Visualizing the Number of Polluting Facilities in Toronto Neighborhoods

Synopsis

The goal of this exercise is to visualize environmental data in the city of Toronto by generating a visual representation of the city and it's various neighborhoods.

This analysis makes use of data from Toronto's Open Data Catalogue for both Geographical Polygons and Environmental Data.

Data Preparation

The first step is to load the packages which will be used for this analysis.

```
# Install package if it is not already installed. Load package.

usePackage <- function(p) {
    if (!is.element(p, installed.packages()[,1]))
        install.packages(p, dep = TRUE)
    library(p, character.only = TRUE)
}

usePackage("rgdal")      # to allow for use with Geospatial Data Abstraction Library (GDAL).
usePackage("spatstat")   # for spatial analysis
usePackage("ggplot2")    # plotting functionality
usePackage("stringr")    # useful for working with strings.
usePackage("maptools")   # will be used in joining datasets
usePackage("xlsx")       # to work with excel files efficiently
usePackage("dplyr")      # work with dataframes
```

Retrieve the data from Toronto's Open Data database. We will download the .zip file containing the ESRI .shp file we are interested in. This is what will allow us to visualize the neighborhood boundaries in the city of Toronto.

The **Neighbourhoods** dataset contains the city's neighborhood boundaries data may be downloaded from the following link: Neighbourhoods

This analysis assumes that the initial .zip file is contained within your working directory.

```
unzip("neighbourhoods_planning_areas_wgs84.zip") # unzip the files
```

Working with the Shape File.

First, let's take a look at some basic information about the .shp file we have obtained.

```
shp_info <- ogrInfo(".", "NEIGHBORHOODS_WGS84")
```

The information above let's us know that the file contains 140 with boundaries contained within the following lat/lon coordinates -79.6392649, 43.580996, -79.1152432, 43.8554572.

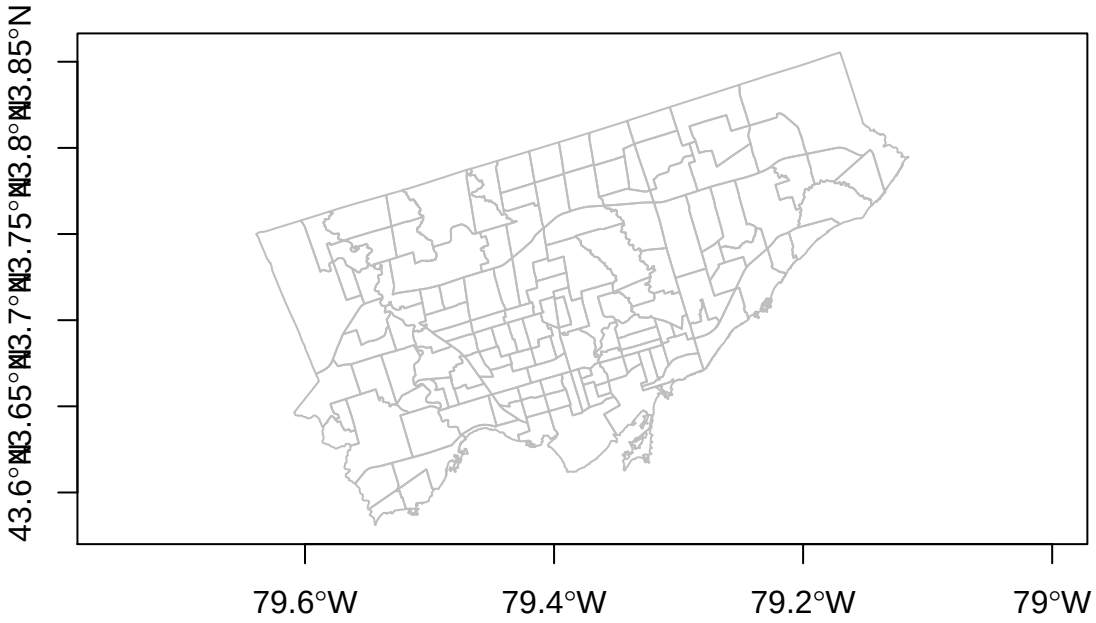
```
neighborhoods.rg <- readOGR(".", "NEIGHBORHOODS_WGS84")
```

Read in the .shp file

```
## OGR data source with driver: ESRI Shapefile
## Source: ".", layer: "NEIGHBORHOODS_WGS84"
## with 140 features
## It has 2 fields
```

Plotting the polygon Let's take a look at the basic structure of Toronto's neighborhoods which we will use to display our data. For this pupose, we will simply use R's basic plotting mechanism.

```
plot(neighborhoods.rg, axes=TRUE, border="gray")
```



Load Environmental Data Now that we have polygons to work with, let's start to fill it in with our data.

This data has also been taken from Open Data Toronto's great Data Catalogue. Specifically the **Wellbeing Toronto - Environment** dataset may be downloaded from the following link: Wellbeing Toronto - Environment

We will use the XLSX package to extract 2008 and 2011 data from the excel spreadsheet of environmental data.

```
# 2008 Data
environment_2008 <- read.xlsx("WB-Environment.xlsx", sheetName = "RawData-Ref Period 2008")

# transform id variable so that it agrees with the neighborhood data
names(environment_2008)[2] <- "id"
environment_2008$id <- as.factor(str_pad(environment_2008$id, 3, pad = "0"))

# 2011 Data
environment_2011 <- read.xlsx("WB-Environment.xlsx", sheetName = "RawData-Ref Period 2011")

# transform id variable so that it agrees with the neighborhood data

names(environment_2011)[2] <- "id"
environment_2011$id <- as.factor(str_pad(environment_2011$id, 3, pad = "0"))
```

```
# change row names to the id variable
row.names(environment_2008) <- environment_2008$id
row.names(environment_2011) <- environment_2011$id

# do the same for .shp
row.names(neighborhoods.rg) <- as.character(neighborhoods.rg$AREA_S_CD)

# We need to reorder the rownames for merging
neighborhoods.rg <- neighborhoods.rg[order(neighborhoods.rg$AREA_S_CD), ]

# binding the datasets together.
ds2008 <- spCbind(neighborhoods.rg, environment_2008)
ds2011 <- spCbind(neighborhoods.rg, environment_2011)
head(neighborhoods.rg@data)
```

Joining the datasets.

```
##     AREA_S_CD                              AREA_NAME
## 001       001            West Humber-Clairville (1)
## 002       002 Mount Olive-Silverstone-Jamestown (2)
## 003       003      Thistletown-Beaumond Heights (3)
## 004       004                   Rexdale-Kipling (4)
## 005       005                  Elms-Old Rexdale (5)
## 006       006     Kingsview Village-The Westway (6)
```

Transform the data The fortify function transforms the object into a dataframe that is useful for plotting.

```
fort <- fortify(neighborhoods.rg) # define region for each polygon.
```

```
## Regions defined for each Polygons
```

Visualizing The Data

Now the data is ready for visualization. To do this, we can use the ggplot2 package.

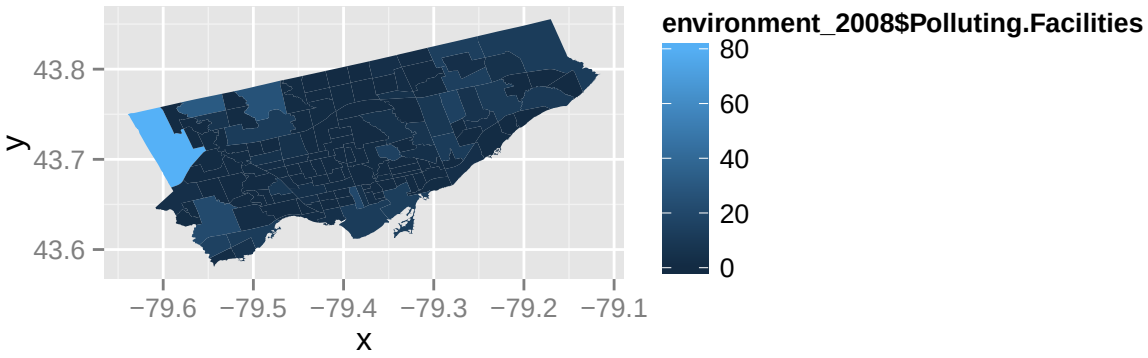
```
m <- ggplot(environment_2008)

m <- m + geom_map(aes(fill = environment_2008$Polluting.Facilities,
                      map_id = id),
                      map = fort)

# ensure that map limits are condusive
m <- m + expand_limits(x = fort$long, y = fort$lat)

# equal scale cartesian coordinates.
m <- m + coord_equal()

m
```



At this point, we have a good representation of our data city of Toronto. To complete the visualization, let's do some aesthetic tweaks to clean up the clutter and aid interpretation through appropriate labels and colours.

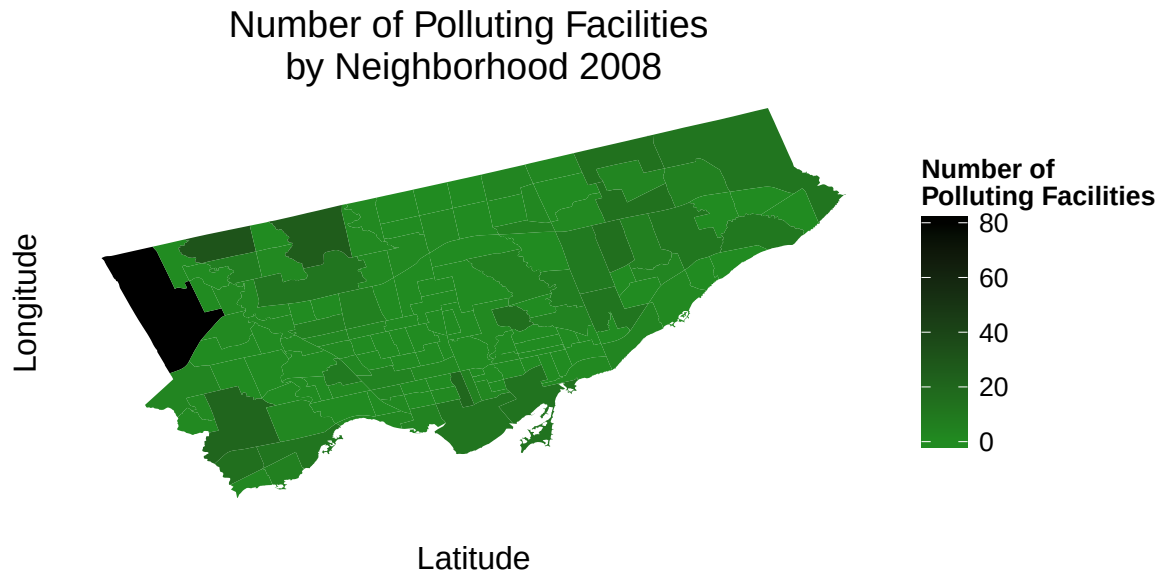
```
# set colour gradient
m <- m + scale_fill_gradient (name ="Number of \nPolluting Facilities",
                              low ="forestgreen",
                              high = "black")

# input title and axis labels
m <- m + xlab( "Latitude") + ylab("Longitude") + ggtitle("Number of Polluting Facilities \nby Neighborh
```

```
# remove unecessary background features.
bGrid <- theme(panel.grid = element_blank())
bBack <-theme(panel.background = element_blank())
bTics <- theme(axis.text = element_blank(),
               axis.text.y = element_blank(),
               axis.ticks = element_blank())

m <- m + bGrid + bBack + bTics

m
```



Each neighborhood within the city of Toronto is now coloured according to the number of Polluting Facilities within the neighborhood boundaries(as of 2008). This analysis could be easily reproduced to visualize any other variables according to neighborhood as well.

To generate a png image from our visualization, we may use the following code.

```
png("./torontoPollutingFacilities.png", width=750,
    height=750)
plot(m)
dev.off()
```

```
## pdf
##   2
```