```
spark_path <- '/usr/local/spark'
if (nchar(Sys.getenv("SPARK_HOME")) < 1) {
  Sys.setenv(SPARK_HOME = spark_path)
}
library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"),
"R", "lib")))
sparkR.session(master = "yarn", sparkConfig = list
(spark.driver.memory = "1g"))

# Library definition
#library(ggplot2)
#library(dplyr)
# For the scope of this analysis, we wish to compare the
phenomenon related to parking tickets over
# three different years - 2015, 2016, 2017. All the analysis
steps mentioned below should be done
# for three different years. Each metric you derive should be
compared across the three years.
# Use the Fiscal years as per the files. You can use calendar
year if you like - you will not lose any
# marks for performing the analysis this way.

# The purpose of this case study is to conduct an exploratory
data analysis that helps you understand the data.
# Since the size of the dataset is large, your queries will take
some time to run, and you will need to identify
# the correct queries quicker. The questions given below will
guide your analysis.

# The analysis has to be performed for all the three years. Apart
from analysis asked in the questions, provide
# a comparison of the findings from three years.

# 2. Create a Spark DataFrame and examine structure

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

nyc_ticket_2015 <- SparkR::read.df
("/common_folder/nyc_parking/Parking_Violations_Issued_-
_Fiscal_Year_2015.csv", "CSV", header="true", inferSchema =
"true")
colnames(nyc_ticket_2015) <- gsub(" ", "_", colnames(nyc_ticket_
2015))
#nyc_ticket_2015 <- nyc_ticket_2015[ , c(1, 3:9, 14:16, 20, 22,
40)]
str(nyc_ticket_2015)
# We can infer that columns are of various data types.

# Here are all the column names from nyc_ticket_2015
# Summons_Number                    Plate_ID
Registration_State                Plate_Type
```

```
# Issue_Date                          Violation_Code
Vehicle_Body_Type                   Vehicle_Make
# Issuing_Agency                      Street_Code1
Street_Code2                        Street_Code3
# Vehicle_Expiration_Date             Violation_Location
Violation_Precinct                  Issuer_Precinct
# Issuer_Code                         Issuer_Command
Issuer_Squad                        Violation_Time
# Time_First_Observed                 Violation_County
Violation_In_Front_Of_Or_Opposite House_Number
# Street_Name                         Intersecting_Street
Date_First_Observed                 Law_Section
# Sub_Division                        Violation_Legal_Code
Days_Parking_In_Effect____          From_Hours_In_Effect
# To_Hours_In_Effect                  Vehicle_Color
Unregistered_Vehicle?               Vehicle_Year
# Meter_Number                        Feet_From_Curb
Violation_Post_Code                 Violation_Description
# No_Standing_or_Stopping_Violation Hydrant_Violation
Double_Parking_Violation            Latitude
# Longitude                           Community_Board
Community_Council_                  Census_Tract
# BIN                                 BBL
NTA

createOrReplaceTempView(nyc_ticket_2015, "SQL_nyc_ticket_2015")

# Before executing any hive-sql query from RStudio, you need to
add a jar file in RStudio
sql("ADD JAR /opt/cloudera/parcels/CDH/lib/hive/lib/hive-
hcatalog-core-1.1.0-cdh5.11.2.jar")

# Examine the data
#
# Find the total number of tickets for each year.
nyc_tickets_2015 <- SparkR::sql("select count(distinct
(Summons_Number)) from SQL_nyc_ticket_2015")
head(nyc_tickets_2015)

# count(DISTINCT Summons_Number)
# 1                        10951256
# The number of tickets issued in 2015 are 10951256

# Find out the number of unique states from where the cars that
got parking tickets came from. (Hint: Use the column
'Registration State')
# There is a numeric entry in the column which should be
corrected. Replace it with the state having maximum entries.

# Give the number of unique states for each year again.

nyc_states_2015 <- SparkR::sql("select count(distinct
```

```
(Registration_State)) from SQL_nyc_ticket_2015")
head(nyc_states_2015)

# count(DISTINCT Registration_State)
# 1                                 69

# The number of distinct registration state are 69.

nyc_states_grouped_2015 <- SparkR::sql("select
Registration_State, count(Registration_State) as Count from
SQL_nyc_ticket_2015 group by Registration_State sort by count
(Registration_State) DESC")
head(arrange(nyc_states_grouped_2015, desc(nyc_states_grouped_
2015$Count)))

# Registration_State   Count
# 1                 NY 9193289
# 2                 NJ 1080414
# 3                 PA  298877
# 4                 CT  160361
# 5                 FL  148868
# 6                 MA  101164

# We can see that NY is the registration state with maximum
violations.

nyc_states_numeric_2015 <- SparkR::sql("select Registration_State
from SQL_nyc_ticket_2015 where Registration_State LIKE '%[^0-
9]%'")
head(nyc_states_numeric_2015)

head(arrange(nyc_states_grouped_2015, asc(nyc_states_grouped_2015
$Count)))

# We dont see any Registration_State entry with numeric value.

# Some parking tickets don’t have the address for violation
location on them, which is a cause for concern. Write a query to
check the number of such tickets.
# The values should not be deleted or imputed here. This is just
a check.

nyc_tickets_null_2015 <- SparkR::sql("select count
(Violation_Location) from SQL_nyc_ticket_2015 where House_Number
IS NULL or Street_Name IS NULL")
head(nyc_tickets_null_2015)

# count(Violation_Location)
# 1                   204880
# We can infer that that there are 204880 tickets which do not
have House number or street name.
```

```
# Aggregation tasks

# 1. How often does each violation code occur? Display the
frequency of the top five violation codes.

ViolationCodeFreq_2015 <- summarize(groupBy(nyc_ticket_2015,
nyc_ticket_2015$Violation_Code), count = n(nyc_ticket_2015
$Violation_Code))
head(arrange(ViolationCodeFreq_2015, desc(ViolationCodeFreq_2015
$count)),5)

# Violation Code   count
# 1             21 1630912
# 2             38 1418627
# 3             14  988469
# 4             36  839197
# 5             37  795918


# 2. How often does each 'vehicle body type' get a parking
ticket? How about the 'vehicle make'?
# (Hint: find the top 5 for both)
BodyTypeFreq_2015 <- summarize(groupBy(nyc_ticket_2015,
nyc_ticket_2015$Vehicle_Body_Type), count = n(nyc_ticket_2015
$Vehicle_Body_Type))
head(arrange(BodyTypeFreq_2015, desc(BodyTypeFreq_2015$count)),5)

# Vehicle Body Type   count
# 1              SUBN 3729346
# 2              4DSD 3340014
# 3               VAN 1709091
# 4              DELV  892781
# 5               SDN  524596

VehicleMakeFreq_2015 <- summarize(groupBy(nyc_ticket_2015,
nyc_ticket_2015$Vehicle_Make), count = n(nyc_ticket_2015
$Vehicle_Make))
head(arrange(VehicleMakeFreq_2015, desc(VehicleMakeFreq_2015
$count)),5)

# Vehicle Make   count
# 1         FORD 1521874
# 2        TOYOT 1217087
# 3        HONDA 1102614
# 4        NISSA  908783
# 5        CHEVR  897845

# 3. A precinct is a police station that has a certain zone of
the city under its command.
#
# 3.1 Find the (5 highest) frequency of tickets for each of the
following:
```

```
#
# 3.2 'Violation Precinct' (this is the precinct of the zone
where the violation occurred). Using this, can you make any
insights
# for parking violations in any specific areas of the city?

VehiclePrecintFreq_2015 <- summarize(groupBy(nyc_ticket_2015,
nyc_ticket_2015$Violation_Precinct), count = n(nyc_ticket_2015
$Violation_Precinct))
head(arrange(VehiclePrecintFreq_2015, desc(VehiclePrecintFreq_
2015$count)),5)

# Violation Precinct   count
# 1                  0 1799170
# 2                 19  598351
# 3                 18  427510
# 4                 14  409064
# 5                  1  329009

# There are many entries which have Violation Precinct as 0 which
are incorrect. Post which 19 has highest number of parking
violations.

# For using SQL, you need to create a temporary view

data_rated_2015_5 <- SparkR::sql("SELECT Violation_County, count
(Violation_County) from SQL_nyc_ticket_2015  where
Violation_Precinct = 19 group by Violation_County")
head(arrange(data_rated_2015_5, desc(count(data_rated_2015_5
$Violation_County))))

# Violation_County count(Violation_County)
# 1               NY                  595515
# 2               BX                      41
# 3                Q                      19
# 4                K                      18
# 5                R                      17
# 6             <NA>                       0

# We can infer that maximum parking violations are in the area of
NY county.

data_rated_2015_4 <- SparkR::sql("SELECT Street_Code1, count
(Street_Code1) from SQL_nyc_ticket_2015  where Violation_Precinct
= 19 group by Street_Code1")
head(arrange(data_rated_2015_4, desc(count(data_rated_2015_4
$Street_Code1))))

# Street_Code1 count(Street_Code1)
# 1       10210               79237
# 2       25390               60830
# 3       24890               58651
```

```
# 4        10010               50537
# 5        10110               31787
# 6        45590               23523

# We can inder that Street Code 10210, 25390, 24890 and 10010 has
high number of parking violations.

# 3.3 'Issuer Precinct' (this is the precinct that issued the
ticket)
# Here you would have noticed that the dataframe has 'Violating
Precinct' or 'Issuing Precinct' as '0'.
# These are the erroneous entries. Hence, provide the record for
five correct precincts.
# (Hint: print top six entries after sorting)

data_rated_2015_6 <- SparkR::sql("SELECT Issuer_Precinct, count
(Issuer_Precinct) from SQL_nyc_ticket_2015 where Issuer_Precinct
!= 0 group by Issuer_Precinct")
head(arrange(data_rated_2015_6, desc(count(data_rated_2015_6
$Issuer_Precinct))))

# Issuer_Precinct count(Issuer_Precinct)
# 1             19                  579998
# 2             18                  417329
# 3             14                  392922
# 4              1                  318778
# 5            114                  314437
# 6             13                  296403

# We can infer that Issuer_Precinct 19 has issued highest parking
violation.

# 4. Find the violation code frequency across three 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?
# Hint: You can analyse the three precincts together using the
'union all' attribute in SQL view. In the SQL view,
# use the 'where' attribute to filter among three precincts and
combine them using 'union all'.

data_rated_2015_7 <- SparkR::sql("
SELECT Violation_Code, count(Violation_Code) as counting from
SQL_nyc_ticket_2015 where Issuer_Precinct = 19 group by
Violation_Code UNION ALL
SELECT Violation_Code, count(Violation_Code) as counting from
SQL_nyc_ticket_2015 where Issuer_Precinct = 18 group by
Violation_Code UNION ALL
SELECT Violation_Code, count(Violation_Code) as counting from
SQL_nyc_ticket_2015 where Issuer_Precinct = 14 group by
Violation_Code")
```

```
head(data_rated_2015_7)

# Violation_Code counting
# 1             31     2462
# 2             85      715
# 3             65        2
# 4             53     2043
# 5             78     1273
# 6             81       22

data_rated_2015_8 <- SparkR::sql("SELECT Violation_Code, count
(Violation_Code) from SQL_nyc_ticket_2015 where Issuer_Precinct =
19 group by Violation_Code")
head(arrange(data_rated_2015_8, desc(count(data_rated_2015_8
$Violation_Code))))

# Violation_Code count(Violation_Code)
# 1             38                 97154
# 2             37                 85007
# 3             14                 64133
# 4             21                 60215
# 5             16                 59675
# 6             46                 46363

data_rated_2015_9 <- SparkR::sql("SELECT Violation_Code, count
(Violation_Code) from SQL_nyc_ticket_2015 where Issuer_Precinct =
18 group by Violation_Code")
head(arrange(data_rated_2015_9, desc(count(data_rated_2015_9
$Violation_Code))))

# Violation_Code count(Violation_Code)
# 1             14                129079
# 2             69                 60618
# 3             31                 32925
# 4             47                 30872
# 5             42                 21026
# 6             38                 20013

data_rated_2015_10 <- SparkR::sql("SELECT Violation_Code, count
(Violation_Code) from SQL_nyc_ticket_2015 where Issuer_Precinct =
14 group by Violation_Code")
head(arrange(data_rated_2015_10, desc(count(data_rated_2015_10
$Violation_Code))))

# Violation_Code count(Violation_Code)
# 1             69                 84895
# 2             14                 81896
# 3             31                 43928
# 4             42                 29868
# 5             47                 28814
# 6             46                 10853
```

```
# We can infer that violation code 14 and 69 seem to be the most
common violations in the 3 precincts.

#-----------------------------------------------------------------
-----------------------------------------------------------------
-
#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.
#-----------------------------------------------------------------
-----------------------------------------------------------------
--

nyc_ticket_2015 <- nyc_ticket_2015[ , c(1, 3:9, 14:16, 20, 22,
40)]

#year-2015
nyc_ticket_2015 <- withColumn(nyc_ticket_2015, "hours",
substr(nyc_ticket_2015$Violation_Time, 1, 2))
nyc_ticket_2015 <- withColumn(nyc_ticket_2015, "Period",
substr(nyc_ticket_2015$Violation_Time, 6, 6))
nyc_ticket_2015 <- withColumn(nyc_ticket_2015, "hours_bin",
ifelse(nyc_ticket_2015$Period == "P",

nyc_ticket_2015$hours + 12,

nyc_ticket_2015$hours))


#-----------------------------------------------------------------
-----------------------------------------------------------------
-
#Dealing  with missing values if
#-----------------------------------------------------------------
-----------------------------------------------------------------
-
#check of data and extract only valid data

str(nyc_ticket_2015)
nrow(nyc_ticket_2015)
#11809233
nrow(where(nyc_ticket_2015, nyc_ticket_2015$hours_bin <= 24))
#11807372 -- valid data
nrow(where(nyc_ticket_2015, nyc_ticket_2015$hours_bin >24))
#142     --|Errouneous data
nrow(where(nyc_ticket_2015, isNull(nyc_ticket_2015$hours)))
#1715

#((142+1715)/11809233)*100 = 0.015% of data which is very low and
```

```
can be omitted for further analysis
nyc_parking_violation_time_2015 <- subset(nyc_ticket_2015,
nyc_ticket_2015$hours_bin <= 24)

#Creating view for running SQL queries
createOrReplaceTempView(nyc_parking_violation_time_2015,
"data_violationtime_view_2015")

#-----------------------------------------------------------------
-----------------------------------------------------------------
----
# 5.3  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.
# Hint: Use the CASE-WHEN in SQL view to segregate into bins. For
finding the most commonly occurring violations,
# a similar approach can be used as mention in the hint for
question 4.

#-----------------------------------------------------------------
-----------------------------------------------------------------
----
# Binning into different hours and attaching data

#year-2015
bins_2015 <- SparkR::sql("SELECT Summons_Number,
Registration_State, Plate_Type, Issue_Date, Violation_Code,
Vehicle_Body_Type,
                    Vehicle_Make, Issuing_Agency,
Violation_Location, Violation_Precinct, Issuer_Precinct,
Violation_Time,
                    hours, Period,hours_bin,\
                    CASE  WHEN (hours_bin >= 0  and hours_bin <=
4 ) THEN 1\
                    WHEN (hours_bin >  4  and hours_bin <= 8 )
THEN 2\
                    WHEN (hours_bin >  8  and hours_bin <= 12)
THEN 3\
                    WHEN (hours_bin >  12 and hours_bin <= 16)
THEN 4\
                    WHEN (hours_bin >  16 and hours_bin <= 20)
THEN 5\
                    ELSE 6 END  as bin_number FROM
data_violationtime_view_2015")

# Attach the bin number to the original DataFrame
nyc_parking_violation_time_2015 <- withColumn(bins_2015,
"bin_number", bins_2015$bin_number)

#cross verifying structure and data
head(nyc_parking_violation_time_2015)
```

```
str(nyc_parking_violation_time_2015)

#------------------------------------------------------------------
-------------------------------------------------------------------
-----
#Summary  and obtaining to most commonly occuring violation codes
#------------------------------------------------------------------
-------------------------------------------------------------------
----
#year-2015
nyc_data_binning_2015 <- summarize(groupBy
(nyc_parking_violation_time_2015, nyc_parking_violation_time_2015
$bin_number,

nyc_parking_violation_time_2015$Violation_Code),
                                   count_violation_code = n
(nyc_parking_violation_time_2015$Violation_Code))
nyc_data_binning_2015 <- arrange(nyc_data_binning_2015, desc
(nyc_data_binning_2015$count_violation_code))

# bin 1
head(where(nyc_data_binning_2015, nyc_data_binning_2015
$bin_number == 1), 3)
# bin_number Violation_Code count_violation_code
# 1          1              38               582774
# 2          1              37               439650
# 3          1              14               339710


# bin 2
head(where(nyc_data_binning_2015, nyc_data_binning_2015
$bin_number == 2), 3)

# bin_number Violation_Code count_violation_code
# 1          2              21               526698
# 2          2              14               305265
# 3          2              38               246291

# bin 3
head(where(nyc_data_binning_2015, nyc_data_binning_2015
$bin_number == 3), 3)
# bin_number Violation_Code count_violation_code
# 1          3              21              1028370
# 2          3              38               589562
# 3          3              36               433328

# bin 4
head(where(nyc_data_binning_2015, nyc_data_binning_2015
$bin_number == 4), 3)

# bin_number Violation_Code count_violation_code
# 1          4              14                    3
```

```
# 2          4            21                    2
# 3          4            46                    2

# bin 5
head(where(nyc_data_binning_2015, nyc_data_binning_2015
$bin_number == 5), 3)

# bin_number Violation_Code count_violation_code
# 1          5            40                    4
# 2          5            20                    3
# 3          5            51                    2

# bin 6
head(where(nyc_data_binning_2015, nyc_data_binning_2015
$bin_number == 6), 3)
# bin_number Violation_Code count_violation_code
# 1          6            46                    3
# 2          6            40                    2
# 3          6            98                    1


#-----------------------------------------------------------------
-----------------------------------------------------------------
5.4
#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)
#-----------------------------------------------------------------
-----------------------------------------------------------------
---
#Year:: 2015
#Top 3 Violation codes for year-2015 are 21,38,14

data_code_binning_2015 <- summarize(groupBy(subset
(nyc_parking_violation_time_2015, nyc_parking_violation_time_2015
$Violation_Code %in% c(21,38,14)),

nyc_parking_violation_time_2015$Violation_Code,

nyc_parking_violation_time_2015$bin_number ),
                                    count_in_bin = n
(nyc_parking_violation_time_2015$bin_number))


head(arrange(data_code_binning_2015, desc(data_code_binning_2015
$count_in_bin)))

# Violation_Code bin_number count_in_bin
# 1             21          3      1028370
# 2             38          3       589562
# 3             38          1       582774
# 4             21          2       526698
```

```
# 5              14          3       343470
# 6              14          1       339710

#Observation for Year ::2015
# It looks like violation codes 21, 38 and 14 mostly happens in
bin 3,
#        which means these codes are mostly issed  between
morning 08:00 AM to 12:00PM


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

# 6.1 First, divide the year into some number of seasons, and
find frequencies of tickets for each season.
# (Hint: Use Issue Date to segregate into seasons)

# YEAR :: 2015
#Extract month  from issued  date values
parsed_2015_Month <- withColumn(nyc_ticket_2015, "Date_Parsed",
to_date(nyc_ticket_2015$Issue_Date, "MM/dd/yyyy"))
parsed_2015_Month <- withColumn(parsed_2015_Month, "Month", month
(parsed_2015_Month$'Date_Parsed'))

#checking for discripencies of data
nrow(where(parsed_2015_Month, parsed_2015_Month$Month <= 12 &
parsed_2015_Month$Month >= 1))
#11809233-- This valid data
nrow(where(parsed_2015_Month, parsed_2015_Month$Month <= 0 |
parsed_2015_Month$Month > 12))
#0  No Issue with this this data 0


nrow(where(parsed_2015_Month, isNull(parsed_2015_Month$Month)))
#0 #No Issue with this this data 0


#So all the data records are valid parsed_2015_Month can be  used
as  is
#Creating view for running SQL queries
createOrReplaceTempView(parsed_2015_Month, "ParsedDate_View_
2015")

data_bins_2015 <- SparkR::sql("SELECT Month,
Violation_Description, Violation_Code, \
                         CASE  WHEN (Month >=1  and Month <= 3)
THEN 1\
                         WHEN (Month > 3  and Month <= 6)  THEN 2
```

```
\
                         WHEN (Month > 6  and Month <= 9)  THEN 3
\
                         WHEN (Month > 9  and Month <= 12) THEN 4
\
                         ELSE 0 END  as bin_number FROM
ParsedDate_View_2015")

# Attach the bin number to the original DataFrame
nyc_parking_ParsedDate_2015 <- withColumn(data_bins_2015,
"bin_number", data_bins_2015$bin_number)


#-----------------------------------------------------------------
--------------------------------------------------------------
#grouping and summarising Viloation  in bins
#-----------------------------------------------------------------
--------------------------------------------------------------
# For year:: 2015
data_ParsedDate_2015 <- summarize(groupBy(nyc_parking_ParsedDate_
2015 ,
                                          nyc_parking_ParsedDate_
2015$bin_number),
                                  count_in_bin = n
(nyc_parking_ParsedDate_2015$bin_number))


#arranging the records in descending order
head(arrange(data_ParsedDate_2015, desc(data_ParsedDate_2015
$count_in_bin)))

# bin_number count_in_bin
# 1          2      3268456
# 2          1      3089975
# 3          3      2911162
# 4          4      2539640

#Observation: Maximum count in bin 2 which is for 4-6(April -
June) of the year
#### So Season Septermber-decmeber has maximum Tickets.


# 6.2 Then, find the three most common violations for each of
these seasons.
# (Hint: A similar approach can be used as mention in the hint
for question 4.)

#-----------------------------------------------------------------
---------------------------------------------------------
#grouping and summarising data to obtain the viloation code count

#Then, find the 3 most common violations for each of these 4
```

```
seasons.
#----------------------------------------------------------------
---------------------------------------------------------

data_ParsedDate_code2015 <- summarize(groupBy
(nyc_parking_ParsedDate_2015 ,

nyc_parking_ParsedDate_2015$bin_number,nyc_parking_ParsedDate_
2015$violation_code ),
                                      count_of_code = n
(nyc_parking_ParsedDate_2015$violation_code))

data_ParsedDate_code2015  <- arrange(data_ParsedDate_code2015,
desc(data_ParsedDate_code2015$count_of_code))
# bin 1
head(where(data_ParsedDate_code2015, data_ParsedDate_code2015
$bin_number == 1),3)
# bin_number violation_code count_of_code
# 1         1             38        419424
# 2         1             21        370713
# 3         1             14        271353

     # SO  Season-1 (Jan-March) have vialotation code 38, 21, 14

     # bin 2
     head(where(data_ParsedDate_code2015,
data_ParsedDate_code2015$bin_number == 2),3)
     # bin_number violation_code count_of_code
     # 1         2             21        471586
     # 2         2             38        346719
     # 3         2             14        262602

     # So Season-2 (Apr-June) have vialotation code 21, 38, 14


     # bin 3
     head(where(data_ParsedDate_code2015,
data_ParsedDate_code2015$bin_number == 3),3)
     # bin_number violation_code count_of_code
     # 1         3             21        412078
     # 2         3             38        352481
     # 3         3             14        240742

     # So Season-3 (July-Sept) have vialotation code 21, 38, 14

     # bin 4
     head(where(data_ParsedDate_code2015,
data_ParsedDate_code2015$bin_number == 4),3)
     # bin_number violation_code count_of_code
     # 1         4             21        376535
     # 2         4             38        300003
     # 3         4             14        213772
```

```
     # SO Season-4 (Oct-Dec) have vialotation code 14, 21, 38


#-----------------------------------------------------------------
-----------------------------------------------------------------
---
     #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.

#-----------------------------------------------------------------
-----------------------------------------------------------------
--------



     #year-2015
     data_violation_count_2015 <- summarize(groupBy
(nyc_parking_ParsedDate_2015, nyc_parking_ParsedDate_2015
$violation_code),
                                                  count_violation_code
= n(nyc_parking_ParsedDate_2015$violation_code))

     head(arrange(data_violation_count_2015, desc
(data_violation_count_2015$count_violation_code)))
     # violation_code count_violation_code
     # 1             21              1630912
     # 2             38              1418627
     # 3             14               988469
     # 4             36               839197
     # 5             37               795918
     # 6              7               719753


     #Observation :: for 3 most commonly occurring codes.
     # Violation code 21 (occurance - 1630912),
     # Viloation code 38 (occurance - 1418627),
     # viloation code 14 (occurance - 988469) are the most
commonly occuring for year 2015



     #Not Let' search the internet
https://www1.nyc.gov/site/finance/vehicles/services-violation-
codes.page  for NYC parking violation code fines.
     #You will find a website (on the nyc.gov URL) that lists
these fines.
     #They’re divided into 2 categories,
#    1) one for the highest-density locations of the city,
```

```
#    2) other for the rest of the city.(Manhattan 96th St & below
v/s All other Area
                                   #For simplicity, take an
average of the two.

#------------------------------------------------------------------
------------------------------------------------------------------
----


#------------------------------------------------------------------
------------------------------------------------------------------
---------
                                   #Reading fine amount dataset
required for question-7 calcaultion.
                                   # Assumptions made about the
fine - total 98 violation codes used for analysis
                                   # for violation code 4 other
areas fine is zero. Average not used here.
                                   # for violation code 6 average
taken on 1st chance and second chance.
                                   # violation code 99 not
included as the fine amount would vary

#------------------------------------------------------------------
------------------------------------------------------------------
---------

                                   # parking_fine_amount_2015 <-
read.df("s3://data-science-buket1/casestudy/fine_amount.csv",
source = "csv",
                                   #
inferSchema = "true", header = "true")



                                   # As per 3.3
                                   # We can infer that
Issuer_Precinct 19 has issued highest parking violation.

                                   # Fine is taken from
https://www1.nyc.gov/site/finance/vehicles/services-violation-
codes.page
                                   # (We take average of the fine
both both Area for simplicty   (in USD )

     # Violation Code Violation Count  Fine_per_Viloation_code
      Total_fine_Viloation_Code
     # 14  275108     115   31637420
     # 16  59675 95    5669125
     # 21  60215 55    3311825
     # 31  76853 115   8838095
     # 37  85007 55    4675385
```

```
    # 38  117167    50   5858350
    # 42  50894 55   2799170
    # 46  57216 115  6579840
    # 47  59686 115  6863890
    # 69  145513    65   9458345


                                  #Observation:
                                  ### Total Fine COllected
                                  # 31637420 fine collected with
Vilation_code  14 in 2015 and ranks top most Fine collected


                                  #For violation code 99 fine
amount varies, so not included
#                                  nrow(where(nyc_ticket_2015,
nyc_ticket_2015$violation_code == 99))
#
#                                  #joining parking data with
fine amount data obtained from NYC government site.
#                                  parking_fineamnt_data_2015
<- join(nyc_ticket_2015, parking_fine_amount,
#                                  nyc_ticket_2015
$violation_code == parking_fine_amount$violation_code,
"left_outer")
#
# parking_fineamnt_data_2015 <- summarize(groupBy
(parking_fineamnt_data_2015, parking_fineamnt_data_2015
$`violation code`),
# Total_Fines_Collected = sum(parking_fineamnt_data_2017
$fine_amount))

#head(arrange(parking_fineamnt_data_2015, desc
(parking_fineamnt_data_2015$Total_Fines_Collected)))


############################################# END 2015
###########################################################


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

    nyc_ticket_2016 <- SparkR::read.df
("/common_folder/nyc_parking/Parking_Violations_Issued_-
_Fiscal_Year_2016.csv", "CSV", header="true", inferSchema =
"true")
    colnames(nyc_ticket_2016) <- gsub(" ", "_", colnames
(nyc_ticket_2016))
    #nyc_ticket_2016 <- nyc_ticket_2016[ , c(1, 3:9, 14:16, 20,
22, 40)]
```

```
     str(nyc_ticket_2016)
     # We can infer that columns are of various data types.

     # Here are all the column names from nyc_ticket_2016
     # Summons_Number                      Plate_ID
Registration_State                Plate_Type
     # Issue_Date                          Violation_Code
Vehicle_Body_Type                 Vehicle_Make
     # Issuing_Agency                      Street_Code1
Street_Code2                      Street_Code3
     # Vehicle_Expiration_Date             Violation_Location
Violation_Precinct                Issuer_Precinct
     # Issuer_Code                         Issuer_Command
Issuer_Squad                      Violation_Time
     # Time_First_Observed                 Violation_County
Violation_In_Front_Of_Or_Opposite House_Number
     # Street_Name                         Intersecting_Street
Date_First_Observed               Law_Section
     # Sub_Division                        Violation_Legal_Code
Days_Parking_In_Effect____        From_Hours_In_Effect
     # To_Hours_In_Effect                  Vehicle_Color
Unregistered_Vehicle?             Vehicle_Year
     # Meter_Number                        Feet_From_Curb
Violation_Post_Code               Violation_Description
     # No_Standing_or_Stopping_Violation Hydrant_Violation
Double_Parking_Violation          Latitude
     # Longitude                           Community_Board
Community_Council_                Census_Tract
     # BIN                                 BBL
NTA

     createOrReplaceTempView(nyc_ticket_2016, "SQL_nyc_ticket_
2016")

     # Before executing any hive-sql query from RStudio, you need
to add a jar file in RStudio
     sql("ADD JAR /opt/cloudera/parcels/CDH/lib/hive/lib/hive-
hcatalog-core-1.1.0-cdh5.11.2.jar")

     # Examine the data
     #
     # Find the total number of tickets for each year.
     nyc_tickets_2016 <- SparkR::sql("select count(distinct
(Summons_Number)) from SQL_nyc_ticket_2016")
     head(nyc_tickets_2016)

     # count(DISTINCT Summons_Number)
     # 1                       10626899
     # The number of tickets issued in 2016 are 10626899

     # Find out the number of unique states from where the cars
that got parking tickets came from. (Hint: Use the column
```

'Registration State')
    # There is a numeric entry in the column which should be corrected. Replace it with the state having maximum entries.

    # Give the number of unique states for each year again.

```
nyc_states_2016 <- SparkR::sql("select count(distinct
(Registration_State)) from SQL_nyc_ticket_2016")
head(nyc_states_2016)

# count(DISTINCT Registration_State)
# 1                                68

# The number of distinct registration state are 68.

nyc_states_grouped_2016 <- SparkR::sql("select
Registration_State, count(Registration_State) as Count from
SQL_nyc_ticket_2016 group by Registration_State sort by count
(Registration_State) DESC")
head(arrange(nyc_states_grouped_2016, desc
(nyc_states_grouped_2016$Count)))

# Registration_State   Count
#1                NY 8260189
#2                NJ  968839
#3                PA  259177
#4                CT  145153
#5                FL  138647
#6                MA   99115

# We can see that NY is the registration state with maximum
violations.

nyc_states_numeric_2016 <- SparkR::sql("select
Registration_State from SQL_nyc_ticket_2016 where
Registration_State LIKE '%[^0-9]%'")
head(nyc_states_numeric_2016)

head(arrange(nyc_states_grouped_2016, asc
(nyc_states_grouped_2016$Count)))
#  Registration_State Count
#1                NT     2
#2                YT     5
#3                MX    11
#4                FO    13
#5                SK    18
#6                MB    32
```

    # Some parking tickets don’t have the address for violation location on them, which is a cause for concern. Write a query to check the number of such tickets.
    # The values should not be deleted or imputed here. This is

just a check.

```
     nyc_tickets_null_2016 <- SparkR::sql("select count
(Violation_Location) from SQL_nyc_ticket_2016 where House_Number
IS NULL or Street_Name IS NULL")
     head(nyc_tickets_null_2016)

     # count(Violation_Location)
     # 1                   174570
     # We can infer that that there are 174570 tickets which do
not have House number or street name.

     # Aggregation tasks

     # 1. How often does each violation code occur? Display the
frequency of the top five violation codes.

     ViolationCodeFreq_2016 <- summarize(groupBy(nyc_ticket_2016,
nyc_ticket_2016$Violation_Code), count = n(nyc_ticket_2016
$Violation_Code))
     head(arrange(ViolationCodeFreq_2016, desc(ViolationCodeFreq_
2016$count)),5)

     #  Violation_Code   count
     #1             21 1531587
     #2             36 1253512
     #3             38 1143696
     #4             14  875614
     #5             37  686610


     # 2. How often does each 'vehicle body type' get a parking
ticket? How about the 'vehicle make'?
     # (Hint: find the top 5 for both)
     BodyTypeFreq_2016 <- summarize(groupBy(nyc_ticket_2016,
nyc_ticket_2016$Vehicle_Body_Type), count = n(nyc_ticket_2016
$Vehicle_Body_Type))
     head(arrange(BodyTypeFreq_2016, desc(BodyTypeFreq_2016
$count)),5)

     #  Vehicle_Body_Type   count
     #1              SUBN 3466037
     #2              4DSD 2992107
     #3               VAN 1518303
     #4              DELV  755282
     #5               SDN  424043

     VehicleMakeFreq_2016 <- summarize(groupBy(nyc_ticket_2016,
nyc_ticket_2016$Vehicle_Make), count = n(nyc_ticket_2016
$Vehicle_Make))
     head(arrange(VehicleMakeFreq_2016, desc(VehicleMakeFreq_2016
$count)),5)
```

```
#  Vehicle_Make   count
#1         FORD 1324774
#2        TOYOT 1154790
#3        HONDA 1014074
#4        NISSA  834833
#5        CHEVR  759663

# 3. A precinct is a police station that has a certain zone of the city under its command.
#
# 3.1 Find the (5 highest) frequency of tickets for each of the following:
#
# 3.2 'Violation Precinct' (this is the precinct of the zone where the violation occurred). Using this, can you make any insights
# for parking violations in any specific areas of the city?

VehiclePrecintFreq_2016 <- summarize(groupBy(nyc_ticket_2016, nyc_ticket_2016$Violation_Precinct), count = n(nyc_ticket_2016$Violation_Precinct))
head(arrange(VehiclePrecintFreq_2016, desc(VehiclePrecintFreq_2016$count)),5)

#  Violation_Precinct   count
#1                  0 1868655
#2                 19  554465
#3                 18  331704
#4                 14  324467
#5                  1  303850

# There are many entries which have Violation Precinct as 0 which are incorrect. Post which 19 has highest number of parking violations.

# For using SQL, you need to create a temporary view

data_rated_2016_5 <- SparkR::sql("SELECT Violation_County, count(Violation_County) from SQL_nyc_ticket_2016  where Violation_Precinct = 19 group by Violation_County")
head(arrange(data_rated_2016_5, desc(count(data_rated_2016_5$Violation_County))))

#  Violation_County count(Violation_County)
#1               NY                  550758
#2                R                      19
#3                Q                      17
#4                K                      11
#5               BX                      10
#6             <NA>                       0
```

```
	# We can infer that maximum parking violations are in the area of NY county.

	data_rated_2016_4 <- SparkR::sql("SELECT Street_Code1, count(Street_Code1) from SQL_nyc_ticket_2016  where Violation_Precinct = 19 group by Street_Code1")
	head(arrange(data_rated_2016_4, desc(count(data_rated_2016_4$Street_Code1))))

	#  Street_Code1 count(Street_Code1)
	#1        10210               79238
	#2        25390               62236
	#       24890               53738
	#4        10010               42117
	#5        10110               34436
	#6        45590               20997
	# We can inder that Street Code 10210, 25390, 24890 and 10010 has high number of parking violations.

	# 3.3 'Issuer Precinct' (this is the precinct that issued the ticket)
	# Here you would have noticed that the dataframe has 'Violating Precinct' or 'Issuing Precinct' as '0'.
	# These are the erroneous entries. Hence, provide the record for five correct precincts.
	# (Hint: print top six entries after sorting)

	data_rated_2016_6 <- SparkR::sql("SELECT Issuer_Precinct, count(Issuer_Precinct) from SQL_nyc_ticket_2016 where Issuer_Precinct != 0 group by Issuer_Precinct")
	head(arrange(data_rated_2016_6, desc(count(data_rated_2016_6$Issuer_Precinct))))

	#  Issuer_Precinct count(Issuer_Precinct)
	#1              19                 540569
	#2              18                 323132
	#3              14                 315311
	#4               1                 295013
	#5             114                 286924
	#6              13                 282635

	# We can infer that Issuer_Precinct 19 has issued highest parking violation.

	# 4. Find the violation code frequency across three 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?
	# Hint: You can analyse the three precincts together using the 'union all' attribute in SQL view. In the SQL view,
	# use the 'where' attribute to filter among three precincts
```

and combine them using 'union all'.

```
     data_rated_2016_7 <- SparkR::sql("
                                      SELECT Violation_Code,
count(Violation_Code) as counting from SQL_nyc_ticket_2016 where
Issuer_Precinct = 19 group by Violation_Code UNION ALL
                                      SELECT Violation_Code,
count(Violation_Code) as counting from SQL_nyc_ticket_2016 where
Issuer_Precinct = 18 group by Violation_Code UNION ALL
                                      SELECT Violation_Code,
count(Violation_Code) as counting from SQL_nyc_ticket_2016 where
Issuer_Precinct = 14 group by Violation_Code")

     head(data_rated_2016_7)

     #  Violation_Code counting
     #1             31     2533
     #2             85      843
     #3             65        1
     #4             53     1367
     #5             78      810
     #6             81       25

     data_rated_2016_8 <- SparkR::sql("SELECT Violation_Code,
count(Violation_Code) from SQL_nyc_ticket_2016 where
Issuer_Precinct = 19 group by Violation_Code")
     head(arrange(data_rated_2016_8, desc(count(data_rated_2016_8
$Violation_Code))))

     #  Violation_Code count(Violation_Code)
     #1             38                 77183
     #2             37                 75641
     #3             46                 73016
     #4             14                 61742
     #5             21                 58719
     #6             16                 52354

     data_rated_2016_9 <- SparkR::sql("SELECT Violation_Code,
count(Violation_Code) from SQL_nyc_ticket_2016 where
Issuer_Precinct = 18 group by Violation_Code")
     head(arrange(data_rated_2016_9, desc(count(data_rated_2016_9
$Violation_Code))))

     #  Violation_Code count(Violation_Code)
     #1             14                 99857
     #2             69                 47881
     #3             47                 24009
     #4             31                 22809
     #5             42                 17678
     #6             46                 14674

     data_rated_2016_10 <- SparkR::sql("SELECT Violation_Code,
```

```
count(Violation_Code) from SQL_nyc_ticket_2016 where
Issuer_Precinct = 14 group by Violation_Code")
     head(arrange(data_rated_2016_10, desc(count(data_rated_2016_
10$Violation_Code))))

     # Violation_Code count(Violation_Code)
     # 1             69                   84895
     # 2             14                   81896
     # 3             31                   43928
     # 4             42                   29868
     # 5             47                   28814
     # 6             46                   10853

     # We can infer that violation code 14 and 69 seem to be the
most common violations in the 3 precincts.


#-----------------------------------------------------------------
-----------------------------------------------------------------
-
     #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.

#-----------------------------------------------------------------
-----------------------------------------------------------------
--

     nyc_ticket_2016 <- nyc_ticket_2016[ , c(1, 3:9, 14:16, 20,
22, 40)]

     #year-2016
     nyc_ticket_2016 <- withColumn(nyc_ticket_2016, "hours",
substr(nyc_ticket_2016$Violation_Time, 1, 2))
     nyc_ticket_2016 <- withColumn(nyc_ticket_2016, "Period",
substr(nyc_ticket_2016$Violation_Time, 6, 6))
     nyc_ticket_2016 <- withColumn(nyc_ticket_2016, "hours_bin",
ifelse(nyc_ticket_2016$Period == "P",

nyc_ticket_2016$hours + 12,

nyc_ticket_2016$hours))



#-----------------------------------------------------------------
-----------------------------------------------------------------
-
     #Dealing  with missing values if
```

```
#----------------------------------------------------------------
-----------------------------------------------------------------
-
     #check of data and extract only valid data

     str(nyc_ticket_2016)
     nrow(nyc_ticket_2016)
     #10626899
     nrow(where(nyc_ticket_2016, nyc_ticket_2016$hours_bin <=
24))
     #10622402 -- valid data
     nrow(where(nyc_ticket_2016, nyc_ticket_2016$hours_bin >24))
     #216     --|Errouneous data
     nrow(where(nyc_ticket_2016, isNull(nyc_ticket_2016$hours)))
     #4280

     #((216+4280)/11809233)*100 = 0.016% of data which is very
low and can be omitted for further analysis
     nyc_parking_violation_time_2016 <- subset(nyc_ticket_2016,
nyc_ticket_2016$hours_bin <= 24)

     #Creating view for running SQL queries
     createOrReplaceTempView(nyc_parking_violation_time_2016,
"data_violationtime_view_2016")


#----------------------------------------------------------------
-----------------------------------------------------------------
----
     # 5.3  Divide 24 hours into 6 equal discrete bins_2016 of
time. The intervals you choose are at your discretion.
     #For each of these groups, find the 3 most commonly
occurring violations.
     # Hint: Use the CASE-WHEN in SQL view to segregate into
bins_2016. For finding the most commonly occurring violations,
     # a similar approach can be used as mention in the hint for
question 4.


#----------------------------------------------------------------
-----------------------------------------------------------------
----
     # Binning into different hours and attaching data

     #year-2016
     bins_2016 <- SparkR::sql("SELECT Summons_Number,
Registration_State, Plate_Type, Issue_Date, Violation_Code,
Vehicle_Body_Type,
                              Vehicle_Make, Issuing_Agency,
Violation_Location, Violation_Precinct, Issuer_Precinct,
Violation_Time,
                              hours, Period,hours_bin,\
```

```
                              CASE  WHEN (hours_bin >= 0  and
hours_bin <= 4 ) THEN 1\
                              WHEN (hours_bin >  4  and hours_bin
<= 8 ) THEN 2\
                              WHEN (hours_bin >  8  and hours_bin
<= 12) THEN 3\
                              WHEN (hours_bin >  12 and hours_bin
<= 16) THEN 4\
                              WHEN (hours_bin >  16 and hours_bin
<= 20) THEN 5\
                              ELSE 6 END  as bin_number FROM
data_violationtime_view_2016")

     # Attach the bin number to the original DataFrame
     nyc_parking_violation_time_2016 <- withColumn(bins_2016,
"bin_number", bins_2016$bin_number)

     #cross verifying structure and data
     head(nyc_parking_violation_time_2016)
     str(nyc_parking_violation_time_2016)


#-----------------------------------------------------------------
-----------------------------------------------------------------
-----
     #Summary  and obtaining to most commonly occuring violation
codes

#-----------------------------------------------------------------
-----------------------------------------------------------------
----
     #year-2016
     nyc_data_binning_2016 <- summarize(groupBy
(nyc_parking_violation_time_2016, nyc_parking_violation_time_2016
$bin_number,

nyc_parking_violation_time_2016$Violation_Code),
                                        count_violation_code = n
(nyc_parking_violation_time_2016$Violation_Code))
     nyc_data_binning_2016 <- arrange(nyc_data_binning_2016, desc
(nyc_data_binning_2016$count_violation_code))

     # bin 1
     head(where(nyc_data_binning_2016, nyc_data_binning_2016
$bin_number == 1), 3)
     #  bin_number Violation_Code count_violation_code
     #1          1             38               463948
     #2          1             36               406670
     #3          1             37               377856


     # bin 2
```

```
     head(where(nyc_data_binning_2016, nyc_data_binning_2016
$bin_number == 2), 3)

     #  bin_number Violation_Code count_violation_code
     #1          2             21               502136
     #2          2             14               280889
     #3          2             36               206156

     # bin 3
     head(where(nyc_data_binning_2016, nyc_data_binning_2016
$bin_number == 3), 3)
     #  bin_number Violation_Code count_violation_code
     #1          3             21               955706
     #2          3             36               640685
     #3          3             38               483001

     # bin 4
     head(where(nyc_data_binning_2016, nyc_data_binning_2016
$bin_number == 4), 3)

     #  bin_number Violation_Code count_violation_code
     #1          4             46                    4
     #2          4             40                    3
     #3          4             99                    1

     # bin 5
     head(where(nyc_data_binning_2016, nyc_data_binning_2016
$bin_number == 5), 3)

     #  bin_number Violation_Code count_violation_code
     #1          5             21                    3
     #2          5             19                    2
     #3          5             70                    1

     # bin 6
     head(where(nyc_data_binning_2016, nyc_data_binning_2016
$bin_number == 6), 3)
     #  bin_number Violation_Code count_violation_code
     #1          6             46                    2
     #2          6             20                    2
     #3          6             98                    2


#-----------------------------------------------------------------
-----------------------------------------------------------------
     5.4
     #Now, try another direction. For the 3 most commonly
occurring violation codes, find the most common times of day
     #(in terms of the bins_2016 from the previous part)

#-----------------------------------------------------------------
-----------------------------------------------------------------
```

```
---
     #Year:: 2016
     #Top 3 Violation codes for year-2016 are 21,36,38

     data_code_binning_2016 <- summarize(groupBy(subset
(nyc_parking_violation_time_2016, nyc_parking_violation_time_2016
$Violation_Code %in% c(21,36,38)),

nyc_parking_violation_time_2016$Violation_Code,

nyc_parking_violation_time_2016$bin_number ),
                                         count_in_bin = n
(nyc_parking_violation_time_2016$bin_number))


     head(arrange(data_code_binning_2016, desc(data_code_binning_
2016$count_in_bin)))

     #  Violation_Code bin_number count_in_bin
     #1             21          3       955706
     #2             36          3       640685
     #3             21          2       502136
     #4             38          3       483001
     #5             38          1       463948
     #6             36          1       406670

     #Observation for Year ::2016
     # It looks like violation codes 21, 36 and 38 mostly happens
in bin 3,
     #       which means these codes are mostly issed  between
morning 08:00 AM to 12:00PM


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

#-----------------------------------------------------------------
-------------------------------------------------------------

     # 6.1 First, divide the year into some number of seasons,
and find frequencies of tickets for each season.
     # (Hint: Use Issue Date to segregate into seasons)

     # YEAR :: 2016
     #Extract month  from issued  date values
     parsed_2016_Month <- withColumn(nyc_ticket_2016,
"Date_Parsed", to_date(nyc_ticket_2016$Issue_Date, "MM/dd/yyyy"))
     parsed_2016_Month <- withColumn(parsed_2016_Month, "Month",
month(parsed_2016_Month$'Date_Parsed'))
```

```
     #checking for discripencies of data
     nrow(where(parsed_2016_Month, parsed_2016_Month$Month <= 12
& parsed_2016_Month$Month >= 1))
     #10626899-- This valid data
     nrow(where(parsed_2016_Month, parsed_2016_Month$Month <= 0 |
parsed_2016_Month$Month > 12))
     #0  No Issue with this this data 0


     nrow(where(parsed_2016_Month, isNull(parsed_2016_Month
$Month)))
     #0 #No Issue with this this data 0


     #So all the data records are valid parsed_2016_Month can be
used as  is
     #Creating view for running SQL queries
     createOrReplaceTempView(parsed_2016_Month, "ParsedDate_View_
2016")

     data_bins_2016 <- SparkR::sql("SELECT Month,
Violation_Description, Violation_Code, \
                                   CASE  WHEN (Month >=1  and
Month <= 3)  THEN 1\
                                   WHEN (Month > 3  and Month <=
6)  THEN 2\
                                   WHEN (Month > 6  and Month <=
9)  THEN 3\
                                   WHEN (Month > 9  and Month <=
12) THEN 4\
                                   ELSE 0 END  as bin_number FROM
ParsedDate_View_2016")

     # Attach the bin number to the original DataFrame
     nyc_parking_ParsedDate_2016 <- withColumn(data_bins_2016,
"bin_number", data_bins_2016$bin_number)



#-----------------------------------------------------------------
-------------------------------------------------------------
     #grouping and summarising Viloation  in bins_2016

#-----------------------------------------------------------------
-------------------------------------------------------------
     # For year:: 2016
     data_ParsedDate_2016 <- summarize(groupBy
(nyc_parking_ParsedDate_2016 ,

nyc_parking_ParsedDate_2016$bin_number),
                                       count_in_bin = n
```

```
(nyc_parking_ParsedDate_2016$bin_number))


     #arranging the records in descending order
     head(arrange(data_ParsedDate_2016, desc(data_ParsedDate_2016
$count_in_bin)))

     #  bin_number count_in_bin
     #1          4      2801028
     #2          3      2728663
     #3          1      2671331
     #4          2      2425877

     #Observation: Maximum count in bin 4 which is for 10-12(Oct
-JDec) of the year
     #### So Season Septermber-decmeber has maximum Tickets.


     # 6.2 Then, find the three most common violations for each
of these seasons.
     # (Hint: A similar approach can be used as mention in the
hint for question 4.)


#-----------------------------------------------------------------
---------------------------------------------------------
     #grouping and summarising data to obtain the viloation code
count

     #Then, find the 3 most common violations for each of these 4
seasons.

#-----------------------------------------------------------------
---------------------------------------------------------

     data_ParsedDate_code2016 <- summarize(groupBy
(nyc_parking_ParsedDate_2016 ,

nyc_parking_ParsedDate_2016$bin_number,nyc_parking_ParsedDate_
2016$violation_code ),
                                           count_of_code = n
(nyc_parking_ParsedDate_2016$violation_code))

     data_ParsedDate_code2016  <- arrange
(data_ParsedDate_code2016, desc(data_ParsedDate_code2016
$count_of_code))
     # bin 1
     head(where(data_ParsedDate_code2016,
data_ParsedDate_code2016$bin_number == 1),3)
     # bin_number violation_code count_of_code
     #1          1              21        349644
     #2          1              36        341787
```

```
    #3          1             38        308999

    # SO  Season-1 (Jan-March) have vialotation code 21, 36, 38

    # bin 2
    head(where(data_ParsedDate_code2016,
data_ParsedDate_code2016$bin_number == 2),3)
    #   bin_number violation_code count_of_code
    #1          2             21        348473
    #2          2             36        294015
    #3          2             38        254909

    # So Season-2 (Apr-June) have vialotation code 21, 36, 38,

    # bin 3
    head(where(data_ParsedDate_code2016,
data_ParsedDate_code2016$bin_number == 3),3)
    #     bin_number violation_code count_of_code
    #1          3             21        403720
    #2          3             38        305360
    #3          3             14        234943

    # So Season-3 (July-Sept) have vialotation code 21, 38, 14

    # bin 4
    head(where(data_ParsedDate_code2016,
data_ParsedDate_code2016$bin_number == 4),3)
    #     bin_number violation_code count_of_code
    #1          4             36        433966
    #2          4             21        429750
    #3          4             38        274428

    # SO Season-4 (Oct-Dec) have vialotation code 36,21,38


#-----------------------------------------------------------------
-----------------------------------------------------------------
---
    #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.

#-----------------------------------------------------------------
-----------------------------------------------------------------
--------


    #year-2016
    data_violation_count_2016 <- summarize(groupBy
```

```
(nyc_parking_ParsedDate_2016, nyc_parking_ParsedDate_2016
$violation_code),
                                            count_violation_code
= n(nyc_parking_ParsedDate_2016$violation_code))

     head(arrange(data_violation_count_2016, desc
(data_violation_count_2016$count_violation_code)))
     #     violation_code count_violation_code
     #1              21              1531587
     #2              36              1253512
     #3              38              1143696
     #4              14               875614
     #5              37               686610
     #6              20               611013


     #Observation :: for 3 most commonly occurring codes.
     # Violation code 21 (occurance - 1531587),
     # Viloation code 36 (occurance - 1253512),
     # viloation code 38 (occurance - 1143696) are the most
commonly occuring for year 2016



     #Not Let' search the internet
https://www1.nyc.gov/site/finance/vehicles/services-violation-
codes.page  for NYC parking violation code fines.
     #You will find a website (on the nyc.gov URL) that lists
these fines.
     #They're divided into 2 categories,
     #    1) one for the highest-density locations of the city,
     #    2) other for the rest of the city.(Manhattan 96th St &
below v/s All other Area
     #For simplicity, take an average of the two.

#-----------------------------------------------------------------
-----------------------------------------------------------------
----


#-----------------------------------------------------------------
-----------------------------------------------------------------
---------
     #Reading fine amount dataset required for question-7
calcaultion.
     # Assumptions made about the fine - total 98 violation codes
used for analysis
     # for violation code 4 other areas fine is zero. Average not
used here.
     # for violation code 6 average taken on 1st chance and
second chance.
     # violation code 99 not included as the fine amount would
```

```
vary

#-------------------------------------------------------------------
-------------------------------------------------------------------
---------

     # parking_fine_amount_2016 <- read.df("s3://data-science-
buket1/casestudy/fine_amount.csv", source = "csv",
     #                              inferSchema = "true", header
= "true")


     # As per 3.3
     # We can infer that Issuer_Precinct 19 has issued highest
parking violation.

     # Fine is taken from
https://www1.nyc.gov/site/finance/vehicles/services-violation-
codes.page
     # (We take average of the fine both both Area for simplicty
(in USD )


     #Violation_Code  Fine_per_Viloation_code
      Total_fine_Viloation_Code
     #38                           50
      3859150
     #37                           55
      4160255
     #46                           115
      10084350
     #14                           115
      18583885
     #21                           55
      9528750
     #16                           95
      4973630
     #69                           62
      8046174
     #47                           115
      4009130
     #31                           115
      13289170
     #42                           55
      972290
     #Observation:
     ### Total Fine COllected
     #18583885 fine collected with Violation_code  14 in 2016 and
ranks top most Fine collected

################################################### END of 2016
############################################################
```

```
################################################### 2017
############################################################

nyc_ticket_2017 <- SparkR::read.df
("/common_folder/nyc_parking/Parking_Violations_Issued_-
_Fiscal_Year_2017.csv", "CSV", header="true", inferSchema =
"true")
colnames(nyc_ticket_2017) <- gsub(" ", "_", colnames(nyc_ticket_
2017))
str(nyc_ticket_2017)
# We can infer that columns are of various data types.
# Remove unwanted columns
#nyc_ticket_2017 <- nyc_ticket_2017[ , c(1, 3:9, 14:16, 20, 22,
40)]

# For using SQL, you need to create a temporary view
createOrReplaceTempView(nyc_ticket_2017, "SQL_nyc_ticket_2017")

# Before executing any hive-sql query from RStudio, you need to
add a jar file in RStudio
sql("ADD JAR /opt/cloudera/parcels/CDH/lib/hive/lib/hive-
hcatalog-core-1.1.0-cdh5.11.2.jar")


# Examine the data
#
# Find the total number of tickets for each year.
nyc_tickets_2017 <- SparkR::sql("select count(distinct
(Summons_Number)) from SQL_nyc_ticket_2017")
head(nyc_tickets_2017)

# count(DISTINCT Summons_Number)
# 1                       10803028
# The number of tickets issued in 2017 are 10803028

# Find out the number of unique states from where the cars that
got parking tickets came from. (Hint: Use the column
'Registration State')
# There is a numeric entry in the column which should be
corrected. Replace it with the state having maximum entries.

# Give the number of unique states for each year again.

nyc_states_2017 <- SparkR::sql("select count(distinct
(Registration_State)) from SQL_nyc_ticket_2017")
head(nyc_states_2017)

# count(DISTINCT Registration_State)
# 1                                 67

# The number of distinct registration state are 67.
```

```
nyc_states_grouped_2017 <- SparkR::sql("select
Registration_State, count(Registration_State) as Count from
SQL_nyc_ticket_2017 group by Registration_State sort by count
(Registration_State) DESC")
head(arrange(nyc_states_grouped_2017, desc(nyc_states_grouped_
2017$Count)))

# Registration_State   Count
# 1                NY 8481061
# 2                NJ  925965
# 3                PA  285419
# 4                FL  144556
# 5                CT  141088
# 6                MA   85547

# We can see that NY is the registration state with maximum
violations.

nyc_states_numeric_2017 <- SparkR::sql("select Registration_State
from SQL_nyc_ticket_2017 where Registration_State LIKE '%[^0-
9]%'")
head(nyc_states_numeric_2017)

head(arrange(nyc_states_grouped_2017, asc(nyc_states_grouped_2017
$Count)))

# We dont see any Registration_State entry with numeric value.

# Some parking tickets don’t have the address for violation
location on them, which is a cause for concern. Write a query to
check the number of such tickets.
# The values should not be deleted or imputed here. This is just
a check.

nyc_tickets_null_2017 <- SparkR::sql("select count
(Violation_Location) from SQL_nyc_ticket_2017 where House_Number
IS NULL or Street_Name IS NULL")
head(nyc_tickets_null_2017)

# count(Violation_Location)
# 1                   225065
# We can infer that that there are 225065 tickets which do not
have House number or street name.


  # Aggregation tasks
  #
  # 1. How often does each violation code occur? Display the
frequency of the top five violation codes.

  ViolationCodeFreq_2017 <- summarize(groupBy(nyc_ticket_2017,
```

```
nyc_ticket_2017$Violation_Code), count = n(nyc_ticket_2017
$Violation_Code))
  head(arrange(ViolationCodeFreq_2017, desc(ViolationCodeFreq_
2017$count)),5)

  #  Violation_Code   count
  #1             21 1528588
  #2             36 1400614
  #3             38 1062304
  #4             14  893498
  #5             20  618593


  # 2. How often does each 'vehicle body type' get a parking
ticket? How about the 'vehicle make'?
  # (Hint: find the top 5 for both)
  BodyTypeFreq_2017 <- summarize(groupBy(nyc_ticket_2017,
nyc_ticket_2017$Vehicle_Body_Type), count = n(nyc_ticket_2017
$Vehicle_Body_Type))
  head(arrange(BodyTypeFreq_2017, desc(BodyTypeFreq_2017
$count)),5)

  #  Vehicle_Body_Type   count
  #1              SUBN 3719802
  #2              4DSD 3082020
  #3               VAN 1411970
  #4              DELV  687330
  #5               SDN  438191

  VehicleMakeFreq_2017 <- summarize(groupBy(nyc_ticket_2017,
nyc_ticket_2017$Vehicle_Make), count = n(nyc_ticket_2017
$Vehicle_Make))
  head(arrange(VehicleMakeFreq_2017, desc(VehicleMakeFreq_2017
$count)),5)

  #  Vehicle_Make   count
  #1         FORD 1280958
  #2        TOYOT 1211451
  #3        HONDA 1079238
  #4        NISSA  918590
  #5        CHEVR  714655

  # 3. A precinct is a police station that has a certain zone of
the city under its command.
  #
  # 3.1 Find the (5 highest) frequency of tickets for each of the
following:
  #
  # 3.2 'Violation Precinct' (this is the precinct of the zone
where the violation occurred). Using this, can you make any
insights
  # for parking violations in any specific areas of the city?
```

```
  VehiclePrecintFreq_2017 <- summarize(groupBy(nyc_ticket_2017,
nyc_ticket_2017$Violation_Precinct), count = n(nyc_ticket_2017
$Violation_Precinct))
  head(arrange(VehiclePrecintFreq_2017, desc(VehiclePrecintFreq_
2017$count)),5)

  #  Violation_Precinct   count
  #1                  0 2072400
  #2                 19  535671
  #3                 14  352450
  #4                  1  331810
  #5                 18  306920

  # There are many entries which have Violation Precinct as 0
which are incorrect. Post which 19 has highest number of parking
violations.


  data_rated_2017_5 <- SparkR::sql("SELECT Violation_County,
count(Violation_County) from SQL_nyc_ticket_2017  where
Violation_Precinct = 19 group by Violation_County sort by
Violation_County DESC")
  head(data_rated_2017_5)

  # Violation_County count(Violation_County)
  # 1                K                      16
  # 2                Q                       6
  # 3               BX                      16
  # 4             <NA>                       0
  # 5                R                      14
  # 6               NY                  532980

  # We can infer that maximum parking violations are in the area
of K county.

  data_rated_2017_4 <- SparkR::sql("SELECT Street_Code1, count
(Street_Code1) from SQL_nyc_ticket_2017  where Violation_Precinct
= 19 group by Street_Code1")
  head(arrange(data_rated_2017_4, desc(count(data_rated_2017_4
$Street_Code1))))

  #  Street_Code1 count(Street_Code1)
  #1        10210               73909
  #2        25390               60768
  #3        24890               48092
  #4        10010               45845
  #5        10110               35885
  #6        45590               22694

  # We can inder that Street Code 10210, 25390, 24890 and 10010
has high number of parking violations.
```

```
  # 3.3 'Issuer Precinct' (this is the precinct that issued the
ticket)
  # Here you would have noticed that the dataframe has 'Violating
Precinct' or 'Issuing Precinct' as '0'.
  # These are the erroneous entries. Hence, provide the record
for five correct precincts.
  # (Hint: print top six entries after sorting)


  data_rated_2017_6 <- SparkR::sql("SELECT Issuer_Precinct, count
(Issuer_Precinct) from SQL_nyc_ticket_2017 where Issuer_Precinct
!= 0 group by Issuer_Precinct")
  head(arrange(data_rated_2017_6, desc(count(data_rated_2017_6
$Issuer_Precinct))))

  #  Issuer_Precinct count(Issuer_Precinct)
  #1              19                 521513
  #2              14                 344977
  #3               1                 321170
  #4              18                 296553
  #5             114                 289950
  #6              13                 240833

  # We can infer that Issuer_Precinct 19 has issued highest
parking violation.

  # 4. Find the violation code frequency across three 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?
  # Hint: You can analyse the three precincts together using the
'union all' attribute in SQL view. In the SQL view,
  # use the 'where' attribute to filter among three precincts and
combine them using 'union all'.



  data_rated_2017_8<- SparkR::sql("SELECT Violation_Code, count
(Violation_Code) as counting from SQL_nyc_ticket_2017 where
Issuer_Precinct = 19  group by Violation_Code")
  head(arrange(data_rated_2017_8, desc(sum(data_rated_2017_8
$Violation_Code))))

  #  Violation_Code counting
  #1             46    86390
  #2             38    72344
  #3             37    72437
  #4             21    54700
  #5             71    15107
  #6             40    21513
```

```
  data_rated_2017_9 <- SparkR::sql("SELECT Violation_Code, count
(Violation_Code) as counting from SQL_nyc_ticket_2017 where
Issuer_Precinct = 14  group by Violation_Code")
  head(arrange(data_rated_2017_9, desc(sum(data_rated_2017_9
$Violation_Code))))

  #  Violation_Code counting
  #1             69    58026
  #2             47    30540
  #3             31    39857
  #4             14    73837
  #5             84    11111
  #6             42    20663



  data_rated_2017_10 <- SparkR::sql("SELECT Violation_Code, count
(Violation_Code) as counting from SQL_nyc_ticket_2017 where
Issuer_Precinct = 1  group by Violation_Code")
  head(arrange(data_rated_2017_10, desc(sum(data_rated_2017_10
$Violation_Code))))

  #  Violation_Code counting
  #1             46    22534
  #2             14    73522
  #3             69    11165
  #4             38    16989
  #5             16    38937
  #6             20    27841


  #

#-----------------------------------------------------------------
-----------------------------------------------------------------
-
#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.
#-----------------------------------------------------------------
-----------------------------------------------------------------
--


#year-2017
nyc_ticket_2017 <- withColumn(nyc_ticket_2017, "hours",
substr(nyc_ticket_2017$Violation_Time, 1, 2))
nyc_ticket_2017 <- withColumn(nyc_ticket_2017, "Period",
substr(nyc_ticket_2017$Violation_Time, 6, 6))
```

```
nyc_ticket_2017 <- withColumn(nyc_ticket_2017, "hours_bin",
ifelse(nyc_ticket_2017$Period == "P",

nyc_ticket_2017$hours + 12,

nyc_ticket_2017$hours))


#-----------------------------------------------------------------
-----------------------------------------------------------------
-
#Dealing  with missing values if
#-----------------------------------------------------------------
-----------------------------------------------------------------
-
#check of data and extract only valid data

str(nyc_ticket_2017)
nrow(nyc_ticket_2017)
#10803028
nrow(where(nyc_ticket_2017, nyc_ticket_2017$hours_bin <= 24))
#10802865 -- valid data
nrow(where(nyc_ticket_2017, nyc_ticket_2017$hours_bin >24))
#99     --|Errouneous data
nrow(where(nyc_ticket_2017, isNull(nyc_ticket_2017$hours)))
#63

#((99+63)/10803028)*100 = 0.001499666% of data which is very low
and can be omitted for further analysis
nyc_parking_violation_time_2017 <- subset(nyc_ticket_2017,
nyc_ticket_2017$hours_bin <= 24)

#Creating view for running SQL queries
createOrReplaceTempView(nyc_parking_violation_time_2017,
"data_violationtime_view_2017")



#-----------------------------------------------------------------
-----------------------------------------------------------------
----
# 5.3  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.
# Hint: Use the CASE-WHEN in SQL view to segregate into bins. For
finding the most commonly occurring violations,
# a similar approach can be used as mention in the hint for
question 4.

#-----------------------------------------------------------------
-----------------------------------------------------------------
```

```
----
# Binning into different hours and attaching data

#year-2017
bins_2017 <- SparkR::sql("SELECT Summons_Number,
Registration_State, Plate_Type, Issue_Date, Violation_Code,
Vehicle_Body_Type,
                    Vehicle_Make, Issuing_Agency,
Violation_Location, Violation_Precinct, Issuer_Precinct,
Violation_Time,
                    hours, Period,hours_bin,\
                    CASE  WHEN (hours_bin >= 0  and hours_bin <=
4 ) THEN 1\
                    WHEN (hours_bin >  4  and hours_bin <= 8 )
THEN 2\
                    WHEN (hours_bin >  8  and hours_bin <= 12)
THEN 3\
                    WHEN (hours_bin >  12 and hours_bin <= 16)
THEN 4\
                    WHEN (hours_bin >  16 and hours_bin <= 20)
THEN 5\
                    ELSE 6 END  as bin_number FROM
data_violationtime_view_2017")

# Attach the bin number to the original DataFrame
nyc_parking_violation_time_2017 <- withColumn(bins_2017,
"bin_number", bins_2017$bin_number)

#cross verifying structure and data
head(nyc_parking_violation_time_2017)
str(nyc_parking_violation_time_2017)

#-----------------------------------------------------------------
-----------------------------------------------------------------
-----
#Summary  and obtaining to most commonly occuring violation codes
#-----------------------------------------------------------------
-----------------------------------------------------------------
----
#year-2017
nyc_data_binning_2017 <- summarize(groupBy
(nyc_parking_violation_time_2017, nyc_parking_violation_time_2017
$bin_number,

nyc_parking_violation_time_2017$Violation_Code ),
                                   count_violation_code = n
(nyc_parking_violation_time_2017$Violation_Code))
nyc_data_binning_2017 <- arrange(nyc_data_binning_2017, desc
(nyc_data_binning_2017$count_violation_code))

# bin 1
head(where(nyc_data_binning_2017, nyc_data_binning_2017
```

```
$bin_number == 1), 3)
#  bin_number Violation Code count_violation_code
#1          1             38               454135
#2          1             36               394403
#3          1             37               339854


# bin 2
head(where(nyc_data_binning_2017, nyc_data_binning_2017
$bin_number == 2), 3)

#  bin_number Violation Code count_violation_code
#1          2             21               498762
#2          2             14               286449
#3          2             20               183214

# bin 3
head(where(nyc_data_binning_2017, nyc_data_binning_2017
$bin_number == 3), 3)
#  bin_number Violation Code count_violation_code
#1          3             21               950249
#2          3             36               826311
#3          3             38               431596


# bin 4
head(where(nyc_data_binning_2017, nyc_data_binning_2017
$bin_number == 4), 3)

#  bin_number Violation Code count_violation_code
#1          4             46                    4
#2          4             21                    4
#3          4             40                    3

# bin 5
head(where(nyc_data_binning_2017, nyc_data_binning_2017
$bin_number == 5), 3)
#  bin_number Violation Code count_violation_code
#1          5             78                    2
#2          5             98                    2
#3          5             40                    1


# bin 6
head(where(nyc_data_binning_2017, nyc_data_binning_2017
$bin_number == 6), 3)
# bin_number Violation Code count_violation_code
#1          6             14                    1
#2          6             78                    1
#3          6             40                    1
```

```
#-----------------------------------------------------------------
-----------------------------------------------------------------
5.4
#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)
#-----------------------------------------------------------------
-----------------------------------------------------------------
---
#Year:: 2017
#Top 3 Violation codes for year-2017 are 21,36,28

data_code_binning_2017 <- summarize(groupBy(subset
(nyc_parking_violation_time_2017, nyc_parking_violation_time_2017
$Violation_Code %in% c(21,36,38)),

nyc_parking_violation_time_2017$Violation_Code,

nyc_parking_violation_time_2017$bin_number ),
                                    count_in_bin = n
(nyc_parking_violation_time_2017$bin_number))


head(arrange(data_code_binning_2017, desc(data_code_binning_2017
$count_in_bin)))

#  Violation Code bin_number count_in_bin
#1             21          3       950249
#2             36          3       826311
#3             21          2       498762
#4             38          1       454135
#5             38          3       431596
#6             36          1       394403

#Observation for Year ::2017
# It looks like violation codes 21 and 36 mostly happens in bin
3,
#        which means these codes are mostly issed  between
morning 08:00 AM to 12:00PM
# violation code 38 happens largely in bin 4 which is from
12:00PM to 04:00PM

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

# 6.1 First, divide the year into some number of seasons, and
```

```
find frequencies of tickets for each season.
# (Hint: Use Issue Date to segregate into seasons)

# YEAR :: 2017
#Extract month  from issued  date values
parsed_2017_Month <- withColumn(nyc_ticket_2017, "Date_Parsed",
to_date(nyc_ticket_2017$Issue_Date, "MM/dd/yyyy"))
parsed_2017_Month <- withColumn(parsed_2017_Month, "Month", month
(parsed_2017_Month$Date_Parsed))

#checking for discripencies of data
nrow(where(parsed_2017_Month, parsed_2017_Month$Month <= 12 &
parsed_2017_Month$Month >= 1))
#10803028-- This valid data
nrow(where(parsed_2017_Month, parsed_2017_Month$Month <= 0 |
parsed_2017_Month$Month > 12))
#0  No Issue with this this data 0


nrow(where(parsed_2017_Month, isNull(parsed_2017_Month$Month)))
#0 #No Issue with this this data 0


#So all the data records are valid parsed_2017_Month can be  used
as  is
#Creating view for running SQL queries
createOrReplaceTempView(parsed_2017_Month, "ParsedDate_View_
2017")

data_bins_2017 <- SparkR::sql("SELECT Month,
Violation_Description, Violation_Code, \
                         CASE  WHEN (Month >=1  and Month <= 3)
THEN 1\
                         WHEN (Month > 3  and Month <= 6)  THEN 2
\
                         WHEN (Month > 6  and Month <= 9)  THEN 3
\
                         WHEN (Month > 9  and Month <= 12) THEN 4
\
                         ELSE 0 END  as bin_number FROM
ParsedDate_View_2017")

# Attach the bin number to the original DataFrame
nyc_parking_ParsedDate_2017 <- withColumn(data_bins_2017,
"bin_number", data_bins_2017$bin_number)


#-------------------------------------------------------------
-------------------------------------------------------------
#grouping and summarising Viloation  in bins
#-------------------------------------------------------------
-------------------------------------------------------------
```

```
# For year:: 2017
data_ParsedDate_2017 <- summarize(groupBy(nyc_parking_ParsedDate_
2017 ,
                                          nyc_parking_ParsedDate_
2017$bin_number),
                                  count_in_bin = n
(nyc_parking_ParsedDate_2017$bin_number))


#arranging the records in descending order
head(arrange(data_ParsedDate_2017, desc(data_ParsedDate_2017
$count_in_bin)))

#  bin_number count_in_bin
#1          2      3018840
#2          1      2671332
#3          4      2648920
#4          3      2463936

#Observation: Maximum count in bin 2 which is for 4-6(April -
June) of the year
#### So Season Septermber-decmeber has maximum Tickets.


# 6.2 Then, find the three most common violations for each of
these seasons.
# (Hint: A similar approach can be used as mention in the hint
for question 4.)

#-----------------------------------------------------------------
-------------------------------------------------------
#grouping and summarising data to obtain the viloation code count

#Then, find the 3 most common violations for each of these 4
seasons.
#-----------------------------------------------------------------
-------------------------------------------------------

data_ParsedDate_code2017 <- summarize(groupBy
(nyc_parking_ParsedDate_2017 ,

nyc_parking_ParsedDate_2017$bin_number,nyc_parking_ParsedDate_
2017$violation_code),
                                      count_of_code = n
(nyc_parking_ParsedDate_2017$violation_code))

data_ParsedDate_code2017  <- arrange(data_ParsedDate_code2017,
desc(data_ParsedDate_code2017$count_of_code))
# bin 1
head(where(data_ParsedDate_code2017, data_ParsedDate_code2017
$bin_number == 1),3)
     #  bin_number violation code count_of_code
```

```
     #1          1             21        374202
     #2          1             36        348240
     #3          1             38        287017

     # SO  Season-1 (Jan-March) have vialotation code 21, 36, 38

     # bin 2
     head(where(data_ParsedDate_code2017,
data_ParsedDate_code2017$bin_number == 2),3)
     # bin_number violation code count_of_code
     #1          2             21        421184
     #2          2             36        369902
     #3          2             38        266909

     # So Season-2 (Apr-June) have vialotation code 21, 36, 38


     # bin 3
     head(where(data_ParsedDate_code2017,
data_ParsedDate_code2017$bin_number == 3),3)
     #  bin_number violation code count_of_code
     #1          3             21        385774
     #2          3             38        244985
     #3          3             36        239879

     # So Season-3 (July-Sept) have vialotation code 21, 38, 36

     # bin 4
     head(where(data_ParsedDate_code2017,
data_ParsedDate_code2017$bin_number == 4),3)
     #  bin_number violation code count_of_code
     #1          4             36        442593
     #2          4             21        347428
     #3          4             38        263393

     # SO Season-4 (Oct-Dec) have vialotation code 36, 21, 38


#-----------------------------------------------------------------
-----------------------------------------------------------------
---
     #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.

#-----------------------------------------------------------------
-----------------------------------------------------------------
--------
```

```
     #year-2017
     data_violation_count_2017 <- summarize(groupBy
(nyc_parking_ParsedDate_2017, nyc_parking_ParsedDate_2017
$violation_code),
                                            count_violation_code
= n(nyc_parking_ParsedDate_2017$violation_code))

     head(arrange(data_violation_count_2017, desc
(data_violation_count_2017$count_violation_code)))
     #  violation code count_violation_code
     #1             21              1528588
     #2             36              1400614
     #3             38              1062304
     #4             14               893498
     #5             20               618593
     #6             46               600012


     #Observation :: for 3 most commonly occurring codes.
     # Violation code 21 (occurance - 1528588),
     # Viloation code 36 (occurance - 1400614),
     # viloation code 38 (occurance - 1062304) are the most
commonly occuring for year 2017



     #Not Let' search the internet
https://www1.nyc.gov/site/finance/vehicles/services-violation-
codes.page  for NYC parking violation code fines.
     #You will find a website (on the nyc.gov URL) that lists
these fines.
     #They’re divided into 2 categories,
# 1) one for the highest-density locations of the city,
# 2) other for the rest of the city.(Manhattan 96th St & below
v/s All other Area
                                   #For simplicity, take an
average of the two.

#-----------------------------------------------------------------
-----------------------------------------------------------------
----


#-----------------------------------------------------------------
-----------------------------------------------------------------
---------
                                   #Reading fine amount dataset
required for question-7 calcaultion.
                                   # Assumptions made about the
fine - total 98 violation codes used for analysis
                                   # for violation code 4 other
```

```
areas fine is zero. Average not used here.
                                   # for violation code 6 average
taken on 1st chance and second chance.
                                   # violation code 99 not
included as the fine amount would vary

#-----------------------------------------------------------------
-----------------------------------------------------------------
---------

                                   # parking_fine_amount <-
read.df("s3://data-science-buket1/casestudy/fine_amount.csv",
source = "csv",
                                   #
inferSchema = "true", header = "true")


                                   # As per 3.3
                                   # We can infer that
Issuer_Precinct 19 has issued highest parking violation.

                                   # Fine is taken from
https://www1.nyc.gov/site/finance/vehicles/services-violation-
codes.page
                                   # (We take average of the fine
both both Area for simplicty   (in USD )

                                   # Violation_Code
     Viloation_Count  Fine_per_Viloation_code
     Total_fine_Viloation_Code
                                   # 46
     108924                115
                12526260
                                   # 38
     89333          50
          4466650
                                   # 37
     72437          55
          3984035
                                   # 21
     54700          55
          3008500
                                   # 71
     15107          65
          981955
                                   # 40
     21513          115
          2473995
                                   # 69
     69191          65
          4497415
                                   # 47
```

```
     30540          115
               3512100
                                      # 31
     39857          115
               4583555
                                      # 14
     147359              115
                    16946285
                                      # 84
     11111          65
               722215
                                      # 42
     20663          55
               1136465
                                      # 16
     38937          95
               3699015
                                      # 20
     27841          62
               1726142


                                      #Observation:
                                      ### Total Fine COllected
                                      # 16946285 fine collected with
Vilation_code  14 in 2017 and ranks top most Fine collected


                                      #For violation code 99 fine
amount varies, so not included
                                      nrow(where(nyc_ticket_2017,
nyc_ticket_2017$violation_code == 99))
                                      # There are 3316 violation
codes with code 99.

                                      #joining parking data with
fine amount data obtained from NYC government site.
                                      #parking_fineamnt_data_2017 <-
join(nyc_ticket_2017, parking_fine_amount,
                                      #nyc_ticket_2017$`violation
code` == parking_fine_amount$violation_code, "left_outer")

#parking_fineamnt_data_2017 <- summarize(groupBy
(parking_fineamnt_data_2017, parking_fineamnt_data_2017
$`violation code`),
#Total_Fines_Collected = sum(parking_fineamnt_data_2017
$fine_amount))

#head(arrange(parking_fineamnt_data_2017, desc
(parking_fineamnt_data_2017$Total_Fines_Collected)))

################################################### END of 2017
```

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

#### Comparisiton between 2015, 2016 and 2017 #############

     # Number of tickets between various years
     # 2015 - 10951256
     # 2016 -  10626899
     # 2017 - 10803028
     # We can see that number of tickets are showing a reducing
trend
     #
     # The number of distinct registration state
     # 2015 - 69
     # 2016 - 68
     # 2017 - 67
     #
     # Tickets which do not have House number or street name
     # 2015 - 204880
     # 2016 - 174570
     # 2017 - 225065
     #
     # Top3 violation codes across the year
     # 2015 - 21, 38, 14
     # 2016 - 21, 36, 38
     # 2017 - 21, 36, 38
     # Violations codes 21 and 38 remain the top codes of
violation across the years.
     #
     # Top 2 BodyTypeFreq which gets parking tickets
     # 2015 - SUBN, 4DSD
     # 2016 - SUBN, 4DSD
     # 2017 - SUBN, 4DSD
     # SUBN and 4DSD remain the top 2 body type vehicles which
get the most number of partking tickets.
     #
     # Top2 VehicleMake which get parking tickets
     # 2015 - FORD, TOYOTA
     # 2016 - FORD, TOYOTA
     # 2017 - FORD, TOYOTA
     # Ford and Toyota remain the top2 vehicles that get most
parking tickets.
     #
     # Violation Precinct 19 is the most common precint from
which cars get tickets  and issue pricent 19 is the most common
pricent that issues tickets.
     #
     # The most number of violations done are between 8am to 12
noon for all 3 years
     #
     # In 2015, maximum tickets were issued in April-June.
     # In 2016, maximum tickets were issued in Oct-Dec
     # In 2017, maximum tickets were issued in April-June.
```

```
	#
	# 31637420 fine collected with Vilation_code  14 in 2015 and ranks top most Fine collected.
	# 18583885 fine collected with Violation_code  14 in 2016 and ranks top most Fine collected.
	# 16946285 fine collected with Vilation_code  14 in 2017 and ranks top most Fine collected.
	# Collection of fine with violatin code 14 is reducing.

sparkR.stop()
```