# **Group3 Project 4 workflow**

# 1. Introduction

In this project, we are going to continue the accessibility analysis of Healthcare facilities for elderly people in San Francisco. Instead of using the 2SFAC model, we are going to use agent-based modeling (ABMs) to calculate accessibility scores in each census block by NetLogo.

## 2. Model and Data Selection

### 2.1 Model

In the healthcare-facility-accessibility model of ABMs, we design agents for potential patients, healthcare facilities. Each potential patient will go to a randomly selected healthcare facility in the given radius of areas at their different departure time. If the randomly selected healthcare facility is full of its capacity, we will assume this patient doesn’t receive healthcare services, if not, then we will regard that the patient receives healthcare services.mark them Each healthcare facility will randomly create physicians based on its capacity. Used to set every clinic with capacity of 8, but if we have the distribution, we will randomly create the physician number it maybe 6 or 9 every 2 hours. If the healthcare facilities still have capacity for patients, they will receive patients until the randomly scheduled time is up. By this process, we want to measure the percentage of patients who are received as the accessibility scores for each census block and draw a healthcare facility accessibility map accordingly.

To simplify the process, we will rewrite the existing project’s code and modify it. The food desert project[1] uses a spatially explicit agent based model to model the population of Washington DC, their accessibility to food, and their health status. (Figure 1)

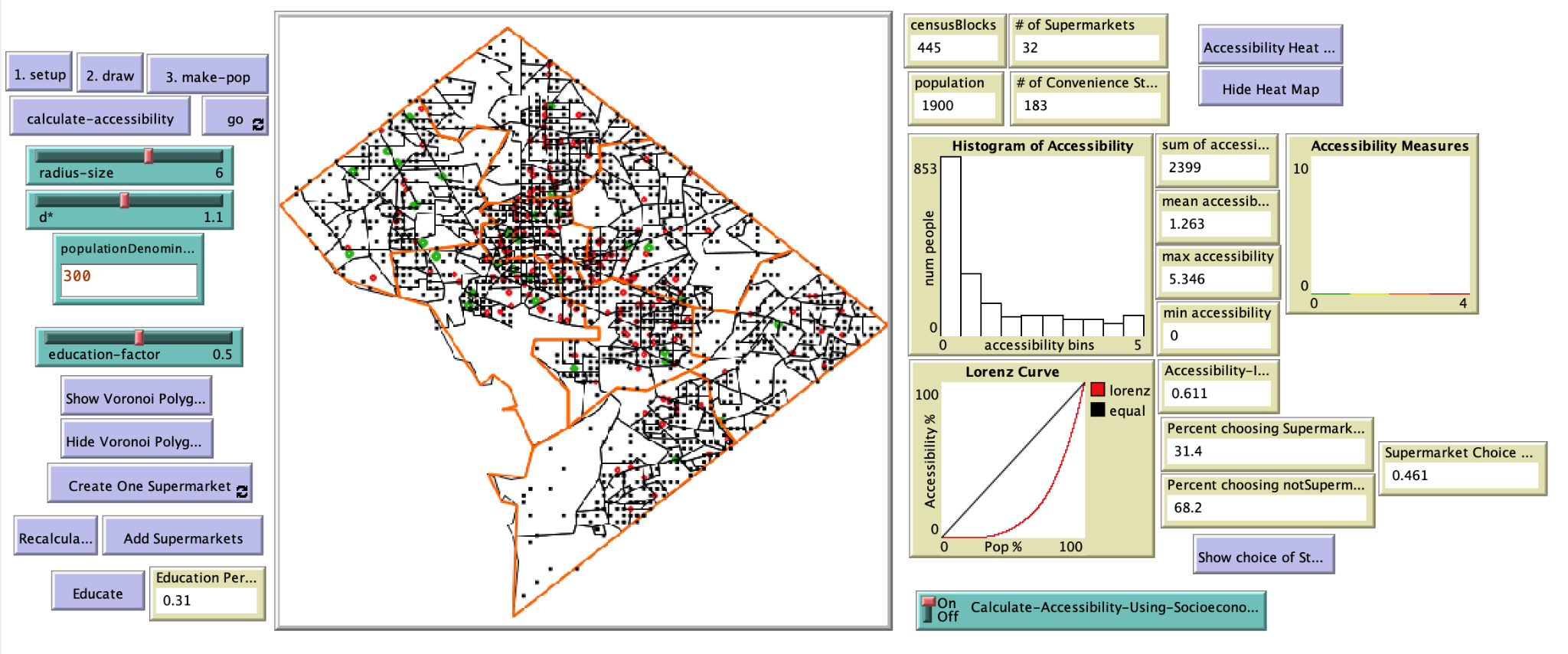


Figure 1. Food Desert NetLogo project

### 2.1 Workflow

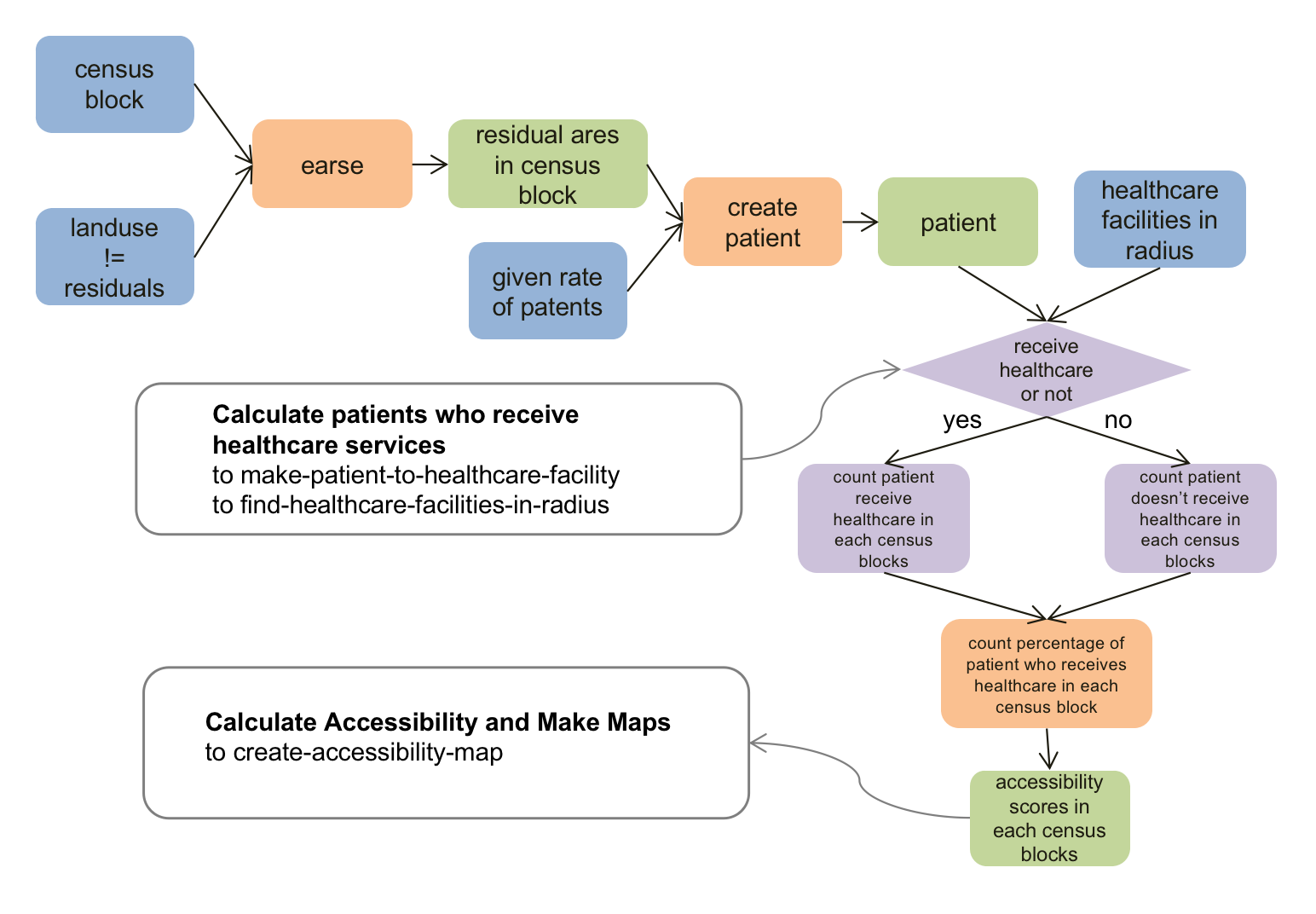


Figure 2. Workflow of our project

### 2.3 Data

The original data in the model:

| dataset | description |
| --- | --- |
| DC supermarkets | Point feature of supermarkets and other grocery stores in DC |
| DC census block data | Census block boundary with population, education, food stamp, and poverty data |
| DC ward and health data | Health data in DS, including obesity, heart disease, and overall health indicator |

The data we use to replace for our project:

| dataset | source | description |
| --- | --- | --- |
| SF Health Facilities | DataSF | Point feature of health facilities in SF(with type senior health or others) |
| SF Census Blocks | DataSF | Boundary and senior population |
| SF Economy Data | ArcGIS Online | Food stamp, Poverty, Health Insurance, Education to enrich the census blocks data |
| Environmental Justice Index(EJI)[2] | CDC | Health Vulnerability Indicators: High estimated prevalence of cancer, high blood pressure and diabetes |
| SF Road[3] | DataSF | Line feature of roads in SF with speed limit |
| SF Landuse[4] | DataSF | Polygons of landuse |

## 3. Data Preparation

### 3.1 Project

According to the NetLogo GIS Extension, NetLogo only supports limited kinds of coordinate systems and projections. To fit the coordinate system to NetLogo, we projected the datasets into NAD83.

### 3.2 Join and enrich

To merge all the data into one layer, I use the Join and Enrich tool in ArcGIS Pro. The eji data I collected from the CDC’s website was joined to the Census Blocks data on the GEO\_ID field. I also use the Enrich tool to join the economy data I found in Esri’s data portal to the census blocks.

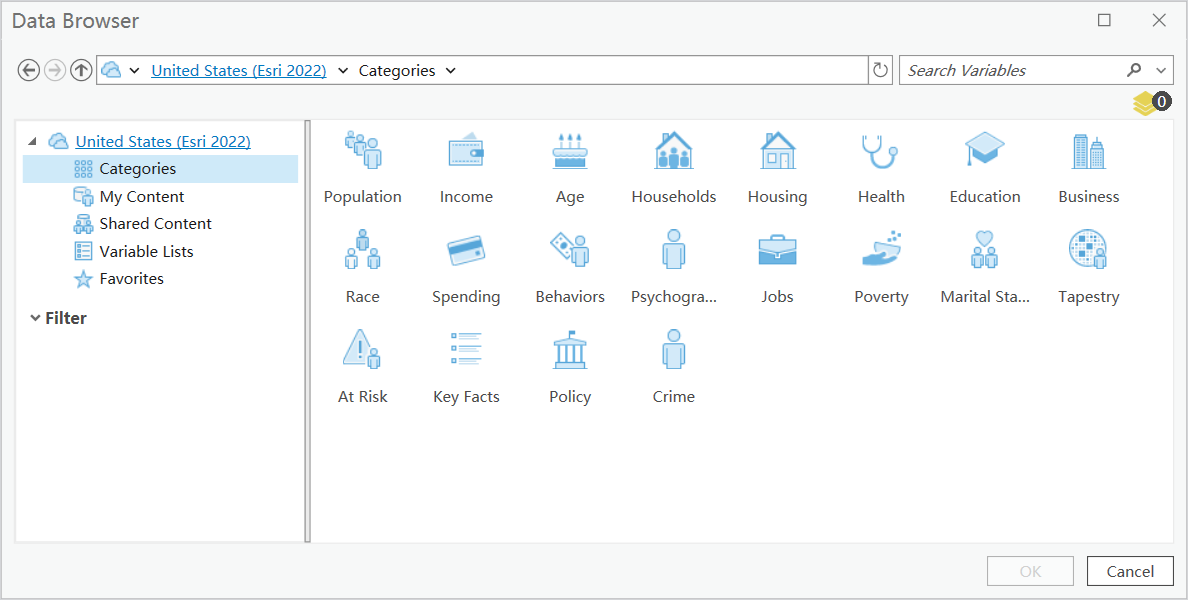


Figure 3. Data enrich

## 4. NetLogo Coding

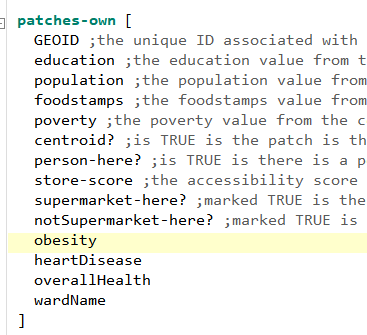
### 4.1 Set Up GIS data

Change variable names：

First, we edit the variable names and add new variables to make it fit the topic in our project. For example, we changed the supermarket into seniorClinics.

| **Original code:** | **New code:** |
| --- | --- |

Edit patches-own:



patches-own [

GEOID ;the unique ID associated with each census block

;data for further research:

education ;the education percentage from the census dataset

population ;the senior population from the census dataset

poverty ;the poverty percentage from the census dataset

disability ;the 1+ persons with disability percentage of household from the census dataset

healthInsurance ;the percentage of senior people with 1 type of health insurance

patient-here? ;is TRUE is there is a person on the patch...used in the creation of the population of people

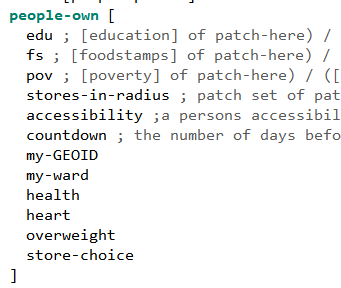
seniorClinic-here? ;marked TRUE is there is a senior clinic on the patch

cancerValue

highBloodPressure

diabetesValue ]

1. people-own



breed [patients patient]

patients-own [

edu ;

fs ;

pov ;

healthcare-facilities-in-radius

patient-A? ;a person whether receive healthcare services or not

my-GEOID

cancer

bpHigh

diabetes

patient-healthcare-facility

]

### 4.2 to setup

to setup ;load the gis data

clear-all

reset-ticks

gis:load-coordinate-system (word "Data/SFCensusBlocks.prj")

set seniorClinics gis:load-dataset "Data/SFSeniorHealth.shp" ;;;this is the clinic location data

set censusBlocks gis:load-dataset "Data/SFCensusBlocks.shp"

set notSeniorClinics gis:load-dataset "Data/DCNOTSupermarkets.shp"

end

### 4.3 to draw

to draw ; draw the map and apply the vector data to the raster in netlogo for the socioeconomic data

clear-drawing

reset-ticks

; gis:set-world-envelope gis:envelope-of censusBlocks

gis:set-world-envelope (gis:envelope-union-of ;(gis:envelope-of sites)

(gis:envelope-of notSeniorClinics)

(gis:envelope-of seniorClinics)

(gis:envelope-of censusBlocks)

)

ask patches [set pcolor white]

gis:apply-coverage censusBlocks "GEO-ID" GEOID

gis:apply-coverage censusBlocks "SeniorPOP" population

gis:apply-coverage censusBlocks "edu" education

gis:apply-coverage censusBlocks "pov" poverty

gis:apply-coverage censusBlocks "disability" disability

gis:apply-coverage censusBlocks "health\_insurance" healthInsurance

gis:apply-coverage censusBlocks "cancer" cancerValue

gis:apply-coverage censusBlocks "bpHigh" highBloodPressure

gis:apply-coverage censusBlocks "diabetes" diabetesValue

gis:set-drawing-color green

gis:draw seniorClinics 3

; mark-seniorClinics

gis:set-drawing-color black

gis:draw censusBlocks 1

end

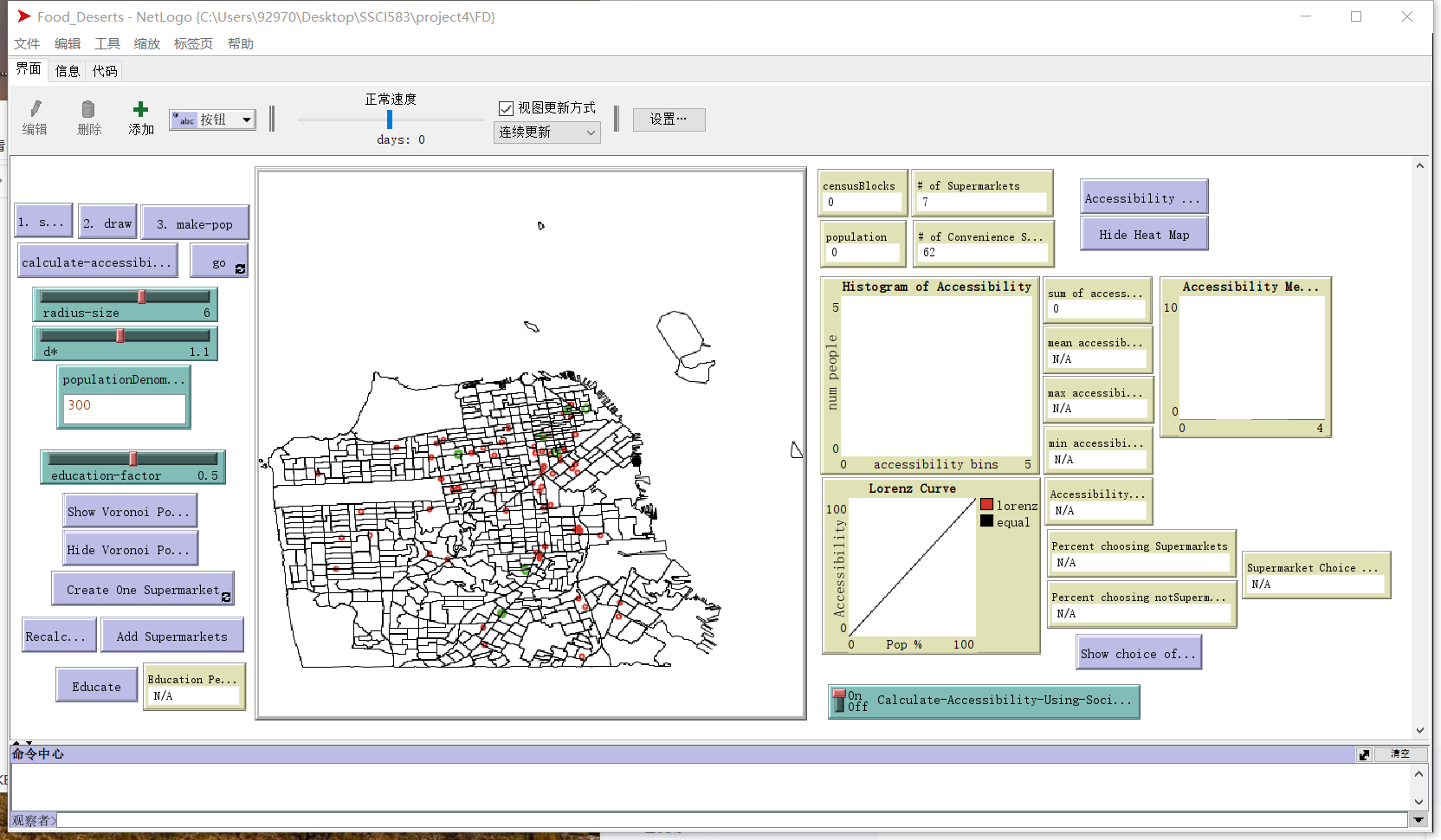


Figure 4. GIS set up

### 4.4 Set up agents (pseudocode)

In this part, we code to create agents in our own way.

to make-patient

Randomly create patients for census block based on the elderly population number

Create patient-arrive-time for each patient based on normal distribution of appointment time

Create patient-duration for each patient based on normal distribution of appointment duration

end

to set-healthcare-facility-capacity

Randomly create physicians representing the capacity for each current tick

end

### 4.5 Calculate patients who receive healthcare services

to make-patient-to-healthcare-facility

;Randomly create patients based on the potential percentage of elderly people who intend for healthcare services.

##codes##

ask patient[

;in the radius of ‘distance-tolerance’ miles

find-healthcare-facilities-in-radius

ifelse any? patient-healthcare-facilities-in-radius

[

;randomly select one healthcare facility

let healthcare-facility-x one-of patient-healthcare-facilities-in-radius

set patient-healthcare-facility healthcare-facility-x

; Check if the demand is less than the capacity at the selected facility

ask healthcare-facility with [who = healthcare-facility-x][

ifelse healthcare-facility-demand < healthcare-facility-capacity[

set [patient-A?] of myself True

set healthcare-facility-demand healthcare-facility-demand+1

set healthcare-facility-patient-list lput [patient-duration] of myself healthcare-facility-patient-list

][set [patient-A?] of myself False]

]

][set patient-A? False]

]

end

to find-healthcare-facilities-in-radius ;called by make-patient-to-healthcare-facility

;need more researches

end

4.6 Calculate Accessibility and Make Maps

to create-accessibility-map

let color-ramp gis:make-color-ramp (list 0.0 blue 1.0 red)

ask censusBlocks[

;count the number of True of patient-A in this censusBlock

let count-patient-A-True count patients with [patient-census-block = [censusBlocks] of myself and patient-A? = True]

let count-patients count patients with [patient-census-block = [censusBlocks] of myself]

set censusBlocks-accessibility count-patient-A-True/count-patients

; set the drawing color for each census block based on its accessibility value

let color gis:set-drawing-color color-ramp (map [cb -> [censusBlocks-accessibility] of cb] self)

gis:fill-polygon gis:centroid self

]

end

to go

make-patient-to-healthcare-facility

ask healthcare-facility[

foreach healthcare-facility-patient-list[

pd ->

set pd pd-1

if pd = 0[

set healthcare-facility-patient-list remove pd healthcare-facility-patient-list

]

]

set healthcare-facility-demand healthcare-facility-demand - 1

]

create-accessibility-map

end

## 5. Potential Results

Since we haven’t fully debugged the code, we will show the accessibility map of the food desert project here, which is very similar to the potential result of our project. (Figure 4)

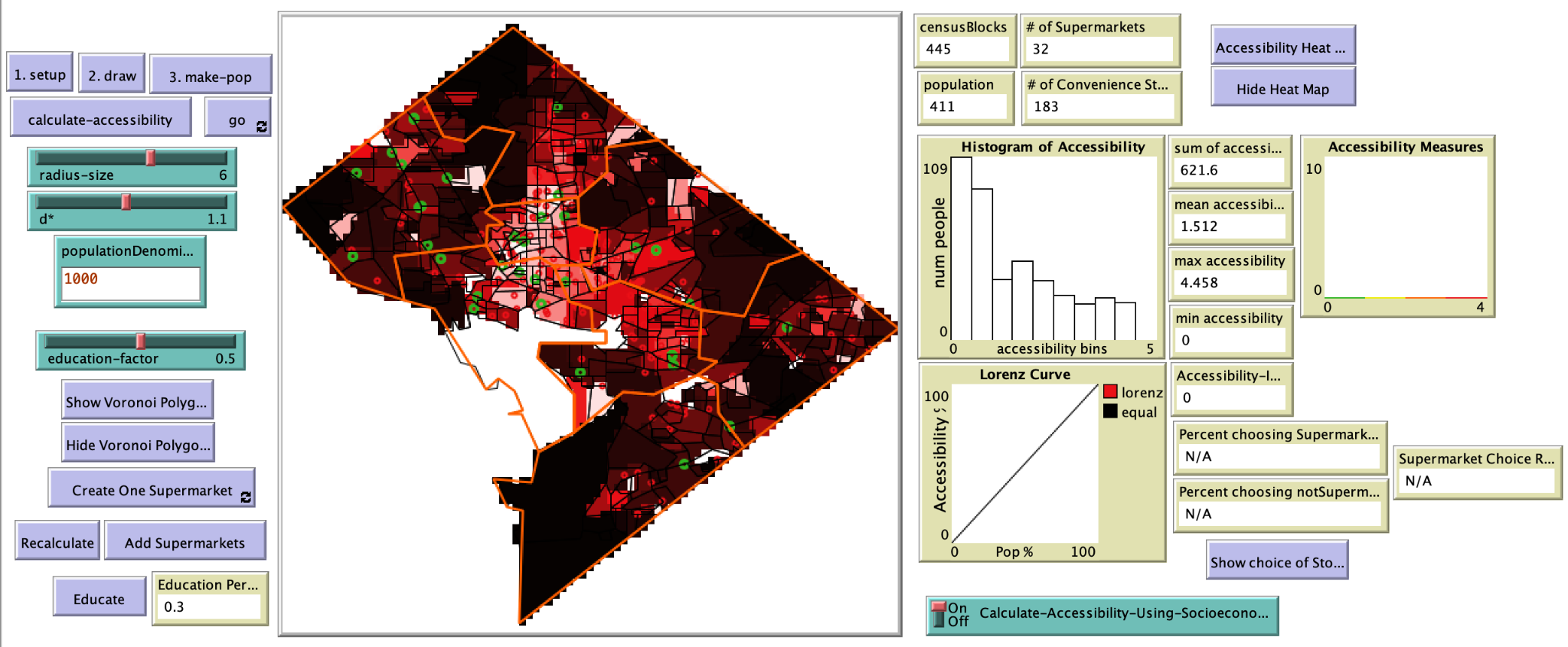


Figure 4. Result of Food Desert Project

## 6. Discussion

We used to consider the catchment area as a drive-distance in 10 minutes, however, it’s difficult to do this in NetLogo, maybe in the report we can realize the traffic part.

What we changed from the Food Desert project is the way to calculate accessibility, which means we will design our own way instead of using the existing codes.

Reference

Data

[1]<https://github.com/acrooks2/ClassModels/blob/master/CSS645Models/Food_Desert/README.md>

[2]<https://www.atsdr.cdc.gov/placeandhealth/eji/index.html#:~:text=The%20Environmental%20Justice%20Index%20uses,health%20for%20every%20census%20tract>.

[3]<https://data.sfgov.org/Transportation/Speed-Limits-per-Street-Segment/3t7b-gebn>]

[4]<https://data.sfgov.org/Housing-and-Buildings/Land-Use/us3s-fp9q>]