

Introduction to the R package **rich**

Jean-Pierre Rossi

CBGP - INRA Montpellier <http://www6.montpellier.inra.fr/cbgp>

*For **rich** version 1.0.1*

2016-08-12

Contents

1	Installation	2
2	What is rich ?	2
3	Basic features	2
4	Comparing species richness	5
4.1	Principle	5
4.2	Functions c2cv and c2m	5
4.2.1	c2cv	5
4.2.2	c2m	6
5	Rarefaction curves	11
5.1	rarc	11
5.2	raref	15
5.3	raref2	17
5.4	shared	17
	References	19

1 Installation

The current stable version (rich 1.0.1) is available from CRAN. You can also install `rich` from R by simply typing:

```
install.packages("rich", dep=TRUE)
```

The development version is hosted by R-Forge and you can install it from R by typing:

```
install.packages("rich", repos="http://R-Forge.R-project.org", dep=TRUE)
```

2 What is rich ?

`rich` is a set of functions designed to perform simple species richness analyses. Readers will find interesting R resources in the package `vegan` (Oksanen et al. 2016).

3 Basic features

The function `rich` computes the species richness on the basis of bootstrap estimation.

```
library(rich)
data(ef)

# Bootstrap estimation based on 99 randomizations
o1 <- rich(matrix=ef, nrandom=99)
o1

## $cr
## [1] 121
##
## $mr
## [1] 10.4
##
## $mrsd
## [1] 4.938239
##
## $bootCR
##   cr.obs  cr.boot cr.bcorr  cr.bias   cr.se  cr.lbn  cr.ubn
##    121  90.74747 151.2525 -30.25253  8.764892 134.0737 168.4314
##
```

```
## $bootMR
##   mr.obs mr.boot mr.bcorr   mr.bias   mr.se mr.lbn mr.ubn
## 79.63333 79.00539 80.26128 -0.6279461 14.93246 50.99419 109.5284
##
## $nrandom
## [1] 99
```

The mean species richness *i.e.* the average value over the sampling units is given in the slot `$mr` and its standard deviation is given in `$mrsd`. The cumulated richness is given in `$cr`. The bootstrap estimate of the cumulated richness is stored in `$bootCR`. `$cr.obs` is simply the observed cumulated value whereas the corresponding bootstrapped value is reported in `$cr.boot`.

We can plot the mean and cumulated bootstrap estimates of species richness:

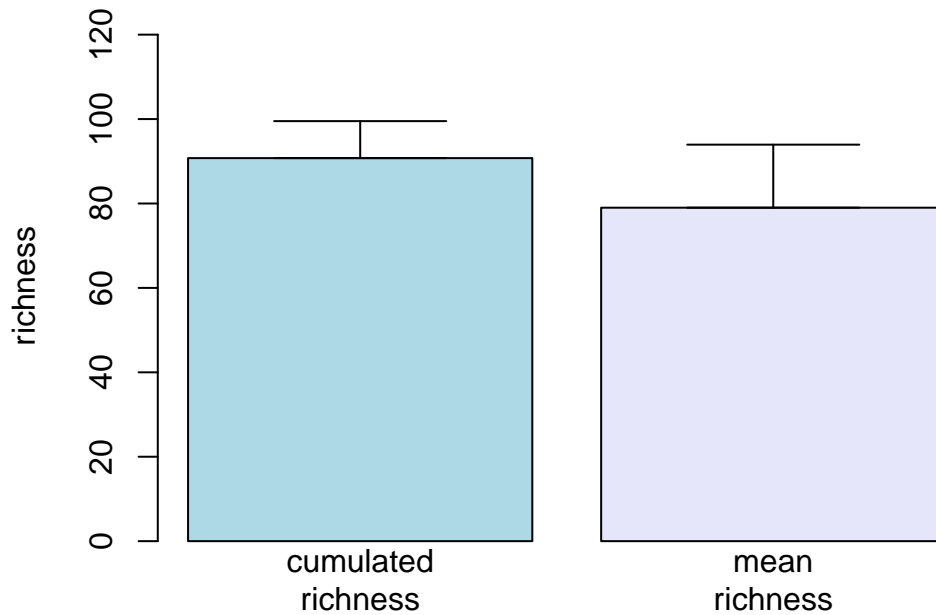
```
# plot bootstrapped cumulated and mean richness values
library(gplots)
col <- c("lightblue", "lavender")
d <- c(o1$bootCR$cr.boot, o1$bootMR$mr.boot)

ci.l <- c(o1$bootCR$cr.boot + o1$bootCR$cr.se,
          o1$bootMR$mr.boot + o1$bootMR$mr.se)
ci.u <- c(o1$bootCR$cr.boot, o1$bootMR$mr.boot)
ci.l ; ci.u
```

```
## [1] 99.51237 93.93785
```

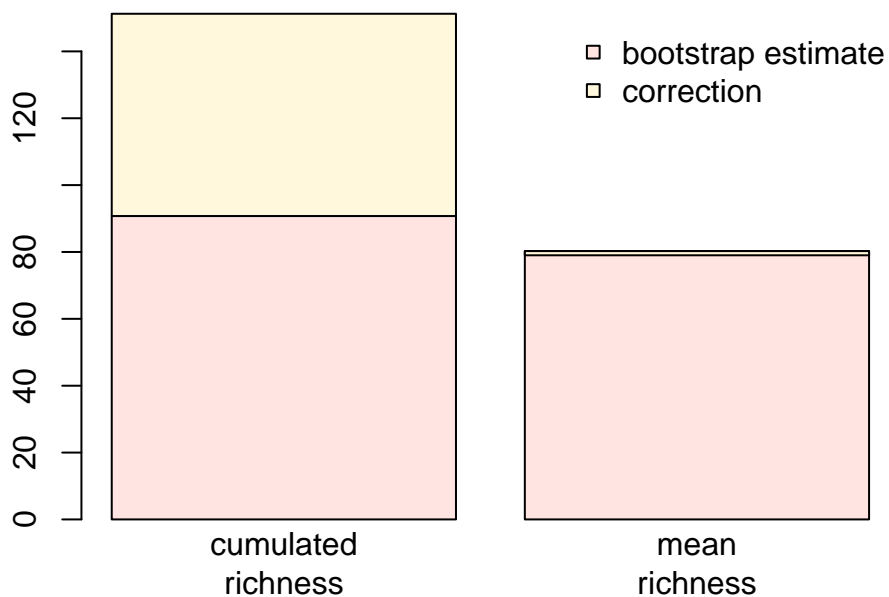
```
## [1] 90.74747 79.00539
```

```
barplot2(d, col = col, ylim=c(0,o1$bootCR$cr.obs),
          plot.ci = TRUE, ci.l = ci.l, ci.u = ci.u, ylab="richness",
          names.arg=c("cumulated\nrichness", "mean\nrichness"))
```



The dispersion of the bootstrap estimates can be used to compute a corrected value (Manly 1997). `rich` provides these corrected estimates and the outputs can be visualized as follows:

```
# plot corrected estimates
col <- c("mistyrose", "cornsilk")
d <- c(o1$bootCR$cr.boot, o1$bootCR$cr.bcorr-o1$bootCR$cr.boot)
d2 <- c(o1$bootMR$mr.bcorr, o1$bootMR$mr.boot-o1$bootMR$mr.bcorr)
dd <- cbind(d,d2)
barplot2(dd, col=col, names.arg=c("cumulated\nrichness", "mean\nrichness"))
legend("topright", legend=c("bootstrap estimate", "correction"),
      col="black", pch=c(22,22), bty="n", pt.bg=col)
```



4 Comparing species richness

4.1 Principle

One very common question is to determine if the species richness estimated in two sampling areas are statistically significant. This is for example the question if we sample fauna in plots under conventional agriculture and organic farming. Usually we perform sampling in each site, using a set of sampling units such as traps for insects, soil monoliths for soil fauna, surbers in rivers surveys etc. Each sampling unit brings a set of species forming a list whose length is the species richness.

Imagine a very simple framework where sites A and B are sampled using n sampling units. Each unit brings one estimation of species richness. We thus end up with n local estimates of the richness S for each site.

If we want to compare the richness of site A and B we must be careful. A very common strategy is to perform a student t test which amounts to compare the mean richness of each site. This corresponds to averaging each set of n values and compare the resulting means. This is not a comparison of the richness in site A and B, but rather a comparisons of the density of species richness in each site. Many users go for a student t test because they need a statistical test and that means they need replicates.

Imagine that 2 sites are sampled with 2 replicates and that the data are as follows:

	site 1		site 2	
	sample 1	sample 2	sample 1	sample 2
species 1	1	0	1	1
species 2	0	1	0	0

Both sites have a mean richness of 1 species per sample. However the cumulated richness is 2 in site 1 and only 1 in site 2. This very simple example illustrates the importance of considering the cumulated richness in biodiversity surveys. The problem is that we end up comparing only 2 values, one per site, with no replicate. This impairs statistical comparisons by mean of usual tests.

Randomization tests offer a solution that is clearly explained in Manly (1997). The function `c2cv` (standing for *compare 2 cumulated values*) implements such comparison of 2 values of species richness using a randomization procedure. Note that the function only handles 2 values comparison and thus does not allow multi-site direct analysis.

4.2 Functions `c2cv` and `c2m`

4.2.1 `c2cv`

`c2cv` stands for *compare 2 cumulated values*.

```

data(efeb)
out <- c2cv(com1=efeb$ef,com2=efeb$eb,nrandom=100,verbose=FALSE)

out$res

##
## cv1                121.00000000
## cv2                22.00000000
## cv1-cv2            99.00000000
## p                  0.00990099
## quantile 0.025     -35.00000000
## quantile 0.975      30.50000000
## randomized cv1-cv2 -0.58415842
## nrandom            100.00000000

```

The difference between the richness in site 1 and site 2 is given by `cv1-cv2` and equals 99 species in the example. The corresponding value after randomizations is indicated by `randomized cv1-cv2` and is much smaller. The n randomized values are used to compute the quantiles at $p = 0.975$ and $p = 0.025$ corresponding to a global interval of 95%. We can see that the observed difference is well above the upper quantile value (ca. 31) indicating that the observed difference is much larger than expected under the null hypothesis of “no difference between sites”.

4.2.2 c2m

`c2m` compares mean richness of two populations and any type of data can be processed. We illustrate the function using the example of the golden jackals given p. 4 in Manly (1997).

```

# The example of mandible length of male and female
# golden jackals from Manly (1997), p.4.
males<-c(120, 107, 110, 116, 114, 111, 113, 117, 114, 112)
females<-c(110, 111, 107, 108, 110, 105, 107, 106, 111, 111)
out <- c2m(pop1=males, pop2=females, nrandom=99)

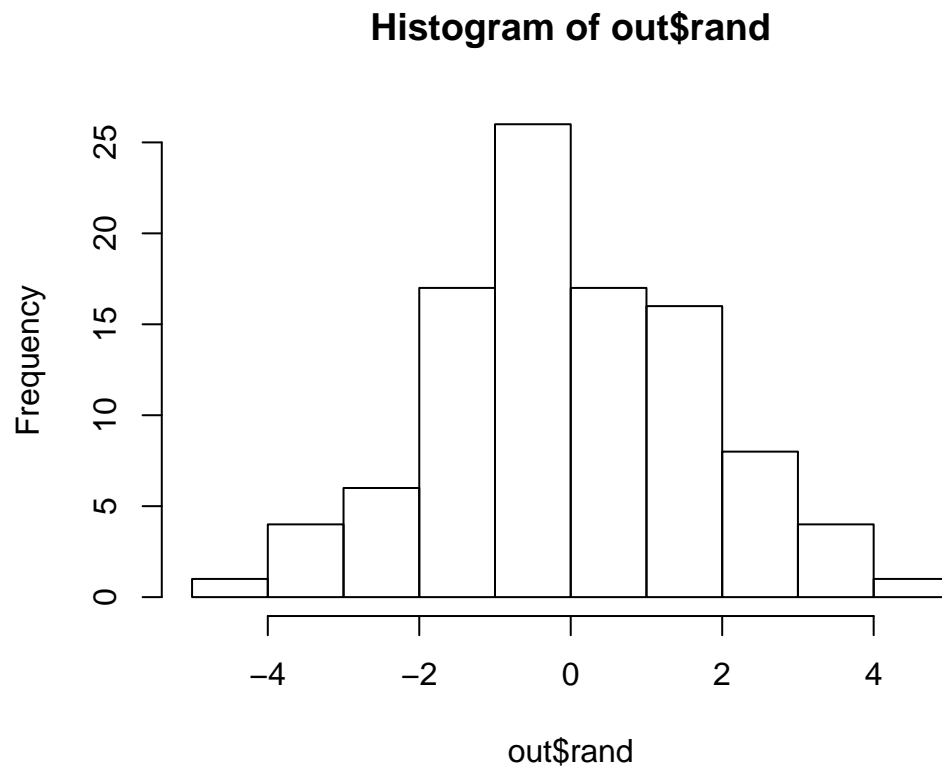
out$res

##
## mv1                113.400
## mv2                108.600
## mv1-mv2            4.800
## p                  0.010
## quantile 0.025     -3.400
## quantile 0.975      3.305
## randomized mv1-mv2  0.062
## nrandom            99.000

```

`out$res` contains the results while `out$rand` gives the values of the difference between the means to be compared after randomization. We can plot an histogram of these values:

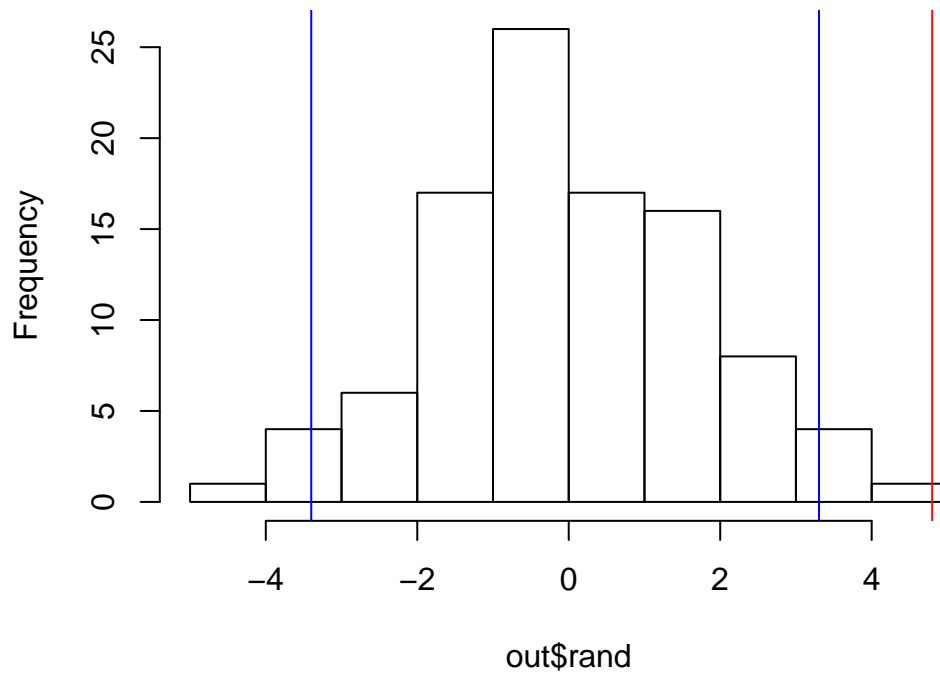
```
hist(out$rand)
```



It is interesting to add the vertical lines corresponding to the observed value (in red) and the quantile values for probability values of 0.975 and 0.025 (in blue):

```
hist(out$rand)
abline(v=out$res[3,1], col="red")
abline(v=out$res[5,1], col="blue")
abline(v=out$res[6,1], col="blue")
```

Histogram of out\$rand

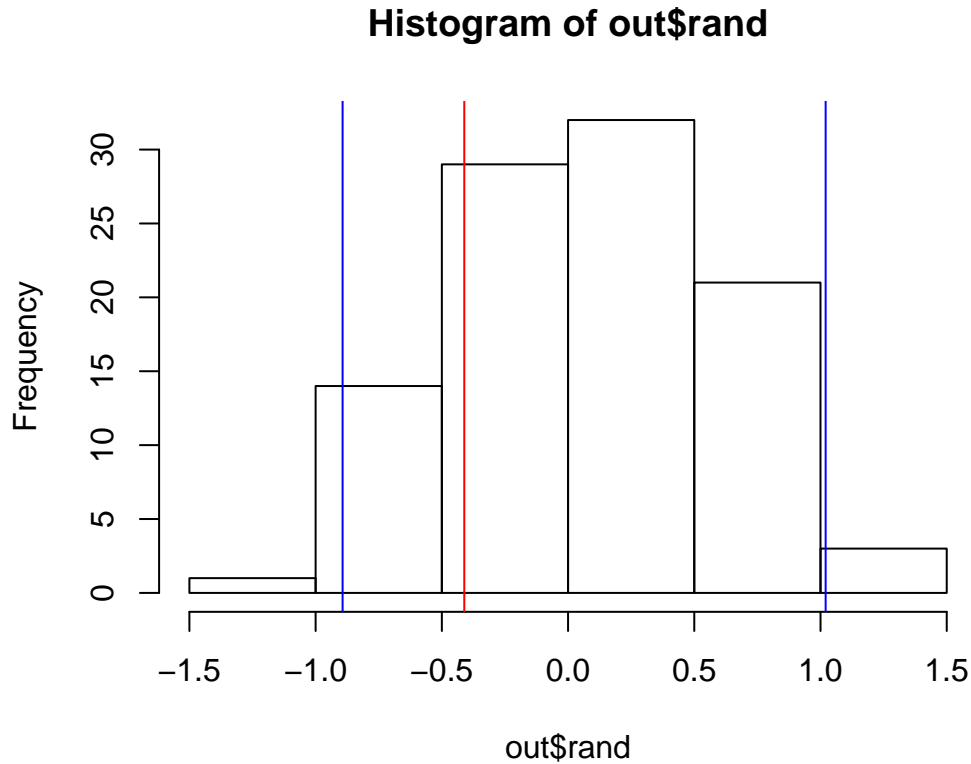


Let's see what happens when the populations are very similar. In the following example we simulate normal populations and compare them using `c2m`:

```
pop1<-rnorm(10)
pop2<-rnorm(10)
out <- c2m(pop1=pop1, pop2=pop2, nrandom=99)
out$res
```

```
##
## mv1                0.04133802
## mv2                0.45244936
## mv1-mv2            -0.41111134
## p                  0.21000000
## quantile 0.025      -0.89318351
## quantile 0.975       1.02027031
## randomized mv1-mv2  0.06220746
## nrandom             99.00000000
```

```
hist(out$rand)
abline(v=out$res[3,1], col="red")
abline(v=out$res[5,1], col="blue")
abline(v=out$res[6,1], col="blue")
```

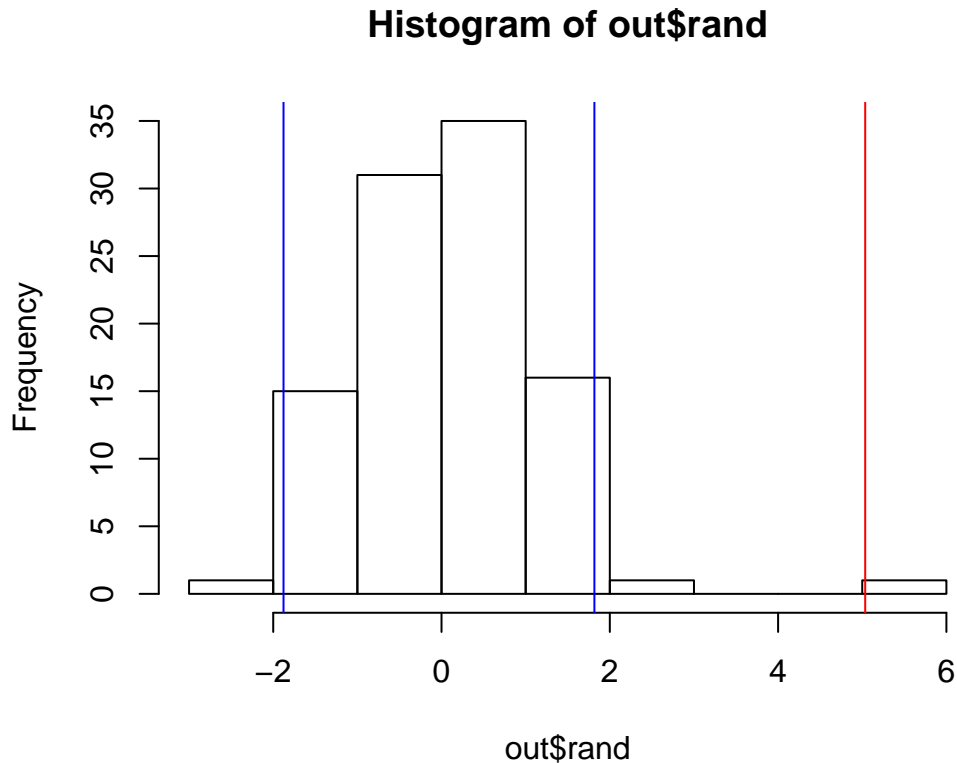
The observed difference lies between the quantiles.

In some cases, the sets of values to be compared can overlap. This is what happen in our second example below. We have recorded the maximum temperature at sites where either *Tomicus destruens* or *T. piniperda*, two closely related species of bark beetles, have been recorded. The species are sympatric in 4 sites which leads to an overlap *i.e.* values common to both species between the distributions to be compared. The values common to both populations are passed to the function `c2m` by the argument `pop3`.

```
data(Tomicus)
out <- c2m(pop1=Tomicus$destruens,pop2=Tomicus$piniperda,
pop3=Tomicus$both, nrandom=99)
out$res
```

```
##
## mv1          19.2432432
## mv2          14.2083333
## mv1-mv2       5.0349099
## p            0.0100000
## quantile 0.025 -1.8774775
## quantile 0.975  1.8168074
## randomized mv1-mv2 0.0698536
## nrandom      99.0000000
```

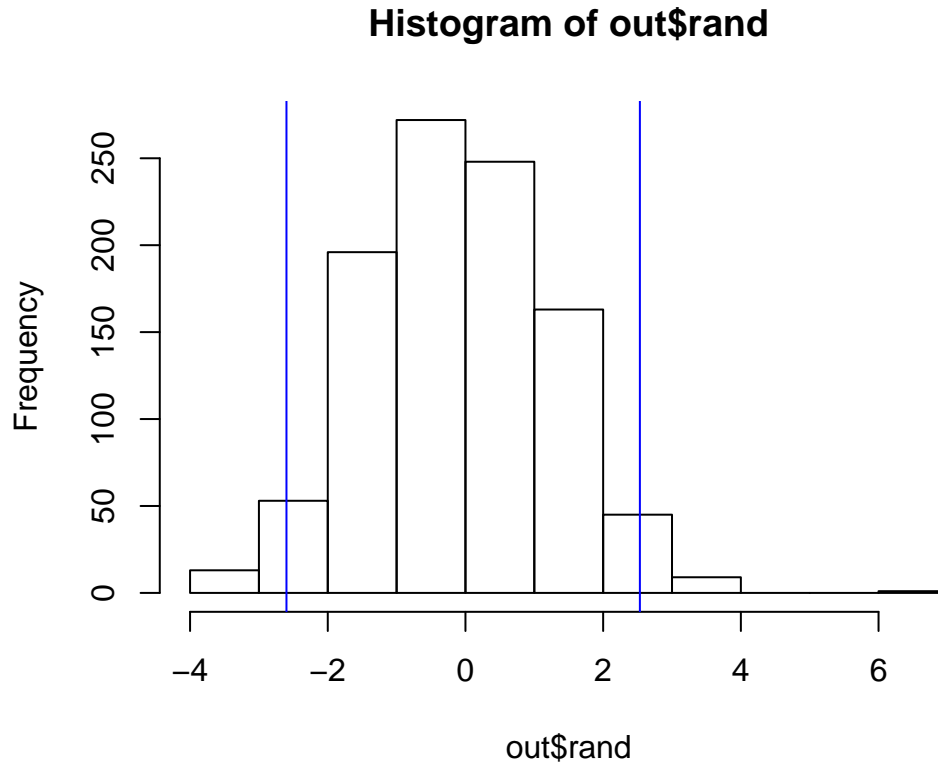
```
hist(out$rand)
abline(v=out$res[3,1], col="red")
abline(v=out$res[5,1], col="blue")
abline(v=out$res[6,1], col="blue")
```



The value of $mv1 - mv2$ lies outside the envelope defined by the quantiles which indicates a difference in species tolerance to temperature.

`c2m` can be used to make comparisons between two objects generated by the function `rich`:

```
data(ef)
o1 <- rich(matrix=ef, nrandom=99, verbose=TRUE)
data(ea)
o2 <- rich(matrix=ea, nrandom=99, verbose=TRUE)
out <- c2m(pop1=o1$sumrow, pop2=o2$sumrow, nrandom=999, verbose=TRUE)
hist(out$rand)
abline(v=out$res[3,1], col="red")
abline(v=out$res[5,1], col="blue")
abline(v=out$res[6,1], col="blue")
```



5 Rarefaction curves

5.1 rarc

`rich` allows to compute the rarefaction curve which corresponds to the changes in the species richness with sampling intensity. User selects the set of values of sampling size for which the estimate of species richness is needed, the number of randomizations to be performed and the values of the upper and lower bounds. `rarc` returns a data frame (`$out`) with the bootstrap estimates of species richness, the corresponding statistical envelop and the average number of individuals for the sample size to be investigated.

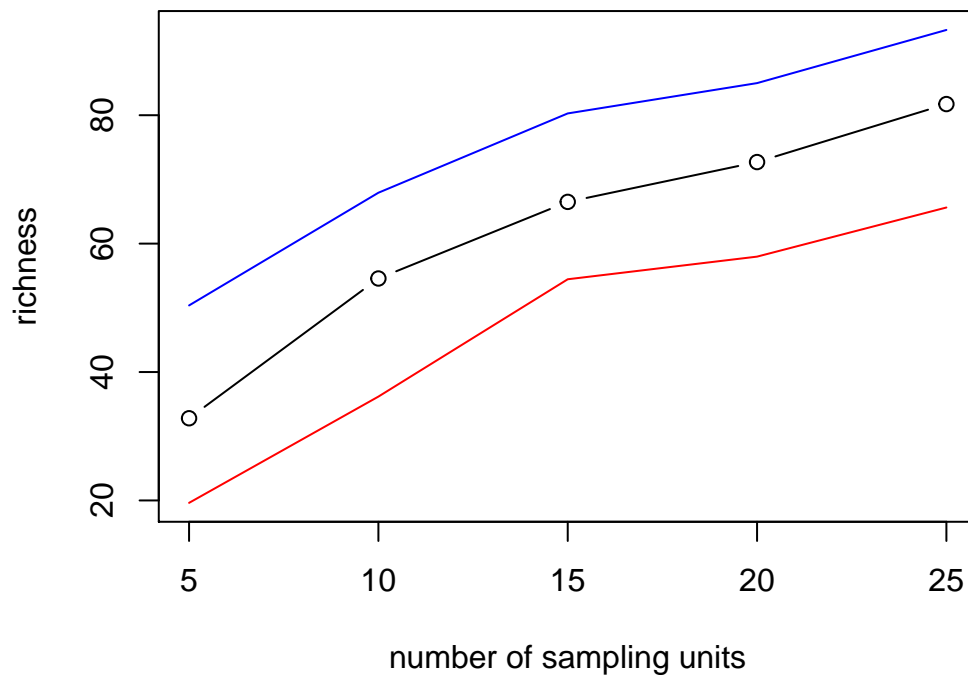
```
data(ef)
t <- rarc(ef,samplesize=c(5,10,15,20,25), nrandom=30, p1=0.975, p2=0.025)
head(t)
```

```
## $out
##   mean.richness lb.richness ub.richness mean.nb.individuals  samples
## 1      32.80000      19.625      50.375           411.7000 200.8661
## 2      54.56667      36.175      67.925           864.6667 260.6376
## 3      66.50000      54.450      80.275          1236.3667 313.5972
## 4      72.70000      57.975      85.000          1512.2000 300.2058
## 5      81.73333      65.625      93.275          1963.7000 472.5379
```

```
## sample
## 1      5
## 2     10
## 3     15
## 4     20
## 5     25
```

`t` can be used to plot the changes in richness according to sample size:

```
plot(t$out[,6],t$out[,1], type="b", ylim=range(c(t$out[,2],t$out[,3])),
      xlab="number of sampling units", ylab="richness")
points(t$out[,6] , t$out[,2], type="l", col="red")
points(t$out[,6] , t$out[,3], type="l", col="blue")
```



Note that the function uses bootstrap which means sampling with replacement. The consequence is that the richness estimated for a sample size equal to the size of the dataset is *not* equal to the observed richness, it is *lower*. For example the species richness of the dataset `ef` is 121 and the number of sampling units is 30:

```
data(ef)
r <- rich(ef)
r$cr
```

```
## [1] 121
```

```
dim(ef)[1]
```

```
## [1] 30
```

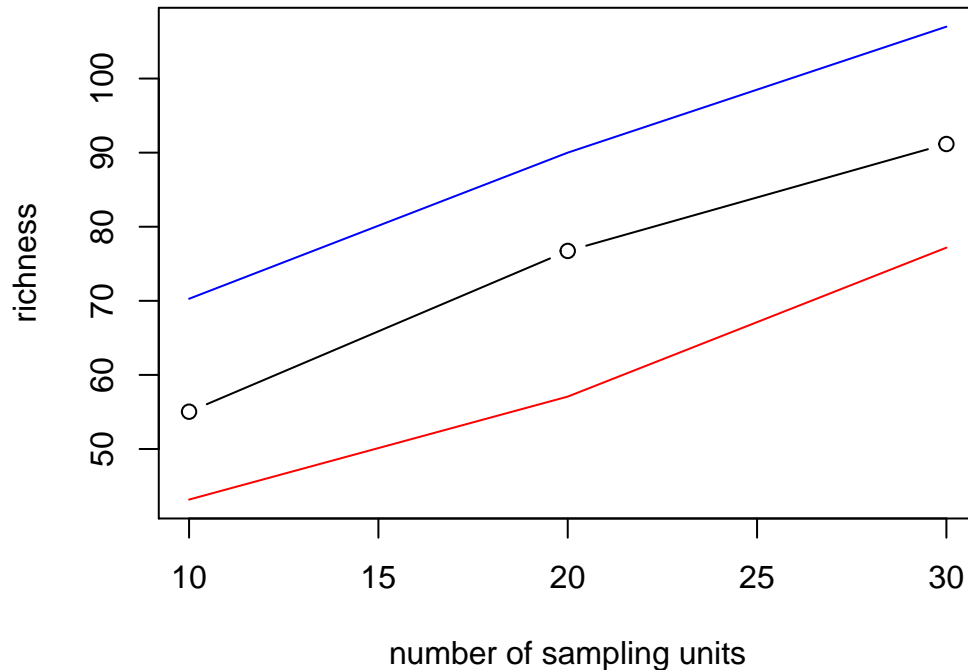
If we perform rarefaction curve for (say) samples of {10, 20, 30} we get:

```
data(ef)
t <- rarc(ef,samplesize=c(10, 20, 30), nrandom=30, p1=0.975, p2=0.025)
t

## $out
##   mean.richness lb.richness ub.richness mean.nb.individuals  samples
## 1      55.03333      43.175      70.275          812.3333 257.7822
## 2      76.73333      57.075      90.000         1735.4667 420.2093
## 3      91.16667      77.175     107.000         2430.7333 474.3358
##   sample
## 1      10
## 2      20
## 3      30
```

For $n = 30$ the bootstrap estimate of the richness is ca. 90 while the observed value is 121. If we plot the curve and add the observed value the difference is clear:

```
plot(t$out[,6],t$out[,1], type="b", ylim=range(c(t$out[,2],t$out[,3])),
     xlab="number of sampling units", ylab="richness")
points(t$out[,6] , t$out[,2], type="l", col="red")
points(t$out[,6] , t$out[,3], type="l", col="blue")
abline(h=r$cr, lty="dashed")
```



The interest of such estimates by bootstrap is that the variance of the estimate is meaningful whatever the sampling size which is not the case of rarefaction curves based on resampling with replacement. When replacement is not allowed, the variance decreases with increasing sampling size and becomes null for the maximum sampling size.

When the argument `save` is set to `TRUE` `rarc` returns an additional list (`$bootstrapped.val`) which corresponds to the raw bootstrapped values used to compute the quantiles. It may be useful for users who want to compute standard errors for example. The example below shows how to compute the standard errors.

```
# Computing the standard deviation instead of the quantiles.
# We set the save argument to TRUE
t <- rarc(ef,samplesize=c(5,10,15,20,25), nrandom=10, p1=0.975,
          p2=0.025, save=TRUE)

# The values of interest are in the list t$bootstrapped.val
# t$bootstrapped.val[[1]] contains the values for sample size #1
# t$bootstrapped.val[[2]] contains the values for sample size #2...
# t$bootstrapped.val[[3]][,1] corresponds to the richness
# t$bootstrapped.val[[3]][,2] corresponds to the number of individuals

# Computing the standard for the third sampling size
standard.error <- sd(t$bootstrapped.val[[3]][,1]) /
  sqrt(length(t$bootstrapped.val[[3]][,1]))

# compute the standard error for all sample sizes:
samplesize <- c(5,10,15,20,25)
```

```

stdev <- rep(NA, times=length(samplesize))
for (i in 1:(length(samplesize))) {
  stdev[i] <- sd(t$bootstrapped.val[[i]][,1])/
    sqrt(length(t$bootstrapped.val[[i]][,1]))
}

```

We plot the results:

```

r <- range(t$out$mean.richness-stdev, t$out$mean.richness+2*stdev)
r

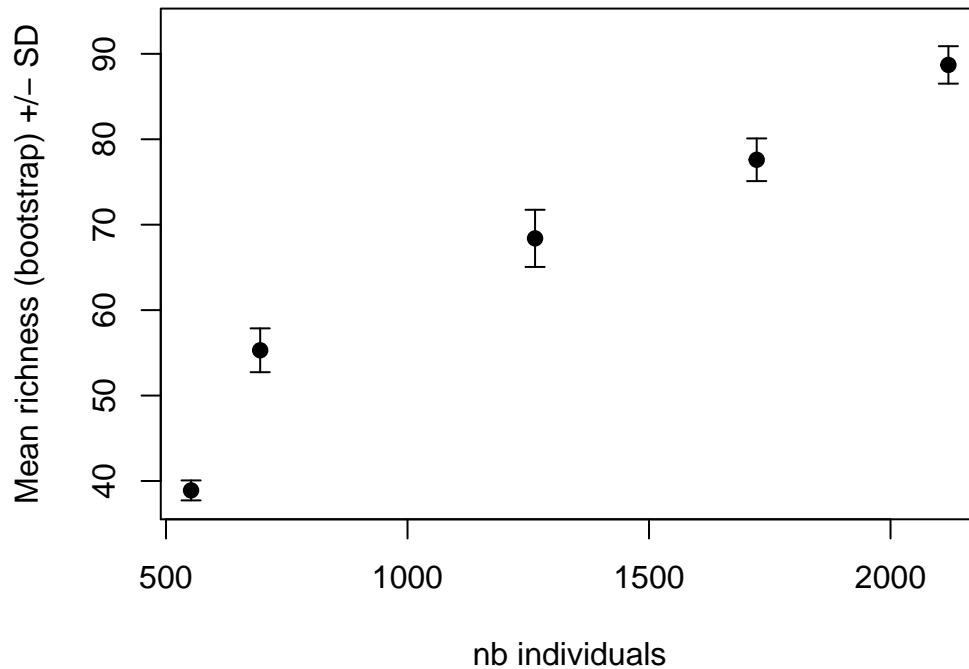
```

```
## [1] 37.73143 93.08229
```

```

plot(t$out$mean.nb.individuals, t$out$mean.richness, pch=19, ylim=r,
xlab="nb individuals", ylab="Mean richness (bootstrap) +/- SD")
arrows(t$out$mean.nb.individuals, t$out$mean.richness-stdev,
t$out$mean.nb.individuals, t$out$mean.richness+stdev,
length=0.05, angle=90, code=3)

```



5.2 raref

`raref` computes the rarefaction curve and interpolates the species richness corresponding to a given density of individuals (not a number of samples!).

```
data(ef)
r <- raref(ef, dens=1100, nrandom=50)
head(r$rar)
```

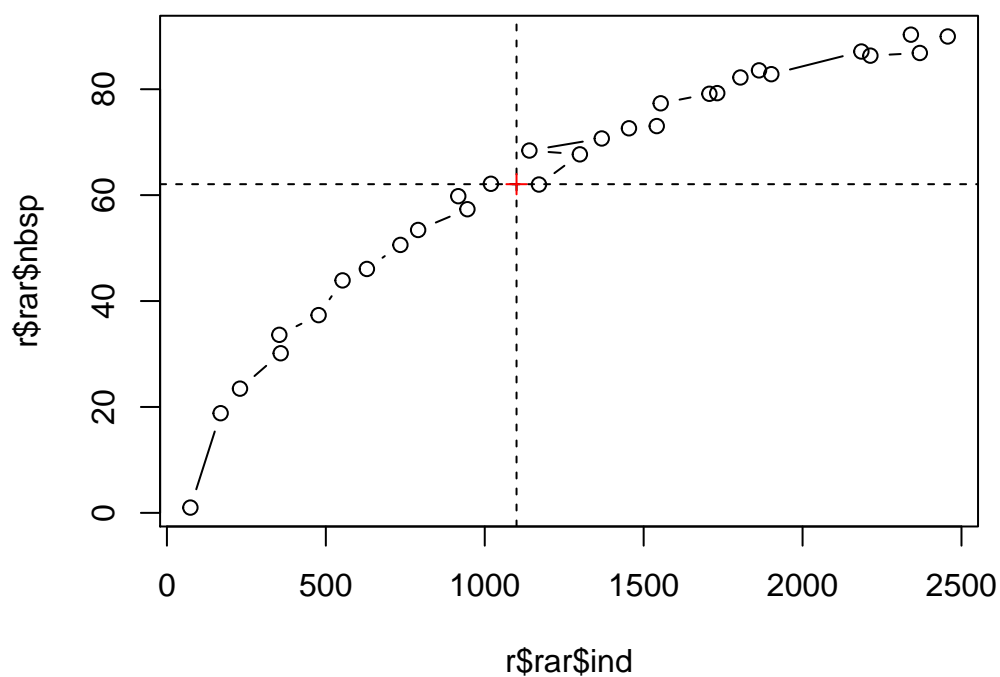
```
##      nbsp      ind sample
## 1  1.00   73.84      1
## 2 18.82 168.96      2
## 3 23.46 229.88      3
## 4 30.14 357.96      4
## 5 33.62 353.64      5
## 6 37.34 477.28      6
```

```
r$Sinterp
```

```
## [1] 1100.00000 62.06532
```

We plot the curve and the interpolated value:

```
plot(r$rar$ind, r$rar$nbsp, type="b")
abline(v=r$Sinterp[1], lty="dashed") ; abline(h=r$Sinterp[2], lty="dashed")
points(r$Sinterp[1], r$Sinterp[2], pch=3, col="red")
```



5.3 raref2

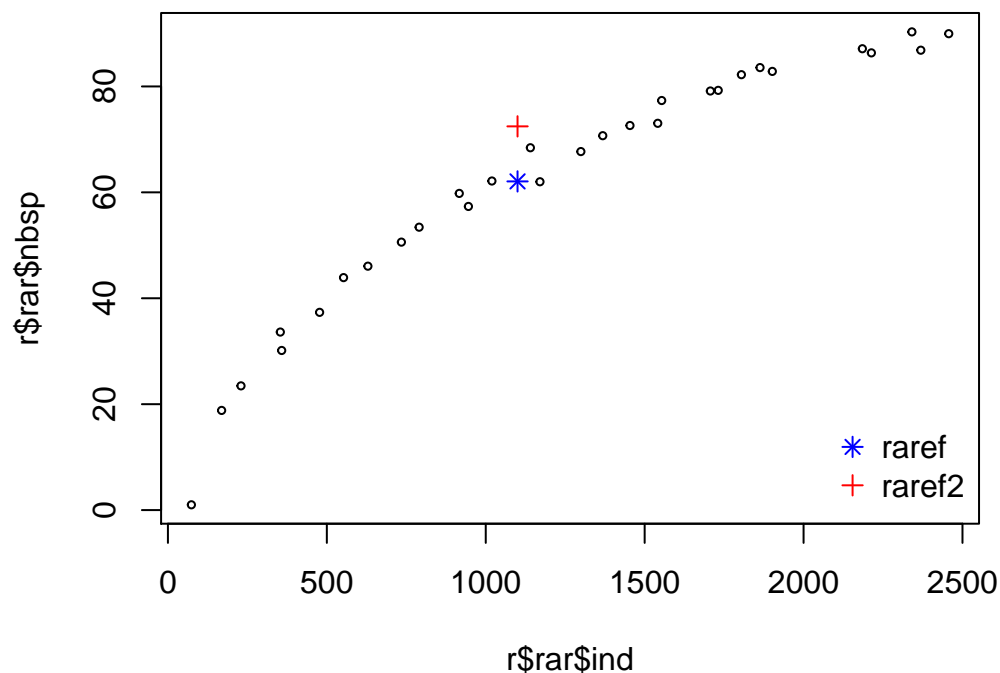
raref2 computes another estimation of the species richness by thinning the data matrix so that the overall corresponding density is comprised in a fixed interval.

```
data(ef)
r2 <- raref2(matrix=ef, dens=1100, tolerance=0.01, nrandom=50)
r2$mean.boot
```

```
## [1] 72.46
```

We can add this second estimate to the rarefaction curve derived from the outputs of raref:

```
plot(r$rar$ind, r$rar$nbsp, type="p", cex=0.5)
points(r$Sinterp[1], r$Sinterp[2], pch=8, col="blue")
points(1100, r2$mean.boot, pch=3, col="red")
legend(x="bottomright", legend=c("raref", "raref2"), bty="n",
      pch=c(8,3), col=c("blue", "red"))
```



5.4 shared

shared computes the richness of each group of sample depicting a community, the number of species shared by pairs of communities and the total number of species for each pairs of community. Two or more communities can be compared :

```

sp1<-c(1,2,3,4,5)
sp2<-c(0,0,0,0,0)
sp3<-c(1,1,0,0,0)
sp4<-c(0,0,0,0,0)
site1<-cbind(sp1, sp2, sp3, sp4)
colnames(site1)<-c("sp1", "sp2", "sp3", "sp4")
site1

```

```

##      sp1 sp2 sp3 sp4
## [1,]  1  0  1  0
## [2,]  2  0  1  0
## [3,]  3  0  0  0
## [4,]  4  0  0  0
## [5,]  5  0  0  0

```

```

sp1<-c(1,2,3)
sp2<-c(1,0,0)
sp3<-c(0,0,0)
sp4<-c(0,0,0)
site2<-cbind(sp1, sp2, sp3, sp4)
colnames(site2)<-c("sp1", "sp2", "sp3", "sp4")
site2

```

```

##      sp1 sp2 sp3 sp4
## [1,]  1  1  0  0
## [2,]  2  0  0  0
## [3,]  3  0  0  0

```

```

sp1<-c(1,2,3,4)
sp2<-c(1,0,0,0)
sp3<-c(1,0,0,0)
sp4<-c(1,0,0,0)
site3<-cbind(sp1, sp2, sp3, sp4)
colnames(site3)<-c("sp1", "sp2", "sp3", "sp4")
site3

```

```

##      sp1 sp2 sp3 sp4
## [1,]  1  1  1  1
## [2,]  2  0  0  0
## [3,]  3  0  0  0
## [4,]  4  0  0  0

```

```
# we create a list containing the sites to be compared:
data<-list(site1,site2, site3)
names(data)<-c("site1","site2","site3")

shared(data)
```

```
##      site1 site2 site3
## site1     2     1     2
## site2     3     2     2
## site3     4     4     4
```

shared retruns a matrix whose diagonal is the richness of each community.

```
data(efeb)
shared(efeb)
```

```
##      ef eb
## ef 121  9
## eb 134 22
```

References

Manly, B.F.J. 1997. *Randomization and Monte Carlo Methods in Biology*. Chapman & Hall.

Oksanen, Jari, F. Guillaume Blanchet, Roeland Kindt, Pierre Legendre, Peter R. Minchin, R. B. O'Hara, Gavin L. Simpson, Peter Solymos, M. Henry H. Stevens, and Helene Wagner. 2016. *Vegan: Community Ecology Package*. <https://CRAN.R-project.org/package=vegan>.