Assignment I: Stochastic Simulation

First, some general comments regarding the report **that apply to both assignments**. The report is a document that shows *that you have understood and carried out the assignment*. This means that, in addition to formulas, R commands, and output, there is *also connecting text*.

Include not only the code you write, but also the relevant *(intermediate) results they produce*. Ignoring the above instructions will affect your grade. The easiest thing to do is probably to make an RMarkdown file, which can be compiled to either pdf or html format. For this option see the end of Chapter 1 of `https://probstatsdata.com`.

Simulation using R

The assignment assumes that you have already done a short introductory tour through R (e.g. see `https://probstatsdata.com`). In this online book (which is written in RMarkdown) you can also find instructions to installation of R and (optionally) RStudio, which is an advanced user interface. R is specifically aimed at statistics and is widely regarded as one of the most important statistical software packages.

Random numbers can be obtained in R with the command `runif`. For example, `runif(3)` generates 3 random numbers.

Example

We check how 'good' the random number generator `runif` does its job. To do this, we do a large number of draws and make a histogram. If it is 'good', the histogram resembles the underlying probability density function (which in this case is uniform on the interval $[0, 1]$). [1]

A Create three histograms side by side. The leftmost one is a histogram with 50 draws from the uniform distribution on $[0, 1]$. You can generate this histogram as

[1] See Section 4.2.1 of the textbook.

follows:

```
x <- runif(50)
hist(x, main='Histogram of 50 draws')
```

Create the middle histogram based on 500 draws, and the rightmost histogram based on 5000 draws. To get the three histograms in one figure, you can use the command `par(mfrow=c(1,3))` before creating the 3 histograms.

Throwing a coin and the arcsin law

In this assignment we will simulate a simple probabilistic experiment. The experiment goes as follows:

- We flip a coin: if the outcome is heads, we receive 1 euro, if the outcome is tails, we pay 1 euro.
- We start with 0 euros (if we have to pay, our balance becomes negative).
- We repeat this experiment 10 times a day for a year, so in total 3650 times.

Let S be the random variable that equals -1 when flipping tails, and 1 when flipping heads. We assume the coin is fair.

B Think about how you can simulate the random variable S using a uniformly distributed random variable U. Write a function `sim.coin(n)`, that generates n realizations of the random variable S. The command `ifelse` can be useful for this.

C Over a year, we flip 3650 times. We can simulate the development of our balance using `cumsum(sim.coin(n))`. Create a figure in which the vertical axis shows the balance against the horizontal axis showing the numbers 1 to 3650. Provide the graph with an appropriate title and axis labels.

D Calculate the fraction of time that we have a positive balance over a year. We expect that the balance is positive about half the time and negative the other half. Is this the case in your generated realization?

E A graph where the balance is positive only 5 percent of the time seems highly unlikely. Make a guess at the probability of this happening.

F To get a better intuition for the situation, we add three more simulations. Create a graph with the total of four simulations. Use the command `lines` to add the simulations. Ensure that the y-axis scale is such that you can clearly see all four simulations.

G We repeat the entire experiment 1000 times and calculate the fraction of time that the balance is positive each year. Store these fractions in a vector `pos.vec`. This can be done with a for-loop. Create a histogram of the values in the vector `pos.vec`. Use the option `probability=TRUE` when creating the histogram so that the histogram is scaled and the total area under the bars equals 1.

H Theoretically, it can be shown that the histogram is well approximated by the curve

$$f(x) = \frac{1}{\pi\sqrt{x(1-x)}}, \qquad x \in (0,1).$$

In other words, if X is the fraction of time that the balance is positive, then the probability density function of X is well approximated by f. This is called the arcsin law because the cumulative distribution function corresponding to f includes the arcsin function. Add the theoretical curve f to the histogram just created.

I Calculate the mean of the simulated fractions of time that the balance is positive.

J Now calculate the probability that we have a positive balance less than 5% of the time. Do this both based on the 1000 simulations and on the theoretical approximation. Indicate how well the results match your intuition and your answer to question E.