Bootstrap 01 - A tutorial

Josep Fortiana 2019-10-29

The *bootstrap* is a generic name for a family of computationally intensive procedures in Statistics based on generating from a single dataset a collection of simulated datasets *(resamples),* from which it is possible to infer information about the stochastic mechanism that originated the dataset.

1. Resamples and *Out-Of-Bag* observations

A *resample* from a set x of n observations is a sample of size n of elements from x, extracted with replacement, that is, after each extraction, the extracted element is replaced to the set and can be chosen again. Thus a given element x_1 can appear more than once in the resample (also it may happen it does not appear). Example:

```
set.seed(24025)
x<-1:12
n<-length(x)
x1<-sample(x,n,replace = TRUE)
x1
```

```
##  [1]  6  4  6  2  3  9  7  6 10  6  4 10
```

Each resample contains n elements, the same as the original set x. Some elements appear twice or more, while other elements of x do not appear. El conjunto de observaciones que no aparece se suele llamar *OOB (Out-of-bag).* Podemos detectarlas haciendo:

```
oob<-x[is.na(match(x,x1))]
oob
```

```
## [1]  1  5  8 11 12
```

2. An example of a simple bootstrap

Failures of Air-conditioning Equipment

Proschan (1963) reported on the times T in hours, between failures of the air-conditioning equipment in 10 Boeing 720 aircraft. The following dataset also apears as `aircondit` in the `boot` package. It contains the intervals for the ninth aircraft.

```
t<-c(3,5,7,18,43,85,91,98,100,130,230,487)
n<-length(t)
```

We want to study T.

For instance, to decide with a given significance level $\alpha = 0.05$, whether the expected value $\tau = \mathrm{E}(T)$ is less than $\tau_0 = 110$ hours, the specification in the maintainance and repair contract.

The observed empirical average $\bar{t}$ is:

```
t.bar<-mean(t)
round(t.bar,4)
```

```
## [1] 108.0833
```

Classical approach to the problem

Steps:

- 1. Try and find some (known) probability distribution to model this dataset and, eventually, perform a goodness-of-fit test.

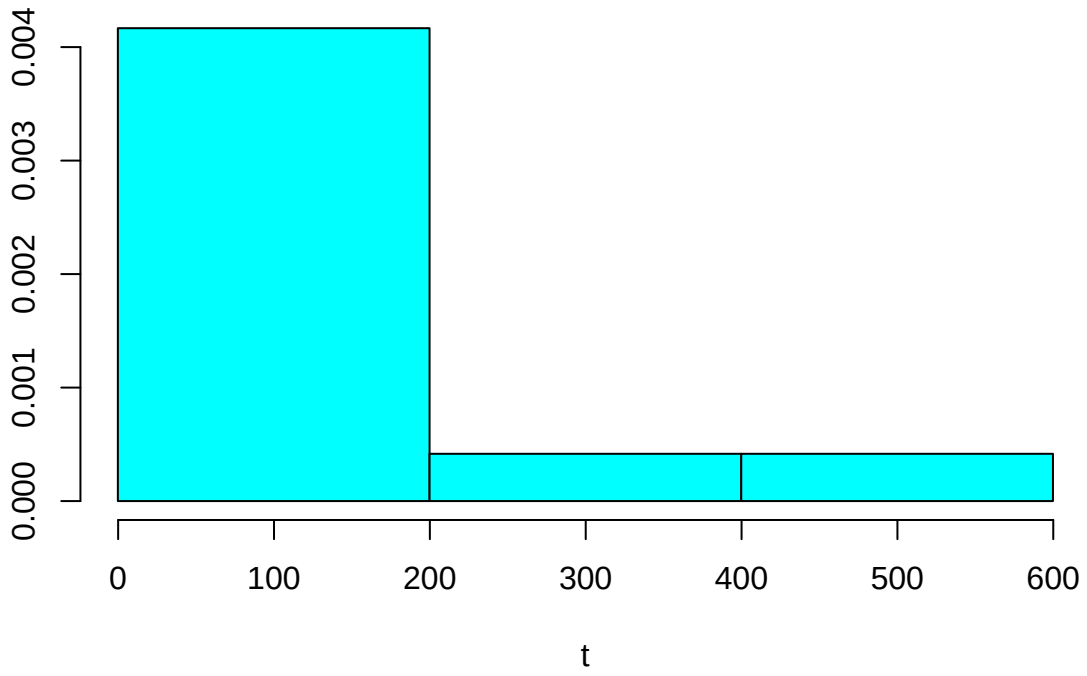
- 2. Estimate parameters (if any) in the given distribution.
- 3. Devise a hypotheses test of $\overline{T} < \tau_0$ versus $\overline{T} \geq \tau_0$, in particular designing (finding) a test statistic Z as a function of the sample, deriving its probability distribution and, by means of it, select an acceptance region $\{Z \in A_1\}$.
- 4. Finally, for the observed sample, compute the observed value z of Z and take the appropriate decision.

Histogram

```
require(MASS)
```

```
## Loading required package: MASS
```

```
options(repr.plot.width=4,repr.plot.height=4)
truehist(t)
```



The shape bears a close resemblance to an exponential distribution.

$$f(t|\lambda) = \lambda \exp\{-\lambda t\}, \quad t > 0, \quad \lambda > 0.$$

In an exponential model Exp(λ) the maximum likelihood estimator of $\tau = 1/\lambda$ is the empirical average $\overline{t} = 108.0833$.

QQ plot

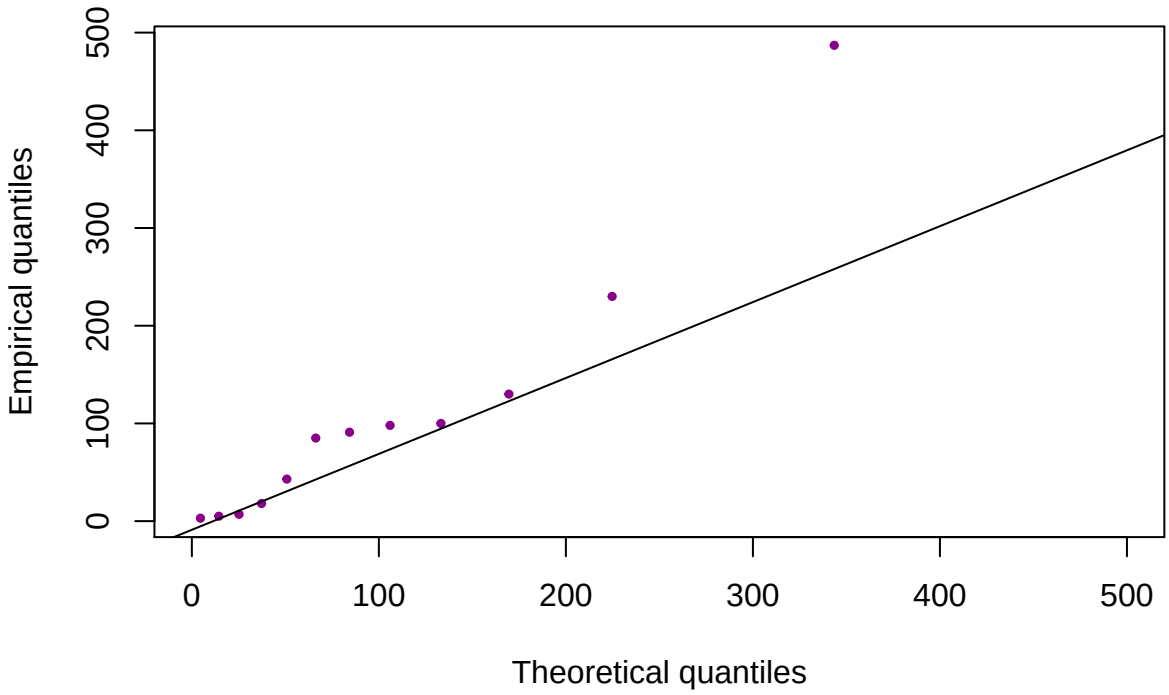
Visually compare the empirical (observed) distribution to the theoretical one, Exp(λ).

Should there be a good agreement, points in the plot would lie on the line.

Exercise: do the same for a small data set generated from some given distribution.

```
# Generate a sample of n=12 values from an exponential distribution with lambda=1/t.bar and obtain its
#
#
```

```
options(repr.plot.width=4,repr.plot.height=4)
qqplot(qexp(ppoints(12),rate=1/t.bar),t,xlim=c(0,500),pch=19,cex=0.5,col="DarkMagenta",
    xlab="Theoretical quantiles",ylab="Empirical quantiles")
qqline(t,distribution=function(p){qexp(p,rate=1/t.bar)})
```



Kolmogorov-Smirnov goodness-of-fit test to check whether we can accept the exponential model.

```
ks.test(t,pexp,1/t.bar)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  t
## D = 0.18729, p-value = 0.7282
## alternative hypothesis: two-sided
```

Accepting the exponential model, and from the property that a sum of independent exponential variates with the same parameter $\tau = 1/\lambda$,

$$Z = \sum_{i=1}^{n} T_i$$

follows a Gamma$(n, 1/\tau)$ distribution, we use this distribution to make inference about τ.

For the observed sample,

```
z<-sum(t)
round(z,2)
```

```
## [1] 1297
```

and the critical p-value to decide on the null hypothesios $H_0 : \tau < \tau_0 = 110$ is:

```
tau.0<-110
p.val<-1-pgamma(z,shape=n,scale=tau.0)
round(p.val,3)
```

```
## [1] 0.486
```

Bootstrap approach to the same problem

We generate a number N of *resamples*, each of them of equal length $n = 12$ as the observed sample t.

Each resample is obtained by selecting n elements of the set t with equal probability $1/n$ and *with replacement.*

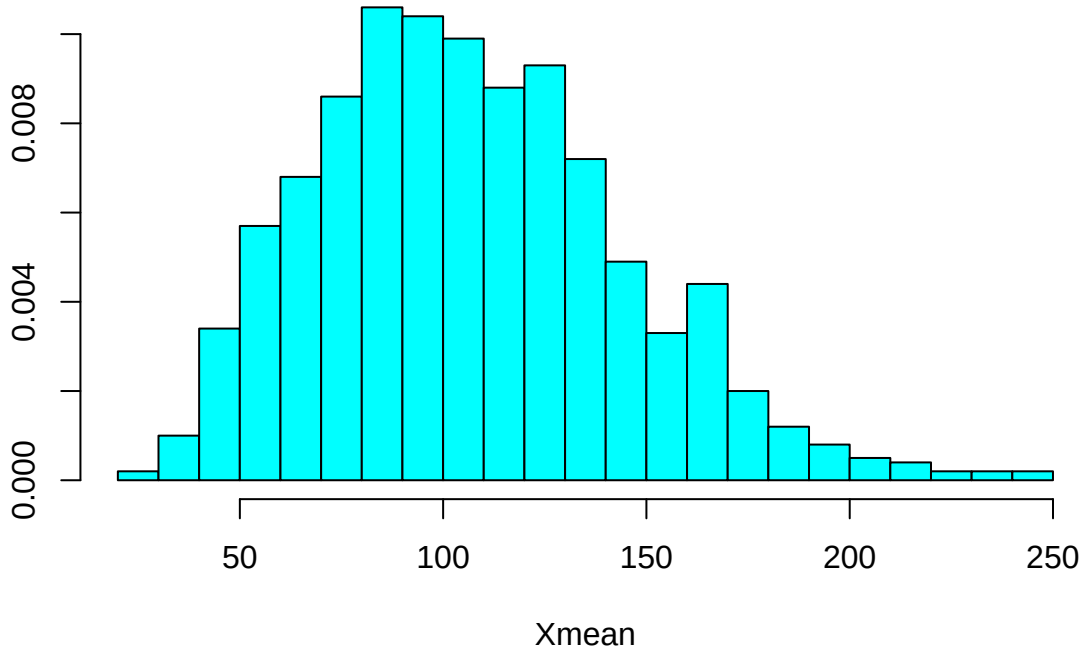
For any statistic $U \equiv U(t)$, function of the sample t, the collection of N values resulting from applying it to the collection of N resamples is the *bootstrap sample* of U:

$$u_1, \ldots, u_N.$$

```
# A function to generate resamples, returning a matrix with a resample in each row:
resample<-function(x,N){
    n<-length(x)
    X<-matrix(0,nrow=N,ncol=n)
    for (i in 1:N){
        X[i,]<-sample(x,n,replace = TRUE)
        }
    return(X)
    }
```

For instance we generate $N = 1000$ resamples of the air conditioning data and, for each of them we compute the arithmetic mean:

```
X<-resample(t,1000)
Xmean<-apply(X,1,mean)
options(repr.plot.width=4,repr.plot.height=4)
truehist(Xmean)
```



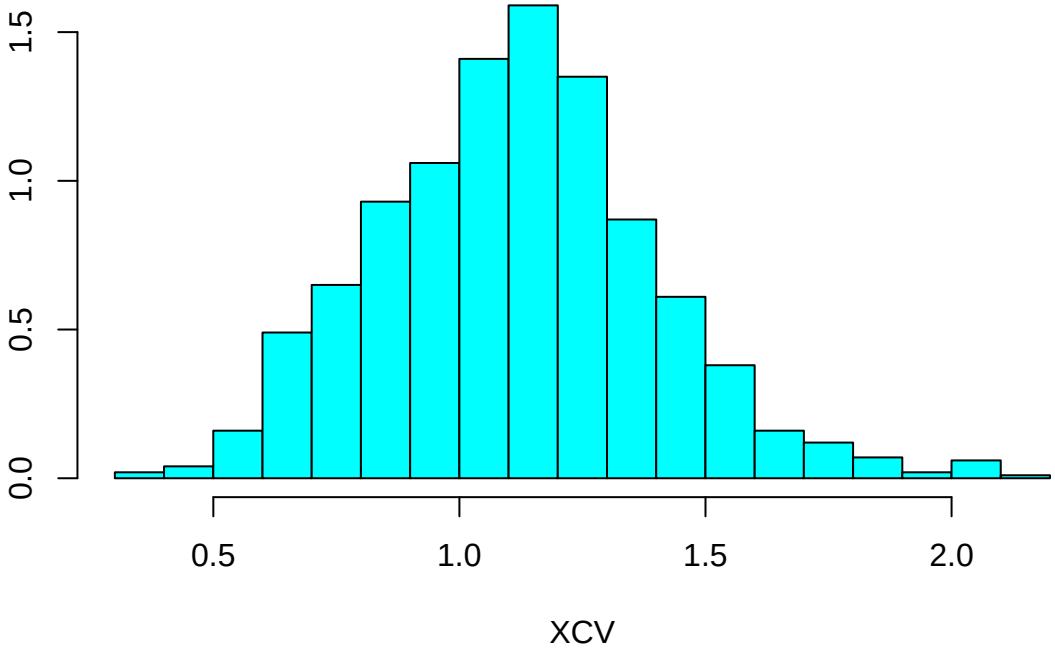
The bootstrap-empirical p-value, $p^\star$, is the relative frequency (proportion) within the bootstrap sample `Xsum` of values greater than the actually observed value $z = 1297$.

```
Xsum<-apply(X,1,sum)
p.star<-sum(Xsum>z)/1000
round(p.star,3)
```

```
## [1] 0.454
```

Other more complicated statistics such as the coefficient of variation CV $= \sigma/\mu$, the ratio of the standard deviation σ, whose theoretical distribution appears unthinkable to derive, can be studied in the same way within the bootstrap 0framework.

```
X<-resample(t,1000)
Xmean<-apply(X,1,mean)
Xsd<-apply(X,1,sd)
XCV<-Xsd/Xmean
options(repr.plot.width=4,repr.plot.height=4)
truehist(XCV)
```



3. Example of bootstrap in ISLR, pp. 189-191

```
#install.packages("ISLR",dependencies=TRUE,repos="https://cloud.r-project.org")
require(ISLR)
```

```
## Loading required package: ISLR
```

```
#install.packages("boot",dependencies=TRUE,repos="https://cloud.r-project.org")
require(boot)
```

```
## Loading required package: boot
```

`Portfolio` dataset

Description

A simple simulated data set containing 100 returns for each of two assets, X and Y. The data is used to estimate the optimal fraction to invest in each asset to minimize investment risk of the combined portfolio. One can then use the Bootstrap to estimate the standard error of this estimate.

Format

A data frame with 100 observations on the following 2 variables.

1. `X`: Returns for Asset X
2. `Y`: Returns for Asset Y

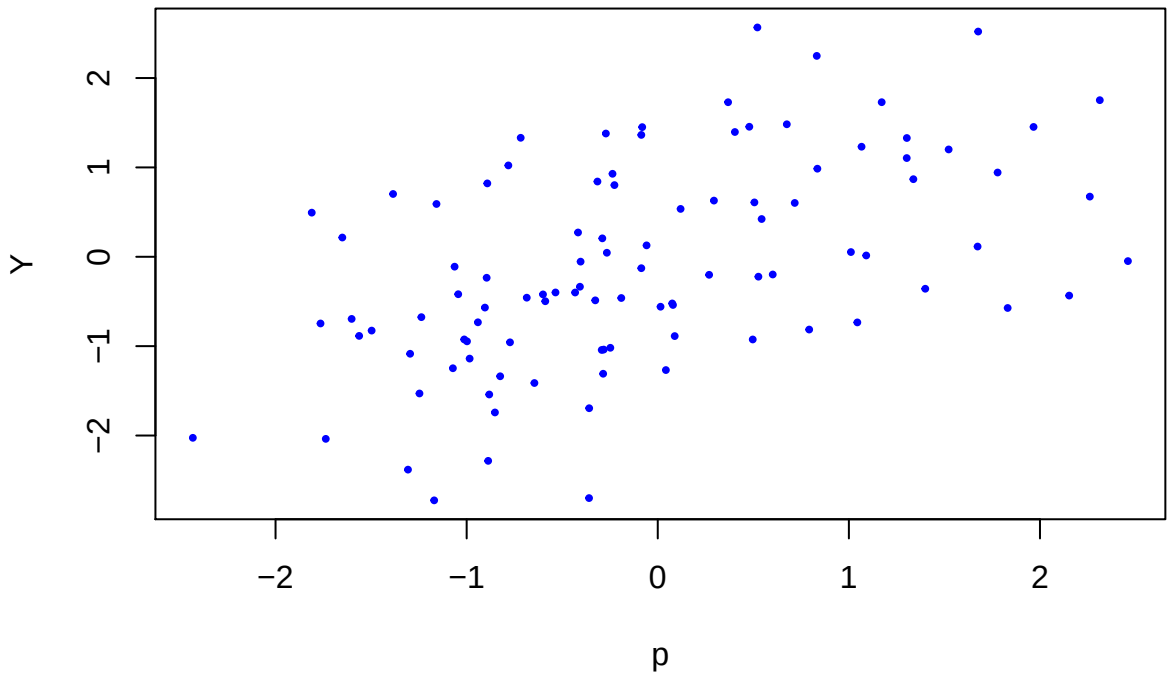
Source

Simulated data

```
data(Portfolio)
str(Portfolio)
```

```
## 'data.frame':    100 obs. of  2 variables:
##  $ X: num  -0.895 -1.562 -0.417 1.044 -0.316 ...
##  $ Y: num  -0.235 -0.885 0.272 -0.734 0.842 ...
```

```
options(repr.plot.width=4,repr.plot.height=4)
plot(Portfolio,"p",pch=19,col="blue",cex=0.4)
```




```
# Proportion of X for a two assets portfolio with a minimum risk (minimum variance)
alpha.fn=function(data,index){
 X=data$X[index]
 Y=data$Y[index]
 return((var(Y)-cov(X,Y))/(var(X)+var(Y)-2*cov(X,Y)))
 }
alpha0<-alpha.fn(Portfolio,1:100)
round(alpha0,3)
```

```
## [1] 0.576
```

```
# ---- SNIP ---- SNIP ---- SNIP ----
# W=Var/(sigma.Y^2); rho=sigma.XY/(sigma.X*sigma.Y); t =sigma.X/sigma.Y
W<-function(alpha,rho,t){
    return(alpha^2*t^2+(1-alpha)^2+2*alpha*(1-alpha)*rho*t)
```

```
    }

alpha.max<-function(rho,t){
    return((1-rho*t)/(1+t^2-2*rho*t))
    }
```

```
index<-1:100
X=Portfolio$X[index]
Y=Portfolio$Y[index]
sigma.X<-sqrt(var(X))
sigma.Y<-sqrt(var(Y))
t<-sigma.X/sigma.Y
rho<-cor(X,Y)
alpha.hat<-alpha.max(rho,t)
round(sigma.X,3)
```

```
## [1] 1.062
```

```
round(sigma.Y,3)
```

```
## [1] 1.144
```

```
round(t,3)
```

```
## [1] 0.929
```

```
round(alpha.hat,3)
```

```
## [1] 0.576
```

```
W.rho.t<-function(rho,t){

}
```

```
set.seed(1)
alpha.fn(Portfolio,sample(100,100,replace=TRUE))
```

```
## [1] 0.7368375
```

```
boot(Portfolio,alpha.fn,R=1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Portfolio, statistic = alpha.fn, R = 1000)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* 0.5758321 -0.001695873  0.09366347
```

4. *Bootstrap* in evaluating prediction methods

There are several ways of using *bootstrap* in evaluating prediction models. A sensible procedure is, for each resample in a sufficiently large collection of N generated resamples, use the resample itself as the *train* subset and the *OOB* subset (individuals in the original sample not present in the resample) as *test* subset.

Then the *bootstrap estimate of the prediction error,* `E.boot`, is the average of the N error proportions.

Efron's 0.632 **rule**

The *naif* procedure, to use the same set both as `train`and `test`, yields another estimate `E.subst`, which has an optimistic bias, due to the fact that the test data are in a way already known by the prediction algorithm. On the other hand, it is known that `E.boot` has a pessimistic bias.

Bradley Efron, the discoverer of bootstrap, proposed the idea to use a weighted mean of both estimators, as a way of compensating both biases. This is explained in two articles, Efrom (1983), *Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation,* and Efron (1997) *Improvements on Cross-Validation: The .632+ Bootstrap Method.*

The 0.632 rule described there, is to use the estimate `E.632` defined as:

$$E.632 = 0.632 * E.boot_1 + 0.368 * E.subst,$$

where $E.boot_1$ is a *leave-one-out bootstrap estimator,* slightly different from the $E.boot$ described above. The motivation for using 0.632 as the weight comes from the approximate computation of the probability that a given individual in a sample of n appears in a resample:

$$1 - (1 - 1/n)^n \approx 1 - 1/e = 0.6321206.$$