

1 Survival time

As a small example, the survival times of 9 rats were 10, 27, 30, 40, 46, 51, 52, 104, and 146 days. Because of the skewness in the data, consider estimating the population median survival time θ through the sample median.

CODE FILENAME: ../R/01_01_load_data.R*****

```

survival_times <- c(10, 27, 30, 40, 46, 51, 52, 104, 146)
sample_median <- median(survival_times)
sample_mean <- mean(survival_times)
seed <- 7
B1 <- 1000; B2 <- 50
n <- 9

bootstrap_fn <- function(estimate = "median", meth = "percentile") {
  # taking 1st level boot
  survival_boot <- sample(survival_times, n, replace = TRUE)

  if (estimate == "median") {
    est_boot <- median(survival_boot)
    if (meth == "percentile") {
      return(est_boot)
    } else if (meth == "bootstrap_t") {
      sample_est <- median(survival_times)
      # taking 2nd level boot
      est_boot2 <- replicate(B2, {
        survival_boot2 <- sample(survival_boot, n, replace = TRUE)
        median(survival_boot2)
      })
    }
  } else if (estimate == "mean") {
    est_boot <- mean(survival_boot)
    if (meth == "percentile") {
      return(est_boot)
    } else if (meth == "bootstrap_t") {
      sample_est <- mean(survival_times)
      #taking 2nd level boot
      est_boot2 <- replicate(B2, {
        survival_boot2 <- sample(survival_boot, n, replace = TRUE)
        mean(survival_boot2)
      })
    }
  }

  se_boot <- sd(est_boot2)
  t_boot <- (est_boot - sample_est) / se_boot
  result_list <- list(r = est_boot, t = t_boot)
  return(result_list)
}

end-----

```

1.1 Bootstrap-t method: median

Compute a 95% CI for θ using the bootstrap- t method. Use $B_1 = 1000$ first-level bootstrap samples and $B_2 = 50$ second level bootstrap samples (to estimate the standard error).

We are 95% confident that the true value of the median is between 20.46346 and 78.24754.

CODE FILENAME: ../R/01_02_bst_median.R*****

```
set.seed(seed)
res = sapply(1:B1, function(.){bootstrap_fn(estimate = "median",
                                             meth = "bootstrap_t")})

ses <- unlist(res[1,])
tbs <- unlist(res[2,])

se_median <- sd(ses) #bootstrap estimate of the SE
lower <- sample_median - quantile(tbs,.975)*se_median
upper <- sample_median - quantile(tbs,.025)*se_median
#c(lower,upper)
```

end-----

1.2 Bootstrap percentile CI: median

Compute a 95% CI for θ using the bootstrap percentile CI with $B = 1000$ bootstrap samples.

We are 95% confident that the true value of the median is between 27 and 53.3.

CODE FILENAME: ../R/01_03_percentile_median.R*****

```
set.seed(seed)
res_median = sapply(1:B1, function(.){bootstrap_fn(estimate = "median",
                                                    meth = "percentile")})

#c(quantile(res_median,.025), quantile(res_median,.975))
```

end-----

1.3 Reverse bootstrap percentile CI (aka basic bootstrap): mean

Compute a 95% confidence interval for the mean time between failures θ using the basic bootstrap method with $B = 1000$ bootstrap samples.

We are 95% confident that the true value of the mean is between 29.21667 and 78.66944.

CODE FILENAME: ../R/01_04_reversepercentile_mean.R*****

```
set.seed(seed)
res_mean = sapply(1:B1, function(.){bootstrap_fn(estimate = "mean",
                                                  meth = "percentile")})

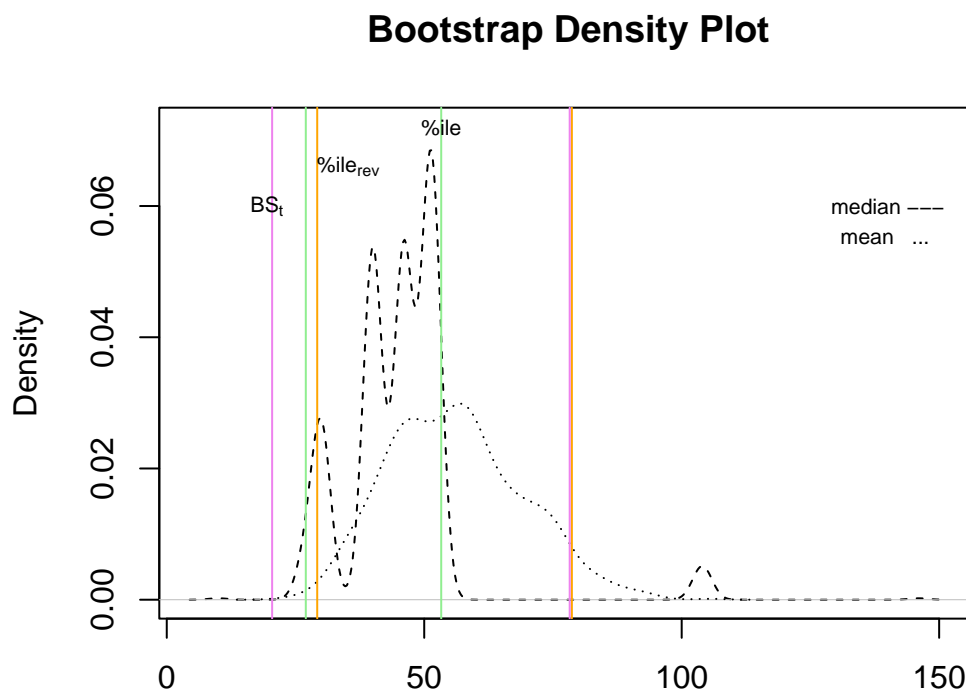
#c(2*sample_mean - quantile(res_mean,.975),
#  2*sample_mean - quantile(res_mean,.025))
```

end-----

1.4 Density estimate

Plot a density estimate of the data. In R, you can do this through the density function. Compare the results in parts 1.1, 1.2, and 1.3.

For the selected seed, the percentile CI captures the bulk of the median values from the bootstrap samples. The bootstrap-t (median) CI appears to cover areas of large concentration from both the percentile (median) and reverse percentile CI (mean). I also tried a number of seeds and noticed across all cases that: 1) The lower CI endpoint of the basic and the percentile intervals are close to each other. 2) The upper CI endpoint of basic and bootstrap-t intervals are close to each other but the lower CI endpoint of the latter is always less than that of the former. 3) Bootstrap-t captures the bulk of the statistics (mean and median). 4) The upper limit of the percentile interval could be erratic due to the presence of outliers.



CODE FILENAME: ../R/01_05_density_plot.R*****

```
plot(density(res_median),
     ylim = range(density(res_median)$y, density(ses)$y),
     lty = 'dashed', main = "Bootstrap Density Plot", xlab = '')
lines(density(res_mean), lty = 'dotted')

abline(v=2*sample_mean - quantile(res_mean,.975),col="orange")
abline(v=2*sample_mean - quantile(res_mean,.025),col="orange")
text(2*sample_mean - quantile(res_mean,.975)+6, 0.066,
     expression(paste("%il", e[rev])), cex=0.7)

abline(v=quantile(res_median,.025),col="lightgreen")
abline(v=quantile(res_median,.975),col="lightgreen")
text(quantile(res_median,.975), 0.072,
```

```

        expression(paste("%", ile)), cex=0.7)

abline(v=quantile(lower,.025),col="violet")
abline(v=quantile(upper,.975),col="violet")
text(quantile(lower,.025)-1, 0.06, expression(BS[t]), cex=0.7)

text(140, 0.06, "median ---", cex=0.7)
text(139.5, 0.055, "mean   ...", cex=0.7)

end-----

```

2 Spatial test

Consider the spatial test data from Table 14.1 of Efron and Tibshirani (1993) shown below. From the table's description, it is clear that the measurements A and B are paired. Suppose the data consist of a random sample from an unknown joint distribution of A and B. Whenever ratios are scientifically or statistically preferred to differences, we gain stability by considering the logarithm of the ratios. Let $\theta_1 = \log E\left(\frac{A_i}{B_i}\right)$, $\theta_2 = E\left(\log \frac{A_i}{B_i}\right)$ for all i . Exclude observation #14 because the logarithm of its ratio is undefined. Use 2000 bootstrap samples.

2.1 Bootstrap percentile CI for θ_1

Compute a bootstrap percentile confidence interval for θ_1 .

We are 95% confident that the true value of θ_1 is between -0.06867 and 0.16487.

```

CODE FILENAME: ../R/02_02_percentile_logmean.R*****

B <- 2000

plug_in_estimator <- function(A,B, est = "theta_1"){
  if (est == "theta_1") {
    return(log(mean(A/B)))
  } else if (est == "theta_2") {
    return(mean(log(A/B)))
  }
}

theta_1_hat_star <- c()
theta_2_hat_star <- c()
set.seed(7)
for (b in 1:B) {
  bs_sample <- spatial_test_data[sample(spatial_test_data[,1],
                                         n1,
                                         replace = TRUE),]
  theta_1_hat_star[b] <- plug_in_estimator(bs_sample$A,
                                           bs_sample$B,
                                           est = "theta_1")
  theta_2_hat_star[b] <- plug_in_estimator(bs_sample$A,
                                           bs_sample$B,
                                           est = "theta_2")
}

```

```

theta_1_pci <- c(quantile(theta_1_hat_star,.025),
                 quantile(theta_1_hat_star,.975))
theta_1_pci
end-----

```

2.2 BC_a CI for θ_1

Compute a BC_a confidence interval for θ_1 . Interpret the CI.

We are 95% confident that the true value of θ_1 is between -0.10674 and 0.13736.

```

CODE FILENAME: ../R/02_03_bca_logmean.R*****

# helper functions=====

bias_correction <- function(bootstrap_estimates,
                           plug_in_estimate,
                           B){
  return(qnorm(sum(bootstrap_estimates < plug_in_estimate)/B))
}

acceleration_parameter <- function(data = spatial_test_data,
                                   est = "theta_1"){
  for (i in n1) {
    summ <- ((sum(
      plug_in_estimator(data$A[-i],
                        data$B[-i],
                        est = est))/n1) - plug_in_estimator(data$A[-i],
                                                              data$B[-i],
                                                              est = est))
    return(sum(summ^3) / (6*((sum(summ)^2))^(3/2)))
  }
}

alpha <- function(confid = .975,
                  est = "theta_1",
                  bootstrap_estimates = theta_1_hat_star,
                  plug_in_estimate = plug_in_estimate_theta_1){

  bc <- bias_correction(bootstrap_estimates = bootstrap_estimates,
                       plug_in_estimate = plug_in_estimate,
                       B = B)

  ap <- acceleration_parameter(data = spatial_test_data,
                              est = est)

  return(pnorm(bc + (bc + qnorm(confid))/(1-(ap*(bc + qnorm(confid))))))
}

BC_a <- function(confid = .975,

```

```

        est = "theta_1",
        bootstrap_estimates = theta_1_hat_star,
        plug_in_estimate = plug_in_estimate_theta_1
    ){
        return(c(quantile(bootstrap_estimates,
                        alpha(1-confid, est = est,
                        plug_in_estimate = plug_in_estimate)),
                quantile(bootstrap_estimates,
                        alpha(confid, est = est,
                        plug_in_estimate = plug_in_estimate))))
    }

# BCa for logmean computation =====

plug_in_estimate_theta_1 <- plug_in_estimator(spatial_test_data$A,
                                             spatial_test_data$B,
                                             est = "theta_1")

theta_1_BCa <- BC_a(confid = .975,
                    est = "theta_1",
                    bootstrap_estimates = theta_1_hat_star,
                    plug_in_estimate = plug_in_estimate_theta_1)

theta_1_BCa
#if bc & ap are 0: (-0.06867057, 0.16486927)

end-----

```

2.3 BC_a CI for θ_2

Compute a BC_a confidence interval for θ_2 .

We are 95% confident that the true value of θ_2 is between -0.26297 and 0.01849.

CODE FILENAME: ../R/02_05_bca_meanlog.R*****

```

plug_in_estimate_theta_2 <- plug_in_estimator(spatial_test_data$A,
                                             spatial_test_data$B,
                                             est = "theta_2")

theta_2_BCa <- BC_a(confid = .975,
                    est = "theta_2",
                    bootstrap_estimates = theta_2_hat_star,
                    plug_in_estimate = plug_in_estimate_theta_2)

#theta_2_BCa

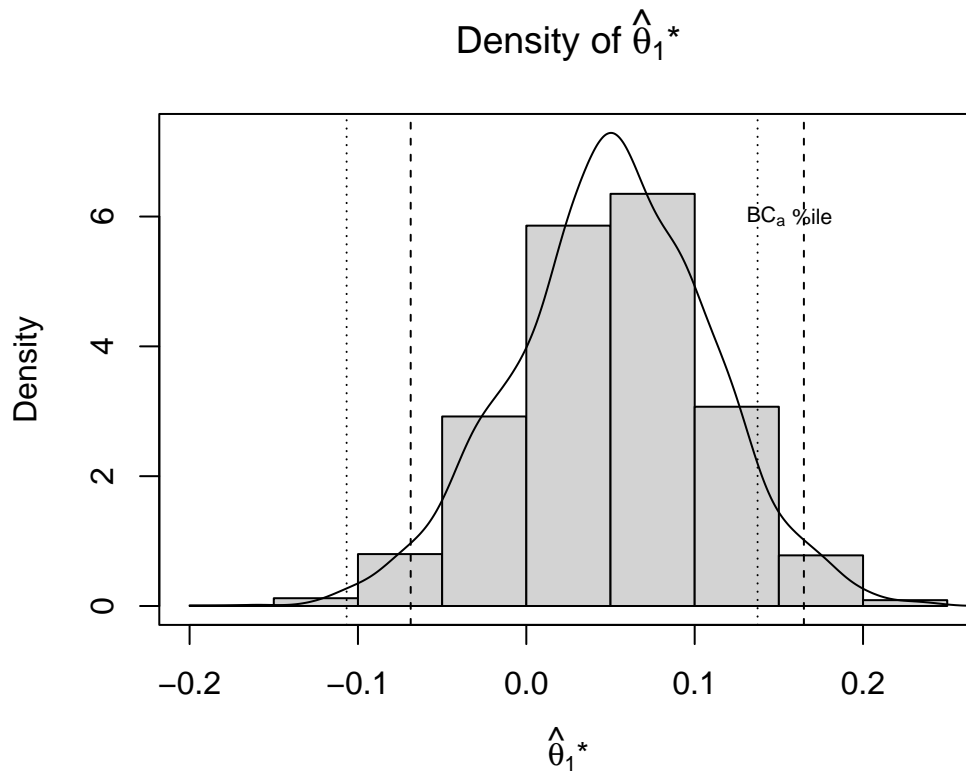
end-----

```

2.4 Bootstrap percentile vs BC_a CI for θ_1

Compare your CIs in 2.1 and 2.2. How different are the two CIs?

The endpoints of the BC_a CI are found to the left of the bootstrap percentile CI. The latter covers the bulk of the statistic while the former is adjusted towards the left.



CODE FILENAME: ../R/02_06_compare_ci.R*****

```
hist(theta_1_hat_star,
     prob = TRUE,
     main = expression(paste("Density of ", hat(theta)[1], "*")),
     xlab = expression(paste(hat(theta)[1], "*")),
     ylim = range(density(theta_1_hat_star)$y, density(theta_1_hat_star)$y))

lines(density(theta_1_hat_star))

box(col = "black")

abline(v=theta_1_pci[1],lty="dashed")
abline(v=theta_1_pci[2],lty="dashed")
text(theta_1_pci[2]+0.005, 6, expression(paste("%", ile)), cex=0.7)

abline(v=theta_1_BCa[1],lty="dotted")
abline(v=theta_1_BCa[2],lty="dotted")
text(theta_1_BCa[2]+0.005, 6, expression(BC[a]), cex=0.7)

end-----
```

3 References

Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap*. Chapman & Hall/CRC.

4 Appendix

4.1 Code to read data for item 2

```
CODE FILENAME: ../R/02_01_load_data.R

data <- "../.../problems/ps_02/datasets/spatial_test_data.RData"

if (file.exists(data)) {
  print(paste(c("The file exists; loading", data), collapse = ' '))
  load(data)
} else {
  paste(c("The file does not exist; creating, loading and saving", data),
        collapse = ' ')
  spatial_test_data <- data.frame(
    'i' = 1:25,
    'A' = c(48, 36, 20, 29, 42, 42, 20, 42, 22, 41, 45, 14, 6,
            33, 28, 34, 4, 32, 24, 47, 41, 24, 26, 30, 41),
    'B' = c(42, 33, 16, 39, 38, 36, 15, 33, 20, 43, 34, 22, 7,
            34, 29, 41, 13, 38, 25, 27, 41, 28, 14, 28, 40)
  )
  n1 <- dim(spatial_test_data)[1]
  seed <- 7

  save(spatial_test_data, seed, n1, file=data)
}

rm(data)
```