Exercise 7

Bootstraping Practicals IV

Olesia Galynskaia 12321492

2025

Fix the random seed for reproducibility

```
set.seed(12321492)
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE, fig.width = 6, fig.height = 4, dpi
```

Task 12

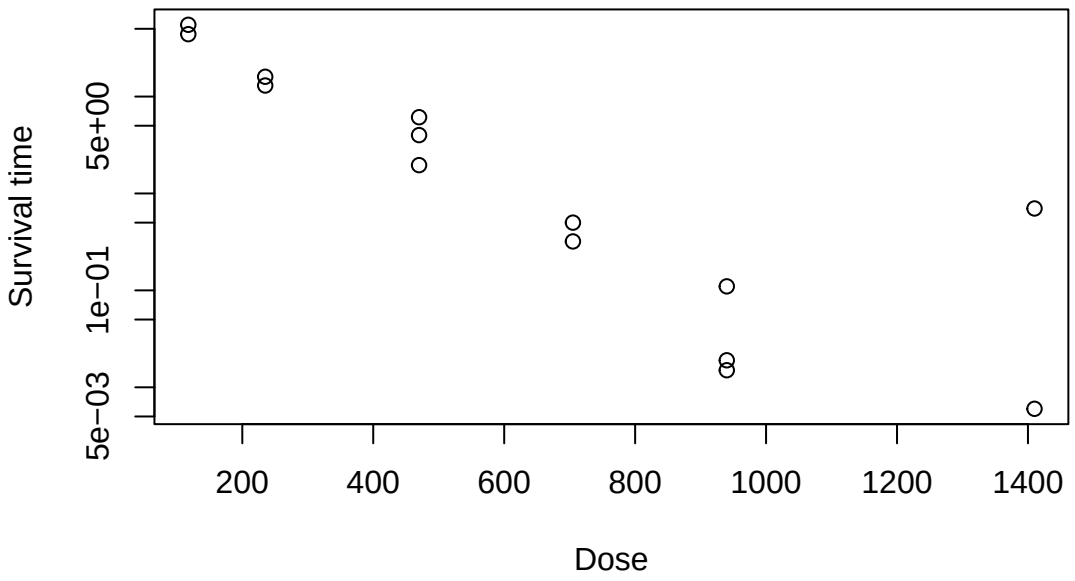
1. Redo the analysis for the survival data but without the outlying observation

I start with the survival dataset used in the lecture example.
First, I inspect the relationship between dose and survival time on a log scale in order to detect possible outlying observations.
From the scatter plot one observation stands out with a much larger survival time than the rest of the sample.
Using which.max(surv), this observation is identified and removed from the data.

```
library(boot)

# load survival data from the boot package
data(survival)

# initial visualization to detect outliers
plot(survival$dose, survival$surv, log = "y",
     xlab = "Dose", ylab = "Survival time")
```




```
# identify the outlying observation
which.max(survival$surv)
```

```
## [1] 2
```

```
# remove the outlying observation
outlier_id <- which.max(survival$surv)
survival_no <- survival[-outlier_id, ]
```

Linear regression without the outlier

```
fit_no <- lm(log(surv) ~ dose, data = survival_no)
summary(fit_no)
```

```
##
## Call:
## lm(formula = log(surv) ~ dose, data = survival_no)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4548 -0.7044 -0.2489  0.5174  4.0549
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.588518   0.927433   3.869 0.002611 **
## dose        -0.005674   0.001153  -4.920 0.000457 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.676 on 11 degrees of freedom
## Multiple R-squared:  0.6876, Adjusted R-squared:  0.6592
## F-statistic: 24.21 on 1 and 11 DF,  p-value: 0.0004568
```

After removing the outlying observation, the linear regression of log(surv) on dose shows a clear negative relationship between dose and survival time. The estimated slope is -0.0057 and is highly significant, indicating that higher doses are associated with shorter survival times on the log scale. The model explains a substantial part of the variability in the data, with an adjusted R-squared of about 0.66.

Pairs bootstrap

```
reg.fun.no <- function(x, i) {
  x.i <- x[i, ]
  fit <- lm(log(surv) ~ dose, data = x.i)
  coef(fit)
}

surv.boot.no <- boot(survival_no, reg.fun.no, R = 1000)
surv.boot.no
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = survival_no, statistic = reg.fun.no, R = 1000)
##
##
## Bootstrap Statistics :
##         original        bias    std. error
## t1*  3.588517947  0.0869702121 0.862342219
## t2* -0.005673805 -0.0002423147 0.001646649
```

The pairs bootstrap confirms the regression results obtained from the reduced sample.
The bootstrap standard error of the slope is relatively small, and the bootstrap distribution is centered close to the original estimate.
This indicates that, after removing the outlier, the estimated effect of dose is stable and not driven by a single influential observation.

Overall, removing the outlying observation leads to more stable regression and bootstrap results, while the negative effect of dose on survival time remains clearly present.

2. Redo the analysis with the full sample but using function rq with tau = 0.5

I use quantile regression with tau = 0.5, which corresponds to median regression.

```
library(quantreg)

# quantile regression (median regression) on the full sample
fit_rq <- rq(log(surv) ~ dose, data = survival, tau = 0.5)
summary(fit_rq)
```

```
##
## Call: rq(formula = log(surv) ~ dose, tau = 0.5, data = survival)
##
## tau: [1] 0.5
##
## Coefficients:
##             coefficients lower bd upper bd
## (Intercept)  4.43254      3.64577  4.79695
## dose        -0.00706     -0.00858 -0.00656
```

The estimated slope is negative, indicating that higher doses are associated with shorter survival times at the median level.
The confidence interval for the slope does not include zero, which suggests that the effect of dose is statistically significant.
Compared to ordinary least squares regression, the median regression is less influenced by the extreme survival time and provides a more robust estimate of the central tendency of the data.

3. Comparison of the results from 1 and 2

```
ols_coef <- coef(fit_no)

rq_coef <- coef(fit_rq)

comparison_table <- data.frame(
  Method = c("OLS without outlier", "Median regression (tau = 0.5)"),
  Intercept = c(ols_coef[1], rq_coef[1]),
  Dose_effect = c(ols_coef[2], rq_coef[2])
)

comparison_table
```

```
##                           Method Intercept  Dose_effect
## 1           OLS without outlier  3.588518 -0.005673805
## 2 Median regression (tau = 0.5)  4.432543 -0.007063636
```

In Task 12.1, the analysis was performed after removing the outlying observation and using ordinary least squares regression.
In Task 12.2, the full sample was used, but median regression was applied instead, which is less sensitive to outliers.
Both approaches lead to a negative and significant effect of dose on survival time, indicating a consistent relationship between dose and survival.
However, the median regression yields more robust estimates in the presence of the outlier, while removing the outlier leads to more stable results for the mean-based regression.
Overall, both methods confirm the same qualitative conclusion, but they address the influence of the outlying observation in different ways.

Task 13

1. Create a sample of size 200 from the model

I generate a sample of size $n = 200$ from the model

$$y = 3 + 2x_1 + x_2 + \varepsilon,$$

where $x_1 \sim \mathcal{N}(2, 3)$, $x_2 \sim U(2, 4)$, $x_3 \sim U(-2, 2)$ and $\varepsilon \sim t_5$.

The predictor x_3 is generated.

```
n <- 200

x1 <- rnorm(n, mean = 2, sd = sqrt(3))
x2 <- runif(n, min = 2, max = 4)
x3 <- runif(n, min = -2, max = 2)

eps <- rt(n, df = 5)

y <- 3 + 2 * x1 + x2 + eps

# create data frame
dat <- data.frame(y, x1, x2, x3)

summary(dat)
```

```
##        y                x1                x2              x3
##  Min.   :-5.053   Min.   :-4.8793   Min.   :2.016   Min.   :-1.999484
##  1st Qu.: 6.942   1st Qu.: 0.5862   1st Qu.:2.549   1st Qu.:-1.026457
##  Median :10.088   Median : 2.1367   Median :3.080   Median : 0.020948
##  Mean   : 9.771   Mean   : 1.9330   Mean   :3.034   Mean   :-0.008272
##  3rd Qu.:12.207   3rd Qu.: 3.1133   3rd Qu.:3.537   3rd Qu.: 1.015216
##  Max.   :19.720   Max.   : 6.2006   Max.   :4.000   Max.   : 1.991945
```

2. Residual bootstrap

I apply the residual bootstrap for the linear regression model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon.$$

This approach treats the residuals as the independent and identically distributed part of the data and assumes that the regression model is correctly specified.

Percentile bootstrap confidence intervals are computed for the regression coefficients.
The confidence interval for β_3 is used to assess whether the predictor x_3 can be excluded.

If the confidence interval contains zero, β_3 is not statistically significant and can be excluded from the model.

```
library(boot)

# fit model once on original data
fit <- lm(y ~ x1 + x2 + x3, data = dat)
yhat <- fitted(fit)
```

```r
res  <- resid(fit)

# statistic function: fit same model and return coefficients
stat_fun <- function(d) {
  coef(lm(y ~ x1 + x2 + x3, data = d))
}

# generator for residual bootstrap: keep x's, resample residuals, rebuild y
ran_gen <- function(d, mle) {
  d$y <- mle$yhat + sample(mle$res, replace = TRUE)
  d
}

# store "mle" (here: yhat and residuals) for ran_gen
mle_obj <- data.frame(yhat = yhat, res = res)

# run residual bootstrap
res_boot <- boot(dat, statistic = stat_fun, R = 1000,
                 sim = "parametric", ran.gen = ran_gen, mle = mle_obj)

# percentile CI for beta_3 (coefficient of x3 is index 4)
boot.ci(res_boot, type = "perc", index = 4)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = res_boot, type = "perc", index = 4)
##
## Intervals :
## Level     Percentile
## 95%   (-0.0788,  0.1848 )
## Calculations and Intervals on Original Scale
```

My first attempt did not produce bootstrap confidence intervals because the bootstrap statistic was identical in all resamples.
This happened because the index argument was not used and the model was fitted on the full data in every replication.
After fixing the resampling step, the residual bootstrap produces a non-degenerate bootstrap distribution and percentile confidence intervals can be computed.

The residual bootstrap produces a percentile confidence interval for the coefficient β_3 of (-0.0788, 0.1848).

Since this interval contains zero, the coefficient β_3 is not statistically significant. Therefore, based on the residual bootstrap results, the predictor x_3 can be excluded from the linear regression model.

3. Pairs bootstrap

I next apply the pairs bootstrap, where entire observation vectors

$$(y_i, x_{1i}, x_{2i}, x_{3i})$$

are resampled with replacement.

This approach does not rely on assumptions about the residuals and is more robust to model misspecification.

Again, percentile bootstrap confidence intervals are computed for the regression coefficients.
The confidence interval for β_3 obtained from the pairs bootstrap is used to decide whether x_3 can be excluded.

```
# bootstrap function for pairs bootstrap
pair_fun <- function(data, i) {
  data_i <- data[i, ]
  coef(lm(y ~ x1 + x2 + x3, data = data_i))
}

# pairs bootstrap
pair_boot <- boot(dat, pair_fun, R = 1000)

# percentile CI for beta_3 (index = 4)
pair_ci_x3 <- boot.ci(pair_boot, type = "perc", index = 4)
pair_ci_x3
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = pair_boot, type = "perc", index = 4)
##
## Intervals :
## Level     Percentile
## 95%   (-0.0895,  0.1851 )
## Calculations and Intervals on Original Scale
```

The pairs bootstrap yields a percentile confidence interval for the coefficient β_3 of (-0.0895, 0.1851).

Since this interval contains zero, the coefficient β_3 is not statistically significant. Therefore, based on the pairs bootstrap results, the predictor x_3 can also be excluded from the model.

4. Summary

For both bootstrap methods, percentile confidence intervals were computed for the coefficient β_3.
In the residual bootstrap, the 95% confidence interval was (-0.0788, 0.1848). In the pairs bootstrap, the corresponding interval was (-0.0895, 0.1851).

Since both confidence intervals contain zero, the coefficient β_3 is not statistically significant under either bootstrap approach.
Therefore, the predictor x_3 can be excluded from the model.

Both bootstrap methods lead to the same conclusion, providing consistent evidence that x_3 does not contribute to explaining the response variable y.