

# ps7 extra\_\_credit

*David Franks, Anita Joshi*

*November 20, 2018*

## Extra credit

Repeat plot with all the tickets:

1

```
no_sticker <- read_csv("data_no_sticker.csv")
```

```
## Parsed with column specification:
```

```
## cols(
```

```
##   .default = col_character(),
```

```
##   ticket_number = col_integer(),
```

```
##   issue_date = col_date(format = ""),
```

```
##   zipcode = col_integer(),
```

```
##   unit = col_integer(),
```

```
##   fine_level1_amount = col_integer(),
```

```
##   fine_level2_amount = col_integer(),
```

```
##   current_amount_due = col_double(),
```

```
##   total_payments = col_double(),
```

```
##   ticket_queue_date = col_datetime(format = ""),
```

```
##   notice_number = col_double(),
```

```
##   year = col_integer()
```

```
## )
```

```
## See spec(...) for full column specifications.
```

```
## Warning in rbind(names(probs), probs_f): number of columns of result is not
```

```
## a multiple of vector length (arg 1)
```

```
## Warning: 15 parsing failures.
```

```
## row # A tibble: 5 x 5 col      row col      expected      actual file      expected
```

```
## ... ..
```

```
## See problems(...) for more details.
```

2

For our analysis lets use the 1% sample first, then the full sample.

Read:

```
one_percent <- read_csv("parking_tickets_one_percent.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
```

```
## cols(
```

```
##   .default = col_character(),
```

```
##   X1 = col_integer(),
```

```
##   ticket_number = col_double(),
```

```
##   issue_date = col_datetime(format = ""),
```

```

## unit = col_integer(),
## fine_level1_amount = col_integer(),
## fine_level2_amount = col_integer(),
## current_amount_due = col_double(),
## total_payments = col_double(),
## ticket_queue_date = col_date(format = ""),
## notice_number = col_double()
## )

## See spec(...) for full column specifications.
data_2016_on <- read_csv("data_2016_on")

## Parsed with column specification:
## cols(
##   .default = col_character(),
##   ticket_number = col_integer(),
##   issue_date = col_date(format = ""),
##   zipcode = col_integer(),
##   unit = col_integer(),
##   fine_level1_amount = col_integer(),
##   fine_level2_amount = col_integer(),
##   current_amount_due = col_double(),
##   total_payments = col_double(),
##   ticket_queue_date = col_datetime(format = ""),
##   notice_number = col_double(),
##   year = col_integer()
## )

## See spec(...) for full column specifications.

## Warning in rbind(names(probs), probs_f): number of columns of result is not
## a multiple of vector length (arg 1)

## Warning: 209 parsing failures.
## row # A tibble: 5 x 5 col      row col      expected      actual file      expected
## ... .....
## See problems(...) for more details.

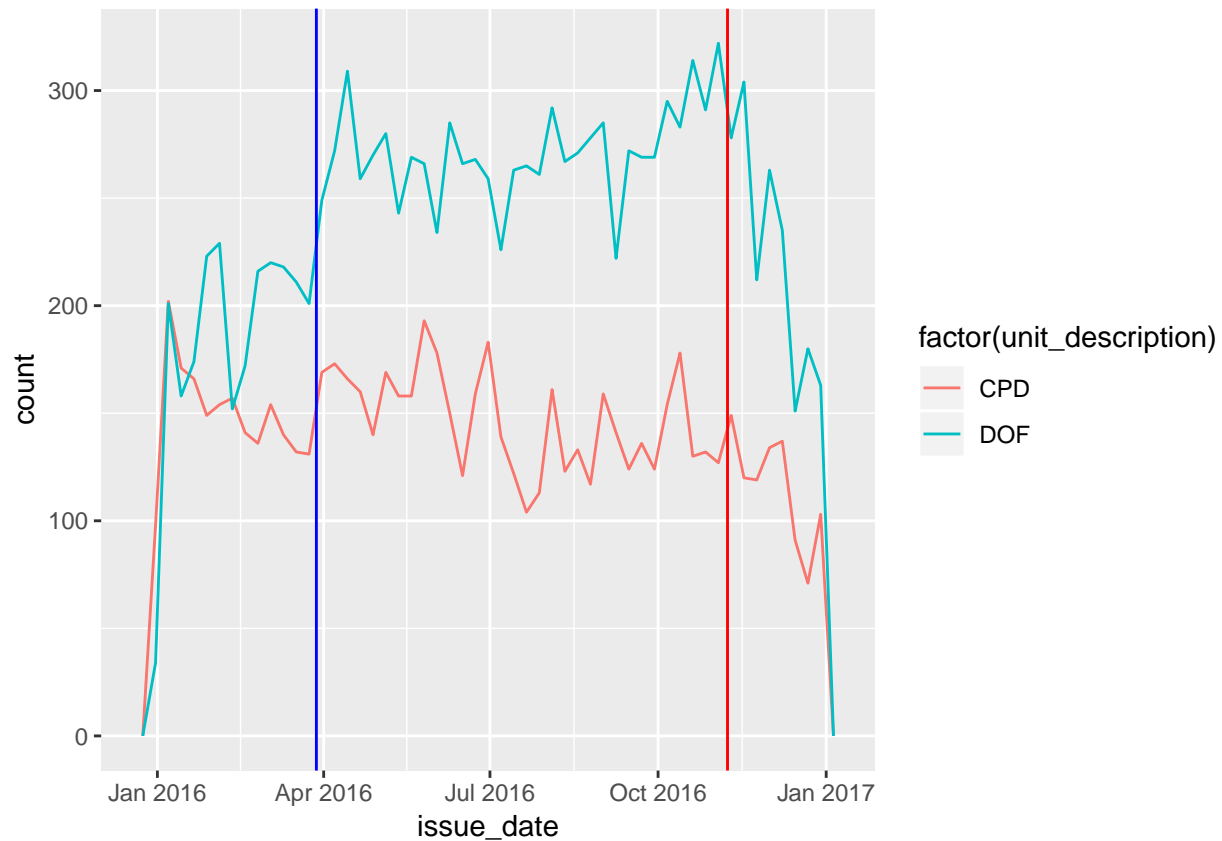
Clean:

one_percent <-
  one_percent %>%
  select(., -X1) %>%
  mutate(paid = as.numeric(str_detect(ticket_queue, "Paid")),
         issue_date = as.Date(issue_date),
         year = lubridate::year(issue_date))

## Warning: package 'bindrcpp' was built under R version 3.4.4

one_percent %>%
  filter(year == 2016 &
         # issue_date > as.Date("2016-01-01") & issue_date < as.Date("2016-06-01")
         unit_description %in% c("CPD", "DOF")) %>%
  ggplot() + geom_freqpoly(aes(x = issue_date, color = factor(unit_description)), binwidth = 7) +
  geom_vline(aes(xintercept = as.Date("2016-03-28")), color = "blue") +
  geom_vline(aes(xintercept = as.Date("2016-11-08")), color = "red")

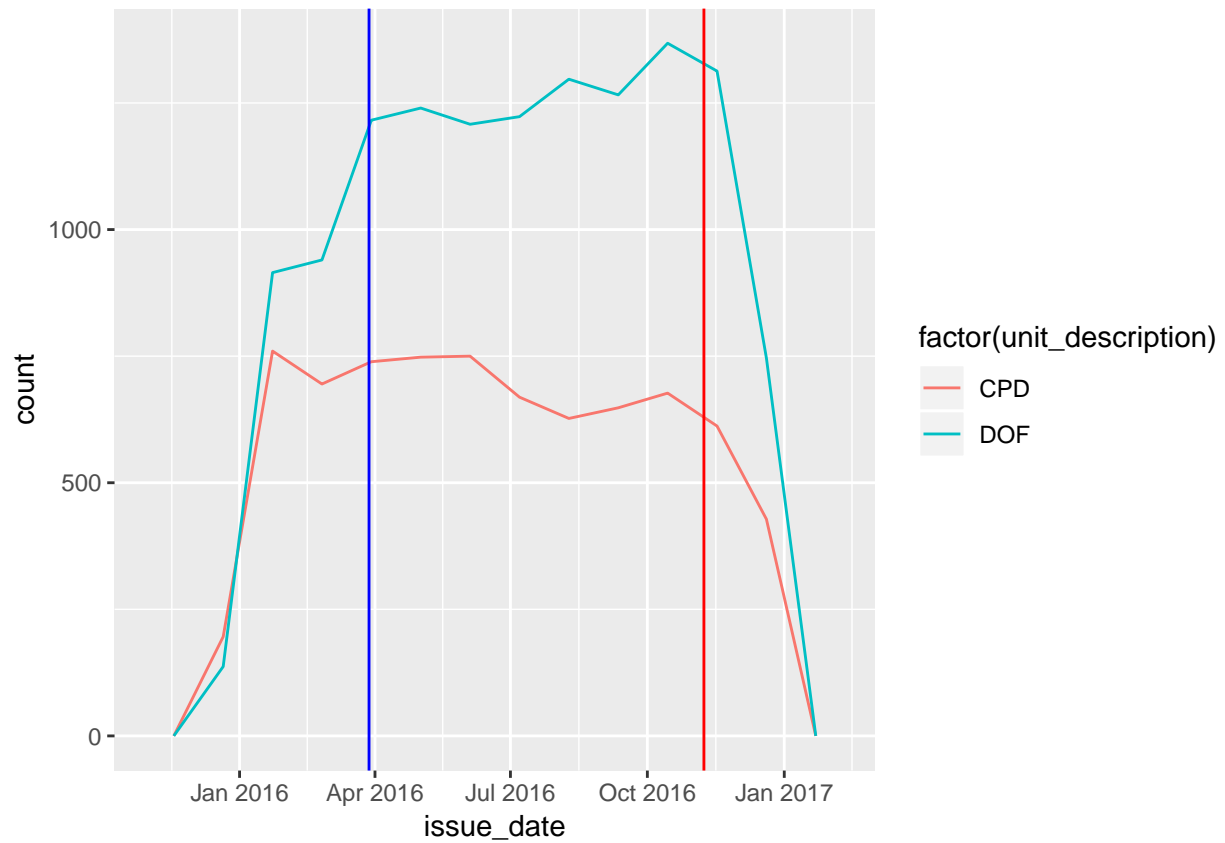
```



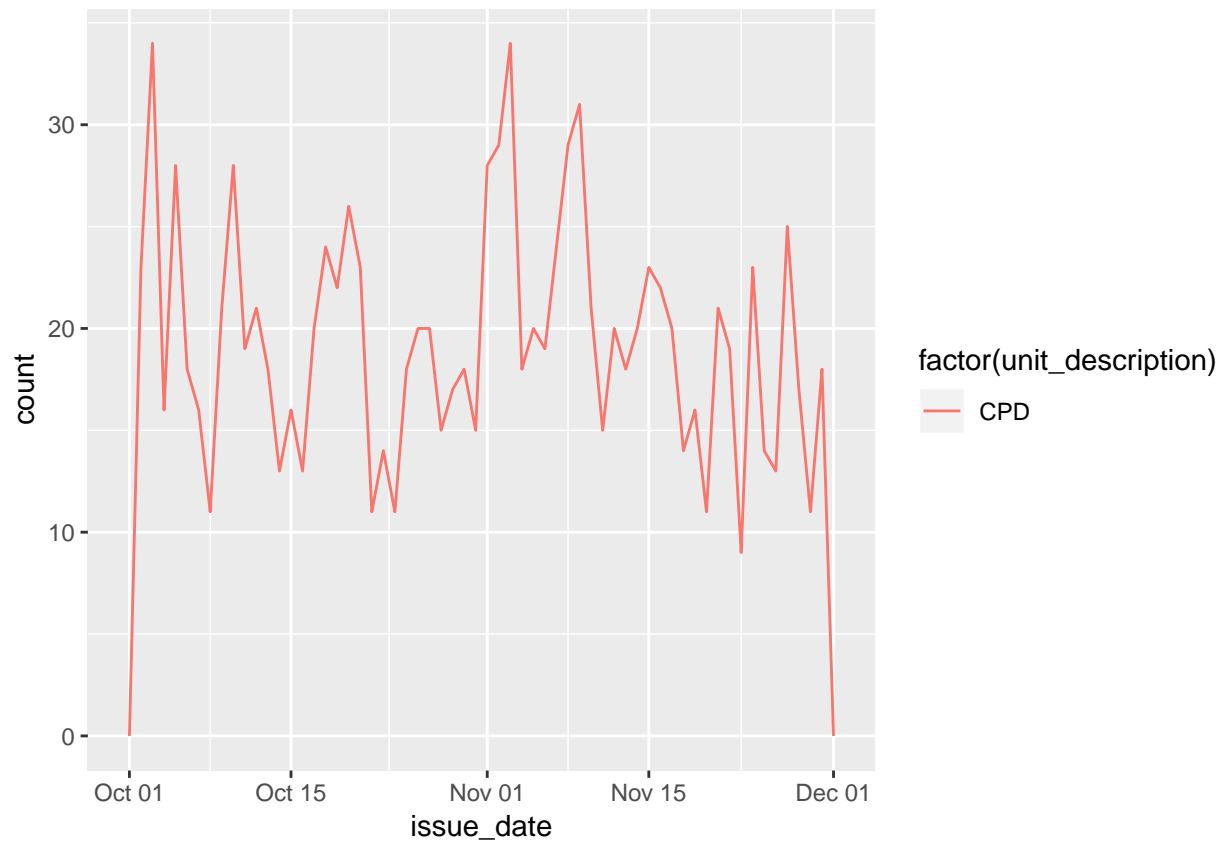
```
one_percent %>%
  group_by(issue_date) %>%
  tally() %>%
  arrange(desc(n))
```

```
## # A tibble: 4,152 x 2
##   issue_date     n
##   <date>       <int>
## 1 2008-02-18     202
## 2 2008-02-20     166
## 3 2008-02-19     159
## 4 2007-07-17     152
## 5 2013-07-16     149
## 6 2009-04-17     147
## 7 2008-07-16     139
## 8 2008-02-22     138
## 9 2008-04-17     138
## 10 2014-10-01     130
## # ... with 4,142 more rows
```

```
one_percent %>%
  filter(year == 2016 &
         # issue_date > as.Date("2016-01-01") & issue_date < as.Date("2016-06-01")
         unit_description %in% c("CPD", "DOF")) %>%
  ggplot() + geom_freqpoly(aes(x = issue_date, color = factor(unit_description)), bins = 12) +
  geom_vline(aes(xintercept = as.Date("2016-03-28")), color = "blue") +
  geom_vline(aes(xintercept = as.Date("2016-11-08")), color = "red")
```



```
one_percent %>%
  filter(unit_description %in% c("CPD") & issue_date > as.Date("2017-10-01") & issue_date < as.Date("2017-12-31")) %>%
  ggplot() + geom_freqpoly(aes(x = issue_date, color = factor(unit_description)), binwidth = 1)
```



```
one_percent %>%
  mutate(
    month = month(issue_date)
  ) %>%
  group_by(month, unit_description, violation_description) %>%
  ggplot() + geom_histogram(aes(x = month, color = violation_description))

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

NO CITY STICKER OR IMPROPER DISPLAY	PARK MOTORCYCLE/SCOOTER PAR
NO CITY STICKER VEHICLE OVER 16,000 LBS.	PARK OR BLOCK ALLEY
NO CITY STICKER VEHICLE UNDER/EQUAL TO 16,000 LBS.	PARK OR STAND IN BUS/TAXI/CARRI
NO DISPLAY OF BACK-IN PERMIT	PARK OR STAND IN VIADUCT/UNDEF
NO OPERATOR SIGNAL	PARK OR STAND NEAR FIRE STATIO
NO OR IMPROPER MUFFLER	PARK OR STAND ON BRIDGE
NO PARK IN PRIVATE LOT	PARK OR STAND ON CHA PROPERT
NO PARK IN PUBLIC LOT	PARK OR STAND ON CITY PROPERT
NO PARKING IN LOOP	PARK OR STAND ON CROSSWALK
NO STANDING/PARKING TIME RESTRICTED	PARK OR STAND ON OR WITHIN 10'
NON PYMT/NON-COM VEH PARKED IN COM LOADING ZONE	PARK OR STAND ON PARKWAY
NONCOMPLIANT PLATE(S)	PARK OR STAND ON SIDEWALK
OBSTRUCT ROADWAY	PARK OR STAND WITHIN INTERSEC
OBSTRUCTED OR IMPROPERLY TINTED WINDOWS	PARK OUTSIDE METERED SPACE
OUTSIDE DIAGONAL MARKINGS	PARK VEHICLE SOLE PURPOSE OF
OUTSIDE METERED SPACE	PARK VEHICLE TO GREASE OR REP
PARK ALLEY	PARK VEHICLE TO SELL MERCHANT
PARK IN CITY LOT OVER 30 DAYS	PARK/STAND IN WRIGLEY BUS PERI
PARK IN CITY LOT WHEN CLOSED	PARK/STAND ON BICYCLE PATH

```
one_percent %>%
  filter(violation_description == "EXPIRED PLATES OR TEMPORARY REGISTRATION") %>% group_by(violation_des
  tally() %>% arrange(desc(n))
```

```
## # A tibble: 43,246 x 3
## # Groups:   violation_description [1]
##   violation_description      license_plate_number      n
##   <chr>                    <chr>                <int>
## 1 EXPIRED PLATES OR TEMPORAR~ 603e09c12c607a2ecfdc8062d4120edd10b2~    518
## 2 EXPIRED PLATES OR TEMPORAR~ 4d4b198e6919d70b3c6d78abc61e18d1f534~      6
## 3 EXPIRED PLATES OR TEMPORAR~ 063be6ca6b136ed4d1ec8bb272ef99c38ca0~      5
## 4 EXPIRED PLATES OR TEMPORAR~ 3cd12a4f08b4aa981012f09e3b262091c069~      5
## 5 EXPIRED PLATES OR TEMPORAR~ 4e0ad89d0504a5949211cf74b27d09c10ae5~      5
## 6 EXPIRED PLATES OR TEMPORAR~ 3691d6797bfc7a0eaf573145954f7504bf47~      4
## 7 EXPIRED PLATES OR TEMPORAR~ 37d1a4b189c5a8a9f15edca9d78a487e6eeb~      4
## 8 EXPIRED PLATES OR TEMPORAR~ 4a500978366012d36bc8d030367731124d8e~      4
## 9 EXPIRED PLATES OR TEMPORAR~ 6e10183be3982ddaa3551f36ce7b95286a47~      4
## 10 EXPIRED PLATES OR TEMPORAR~ a0688b63f712b97004d2953439722d845f13~      4
## # ... with 43,236 more rows

bad_plates <- data_2016_on %>% filter(year == 2017 & violation_description == "EXPIRED PLATES OR TEMPORAR

no_sticker <- data_2016_on %>%
  filter(year == 2017 & violation_description == "NO CITY STICKER VEHICLE UNDER/EQUAL TO 16,000 LBS.") %>%
  select(license_plate_number)
```

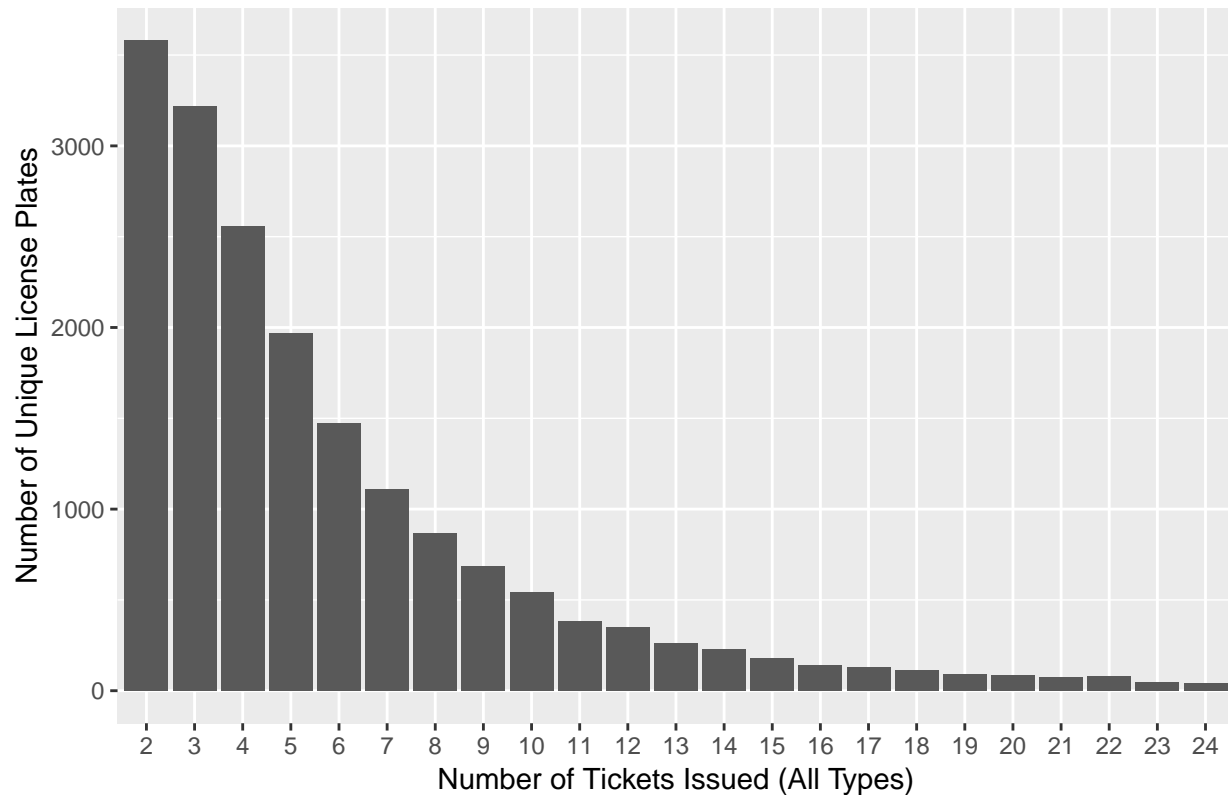
```
doubles <- bad_plates %>% filter(license_plate_number %in% no_sticker$license_plate_number)

plot_me <-

  data_2016_on %>%
  filter(year == 2017 & license_plate_number %in% doubles$license_plate_number) %>% group_by(license_plate_number)

plot_me %>% filter(n < 25) %>% ggplot() + geom_bar(aes(x = factor(n))) +
  labs(y = "Number of Unique License Plates", x = "Number of Tickets Issued (All Types)",
       title = "Count of All Tickets for Double-Ticketed Plates")
```

Count of All Tickets for Double-Ticketed Plates



```
plot_me %>% write_csv("plot_me.csv")

data_2016_on %>%
  mutate(zipcode = str_trim(substr(zipcode, 1, 5))) %>%
  filter(year == 2017 & license_plate_number %in% doubles$license_plate_number) %>% group_by(zipcode) %>%
  summarise(n = n())
```

```
## # A tibble: 439 x 2
##   zipcode      n
##   <chr>    <int>
## 1 <NA>    70993
## 2 60620    4357
## 3 60619    4217
## 4 60628    4132
## 5 60651    4029
## 6 60623    4005
## 7 60629    3939
```

```
## 8 60647 3866
## 9 60617 3745
## 10 60644 3637
## # ... with 429 more rows
```

```
data_2016_on %>%
  mutate(zipcode = str_trim(substr(zipcode, 1, 5))) %>%
  filter(year == 2017) %>% group_by(zipcode) %>% tally() %>% arrange(desc(n))
```

```
## # A tibble: 11,154 x 2
##   zipcode      n
##   <chr>    <int>
## 1 <NA>    382352
## 2 60639    48953
## 3 60629    47061
## 4 60647    47011
## 5 60618    46646
## 6 60623    40800
## 7 60632    40660
## 8 60608    36290
## 9 60651    34489
## 10 60641    34195
## # ... with 11,144 more rows
```

```
data_2016_on %>%
  filter(year == 2017 & license_plate_state == "IL") %>%
  select(license_plate_number) %>%
  distinct()
```

```
## # A tibble: 946,105 x 1
##   license_plate_number
##   <chr>
## 1 07bef58da5eabab43fb335ae30ef7fb6ee02230e34756afd8a3f0217deb41cdc
## 2 3eb394ef9edf5058fc6d435f29247d49417aa2c985e16fb8b0e689281f807ad0
## 3 8475a462bb07ab6af1f8d7d0745edbc30b2dd780fc63da4d66007d662b340d95
## 4 5a49bbe6bf095904d0ce194470d81c0ebf8f352b01cd2236a207006dda0363b
## 5 77f77b0713cc7eb0c47ad5bb0d614e255034ca74beb91a5d2379e5a8642ae1db
## 6 603e09c12c607a2ecfdc8062d4120edd10b2f5499d76fb4cc5d7a8ec73f9e04d
## 7 2fa50964a6352bb0b8533bd0406366de95a3b354398e3aecd0042557a5d04e4f
## 8 4f1d5fc829c305de3507cbf62313e01968f4a75706ab315e813fead1f07ec5d9
## 9 9235b626f12f0b59bd94edfc2dc73446520e51524c8ea46d230c5b557dbfc0c3
## 10 5607eeea5097e3005a426d73ab50410978ff2469910b1e9b1378a7ffadf27ebc
## # ... with 946,095 more rows
```

```
data_2016_on %>%
  filter(year == 2017 & license_plate_number %in% doubles$license_plate_number & str_detect(violation_d
```

```
## # A tibble: 4,647 x 1
##   license_plate_number
##   <chr>
## 1 603e09c12c607a2ecfdc8062d4120edd10b2f5499d76fb4cc5d7a8ec73f9e04d
## 2 efa2070472920671da66b36d3d20dd2d43ff71d3abdc7b58c296cb352e46f5d2
## 3 1846c3e0f4fe41a6d653133094286987d668a43d9318111970263016e099a531
## 4 7816e93c21f8dc9e7d11f130158693d390299bef0764f9835807fb24a1000ecd
## 5 603e09c12c607a2ecfdc8062d4120edd10b2f5499d76fb4cc5d7a8ec73f9e04d
## 6 efa2070472920671da66b36d3d20dd2d43ff71d3abdc7b58c296cb352e46f5d2
```



```
## 7 603e09c12c607a2ecfdc8062d4120edd10b2f5499d76fb4cc5d7a8ec73f9e04d
## 8 603e09c12c607a2ecfdc8062d4120edd10b2f5499d76fb4cc5d7a8ec73f9e04d
## 9 603e09c12c607a2ecfdc8062d4120edd10b2f5499d76fb4cc5d7a8ec73f9e04d
## 10 953e4297a3c85b22b6fc8738f838f70d122c4c07a3464e842aba885696e3f0e4
## # ... with 4,637 more rows

data_2016_on %>%
  filter(year == 2017 & license_plate_number %in% doubles$license_plate_number)%>%
  mutate(zipcode = str_trim(substr(zipcode, 1, 5))) %>%
  select(zipcode, license_plate_number) %>%
  distinct() %>%
  group_by(zipcode) %>%
  tally() %>% arrange(desc(n))

## # A tibble: 439 x 2
##   zipcode      n
##   <chr>    <int>
## 1 <NA>      2599
## 2 60628       730
## 3 60620       709
## 4 60629       670
## 5 60619       649
## 6 60617       638
## 7 60647       621
## 8 60623       606
## 9 60651       587
## 10 60639       583
## # ... with 429 more rows
```