PS3

Ananya Sharma

1/29/2024

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2

## Warning: package 'tibble' was built under R version 4.1.2

## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'readr' was built under R version 4.1.2

## Warning: package 'purrr' was built under R version 4.1.2

## Warning: package 'dplyr' was built under R version 4.1.2

## Warning: package 'stringr' was built under R version 4.1.2

## Warning: package 'forcats' was built under R version 4.1.2

## Warning: package 'lubridate' was built under R version 4.1.2

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tidycensus)
```

```
## Warning: package 'tidycensus' was built under R version 4.1.2
```

```
library(ggplot2)
library(sf)
```

```
## Warning: package ’sf’ was built under R version 4.1.2
```

```
## Linking to GEOS 3.10.2, GDAL 3.4.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(tigris)
```

```
## Warning: package ’tigris’ was built under R version 4.1.2
```

```
## To enable caching of data, set ‘options(tigris_use_cache = TRUE)‘
## in your R script or .Rprofile.
```

```
df <- read.csv("parking_tickets_one_percent2.csv")
```

Part I. Cleaning the data and benchmarking

Q1. How many tickets were issued in the data in 2017? How many tickets does that imply were issued in the full data in 2017? How many tickets are issued each year according to the ProPublica article?

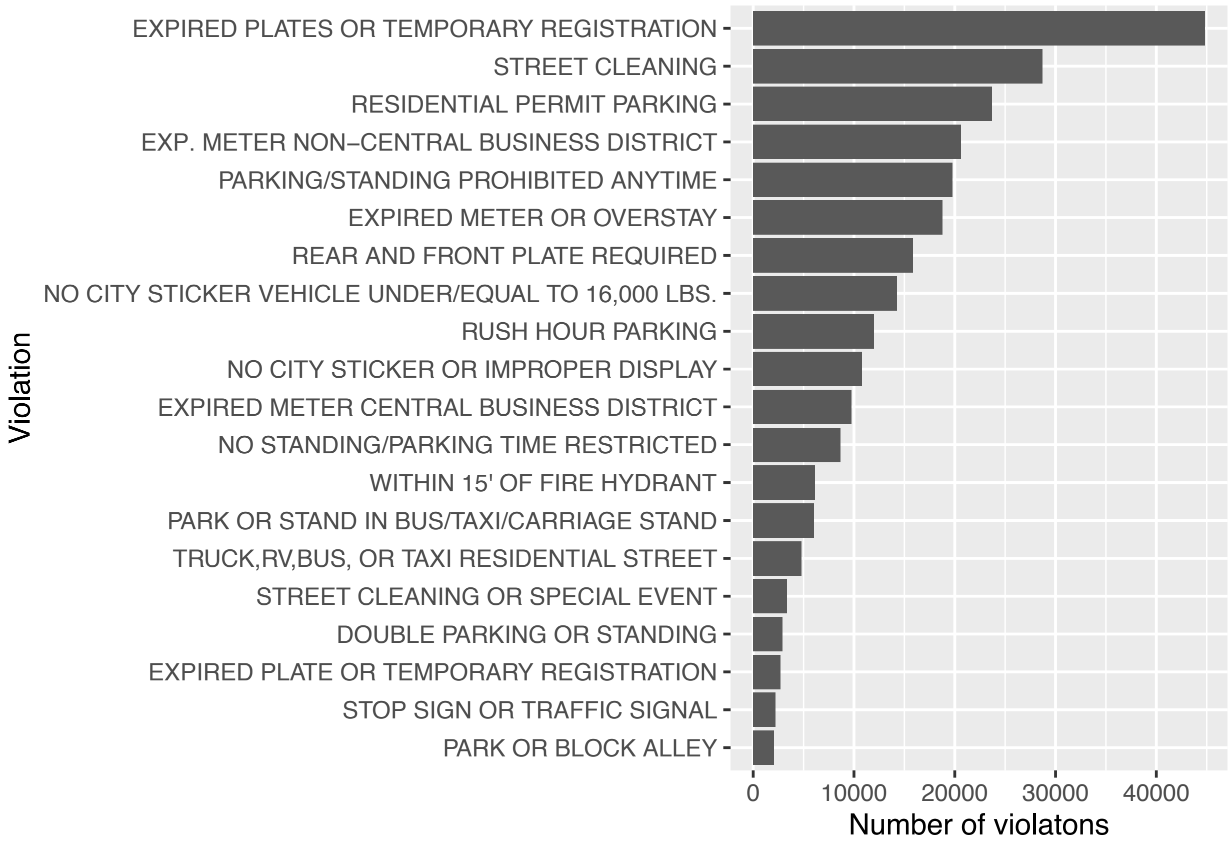
```
df %>%
filter(issue_date >= as_datetime("2017-01-01 00:00:00")) %>%
filter(issue_date < as_datetime("2018-01-01 00:00:00")) %>%
nrow()
```

```
## [1] 22364
```

Q2. In the whole dataset, what are the top 20 most frequent violation types? Make a bar graph to show the frequency of these ticket types.

```
library(ggplot2)

df %>%
count(violation_description) %>%
top_n(20, n) %>%
arrange(desc(n)) %>%
ggplot(aes(
y = reorder(violation_description, n),
x = n)) +
geom_col() +
labs(
x = "Number of violatons",
y = "Violation")
```



Part II. The data also contains information telling us what unit of city government issued each ticket, but this is only added as a code. We need to join with another dataset to get the actual names of the units.

Q1. For how many tickets is unit missing?

```
df %>%
select(unit) %>%
is.na() %>%
sum()
```

```
## [1] 29
```

Q2. Read in unit_key.csv. How many units are there?

```
library(readr)
df_units <- read_csv("unit_key-1.csv", skip = 2)
```

```
## New names:
## Rows: 385 Columns: 7
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (6): Reporting District...1, Department Name, Department Description, De... lgl
## (1): ...5
## i Use ‘spec()‘ to retrieve the full column specification for this data. i
## Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## * ‘Reporting District‘ -> ‘Reporting District...1‘
## * ‘Department Category‘ -> ‘Department Category...4‘
## * ‘‘ -> ‘...5‘
## * ‘Reporting District‘ -> ‘Reporting District...6‘
## * ‘Department Category‘ -> ‘Department Category...7‘
```

```
df_units <- df_units %>%
mutate(unit = as.numeric(`Reporting District...1`))
```

```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘unit = as.numeric(‘Reporting District...1‘)‘.
## Caused by warning:
## ! NAs introduced by coercion
```

```
df_units %>%
select(unit) %>%
unique() %>%
nrow()
```

```
## [1] 375
```

Q3. Use joins to answer the following questions. Use unit as the key column to do the joins. - How many rows in the tickets data have a match in the unit table? - How many rows are unmatched? - How many rows in the unit table have a match in the tickets data? - How many do not?

```
nrow(semi_join(df, df_units, by = "unit"))
```

```
## [1] 287458
```

```
nrow(anti_join(df, df_units, by = "unit"))
```

```
## [1] 0
```

```
nrow(semi_join(df_units, df, by = "unit"))
```

```
## [1] 139
```

```
nrow(anti_join(df_units, df, by = "unit"))
```

```
## [1] 246
```

Interpretation: All of the rows in tickets data have a match in the unit table and 0 are unmatched. 139 rows in the unit table have a match in the tickets data. 246 do not.

Q4. What is the name of the department which issues more tickets – Department of Finance or Chicago Police? Within Chicago Police, what are the top 5 department descriptions that are issuing the most tickets? Be careful what you group by here and avoid columns with ambiguities.

```r
library(tidyr)

df_unit_joined <- left_join(df, df_units %>% drop_na(unit), by = "unit")
df_unit_joined %>%
filter(`Department Name` %in% c("CPD","CPD-Other","CPD-Airport")) %>%
nrow()
```

```
## [1] 127078
```

```r
df_unit_joined %>%
filter(`Department Name` == "DOF") %>%
nrow()
```

```
## [1] 143909
```

Therefore, DOF has more tickets issued.

```r
df_unit_joined %>%
filter(`Department Name` %in% c("CPD","CPD-Other","CPD-Airport")) %>%
group_by(`Department Description`) %>%
summarise(n = n()) %>%
top_n(5, n) %>%
arrange(desc(n))
```

```
## # A tibble: 5 x 2
##   `Department Description`     n
##   <chr>                    <int>
## 1 1160 N. Larrabee          9478
## 2 6464 N. Clark             7946
## 3 OEMC                      7374
## 4 3315 W. Ogden             5469
## 5 5555 W. Grand             5464
```

Part III - Replicate the key finding in the Propublica by ranking ZIPs by the number of unpaid tickets (i.e. ticket with no payment) per resident by ZIP in five steps

Q1. Using library(tidycensus), download 2014 data from the American Community Survey (ACS) by ZIP for Chicago with total population, total black population and median household income. (Hint: the "ZCTA" geography aggregation would return all zip codes; Use the load_variable function to help find the codes for the necessary variables, or online, eg: https://api.census.gov/data/2014/acs /acs5/groups/. the chi_zips.csv contains all the zipcodes needed)

```r
library(tidycensus)
df_zips <- read_csv("chi_zips.csv")
```

```
## Rows: 68 Columns: 1
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1): ZIP
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
zta_vars <- load_variables(2014, "acs5", cache = TRUE) %>%
filter(concept %in% c(
"MEDIAN HOUSEHOLD INCOME IN THE PAST 12 MONTHS (IN 2014 INFLATION-ADJUSTED DOLLARS)",
"UNWEIGHTED SAMPLE COUNT OF THE POPULATION",
"RACE"
))
chicago_df <- get_acs(geography = "zcta",
variables = c(
med_income = "B19013_001",
population_black = "B02001_003",
population = "B01001_001"
),
year = 2014,
zcta = df_zips$ZIP,
state = "IL"
) %>%
select(-NAME, -moe) %>%
pivot_wider(id_cols = GEOID, names_from = variable, values_from = estimate) %>%
mutate(share_black = population_black/population)
```

```
## Getting data from the 2010-2014 5-year ACS
```

```
## Warning: * You have not set a Census API key. Users without a key are limited to 500
## queries per day and may experience performance limitations.
## i For best results, get a Census API key at
## http://api.census.gov/data/key_signup.html and then supply the key to the
## ‘census_api_key()‘ function to use it throughout your tidycensus session.
## This warning is displayed once per session.
```

```
chicago_df
```

```
## # A tibble: 67 x 5
##    GEOID population population_black med_income share_black
##    <chr>      <dbl>            <dbl>      <dbl>       <dbl>
##  1 60007      33830              213      68559     0.00630
##  2 60018      29027              423      54817     0.0146
##  3 60068      37511              344      87626     0.00917
##  4 60106      20150              736      60584     0.0365
##  5 60131      18103              141      57269     0.00779
##  6 60176      11842              142      45646     0.0120
##  7 60601      10894             1115     101250     0.102
##  8 60602       1429               24      73971     0.0168
##  9 60603       1002               10     111125     0.00998
## 10 60604        419               13     155750     0.0310
## # i 57 more rows
```

Q2. Calculate the sum of the unpaid counts of the ticket data by zip code.

```
library(stringr)

df %>%
mutate(GEOID = str_extract(zipcode, "[0-9]{5}")) %>%
group_by(GEOID) %>%
summarise(unpaid = sum(total_payments == 0))
```

```
## # A tibble: 5,288 x 2
##    GEOID unpaid
##    <chr>  <int>
##  1 00000      6
##  2 00006      0
##  3 00100      0
##  4 00130      0
##  5 00210      0
##  6 00212      1
##  7 00300      1
##  8 00317      0
##  9 00330      0
## 10 00453      1
## # i 5,278 more rows
```

Q3. Join this with the data from you got from the previous step (remember to clean the tickets data to match the census data format!)

```
df <- df %>%
mutate(GEOID = str_extract(zipcode, "[0-9]{5}")) %>%
left_join(chicago_df, by = "GEOID")
df
```

```
##      ticket_number           issue_date       violation_location
## 1         51482901 2007-01-01T01:25:00Z          5762 N AVONDALE
## 2         50681501 2007-01-01T01:51:00Z          2724 W FARRAGUT
## 3         51579701 2007-01-01T02:22:00Z             1748 W ESTES
## 4         51262201 2007-01-01T02:35:00Z          4756 N SHERIDAN
## 5         51898001 2007-01-01T03:50:00Z          7134 S CAMPBELL
## 6         50681401 2007-01-01T04:10:00Z           2227 W FOSTERT
## 7         51226001 2007-01-01T04:36:00Z           1411 S KOSTNER
## 8         51376701 2007-01-01T05:40:00Z           6954 S ASHLAND
## 9         51262301 2007-01-01T06:00:00Z         2630 N CANNON DR
## 10        51226201 2007-01-01T08:35:00Z       4401 W 28TH STREET
## 11        51367201 2007-01-01T08:48:00Z          1936 N RIDGEWAY
## 12        51574901 2007-01-01T09:40:00Z         6252 S HERMITAGE
## 13        51536501 2007-01-01T10:48:00Z              3630 W EDDY
## 14        52432501 2007-01-01T10:50:00Z         3240 W ROOSEVELT
## 15        51262101 2007-01-01T10:51:00Z          4325 N BROADWAY
## 16        53558001 2007-01-01T12:12:00Z            175 E PEARSON
## 17        51536001 2007-01-01T12:20:00Z       4838 N SPRINGFIELD
## 18        51492401 2007-01-01T12:51:00Z                60 W ERIE
## 19        50482401 2007-01-01T14:05:00Z            10000 W OHARE
## 20        51224901 2007-01-01T14:20:00Z           2139 W COULTER
## 21        51522301 2007-01-01T14:30:00Z        3159 W 47TH PLACE
## 22        51380001 2007-01-01T14:35:00Z           4013 W MADISON
## 23        51262401 2007-01-01T15:00:00Z       3909 N SHERIDAN RD
## 24        51224001 2007-01-01T15:49:00Z             2755 W OGDEN
## 25        51574801 2007-01-01T15:53:00Z              6302 S WOOD
## 26        51551301 2007-01-01T17:04:00Z          7047 S ROCKWELL
## 27        51638401 2007-01-01T17:06:00Z             1136 W PRATT
## 28        51496301 2007-01-01T17:25:00Z           3519 W LEMOYNE
## 29        51575001 2007-01-01T17:40:00Z        5724 S WINCHESTER
## 30        51549401 2007-01-01T17:55:00Z            5050 S KEDZIE
## 31        51367801 2007-01-01T18:14:00Z             2150 N MAJOR
## 32        51551501 2007-01-01T18:15:00Z            7601 S CICERO
## 33        53119201 2007-01-01T18:42:00Z              3009 W 19TH
## 34        51484301 2007-01-01T18:45:00Z           3238 N PACIFIC
## 35        51579301 2007-01-01T19:20:00Z             1641 W CHASE
## 36        51580901 2007-01-01T19:40:00Z           164 W ILLINOIS
## 37        51581001 2007-01-01T20:09:00Z            701 W WEBSTER
## 38        51427201 2007-01-01T20:22:00Z               920 W LAKE
## 39        51507001 2007-01-01T20:25:00Z         4805 S VINCENNES
## 40        51320701 2007-01-01T21:06:00Z            1501 S WABASH
## 41        51507201 2007-01-01T21:44:00Z          5822 S MICHIGAN
## 42        51377201 2007-01-01T22:12:00Z           6854 S ASHLAND
## 43        51532601 2007-01-01T22:13:00Z          4737 N ST LOUIS
## 44        51496201 2007-01-01T22:50:00Z             1538 W NORTH
## 45        51067501 2007-01-01T23:19:00Z           3105 N KENMORE
## 46        51580801 2007-01-01T23:25:00Z          402 W BLACKHAWK
## 47        51540701 2007-01-01T23:56:00Z           6220 S KIMBARK
## 48        51511801 2007-01-02T00:51:00Z           324 E PERSHING
## 49        51543801 2007-01-02T01:01:00Z           8050 S KENWOOD
## 50        51321401 2007-01-02T01:05:00Z          541 S JEFFERSON
## 51        51357101 2007-01-02T01:36:00Z             7719 S CLYDE
## 52        51262801 2007-01-02T01:50:00Z             913 W CULLOM
## 53        51308401 2007-01-02T01:55:00Z            4429 S DREXEL
## 54        51425201 2007-01-02T03:40:00Z          417 N CARPENTER
## 55        51535601 2007-01-02T05:33:00Z          3222 N RICHMOND
## 56        51532101 2007-01-02T05:34:00Z         3240 N WASHTENAW
## 57        51097301 2007-01-02T06:05:00Z        200 E SAINT CLAIR
## 58        51438301 2007-01-02T07:05:00Z            444 E ONTARIO
## 59        50681701 2007-01-02T07:10:00Z            2721 W FOSTER
## 60        51225401 2007-01-02T07:45:00Z              2257 S TROY
## 61        51495201 2007-01-02T07:55:00Z             1442 N HOMAN
## 62        51483301 2007-01-02T08:22:00Z          5015 W MONTROSE
## 63        51442801 2007-01-02T08:29:00Z      8247 S STONY ISLAND
## 64        51145701 2007-01-02T08:40:00Z         5738 S FAIRFIELD
## 65        51148501 2007-01-02T08:47:00Z         6340 S WASHTENAW
## 66      9057403101 2007-01-02T09:02:00Z         2156 W EVERGREEN
## 67      9065923101 2007-01-02T09:05:00Z         8044 S LAFAYETTE
## 68        51401001 2007-01-02T09:09:00Z           72 E BENTON PL
## 69        51204301 2007-01-02T09:10:00Z           2736 W GREGORY
## 70        51581201 2007-01-02T09:10:00Z          914 N CAMBRIDGE
## 71      9058168901 2007-01-02T09:11:00Z          3629 N BROADWAY
## 72        51581101 2007-01-02T09:11:00Z              1916 N MAUD
## 73        51400901 2007-01-02T09:16:00Z            160 W LAKE ST
## 74      9056607701 2007-01-02T09:16:00Z            5643 S KOLMAR
## 75        51507501 2007-01-02T09:30:00Z           5139 S PRAIRIE
## 76        51521101 2007-01-02T09:43:00Z           3352 S LEAVITT
## 77      9053627501 2007-01-02T09:45:00Z        5212 S BLACKSTONE
## 78        51535701 2007-01-02T09:57:00Z         4009 N FRANCISCO
## 79        51297201 2007-01-02T10:00:00Z             305 S KEDZIE
## 3526      60315 0.734815629
## 3527         NA          NA
## 3528         NA          NA
## 3529         NA          NA
## 3530         NA          NA
## 3531      55324 0.016507337
## 3532      34153 0.971857732
## 3533      48786 0.026359371
## 3534         NA          NA
## 3535         NA          NA
## 3536      38686 0.154820823
## 3537         NA          NA
## 3538      41882 0.039997159
## 3539      32747 0.918705686
## 3540      54400 0.063476797
## 3541      43554 0.253916737
## 3542      56763 0.024431051
## 3543         NA          NA
## 3544         NA          NA
## 3545      38825 0.544923505
## 3546      56763 0.024431051
## 3547      34153 0.971857732
## 3548      38825 0.544923505
## 3549      62859 0.009652848
## 3550         NA          NA
## 3551         NA          NA
## 3552      50237 0.074507580
## 3553         NA          NA
## 3554      50237 0.074507580
## 3555         NA          NA
## 3556         NA          NA
## 3557         NA          NA
## 3558      32494 0.635695297
## 3559      26997 0.774047614
## 3560      38196 0.014492595
## 3561      38196 0.014492595
## 3562      34153 0.971857732
## 3563      40835 0.155920735
## 3564         NA          NA
## 3565      26299 0.940985589
## 3566         NA          NA
## 3567         NA          NA
## 3568         NA          NA
## 3569      53394 0.085176219
## 3570         NA          NA
## 3571         NA          NA
##  [ reached 'max' / getOption("max.print") -- omitted 283887 rows ]
```

Q4. Replicate the key finding in the Propublica by ranking ZIPs by the number of unpaid tickets per resident by ZIP. What are the names of the three neighborhoods with the most unpaid tickets?

```r
df_final <- df %>%
mutate(GEOID = str_extract(zipcode, "[6][0-9]{4}")) %>%
group_by(GEOID) %>%
summarise(sum_unpaid = sum((total_payments == 0))) %>%
ungroup() %>%
inner_join(chicago_df, by = "GEOID") %>%
mutate(ratio_unpaid = sum_unpaid/population)
df_final %>%
top_n(3, ratio_unpaid)
```

```
## # A tibble: 3 x 7
##   GEOID sum_unpaid population population_black med_income share_black
##   <chr>      <int>      <dbl>            <dbl>      <dbl>       <dbl>
## 1 60604         25        419               13     155750      0.0310
## 2 60621       1156      32619            31146      19190      0.955
## 3 60644       1721      49615            46687      26299      0.941
## # i 1 more variable: ratio_unpaid <dbl>
```

Q5. Make #3 into a map

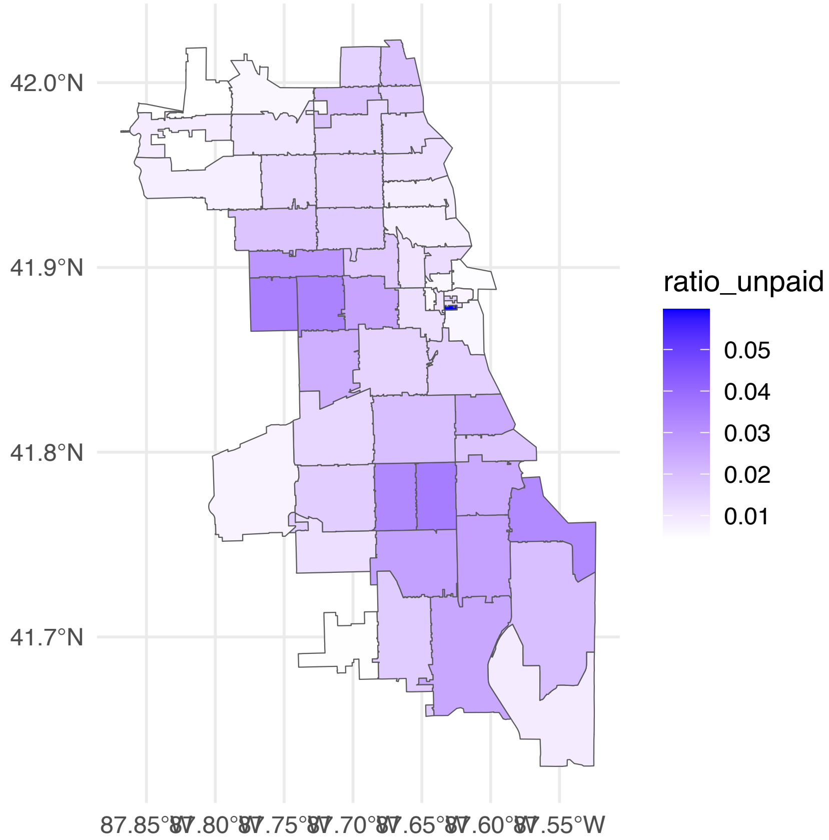
```
library(sf)
library(tigris)
il_zctas <- zctas(starts_with = "606", class = "sf")
```

```
## Retrieving data for the year 2021
```

```
## ZCTAs can take several minutes to download.  To cache the data and avoid re-downloading in future R s
```

```
##   |                                                                              |
```

```
df_sf <- left_join(il_zctas, df_final, join_by("GEOID20" == "GEOID"))
ggplot(data = df_sf) +
geom_sf(aes(fill = ratio_unpaid)) +
scale_fill_continuous(low="white", high = "blue") +
theme_minimal()
```



Part IV - Understanding the structure of the data

Q1. Most violation types double in price if unpaid.Does this hold for all violations? If not, find all violations with at least 100 citations that do not double. How much does each ticket increase if unpaid?

```
df %>%
group_by(violation_description) %>%
summarise(
n = n(),
fine_level1_amount_mean = mean(fine_level1_amount),
fine_level2_amount_mean = mean(fine_level2_amount)) %>%
ungroup() %>%
filter(n >= 100) %>%
mutate(ratio = fine_level2_amount_mean/fine_level1_amount_mean) %>%
filter(ratio != 2) %>%
arrange(desc(ratio))
```

```
## # A tibble: 7 x 5
##   violation_description          n fine_level1_amount_m~1 fine_level2_amount_m~2
##   <chr>                      <int>                  <dbl>                  <dbl>
## 1 PARK/STAND ON BICYCLE PATH   236                   143.                   279.
## 2 NO CITY STICKER VEHICLE O~   131                   500                    955.
## 3 BLOCK ACCESS/ALLEY/DRIVEW~  1579                   142.                   267.
## 4 PARK OR BLOCK ALLEY         2050                   150                    260.
## 5 DISABLED PARKING ZONE       2034                   217.                   358.
## 6 OBSTRUCTED OR IMPROPERLY ~   271                   156.                   226.
## 7 SMOKED/TINTED WINDOWS PAR~  1697                   151.                   210.
## # i abbreviated names: 1: fine_level1_amount_mean, 2: fine_level2_amount_mean
## # i 1 more variable: ratio <dbl>
```

Q2. Are any violation descriptions associated with multiple violation codes? If so, which descriptions have multiple associated codes and how many tickets are there in each description-code pair?

```
df %>%
count(violation_description, violation_code) %>%
group_by(violation_description) %>%
filter(n()>1) %>%
ungroup()
```

```
## # A tibble: 10 x 3
##    violation_description               violation_code     n
##    <chr>                               <chr>          <int>
##  1 3-7 AM SNOW ROUTE                   0964060          827
##  2 3-7 AM SNOW ROUTE                   0964060B          12
##  3 CURB LOADING ZONE                   0964160A           1
##  4 CURB LOADING ZONE                   0964160B        1204
##  5 INDUSTRIAL PERMIT PARKING           0964091          117
##  6 INDUSTRIAL PERMIT PARKING           0964091B           3
##  7 NO CITY STICKER OR IMPROPER DISPLAY 0964125        10758
##  8 NO CITY STICKER OR IMPROPER DISPLAY 0976170           15
##  9 SPECIAL EVENTS RESTRICTION          0964041          245
## 10 SPECIAL EVENTS RESTRICTION          0964041B         217
```

Q3. Are any violation codes associated with multiple violation descriptions? If so, which codes have multiple associated descriptions and how many tickets are there in each description-code pair?

```
df %>%
count(violation_description, violation_code) %>%
group_by(violation_code) %>%
filter(n()>1) %>%
ungroup()
```

```
## # A tibble: 16 x 3
##    violation_description                         violation_code     n
##    <chr>                                         <chr>          <int>
##  1 EXPIRED PLATE OR TEMPORARY REGISTRATION       0976160B        2720
##  2 HAZARDOUS DILAPIDATED VEHICLE                 0980110B         148
##  3 HAZARDOUS DILAPITATED VEHICLE                 0980110B         298
##  4 MISSING/NONCOMPLIANT FRONT AND/OR REAR PLATE 0976160A        1024
##  5 OUTSIDE METERED SPACE                         0964200B          63
##  6 PARK OUTSIDE METERED SPACE                    0964200B         278
##  7 REAR AND FRONT PLATE REQUIRED                 0976160A       15829
##  8 REAR PLATE REQUIRED MOTORCYCLE/TRAILER        0976160B         352
##  9 SNOW ROUTE: 2' OF SNOW OR MORE                0964070           20
## 10 SNOW ROUTE: 2'' OF SNOW OR MORE               0964070          144
## 11 SPECIAL EVENTS RESTRICTION                    0964041B         217
## 12 STREET CLEANING                               0964040B       28712
## 13 STREET CLEANING OR SPECIAL EVENT              0964040B        3370
## 14 Special Events                               0964041B          25
## 15 TRUCK OR SEMI-TRAILER PROHIBITED             0964170D         145
## 16 TRUCK TRAILOR/SEMI/TRAILER PROHIBITED        0964170D         157
```

Q4. Review the 50 most common violation descriptions. Do any of them seem to be redundant? If so, can you find a case where what looks like a redundancy actually reflects the creation of a new violation code?

```
df %>%
count(violation_description) %>%
top_n(50, n) %>%
arrange(violation_description)
```

```
##                                violation_description     n
## 1                                    20’OF CROSSWALK   393
## 2                                  3-7 AM SNOW ROUTE   839
## 3            ABANDONED VEH. FOR 7 DAYS OR INOPERABLE  1104
## 4               BLOCK ACCESS/ALLEY/DRIVEWAY/FIRELANE  1579
## 5                                  CURB LOADING ZONE  1205
## 6                                  DISABLED CURB CUT   436
## 7                              DISABLED PARKING ZONE  2034
## 8                         DOUBLE PARKING OR STANDING  2904
## 9           EXP. METER NON-CENTRAL BUSINESS DISTRICT 20600
## 10           EXPIRED METER CENTRAL BUSINESS DISTRICT  9736
## 11                         EXPIRED METER OR OVERSTAY 18756
## 12           EXPIRED PLATE OR TEMPORARY REGISTRATION  2720
## 13          EXPIRED PLATES OR TEMPORARY REGISTRATION 44811
## 14                     HAZARDOUS DILAPITATED VEHICLE   298
## 15                  IMPROPER DISPLAY OF CITY STICKER   399
## 16      MISSING/NONCOMPLIANT FRONT AND/OR REAR PLATE  1024
## 17               NO CITY STICKER OR IMPROPER DISPLAY 10773
## 18 NO CITY STICKER VEHICLE UNDER/EQUAL TO 16,000 LBS. 14246
## 19                            NO PARK IN PRIVATE LOT   378
## 20               NO STANDING/PARKING TIME RESTRICTED  8640
## 21                             NONCOMPLIANT PLATE(S)  1920
## 22                                  OBSTRUCT ROADWAY  1577
## 23           OBSTRUCTED OR IMPROPERLY TINTED WINDOWS   271
## 24                                        PARK ALLEY   998
## 25                               PARK OR BLOCK ALLEY  2050
## 26          PARK OR STAND IN BUS/TAXI/CARRIAGE STAND  6004
## 27                PARK OR STAND IN VIADUCT/UNDERPASS   247
## 28                        PARK OR STAND ON CROSSWALK  1953
## 29                          PARK OR STAND ON PARKWAY   495
## 30                         PARK OR STAND ON SIDEWALK  1036
## 31                        PARK OUTSIDE METERED SPACE   278
## 32  PARK VEHICLE SOLE PURPOSE OF DISPLAYING FOR SALE   664
## 33               PARKING/STANDING PROHIBITED ANYTIME 19753
## 34                     REAR AND FRONT PLATE REQUIRED 15829
## 35            REAR PLATE REQUIRED MOTORCYCLE/TRAILER   352
## 36                        RESIDENTIAL PERMIT PARKING 23683
## 37                                 RUSH HOUR PARKING 11965
## 38                             SAFETY BELTS REQUIRED   981
## 39             SMOKED/TINTED WINDOWS PARKED/STANDING  1697
## 40                        SPECIAL EVENTS RESTRICTION   462
## 41              STAND, PARK, OR OTHER USE OF BUS LANE  1233
## 42                        STOP SIGN OR TRAFFIC SIGNAL  2191
## 43                                    STREET CLEANING 28712
## 44                   STREET CLEANING OR SPECIAL EVENT  3370
## 45              TRUCK,MOTOR HOME, BUS BUSINESS STREET   456
## 46           TRUCK,RV,BUS, OR TAXI RESIDENTIAL STREET  4789
## 47              TWO HEAD LAMPS REQUIRED VISIBLE 1000’   443
## 48                WINDOWS MISSING OR CRACKED BEYOND 6   576
## 49                         WITHIN 15’ OF FIRE HYDRANT  6104
## 50                  WRONG DIRECTION OR 12’’ FROM CURB  1111
```

There are a few matching/redundant ones - - “BLOCK ACCESS/ALLEY/DRIVEWAY/FIRELANE” - “PARK ALLEY” - “PARK OR BLOCK ALLEY” - “SPECIAL EVENTS RESTRICTION” - “STREET CLEANING” - “STREET CLEANING OR SPECIAL EVENT” - “EXPIRED PLATE OR TEMPORARY REGISTRATION” - “EXPIRED PLATES OR TEMPORARY REGISTRATION” - “EXPIRED METER OR OVERSTAY” - “EXPIRED METER CENTRAL BUSINESS DISTRICT” - “EXP. METER NON-CENTRAL BUSINESS DISTRICT”

```r
df %>%
filter(violation_description %in% c(
"EXPIRED METER OR OVERSTAY",
"EXPIRED METER CENTRAL BUSINESS DISTRICT",
"EXP. METER NON-CENTRAL BUSINESS DISTRICT")
) %>%
count(year(issue_date), violation_code, violation_description)
```

```
##    year(issue_date) violation_code                    violation_description
## 1              2007       0964190A EXP. METER NON-CENTRAL BUSINESS DISTRICT
## 2              2007       0964190B  EXPIRED METER CENTRAL BUSINESS DISTRICT
## 3              2008        0964190                EXPIRED METER OR OVERSTAY
## 4              2008       0964190A EXP. METER NON-CENTRAL BUSINESS DISTRICT
## 5              2008       0964190B  EXPIRED METER CENTRAL BUSINESS DISTRICT
## 6              2009        0964190                EXPIRED METER OR OVERSTAY
## 7              2009       0964190B  EXPIRED METER CENTRAL BUSINESS DISTRICT
## 8              2010        0964190                EXPIRED METER OR OVERSTAY
## 9              2011        0964190                EXPIRED METER OR OVERSTAY
## 10             2012        0964190                EXPIRED METER OR OVERSTAY
## 11             2012       0964190A EXP. METER NON-CENTRAL BUSINESS DISTRICT
## 12             2012       0964190B  EXPIRED METER CENTRAL BUSINESS DISTRICT
## 13             2013       0964190A EXP. METER NON-CENTRAL BUSINESS DISTRICT
## 14             2013       0964190B  EXPIRED METER CENTRAL BUSINESS DISTRICT
## 15             2014       0964190A EXP. METER NON-CENTRAL BUSINESS DISTRICT
## 16             2014       0964190B  EXPIRED METER CENTRAL BUSINESS DISTRICT
## 17             2015       0964190A EXP. METER NON-CENTRAL BUSINESS DISTRICT
## 18             2015       0964190B  EXPIRED METER CENTRAL BUSINESS DISTRICT
## 19             2016       0964190A EXP. METER NON-CENTRAL BUSINESS DISTRICT
## 20             2016       0964190B  EXPIRED METER CENTRAL BUSINESS DISTRICT
## 21             2017       0964190A EXP. METER NON-CENTRAL BUSINESS DISTRICT
## 22             2017       0964190B  EXPIRED METER CENTRAL BUSINESS DISTRICT
## 23             2018       0964190A EXP. METER NON-CENTRAL BUSINESS DISTRICT
## 24             2018       0964190B  EXPIRED METER CENTRAL BUSINESS DISTRICT
##       n
## 1  3071
## 2  1016
## 3  3542
## 4   432
## 5   116
## 6  4679
## 7     1
## 8  4929
## 9  4967
## 10  639
## 11 3013
## 12 1221
## 13 3173
## 14 1456
## 15 2434
## 16 1421
## 17 2661
## 18 1272
## 19 2436
## 20 1222
## 21 2393
## 22 1330
## 23  987
## 24  681
```

This could be a case of a specific code being the preferred option now but between 2008 and 2011, it was primarily the generic code being used.

Part V - Revenue increase from 'Missing City Sticker'

Q1. What was the old violation code and what is the new violation code? How much was the cost of an initial offense under each code? (You can ignore the ticket for a missing city sticker on vehicles over 16,000 pounds.)

```
df %>% filter(violation_description %in% c(
"NO CITY STICKER VEHICLE UNDER/EQUAL TO 16,000 LBS.","NO CITY STICKER OR IMPROPER DISPLAY")) %>%
group_by(violation_description, violation_code) %>%
summarise(n = n())
```

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

```
## # A tibble: 3 x 3
## # Groups:   violation_description [2]
##   violation_description                              violation_code     n
##   <chr>                                              <chr>          <int>
## 1 NO CITY STICKER OR IMPROPER DISPLAY                0964125        10758
## 2 NO CITY STICKER OR IMPROPER DISPLAY                0976170           15
## 3 NO CITY STICKER VEHICLE UNDER/EQUAL TO 16,000 LBS. 0964125B       14246
```

Answer: 0964125, 0964125B

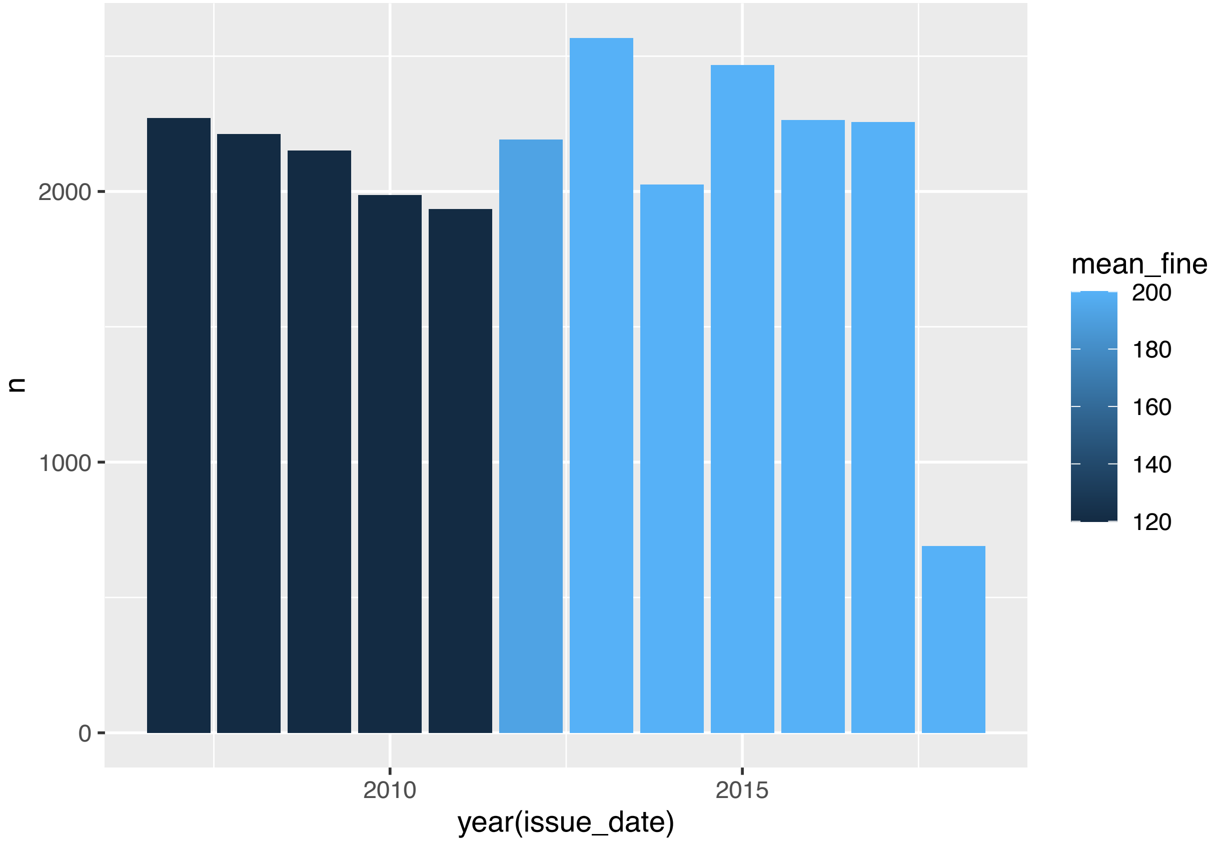
Using these 3 codes, the output gives us the cost of each code.

```
df %>%
filter(violation_code %in% c("0964125B","0964125","0976170")) %>%
group_by(violation_code) %>%
summarise(mean_fine = mean(fine_level1_amount))
```

```
## # A tibble: 3 x 2
##   violation_code mean_fine
##   <chr>              <dbl>
## 1 0964125              120
## 2 0964125B             200
## 3 0976170              120
```

Q2. Combining the two codes, how have the number of missing sticker tickets evolved over time?

```
df %>%
filter(violation_code %in% c("0964125B","0964125","0976170")) %>%
group_by(year(issue_date)) %>%
summarise(mean_fine = mean(fine_level1_amount), n = n()) %>%
ggplot() +
geom_col(aes(x=`year(issue_date)`, y = n, fill = mean_fine))
```



Q3. Using the dates on when tickets were issued, when did the price increase occur?

```
df %>%
filter(violation_code == "0964125") %>%
summarise(last_old_ticket = as.Date(max(issue_date)),
cost = mean(fine_level1_amount))
```

```
##   last_old_ticket cost
## 1      2012-02-24  120
```

```
df %>%
filter(violation_code == "0964125B") %>%
summarise(first_new_ticket = as.Date(min(issue_date)),
cost = mean(fine_level1_amount))
```

```
##   first_new_ticket cost
## 1       2012-02-25  200
```

Q4. The City Clerk said the price increase would raise revenue by $16 million per year. Using only the data available in the calendar year prior to the increase, how much of a revenue increase should she have projected? Assume that the number of tickets of this type issued afterward would be constant and you can assume that there are no late fees or collection fees, so a ticket is either paid at its face value or is never paid.

```
df %>%
filter(year(issue_date) == 2011) %>%
filter(violation_code == "0964125") %>%
group_by(ticket_queue == "Paid") %>%
summarise(n = n()) %>%
mutate(share = n/sum(n))
```

```
## # A tibble: 2 x 3
##   `ticket_queue == "Paid"`     n share
##   <lgl>                    <int> <dbl>
## 1 FALSE                      891 0.461
## 2 TRUE                      1042 0.539
```

These are the tickets paid. 1042 x 100 (since we have a 1% sample) x 0.54 x 80 = $4.5 million

Q5. What happened to repayment rates on this type of ticket in the calendar year after the price increase went into effect? Suppose for a moment that the number of tickets issued was unchanged after the price increase. Taking into account the change in repayment rates, what would the change in revenue have been?

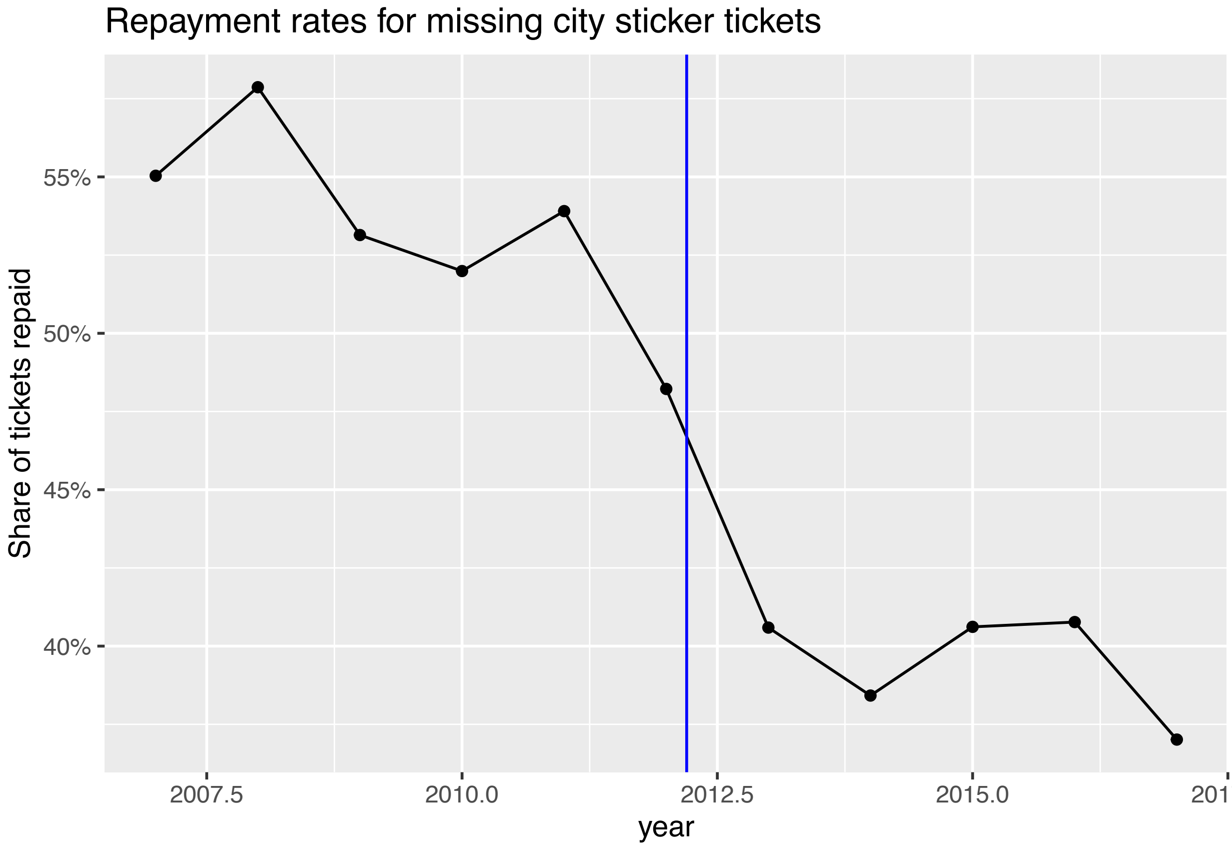
```
df %>%
filter(year(issue_date) == 2013) %>%
filter(violation_code == "0964125B") %>%
group_by(ticket_queue == "Paid") %>%
summarise(n = n()) %>%
mutate(share = n/sum(n))
```

```
## # A tibble: 2 x 3
##   `ticket_queue == "Paid"`     n share
##   <lgl>                    <int> <dbl>
## 1 FALSE                     1525 0.594
## 2 TRUE                      1042 0.406
```

Q6. Make a plot with the repayment rates on no city sticker tickets and a vertical line at when the new policy was introduced.

```
df %>%
filter(violation_code %in% c("0964125", "0964125B") &
year(issue_date) <= 2017) %>%
group_by(year = year(issue_date), paid = ticket_queue == "Paid") %>%
summarise(n = n()) %>%
mutate(share = n/sum(n)) %>%
filter(paid) %>%
ggplot(aes(x = year, y = share)) +
geom_line() + geom_point() +
scale_y_continuous(labels = scales::percent) +
labs(y = "Share of tickets repaid",
title = "Repayment rates for missing city sticker tickets") +
geom_vline(xintercept = 2012.2, color = "blue")
```

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



Q7. In that same year before this city sticker price increase went into force, suppose that the City Clerk were committed to getting revenue from tickets–which we are not advocating. What ticket types would you as an analyst have recommended she increase and why? Name up to three ticket types. Assume there is no behavioral response (ie. people continue to commit violations at the same rate and repay at the same rate), but consider both ticket numbers and repayment rates.

```
df %>%
filter(year(issue_date) == 2011) %>%
group_by(violation_description) %>%
summarise(sum_payments = sum(total_payments),
repay_rate = sum(ifelse(ticket_queue == "Paid",1,0))/n()) %>%
arrange(desc(sum_payments))
```

```
## # A tibble: 77 x 3
##    violation_description                      sum_payments repay_rate
##    <chr>                                             <dbl>      <dbl>
##  1 EXPIRED METER OR OVERSTAY                        257765.      0.823
##  2 NO CITY STICKER OR IMPROPER DISPLAY              212393.      0.540
##  3 EXPIRED PLATES OR TEMPORARY REGISTRATION         203546.      0.636
##  4 STREET CLEANING                                  148205.      0.829
##  5 RESIDENTIAL PERMIT PARKING                       114377.      0.774
##  6 PARKING/STANDING PROHIBITED ANYTIME               92532.      0.723
##  7 REAR AND FRONT PLATE REQUIRED                     61787.      0.585
##  8 RUSH HOUR PARKING                                 58778.      0.786
##  9 PARK OR STAND IN BUS/TAXI/CARRIAGE STAND          48192.      0.724
## 10 NO STANDING/PARKING TIME RESTRICTED               44751.      0.776
## # i 67 more rows
```