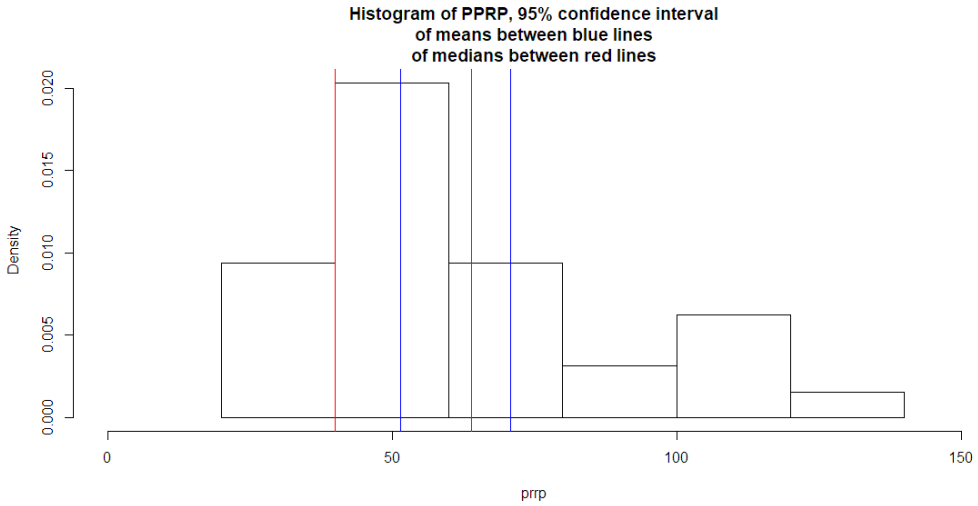
5.1

The 95%-bootstrap confidence interval of the mean of β-thromboglobuline levels of patients without organ impediments (PRRP) is [51.45, 71.08].* The 95%-bootstrap confidence interval of the medians of PRRP is [40.00, 64.00]. Considering the 61.56 mean and 28.82 standard deviation, we prefer the confidence interval of the mean estimator of location. It is more centered on the data, as is visualized in the following histogram.**



The 95%-bootstrap confidence interval of the difference between the means of PRRP and β-thromboglobuline levels patients with scleroderma (SDRP) is [-16.36, 15.13]. The fact that the value 0 is quite central in this interval makes us doubt the existence of a systematic difference between the two groups of patients, although to conclude much based on just the means would be presumptuous.

* Here and throughout this report randomness is involved in the process of bootstrapping, and reported results can be the those of single *runs*.

** Although this is almost true by the definitions of *mean* and *location*.

5.2

A. We try to bootstrap-estimate the variance of an unbiased estimator for the paramater θ of a sample size 50 from a unif(0, θ)-distribution, with 'unknown' $\theta = 1$. With the unbiased estimator $T_n = \frac{n+1}{n} X_{(n)}$ and 1000 bootstrap samples we get a value var $\bar{T}_n = 0.00077$ for the variance of our unbiased estimator T_n.

B. Here we see multiple attempts at the bootstrap estimation of the variance of $\hat{\theta}_1$ with 1000 bootstrap samples:

Attempt	1	2	3	4	5	mean ± sd over 100 attempts
Result	0.028	0.021	0.005	0.012	0.034	0.021 ± 0.012

Here we use the standard deviation s to more clearly illustrate the differences between each estimation. We can see a lot of variance in the results. For a bit more meaningful conclusion, we must derive the theorethical value for the variance of $\hat{\theta}_1$. So if we look at a general i.i.d. sample from a unif(0, θ) distribution we have:

$$\begin{aligned}
\mathbb{E}_\theta X_{(n)} &= \int_0^\theta x n x^{n-1} \frac{1}{\theta^n} \mathrm{d}x = \frac{n}{n+1}\theta \\
\mathbb{E}_\theta (X_{(n)}^2) &= \int_0^\theta x^2 n x^{n-1} \frac{1}{\theta^n} \mathrm{d}x = \frac{n}{n+2}\theta \\
\mathrm{var}_\theta X_{(n)} &= \mathbb{E}_\theta (X_{(n)}^2) - (\mathbb{E}_\theta X_{(n)})^2 = \theta^2 \frac{n}{(n+1)^2(n+2)} \\
\mathrm{var}_\theta T_n &= \frac{\theta^2}{n(n+2)} \\
\mathrm{var}_\theta T_n|_{\theta=1} &= \frac{1}{52 \times 50} \approx 0.00038 \\
s &= \sqrt{\frac{1}{52 \times 50}} \approx 0.0196.
\end{aligned}$$

As we see in the tabel of our attempts, this is relatively close (about one standard deviation off) the average of our bootstrap results.

C. Now we use the paramatric bootstrap method to again estimate the variance of a sample X from unif(0, θ), with underlying 'unknown' parameter $\theta = 1$. Suppose θ is unknown to us, then we could use the unbiased

$$\dot{\theta} = 2 \sum_{i=1}^{50} \frac{X_i}{n}$$

as a bootstrap parameter since the mean of an unif(a, b)-distribution equals $\frac{1}{2}(b-a)$ or we could just use T_n – we are making an estimation error (see slide 16/21 of lecture 4) but perhaps using 2θ offers the same amount risks. We opted to go with the estimator T_n and show the standard deviations s over multiple attempts using the parametric method in the following table.

Attempt	1	2	3	4	5	mean ± sd over 100 attempts
Result	0.0209	0.0201	0.0197	0.0177	0.0188	0.019 ± 1e-3

This time, the variances (and standard deviations) are consistent and close to the theoretical value.

D. The empirical bootstrap seems to struggle with too much bias given this situation. We would prefer using the apparently more robust parametric approach in this case.

We could try to alleviate the former problem somewhat by using a biased estimator $T_n^* = \frac{n+2}{n+1} X_{(n)}$ for the empirical bootstrap, resulting in a smaller error. (Consult the following table for results.)

Attempt	1	2	3	4	5	mean ± sd over 100 attempts
Result	0.0177	0.0186	0.0188	0.0196	0.0179	0.020 ± 1e-3

Appendix

```
## Exercise 5.1
## A
source("./thromboglobulin.txt")
prrp = thromboglobulin$PRRP
B = 1000; alph = .05
means = bootstrap(prrp, mean, B=B)
hist(means);
t = mean(prrp)
z = means - t
meanC = c(t-quantile(z, probs=1-alph/2), t-quantile(z, probs=alph/2))
meanC

## B
medians = bootstrap(prrp, median, B=B)
hist(medians)
t2 = median(prrp)
z2 = medians - t2;
mediC = c(t2-quantile(z2, probs=1-alph/2), t2-quantile(z2, probs=alph/2))
mediC # [40.00, 64.00]

## C
hist(prrp, prob=TRUE, xlim=c(0,150))
abline(v=meanC[1], col="blue")
abline(v=meanC[2], col="blue")
abline(v=mediC[1], col="red")
abline(v=mediC[2], col="red")
mean(prrp); sd(prrp)

## D
sdrp = thromboglobulin$SDRP
diffs = numeric(B)
for (i in 1:B) {
  diffs[i] = mean(sample(prrp, replace=TRUE)) -
      mean(sample(sdrp, replace=TRUE)) - (mean(prrp) - mean(sdrp))
}
hist(diffs)
diffC = c(quantile(diffs, probs=alph/2), quantile(diffs, probs=1-alph/2))
diffC # [-16.36, 15.13]

## Exercise 5.2
## A
n = 50; X = runif(n, 0, 1)
B = 1000
EmpBS = numeric(B);
for (i in 1: B) {
  xstar = sample(X, replace = TRUE)
  EmpBS[i] = ((n + 1) / n) * max(xstar)
}
var(EmpBS)
sd(EmpBS)

## B
sdevs = numeric(100)
for (j in 1:100) {
  n = 50; X = runif(n, 0, 1)
  B = 1000
  EmpBS = numeric(B);
```

```
  for (i in 1: B) {
    xstar = sample(X, replace = TRUE)
    EmpBS[i] = ((n + 1) / n) * max(xstar)
  }
  sdevs[j] = sd(EmpBS)
}
mean(sdevs) # 0.0209127
sd(sdevs) # 0.01150859

## C
X = runif(50,0,1)
B = 1000
theta = (50+1)/50*max(X)
ParBS = numeric(B);
for (i in 1: B)
{
 xstar = runif(length(X), 0,theta)
 ParBS[i] = ((n+1)/(n))*max((xstar))
}
var(ParBS)
sd(parBS)

sdevs = numeric(100)
for (j in 1:100) {
  X = runif(50,0,1)
  B = 1000
  theta = (50+1)/50*max(X)
  ParBS = numeric(B)
  for (i in 1: B)
  {
    xstar = runif(length(X), 0,theta)
    ParBS[i] = ((n+1)/(n))*max((xstar))
  }
  sdevs[j] = sd(ParBS)
}
mean(sdevs)
sd(sdevs)

## D
m = 100
Xpar = numeric(m)
for (j in 1: m)
{
 X=runif(50,0,1)

 B=1000
 theta = ((50+2)/(50+1))*max(X)
 ParBS = numeric(B);
 for (i in 1: B)
 {
   xstar = runif(length(X), 0, theta)
   ParBS[i] = ((n + 1) / n) * max((xstar))
 }
 Xpar[j] = sd(ParBS)
}
mean(Xpar)
sd(Xpar)
```