Introduction to bootstrap

Kevin Mottin & Vincent Vandewalle

December 2023

Contents

TL;DR : The key idea of the boostrap is to use random draws in the data itself to mimic sampling fluctuations and to derive an approximation of the variance of the estimators.

Pedagogical goals :

1. Be able to sample from the empirical distribution of the data
2. Be able to explain the principles of the bootstrap
3. Be able to design and run an empirical bootstrap to calculate confidence intervals
4. Be able to design and run a parametric bootstrap to calculate confidence intervals

1 Introduction

1.1 Base of the bootstrap

The data "pulling itself up by its own bootstrap".

These methods, although very simple to implement, would not be possible without modern computing facilities. One of the main applications of the boostrap is the calculation of confidence intervals.

Idea : confidence intervals on some parameter μ need the knowledge of the distribution F. How to compute confidence intervals on μ when F is not known ? One solution is to use the bootstrap!

Question : Recall other strategies to obtain confidence intervals?

Figure 1: Illustation

The bootstrap can be used to obtain confidence intervals on other statistics such as: the median, other percentiles or the truncated mean.

Notion of empirical distribution of data noted F^* or called resampling distribution.

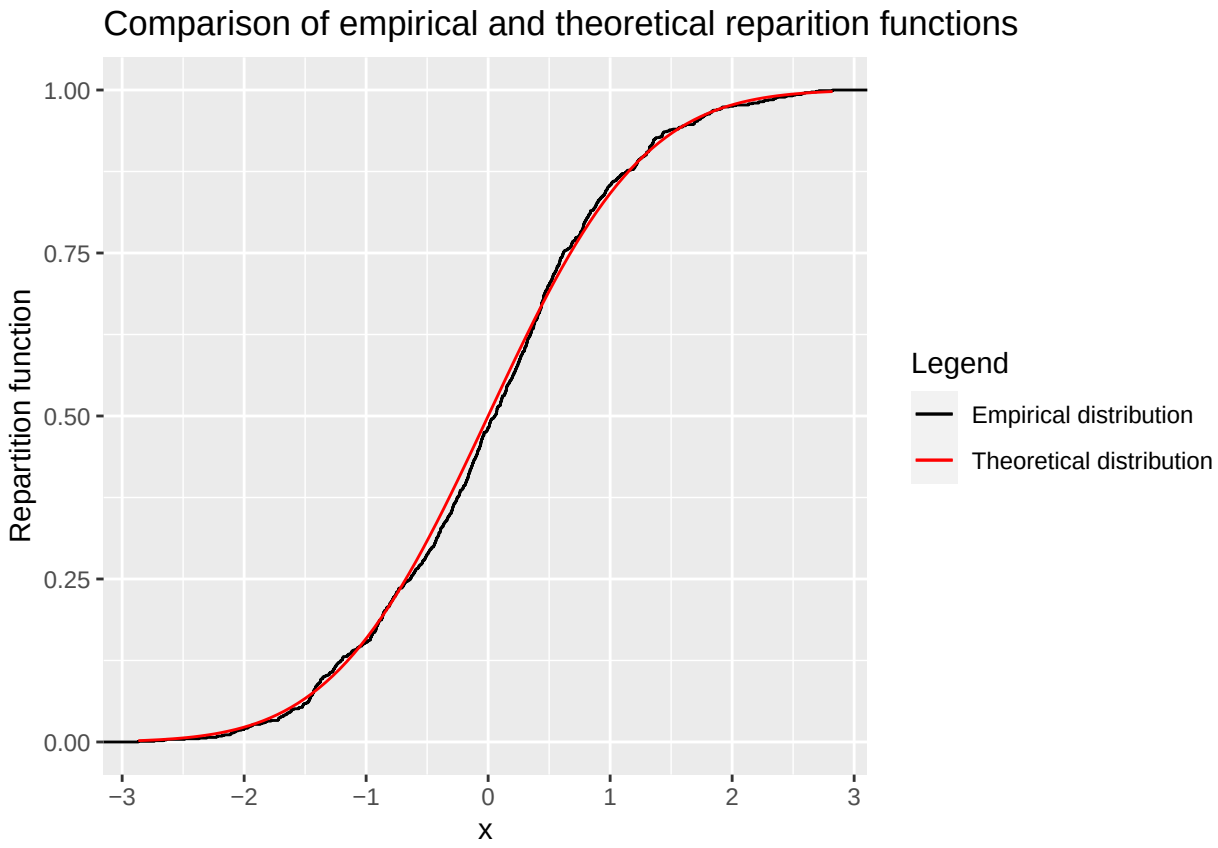
The law of large numbers guarantees that if we have enough data, F^* is a good approximation of F.

```r
library(ggplot2)

df <- data.frame(x = c(rnorm(1000)))

colors <- c("Empirical distribution"="black","Theoretical distribution" = "red")

ggplot(df, aes(x)) +
  stat_ecdf(geom = "step", aes(color = "Empirical distribution")) +
   geom_function(fun = pnorm, aes(color = "Theoretical distribution")) +
   labs(x = "x",
         y = "Repartition function",
         color = "Legend") +
  scale_color_manual(values = colors) +
  ggtitle("Comparison of empirical and theoretical reparition functions")
```



Examples on ten sampling according a uniform distribution over the integers from 1 to 8:

```
x = factor(sample(8, 10, replace = TRUE), levels = 1:8)
table(x)
```

```
## x
## 1 2 3 4 5 6 7 8
## 4 0 2 1 0 3 0 0
```

```
# Empirical frenquency
prop.table(table(factor(sample(8, 10, replace = TRUE), levels = 1:8)))
```

```
##
##   1   2   3   4   5   6   7   8
## 0.0 0.1 0.1 0.3 0.1 0.0 0.2 0.2
```

```
# Theoretical frequency
th <- rep(1/8,8)
names(th) <- 1:8
th
```

```
##     1     2     3     4     5     6     7     8
## 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125
```

1.2 Bootstrap empirical or parametric

In the following, we will distinguish between two main versions of the bootstrap:

- the **empirical bootstrap** which consists in resampling directly in the data, it does not require any assumption on the distribution of the data

- the **parametric bootstrap** which consists in inferring a parametric model on the distribution of the data, then drawing new data via this parametric model

In practice you will encounter more empirical approaches. However, when the number of data is small, parametric approaches can be particularly useful, where non-parametric approaches require more data.

1.3 Case of the empirical bootstrap

1.3.1 Re-sampling

Resampling in the data with replacement:

1. The same data may reappear several times in the resample
2. You can simulate a sample of any size you want

Star notation:

Assume a sample of size n.

$$x_1, x_2, \ldots, x_n$$

Then the resample of size m is denoted by

$$x_1^*, x_2^*, \ldots, x_m^*$$

The average on resampled data is denoted by $\bar{x}^*$.

1.3.2 Principle of the empirical bootstrap

Create a new sample of the same size as the initial data.

Whatever the u statistic calculated on the initial data, we are able to calculate the u^* statistic on the bootstrap sample.

The empirical bootstrap follows the following steps:

1. $x_1, x_2, \ldots, x_n$ is a sample from the F distribution
2. u is the statistic computed from the sample
3. F^* is the empirical distribution of the data
4. $x_1^*, x_2^*, \ldots, x_n^*$ is a resample of the same size as the original data
5. u^* is the statistic computed from the resample

The basics of the bootstrap are:

1. $F^* \approx F$
2. The variations of u are well approximated by the variations of u^* (this will be used for the calculation of confidence intervals)

1.3.3 Calculs of confidence intervals

We would like to know the distribution of

$$\delta = \bar{x} - \mu$$

If we knew this distribution we could obtain

$$P(\delta_{1-\alpha/2} \leq \bar{x} - \mu \leq \delta_{1-\alpha/2}|\mu) = 1 - \alpha \Leftrightarrow P(\bar{x} - \delta_{1-\alpha/2} \geq \mu \geq \bar{x} - \delta_{1-\alpha/2}|\mu) = 1 - \alpha$$

This gives the confidence interval of amplitude $1 - \alpha$:

$$[\bar{x} - \delta_{\alpha/2}; \bar{x} - \delta_{1-\alpha/2}]$$

Warning : as a reminder, it is the interval that is random.

In the bootstrap the distribution of δ is approximated by the distribution of δ^* defined by

$$\delta^* = \bar{x}^* - \bar{x}$$

The distribution of δ^* can be estimated as precisely as one wishes by resampling enough times in the initial data.

1.3.4 Illustration of the empirical boostrap for the calculation of confidence intervals on the mean

The data considered are

```
x = c(30,37,36,43,42,43,46,41,42)
n = length(x)
```

The sample mean is :

```
xbar = mean(x); xbar
```

```
## [1] 40
```

The number of bootstrap samples is set to 20:

```
nboot = 20
```

We generate 20 bootstrap samples, i.e. an array of dimensions $n \times 20$ of random samples of `x` :

```
tmpdata = sample(x,n*nboot, replace=TRUE)
bootstrapsample = matrix(tmpdata, nrow=n, ncol=nboot)
bootstrapsample
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]   36   36   41   36   41   42   36   41   43    42    43    43    37
##  [2,]   42   41   36   43   41   36   43   42   42    43    46    42    46
##  [3,]   43   42   42   42   46   42   43   30   46    41    36    42    42
##  [4,]   41   43   46   46   30   42   37   42   43    36    42    42    42
##  [5,]   30   42   42   42   30   42   42   46   42    30    37    30    42
##  [6,]   42   42   36   30   43   37   36   37   46    43    36    30    46
##  [7,]   43   42   37   43   43   37   43   42   43    42    30    37    43
##  [8,]   42   43   43   42   30   42   43   36   41    41    43    37    42
##  [9,]   36   30   36   36   43   37   30   46   42    36    42    43    37
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20]
##  [1,]    42    37    46    42    37    37    43
##  [2,]    43    43    36    42    41    42    37
##  [3,]    42    46    42    41    43    46    42
##  [4,]    42    42    37    46    30    43    36
##  [5,]    36    41    43    46    36    43    43
##  [6,]    37    43    46    37    30    42    46
##  [7,]    42    30    36    42    43    43    42
##  [8,]    43    37    37    30    37    36    43
##  [9,]    41    42    36    36    30    30    42
```

For each bootstrap sample we compute the empirical mean x^* :

```
bsmeans = colMeans(bootstrapsample)
bsmeans
```

```
##  [1] 39.44444 40.11111 39.88889 40.00000 38.55556 39.66667 39.22222 40.22222
##  [9] 43.11111 39.33333 39.44444 38.44444 41.88889 40.88889 40.11111 39.88889
```

```
## [17] 40.22222 36.33333 40.22222 41.55556
```

We deduce δ^* associated to each bootstrap sample:

```
deltastar = bsmeans - xbar
```

We deduce the empirical quantiles of δ^* in 0.1 and 0.9 :

```
d = quantile(deltastar, c(0.1, 0.9))
d
```

```
##       10%       90%
## -1.455556  1.588889
```

Finally, we obtain the 80 % confidence interval on the expectation by :

```
ci = xbar - c(d[2], d[1])
cat('Confidence interval: ',ci, '\n')
```

```
## Confidence interval:  38.41111 41.45556
```

In an equivalent way we can write :

```
2*xbar - quantile(bsmeans, c(0.9,0.1))
```

```
##      90%      10%
## 38.41111 41.45556
```

Alternative: determine the quantiles by hand (without the quantile function), for a better understanding of what is being done:

```
sorteddeltastar = sort(deltastar)
# Sorted result
hist(sorteddeltastar, nclass=6)
```

```r
print(sorteddeltastar)
```

```
##  [1] -3.6666667 -1.5555556 -1.4444444 -0.7777778 -0.6666667 -0.5555556
##  [7] -0.5555556 -0.3333333 -0.1111111 -0.1111111  0.0000000  0.1111111
## [13]  0.1111111  0.2222222  0.2222222  0.2222222  0.8888889  1.5555556
## [19]  1.8888889  3.1111111
```

```r
# Find the critical values .1 and .9 for deltastar
d9alt = sorteddeltastar[2]
d1alt = sorteddeltastar[18]
# Find and display the 80% interval for the mean
ciAlt = xbar - c(d1alt,d9alt)
cat('Alternative confidence interval: ',ciAlt, '\n')
```

```
## Alternative confidence interval:  38.44444 41.55556
```

Note: the bootstrap percentile method should not be used. The idea is not to calculate the difference δ^* but to use the distribution of the boostraped statistic as a direct approximation of the distribution of the statistic on the data.

1.3.5 Studentized version

We can look at studentized versions of confidence intervals by considering:

$$\sqrt{n}\frac{(\bar{X}-\theta)}{\sigma(X)}$$

Whose fluctuations are approximated by the fluctuations of

$$\sqrt{n}\frac{(\bar{X}^*-\theta)}{\sigma(F^*)}$$

1.3.6 Hypothesis tests

It is also possible to perform hypothesis tests by comparing the bootstrapped test statistic under H_0 to the observed value of the test statistic on the sample.

1.4 Parametric boostrap case

The approach requires to have a parametric distribution of the data. Once the parameter is estimated on the initial sample, it is possible to generate bootstrap data sets from this parametric distribution and then to recalculate the statistic of interest on these bootstrap data. This allows to obtain confidence intervals or to do hypothesis tests.

1.5 Use of the bootstrap in prediction models

1.5.1 Case sampling or error sampling

Let's imagine that we want to fit a model of the type

$$Y = g(X) + \epsilon$$

We can either :

- resample the individuals (**case sampling**) and refit the predictive model on the new dataset.
- resample the ϵ errors (**error sampling**), deduce new synthetic values of Y, and readjust the model on these new data.

1.5.2 Out-of-bag sampling

Like cross-validation, the bootstrap can offer an attractive solution to evaluate the performance of predictive models. Indeed, we show that about 1/3 of the data is not present in the bootstrap sample (out of bag):

$$\lim_{n\to+\infty}\left(1-\frac{1}{n}\right)^n = e^{-1} \simeq \frac{1}{3}.$$

Thus, during the training of a predictive model, these data will allow us to estimate the performance of the trained model from the bootstrap sample in the case of case sampling.

This is known as the **Out of Bag Error** or OOB error. We then average these different out-of-bag errors to obtain the average "out-of-bag" error.

In the random forest method, which makes particular use of the boostrap (each tree of the forest is built from a bootstrap sample) this OOB error is particularly used and can be used to determine a number of tuning parameters.

2 More formal approach to presenting the bootstrap

2.1 Objectives of inferential statistics

- $\mathcal{X}_n = (X_1, \ldots, X_n)$ sample i.i.d. from F ($F(x) = \mathbb{P}(X_i \leq x)$)
- $\theta(F)$ quantity of interest which depends on F.
- $T(\mathcal{X}_n)$ a statistic, estimator of $\theta(F)$

We want:

- the bias of $T(\mathcal{X}_n)$: $\mathbb{E}_F(T(\mathcal{X}_n)) - \theta(F)$
- the variance of $T(\mathcal{X}_n)$: $\mathbb{E}_F(T^2(\mathcal{X}_n)) - \mathbb{E}_F^2(T(\mathcal{X}_n))$
- the MSE of $T(\mathcal{X}_n)$: $\mathbb{E}_F((T(\mathcal{X}_n) - \theta(F))^2)$
- the law of $T(\mathcal{X}_n)$: $G^n(x) = \mathbb{P}_F(T(\mathcal{X}_n) \leq x)$

Problem : All these quantities depend on the unknown law F !

2.2 Empirical repartition function

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} 1_{X_i \leq x}, \forall x$$

Plug-in estimation : $T(\mathcal{X}_n) = \theta(F_n) = \hat{\theta}$

Examples: replace expectation, variance, median, ... by their counterpart from the sample.

$$\mathbb{E}_F(X) = \int x dF(x) \qquad \text{replaced by} \qquad \mathbb{E}_{F_n}(X) = \int x dF_n(x) = \frac{1}{n}\sum_{i=1}^{n} X_i = \bar{X}$$

2.3 The bootstrap world

We define bootstrap samples as follows:

$$\mathcal{X}_1^* = (X_{1,1}^* = X_{m_1}, \ldots, X_{1,n}^* = X_{m_n})$$

$$\ldots$$

$$\mathcal{X}_b^* = (X_{b,1}^* = X_{m_{(b-1)n+1}}, \ldots, X_{b,n}^* = X_{m_{bn}})$$

$$\ldots$$

where the m_k are drawn randomly with discount in $1, \ldots, n$

The law of $X^*_{b,j}$ is conditional on $\mathcal{X}_n$.

Conditionally to $\mathcal{X}_n$, $X^*_{b,j}$ is a r.v. of distribution function F_n, the distribution function of $X_1, \ldots, X_n$. Thus the distribution of $X^*_{b,j}$ knowing $\mathcal{X}_n$ is **fully known**

2.4 Classic bootstrap estimator

real world

- $\hat{\theta} = \theta(F_n) = T(\mathcal{X}_n)$ estimator from the initial sample
- G^n the distribution function of $\hat{\theta}$, which depends on F (and on n) is unknown

Bootstrap world

- $\hat{\theta}^*_b = T(\mathcal{X}^*_b)$ for each sample $\mathcal{X}^*_b$
- Conditionally to F_n, the distribution $G^{n,*}$ is known
- $G^{n,*}$ is estimated by

$$\hat{G}^*_{n,B}(t) = \frac{1}{B}\sum_{b=1}^{B} 1_{\hat{\theta}^*_b \leq t}$$

2.5 Examples

2.5.1 Estimation of the law of $\hat{\theta}$

The f.d.r. G^n of $\hat{\theta} = T(\mathcal{X}_n)$ is defined for $t \in \mathbb{R}$ by

$$G^n(t) = \int 1_{x \leq t} dG^n(x) = \mathbb{P}(\hat{\theta} \leq t)$$

Estimated by

$$G^{n,*}(t) = \int 1_{x \leq t} dG^{n,*}(x) = \mathbb{P}(\hat{\theta}^*_b \leq t)$$

then

$$\hat{G}^*_{n,B} = \int 1_{x \leq t} d\hat{G}^*_{n,B}(x) = \frac{1}{B}\sum_{b=1}^{B} 1_{\hat{\theta}^*_b \leq t}$$

2.5.2 Estimation of the bias of $\hat{\theta}$

$$\mathbb{E}_F(T(\mathcal{X}_n)) - \theta(F) = \int x dG^n(x) - \theta(F)$$

estimated by

$$\int x dG^{n,*}(x) - \theta(F_n)$$

then approximated by

$$\int x d\hat{G}^*_{n,B}(x) - \theta(F_n) = \frac{1}{B}\sum_{b=1}^{B} \hat{\theta}^*_b - \theta(F_n)$$

We can also estimate the variance of the estimator by :

$$\frac{1}{B}\sum_{b=1}^{B}\left(\hat{\theta}^*_b - \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^*_b\right)^2$$

We have two successive approximations

$$G^n \to G^{n,*} \to \hat{G}^*_{n,B}$$

2.5.3 Basic bootstrapped CI

By ordering the different bootstrapped values obtained

$$\hat{G}^*_{(1)}, \hat{G}^*_{(2)}, \ldots, \hat{G}_{(B)}$$

We show that

$$\widehat{IC}^*_{\text{basic}}(1-\alpha) = [2\hat{\theta} - \hat{\theta}^*_{(\lceil B(1-\alpha/2)\rceil)}; 2\hat{\theta} - \hat{\theta}^*_{(\lceil B(1-\alpha/2)\rceil)}]$$

with $\lceil\rceil$ rounded up to the nearest integer.

For justification see informal presentation section.

2.5.4 Bootstrapped percentile CI

We can also simply use the empirical distribution of $\hat{\theta}^*_b$ to establish the confidence interval, which gives :

$$\widehat{IC}^*_{\text{empirical}}(1-\alpha) = [\hat{\theta}^*_{(\lceil B\alpha/2\rceil)}; \hat{\theta}^*_{(\lceil B(1-\alpha/2)\rceil)}]$$

This interval is not recommended because it assumes that the distribution $\hat{\theta}^*_b$ is a good approximation of the distribution of $\hat{\theta}$ where we rather have $\hat{\theta}^*_b - \hat{\theta}$ which is a good approximation of the distribution of $\hat{\theta} - \theta(F)$ (property which is used in the construction of the basic bootstrap interval).

2.5.5 IC t-bootstrap

When we are able to have an approximate version of the variance of the estimator it is recommended to construct a t-bootstrap confidence interval as follows:

$$S = \sqrt{n}\frac{\hat{\theta} - \theta}{\sigma(\mathcal{X}_n)}$$

where $\frac{\sigma^2(\mathcal{X}_n)}{n}$ is the variance of the estimator calculated from the initial sample.

The corresponding bootstrapped version is

$$S^*_b = \sqrt{n}\frac{\hat{\theta}^*_b - \hat{\theta}}{\sigma(\mathcal{X}^*_b)}$$

and we deduce the following boostrap interval:

$$\widehat{IC}^*_{\text{t-boot}}(1-\alpha) = \left[\hat{\theta} - \frac{\sigma(\mathcal{X}_n)}{\sqrt{n}} S_{(\lceil B(1-\alpha/2)\rceil)}; \hat{\theta} - \frac{\sigma(\mathcal{X}_n)}{\sqrt{n}} S_{(\lceil B\alpha/2\rceil)}\right]$$

We will see in the tutorial how to use the `boot` package to easily get all these intervals. Other ranges can be used but they will not be detailed here.

2.6 Test via the t-bootstrap

$$H_0 : \theta = \theta_0 \qquad \text{vs} \qquad H_1 : \theta \neq \theta_0$$

We calculate

$$\bar{S} = \left|\sqrt{n}\frac{\hat{\theta} - \theta_0}{\sigma(\mathcal{X}_n)}\right|$$

and

$$\bar{S}^*_b = \left|\sqrt{n}\frac{\hat{\theta}^*_b - \hat{\theta}}{\sigma(\mathcal{X}^*_b)}\right|$$

The critical probability is approximated by

$$\hat{p}_B = \frac{\#\{b : \bar{S}_b^* > \bar{S} + 1\}}{B + 1}$$

If H_0 is false we should have quite rarely the gap $|\hat{\theta}_b^* - \hat{\theta}|$ greater than the gap $|\hat{\theta} - \theta(F)|$. That is to say $\bar{S}_b^* > \bar{S}$ with low probability (low critical probability).

2.7 Regression problem

$$\mathcal{S} = ((Y_1, X_1), \ldots, (Y_n, X_n))$$

with

- $Y_i(\Omega) \subset \mathbb{R}$
- $X_i(\Omega) \subset \mathbb{R}^p$

We want to estimate $E(Y_i|X_i) = g(X_i)$.

Linear regression

$$Y_i = X_i + \epsilon_i$$

$\epsilon \sim \mathcal{L}_\epsilon(0, \sigma^2)$ from which $Y_i|X_i \sim \mathcal{L}_\epsilon(X_i\beta, \sigma^2)$.

$g(x) = x\hat{\beta}$ with $\hat{\beta}$ the least squares estimator of β.

Logistic regression

$Y_i(\Omega) = \{0, 1\}$ and

$$\mathbb{E}(Y_i|X_i) = P(Y_i = 1|X_i) = \frac{\exp(X_i\beta)}{1 + \exp(X_i\beta)} = \pi(X_i\beta)$$

then $Y_i|X_i \sim \mathcal{L}(\pi(X_i\beta), \pi(X_i\beta)(1 - \pi(X_i\beta)))$. $\hat{g}(x) = \pi(x\hat{\beta})$ with $\hat{\beta}$ the estimator of the maximum of likelihood of β.

2.7.1 Resampling of the individuals "case sampling"

As the individuals (Y_i, X_i) are supposed to be i.i.d :

1. we randomly sample with discount in the sample to obtain

$$\mathcal{S}_1^* = \big((Y_{1,1}^*, X_{1,1}^*), \ldots, (Y_{1,n}^*, X_{1,n}^*)\big)$$

$$\ldots$$

$$\mathcal{S}_B^* = \big((Y_{B,1}^*, X_{B,1}^*), \ldots, (Y_{B,n}^*, X_{B,n}^*)\big)$$

2. we compute for each bootstrapped sample $\hat{g}_{\mathcal{S}_b^*}\$$

2.7.2 Re-sampling of errors "errors sampling

- Residuals $e_i = Y_i - X_i\hat{\beta}$
- $E(e_i) = 0$ and $V(e_i) = (1 - H_{ii})\sigma^2$ with $H = X(X^\top X)^{-1}X^\top$
- Studentized residuals

$$e_i^S = \frac{e_i}{\sqrt{1 - H_{ii}}}\hat{\sigma}_{-i}$$

these residuals are supposed to be close to the law of e_i/σ^2, and they are the ones that will be resampled afterwards.

The algorithm is as follows:

1. We compute the estimators $\hat{\beta}$ and $\hat{\sigma}^2$ from $\mathcal{S}$, then the studentized residuals $e_1^S, \ldots, e_n^S$.
2. Randomly sample with discount in the sample $(e_1^S, \ldots, e_n^S)$ to obtain

$$(e_{1,1}^{S,*}, \ldots, e_{1,n}^{S,*})$$

$$\ldots$$

$$(e_{B,1}^{S,*}, \ldots, e_{B,n}^{S,*})$$

3. We reconstruct for each b and each i a synthetic $Y_{i,b}^*$:

$$Y_{i,b}^* = X_i + \hat{\sigma} e_{b,i}^{S,*}$$

4. We compute for each bootstrapped sample the estimators of β and σ.

Adaptations are necessary in the case of logistic regression. They are not detailed here but will be the subject of an exercise.

2.7.3 Boostrap test applied to the case of error sampling

The main philosophy of this approach is to resample the errors under H_0 and to derive an approximate distribution of the test statistic under H_0. The test statistic considered will be :

- the **Fisher statistic** in the linear regression model
- the **maximum likelihood ratio statistic** in the logistic model

An example of this will be given in the exercise.

3 Openings to other approaches

Here we have focused on the bootstrap but many resampling approaches can be useful in practice. We can mention :

- subsampling (sub-sampling without replacement) and in particular the Jackknife used historically to determine the variance of certain estimators and reduce their bias
- leave-one-out and k-fold cross-validation approaches
- permutation testing approaches

4 Sources

- http://www.math-evry.cnrs.fr/_media/members/aguilloux/enseignements/bootstrap/slides.pdf
- https://math.mit.edu/~dav/05.dir/class24-prep-a.pdf