

MATH 20 – LAB 2

due Monday, August 21

Instructions: Complete all sections. Start this lab **early**, so you have plenty of time to ask questions.

Each experiment asks you to submit code. You must name the files as instructed and submit them with your completed assignment. Please comment your code!

The experiments ask you to perform some kind of simulation and describe your results. When describing, use the data you generated to make guesses at the expected value, etc. What does the distribution look like? Do you see a pattern?

You don't have to write a lot for each part! Just a few sentences (with pictures if necessary) will suffice.

Submission: You will submit your written answers (without your code) in class as a single document on Monday, August 21. **This must be typed.** It is okay to leave blanks for math equations and handwrite those if you need them. All graphs involved should be generated by R. Submit your code in two separate files to me via email, with the subject line Math 20 Lab 2 - [your first and last name]. Your code should be submitted by the time class starts on August 21. (It's fine to send it in earlier, too.)

EXPERIMENT 1 — MONTE CARLO

One way to estimate the area or volume of a geometric figure is to select points uniformly at random from a simple region containing the figure (such as a box or a cube) whose area or volume you know, then calculate the proportion of those points that fell within the figure whose area or volume is to be estimated. In this experiment, we will use Monte Carlo simulations to estimate the volume of the unit n -dimensional sphere for several values of n .

1. First, let's estimate the area of the unit circle. The unit circle is a circle of radius 1, centered at $(0,0)$. Pick points uniformly at random from the square with opposite corners $(-1, -1)$ and $(1, 1)$. A randomly chosen point (x, y) falls within the circle if and only if $x^2 + y^2 \leq 1$. Pick points until the ratio

$$\frac{\text{number of points in the circle}}{\text{number of points in the box}}$$

starts to converge to a few decimal places, then use this to estimate the area of the unit circle. (This may take 10s or 100s of millions of points—convergence is slow.) Check your answer against what you know the true area of the unit circle to be.¹

2. Repeat the same experiment to estimate the volume of the unit sphere. Now, you need to randomly pick points within the cube with opposite corners $(-1, -1, -1)$ and $(1, 1, 1)$, and a point (x, y, z) is in the sphere if and only if $x^2 + y^2 + z^2 \leq 1$.
3. Repeat the same experiment to estimate the volume of the n -dimensional sphere for all n between 4 and 10 inclusive. Present a table that summarizes your findings for all n between 2 and 10, and describe your results. Form a conjecture.
4. Gather some more evidence for your conjecture by testing higher n values.

¹It will help to have your code print out the updated ratio every so often, but if you print too often it will become the bottleneck for your program. I'd recommend something like every 100k or 1m iterations. Ask me if you need help figuring out how to do this.

EXPERIMENT 2 — BITCOIN INVESTING

You decide to start investing in bitcoin on a day when the price of bitcoin is exactly \$100 per bitcoin. Bitcoins can be bought or sold in any fractional quantity—for example, if the price of bitcoin is \$30 one day, and you have only \$10, you can buy exactly 1/3 of a bitcoin.

In this experiment, we will simulate several buying/selling techniques to see if we can beat the system and become millionaires. For simplicity, we will assume that the price of bitcoin changes exactly once per day, and that at the end of each day you can decide whether to buy more bitcoin (i.e., converting cash to bitcoin), or sell bitcoin (i.e., converting bitcoin back to cash). To simulate the fluctuation in bitcoin, use the following R function.

```
prices_over_time = function(number_of_time_increments, initial_price) {  
  return(cumprod(c(initial_price,exp(0.02*rnorm(number_of_time_increments)))));  
}
```

Call this function with two variables: the first is the number of days you want to simulate, and the second is the initial price. The function will return a vector V of prices such that $V[i]$ is the price on day i .

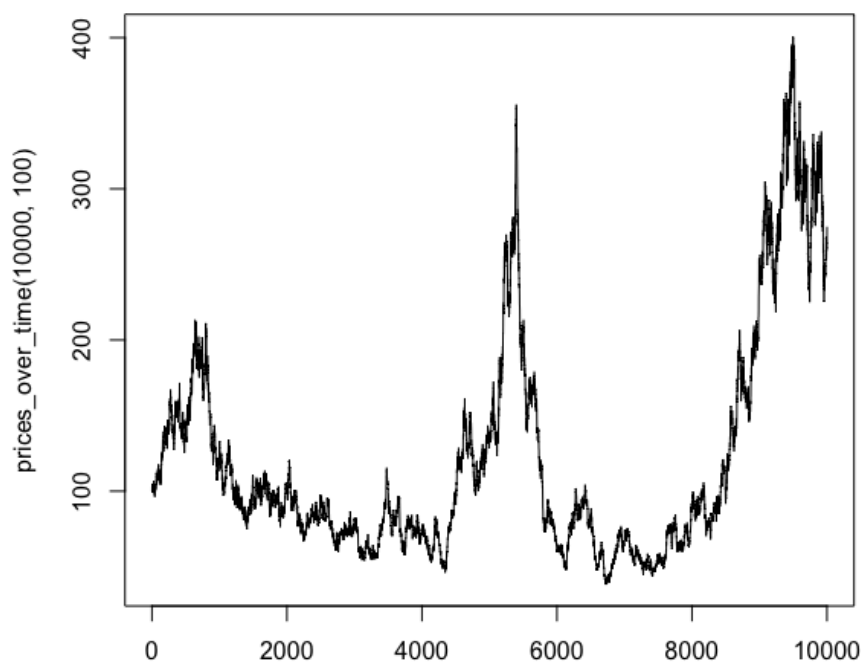
For example:

```
> prices_over_time(10,100)  
[1] 100.00000 98.01108 97.00782 100.08954 102.18466 101.90205 104.61192 107.51071 108.81351 110.71530  
110.28945
```

You can also look at plots using the command

```
> plot(prices_over_time(10000,100), type="l")
```

Here's an example:



Your code will “know” the prices for all future days before you start trading, but no cheating! You can only

use the current days information to make buy/sell decisions.

1. First we'll try Strategy A. You start with \$1000, and on Day 1 when the price is \$100 per bitcoin, you convert it all to bitcoin. (In other words, you buy 10 bitcoins.) Then, you wait. If the price of bitcoin ever reaches $\geq 20\%$ more than what you bought it for, you sell all of your bitcoin. After selling, you wait for the price to go down at least 10% from your sell point, then you use all of your cash to buy again, repeating the process. Suppose you follow this procedure for 10,000 days. At the end, your net worth is equal to your cash holdings plus the cash value of your bitcoin holdings on that day. Run many simulations to find your expected net worth on day 10,000. Perhaps a much better metric is the following: how did you do compared to bitcoin as a whole? Would you be better off if you had just bought bitcoin on the first day and held it? Include plots in your write up!
2. Next we'll try Strategy B, which is a less extreme version of Strategy A. You start on Day 1 with \$1000 dollars, and you spend \$200 of it on bitcoin. (In other words, you buy 2 bitcoins.) Every day, you do the following. If the price of bitcoin goes up from the previous day, you sell 20% of your current bitcoin holdings. If the price of bitcoin goes down from the previous day, you use 20% of your cash holdings to buy more bitcoin. Run simulations to find your expected net worth after 10,000 days. How does the strategy do compared to just buying and hodling bitcoin? Include plots!
3. Come up with another trading strategy (no cheating, no time travel!), and test it against simulations. How good of a strategy can you come up with? How well can you outperform "the market"?²

²Disclaimer: The stock market is much more complicated than this. Do not make any investment decisions based on this lab assignment alone!