Bridging agent-based modelling and R with `nlrx`: simulating pedestrian’s long-term exposure to air pollution

Hyesop Shin
Ph.D Candidate
Dept. of Geography and Fitzwilliam College
University of Cambridge
@hyesop
Objective
Agent-based modelling
R Codes
Outcomes
Summary

Files, Codes and Tutorials:
http://tiny.cc/nlrx
Project Objective
Air Pollution in South Korea  (March, 2019)

PM$_{2.5}$ > 200

Songdo, Incheon  (Mar.5th, 2019)

PM$_{2.5}$ > 140

Central Seoul  (Mar.6th, 2019)
Project objective

- This study aims to estimate pedestrian’s exposure to acute air pollution in Seoul districts using agent-based simulation
  - How does socioeconomic group potentially affect health outcomes?
  - How could health levels change under different pollution scenarios?
Agent-based modelling

Agent-based modelling (ABM) is a computational method that focuses on individual’s movements and interactions that can affect the system structure e.g. Social media, epidemics, decision-making

Components
- Agents = turtles
- Environments = patches
ABM setup
How ABM works: NetLogo example
Agent-based modelling: Settings

- Study area + demographic info
- Agent setup and create destination
- Measure health loss and recovery
- Export file as export.csv
- Import files to R

Gangnam district

39.5 km²

96,822 patches

Study Population
1% sample of Gangnam district

<table>
<thead>
<tr>
<th>Age Group</th>
<th>Male</th>
<th>Female</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Over 65</td>
<td>397</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15-64</td>
<td>4147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Under 15</td>
<td>506</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5050 agents were sampled
Agent-based modelling: Settings

- The population of Gangnam is allocated into three groups: Under 15, 15-64, Over 65
- A day is split into two time sequences: Work hours (09-19 hrs), Home (20-08 hrs)
- Agents have no previous exposure experienced
- Agents aged 15-64 follow OD matrix while restricting other groups' movement range close to their origin
Agent-based modelling: Settings

Study area + demographic info
Agent setup and create destination
Measure health loss and recovery
Export file as export.csv
Import files to R

\[
\frac{dH}{dt} = -\alpha (H_{\text{max}} - H(t)) + H_{\text{reco}}
\]

- If PM$_{10} > 100$
- \( H = \) health
- \( t = \) time
- \( \alpha = \) random float between 0 – 0.001
- \( H_{\text{reco}} = \) health recovery rate by real estate prices

Patches
Each patch considered as:
1. Roads
2. Pollution parcel

Agents
Health loss: set to decrease exponentially, assuming an individual loses a greater amount of health as it goes down
Health recovery: dependent on local real estate prices as a proxy for health
Agent-based modelling: Settings

- Study area + demographic info
- Agent setup and create destination
- Measure health loss and recovery
- Export file as export.csv
- Import files to R

1.2 hours for a single run

50 iterations

Low Quality Images

Exported as (uncleaned).csv
You will end up doing this...

Time-consuming!
Solving current problems from NetLogo

01 How can R reduce human intervention during analysis?

02 How can R improve the quality of figures?

03 How can R increase the simulation's clock speed?
ABM with nlrx:
*a tool to setup and execute NetLogo simulations from R*

*Netlogo + R + XML*
Stage 1: Install packages `nlrx`

**Java setup**
```
Sys.setenv(JAVA_HOME = 'XXXX')  #Varies by OS
```

**Load pkgs**
```
library(nlrx)
library(tidyverse)
library(rcartocolor)
library(ggthemes)
```

**Assign path**
```
nlpath <- file.path("/home/hs621/NetLogo 6.0.4")
modelpath <- file.path(path, "Gangnam.nlogo")
outpath <- file.path("/home/hs621/out")
```

**Create an nl object**
```
nl <- nl(nlversion = "6.0.4",
         nlpath = nlpath,
         modelpath = modelpath,
         jvmmem = 1024)
```

Codes and tutorials: [http://tiny.cc/nlrx](http://tiny.cc/nlrx)
Stage 2: Attach an experiment (1/2)

```r
nl@experiment <- experiment(expname = "seoul",
                           outpath = outpath,
                           repetition = 1,
                           tickmetrics = "true",
                           idsetup = "setup",
                           idgo = "go",
                           runtime = 8764,
                           evalticks = seq(1, 8764, by = 100),
)
```

Codes and tutorials: [http://tiny.cc/nlrx](http://tiny.cc/nlrx)
constants = list("PM10-parameters" = 100,
           "Scenario" = "BAU",
           "scenario-percent" = "inc-sce"),

variables = list('AC' = list(values=c(100,150,200))),

metrics.turtles = list("people" = c("xcor", "ycor", "homename", "destinationName", "age", "health")

metrics.patches = list("patch" = c("pxcor", "pycor", "pcolor"))
Stage 4: Attach a simulation design and run model

```
nl@simdesign <- simdesign_distinct(nl=nl, nseeds=1)

init <- Sys.time()
results <- run_nl_all(nl = nl)
Sys.time() - init

setsim(nl, "simoutput") <- results
write_simoutput(nl)
```

Codes and tutorials: [http://tiny.cc/nlrx](http://tiny.cc/nlrx)
Stage 5: Submit batch jobs to reduce time

```
#!/bin/bash
#
# Example SLURM job script for Peta4-Skylake (Skylake CPUs, OPA)
# Last updated: Mon 13 Nov 12:25:17 GMT 2017
#

# Modify the options in this section as appropriate

# sbatch directives begin here
# Name of the job:
#SBATCH -J nlrx
# Which project should be charged:
#SBATCH -A DITHELL-513-CPU
# How many whole nodes should be allocated?
#SBATCH --nodes=1
# How many (MPI) tasks will there be in total? (<= nodes*32)
#SBATCH --ntasks=32
# The Skylake/Skylake-himem nodes have 32 CPUs (cores) each.
#SBATCH --mem=99999
# How much wallclock time will be required?
#SBATCH --time=12:00:00
# What types of email messages do you wish to receive?
#SBATCH --mail-type=END
# Uncomment this to prevent the job from being requeued (e.g. if interrupted by node failure or system downtime):
##SBATCH --no-requeue

# For 6GB per CPU, set "-p skylake"; for 12GB per CPU, set "-p skylake-himem":
#SBATCH -p skylake-himem
```
Post-simulation
Result structure: nested tibble

⇒ unnested result

| General info: Iteration, Scenario, Random seed, Step, |
| Agent info: Count turtles, Breed, X, Y, home, destination, age, health |
| Patch info: X, Y, patch colour |
Mapping unhealthy population with \texttt{ggplot2}

\begin{verbatim}
  ggplot() +
  facet_wrap(~[step], ncol=10) +
  coord_equal() +
  geom_tile(data=patches, aes(x=pxcor, y=pycor, fill=pcolor), alpha=.2) +
  geom_point(data=turtles, aes(x=pxcor, y=pycor, color=age), size=1) +
  scale_fill_gradient(low="white", high="grey20") +
  scale_color_manual(breaks=c("young", "active", "old"), values=c("young"="#56B4E9", "active"="#E69F00", "old"="#999999")) +
  guides(fill=guide_legend(title="PM10")) +
  ggtitle("Unhealthy Population after a long-term exposure") +
  theme_minimal() +
  theme(axis.line=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks=element_blank())
\end{verbatim}

Codes and tutorials: \url{http://tiny.cc/nlrx}
Density plot with ggplot2 & Animations with gganmitate

Codes and tutorials: https://tiny.cc/nlrx
gridextra & directlabels for HQ images

direct.label()

grid.arrange()
Summary and Contribution
Summary

● Contents
  ○ Disparities in health outcomes are likely to depend on demographic status
  ○ When the vulnerable group (the old and young) is exposed over a long period, road proximity causes additional health degradation

● R & nlrx
  ○ Workload: 60% on NetLogo, 30% on R, 10% HPC
  ○ nlrx works as a compiler
  ○ nlrx results in a tibble format that can be plotted in a variety of figures for different purposes with ggplot, ganimate
Contributions from nlrx

● **Time saving**
  ○ Fast iterative process
  ○ The simulation ends with only a paragraph of codes

● **Solves fat finger issues**
  ○ No need to delete readme text from the NetLogo output

● **Convenience**
  ○ Doesn’t need *rJava* installation *(Really helps when running HPC)*
  ○ Silent machines
  ○ Code categorical variables
An Agent-Based Assessment of Health Vulnerability to Long-Term Particulate Exposure in Seoul Districts

Hyesop Shin¹ and Mike Bithell¹

¹Department of Geography, University of Cambridge, Downing Place CB2 3EN, United Kingdom

Correspondence should be addressed to hs627@cam.ac.uk

Journal of Artificial Societies and Social Simulation 22(1), 2, 2019
DOI: 10.18564/jasss.3940  URL: http://jasss.soc.surrey.ac.uk/22/1/2.html

Received: 22-01-2019  Accepted: 11-01-2019  Published: 31-01-2019

For more information...

Paper

Tutorial https://tiny.cc/nlrx
Thank you!

@hyesop
hs621@cam.ac.uk
@mrsensible
Agent-based modelling: **advantages**

- Is able to simulate human-environment interaction
- Can analyse adverse health impact by social groups
- Follows trajectories of individuals over time
- Envisages effects from possible scenarios ("What if...?")
- NetLogo is the most widely used software in the ABM world
Air Pollution in South Korea (March, 2019)

- Songdo, Incheon (Mar.5th, 2019) : $\text{PM}_{2.5} > 200$
- Central Seoul (Mar.6th, 2019) : $\text{PM}_{2.5} > 140$
Urbanisation and Air Pollution

$\text{NO}_2$ trend in a decade (2005-2015), Satellite imagery taken by NASA, 2015
Urbanisation and Air Pollution

NO$_2$ change in East Asian countries

Traffic in Seoul CBD
Health threat may differ by demographic factors

- **Children** are inherently more susceptible to air pollution as their lung function and immunological systems are still developing (Pearce et al., 2006)
- Higher risks due to the PM$_{10}$ exposure were observed for **elderly individuals** - COPD, stroke, etc (Halonen et al., 2016; Wang et al., 2016)

More attention should be given to how travel behaviours differ by social groups (e.g. age), and how health loss are manifested in each group after a long-term pollution exposure
Stage 1: Install and finish coding in NetLogo

- Install NetLogo >= 5.3, NetLogo 6 is preferred
- Java required
Agent-based modelling: advantages

- Is able to simulate human-environment interaction
- Can analyse adverse health impact by social groups
- Follows trajectories of individuals over time
- Envisages effects from possible scenarios ("What if...?")
- **NetLogo** is the most widely used software in the ABM world