Final Project Proposal

Priya Shaji

3/21/2020

NYC 311 complaint analysis

Data Source

- 311 Service Requests from 2010 to Present
- Socrata API
- NYC 311 Twitter

About NYC Open Data set

Beginning in 2010, NYC launched an initiative to expose government data via NYC Open Data in an effort to "improve the accessibility, transparency, and accountability of City government, this catalog offers access to a repository of government-produced, machine-readable data sets."

Motivation

NYC 311's mission is to provide the public with quick, easy access to all New York City government services and information while offering the best customer service. It help Agencies improve service delivery by allowing them to focus on their core missions and manage their workload efficiently.

NYC 311 data is updated on a daily basis and is provided by DoITT where currently I am pursuing my internship. Therefore, I wanted to apply visualization concepts studied in Data 608 to analyze this data set.

Aim

- To analyze and build visualizations for issues around New York City (including Manhattan, Queens, Brooklyn, and Bronx) by frequency of reported incidents in each area.
- NYC 311 Service Requests & Resolution Analysis through Text Mining
- Explore and analyze NYC 311 Service requests (historical data sets) to understand diverse patterns, regular themes and trends, as well as community satisfaction levels derived from resolution categories and timing.

Visualizations I plan to include are:

- Most common complain types by borough
- Resolutions for service requests
- Map plotting

Tweet sentiment analysis

I would also want to do sentiment analysis using `Syuzhet Package` on the NYC 311 twitter comments to determine "nyc311" Tweet's Emotions especially during the period of virus outbreak and also create visualization for same.

Import libraries

```r
library(plyr)
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts -------------------------------------------- tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
```

```r
library(knitr)
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
```

```
## The following object is masked from 'package:purrr':
##
##     flatten
```

Load the data using socrata API

```r
api_endpoint <- "https://data.cityofnewyork.us/resource/erm2-nwe9.json"
json_dataset311 <- fromJSON(paste0(api_endpoint))
class(json_dataset311)
```

```
## [1] "data.frame"
```

Display column Names

```r
colnames(json_dataset311)
```

```
##  [1] "unique_key"                      "created_date"
##  [3] "agency"                          "agency_name"
##  [5] "complaint_type"                  "descriptor"
##  [7] "location_type"                   "incident_zip"
##  [9] "incident_address"                "street_name"
## [11] "cross_street_1"                  "cross_street_2"
## [13] "intersection_street_1"           "intersection_street_2"
## [15] "city"                            "landmark"
## [17] "status"                          "community_board"
## [19] "bbl"                             "borough"
## [21] "x_coordinate_state_plane"        "y_coordinate_state_plane"
## [23] "open_data_channel_type"          "park_facility_name"
## [25] "park_borough"                    "latitude"
## [27] "longitude"                       "location"
## [29] ":@computed_region_efsh_h5xi"    ":@computed_region_f5dn_yrer"
## [31] ":@computed_region_yeji_bk3q"    ":@computed_region_92fq_4b7q"
## [33] ":@computed_region_sbqj_enih"    "address_type"
## [35] "resolution_description"          "resolution_action_updated_date"
## [37] "closed_date"                     "facility_type"
## [39] "taxi_pick_up_location"           "due_date"
```

```r
nrow(json_dataset311)
```

```
## [1] 1000
```

Technologies planned to use

- RStudio, R programming language.