Session 4: AM

Research Design and Model Verification
Economic CAS

- Increasing returns
- Lock-in
- Network effects
- Implications for International Studies
Outline

- Research design
- Verification
- Experimentation
- NetLogo hints: building networks
General Research Design

- Theory
- Hypotheses
- Sampling and measurement
- Experimentation
  - Description
  - Inference: testing hypotheses
- Write up findings
Simulation Research Design

- Theory
- Hypotheses
- Observation of the target and measurement
- **Quasi** Experimentation
  - Model construction
  - Model verification (today)
  - Testing hypotheses (today)
- Write up findings

Session 4A: July 14
Model Construction

- Inductive vs. Deductive approaches (see week 5)
- Miller and Page: the “Eightfold Way”
➤ Method is subordinate to design

➤ Miller and Page, p. 246: “The most important feature of a computational model is the model, not the computer.”
Model Verification
"Verification"

- Is my code doing what I want it to do?
“Errors”

- Compile time
  - Model will not run
- Run time
  - Model will run
  - Pernicious
Suggestions

- Follow simple conventions
- Add extensive comments
- Debug
- Keep a log
- Maintain separate “versions” of the model
- Have another check your work
1. Conventions

- Name true/false variables with a “?”
- One line per instruction
- Indent
- Keep brackets on separate lines

```
globals [ police-size protester-size ]
patches-own [ walkable? ]

let target one-of patches with [ walkable? ]
sety: [ pcor ] of target [ pcor ] of target
set shape "person" set protester? false set militia? false set police? false

ifelse (random 1000 < 200) [ !false (random 300 < 50) [ set police? true set militia? false ] [ set police? false set militia? true ] ]
set protester? true

setup-patches]
```

Session 4A: July 14
2. Add Extensive Comments

- “Comments” are human-language remarks embedded in computer code
  - Explains the programmer’s logic for a particular procedure
  - Can identify procedures that are inactive or under development
- In NetLogo, use “;;” before each line of comments
Comments Example (1)

5th am Intro
5th pm NetLogo
7th am Theory
7th pm Modeling
12th am Emergence
12th pm Systems
14th am Design
14th pm Experiments
19th am Criticisms
19th pm Validation
21st am Smart Agents
21st pm Networks
26th Applications
28th Presentations

Session 4A: July 14

```plaintext
set my-payoff find-my-payoff my-status
ask my-partner [ set my-payoff find-my-payoff my-status ]

set partner-payoff [ my-payoff ] of my-partner
set [ partner-payoff ] of my-partner my-payoff

if (relative-gains?)
    set my-payoff weight-payoffs my-payoff partner-payoff
    ask my-partner
    [ set my-payoff weight-payoffs my-payoff partner-payoff ]

set my-total my-total + my-payoff
ask my-partner [ set my-total my-total + my-payoff ]

ask my-partner [ set have-played? true ]
set have-played? true

;; inspect self
;; inspect my-partner
;; user-message "halt"
```
Comments Example (2)

Session 4A: July 14

```plaintext
;; the selection rectangle and start over
if not any? selected [ deselect ]
end

to hand-drag
;; remember where the mouse pointer was located when
;; the user pressed the mouse button
let old-x mouse-xcor
let old-y mouse-ycor
if selected? old-x old-y [ 
  while [mouse-down?]
  let new-x mouse-xcor
  let new-y mouse-ycor
  ;; we need to move both the selected turtles and the sides
  ;; of the selection rectangle by the same amount that the
  ;; mouse has moved. we do this by subtracting the current
  ;; mouse coordinates from the previous mouse coordinates
  ;; and adding the results to the coordinates of the turtles
  ;; and sides.
  ask selected
    [ setxy xcor + new-x - old-x
    ycor + new-y - old-y ]
  ask sides
    [ setxy xcor + new-x - old-x
    ycor + new-y - old-y ]
  set old-x new-x
  set old-y new-y
  ;; update the view, otherwise the user can't see
  ;; what's going on
display
  ]
end
```
3. Debugging

➤ NOTE: No built-in debugger in NetLogo
➤ To “step into” NetLogo or to catch errors, use:

\begin{verbatim}
user-message "Error!"
\end{verbatim}
4. Keep a Log

- Easy to lose track of changes
- A log can remind you
5. Maintain Separate Versions

- Save both old and new versions of the model
  - e.g. “model v1.nlogo” “model v2.3.nlogo”
  - Preserves old code
  - Fall back if you “break” your model

- Create a new version with each major development
  - New agents
  - New plots, histograms, outputs
6. Check Each Other

- A second set of eyes
- Importance of good comments, standardized coding conventions
Questions?
Experimentation
Quasi-Experimentation

- An Example: cash flow
The Cash Flow Model

- Two parameters
  - Number of people $N$: $0 \leq N \leq 500$
  - Bank Reserve Ratio $R$: $0.00 \leq R \leq 1.00$

- Combinations: 50,000
  - 10 simulations per combination implies 500,000 simulations

- **Important issue**: how many simulations per combination?
Ethnocentrism
Ethnocentrism

- Eight parameters
  - Mutation rate: $0.000 \leq r \leq 1.000$
  - Initial potential to reproduce: $0.00 \leq ptr \leq 1.00$
  - Death rate: $0.00 \leq d \leq 1.00$
  - Cost of giving: $0.00 \leq c \leq 1.00$
  - Immigrants per day: $0 \leq i/day \leq 100$
  - Gain of receiving: $0.00 \leq g \leq 1.00$
  - P1 cooperate: $0.00 \leq p1 \leq 1.00$
  - P2 cooperate: $0.00 \leq p2 \leq 1.00$
Combinations

\[ 10^3 \times \left(10^2\right)^7 = 10^{17} \]
Quasi-Experimentation

- In NetLogo
  - Systematically vary parameters
    - Similar to true experimentation
  - Set a number of simulations per combination
  - Identify variables to measure
  - Define a “stop” condition
    - Time limit
    - Some other event
  - Record data to a spreadsheet
Session 4A: July 14

Behaviorspace

Cash Flow - NetLogo

Interface

Tools

Halt
Globals Monitor
Turtle Monitor
Patch Monitor
Link Monitor
Hide Command Center Ctrl+Slash
3D View
Color Swatches
Turtle Shapes Editor
Link Shapes Editor

BehaviorSpace
System Dynamics Modeler
HubNet Client Editor
HubNet Control Center

Savings & Wallets

1100

Money & Loans

Then add text here regarding the Behaviorspace tool and its features. Discuss how it can be used to model and analyze complex systems.
Systematically vary the model parameters

Simulations for each Parameter combination

Variables to measure

WARNING!

When to stop a given simulation
Sampling the parameter space

➤ How do you choose values when varying parameters?
➤ What are the risks?
5th am Intro
5th pm NetLogo
7th am Theory
7th pm Modeling
12th am Emergence
12th pm Systems
14th am Design
14th pm Experiments
19th am Criticisms
19th pm Validation
21st am Smart Agents
21st pm Networks
26th Applications
28th Presentations

Session 4A: July 14
Session 4A: July 14
Questions?
Data Analysis

➢ Ideas?

➢ Four suggestions
  ➢ Visual analysis
  ➢ Statistical analysis
  ➢ “Strategic ecology”
  ➢ Genetic algorithms
1. Visual Analysis

➢ The Cash Flow Model
Problems with Visual Analysis

- Limited to relationships between three variables
- Sensitive to choice of parameter values
- Impressionistic: what are decision criteria for hypothesis tests?
  - May be indeterminate
## 2. Statistical Analysis

### Correlations

<table>
<thead>
<tr>
<th></th>
<th>people</th>
<th>reserves</th>
<th>resXpeo</th>
<th>poorperc</th>
</tr>
</thead>
<tbody>
<tr>
<td>people</td>
<td>1.000</td>
<td>.004</td>
<td>.487**</td>
<td>.067</td>
</tr>
<tr>
<td>Sig. (2-tailed)</td>
<td>.945</td>
<td>1.000</td>
<td>.791**</td>
<td>-.958**</td>
</tr>
<tr>
<td>N</td>
<td>300.000</td>
<td>300</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>reserves</td>
<td>.004</td>
<td>1.000</td>
<td>.791**</td>
<td>-958**</td>
</tr>
<tr>
<td>Sig. (2-tailed)</td>
<td>.945</td>
<td>1.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>N</td>
<td>300</td>
<td>300.000</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>resXpeo</td>
<td>.487**</td>
<td>.791**</td>
<td>1.000</td>
<td>-.738**</td>
</tr>
<tr>
<td>Sig. (2-tailed)</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>N</td>
<td>300</td>
<td>300</td>
<td>300.000</td>
<td>300</td>
</tr>
<tr>
<td>poorperc</td>
<td>.067</td>
<td>-.958**</td>
<td>-.738**</td>
<td>1.000</td>
</tr>
<tr>
<td>Sig. (2-tailed)</td>
<td>.246</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>N</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>300.000</td>
</tr>
</tbody>
</table>

**. Correlation is significant at the 0.01 level (2-tailed).
### Regression Analysis

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs = 300</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>49756.8281</td>
<td>3</td>
<td>16585.6094</td>
<td>F(3, 296) = 2425.80</td>
</tr>
<tr>
<td>Residual</td>
<td>2023.80262</td>
<td>296</td>
<td>6.83717103</td>
<td>Prob &gt; F = 0.0000</td>
</tr>
<tr>
<td>Total</td>
<td>51780.6307</td>
<td>299</td>
<td>173.179367</td>
<td>R-squared = 0.9609</td>
</tr>
</tbody>
</table>

| poorpercap | Coef.    | Std. Err. | t     | P>|t| | Beta       |
|------------|----------|-----------|-------|-----|-----------|
| reserves   | -.5183937| .0145896  | -35.53| 0.000| -.9575322 |
| people     | .0057381 | .001755   | 3.27  | 0.001| .0617671  |
| resXpeo    | -.0000388| .000044   | -0.88 | 0.379| -.0271697 |
| _cons      | 45.28516 | .5820608  | 77.80 | 0.000| .           |

**NOTE:** This is static modeling

---

Session 4A: July 14
### Hazard Modeling

**Dependent Variable:** Survival time from beginning of fishing, measured in “ticks”

<table>
<thead>
<tr>
<th>Parameter</th>
<th>1. Accelerated failure-time estimates (with robust standard errors)</th>
<th>2. Exponentiated failure time estimates</th>
<th>3. Relative Hazard Estimates (with robust s.e.s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration Dependence</td>
<td>$\sigma = 1.122 (0.041)^{***}$</td>
<td>0.99</td>
<td>$a = .891 (.0322)^{***}$</td>
</tr>
<tr>
<td>Memory ($M$)</td>
<td>$-.002 (.007)$</td>
<td>0.99</td>
<td>$0.002 (.006)$</td>
</tr>
<tr>
<td>Fishing radius ($R$)</td>
<td>$-.974 (.028)^{***}$</td>
<td>0.38</td>
<td>$.868 (.040)^{***}$</td>
</tr>
<tr>
<td>Shared knowledge dummy ($I = shared$)</td>
<td>$5.01 (.177)^{***}$</td>
<td>149.90</td>
<td>$-4.46 (.234)^{***}$</td>
</tr>
<tr>
<td>Global knowledge dummy ($I = global$)</td>
<td>$1.51 (.106)^{***}$</td>
<td>4.53</td>
<td>$-1.35 (.085)^{***}$</td>
</tr>
<tr>
<td>Constant</td>
<td>$9.51 (.133)^{***}$</td>
<td>$-8.47 (.308)^{***}$</td>
<td></td>
</tr>
<tr>
<td>Log-likelihood</td>
<td>$-1463$</td>
<td>$-1463$</td>
<td></td>
</tr>
</tbody>
</table>

* Significant at $p < 0.05$
** Significant at $p < 0.01$
*** Significant at $p < 0.001$

+ Baseline institution is $i = local$.

Coefficients in column 1 are maximum likelihood estimates of a Weibull duration model. Robust standard errors are in parentheses.
May assumes linearity or log-linearity
  e.g. ordinary least squares

Risks overlooking interactive effects
  Use all possible interaction terms (cross products)

Risks overlooking nonstationarity, serial correlation
  Use hazard models, lagged values

Confuses readers
  An analysis of the model, not the target system
3. Strategic Ecology

- Measure distribution of agent strategies or attributes over time
  - See Miller and Page, p. 197 figure 10.8
4. Genetic Algorithms

➤ A form of artificial intelligence

➤ Computer learns where the “peaks” in the fitness landscape

➤ Reports parameter sets

➤ We’ll discuss this in two weeks
Questions?
Write Up Findings

- Abstract
- Introduction
- Literature Review
- Discuss emergent behavior of interest
- Describe variables/hypotheses
- Describe model in detail
- Discuss verification and validation
- Results & Next Steps
- Conclusions
NetLogo Hints
Networks

- **Actors**
  - “Vertices”
  - “Nodes”
  - *Turtles*

- **Relations**
  - “Edges”
  - *Links*
Network Types

- Classified by
  - Degree distribution
  - Average path length
  - Clustering coefficient

- Examples
  - “Small World”
  - “Random” (Erdos/Renyi)
  - “Scale-free”
Directed vs. Undirected Networks

- Directed ("digraph")
  - Relations between nodes are one-way
  - Order of the pairs of nodes matters
  - e.g. military personnel

- Undirected
  - Relations between nodes are two-way
  - Order of the pairs of nodes does not matter
  - e.g. spouses
NetLogo Primitives

- Any turtle/agent can be a node in a network
- “Links” between turtles themselves are agents
  - Have colors
  - Declare variables
  - Identifiers are two numbers:
    - E.g. “link 28 30”
    - The link that connects turtle 28 (end1) and turtle 30 (end2)
NetLogo Primitives

- Directed Graphs

```lilypond
ask turtles
[ create-links-to n-of 5 turtles with [ self != myself ] ]
```

- Undirected Graphs

```lilypond
ask turtles
[ create-links-with n-of 5 turtles with [ self != myself ] ]
```
NetLogo Example
Network Primitives

➤ NOTE: No innate measures of network properties
  ➤ Requires analysis outside of NetLogo
  ➤ See NetLogo Dictionary “Links” category
Networks

Questions?