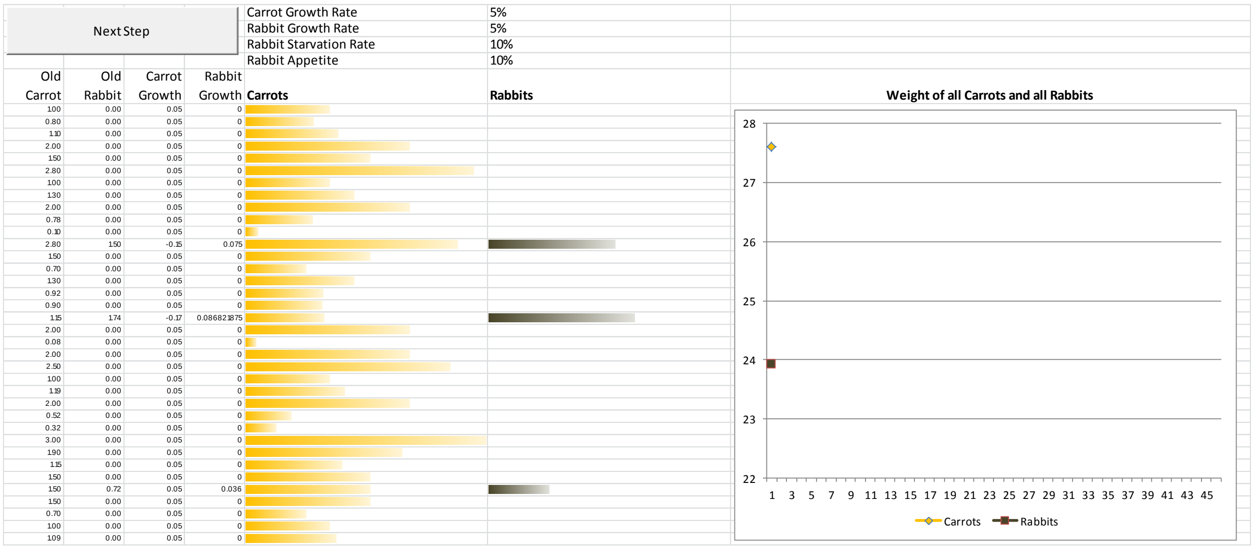
Carrot and Rabbit Simulation

Bruce Schafer, bruce@schafer.net

February 23, 2014

Compatibility

This simulation has been successfully tested on PC Excel 2010 and Mac Excel 2011.



Carrot Growth Rate	5%
Rabbit Growth Rate	5%
Rabbit Starvation Rate	10%
Rabbit Appetite	10%

Old Carrot	Old Rabbit	Carrot Growth	Rabbit Growth	Carrots	Rabbits
1.00	0.00	0.05	0		
0.80	0.00	0.05	0		
1.10	0.00	0.05	0		
2.00	0.00	0.05	0		
1.50	0.00	0.05	0		
2.80	0.00	0.05	0		
1.00	0.00	0.05	0		
1.30	0.00	0.05	0		
2.00	0.00	0.05	0		
0.78	0.00	0.05	0		
0.10	0.00	0.05	0		
2.80	1.50	-0.15	0.075		
1.50	0.00	0.05	0		
0.70	0.00	0.05	0		
1.30	0.00	0.05	0		
0.92	0.00	0.05	0		
0.90	0.00	0.05	0		
1.15	1.74	-0.17	0.086821875		
2.00	0.00	0.05	0		
0.08	0.00	0.05	0		
2.00	0.00	0.05	0		
2.50	0.00	0.05	0		
1.00	0.00	0.05	0		
1.19	0.00	0.05	0		
2.00	0.00	0.05	0		
0.52	0.00	0.05	0		
0.32	0.00	0.05	0		
3.00	0.00	0.05	0		
1.90	0.00	0.05	0		
1.15	0.00	0.05	0		
1.50	0.00	0.05	0		
1.50	0.72	0.05	0.036		
1.50	0.00	0.05	0		
0.70	0.00	0.05	0		
1.00	0.00	0.05	0		
1.09	0.00	0.05	0		

What it does

The setting for the simulation is a carrot patch where there can be up to 40 carrots and 40 rabbits. It starts with a specific number of carrots and rabbits of various sizes. In each step the carrots grow a little unless they are being eaten by a rabbit in which case they shrink. Each rabbit grows if it has a carrot to eat but shrinks if it doesn't. During each step each rabbit moves to another carrot if its current carrot is nearly gone. Each rabbit randomly chooses to move up or down to find another carrot. If there is no carrot in the direction it chooses it stays where it is.

The amount a carrot grows in each step that is not being eaten, is determined by the Carrot Growth Rate. If a carrot is being eaten the amount it shrinks is determined by Rabbit Appetite. If a rabbit has a carrot to eat, the amount it grows during a step is controlled by Rabbit Growth Rate. If it does not have a carrot, the amount it shrinks is determined by Rabbit Starvation Rate. You can change any of these values and run the simulation to see what effect the change has on the number and size of carrots and rabbits. It's best to load a fresh copy of the Excel file before changing the variables so you're starting with the same number of Carrots and Rabbits each time.

The Old Carrot and Old Rabbit columns give the size of the carrots and rabbits at the beginning of each step. These values can be changed. For instance, to add a carrot to the simulation, set the size value of one of the values in the Old Carrot column to something larger than 0.2.
The amount each carrot and each rabbit will grow is shown in the Carrot Growth and Rabbit Growth columns. Shrinkage due to a carrot being eaten or a rabbit starving is shown as negative

numbers in these columns. The size of each carrot and each rabbit at the end of each step is shown by the orange or brown bar in the Carrot and Rabbit columns.   The total weight of the carrots and the total weight of the rabbits are graphed in the chart at the right.

To initiate the next step, click once on the Next Step button in the upper left.

To change the number of carrots at the beginning of the simulation, make changes to the values in the Old Carrot column.   Likewise, to make changes to the number of rabbits, make changes to the Old Rabbit column.

How it works.

The formulas in the Carrot Growth and the Rabbit Growth columns use the input values (independent variables) described earlier.   The formula for Carrot Growth uses the values of Carrot Growth Rate and the Rabbit Appetite.

To be more specific if there is a rabbit of size 1 or larger in the row and if the carrot size is larger than 0, the Carrot Growth is set to Rabbit Appetite multiplied by the size of the rabbit as a negative value because the carrot will shrink. If the there is no rabbit of size 1 or larger in the row and there is a carrot, the Carrot Growth is set to the Carrot Growth Rate.

Turning to the Rabbit Growth, if there is a carrot in the same row as the rabbit grows by its current size multiplied by Rabbit Growth Rate. If there is no carrot in the row, the rabbit shrinks by its current size multiplied by Rabbit Starvation Rate.

The Carrot values are calculated as the sum of the Old Carrot size plus the Carrot Growth.   Likewise, the Rabbit values are calculated as the sum of the Old Rabbit value plus the Rabbit Growth.

The Visual Basic Programs

```
Sub Move_Rabbits()
Application.Calculation = xlManual 'Set the calculation mode to manual
For i = 5 To 44
  Cells(i, 1).Value = Cells(i, 5) 'Copy the new carrots to the old carrot position
Next i
For i = 1 To 40 'Copy the new rabbits to the old rabbit positions but move them around if no carrot
  From_Center = Round(i / 2 - 0.49) 'distance from center of carrots
  Direction = (i Mod 2) * 2 - 1 'direction from center of carrots
  j = From_Center * Direction + 25 'Start with the rabbits in the middle and work outword
  If Cells(j, 6).Value > 0 Then 'If there's new rabbit in this row
    If Cells(j, 1).Value >= 0.2 Then 'If there is significant carrot left for the rabbit
      Cells(j, 2).Value = Cells(j, 6).Value 'Leave it where it is by copying it to old rabbit in same row
    ElseIf Rnd() > 0.5 Then   'otherwise try to move the rabbit by randomly choosing up or down
      Find_Empty_Row_Above (j)
    Else
      Find_Empty_Row_Below (j)
    End If
  End If
```

```
Next i
Calculate 'Update the spreadsheet by recalcuting all the cells
'Record the sum of weights of the carrots and rabbits so they are graphed
j = Cells(46, 3).Value 'Current row for recording the weights
Cells(j, 1).Value = Cells(46, 5).Value 'Record carrot weight
Cells(j, 2).Value = Cells(46, 6).Value 'Record rabbit weight
Cells(46, 3) = j + 1 'Update the row to be used for recording the next weight.
Application.Calculation = xlAutomatic 'Reset the calculation mode to automatic
End Sub
```

The simulation of the rabbits’ movements to find carrots requires a Visual Basic program.   This can be viewed by clicking on the Developer tab and then click on Visual Basic. The main program is called Move_Rabbits and found at the bottom of the Visual Basic window. It starts by turning off Excel's automatic recalculation feature so the values in the Old Rabbit column can be changed with immediately being reflected in the values in the Rabbit column. Before moving the rabbits it copies the values in the Carrot column 5 to the Old Carrot column 1.

It then repeats a series of steps for each of the 40 rabbits. Rather than starting with the first rabbit in row 5, it starts with the rabbit in the middle row and moves up and down from there.   The selection of rows in controlled by the formulas for from_center and Direction. The actual row number is calculated as j based on from_center and Direction.

For each rabbit it proceeds as follows: If the size of the rabbit is greater than zero it does a series of things, otherwise there is no rabbit to worry about. The series of things starts by considering whether the carrot in the current row is size 0.2 or larger. If so, the rabbit has something to eat so the value in the Rabbit column 6 is copied to the Old Rabbit column 2.   Otherwise, it uses the Visual Basic random number function called Rnd to generate a number between 0.0 and 1.0.   If that number is greater than 0.5 it calls a subroutine called Find_Empty_Row_Above. Otherwise is called a subroutine called Find_Empty_Row_Below.

```
Sub Find_Empty_Row_Below(Current_Row As Integer) 'Try to find an empty row below rabbit's current row
   For j = 1 To 49 'Search down for an empty row for the rabbit
    New_Row = Current_Row + j
    If New_Row > 44 Then 'no more rows that might have carrots below; must leave rabbit where it is
      Cells(Current_Row, 2).Value = Cells(Current_Row, 6).Value
      Exit Sub
    Else 'Check to see if no existing rabbit in row and carrot available
      If Cells(New_Row, 2).Value = 0 Then 'No rabbit in this row
       If Cells(New_Row, 1).Value >= 0.2 Then 'Carrot available
        Cells(New_Row, 2).Value = Cells(Current_Row, 6).Value 'Move rabbit to this empty row
        Cells(Current_Row, 2).Value = 0 'Mark the rabbit's previous spot as vacant
        Exit Sub
       End If
      End If
    End If
   Next j
End Sub
```

Let's consider the subroutine Find_Empty_Row_Below, which can be found at the top of the Visual Basic window. Its purpose is to find a row below a rabbit's Current_Row where there is no rabbit but there is a carrot. Depending where which row the rabbit is currently it may need to check up to 49 other rows. It uses the variable j to do this. The row number of New_Row that may have a carrot and no rabbit is calculated as Current_Row + j. If New_Row is greater than 44 than we're past the last row that might have a carrot so the size of the rabbit is copied in the Current Row from the Rabbit column 6 to the Old Rabbit column 2. In other words the rabbit doesn't move to a new row. Otherwise the subroutine proceeds to check whether there is a rabbit in the New_Row and whether there is a carrot of at least size 0.2. If both things are true, the size value for the rabbit in the Current_Row and the Rabbit column 6 is copies to the cell in the New Row and the Old Rabbit column 2. Because the rabbit is now longer in the Current_Row, the value in the Old Rabbit column 2 of the Current_Row is set to zero. If the New_Row either has an existing rabbit or is missing a carrot of an edible size, the subroutine keeps looking by proceeding to the next value of j and thus the new row below where it just looked.

```
Sub Find_Empty_Row_Above(Current_Row As Integer) 'Try to find an empty row above rabbit's current row
    For j = 1 To 49 'Search up for an empty row for the rabbit
      New_Row = Current_Row - j
      If New_Row < 5 Then 'no more rows above; must leave rabbit where it is
        Cells(Current_Row, 2).Value = Cells(Current_Row, 6).Value
        Exit Sub
     Else 'Check to see if no existing rabbit in the row and a carrot is availalbe
        If Cells(New_Row, 2).Value = 0 Then 'No rabbit in this row
          If Cells(New_Row, 1).Value >= 0.2 Then 'Carrot available
            Cells(New_Row, 2).Value = Cells(Current_Row, 6).Value 'Move rabbit to this empty row
            Cells(Current_Row, 2).Value = 0 'Mark the rabbit's previous spot as vacant
            Exit Sub
          End If
        End If
     End If
    Next j
End Sub
```

The operation of the Find_Empty_Row_Above subroutine is very similar to Find_Empty_Row_Below except that it checks rows above the Current Row.

Examples of possible enhancements to the simulation

While the simulation is somewhat realistic in that it shows the some of the effect of carrots growing and rabbits eating them, there are many aspects of a real carrot and rabbit patch that are not adequately represented.

1. Real carrots will stop growing if a rabbit eats the green part at the top. How would you change the simulation to reflect this?

2. A wild carrot patch has many carrot seeds from previous seasons. As rabbits and other things disturb the soil the seeds may be exposed to the right amount of moisture and heat and germinate. How would you change the simulation to reflect this?

3. The simulation assumes that a rabbit will continue to grow as long as there is food available. While that is somewhat true of real rabbits, a young rabbit will tend to grow faster than an older rabbit and an older rabbit would grow by gaining weight, which might cause it to move more slowly. How would you change the simulation to reflect these factors?

Advanced enhancements

4. The simulation reflects rabbit population decreases due to starvation but it doesn't have a way for the population to increase. In real life, well-fed adult rabbits will produce baby rabbits. How would you change the simulation to reflect this?

5. In most wild situations, rabbits would have predators. How would you add predators such a foxes or wolves to the simulation?