

RjpWiki アーカイブス

【古典的検定関数一覧（超訳付き！）（06.06.25）】

R は豊富な代表的古典的検定関数（パラメトリック、ノンパラメトリック検定）を持つ。

注意： 帰無分布（帰無仮説の下での検定統計量の分布）に付いては、正確な分布を用いるものと、近似分布を用いるものがある。両者がオプションで選べるものもあれば、近似分布しか使えないものもある。

R の検定関数は検定の副産物として、関連パラメータの信頼区間を同時に計算するようになっている。

注意： 統計量の帰無仮説の下での分布を漸近近似で求めるものが多数を占める。こうした場合一般論としてデータの数がある程度大きいことが前提条件になる。特にヘルプドキュメントに注意が無いことが多いが、実際は近似が不十分と思われる場合は警告が出る関数がある。

1 ノンパラメトリック検定関数

ノンパラメトリック検定とは、通常母集団分布に対しては密度関数を持つことだけを仮定し、順序（特にランク）統計量を基礎とする検定統計量を用いる検定のことである。これらの検定関数の中にはオプションでパラメトリック検定を実行できるものもある。

1.1 スケールパラメータ（散布度）の差の二標本 Ansari-Bradley 検定

尺度パラメータの同一性についての検定である。

1.1.1 使用法

```
ansari.test(x, y, alternative = c("two.sided", "less", "greater"),
            exact = NULL, conf.int = FALSE, conf.level = 0.95, ...)
ansari.test(formula, data, subset, na.action, ...)

### exactRankTests ライブラリ
ansari.exact(x, y, alternative = c("two.sided", "less", "greater"),
            exact = NULL, conf.int = FALSE, conf.level = 0.95, ...)
ansari.exact(formula, data, subset, na.action, ...)
```

1.1.2 引数

<code>x, y</code>	2 つのデータベクトル
<code>alternarive</code>	対立仮説の種類 "two.sided", "less", "greater" のいずれか 最初の 1 文字だけでもよい 両側検定なら "two.sided" (デフォルト) 片側検定なら "less" または "greater"
<code>exact</code>	正確な P 値を計算するかどうかを表す論理値 デフォルトは <code>NULL</code> であり, その場合には, それぞれの群で有効なデータ数が 50 未満で, タイがないときには正確な P 値を計算する <code>ansari.exact</code> では, タイがある場合にも正確な P 値を計算できる
<code>conf.int</code>	信頼区間を計算する場合には <code>TRUE</code> を指定する デフォルトでは <code>FALSE</code> (信頼区間は計算しない)
<code>conf.level</code>	信頼率 デフォルトは 0.95
<code>formula</code>	「左辺値 ~ 右辺値」の形をしたモデル式 左辺値は尺度母数を検定する数値変数 右辺値は 2 群を表す 2 つの水準を持つ <code>factor</code> 変数
<code>data</code>	モデル式に出てくる変数がデータフレームに含まれる場合は, それを含む データフレーム名
<code>subset</code>	観察データのサブセット
<code>na.action</code>	データに <code>NA</code> が含まれるときに適用される関数 (注) デフォルトは <code>getOption("na.action")</code> により示されるもの (通常は <code>na.omit</code> 関数)
<code>...</code>	その他の引数

1.1.3 戻り値

1.1.4 関数への入力

```
$ alternarive: 対立仮説の種類
```

1.1.5 関数からの出力

```
$ statistic : 検定統計量  
$ p.value   : P 値  
$ null.vallue: 尺度パラメータの比. デフォルトでは 1 (尺度パラメータが等しい)  
$ conf.int   : 尺度パラメータの比の信頼区間  
$ estimate   : 尺度パラメータの比の推定値 (標本値)  
$ method     : 検定の種別  
$ data.name  : データの記述
```

1.1.6 例1 2つのデータベクトルで検定

```
> x <- c(1,2,3,2,3,4,3,3,2,2,1,2,3)
> y <- c(2,1,3,4,3,2,2,3,4,5,3,1,2)

> ansari.test(x, y)

      Ansari-Bradley test

data:  x and y
AB = 98, p-value = 0.3725
alternative hypothesis: true ratio of scales is not equal to 1

Warning message:
タイがあるため、正確な p 値を計算することができません
                                in: ansari.test.default(x, y)

> library(exactRankTests)
> ansari.exact(x, y) # 同値があっても正確な P 値が計算される

      Ansari-Bradley test

data:  x and y
AB = 98, p-value = 0.4624
alternative hypothesis: true ratio of scales is not equal to 1

> var.test(x, y) # 対応するパラメトリック検定

      F test to compare two variances

data:  x and y
F = 0.5413, num df = 12, denom df = 12, p-value = 0.3014
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1651628 1.7739390
sample estimates:
ratio of variances
 0.5412844
```

1.1.7 例2 formula で検定

例1と同じデータをデータフレームにして formula を用いて検定

```
> x <- c(1,2,3,2,3,4,3,3,2,2,1,2,3)
> y <- c(2,1,3,4,3,2,2,3,4,5,3,1,2)
> df <- data.frame(data=c(x, y), group=rep(1:2, each=13))

> ansari.test(data ~ group, df)

      Ansari-Bradley test

data:  data by group
AB = 98, p-value = 0.3725
alternative hypothesis: true ratio of scales is not equal to 1

Warning message:
タイがあるため、正確な p 値を計算することができません
  in: ansari.test.default(x = c(1, 2, 3, 2, 3, 4, 3, 3, 2, 2, 1, 2,
> var.test(data ~ group, df) # 対応するパラメトリック検定

      F test to compare two variances

data:  data by group
F = 0.5413, num df = 12, denom df = 12, p-value = 0.3014
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1651628 1.7739390
sample estimates:
ratio of variances
 0.5412844
```

1.2 分散同一性の Fligner-Killeen(メディアン) 検定

各標本群の分散が同一であるという帰無仮説に対する Fligner-Killeen (メディアン) 検定を行う。

```
fligner.test(x, ...)  
fligner.test(x, g, ...)  
fligner.test(formula, data, subset, na.action, ...)
```

1.2.1 引数

x	データベクトルまたはリスト
g	データベクトルまたは factor オブジェクト (x がリストの場合には無視される)
formula	「左辺値 ~ 右辺値」の形をしたモデル式 左辺値は尺度母数を検定する数値変数 右辺値は水準を表す factor 変数
data	モデル式に出てくる変数がデータフレームに含まれるなら、 それを含むデータフレーム名
subset	観察データのサブセット
na.action	データに NA が含まれるときに適用される関数 (注) デフォルトは <code>getOption("na.action")</code> により示されるもの (通常は <code>na.omit</code> 関数)
...	その他の引数

1.2.2 戻り値

1.2.3 関数からの出力

```
$ statistic : 検定統計量 (カイ二乗分布に従う)  
$ parameter : カイ二乗分布の自由度  
$ p.value   : P 値  
$ method    : 検定の種別  
$ data.name : データの記述
```

1.2.4 例 3 つの方法で同じデータを分析してみる

```
> x <- c(1,2,3,2,3,4,3,2)
> y <- c(3,2,4,3,4,4,5,7)
> z <- c(4,5,4,5,6,7,6)

> fligner.test(list(x, y, z))           # リストを使う

      Fligner-Killeen test of homogeneity of variances

data:  list(x, y, z)
Fligner-Killeen:med chi-squared = 0.443, df = 2, p-value = 0.8013

> bartlett.test(list(x, y, z))         # 対応するパラメトリック検定

      Bartlett test of homogeneity of variances

data:  list(x, y, z)
Bartlett's K-squared = 1.6265, df = 2, p-value = 0.4434

> d <- c(x, y, z)
> g <- rep(1:3, c(8, 8, 7))
> fligner.test(d, g)                   # データベクトルと factor を使う

      Fligner-Killeen test of homogeneity of variances

data:  d and g
Fligner-Killeen:med chi-squared = 0.443, df = 2, p-value = 0.8013

> bartlett.test(d, g)                  # 対応するパラメトリック検定

      Bartlett test of homogeneity of variances

data:  d and g
Bartlett's K-squared = 1.6265, df = 2, p-value = 0.4434

> df <- data.frame(data=c(x, y, z), group=rep(1:3, c(8,8,7)))
> fligner.test(data ~ group, df)       # データフレームを使う

      Fligner-Killeen test of homogeneity of variances

data:  data by group
Fligner-Killeen:med chi-squared = 0.443, df = 2, p-value = 0.8013
```

```
> bartlett.test(data ~ group, df)      # 対応するパラメトリック検定

      Bartlett test of homogeneity of variances

data:  data by group
Bartlett's K-squared = 1.6265, df = 2, p-value = 0.4434
```

1.3 Friedman の順位和検定

繰りかえしの無いブロックデータに対する Friedman の順位和検定を行う。帰無仮説は、「全ての標本の位置母数は等しい」である。

```
friedman.test(y, ...)
friedman.test(y, groups, blocks, ...)
friedman.test(formula, data, subset, na.action, ...)
```

1.3.1 引数

y	データベクトルまたはデータ行列
groups	y がデータベクトルのときは、y の要素がどのグループに属するかを表す factor オブジェクト factor でない場合には変換される x が行列の場合には無視される
blocks	y がデータベクトルのときは、y の要素がどのブロックに属するかを表す factor オブジェクト factor でない場合には変換される x が行列の場合には無視される
formula	「a ~ b c」の形をしたモデル式 a は位置母数を検定する数値変数 b, c はグループとブロックを表す factor 変数
data	モデル式に出てくる変数がデータフレームに含まれる者ならば、それを含むデータフレーム名
subset	観察データのサブセット
na.action	データに NA が含まれるときに適用される関数 (注) デフォルトは <code>getOption("na.action")</code> により示されるもの (通常は <code>na.omit</code> 関数)
...	その他の引数

1.3.2 戻り値

1.3.3 関数からの出力

```
$ statistic : 検定統計量 (カイ二乗分布に従う)
$ parameter : カイ二乗分布の自由度
$ p.value   : P 値
$ method    : 検定の種別
$ data.name : データの記述
```

1.3.4 例 3つの方法で同じデータを分析してみる

```
> y <- matrix(c(8.8, 9.4, 9.9, 8.8, 9.4, 10.2, 10.1, 10.8, 10, 10.5,
+ 11.1, 8.7, 8.2, 10.3, 9.7), 5, 3)
> y
      [,1] [,2] [,3] # 3 グループ, 5 ブロックのデータ
[1,]  8.8 10.2 11.1
[2,]  9.4 10.1  8.7
[3,]  9.9 10.8  8.2
[4,]  8.8 10.0 10.3
[5,]  9.4 10.5  9.7
> friedman.test(y) # データ行列を使う場合

      Friedman rank sum test

data:  y
Friedman chi-squared = 3.6, df = 2, p-value = 0.1653

> d <- c(8.8, 9.4, 9.9, 8.8, 9.4, 10.2, 10.1, 10.8, 10, 10.5,
+ 11.1, 8.7, 8.2, 10.3, 9.7)
> group <- rep(1:3, each=5)
> block <- rep(1:5, 3)
> group
[1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
> block
[1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```



```
> friedman.test(d, group, block) # 3 つのデータベクトルを使う場合

Friedman rank sum test

data:  d, group and block
Friedman chi-squared = 3.6, df = 2, p-value = 0.1653

> df <- data.frame(data=d, gr= group, bl=block)
> friedman.test(data ~ group | block, df) # データフレームを使う場合

Friedman rank sum test

data:  data and group and block
Friedman chi-squared = 3.6, df = 2, p-value = 0.1653
```

1.4 Kruskal-Wallis の順位和検定

Kruskal-Wallis の順位和検定を実行する。独立二標本の `wilcox.test` を、3 標本以上に拡張した検定である。帰無仮説は、「全ての標本の位置母数は等しい」である。

```
kruskal.test(x, ...)
kruskal.test(x, g, ...)
kruskal.test(formula, data, subset, na.action, ...)
```

1.4.1 引数

<code>x</code>	データベクトルまたはリスト
<code>g</code>	データベクトルまたは <code>factor</code> オブジェクト (<code>x</code> がリストの場合には無視される)
<code>formula</code>	「左辺値 ~ 右辺値」の形をしたモデル式 左辺値は位置母数を検定する数値変数 右辺値は水準を表す <code>factor</code> 変数
<code>data</code>	モデル式に出てくる変数がデータフレームに含まれる場合には、 それを含むデータフレーム名
<code>subset</code>	観察データのサブセット
<code>na.action</code>	データに <code>NA</code> が含まれるときに適用される関数 (注) デフォルトは <code>getOption("na.action")</code> により示されるもの (通常は <code>na.omit</code> 関数)
<code>...</code>	その他の引数

1.4.2 戻り値

1.4.3 関数からの出力

```
$ statistic : 検定統計量 (カイ二乗分布に従う)
$ parameter : カイ二乗分布の自由度
$ p.value   : P 値
$ method    : 検定の種別
$ data.name : データの記述
```

1.4.4 例 3つの方法で同じデータを分析してみる

```
> x <- c(1,2,3,2,3,4,3,2)
> y <- c(3,2,4,3,4,4,5,7)
> z <- c(4,5,4,5,6,7,6)
> kruskal.test(list(x, y, z))           # リストを使う

      Kruskal-Wallis rank sum test

data:  list(x, y, z)
Kruskal-Wallis chi-squared = 12.0914, df = 2, p-value = 0.002368

> d <- c(x, y, z)
> g <- rep(1:3, c(8, 8, 7))
> kruskal.test(d, g)                   # データベクトルと factor を使う

      Kruskal-Wallis rank sum test

data:  d and g
Kruskal-Wallis chi-squared = 12.0914, df = 2, p-value = 0.002368

> df <- data.frame(data=c(x, y, z), group=rep(1:3, c(8,8,7)))
> kruskal.test(data ~ group, df)       # データフレームを使う

      Kruskal-Wallis rank sum test

data:  data by group
Kruskal-Wallis chi-squared = 12.0914, df = 2, p-value = 0.002368
```

1.5 分布の同一性に対する Kolmogorov-Smirnov 検定

一標本コルモゴロフ・スミルノフ検定または二標本コルモゴロフ・スミルノフ検定を行う。

```
ks.test(x, y, ..., alternative = c("two.sided", "less", "greater"),
        exact = NULL)
```

1.5.1 引数

x	データベクトル
y	一標本検定の場合には分布を表す文字列 "pnorm", "punif" など "p*" 二標本検定の場合にはデータベクトル
...	一標本検定で y で分布を指定したとき, その分布のパラメータ
alternative	対立仮説の種類 "two.sided", "less", "greater" のいずれか 最初の 1 文字だけでもよい 両側検定なら "two.sided" (デフォルト) 片側検定なら "less" または "greater"
exact	NULL または 正確な P 値を計算するかどうかを TRUE か FALSE で指定する 二標本検定の片側検定や同値がある場合には正確な P 値は計算できない NULL を指定しておく, 一標本検定でサンプルサイズが 100 未満の場合, および, 二標本検定の両側検定で両群のサンプルサイズの積が 10000 未満の 場合には正確な P 値が計算される

1.5.2 戻り値

1.5.3 関数への入力

```
$ alternative: 対立仮説の種類
```

1.5.4 関数からの出力

```
$ statistic : 検定統計量
$ p.value   : P 値
$ method    : 検定の種別
$ data.name : データの記述
```

1.5.5 例1 一標本検定

10 ~ 20 に分布する一様乱数を生成した。このデータが実際に一様分布しているかどうか検定する。

```
> x <- c(15.01, 19.7, 17.88, 19.02, 18.14, 18, 11.16, 17.17, 11.02,
+ 16.86, 15.72, 13.43, 10.96, 17.42, 14.23)
> ks.test(x, "punif", min=10, max=20) # 10 ~ 20 の範囲の一様分布か?

One-sample Kolmogorov-Smirnov test

data:  x
D = 0.2193, p-value = 0.4073 # もっともらしい
alternative hypothesis: two.sided

> ks.test(x, "pnorm", mean=15, sd=2.5) # 平均値 15, 標準偏差 2.5 の正規分布か?

One-sample Kolmogorov-Smirnov test

data:  x
D = 0.3049, p-value = 0.09821 # これも否定はできない
alternative hypothesis: two.sided
```

1.5.6 例2 二標本検定

```
> x <- c(1,2,3,2,1,2,3,4,3,4,5,3,4,5,3,2)
> y <- c(3,2,3,4,5,3,2,2,2,2,2,3,4,5)
> ks.test(x, y)

Two-sample Kolmogorov-Smirnov test

data:  x and y
D = 0.125, p-value = 0.9998
alternative hypothesis: two.sided

Warning message:
タイがあるため、正しい p 値を計算することができません in: ks.test(x, y)
注: 上の警告メッセージは正確ではない。
「正確な P 値を計算できない」ということであって、
誤った P 値を計算しているわけではない。
```

1.6 二次元分割表の対称性に対する McNemar 検定

二次元分割表の行と列の間の対称性に対する McNemar のカイ自乗検定を行う。狭義のマクネマー検定は 2×2 分割表の場合である。R ではそれを 3×3 以上の分割表に拡張した対称性の適合度検定 (goodness of fit test of symmetry) をカバーしている。なお、SPSS がサポートしているのは R とは別の拡張である。

```
mcnemar.test(x, y = NULL, correct = TRUE)
```

1.6.1 引数

x	二次元行列の分割表, または factor オブジェクト
y	factor オブジェクト (x が行列の場合には無視される)
correct	2×2 分割表のときにのみ有効であり, 連続性の補正を行うかどうかを指定する
	デフォルトは TRUE (連続性の補正を行う) である

1.6.2 戻り値

1.6.3 関数からの出力

```
$ statistic : 検定統計量 (カイ二乗分布に従う)
$ parameter : カイ二乗分布の自由度
$ p.value   : P 値
$ method    : 検定の種別
$ data.name : データの記述
```

1.6.4 例 1 狭義のマクネマー検定

狭義のマクネマー検定は 2×2 分割表についてのものである

```
> x <- c(2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 2, 1, 2)
> y <- c(2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 1, 1, 2, 1, 2, 2, 1)
> mcnemar.test(x, y) # 2 つのデータベクトルを与える場合
```

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

```
data: x and y
```

```
McNemar's chi-squared = 0.3636, df = 1, p-value = 0.5465
```

```
> z <- table(x, y)
> z
      y
x  1 2
  1 3 7
  2 4 6
> mcnemar.test(z) # 行列として与える場合

McNemar's Chi-squared test with continuity correction

data:  z
McNemar's chi-squared = 0.3636, df = 1, p-value = 0.5465

> binom.test(4, 11)      # 2×2 分割表のときのマクネマー検定は近似検定なので,
                        # 常に binom.test を行うのがよい

Exact binomial test

data:  4 and 11
number of successes = 4, number of trials = 11, p-value = 0.5488
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.1092634 0.6920953
sample estimates:
probability of success
      0.3636364
```

1.6.5 例2 拡張されたマクネマー検定

```
> x <- matrix(c(9, 3, 1, 4, 12, 5, 2, 4, 13), 3, 3)
> x
      [,1] [,2] [,3]
[1,]    9    4    2
[2,]    3   12    4
[3,]    1    5   13
> mcnemar.test(x)

McNemar's Chi-squared test

data:  x
McNemar's chi-squared = 0.5873, df = 3, p-value = 0.8993
```

1.7 二標本のスケールパラメータ（散布度）の差についての Mood 検定

二標本のスケールパラメータ（散布度）の差についての Mood 検定を行う。

```
mood.test(x, y, alternative = c("two.sided", "less", "greater"), ...)  
mood.test(formula, data, subset, na.action, ...)
```

1.7.1 引数

x, y	2 つのデータベクトル
alternarive	対立仮説の種類 "two.sided", "less", "greater" のいずれか 最初の 1 文字だけでもよい 両側検定なら "two.sided" (デフォルト) 片側検定なら "less" または "greater"
formula	「左辺値 ~ 右辺値」の形をしたモデル式 左辺値は尺度母数を検定する数値変数 右辺値は 2 群を表す 2 つの水準を持つ factor 変数
data	モデル式に出てくる変数がデータフレームに含まれる場合は、 それを含むデータフレーム名
subset	観察データのサブセット
na.action	データに NA が含まれるときに適用される関数 (注) デフォルトは <code>getOption("na.action")</code> により示されるもの (通常は <code>na.omit</code> 関数)
...	その他の引数

1.7.2 戻り値

1.7.3 関数への入力

```
$ alternarive: 対立仮説の種類
```

1.7.4 関数からの出力

```
$ statistic : 検定統計量
$ p.value   : P 値
$ null.vallue: 尺度パラメータの比。デフォルトでは 1
              (つまり, 尺度パラメータが等しい)
$ conf.int  : 尺度パラメータの比の信頼区間
$ estimate  : 尺度パラメータの比の推定値 (標本値)
$ method    : 検定の種別
$ data.name : データの記述
```

1.7.5 例1 2つのデータベクトルで検定

```
> x <- c(1,2,3,2,3,4,3,3,2,2,1,2,3)
> y <- c(2,1,3,4,3,2,2,3,4,5,3,1,2)

> mood.test(x, y)

      Mood two-sample test of scale

data:  x and y
Z = -1.3523, p-value = 0.1763
alternative hypothesis: two.sided

> ansari.test(x, y) # 別のノンパラメトリック検定

      Ansari-Bradley test

data:  x and y
AB = 98, p-value = 0.3725
alternative hypothesis: true ratio of scales is not equal to 1

Warning message:
タイがあるため、正確な p 値を計算することができません
      in: ansari.test.default(x, y)
```



```
> var.test(x, y) # 対応するパラメトリック検定

      F test to compare two variances

data:  x and y
F = 0.5413, num df = 12, denom df = 12, p-value = 0.3014
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1651628 1.7739390
sample estimates:
ratio of variances
 0.5412844
```

1.7.6 例2 formula で検定

例1と同じデータをデータフレームにして formula を用いて検定

```
> x <- c(1,2,3,2,3,4,3,3,2,2,1,2,3)
> y <- c(2,1,3,4,3,2,2,3,4,5,3,1,2)
> df <- data.frame(data=c(x, y), group=rep(1:2, each=13))

> mood.test(data ~ group, df)

      Mood two-sample test of scale

data:  data by group
Z = -1.3523, p-value = 0.1763
alternative hypothesis: two.sided

> ansari.test(data ~ group, df) # 別のノンパラメトリック検定

      Ansari-Bradley test

data:  data by group
AB = 98, p-value = 0.3725
alternative hypothesis: true ratio of scales is not equal to 1

Warning message:
タイがあるため、正確な p 値を計算することができません
in: ansari.test.default(x = c(1, 2, 3, 2, 3, 4, 3, 3, 2, 2, 1, 2,
```

```
> var.test(data ~ group, df) # 対応するパラメトリック検定

      F test to compare two variances

data:  data by group
F = 0.5413, num df = 12, denom df = 12, p-value = 0.3014
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1651628 1.7739390
sample estimates:
ratio of variances
 0.5412844
```

1.8 繰り返しの無いブロックデータに対する Quade 検定

繰り返しの無いブロックデータに対する Quade 検定を実行する。

```
quade.test(y, ...)
## 既定の S3 メソッド:
quade.test(y, groups, blocks, ...)
## クラス "formula" に対する S3 メソッド:
quade.test(formula, data, subset, na.action, ...)
```

1.9 正規性に対する Shapiro-Wilk 検定

正規性に対する Shapiro-Wilk 検定を行う。

```
shapiro.test(x)
```

1.10 Wilcoxon の順位和検定

1. ウィルコクソンの符号付順位和検定を行う。
2. ウィルコクソンの符号順位検定を行う。これは、マン・ホイットニー検定と等価な結果を与える。

注意：名前が似ているので紛らわしい。(1) は一標本または対応のある二標本の場合の検定である。対応のあるデータの差または一標本データから分布の中心の値を引いた値に順位を付け、符号ごとに和を取る。(2) は、二標本の場合の検定であり、データをまとめて（プールして）順位を付け標本ごとに順位の和をとる。

```
wilcox.test(x, y = NULL, alternative = c("two.sided", "less", "greater"),
            mu = 0, paired = FALSE, exact = NULL, correct = TRUE,
            conf.int = FALSE, conf.level = 0.95, ...)
wilcox.test(formula, data, subset, na.action, ...)
```

exactRankTests ライブラリ

```
wilcox.exact(x, y = NULL, alternative = c("two.sided", "less", "greater"),
            mu = 0, paired = FALSE,
            exact = NULL, conf.int = FALSE, conf.level = 0.95, ...)
wilcox.exact(formula, data, subset, na.action, ...)
```

1.10.1 引数

x	データベクトル
y	データベクトル（二標本データの場合） x だけが与えられた場合、または x, y が与えられて paired=TRUE が指定された場合は、 ウィルコクソンの符号付順位和検定をおこなう（mu を中心にして左右対称であることの検定） x, y が与えられたときは、その差を取って利用するわけだ。だから、差だけを与えてもよいわけだ。
alternative	対立仮説の種類 "two.sided", "less", "greater" のいずれか 最初の 1 文字だけでもよい 両側検定なら "two.sided"（デフォルト） 片側検定なら "less" または "greater"
mu	一標本の場合は位置パラメータ 二標本の場合は検定する位置の差 デフォルトでは 0
paired	対応のある二標本データの場合に TRUE を指定する デフォルトでは FALSE

<code>exact</code>	正確な P 値を計算する場合に <code>TRUE</code> を指定する デフォルトでは <code>NULL</code> であるが、データ数が 50 未満で、同順位がない場合には内部処理により <code>TRUE</code> を指定したのと同じになる (正確な P 値が計算される) <code>wilcox.exact</code> では、同順位があっても正確な P 値が計算される
<code>correct</code>	連続性の補正をしない場合に <code>FALSE</code> を指定する デフォルトでは <code>TRUE</code>
<code>conf.int</code>	一標本データの場合の疑似中央値、二標本データの場合の位置の差の信頼限界を計算する場合に <code>TRUE</code> を指定する デフォルトでは <code>FALSE</code>
<code>conf.level</code>	信頼率 デフォルトは 0.95
<code>formula</code>	「左辺値 ~ 右辺値」の形をしたモデル式 左辺値は位置母数を検定する数値変数 右辺値は群を表す 2 つの水準を持つ <code>factor</code> 変数 <code>formula</code> で指定する使用法では、二標本データのみをサポートする
<code>data</code>	モデル式に出てくる変数がデータフレームに含まれる場合は、それを含むデータフレーム名
<code>subset</code>	観察データのサブセット
<code>na.action</code>	データに <code>NA</code> が含まれるときに適用される関数 (注) デフォルトは <code>getOption("na.action")</code> により示されるもの (通常は <code>na.omit</code> 関数)
<code>...</code>	その他の引数

1.10.2 戻り値

1.10.3 関数への入力

```
$ alternarive: 対立仮説の種類
$ null.value : 引数で指定した mu
```

1.10.4 関数からの出力

```
$ statistic : 検定統計量
$ parameter : 検定統計量のパラメータ
$ p.value   : P 値
$ conf.int  : 位置母数の信頼区間
$ estimate  : 位置母数の推定値
$ method    : 検定の種別
$ data.name : データの記述
```

1.10.5 例1 一標本データ $\mu=5$ で対称か

```
> x <- c(5, 3.9, 6.1, 5.6, 7, 4.9, 4.5, 6.6, 4.7, 4.6)
> wilcox.test(x, mu=5, conf.int=TRUE) # データから mu を引いて符号付順位和検定

Wilcoxon signed rank test with continuity correction

data: x
V = 28, p-value = 0.5536
alternative hypothesis: true mu is not equal to 5
95 percent confidence interval:
 4.499986 6.100022
sample estimates:
(pseudo)median # 疑似中央値
 5.292418
```

Warning messages:

- 1: ゼロ値のため、正確な p 値を計算することができません
in: wilcox.test.default(x, mu = 5, conf.int = TRUE)
- 2: ゼロ値のため、正確な信頼区間を計算することができません
in: wilcox.test.default(x, mu = 5, conf.int = TRUE)

注:「ゼロ値」というのは、差を取って 0 になるデータがあると、そのデータは有効な情報を与えないため検定から外すが、標本サイズをどのように調整したらよいかわからないというような感じかもしれない。しかし、上のデータから、事前に μ と等しいデータを除いてから検定すると

```
> x <- c(3.9, 6.1, 5.6, 7, 4.9, 4.5, 6.6, 4.7, 4.6)
> wilcox.test(x, mu=5, conf.int=TRUE)
```

Wilcoxon signed rank test

```
data: x
V = 28, p-value = 0.5703
alternative hypothesis: true mu is not equal to 5
95 percent confidence interval:
 4.5 6.3
sample estimates:
(pseudo)median
 5.3
```

となるのだから、前もって 0 のデータを除いてから検定すればいいだけなのじゃないかなあ？

1.10.6 例2 対応のある二標本データ (差は $\mu=0$ か)

```
> x <- c(1,2,3,2,3,4,3,3,2,2,1,2,3)
> y <- c(2,1,3,4,3,2,2,3,4,5,3,1,2)

> wilcox.test(x, y, paired=TRUE) # 2つのデータベクトルを与え,
                                paired=TRUE を指定する

Wilcoxon signed rank test with continuity correction

data: x and y
V = 19.5, p-value = 0.4355
alternative hypothesis: true mu is not equal to 0

Warning messages:
1: タイがあるため、正確な p 値を計算することができません
   in: wilcox.test.default(x, y, paired = TRUE)
2: ゼロ値のため、正確な p 値を計算することができません
   in: wilcox.test.default(x, y, paired = TRUE)

> library(exactRankTests)
> wilcox.exact(x, y, paired=TRUE) # 正確な P 値を求めたいなら exactRankTest ライブラリを

Exact Wilcoxon signed rank test

data: x and y
V = 19.5, p-value = 0.4336
alternative hypothesis: true mu is not equal to 0

> wilcox.test(x-y) # 引き算をした結果が入っている 1 つのデータベクトルを与える

Wilcoxon signed rank test with continuity correction

data: x - y
V = 19.5, p-value = 0.4355 # 当然であるが、同じ結果になる
alternative hypothesis: true mu is not equal to 0

Warning messages:
1: タイがあるため、正確な p 値を計算することができません
   in: wilcox.test.default(x - y)
2: ゼロ値のため、正確な p 値を計算することができません
   in: wilcox.test.default(x - y)
```

```
> wilcox.exact(x-y)

      Exact Wilcoxon signed rank test

data:  x - y
V = 19.5, p-value = 0.4336
alternative hypothesis: true mu is not equal to 0
```

1.10.7 例3 対応のない二標本データ（差は $\mu=0$ か）

wilcox.test という名前でありながら、統計量の名前も W でありながら、計算される検定統計量の数値はマン・ホイットニー検定で定義する統計量である

```
> x <- c(1,2,3,2,3,4,3,3,2,2,1,2,3)
> y <- c(2,1,3,4,3,2,2,3,4,5,3,1,2)

> wilcox.test(x, y) # 対応のない二標本データとして検定

      Wilcoxon rank sum test with continuity correction

data:  x and y
W = 73, p-value = 0.555
alternative hypothesis: true mu is not equal to 0

Warning message:
タイがあるため、正確な p 値を計算することができません
  in: wilcox.test.default(x, y)

> wilcox.exact(x, y) # 正確な P 値が欲しいとき

      Exact Wilcoxon rank sum test

data:  x and y
W = 73, p-value = 0.5863
alternative hypothesis: true mu is not equal to 0

> x <- c(1,2,3,2,3,4,3,3,2,2,1,2,3)
> y <- c(2,1,3,4,3,2,2,3,4,5,3,1,2)
> df <- data.frame(data=c(x, y), group=rep(1:2, each=13))
```

```
> wilcox.test(data ~ group, df) # formula を使って、対応のない二標本データとして  
検定
```

```
Wilcoxon rank sum test with continuity correction
```

```
data: data by group
```

```
W = 73, p-value = 0.555
```

```
alternative hypothesis: true mu is not equal to 0
```

```
Warning message:
```

```
タイがあるため、正確な p 値を計算することができません
```

```
in: wilcox.test.default(x = c(1, 2, 3, 2, 3, 4, 3, 3, 2, 2, 1, 2,
```

1.10.8 例4 データ数が多いときに exact=FALSE を指定してみる

```
> set.seed(11111) # これは、追試のときに同じ結果が出るようにとのおまじない
```

```
> x <- rnorm(50)
```

```
> y <- rnorm(50, mean=1)
```

```
> wilcox.test(x, y)
```

```
Wilcoxon rank sum test with continuity correction
```

```
data: x and y
```

```
W = 661, p-value = 4.97e-05
```

```
alternative hypothesis: true mu is not equal to 0
```

```
> wilcox.test(x, y, exact=TRUE)
```

```
Wilcoxon rank sum test
```

```
data: x and y
```

```
W = 661, p-value = 3.355e-05 # 確かに違う (差は問題にならないが)
```

```
alternative hypothesis: true mu is not equal to 0
```


2 パラメトリック検定関数

2.1 多標本の分散の同一性に対する Bartlett 検定

「幾つかの標本の分散が同一」という帰無仮説を検定する.

```
bartlett.test(x, ...)  
bartlett.test(x, g, ...)  
bartlett.test(formula, data, subset, na.action, ...)
```

2.1.1 引数

x	データベクトルまたはリスト
g	データベクトルまたは <code>factor</code> オブジェクト (x がリストの場合には無視される)
formula	「左辺値 ~ 右辺値」の形をしたモデル式 左辺値は母分散を検定する数値変数 右辺値は水準を表す <code>factor</code> 変数
data	モデル式に出てくる変数がデータフレームに含まれるなら、 それを含むデータフレーム名
subset	観察データのサブセット
na.action	データに <code>NA</code> が含まれるときに適用される関数 (注) デフォルトは <code>getOption("na.action")</code> により示されるもの (通常は <code>na.omit</code> 関数)
...	その他の引数

2.1.2 戻り値

2.1.3 関数からの出力

```
$ statistic : 検定統計量 (カイ二乗分布に従う)  
$ parameter : カイ二乗分布の自由度  
$ p.value   : P 値  
$ method    : 検定の種別  
$ data.name : データの記述
```

2.1.4 例 3つの方法で同じデータを分析してみる

```
> x <- c(1,2,3,2,3,4,3,2)
> y <- c(3,2,4,3,4,4,5,7)
> z <- c(4,5,4,5,6,7,6)
> bartlett.test(list(x, y, z))          # リストを使う

      Bartlett test of homogeneity of variances

data:  list(x, y, z)
Bartlett's K-squared = 1.6265, df = 2, p-value = 0.4434

> d <- c(x, y, z)
> g <- rep(1:3, c(8, 8, 7))
> bartlett.test(d, g)                  # データベクトルと factor を使う

      Bartlett test of homogeneity of variances

data:  d and g
Bartlett's K-squared = 1.6265, df = 2, p-value = 0.4434

> df <- data.frame(data=c(x, y, z), group=rep(1:3, c(8,8,7)))
> bartlett.test(data ~ group, df)      # データフレームを使う

      Bartlett test of homogeneity of variances

data:  data by group
Bartlett's K-squared = 1.6265, df = 2, p-value = 0.4434
```

2.2 正確な二項検定

ベルヌイ試行の成功確率に関する単純帰無仮説の正確な検定，すなわち母比率の検定を行う。正規近似による検定ではない，正確な検定である。データの大きさ n のうち，注目している特性を持つものの数を x とする。母比率が p であるかどうかを検定するものである。 n が小さいときには，正確な P 値を得るためには二項検定を適用せよと書いてあることがあるが，当然のことながら n が大きくても一向に差し支えない。 n が大きくても同じく正確な P 値を得ることができる。

```
binom.test(x, n, p = 0.5, alternative = c("two.sided", "less", "greater"),
           conf.level = 0.95)
```

2.2.1 引数

<code>x</code>	注目している特性を持つものの数
<code>n</code>	データの大きさ (データの総数)
<code>p</code>	母比率 (帰無仮説における母比率) デフォルトは 0.5
<code>alternative</code>	対立仮説の種類 "two.sided", "less", "greater" のいずれか 両側検定なら "two.sided" 片側検定なら "less" または "greater" デフォルトは "two.sided"
<code>conf.level</code>	信頼率 デフォルトは 0.95

2.2.2 戻り値

2.2.3 関数への入力

```
$ statistic : 特性を持つものの数 (引数 x で指定したもの)
$ parameter : 標本の大きさ (引数 n で指定したもの)
$ null.value : 母比率 (引数 p で指定したもの)
$ alternative: 対立仮説の種類 (引数 alternative で指定したもの)
```

2.2.4 関数からの出力

```
$ p.value      : P 値 (計算結果)
$ conf.int     : 信頼限界値 (計算結果)
$ estimate     : 標本比率 (計算結果)
$ method       : 検定手法名 ("Exact binomial test")
$ data.name    : データの記述
```

2.2.5 例 1

ある映画を見に行ったら、観客は 20 人だった。そのうち男は 6 人であった。この映画は性別により好き嫌いがあるのではないか？

帰無仮説：男女による違いはない (母比率 = 0.5)

対立仮説：男女による違いがある (母比率 < 0.5)

実行例

```
> binom.test(6, 20, p=0.5, alternative="two.sided")

      Exact binomial test

data:  6 and 20
number of successes = 6, number of trials = 20, p-value = 0.1153
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.1189316 0.5427892
sample estimates:
probability of success
                0.3
```

解釈

P 値が 0.1153 であるため、帰無仮説を採択する。つまり、この映画は性別により好き嫌いがあるとはいえない。なお、95% 信頼区間は 0.1189316 ~ 0.5427892 であり、標本比率は 0.3 である。

2.2.6 例 2

500 円硬貨を 20 回投げたら、表が 4 回しか出なかった。この 500 円硬貨は表が出にくい（母比率が 0.5 以下）のではないか？

帰無仮説：表も裏も同じ確率で出る（母比率 = 0.5）

対立仮説：表が出にくい（母比率 < 0.5）

実行例

```
> binom.test(4, 20, p=0.5, alternative="less")

      Exact binomial test

data: 4 and 20
number of successes = 4, number of trials = 20, p-value = 0.005909
alternative hypothesis: true probability of success is less than 0.5
95 percent confidence interval:
 0.0000000 0.4010281
sample estimates:
probability of success
                0.2
```

解釈

P 値が 0.005909 ゆえ、帰無仮説を棄却する。すなわち、この 500 円硬貨は表が出にくい。なお、95% 信頼区間は 0 ~ 0.4010281 であり、標本比率は 0.2 である。

2.3 計数データに対する Pearson のカイ自乗検定

分割表に対するカイ自乗検定を実行する¹。

- クロス集計表の独立性の検定（通称：カイ二乗検定）を行う。
- 単純集計（度数分布表）の適合度の検定を行う。

```
chisq.test(x, y = NULL, correct = TRUE, p = rep(1/length(x), length(x)),
           rescale.p = FALSE, simulate.p.value = FALSE, B = 2000)
```

¹注：カイ二乗検定と呼ぶのは勧められない

2.3.1 引数

<code>x</code>	(1) の場合はクロス集計表を表す行列か二変数のベクトルの片方 (2) の場合は、度数分布を表すベクトル
<code>y</code>	(1) で、 <code>x</code> が行列の場合は省略される、そうでない場合は二変数のベクトルの他方
<code>correct</code>	連続性の補正をしないとき <code>FALSE</code> を指定する デフォルトは <code>TRUE</code> (連続性の補正をする) <code>simulate.p.value=TRUE</code> のときには <code>correct=FALSE</code> として処理される
<code>p</code>	(2) の場合に、各カテゴリーの理論比を指定する 省略されたときには、理論比が一定とされる (<code>rep(1/length(x), length(x))</code>)
<code>rescale.p</code>	<code>TRUE</code> のとき、 <code>p</code> の合計が 1 になるように変換する デフォルトは <code>FALSE</code> これが <code>FALSE</code> で <code>p</code> の合計値が 1 にならないとエラーにする
<code>simulate.p.value</code>	<code>TRUE</code> のときには、モンテ・カルロ法によるシミュレーションを行って P 値を求める 周辺和が固定された分割表をランダムに生成し、そのカイ二乗値が観察されたものよりも小さい分割表の割合を P 値とするものである デフォルトは <code>FALSE</code> (普通の場合：カイ二乗分布を用いる漸近検定)
<code>B</code>	モンテ・カルロ法による場合のシミュレーション回数 デフォルトは 2000 回

2.3.2 戻り値

2.3.3 関数への入力

`$ observed` : 観測度数 (度数)

2.3.4 関数からの出力

`$ statistic` : カイ二乗統計量
`$ parameter` : カイ二乗分布の自由度
`$ p.value` : P 値
`$ expected` : 帰無仮説の下での期待値
`$ residuals` : 残差 (`bserve` - `expected`) / `sqrt(expected)`
`$ method` : 検定の種別
`$ data.name` : データの記述

2.3.5 例1 カイ二乗検定 (独立性の検定, 2×2 分割表)

```
> x <- matrix(c(5,3,2,9),2,2)
> chisq.test(x) # デフォルトでは連続性の補正を行う

Pearson's Chi-squared test with Yates' continuity correction

data: x
X-squared = 2.2368, df = 1, p-value = 0.1348

Warning message: # 期待値が 1 以下になるセルがあるか, 5 以下になるセルが全体の 20% を超える場合にこの警告メッセージが出る
カイ自乗近似は不正確かもしれません in: chisq.test(x)

> chisq.test(x, correct=FALSE) # 連続性の補正を行わないときには明示的に指定する

Pearson's Chi-squared test

data: x
X-squared = 3.9095, df = 1, p-value = 0.04801

Warning message:
カイ自乗近似は不正確かもしれません in: chisq.test(x, correct = FALSE)

> x <- matrix(c(52, 31, 22, 89)) # 期待値について問題がなければ, 警告メッセージは出ない
> chisq.test(x)

Chi-squared test for given probabilities

data: x
X-squared = 54.866, df = 3, p-value = 7.333e-12
```

2.3.6 例2 カイ二乗検定（独立性の検定， 2×2 分割表より大きいとき）

```
> x <- matrix(c(20, 32, 13, 43, 54, 21), 2, 3)
> x
      [,1] [,2] [,3]
[1,]   20   13   54
[2,]   32   43   21
> chisq.test(x)

      Pearson's Chi-squared test
data:  x
X-squared = 32.9978, df = 2, p-value = 6.833e-08
```

2.3.7 例3 カイ二乗検定（2変数のデータをベクトルとして与えるとき）

```
> x <- c(2,1,2,1,1,2,1,2,1,2,3,2,1,2,1,2,3,2,2,1,2,3,1)
> y <- c(2,1,2,2,2,1,1,1,1,1,2,1,2,2,2,1,2,1,2,1,1,2,1)
> chisq.test(x, y)

      Pearson's Chi-squared test
data:  x and y
X-squared = 3.8932, df = 2, p-value = 0.1428

Warning message:
カイ自乗近似は不正確かもしれません in: chisq.test(x, y)

> z <- table(x,y) # 上と同じことは, x, y を table 関数でクロス集計して
> z
      y
x     1 2
  1  5 4
  2  7 4
  3  0 3
> chisq.test(z) # 結果を chisq.test に引き渡すのと同じである

      Pearson's Chi-squared test
data:  z
X-squared = 3.8932, df = 2, p-value = 0.1428

Warning message:
カイ自乗近似は不正確かもしれません in: chisq.test(z)
```


2.3.8 例4 カイ二乗検定 (モンテカルロシミュレーションによる場合)

```
> x <- matrix(c(31,24,52,78), 2, 2)
> chisq.test(x)

Pearson's Chi-squared test with Yates' continuity correction

data: x
X-squared = 3.5483, df = 1, p-value = 0.05961

> chisq.test(x, simulate.p.value=TRUE) # デフォルトは 2000 回のシミュレーション
による結果

Pearson's Chi-squared test with simulated p-value (based on 2000 replicates)

data: x
X-squared = 4.1837, df = NA, p-value = 0.05047

> chisq.test(x, simulate.p.value=TRUE, B=20000) # 20000 回のシミュレーションに
による結果

Pearson's Chi-squared test with simulated p-value (based on 20000 replicates)

data: x
X-squared = 4.1837, df = NA, p-value = 0.04945
```

2.3.9 例5 適合度の検定 (一様性の検定)

サイコロを 60 回振ったら以下のようになった。このサイコロの出目はそれぞれ 1/6 ずつであるといつてよいか。

```
> x <- c(13, 27, 21, 24, 18, 17)
> chisq.test(x) # chisq.test(x, p=rep(1/6, 6)) と同じ

Chi-squared test for given probabilities

data: x
X-squared = 6.4, df = 5, p-value = 0.2692
```

2.3.10 例6 適合度の検定

メンデルの遺伝法則 9:3:3:1 に従うか。

```
> x <- c(312, 98, 113, 40)
> chisq.test(x, p=c(9,3,3,1), rescale.p=TRUE) # 合計して 1 にならない割合を使う
ときには rescale.p=TRUE を指定する

      Chi-squared test for given probabilities

data:  x
X-squared = 1.7934, df = 3, p-value = 0.6164

> chisq.test(x, p=c(9,3,3,1)/16) # 合計して 1 になるように指定すればよい

      Chi-squared test for given probabilities

data:  x
X-squared = 1.7934, df = 3, p-value = 0.6164

> chisq.test(x, p=c(9,3,3,1)) # 合計して 1 にならない場合には, エラーが発生する
以下にエラー chisq.test(x, p = c(9, 3, 3, 1)) :
  確率の総和は 1 でなければなりません
```

2.3.11 例7 適合度の検定 (モンテ・カルロ法による場合)

分割表の場合と異なり, R で書かれたプログラムで実行するために時間がかかる。

```
> chisq.test(c(3,2,5,4), simulate.p.value=TRUE)

      Chi-squared test for given probabilities with simulated p-value (based
on 2000 replicates)

data:  c(3, 2, 5, 4)
X-squared = 1.4286, df = NA, p-value = 0.8041
```

2.4 対になった二標本間の関連／相関の検定

二変数間の相関関係の検定を行う。対になった標本間の関連度の検定を、Pearson の積率相関係数 (通常の意味の相関係数), Kendall の順位相関係数 τ , または Spearman の順位相関係数 ρ を用いて実行する。

```
cor.test(x, ...)
cor.test(x, y, alternative = c("two.sided", "less", "greater"),
        method = c("pearson", "kendall", "spearman"),
        exact = NULL, conf.level = 0.95, ...)
cor.test(formula, data, subset, na.action, ...)
```

2.4.1 引数

<code>x, y</code>	データベクトル
<code>alternarive</code>	対立仮説の種類 "two.sided", "less", "greater" のいずれか 最初の 1 文字だけでもよい 両側検定なら "two.sided" (デフォルト) 片側検定なら "less" または "greater"
<code>method</code>	3 種の相関係数のどれを用いるかを "pearson", "kendall", "spearman" のいずれかで指定する 1 文字でもかまわない
<code>exact</code>	<code>method="kendall"</code> の場合にのみ指定できる, デフォルトでは NULL となっており, データの組数が 50 未満で同値がなければ正確な P 値を求める ヘルプには書かれていないが, プログラムソースを読むと, <code>method="spearman"</code> の場合も, <code>n</code> が 1290 以下なら, 正確な <code>p</code> 値を返すことがわかる もっとも, 同値がある場合には <code>n</code> がこれらより少なくても 正確な P 値ではないというエラーメッセージが出る
<code>conf.level</code>	信頼率 デフォルトでは 95%
<code>formula</code>	「 <code>~ u + v</code> 」の形をしたモデル式 <code>u, v</code> は同じ長さを持つデータでないといけない
<code>data</code>	モデル式に出てくる <code>u, v</code> がデータフレームの変数の場合には, そのデータフレーム名
<code>subset</code>	観察データのサブセット
<code>na.action</code>	データに NA が含まれるときに適用される関数 (注) デフォルトは <code>getOption("na.action")</code> により示されるもの (通常は <code>na.omit</code> 関数)

2.4.2 戻り値

2.4.3 関数への入力

```
$ alternative: 帰無仮説の種類
```

2.4.4 関数からの出力

```
$ statistic : 検定統計量  
$ parameter : t 分布に従う検定統計量の場合はその自由度  
$ p.value   : P 値  
$ estimate  : 標本相関係数  
$ conf.int  : 母相関係数の信頼区間  
$ null.value : 帰無仮説の下での母相関係数 (常に 0 である。  
              すなわち, 無相関検定である)  
$ method    : 検定の種別  
$ data.name : データの記述
```

2.4.5 例 1 3 種類の相関係数について検定してみる

```
> x <- c(3,2,5,4,6,5,7,5,8,7,9)  
> y <- c(1,3,4,2,6,5,8,6,5,2,5)  
  
> cor.test(x, y, method="p") # ピアソンの積率相関係数  
  
Pearson's product-moment correlation  
  
data: x and y  
t = 1.7463, df = 9, p-value = 0.1147  
alternative hypothesis: true correlation is not equal to 0  
95 percent confidence interval:  
 -0.1386361  0.8472623  
sample estimates:  
      cor  
0.503077
```

```
> cor.test(x, y, method="s") # スピアマンの順位相関係数
```

```
      Spearman's rank correlation rho
```

```
data:  x and y
```

```
S = 110.7689, p-value = 0.1203
```

```
alternative hypothesis: true rho is not equal to 0
```

```
sample estimates:
```

```
      rho
```

```
0.4965048
```

```
Warning message:
```

```
タイのため正確な p 値を計算することができません
```

```
in: cor.test.default(x, y, method = "s")
```

```
> cor.test(x, y, method="k") # ケンドールの順位相関係数
```

```
      Kendall's rank correlation tau
```

```
data:  x and y
```

```
z = 1.6063, p-value = 0.1082
```

```
alternative hypothesis: true tau is not equal to 0
```

```
sample estimates:
```

```
      tau
```

```
0.396059
```

```
Warning message:
```

```
タイのため正確な p 値を計算することができません
```

```
in: cor.test.default(x, y, method = "k")
```

2.4.6 例2 formula を用いてみる

```
> x <- c(3,2,5,4,6,5,7,5,8,7,9)
```

```
> y <- c(1,3,4,2,6,5,8,6,5,2,5)
```

```
> df <- data.frame(dx=x, dy=y)
```

```
> cor.test(~ dx + dy, df, method="p") # ピアソンの積率相関係数

Pearson's product-moment correlation

data: dx and dy
t = 1.7463, df = 9, p-value = 0.1147
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1386361  0.8472623
sample estimates:
 cor
0.503077

> cor.test(~ dx + dy, df, method="s") # スピアマンの順位相関係数

Spearman's rank correlation rho

data: dx and dy
S = 110.7689, p-value = 0.1203
alternative hypothesis: true rho is not equal to 0
sample estimates:
 rho
0.4965048

Warning message:
タイのため正確な p 値を計算することができません
in: cor.test.default(x = c(3, 2, 5, 4, 6, 5, 7, 5, 8, 7, 9), ...

> cor.test(~ dx + dy, df, method="k") # ケンドールの順位相関係数

Kendall's rank correlation tau

data: dx and dy
z = 1.6063, p-value = 0.1082
alternative hypothesis: true tau is not equal to 0
sample estimates:
 tau
0.396059

Warning message:
タイのため正確な p 値を計算することができません
in: cor.test.default(x = c(3, 2, 5, 4, 6, 5, 7, 5, 8, 7, 9), ...
```

2.5 計数データに対する Fisher の正確検定

周辺和が固定された分割表の行と列が独立であるという帰無仮説に対する Fisher の正確検定を実行する。2 × 2 分割表の場合には超幾何分布を応用して容易に検定が行われるが、それより大きい分割表だと計算量が大きくなって検定ができない場合がある。

```
fisher.test(x, y = NULL, workspace = 200000, hybrid = FALSE,  
            control = list(), or = 1, alternative = "two.sided",  
            conf.level = 0.95)
```

2.5.1 引数

x	行列の形式で表された二次元分割表, または factor オブジェクト
y	x が factor オブジェクトのとき, もう一つの factor オブジェクト x と y の長さは等しくなければならない 何れか一方または両方が欠損値の場合には取り除かれる x が行列のときには無視される
workspace	ネットワーク・アルゴリズムで用いられる作業領域の大きさ (4 バイト単位)
hybrid	2 × 2 よりも大きな分割表のときにのみ指定できる モンテ・カルロ法により近似的な P 値を求める デフォルトは FALSE
alternarive	対立仮説の種類 "two.sided", "less", "greater" のいずれか 最初の 1 文字だけでもよい 両側検定なら "two.sided" (デフォルト) 片側検定なら "less" または "greater" 指定できるのは, 2 × 2 分割表の場合のみ それ以上の場合にはデフォルトの両側検定のみ
or	帰無仮説でのオッズ比 デフォルトでは 1 (行と列が独立であること, あるいは比率の差が 0 であることと等価) 指定できるのは, 2 × 2 分割表の場合のみ
conf.int	信頼限界値を計算するかどうかを指定する デフォルトでは TRUE 指定できるのは, 2 × 2 分割表の場合のみ
conf.level	信頼率 デフォルトは 0.95 指定できるのは, 2 × 2 分割表の場合のみ

2.5.2 戻り値

2.5.3 関数への入力

```
$ null.value : 引数で指定された or の値  
$ alternarive: 対立仮説の種類
```

2.5.4 関数からの出力

```
$ p.value      : P 値  
$ conf.int     : オッズ比の信頼区間 (2 × 2 分割表の場合のみ)  
$ estimate     : オッズ比の推定値 (条件付最尤推定値であることに注意)  
$ method       : 検定の種別  
$ data.name    : データの記述
```

2.5.5 例 1 2 × 2 分割表の場合

```
# x, y を与える場合      factor オブジェクトでなくても良い (内部で変換される)  
> x <- c("a", "a", "b", "b", "a", "b", "a")  
> y <- c("m", "m", "m", "f", "f", "f", "m")  
> fisher.test(x, y)
```

Fisher's Exact Test for Count Data

```
data: x and y  
p-value = 0.4857  
alternative hypothesis: true odds ratio is not equal to 1  
95 percent confidence interval:  
 0.002141305 8.845590751  
sample estimates:  
odds ratio  
 0.2247395
```

```
# 行列を与える場合      x, y を与える場合でも, 結局は内部で分割表を作っている  
> table(x, y)  
      y  
x     f m  
a     1 3  
b     2 1  
> tbl <- table(x, y)
```



```
> fisher.test(tbl)

      Fisher's Exact Test for Count Data

data:  tbl
p-value = 0.4857
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.002141305 8.845590751
sample estimates:
odds ratio
 0.2247395
```

2.5.6 例2 大きな分割表の場合

```
> x <- matrix(c(3,2,4, 5,4,2, 1,2,4, 5,4,3), 3, 4)
> x
      [,1] [,2] [,3] [,4]
[1,]    3    5    1    5 # このような, 3×4 分割表でもあつという間に計算される
[2,]    2    4    2    4
[3,]    4    2    4    3
> fisher.test(x)

      Fisher's Exact Test for Count Data

data:  x
p-value = 0.6977
alternative hypothesis: two.sided      # 2×2 より大きな分割表では常に両側検定
である
                                         # オッズ比やその信頼限界は計算されない
```

2.5.7 例3 計算量が大きくなり作業領域を大きくする必要がある場合

```
> x <- matrix(c(12,43,25, 12,46, 20, 10,50,32), 3, 3)
> x
      [,1] [,2] [,3]
[1,]   12   12   10
[2,]   43   46   50
[3,]   25   20   32
```

```
> fisher.test(x)
以下にエラー fisher.test(x) : FEXACT error 6. # 作業領域が不足したというエラーメッセージ
LDKEY is too small for this problem.
Try increasing the size of the workspace.
> fisher.test(x, workspace=3000000) # 作業領域を増やしてみる

Fisher's Exact Test for Count Data

data: x
p-value = 0.6955
alternative hypothesis: two.sided
> fisher.test(x, workspace=3000000, hybrid=TRUE) # hybrid=TRUE で検定してみる

Fisher's Exact Test for Count Data

data: x
p-value = 0.6926
alternative hypothesis: two.sided

Warning message:
p 値は正しくないかもしれません
in: fisher.test(x, workspace = 3e+06, hybrid = TRUE)
注:「正しくないかも」というのは言い過ぎで、「正確でないかも」という程度。間違えた P 値ではないのだから
> chisq.test(x) # カイ二乗分布を用いる漸近近似

Pearson's Chi-squared test

data: x
X-squared = 2.1903, df = 4, p-value = 0.7008 # 近似がよいと見るか悪いと見るかは人によるかも
```

2.6 Cochran-Mantel-Haenszel カイ自乗検定

3次の相互作用が無いという仮定の下で、二つの名義尺度変数が各層毎に条件付き独立であるという帰無仮説の Cochran-Mantel-Haenszel カイ自乗検定を実行する。

```
mantelhaen.test(x, y = NULL, z = NULL,
                alternative = c("two.sided", "less", "greater"),
                correct = TRUE, exact = FALSE, conf.level = 0.95)
```

2.7 一元配置正規標本の平均の同一性検定

一元配置の 2 もしくはそれ以上の正規標本の平均の同一性を検定する。分散は必ずしも同一でなくても良い。

```
oneway.test(formula, data, subset, na.action, var.equal = FALSE)
```

2.7.1 引数

formula	左辺値 ~ 右辺値」の形をしたモデル式 左辺値は平均値を検定する数値変数 右辺値は群の水準をを表す変数 必ずしも factor でなくてもよいようだが、他の関数との関連もあるので、factor にしておくのが無難
data	モデル式に出てくる変数がデータフレームに含まれる場合は、それを含むデータフレーム名
subset	オプション。観察データのサブセット
na.action	データに NA が含まれるときに適用される関数 (注) デフォルトは <code>getOption("na.action")</code> により示されるもの (通常は <code>na.omit</code> 関数)
var.equal	等分散を仮定する場合に TRUE を指定する (教科書に多く取り上げられている方法) デフォルトは FALSE になっているので注意が必要 (独立二標本のときの Welch の方法の拡張)

2.7.2 戻り値

```
$ statistic : F 統計量  
$ parameter : 自由度 (var.equal=FALSE の場合は小数自由度になることがある)  
$ p.value   : P 値  
$ method    : 検定の種別  
$ data.name : データの記述
```

2.7.3 例

```
> x <- c(1,2,3,2,1,2,3,4)      # 8 個のデータ  
> y <- c(2,1,2,3,2,2,1)       # 7 個のデータ  
> z <- c(3,2,1,2,3,4,5)       # 7 個のデータ  
> d <- c(x, y, z)             # 一つにまとめる  
> d
```

```
[1] 1 2 3 2 1 2 3 4 2 1 2 3 2 2 1 3 2 1 2 3 4 5
> g <- factor(rep(1:3, c(8, 7, 7)))      # グループを表す factor を作る
> g
[1] 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3
Levels: 1 2 3      # 数値データのように見えるが実は factor

> oneway.test(d ~ g)      # デフォルトでは等分散を仮定しない検定

      One-way analysis of means (not assuming equal variances)

data:  d and g
F = 1.5355, num df = 2.000, denom df = 11.848, p-value = 0.2553

> oneway.test(d ~ g, var.equal=TRUE)    # 普通の教科書に載っている等分散を仮
定する検定

      One-way analysis of means

data:  d and g
F = 1.5935, num df = 2, denom df = 19, p-value = 0.2292
```

2.8 複数のグループの比率の同一性の検定

複数のグループの比率(成功確率)が等しい, または与えられた値に等しいという帰無仮説を検定する.

- 複数の母比率が等しいかどうか検定する。
- 母比率がある値であるかどうか検定する。(一標本とは限らない)

```
prop.test(x, n, p = NULL,
          alternative = c("two.sided", "less", "greater"),
          conf.level = 0.95, correct = TRUE)
```

2.8.1 引数

<code>x</code>	注目している特性を持つものの数（成功数）を表すベクトルか、特性を持っているものを持っていないものの数を表す2列の行列
<code>n</code>	各群の標本の大きさを表すベクトル（ <code>x</code> が行列のときには無視される）
<code>p</code>	母比率と等しいかどうか比べられる特定の値（成功比率） 一標本のときのデフォルトは 0.5 複数標本の場合に <code>p</code> を与えるときには、 <code>p</code> の要素数は群の数と同じでなくてはならない (群毎に指定するということ)
<code>alternative</code>	対立仮説の種類 "two.sided", "less", "greater" のいずれか 最初の1文字だけでもよい 両側検定なら "two.sided" (デフォルト) 片側検定なら "less" または "greater" 指定できるのは、一標本の場合と、二標本で <code>p</code> を指定しない場合のみ 三標本以上の場合はデフォルトの両側検定のみ
<code>conf.level</code>	信頼率 デフォルトは 0.95 指定できるのは、一標本の場合と二標本の場合のみ
<code>correct</code>	イエーツの補正をしないときには FALSE を指定する デフォルトは TRUE

2.8.2 戻り値

2.8.3 関数への入力

```
$ null.value : 引数で指定された p の値  
$ alternative: 対立仮説の種類
```

2.8.4 関数からの出力

```
$ statistic : カイ二乗統計量  
$ parameter : カイ二乗分布の自由度  
$ p.value : P 値  
$ estimate : 標本比率  
$ conf.int : 一標本の場合、二標本で p が指定されない比率の差の検定  
$ method : 検定の種別  
$ data.name : データの記述
```

2.8.5 例 1 一標本の場合（母比率の検定）

成功確率 0.5 で、50 施行中 35 回成功した場合の検定。

```
> prop.test(35, 50, p=0.5) # 正規近似によるので、お勧めではない。  
                           常に binom.test を使うべし。  
                           # p=0.5 はデフォルトなので指定しなくても同じ  
1-sample proportions test with continuity correction
```

```
data: 35 out of 50, null probability 0.5  
X-squared = 7.22, df = 1, p-value = 0.00721  
alternative hypothesis: true p is not equal to 0.5  
95 percent confidence interval:  
 0.5521660 0.8171438  
sample estimates:  
 p  
0.7
```

```
> binom.test(35, 50, p=0.5) # 同じことを binom.test でやった場合（こちらが  
正確）
```

```
Exact binomial test
```

```
data: 35 and 50  
number of successes = 35, number of trials = 50, p-value = 0.0066  
alternative hypothesis: true probability of success is not equal to 0.5  
95 percent confidence interval:  
 0.5539177 0.8213822  
sample estimates:  
probability of success  
 0.7
```

2.8.6 例2 複数標本で母比率を指定したとき

成功数の期待値を p , 失敗数の期待値を $1-p$ から求め, 後は普通のカイ二乗検定と同じように検定を進めるものである。

```
> prop.test(c(35,30,32), c(50,60,65), p=c(0.5, 0.4, 0.6))

      3-sample test for given proportions without continuity correction

data:  c(35, 30, 32) out of c(50, 60, 65), null probabilities c(0.5, 0.4, 0.6)
X-squared = 13.641, df = 3, p-value = 0.003437
alternative hypothesis: two.sided
null values:
prop 1 prop 2 prop 3
  0.5   0.4   0.6
sample estimates:
  prop 1   prop 2   prop 3
0.7000000 0.5000000 0.4923077
```

2.8.7 例3 二標本の場合 (比率の差の検定)

イエーツの補正がデフォルト, 片側検定も行える, 比率の差の信頼区間も求められる

以下のような 2×2 分割表にまとめられるデータの検定を行う

	群A	群B	合計
成功	35	30	65
失敗	15	30	45
合計	50	60	110

```
> prop.test(c(35,30), c(50,60)) # 各群の成功数と標本サイズを入力する
とき
```

```
      2-sample test for equality of proportions with continuity correction

data:  c(35, 30) out of c(50, 60)
X-squared = 3.7234, df = 1, p-value = 0.05366
alternative hypothesis: two.sided
95 percent confidence interval:
 0.002389771 0.397610229
sample estimates:
prop 1 prop 2
  0.7   0.5
```

```
> prop.test(matrix(c(35, 30, 15, 30), 2,2)) # 成功数の列と失敗数の列からなる行
列を入力するとき

      2-sample test for equality of proportions with continuity correction

data:  matrix(c(35, 30, 15, 30), 2, 2)
X-squared = 3.7234, df = 1, p-value = 0.05366      # 当然ながら上と同じ結果に
なる
alternative hypothesis: two.sided
95 percent confidence interval:
 0.002389771 0.397610229
sample estimates:
prop 1 prop 2
 0.7    0.5

> chisq.test(matrix(c(35,15,30,30),2,2))      # 同じデータを chisq.test で検
定しても（普通は）同じ結果になる

      Pearson's Chi-squared test with Yates' continuity correction

data:  matrix(c(35, 15, 30, 30), 2, 2)
X-squared = 3.7234, df = 1, p-value = 0.05366
```

2.8.8 例4 複数の群の場合の比率の差の検定

```
> prop.test(c(35,30,32), c(50,60,65)) # correct=FALSE を指定しても無視される

      3-sample test for equality of proportions without continuity correction

data:  c(35, 30, 32) out of c(50, 60, 65)
X-squared = 6.0235, df = 2, p-value = 0.0492
alternative hypothesis: two.sided # 常に両側検定になる
sample estimates:
 prop 1    prop 2    prop 3
0.7000000 0.5000000 0.4923077
```


2.9 比率中の傾向に対するカイ自乗検定

比率中の傾向に対するカイ自乗検定を行う。つまり、対数オッズが score に応じて変化するという局所対立仮説に対する漸近的に最適な検定である。既定では score はグループ番号 (1,2,,k) とされる。

```
prop.trend.test(x, n, score = 1:length(x))
```

2.10 Student の t 検定

データベクトルに対し一・二標本 t 検定を行う。

- 独立二標本の平均値の差の検定いわゆる「t 検定」を行う。
 - 二標本の分散が等しいことを仮定する場合
 - 二標本の分散が等しいことを仮定しない場合 (こちらがデフォルトであることに注意)
- 対応のある二標本の平均値の差の検定を行う。
- 一標本の母平均の検定を行う。

```
t.test(x, y = NULL, alternative = c("two.sided", "less", "greater"),  
       mu = 0, paired = FALSE, var.equal = FALSE, conf.level = 0.95, ...)  
t.test(formula, data, subset, na.action, ...)
```

2.10.1 引数

x	データを表すベクトル
y	データを表すベクトル (一標本の検定の場合には省略)
alternative	対立仮説の種類 "two.sided", "less", "greater" のいずれか 最初の 1 文字だけでもよい 両側検定なら "two.sided" 片側検定なら "less" または "greater" デフォルトは "two.sided"
mu	二標本検定の場合には母平均の差 一標本検定の場合には母平均の値
paired	対応のある二標本の母平均の差の検定を行う場合に TRUE を指定する デフォルトは FALSE になっている
var.equal	独立二標本検定の場合で等分散を仮定する場合に TRUE を指定する デフォルトは FALSE になっている (Welch の方法による t 検定が行われる)

<code>conf.level</code>	信頼率 デフォルトは 0.95
<code>formula</code>	「左辺値 ~ 右辺値」の形をしたモデル式 左辺値は平均値を検定する数値変数 右辺値は2群を表す2つの水準をもつ <code>factor</code> 変数
<code>data</code>	モデル式に出てくる変数がデータフレームに含まれる場合はそれ、 を含むデータフレーム名
<code>subset</code>	観察データのサブセット
<code>na.action</code>	データに NA が含まれるときに適用される関数 (注) デフォルトは <code>getOption("na.action")</code> により示されるもの (通常は <code>na.omit</code> 関数)
<code>...</code>	その他の引数

2.10.2 戻り値

2.10.3 関数への入力

<code>\$ null.value</code>	: 母平均の差または母平均 (<code>mu</code> で指定したもの)
<code>\$ alternative</code>	: 対立仮説の種類

2.10.4 関数からの出力

<code>\$ statistic</code>	: t 統計量
<code>\$ parameter</code>	: 自由度 (Welch の方法の場合は小数自由度になることがある)
<code>\$ p.value</code>	: P 値
<code>\$ conf.int</code>	: 母平均の差または母平均の信頼限界値
<code>\$ estimate</code>	: 標本平均 (母平均の推定値)
<code>\$ method</code>	: 検定の種別
<code>\$ data.name</code>	: データの記述

2.10.5 例1 独立二標本の平均値の差の検定；等分散を仮定しない場合—Welch の方法

二群のデータを、二つのベクトル (x, y など) として用意して、`t.test(x, y)` などとすればよいだけである。データフレームとして用意されているときは、例7も参照のこと。

```
> x <- c(2,3,2,1,3,5,6)
> y <- c(3,2,4,3,6,7,9,6)
> t.test(x, y)      # t.test(x, y, var.equal=FALSE) と同じ
```

Welch Two Sample t-test # Welch の方法による検定

```
data:  x and y
t = -1.7219, df = 12.708, p-value = 0.1093 # 小数自由度の P 値が計算される
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: # 95% 信頼区間
-4.192659  0.478373 # 母平均の差の信頼限界値 (下側信頼限界値と上側信頼限界値)
sample estimates: # 標本平均
mean of x mean of y
 3.142857  5.000000
```

引数 `mu` により指定した母平均の差の点推定値は以下のようにすれば出力できる

```
> a <-t.test(x, y)
> a$null.value
difference in means # 母平均の差
          0
```

2.10.6 例2 独立二標本の平均値の差の検定；等分散を仮定する場合

```
> x <- c(2,3,2,1,3,5,6)
> y <- c(3,2,4,3,6,7,9,6)
> t.test(x, y, var.equal=TRUE) # var.equal=TRUE を明示しないとイケない
```

Two Sample t-test # 普通の t 検定が行われた

```
data:  x and y
t = -1.6864, df = 13, p-value = 0.1155
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-4.2362138  0.5219281 # 当然ながら信頼限界値も異なる
sample estimates:
mean of x mean of y
 3.142857  5.000000
```

2.10.7 例3 対応のある二標本の平均値の差の検定

```
> a <- c(1,3,2,4,3,2) # 対応のあるデータ
> b <- c(2,4,1,6,5,4)
> t.test(a, b, paired=TRUE) # paired=TRUE を指定すること

Paired t-test # いわゆる「対応のある t 検定」が行われた

data:  a and b
t = -2.4445, df = 5, p-value = 0.05833
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: # 差の母平均の信頼区間
-2.39350436  0.06017103
sample estimates: 差の標本平均
mean of the differences
-1.166667
```

2.10.8 例4 母平均の検定（一標本）

例3の対応のあるデータの差を取ったデータを使うときは、一標本の検定（母平均の検定）になる（結果は対応のあるデータを使ったときと同じ）。

```
> a <- c(1,3,2,4,3,2) # 対応のあるデータ
> b <- c(2,4,1,6,5,4)
> dif <- a-b
> dif
[1] -1 -1  1 -2 -2 -2
> t.test(dif) # 名前のない引数が 1 つの場合は、一標本検定になる

One Sample t-test # 一標本検定（母平均の検定）である

data:  dif
t = -2.4445, df = 5, p-value = 0.05833
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval: 平均値の差の信頼区間（母数はデフォルトの 0）
-2.39350436  0.06017103
sample estimates:
mean of x
-1.166667
```

2.10.9 例5 母平均の検定 (一標本)

```
> x <- c(9, 9.3, 11.1, 10, 9.1, 10.6, 8.9, 9.6, 8.5, 9.2)
> t.test(x, mu=9) # 母平均が 9 であるかの検定

One Sample t-test # 一標本検定 (母平均の検定) である

data: x
t = 2.0663, df = 9, p-value = 0.06877
alternative hypothesis: true mean is not equal to 9
95 percent confidence interval: # 母平均の信頼区間
 8.949772 10.110228
sample estimates: # 標本平均 (母平均の推定値)
mean of x
 9.53
```

2.10.10 例6 モデル式による指定

例1と同じことをやる別法

```
> x <- c(2,3,2,1,3,5,6)
> y <- c(3,2,4,3,6,7,9,6)
> data <- c(2,3,2,1,3,5,6, 3,2,4,3,6,7,9,6)
> group <- factor(c(rep("male", 7), rep("female", 8)),
+               levels=c("male", "female"))
> group # factor 関数の levels 引数により適切に指定する方が混乱がない
[1] male  male  male  male  male  male  male  female female
[10] female female female female female female
Levels: male female
> t.test(data ~ group)

Welch Two Sample t-test

data: data by group
t = -1.7219, df = 12.708, p-value = 0.1093
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.192659  0.478373
sample estimates:
 mean in group male mean in group female
 3.142857           5.000000
```

2.10.11 例7 モデル式による指定

例1と同じことをやる別法例6と同じだが、データフレーム中の変数を使う場合

```
> df <- read.table("df.dat") # ファイルからデータフレームを読む
> df # 以下のようなデータフレーム (例6と同じ内容)
  data group
1     2  male
2     3  male
3     2  male
4     1  male
5     3  male
6     5  male
7     6  male
8     3 female
9     2 female
10    4 female
11    3 female
12    6 female
13    7 female
14    9 female
15    6 female

> t.test(data ~ group, df) # 第二引数でデータフレーム名を指定する

Welch Two Sample t-test

data: data by group
t = 1.7219, df = 12.708, p-value = 0.1093
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.478373  4.192659
sample estimates:
mean in group female    mean in group male
      5.000000           3.142857
```

2.11 二つの分散の同一性に対する F 検定

正規データからの二標本の分散を比較する F 検定を行う。

```
## 既定の S3 メソッド:  
var.test(x, y, ratio = 1, alternative = c("two.sided", "less", "greater"),  
         conf.level = 0.95, ...)  
## クラス "formula" に対する S3 メソッド:  
var.test(formula, data, subset, na.action, ...)
```

2.11.1 引数

<code>x, y</code>	2 つのデータベクトル, または, <code>lm</code> クラスを持つオブジェクト
<code>ratio</code>	帰無仮説の下での <code>x</code> と <code>y</code> の分散比 デフォルトでは 1 (つまり, 等分散である)
<code>alternarive</code>	対立仮説の種類 <code>"two.sided"</code> , <code>"less"</code> , <code>"greater"</code> のいずれか 最初の 1 文字だけでもよい 両側検定なら <code>"two.sided"</code> (デフォルト) 片側検定なら <code>"less"</code> または <code>"greater"</code>
<code>conf.level</code>	信頼率 デフォルトは 0.95
<code>formula</code>	「左辺値 ~ 右辺値」の形をしたモデル式 左辺値は母分散を検定する数値変数 右辺値は 2 群を表す 2 つの水準を持つ <code>factor</code> 変数
<code>data</code>	モデル式に出てくる変数がデータフレームに含まれる場合は, それを含むデータフレーム名
<code>subset</code>	観察データのサブセット
<code>na.action</code>	データに <code>NA</code> が含まれるときに適用される関数 (注) デフォルトは <code>getOption("na.action")</code> により示されるもの (通常は <code>na.omit</code> 関数)
<code>...</code>	その他の引数

2.11.2 戻り値

2.11.3 関数への入力

```
$ alternarive: 対立仮説の種類
```

2.11.4 関数からの出力

```
$ statistic : F 統計量
$ parameter : F 分布の自由度
$ p.value   : P 値
$ conf.int  : 母分散比の信頼区間
$ estimate  : 標本分散比
$ method    : 検定の種別
$ data.name : データの記述
```

2.11.5 例1 2つのデータベクトルで検定

```
> x <- c(1,2,3,2,3,4,3,3,2,2,1,2,3)
> y <- c(2,1,3,4,3,2,2,3,4,5,3,1,2)
> var.test(x, y)

      F test to compare two variances

data:  x and y
F = 0.5413, num df = 12, denom df = 12, p-value = 0.3014
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1651628 1.7739390
sample estimates:
ratio of variances
 0.5412844

> var.test(lm(x ~ 1), lm(y ~ 1)) # lm クラスがどうか言っているのは、このこと

      F test to compare two variances

data:  lm(x ~ 1) and lm(y ~ 1)
F = 0.5413, num df = 12, denom df = 12, p-value = 0.3014
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1651628 1.7739390
sample estimates:
ratio of variances
 0.5412844
```


2.11.6 例2 formula で検定

例1と同じデータをデータフレームにして formula を用いて検定

```
> x <- c(1,2,3,2,3,4,3,3,2,2,1,2,3)
> y <- c(2,1,3,4,3,2,2,3,4,5,3,1,2)
> df <- data.frame(data=c(x, y), group=rep(1:2, each=13))
```

```
> var.test(data ~ group, df)
```

```
      F test to compare two variances
```

```
data:  data by group
```

```
F = 0.5413, num df = 12, denom df = 12, p-value = 0.3014
```

```
alternative hypothesis: true ratio of variances is not equal to 1
```

```
95 percent confidence interval:
```

```
 0.1651628 1.7739390
```

```
sample estimates:
```

```
ratio of variances
```

```
 0.5412844
```

3 多重比較補正付きの検定関数

3.1 p 値のベクトルの多重比較補正

p 値のベクトルを与えて、幾つかの方法を用いて多重比較補正された p 値のベクトルを返す。

```
p.adjust(p, method=p.adjust.methods, n=length(p))  
## 補正法の名称の文字列ベクトル  
p.adjust.methods
```

3.2 多重比較補正を伴う対毎の比率の比較

多重比較補正を伴う、対毎の比率の比較を行う。

```
pairwise.prop.test(x, n, p.adjust.method = p.adjust.methods, ...)
```

3.3 多重比較補正を伴うグループ水準間の対毎の t 検定による比較

多重比較補正を伴うグループ水準間の、対毎の t 検定による比較を行う。

```
pairwise.t.test(x, g, p.adjust.method = p.adjust.methods,  
                pool.sd = TRUE, ...)
```

3.4 Wilcoxon の順位和検定による、p 値に対する多重比較補正

グループ水準の各組合せ毎に Wilcoxon の順位和検定を実施し、p 値に対する多重比較補正を行う。

```
pairwise.wilcox.test(x, g, p.adjust.method=p.adjust.methods, ...)
```

4 検定の検出力の計算

4.1 一元配置分散分析検定の検出力計算

一元配置分散分析検定の検出力 (パワー) を計算する. または目的の検出力を得るためのパラメータを計算する.

```
power.anova.test(groups = NULL, n = NULL, between.var = NULL,  
                 within.var = NULL, sig.level = 0.05, power = NULL)
```

4.2 比率の検定に対する検出力

比率の検定に対する検出力を計算する. または目標の検出力を得るために必要なパラメータを計算する.

```
power.prop.test(n = NULL, p1 = NULL, p2 = NULL, sig.level = 0.05,  
               power = NULL, alternative = c("two.sided", "one.sided"),  
               strict = FALSE)
```

4.3 t 検定の検出力の計算

一・二標本の t 検定を行う. または, 目標の検出力を得るために必要なパラメータの決定を行う.

```
power.t.test(n=NULL, delta=NULL, sd=1, sig.level=0.05, power=NULL,  
            type=c("two.sample", "one.sample", "paired"),  
            alternative=c("two.sided", "one.sided"), strict=FALSE)
```

4.4 検出力計算オブジェクトのプリントメソッド

検出力計算オブジェクトのプリントメソッド. クラス "power.htest" のオブジェクトを整形して出力する.

```
## クラス "power.htest" に対する S3 メソッド:  
print(x, ...)
```

【参考】NAの扱い

na.omit, na.exclude, na.fail, na.pass

```
> x <- c(1,2,3,2,5,4,3,NA,4)

> na.omit(x)
[1] 1 2 3 2 5 4 3 4          # na.omit は NA を除く
attr(,"na.action")
[1] 8                      # NA のあった場所
attr(,"class")
[1] "omit"                 # クラスは "omit"

> na.exclude(x)
[1] 1 2 3 2 5 4 3 4          # na.exculc は NA を除く
attr(,"na.action")
[1] 8                      # NA のあった場所
attr(,"class")
[1] "exclude"              # クラスは "exclude"

> na.fail(x)                # na.fail は NA があるとエラーにする
以下にエラー na.fail.default(x) : オブジェクト中に欠損値があります

> na.pass(x)
[1] 1 2 3 2 5 4 3 NA 4      # na.pass は NA を素通しする
```