Book Chapter: 'Crowdsourcing'

David Buil-Gil and Reka Solymosi

18/03/2020

Introduction/background:

What is crowdsourcing?

What types of data are commonly crowdsourced?

What are the strengths of crowdsourced data?

What are some weaknesses of crowdsourced data?

Step-by-step example in R:

Crowdsourcing public perceptions of safety

The Place Pulse project

Download and explore Place Pulse data

We have saved some Place Pulse data in a data repository on FigShare. You can download this directly into R by using the `read.csv()` function. It is a large file so it may take some time to read in, be patient!

```r
pp_data <- read.csv('https://ndownloader.figshare.com/files/21739137')
```

This data set includes **DESCRIPTION OF DATA SET HERE**

Some descriptive exploration of PP data.

Cleaning Place Pulse data

When it comes to crowdsourced data, you will have to be an expert data wrangler to make sure you can get the data to behave like you want it to - in other words to make the data available in a format that allows you to answer your research questions. For example, in this case, we want to map the perceives safety of areas in Atlanta. To do this, first we have to select the area of Atlanta, and the votes about safety:

Let's focus on the city of Atlanta. SOME CONTEXT.

```r
pp_atl <- pp_data[which(pp_data$place_name_right == "Atlanta" | pp_data$place_name_left == "Atlanta"), ]
```

Let's also focus on the ratings of areas as safer, as we are interested in people's perceptions of the environment:

```r
pp_atl_s <- pp_atl[ which(pp_atl$study_question == "safer"), ]
```

You can see now we have a dataframe of 37214 votes about the safety of places in Atlanta. But we need not only one, but both of the locations to be in Atlanta (EXPLAIN)

BREAK THIS DOWN AND EXPLAIN CLEARLY AND MOTIVATE

```
table(pp_atl_s$place_name_right == "Atlanta" & pp_atl_s$place_name_left == "Atlanta") #count votes in w
```

```
##
## FALSE  TRUE
## 36536   678
```

```
pp_atl_s <- pp_atl_s[order(pp_atl_s$X), ] #order file by vote number

pp_atl_s_dup <- pp_atl_s[which(pp_atl_s$place_name_right == "Atlanta" & pp_atl_s$place_name_left == "Atl

pp_atl_s$long_Atl[pp_atl_s$place_name_left == "Atlanta" & pp_atl_s$place_name_right != "Atlanta"] <- pp
pp_atl_s$lat_Atl[pp_atl_s$place_name_left == "Atlanta" & pp_atl_s$place_name_right != "Atlanta"] <- pp_

pp_atl_s$long_Atl[pp_atl_s$place_name_right == "Atlanta" & pp_atl_s$place_name_left != "Atlanta"] <- pp
pp_atl_s$lat_Atl[pp_atl_s$place_name_right == "Atlanta" & pp_atl_s$place_name_left != "Atlanta"] <- pp_

pp_atl_s$long_Atl[pp_atl_s$place_name_right == "Atlanta" & pp_atl_s$place_name_left == "Atlanta"] <- pp
pp_atl_s$lat_Atl[pp_atl_s$place_name_right == "Atlanta" & pp_atl_s$place_name_left == "Atlanta"] <- pp_

pp_atl_s_dup$long_Atl <- pp_atl_s_dup$long_right #allocate coordinates from right image to votes in whi
pp_atl_s_dup$lat_Atl <- pp_atl_s_dup$lat_right

pp_atl_s <- rbind(pp_atl_s, pp_atl_s_dup) #merge duplicated cases where both images are from Atlanta

pp_atl_s$win[pp_atl_s$long_right == pp_atl_s$long_Atl & pp_atl_s$choice == "right"] <- 1 #code 'safer'
pp_atl_s$win[pp_atl_s$long_left == pp_atl_s$long_Atl & pp_atl_s$choice == "left"] <- 1

pp_atl_s$win[pp_atl_s$long_right == pp_atl_s$long_Atl & pp_atl_s$choice == "left"] <- 0
pp_atl_s$win[pp_atl_s$long_right == pp_atl_s$long_Atl & pp_atl_s$choice == "equal"] <- 0

pp_atl_s$win[pp_atl_s$long_left == pp_atl_s$long_Atl & pp_atl_s$choice == "right"] <- 0
pp_atl_s$win[pp_atl_s$long_left == pp_atl_s$long_Atl & pp_atl_s$choice == "equal"] <- 0

table(pp_atl_s$win) #count frequency of 'safer' votes in Atlanta against 'equal' and 'less safe' votes
```

```
##
##     0     1
## 19143 18749
```

```
prop.table(table(pp_atl_s$win))*100 #count % of 'safer' votes in Atlanta against 'equal' and 'less safe
```

```
##
##       0       1
## 50.5199 49.4801
```

Map Place Pulse data

It is possible to use mapping techniques learned in other chapters (LINK WITH MAPPING CHAPTER PEOPLE) to map the crowdsourced data.

Let's plot this, to create a map of perceived safety of built environment across the city of Atlanta. For this, we will be using the `sf` and `ggplot2` libraries.

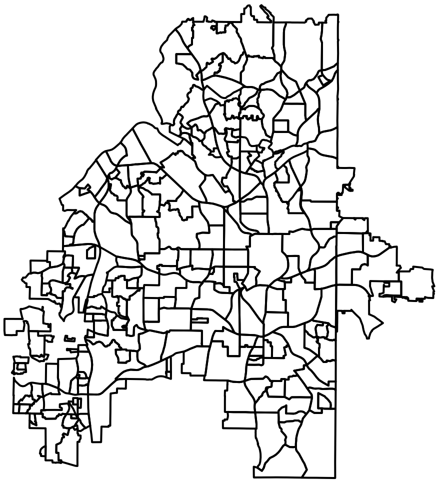
First, acquire a shapefile for Atlanta. Let's use the neighbourhoods shapefile from the department of city planning. You can go on their website to find out more about this boundary data: https://dcp-coaplangis.opendata.arcgis.com/datasets/neighborhoods. We can download the shapefile directly using their Application Programme Interfact (or API) - this is something discussed in greater detail on the chapter on Open Data (CHAPTER REF). For now, you can just use the code below, with the `st_read()` function in the `sf` package:

```
atl <- st_read("https://opendata.arcgis.com/datasets/297d3d69d8ab4c6ba5f9264ad5e75c0a_3.geojson")
```

```
## Reading layer `OGRGeoJSON' from data source `https://opendata.arcgis.com/datasets/297d3d69d8ab4c6ba5
## Simple feature collection with 244 features and 18 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -84.55085 ymin: 33.64799 xmax: -84.28962 ymax: 33.88687
## epsg (SRID):    4326
## proj4string:    +proj=longlat +datum=WGS84 +no_defs
```

We can see what this looks like by using the `plot()` function to plot the geometry of the atl object we created, called with the `st_geometry()` function:

```
plot(st_geometry(atl))
```



Now to be able to plot the safety votes on this map, we first need to make our votes a spatial object, by specfying that the “long_Atl” and “lat_Atl” contain our longitude and latitude information. We use the `st_as_sf()` function for this:

```r
points_atl_s <- st_as_sf(pp_atl_s, coords = c("lat_Atl", "long_Atl")) #geocode 'safe' votes in Atlanta
```

Now to be able to plot both these spatial layers on the same map, their coordinate reference systems need to match. We can check these with the `st_crs()` function:

```r
st_crs(points_atl_s) == st_crs(atl) #check if coordinate reference system is the same of both layers
```

```
## [1] FALSE
```

We see they do not! We can fix this by setting the CRS of our point data to that of the atl shapefile using the `st_crs()` function:

```r
st_crs(points_atl_s) <- st_crs(atl)
```

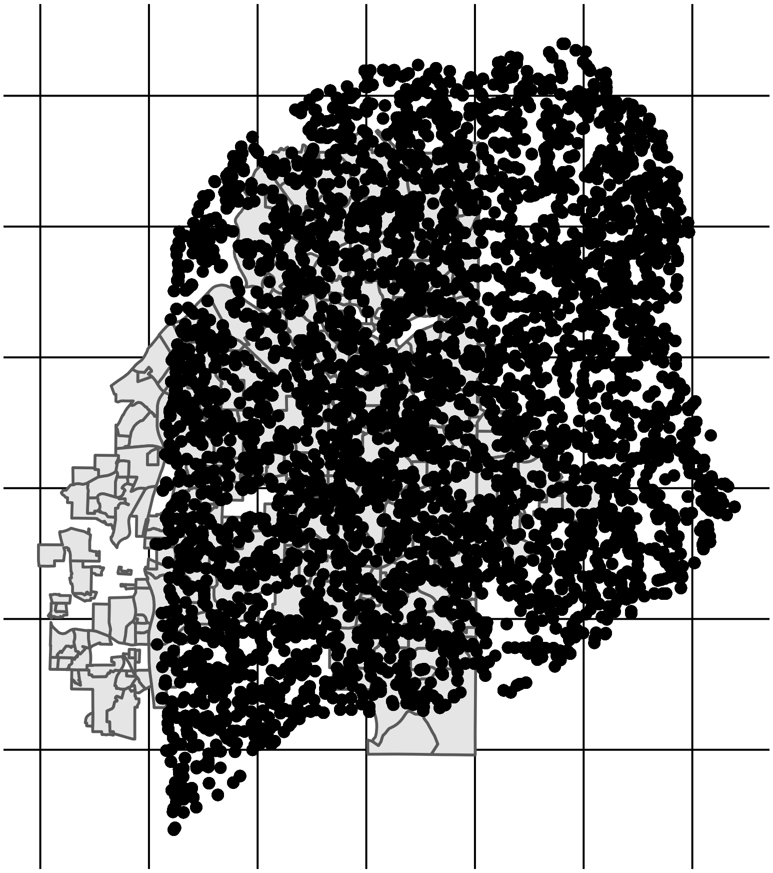
Now if we check, they should have the same CRS:

```r
st_crs(points_atl_s) == st_crs(atl) #check if coordinate reference system is the same of both layers
```

```
## [1] TRUE
```

Indeed! Now we can map our data!

```r
ggplot(data = atl) +
  geom_sf() +
  theme_void() +
  geom_sf(data = points_atl_s) #visualise points on the map
```



This is a very busy map! Maybe instead we want to get some sort of average score for each neighbourhood.

Get win score for each nhood

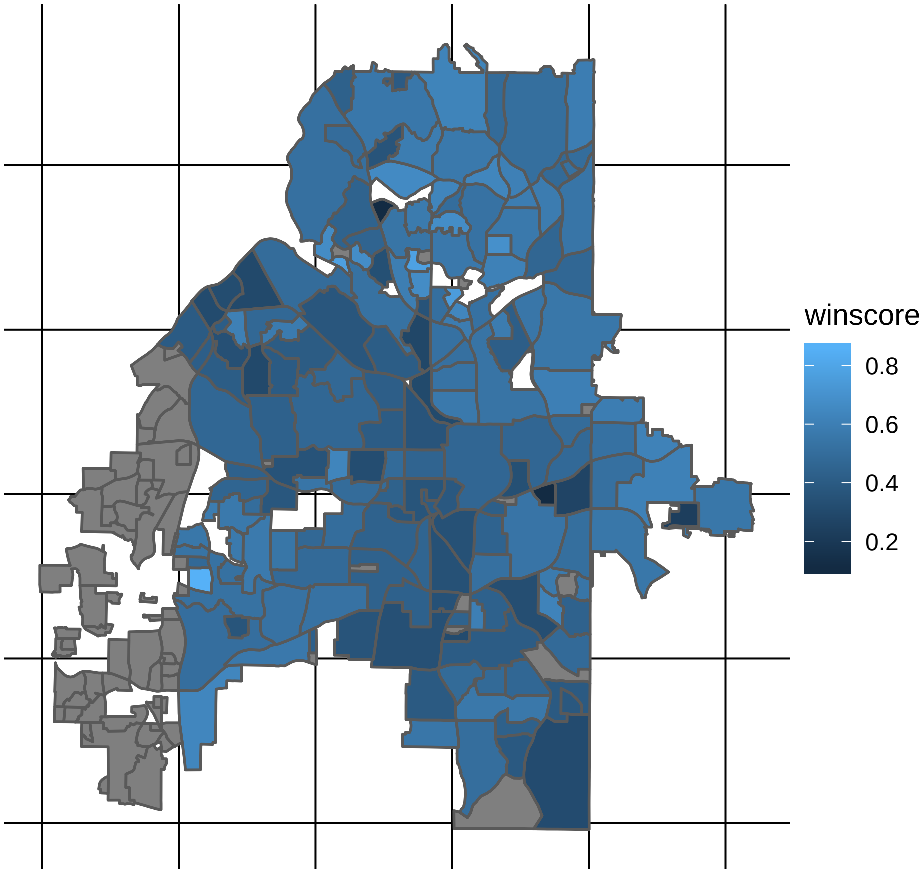
```
points_atl_s_nhood <- st_intersection(atl, points_atl_s) %>%
  group_by(NAME) %>%
  summarise(winscore = mean(win, na.rm = TRUE),
            num_votes = n())
```

Join to shapefile

```
st_geometry(points_atl_s_nhood) <- NULL

atl_pp_wins <- left_join(atl, points_atl_s_nhood, by = c("NAME" = "NAME"))
```

```
ggplot(data = atl_pp_wins) +
  geom_sf(aes(fill = winscore)) +
  theme_void()
```



About this: Note: Salesses et al. (2013) suggest computing a Q-score corrected by the "win" and "loss" ratio of imagines with which each vote is compared for the purpose of this chapter we will compute a simple proportion - should we include this for an activity?

There are many more things you can do with these data. for example you could look at the descriptive statistics:

```
summary(atl_pp_wins$winscore) #descriptive statistics of proportion of 'safer' votes per tract
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  0.1111  0.4084  0.4929  0.4867  0.5604  0.8571      67
```

Important also to consider variation in the sample size of number of votes in each neighbourhood:

```
summary(atl_pp_wins$num_votes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##     2.0    47.0    81.0   106.1   139.0   569.0      67
```

Talk a little about the issues with this and then discuss this: Note: Buil-Gil et al. (2020) propose a new method to compute estimates for areas with small sample sizes

You can do much more, but here we will focus on the specific issues to explore due to the crowdsourced nature of these data!

Exploring the known issues of crowdsourced data within Place Pulse

We mentioned a few issues that are usually present in crowdsourced data, and which are important to keep in mind. Here we explore whether they are present in Place Pulse, and what that might mean for any conclusions we draw from our analyses.

First, create a new dataframe showing the number of votes that each study participant had made. To do this, we will use code from the `dplyr` library:

```
voter <- pp_data %>%
  group_by(voter_uniqueid) %>%
  summarise(num_votes = n())
```

If you want, you can have a look at this new dataframe using the `View()` function. If you do this, you might see, we have some very active participants! The top voter there has made 7168 votes on places! That is some very prolific participation! On the other hand, you can also see that 7494 of the participants made only one vote! We are definitely seeing signs of participation inequality in these data.

Participation inequality (supercontributors)

In fact, let's see, how much of the votes do the "supercontributors" account for? Let's say we want to consider the top 1% of voters. We can do this using the `subset()` and `quantile()` functions:

```
top_1percent <- subset(voter, num_votes > quantile(num_votes, prob = 1 - 1/100))
```

We can see that this new dataframe contains 954 people, who are our top 1% contributors to the Place Plus data set. Let's see how much of the total number of votes they contribute:

```
sum(top_1percent$num_votes) / sum(voter$num_votes) * 100
```

```
## [1] 17.87468
```

That is a lot! 17.87% of the votes are made by the top 1% of contributors!

Activity 1

Can you tell me what proportion of the votes were made by the top 10% of participants? What about the top 25%?

Quantifying participation inequality

One way to quantify the extent to which participation inequality exists in our data is by using a Gini index, and visualising using a Lorenz curve.

For this you will need the library `ineq` so if you don't already have this you must install with the command:

```
install.packages("ineq")
```

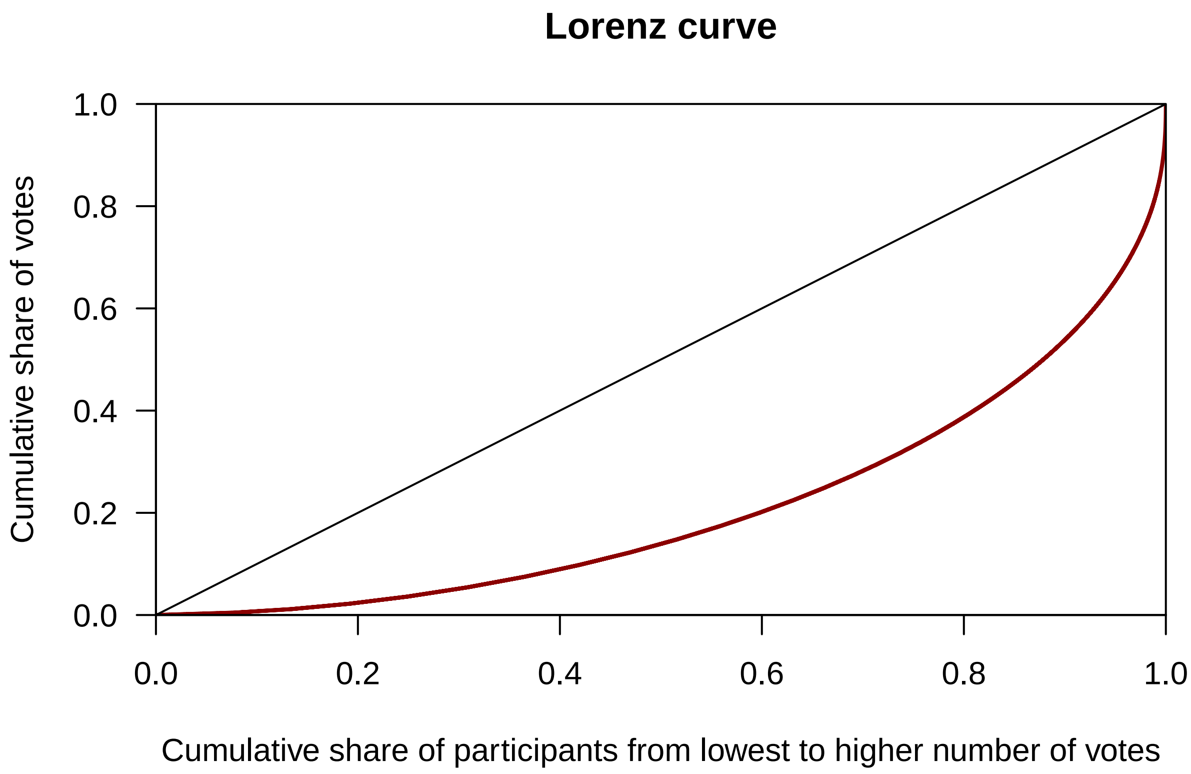
Then you can load this library and calculate the index using the `Gini()` function:

```
Gini(voter$num_votes)
```

```
## [1] 0.5777568
```

Remember that a score of 0 is perfect equality (everyone makes equal number of votes), while 1 is perfect inequality (only one person making all the reports, and no one else making any) . Our answer of 0.58 shows some serious inequality. To put this into context, in 2017, according to the OECD, income inequality in the United States of America showed a Gini coefficient of 0.39.
To visualise this we can use a Lorenz curve using the `plot()` and `Lc()` functions:

```
plot(Lc(voter$num_votes), xlab = "Cumulative share of participants from lowest to higher number of votes
     ylab = "Cumulative share of votes",col="darkred",lwd=2)
```



The Lorenz curve (red line) above shows how the top few percent of reporters contribute the majority of the reports. If we had perfect equality, we would expect to see the red line align perfectly with the black line with the slope of 1.

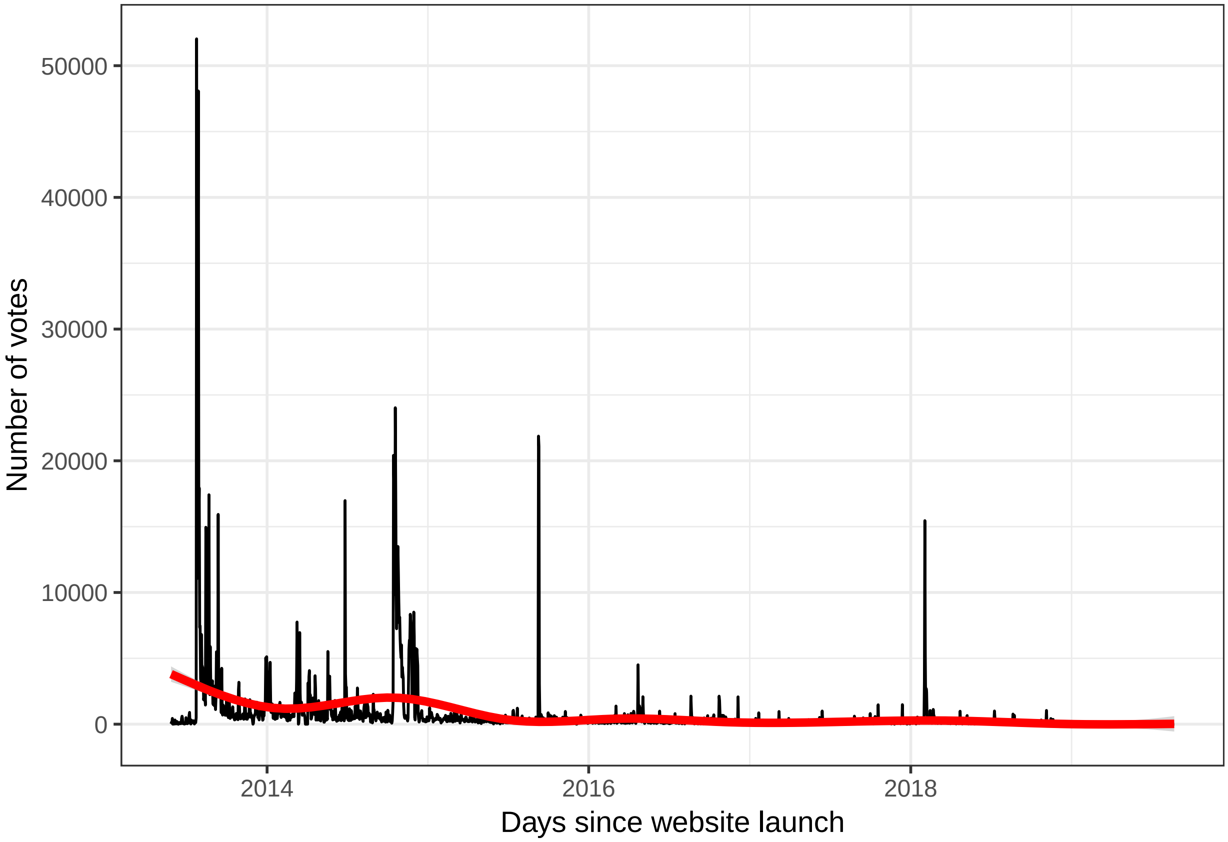
With this information we can now quantify how severe the participation inequality in our data, and compare with other crowdsourced data for context and understanding.

Study participation descrease in Place Pulse

!!!This should also be broken up and motivated/exaplained!!!

```r
by_day <- pp_data %>%
  mutate(day = ymd(day)) %>%
  group_by(day) %>%
  summarise(num_votes = n()) %>%
  complete(day = seq.Date(min(day), max(day), by="day")) %>%
  mutate(num_votes = replace_na(num_votes, 0))


ggplot(by_day, aes(x = day, y = num_votes)) +
  geom_line() +
  geom_smooth(lwd = 1.5, col = "red") +
  theme_bw() +
  xlab("Days since website launch") +
  ylab("Number of votes")
```



Note: large peak beginning on July 24th 2013: publication of Salesses et al. (2013) on July 24th and press release via MIT News (http://news.mit.edu/2013/quantifying-urban-perceptions-0724) Note: large peak beginning on October 15th 2014: publication of Harvey's (2014) MSc thesis

Wrap-up/Where to next:

Open topics in crowdsourced data

Further applications to crime and place research