ρ and order: computing p-values of Spearman's rank correlation coefficient ρ with ties.

@fronori

11/9/2017

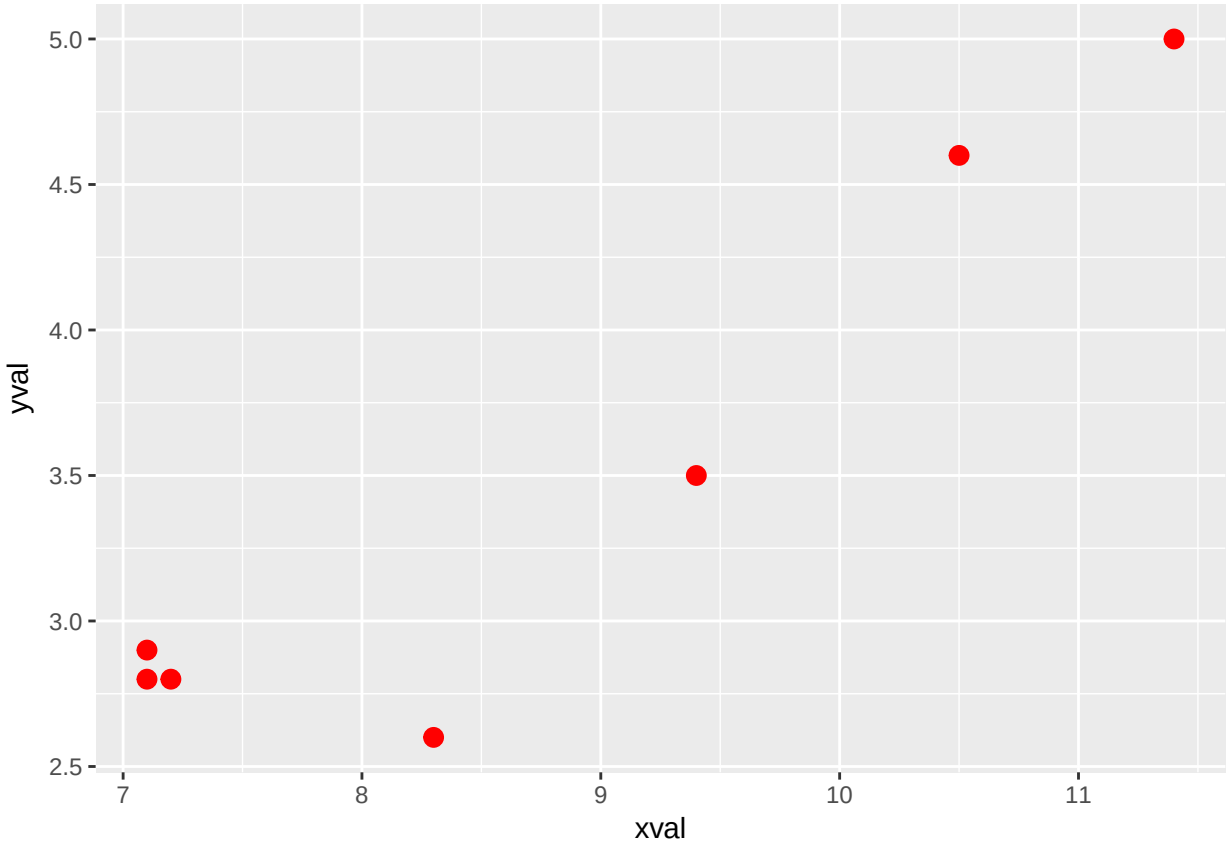
同順位がある場合のスピアマン順位相関のp値について検討する。

たとえばこのような対応のあるデータがあったとする。

```
x <- c(7.1, 7.1, 7.2, 8.3, 9.4, 10.5, 11.4)
y <- c(2.8, 2.9, 2.8, 2.6, 3.5, 4.6, 5.0)
sampledat <- data.frame(xval = x, yval = y)
```

散布図にするとこんな感じ。

```
library(ggplot2)
ggplot(sampledat, aes(x=xval, y=yval)) + geom_point(color = "red", size = 3)
```



このデータのSpearman's rank correlation ρは下記のように計算できる。

```
cor.test(x, y, method = "spearman")
```

```
## Warning in cor.test.default(x, y, method = "spearman"): Cannot compute
## exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  x and y
## S = 16.8, p-value = 0.07992
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 0.7
```

ρ = 0.7と求まった。ちなみに信頼区間についてはここを参照。ただし、この例のように同順位（タイ）があると

```
## Cannot compute exact p-value with ties
```

という**Warning**が出るので気持ちわるい。

警告を消すもっとも手っ取り早い対処法は、厳密性を求め過ぎないことである：

```
cor.test(x, y, method = "spearman", exact = FALSE)
```

```
##
##  Spearman's rank correlation rho
##
## data:  x and y
## S = 16.8, p-value = 0.07992
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 0.7
```

もしくは、

```
suppressWarnings(cor.test(x, y, method = "spearman"))
```

```
##
##  Spearman's rank correlation rho
##
## data:  x and y
## S = 16.8, p-value = 0.07992
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 0.7
```

と警告を抑制し非表示にする方法もある。しかしこれでは疑問の解消にならない。

そもそも、どのようにp値を計算しているのかを知ることがヒントになる（かもしれない）。cor.test()では、

> Algorithm AS 89 has two methods, the “exact” method and the “semi-exact” method. cor.test() uses the former when n < 10 and the latter when 9 < n < 1290. 参照リンク

というアルゴリズムで計算している。今の例の場合、n < 10なので、AS 89の“exact”法で計算していることになる
（後に明らかになるが、同順位がある場合はたとえexact = TRUEを指定してもAS 89法は適用されず
もっと単純な手法が実行されることが判明する。）

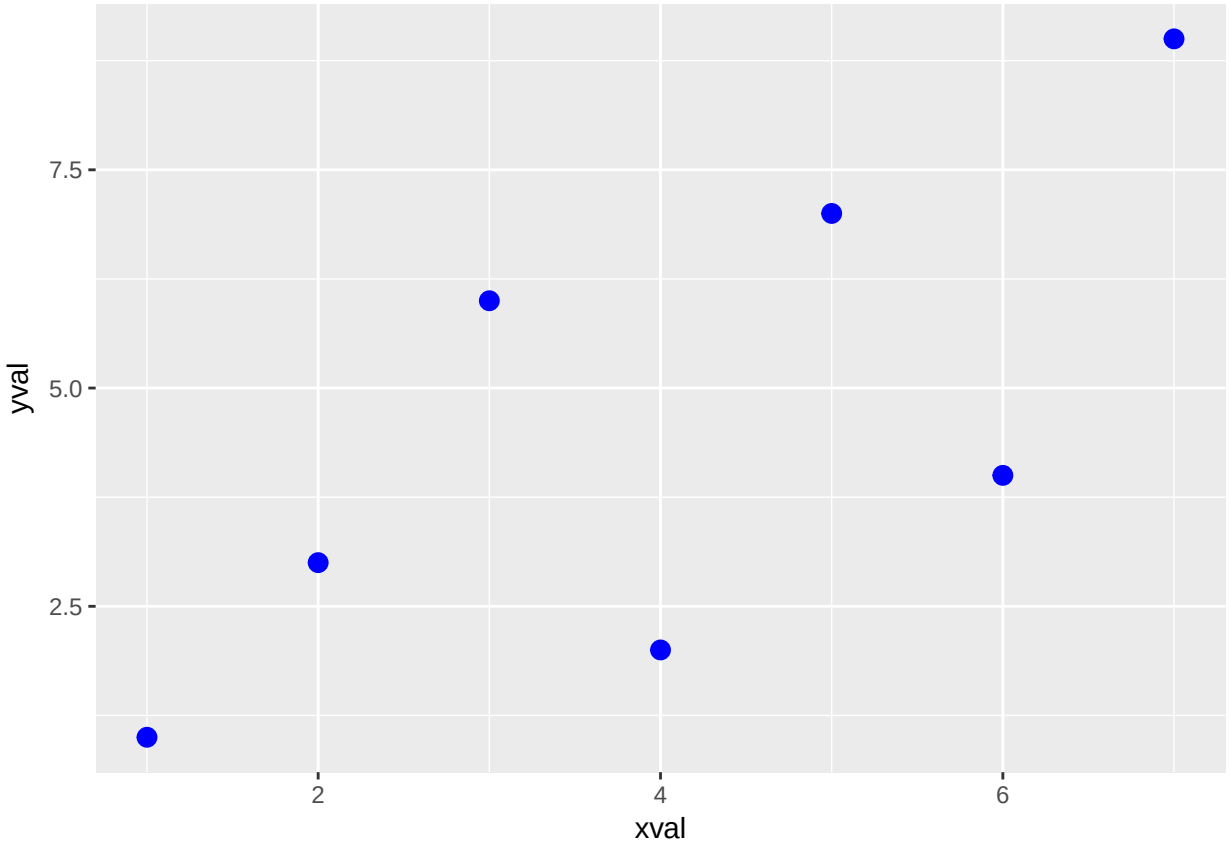
このアルゴリズムが内部でどのように計算しているかを検証するために、まずは同順位がない場合の別の例を見る。

同順位がない例でまずは検証

```
x2 <- 1:7
y2 <- c(1, 3, 6, 2, 7, 4, 9)
sampledat2 <- data.frame(xval = x2, yval = y2)
```

散布図にするとこんな感じ。

```
ggplot(sampledat2, aes(x=xval, y=yval)) + geom_point(color = "blue", size = 3)
```




```
(rho <- cor.test(x2, y2, method="spearman")$estimate)
```

```
##  rho
## 0.75
```

このデータでは$\rho = 0.75$となる。同順位がないので今回は警告が出ていないことに注意。
サンプルサイズが７と小さいので、全ての場合について並べ替え検定（permutation test）を手軽に実行できる。

ここにある求め方だと、

```
library(e1071)
par(family="Osaka")
# permutations()を使うためにロード
permutation <- permutations(7)
n <- length(x2)

# 引数とy2の相関係数（同順位がない場合）を計算する関数
f_rho <- function(a) 1 - 6 * sum((rank(a) - rank(y2))^2) / (n^3 - n)
```

```
x2_all_perm <- matrix(x2[permutation], ncol=7)
x2_all_perm_rho <- apply(x2_all_perm, 1, function(a) f_rho(a))

sum(x2_all_perm_rho > rho) / factorial(7) * 2
```

```
## [1] 0.06626984
```

```
cor.test(x2, y2, method="spearman")$p.value
```

```
## [1] 0.06626984
```

となり結果が一致する。最後に２倍しているのは両側検定するためである。絶対値を取って、

```
sum(abs(x2_all_perm_rho) > abs(rho)) / factorial(7)
```

```
## [1] 0.06626984
```

としても同じことになる（はず）。しかし、こちらのwhuberさんの解法はやや異なる結果を返す。

```
test <- function(x, y) suppressWarnings(cor.test(x, y, method="spearman")$estimate)
rho <- test(x2, y2)
# 並べ替え分布をシミュレーション
set.seed(1234)
p <- replicate(10^5, test(x2, sample(y2, length(y2))))

# 絶対値が厳密に超える場合をカウント
p_out <- sum(abs(p) > rho)

# もしあるなら、一致する場合をカウント
p_at <- sum(abs(p) == rho)

# 超える場合の割合、すなわちp値
(p_out + p_at /2) / length(p)
```

```
## [1] 0.05627
```

一致するときの場合の数を半分にしている（厳密に一致する期待値が半々だから？）ところが気になる。
シミュレーション回数がこの程度のオーダーで十分なのか不安が残るが、これ以上桁を増やすと、手元の環境で時間がかかりすぎる。

ここには別の算出法が紹介されている。その一つが、coin::spearman_test()を用いる方法。

```
library(coin)
# spearman_test()とapproximate()を使うためにロード
```

```
spearman_test(x2 ~ y2, distribution = approximate(B=10000000))
```

```
##
##  Approximative Spearman Correlation Test
##
## data:  x2 by y2
## Z = 1.8371, p-value = 0.06627
## alternative hypothesis: true rho is not equal to 0
```

approximate()はMonte-Carloリサンプリングを B回繰り返して検定統計量の帰無仮説の分布を作ってくれる関数。
今回のサンプルサイズだとこんなに繰り返さなくても良さそう。

もう一つが、combinat::permn()で全ての場合を計算する場合。上のe1071::permutations()を使った場合とやっていることはほぼ同じ。

```
library(combinat)
# permn()を使うためにロード
```

```
spcor <- sapply(permn(x2), y=y2, method="spearman", cor)
mean(abs(spcor) >= rho)
```

```
## [1] 0.06626984
```

ほぼ同様だが、Jorge Ortiz Pinillaさんの解法はこれ：

```
n <- length(y2)
NP <- factorial(n)
y2s <- permn(y2)
Rs <- rep(NA, NP)
for(i in 1:NP) {
  Rs[i]<- cor(x2, y2s[[i]], method = "spearman")
}
sum(abs(Rs) >= abs(rho)) / NP
```

```
## [1] 0.06626984
```

ρ の大文字は奇しくも P だが、それはともかく。閑話休題

そろそろ、同順位（タイ）のある場合の話に戻そう。これまで見てきた手法を用いて、最初のsampledatの場合にp値を求めてみよう。

```
(rho <- cor.test(x, y, method="spearman")$estimate)
```

```
## Warning in cor.test.default(x, y, method = "spearman"): Cannot compute
## exact p-value with ties
```

```
## rho
## 0.7
```

```
set.seed(1234)
p <- replicate(10^5, test(x, sample(y, length(y))))

# 絶対値が厳密に超える場合をカウント
p_out <- sum(abs(p) > rho)

# もしあるなら、一致する場合をカウント
p_at <- sum(abs(p) == rho)

# 超える場合の割合、すなわちp値
(p_out + p_at /2) / length(p)
```

```
## [1] 0.086005
```

```
spearman_test(x ~ y, distribution = approximate(B=10000000))
```

```
##
##  Approximative Spearman Correlation Test
##
## data:  x by y
## Z = 1.7146, p-value = 0.08854
## alternative hypothesis: true rho is not equal to 0
```

```
spcor <- sapply(combinat::permn(x), y=y, method="spearman", cor)
mean(abs(spcor) >= rho)
```

```
## [1] 0.08849206
```

```
n = length(y)
NP = factorial(n)
ys = permn(y)
Rs = rep(NA, NP)
for(i in 1:NP) {
  Rs[i] = cor(x, ys[[i]], method = "spearman")
}
sum(abs(Rs) >= abs(rho)) / NP
```

```
## [1] 0.08849206
```

ここで、最初に戻ってcor.test()の出力をもう一度見て見ると

```
cor.test(x, y, method = "spearman")
```

```
## Warning in cor.test.default(x, y, method = "spearman"): Cannot compute
## exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  x and y
## S = 16.8, p-value = 0.07992
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 0.7
```

これまで計算したp値と少しずれているようだが許容範囲内だろうか。
そもそも、同順位がある場合にこのcor.test()によるp値はどのように計算されたのか？AS 89アルゴリズムの原論文を見に行くとはっきりするが、これは同順位がない場合の計算である。

ここで、cor.test()のソースを

```
getS3method("cor.test", "default")
```

として中身を見てみる（最初からそうすべきだったかもしれない）と、同順位がある場合の判定を

```
n <- length(x)
(TIES <- (min(length(unique(x)), length(unique(y))) < n))
```

```
## [1] TRUE
```

で行っている。sampledatの場合、TIES = TRUEである。そして、

```
exact = TRUE
if (TIES && exact) {
  exact <- FALSE
  warning("Cannot compute exact p-value with ties")
  }
```

```
## Warning: Cannot compute exact p-value with ties
```

フラグをexact <- FALSEに変えて、問題の警告を発する。その上で、その後の具体的にp値を計算している部分だけを抽出すると、

```
r <- cor(rank(x) , rank(y))
q <- (n^3 - n) * (1 - r)/6
den <- (n * (n^2 - 1))/6
r <- 1 - q/den
pt(r/sqrt((1 - r^2)/(n - 2)), df = n - 2,
   lower.tail = FALSE) * 2
```

```
## [1] 0.07991669
```

という計算を中で行なっていたことがついに判明する。
同順位がある場合は、なんとざっくりt分布で近似してp値を求めているだけなのだった。つまり、

```
cor.test(rank(x), rank(y), method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  rank(x) and rank(y)
## t = 2.1918, df = 5, p-value = 0.07992
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1122070  0.9514894
## sample estimates:
## cor
## 0.7
```

と完全に同じである。意外と大胆な計算であった。

ちなみに、同順位がないときに $n \leq 22$ の場合の統計量Sの数表が知られていて、これを用いた厳密値を求める
pspearmanパッケージのpspearman::spearman.test()関数がある。

sampledata2において、

```
pspearman::spearman.test(x2, y2, approximation="exact")
```

```
##
##  Spearman's rank correlation rho
##
## data:  x2 and y2
## S = 14, p-value = 0.06627
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##  rho
## 0.75
```

また、approximation = “AS89”としても

```
pspearman::spearman.test(x2, y2, approximation="AS89")
```

```
##
##  Spearman's rank correlation rho
##
## data:  x2 and y2
## S = 14, p-value = 0.06627
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##  rho
## 0.75
```

厳密解と（有効数字の桁数において）同じ値を返すようだ。そして、sampledataにおいて、

```
pspearman::spearman.test(x, y, approximation = "t-distribution")
```

```
##
##  Spearman's rank correlation rho
##
## data:  x and y
```

S = 16.8, p-value = 0.07992
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.7

と指定するとcor.test()と同じ結果を返す。こちらの方が、どの近似法を用いたか明示的に指定するので意味を掴みやすい。

まとめると、同順位がある場合のSpearmanの相関係数ρの無相関検定（帰無仮説： $\rho = 0$ ）のp値の計算法のバリエーションを調べた。
デフォルトのcor.test(x, y, method = “spearman”)ではオプションでexact = TRUEを指定したとしても中では結局t分布の近似を行う。
一方、Monte-Carloシミュレーションや並べ替え検定による全パターンの網羅計算もサンプルサイズがある程度小さければ可能（tractable）である。
それらの結果はそれほど違わないようだ。

MCサンプリングや並べ替え検定よりもt分布による近似がほんの少し非保守的になったのが一般的な傾向なのか、
それとも今回の例においてたまたまそうなったのかは今後検証する必要がある。

また本題からは脇道にそれたが、同順位なしの場合にAS 89アルゴリズム（Best & Roberts, 1975）
という手法が精度の良い計算法として用いられていることがわかった。

最後に、検定のp値だけではなく効果量（相関係数の場合はその絶対値）やその信頼区間にも目を向けるべきであろう。
それに関しては、冒頭でも触れたが、Spearman’s ρの信頼区間の算出法が参考になる。

結論として、

```
## Cannot compute exact p-value with ties
```

という警告は普段はあまり気にせず無視するのが一番の得策かもしれない。