FCT NOVA

Project Report

Bootstrap and Jackknife

Leonard Storcks

Course by Dr. Regina Bispo

January 12, 2024

Abstract

This project is all about the usage of resampling in statistics, e.g. to estimate standard errors of estimators. We will employ bootstrap techniques for estimation and testing in different contexts as well as the Jackknife.

Contents

1 Bootstrap Methods **1**

1.1 Caveats of Bootstrap 1

1.2 Application of Bootstrap for Estimating the Difference between Means (task 1a) 3

1.3 Bootstrap Confidence Intervals 6

1.3.1 Standard Normal Bootstrap Confidence Interval 6

1.3.2 Percentile Bootstrap Confidence Interval 7

1.3.3 Basic Bootstrap Confidence Interval 7

1.3.4 Bootstrap Confidence Intervals for Task 1b 7

1.4 Bootstrap Hypothesis Testing 9

1.4.1 Bootstrap Hypothesis Test on the Difference of Means I: Theory . . . 11

1.4.2 Bootstrap Hypothesis Test on the Difference of Means I: Example Application 12

1.4.3 Bootstrap Hypothesis Test on the Difference of Means III: Task 1c . . 13

1.5 Bootstrapping on Linear Regression Coefficients (task 2) 16

1.5.1 Linear Fit on the Geysir Data (task 2a) 16

1.6 Bootstrap estimates by resampling cases (task 2b) 17

1.7 Standard normal bootstrap confidence intervals (task 2c) 19

1.8 Testing on $H_0 : \beta_0 = 0$ or rather $H_0 : \beta_0 = c$ (task 2d) 19

2 Jackknife and Further Bootstrap Applications **22**

2.1 Bias correction and standard error estimation using Bootstrap and Jackknife (task 3a, 3b) 22

2.2 Non-parametric bootstrap method to test the hypothesis $H_0 : X \sim \mathcal{N}(2, 1)$ (task 3c) 26

References **29**

1 Bootstrap Methods

At the heart of bootstrap lies random sampling with replacement to better assess statistical estimates (assign measures of accuracy[1]) - we use a sample as an estimate of a population and bootstrap samples from the sample as estimates of the sampling distribution. Consider we have observed a sample $X_{\text{sample}} = (x_1, \ldots, x_n)$ and are interested in a population parameter θ which we can estimate as $\hat{\theta}$ from a sample. The non-parametric bootstrap method is given by

1. $\forall b \in \{1, \ldots, B\}$ (where B is the number of bootstrap samples): Generate a bootstrap sample $X^{(b)} = (x_1^{(b)}, \ldots, x_n^{(b)})$ by sampling from X with replacement and calculate $\hat{\theta}^{(b)}$ on the bootstrap sample.
2. This yields an empirical distribution of estimates $\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(B)}$ from which we can calculate the mean $\hat{\theta}^*$ and standard error $\text{SE}\left(\hat{\theta}^*\right)$.

with the bootstrap estimate and the estimated standard error given by

$$\begin{aligned} \hat{\theta}^* &= \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^{(b)} \\ \hat{\text{SE}}(\hat{\theta}) = \text{SE}\left(\hat{\theta}^*\right) &= \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}^{(b)} - \hat{\theta}^*\right)^2} \end{aligned} \tag{1}$$

The bootstrap estimate of the bias $\left(\hat{\theta}\right) = E\left[\hat{\theta}\right] - \theta$ of the estimator $\hat{\theta}$ is given by

$$\hat{\text{bias}}\left(\hat{\theta}\right) = \hat{\theta}^* - \hat{\theta} \tag{2}$$

1.1 Caveats of Bootstrap

Beware, that bootstrap is not magic - it will not retrieve information not present in a sample or fix a bad sample. Also, the bootstrap sample is generally centered at the observed statistic, not the population parameter (e.g. $\bar{X}$ not μ) - using bootstrap we will not improve on $\bar{X}$ (Hesterberg, 2015, section 2.3 (also see there for exceptions)). And if there is bias (as estimated above), the bootstrap estimate is indeed not a bias-corrected estimate - if $\hat{\theta}^*$ is greater than $\hat{\theta}$, the bias corrected estimate $\bar{\theta} = \hat{\theta} - \hat{\text{bias}}$ should be less than $\hat{\theta}$, and this

[1] For the mean e.g. a measure of accuracy is the standard error for which we have the explicit formula $\text{SE} = \sqrt{\frac{s^2}{n}}$, $\bar{X}_{\text{sample}} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X}_{\text{sample}})^2$, $X_{\text{sample}} = (x_1, \ldots, x_n)$. For most other statistics (e.g. the median) we do not have such an explicit formula. See Efron and Tibshirani, 1994, chapter 1.

kind of bias correction also has pitfalls[2] (Efron and Tibshirani, 1994, chapter 10.6, p. 138). Simply put, if from resampling a sample we get a positive bias then to get to the unbiased one we have to go "two steps back", $\hat{\theta} = \hat{\theta}^* - 2 \cdot (\hat{\theta}^* - \hat{\theta}) = 2\hat{\theta} - \hat{\theta}^*$ (eq. 10.41 in Efron and Tibshirani, 1994).

In figure 1 we illustrate that as the sample carries distributional information so do the bootstrap samples. In figure 2 we illustrate the difference between the sample estimate, bootstrap estimate and corrected estimate for a biased estimator, the standard deviation. Here the bias comes from the fact that in the calculation of the standard deviation we also estimate the mean from which the points in the sample naturally deviate less than from the true mean, so the standard deviation is (with the biased estimator) underestimated.

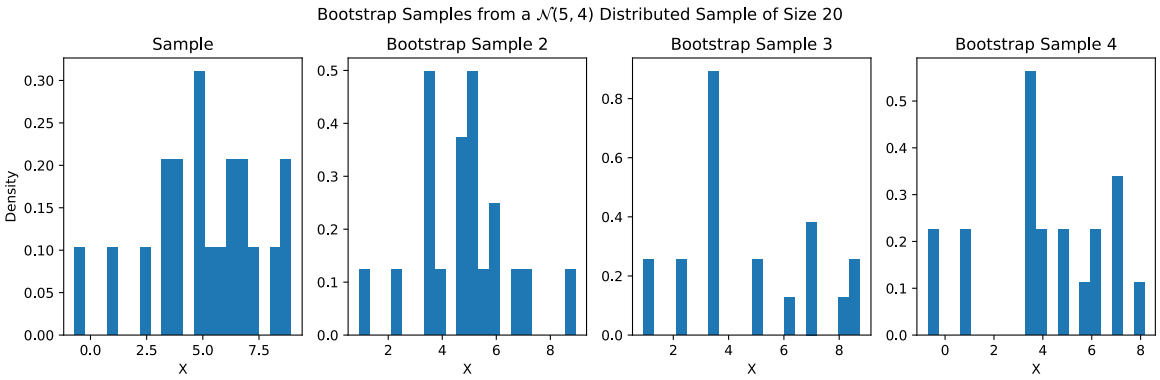


Figure 1: Bootstrap Samples from a sample.

[2] $\bar{\theta}$ might have a much larger standard error than $\hat{\theta}$.

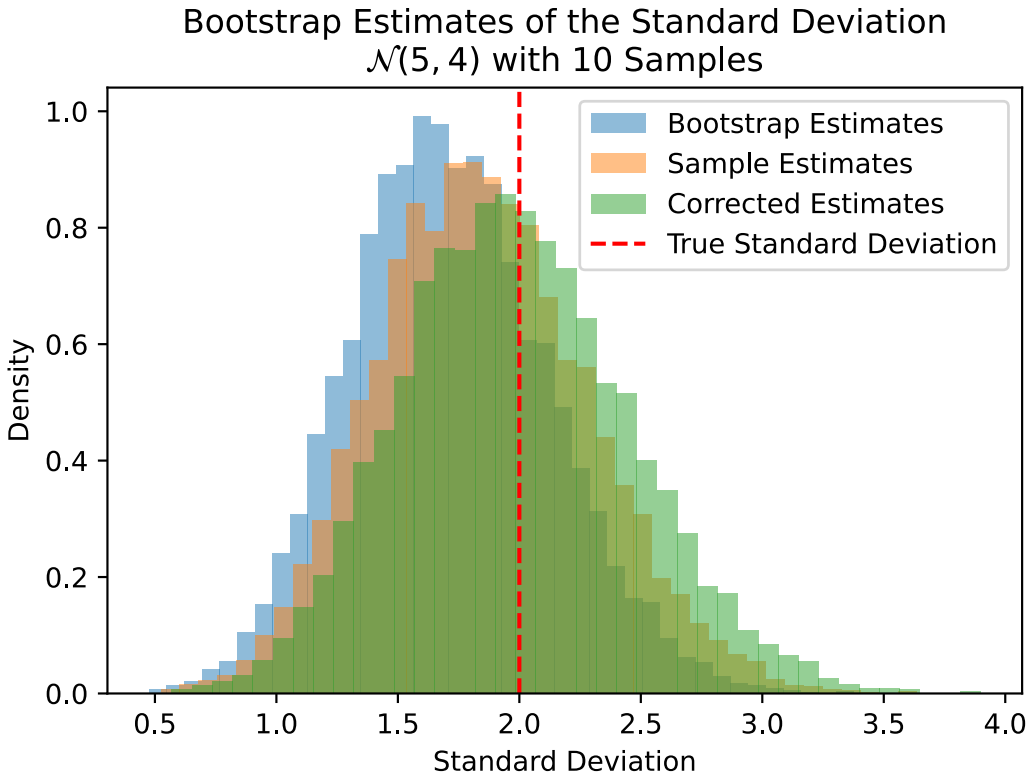


Figure 2: Illustration of the difference between the sample estimate, bootstrap estimate and corrected estimate for the biased estimate of the standard deviation.

1.2 Application of Bootstrap for Estimating the Difference between Means (task 1a)

In the following consider survival times of patients with breast cancer (sample 1) and stomach cancer (sample 2).

$$\begin{aligned} D^{(1)} &= (107, 353, 1764, 667, 990, 78, 667, 44, 9, 27), \quad n_1 = 10 \\ D^{(2)} &= (374, 253, 812, 246, 95, 367, 251, 309, 594, 826, 593, 97), \quad n_2 = 12 \end{aligned} \tag{3}$$

We will work with the logged lifetimes $X^{(1)}, X^{(2)}, x_j^{(k)} = \log d_j^{(k)}$ (see e.g. Overduin, 2004) and present kernel density estimates of both the original and log-transformed data in figure 3. Note that in the sample 1 we have an outlier at 1764 which strongly pulls the mean of the non-logged data to higher values. The log-transforms crunches down especially the outlier, shifting the mean to lower values.

As $\bar{X}^{(1)}$ is smaller than $\bar{X}^{(2)}$ (see figure 3), we will consider $\mu_2 - \mu_1$ instead of $\mu_1 - \mu_2$ throughout the task contrary to the task description. This seems to be more in alignment of what the later test for $H_0 : \mu_2 - \mu_1 \leq 0.25$ tries to get at: While $\bar{X}^{(2)} - \bar{X}^{(1)} > 0.25$ we

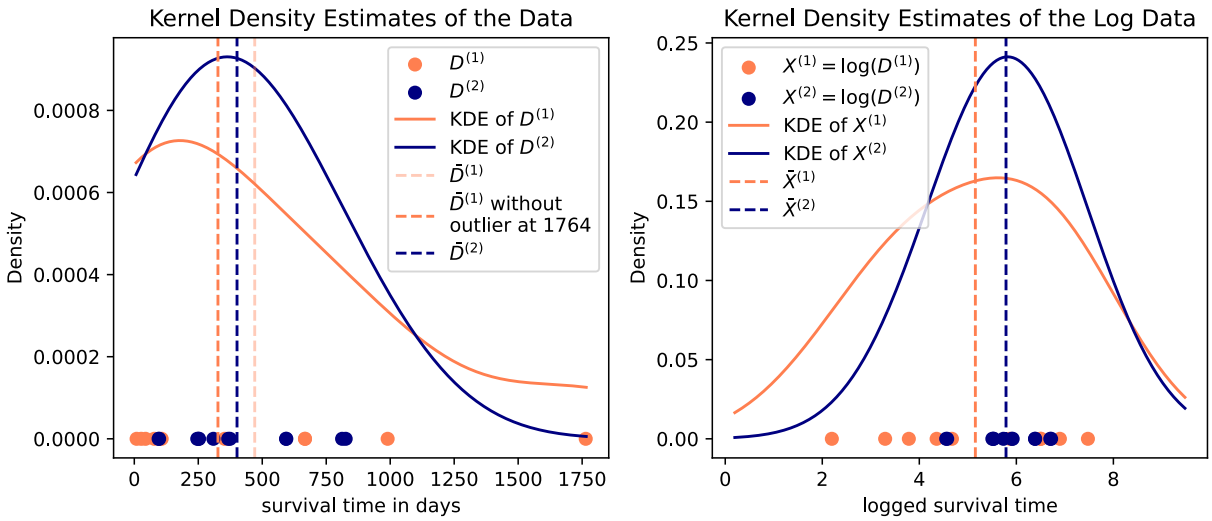


Figure 3: Kernel Density Estimates of the original and log-transformed data.

cannot reject H_0 fundamentally as of the large stray in the distributions (see figure 3 again).

Let us come to the task and give a bootstrap-corrected estimate of $\mu_2 - \mu_1$ with corresponding standard error.

The bootstrap procedure here is

1. $\forall b \in \{1, \ldots, B\}$: Generate bootstrap samples $X^{(1,b)}$ and $X^{(2,b)}$ by sampling respectively n_1 and n_2 values from $X^{(1)}$ and $X^{(2)}$ with replacement and calculate $t^{(b)} = \bar{X}^{(2,b)} - \bar{X}^{(1,b)}$ on the bootstrap samples.

2. This yields an empirical distribution of estimates $t^{(1)}, \ldots, t^{(B)}$ from which we can calculate the mean $\hat{t}^*$ and standard error $\mathrm{SE}(t^*)$.

The estimate of the bias is given by $\hat{\mathrm{bias}}(\hat{t}) = \hat{t}^* - \hat{t}$ with $\hat{t} = \bar{X}^{(2)} - \bar{X}^{(1)}$, so we get the bias corrected estimate $\bar{t} = \hat{t} - \hat{\mathrm{bias}}(\hat{t}) = 2\hat{t} - \hat{t}^*$.

Note that by the bias correction we introduce a new source of variance (the variance of the bias) which we do not account for in the standard error. Using further bootstrapping we can estimate the standard error of $\bar{t}$, but if $\hat{\mathrm{bias}}(\hat{t})$ is small compared to $\hat{\mathrm{SE}}(\hat{t})$ it is safer to use $\hat{t}$ than $\bar{t}$ (see Efron and Tibshirani, 1994, chapter 10.6, p. 138).

For the case at hand the estimates are given in table 1 and the code to obtain them in code-snippet 1. The bootstrap distribution is shown in figure 4. As of the small bias compared to the standard error we use $\hat{t}$ as our estimate of $\mu_2 - \mu_1$ with standard error $\hat{\mathrm{SE}}(\hat{t})$.

```r
bootstrap_mean_difference <- function(sample1, sample2, B = 1000) {
    n1 <- length(sample1)
    n2 <- length(sample2)
    # our statistic is a difference of means
    t <- function(a, b) {
        mean(b) - mean(a)
    }
    # point estimate
    t_hat <- t(sample1, sample2)
    # bootstrap samples
    t_bs <- replicate(B, t(sample(sample1, n1, replace = TRUE),
    ↪  sample(sample2, n2, replace = TRUE)))
    # bootstrap estimate
    t_hat_star <- mean(t_bs)
    # standard error
    se <- sd(t_bs)
    # bias
    bias <- t_hat_star - t_hat
    # bias-corrected estimate
    t_hat_star_bc <- 2 * t_hat - t_hat_star
    return(list("t_hat" = t_hat, "t_bs" = t_bs, "t_hat_star" =
    ↪  t_hat_star, "se" = se, "bias" = bias, "t_hat_star_bc" =
    ↪  t_hat_star_bc))
}

# Define the data
sample1 <- c(107, 353, 1764, 667, 990, 78, 667, 44, 9, 27)
sample2 <- c(374, 253, 812, 246, 95, 367, 251, 309, 594, 826, 593, 97)

# convert to log scale
sample1 <- log(sample1)
sample2 <- log(sample2)

res <- bootstrap_mean_difference(sample1, sample2)
```

Code-Snippet 1: Bootstrap estimates of $\mu_2 - \mu_1$ for the samples $X^{(1)}$ and $X^{(2)}$ for $B = 1000$.

point estimate $\hat{t}$	$\hat{\mathrm{SE}}(\hat{t})$	bootstrap estimate $\hat{t}^*$	$\hat{\mathrm{bias}}(\hat{t})$	bias-corrected estimate $\bar{t}$
≈ 0.6318	≈ 0.574	≈ 0.6355	≈ 0.0037	≈ 0.6281

Table 1: Estimates regarding $\mu_2 - \mu_1$ for the samples $X^{(1)}$ and $X^{(2)}$ for $B = 1000$.

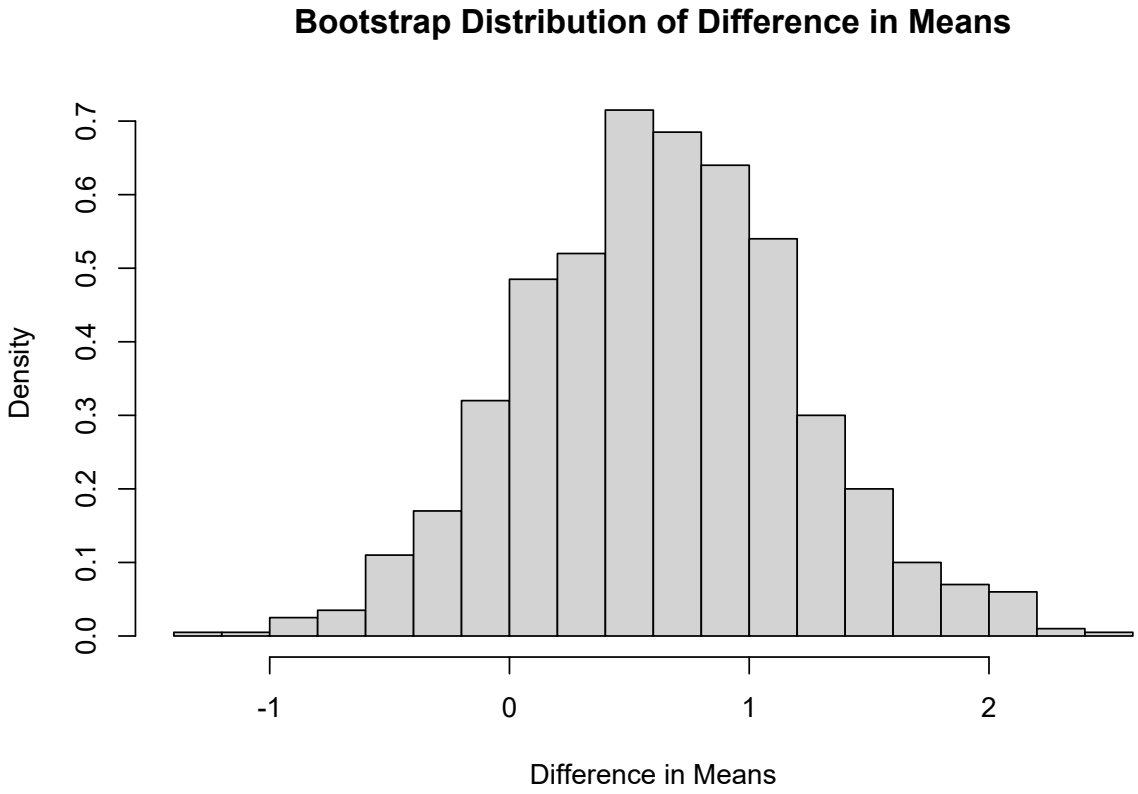


Figure 4: Bootstrap distribution of the difference in means of the bootstrap samples.

1.3 Bootstrap Confidence Intervals

Note in general that bootstrap distributions tend to be narrow on average, so the bootstrap confidence intervals often under-cover (Hesterberg, 2015).

1.3.1 Standard Normal Bootstrap Confidence Interval

If $\hat{\theta}$ is a sample mean, then for large sample sizes, we can apply the central limit theorem

$$\frac{\hat{\theta} - \theta}{\mathrm{SE}(\hat{\theta})} \underset{a}{\sim} N(0, 1) \tag{4}$$

so based on the bootstrap estimate of the standard error $\hat{\mathrm{SE}}(\hat{\theta})$ we can calculate a $1 - \alpha$-confidence interval for θ as

$$\hat{\theta} \pm z_{1-\alpha/2}\hat{\mathrm{SE}}(\hat{\theta}), \quad z_{1-\alpha/2} \text{ is the } 1-\alpha/2 \text{ quantile of } N(0,1) \tag{5}$$

assuming that $\hat{\theta}$ is approximately normally distributed, $\hat{\theta}$ is unbiased and $\hat{\mathrm{SE}}(\hat{\theta})$ is a good estimate of $\mathrm{SE}(\hat{\theta})$.

1.3.2 Percentile Bootstrap Confidence Interval

Here, we use quantiles based on the empirical distribution of the bootstrap samples, so a $1-\alpha$-confidence interval for θ is given by

$$\begin{aligned} &\left[\hat{\theta}^*_{\alpha/2}, \hat{\theta}^*_{1-\alpha/2}\right], \quad \hat{\theta}^*_{\alpha/2} \text{ is the } \alpha/2 \text{ quantile of } \hat{\theta}^*, \\ &\text{with } \hat{\theta}^*_{1-\alpha/2}, \hat{\theta}^*_{1-\alpha/2} \text{ being the quantiles of the empirical distribution } \left\{\hat{\theta}^{(b)}\right\}_{b=1,\dots,B} \end{aligned} \tag{6}$$

1.3.3 Basic Bootstrap Confidence Interval

Akin to the bias-corrected estimate $\bar{\theta}$, we get a basic bootstrap confidence interval for θ as

$$\begin{aligned} &\left[2\hat{\theta} - \hat{\theta}^*_{1-\alpha/2}, 2\hat{\theta} - \hat{\theta}^*_{\alpha/2}\right], \quad \hat{\theta}^*_{\alpha/2} \text{ is the } \alpha/2 \text{ quantile of } \hat{\theta}^*, \\ &\text{with } \hat{\theta}^*_{1-\alpha/2}, \hat{\theta}^*_{1-\alpha/2} \text{ being the quantiles of the empirical distribution } \left\{\hat{\theta}^{(b)}\right\}_{b=1,\dots,B} \end{aligned} \tag{7}$$

1.3.4 Bootstrap Confidence Intervals for Task 1b

The $\alpha = 0.05$-confidence intervals for $\mu_2 - \mu_1$ are given in table 2 and the code to obtain them in code-snippet 2 and 3. As expected based on the large stray in the distributions (see figure 3 again), the confidence intervals for the difference in the means are large, so our estimate of $\mu_2 - \mu_1$ might not reflect the true value very well (even the sign might be wrong).

standard normal	percentile	basic
$[-0.49, 1.76]$	$[-0.49, 1.87]$	$[-0.61, 1.76]$

Table 2: Bootstrap confidence intervals for $\mu_2 - \mu_1$ for the samples $X^{(1)}$ and $X^{(2)}$ for $B = 1000$.

```r
standard_normal_interval <- function(SE_hat, t_hat, alpha = 0.05) {
    # SE_hat: standard error estimate
    # t_hat: point estimate
    # alpha: significance level
    # returns: confidence interval

    # critical value
    z <- qnorm(1 - alpha / 2)
    # lower bound
    lower <- t_hat - z * SE_hat
    # upper bound
    upper <- t_hat + z * SE_hat
    return(c(lower, upper))
}

percentile_boots_interval <- function(t_bs, alpha = 0.05) {
    # t_bs: bootstrap samples
    # alpha: significance level
    # returns: confidence interval

    # critical values
    lower <- quantile(t_bs, alpha / 2)
    upper <- quantile(t_bs, 1 - alpha / 2)
    return(c(lower, upper))
}

basic_boots_interval <- function(t_bs, t_hat, alpha = 0.05) {
    # t_bs: bootstrap samples
    # t_hat: point estimate
    # alpha: significance level
    # returns: confidence interval

    # critical values
    lower <- 2 * t_hat - quantile(t_bs, 1 - alpha / 2)
    upper <- 2 * t_hat - quantile(t_bs, alpha / 2)
    return(c(lower, upper))
}
```

Code-Snippet 2: General functions for calculating bootstrap confidence intervals.

```r
# Define the data
sample1 <- c(107, 353, 1764, 667, 990, 78, 667, 44, 9, 27)
sample2 <- c(374, 253, 812, 246, 95, 367, 251, 309, 594, 826, 593, 97)

# convert to log scale
sample1 <- log(sample1)
sample2 <- log(sample2)

res <- bootstrap_mean_difference(sample1, sample2)

# print and calculate confidence intervals
print(paste("standard normal interval:", standard_normal_interval(res$se, res$t_hat)))
print(paste("percentile boots interval:", percentile_boots_interval(res$t_bs)))
print(paste("basic boots interval:", basic_boots_interval(res$t_bs, res$t_hat)))
```

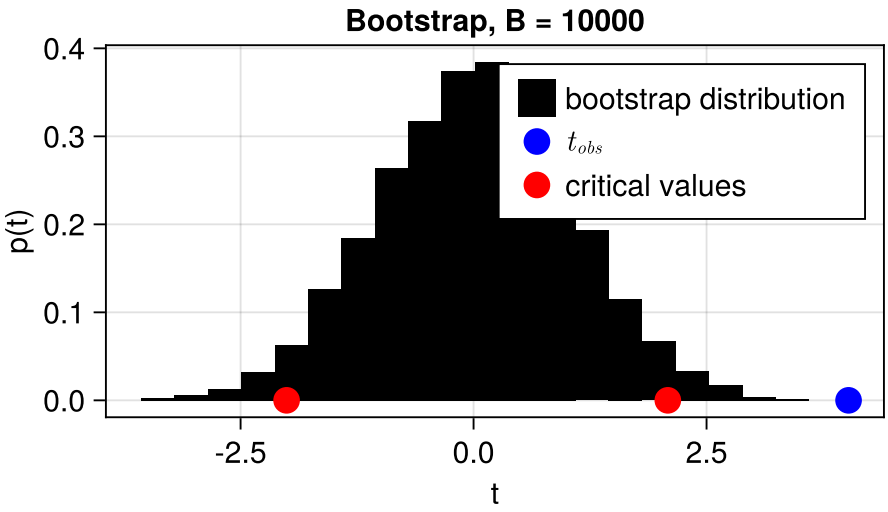
Code-Snippet 3: Bootstrap confidence intervals for $\mu_2 - \mu_1$ for the samples $X^{(1)}$ and $X^{(2)}$ for $B = 1000$.

1.4 Bootstrap Hypothesis Testing

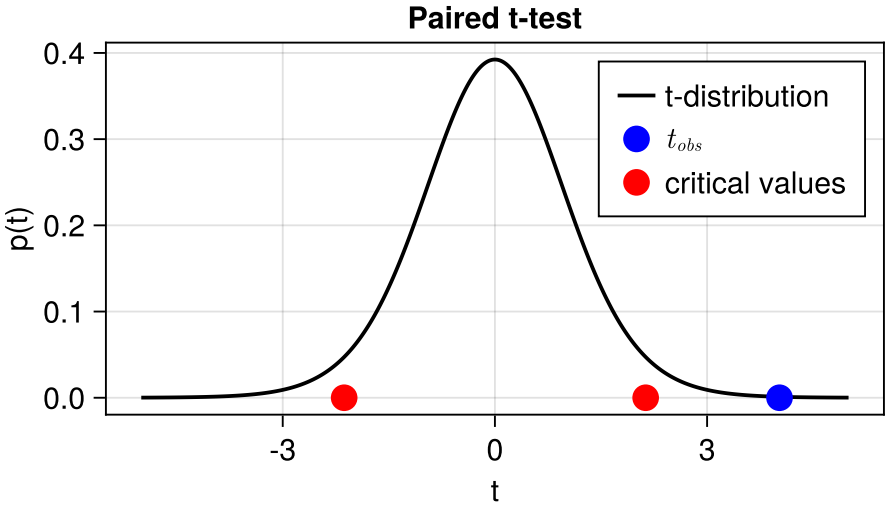
Consider we want to test a null hypothesis H_0 and we have a test statistic T where observed values of T under H_0 can be calculated for a sample as $t_{\text{obs}}(X_{\text{sample}})$. In a setting where we make assumptions on the distributions of the test statistic we would test based on where our observed value falls in the distribution of the test statistic under H_0.

In the bootstrap setting we can test based on an empirical distribution based on resampling (see figure 5a), adapted to follow H_0 if this is not the case by default. Consider for instance for a sample $X_{\text{sample}} = (x_1, \ldots, x_n)$ we want to test the null hypothesis $H_0 : \mu = \mu_0$ vs $H_1 : \mu \neq \mu_0$ (one-sample test). We then base our bootstrap samples on $z_i = x_i - \bar{X}_{\text{sample}} + \mu_0$ and use the test statistic $t^{(b)} = t^*\left(Z^{(b)}\right) = \frac{\bar{Z}^{(b)} - \mu_0}{s_Z^{(b)}/\sqrt{n}}$ where $Z^{(b)} = (z_1^{(b)}, \ldots, z_n^{(b)})$ and $s_Z^{(b)} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(z_i^{(b)} - \bar{Z}^{(b)}\right)^2}$ and $t_{\text{obs}} = t^*(X_{\text{sample}})$.

Here we assume that as the mean varies, the distributions are just translated versions of each other (translation family) where if our x_i are lifetimes it might make sense to use logged lifetimes as they are more likely to satisfy a translation or normal family assumption (Efron and Tibshirani, 1994, chapter 16.4).



(a) (Two-sided) Bootstrap Hypothesis Testing based on empirical distribution from bootstrap samples.



(b) (Two-sided) Standard Hypothesis test with distributional assumption.

Figure 5: Bootstrap vs Distributional Assumption Hypothesis Testing.

Practically, we calculate estimates of the p values (where for $p < \alpha$ we reject H_0) as

$$\begin{aligned} \hat{p}^*_{\text{right}} &= \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}_{\left[t^{(b)} \geq t_{\text{obs}}\right]} \\ \hat{p}^*_{\text{left}} &= \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}_{\left[t^{(b)} \leq t_{\text{obs}}\right]} \\ \hat{p}^*_{\text{two-sided}} &= \frac{1}{B}\left[\min\left\{\sum_{b=1}^{B} \mathbb{1}_{\left[t^{(b)} \geq t_{\text{obs}}\right]}, \sum_{b=1}^{B} \mathbb{1}_{\left[t^{(b)} \leq t_{\text{obs}}\right]}\right\}\right] \end{aligned} \tag{8}$$

1.4.1 Bootstrap Hypothesis Test on the Difference of Means I: Theory

Consider we have two samples $X^{(1)} = (x_1^{(1)}, \ldots, x_n^{(1)})$ and $X^{(2)} = (x_1^{(2)}, \ldots, x_n^{(2)})$. And we want to test the null-hypothesis

$$H_0 : \mu^{(2)} - \mu^{(1)} \leq c \quad \text{vs} \quad H_1 : \mu^{(2)} - \mu^{(1)} > c \tag{9}$$

where $c \in \mathbb{R}$. Assume the true means are $\mu^{(1)}$ and $\mu^{(2)}$. We can generate a bootstrap distribution under H_0 by sampling from

$$\begin{gathered} z_i^{(1)} = x_i^{(1)} - \bar{X}^{(1)}_{\text{sample}} - \frac{c}{2} \quad \text{and} \quad z_i^{(2)} = x_i^{(2)} - \bar{X}^{(2)}_{\text{sample}} + \frac{c}{2} \\ Z^{(1)} = (z_1^{(1)}, \ldots, z_n^{(1)}) \quad \text{and} \quad Z^{(2)} = (z_1^{(2)}, \ldots, z_n^{(2)}) \end{gathered} \tag{10}$$

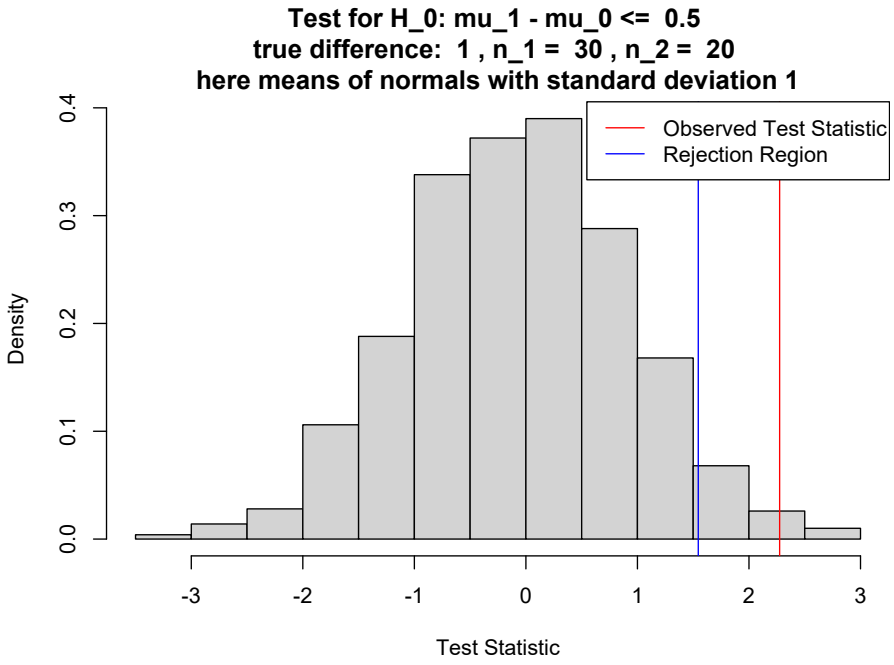
and using the test statistic

$$\begin{aligned} t_{\text{obs}} &= \frac{\bar{X}^{(2)}_{\text{sample}} - \bar{X}^{(1)}_{\text{sample}} - c}{S_p\sqrt{\frac{1}{n^{(1)}} + \frac{1}{n^{(2)}}}} \\ S_p &= \sqrt{\frac{(n^{(1)} - 1)\left(S^{(1)}\right)^2 + (n^{(2)} - 1)\left(S^{(2)}\right)^2}{n^{(1)} + n^{(2)} - 2}} \end{aligned} \tag{11}$$

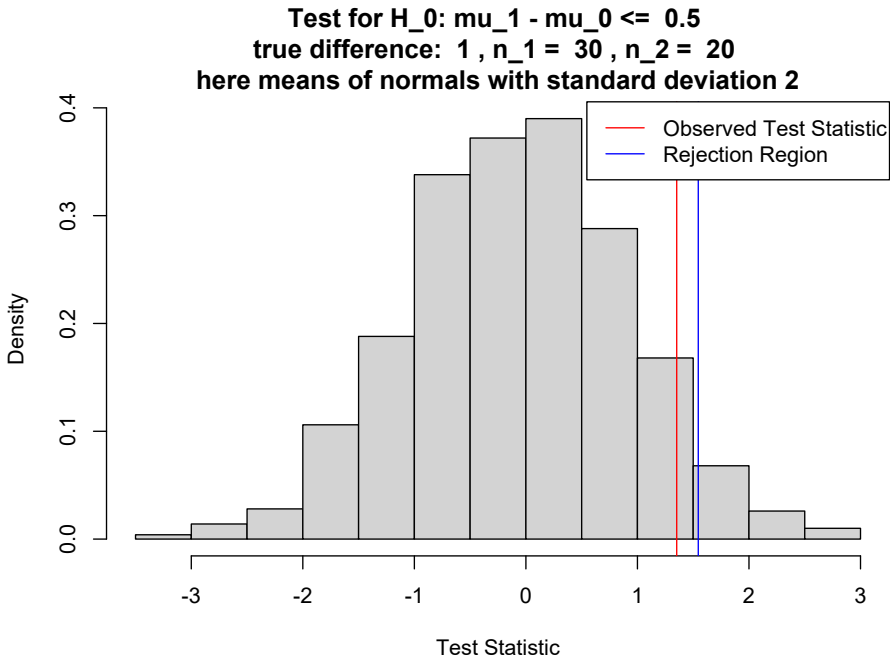
with analogous calculations of $t^{(b)}$ on the bootstrap samples (the basic reasoning behind the denominator is that the scale on which means can be compared is the standard deviation, without which a statement like the means are 1 apart is meaningless (very different statement for standard deviations of the order of 10 or 10^{-2})).

1.4.2 Bootstrap Hypothesis Test on the Difference of Means I: Example Application

To test the described method, we use samples from two normal distributions where we specify the means and standard deviations. Based on our results, the test seems appropriate. Two examples are shown in figure 6. Note that as in hypothesis testing we are very cautious and go the devils-advocate route, we only reject H_0 if we are very sure so in the test with large standard deviation (figure 6b) we wrongly accept H_0, which is still a reasonable choice given a standard deviation of 2 in the distributions and only a difference in means of 1.



(a) Correctly Rejecting H_0.



(b) Wrongly Accepting H_0.

Figure 6: Bootstrap Hypothesis Test on the Difference of Means

1.4.3 Bootstrap Hypothesis Test on the Difference of Means III: Task 1c

We choose a $\alpha = 0.05$-significance level and test $H_0 : \mu^{(2)} - \mu^{(1)} \leq 0.25$ vs $H_1 : \mu^{(2)} - \mu^{(1)} > 0.25$.

Let us apply the previously introduced and tested methods to the samples from the exercise. Based on the samples we get a p-value of $\hat{p}^*_{\text{right}} = 0.254$, so H_0 has to be accepted. As previously discussed, it would be unreasonable to reject H_0 as of the large stray in the distributions compared to the difference in means (see figure 3 again). The test distribution is shown in figure 7 and the code to obtain the results in code-snippet 4.

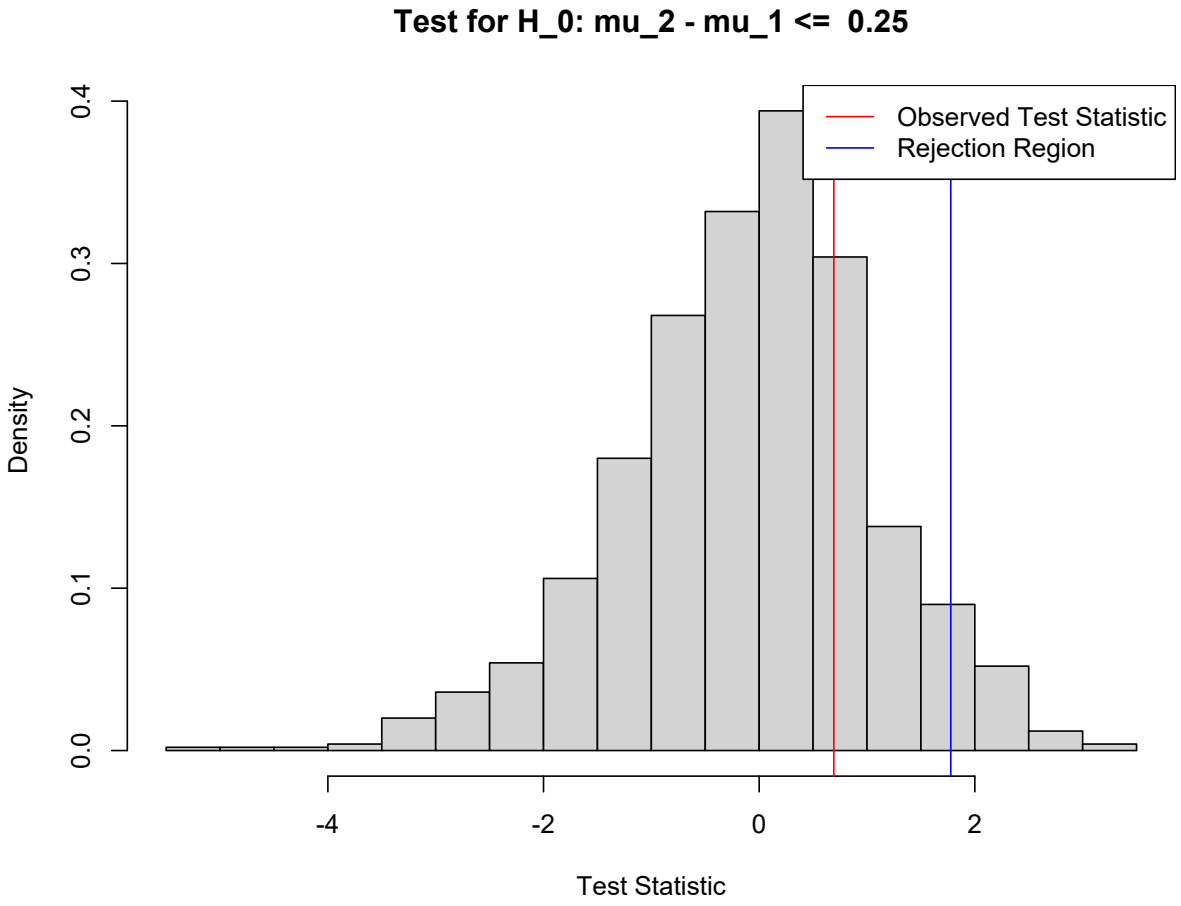


Figure 7: Bootstrap distribution of the test statistic $t^{(b)}$ for the samples $X^{(1)}$ and $X^{(2)}$ for $B = 1000$.

```r
bootstrap_mean_difference_greater_than_test <- function(sample1, sample2, diff, B = 1000) {
    # sample1: vector of observations from sample 1
    # sample2: vector of observations from sample 2
    # diff: H_0: mu_1 - mu_2 <= diff
    # B: number of bootstrap samples to use
    # returns: evals of statistic on bootstrap samples, p-value for test

    n1 <- length(sample1)
    n2 <- length(sample2)

    # test statistic used
    t <- function(a, b) {
        S <- sqrt(((n1 - 1) * var(a) + (n2 - 1) * var(b)) / (n1 + n2 - 2))
        Tstat <- (mean(a) - mean(b) - diff) / (S * sqrt(1/n1 + 1/n2))
        return(Tstat)
    }

    # observed test statistic
    t_obs <- t(sample1, sample2)

    # adapted samples for generating a bootstrap distribution under H_0
    # note that we test for the extreme, so mu_1 - mu_2 = diff
    # and then make a right-sided test

    s1adapt <- sample1 - mean(sample1) + diff / 2
    s2adapt <- sample2 - mean(sample2) - diff / 2

    # generate bootstrap samples
    tbs <- replicate(B, t(sample(s1adapt, n1, replace = TRUE), sample(s2adapt, n2, replace = TRUE)))

    # compute p-value
    pval <- mean(tbs >= t_obs)

    return(list("t_obs" = t_obs, "tbs" = tbs, "pval" = pval))
}

# apply to the given samples, switch the order of the arguments, as
# I programmed the function before noticing, mu_1 - mu_2 is not the
# test the teacher probably had in mind
res <- bootstrap_mean_difference_greater_than_test(sample2, sample1, 0.25, 1000)
```

Code-Snippet 4: Bootstrap hypothesis test for $H_0 : \mu^{(2)} - \mu^{(1)} \leq 0.25$ vs $H_1 : \mu^{(2)} - \mu^{(1)} > 0.25$.

1.5 Bootstrapping on Linear Regression Coefficients (task 2)

In this task, we regress on the waiting time between eruptions of a geyser given the length of the subsequent eruption.

1.5.1 Linear Fit on the Geysir Data (task 2a)

The code for the linear fit is given in code-snippet 5 and the fit is shown in figure 8. We get $\beta_0 = 99 \pm 2$ and $\beta_1 = -7.8 \pm 0.6$ (rounding the errors up and the coefficients to the significant digits). Note that based on the plot we can see that using a linear model only makes limited sense.

```r
# Load the Data
library(MASS)
data(geyser)
# Fit the Model
model <- lm(waiting ~ duration, data = geyser)
# Summarize the Model
print(summary(model))
# Plot the Model
plot(geyser$duration, geyser$waiting, xlab = "Duration", ylab =
    ↪ "Waiting", main = "Waiting vs. Duration Linear Regression")
abline(model, col = "red")
```

Code-Snippet 5: Linear fit on the geysir data.

Figure 8: Linear fit on the geysir data.

1.6 Bootstrap estimates by resampling cases (task 2b)

By resampling data points and fitting multiple models, we can bootstrap-estimate the standard error of the coefficients. We just employ the previously introduced bootstrap methods, caveats mentioned there still apply.

The code for the bootstrap estimates is given in code-snippet 6. We get $\hat{\beta}_0^* \approx 99.3$, $\hat{\mathrm{SE}}\left(\hat{\beta}_0\right) \approx 1.4$, $\hat{\beta}_1^* \approx -7.8$ and $\hat{\mathrm{SE}}\left(\hat{\beta}_1\right) \approx 0.5$.

```r
bootstrap_lin_regr <- function(x, y, B = 1000) {
    n <- length(x)
    # point-estimate
    mod <- lm(y ~ x)
    b0_hat <- mod$coefficients[1]
    b1_hat <- mod$coefficients[2]
    # bootstrap
    b0_bs <- rep(NA, B)
    b1_bs <- rep(NA, B)
    # bootstrap samples
    for (i in 1:B) {
        idx <- sample(1:n, n, replace = TRUE)
        mod_b <- lm(y[idx] ~ x[idx])
        b0_bs[i] <- mod_b$coefficients[1]
        b1_bs[i] <- mod_b$coefficients[2]
    }
    # bootstrap estimates
    b0_hat_star <- mean(b0_bs)
    b1_hat_star <- mean(b1_bs)
    # standard errors
    se_b0 <- sd(b0_bs)
    se_b1 <- sd(b1_bs)
    # nicely format the output
    return(list("b0_hat" = b0_hat, "b1_hat" = b1_hat, "b0_bs" =
    ↪  b0_bs, "b1_bs" = b1_bs, "b0_hat_star" = b0_hat_star,
    ↪  "b1_hat_star" = b1_hat_star, "se_b0" = se_b0, "se_b1" =
    ↪  se_b1))
}

est <- bootstrap_lin_regr(geyser$duration, geyser$waiting, B = 1000)
print(est)
# calculate the confidence intervals
# function to do so previously defined
alpha <- 0.04
b0_interval <- standard_normal_interval(est$se_b0, est$b0_hat, alpha
↪  = alpha)
b1_interval <- standard_normal_interval(est$se_b1, est$b1_hat, alpha
↪  = alpha)
```

Code-Snippet 6: Bootstrap estimates on the geysir data.

1.7 Standard normal bootstrap confidence intervals (task 2c)

Based on the code from code-snippet 2 we get the normal bootstrap confidence interval for $\alpha = 0.04$ as $[96.45, 102.16]$ for β_0 and $[-8.73, -6.87]$ for β_1.

1.8 Testing on $H_0 : \beta_0 = 0$ or rather $H_0 : \beta_0 = c$ (task 2d)

We are supposed to test $H_0 : \beta_0 = 0$ vs $H_1 : \beta_0 \neq 0$. However, given the small standard error and the large value of β_0, p-values will be pretty much zero, so we will always reject H_0.

Let us therefore more generally test $H_0 : \beta_0 = c$ vs $H_1 : \beta_0 \neq c$ for $c \in \mathbb{R}$ using the test statistic

$$\begin{aligned} T &= \sqrt{\frac{nS_{xx}}{\sum_{i=1}^{N}}} \frac{\hat{\beta}_0 - c}{\hat{\sigma}} \sim t_{n-2} \\ S_{xx} &= \sum_{i=1}^{n} (x_i - \bar{x})^2, \quad \hat{\sigma}^2 = \frac{\sum_{i=1}^{N} (Y_i - \bar{Y})^2 - \hat{\beta}_1^2 S_{xx}}{n-2} \end{aligned} \tag{12}$$

Test using the given test statistic (task 2d.i)

The code for doing the two-sided test is given in code-snippet 7. We choose an $\alpha = 0.05$-significance level and get report p-values for different values of c in table 3.

c	p-value	reject $H_0 : \beta_0 = c$ at $\alpha = 0.05$
0	0	yes
96	9.1%	no
98	50.4%	no
100	72.5%	no
102	17.0%	no
104	1.7%	yes

Table 3: Hypothesis test for $H_0 : \beta_0 = c$ vs $H_1 : \beta_0 \neq c$ for $c \in \mathbb{R}$.

```r
beta_0_statistic <- function(x, y, c, beta1, beta0) {
    n <- length(x)
    Sxx <- sum((x - mean(x))^2)
    sigma <- sqrt((sum((y - mean(y))^2) - beta1^2 * Sxx) / (n - 2))
    T <- sqrt((n * Sxx) / (sum(x^2))) * (beta0 - c) / sigma
    return(T)
}

beta_0_t_test <- function(x, y, c, B = 1000) {
    # calculate point estimates
    mod <- lm(y ~ x)
    beta0 <- mod$coefficients[1]
    beta1 <- mod$coefficients[2]
    # calculate the test statistic
    t_obs <- beta_0_statistic(x, y, c, beta1, beta0)
    n <- length(x)
    # get p-value from t-distribution with n-2 degrees of freedom
    p_value <- 2 * pt(-abs(t_obs), df = n - 2)
    # nicely format the output
    return(list("t_obs" = t_obs, "p_value" = p_value))
}

# do the t-test
res <- beta_0_t_test(geyser$duration, geyser$waiting, c = 104, B =
↪ 1000)
print(res)
```

Code-Snippet 7: Hypothesis test for $H_0 : \beta_0 = c$ vs $H_1 : \beta_0 \neq c$ for $c \in \mathbb{R}$.

Test using the bootstrap distribution (task 2d.ii)

Note that the estimate of $\hat{\beta}_1$ does not change, when we mean-shift Y. We can therefore enforce H_0 on the data, by

$$c = \bar{Y}^* - \hat{\beta}_1 \bar{x} \rightarrow \bar{Y}^* = \hat{\beta}_1 \bar{x} + c \rightarrow y_i^* = y_i - \bar{Y} + \bar{Y}^* = y_i - \bar{Y} + \hat{\beta}_1 \bar{x} + c \tag{13}$$

The code for doing the two-sided test is given in code-snippet 8. We choose an $\alpha = 0.05$-significance level and get report p-values for different values of c in table 4.

As expected the bootstrap distribution is more narrow than the t-distribution, so the bootstrap test rejects H_0 more often than the t-test.

```r
beta_0_statistic <- function(x, y, c, beta1, beta0) {
    n <- length(x)
    Sxx <- sum((x - mean(x))^2)
    sigma <- sqrt((sum((y - mean(y))^2) - beta1^2 * Sxx) / (n - 2))
    T <- sqrt((n * Sxx) / (sum(x^2))) * (beta0 - c) / sigma
    return(T)
}

beta_0_bootstrap_test <- function(x, y, c, B = 1000) {
    # calculate point estimates
    mod <- lm(y ~ x)
    beta0 <- mod$coefficients[1]
    beta1 <- mod$coefficients[2]
    # calculate the test statistic
    t_obs <- beta_0_statistic(x, y, c, beta1, beta0)
    # impose the null hypothesis on the data
    y <- y - mean(y) + beta1 * mean(x) + c
    # check that we correctly imposed the null hypothesis
    beta0new <- lm(y ~ x)$coefficients[1]
    if (abs(beta0new - c) > 1e-10) {
        stop("Error: Null hypothesis not imposed correctly")
    }
    # bootstrap
    t_bs <- rep(0, B)
    for (i in 1:B) {
        idx <- sample(1:length(x), length(x), replace = TRUE)
        # for H0 we assume that beta1 is known
        # if we would estimate beta1 from the data
        # the null hypothesis would not be strictly imposed
        t_bs[i] <- beta_0_statistic(x[idx], y[idx], c, beta1,
        ↪  lm(y[idx] ~ x[idx])$coefficients[1])
    }
    # calculate the p-value (two-sided)
    p_left <- mean(t_bs <= t_obs)
    p_right <- mean(t_bs >= t_obs)
    p_value <- min(p_left, p_right) * 2
    # nicely format the output
    return(list("t_obs" = t_obs, "t_bs" = t_bs, "p_value" =
    ↪  p_value))
}

# do the bootstrap test
res <- beta_0_bootstrap_test(geyser$duration, geyser$waiting, c =
↪  99, B = 1000)
```

Code-Snippet 8: Bootstrap hypothesis test for $H_0 : \beta_0 = c$ vs $H_1 : \beta_0 \neq c$ for $c \in \mathbb{R}$.

c	p-value	reject $H_0 : \beta_0 = c$ at $\alpha = 0.05$
0	0	yes
96	1.4%	yes
98	35%	no
100	65.8%	no
102	8.4%	no
104	0.4%	yes

Table 4: Bootstrap hypothesis test for $H_0 : \beta_0 = c$ vs $H_1 : \beta_0 \neq c$ for $c \in \mathbb{R}$.

2 Jackknife and Further Bootstrap Applications

Consider the following sample from $X \sim \mathcal{N}(\mu, \sigma^2)$:

$$X_{\text{sample}} = \{2.8, 3, 2.3, 1.3, 1.4, 3.2, 2.9, 1.1, 2.2, 3.5, 1.8, 3.5, 2.5, 2, 2.8\} \tag{14}$$

2.1 Bias correction and standard error estimation using Bootstrap and Jackknife (task 3a, 3b)

Consider the estimator of the mean

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{15}$$

Using the Bootstrap method (as introduced in section 1), we get the results shown in table 5 using the code in code-snippet 9 (task b). As the calculated bias is smaller than the standard error, it makes no sense to correct by it. This is also the right conclusion, as the mean is an unbiased estimator.

point estimate $\hat{\mu}$	$\hat{\text{SE}}(\hat{\mu})$	bootstrap estimate $\hat{\mu}^*$	$\hat{\text{bias}}(\hat{\mu})$	bias-corrected estimate $\bar{\mu}$
≈ 2.420	≈ 2.415	≈ 0.19	≈ -0.0053	≈ 2.4253

Table 5: Bootstrap estimates regarding μ for X_{sample} for $B = 1000$.

Different from bootstrapping, in the Jackknife method, we generate n subsamples by removing one observation at a time,

$$X_{\text{sample}(-i)} = \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\} \tag{16}$$

which is computationally less expensive than bootstrapping.

Jackknife-Intuition: Note, that the idea of the Jackknife is a bit different. In Jackknife,

```r
bootstrap <- function(X, T, B = 1000) {
# Sample estimate
sample_estimate <- T(X)
    n <- length(X)

    # Generate bootstrap samples and apply function T
    bootstrap_samples <- replicate(B, T(sample(X, size = n, replace
    ↪  = TRUE)))

    # Sort the bootstrap samples
    stat <- sort(bootstrap_samples)

    # The estimator is the mean
    bootstrap_estimate <- mean(stat)

    # The standard error is the standard deviation of the statistic
    sd <- sd(stat)

    # The estimated bias is the difference between the sample
    ↪  estimate
    # and the bootstrap estimate
    bias <- bootstrap_estimate - sample_estimate

    # Bias-corrected estimate
    corrected_estimate <- sample_estimate - bias

    # Return the estimate, standard error, and bias
    return(list(sample_estimate = sample_estimate,
                bootstrap_estimate = bootstrap_estimate,
                sd = sd,
                bias = bias,
                corrected_estimate = corrected_estimate))
}
X <- c(2.8, 3, 2.3, 1.3, 1.4, 3.2, 2.9, 1.1, 2.2, 3.5, 1.8, 3.5,
↪  2.5, 2, 2.8)
# Bootstrap estimate of the mean
res <- bootstrap(X, mean)
print(res)
```

Code-Snippet 9: Code for Bootstrap estimates regarding X_{sample}.

we ask ourselves, what happens, when one subsample to the other differs by having one observation each the other does not have while sharing the rest (so similar subsamples). For the estimate of the bias and variance of the estimator we then have to scale this up to "what if we had totally different samples". This leads to the Jackknife estimates

$$\hat{t}_{\text{jack}} = \frac{1}{n}\sum_{i=1}^{n}\hat{t}_{(-i)}, \quad \hat{\text{bias}}_{\text{jack}} = (n-1)\left(\hat{t}_{\text{jack}} - \hat{t}\right), \quad \hat{\text{SE}}_{\text{jack}} = \sqrt{\frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{t}_{(-i)} - \hat{t}_{\text{jack}}\right)^2} \tag{17}$$

Note that for the case of the mean

$$\begin{aligned}
\hat{\mu}_{\text{jack}} &= \frac{1}{n}\sum_{i=1}^{n}\hat{\mu}_{(-i)} \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n-1}\sum_{j=1,j\neq i}^{n}x_j \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{n-1}\left(\sum_{j=1}^{n}x_j - x_i\right)\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{n-1}\left(n\hat{\mu} - x_i\right)\right) \\
&= \frac{n}{n-1}\hat{\mu} - \frac{1}{(n-1)}\hat{\mu} \\
&= \hat{\mu}
\end{aligned} \tag{18}$$

so the Jackknife estimate of the mean is actually just the sample mean itself (at least from my calculations), so we do not expect any bias so also no bias correction. The numerical results are shown in table 6 using the code in code-snippet 10 (task a). As expected, the Jackknife estimate of the mean is the sample mean itself, so we reflect, that there is no bias and nothing to correct.

point estimate $\hat{\mu}$	$\hat{\text{SE}}(\hat{\mu})$	jackknife estimate $\hat{\mu}_{\text{jack}}$	$\hat{\text{bias}}(\hat{\mu})$
≈ 2.42	≈ 0.20	≈ 2.42	$= 0$

Table 6: Jackknife estimates regarding μ for X_{sample}.

```r
jackknife <- function(X, T) {
    n <- length(X)

    # Array to store jackknife samples
    jackknife_samples <- numeric(n)

    # Jackknife samples, leaving one out
    for (i in 1:n) {
        jackknife_samples[i] <- T(X[-i])
    }

    # The sample estimate
    sample_estimate <- T(X)

    # The jackknife estimate is the mean of the jackknife samples
    jackknife_estimate <- mean(jackknife_samples)

    # Standard error calculation
    jackknife_se <- sqrt((n - 1) / n * sum((jackknife_samples -
    ↪  jackknife_estimate)^2))

    # Bias estimation
    bias <- (n - 1) * (jackknife_estimate - sample_estimate)

    # Bias-corrected estimate
    corrected_estimate <- sample_estimate - bias

    # Return the estimate, standard error, and bias
    return(list(sample_estimate = sample_estimate,
                jackknife_estimate = jackknife_estimate,
                se = jackknife_se,
                bias = bias,
                corrected_estimate = corrected_estimate))
}

# sample
X <- c(2.8, 3, 2.3, 1.3, 1.4, 3.2, 2.9, 1.1, 2.2, 3.5, 1.8, 3.5,
↪  2.5, 2, 2.8)

# Jackknife estimate of the mean
res <- jackknife(X, mean)
print(res)
```

Code-Snippet 10: Code for Jackknife estimates regarding X_{sample}.

2.2 Non-parametric bootstrap method to test the hypothesis H_0 : $X \sim \mathcal{N}(2,1)$ (task 3c)

Let us first get a feeling for how realistic H_0 might be, based on the plot in figure 9. The centering seems to be somewhat off, but just from looking at the data, I could not straight reject H_0. The best we could do, I would say, is to give the likelihood, that this data comes from $\mathcal{N}(2,1)$, but let us do a fancy test with Kolmogorov-Smirnov statistic instead[3].

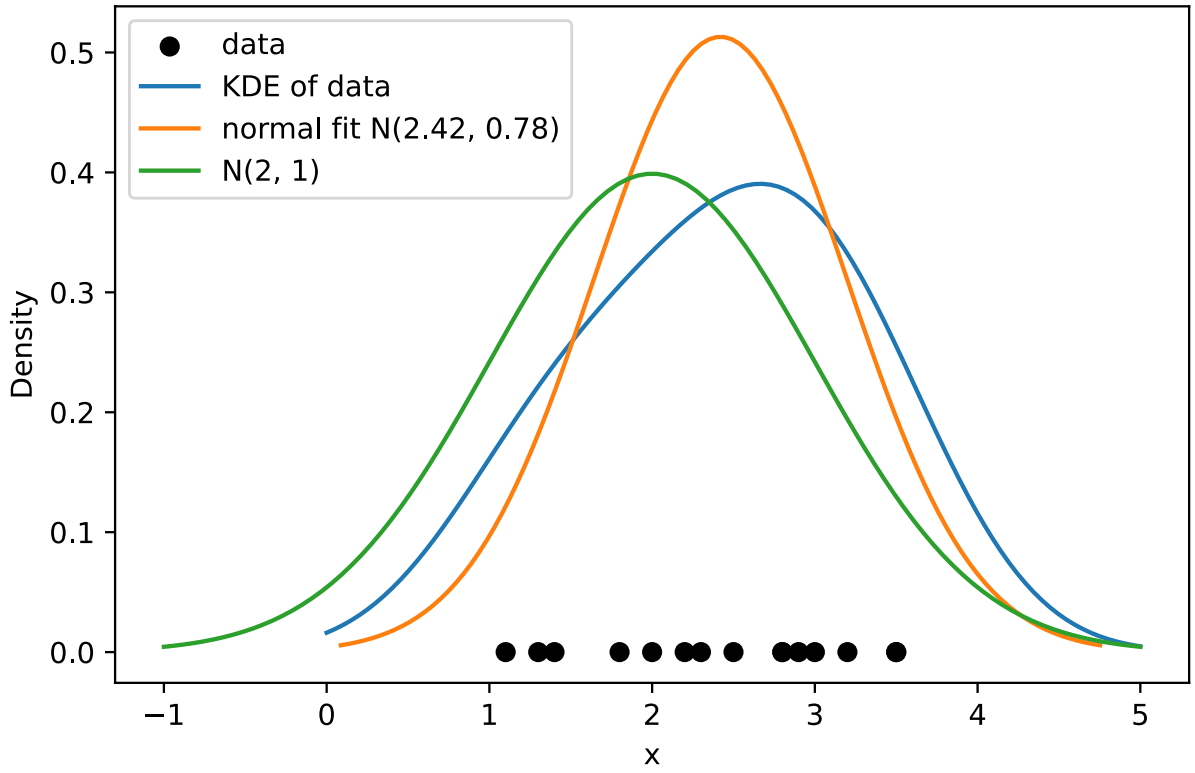


Figure 9: Kernel density estimate of X_{sample} with normal fit.

The Kolmogorov-Smirnov statistic is defined as

$$D = \max_{i=1,\dots,n} \left\{ \frac{i}{n} - F_0(x_{(i)}), F_0(x_{(i)}) - \frac{i-1}{n} \right\}, \quad F_0(x) = P_0(X \leq x) \text{ under } H_0 \tag{19}$$

As in the task it is given, that the data comes from some normal distribution, we can impose H_0 onto the empirical distribution by sampling from

$$x_i^* = \frac{x_i - \bar{X}}{\sigma_X} \cdot \sigma_0 + \mu_0, \quad \sigma_0 = 1, \quad \mu_0 = 2 \tag{20}$$

[3]Testing has the very inherent problem that we try to crunch the information of data into a single yes/no decision with a p-value, which one who lacks integrity might p-hack on.

Using the code in code-snippet 11, we get a p-value of 74.2%, so we cannot reject H_0 at a significance level of 5%. Just for comparison, the p-value for the test for $H_0 : X \sim \mathcal{N}(1.5, 1)$ is 3.4%, so we can reject H_0 at a significance level of 5%.

```r
# to have pnorm available
library(stats)
# Kolmogorov-Smirnov statistic
ks_stat <- function(sample, F0) {
    n <- length(sample)
    sorted_sample <- sort(sample)
    # initialize vector of KS values
    ks_vals <- rep(NA, n)
    for (i in 1:n) {
        ks_vals[i] <- max(i/n - F0(sorted_sample[i]),
        ↪  F0(sorted_sample[i]) - (i-1)/n)
    }
    # return the maximum KS value
    return(max(ks_vals))
}
# Bootstrap test
bootstrap_ks_test <- function(X, F0, sd0, mu0, B = 1000) {
    n <- length(X)
    original_ks <- ks_stat(X, F0)
    bootstrap_ks <- rep(NA, n)
    # impose H_0 on the data, assuming
    # X is normally distributed
    X <- (X - mean(X)) / sd(X) * sd0 + mu0
    for (b in 1:B) {
        # Generate bootstrap sample and compute KS stat
        sample <- sample(X, size = n, replace = TRUE)
        bootstrap_ks[b] <- ks_stat(sample, F0)
    }
    # Compute p-value
    p_left <- mean(bootstrap_ks <= original_ks)
    p_right <- mean(bootstrap_ks >= original_ks)
    p_value <- min(p_left, p_right) * 2
    list(original_ks = original_ks, p_value = p_value)
}
# Sample data X
X <- c(2.8, 3, 2.3, 1.3, 1.4, 3.2, 2.9, 1.1, 2.2, 3.5, 1.8, 3.5,
↪  2.5, 2, 2.8)
# Null hypothesis parameters
mu0 <- 2
sd0 <- 1
# Null hypothesis: X ~ N(mu0, sd0)
F0 <- function(x) pnorm(x, mean = mu0, sd = sd0)
# Perform the bootstrap KS test
result <- bootstrap_ks_test(X, F0, sd0, mu0, B = 1000)
print(result)
```

Code-Snippet 11: Code for Kolmogorov-Smirnov test regarding X_{sample}.

References

Efron, Bradley and R.J. Tibshirani (May 1994). *An Introduction to the Bootstrap*. Chapman and Hall/CRC. DOI: 10.1201/9780429246593. URL: http://dx.doi.org/10.1201/9780429246593.

Hesterberg, Tim C. (2015). »What Teachers Should Know About the Bootstrap: Resampling in the Undergraduate Statistics Curriculum«. In: *The American Statistician* 69.4. PMID: 27019512, pages 371–386. DOI: 10.1080/00031305.2015.1089789. URL: https://doi.org/10.1080/00031305.2015.1089789.

Overduin, Stephen (2004). »USE OF THE LOGNORMAL DISTRIBTUON FOR SURVIVAL DATA: INFERENCE AND ROBUSTNESS«. Master's thesis. Simon Fraser University. URL: https://core.ac.uk/download/pdf/56374045.pdf.