

Sampling Intervals for Models

Teodor Chakarov

2023-11-15

Contents

Task 1	2
Task 1.2 - Perform sampling with replacement from each group + centering both samples and then resample from the combined samples x1 and x2 for n1 and n2 times.	2
Taks 1.3 - Bootstrap using both strategies mentioned above using the t-test statistic. Calculate the bootstrap p-value based on 10000 bootstrap samples and 0.95 as well as 0.99 confidence intervals. Make your decision at the significance level 0.05 or 0.01, respectively.	3
Task 1.4 - What would be a permutation version of the test? Implement the corresponding permutation test and obtain p-value and confidence intervals as in 3. to get a corresponding test decision at the same significance levels.	5
1.5 The Wilcoxon rank sum test statistic is the sum of ranks of the observations of sample 1 computed in the combined sample. Use bootstrapping with both strategies mentioned above and obtain p-value and confidence intervals as in 3. to get a corresponding test decision at the same significance levels.	7
1.6 Compare your results to the results using t.test and wilcox.test.	9
Task 2	9
2.1 Create a sample of size 200 from the model above and for the independent predictor x3.	10
2.2 Do residual bootstrap for linear regression and fit the model y follow x1+x2+x3 . Get the percentile CI for the coefficients. Can you exclude x3 ?	10
2.3 Do pairs bootstrap for linear regression and fit the model yfollowx1+x2+x3 . Get the percentile CI for the coefficients. Can you exclude x3 ?	11
2.4 Compare the two approaches in 2. and 3. and explain the differences in the sampling approach and how this (might) affect(s) the results.	12
Task 3	12

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

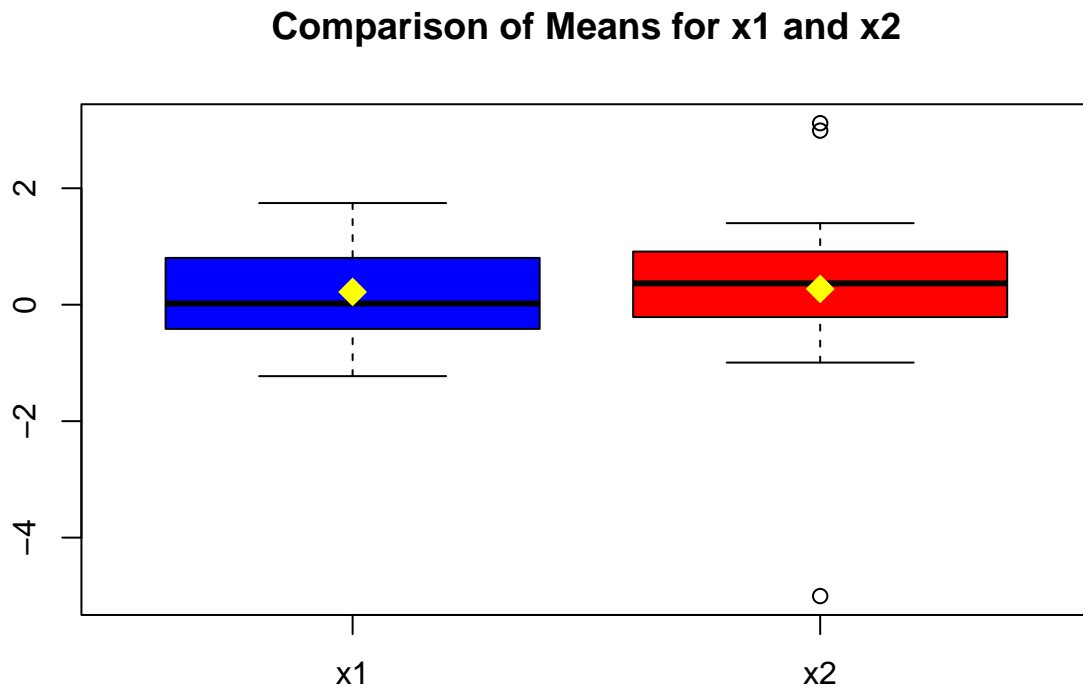
```
library(stats)
```

Task 1

```
x1 <- c(-0.673, -0.584, 0.572, -0.341, -0.218, 0.603, -0.415, -0.013, 0.763, 0.804, 0.054, 1.746, -0.477)
x2 <- c(0.913, -0.639, 2.99, -5.004, 3.118, 0.1, 1.128, 0.579, 0.32, -0.488, -0.994, -0.212, 0.413, 1.404)
```

Task 1.1 Visualize the data

```
boxplot(x1, x2,
        names = c("x1", "x2"),
        col = c("blue", "red"),
        main = "Comparison of Means for x1 and x2")
points(x = c(1,2), y = c(mean(x1), mean(x2)), col = "yellow", pch = 18, cex = 2)
```



Task 1.2 - Perform sampling with replacement from each group + centering both samples and then resample from the combined samples x1 and x2 for n1 and n2 times.

```
set.seed(12141198)
bootstrap_samples = 10000
```

```

samples = numeric(bootstrap_samples)

# Sample with replacement
sample_x1 <- replicate(bootstrap_samples, sample(x1, size=length(x1), replace=TRUE))
sample_x2 <- replicate(bootstrap_samples, sample(x2, size=length(x2), replace=TRUE))

# Centering both samples, combine them and then resampling
data1 <- x1-mean(x1)
data2 <- x2-mean(x2)
combined_data <- c(data1,data2)

cent_sample_x1 <- replicate(bootstrap_samples, sample(combined_data, size=bootstrap_samples, replace=TRUE))
cent_sample_x2 <- replicate(bootstrap_samples, sample(combined_data, size=bootstrap_samples, replace=TRUE))

```

With the bootstrapping: we bootstrap within each group to create new dataset. This preserves the structure and distribution for each group. It allows you to estimate the sampling distribution of the statistic of interest for each group independently.

With centering both samples and then resampling from the combined samples: we adjust both samples so that they have the same mean and standard deviation, and then combine them to create a new sample. Then you resample from this combined sample n_1 and n_2 times to create your bootstrap samples for x_1 and x_2 .

When performing t-statistic values will contain less information with the centered data but may give better results if the samples size are significantly different but the first one is easier to understand.

Taks 1.3 - Bootstrap using both strategies mentioned above using the t-test statistic. Calculate the bootstrap p-value based on 10000 bootstrap samples and 0.95 as well as 0.99 confidence intervals. Make your decision at the significance level 0.05 or 0.01, respectively.

```

test_stat <- function(x1,x2){
  return((mean(x1) - mean(x2))/sd(x1))
}

tests_h0 <- test_stat(x1,x2) #H0: x1 = x2
tests_h0

```

```
## [1] -0.06503765
```

Check the t-stat for bootstrap sample and original sample

```

set.seed(12141198)
bootstap_h0 <- numeric(bootstrap_samples)
p_values_first <- 1

for(i in 1:bootstrap_samples){
  bootstap_h0[i] <- test_stat(sample_x1[,i],x1)
  if(abs(bootstap_h0[i]) > abs(tests_h0)){
    p_values_first <- p_values_first + 1}
}

```

```
p_values_first <- p_values_first/(bootstrap_samples+1)
```

```
# CI
```

```
a = .025
```

```
b = 1-a
```

```
x1_ci_95values <- quantile(bootstap_h0,c(a,b))
```

```
a = .005
```

```
b = 1-a
```

```
x1_ci_99values <- quantile(bootstap_h0,c(a,b))
```

```
print(paste0("P-value: ", p_values_first))
```

```
## [1] "P-value: 0.768623137686231"
```

```
print(paste0("95% CI: "))
```

```
## [1] "95% CI: "
```

```
print(x1_ci_95values)
```

```
##      2.5%      97.5%
```

```
## -0.4693935  0.4216603
```

```
print(paste0("99% CI: "))
```

```
## [1] "99% CI: "
```

```
print(x1_ci_99values)
```

```
##      0.5%      99.5%
```

```
## -0.6289953  0.5853703
```

Our p-value is 0.7686231 for CI 95% (2,5%: -0.4693935 and 97,5%: 0.4216603).

For CI 99% we have (0,5%: -0.6289953 and for 99,5%: 0.5853703)

We can say that based on the results there is not statistically significant difference between the two sets' means, thus we can't reject the $H_0: x_1 = x_2$

Calculation the p-values for both bootstrap samples.

```
t_values1 = numeric(bootstrap_samples)
```

```
p_values_sec <- 1
```

```
for(i in 1:bootstrap_samples){
```

```
  t_values1[i] <- test_stat(cent_sample_x1[,i],cent_sample_x2[,i]) # bootstrap test statistic
```

```

    if( (abs(t_values1[i]) > abs(tests_h0)) ){
      p_values_sec <- p_values_sec + 1}
}

p_values_sec <- p_values_sec/(bootstrap_samples+1)

# CI
a = .025
b = 1-a

x2_ci_95values <- quantile(t_values1,c(a,b))

a = .005
b = 1-a

x2_ci_99values <- quantile(t_values1,c(a,b))

print(paste0("P-value: ", p_values_sec))

## [1] "P-value: 9.99900009999e-05"

print(paste0("95% CI: "))

## [1] "95% CI: "

print(x1_ci_95values)

##          2.5%          97.5%
## -0.4693935   0.4216603

print(paste0("99% CI: "))

## [1] "99% CI: "

print(x1_ci_99values)

##          0.5%          99.5%
## -0.6289953   0.5853703

```

Our p-value is 9.9990001×10^{-5} for CI 95% (2,5%: -0.0274262 and 97,5%: 0.028039).

For CI 99% we have (0,5%: -0.0355358 and for 99,5%: 0.0379204)

We can say that based on the results there is not statistically significant difference between the two bootstrap sets' means, thus we can't reject the $H_0: x_1 = x_2$

Task 1.4 - What would be a permutation version of the test? Implement the corresponding permutation test and obtain p-value and confidence intervals as in 3. to get a corresponding test decision at the same significance levels.

```

ids <- seq(from = 1, to = length(x1) + length(x2), by = 1)
combined <- c(x1, x2)

t_stat_x1 <- c()

for (i in 1:bootstrap_samples) {
  x1_ids <- sample(ids, size = length(x1), replace = FALSE)
  boot_x1 <- combined[x1_ids]
  t_stat_x1 <- c(t_stat_x1, test_stat(boot_x1,x2))
}

p_values_third <- 1

for(i in 1:bootstrap_samples){
  if(abs(t_stat_x1[i]) > abs(tests_h0)){
    p_values_third <- p_values_third + 1}
}
p_value3_x1 <- p_values_third/(bootstrap_samples+1)

# CI
a = .025
b = 1-a

x3_ci_95values <- quantile(t_stat_x1,c(a,b))

a = .005
b = 1-a

x3_ci_99values <- quantile(t_stat_x1,c(a,b))

print(paste0("P-value: ", p_values_third))

## [1] "P-value: 6949"

print(paste0("95% CI: "))

## [1] "95% CI: "

print(x3_ci_95values)

##          2.5%          97.5%
## -0.2861548  0.3155396

print(paste0("99% CI: "))

## [1] "99% CI: "

print(x3_ci_99values)

```

```
##      0.5%      99.5%
## -0.3664215  0.4004514
```

Still we cannot reject the H_0 hypothesis because also with permutation technique from the combined data, still the sets are not different. $\hat{}$

1.5 The Wilcoxon rank sum test statistic is the sum of ranks of the observations of sample 1 computed in the combined sample. Use bootstrapping with both strategies mentioned above and obtain p-value and confidence intervals as in 3. to get a corresponding test decision at the same significance levels.

```
# First method
wil_basic <- wilcox.test(x1, x2, paired = FALSE)$statistic

boot_wilcox <- c()
for(i in 1:bootstrap_samples){
  boot_wilcox <- c(boot_wilcox, wilcox.test(sample_x1[,i], x1, paired = FALSE)$statistic)
}

p_values_fourth <- 1
for(i in 1:bootstrap_samples){
  if(abs(boot_wilcox[i]) > abs(wil_basic)){
    p_values_fourth <- p_values_fourth + 1}
}
p_values_fourth <- p_values_fourth/(bootstrap_samples+1)

# CI
a = .025
b = 1-a
ci_wil_95 <- quantile(boot_wilcox,c(a,b))

a = .005
b = 1-a
ci_wil_99 <- quantile(boot_wilcox,c(a,b))
```

```
print(paste0("P-value: ", p_values_fourth))
```

```
## [1] "P-value: 0.97970202979702"
```

```
print(paste0("95% CI: "))
```

```
## [1] "95% CI: "
```

```
print(ci_wil_95)
```

```
## 2.5% 97.5%
## 184 299
```

```

print(paste0("99% CI: "))

## [1] "99% CI: "

print(ci_wil_99)

## 0.5% 99.5%
## 168 316

boot_wilcox2 <- c()
for(i in 1:bootstrap_samples){
  boot_wilcox2 <- c(boot_wilcox2, wilcox.test(cent_sample_x1[,i], x1, paired = FALSE)$statistic)
}
p_values_fifth <- 1
for(i in 1:bootstrap_samples){
  if(abs(boot_wilcox2[i]) > abs(wil_basic)){
    p_values_fifth <- p_values_fifth + 1}
}
p_values_fifth <- p_values_fifth/(bootstrap_samples+1)

# CI
a = .025
b = 1-a

boot_wilcox2_95 <- quantile(boot_wilcox2,c(a,b))
a = .005
b = 1-a

boot_wilcox2_99 <- quantile(boot_wilcox2,c(a,b))

print(paste0("P-value: ", p_values_fifth))

## [1] "P-value: 1"

print(paste0("95% CI: "))

## [1] "95% CI: "

print(boot_wilcox2_95)

## 2.5% 97.5%
## 92692.93 95273.03

print(paste0("99% CI: "))

## [1] "99% CI: "

```



```
print(boot_wilcox2_99)
```

```
##      0.5%      99.5%  
## 92254.99 95699.03
```

1.6 Compare your results to the results using t.test and wilcox.test.

```
ttest <- t.test(x1,x2, paired = FALSE)  
wilcox <- wilcox.test(x1, x2, paired = FALSE)  
ttest
```

```
##  
## Welch Two Sample t-test  
##  
## data:  x1 and x2  
## t = -0.11881, df = 23.027, p-value = 0.9065  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.9556081 0.8518000  
## sample estimates:  
## mean of x mean of y  
## 0.2198182 0.2717222
```

```
wilcox
```

```
##  
## Wilcoxon rank sum exact test  
##  
## data:  x1 and x2  
## W = 181, p-value = 0.6572  
## alternative hypothesis: true location shift is not equal to 0
```

Task 2

Consider the model $y = 3 + 2 * x_1 + x_2 + e$ where x_1 comes from a normal distribution with mean 2 and variance 3, x_2 comes from a uniform distribution between 2 and 4 and e from a student's t distribution with 5 degrees of freedom . In addition, there is a predictor x_3 coming from a uniform distribution between -2 and 2.

```
model <- function(n){  
  x1 <- rnorm(n, mean = 2, sd = 3)  
  x2 <- runif(n, min = 2, max = 4)  
  eps <- rt(n, df = 5)  
  x3 <- runif(n, min = -2, max = 2)  
  y <- 3 + 2 * x1 + x2 + eps  
  df <- data.frame(y,x1,x2,x3)  
  return(df)  
}
```

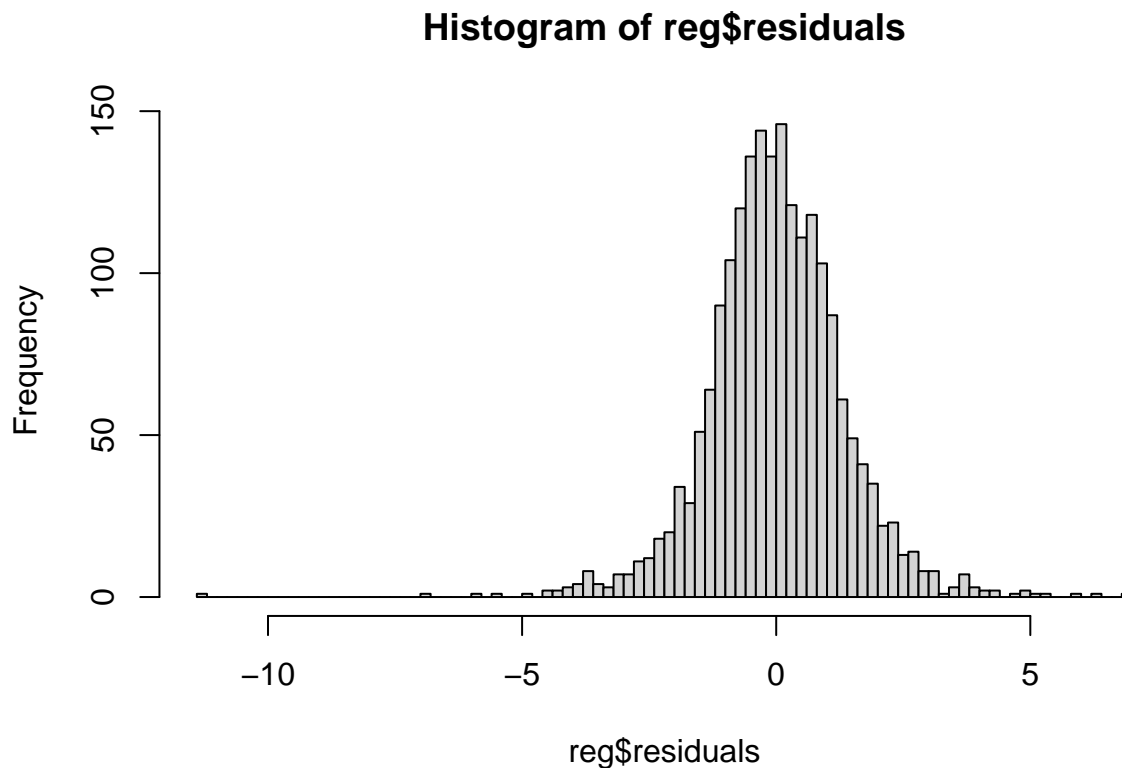
```
x3 <- function(){
  return(runif(1,-2,2))
}
```

2.1 Create a sample of size 200 from the model above and for the independent predictor x3.

```
data <- model(2000)
```

2.2 Do residual bootstrap for linear regression and fit the model y follow $x_1 + x_2 + x_3$. Get the percentile CI for the coefficients. Can you exclude x_3 ?

```
reg <- lm(y~., data=data)
hist(reg$residuals, breaks = 100)
```



The residual distribution is following normal distribution.

```
library("boot")

regfun <- function(x){
  return(coef(lm(y ~ x1+x2+x3, data = x)))
}
```

```

}
sim <- function(x, resi){
  x$y <- (resi$yhat + sample(resi$res, replace = TRUE))
  return(x)
}

res <- resid(reg)
yhat <- fitted(reg)
reg_df <- data.frame(yhat, res)

set.seed(12141198)
bootmodel <- boot(data, regfun, R=1000, sim="parametric", ran.gen=sim, mle = reg_df)
x1_ci <- quantile(bootmodel$t[,2], c(0.005,0.025,0.975, 0.995))
x2_ci <- quantile(bootmodel$t[,3], c(0.005,0.025,0.975, 0.995))
x3_ci <- quantile(bootmodel$t[,4], c(0.005,0.025,0.975, 0.995))

df <- data.frame(x1 = x1_ci, x2 = x2_ci, x3 = x3_ci)

df

##           x1           x2           x3
## 0.5%  1.983715  0.8547834 -0.06247829
## 2.5%  1.988378  0.8995340 -0.04568112
## 97.5% 2.030385  1.1065418  0.05608774
## 99.5% 2.035937  1.1323513  0.07697526

```

Thanks to the confidence interval in the previous graph we can assume that x_3 is not significant and we can exclude x_3 .

2.3 Do pairs bootstrap for linear regression and fit the model $y \sim x_1 + x_2 + x_3$

. Get the percentile CI for the coefficients. Can you exclude x_3 ?

```

reg_fun <- function(x, idx){
  return(coef(lm(y ~ x1+x2+x3, data = x[idx, ])))
}

fit <- lm(y ~ x1+x2+x3, data = data)
res <- resid(fit)
yhat <- fitted(fit)
reg_model <- data.frame(yhat, res)

fit.boot_pairs <- boot(data, reg_fun, R = 1000)

bot_pair_sol <- fit.boot_pairs$t
x1_pairs <- quantile(bot_pair_sol[,2], c(0.005,0.025,0.975, 0.995))
x2_pairs <- quantile(bot_pair_sol[,3], c(0.005,0.025,0.975, 0.995))
x3_pairs <- quantile(bot_pair_sol[,4], c(0.005,0.025,0.975, 0.995))

df <- data.frame(x1 = x1_ci, x2 = x2_ci, x3 = x3_ci)

```

df

```
##           x1           x2           x3
## 0.5%  1.983715 0.8547834 -0.06247829
## 2.5%  1.988378 0.8995340 -0.04568112
## 97.5% 2.030385 1.1065418  0.05608774
## 99.5% 2.035937 1.1323513  0.07697526
```

Our results are very similar to what we obtained with residual bootstrap.

2.4 Compare the two approaches in 2. and 3. and explain the differences in the sampling approach and how this (might) affect(s) the results.

The distinction between the two methods is that the first assumes residuals are normally distributed, in bootstrapping where new samples are created. While in pair bootstrapping, existing computed residuals are used. Both methods are similar when applied to a large dataset and a well-specified model. However, they can give different results with small sample sizes. The residual method is generally more precise for the correct model, while the pair method is more reliable if the model does not accurately reflect the data.

Task 3

Summarize the bootstrapping methodology, its advantages and disadvantages based on your exercises for constructing parametric and non-parametric confidence bounds for estimators, test statistics or model estimates.

Based on the exercises, we've been provided with valuable insights into the variability and potential bias of estimators or test statistics. Parametric bootstrapping is efficient when the model is correct and assumptions are met (that the data follow a normal distribution). In contrast, non-parametric bootstrapping is more robust against model misspecification, though it may not always provide the most accurate confidence intervals, especially in small samples. The choice between the two methods should be guided by the sample size, the distribution of the data, and the specifics of the problem.