Maps of boston

Ari Anisfeld

12/7/2021

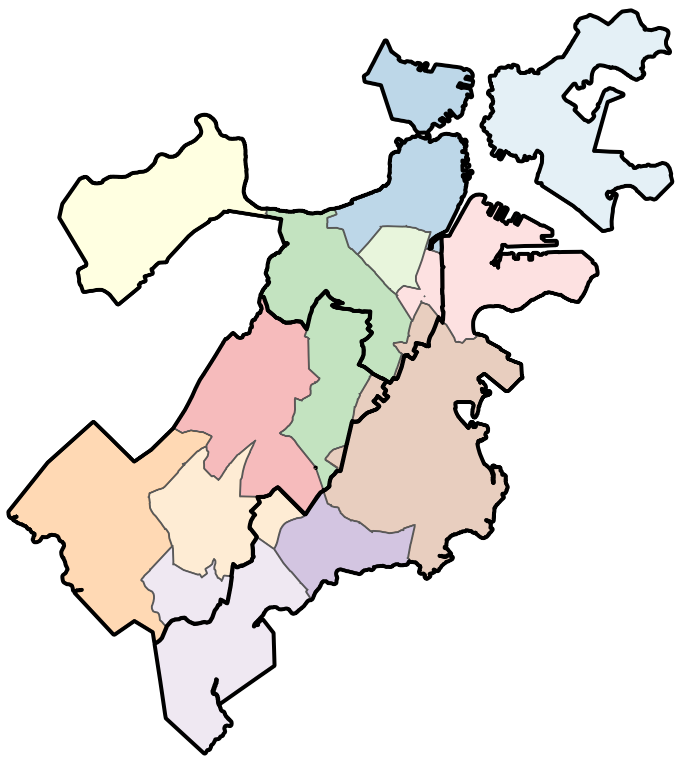
Making maps from geocodes

```r
geocode_to_neighborhood <- read_csv("data/geocode_to_neighborhoods.csv")
geocode_to_zone  <- haven::read_dta("data/geocodetozone.dta")

geocodes <-
  st_read("data/geocodes/Geocodes.shp", quiet = TRUE) %>%
  # geocodes shapefile has small issues which make certain polygons
  # "invalid" (e.g. st_is_valid(.) returns FALSE).
  # this buffering trick fixes those issues.
  st_simplify() %>%
  st_buffer(dist = 0) %>%
  left_join(geocode_to_zone, "geocode") %>%
  left_join(geocode_to_neighborhood, "geocode") %>%
  # some codes are not assigned zones or neighborhoods.
  mutate(neighborhoodid2 = case_when(!is.na(neighborhoodid2) ~ neighborhoodid2,
                                  geocode == 208 ~ 5,
                                  geocode == 652 ~ 6,
                                  geocode %in% c(729, 746) ~ 8,
                                  geocode == 372 ~ 12
                                ),
         zones = case_when(!is.na(zones) ~ zones,
                           geocode == 208 ~ 2,
                           geocode == 729 ~ 3,))

these_zones <-
  combine_and_clean_polygons(geocodes, zones, 100)

these_neighborhoods <-
  combine_and_clean_polygons(geocodes, neighborhoodid2, 100)
```

To pick colors, you can use a predefined scale: This resource walks through the options: https://cfss.uchicago.edu/notes/optimal-color-palettes/ The downside of this option is that many scales don't have 12 options and default to white. Alternatively, you can pick colors "by hand" the colors in `colors_1` and `colors_2` are represented in hexidecimal form and picked from here: https://colorbrewer2.org/#type=qualitative&scheme=Paired&n=12.

```r
colors_1 <- c('#a6cee3','#1f78b4','#b2df8a','#33a02c','#fb9a99','#e31a1c','#fdbf6f','#ff7f00','#cab2d6'
colors_2 <- c('#8dd3c7','#ffffb3','#bebada','#fb8072','#80b1d3','#fdb462','#b3de69','#fccde5','#d9d9d9'


ggplot() +
  geom_sf(data = these_neighborhoods,
          aes(fill = as.factor(neighborhoodid2)), lwd = .3, alpha = .3, show.legend = FALSE)  +
  geom_sf(data = these_zones, lwd = .7, fill=NA, color = "black") +
  theme_void() +
  scale_fill_manual( values = colors_1) +
  # uncomment the following line to see manual colors (also comment out the above line)
  # scale_fill_brewer(palette = "Spectral") +
  labs(title = "Geocode-based map putting neighborhoods in foreground")
```

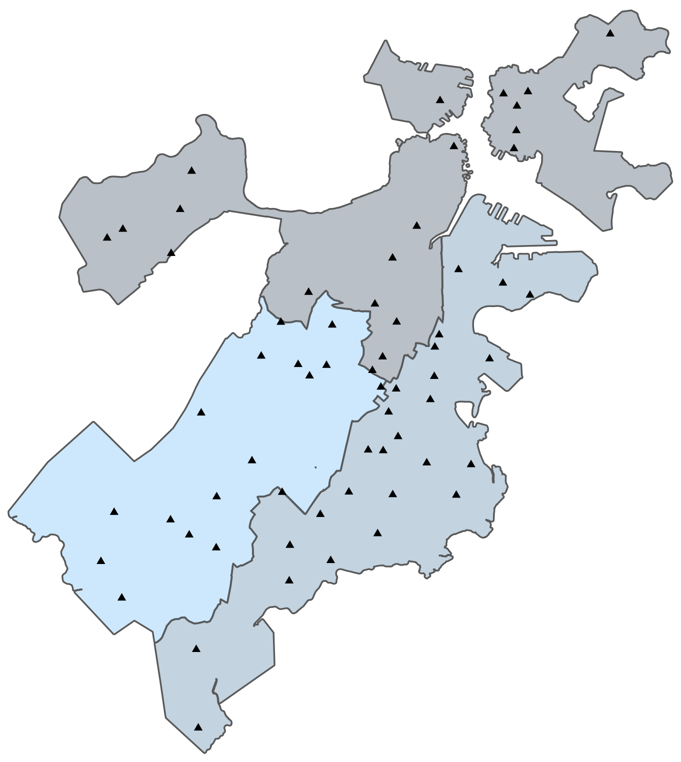
Geocode−based map putting neighborhoods in foreground



Map

```
pre_schools <- st_read("data/prekschools10/prekschools10.shp", quiet = TRUE)

ggplot() +
  geom_sf(data = these_zones,
          aes(fill = zones), lwd = .3, alpha = .3, show.legend = FALSE) +
  geom_sf(data = pre_schools, shape = 17, size = .8) +
theme_void() +
  labs(title = "Geocode-based map putting zones in foreground")
```

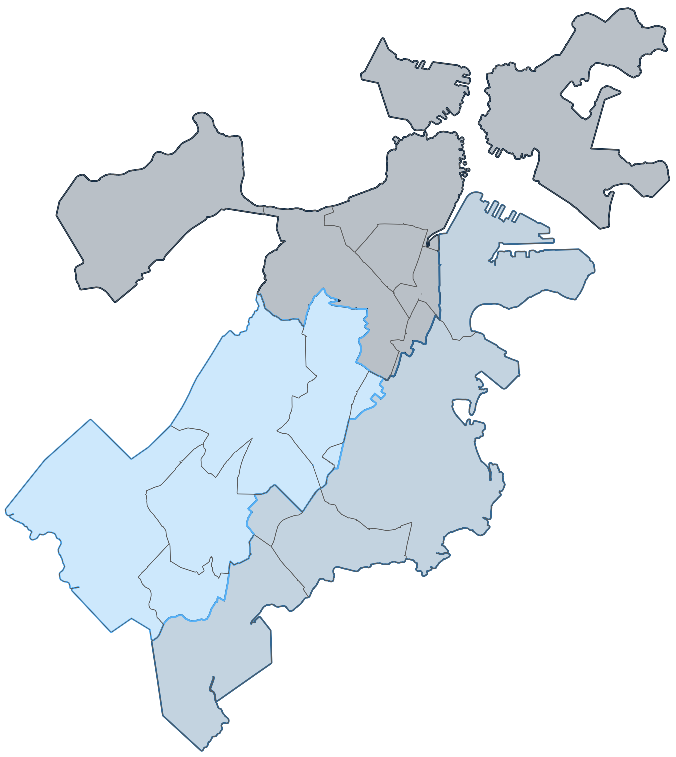
Geocode−based map putting zones in foreground



Map with zones foregrounded

```
ggplot() +
  geom_sf(data = these_zones,
          aes(fill = zones, color = zones),
          lwd = .3, alpha = .3, show.legend = FALSE)  +
  geom_sf(data= these_neighborhoods,
          lwd = .1, fill=NA) +
  # geom_sf_label(aes(label = zones),data = these_zones) +
    theme_void() +
  labs(title = "Using geocodes")
```

Using geocodes



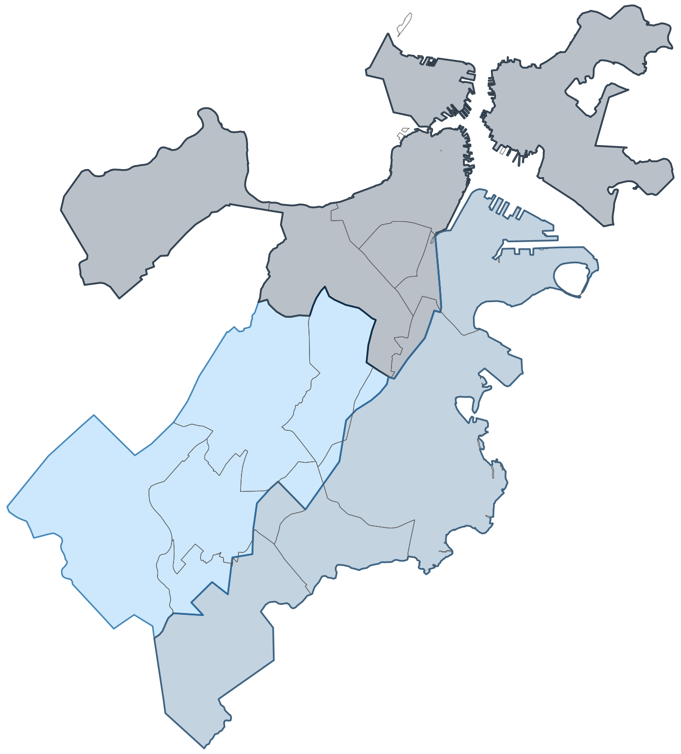
Appendix:

Using non zone data

This first pass uses the neighborhood shapes from Boston open data. (https://bostonopendata-boston.opendata.arcgis.com/datasets/3525b0ee6e6b427f9aab5d0a1d0a1a28_0) and using the zones shapefile you previously made.

```
zones <- st_read("data/zones/zones.shp", quiet = TRUE) %>%
          transmute(zone = ifelse(!is.na(north), north, 3),
                 geometry)

neighborhoods_xwalk <-
  read_csv("data/neighborhood_crosswalk.csv",
          skip = 1,
          col_names = c("Name", "neighborhood_code", "x"))  %>%
  select(-x)

raw_neighborhoods <-
    st_read("data/Boston_Neighborhoods/Boston_Neighborhoods.shp", quiet = TRUE)

neighborhoods <-
raw_neighborhoods %>%
  left_join(neighborhoods_xwalk) %>%
  group_by(neighborhood_code) %>%
  combine_polygons() %>%
  filter(!is.na(neighborhood_code))

ggplot() +
  geom_sf(data = zones, aes(fill = zone, color = zone), lwd = .3, alpha = .3, show.legend = FALSE)  +
  geom_sf(data=neighborhoods, lwd = .1, fill=NA) +
  theme_void() +
  labs(title = "Using your zone shape file + Boston open data")
```

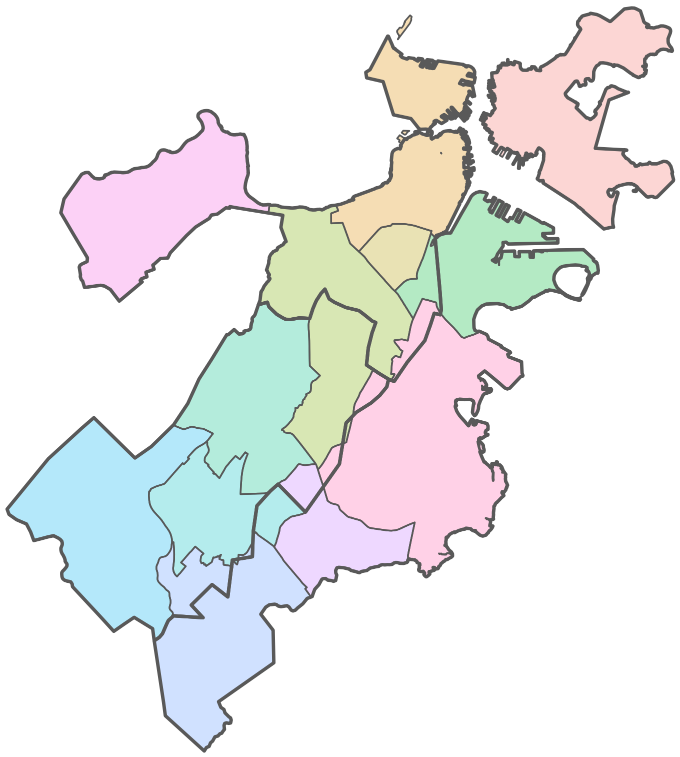
Using your zone shape file + Boston open data



We can also foreground the neighborhood

```
ggplot() +
  geom_sf(data = neighborhoods, aes(fill = as.factor(neighborhood_code)), lwd = .3, alpha = .3, show.leg
  geom_sf(data = zones, lwd = .6, fill=NA) +
  theme_void() +
  labs(title = "Using your zone shape file + Boston open data (alt)")
```

Using your zone shape file + Boston open data (alt)

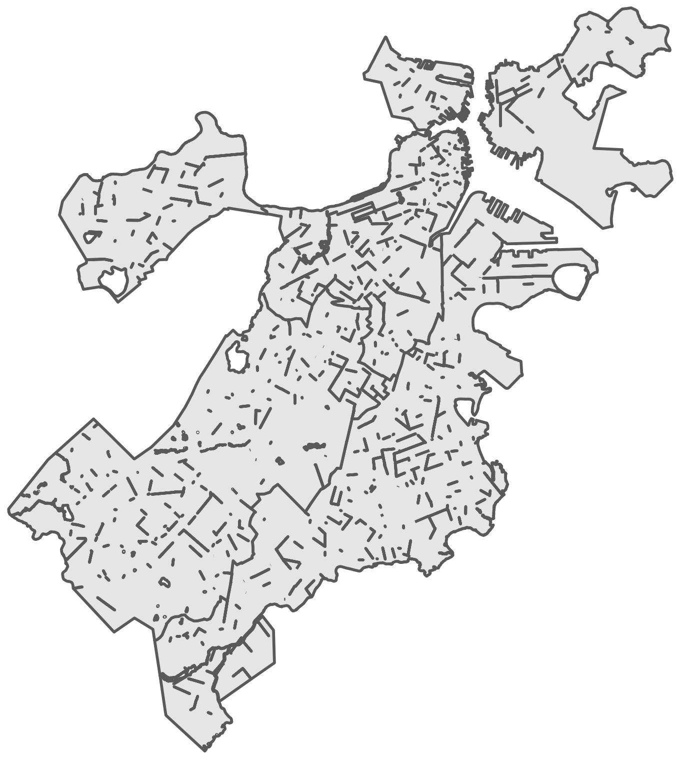


Why are there random lines?

The geocodes shapefile has many extraneous lines, which makes it difficult to collapse into polygons.

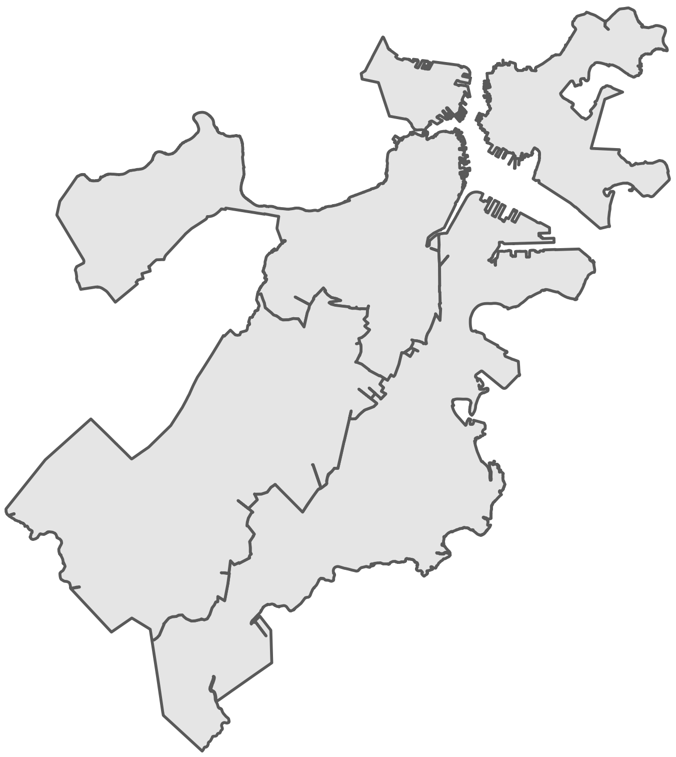
```r
my_plot <- function(data) data %>% ggplot() + geom_sf() + theme_void()

geocodes  %>%
  group_by(zones) %>%
  combine_polygons() %>%
  my_plot() +
  labs(title = "Simply combining leaves a lot of issues")
```

Simply combining leaves a lot of issues



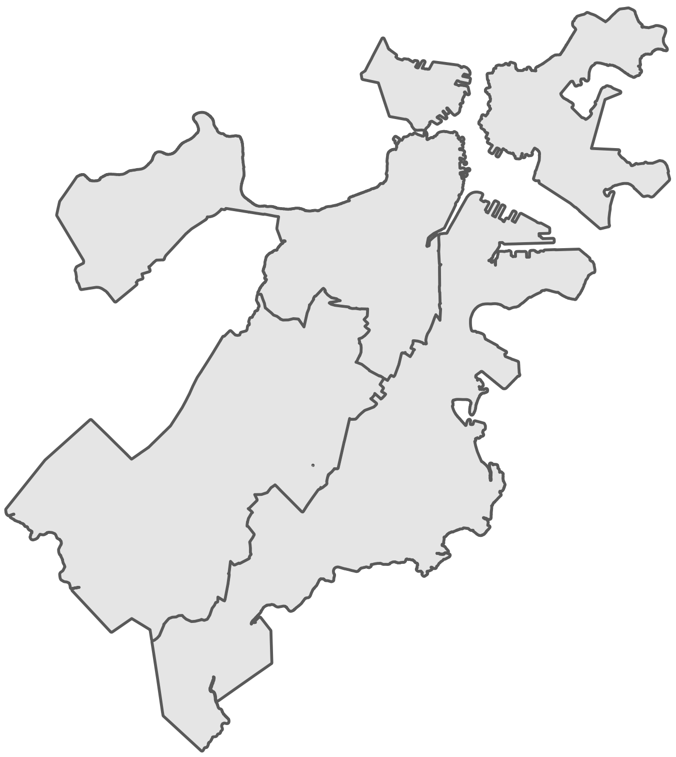
```r
geocodes  %>%
  group_by(zones) %>%
  combine_polygons() %>%
  sf_remove_holes() %>%
  my_plot() +
  labs(title = "Removing holes gets rid of many of the small lines",
       subitle = "But we still have lines")
```

Removing holes gets rid of many of the small lines



```
geocodes  %>%
  group_by(zones) %>%
  combine_polygons() %>%
  sf_remove_holes() %>%
  st_buffer(-100) %>%
  st_buffer(110) %>%
  my_plot() +

  labs(title = "Buffering almost does the job")
```

Buffering almost does the job



```
geocodes  %>%
  group_by(zones) %>%
  combine_polygons() %>%
  sf_remove_holes() %>%
  st_buffer(-200) %>%
  st_buffer(220) %>%
  sf_remove_holes() %>%
  my_plot() +
  labs(title = "But buffering too much makes things worse",
       subtitle = "(We lose some corners")
```

But buffering too much makes things worse

(We lose some corners