Criminal Justice Analytics Final Project: 2017 Los Angeles Parking Citations

Pauline I. Alvarado

12/13/2018

This final project was submitted for University of Pennsylvania's Criminal Justice Data Science Course taught by Dr. Gregory Ridgeway, who conducted the one-on-one code review. Data consists of parking citations from the city of Los Angeles in 2017 taken from the LA Open Data Website. Available data, to date (2010-2018), consists of more than 8 million entries. The filter function on the website was used to download only 2017 data. California city shapes, California Census tracts, and Los Angeles County Rods were acquired from the Census website.

Research Question

In which areas in Los Angeles is a person most likely to get a ticket from an expired parking meter?

Data Management

Load Relevant Packages and Review Data

```
# Load package
  library(sqldf)

# Review data
  scan("parking-citations-2017.csv", what = "", nlines = 10, sep = "\n")
```

```
##  [1] "Ticket number,Issue Date,Issue time,Meter Id,Marked Time,RP State Plate,Plate Expiry Date,VIN,
##  [2] "1105449822,01/01/2017,1215,,,CA,201408,,MIST,PA,BL,WHITNALL HWY/CALIFA,00015,1,099,5204,25,999
##  [3] "1105797545,01/01/2017,1325,,,CA,201209,,ACUR,PA,WH,10043 HALBRENT AVE,A75,1,4000A1,NO EVIDENCE
##  [4] "1107542785,01/01/2017,1110,,,OR,201016,,LEX,PA,WH,LAUREL CYN BLVD/FRYM,15A67,1,8069B,NO PARKIN
##  [5] "1107593841,01/01/2017,2300,,,CA,201710,,CHRY,PA,BK,WHITLEY/HOLLYWOOD,,1,8056E4,RED ZONE,93,999
##  [6] "1107637624,01/01/2017,1335,,,CA,201608,,CHEV,TR,BR,1790 BAKER ST,1SL1,1,5204A,EXPIRED TAGS,25,
##  [7] "1107637635,01/01/2017,1350,,,CA,201607,,DODG,PA,WH,1009 N MAIN,1SL1,1,5204A,EXPIRED TAGS,25,64
##  [8] "1107637646,01/01/2017,1445,,,CA,201604,,KIA,PA,GY,944 AVILA ST,1SL1,1,5204A,EXPIRED TAGS,25,64
##  [9] "1107671213,01/01/2017,200,,,CA,201710,,NISS,PA,BL,BEDFORD/GUITHRIE,,1,8056E4,RED ZONE,93,99999
## [10] "1108187150,01/01/2017,2025,,,CA,201703,,TOYO,PA,RE,S FIGUEROA ST & CAME,1FB96,1,8069A,NO STOPP
```

Clean Up the File

Space between words need to be moved for variables. Multiple commas indicate no data and need to be converted to NULL.

```
  # Set up ability to read the file
    infile  <- file("parking-citations-2017.csv",       'r')
    outfile <- file("parking-citations-2017-clean.csv", 'w')

  # Clean up column names
    a <- readLines(infile, n=1) # Get variable names from Row 1
    a <- gsub(",",";",a)
    a <- gsub(" ","",a)
    writeLines(a, con=outfile)
    flush(outfile)

  # Clean the rest of the data with a loop
    c.lines <- 0

    system.time(
      while ((length(a <- readLines(infile, n=100000)) >  0))
        {
          c.lines <- c.lines + length(a)
          print(c.lines)

          # Change all commas to a semi-colon, lookahead looks for paired quotes
          a <- gsub("(,)(?=(?:[^\"]|\"[^\"]*\")*$)",";",a,perl=TRUE)

          # Remove semi-colon for a specific line
          a <- gsub("605; WILSHIRE", "605 WILSHIRE", a)

          # Write the cleaned up data to storage
          writeLines(a, con=outfile)
        }
    )
```

```
## [1] 1e+05
## [1] 2e+05
## [1] 3e+05
## [1] 4e+05
## [1] 5e+05
## [1] 6e+05
## [1] 7e+05
## [1] 8e+05
## [1] 9e+05
## [1] 1e+06
## [1] 1100000
## [1] 1200000
## [1] 1300000
## [1] 1400000
## [1] 1500000
## [1] 1600000
## [1] 1700000
## [1] 1800000
```

```
## [1] 1900000
## [1] 2e+06
## [1] 2100000
## [1] 2200000
## [1] 2254357
```

```
##    user  system elapsed
## 126.902   0.921 128.283
```

```
    close(infile)
    close(outfile)

  # Spotcheck lines
    scan("parking-citations-2017-clean.csv", what="",sep="\n",skip=339410,nlines=3)
```

```
## [1] "4303080316;02/27/2017;919;;;CA;201701;;OLDS;TK;BK;616 57TH ST W;00546;55;80.69BS;NO PARK/STREET
## [2] "4303080320;02/27/2017;926;;;CA;201703;;MERZ;PA;SL;931 54TH ST W;00546;55;80.69BS;NO PARK/STREET
## [3] "4303080331;02/27/2017;927;;;CA;201712;;FORD;CM;WT;931 54TH ST W;00546;55;80.69BS;NO PARK/STREET
```

Build a SQL Database

Set up the database

```
  # Look over the first few rows of the dataset
    scan(what="",file="parking-citations-2017-clean.csv",nlines=5,sep="\n")
```

```
## [1] "Ticketnumber;IssueDate;Issuetime;MeterId;MarkedTime;RPStatePlate;PlateExpiryDate;VIN;Make;BodyS
## [2] "1105449822;01/01/2017;1215;;;CA;201408;;MIST;PA;BL;WHITNALL HWY/CALIFA;00015;1;099;5204;25;9999
## [3] "1105797545;01/01/2017;1325;;;CA;201209;;ACUR;PA;WH;10043 HALBRENT AVE;A75;1;4000A1;NO EVIDENCE
## [4] "1107542785;01/01/2017;1110;;;OR;201016;;LEX;PA;WH;LAUREL CYN BLVD/FRYM;15A67;1;8069B;NO PARKING
## [5] "1107593841;01/01/2017;2300;;;CA;201710;;CHRY;PA;BK;WHITLEY/HOLLYWOOD;;1;8056E4;RED ZONE;93;9999
```

```
    a <- read.table("parking-citations-2017-clean.csv",sep=";",nrows=5,header=TRUE)

  # Run once to set up a database
    # Connect or create a new GSQLite database
      con <- dbConnect(SQLite(), dbname="la-parking-citations-2017.db")
      variabletypes <- dbDataType(con, a)
      if(dbExistsTable(con, "parking")) dbRemoveTable(con, "parking")
      dbWriteTable(con, "parking",
                   "parking-citations-2017-clean.csv",
                   row.names=FALSE,
                   header=TRUE,
                   field.types=variabletypes,
                   sep=";")
      dbListFields(con,"parking")
```

```
##  [1] "Ticketnumber"         "IssueDate"            "Issuetime"
##  [4] "MeterId"              "MarkedTime"           "RPStatePlate"
##  [7] "PlateExpiryDate"      "VIN"                  "Make"
```

```
## [10] "BodyStyle"            "Color"                "Location"
## [13] "Route"                "Agency"               "Violationcode"
## [16] "ViolationDescription" "Fineamount"           "Latitude"
## [19] "Longitude"            "AgencyDescription"    "ColorDescription"
## [22] "BodyStyleDescription"
```

```
    dbDisconnect(con)
```

Format Date/Time Column

Load related package and view column characteristics

```
  # Related package
    library(lubridate)

  # Info on internal data
    con <- dbConnect(SQLite(), dbname="la-parking-citations-2017.db") # reconnect to database
    res <- dbSendQuery(con, "PRAGMA table_info(parking)")
    fetch(res, n = -1)
```

```
##    cid                 name     type notnull dflt_value pk
## 1    0         Ticketnumber      INT       0         NA  0
## 2    1            IssueDate     TEXT       0         NA  0
## 3    2            Issuetime      INT       0         NA  0
## 4    3              MeterId SMALLINT       0         NA  0
## 5    4           MarkedTime SMALLINT       0         NA  0
## 6    5         RPStatePlate     TEXT       0         NA  0
## 7    6      PlateExpiryDate      INT       0         NA  0
## 8    7                  VIN SMALLINT       0         NA  0
## 9    8                 Make     TEXT       0         NA  0
## 10   9            BodyStyle     TEXT       0         NA  0
## 11  10                Color     TEXT       0         NA  0
## 12  11             Location     TEXT       0         NA  0
## 13  12                Route     TEXT       0         NA  0
## 14  13               Agency      INT       0         NA  0
## 15  14        Violationcode     TEXT       0         NA  0
## 16  15 ViolationDescription     TEXT       0         NA  0
## 17  16           Fineamount      INT       0         NA  0
## 18  17             Latitude   DOUBLE       0         NA  0
## 19  18            Longitude   DOUBLE       0         NA  0
## 20  19    AgencyDescription SMALLINT       0         NA  0
## 21  20     ColorDescription SMALLINT       0         NA  0
## 22  21 BodyStyleDescription SMALLINT       0         NA  0
```

```
    dbClearResult(res)
```

Convert to proper date format

Date column is currently categorized as “TEXT”.

```
  # Convert to proper date format
    res <- dbSendQuery(con, "SELECT Ticketnumber, IssueDate FROM parking")
    data <- fetch(res, n = -1)
    dbClearResult(res)

    data$datefix <- mdy(data$IssueDate)
    data$datefix <- as.character(data$datefix)
    data$Date <- NULL

    if(dbExistsTable(con,"DateFix")) dbRemoveTable(con, "DateFix")
    dbWriteTable(con, "DateFix", data, row.names=FALSE)
    dbListTables(con)
```

```
## [1] "DateFix" "parking"
```

```
  # Inspect the new DateFix table
    res <- dbSendQuery(con, "
                            SELECT TicketNumber, IssueDate,
                            datefix,
                            strftime('%w',datefix)
                            FROM DateFix")
    fetch(res, n = 10)
```

```
##    Ticketnumber  IssueDate    datefix strftime('%w',datefix)
## 1    1105449822 01/01/2017 2017-01-01                      0
## 2    1105797545 01/01/2017 2017-01-01                      0
## 3    1107542785 01/01/2017 2017-01-01                      0
## 4    1107593841 01/01/2017 2017-01-01                      0
## 5    1107637624 01/01/2017 2017-01-01                      0
## 6    1107637635 01/01/2017 2017-01-01                      0
## 7    1107637646 01/01/2017 2017-01-01                      0
## 8    1107671213 01/01/2017 2017-01-01                      0
## 9    1108187150 01/01/2017 2017-01-01                      0
## 10   1109110962 01/01/2017 2017-01-01                      0
```

```
    dbClearResult(res)

  # Check if '%w' work
    wday("2017-01-01",label=TRUE)
```

```
## [1] Sun
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

```
  # Drop the old parking table & create a new parking table with formatted dates
      # Rename old table
      res <- dbSendQuery(con, "
                              ALTER TABLE parking RENAME TO parking_old")
                              dbClearResult(res)
      dbListTables(con) # view all existing tables
```

```
## [1] "DateFix"     "parking_old"
```

```
      # List all variables to keep
        a <- dbListFields(con,"parking_old")
        paste(a,collapse=",")
```

```
## [1] "Ticketnumber,IssueDate,Issuetime,MeterId,MarkedTime,RPStatePlate,PlateExpiryDate,VIN,Make,BodyS
```

```
        res <- dbSendQuery(con, "
                                CREATE TABLE parking AS
                                SELECT parking_old.Ticketnumber,
                                        DateFix.datefix AS IssueDate,
                                        parking_old.Issuetime,
                                        parking_old.MeterId,
                                        parking_old.MarkedTime,
                                        parking_old.PlateExpiryDate,
                                        parking_old.VIN,
                                        parking_old.Make,
                                        parking_old.BodyStyle,
                                        parking_old.Color,
                                        parking_old.Location,
                                        parking_old.Route,
                                        parking_old.Agency,
                                        parking_old.Violationcode,
                                        parking_old.ViolationDescription,
                                        parking_old.Fineamount,
                                        parking_old.Latitude,
                                        parking_old.Longitude
                                FROM parking_old,DateFix
                                WHERE parking_old.Ticketnumber=DateFix.Ticketnumber")

      # Look at the new table
        res <- dbSendQuery(con, "
                                SELECT *
                                FROM parking")
        fetch(res, n = 10)
```

```
##    Ticketnumber  IssueDate Issuetime MeterId MarkedTime PlateExpiryDate VIN
## 1    1105449822 2017-01-01      1215                              201408
## 2    1105797545 2017-01-01      1325                              201209
## 3    1107542785 2017-01-01      1110                              201016
## 4    1107593841 2017-01-01      2300                              201710
## 5    1107637624 2017-01-01      1335                              201608
## 6    1107637635 2017-01-01      1350                              201607
## 7    1107637646 2017-01-01      1445                              201604
## 8    1107671213 2017-01-01       200                              201710
## 9    1108187150 2017-01-01      2025                              201703
## 10   1109110962 2017-01-01      1130                                   0
##    Make BodyStyle Color             Location Route Agency Violationcode
## 1  MIST        PA    BL  WHITNALL HWY/CALIFA 00015      1           099
## 2  ACUR        PA    WH   10043 HALBRENT AVE   A75      1        4000A1
## 3   LEX        PA    WH LAUREL CYN BLVD/FRYM 15A67      1         8069B
## 4  CHRY        PA    BK    WHITLEY/HOLLYWOOD            1        8056E4
## 5  CHEV        TR    BR        1790 BAKER ST  1SL1      1         5204A
## 6  DODG        PA    WH          1009 N MAIN  1SL1      1         5204A
```

```
## 7   KIA        PA    GY         944 AVILA ST  1SL1      1         5204A
## 8  NISS        PA    BL     BEDFORD/GUITHRIE            1        8056E4
## 9  TOYO        PA    RE S FIGUEROA ST & CAME 1FB96      1         8069A
## 10  KIA        PA    GY        3200 CANYON DR           4        8056E4
##    ViolationDescription Fineamount Latitude Longitude
## 1                  5204         25    99999     99999
## 2    NO EVIDENCE OF REG         50  6420792   1914804
## 3            NO PARKING         73    99999     99999
## 4               RED ZONE         93    99999     99999
## 5          EXPIRED TAGS         25  6493037   1848237
## 6          EXPIRED TAGS         25  6490775   1845291
## 7          EXPIRED TAGS         25  6491076   1843458
## 8               RED ZONE         93    99999     99999
## 9  NO STOPPING/STANDING         93    99999     99999
## 10              RED ZONE         93  6467477   1868721
```

```
        dbClearResult(res)

      # Look at all the existing tables & drop unecessary tables
        dbListTables(con)
```

```
## [1] "DateFix"     "parking"     "parking_old"
```

```
        res <- dbSendQuery(con, "DROP TABLE parking_old")
        dbClearResult(res)

        res <- dbSendQuery(con, "DROP TABLE DateFix")
        dbClearResult(res)

  # Clean up unused space
    system.time(res <- dbSendQuery(con, "VACUUM"))
```

```
##    user  system elapsed
##   0.533   2.246   3.153
```

Select Data to Analyze

View top 50 violations

```
  # Reconnect to the database
    con <- dbConnect(SQLite(), dbname="la-parking-citations-2017.db")

  # Test query
    res <- dbSendQuery(con, "SELECT COUNT (*), ViolationDescription
                             FROM parking
                             GROUP BY ViolationDescription
                             ORDER BY COUNT(*) DESC
                             LIMIT 50")
    top50_violations <- fetch(res, n = -1)
    dbClearResult(res)
```

```
dbDisconnect(con)

top50_violations
```

```
##    COUNT (*)            ViolationDescription
## 1     646135            NO PARK/STREET CLEAN
## 2     373635                       METER EXP.
## 3     179275                         RED ZONE
## 4     149557             PREFERENTIAL PARKING
## 5     143300                  DISPLAY OF TABS
## 6     102574                       NO PARKING
## 7      71865           PARKED OVER TIME LIMIT
## 8      64874                DISPLAY OF PLATES
## 9      44506                       WHITE ZONE
## 10     40181                 NO STOP/STANDING
## 11     34181                 STANDNG IN ALLEY
## 12     31539                BLOCKING DRIVEWAY
## 13     25904                      YELLOW ZONE
## 14     25229               NO EVIDENCE OF REG
## 15     24372              STOP/STAND PROHIBIT
## 16     23619  NO STOPPING/ANTI-GRIDLOCK ZONE
## 17     22750               PARKED ON SIDEWALK
## 18     21586                    NO STOP/STAND
## 19     21343                  EXCEED 72HRS-ST
## 20     20495                18 IN. CURB/2 WAY
## 21     18574                     FIRE HYDRANT
## 22     14988                   DOUBLE PARKING
## 23     12359              OUTSIDE LINES/METER
## 24     11408             OFF STR/OVERTIME/MTR
## 25      8946         COMM VEH OVER TIME LIMIT
## 26      8048        DISABLED PARKING/NO DP ID
## 27      7817                PARKED IN PARKWAY
## 28      6439                     EXPIRED TAGS
## 29      5165             NO STOPPING/STANDING
## 30      4767                 PRIVATE PROPERTY
## 31      4757                       WHITE CURB
## 32      4692                  PK IN PROH AREA
## 33      4049                    METER EXPIRED
## 34      4046             OVNIGHT PRK W/OUT PE
## 35      3317              18 IN/CURB/COMM VEH
## 36      2828                     PREF PARKING
## 37      2784             RESTRICTED TAXI ZONE
## 38      2752                HANDICAP/NO DP ID
## 39      2672              COMM TRAILER/22 FT.
## 40      2502                           22500H
## 41      2452                  DSPLYPLATE A
## 42      2216               PARKED IN BUS ZONE
## 43      2105              PARKED IN CROSSWALK
## 44      2013                 NO STOP/STAND PM
## 45      1999                 CITY PARK/PROHIB
## 46      1961            BLK BIKE PATH OR LANE
## 47      1846                           22514
## 48      1673 NO PARKING BETWEEN POSTED HOURS
```

```
## 49      1576             WITHIN INTERSECTION
## 50      1558                         RED CURB
```

Create working dataset

```
con <- dbConnect(SQLite(), dbname="la-parking-citations-2017.db")
res <- dbSendQuery(con, "SELECT Latitude,Longitude, ViolationDescription
                         FROM parking
                         WHERE
                               ViolationDescription='METER EXP.' OR
                               ViolationDescription='METER EXPIRED'")
meter_expired_2017 <- fetch(res, n = -1)
dbClearResult(res)
dbDisconnect(con)


# Remove Latitude and Longitude null value of 99999
  meter_expired_2017 <- subset(meter_expired_2017, Latitude!="99999")
```

Plot Expired Meter Citation Data

Ensure both the expired meter data and shapefile are on the coordinate system

```
# Load package to work with simple features files
  library(sf)

# Load map of Los Angeles shapefile
  map_ca <- st_read("tl_2014_06_place/tl_2014_06_place.shp")
```

```
## Reading layer `tl_2014_06_place' from data source `/Users/leng/Dropbox (Personal)/Programming/OG Gitl
## Simple feature collection with 1516 features and 16 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -124.2695 ymin: 32.53417 xmax: -114.229 ymax: 41.99323
## epsg (SRID):    4269
## proj4string:    +proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs
```

```
  map_la <- subset(map_ca, NAMELSAD=="Los Angeles city")

# Add simple features data to meter_expired_2017 dataframe
  meter_expired_shape <- st_as_sf(meter_expired_2017,
                  coords=c("Latitude","Longitude"),
                  crs="+proj=lcc +lat_1=34.03333333333333 +lat_2=35.46666666666667
                       +lat_0=33.5 +lon_0=-118 +x_0=2000000 +y_0=500000.0000000002
                       +ellps=GRS80 +datum=NAD83 +to_meter=0.3048006096012192 no_defs")
    meter_expired_shape<- st_transform(meter_expired_shape, crs = 4269)

# View coordinate systems
  st_crs(map_la)
```

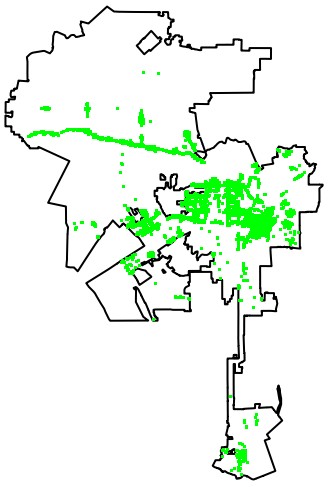
```
## Coordinate Reference System:
##   EPSG: 4269
##   proj4string: "+proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs"
```

```
    st_crs(meter_expired_shape)
```

```
## Coordinate Reference System:
##   EPSG: 4269
##   proj4string: "+proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs"
```

Plot simple features map

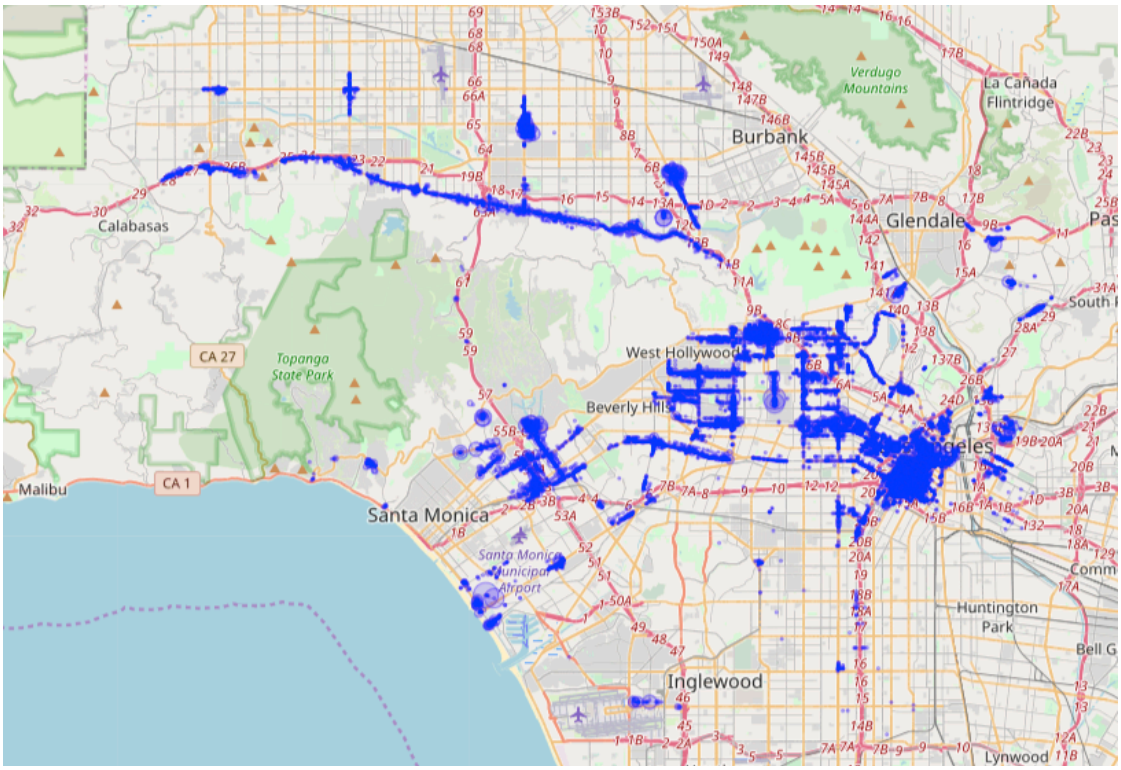
```
    plot(st_geometry(map_la))
    plot(st_geometry(meter_expired_shape), pch=".", col="green", add=TRUE)
```



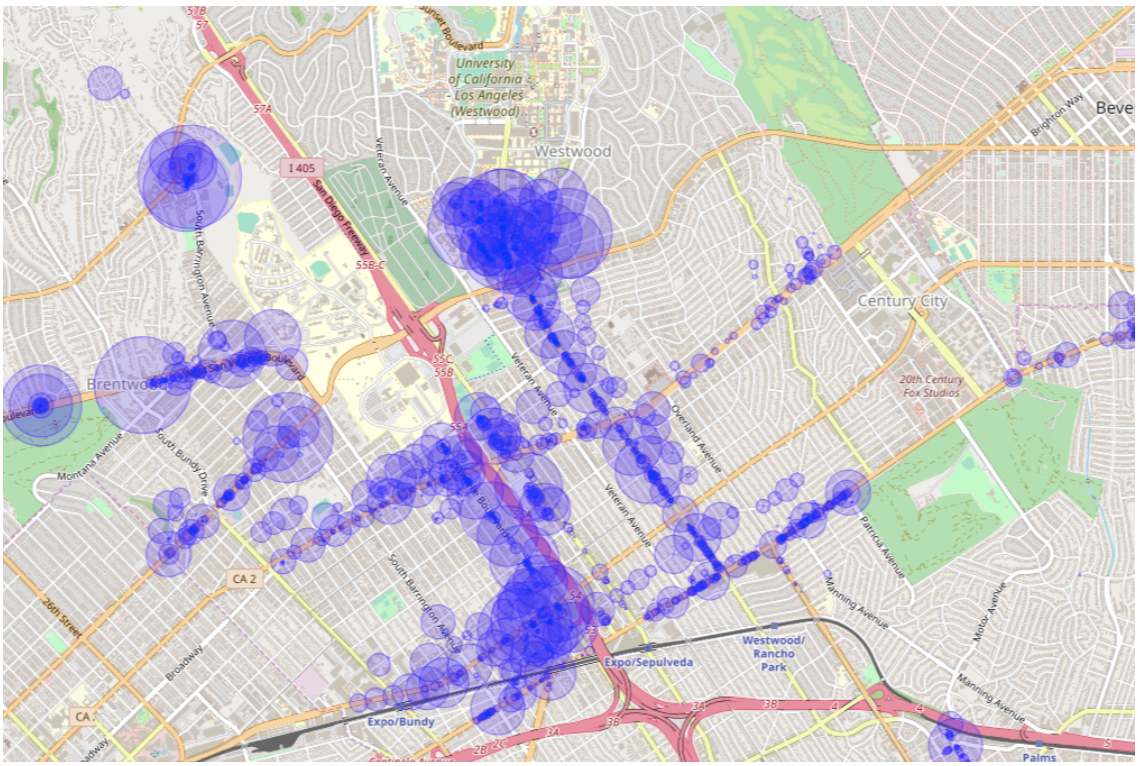
Plot point data on dynamic map using Leaflet

```
 # Load Leaflet
   library(leaflet)

# Extract coordinates
   meter_expired_leaf <- st_coordinates(meter_expired_shape)
   meter_expired_leaf <- data.frame(meter_expired_leaf)
   names(meter_expired_leaf) <- c("longitude","latitude")

# Aggregate by coordinates
  library(tidyverse)
  meter_expired_leaf <- meter_expired_leaf %>% group_by(latitude, longitude) %>% summarise(count = n()

 # Plot data on in a bubble map format
   leaflet(meter_expired_leaf) %>%
   addTiles() %>%
   addCircles(lng = ~longitude, lat = ~latitude, weight = 1, radius = ~sqrt(count) * 10)
```

Bird's Eye View



"Zoomed In"



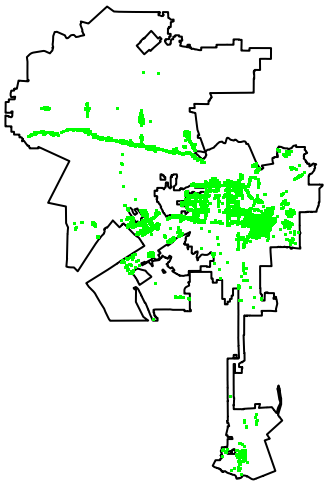
```
## Coordinate Reference System:
##   EPSG: 4269
##   proj4string: "+proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs"
```

```
    st_crs(meter_expired_shape)
```

```
## Coordinate Reference System:
##   EPSG: 4269
##   proj4string: "+proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs"
```

Plot simple features map

```
    plot(st_geometry(map_la))
    plot(st_geometry(meter_expired_shape), pch=".", col="green", add=TRUE)
```



Plot point data on dynamic map using Leaflet

```
 # Load Leaflet
   library(leaflet)

# Extract coordinates
   meter_expired_leaf <- st_coordinates(meter_expired_shape)
   meter_expired_leaf <- data.frame(meter_expired_leaf)
   names(meter_expired_leaf) <- c("longitude","latitude")

# Aggregate by coordinates
  library(tidyverse)
  meter_expired_leaf <- meter_expired_leaf %>% group_by(latitude, longitude) %>% summarise(count = n()

 # Plot data on in a bubble map format
   leaflet(meter_expired_leaf) %>%
   addTiles() %>%
   addCircles(lng = ~longitude, lat = ~latitude, weight = 1, radius = ~sqrt(count) * 10)
```