。それぞれ信頼区間を計算して，例えば 95%信頼区間が 1 を含まなければ，5%水準で有意な関連が見られるといえる[*32]。

ところで，病気のリスクは，全体（期首人口）のうちで病気を発症する人の割合であったから，まず全体を把握していないと定義できない。つまり，まず観察対象全体で曝露群と対照群を把握しておいて，経時的に追跡調査して，それぞれの群で何人ずつ発症するかを調べるという，「前向き研究 (prospective study)」（この意味ではコホート研究 (cohort study) と

[*32] もっとも，Rothman や Greenland に代表される現代の疫学者は，有意性をみる仮説検定は，せっかく関連性の程度が得られているのに，それを有無という２値に還元してしまうので情報量の損失が大きく，あまり意味がないと言っている。疫学研究では検定結果よりも 95%信頼区間そのものの方が重要である。Rothman は関連の程度に応じた有意確率の変化を示すという意味で，p-value 関数を求めるべきだと主張している。

かフォローアップ研究 (follow-up study) と言ってもいい）でないと，リスク比（に限らず相対危険すべて）は計算できないことになる。

これに対して，症例対照研究（case-control study）[*33]とか断面研究（cross-sectional study）[*34]では，曝露時点での全体が未知なので，原理的にリスクを計算できないことになる。激しい曝露を受けた人は調査時点よりずっと前に病気を発症して死んでしまった可能性があるので，患者対照研究や断面研究から無理にリスクを見積もろうとするとリスクを過小評価してしまうことになるからである。

一方，オッズ比はどんなデザインの研究でも計算できることが利点である。断面研究や症例対照研究における曝露オッズの比を曝露オッズ比 (exposure-odds ratio), 断面研究やコホート研究における疾病オッズの比を疾病オッズ比 (disease-odds ratio) と呼ぶ。

では，クロス集計表から，これらの値を計算してみよう。以下の表（表☆としよう）を考えてみる。

	疾病あり	疾病なし	合計
曝露あり	a	b	m_1
曝露なし	c	d	m_2
合計	n_1	n_2	N

点推定量の計算は簡単である。この表でいえば，リスク比は

$$\frac{a/m_1}{c/m_2} = \frac{am_2}{cm_1}$$

となる。疾病オッズ比は

$$\frac{a/b}{c/d} = \frac{ad}{bc}$$

である。曝露オッズ比は

$$\frac{a/c}{b/d} = \frac{ad}{bc}$$

となり，**数値としては疾病オッズ比と一致する。**

ただし，R の `fisher.test()` で計算されるオッズ比は，ad/bc というこの単純な計算式から得られる値と異なっている。`fisher.test()` では周辺分布をすべて固定したクロス集計表の最初の要素に対して，非心度パラメータがオッズ比で与えられるような非心超幾何分布を仮定して最尤推定がなされる。また，`vcd` ライブラリの `oddsratio()` 関数で `log=F` オプションを付けると，どこかのセルが 0 のときは，$\frac{(a+0.5)(d+0.5)}{(b+0.5)(c+0.5)}$ が計算されるので一致しないが，そうでなければ定義通りの計算をしてくれる（この関数では対数オッズにしないと `summary(oddsratio())` による有意性の検定はできないが，`confint(oddsratio())` による信頼区間の推定は，対数オッズでなくてもできる。）。

オッズ比が重要なのは，稀な現象をみるときに，リスク比のよい近似になるからであると言われている。例えば，送電線からの高周波が白血病の原因になるという仮説を検証するために，送電線からの距離が近い場所に住んでいる人（曝露群）と，遠いところに住んでいる人（対照群）をサンプリングして，5年間の追跡調査をして，5年間の白血病の累積罹患率（リスク）を調査することを考えよう。白血病は稀な疾患だし，高周波に曝露しなくても発症することはあるので，このデザインでリスク比を計算するためには，莫大な数のサンプルをフォローアップする必要があり，大規模な予算とマンパワーが投入される必要があるだろう。

仮に調査結果が下表のようであったとすると，

	白血病発症	発症せず	合計
送電線近くに居住	4	99996	100000
送電線から離れて居住	2	99998	100000
合計	6	199994	200000

[*33] 調査時点で，患者を何人サンプリングすると決め，それと同じ人数の対照（その病気でないことだけが患者と違って，それ以外の条件はすべて患者と同じことが望ましい。ただし原則としてマッチングに使った変数で層別解析しなくてはいけない）を選んで，それぞれが過去に受けた曝露要因や，現在の生活習慣，態度などを調べることによって，その病気の原因を探る方法論。

[*34] 調べてみないと患者かどうかさえわからないような場合や，因果の向きがはっきりしない変数間の関係を見たいときは，全体で何人サンプリングすると決めて一時点で調査する。こういう方法論を断面研究という。

$(4/100000)/(2/100000) = 2$ から，リスク比が 2 なので，送電線の近くに住むことで白血病を発症するリスクは，送電線から離れて住む場合の 2 倍になったということができる。ここで疾病オッズ比をみると，$(4*99998)/(2*99996) \simeq 2.00004$ と，ほぼリスク比と一致していることがわかる。

こうして得られるリスク比は，確かに原理的に正しくリスクを評価するのだが，稀なリスクの評価のためには大規模な調査が必要になるので，効率が良いとはいえない。そこで，通常は，前向きコホート研究ではなく，症例対照研究を行って，過去の曝露との関係をみる。この場合だったら，白血病患者 100 人と対照 100 人に対して，過去に送電線の近くに居住していたかどうかを聞くわけである。それで得られた結果が，仮に下表のようになったとしよう。

	白血病	白血病でない	合計
送電線近くに居住した経験あり	20	10	30
送電線から離れて居住	80	90	170
合計	100	100	200

この場合，白血病かつ送電線の近くに居住した経験がある 20 人は，送電線の近くに住んだ経験がある人からのサンプルではなく，白血病患者からのサンプルなので，リスク（累積罹患率）が定義できず，リスク比も計算できない。形の上から無理やり計算しても意味はない。しかし，曝露オッズは計算できる。白血病の人の送電線の近くに居住した経験の曝露オッズは 0.25 となり，白血病でない人ではそのオッズが 0.111... となるので，これらの曝露オッズの比は 2.25 となる。この値は母集団におけるリスク比のよい近似になることが知られているので，このように稀な疾患の場合は，大規模コホート研究をするよりも，症例対照研究で曝露オッズ比を求める方が効率が良い。

原理的に前向き調査ができない場合もある。とくに，薬害と呼ばれる現象は，妙な病気が見つかったときに，後付けで原因を探ることになるので，症例対照研究にならざるを得ない。例えば，スモンとかサリドマイドは，そうやって原因がわかった問題である。腕が短く生まれた子どもの母親と，そうでない子どもの母親に，妊娠中に飲んだ薬の有無を尋ねて，特定の時期にサリドマイドを飲んだという曝露による疾病オッズ比が有意に大きい結果が得られたのだ[*35]。

また，問題があるかどうかが事前に明らかでない場合は，断面研究をせざるを得ない。聞き取りや質問紙などで調べる，心理学的，あるいは社会学的な調査項目間の関係を見る場合は，断面研究をする場合が多い。

目的によっては，リスク比やオッズ比の他に，２つのカテゴリ変数の関連性を表す指標として，寄与危険（＝リスク差），寄与割合（＝曝露寄与率），相対差，母集団寄与率，Yule の Q，ファイ係数といった指標も用いられるけれども，これらは点推定量だけが求められることが多い。また，同じ質問を 2 回した場合に同じ変数がどれくらい一致するかについては，普通にクロス集計表を作って独立性の検定ができそうな気がするかもしれないが，してはいけない。独立でないことは自明だからである。この場合は test-retest-reliability を測ることになるので，κ 係数などの一致度の指標を計算するべきである（これらについては後述する）。

リスク比とオッズ比の 95%信頼区間を考えよう。**まずリスク比の場合**から考えると，原則としては前向き研究でないとリスク比は計算できないので，曝露あり群となし群をそれぞれ m_1 人，m_2 人フォローアップして，曝露あり群で X 人，なし群で Y 人が病気を発症したとする。得られる表は，

	発症	発症なし	合計
曝露あり	X	$m_1 - X$	m_1
曝露なし	Y	$m_2 - Y$	m_2
合計	$X + Y$	$N - X - Y$	N

となる。このとき，母集団でのリスクの点推定量は，曝露があったとき $\pi_1 = X/m_1$，曝露がなかったとき $\pi_2 = Y/m_2$ である。リスク比の点推定量は $RR = \pi_1/\pi_2 = (Xm_2)/(Ym_1)$ となる。

リスク比の分布は N が大きくなれば正規分布に近づくので，正規分布を当てはめて信頼区間を求めることができるが，普通は右裾を引いているので対数変換か立方根変換（Bailey の方法）をしなくてはならない。対数変換の場合，95%信頼区間の下限と上限はそれぞれ，

$$RR \cdot \exp(-\mathrm{qnorm}(0.975)\sqrt{1/X - 1/m_1 + 1/Y - 1/m_2}) \quad \text{（下限）} \tag{1}$$

[*35] ここで有意と書いたが，統計的に有意かどうかをいうためには，検定するか，95%信頼区間を出さねばならない。その方法は後述する。

$$RR \cdot \exp(\text{qnorm}(0.975)\sqrt{1/X - 1/m_1 + 1/Y - 1/m_2}) \quad \text{（上限）} \tag{2}$$

となる。なお，RR が大きくなると対数変換ではうまく近似できないので，立方根変換しなくてはいけないが，複雑なのでここでは説明しない。

前述の白血病の例での計算は，下枠内のようにすればよい。リスク比の点推定量は 2 となり，95%信頼区間は，$(0.37, 10.9)$ となる。つまりこの曝露によって白血病の罹患リスクは 2 倍になったと考えられるが，95%信頼区間をみると 1 を跨いでいるので 5%水準で有意ではなく，曝露の有無によって白血病罹患リスクに差がないという帰無仮説は棄却されない。

```
> riskratio2 <- function(X,Y,m1,m2) {
+  data <- matrix(c(X,Y,m1-X,m2-Y,m1,m2),nr=2)
+  colnames(data) <- c("疾病あり","疾病なし","合計")
+  rownames(data) <- c("曝露群","対照群")
+  print(data)
+  RR <- (X/m1)/(Y/m2)
+  RRL <- RR*exp(-qnorm(0.975)*sqrt(1/X-1/m1+1/Y-1/m2))
+  RRU <- RR*exp(qnorm(0.975)*sqrt(1/X-1/m1+1/Y-1/m2))
+  cat("リスク比の点推定量: ",RR,"  95%信頼区間 = [ ",RRL," , ",RRU," ]\n")
+ }
> riskratio2(4,2,100000,100000)
```

ちなみに epitools ライブラリの riskratio() 関数は，この意味でのリスク比を計算する関数ではないので注意が必要である。ただし，別に rateratio() という関数があって，分母を観察人年とした率比とその信頼区間は計算してくれる。信頼区間の計算は”midp”または”wald”または”boot”の 3 種類が指定できる。対照群のデータを曝露群のデータより先に指定することに注意しなければならないが，比較的使い方は簡単である。なお，この関数は，method="wald"オプションをつけないと，点推定量についても median-unbiased な推定値を計算するので，率比といっても単純な率の比とはやや異なる。epitools ライブラリがインストールされているコンピュータであれば，下枠内を入力することによって率比が計算できる（簡単のため曝露群でも対照群でも白血病発症時点は観察終了直前だったとする）。この場合，率比の点推定量は 2，95%信頼区間は $(0.37, 10.9)$ であり，上枠内の計算結果と一致する（ただし，median-unbiased な推定結果だと，これよりかなり幅が広くなる）。

```
> library(epitools)
> rateratio(c(2,4,5*100000,5*100000),method="wald")
```

次に**オッズ比の信頼区間**を考える。表☆の a, b, c, d という記号を使うと，オッズ比の点推定値 OR は，$OR = (ad)/(bc)$ である。オッズ比の分布も右裾を引いているので，対数変換または Cornfield (1956) の方法によって正規分布に近づけ，正規近似を使って 95%信頼区間を求めることになる。対数変換の場合，95%信頼区間の下限は

$$OR \cdot \exp(-\text{qnorm}(0.975)\sqrt{1/a + 1/b + 1/c + 1/d})$$

であり，上限は

$$OR \cdot \exp(\text{qnorm}(0.975)\sqrt{1/a + 1/b + 1/c + 1/d})$$

となる。Cornfield の方法の方が大きなオッズ比については近似がよいが，手順がやや複雑である（高次方程式の解を Newton 法などで数値的に求める必要がある）ため，ここでは扱わない。

白血病の例で計算するには，下枠内を打てばよい。疾病オッズ比の点推定量は前述の通り 2.00004 であり，95%信頼区間は $(0.37, 10.9)$ となって，リスク比とほぼ一致することがわかる（稀な疾患であるため）。

```
> oddsratio2 <- function(a,b,c,d) {
+  data <- matrix(c(a,b,a+b,c,d,c+d,a+c,b+d,a+b+c+d),nr=3)
+  colnames(data) <- c("疾病あり","疾病なし","合計")
+  rownames(data) <- c("曝露群","対照群","合計")
+  print(data)
+  OR <- (a*d)/(b*c)
+  ORL <- OR*exp(-qnorm(0.975)*sqrt(1/a+1/b+1/c+1/d))
+  ORU <- OR*exp(qnorm(0.975)*sqrt(1/a+1/b+1/c+1/d))
+  cat("オッズ比の点推定量: ",OR,"  95%信頼区間 = [ ",ORL," , ",ORU," ]\n")
+ }
> oddsratio2(4,2,99996,99998)
```

オッズ比を最尤推定しようとすると，Windows マシンでは，Fisher の正確な確率を計算する過程でメモリが不足しないようにするため，デフォルトでは 200000 になっている workspace を大きめに指定しなければならない。vcd ライブラリの oddsratio() 関数では定義式通り 2.00004 が得られる。

```
> fisher.test(matrix(c(4,2,99996,99998), nr=2),workspace=10000000)
        Fisher's Exact Test for Count Data

data:  matrix(c(4, 2, 99996, 99998), nr = 2)
p-value = 0.6875
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
  0.2866293 22.1101755
sample estimates:
odds ratio
  2.000022
> require(vcd)
> oddsratio(matrix(c(4,2,99996,99998), nr=2),log=F)
[1] 2.00004
```

vcd ライブラリの oddsratio() 関数を使って 95%信頼区間を推定するには，confint() 関数を用いる。点推定量が 2.00004，95%信頼区間が $[0.43, 9.39]$ となり，対数変換を使った計算結果よりも，やや幅が狭いけれども，本質的な違いはないことがわかる。

```
> require(vcd)
> OR <- oddsratio(matrix(c(4,2,99996,99998),nr=2),log=F)
> ORCI <- confint(OR)
> cat("オッズ比の点推定量: ",OR,"  95%信頼区間 = [ ",ORCI[1]," , ",ORCI[2]," ]\n")
```

9.3 その他の関連性の指標

寄与危険（リスク差） 曝露群のリスクと対照群のリスクの差である。リスク比の計算で用いた記号で表せば，$\pi_1 - \pi_2$

寄与割合（曝露寄与率） 真に要因の影響によって発症した者の割合。同じ記号で表せば，$(\pi_1 - \pi_2)/\pi_1$

相対差 要因ももたず発症もしていない者のうち，要因をもった場合にのみ発症する割合。同じ記号で表せば，$(\pi_1 - \pi_2)/(1 - \pi_2)$

母集団寄与率 母集団において真に要因の影響によって発症した者の割合。$\pi = (X + Y)/(m_1 + m_2)$ として，$(\pi - \pi_2)/\pi$

ユールの Q オッズ比を -1 から 1 の値を取るようにスケーリングしたもの。$Q = (OR - 1)/(OR + 1)$。独立な場合は 0 となる。

ファイ係数 (ϕ)　要因の有無，発症の有無を 1,0 で表した場合のピアソンの積率相関係数である。θ_1, θ_2 を発症者中の要因あり割合，非発症者中の要因あり割合として，$\phi = \sqrt{(\pi_1 - \pi_2)(\theta_1 - \theta_2)}$。この値は 2×2 に限らず，一般の $k \times m$ の分割表について計算でき，ピアソンのカイ二乗統計量 χ_0^2 と総人数 n を用いて，$\sqrt{\chi_0^2/n}$ と定義される。k と m のどちらか小さな方の値が t だとすると，ファイ係数は 0 から $\sqrt{t-1}$ の範囲をとる。

ピアソンのコンティンジェンシー係数 C　ファイ係数はカテゴリ数の影響を受けるので，それを除去したものである。ファイ係数を用いて，$C = \sqrt{\phi^2/(1+\phi^2)}$ として計算される。取りうる値の範囲は 0 から $\sqrt{(t-1)/t}$ である。

クラメールの V　ファイ係数を用いて，$V = \phi/\sqrt{t-1}$ と表せる。取りうる値の範囲は 0 から 1 となり，変数のカテゴリ数によらないのが利点である。

なお，ファイ係数，ピアソンのコンティンジェンシー係数，クラメールの V（これらは総称して属性相関係数と呼ばれることがある）は vcd ライブラリの assocstats() 関数で計算できる。この関数は，これらの係数の他，関連がないという仮説検定を実行してピアソンのカイ二乗統計量と尤度比カイ二乗統計量（ここでは説明しないが，多くの場合にピアソンのカイ二乗統計量を使った通常のカイ二乗検定よりもよいとされる），さらに，それらの有意確率を計算してくれる。属性相関係数はすべてピアソンのカイ二乗統計量に基づいて計算されるので，その有意性検定はカイ二乗検定の結果と等価と考えてよい。上記白血病のコホート研究の例でこれらを計算するには下枠内を打てばよいが，これらの係数の値はすべて 0.002 となるので，ほぼ関連はないと判定される。

```
> require(vcd)
> assocstats(matrix(c(4,2,99996,99998),nr=2))
```

9.4　κ 統計量

2 回の繰り返し調査をしたときに，あるカテゴリ変数がどれくらい一致するかを示す指標である。test-retest reliability（検査再検査信頼性）の指標といえる。カテゴリ変数間の一致度をみるための作図には，vcd ライブラリに含まれている agreementplot() という関数が有用である。

	2 回目○	2 回目×	合計
1 回目○	a	b	m_1
1 回目×	c	d	m_2
合計	n_1	n_2	N

という表から，偶然でもこれくらいは一致するだろうと思われる値は，1 回目と 2 回目の間に関連がない場合の各セルの期待値を足して全数で割った値になるので $P_e = (n_1 \cdot m_1/N + n_2 \cdot m_2/N)/N$，実際の一致割合（1 回目も 2 回目も○か，1 回目も 2 回目も×であった割合）は $P_o = (a+d)/N$ とわかる。ここで，$\kappa = (P_o - P_e)/(1 - P_e)$ と定義すると，κ は，完全一致のとき 1，偶然と同じとき 0，それ以下で負となる統計量となる。

κ の分散 $V(\kappa)$ は，$V(\kappa) = P_e/(N \cdot (1 - P_e))$ となるので，$\kappa/\sqrt{V(\kappa)}$ が標準正規分布に従うことを利用して，帰無仮説「$\kappa = 0$」を検定したり，κ の 95%信頼区間を求めたりすることができる。下枠内は，2×2 のクロス集計表を与えたときに，κ の点推定量と 95%信頼区間と有意確率を計算する R の関数を定義してから，○×で回答する項目について 2 回の繰り返し調査をしたときに，1 度目も 2 度目も○であった人数が 10 人，1 度目は○で 2 度目は×であった人数が 2 人，1 度目は×で 2 度目は○であった人数が 3 人，1 度目も 2 度目も×であった人数が 19 人であったときにその計算を実行させる命令である。

```
> kappa.test <- function(x) {
+  x <- as.matrix(x)
+  a <- x[1,1]; b <- x[1,2]; c <- x[2,1]; d <- x[2,2]
+  m1 <- a+b; m2 <- c+d; n1 <- a+c; n2 <- b+d; N <- sum(x)
+  Pe <- (n1*m1/N+n2*m2/N)/N
+  Po <- (a+d)/N
+  kappa <- (Po-Pe)/(1-Pe)
+  seK0 <- sqrt(Pe/(N*(1-Pe)))
+  seK <- sqrt(Po*(1-Po)/(N*(1-Pe)^2))
+  p.value <- 1-pnorm(kappa/seK0)
+  kappaL<-kappa-qnorm(0.975)*seK
+  kappaU<-kappa+qnorm(0.975)*seK
+  list(kappa=kappa,conf.int=c(kappaL,kappaU),p.value=p.value)
+ }
> kappa.test(matrix(c(10,3,2,19),nr=2))
$kappa
[1] 0.6840149

$conf.int
[1] 0.4282215 0.9398082

$p.value
[1] 9.907563e-05
```

vcd ライブラリの Kappa() 関数は $m \times m$ のクロス集計表について，重みなしと重みつきで κ 係数を計算してくれる[*36]。結果を confint() 関数に渡せば信頼区間も推定できる。同じデータに適用するには，下枠内のように打つ。上枠内の結果と同じ結果が得られる[*37]。

```
> require(vcd)
> print(myKappa <- Kappa(matrix(c(10,3,2,19),nr=2)))
> confint(myKappa)
```

10　2 変数の検定のまとめ

よく用いられる 2 変数の検定の一覧表を以下にまとめる。なお，3 変数以上の関係については，次節で述べるモデルの当てはめをするか，限定や層別化などで 1 つ以上の交絡要因の影響を調整して 2 つの変数間の関係をみるか，大雑把に言って 2 つの戦略のどちらかを取ることになる。

例えば，カテゴリ変数 C3 で層別したどの層でも 2 つのカテゴリ変数 C1 と C2 が独立かをみたいときは，mantelhaen.test(C1,C2,C3) とする。または，TMP <- table(C1,C2,C3) として 3 次元のクロス集計表 TMP を作ってから，mantelhaen.test(TMP) としてもよい。カテゴリ変数 C で層別したどの層でも 2 つの 2 値変数 B1 と B2 の間に

[*36] 重みは，Po や Pe を計算する際に weights=オプションを指定しないとき，あるいは weights="Equal-Spacing"にマッチしない任意の文字を指定した場合は，weights="Fleiss-Cohen"と指定したのと同じで，カテゴリ数が nc として 1-(abs(outer(1:nc,1:nc,"-"))/(nc-1))^2 となり，weights="Equal-Spacing"を指定したときは 1-abs(outer(1:nc,1:nc,"-"))/(nc-1) が重みとなる。つまり，○×の一致をみるならカテゴリ数は 2 なので，重みはどちらの方法でも matrix(c(1,0,0,1),nc=2) となる。

[*37] 有意確率は出ないが，κ 係数は，有意性の検定をするよりも，95 %信頼区間を示すことと，目安としての一致度の判定基準（負だと poor な一致，0-0.2 で slight な一致，0.21-0.4 で fair な一致，0.41-0.6 で moderate な一致，0.61-0.8 で substantial な一致，0.81-0.99 で almost perfect な一致，1 で perfect な一致とする，Landis and Koch, 1977, Biometrics, 33: 159-174 など）を参照して一致度を判定するという使い方が普通らしく，vcd ライブラリでもそのような実装がされているのだと思われる。考えてみれば，一致度を評価する上で $\kappa = 0$ という帰無仮説の検定には意味が乏しいのは当然かもしれない。

同じ関連があるかどうかを見たい場合，例えばオッズ比の均質性を検定するための Woolf の検定は，vcd ライブラリの woolf.test(table(B1,B2,C)) で実行できる。また，サンプルサイズが十分に大きければ，カテゴリ変数で層別したそれぞれの層で別々に 2 つの変数間の関係を分析する「限定」も強力な分析法である。例えば subset() 関数を使ってデータフレームを男性と女性別々に分けておき，男性についての分析と女性についての分析を別々に行い，解釈も別々にすることは「限定」に当たる。目的によってはすべての層について分析するのではなく，特定の層についてだけ分析することもある。

カテゴリ変数間の独立性のカイ二乗検定	chisq.test(table(C1,C2))
カテゴリ変数間の独立性の Fisher の直接確率	fisher.test(table(C1,C2))
オッズ比とその信頼区間	library(vcd); summary(oddsratio(table(B1,B2),log=F))
カテゴリ変数間の関連性：ファイ係数とクラメールの V	library(vcd); assocstats(table(C1,C2))
2 回の繰り返しの一致度：カッパ係数	library(vcd); Kappa(table(C1,C2))
順序変数×カテゴリ変数の出現頻度の傾向= Cochran-Armitage の検定	prop.trend.test(table(B,I)[2,],table(I))
2 つのカテゴリ間で正規分布する量の分散に差があるか：等分散性の検定（いわゆる F 検定）	var.test(X~B)
2 群間で正規分布する量に差があるか（等分散のとき）：平均値の差の検定（t 検定）	t.test(X~B,var.equal=T)
2 群間で正規分布する量に差があるか（不等分散のとき）：平均値の差の Welch の方法	t.test(X~B)
2 群間で正規分布しない量に差があるか：Wilcoxon の順位和検定	wilcox.test(X~B)
対応のある 2 つの正規分布する量の差の検定：paired-t 検定	t.test(X,Y,paired=T)
正規分布する量の分散がカテゴリ間で差がないか：バートレットの検定	bartlett.test(X~C)
正規分布する量がカテゴリ間で差がないか：一元配置分散分析＋多重比較	等分散なら aov(X~C) で一元配置分散分析し，C の主効果が有意なら TukeyHSD(aov(X~C)) または pairwise.t.test(X,C)。前者は Tukey の方法，後者は Holm の方法で多重比較。library(multcomp) すれば，simtest(X~C,type="Dunnett") でダネットの多重比較（対照群との比較），simtest(X~C,type="Williams") でウィリアムズの多重比較もできる。不等分散でもやってしまう場合もあるが，クラスカル・ウォリスの検定を使うこともある。
正規分布しない量×カテゴリ変数：クラスカル・ウォリスの検定＋ホルムの方法で調整した Wilcoxon の順位和検定を多重実行	kruskal.test(X~C) と pairwise.wilcox.test(X,C)
量×量：ピアソンの相関係数を用いた無相関の検定	cor.test(X,Y)
量×量：スピアマンの順位相関係数を用いた無相関の検定	cor.test(X,Y,method="spearman")
量×量：ケンドールの順位相関係数を用いた無相関の検定	cor.test(X,Y,method="kendall")

11 モデルのあてはめ

かなり強い法則性を仮定して立てたモデルを，データに当てはめることによって，データのばらつきがかなりの程度説明されれば，そのモデルはデータに内在する法則性の妥当な解釈を与えると判断できる。具体的なモデルとしては，単回帰，重回帰，共分散，ロジスティック回帰を扱う。一般化線型モデル (Generalized Linear Model) は，基本的には，

$$Y = \beta_0 + \beta X + \varepsilon$$

という形で表される（Y が従属変数群[*38]，X が独立変数群（及びそれらの交互作用項），β_0 が切片群，β が係数群，ε が誤差項である)。係数群は未定であり，そのモデルがもっとも良くデータに当てはまるようになる数値を，最小二乗法または最尤法で求めるのが普通である。こうして得られる係数は，通常，偏回帰係数と呼ばれ，互いに他の独立変数の影響を調整した，各独立変数独自の従属変数への影響を示す値と考えられる（なお，相対的にどの独立変数の影響が大きいかをみるときは，独立変数のスケールに依存してしまう偏回帰係数で比較することはできず，標準化偏回帰係数を用いる[*39])。R では，`glm()` という関数を使ってモデルを記述するのが基本である。通常の線型モデル `lm()` に比べて「一般化」されているのは，従属変数が従う分布である。通常の線型モデルでは従属変数も正規分布に従う必要があるが，一般化線型モデルでは二項分布でもいいしポアソン分布でもいい。外部ライブラリとして，もっと凝ったモデル記述とその当てはめを行うためのパッケージがいくつも開発され，CRAN で公開されている。また，一般化線型モデルとは違うモデルとして，独立変数群の効果が線型結合でない（例えば，ある独立変数の二乗に比例した大きさの効果があるような場合)，いわゆる非線型モデルも `nls()` という関数で扱うことができる。

11.1 モデルの記述法

R の `glm()` 関数における一般化線型モデルの記述は，例えば，(1) 独立変数群が `X1` と `X2` で，従属変数が `Y` であり，`Y` が正規分布に従う場合，(2)(1) と同じ構造だが切片がゼロとして係数を推定したい場合，(3)`dat` というデータフレームに従属変数 `Y` と，その他すべての変数が独立変数として含まれていて，`Y` が 2 値変数である場合，(4) 独立変数群がカテゴリ変数 `C1`，`C2` と，それらの交互作用項で，従属変数が正規分布に従う量的変数 `Y` である場合，について順に示すと，下枠内のようになる。

```
> glm(Y ~ X1+X2)
> glm(Y ~ X1+X2-1)
> glm(Y ~ ., data=dat, family="binomial")
> glm(Y ~ C1+C2+C1:C2)
```

`family` のデフォルトは`"gaussian"`なので，上 2 行のように `family` を指定しなければ正規分布を仮定することになる。この場合，モデルとしては単純な線型重回帰モデルとなるため，例えば (1) の場合なら `lm(Y ~ X1+X2)` と同等である。`summary(lm())` ならば自由度調整済み重相関係数の二乗が得られるので，従属変数にも正規母集団を仮定できる単純な線型重回帰で済むときは，`lm()` を使うことを薦める。(4) も従属変数が正規分布に従うので，`lm()` の方がよい。また，独立変数が複数のカテゴリ変数であるときに，主効果と交互作用項のすべてを指定するには，`*`で変数名をつなぐ方法もあり，(4) の右辺は `C1*C2` とも書ける。(4) のモデルは二元配置分散分析なので，結局，`anova(lm(Y ~ C1*C2))` とするのが普通である。

また，これらのモデルの当てはめの結果は，`res <- glm(Y ~ X1+X2)` のようにオブジェクトに保存しておくことができ，`plot(residuals(res))` として残差プロットをしたり，`summary(res)` として詳細な結果を出力させたり，`AIC(res)` として AIC を計算させたり，`step(res)` として変数選択をさせたりするのに使える。

[*38] 変換したものである場合もある

[*39] なお，標準化偏回帰係数は，各偏回帰係数に各独立変数の不偏標準偏差を掛け，従属変数の不偏標準偏差で割れば得られる。R での求め方は後述する。

11.2 変数の種類と数の違いによる線型モデルの分類

以下のように整理すると，t 検定，分散分析，回帰分析といった分析法が，すべて一般化線型モデルの枠組みで扱えることがわかる。

分析名	従属変数 (Y)	独立変数 (X)
t 検定（注 1）	量的変数１つ	２値変数１つ
一元配置分散分析	量的変数１つ	カテゴリ変数１つ
多元配置分散分析（注 2）	量的変数１つ	カテゴリ変数複数
（単）回帰分析	量的変数１つ	量的変数１つ
重回帰分析	量的変数１つ	量的変数複数（注 3）
共分散分析	量的変数１つ	（注 4）
ロジスティック回帰分析	２値変数１つ	２値変数，カテゴリ変数，量的変数複数
正準相関分析	量的変数複数	量的変数複数

（注 1） Welch の方法でない場合。

（注 2） 独立変数となるカテゴリ変数（因子）が２つの場合は二元配置分散分析，３つなら三元配置分散分析と呼ばれる。独立変数はカテゴリ変数そのものだけでなく，交互作用項も含めるのが普通である。なお，分散分析をするときには変数ごとに平方和を求めるわけだが，二元配置以上では平方和の求め方が Type I から Type IV まで 4 通りあるので注意が必要である[*40]。

（注 3） カテゴリ変数はダミー変数化せねばならない。

（注 4） ２値変数１つと量的変数１つの場合が多いが，「２値変数またはカテゴリ変数１つまたは複数」と「量的変数１つまたは複数」を両方含めば使える。

11.3 重回帰分析についての留意点

重回帰分析が独立変数１つの回帰分析よりも優れている点は，複数の独立変数を同時にモデルに投入することにより，従属変数に対する，他の影響を調整した個々の変数の影響をみることができることである。

重回帰分析は，何よりもモデル全体で評価することが大切である。例えば，独立変数が年齢と体重と一日当たりエネルギー摂取量，従属変数が血圧というモデルを立てれば，年齢の偏回帰係数（または偏相関係数または標準化偏回帰係数）は，

[*40] 分散分析表にでてくる因子の残差平方和の出し方としては，因子が直交していれば（因子間の交互作用がなければ），他の因子を加える順序によらず一定になるので，他の因子を含まない単独のモデルで出した平方和をそのままその因子の平方和とみなしていい（これが逐次平方和と呼ばれる Type I SS）けれど，因子が直交していないときは別の考え方をする必要があって，そこで出てくるのが，Type II とか Type III の平方和である。

Type II は，まずすべての因子の主効果を含むモデルを基準にして，それから１つの因子を取り去ったモデルのモデル平方和と元のモデルのモデル平方和の差を，取り去った因子の寄与とみなして，その因子の偏平方和（Type II SS）とし，次に 2 因子交互作用を含むモデルを基準にして，交互作用を取り去ったモデルのモデル平方和とのモデル平方和の差を交互作用効果の偏平方和とするというもの。

Type III は繰り返し数が不揃いのときにデータ数の少ないセルを他のセルと同等とみなす目的で使うものだが，同等とみなすと逆にバイアスが生じる可能性もあるので，不揃いでも Type II を使うべきという意見もある。Type IV は SAS には入っているが，あまり使われない。高橋・大橋・芳賀「SAS による実験データの解析」（東大出版会）によると，数量化一類をするときや，乱塊法の場合や， MANOVA の場合や欠損値がある場合は Type II の使用が薦められるとあるので，とりあえず Type I と Type II だけ出せれば充分ではないかと思う。なお，同書の 16 章には，行列言語 IML で Type III を計算する方法が載っている。

R の場合，標準の `anova()` や `aov()` では Type I の平方和が計算されるが（`anova(lm())` が `aov()` と同じ意味），`car` パッケージの `Anova()` では Type II または Type III（後者を出すには type="III"という引数をつける）の平方和が計算できる。ただ，`library(car)` してから `help(Anova)` すると，`Anova()` 関数で計算される Type II は SAS の Type II と同じだが Type III は微妙に違うので注意して使えと書かれている。`car` パッケージの開発者 John Fox の著書 “An R and S-PLUS companion to applied regression.” の p.140 の Type III の説明によると，例えば因子 A の主効果を，因子 B の主効果と因子 A と因子 B の交互作用効果をテストした後でテストしたいような場合に他の効果のすべてを出した後で因子 A によって加えられる分を Type III として計算するとのことである。たしかに SAS の計算アルゴリズムとは違うようである。

結論としては，R で，因子が直交していなくてセルごとの繰り返し数が不揃いの二元配置分散分析をしたいときは，`library(car)` としてから，`Anova(lm(Y~C1*C2))` を使えば Type II の平方和，つまり偏平方和が計算されるので，そうすることをお薦めする。

体重と一日当たりエネルギー摂取量の影響を調整した（取り除いた）後の年齢と血圧の関係を示す値だし，体重の偏回帰係数は年齢と一日当たりエネルギー摂取量の影響を調整した後の体重と血圧の関係を示す値だし，一日当たりエネルギー摂取量の偏回帰係数は，年齢と体重の影響を調整した後の一日当たりエネルギー摂取量と血圧の関係を示す値である。独立変数が年齢と体重で従属変数が血圧である場合の年齢の偏回帰係数は，独立変数に一日当たりエネルギー摂取量も入っている場合の年齢の偏回帰係数とは**異なる**。したがって，別のモデル間で変数の寄与の大小を比較することは，原則として不可能である。

モデル全体としてのデータへの当てはまりは，重相関係数の２乗（決定係数）や，AIC で評価する。

あるモデルの中で，各独立変数が他の独立変数の影響を調整した上でも従属変数に有意な影響を与えているかどうかをみるには，独立変数ごとに，偏回帰係数の有意性検定を行う。ある独立変数の偏回帰係数がゼロという帰無仮説を検定するには，その変数と従属変数の間の偏相関係数がゼロという帰無仮説を t 分布を使って検定すればよい。また，１つの重回帰モデルの中で，相対的にどの独立変数が従属変数（の分散）に対して大きな影響を与えているかは，偏相関係数の二乗の大小によって評価するか，または標準化偏回帰係数によって比較することができる。標準化偏回帰係数の計算は，例えば下枠内のようにすると，すべての独立変数について一度に得られるので便利である。

```
> res <- lm(Y ~ X+Z)
> sdd <- c(0,sd(X),sd(Z))
> stb <- coef(res)*sdd/sd(Y)
> print(stb)
```

11.4 **多重共線性** (multicolinearity)

一般に，複数の独立変数がある場合の回帰で，独立変数同士に強い相関があると，重回帰の係数推定が不安定になるのでうまくない。ごく単純な例でいえば，従属変数 `Y` に対して独立変数群 `X1` と `X2` が相加的に影響していると考えられる場合，`lm(Y ~ X1+X2)` という重回帰モデルを立てるとしよう。ここで，実は `X1` が `X2` と強い相関をもっているとすると，もし `X1` の標準化偏回帰係数の絶対値が大きければ，`X2` による効果もそちらで説明されてしまうので，`X2` の標準化偏回帰係数の絶対値は小さくなるだろう。まったくの偶然で，その逆のことが起こるかもしれない。従って，係数推定は必然的に不安定になる。この現象は，独立変数群が従属変数に与える線型の効果を共有しているという意味で，多重共線性 (multicolinearity) と呼ばれる。

多重共線性があるかどうかを判定するには，独立変数間の散布図を１つずつ描いてみるなど，丁寧な吟味をすることが望ましいが，各々の独立変数を，それ以外の独立変数の従属変数として重回帰分析したときの重相関係数の 2 乗を 1 から引いた値の逆数を VIF（Variance Inflation Factor; 定訳は不明だが，分散増加因子と訳しておく）として，VIF が 10 を超えたら多重共線性を考えねばならないという基準を使う（Armitage et al. 2002）のが簡便である。多重共線性があるときは，拡張期血圧（DBP）と収縮期血圧（SBP）のように本質的に相関するものだったら片方だけを説明変数に使うのが１つの対処法である。除かずに調整する方法としては，centring という方法がある。リッジ回帰（R では `MASS` ライブラリの `lm.ridge()`）によっても対処可能である。また，`DAAG` ライブラリ（Maindonald and Braun, 2003）の `vif()` 関数を使えば，自動的に VIF の計算をさせることができる[*41]。

[*41] 但し Armitage et al. が説明している方法と若干計算方法が異なり，結果も微妙に異なる。

例題

data(airquality) とすると，1973 年 5 月 1 日から 9 月 30 日まで 154 日間のニューヨーク市の大気環境データが使えるようになる。含まれている変数は，Ozone（ppb 単位でのオゾン濃度），Solar.R（セントラルパークでの 8:00 から 12:00 までの 4000 から 7700 オングストロームの周波数帯の太陽放射の強さを Langley 単位で表した値），Wind（LaGuardia 空港での 7:00 から 10:00 までの平均風速，マイル／時），Temp（華氏での日最高気温），Month（月），Day（日）である。
ニューヨーク市のオゾン濃度を，セントラルパークの日照，LaGuardia 空港の平均風速，日最高気温によって説明する重回帰モデルを，このデータに当てはめよ。

重回帰モデルの当てはめと，3 つの独立変数すべてについて Armitage らの方法で VIF の算出を行う R のプログラムは下枠内の通り。

```
> data(airquality)
> attach(airquality)
> res <- lm(Ozone ~ Solar.R+Wind+Temp)
> VIF <- function(X) { 1/(1-summary(X)$r.squared) }
> VIF(lm(Solar.R ~ Wind+Temp))
> VIF(lm(Wind ~ Solar.R+Temp))
> VIF(lm(Temp ~ Solar.R+Wind))
> summary(res)
Call:
lm(formula = Ozone ~ Solar.R + Wind + Temp)

Residuals:
    Min      1Q  Median      3Q     Max
-40.485 -14.219  -3.551  10.097  95.619

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -64.34208   23.05472  -2.791  0.00623 **
Solar.R       0.05982    0.02319   2.580  0.01124 *
Wind         -3.33359    0.65441  -5.094 1.52e-06 ***
Temp          1.65209    0.25353   6.516 2.42e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 21.18 on 107 degrees of freedom
Multiple R-Squared: 0.6059,     Adjusted R-squared: 0.5948
F-statistic: 54.83 on 3 and 107 DF,  p-value: < 2.2e-16
> detach(airquality)
```

3 つの独立変数の VIF はすべて 10 より遥かに小さく，多重共線性には問題ないと考えられる。summary(res) の結果は下枠内の通り得られるので，すべての係数が 5%水準でゼロと有意差があり，3 つの独立変数すべてがオゾン濃度に有意に影響しているといえる。また，Adjusted R-squared（自由度調整済み重相関係数の 2 乗）の値から，オゾン濃度のばらつきが，これら 3 つの独立変数のばらつきによって約 60%説明されることがわかる。

11.5 モデルの評価

モデルの当てはめで大事なのは，(1) どのモデルがよりよくデータを説明するのか？　(2) そのモデルはどの程度よくデータを説明しているのか？　を評価することである。以下，簡単にまとめてみる。

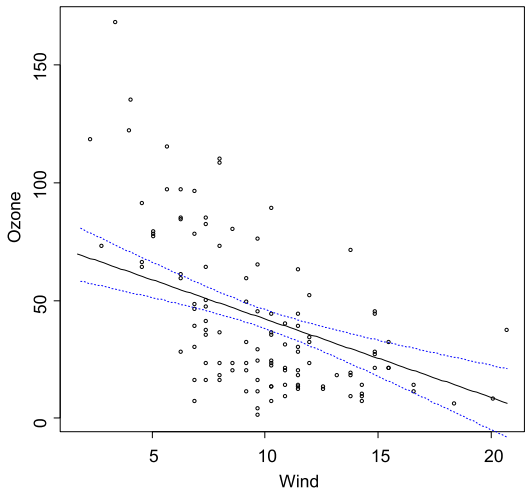
線型回帰モデルならば決定係数，すなわち自由度調整済み（重）相関係数の二乗が大きいモデルを採用するというのが１つの考え方である。しかし，この基準はかなりナイーブである。一般に，モデルの採否を決定するための基準としてよく使われるのは，残差分析，尤度比検定，AIC である。

11.6 残差分析と信頼区間

残差分析を行うと，モデルがデータから系統的にずれていないかどうかを検討することができる。系統的なズレは，とくにモデルを予測や信頼区間の推定に用いる場合に大きな問題となる。回帰モデルの結果を `res` に付値しておけば，例えば，`Wind` の大小と残差の大小の間に関連があるかどうか見るためには，`plot(residuals(res)~res$model$Wind)` とすることで，回帰の結果から残差を取り出してプロットすることができる。横軸としてはすべての独立変数について試してみるべきである。横軸の大小によらず，縦軸のゼロの近辺の狭い範囲にプロットが集中していれば，残差に一定の傾向がないことになり，系統的なズレはなさそうだと判断できる。なお，横軸の変数を指定せずに，`plot(residuals(res))` したときの横軸は，オブザーベーションの出現順を意味するインデックス値になる。

残差分析の裏返しのようなイメージになるが，信頼区間の推定も有用である。線型モデルであれば，信頼区間の推定には `predict()` 関数を用いることができる。例えば `Wind` のとる範囲に対して 95%信頼区間を得るためには，他の２つの変数が平均値で固定されていると仮定して，下枠内のプログラムを用いれば，`Wind` を横軸に，`Ozone` を縦軸にしたデータそのものがプロットされた上で，重回帰モデルによる推定値が実線で，その 95%信頼区間が青い点線で重ね描きされる[*42]。

```
> data(airquality)
> attach(airquality)
> res <- lm(Ozone ~ Solar.R+Wind+Temp)
> EW <- seq(min(Wind),max(Wind),len=100)
> ES <- rep(mean(Solar.R,na.rm=T),100)
> ET <- rep(mean(Temp,na.rm=T),100)
> Ozone.EWC <- predict(res,list(Wind=EW,Solar.R=ES,Temp=ET),interval="conf")
> plot(Ozone~Wind)
> lines(EW,Ozone.EWC[,1],lty=1)
> lines(EW,Ozone.EWC[,2],lty=2,col="blue")
> lines(EW,Ozone.EWC[,3],lty=2,col="blue")
> detach(airquality)
```



[*42] ちなみに，青い点線をデータ点の 95%が含まれるような範囲（95%予測区間）にするには，`predict()` のオプションで，`interval="conf"`となっているところを `interval="pred"`に変えればよい。

11.7 尤度比検定

次に，モデルの相対的な尤もらしさを考えよう。重回帰分析で独立変数が 3 つの場合とそのうち 1 つを除いた 2 つの場合，あるいは 3 次回帰と 2 次回帰のように，一方が他方を一般化した形になっている場合は，これら 2 つのモデルを比較することができる。

一般に，$f(x,\theta)$ で与えられる確率密度関数からの観測値を $\{x_1, x_2, ..., x_n\}$ とするとき，θ の関数として，$L(\theta) = f(x_1,\theta)f(x_2,\theta)\cdots f(x_n,\theta)$ を考えると，確率密度関数の値が大きいところほど観測されやすいため，$L(\theta)$ の値を最大にするような θ を真の θ の推定値とみなすのが最も尤もらしい。この意味で $L(\theta)$ を尤度関数と呼び，この θ のような推定量のことを最尤推定量と呼ぶ。尤度関数を最大にすることはその対数をとったもの（対数尤度）を最大にすることと同値なので，対数尤度を θ で偏微分した式の値をゼロにするような θ の中から $\ln L(\theta)$ を最大にするものが，最尤推定量となる。例えば，正規分布に従うサンプルデータについて得られる尤度関数を母平均 μ で偏微分したものをゼロとおいた「最尤方程式」を解けば，母平均の最尤推定量が標本平均であることがわかる。詳しくは鈴木 (1995) を参照されたい。

一般に，より一般性の低いモデルをデータに当てはめたときの最大尤度を，より一般的なモデルの最大尤度で割った値の自然対数をとって-2 を掛けた値 λ は，「尤度に差がない」という帰無仮説の下で，自由度 1（比較するモデル間のパラメータ数の差）のカイ二乗分布に従うので，検定ができる。この検定を尤度比検定と呼ぶ。R では，`logLik()` が対数尤度とパラメータ数を計算する関数なので，この関数を使えばよい。

例題

先の例題と同じデータで，独立変数が日照，風速，気温すべてであるモデルと，独立変数が日照と風速だけのモデルを尤度比検定せよ。

下枠内のように入力すれば，尤度比検定した有意確率は 10^{-9} のオーダーなので，有意水準 5%で帰無仮説は棄却される。したがってこの場合は 2 変数よりも 3 変数のモデルを採用すべきである。

```
> data(airquality)
> attach(airquality)
> res.3 <- lm(Ozone ~ Solar.R+Wind+Temp)
> res.2 <- lm(Ozone ~ Solar.R+Wind)
> lambda <- -2*(logLik(res.2)-logLik(res.3))
> print(1-pchisq(lambda,1))
'log Lik.' 1.123134e-09 (df=4)
> detach(airquality)
```

この例題では線型重回帰の関数 `lm()` を扱ったが，この尤度比検定の考え方は，2 つのモデルが包含関係にありさえすれば，一般化線型モデル `glm()` でも非線型モデル `nls()` でも，同じように使える。

11.8 AIC: モデルの当てはまりの悪さの指標

さて一方，AIC はパラメータ数と最大尤度からモデルの当てはまりの悪さを表すものとして計算される指標で，数式としては，L を最大尤度，n をパラメータ数として，

$$\mathrm{AIC} = -2\ln L + 2n$$

で表される。AIC が小さなモデルほど当てはまりがいいと考える。

実は，R には，`AIC()` という関数と `extractAIC()` という 2 つの関数がある。前者は “Akaike’s An Information Criterion” となっていて，後者は “The (generalized) Akaike *A*n *I*nformation *C*riterion for a fitted parametric model” となっている。前者が以前からある汎用関数である。`extractAIC()` は MASS ライブラリに含まれていたのが S4 メソッドとして標準実装されるようになった関数で，変数選択のために `step()` 関数の中から呼び出されるのが主な用途で

ある。

例えば，上の例題の 2 つのモデルについて AIC を計算すると，AIC(res.3) は，確かに

```
-2*logLik(res.3)+2*attr(logLik(res.3),"df")
```

と同じで 998.7 となり，extractAIC(res.3) の結果は 681.7 となる。res.2 についても同様で，AIC(res.2) は 1033.8，extractAIC(res.2) は 716.8 を返す。実は，定義通りの AIC を返すのは AIC() 関数なのだが，変数選択に使うためならそれと定数の差があってもいいので，計算量が少ない extractAIC() 関数が step() では使われているということである[*43]。

11.9 変数選択

このように，重回帰モデルの独立変数の取捨選択を行うことを変数選択と呼ぶ。非線型モデルの場合は自動的にはできないので，残差分析や尤度比検定や AIC の結果を見ながら手作業でモデリングを進めていくしかないが[*44]，線型重回帰モデルならば，step(lm(Ozone~Wind+Solar.R+Temp)) のように step() 関数を使って，自動的に変数選択を行わせることができる。変数増加法 (direction="forward")，変数減少法 (direction="backward")，変数増減法 (direction="both") などがある。例えば減少法の場合，direction="backward"オプションをつけるが，変数選択候補範囲を明示的に与えない場合，step() 関数のデフォルトは減少法になっているので，線型重回帰分析の結果を step() に渡す場合には，direction 指定はしなくても同じ結果になる。

このデータの場合，ress <- step(lm(Ozone~Solar.R+Wind+Temp)) とすると，3 つすべての変数が残った場合の AIC である 682 が最小であることがわかり，採択されたモデルが ress に保存される。ここで表示された AIC は step() 関数が，内部的に extractAIC() 関数を使って得た値なので，通常の AIC を表示するには，採択されたモデルに対して AIC(ress) としなくてはならない。lm() で使われたオブザーベーションが 111 しかないので（Ozone と Solar.R に欠損値が多いため），AIC(ress)-111*(1+log(2*pi))-2 とすると，確かに 681.7127 という結果になり，step() 関数の出力に出てくる値と一致することがわかる。まとめると，変数減少法で変数選択をさせ，最終的に採択されたモデルについての情報を表示させるには，下枠内のように入力すればよい。

```
> data(airquality)
> attach(airquality)
> res <- lm(Ozone~Solar.R+Wind+Temp)
> ress <- step(res)
> summary(ress)
> AIC(ress)
```

重回帰分析では，たくさんの独立変数の候補から比較的少数の独立変数を選択することが良く行われるが，モデル全体で評価するという観点からは，あまり薦められない。数値以外の根拠により投入する変数を決めて，各々の偏回帰係数（または偏相関係数）が有意であるかないかを見る方が筋がよい。十分な理由があれば，有意でない変数も含めた重回帰式を作っても良い。

しかし，数値以外の根拠が薄い場合もあるし，偏回帰係数が有意でない（偏相関係数がゼロであるという帰無仮説が成り立つ確率が 5%より大きい）変数を重回帰モデルに含めることを嫌う立場もある。従って，数値から最適なモデルを求める必要もありうる。そのためには，独立変数が 1 個の場合，2 個の場合，3 個の場合，……，のそれぞれについてすべての組み合わせの重回帰モデルを試して，最も重相関係数の二乗が大きなモデルを求めて，独立変数が n 個の場合が，$n-1$ 個の

[*43] http://www.is.titech.ac.jp/~shimo/class/gakubu200409.html（東工大・下平英寿さんの講義「R による多変量解析入門」の第 8 回「モデル選択」の資料）に，それぞれが使っている式の説明があり，AIC() 関数は $-2\ln L+2\theta$（L は最大尤度，θ はパラメータベクトルの次元）を計算する汎用関数であって，オブザーベーション数 n，パラメータ数 p，標準偏差 σ として，線型重回帰の場合は $n(1+\ln(2\pi\sigma^2))+2(p+1)$ を計算し（正規分布を仮定するから），extractAIC() 関数は線型重回帰のときだけ使える関数で，$n\ln(\sigma^2)+2p$ を計算する。前者から後者を引けば $n(1+\ln(2\pi))+2$ と，オブザーベーション数は含むけれどもパラメータ数には依存しない定数になるので，変数選択はこちらでやっても問題ないことになる。

[*44] R-help メーリングリストによると，S-plus には step.glm() という関数があるらしいが，R では敢えて実装しなかったらしい

場合のすべての変数を含むならば尤度比検定を行って，尤度が有意に大きくならないところまでの $n-1$ 個を独立変数として採用するのが良い。これを総当り法と呼ぶ。M.G. ケンドール著（奥野忠一，大橋靖雄訳）『多変量解析』（培風館，1981）では総当り法が薦められているが，R の step() 関数では提供されていない[*45]。

11.10 採択されたモデルを使った予測

モデルの当てはめがうまくできれば，独立変数群の値から従属変数の値を予測することができる。信頼区間の計算で示したように，predict() 関数を使えばよい。例えば，風速も日照も気温も観測値の平均値になった日に，オゾン濃度がいくらになるかを予測するには次のようにする。

```
> data(airquality)
> attach(airquality)
> res<-lm(Ozone~Solar.R+Wind+Temp)
> predict(res, list(Solar.R=mean(res$model$Solar.R),
+  Wind=mean(res$model$Wind), Temp=mean(res$model$Temp)))
[1] 42.0991
> detach(airquality)
```

他の観測値がわかっていて，オゾン濃度だけを測れなかった日の値を推定する（補間することになる）のにも，同じ方法が使える。ただし，回帰の外挿には慎重でなければならない。こうして推定された回帰係数を用いて，Solar.R と Temp がそれぞれこの重回帰分析で使われた値の平均値（なお，重回帰分析で使われた値だけでなく，できるだけ多くの値を使いたい場合なら，Solar.R には欠損値が含まれているので，平均を計算するときに，mean(Solar.R,na.rm=T) としなくてはならない）で，Wind=25 のときのオゾン濃度を点推定すると約-8.1 となってしまって，やはり採用できない（95%信頼区間はゼロを跨いでいるが）。結局，いくら AIC が小さくなっても，論理的に問題がある線型回帰を適用して予測をしてはいけないということである。

そこで登場するのが非線型回帰である。Wind と Ozone が負の相関関係があるので Wind が大きくなると Ozone がマイナスになるという線型回帰の弱点を避けるために，例えば Wind と Solar.R の 2 パラメータで，風速が係数が負の指数関数の形でオゾン濃度に影響する非線型関係を仮定したモデルで回帰分析を行うには，下枠内のようにする。

```
> data(airquality)
> attach(airquality)
> resmr <- nls(Ozone ~ a*exp(-b*Wind) + c*Solar.R, start=list(a=200,b=0.2,c=1))
> summary(resmr)
Formula: Ozone ~ a * exp(-b * Wind) + c * Solar.R

Parameters:
   Estimate Std. Error t value Pr(>|t|)
a 215.42457   33.11390   6.506 2.49e-09 ***
b   0.24432    0.03331   7.335 4.32e-11 ***
c   0.08639    0.02014   4.290 3.90e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22.01 on 108 degrees of freedom
> AIC(resmr)
[1] 1006.24
```

[*45] http://aoki2.si.gunma-u.ac.jp/R/All_possible_subset_selection.html に，群馬大学社会情報学部の青木繁伸教授が開発された R コードが公開されている。

AIC は線型の 2 パラメータモデルより小さい（extractAIC() 関数は，非線型モデルには使えない)。独立変数が 1 つだけの場合に比べるとずっと小さい（もっとも，十分に小さいとはいえないので，このデータに含まれていない，他の要因の影響が大きいのであろう)。Temp も入れたモデルと尤度比検定をすると有意ではないので，このモデルが採用できる。そこで続けて下枠内を入力すれば,

```
> SRM <- mean(subset(Solar.R,!is.na(Ozone)&!is.na(Solar.R)&!is.na(Wind)&!is.na(Temp)))
> predict(resmr,list(Wind=25,Solar.R=SRM))
```

約 16.4 となるので，風速 25 マイル／時のときのオゾン濃度は，太陽放射が平均的な条件なら，約 16.4 ppb になると予測される。

11.11 共分散分析

共分散分析は，典型的には，$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2 + \varepsilon$ というモデルになる。2 値変数 X_1 によって示される 2 群間で，量的変数 Y の平均値に差があるかどうかを比べるのだが，Y が量的変数 X_2 と相関がある場合に（このとき X_2 を共変量と呼ぶ)，X_2 と Y の回帰直線の傾き (slope) が X_1 の示す 2 群間で差がないときに，X_2 による影響を調整した Y の修正平均（adjusted mean; 調整平均ともいう）に，X_1 の 2 群間で差があるかどうかを検定する。

R では，X_1 を示す変数名を C（注：C は factor である必要がある)，X_2 を示す変数名を X とし，Y を示す変数名を Y とすると，summary(lm(Y~C+X)) とすれば，X の影響を調整した上で，C 間で Y の修正平均（調整平均）が等しいという帰無仮説についての検定結果が得られる（C のカテゴリが 1 と 2 である場合，C2 と表示される行の右端に出ているのがその有意確率である)。ただし，この検定をする前に，2 本の回帰直線がともに有意にデータに適合していて，かつ 2 本の回帰直線の間で傾き (slope) が等しいかどうかを検定して，傾きが等しいことを確かめておかないと，修正平均の比較には意味がない。そこで，まず例えば，summary(lm(Y[C==1]~X[C==1]); summary(lm(Y[C==2]~X[C==2]) として 2 つの回帰直線それぞれの適合を確かめ，summary(lm(Y~C+X+C:X))（または summary(lm(Y~C*X))）として傾きが等しいかどうかを確かめなければならない。傾きが有意に違っていることは，C と X の交互作用項が有意に Y に効いていることと同値なので，Coefficients の C2:X と書かれている行の右端を見れば，「傾きに差がない」という帰無仮説の検定の有意確率が得られる。そもそも回帰直線の適合が悪ければその独立変数は共変量として考慮する必要がないし，傾きが違っていれば群分け変数と独立変数の交互作用が従属変数に関して有意に影響しているということなので，2 群を層別して別々に解釈する方が良い。

例題

R の組み込みデータ ToothGrowth は，各群 10 匹ずつのモルモットに 3 段階の用量のビタミン C をアスコルビン酸としてあるいはオレンジジュースとして投与したときの象牙芽細胞（歯）の長さを比較するデータである。変数 len が長さ，supp が投与方法，dose が用量を示す。用量と長さの関係が投与方法によって異なるかどうかを共分散分析を使って調べよう。

例によってデータを使えるようにしてから，まずグラフを描いてみる。共分散分析をするような場面では，通常，下枠内のように，群によってマークを変えて散布図を重ね描きし，さらに線種を変えて群ごとの回帰直線を重ね描きするのだが，coplot(len~dose | supp) として横に 2 枚のグラフが並べて描かれるようにすることも可能である。

```
> data(ToothGrowth)
> attach(ToothGrowth)
> plot(dose,len,pch=as.integer(supp),ylim=c(0,35))
> legend(max(dose)-0.5,min(len)+1,levels(supp),pch=c(1,2))
> abline(lm1 <- lm(len[supp=='VC']~dose[supp=='VC']))
> abline(lm2 <- lm(len[supp=='OJ']~dose[supp=='OJ']),lty=2)
> summary(lm1)
> summary(lm2)
```

summary(lm1) と summary(lm2) をみると，投与方法別の回帰係数がゼロと有意差があることがわかる。そこで次に，これらの回帰係数間に有意差がないという帰無仮説を検定する。モデルの右辺に独立変数間の交互作用項を含めればいい。

```
> lm3 <- lm(len ~ supp*dose)
> summary(lm3)
Call:
lm(formula = len ~ supp * dose)

Residuals:
     Min       1Q   Median       3Q      Max
-8.22643 -2.84625  0.05036  2.28929  7.93857

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   11.550      1.581   7.304 1.09e-09 ***
suppVC        -8.255      2.236  -3.691 0.000507 ***
dose           7.811      1.195   6.534 2.03e-08 ***
suppVC:dose    3.904      1.691   2.309 0.024631 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.083 on 56 degrees of freedom
Multiple R-Squared: 0.7296,     Adjusted R-squared: 0.7151
F-statistic: 50.36 on 3 and 56 DF,  p-value: 6.521e-16
```

この結果から，suppVC:dose の従属変数 len への効果（交互作用効果）がゼロという帰無仮説の検定の有意確率が 0.024631 となるので，有意水準 5%で帰無仮説は棄却される。従って，この場合は，**投与経路によって投与量と長さの関係の傾きが有意に異なるので，と提示した上で，**先に計算済みの，投与経路別の回帰分析の結果を解釈すればよい（修正平均の差の検定はしても意味がない）。

11.12 ロジスティック回帰分析

ロジスティック回帰分析は，従属変数（ロジスティック回帰分析では反応変数と呼ぶこともある）が 2 値変数であり，正規分布に従わないので glm() を使う。

思想としては，例えば疾病の有無を，複数のカテゴリ変数によって表される要因の有無で説明する（量的な変数によって表される交絡を調整しながらオッズ比を計算できるのが利点であり，医学統計ではもっともよく使われる手法の一つである）。

この問題は，疾病の有病割合を P とすると，$\ln(P/(1-P)) = b_0 + b_1X_1 + ...b_kX_k$ と定式化できる。X_1 が要因の有無を示す 2 値変数で，$X_2, ...X_k$ が交絡であるとき，$X_1 = 0$ の場合を $X_1 = 1$ の場合から引けば，

$$b1 = \ln(P_1/(1-P_1)) - \ln(P_0/(1-P_0)) = \ln(P_1 * (1-P_0)/(P_0 * (1-P_1)))$$

となるので，b_1 が他の変数の影響を調整したオッズ比の対数になる。対数オッズ比が正規分布するとすれば，オッズ比の95%信頼区間が

$$\exp(b_1 \pm 1.96 \times \mathrm{SE}(b_1))$$

として得られる。

例題として，`library(MASS)` にある `data(birthwt)` を使った実行例を示す。

Springfield の Baystate 医療センターの 189 の出生について，低体重出生とそのリスク因子の関連を調べるためのデータである。`str(birthwt)` とすると変数がわかる。

low　低体重出生の有無を示す 2 値変数（児の出生時体重 2.5 kg 未満が 1）
age　年齢
lwt　最終月経時体重（ポンド[a]）
race　人種（1＝白人，2＝黒人，3＝その他）
smoke　喫煙の有無（1＝あり）
ptl　非熟練労働経験数
ht　高血圧の既往（1＝あり）
ui　子宮神経過敏の有無（1＝あり）
ftv　妊娠の最初の 3 ヶ月の受診回数
bwt　児の出生時体重 (g)

[a] 略号 lb. で，1 lb. は 0.454 kg に当たる。

```
> require(MASS)
> data(birthwt)
> attach(birthwt)
> low <- factor(low)
> race <- factor(race, labels=c("white","black","other"))
> ptd <- factor(ptl>0)
> smoke <- (smoke>0)
> ht <- (ht>0)
> ui <- (ui>0)
> ftv <- factor(ftv)
> levels(ftv)[-(1:2)] <- "2+"
> bw <- data.frame(low,age,lwt,race,smoke,ptd,ht,ui,ftv)
> detach(birthwt)
> summary(res <- glm(low ~ ., family=binomial, data=bw))
> summary(res2 <- step(res))
```

変数選択後の結果をみると，`smokeTRUE` の係数（対数オッズ比）は 0.866582 で，その `SE` が 0.404469 である。したがって，最終的なモデルに含まれる他の変数（最終月経時体重，黒人，他の有色人種，非熟練労働経験あり，高血圧既往あり，子宮神経過敏あり）の影響を調整した喫煙の低体重出生への効果（オッズ比とその 95%信頼区間）は，下枠内によって得られる。なお，人種は 3 つのカテゴリがあるので，自動的にダミー変数化されて処理される。

```
> exp(0.866582)
[1] 2.378766
> exp(0.866582 - qnorm(0.975)*0.404469)
[1] 1.076616
> exp(0.866582 + qnorm(0.975)*0.404469)
[1] 5.255847
```

この結果から，喫煙者は非喫煙者に比べて約 2.38 倍（95%信頼区間は [1.08, 5.26]），低体重出生児をもちやすいということを示している（95%信頼区間の下限が 1 より大きいので，有意水準 5%で有意な影響があったといえる）。

12 生存時間解析

最後に，生物統計解析でよく用いられる，生存時間解析について，ライブラリ `survival` の利用例を示しておく。大橋・浜田 (1995) で説明に用いられている Gehan の白血病治療データは，R では `MASS` ライブラリに含まれているので，これら 2 つのライブラリを呼び出す必要がある。既にインストールされているライブラリを呼び出すには，`library()` または `require()` を用いる。

12.1 カプラン＝マイヤ推定

大橋・浜田 (1995) の p.60-61 にあるような，Gehan の白血病治療データでの 6-MP 投与群と対照群別々のカプラン＝マイヤ推定をするには，R では以下のようにする。

`Surv()` は期間データと打ち切りフラグから生存時間型のデータを構成する関数であり，生存時間解析の関数は，この型のデータを扱うことができる。`summary()` は詳しい出力が欲しいときにつける[*46]。

[*46] 生存時間解析の結果に限らず，多くの結果オブジェクトに使える。ただし，オブジェクトによっては summary メソッドを持っていない場合もあり，その場合は詳しい出力とはならない。

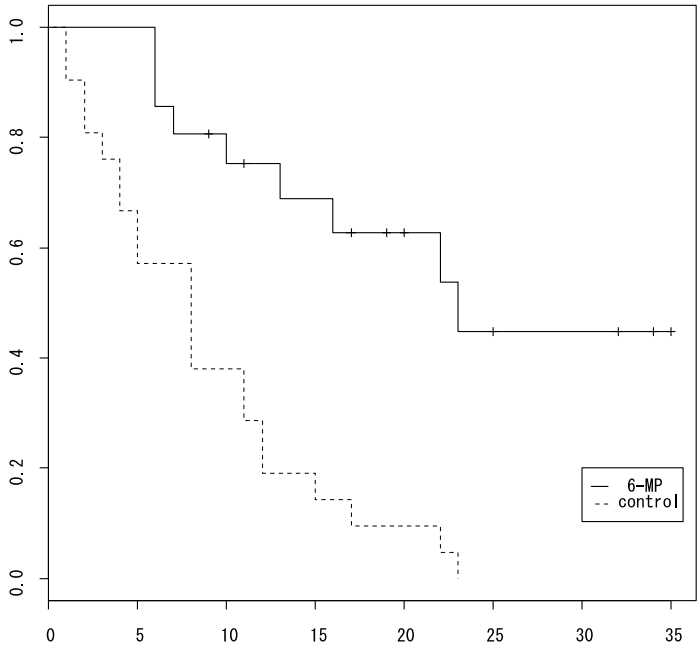
```
> require(MASS)
> require(survival)
> print(res<-survfit(Surv(time,cens)~treat,data=gehan))
Call: survfit(formula = Surv(time, cens) ~ treat, data = gehan)

               n events median 0.95LCL 0.95UCL
treat=6-MP    21      9     23      16     Inf
treat=control 21     21      8       4      12
> summary(res)
Call: survfit(formula = Surv(time, cens) ~ treat, data = gehan)

                treat=6-MP
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    6     21       3    0.857  0.0764        0.720        1.000
    7     17       1    0.807  0.0869        0.653        0.996
   10     15       1    0.753  0.0963        0.586        0.968
   13     12       1    0.690  0.1068        0.510        0.935
   16     11       1    0.627  0.1141        0.439        0.896
   22      7       1    0.538  0.1282        0.337        0.858
   23      6       1    0.448  0.1346        0.249        0.807

                treat=control
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1     21       2   0.9048  0.0641      0.78754        1.000
    2     19       2   0.8095  0.0857      0.65785        0.996
    3     17       1   0.7619  0.0929      0.59988        0.968
    4     16       2   0.6667  0.1029      0.49268        0.902
    5     14       2   0.5714  0.1080      0.39455        0.828
    8     12       4   0.3810  0.1060      0.22085        0.657
   11      8       2   0.2857  0.0986      0.14529        0.562
   12      6       2   0.1905  0.0857      0.07887        0.460
   15      4       1   0.1429  0.0764      0.05011        0.407
   17      3       1   0.0952  0.0641      0.02549        0.356
   22      2       1   0.0476  0.0465      0.00703        0.322
   23      1       1   0.0000      NA           NA           NA
> plot(res,lty=c(1,2))
> legend(30,0.2,lty=c(1,2),legend=levels(gehan$treat))
```

グラフィック出力ウィンドウに下図が得られる（本資料は LaTeX で作成しているが，LaTeX に画像を取り込むために，graphicx パッケージを使って eps ファイルを読み込んだ。美しい eps ファイルを作成するため，グラフィック出力ウィンドウから OpenOffice.org の Draw に画像をコピー＆ペーストしてサイズを決め，加工してからエクスポート機能で eps 出力を行っている）。



日時を扱う関数

生データとして生存時間が与えられず，観察開始とイベント発生の日付を示している場合，それらの間隔として生存時間を計算するには，`difftime()` 関数や `ISOdate()` 関数を使うと便利である。例えば，下枠内のように打てば，まず x というデータフレームに変数 `names`（名前），`dob`（誕生年月日）と `dod`（死亡年月日）が付値される。次に `difftime()` 関数で 4 人分の死亡年月日と誕生年月日の差（＝生存日数）が計算され，`[x$names=="Robert"]` で Robert（これは言うまでもなくロベルト・コッホのことである）についてだけの生存日数が得られ，それが `alivedays` に付値される。次の行のように 365.24 で割れば，生存年数に換算される。日数の与え方は，ダブルクォーテーションマークで括って，年，月，日がハイフンでつながれた形で与えることもできるし，最終行のように `ISOdate(`**年，月，日**`)` という形で与えることもできる。

```
> x <- data.frame(
+      names = c("Edward","Shibasaburo","Robert","Hideyo"),
+      dob = c("1749-5-17","1853-1-29","1843-12-11","1876-11-9"),
+      dod = c("1823-1-26","1931-6-13","1910-5-27","1928-5-21"))
> alivedays <- difftime(x$dod,x$dob)[x$names=="Robert"]
> alivedays/365.24
> difftime(ISOdate(2005,1,31),x$dob)
```

12.2 ログランク検定

8 匹のラットを 4 匹ずつ 2 群に分け，第 1 群には毒物 A を投与し，第 2 群には毒物 B を投与して，生存期間を追跡したときに，第 1 群のラットが 4,6,8,9 日目に死亡し，第 2 群のラットが 5,7,12,14 日目に死亡したとする。この場合，観察期間内にすべてのラットが死亡し，正確な生存時間がわかっているので，観察打ち切りがないデータとなっていて計算しやすい。

ログランク検定の思想は，大雑把に言えば，死亡イベントが起こったすべての時点で，群と生存／死亡個体数の 2 × 2 クロス集計表を作り，それをコクラン＝マンテル＝ヘンツェル流のやり方で併合するということである。

このラットの例では，死亡イベントが起こった時点 1〜8 において各群の期待死亡数を計算し，各群の実際の死亡数との差をとって，それに時点の重みを掛けたものを，各時点における各群のスコアとして，群ごとのスコアの合計を求める。2

群しかないので，各時点において群１と群２のスコアの絶対値は同じで符号が反対になる。２群の生存時間に差がないという帰無仮説を検定するためには，群１の合計スコアの２乗を分散で割った値をカイ二乗統計量とし，帰無仮説の下でこれが自由度１のカイ二乗分布に従うことを使って検定する。

なお，重みについては，ログランク検定ではすべて１である。一般化ウィルコクソン検定では，重みを，2 群を合わせたリスク集合の大きさとする（そうした場合，もし打ち切りがなければ，検定結果は，ウィルコクソンの順位和検定の結果と一致する)。つまり，ログランク検定でも一般化ウィルコクソン検定でも，実は期間の情報はまったく使われず，死亡順位の情報だけが使われているのである。

記号で書けば次の通りである。第 i 時点の第 j 群の期待死亡数 e_{ij} は，時点 i における死亡数の合計を d_i，時点 i における j 群のリスク集合の大きさを n_{ij}，時点 i における全体のリスク集合の大きさを n_i とすると，

$$e_{ij} = d_i \cdot n_{ij}/n_i$$

と表される[*47]。上の例では，$e_{11} = 1 \cdot n_{11}/n_1 = 4/8 = 0.5$ となる。時点 i における第 j 群の死亡数を d_{ij}，時点の重みを w_i と表せば，時点 i における群 j のスコア u_{ij} は，

$$u_{ij} = w_i(d_{ij} - e_{ij})$$

となり，ログランク検定の場合（以下，重みは省略してログランク検定の場合のみ示す）の群 1 の合計スコアは

$$u_1 = \sum_i (d_{i1} - e_{i1})$$

となる。上の例では，

$$u_1 = (1 - 4/8) + (0 - 3/7) + (1 - 3/6) + (0 - 2/5) + (1 - 2/4) + (1 - 1/3) + (0 - 0/2) + (0 - 0/1)$$

である。これを計算すると約 1.338 となる。分散は，分散共分散行列の対角成分を考えればいいので，

$$V = V_{jj} = \sum_i \frac{(n_i - n_{ij})n_{ij}d_i(n_i - d_i)}{{n_i}^2(n_i - 1)}$$

となる。この例の数値を当てはめると，

$$V = \frac{(8-4)\times 4}{8^2} + \frac{(7-3)\times 3}{7^2} + \frac{(6-3)\times 3}{6^2} + \frac{(5-2)\times 2}{5^2} + \frac{(4-2)\times 2}{4^2} + \frac{(3-1)\times 1}{3^2}$$

となり，`4*4/64+4*3/49+3*3/36+3*2/25+2*2/16+2*1/9` で計算すると，約 1.457 となる。したがって，$\chi^2 = 1.338^2/1.457 = 1.23$ となり，この値は自由度 1 のカイ二乗分布の 95%点である 3.84 よりずっと小さいので，有意水準 5%で帰無仮説は棄却されない。つまりこれだけのデータでは，差があるとはいえないことになる（もちろん，サンプルサイズを大きくすれば違う結果になる可能性もある)。

R でログランク検定を実行するには，観察時間を示す変数を `time`，打ち切りフラグを `event`，グループを `group` として，`survdiff(Surv(time,event)~group)` とすればよい。この例の場合なら，下枠内の通り。

```
> require(survival)
> time2 <- c(4,6,8,9,5,7,12,14)
> event <- c(1,1,1,1,1,1,1,1)
> group <- c(1,1,1,1,2,2,2,2)
> survdiff(Surv(time2,event)~group)
```

出力結果を見ると，$\chi^2 = 1.2$，自由度 1，$p = 0.268$ となっているので，有意水準 5%で，2 群には差がないことがわかる。なお，ログランク検定だけするのではなく，カプラン＝マイヤ法により生存時間の中央値と生存曲線の図示もするのが普通である。

[*47] 打ち切りデータは，リスク集合の大きさが変わることを通してのみ計算に寄与する。打ち切り時点ではスコアは計算されないことに注意しよう。

12.3 コックス回帰—比例ハザードモデル—の考え方

カプラン＝マイヤ推定やログランク検定は，まったく母数の分布を仮定しない方法だった。コックス回帰は，「比例ハザード性」を仮定する。そのため，比例ハザードモデルとも呼ばれる。

コックス回帰の基本的な考え方は，イベント発生に影響する共変量ベクトル $z_i = (z_{i1}, z_{i2}, ..., z_{ip})$ をもつ個体 i の，時点 t における瞬間イベント発生率 $h(z_i, t)$（これをハザード関数と呼ぶ）として，

$$h(z_i, t) = h_0(t) \cdot \exp(\beta_1 z_{i1} + \beta_2 z_{i2} + ... + \beta_p z_{ip})$$

を想定するものである。$h_0(t)$ は基準ハザード関数と呼ばれ，すべての共変量のイベント発生への影響がゼロである「基準人」の，時点 t における瞬間死亡率を意味する。$\beta_1, \beta_2, ..., \beta_p$ が推定すべき未知パラメータであり，共変量が $\exp(\beta_x z_{ix})$ という比例定数の形でイベント発生に影響するので，このことを「比例ハザード性」と呼ぶ。なお，Cox が立てたオリジナルのモデルでは，z_i が時間とともに変わる，時間依存性共変量の場合も考慮されていたが，現在，通常行われるコックス回帰では，**共変量の影響は時間に依存しないもの**（時間が経過しても増えたり減ったりせず一定）として扱う。

そのため，個体間のハザード比は時点によらず一定になるという特徴をもつ。つまり，個体 1 と個体 2 で時点 t のハザードの比をとると基準ハザード関数 $h_0(t)$ が分母分子からキャンセルされるので，ハザード比は常に，

$$\frac{\exp(\beta_1 z_{11} + \beta_2 z_{12} + ... + \beta_p z_{1p})}{\exp(\beta_1 z_{21} + \beta_2 z_{22} + ... + \beta_p z_{2p})}$$

となる。このため，比例ハザード性を仮定できれば，基準ハザード関数の形について（つまり，生存時間分布について）特定のパラメトリックモデルを仮定する必要がなくなる。この意味で，比例ハザードモデルはセミパラメトリックであるといわれる。

ここで生存関数とハザード関数の関係について整理しておこう。まず，T をイベント発生までの時間を表す非負の確率変数とする。生存関数 $S(t)$ は，$T \geq t$ となる確率である。$S(0) = 1$ となることは定義より自明である。ハザード関数 $h(t)$ は，ある瞬間 t にイベントが発生する確率なので，

$$h(t) = \lim_{\Delta t \to 0} \frac{\Pr(t \leq T < t + \Delta t | T \geq t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{S(t) - S(t + \Delta t)}{\Delta t S(t)} = -\frac{dS(t)}{dt} \frac{1}{S(t)} = -\frac{d(\log(S(t))}{dt}$$

である。累積ハザード関数は，$H(t) = \int_0^t h(u)du = -\log S(t)$ となる。これを式変形すると，$S(t) = \exp(-H(t))$ とも書ける。

そこで，共変量ベクトルが z である個体の生存関数を $S(z, t)$，累積ハザード関数を $H(z, t)$ とすれば，

$$H(z, t) = \int_0^t h(z, u)du = \int_0^t h_0(u) \exp(\beta z)du = \exp(\beta z) H_0(t)$$
$$S(z, t) = \exp(-H(z, t)) = \exp\{-\exp(\beta z) H_0(t)\}$$

となる。したがって，比例ハザード性が成立していれば，

$$\log(-\log S(z, t)) = \beta z + \log H_0(t)$$

が成り立つことになるので，共変量で層別して，横軸に生存期間の対数をとり，縦軸に生存関数の対数の符号を逆にしてもう一度対数をとった値をとって散布図を描くと，層間で βz だけ平行移動したグラフが描かれることになる。これを**二重対数プロット**と呼ぶ。

12.4 コックス回帰のパラメータ推定

パラメータ β の推定には，部分尤度という考え方が用いられる。時点 t において個体 i にイベントが発生する確率を，時点 t においてイベントが 1 件起こる確率と，時点 t でイベントが起きたという条件付きでそれが個体 i である確率の積に分

解すると，前者は生存時間分布についてパラメトリックなモデルを仮定しないと不明だが，後者はその時点でのリスク集合内の個体のハザードの総和を分母として，個体 i のハザードを分子として推定できる。すべてのイベント発生について，後者の確率だけをかけあわせた結果を L とおくと，L は，全体の尤度から時点に関する尤度を除いたものになり，その意味で部分尤度とか偏尤度と呼ばれる。

サンプルサイズを大きくすると真の値に収束し，分布が正規分布で近似でき，分散もその推定量としては最小になるという意味での，「良い」推定量として，パラメータ β を推定するには，この部分尤度 L を最大にするようなパラメータを得ればよいことを Cox が予想したので（後にマルチンゲール理論によって証明された），比例ハザードモデルをコックス回帰という。なお，同時に発生したイベントが 2 つ以上ある場合は，その扱い方によって，Exact 法とか，Breslow の方法，Efron の方法，離散法などがあるが，可能な場合は Exact 法を常に使うべきである（なお，離散法は，離散ロジスティックモデルに対応する推定法となっていて，生存時間が連続量でなく，離散的にしか得られていない場合に適切である）。Breslow 法を使うパッケージが多いが，R の `coxph()` 関数のデフォルトは Efron 法である。Breslow 法よりも Efron 法の方が Exact 法に近い結果となる。

群分け変数も共変量となりうるので，生存時間を表す変数を `time`，打ち切りフラグを `event`，グループを `group` として，`coxph(Surv(time,event)~group)` とすれば，群間のハザード比が推定でき，それがゼロと差がないという帰無仮説が検定できる。イベント発生時間が同じ個体が 2 つ以上あるときの扱い方として Exact 法を用いるには，`coxph(Surv(time,event)~group, method="exact")` とすればよい。

Gehan の白血病治療データで対照群に対する 6-MP 処置群のハザード比を推定するには以下のようにする。

```
> require(MASS)
> require(survival)
> res <- coxph(Surv(time,cens)~treat,data=gehan)
> summary(res)
Call:
coxph(formula = Surv(time, cens) ~ treat, data = gehan)

  n= 42
             coef exp(coef) se(coef)    z       p
treatcontrol 1.57      4.82    0.412 3.81 0.00014

             exp(coef) exp(-coef) lower .95 upper .95
treatcontrol      4.82      0.208      2.15      10.8

Rsquare= 0.322   (max possible= 0.988 )
Likelihood ratio test= 16.4  on 1 df,   p=5.26e-05
Wald test            = 14.5  on 1 df,   p=0.000138
Score (logrank) test = 17.3  on 1 df,   p=3.28e-05
> plot(survfit(res))
```

どの検定結果をみても有意水準 5%で「6-MP 処置が死亡ハザードに与えた効果がない」という帰無仮説は棄却される。`exp(coef)` の値 4.82 が，2 群間のハザード比の推定値になるので，6-MP 処置群に比べて対照群では 4.82 倍（95%信頼区間が [2.15, 10.8]）死亡ハザードが高いと考えられ，6-MP 処置は有意な延命効果をもつと解釈できる。

最後の行の `plot()` 関数により，2 群を併せてコックス回帰を当てはめた生存曲線が，95%信頼区間付きでプロットされる[*48]。

[*48] コックス回帰の場合は，通常，群の違いは比例ハザード性を前提として 1 つのパラメータに集約させ，生存関数の推定には 2 つの群の情報を両方用いる。2 群の生存曲線を別々に描きたい場合は，`coxph()` 関数の中で，`subset=(x=="6-MP")` のように指定することによって，群ごとにパラメータ推定をさせる必要がある。ただし信頼区間まで重ね描きされると見にくい。

12.5 コックス回帰における共変量の扱い

コックス回帰で，共変量の影響をコントロールできることの意味をもう少し説明しておく。例えば，がんの生存時間を分析するとき，進行度のステージ別の影響は無視できないけれども，これを調整するには，大別して３つの戦略がありうる。

1. ステージごとに別々に分析する。
2. 他の共変量の影響はステージを通じて共通として，ステージを層別因子として分析する
3. ステージも共変量としてモデルに取り込む

3 番目の仮定ができれば，ステージも共変量としてイベント発生への影響を定量的に評価できるメリットがあるが，そのためには，ステージが違ってもベースラインハザード関数が同じでなければならず，やや非現実的である。また，ステージをどのように共変量としてコード化するかによって結果が変わってくる（通常はダミー変数化することが多い)。2 番目の仮定は，ステージによってベースラインハザード関数が異なることを意味する。R の `coxph()` 関数で，層によって異なるベースラインハザードを想定したい場合は，`strata()` を使ってモデルを指定する。例えば，この場合のように，がんの生存時間データで，生存時間の変数が `time`，打ち切りフラグが `event`，治療方法を示す群分け変数が `treat`，がんの進行度を表す変数が `stage` であるとき，進行度によってベースラインハザード関数が異なることを想定して，治療方法によって生存時間に差が出るかどうかコックス回帰で調べたければ，`coxph(Surv(time,event)~treat+strata(stage))` とすればよい。

なお，コックス回帰はモデルの当てはめなので，一般化線型モデルで説明したのと同様，残差分析や尤度比検定，重相関係数の 2 乗などを用いて，よりよいモデル選択をすることができる。ただし，基準ハザード関数の型に特定の仮定を置かないと AIC は計算できない。また，コックス回帰のパラメータ推定で，同時に発生したイベントが２つ以上ある場合の扱い方は，Breslow 法を採用しているパッケージが多いが，R のデフォルトは，より Exact 法に近いと言われている Efron 法である。

12.6 その他の技法

加速モデルは `survreg()` 関数で実行可能である。この他にも R には数多くの関数やライブラリが存在するので，前述の RjpWiki からリンクを辿って探せば，大抵のデータ解析はできるだろう。

13 参考文献

1. 大橋靖雄，浜田知久馬 (1995) 『生存時間解析　SAS による生物統計』（東京大学出版会）
2. 中澤　港 (2003)『R による統計解析の基礎』（ピアソン・エデュケーション）
3. 間瀬 茂・神保 雅一・鎌倉 稔成・金藤 浩司 (2004)『工学のための数学３　工学のための データサイエンス入門 ― フリーな統計環境 R を用いたデータ解析 ― 』（数理工学社）
4. 岡田昌史（編） (2004)『The R Book －データ解析環境 R の活用事例集－』（九天社）
5. 舟尾暢男 (2005)『The R Tips －データ解析環境 R の基本技・グラフィック活用集』（九天社）
6. 鈴木義一郎 (1995) 『情報量基準による統計解析』（講談社サイエンティフィク）
7. Armitage P, Berry G, Matthews JNS (2002) 『Statistical Methods in Medical Research, 4th ed.』(Blackwell Publishing)
8. Maindonald J, Braun J (2003) 『Data analysis and graphics using R』(Cambridge Univ. Press)