# Applied Problem Set 2

Cesar Anzola and Gabriel Angarita

11/06/2021

```r
library(tidyverse)
library(testthat)
library(lubridate)
library(tidycensus)
```

# 1 Front matter

This submission is my work alone and complies with the 30535 integrity policy.

Add your initials to indicate your agreement: CA and GA

Add your collaborators: Gabriel Angarita

Late coins used this pset: 2. Late coins left: 6.

# 2 Part 1

## 2.1 Read in one percent sample (10 Points)

```r
start_time <- Sys.time()
    data <-read.csv("parking_tickets_one_percent.csv")
end_time <- Sys.time()

# calculate the time for loading the dataset
end_time - start_time
```

```
## Time difference of 5.999133 secs
```

```r
# Use test_that to check that there are
# 287458 rows.

test_that("Number of rows of dataset",{
  expect_that(nrow(data),equals(287458))
})
```

```
## Test passed
```

How many megabytes is the file?

```
# memory size of the data on megabytes
a <- object.size(data)/1e+6

print(paste0(a[1]," megabytes"))
```

```
## [1] "145.230384 megabytes"
```

```
# prediction
print(paste0(a[1]/0.01/1000," Gigabytes"))
```

```
## [1] "14.5230384 Gigabytes"
```

```
# I pass it to GIGABYTES to make it clear!!!!!
```

As you can see the dataset use 145 megabytes

Using math, how large would you predict the full data set is?

Given that this dataset contains the 1% of all the tickets then the complete database should be around 100 times bigger than that around 14.5 gigabytes.

How are the rows ordered?

```
data %>%
  select(X,ticket_number,issue_date,violation_location) %>%
    head()
```

```
##   X ticket_number          issue_date violation_location
## 1 1      51482901 2007-01-01 01:25:00    5762 N AVONDALE
## 2 2      50681501 2007-01-01 01:51:00    2724 W FARRAGUT
## 3 3      51579701 2007-01-01 02:22:00       1748 W ESTES
## 4 4      51262201 2007-01-01 02:35:00    4756 N SHERIDAN
## 5 5      51898001 2007-01-01 03:50:00    7134 S CAMPBELL
## 6 6      50681401 2007-01-01 04:10:00     2227 W FOSTERT
```

```
data %>%
  select(X,ticket_number,issue_date,violation_location) %>%
    tail()
```

```
##             X ticket_number          issue_date   violation_location
## 287453 287453      9.19e+09 2018-05-14 14:30:00     1601 W CULLERTON
## 287454 287454      9.19e+09 2018-05-14 14:51:00        1128 W MONROE
## 287455 287455      9.19e+09 2018-05-14 16:34:00 1820 N MILWAUKEE AVE
## 287456 287456      9.19e+09 2018-05-14 16:52:00        122 E 21ST ST
## 287457 287457      9.19e+09 2018-05-14 18:04:00      10 S DEARBORN ST
## 287458 287458      9.19e+09 2018-05-14 20:56:00        2201 W ARTHUR
```

As you can see from the above rows, the dataset is order according to the issue_date of the ticket.

For each column, how many rows are NA? Write a parsimonious command which calculates this. You will not get credit for a command which writes out every variable name.

As you can see below is the list of NA values for each variable in the dataset:

```r
colSums(is.na(data))
```

```
##                     X         ticket_number            issue_date 
##                     0                     0                     0 
##    violation_location  license_plate_number   license_plate_state 
##                     0                     0                    97 
##    license_plate_type               zipcode        violation_code 
##                  2054                 54115                     0 
## violation_description                  unit      unit_description 
##                     0                    29                     0 
##          vehicle_make    fine_level1_amount    fine_level2_amount 
##                     0                     0                     0 
##    current_amount_due        total_payments          ticket_queue 
##                     0                     0                     0 
##     ticket_queue_date          notice_level   hearing_disposition 
##                     0                 84068                259899 
##         notice_number               officer               address 
##                     0                     0                     0
```

The three variables with a large amount of missing variables are: hearin_disposition, notice_level and the zipcode.

WHY?

According to the dictionary of propublica, the hearing disposition variable is blank when the ticket was not contested by the person who was fined with the ticket. Thus, many people did not contested that communications from the city. In addition, not all the tickets are sent with a notice, so there are many tickets that were sent with no notice level and therefore are blank. Moreover, they could not get the information of the zipcode associated with the vehicle registration in most of the cases where people did not respond to the notification (as you can see in the table below).

```r
data %>%
  filter(is.na(zipcode)) %>%
    count(hearing_disposition)
```

```
##   hearing_disposition     n
## 1              Liable     6
## 2          Not Liable     2
## 3                <NA> 54107
```

## 2.2 Cleaning the data and benchmarking (10 points)

Im going to anwser the following questions in the same chunk:

```r
# How many tickets were issued in tickets_1pct in 2017?
data <-
data %>%
  mutate(year = year(issue_date),.after = issue_date )

dat_2017 <-
data %>%
  filter(year==2017)

print(dim(dat_2017)[1])
```

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

```r
# How many tickets does that imply were issued
# in the full data in 2017?

# 100 times more tickets than in this 1% sample:
print(dim(dat_2017)[1]/0.01)
```

```
## [1] 2236400
```

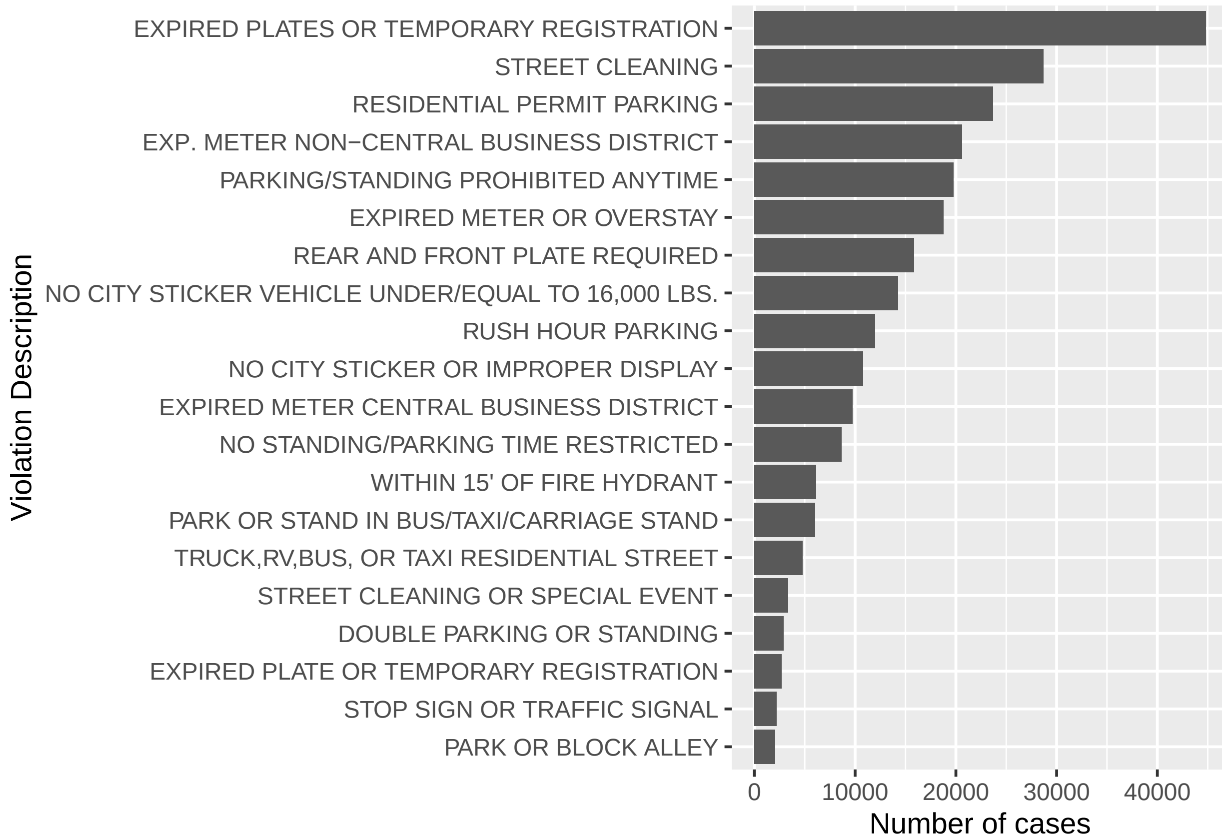
How many tickets are issued each year according to the ProPublica article? Do you think that there is a meaningful difference?

According to the ProPublica article the city of Chicago issued more than 3 million tickets a year for parking and traffic cameras. However, the number of tickets imply in the complete dataset are around 2.24 million tickets. This is a very significant difference of around 760000 tickets respect to the ProPublica article.

What are the top 20 most frequent violation types? Make a bar graph to show the frequency of these ticket types. Make sure to format the graph such that the violation descriptions are legible and no words are cut off.

```r
com_viol <-
data %>%
  count(violation_description) %>%
    arrange(desc(n)) %>%
      head(20)

com_viol %>%
  ggplot()+
  geom_bar(aes(x = reorder(violation_description,n),y = n),stat = "identity")+
    xlab("Violation Description")+
    ylab("Number of cases")+
  coord_flip()
```




## 3 Joins - unit (10 points)

The data tell us what unit of city government issued the ticket, but we need to merge on a crosswalk.

For how many tickets is unit missing?

```
sum(is.na(data$unit))
```

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

As you can see for 29 tickets the unit is missing.

Read in unit_key.csv.

```
unit_key <- read.csv("unit_key.csv")

name = c("Reporting_District", "Department_Name", "Department_Description", "Department_Category")

#name <- str_replace(unit_key[2,1:4]," ","_")

unit_key<- unit_key[-1:-2,1:4]
names(unit_key) <-name

unit_key %>%
```

```
  filter(!is.na(Reporting_District)) %>%
    summarise(tot = n())
```

```
##   tot
## 1 385
```

How many units are there?

As you can see there are 385 unit districts that are not missing values.

Join unit key to the tickets data. How many rows in the tickets data have a match in the unit table?

```
data <- arrange(data,unit)

unit_key = unit_key %>% mutate(unit = as.numeric(Reporting_District),.before = Reporting_District) %>% a
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
data_new <- left_join(data,unit_key, by = "unit",na_matches = "never")

# check if thbere are different values between
# the new database and the previous dataset

sum(is.na(data_new$Department_Name))
```

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

From the previous code you can see that for the variable Department_Name that comes from the unit database, there are no ADDITIONAL missing values than the previous missing values that we calculate before. This means that all of the units from the tickets database were found in the unit_key database. So the total amount of thickets that match are: 287458-29 = 287429 values match.

How many rows are unmatched?

As you can see from the comparison above every unit that is found in the tickets database can be found in the unit database.However, there are some NA values of the unit variable in the tickets database. By construction this NA values could not match with any of the units from the unit_key database. Therefore, the rows that would not be match are the ones with NA values in the tickets dataset which are 29 observations.

How many rows in the unit table have a match in the tickets data? How many do not?

```
# first clean the na variables from the key variable
unit_key <-
unit_key %>%
  filter(!is.na(unit))

data <-
data %>%
  filter(!is.na(unit))

# use anti_join to check mismatches
aux_d <- anti_join(unit_key,data,by = "unit")

# matches of unit data and tickets data
dim(unit_key)[1]-dim(aux_d)[1]
```

```
## [1] 128
```

```r
# unmatches of unit data and tickets data
dim(aux_d)[1]
```

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

Above you can see that 128 units match and that 246 units don't match.

Who issues more tickets - Department of Finance or Chicago Police?

As you can see the department of finance issue more tickets than all of the departments of the chicago policie combined.

```r
dof_tick  <-
  data_new %>%
  filter(Department_Name == "DOF") %>%
  count()

cpd_li <- c("CPD","CPD-Other","CPD-Airport")

cpd_tick <-
data_new %>%
  filter(Department_Name %in% cpd_li) %>%
   count()

print(paste0("Department of finance ", dof_tick," Chicago police department ",cpd_tick))
```

```
## [1] "Department of finance 143909 Chicago police department 127078"
```

Within Chicago Police, what are the top 5 departments that are issuing the most tickets? Be careful what your group by here and avoid columns with ambiguities.

```r
data_new <-
data_new %>%
  mutate(ChicPolice = case_when(Department_Name %in% cpd_li ~ TRUE,
                                TRUE ~ FALSE))


data_new %>%
  group_by(ChicPolice) %>%
    filter(ChicPolice == TRUE) %>%
      count(Department_Description) %>%
        arrange(desc(n)) %>%
         head(5)
```

```
## # A tibble: 5 x 3
## # Groups:   ChicPolice [1]
##   ChicPolice Department_Description     n
##   <lgl>      <chr>                  <int>
## 1 TRUE       1160 N. Larrabee        9478
## 2 TRUE       6464 N. Clark           7946
## 3 TRUE       OEMC                    7374
## 4 TRUE       3315 W. Ogden           5469
## 5 TRUE       5555 W. Grand           5464
```

# 4 Joins - ZIP code (15 points)

1. Download recent census data by ZIP for Chicago with population, share black and median household income. chi_zips.csv

```
census_api_key("8102bfd22541083fbad1ea6ff7660316470aed2e",overwrite = TRUE)
```

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

```
CENSUS_KEY <- Sys.getenv("CENSUS_API_KEY")

zips_chicago <- read.csv("chi_zips.csv")

# zips_chicago <- rename(zips_chicago,"ï..ZIP = zip")

zips_chicago <- rename(zips_chicago,"ZIP" = "ï..ZIP")


# I used this dictionary to check on the name of the variables
#dp14 <- load_variables(2014, "acs5", cache = TRUE)

data_ill <- get_acs(
   geography = "zcta",
   state = "Il",
  variables = c(medincome = "B19013_001", tot_pop = "B01003_001",black_pop =    "C02003_004"),
  year = 2014
)
```

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

```
data_ill <-
data_ill %>%
  pivot_wider(values_from = c("estimate","moe"),names_from = variable) %>%
  mutate(black_share = estimate_black_pop/estimate_tot_pop)

# Lets filter only the zip codes of Chicago
data_ill <- rename(data_ill,"zipcode" = "GEOID")

data_chic <- data_ill %>% filter(zipcode %in% zips_chicago$ZIP)
```

2. Clean vehicle registration ZIP and then join the Census data to the tickets data

```
# filter the na values for zipcode
data_clean <-
data_new %>%
  filter(zipcode %in% zips_chicago$ZIP)

# merge the census data with the tickets data
join_data <- left_join(data_clean,data_chic,by = "zipcode")
```

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
# create a paid and not paid variable
join_data %>%
  count(ticket_queue)
```

```
##   ticket_queue     n
## 1   Bankruptcy  1362
## 2        Court   114
## 3       Define  1483
## 4    Dismissed  4368
## 5  Hearing Req     4
## 6       Notice 15436
## 7         Paid 34034
```

```r
join_data <-
join_data %>%
  mutate(paid_ticket = ifelse(ticket_queue=="Paid" | ticket_queue=="Dismissed" ,"Yes","No"))

# zips with more tickets
top_zips <-
join_data %>%
  filter(paid_ticket == "No") %>%
    count(zipcode,sort = TRUE) %>%
      head(10)

# join with population

tick_res <- left_join(top_zips,data_chic,by = "zipcode")

tick_res %>%
  mutate(tick_per_resid = n/estimate_tot_pop) %>%
    arrange(desc(tick_per_resid)) %>%
      select(zipcode,n,estimate_tot_pop,tick_per_resid)
```

```
##    zipcode    n estimate_tot_pop tick_per_resid
## 1    60636  696            40164    0.017328951
## 2    60624  676            39706    0.017025135
## 3    60644  830            49615    0.016728812
## 4    60651  843            60938    0.013833733
## 5    60623 1094            87836    0.012455030
## 6    60620  813            71907    0.011306271
## 7    60619  668            64245    0.010397696
## 8    60628  722            69921    0.010325939
## 9    60639  770            92339    0.008338838
## 10   60629  833           115013    0.007242660
```

The top three neighborhoods of more unpaid tickets per resident population are:

1. West Englewood
2. West Garfield Park
3. Austin

# 5 Understanding the structure of the data (20 points)

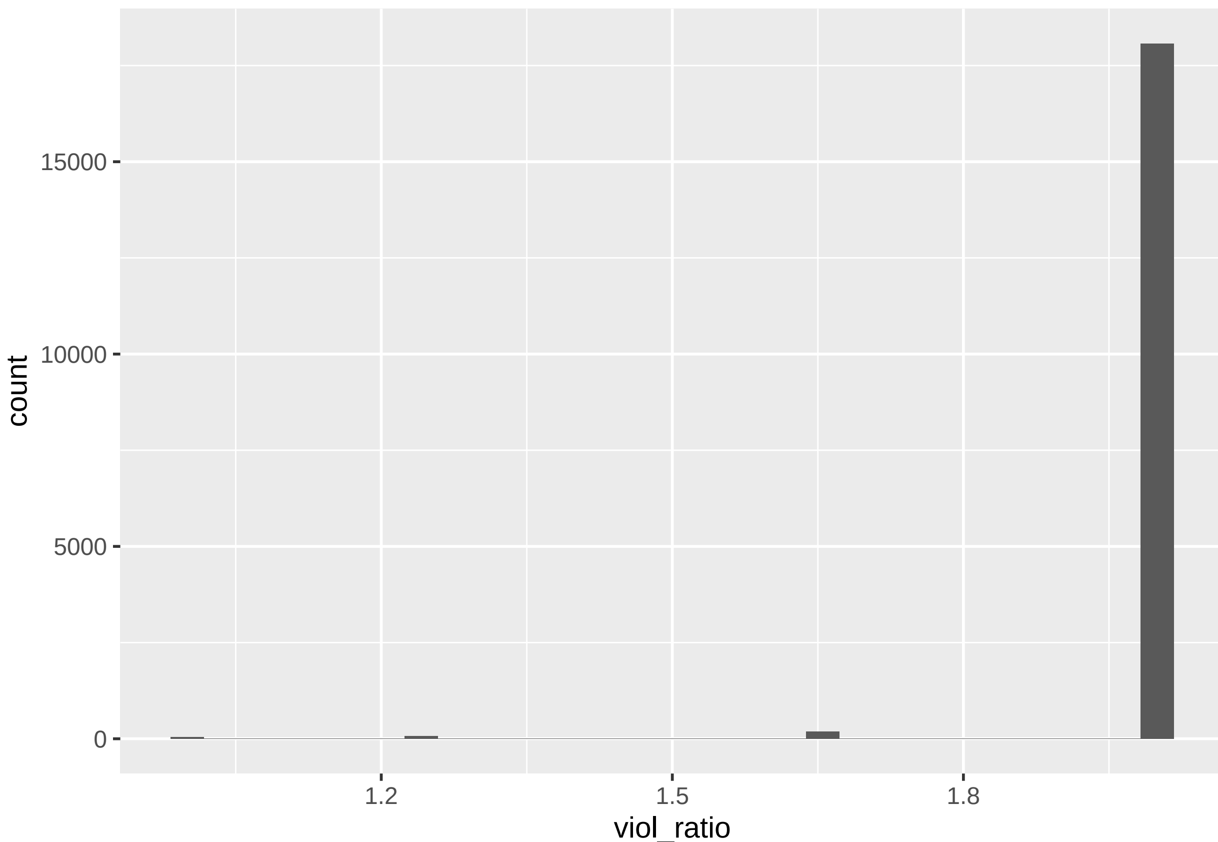
Most violation types double in price if unpaid. Does this hold for all violations?

As you can see in the histogram of the increase in the ticket price, not all violations double the price (although most of them do)

```
fine_increase <-
join_data %>%
  filter(paid_ticket == "No") %>%
   select(ticket_number,ticket_queue,fine_level1_amount,fine_level2_amount,violation_description,paid_ti
     mutate(viol_ratio = fine_level2_amount/fine_level1_amount)

# graph of the distribution of the increase in tickets amount to paid
ggplot(fine_increase)+
  geom_histogram(aes(viol_ratio))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




If not, find all violations with at least 100 citations that do not double. How much does each ticket increase if unpaid?

```
specific_case <-
fine_increase %>%
   filter(viol_ratio<2) %>%
      count(violation_description,sort = TRUE) %>%
```

```r
      filter(n>=100)

fine_increase %>%
  filter(violation_description %in% specific_case$violation_description) %>%
    summarise(mean_increase = mean(viol_ratio))
```

```
##   mean_increase
## 1      1.723618
```

As you can see for the tickets violations that didn't double it's price when unpaid was on average around 70% increase.

Many datasets implicitly contain information about how a case can progress. Draw a diagram explaining the process of moving between the different values of notice_level (if you draw it on paper, take a picture and include the image using knitr::include_graphics). Draw a second diagram explaining the different values of ticket_queue. If someone contests their ticket and is found not liable, what happens to notice_level and to ticket_queue? Include this in your drawings.

```r
unique(join_data$notice_level)
```

```
## [1] "FINL" "DETR" "VIOL" NA     "SEIZ" "DLS"
```

```r
# "VIOL," which means a notice of violation was sent;
#
# "SEIZ" indicates the vehicle is on the city's boot list;
#
# "DETR" indicates a hearing officer found the vehicle owner was found liable for the citation;
#
# "FINL" indicates the unpaid ticket was sent to collections;
#
# "DLS" means the city intends to seek a license suspension.

# If the field is blank, no notice was sent.

join_data %>%
  filter(hearing_disposition =="Not Liable") %>%
    count(ticket_queue)
```

```
##   ticket_queue    n
## 1    Dismissed 3800
## 2         Paid    2
```

```r
join_data %>%
  filter(hearing_disposition =="Not Liable") %>%
    count(notice_level)
```

```
##   notice_level    n
## 1         DETR  411
## 2         VIOL 2239
## 3         <NA> 1152
```

```
join_data %>%
  filter(hearing_disposition =="Liable") %>%
    count(ticket_queue)
```

```
##   ticket_queue    n
## 1   Bankruptcy   22
## 2        Court    1
## 3    Dismissed    7
## 4       Notice  299
## 5         Paid 1743
```

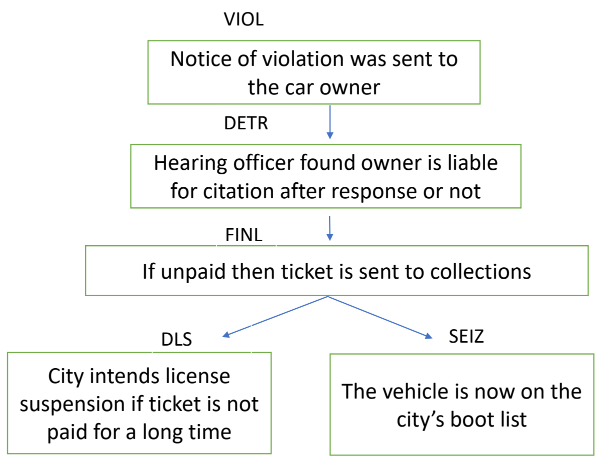



Figure 1: Notice Level Steps

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? (Hint: this can be done in just four lines of code)

```
dup <-
join_data %>%
  group_by(violation_description,violation_code) %>%
        count(violation_code) %>%
          summarise(n = n()) %>%
           count(violation_description) %>%
           filter(n>1)
```

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

```
join_data %>%
    group_by(violation_description,violation_code) %>%
      filter(violation_description %in% dup$violation_description) %>%
         count(violation_code)
```

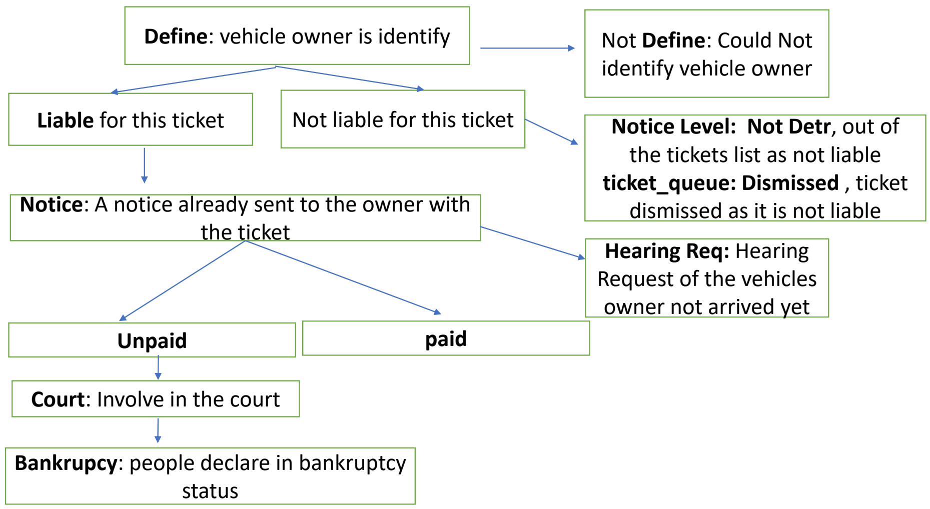



Figure 2: Ticket queue

```
## # A tibble: 6 x 3
## # Groups:   violation_description, violation_code [6]
##   violation_description                violation_code     n
##   <chr>                                <chr>          <int>
## 1 3-7 AM SNOW ROUTE                    0964060          166
## 2 3-7 AM SNOW ROUTE                    0964060B           2
## 3 NO CITY STICKER OR IMPROPER DISPLAY 0964125         4397
## 4 NO CITY STICKER OR IMPROPER DISPLAY 0976170            3
## 5 SPECIAL EVENTS RESTRICTION           0964041           36
## 6 SPECIAL EVENTS RESTRICTION           0964041B          32
```

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?

```
dup <-
join_data %>%
  group_by(violation_code,violation_description) %>%
        count(violation_description) %>%
          summarise(n = n()) %>%
           count(violation_code) %>%
           filter(n>1)
```

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

```
join_data %>%
    group_by(violation_code,violation_description) %>%
      filter(violation_code %in% dup$violation_code) %>%
         count(violation_description)
```

```
## # A tibble: 16 x 3
```

```
## # Groups:   violation_code, violation_description [16]
##    violation_code violation_description                              n
##    <chr>          <chr>                                          <int>
##  1 0964040B       STREET CLEANING                                 5542
##  2 0964040B       STREET CLEANING OR SPECIAL EVENT                 730
##  3 0964041B       Special Events                                     4
##  4 0964041B       SPECIAL EVENTS RESTRICTION                        32
##  5 0964070        SNOW ROUTE: 2'' OF SNOW OR MORE                   27
##  6 0964070        SNOW ROUTE: 2' OF SNOW OR MORE                     5
##  7 0964170D       TRUCK OR SEMI-TRAILER PROHIBITED                  20
##  8 0964170D       TRUCK TRAILOR/SEMI/TRAILER PROHIBITED             10
##  9 0964200B       OUTSIDE METERED SPACE                              8
## 10 0964200B       PARK OUTSIDE METERED SPACE                        60
## 11 0976160A       MISSING/NONCOMPLIANT FRONT AND/OR REAR PLATE     166
## 12 0976160A       REAR AND FRONT PLATE REQUIRED                   3104
## 13 0976160B       EXPIRED PLATE OR TEMPORARY REGISTRATION          184
## 14 0976160B       REAR PLATE REQUIRED MOTORCYCLE/TRAILER            26
## 15 0980110B       HAZARDOUS DILAPIDATED VEHICLE                     35
## 16 0980110B       HAZARDOUS DILAPITATED VEHICLE                     74
```

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?

```
viol_list <-
join_data %>%
  count(violation_description,sort = TRUE) %>%
    head(50)

viol_list %>%
  mutate( stick = str_count(violation_description,"STICKER")) %>%
    filter(stick==1)
```

```
##                                violation_description    n stick
## 1 NO CITY STICKER VEHICLE UNDER/EQUAL TO 16,000 LBS. 6263     1
## 2                NO CITY STICKER OR IMPROPER DISPLAY 4400     1
## 3                   IMPROPER DISPLAY OF CITY STICKER  100     1
## 4           NO CITY STICKER VEHICLE OVER 16,000 LBS.   56     1
```

As you can see from the table above the city sticker violation description might look like a redundant category (why not group them all in just one variable?). However, you can see for example that there is one category of "no city sticker for vehicles under or equal to 16000 LBS" (small vehicles) and another one for vehicles larger than this (big vehicles). For a particular reason they like to separate this two kind of vehicles and that's why there are 2 categories.

# 6 Revenue increase from "missing city sticker" tickets

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.)

1. The difference in the code is the "D" and "B" value.

```r
join_data <-
  join_data %>%
     mutate(year = year(issue_date))

df_sticker <- join_data %>% mutate(stick = str_count(violation_description,"STICKER")) %>%
    filter(stick==1)

df_sticker = df_sticker %>%
             filter(!violation_description == "NO CITY STICKER VEHICLE OVER 16,000 LBS.") %>%
     mutate(year = year(issue_date))



df_sticker_n <-
  df_sticker %>%
      group_by(year,violation_code, violation_description) %>% summarise(Total = n())
```

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

```r
head(df_sticker_n)
```

```
## # A tibble: 6 x 4
## # Groups:   year, violation_code [6]
##    year violation_code violation_description               Total
##   <dbl> <chr>          <chr>                               <int>
## 1  2007 0964125        NO CITY STICKER OR IMPROPER DISPLAY   950
## 2  2007 0976170        NO CITY STICKER OR IMPROPER DISPLAY     1
## 3  2008 0964125        NO CITY STICKER OR IMPROPER DISPLAY   893
## 4  2009 0964125        NO CITY STICKER OR IMPROPER DISPLAY   861
## 5  2009 0976170        NO CITY STICKER OR IMPROPER DISPLAY     2
## 6  2010 0964125        NO CITY STICKER OR IMPROPER DISPLAY   781
```

```r
df_sticker_cost = df_sticker %>%
            group_by(year,violation_code, violation_description) %>%
            summarise(Total_Cost = mean(fine_level1_amount , na.rm = TRUE))
```

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

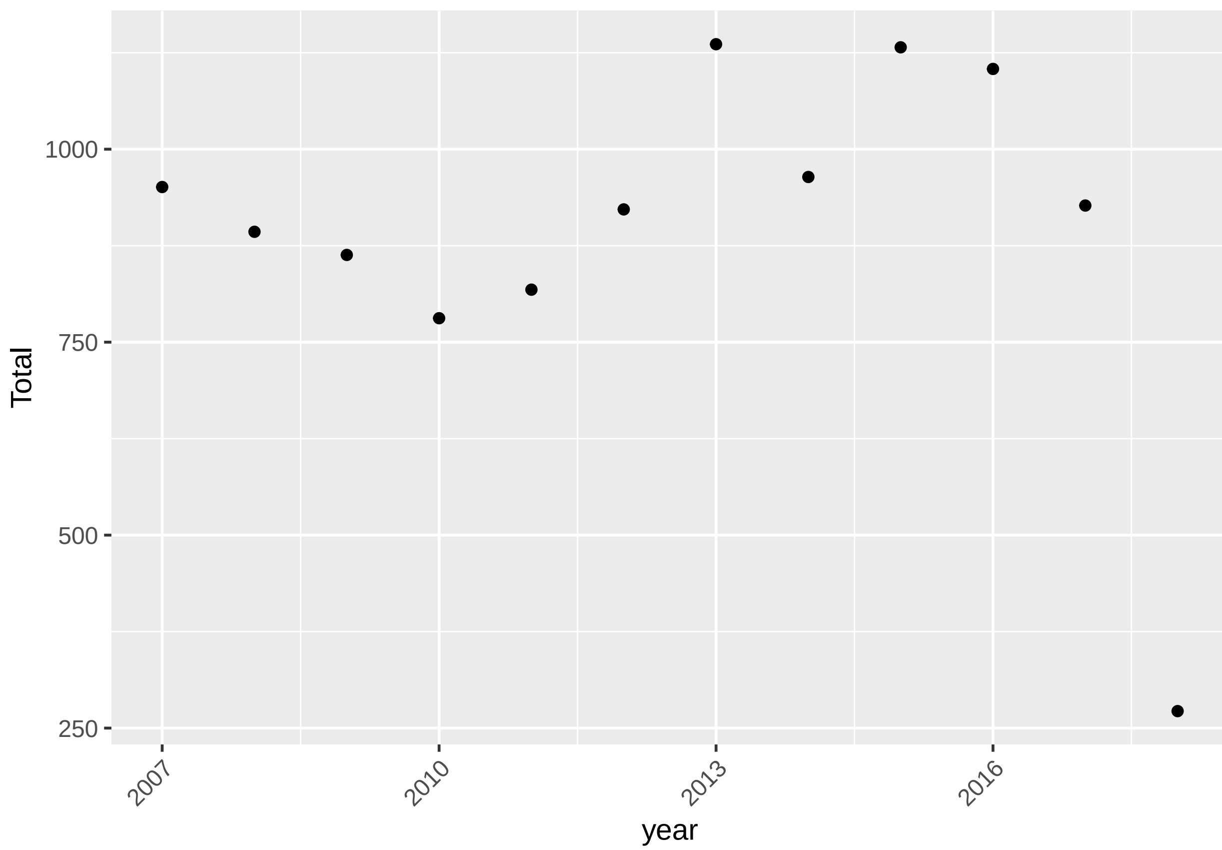
```r
head(df_sticker_cost)
```

```
## # A tibble: 6 x 4
## # Groups:   year, violation_code [6]
##    year violation_code violation_description               Total_Cost
##   <dbl> <chr>          <chr>                                    <dbl>
## 1  2007 0964125        NO CITY STICKER OR IMPROPER DISPLAY        120
## 2  2007 0976170        NO CITY STICKER OR IMPROPER DISPLAY        120
## 3  2008 0964125        NO CITY STICKER OR IMPROPER DISPLAY        120
## 4  2009 0964125        NO CITY STICKER OR IMPROPER DISPLAY        120
## 5  2009 0976170        NO CITY STICKER OR IMPROPER DISPLAY        120
## 6  2010 0964125        NO CITY STICKER OR IMPROPER DISPLAY        120
```

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

```
df_sticker_n_year = df_sticker %>%
             group_by(year) %>% summarise(Total = n())

ggplot(df_sticker_n_year) + geom_point(mapping = aes(x = year, y = Total)) + theme(axis.text.x= element_
```




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

```
df_sticker_time = df_sticker %>%
                  mutate (day = as.Date(issue_date))

df_sticker_time = df_sticker_time %>%
            group_by(day) %>%
            summarise(Total_Cost = mean(fine_level1_amount , na.rm = TRUE))

df_sticker_time_change = df_sticker_time %>%
                         filter (Total_Cost > 120)
```

Day of the change: 2012-02-25

4. The City Clerk said the price increase would raise 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 is face value or is never paid.

```
df_sticker_time_prior = df_sticker %>%
                         filter (year<2012)

df_sticker_time_prior = df_sticker_time_prior %>% group_by(year) %>%
                        summarise(Total_Revenue = sum(total_payments))

projected_2012 =  df_sticker_time_prior[5,2] + 160000
projected_2012
```

```
##   Total_Revenue
## 1      253125.4
```

5. What happened to repayment rates on this type of ticket in the calendar year after the increase went into effect?

How many tickets are paid divided by total number of tickets.

```
df_sticker_repayment = df_sticker %>% group_by(year, paid_ticket) %>%
                       summarise(Total = n()) %>% group_by(year) %>%
                       mutate(Total_tickets = sum(Total))  %>%
                       mutate(repayment = Total/Total_tickets) %>%
                       filter(paid_ticket == "Yes")
```

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

The repayment rate shows a drop after the price increase. At that time, it went from 0.53 in 2011 to 0.42 in 2012, then in 2013 it stood at 0.38.

If the City had not started issuing more of these tickets, what would its change in revenue have been?:

```
# asuming the number of tickets from 2011 constant = 818
df_sticker_repayment %>%
  select(year,Total_tickets) %>%
     filter(year==2012)
```

```
## # A tibble: 1 x 2
## # Groups:   year [1]
##    year Total_tickets
##   <dbl>         <int>
## 1  2012           922
```

```
# Maintaining the price in 2012 constant
df_sticker %>%
  filter(year==2012) %>%
    summarise(mean_p = mean(fine_level1_amount))
```

```
##   mean_p
## 1    190
```

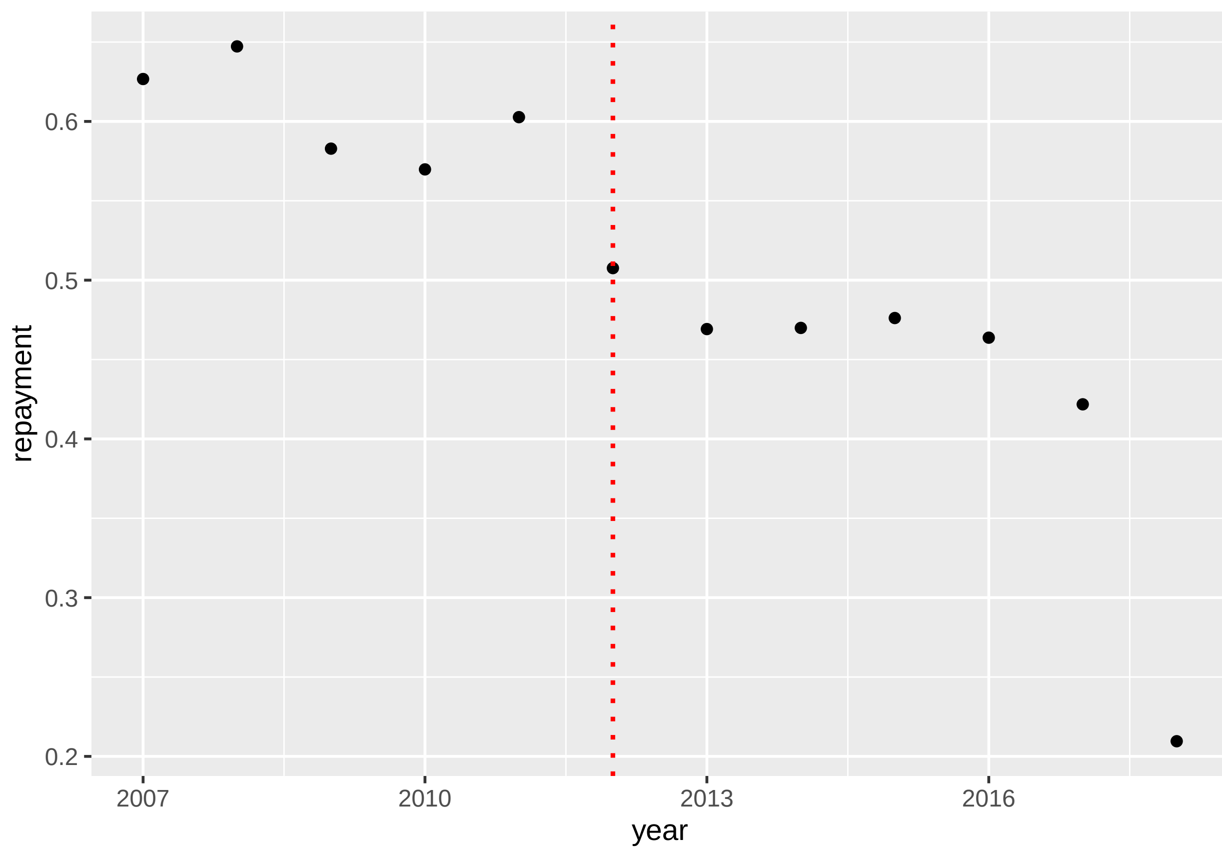
```
922*190 - 818*190
```

```
## [1] 19760
```

Keeping the number of tickets of 2011 constant at 818 and keeping the price of 2012 (price increase), they would have recieved around 20 thousand dollars less without the ticket increase.

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

```
ggplot (data = df_sticker_repayment) + geom_point(mapping = aes(x = year,y = repayment)) + geom_vline(x
```




In this case, it is observed that the repayment rate presents a change as of 2012, just at the moment when the price increase occurred. This change has become more pronounced in subsequent years.

Help: http://www.sthda.com/english/wiki/ggplot2-add-straight-lines-to-a-plot-horizontal-vertical-and-regression-lines

7. Still focusing on the period before the policy change, suppose that the City Clerk were committed to getting revenue from tickets rather than other sources. 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_policy = join_data %>%
  filter (year<2012) %>%
    group_by(violation_description) %>%
      summarise(Total = n(), Total_Cost = mean(fine_level1_amount , na.rm = TRUE))

df_repayment = join_data %>% filter (year<2012) %>%
               group_by(paid_ticket, violation_description) %>%
               summarise(Total = n()) %>% group_by(violation_description) %>%
               mutate(Total_tickets = sum(Total))  %>%
               mutate(repayment = Total/Total_tickets) %>%
               filter(paid_ticket == "Yes") %>% arrange(-Total)
```

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

```
head(df_repayment)
```

```
## # A tibble: 6 x 5
## # Groups:   violation_description [6]
##   paid_ticket violation_description                Total Total_tickets repayment
##   <chr>       <chr>                                <int>         <int>     <dbl>
## 1 Yes         NO CITY STICKER OR IMPROPER DISPLAY   2615          4306     0.607
## 2 Yes         EXPIRED METER OR OVERSTAY             2163          2617     0.827
## 3 Yes         STREET CLEANING                       1751          2190     0.800
## 4 Yes         RESIDENTIAL PERMIT PARKING            1631          2191     0.744
## 5 Yes         EXPIRED PLATES OR TEMPORARY REGISTRATION  1603      2820     0.568
## 6 Yes         PARKING/STANDING PROHIBITED ANYTIME   1408          1861     0.757
```

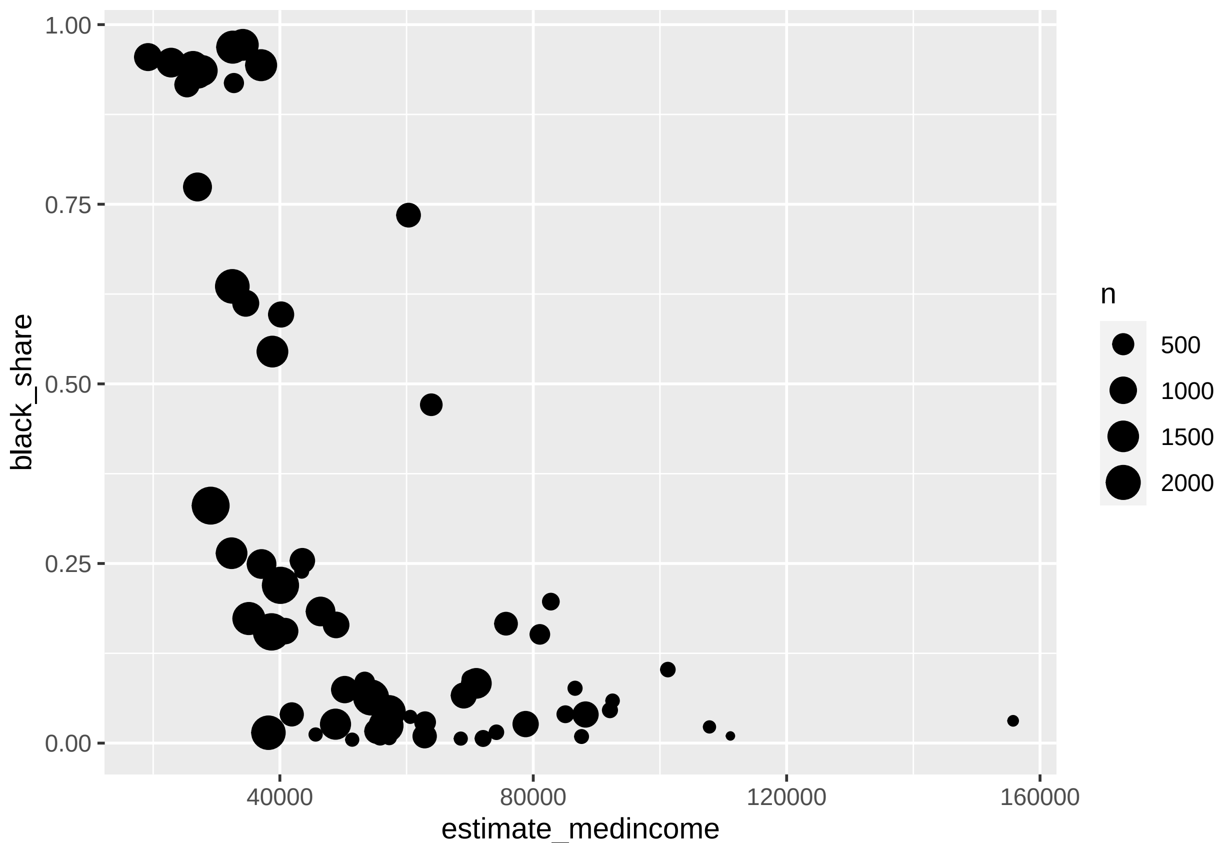
Following the criterion of maximizing income, our advice would be to review the most frequent infractions that have a high repayment rate, greater than 70%. In other words, I would recommend that a price increase be made on these 3 infractions. The three violations I would recommend would be: EXPIRED METER OR OVERSTAY, STREET CLEANING and RESIDENTIAL PERMIT PARKING

8. In the previous question, the City Clerk was only optimizing gross revenue. Melissa Sanchez argue that ticketing is inherently regressive. Let's say the City Clerk took this critique to heart and determined to raise ticket prices for violations that would affect households in high income zip codes more than low income zip codes.

8a. What ticket types would you as an analyst recommend she increase and why? Make a data visualization to support your argument.

```
ggplot(data = join_data) + geom_count(mapping = aes(x = estimate_medincome , y = black_share))
```

```
## Warning: Removed 134 rows containing non-finite values (stat_sum).
```

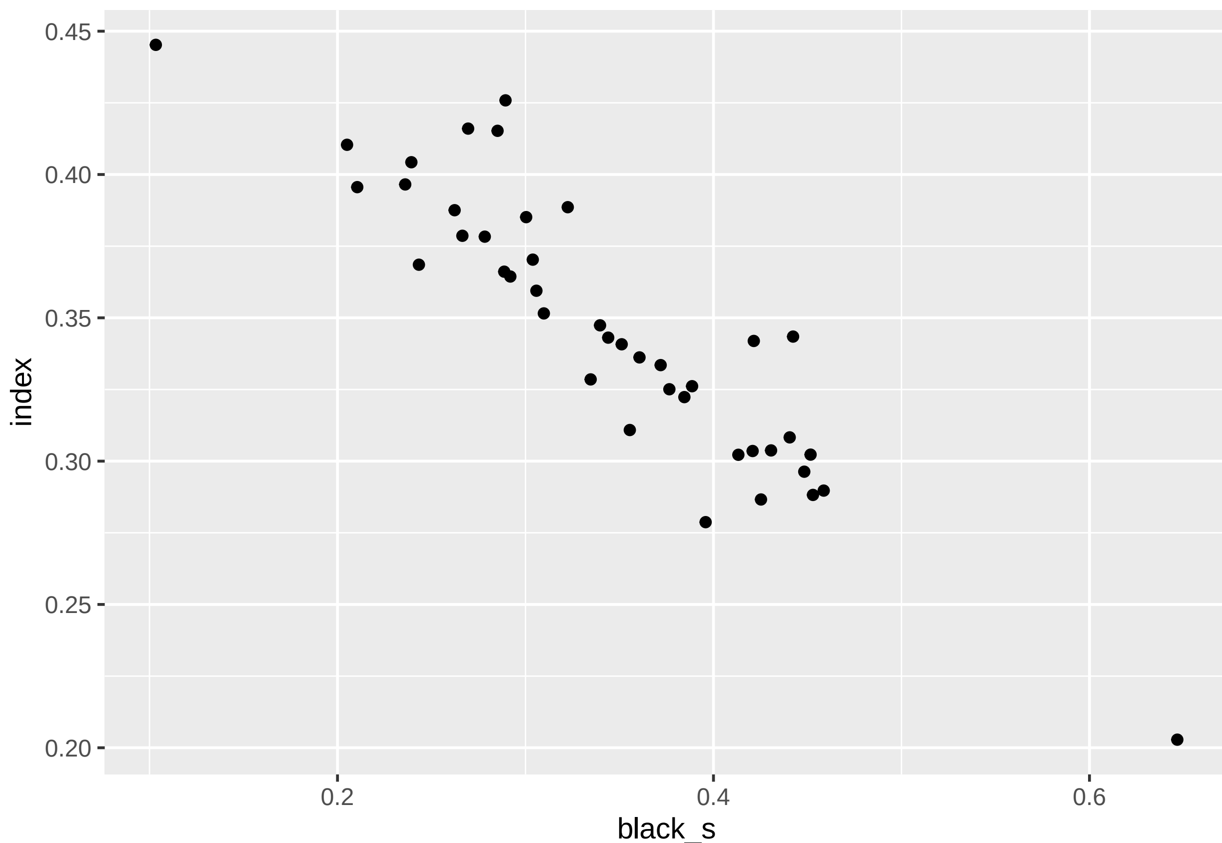



```
df_progressive = join_data %>% filter (year<2012) %>%
                 filter (!violation_description == "NO CITY STICKER OR IMPROPER DISPLAY") %>%
                 group_by(violation_description) %>%
                 summarise(black_s = mean(black_share , na.rm = TRUE))

df_progressive = df_progressive %>% left_join(df_repayment, by = "violation_description")


df_progressive =  df_progressive %>% filter(!repayment==1) %>% filter(Total>10) %>%
                  ungroup() %>% mutate(Total_City = sum(Total_tickets)) %>%
                  mutate(Total_City_Share = Total_tickets/Total_City) %>%
                  mutate(n_black_share = 1-black_s) %>%
                  mutate(index = 0.4*n_black_share+0.5*Total_City_Share+0.1*repayment)

ggplot(data = df_progressive) + geom_point(mapping = aes(x = black_s , y = index))
```




```
df_progressive = df_progressive %>% arrange (index) %>% top_n(5)
```

```
## Selecting by index
```

```
head (df_progressive)
```

```
## # A tibble: 5 x 10
##   violation_descri~ black_s paid_ticket Total Total_tickets repayment Total_City
##   <chr>                <dbl> <chr>       <int>         <int>     <dbl>      <int>
## 1 TRUCK,RV,BUS, OR~    0.205 Yes           423           525     0.806      22189
## 2 STREET CLEANING      0.285 Yes          1751          2190     0.800      22189
## 3 RESIDENTIAL PERM~    0.269 Yes          1631          2191     0.744      22189
## 4 EXPIRED METER OR~    0.289 Yes          2163          2617     0.827      22189
## 5 TRUCK,MOTOR HOME~    0.103 Yes            57            67     0.851      22189
## # ... with 3 more variables: Total_City_Share <dbl>, n_black_share <dbl>,
## #   index <dbl>
```

First, there is a negative relationship between average income and the percentage of black people. In order to carry out a more progressive policy, then, the percentage of black people in the neighborhoods should be considered according to each type of violation.

Based on this, we created an index was built to identify infractions that occur to a lesser extent in neighborhoods with black population, violations with a high repayment rate and a high participation in total fines.

From the value of the index, the top 5 infractions with the highest value in the index were selected, so that the infractions that should increase in price would be given by the value of the index.

8b. If she raises the ticket price by $80 for each of these tickets, how much additional revenue can she expect? Assume there is no behavioral response (ie. people continue to commit violations at the same rate and repay at the same rate).

```
df_projected =  join_data %>% filter (year==2011) %>%
                group_by(year, violation_description) %>%
                summarise(fine = mean(fine_level1_amount), Tickets_total = n()) %>%
                mutate(revenue = fine*Tickets_total)
```

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

```
df_projected_final = df_progressive %>% left_join(df_projected, by = "violation_description") %>%
                    mutate(new_fine = fine+80, revenue_new =new_fine*Tickets_total)

df_projected_final =  df_projected_final %>%
                      summarise(total_new_revenue = sum(revenue_new),
                                total_revenue = sum(revenue))
head(df_projected_final)
```

```
## # A tibble: 1 x 2
##   total_new_revenue total_revenue
##               <dbl>         <dbl>
## 1            218455         83655
```

```
print(1-(df_projected_final$total_revenue/df_projected_final$total_new_revenue))
```

```
## [1] 0.6170607
```

There would be an increase in the budget of 61% for 2012. This taking as a reference the 5 violations selected in the previous point and applying the increase of 80 dollars. The same number of infractions is assumed.