Bootstrap confidence intervals for Pearson correlation coeficient. Calculation on 'Law School' data

Ocaña Rebull, Jordi

8 de maig de 2015

We will use a sample of 82 law schools. Each faculty has the LSAT and GPA average rating of candidates for admission. LSAT (Law School Admission Test) is the score of tests made by accredited evaluation centers, independent of the university system. GPA (Grade Point Average) is a score that assesses the record of each student in their pre-university studies. The data will be organized into a data.frame of 82 rows for law schools and 2 columns for variables LSAT and GPA

```
> lawSchool <- read.table(file="Law_School.txt", header = TRUE)
> n <- nrow(lawSchool)
> n
```

```
[1] 82
```

Pearson correlation coefficient on the original sample:

```
> r <- cor(lawSchool)[2,1]
> r
```

```
[1] 0.7599979
```

Stantdard error of Pearson correlation coefficient sampling estimation. It is based on a normal parametric approximation:

```
> se.r <- (1 - r*r) / sqrt(n - 3)
> se.r
```

```
[1] 0.04752408
```

These are bivariate data. Each faculty has a pair of values (LSAT, GPA). The empirical distribution would have a bibariate density function that gives a probability of 1/n to each observed pair. The bootstrap resampling associated with this distribution consists in choosing whole rows (schools) randomly and with replacement.

ROW INDEXES generation randomly and with replacement (i.e, of the schools) that belong to each non-parametric boostrap resample:

```
> set.seed(123)
> indexs = sample(1:n,replace=TRUE)
> indexs
```

```
 [1] 24 65 34 73 78  4 44 74 46 38 79 38 56 47  9 74 21  4 27 79 73 57 53 82 54
[26] 59 45 49 24 13 79 74 57 66  3 40 63 18 27 19 12 34 34 31 13 12 20 39 22 71
[51]  4 37 66 10 46 17 11 62 74 31 55  8 32 23 67 37 67 67 66 37 62 52 59  1 39
[76] 19 32 51 29 10 20 55
```

The bootstrap resample will be a new data.frame formed by choosing rows 24, 65, 34, 73, ..., 10, 20, 55 of the original sample:

```
> lawSchool[indexs, ]
```

```
     LSAT  GPA
24    575 3.01
65    562 3.01
34    591 3.02
73    605 3.45
78    590 3.15
4     653 3.12
44    644 3.38
74    565 3.15
46    645 3.27
38    606 3.20
79    558 2.81
38.1  606 3.20
56    641 3.28
47    651 3.36
9     553 2.97
74.1  565 3.15
21    546 2.99
4.1   653 3.12
27    608 3.04
79.1  558 2.81
73.1  605 3.45
57    512 3.01
53    594 2.96
82    575 2.74
54    594 3.05
59    597 3.32
45    545 2.76
```

49	609	3.17
24.1	575	3.01
13	635	3.30
79.2	558	2.81
74.2	565	3.15
57.1	512	3.01
66	635	3.30
3	579	3.24
40	535	2.98
63	572	3.08
18	646	3.47
27.1	608	3.04
19	622	3.15
12	596	3.24
34.1	591	3.02
34.2	591	3.02
31	605	3.13
13.1	635	3.30
12.1	596	3.24
20	611	3.33
39	603	3.23
22	614	3.19
71	570	3.01
4.2	653	3.12
37	615	3.37
66.1	635	3.30
10	607	2.91
46.1	645	3.27
17	599	3.23
11	558	3.11
62	637	3.33
74.3	565	3.15
31.1	605	3.13
55	560	2.93
8	615	3.40
32	704	3.36
23	628	3.03
67	614	3.15
37.1	615	3.37
67.1	614	3.15
67.2	614	3.15
66.2	635	3.30
37.2	615	3.37
62.1	637	3.33
52	580	3.07
59.1	597	3.32

```
1      622 3.23
39.1   603 3.23
19.1   622 3.15
32.1   704 3.36
51     586 3.11
29     587 3.16
10.1   607 2.91
20.1   611 3.33
55.1   560 2.93
```

Pay attention at row names, they correspond to the row in the original sample, with repetitions indicated by a notation with two points: first repetition of row 38: 38.1, etc.

Correlation coefficient calculation for the previous sample:

```
> r.boot <- cor(lawSchool[indexs,])[2,1]
> r.boot
```

```
[1] 0.635637
```

Studentized correlation coefficient calculation for the previous sample:

```
> t.r.boot <- (r.boot - r) / ((1 - r.boot*r.boot) / sqrt(n - 3))
> t.r.boot
```

```
[1] -1.85471
```

Or, preferably:

```
> se.fact <- sqrt(n - 3)
> t.r.boot <- se.fact * (r.boot - r) / (1 - r.boot*r.boot)
> t.r.boot
```

```
[1] -1.85471
```

Bootstap simulation:

```
> B <- 10000
> se.fact <- sqrt(n - 3)  # calculado al principio, una sola vez
> set.seed(123)
> t.r.boots <- replicate(B,
+ {
+   r.boot <- cor(lawSchool[sample(1:n, replace = TRUE),])[2,1]
+   se.fact * (r.boot - r) / (1 - r.boot*r.boot)
+ }
+ )
```

The 20 first values of the Studentized correlation coefficient:

```
> t.r.boots[1:20]

 [1] -1.85471029  0.60561186  2.73633475  0.23068329  1.03394809 -1.44749590
 [7]  0.06761746 -0.42309865  0.52450770 -1.88063124 -0.99383209 -0.84213093
[13]  1.91134844  0.73770516 -0.94541225 -0.76885145  0.17755766 -1.46271684
[19]  0.21724256 -0.09909884
```

Bootstrap-t confidence interval for the correlation coefficient:

```
> alpha <- 0.05
> confLevel <- 1 - alpha
```

Critical values that leave alpha/2 probability on the left tail and alpha/2 on the right tail of the bootstrap distribution:

```
> quantile(t.r.boots, probs = c(alpha/2, 1 - alpha/2))

     2.5%     97.5%
-1.716240  2.729268
```

Bootstrap-t confidence interval:

```
> icBoot.t <- r - quantile(t.r.boots, probs = c(1 - alpha/2, alpha/2)) * se.r
> names(icBoot.t) <- NULL
> attr(icBoot.t, "conf.level") = confLevel
> icBoot.t

[1] 0.6302919 0.8415606
attr(,"conf.level")
[1] 0.95
```

Symmetrized Bootstrap-t confidence interval:

```
> t_alpha <- quantile(abs(t.r.boots), probs = 1 - alpha)
> icBoot.t.sym <- r + c(-t_alpha, t_alpha) * se.r
> names(icBoot.t.sym) <- NULL
> attr(icBoot.t.sym, "conf.level") = confLevel
> icBoot.t.sym

[1] 0.6543610 0.8656347
attr(,"conf.level")
[1] 0.95
```

”Parametric bootstrap”version of these CI

We use the library ’mvtnorm’ that provides the function ’rmvnorm’ to generate the multivariate normal distribution.
Function in the library ’mvtnorm’ that generates vectors according to a normal disribution with means vector ’mean’ anc covariance matrix ’sigma’. For example:

```
> require(mvtnorm)
> rmvnorm(n = 5, mean = c(1,2), sigma = matrix(c(10,3,3,2), ncol = 2))
```

```
         [,1]     [,2]
[1,] 2.182585 3.254844
[2,] 1.414532 1.850358
[3,] 5.977932 2.777280
[4,] 4.860645 3.004644
[5,] 3.207239 1.024208
```

Normal resample from means and variances-covariances estimated on the original sample:

```
> medias = colMeans(lawSchool)
> medias
```

```
      LSAT        GPA
597.548780   3.134878
```

```
> var.covs = cov(lawSchool)
> var.covs
```

```
            LSAT        GPA
LSAT 1481.337097 5.54296899
GPA     5.542969 0.03590924
```

```
> rmvnorm(n = n, mean = medias, sigma = var.covs)
```

```
          LSAT      GPA
 [1,] 560.8508 3.167428
 [2,] 586.0177 3.312741
 [3,] 589.5335 3.139369
 [4,] 567.3115 3.037785
 [5,] 565.8043 3.053365
 [6,] 642.5043 3.057380
 [7,] 600.9321 2.817505
 [8,] 596.1191 3.038841
 [9,] 641.3206 3.271762
[10,] 587.4629 3.203548
[11,] 698.1692 3.358018
[12,] 611.2894 3.310096
[13,] 610.1641 3.049366
[14,] 516.7369 2.877911
[15,] 654.9706 3.412059
[16,] 629.3020 3.467297
[17,] 653.0346 3.308455
[18,] 532.5582 2.834267
[19,] 571.8885 3.055301
```

```
[20,] 566.6974 2.993398
[21,] 616.4804 3.219400
[22,] 628.6969 3.097368
[23,] 615.1402 3.386822
[24,] 569.0196 2.987336
[25,] 612.9276 3.316990
[26,] 589.6530 2.955557
[27,] 635.8053 3.191769
[28,] 615.7221 3.103998
[29,] 535.4343 2.962828
[30,] 634.9443 3.159491
[31,] 597.7344 3.466187
[32,] 575.7241 3.001823
[33,] 564.1862 3.049644
[34,] 630.4149 3.412947
[35,] 518.3740 2.775397
[36,] 536.3250 2.953036
[37,] 609.5974 3.240238
[38,] 655.6501 3.415887
[39,] 621.8865 3.106758
[40,] 533.3277 2.886385
[41,] 573.0728 3.091441
[42,] 574.4478 3.112135
[43,] 552.3868 2.964277
[44,] 536.0939 2.734413
[45,] 611.2460 3.311614
[46,] 628.1122 3.264423
[47,] 576.2499 2.904821
[48,] 614.7005 3.237313
[49,] 549.8597 2.792294
[50,] 563.0267 3.077204
[51,] 638.3168 3.309760
[52,] 610.9203 3.127860
[53,] 571.5366 3.030358
[54,] 579.1346 2.941538
[55,] 499.0289 2.775437
[56,] 654.3605 3.353128
[57,] 648.9890 3.262168
[58,] 631.1873 3.228958
[59,] 658.1263 3.172976
[60,] 600.3737 3.338386
[61,] 615.7098 3.309928
[62,] 578.9159 3.036215
[63,] 597.3075 3.030441
[64,] 527.4882 2.744620
[65,] 597.6372 3.131173
```

```
[66,] 577.2050 3.236299
[67,] 624.8316 3.137156
[68,] 616.9133 3.223003
[69,] 607.3059 3.070214
[70,] 594.4924 3.048205
[71,] 587.2643 3.146199
[72,] 640.4385 3.333733
[73,] 569.1573 2.931734
[74,] 578.1500 2.965237
[75,] 556.0460 2.884527
[76,] 629.6007 3.193740
[77,] 581.8248 3.032912
[78,] 623.8038 3.200184
[79,] 514.5791 2.844006
[80,] 604.2256 3.299752
[81,] 602.2256 3.326774
[82,] 571.6791 2.957704
```

Normal parametric bootstrap simulation:

```
> B <- 10000
> se.fact <- sqrt(n - 3)  # computed at first, only once
> set.seed(123)
> t.r.boots <- replicate(B,
+ {
+   r.boot <- cor(rmvnorm(n = n, mean = medias, sigma = var.covs))[2,1]
+   se.fact * (r.boot - r) / (1 - r.boot*r.boot)
+ }
+ )
```

The 20 first values of studentized correlation coefficient:

```
> t.r.boots[1:20]
```

```
 [1] -1.0433796  0.3878359 -1.5615257 -0.2041576  0.6439045 -0.9727124
 [7] -0.9196693 -0.5860349  0.3918097  1.1371120 -0.9200732 -1.9544386
[13] -0.1012589  0.2630677  1.1499794 -0.3482270  1.0672779  0.8965858
[19] -0.8473069 -0.1959695
```

The final calculation of confidence intervals would be identical to above, the only difference is the procedure to generate resamples.
Bootstrap-t Confidence Interval:

```
> icBoot.t <- r - quantile(t.r.boots, probs = c(1 - alpha/2, alpha/2)) * se.r
> names(icBoot.t) <- NULL
> icBoot.t
```

```
[1] 0.6473788 0.8381865
```

Symetrized Bootstrap-t Confidence Interval:

```
> t_alpha <- quantile(abs(t.r.boots), probs = 1 - alpha)
> icBoot.t.sym <- r + c(-t_alpha, t_alpha) * se.r
> names(icBoot.t.sym) <- NULL
> attr(icBoot.t.sym, "conf.level") = confLevel
> icBoot.t.sym
```

```
[1] 0.6630020 0.8569938
attr(,"conf.level")
[1] 0.95
```

Since this is the mean score of all candidate students in each faculty, the assumption of normality is reasonable. It's interesting to compare bootstrap intervals above with the usual normal parametric confidence interval:

```
> cor.test(lawSchool[,"LSAT"], lawSchool[,"GPA"])$conf.int
```

```
[1] 0.6502299 0.8386849
attr(,"conf.level")
[1] 0.95
```

Bootstrap-t. Standard error estimation by jackknife

An alternative possibility to estimate se(r): the jackknife

```
> seJack.r <- function(xy) {
+   n <- nrow(xy)
+   r_i <- numeric(n)
+   x <- xy[,1]
+   y <- xy[,2]
+   for (i in 1:n) {
+     r_i[i] <- cor(x[-i], y[-i])
+   }
+   return(sqrt(((n - 1) / n) * sum((r_i - mean(r_i))^2)))
+ }
> # Slightly fastest version:
> seJ.r <- function(xy) {
+   n <- nrow(xy)
+   x <- xy[,1]
+   y <- xy[,2]
+   r_i <- vapply(1:n, function(i) cor(x[-i], y[-i]), FUN.VALUE = 0.0)
+   return(sqrt(((n - 1) / n) * sum((r_i - mean(r_i))^2)))
+ }
> require(microbenchmark)
> microbenchmark(
+   se.r <- seJack.r(lawSchool),
+   se.r <- seJ.r(lawSchool)
+ )
```

```
Unit: milliseconds
                         expr      min       lq     mean   median       uq
 se.r <- seJack.r(lawSchool) 2.796962 2.947059 3.164097 3.077342 3.177835
    se.r <- seJ.r(lawSchool) 2.657556 2.876928 3.400500 2.998803 3.124382
       max neval
  4.788562   100
 34.000583   100
```

```
> # Nearly no difference...
>
> se.r <- seJ.r(lawSchool)
> se.r
```

```
[1] 0.05334785
```

Bootstrap resampling process (patience...)

```
> t.r.boots <- replicate(B,
+ {
+   lawSchool.boot <- lawSchool[sample(1:n, replace = TRUE),]
+   (cor(lawSchool.boot)[2,1] - r) / seJ.r(lawSchool.boot)
+ }
+ )
```

Bootstrap-t confidence interval:

```
> icBoot.t <- r - quantile(t.r.boots, probs = c(1 - alpha/2, alpha/2)) * se.r
> names(icBoot.t) <- NULL
> attr(icBoot.t, "conf.level") = confLevel
> icBoot.t
```

```
[1] 0.6397181 0.8592908
attr(,"conf.level")
[1] 0.95
```

Symetrized Bootstrap-t confidence interval:

```
> t_alpha <- quantile(abs(t.r.boots), probs = 1 - alpha)
> icBoot.t.sym <- r + c(-t_alpha, t_alpha) * se.r
> names(icBoot.t.sym) <- NULL
> attr(icBoot.t.sym, "conf.level") = confLevel
> icBoot.t.sym
```

```
[1] 0.6508230 0.8691727
attr(,"conf.level")
[1] 0.95
```

```
>
```

Bootstrap percentile interval

```
> B <- 10000
> alpha <- 0.05
> confLevel = 1 - alpha
```

B values of r*

```
> r.boot <- replicate(B, cor(lawSchool[sample(1:n, replace=TRUE),])[2,1])
```

The 10 first r*:

```
> r.boot[1:10]
```

```
 [1] 0.7205762 0.7883022 0.7253621 0.7593219 0.7217442 0.7956063 0.7645087
 [8] 0.7534435 0.7860828 0.7284076
```

Bootstrap percentile confidence interval at 95

```
> icBoot.perc = quantile(r.boot, probs = c(alpha/2, 1 - alpha/2))
> names(icBoot.perc) = NULL
> attr(icBoot.perc, "conf.level") = confLevel
> icBoot.perc
```

```
[1] 0.6496211 0.8465200
attr(,"conf.level")
[1] 0.95
```

Bootstrap BCa percentile interval

Data matrix without row 4:

```
> lawSchool[-4,]
```

```
   LSAT  GPA
1   622 3.23
2   542 2.83
3   579 3.24
5   606 3.09
6   576 3.39
7   620 3.10
8   615 3.40
9   553 2.97
10  607 2.91
11  558 3.11
12  596 3.24
13  635 3.30
```

14	581	3.22
15	661	3.43
16	547	2.91
17	599	3.23
18	646	3.47
19	622	3.15
20	611	3.33
21	546	2.99
22	614	3.19
23	628	3.03
24	575	3.01
25	662	3.39
26	627	3.41
27	608	3.04
28	632	3.29
29	587	3.16
30	581	3.17
31	605	3.13
32	704	3.36
33	477	2.57
34	591	3.02
35	578	3.03
36	572	2.88
37	615	3.37
38	606	3.20
39	603	3.23
40	535	2.98
41	595	3.11
42	575	2.92
43	573	2.85
44	644	3.38
45	545	2.76
46	645	3.27
47	651	3.36
48	562	3.19
49	609	3.17
50	555	3.00
51	586	3.11
52	580	3.07
53	594	2.96
54	594	3.05
55	560	2.93
56	641	3.28
57	512	3.01
58	631	3.21
59	597	3.32

```
60  621 3.24
61  617 3.03
62  637 3.33
63  572 3.08
64  610 3.13
65  562 3.01
66  635 3.30
67  614 3.15
68  546 2.82
69  598 3.20
70  666 3.44
71  570 3.01
72  570 2.92
73  605 3.45
74  565 3.15
75  686 3.50
76  608 3.16
77  595 3.19
78  590 3.15
79  558 2.81
80  611 3.16
81  564 3.02
82  575 2.74
```

Obtaining n jackknife replicas of correlation coefficient

```
> r_i <- numeric(n)
> for (i in 1:n) r_i[i] <- cor(lawSchool[-i,])[2,1]
> r_i <- mean(r_i) - r_i
> a <- sum(r_i^3) / (6 * sum(r_i^2)^1.5)
> a
```

```
[1] 0.02353688
```

```
> z0 <- qnorm(sum(r.boot <= r) / B)
> z0
```

```
[1] -0.03836082
```

```
> zalpha <- - qnorm(alpha/2)
> zalpha
```

```
[1] 1.959964
```

```
> icBoot.BCa = quantile(r.boot,
+   probs = c(
+     pnorm(z0 + (z0 - zalpha) / (1 - a * (z0 - zalpha))),
+     pnorm(z0 + (z0 + zalpha) / (1 - a * (z0 + zalpha)))
```

```
+   )
+ )
> names(icBoot.BCa) = NULL
> attr(icBoot.BCa, "conf.level") = confLevel
> icBoot.BCa

[1] 0.6503347 0.8469109
attr(,"conf.level")
[1] 0.95
```

In this example, all intervals (Bootstrap-t, percentil, BCa and normal parametric) are very similar, possibly due to the large sample size (82) and the validity of the assuption of existence of a normalizing transformation (for percentile and BCa)