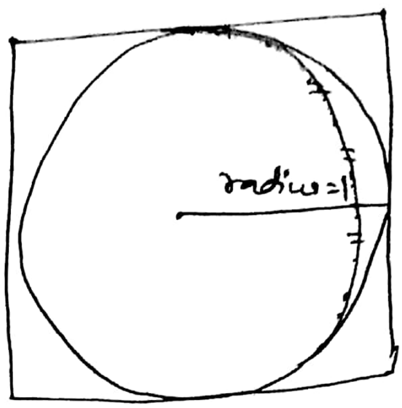
Estimating 'pi' by throwing darts.

• We will discuss a method to approximate 'pi' by throwing darts at a dart board by using Monte Carlo simulation.



length = 2.

Fig: 1.

Let us consider a black-board of length 2 ~~cm~~ and let the dart-board be a circle fitting perfectly inside the square with radius 1 (eg- Fig. 1).

Now, the circle with radius 1 has area 'pi'; whereas the square with sides of length 2 has an area of 4. So, the ratio of the area of the circle to the area of the square is 'pi/4' or $\pi/4$.

If we repeatedly throw darts at the board and they are to land randomly within the square, some will land on the square & some will land on the circular dartboard. If these throws are truly random, then the no. of darts that land on the dartboard, divided by the no. that we throw in total, will be the ratio described above (i.e. $\pi/4$).

So, $\frac{\pi}{4} \times 4 \approx \pi$ i.e. if we multiply the no. by 4, we have an estimate of π.

Scanned by CamScanner

Now, one thing remains is to ensure if the dart lands within the circle or not. Let, each point in which the dart pinns, has an ~~cord~~ co-ordinate (x, y) between ~~(-1, 1)~~ -1 and 1 [(0, 0) being the center of the dart board). Using Pythagoras' theorem, if

$$\sqrt{x^2+y^2} < 1,$$ then we know the dart landed inside the circle.

<u>R-code to estimate π:</u>

```
pi_est = data.frame(dart_throw = numeric(0), pi_est =
                    numeric(0)).
# The above creates an ~~data~~ empty data-frame to
  append the estimates.
for (i in 2^(2:25))
{
      throws = i
      x_cor = runif(throws, 0, 1).
      y_cor = runif(throws, 0, 1).
      dart_in_circle = sqrt((x_cor^2 + y_cor^2)).
      dart_in_circle_total = sum(dart_in_circle < 1).
      pi_est = data.frame(dart_throw = throws,
                          pi_est = (dart_in_circle_total/
                                    throws)*4)
      pi_est_final = rbind(pi_est_final, pi_est)
}.
```

Run this program in R to get an estimate of π.