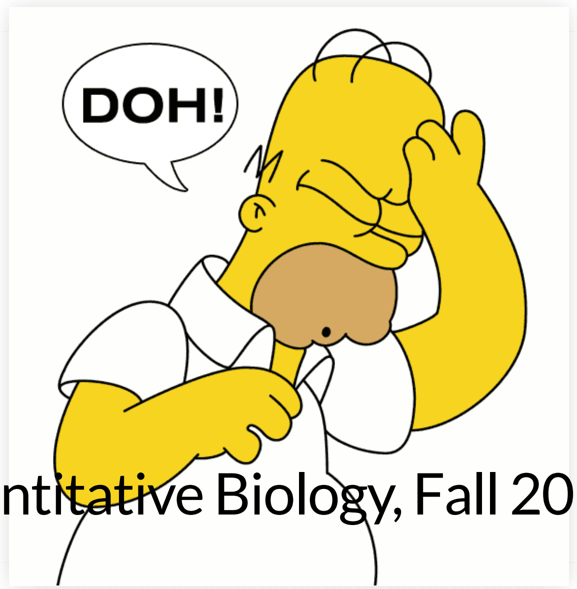
Testing Your Program (cont'd)

M. Drew LaMar
October 19, 2016



Introduction to Quantitative Biology, Fall 2016

Class announcements

- Reading quiz for Friday - Railsback & Grimm, Chapters 7 and 8 (NO QUIZ)
- Exam #2 discussion

Part II. Debugging Techniques

Statistical Analysis of File Output

Question: Is the probability butterflies move to the highest neighbor patch really q?

Answer: No. It is the approximate proportion

$$q + \frac{1-q}{8}.$$

For $q = 0.4$, we would expect the butterfly to move to the highest neighbor patch with probability 0.475.

Part II. Debugging Techniques

Statistical Analysis of File Output

```
file-type
file-print
file-open
file-close
```

Part II. Debugging Techniques

Statistical Analysis of File Output

```
mydata <- read.csv("TestOutput.csv",
header=FALSE)
str(mydata)
```

```
'data.frame':	1000 obs. of  9 variables:
 $ V1: num  15.6 16.9 16.3 18.9 17.7 ...
 $ V2: num  15.5 16.1 16.1 16.9 19.8 ...
 $ V3: num  14.8 15.4 17.7 18.4 20.5 ...
 $ V4: num  15.8 15.6 15.5 19.1 18.9 ...
 $ V5: num  15.6 16.3 16.8 18.2 18.4 ...
 $ V6: num  14.7 14.9 17.5 19.7 18.3 ...
 $ V7: num  15.5 14.8 18.3 17.5 19.7 ...
 $ V8: num  16.3 14.7 17 17.7 19.2 ...
 $ V9: num  15.5 16.9 18.3 19.1 18.3 ...
```

Part II. Debugging Techniques

Statistical Analysis of File Output

```
moved.to.highest <- sapply(1:1000, function (x)
{max(mydata[x,1:8]) == mydata[x,9]})

moved.to.highest <- as.integer(moved.to.highest)
```

Part II. Debugging Techniques

Statistical Analysis of File Output

```
prop.test(sum(moved.to.highest), 1000, p =
0.475)
```

```

    1-sample proportions test with continuity
correction

data:  sum(moved.to.highest) out of 1000, null
probability 0.475
X-squared = 16.683, df = 1, p-value = 4.418e-05
alternative hypothesis: true p is not equal to
0.475
95 percent confidence interval:
 0.3794248 0.4412753
sample estimates:
   p
0.41
```

Part II. Debugging Techniques

Statistical Analysis of File Output

Discuss: Wait, what?!? Result doesn't even contain 0.475 in the confidence interval. Explain why the estimate is smaller than 0.475.

Hint: What if you are at the top of a hill?

Regardless, something is wrong. We are having a problem with *verification*, i.e. the program is not doing what it is supposed to do.

Part II. Debugging Techniques

Independent Reimplementation of Submodels

Test movement submodel.

```
mydata <- read.csv("SubmodelOutput.csv",
header=FALSE)
str(mydata)
```

Part II. Debugging Techniques

Independent Reimplementation of Submodels

Test movement submodel.

```
'data.frame':   1000 obs. of  11 variables:
 $ V1 : num  15.6 15.6 17.8 19 19.9 ...
 $ V2 : num  16.3 16.8 15.9 18.9 18.9 ...
 $ V3 : num  15.5 15.9 15.8 16.9 18 ...
 $ V4 : num  15.6 14.9 16.8 18.9 19 ...
 $ V5 : num  15.5 16.9 15.9 17.9 20 ...
 $ V6 : num  14.8 14.9 17.9 17 18 ...
 $ V7 : num  15.8 16.6 16.9 16.9 20 ...
 $ V8 : num  14.7 14.8 17.9 18 17.9 ...
 $ V9 : num  0.939 0.474 0.277 0.193 0.699 ...
 $ V10: num  0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4
0.4 0.4 ...
 $ V11: num  15.8 16.9 17.9 19 18.9 ...
```

Part II. Debugging Techniques

Independent Reimplementation of Submodels

```
moved.up <- sapply(1:1000, function (x)
{max(mydata[x,1:8]) == mydata[x,11]})

should.moved.up <- (mydata[,9] < mydata[,10])

(diff <- which((moved.up == FALSE) &
(should.moved.up == TRUE)))
```

```
  [1]  50  51  64  65  82  85  86  87  88  89
101 106 109 119 120 123 142
 [18] 143 153 154 166 167 168 171 184 185 186
187 207 208 209 212 213 214
 [35] 218 229 230 233 237 238 241 246 252 276
277 280 283 317 337 338 353
 [52] 390 419 424 425 430 443 446 458 459 460
465 466 467 470 482 485 490
 [69] 491 492 493 498 499 523 526 532 533 577
594 595 604 609 610 614 619
 [86] 625 626 627 635 641 642 646 647 656 669
678 691 694 695 702 706 735
[103] 739 740 741 744 751 752 753 756 762 763
776 780 781 782 785 802 803
[120] 806 819 828 839 851 864 869 870 871 872
```

```
881 885 886 887 902 923 929
[137] 930 931 938 943 946 947 956 957 958 975
981 990
```

Part II. Debugging Techniques

Independent Reimplementation of Submodels

```
   V1 V2       V3 V4
50 49 49 48.58579 49
```

```
          V5       V6       V7 V8 V11
50 48.58579 48.58579 48.58579 49  50
```

Discuss: Given that V1-V8 are the neighbors elevation, and V11 is the elevation of the patch moved to, what does this tell you?

Answer: If at top of the hill, turtle stays put!

Part II. Debugging Techniques

Independent Reimplementation of Submodels

Discuss: Does this explain $q \neq 0.475$ discrepancy from earlier?

Answer: Yes! If at top of hill, will not be equal to max of neighbors.

Part II. Debugging Techniques

Independent Reimplementation of Submodels

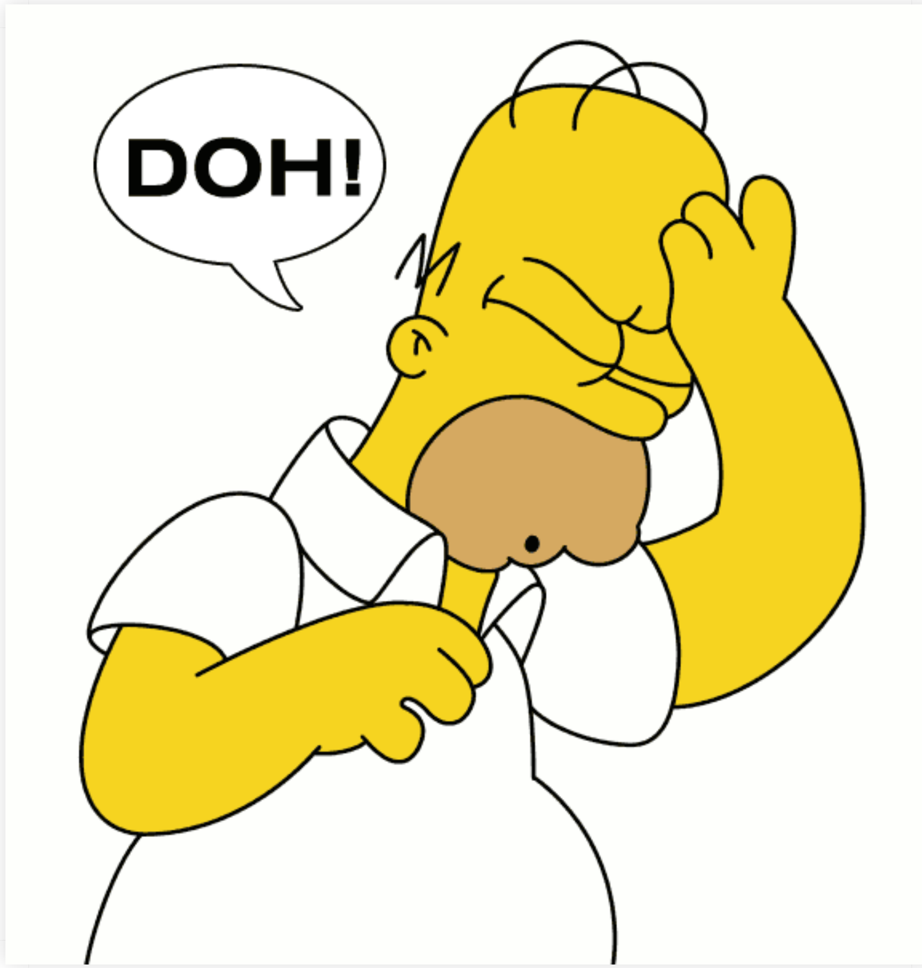
From ODD Submodels:

“..to 'move uphill' is defined specifically as *moving to the neighbor patch that has the highest elevation*; if two patches have the same elevation, one is chosen randomly. 'Move randomly' is defined as *moving to one of the neighboring patches*, with equal probability of choosing any patch. 'Neighbor patches' are the eight patches surrounding the butterfly's current patch.”

OOPS!

Part II. Debugging Techniques

Independent Reimplementation of Submodels



Part II. Debugging Techniques

Independent Reimplementation of Submodels

How to fix? Change

```
[uphill elevation]
```

to

```
[move-to max-one-of neighbors [elevation]]
```

This was a “Misunderstanding Primitives” error!!!