Toy example to experiment with spatial availability

Introduction

So Anastasia and I have been experimenting with proportional allocation in accessibility measures, and have discussed the concept of *spatial availability*. Briefly, consider the following accessibility measure:

$$A_i = \sum_{j=1}^{J} W_j f(c_{ij})$$

where W_j is the number of opportunities at location j, c_{ij} is a measure of the cost of moving between i and j, and $f(\cdot)$ is an impedance (or distance-decay) function (a monotonically non-increasing or decreasing function of c_{ij}). In this way, the accessibility A_i is the weighted sum of opportunities that can be reached from location i. It is well known that this is a gravity-type measure of accessibility, and it is a widely used tool in transportation studies.

Accessibility A_i, by summing the opportunities in the neighborhood of i (the neighborhood is defined by the impedance function), is an estimation of the total number of opportunities that can be reached from i at a certain cost.

As we have discussed proportional allocations (see for instance Paez, Higgins, Vivona, 2019), it has become increasingly clear that a lot of multiple-counting happens when calculating A_i for $i = 1, \cdots, n$, since every opportunity that can be reached from every i enters the sum.

For this reason, we have thought of a related concept that we are provisionally calling *spatial availability*, and that we think of as a singly-constrained measure of accessibility.

How does this work? It all comes down to proportional allocation. We wish to allocate opportunities proportionally, based on the number of size of, and the distance to, the demand for opportunities.

This framework is appropriate for non-divisible opportunities. Examples of non-divisible opportunities are jobs (when a person takes up a job, the same job is no longer available to anyone else), seats at schools (once a seat has been taken, it is no longer available to another student). From a different perspective, employers may see workers as opportunities, but when a worker takes a job, they are no longer in the available pool of candidates.

We distinguish between opportunities and demand.

To illustrate our analytical framework, we will think of demand as 'population' and opportunities as 'jobs'.

We begin with allocation based on demand (the number of individuals in the population in the labor market). Consider an employment center j with W_j^r jobs of type r. There are also K population centers in the region. To allocate the jobs proportionally based on the population of each center we define the following proportional allocation factor:

$$f_{ij}^p = \frac{P_{i\in r}^{\alpha}}{\sum_{k=1}^{K} P_{k\in r}^{\alpha}}$$

where f_{ij}^p is a factor to proportionally allocate a share of the jobs at j to population center i. On the right hand side of the equation, $P_{k\in r}$ is the population at location k eligible for taking jobs of type r (maybe those with a certain level of training, or in a designated age group). Here we add a parameter α that can be used to modulate the effect of size in the calculations. The summation in the bottom is over $k = 1, \cdots, K$, the number of population centers in the region. The factors f_{kj}^p satisfy the property that $\sum_k^K f_{kj}^p = 1$.

The share of jobs allocated to (i.e., available for) each population center is:

$$V_{kj} = W_j f^p_{kj}$$

and since $\sum_k^K f^p_{kj} = 1$ it follows that:

$$\sum_{k=1}^{K} V_{kj} = W_j$$

In other words, the number of jobs is preserved.

As an example, consider an employment center j in a region with two population centers (say j and k). For simplicity, assume that all jobs in the population centers are eligible for the population in this region. The allocation factors for the jobs at j would be:

$$f^p_{ij} = \frac{P_i^\alpha}{P_i^\alpha + P_k^\alpha}$$
$$f^p_{kj} = \frac{P_k^\alpha}{P_i^\alpha + P_k^\alpha}$$

Suppose that there are three hundred jobs in the employment center ($W_j = 300$), and that the populations are $P_j = 240$ and $P_k = 120$. The jobs would be allocated as follows (assuming that $\alpha = 1$):

$$V_{ij} = W_j \frac{P_i^\alpha}{P_i^\alpha + P_k^\alpha} = 300 \frac{240}{240+120} = 300 \frac{240}{360} = 200$$
$$V_{kj} = W_j \frac{P_k^\alpha}{P_i^\alpha + P_k^\alpha} = 300 \frac{120}{240+120} = 300 \frac{120}{360} = 100$$

Proportionally more jobs are allocated to the bigger center and the total number of jobs is preserved ($\sum_{k=1}^K W_{kj} = W_j$).

The proportional allocation factors above account for the capacity of the employment centers, but they do not account for their location relative to the population centers. The proportional allocation procedure above is insensitive to how far population center i is from employment center j. To account for this effect we can define a second set of allocation factors based on distance to the employment centers. These are defined as:

$$f^d_{ij} = \frac{f(c_{ij})}{\sum_{k=1}^K f(c_{kj})}$$

where c_{ij} is the distance/(travel time, etc.) to employment center j from population center i, and $f(\cdot)$ is an impedance function (a monotonically decreasing function of c_{kj}. The idea is that proportionally more jobs are allocated to closer locations. Assume that the impedance function is:

$$f(c_{ij}) = \exp(-\beta c_{ij})$$

Continuing the example, suppose that the distance from population center i to employment center j is 0.6 km, and the distance from population center k to employment center j is 0.3 km. Being closer, we would expect more jobs to be allocated to the population of i. The jobs would be sorted as follows:

$$f^d_{ij} = \frac{\exp(-\beta D_{ij})}{\exp(-\beta D_{ij}) + \exp(-\beta D_{kj})}$$
$$f^d_{kj} = \frac{\exp(-\beta D_{kj})}{\exp(-\beta D_{ij}) + \exp(-\beta D_{kj})}$$

Numerically, the jobs allocated to each population center would be:

$$V^d_{ij} = W_j \frac{\exp(-D_{ij})}{\exp(-D_{ij}) + \exp(-D_{kj})} = 300 \frac{\exp(-0.6)}{\exp(-0.6) + \exp(-0.3)} = 3 \times 0.426 = 127.8$$
$$V^d_{kj} = W_j \frac{\exp(-D_{kj})}{\exp(-D_{ij}) + \exp(-D_{kj})} = 300 \frac{\exp(-0.3)}{\exp(-0.6) + \exp(-0.3)} = 3 \times 0.574 = 172.2$$

Proportionally more jobs are allocated to the closer population center. As before, the sum of jobs allocated to the population centers matches the total number of jobs available.

We can combine the proportional allocation factors by population and distance as follows:

$$V_{ij} = W_i \frac{f_{ij}^p \cdot f_{ij}^d}{\sum_{k=1}^{K} f_{kj}^p \cdot f_{kj}^d}$$

In the example:

$$V_{ij} = W_j \cdot \frac{f_{ij}^o \cdot f_{ij}^d}{f_{ij}^o \cdot f_{ij}^d + f_{kj}^o \cdot f_{kj}^d} = 300 \frac{\left(\frac{2}{3}\right)\left(0.426\right)}{\left(\frac{2}{3}\right)\left(0.426\right) + \left(\frac{1}{3}\right)\left(0.574\right)} = \left(300\right)\left(\frac{0.284}{0.475}\right) = 179.4$$

$$V_{kj} = W_j \cdot \frac{f_{kj}^o \cdot f_{kj}^d}{f_{ij}^o \cdot f_{ij}^d + f_{ik}^o \cdot f_{ik}^d} = 300 \frac{\left(\frac{1}{3}\right)\left(0.574\right)}{\left(\frac{2}{3}\right)\left(0.426\right) + \left(\frac{1}{3}\right)\left(0.574\right)} = \left(300\right)\left(\frac{0.191}{0.475}\right) = 120.6$$

Fewer jobs are allocated to population center i compared to the allocation by population size only, to account for the longer distance to reach those jobs. Considering distance alone allocated more jobs to the closer population center (i.e., k), but since it is smaller, it also get a smaller proportion of jobs overall. Again, the sum of jobs at employment center j that are allocated to population centers i and k simultaneously based on *population* and *distance* is preserved (i.e., $W_{ij} + W_{kj} = W_j$.

Availability is the sum of available jobs by origin:

$$V_i = \sum_{j=1}^{J} V_{ij}$$

Step-by-step Numerical Example

In this section we present a numerical example.

Load packages needed for the example:

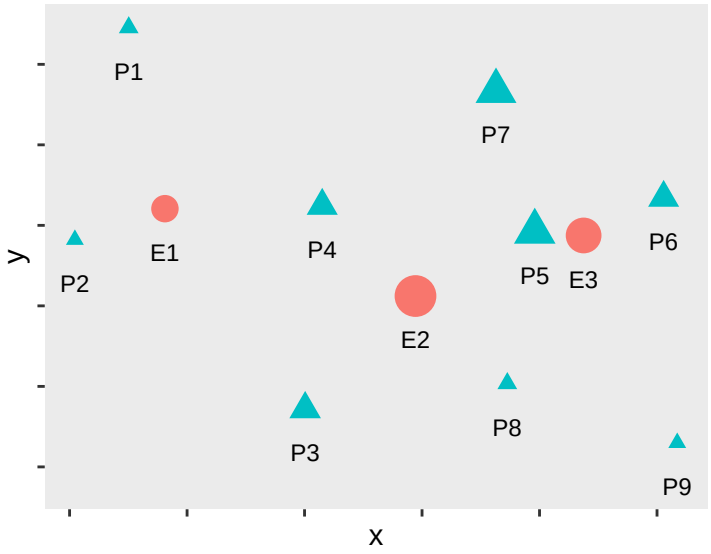
```
library(kableExtra)
library(patchwork)
library(sf)
library(tidyverse)
```

Load data for example:

```
load("od_table.rda")
load("simulated_data.rda")
load("trips.rda")
```

This is the setup for the example: three employment centers and nine population centers.

```
ggplot(data = simulated_data) +
  geom_sf(aes(color = type,
              shape = type,
              size = number)) +
  geom_sf_text(aes(label = id_short),
               size = 3,
               nudge_y = -600) +
  scale_size(range = c(2, 7)) +
  theme(axis.text = element_blank(),
        legend.position = "none",
        panel.grid = element_blank())
```



This is the distribution of jobs and population:

```
simulated_data %>%
  st_drop_geometry() %>%
  mutate(fig = "") %>%
  kable(format = "latex",
        booktabs = TRUE) %>%
  column_spec(5, image = "images/figure-1.png") %>%
  collapse_rows(columns = 5, latex_hline = "major", valign = "middle")
```

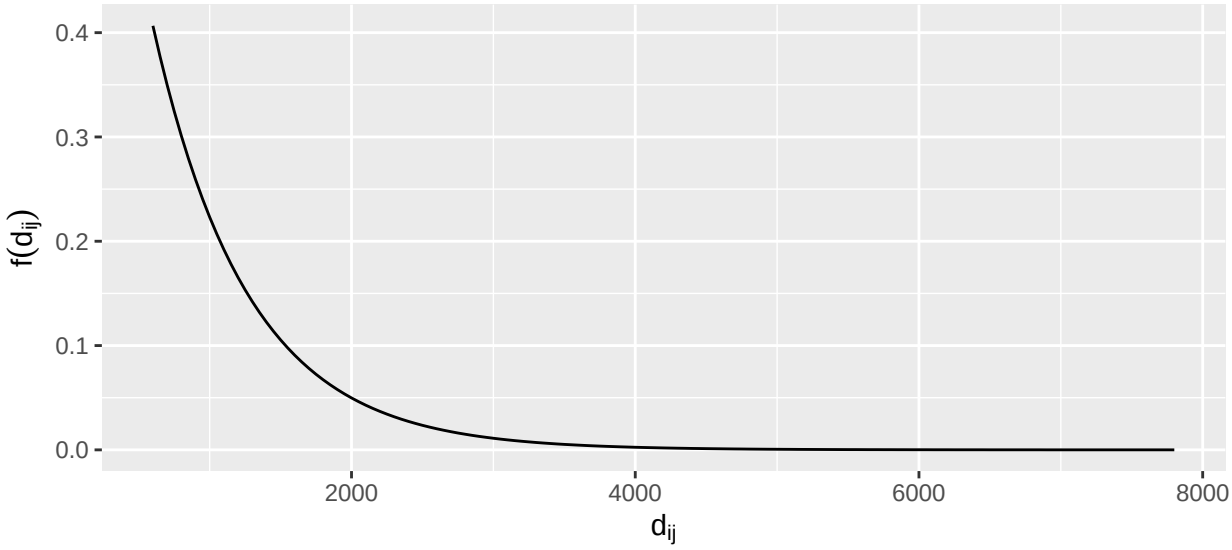
id	id_short	number	type	fig
Employment Center 1	E1	750	jobs	
Employment Center 2	E2	2250	jobs	
Employment Center 3	E3	1500	jobs	
Population 1	P1	260	population	
Population 2	P2	255	population	
Population 3	P3	510	population	
Population 4	P4	495	population	
Population 5	P5	1020	population	
Population 6	P6	490	population	
Population 7	P7	980	population	
Population 8	P8	260	population	
Population 9	P9	255	population	

Use a negative exponential function for the example:

$$f(c_{ij}) = exp(-\beta c_{ij})$$

This is the shape of the impedance function (the rate of decay is modulated by β):

```
beta <- 0.0015
data.frame(d_ij = seq(600, 7800, 50)) %>%
  mutate(f = exp(-beta * d_ij)) %>%
  ggplot() +
  geom_line(aes(x = d_ij, y = f)) +
  xlab(expression(d[ij])) +
  ylab(expression(f(d[ij])))
```



Calculate the impedance function in the OD-table:

```
od_table <- od_table %>%
  mutate(f = exp(-beta * distance))
```

For this numerical example we will assume that $\alpha = 1$ (i.e., we disregard any size effects):

```
alpha <- 1
```

First, calculate the denominator of the proportional allocation factor f_{ij}^{p} based on population, assuming that there are no catchment/eligibility conditions (i.e., $r = 1$ for all population):

```
sum_pop <- od_table %>%
  mutate(r = 1) %>%
  group_by(Destination) %>%
  summarize(sum_pop = sum(r * Population^alpha))
sum_pop
```

```
## # A tibble: 3 x 2
##   Destination         sum_pop
##   <fct>                 <dbl>
## 1 Employment Center 1    4525
## 2 Employment Center 2    4525
## 3 Employment Center 3    4525
```

Since there are no catchment/eligibility conditions for the jobs, the jobs at each employment center are in principle open to all members of the population. Next, join the sums of the population to the OD-table:

```
od_table <- od_table %>%
  left_join(sum_pop,
            by = "Destination")
```

Calculate the proportional allocation factor:

```
od_table <- od_table %>%
  mutate(f_p = Population^alpha / sum_pop)
```

Calculate the jobs allocated to each population center from each employment center:

```
od_table <- od_table %>%
  mutate(V_ij = Jobs * f_p)
```

Verify that the sum of all jobs allocated is consistent with the total number of jobs:

```
sum(od_table$V_ij)
```

```
## [1] 4500
```

The total number of jobs is preserved.

The spatial availability is the sum of available jobs by origin:

```
availability <- od_table %>%
  group_by(Origin) %>%
  summarize(availability = sum(V_ij))
availability
```

```
## # A tibble: 9 x 2
##   Origin       availability
##   <fct>               <dbl>
## 1 Population 1         259.
## 2 Population 2         254.
## 3 Population 3         507.
## 4 Population 4         492.
## 5 Population 5        1014.
## 6 Population 6         487.
## 7 Population 7         975.
## 8 Population 8         259.
## 9 Population 9         254.
```

The above calculations are based on the size of the population centers only. Now we illustrate the calculations based on location, and calculate the denominator of the proportional allocation factors based on the impedance function:

```
sum_impedance <- od_table %>%
  group_by(Destination) %>%
  summarize(sum_impedance = sum(f))
sum_impedance
```

```
## # A tibble: 3 x 2
##   Destination         sum_impedance
##   <fct>                       <dbl>
## 1 Employment Center 1         0.208
## 2 Employment Center 2         0.257
## 3 Employment Center 3         0.600
```

Employment center 1 is relatively the most distant from the population centers (the sum of the impedances gives a smaller value). Check:

```
filter(od_table, Destination == "Employment Center 1") %>% select(Origin, Destination, distance, f)
```

```
##         Origin         Destination distance            f
## 1 Population 1 Employment Center 1 2548.060 2.188201e-02
## 2 Population 2 Employment Center 1 1314.074 1.393020e-01
## 3 Population 3 Employment Center 1 3374.923 6.330450e-03
## 4 Population 4 Employment Center 1 2170.200 3.856924e-02
## 5 Population 5 Employment Center 1 5111.631 4.678109e-04
## 6 Population 6 Employment Center 1 6881.320 3.290191e-05
```

```
## 7 Population 7 Employment Center 1 4846.602 6.961791e-04
## 8 Population 8 Employment Center 1 5302.901 3.511307e-04
## 9 Population 9 Employment Center 1 7770.661 8.666962e-06
```

```
#filter(od_table, Destination == "Employment Center 1") %>% select(Origin, Destination, distance, f) %>
```

```
filter(od_table, Destination == "Employment Center 2") %>% select(Origin, Destination, distance, f)
```

```
##         Origin         Destination distance           f
## 1 Population 1 Employment Center 2 5419.120 0.0002949572
## 2 Population 2 Employment Center 2 4762.588 0.0007896807
## 3 Population 3 Employment Center 2 2179.648 0.0380265024
## 4 Population 4 Employment Center 2 1790.100 0.0682109588
## 5 Population 5 Employment Center 2 1869.645 0.0605389549
## 6 Population 6 Employment Center 2 3681.931 0.0039942599
## 7 Population 7 Employment Center 2 3037.461 0.0105019747
## 8 Population 8 Employment Center 2 1748.847 0.0725651754
## 9 Population 9 Employment Center 2 4140.112 0.0020088998
```

```
#filter(od_table, Destination == "Employment Center 2") %>% select(Origin, Destination, distance, f) %>
```

```
filter(od_table, Destination == "Employment Center 3") %>% select(Origin, Destination, distance, f)
```

```
##         Origin         Destination  distance           f
## 1 Population 1 Employment Center 3 6899.7724 3.200371e-05
## 2 Population 2 Employment Center 3 7018.4511 2.678478e-05
## 3 Population 3 Employment Center 3 4527.2137 1.124046e-03
## 4 Population 4 Employment Center 3 3629.1377 4.323428e-03
## 5 Population 5 Employment Center 3  676.0903 3.627159e-01
## 6 Population 6 Employment Center 3 1220.2840 1.603452e-01
## 7 Population 7 Employment Center 3 2329.7116 3.036188e-02
## 8 Population 8 Employment Center 3 2296.5402 3.191082e-02
## 9 Population 9 Employment Center 3 3138.4741 9.025411e-03
```

```
#filter(od_table, Destination == "Employment Center 3") %>% select(Origin, Destination, distance, f) %>
```

Next, join the sums of the impedances to the OD-table:

```
od_table <- od_table %>%
  left_join(sum_impedance,
            by = "Destination")
```

Calculate the proportional allocation factor:

```
od_table <- od_table %>%
  mutate(f_c = f / sum_impedance)
```

Calculate the jobs allocated to each population center from each employment center:

```
od_table <- od_table %>%
  mutate(v_ij = Jobs * f_c)
```

Verify that the sum of all jobs allocated is consistent with the total number of jobs:

```
sum(od_table$V_ij)
```

```
## [1] 4500
```

We will now calculate the availability using the two proportional allocation factors simultaneously:

```
sum_pa <- od_table %>%
  group_by(Destination) %>%
  summarize(sum_pa= sum(f_p * f_c))
sum_pa
```

```
## # A tibble: 3 x 2
##   Destination         sum_pa
##   <fct>                <dbl>
## 1 Employment Center 1 0.0690
## 2 Employment Center 2 0.126
## 3 Employment Center 3 0.181
```

Next, join the sums of the impedance to the OD-table:

```
od_table <- od_table %>%
  left_join(sum_pa,
            by = "Destination")
```

Calculate the combined proportional allocation factor:

```
od_table <- od_table %>%
  mutate(f_t = (f_p * f_c) / sum_pa)
```

Calculate the jobs allocated to each population center from each employment center:

```
od_table <- od_table %>%
  mutate(V_ij = Jobs * f_t)
```

Verify that the sum of all jobs allocated is consistent with the total number of jobs:

```
sum(od_table$V_ij)
```

```
## [1] 4500
```

The spatial availability of jobs for each population center is the sum of the jobs proportionally allocated to them from each employment center:

```
availability <- od_table %>%
  group_by(Origin) %>%
  summarize(availability = sum(V_ij))
availability
```

```
## # A tibble: 9 x 2
##   Origin       availability
##   <fct>               <dbl>
## 1 Population 1         67.0
## 2 Population 2        414.
## 3 Population 3        336.
## 4 Population 4        745.
## 5 Population 5       2081.
## 6 Population 6        270.
## 7 Population 7        256.
## 8 Population 8        316.
## 9 Population 9         14.9
```

Again, check that the spatial availability is consistent with the total:

```
sum(availability$availability)
```

```
## [1] 4500
```

Pack all these procedure of proportional allocation of opportunities into a function:

```
p_allocation <- function(x, o_id, d_id, pop, opp, r, f, alpha = 1){
  # The input is an OD-table with Origins, Destinations, Population, Jobs, and a pre-calculated impedan
  #' Multiply two numbers
  #
  #' @param x A data frame with origin-destination information, including identifiers for origins, dest
  #' @param o_id  A character string with the name of the column in data frame x that contains the uniq
  #' @param d_id  A character string with the name of the column in data frame x that contains the uniq
  #' @param pop   A character string with the name of the column in data frame x that contains the popu
  #' @param opp   A character string with the name of the column in data frame x that contains the oppo
  #' @param r     A character string with the name of the column in data frame x that contains catchmen
  #' @param f     A character string with the name of the column in data frame x that contains the valu
  #' @param alpha A number with the parameter \alpha
  #' @return A vector with the number of opportunities available to o_id from d_id
  #' @export

  x <- x %>%
    rename(o = as.name(o_id),
           d = as.name(d_id),
           p = as.name(pop),
           w = as.name(opp),
           f = as.name(f))

  if(!missing(r)){
    x <- x %>%
      rename(r = as.name(r))
  }else{
    r = 1
  }

  sum_pop <- x %>%
    group_by(d) %>%
    summarize(sum_pop = sum(r * p^alpha))

  x <- x %>%
    left_join(sum_pop,
              by = "d")

  x <- x %>%
    mutate(f_p = r * p^alpha / sum_pop)

  sum_impedance <- x %>%
    group_by(d) %>%
    summarize(sum_impedance = sum(f))

  x <- x %>%
    left_join(sum_impedance,
              by = "d")

  x <- x %>%
    mutate(f_c = f / sum_impedance)

  sum_pa <- x %>%
```

```
    group_by(d) %>%
    summarize(sum_pa= sum(f_p * f_c))

  x <- x %>%
    left_join(sum_pa,
              by = "d")

  x <- x %>%
    mutate(f_t = (f_p * f_c) / sum_pa)

  x <- x %>%
    mutate(v_ij = w * f_t)

  return(x$v_ij)
}
```

Examples of applications

Refresh the od_table:

```
load("od_table.rda")
```

Recalculate the impedance function:

```
beta <- 0.0015
od_table <- od_table %>%
  mutate(f = exp(-beta * distance))
```

Scenario 1

Calculate the proportional allocation of jobs to population:

```
od_table <- od_table %>%
  mutate(V_ij = p_allocation(.,
                                  o_id = "Origin",
                                  d_id = "Destination",
                                  pop = "Population",
                                  opp = "Jobs",
                                  f = "f"))
```

Verify that the sum of all jobs allocated is consistent with the total number of jobs:

```
sum(od_table$V_ij)
```

```
## [1] 4500
```

The total number of jobs is preserved.

Aggregate available jobs by origin:

```
availability <- od_table %>%
  group_by(Origin) %>%
  summarize(avail_jobs = sum(V_ij))
availability
```

```
## # A tibble: 9 x 2
##   Origin       avail_jobs
```

```
##   <fct>             <dbl>
## 1 Population 1       67.0
## 2 Population 2      414.
## 3 Population 3      336.
## 4 Population 4      745.
## 5 Population 5     2081.
## 6 Population 6      270.
## 7 Population 7      256.
## 8 Population 8      316.
## 9 Population 9       14.9
```

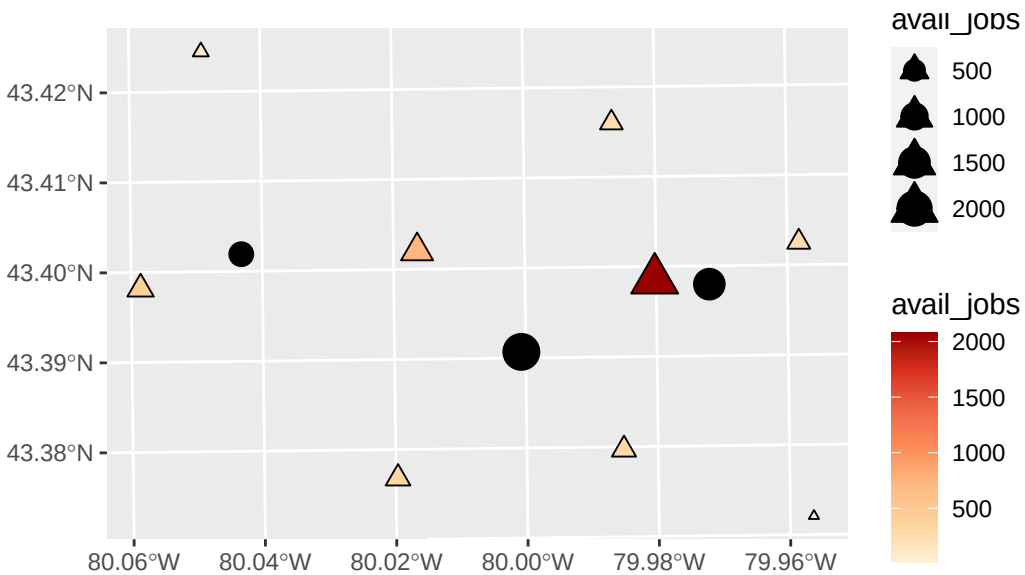
Join the availability to the simulated_data:

```
simulated_data <- simulated_data %>%
  left_join(availability,
            by = c("id" = "Origin"))
```

Plot the availability estimates:

```
avail_jobs_plot <- ggplot() +
  geom_sf(data = simulated_data %>%
            filter(type == "population"),
          aes(color = avail_jobs,
              size = avail_jobs),
          shape = 17) +
  geom_sf(data = simulated_data %>%
            filter(type == "population"),
          aes(size = avail_jobs),
          shape = 2) +
  geom_sf(data = simulated_data %>%
            filter(type == "jobs"),
          aes(size = number,
              shape = )) +
  scale_color_distiller(palette = "OrRd",
                        direction = 1)

avail_jobs_plot
```



How do we interpret this? Accessibility is the number of jobs that can be reached at a given cost. Here, the total number of jobs is a constant. Population center 5 has the greatest availability, due to being a large population center that is moreover relatively close to jobs.

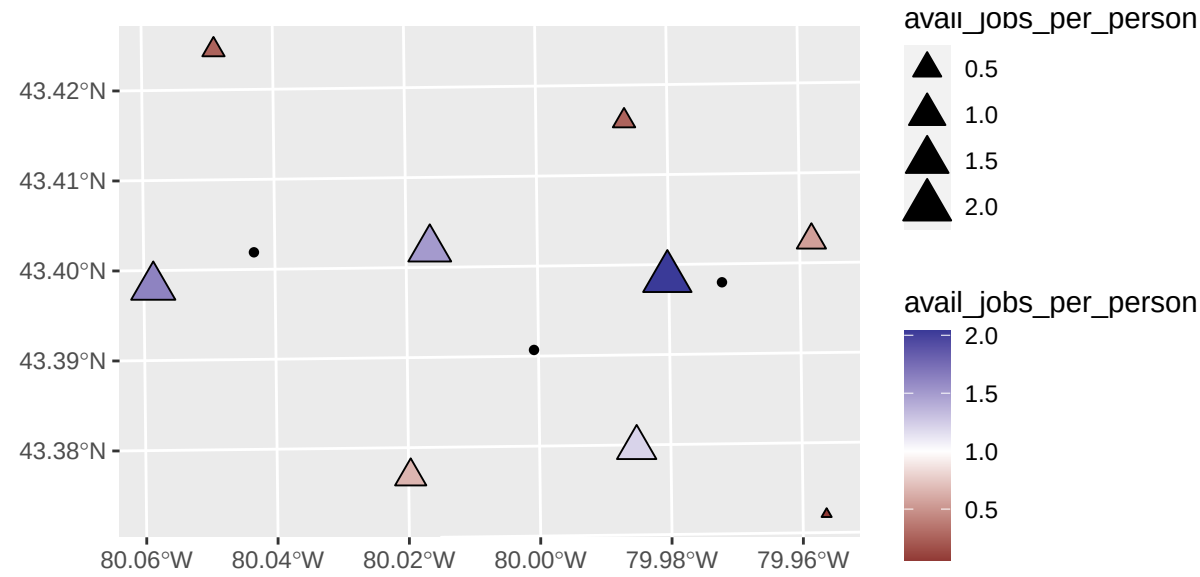
Since the total number of jobs is constant, we can calculate the available jobs per person:

```
simulated_data <- simulated_data %>%
  mutate(avail_jobs_per_person = avail_jobs/number)
```

Plot the availability per person:

```
avail_jobs_person_plot <- ggplot() +
  geom_sf(data = simulated_data %>%
            filter(type == "population"),
          aes(color = avail_jobs_per_person,
              size = avail_jobs_per_person),
          shape = 17) +
  geom_sf(data = simulated_data %>%
            filter(type == "population"),
          aes(size = avail_jobs_per_person),
          shape = 2) +
    geom_sf(data = simulated_data %>%
            filter(type == "jobs"),
              shape = 16) +
  scale_color_gradient2(midpoint = 1)


avail_jobs_person_plot
```



Some population centers have almost two jobs available per person, and others less than one job available per person. This does not mean that people are not taking some of the jobs. It means that controlling for the cost of reaching jobs, they are worse off than those with more jobs spatially available.

Scenario 2

We can also examine the pool of workers available to each employment center by considering the workers as opportunities and the jobs as the population.

Calculate the proportional allocation of jobs to population:

```
od_table <- od_table %>%
  mutate(W_ij = p_allocation(.,
                                  o_id = "Destination",
                                  d_id = "Origin",
                                  pop = "Jobs",
                                  opp = "Population",
                                  f = "f"))
```

Verify that the sum of all jobs allocated is consistent with the total number of jobs:

```
sum(od_table$W_ij)
```

```
## [1] 4525
```

The total population is preserved.

Aggregate available workers by employment center:

```
availability <- od_table %>%
  group_by(Destination) %>%
  summarize(avail_workers = sum(W_ij))
availability
```

```
## # A tibble: 3 x 2
##   Destination         avail_workers
##   <fct>                       <dbl>
## 1 Employment Center 1          610.
## 2 Employment Center 2         1710.
## 3 Employment Center 3         2205.
```

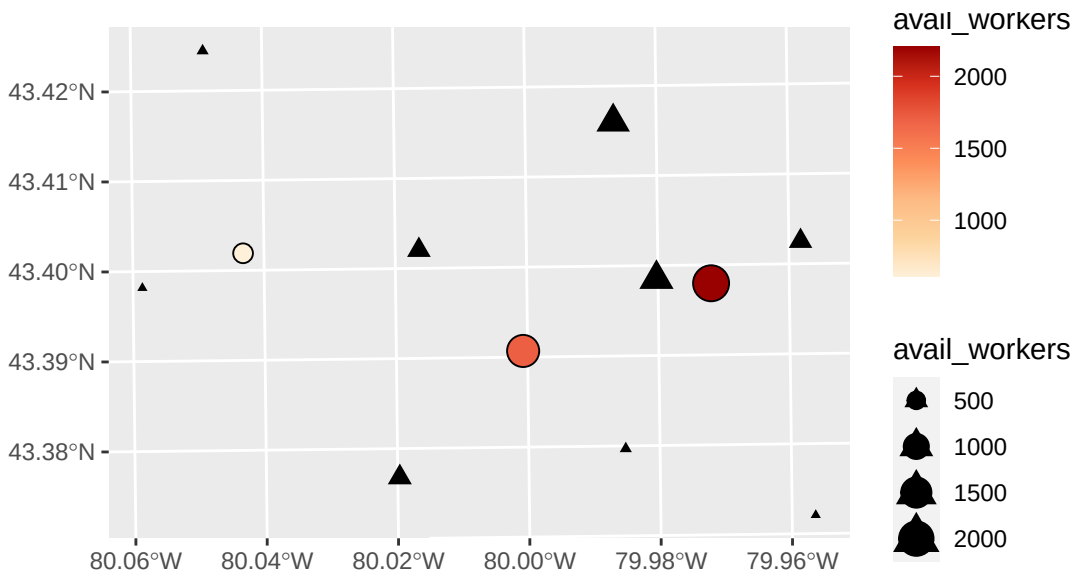
Join the availability to the simulated_data:

```
simulated_data <- simulated_data %>%
  left_join(availability,
            by = c("id" = "Destination"))
```

Plot the availability estimates:

```
avail_workers_plot <- ggplot() +
  geom_sf(data = simulated_data %>%
            filter(type == "jobs"),
          aes(color = avail_workers,
              size = avail_workers),
          shape = 16) +
  geom_sf(data = simulated_data %>%
            filter(type == "jobs"),
          aes(size = avail_workers),
          shape = 1) +
  geom_sf(data = simulated_data %>%
            filter(type == "population"),
          aes(size = number),
              shape = 17) +
  scale_color_distiller(palette = "OrRd",
                        direction = 1)

avail_workers_plot
```



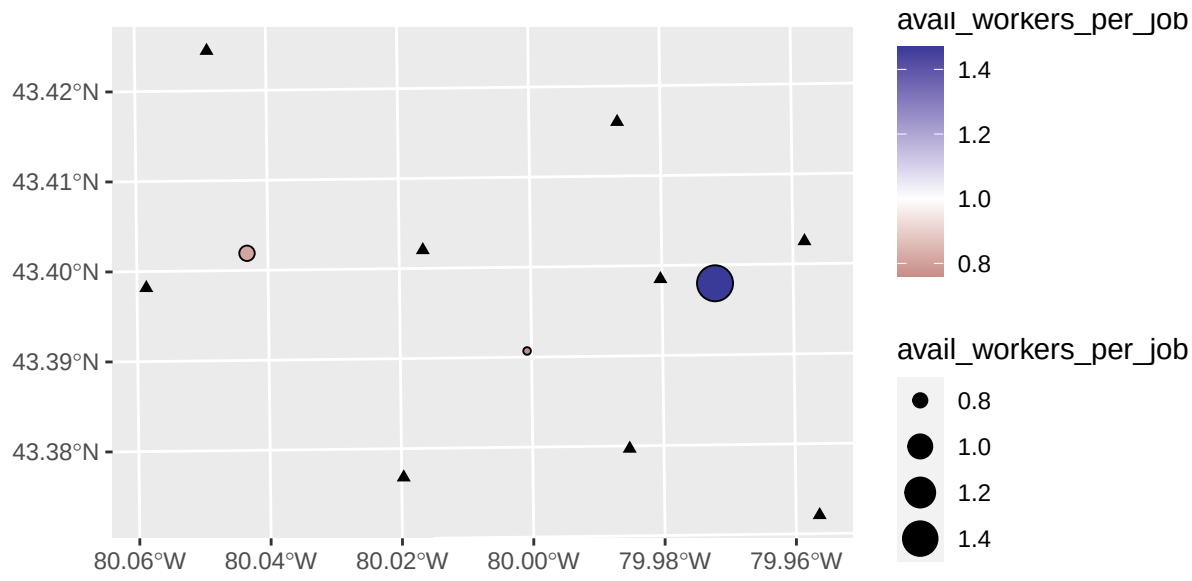
Since the total population is constant, we can calculate the available workers per job:

```
simulated_data <- simulated_data %>%
  mutate(avail_workers_per_job = avail_workers/number)
```

Plot the population availability per job:

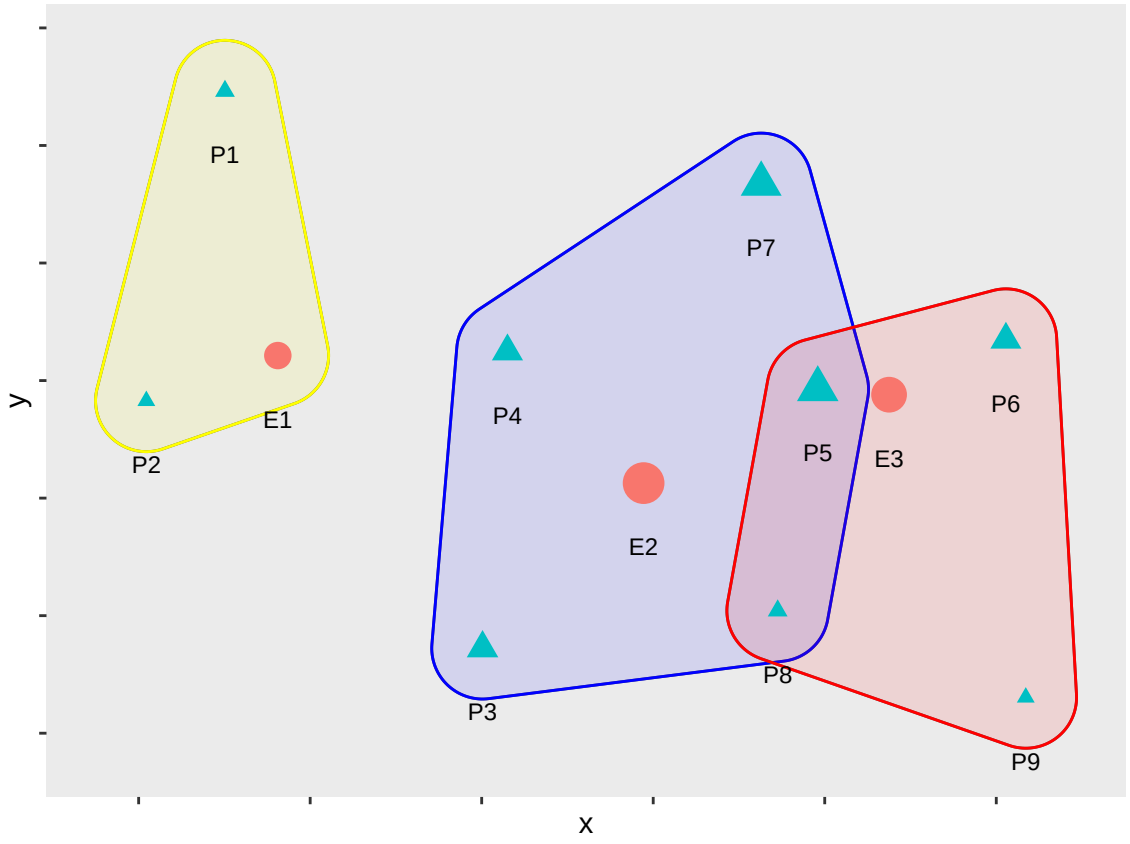
```
avail_workers_job_plot <- ggplot() +
  geom_sf(data = simulated_data %>%
            filter(type == "jobs"),
          aes(color = avail_workers_per_job,
              size = avail_workers_per_job),
          shape = 16) +
  geom_sf(data = simulated_data %>%
            filter(type == "jobs"),
          aes(size = avail_workers_per_job),
          shape = 1) +
  geom_sf(data = simulated_data %>%
            filter(type == "population"),
              shape = 17) +
  scale_color_gradient2(midpoint = 1)

avail_workers_job_plot
```



Scenario 3

In this scenario we introduce catchment/eligibility constraints. Due to differences in educational achievement among the population, the jobs in Employment Center 1 can only be taken by individuals in population centers 1 and 2. Jobs in Employment Center 2 can be taken by individuals in population centers 3, 4, 5, 7, and 8. Lastly, jobs in Employment Center 3 require qualifications available only among individuals in population centers 5, 6, 8, and 9. See figure below.



Calculate the proportional allocation of opportunities but now with catchment constraints.:

```
od_table <- od_table %>%
  mutate(V_ij_r = p_allocation(.,
                               o_id = "Origin",
                               d_id = "Destination",
                               pop = "Population",
                               opp = "Jobs",
                               r = "catchments",
                               f = "f"))
```

Verify that the sum of all jobs allocated is consistent with the total number of jobs:

```
sum(od_table$V_ij_r)
```

```
## [1] 4500
```

The total number of jobs is preserved.

Repeat the availability calculation:

```
availability <- od_table %>%
  group_by(Origin) %>%
  summarize(avail_r = sum(V_ij_r))
availability
```

```
## # A tibble: 9 x 2
##   Origin       avail_r
##   <fct>          <dbl>
## 1 Population 1  104.
## 2 Population 2  646.
## 3 Population 3  303.
## 4 Population 4  527.
## 5 Population 5 2173.
## 6 Population 6  257.
## 7 Population 7  161.
## 8 Population 8  322.
## 9 Population 9    7.52
```

Join the availability to the simulated_data:

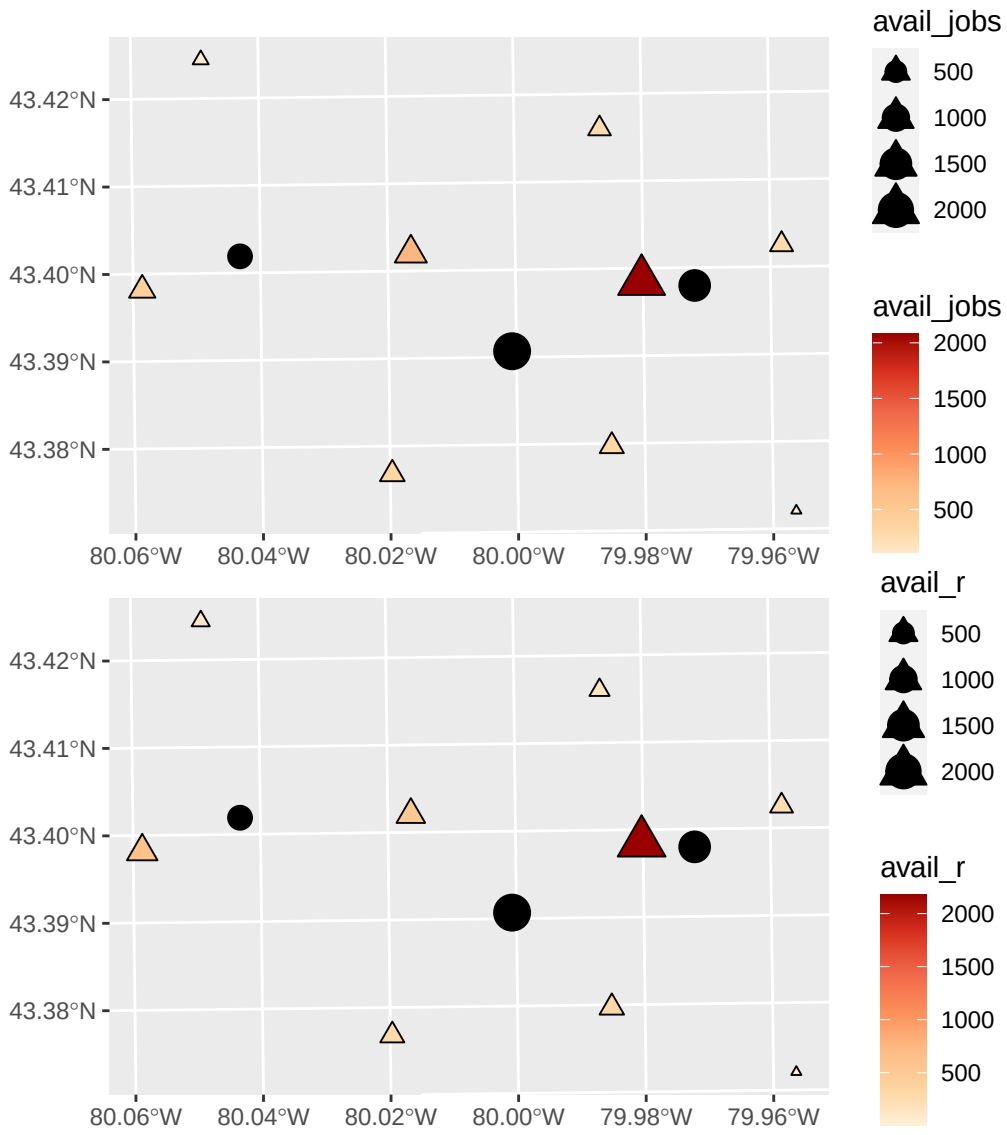
```
simulated_data <- simulated_data %>%
  left_join(availability,
            by = c("id" = "Origin"))
```

Plot the availability estimates:

```
avail_r_plot <- ggplot() +
  geom_sf(data = simulated_data %>%
            filter(type == "population"),
          aes(color = avail_r,
              size = avail_r),
          shape = 17) +
  geom_sf(data = simulated_data %>%
            filter(type == "population"),
          aes(size = avail_r),
          shape = 2) +
  geom_sf(data = simulated_data %>%
            filter(type == "jobs"),
```

```
          aes(size = number,
              shape = )) +
  scale_color_distiller(palette = "OrRd",
                        direction = 1)

avail_jobs_plot / avail_r_plot
```



Available jobs per person with catchment/eligibility conditions:

```
simulated_data <- simulated_data %>%
  mutate(avail_r_per_person = avail_r/number)
```

Plot the availability per person:

```
avail_r_person_plot <- ggplot() +
  geom_sf(data = simulated_data %>%
            filter(type == "population"),
          aes(color = avail_r_per_person,
```

```
              size = avail_r_per_person),
          shape = 17) +
  geom_sf(data = simulated_data %>%
            filter(type == "population"),
          aes(size = avail_r_per_person),
          shape = 2) +
    geom_sf(data = simulated_data %>%
            filter(type == "jobs"),
              shape = 16) +
  scale_color_gradient2(midpoint = 1)

avail_jobs_person_plot / avail_r_person_plot
```