

Drawing maps with R

Making maps in R

- Spatial data comes with locations (perhaps with information about those locations).
- A good way to draw spatial data is on a map.
- The leaflet package is the easiest way to draw maps in R.
- Install these two packages, with two familiar ones:

```
1 library(tmaptools)
2 library(leaflet)
3 library(tidyverse)
4 library(conflicted)
5 conflicts_prefer(dplyr::mutate)
6 conflicts_prefer(dplyr::arrange)
```

Hockey league map

The Ontario hockey divisions (the last example for cluster analysis) came with a very bad map. Can we do better?

- reload the Ontario road distances

```
1 my_url <-  
2   "http://ritsokiguess.site/datafiles/ontario-road-distances.c  
3 # my_url <- "ontario-road-distances.csv"  
4 ontario <- read_csv(my_url)
```

Ontario road distances (some)

1 ontario

place <chr>	Barrie <dbl>	Belleville <dbl>	
Barrie	0	260	
Belleville	260	0	
Brantford	190	290	
Brockville	405	155	
Cornwall	500	250	
Hamilton	145	255	
Huntsville	125	280	
Kingston	330	75	
Kitchener	180	280	

Place	Barrie	260	Belleville	360
<small><chr></small>	<small><dbl></small>		<small><dbl></small>	

1-10 of 21 rows | 1-3 of 22 columns [Previous](#) [1](#) [2](#) [3](#) [Next](#)

Grab the places

- and append province (“ON”) for reasons shortly to become clear:

```
1 tibble(place = ontario$place) %>%  
2   mutate(prov = "ON") %>%  
3   unite(place1, c(place, prov), sep = " ") -> ontario2  
4 ontario2
```


Geocode 1/2

- find their latitudes and longitudes (“geocode”; slow).
- Save the geocoded places.

```
1 ontario2 %>%  
2   rowwise() %>%  
3   mutate(ll = list(geocode_OSM(place1))) -> d
```

```
1 d
```


Geocode 2/2

Untangle the lats and longs:

```
1 d %>%  
2   unnest_wider(ll) %>%  
3   unnest_wider(coords) -> ontario3  
4 ontario3
```


Make map

- finally:

```
1 leaflet(data = ontario3) %>%  
2   addTiles() %>%  
3   addCircleMarkers(lng = ~x, lat = ~y)
```





Cluster analysis revisited

```
1 ontario %>% select(-1) %>% as.dist() -> ontario.d  
2 ontario.hc <- hclust(ontario.d, method = "ward.D")
```

Seven clusters:

```
1 plot(ontario.hc)
2 rect.hclust(ontario.hc, 7)
```

Get the clusters

```
1 tibble(place = ontario$place, cluster = cutree(ontario.hc, 7))  
2 clusters %>% arrange(cluster)
```


Combine clusters

- combine clusters 6 and 7 with 4 (“north”)
- combine clusters 2 and 3 (“east”)
- make named divisions

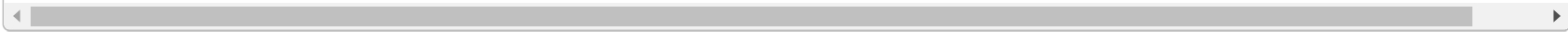
```
1 clusters %>%  
2   mutate(division = fct_collapse(factor(cluster),  
3                                   "north" = c("4", "6", "7"),  
4                                   "east" = c("2", "3"),  
5                                   "west" = "5",  
6                                   "central" = "1")) %>%  
7   arrange(division) -> divisions
```

The divisions

1 divisions

Take “ON” off of **ontario3**

```
1 ontario3 %>%  
2   mutate(place = str_replace(place1, " ON$", "")) -> ontario3  
3 ontario3
```



Add the divisions, matching by place

- and draw map

```
1 pal <- colorFactor("Set1", divisions$division)
2
3 ontario3 %>% left_join(divisions) %>%
4   select(place, x, y, division) %>%
5   leaflet() %>%
6   addTiles() %>%
7   addCircleMarkers(lng = ~x, lat = ~y,
8                     color = ~pal(division))
```

+

-



Original seven clusters

The same idea gets a map of the original seven clusters:

```
1 pal <- colorFactor("Set1", divisions$cluster)
2 ontario3 %>% left_join(divisions) %>%
3   select(place, x, y, cluster) %>%
4   leaflet() %>%
5   addTiles() %>%
6   addCircleMarkers(lng = ~x, lat = ~y,
7                     color = ~pal(cluster))
```

±

—

