

Random Portfolios

Mike Flynn, Angel Zhou, Dave Kane

August 16, 2013

Introduction

Put yourself in a scenario where you have 6,250 dollars. Let's say that you want to invest this money in two stocks: General Electric (GE) and International Business Machines (IBM). There is an endless number of ways that you could split your money into the two stocks. For example, you could invest 25 percent of your 6,250 dollars in GE and the remaining 75 percent in IBM. In another scenario, you could halve your money by putting 3,125 dollars into GE and 3,125 dollars into IBM. The possibilities stretch on.

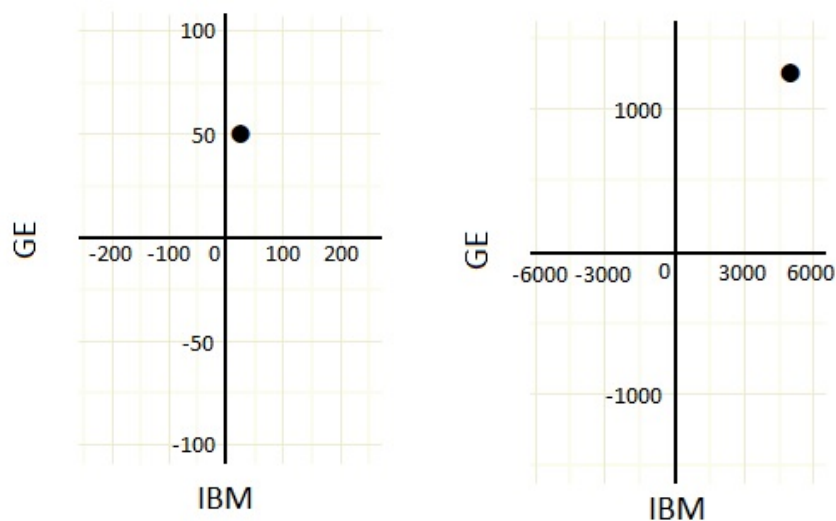


Figure 1: Consider the situation where you invest in 25 shares of IBM and 50 shares of GE. This coordinate, where $x = 25$ and $y = 50$, is displayed on the left plot. This graph thus shows a single portfolio in what is called a share space, meaning that the x and y coordinates represent the amount of shares invested in each stock. Alternatively, one can view the portfolio in dollar terms. This is done by multiplying the stock price by the number of shares, resulting in the total amount spent in each stock. The price of GE is currently at around 25 dollars apiece while the value of each IBM stock is approximately 200 dollars. Thus, 25 shares of IBM becomes 5,000 dollars and 50 shares of GE costs 1,250 dollars. The plot on the right displays the same portfolio as the one on the left, but plots the portfolio in dollar space. This means that the x and y coordinates represent the amount of money that is invested into each respective stock.

Now that we have seen portfolios in both share and dollar space, what else is there?

Portion space refers to percentages. Say you spend 30 percent of your money on IBM. In other words, .3 of your total money is invested in IBM. For the rest of this document, we will be referring to portfolios plotted in portion space.

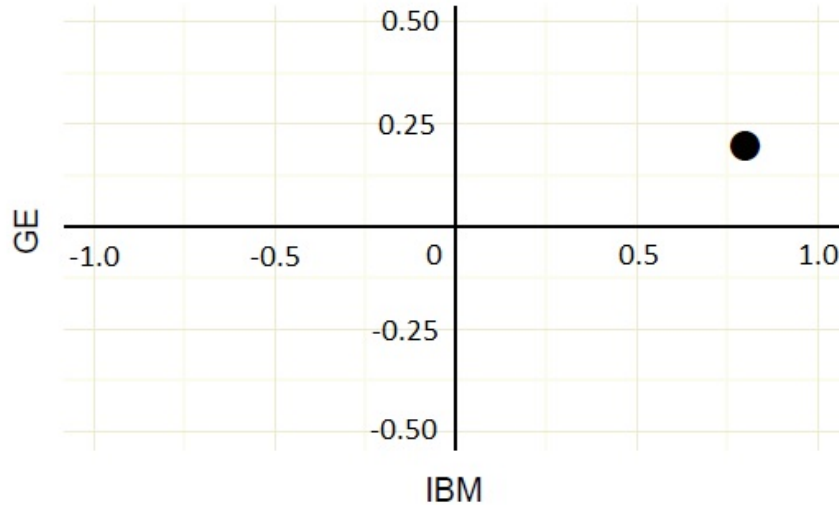


Figure 2: The same portfolio as that in Figure 1 is considered again in the above plot. Instead of share or dollar space, the portfolio (consisting of 25 and 50 shares of IBM and GE respectively) is now shown in portion space. Given that you hypothetically invest 5,000 dollars on IBM and 1,250 dollars on GE, you expend a total of 6,250 dollars (since all of your money was spent, this portfolio is fully invested; no money is put aside for other purposes). $1250/6250 = 0.2$, so 20 percent of your budget is spent in GE. Transitively, 80 percent is put into IBM. These percentages are plotted as decimals.

As mentioned before, there is an infinite number of ways to split the invested money. No matter how you divide your total dollar amount, whether you put .5 in IBM and .5 in GE, or .2 in IBM and .8 in GE, the two values must sum to 1, or 100 percent. In other words, every portfolio that we will consider will be fully invested.

In order to visualize how all the possible portfolios would appear in a 2-D space, you can generate random weights, creating a completely arbitrary way of splitting the money. Some constraints that must be followed when generating random weights are as follows: $x + y = 1$ and $x, y > 0$.

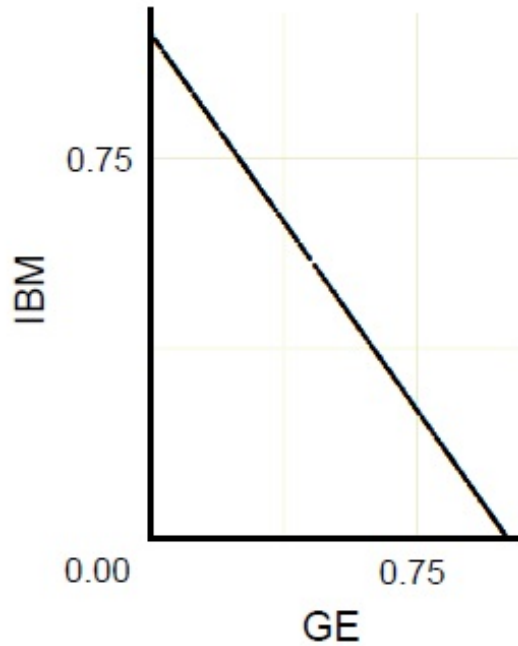


Figure 3: 1,000 random samples of General Electric (GE) and International Business Machines (IBM) are taken. Each point represents a possible portfolio. The result is a diagonal line extending from the point (0,1) to the point (1,0). Because x and y must sum to 1, a line is formed where every set of x and y coordinates add up to 1.

Let's say that you want to add Coca-Cola (KO) to the stocks that you are investing in. Now that you have three stocks instead of two, you must generate three random weights that sum to 1. To create an example, you could invest .3, .3, and .4 of your total funding in KO, GE, and IBM respectively.

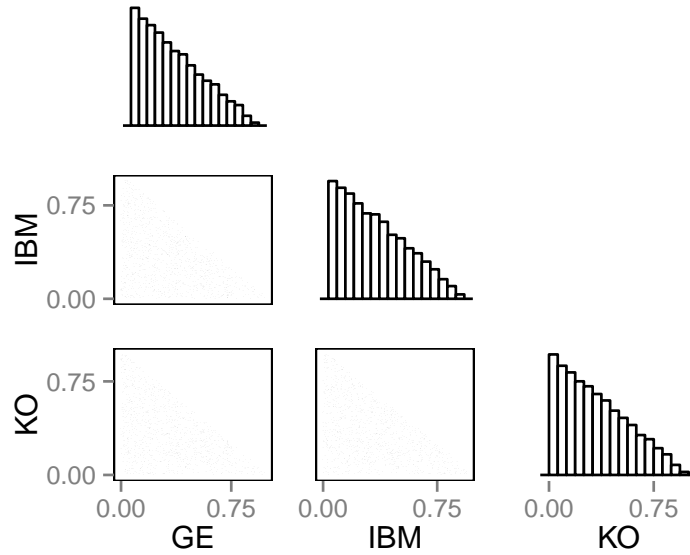


Figure 4: 1,000 random weights were sampled for Google (GE), Coca-Cola (KO), and International Business Machines (IBM). Once again, these portfolios are all fully invested. The pairwise scatterplots display the same distribution of the 1,000 random weights. According to the plots, the maximum sum for the weight distributions for any two variables is 1, and the minimum sum is 0.

The three histograms on the diagonal represent density plots of 10,000 randomly sampled weights. More samples are taken for the histograms in order to increase their resolution. The histograms display the distributions of the variables and they indicate that the most likely value for any given variable is close to 0.

Now let's consider yet another scenario, where you are investing in five different stocks: General Electric (GE), International Business Machines (IBM), Coca-Cola (KO), General Motors (GM), and Microsoft (MSFT). Like before, the generated portfolios are fully invested. For example, one possible portfolio could consist of the weights 0.2, 0.3, 0.1, 0.15, 0.25 for each respective stock.

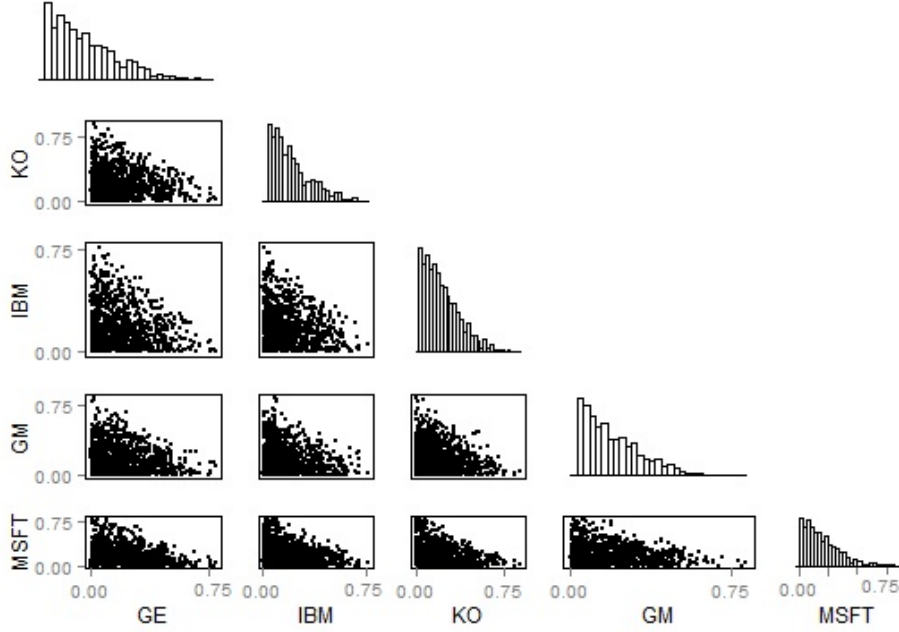


Figure 5: 1,000 random weights were sampled for Google (GE), Coca-Cola (KO), International Business Machines (IBM), General Motors (GM), and Microsoft (MSFT).

10,000 weights were randomly sampled for the five histograms. These graphs function as density plots.

hitandrun()

In order to create the scatterplots and histograms previously shown in this document, random points had to be generated. These points were created using a function called `hitandrun()`. The algorithm for this function requires several steps:

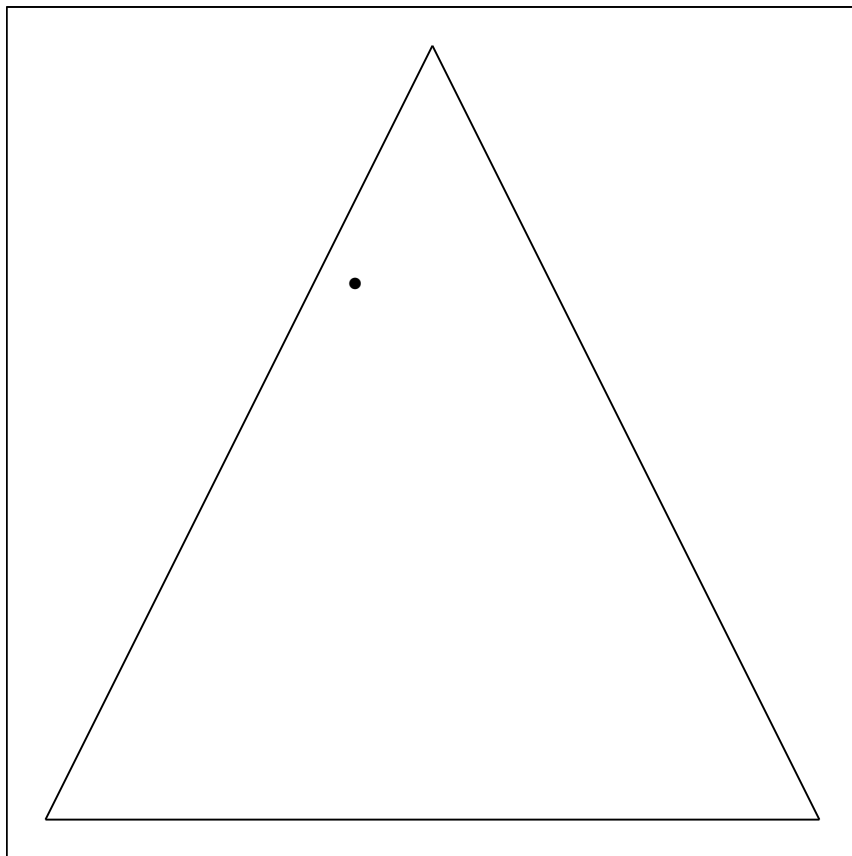


Figure 6: `hitandrun()` first picks a random point within a simplex.

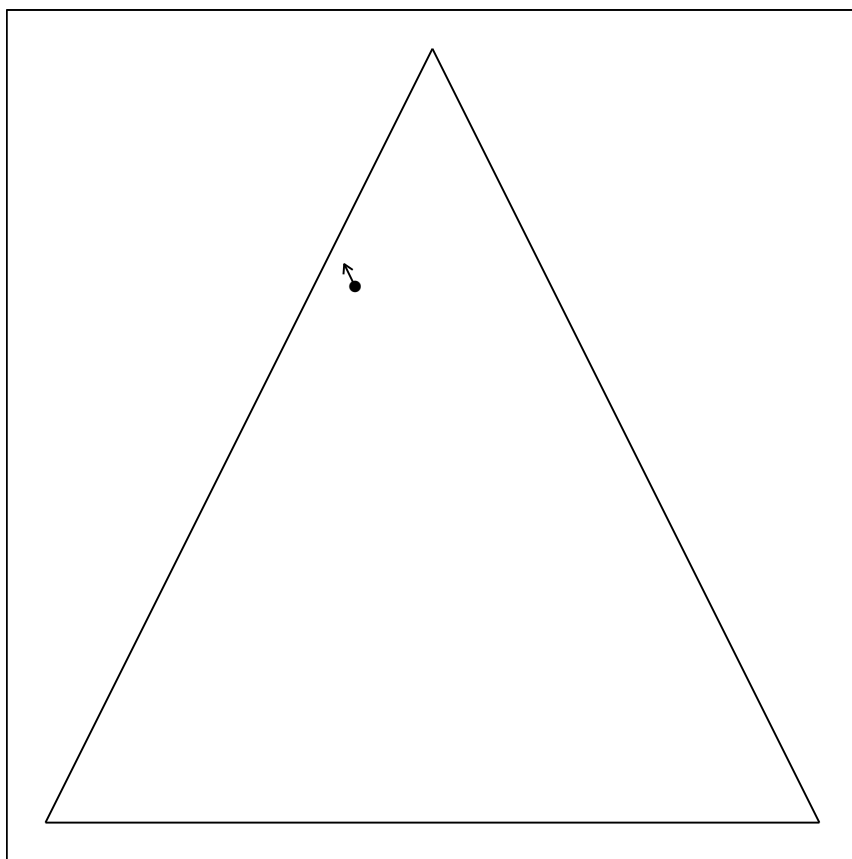


Figure 7: The function then chooses a random direction, as shown by the arrow.

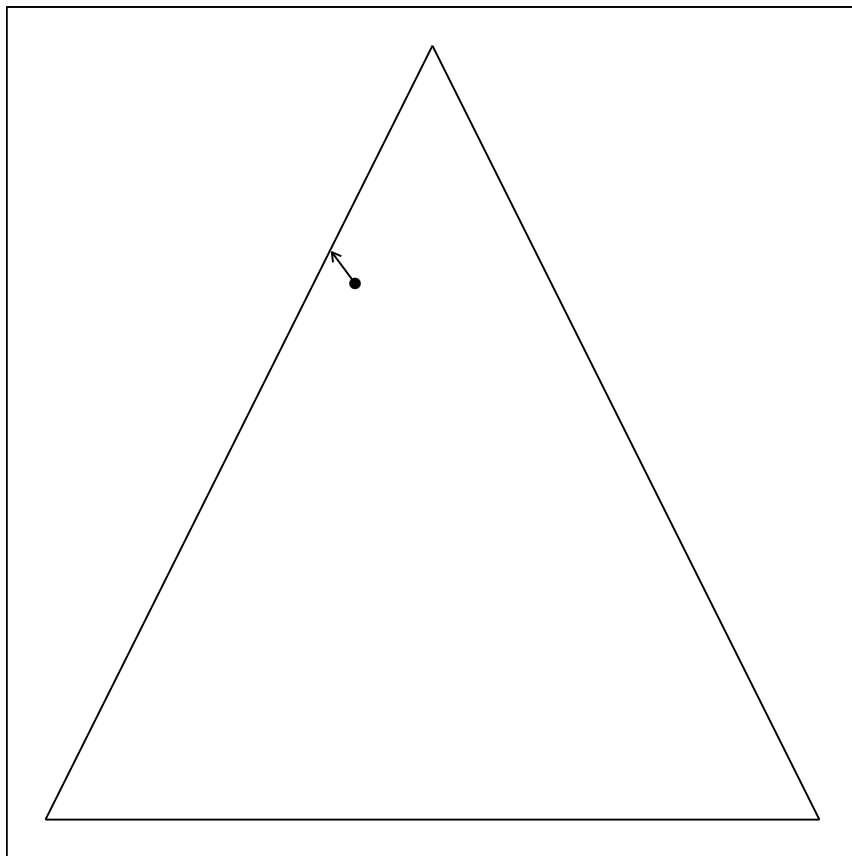


Figure 8: `hitandrun()` goes in the chosen direction until it hits an edge of the simplex. In this figure, the arrow was extended until it hit a wall.

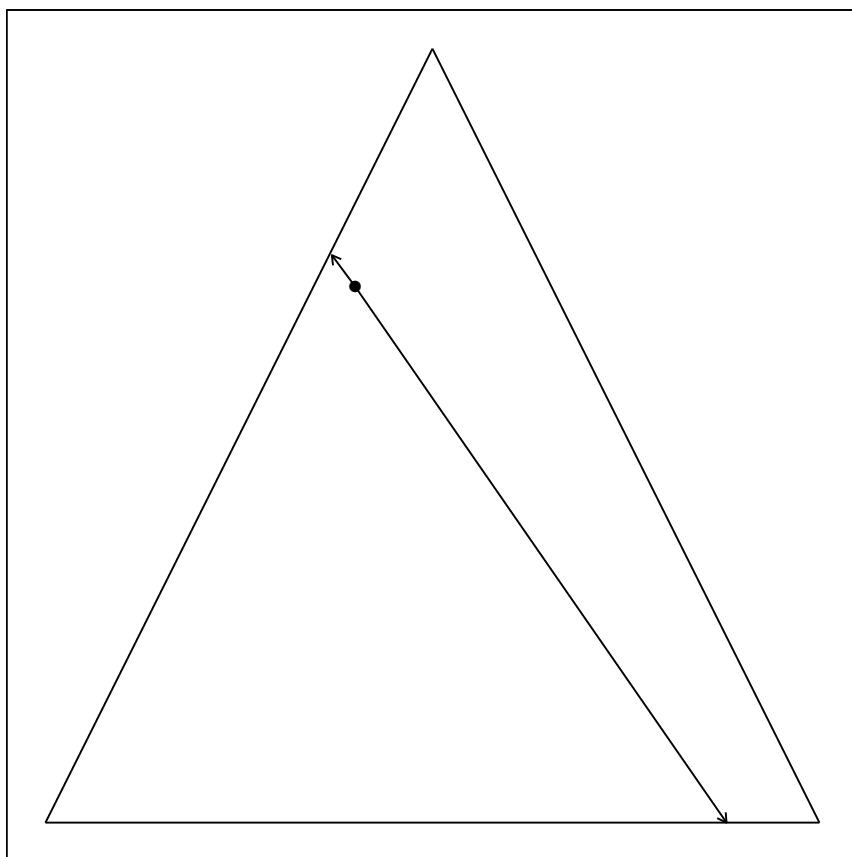


Figure 9: Next, `hitandrun()` goes in the negative direction until it hits another point on the edge of the simplex. This is shown with another arrow.

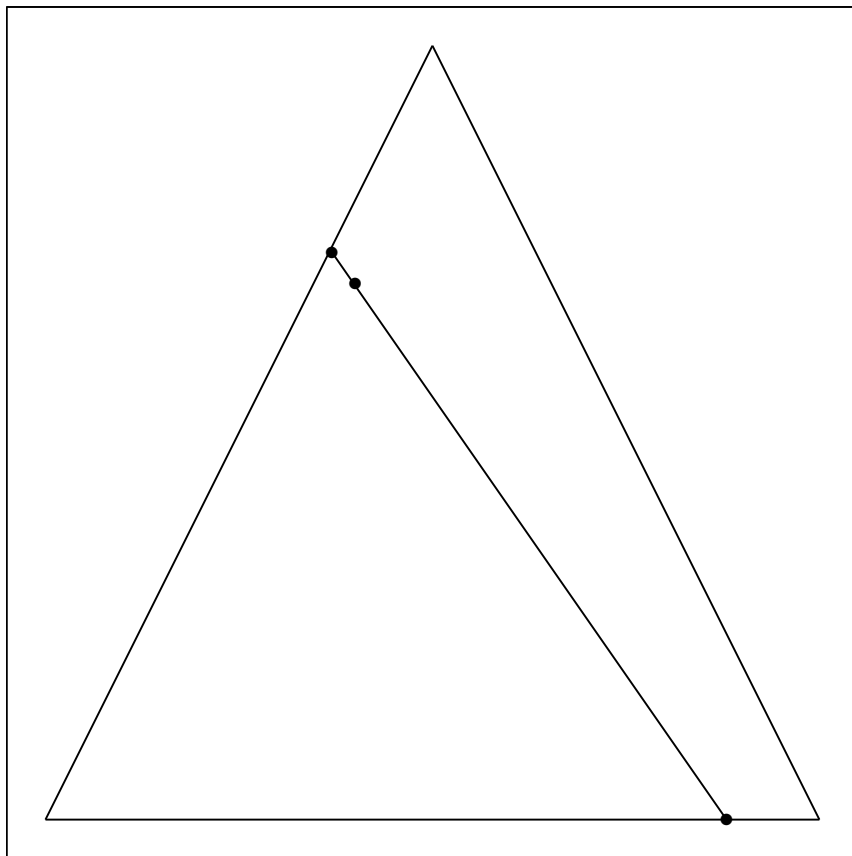


Figure 10: A line segment is formed. It passes through the initial point and connects the two points where the directional arrows intersected the simplex.

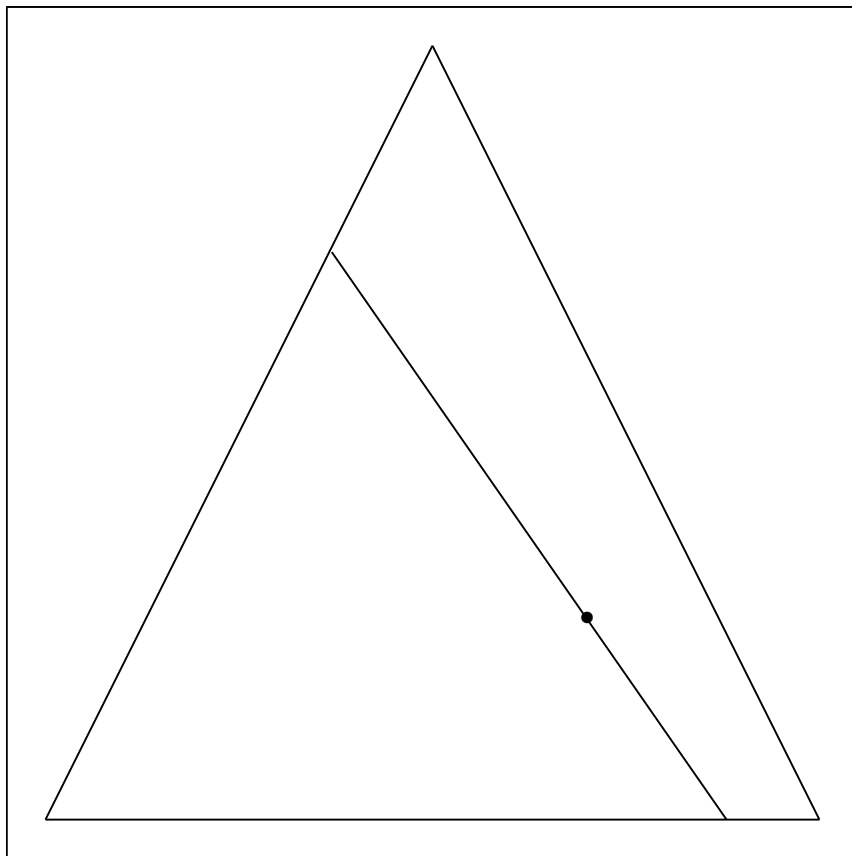


Figure 11: Finally, a random point is picked on the line segment shown in figure 10.

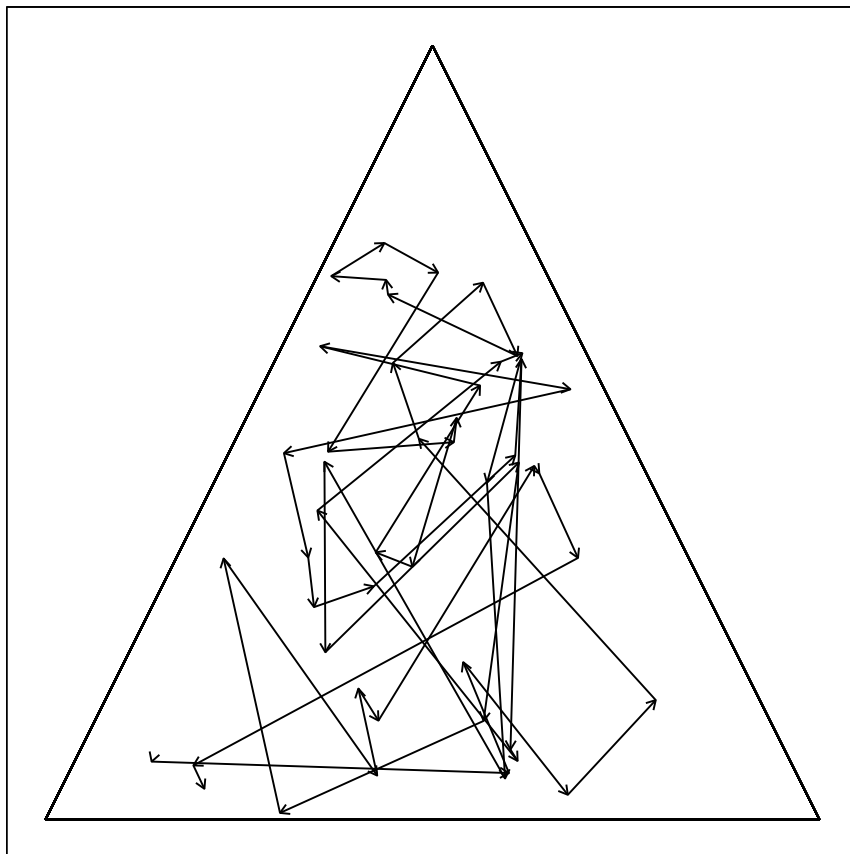


Figure 12: `hitandrun()` then repeats the previous steps, resulting in the creation of many randomized points. This figure displays the outcome of 50 repetitions of the `hitandrun` algorithm that was just explained.

`mirror()`

`mirror()` is another function used in the process of creating random portfolios. This function also consists of points jumping from one random place to another.

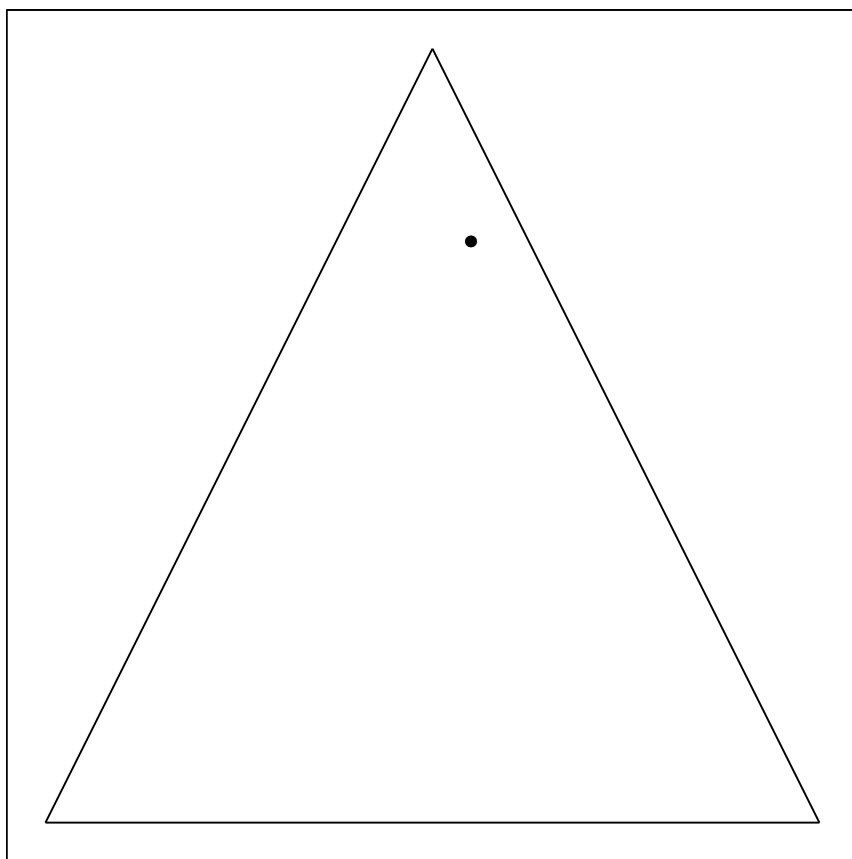


Figure 13: Similar to `hitandrun()`, `mirror()` begins by selecting a random point.

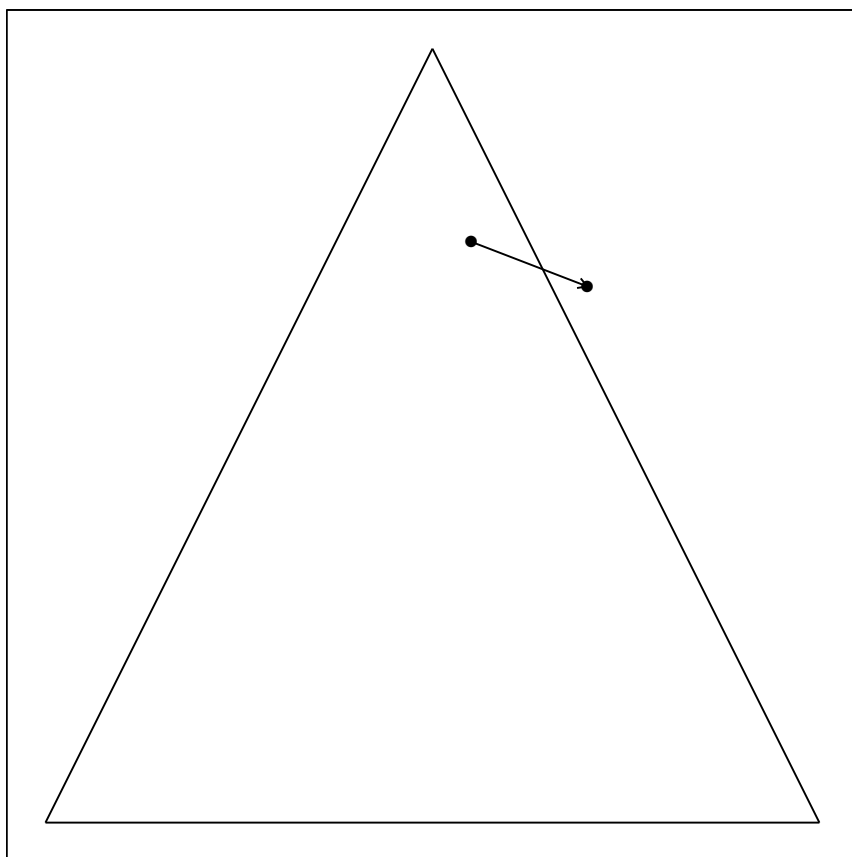


Figure 14: The point then jumps to a random place in space. It does not matter if that point is within or outside of the simplex.

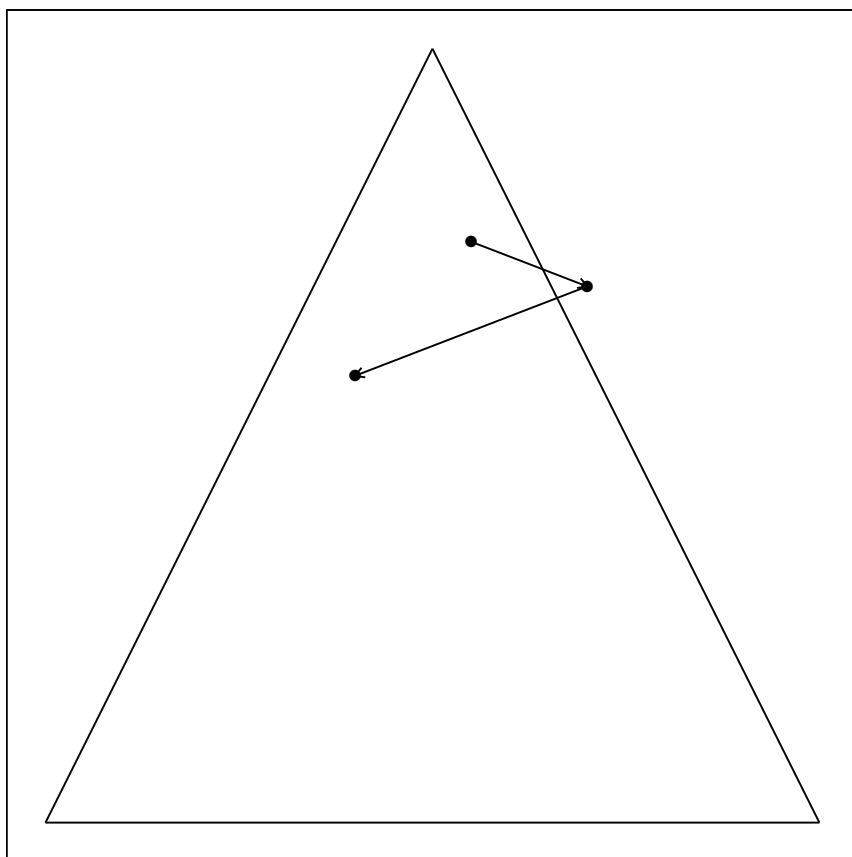


Figure 15: The step displayed in Figure 14 is then repeated.