18.06 Computational PSet 3

You may use any computer language. For this set it is best if your language has a row reduced echelon form command such as `rref` but you can use the code in your textbook to write your own in your favorite language. Please submit printouts with the problem set.

Computers can be very useful to detect patterns. If a pattern becomes clear, one can write down the pattern. Mathematicians would call the pattern a conjecture, because even if the pattern holds for $n = 1, 2, 3, \ldots, 10^6$, there is no certainty that it will hold for $n = 10^6 + 1$. Very often the pattern will continue forever, though not always. The only way to be sure is to have a mathematical argument that is valid for all n, not just up to one million.

In this exercise we want you to use the computer to form a conjecture. We are not asking for a proof of this conjecture. The goal of this exercise is learn to use the computer to detect patterns.

Take the $n \times n$ matrix A_n that is 1 on the diagonal, the subdiagonal, and the super diagonal and 0 everywhere else. For example,

$$A_5 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

In Julia, this is

In[1]:
```
A(n)=[int(abs(i-j)<2) for i=1:n, j=1:n]
A(5)
```

Out[1]:
```
  5x5 Array{Int64,2}:
1  1  0  0  0
1  1  1  0  0
0  1  1  1  0
0  0  1  1  1
0  0  0  1  1
```

We would like you to look at `rref(n)` only for perhaps $n = 1, 2, \ldots, 20$.

1) Print out `rref(A(8))` .

2) Without any further computation, use the result of the above to immediately write down the nullspace of `A(8)`.

3) Investigate `rref(A(n))` for a range of n. Do not print your results. In Julia, a fancy convenient

way to do this investigation is

In[2]:
```
using Interact
```

In[3]:
```
@manipulate for n=1:30
    int(rref(A(n)))
end
```

Based on what you see, form a conjecture about the nullspace of A_n that you believe to be valid for all n. We are not asking you to prove your conjecture mathematically.