30535 Applied Problem Set 3

05/05/2022

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

Add your initials to indicate your agreement: **E.C.,M.J.**

Late coins used this pset: 2. Late coins left: 3.

Submission Notes: Total page of the file is less than 25 pages and we've submitted on github

Working Directory and Loading Packages

2 Read in one percent sample

2.1

```
# Read data and Calculate time
set.seed(1)
system.time({
  read_csv("parking_tickets_one_percent-3.csv")
})
```

```
##    user  system elapsed
##   2.081   0.323   2.116
```

```
tickets_1pct <- read_csv("parking_tickets_one_percent-3.csv")
# Test the number of rows
test_that("We have the right number of rows", expect_equal(nrow(tickets_1pct), 287458))
```

```
## Test passed
```

Answer:
It takes 2.327 seconds to read in this file. And there are 287458 rows in the dataset.

2.2

```
# Calculate megabytes
fs::file_size(system.file("data/Rdata.rdb", package = "datasets"))
```

```
## 113K
```

```r
# Estimate
113 * 100
```

```
## [1] 11300
```

Answer:
The sample data set has approximately 113MB. Since the sample is randomly chosen as 1% of the full data, we would expect the full data set to be as large as 11300 MB.
Reference https://stackoverflow.com/questions/30580798/how-to-check-file-size-before-opening

2.3

The rows are ordered in ascending order based on the first column. With further exploration of the data set, we notice that the number of the first column is decided by the *issue_date* column. The earlier the date and time is, it will be ranked with a smaller number.

2.4

```r
# Find out the NA values in each column
as.matrix(colSums(is.na(tickets_1pct)))
```

```
##                          [,1]
## ...1                        0
## ticket_number               0
## issue_date                  0
## violation_location          0
## license_plate_number        0
## license_plate_state        97
## license_plate_type       2054
## zipcode                 54115
## violation_code              0
## violation_description       0
## unit                       29
## unit_description            0
## vehicle_make                0
## fine_level1_amount          0
## fine_level2_amount          0
## current_amount_due          0
## total_payments              0
## ticket_queue                0
## ticket_queue_date           0
## notice_level            84068
## hearing_disposition    259899
## notice_number               0
## officer                     0
## address                     0
```

Answer:
As shown in the matrix above, we can observe the number of NAs in each column. There are 6 columns out of 24 which have NAs.

2.5

Answer:
zipcode, *hearing_disposition*, *notice_level* have more NA values than all the other columns. By reading the *parking_tickets_data_dictionary-1.txt* file, we notice that:

- For *zipcode*, it represents the ZIP code associated with the vehicle registration. Chances are that lots of cars have registered for a long time so there is no available information. Also, since those are data of cars which faced with tickets, the percentage of *illegal* cars is likely to be high. They may never registered officially so there is no zipcode information.
- For *hearing_disposition*, if the ticket was not contested, then the data would be blank. There could be lots of tickets remain uncontested.
- For *notice_level*, it describes the type of notice the city has sent a motorist. If the motorist never violated any regulation and received no notice from the city, then it's the blank. It tells us that nearly 29% of data in this data set is from motorist who never received notice from the city.

3 Cleaning the data and benchmarking

3.1

```
# Separate the data column
tickets_1pct <- tickets_1pct %>%
  mutate(
    year = lubridate::year(issue_date),
    month = lubridate::month(issue_date),
    day = lubridate::day(issue_date),
    hour = lubridate::hour(issue_date),
    min = lubridate::minute(issue_date)
  )
tickets_1pct %>%
  filter(year == 2017) %>%
  nrow()
```

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

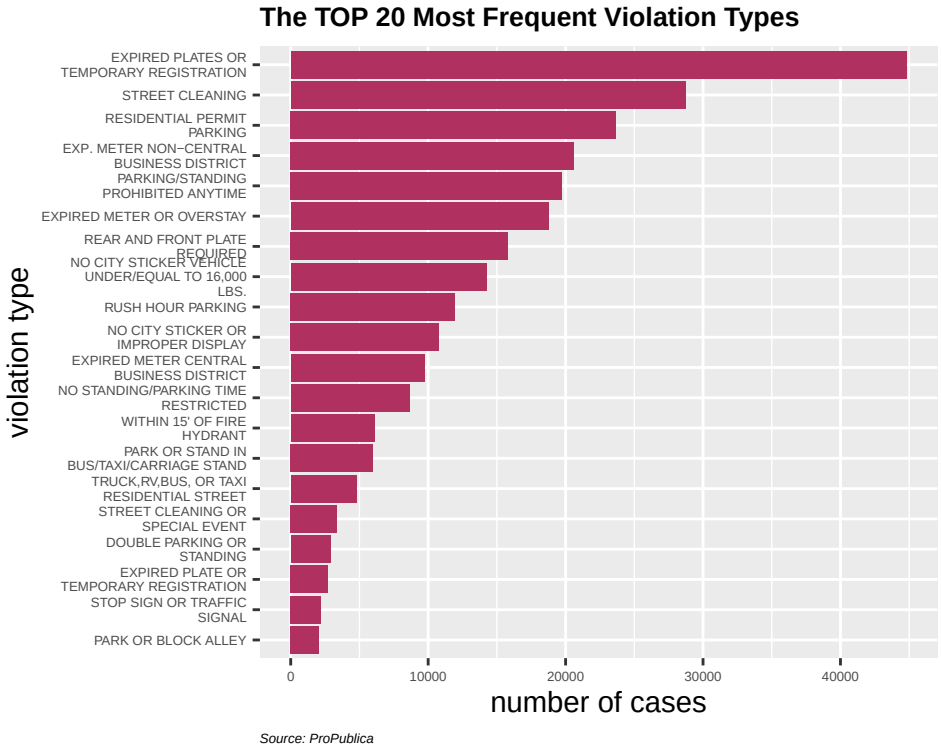
```
# Estimate the difference
22364 * 100 / 3000000
```

```
## [1] 0.7454667
```

Answer:
There were **22364** tickets issued in ***tickets_1pct*** in 2017. Since the dataset we have is only 1% of the total data, we can expect **2,236,400** in the ***full data*** in 2017. According to the article, however, it indicates there are more than 3 million tickets per year on average.
It could be a meaningful full difference since we are using a sample data to estimate the full dataset, and there will be some deviation. The estimated number accounts for only 74.5% of the exact number, therefore it's likely that the sample data will deviate a bit from the whole dataset, making it a meaningful difference. That being said, since the data is randomly chosen from the whole data set, it's still a good illustration of the data pattern in the full data set and we can use it to do estimation, although it's not 100% precise.

3.2

```
tickets_1pct %>%
  group_by(violation_description) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  head(20) %>%
  ggplot() +
  geom_col(aes(reorder(violation_description, count), count),
    fill = "maroon"
  ) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 25)) +
  labs(
    title = "The TOP 20 Most Frequent Violation Types",
    x = "violation type",
    y = "number of cases",
    caption = "Source: ProPublica"
  ) +
  coord_flip() +
  theme(
    plot.title = element_text(size = 10, face = "bold"),
    plot.caption = element_text(face = "italic", hjust = 0, size = 5),
    axis.text = element_text(size = 5)
  )
```



Answer:
The top 20 most frequent violation types are illustrated in the above plot.

Reference https://stackoverflow.com/questions/21878974/wrap-long-axis-labels-via-labeller-label-wrap-in-

ggplot2

4 joins-unit

4.1

```
sum(is.na(tickets_1pct$unit))
```

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

Answer:
There are 29 tickets in the data set which are unit missing

4.2

```
# Read unit_key
unit_key <- read_csv("unit_key-1.csv",
  show_col_types = FALSE
)
unit_key <- unit_key %>%
  select(-`TABLE FOR WBEZ`, -...5, -...7) %>%
  row_to_names(row_number = 1)
colnames(unit_key)[1] <- "unit"
# Calculate the number of unit
unit_key <- unit_key %>%
  mutate(unit = as.numeric(unit))
n_distinct(unit_key$unit, na.rm = T)
```

```
## [1] 374
```

Answer:
There are 374 units in the *unit_key* data

4.3

```
# ticket in unit
ticket_unit_join <- left_join(tickets_1pct, unit_key, by = "unit")
nrow(ticket_unit_join)
```

```
## [1] 287748
```

```
anti_join(tickets_1pct, unit_key, by = "unit") %>%
  nrow()
```

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

```r
# unit in ticket
left_join(unit_key, tickets_1pct, by = "unit") %>%
  nrow()
```

```
## [1] 287994
```

```r
anti_join(unit_key, tickets_1pct, by = "unit") %>%
  nrow()
```

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

Answer:
There are 287748 rows in the tickets data have a match in the unit table and none is unmatched.(since the orginal data set only has 2877498 rows). However, there are 287994 rows in the unit table have a match in the tickets data while 246 rows are unmatched.

4.4

```r
# DOF & CPD
table(ticket_unit_join$unit_description)
```

```
##
##           CPD   CPD-Airport     CPD-Other           DOF Miscellaneous
##        120712          2617          3760        143909         16442
##  Unidentified
##           308
```

```r
# CPD Departments
ticket_unit_join %>%
  filter(grepl("CPD", unit_description)) %>%
  group_by(`Department Description`) %>%
  summarise(cases = n()) %>%
  arrange(desc(cases)) %>%
  head(5)
```

```
## # A tibble: 5 x 2
##   `Department Description` cases
##   <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
```

Answer:
- Department of Finance issued 143909 tickets in 2017, which is more than that of Chicago Police with 127089 tickets.

- As we can see from the tibble above, Departments located at 1160 N. Larrabee, 6464 N. Clark, OEMC, 3315 W. Ogden, 5555 W. Grand are issuing the most tickets within Chicago Police.

5 Joins - ZIP code

5.1

```
chi_zips <- read.csv("chi_zips.csv")
Sys.getenv("CENSUS_API_KEY")
```

```
## [1] "fae9a3fef407d775a7565ef39141e191b61af0af"
```

```
acs_variables_2014 <- load_variables(2014, "acs1", cache = TRUE)
acs_variables_2014
```

```
## # A tibble: 31,526 x 3
##    name       label                                  concept
##    <chr>      <chr>                                  <chr>
##  1 B00001_001 Estimate!!Total                        UNWEIGHTED SAMPLE COUNT OF~
##  2 B00002_001 Estimate!!Total                        UNWEIGHTED SAMPLE HOUSING ~
##  3 B01001_001 Estimate!!Total                        SEX BY AGE
##  4 B01001_002 Estimate!!Total!!Male                  SEX BY AGE
##  5 B01001_003 Estimate!!Total!!Male!!Under 5 years   SEX BY AGE
##  6 B01001_004 Estimate!!Total!!Male!!5 to 9 years    SEX BY AGE
##  7 B01001_005 Estimate!!Total!!Male!!10 to 14 years  SEX BY AGE
##  8 B01001_006 Estimate!!Total!!Male!!15 to 17 years  SEX BY AGE
##  9 B01001_007 Estimate!!Total!!Male!!18 and 19 years SEX BY AGE
## 10 B01001_008 Estimate!!Total!!Male!!20 years        SEX BY AGE
## # ... with 31,516 more rows
```

```
census_2014 <- get_acs(
  state = "IL",
  geography = "zcta",
  geometry = TRUE,
  variables = c(medincome = "B19013_001", pct_black = "B02001_003", pop = "B01003_001"),
  year = 2014
)
```

```
##   |                                                                              |
```

```
census_2014 <- census_2014 %>%
  select(-moe) %>%
  pivot_wider(names_from = "variable", values_from = "estimate") %>%
  filter(GEOID %in% chi_zips$ZIP)
```

5.2

```
# Clean zipcode
ticket_unit_join$zipcode <- str_sub(ticket_unit_join$zipcode, 1, 5)
colnames(census_2014)[1] <- "zipcode"
# Join census data to the tickets data
ticket_join_final <- inner_join(ticket_unit_join, census_2014, by = "zipcode")
```

5.3

```
# Data for Replicate
table(ticket_join_final$ticket_queue)
```

```
##
##  Bankruptcy       Court      Define   Dismissed Hearing Req      Notice
##        2687         380        3770       13849           9       28899
##        Paid
##      110834
```

```
unpaid_resi_zip <- ticket_join_final %>%
  drop_na(zipcode) %>%
  filter(ticket_queue %in% c("Notice", "Bankruptcy", "Court", "Define")) %>%
  group_by(zipcode) %>%
  mutate(
    unpaid_cases = n(),
    unpaid_per_resi = unpaid_cases / pop
  ) %>%
  select(
    zipcode, unpaid_cases, unpaid_per_resi,
    everything()
  ) %>%
  distinct(zipcode, .keep_all = TRUE) %>%
  arrange(-unpaid_cases)
head(unpaid_resi_zip, 3)
```

```
## # A tibble: 3 x 39
## # Groups:   zipcode [3]
##   zipcode unpaid_cases unpaid_per_resi  ...1 ticket_number issue_date
##   <chr>          <int>           <dbl> <dbl>         <dbl> <chr>
## 1 60623           1875          0.0213   185      50726901 2007/1/3 13:37
## 2 60620           1570          0.0218    70      51581201 2007/1/2 09:10
## 3 60629           1542          0.0134    65      51148501 2007/1/2 08:47
## # ... with 33 more variables: violation_location <chr>,
## #   license_plate_number <chr>, license_plate_state <chr>,
## #   license_plate_type <chr>, violation_code <chr>,
## #   violation_description <chr>, unit <dbl>, unit_description <chr>,
## #   vehicle_make <chr>, fine_level1_amount <dbl>, fine_level2_amount <dbl>,
## #   current_amount_due <dbl>, total_payments <dbl>, ticket_queue <chr>,
## #   ticket_queue_date <chr>, notice_level <chr>, hearing_disposition <chr>, ...
```

Answer:
Neighborhoods with zipcode at 60623, 60620 and 60629 are the three neighborhoods with the most unpaid tickets by looking at the absolute sum.

5.4

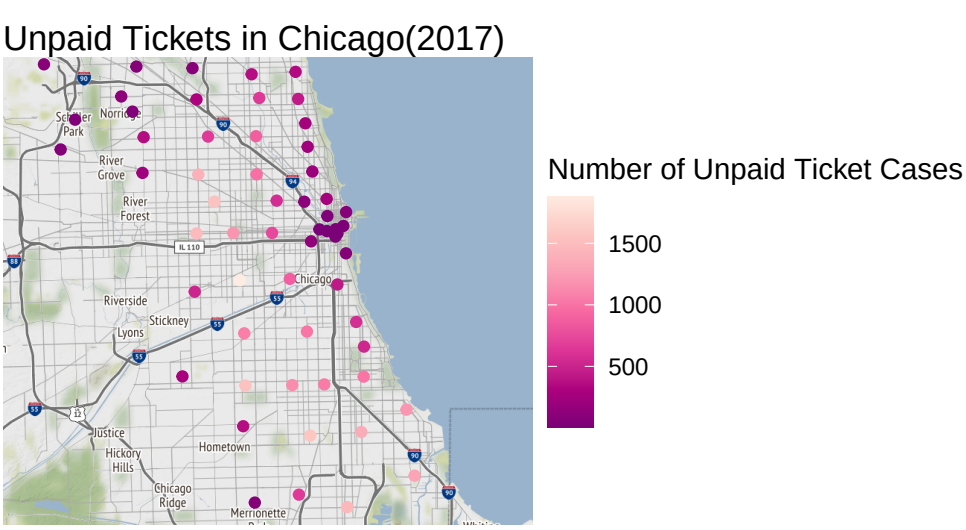
```
chi_bb <- c(
  left = -87.936287,
  bottom = 41.679835,
  right = -87.447052,
  top = 42.000835
)

chicago_stamen <- get_stamenmap(
  bbox = chi_bb,
  zoom = 11
)

geometry <- read_excel("uszips.xlsx")
geometry <- geometry %>%
  filter(city == "Chicago") %>%
  select(zip, lat, lng) %>%
  mutate(zipcode = as.character(zip)) %>%
  select(-1)
unpaid_resi_zip_plot <- left_join(unpaid_resi_zip, geometry, by = "zipcode")
```

Data Source https://simplemaps.com/data/us-zips

```
library(RColorBrewer)
ggmap(chicago_stamen) +
  geom_point(data = unpaid_resi_zip_plot, aes(x = lng, y = lat, color = unpaid_cases)) +
  scale_color_distiller(
    name = "Number of Unpaid Ticket Cases",
    palette = "RdPu"
  ) +
  labs(
    title = "Unpaid Tickets in Chicago(2017)",
    caption = "Source: Census Data, Propublica"
  ) +
  theme(plot.caption = element_text(hjust = 0, face = "italic")) +
  theme_void()
```

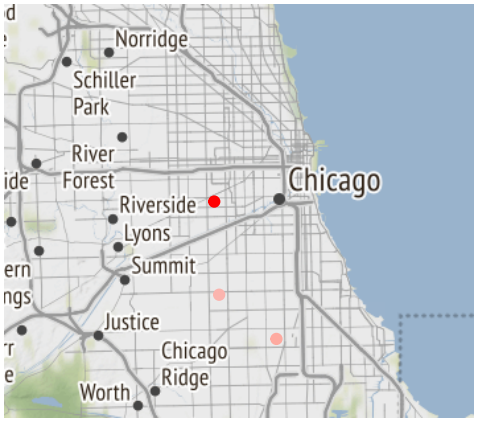


Source: Census Data, Propublica

```
unpaid_resi_zip_small <- unpaid_resi_zip_plot %>%
  filter(zipcode %in% c("60623", "60620", "60629"))

get_stamenmap(
  bbox = chi_bb,
  zoom = 10
) %>%
  ggmap(chicago_stamen) +
  geom_point(data = unpaid_resi_zip_small, aes(x = lng, y = lat, color = unpaid_cases)) +
  scale_color_continuous(
    low = "#FBB4AE",
    high = "red",
    name = "Unpaid Tickets in Chicago(2017)"
  ) +
  labs(
    title = "Chicago Neighborhood with Highest\nUnpaid Ticket Cases(2017)",
    caption = "Source: Census Data, Propublica"
  ) +
  theme(
    plot.title = element_text(size = 11, face = "bold"),
    plot.caption = element_text(hjust = 0, face = "italic")
  ) +
  theme_void()
```

Chicago Neighborhood with Highest
Unpaid Ticket Cases(2017)



Source: Census Data, Propublica

Reference https://cfss.uchicago.edu/notes/vector-maps/

6 understanding the structure of the data

6.1

```
# All data
unpaid <- ticket_join_final %>%
  drop_na(zipcode) %>%
  filter(ticket_queue %in% c("Notice", "Bankruptcy", "Court", "Define"))

unpaid_violation <- unpaid %>%
  group_by(violation_description) %>%
  mutate(diff_fine = fine_level2_amount / fine_level1_amount) %>%
  arrange(diff_fine) %>%
  filter(diff_fine < 2)

unpaid_violation %>%
  distinct(violation_description, .keep_all = TRUE) %>%
  nrow()
```

```
## [1] 8
```

```
# 100 citations
unpaid_violation %>%
  group_by(violation_description) %>%
  mutate(count = n()) %>%
  filter(count >= 100) %>%
  arrange(-count) %>%
  distinct(violation_description, .keep_all = TRUE) %>%
  select(violation_description, violation_code, diff_fine, count, everything())
```

```
## # A tibble: 2 x 39
```

```
## # Groups:   violation_description [2]
##   violation_descr~ violation_code diff_fine count  ...1 ticket_number issue_date
##   <chr>            <chr>              <dbl> <int> <dbl>         <dbl> <chr>
## 1 PARK OR BLOCK A~ 964130              1.67   295  1067      51672701 2007/1/12~
## 2 DISABLED PARKIN~ 0964050J            1.25   112  1677      51501801 2007/1/21~
## # ... with 32 more variables: violation_location <chr>,
## #   license_plate_number <chr>, license_plate_state <chr>,
## #   license_plate_type <chr>, zipcode <chr>, unit <dbl>,
## #   unit_description <chr>, vehicle_make <chr>, fine_level1_amount <dbl>,
## #   fine_level2_amount <dbl>, current_amount_due <dbl>, total_payments <dbl>,
## #   ticket_queue <chr>, ticket_queue_date <chr>, notice_level <chr>,
## #   hearing_disposition <chr>, notice_number <dbl>, officer <chr>, ...
```

Answer:
It's not true all violations will double in price if unpaid. For those with at least 100 citations, there are 2 types of violation that does not double in price if unpaid. More specifically:

- PARK OR BLOCK ALLEY: the price of ticket increases to 1.67 times as that of the first time
- DISABLED PARKING ZONE: the price of ticket increases to 1.25 times as that of the first time

6.2

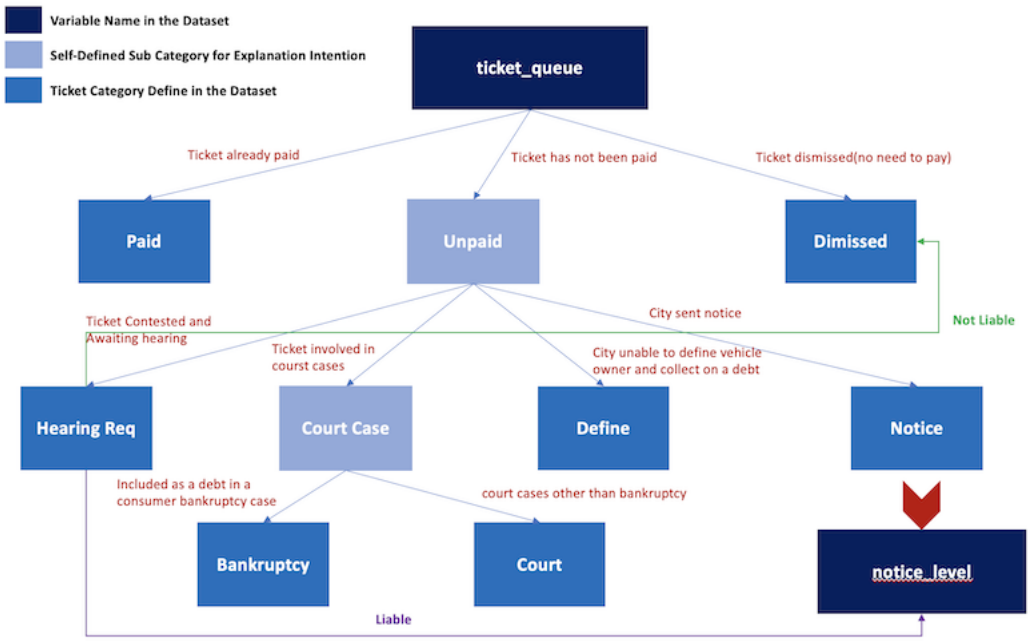
```
table(unpaid_resi_zip$notice_level)
```

```
##
##  DLS FINL SEIZ
##    7    8   37
```

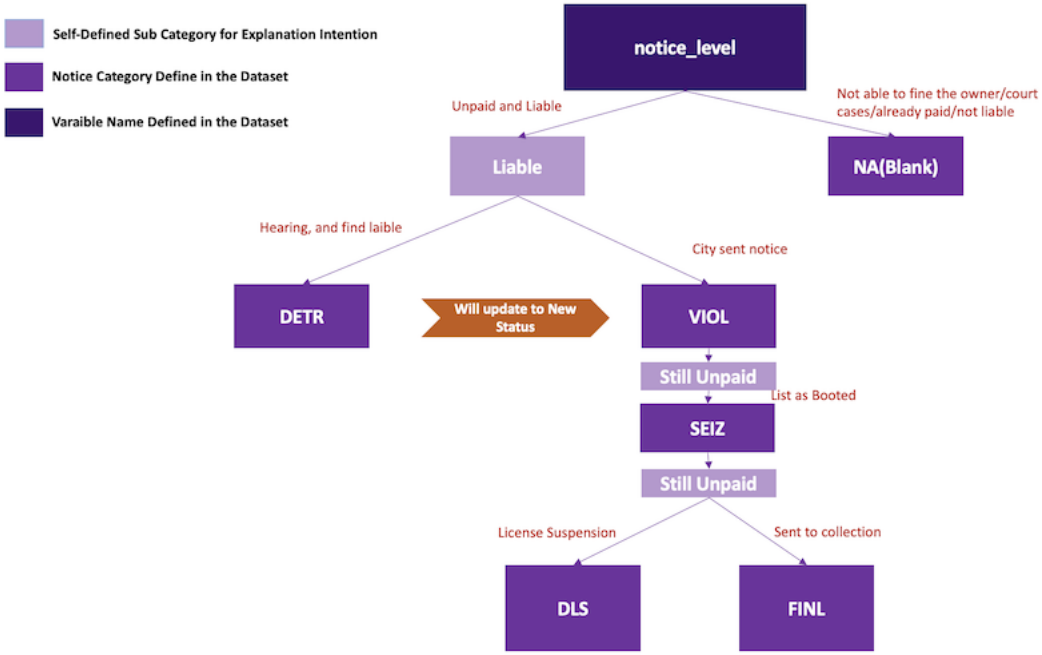
```
table(unpaid_resi_zip$ticket_queue)
```

```
##
## Bankruptcy     Define     Notice
##          4         13         50
```

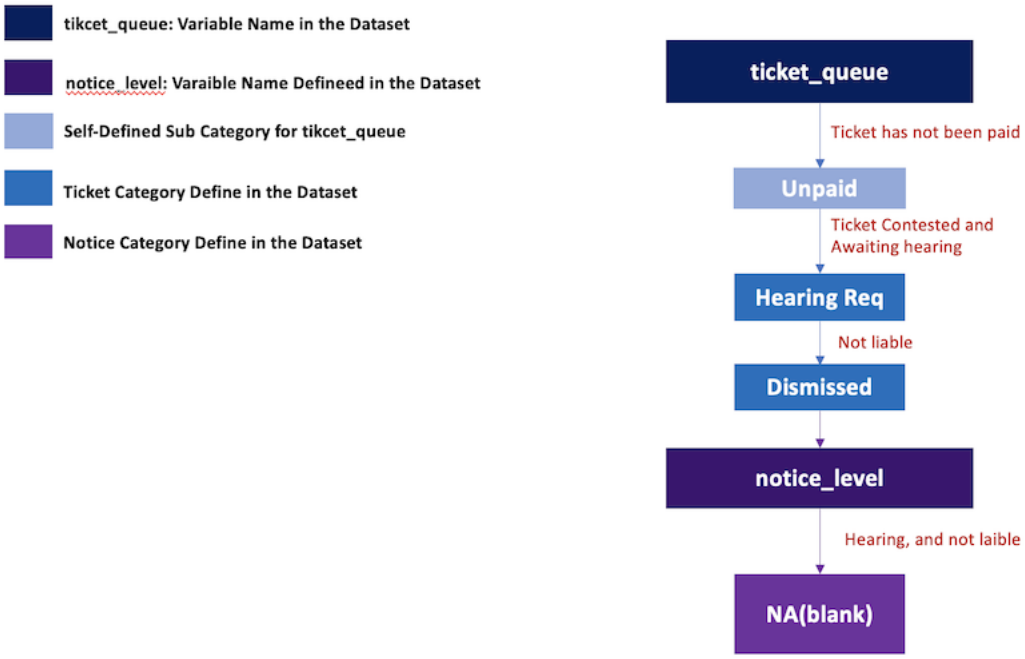
```
knitr::include_graphics("ticket_queue.png")
```




```
knitr::include_graphics("notice_level.png")
```




```
knitr::include_graphics("explain.png")
```



Answer:
If someone contests their ticket and is found not liable, then *notice_level* will be *NA* and *ticket_queue* will be *dismissed*

6.3

```
unpaid %>%
  count(violation_description, violation_code) %>%
  count(violation_description) %>%
  filter(n > 1, !is.na(violation_description))
```

```
## # A tibble: 2 x 2
##   violation_description                  n
##   <chr>                              <int>
## 1 NO CITY STICKER OR IMPROPER DISPLAY     2
## 2 SPECIAL EVENTS RESTRICTION              2
```

Answer:
There are 2 violation description associated with multiple violation codes. THe first one is *NO CITY STICKER OR IMPROPER DISPLAY*, and the second is *SPECIAL EVENTS RESTRICTION*.

6.4

```
mlt_pair <- as.data.frame(
  unpaid %>%
    count(violation_code, violation_description) %>%
    count(violation_code) %>%
```

```
    filter(n > 1, !is.na(violation_code))
)
mlt_pair$violation_code
```

```
## [1] "0964040B" "0964041B" "0964170D" "0964200B" "0976160A" "0976160B" "0980110B"
## [8] "964070"
```

```
unpaid %>%
  group_by(violation_code, violation_description) %>%
  filter(violation_code %in% mlt_pair$violation_code) %>%
  summarise(count = n()) %>%
  arrange(-count)
```

```
## # A tibble: 16 x 3
## # Groups:   violation_code [8]
##    violation_code violation_description                       count
##    <chr>          <chr>                                      <int>
##  1 0964040B       STREET CLEANING                             2199
##  2 0976160A       REAR AND FRONT PLATE REQUIRED               1960
##  3 0976160B       EXPIRED PLATE OR TEMPORARY REGISTRATION      726
##  4 0964040B       STREET CLEANING OR SPECIAL EVENT             288
##  5 0976160A       MISSING/NONCOMPLIANT FRONT AND/OR REAR PLATE  196
##  6 0980110B       HAZARDOUS DILAPITATED VEHICLE                 76
##  7 0976160B       REAR PLATE REQUIRED MOTORCYCLE/TRAILER        40
##  8 0980110B       HAZARDOUS DILAPIDATED VEHICLE                 39
##  9 964070         SNOW ROUTE: 2'' OF SNOW OR MORE               16
## 10 0964041B       SPECIAL EVENTS RESTRICTION                    15
## 11 0964200B       PARK OUTSIDE METERED SPACE                    15
## 12 0964170D       TRUCK OR SEMI-TRAILER PROHIBITED               8
## 13 0964170D       TRUCK TRAILOR/SEMI/TRAILER PROHIBITED          8
## 14 0964041B       Special Events                                 2
## 15 0964200B       OUTSIDE METERED SPACE                          2
## 16 964070         SNOW ROUTE: 2' OF SNOW OR MORE                 2
```

Answer:
There are 8 violation codes associated with multiple violation description, which are *0964040B*, *0964041B*, *0964170D*, *0964200B*, *0976160A*, *0976160B*, *0980110B*, *964070*. Among those description-code pairs, *0964040B-STREET CLEANING* has the most tickets at 2199. The detailed tickets number could be found in the above tibble.

6.5

```
ticket_join_final %>%
  group_by(violation_description) %>%
  mutate(count = n()) %>%
  arrange(desc(count)) %>%
  select(violation_description, violation_code, issue_date, count) %>%
  distinct(violation_description, .keep_all = TRUE) %>%
  head(50) %>%
  arrange(violation_description, issue_date)
```

```
## # A tibble: 50 x 4
## # Groups:   violation_description [50]
##    violation_description                     violation_code issue_date      count
##    <chr>                                     <chr>          <chr>           <int>
##  1 20'OF CROSSWALK                           0964100F       2007/7/20 23:49   240
##  2 3-7 AM SNOW ROUTE                         964060         2007/1/1 04:10    452
##  3 ABANDONED VEH. FOR 7 DAYS OR INOPERABLE   0980110A       2007/1/3 19:00    488
##  4 BLOCK ACCESS/ALLEY/DRIVEWAY/FIRELANE      0964100C       2007/1/8 10:14    912
##  5 CURB LOADING ZONE                         0964160B       2007/1/11 00:45   579
##  6 DISABLED CURB CUT                         0964100D       2007/1/4 12:08    237
##  7 DISABLED PARKING ZONE                     0964050J       2007/1/1 10:50   1245
##  8 DOUBLE PARKING OR STANDING                0964110A       2007/1/1 14:35   1902
##  9 EXP. METER NON-CENTRAL BUSINESS DISTRICT  0964190A       2007/1/2 10:36  10604
## 10 EXPIRED METER CENTRAL BUSINESS DISTRICT   0964190B       2007/1/2 13:25   4133
## # ... with 40 more rows
```

Answer:
There are multiple pairs which seem to include redundant violation description. An example is *EXPIRED PLATES OR TEMPORARY REGISTRATIO & EXPIRED PLATE OR TEMPORARY REGISTRATION*. The violation code is *0976160B* and *0976160F* respectively. And we can notice the earliest issue_date of the first violation description is on 2007/1/1, which is 10 years ahead of the 2nd one on 2017/7/26. Therefore, it reflects the creation of a new violation code.

7 Revenue increase from missing city sticker tickets

7.1

```
sticker <- ticket_join_final %>%
  filter(grepl("CITY STICKER", violation_description)) %>%
  group_by(violation_description) %>%
  mutate(count = n()) %>%
  filter(violation_description != "NO CITY STICKER VEHICLE OVER 16,000 LBS.")

sticker %>%
  select(violation_description, violation_code, issue_date, count,
         fine_level1_amount, everything()) %>%
  distinct(violation_description, .keep_all = TRUE)
```

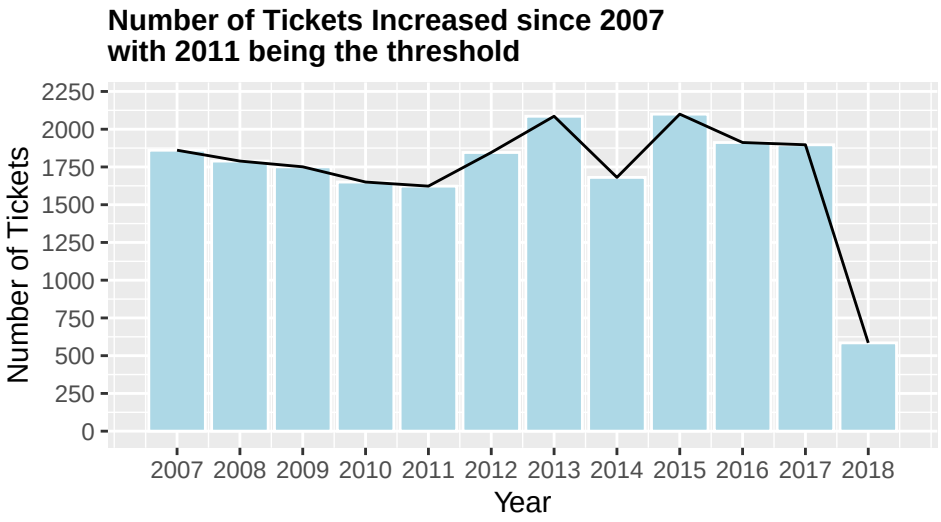
```
## # A tibble: 3 x 38
## # Groups:   violation_description [3]
##   violation_description  violation_code issue_date count fine_level1_amo~   ...1
##   <chr>                  <chr>          <chr>      <int>            <dbl>  <dbl>
## 1 NO CITY STICKER OR IM~ 964125         2007/1/1 ~  8847              120     15
## 2 NO CITY STICKER VEHIC~ 0964125B       2012/2/25~ 11934              200 138605
## 3 IMPROPER DISPLAY OF C~ 0964125D       2012/2/28~   336               30 138872
## # ... with 32 more variables: ticket_number <dbl>, violation_location <chr>,
## #   license_plate_number <chr>, license_plate_state <chr>,
## #   license_plate_type <chr>, zipcode <chr>, unit <dbl>,
## #   unit_description <chr>, vehicle_make <chr>, fine_level2_amount <dbl>,
## #   current_amount_due <dbl>, total_payments <dbl>, ticket_queue <chr>,
## #   ticket_queue_date <chr>, notice_level <chr>, hearing_disposition <chr>,
## #   notice_number <dbl>, officer <chr>, address <chr>, year <dbl>, ...
```

Answer:
The old violation code is *964125*, and the new violation code are *0964125B* and *0964125D* without data from vehicles over 16,000 pounds. Since the cases of *0964125D* only accounts for a small portion of the data, we can drop it. Therefore, the initial offense under the old code is 120 dollars while that under the new code is 200 dollars.

7.2

```
sticker <- sticker %>%
  filter(violation_description != "IMPROPER DISPLAY OF CITY STICKER") %>%
  mutate(violation_description = ifelse(
    violation_code == "0964125B",
    "NO CITY STICKER OR IMPROPER DISPLAY",
    violation_description
  ))

sticker %>%
  ungroup() %>%
  group_by(year) %>%
  mutate(count_date = n()) %>%
  select(violation_description, violation_code, count_date, everything()) %>%
  distinct(year, .keep_all = TRUE) %>%
  ggplot(aes(x = year, y = count_date)) +
  geom_col(
    fill = "lightblue",
    color = "white"
  ) +
  geom_line() +
  scale_x_continuous(
    breaks = seq(2007, 2018, 1),
    limits = c(2006.5, 2018.5)
  ) +
  scale_y_continuous(
    breaks = seq(0, 2250, 250),
    limits = c(0, 2200)
  ) +
  labs(
    title = "Number of Tickets Increased since 2007 \nwith 2011 being the threshold",
    x = "Year",
    y = "Number of Tickets",
    caption = "Source: Census data, ProPublica"
  ) +
  theme(
    plot.title = element_text(size = 11, face = "bold"),
    plot.caption = element_text(hjust = 0, face = "italic")
  )
```



Source: Census data, ProPublica

Answer:

As shown in the plot above, the number of tickets continued to decrease from 2007 to 2011 and reached the lowest value, while the number started to increase unstably since 2012. There is a slightly decreasing trend in recent years. However, since the latest we get in on 2018/5/14, we cannot say the number of tickets in 2018 was decreasing since we only got half of the data.

7.3

```
sticker <- sticker %>%
  mutate(issue_day = as.Date(with(sticker, paste(year, month, day, sep = "-")), "%Y-%m-%d")) %>%
  arrange(issue_day) %>%
  mutate(diff_time = lag(fine_level1_amount) - fine_level1_amount) %>%
  select(diff_time, fine_level1_amount, issue_day, issue_date, everything()) %>%
  arrange(diff_time)
sticker %>%
  filter(diff_time != 0)
```

```
## # A tibble: 1 x 40
## # Groups:   violation_description [1]
##   diff_time fine_level1_amount issue_day  issue_date        ...1 ticket_number
##       <dbl>              <dbl> <date>     <chr>            <dbl>         <dbl>
## 1       -80                200 2012-02-25 2012/2/25 02:00 138605      61529401
## # ... with 34 more variables: violation_location <chr>,
## #   license_plate_number <chr>, license_plate_state <chr>,
## #   license_plate_type <chr>, zipcode <chr>, violation_code <chr>,
## #   violation_description <chr>, unit <dbl>, unit_description <chr>,
## #   vehicle_make <chr>, fine_level2_amount <dbl>, current_amount_due <dbl>,
## #   total_payments <dbl>, ticket_queue <chr>, ticket_queue_date <chr>,
## #   notice_level <chr>, hearing_disposition <chr>, notice_number <dbl>, ...
```

```
sticker %>%
  filter(year == 2012, month == 2, day > 20) %>%
  arrange(day) %>%
  distinct(issue_day, .keep_all = TRUE)
```

```
## # A tibble: 9 x 40
## # Groups:   violation_description [1]
##   diff_time fine_level1_amount issue_day  issue_date        ...1 ticket_number
##       <dbl>              <dbl> <date>     <chr>            <dbl>         <dbl>
## 1         0                120 2012-02-21 2012/2/21 07:45 138378    9180000000
## 2         0                120 2012-02-22 2012/2/22 11:02 138442    9180000000
## 3         0                120 2012-02-23 2012/2/23 12:23 138505      60339801
## 4         0                120 2012-02-24 2012/2/24 08:03 138543    9180000000
## 5       -80                200 2012-02-25 2012/2/25 02:00 138605      61529401
## 6         0                200 2012-02-26 2012/2/26 09:40 138702      61492201
## 7         0                200 2012-02-27 2012/2/27 00:50 138753      61442401
## 8         0                200 2012-02-28 2012/2/28 01:48 138826      61487401
## 9         0                200 2012-02-29 2012/2/29 02:27 138918      61223501
## # ... with 34 more variables: violation_location <chr>,
## #   license_plate_number <chr>, license_plate_state <chr>,
## #   license_plate_type <chr>, zipcode <chr>, violation_code <chr>,
## #   violation_description <chr>, unit <dbl>, unit_description <chr>,
## #   vehicle_make <chr>, fine_level2_amount <dbl>, current_amount_due <dbl>,
## #   total_payments <dbl>, ticket_queue <chr>, ticket_queue_date <chr>,
## #   notice_level <chr>, hearing_disposition <chr>, notice_number <dbl>, ...
```

Answer:
The price increase occurred on 2012-02-25. Before that, no city stickers would be penalized with 120 dollars. Starting from 2012-02-25(including this day), the penalty increased to 200 dollars.

7.4

```
sticker %>%
  filter(year == 2011) %>%
  nrow()
```

```
## [1] 1623
```

```
sticker %>%
  filter(year == 2011) %>%
  group_by(violation_description, ticket_queue) %>%
  count()
```

```
## # A tibble: 5 x 3
## # Groups:   violation_description, ticket_queue [5]
##   violation_description               ticket_queue     n
##   <chr>                               <chr>        <int>
## 1 NO CITY STICKER OR IMPROPER DISPLAY Bankruptcy      53
## 2 NO CITY STICKER OR IMPROPER DISPLAY Define          48
## 3 NO CITY STICKER OR IMPROPER DISPLAY Dismissed      138
## 4 NO CITY STICKER OR IMPROPER DISPLAY Notice         440
## 5 NO CITY STICKER OR IMPROPER DISPLAY Paid           944
```

```
# Payment Rates
944 / 1623
```

```
## [1] 0.5816389
```

```
# Total Revenue
0.5816389 * 1623 * 80 * 100
```

```
## [1] 7551999
```

Answer:
She should projected 7,551,999 dollars revenue increase per year given the condition in this question.

7.5

what impact the change in repayment rates due to the price increase has on revenue, holding all else equal.

Difference: 7,536,000 - 7,280,000 = 256,000

The change in revenue would have been 256,000

```
sticker %>%
  ungroup() %>%
  filter(year == 2013) %>%
  nrow()
```

```
## [1] 2086
```

```
sticker %>%
  ungroup() %>%
  filter(year == 2013 & ticket_queue == "Paid") %>%
  select(ticket_queue_date, fine_level1_amount, current_amount_due, total_payments) %>%
  nrow()
```

```
## [1] 910
```

```
# Repayment Rates
910 / 2086
```

```
## [1] 0.4362416
```

```
# Revenue
200 * 910 * 100
```

```
## [1] 18200000
```

```
# Difference in total revenue
18200000 - 944 * 120 * 100
```

```
## [1] 6872000
```

```
# Difference in revenue increase
7551999 - 80 * 910 * 100
```

```
## [1] 271999
```

Answer:
The repayment rates will decline to 43.62% and the total revenue per year would increase 6872000 dollars per year. However, the per year revenue increase will be 271999 dollars less than that of the calendar yera prior to the new policy.

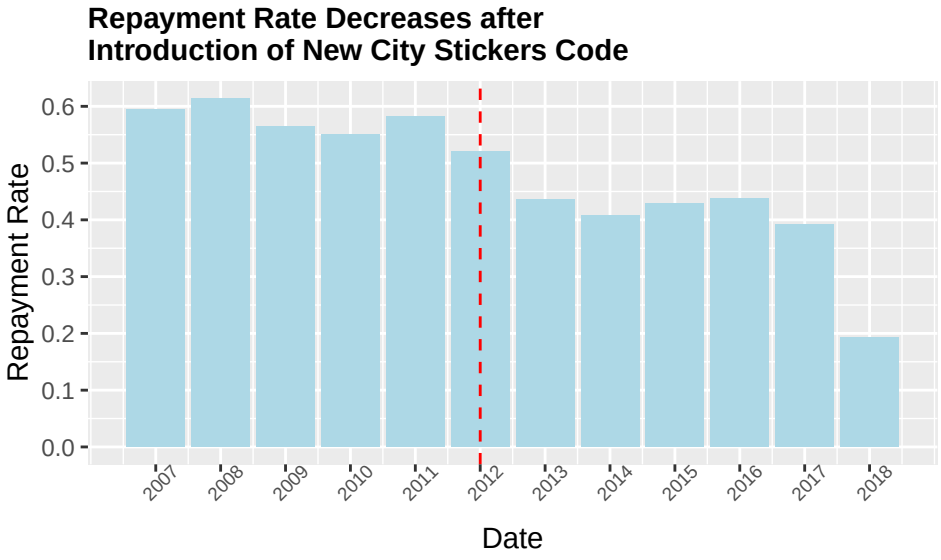
7.6

```
repay_sticker <- sticker %>%
  ungroup() %>%
  mutate(repayment = ifelse(ticket_queue == "Paid",
    1,
    0
  )) %>%
  arrange(issue_day) %>%
  group_by(year) %>%
  mutate(
    ticket_number = n(),
    repay_cases = sum(repayment),
    repay_rate = repay_cases / ticket_number
  ) %>%
  arrange(year) %>%
  select(year, issue_date, repay_rate, fine_level1_amount, current_amount_due,
         total_payments, ticket_queue_date, everything())
head(repay_sticker)
```

```
## # A tibble: 6 x 43
## # Groups:   year [1]
##    year issue_date   repay_rate fine_level1_amo~ current_amount_~ total_payments
##   <dbl> <chr>             <dbl>            <dbl>            <dbl>          <dbl>
## 1  2007 2007/1/1 17~      0.594              120             293.            0
## 2  2007 2007/1/1 10~      0.594              120               0           240
## 3  2007 2007/1/2 10~      0.594              120               0           293.
## 4  2007 2007/1/2 12~      0.594              120             293.            0
## 5  2007 2007/1/2 13~      0.594              120               0           154
## 6  2007 2007/1/3 11~      0.594              120               0           120
## # ... with 37 more variables: ticket_queue_date <chr>, diff_time <dbl>,
## #   issue_day <date>, ...1 <dbl>, ticket_number <int>,
## #   violation_location <chr>, license_plate_number <chr>,
## #   license_plate_state <chr>, license_plate_type <chr>, zipcode <chr>,
## #   violation_code <chr>, violation_description <chr>, unit <dbl>,
## #   unit_description <chr>, vehicle_make <chr>, fine_level2_amount <dbl>,
## #   ticket_queue <chr>, notice_level <chr>, hearing_disposition <chr>, ...
```

```
ggplot(repay_sticker) +
  geom_bar(aes(
```

```
    x = year,
    y = repay_rate
  ),
  stat = "summary",
  fill = "lightblue"
  ) +
  geom_vline(
    xintercept = 2012,
    linetype = 2,
    color = "red"
  ) +
  scale_x_continuous(breaks = seq(2007, 2018, 1)) +
  scale_y_continuous(breaks = seq(0, 1, 0.1)) +
  labs(
    title = "Repayment Rate Decreases after\nIntroduction of New City Stickers Code",
    x = "Date",
    y = "Repayment Rate"
  ) +
  theme(
    plot.title = element_text(face = "bold", size = 11),
    axis.text.x = element_text(angle = 45, size = 7)
  )
```



Answer:

As shown in the plot above, the repayment rates on no city sticker tickets decreased greatly after the introduction of the new policy, i.e. we notice a huge decline in the repayment rate in the following calendar year after 2012. It tells us that the new policy may have some effect on repayment rate but we need more statistical analysis.

7.7

```
ticket_repayment_17 <- ticket_join_final %>%
  mutate(repayment = ifelse(ticket_queue == "Paid",
    1,
    0
```

```
  )) %>%
  filter(year == 2017) %>%
  group_by(violation_description) %>%
  mutate(
    ticket_number = n(),
    repay_cases = sum(repayment),
    repay_rate = repay_cases / ticket_number,
    rev_repay = repay_rate * ticket_number,
    rev_repay_total = rev_repay * fine_level1_amount
  ) %>%
  distinct(violation_description, .keep_all = TRUE) %>%
  arrange(desc(rev_repay_total), desc(rev_repay), desc(repay_rate), desc(ticket_number)) %>%
  select(rev_repay_total, rev_repay, repay_rate, ticket_number, repay_cases,
         fine_level1_amount, current_amount_due)
head(ticket_repayment_17)
```

```
## # A tibble: 6 x 8
## # Groups:   violation_description [6]
##   violation_description       rev_repay_total rev_repay repay_rate ticket_number
##   <chr>                                 <dbl>     <dbl>      <dbl>         <int>
## 1 NO CITY STICKER VEHICLE UN~          148400       742      0.391          1897
## 2 STREET CLEANING                       66300      1105      0.763          1448
## 3 RESIDENTIAL PERMIT PARKING            56325       751      0.712          1055
## 4 EXP. METER NON-CENTRAL BUS~           45150       903      0.731          1236
## 5 PARKING/STANDING PROHIBITE~           43650       582      0.668           871
## 6 EXPIRED PLATES OR TEMPORAR~           33180       553      0.522          1059
## # ... with 3 more variables: repay_cases <dbl>, fine_level1_amount <dbl>,
## #   current_amount_due <dbl>
```

Answer:
We will recommend tickets which have high cases and also high repayment rates, which is represented by the index *rev_repay*. We also need to consider the fine amount since higher penalty means higher increase in revenue per ticket. Then, the evaluation index is *rev_repay_total* in the tibble. As shown in the tibble, the top 3 types of violation which has the highest repay in ticket volume taking repayment rate into consideration of fine level is *NO CITY STICKER OR IMPROPER DISPLAY*, *RESIDENTIAL PERMIT PARKING*, *STREET CLEANING*.
We assume there is no behavioral response. And since year 2017 is the nearest year we get full data, we use data in 2017 to conduct analysis and use the result for government officials to make decision in 2019. Using full data during the past 10 years is less convincing since the fine level and violation type changed during the 10 years.

7.8.a

```
zipcode_income <- ticket_join_final %>%
  group_by(zipcode) %>%
  distinct(medincome, .keep_all = TRUE) %>%
  arrange(-medincome) %>%
  select(zipcode, medincome)
rich <- zipcode_income %>%
  head(10) %>%
  select(zipcode)
```

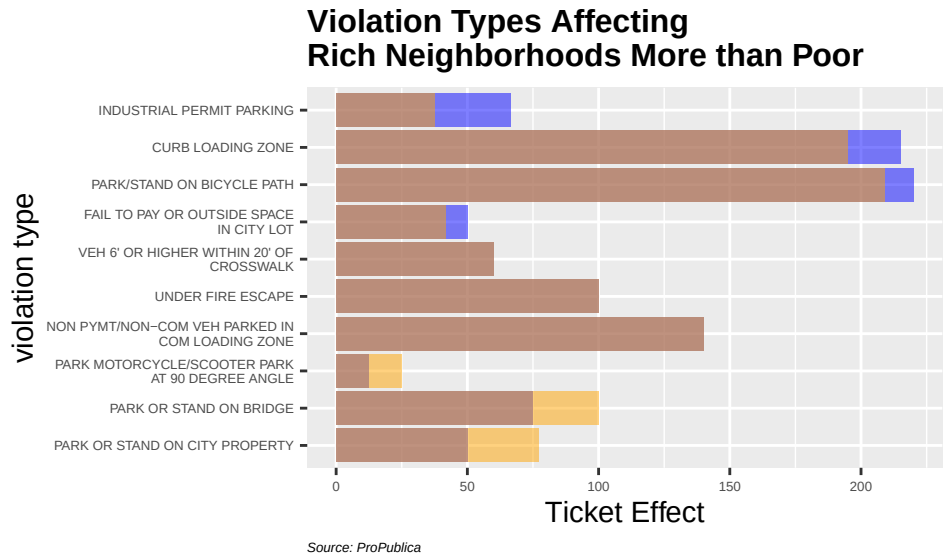
```
poor <- zipcode_income %>%
  tail(10) %>%
  select(zipcode)

ticket_repayment_rich <- ticket_join_final %>%
  filter(zipcode %in% rich$zipcode) %>%
  mutate(repayment = ifelse(ticket_queue == "Paid",
    1,
    0
  )) %>%
  group_by(violation_description, year) %>%
  mutate(
    ticket_number = n(),
    repay_cases = sum(repayment),
    repay_rate = repay_cases / ticket_number,
    rev_repay = repay_rate * ticket_number,
    rev_repay_total = rev_repay * fine_level1_amount
  ) %>%
  distinct(violation_description, year, .keep_all = TRUE) %>%
  ungroup() %>%
  group_by(violation_description) %>%
  mutate(
    avg_rev_repay_total_rich = mean(rev_repay_total),
    avg_rev_repay_rich = mean(rev_repay)
  ) %>%
  arrange(violation_description) %>%
  distinct(violation_description, .keep_all = TRUE) %>%
  select(avg_rev_repay_total_rich, avg_rev_repay_rich, rev_repay_total,
         rev_repay, repay_rate, ticket_number, repay_cases)

ticket_repayment_poor <- ticket_join_final %>%
  filter(zipcode %in% poor$zipcode) %>%
  mutate(repayment = ifelse(ticket_queue == "Paid",
    1,
    0
  )) %>%
  group_by(violation_description, year) %>%
  mutate(
    ticket_number = n(),
    repay_cases = sum(repayment),
    repay_rate = repay_cases / ticket_number,
    rev_repay = repay_rate * ticket_number,
    rev_repay_total = rev_repay * fine_level1_amount
  ) %>%
  distinct(violation_description, year, .keep_all = TRUE) %>%
  ungroup() %>%
  group_by(violation_description) %>%
  mutate(
    avg_rev_repay_total_poor = mean(rev_repay_total),
    avg_rev_repay_poor = mean(rev_repay)
  ) %>%
  arrange(violation_description) %>%
  distinct(violation_description, .keep_all = TRUE) %>%
```

```
  select(avg_rev_repay_total_poor, avg_rev_repay_poor, rev_repay_total,
         rev_repay, repay_rate, ticket_number, repay_cases)

repayment_compare <- left_join(ticket_repayment_rich, ticket_repayment_poor,
                               by = "violation_description")
repayment_compare <- repayment_compare %>%
  mutate(diff_rev = avg_rev_repay_total_rich - avg_rev_repay_total_poor) %>%
  arrange(-diff_rev) %>%
  select(violation_description, diff_rev, everything())
# Visualization
repayment_compare %>%
  head(10) %>%
  ggplot() +
  geom_col(aes(
    x = reorder(violation_description, diff_rev), y = avg_rev_repay_total_rich,
    group = 1
  ),
  fill = "blue",
  alpha = 0.5
  ) +
  geom_col(aes(
    x = reorder(violation_description, diff_rev), y = avg_rev_repay_total_poor,
    group = 1
  ),
  fill = "orange",
  alpha = 0.5
  ) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 30)) +
  coord_flip() +
  labs(
    title = "Violation Types Affecting\nRich Neighborhoods More than Poor",
    x = "violation type",
    y = "Ticket Effect",
    caption = "Source: ProPublica"
  ) +
  theme(
    plot.title = element_text(size = 12, face = "bold"),
    plot.caption = element_text(face = "italic", hjust = 0, size = 5),
    axis.text = element_text(size = 5)
  )
```



Answer

Based on the analysis above, we would suggest she to increase the fine for violation type *NDUSTRIAL PERMIT PARKING*, *CURB LOADING ZONE*, *PARK/STAND ON BICYCLE PATH*, *FAIL TO PAY OR OUTSIDE SPACE IN CITY LOT*, since those violation types affect the richest neighborhoods more on average during the past ten years than that of the poorest communities, using the evaluation index *diff_rev*. *diff_rev* here is the difference of revenue increase generated from increasing tickets among the richest and poorest neighborhoods. We first calculate the ticket numbers, fine level, repayment rate for each violation type during each year, and multiply them to get the revenue generated from tickets. Then, we take the average between year 2007 and 2018 to evaluate the average revenue generated from tickets each year. The *revenue generated from government's perspective* is the *negative effect of tickets from residents' perspective*. Therefore, we choose violation types which generated more average revenue for government in rich communities than poor communities as our recommendation. It's shown in the first plot as those bars which has blue color exceed orange.

Note: we define *rich* by choosing the top 10 communities with the highest median income, and define *poor* by choosing the top 10 communities with the loest median income. The sample size 10 for rich and poor group respectively is also used by the (government)[https://data.cityofchicago.org/Health-Human-Services/below-poverty-level-by-community/b7zw-zvm2]

7.8.b

```
repayment_compare %>%
  head(4) %>%
  mutate(rev_increase = avg_rev_repay_rich * 80 + avg_rev_repay_poor * 80) %>%
  ungroup() %>%
  summarise(sum_rev_increase = sum(rev_increase))
```

```
## # A tibble: 1 x 1
##   sum_rev_increase
##              <dbl>
## 1            1124.
```

The government shoudl expect to gain 1124.364 dollars additional revenue.