**Introduction to R**
**Special Exercise**

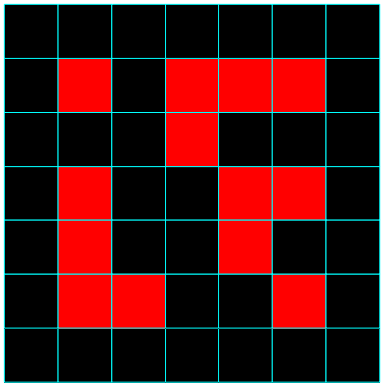
This exercise is for you to try in the evening session. We will review various solutions to it in the final session – so even if you don't get a complete solution, tackling some of it will be useful preparation for the final session.

The goal is to implement Conway's Game of Life in R. This 'game' is a simple evolutionary model, where cells on a grid either 'live' or 'die' according to the following rules;

1. Any live cell with fewer than two live neighbors dies, as if caused by under-population.
2. Any live cell with two or three live neighbors lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by overcrowding.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

At each generation, we count the neighbors, then do all the updates to status (i.e. live/dead). For example;

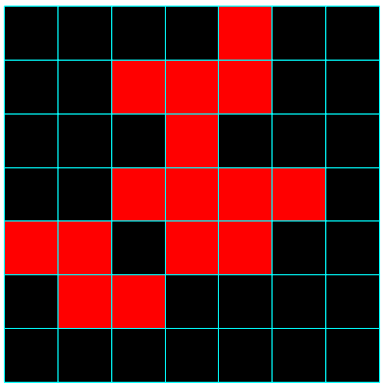
First Generation



Count of Neighbors

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 2 | 3 | 2 | 1 |
| 1 | 0 | 3 | 2 | 3 | 1 | 1 |
| 2 | 2 | 4 | 3 | 6 | 4 | 2 |
| 2 | 1 | 3 | 3 | 3 | 2 | 1 |
| 3 | 3 | 4 | 3 | 3 | 4 | 2 |
| 2 | 2 | 2 | 2 | 2 | 1 | 1 |
| 1 | 2 | 2 | 1 | 1 | 1 | 1 |

Second Generation



Your code should plot the grid of alive/dead cells in the R graphics window, for several generations. Some animations showing examples are on the course site.

Some tips: (note we'll give more tips in the evening session)

- Start with a small grid, i.e. 10x10, but write your code so that it can be 'scaled up' later
- Use the `rect()` function to draw the grid
- At each generation, use a double `for()` loop, i.e.

```
for(i in 1:nrows){
   for(j in 1:ncols){
      <do some operations on cell[i,j]>
   }}
```

- There are two options for what to do with the edges and corners;
  - You may count the neighbors based on the visible grid, so e.g. cells on the edges can have 5 neighbors, while those in the corners have up to 3. With this approach, your code for counting neighbors will be different for edge cells, corner cells, and regular internal cells.
  - You may 'wrap around' the definition of neighbors, so e.g. the left edge cells neighbor those on the right edge. With this approach, 'modular arithmetic' will be useful; to do this in R try e.g. `1:20 %% 7`, to generate the 'remainders' when you divide 1:20 by 7.
- Once your code works on small examples, try it on larger grids, initiated at random. For example, `matrix(rbinom(40*40, 1, 0.3),40,40)` generates a 40x40 matrix with entries that have 30% chance of being 0, and 70% chance of being 1.
- [For keen people] Keep track of how long a cell has been alive, and color-code according to age.