The Law of Large Numbers & The Central Limit Theorem

Ulrich Matter

22 February 2017

The Law of Large Numbers

The Law of Large Numbers (LLN) is an important statistical property which essentially describes how the behavior of sample averages is related to sample size. Particularly, it states that the sample mean can come as close as we like to the mean of the population from which it is drawn by simply increasing the sample size. That is, the larger our randomly selected sample from a population, the closer is that sample's mean to the mean of the population.

Think of playing dice. Each time we roll a fair die, the result is either 1, 2, 3, 4, 5, or 6, whereby each possible outcome can occur with the same probability (1/6). In other words we randomly draw die-values. Thus we can expect that the average of the resulting die values is $(1 + 2 + 3 + 4 + 5 + 6)/6 = 3.5$.

We can investigate this empirically: We roll a fair die once and record the result. We roll it again, and again we record the result. We keep rolling the die and recording results until we get 100 recorded results. Intuitively, we would expect to observe each possible die-value about equally often (given that the die is fair) because each time we roll the die, each possible value (1,2,..,6) is equally likely to be the result. And we would thus expect the average of the resulting die values to be around 3.5. However, just by chance it can obviously be the case that one value (say 5) occurs slightly more often than another value (say 1), leading to a sample mean slightly larger than 3.5. In this context, the LLN states that by increasing the number of times we are rolling the die, we will get closer and closer to 3.5. Now, let's implement this experiment in R:

```r
# first we define the potential values a die can take
dvalues <- 1:6 # the : operater generates a regular sequence of numbers (from:to)
dvalues
```

```
## [1] 1 2 3 4 5 6
```

```r
# define the size of the sample n (how often do we roll the die...)
# for a start, we only roll the die ten times
n <- 10
# draw the random sample: 'roll the die n times and record each result'
results <- sample( x = dvalues, size = n, replace = TRUE)
# compute the mean
mean(results)
```

```
## [1] 3.1
```

As you can see we are relatively close to 3.5, but not quite there. So let's roll the die more often and calculate the mean of the resulting values again:

```r
n <- 100
# draw the random sample: 'roll the die n times and record each result'
results <- sample( x = dvalues, size = n, replace = TRUE)
# compute the mean
mean(results)
```

```
## [1] 3.4
```

We are getting closer to 3.5! Now let's scale up these comparisons and show how the sample means are getting closer to 3.5 when increasing the number of times we roll the die up to 50'000.

```r
# Essentially, what we are doing here is repeating the experiment above many times, each time increasin
# define the set of sample sizes
ns <- seq(from = 10, to = 50000, by = 10)
# initiate an empty list to record the results
means <- list()
length(means) <- length(ns)
# iterate through each sample size: 'repeat the die experiment for each sample size'
for (i in 1:length(ns)) {

    means[[i]] <- mean(sample( x = dvalues, size = ns[i], replace = TRUE))
}

# visualize the result: plot sample means against sample size
plot(ns, unlist(means),
     ylab = "Sample Mean",
     xlab = "Sample Size",
     pch = 16,
     cex = .6)
abline(h = 3.5, col = "red")
```
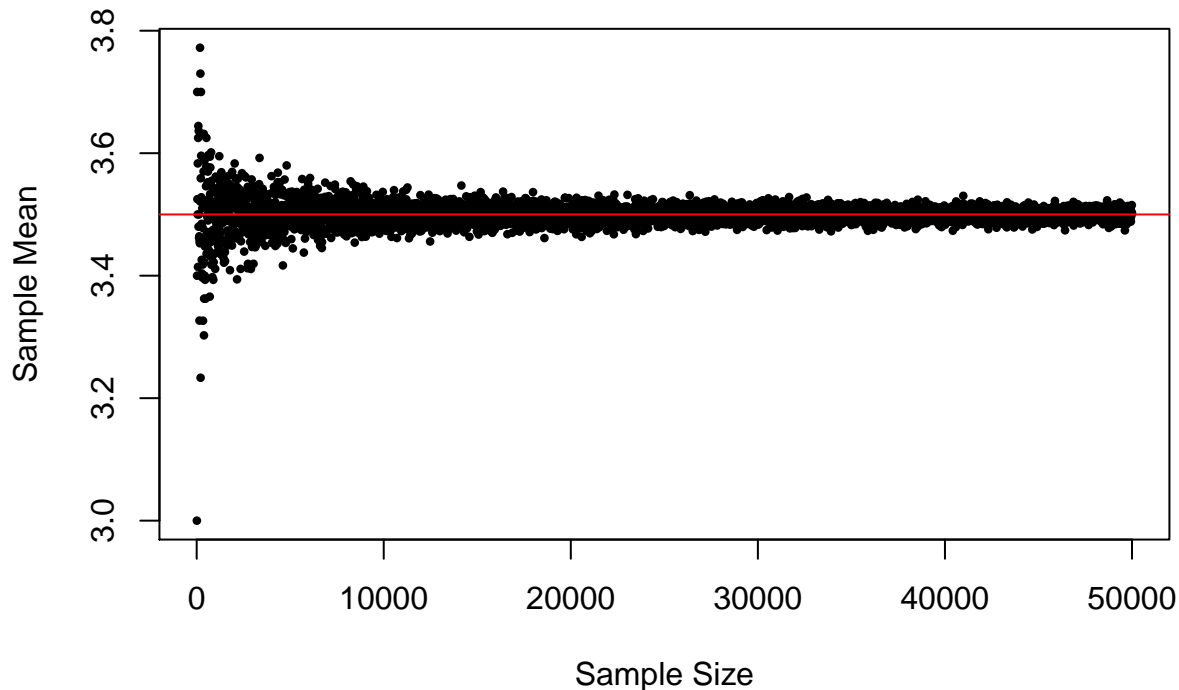

We observe that with smaller sample sizes the sample means are broadly spread around the population mean of 3.5. However, the more we go to the right extreme of the x-axis (and thus the larger the sample size), the narrower the sample means are spread around the population mean.

The Central Limit Theorem

The Central Limit Theorem (CLT) is an almost miraculous statistical property enabling us to test the statistical significance of a statistic such as the mean. In essence, the CLT states that as long as we have a large enough sample, the t-statistic (applied, e.g., to test whether the mean is equal to a particular value)

is approximately standard normal distributed. This holds independently of how the underlying data is distributed.

Consider the dice play example above. We might want to statistically test whether we are indeed playing with a fair die (which would imply an expected mean value of 3.5!). In order to test that we would roll the die 100 times and record each resulting value. We would then compute the sample mean and standard deviation in order to assess how likely it was to observe the mean we observe if the population mean actually is 3.5 (thus our H0 would be pop_mean = 3.5, or in plain English 'the die is fair'). However, the distribution of the resulting die values are not standard normal distributed. So how can we interpret the sample standard deviation and the sample mean in the context of our hypothesis?

The simplest way to interpret these measures is by means of a *t-statistic*. A t-statistic for the sample mean under our working hypothesis that pop_mean = 3.5 is constructed as `t(3.5) = (sample_mean - 3.5) / (sample_sd/sqrt(n))`. Let's illustrate this in R:

```r
# First we roll the die like above
n <- 100
# draw the random sample: 'roll the die n times and record each result'
results <- sample( x = dvalues, size = n, replace = TRUE)
# compute the mean
sample_mean <- mean(results)
# compute the sample sd
sample_sd <- sd(results)
# estimated standard error of the mean
mean_se <- sample_sd/sqrt(n)

# compute the t-statistic:
t <- (sample_mean - 3.5) / mean_se
t
```

```
## [1] -1.312481
```

At this point you might wonder what the use of t is if the underlying data is not drawn from a normal distribution. In other words: what is the use of t if we cannot interpret it as a value that tells us how likely it is to observe this sample mean, given our null hypothesis? Well, actually we can. And here is where the magic of the CLT comes in: It turns out that there is a mathematical proof (i.e. the CLT) which states that the t-statistic itself can arbitrarily well be approximated by the standard normal distribution. This is true independent of the distribution of the underlying data in our sample! That is, if we have a large enough sample, we can simply compute the t-statistic and look up how likely it is to observe a value at least as large as t, given the null hypothesis is true (-> the p-value):

```r
# calculate the p-value associated with the t-value calculated above
2*pnorm(-abs(t))
```

```
## [1] 0.189358
```

In that case we could not reject the null hypothesis of a fair die. However, as pointed out above, the t-statistic is only assymptotically (meaning with very large samples) normally distributed. We might not want to trust this hypothesis test too much because we were using a sample of only 100 observations.

Let's turn back to R in order to illustrate the CLT at work. Similar to the illustration of the LLN above, we will repeatedly compute the t-statistic of our dice play experiment and for each trial increase the number of observations.

```r
# define the set of sample sizes
ns <- c(10, 40, 100)
# initiate an empty list to record the results
ts <- list()
length(ts) <- length(ns)
```

```r
# iterate through each sample size: 'repeat the die experiment for each sample size'
for (i in 1:length(ns)) {

    samples.i <- sapply(1:500000, function(j) sample( x = dvalues, size = ns[i], replace = TRUE))
    ts[[i]] <- apply(samples.i, function(x) (mean(x) - 3.5) / sd(x), MARGIN = 2)
}

# visualize the result: plot the density for each sample size

# plot the density for each set of t values
hist(ts[[1]], main = "Sample size: 10", xlab = "T-value")
```
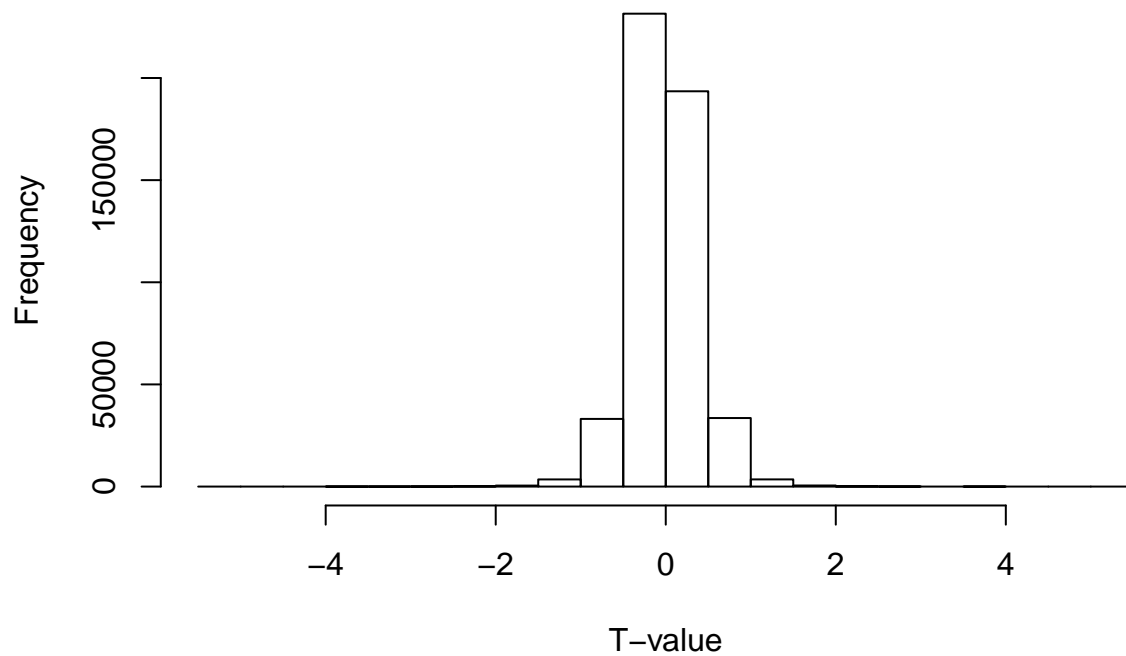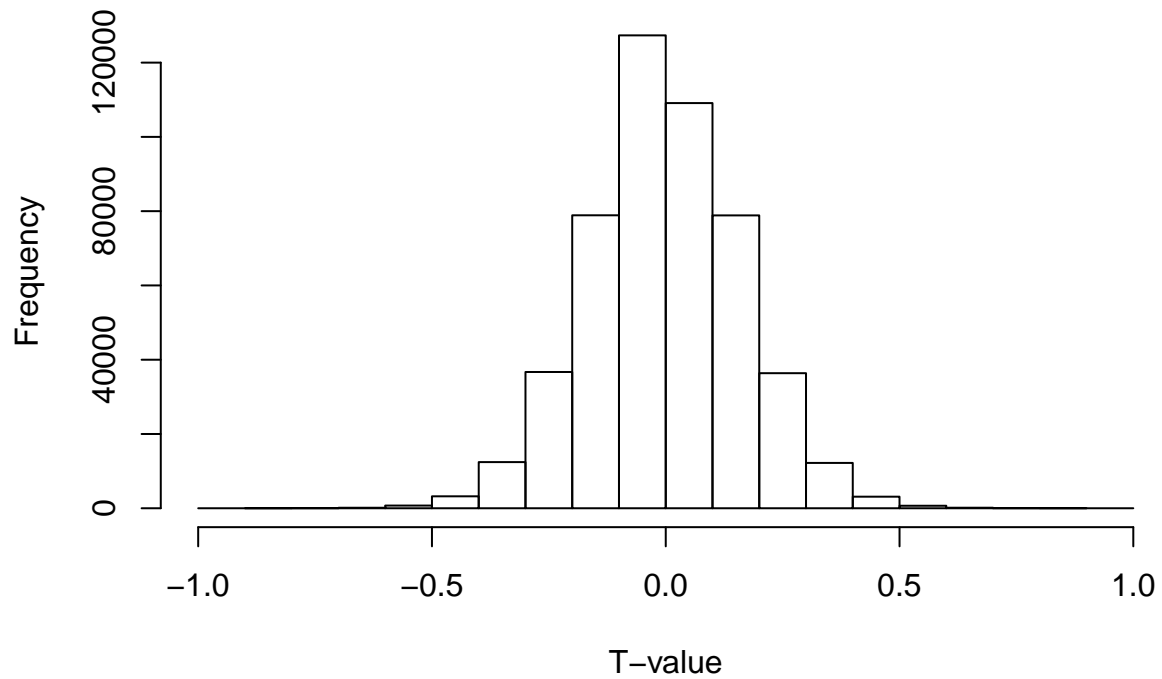
Sample size: 10


```r
hist(ts[[2]], main = "Sample size: 40", xlab = "T-value")
```
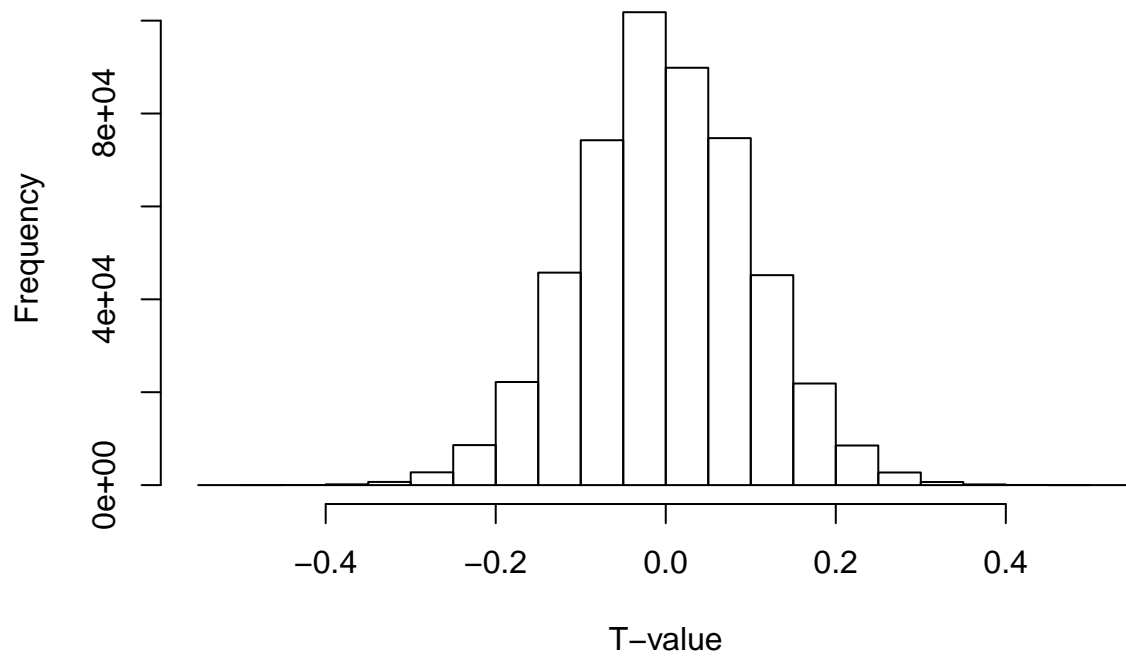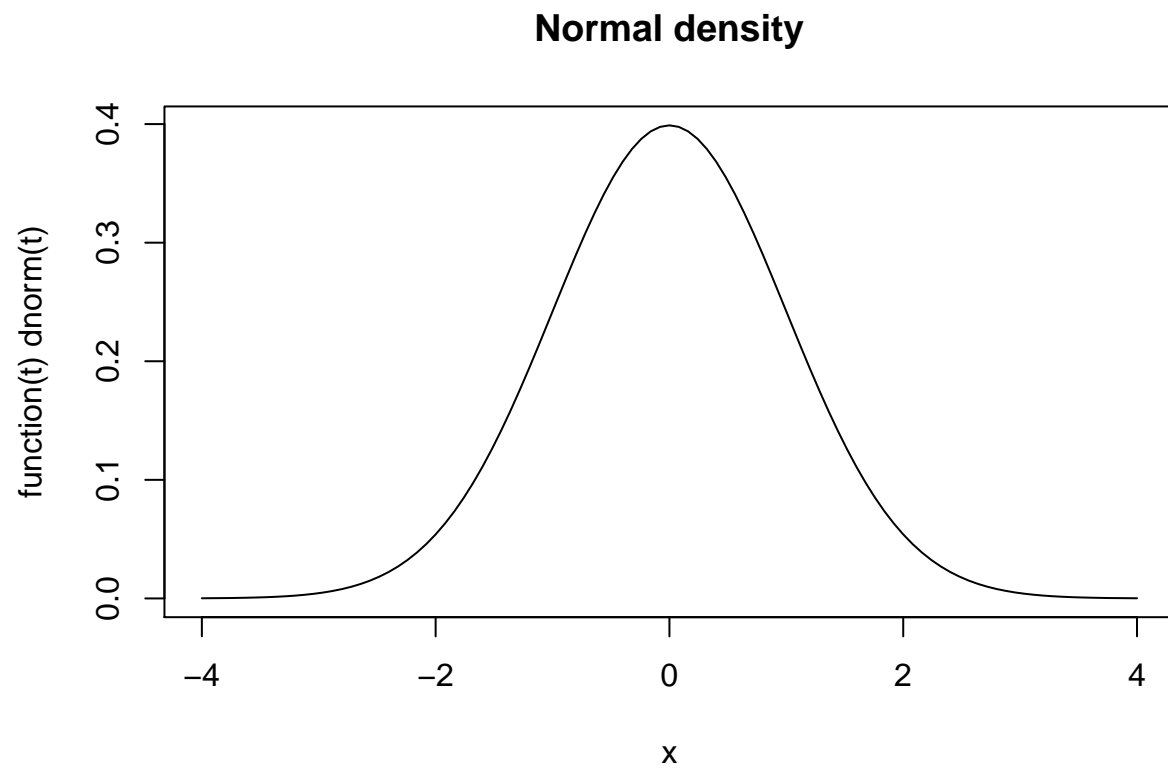
Sample size: 40


```
hist(ts[[3]], main = "Sample size: 100", xlab = "T-value")
```

Sample size: 100


```
# finally have a look at the actual standard normol distribution as a reference point
plot(function(t)dnorm(t), -4, 4, main = "Normal density")
```

Normal density

Note how the histogram is getting closer to a normal distribution with increasing sample size.