Random Portfolios

Mike Flynn, Angel Zhou, Dave Kane

August 14, 2013

Introduction

Imagine you have one thousand dollars that you want to invest in two stocks: General Electric (GE) and International Business Machines (IBM). You could invest 25 percent of your million dollars in GE and the remaining 75 percent in IBM, or, to present another scenario, you could halve your money and put five hundred thousand dollars into GE and five hundred thousand dollars into IBM. There is an endless amount of ways that you can invest one million dollars into the two stocks.

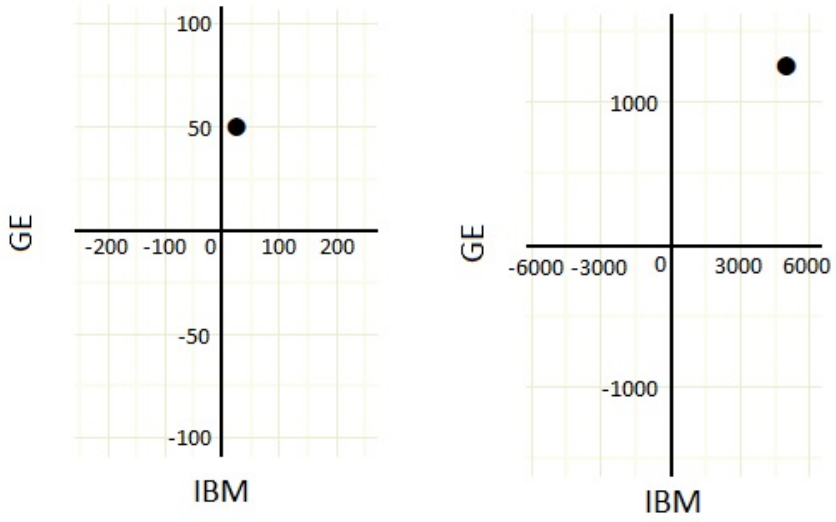

Figure 1: Let's say your portfolio consists of 25 shares of IBM and 50 shares of GE. The plot on the left displays this coordinate in share space, meaning the x and y coordinates of the point represent the amount of shares invested in each stock. Alternatively, you can view the portfolio in dollar terms. This is done by multiplying the stock price by the number of shares, resulting in the total amount spent in each stock.

The current price of GE stock is around 25 dollars while the value of IBM stock is approximately 200 dollars. Thus, 25 shares of IBM costs 5,000 dollars and 50 shares of GE is valued at 1,250 dollars. The above plot on the right displays the same portfolio as the one on the left, but plotted in dollar space. This means that the x and y coordinates represent the total amount of money that you have invested into each respective stock.

Figure 2: The same portfolio as that in Figure 1 is replotted in portion space. In other words, this figure displays the portfolio on the x-y axes so that the x and y coordinates represent the percentage of the total amount of money that is invested into each respective stock. To use the same example as before, since you spent 5,000 dollars on IBM and 1,250 dollars on GE, a total of 6,250 dollars was expended. $1250/6250 = 0.2$, so 20 percent of your budget is spent in GE. Transitively, we find that 80 percent is spent on IBM. These percentages are plotted in decimal form in the above figure. In the rest of this document, we will be referring to portfolios plotted in portion space.

As mentioned before, there are an infinite amount of ways to split the invested money. Earlier, we thought about dividing the money in dollar terms. Now let's think about it in percentages. [Note: whenever percentages are mentioned or used in this document, they are converted to decimal form. For example, 75 percent is equivalent 0.75.] No matter how you split your money, 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, this portfolio is fully invested; no money is put aside for other uses. 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: $x + y = 1$ and $x, y > 0$.

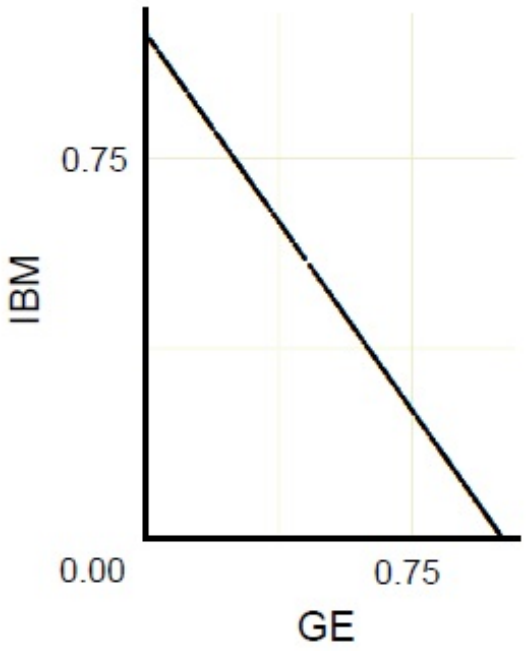

Figure 3: 1,000 random samples of two stocks are taken, where each stocks' value represents the percentage of the total money that is invested in that stock. Each point represents a possible portfolio. The result is a diagonal line extending from the point (0,1) to the point (1,0). A line is formed becauseone of the constraints when generating weights, as previously mentioned, is that the weights must sum 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.

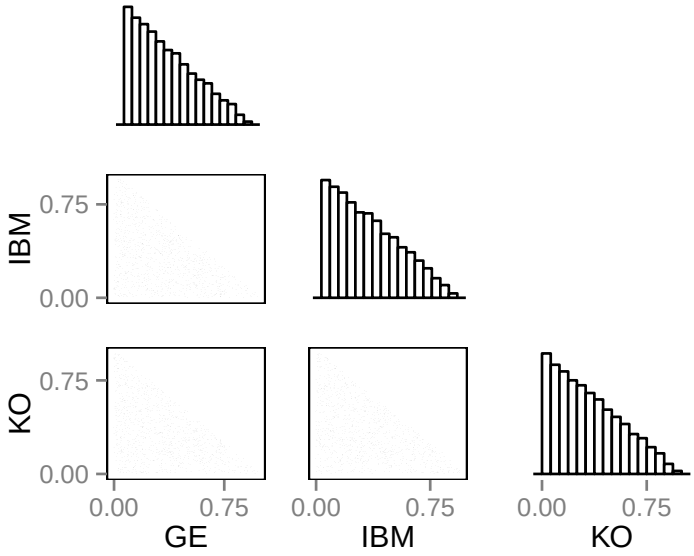

Figure 4: Random weights were sampled for Google (GE), Coca-Cola (KO), and International Business Machines (IBM). Once again, these portfolios are fully invested. The pairwise scatterplots display the same distribution of 1000 random weights. The maximum sum for the weight distributions for any two variables is 1, and the minimum sum is 0.

The three histograms represent density plots of 10,000 randomly sampled weights. More samples are taken for histograms in order to up 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 take a scenario where you are investing money 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 and long-only.

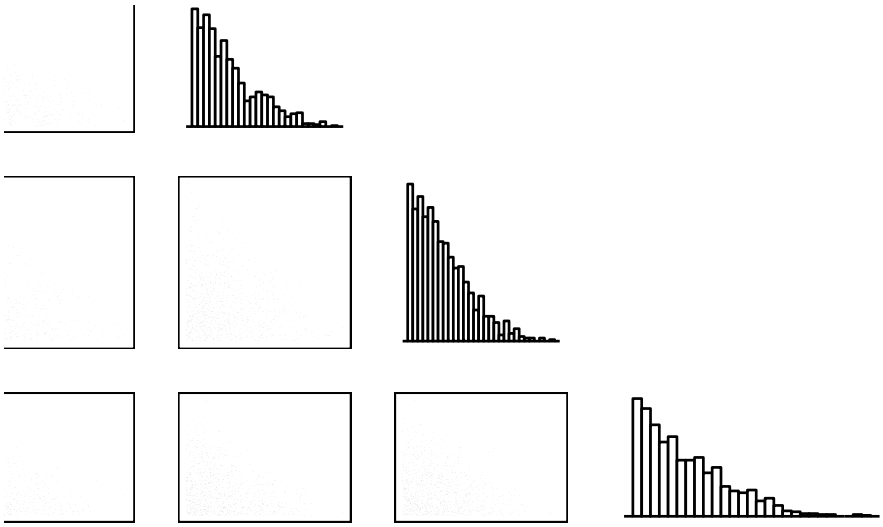

Figure 5: Random weights were sampled for Google (GE), Coca-Cola (KO), International Business Machines (IBM), General Motors (GM), and Microsoft (MSFT). Like before, these portfolios are fully invested and long-only (no stocks are short-sold). The pairwise scatterplots display 1000 random weights, and show that they have the same distribution.

The histograms are density plots of 10,000 randomly sampled weights and display the distributions of the variables. They indicate that the most likely value for any given variable is close to 0. (***)

hitandrun()

In the scatterplots and histograms shown previously in the document, random points were generated. These points were created using the function hitandrun(). The algorithm for this function requires several steps:

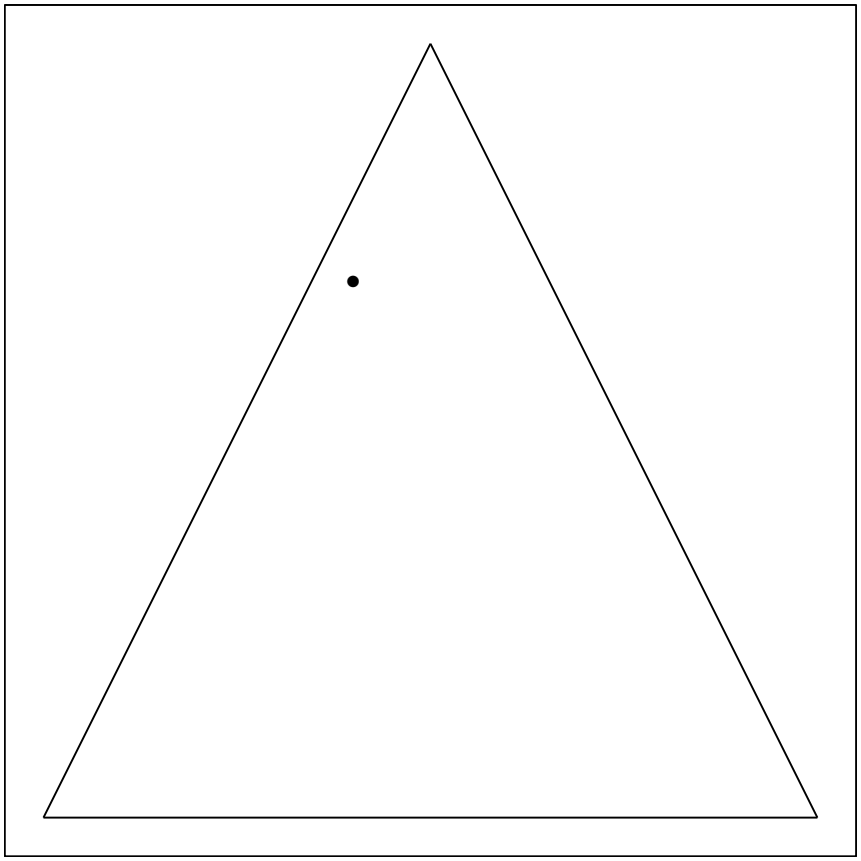

Figure 6: hitandrun() first picks a random point within a simplex.

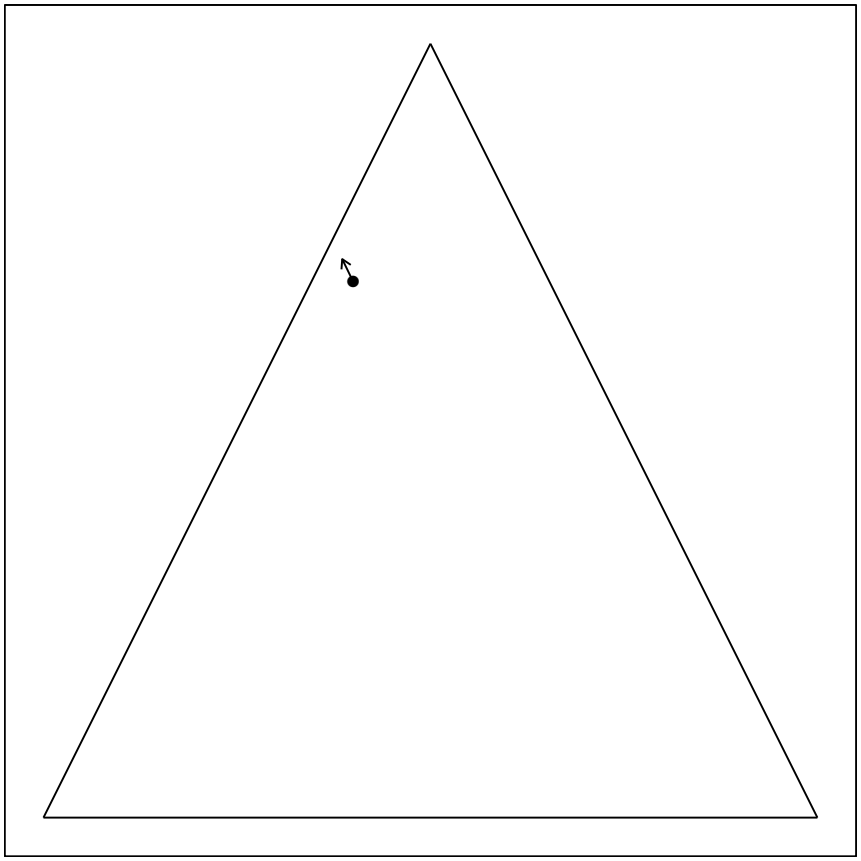

Figure 7: Next, hitandrun() picks a random direction, as shown by the arrow.

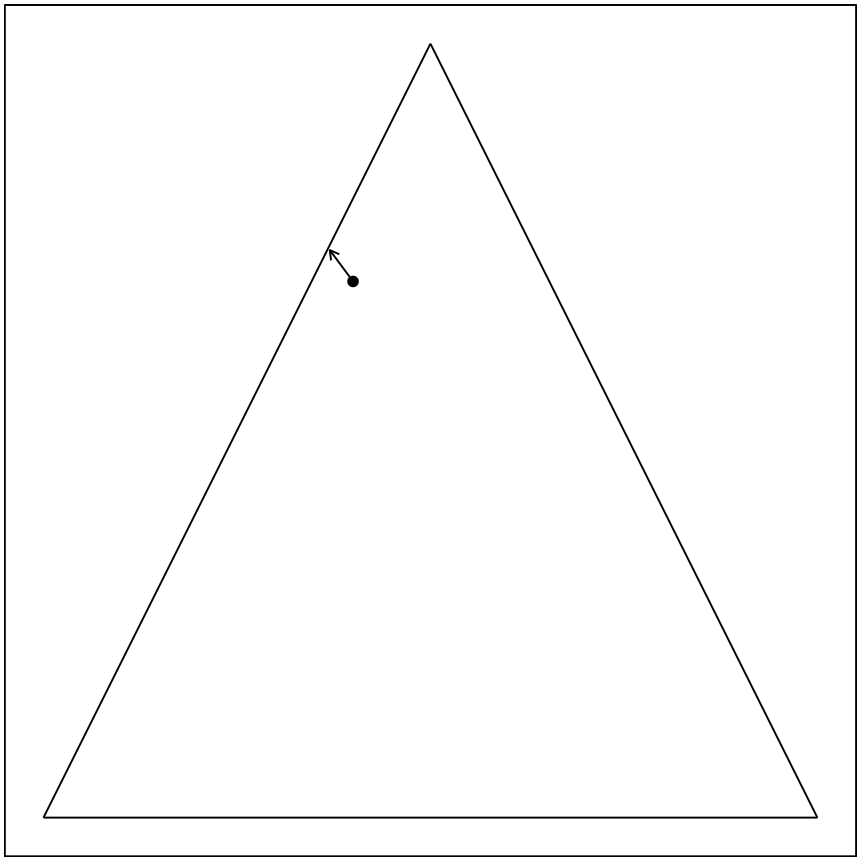

Figure 8: The function then extends that arrow until it hits an edge of the simplex.

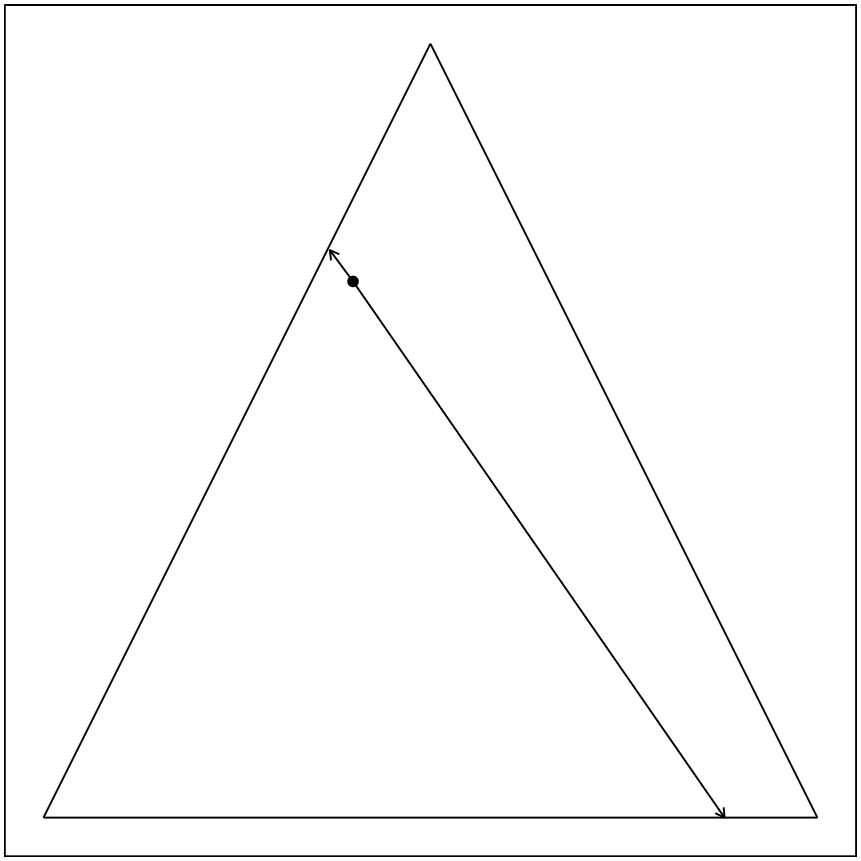

Figure 9: Next, hitandrun() extends the arrow in the other direction until it hits another point on the edge of the simplex.

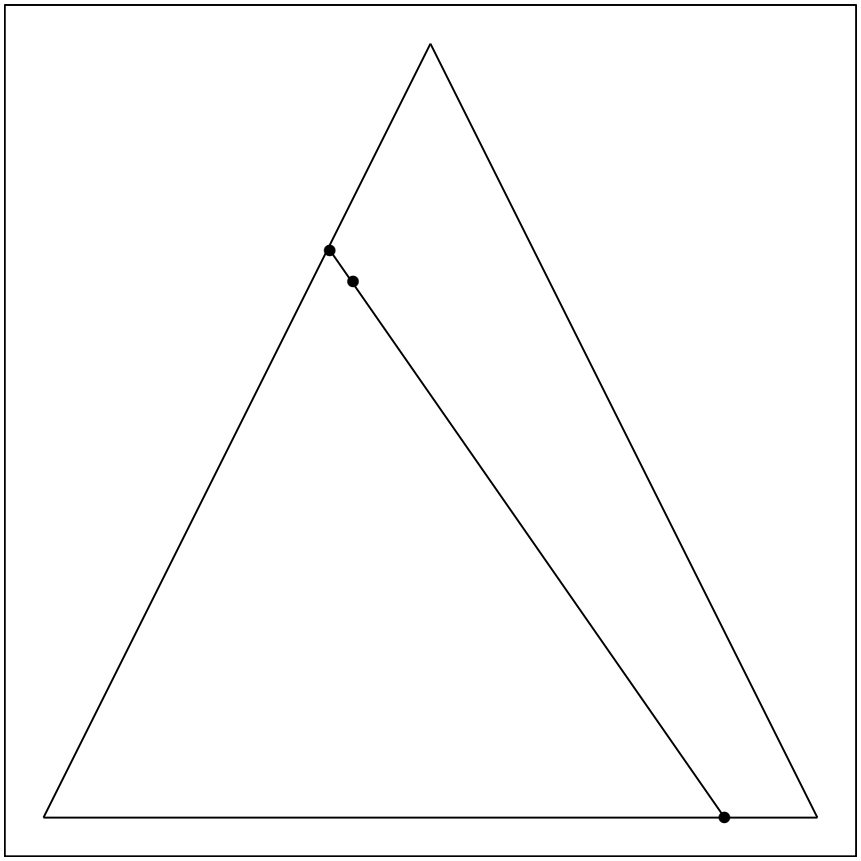

Figure 10: A line segment is formed. It passes through the initial point and connects the two points where the directional arrows intersected the simplex.

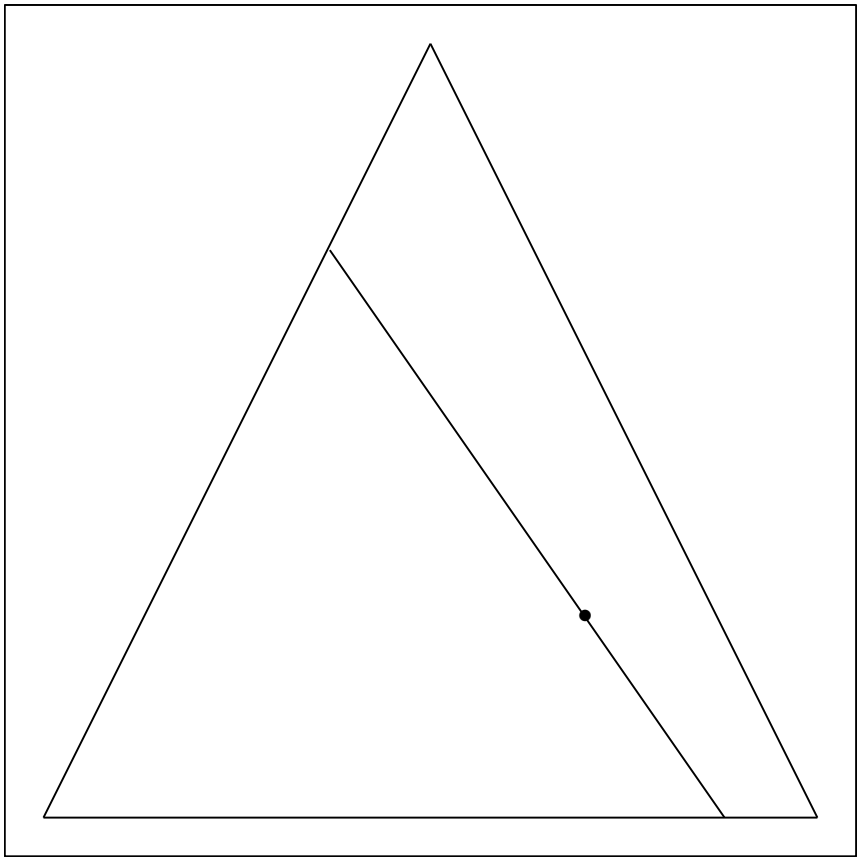

Figure 11: Finally, a random point is picked on the line segment shown in figure 10.

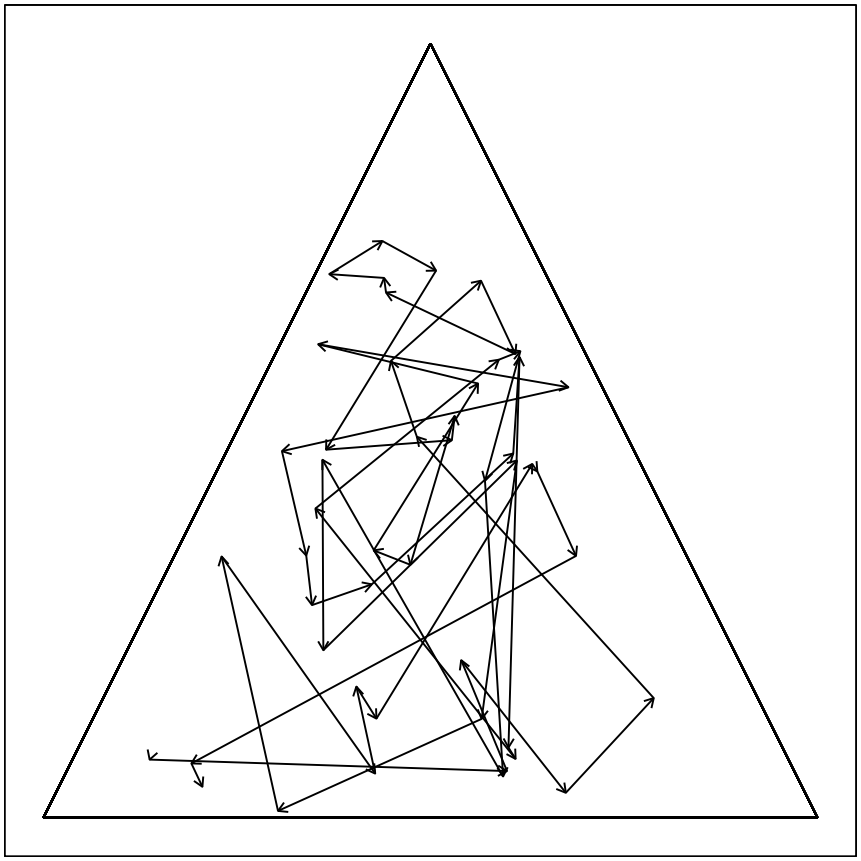

Figure 12: hitandrun() then repeats the previous steps, resulting in the creation of many points. This figure displays the outcome of 50 repetitions of the hitandrun algorithm that was just explained.

mirror()

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

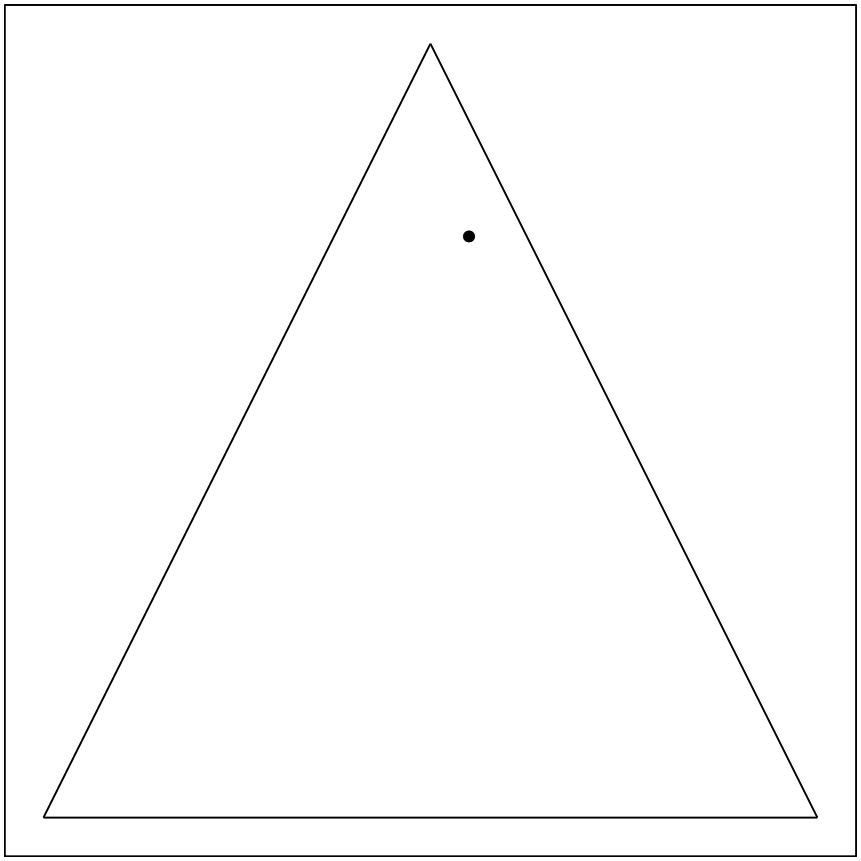

Figure 13: Similar to hitandrun(), mirror() begins by selecting a random point.

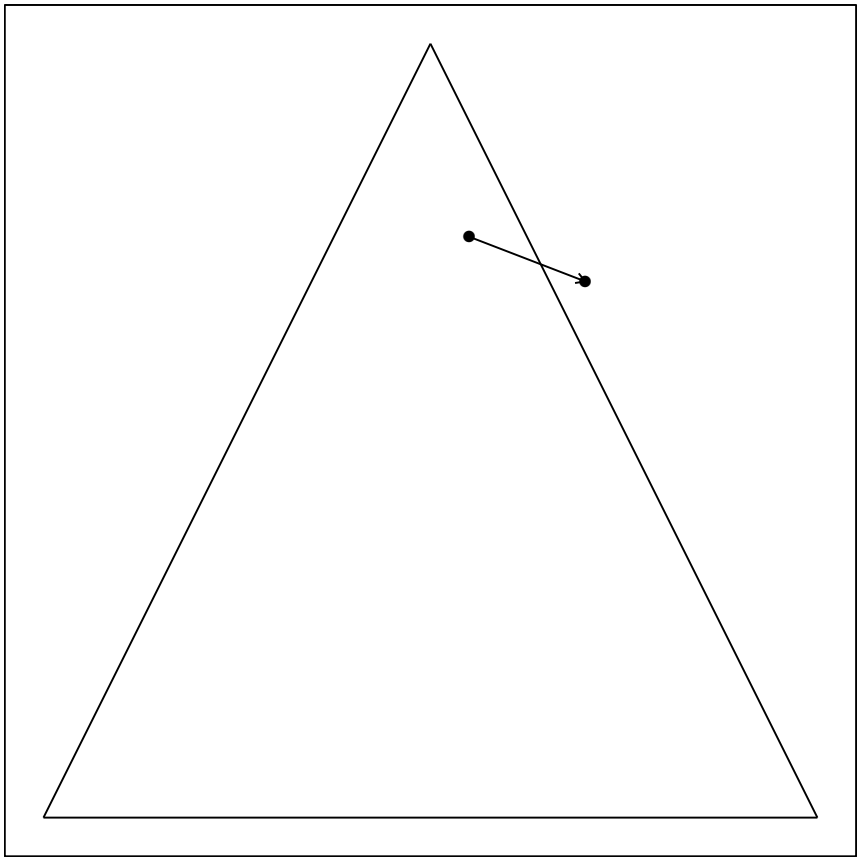

Figure 14: The point then jumps to a random place in space.

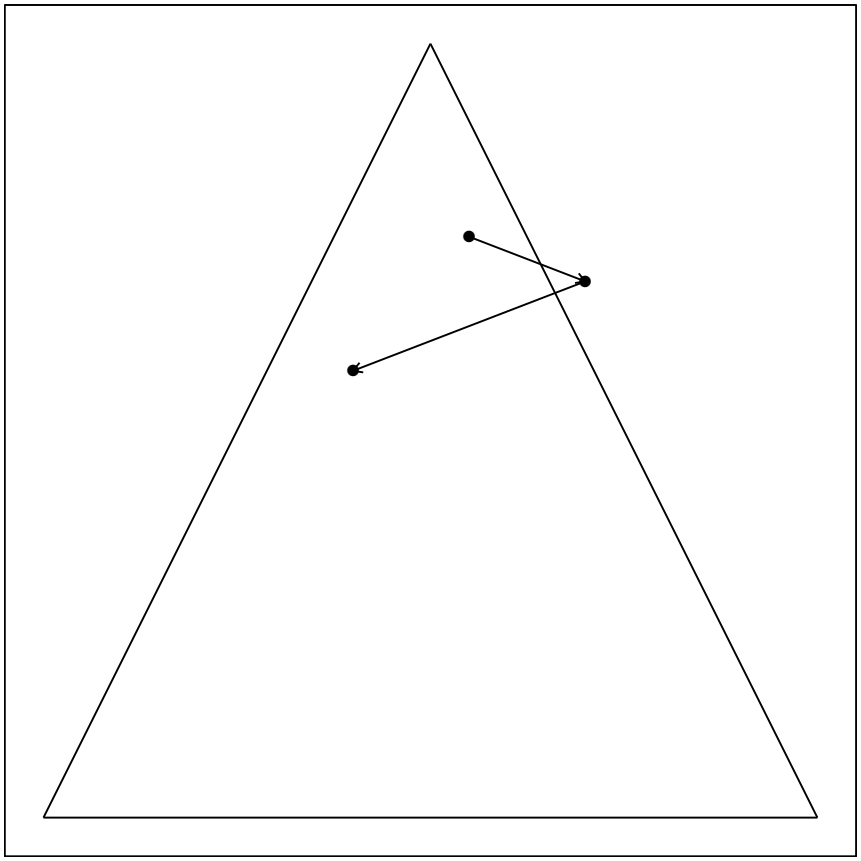

Figure 15: The step from Figure 14 is then repeated.

```
## Error:  object '.x37' not found
```