

リトルブック：Rによる生物医学統計

A Little Book of R For Biomedical Statistics, Release 0.1

2011年8月29日

Avril Coghlan

日本語訳 荒木 孝治

2012年2月18日

著者 : Avril Coghlan (University College Cork, Cork, Ireland). Eメール : a.coghlan@ucc.ie

このブックレットは, R を用いた生物医学統計への簡単な入門書である*¹.

このブックレットの pdf 版は, https://github.com/avrilcoghlan/LittleBookofRBiomedicalStats/raw/master/_build/latex/BiomedicalStats.pdf より取得可能である.

このブックレットが気に入った人は, “R による多変量解析” (<http://little-book-of-r-for-multivariate-analysis.readthedocs.org/>) と “R によるに時系列解析” (<http://little-book-of-r-for-time-series.readthedocs.org/>) も参照してほしい.

*¹ 本翻訳に関するコメント・問い合わせ等は, 荒木孝治 (関西大学商学部, arakit@kansai-u.ac.jp) までお願いします.

目次

第 1 章	R のインストール方法	1
1.1	R とは	1
1.2	R のインストール	1
1.3	R のパッケージのインストール	3
1.4	R の起動	4
1.5	R 超入門	5
1.6	リンクと参考文献	8
1.7	謝辞	8
1.8	連絡	8
1.9	ライセンス	8
第 2 章	R による生物医学統計	9
2.1	生物医学統計	9
2.2	コホート研究における相対危険度の計算	9
2.3	コホートあるいは症例対照研究に対するオッズ比の計算	11
2.4	コホート研究または症例対照研究における曝露と疾病との関連の検定	13
2.5	層別変数がある場合の (Mantel-Haenszel) オッズ比の計算	13
2.6	対症例対照研究における曝露と疾病との関連の検定	16
2.7	用量反応分析	17
2.8	ランダム化比較試験に必要なサンプルサイズ	19
2.9	ランダム化比較試験の検出力の計算	20
2.10	複数のランダム化比較試験のメタ分析に対するフォレストプロットの作成	21
2.11	リンクと参考文献	23
2.12	謝辞	23
2.13	連絡	23
2.14	ライセンス	23
第 3 章	謝辞	24
第 4 章	連絡	25
第 5 章	ライセンス	26

第 1 章

R のインストール方法

1.1 R とは

本ブックレットは、生物医学統計で R を利用する方法に関する入門書である。

R (www.r-project.org) は、一般的に用いられるフリーの統計ソフトウェアである。R は、対話モードのみならず簡単なプログラミングにより統計分析を実行することができる。

1.2 R のインストール

R を使うには、R プログラムがコンピュータにインストールされている必要がある。

1.2.1 R が Windows パソコンにインストールされているかどうかを確認する方法

R をコンピュータにインストールする前にすべきことは、R が、例えば前のユーザーによって既にインストールされているかどうかを調べることである。

以下、主に Windows パソコンに関してその方法について説明するが、Macintosh または Linux コンピュータに関しても少し触れる。Windows パソコンの場合、R がコンピュータに既にインストールされているかどうか調べる方法には、次に示す 2 つがある。

1. “R” のアイコンがコンピュータのデスクトップにあるかどうかを調べる。もしあれば、“R” アイコンをダブルクリックすることにより、R を起動することができる。無いときは、手順 2 に行く。
2. Windows のデスクトップの左下にある“スタート”ボタンをクリックし、ポップアップしたメニューの [全てのプログラム] の上にマウスを移動させる。表示されるプログラムのリストに“R”があるかどうかを確認する。もしあれば、すでに R がインストールされているので、リストから“R” (例えば、R 2.10.0 (R X.X.X の X.X.X は R のバージョンを意味する)) を選択することによって R をスタートさせることができる。

上記の (1) か (2) により R を起動できたら、既にコンピュータに R がインストールされることを意味する (どちらもだめなときは、R はまだインストールされていない)。インストールされている R のバージョンが古い場合、最新版をインストールする方がよい。最新の R の機能を利用することができるからである。

1.2.2 R の最新版を知る方法

R の最新バージョンは、CRAN (The Comprehensive R Archive Network) <http://cran.r-project.org/> で確認することができる。

“The latest release” (最新のリリース) の他にも (ページの途中で)、“R-X.X.X.tar.gz” (例えば、

“R-2.12.1.tar.gz”) のような表示がある。これは、R の最新のバージョンが X.X.X (今の場合、2.12.1) であることを意味する。

R の開発は非常に活発なため、R の新しいバージョンのリリースは定期的に行われている (年におよそ 2 回)。定期的に R の新しいバージョンを確認し、それをインストールすることには価値がある (それにより、ダウンロードした R のパッケージのすべての最新版との互換性を確実にすることができる)。

1.2.3 Windows パソコンへの R のインストール

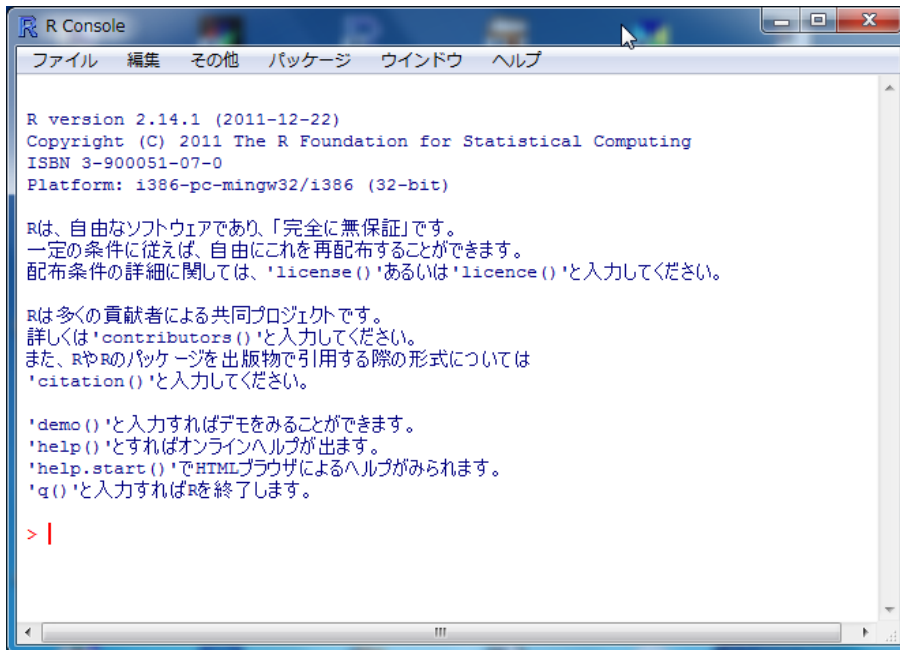
Windows パソコンに R をインストールするには、次の手順を実行する。

1. <http://essrc.hyogo-u.ac.jp/cran/> へ行く*1
2. “Download R for Windows” をクリック。
3. “Subdirectories” で “base” をクリック。
4. 次のページに、“Download R 2.12.1 for Windows” (R X.X.X. の “X.X.X” は R のバージョンを示す。例えば、R 2.12.1)*2といった表示がある。このリンクをクリック。
5. ファイル “R-2.12.1-win32.exe” をどう処理するか、つまり、保存するか開く (実行する) かを聞かれるので、“ファイルを保存する”を選択し、保存場所をデスクトップに指定する。保存後、ファイルをダブルクリックしてインストールを実行する。
6. インストール中に利用する言語を聞かれるので、英語 (English) を選択する*3。
7. R のセットアップウィザードが起動するので、下にある “Next” (“次へ”) ボタンをクリックする。
8. “Information” (“情報”) ページが開かれる。“Next” (“次へ”) をクリック。
9. “Select Destination Location” (“インストール先の指定”) ページが開かれる。デフォルトのインストール場所は “C:\Program Files” である。
10. セットアップウィザードの下にある “Next” (“次へ”) をクリックする。
11. “Select Components” (“コンポーネントの選択”) ページが開かれる。“Next” (“次へ”) をクリック。
12. “Startup options” ページが開かれる。“Next” (“次へ”) をクリック。
13. “Select Start Menu Folder” ページが開かれる。“Next” をクリック。
14. “Select Additional Tasks” (“追加タスクの選択”) ページが開かれる。“Next” (“次へ”) をクリック。
15. これで R がインストールされる。インストールには 1 分くらいかかる。終了すると “Completing the R for Windows Setup Wizard” (“セットアップウィザードの完了”) が表示される。“Finish” (“完了”) をクリックする。
16. R を起動するには、手順 18 または 19 を実行する。
17. “R” のアイコンがコンピュータのデスクトップに表示されているかどうかを確認する。表示されている場合、R を起動するには “R” をダブルクリックする。表示されていない場合、手順 19 を実行する。
18. コンピュータ・スクリーンの左下にある “スタート” ボタンをクリックし、“すべてのプログラム” を選択する。次にプログラムのメニューから “R” (R X.X.X の “X.X.X” は R のバージョンを示し、例えば、R 2.12.1) を選択することにより、R を起動する。
19. R コンソールが表示される。

*1 (訳注) 日本のミラーサイトに変更。原著では、<http://ftp.heanet.ie/mirrors/cran.r-project.org>。

*2 (訳注) 原著では、バージョンが様々なので、表記を “R 2.12.1” に統一した (以下、同様)。

*3 (訳注) Japanese (日本語) でのインストールを選択できるが、一部文字化けするので、英語でインストールする方がよい。



1.2.4 非 Windows オペレーティングシステム（例えば、Macintosh, または Linux）へのインストール

上記は、Windows パソコンへの R のインストール方法である。非 Windows オペレーティングシステム (OS) を利用するコンピュータ（例えば、Macintosh または Linux）の場合、<http://ftp.heanet.ie/mirrors/cran.r-project.org> から、OS に適切な R インストーラをダウンロードし、<http://ftp.heanet.ie/mirrors/cran.r-project> にある R のインストール方法に従う。

1.3 R のパッケージのインストール

R をインストールするとき、いくつかの標準パッケージも一緒にインストールされる。有用な他の R パッケージ（たとえば、“rmeta” パッケージ）を使う方法を本ブックレットでは説明する。これらの付加的なパッケージは R と一緒にインストールされないで、別にインストールする必要がある。

1.3.1 R のパッケージのインストール法

R を Windows パソコンに（上記のステップに従って）インストールした後、下記の手順でパッケージを追加的にインストールすることができる。

1. R を起動するために、次の手順 2 または 3 を実行する。
2. “R” のアイコンがデスクトップにあるかどうか調べる。あるなら、“R” アイコンをダブルクリックして R を起動する。ない場合、手順 3 を実行する。
3. コンピュータ・スクリーンの左下にある“スタート”ボタンをクリックし、“すべてのプログラム”から“R”（R X.X.X の X.X.X は R のバージョンで、例えば、R 2.12.1）を選択することにより、R を起動する。
4. R コンソールが表示される。
5. R を起動すると、R コンソールの上にある“パッケージのインストール”メニューより R のパッケージ（例えば、“rmeta”）をインストールすることができる。どのウェブサイトからダウンロードするかを聞かれるので、“Japan”（または他の国）を選択する。インストール可能なパッケージのリストが表示されるので、そのリストからインストールしたいパッケージ（例えば、

“rmeta”)を指定する.

6. これにより “rmeta” パッケージのインストールが始まる.
7. “rmeta” パッケージのインストールが終了する. この後, R を起動した後, R コンソールに次のコマンドを入力することにより “rmeta” パッケージをロードし, 利用することができる.

```
> library("rmeta")
```

Bioconductor (<http://www.bioconductor.org>) というバイオインフォマティックスのための R パッケージの特殊な集合がある (例えば, “yeastExpData” や “Biostrings” といったパッケージの集まり). Bioconductor パッケージ群は, Bioconductor に特有の手順 (Bioconductor の R パッケージをインストールする方法を参照) によりインストールする必要がある.

1.3.2 Bioconductor のパッケージのインストール法

1.3.1 項に示した手順で, R パッケージの大部分をインストールすることができる. しかし, バイオインフォマティックス用の R パッケージの集合である Bioconductor に関しては, 特別な手順に従う必要がある. Bioconductor (<http://www.bioconductor.org>) は, バイオインフォマティックスのために開発されたパッケージ群である.

1. R を起動するには, 次の手順 2 または 3 を実行する.
2. “R” のアイコンがデスクトップにあるかどうか調べる. あるなら, “R” アイコンをダブルクリックして R を起動する. “R” アイコンがない場合, 手順 3 を実行する.
3. コンピュータ・スクリーンの左下にある “スタート” ボタンをクリックし, “すべてのプログラム” から “R” (R X.X.X の X.X.X は R のバージョンで, 例えば, R 2.12.1) を選択することにより, R を起動する.
4. R コンソールが表示される.
5. R を起動した後, R コンソールに次を入力する.

```
> source("http://bioconductor.org/biocLite.R")
> biocLite()
```

6. これにより Bioconductor の主要パッケージ (“affy”, “affydata”, “affyPLM”, “annaffy”, “annotate”, “Biobase”, “Biostrings”, “DynDoc”, “gcrma”, “genefilter”, “genefilter”, “genefilter”, “hgu95av2.db”, “limma”, “marray”, “matchprobes”, “multtest”, “ROC”, “vsn”, “xtable”, “affyQCReport”) がインストールされる. これには少し時間がかかる.
7. 後日, Bioconductor の主要パッケージ以外のパッケージ, 例えば, “yeastExpData” をインストールする必要があるときは, R コンソールに次を入力する.

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("yeastExpData")
```

8. パッケージをインストールした後, それを利用するには R コンソールに次を入力する.

```
> library("yeastExpData")
```

1.4 R の起動

R を利用するには, 先ず R プログラムを起動する. それには, R がインストールされている必要がある. R を起動するには次の手順 1 または 2 を実行する.

1. “R” のアイコンがデスクトップにあるかどうか調べる. あるなら, “R” アイコンをダブルクリッ

くして R を起動する。“R” アイコンがない場合、手順 2 を実行する。

2. コンピュータ・スクリーンの左下にある“スタート”ボタンをクリックし、“すべてのプログラム”から“R”（R X.X.X の“X.X.X”は R のバージョンで、例えば、R 2.12.1）を選択することにより、R を起動する。

これにより、R コンソールという新しいウインドウが表示される。

1.5 R 超入門

R 内で分析を実行するには、R コンソールにコマンドを入力する。R コンソールには次に示す記号が表示されている。

```
>
```

この記号 (>) は R のプロンプトである。プロンプトの後ろに、特定の作業をするのに必要なコマンドを入力する。コマンドは、Return キーを入力後に実行される。R をいったん起動すると、コマンドを入力することにより R を利用でき、結果はすぐに表示される。例えば：

```
> 2*3
[1] 6
> 10-3
[1] 7
```

R が生成した全ての変数（スカラー、ベクトル、行列等）はオブジェクトと呼ばれる。R では、変数に値を与えるとき、矢印 (<-) を用いる。例えば、値 `2*3` を変数 x に与えるには次のようにする。

```
> x <- 2 * 3
```

R のオブジェクトの内容を見るには、その名前を入力するだけでよい。その内容が表示される。

```
> x
[1] 6
```

R には、スカラー、ベクトル、行列、配列、データフレーム、テーブル、リストといった様々な種類のオブジェクトがある。上記のスカラー変数 x は、R オブジェクトの 1 例である。スカラー変数はちょうど 1 つの要素を持つが、ベクトルはいくつかの要素から構成される。c() 関数 (c は combine の c) を用いて、ベクトルを作成することができる。例えば値 8, 6, 9, 10, 5 の要素から構成される、*myvector* という名前のベクトルを作成するには、次を入力する。

```
> myvector <- c(8, 6, 9, 10, 5)
```

変数 *myvector* の内容を表示するには、その名前を入力するだけでよい。

```
> myvector
[1] 8 6 9 10 5
```

[1] はベクトルの 1 番目の要素を示すインデックスである。ベクトルの任意の要素を抽出するには、角括弧 ([]) の中に要素のインデックスを与えたものをベクトルの名前の後ろに記載する。例えば、ベクトル *myvector* の 4 番目の要素の値を抽出するには、次のようにする。

```
> myvector[4]
[1] 10
```

リストは、ベクトルと異なり、数値と記号といった異なる型の要素を保持することができる。リストはまた、ベクトルなどの他の変数を含むこともできる。リストの作成には list() 関数を用いる。

例えば、*mylist* というリストを作るには、次のコマンドを入力する。


```
> mylist <- list(name="Fred", wife="Mary", myvector)
```

リスト *mylist* の内容を表示するには、その名前を入力する。

```
> mylist
$name
[1] "Fred"

$wife
[1] "Mary"

[[3]]
[1] 8 6 9 10 5
```

リストの要素には番号が付けられており、それをインデックスとして参照することができる。リストの要素を抽出するには、リスト名の後ろに、2重の角括弧 ([[]]) の中に要素のインデックスを与えたものを記載することにより可能である。だから、2番目と3番目の要素を *mylist* から抽出するには次のようにする。

```
> mylist[[2]]
[1] "Mary"
> mylist[[3]]
[1] 8 6 9 10 5
```

リストの要素に名前を付けることができる。この場合、名前の後ろに "\$" をつけ、続いて要素名を記載することでリストの要素を参照することができる。例えば、*mylist\$name* は *mylist[[1]]* と同じであり、*mylist\$wife* は *mylist[[2]]* と同じである。

```
> mylist$wife
[1] "Mary"
```

リスト中の名前が与えられた要素の名前を知るには、*attributes()* 関数を用いて次のようにする。

```
> attributes(mylist)
$names
[1] "name" "wife" ""
```

リスト変数を持つ名前付きの要素を知るために *attributes()* 関数を用いると、名前付き要素には常に、"\$names" というヘッダーが付けられる。よって、リスト変数 *mylist* の名前付き要素の名前が "name" と "wife" であることがわかる。*mylist\$name* や *mylist\$wife* と入力することによってそれらの値を切り出すことができる。

R で利用する別のオブジェクトとして、テーブル変数がある。例えば、学級内の生徒の名前を含む名前のベクトル変数 *mynames* があるとすると、*table()* 関数を利用して、*mynames* 内の同じ名前を持つ子供の数のテーブル変数を作るには次のようにする。

```
> mynames <- c("Mary", "John", "Ann", "Sinead", "Joe", "Mary", "Jim", "John", "Simon")
> table(mynames)
mynames
  Ann   Jim   Joe  John  Mary  Simon Sinead
  1     1     1     2     2     1     1
```

関数 *table()* によって作ったテーブル変数に、名前 "*mytable*" をつけて保存するには次を入力する。

```
> mytable <- table(mynames)
```

テーブル変数の要素にアクセスするには、リストの要素にアクセスするときと同様に、二重の角括弧 ([[]]) を使う。例えば、*mytable* の4番目の要素 ("John" という子供の数) にアクセスするには、

次を入力する.

```
> mytable[[4]]  
[1] 2
```

表の4番目の要素の名前 (“John”) を用いてテーブル要素の値を知ることができる.

```
> mytable[["John"]]  
[1] 2
```

R の関数は通常、引数を必要とする。引数は、関数に渡される入力変数 (オブジェクト) であり、それは、演算を実行するときに利用される。例えば、`log10()` 関数は数を渡され、その数の、底が 10 の対数の値を計算する。

```
> log10(100)  
[1] 2
```

R では、`help()` 関数を用いて特定の関数についてヘルプ画面を参照することができる。例えば、`log10()` 関数についてヘルプを参照したいとき、次を入力する。

```
> help("log10")
```

`help()` 関数を利用するとき、ウィンドウまたはウェブページが出現し、ヘルプを求める関数に関する情報が表示される。

関数の名前に確信がないが、その名前の一部がわかっているとき、`help.search()` と `RSiteSearch()` 関数を使って関数名を探すことができる。`help.search()` 関数は、興味があるトピックに関連する関数がすでにインストールされて (インストールされているパッケージのどれかに入って) いるかどうかを探す。`RSiteSearch()` 関数は、興味があるトピックに関連した関数を、すべての R 関数 (まだインストールされていないパッケージに含まれるものを含んで) の中から探す。

例えば、標準偏差を計算する関数があるかどうかを知りたいとき、次を入力することにより、関数に関する記述の中から “deviation” (偏差) という語を含むすべてのインストールされた関数を探すことができる。

```
> help.search("deviation")  
Help files with alias or concept or title matching  
'deviation' using fuzzy matching:  
  
genefilter::rowSds  
      Row variance and standard deviation of  
      a numeric array  
nlme::pooledSD  
      Extract Pooled Standard Deviation  
stats::mad  
      Median Absolute Deviation  
stats::sd  
      Standard Deviation  
vsn::meanSdPlot  
      Plot row standard deviations versus row
```

見つけた関数の中に、“stats” パッケージ (R のインストールとともにインストールされる基本パッケージ) の中に `sd()` 関数があり、これにより標準偏差を計算することができることがわかる。

上記の例では、`help.search()` 関数は関連した関数 (`sd()`) を見つけた。`help.search()` で期待していたものを見つけることができない場合、`RSiteSearch()` 関数を使うことにより、R のウェブサイトで記述されているすべての関数の中から、興味があるトピックに関連するものを見つけることができるかもしれない。

```
> RSiteSearch("deviation")
```

RSiteSearch() 関数の結果は、R 関数の記述への、ならびにそれらの関数に関する R メールングリスト議論へのヒット結果である。

R では、スカラーとベクトルといったオブジェクトを利用して計算することができる。例えば、ベクトル myvector の値の平均を計算する（すなわち、8, 6, 9, 10, 5 の平均）ために、mean() 関数を使うことができる。

```
> mean(myvector)
[1] 7.6
```

mean(), length(), print(), plot() といった R の組み込み関数を利用することができる。これに対し、よく利用する機能を持つ関数を独自に作成することもできる。例えば、入力された数の 2 乗に 20 を加えるための関数を作成するには次のようにする。

```
> myfunction <- function(x) { return(20 + (x*x)) }
```

return() 関数は、計算値を返す機能を持つ。この関数を入力した後、関数を使うことができる。例えば、異なる入力値（例えば、10, 25）に対してこの関数を使うことができる。

```
> myfunction(10)
[1] 120
> myfunction(25)
[1] 645
```

R を終了するには次を入力する。

```
> q()
```

1.6 リンクと参考文献

さらに学習するためのリンクを紹介する。

R へのより詳細に入門するための良いオンライン・チュートリアルが、“Kickstarting R” というウェブサイト (cran.rproject.org/doc/contrib/Lemon-kickstart) にある。

別の素晴らしい、もう少し詳細なチュートリアルが、“Introduction to R” というウェブサイト (cran.rproject.org/doc/manuals/R-intro.html) にある。

1.7 謝辞

Friedrich Leisch と Phil Spector には、R のインストールに記述に関して非常に有益なコメントと提案をいただいた。感謝する。

1.8 連絡

訂正や改良に関する提案があれば、著者 (Avril Coghlan) のメールアドレス (a.coghlan@ucc.ie) まで送付していただければ幸いである。

1.9 ライセンス

本書の内容は、Creative Commons Attribution 3.0 のもとで公開されている。

第2章

Rによる生物医学統計

2.1 生物医学統計

このブックレットは、生物医学統計で利用される簡単な手法をRを用いて実行する方法を示す。特に、特定の因子が疾病と関連しているかどうかを検定することを目的とするコホート研究と症例対照研究、ランダム化試験、メタ分析に焦点を置く。

読者には、生物医学統計の入門的な知識を持つことを前提とする。このブックレットの主目的は、生物医学統計自体を説明することにはなく、Rを用いてその手法を実践する方法を説明することにある。

生物医学統計を知らなかったり、このブックレットで説明する概念についてもっと知りたかったりするときは、Open University 刊行の本“Medical Statistics“（製品コード M249/01）を参照することを強く勧める（Open University のショップより購入可）。

このブックレットの pdf 版が、ウェブサイト https://github.com/avrilcoghlan/LittleBookofRBiomedicalStats/raw/master/_build/latex/BiomedicalStats.pdf にある。

このブックレットを気に入った人は、“Rを用いた時系列解析” (<http://a-little-book-of-r-for-time-series.readthedocs.org/>) や “Rを用いた多変量解析” (<http://little-book-of-r-for-multivariate-analysis.readthedocs.org/>) も参照してほしい。

2.2 コホート研究における相対危険度の計算

生物医学統計における最も一般的なデータセットは、コホート研究で用いるものである。これは、人がある処理または環境に曝露されている（例えば、ある薬を飲んだか、たばこを吸うかといったもの）かどうかという情報、および、同じ人が特定の疾病を持つかどうかという情報である。このデータセットは、次のような形をしている：

	疾病	対照
曝露	156	9421
非曝露	1531	14797

次のコマンドにより、Rにこのデータを読み込むことができる：

```
> mymatrix <- matrix(c(156,9421,1531,14797),nrow=2,byrow=TRUE)
> colnames(mymatrix) <- c("疾病","対照")
> rownames(mymatrix) <- c("曝露","非曝露")
> print(mymatrix)
      疾病  対照
曝露   156 9421
非曝露 1531 14797
```

曝露により発症する相対危険度（相対リスク）は、処理や環境因子に曝露された人が発症する確率を、その処理や環境因子に曝露されなかった人が発症する確率で割った値である。

曝露したときの発症の相対危険度を R で計算するには、関数 `calcRelativeRisk()` を利用することができる。この関数を利用するには、次のコードをコピーし、R に貼り付けておくだけでよい：

```
> calcRelativeRisk <- function(mymatrix,alpha=0.05,referencerow=2)
{
  numrow <- nrow(mymatrix)
  myrownames <- rownames(mymatrix)
  for (i in 1:numrow)
  {
    rowname <- myrownames[i]
    DiseaseUnexposed <- mymatrix[referencerow,1]
    ControlUnexposed <- mymatrix[referencerow,2]
    if (i != referencerow)
    {
      DiseaseExposed <- mymatrix[i,1]
      ControlExposed <- mymatrix[i,2]
      totExposed <- DiseaseExposed + ControlExposed
      totUnexposed <- DiseaseUnexposed + ControlUnexposed
      probDiseaseGivenExposed <- DiseaseExposed/totExposed
      probDiseaseGivenUnexposed <- DiseaseUnexposed/totUnexposed

      # 相対危険度の計算
      relativeRisk <- probDiseaseGivenExposed/probDiseaseGivenUnexposed
      print(paste("category =", rowname, ", relative risk = ",relativeRisk))

      # 信頼区間の計算
      confidenceLevel <- (1 - alpha)*100
      sigma <- sqrt((1/DiseaseExposed) - (1/totExposed) +
                    (1/DiseaseUnexposed) - (1/totUnexposed))
      # sigma は相対危険度の対数の推定値の標準誤差
      z <- qnorm(1-(alpha/2))
      lowervalue <- relativeRisk * exp(-z * sigma)
      uppervalue <- relativeRisk * exp( z * sigma)
      print(paste("category =", rowname, ", ", confidenceLevel,
                  "% 信頼区間 = [",lowervalue,",",uppervalue,"]"))
    }
  }
}
```

関数 `calcRelativeRisk()` を用いて、曝露による疾病の相対危険度、およびその信頼区間を求めることができる。例えば、99% 信頼区間を求めるには、次を入力する。

```
> calcRelativeRisk(mymatrix,alpha=0.01)
[1] "category = 曝露 , relative risk = 0.173721236521721"
[1] "category = 曝露 , 99 % 信頼区間 = [ 0.140263410926649 , 0.215159946697844 ]"
```

出力より、相対危険度は 0.174 であり、99% 信頼区間は [0.140, 0.215] である。相対危険度が 0.174 であるということは、曝露された（研究している処理または環境因子に）人は、曝露されていない人と比べて 0.174 倍疾病の危険があることを意味する。

相対危険度が 1（すなわち、信頼区間が 1 を含む）とき、曝露と疾病の間に関係はないことを意味する。他方、相対危険度 > 1 のとき、曝露と疾病との間に正の関連があり、相対危険度 < 1 のとき、負の関連がある。相対危険度はコホート研究で用い、症例対照研究では用いない。

曝露に 2 つ以上のカテゴリがある（例えば、非喫煙に対する巻きたばこの喫煙、葉たばこの喫煙）場合にも、`calcRelativeRisk()` 関数を利用することができることに注意。この目的には、後で述べる `calcOddsRatio()` と同様に利用する。

2.3 コホートあるいは症例対照研究に対するオッズ比の計算

曝露（喫煙や薬といった処理や環境因子）による疾病の相対危険度と同様、コホート研究において曝露と疾病の間の関連に対するオッズ比を計算することができる。オッズ比は、症例対照研究においてもよく利用される。

曝露と疾病の間の関連のオッズ比は、次の2つの値の比である：(i) 処理または環境因子に曝露された人が発症する確率を、曝露された人が発症しない確率で割った値、(ii) 処理または環境因子に曝露されなかった人が発症する確率を、曝露されなかった人が発症しない確率で割った値。

コホート研究または症例対照研究において、この場合も、データは次のような形になる：

	疾病	コントロール
曝露	156	9421
非曝露	1531	14797

次を入力することにより、Rにこのデータを読み込むことができる：

```
> mymatrix <- matrix(c(156,9421,1531,14797),nrow=2,byrow=TRUE)
> colnames(mymatrix) <- c("疾病","対照")
> rownames(mymatrix) <- c("曝露","非曝露")
> print(mymatrix)
      疾病 対照
曝露  156 9421
```

次に示す関数 `calcOddsRatio()` を用いて、曝露と疾病の関連に対するオッズ比を計算することができる。これを利用するには、先ずRにコピー&ペーストする必要がある。

```
> calcOddsRatio <- function(mymatrix,alpha=0.05,referencerow=2,quiet=FALSE)
{
  numrow <- nrow(mymatrix)
  myrownames <- rownames(mymatrix)

  for (i in 1:numrow)
  {
    rowname <- myrownames[i]
    DiseaseUnexposed <- mymatrix[referencerow,1]
    ControlUnexposed <- mymatrix[referencerow,2]
    if (i != referencerow)
    {
      DiseaseExposed <- mymatrix[i,1]
      ControlExposed <- mymatrix[i,2]

      totExposed <- DiseaseExposed + ControlExposed
      totUnexposed <- DiseaseUnexposed + ControlUnexposed

      probDiseaseGivenExposed <- DiseaseExposed/totExposed
      probDiseaseGivenUnexposed <- DiseaseUnexposed/totUnexposed
      probControlGivenExposed <- ControlExposed/totExposed
      probControlGivenUnexposed <- ControlUnexposed/totUnexposed

      # オッズ比の計算
      oddsRatio <- (probDiseaseGivenExposed*probControlGivenUnexposed)/
                    (probControlGivenExposed*probDiseaseGivenUnexposed)
      if (quiet == FALSE)
      {
```

```

    print(paste("category =", rowname, ", odds ratio = ",oddsRatio))
  }

  # 信頼区間の計算
  confidenceLevel <- (1 - alpha)*100
  sigma <- sqrt((1/DiseaseExposed)+(1/ControlExposed)+
                (1/DiseaseUnexposed)+(1/ControlUnexposed))
  # sigma は, オッズ比の対数の推定値の標準誤差
  z <- qnorm(1-(alpha/2))
  lowervalue <- oddsRatio * exp(-z * sigma)
  upervalue <- oddsRatio * exp( z * sigma)
  if (quiet == FALSE)
  {
    print(paste("category =", rowname, ", ", confidenceLevel,
               "% confidence interval = [",lowervalue,",",upervalue,"]"))
  }
}
}
if (quiet == TRUE && numrow == 2) # 処理水準が2つのみの場合 (曝露/非曝露)
{
  return(oddsRatio)
}
}

```

これにより、曝露と疾病との間の関連のオッズ比の計算とその信頼区間を計算するためにこの関数を利用することができる。立て叔母、オッズ比と 95% 信頼区間を求めるにや次のようにする：

```

> calcOddsRatio(mymatrix,alpha=0.05)
[1] "category = 曝露 , odds ratio = 0.160039091621751"
[1] "category = 曝露 , 95 % confidence interval = [ 0.135460641900536 , 0.189077140693912 ]"

```

出力より、オッズ比の推定値は 0.160 であり、オッズ比の 95% 信頼区間は [0.135, 0.189] である。

オッズ比が 1 (すなわち、信頼区間が 1 を含む) とき、曝露と疾病との間には関係はないことを意味する。他方、オッズ比 > 1 のとき、曝露と疾病との間に正の関連があり、オッズ比 < 1 のとき、負の関連がある。オッズ比はコホート研究でも症例対照研究でも用いることができる。

曝露に複数のカテゴリがある (例えば、非喫煙に対する巻きたばこの喫煙、葉たばこの喫煙) 場合もある。この場合、データは次のようになる。

	疾病	対照
曝露 1	30	24
曝露 2	76	241
非曝露	82	509

このデータを R に入力するには、次のようにする (今、データは 3 行あるので、“nrow=3” と指定する必要があることに注意)。

```

> mymatrix <- matrix(c(30, 24, 76, 241, 82, 509), nrow=3, byrow=TRUE)
> colnames(mymatrix) <- c("疾病", "対照")
> rownames(mymatrix) <- c("曝露 1", "曝露 2", "非曝露")
> print(mymatrix)
      疾病 対照
曝露 1   30  24
曝露 2   76 241
非曝露   82 509

```

このデータに対して、`calcOddsRatio()` 関数を用いて非曝露に対して各曝露のオッズ比を計算することもできる。このデータ行列には3行あるので、“`referencerow=`” 引数により指定する。

```
> calcOddsRatio(mymatrix, referencerow=3)
[1] "category = 曝露 1 , odds ratio = 7.75914634146342"
[1] "category = 曝露 1 , 95 % confidence interval = [ 4.32163714854064 , 13.9309131884372 ]"
[1] "category = 曝露 2 , odds ratio = 1.95749418075094"
[1] "category = 曝露 2 , 95 % confidence interval = [ 1.38263094540732 , 2.77137111707344 ]"
```

コホート研究のデータの場合（症例対照研究ではなく）には、曝露の各カテゴリに対して相対危険度を求めることができる：

```
> calcRelativeRisk(mymatrix, referencerow=3)
[1] "category = 曝露 1 , relative risk = 4.00406504065041"
[1] "category = 曝露 1 , 95 % 信頼区間 = [ 2.93130744422409 , 5.46941498113737 ]"
[1] "category = 曝露 2 , relative risk = 1.72793721628068"
[1] "category = 曝露 2 , 95 % 信頼区間 = [ 1.30507489771431 , 2.2878127750653 ]"
```

2.4 コホート研究または症例対照研究における曝露と疾病との関連の検定

コホート研究または症例対照研究において、ある処理または環境への曝露（例えば、喫煙や薬の服用）と疾病との間の関連の統計的検定を行いたいことがある。

R では、カイ 2 乗検定 (`chisq.test()`) または Fisher (フィッシャー) の正確検定 (`fisher.test()`) により関連の検定 (独立性検定) を行うことができる。例えば、上記の例に適用するには、次のようにする：

```
> print(mymatrix)
      疾病 対照
曝露 1    30   24
曝露 2    76  241
非曝露    82  509
> chisq.test(mymatrix)

Pearson's Chi-squared test

data: mymatrix
X-squared = 60.5762, df = 2, p-value = 7.015e-14

> fisher.test(mymatrix)

Fisher's Exact Test for Count Data

data: mymatrix
p-value = 5.263e-12
alternative hypothesis: two.sided
```

カイ 2 乗検定の p 値は約 $7e-14$ であり、Fisher の正確検定の p 値は $5e-12$ である。どちらも小さな値なので ($P < 0.05$)、曝露と疾病との間の関連は有意である (有意水準を 0.05 としている)。

2.5 層別変数がある場合の (Mantel-Haenszel) オッズ比の計算

コホート研究または症例対照研究において、研究対象となる人の性別といった変数によって層別されたデータを扱う場合がある。例えば、曝露 (薬の服用や喫煙) と特定の疾病との関係に関する表形式の

データであるが、女性に対する情報の表と、男性に対する情報の表に層別されているようなものである。

女性データ

	疾病	対照
曝露	4	5
非曝露	5	103

男性データ

	疾病	対照
曝露	10	3
非曝露	5	43

このデータを R に入力するには、次のようにする：

```
> mymatrix1 <- matrix(c(4, 5, 5, 103), nrow=2, byrow=TRUE)
> colnames(mymatrix1) <- c("疾病", "対照")
> rownames(mymatrix1) <- c("曝露", "非曝露")
> print(mymatrix1)
      疾病 対照
曝露    4   5
非曝露   5 103
> mymatrix2 <- matrix(c(10,3,5,43), nrow=2, byrow=TRUE)
> colnames(mymatrix2) <- c("疾病", "対照")
> rownames(mymatrix2) <- c("曝露", "非曝露")
> print(mymatrix2)
      疾病 対照
曝露   10   3
非曝露   5  43
```

Mantel-Haenszel (マンテル・ヘンツェル) のオッズ比は、層別変数 (ここでは性別) による交絡効果をコントロールして、曝露と疾病との間の関連のオッズ比の推定を行う。Mantel-Haenszel のオッズ比を計算するための関数 “cmh.test()” は、“lawstat” というパッケージにある。この関数を利用するには、“lawstat” パッケージをインストールしておく必要がある (R のパッケージのインストールについては、第 1.3 節の「R パッケージのインストール法」を参照)。一度 “lawstat” パッケージをインストールしておくこと、次を入力することにより、これを利用できる：

```
> library("lawstat")
```

“cmh.test()” 関数を用いて Mantel-Haenszel オッズ比を計算することができる：

```
> myarray <- array(c(mymatrix1, mymatrix2), dim=c(2,2,2))
> cmh.test(myarray)
```

Cochran-Mantel-Haenszel Chi-square Test

```
data: myarray
CMH statistic = 40.512, df = 1.000, p-value = 0.000, MH Estimate = 23.001, Pooled
Odd Ratio = 25.550, Odd Ratio of level 1 = 16.480, Odd Ratio of level 2 = 28.667
```

結果より、第 1 の層 (女性) に対するオッズ比は 16.480、第 2 の層 (男性) に対するオッズ比は 28.667、データをプールしたときのオッズ比は 25.550 であることがわかる。Mantel-Haenszel のオッズ比は、23.001 と推定される。

cmh.test() 関数は、層別変数 (ここでは性別) をコントロールしたときの曝露と疾病との関連の検定である Cochran-Mantel-Haenszel のカイ 2 乗値も出力 (CMH statistics) している。今の場合、この

検定の p 値は 0.000 なので、曝露と疾病との関連は有意である。

2つの層のオッズ比がかなり異なる場合、層別因子（今の例では性別）は交絡因子であり、このとき Mantel-Haenszel のオッズ比を用いるべきではない。層によるオッズ比が異なるかどうかを検定するには、Tarone の検定を利用する。Tarone の検定を行うには、“metafor” パッケージにある関数を利用することができる。この関数を利用するには、“metafor” パッケージをインストールしておく必要がある（R のパッケージのインストールについては、第 1.3 節の「R パッケージのインストール法」を参照）。一度 “metafor” パッケージをインストールしておくこと、次を入力することにより、これを利用できる：

```
> library("metafor")
```

Tarone の検定を行うために、次に示す calcTaronesTest() 関数を利用することができる。これを利用するには、この関数を R にコピー&ペーストする：

```
> calcTaronesTest <- function(mylist,referencerow=2)
{
  numstrata <- length(mylist)
  # 各層における曝露グループの人数のアレイ"ntrt"を作る
  # 各層における非曝露グループの人数のアレイ"nctrl"を作る
  # 各層における疾病を持つ曝露グループの人数のアレイ"ptrt"を作る
  # 各層における疾病を持つ非曝露グループの人数のアレイ"pctrl"を作る
  # 各層における疾病を持たない曝露グループの人数のアレイ"htrt"を作る
  # 各層における疾病を持たない非曝露グループの人数のアレイ"hctrl"を作る
  ntrt <- vector()
  nctrl <- vector()
  ptrt <- vector()
  pctrl <- vector()
  htrt <- vector()
  hctrl <- vector()
  if (referencerow == 1) { nonreferencerow <- 2 }
  else { nonreferencerow <- 1 }
  for (i in 1:numstrata)
  {
    mymatrix <- mylist[[i]]
    DiseaseUnexposed <- mymatrix[referencerow, 1]
    ControlUnexposed <- mymatrix[referencerow, 2]
    totUnexposed <- DiseaseUnexposed + ControlUnexposed
    nctrl[i] <- totUnexposed
    pctrl[i] <- DiseaseUnexposed
    hctrl[i] <- ControlUnexposed
    DiseaseExposed <- mymatrix[nonreferencerow, 1]
    ControlExposed <- mymatrix[nonreferencerow, 2]
    totExposed <- DiseaseExposed + ControlExposed
    ntrt[i] <- totExposed
    ptrt[i] <- DiseaseExposed
    htrt[i] <- ControlExposed
  }
  # "metafor"パッケージにある rma.mh 関数を利用して Tarone の均一性の検定を計算する
  tarone <- rma.mh(ptrt, htrt, pctrl, hctrl, ntrt, nctrl)
  pvalue <- tarone$TAp
  print(paste("Tarone の検定の p 値 =", pvalue))
}
```

“calcTaronesTest()” 関数を利用して、Tarone の検定を行うことができる。

```
> mylist <- list(mymatrix1, mymatrix2)
> calcTaronesTest(mylist)
```

```
[1] "Tarone の検定の p 値 = 0.627420741721689"
```

Tarone の検定の p 値は 0.05 より大きいので、有意水準を 0.05 とするとき、異なる層間（本例では、男性と女性）のオッズ比に違いがあるとはいえない。

2.6 対症例対照研究における曝露と疾病との関連の検定

対症例対照研究（Matched Case-Control Study）では、疾病を持つ各個人にマッチする対照の個人がいる。マッチしている対照個人は、疾病を持つ人と年齢、人種、性別等が同じで人である。このとき、対照の人と疾病を持つ人が同じ因子に曝露されているかどうかを調べる。データは次のようになる：

	対照, 曝露	対照, 非曝露
疾病, 曝露	10	57
疾病, 非曝露	13	96

このデータは、次のようにして R に入力する：

```
> mymatrix <- matrix(c(10, 57, 13, 95), nrow=2, byrow=TRUE)
> colnames(mymatrix) <- c("対照-曝露", "対照-非曝露")
> rownames(mymatrix) <- c("疾病-曝露", "疾病-非曝露")
> print(mymatrix)
```

```
      対照-曝露 対照-非曝露
疾病-曝露      10         57
疾病-非曝露     13         95
```

次に示す関数 `calcMHRatio()` を用いて、曝露と疾病との関連の Mantel-Haenszel のオッズ比を計算することができる。これを利用するには、先ず R にコピー&ペーストしておく。

```
> calcMHRatio <- function(mymatrix, alpha=0.05)
{
  caseExposedControlUnexposed <- mymatrix[1, 2]
  caseUnexposedControlExposed <- mymatrix[2, 1]
  MHRatio <- caseExposedControlUnexposed / caseUnexposedControlExposed
  print(paste("Mantel-Haenszel のオッズ比 =", MHRatio))

  # 信頼区間の計算
  confidenceLevel <- (1 - alpha) * 100
  sigma <- sqrt((1 / caseExposedControlUnexposed) + (1 / caseUnexposedControlExposed))
  # sigma は、オッズ比の対数の推定値の標準誤差
  z <- qnorm(1 - (alpha / 2))
  lowervalue <- MHRatio * exp(-z * sigma)
  uppervalue <- MHRatio * exp( z * sigma)
  print(paste(confidenceLevel, "% confidence interval = [", lowervalue, ", ", uppervalue, "]"))
}
```

関数 `calcMHRatio()` を用いて、今のデータセットに対して Mantel-Haenszel のオッズ比を計算することができる。

```
> calcMHRatio(mymatrix)
[1] "Mantel-Haenszel のオッズ比 = 4.38461538461539"
[1] "95 % confidence interval = [ 2.40054954520192 , 8.00852126107185 ]"
```

出力より、Mantel-Haenszel のオッズ比の推定値は 4.38 であり、オッズ比の 95% 信頼区間は [2.40, 8.01] である。

1対1に対応のある症例対照研究において、曝露と疾病との間の関連の検定として McNemar (マクネマ) の検定を用いることができる。McNemar の検定は、関数 “`mcnemar.test()`” を利用して実行できる。

```
> mcnemar.test(mymatrix)
```

```
McNemar's Chi-squared test with continuity correction
```

```
data: mymatrix
```

```
McNemar's chi-squared = 26.4143, df = 1, p-value = 2.755e-07
```

McNemar の検定の p 値は 0.05 より小さいので、曝露と疾病との間の関連は有意である (有意水準を 0.05 としている)。

2.7 用量反応分析

用量反応分析においては、因子に関する異なる用量 (例えば、1日当たりの喫煙本数、薬の服用量) に対する疾病の情報がある。例えば、データは次のような形をしている:

	疾病	対照
用量=2	35	82
用量=9.5	250	293
用量=19.5	196	190
用量=37	136	71
用量=50	32	13

このデータを R に入力するには、次のようにする (今、データは 5 行あるので、“`nrow=5`” と指定する必要があることに注意)。

```
> mymatrix <- matrix(c(35, 82, 250, 293, 196, 190, 136, 71, 32, 13), nrow=5, byrow=TRUE)
> colnames(mymatrix) <- c("疾病", "対照")
> rownames(mymatrix) <- c("2", "9.5", "19.5", "37", "50")
> print(mymatrix)
```

```
  疾病 対照
2      35  82
9.5    250 293
19.5   196 190
37     136  71
50     32  13
```

今の場合、通常、用量水準 (曝露の強さ) と疾病との関連に対して、最低用量と比較するオッズ比を計算する。次に示す関数 “`doseSpecificOddsRatios()`” を用いてこのオッズ比を計算することができるが、そのためにはまず、R にコピー&ペーストする必要がある。

```
> doseSpecificOddsRatios <- function(mymatrix, referencerow=1)
{
  numstrata <- nrow(mymatrix)
  # 階層ごとのオッズ比と疾病とのオッズの計算:
  doses <- as.numeric(rownames(mymatrix))
  for (i in 1:numstrata)
  {
    dose <- doses[i]
    # オッズ比の計算:
    DiseaseExposed <- mymatrix[i, 1]
    DiseaseUnexposed <- mymatrix[i, 2]
```

```

ControlExposed <- mymatrix[referencerow, 1]
ControlUnexposed <- mymatrix[referencerow, 2]
totExposed <- DiseaseExposed + ControlExposed
totUnexposed <- DiseaseUnexposed + ControlUnexposed
probDiseaseGivenExposed <- DiseaseExposed / totExposed
probDiseaseGivenUnexposed <- DiseaseUnexposed / totUnexposed
probControlGivenExposed <- ControlExposed / totExposed
probControlGivenUnexposed <- ControlUnexposed / totUnexposed
oddsRatio <- (probDiseaseGivenExposed * probControlGivenUnexposed) /
              (probControlGivenExposed * probDiseaseGivenUnexposed)
print(paste("用量 =", dose, ", オッズ比 =", oddsRatio))
}
}

```

この関数を用いて、上記のデータに対する各用量のオッズ比を計算することができる。

```

> doseSpecificOddsRatios(mymatrix)
[1] "用量 = 2 , オッズ比 = 1"
[1] "用量 = 9.5 , オッズ比 = 1.99902486591906"
[1] "用量 = 19.5 , オッズ比 = 2.41684210526316"
[1] "用量 = 37 , オッズ比 = 4.48772635814889"
[1] "用量 = 50 , オッズ比 = 5.76703296703297"

```

他によく行われる分析として、用量に対する疾病の対数オッズの線形回帰直線を当てはめ、回帰直線の傾きが 0 と有意に異なるかどうかの検定がある。回帰直線の傾きが有意に 0 と異なるならば、曝露したとき、用量と疾病とのオッズの間に有意な線形関係があることになる。次の関数 `doseOddsDiseaseRegression()` を用いて、回帰直線への当てはめと、傾きが 0 かどうかの検定を行うことができる。利用するには、R にコピー&ペーストする必要がある。

```

> doseOddsDiseaseRegression <- function(mymatrix, referencerow=1)
{
  numstrata <- nrow(mymatrix)
  # 階層ごとのオッズ比と疾病とのオッズの計算 :
  myodds <- vector()
  doses <- as.numeric(rownames(mymatrix))
  for (i in 1:numstrata)
  {
    dose <- doses[i]
    # 曝露と疾病とのオッズの計算 :
    DiseaseExposed <- mymatrix[i,1]
    ControlExposed <- mymatrix[i,2]
    totExposed <- DiseaseExposed + ControlExposed
    probDiseaseGivenExposed <- DiseaseExposed / totExposed
    probNotDiseaseGivenExposed <- ControlExposed / totExposed
    odds <- probDiseaseGivenExposed / probNotDiseaseGivenExposed
    logodds <- log(odds) # 自然対数
    myodds[i] <- logodds
  }
  # 用量に対する log(odds) の回帰直線の傾きが 0 かどうかの検定 :
  lm1 <- lm(myodds ~ doses)
  summarylm1 <- summary(lm1)
  coeff1 <- summarylm1$coefficients
  # 傾きが 0 かどうかの F 検定に対する p 値の計算 :
  pvalue <- coeff1[2,4]
  print(paste("傾きが 0 の F 検定の p 値 =", pvalue))
  # 用量に対する log(odds) のプロット :

```

```

plot(doses, myodds, xlab="用量", ylab="log(odds)", main="用量に対する log(odds) のプロット")
}

```

用量に対する $\log(\text{odds})$ (対数オッズ) の線形回帰直線の傾きが 0 かどうかの検定, および用量に対する $\log(\text{odds})$ のプロットを `doseOddsDiseaseRegression()` 関数を用いて行うことができる.

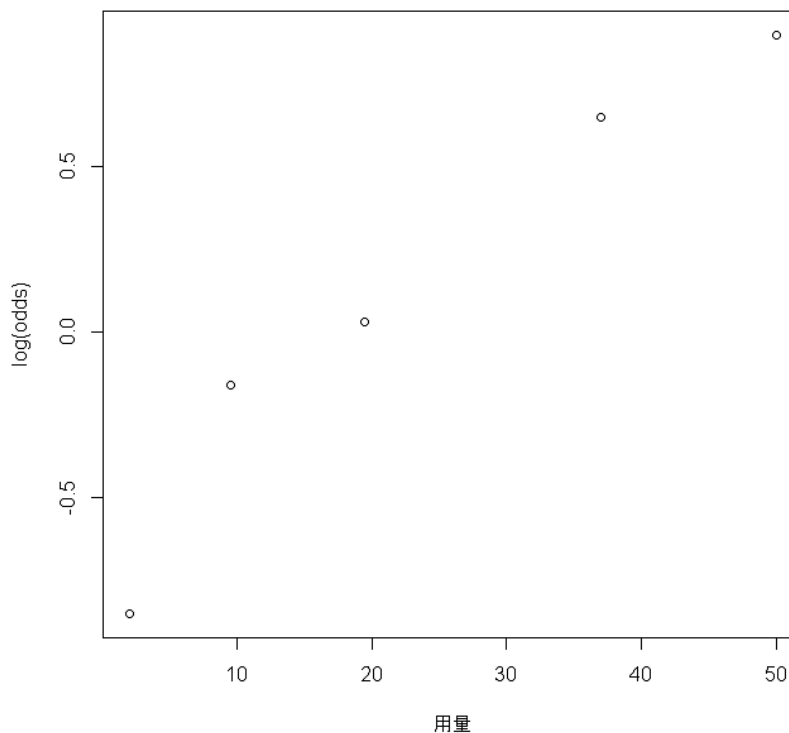
```

> doseOddsDiseaseRegression(mymatrix)
[1] "傾きが 0 の F 検定の p 値 = 0.00659217584881777"

```

この検定の p 値は 0.05 より小さいので, 有意水準を 0.05 とするとき, 線形回帰直線の傾きは 0 と有意に異なる. すなわち, 用量と曝露したときに発症するオッズの間には有意な関係がある.

用量に対する $\log(\text{odds})$ のプロット



2.8 ランダム化比較試験に必要なサンプルサイズ

2 群 (例えば, 検証したい薬を服用したグループと, プラセボを服用したグループ) を持つランダム化比較試験 (randomized control trial) を実行したい場合, 生物医学統計に一般的なタスクとして, 試験に必要なサンプルサイズの計算がある. 次の関数 “`calcSampleSizeForRCT()`” を用いて各グループに必要なサンプルサイズを計算することができる. この関数を R にコピー&ペーストする.

```

> calcSampleSizeForRCT <- function(alpha,gamma,piT,piC,p=0)
{
  # p is the estimated of the likely fraction of losses to follow-up
  qalpha <- qnorm(p=1-(alpha/2))
  qgamma <- qnorm(p=gamma)
  pi0 <- (piT + piC)/2
  numerator <- 2 * ((qalpha + qgamma)^2) * pi0 * (1 - pi0)
  denominator <- (piT - piC)^2
  n <- numerator/denominator
}

```

```

n <- ceiling(n) # 整数に切り上げ
# adjust for likely losses to follow-up
n <- n/(1-p)
n <- ceiling(n) # 整数に切り上げ
print(paste("各試験群に必要なサンプルサイズ = ",n))
}

```

“calcSampleSizeForRCT()” 関数を使うとき、有意水準と検出力、対照群（プラセボの服用グループ）における疾病のインシデントの推定値、処理群（薬の服用グループ）の疾病のインシデントの推定値を指定する必要がある。例えば、有意水準を 5%、検出力を 90%、処理群と対照群の疾病のインシデントの推定値をそれぞれ 0.20 と 0.15 としたい場合、次を入力する：

```

> calcSampleSizeForRCT(alpha=0.05, gamma=0.90, piT=0.15, piC=0.2)
[1] "各試験群に必要なサンプルサイズ = 1214"

```

これにより、必要なサンプルサイズは各グループで 1214 名であり、ランダム化比較試験全体としては $1214 * 2 = 2428$ 人となる。

追跡調査に失敗する可能性があり、それを見積もることができる場合、試験に必要な人数の推定値を調整することができる。例えば、追跡調査に 10% 失敗すると見積もるとき、試験に必要な人数を次のように計算することができる。

```

> calcSampleSizeForRCT(alpha=0.05, gamma=0.90, piT=0.15, piC=0.2, p=0.1)
[1] "各試験群に必要なサンプルサイズ = 1349"

```

出力より、追跡調査で 10% が失われるので、試験において各群で必要なのは 1349 名となるので、全体では $1349 * 2 = 2698$ 名となる。

2.9 ランダム化比較試験の検出力の計算

実務上の理由から、ランダム化比較試験の各群の人数の最大数が決まっている場合、この試験の検出力を計算することができる。これには次の関数 “calcPowerForRC()” で実行できる：

```

> calcPowerForRCT <- function(alpha,piT,piC,n)
{
  qalpha <- qnorm(p=1-(alpha/2))
  pi0 <- (piT + piC)/2
  denominator <- 2 * pi0 * (1 - pi0)
  fraction <- n/denominator
  qgamma <- (abs(piT - piC) * sqrt(fraction)) - qalpha
  gamma <- pnorm(qgamma)
  print(paste("ランダム化比較試験の検出力 = ",gamma))
}

```

例えば、500 人の子供（対照群で 250 名、処理群で 250 名）、有意水準を 0.05、対照群と処理群の疾病とのインシデントの推定値をそれぞれ 0.3 と 0.2 とするとき、次を入力する：

```

> calcPowerForRCT(alpha=0.05, piT=0.2, piC=0.3, n=250)
[1] "ランダム化比較試験の検出力 = 0.73303725668939"

```

出力より、ランダム化比較試験の検出力は 73% である。

2.10 複数のランダム化比較試験のメタ分析に対するフォレストプロットの作成

複数のランダム化比較試験のメタ分析を行いたい場合、フォレストプロットによるグラフ化が有益である。例えば、次のような異なるランダム化比較試験の結果があるとする。

試験 1 のデータ:

	疾病	対照
曝露	198	728
非曝露	128	576

試験 2 のデータ:

	疾病	対照
曝露	96	437
非曝露	101	342

試験 3 のデータ:

	疾病	対照
曝露	1105	4243
非曝露	1645	6703

試験 4 のデータ:

	疾病	対照
曝露	741	2905
非曝露	594	2418

試験 5 のデータ:

	疾病	対照
曝露	264	1091
非曝露	907	3671

試験 6 のデータ:

	疾病	対照
曝露	105	408
非曝露	348	1248

試験 7 のデータ:

	疾病	対照
曝露	138	431
非曝露	436	1576

このデータを R に入力するには、次のようにする：

```
> mymatrix1 <- matrix(c(198, 728, 128, 576), nrow=2, byrow=TRUE)
> mymatrix2 <- matrix(c(96, 437, 101, 342), nrow=2, byrow=TRUE)
> mymatrix3 <- matrix(c(1105, 4243, 1645, 6703), nrow=2, byrow=TRUE)
> mymatrix4 <- matrix(c(741, 2905, 594, 2418), nrow=2, byrow=TRUE)
> mymatrix5 <- matrix(c(264, 1091, 907, 3671), nrow=2, byrow=TRUE)
```



```

> mymatrix6 <- matrix(c(105, 408, 348, 1248), nrow=2, byrow=TRUE)
> mymatrix7 <- matrix(c(138, 431, 436, 1576), nrow=2, byrow=TRUE)
> mylist <- list(mymatrix1, mymatrix2, mymatrix3, mymatrix4, mymatrix5, mymatrix6, mymatrix7)

```

次に示す関数 “makeForestPlotForRCTs()” を用いてフォレストプロットを作成することができる。
この関数は、“rmeta” パッケージを利用するので、これがインストールされている必要がある。

```

> makeForestPlotForRCTs <- function(mylist, referencerow=2)
{
  library("rmeta")
  numstrata <- length(mylist)
  # 各層における曝露グループの人数のアレイ"ntrt"を作る
  # 各層における非曝露グループの人数のアレイ"nctrl"を作る
  # 各層における疾病を持つ曝露グループの人数のアレイ"ptrt"を作る
  # 各層における疾病を持つ非曝露グループの人数のアレイ"pctrl"を作る
  ntrt <- vector()
  nctrl <- vector()
  ptrt <- vector()
  pctrl <- vector()
  if (referencerow == 1) { nonreferencerow <- 2 }
  else { nonreferencerow <- 1 }
  for (i in 1:numstrata)
  {
    mymatrix <- mylist[[i]]
    DiseaseUnexposed <- mymatrix[referencerow,1]
    ControlUnexposed <- mymatrix[referencerow,2]
    totUnexposed <- DiseaseUnexposed + ControlUnexposed
    nctrl[i] <- totUnexposed
    pctrl[i] <- DiseaseUnexposed
    DiseaseExposed <- mymatrix[nonreferencerow,1]
    ControlExposed <- mymatrix[nonreferencerow,2]
    totExposed <- DiseaseExposed + ControlExposed
    ntrt[i] <- totExposed
    ptrt[i] <- DiseaseExposed
  }
  names <- as.character(seq(1,numstrata))
  myMH <- meta.MH(ntrt, nctrl, ptrt, pctrl, conf.level=0.95, names=names)
  print(myMH)
  tabletext<-cbind(c("", "Study", myMH$names, NA, "Summary"),
    c("疾病", "(曝露)", ptrt, NA, NA),
    c("疾病", "(非曝露)", pctrl, NA, NA),
    c("", "オッズ比", format(exp(myMH$logOR), digits=2), NA, format(exp(myMH$logMH), digits=2)))
  print(tabletext)
  m<- c(NA, NA, myMH$logOR, NA, myMH$logMH)
  l<- m-c(NA, NA, myMH$selogOR, NA, myMH$selogMH) * 2
  u<- m+c(NA, NA, myMH$selogOR, NA, myMH$selogMH) * 2
  forestplot(tabletext, m, l, u, zero=0, is.summary=c(TRUE, TRUE, rep(FALSE, 8), TRUE),
    clip=c(log(0.1), log(2.5)), xlog=TRUE,
    col=meta.colors(box="royalblue", line="darkblue", summary="royalblue"))
}

```

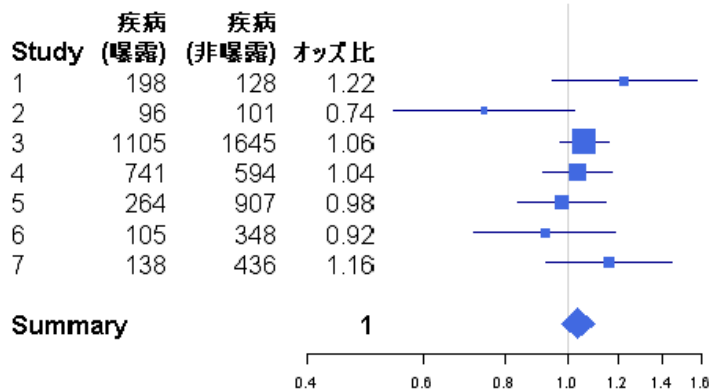
次により、7つの異なる試験からのデータのフォレストプロットを作成することができる。

```

> makeForestPlotForRCTs(mylist)

```

7試験における曝露と疾病との関連に関するオッズ比に、有意な差があるかどうかを検定する Tarone の検定を実行するために、“calcTaronesTest()” 関数を用いることができる。



```
> calcTaronesTest(mylist)
[1] "Tarone の検定の P 値 = 0.190239054737704"
```

Tarone の検定の p 値は 0.05 より大きいので、有意水準を 0.05 とするとき、異なる層（本例では、7 試験）間のオッズ比に違いがあるとはいえない。

2.11 リンクと参考文献

さらに学習するためのリンク・参考文献を紹介する。

R をより詳細に知るには、“Kickstarting R” ウェブサイト (cran.rproject.org/doc/contrib/Lemon-kickstart) にある良いオンライン・チュートリアルを参照のこと。

別のチュートリアルとしてもう少し詳細なものが、“Introduction to R” ウェブサイト (cran.rproject.org/doc/manuals/R-intro.html) にある。

生物医学統計を学ぶには、Open University 発行の本“Medical statistics”（製品コード M249/01. Open University のショップより購入可）を強く推薦する。

2.12 謝辞

本書を作成する際の Sphinx (<http://sphinx.pocoo.org>), 異なるバージョンの管理のための github (<https://github.com/>), ビルドし、ディストリビュートするための readthedocs (<http://readthedocs.org/>) の使い方について支援を受けたことに対して、Noel O’Boyle に感謝する。

本ブックレットの例の多くは、Open University のショップで入手できる優れた本、“Medical Statistics”（製品コード M249/01）より着想を得た。

本ブックレットを改良するために有益なコメントおよび示唆をいただいた Tony Burton, Richard A. Friedman, Duleep Samuel, R.Heberto Ghezzeo, David Levine, Lavinia Gordon, Friedrich Leisch, Phil Spector に感謝する。

2.13 連絡

本書に関して間違いや提案があれば私 (Avril Coghlan) にメール (アドレス a.coghlan@ucc.ie) を送付していただくと幸いです。

2.14 ライセンス

本書の内容は、Creative Commons Attribution 3.0 License の下で許諾されている。

第 3 章

謝辞

本書を作成する際の Sphinx (<http://sphinx.pocoo.org>), 異なるバージョンの管理のための github (<https://github.com/>), ビルドし, ディストリビュー特するための readthedocs (<http://readthedocs.org/>) の使い方について支援を受けたことに対して, Noel O'Boyle に感謝する.

第 4 章

連絡

本書に関して間違いや提案があれば私 (Avril Coghlan) にメール (a.coghlan@ucc.ie) を送付していただければ幸いです。

第 5 章

ライセンス

本書の内容は、クリエイティブ・コモンズ (Creative Commons) Attribution 3.0 ライセンスで公開している