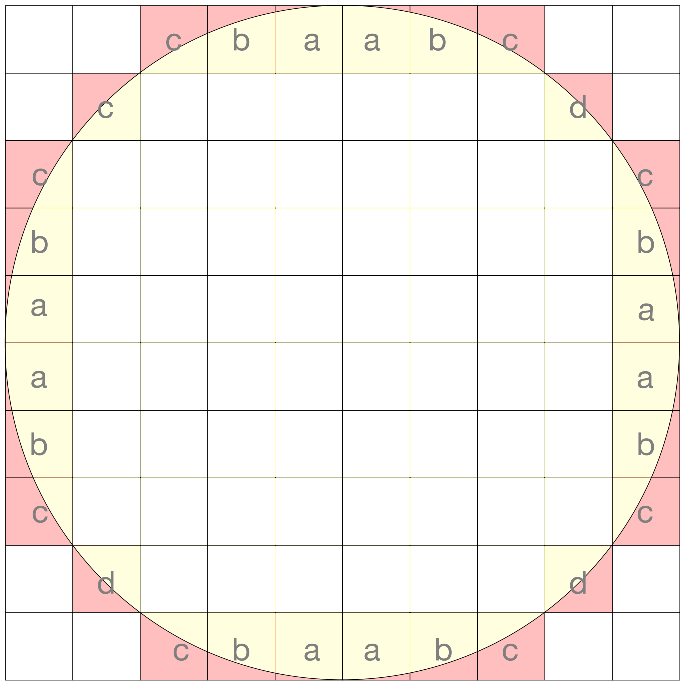
Module 14 — The Module random: Activities

1. Implement the Monte Carlo method in the presentation to approximate the area of the unit circle and experimentally, determine convergence towards the true value for 100, 1 000, 10 000, 100 000, 1 000 000 and 10,000,000 sample points.



2. Some of the disappointing results in the quest for higher precision is that we waste a lot of work. The figure above shows a unit circle superimposed of a grid of 0.2 by 0.2 squares. There is a total of 100 squares, of which 12 lie completely outside the circle, 60 lie completely inside the circle, and only 28 lie partially in the circle. Just by looking at this picture, we know that the area of the unit circle is between $60 \times 0.2^2 = 2.4$ and $4 - 12 \times 0.2^2 = 3.52$. By symmetry, we see that the 28 squares that overlap with the circle fall into only 4 categories defined in the figure above. We have 8 each of types A, B, and C and four of type D. To obtain a better estimate of the area of the circle, we can use Monte-Carlo to just determine the area of the intersection of the circle with one of those squares. For instance, in order to calculate the area of the intersection of circle and square of type B, we use the right B square on the top. The random numbers are generated with an x-value of $0.2 + u$, where u is a random number between 0 and 0.2 and an y-value of $0.8 + v$, where v is also a random number between 0 and 0.2.

```
import random

def calc_b(samples):
    count = 0
```

```
    for _ in range(samples):  #do samples times
        x = 0.2+random.uniform(0,0.2)
        y = 0.8+random.uniform(0,0.2)
        if x*x+y*y<1:
            count += 1
    return count/samples*0.2*0.2  # box is 0.2 times 0.2

for s in [100, 1000, 10000, 100000]:
    for _ in range(5):
        print(s, calc_b(s))
```

Calculate the areas of the squares of types A, C, and D as well using 100,000 samples and then compare the result with the true area of the circle.