Bootstrapping Confidence Intervals

January 9, 2022

Arman Rezaei - 9723034

[1]:
```
set.seed(9723034)
```

We want to generate 16 normal random variates with $\mu = 4$ (the last digit of my student number) and standard deviation $\sigma = 2$.

[2]:
```
# generate 16 random samples from normal distribution
mu <- 4
sigma <- 2
size <- 16

samples <- rnorm(size, mean = mu, sd = sigma)
```

The 95% confidence interval for such a distribution is calculated below.

[3]:
```
# calculate 95% confidence interval
round(c(mu - 1.96*sigma, mu + 1.96*sigma), 2)
```

1. 0.08 2. 7.92

0.0.1 Estimated Mean Confidence Interval

Now, assuming the standard deviation is unknown, we would like to calculate the mean's confidence interval estimate using bootstrapping.

[4]:
```
m <- 1000
# the mean estimate array
theta.hat <- numeric(m)
for (i in 1:m) {
    new.samples <- sample(samples, size = size, replace = TRUE)
    theta.hat[i] <- mean(new.samples)
}

# mean of means
final.mean <- mean(theta.hat)
final.mean

# standard error
mean.se <- sd(theta.hat)
```

```
mean.se
```

4.80886789975342

0.525517116379012

The first method is using the *Standard Normal Confidence Interval* approach:

$$\hat{\theta} \pm z_{\alpha/2} se(\hat{\theta})$$

[5]:
```
# 95% CI estimate
stdnorm.ci <- c(final.mean - 1.96*mean.se, final.mean + 1.96*mean.se)
stdnorm.ci
```

1. 3.77885435165056 2. 5.83888144785628

The second approach uses *percentiles*:

$$\left(\hat{\theta}_{\alpha/2}, \quad \hat{\theta}_{1-\alpha/2}\right)$$

[6]:
```
# 95% CI estimate using percentiles
percentil.ci <- quantile(theta.hat, c(0.025, 0.975))
percentil.ci
```

2.5\% 3.7182633165611 **97.5\%** 5.80055015154562

Another method would be using the *Basic Confidence Interval Method*:

$$\left(2\hat{\theta} - \hat{\theta}_{1-\alpha/2}, \quad 2\hat{\theta} - \hat{\theta}_{\alpha/2}\right)$$

[7]:
```
basic.ci <- c(2*final.mean - quantile(theta.hat, 0.975), 2*final.mean -␣
 ↪quantile(theta.hat, 0.025))
names(basic.ci) <- c("2.5%", "97.5%")
basic.ci
```

2.5\% 3.81718564796122 **97.5\%** 5.89947248294574

Finally, we can use the `boot` package and its `boot.ci` function to calculate all these statistics at once.

[8]:
```
# install.packages("boot")
library(boot)

theta.mean.boot <- function(data, index) {
    mean(data[index])
}

boot.obj <- boot(samples, statistic = theta.mean.boot, R = m)
boot.obj
```

```
Warning message:
"package 'boot' was built under R version 4.1.2"


ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = samples, statistic = theta.mean.boot, R = m)


Bootstrap Statistics :
    original        bias    std. error
t1* 4.785733 0.003897573   0.5137945
```

[9]:

```
boot.ci(boot.obj, type = c("norm", "perc", "basic"))
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot.obj, type = c("norm", "perc", "basic"))

Intervals :
Level      Normal              Basic              Percentile
95%   ( 3.775,  5.789 )   ( 3.785,  5.773 )   ( 3.798,  5.786 )
Calculations and Intervals on Original Scale
```

Between all these CIs, the one with the largest range is the most suitable one; namely, the *standard normal* approach.

0.0.2 Estimated Variance Confidence Interval

We will repeat the same process for the variance.

[10]:

```
m <- 1000
# the var estimate array
theta.hat <- numeric(m)
for (i in 1:m) {
    new.samples <- sample(samples, size = size, replace = TRUE)
    theta.hat[i] <- var(new.samples)
}

# mean of variances
final.var <- mean(theta.hat)
final.var

# standard error
var.se <- sd(theta.hat)
```

```
var.se
```

4.18094908165836

1.43681269797142

[11]:
```
# 95% CI estimate
stdnorm.ci <- c(final.var - 1.96*var.se, final.var + 1.96*var.se)
stdnorm.ci
```

1. 1.36479619363437 2. 6.99710196968235

[12]:
```
# 95% CI estimate using percentiles
percentil.ci <- quantile(theta.hat, c(0.025, 0.975))
percentil.ci
```

2.5\% 1.74776771688894 **97.5\%** 7.1926556511908

[13]:
```
# 95% using basic CI
basic.ci <- c(2*final.var - quantile(theta.hat, 0.975), 2*final.var -␣
 ↪quantile(theta.hat, 0.025))
names(basic.ci) <- c("2.5%", "97.5%")
basic.ci
```

2.5\% 1.16924251212592 **97.5\%** 6.61413044642778

[14]:
```
# 95% CI using boot.ci
library(boot)

theta.var.boot <- function(data, index) {
    var(data[index])
}

boot.obj <- boot(samples, statistic = theta.var.boot, R = m)
boot.obj
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = samples, statistic = theta.var.boot, R = m)


Bootstrap Statistics :
    original     bias    std. error
t1* 4.505634 -0.2364257     1.52481
```

[15]:
```
boot.ci(boot.obj, type = c("norm", "perc", "basic"))
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot.obj, type = c("norm", "perc", "basic"))

Intervals :
Level      Normal              Basic              Percentile
95%   ( 1.753,  7.731 )   ( 1.584,  7.257 )   ( 1.754,  7.428 )
Calculations and Intervals on Original Scale
```