1 Breast cancer

Patients with advanced terminal cancer of the breast were treated with ascorbate in an attempt to prolong survival. The table below shows survival times (days).

Table 1: Survival Times (Days) for Patients with Breast Cancer

24	40	719	727	791	1,166	1,235	1,581	1,804	3,460	3,808

1.1 Bootstrap-t

Use the bootstrap-t method to construct 95% CI for the mean survival time. *When doing the bootstrap, work with the data on the natural log scale and then exponentiate the resulting interval boundaries.* Interpret the resulting interval.

We are 95% confident that the true value of mean survival time is between 296.8852574 and 2034.9199081.

```
CODE FILENAME: ../R/01_01_bst.R-------------------------------------------

    data <- c(24, 40, 719, 727, 791, 1166, 1235, 1581, 1804, 3460, 3808)
    log_data <- log(data)
    sample_mean <- mean(log_data)
    seed <- 7
    B1 <- 1000
    B2 <- 25
    n <- length(data)

    bootstrap_fn <- function(data = log_data, meth = "percentile") {
      # taking 1st level boot
      boot <- sample(data, n, replace = TRUE)

      est_boot <- mean(boot)
      if (meth == "percentile") {
        return(est_boot)
      } else if (meth == "bootstrap_t") {
        sample_est <- mean(data)
        #taking 2nd level boot
        est_boot2 <- replicate(B2, {
          boot2 <- sample(data, n, replace = TRUE)
          mean(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)
    }

    set.seed(seed)
    res = sapply(1:B1, function(.){bootstrap_fn(data = log_data,
                                              meth = "bootstrap_t")})
    ses <- unlist(res[1,]); tbs <- unlist(res[2,])
    se_mean <- sd(ses) #bootstrap estimate of the SE
    lower <- sample_mean - quantile(tbs,.975)*se_mean
    upper <- sample_mean - quantile(tbs,.025)*se_mean

    #c(exp(lower),exp(upper)): 296.8853 2034.9199

end--------------------------------------------------------------------------
```

1.2 BCa

Use the *BCa* method to construct a 95% CI for the mean survival time.

We are 95% confident that the true value of mean survival time is between 213.9662033 and 1448.1353995.

```
CODE FILENAME: ../R/01_02_bca.R---------------------------------------------

    plug_in_estimator <- function(data) return(mean(data))

    B <- 1000
    theta_hat_star <- c()
    set.seed(seed)

    for (b in 1:B) {
      bs_sample <- sample(log_data, n, replace = TRUE)
      theta_hat_star[b] <- plug_in_estimator(bs_sample)
    }

    # helper functions===================================================

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

    acceleration_parameter <- function(data = log_data){

      th.jack <- sapply(1:n, function(x){plug_in_estimator(data[-x])})

      L <- mean(th.jack) - th.jack

      return( sum(L^3)/(6 * sum(L^2)^1.5) )
    }

    alpha <- function(confid = .975,
                      bootstrap_estimates = theta_hat_star,
                      plug_in_estimate = plug_in_estimate,
                      data = log_data){
```

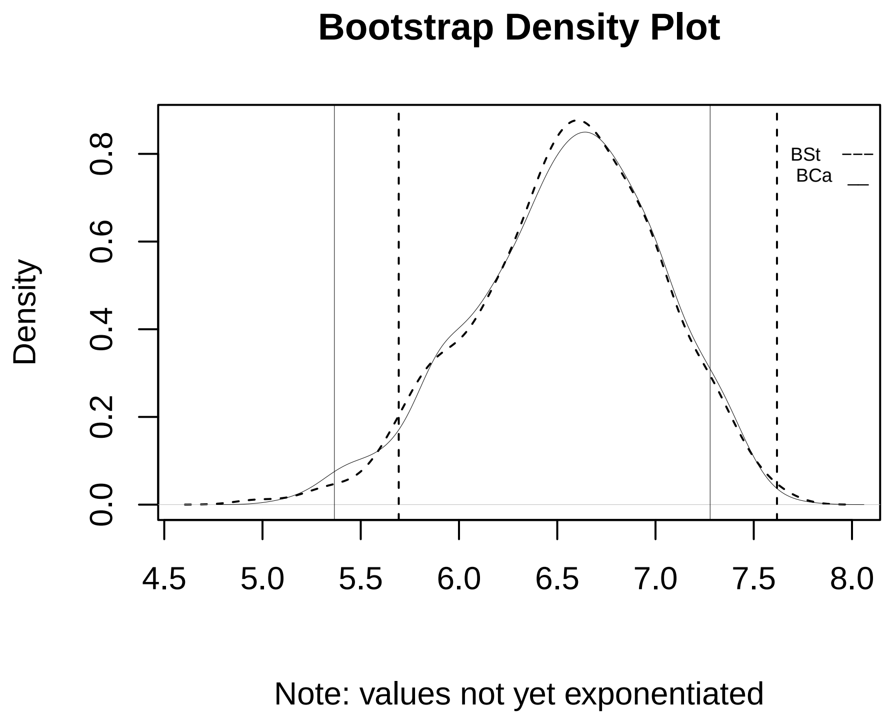
```
      bc <- bias_correction(bootstrap_estimates = bootstrap_estimates,
                            plug_in_estimate = plug_in_estimate,
                            B = B)

      ap <- acceleration_parameter(data)

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

    BC_a <- function(confid = .975,
                     bootstrap_estimates = theta_hat_star,
                     plug_in_estimate = plug_in_estimate
    ){
      return(c(quantile(bootstrap_estimates,
                        alpha(1-confid,
                              bootstrap_estimates = bootstrap_estimates,
                              plug_in_estimate = plug_in_estimate)),
               quantile(bootstrap_estimates,
                        alpha(confid,
                              bootstrap_estimates = bootstrap_estimates,
                              plug_in_estimate = plug_in_estimate))))
    }

    # BCa for mean computation ============================================

    plug_in_estimate <- plug_in_estimator(log_data)

    theta_BCa <- BC_a(confid = .975,
                      bootstrap_estimates = theta_hat_star,
                      plug_in_estimate = plug_in_estimate)

    # exp(theta_BCa): 213.9662  1448.1354
end-------------------------------------------------------------------------
```

1.3 Comparison

Compare your intervals.

Both intervals (shown below - NOT yet exponentiated) for BSt and BCa contain the sample mean survival time, 6.5586034. It is a bit wider for BSt (upper - lower: 1.924866) than BCa (upper - lower: 1.912214). Upon exponentiation, the difference between the two CI's became larger. BSt (upper - lower: 1738.0346508) is hugely wider than that of BCa (upper - lower: 1234.1691961).

Both of these CI's are second order accurate but BCa is range preserving and transformation respecting, unlike BSt



```
CODE FILENAME: ../R/01_03_compare.R---------------------------------------------

    plot(density(ses),
         ylim = range(density(ses)$y, density(ses)$y),
         lty = 'dashed', main = "Bootstrap Density Plot", xlab = '')
    lines(density(theta_hat_star), lwd = 0.25)

    abline(v=theta_BCa[1], lwd = 0.25)
    abline(v=theta_BCa[2], lwd = 0.25)

    abline(v=lower,lty = 'dashed')
    abline(v=upper,lty = 'dashed')

    text(7.9, 0.8, "BSt    ---", cex=0.6)
    text(7.9, 0.75, "BCa   __", cex=0.6)
    title(sub="Note: values not yet exponentiated")

end-------------------------------------------------------------------------
```

2 Blood flow

Does drinking coffee affect blood flow, particularly during exercise? Doctors studying healthy subjects measured myocardial blood flow (MBF) during bicycle exercise before and after giving the subjects a dose of caffeine that was equivalent to drinking two cups of coffee. The table below shows the MBF levels before (baseline) and after (caffeine) the subjects took a tablet containing 200 mg of caffeine. Note that the observations are paired. We would like to test whether or not the mean MBF is the same at baseline as it is after taking caffeine. However, we cannot guarantee normality of the observations, and therefore the paired t-test cannot be used.

Table 2: MBF (ml/min/g) for eight subjects

Subject	Baseline	Caffeine
1	6.370	4.520
2	5.690	5.440
3	5.580	4.700
4	5.270	3.810
5	5.110	4.060
6	4.890	3.220
7	4.700	2.960
8	3.530	3.200

2.1 Bootstrap test

For this kind of data, explain how you would perform a bootstrap hypothesis test to check if there is a difference in mean MBF.

Some notes:

- Since this is a test not on the distribution, but on the value of the parameter, the bootstrap test will be used.
- Since the observations are paired or matched, samples will be taken pairwise.
- Test the hypothesis that the population's average difference in MBF levels during exercise after taking nothing - baseline(X), and taking 200 mg caffeine(Y) is equal to zero.

$$\begin{aligned} H_0 &: D = \sum_{i=1}^{N} (x_i - y_i) = 0 \\ H_1 &: D = \sum_{i=1}^{N} (x_i - y_i) \neq 0 \end{aligned} \tag{2.1}$$

To administer the bootstrap hypothesis test,

1. Transform the data so that it follows the null hypothesis.

   Given X and Y, the differences of pairs are given by

$$d_i = x_i - y_i; \quad i = 1, \ldots, n \tag{2.2}$$

   The following transformation is applied so that the null hypothesis applies to the sample.

$$\tilde{d_i} = d_i - \bar{d}; \quad \bar{d} = \frac{\sum_{i=1}^{n} d_i}{n} \tag{2.3}$$

   Below is the proof that this transformation is guaranteed to result in 0.

$$\frac{1}{n}\sum_{i=1}^{n}\tilde{d_i} = \frac{1}{n}\sum_{i=1}^{n}\left(d_i - \bar{d}\right) = \frac{1}{n}\sum_{i=1}^{n} d_i - \frac{1}{n}\sum_{i=1}^{n}\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i - \frac{1}{n}n\bar{d} = \bar{d} - \bar{d} = 0 \tag{2.4}$$

2. Calculate the test statistic for the observed data: t-statistic for paired or matched samples

$$t_{obs} = \frac{\bar{\tilde{d}}}{\frac{s}{\sqrt{n}}}; \quad s = \sqrt{\frac{\sum_{i=1}^{n} \left(\tilde{d} - \bar{\tilde{d}}\right)^2}{n-1}} \tag{2.5}$$

3. Take B bootstrap samples by sampling pairs (or matched differences), with replacement, of size n from the transformed data. Let $b = 1, \ldots, B$

$$\begin{aligned} B_1^* = \{\tilde{d}_{11}^*, \ldots, \tilde{d}_{1n}^*\} &\overset{SRSWR}{\sim} \{\tilde{d}_1, \ldots, \tilde{d}_n\} \\ B_2^* = \{\tilde{d}_{21}^*, \ldots, \tilde{d}_{2n}^*\} &\overset{SRSWR}{\sim} \{\tilde{d}_1, \ldots, \tilde{d}_n\} \\ &\vdots \\ B_B^* = \{\tilde{d}_{B1}^*, \ldots, \tilde{d}_{Bn}^*\} &\overset{SRSWR}{\sim} \{\tilde{d}_1, \ldots, \tilde{d}_n\} \end{aligned} \tag{2.6}$$

4. Compute the test statistic for each bootstrap sample. This gives B values.

$$t_b^* = \frac{\bar{\tilde{d}}^*}{\frac{s^*}{\sqrt{n}}}; \quad s^* = \sqrt{\frac{\sum_{i=1}^{n} \left(\tilde{d}^* - \bar{\tilde{d}}^*\right)^2}{n-1}} \tag{2.7}$$

5. Estimate the p-value. If it is less than the significance level (α), 0.05, reject the null hypothesis.

$$P(|t^*| \geq |t_{obs}|) = \frac{\#\{|t^*| \geq |t_{obs}|\}}{B} \tag{2.8}$$

2.2 Implementation

Perform the bootstrap hypothesis test at $\alpha = 0.05$.

With the estimated p-value of TRUE, there is not enough evidence to say that there is a difference in the average MBR levels when taking 200 mg caffeine and not, during exercise.

```
CODE FILENAME: ../R/02_01_bstest.R-------------------------------------------

    caffeine_data <- data.frame(
      'baseline' = c(6.37, 5.69, 5.58, 5.27, 5.11, 4.89, 4.70, 3.53),
      'caffeine' = c(4.52, 5.44, 4.70, 3.81, 4.06, 3.22, 2.96, 3.20)
    )

    n <- nrow(caffeine_data)
    d <- caffeine_data$baseline - caffeine_data$caffeine
    x.bar <- mean(d); d_tilde <- d-x.bar # transformed
    mean_d_tilde <- mean(d_tilde)
    sigma_d_tilde <- sqrt(sum((d_tilde-mean_d_tilde)^2)/(n-1))
    t_obs_d_tilde <- mean_d_tilde/(sigma_d_tilde/sqrt(n))

    indicator <- c() #1 if the statement is TRUE, 0 if FALSE
    set.seed(seed)
    for(b in 1:B){
      caffeine_data_boot <- caffeine_data[sample(nrow(caffeine_data),
                                                 n, TRUE), ]
      d <- caffeine_data_boot$baseline - caffeine_data_boot$caffeine
      x.bar <- mean(d)
      d_tilde <- d-x.bar
      mean_d_tilde <- mean(d_tilde)
      sigma_d_tilde <- sqrt(sum((d_tilde-mean_d_tilde)^2)/(n-1))
      t_b <- mean_d_tilde/(sigma_d_tilde/sqrt(n))

      indicator[b] <- abs(t_b) >= abs(t_obs_d_tilde)
    }

    # sum(indicator)/B: 0.508

end---------------------------------------------------------------------------
```

3 Earthquake

The National Earthquake Information Center has provided annual data on the number of earthquakes per year exceeding magnitude 7.0 for the years from 1900 to 1998. Take the first difference of the data so that the number for each year represents the change since the previous year. Note that the resulting differenced series can be interpreted as annual change.

3.1 Lag-1 autocorrelation

Estimate the lag-1 autocorrelation of the annual change. In estimating the lag-1 autocorrelation, use the following formula:

$$\hat{\rho}_1 = r_1 = \frac{\sum_{t-2}^{n}(X_t - \bar{X})(X_{t-1} - \bar{X})}{\sum_{t-1}^{n}(X_t - \bar{X})^2} \tag{3.1}$$

Estimate the lag-1 autocorrelation of the annual change: -0.3661926

```
CODE FILENAME: ../R/03_01_acf1.R------------------------------------------------

    earthquake <- read.csv("earthquake.csv")
    earthquake<-ts(earthquake, start = 1990, frequency = 1)

    annual_change <- diff(earthquake, lag = 1)
    n <- length(annual_change)

    get_acf1 <- function(earthquake_data){
      annual_change <- diff(earthquake_data, lag = 1)
      n <- length(annual_change)

      num <- c(); den <- c()
      for (t in 2:n) {
        num[t-1] <- (
          annual_change[t] - mean(annual_change)
          )*(
            annual_change[t-1] - mean(annual_change)
          )
      }

      for (t in 1:n) {
        den[t] <- (annual_change[t] - mean(annual_change))^2
      }
      return(sum(num)/sum(den))
    }

    # get_acf1(earthquake): -0.3661926

end-------------------------------------------------------------------------
```

Note: Estimated autocorrelation has the same output as `acf(annual_change, lag = 1, type = "correlation", plot = FALSE)$acf[„1][2]`

3.2 Block bootstrap

Using the block bootstrap (no need to do a circular block bootstrap) with block size b = 9, k = n/b blocks, and M = 10, 000 bootstrap samples, estimate the standard error of the lag-1 autocorrelation. Use the adjacent, nonoverlapping blocking scheme in forming the blocks.

Estimated standard error of the lag-1 autocorrelation: 0.1036793

```
  CODE FILENAME: ../R/03_02_blockbs.R-------------------------------------------

      M <- 10000
      n <- length(earthquake)
      b <- 9
      k <- ceiling(n/b)

      acf1_star <- c()
      set.seed(seed)
      for (i in 1:M) {
        boot <- c()
        for (j in 1:k) {
          left <- sample(seq(1, n, b), 1, replace = TRUE)
          boot_s <- earthquake[left:(left + b - 1)]
          boot <- append(boot, boot_s)
        }
        acf1_star[i] <- get_acf1(boot)
      }

      # sd(acf1_star): 0.1036793

  end-----------------------------------------------------------------------
```

Note: `sd` has the same output as `sqrt((sum((acf1_star - mean(acf1_star))^2))/(M-1))`