P5 - Bootstrap

Demagun gure populazioa adierazten duela X zorizko aldagaiak. Gure helburua banaketa honen parametro ezberdinen estimazioa egitea izango da eta estimatzaile horien banaketak eta propietateak ezagutzen saiatzea teknika ezberdinak erabiliz.

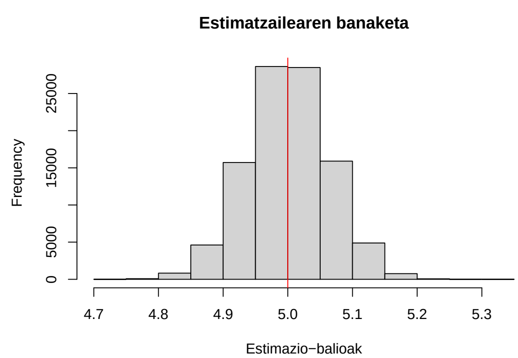
Estimatzailearen banaketa

Klasean azaldu moduan, estimatzaileak zorizko aldagaiak dira, lagin ezberdinen arabera balio ezberdinak (estimazioak) hartzen dituztenak. Hau hala izanik, estimatzailearen banaketa ezagutzeak bere propietateei buruzko informazioa emango digu. Estimatzailearen banaketa lortzeko bide ezberdin batzuk azter ditzagun:

Banaketa teorikoa

Demagun Xen banaketa ezaguna dela, adibidez, $X \sim \mathcal{N}(5, 2^2)$. $EX = \mu$ parametroaren estimatzaile gisa $\hat{\mu} = \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ hartzen badugu, $\hat{\mu} = \overline{X} \sim \mathcal{N}(5, \frac{2}{\sqrt{n}})$ dela dakigu limite zentralaren teoremagatik. $n = 1000$ laginaren tamaina izanik, irudikatu dezagun estimatzailearen banaketa, laginduz eta histograma bat eginez:

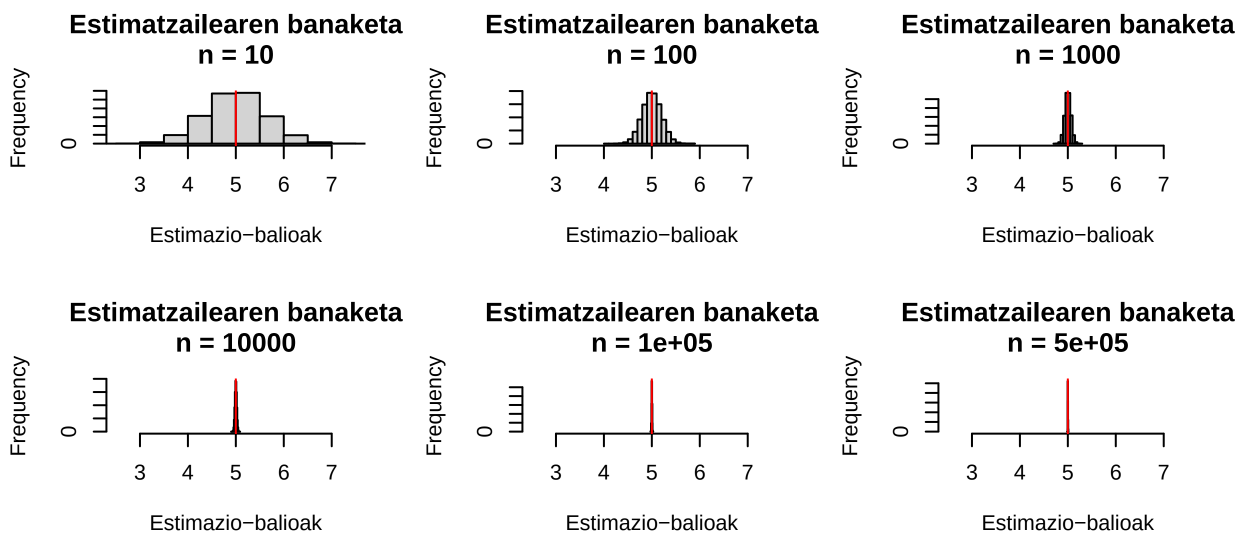
```
n <- 1000
hist(rnorm(100000, 5, 2/sqrt(n)), main="Estimatzailearen banaketa", xlab="Estimazio-balioak")
abline(v=5, col="red")
```



Ariketa: Demagun $X \sim \mathcal{N}(5, 2^2)$ banaketa dugula, irudikatu $\hat{\mu} = \overline{X}$ estimatzailearen banaketa n balio ezberdinetarako (batzuk handiak eta besteak txikiak). Zer esan dezakezu?

```
par(mfrow=c(3, 3))
for (n in c(10, 100, 1000, 10000, 100000, 500000)){
  hist(rnorm(100000, 5, 2/sqrt(n)), main=paste("Estimatzailearen banaketa\nn =", n),
      xlab="Estimazio-balioak", xlim=c(2.5, 7.5))
```

```
  abline(v=5, col="red")
}
```



Soluzioa: lagin tamaina handitu ahala bariantza txikitzen da.

OHARRA: Estimatzailearen banaketa teorikoa oso kasu gutxietan ezagutuko dugu, izan ere, X aldagaiaren banaketa ezaguna izatea ez da ohikoa, eta gainera, ezaguna izanda ere, estimatzaile guztientzat ez da tribiala banaketa analitikoki kalkulatzea.

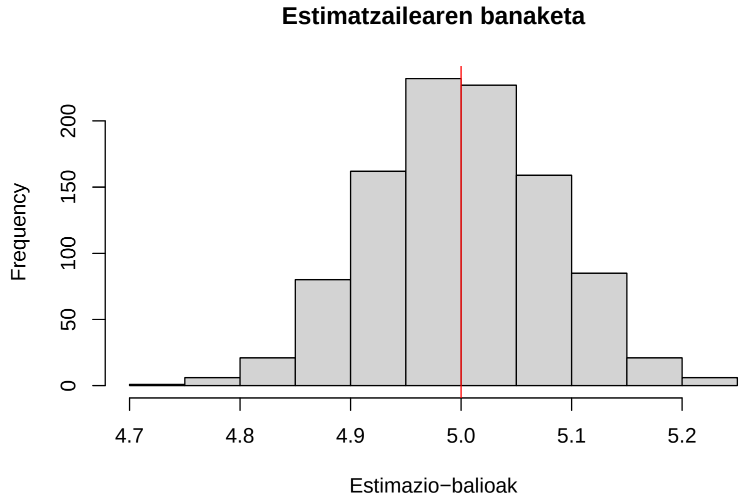
Lagin banaketa

Demagun Xren banaketa ezaguna dela, adibidez, $X \sim \mathcal{N}(5, 2^2)$ eta demagun banaketaren mediana (θ) estimatzeko lagin mediana erabiliko dugula: $\hat{\theta} = Median\{X_1, X_2, ..., X_n\}$. Aurreko atalean ez bezala, kasu honetan ez da erraza estimatzailearen benetako probabilitate banaketa zein den jakitea. Egoera honetan, X-ren banaketatik lagin ezberdin asko ateraz, $\hat{\theta}$ estimatzailearen banaketa hurbildu dezakegu. Lehenik, banaketaren laginak sortuko ditugu.

```
banaketa.laginak <- function(lagin.kopurua, lagin.tamaina, mu, sigma){
  laginak <- matrix(rnorm(lagin.kopurua*lagin.tamaina, mu, sigma),
                    nrow=lagin.kopurua, ncol=lagin.tamaina)
  return(laginak)
}

laginak <- banaketa.laginak(1000, 1000, 5, 2)
```

Ondoren, lagin bakoitzeko estimazioa lortuko dugu (mediana aplikatuz) eta emaitza (estimatzailearen lagina) histograma baten bidez irudikatuko dugu.

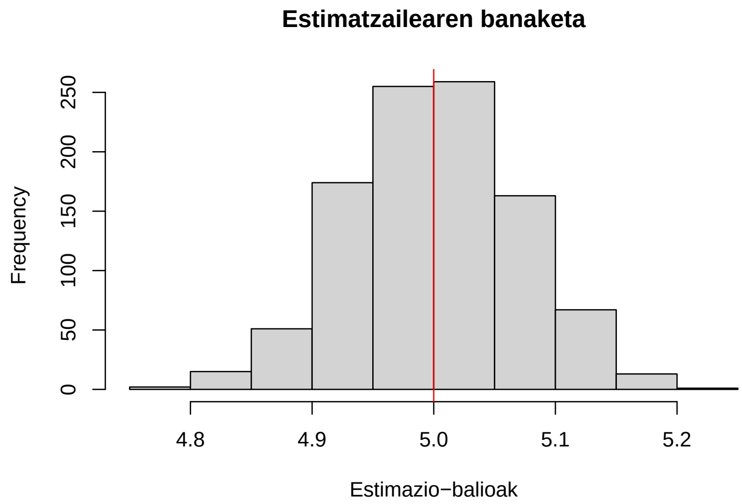
```
estimazioak <- apply(laginak, 1, median)
hist(estimazioak, main="Estimatzailearen banaketa", xlab="Estimazio-balioak")
abline(v=5, col="red")
```



OHARRA: Egoera errealetan populazioaren banaketa ez da ezaguna izaten, beraz, banaketa hau ez dugu eskura izaten eta ezin dugu lagindu.

Ariketa: Atal honetako populazio berdina hartuz, irudikatu $\hat{\theta}$ moztutako batezbestekoaren lagin-banaketa. Moztutako batezbestekoa lagineko datuen $\%50$ zentralen batezbestekoa da eta estatistiko hau egoera batzuetan mediana eta batezbestekoa baino egokiagoa da banaketaren "zentroa" adierazteko.
OHARRA: Moztutako batezbestekoa inplementatzeko, aztertu **mean** funtzioaren **trim** argumentua.

```
estimazioak <- apply(laginak, 1, mean, trim=0.25)
hist(estimazioak, main="Estimatzailearen banaketa", xlab="Estimazio-balioak")
abline(v=5, col="red")
```



Bootstrap banaketa

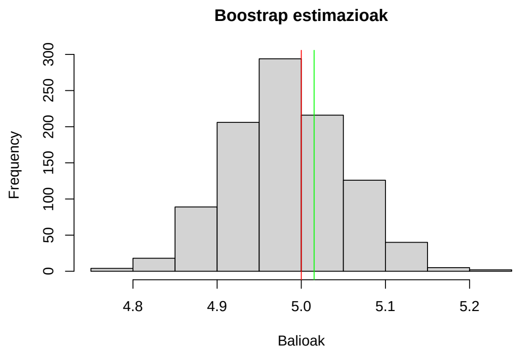
Suposatu dezagun Xren banaketa **ez** dugula ezagutzen eta dugun informazio bakarra banaketa honetatik ateratako $n = 1000$ tamainako lagin bat dela (`lagina` objetuan gordeta)[1]. Mediana estimatzailearen ($\hat{\theta}$) banaketa, hurbildu nahi dugu baina kasu honetan populazioaren banaketa ez dugu ezagutzen, beraz ezin ditugu bertatik laginak atera, soilik lagin bat daukagu. Beraz, estimatzailearen banaketa lortzeko bootstrap ez parametrikoa erabiliko dugu. Gure laginetik abiatuz, itzuleradun laginketa erabiliz $B = 1000$ bootstrap lagin sortuko ditugu, baita ere $n = 1000$ tamainakoak. Kontuan izan n eta Bk ez dutela zertan beti zenbaki berdinak izan.

```
bootstrapLaginakLortu <- function(hasierako.lagina, boot.lagin.kopurua, boot.lagin.tamaina){
  boot.laginak <- matrix(
    sample(hasierako.lagina, boot.lagin.kopurua*boot.lagin.tamaina, replace=TRUE),
    nrow=boot.lagin.kopurua, ncol=boot.lagin.tamaina)
  return(boot.laginak)
}

boot.laginak <- bootstrapLaginakLortu(lagina, 1000, 1000)
```

Behin aldagaiaren (bootstrap) laginketa dugula, lagin bakoitzari estimatzailea aplikatu eta emaitza histograma baten bidez irudikatuko dugu. Horrez gain bi marra bertikal marraztuko ditugu, bat benetako parametroaren balioa (gorriz) adierazteko eta bestea `lagina` laginaren bidez lortutako estimazioa (berdez) adierazteko.

```
# boot.estimazioak <- apply(boot.laginak, 1, median)
boot.estimazioak <- apply(boot.laginak, 1, mean, trim=0.25)
hist(boot.estimazioak, main="Boostrap estimazioak", xlab="Balioak")
estimazioa <- median(lagina)
abline(v=5, col="red") # Hau egoera errealetan ez dugu ezagutuko!
abline(v=estimazioa, col="green")
```

[1] Tutorialeko ariketak egiteko, lagina artifizialki sortuko dugu `lagina <- rnorm(1000, 5, 2)` eginez



OHARRA: Azken egoera hau da errealistena, populazioaren eta estimatzailearen banaketak ez dira ezagunak. Hala ere, bootstrap-banaketa erabili dezakegu lagin-banaketa hurbiltzeko eta gure estimatzailearen hainbat propietate ezagutzeko edo konfiantza-tarteak eraikitzeko. Ikusten denez, bootstrap metodoak ez du balio estimazioa hobetuko, izan ere, lagin-banaketa benetako parametroan dago zentratuta eta, aldiz, bootstrap-banaketa laginaren estimazioan.

Ariketa: Errepikatu aurreko kode blokeetan egindakoa baina kasu honetan $\hat{\theta}$ moztutako batezbestekoaren bootstrap banaketa irudikatuz. Gainera, egin proba ezberdinak hasierako laginaren tamaina eta bootstrap laginen kopurua aldatuz.

Alborapena eta errore estandarra

Estimatzailearen banaketatik informazio ezberdina atera dezakegu. Tinkotasunari buruzko informazioa lortzeko adibidez alborapena eta SE hurbildu ditzakegu.

Ariketa: Bete itzazu hurrengo bi funtzioetan falta diren kode-lerroak.

Implementatuko dugun lehenengo funtzioak estimatzailearen alborapena hurbiltzen du. Alborapena $b(\hat{\theta}) = E(\hat{\theta}) - \theta$ da. Hau hurbiltzeko bootstrap metodoa eta plug-in printzipioa aplikatuko ditugu, itxaropena bootstrap estimazioen batezbestekoarengatik ordezkatuz ($E(\hat{\theta}) = \bar{\hat{\theta}}^*$) eta benetako parametroaren balioa estimazioa erabiliz ordezkatuz ($\theta \sim \hat{\theta}$).

$$b(\hat{\theta}) \simeq b^*(\hat{\theta}) = \bar{\hat{\theta}}^* - \bar{\theta}_{\mathbf{x}}$$

```r
estimatuAlborapena <- function(boot.estimazioak, estimazioa){
  #Kalkulatu bootstrap estimazioak erabiliz alborapena (44. gardenkia)
  mean(boot.estimazioak) - estimazioa
}
```

Orain hurbildu dezagun estimatzailearen errore estandarra:

$$SE^*(\hat{\theta}) = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}(\hat{\theta}_i^* - \bar{\hat{\theta}}^*)^2}$$

```r
estimatuSE <- function(boot.estimazioak){
  #Kalkulatu bootstrap estimazioak erabiliz SE (45. gardenkia)
  sd(boot.estimazioak)
}
```

Ariketa: Aurreko ataleko 'lagina' objektua erabiliz eta bootstrap laginketa ezberdinak erabiliz (lagin kopurua eta lagin tamainak aldatuz), hurbildu itzazu $\hat{\mu} = \overline{X}$ eta $\hat{\theta} = Median\{X_1, X_2, ..., X_n\}$ estimatzaileen alborapena eta errore estandarra. Zer esan dezakezu?

```r
for (n in c(10, 100, 1000, 10000)){
  print(paste("n", n))
  for (B in c(10, 100, 500, 1000, 10000)){
    estimazioa <- mean(lagina, trim=0.25)
    boot.laginak <- bootstrapLaginakLortu(lagina, B, n)
    boot.estimazioak <- apply(boot.laginak, 1, mean, trim=0.25)

    alborapena <- estimatuAlborapena(boot.estimazioak, estimazioa)
    errorea <- estimatuSE(boot.estimazioak)

    print(paste("    B", B, "Alborapena:", alborapena, "SD:", errorea))

    #alborapenak <- c(alborapenak, estimatuAlborapena(boot.estimazioak, estimazioa))
    #erroreak <- c(erroreak, estimatuSE(boot.estimazioak))

  }
}
```

```
## [1] "n 10"
## [1] "    B 10 Alborapena: -0.13127736823246 SD: 1.08390397055229"
## [1] "    B 100 Alborapena: 0.0468325647014005 SD: 0.69593556137922"
## [1] "    B 500 Alborapena: 0.00642157313049019 SD: 0.631539530973021"
## [1] "    B 1000 Alborapena: 0.0181645866883757 SD: 0.663689422941592"
## [1] "    B 10000 Alborapena: -0.00479046185552612 SD: 0.671336364151609"
## [1] "n 100"
## [1] "    B 10 Alborapena: -0.0331234100410036 SD: 0.226580263091263"
## [1] "    B 100 Alborapena: 0.0278830518453308 SD: 0.221775366280681"
## [1] "    B 500 Alborapena: -0.0119047632577649 SD: 0.209295066063332"
## [1] "    B 1000 Alborapena: 0.000907884682579407 SD: 0.218245748464768"
## [1] "    B 10000 Alborapena: 0.0016937008148048 SD: 0.220839913559175"
## [1] "n 1000"
## [1] "    B 10 Alborapena: 0.000251290831617901 SD: 0.0739964152083911"
## [1] "    B 100 Alborapena: 0.00867272469734193 SD: 0.0792518985186003"
## [1] "    B 500 Alborapena: -0.0032598733727518 SD: 0.0713700932348378"
## [1] "    B 1000 Alborapena: -4.51587183833979e-05 SD: 0.0692942916887815"
## [1] "    B 10000 Alborapena: 0.000334394471956934 SD: 0.070809614437748"
## [1] "n 10000"
## [1] "    B 10 Alborapena: -0.0038695889509599 SD: 0.0280050695910791"
## [1] "    B 100 Alborapena: 0.00271764023082355 SD: 0.0215463185609258"
## [1] "    B 500 Alborapena: -0.00112111244936219 SD: 0.0222832128602949"
## [1] "    B 1000 Alborapena: 0.000624194533909161 SD: 0.0220180031496807"
```

```
## [1] "    B 10000 Alborapena: -0.000385564230665381 SD: 0.0221149847033518"
```

```
#alborapenak <- matrix(alborapenak, nrow = 4, ncol = 5)
#erroreak <- matrix(erroreak, nrow=4, ncol=5)

# ALBORAPENAK
# alborapenak

# ERRORE ESTANDARRAK
# erroreak
```

Interpretazioa: Lagin tamaina (n) handitzean zehaztasuna ere handitzen da, nahiz eta hasierako laginaren tamaina gainditu. Bootstrap tamaina (B) handitzean ere berdina gertatzen da, baina tamaina batetik aurrera ez da nabaria.

Konfiantza-tarteak

Konfiantza tarte teorikoa

Konfiantza tarte teorikoak estimatzailearen menpeko estatistiko jakin baten banaketa ezaguna denean erabili ditzakegu. Adibidez, hasierako populazioa normala (edo nahikoa handia) denean eta batezbestekoaren estimatzaile moduan $\hat{\mu} = \overline{X}$ hartuta, errore estandarrean oinarritutako konfiantza tarteak eraiki ditzakegu μ parametroarentzat era honetan:

$$\frac{\overline{X} - \mu}{\frac{S_{n-1}}{\sqrt{n}}} : t_{n-1} \Rightarrow KT = (\overline{x} - t_{\alpha/2;n-1}\frac{S_{n-1}}{\sqrt{n}}, \overline{x} + t_{\alpha/2;n-1}\frac{S_{n-1}}{\sqrt{n}})$$

Ariketa: Goiko formulan oinarrituz, bete kode-hutsuneak hurrengo funtzioan batezbestekorako ohiko t-konfiantza tarteak kalkulatzeko.

```
getCI <- function (x, alpha = 0.05)
{
  m <- mean(x)  # mean
  se <- sd(x) / sqrt(length(x))  # Standard error
  df <- length(x) - 1  # Degrees of freedom
  t <- qt(1 - alpha/2, df)  # t-critical value

  d <- t * se
  ci <- c(m - d, m + d)
  names(ci) <- paste("%", c((alpha/2)*100, (1-alpha/2)*100), sep = "")
  return(ci)
}
```

```
boot.laginak <- bootstrapLaginakLortu(lagina, 1000, 1000)
boot.estimazioak <- apply(boot.laginak, 1, mean, trim=0.25)
```

```
getCI(lagina)
```

```
##     %2.5    %97.5
## 4.838818 5.083120
```

Ariketa: Erabili aurreko funtzioa 'lagina' objetutik batezbestekorako $\%95$ko konfiantza-tartea kalkulatzeko. Kontuan izan $0.95 = 1 - \alpha$ dela.

Honelako konfiantza tarteak `t.test` funtzioarekin ere kalkulatu ditzakegu honela:

```
t.test(lagina, conf.level=0.95)$conf.int
```

```
## [1] 4.838818 5.083120
## attr(,"conf.level")
## [1] 0.95
```

OHARRA: Ezin badugu lortu estatistikoaren benetako banaketa bootstrap konfiantza-tarteak erabili ditzakegu.

Pertzentiletan oinarritutako bootstrap konfiantza tarteak

Honelako konfiantza tarteak eraikitzeko, estimatzailearen bootstrap banaketaren pertzentilak erabiliko ditugu zuzenean. Adibidez, $\%95$ konfiantza mailarako, 2.5. eta 97.5. pertzentilak erabiliko ditugu:

$$KT = (\hat{\theta}_{0.025}, \hat{\theta}_{0.975})$$

Ariketa: Pertzentiletan oinarritutako bootstrap konfiantza tarteak $\hat{\mu}$ parametroarentzat honako kodea erabiliz eraiki ditzakegu. Bete kode-hutsuneak 'quantile' funtzioa erabiliz eta erabili 'lagina' objetutik batezbestekorako $\%95$ko konfiantza-tartea kalkulatzeko.

```
getCIBootQuantile <- function(x, B=1000, alpha=0.05, FUN=mean, ...)
{
    n <- length(x)
    boot.laginak  <- bootstrapLaginakLortu(x, B, n)
    boot.estimazioak <- apply(boot.laginak, 1, FUN, ...)
    ci <- quantile(boot.estimazioak, c((alpha/2), (1-alpha/2)))
    names(ci) <- paste("%", c((alpha/2)*100, (1-alpha/2)*100), sep="")
  return(ci)
}

getCIBootQuantile(lagina, alpha=0.05)
```

```
##     %2.5    %97.5
## 4.834878 5.087846
```

```
getCIBootQuantile(lagina, alpha=0.05, FUN=mean, trim=0.25)
```

```
##     %2.5    %97.5
## 4.849970 5.112529
```

```
getCIBootQuantile(lagina, alpha=0.05, FUN=median)
```

```
##     %2.5    %97.5
## 4.848201 5.137155
```

Ariketa: Orokortu aurreko funtzioa, $\hat{\mu}$ parametroarentzat erabili nahi den estimatzailea (adibidez, 'mean', 'median',...) funtzioari argumentu moduan zehaztuz. Erabili funtzio berria mediana erabiliz.

OHARRA Aurreko atalean kalkulatutako tarteak sinpleak dira, baina kasu askotan ez dituzte emaitza onak ematen. Hurrengo atalean tarte apur bat sofistikatuagoak nola eraiki ikusiko dugu.

Bootstrap t-tarteak

Azkenik, populazioaren banaketa ez bada normala edo ez badugu ezagutzen, ohiko konfiantza tartean agertzen den t-banaketa ez dugu izango, baina benetako banaketa hurbiltzen saiatu gaitezke bootstrap teknika erabiliz. Ondoren, banaketa hau erabiliz, konfiantza tartea honela eraiki dezakegu:

$$KT = \left(\hat{\theta} - t^*_{1-\alpha/2;n-1} \cdot SE(\hat{\theta}),\ \hat{\theta} - t^*_{\alpha/2;n-1} \cdot SE(\hat{\theta})\right)$$

Ariketa: $\hat{\mu}$ parametrorako bootstrap t-konfiantza tarteak honako kodea erabiliz eraiki ditzakegu. Kontuan izan, *batezbestekoa* estimatzailearen errore estandarrak (SE) formula itxia duela. Zein da? Zer egin dezakegu errore estandarra ezin badugu modu itxian kalkulatu? Bete kode-hutsuneak:

$$t = \frac{\hat{\theta} - \theta}{SE(\hat{\theta})} \xrightarrow{Bootstrap} t^* = \frac{\hat{\theta}^* - \hat{\theta}}{SE(\hat{\theta}^*)}$$

```
getCIBootTmean <- function(x, alpha=0.05, B=1000, closed=TRUE)
{
  estimazioa <- mean(x)
  n <- length(x)
  boot.laginak  <- bootstrapLaginakLortu(x, B, n)
  boot.estimazioak <- apply(boot.laginak, 1, mean)
  se <- sd(x)/length(x)
  boot.t <- approxDistributionTBootmean(estimazioa, boot.laginak, boot.estimazioak)
  ci_inf <- estimazioa - quantile(boot.t, 1-alpha/2)*se
  ci_sup <- estimazioa - quantile(boot.t, alpha/2)*se
  ci <- c(ci_inf, ci_sup)
  names(ci) <- paste("%", c((alpha/2)*100, (1-alpha/2)*100), sep="")
  return(ci)
}

approxDistributionTBootmean <-function(estimation, boot.samples, boot.estimations){
  n <- ncol(boot.samples)
  B <- nrow(boot.samples)
  boot.se <- apply(boot.samples, 1, sd)/sqrt(n)
  t.balioak <- (boot.estimations - estimation)/boot.se
}

getCIBootTmean(lagina)
```

```
##     %2.5    %97.5
## 4.957104 4.964586
```

Errore estandarrerako formula itxia ez dugunean bootstrap lagin bakoitzerako bootstrap lagin gehiago atera behar ditugu:

```
getCIBootT <- function(x, alpha=0.05, B=1000, B2=100, estim_fun=mean, ...)
{
  estimazioa <- estim_fun(x, ...)
  n <- length(x)
  boot.laginak  <- bootstrapLaginakLortu(x, B, n)
  boot.estimazioak <- apply(boot.laginak, 1, estim_fun, ...)
  se <- sd(boot.estimazioak)
  boot.t <- approxDistributionTBoot(estimazioa, boot.laginak, boot.estimazioak, B2, estim_fun, ...)
  ci_inf <- estimazioa - quantile(boot.t, 1-alpha/2)*se
  ci_sup <- estimazioa - quantile(boot.t, alpha/2)*se
  ci <- c(ci_inf, ci_sup)
  names(ci) <- paste("%", c((alpha/2)*100, (1-alpha/2)*100), sep="")
  return(ci)
}

approxDistributionTBoot <-function(estimation, boot.samples, boot.estimations, B2=100, estim_fun=mean,
  n <- ncol(boot.samples)
  B <- nrow(boot.samples)

  t.values <- numeric(B)

  for (i in 1:B){
    # Bootstrap lagin bakoitzeko azpi-bootstrap bat egin
    sample_i <- boot.samples[i,]

    # Bigarren mailako bootsrap sortu
    boot2.laginak <- bootstrapLaginakLortu(sample_i, B2, n)
    boot2.estimazioak <- apply(boot2.laginak, 1, estim_fun, ...)

    # Errore estandarraren urbilpena
    boot.se <- sd(boot2.estimazioak)

    # t.balioetara gehitu
    t.values[i] <- (boot.estimations[i] - estimation)/boot.se
  }

  return(t.values)
}

getCIBootT(lagina)
```

```
##     %2.5    %97.5
## 4.842079 5.089675
```

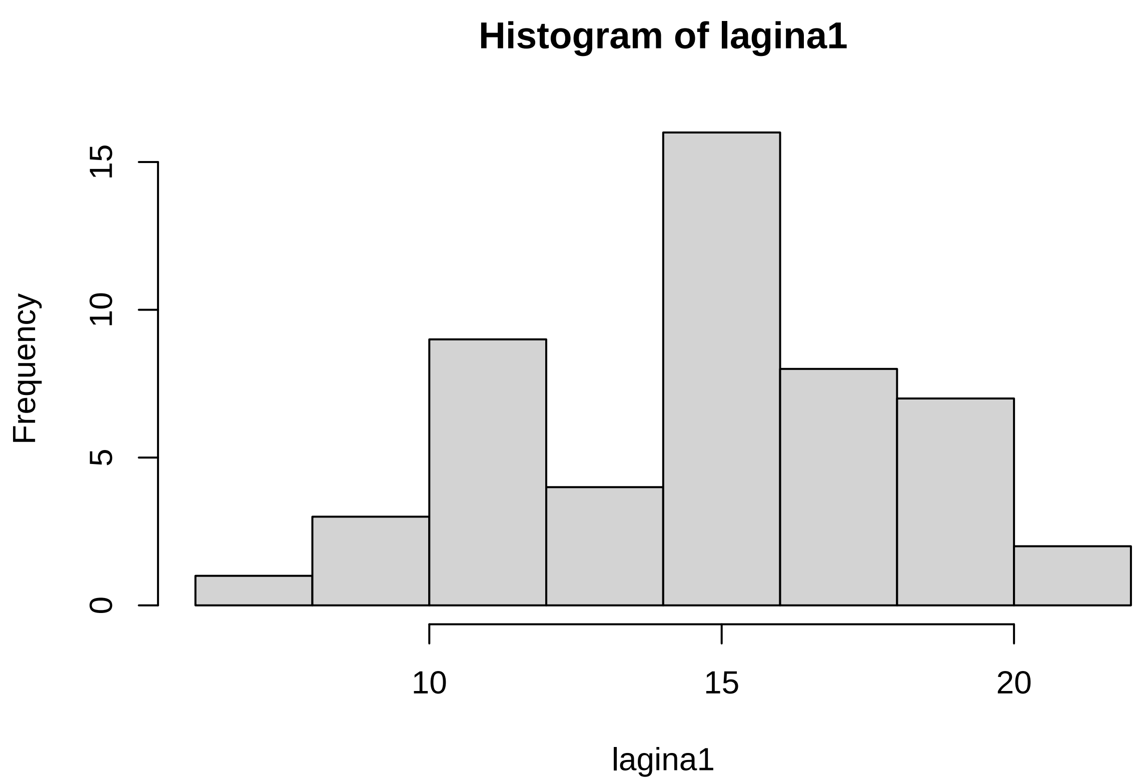
Ariketak

1. ariketa Demagun honako bi laginak ditugula:

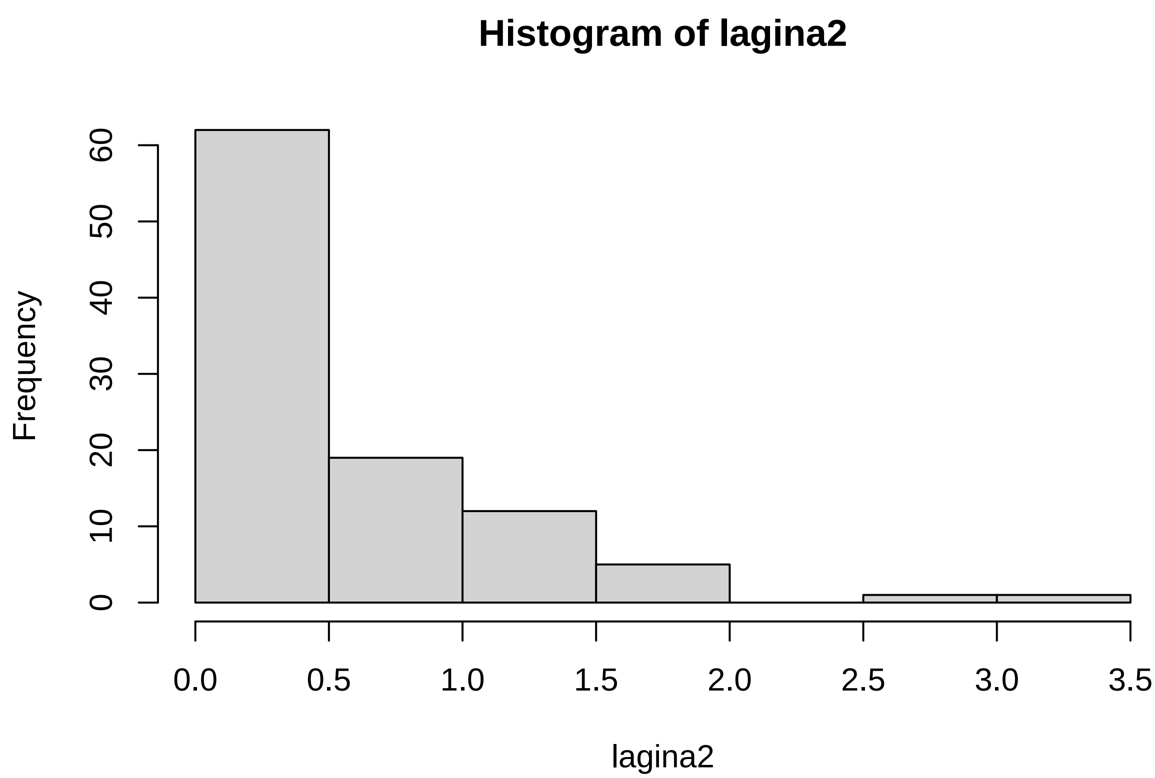
```
lagina1 <- c(11.8, 9.3, 11.4, 20.4, 15.2, 17.8, 15.2, 13.5, 15.1, 10.6, 18.4, 7.5, 17.7, 19.9,
             12.4, 14.6, 21.2, 10.1, 10.2, 13.7, 15.8, 18.1, 16.2, 19.8, 14.2, 14.1, 11.3,
             11.1, 16.8, 14.8, 15.8, 14.6, 15.0, 15.7, 10.3, 8.8, 17.0, 15.7, 18.6, 12.1, 16.4,
             19.3, 9.9, 15.9, 19.7, 17.6, 15.9, 16.2, 11.1, 15.6)


lagina2 <- c(0.10, 0.03, 0.03, 1.11, 0.50, 0.59, 0.22, 0.26, 3.19, 0.49, 1.14, 0.36, 0.38,
             1.10, 0.02, 0.18, 0.51, 0.41, 0.01, 0.51, 0.15, 0.36, 0.09, 1.50, 0.51, 0.57,
             0.66, 0.16, 1.10, 0.39, 0.01, 2.52, 1.94, 0.68, 1.06, 0.01, 0.09, 0.74, 0.07,
             1.39, 0.66, 0.92, 0.07, 0.13, 0.42, 0.02, 0.21, 0.47, 0.78, 1.71, 0.01, 0.44,
             0.31, 0.28, 0.26, 0.55, 1.44, 0.17, 0.20, 0.27, 0.07, 0.12, 0.33, 1.02, 1.54,
             0.43, 0.47, 1.99, 0.96, 0.52, 0.02, 0.65, 0.76, 0.03, 0.95, 0.36, 1.33, 0.69,
             0.15, 0.00, 0.46, 1.05, 1.23, 0.84, 0.49, 0.04, 0.04, 0.09, 0.42, 0.14, 0.20,
             0.01, 0.21, 1.88, 0.39, 0.16, 0.10, 0.07, 0.06, 0.17)
```

- Irudikatu laginen histogramak. Zer esan dezakegu banaketei buruz?

```
hist(lagina1)
```



```r
hist(lagina2)
```



Erantzuna: 1. lagina banaketa normalaren itxura dauka eta 2. lagina banaketa exponentzialarena

- Batezbestekoaren bootstrap banaketa irudikatu eta kalkulatu alborapena eta errore estandarra bi kasuetan.

```r
boot.laginak1 <- bootstrapLaginakLortu(lagina1, 1000, length(lagina1))
boot.estimazioak1 <- apply(boot.laginak1, 1, mean)
hist(boot.estimazioak1)
```


```r
se1 <- estimatuSE(boot.estimazioak1)
se1
```

```
## [1] 0.4498639
```

```r
alb1 <- estimatuAlborapena(boot.estimazioak1, mean(lagina1))
alb1
```

```
## [1] 0.01049
```

```r
boot.laginak2 <- bootstrapLaginakLortu(lagina2, 1000, length(lagina2))
boot.estimazioak2 <- apply(boot.laginak2, 1, mean)
hist(boot.estimazioak2)
```


```
se2 <- estimatuSE(boot.estimazioak2)
se2
```

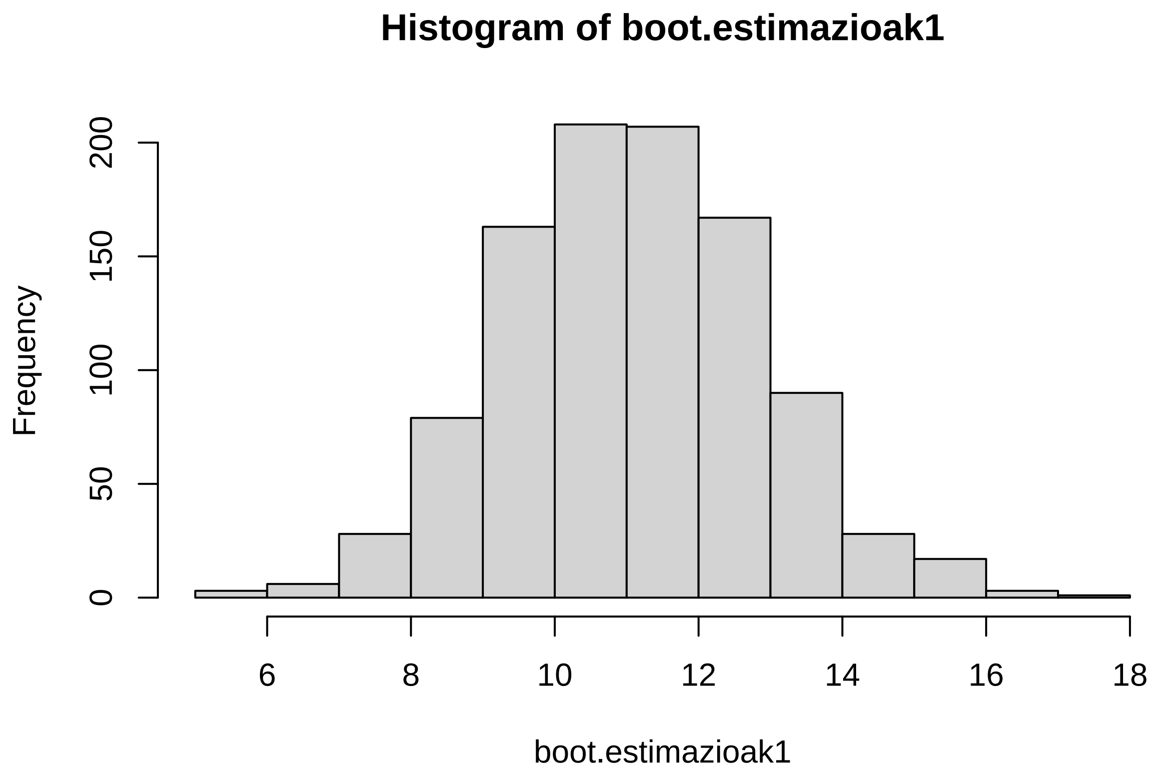
```
## [1] 0.05879059
```

```
alb2 <- estimatuAlborapena(boot.estimazioak2, mean(lagina2))
alb2
```

```
## [1] 0.0014974
```

- Bariantzaren bootstrap banaketa irudikatu eta kalkulatu alborapena eta errore estandarra bi kasuetan.

```
boot.laginak1 <- bootstrapLaginakLortu(lagina1, 1000, length(lagina1))
boot.estimazioak1 <- apply(boot.laginak1, 1, var)
hist(boot.estimazioak1)
```



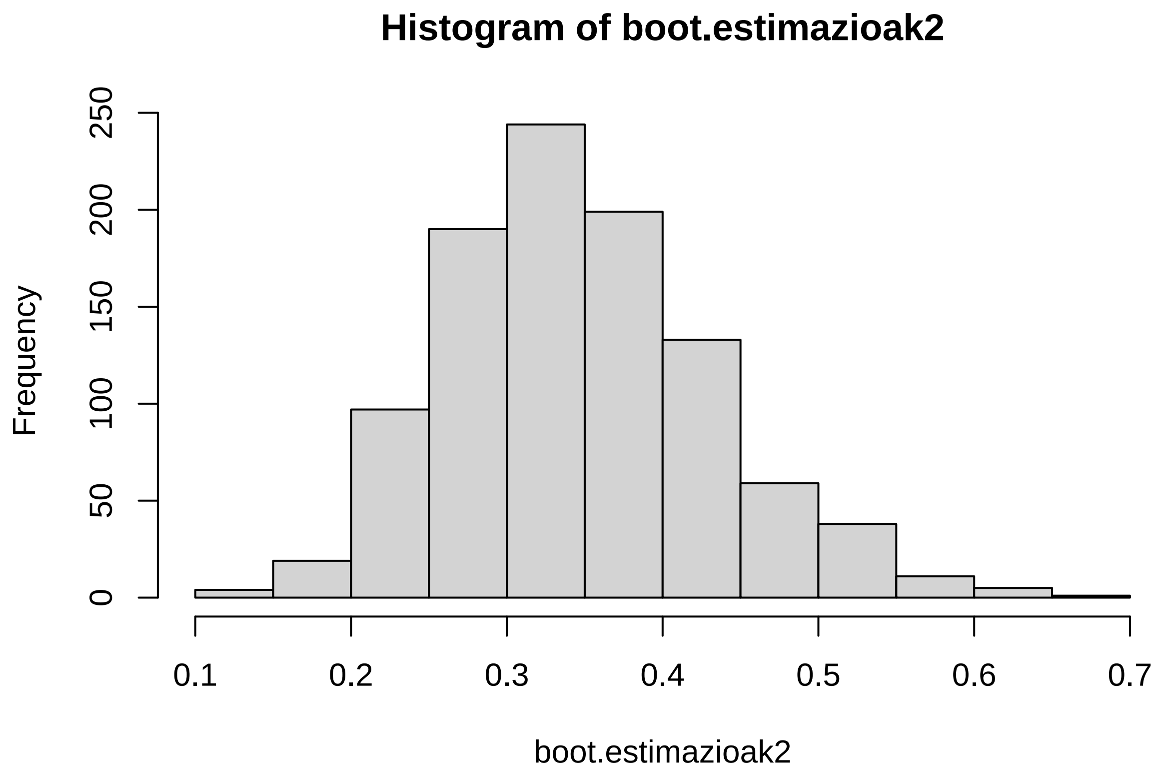

```
se1 <- estimatuSE(boot.estimazioak1)
se1
```

```
## [1] 1.790575
```

```
alb1 <- estimatuAlborapena(boot.estimazioak1, mean(lagina1))
alb1
```

```
## [1] -3.708843
```

```
boot.laginak2 <- bootstrapLaginakLortu(lagina2, 1000, length(lagina2))
boot.estimazioak2 <- apply(boot.laginak2, 1, var)
hist(boot.estimazioak2)
```




```
se2 <- estimatuSE(boot.estimazioak2)
se2
```

```
## [1] 0.08509209
```

```
alb2 <- estimatuAlborapena(boot.estimazioak2, mean(lagina2))
alb2
```

```
## [1] -0.2027345
```

- Kalkulatu batezbestekorako:
    - Ohiko konfiantza tarteak, asumituz hasierako populazioak banaketa normala jarraitzen duela.

$$\frac{\overline{X}-\mu}{\frac{S_{n-1}}{\sqrt{n}}} : t_{n-1} \Rightarrow KT = (\overline{x} - t_{\alpha/2;n-1}\frac{S_{n-1}}{\sqrt{n}}, \overline{x} + t_{\alpha/2;n-1}\frac{S_{n-1}}{\sqrt{n}})$$

```
getCI(lagina1)
```

```
##     %2.5    %97.5
## 13.83125 15.74475
```

```
getCI(lagina2)
```

```
##      %2.5      %97.5
## 0.4323685 0.6656315
```

- Pertzentiletan oinarritutako bootstrap konfiantza tarteak.

```r
getCIBootQuantile(lagina1, FUN = mean)
```

```
##    %2.5    %97.5
## 13.8720 15.6641
```

```r
getCIBootQuantile(lagina2, FUN = mean)
```

```
##      %2.5      %97.5
## 0.4378925 0.6666250
```

- Bootstrap t-konfiantza tarteak.

batazbestekoaren SE-rako formula itxia dago

```r
getCIBootTmean(lagina1)
```

```
##     %2.5     %97.5
## 14.64185 14.92048
```

```r
getCIBootTmean(lagina2)
```

```
##      %2.5      %97.5
## 0.5392687 0.5625353
```

- (ARIKETA EXTRA) Kalkulatu bariantzarako:
  - Pertzentiletan oinarritutako bootstrap konfiantza tarteak.

```r
getCIBootQuantile(lagina1, FUN=var)
```

```
##      %2.5      %97.5
##  7.642175 14.548261
```

```r
getCIBootQuantile(lagina2, FUN=var)
```

```
##      %2.5      %97.5
## 0.1932776 0.5207369
```

- Bootstrap t-konfiantza tarteak (OHARRA: Kasu honetan estimatzailearen benetako SE ez da ezaguna, ikus 53. gardenkia).

```r
getCIBootT(lagina1, estim_fun = var)
```

```
##      %2.5      %97.5
##  8.568885 15.933807
```

```r
getCIBootT(lagina2, estim_fun = var)
```

```
##      %2.5      %97.5
## 0.2129522 0.7431976
```

2. ariketa $\mathcal{N}(20, 5^2)$ populazioan oinarrituz, aipaturiko hiru motako μrako konfiantza-tarteek (KT) adierazitako konfiantza maila betetzen duten ala ez aztertu. Horretarako oinarritu konfiantza tarteen interpretazioan:

- Sor ditzagun zoriz $n = 100$ tamainako $N = 10000$ lagina gure populaziotik ($\mathcal{N}(20, 5^2)$).
- Lagin bakoitzerako kalkula dezagun μrako %95eko konfiantza-tartea eta begira dezagun ia parametroaren benetako balioa tartean dagoen.
- Konfiantza-tartea %95ekoa bada, parametroa tartean egon beharko luke 0.95 probabilitatearekin. Konprobatu ea hau betetzen den.

```r
n = 100
N = 1000
laginak <- matrix(rnorm(n*N, 20, 5), ncol = n, nrow = N)

testCI <- function(lagina, FUN1=getCI, ...){
    apply(laginak, 1, function(x){
    ci <- FUN1(x, ...)
    return (20>ci[1]&&20<ci[2])
  })
}

barruanCI <- testCI(laginak, FUN1=getCI)

barruanCIQuantile <- testCI(laginak, FUN1=getCIBootQuantile)
#barruanCIT <- testCI(laginak, FUN1 = getCIBootT)

sum(barruanCI)/N
sum(barruanCIQuantile)/N
#sum(barruanCIT)/N
```

3. ariketa Errepikatu aurreko ariketa baina hasierako populazioa banaketa esponentzial bat dela asumituz ($X \sim Exp(\lambda = 2)$). Zer ondorio atera dituzu?