Adaptation Strategies

EES 4760/5760

Agent-Based and Individual-Based Computational Modeling

Jonathan Gilligan

Class #13: Wednesday, October 02 2024

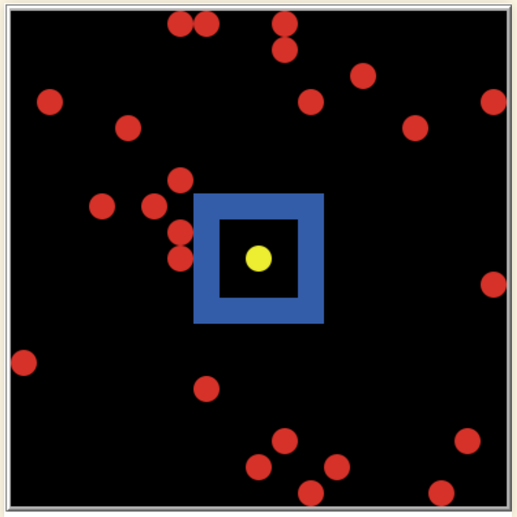
Getting Started

- Sit with your team partners
- Download model:
  - https://ees4760.jonathangilligan.org/models/class_12/business-investor.nlogo

Subsetting

Subsetting

- Open the BusinessInvestor model in NetLogo
- Click `setup`
- Turn all the turtles red and the patches black
- Turn turtle 5 yellow and move it to `patch 0 0`
- Ask turtle 5 to identify all the patches that are exactly 2 patches away from the turtle's patch (not a 2-patch radius from turtle-2)



Hints:

- There are many ways to do this. Let's look at a way to do this with the `neighbors` primitive.
- Hints:
  - Use `member?` primitive (`member <agent> <agent-set>`)
  - Use `patch-set` primitive to turn an list of many patch-sets into a single patch-set
- Suggestion:
  1. Start by turning all neighbor patches (patches exactly 1 patch away) blue
  2. Next turn all patches within 2 patches blue
  3. Now turn all patches black again
  4. Now turn all patches within a 2-patch distance blue *except* the turtle's patch
  5. Now turn all patches black again
  6. Now turn all patches within a 2-patch distance blue *except* the turtle's patch and the patches 1 patch away.

A solution

```
ask turtle 5 [
  ask (patch-set [neighbors] of [neighbors] of self) with
    [not member? self [(patch-set neighbors patch-here)] of myself]
  [
  set pcolor blue
  ]
]
```

- What does `self` refer to in `patch-set [neighbors] of [neighbors] of self`?
  - `self` refers to `turtle 5`
  - `ask turtle 5 [ ... ]` puts the `[...]` in the context of `turtle 5`, so `self` refers to `turtle 5`
- What does `self` refer to in `not member? self [(patch-set neighbors patch-here)] of myself`?
  - `self` refers to the various patches in the `patch-set`: `(patch-set [neighbors] of [neighbors] of self)`
  - `x with [...]`, where `x` is an `agent-set` evaluates `[...]` for each of the agents (patches, turtles, links) in `x`, so `self` in the `[...]` refers, in turn, to each patch in the `patch-set`
- What does `myself` refer to in `not member? self [(patch-set neighbors patch-here)] of myself`?
  - `myself` refers to `turtle 5`
  - `myself` refers to the agent doing the asking

Self vs. Myself

```
to test-self-myself
  ask turtle 5
  [
    ask turtle 7
    [
      print (word "first self = " self)
      print (word "first myself = " myself)
      ask turtle 2
      [
        print (word "second self = " self)
        print (word "second myself = "
myself)
      ]
    ]
  ]
end
```

```
observer> test-self-myself

first self = (turtle 7)
first myself = (turtle 5)
second self = (turtle 2)
second myself = (turtle 7)
```

Telemarketer Model

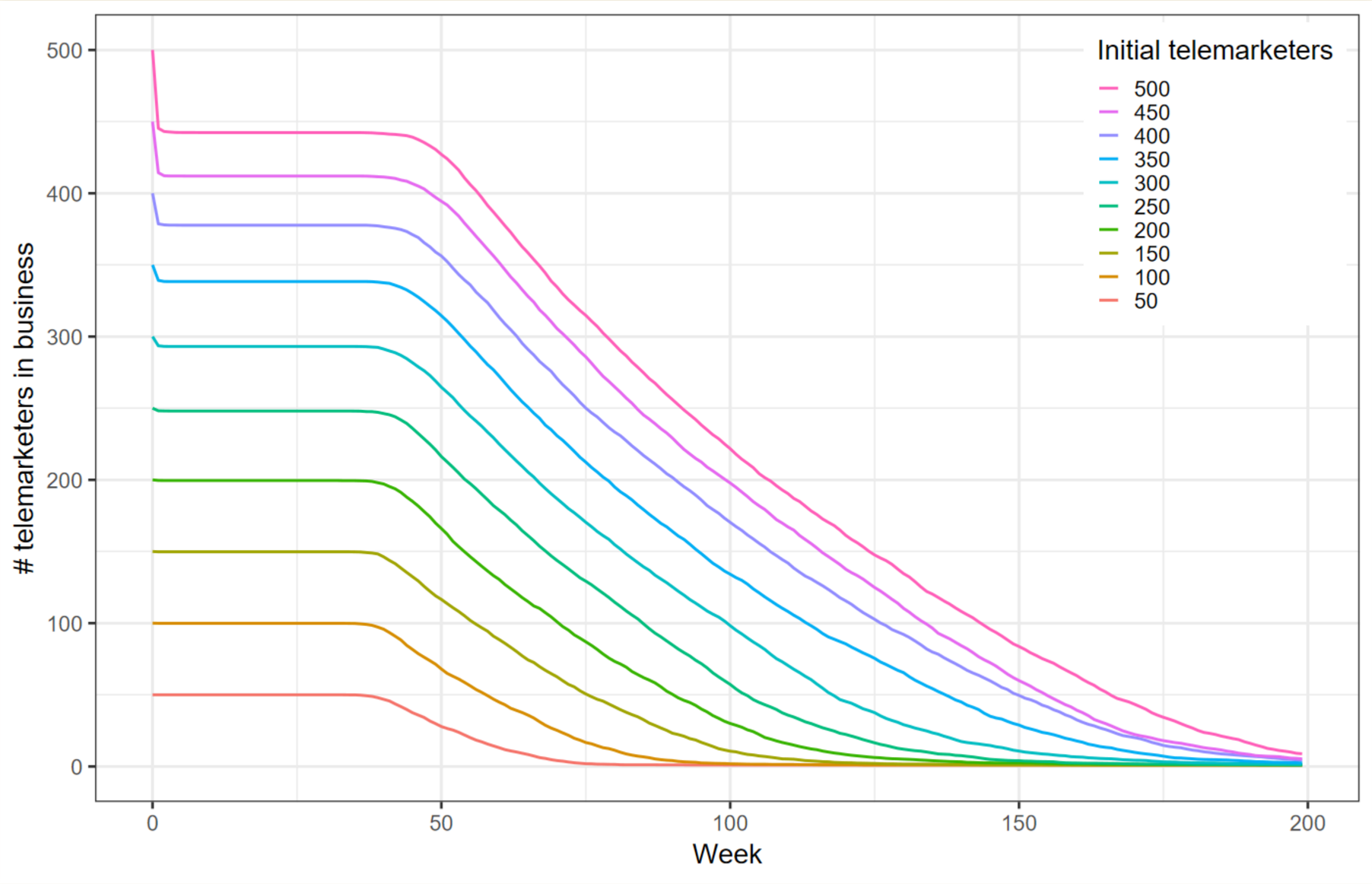
- `self` refers to the agent *being asked*.
- `myself` refers to the agent *doing the asking*.
- First: `turtle 5` is asking `turtle 7` to do something.
  - `self` is `turtle 7`, `myself` is `turtle 5`
- Second: `turtle 7` is asking `turtle 2` to do something.
  - `self` is `turtle 2`, `myself` is `turtle 7`

Telemarketer Model

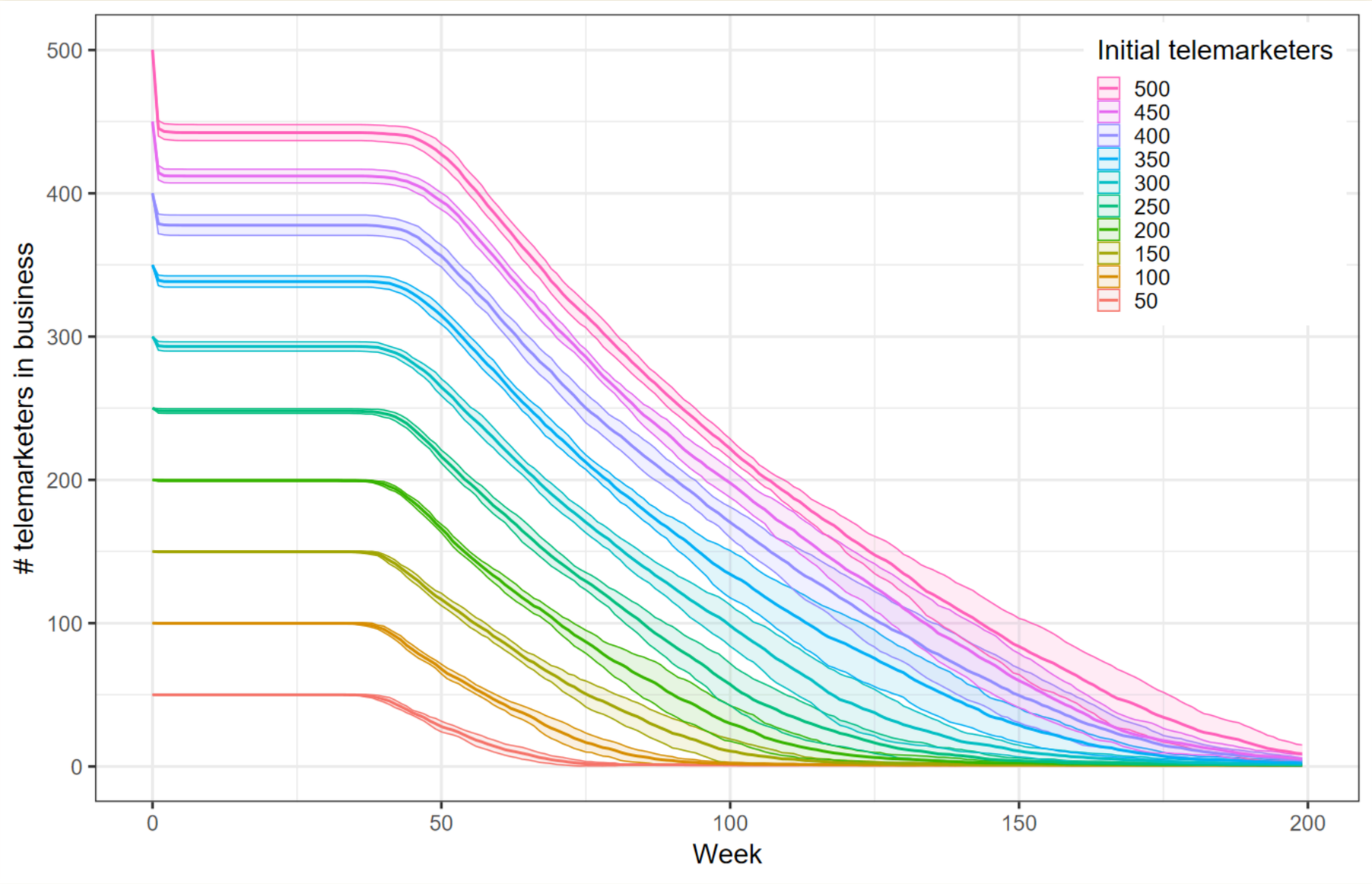
- Telemarketing firms interact
  - Telemarketer calls patches
  - If patch has received a previous call that tick, it hangs up
  - If patch has not received a previous call that tick, it buys something
  - Interaction is indirect, mediated by patches
- Accounting:
  - Net profit = 2 × sales - 50 × size
  - If balance < 0, firm goes bankrupt
- Growth
  - If balance > growth threshold, firm increases size proportional to excess balance

Results

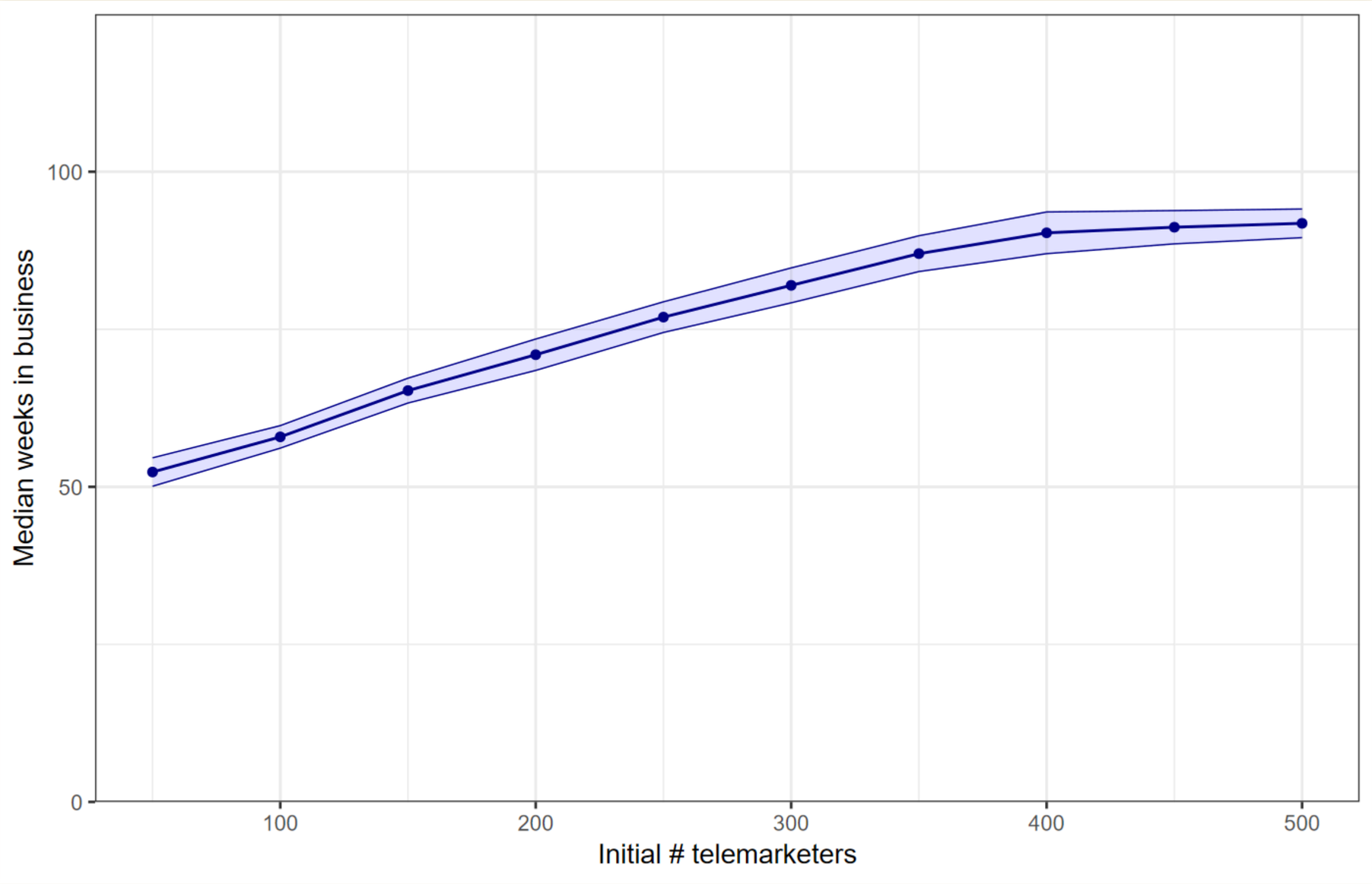
Results


Initial telemarketers
500
450
400
350
300
250
200
150
100
50
telemarketers in business
500
400
300
200
100
0
0
50
100
150
200
Week

Variation

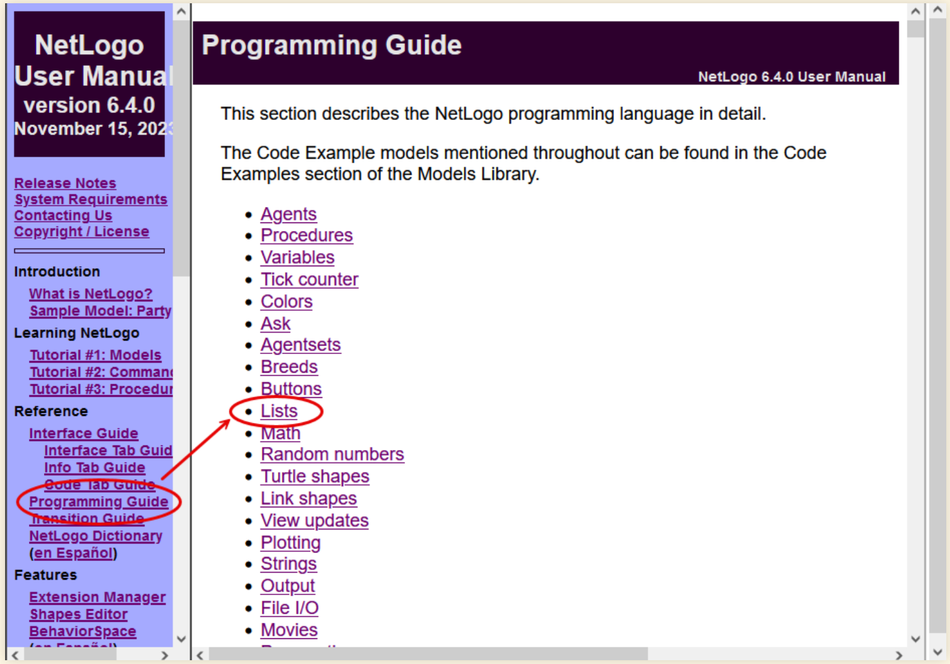

Initial telemarketers
500
450
400
350
300
250
200
150
100
50
telemarketers in business
0
100
200
300
400
500
0
50
100
150
200
Week

Median Weeks in Business



How to Calculate Median Weeks in Business?

- There are 50–500 telemarketers (turtles)
- What happens when a telemarketer goes out of business?
- What do you need, to calculate a median?
- We need a list of how long each turtle survives
- Use NetLogo `list` primitive in a global variable
- How do you figure out the age of a turtle in this model?
- How might we update the list? (Hint: consult the NetLogo dictionary)
  - You may want to use the `replace-item` primitive
  - See the “Lists” section in the NetLogo Programming Guide in the User Manual for more guidance on how to update lists.

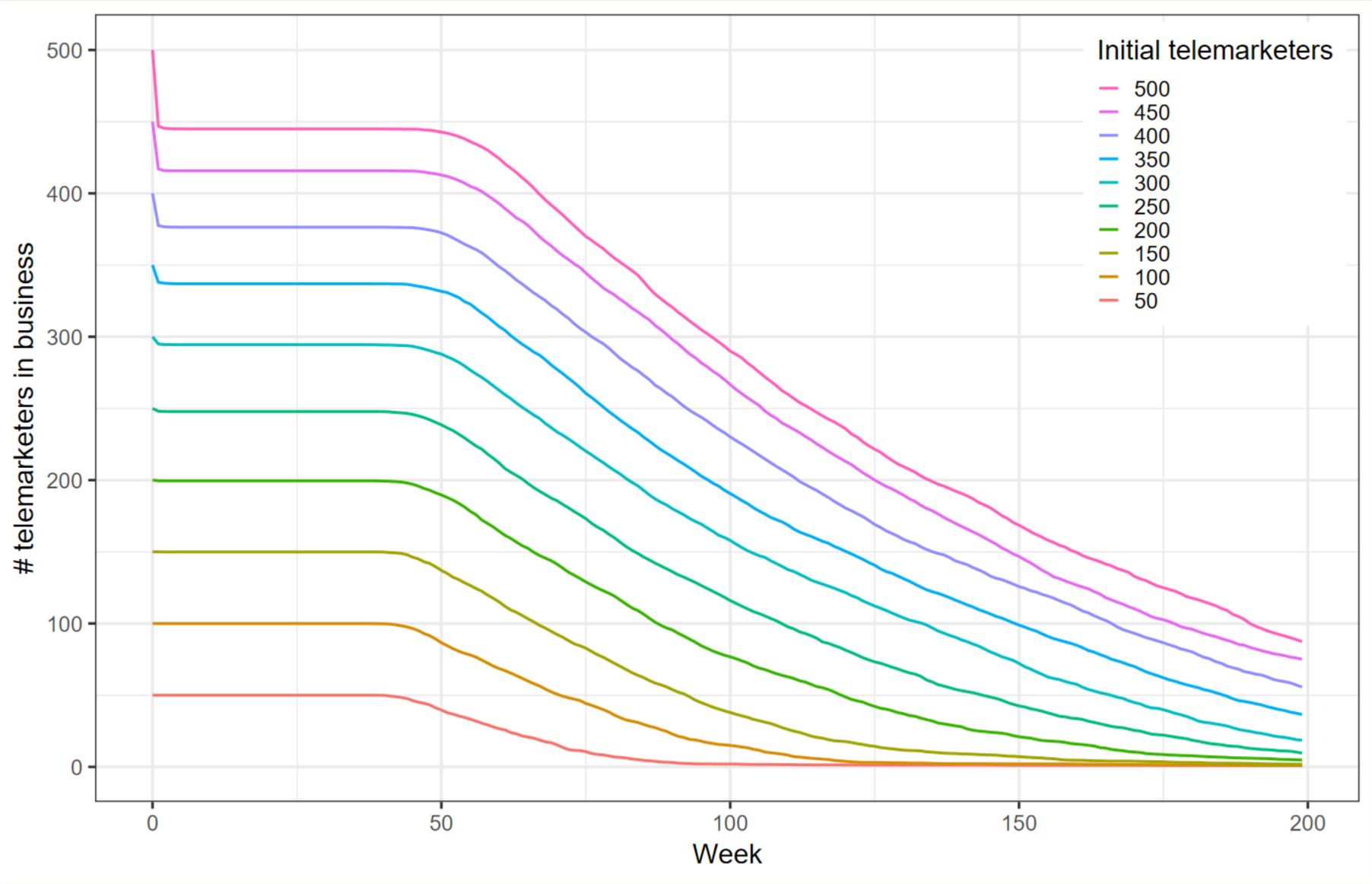


Mergers

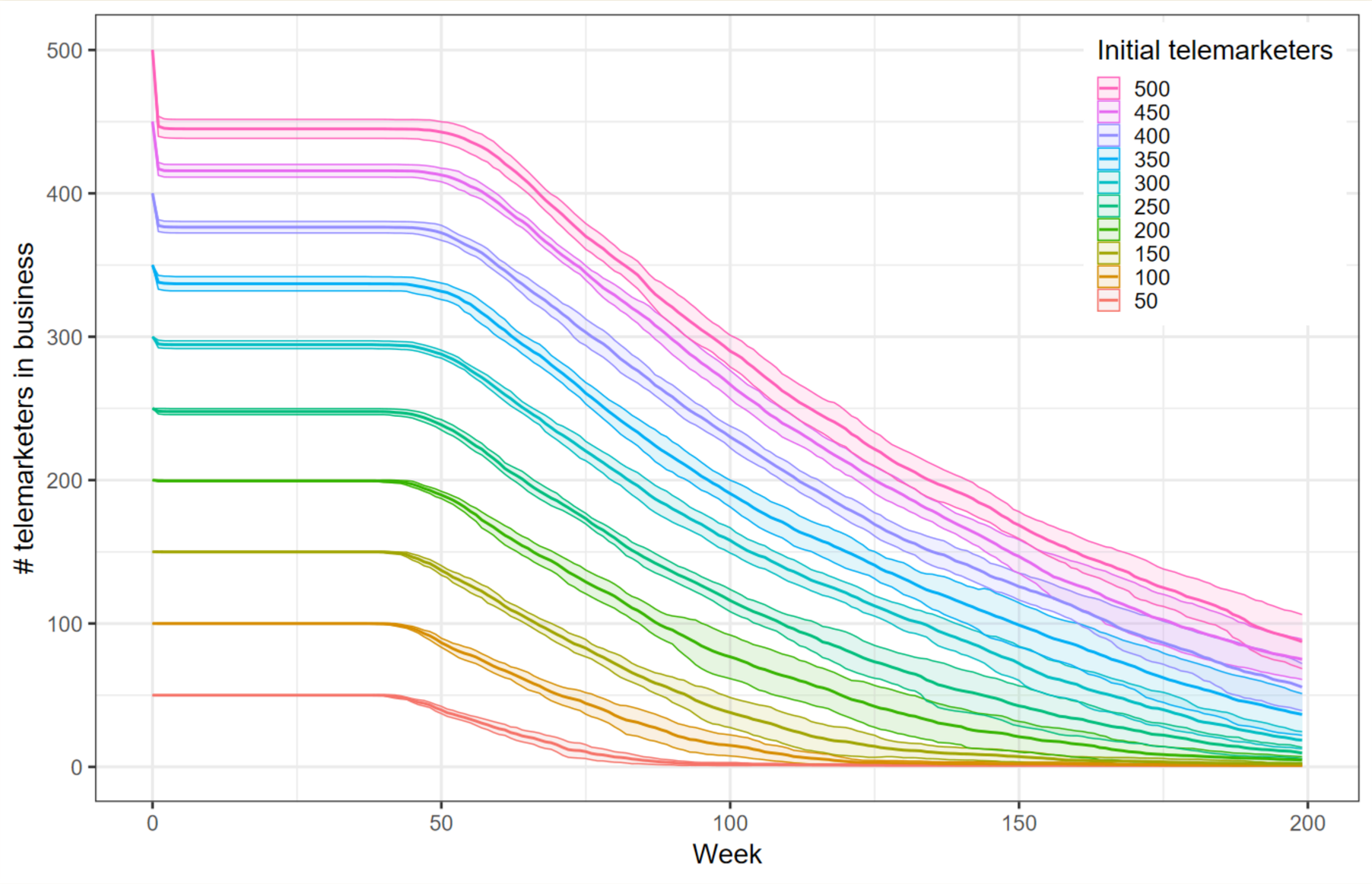
Mergers

- Instead of going bankrupt when the bank balance drops below 0, firms look for acquisition partner
  - Find a company that’s bigger and has enough money to pay off deficit.
  - If it finds a parent, parent pays off deficit (child firm ends up with 0 balance)
  - In future turns, child pays parent 50% of its net profits.
  - In future, if child’s balance becomes negative:
    - If parent has enough money, it pays child’s deficit
    - If parent does not have enough money, child dies.

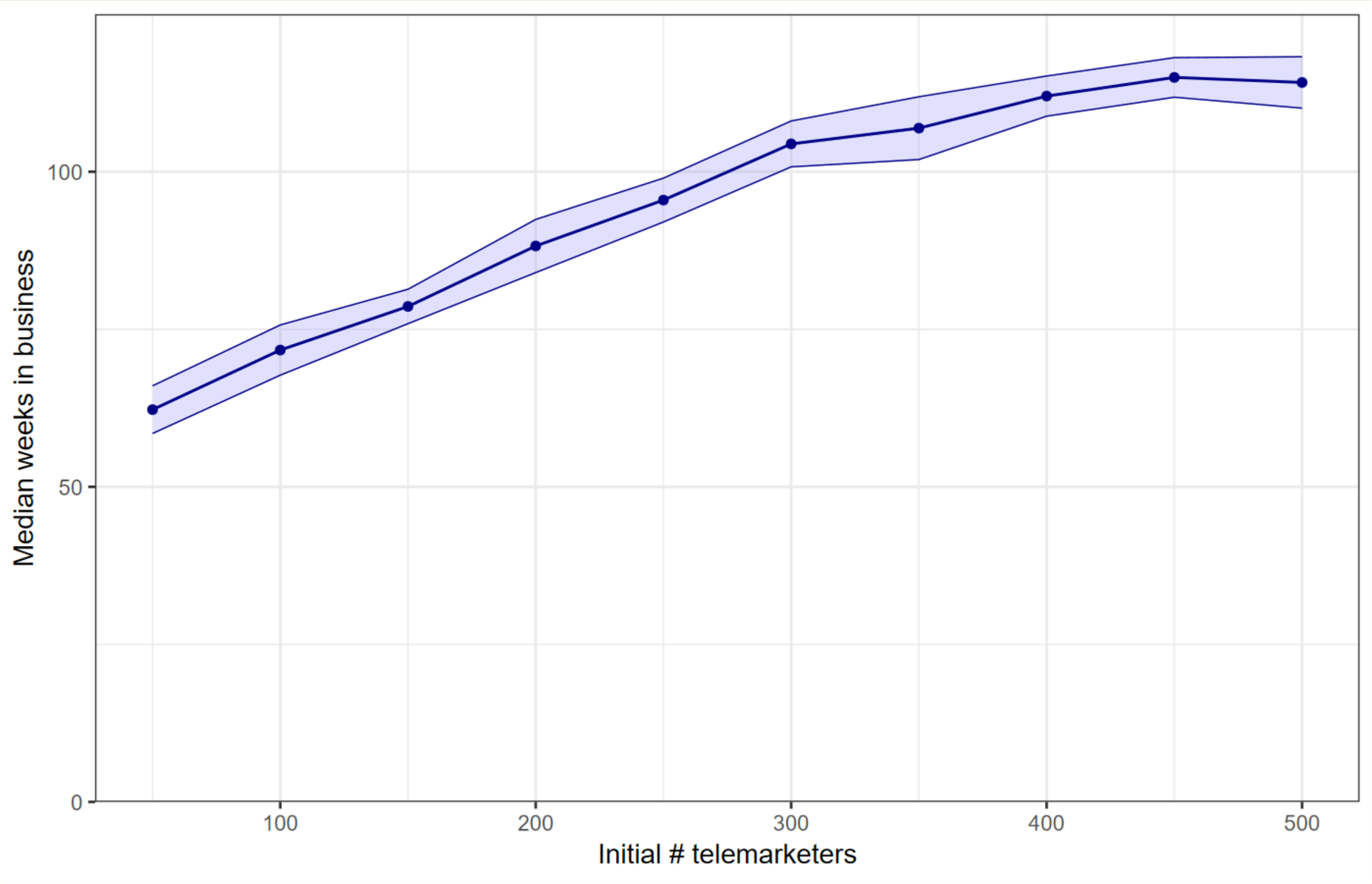
Results


Initial telemarketers
500
450
400
350
300
250
200
150
100
50
telemarketers in business
0
100
200
300
400
500
0
50
100
150
200
Week

Variation


Initial telemarketers
500
450
400
350
300
250
200
150
100
50
telemarketers in business
0
100
200
300
400
500
0
50
100
150
200
Week

Median Weeks in Business



Work With Partners on Team Projects