Assignment 4

Sample distribution and Central Limit Theorem

Roman Grebnev

2022-11-15

Contents

Task 1 . . . 1
Task 1.1 . . . 1
Task 1.2 . . . 1
Task 1.3 . . . 1
Task 1.4 . . . 4
Task 2 . . . 6
Task 2.1 . . . 6
Task 2.2 . . . 7
Task 2.3 . . . 7
Task 2.4 . . . 8
Task 2.5 . . . 11
Task 3 . . . 12

Task 1

Consider the 12 sample data points: 4.94 5.06 4.53 5.07 4.99 5.16 4.38 4.43 4.93 4.72 4.92 4.96

```
sample_emp <- c(4.94, 5.06, 4.53, 5.07, 4.99, 5.16, 4.38, 4.43, 4.93, 4.72, 4.92, 4.96)
```

Task 1.1

How many possible bootstrap samples are there, if each bootstrap sample has the same size as the original?

We can compute 5200300 different bootstrap samples for sample of size 12.

Number of boostrap samples $= \binom{2n+1}{n} = \binom{25}{12} = 5200300$

Task 1.2

Compute the mean and the median of the original sample.

```
mean(sample_emp)
```

```
## [1] 4.840833
```

```
median(sample_emp)
```

```
## [1] 4.935
```

Task 1.3

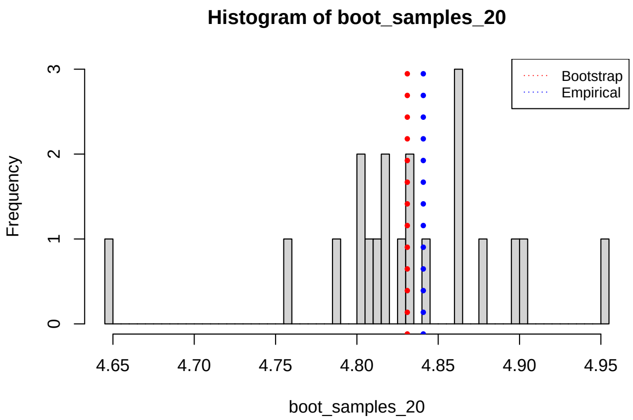
Create 2000 bootstrap samples and compute their means.

Task 1.3.1 Compute the mean on the first 20 bootstrap means.

```
m_20 <- 20
boot_samples_20 <- replicate(m_20, mean(sample(sample_emp, replace=TRUE)))
```

```
hist(boot_samples_20, breaks = 50)
abline(v = mean(boot_samples_20), col = 'red', lty = 3, lwd = 5)
abline(v = mean(sample_emp), col = 'blue', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap", "Empirical"),
       col=c("red", "blue"), lty=3, cex=0.8)

mean_20 <- mean(boot_samples_20)
mean_20
```

```
## [1] 4.831
```

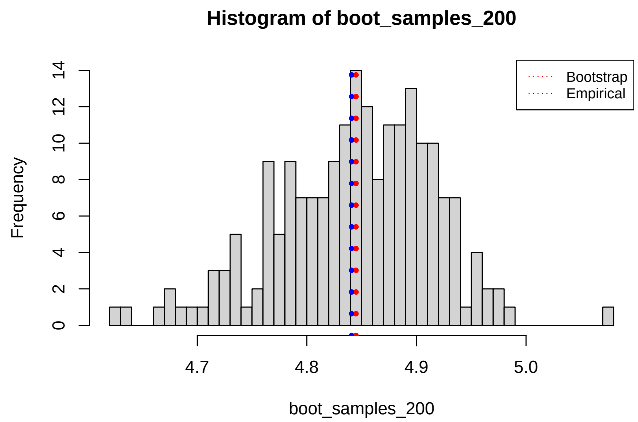


Task 1.3.2 Compute the mean of the first 200 bootstrap means.

```
m_200 <- 200
boot_samples_200 <- replicate(m_200, mean(sample(sample_emp, replace=TRUE)))
```

```
hist(boot_samples_200, breaks = 50)
abline(v = mean(boot_samples_200), col = 'red', lty = 3, lwd = 5)
abline(v = mean(sample_emp), col = 'blue', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap", "Empirical"),
       col=c("red", "blue"), lty=3, cex=0.8)

mean_200 <- mean(boot_samples_200)
mean_200
```

```
## [1] 4.844879
```

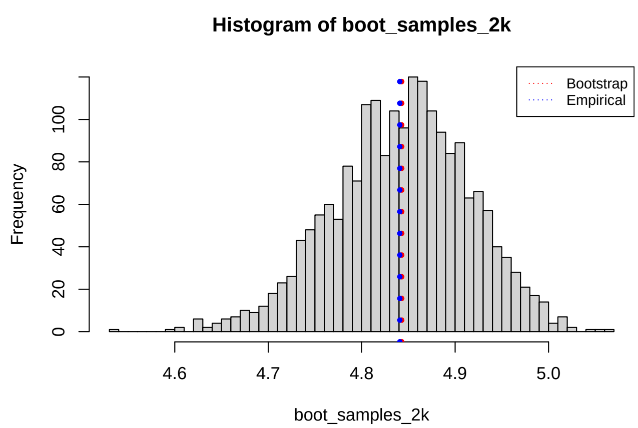


Task 1.3.3 Compute the mean based on all 2000 bootstrap means.

```
m_2k <- 2000
boot_samples_2k <- replicate(m_2k, mean(sample(sample_emp, replace=TRUE)))
```

```
hist(boot_samples_2k, breaks = 50)
abline(v = mean(boot_samples_2k), col = 'red', lty = 3, lwd = 5)
abline(v = mean(sample_emp), col = 'blue', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap", "Empirical"),
       col=c("red", "blue"), lty=3, cex=0.8)

mean_2k <- mean(boot_samples_2k)
mean_2k
```

```
## [1] 4.842868
```



Task 1.3.4 Visualize the distribution all the different bootstrap means to the sample mean. Does the Central Limit Theorem kick in?

Yes, CLT starts working, because the distribution of bootstrap sample means started to converge to the normal distribution.

Task 1.3.5 Based on the three different bootstrap sample lengths in 3. compute the corresponding 0.025 and 0.975 quantiles. Compare the three resulting intervals against each other and the "true" confidence interval of the mean under the assumption of normality. (Use for example the function t.test to obtain the 95% percent CI based on asympotic considerations for the mean.)

Percentile bootstrap CIs can be computed using the quantiles of the bootstrap samples.

```r
# Compute mean distribution quantiles based on 20 bootstrap samples
quant_20_025 <- quantile(boot_samples_20, 0.025, type=1)
quant_20_975 <- quantile(boot_samples_20, 0.975, type=1)
quantiles_20 <- c(quant_20_025, quant_20_975)

# Compute mean distribution quantiles based on 200 bootstrap samples
quant_200_025 <- quantile(boot_samples_200, 0.025, type=1)
quant_200_975 <- quantile(boot_samples_200, 0.975, type=1)
quantiles_200 <- c(quant_200_025, quant_200_975)

# Compute mean distribution quantiles based on 2k bootstrap samples
quant_2k_025 <- quantile(boot_samples_2k, 0.025, type=1)
quant_2k_975 <- quantile(boot_samples_2k, 0.975, type=1)
quantiles_2k <- c(quant_2k_025, quant_2k_975)

# Compute mean distribution quantiles under assumption of normality
t_a2 <- qt(0.95, df = length(sample_emp)-1)
se_emp <- sd(sample_emp)
mean_emp <- mean(sample_emp)
quant_true_025 <- mean_emp - t_a2 * se_emp
quant_true_975 <- mean_emp + t_a2 * se_emp
quantiles_true <- c(quant_true_025, quant_true_975)

comp_table_boot_cis <- data.frame(quantiles_20, quantiles_200, quantiles_2k, quantiles_true)
```

```r
library(knitr)
kable(comp_table_boot_cis)
```

	quantiles_20	quantiles_200	quantiles_2k	quantiles_true
2.5%	4.645833	4.680000	4.694167	4.370247
97.5%	4.955000	4.960833	4.979167	5.311420

Based on the obtained mean CI for different values of bootstrap samples and based on empirical distribution (with normality assumption), we can conclude, that:

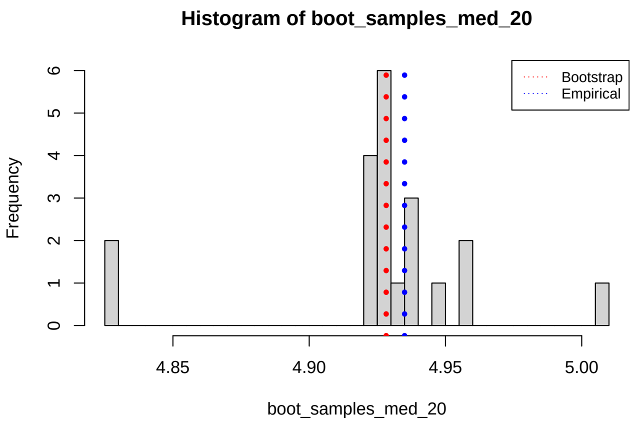
- The higher the number of bootstrap samples, the more precise is the estimation of the two-side 95% CI.
- Bootstrap method allows to compute more precise CIs than based on the t-statistic and empirical values of mean and se.

Task 1.4

Create 2000 bootstrap samples and compute their medians.

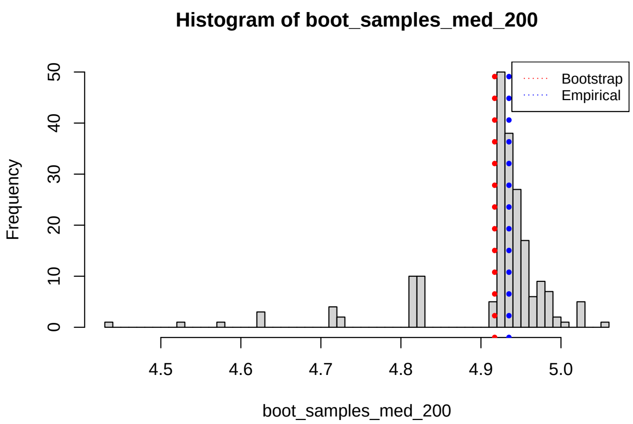
Task 1.4.1 Compute the mean on the first 20 bootstrap medians.

```r
m_20 <- 20
boot_samples_med_20 <- replicate(m_20, median(sample(sample_emp, replace=TRUE)))

hist(boot_samples_med_20, breaks = 50)
abline(v = mean(boot_samples_med_20), col = 'red', lty = 3, lwd = 5)
abline(v = median(sample_emp), col = 'blue', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap", "Empirical"),
       col=c("red", "blue"), lty=3, cex=0.8)
```



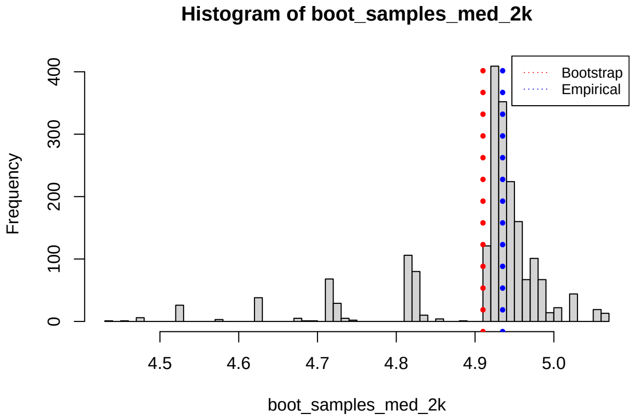
Task 1.4.2 Compute the mean of the first 200 bootstrap medians.

```
m_200 <- 200
boot_samples_med_200 <- replicate(m_200, median(sample(sample_emp, replace=TRUE)))

hist(boot_samples_med_200, breaks = 50)
abline(v = mean(boot_samples_med_200), col = 'red', lty = 3, lwd = 5)
abline(v = median(sample_emp), col = 'blue', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap", "Empirical"),
       col=c("red", "blue"), lty=3, cex=0.8)
```



Task 1.4.3 Compute the mean based on all 2000 bootstrap medians.

```
m_2k <- 2000
boot_samples_med_2k <- replicate(m_2k, median(sample(sample_emp, replace=TRUE)))

hist(boot_samples_med_2k, breaks = 50)
abline(v = mean(boot_samples_med_2k), col = 'red', lty = 3, lwd = 5)
abline(v = median(sample_emp), col = 'blue', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap", "Empirical"),
       col=c("red", "blue"), lty=3, cex=0.8)
```



Task 1.4.4 Visualise the distribution all the different bootstrap medians to the sample median.

Done above.

Task 1.4.5 Based on the three different bootstrap sample lengths in 3. compute the corresponding 0.025 and 0.975 quantiles. Compare the three resulting intervals against each other.

```
# Compute median distribution quantiles based on 20 bootstrap samples
quant_med_20_025 <- quantile(boot_samples_med_20, 0.025, type=1)
quant_med_20_975 <- quantile(boot_samples_med_20, 0.975, type=1)
quantiles_med_20 <- c(quant_med_20_025, quant_med_20_975)

# Compute median distribution quantiles based on 200 bootstrap samples
quant_med_200_025 <- quantile(boot_samples_med_200, 0.025, type=1)
quant_med_200_975 <- quantile(boot_samples_med_200, 0.975, type=1)
quantiles_med_200 <- c(quant_med_200_025, quant_med_200_975)

# Compute median distribution quantiles based on 2k bootstrap samples
quant_med_2k_025 <- quantile(boot_samples_med_2k, 0.025, type=1)
quant_med_2k_975 <- quantile(boot_samples_med_2k, 0.975, type=1)
quantiles_med_2k <- c(quant_med_2k_025, quant_med_2k_975)

comp_table_boot_med_cis <- data.frame(quantiles_med_20, quantiles_med_200, quantiles_med_2k)
```

```
library(knitr)
kable(comp_table_boot_med_cis)
```

	quantiles_med_20	quantiles_med_200	quantiles_med_2k
2.5%	4.825	4.625	4.625
97.5%	5.010	5.025	5.025

Task 2

We wish to explore the effect of outliers on the outcomes of Bootstrap Sampling.

Task 2.1

Set your seed to 1234. And then sample 1960 points from a standard normal distribution to create the vector x.clean then sample 40 observations from uniform(4,5) and denote them as x.cont. The total data is x <- c(x.clean,x.cont). After creating the sample set your seed to your immatriculation number.

```r
set.seed(1234)
x.clean <- rnorm(1960)

x.cont <- runif(40, 4,5)


x <- c(x.clean, x.cont)
set.seed(12202120)
```

Task 2.2

Estimate the median, the mean and the trimmed mean with alpha = 0.05 for x and x.clean.

```r
x_med <- median(x)
x_mean <- mean(x)
x_mean_trim <- mean(x, trim = 0.05)

xc_med <- median(x.clean)
xc_mean <- mean(x.clean)
xc_mean_trim <- mean(x.clean, trim = 0.05)
```

Task 2.3

Use nonparametric bootstrap (for x and x.clean) to calculate standard error 95 percentile CI of all 3 estimators.

```r
se_3_stats <- function(samp, m) {

  boot_mean <- replicate(m, mean(sample(samp, replace=TRUE)))
  boot_median <- replicate(m, median(sample(samp, replace=TRUE)))
  boot_mean_trimmed <- replicate(m, mean(sample(samp, replace=TRUE), trim = 0.05))

  se_mean_CI <- c(quantile(boot_mean, 0.025, type=1),
  quantile(boot_mean, 0.975, type=1))

  se_median_CI <- c(quantile(boot_median, 0.025, type=1),
  quantile(boot_median, 0.975, type=1))

  se_mean_trimmed_CI <- c(quantile(boot_mean_trimmed, 0.025, type=1),
  quantile(boot_mean_trimmed, 0.975, type=1))

  data.frame(se_mean_CI, se_median_CI, se_mean_trimmed_CI)
}

se_x <- se_3_stats(x, 1000)
se_xc <- se_3_stats(x.clean, 1000)
```

```r
se_x
```

```
##       se_mean_CI se_median_CI se_mean_trimmed_CI
## 2.5%  0.03217808  -0.04808725       -0.005639823
## 97.5% 0.13759855   0.06192612        0.085060525
```

```r
se_xc
```

```
##        se_mean_CI se_median_CI se_mean_trimmed_CI
## 2.5%  -0.04834610  -0.06277449        -0.04640577
## 97.5%  0.03449651   0.03958956         0.03898962
```

Task 2.4

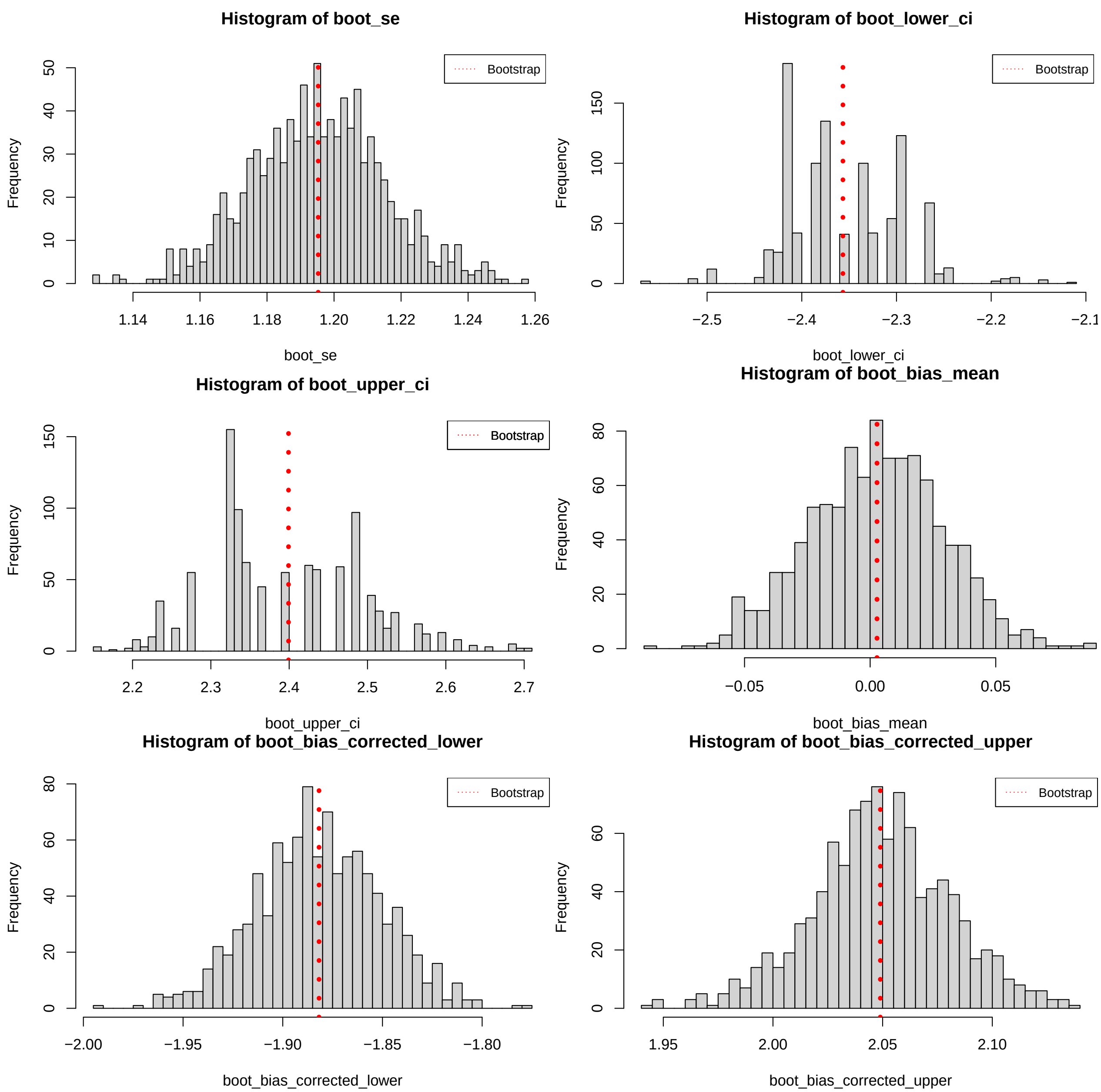
Use parametric bootstrap (based on x and x.clean) to calculate - bias - standard error - 95 percentile CI - bias corrected estimate for the mean and the trimmed mean.

In order to implement parametric bootstrap procedure, we need to estimate $\overline{x}$ and s_n^2 values as approximation of μ and σ. Then we need to simulate artificial samples from the distribution with parameters $\overline{x}$ and s_n^2. Calculate estimators.

Let's compute estimators for x sample.

```
samp <- x
mean_param <- mean(samp)
sd_param <- sd(samp)

sample_norm <- rnorm(2000, mean = mean_param, sd = sd_param)
boot_se <- replicate(1000, sd(sample(sample_norm, replace=TRUE)))
estim_boot_se <- mean(boot_se)

hist(boot_se, breaks = 50)
abline(v = mean(boot_se), col = 'red', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)

boot_lower_ci <- replicate(1000, quantile(sample(sample_norm, replace=TRUE), 0.025, type = 1))
estim_boot_lower_ci <- mean(boot_lower_ci)

hist(boot_lower_ci, breaks = 50)
abline(v = mean(boot_lower_ci), col = 'red', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)

boot_upper_ci <- replicate(1000, quantile(sample(sample_norm, replace=TRUE), 0.975, type = 1))
estim_boot_upper_ci <- mean(boot_upper_ci)

hist(boot_upper_ci, breaks = 50)
abline(v = mean(boot_upper_ci), col = 'red', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)

boot_bias_mean <- replicate(1000, mean(sample(sample_norm, replace=TRUE) - mean_param))
estim_bias_mean <- mean(boot_bias_mean)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)

hist(boot_bias_mean, breaks = 50)
abline(v = mean(boot_bias_mean), col = 'red', lty = 3, lwd = 5)


boot_bias_corrected_lower <- replicate(1000, mean_param - qnorm(0.95) * sd(sample(sample_norm, replace=
estim_bias_corrected_lower <- mean(boot_bias_corrected_lower)
hist(boot_bias_corrected_lower, breaks = 50)
abline(v = mean(estim_bias_corrected_lower), col = 'red', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)

boot_bias_corrected_upper  <- replicate(1000, mean_param + qnorm(0.95) * sd(sample(sample_norm, replace
estim_bias_corrected_upper <- mean(boot_bias_corrected_upper)
hist(boot_bias_corrected_upper, breaks = 50)
abline(v = mean(estim_bias_corrected_upper), col = 'red', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)
```

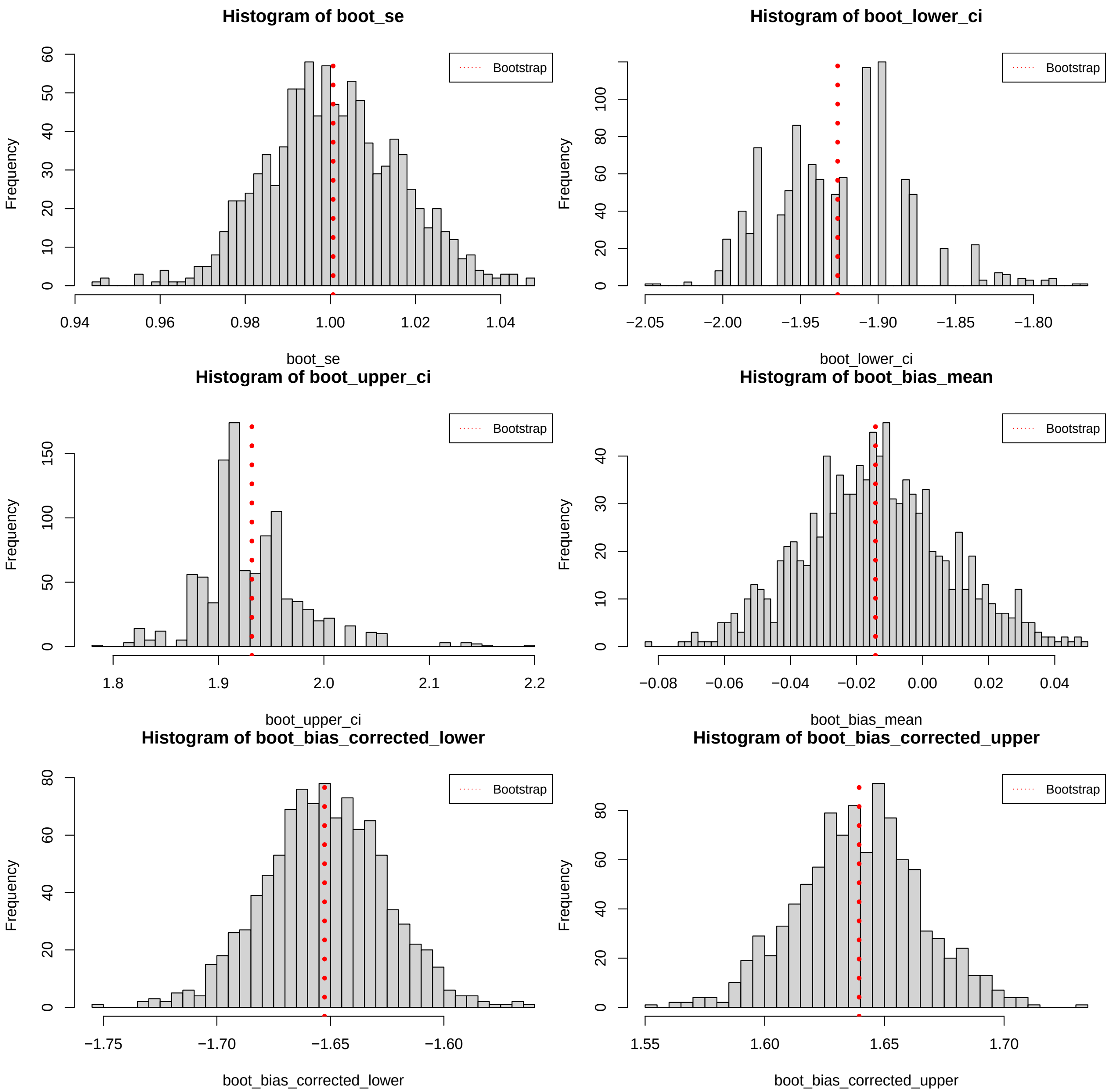
```
rownames_ <- c("Mean", "SD", "Estim boot SE", "Estim boot lower CI", "Estim boot upper CI", "Estim bias

row_x <- c(mean_param, sd_param, estim_boot_se, estim_boot_lower_ci, estim_boot_upper_ci, estim_bias_mea

df_comp <- data.frame(row_x)
rownames(df_comp) <- rownames_
```



Let's compute estimators for x.clean sample.

```
samp <- x.clean
mean_param <- mean(samp)
sd_param <- sd(samp)

sample_norm <- rnorm(2000, mean = mean_param, sd = sd_param)
```

```
boot_se <- replicate(1000, sd(sample(sample_norm, replace=TRUE)))
estim_boot_se <- mean(boot_se)

hist(boot_se, breaks = 50)
abline(v = mean(boot_se), col = 'red', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)

boot_lower_ci <- replicate(1000, quantile(sample(sample_norm, replace=TRUE), 0.025, type = 1))
estim_boot_lower_ci <- mean(boot_lower_ci)

hist(boot_lower_ci, breaks = 50)
abline(v = mean(boot_lower_ci), col = 'red', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)

boot_upper_ci <- replicate(1000, quantile(sample(sample_norm, replace=TRUE), 0.975, type = 1))
estim_boot_upper_ci <- mean(boot_upper_ci)

hist(boot_upper_ci, breaks = 50)
abline(v = mean(boot_upper_ci), col = 'red', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)

boot_bias_mean <- replicate(1000, mean(sample(sample_norm, replace=TRUE) - mean_param))
estim_bias_mean <- mean(boot_bias_mean)

hist(boot_bias_mean, breaks = 50)
abline(v = mean(boot_bias_mean), col = 'red', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)


boot_bias_corrected_lower <- replicate(1000, mean_param - qnorm(0.95) * sd(sample(sample_norm, replace=T
bias_corrected_lower <- mean(boot_bias_corrected_lower)
hist(boot_bias_corrected_lower, breaks = 50)
abline(v = mean(bias_corrected_lower), col = 'red', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)

boot_bias_corrected_upper  <- replicate(1000, mean_param + qnorm(0.95) * sd(sample(sample_norm, replace=
bias_corrected_upper <- mean(boot_bias_corrected_upper)
hist(boot_bias_corrected_upper, breaks = 50)
abline(v = mean(bias_corrected_upper), col = 'red', lty = 3, lwd = 5)
legend(x = "topright", legend=c("Bootstrap"), col=c("red"), lty=3, cex=0.8)

row_xc <- c(mean_param, sd_param, estim_boot_se, estim_boot_lower_ci, estim_boot_upper_ci, estim_bias_m

df_comp <- cbind(df_comp, row_xc)
colnames(df_comp) <- c("x sample", "x.clean sample")
```



Comparison of obtained estimates of statistics based on bootstrap sampling for parametric bootstrap.

When estimating the scale of the of the data in the "robust" case use the mad.

Task 2.5

Compare and summarize your findings with tables and graphically.

```
kable(df_comp)
```

	x sample	x.clean sample
Mean	0.0839551	-0.0059690
SD	1.1656768	0.9899657
Estim boot SE	1.1952710	1.0006490
Estim boot lower CI	-2.3565676	-1.9260527
Estim boot upper CI	2.3991598	1.9316861

	x sample	x.clean sample
Estim bias	0.0027297	-0.0142676
Estim bias corrected lower CI	-1.8818163	-1.8818163
Estim bias corrected upper CI	2.0489415	2.0489415

Visualizations can be found above.

Task 3

Based on the above tasks and your lecture materials, explain the methodology of bootstrapping for the construction of confidence intervals and parametric or non-parametric tests.

Bootstrap is the technique, that is used to evaluate statistical parameters of the population by producing multiple samples from empirical sample based on resampling procedure. Resampling is performed with repetitions, that is why some of the samples might appear in bootstrap dataset more than once. Bootstrap can be used for any statistics and statistical hypothesis testing.

Parametric bootstrap: Bootstrap samples are created based on generated sample from particular distribution using parameters, estimated based on the empirical sample. It is important to note that quality of the parameters estimation depends on the quality of theoretical distribution, used for bootstrapping.

Approaches for CI computation:

- CIs are constructed based on estimated quantiles of bootstrap samples for distribution of statistic θ
- Can be estimated based on normality assumption, using z-statistic and se of the normal distribution (under normality assumption)

Non-parametric bootstrap: Bootstrap samples are created based on the empirical sample. For non-parametric bootstrap, similar approaches for CI computation work, but they are computed using bootstrap samples, drawn from the empirical distribution itself, while for parametric bootstrap samples are drawn from approximated distribution based on parameters of the empirical sample.