

# New York City Parking Violations

VISUALIZATION OF TICKET ISSUANCES  
USING R AND GGPLOT2



# Summary



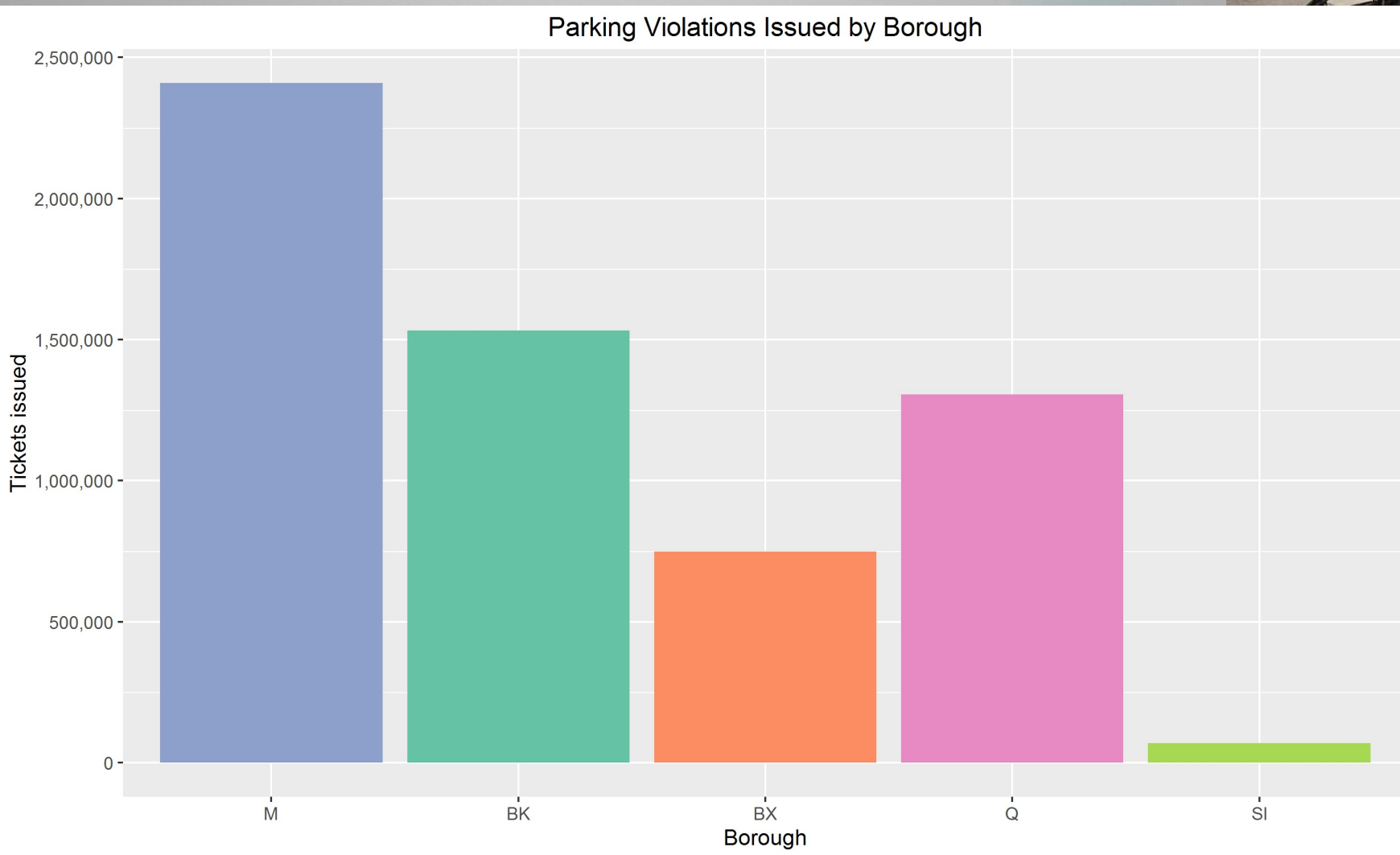
- ▶ New York City offers detailed parking violations information through the NYC Open Data initiative, <https://nycopendata.socrata.com/>
- ▶ This analysis includes tickets issued for FY 2016:  
<https://data.cityofnewyork.us/City-Government/Parking-Violations-Issued-Fiscal-Year-2016/kiv2-tbus>
- ▶ In spite of the title, the data appears to stretch from 1/1/15 thru 12/31/16.
- ▶ The selected subset of data encompasses 7.4 million records.

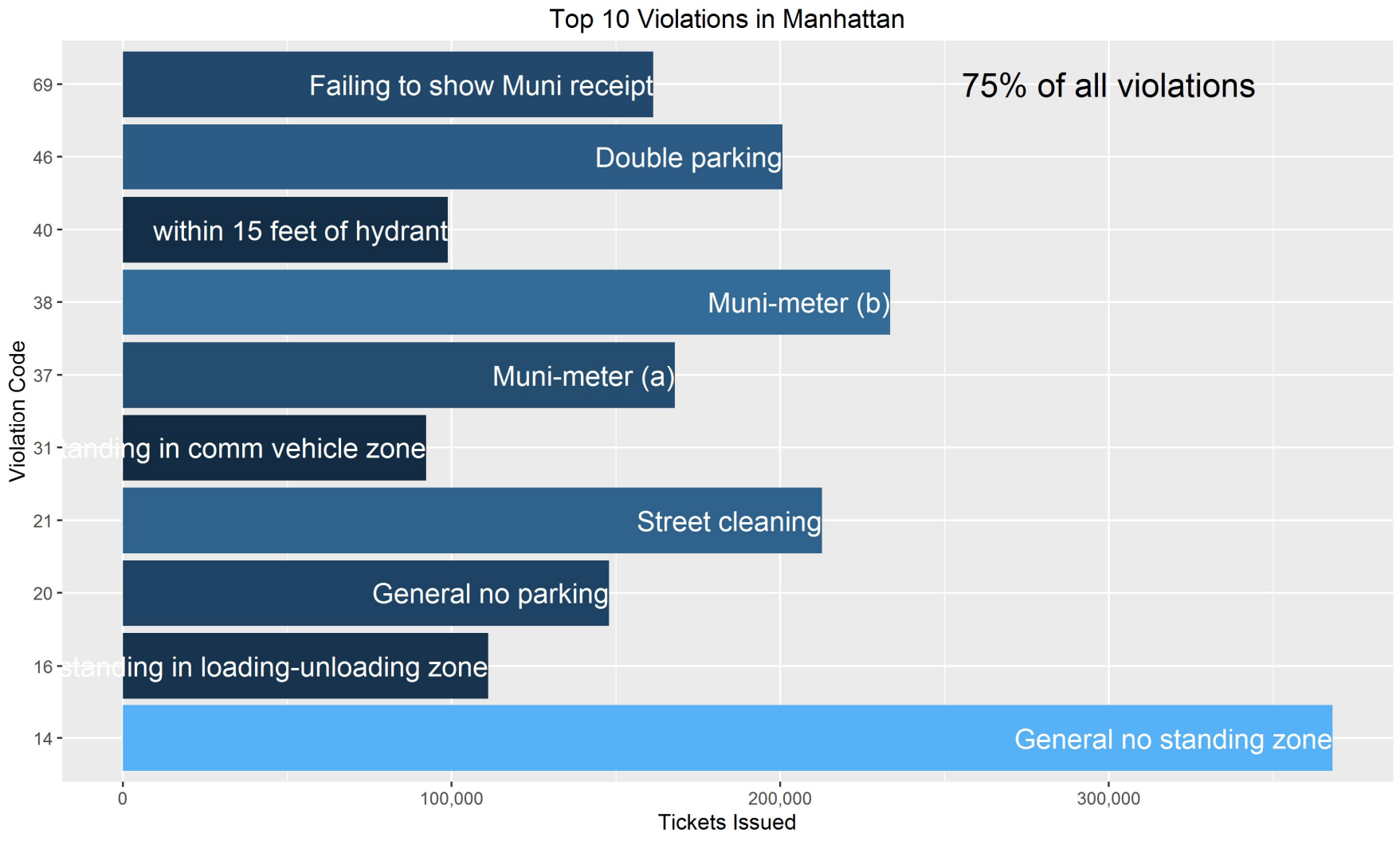


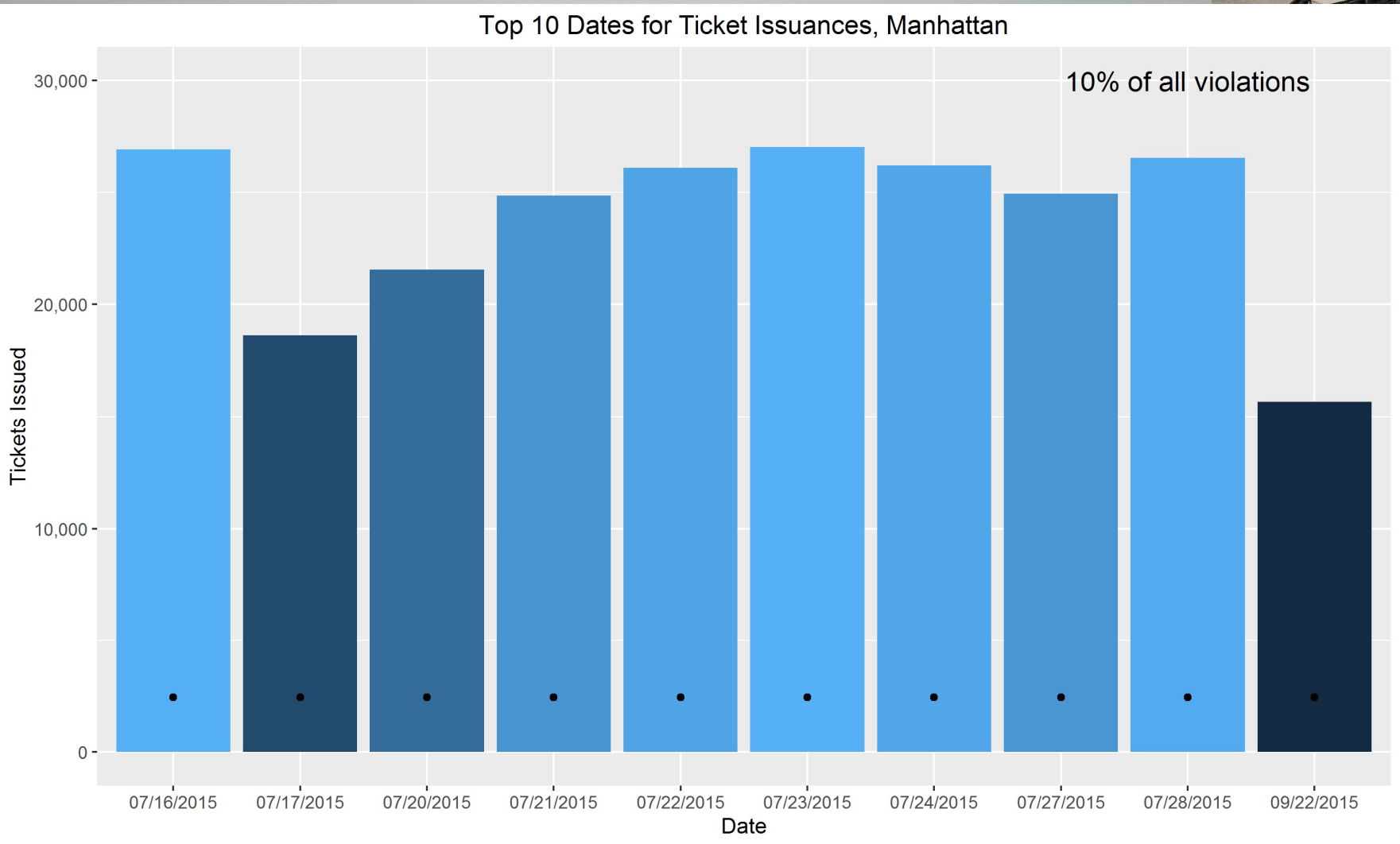
# Issues

- ▶ Entire database is discrete data
  - ▶ Solution: None. I've got no continuous charts...
- ▶ Large database makes data-analysis time consuming
  - ▶ Solution: analysis encompasses Manhattan only, about ½ dataset
- ▶ Data is not clean: Missing values, incorrect factors
  - ▶ Solution: missing values are excluded only if they are plotted in the chart. Otherwise, all records are included in the charts.



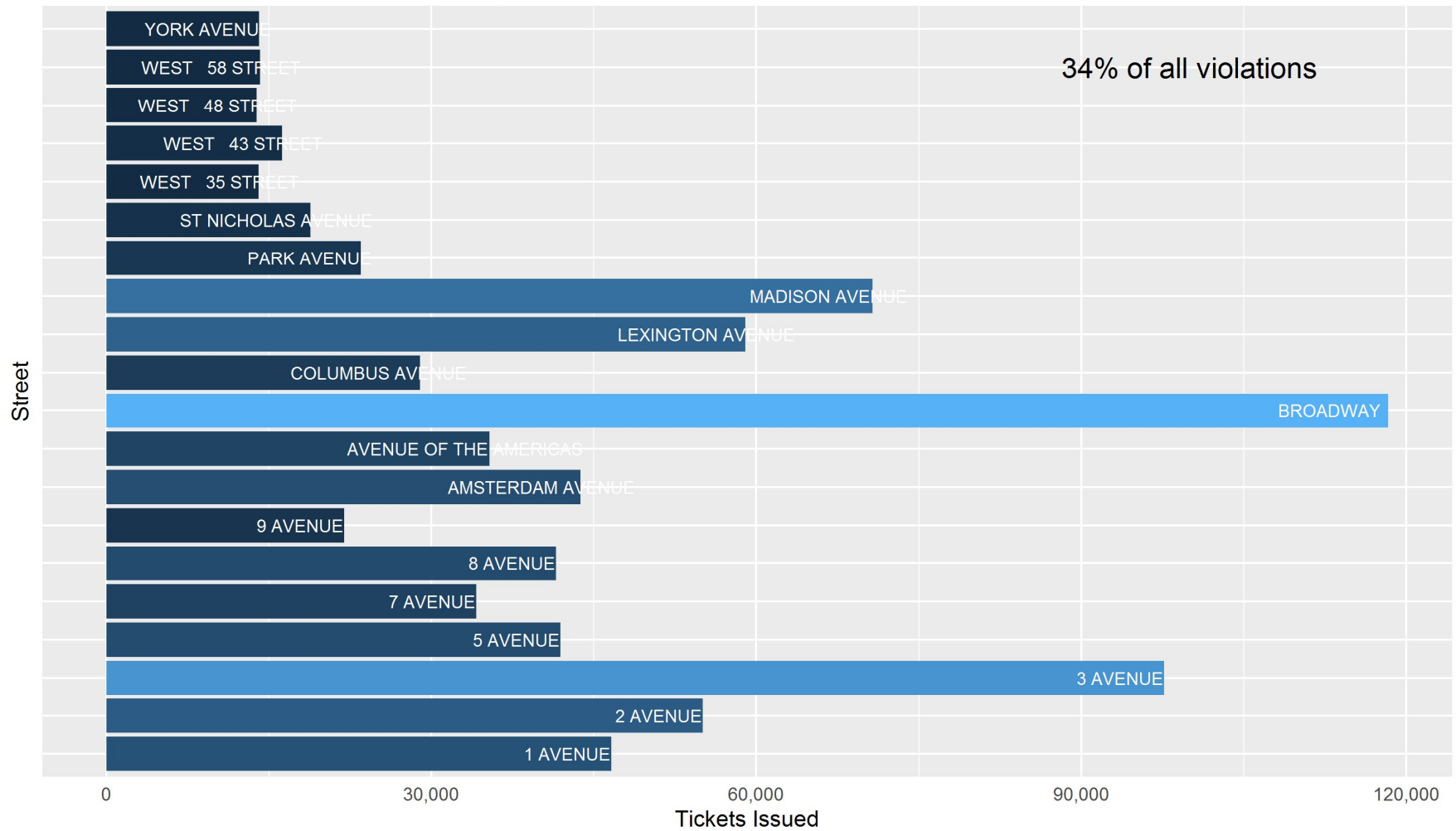




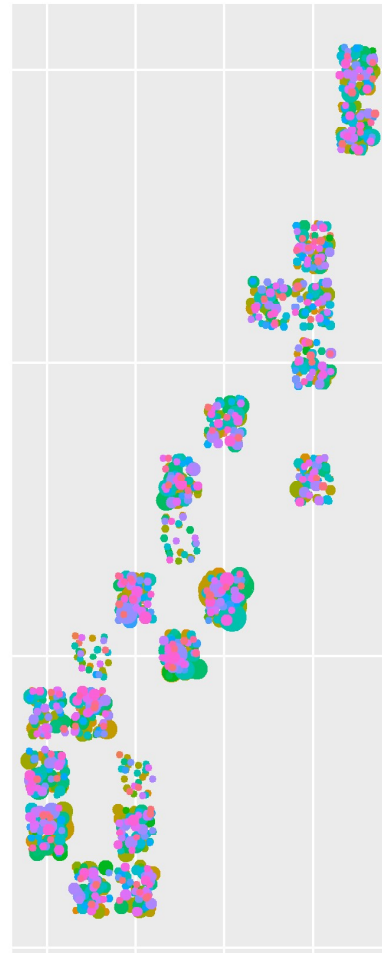




Top 20 Streets for Ticket Issuances, Manhattan



Heat Map for Manhattan Island by Precinct

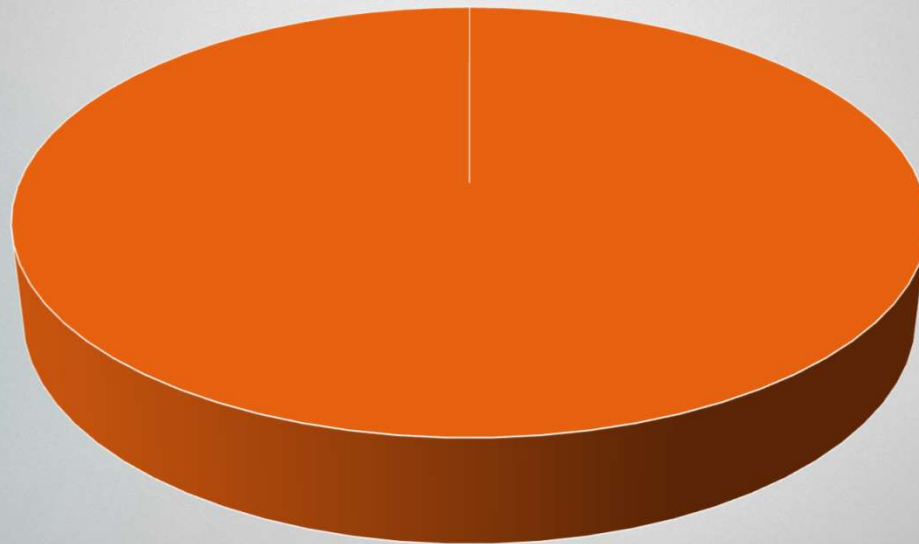




# Recipients of Parking Tickets Happy vs. Unhappy



Recipients of Tickets



■ Happy ■ Unhappy

## A photograph of a severely damaged, rusted-out car body, likely a sedan, with the interior and engine compartment exposed. An orange and white sign is placed on the front of the car.



# Code to Load Data

```
1  setwd("C:/Users/steve/OneDrive/Documents/BootCamp/R Working Dir/Visualization Project")
2  # Load the packages
3  library("dplyr", lib.loc=~R/win-library/3.2")
4  |
5  # READ THE RAW DATA FILES.....
6  PVI <- read.csv("Parking_Violations_Issued_-_Fiscal_Year_2016.csv", stringsAsFactors = FALSE)
7  ViolCodes <- read.csv('DOF_Parking_Violation_codes.csv', stringsAsFactors = FALSE)
8  StreetDB <- read.fwf('snd15Dcow.txt', widths = c(1,1,32,1,1,1,5,2,3,2,1,1,2,32,2,20,1,92))
9
10 names(StreetDB) = c('F1','BoCode','FeatName','Primary','Princ','BoCode2','Street.Code1','GrpCode','SpellVar','F2','NNInd',
11                    'FeatType','LenProg','Prog','MinNameLen','StName2','HTTCode','F3')
12 FeatType <- read.csv('GeoFeatureTypes.csv', stringsAsFactors = FALSE)
13 PctBoroughs = read.csv('PctBorough.csv', stringsAsFactors = FALSE)
14 ShortPVCodes = read.csv('Short PV Codes.csv')
15 # Rename some fields to make the joins easy
16 names(ViolCodes)[3] = 'Manhattan'
17 names(PctBoroughs)[1] = 'Violation.Precinct'
18 names(FeatType)[2] = 'FeatType'
19
20 # Need to convert ViolCodes.Violation.Code to INTEGER
21 ViolCodesa1 = mutate(ViolCodes, Violation.Code = as.numeric(ViolCodes$CODE))
22 ViolCodes2 = left_join(ViolCodesa1, ShortPVCodes, by = c('Violation.Code'='CODE_Num'))
23
24 # JOINS!
25 StreetDB2 = filter(StreetDB, Primary=='P') %>%
26   left_join(., FeatType, by = 'FeatType')
27
28 PVIRemoveColumns = c(31:33, 37:42)
29 PVIA1 = mutate(PVI, DDate = as.Date(Issue.Date, format='%m/%d/%Y'))
30 PVIA2 = select(PVIA1, -PVIRemoveColumns)
31 PVIA3 = left_join(PVIA2, PctBoroughs, by = 'Violation.Precinct')
32 PVIA4 = left_join(PVIA3, ViolCodes2, by = 'Violation.Code')
33
34 PVI2 = left_join(PVIA4, StreetDB2, by = c('Street.Code1', 'BoCode'))
35
36 # Finally, clean up original tables
37 rm(PVI, PVIA1, PVIA2, PVIA3, PVIA4, StreetDB, ViolCodes, ViolCodesa1, FeatType)
38
```



# Code to Create Charts

```
1 library("ggplot2", lib.loc=~R/win-library/3.2")
2 library(RColorBrewer)
3 library(scales)
4
5 # Histogram of tickets issued for all 5 boroughs
6 ggplot(PVI2,aes(x=BoroughNm,fill=BoroughNm)) +
7   geom_bar() +
8   ggtitle('Parking violations Issued by Borough') +
9   xlab('Borough') + ylab('Tickets issued') +
10  scale_x_discrete(limits=c("M","BK","BX","Q","SI")) +
11  scale_y_continuous(labels=comma) +
12  scale_fill_brewer(palette='Set2') +
13  theme(legend.position = 'none')
14  # ggsave('chart1.png',width=10,height=6)
15
16 ManhattanOnly = mutate(filter(PVI2,BoroughNm=='M', Issuer.Code != 0),FPrecinct = as.factor(Violation.Precinct),
17                          VCode = as.factor(Violation.Code))
18 PVI3 = mutate(PVI2,FPrecinct = as.factor(Violation.Precinct),VCode = as.factor(Violation.Code))
19
20 #####
21 # Change this to PVI3 for all #
22 # boroughs, or ManhattanOnly #
23
24 SelectDB = ManhattanOnly
25 SelectDBName = 'Manhattan'
26
27
28 # Top 10 Violation Types issued for Manhattan
29 Top10Violations = group_by(SelectDB,VCode) %>%
30   summarise(Cnt = n()) %>%
31   arrange(desc(Cnt)) %>%
32   slice(1:10)
33
34 PlotLegendTextLookup = group_by(SelectDB,ShortDefn) %>%
35   summarise(Cnt=n()) %>%
36   arrange(desc(Cnt)) %>%
37   slice(1:10)
38
39 PlotLegendText2 = PlotLegendTextLookup$ShortDefn
40 Top10ViolationRatio = round(sum(Top10Violations$Cnt)/nrow(SelectDB) * 100)
41 Top10ViolationText = paste0(Top10ViolationRatio, '% of all violations')
```

# Code to Create Charts, 2

```
42 |
43 ggplot(Top10Violations,aes(x=vCode,y=Cnt,fill=Cnt)) +
44   geom_bar(stat='sum') +
45   geom_text(label=PlotLegendText2,hjust=1,size=5,color='white') +
46   coord_flip() +
47   ggtitle(paste0('Top 10 violations in ',SelectDBName)) +
48   xlab('violation code') +
49   ylab('Tickets Issued') +
50   theme(legend.position = 'none') +
51   annotate("text", x=10, y=300000, label= Top10violationText,size = 6) +
52   scale_y_continuous(labels=comma,breaks=seq(0, 700000, 100000))
53
54   #ggsave('Chart2.png',width=10,height=6)
55
56 # Count of all violations
57 # NOT USED
58 ggplot(SelectDB,aes(x=vCode,fill=vCode)) +
59   geom_bar() +
60   coord_flip() +
61   scale_x_discrete() +
62   theme(legend.position = 'none')
63
64 # Top 10 Dates of Ticket Issuances
65 Top10Dates = group_by(SelectDB,Issue.Date) %>%
66   summarise(Cnt = n()) %>%
67   arrange(desc(Cnt)) %>%
68   slice(1:10)
69
70 Top10DatesViolationRatio = round(sum(Top10Dates$Cnt)/nrow(SelectDB) * 100)
71 Top10DatesviolationText = paste0(Top10DatesViolationRatio, '% of all violations')
72
73 TicketsperDayGrp = group_by(SelectDB,Issue.Date) %>%
74   summarise(Cnt=n())
75 AvgTicketsperDay = rep(mean(TicketsperDayGrp$Cnt),10)
76
77 ggplot(Top10Dates,aes(x=Issue.Date, y = Cnt,fill=Cnt)) +
78   geom_bar(stat='sum') +
79   ggtitle(paste0('Top 10 Dates for Ticket Issuances, ',SelectDBName)) +
80   xlab('Date') +
81   ylab('Tickets Issued') +
82   geom_point(aes(x=Issue.Date,y=AvgTicketsperDay)) +
```



# Code to Create Charts, 3

```
83     theme(legend.position = 'none') +
84     annotate("text", x=9, y=30000, label= Top10DatesviolationText,size=5) +
85     scale_y_continuous(labels=comma,breaks=seq(0, 50000, 10000))
86
87     #ggsave('Chart4.png',width=10,height=6)
88
89 # Violations for all Dates, All Manhattan - NOT USED, for support only
90
91     ggplot(SelectDB,aes(x=Issue.Date,fill=Issue.Date)) +
92     geom_bar() +
93     coord_flip() +
94     theme(legend.position = 'none')
95     #ggsave('Chart6.png',width=10,height=6)
96
97 # Top 20 Street Locations of Issuances
98     Top20Streets = group_by(SelectDB,FeatName) %>%
99     summarise(Cnt = n()) %>%
100     filter(!is.na(FeatName) & !(FeatName == '')) %>%
101     arrange(desc(Cnt)) %>%
102     slice(1:20)
103
104     StreetNames = Top20Streets$FeatName
105     Top20StreetsViolationRatio = round(sum(Top20Streets$Cnt)/nrow(SelectDB) * 100)
106     Top20StreetsViolationText = paste0(Top20StreetsViolationRatio, '% of all violations')
107
108     ggplot(Top20Streets,aes(x=FeatName, y = Cnt,fill=Cnt),na.rm='TRUE') +
109     geom_bar(stat='sum') +
110     coord_flip() +
111     geom_text(label=StreetNames, size=3,color='white') +
112     ggtitle(paste0('Top 20 Streets for Ticket Issuances, ',SelectDBName)) +
113     xlab('Street') +
114     ylab('Tickets Issued') +
115     theme(legend.position = 'none',
116           axis.text.y=element_blank(),
117           axis.ticks = element_blank()) +
118     annotate("text", x=19, y=100000, label= Top20StreetsviolationText,size=5) +
119     scale_y_continuous(label=comma,breaks=seq(0, 120000, 30000))
120     #ggsave('Chart4.png',width=10,height = 6)
121
122
```