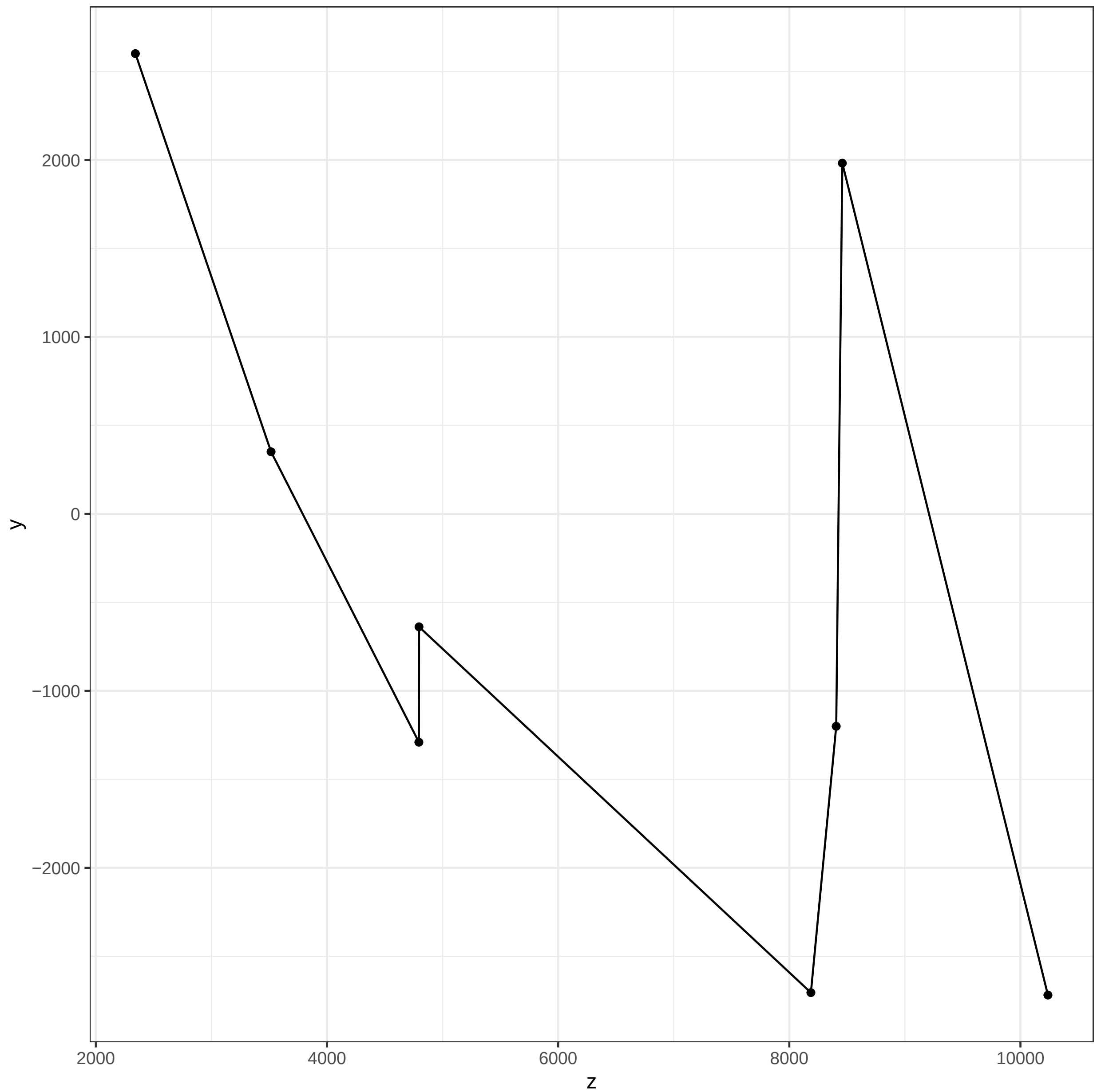
Tema 6. Estimación de sesgos mediante remuestreo

Se cargan los datos de tipos de parches de hormonas *patch* desde la librería `bootstrap` (que contiene los ficheros de datos del libro de Efron y Tibshirani (1993)).

```
library(bootstrap)
data(patch)

library(ggplot2)
qplot(z, y, data = patch, geom = c("point", "line")) + theme_bw()
```



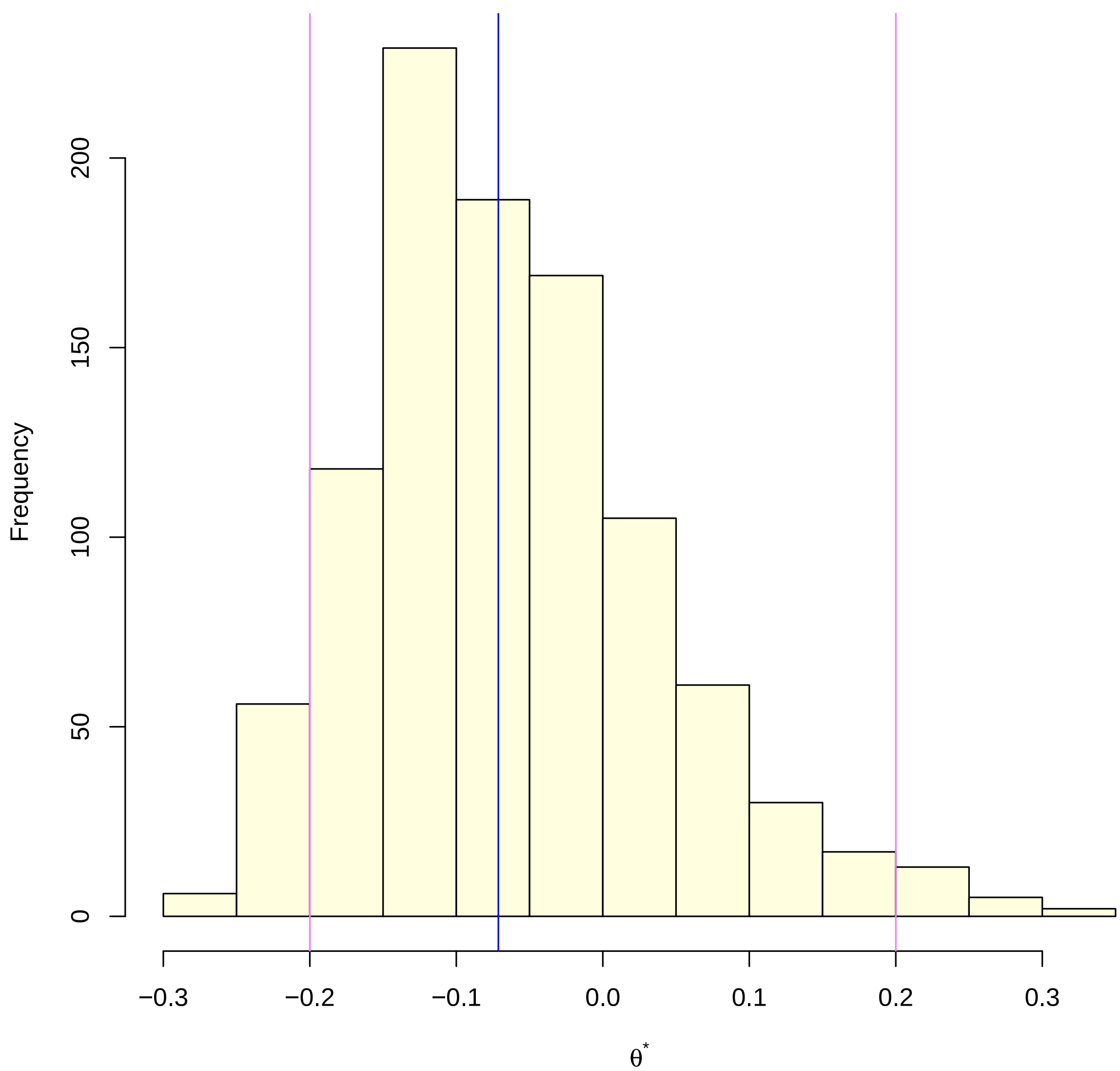
Aplicando la librería `boot` se tiene que:

```
library(boot)

ratiofun = function(datos, ind) {
    Y = datos[ind, 6]
    Z = datos[ind, 5]
    return(mean(Y)/mean(Z))
}
```

```r
patch.boot = boot(patch, statistic = ratiofun, R = 1000)

(theta.hat = ratiofun(patch, 1:8))  # Estimador plug-in original
```

```
[1] -0.0713061
```

```r
hist(patch.boot$t, xlab = expression(theta^"*"), col = "lightyellow", main = "")
abline(v = c(-0.2, 0.2), col = "violet")
abline(v = theta.hat, col = "blue")
```



Se obtiene el error estándar bootstrap

```
sd(patch.boot$t)
```

```
[1] 0.1006858
```

Se calcula también el estimador bootstrap del sesgo (*bias*):

```
(mean(patch.boot$t) - theta.hat)
```

```
[1] 0.007237734
```

O bien a partir de la salida del comando de `boot`.

```
patch.boot
```

```

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = patch, statistic = ratiofun, R = 1000)


Bootstrap Statistics :
      original       bias    std. error
t1* -0.0713061 0.007237734   0.1006858
```

Alternativamente

```
N = dim(patch)[1]
B = 1000

ratioHatBoot = replicate(B, {
    ind = sample(1:N, replace = TRUE)
    with(patch[ind, ], mean(y)/mean(z))
})
```

Probabilidad que $|\hat{\theta}| > 0.2$

```
(sum(abs(ratioHatBoot) > 0.2)/B)
```

```
[1] 0.09
```

Estimador *plug-in* original

```
(thetaHat = with(patch, mean(y)/mean(z)))
```

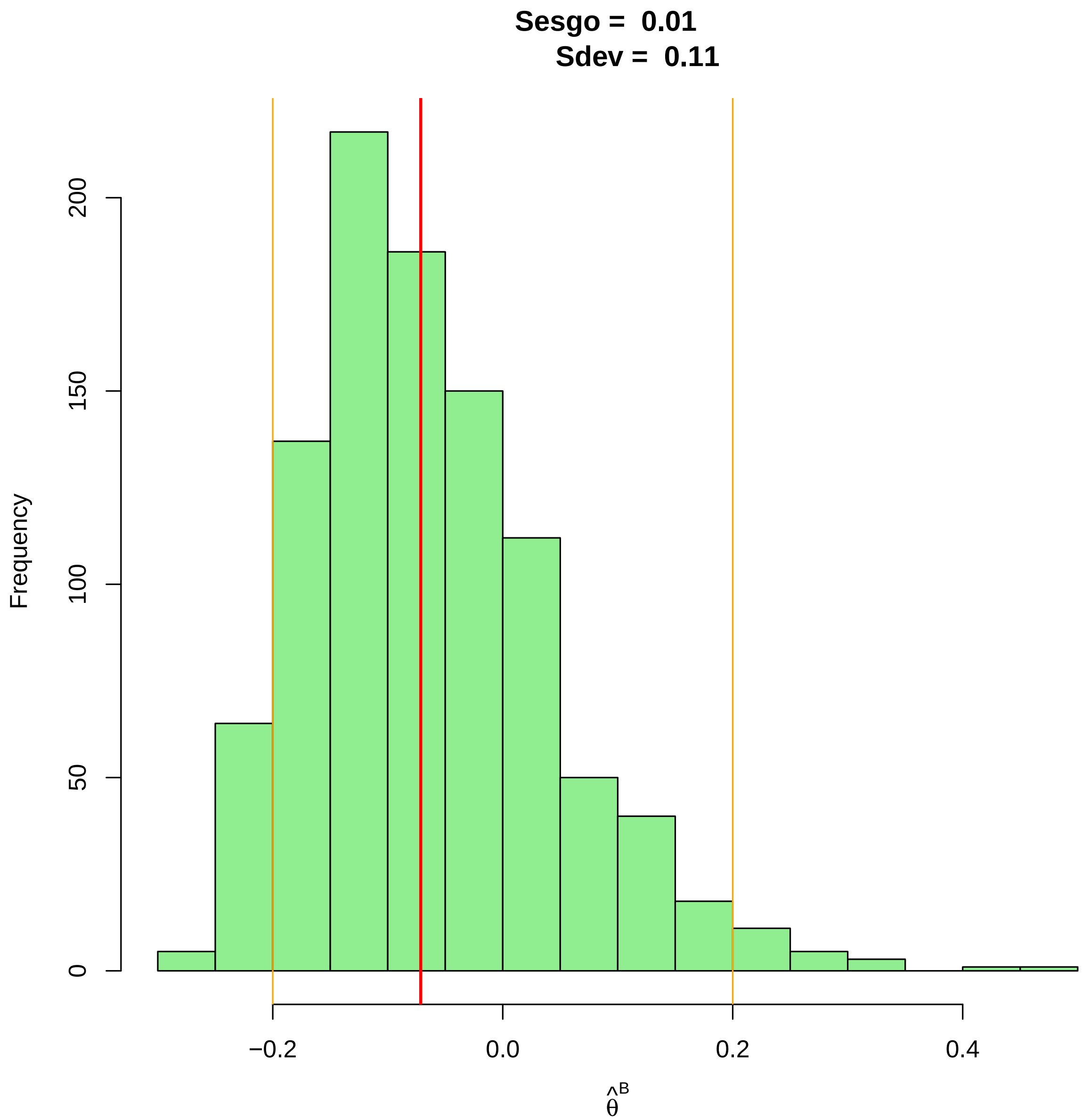
```
[1] -0.0713061
```

Estimador bootstrap del sesgo (*bias*):

```
(bias = mean(ratioHatBoot) - thetaHat)
```

```
[1] 0.006742871
```

```
hist(ratioHatBoot, main = paste("Sesgo = ", round(abs(thetaHat - mean(ratioHatBoot)),
    2), "
     Sdev = ", round(sd(ratioHatBoot), 2)), col = "lightgreen",
    xlab = expression(hat(theta)^B))
abline(v = thetaHat, col = "red", lwd = 2)
abline(v = c(-0.2, 0.2), col = "orange")
```



Estimador del sesgo mediante **jackknife**:

```
datos = patch
D = data.frame(Y = patch[, 6], Z = patch[, 5])

ratiofun = function(ind, D) {
    return(mean(D[ind, 1])/mean(D[ind, 2]))
}

jackknife(1:8, theta = ratiofun, D)
```

```
$jack.se
[1] 0.1055278

$jack.bias
[1] 0.008002488

$jack.values
[1] -0.05711856 -0.12849970 -0.02145610 -0.13245033 -0.05067038 -0.08404803
[7] -0.06486298 -0.02219698

$call
jackknife(x = 1:8, theta = ratiofun, D)
```

Corrección del sesgo

```
n = dim(patch)[[1]]

z = patch$oldpatch - patch$placebo
y = patch$newpatch - patch$oldpatch
```

Media ponderada por pesos

```
mediaPond = function(p, x) {
    sum(x * p)
}
```

Cálculo del ratio usando medias ponderadas

```
elratioP = function(p, x, y) {
    mediaPond(p, x)/mediaPond(p, y)
}
```

Estimador *plug-in* de θ:

```
p.hat = rep(1/n, n)

(theta.hat = elratioP(p.hat, y, z))
```

```
[1] -0.0713061
```

Calculas el vector $\overline{\mathbf{P}}^*$

```
nboot = 1000
theta.star = numeric(nboot)

n = dim(patch)[[1]]
p.barpre = rep(0, n)

for (i in 1:nboot) {
    k.star = sample(n, replace = TRUE)

    # tabulate() cuenta el número de ocurrencias de los valores enteros en
    # un vector

    # Se calcula la proporción de apariciones de cada valor del vector
    p.star = tabulate(k.star)/n

    theta.star[i] = elratioP(p.star, y, z)
    p.barpre = p.barpre + p.star
}
```

Vector $\overline{\mathbf{P}}^*$:

```
(P.bar.ast = p.barpre/nboot)
```

```
[1] 0.127375 0.121875 0.118000 0.124625 0.121625 0.135875 0.144125 0.172125
```

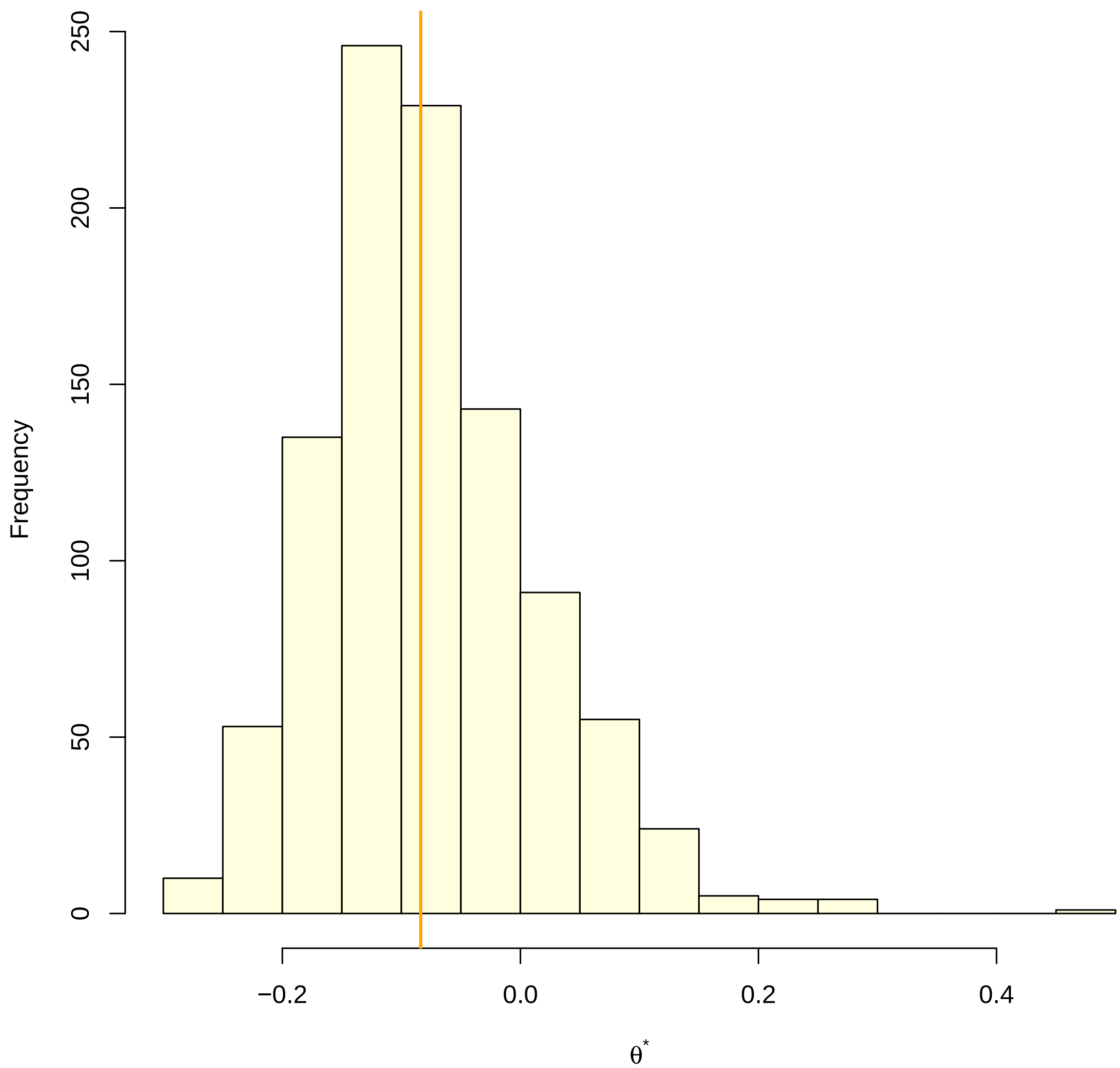
Estimador bootstrap del sesgo mejorado

Estimador $\bar{\theta}$:

```r
(theta.bar = elratioP(P.bar.ast, y, z))
```

```
[1] -0.0837641
```

```r
hist(theta.star, xlab = expression(theta^"*"), col = "lightyellow", main = "")
abline(v = theta.bar, col = "orange", lwd = 2)
```



Estimador del **sesgo corregido**:

```r
(bias.hat = mean(theta.star) - theta.bar)
```

```
[1] 0.006545009
```

Estimador de $\hat{\theta}$ (corregido del sesgo):

```r
theta.hat - bias.hat
```

```
[1] -0.0778511
```