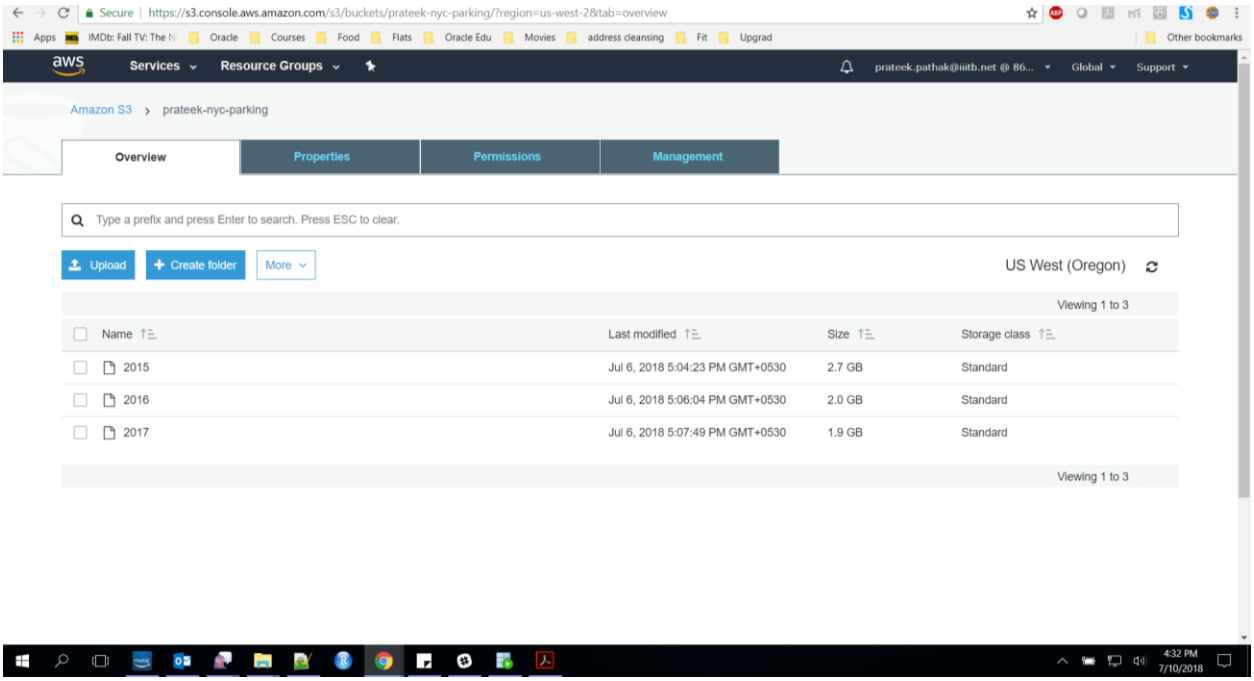
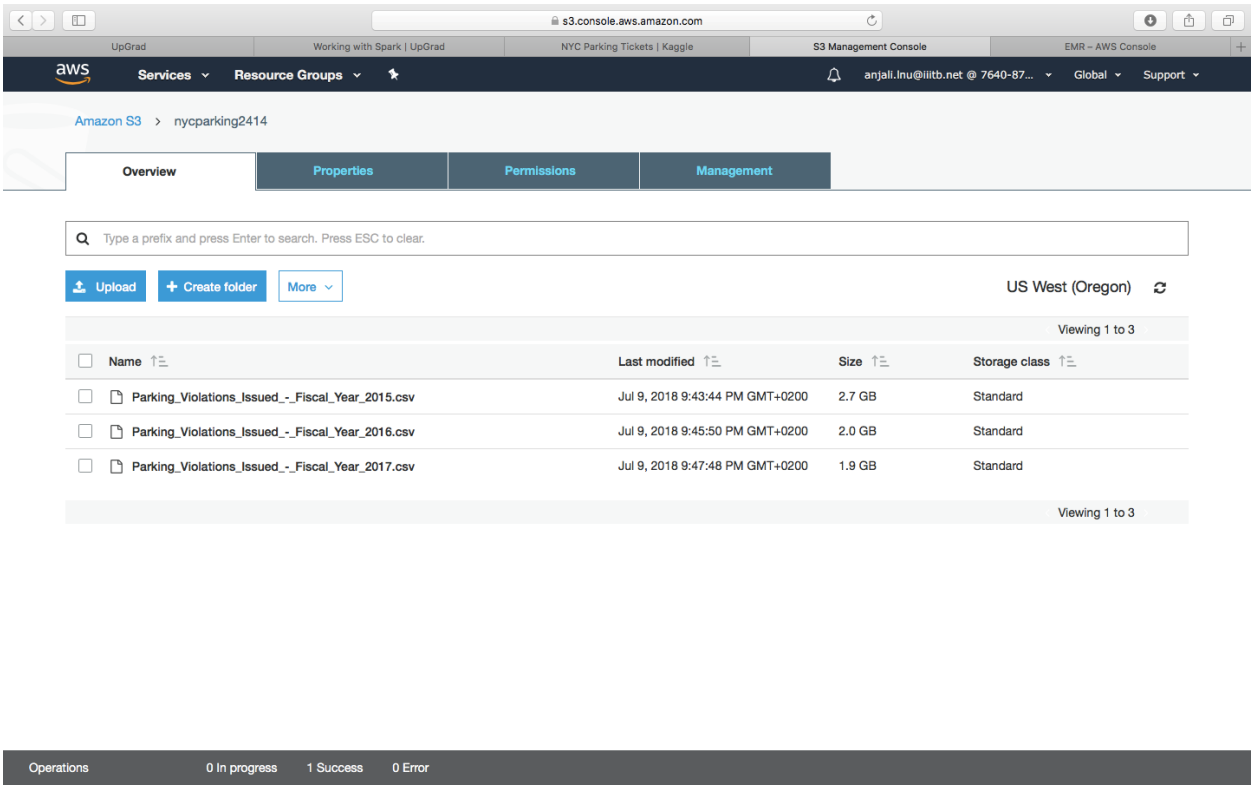
NYC Parking Ticket Case Study

Screen Shot: Prateek Pathak

Note: The file names have been changed for ease of use below.



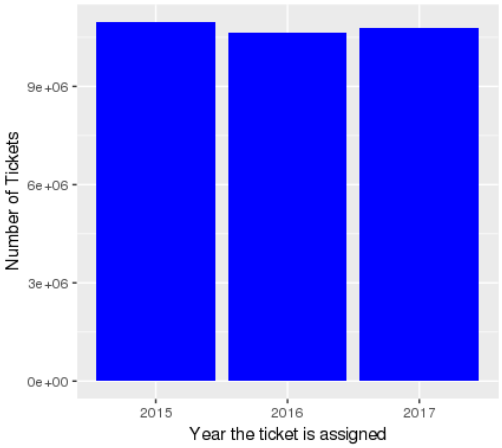
Screen Shot: Anjali Sinha



Examine the Data

1. Comparison of tickets across 3 years

   The total tickets created each year hasn't changed much across 3 fiscal years.



2015: 10951256 Tickets

2016: 10626899 Tickets

2017: 10803028 Tickets

2. The states to which the cars given a tickets have remained nearly constant across 3 years.

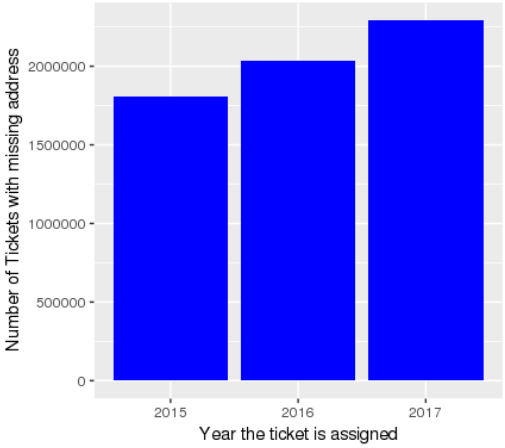
2015: 69 states

2016: 68 states

2017: 67 states

3. Comparison of tickets with address missing

   The tickets with address missing is constantly increasing.



2015: 1807864 Tickets

2016: 2035232 Tickets

2017: 2289944 Tickets

Aggregation tasks

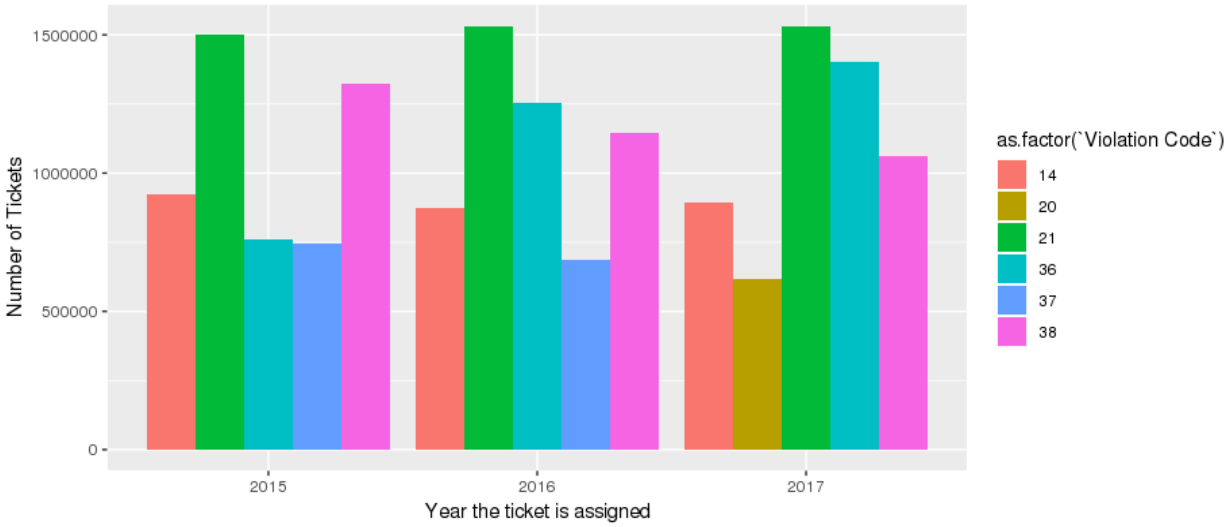
1. How often does each violation code occur? (Frequency of violation codes - find the top 5)

Each year the Violation Codes 21, 14, 36 and 38 are common and have a similar frequencies.

#	Violation Code	cnt_violations
#	21	1501614
#	38	1324586
#	14	924627
#	36	761571
#	37	746278

#	Violation Code	cnt_violations
#	21	1531587
#	36	1253512
#	38	1143696
#	14	875614
#	37	686610

#	Violation Code	cnt_violations
#	21	1528588
#	36	1400614
#	38	1062304
#	14	893498
#	20	618593



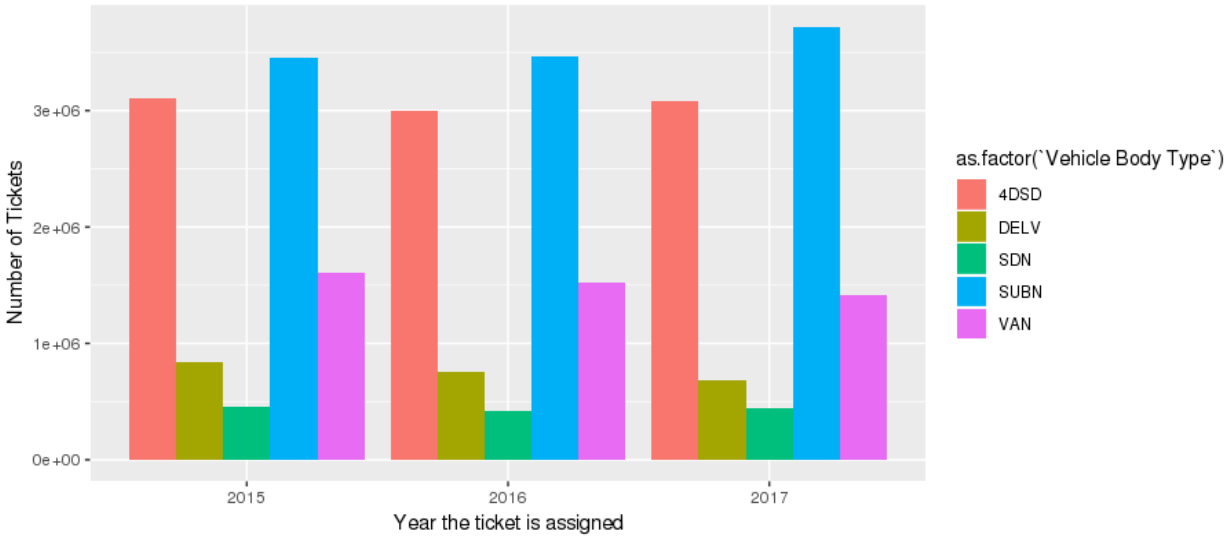
2. How often does each vehicle body type get a parking ticket? (Find the top 5)

Almost each year Body Type SUBN gets the highest number of tickets. Every year the same body types get the highest number of tickets.

#	Year	Vehicle Body Type	cnt_tickets
# 1	2015	SUBN	3451963
# 2	2015	4DSD	3102510
# 3	2015	VAN	1605228
# 4	2015	DELV	840441
# 5	2015	SDN	453992

#	Year	Vehicle Body Type	cnt_tickets
# 1	2016	SUBN	3466037
# 2	2016	4DSD	2992107
# 3	2016	VAN	1518303
# 4	2016	DELV	755282
# 5	2016	SDN	424043

#	Year	Vehicle Body Type	cnt_tickets
# 1	2017	SUBN	3719802
# 2	2017	4DSD	3082020
# 3	2017	VAN	1411970
# 4	2017	DELV	687330
# 5	2017	SDN	438191



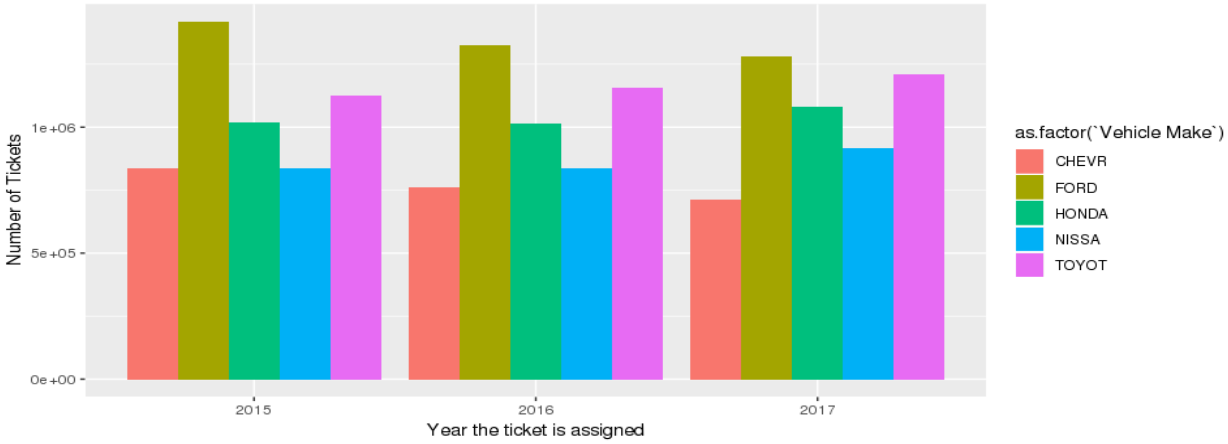
2.2 How about the vehicle make? (Find the top 5)

Almost each year Vehicle make FORD gets the highest number of tickets. Every year the same vehicle makes get the highest number of tickets.

#	Year	Vehicle Make	cnt_tickets
# 1	2015	FORD	1417303
# 2	2015	TOYOT	1123523
# 3	2015	HONDA	1018049
# 4	2015	NISSA	837569
# 5	2015	CHEVR	836389

#	Year	Vehicle Make	cnt_tickets
# 1	2016	FORD	1324774
# 2	2016	TOYOT	1154790
# 3	2016	HONDA	1014074
# 4	2016	NISSA	834833
# 5	2016	CHEVR	759663

#	Year	Vehicle Make	cnt_tickets
# 1	2017	FORD	1280958
# 2	2017	TOYOT	1211451
# 3	2017	HONDA	1079238
# 4	2017	NISSA	918590
# 5	2017	CHEVR	714655



3. A precinct is a police station that has a certain zone of the city under its command. Find the (5 highest) frequencies of:

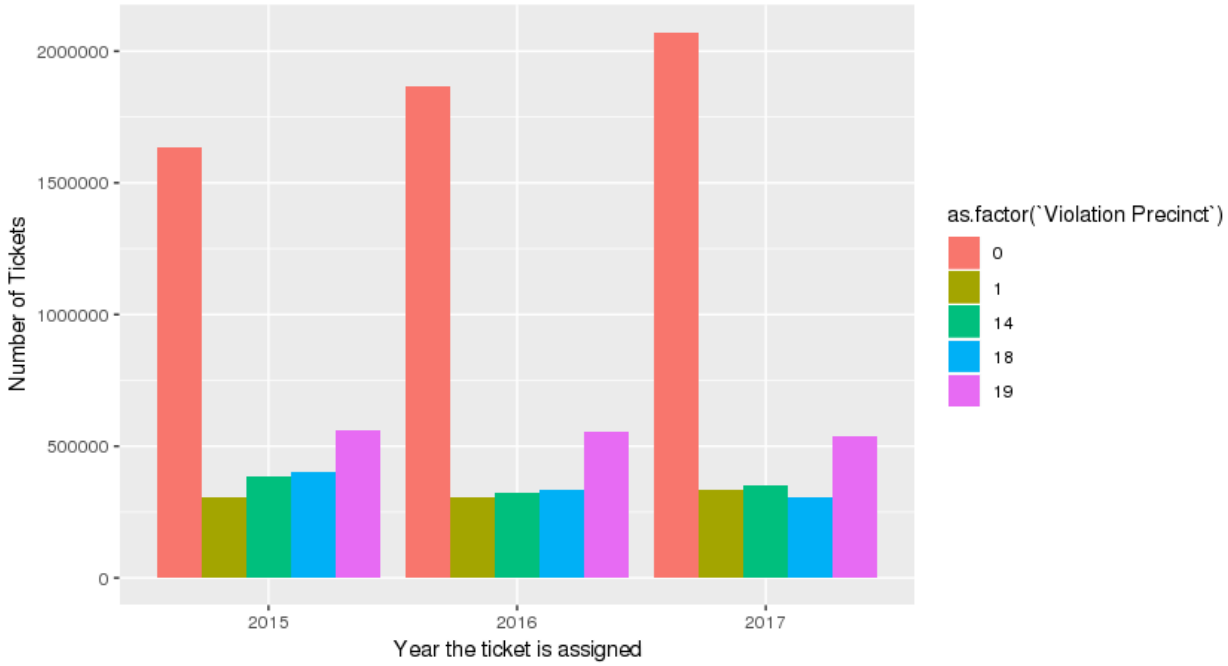
3.1. Violating Precincts (this is the precinct of the zone where the violation occurred)

Clearly each year Violation Precinct 0 has an exceptional number of tickets which keeps increasing year on year.

#	Year	Violation Precinct	cnt_tickets
# 1	2015	0	1633006
# 2	2015	19	559716
# 3	2015	18	400887
# 4	2015	14	384596
# 5	2015	1	307808

#	Year	Violation Precinct	cnt_tickets
# 1	2016	0	1868655
# 2	2016	19	554465
# 3	2016	18	331704
# 4	2016	14	324467
# 5	2016	1	303850

#	Year	Violation Precinct	cnt_tickets
# 1	2017	0	2072400
# 2	2017	19	535671
# 3	2017	14	352450
# 4	2017	1	331810
# 5	2017	18	306920



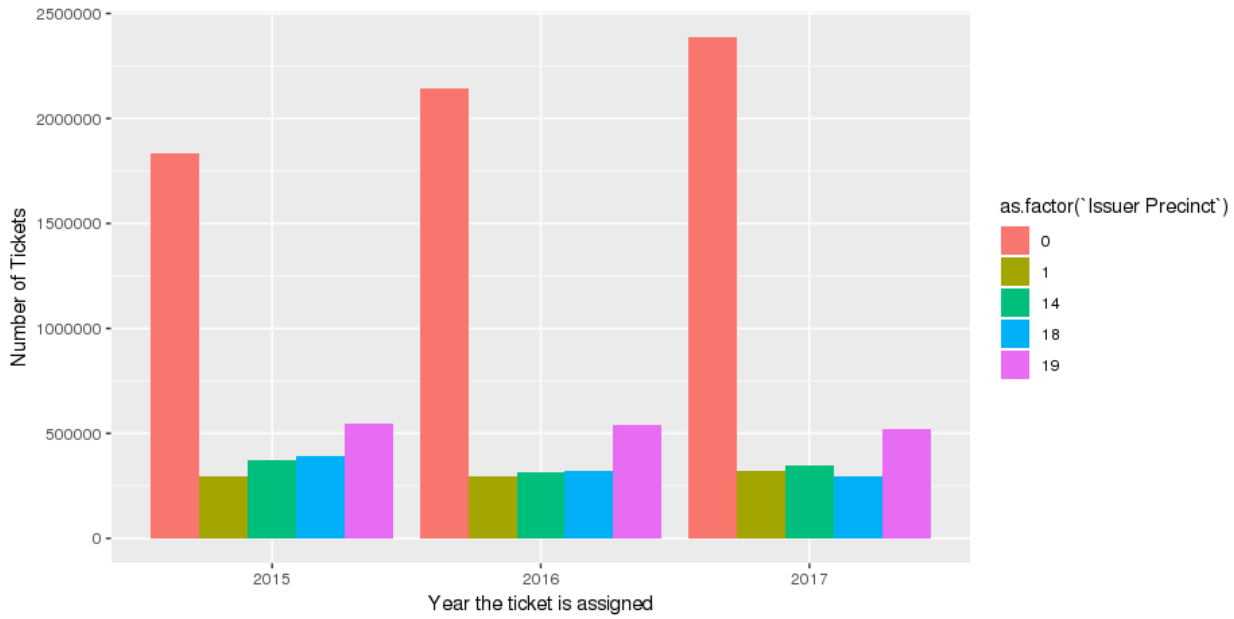
3.2. Issuing Precincts (this is the precinct that issued the ticket)

Clearly each year Issuer Precinct 0 has an exceptional number of tickets which keeps increasing year on year.

#	Year	Issuer Precinct	cnt_tickets
# 1	2015	0	1834343
# 2	2015	19	544946
# 3	2015	18	391501
# 4	2015	14	369725
# 5	2015	1	298594

#	Year	Issuer Precinct	cnt_tickets
# 1	2016	0	2140274
# 2	2016	19	540569
# 3	2016	18	323132
# 4	2016	14	315311
# 5	2016	1	295013

#	Year	Issuer Precinct	cnt_tickets
# 1	2017	0	2388479
# 2	2017	19	521513
# 3	2017	14	344977
# 4	2017	1	321170
# 5	2017	18	296553



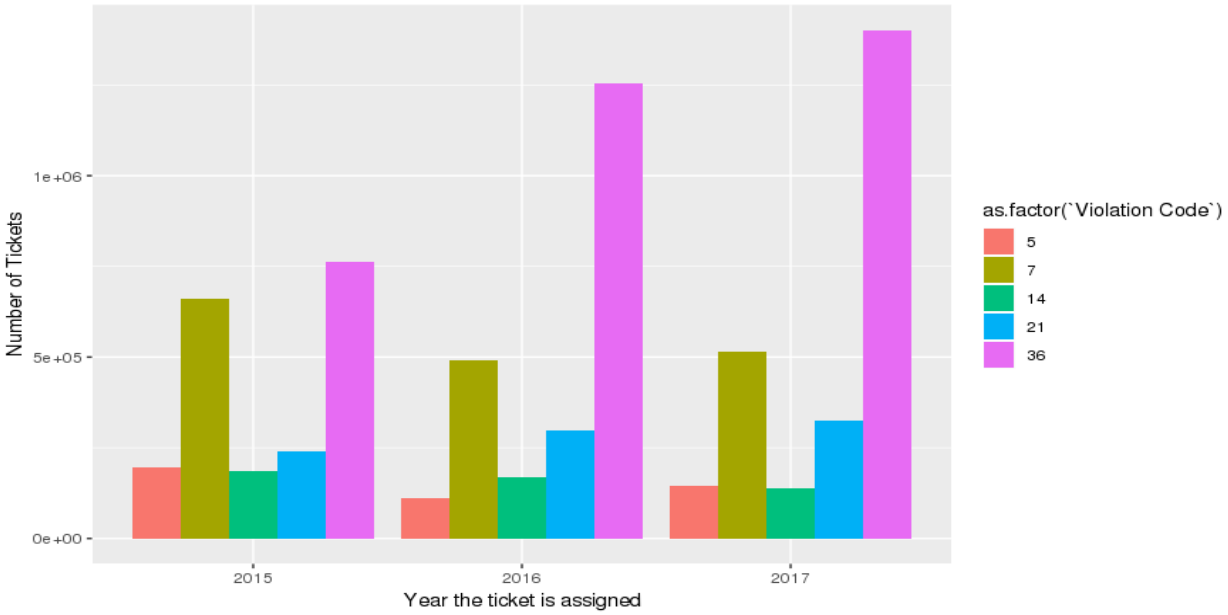
4. Find the violation code frequency across 3 precincts which have issued the most number of tickets - do these precinct zones have an exceptionally high frequency of certain violation codes? Are these codes common across precincts?

The same 5 Violation codes are repeated each year. With Violation Code 36 having an exceptional number of tickets assigned.

#	Year	Violation Code	cnt_tickets
# 1	2015	36	761571
# 2	2015	7	662203
# 3	2015	21	240604
# 4	2015	5	195353
# 5	2015	14	185733

#	Year	Violation Code	cnt_tickets
# 1	2016	36	1253511
# 2	2016	7	492469
# 3	2016	21	299409
# 4	2016	14	167587
# 5	2016	5	112376

#	Year	Violation Code	cnt_tickets
# 1	2017	36	1400614
# 2	2017	7	516390
# 3	2017	21	325435
# 4	2017	5	145643
# 5	2017	14	138488



5 You'd want to find out the properties of parking violations across different times of the day:
The Violation Time field is specified in a strange format. Find a way to make this into a time attribute that you can use to divide into groups. Find a way to deal with missing values, if any. Divide 24 hours into 6 equal discrete bins of time. The intervals you choose are at your discretion.

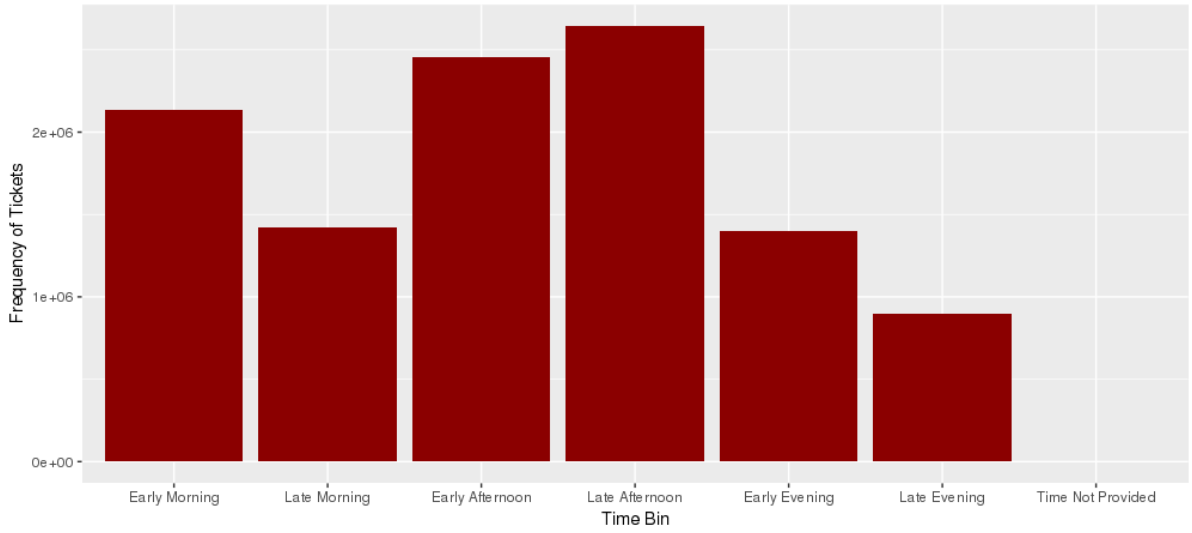
5.1 For each of these groups, find the 3 most commonly occurring violations
The 3 most common times of day where most tickets are given are during **Late Afternoon, Early Afternoon and Early morning** across all 3 years
Logic Used to divide the day

Time Bin	**Violation Time From (Inclusive)**	**Violation Time To (Excluding)**
Early Morning	12:00 AM	4:00 AM
Late Morning	4:00 AM	8:00 AM
Early Afternoon	8:00 AM	12:00 PM
Late Afternoon	12:00 PM	4:00 PM
Early Evening	4:00 PM	8:00 PM
Late Evening	8:00 PM	12:00 AM
Time Not Provided	NA	NA

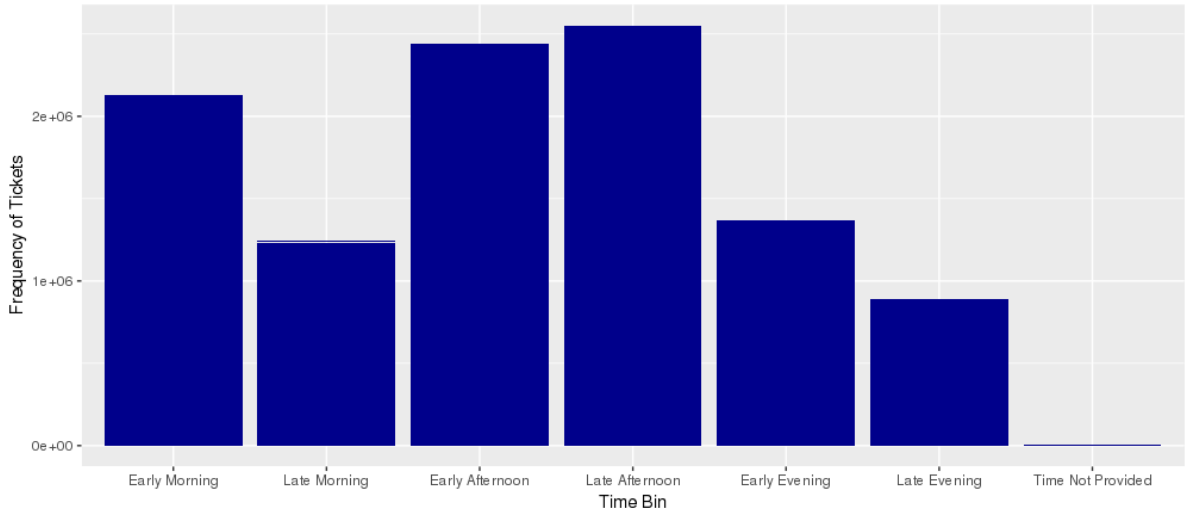
```
# Year 2015
head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Early Morning',], n = 3)
#       time_bin Violation Code cnt_tickets
# 1 Early Morning             21      734165
# 2 Early Morning             38      205820
# 3 Early Morning             14      168314

head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Late Morning',], n = 3)
#        time_bin Violation Code cnt_tickets
# 1 Late Morning             38      241327
# 2 Late Morning             37      175802
# 3 Late Morning              7      168888

head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Early Afternoon',], n = 3)
#          time_bin Violation Code cnt_tickets
# 1 Early Afternoon             21      525430
# 2 Early Afternoon             38      243897
# 3 Early Afternoon             36      196896

head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Late Afternoon',], n = 3)
#          time_bin Violation Code cnt_tickets
# 1 Late Afternoon             38      432218
# 2 Late Afternoon             37      324892
# 3 Late Afternoon             36      220661

head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Early Evening',], n = 3)
#         time_bin Violation Code cnt_tickets
# 1 Early Evening             38      198472
# 2 Early Evening             21      130163
# 3 Early Evening              7      124456

head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Late Evening',], n = 3)
#        time_bin Violation Code cnt_tickets
# 1 Late Evening             14      134458
# 2 Late Evening             21      106858
# 3 Late Evening             40       91344


# Year 2016
head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Early Morning',], n = 3)
#       time_bin Violation Code cnt_tickets
# 1 Early Morning             21      754150
# 2 Early Morning             36      262974
# 3 Early Morning             38      173463

head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Late Morning',], n = 3)
#         time_bin Violation Code cnt_tickets
# 1 Late Morning             38      211267
# 2 Late Morning             37      161655
# 3 Late Morning             14      134976

head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Early Afternoon',], n =
#         time_bin Violation Code cnt_tickets
# 1 Early Afternoon             21      527202
# 2 Early Afternoon             36      323818
# 3 Early Afternoon             38      215021

head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Late Afternoon',], n = 3
#          time_bin Violation Code cnt_tickets
# 1 Late Afternoon             36      378435
# 2 Late Afternoon             38      367579
# 3 Late Afternoon             37      297619

head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Early Evening',], n = 3)
#         time_bin Violation Code cnt_tickets
# 1 Early Evening             38      173897
# 2 Early Evening             36      167282
# 3 Early Evening             21      131120

head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Late Evening',], n = 3)
#         time_bin Violation Code cnt_tickets
# 1 Late Evening             14      140111
# 2 Late Evening             21      114029
# 3 Late Evening             40       91692


# Year 2017
head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Early Morning',], n = 3)
#       time_bin Violation Code cnt_tickets
# 1 Early Morning             21      746351
# 2 Early Morning             36      335271
# 3 Early Morning             38      154809

head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Late Morning',], n = 3)
#       time_bin Violation Code cnt_tickets
# 1 Late Morning             38      203232
# 2 Late Morning             37      145784
# 3 Late Morning             14      144749

head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Early Afternoon',], n = 3
#         time_bin Violation Code cnt_tickets
# 1 Early Afternoon             21      513799
# 2 Early Afternoon             36      416151
# 3 Early Afternoon             38      192173

head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Late Afternoon',], n = 3)
#          time_bin Violation Code cnt_tickets
# 1 Late Afternoon             36      376961
# 2 Late Afternoon             38      356253
# 3 Late Afternoon             37      265848

head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Early Evening',], n = 3)
#         time_bin Violation Code cnt_tickets
# 1 Early Evening             36      211434
# 2 Early Evening             38      153537
# 3 Early Evening             21      144082

head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Late Evening',], n = 3)
#        time_bin Violation Code cnt_tickets
# 1 Late Evening             14      141276
# 2 Late Evening             21      119469
# 3 Late Evening             40      112186
```

Plotting the total tickets given during different time Bins data across 3 years.

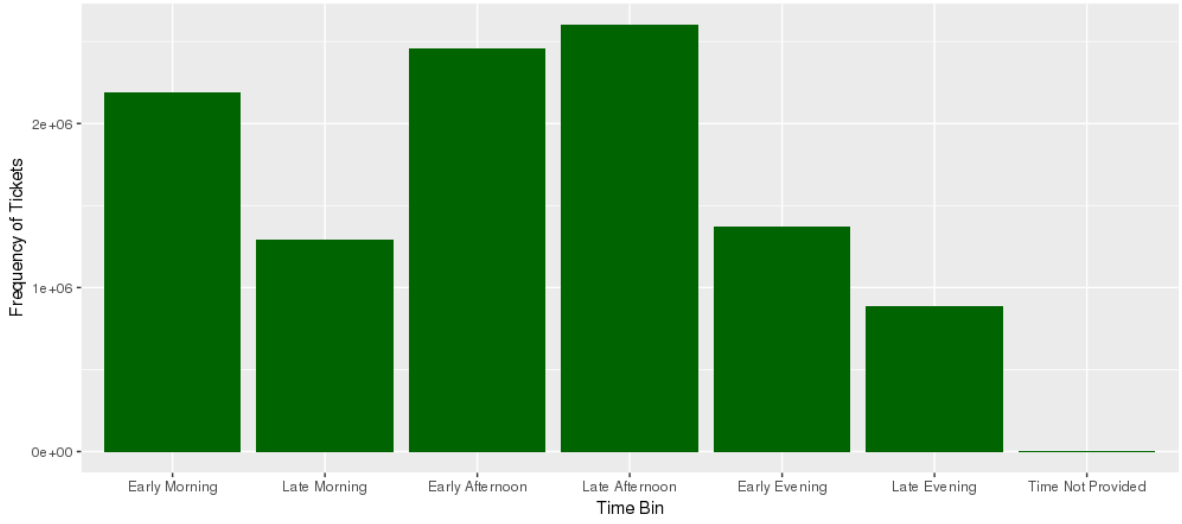
Year 2015



Year 2016



Year 2017



5.2 For the 3 most commonly occurring violation codes, find the most common times of day (in terms of the bins from the previous part)

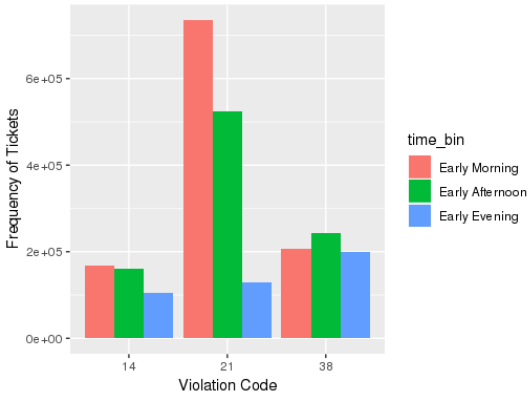
Clearly violation code 21 across all years has been most given in early mornings

```
# Year 2015
df_code_1_2015
#          time_bin Violation Code cnt_tickets
# 1 Early Afternoon             21      734165
# 2   Early Morning             21      525430
# 3    Late Evening             21      130163

df_code_2_2015
#          time_bin Violation Code cnt_tickets
# 1 Late Afternoon             38      432218
# 2  Early Morning             38      243897
# 3  Early Evening             38      241327

df_code_3_2015
#          time_bin Violation Code cnt_tickets
# 1  Late Afternoon             14      207927
# 2 Early Afternoon             14      168314
# 3   Early Morning             14      159848
```

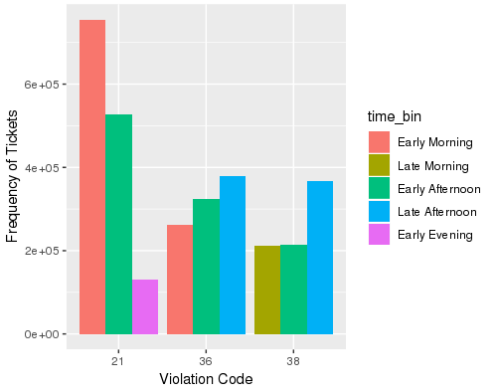
```
# Year 2016
df_code_1_2016
#          time_bin Violation Code cnt_tickets
# 1   Early Morning             21      754150
# 2 Early Afternoon             21      527202
# 3   Early Evening             21      131120

df_code_2_2016
#          time_bin Violation Code cnt_tickets
# 1  Late Afternoon             36      378435
# 2 Early Afternoon             36      323818
# 3   Early Morning             36      262974

df_code_3_2016
#          time_bin Violation Code cnt_tickets
# 1  Late Afternoon             38      367579
# 2 Early Afternoon             38      215021
# 3    Late Morning             38      211267
```

```
# Year 2017
df_code_1_2017
#          time_bin Violation Code cnt_tickets
# 1   Early Morning             21      746351
# 2 Early Afternoon             21      513799
# 3   Early Evening             21      144082

df_code_2_2017
#          time_bin Violation Code cnt_tickets
# 1 Early Afternoon             36      416151
# 2  Late Afternoon             36      376961
# 3   Early Morning             36      335271

df_code_3_2017
#          time_bin Violation Code cnt_tickets
# 1  Late Afternoon             38      356253
# 2    Late Morning             38      203232
# 3 Early Afternoon             38      192173
```

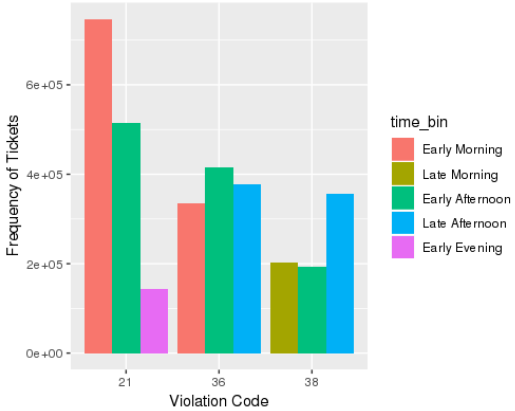
Year 2015



Year 2016



Year 2017



6. Let's try and find some seasonality in this data. First, divide the year into some number of seasons
   Logic Used to divide into Seasons

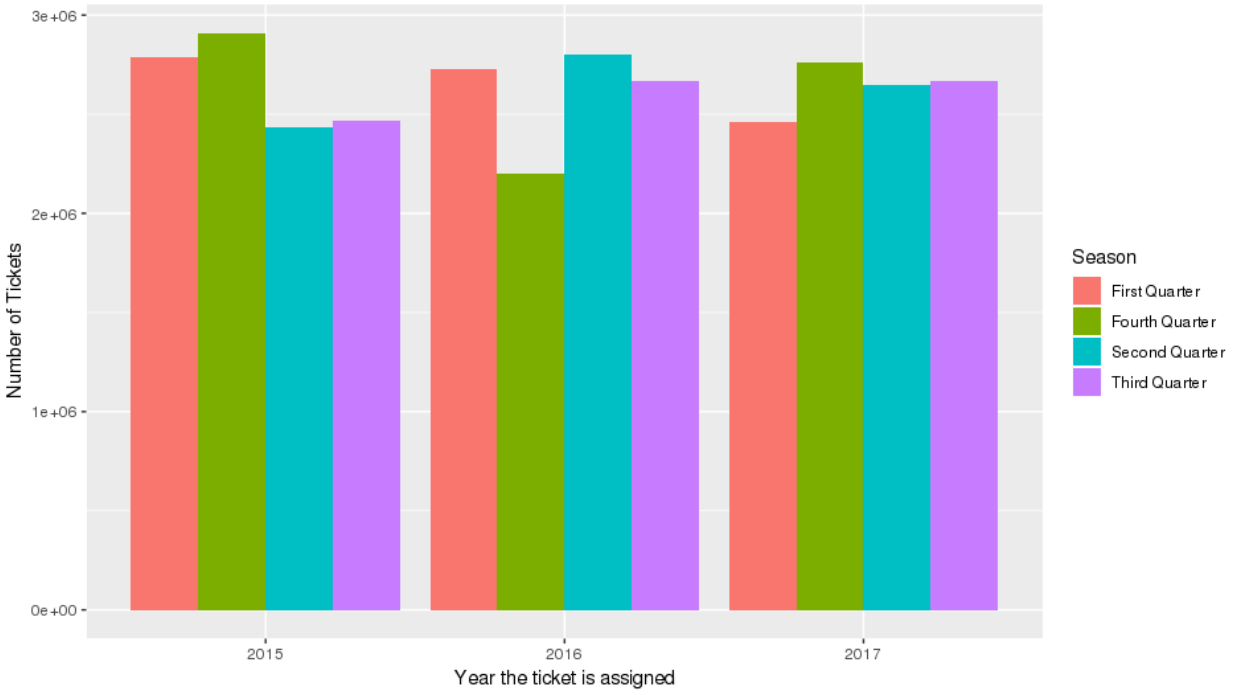
Time Bin	Issue Date From (Inclusive)	Issue Date To (Excluding)
First Quarter	1st July Previous Year	30th September Previous Year
Second Quarter	1st October Previous Year	31st December Previous Year
Third Quarter	1st January current year	31st March Current year
Fourth Quarter	1st April current year	30th June current year
Season Not Defined	NA	NA

6.1 Find frequencies of tickets for each season.
   The frequency of tickets each season is shifting from high number of tickets from First quarter to Fourth quarter from 2015 to 2017

```
# 2015
# Season          count_tickets
# Fourth Quarter       2907331
# First Quarter        2788963
# Third Quarter        2466640
# Second Quarter       2435101
```

```
# 2016
# Season          count_tickets
# Second Quarter       2799402
# First Quarter        2726774
# Third Quarter        2668423
# Fourth Quarter       2202295
```

```
# 2017
# Season          count_tickets
# Fourth Quarter       2760833
# Third Quarter        2669069
# Second Quarter       2647391
# First Quarter        2462270
```



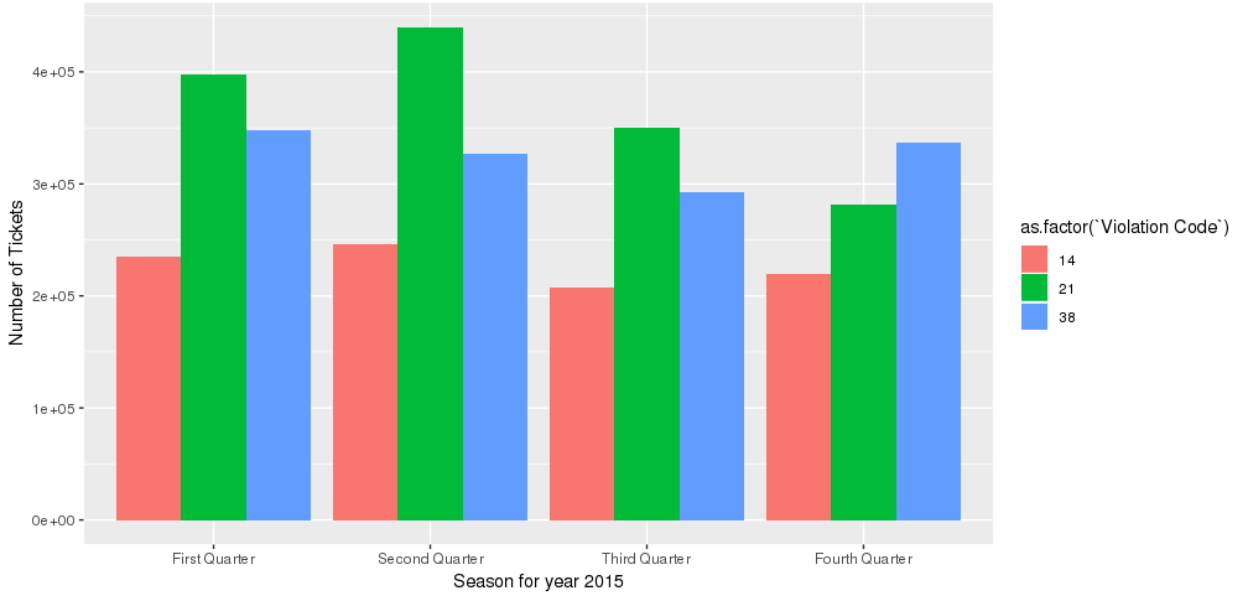
6.2 Find the 3 most common violations for each of the season

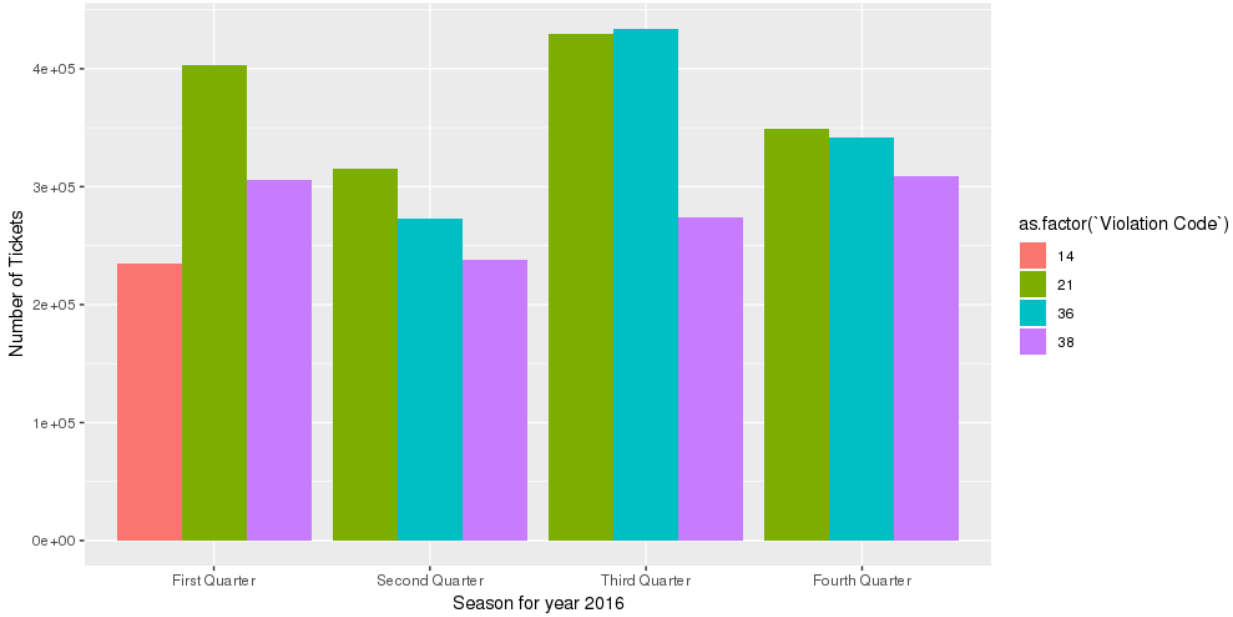
Violation Codes 21 and 38 are common across all seasons and every year

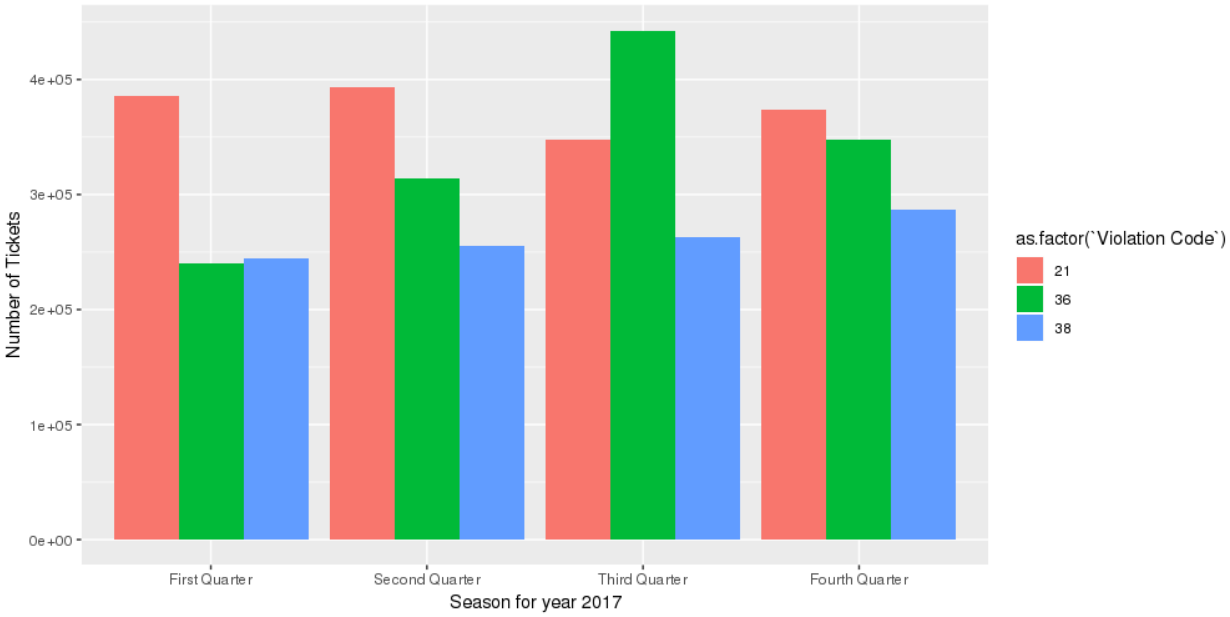
```
# Year 2015
# First Quarter
df_violation_first_quarter_2015
#         Season Violation Code count_tickets
# 1 First Quarter             21         397809
# 2 First Quarter             38         348466
# 3 First Quarter             14         234565

# Second Quarter
df_violation_second_quarter_2015
# 1 Second Quarter             21         350517
# 2 Second Quarter             38         292637
# 3 Second Quarter             14         207365

# Third Quarter
df_violation_third_quarter_2015
#         Season Violation Code count_tickets
# 1 Third Quarter             38         336746
# 2 Third Quarter             21         281386
# 3 Third Quarter             14         219828

# Fourth Quarter
df_violation_fourth_quarter_2015
# 1 Fourth Quarter             21         439516
# 2 Fourth Quarter             38         327158
# 3 Fourth Quarter             14         246660
```

```
# Year 2016
# First Quarter
df_violation_first_quarter_2016
#         Season Violation Code count_tickets
# 1 First Quarter             21         403309
# 2 First Quarter             38         305341
# 3 First Quarter             14         234798

# Second Quarter
df_violation_second_quarter_2016
#          Season Violation Code count_tickets
# 1 Second Quarter             36         433966
# 2 Second Quarter             21         429429
# 3 Second Quarter             38         274424

# Third Quarter
df_violation_third_quarter_2016
#         Season Violation Code count_tickets
# 1 Third Quarter             21         349297
# 2 Third Quarter             36         341787
# 3 Third Quarter             38         308987

# Fourth Quarter
df_violation_fourth_quarter_2016
#         Season Violation Code count_tickets
# 1 Fourth Quarter             21         315234
# 2 Fourth Quarter             36         273455
# 3 Fourth Quarter             38         238083
```

```
# Year 2017
# First Quarter
df_violation_first_quarter_2017
#         Season Violation Code count_tickets
# 1 First Quarter             21         385410
# 2 First Quarter             38         244972
# 3 First Quarter             36         239879

# Second Quarter
df_violation_second_quarter_2017
#          Season Violation Code count_tickets
# 1 Second Quarter             36         442593
# 2 Second Quarter             21         347227
# 3 Second Quarter             38         263382

# Third Quarter
df_violation_third_quarter_2017
#         Season Violation Code count_tickets
# 1 Third Quarter             21         373874
# 2 Third Quarter             36         348240
# 3 Third Quarter             38         287000

# Fourth Quarter
df_violation_fourth_quarter_2017
#         Season Violation Code count_tickets
# 1 Fourth Quarter             21         393885
# 2 Fourth Quarter             36         314525
# 3 Fourth Quarter             38         255064
```







7 The fines collected from all the parking violation constitute a revenue source for the NYC police department. Let's take an example of estimating that for the 3 most commonly occurring codes.

7.1 Find total occurrences of the 3 most common violation codes.

Across all years Violation code has the most tickets assigned.

```
# Year 2015
#   Violation code Cnt_Tickets
#     21           1501614
#     38           1324586
#     14            924627
```

```
# Year 2016
# Violation code Cnt_Tickets
#  21             1531587
#  36             1253512
#  38             1143696
```

```
# Year 2017
# Violation code Cnt_Tickets
#  21             1528588
#  36             1400614
#  38             1062304
```

7.2 Then, search the Internet for NYC parking violation code fines. You will find a website (on the nyc.gov URL) that lists these fines. They're divided into two categories, one for the highest-density locations of the city, the other for the rest of the city. For simplicity, take an average of the two. Using this information, find the total amount collected for all of the fines. State the code which has the highest total collection.

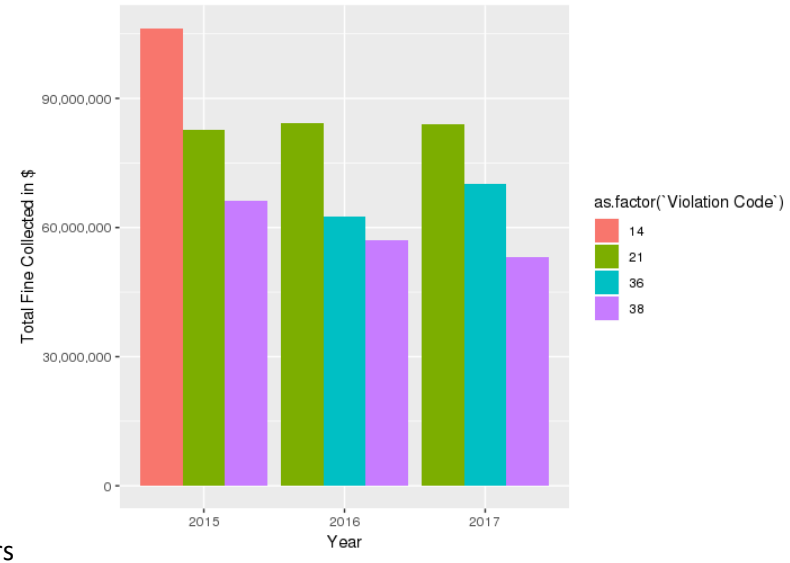
```
# From the NYC website we get the below info on Violation Codes fines
# 21    Street Cleaning: No parking where parking is not allowed by sign, street marking or traffic control device.
# 14    General No Standing: Standing or parking where standing is not allowed by sign, street marking or; traffic control device.
# 38    Failing to show a receipt or tag in the windshield. Drivers get a 5-minute grace period past the expired time on Muni-Meter receipts.
# 36    Exceeding the posted speed limit in or near a designated school zone.

# Violation Code   Average Fine
#       14        $115
#       21        $55
#       36        $50
#       38        $50
```

Total Fine collected

```
#   Year Violation Code Total_Fine
# 1 2015             21  82588770
# 2 2015             38  66229300
# 3 2015             14 106332105
```

```
#   Year Violation Code Total_Fine
# 1 2016             21 84237285
# 2 2016             36 62675600
# 3 2016             38 57184800
```

```
#   Year Violation Code Total_Fine
# 1 2017             21 84072340
# 2 2017             36 70030700
# 3 2017             38 53115200
```



Plotting the above across all years

7.3 What can you intuitively infer from these findings?

Violation Code 14 gave the highest fine collected in 2015 but in later years Violation Code 21 is the major source of total fine collection for both 2016 and 2017. Every year most of the tickets are given to people parking there vehicle in No Parking zone (Code 21).

Solution Code

```
##############################################################################################################

############### Load data to Spark DataFrames

##############################################################################################################

# load data to SparkR

library(SparkR)

library(ggplot2)


# initiating the spark session

sparkR.session(master='local')


# Load all 3 years files CSV files from S3 bucket

nycTickets2015 <- SparkR::read.df("s3://prateek-nyc-parking/2015", "csv", header="true", inferSchema = "true")

nycTickets2016 <- SparkR::read.df("s3://prateek-nyc-parking/2016", "csv", header="true", inferSchema = "true")

nycTickets2017 <- SparkR::read.df("s3://prateek-nyc-parking/2017", "csv", header="true", inferSchema = "true")


##############################################################################################################

############### General Analysis

##############################################################################################################

## Check general statistics for 2015 parking tickets

head(nycTickets2015)


ncol(nycTickets2015)

# 51 Columns

nrow(nycTickets2015)

# 11809233 Rows


## Check general statistics for 2016 parking tickets

head(nycTickets2016)


ncol(nycTickets2016)

# 51 Columns

nrow(nycTickets2016)

# 10626899 Rows


## Check general statistics for 2017 parking tickets
```

```
head(nycTickets2017)


ncol(nycTickets2017)
# 43 Columns
nrow(nycTickets2017)
# 10803028 Rows

## Remove absolute duplicate rows
nycTickets2015 <- dropDuplicates(nycTickets2015)
nycTickets2016 <- dropDuplicates(nycTickets2016)
nycTickets2017 <- dropDuplicates(nycTickets2017)

## Create Temp Views for all years
createOrReplaceTempView(nycTickets2015, "nycTickets2015_tbl")
createOrReplaceTempView(nycTickets2016, "nycTickets2016_tbl")
createOrReplaceTempView(nycTickets2017, "nycTickets2017_tbl")

# Verify if the data is from correct fiscal year for each year 1st July previous year to 30th June current year
# 2015
cnt_wrong_year_2015 <- SparkR::sql("SELECT count(*) cnt_tickets
                        FROM nycTickets2015_tbl \
                        where TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                            TO_DATE(CAST(UNIX_TIMESTAMP('07/01/2014', 'MM/dd/yyyy') AS TIMESTAMP))
                         AND TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >
                            TO_DATE(CAST(UNIX_TIMESTAMP('06/01/2015', 'MM/dd/yyyy') AS TIMESTAMP))")
head(cnt_wrong_year_2015)
# 0

# 2016
cnt_wrong_year_2016 <- SparkR::sql("SELECT count(*) cnt_tickets
                        FROM nycTickets2016_tbl \
                        where TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                            TO_DATE(CAST(UNIX_TIMESTAMP('07/01/2015', 'MM/dd/yyyy') AS TIMESTAMP))
                         AND TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >
                            TO_DATE(CAST(UNIX_TIMESTAMP('06/01/2016', 'MM/dd/yyyy') AS TIMESTAMP))")
head(cnt_wrong_year_2016)
```

```
# 0


# 2017
cnt_wrong_year_2017 <- SparkR::sql("SELECT count(*) cnt_tickets
                        FROM nycTickets2017_tbl \
                        where TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                            TO_DATE(CAST(UNIX_TIMESTAMP('07/01/2016', 'MM/dd/yyyy') AS TIMESTAMP))
                         AND TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >
                            TO_DATE(CAST(UNIX_TIMESTAMP('06/01/2017', 'MM/dd/yyyy') AS TIMESTAMP))")
head(cnt_wrong_year_2017)
# 0



# Assumptions: There are still some duplicate summons numbers left in all 3 data sets. But we will not be removing them
# as we will be using distinct in all counts.

#############################################################################################
############## Examine the data
#############################################################################################


###############
##          1: Find total number of tickets for each year.
###############


############## 2015 ##############
total_tickets_2015 <- SparkR::sql("SELECT '2015' as Year,count(distinct `Summons Number`)  as cnt_tickets FROM
nycTickets2015_tbl")
df_total_tickets_2015 <- collect(total_tickets_2015)
head(df_total_tickets_2015)
# 10951256 Tickets


############## 2016 ##############
total_tickets_2016 <- SparkR::sql("SELECT '2016' as Year, count(distinct `Summons Number`)  as cnt_tickets FROM
nycTickets2016_tbl")
df_total_tickets_2016 <- collect(total_tickets_2016)
head(df_total_tickets_2016)
```

```
# 10626899 Tickets


############## 2017 ##############

total_tickets_2017 <- SparkR::sql("SELECT '2017' as Year, count(distinct `Summons Number`)  as cnt_tickets FROM
nycTickets2017_tbl")

df_total_tickets_2017 <- collect(total_tickets_2017)

head(df_total_tickets_2017)

# 10803028 Tickets


# Plot across all years

df_total_tickets <- rbind(df_total_tickets_2015,df_total_tickets_2016)

df_total_tickets <- rbind(df_total_tickets,df_total_tickets_2017)


plot_total_tickets <- ggplot(df_total_tickets, aes(Year , cnt_tickets)) +

 geom_bar(stat = "identity", fill = "blue") +

 xlab("Year the ticket is assigned") +

 ylab("Number of Tickets")


plot_total_tickets


# Conclusion : The count of tickets has not changed much over the 3 years.

#                                  The count came down slightly from 2015 to 2016 and increased slightly in 2017.


###############

##          2: Find out how many unique states the cars which got parking tickets came from.

###############


############## 2015 ##############

unique_states_2015 <- SparkR::sql("SELECT count(distinct `Registration State`) as cnt_unique_states

                          FROM nycTickets2015_tbl

                          WHERE `Registration State` IS NOT NULL")

head(unique_states_2015)

# 69 states for 2015


############## 2016 ##############

unique_states_2016 <-  SparkR::sql("SELECT count(distinct `Registration State`) as cnt_unique_states
```

```
                        FROM nycTickets2016_tbl
                       WHERE `Registration State` IS NOT NULL")
head(unique_states_2016)
# 68 states for 2016

############## 2017 ##############
unique_states_2017 <-  SparkR::sql("SELECT count(distinct `Registration State`) as cnt_unique_states
                        FROM nycTickets2017_tbl
                       WHERE `Registration State` IS NOT NULL")
head(unique_states_2017)
# 67 states for 2017

# Conclusion : The count of states has not changed much over the 3 years.

##############
##          3: Some parking tickets don't have addresses on them, which is cause for concern. Find out how many such tickets there are.
##############

# For analysis we take address as House Number and Street Name
# If either of House Number or Street Name or both are missing we call it as ticket with missing address.

############## 2015 ##############
cnt_missing_address_2015 <- SparkR::sql("SELECT '2015' as Year,count(distinct `Summons Number`) as cnt_tickets
                        FROM nycTickets2015_tbl
                        WHERE `House Number` IS NULL
                        OR `Street Name` IS NULL")
df_cnt_missing_address_2015 <- collect(cnt_missing_address_2015)
head(df_cnt_missing_address_2015)
# 1807864 Tickets have address missing

############## 2016 ##############
cnt_missing_address_2016 <- SparkR::sql("SELECT '2016' as Year,count(distinct `Summons Number`) as cnt_tickets
                        FROM nycTickets2016_tbl
                        WHERE `House Number` IS NULL
                        OR `Street Name` IS NULL")
```

```
df_cnt_missing_address_2016 <- collect(cnt_missing_address_2016)
head(df_cnt_missing_address_2016)
# 2035232 Tickets have address missing

############### 2017 ###############
cnt_missing_address_2017 <- SparkR::sql("SELECT '2017' as Year,count(distinct `Summons Number`) as cnt_tickets
                          FROM nycTickets2017_tbl
                          WHERE `House Number` IS NULL
                          OR `Street Name` IS NULL")
df_cnt_missing_address_2017 <- collect(cnt_missing_address_2017)
head(df_cnt_missing_address_2017)
# 2289944 Tickets have address missing

# Plot across all years
df_cnt_missing_address <- rbind(df_cnt_missing_address_2015,df_cnt_missing_address_2016)
df_cnt_missing_address <- rbind(df_cnt_missing_address,df_cnt_missing_address_2017)

plot_cnt_missing_address <- ggplot(df_cnt_missing_address, aes(Year , cnt_tickets)) +
  geom_bar(stat = "identity", fill = "blue") +
  xlab("Year the ticket is assigned") +
  ylab("Number of Tickets with missing address")
plot_cnt_missing_address

# Conclusion: Count of tickets with missing address is increasing each year by 200000.

#############################################################################################
############### Aggregation tasks
#############################################################################################

###############
## 1: How often does each violation code occur? (frequency of violation codes - find the top 5)
###############

############### Year 2015 ###############
freq_violation_2015 <- SparkR::sql("SELECT '2015' as Year, `Violation Code`,
                            count(distinct `Summons Number`) as cnt_violations
```

```
                        FROM nycTickets2015_tbl
                        WHERE `Violation Code` IS NOT NULL
                        GROUP BY `Violation Code`
                        ORDER BY count(*) desc")

df_freq_violation_2015 <- head(freq_violation_2015, n = 5)
df_freq_violation_2015

# Violation Code cnt_violations
#      21       1501614
#      38       1324586
#      14        924627
#      36        761571
#      37        746278

############### Year 2016 ###############
freq_violation_2016 <- SparkR::sql("SELECT '2016' as Year, `Violation Code`,
                                count(distinct `Summons Number`) as cnt_violations
                        FROM nycTickets2016_tbl
                        WHERE `Violation Code` IS NOT NULL
                        GROUP BY `Violation Code`
                        ORDER BY count(*) desc")
df_freq_violation_2016 <- head(freq_violation_2016, n = 5)
df_freq_violation_2016

# Violation Code cnt_violations
#     21       1531587
#     36       1253512
#     38       1143696
#     14        875614
#     37        686610

############### Year 2017 ###############
freq_violation_2017 <- SparkR::sql("SELECT '2017' as Year, `Violation Code`,
                                    count(distinct `Summons Number`) as cnt_violations
                                FROM nycTickets2017_tbl
```

```
                                                                WHERE `Violation Code` IS NOT NULL
                        GROUP BY `Violation Code`
                        ORDER BY count(*) desc")
df_freq_violation_2017 <- head(freq_violation_2017, n = 5)
df_freq_violation_2017

# Violation Code cnt_violations
#     21       1528588
#     36       1400614
#     38       1062304
#     14        893498
#     20        618593

# Plot across all years
df_freq_violation_code <- rbind(df_freq_violation_2015,df_freq_violation_2016)
df_freq_violation_code <- rbind(df_freq_violation_code,df_freq_violation_2017)

plot_freq_violation_code <- ggplot(df_freq_violation_code, aes(Year , cnt_violations, fill = as.factor(`Violation Code`))) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Year the ticket is assigned") +
  ylab("Number of Tickets")
plot_freq_violation_code

# Conclusion: Across all 3 years Violation Codes 21,14, 36 and 38 are common and have a similar frequencies.

###############
## 2: How often does each vehicle body type get a parking ticket?
##    How about the vehicle make? (find the top 5 for both)
###############

## Count of tickets for each vehicle body type

############### Year 2015 ###############
freq_body_type_2015 <- SparkR::sql("SELECT '2015' as Year, `Vehicle Body Type`,
                                   count(distinct `Summons Number`) as cnt_tickets \
                        FROM nycTickets2015_tbl \
```

```
                        GROUP BY `Vehicle Body Type`
                        ORDER BY count(*) desc")

df_freq_body_type_2015 <- head(freq_body_type_2015, n = 5)
df_freq_body_type_2015

#   Year Vehicle Body Type cnt_tickets
# 1 2015            SUBN     3451963
# 2 2015            4DSD     3102510
# 3 2015             VAN     1605228
# 4 2015            DELV      840441
# 5 2015             SDN      453992

############### Year 2016 ###############
freq_body_type_2016 <- SparkR::sql("SELECT '2016' as Year, `Vehicle Body Type`,
                                   count(distinct `Summons Number`) as cnt_tickets \
                        FROM nycTickets2016_tbl \
                        GROUP BY `Vehicle Body Type`
                        ORDER BY count(*) desc")

df_freq_body_type_2016 <- head(freq_body_type_2016, n = 5)
df_freq_body_type_2016

#   Year Vehicle Body Type cnt_tickets
# 1 2016            SUBN     3466037
# 2 2016            4DSD     2992107
# 3 2016             VAN     1518303
# 4 2016            DELV      755282
# 5 2016             SDN      424043

############### Year 2017 ###############
freq_body_type_2017 <- SparkR::sql("SELECT '2017' as Year, `Vehicle Body Type`,
                                   count(distinct `Summons Number`) as cnt_tickets \
                        FROM nycTickets2017_tbl \
                        GROUP BY `Vehicle Body Type`
                        ORDER BY count(*) desc")
```

```
df_freq_body_type_2017 <- head(freq_body_type_2017, n = 5)
df_freq_body_type_2017

#   Year Vehicle Body Type cnt_tickets
# 1 2017            SUBN     3719802
# 2 2017            4DSD     3082020
# 3 2017             VAN     1411970
# 4 2017            DELV      687330
# 5 2017             SDN      438191

# Plot across all years
df_freq_body_type <- rbind(df_freq_body_type_2015,df_freq_body_type_2016)
df_freq_body_type <- rbind(df_freq_body_type,df_freq_body_type_2017)

plot_freq_body_type <- ggplot(df_freq_body_type, aes(Year , cnt_tickets, fill = as.factor(`Vehicle Body Type`))) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Year the ticket is assigned") +
  ylab("Number of Tickets")
plot_freq_body_type

# Conclusion: Almost each year Body Type SUBN gets the highest number of tickets. Every year the same body types get the highest number of tickets.

## Count of tickets for each Vehicle Make

############### Year 2015 ###############
freq_vehicle_make_2015 <- SparkR::sql("SELECT '2015' as Year, `Vehicle Make`,
                                      count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2015_tbl \
                                GROUP BY `Vehicle Make`
                                ORDER BY count(*) desc")

df_freq_vehicle_make_2015 <- head(freq_vehicle_make_2015, n = 5)
df_freq_vehicle_make_2015
```

```
#   Year Vehicle Make cnt_tickets
# 1 2015        FORD     1417303
# 2 2015       TOYOT     1123523
# 3 2015       HONDA     1018049
# 4 2015       NISSA      837569
# 5 2015       CHEVR      836389

############### Year 2016 ###############
# Count of tickets for each vehicle body type
freq_vehicle_make_2016 <- SparkR::sql("SELECT '2016' as Year, `Vehicle Make`,
                                      count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2016_tbl \
                                GROUP BY `Vehicle Make`
                                ORDER BY count(*) desc")

df_freq_vehicle_make_2016 <- head(freq_vehicle_make_2016, n = 5)
df_freq_vehicle_make_2016

#  Year Vehicle Make cnt_tickets
# 1 2016        FORD     1324774
# 2 2016       TOYOT     1154790
# 3 2016       HONDA     1014074
# 4 2016       NISSA      834833
# 5 2016       CHEVR      759663

############### Year 2017 ###############
# Count of tickets for each vehicle body type
freq_vehicle_make_2017 <- SparkR::sql("SELECT '2017' as Year, `Vehicle Make`,
                                      count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2017_tbl \
                                GROUP BY `Vehicle Make`
                                ORDER BY count(*) desc")

df_freq_vehicle_make_2017 <- head(freq_vehicle_make_2017, n = 5)
df_freq_vehicle_make_2017
```

```
#  Year Vehicle Make cnt_tickets
# 1 2017        FORD     1280958
# 2 2017       TOYOT     1211451
# 3 2017       HONDA     1079238
# 4 2017       NISSA      918590
# 5 2017       CHEVR      714655


# Plot across all years
df_freq_vehicle_make <- rbind(df_freq_vehicle_make_2015,df_freq_vehicle_make_2016)
df_freq_vehicle_make <- rbind(df_freq_vehicle_make,df_freq_vehicle_make_2017)


plot_freq_vehicle_make <- ggplot(df_freq_vehicle_make, aes(Year , cnt_tickets, fill = as.factor(`Vehicle Make`))) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Year the ticket is assigned") +
  ylab("Number of Tickets")
plot_freq_vehicle_make


# Conclusion: Almost each year Vehicle make FORD gets the highest number of tickets. Every year the same vehicle makes get the
highest number of tickets.


###############
## 3: A precinct is a police station that has a certain zone of the city under its command.
##    Find the (5 highest) frequencies of:
##    1:  Violating Precincts (this is the precinct of the zone where the violation occurred)
##    2:  Issuing Precincts (this is the precinct that issued the ticket)
###############


## Count of tickets for each Violation Precinct


############### Year 2015 ###############
freq_violation_precinct_2015 <- SparkR::sql("SELECT '2015' as Year,`Violation Precinct`,
                                            count(distinct `Summons Number`) as cnt_tickets \
                                    FROM nycTickets2015_tbl \
                                    GROUP BY `Violation Precinct`
                                    ORDER BY count(*) desc")
```

```
df_freq_violation_precinct_2015 <- head(freq_violation_precinct_2015, n = 5)
df_freq_violation_precinct_2015

#  Year Violation Precinct cnt_tickets
# 1 2015              0    1633006
# 2 2015             19     559716
# 3 2015             18     400887
# 4 2015             14     384596
# 5 2015              1     307808

############### Year 2016 ###############
freq_violation_precinct_2016 <- SparkR::sql("SELECT '2016' as Year,`Violation Precinct`,
                                    count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2016_tbl \
                                GROUP BY `Violation Precinct`
                                ORDER BY count(*) desc")

df_freq_violation_precinct_2016 <- head(freq_violation_precinct_2016, n = 5)
df_freq_violation_precinct_2016

#  Year Violation Precinct cnt_tickets
# 1 2016              0    1868655
# 2 2016             19     554465
# 3 2016             18     331704
# 4 2016             14     324467
# 5 2016              1     303850

############### Year 2017 ###############
freq_violation_precinct_2017 <- SparkR::sql("SELECT '2017' as Year,`Violation Precinct`,
                                    count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2017_tbl \
                                GROUP BY `Violation Precinct`
                                ORDER BY count(*) desc")

df_freq_violation_precinct_2017 <- head(freq_violation_precinct_2017, n = 5)
df_freq_violation_precinct_2017
```

```
#  Year Violation Precinct cnt_tickets
# 1 2017                0    2072400
# 2 2017               19     535671
# 3 2017               14     352450
# 4 2017                1     331810
# 5 2017               18     306920

# Plot across all years
df_freq_violation_precinct <- rbind(df_freq_violation_precinct_2015,df_freq_violation_precinct_2016)
df_freq_violation_precinct <- rbind(df_freq_violation_precinct,df_freq_violation_precinct_2017)

plot_freq_violation_precinct <- ggplot(df_freq_violation_precinct,
                                       aes(Year , cnt_tickets, fill = as.factor(`Violation Precinct`))) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Year the ticket is assigned") +
  ylab("Number of Tickets")
plot_freq_violation_precinct

# Conclusion: Clearly each year Violation Precinct 0 has an exceptional number of tickets which keeps increasing year on year.

##
## Count of tickets for each Issuer Precinct
##
############### Year 2015 ###############
freq_issuer_2015 <- SparkR::sql("SELECT '2015' as Year,`Issuer Precinct`,
                                    count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2015_tbl \
                                GROUP BY `Issuer Precinct`
                                ORDER BY count(*) desc")

df_freq_issuer_2015 <- head(freq_issuer_2015, n = 5)
df_freq_issuer_2015

#  Year Issuer Precinct cnt_tickets
# 1 2015              0    1834343
```

```
# 2 2015           19     544946
# 3 2015           18     391501
# 4 2015           14     369725
# 5 2015            1     298594

############### Year 2016 ###############
freq_issuer_2016 <- SparkR::sql("SELECT '2016' as Year,`Issuer Precinct`,
                                count(distinct `Summons Number`) as cnt_tickets \
                            FROM nycTickets2016_tbl \
                            GROUP BY `Issuer Precinct`
                            ORDER BY count(*) desc")

df_freq_issuer_2016 <- head(freq_issuer_2016, n = 5)
df_freq_issuer_2016

#  Year Issuer Precinct cnt_tickets
# 1 2016            0    2140274
# 2 2016           19     540569
# 3 2016           18     323132
# 4 2016           14     315311
# 5 2016            1     295013

############### Year 2017 ###############
freq_issuer_2017 <- SparkR::sql("SELECT '2017' as Year,`Issuer Precinct`,
                                count(distinct `Summons Number`) as cnt_tickets \
                            FROM nycTickets2017_tbl \
                            GROUP BY `Issuer Precinct`
                            ORDER BY count(*) desc")

df_freq_issuer_2017 <- head(freq_issuer_2017, n = 5)
df_freq_issuer_2017

#  Year Issuer Precinct cnt_tickets
# 1 2017            0    2388479
# 2 2017           19     521513
# 3 2017           14     344977
```

```
# 4 2017            1     321170
# 5 2017           18     296553

# Plot across all years
df_freq_issuer_precinct <- rbind(df_freq_issuer_2015,df_freq_issuer_2016)
df_freq_issuer_precinct <- rbind(df_freq_issuer_precinct,df_freq_issuer_2017)

plot_freq_issuer_precinct <- ggplot(df_freq_issuer_precinct,
                          aes(Year , cnt_tickets, fill = as.factor(`Issuer Precinct`))) +
 geom_bar(stat = "identity", position = "dodge") +
 xlab("Year the ticket is assigned") +
 ylab("Number of Tickets")
plot_freq_issuer_precinct

# Conclusion: Clearly each year Issuer Precinct 0 has an exceptional number of tickets which keeps increasing year on year.

###############
## 4: Find the violation code frequency across 3 precincts which have issued the most number of tickets
##    - do these precinct zones have an exceptionally high frequency of certain violation codes?
##    Are these codes common across precincts?
###############

# From the above question we know the Issuer Precinct with highest number of tickets each year as follows.
# 2015 : 0, 19, 18
# 2016 : 0, 19, 18
# 2017 : 0, 19, 14
# Using the above data

############### Year 2015 ###############
# Count of tickets for each Violation code for issuer precinct (0,19,18)
freq_violation_codes_2015 <- SparkR::sql("SELECT '2015' as Year, `Violation Code`,
                                    count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2015_tbl \
                                WHERE `Issuer Precinct` IN (0,19,18) \
                                GROUP BY `Violation Code` \
                                ORDER BY count(*) desc")
```

```
df_freq_violation_codes_2015 <- head(freq_violation_codes_2015, n = 5)
df_freq_violation_codes_2015

#  Year Violation Code cnt_tickets
# 1 2015           36     761571
# 2 2015            7     662203
# 3 2015           21     240604
# 4 2015            5     195353
# 5 2015           14     185733

############## Year 2016 ##############
# Count of tickets for each Violation code for issuer precinct (0,19,18)
freq_violation_codes_2016 <- SparkR::sql("SELECT '2016' as Year, `Violation Code`,
                                         count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2016_tbl \
                                WHERE `Issuer Precinct` IN (0,19,18) \
                                GROUP BY `Violation Code` \
                                ORDER BY count(*) desc")
df_freq_violation_codes_2016 <- head(freq_violation_codes_2016, n = 5)
df_freq_violation_codes_2016

#  Year Violation Code cnt_tickets
# 1 2016           36    1253511
# 2 2016            7     492469
# 3 2016           21     299409
# 4 2016           14     167587
# 5 2016            5     112376

############## Year 2017 ##############
# Count of tickets for each Violation code for issuer precinct (0,19,14)
freq_violation_codes_2017 <- SparkR::sql("SELECT '2017' as Year, `Violation Code`,
                                         count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2017_tbl \
                                WHERE `Issuer Precinct` IN (0,19,14) \
                                GROUP BY `Violation Code` \
                                ORDER BY count(*) desc")
```

```
df_freq_violation_codes_2017 <- head(freq_violation_codes_2017, n = 5)
df_freq_violation_codes_2017

#  Year Violation Code cnt_tickets
# 1 2017           36     1400614
# 2 2017            7      516390
# 3 2017           21      325435
# 4 2017            5      145643
# 5 2017           14      138488

# Plot across all years
df_freq_violation_codes <- rbind(df_freq_violation_codes_2015,df_freq_violation_codes_2016)
df_freq_violation_codes <- rbind(df_freq_violation_codes,df_freq_violation_codes_2017)

plot_freq_violation_codes <- ggplot(df_freq_violation_codes,
                          aes(Year , cnt_tickets, fill = as.factor(`Violation Code`))) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Year the ticket is assigned") +
  ylab("Number of Tickets")
plot_freq_violation_codes

# Conclusion: For all years across top issuer precincts the following violation codes are top 5 always
# Violation Code 36, 7, 21, 14 and 5 in descending order of occurrence

###############
## 5: You'd want to find out the properties of parking violations across different times of the day:
##          1 >> The Violation Time field is specified in a strange format. Find a way to make this into a time attribute that you can use to divide into groups.
##          2 >> Find a way to deal with missing values, if any.
##          3 >> Divide 24 hours into 6 equal discrete bins of time.
##                    The intervals you choose are at your discretion. For each of these groups, find the 3 most commonly occurring violations
##    4 >> Now, try another direction. For the 3 most commonly occurring violation codes,
##                                find the most common times of day (in terms of the bins from the previous part)
###############
```

```
##

## For each of these groups, find the 3 most commonly occurring violations

##


############## Year 2015 ##############

Violation_time_IsNull <- where(nycTickets2015, isNull(nycTickets2015$`Violation Time`))

count(Violation_time_IsNull)

# 1715 Na values


time_evaluation_2015 <- SparkR::sql("SELECT `Summons Number` as Summons_Number,CONCAT(CONCAT(CONCAT(Hr,':'), Min),
CONCAT(' ',am_pm)) as Violation_Time, \
                        CASE  WHEN time_of_day >= 0 AND time_of_day < 4 THEN 'Early Morning' \
                        WHEN time_of_day >= 4 AND time_of_day < 8 THEN 'Late Morning' \
                        WHEN time_of_day >= 8 AND time_of_day < 12 THEN 'Early Afternoon' \
                        WHEN time_of_day >= 12 AND time_of_day < 16 THEN 'Late Afternoon' \
                        WHEN time_of_day >= 16 AND time_of_day < 20 THEN 'Early Evening' \
                        WHEN time_of_day >= 20 THEN 'Late Evening' \
                        ELSE 'Time Not Provided' \
                        END as time_bin \
                        FROM \
                        (SELECT `Summons Number`,substr(`Violation Time`,1,2) as Hr, \
                        substr(`Violation Time`,1,2) as Min, \
                        CASE WHEN substr(`Violation Time`,-1) = 'A' THEN 'AM' \
                        ELSE 'PM' \
                        END as am_pm,  \
                        CASE WHEN substr(`Violation Time`,-1) = 'A' \
                        THEN substr((substr(`Violation Time`,1,2) + 0),1,1) \
                        ELSE substr((substr(`Violation Time`,1,2) + 12),1,2) \
                        END as time_of_day
                        FROM nycTickets2015_tbl) ")


nycTickets2015_time <- join(nycTickets2015, time_evaluation_2015, nycTickets2015$`Summons Number` ==
time_evaluation_2015$Summons_Number, "left")


createOrReplaceTempView(nycTickets2015_time, "nycTickets2015_time_tbl")
```

```
freq_violation_2015 <- SparkR::sql("SELECT time_bin,`Violation Code`,count(distinct `Summons Number`) as cnt_tickets \
                        FROM nycTickets2015_time_tbl \
                        GROUP BY time_bin,`Violation Code` \
                        ORDER BY time_bin,count(*) desc")

df_freq_violation_2015 <- collect(freq_violation_2015)

# Most commonly occurring violation codes for different time of days

# Year 2015
head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Early Morning',], n = 3)
#      time_bin Violation Code cnt_tickets
# 1 Early Morning           21     734165
# 2 Early Morning           38     205820
# 3 Early Morning           14     168314

head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Late Morning',], n = 3)
#       time_bin Violation Code cnt_tickets
# 1 Late Morning           38     241327
# 2 Late Morning           37     175802
# 3 Late Morning            7     168888

head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Early Afternoon',], n = 3)
#         time_bin Violation Code cnt_tickets
# 1 Early Afternoon           21     525430
# 2 Early Afternoon           38     243897
# 3 Early Afternoon           36     196896

head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Late Afternoon',], n = 3)
#         time_bin Violation Code cnt_tickets
# 1 Late Afternoon           38     432218
# 2 Late Afternoon           37     324892
# 3 Late Afternoon           36     220661

head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Early Evening',], n = 3)
#        time_bin Violation Code cnt_tickets
```

```
# 1 Early Evening          38     198472
# 2 Early Evening          21     130163
# 3 Early Evening           7     124456

head(df_freq_violation_2015[df_freq_violation_2015$time_bin == 'Late Evening',], n = 3)
#       time_bin Violation Code cnt_tickets
# 1 Late Evening          14     134458
# 2 Late Evening          21     106858
# 3 Late Evening          40      91344

df_freq_violation_2015$time_bin <- as.factor(df_freq_violation_2015$time_bin)
levels(df_freq_violation_2015$time_bin) <- c("Early Morning", "Late Morning",
                                "Early Afternoon","Late Afternoon",
                                "Early Evening","Late Evening","Time Not Provided")

plot_freq_violation_2015 <- ggplot(df_freq_violation_2015, aes(time_bin, cnt_tickets)) +
  geom_bar(stat = "identity", fill = "dark red") +
  xlab("Time Bin") +
  ylab("Frequency of Tickets")
plot_freq_violation_2015


############### Year 2016 ###############
Violation_time_IsNull <- where(nycTickets2016, isNull(nycTickets2016$`Violation Time`))
count(Violation_time_IsNull)
# 4280 Na values

time_evaluation_2016 <- SparkR::sql("SELECT `Summons Number` as Summons_Number,CONCAT(CONCAT(CONCAT(Hr,':'), Min),
CONCAT(' ',am_pm)) as Violation_Time, \
                        CASE  WHEN time_of_day >= 0 AND time_of_day < 4 THEN 'Early Morning' \
                        WHEN time_of_day >= 4 AND time_of_day < 8 THEN 'Late Morning' \
                        WHEN time_of_day >= 8 AND time_of_day < 12 THEN 'Early Afternoon' \
                        WHEN time_of_day >= 12 AND time_of_day < 16 THEN 'Late Afternoon' \
                        WHEN time_of_day >= 16 AND time_of_day < 20 THEN 'Early Evening' \
                        WHEN time_of_day >= 20 THEN 'Late Evening' \
                        ELSE 'Time Not Provided' \
```

```
                    END as time_bin \
                    FROM \
                    (SELECT `Summons Number`,substr(`Violation Time`,1,2) as Hr, \
                    substr(`Violation Time`,1,2) as Min, \
                    CASE WHEN substr(`Violation Time`,-1) = 'A' THEN 'AM' \
                    ELSE 'PM' \
                    END as am_pm,  \
                    CASE WHEN substr(`Violation Time`,-1) = 'A' \
                    THEN substr((substr(`Violation Time`,1,2) + 0),1,1) \
                    ELSE substr((substr(`Violation Time`,1,2) + 12),1,2) \
                    END as time_of_day
                    FROM nycTickets2016_tbl) ")

nycTickets2016_time <- join(nycTickets2016, time_evaluation_2016, nycTickets2016$`Summons Number` ==
time_evaluation_2016$Summons_Number, "left")

createOrReplaceTempView(nycTickets2016_time, "nycTickets2016_time_tbl")

freq_violation_2016 <- SparkR::sql("SELECT time_bin,`Violation Code`,count(distinct `Summons Number`) as cnt_tickets \
                    FROM nycTickets2016_time_tbl \
                    GROUP BY time_bin,`Violation Code` \
                    ORDER BY time_bin,count(*) desc")

df_freq_violation_2016 <- collect(freq_violation_2016)

# Most commonly occurring violation codes for different time of days

# Year 2016
head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Early Morning',], n = 3)
#      time_bin Violation Code cnt_tickets
# 1 Early Morning          21     754150
# 2 Early Morning          36     262974
# 3 Early Morning          38     173463

head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Late Morning',], n = 3)
#       time_bin Violation Code cnt_tickets
```

```
# 1 Late Morning          38     211267
# 2 Late Morning          37     161655
# 3 Late Morning          14     134976

head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Early Afternoon',], n = 3)
#        time_bin Violation Code cnt_tickets
# 1 Early Afternoon          21     527202
# 2 Early Afternoon          36     323818
# 3 Early Afternoon          38     215021

head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Late Afternoon',], n = 3)
#         time_bin Violation Code cnt_tickets
# 1 Late Afternoon          36     378435
# 2 Late Afternoon          38     367579
# 3 Late Afternoon          37     297619

head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Early Evening',], n = 3)
#        time_bin Violation Code cnt_tickets
# 1 Early Evening          38     173897
# 2 Early Evening          36     167282
# 3 Early Evening          21     131120

head(df_freq_violation_2016[df_freq_violation_2016$time_bin == 'Late Evening',], n = 3)
#        time_bin Violation Code cnt_tickets
# 1 Late Evening          14     140111
# 2 Late Evening          21     114029
# 3 Late Evening          40      91692

df_freq_violation_2016$time_bin <- as.factor(df_freq_violation_2016$time_bin)

levels(df_freq_violation_2016$time_bin) <- c("Early Morning", "Late Morning",
                                             "Early Afternoon","Late Afternoon",
                                             "Early Evening","Late Evening","Time Not Provided")

plot_freq_violation_2016 <- ggplot(df_freq_violation_2016, aes(time_bin, cnt_tickets)) +
  geom_bar(stat = "identity", fill = "dark blue") +
```

```
  xlab("Time Bin") +
  ylab("Frequency of Tickets")
plot_freq_violation_2016



############## Year 2017 ##############
Violation_time_IsNull <- where(nycTickets2017, isNull(nycTickets2017$`Violation Time`))
count(Violation_time_IsNull)
# 63 Na values


time_evaluation_2017 <- SparkR::sql("SELECT `Summons Number` as Summons_Number,CONCAT(CONCAT(CONCAT(Hr,':'), Min), CONCAT(' ',am_pm)) as Violation_Time, \
                        CASE  WHEN time_of_day >= 0 AND time_of_day < 4 THEN 'Early Morning' \
                        WHEN time_of_day >= 4 AND time_of_day < 8 THEN 'Late Morning' \
                        WHEN time_of_day >= 8 AND time_of_day < 12 THEN 'Early Afternoon' \
                        WHEN time_of_day >= 12 AND time_of_day < 16 THEN 'Late Afternoon' \
                        WHEN time_of_day >= 16 AND time_of_day < 20 THEN 'Early Evening' \
                        WHEN time_of_day >= 20 THEN 'Late Evening' \
                        ELSE 'Time Not Provided' \
                        END as time_bin \
                        FROM \
                        (SELECT `Summons Number`,substr(`Violation Time`,1,2) as Hr, \
                        substr(`Violation Time`,1,2) as Min, \
                        CASE WHEN substr(`Violation Time`,-1) = 'A' THEN 'AM' \
                        ELSE 'PM' \
                        END as am_pm,  \
                        CASE WHEN substr(`Violation Time`,-1) = 'A' \
                        THEN substr((substr(`Violation Time`,1,2) + 0),1,1) \
                        ELSE substr((substr(`Violation Time`,1,2) + 12),1,2) \
                        END as time_of_day
                        FROM nycTickets2017_tbl) ")


nycTickets2017_time <- join(nycTickets2017, time_evaluation_2017, nycTickets2017$`Summons Number` == time_evaluation_2017$Summons_Number, "left")


createOrReplaceTempView(nycTickets2017_time, "nycTickets2017_time_tbl")
```

```
freq_violation_2017 <- SparkR::sql("SELECT time_bin,`Violation Code`,count(distinct `Summons Number`) as cnt_tickets \
                        FROM nycTickets2017_time_tbl \
                        GROUP BY time_bin,`Violation Code` \
                        ORDER BY time_bin,count(*) desc")

df_freq_violation_2017 <- collect(freq_violation_2017)

# Most commonly occurring violation codes for different time of days

# Year 2017
head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Early Morning',], n = 3)
#      time_bin Violation Code cnt_tickets
# 1 Early Morning           21     746351
# 2 Early Morning           36     335271
# 3 Early Morning           38     154809

head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Late Morning',], n = 3)
#      time_bin Violation Code cnt_tickets
# 1 Late Morning           38     203232
# 2 Late Morning           37     145784
# 3 Late Morning           14     144749

head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Early Afternoon',], n = 3)
#       time_bin Violation Code cnt_tickets
# 1 Early Afternoon           21     513799
# 2 Early Afternoon           36     416151
# 3 Early Afternoon           38     192173

head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Late Afternoon',], n = 3)
#         time_bin Violation Code cnt_tickets
# 1 Late Afternoon           36     376961
# 2 Late Afternoon           38     356253
# 3 Late Afternoon           37     265848

head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Early Evening',], n = 3)
```

```
#        time_bin Violation Code cnt_tickets
# 1 Early Evening           36     211434
# 2 Early Evening           38     153537
# 3 Early Evening           21     144082

head(df_freq_violation_2017[df_freq_violation_2017$time_bin == 'Late Evening',], n = 3)
#        time_bin Violation Code cnt_tickets
# 1 Late Evening           14     141276
# 2 Late Evening           21     119469
# 3 Late Evening           40     112186

df_freq_violation_2017$time_bin <- as.factor(df_freq_violation_2017$time_bin)
levels(df_freq_violation_2017$time_bin) <- c("Early Morning", "Late Morning",
                                "Early Afternoon","Late Afternoon",
                                "Early Evening","Late Evening","Time Not Provided")

plot_freq_violation_2017 <- ggplot(df_freq_violation_2017, aes(time_bin, cnt_tickets)) +
  geom_bar(stat = "identity", fill = "dark green") +
  xlab("Time Bin") +
  ylab("Frequency of Tickets")
plot_freq_violation_2017

##
## For the 3 most commonly occurring violation codes, find the most common times of day (in terms of the bins from the previous part).
##

############### Year 2015 ###############

# Top 3 violation codes for year 2015
freq_code_violation_2015 <- SparkR::sql("SELECT `Violation Code`,
                                count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2015_time_tbl \
                                GROUP BY `Violation Code` \
                                ORDER BY count(*) desc")
```

```
head(select(freq_violation_2015,freq_violation_2015$`Violation Code`), n = 3)
# Violation Code
#    21
#    38
#    14

# By frequency the 3 most commonly occurring Violation Codes are 21, 38 and 14
# For each of them the most commonly occurring time bins are
df_code_1_2015 <- head(df_freq_violation_2015[df_freq_violation_2015$`Violation Code` == '21',]
            [order(-df_freq_violation_2015[df_freq_violation_2015$`Violation Code` == '21',]$cnt_tickets),], n=3)
df_code_2_2015 <- head(df_freq_violation_2015[df_freq_violation_2015$`Violation Code` == '38',]
                [order(-df_freq_violation_2015[df_freq_violation_2015$`Violation Code` == '21',]$cnt_tickets),], n=3)
df_code_3_2015 <- head(df_freq_violation_2015[df_freq_violation_2015$`Violation Code` == '14',]
                [order(-df_freq_violation_2015[df_freq_violation_2015$`Violation Code` == '21',]$cnt_tickets),], n=3)

# Year 2015
df_code_1_2015
#          time_bin Violation Code cnt_tickets
# 1 Early Afternoon           21      734165
# 2   Early Morning           21      525430
# 3    Late Evening           21      130163

df_code_2_2015
#          time_bin Violation Code cnt_tickets
# 1 Late Afternoon           38      432218
# 2  Early Morning           38      243897
# 3  Early Evening           38      241327

df_code_3_2015
#          time_bin Violation Code cnt_tickets
# 1  Late Afternoon           14      207927
# 2 Early Afternoon           14      168314
# 3   Early Morning           14      159848

df_code_bins_2015 <- rbind(rbind(df_code_1_2015,df_code_2_2015),df_code_3_2015)
df_code_bins_2015
```

```
plot_code_bins_2015 <- ggplot(df_code_bins_2015, aes(as.factor(`Violation Code`), cnt_tickets,fill = time_bin)) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Violation Code") +
  ylab("Frequency of Tickets")
plot_code_bins_2015

############### Year 2016 ###############

# Top 3 violation codes for year 2016
freq_code_violation_2016 <- SparkR::sql("SELECT `Violation Code`,
                                count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2016_time_tbl \
                                GROUP BY `Violation Code` \
                                ORDER BY count(*) desc")

head(select(freq_code_violation_2016,freq_code_violation_2016$`Violation Code`), n = 3)
# Violation Code
#    21
#    36
#    38

# By frequency the 3 most commonly occurring Violation Codes are 21, 38 and 14
# For each of them the most commonly occurring time bins are

df_code_1_2016 <- head(df_freq_violation_2016[df_freq_violation_2016$`Violation Code` == '21',]
    [order(-df_freq_violation_2016[df_freq_violation_2016$`Violation Code` == '21',]$cnt_tickets),], n=3)
df_code_2_2016 <- head(df_freq_violation_2016[df_freq_violation_2016$`Violation Code` == '36',]
                  [order(-df_freq_violation_2016[df_freq_violation_2016$`Violation Code` == '36',]$cnt_tickets),], n=3)
df_code_3_2016 <- head(df_freq_violation_2016[df_freq_violation_2016$`Violation Code` == '38',]
                  [order(-df_freq_violation_2016[df_freq_violation_2016$`Violation Code` == '38',]$cnt_tickets),], n=3)

# Year 2016
df_code_1_2016
#          time_bin Violation Code cnt_tickets
# 1   Early Morning            21     754150
```

```
# 2 Early Afternoon           21     527202
# 3   Early Evening           21     131120

df_code_2_2016
#          time_bin Violation Code cnt_tickets
# 1  Late Afternoon           36     378435
# 2 Early Afternoon           36     323818
# 3   Early Morning           36     262974

df_code_3_2016
#          time_bin Violation Code cnt_tickets
# 1  Late Afternoon           38     367579
# 2 Early Afternoon           38     215021
# 3    Late Morning           38     211267

df_code_bins_2016 <- rbind(rbind(df_code_1_2016,df_code_2_2016),df_code_3_2016)
df_code_bins_2016

plot_code_bins_2016 <- ggplot(df_code_bins_2016, aes(as.factor(`Violation Code`), cnt_tickets,fill = time_bin)) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Violation Code") +
  ylab("Frequency of Tickets")
plot_code_bins_2016

############### Year 2017 ###############

# Top 3 violation codes for year 2016
freq_code_violation_2017 <- SparkR::sql("SELECT `Violation Code`,
                                count(distinct `Summons Number`) as cnt_tickets \
                                FROM nycTickets2017_time_tbl \
                                GROUP BY `Violation Code` \
                                ORDER BY count(*) desc")

head(select(freq_code_violation_2017,freq_code_violation_2017$`Violation Code`), n = 3)
# Violation Code
#    21
```

```
#    36
#    38

# By frequency the 3 most commonly occurring Violation Codes are 21, 38 and 14
# For each of them the most commonly occurring time bins are

df_code_1_2017 <- head(df_freq_violation_2017[df_freq_violation_2017$`Violation Code` == '21',]
                       [order(-df_freq_violation_2017[df_freq_violation_2017$`Violation Code` == '21',]$cnt_tickets),], n=3)
df_code_2_2017 <- head(df_freq_violation_2017[df_freq_violation_2017$`Violation Code` == '36',]
                       [order(-df_freq_violation_2017[df_freq_violation_2017$`Violation Code` == '36',]$cnt_tickets),], n=3)
df_code_3_2017 <- head(df_freq_violation_2017[df_freq_violation_2017$`Violation Code` == '38',]
                       [order(-df_freq_violation_2017[df_freq_violation_2017$`Violation Code` == '38',]$cnt_tickets),], n=3)

# Year 2017
df_code_1_2017
#         time_bin Violation Code cnt_tickets
# 1   Early Morning           21     746351
# 2 Early Afternoon           21     513799
# 3   Early Evening           21     144082

df_code_2_2017
#         time_bin Violation Code cnt_tickets
# 1 Early Afternoon           36     416151
# 2  Late Afternoon           36     376961
# 3   Early Morning           36     335271

df_code_3_2017
#         time_bin Violation Code cnt_tickets
# 1  Late Afternoon           38     356253
# 2    Late Morning           38     203232
# 3 Early Afternoon           38     192173

df_code_bins_2017 <- rbind(rbind(df_code_1_2017,df_code_2_2017),df_code_3_2017)
df_code_bins_2017

plot_code_bins_2017 <- ggplot(df_code_bins_2017, aes(as.factor(`Violation Code`), cnt_tickets,fill = time_bin)) +
```

```
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Violation Code") +
  ylab("Frequency of Tickets")
plot_code_bins_2017



###############
##          6: Let’s try and find some seasonality in this data
##                    1 >> First, divide the year into some number of seasons, and find frequencies of tickets for each season.
##                    2 >> Then, find the 3 most common violations for each of these season
###############

# We will be dividing the year in the following Seasons
# First Quarter : 1st July to 30th September
# Second Quarter : 1st October to 31st December
# Third Quarter : 1st January to 31st March
# Fourth Quarter : 1st April to 30th June

##
## frequencies of tickets for each season.
##

############### Year 2015 ###############
season_evaluation_2015 <- SparkR::sql("SELECT `Summons Number` as Summons_Number,
                          TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) AS Issue_Date,
                          CASE
                          WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
                          TO_DATE(CAST(UNIX_TIMESTAMP('07/01/2014', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                          TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                          TO_DATE(CAST(UNIX_TIMESTAMP('10/01/2014', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'First Quarter'
                          WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
                          TO_DATE(CAST(UNIX_TIMESTAMP('10/01/2014', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                          TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                          TO_DATE(CAST(UNIX_TIMESTAMP('01/01/2015', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'Second Quarter'

                          WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
```

```
                    TO_DATE(CAST(UNIX_TIMESTAMP('01/01/2015', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                    TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                    TO_DATE(CAST(UNIX_TIMESTAMP('04/01/2015', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'Third Quarter'
                    WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
                    TO_DATE(CAST(UNIX_TIMESTAMP('04/01/2015', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                    TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                    TO_DATE(CAST(UNIX_TIMESTAMP('07/01/2015', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'Fourth Quarter'
                    ELSE 'Season Not Defined'
                    END AS season_bin

                    FROM nycTickets2015_tbl")
nycTickets2015_season <- join(nycTickets2015, season_evaluation_2015, nycTickets2015$`Summons Number` ==
season_evaluation_2015$Summons_Number, "left")


createOrReplaceTempView(nycTickets2015_season, "nycTickets2015_season_tbl")


# Frequencies of tickets for each season.
season_evaluation_2015 <- SparkR::sql("SELECT '2015' as Year, season_bin as Season, \
                    count( distinct `Summons Number`) as count_tickets \
                    FROM nycTickets2015_season_tbl \
                    WHERE season_bin <> 'Season Not Defined' \
                    GROUP BY season_bin \
                    ORDER BY count( distinct `Summons Number`) desc")


df_season_evaluation_2015 <- head(season_evaluation_2015, n = 4)
df_season_evaluation_2015


#   2015
#           Season                          count_tickets
#           Fourth Quarter      2907331
#           First Quarter       2788963
#           Third Quarter       2466640
#           Second Quarter      2435101


# The frequency of tickets given are approximately equally divided across the quarters. Fourth quarter has the highest no of tickets.
```

```
############## 2016 ##############
season_evaluation_2016 <- SparkR::sql("SELECT `Summons Number` as Summons_Number,
                        TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) AS Issue_Date,
                        CASE
                        WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
                        TO_DATE(CAST(UNIX_TIMESTAMP('07/01/2015', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                        TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                        TO_DATE(CAST(UNIX_TIMESTAMP('10/01/2015', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'First Quarter'
                        WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
                        TO_DATE(CAST(UNIX_TIMESTAMP('10/01/2015', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                        TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                        TO_DATE(CAST(UNIX_TIMESTAMP('01/01/2016', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'Second Quarter'

                        WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
                        TO_DATE(CAST(UNIX_TIMESTAMP('01/01/2016', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                        TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                        TO_DATE(CAST(UNIX_TIMESTAMP('04/01/2016', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'Third Quarter'
                        WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
                        TO_DATE(CAST(UNIX_TIMESTAMP('04/01/2016', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                        TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                        TO_DATE(CAST(UNIX_TIMESTAMP('07/01/2016', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'Fourth Quarter'
                        ELSE 'Season Not Defined'
                        END AS season_bin

                        FROM nycTickets2016_tbl")

nycTickets2016_season <- join(nycTickets2016, season_evaluation_2016, nycTickets2016$`Summons Number` ==
season_evaluation_2016$Summons_Number, "left")

createOrReplaceTempView(nycTickets2016_season, "nycTickets2016_season_tbl")

# Frequencies of tickets for each season.
season_evaluation_2016 <- SparkR::sql("SELECT '2016' as Year, season_bin as Season, \
                        count( distinct `Summons Number`) as count_tickets \
                        FROM nycTickets2016_season_tbl \
                        WHERE season_bin <> 'Season Not Defined' \
                        GROUP BY season_bin \
```

```
                    ORDER BY count( distinct `Summons Number`) desc")

df_season_evaluation_2016 <- head(season_evaluation_2016, n = 4)
df_season_evaluation_2016

#  2016
#          Season                          count_tickets
#          Second Quarter      2799402
#          First Quarter       2726774
#          Third Quarter       2668423
#          Fourth Quarter      2202295

# The frequency of tickets given are approximately equally divided across the quarters. Second quarter has the highest no of tickets.

############### 2017 ###############
season_evaluation_2017 <- SparkR::sql("SELECT `Summons Number` as Summons_Number,
                        TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) AS Issue_Date,
                        CASE
                        WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
                        TO_DATE(CAST(UNIX_TIMESTAMP('07/01/2016', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                        TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                        TO_DATE(CAST(UNIX_TIMESTAMP('10/01/2016', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'First Quarter'
                        WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
                        TO_DATE(CAST(UNIX_TIMESTAMP('10/01/2016', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                        TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                        TO_DATE(CAST(UNIX_TIMESTAMP('01/01/2017', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'Second Quarter'

                        WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
                        TO_DATE(CAST(UNIX_TIMESTAMP('01/01/2017', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                        TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) < \
                        TO_DATE(CAST(UNIX_TIMESTAMP('04/01/2017', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'Third Quarter'
                        WHEN TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) >=
                        TO_DATE(CAST(UNIX_TIMESTAMP('04/01/2017', 'MM/dd/yyyy') AS TIMESTAMP)) AND
                        TO_DATE(CAST(UNIX_TIMESTAMP(`Issue Date`, 'MM/dd/yyyy') AS TIMESTAMP)) <
                        TO_DATE(CAST(UNIX_TIMESTAMP('07/01/2017', 'MM/dd/yyyy') AS TIMESTAMP)) THEN 'Fourth Quarter'
                        ELSE 'Season Not Defined'
```

```
                        END AS season_bin

                        FROM nycTickets2017_tbl")

nycTickets2017_season <- join(nycTickets2017, season_evaluation_2017, nycTickets2017$`Summons Number` ==
season_evaluation_2017$Summons_Number, "left")

createOrReplaceTempView(nycTickets2017_season, "nycTickets2017_season_tbl")

# Frequencies of tickets for each season.
season_evaluation_2017 <- SparkR::sql("SELECT '2017' as Year, season_bin as Season, \
                        count( distinct `Summons Number`) as count_tickets \
                        FROM nycTickets2017_season_tbl \
                        WHERE season_bin <> 'Season Not Defined' \
                        GROUP BY season_bin \
                        ORDER BY count( distinct `Summons Number`) desc")

df_season_evaluation_2017 <- head(season_evaluation_2017, n = 4)
df_season_evaluation_2017

#   2017
#           Season                                  count_tickets
#           Fourth Quarter      2760833
#           Third Quarter       2669069
#           Second Quarter      2647391
#           First Quarter       2462270

# The frequency of tickets given are approximately equally divided across the quarters. Fourth quarter has the highest no of tickets.

# Plot across all years
df_season_evaluation <- rbind(df_season_evaluation_2015,df_season_evaluation_2016)
df_season_evaluation <- rbind(df_season_evaluation,df_season_evaluation_2017)

levels(df_season_evaluation$Season) <- c("First Quarter", "Second Quarter", "Third Quarter","Fourth Quarter")

plot_season_evaluation <- ggplot(df_season_evaluation,
```

```
                        aes(Year , count_tickets, fill = Season)) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Year the ticket is assigned") +
  ylab("Number of Tickets")
plot_season_evaluation

##
## Find the 3 most common violations for each of these season
##
############## Year 2015 ##############
# Frequency of most common violations for each of these season
violation_first_quarter_2015 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                    count( distinct `Summons Number`) as count_tickets \
                                FROM nycTickets2015_season_tbl \
                                WHERE season_bin = 'First Quarter' \
                                GROUP BY season_bin,`Violation Code` \
                                ORDER BY count( distinct `Summons Number`) desc")

violation_second_quarter_2015 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                    count( distinct `Summons Number`) as count_tickets \
                                FROM nycTickets2015_season_tbl \
                                WHERE season_bin = 'Second Quarter' \
                                GROUP BY season_bin,`Violation Code` \
                                ORDER BY count( distinct `Summons Number`) desc")

violation_third_quarter_2015 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                    count( distinct `Summons Number`) as count_tickets \
                                FROM nycTickets2015_season_tbl \
                                WHERE season_bin = 'Third Quarter' \
                                GROUP BY season_bin,`Violation Code` \
                                ORDER BY count( distinct `Summons Number`) desc")

violation_fourth_quarter_2015 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                    count( distinct `Summons Number`) as count_tickets \
                                FROM nycTickets2015_season_tbl \
                                WHERE season_bin = 'Fourth Quarter' \
```

```
                    GROUP BY season_bin,`Violation Code` \
                    ORDER BY count( distinct `Summons Number`) desc")

df_violation_first_quarter_2015 <- head(violation_first_quarter_2015,n=3)
df_violation_second_quarter_2015 <- head(violation_second_quarter_2015,n=3)
df_violation_third_quarter_2015 <- head(violation_third_quarter_2015,n=3)
df_violation_fourth_quarter_2015 <- head(violation_fourth_quarter_2015,n=3)

# Year 2015
# First Quarter
df_violation_first_quarter_2015
#       Season Violation Code count_tickets
# 1 First Quarter          21      397809
# 2 First Quarter          38      348466
# 3 First Quarter          14      234565

# Second Quarter
df_violation_second_quarter_2015
# 1 Second Quarter         21      350517
# 2 Second Quarter         38      292637
# 3 Second Quarter         14      207365

# Third Quarter
df_violation_third_quarter_2015
#       Season Violation Code count_tickets
# 1 Third Quarter          38      336746
# 2 Third Quarter          21      281386
# 3 Third Quarter          14      219828

# Fourth Quarter
df_violation_fourth_quarter_2015
# 1 Fourth Quarter         21      439516
# 2 Fourth Quarter         38      327158
# 3 Fourth Quarter         14      246660

df_season_2015 <- rbind(rbind(rbind(df_violation_first_quarter_2015,
```

```
                    df_violation_second_quarter_2015),
                df_violation_third_quarter_2015),
            df_violation_fourth_quarter_2015)

df_season_2015$Season <- as.factor(df_season_2015$Season)
levels(df_season_2015$Season) <- c("First Quarter", "Second Quarter", "Third Quarter","Fourth Quarter")

plot_season_violation_2015 <- ggplot(df_season_2015,
                    aes(Season , count_tickets, fill = as.factor(`Violation Code`))) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Season for year 2015") +
  ylab("Number of Tickets")
plot_season_violation_2015

############### Year 2016 ###############
# Frequency of most common violations for each of these season
violation_first_quarter_2016 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                count( distinct `Summons Number`) as count_tickets \
                                FROM nycTickets2016_season_tbl \
                                WHERE season_bin = 'First Quarter' \
                                GROUP BY season_bin,`Violation Code` \
                                ORDER BY count( distinct `Summons Number`) desc")

violation_second_quarter_2016 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                 count( distinct `Summons Number`) as count_tickets \
                                 FROM nycTickets2016_season_tbl \
                                 WHERE season_bin = 'Second Quarter' \
                                 GROUP BY season_bin,`Violation Code` \
                                 ORDER BY count( distinct `Summons Number`) desc")

violation_third_quarter_2016 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                count( distinct `Summons Number`) as count_tickets \
                                FROM nycTickets2016_season_tbl \
                                WHERE season_bin = 'Third Quarter' \
                                GROUP BY season_bin,`Violation Code` \
                                ORDER BY count( distinct `Summons Number`) desc")
```

```
violation_fourth_quarter_2016 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                count( distinct `Summons Number`) as count_tickets \
                                FROM nycTickets2016_season_tbl \
                                WHERE season_bin = 'Fourth Quarter' \
                                GROUP BY season_bin,`Violation Code` \
                                ORDER BY count( distinct `Summons Number`) desc")

df_violation_first_quarter_2016 <- head(violation_first_quarter_2016,n=3)
df_violation_second_quarter_2016 <- head(violation_second_quarter_2016,n=3)
df_violation_third_quarter_2016 <- head(violation_third_quarter_2016,n=3)
df_violation_fourth_quarter_2016 <- head(violation_fourth_quarter_2016,n=3)

# Year 2016
# First Quarter
df_violation_first_quarter_2016
#        Season Violation Code count_tickets
# 1 First Quarter          21       403309
# 2 First Quarter          38       305341
# 3 First Quarter          14       234798

# Second Quarter
df_violation_second_quarter_2016
#         Season Violation Code count_tickets
# 1 Second Quarter          36       433966
# 2 Second Quarter          21       429429
# 3 Second Quarter          38       274424

# Third Quarter
df_violation_third_quarter_2016
#        Season Violation Code count_tickets
# 1 Third Quarter          21       349297
# 2 Third Quarter          36       341787
# 3 Third Quarter          38       308987

# Fourth Quarter
```

```
df_violation_fourth_quarter_2016
#         Season Violation Code count_tickets
# 1 Fourth Quarter           21       315234
# 2 Fourth Quarter           36       273455
# 3 Fourth Quarter           38       238083

df_season_2016 <- rbind(rbind(rbind(df_violation_first_quarter_2016,
                            df_violation_second_quarter_2016),
                      df_violation_third_quarter_2016),
                df_violation_fourth_quarter_2016)

df_season_2016$Season <- as.factor(df_season_2016$Season)
levels(df_season_2016$Season) <- c("First Quarter", "Second Quarter", "Third Quarter","Fourth Quarter")

plot_season_violation_2016 <- ggplot(df_season_2016,
                             aes(Season , count_tickets, fill = as.factor(`Violation Code`))) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Season for year 2016") +
  ylab("Number of Tickets")
plot_season_violation_2016

############### Year 2017 ###############
# Frequency of most common violations for each of these season
violation_first_quarter_2017 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                count( distinct `Summons Number`) as count_tickets \
                                FROM nycTickets2017_season_tbl \
                                WHERE season_bin = 'First Quarter' \
                                GROUP BY season_bin,`Violation Code` \
                                ORDER BY count( distinct `Summons Number`) desc")

violation_second_quarter_2017 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                 count( distinct `Summons Number`) as count_tickets \
                                 FROM nycTickets2017_season_tbl \
                                 WHERE season_bin = 'Second Quarter' \
                                 GROUP BY season_bin,`Violation Code` \
                                 ORDER BY count( distinct `Summons Number`) desc")
```

```
violation_third_quarter_2017 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                count( distinct `Summons Number`) as count_tickets \
                                FROM nycTickets2017_season_tbl \
                                WHERE season_bin = 'Third Quarter' \
                                GROUP BY season_bin,`Violation Code` \
                                ORDER BY count( distinct `Summons Number`) desc")

violation_fourth_quarter_2017 <- SparkR::sql("SELECT season_bin as Season, `Violation Code`,\
                                 count( distinct `Summons Number`) as count_tickets \
                                 FROM nycTickets2017_season_tbl \
                                 WHERE season_bin = 'Fourth Quarter' \
                                 GROUP BY season_bin,`Violation Code` \
                                 ORDER BY count( distinct `Summons Number`) desc")

df_violation_first_quarter_2017 <- head(violation_first_quarter_2017,n=3)
df_violation_second_quarter_2017 <- head(violation_second_quarter_2017,n=3)
df_violation_third_quarter_2017 <- head(violation_third_quarter_2017,n=3)
df_violation_fourth_quarter_2017 <- head(violation_fourth_quarter_2017,n=3)

# Year 2017
# First Quarter
df_violation_first_quarter_2017
#        Season Violation Code count_tickets
# 1 First Quarter          21       385410
# 2 First Quarter          38       244972
# 3 First Quarter          36       239879

# Second Quarter
df_violation_second_quarter_2017
#         Season Violation Code count_tickets
# 1 Second Quarter          36       442593
# 2 Second Quarter          21       347227
# 3 Second Quarter          38       263382

# Third Quarter
```

```
df_violation_third_quarter_2017
#         Season Violation Code count_tickets
# 1 Third Quarter          21       373874
# 2 Third Quarter          36       348240
# 3 Third Quarter          38       287000


# Fourth Quarter
df_violation_fourth_quarter_2017
#         Season Violation Code count_tickets
# 1 Fourth Quarter          21       393885
# 2 Fourth Quarter          36       314525
# 3 Fourth Quarter          38       255064


df_season_2017 <- rbind(rbind(rbind(df_violation_first_quarter_2017,
                            df_violation_second_quarter_2017),
                      df_violation_third_quarter_2017),
                df_violation_fourth_quarter_2017)


df_season_2017$Season <- as.factor(df_season_2017$Season)
levels(df_season_2017$Season) <- c("First Quarter", "Second Quarter", "Third Quarter","Fourth Quarter")


plot_season_violation_2017 <- ggplot(df_season_2017,
                             aes(Season , count_tickets, fill = as.factor(`Violation Code`))) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Season for year 2017") +
  ylab("Number of Tickets")
plot_season_violation_2017


###############
##          7: The fines collected from all the parking violation constitute a revenue source for the NYC police department.
##     Let’s take an example of estimating that for the 3 most commonly occurring codes.
##                        1 >> Find total occurrences of the 3 most common violation codes


##                        2 >> Then, search the Internet for NYC parking violation code fines. You will find a website (on the nyc.gov URL)
that lists these fines.
```

They’re divided into two categories, one for the highest-density locations of the city, the other for the rest of the city.

For simplicity, take an average of the two

3 >> Using this information, find the total amount collected for all of the fines. State the code which has the highest total collection.

4 >> What can you intuitively infer from these findings?

##############

##

Find total occurrences of the 3 most common violation codes

##

############## 2015 ##############

```
Cnt_Violation_2015 <-SparkR::sql("SELECT `Violation code`,count(DISTINCT `Summons Number`) as Cnt_Tickets \
                    FROM nycTickets2015_tbl GROUP BY `Violation Code` \
                    Order By count(DISTINCT `Summons Number`) Desc")
df_Cnt_Violation_2015 <- head(Cnt_Violation_2015, n = 3)
df_Cnt_Violation_2015
```

Year 2015

#	Violation code	Cnt_Tickets
#	21	1501614
#	38	1324586
#	14	924627

############## 2016 ##############

```
Cnt_Violation_2016 <-SparkR::sql("SELECT `Violation code`,count(DISTINCT `Summons Number`) as Cnt_Tickets \
                    FROM nycTickets2016_tbl GROUP BY `Violation Code` \
                    Order By count(DISTINCT `Summons Number`) Desc")
df_Cnt_Violation_2016 <- head(Cnt_Violation_2016, n = 3)
df_Cnt_Violation_2016
```

Year 2016

Violation code Cnt_Tickets

```
# 21          1531587
# 36          1253512
# 38          1143696

############### 2017 ###############
Cnt_Violation_2017 <-SparkR::sql("SELECT `Violation code`,count(DISTINCT `Summons Number`) as Cnt_Tickets \
                    FROM nycTickets2017_tbl GROUP BY `Violation Code` \
                    Order By count(DISTINCT `Summons Number`) Desc")
df_Cnt_Violation_2017 <- head(Cnt_Violation_2017, n = 3)
df_Cnt_Violation_2017

# Year 2017
# Violation code Cnt_Tickets
# 21          1528588
# 36          1400614
# 38          1062304

##
## Find the total amount collected for all of the fines. State the code which has the highest total collection.
##

############### 2015 ###############
# By frequency the 3 most commonly occurring Violation Codes are 21, 38 and 14

# From the NYC website we get the below info on Violation Codes fines
# 21        Street Cleaning: No parking where parking is not allowed by sign, street marking or traffic control device.
# 14        General No Standing: Standing or parking where standing is not allowed by sign, street marking or; traffic control device.
# 38        Failing to show a receipt or tag in the windshield. Drivers get a 5-minute grace period past the expired time on Muni-Meter receipts.

# Violation Code   Average Fine
#      21      $55
#      38      $50
#      14      $115

createOrReplaceTempView(Cnt_Violation_2015, "Cnt_Violation_2015_tbl")
```

```
TotalFine_Violation_2015 <- SparkR::sql("SELECT '2015' as Year,`Violation Code`, \
                                  CASE \
                                     WHEN `Violation Code` = 21 THEN (Cnt_Tickets * 55) \
                                     WHEN `Violation Code` = 38 THEN (Cnt_Tickets * 50) \
                                     WHEN `Violation Code` = 14 THEN (Cnt_Tickets * 115) \
                                     ELSE 0 \
                                   END AS Total_Fine \
                                 FROM Cnt_Violation_2015_tbl
                                 WHERE `Violation Code` IN (21, 38, 14)")
df_TotalFine_Violation_2015 <- head(TotalFine_Violation_2015)
df_TotalFine_Violation_2015


#   Year Violation Code Total_Fine
# 1 2015           21  82588770
# 2 2015           38  66229300
# 3 2015           14 106332105


# Violation Code 14 has the highest total collection for year 2015 .. Approx $106 Million


############### 2016 ###############
# By frequency the 3 most commonly occurring Violation Codes are 21, 36 and 38


# From the NYC website we get the below info on Violation Codes fines
# 21        Street Cleaning: No parking where parking is not allowed by sign, street marking or traffic control device.
# 36        Exceeding the posted speed limit in or near a designated school zone.
# 38        Failing to show a receipt or tag in the windshield. Drivers get a 5-minute grace period past the expired time on Muni-Meter receipts.


# Violation Code   Average Fine
#      21       $55
#      36       $50
#      38       $50


createOrReplaceTempView(Cnt_Violation_2016, "Cnt_Violation_2016_tbl")
```

```
TotalFine_Violation_2016 <- SparkR::sql("SELECT '2016' as Year,`Violation Code`, \
                                CASE \
                                   WHEN `Violation Code` = 21 THEN (Cnt_Tickets * 55) \
                                   WHEN `Violation Code` = 36 THEN (Cnt_Tickets * 50) \
                                   WHEN `Violation Code` = 38 THEN (Cnt_Tickets * 50) \
                                   ELSE 0 \
                                END AS Total_Fine \
                               FROM Cnt_Violation_2016_tbl
                               WHERE `Violation Code` IN (21, 36, 38)")
head(TotalFine_Violation_2016)

#   Year Violation Code Total_Fine
# 1 2016           21 84237285
# 2 2016           36 62675600
# 3 2016           38 57184800

# Violation Code 21 has the highest total collection for year 2016 .. Approx $84 Million

############### 2017 ###############
# By frequency the 3 most commonly occurring Violation Codes are 21, 36 and 38

# From the NYC website we get the below info on Violation Codes fines
# 21       Street Cleaning: No parking where parking is not allowed by sign, street marking or traffic control device.
# 36       Exceeding the posted speed limit in or near a designated school zone.
# 38       Failing to show a receipt or tag in the windshield. Drivers get a 5-minute grace period past the expired time on Muni-Meter receipts.

# Violation Code   Average Fine
#      21       $55
#      36       $50
#      38       $50

createOrReplaceTempView(Cnt_Violation_2017, "Cnt_Violation_2017_tbl")

TotalFine_Violation_2017 <- SparkR::sql("SELECT '2017' as Year,`Violation Code`, \
                                CASE \
```

```
                                    WHEN `Violation Code` = 21 THEN (Cnt_Tickets * 55) \
                                    WHEN `Violation Code` = 36 THEN (Cnt_Tickets * 50) \
                                    WHEN `Violation Code` = 38 THEN (Cnt_Tickets * 50) \
                                    ELSE 0 \
                                  END AS Total_Fine \
                                FROM Cnt_Violation_2017_tbl
                                WHERE `Violation Code` IN (21, 36, 38)")

df_TotalFine_Violation_2017 <- head(TotalFine_Violation_2017)
df_TotalFine_Violation_2017

#   Year Violation Code Total_Fine
# 1 2017           21 84072340
# 2 2017           36 70030700
# 3 2017           38 53115200

# Violation Code 21 has the highest total collection for year 2017 .. Approx $84 Million

# Plot for total fine collected across all years
df_toal_fine <- rbind(rbind(df_TotalFine_Violation_2015,df_TotalFine_Violation_2016),df_TotalFine_Violation_2017)
df_code_bins_2017

plot_code_bins_2017 <- ggplot(df_code_bins_2017, aes(as.factor(Year), Total_Fine,fill = as.factor(`Violation Code`))) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous(labels=dollar_format(prefix="$")) +
  xlab("Year") +
  ylab("Total Fine Collected")
plot_code_bins_2017
```