Estimation of Pi using Monte Carlo Simulation

Trelan Hakkem-Barron

4/18/2020

For this project, I will be show you how to use the Mnte Carlo Method to simulate the estimation of Pi. As we know, if r is the radius of a circle, then the surface of this circle is PI * r * r. Therefore, if we know the radius of a circle, and if we can estimate the surface of this circle, we can estimate PI. We know that the math constant Pi can be approximated by 4 times of the number of points inside a circle divided by the total number of points. How can we estimate the surface of a circle with a random method. The answer is found in the square!

To show the simulation, I will be generating a large number of random points and see how many fall in the circle enclosed by the unit square. The points inside the circle will be shown in black. The points outside the circle will be shown in blue.

The Use of 100 Darts

In the figure below, we see how the points are quite spread and cannot get a accurate picture. This is no where near close to pi. We need to get to a closer apprximation to pi. Let us try this again with 1,000 darts.
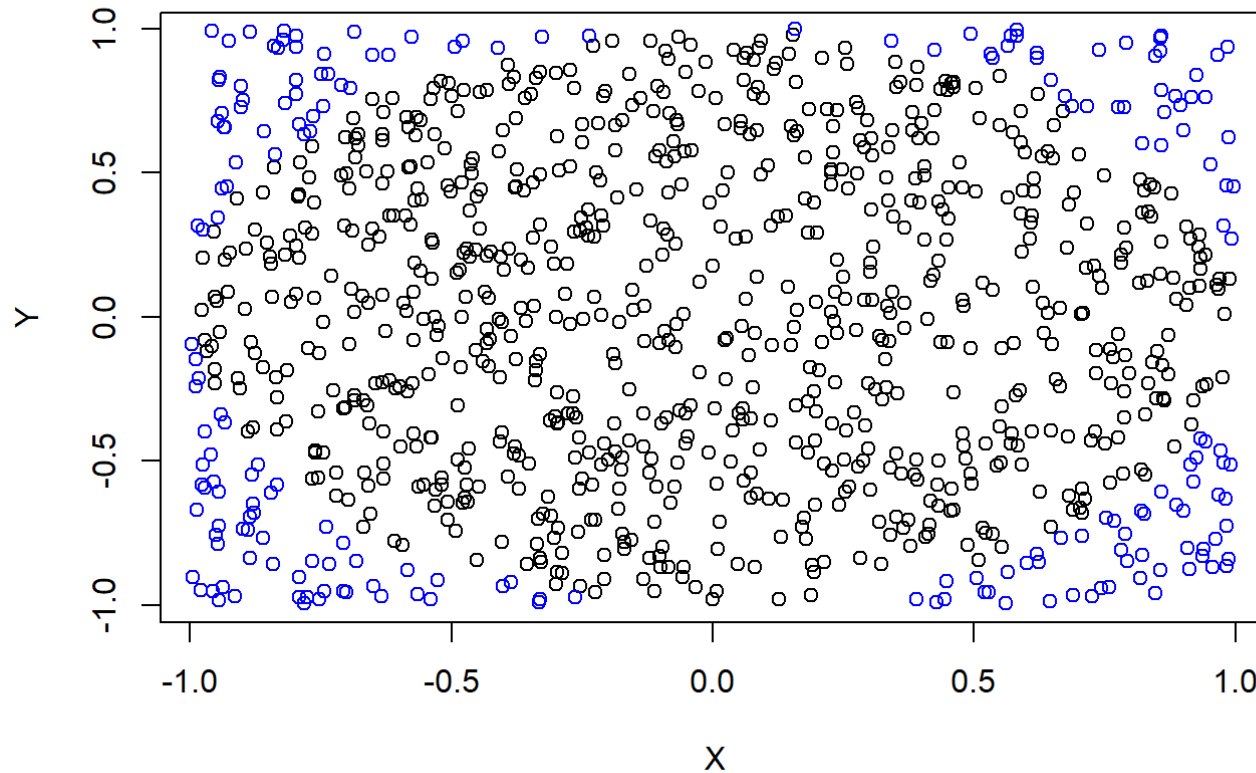
Simulation using 100 Darts
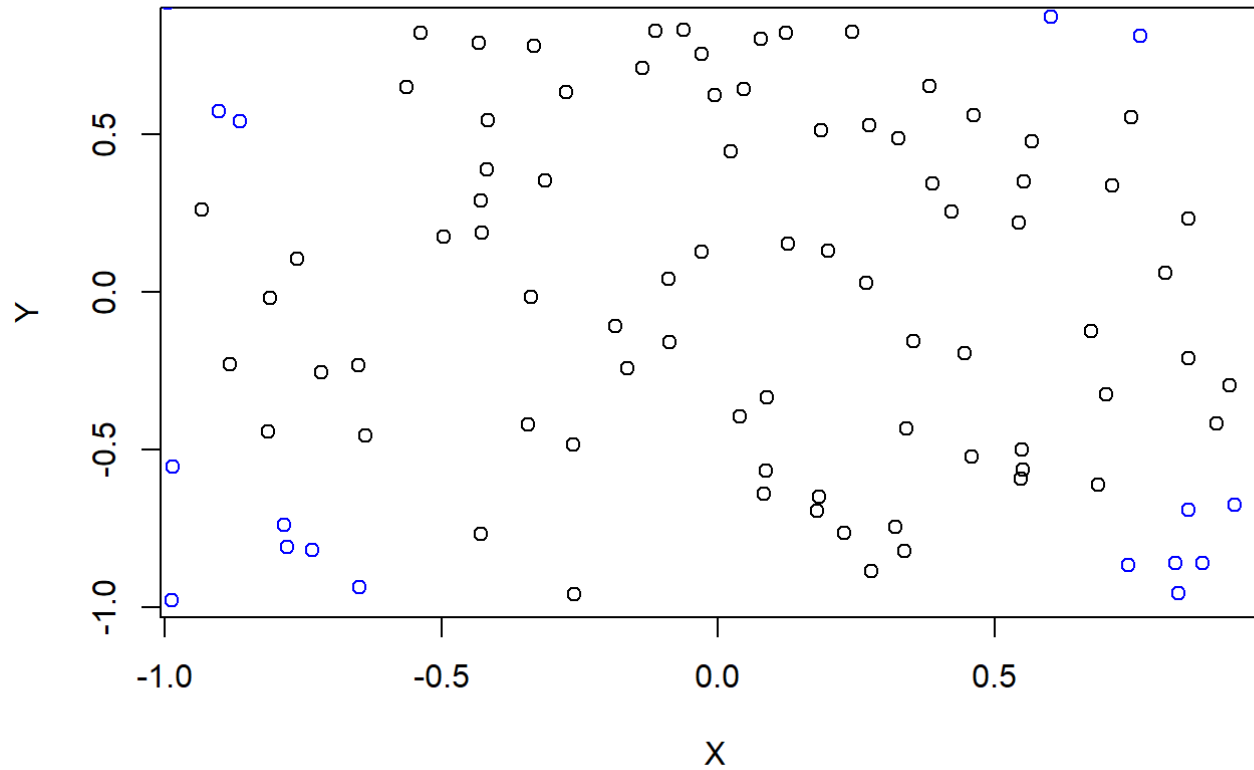

```
## [1] 3.16
```

The Use of 1,000 Darts

It is staring to show a better picture now. We are slowly but surely getting closer to pi. Let's try with 10,000 darts.
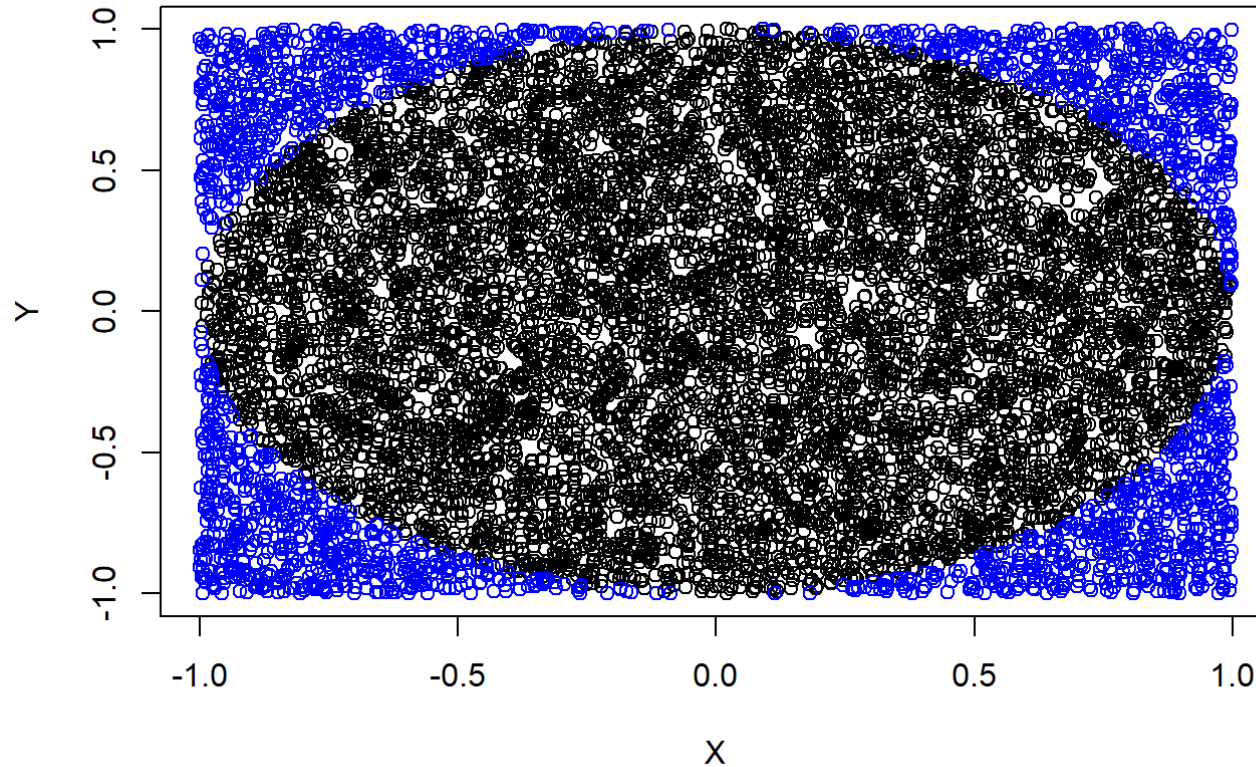
Simulation using 1,000 Darts


```
## [1] 3.192
```

The Use of 10,000 Darts

Now this is really good. The appproximation of pi is looking good as well. I think we can try this one more time for a better picture overall. Let's try with 100,000 darts this time.
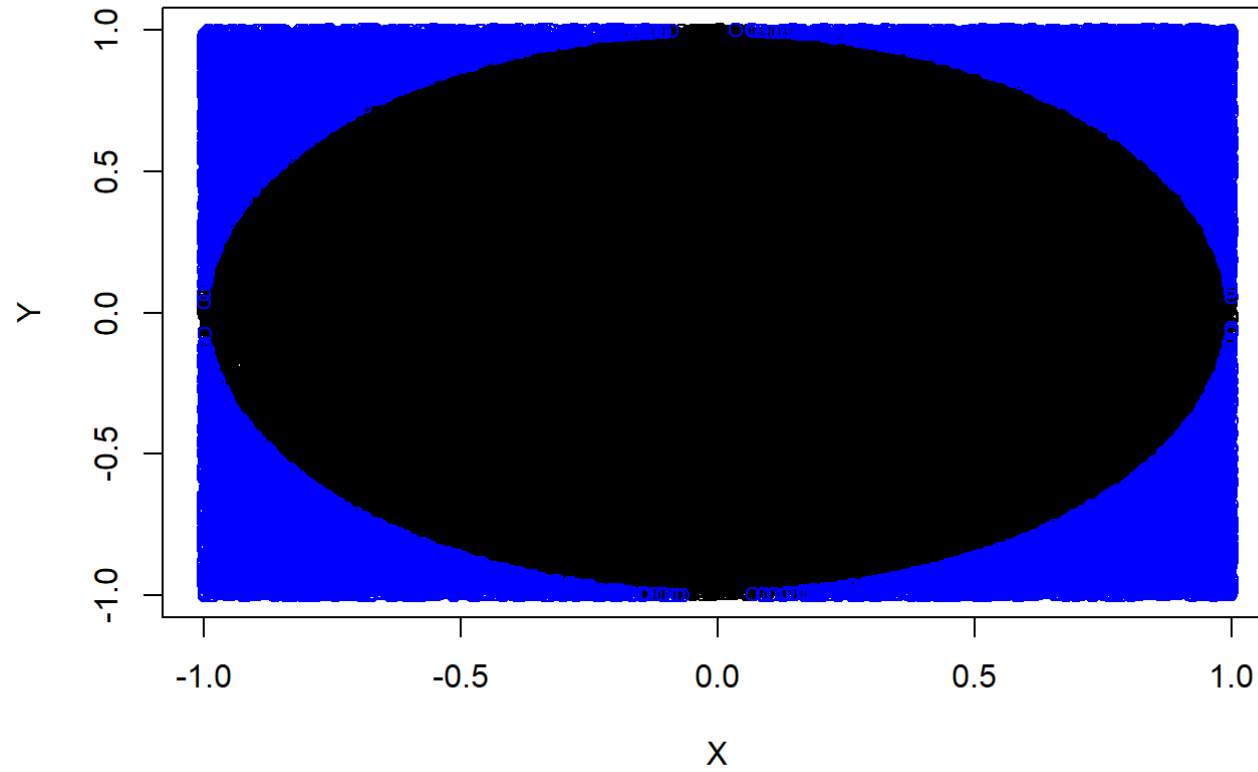
Simulation using 10,000 Darts


```
## [1] 3.1292
```

The Use of 100,000 Darts

This is our finished product. Our algorithm has stopped. Looks good doesn't it? How is the estimation of Pi? Our number stops below 3.14 right? This gives us an accurate representation of the estimation of pi using the Monte Carlo Method. If you would like, you could do this simulation over and over again to get differnet results each time. Have fun!

Simulation using 100,000 Darts


```
## [1] 3.14096
```