使用bootstrap和randomization进行统计推断

Alex / 2021-11-02 / free_learner@163.com / AlexBrain.cn

更新于2023-09-27，主要是文字排版上的更新，内容基本保持不变。

使用bootstrap和randomization的方法对（皮尔逊）相关系数进行统计推断，并与使用数学模型的方法进行比较。

一、背景

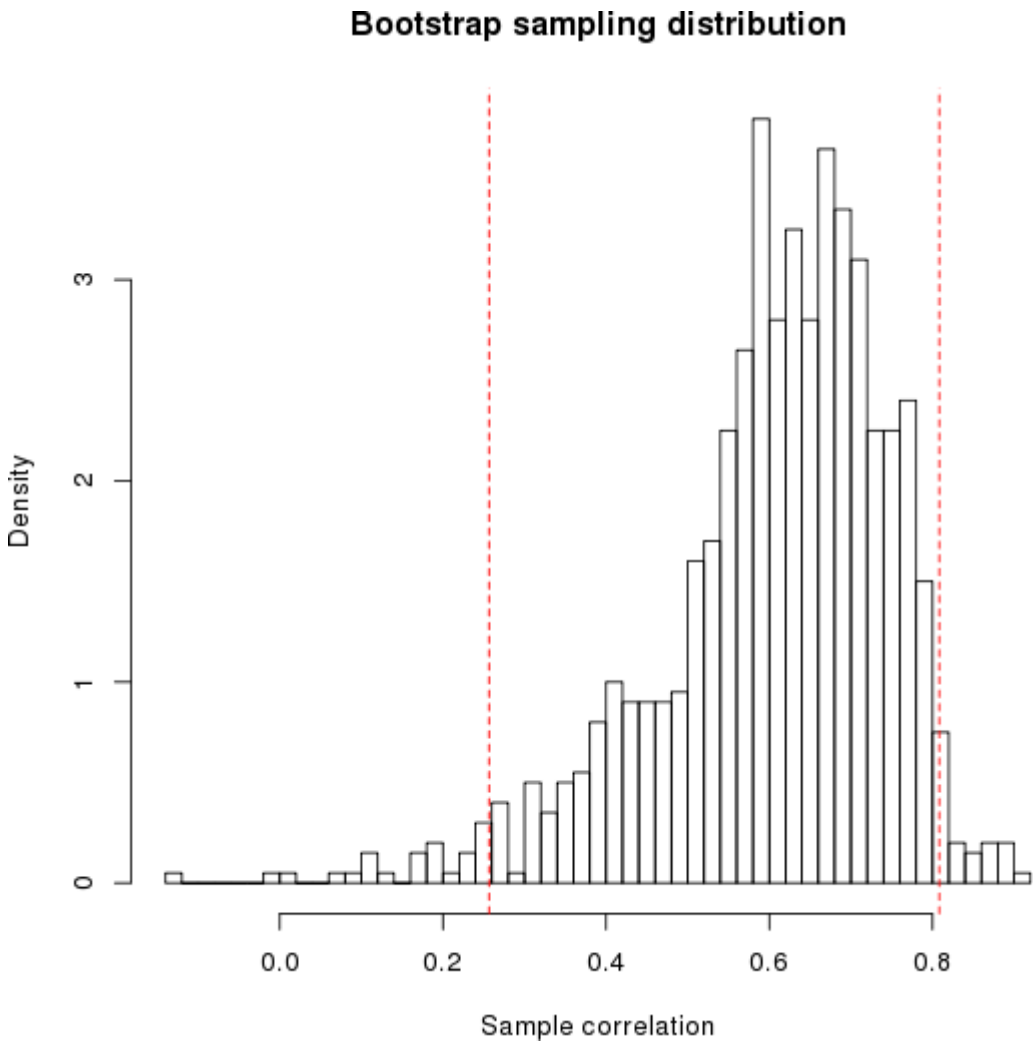
假设收集了一批数据，样本是15个被试（`N=15`），包含两个正态分布的变量X和Y，然后计算了这两个变量之间的(皮尔逊)相关系数`R(X,Y)=0.60`。现在的问题是，根据当前样本得到的`R=0.60`的变异性是怎样的？比如，重新采集一批数据，按照同样的流程，还会得到`R=0.60`吗？有多大可能性是偶然情况下得到的？下面分别采用bootstrap、randomization和数学模型的方法回答这个问题。

```
## observed sample data
dat <- data.frame(X=c(-43,-43, 52,-55,287,-36,49,77,33,-130,-52,-86,10,-95,34),
                  Y=c(65,-38,48,-166,144,-64,128,-67,159,-5,-36,-55,45,-164,6))
## number of subjects
N <- nrow(dat)
## calculate correlation
R <- cor(dat$X, dat$Y)
```

二、bootstrap

似乎需要重复采集新的样本才有可能知道`R`在不同样本中的变异性如何（实际中不具有可行性），而bootstrap方法表明似乎只要从当前样本重复抽样形成新的样本就足以反映`R`的变异性了。具体地，bootstrap方法重复从当前样本中选取`N=15`的样本，同一个被试可以选取多次（resample with replacement），在每一个bootstrap得到的样本中计算`R`，这样就得到了关于`R`的采样分布，这个分布也就反映了`R`由于随机采样引起的变异性，根据这个分布可以计算95%的置信区间，比如可以取该分布的2.5%和97.5%分位数作为置信区间（percentile method）。置信区间的意义就是，假设采用同样的方法采集多批数据并计算`R`，那么在95%的样本里得到的`R`会位于置信区间内，实际上也就是反映了`R`由于随机采样引起的变异性，如果置信区间很窄，那么R的变异性就小。

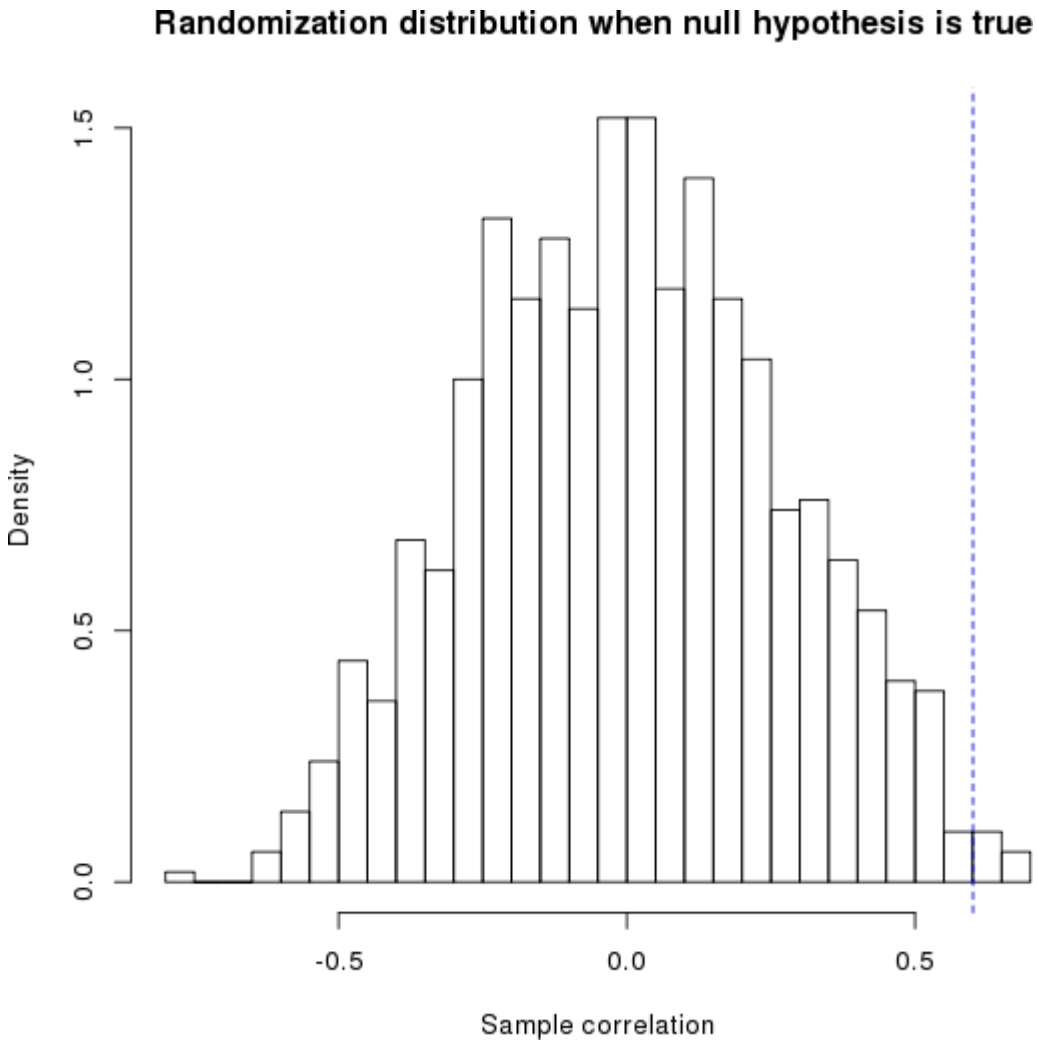
```r
## bootstrap percentile method
## set seed for reproducible purpose
set.seed(100)
## resample 1000 times
Nboot=1000
## R calculated in each bootstrap sample
Rboot=numeric(length=Nboot)
for (i in 1:Nboot){
  bootIdx <- sample(1:N, replace=TRUE)
  bootSample <- dat[bootIdx,]
  Rboot[i] <- cor(bootSample$X, bootSample$Y)
}
## CI = (0.2564250, 0.8083004)
CI <- quantile(Rboot, probs = c(0.025, 0.975))
```



三、randomization

除了置信区间，p值从另一个角度反映了统计量的变异性，p值的意义是，假设X和Y没有任何相关性、完全随机情况下(null hypothesis为真)，那么在当前样本中有多大可能性得到`R=0.60`？randomization方法的思路就是，假设X和Y没有任何相关性，那么随机打乱X和Y的样本的对应关系，然后在打乱的样本中计算`R`，就得到了完全随机情况下可能得到的`R`的分布情况，这样也就知道了当前样本得到的`R=0.60`是随机误差导致的可能性。randomization有时也称为permutation，似乎这两者之间有一些区别。

```
## randomization method
set.seed(100)
## randomize 1000 times
Nrand=1000
Rrand=numeric(length=Nrand)
for (i in 1:Nrand){
  randIdx <- sample(1:N, replace=FALSE)
  randSample <- dat$X[randIdx]
  Rrand[i] <- cor(randSample, dat$Y)
}
## multiply 2 to get two-sided p-value
## p-value = 0.01624078
Pvalue <- 2* sum(Rrand >= R)/Nrand
```



四、数学模型 (Fisher's Z)

Fisher发现将相关系数`R`进行一定函数变换后得到的统计量`Z=(1/2)log((1+R)/(1-R))`近似服从正态分布，`Z`的标准差为`S=1/sqrt(N-3)`，N为样本量，且这个标准差与`Z`的大小无关，也就是说不论`Z=0.1`还是`Z=0.9`，`Z`的标准差都只由样本量N决定。这样我们就知道`Z`的分布了，也即知道了`Z`的变异性，比如对于标准正态分布，大约95%的数据位于2个标准差之内。计算95%的置信区间`CI=(Z-1.96*S, Z+1.96*S)`，现在得到的是`Z`的置信区间，需要通过逆变换将`Z`转换为`R`，从而得到`R`的95%置信区间。同样地，在假设`Z=0`的情况下（null hypothesis为真），可以计算观测到当前的`Z`值的概率（即p值）。注意，R里面的`cor.test`在计算p值时使用的方法不同，计算置信区间使用的是相同的方法。

```r
## math model method using Fisher's Z transformation
Z <- atanh(R)
S <- 1/sqrt(N-3)
CI <- c(Z-1.96*S, Z+1.96*S)
## CI=(0.1266600, 0.8507745)
CI <- tanh(CI)
## p-value=0.0162
Pvalue <- 2*pnorm(abs(Z), mean=0,sd=S, lower.tail = F)
## compare with R's cor.test
## CI=(0.1273311, 0.8509570)
## p-value=0.01795
cor.test(dat$X, dat$Y)
```

五、小结

从结果来看，bootstrap方法得到的置信区间要比数学模型方法的结果窄一些，可能跟这里使用的percentile method不够准确有关，randomization方法得到的p值与数学模型方法的结果非常接近。bootstrap和randomization比较适合没有现成数学模型可以使用的统计推断问题。这里介绍的只是一些基本概念，不同方法也有一些局限性，要根据实际数据情况进行考虑。以上只是我自己的学习总结，多有谬误，谨慎参考，欢迎指正。

六、参考

1. Mine Çetinkaya-Rundel and Johanna Hardin, *Introduction to Modern Statistics*, https://www.openintro.org/book/ims/
2. <http://www.di.fc.ul.pt/~jpn/r/bootstrap/resamplings.html#example-1---obtaining-a-confidence-interval>