Suppose an urban area is divided up into $Z$ zones. As planners, we are interested in choosing the location of a new facility, like a library.

We assume that each zone’s population is concentrated at the zone’s areal centroid, which we assume is known. (We can make this more plausible by thinking of our urban area as divided up into smaller, ie more, zones). The number of users of the facility depends on the facility’s accessibility, which we measure by travel cost: the more it costs people to get to the facility, the less they will use it. We assume that this dependence on cost is linear in travel distance. If $P_j$ is the population in zone $j$ (or, alternatively, the total number of users in zone $j$ if the facility were to be located in zone $j$, so that transport costs for zone-$j$ users would be zero) and $k_j$ is the transport cost per mile for these residents, the number of users will be

$$u_j = P_j - k_j d(j, F)$$

where $d(j, F)$ is the distance from the centroid of zone $j$ to the facility. We assume that everything takes place on a von Thunen plain, and that transport is available in any direction.

Total transport costs for the $u_j$ users of the facility are

$$c_j = u_j k_j d(j, F)$$

and we want to choose the facility location $F = (F_s, F_t)$ to minimize aggregate (total) transportation costs:

$$TTC = \sum_{j=1}^{Z} c_j = \sum_{j=1}^{Z} (P_j - k_j d(j, F)) k_j d(j, F)$$

Note that this is much like the Weber problem, except that there may be more than two resources (zones), and there’s no element of the total transportation costs corresponding to the shipment of final product from factory to market. (It would be even closer if we assumed that the number of users in each zone was fixed at $P_j$). As with the Weber problem, this must be solved numerically.

The script `fac-loc.R` (included in `weber-script.zip`) has functions to solve this problem in R. The script works much like the Weber problem script, except that the data structure calls the 3rd column `pop`, and it is supposed to be $P_j$ as defined above. The $s$- and $t$-coordinates are meant to represent the zonal centroids. We take the distance metric $d$ to be the Euclidean metric: you could experiment by defining other metrics if you’re interested.\(^1\) Given a data structure `dat` you can solve the facility location problem by loading the script and then doing `facloc(dat)`. You can get a plot by doing `facloc(dat,plot=TRUE)`. The plot shows the convex hull of the zonal centroids, plus the optimal location as a black dot.

The script comes with an example dataset, with 5 zones. If you run the problem with this data, you should find that the best location for the facility is at $(0.9345, 2.0061)$.\(^1\)

---

\(^1\)The cost function calls the function `dist` to compute $d(j, F)$, and the script defines `dist` to be `edist`, which computes the Euclidean distance. So to use another metric, define, say `metric` as a function that does the computation using your metric: this should have the same arguments as `edist` in the same order. Then define `dist<-mydist`, to use it in computations. To restore the Euclidean distance metric just redefine `dist` to be `edist`: `dist<-edist`. Caution: your metric should be continuous in the $s$- and $t$-coordinates.
There is also a function that allows you try out your own solution, to see how it compares to the optimal one. First define a candidate location via (say) myloc <- c(1.0,1.0). Then see what happens if you run testloc(myloc,dat). (Of course, you can bypass the explicit definition of the candidate point by doing testloc(c(1.0,1.0),dat) directly). The display of the answer in this case looks very much like the display for the optimal solution, except that now it begins For this location choice: whereas the optimal solution begins Optimal solution: . You should find that, with the given data (dat), locating the facility at (1.0, 1.0) results in a total transport cost of 144.91, as opposed to the optimal (lowest possible) cost of 134.33. You can also plot your guess by including plot=TRUE when you run testloc. However, note that the plot can be misleading: it shows just the zonal centroids and not their populations, which is really what matters.