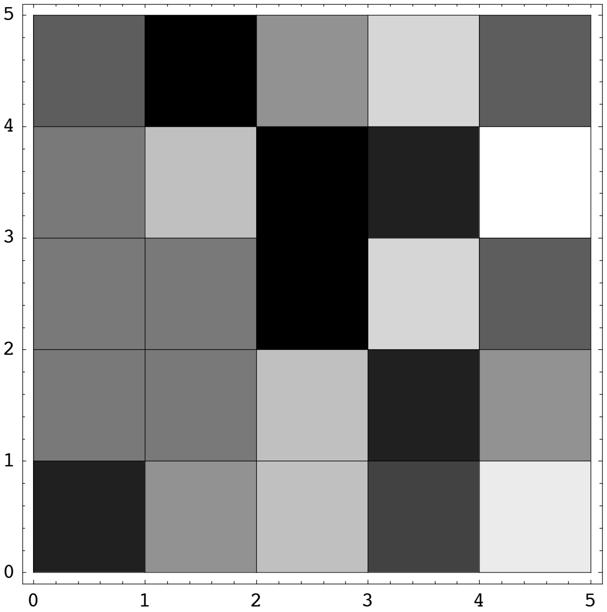
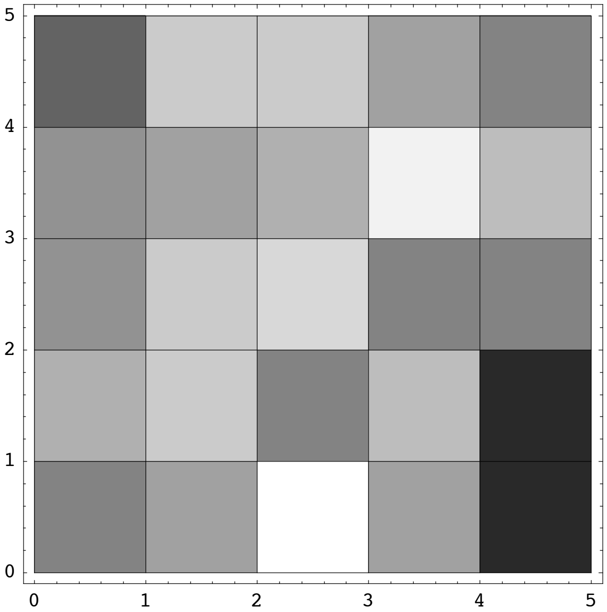
■ Poisson clumping

■ Samples

Which of the following grids represents a random process at work (ie, cells picked at random) and which represents the "random" choices you picked in class? It's pretty hard to tell from these pictures. What would be a better way to discriminate between the two?





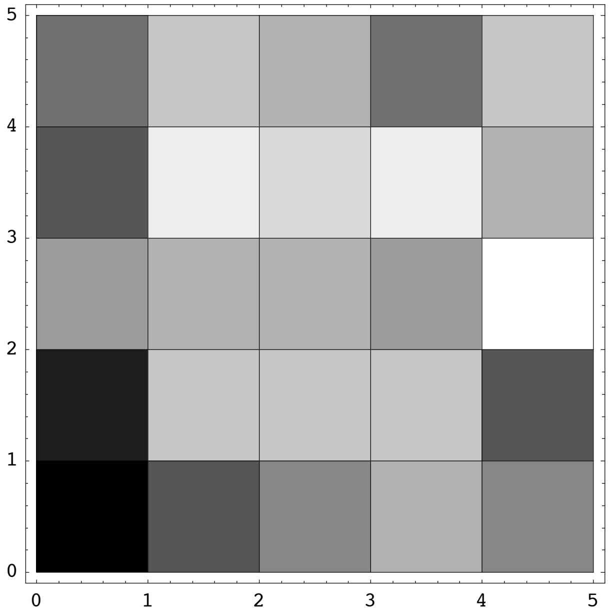
■ Generating grids from choices

Start off by picking the **size** of the array. The array will have size^2 elements. Now "populate" the grid by incrementing the grid elements as determined by the cells **pick**ed in the input list; *Mathematica* uses ++ to denote the "add one" operation as in the computer language C. So, ++grid[i] increments the count associated with cell i. The input list indicates which cells to increment. The following function shows the plot of the cells as a colored grid (white for empty, black for those with more "hits") as a side effect, and returns the list of grid counts (ie, how many hits in cell 1, how many in cell 2, etc).

```
In[1]:= drawGrid[size_, picks_] := Module [
          {grid = Table[0, {size^2}]},
          ++grid⟦#⟧ & /@ picks;
          ListDensityPlot[Partition[grid, size]];
          grid
         ]
```

■ Random grid

```
In[10]:= With[{size = 5},
           randomChoices = Table[Random[Integer, {1, size^2}], {230}];
           randomGrid = drawGrid[size, randomChoices]];
```



■ Class grid

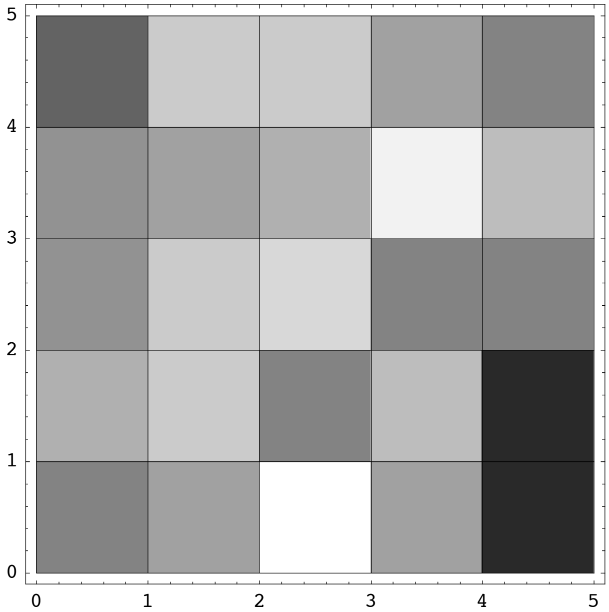
Here are the "random" choices collected in class.

```
In[15]:= class = {4, 8, 18, 21, 25, 11, 8, 13, 18, 20, 6, 9, 16, 18, 23, 2, 9, 16, 22, 24, 3, 7, 12, 16,
            19, 1, 17, 19, 12, 7, 11, 13, 19, 22, 22, 7, 13, 21, 16, 3, 8, 13, 17, 22, 1, 3, 7,
            20, 23, 3, 6, 14, 22, 24, 1, 2, 3, 4, 5, 8, 14, 15, 22, 23, 1, 7, 13, 15, 19, 25, 2,
            17, 22, 23, 24, 1, 9, 17, 19, 21, 2, 6, 12, 16, 22, 3, 6, 17, 20, 23, 3, 4, 9, 14,
            20, 6, 9, 11, 12, 14, 3, 8, 11, 18, 24, 3, 12, 18, 19, 23, 6, 13, 17, 22, 24, 13, 4,
            21, 17, 2, 2, 9, 12, 20, 23, 3, 7, 15, 16, 20, 7, 12, 15, 20, 24, 1, 3, 14, 20, 25,
            11, 13, 17, 19, 23, 19, 9, 21, 18, 23, 3, 7, 11, 19, 23, 2, 4, 6, 12, 24, 4, 8, 14,
            23, 24, 1, 3, 10, 12, 19, 6, 9, 9, 13, 24, 3, 12, 16, 19, 19, 3, 9, 14, 16, 20, 5,
            12, 18, 22, 25, 2, 7, 15, 18, 19, 7, 10, 13, 25, 25, 4, 6, 13, 15, 22, 7, 12, 13,
            19, 23, 7, 13, 17, 20, 25, 3, 8, 11, 18, 22, 2, 9, 11, 15, 20, 4, 4, 6, 18, 19};

         Length[class] / 5

         46

In[16]:= classGrid = drawGrid[5, class]
```

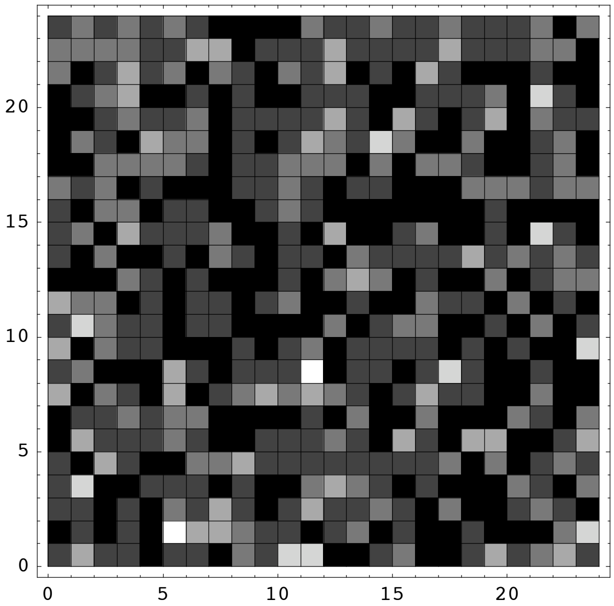


```
Out[16]= {7, 9, 16, 9, 2, 10, 12, 7, 11, 2, 8, 12, 13, 7, 7, 8, 9, 10, 15, 11, 5, 12, 12, 9, 7}
```

■ A larger random grid

Some larger grids are perhaps more interesting. Here's one with size on the order of that for London in the V-1 example. That grid had 576 elements, which is 24^2 .

```
With[{size = 24},
 drawGrid[size,
  Table[Random[Integer, {1, size^2}], {600}]]]
```

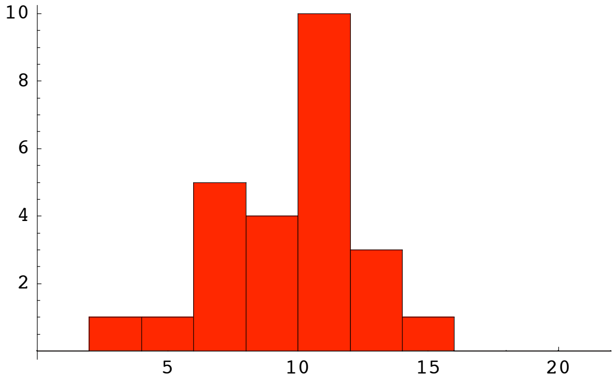


■ Histograms and the Poisson

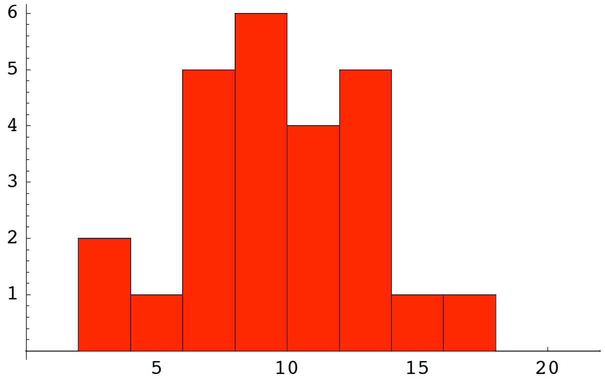
These commadns load some optional functions.

```
In[13]:= << Graphics`Graphics`

         Histogram[randomGrid,
           HistogramRange → {0, 20}, HistogramCategories → 10];
```



```
In[34]:= Histogram[classGrid,
           HistogramRange → {0, 20}, HistogramCategories → 10];
```



We can use the Poisson PMF as an approximate model for what ought to happen under random behavior.

$$p[x\_] \ := \mathrm{With}\left[\{\lambda = 230 / 25\}, \frac{e^{-\lambda}\lambda^{x}}{x\,!}\right]$$

Now count how many cells get no hits, one hit, two hits, and so forth.

```
In[35]:= randomCounts = Table[ Count[randomGrid, _? (# == x &)], {x, 0, 20}]
         classCounts  = Table[ Count[classGrid, _? (# == x &)], {x, 0, 20}]

Out[35]= {0, 0, 0, 1, 1, 0, 3, 2, 2, 2, 5, 5, 1, 2, 1, 0, 0, 0, 0, 0, 0}

Out[36]= {0, 0, 2, 0, 0, 1, 0, 5, 2, 4, 2, 2, 4, 1, 0, 1, 1, 0, 0, 0, 0}
```

Compare these to what the Poisson model predicts. With any of 25 cells possible, the share of these 25 devoted to any one position is just 25 times the Poisson probability for this many in a cell.

```
In[38]:= poisson = Table[ 25. p[x], {x, 0, 20}]

Out[38]= {0.00252599, 0.0232391, 0.1069, 0.327826, 0.753999, 1.38736,
          2.12728, 2.79586, 3.21524, 3.28669, 3.02375, 2.52896, 1.93887, 1.37212,
          0.901679, 0.55303, 0.317992, 0.17209, 0.087957, 0.0425897, 0.0195913}
```

Look at the squared errors between what the Poisson predicts and what happens. Which gets closer to the Poisson predictions?

```
In[39]:= Plus @@ (randomCounts - poisson)^2

Out[39]= 18.72

In[40]:= Plus @@ (classCounts - poisson)^2

Out[40]= 23.0117
```