The law of large numbers

Chester Wang

2025-05-27

```
library(tidyverse)
library(reshape)
library(Hmisc)
```

Introduction

The law of large numbers describes that, as the sample sizes increases, the sample means will get closer to the expected value. In order to understand how the law of large numbers works, we need to know what expected values of sampling means first.

Expected values of sampling

The expected value of a discrete random variable is the weighted average of its possible outcomes, which equals to the sum of each outcome times the probability this outcome to happen.

Suppose we have a six-sided die. It is considered a fair die if the probability of each side being rolled is equal. In other words, each side of a fair six-sided die has a 1/6 or 16.67% chance of being rolled. In this case, the expected value of rolling a fair six-sided die is 3.5:

```
die <- data.frame(n = seq(1, 6, 1))
expected_value <- sum(die * 1/6)
print(paste0("The expected value of rolling a fair six-sided die is ",
             expected_value))
```

```
## [1] "The expected value of rolling a fair six-sided die is 3.5"
```

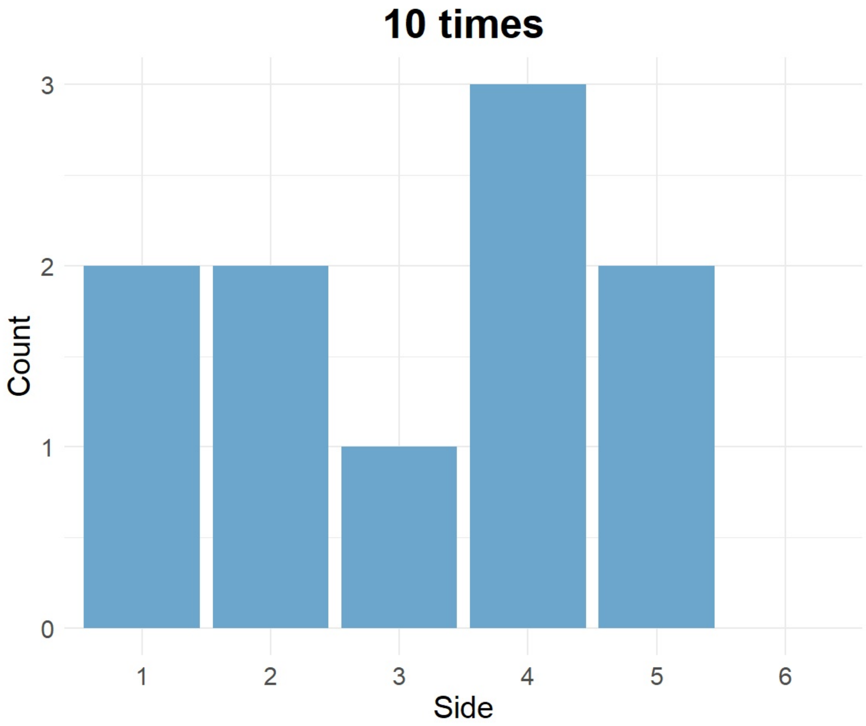
Simulation and Results

To see how the law of large numbers works in practice, we simulate die-rolling repeatedly for 10, 100, 1000, 10000, and 100000 times, and compare the probabilities of each side being rolled and the resulting sample means for each simulation. According to the law of large numbers, with increasing times of rolling a die repeatedly, the sample means are expected to get closer to the expected value 3.5, and the probability of each side being rolled to be closer to 16.67%.

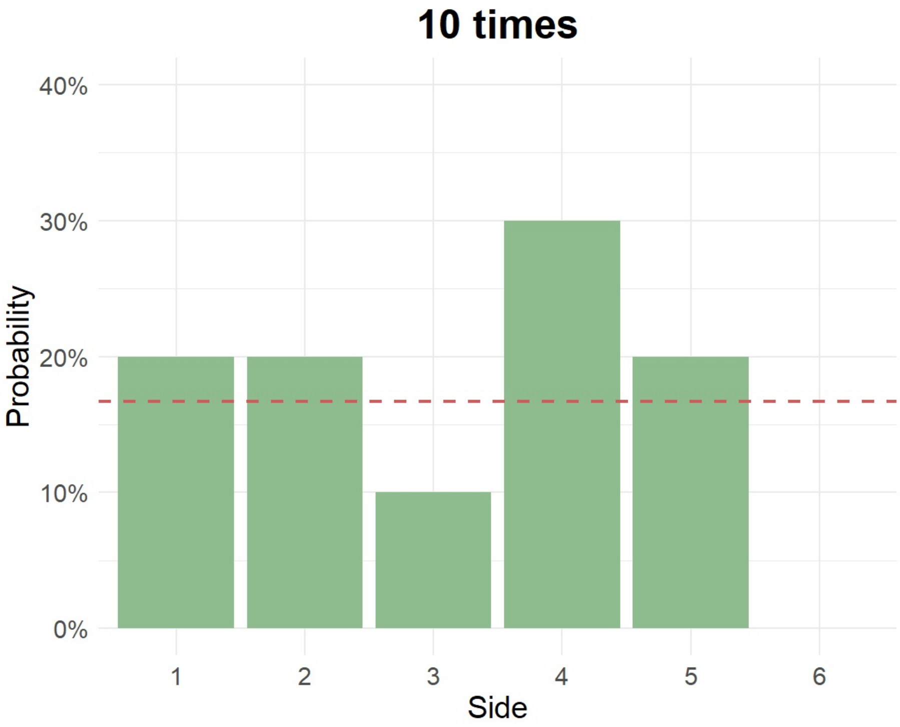
```r
die_rolling <- function(repeat_times) {
  set.seed(345)
  roll_n_times <- die %>%
    sample_n(repeat_times, replace = TRUE) %>%
    group_by(n) %>%
    dplyr::summarize(count = n()) %>%
    mutate(probability = count / sum(count)) %>%
    mutate(n_fct = factor(n,
                          levels = c(1, 2, 3, 4, 5, 6)))
  colnames(roll_n_times)[2:3] <-
    c(paste0("count_", format(repeat_times, scientific = F)),
      paste0("probability_", format(repeat_times, scientific = F)))
  return(roll_n_times)
}

count_plot <- function(roll_n_times) {
  ggplot(roll_n_times, aes_string(x = names(roll_n_times[4]),
                                  y = names(roll_n_times[2]))) +
    geom_col(fill = "skyblue3") +
    theme_minimal() +
    ggtitle(paste0(format(repeat_times, scientific = F), " times")) +
    xlab("Side") +
    ylab("Count") +
    theme(plot.title = element_text(size = 20, face = "bold", hjust = .5),
          axis.title = element_text(size = 15),
          axis.text = element_text(size = 12),
          aspect.ratio = .75) +
    scale_x_discrete(drop = FALSE)
}

prob_plot <- function(roll_n_times) {
  ggplot(roll_n_times, aes_string(x = names(roll_n_times[4]),
                                  y = names(roll_n_times[3]))) +
  geom_col(fill = "darkseagreen") +
  theme_minimal() +
  ggtitle(paste0(format(repeat_times, scientific = F), " times")) +
  xlab("Side") +
  ylab("Probability") +
  theme(plot.title = element_text(size = 20, face = "bold", hjust = .5),
        axis.title = element_text(size = 15),
        axis.text = element_text(size = 12),
        aspect.ratio = .75) +
  scale_x_discrete(drop = FALSE) +
  scale_y_continuous(labels = scales::percent, limits = c(0, 0.4)) +
  geom_hline(yintercept = 1/6, color = "indianred",
             linetype = "dashed", linewidth = 0.75)
}

sample_mean <- function(roll_n_times) {
  sum(roll_n_times[1] * roll_n_times[3])
}
```

Simulation 1: Rolling a die 10 times

```r
repeat_times <- 10
count_plot_10 <- count_plot(die_rolling(repeat_times))
count_plot_10
```



```
prob_plot_10 <- prob_plot(die_rolling(repeat_times))
prob_plot_10
```



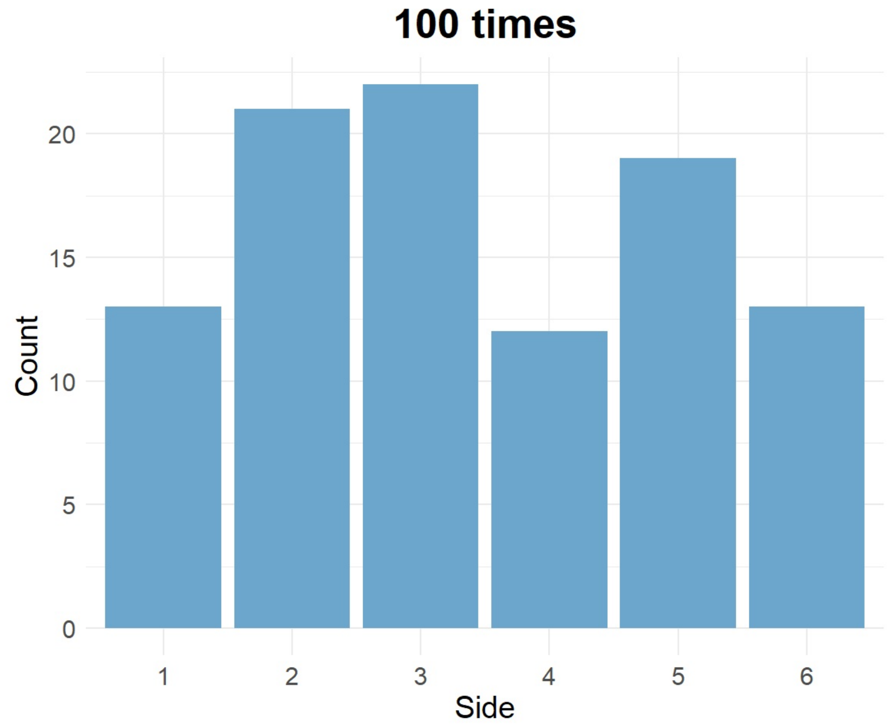
```
sample_mean_10 <- sample_mean(die_rolling(repeat_times))
print(paste0("The sample mean of rolling a die ",
             format(repeat_times, scientific = F),
             " times is ",
             sample_mean_10))
```

```
## [1] "The sample mean of rolling a die 10 times is 3.1"
```

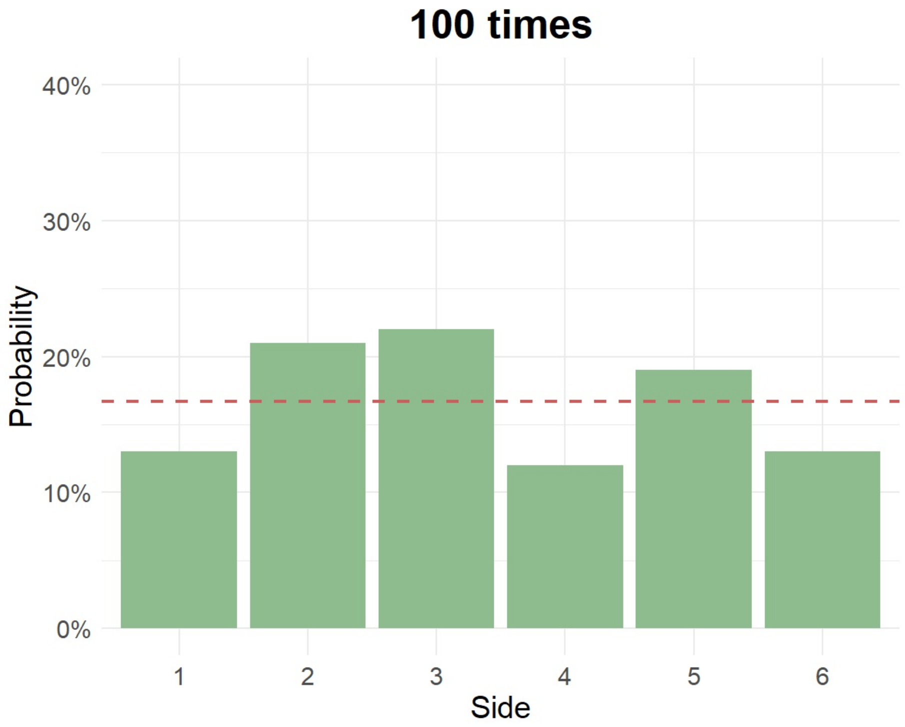
The sample mean of rolling a die repeatedly 10 times is 3.1. It is observed that the probability of each side being rolled varies, ranging from 10% to 30%, with side numbered 6 shows 0 appearance in this simulation.

Simulation 2: Rolling a die 100 times

```
repeat_times <- 100
count_plot_100 <- count_plot(die_rolling(repeat_times))
count_plot_100
```



```
prob_plot_100 <- prob_plot(die_rolling(repeat_times))
prob_plot_100
```



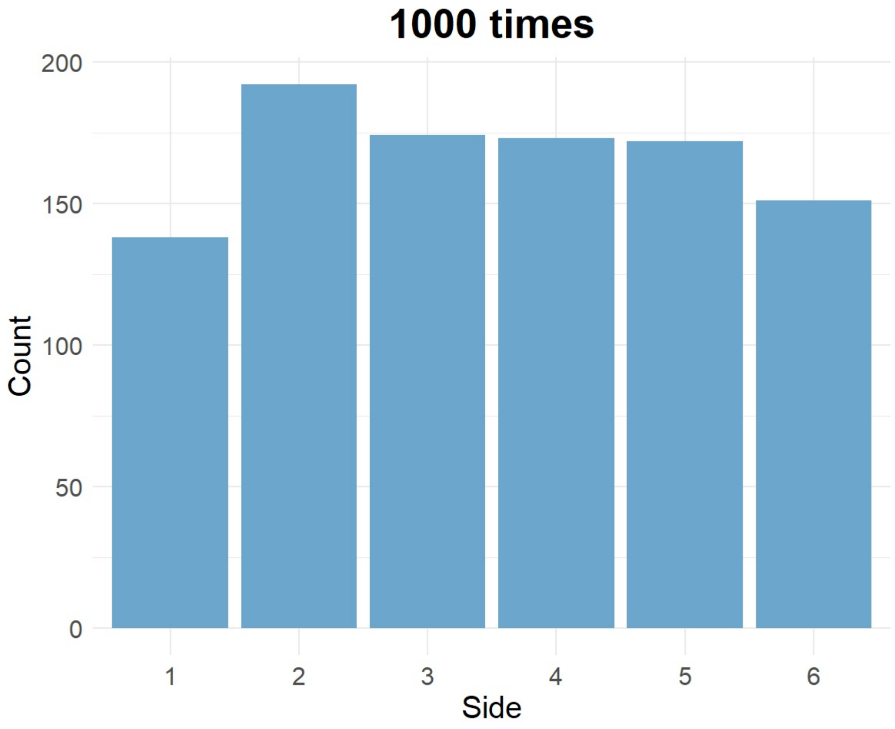
```
sample_mean_100 <- sample_mean(die_rolling(repeat_times))
print(paste0("The sample mean of rolling a die ",
             format(repeat_times, scientific = F),
             " times is ",
             sample_mean_100))
```

```
## [1] "The sample mean of rolling a die 100 times is 3.42"
```

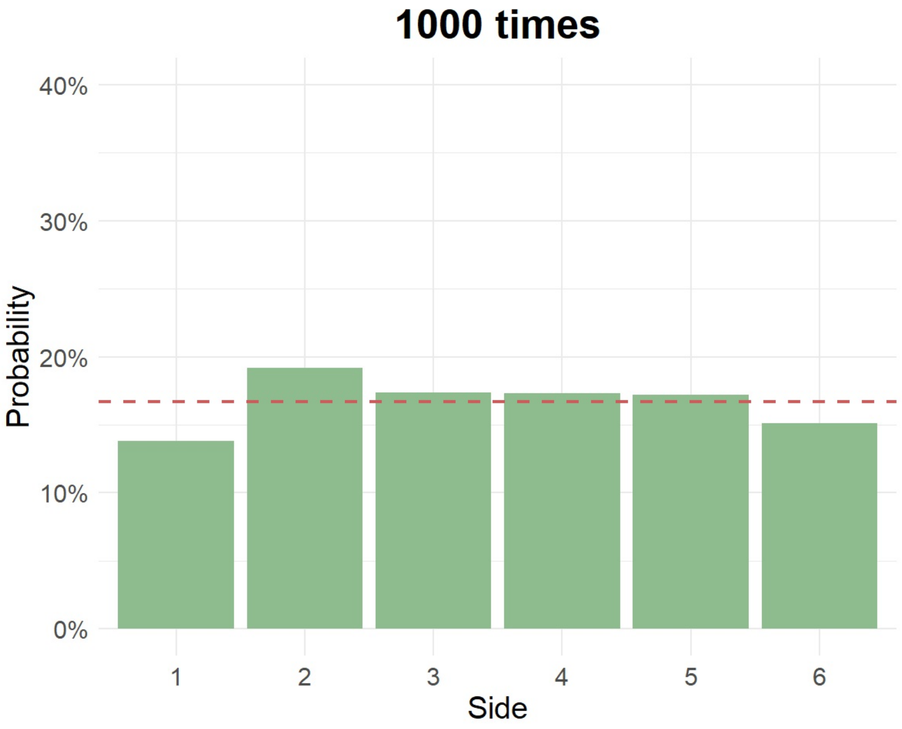
The sample mean of rolling a die repeatedly 100 times is 3.42, which is closer to the expected value 3.5 when compared with the result of Simulation 1. While variations of probabilities of each side being rolled is still observed, the differences decrease and are closer to the theoretical value 16.67%.

Simulation 3: Rolling a die 1000 times

```
repeat_times <- 1000
count_plot_1000 <- count_plot(die_rolling(repeat_times))
count_plot_1000
```



```
prob_plot_1000 <- prob_plot(die_rolling(repeat_times))
prob_plot_1000
```



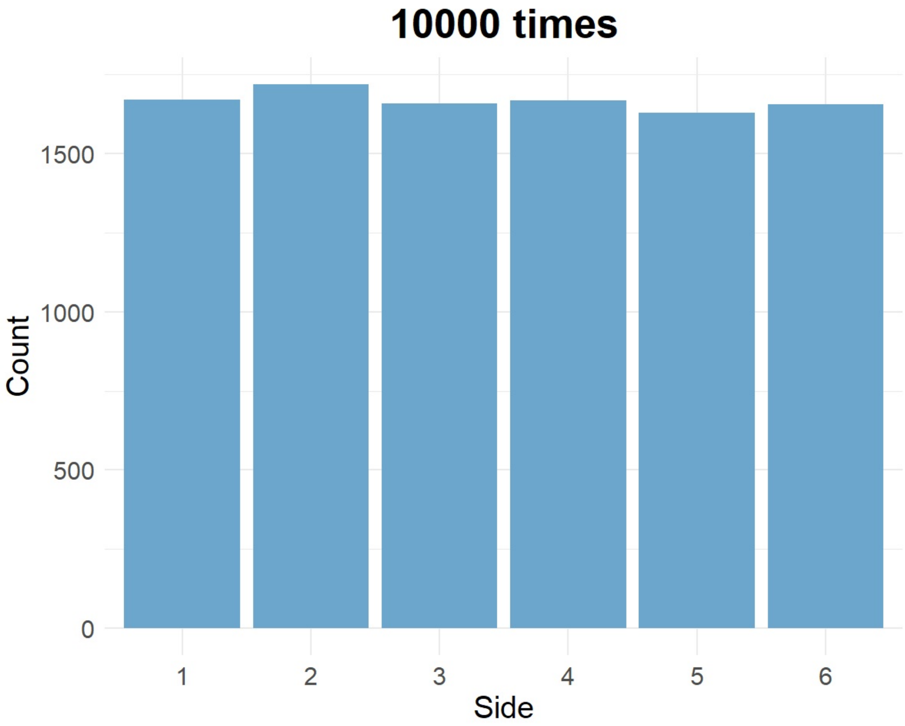
```
sample_mean_1000 <- sample_mean(die_rolling(repeat_times))
print(paste0("The sample mean of rolling a die ",
             format(repeat_times, scientific = F),
             " times is ",
             sample_mean_1000))
```

```
## [1] "The sample mean of rolling a die 1000 times is 3.502"
```

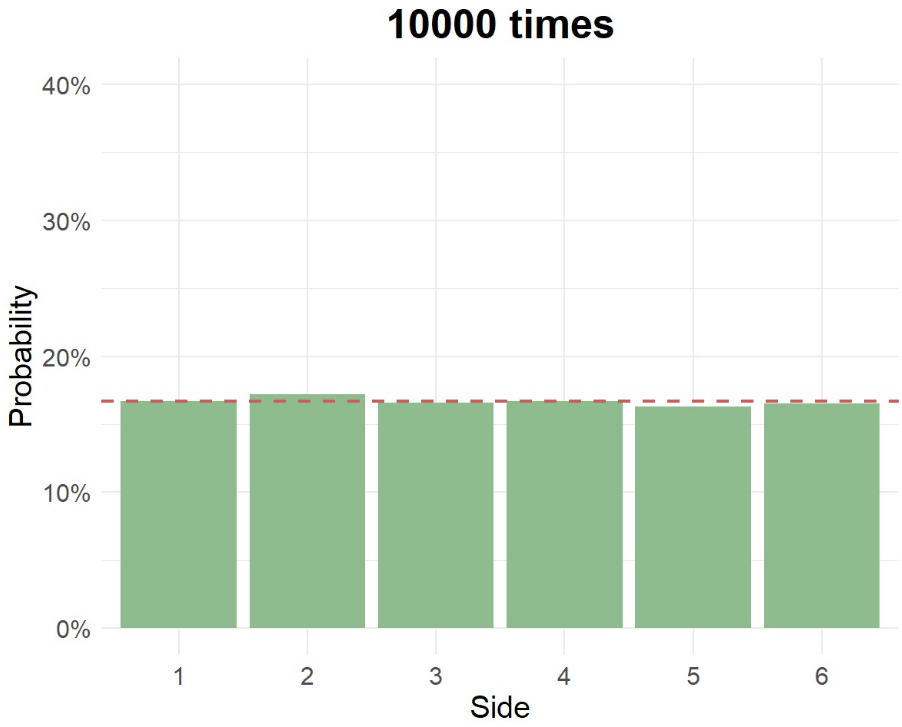
The sample mean of rolling a die repeatedly 1000 times is 3.502, which is very close to the expected value 3.5. The probabilities of each side being rolled are even more closer to the theoretical value when compared to the previous two simulations.

Simulation 4: Rolling a die 10000 times

```
repeat_times <- 10000
count_plot_10000 <- count_plot(die_rolling(repeat_times))
count_plot_10000
```



```
prob_plot_10000 <- prob_plot(die_rolling(repeat_times))
prob_plot_10000
```



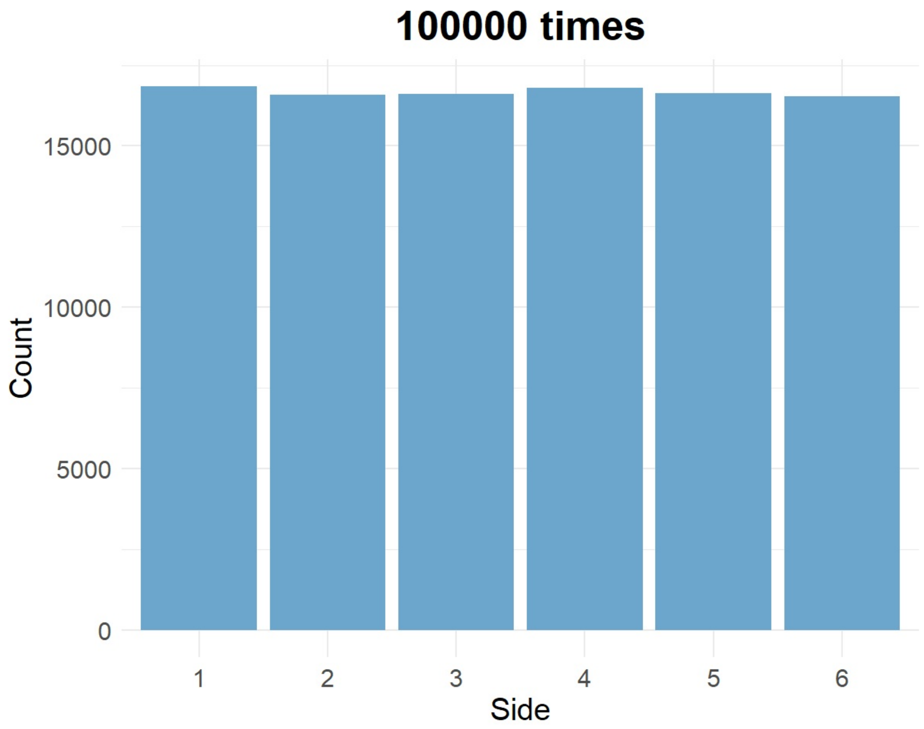
```
sample_mean_10000 <- sample_mean(die_rolling(repeat_times))
print(paste0("The sample mean of rolling a die ",
             format(repeat_times, scientific = F),
             " times is ",
             sample_mean_10000))
```

```
## [1] "The sample mean of rolling a die 10000 times is 3.4833"
```

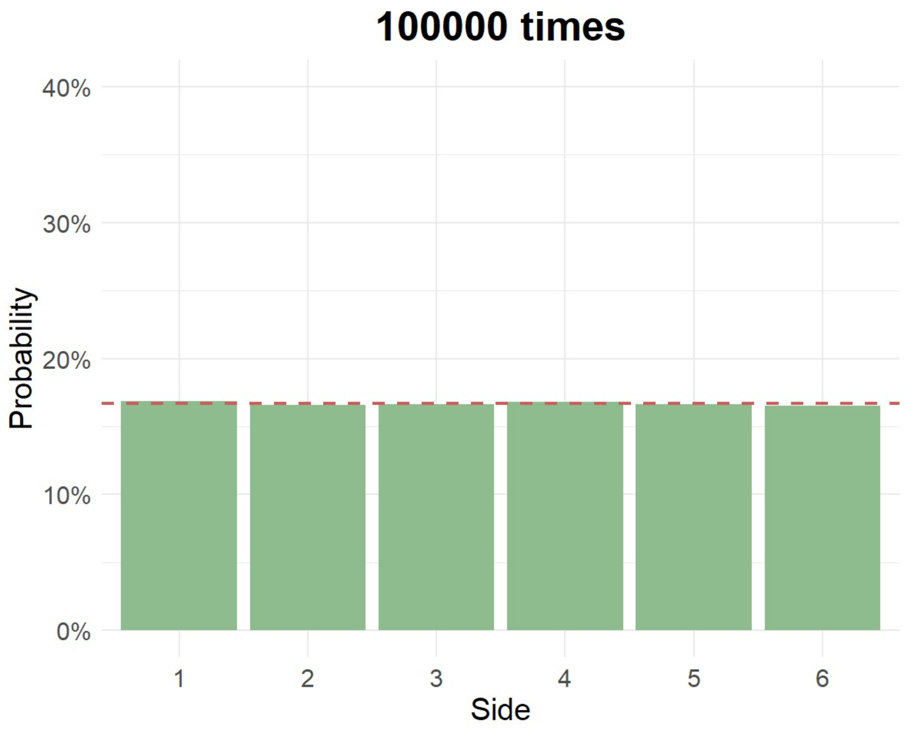
The sample mean of rolling a die repeatedly 10000 times is 3.4833, which shows a larger difference to the expected value when compared to the previous simulation. Nevertheless, distribution of the possibilities of each side facing up is more consistent then those in the previous simulations.

Simulation 5: Rolling a die 100000 times

```
repeat_times <- 100000
count_plot_100000 <- count_plot(die_rolling(repeat_times))
count_plot_100000
```



```
prob_plot_100000 <- prob_plot(die_rolling(repeat_times))
prob_plot_100000
```



```
sample_mean_100000 <- sample_mean(die_rolling(repeat_times))
print(paste0("The sample mean of rolling a die ",
             format(repeat_times, scientific = F),
             " times is ",
             sample_mean_100000))
```

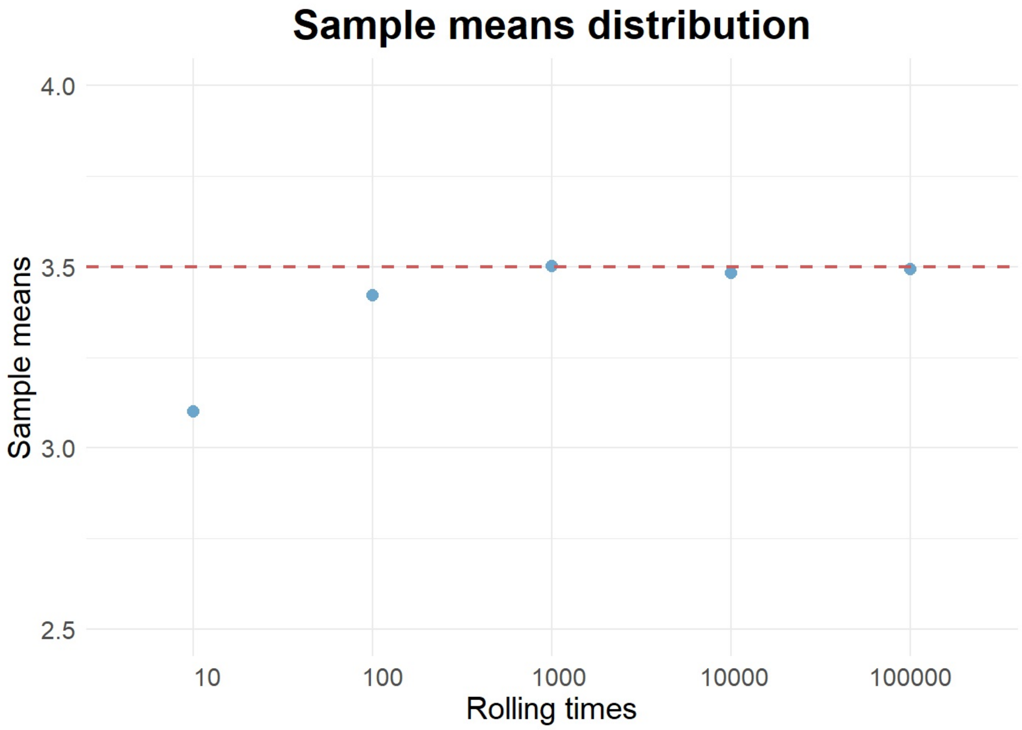
```
## [1] "The sample mean of rolling a die 100000 times is 3.49373"
```

The sample mean of rolling a die repeatedly 100000 times is 3.49373, whcih converges to the expected value again compared to the last simulation. The probabilities of each side being rolled are very close to the theoretical value of a fair six-sided die.

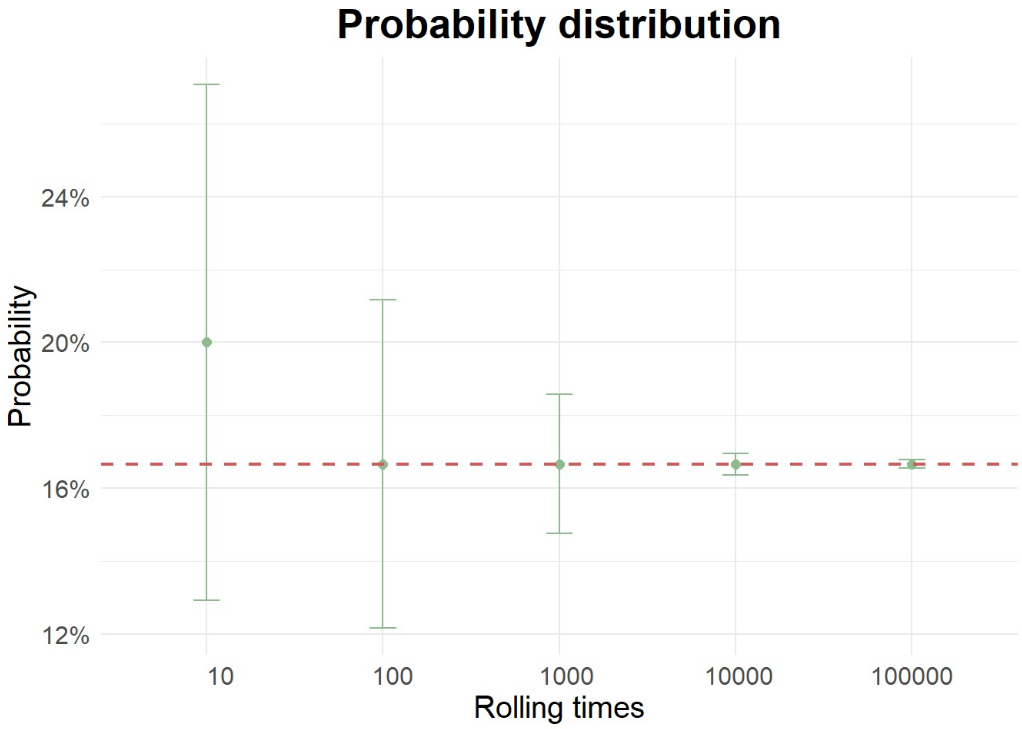
Discussion

At the beginning of this article, we mentioned that with increasing sample sizes, the sample means become closer to the expected value, but how large of sample size should be for this law to work? We simulated by rolling a die repeatedly for differnt times, starting at 10 times and a ten-fold increase each time for total of five simulations.

```
repeat_times <- c(10, 100, 1000, 10000, 100000)
sample_means <- c(sample_mean_10, sample_mean_100, sample_mean_1000,
                  sample_mean_10000, sample_mean_100000)
sample_means_distribution <- data.frame(format(repeat_times, scientific = F),
                                        sample_means)

ggplot(sample_means_distribution, aes(x = format(repeat_times, scientific = F),
                                      y = sample_means)) +
  geom_point(size = 2.5, col = "skyblue3") +
  theme_minimal() +
  ggtitle("Sample means distribution") +
  xlab("Rolling times") +
  ylab("Sample means") +
  ylim(c(2.5, 4)) +
  theme(plot.title = element_text(size = 20, face = "bold", hjust = .5),
        axis.title = element_text(size = 15),
        axis.text = element_text(size = 12)) +
  geom_hline(yintercept = 3.5, color = "indianred",
             linetype = "dashed", linewidth = 0.75)
```



```
prob_distribution <- data.frame(n_fct = factor(seq(1, 6),
                                               levels = c(1, 2, 3, 4, 5, 6)))
for(i in repeat_times) {
   prob_distribution <- full_join(prob_distribution, die_rolling(i)[, 2:4],
                                  by = "n_fct")
}
prob_distribution <- select(prob_distribution, contains("probability")) %>%
  melt()

ggplot(prob_distribution, aes(x = variable, y = value)) +
  stat_summary(fun = mean, geom = "point", size = 2, col = "darkseagreen") +
  stat_summary(fun.data = mean_sdl,
               fun.args = list(mult = 1),
               geom = "errorbar",
               col = "darkseagreen", width = 0.15) +
  theme_minimal() +
  ggtitle("Probability distribution") +
  xlab("Rolling times") +
  ylab("Probability") +
  theme(plot.title = element_text(size = 20, face = "bold", hjust = .5),
        axis.title = element_text(size = 15),
        axis.text = element_text(size = 12)) +
  scale_y_continuous(labels = scales::percent) +
  scale_x_discrete(breaks = unique(prob_distribution$variable),
        labels = format(repeat_times, scientific = F)) +
  geom_hline(yintercept = 1/6, color = "indianred",
             linetype = "dashed", linewidth = 0.75)
```



In these simulations, a trend is observed that, by increasing the times of die-rolling repeatedly, the sample means become closer to the expected value 3.5, and the probabilities of each side being rolled are closer to 16.67% with decreasing variations.

It was observed that, while the sample means seemed to converge to the expected value in the first three simulations, it diverged when repetition was further increased in Simulation 4. Based on the probability distribution of Simulation 3, the sample mean closer to the expected value than Simulation 4 was offset by the larger variations of probabilities between different sides. In other words, 1000 times of die-rolling was not a large enough sample size in our simulations.

In the last two simulations, both the sample means were closer to the expected value and the probabilities of each side being rolled were approaching a uniform distribution. Therefore, according to our simulations, a sample size of at least 10000 would be considered large enough for the law of large numbers to apply.

Conclusion

While there is no definitions for sample sizes to be considered “large” and the number is subjected to vary depending on the context, a sample size of 10000 in die-rolling could be considered large in our simulations for the law of large numbers to work.