London: Generating dot density coordinates for ethnic census data

Nicole Dwenger and Orla Mallon

02/05/2021

This script generates dots and their associated coordinates for London's dot density map of ethnicity.

The data is taken from the most recent UK census, 2011, and the ethnicities are grouped into the following 6 categories - as denoted in the census: (Arab, Asian, Black, Mixed, White, and Other)

Density ratio: 1 dot = 100 people

1. Preparation

1.1. Packages

```r
# Loading up the packages
library(maptools)
library(rgeos)
library(tidyverse)
library(rgdal)
library(ggthemes)
library(sf)
library(dplyr)
```

1.2 Load and clean the data

The data used is taken from the UK 2011 Census

```r
# --- Load in the census data ---
ethnic_data <-  read.csv("london/raw_data/ethnicity/WD201EWDATA.CSV")

# --- Rename the columns to be more intuitive ---
ethnic_data <- ethnic_data %>% rename(
                            output_area_code = GeographyCode,
                            All_Ethnicities = WD201EW0001,
                            White_British = WD201EW0002,
                            White_Irish = WD201EW0003,
                            White_Gypsy = WD201EW0004,
                            White_Other = WD201EW0005,
                            Mixed_Caribbean = WD201EW0006,
                            Mixed_African = WD201EW0007,
                            Mixed_Asian = WD201EW0008,
                            Mixed_Other = WD201EW0009,
                            Asian_Indian = WD201EW0010,
                            Asian_Pakistani = WD201EW0011,
                            Asian_Bangladeshi = WD201EW0012,
                            Asian_Chinese = WD201EW0013,
                            Asian_Other = WD201EW0014,
                            Black_African = WD201EW0015,
                            Black_Caribbean = WD201EW0016,
                            Black_Other = WD201EW0017,
                            Arab = WD201EW0018,
                            Other = WD201EW0019)
```

1.3 Group the ethnicities

Alphabetic ordering used to list ethnicites with 'Other' at the end

```
# --- Create columns of ethnicity grouped ---
ethnic_data <- ethnic_data %>%
  mutate(White = rowSums(ethnic_data[3:6])) %>%
  mutate(Mixed = rowSums(ethnic_data[7:10])) %>%
  mutate(Asian = rowSums(ethnic_data[11:15])) %>%
  mutate(Black = rowSums(ethnic_data[16:18]))

# --- Reduce ethnities dtaaframe to include only the 6 categories (Arab, Asian, Black, Mixed, White, Ot
ethnic_data <- ethnic_data %>%
  select(output_area_code, Arab, Asian, Black, Mixed, White, Other, All_Ethnicities) %>%
  group_by(output_area_code) %>%
  summarise(Arab = sum(Arab),  Asian = sum(Asian), Black = sum(Black), Mixed = sum(Mixed),
            White = sum(White), Other = sum(Other), OA_Population = sum(All_Ethnicities))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# ---Cut down to only London output areas ---
# File to match Output area to borough: London Output Area Classification file (LOAC)
borough_matching <- read.csv("london/raw_data/ethnicity/LOAC classification.csv", as.is = 1)

#Cut the ethnic data down to only include the london output areas (~25,053)
ethnic_data <- ethnic_data[ethnic_data$output_area_code %in% borough_matching$OA, ]
```

1.4 Match the output areas to their Borough

```
# --- Reduce borough_matching to only have the columns of name and geometry ---
borough_matching <- borough_matching %>% dplyr::select(OA, Local.Authority)

# --- Merge the ethnic data file with the boroughs ---
ethnic_data <- merge(ethnic_data, borough_matching, by.x="output_area_code", by.y="OA", sort = FALSE)

# --- Rename Local Authority to 'name' ---
ethnic_data <- ethnic_data %>% rename(name = Local.Authority)

# --- Create a column of the total population per borough ---
ethnic_data <- ethnic_data %>% group_by(name) %>% mutate(Borough_population = sum(OA_Population, na.rm
```

1.5 Calculate the ethnicity percentages per borough

```
# --- Create a smaller dataframe with the percentage counts of each ethnicity ---
ethnic_data_percentages <- ethnic_data %>%
  group_by(name) %>%
  summarise(ethnicity_perc_arab = sum(Arab/Borough_population), ethnicity_perc_asian = sum(Asian/ Boroug
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# --- Append this onto the main ethnic data frame ---
ethnic_data <- merge(ethnic_data, ethnic_data_percentages, by = "name", sort = FALSE)
```

1.6 Load in the London output area shapefiles

Shapefiles of smaller geographical areas, necessary for generating the dots coordinates.

```
# --- Load in the output areas and rename column ---
OA_shapefile<- st_read("london/raw_data/ethnicity/OA_2011_London_gen_MHW.shp") %>% select(output_area_co
```

```
## Reading layer `OA_2011_London_gen_MHW' from data source `/Users/Orla/Desktop/Shiny/london/raw_data/et
## Simple feature collection with 25053 features and 17 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 503574.2 ymin: 155850.8 xmax: 561956.7 ymax: 200933.6
## CRS:            27700
```

```
# --- Add the geometry column to the ethnic_data by joining the datasets ---
ethnic_data <- left_join(ethnic_data, OA_shapefile) %>% st_as_sf()
```

```
## Joining, by = "output_area_code"
```

2. Generating the Dots and their Coordinates

2.1 Defining functions

randomround = A function which helps to soften the threshold of 100 people = 1 dot. This function randomly rounds the data so that if a group has over 50 people, but is not quite 100, sometimes it will be generated as a dot as the number will be randomly rounded up instead of down. This is preferred over a simple rounding or hard threshold, which can often lead to systematic bias against minority groups.

credit: Jens von Bergmann https://github.com/mountainMath/dotdensity/blob/master/R/dot-density.R

count_dots = A function which generates how many dots should be created for each ethnicity, in each output area. The output is dataframe with a column for each ethnicity and a row for each output area, holding the number of dots which should be plotted for each.

generate_dots = A function which generates random coordinates for the dot, within the specified geographical area. Here, we use the smallest area shapefiles available, meaning that even though the coordinates are randomly generated, they should reflect the ethnicity of the area to an accurate degree.

```r
# --- randomround ---
#Function for rounding data to soften density threshold
random_round <- function(x) {
    v=as.integer(x)
    r=x-v
    test=runif(length(r), 0.0, 1.0)
    add=rep(as.integer(0),length(r))
    add[r>test] <- as.integer(1)
    value=v+add
    ifelse(is.na(value) | value<0,0,value)
    return(value)
}

# --- count_dots ---
# A function which takes the borough sf object and counts how many dots should be created
count_dots <- function(x) {
  district_dots = as.data.frame(x) %>%
    select(Arab:Other) %>%
    mutate_all(funs(. / 100)) %>%
    mutate_all(random_round)
    return(district_dots)
  }

# --- generate_dots ---
# A function which determines the spatial coordinates of the dots
generate_dots <- function(district_dots, sf_object) {
  district_coords <- map_df(names(district_dots),
                           ~ st_sample(sf_object, size = district_dots[,.x], type = "random") %>%
                             st_cast("POINT") %>%
                             st_coordinates() %>%
                             st_transform(4326) %>%
                             as_tibble() %>%
                             setNames(c("lon","lat")) %>%
                             mutate(ethnicity = .x)) %>%
                    slice(sample(1:n()))
                    return(district_coords)
}
```

2.2 Generating the dot coordinates

```r
# --- Create empty df to append into ---
london_df = data.frame()

# --- Loop to run through each borough and genrate the dot coordinates ---
for(Boroughname in unique(ethnic_data$name)){
  Borough_sf = filter(ethnic_data, name == Boroughname)
  Borough_dots = count_dots(Borough_sf)
  Borough_coords = generate_dots(Borough_dots, Borough_sf)
  Borough_coords$name = as.character(Boroughname)
  london_df = rbind(london_df, Borough_coords)
}
```

2.3 Transform the coordinates into the 4362 CRS

```
# --- Make dataframe into an sf object (by combining lon and lat into a geometry column) ---
london_df <- st_as_sf(london_df, coords = c("lon", "lat"), crs= 27700)

# --- Transform into the 4326 crs for leaflet ---
london_df <- st_transform(london_test, crs = 4326)
```

3. Save the data

```
# --- Save the spatial object as an RDS ---
saveRDS(london_df, file = "london/preprocessed_data/london_dot_coortinates.RDS")

# --- Save the Ethnic dataframe with percentages of each ethnicity per borough ---
write.csv(ethnic_data_percentages, "london/preprocessed_data/london_percentages.csv", row.names = F)

# --- Save a csv file of the pre-transformed coordinates ---
ethnic_csv <- st_drop_geometry(ethnic_data)
write.csv(ethnic_csv, "london/preprocessed_data/london_ethnicity.csv", row.names = F)
```