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 415954 22.3     855035 45.7         NA   666808 35.7
## Vcells 789119  6.1    8388608 64.0      16384  1824213 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,
tidycensus)
```

Functions Used:

```
source("./Exploration_Functions.R")
```

Data Exploration of 311 Data

Part 1: Exploring different agencies and setting API up:

First Test of 311 data:

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

First step will be to set up our environment and begin querying using the socrata package. I will test 10,000 entries first.

```
query_params <- soql() %>%
    soql_add_endpoint(api_endpoint) %>%
    soql_limit(10000)

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

```
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
```

```
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)
```

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
```

```
three11
```

```
##                       .
## 1 People Created Noise
```

```
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
```

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.

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

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
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	4330	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")


# For all Complaints
DEP_time_series_all <- DEP_Query %>%
  group_by(created_date,complaint_type) %>%
  count() %>%
  mutate(month = month(created_date),
         year = year(created_date))

DEP_monthly_all <- DEP_time_series_all %>%
  group_by(month,year,complaint_type) %>%
  summarise(n = sum(n)) %>%
  mutate(plot_time = ymd(paste0(year,"-",month,"-1")),
         perc = n/sum(n)) %>%
  filter(complaint_type %in% complaints)
```

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

```
DEP_annual_all <- DEP_time_series_all %>%
  group_by(year,complaint_type) %>%
  summarise(n = sum(n)) %>%
  mutate(year = ymd(paste0(year,"-1-1")),
         perc = n/sum(n)) %>%
  filter(complaint_type %in% complaints)
```

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

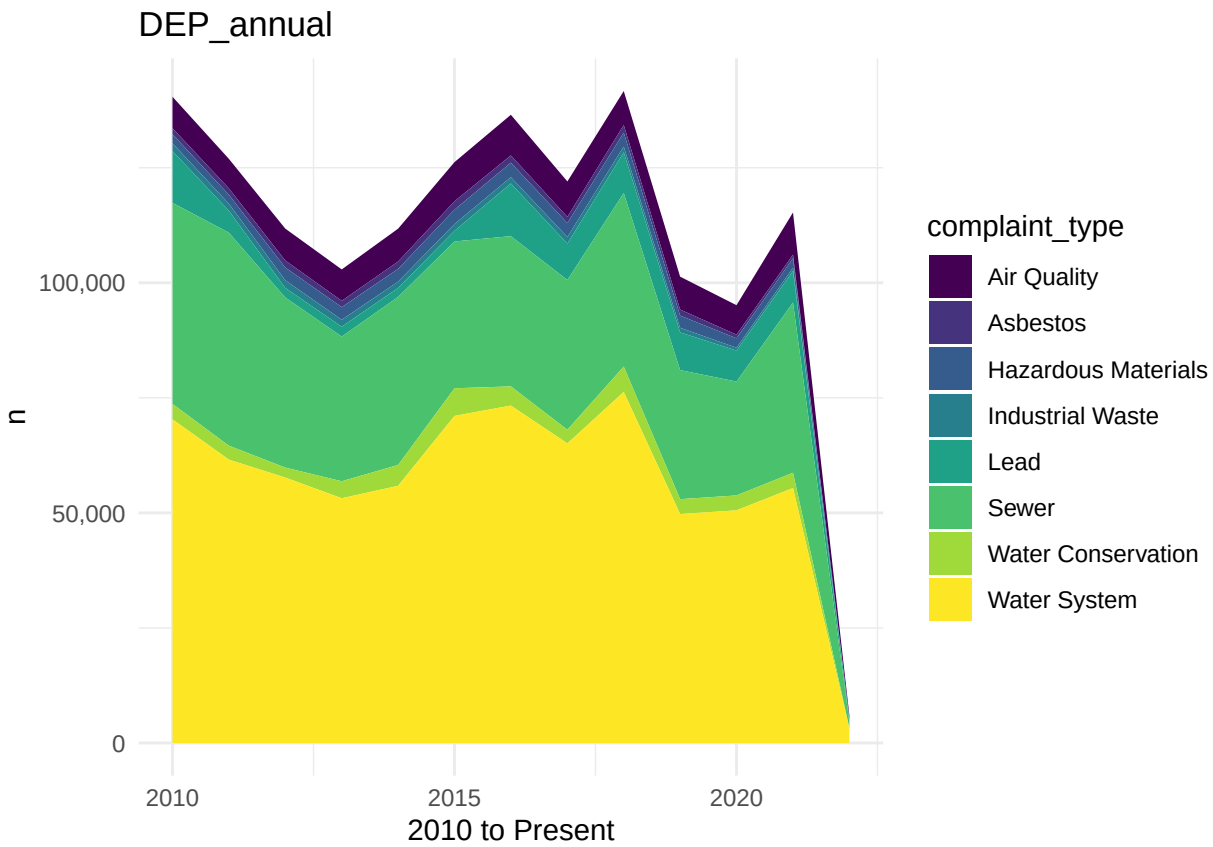
```
# Subset of complaints
DEP_time_series <- DEP_Query %>%
  filter(complaint_type %in% complaints) %>%
  group_by(created_date,complaint_type) %>%
  count() %>%
  mutate(month = month(created_date),
         year = year(created_date))

DEP_monthly <- DEP_time_series %>%
  group_by(month,year,complaint_type) %>%
  summarise(n = sum(n)) %>%
  mutate(plot_time = ymd(paste0(year,"-",month,"-1")),
         perc = n/sum(n))
```

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

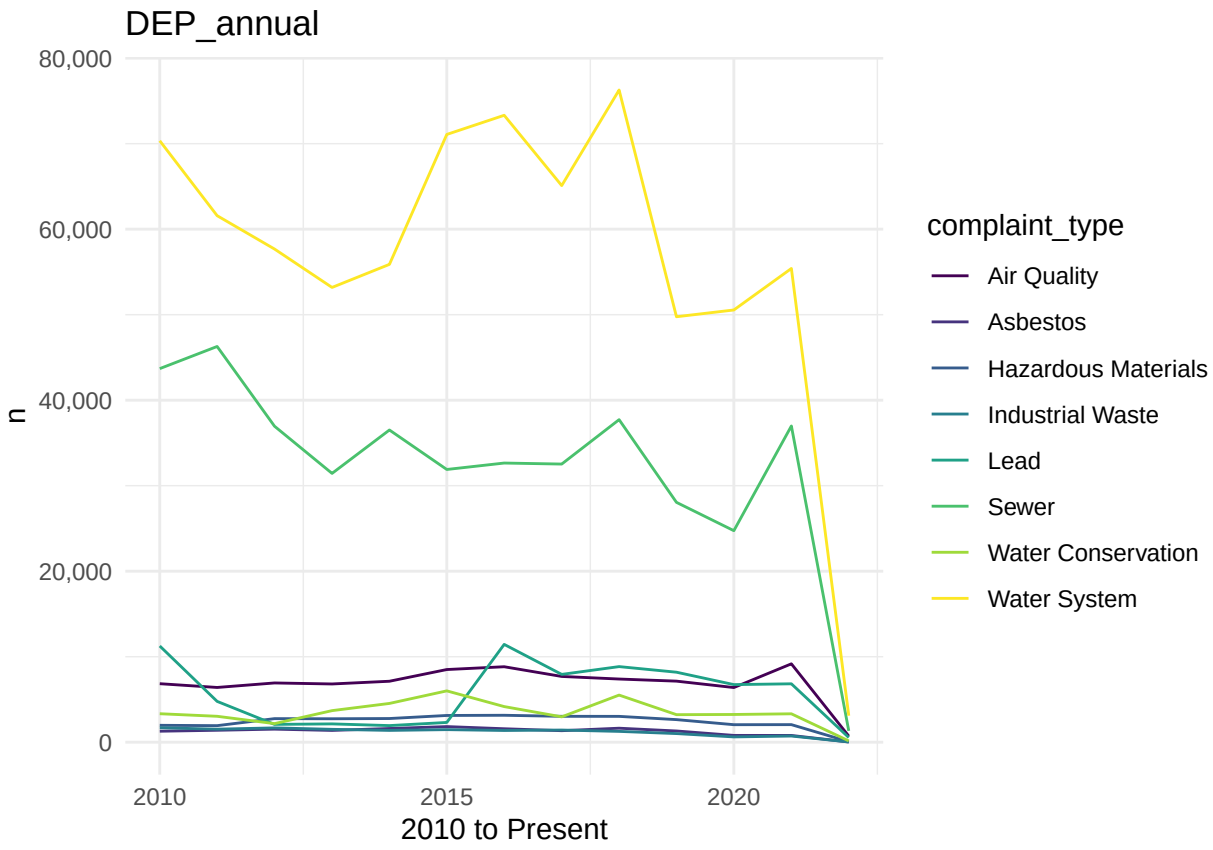
```
DEP_annual <- DEP_monthly %>%
  group_by(year,complaint_type) %>%
  summarise(n = sum(n)) %>%
  mutate(year = ymd(paste0(year,"-1-1")),
         perc = n/sum(n))
```

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

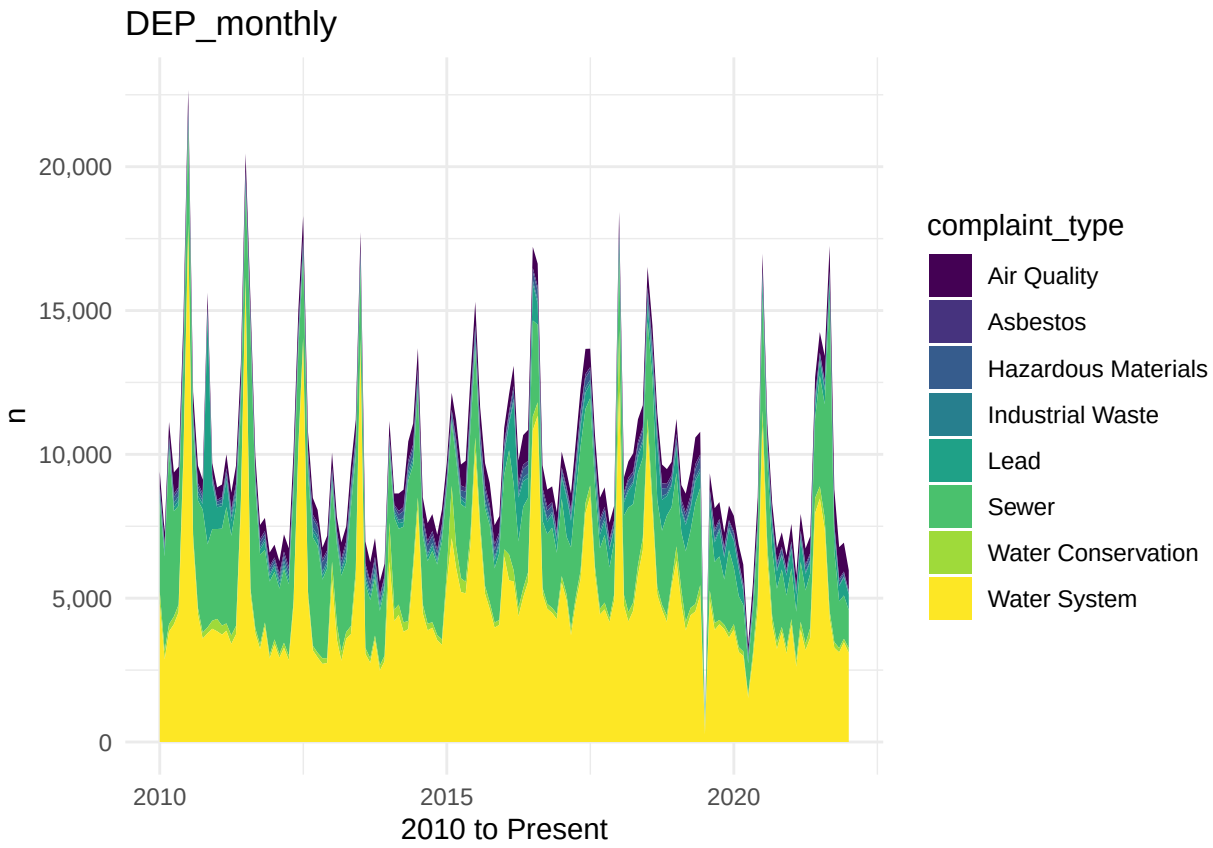
```
# Volume Plots
dep_filtered_annual_plot_volume <-plotComplaints(DEP_annual,"year","n","complaint_type")
dep_filtered_annual_plot_line_volume <-plotline(DEP_annual,"year","n","complaint_type")
dep_filtered_monthly_plot_volume <-plotComplaints(DEP_monthly,"plot_time","n","complaint_type")
dep_filtered_monthly_plot_line_volume <-plotline(DEP_monthly,"plot_time","n","complaint_type")

# Proportion Plots:
dep_filtered_annual_plot_perc <-plotComplaints(DEP_annual,"year","perc","complaint_type")
dep_filtered_monthly_plot_perc <-plotComplaints(DEP_monthly,"plot_time","perc","complaint_type")
dep_annual_plot_perc <-plotComplaints(DEP_annual_all,"year","perc","complaint_type")
dep_monthly_plot_perc <-plotComplaints(DEP_monthly_all,"plot_time","perc","complaint_type")

# Look at all plots:
dep_filtered_annual_plot_volume
```



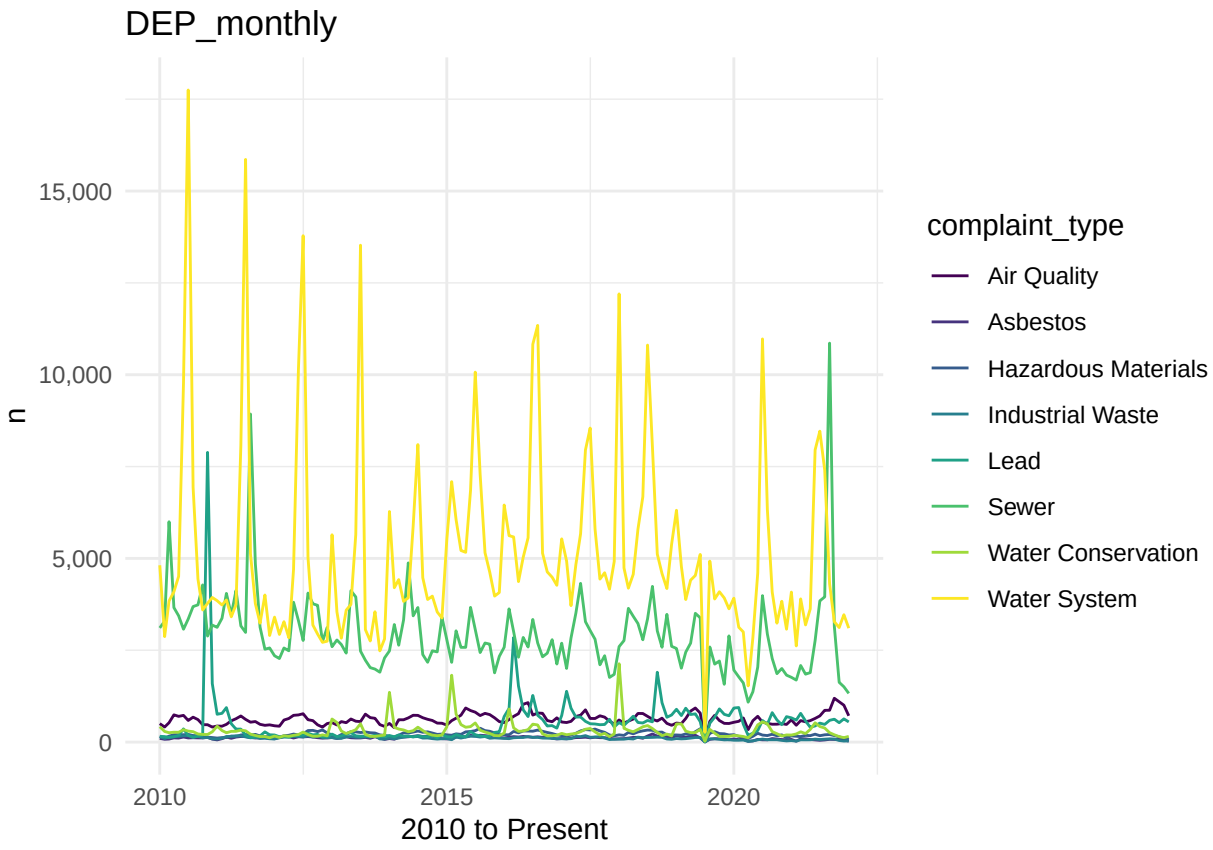

```
dep_filtered_annual_plot_line_volume
```



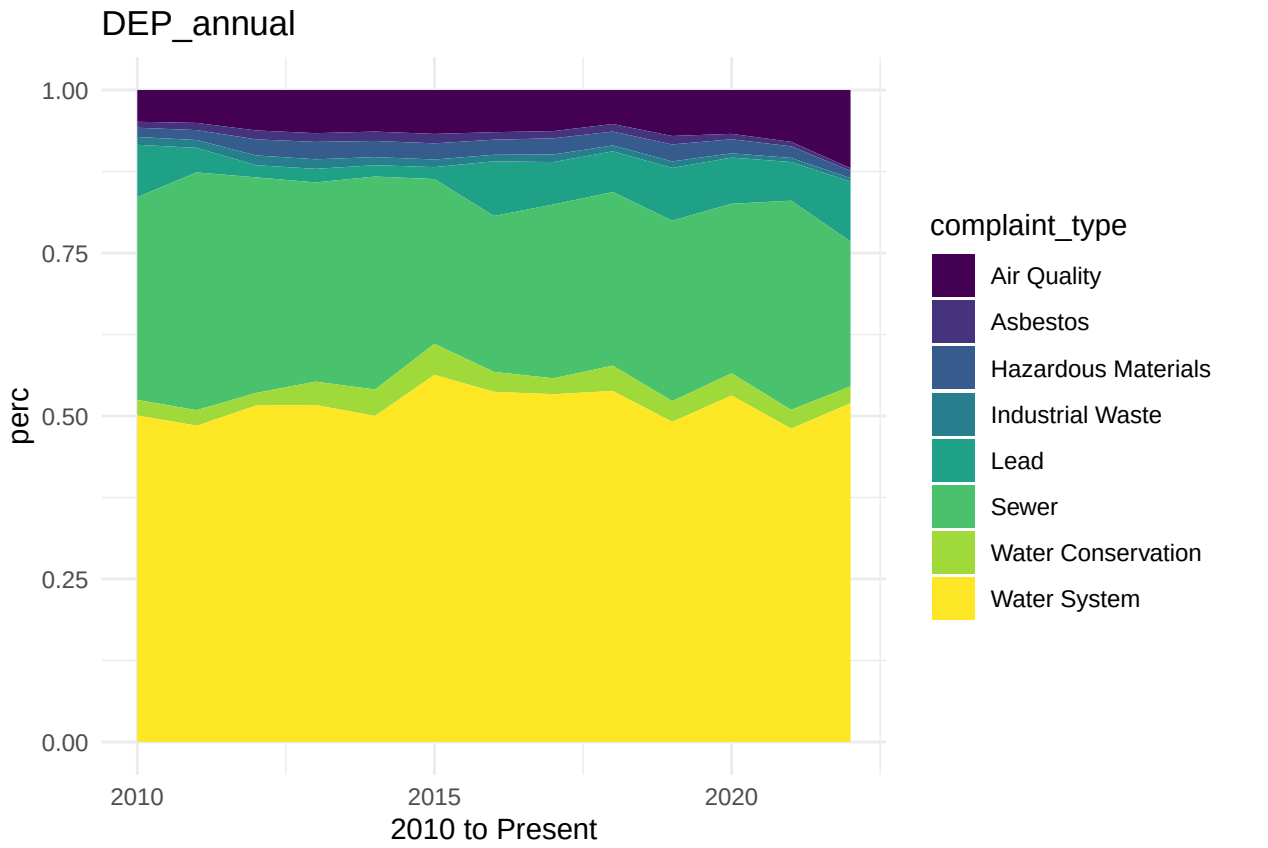

```
dep_filtered_monthly_plot_volume
```



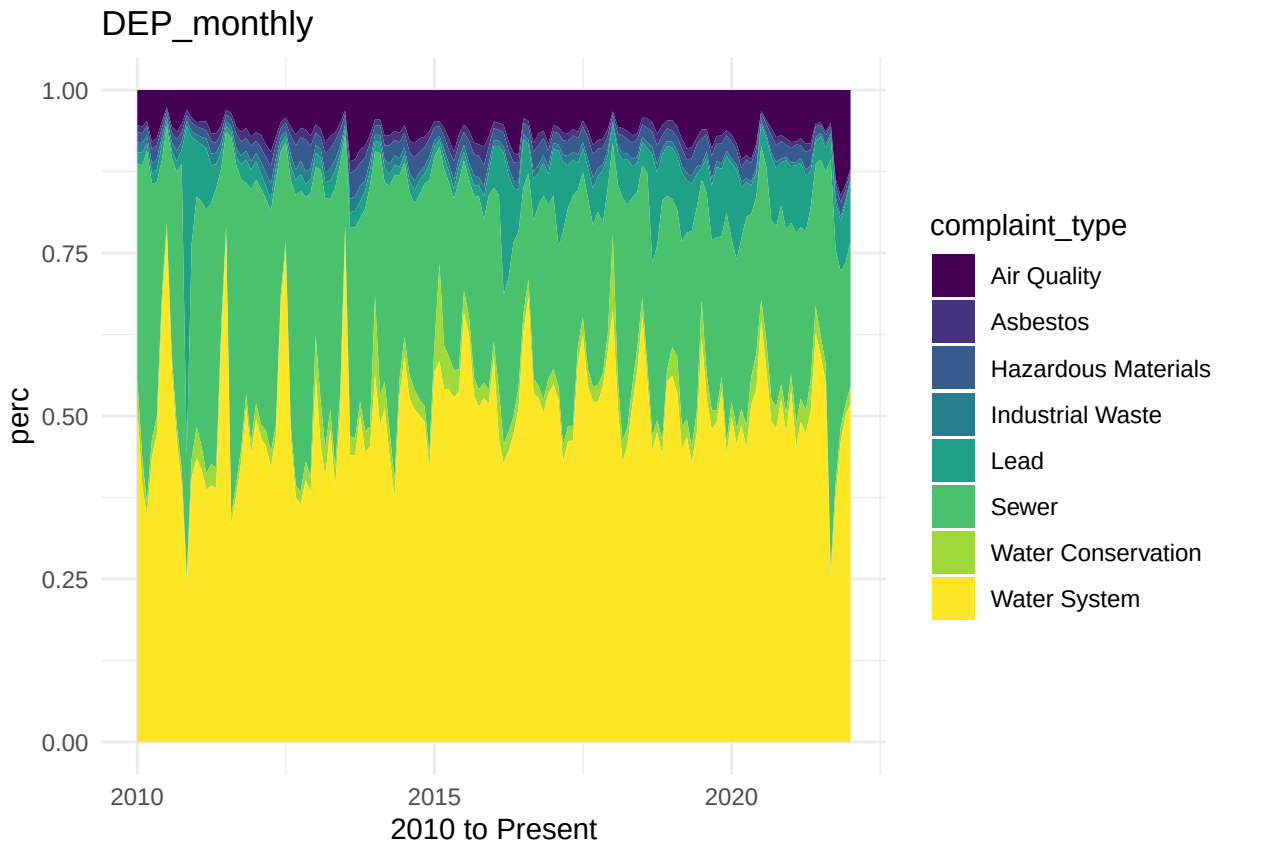

```
dep_filtered_monthly_plot_line_volume
```



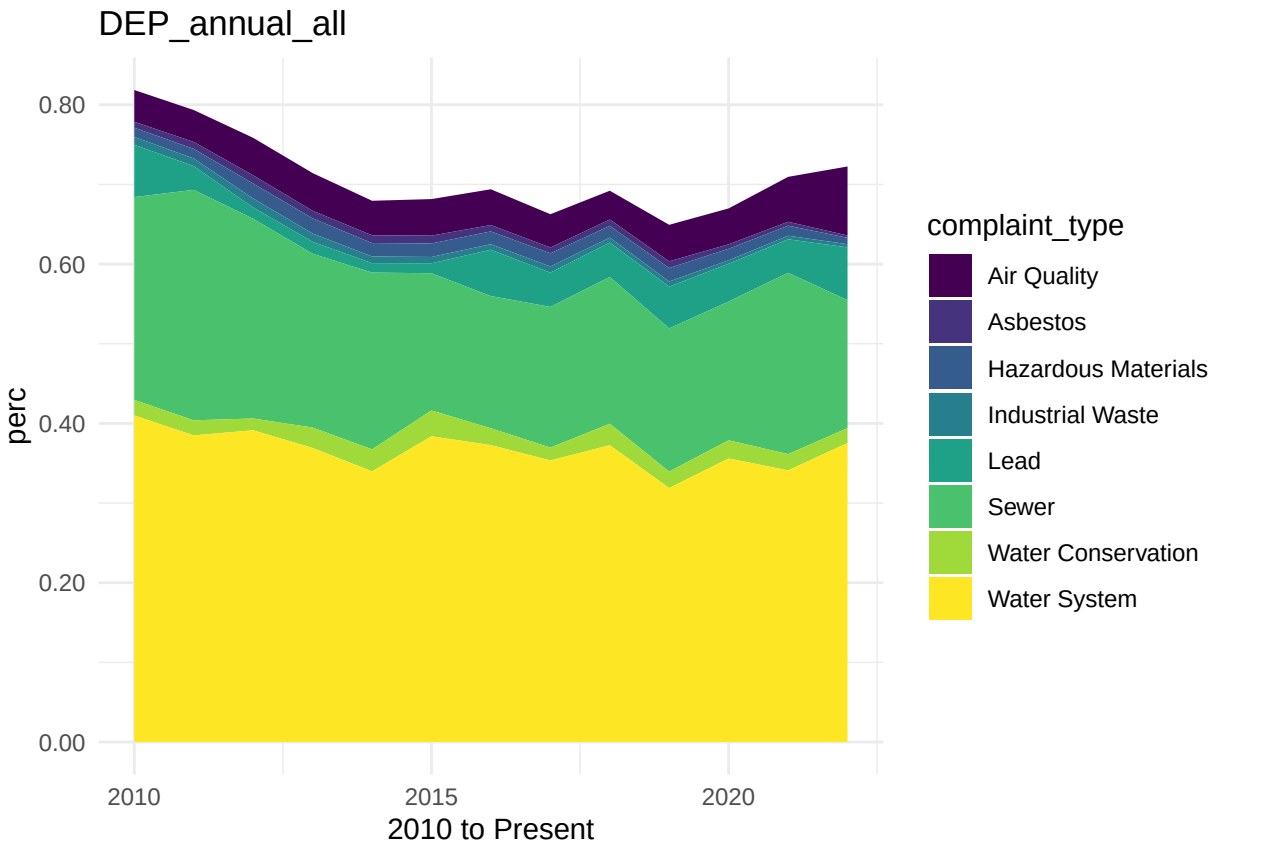

```
dep_filtered_annual_plot_perc
```



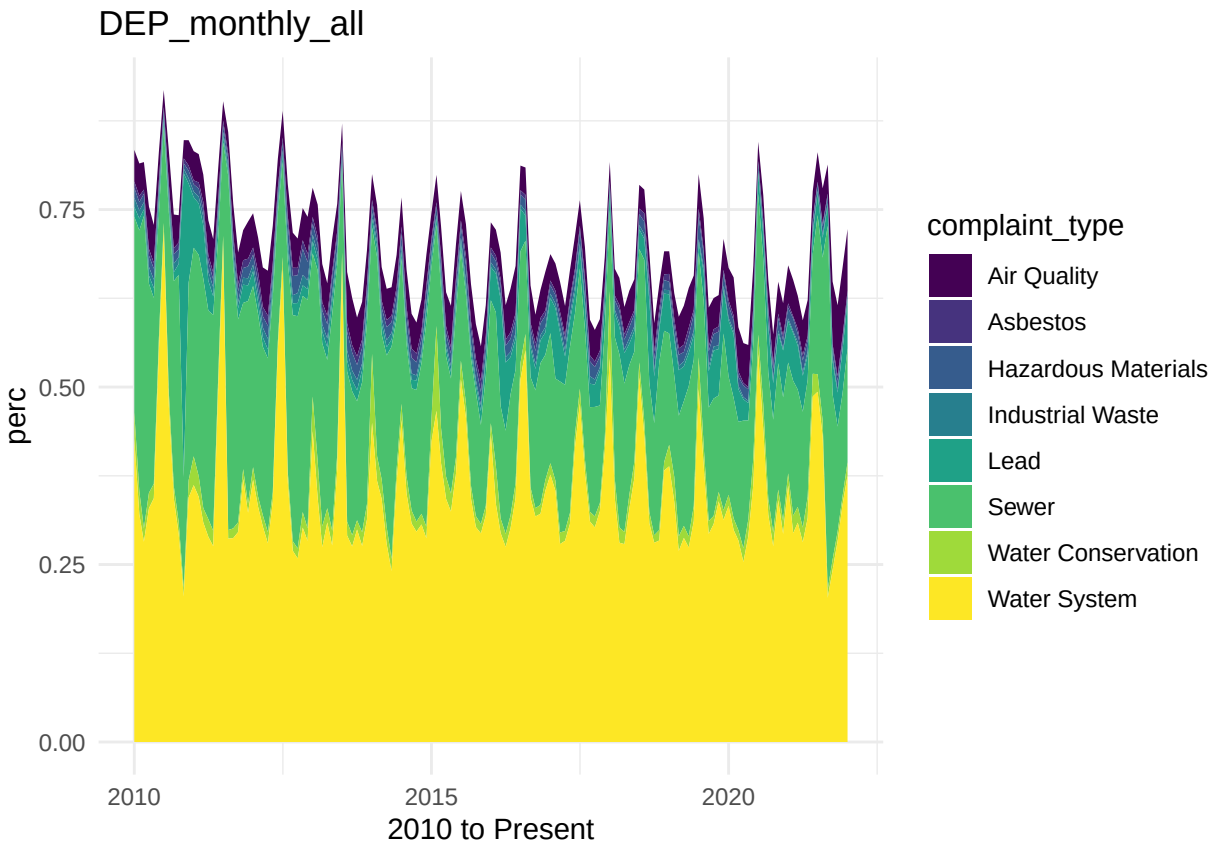

```
dep_filtered_monthly_plot_perc
```




```
dep_annual_plot_perc
```




```
dep_monthly_plot_perc
```



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 in Data:

```
# 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/De
##   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)
```

```
# 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 1000 to test out functions
dep_sample <- DEP_points[sample(nrow(DEP_points), 1000), ]
# spatial join to census tracts
ct_complaints <- ct %>%
  st_intersection(dep_sample) %>%
  group_by(ct2010,boro_name,complaint_type,year) %>%
  count()
```

Test out Sample

```
## 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.
```

```
test_map<-complaint_mapping(ct,test,"ct",years,view_complaints)
```

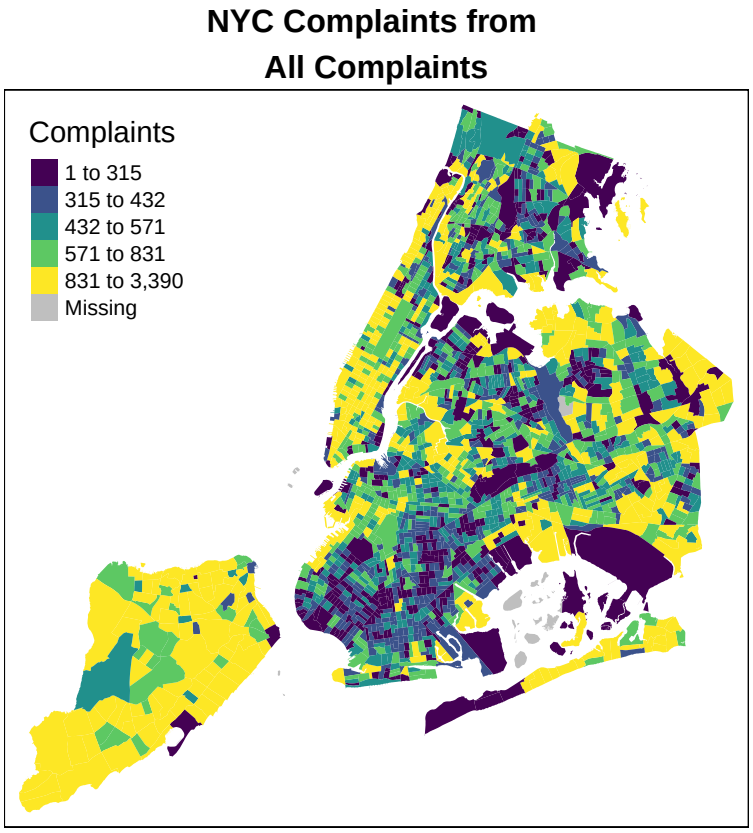
For the entire dataset:

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:

```
years = c(2020:2010)
view_complaints = complaints
## Spatial Joins. These take extremely long, so saving for optimizing for time.

# 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.
```

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



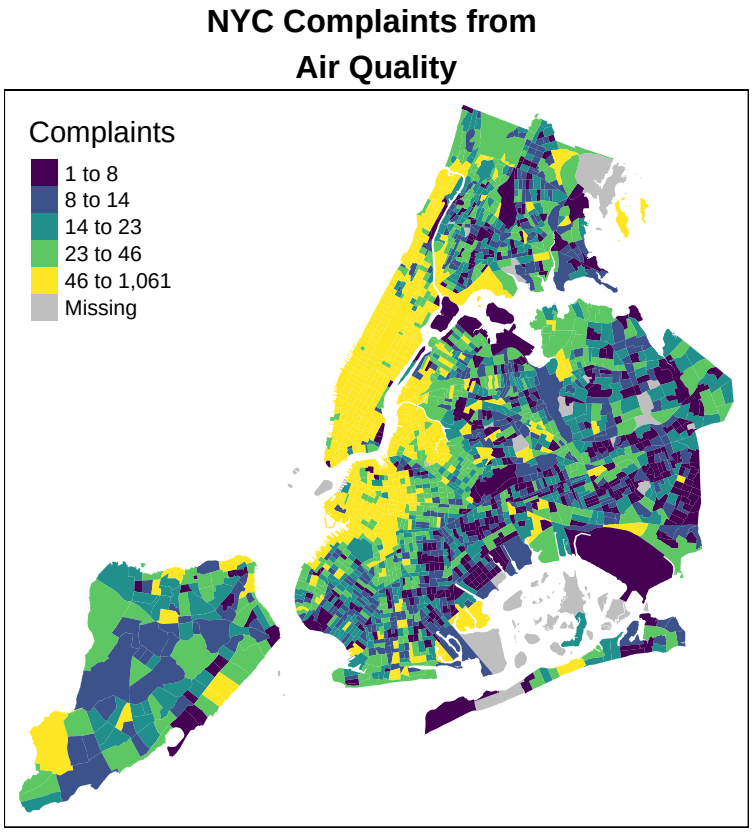
Looking at complaints by type:

```
## 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),3)

## Map over the complaints 8 times
map_complaints <- c("Air Quality",
                    "Sewer","Water System")

## Input "map" 8 times
map_ct <- rep("map",3)

## For now, put in all years, 8 times.
map_years <- rep(list(years),3)

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

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

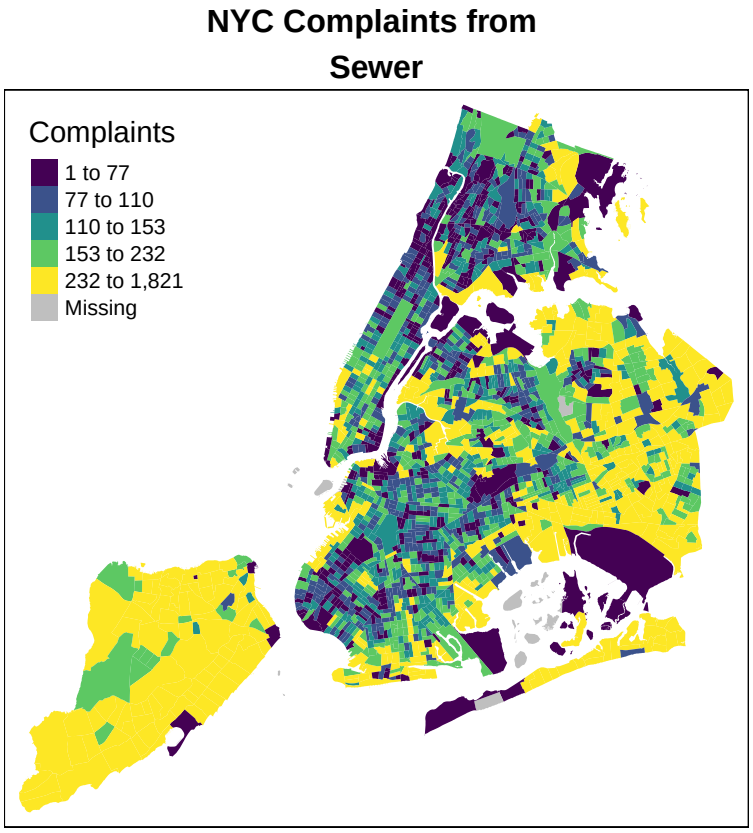
```
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'ct2010'. You can override using the `.groups` argument.
```

```
## Prepare inputs for plotting:
map_ct_shapes <- rep(list(ct),3)

input_ct_plot <- list(map_ct_shapes,filtered_ct_all, map_ct,map_years,map_complaints)


plots_ct <- pmap(input_ct_plot,complaint_mapping)
plots_ct
```

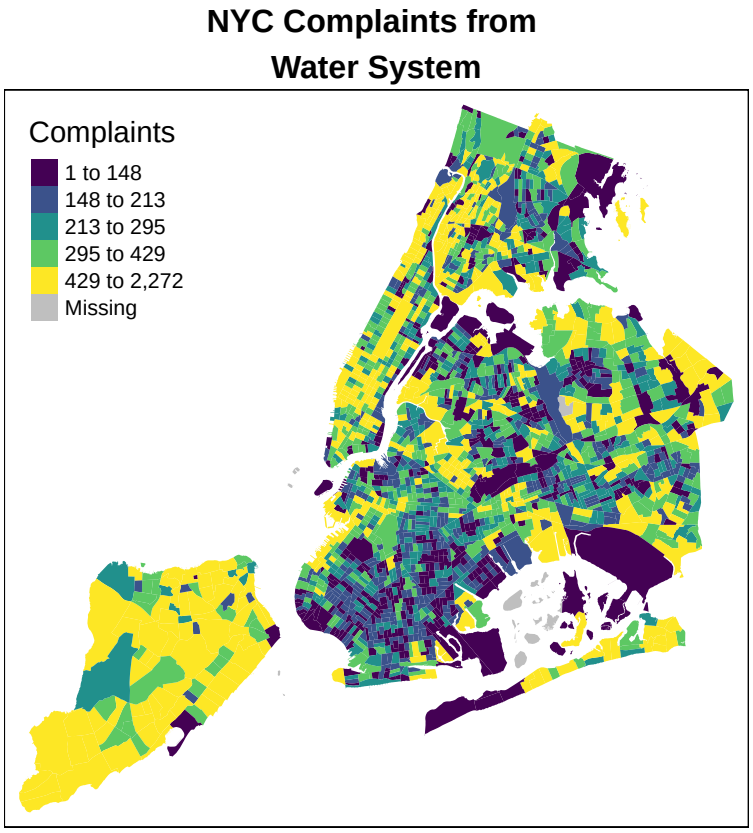
```
## [[1]]
```




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


NYC Complaints from Sewer
Complaints
1 to 77
77 to 110
110 to 153
153 to 232
232 to 1,821
Missing

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




```
plotnames = imap(complaints, ~paste0("../4_Outputs/",., ".png")) %>%
     flatten()
plotnames
```

```
## [[1]]
## [1] "../4_Outputs/Air Quality.png"
##
## [[2]]
## [1] "../4_Outputs/Lead.png"
##
## [[3]]
## [1] "../4_Outputs/Water Conservation.png"
##
## [[4]]
## [1] "../4_Outputs/Industrial Waste.png"
##
## [[5]]
## [1] "../4_Outputs/Sewer.png"
##
## [[6]]
## [1] "../4_Outputs/Water System.png"
##
## [[7]]
## [1] "../4_Outputs/Asbestos.png"
##
## [[8]]
## [1] "../4_Outputs/Hazardous Materials.png"
```

```
# pwalk(list(plots_ct,plotnames), tmap_save)
```

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.

Normalization Process:

Read in Census Data:

```
census_api_key("4878f9292ac0c11cddc6fabc8cd8530c4d0e12f0")
```

```
## To install your API key for use in future sessions, run this function with `install = TRUE`.
```

```
v2019 <- load_variables(2019, "acs5", cache = TRUE)

demographics <- v2019 %>%
  filter(grepl("B02001",name))

demographics <- c(white = "B02001_002",
                  total = "B02001_001",
                  income ="B19113_001",
                  language_bad = "B16004_045",
                  language_not_well = "B16004_044",
                  native_born = "B05002_002")
Census_2019 <-
  get_acs(
    state = 36,
    county = c("061", "047", "081", "005", "085"),
    geography = "tract",
    variables = demographics,
    year = 2019
  ) %>%
  transmute(
    cnty = str_sub(GEOID, 3, 5),
    tract = as.integer(str_sub(GEOID, 6, 12)),
```

```
    measure = estimate,
    boro_name =
      case_when(
        cnty == "061" ~ "Manhattan",
        cnty == "047" ~ "Brooklyn",
        cnty == "081" ~ "Queens",
        cnty == "005" ~ "Bronx",
        cnty == "085" ~ "Staten Island"
      ),
    ctlabel = as.character(tract / 100),
    variable = variable)
```

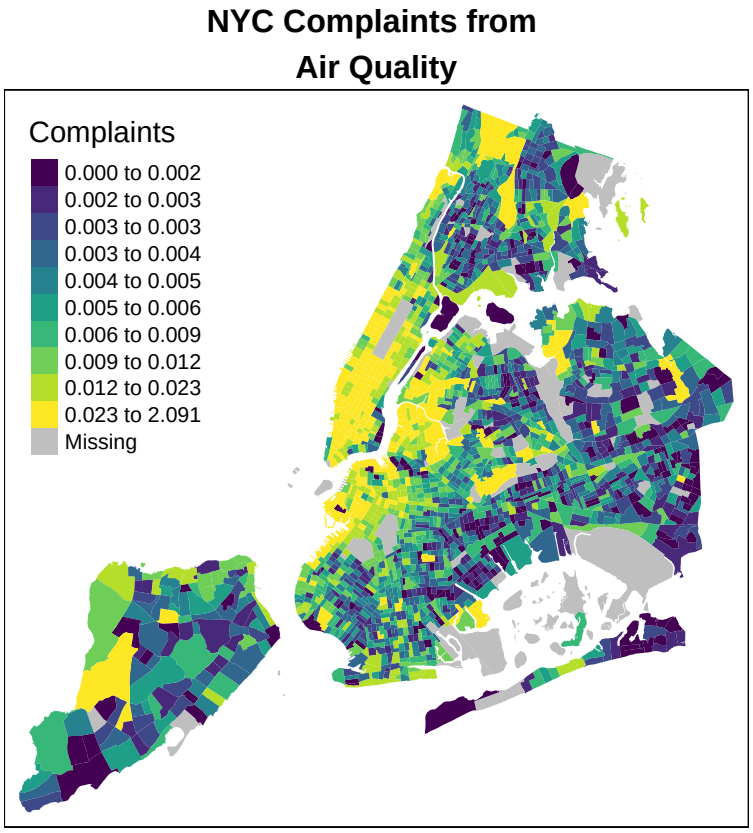
```
## Getting data from the 2015-2019 5-year ACS
```

```
Population_2019 <- Census_2019 %>%
  filter(variable == 'total')

census_shapes <- ct %>%
    left_join(Population_2019, by = c("boro_name", "ctlabel"))
```

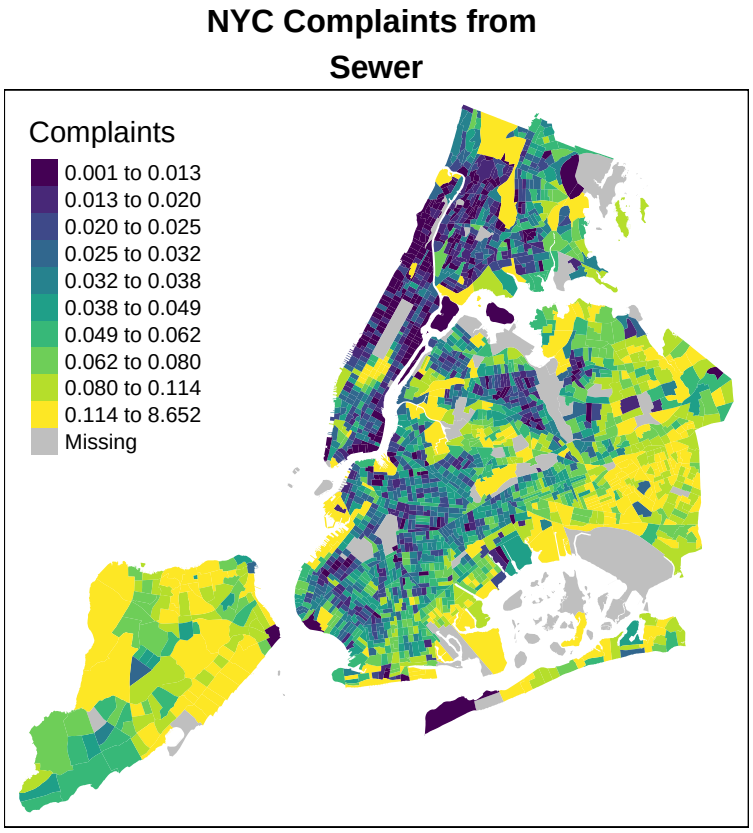
Plotting for all years, all complaints

```
shape_ct <- rep("shape",3)
input_ct_shapes <- list(map_ct_shapes,filtered_ct_all, shape_ct,map_years,map_complaints)
map_pop <- rep(list(Population_2019),3)
shapes_ct_all <- pmap(input_ct_shapes,complaint_mapping)
join_pop_inputs <- list(shapes_ct_all,map_pop)
ct_totalpop <- pmap(join_pop_inputs,popjoin)

percapita_inputs <-list(ct_totalpop,map_complaints)
ct_percapita_all <- pmap(percapita_inputs,plotting_percapita)
ct_percapita_all
```

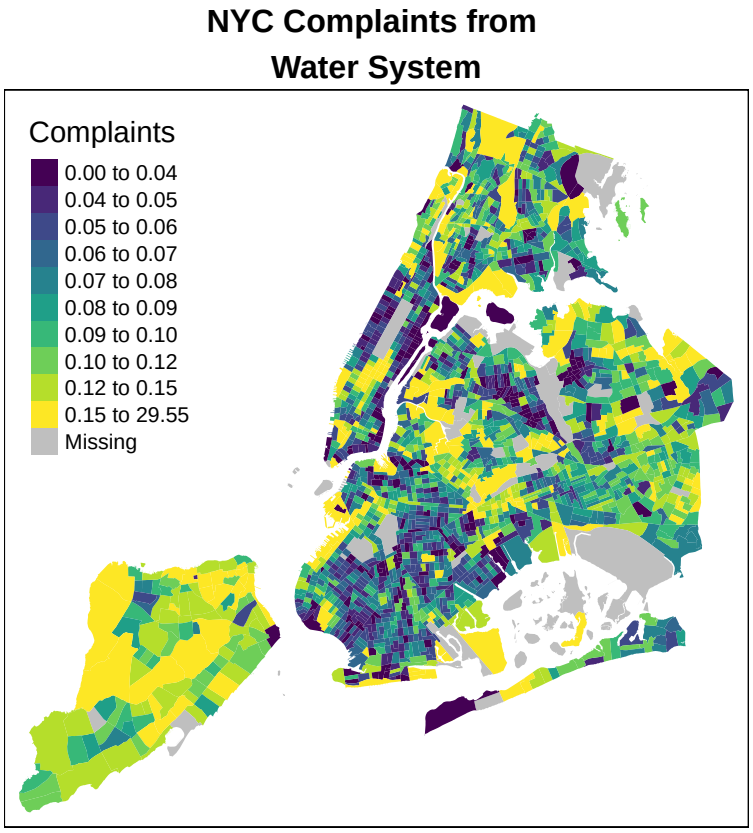
```
## [[1]]
```


NYC Complaints from
Air Quality
Complaints
0.000 to 0.002
0.002 to 0.003
0.003 to 0.003
0.003 to 0.004
0.004 to 0.005
0.005 to 0.006
0.006 to 0.009
0.009 to 0.012
0.012 to 0.023
0.023 to 2.091
Missing

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


NYC Complaints from Sewer
Complaints
0.001 to 0.013
0.013 to 0.020
0.020 to 0.025
0.025 to 0.032
0.032 to 0.038
0.038 to 0.049
0.049 to 0.062
0.062 to 0.080
0.080 to 0.114
0.114 to 8.652
Missing

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




```
norm_plotnames = imap(map_complaints, ~paste0("../4_Outputs/",., "_perCapita_quant_allyrs.png")) %>%
     flatten()

norm_plotnames
```

```
## [[1]]
## [1] "../4_Outputs/Air Quality_perCapita_quant_allyrs.png"
##
## [[2]]
## [1] "../4_Outputs/Sewer_perCapita_quant_allyrs.png"
##
## [[3]]
## [1] "../4_Outputs/Water System_perCapita_quant_allyrs.png"
```

```
pwalk(list(ct_percapita_all,norm_plotnames), tmap_save)
```

```
## Map saved to /Users/seanchew/Desktop/Bo_Assistantship/4_Outputs/Air Quality_perCapita_quant_allyrs.pn
## Resolution: 2110.23 by 2089.82 pixels
## Size: 7.034099 by 6.966066 inches (300 dpi)
## Map saved to /Users/seanchew/Desktop/Bo_Assistantship/4_Outputs/Sewer_perCapita_quant_allyrs.png
## Resolution: 2110.23 by 2089.82 pixels
## Size: 7.034099 by 6.966066 inches (300 dpi)
## Map saved to /Users/seanchew/Desktop/Bo_Assistantship/4_Outputs/Water System_perCapita_quant_allyrs.p
## Resolution: 2110.23 by 2089.82 pixels
```

```
## Size: 7.034099 by 6.966066 inches (300 dpi)
```