Bootstrapping

The bootstrap method is a simulation procedure that approximates the distribution of things like the sample mean of a finite sample size. The method is generally applicable to other sample statistics than the sample mean as well.

In practice, let's say you have a sample set S of size n. To compute the bootstrapped sample mean of S, one would pick n elements from S, where picking multiple identical elements is allowed. The mean of those n elements is then calculated and stored. This process is done usually 10 000 times. What we end up with is the bootstrapped distribution of the sample mean of S. The expected value or average of that 'new' distribution is called the bootstrapped sample mean.

Put simply, bootstrapping consists of five steps:

Suppose you have a dataset S that conists of n elements.

1. Make a bootstrapped dataset
    - From the original dataset select n random elements with replacements i.e., duplicate element are allowed. This is called *sampling with replacement*
2. Calculate a statistic
    - If you do empirical bootstrapping then this could be: sample mean, sample variance, sample median, standard errors, confidence intervals or even p-values
    - If you do parametric bootstrapping then this could be the Kolmogorov-Smirnov distance between the bootstrapped dataset and the original dataset.
3. Save the calculation somewhere
4. Repeat step 1 through 3 a bunch of times usually 10 000 times.
5. Compare the calculated statistic with the 'real' statistic.
    - In empirical bootstrapping you could e.g., compare the sample mean with the population or 'true' mean
    - In parametric bootstrapping you could e.g., compare the Kolmogorov-Smirnov distances from the bootstrapped dataset and the Kolmogorov-Smirnov distance between the

Below are some common parameters with their corresponding sample statistic one might be interested in measuring

Measurement	**Sample statistic**	**Population parameter**
Mean	$\bar{x}$	μ (mu)
Standard deviation	s	σ (sigma)
Variance	s^2	σ^2 (sigma squared)
Proportion	p	π (pi)
Correlation	r	ρ (rho)
Regression coefficient	b	β (beta)

Table 1: Common parameters with their corresponding sample statistic

In any problem, we are always interested in measuring the population parameter. However, it's often too time-consuming, too costly, or simply not possible to actually measure every single individual element in the population, which is why we instead calculate a sample statistic and use that statistic to estimate the true population parameter. This can become more 'accurate' with bootstrapping as we know from the law of large numbers.

Empirical bootstrapping

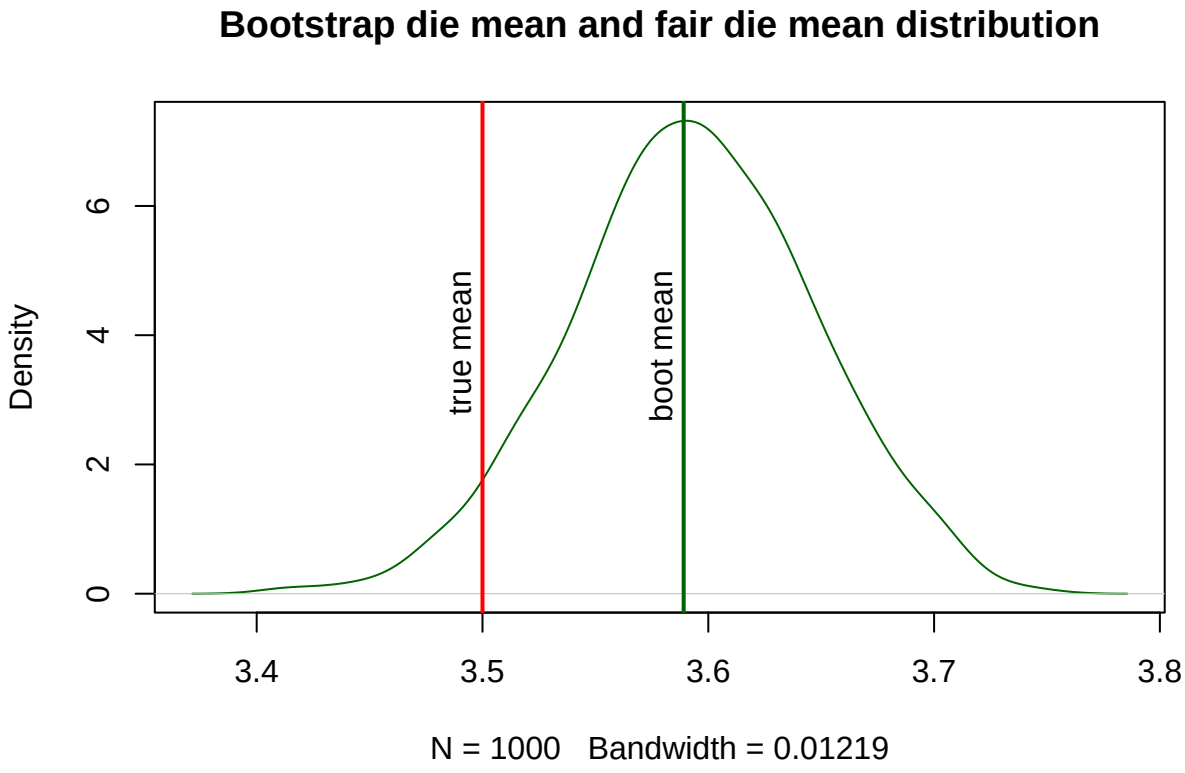
The method described above can be used to do what is called *empirical bootstrapping* which you do when you have no knowledge about the distribution from which the random sample is from.

Imagine a situation of purchasing an antique die from the internet for gambling. Before purchasing, you want to make sure that the die is fair. The seller provides 1000 samples of the outcomes available in the file `die_samples.Rdata`

```
load(here::here('assets', 'die_samples.Rdata'))
```

Determine whether the die is fair or not using bootstrapping

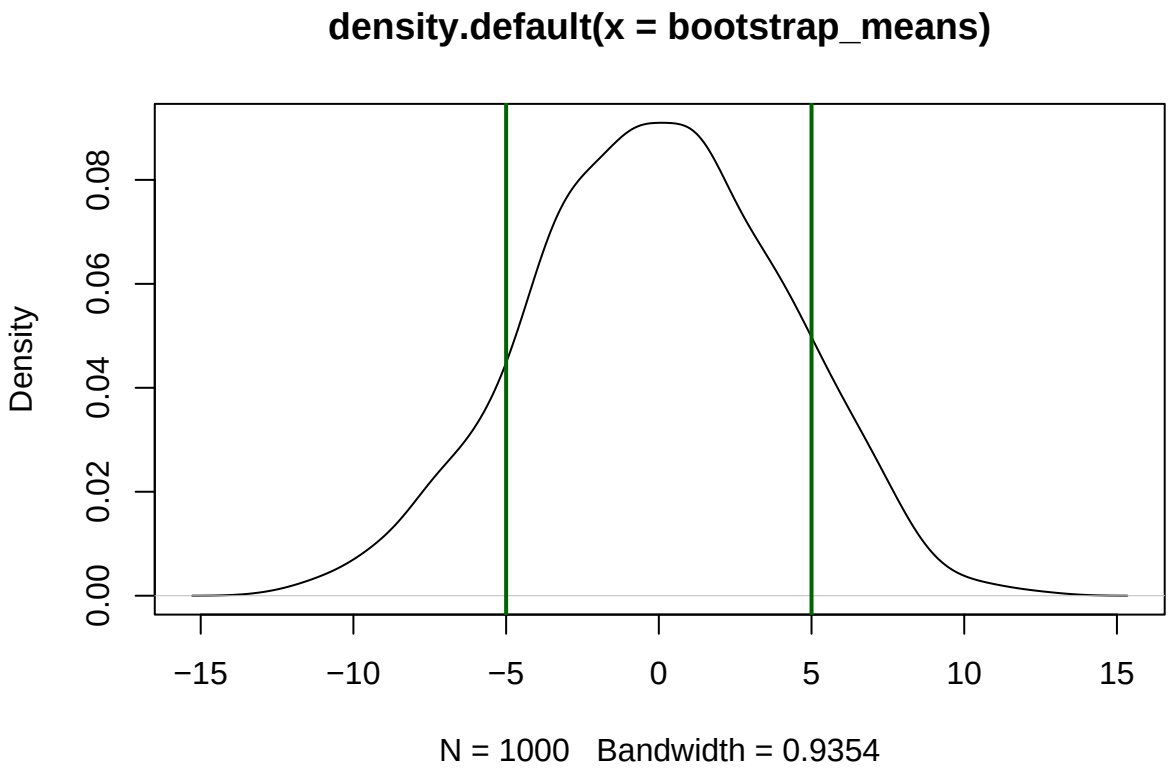
```
boots <- function(data) {
  means <- c()
  for (i in 1:1000) {
    s <- sample(data, 1000, replace = T)
    means <- append(means, mean(s))
    }
    return(means)
    }
bootstrap_mean <- mean(boots(die_samples))
plot(density(boots(die_samples)),
    main = "Bootstrap die mean and fair die mean distribution",
      col = "darkgreen")
abline(v = mean(c(1, 2, 3, 4, 5, 6)), col = "red", lwd = 2)
text(3.5, 5, "true mean", srt = 90, pos = 2)
abline(v = bootstrap_mean, col = "darkgreen", lwd = 2)
text(bootstrap_mean, 5, "boot mean", srt = 90, pos = 2)
```



As is visible from the plot the die is not totally fair as it has a mean of 3.589094 whereas a completely fair die has a mean of 3.5.

We can also use these bootstrapped means to bound the probability that the sample mean is a certain distance away from the norm. If we take the Old Faithful dataset again. Suppose we want to know the probability that the sample mean is more than say 5 seconds away from the norm, we can do as follows

```r
eruptions <- faithful$eruptions * 60 # multiplying by 60 because dataset is in minutes
sample_mean <- mean(eruptions)
bootstrap_means <- c()
for (i in 1:1000) {
  bootstrap_sample <- sample(eruptions, size = length(eruptions), replace = TRUE)
  bootstrap_means <- c(bootstrap_means, sample_mean - mean(bootstrap_sample))
}
plot(density(bootstrap_means))
abline(v = 5, lwd = 2, col = "darkgreen")
abline(v = -5, lwd = 2, col = "darkgreen")
```



Since we want to know the probability that the sample mean is more than 5 seconds away from the norm we simply have to subtract the probability that the sample mean is $+5$ seconds away and -5 seconds away from the norm. This would give us the probability that the sample mean is between -5 and $+5$ seconds away from the norm so in order to find out the probability that the sample mean is more and 5 seconds away from the norm we have to subtract the whole thing by 1

```r
bootstrap_means_diff_dist <- ecdf(bootstrap_means)
probability_outside <- 1 - (bootstrap_means_diff_dist(5) - bootstrap_means_diff_dist(-5))
# calling ecdf(bootstrap_means)(5) tells us probability that it is exactly 5
# seconds away from mean
probability_outside
```

```
## [1] 0.231
```

As such the probability that the sample mean is more than 5 seconds away from the norm is $\approx 23\%$

Parametric bootstrapping

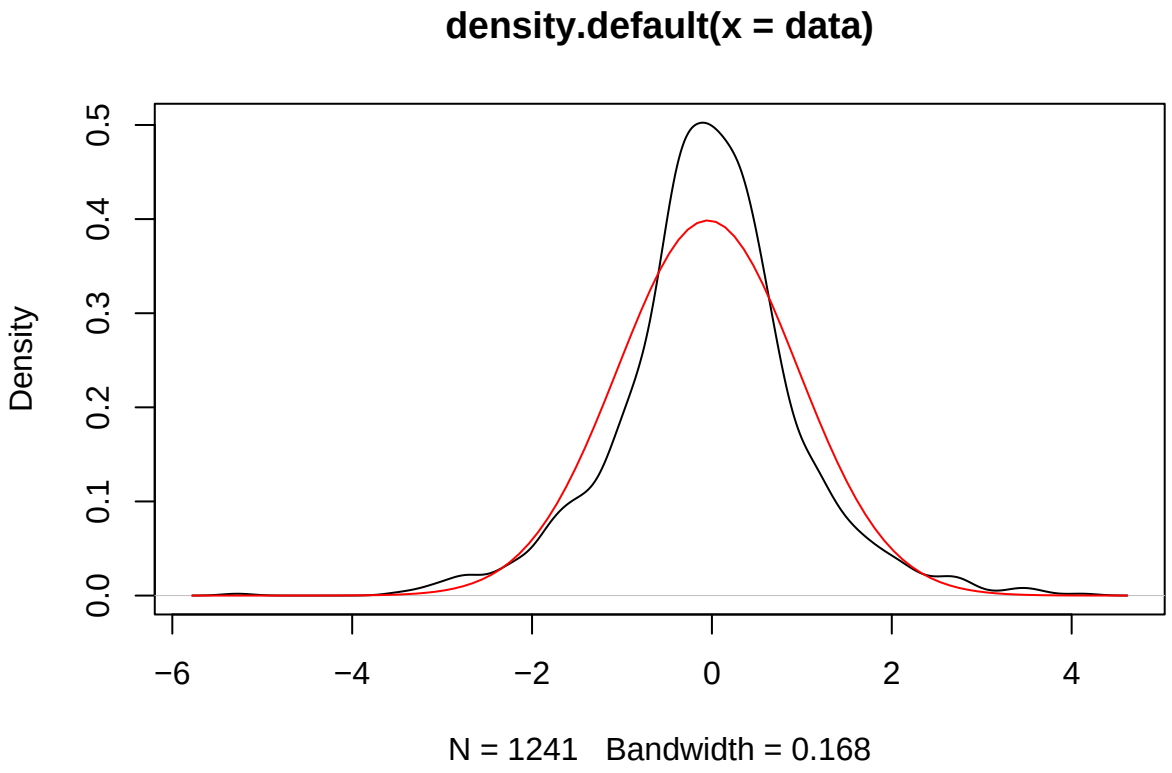
Parametric bootstrapping is on the other hand used when we want to know whether or not it is 'safe' to assume that a dataset can be decribed by a certain distribution.

The dataset `arctic.oscillations` (in package UsingR) contains a time series from January to June 2002 of sea-level pressure measurement at the arctic, relative to some base line. Use parametric bootstrap to judge whether it is safe to assume that the measurements are samples from normal distribution or not.

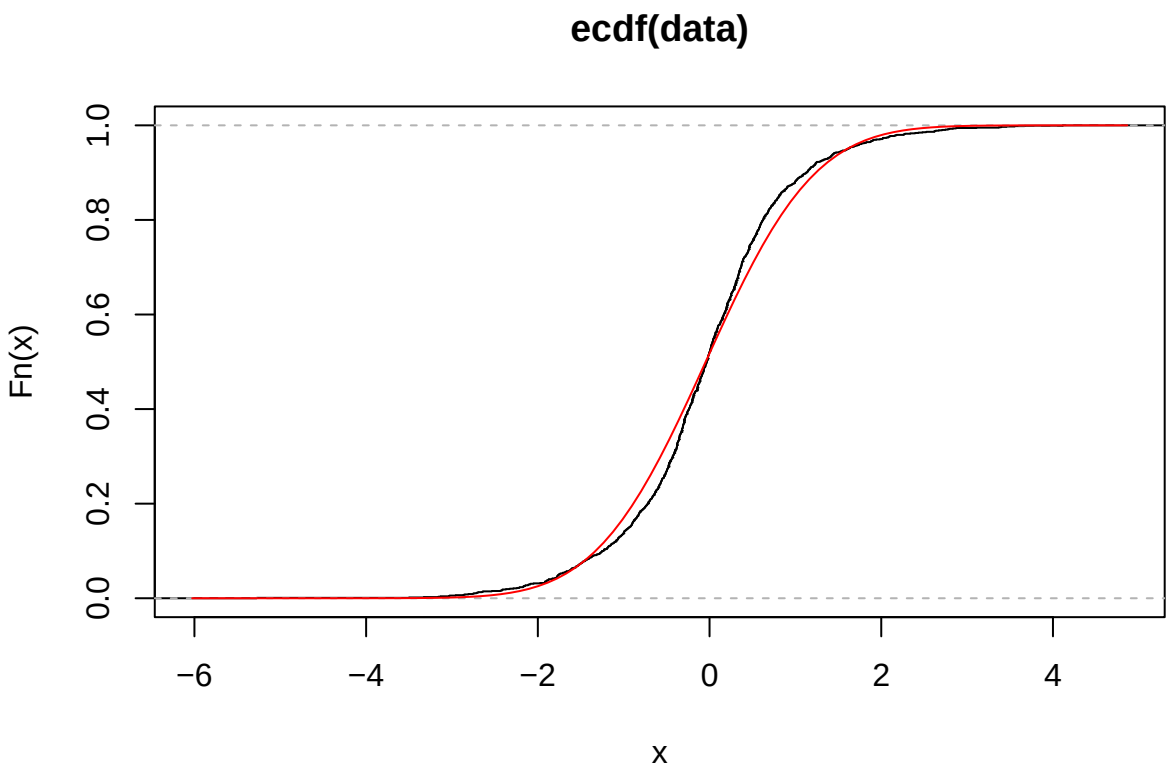
```
data <- na.omit(arctic.oscillations)

sample_mean <- mean(data)
sample_sd <- sd(data)

xs <- seq(from = min(data), max(data), length.out = 100)
plot(density(data))
curve(dnorm(x, mean = sample_mean, sample_sd), col = "red", add = T)
```



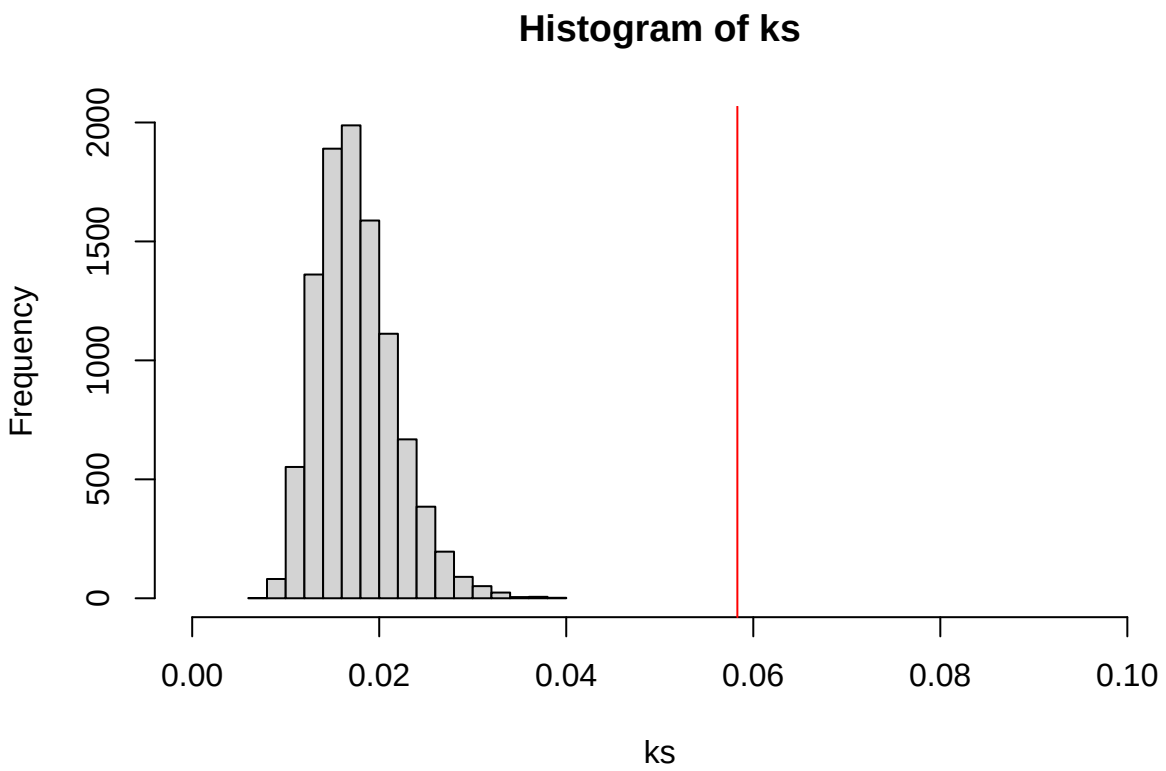

```
plot(ecdf(data))
curve(pnorm(x, mean = sample_mean, sample_sd), col = "red", add = T)
```




```r
# now we calculate the Kolmogorov-Smirnov distance
ks_dist_norm <- function(data) {
  emp_dist <- ecdf(data)
  max(abs(emp_dist(data) - pnorm(data, mean(data), sd = sd(data))))
}
ks_estimate <- ks_dist_norm(data)
m <- 10 ** 4
n <- length(data)

ks <- c()
for (i in 1:m) {
  bootstrap_sample <- rnorm(n, mean = sample_mean, sd = sample_sd)
  ks <- c(ks, ks_dist_norm(bootstrap_sample))
}

hist(ks, xlim = c(0, 0.1))
abline(v = ks_estimate, col = "red")
```



What is the probability that this collection of samples could have been drawn from that probability distribution?