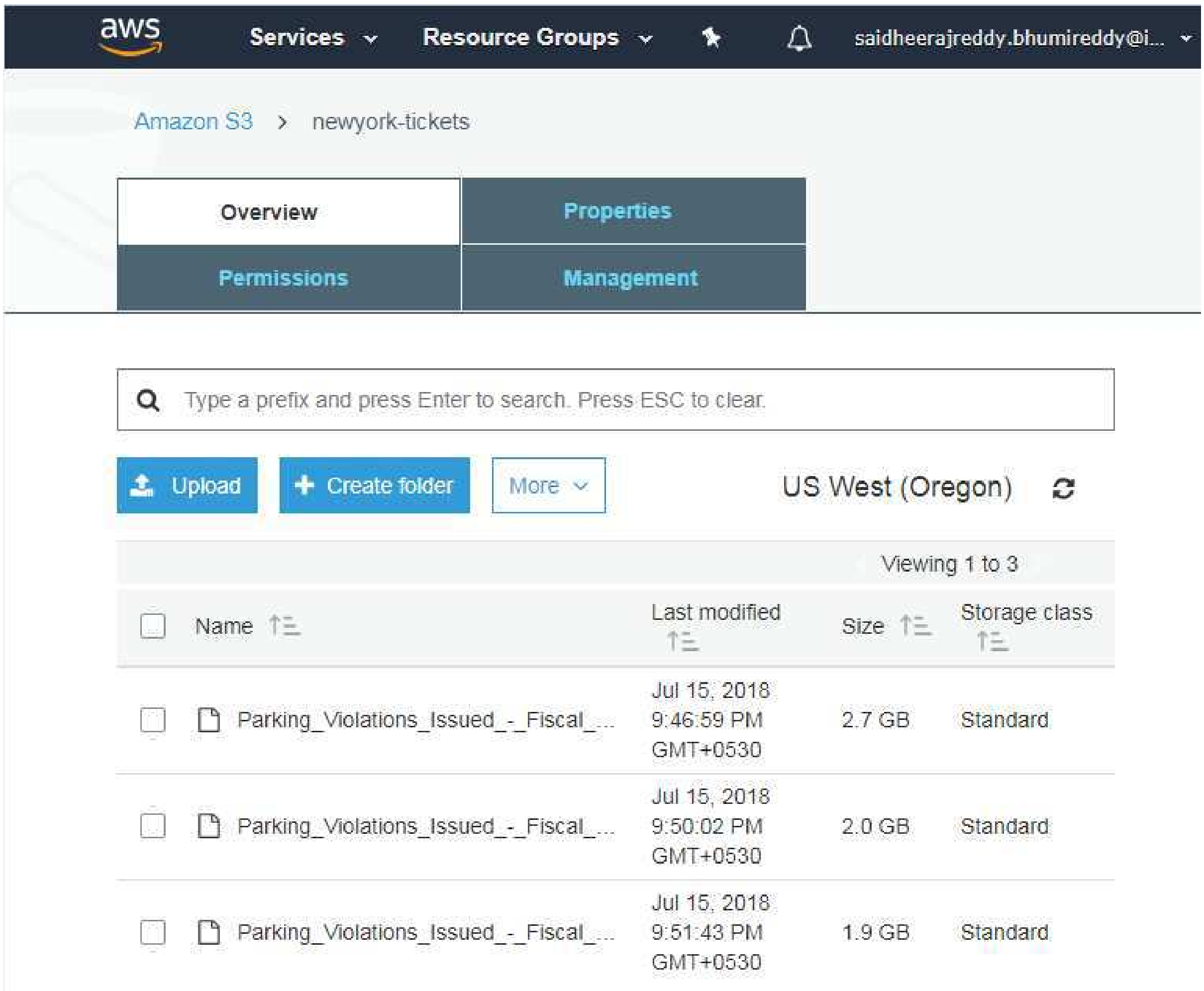
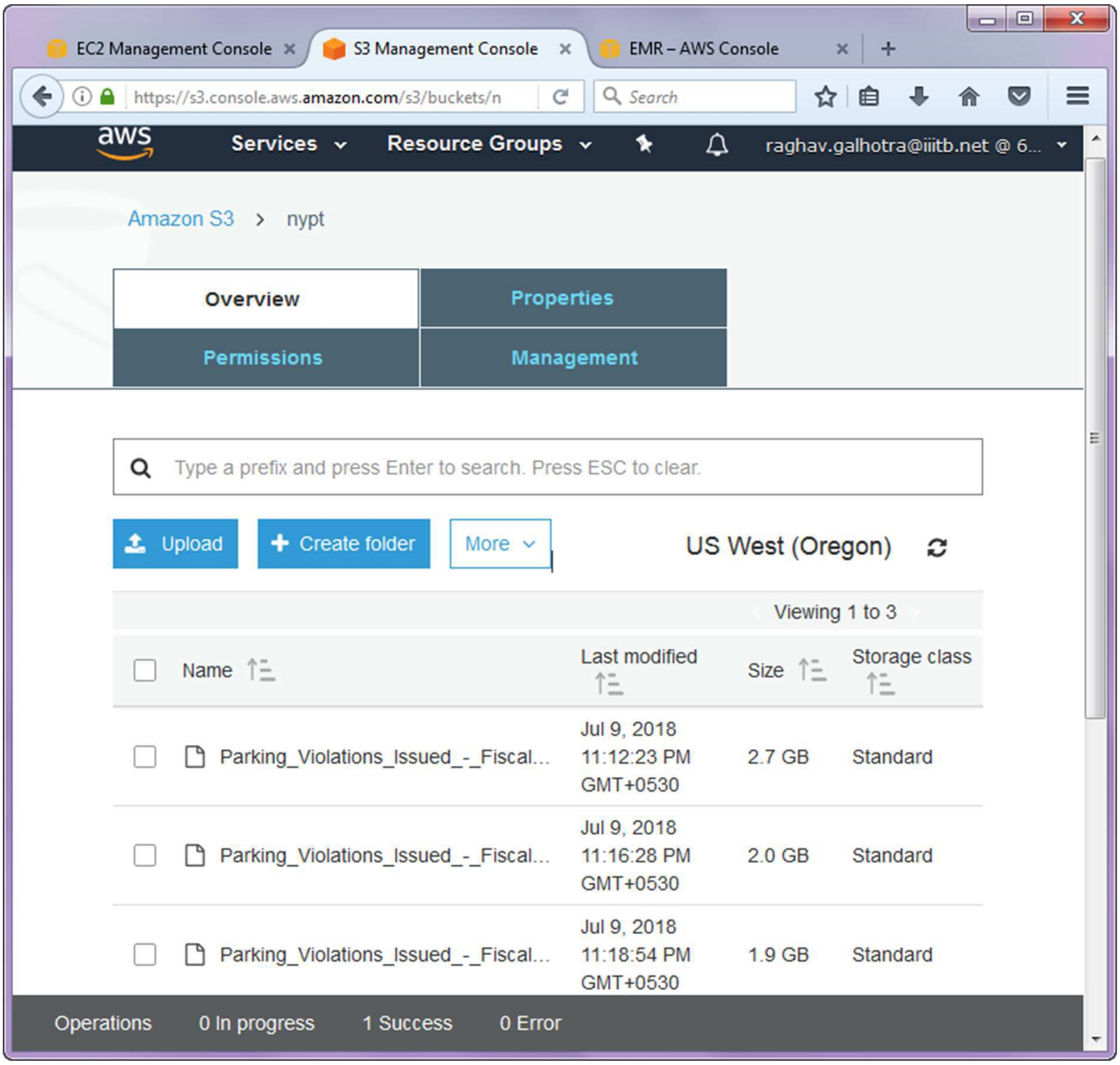
S3 bucket screen shots

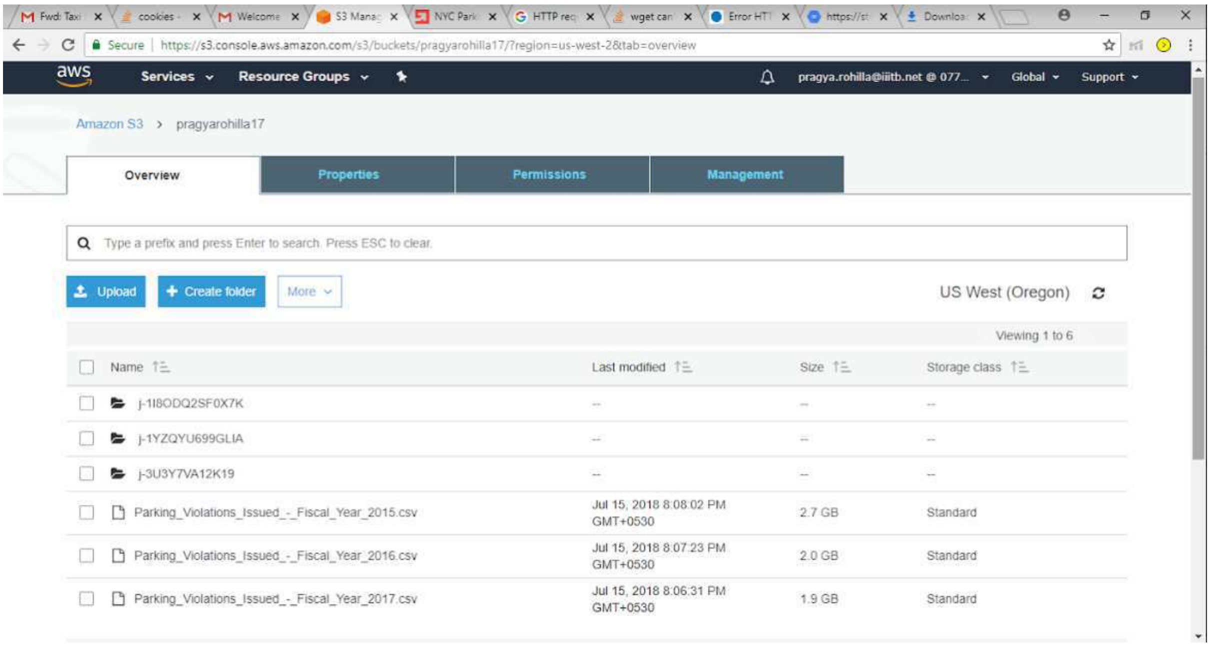

aws
Services
Resource Groups
saidheerajreddy.bhumireddy@i...
Amazon S3 > newyork-tickets
Overview
Properties
Permissions
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
More
US West (Oregon)
Viewing 1 to 3

☐	Name	Last modified	Size	Storage class
☐	Parking_Violations_Issued_-_Fiscal_...	Jul 15, 2018 9:46:59 PM GMT+0530	2.7 GB	Standard
☐	Parking_Violations_Issued_-_Fiscal_...	Jul 15, 2018 9:50:02 PM GMT+0530	2.0 GB	Standard
☐	Parking_Violations_Issued_-_Fiscal_...	Jul 15, 2018 9:51:43 PM GMT+0530	1.9 GB	Standard

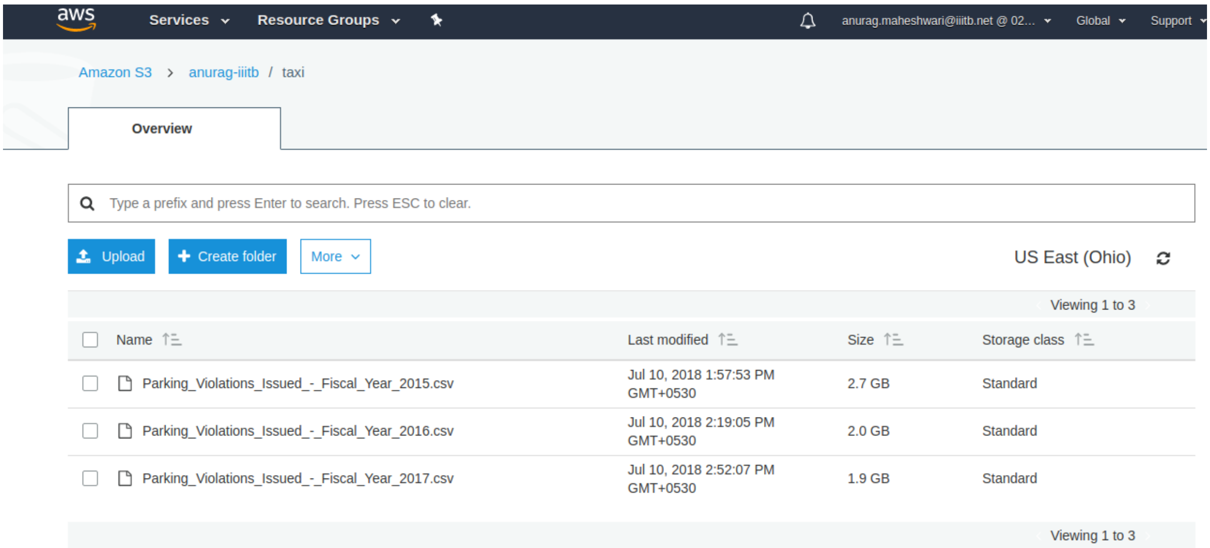


□	Name	Last modified	Size	Storage class
□	Parking_Violations_Issued_-_Fiscal...	Jul 9, 2018 11:12:23 PM GMT+0530	2.7 GB	Standard
□	Parking_Violations_Issued_-_Fiscal...	Jul 9, 2018 11:16:28 PM GMT+0530	2.0 GB	Standard
□	Parking_Violations_Issued_-_Fiscal...	Jul 9, 2018 11:18:54 PM GMT+0530	1.9 GB	Standard

Operations 0 In progress 1 Success 0 Error



□	Name	Last modified	Size	Storage class
□	j-1I8ODQ2SF0X7K	--	--	--
□	j-1YZQYU699GLIA	--	--	--
□	j-3U3Y7VA12K19	--	--	--
□	Parking_Violations_Issued_-_Fiscal_Year_2015.csv	Jul 15, 2018 8:08:02 PM GMT+0530	2.7 GB	Standard
□	Parking_Violations_Issued_-_Fiscal_Year_2016.csv	Jul 15, 2018 8:07:23 PM GMT+0530	2.0 GB	Standard
□	Parking_Violations_Issued_-_Fiscal_Year_2017.csv	Jul 15, 2018 8:06:31 PM GMT+0530	1.9 GB	Standard


aws
Services
Resource Groups
anurag.maheshwari@iiitb.net @ 02...
Global
Support
Amazon S3 › anurag-iiitb / taxi
Overview
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
More
US East (Ohio)
Viewing 1 to 3

□ Name	Last modified	Size	Storage class
□ Parking_Violations_Issued_-_Fiscal_Year_2015.csv	Jul 10, 2018 1:57:53 PM GMT+0530	2.7 GB	Standard
□ Parking_Violations_Issued_-_Fiscal_Year_2016.csv	Jul 10, 2018 2:19:05 PM GMT+0530	2.0 GB	Standard
□ Parking_Violations_Issued_-_Fiscal_Year_2017.csv	Jul 10, 2018 2:52:07 PM GMT+0530	1.9 GB	Standard

```
###########################################################################################################################################
# loading the spark R library and initiating the spark session
library(SparkR)
sparkR.session(master = "local")
###########################################################################################################################################

###########################################################################################################################################
# loading the data into respective SparkDataFrames and also creating their temp tables to run the sql queries

d_2015 <- read.df("s3://nypt/Parking_Violations_Issued_-_Fiscal_Year_2015.csv", source = "csv", inferSchema = "true", header = "true")
d_2016 <- read.df("s3://nypt/Parking_Violations_Issued_-_Fiscal_Year_2016.csv", source = "csv", inferSchema = "true", header = "true")
d_2017 <- read.df("s3://nypt/Parking_Violations_Issued_-_Fiscal_Year_2017.csv", source = "csv", inferSchema = "true", header = "true")
###########################################################################################################################################
head(d_2015)

nrow(d_2015)
#11809233

ncol(d_2015)
#51

str(d_2015)
# $ Summons Number                   : chr "8002531292" "8015318440" "7611181981" "7445908067" "7037692864" "7704791394"
# $ Plate ID                         : chr "EPC5238" "5298MD" "FYW2775" "GWE1987" "T671196C" "JJF6834"
# $ Registration State               : chr "NY" "NY" "NY" "NY" "NY" "PA"
# $ Plate Type                       : chr "PAS" "COM" "PAS" "PAS" "PAS" "PAS"
# $ Issue Date                       : chr "10/01/2014" "03/06/2015" "07/28/2014" "04/13/2015" "05/19/2015" "11/20/2014"
# $ Violation Code                   : chr "21" "14" "46" "19" "19" "21"
# $ Vehicle Body Type                : chr "SUBN" "VAN" "SUBN" "4DSD" "4DSD" "4DSD"
# $ Vehicle Make                     : chr "CHEVR" "FRUEH" "SUBAR" "LEXUS" "CHRYS" "NISSA"
# $ Issuing Agency                   : chr "T" "T" "T" "T" "T" "T"
# $ Street Code1                     : chr "20390" "27790" "8130" "59990" "36090" "74230"
# $ Street Code2                     : chr "29890" "19550" "5430" "16540" "10410" "37980"
# $ Street Code3                     : chr "31490" "19570" "5580" "16790" "24690" "38030"
# $ Vehicle Expiration Date          : chr "01/01/20150111 12:00:00 PM" "01/01/88888888 12:00:00 PM" "01/01/20160524 12:0
# $ Violation Location               : chr "0007" "0025" "0072" "102" "0028" "0067"
# $ Violation Precinct               : chr "7" "25" "72" "102" "28" "67"
# $ Issuer Precinct                  : chr "7" "25" "72" "102" "28" "67"
# $ Issuer Code                      : chr "345454" "333386" "331845" "355669" "341248" "357104"
# $ Issuer Command                   : chr "T800" "T103" "T302" "T402" "T103" "T302"
# $ Issuer Squad                     : chr "A2" "B" "L" "D" "X" "A"
# $ Violation Time                   : chr "0011A" "0942A" "1020A" "0318P" "0410P" "0839A"
# $ Time First Observed              : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Violation County                 : chr "NY" "NY" "K" "Q" "NY" "K"
# $ Violation In Front Of Or Opposite: chr "F" "F" "F" "F" "F" "F"
# $ House Number                     : chr "133" "1916" "184" "120-20" "66" "1013"
# $ Street Name                      : chr "Essex St" "Park Ave" "31st St" "Queens Blvd" "W 116th St" "Rutland Rd"
# $ Intersecting Street              : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Date First Observed              : chr "01/05/0001 12:00:00 PM" "01/05/0001 12:00:00 PM" "01/05/0001 12:00:00 PM" "01
# $ Law Section                      : chr "408" "408" "408" "408" "408" "408"
# $ Sub Division                     : chr "d1" "c" "f1" "c3" "c3" "d1"
# $ Violation Legal Code             : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Days Parking In Effect           : chr "Y Y Y" "YYYYY" "NA" "YYYYY" "YYYYYYY" "Y"
# $ From Hours In Effect             : chr "1200A" "0700A" "NA" "0300P" "NA" "0830A"
# $ To Hours In Effect               : chr "0300A" "1000A" "NA" "1000P" "NA" "0900A"
# $ Vehicle Color                    : chr "BL" "BROWN" "BLACK" "GY" "BLACK" "WHITE"
# $ Unregistered Vehicle?            : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Vehicle Year                     : chr "2005" "0" "2010" "2015" "0" "0"
# $ Meter Number                     : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Feet From Curb                   : chr "0" "0" "0" "0" "0" "0"
# $ Violation Post Code              : chr "A 77" "CC3" "J 32" "01 4" "19 7" "C 32"
# $ Violation Description            : chr "21-No Parking (street clean)" "14-No Standing" "46A-Double Parking (Non-COM)"
# $ No Standing or Stopping Violation: chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Hydrant Violation                : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Double Parking Violation         : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Latitude                         : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Longitude                        : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Community Board                  : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Community Council                : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Census Tract                     : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ BIN                              : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ BBL                              : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ NTA                              : chr "NA" "NA" "NA" "NA" "NA" "NA"


printSchema(d_2015)
 |-- Summons Number: long (nullable = true)
 |-- Plate ID: string (nullable = true)
 |-- Registration State: string (nullable = true)
 |-- Plate Type: string (nullable = true)
 |-- Issue Date: string (nullable = true)
 |-- Violation Code: integer (nullable = true)
 |-- Vehicle Body Type: string (nullable = true)
 |-- Vehicle Make: string (nullable = true)
 |-- Issuing Agency: string (nullable = true)
 |-- Street Code1: integer (nullable = true)
 |-- Street Code2: integer (nullable = true)
 |-- Street Code3: integer (nullable = true)
 |-- Vehicle Expiration Date: string (nullable = true)
 |-- Violation Location: integer (nullable = true)
 |-- Violation Precinct: integer (nullable = true)
 |-- Issuer Precinct: integer (nullable = true)
 |-- Issuer Code: integer (nullable = true)
 |-- Issuer Command: string (nullable = true)
 |-- Issuer Squad: string (nullable = true)
 |-- Violation Time: string (nullable = true)
 |-- Time First Observed: string (nullable = true)
 |-- Violation County: string (nullable = true)
 |-- Violation In Front Of Or Opposite: string (nullable = true)
 |-- House Number: string (nullable = true)
 |-- Street Name: string (nullable = true)
 |-- Intersecting Street: string (nullable = true)
 |-- Date First Observed: string (nullable = true)
 |-- Law Section: integer (nullable = true)
 |-- Sub Division: string (nullable = true)
 |-- Violation Legal Code: string (nullable = true)
 |-- Days Parking In Effect    : string (nullable = true)
 |-- From Hours In Effect: string (nullable = true)
 |-- To Hours In Effect: string (nullable = true)
 |-- Vehicle Color: string (nullable = true)
 |-- Unregistered Vehicle?: integer (nullable = true)
 |-- Vehicle Year: integer (nullable = true)
 |-- Meter Number: string (nullable = true)
 |-- Feet From Curb: integer (nullable = true)
 |-- Violation Post Code: string (nullable = true)
 |-- Violation Description: string (nullable = true)
 |-- No Standing or Stopping Violation: string (nullable = true)
 |-- Hydrant Violation: string (nullable = true)
 |-- Double Parking Violation: string (nullable = true)
 |-- Latitude: string (nullable = true)
 |-- Longitude: string (nullable = true)
 |-- Community Board: string (nullable = true)
 |-- Community Council : string (nullable = true)
 |-- Census Tract: string (nullable = true)
 |-- BIN: string (nullable = true)
 |-- BBL: string (nullable = true)
 |-- NTA: string (nullable = true)

###########################################################################################################################################
createOrReplaceTempView(d_2015, "data_2015")
###########################################################################################################################################
#Following are the columns of interest
#`Summons Number`,`Registration State`,`Issue Date`,`Violation Code`,`Vehicle Body Type`,`Vehicle Make`,`Violation Location`,`Violation Precinct`,
#`Issuer Precinct`,`Issuer Code`,`Violation Time`,`House Number`,`Street Name`,`Law Section`,`Sub Division`,`Days Parking In Effect    `,
#`From Hours In Effect`,`To Hours In Effect`

###########################################################################################################################################
# Filtering the data containing the columns of interest
selected_2015 <- SparkR::sql("select `Summons Number`,`Registration State`,`Issue Date`,`Violation Code`,`Vehicle Body Type`,`Vehicle Make`,`Violation Location`,`Violation Precinct`,`Issuer Precinct`,`Issuer Code`,`Violation Time`,`House Number`,`Street Name`,`Law Section`,`Sub Division`,`Days Parking In Effect    `,`From Hours In Effect`,`To Hours In Effect` from data_2015")
###########################################################################################################################################
createOrReplaceTempView(selected_2015, "data_2015")
###########################################################################################################################################
##Examine the data.

##1. Q1 Find total number of tickets for each year.
ticket_count_2015 <- SparkR::sql("select count(distinct(`Summons Number`)) from data_2015")
head(ticket_count_2015)
#10951256
head(summarize(select(selected_2015,selected_2015$`Summons Number`), count = countDistinct(selected_2015$`Summons Number`)))
#10951256

###########################################################################################################################################
# this suggests that there are some duplicate values present in the Summons Number field
###########################################################################################################################################
###########################################################################################################################################
##2. Q2 Find out how many unique states the cars which got parking tickets came from.
unique_states_2015 <- SparkR::sql("select count(distinct(`Registration State`)) from data_2015")
head(unique_states_2015)
#69 different states does these cars belong to
head(summarize(select(selected_2015,selected_2015$`Registration State`), count = countDistinct(selected_2015$`Registration State`)))
#69
###########################################################################################################################################
###########################################################################################################################################
##3. Q3 Some parking tickets don’t have addresses on them, which is cause for concern. Find out how many such tickets there are.
head(count(where(selected_2015, ((isNull(selected_2015$`House Number`) & isNull(selected_2015$`Street Name`))|((selected_2015$`House Number` == "") & (selected_2015$`Street Name` == ""))))))
#4413 records don't have the valid address in them
###########################################################################################################################################
#Performing some more quality checks and preparing the final filtered dataset for analysis
head(count(where(selected_2015, (isNull(selected_2015$`Summons Number`))|(selected_2015$`Summons Number` == ""))))
#0
```

```
head(count(where(selected_2015, (isNull(selected_2015$`Registration State`))|(selected_2015$`Registration State` == ""))))
#0
head(count(where(selected_2015, (isNull(selected_2015$`Issue Date`))|(selected_2015$`Issue Date` == ""))))
#0
head(count(where(selected_2015, (isNull(selected_2015$`Violation Code`))|(selected_2015$`Violation Code` == ""))))
#0
head(count(where(selected_2015, (isNull(selected_2015$`Vehicle Body Type`))|(selected_2015$`Vehicle Body Type` == ""))))
#45747
head(count(where(selected_2015, (isNull(selected_2015$`Vehicle Make`))|(selected_2015$`Vehicle Make` == ""))))
#75517
head(count(where(selected_2015, (isNull(selected_2015$`Violation Location`))|(selected_2015$`Violation Location` == ""))))
#1799170
head(count(where(selected_2015, (isNull(selected_2015$`Violation Precinct`))|(selected_2015$`Violation Precinct` == ""))))
#0
head(count(where(selected_2015, (isNull(selected_2015$`Issuer Precinct`))|(selected_2015$`Issuer Precinct` == ""))))
#0
head(count(where(selected_2015, (isNull(selected_2015$`Issuer Code`))|(selected_2015$`Issuer Code` == ""))))
#0
head(count(where(selected_2015, (isNull(selected_2015$`Violation Time`))|(selected_2015$`Violation Time` == ""))))
#1715
head(count(where(selected_2015, (isNull(selected_2015$`Days Parking In Effect    `))|(selected_2015$`Days Parking In Effect    ` == ""))))
#2838555
head(count(where(selected_2015, (isNull(selected_2015$`From Hours In Effect`))|(selected_2015$`From Hours In Effect` == ""))))
#5186602
head(count(where(selected_2015, (isNull(selected_2015$`To Hours In Effect`))|(selected_2015$`To Hours In Effect` == ""))))
#5186602
###########################################################################################################################################
# we found that there are 4413 records in the house number and street names which needs to be excluded
# also there are 1715 records on which violation time is not present and those should be excluded as well
# there are some duplicate values in the summons number field, we should remove them as well

selected_2015 <- dropDuplicates(selected_2015, "Summons Number")

selected_2015 <- filter(selected_2015, ((isNotNull(selected_2015$`House Number`) | isNotNull(selected_2015$`Street Name`))|((selected_2015$`House Number` != "") | (selected_2015$`Street Name` != ""))))

selected_2015 <- filter(selected_2015, (isNotNull(selected_2015$`Violation Time`))|(selected_2015$`Violation Time` != ""))
createOrReplaceTempView(selected_2015,"data_2015")
###########################################################################################################################################
###########################################################################################################################################
##Aggregation tasks
##1. Q1 How often does each violation code occur? (frequency of violation codes - find the top 5)
v_code_count_2015 <- summarize(groupBy(selected_2015,selected_2015$`Violation Code`),count = n(selected_2015$`Violation Code`))
head(arrange(v_code_count_2015, desc(v_code_count_2015$count)))

vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from data_2015_1 group by `Violation Code` order by cnt desc limit 5")
head(vio_code_count)
#   Violation Code  cnt
#1            21    1501128
```

```
#2              38   1324529
#3              14   924113
#4              36   761571
#5              37   746229
#6               7   662209
###########################################################################################################################################
# The most common violation code is 21
###########################################################################################################################################
###########################################################################################################################################
##2. Q2 How often does each vehicle body type get a parking ticket? How about the vehicle make? (find the top 5 for both)
vbc_2015 <- summarize(groupBy(selected_2015,selected_2015$`Vehicle Body Type`),count = n(selected_2015$`Vehicle Body Type`))
head(arrange(vbc_2015, desc(vbc_2015$count)))

v_body_count <- SparkR::sql("select `Vehicle Body Type`, count(*) as cnt from data_2015 group by `Vehicle Body Type` order by cnt desc limit 5")
head(v_body_count)

#Vehicle Body Type      Count
#1           SUBN        3450976
#2            4DSD       3102383
#3            VAN        1604777
#4            DELV       840097
#5            SDN        452714
#6            2DSD       296919

vmc_2015 <- summarize(groupBy(selected_2015,selected_2015$`Vehicle Make`),count = n(selected_2015$`Vehicle Make`))
head(arrange(vmc_2015, desc(vmc_2015$count)))

v_make_count <- SparkR::sql("select `Vehicle Make`, count(*) as cnt from data_2015 group by `Vehicle Make` order by cnt desc limit 5")
head(v_body_count)

#   Vehicle Make    count
#1        FORD      1416869
#2        TOYOT     1123165
#3        HONDA     1017711
#4        NISSA     837301
#5        CHEVR     836165
#6        FRUEH     408150
###########################################################################################################################################
###########################################################################################################################################
##3. Q3 A precinct is a police station that has a certain zone of the city under its command. Find the (5 highest) frequencies of:
#a. Violating Precincts (this is the precinct of the zone where the violation occurred)
vio_pre_2015 <- summarize(groupBy(selected_2015,selected_2015$`Violation Precinct`),count = n(selected_2015$`Violation Precinct`))
head(arrange(vio_pre_2015, desc(vio_pre_2015$count)))

vio_pre_sql_2015 <- SparkR::sql("select `Violation Precinct`, count(*) as cnt from data_2015 group by `Violation Precinct` order by cnt desc limit 5")
head(vio_pre_sql_2015)
#Violation Precinct     count
#1                  0   1630789
#2                 19   559682
#3                 18   400845
#4                 14   384563
#5                  1   307766
#6                114   300519

b. Issuing Precincts (this is the precinct that issued the ticket)
iss_pre_2015 <- summarize(groupBy(selected_2015,selected_2015$`Issuer Precinct`),count = n(selected_2015$`Issuer Precinct`))
head(arrange(iss_pre_2015, desc(iss_pre_2015$count)))

```

```
iss_pre_sql_2015 <- SparkR::sql("select `Issuer Precinct`, count(*) as cnt from data_2015 group by `Issuer Precinct` order by cnt desc limit 5")
head(iss_pre_sql_2015)

#Issuer Precinct    count
#1              0    1831810
#2             19    544924
#3             18    391464
#4             14    369692
#5              1    298562
#6            114    295574

##4. Q4 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?

vio_pre_code_2015 <- SparkR::sql("select `Issuer Precinct`,`Violation Code`, count(*) as cnt from data_2015 group by `Issuer Precinct`,`Violation Code` order by cnt desc limit 10")
head(vio_pre_code_2015)

#   Issuer Precinct Violation Code      cnt
#1                0             36      683945
#2                0              7      604657
#3                0              5      166188
#4                0             21      156417
#5               18             14      112977
#6               19             38      83720

##5. You'd want to find out the properties of parking violations across different times of the day:
##a. 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.

#Violation Time  : chr "0011A" "0942A" "1020A" "0318P" "0410P" "0839A"
#From summary of data we see that Violation time is stored as characters having alphabets A and P denoting AM and PM probably



##b. Find a way to deal with missing values, if any.

selected_2015$hr <- substr(selected_2015$`Violation Time`, 1, 2)
selected_2015$ampm <- substr(selected_2015$`Violation Time`, 6, 6)

selected_2015$vt_bin <- ifelse(selected_2015$hr != 12 & selected_2015$ampm == "P", selected_2015$hr + 12, selected_2015$hr)



createOrReplaceTempView(selected_2015,"data_2015")

##c. Divide 24 hours into 6 equal discrete bins of time. The intervals you choose are at your discretion.
selected_2015_hr <- SparkR::sql("select `Violation Code`, \
                   CASE  WHEN (vt_bin >= 4  and vt_bin < 8)  THEN 'early_morning'\
                   WHEN (vt_bin >= 8  and vt_bin < 12) THEN 'morning'\
                   WHEN (vt_bin >= 12 and vt_bin < 16) THEN 'after_noon'\
                   WHEN (vt_bin >= 16 and vt_bin < 20) THEN 'evening'\
                   WHEN (vt_bin >= 20 and vt_bin < 24) THEN 'night'\
                   ELSE 'late_night' END  as time_group FROM data_2015")
createOrReplaceTempView(selected_2015_hr,"selected_2015_hr")


##For each of these groups, find the 3 most commonly occurring violations
# For early_morning
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr where time_group = 'early_morning' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code    cnt
#14            134335
#21            106782
#40             91336
```

```

# For Morning
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr where time_group = 'morning' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code     cnt
#21             1191837
#38              449046
#36              360365

# For after_noon
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr where time_group = 'after_noon' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code    cnt
#38             568324
#37             417605
#36             323526

# For evening
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr where time_group = 'evening' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code    cnt
#38             241317
#37             175785
#7              168888

# For night
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr where time_group = 'night' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code   cnt
#7              81981
#38             62414
#14             45791

# For late_night
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr where time_group = 'late_night' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code   cnt
#21             63571
#40             36485
#78             34806


most_vio_2015 <- SparkR::sql("select time_group,`Violation Code`,count(*) as cnt from selected_2015_hr group by time_group,`Violation Code` order by cnt desc")

head(most_vio_2015)

##d. 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)
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr group by `Violation Code` order by cnt desc limit 3"))
#Violation Code     cnt
#21             1501128
#38             1324529
#14              924113
#Three most common Violation Code are 21, 38 and 14
#The most common times of the day for these codes
head(SparkR::sql("Select time_group, count(*) as cnt from selected_2015_hr where `Violation Code` IN (21,38,14) group by time_group order by cnt desc"))
##  time_group     cnt
#      morning  1938451
#    after_noon  977239
#       evening  390627
# early_morning  243967
#         night  108792
#    late_night   90694

##6. Let’s try and find some seasonality in this data
##a. First, divide the year into some number of seasons, and find frequencies of tickets for each season.
seasons_2015 <- SparkR::sql("select `Violation Code`, \
                   CASE  WHEN (month(`Issue Date`) >= 1  and month(`Issue Date`) <= 3)  THEN 'Q1'\
                   WHEN (month(`Issue Date`) >= 4  and month(`Issue Date`) <= 6) THEN 'Q2'\
                   WHEN (month(`Issue Date`) >= 7 and month(`Issue Date`) <= 9) THEN 'Q3'\
                   ELSE 'Q4' END  as season FROM data_2015")

createOrReplaceTempView(seasons_2015,"seasons_2015")
##b. Then, find the 3 most common violations for each of these season
vio_seas_2015 <- SparkR::sql("select season,`Violation Code`,count(*) as cnt from seasons_2015 group by season,`Violation Code` order by cnt desc")
head(vio_seas_2015)
#   season Violation Code   cnt
#1     Q4             21    1501128
#2     Q4             38    1324529
#3     Q4             14     924113
#4     Q4             36     761571
#5     Q4             37     746229
#6     Q4              7     662209

##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.
##a. Find total occurrences of the 3 most common violation codes
vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from data_2015 group by `Violation Code` order by cnt desc limit 5")
head(vio_code_count)
#Code   Count
#21     1382405
#38     1231003
#14     865822
#37     696895
#36     683945

##b. 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.
#Code   Average
#21     55
#38     50
#14     115
#37     50
#36     50

##c. Using this information, find the total amount collected for all of the fines. State the code which has the highest total collection.
#Code   Total Collection
#21     76032275
#38     61550150
#14     129873300
#37     34844750
#36     34197250

##d. What can you intuitively infer from these findings?
# For 2015, we can infer that although Code #21 and #38 sees highest number of fines but the average cost is highest for code #14 which draws the highest collection among aall the fines.


##########################################################################################################################################################
#Analyses for 2016 data
##########################################################################################################################################################
head(d_2016)

nrow(d_2016)
#10626899

ncol(d_2016)
#51

str(d_2016)
```

```
# $ Summons Number                   : num 1363745270 1363745293 1363745438 1363745475 1363745487 1363745517
# $ Plate ID                         : chr "GGY6450" "KXD355" "JCK7576" "GYK7658" "GMT8141" "GYK3760"
# $ Registration State               : chr "99" "SC" "PA" "NY" "NY" "NY"
# $ Plate Type                       : chr "PAS" "PAS" "PAS" "OMS" "PAS" "PAS"
# $ Issue Date                       : chr "07/09/2016" "07/09/2016" "07/09/2016" "07/09/2016" "07/09/2016" "07/09/2016"
# $ Violation Code                   : int 46 21 21 21 21 21
# $ Vehicle Body Type                : chr "SDN" "SUBN" "SDN" "SUBN" "P-U" "SUBN"
# $ Vehicle Make                     : chr "HONDA" "CHEVR" "ME/BE" "NISSA" "LINCO" "HONDA"
# $ Issuing Agency                   : chr "P" "P" "P" "P" "P" "P"
# $ Street Code1                     : int 0 55730 42730 58130 58130 46730
# $ Street Code2                     : int 40404 67030 26730 18630 67030 58730
# $ Street Code3                     : int 40404 58730 26830 67030 58730 85730
# $ Vehicle Expiration Date          : int 20170602 20160288 0 0 20160206 20160709
# $ Violation Location               : int 74 79 79 79 79 79
# $ Violation Precinct               : int 74 79 79 79 79 79
# $ Issuer Precinct                  : int 301 301 0 301 301 301
# $ Issuer Code                      : int 358160 358160 358114 358114 358114 358114
# $ Issuer Command                   : chr "T301" "T301" "TEBN" "T301" "T301" "T301"
# $ Issuer Squad                     : chr "0000" "0000" "0000" "0000" "0000" "0000"
# $ Violation Time                   : chr "1037A" "1206P" "0820A" "0918A" "0925A" "0948A"
# $ Time First Observed              : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Violation County                 : chr "K" "K" "K" "K" "K" "K"
# $ Violation In Front Of Or Opposite: chr "F" "F" "F" "F" "F" "F"
# $ House Number                     : chr "142" "331" "1087" "207" "237" "201"
# $ Street Name                      : chr "MACDOUNGH ST" "LEXINGTON AVE" "FULTON ST" "MADISON ST" "MADISON ST" "HALSEY S
# $ Intersecting Street              : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Date First Observed              : int 0 0 0 0 0 0
# $ Law Section                      : int 408 408 408 408 408 408
# $ Sub Division                     : chr "D1" "F1" "D1" "D1" "D1" "D1"
# $ Violation Legal Code             : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Days Parking In Effect           : chr "BBBBBBB" "YBBYBBB" "YBBYBBB" "YBBYBBB" "YBBYBBB" "YBBYBBB"
# $ From Hours In Effect             : chr "ALL" "1100A" "0800A" "0900A" "0900A" "0900A"
# $ To Hours In Effect               : chr "ALL" "1230P" "0930A" "1030" "1030A" "1030A"
# $ Vehicle Color                    : chr "WHITE" "RED" "WHITE" "BK" "BLK" "OTHER"
# $ Unregistered Vehicle?            : chr "0" "0" "0" "0" "0" "0"
# $ Vehicle Year                     : int 2010 0 0 2016 2006 2006
# $ Meter Number                     : chr "-" "-" "-" "-" "-" "-"
# $ Feet From Curb                   : int 0 0 0 0 0 0
# $ Violation Post Code              : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Violation Description            : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ No Standing or Stopping Violation: chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Hydrant Violation                : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Double Parking Violation         : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Latitude                         : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Longitude                        : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Community Board                  : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Community Council                : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ Census Tract                     : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ BIN                              : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ BBL                              : chr "NA" "NA" "NA" "NA" "NA" "NA"
# $ NTA                              : chr "NA" "NA" "NA" "NA" "NA" "NA"


printSchema(d_2016)
 |-- Summons Number: long (nullable = true)
 |-- Plate ID: string (nullable = true)
 |-- Registration State: string (nullable = true)
 |-- Plate Type: string (nullable = true)
 |-- Issue Date: string (nullable = true)
 |-- Violation Code: integer (nullable = true)
 |-- Vehicle Body Type: string (nullable = true)
 |-- Vehicle Make: string (nullable = true)
 |-- Issuing Agency: string (nullable = true)
 |-- Street Code1: integer (nullable = true)
 |-- Street Code2: integer (nullable = true)
 |-- Street Code3: integer (nullable = true)
 |-- Vehicle Expiration Date: integer (nullable = true)
 |-- Violation Location: integer (nullable = true)
 |-- Violation Precinct: integer (nullable = true)
 |-- Issuer Precinct: integer (nullable = true)
 |-- Issuer Code: integer (nullable = true)
 |-- Issuer Command: string (nullable = true)
 |-- Issuer Squad: string (nullable = true)
 |-- Violation Time: string (nullable = true)
 |-- Time First Observed: string (nullable = true)
 |-- Violation County: string (nullable = true)
 |-- Violation In Front Of Or Opposite: string (nullable = true)
 |-- House Number: string (nullable = true)
 |-- Street Name: string (nullable = true)
 |-- Intersecting Street: string (nullable = true)
 |-- Date First Observed: integer (nullable = true)
 |-- Law Section: integer (nullable = true)
 |-- Sub Division: string (nullable = true)
 |-- Violation Legal Code: string (nullable = true)
 |-- Days Parking In Effect    : string (nullable = true)
 |-- From Hours In Effect: string (nullable = true)
 |-- To Hours In Effect: string (nullable = true)
 |-- Vehicle Color: string (nullable = true)
 |-- Unregistered Vehicle?: string (nullable = true)
 |-- Vehicle Year: integer (nullable = true)
 |-- Meter Number: string (nullable = true)
 |-- Feet From Curb: integer (nullable = true)
 |-- Violation Post Code: string (nullable = true)
 |-- Violation Description: string (nullable = true)
 |-- No Standing or Stopping Violation: string (nullable = true)
 |-- Hydrant Violation: string (nullable = true)
 |-- Double Parking Violation: string (nullable = true)
 |-- Latitude: string (nullable = true)
 |-- Longitude: string (nullable = true)
 |-- Community Board: string (nullable = true)
 |-- Community Council : string (nullable = true)
 |-- Census Tract: string (nullable = true)
 |-- BIN: string (nullable = true)
 |-- BBL: string (nullable = true)
 |-- NTA: string (nullable = true)

############################################################################################################################################
createOrReplaceTempView(d_2016, "data_2016")
############################################################################################################################################
#Following are the columns of interest
#`Summons Number`,`Registration State`,`Issue Date`,`Violation Code`,`Vehicle Body Type`,`Vehicle Make`,`Violation Location`,`Violation Precinct`,
#`Issuer Precinct`,`Issuer Code`,`Violation Time`,`House Number`,`Street Name`,`Law Section`,`Sub Division`,`Days Parking In Effect    `,
#`From Hours In Effect`,`To Hours In Effect`

############################################################################################################################################
# Filtering the data containing the columns of interest
selected_2016 <- SparkR::sql("select `Summons Number`,`Registration State`,`Issue Date`,`Violation Code`,`Vehicle Body Type`,`Vehicle Make`,`Violation Location`,`Violation Precinct`,`Issuer Precinct`,`Issuer Code`,`Violation Time`,`House Number`,`Street Name`,`Law Section`,`Sub Division`,`Days Parking In Effect    `,`From Hours In Effect`,`To Hours In Effect` from data_2016")
############################################################################################################################################
createOrReplaceTempView(selected_2016, "data_2016")
############################################################################################################################################
##Examine the data.

##1. Q1 Find total number of tickets for each year.
ticket_count_2016 <- SparkR::sql("select count(distinct(`Summons Number`)) from data_2016")
```

```
head(ticket_count_2016)
#10626899
head(summarize(select(selected_2016,selected_2016$`Summons Number`), count = countDistinct(selected_2016$`Summons Number`)))
#10626899

#############################################################################################################################################
# this suggests that there are some duplicate values present in the Summons Number field
#############################################################################################################################################
#############################################################################################################################################
##2. Q2 Find out how many unique states the cars which got parking tickets came from.
unique_states_2016 <- SparkR::sql("select count(distinct(`Registration State`)) from data_2016")
head(unique_states_2016)
#68 different states does these cars belong to
head(summarize(select(selected_2016,selected_2016$`Registration State`), count = countDistinct(selected_2016$`Registration State`)))
#68
#############################################################################################################################################
#############################################################################################################################################
##3. Q3 Some parking tickets don’t have addresses on them, which is cause for concern. Find out how many such tickets there are.
head(count(where(selected_2016, ((isNull(selected_2016$`House Number`) & isNull(selected_2016$`Street Name`))|((selected_2016$`House Number` == "") & (selected_2016$`Street Name` == ""))))))
#6462 records don't have the valid address in them
#############################################################################################################################################
#Performing some more quality checks and preparing the final filtered dataset for analysis
head(count(where(selected_2016, (isNull(selected_2016$`Summons Number`))|(selected_2016$`Summons Number` == ""))))
#0
head(count(where(selected_2016, (isNull(selected_2016$`Registration State`))|(selected_2016$`Registration State` == ""))))
#0
head(count(where(selected_2016, (isNull(selected_2016$`Issue Date`))|(selected_2016$`Issue Date` == ""))))
#0
head(count(where(selected_2016, (isNull(selected_2016$`Violation Code`))|(selected_2016$`Violation Code` == ""))))
#0
head(count(where(selected_2016, (isNull(selected_2016$`Vehicle Body Type`))|(selected_2016$`Vehicle Body Type` == ""))))
#39277
head(count(where(selected_2016, (isNull(selected_2016$`Vehicle Make`))|(selected_2016$`Vehicle Make` == ""))))
#63583
head(count(where(selected_2016, (isNull(selected_2016$`Violation Location`))|(selected_2016$`Violation Location` == ""))))
#1868656
head(count(where(selected_2016, (isNull(selected_2016$`Violation Precinct`))|(selected_2016$`Violation Precinct` == ""))))
#1
head(count(where(selected_2016, (isNull(selected_2016$`Issuer Precinct`))|(selected_2016$`Issuer Precinct` == ""))))
#1
head(count(where(selected_2016, (isNull(selected_2016$`Issuer Code`))|(selected_2016$`Issuer Code` == ""))))
#1
head(count(where(selected_2016, (isNull(selected_2016$`Violation Time`))|(selected_2016$`Violation Time` == ""))))
#4280
head(count(where(selected_2016, (isNull(selected_2016$`Days Parking In Effect `))|(selected_2016$`Days Parking In Effect    ` == ""))))
#2867416
head(count(where(selected_2016, (isNull(selected_2016$`From Hours In
Effect`))|(selected_2016$`From Hours In Effect` == ""))))
#4976147
head(count(where(selected_2016, (isNull(selected_2016$`To Hours In Effect`))|(selected_2016$`To Hours In Effect` == ""))))
#4976147
#############################################################################################################################################
# we found that there are 6462 records in the house number and street names which needs to be excluded
# also there are 4280 records on which violation time is not present and those should be excluded as well
# there are some duplicate values in the summons number field, we should remove them as well

selected_2016 <- dropDuplicates(selected_2016, "Summons Number")

selected_2016 <- filter(selected_2016, ((isNotNull(selected_2016$`House Number`) | isNotNull(selected_2016$`Street Name`))|((selected_2016$`House Number` != "") | (selected_2016$`Street Name` != ""))))

selected_2016 <- filter(selected_2016, (isNotNull(selected_2016$`Violation Time`))|(selected_2016$`Violation Time` != ""))
createOrReplaceTempView(selected_2016,"data_2016")
#############################################################################################################################################
#############################################################################################################################################
##Aggregation tasks
##1. Q1 How often does each violation code occur? (frequency of violation codes - find the top 5)
v_code_count_2016 <- summarize(groupBy(selected_2016,selected_2016$`Violation Code`),count = n(selected_2016$`Violation Code`))
head(arrange(v_code_count_2016, desc(v_code_count_2016$count)))

#####vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from data_2016_1 group by `Violation Code` order by cnt desc limit 5")
#####head(vio_code_count)
#   Violation Code  cnt
#1             21 1530427
#2              36 1253511
#3              38 1143394
#4              14  874901
#5              37  686460
#6              20  610599
#############################################################################################################################################
# The most common violation code is 21
#############################################################################################################################################
#############################################################################################################################################
##2. Q2 How often does each vehicle body type get a parking ticket? How about the vehicle make? (find the top 5 for both)
vbc_2016 <- summarize(groupBy(selected_2016,selected_2016$`Vehicle Body Type`),count = n(selected_2016$`Vehicle Body Type`))
head(arrange(vbc_2016, desc(vbc_2016$count)))

v_body_count <- SparkR::sql("select `Vehicle Body Type`, count(*) as cnt from data_2016 group by `Vehicle Body Type` order by cnt desc limit 5")
head(v_body_count)

#Vehicle Body Type      Count
#1          SUBN        3450976
#2           4DSD       3102383
#3           VAN        1604777
#4           DELV       840097
#5           SDN        452714
#6           2DSD       296919

vmc_2016 <- summarize(groupBy(selected_2016,selected_2016$`Vehicle Make`),count = n(selected_2016$`Vehicle Make`))
head(arrange(vmc_2016, desc(vmc_2016$count)))

```

```
v_make_count <- SparkR::sql("select `Vehicle Make`, count(*) as cnt from data_2016 group by `Vehicle Make` order by cnt desc limit 5")
head(v_body_count)

#Vehicle Body Type   count
#1              SUBN 3463919
#2              4DSD 2991385
#3               VAN 1517704
#4              DELV  754966
#5               SDN  422240
#6              2DSD  276375
###########################################################################################################################################
###########################################################################################################################################
##3. Q3 A precinct is a police station that has a certain zone of the city under its command. Find the (5 highest) frequencies of:
#a. Violating Precincts (this is the precinct of the zone where the violation occurred)
vio_pre_2016 <- summarize(groupBy(selected_2016,selected_2016$`Violation Precinct`),count = n(selected_2016$`Violation Precinct`))
head(arrange(vio_pre_2016, desc(vio_pre_2016$count)))

vio_pre_sql_2016 <- SparkR::sql("select `Violation Precinct`, count(*) as cnt from data_2016 group by `Violation Precinct` order by cnt desc limit 5")
head(vio_pre_sql_2016)
# Violation Precinct   count
#1                   0 1867301
#2                  19  554325
#3                  18  331615
#4                  14  324389
#5                   1  303745
#6                 114  291235

b. Issuing Precincts (this is the precinct that issued the ticket)
iss_pre_2016 <- summarize(groupBy(selected_2016,selected_2016$`Issuer Precinct`),count = n(selected_2016$`Issuer Precinct`))
head(arrange(iss_pre_2016, desc(iss_pre_2016$count)))

iss_pre_sql_2016 <- SparkR::sql("select `Issuer Precinct`, count(*) as cnt from data_2016 group by `Issuer Precinct` order by cnt desc limit 5")
head(iss_pre_sql_2016)
# Issuer Precinct   count
#1               0 2138264
#2              19  540458
#3              18  323058
#4              14  315241
#5               1  294899
#6             114  286835

##4. Q4 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?
df1 <- groupBy(selected_2016, selected_2016$`Violation Code`)
df2 <- agg(df1, precinct = n_distinct(selected_2016$`Issuer Precinct`), count = n(selected_2016$`Violation Code`))
head(df2)

vio_pre_code_2016 <- SparkR::sql("select `Issuer Precinct`,`Violation Code`, count(*) as cnt from data_2016 group by `Issuer Precinct`,`Violation Code` order by cnt desc limit 10")
head(vio_pre_code_2016)

#  Violation Code precinct   count
#1             31       86 139082
#2             85       96  27921
#3             65       48    126
#4             53      118  31676
#5             78      159  60532
#6             34       17     32

##5. You'd want to find out the properties of parking violations across different times of the day:
```

```
##a. 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.

#Violation Time  : chr "0011A" "0942A" "1020A" "0318P" "0410P" "0839A"
#From summary of data we see that Violation time is stored as characters having alphabets A and P denoting AM and PM probably



##b. Find a way to deal with missing values, if any.

selected_2016$hr <- substr(selected_2016$`Violation Time`, 1, 2)
selected_2016$ampm <- substr(selected_2016$`Violation Time`, 6, 6)

selected_2016$vt_bin <- ifelse(selected_2016$hr != 12 & selected_2016$ampm == "P", selected_2016$hr + 12, selected_2016$hr)


#selected_2016_hr <- dapplyCollect (
#df1,
#function(x) {
#hr <- as.numeric(substr(x$`Violation Time`, 1, 2))
#ampm <- substr(x$`Violation Time`, 5, 5)
#x <- cbind(x, "hr" = hr)
#x <- cbind(x, "ampm" = ampm)
#})

#selected_2016_hr$hr <- ifelse(selected_2016_hr$hr != 12 & selected_2016_hr$ampm == "P", selected_2016_hr$hr + 12, selected_2016_hr$hr)
#selected_2016_hr$`Violation Time` <- NULL
#selected_2016_hr$ampm <- NULL

createOrReplaceTempView(selected_2016,"data_2016")

##c. Divide 24 hours into 6 equal discrete bins of time. The intervals you choose are at your discretion.
selected_2016_hr <- SparkR::sql("select `Violation Code`, \
                   CASE  WHEN (vt_bin >= 4  and vt_bin < 8)  THEN 'early_morning'\
                   WHEN (vt_bin >= 8  and vt_bin < 12) THEN 'morning'\
                   WHEN (vt_bin >= 12 and vt_bin < 16) THEN 'after_noon'\
                   WHEN (vt_bin >= 16 and vt_bin < 20) THEN 'evening'\
                   WHEN (vt_bin >= 20 and vt_bin < 24) THEN 'night'\
                   ELSE 'late_night' END  as time_group FROM data_2016")
createOrReplaceTempView(selected_2016_hr,"selected_2016_hr")


##For each of these groups, find the 3 most commonly occurring violations
# For early_morning
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr where time_group = 'early_morning' group by `Violation Code` order by cnt desc limit 3"))
# Violation Code   cnt
#1             14 140033
#2             21 113985
#3             40  91680

# For Morning
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr where time_group = 'morning' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code     cnt
#1             21 1209001
#2             36  586791
#3             38  388080

# For after_noon
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr where time_group = 'after_noon' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code    cnt
#38             568324
#37             417605
#36             323526

# For evening
```

```
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr where time_group = 'evening' group by `Violation Code` order by cnt desc limit 3"))
#  Violation Code    cnt
#1             38 211262
#2             37 161647
#3             14 134917

# For night
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr where time_group = 'night' group by `Violation Code` order by cnt desc limit 3"))
# Violation Code   cnt
#1              7 60924
#2             38 53173
#3             40 44952

# For late_night
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr where time_group = 'late_night' group by `Violation Code` order by cnt desc limit 3"))
# Violation Code   cnt
#1             21 67818
#2             40 37256
#3             78 29451


most_vio_2016 <- SparkR::sql("select time_group,`Violation Code`,count(*) as cnt from selected_2016_hr group by time_group,`Violation Code` order by cnt desc")

head(most_vio_2016)
# time_group Violation Code     cnt
#1    morning             21 1209001
#2    morning             36  586791
#3 after_noon             36  545717
#4 after_noon             38  488347
#5    morning             38  388080
#6 after_noon             37  383352

##d. 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)
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr group by `Violation Code` order by cnt desc limit 3"))
#Violation Code     cnt
#1             21 1530427
#2             36 1253511
#3             38 1143394
#Three most common Violation Code are 21, 38 and 14
#The most common times of the day for these codes
head(SparkR::sql("Select time_group, count(*) as cnt from selected_2016_hr where `Violation Code` IN (21,38,14) group by time_group order by cnt desc"))
 time_group     cnt
1       morning 1873228
2    after_noon  877934
3       evening  346780
4 early_morning  256227
5         night   97801
6    late_night   96752

##6. Let’s try and find some seasonality in this data
##a. First, divide the year into some number of seasons, and find frequencies of tickets for each season.
`Issue Date`
seasons_2016 <- SparkR::sql("select `Violation Code`, \
                   CASE  WHEN (month(`Issue Date`) >= 1  and month(`Issue Date`) <= 3)  THEN 'Q1'\
                   WHEN (month(`Issue Date`) >= 4  and month(`Issue Date`) <= 6) THEN 'Q2'\
                   WHEN (month(`Issue Date`) >= 7 and month(`Issue Date`) <= 9) THEN 'Q3'\
                   ELSE 'Q4' END  as season FROM data_2016")

##b. Then, find the 3 most common violations for each of these season
createOrReplaceTempView(seasons_2016,"seasons_2016")
vio_seas_2016 <- SparkR::sql("select season,`Violation Code`,count(*) as cnt from seasons_2016 group by season,`Violation Code` order by cnt desc")
```

```
head(vio_seas_2016)
#  season Violation Code     cnt
#1     Q4             21 1530427
#2     Q4             36 1253511
#3     Q4             38 1143394
#4     Q4             14  874901
#5     Q4             37  686460
#6     Q4             20  610599
##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.
##a. Find total occurrences of the 3 most common violation codes
vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from data_2016 group by `Violation Code` order by cnt desc limit 5")
head(vio_code_count)
# Violation Code     cnt
#1             21 1530427
#2             36 1253511
#3             38 1143394
#4             14  874901
#5             37  686460

##b. 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.
#Code   Average
#21     55
#36     50
#38     50
#14     115
#37     50

##c. Using this information, find the total amount collected for all of the fines. State the code which has the highest total collection.
#Code   Total Collection
#21     84173485
#36     62675550
#38     57169700
#14     100613615
#37     34323000

##d. What can you intuitively infer from these findings?
# For 2016, we can infer that although Code #21 and #36 sees highest number of fines but the average cost is highest for code #14 which draws the highest collection among aall the fines.
##################################################################################################################################################
#Analyses for 2017 data
##################################################################################################################################################
head(d_2017)

nrow(d_2017)
#10803028
ncol(d_2017)
#43

str(d_2017)
#$ Summons Number                    : num 5092469481 5092451658 4006265037 8478629828 7868300310 5096917368
#$ Plate ID                          : chr "GZH7067" "GZH7067" "FZX9232" "66623ME" "37033JV" "FZD8593"
#$ Registration State                : chr "NY" "NY" "NY" "NY" "NY" "NY"
#$ Plate Type                        : chr "PAS" "PAS" "PAS" "COM" "COM" "PAS"
#$ Issue Date                        : chr "07/10/2016" "07/08/2016" "08/23/2016" "06/14/2017" "11/21/2016" "06/13/2017"
#$ Violation Code                    : int 7 7 5 47 69 7
#$ Vehicle Body Type                 : chr "SUBN" "SUBN" "SUBN" "REFG" "DELV" "SUBN"
#$ Vehicle Make                      : chr "TOYOT" "TOYOT" "FORD" "MITSU" "INTER" "ME/BE"
#$ Issuing Agency                    : chr "V" "V" "V" "T" "T" "V"
#$ Street Code1                      : int 0 0 0 10610 10510 0
#$ Street Code2                        : int 0 0 0 34330 34310 0
#$ Street Code3                        : int 0 0 0 34350 34330 0
#$ Vehicle Expiration Date             : int 0 0 0 20180630 20170228 0
#$ Violation Location                  : int NA NA NA 14 13 NA
#$ Violation Precinct                  : int 0 0 0 14 13 0
#$ Issuer Precinct                     : int 0 0 0 14 13 0
#$ Issuer Code                         : int 0 0 0 359594 364832 0
#$ Issuer Command                      : chr "NA" "NA" "NA" "T102" "T102" "NA"
#$ Issuer Squad                        : chr "NA" "NA" "NA" "J" "M" "NA"
#$ Violation Time                      : chr "0143A" "0400P" "0233P" "1120A" "0555P" "0852P"
#$ Time First Observed                 : chr "NA" "NA" "NA" "NA" "NA" "NA"
#$ Violation County                    : chr "BX" "BX" "BX" "NY" "NY" "QN"
#$ Violation In Front Of Or Opposite: chr "NA" "NA" "NA" "O" "F" "NA"
#$ House Number                        : chr "NA" "NA" "NA" "330" "799" "NA"
#$ Street Name                         : chr "ALLERTON AVE (W/B) @" "ALLERTON AVE (W/B) @" "SB WEBSTER AVE @ E 1" "7th Ave"
#$ Intersecting Street                 : chr "BARNES AVE" "BARNES AVE" "94TH ST" "NA" "NA" "@ MARATHON PKWY"
#$ Date First Observed                 : int 0 0 0 0 0 0
#$ Law Section                         : int 1111 1111 1111 408 408 1111
#$ Sub Division                        : chr "D" "D" "C" "l2" "h1" "D"
#$ Violation Legal Code                 : chr "T" "T" "T" "NA" "NA" "T"
#$ Days Parking In Effect              : chr "NA" "NA" "NA" "Y" "Y" "NA"
#$ From Hours In Effect                 : chr "NA" "NA" "NA" "0700A" "0700A" "NA"
#$ To Hours In Effect                   : chr "NA" "NA" "NA" "0700P" "0700P" "NA"
#$ Vehicle Color                       : chr "GY" "GY" "BK" "WH" "WHITE" "WH"
#$ Unregistered Vehicle?               : int NA NA NA NA NA NA
#$ Vehicle Year                        : int 2001 2001 2004 2007 2007 2012
#$ Meter Number                        : chr "NA" "NA" "NA" "NA" "NA" "NA"
#$ Feet From Curb                      : int 0 0 0 0 0 0
#$ Violation Post Code                 : chr "NA" "NA" "NA" "04" "31 6" "NA"
#$ Violation Description               : chr "FAILURE TO STOP AT RED LIGHT" "FAILURE TO STOP AT RED LIGHT" "BUS LANE VIOLAT
#$ No Standing or Stopping Violation: chr "NA" "NA" "NA" "NA" "NA" "NA"
#$ Hydrant Violation                   : chr "NA" "NA" "NA" "NA" "NA" "NA"
#$ Double Parking Violation            : chr "NA" "NA" "NA" "NA" "NA" "NA"
############################################################################################################################################
createOrReplaceTempView(d_2017, "data_2017")
############################################################################################################################################
# Filtering the data containing the columns of interest
selected_2017 <- SparkR::sql("select `Summons Number`,`Registration State`,`Issue Date`,`Violation Code`,`Vehicle Body Type`,`Vehicle Make`,`Violation Location`,`Violation Precinct`,`Issuer Precinct`,`Issuer Code`,`Violation Time`,`House Number`,`Street Name`,`Law Section`,`Sub Division`,`Days Parking In Effect    `,`From Hours In Effect`,`To Hours In Effect` from data_2017")
createOrReplaceTempView(selected_2017, "data_2017")
############################################################################################################################################
##Examine the data.

##1. Q1 Find total number of tickets for each year.
ticket_count_2017 <- SparkR::sql("select count(distinct(`Summons Number`)) from data_2017")
head(ticket_count_2017)
head(summarize(select(selected_2017,selected_2017$`Summons Number`), count = countDistinct(selected_2017$`Summons Number`)))
##10803028
############################################################################################################################################
##2. Q2 Find out how many unique states the cars which got parking tickets came from.
unique_states_2017 <- SparkR::sql("select count(distinct(`Registration State`)) from data_2017")
head(unique_states_2017)
head(summarize(select(selected_2017,selected_2017$`Registration State`), count = countDistinct(selected_2017$`Registration State`)))
#67
############################################################################################################################################
##3. Q3 Some parking tickets don’t have addresses on them, which is cause for concern. Find out how many such tickets there are.
```

```
head(count(where(selected_2017, ((isNull(selected_2017$`House Number`) & isNull(selected_2017$`Street Name`))|((selected_2017$`House Number` == "") & (selected_2017$`Street Name` == ""))))))
#2683
##############################################################################################################################################
#Performing some more quality checks and preparing the final filtered dataset for analysis
head(count(where(selected_2017, (isNull(selected_2017$`Summons Number`))|(selected_2017$`Summons Number` == ""))))
#0
head(count(where(selected_2017, (isNull(selected_2017$`Registration State`))|(selected_2017$`Registration State` == ""))))
#0
head(count(where(selected_2017, (isNull(selected_2017$`Issue Date`))|(selected_2017$`Issue Date` == ""))))
#0
head(count(where(selected_2017, (isNull(selected_2017$`Violation Code`))|(selected_2017$`Violation Code` == ""))))
#0
head(count(where(selected_2017, (isNull(selected_2017$`Vehicle Body Type`))|(selected_2017$`Vehicle Body Type` == ""))))
#42697
head(count(where(selected_2017, (isNull(selected_2017$`Vehicle Make`))|(selected_2017$`Vehicle Make` == ""))))
#73048
head(count(where(selected_2017, (isNull(selected_2017$`Violation Location`))|(selected_2017$`Violation Location` == ""))))
#2072400
head(count(where(selected_2017, (isNull(selected_2017$`Violation Precinct`))|(selected_2017$`Violation Precinct` == ""))))
#0
head(count(where(selected_2017, (isNull(selected_2017$`Issuer Precinct`))|(selected_2017$`Issuer Precinct` == ""))))
#0
head(count(where(selected_2017, (isNull(selected_2017$`Issuer Code`))|(selected_2017$`Issuer Code` == ""))))
#0
head(count(where(selected_2017, (isNull(selected_2017$`Violation Time`))|(selected_2017$`Violation Time` == ""))))
#63
head(count(where(selected_2017, (isNull(selected_2017$`Days Parking In Effect    `))|(selected_2017$`Days Parking In Effect    ` == ""))))
#2712416
head(count(where(selected_2017, (isNull(selected_2017$`From Hours In Effect`))|(selected_2017$`From Hours In Effect` == ""))))
#5450946
head(count(where(selected_2017, (isNull(selected_2017$`To Hours In Effect`))|(selected_2017$`To Hours In Effect` == ""))))
#5450943
##############################################################################################################################################


selected_2017 <- dropDuplicates(selected_2017, "Summons Number")

selected_2017 <- filter(selected_2017, ((isNotNull(selected_2017$`House Number`) | isNotNull(selected_2017$`Street Name`))|((selected_2017$`House Number` != "") | (selected_2017$`Street Name` != ""))))

selected_2017 <- filter(selected_2017, (isNotNull(selected_2017$`Violation Time`))|(selected_2017$`Violation Time` != ""))
createOrReplaceTempView(selected_2017,"data_2017")
##############################################################################################################################################
##Aggregation tasks
##1. Q1 How often does each violation code occur? (frequency of violation codes - find the top 5)
v_code_count_2017 <- summarize(groupBy(selected_2017,selected_2017$`Violation Code`),count = n(selected_2017$`Violation Code`))
head(arrange(v_code_count_2017, desc(v_code_count_2017$count)))

vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from
data_2017_1 group by `Violation Code` order by cnt desc limit 5")
head(vio_code_count)
#   Violation Code  cnt
#1             21   1528184
#2             36   1400614
#3             38   1062063
#4             14   893125
#5             20   618466
#6             46   599778
#############################################################################################################################################
##2. Q2 How often does each vehicle body type get a parking ticket? How about the vehicle make? (find the top 5 for both)
vbc_2017 <- summarize(groupBy(selected_2017,selected_2017$`Vehicle Body Type`),count = n(selected_2017$`Vehicle Body Type`))
head(arrange(vbc_2017, desc(vbc_2017$count)))

v_body_count <- SparkR::sql("select `Vehicle Body Type`, count(*) as cnt from data_2017 group by `Vehicle Body Type` order by cnt desc limit 5")
head(v_body_count)

#Vehicle Body Type      Count
#1            SUBN      3719191
#2            4DSD      3081839
#3            VAN       1411708
#4            DELV      687139
#5            SDN       437603
#6            2DSD      274362

vmc_2017 <- summarize(groupBy(selected_2017,selected_2017$`Vehicle Make`),count = n(selected_2017$`Vehicle Make`))
head(arrange(vmc_2017, desc(vmc_2017$count)))

v_make_count <- SparkR::sql("select `Vehicle Make`, count(*) as cnt from data_2017 group by `Vehicle Make` order by cnt desc limit 5")
head(v_body_count)

#   Vehicle Make    count
#1         FORD     1280743
#2        TOYOT     1211222
#3        HONDA     1079024
#4        NISSA     918433
#5        CHEVR     714510
#6        FRUEH     429090
#############################################################################################################################################
##3. Q3 A precinct is a police station that has a certain zone of the city under its command. Find the (5 highest) frequencies of:
#a. Violating Precincts (this is the precinct of the zone where the violation occurred)
vio_pre_2017 <- summarize(groupBy(selected_2017,selected_2017$`Violation Precinct`),count = n(selected_2017$`Violation Precinct`))
head(arrange(vio_pre_2017, desc(vio_pre_2017$count)))

vio_pre_sql_2017 <- SparkR::sql("select `Violation Precinct`, count(*) as cnt from data_2017 group by `Violation Precinct` order by cnt desc limit 5")
head(vio_pre_sql_2017)
#Violation Precinct     count
#1                 0     2071293
#2                19     535633
#3                14     352413
#4                 1     331752
#5                18     306882
#6               114     296482

b. Issuing Precincts (this is the precinct that issued the ticket)
iss_pre_2017 <- summarize(groupBy(selected_2017,selected_2017$`Issuer Precinct`),count = n(selected_2017$`Issuer Precinct`))
head(arrange(iss_pre_2017, desc(iss_pre_2017$count)))

iss_pre_sql_2017 <- SparkR::sql("select `Issuer Precinct`, count(*) as cnt from data_2017 group by `Issuer Precinct` order by cnt desc limit 5")
head(iss_pre_sql_2017)
```

```

#Issuer Precinct    count
#1              0    2387057
#2             19    521491
#3             14    344942
#4              1    321129
#5             18    296532
#6            114    289921
###########################################################################################################################################
##4. Q4 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?

vio_pre_code_2017 <- SparkR::sql("select `Issuer Precinct`,`Violation Code`, count(*) as cnt from data_2017 group by `Issuer Precinct`,`Violation Code` order by cnt desc limit 10")
head(vio_pre_code_2017)

#   Issuer Precinct Violation Code      cnt
#1                0             36      1400614
#2                0              7      516389
#3                0             21      268249
#4                0              5      145642
#5               18             14      91477
#6               19             46      86373
###########################################################################################################################################
##5. You'd want to find out the properties of parking violations across different times of the day:
##a. 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.

#Violation Time  : chr "0011A" "0942A" "1020A" "0318P" "0410P" "0839A"
#From summary of data we see that Violation time is stored as characters having alphabets A and P denoting AM and PM probably


##b. Find a way to deal with missing values, if any.

selected_2017$hr <- substr(selected_2017$`Violation Time`, 1, 2)
selected_2017$ampm <- substr(selected_2017$`Violation Time`, 6, 6)

selected_2017$vt_bin <- ifelse(selected_2017$hr != 12 & selected_2017$ampm == "P", selected_2017$hr + 12, selected_2017$hr)

createOrReplaceTempView(selected_2017,"data_2017")

##c. Divide 24 hours into 6 equal discrete bins of time. The intervals you choose are at your discretion.
selected_2017_hr <- SparkR::sql("select `Violation Code`, \
                   CASE  WHEN (vt_bin >= 4  and vt_bin < 8)  THEN 'early_morning'\
                   WHEN (vt_bin >= 8  and vt_bin < 12) THEN 'morning'\
                   WHEN (vt_bin >= 12 and vt_bin < 16) THEN 'after_noon'\
                   WHEN (vt_bin >= 16 and vt_bin < 20) THEN 'evening'\
                   WHEN (vt_bin >= 20 and vt_bin < 24) THEN 'night'\
                   ELSE 'late_night' END  as time_group FROM data_2017")

createOrReplaceTempView(selected_2017_hr,"selected_2017_hr")


##For each of these groups, find the 3 most commonly occurring violations
# For early_morning
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr where time_group = 'early_morning' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code     cnt
#1           14     141214
#2           21     119414
#3           40     112158

# For Morning
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr
where time_group = 'morning' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code     cnt
#1           21     1182416
#2           36     751422
#3           38     346409

# For after_noon
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr where time_group = 'after_noon' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code     cnt
#1           36     588395
#2           38     462765
#3           37     337045

# For evening
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr where time_group = 'evening' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code     cnt
#1           38     203203
#2           37     145773
#3           14     144704

# For night
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr where time_group = 'night' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code     cnt
#1            7     65593
#2           38     47025
#3           14     44755

# For late_night
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr where time_group = 'late_night' group by `Violation Code` order by cnt desc limit 3"))
#Violation Code     cnt
#1           21     73170
#2           40     45942
#3           14     29310

most_vio_2017 <- SparkR::sql("select time_group,`Violation Code`,count(*) as cnt from selected_2017_hr group by time_group,`Violation Code` order by cnt desc")

head(most_vio_2017)

##d. 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)
head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr group by `Violation Code` order by cnt desc limit 3"))
#Violation Code     cnt
#1           21     1528184
#2           36     1400614
#3           38     1062063

#Three most common Violation Code are
#The most common times of the day for these codes
head(SparkR::sql("Select time_group, count(*) as cnt from selected_2017_hr where `Violation Code` IN (21,38,14) group by time_group order by cnt desc"))
#   time_group      cnt
#1       morning    1803002
#2    after_noon    874002
#3       evening    348456
#4 early_morning    262923
#5    late_night    102846
#6         night    92143
#######################################################################################################################################
##6. Let’s try and find some seasonality in this data
##a. First, divide the year into some number of seasons, and find frequencies of tickets for each season.
seasons_2017 <- SparkR::sql("select `Violation Code`, \
                      CASE  WHEN (month(`Issue Date`) >= 1  and month(`Issue Date`) <= 3)  THEN 'Q1'\
                      WHEN (month(`Issue Date`) >= 4  and month(`Issue Date`) <= 6) THEN 'Q2'\
                   WHEN (month(`Issue Date`) >= 7 and month(`Issue Date`) <= 9) THEN 'Q3'\
                   ELSE 'Q4' END  as season FROM data_2017")

createOrReplaceTempView(seasons_2017,"seasons_2017")
##b. Then, find the 3 most common violations for each of these season
vio_seas_2017 <- SparkR::sql("select season,`Violation Code`,count(*) as cnt from seasons_2017 group by season,`Violation Code` order by cnt desc")
head(vio_seas_2017)
#   season Violation Code       cnt
#1     Q4              21         1528184
#2     Q4              36         1400614
#3     Q4              38         1062063
#4     Q4              14         893125
#5     Q4              20         618466
#6     Q4              46         599778

##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.
##a. Find total occurrences of the 3 most common violation codes
vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from data_2017 group by `Violation Code` order by cnt desc limit 5")
head(vio_code_count)
#   Violation Code      cnt
#1              21       1528184
#2              36       1400614
#3              38       1062063
#4              14       893125
#5              20       618466

##b. 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.
#Code   Average
#21     55
#36     50
#38     50
#14     115
#20     62.5

##c. Using this information, find the total amount collected for all of the fines. State the code which has the highest total collection.
#Code   Total Collection
#21     84050120
#36     70030700
#38     53103150
#14     102709375
#20     38654125

##d. What can you intuitively infer from these findings?
For 2017, we can infer that although Code #21 and #36 sees highest number of fines but the average cost is highest for code #14 which draws the highest collection among aall the fines.
```