

Drawing maps with Leaflet

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:

```
library(tmaptools)
library(leaflet)
library(tidyverse)
library(conflicted)
```

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

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

```
## Rows: 21 Columns: 22  
## -- Column specification -----  
## Delimiter: ","  
## chr  (1): place  
## dbl  (21): Barrie, Belleville, Brantford, Brockvi...  
##  
## i Use `spec()` to retrieve the full column specification for  
## i Specify the column types as get `show_col_types = FALSE`
```

Ontario road distances (some)

```
ontario
```

```
## # A tibble: 21 x 22
```

```
##   place      Barrie Belleville Brantford Brockville
##   <chr>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 Barrie          0        260        190        405
## 2 Belleville    260          0        290        155
## 3 Brantford    190        290          0        420
## 4 Brockville   405        155        420          0
## 5 Cornwall     500        250        535          95
## 6 Hamilton     145        255         40        405
## 7 Huntsville   125        280        300        405
## 8 Kingston     330         75        340          80
## 9 Kitchener    180        280         40        425
## 10 London      260        360         85        510
```

```
## # ... with 11 more rows, and 17 more variables:
```

```
## #   Cornwall <dbl> Hamilton <dbl>
```

Drawing maps with Leaflet

Grab the places

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

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

```
## # A tibble: 21 x 1  
##   place1  
##   <chr>  
## 1 Barrie ON  
## 2 Belleville ON  
## 3 Brantford ON  
## 4 Brockville ON  
## 5 Cornwall ON  
## 6 Hamilton ON  
## 7 Huntsville ON
```

Geocode 1/2

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

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

```
## # A tibble: 21 x 2  
##   place1      ll  
##   <chr>      <list>  
## 1 Barrie ON   <named list [3]>  
## 2 Belleville ON <named list [3]>  
## 3 Brantford ON <named list [3]>  
## 4 Brockville ON <named list [3]>  
## 5 Cornwall ON  <named list [3]>  
## 6 Hamilton ON  <named list [3]>
```

Geocode 2/2

Untangle the lats and longs:

```
d %>%  
  unnest_wider(ll) %>%  
  unnest_wider(coords) -> ontario3  
ontario3
```

```
## # A tibble: 21 x 5  
##   place1      query      x      y bbox  
##   <chr>      <chr>    <dbl> <dbl> <list>  
## 1 Barrie ON   Barrie ON   -79.7  44.4 <bbox>  
## 2 Belleville ON Belleville ON -77.4  44.2 <bbox>  
## 3 Brantford ON Brantford ON -80.3  43.1 <bbox>  
## 4 Brockville ON Brockville ON -75.7  44.6 <bbox>  
## 5 Cornwall ON  Cornwall ON  -74.7  45.0 <bbox>  
## 6 Hamilton ON  Hamilton ON  -79.9  43.3 <bbox>  
## 7 Huntsville ON Huntsville ON -79.2  45.3 <bbox>
```

Make map

- finally:

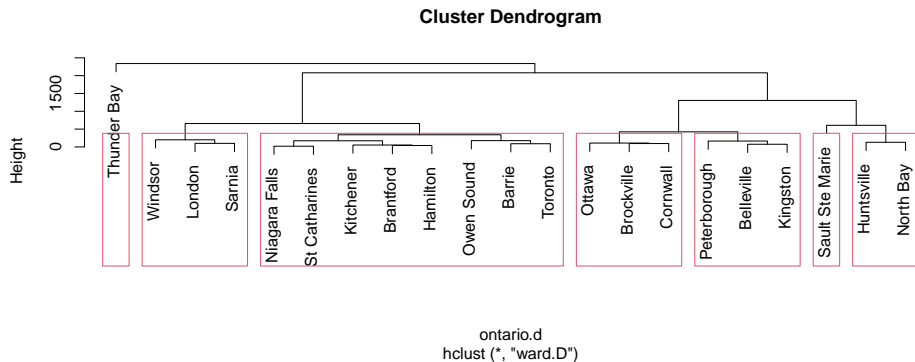
```
leaflet(data = ontario3) %>%  
  addTiles() %>%  
  addCircleMarkers(lng = ~x, lat = ~y) -> locs  
locs
```


Cluster analysis revisited

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

Seven clusters:

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



Get the clusters

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

```
## # A tibble: 21 x 2
##   place      cluster
##   <chr>      <int>
## 1 Barrie      1
## 2 Brantford   1
## 3 Hamilton    1
## 4 Kitchener   1
## 5 Niagara Falls 1
## 6 Owen Sound  1
## 7 St Catharines 1
## 8 Toronto     1
## 9 Belleville  2
## 10 Kingston   2
## # ... with 11 more rows
```

Combine clusters

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

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

The divisions

```
divisions
```

```
## # A tibble: 21 x 3
```

```
##   place      cluster division
```

```
##   <chr>      <int> <fct>
```

```
## 1 Barrie      1 central
```

```
## 2 Brantford   1 central
```

```
## 3 Hamilton    1 central
```

```
## 4 Kitchener   1 central
```

```
## 5 Niagara Falls 1 central
```

```
## 6 Owen Sound  1 central
```

```
## 7 St Catharines 1 central
```

```
## 8 Toronto     1 central
```

```
## 9 Belleville  2 east
```

```
## 10 Brockville 3 east
```

```
## # ... with 11 more rows
```

Take "ON" off of ontario3

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

```
## # A tibble: 21 x 6  
##   place1      query      x      y bbox  place  
##   <chr>      <chr>    <dbl> <dbl> <list> <chr>  
## 1 Barrie ON    Barrie ON  -79.7  44.4 <bbox> Barr~  
## 2 Belleville ON Bellevill~ -77.4  44.2 <bbox> Bell~  
## 3 Brantford ON Brantford~ -80.3  43.1 <bbox> Bran~  
## 4 Brockville ON Brockvill~ -75.7  44.6 <bbox> Broc~  
## 5 Cornwall ON  Cornwall ~ -74.7  45.0 <bbox> Corn~  
## 6 Hamilton ON  Hamilton ~ -79.9  43.3 <bbox> Hami~  
## 7 Huntsville ON Huntsvill~ -79.2  45.3 <bbox> Hunt~  
## 8 Kingston ON  Kingston ~ -76.5  44.2 <bbox> King~  
## 9 Kitchener ON Kitchener~ -80.5  43.5 <bbox> Kitc~  
## 10 London ON   London ON  -81.2  43.0 <bbox> Lond~  
## # ... with 11 more rows
```

Add the divisions, matching by place

- and draw map

```
pal <- colorFactor("Set1", divisions$division)

ontario3 %>% left_join(divisions) %>%
  select(place, x, y, division) %>%
  leaflet() %>%
  addTiles() %>%
  addCircleMarkers(lng = ~x, lat = ~y, color = ~pal(division))

## Joining, by = "place"
```

The map with divisions

```
divmap
```


Original seven clusters

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

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

## Joining, by = "place"
```

Map with seven clusters

```
divmap
```