

W2 Exercise: Introduction to basic mapmaking in R

Scatter Map, Graduated Symbolology, Basemap / 40 points

There are a lot of great resources for mapmaking in R. Here is just a simple introduction. Thanks for Eric Delmelle (Geography, University of North Carolina at Charlotte) for refining this exercise.

Part 1: Map with dots (scattermap)

R makes placing dots on a map easy! The "maps" package does most of the work. Go ahead and install it via the Package installer, or use `install.packages()` in the console. When installed, load it into the workspace:

```
library(maps)
```

Next step: load the data. Use geocoded locations of [Costco Wholesale stores](#), available on Canvas:

```
costcos <- read.csv("[path-to-file]", sep=",")
```

Now on to mapping. When you create your maps, it is useful to think of them as layers (regardless of what software you use). The bottom layer is usually the base map that shows geographical boundaries, and then you place layers on top of that. In case the bottom layer is a map of the United States, the second layer is Costco locations. Here's how to make the first layer:

```
map(database="state")
```

The second layer, or Costco's, are then mapped with the `symbols()` function, to which you pass latitudes and longitudes of the stores. Also set `add` to `TRUE` to indicate that you want symbols to be added to the map rather than creating a new plot)

```
symbols(costcos$Longitude, costcos$Latitude, circles=rep(1, length(costcos$Longitude)),  
inches=0.05, add=TRUE)
```

All the circles are the same size because you set `circles` to an array of ones with the length equal to the number of Costco store locations. You also set `inches` to 0.05, which scales the circles to

that number. If you want smaller markers, all you need to do is decrease that value.

You can change the colors of both the map and the circles so that the locations stand out and boundary lines sit in the background. Change the dots to a nice Costco red and the state boundaries to a light gray.

```
map(database="state", col="#cccccc")
```

```
symbols(costcos$Longitude, costcos$Latitude, bg="#e2373f", fg="ffffff",lwd=0.5, circles=rep(1,
length(costcos$Longitude)), inches=0.05, add=TRUE)
```

In your first attempt, the unfilled circles and the state boundaries were all the same color and line width, so everything blended together, but with the right colors, you can make the data sit front and center.

Not bad for a few lines of code! Costco has clearly focused on opening locations on the coasts with clusters in southern and northern California, northwest Washington and in the northeast of the country.

However, there is a glaring omission here. Where are Alaska and Hawaii? They are part of the United States too, but are nowhere to be found even though you use the “state” database with `map()`. The two states are actually included in the “world” database, so if you want to see Costco store locations in Alaska and Hawaii, you need to map the entire world.

```
map(database="world", col="#cccccc")
```

```
symbols(costcos$Longitude, costcos$Latitude, bg="#e2373f", fg="ffffff",lwd=0.3, circles=rep(1,
length(costcos$Longitude)), inches=0.03, add=TRUE)
```

If you want to only map Costco store locations for a few states, you can use the `region` argument.

```
map(database="state", region=c("California", "Nevada", "Oregon", "Washington"),col="#cccccc")
```

```
symbols(costcos$Longitude, costcos$Latitude, bg="#e2373f", fg="ffffff",lwd=0.5, circles=rep(1,
length(costcos$Longitude)), inches=0.05, add=TRUE)
```

Some dots are not in those states but they’re in the plotting region, so they still appear.

Task 1: Produce a similar map with your own point data. You may have the locations of your favorite breweries, farmer’s markets in Salt Lake City, thunderstorms, hurricane landfalls, or you may have collected the locations of tweets or trees or your some GPS locations of your cat. If you don’t have any of that, be creative and construct some random points! Also, change the appearance of the dots. Check the documentation [here](#). You have multiple options (size, colors, shape, fill, outline).

Part 2: Graduated symbols

Often you don't just have locations, you have another value attached to them, such as sales for business locations, or city population. You can still map with points, but you can take the principle of the bubble plot (graduated symbols) and use it on a map.

Remember to size circles by area, not by radius, ok?

In this example, we look at adolescent fertility rates as reported by the United Nations Human Development Report – that is, the number of births per 1,000 women aged 15 to 19 in 2008. You want to size the circles in proportion to these rates.

The code is similar to Part 1, but remember you passed an array of ones for circle size in the `symbols()` function. Instead, we use `sqrt()` of the rates to indicate size.

```
fertility <- read.csv("[path-to-file]", sep=",")
```

```
map("world", fill=FALSE, col="#cccccc")
```

```
symbols(fertility$longitude, fertility$latitude, circles=sqrt(fertility$ad_fert_rate), add=TRUE,  
inches=0.15, bg="#93ceef", fg="#ffffff")
```

Task 2: Produce a map using graduated symbols with your own point data. Change the appearance of the bubbles.