Data_Exploration

Sean

1/17/2022

```
knitr::opts_chunk$set(echo = TRUE)
### Set Environment ------------------------------------------------------------
# Clear the environment
rm(list=ls())
gc()
```

```
##          used (Mb) gc trigger (Mb) limit (Mb) max used (Mb)
## Ncells 415802 22.3     854594 45.7         NA   666822 35.7
## Vcells 788846  6.1    8388608 64.0      16384  1824057 14.0
```

```
# So the code will compile warnings as per usual
options(warn = 0)
# Turn off scientific notation
options(scipen = 999)
### Load Packages --------------------------------------------------------------
if(!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
pacman::p_load(dplyr, magrittr, stringr, reshape2, janitor,
lubridate, readxl, data.table, ggplot2, scales, readr,
tidyr, zoo, skimr, openxlsx,ggspatial, rgeos, data.table,RColorBrewer,
tidyverse, rio, collapse, sf, glue, XML, tm, here, purrr, repurrrsive,
tmap,tidygraph, nabor,igraph, viridis, hrbrthemes,RSocrata,soql,xts)
```

Data Exploration of 311 Data

Part 1: Exploring different agencies and setting API up:

First Test of 311 data:

First step will be to set up our environment and begin querying using the socrata package. I will test 100,000 entries first.

```
dataset_id <- "erm2-nwe9"
api_endpoint <- paste0('https://data.cityofnewyork.us/resource/', dataset_id, '.json')
api_token <- "YRWk8RdZZ9xPMVbxc0QaHgYFU"

query_params <- soql() %>%
    soql_add_endpoint(api_endpoint) %>%
    soql_limit(100000)

query <- read.socrata(
  query_params,
```

```
  app_token = api_token
)
names(query)
```

```
##  [1] "unique_key"                     "created_date"
##  [3] "closed_date"                    "agency"
##  [5] "agency_name"                    "complaint_type"
##  [7] "descriptor"                     "location_type"
##  [9] "incident_zip"                   "incident_address"
## [11] "street_name"                    "cross_street_1"
## [13] "cross_street_2"                 "address_type"
## [15] "city"                           "facility_type"
## [17] "status"                         "resolution_description"
## [19] "resolution_action_updated_date" "community_board"
## [21] "bbl"                            "borough"
## [23] "x_coordinate_state_plane"       "y_coordinate_state_plane"
## [25] "open_data_channel_type"         "park_facility_name"
## [27] "park_borough"                   "latitude"
## [29] "longitude"                      "location.latitude"
## [31] "location.longitude"             "location.human_address"
## [33] "due_date"                       "intersection_street_1"
## [35] "intersection_street_2"          "taxi_pick_up_location"
## [37] "taxi_company_borough"           "vehicle_type"
## [39] "bridge_highway_name"            "bridge_highway_direction"
## [41] "road_ramp"                      "bridge_highway_segment"
## [43] "landmark"
```

```
agencies <- unique(query$agency)
agencies
```

```
##  [1] "DSNY"  "DOT"   "DPR"   "TLC"   "DOB"   "DFTA"  "DOF"   "NYPD"  "DOHMH"
## [10] "EDC"   "DOE"   "DHS"   "HPD"   "3-1-1" "DCA"   "HRA"   "DEP"   "NYCEM"
## [19] "DOITT"
```

Exploring different agencies:

Because there are many agencies, I will need to dive in to see what agencies there are. Agencies I'm interested in include Department of Environmental Protection, Emergency Management, Health and Mental Hygenie, 311, and buildings

```
descriptors <- function(department) {
    descriptors <- query %>%
        filter(agency == department) %>%
        pull(descriptor) %>%
        unique() %>%
        as.data.frame()
    return(descriptors)
}

agencies <- c("DOHMH","DEP","DOB","3-1-1","NYCEM")

DOHMH <- descriptors("DOHMH")
DEP <- descriptors("DEP")
DOB <- descriptors("DOB")
three11 <- descriptors("3-1-1")
```

```
NYCEM <- descriptors("NYCEM")

DOHMH
```

```
##                                     .
## 1                         Rat Sighting
## 2         Condition Attracting Rodents
## 3                                1 or 2
## 4                          Ventilation
## 5                        Mouse Sighting
## 6          Chemical Vapors/Gases/Odors
## 7                          Sewage Odor
## 8                        Letter Grading
## 9                 Other - Explain Below
## 10                     Puddle in Ground
## 11             Rodents/Insects/Garbage
## 12    Public Complaint - Comm Location
## 13                     Signs of Rodents
## 14 Failure to Post Calorie Information
## 15                         Animal Waste
## 16                                  N/A
## 17              Dust from Construction
## 18       Bare Hands in Contact w/ Food
## 19             Unleashed Dog in Public
## 20                   Smoking Violation
## 21                Food Worker Activity
## 22           Food Preparation Location
## 23                         Food Spoiled
## 24        Swimming Pool - Unmaintained
## 25                          Animal Odor
## 26                                  Cat
## 27                     Toilet Facility
## 28                        Pigeon Waste
## 29              Kitchen/Food Prep Area
## 30                              Rooster
## 31               Other (Explain Below)
## 32                   Food Contaminated
## 33       Food Contains Foreign Object
## 34                               Other
## 35                           3 or More
## 36          Permit/License/Certificate
## 37              Beekeeping - Honeybees
## 38               Inadequate or No Heat
## 39          Dry Cleaning Vapors (PERC)
## 40    Illness Caused by Drinking Water
## 41                         Sewage Leak
## 42                Dishwashing/Utensils
## 43         Bees/Wasps - Not a beekeper
## 44                      No Consent Form
## 45                          Pigeon Odor
## 46                          Handwashing
## 47                           Pet/Animal
## 48                Food Worker Hygiene
## 49                Food Worker Illness
```

```
## 50                             Sewage
## 51                        Farm Animal
## 52                               Odor
## 53                   Beach/Pool Water
## 54                No Permit or License
## 55          Turtle Under 4 inches Long
## 56                            Lighting
## 57                      Sewer or Drain
## 58    Dirty/Inadequate Equip./Facility
## 59                     Food Temperature
## 60           Container - Over 5 Gallons
## 61                   Puddle on Sidewalk
## 62           Fountain - Under 5 Gallons
## 63                 Facility Maintenance
## 64                      Food Protection
## 65                Minor Received Tattoo
## 66                  Swimming Pool Cover
## 67         Beach/Pool/Sauna Unpermitted
## 68                       Tenant Refusal
## 69                      Flower Planters
## 70                               Snake
## 71                   Puddle on Driveway
## 72            Fountain - Over 5 Gallons
## 73                      Puddle on Roof
## 74                                 Dog
## 75            Staff/Supervision/Permits
## 76                           Bird Bath
## 77          Container - Under 5 Gallons
## 78                   Contamination Risk
## 79                            Basement
## 80                            Pesticide
## 81         Workplace - 10 or Less Staff
```

```
DEP
```

```
##                                                    .
## 1                    Noise: Private Carting Noise (NQ1)
## 2                              Noise, Barking Dog (NR5)
## 3                    Noise: Construction Equipment (NC1)
## 4                              Leak (Use Comments) (WA2)
## 5                           Fire Hydrant Emergency (FHE)
## 6                           Chemical Spill/Release (HA1)
## 7                     Air: Odor/Fumes, Restaurant (AD2)
## 8          Noise: Construction Before/After Hours (NM1)
## 9                            Noise: Jack Hammering (NC2)
## 10     Noise: air condition/ventilation equipment (NV1)
## 11                Air: Odor/Fumes, Vehicle Idling (AD3)
## 12                    Wastewater Into Catch Basin (IEB)
## 13                    Horn Honking Sign Requested (NR9)
## 14                                 Noise: Alarms (NR3)
## 15                           Hydrant Running Full (WA4)
## 16  Hydrant Locking Device Request (Use Comments) (WC5)
## 17                                 Hydrant Leaking (WC1)
## 18                               Hydrant Defective (WC2)
## 19     Catch Basin Clogged/Flooding (Use Comments) (SC)
```

```
## 20              Catch Basin Sunken/Damaged/Raised (SC1)
## 21                   Defective/Missing Curb Piece (SC4)
## 22                  Culvert Blocked/Needs Cleaning (SE)
## 23                     Sewer Backup (Use Comments) (SA)
## 24                                     Dirty Water (WE)
## 25 Water Meter Stolen/Missing - Private Residence (CLR)
## 26                          Noise, Ice Cream Truck (NR4)
## 27 Water Meter Broken/Leaking - Private Residence (CMR)
## 28                           No Water/Low Pressure (WA5)
## 29                                   Street Flooding (SJ)
## 30                                 Hydrant Running (WC3)
## 31                   Plants- Odor Related Problems (PO1)
## 32                 Manhole Overflow (Use Comments) (SA1)
## 33                               Asbestos Complaint (B1)
## 34                     Oil Spill On Street, Large (HQL)
## 35                Manhole Cover Broken/Making Noise (SB)
## 36                     Unsafe Chemical, Abandoned (HC2)
## 37            Oil Spill Into Basin/Sewer - Small (IABS)
## 38                    Hydrant Knocked Over/Missing (WC)
## 39                  Lead Kit Request (Residential) (L10)
## 40                                  Chemical Odor (HD1)
## 41                                     Sewer Odor (SA2)
## 42               Manhole Cover Missing (Emergency) (SA3)
## 43                                  Chemical Spill (IAC)
## 44              Water Meter Broken/Leaking - Other (CMO)
## 45                    Noise:  lawn care equipment (NCL)
## 46              Air: Dust, Construction/Demolition (AE4)
## 47                       Illegal Use Of A Hydrant (CIN)
## 48                          Noise, Other Animals (NR6)
## 49                       Taste/Odor, Musty/Stale (QA4)
## 50                  Remove Hydrant Locking Device (WC6)
## 51                           Air: Smoke, Vehicular (AA4)
## 52            Oil Spill Into Basin/Sewer - Large (IABL)
## 53       Possible Water Main Break (Use Comments) (WA1)
## 54                                     Sewer Break (SBR)
## 55    No Sampling Required, Requested Information (QG2)
## 56              Other Water Problem (Use Comments) (WZZ)
## 57                       Unsafe Chemical, Storage (HC1)
## 58                                   Highway Flooding (SH)
## 59                  Manhole Sunken/Damaged/Raised (SB1)
## 60                      Plate Noisy/Sunken/Raised (SB5)
## 61                     Oil Spill On Street, Small (HQS)
## 62                             LOW WATER PRESSURE - WLWP
## 63                     Air: Smoke, Chimney or vent (AS1)
## 64                        Concrete In Catch Basin (IEA)
## 65          Air: Other Air Problem (Use Comments) (AZZ)
## 66                     Street Cave-In / Depression (SG)
## 67                   Excessive Water In Basement (WEFB)
```

```
DOB
```

```
##                                                                                .
## 1                                                                 Initial - Construction
## 2                                  Illegal Conversion Of Residential Building/Space
## 3                                     Cons - Contrary/Beyond Approved Plans/Permits
```

```
## 4                                                Electrical Wiring Defective/Exposed
## 5                                 Plumbing Work - Illegal/No Permit/Standpipe/Sprinkler
## 6                                                        Fence - None/Inadequate
## 7                                                            Failure To Maintain
## 8                                                  Elevator - Defective/Not Working
## 9                           Advertising Sign/Billboard/Posters/Flexible Fabric - Illegal
## 10                             Egress - Doors Locked/Blocked/Improper/No Secondary Means
## 11                                           Curb Cut/Driveway/Carport - Illegal
## 12                              Sidewalk Shed/Pipe Scafford - Inadequate Defective/None
## 13                                      Illegal. Commercial Use In Resident Zone
## 14                                   Illegal Hotel Rooms In Residential Building
## 15                          Plumbing Work - Unlicensed/Illegal/Improper Work In Progress
## 16                                  SRO - Illegal Work/No Permit/Change In Occupancy/Use
## 17                                       Building - Vacant, Open And Unguarded
## 18                                  Zoning - Non-Conforming/Illegal Vehicle Storage
## 19 Safety Netting/Guard Rails - Damaged/Inadequate/None (6 Stories/75 Feet Or Less)
## 20                                       Facade - Defective/Cracking (Ll11/98)
## 21                                 Failure To Retain Water/Improper Drainage- (LL103/89)
## 22                                      Debris - Falling Or In Danger Of Falling
## 23                                      Boiler - Defective/Inoperative/No Permit
## 24                                       Sign/Awning/Marquee - Illegal/No Permit
## 25                                           Working Contrary To Stop Work Order
## 26    Safety Netting/Guard Rails - Damaged/Inadequate/None (Over 6 Stories/75 Feet)
## 27                                       Site Conditions Endangering Workers
## 28                                                                Initial - BPP
## 29                                                    Re-Inspect - Unprepared
## 30                                                                 Initial - PA
## 31                                    Building Shaking/Vibrating/Structural Stability
## 32                                  No Certificate Of Occupancy/Illegal/Contrary To CO
## 33                                           Re-Inspect - Rslve Objections
## 34                                         Vent/Exhaust - Illegal/Improper
## 35                                Elevator - Dangerous Condition/Shaft Open/Unguarded
## 36                           Illegal Conversion Of Commercial Bldg/Space To Other Uses
## 37                                                                 Initial - CO
## 38                                             Landmark Bldg - Illegal Work
## 39                                        Wall/Retaining Wall - Bulging/Cracked
## 40                                     Gas Hook-Up/Piping - Illegal Or Defective
## 41                           Electrical - Unlicensed/Illegal/Improper Work In Progress
## 42                                           Sprinkler System - Inadequate
## 43                                    Contrary To LL 58/87(Handicapped Access)
## 44                                                        Routine Inspection
## 45                                   Posted Notice Or Order Removed/Tampered With
## 46                                                        Accident - Elevator
## 47                                    Illegal Tree Removal/Topo. Change in SNAD
## 48                        Suspended (Hanging) Scaffolds - No Pmt/Lic/Dangerous/Accident
## 49                                           Failure to Comply with Vacate Order
## 50                                    Plumbing-Defective/Leaking/Not Maintained
## 51                                                    Material Storage - Unsafe
## 52                                                        FDNY Referral - Pilot
## 53                                        Boiler - Fumes/Smoke/Carbon Monoxide
## 54                                                     Building Permit - None
## 55                                    Excavation Undermining Adjacent Building
## 56                                                     Re-Inspect - No Show
## 57                                    Elevator Not Inspected/Illegal/No Permit
```

```
## 58               Crane/Suspension Scaffold - No Permit/License/Cert./Unsafe/Illegal
## 59                                                        Investigative Inspection
## 60                                                             Demolition - Unsafe
## 61             Smoking Signs - "No Smoking" Signs Not Observed on Construction Site
## 62                                                     Sign - In Danger Of Falling
## 63                                                      After Hours Work - Illegal
## 64                                                             Con Edison Referral
## 65                                  Accident - Cranes/Derricks/Suspension Scaffold
## 66                                     Lights From Parking Lot Shining On Building
## 67                                                   Stalled Construction Site
## 68                                                             Structure - Indoors
## 69                                                 Demolition Notification Received
## 70                                    Privately Owned Public Space/Non-Compliance
## 71                                                              Debris - Excessive
## 72                                                   Inadequate Support Shoring
```

```
three11
```

```
##                      .
## 1 People Created Noise
## 2                  N/A
```

```
NYCEM
```

```
##                                                  .
## 1                    Ready NY - English - Full Size
## 2  Ready NY - Seniors and Disabled - Audio Cassette
## 3                   Hurricane Preparedness - English
## 4                     Ready NY - Spanish - Full Size
## 5         Ready NY - Chinese Traditional - Full Size
## 6            Ready NY Guide - Pocket Sized - English
## 7                          Ready NY - Reference Card
## 8          Ready NY - Seniors and Disabled - English
## 9                              Ready NY - Businesses
## 10              Ready NY My Emergency Plan - Russian
## 11                  Ready NY - English - Pocket Size
## 12              Ready NY My Emergency Plan - English
## 13                         Ready NY - Pets - English
## 14 Ready NY My Emergency Plan - Traditional Chinese
## 15              Ready NY My Emergency Plan - Spanish
## 16                             Summer Heat - English
## 17         Ready NY - Chinese Simplified - Full Size
## 18         Ready NY - Seniors and Disabled - Chinese
```

Looks like 311 complains are from noise, and the only relevant agencies are DEP and potentially NYCEM. I want to take a closer look at NYCEM before diving into DEP.

```
NYCEM <- query %>%
    filter(agency == "NYCEM") %>%
    pull(complaint_type) %>%
    unique()

NYCEM
```

```
## [1] "OEM Literature Request"
```

Looks like OEM literature requests for what to do during emergencies. Found here, https://www1.nyc.gov/site/em/ready/guides-resources.page

My Emergency Plan is a workbook designed to help New Yorkers — especially those with disabilities and access and functional needs — create an emergency plan.

This is not what we want. I will continue with DEP, querying only DEP requests.

Part 2: Diving into DEP data:

Pulling DEP 311 complaints:

Because there are many entries, I created an API token to not encounter throttle limits. There are millions of entries, so I will save in a more compact data type, RDS. We can load this in quicker next time. The Query will be commented out so we can query again if needed.

```
DEP_params <- soql() %>%
    soql_add_endpoint(api_endpoint) %>%
    soql_simple_filter("agency","DEP")


## For Querying:
# DEP_Query <- read.socrata(
#   DEP_params,
#   app_token = api_token
# )

# saveRDS(DEP_Query,"DEP_Data.rds")

DEP_Query <- read_rds("../3_Intermediate/DEP_Data.rds")
skim(DEP_Query)
```

Table 1: Data summary

Name	DEP_Query
Number of rows	2027029
Number of columns	35
Column type frequency:	
character	31
POSIXct	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
unique_key	0	1.00	8	8	0	2027029	0
agency	0	1.00	3	3	0	1	0
agency_name	0	1.00	38	38	0	1	0
complaint_type	0	1.00	3	19	0	23	0
descriptor	981	1.00	13	104	0	188	0
incident_zip	38893	0.98	5	5	0	236	0
incident_address	445011	0.78	3	39	0	524273	0
street_name	445011	0.78	2	32	0	16537	0
cross_street_1	173280	0.91	1	34	0	18465	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
cross_street_2	175098	0.91	1	34	0	19312	0
address_type	1304	1.00	7	12	0	4	0
city	38456	0.98	5	19	0	90	0
facility_type	284557	0.86	3	3	0	1	0
status	0	1.00	4	8	0	5	0
resolution_description	10213	0.99	44	335	0	232	0
community_board	234	1.00	8	25	0	77	0
bbl	560324	0.72	10	10	0	386493	0
borough	234	1.00	5	13	0	6	0
x_coordinate_state_plane	44694	0.98	6	7	0	119381	0
y_coordinate_state_plane	44694	0.98	6	6	0	127662	0
open_data_channel_type	0	1.00	5	7	0	5	0
park_facility_name	0	1.00	11	11	0	1	0
park_borough	234	1.00	5	13	0	6	0
latitude	44694	0.98	12	18	0	542599	0
longitude	44694	0.98	3	18	0	542603	0
location.latitude	44694	0.98	12	18	0	542599	0
location.longitude	44694	0.98	5	18	0	542603	0
location.human_address	44694	0.98	51	51	0	1	0
intersection_street_1	1579911	0.22	2	34	0	12164	0
intersection_street_2	1579911	0.22	1	35	0	12869	0
location_type	2026999	0.00	3	8	0	2	0

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
created_date	0	1.00	2010-01-01 00:24:00	2022-01-20 23:59:00	2016-03-16 22:05:00	1512669
closed_date	15523	0.99	2005-07-18 08:05:00	2022-01-20 23:15:00	2016-03-15 10:00:00	975856
resolution_action_updated_date	4830	1.00	2010-01-01 01:15:00	2927-03-06 12:30:00	2016-03-21 11:00:00	981573
due_date	2026673	0.00	1900-01-02 00:00:00	1900-01-02 00:00:00	1900-01-02 00:00:00	1

Looking at the most important complaint types:

Overall, it looks like the highest complaints are from water system, followed by Noise, and Sewer.

```
DEP_Query_by_type <-DEP_Query %>%
  group_by(complaint_type) %>%
  count() %>%
  arrange(by = n)

DEP_Query_by_type
```

```
## # A tibble: 23 x 2
## # Groups:   complaint_type [23]
##    complaint_type         n
##    <chr>              <int>
##  1 MSOTHER                1
```

```
##  2 SG-99                   1
##  3 SRGOVG                  1
##  4 ZSYSTEST                2
##  5 Hazardous Material      4
##  6 ZTESTINT                7
##  7 Internal Code          12
##  8 SRDE                   21
##  9 Water Maintenance      26
## 10 FCST                  183
## # ... with 13 more rows
```

Visualizing Time Series:

Let's visualize the DEP dataset at a couple different time frames. I'm going to create a plotting function so we can do this quickly. From looking on aggregate, I'm mostly interested in Air Quality, Lead, Water Conservation, Industrial Waste, Sewer, Water System, Asbestos, Hazardous Materials

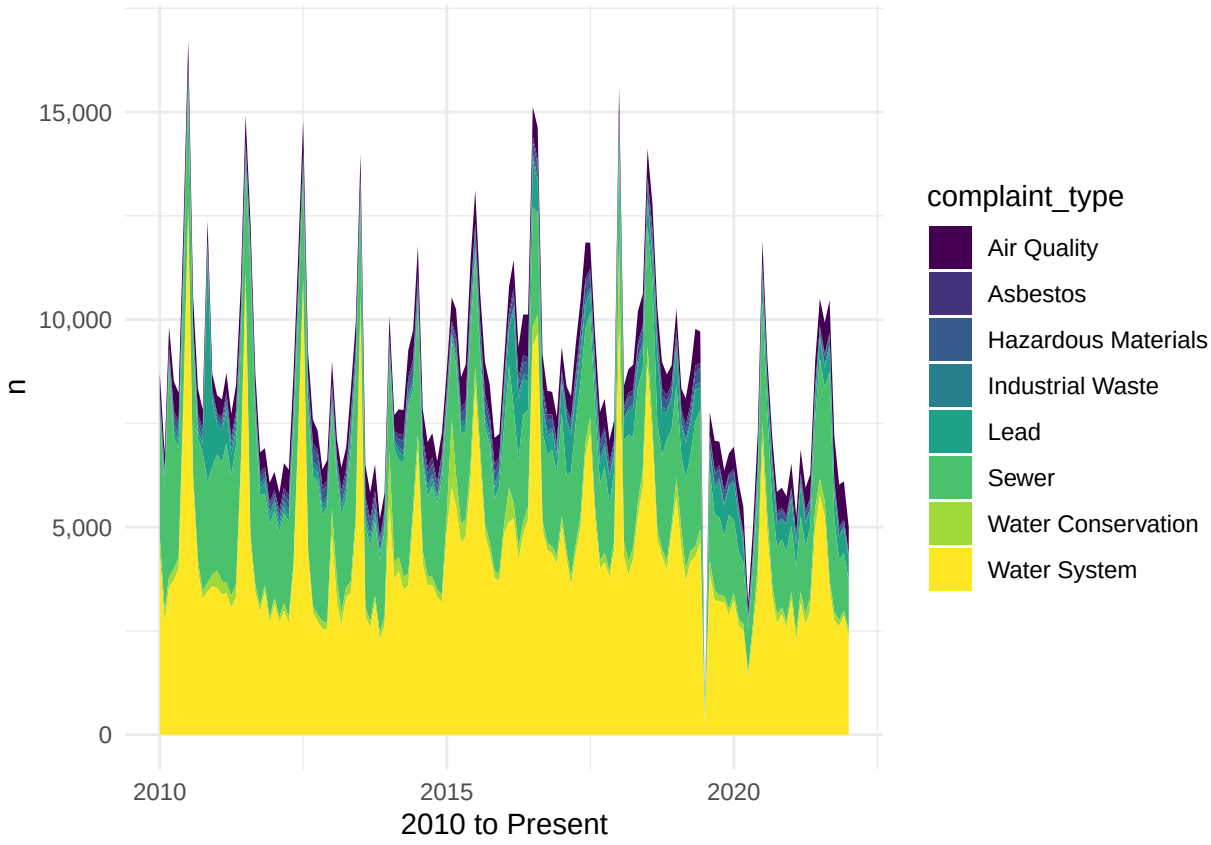
We will do monthly and annual because daily is rather large.

```
complaints <- c("Air Quality","Lead","Water Conservation","Industrial Waste",
                "Sewer","Water System","Asbestos","Hazardous Materials")

plotComplaints <- function(data,time_frame,n,complaint_type) {
  plot <- ggplot(data, aes_string(time_frame, n ,fill = complaint_type)) +
    geom_area() +
    scale_fill_viridis(discrete = T) +
    theme_minimal()  +
    xlab("2010 to Present") +
    scale_y_continuous(label=comma)
  return(plot)
}
plotline <- function(data,time_frame,n,complaint_type) {
  plot <- ggplot(data, aes_string(time_frame, n ,group = complaint_type, color = complaint_type)) +
    geom_line() +
    scale_color_viridis(discrete = T) +
    theme_minimal()  +
    xlab("2010 to Present") +
    scale_y_continuous(label=comma)
  return(plot)
}

DEP_time_series <- DEP_Query %>%
  filter(complaint_type %in% complaints) %>%
  group_by(created_date,complaint_type) %>%
  count() %>%
  mutate(month = month(created_date),
         year = year(created_date))

DEP_monthly <- DEP_time_series %>%
  group_by(month,year,complaint_type) %>%
  count() %>%
  mutate(plot_time = ymd(paste0(year,"-",month,"-1")))

DEP_annual <- DEP_monthly %>%
  group_by(year,complaint_type) %>%
```

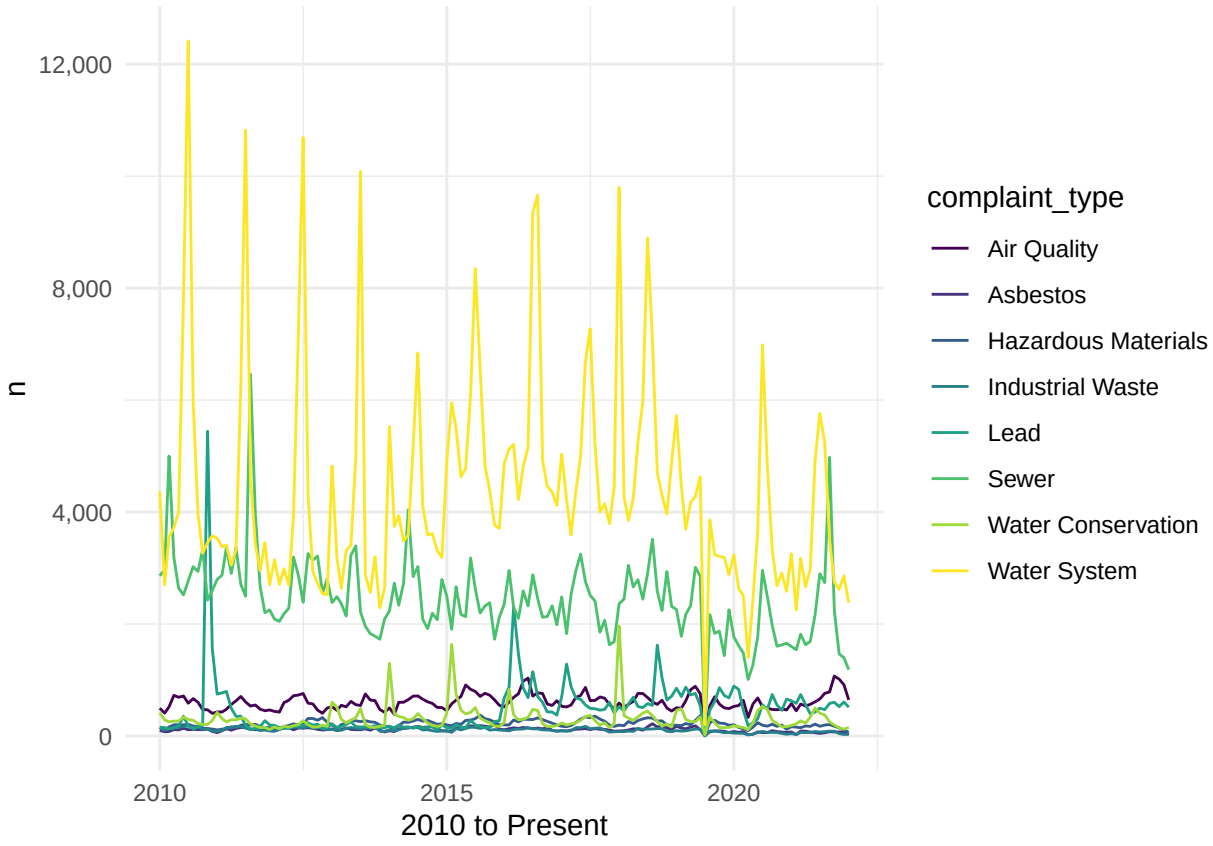
```
  summarise(n = sum(n)) %>%
  mutate(year = ymd(paste0(year,"-1-1")))
```

```
## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

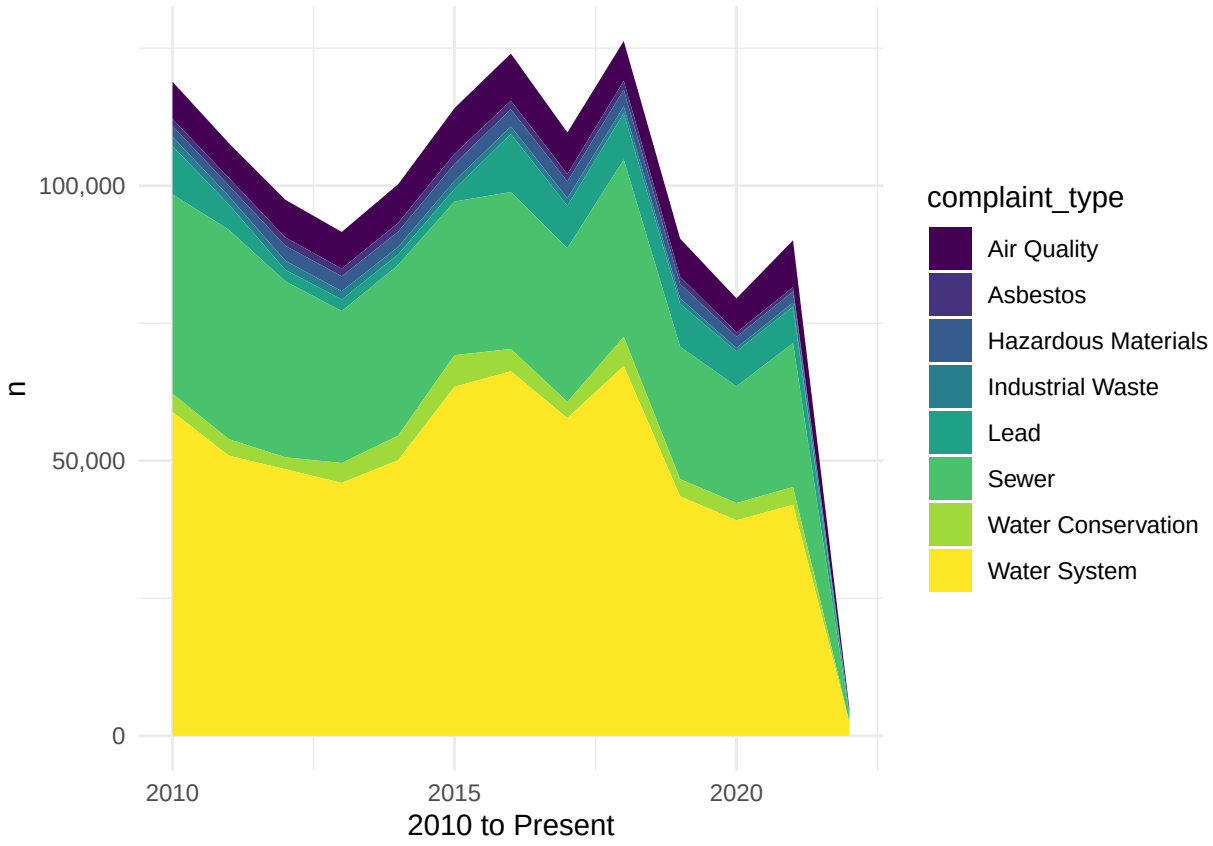
```
annual_plot <-plotComplaints(DEP_annual,"year","n","complaint_type")
annual_plot_line <-plotline(DEP_annual,"year","n","complaint_type")
monthly_plot <-plotComplaints(DEP_monthly,"plot_time","n","complaint_type")
monthly_plot_line <-plotline(DEP_monthly,"plot_time","n","complaint_type")

monthly_plot
```



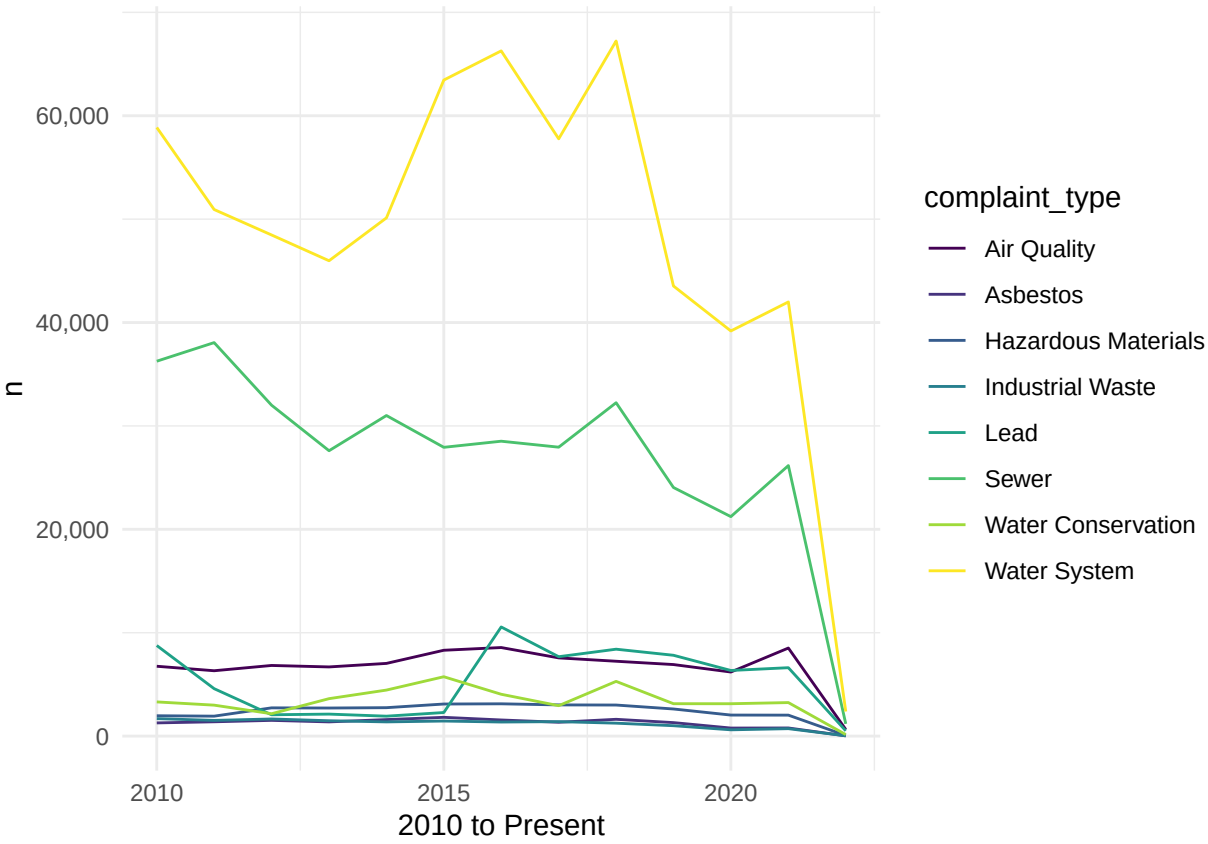

```
monthly_plot_line
```


12,000
8,000
4,000
0
n
2010
2015
2020
2010 to Present
complaint_type
Air Quality
Asbestos
Hazardous Materials
Industrial Waste
Lead
Sewer
Water Conservation
Water System

```
annual_plot
```




```
annual_plot_line
```



There's definitely a seasonality to the data for the water system. Perhaps this is due to flooding during the summer. Air quality complaints seem to have stayed the same. We can explore the lines closer by re-running the functions with different values of complaint_type if needed.

Looking at DEP complaint distribution by block and tract:

We can aggregate the complaints by census block or census tract to see where they are generally located. Because the data is large, we will do this on a sample and then expand this to all data from DEP.

```
# read shapefiles for census tract (New York Long Island CRS)
ct <- st_read("../2_Data/ct_2010/geo_export_1c19ce5f-d77c-4a3f-adbe-7456e4a782a6.shp") %>%
    st_transform(crs = 2263) %>%
    mutate(unique = paste0(boro_name,ct2010))
```

```
## Reading layer `geo_export_1c19ce5f-d77c-4a3f-adbe-7456e4a782a6' from data source `/Users/seanchew/Des
##   using driver `ESRI Shapefile'
## Simple feature collection with 2165 features and 11 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -74.25559 ymin: 40.49613 xmax: -73.70001 ymax: 40.91553
## Geodetic CRS:  WGS84(DD)
```

```
# read shapefiles for census blocks (New York Long Island CRS)
cb <- st_read("../2_Data/cb_2010/geo_export_b01427ec-0671-4e2f-b058-710f1b002026.shp") %>%
    st_transform(crs = 2263) %>%
    mutate(unique = paste0(boro_name,ct2010))
```

```
## Reading layer `geo_export_b01427ec-0671-4e2f-b058-710f1b002026' from data source `/Users/seanchew/Des
##   using driver `ESRI Shapefile'
```

```
## Simple feature collection with 38798 features and 7 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -74.25559 ymin: 40.49613 xmax: -73.70001 ymax: 40.91553
## Geodetic CRS:  WGS84(DD)
```

```
# Function for filtering complaints depending on what years and complaints we will
# later want to visualize
complaint_filter <- function(complaints,complaint_types,level,years){
  if(level == "cb"){
  filtered <- complaints %>%
    st_drop_geometry %>%
    filter(complaint_type %in% complaint_types,
           year %in% years) %>%
    group_by(cb2010,ct2010, boro_name) %>%
    summarise(n = sum(n))
  }
  else if( level == "ct"){
  filtered <- complaints %>%
    st_drop_geometry %>%
    filter(complaint_type %in% complaint_types,
           year %in% years) %>%
    group_by(ct2010, boro_name) %>%
    summarise(n = sum(n))
  return(filtered)
  }
}

# Function for creating choropleth maps depending on the filtered data
complaint_mapping <- function(shapefile,filtered,level,years,complaints){

  if(length(complaints) == length(c("Air Quality","Lead","Water Conservation","Industrial Waste",
                "Sewer","Water System","Asbestos","Hazardous Materials"))){
    complaints = "All Complaints"
  } else {
    complaints = glue_collapse(complaints,", ")
  }
      # tm_polygons(col = "n", palette = "viridis", lwd = .01) +
  if(level == "cb"){
    cb <- shapefile %>%
      left_join(filtered,by= c("cb2010","ct2010","boro_name"))
    map <- tm_shape(cb) +
      tm_fill("n",title="Complaints",style="jenks", palette = "viridis", lwd = .01) +
      tm_layout(main.title = paste0("NYC Complaints from \n",complaints),
                main.title.position = "center",
                main.title.size = 1,
                main.title.fontface = 2
      )
    return(map)
  } else if(level == "ct") {
    ct <- shapefile %>%
      left_join(filtered,by= c("ct2010","boro_name"))
    map <- tm_shape(ct) +
      tm_fill("n",title="Complaints",style="jenks", palette = "viridis", lwd = .01) +
```

```
      tm_layout(main.title = paste0("NYC Complaints from \n",complaints),
                main.title.position = "center",
                main.title.size = 1,
                main.title.fontface = 2
      )
    return(map)
  }
}

# Turn Query into SF object with points.
# We can use this later if we need to redo it, otherwise save it for faster processing.

# DEP_points <- DEP_Query %>%
#   filter(complaint_type %in% complaints) %>%
#   select(c("unique_key",
#            "complaint_type",
#            "created_date",
#            "x_coordinate_state_plane",
#            "y_coordinate_state_plane")) %>%
#   drop_na() %>%
#   st_as_sf(coords = c("x_coordinate_state_plane","y_coordinate_state_plane"),
#            crs = 2263) %>%
#   mutate(year = year(created_date))
#
# saveRDS(DEP_points,"../3_Intermediate/DEP_points.rds")

## We can use a smaller test sample size to see if our visualization is working:

DEP_points <- read_rds("../3_Intermediate/DEP_points.rds")

# sample of 10,000 to test out functions
dep_sample <- DEP_points[sample(nrow(DEP_points), 10000), ]

# spatial join to census blocks
cb_complaints <- cb %>%
  st_intersection(dep_sample) %>%
  group_by(cb2010,ct2010,boro_name,complaint_type,year) %>%
  count()
```

```
## Warning: attribute variables are assumed to be spatially constant throughout all
## geometries
```

```
# spatial join to census tracts
ct_complaints <- ct %>%
  st_intersection(dep_sample) %>%
  group_by(ct2010,boro_name,complaint_type,year) %>%
  count()
```

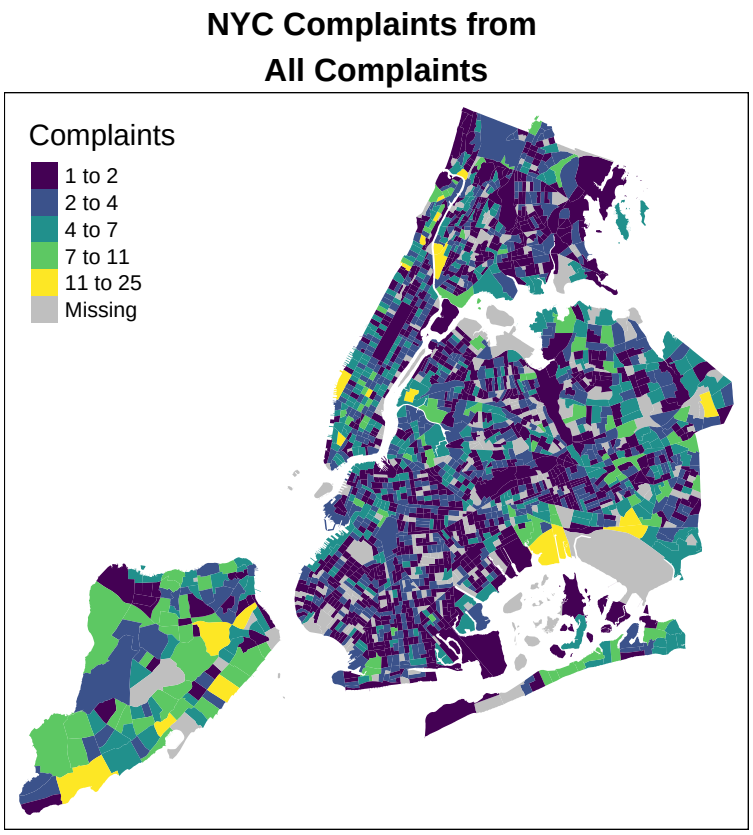
```
## Warning: attribute variables are assumed to be spatially constant throughout all
## geometries
```

```
# we can play around with the two inputs below:
years = c(2020:2010)
view_complaints = complaints
```

```
test<-complaint_filter(ct_complaints,view_complaints,"ct",years)
```

```
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
```

```
complaint_mapping(ct,test,"ct",years,view_complaints)
```



For the entire dataset:

Functions work for small sample, so now we can tweak those functions from above for the entire dataset, then break it down, by type, and by year if needed:

```
## Spatial Joins. These take extremely long, so saving for optimizing for time.

# cb_complaints_all <- cb %>%
#   st_intersection(DEP_points) %>%
#   group_by(cb2010,ct2010,boro_name,complaint_type,year) %>%
#   count()
#
# saveRDS(cb_complaints_all,"../3_Intermediate/cb_complaints_all.rds")
#
# ct_complaints_all <- ct %>%
#   st_intersection(DEP_points) %>%
#   group_by(ct2010,boro_name,complaint_type,year) %>%
#   count()
#
# saveRDS(ct_complaints_all,"../3_Intermediate/ct_complaints_all.rds")

cb_complaints_all <- read_rds("../3_Intermediate/cb_complaints_all.rds")
```

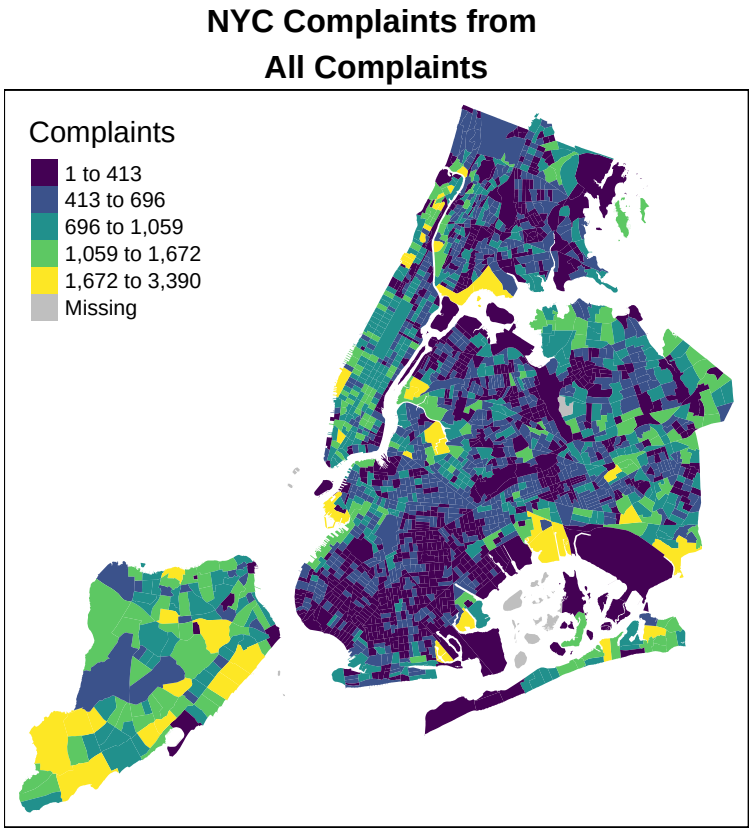
```
ct_complaints_all <- read_rds("../3_Intermediate/ct_complaints_all.rds")

# We can adjust input years, and input complaints as needed.
years_all = c(2020:2010)
view_complaints_all = complaints

# Filter data to what inputs we want.
complaint_filtered_ct <-complaint_filter(ct_complaints_all,view_complaints_all,"ct",years_all)
```

```
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
```

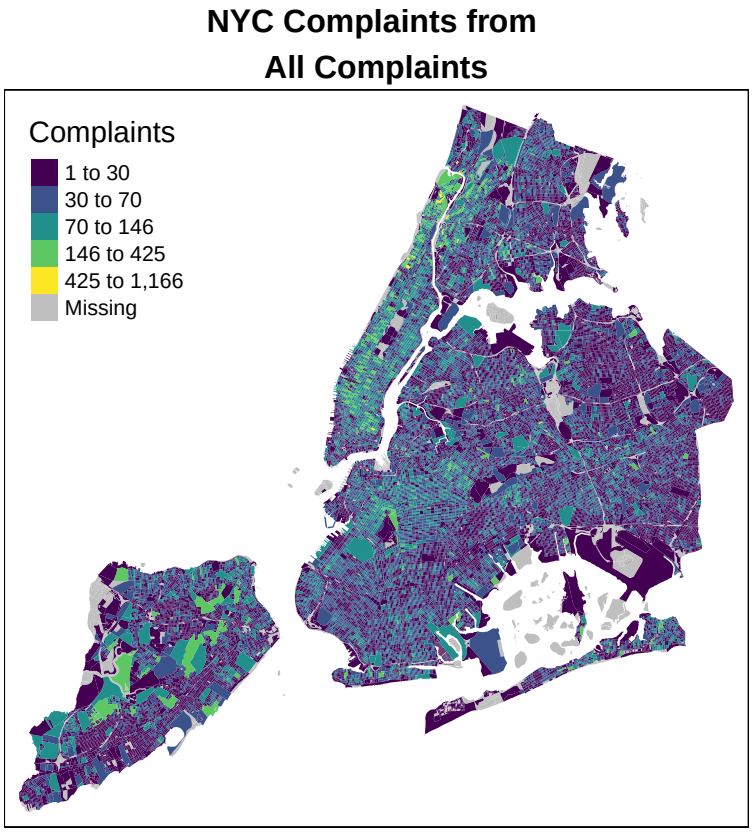
```
complaint_filtered_cb <-complaint_filter(cb_complaints_all,view_complaints_all,"cb",years_all)
```

```
## `summarise()` has grouped output by 'cb2010', 'ct2010'. You can override using the `.groups` argumen
```

```
# Map all complaints, over all years.
complaint_mapping(ct,complaint_filtered_ct,"ct",years_all,view_complaints)
```




```
complaint_mapping(cb,complaint_filtered_cb,"cb",years_all,view_complaints)
```



Looking at complaints by type:

```
## Adjust inputs if needed:
years_all = c(2020:2010)
view_complaints_all = complaints

## Mapping 8 times, we need 8 datasets for ct and cb
map_ct_complaints <- rep(list(ct_complaints_all),8)
map_cb_complaints <- rep(list(cb_complaints_all),8)

## Map over the complaints 8 times
map_complaints <- c("Air Quality","Lead","Water Conservation","Industrial Waste",
                "Sewer","Water System","Asbestos","Hazardous Materials")

## Input "ct" level and "cb" level 8 times each
map_ct <- rep("ct",8)
map_cb <- rep("cb",8)

## For now, put in all years, 8 times.
map_years <- rep(list(years),8)

## Prepare inputs for filtering
input_ct <- list(map_ct_complaints,map_complaints,map_ct,map_years)
input_cb <- list(map_cb_complaints,map_complaints,map_cb,map_years)

## Filter ct and cb inputs.
filtered_ct_all <- pmap(input_ct,complaint_filter)
```

```
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
```

```
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
```

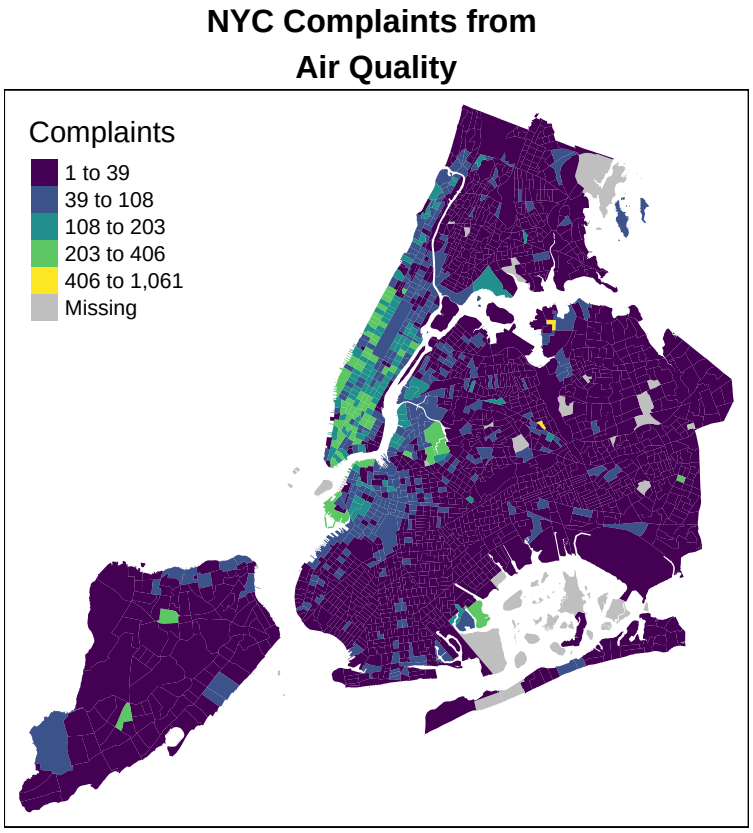
```
filtered_cb_all <- pmap(input_cb,complaint_filter)
```

```
## `summarise()` has grouped output by 'cb2010', 'ct2010'. You can override using the `.groups` argumen
```

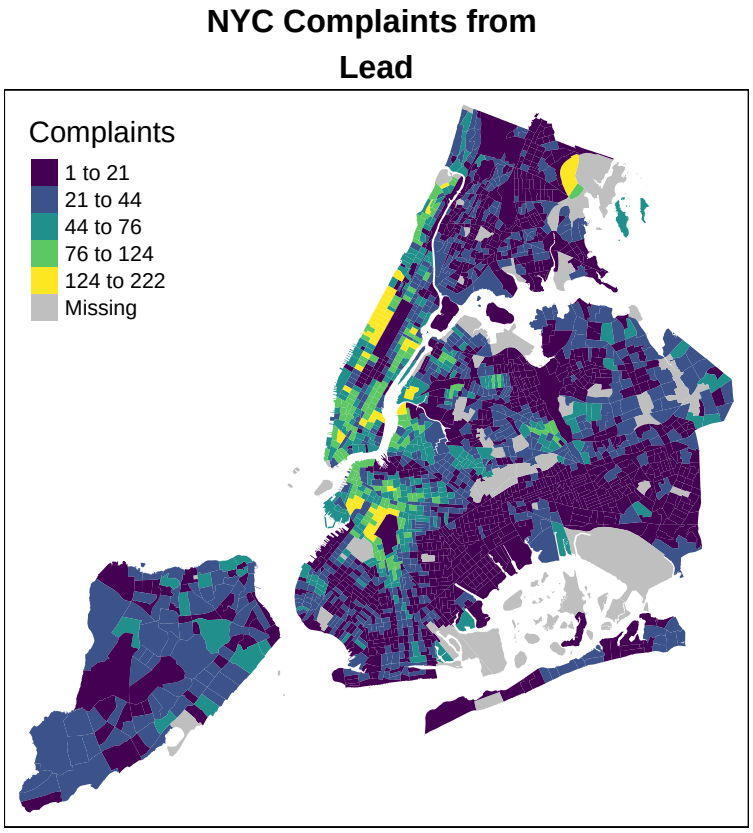
```
## `summarise()` has grouped output by 'cb2010', 'ct2010'. You can override using the `.groups` argumen
## `summarise()` has grouped output by 'cb2010', 'ct2010'. You can override using the `.groups` argumen
## `summarise()` has grouped output by 'cb2010', 'ct2010'. You can override using the `.groups` argumen
## `summarise()` has grouped output by 'cb2010', 'ct2010'. You can override using the `.groups` argumen
## `summarise()` has grouped output by 'cb2010', 'ct2010'. You can override using the `.groups` argumen
## `summarise()` has grouped output by 'cb2010', 'ct2010'. You can override using the `.groups` argumen
```

```
## Prepare inputs for plotting:
map_ct_shapes <- rep(list(ct),8)
map_cb_shapes <- rep(list(cb),8)

input_ct_plot <- list(map_ct_shapes,filtered_ct_all, map_ct,map_years,map_complaints)
input_cb_plot <- list(map_cb_shapes,filtered_cb_all, map_cb,map_years,map_complaints)

plots_ct <- pmap(input_ct_plot,complaint_mapping)
plots_cb <- pmap(input_cb_plot,complaint_mapping)

plots_ct
```

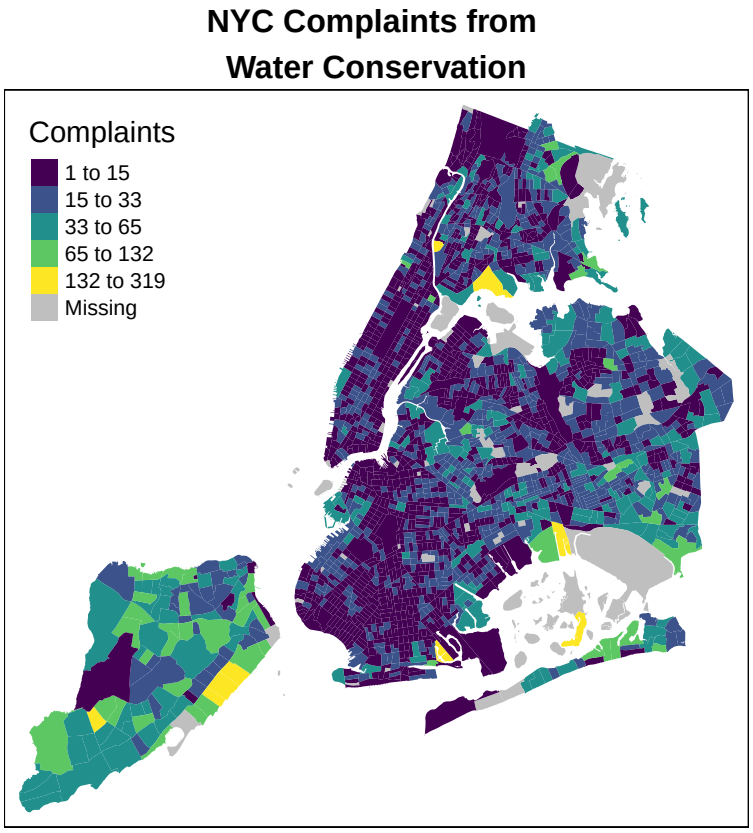
```
## [[1]]
```


NYC Complaints from
Air Quality
Complaints
1 to 39
39 to 108
108 to 203
203 to 406
406 to 1,061
Missing

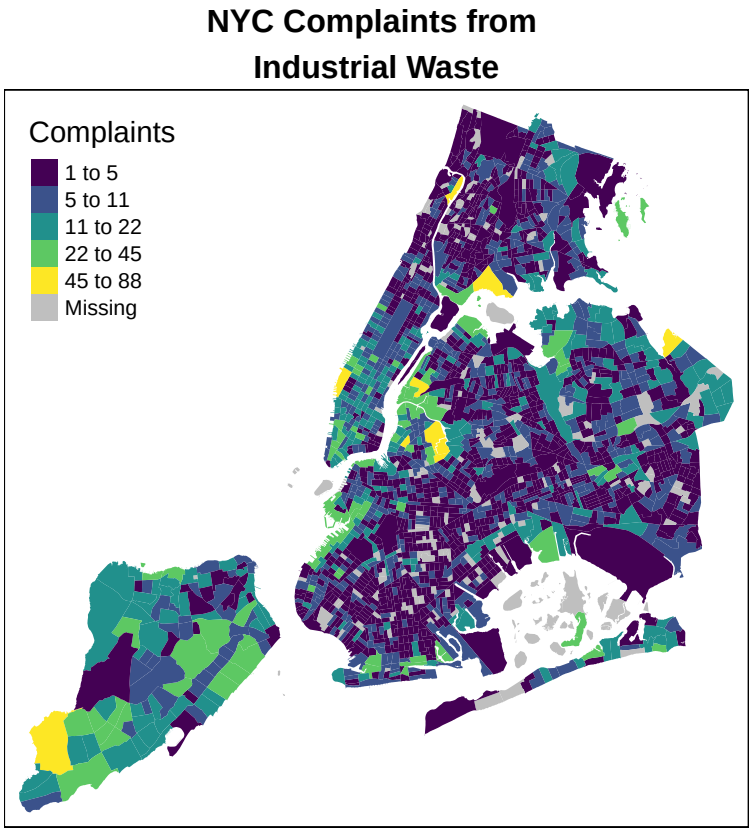
```
##
## [[2]]
```


NYC Complaints from Lead
Complaints
1 to 21
21 to 44
44 to 76
76 to 124
124 to 222
Missing

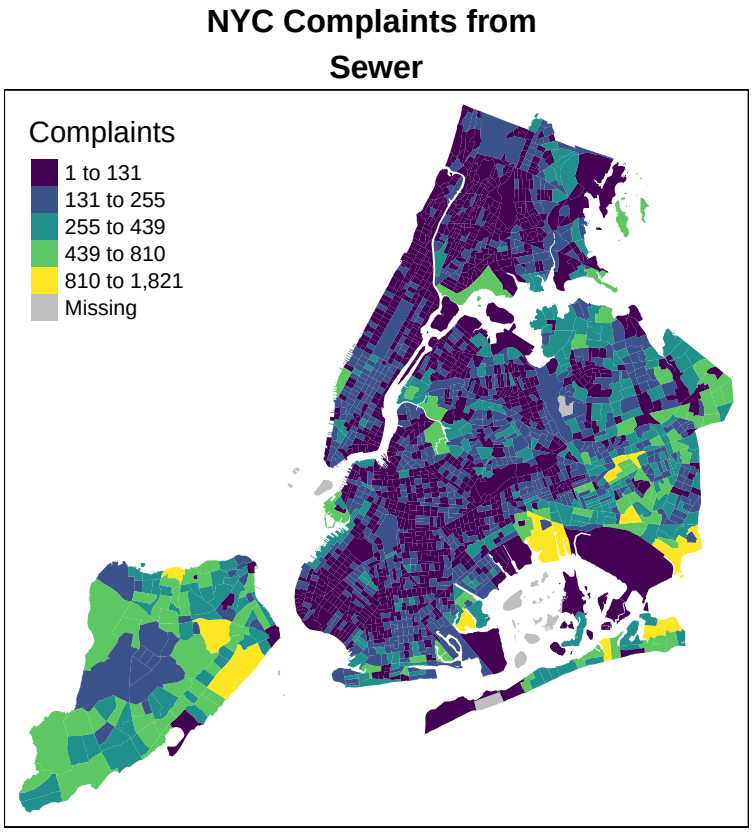
```
##
## [[3]]
```


NYC Complaints from
Water Conservation
Complaints
1 to 15
15 to 33
33 to 65
65 to 132
132 to 319
Missing

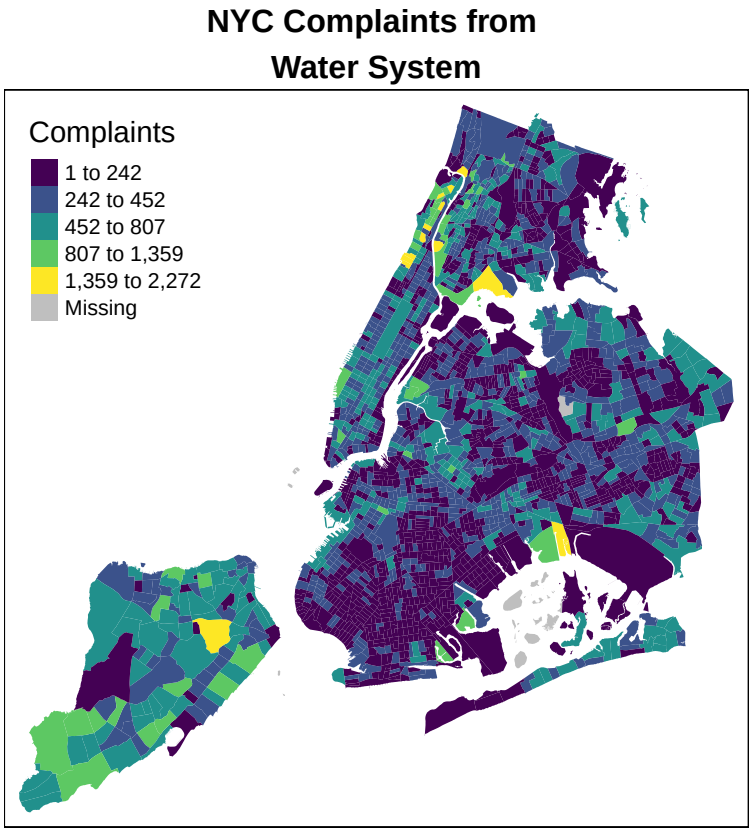
```
##
## [[4]]
```


NYC Complaints from
Industrial Waste
Complaints
1 to 5
5 to 11
11 to 22
22 to 45
45 to 88
Missing

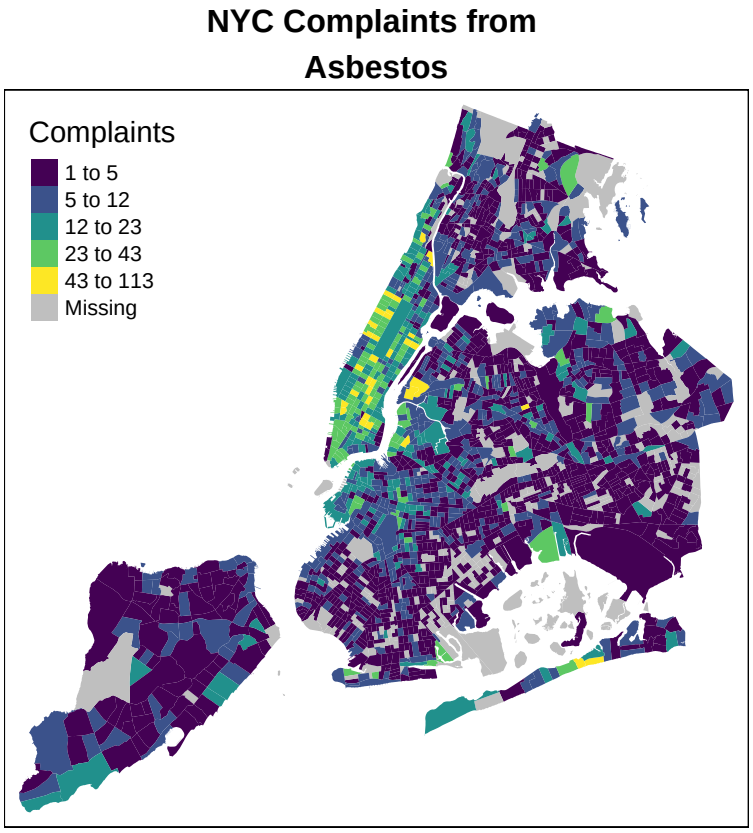
```
##
## [[5]]
```


NYC Complaints from Sewer
Complaints
1 to 131
131 to 255
255 to 439
439 to 810
810 to 1,821
Missing

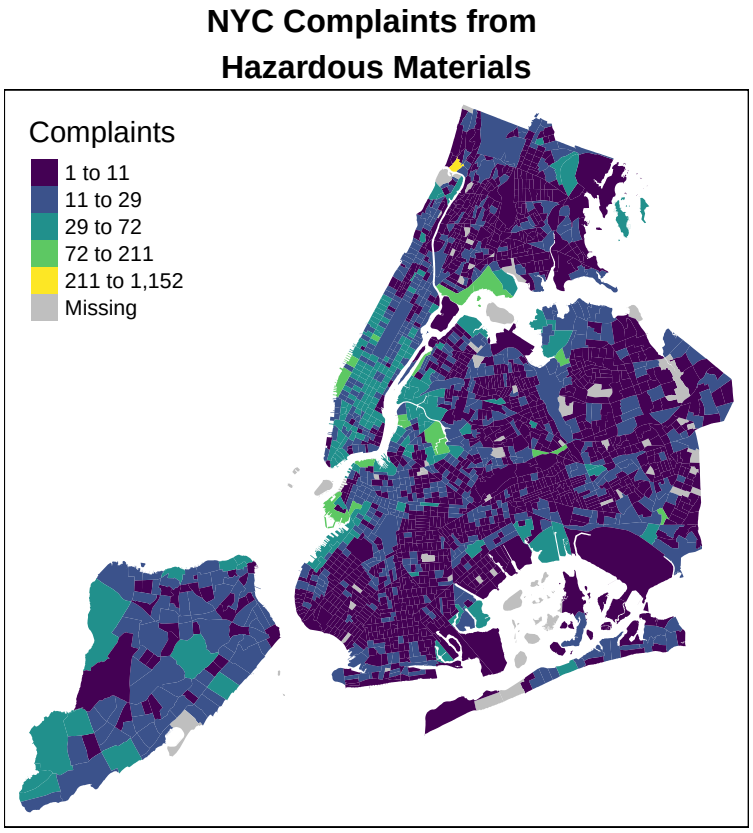
##
[[6]]


NYC Complaints from
Water System
Complaints
1 to 242
242 to 452
452 to 807
807 to 1,359
1,359 to 2,272
Missing

```
##
## [[7]]
```

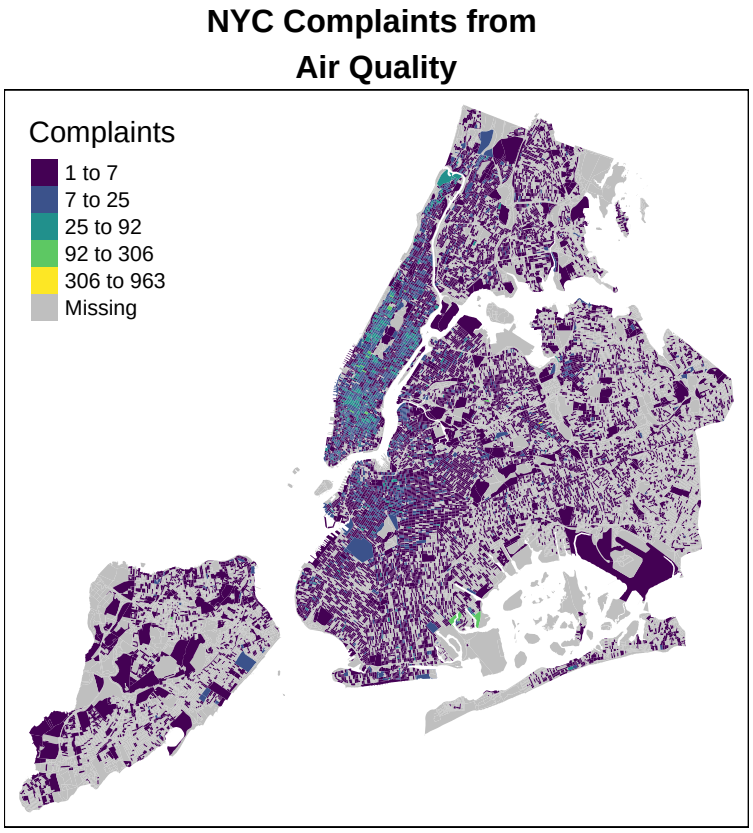

NYC Complaints from
Asbestos
Complaints
1 to 5
5 to 12
12 to 23
23 to 43
43 to 113
Missing

```
##
## [[8]]
```

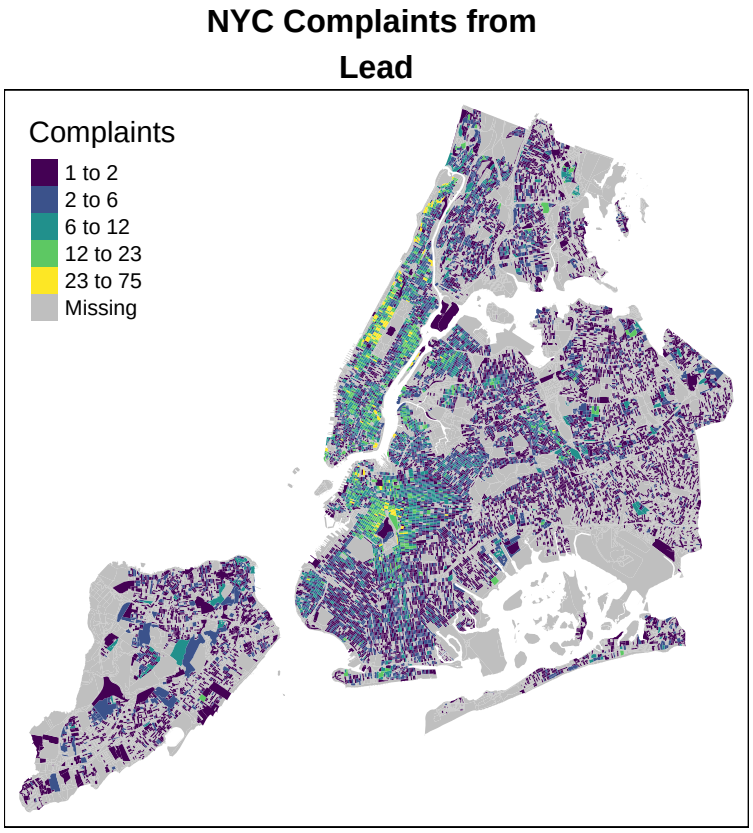

NYC Complaints from
Hazardous Materials
Complaints
1 to 11
11 to 29
29 to 72
72 to 211
211 to 1,152
Missing

```
plots_cb
```

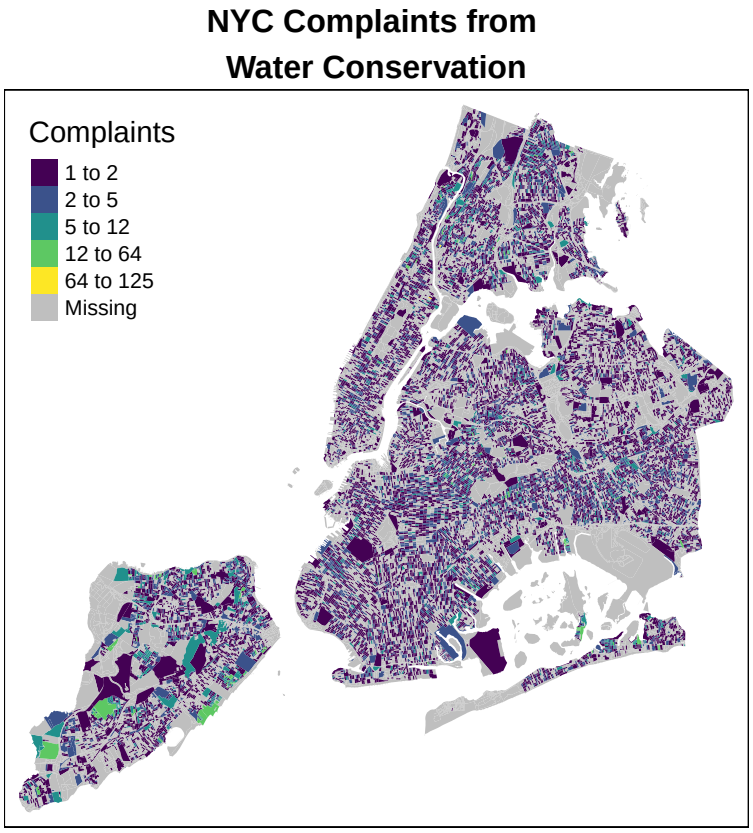
```
## [[1]]
```


NYC Complaints from
Air Quality
Complaints
1 to 7
7 to 25
25 to 92
92 to 306
306 to 963
Missing

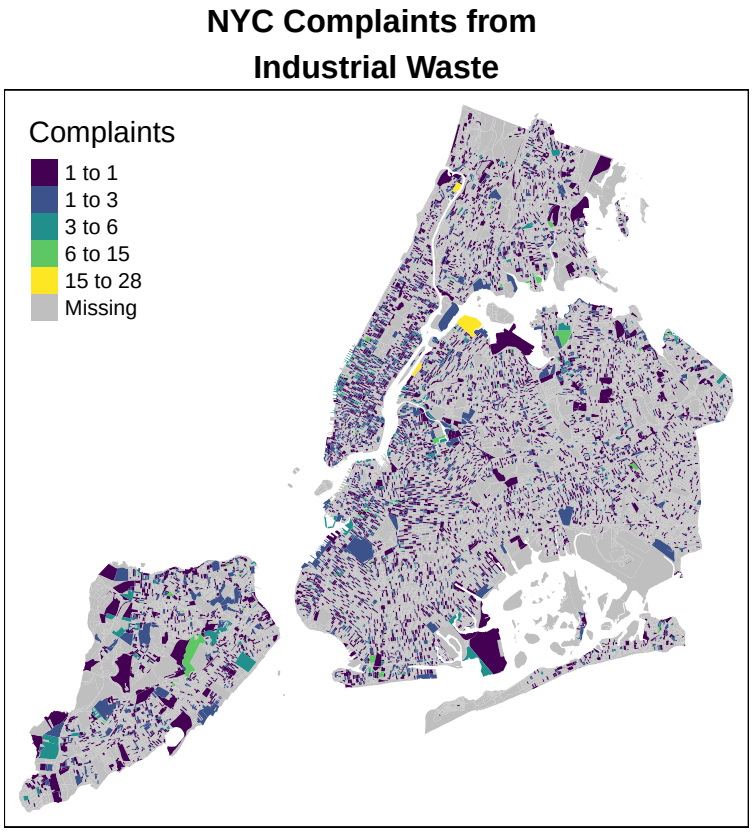
```
##
## [[2]]
```


NYC Complaints from Lead
Complaints
1 to 2
2 to 6
6 to 12
12 to 23
23 to 75
Missing

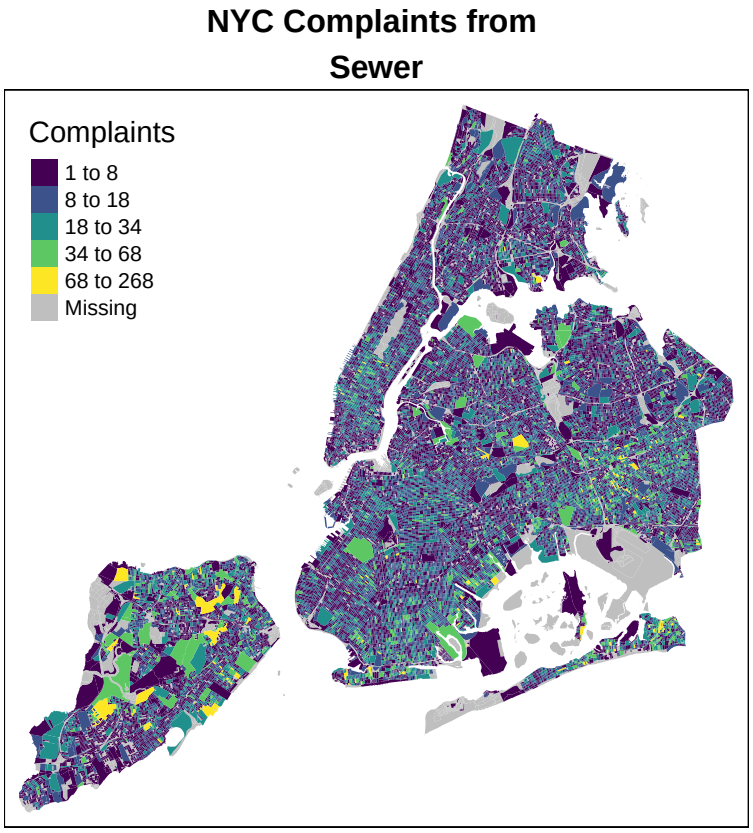
```
##
## [[3]]
```


NYC Complaints from
Water Conservation
Complaints
1 to 2
2 to 5
5 to 12
12 to 64
64 to 125
Missing

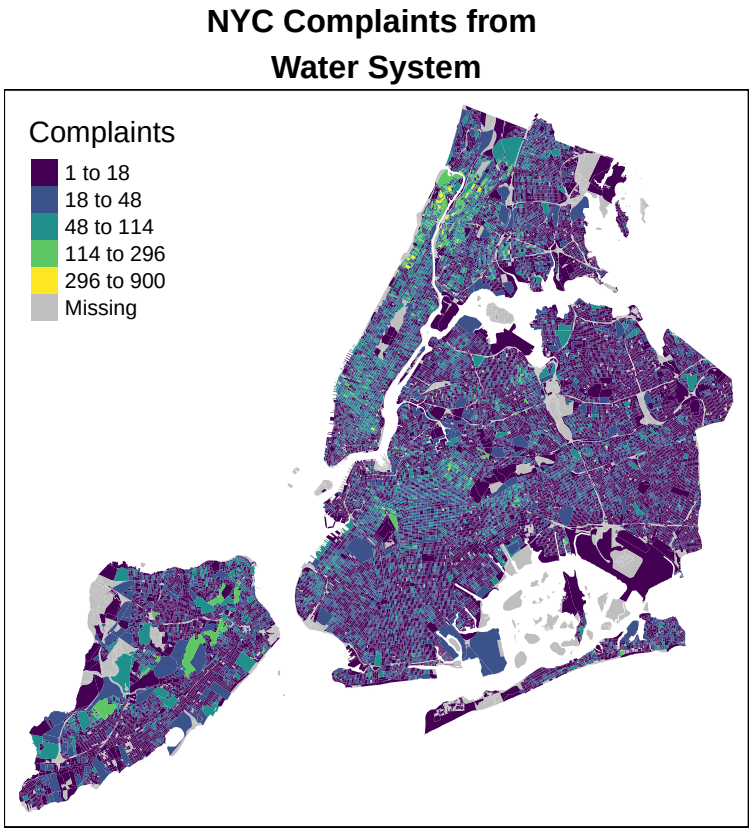
```
##
## [[4]]
```


NYC Complaints from
Industrial Waste
Complaints
1 to 1
1 to 3
3 to 6
6 to 15
15 to 28
Missing

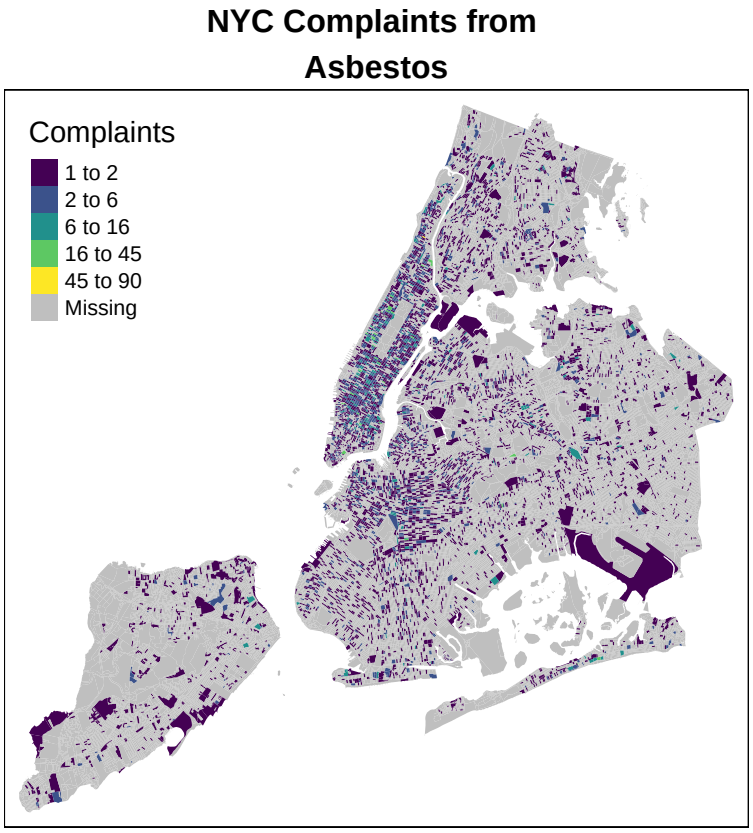
```
##
## [[5]]
```


NYC Complaints from Sewer
Complaints
1 to 8
8 to 18
18 to 34
34 to 68
68 to 268
Missing

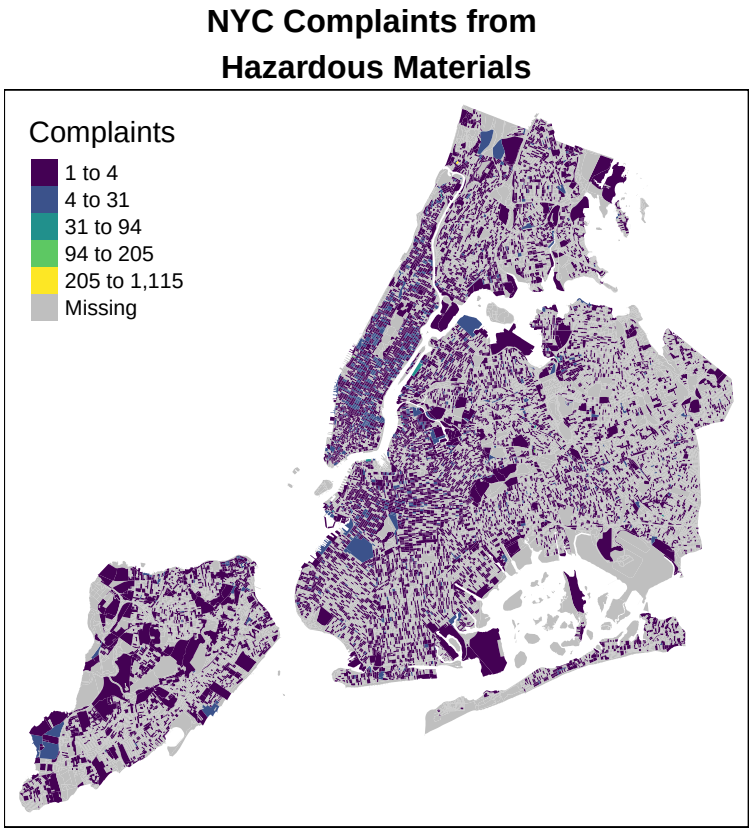
```
##
## [[6]]
```


NYC Complaints from
Water System
Complaints
1 to 18
18 to 48
48 to 114
114 to 296
296 to 900
Missing

```
##
## [[7]]
```


NYC Complaints from
Asbestos
Complaints
1 to 2
2 to 6
6 to 16
16 to 45
45 to 90
Missing

```
##
## [[8]]
```



First Impressions:

Air Quality is the worst in Manhattan, with some places in queens with extremely high complaint levels.

Lead complaints occur in the upper west side (yikes!) with large amounts in Brooklyn as well.

Water conservation complaints occur around coastlines, and around Staten Island.

Industrial Waste occurs in some particular regions, where there probably are larger amounts of industrial processes (construction, near Chelsea for example)

Problem spots for the water system seem to occur in the Bronx and in Staten Island.

Asbestos is the most concentrated in Manhattan and the portions of Manhattan and Queens that border Manhattan.

Certain hot spots of hazardous waste occur in Statten island, lower Manhattan, and certain areas of Queens.

Next Steps:

Some additional thoughts: 1) Can look at complaint channel (mobile, online, other) trends (over time, and distribution) - these could be indicators of access - Maybe we can add in socio-economic status, internet access as additional information? 2) Can look at the yearly change of certain complaints. 3) Can incorporate machine learning techniques to predict where future clusters may be held.