```
#
-------------------------------------------------------------------
-----------------------------------
# --                        NYC Parking Tickets: An Exploratory
Analysis using SparkR                      --
# --
--
# --                       Authors : Vijay Narayanan, Arunachalam
Meenakshisundaram,                      --
# --                                 Akash Ashokan and Dharmarajan
Thiagarajan                           --
#
-------------------------------------------------------------------
-----------------------------------

#
-------------------------------------------------------------------
-----------------------------------
# --                                      Problem Statement
--
# --  One of the biggest problems citizens of New York City face is
parking. The classic combination   --
# --  of a huge number of cars, and a cramped geography leads to a
huge number of parking tickets.     --
# --  NYPD have collected data for parking tickets from 2014 to
2017.                                      --
#
-------------------------------------------------------------------
-----------------------------------

#
-------------------------------------------------------------------
-----------------------------------
# --                                          Objectives
--
# -- Perform Exploratory Data Analysis on the parking tickets data
for years 2015, 2016 and 2017.      --
#
-------------------------------------------------------------------
-----------------------------------

library(dplyr)

# load SparkR
Sys.setenv(SPARK_HOME = "/usr/local/spark")
library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R",
"lib")))
sparkR.session(master = "yarn")

# Read the 2015 data file
path <- "/common_folder/nyc_parking/Parking_Violations_Issued_-
_Fiscal_Year_2015.csv"
nyc_parking_tickets_raw_data_2015 <- read.df(path, source = "CSV",
header = "true", inferSchema = "true")
```

```
# Examine data
nrow(nyc_parking_tickets_raw_data_2015)
# Total parking records were 11809233

ncol(nyc_parking_tickets_raw_data_2015)
# Dataset contained 51 variables

str(nyc_parking_tickets_raw_data_2015)
# 'SparkDataFrame': 51 variables:
# $ Summons Number                   : num 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                   : int 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                     : int 20390 27790 8130 59990
36090 74230
# $ Street Code2                     : int 29890 19550 5430 16540
10410 37980
# $ Street Code3                     : int 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               : int 7 25 72 102 28 67
# $ Violation Precinct               : int 7 25 72 102 28 67
# $ Issuer Precinct                  : int 7 25 72 102 28 67
# $ Issuer Code                      : int 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                      : int 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?            : int NA NA NA NA NA NA
# $ Vehicle Year                     : int 2005 0 2010 2015 0 0
# $ Meter Number                     : chr "NA" "NA" "NA" "NA" "NA"
"NA"
# $ Feet From Curb                   : int 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"

head(nyc_parking_tickets_raw_data_2015)
# Summons Number Plate ID Registration State Plate Type Issue Date
Violation Code
# 1     8002531292  EPC5238                NY        PAS 10/01/2014
21
# 2     8015318440   5298MD                NY        COM 03/06/2015
```

```
14
# 3     7611181981  FYW2775                 NY        PAS 07/28/2014
46
# 4     7445908067  GWE1987                 NY        PAS 04/13/2015
19
# 5     7037692864 T671196C                 NY        PAS 05/19/2015
19
# 6     7704791394  JJF6834                 PA        PAS 11/20/2014
21
# Vehicle Body Type Vehicle Make Issuing Agency Street Code1 Street
Code2 Street Code3
# 1              SUBN        CHEVR              T        20390
29890        31490
# 2               VAN        FRUEH              T        27790
19550        19570
# 3              SUBN        SUBAR              T         8130
5430         5580
# 4              4DSD        LEXUS              T        59990
16540        16790
# 5              4DSD        CHRYS              T        36090
10410        24690
# 6              4DSD        NISSA              T        74230
37980        38030
# Vehicle Expiration Date Violation Location Violation Precinct
Issuer Precinct Issuer Code
# 1 01/01/20150111 12:00:00 PM                  7                  7
7      345454
# 2 01/01/88888888 12:00:00 PM                 25                 25
25      333386
# 3 01/01/20160524 12:00:00 PM                 72                 72
72      331845
# 4 01/01/20170111 12:00:00 PM                102                102
102      355669
# 5 01/01/88888888 12:00:00 PM                 28                 28
28      341248
# 6 01/01/20150688 12:00:00 PM                 67                 67
67      357104
# Issuer Command Issuer Squad Violation Time Time First Observed
Violation County
# 1           T800           A2          0011A                <NA>
NY
# 2           T103            B          0942A                <NA>
NY
# 3           T302            L          1020A                <NA>
K
# 4           T402            D          0318P                <NA>
Q
# 5           T103            X          0410P                <NA>
NY
# 6           T302            A          0839A                <NA>
K
# Violation In Front Of Or Opposite House Number Street Name
Intersecting Street
#  1                                F          133    Essex St
```

<NA>
2   F   1916   Park Ave
<NA>
3   F   184   31st St
<NA>
4   F   120-20 Queens Blvd
<NA>
5   F   66 W 116th St
<NA>
6   F   1013 Rutland Rd
<NA>
Date First Observed Law Section Sub Division Violation Legal Code
1 01/05/0001 12:00:00 PM   408   d1
<NA>
2 01/05/0001 12:00:00 PM   408   c
<NA>
3 01/05/0001 12:00:00 PM   408   f1
<NA>
4 01/05/0001 12:00:00 PM   408   c3
<NA>
5 01/05/0001 12:00:00 PM   408   c3
<NA>
6 01/05/0001 12:00:00 PM   408   d1
<NA>
Days Parking In Effect   From Hours In Effect To Hours In
Effect Vehicle Color
1   Y Y Y   1200A
0300A   BL
2   YYYYY   0700A
1000A   BROWN
3   <NA>   <NA>
<NA>   BLACK
4   YYYYY   0300P
1000P   GY
5   YYYYYYY   <NA>
<NA>   BLACK
6   Y   0830A
0900A   WHITE
Unregistered Vehicle? Vehicle Year Meter Number Feet From Curb
Violation Post Code
1   NA   2005   <NA>   0
A 77
2   NA   0   <NA>   0
CC3
3   NA   2010   <NA>   0
J 32
4   NA   2015   <NA>   0
01 4
5   NA   0   <NA>   0
19 7
6   NA   0   <NA>   0
C 32
Violation Description No Standing or Stopping Violation Hydrant
Violation

```
#  1 21-No Parking (street clean)                                  <NA>
<NA>
#  2               14-No Standing                                  <NA>
<NA>
#  3 46A-Double Parking (Non-COM)                                  <NA>
<NA>
#  4       19-No Stand (bus stop)                                  <NA>
<NA>
#  5       19-No Stand (bus stop)                                  <NA>
<NA>
#  6 21-No Parking (street clean)                                  <NA>
<NA>
#  Double Parking Violation Latitude Longitude Community Board
Community Council  Census Tract
#  1                      <NA>     <NA>      <NA>            <NA>
<NA>         <NA>
#  2                      <NA>     <NA>      <NA>            <NA>
<NA>         <NA>
#  3                      <NA>     <NA>      <NA>            <NA>
<NA>         <NA>
#  4                      <NA>     <NA>      <NA>            <NA>
<NA>         <NA>
#  5                      <NA>     <NA>      <NA>            <NA>
<NA>         <NA>
#  6                      <NA>     <NA>      <NA>            <NA>
<NA>         <NA>
#  BIN  BBL  NTA
#  1 <NA> <NA> <NA>
#  2 <NA> <NA> <NA>
#  3 <NA> <NA> <NA>
#  4 <NA> <NA> <NA>
#  5 <NA> <NA> <NA>
#  6 <NA> <NA> <NA>

printSchema(nyc_parking_tickets_raw_data_2015)
# root
# |-- 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)

#
------------------------------------------------------------------------
----------------------------------
# --                                   Analysis for 2015
--
#
------------------------------------------------------------------------
----------------------------------

# Rename Issue Date and Summons Number columns
nyc_parking_tickets_raw_data_2015 <-
withColumnRenamed(nyc_parking_tickets_raw_data_2015, "Issue Date",
"Issue_Date")
nyc_parking_tickets_raw_data_2015 <-
withColumnRenamed(nyc_parking_tickets_raw_data_2015, "Summons
Number", "Summons_Number")

# Create a Temp View of Parking Tickets 2015 Data Frame for
performing SQL operations
createOrReplaceTempView(nyc_parking_tickets_raw_data_2015,
```

```
"nyc_parking_tickets_2015_df_view")

# Check if there are any Data Quality Issues?

# 1, Does the file contain parking tickets only issued in 2015?
recs_by_issue_date <- SparkR::sql("select
distinct(substring(Issue_Date, -4)) as Year_Of_Issue from
nyc_parking_tickets_2015_df_view order by Year_Of_Issue")
showDF(recs_by_issue_date, 100, FALSE)
# Although the file says 2015 but it can be seen that there are
parking tickets issued from other years
# such as 1985, 1986. 1988, 1991, 2000 ... 2010, 2011, 2012, 2015

#
-----------------------------------------------------------------
-----------------------------------
# Assumption:
# As this analysis is for the year 2015, parking tickets only
pertaining to 2015 are considered
#
-----------------------------------------------------------------
-----------------------------------

# Filter out parking tickets issued from other years and only retain
for year 2015
nyc_parking_tickets_only_2015 <- SparkR::sql("select * from
nyc_parking_tickets_2015_df_view where substring(Issue_Date, -4) =
2015")

head(nyc_parking_tickets_only_2015)
# ID Registration State Plate Type Issue_Date Violation Code Vehicle
Body Type
# 1     8015318440   5298MD                NY        COM 03/06/2015
14              VAN
# 2     7445908067  GWE1987                NY        PAS 04/13/2015
19             4DSD
# 3     7037692864 T671196C                NY        PAS 05/19/2015
19             4DSD
# 4     8017159560  GKX8095                NY        PAS 01/20/2015
71             4DSD
# 5     8017159560  GKX8095                NY        PAS 01/20/2015
71             4DSD
# 6     7002571382  CXT8949                TX        PAS 02/17/2015
20             SUBN
# Vehicle Make Issuing Agency Street Code1 Street Code2 Street Code3
Vehicle Expiration Date
# 1        FRUEH              T        27790        19550
19570 01/01/88888888 12:00:00 PM
# 2        LEXUS              T        59990        16540
16790 01/01/20170111 12:00:00 PM
# 3        CHRYS              T        36090        10410
24690 01/01/88888888 12:00:00 PM
# 4        LEXUS              T        35490        35780
22670 01/01/20151207 12:00:00 PM
```

```
# 5        LEXUS              T        35490        35780
22670 01/01/20151207 12:00:00 PM
# 6        MAZDA              T        51190         9140
61090 01/01/88880088 12:00:00 PM
# Violation Location Violation Precinct Issuer Precinct Issuer Code
Issuer Command Issuer Squad Violation Time
# 1                  25                  25              25
333386           T103            B          0942A
# 2                 102                 102             102
355669           T402            D          0318P
# 3                  28                  28              28
341248           T103            X          0410P
# 4                 113                 113             113
361082           T402            R          0259P
# 5                 113                 113             113
361082           T402            R          0259P
# 6                 109                 109             109
359625           T401            G          0459P
# Time First Observed Violation County Violation In Front Of Or
Opposite House Number Street Name
# 1                <NA>               NY
F         1916    Park Ave
# 2                <NA>                Q
F       120-20 Queens Blvd
# 3                <NA>               NY
F           66  W 116th St
# 4                <NA>                Q
F       137-22   Bedell St
# 5                <NA>                Q
F       137-22   Bedell St
# 6                <NA>                Q
O        39-15    Janet Pl
# Intersecting Street    Date First Observed Law Section Sub
Division Violation Legal Code
#  1                <NA> 01/05/0001 12:00:00 PM         408
c                 <NA>
#  2                <NA> 01/05/0001 12:00:00 PM         408
c3                <NA>
#  3                <NA> 01/05/0001 12:00:00 PM         408
c3                <NA>
#  4                <NA> 01/05/0001 12:00:00 PM         408
j6                <NA>
#  5                <NA> 01/05/0001 12:00:00 PM         408
j6                <NA>
#  6                <NA> 01/05/0001 12:00:00 PM         408
d                 <NA>
#  Days Parking In Effect     From Hours In Effect To Hours In
Effect Vehicle Color Unregistered Vehicle?
# 1                     YYYYY                0700A
1000A         BROWN                    NA
# 2                     YYYYY                0300P
1000P            GY                    NA
# 3                   YYYYYYY                 <NA>
<NA>          BLACK                    NA
```

```
# 4                    YYYYYYY                 <NA>
<NA>         GREEN                    NA
# 5                    YYYYYYY                 <NA>
<NA>         GREEN                    NA
# 6                      YYYYY               0800A
0600P         WHITE                     NA
# Vehicle Year Meter Number Feet From Curb Violation Post Code
Violation Description
# 1            0         <NA>              0                 CC3
14-No Standing
# 2         2015         <NA>              0               01 4
19-No Stand (bus stop)
# 3            0         <NA>              0               19 7
19-No Stand (bus stop)
# 4         1993         <NA>              0               N 42
71A-Insp Sticker Expired (NYS)
# 5         1993         <NA>              0               N 42
71A-Insp Sticker Expired (NYS)
# 6            0         <NA>              0               17 4
20A-No Parking (Non-COM)
# No Standing or Stopping Violation Hydrant Violation Double Parking
Violation Latitude Longitude
#  1                              <NA>              <NA>
<NA>     <NA>      <NA>
#  2                              <NA>              <NA>
<NA>     <NA>      <NA>
#  3                              <NA>              <NA>
<NA>     <NA>      <NA>
#  4                              <NA>              <NA>
<NA>     <NA>      <NA>
#  5                              <NA>              <NA>
<NA>     <NA>      <NA>
#  6                              <NA>              <NA>
<NA>     <NA>      <NA>
#  Community Board Community Council  Census Tract  BIN  BBL  NTA
#  1            <NA>               <NA>         <NA> <NA> <NA> <NA>
#  2            <NA>               <NA>         <NA> <NA> <NA> <NA>
#  3            <NA>               <NA>         <NA> <NA> <NA> <NA>
#  4            <NA>               <NA>         <NA> <NA> <NA> <NA>
#  5            <NA>               <NA>         <NA> <NA> <NA> <NA>
#  6            <NA>               <NA>         <NA> <NA> <NA> <NA>

nrow(nyc_parking_tickets_only_2015)
# 5986831

# Create a Temp of Parking Tickets with Only 2015 records
createOrReplaceTempView(nyc_parking_tickets_only_2015,
"nyc_parking_tickets_only_2015_df_view")

# 2. Check if all tickets have a Summons Number
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_only_2015_df_view where Summons_Number is null
or Summons_Number in ('', 'NA')"))
#Count
```

```
# 0
# All records have a Summons Number for parking ticket

# 3. Are there any duplicate parking tickets i.e duplicate Summons
Number
head(SparkR::sql("select Summons_Number as Summons_Number,
Issue_date as Issue_Date, count(*) as Count from
nyc_parking_tickets_only_2015_df_view group by Summons_Number,
Issue_Date having count(*) > 1"))
# Summons_Number Issue_Date Count
#     7535505545 01/22/2015     2
#     7663790881 01/09/2015     2
#     7130792528 01/15/2015     2
#     7093099221 01/15/2015     2
#     7611240808 01/14/2015     2
#     7563956165 01/26/2015     2
# There were several duplicate parking tickets issued in the same
year 2015

#
-----------------------------------------------------------------
-----------------------------------
# Assumption:
# Duplicate parking tickets are in the dataset by mistake. These
will be removed for further analysis.
#
-----------------------------------------------------------------
-----------------------------------

# Remove duplicate parking ticket rows from dataset
nyc_parking_tickets_2015 <-
dropDuplicates(nyc_parking_tickets_only_2015, "Summons_Number")
nrow(nyc_parking_tickets_2015)
# 5373971

# Rename Registration State to Registration_State
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Registration State",
"Registration_State")
# Rename Plate ID to Plate_ID
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Plate ID", "Plate_ID")
# Rename Violation Code to Violation_Code
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Violation Code",
"Violation_Code")

# Create a Temp of Parking Tickets with duplicate records removed
createOrReplaceTempView(nyc_parking_tickets_2015,
"nyc_parking_tickets_2015_df_view")

# Check if duplicate records have been removed. For example check
only one row of Summons Number = 7535505545 exists in the data frame
head(SparkR::sql("select count(*) as Count from
```

```
nyc_parking_tickets_2015_df_view where Summons_Number =
'7535505545'"))
#  Count
#  1

# 4. Are there missing values for Issue Date?
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2015_df_view where Issue_Date is null or
Issue_Date in ('', 'NA')"))
# Count
# 0
# All parking tickets have an issue date and no rows have a missing
value

# 5. Is Registration State of vehicle missing in any parking
tickets?
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2015_df_view where Registration_State is null or
Registration_State in ('', 'NA')"))
# Count
# 0
# All parking tickets have a Registration State of car

# 6. Is Plate ID of vehicle missing in any parking tickets?
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2015_df_view where Plate_ID is null or Plate_ID
in ('', 'NA')"))
# Count
# 15
# There were 15 parking tickets that did not have Plate ID of a
vehicle
# As the number is insignificant these rows are retained

# 7. Is Violation Code missing in any parking tickets?
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2015_df_view where Violation_Code is null or
Violation_Code in ('', 'NA')"))
# Count
#  0
# All parking tickets have Violation Code

#
-----------------------------------------------------------------------
-----------------------------------
#                                Questions to be answered in the
analysis for 2015
#
-----------------------------------------------------------------------
-----------------------------------

# ----------------------------#
#       Examine the data      #
# ----------------------------#
```

```
# 1. Find total number of tickets for the year
head(SparkR::sql("select count(*) from
nyc_parking_tickets_2015_df_view"))
# 5373971

# 2. Find out how many unique states the cars which got parking
tickets came from
head(SparkR::sql("select count(distinct(Registration_State)) as
Count from nyc_parking_tickets_2015_df_view"))
# Count
# 68
# Cars from 68 states received parking tickets in 2015
states_df <- SparkR::sql("select distinct(Registration_State) from
nyc_parking_tickets_2015_df_view")
showDF(states_df, 100, FALSE)
#  +------------------+
#  |Registration_State|
#  +------------------+
#  |AZ                |
#  |SC                |
#  |NS                |
#  |LA                |
#  |MN                |
#  |NJ                |
#  |MX                |
#  |DC                |
#  |OR                |
#  |99                |
#  |NT                |
#  |VA                |
#  |RI                |
#  |KY                |
#  |WY                |
#  |BC                |
#  |NH                |
#  |MI                |
#  |GV                |
#  |NV                |
#  |QB                |
#  |WI                |
#  |ID                |
#  |CA                |
#  |CT                |
#  |NE                |
#  |MT                |
#  |NC                |
#  |VT                |
#  |MD                |
#  |DE                |
#  |MO                |
#  |IL                |
#  |ME                |
#  |MB                |
#  |WA                |
```

```
#  |ND                |
#  |MS                |
#  |IN                |
#  |AL                |
#  |OH                |
#  |TN                |
#  |NM                |
#  |IA                |
#  |PA                |
#  |SD                |
#  |FO                |
#  |NY                |
#  |ON                |
#  |SK                |
#  |AB                |
#  |PE                |
#  |TX                |
#  |WV                |
#  |GA                |
#  |MA                |
#  |KS                |
#  |FL                |
#  |CO                |
#  |AK                |
#  |AR                |
#  |NB                |
#  |OK                |
#  |PR                |
#  |NF                |
#  |UT                |
#  |DP                |
#  |HI                |
#  +------------------+
# Dataset shows that cars from 50 States of USA, 17 States of Canada
received parking tickets.
# There was 1 state with value 99

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

#
-------------------------------------------------------------------
-----------------------------------
# Assumption:
# Address can be of two types,
# 1. Address where the violation occurred and
# 2. Address where the vehicle is registered
#
-------------------------------------------------------------------
-----------------------------------

# 1. Address where violation occurred
# Rename Street Code1 to Street_Code1
```

```
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Street Code1",
"Street_Code1")
# Rename Street Code2 to Street_Code2
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Street Code2",
"Street_Code2")
# Rename Street Code3 to Street_Code3
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Street Code3",
"Street_Code3")
# Rename Violation Location to Violation_Location
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Violation Location",
"Violation_Location")
# Rename Intersecting Street to Intersecting_Street
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Intersecting Street",
"Intersecting_Street")
# Rename Violation Post Code to Violation_Post_Code
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Violation Post Code",
"Violation_Post_Code")

# 2. Address where the vehicle is registered
# Rename House Number to House_Number
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "House Number",
"House_Number")
# Rename Street Name to Street_Name
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Street Name",
"Street_Name")

createOrReplaceTempView(nyc_parking_tickets_2015,
"nyc_parking_tickets_clean_2015_df_view")

# Parking tickets with missing Address where the violation occurred
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_clean_2015_df_view where Street_Code1 is null or
Street_Code2 is null or Street_Code3 is null or
                 Violation_Location is null or Intersecting_Street
is null or Violation_Post_Code is null"))
# Count
# 4729370
# There were 4729370 parking tickets that were missing address where
violation occurred

# Parking tickets with missing Address of where the vehicle is
registered
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_clean_2015_df_view where House_Number is null or
House_Number in ('', 'NA') or Street_Name is null or Street_Name in
('', 'NA')"))
```

```
# Count
# 799017
# There were 799017 parking tickets that either does not have a
House Number or missing a Street Name

# So, a total of 5528387 parking tickets had incomplete address

# ----------------------------#
#      Aggregation tasks       #
# ----------------------------#

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

violation_code_counts_2015 <-
summarize(groupBy(nyc_parking_tickets_2015,
nyc_parking_tickets_2015$Violation_Code), Count =
n(nyc_parking_tickets_2015$Violation_Code))
head(arrange(violation_code_counts_2015,
desc(violation_code_counts_2015$count)), n = 5)
# Violation_Code  Count
# 21              720902
# 38              663904
# 14              466488
# 36              406249
# 37              373229
# Top 5 commonly occurring violation codes were 21, 38, 14, 36 and
37

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

# Rename Vehicle Body Type to Vehicle_Body_Type
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Vehicle Body Type",
"Vehicle_Body_Type")
# Rename Vehicle Make to Vehicle_Make
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Vehicle Make",
"Vehicle_Make")

vehicle_body_type_counts_2015 <-
summarize(groupBy(nyc_parking_tickets_2015,
nyc_parking_tickets_2015$Vehicle_Body_Type),
                                            Count =
n(nyc_parking_tickets_2015$Vehicle_Body_Type))
head(arrange(vehicle_body_type_counts_2015,
desc(vehicle_body_type_counts_2015$count)), n = 5)
# Vehicle_Body_Type   Count
# SUBN                1715517
# 4DSD                1514580
# VAN                 795457
# DELV                419548
# SDN                 209381
```

```
# Suburban, 4 Door Sedan and Vans were the vehicle types that
received maximum parking tickets

vehicle_make_counts_2015 <-
summarize(groupBy(nyc_parking_tickets_2015,
nyc_parking_tickets_2015$Vehicle_Make),
                                      Count =
n(nyc_parking_tickets_2015$Vehicle_Make))
head(arrange(vehicle_make_counts_2015,
desc(vehicle_make_counts_2015$count)), n = 5)
#  Vehicle_Make  Count
#  FORD          685900
#  TOYOT         554392
#  HONDA         498858
#  NISSA         411857
#  CHEVR         404841
# FORD, TOYOTA and HONDA vehicles received the most number of
parking tickets.

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

# Renaming Violation Precinct to Violation_Precinct
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Violation Precinct",
"Violation_Precinct")
# Renaming Issuer Precinct to Issuer_Precinct
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Issuer Precinct",
"Issuer_Precinct")

violation_precinct_counts_2015 <-
summarize(groupBy(nyc_parking_tickets_2015,
nyc_parking_tickets_2015$Violation_Precinct),
                                           Count =
n(nyc_parking_tickets_2015$Violation_Precinct))
head(arrange(violation_precinct_counts_2015,
desc(violation_precinct_counts_2015$count)), n = 5)
# Violation_Precinct  Count
# 0                   721275
# 19                  287403
# 14                  197011
# 18                  193593
# 1                   127483

#
---------------------------------------------------------------------
----------------------------------
# Assumption:
```

```
# Precinct 0 is not a valid zone and does not appear in the NYPD
precincts list available on
# https://www1.nyc.gov/site/nypd/bureaus/patrol/precincts-
landing.page
# It could be that Precinct 0 refers to an incorrect value. So,
ignoring Precinct 0 although it has the
# highest count
#
-------------------------------------------------------------------
-----------------------------------
# Zone 19 has the next maximum number of parking tickets. The 19th
Precinct command serves the Upper East Side of Manhattan.
# Zone 14 is Manhattan Midtown South
# Zone 114 is northwestern portion of Queens
# Zone 18 is Manhattan Midtown North
# Zones in Manhattan (Upper East, Midtown North and South) and
Northwest Queens have had the maximum number of parking tickets
issued in 2015

issuer_precinct_counts_2015 <-
summarize(groupBy(nyc_parking_tickets_2015,
nyc_parking_tickets_2015$Issuer_Precinct),
                                          Count =
n(nyc_parking_tickets_2015$Issuer_Precinct))
head(arrange(issuer_precinct_counts_2015,
desc(issuer_precinct_counts_2015$count)), n = 5)
# Issuer_Precinct  Count
# 0                828570
# 19               279931
# 14               190403
# 18               190337
# 114              149532
# Ignoring Issuer Precinct 0 as it appears to be an invalid valid
# Police Stations of Manhattan (Upper East, Midtown North and South)
and Northwest Queens have issued the most number of
# parking tickets in 2015

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

# Renaming Violation Code to Violation_Code
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Violation Code",
"Violation_Code")

createOrReplaceTempView(nyc_parking_tickets_2015,
"nyc_parking_tickets_2015_df_view")

# Violcation codes for precincts (19,14,18) that have issued the
most number of tickets
violation_codes_for_issuer_precincts <- SparkR::sql("select
```

```
Issuer_Precinct, Violation_Code, count(*) as Count from
nyc_parking_tickets_2015_df_view
                                                    where
Issuer_Precinct in (19,14,18) group by Issuer_Precinct,
Violation_Code")

head(arrange(violation_codes_for_issuer_precincts,desc(violation_cod
es_for_issuer_precincts$count)), n = 10)
# Issuer_Precinct Violation_Code  Count
# 18              14              59616
# 19              38              45647
# 14              69              41004
# 9               37              40665
# 14              14              38696
# 19              14              31295
# 19              16              29738
# 18              69              28149
# 19              46              27049
# 19              21              25916
# Precinct Zone 18 had the highest frequency of violation code 14
# Precinct Zone 19 had violation code 38 as the most occurring
# Precinct Zone 14 had violation code 69 as the most frequently
occurring
# Yes, there are violation codes such as 14, 69 that occur commonly
across Precincts

common_violation_codes <- SparkR::sql("select Violation_Code,
count(*) as Count from nyc_parking_tickets_2015_df_view
                                      where Issuer_Precinct in
(19,14,18) group by Violation_Code")

head(arrange(common_violation_codes,desc(common_violation_codes
$count)), n = 5)
# Violation_Code Count
# 14             129607
# 69             71598
# 38             58288
# 37             46583
# 46             39829
# Violation code 14 had a very high frequency
# Violation codes 14, 69 and 38 were the top 3 most commonly
occuring violation codes in Zones 19, 14 and 18

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

# Renaming Violation Time to Violation_Time
nyc_parking_tickets_2015 <-
withColumnRenamed(nyc_parking_tickets_2015, "Violation Time",
"Violation_Time")

createOrReplaceTempView(nyc_parking_tickets_2015,
"nyc_parking_tickets_2015_df_view")

# Determine if there are any missing values for Violation_Time
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2015_df_view where Violation_Time is null or
Violation_Time in ('na', '')"))
# 664 parking tickets issued have missing Violation Time
# There are very few records i.e 644 out of 5986831 with missing
Violation Time.
# So, ignoring these records from analysis as the number is
insignificant

# Check for Time consistency
head(SparkR::sql("select Violation_Time from
nyc_parking_tickets_2015_df_view where substring(Violation_Time, 1,
2) = '12' and substring(Violation_Time, -1) = 'A'"))
#Violation_Time
# 1201A
# 1230A
# 1251A
# 1204A
# 1200A
# 1203A
# There are many parking tickets that have time recorded with 12:nn
AM hours. These records will be binned
# along with 00 AM hours.

head(SparkR::sql("select Violation_Time from
nyc_parking_tickets_2015_df_view where substring(Violation_Time, 1,
2) = '03' and substring(Violation_Time, -1) = 'P'"))
#Violation_Time
# 0346P
# 0320P
# 0328P
# 0310P
# 0314P
# 0326P
# It can be seen that a proper 24 Hour Time convention was not been
followed. So, care must be taken whilst binning.

# Are there any records with Violation Time length greater than or
lesser than 5
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2015_df_view where length(Violation_Time) > '5'
or length(Violation_Time) < '5'"))
# Count
```

```
# 0
# There were no rows with invalid time length i.e <5 or >5

# Are there any records with Violation Time not in A or P
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2015_df_view where
upper(substring(Violation_Time, -1)) not in ('A', 'P')"))
# Count
# 0
# There were no parking tickets that are neither A or P

# Are there any records with Violation Time not in the 24 hour time
window
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2015_df_view where substring(Violation_Time, 1,
2) not in
('00','01','02','03','04','05','06','07','08','09','10','11','12','1
3','14','15','16','17','18','19','20','21','22','23')"))
# Count
#  68
# There were 68 parking tickets that had invalid time and these
records will be excluded

# Create 6 bins of 24 hour time period
time_bins_sql_2015 <- "select case when substring(Violation_Time,
1,2) in ('00','01','02','03','12') and
upper(substring(Violation_Time,-1)) = 'A' then 'Bin 1'
when substring(Violation_Time,1,2) in ('04','05','06','07') and
upper(substring(Violation_Time,-1)) = 'A' then 'Bin 2'
when substring(Violation_Time,1,2) in ('08','09','10','11') and
upper(substring(Violation_Time,-1)) = 'A' then 'Bin 3'
when substring(Violation_Time,1,2) in
('12','13','14','15','00','01','02','03') and
upper(substring(Violation_Time,-1)) = 'P' then 'Bin 4'
when substring(Violation_Time,1,2) in
('16','17','18','19','04','05','06','07') and
upper(substring(Violation_Time,-1)) = 'P' then 'Bin 5'
when substring(Violation_Time,1,2) in
('20','21','22','23','08','09','10','11') and
upper(substring(Violation_Time,-1)) = 'P' then 'Bin 6'
else null
end as Violation_time_bin, Violation_Code, Violation_Time
from nyc_parking_tickets_2015_df_view where Violation_Time is not
null and
substring(Violation_Time, 1, 2) in
('00','01','02','03','04','05','06','07','08','09','10','11','12','1
3','14','15','16','17','18','19','20','21','22','23')"

violations_time_bins_2015 <- SparkR::sql(time_bins_sql_2015)
head(violations_time_bins_2015)
# Violation_time_bin  Violation_Code Violation_Time
# Bin 5               70             0520P
# Bin 2               74             0443A
# Bin 1               71             0208A
```

```
# Bin 1               71             0127A
# Bin 1               71             0142A
# Bin 6               46             0839P

createOrReplaceTempView(violations_time_bins_2015,
"violations_time_bins_2015_df_view")

violation_code_count_in_time_bins_2015 <- SparkR::sql("select
Violation_time_bin, Violation_Code, count(*) Count from
violations_time_bins_2015_df_view
                                                    group by
Violation_time_bin, Violation_Code")

# Use Collect action to get results in df to driver node for faster
aggregation
violation_code_count_coll_2015 <-
SparkR::collect(violation_code_count_in_time_bins_2015)

getTop3ViolationCodesInTimeBins <- function(bin) {
  dplyr::filter(violation_code_count_coll_2015, Violation_time_bin
== bin) %>% dplyr::arrange(desc(Count)) %>% head(n = 3)
}

# Get top 3 Violation Codes in Bin 1 ('00','01','02','03','12') AM
getTop3ViolationCodesInTimeBins('Bin 1')
# Violation_time_bin Violation_Code Count
# Bin 1              21             30663
# Bin 1              40             20613
# Bin 1              78             17198
# Stopping closer to 15 feet of fire hydrant is common during very
early mornings.

# Get top 3 Violation Codes in Bin 2 ('04','05','06','07') AM
getTop3ViolationCodesInTimeBins('Bin 2')
# Violation_time_bin Violation_Code Count
# Bin 2              14             68994
# Bin 2              21             49098
# Bin 2              40             46783

# Get top 3 Violation Codes in Bin 3 ('08','09','10','11') AM
getTop3ViolationCodesInTimeBins('Bin 3')
# Violation_time_bin Violation_Code Count
# Bin 3              21             573741
# Bin 3              38             235956
# Bin 3              36             189347
# Violations were high between 8 and 11 AM. No parking where not
allowed, Failing to show a receipt and exceeding allowed time
# are the most common reasons

# Get top 3 Violation Codes in Bin 4
('12','13','14','15','00','01','02','03') PM
getTop3ViolationCodesInTimeBins('Bin 4')
# Violation_time_bin Violation_Code Count
# Bin 4              38             287564
```

```
# Bin 4              37             212536
# Bin 4              36             177439

# Get top 3 Violation Codes in Bin 5
('16','17','18','19','04','05','06','07') PM
getTop3ViolationCodesInTimeBins('Bin 5')
# Violation_time_bin Violation_Code Count
# Bin 5              38             111178
# Bin 5              37             83676
# Bin 5              14             73424
# Violations were the highest between 4 and 7 PM
# Parking in excess of the allowed time or failing to show a receipt
and parking where it is not allowed
# are the most common reasons for receiving parking ticket during
these hours

# Get top 3 Violation Codes in Bin 6
('20','21','22','23','08','09','10','11') PM
getTop3ViolationCodesInTimeBins('Bin 6')
# Violation_time_bin Violation_Code Count
# Bin 6              7              29936
# Bin 6              38             27571
# Bin 6              40             22491
# Going through red light at an intersection is the most common
violation after 10 PM.
# Stopping closer to 15 feet of fire hydrant is also common during
late nights and early mornings.

# Three most commonly occurring Violation codes
most_popular_violation_codes_2015 <-
summarize(groupBy(violations_time_bins_2015,
violations_time_bins_2015$Violation_Code),
                                                Count =
n(violations_time_bins_2015$Violation_Code))
head(arrange(most_popular_violation_codes_2015,
desc(most_popular_violation_codes_2015$Count)), n = 3)
# Violation_Code  Count
# 21             720890
# 38             663904
# 36             466479
# Violation codes 21, 38 and 36 were the top 3 commonly occurring
violations

# Get time bins of most commonly occuring violations
filtered_violation_codes_2015 <-
dplyr::filter(violation_code_count_coll_2015,
violation_code_count_coll_2015$Violation_Code %in% c(21,38,36))
dplyr::arrange(dplyr::summarise(dplyr::group_by(filtered_violation_c
odes_2015, filtered_violation_codes_2015$Violation_time_bin),
Violation_count=sum(Count)), desc(Violation_count))  %>% head(n=3)
# Violation_time_bin Violation_count
# Bin 3                      999044
# Bin 4                      531823
# Bin 5                      119943
```

```
# Most commonly occurring violations 21, 38 and 36 are in the bins
3, 4 and 5 which means that
# these violations occur between times 8am and 7pm
# No parking where not allowed, Failing to show receipt and
Exceeding allowed time all occur during the day and evenings

# 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

# Let us divide the year into 4 quarters representing 4 seasons
# For simplicity we shall use convention 1 to 3 months for Spring, 4
to 6 as Summer, 7 to 9 as Autumn and
# 10 to 12 as Winter

season_bins_sql_2015 <- "select case when substring(Issue_Date,1,2)
in ('01','02','03') then 'Bin 1'
when substring(Issue_Date,1,2) in ('04','05','06') then 'Bin 2'
when substring(Issue_Date,1,2) in ('07','08','09') then 'Bin 3'
when substring(Issue_Date,1,2) in ('10','11','12') then 'Bin 4'
else null
end as Violation_season_bin, Violation_Code
from nyc_parking_tickets_2015_df_view where Issue_Date is not null
or Issue_Date not in ('NA', '')"

violations_season_bins_2015 <- SparkR::sql(season_bins_sql_2015)

createOrReplaceTempView(violations_season_bins_2015,
"violations_season_bins_2015_df_view")

violation_code_count_in_season_bins_2015 <- SparkR::sql("select
Violation_season_bin, Violation_Code, count(*) Count from
violations_season_bins_2015_df_view
                                                        group by
Violation_season_bin, Violation_Code")

# Use Collect action to get results in df to driver node for faster
aggregation
violation_code_count_season_coll_2015 <-
SparkR::collect(violation_code_count_in_season_bins_2015)

getTop3ViolationCodesInSeasonBins <- function(bin) {
  dplyr::filter(violation_code_count_season_coll_2015,
Violation_season_bin == bin) %>% dplyr::arrange(desc(Count)) %>%
head(n = 3)
}

# Get top 3 Violation Codes in Season Bin 1
getTop3ViolationCodesInSeasonBins('Bin 1')
# Violation_season_bin Violation_Code  Count
# Bin 1                38              336746
# Bin 1                21              281386
```

```
# Bin 1                14              219828
# Failing to show parking ticket and parking where not allowed are
the common reasons for receiving parking tickets

# Get top 3 Violation Codes in Season Bin 2
getTop3ViolationCodesInSeasonBins('Bin 2')
# Violation_season_bin Violation_Code  Count
# Bin 2                21             439516
# Bin 2                38             327158
# Bin 2                36             246660
# Failing to show parking ticket, exceeding time limit and parking
where not allowed are the common reasons for receiving parking
tickets

# Get top 3 Violation Codes in Season Bin 3
getTop3ViolationCodesInSeasonBins('Bin 3')
# There were no records in Bin 3 i.e no records for Season 3

# Get top 3 Violation Codes in Season Bin 4
getTop3ViolationCodesInSeasonBins('Bin 4')
# There were no records in Bin 4 i.e no records for Season 4

# 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.
#    Find total occurrences of the 3 most common violation codes
#    Then, search the internet for NYC parking violation code fines.
You will find a website (on the nyc.gov URL) that lists these fines.
They're divided into two categories, one for the highest-density
locations of the city, the other for the rest of the city. For
simplicity, take an average of the two.
#    Using this information, find the total amount collected for all
of the fines. State the code which has the highest total collection.
#    What can you intuitively infer from these findings?

# Three most commonly occurring Violation codes
most_common_violation_codes_2015 <-
summarize(groupBy(nyc_parking_tickets_2015,
nyc_parking_tickets_2015$Violation_Code),
                                          Count =
n(nyc_parking_tickets_2015$Violation_Code))
head(arrange(most_common_violation_codes_2015,
desc(most_common_violation_codes_2015$Count)), n = 5)
# Violation_Code  Count
# 21              720902
# 38              663904
# 14              466488
# 36              406249
# 37              373229
# Violation codes 21, 38 and 14 were the top 3 commonly occurring
violations

# Define a dataframe that has a specific fine for each Violation
```

```
Code from 0 to 100
# Source for Violation Code and Fines is https://www1.nyc.gov/site/
finance/vehicles/services-violation-codes.page
# Average fine has been used from two columns "Manhattan 96th St. &
below" and "All Other Areas"
# Where there are no values "NA" is used
all_violation_codes_2015 <- c(0:100)
all_avg_fines_2015 <-
c("NA","515",    "515",   "515",   "115",   "115",   "390",   "50",
"115",   "115",   "115",   "115",   "95",    "115",   "115",   "NA",
"95",    "95",    "115",   "115",   "62.5",  "55",    "60",    "62.5",
"62.5",  "115",   "115",   "180",   "95",    "515",   "515",   "115",
"50",    "50",    "50",    "50",    "50",    "50",    "50",    "62.5",
"115",   "NA",    "50",    "50",    "50",    "115",   "115",   "115",
"115",   "95",    "115",   "115",   "115",   "115",   "NA",    "115",
"115",   "65",    "55",    "115",   "55",    "55",    "55",    "95",
"95",    "95",    "55",    "165",   "65",    "65",    "65",    "65",
"65",    "65",    "65",    "65",    "NA",    "55",    "65",    "115",
"55",    "95",    "115",   "65",    "55",    "65",    "115",   "NA",
"NA",    "115",   "NA",    "55",    "55",    "65",    "100",   "NA",
"95",    "55",    "95",    "NA",    "NA")
fines_for_violation_codes_df_2015 <-
data.frame(all_violation_codes_2015, all_avg_fines_2015)
names(fines_for_violation_codes_df_2015) <- c("Violation_Code",
"Average_Fine")

# Merge Fine with Common Violation Codes dataframe
fines_for_violation_codes_spark_df_2015 <-
as.DataFrame(fines_for_violation_codes_df_2015)
total_collection_2015 <- drop(join(most_common_violation_codes_2015,
fines_for_violation_codes_spark_df_2015,
most_common_violation_codes_2015$Violation_Code ==
fines_for_violation_codes_spark_df_2015$Violation_Code),
fines_for_violation_codes_spark_df_2015$Violation_Code)
head(total_collection_2015)
# Violation_Code Count          Average_Fine
#  31            80223          115
#  85            16632           65
#  65            40              95
#  53            16830          115
#  78            32350           65
#  34            13               50

# Total fine for each Violation Code
total_collection_2015$TotalFine <- total_collection_2015$Count *
total_collection_2015$Average_Fine
head(total_collection_2015)
# Violation_Code Count  Average_Fine TotalFine
# 31             80223  115          9225645
# 85             16632  65           1081080
# 65             40     95           3800
# 53             16830  115          1935450
# 78             32350  65           2102750
# 34             13     50           650
```

```
createOrReplaceTempView(total_collection_2015,
"total_collection_2015_df_view")

# Total amount collected from fines for all Violation Codes
head(SparkR::sql("select sum(TotalFine) as Total_Amount from
total_collection_2015_df_view"))
# Total_Amount
# 405129342
# Total amount collected from all violations is $405129342

# Violation code that has the highest collection
head(arrange(total_collection_2015,
desc(total_collection_2015$TotalFine)), n = 3)
# Violation_Code  Count     Average_Fine TotalFine
# 14              466488    115          53646120
# 21              720902    55           39649610
# 38              663904    50           33195200
# Violation code 14 has the highest total collection of $53646120

#
---------------------------------------------------------------------
-----------------------------------
#                       Inferences for Parking Violations in New
York City for 2015
#
---------------------------------------------------------------------
-----------------------------------
#
# 1. Top 3 most commonly occurring violation codes were 21, 38 and
14
# 2. Top 3 reasons for parking violations are
#      a. No parking where parking is not allowed by sign, Parking
in excess of the allowed time or
#      b. Exceeding the posted speed limit in or near a designated
school zone
#      c  Standing or parking where standing is not allowed by sign,
street marking or; traffic control device.
# 3. Suburban, 4 Door Sedan and Vans were the vehicle types that
received maximum parking tickets
# 4. FORD, TOYOTA and HONDA vehicles received the most number of
parking tickets
# 5. Zones in Manhattan (Upper East, Midtown North and South) and
Northwest Queens have had the maximum number of parking tickets
issued
# 6. Police Stations of Manhattan (Upper East, Midtown North and
South) and Northwest Queens have issued the most number of parking
tickets
# 7. Violations were the highest between 4 and 7 PM. Parking in
excess of the allowed time or failing to show a receipt and parking
where it is not allowed
#    are the most common reasons for receiving parking ticket during
these hours
# 8. Violations were high between 8 and 11 AM. No parking where not
```

```
allowed, Failing to show a receipt and exceeding allowed time
#    are the most common reasons
# 9. Going through red light at an intersection was the most common
violation after 10 PM.
# 10. Stopping closer to 15 feet of fire hydrant was also common
during late nights and early mornings.
# 11. Most common violations all round the year were,
#      a. Failing to show parking ticket,
#      b. Exceeding time limit and
#      c. Parking where not allowed
# 12. Total fine of $405129342 was collected from all violation
codes
# 13. Violation code 14 (Standing or parking where standing is not
allowed by sign, street marking or; traffic control device)
#     collected the most fine
# 14. Even though the total count of violation code 14 was lesser
than codes 21 and 38 the total
#     revenue collected was more because the fine levied for code 14
was higher than the other two codes.


# Start Spark and Initialize Spark session

Sys.setenv(SPARK_HOME = "/usr/local/spark")
library(SparkR,lib.loc =
c(file.path(Sys.getenv("SPARK_HOME"),"R","lib")))
sparkR.session(master = "yarn")

# Load Required R libraries to help ease the analysis

library(dplyr)

# Read dataset (2016 New York parking tickets data) to Spark
Dataframe.

nyc_parking_tickets_raw_data_2016 <- SparkR::read.df("/
common_folder/nyc_parking/Parking_Violations_Issued_-
_Fiscal_Year_2016.csv",header = T, inferSchema = T,"CSV")

# Get the structure of meta data

printSchema((nyc_parking_tickets_raw_data_2016))

nrow(nyc_parking_tickets_raw_data_2016)
# Total parking records are 10626899
ncol(nyc_parking_tickets_raw_data_2016)
# Total number of variables - 51

#-----------------------------------------------------------------
--------------#
#                           Analysis for 2016
#
#-----------------------------------------------------------------
--------------#
```

```
# Let's do basic sanity checking to check if data has quality
issues.
# and essential variable has any missing values or contain duplicate
records

# Check if we have only 2016 data in the 2016 parking tickets
dataset.
# Create SQL view on SparkDataframe

createOrReplaceTempView(nyc_parking_tickets_raw_data_2016,
"nyc_parking_tickets_2016_df_view")

recs_by_issue_date <- SparkR::sql("select distinct(substr(`Issue
Date`,-4)) as Year_of_Issue
                                  from
nyc_parking_tickets_2016_df_view
                                  order by Year_of_Issue")

head(recs_by_issue_date,100)

# It is evident that 2016 dataset contains parking tickets records
that do not belong
# to year 2016. Therefore, we will remove them and retain only 2016
records for
# this analysis

nyc_parking_tickets_only_2016 <-
SparkR::filter(nyc_parking_tickets_raw_data_2016,substr(nyc_parking_
tickets_raw_data_2016$`Issue date`,-4,4) == '2016')

nrow(nyc_parking_tickets_only_2016)
# Total records with year of parking tickets being only 2016 =>
4872621

# Let's verify if we have correctly extracted 2016 data
# Total Records with year other than 2016
nrow(SparkR::filter(nyc_parking_tickets_raw_data_2016,substr(nyc_par
king_tickets_raw_data_2016$`Issue date`,-4,4) != '2016'))
# 5754278

# If we add 5754278 with 4872621 the total becomes 10626899 which is
exactly matching
# with raw data total i.e. 10626899

createOrReplaceTempView(nyc_parking_tickets_only_2016,
"nyc_parking_tickets_only_2016_df_view")

# Check if there are any records with missing summons number.

head(SparkR::sql("select count(*)
                 from nyc_parking_tickets_only_2016_df_view
                 where `Summons Number` is null
                 or `Summons Number` in ('NA', ' ') "))
```

```
# Count 0 which means all records have summons number associated
with parking ticket

# Check if there are any records with missing Issue Date.

head(SparkR::sql("select count(*)
                 from nyc_parking_tickets_only_2016_df_view
                 where `Issue Date` is null
                 or `Issue Date` in ('NA', ' ') "))

# No records without Issue date

# Check if there are any duplicate parking tickets with same summons
number

head(SparkR::sql("select `Summons Number`
                 , `Issue Date`
                 , count (1)
                 from nyc_parking_tickets_only_2016_df_view
                 group by `Summons Number`
                 , `Issue Date`
                 having count(*) > 1"),1)

# There are no duplicate records with same summons number on the
same date

# Check if there are any records with Registration State.

head(SparkR::sql("select count(*)
                 from nyc_parking_tickets_only_2016_df_view
                 where `Registration State` is null
                 or `Registration State` in ('NA', ' ') "))

# No records without registration state

# Check records without Plate ID

head(SparkR::sql("select count(*)
                 from nyc_parking_tickets_only_2016_df_view
                 where `Plate ID` is null
                 or `Plate ID` in ('NA',' ')"))

# There are 13 records without Plate ID. Since 13 records are
relatively low
# compared to total number of parking tickets, lets ignore these 13
records
# for our analysis

nyc_parking_tickets_only_2016 <-
SparkR::filter(nyc_parking_tickets_only_2016,!
nyc_parking_tickets_only_2016$`Plate ID` %in% c(' ', 'NA'))
nrow(nyc_parking_tickets_only_2016)
#4872608
```

```
# Check records without Violation Code

nrow(SparkR::filter(nyc_parking_tickets_only_2016,nyc_parking_ticket
s_only_2016$`Violation Code` %in% c(' ','NA','')))

# No records without violation code

# Recreate SQL view

createOrReplaceTempView(nyc_parking_tickets_only_2016,
"nyc_parking_tickets_2016_df_view")

#-----------------------------------------------------------------
--------------#
#                            Questions to be answered from the
analysis of 2016     #
#-----------------------------------------------------------------
--------------#

# 1. Find total number of tickets for the year

nrow(nyc_parking_tickets_only_2016)
# 4872608

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

head(SparkR::sql("select count(distinct `Registration State`)
                 from nyc_parking_tickets_2016_df_view"))
# Cars from 67 states received parking tickets in the year 2016

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

#
-----------------------------------------------------------------
----------------------------------#
# Assumption:
# Address can be of two types,
# 1. Address where the violation occurred and
# 2. Address where the vehicle is registered
#
-----------------------------------------------------------------
----------------------------------#

# Parking tickets with missing Address where the violation occurred

head(SparkR::sql("select count(*) as Count
                 from nyc_parking_tickets_2016_df_view
                 where `Street Code1`            is null
                 or `Street Code2`            is null
                 or `Street Code3`            is null
                 or `Violation Location`      is null
```

```
                or `Intersecting Street`     is null
                or `Violation Post Code`     is null"))

# There were 4314557 parking tickets with missing address of the
place where violation occurred

head(SparkR::sql("select count(*) as Count
                from nyc_parking_tickets_2016_df_view
                where `House Number`       is null
                or `House Number`       in ('', 'NA')
                or `Street Name`        is null
                or `Street Name`        in ('', 'NA')"))

# There were 895752 parking tickets that do not have either house
number or street name
# In total 4314557 + 895752 = 5210309 parking tickets had missing
address details

# ------------------------------#
#      Aggregation tasks       #
# ------------------------------#

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

Violation_code_counts_2016 <-
SparkR::summarize(SparkR::groupBy(nyc_parking_tickets_only_2016,nyc_
parking_tickets_only_2016$`Violation Code`),Count =
SparkR::n(nyc_parking_tickets_only_2016$`Violation Code`))
head(SparkR::arrange(Violation_code_counts_2016,SparkR::desc(Violati
on_code_counts_2016$Count)), n = 5)

#Violation Code   Count
#  21             664946
#  36             615242
#  38             547080
#  14             405883
#  37             330489
# Top 5 frequently occurring violation codes are 21, 36, 38, 14 and
37

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

vehicle_body_type_counts_2016 <-
SparkR::summarize(SparkR::groupBy( nyc_parking_tickets_only_2016

, nyc_parking_tickets_only_2016$`Vehicle Body Type`)
                                                , Count =
SparkR::n(nyc_parking_tickets_only_2016$`Vehicle Body Type`))
head(SparkR::arrange(vehicle_body_type_counts_2016,SparkR::desc(vehi
cle_body_type_counts_2016$Count)), n = 5)

#Vehicle Body Type    Count
```

```
#      SUBN           1596325
#      4DSD           133994
#      VAN            722234
#      DELV           354388
#      SDN            178954

vehicle_make_counts_2016 <-
SparkR::summarize(SparkR::groupBy( nyc_parking_tickets_only_2016
                                                                ,
nyc_parking_tickets_only_2016$`Vehicle Make`)
                                                , Count =
SparkR::n(nyc_parking_tickets_only_2016$`Vehicle Make`))
head(SparkR::arrange(vehicle_make_counts_2016,SparkR::desc(vehicle_m
ake_counts_2016$Count)), n = 5)

#Vehicle Make   Count
# FORD          612276
# TOYOT         529115
# HONDA         459465
# NISSA         382080
# CHEVR         339466

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

violation_precinct_counts_2016 <-
SparkR::summarize(SparkR::groupBy(nyc_parking_tickets_only_2016

, nyc_parking_tickets_only_2016$`Violation Precinct`)
                                                      , Count =
SparkR::n(nyc_parking_tickets_only_2016$`Violation Precinct`))

head(SparkR::arrange(violation_precinct_counts_2016,
SparkR::desc(violation_precinct_counts_2016$count)), n = 5)

#Violation Precinct   Count
#    0               828348
#   19               264299
#   13               156143
#    1               152231
#   14               150637

#
-----------------------------------------------------------------------
-----------------------------------
# Assumption:
# Precinct 0 is not a valid zone and does not appear in the NYPD
precincts list available on
# https://www1.nyc.gov/site/nypd/bureaus/patrol/precincts-
```

```
landing.page
# It could be that Precinct 0 refers to an incorrect value. So,
ignoring Precinct 0 although it has the
# highest count
#
-----------------------------------------------------------------
---------------------------------
# Zone 19 has the next maximum number of parking tickets. The 19th
Precinct command serves the Upper East Side of Manhattan.
# Zone 13 is New York East 21st Street
# Zone 1 is New York Ericcson place
# Zone 14 is Manhattan Midtown South
# Zones in Manhattan (Upper East,South) and Newyork East 21st street
and Newyork Ericcson Place have had the maximum number of parking
tickets issued in 2016

issuer_precinct_counts_2016 <-
SparkR::summarize(SparkR::groupBy(nyc_parking_tickets_only_2016
                                                                  ,ny
c_parking_tickets_only_2016$`Issuer Precinct`)
                                                 ,Count =
SparkR::n(nyc_parking_tickets_only_2016$`Issuer Precinct`))

head(SparkR::arrange(issuer_precinct_counts_2016,
SparkR::desc(issuer_precinct_counts_2016$count)), n = 5)

#   Issuer Precinct     Count
#               0       948438
#              19       258049
#              13       153477
#               1       146987
#              14       146165
# Ignoring Issuer Precinct 0 as it appears to be an invalid Precinct
# Police stations of Manhattan (Upper East,South) and Newyork East
21st street and Newyork Ericcson Place have had the maximum number
of parking tickets issued in 2016

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

# Violcation codes for precincts (19,13,1) that have issued the most
number of tickets

violation_codes_for_issuer_precincts <- SparkR::sql ("select `Issuer
Precinct`
                                                     , `Violation
Code`
                                                     , count(*) as
Count
                                                     from
nyc_parking_tickets_only_2016_df_view
```

```
                                                    where `Issuer
Precinct` in (19,13,1)
                                                    group by
`Issuer Precinct`
                                                    , `Violation
Code`")

head(SparkR::arrange(violation_codes_for_issuer_precincts,SparkR::de
sc(violation_codes_for_issuer_precincts$count)), n = 10)

#         Issuer Precinct     Violation Code  Count
#               19                    37      38052
#               19                    38      37855
#               19                    46      36442
#                1                    14      34101
#               19                    14      28772
#               19                    21      25588
#               19                    16      24647
#               13                    69      23356
#                1                    16      19782
#               13                    47      17532


# Precinct Zone 19 had the highest frequency of violation code 37
# Precinct Zone 1 had violation code 14 as the most occurring
# Yes, there is a violation code 14 that occur commonly across
Precincts 1 and 19.

common_violation_codes <- SparkR::sql("select `Violation Code`
                                      , count(*) as Count
                                      from
nyc_parking_tickets_only_2016_df_view
                                      where `Issuer Precinct` in
(19,13,1)
                                      group by `Violation Code`")
head(SparkR::arrange(common_violation_codes,SparkR::desc(common_viol
ation_codes$count)), n = 5)

#     Violation Code  Count
#             14      78685
#             38      62192
#             37      57678
#             46      49863
#             16      45358

# Violation code 14 has a very high frequency
# Violation codes 14, 38 and 37 are the top 3 most commonly occuring
violation codes in Zones 19, 13 and 1

# 5. You'd want to find out the properties of parking violations
across different times of the day:
#    The Violation Time field is specified in a strange format. Find
a way to make this into a time attribute that you can use to divide
into groups.
```

```
#    Find a way to deal with missing values, if any.
#    Divide 24 hours into 6 equal discrete bins of time. The
intervals you choose are at your discretion. For each of these
groups, find the 3 most commonly occurring violations
#    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)

# Determine if there are any missing values for Violation_Time
head(SparkR::sql("select count(*)
                 from nyc_parking_tickets_only_2016_df_view
                 where `Violation Time` is null
                 or `Violation Time` in ('na', '')"))
# 74 parking tickets issued without violation time.
# Since the number is relatively insignificant with total number of
records
# ignoring these records from analysis will certainly be no harm.

# Check for time consistency

head(SparkR::sql("select `Violation Time`
                 from nyc_parking_tickets_only_2016_df_view
                 where substring(`Violation Time`, 1, 2) = '12'
                 and substring(`Violation Time`, -1) = 'A'"))
#     Violation Time
#          1230A
#          1234A
#          1230A
#          1229A
#          1225A
#          1250A

# There are many parking tickets that have time recorded with 12:nn
AM hours. These records will be binned
# along with 00 AM hours.

head(SparkR::sql("select `Violation Time`
                 from nyc_parking_tickets_only_2016_df_view
                 where substring(`Violation Time`, 1, 2) = '03'
                 and substring(`Violation Time`, -1) = 'P'"))
#     Violation Time
#          0309P
#          0320P
#          0309P
#          0347P
#          0309P
#          0308P

# It can be seen that a proper 24 Hour Time convention was not been
followed. So, care must be taken whilst binning.

# Are there any records with Violation Time length greater than or
lesser than 5
```

```
head(SparkR::sql("select count(*) as Count
                 from nyc_parking_tickets_only_2016_df_view
                 where length(`Violation Time`) > '5'
                 or length(`Violation Time`) < '5'"))
# 7 parking tickets with violation time having neither A or P.
# Let's remove them for further analysis

nyc_parking_tickets_only_2016 <-
SparkR::filter(nyc_parking_tickets_only_2016,
length(nyc_parking_tickets_only_2016$`Violation Time`) == '5')

#Recreate SQL view

createOrReplaceTempView(nyc_parking_tickets_only_2016,
"nyc_parking_tickets_only_2016_df_view")

# Are there any records with Violation Time not in A or P
head(SparkR::sql("select count(*) as Count
                 from nyc_parking_tickets_only_2016_df_view
                 where upper(substring(`Violation Time`, -1)) not in
('A', 'P')"))
# Count
# 0
# There were no parking tickets that are neither A or P

# Are there any records with Violation Time not in the 24 hour time
window

head(SparkR::sql("select count(*) as Count
                 from nyc_parking_tickets_only_2016_df_view
                 where substring(`Violation Time`, 1, 2) not in
('00','01','02','03','04','05','06','07','08','09','10','11','12','1
3','14','15','16','17','18','19','20','21','22','23')"))
# Count
#  108
# There were 108 parking tickets that had invalid time and these
records will be excluded

# Create 6 bins of 24 hour time period

time_bins_sql_2016 <- "select case when substring(`Violation Time`,
1,2) in ('00','01','02','03','12') and upper(substring(`Violation
Time`,-1)) = 'A' then 'Bin 1'
when substring(`Violation Time`,1,2) in ('04','05','06','07') and
upper(substring(`Violation Time`,-1)) = 'A' then 'Bin 2'
when substring(`Violation Time`,1,2) in ('08','09','10','11') and
upper(substring(`Violation Time`,-1)) = 'A' then 'Bin 3'
when substring(`Violation Time`,1,2) in
('12','13','14','15','00','01','02','03') and
upper(substring(`Violation Time`,-1)) = 'P' then 'Bin 4'
when substring(`Violation Time`,1,2) in
('16','17','18','19','04','05','06','07') and
upper(substring(`Violation Time`,-1)) = 'P' then 'Bin 5'
when substring(`Violation Time`,1,2) in
```

```
('20','21','22','23','08','09','10','11') and
upper(substring(`Violation Time`,-1)) = 'P' then 'Bin 6'
else null
end as Violation_time_bin, `Violation Code`as Violation_Code,
`Violation Time` as Violation_Time
from nyc_parking_tickets_only_2016_df_view where `Violation Time` is
not null and
substring(`Violation Time`, 1, 2) in
('00','01','02','03','04','05','06','07','08','09','10','11','12','1
3','14','15','16','17','18','19','20','21','22','23')"

violations_time_bins_2016 <- SparkR::sql(time_bins_sql_2016)

head(violations_time_bins_2016)

#       Violation_time_bin  Violation_Code  Violation_Time
#              Bin 3             24          0924A
#              Bin 5             40          0530P
#              Bin 6             67          1020P
#              Bin 4             20          0148P
#              Bin 3             21          1135A
#              Bin 5             98          0448P

createOrReplaceTempView(violations_time_bins_2016,
"violations_time_bins_2016_df_view")

violation_code_count_in_time_bins_2016 <- SparkR::sql("select
Violation_time_bin
                                                      ,
Violation_Code
                                                      , count(*)
Count
                                                      from
violations_time_bins_2016_df_view
                                                      group by
Violation_time_bin
                                                      ,
Violation_Code")

# Use Collect action to get results in df to driver node for faster
aggregation

violation_code_count_coll_2016 <-
SparkR::collect(violation_code_count_in_time_bins_2016)

getTop3ViolationCodesInTimeBins <- function(bin) {
  dplyr::filter(violation_code_count_coll_2016, Violation_time_bin
== bin) %>% dplyr::arrange(desc(Count)) %>% head(n = 3)
}

# Get top 3 Violation Codes in Bin 1 ('00','01','02','03','12') AM
getTop3ViolationCodesInTimeBins('Bin 1')

#       Violation_time_bin  Violation_Code  Count
```

```
#              Bin 1             21         31956
#              Bin 1             40         19078
#              Bin 1             78         14706

# Get top 3 Violation Codes in Bin 2 ('04','05','06','07') AM

getTop3ViolationCodesInTimeBins('Bin 2')

#       Violation_time_bin Violation_Code Count
#              Bin 2             14       65347
#              Bin 2             21       48239
#              Bin 2             40       42306

# Get top 3 Violation Codes in Bin 3 ('08','09','10','11') AM
getTop3ViolationCodesInTimeBins('Bin 3')

#       Violation_time_bin  Violation_Code  Count
#              Bin 3             21         525280
#              Bin 3             36         284279
#              Bin 3             38         185395
# Violations were high between 8 and 11 AM. No parking where not
allowed, Failing to show a receipt and exceeding allowed time
# were the most common reasons

# Get top 3 Violation Codes in Bin 4
('12','13','14','15','00','01','02','03') PM
getTop3ViolationCodesInTimeBins('Bin 4')

#       Violation_time_bin  Violation_Code  Count
#              Bin 4             36         273581
#              Bin 4             38         234221
#              Bin 4             37         183854

# Get top 3 Violation Codes in Bin 5
('16','17','18','19','04','05','06','07') PM
getTop3ViolationCodesInTimeBins('Bin 5')

#       Violation_time_bin  Violation_Code  Count
#              Bin 5             38         105657
#              Bin 5             37         79991
#              Bin 5             14         63778

# Violations were the highest between 4 and 7 PM
# Parking in excess of the allowed time or failing to show a receipt
and parking where it is not allowed
# are the most common reasons for receiving parking ticket during
these hours

# Get top 3 Violation Codes in Bin 6
('20','21','22','23','08','09','10','11') PM
getTop3ViolationCodesInTimeBins('Bin 6')

#     Violation_time_bin  Violation_Code  Count
#              Bin 6             38       20851
```

```
#              Bin 6              7       20246
#              Bin 6             40       20030

# Three most commonly occurring Violation codes
createOrReplaceTempView(violations_time_bins_2016,"violations_time_b
ins_2016_df_view")
most_popular_violation_codes_2016 <- SparkR::sql("select
Violation_Code
                                                 , count(*) as Count
                                                 from
violations_time_bins_2016_df_view
                                                 group by
Violation_Code
                                                 order by Count desc
                                                 limit 3")

head(most_popular_violation_codes_2016)

#     Violation_Code  Count
#              21      664914
#              36      615242
#              38      547080

# Violation codes 21, 38 and 36 are the top 3 commonly occurring
violations

# Get time bins of most commonly occuring violations

filtered_violation_codes_2016 <-
dplyr::filter(violation_code_count_coll_2016
                                                 ,
violation_code_count_coll_2016$Violation_Code %in% c(21,38,36))

dplyr::arrange(dplyr::summarise(dplyr::group_by(filtered_violation_c
odes_2016, filtered_violation_codes_2016$Violation_time_bin),
Violation_count=sum(Count)), desc(Violation_count))  %>% head(n=3)

# Violation_time_bin    Violation_count
#       Bin 3           994954
#       Bin 4           566791
#       Bin 5           124172

# Most commonly occurring violations 21, 38 and 36 are in the bins
3, 4 and 5 which means that
# these violations occur between times 8am and 7pm

# 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

# Let us divide the year into 4 quarters representing 4 seasons
# For simplicity we shall use convention 1 to 3 months for Spring, 4
```

```
to 6 as Summer, 7 to 9 as Autumn and
# 10 to 12 as Winter

season_bins_sql_2016 <- "select case when substring(`Issue Date`,
1,2) in ('01','02','03') then 'Bin 1'
when substring(`Issue Date`,1,2) in ('04','05','06') then 'Bin 2'
when substring(`Issue Date`,1,2) in ('07','08','09') then 'Bin 3'
when substring(`Issue Date`,1,2) in ('10','11','12') then 'Bin 4'
else null
end as Violation_season_bin, `Violation Code` as Violation_Code
from nyc_parking_tickets_only_2016_df_view
where `Issue Date` is not null or `Issue Date` not in ('NA', '')"

violations_season_bins_2016 <- SparkR::sql(season_bins_sql_2016)

createOrReplaceTempView(violations_season_bins_2016,
"violations_season_bins_2016_df_view")

violation_code_count_in_season_bins_2016 <- SparkR::sql("select
Violation_season_bin
                                                        ,
Violation_Code
                                                        , count(*)
Count
                                                        from
violations_season_bins_2016_df_view
                                                        group by
Violation_season_bin
                                                        ,
Violation_Code")

# Use Collect action to get results in df to driver node for faster
aggregation
violation_code_count_season_coll_2016 <-
SparkR::collect(violation_code_count_in_season_bins_2016)

getTop3ViolationCodesInSeasonBins <- function(bin) {
  dplyr::filter(violation_code_count_season_coll_2016,
Violation_season_bin == bin) %>% dplyr::arrange(desc(Count)) %>%
head(n = 3)
}

# Get top 3 Violation Codes in Season Bin 1
getTop3ViolationCodesInSeasonBins('Bin 1')

#       Violation_season_bin  Violation_Code  Count
#                Bin 1             21         349296
#                Bin 1             36         341787
#                Bin 1             38         308987

# Failing to show parking ticket, exceeding time limit and parking
where not allowed are the common reasons for receiving parking
tickets
```

```
# Get top 3 Violation Codes in Season Bin 2
getTop3ViolationCodesInSeasonBins('Bin 2')

#     Violation_season_bin  Violation_Code  Count
#                Bin 2             21       315234
#                Bin 2             36       273455
#                Bin 2             38       238083

# Failing to show parking ticket, exceeding time limit and parking
where not allowed are the common reasons for receiving parking
tickets

# Get top 3 Violation Codes in Season Bin 3
getTop3ViolationCodesInSeasonBins('Bin 3')

#     Violation_season_bin  Violation_Code Count
#                Bin 3             21       248
#                Bin 3             46       214
#                Bin 3             40       89

# Get top 3 Violation Codes in Season Bin 4
getTop3ViolationCodesInSeasonBins('Bin 4')

#       Violation_season_bin  Violation_Code Count
#                Bin 4             21         167
#                Bin 4             46         164
#                Bin 4             40         80

# 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.
#    Find total occurrences of the 3 most common violation codes
#    Then, search the internet for NYC parking violation code fines.
You will find a website (on the nyc.gov URL) that lists these fines.
They’re divided into two categories, one for the highest-density
locations of the city, the other for the rest of the city. For
simplicity, take an average of the two.
#    Using this information, find the total amount collected for all
of the fines. State the code which has the highest total collection.
#    What can you intuitively infer from these findings?

most_common_violation_codes_2016 <-  SparkR::sql("select `Violation
Code`
                                                 , count(*) as Count
                                                 from
nyc_parking_tickets_only_2016_df_view
                                                 group by `Violation
Code`
                                                 order by Count desc
                                                 limit 5")
head(most_common_violation_codes_2016)

#     Violation Code  Count
```

```
#             21      664945
#             36      615242
#             38      547080
#             14      405883
#             37      330489
# Violation codes 21, 36 and 38 are the top 3 commonly occurring
violations

# Define a dataframe that has a specific fine for each Violation
Code from 0 to 100
# Source for Violation Code and Fines is https://www1.nyc.gov/site/
finance/vehicles/services-violation-codes.page
# Average fine has been used from two columns "Manhattan 96th St. &
below" and "All Other Areas"
# Where there are no values "NA" is used

all_violation_codes_2016 <- c(0:100)
all_avg_fines_2016 <-
c("NA","515",    "515",   "515",   "115",   "115",   "390",   "50",
"115",   "115",   "115",   "115",   "95",    "115",   "115",   "NA",
"95",    "95",    "115",   "115",   "62.5",  "55",    "60",    "62.5",
"62.5",  "115",   "115",   "180",   "95",    "515",   "515",   "115",
"50",    "50",    "50",    "50",    "50",    "50",    "50",    "62.5",
"115",   "NA",    "50",    "50",    "50",    "115",   "115",   "115",
"115",   "95",    "115",   "115",   "115",   "115",   "NA",    "115",
"115",   "65",    "55",    "115",   "55",    "55",    "55",    "95",
"95",    "95",    "55",    "165",   "65",    "65",    "65",    "65",
"65",    "65",    "65",    "65",    "NA",    "55",    "65",    "115",
"55",    "95",    "115",   "65",    "55",    "65",    "115",   "NA",
"NA",    "115",   "NA",    "55",    "55",    "65",    "100",   "NA",
"95",    "55",    "95",    "NA",    "NA")
fines_for_violation_codes_df_2016 <-
data.frame(all_violation_codes_2016, all_avg_fines_2016)
names(fines_for_violation_codes_df_2016) <- c("Violation_Code",
"Average_Fine")

# Merge Fine with Common Violation Codes dataframe

fines_for_violation_codes_spark_df_2016 <-
as.DataFrame(fines_for_violation_codes_df_2016)
total_collection_2016 <- drop(join(most_common_violation_codes_2016,
fines_for_violation_codes_spark_df_2016,
most_common_violation_codes_2016$`Violation Code` ==
fines_for_violation_codes_spark_df_2016$Violation_Code),
fines_for_violation_codes_spark_df_2016$Violation_Code)
head(total_collection_2016)

#     Violation Code  Count       Average_Fine
#             14      405883          115
#             21      664945           55
#             36      615242           50
#             37      330489           50
#             38      547080           50
```

```
# Total fine for each Violation Code
total_collection_2016$TotalFine <- total_collection_2016$Count *
total_collection_2016$Average_Fine
head(total_collection_2016)

createOrReplaceTempView(total_collection_2016,"total_collection_2016
_df_view")

head(SparkR::sql("select sum(TotalFine) as Total_Amount
                 from total_collection_2016_df_view"))

# Total_Amount collected from all violations is $157889070

# Violation code that has the highest collection

head(SparkR::arrange(total_collection_2016,
SparkR::desc(total_collection_2016$TotalFine)), n = 3)

#     Violation Code  Count       Average_Fine TotalFine
#             14      405883          115       46676545
#             21      664945           55       36571975
#             36      615242           50       30762100

#
-----------------------------------------------------------------------
-----------------------------------
#                       Inferences for Parking Violations in New
York City for 2016
#
-----------------------------------------------------------------------
-----------------------------------
#
# 1. Top 3 most commonly occurring violation codes were 21, 36 and
38
# 2. Top 3 reasons for parking violations are
#      a. No parking where parking is not allowed by sign, street
marking or traffic control device
#      b. Exceeding the posted speed limit in or near a designated
school zone.
#      c  Failing to show a receipt or tag in the windshield.
#         Drivers get a 5-minute grace period past the expired time
on Muni-Meter receipts.
# 3. Suburban, 4 Door Sedan and Vans were the vehicle types that
received maximum parking tickets
# 4. FORD, TOYOTA and HONDA vehicles received the most number of
parking tickets
# 5. Zones in Manhattan (Upper East), Newyork East and Ericcson
areas have had the maximum number of parking tickets issued
# 6. Police Stations of Manhattan (Upper East), Newyork East and
Ericcson have issued the most number of parking tickets
# 7. Violations were the highest between 4 and 7 PM. Parking in
excess of the allowed time or failing to show a receipt and parking
where it is not allowed
#    are the most common reasons for receiving parking ticket during
```

```
these hours
# 8. Violations were high between 8 and 11 AM. No parking where not
allowed, Failing to show a receipt and exceeding allowed time
#    are the most common reasons
# 9. Going through red light at an intersection was the most common
violation after 10 PM.
# 10. Stopping closer to 15 feet of fire hydrant was also common
during late nights and early mornings.
# 11. Most common violations all round the year were,
#      a. Failing to show parking ticket,
#      b. Exceeding time limit and
#      c. Parking where not allowed
# 12. Total fine of $157889070 was collected from all violation
codes
# 13. Violation code 14 (Standing or parking where standing is not
allowed by sign, street marking or; traffic control device)
#     collected the most fine
# 14. Even though the total count of violation code 14 was lesser
than codes 21 and 36 the total
#     revenue collected was more because the fine levied for code 14
was higher than the other two codes.
#-----------------------------------------------------------------
-----------------------------------------------------------------
--#


#-----------------------------------------------------------------
--------------#
#                            Analysis for 2017
#
#-----------------------------------------------------------------
--------------#

# load SparkR
Sys.setenv(SPARK_HOME = "/usr/local/spark")
library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R",
"lib")))
sparkR.session(master = "yarn")

# Read the data file
path <- "/common_folder/nyc_parking/Parking_Violations_Issued_-
_Fiscal_Year_2017.csv"
nyc_parking_tickets_raw_data_2017 <- read.df(path, source = "CSV",
header = "true", inferSchema = "true")

# Examine data
nrow(nyc_parking_tickets_raw_data_2017)
# Total parking records were 10803028

ncol(nyc_parking_tickets_raw_data_2017)
# Dataset contained 43 variables

str(nyc_parking_tickets_raw_data_2017)
# 'SparkDataFrame': 43 variables:
```

```
#$ 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"

head(nyc_parking_tickets_raw_data_2017)
#Summons Number Plate ID Registration State Plate Type Issue Date
Violation Code Vehicle Body Type Vehicle Make
#1     5092469481  GZH7067                 NY        PAS 07/10/2016
7              SUBN        TOYOT
#2     5092451658  GZH7067                 NY        PAS 07/08/2016
7              SUBN        TOYOT
#3     4006265037  FZX9232                 NY        PAS 08/23/2016
5              SUBN         FORD
#4     8478629828  66623ME                 NY        COM 06/14/2017
47              REFG        MITSU
#5     7868300310  37033JV                 NY        COM 11/21/2016
69              DELV        INTER
#6     5096917368  FZD8593                 NY        PAS 06/13/2017
7              SUBN        ME/BE
#Issuing Agency Street Code1 Street Code2 Street Code3 Vehicle
Expiration Date Violation Location
#1              V            0            0            0
0                NA
#2              V            0            0            0
0                NA
#3              V            0            0            0
0                NA
#4              T        10610        34330        34350
20180630                 14
#5              T        10510        34310        34330
20170228                 13
#6              V            0            0            0
0                NA
#Violation Precinct Issuer Precinct Issuer Code Issuer Command
Issuer Squad Violation Time Time First Observed
#1                  0               0           0           <NA>
<NA>          0143A               <NA>
#2                  0               0           0           <NA>
<NA>          0400P               <NA>
#3                  0               0           0           <NA>
```

```
<NA>          0233P                <NA>
#4                14              14      359594           T102
J          1120A                <NA>
#5                13              13      364832           T102
M          0555P                <NA>
#6                 0               0           0           <NA>
<NA>          0852P                <NA>
# Violation County Violation In Front Of Or Opposite House Number
Street Name Intersecting Street
#1               BX                                 <NA>         <NA>
ALLERTON AVE (W/B) @          BARNES AVE
#2               BX                                 <NA>         <NA>
ALLERTON AVE (W/B) @          BARNES AVE
#3               BX                                 <NA>         <NA>
SB WEBSTER AVE @ E 1             94TH ST
#4               NY                                    O          330
7th Ave                <NA>
#5               NY                                    F          799
6th Ave                <NA>
#6               QN                                 <NA>         <NA>
NORTHERN BLVD (E/B)     @ MARATHON PKWY
#Date First Observed Law Section Sub Division Violation Legal Code
Days Parking In Effect     From Hours In Effect
#1                   0        1111            D                    T
<NA>                 <NA>
#2                   0        1111            D                    T
<NA>                 <NA>
#3                   0        1111            C                    T
<NA>                 <NA>
#4                   0         408           l2                 <NA>
Y                0700A
#5                   0         408           h1                 <NA>
Y                0700A
#6                   0        1111            D                    T
<NA>                 <NA>
#  To Hours In Effect Vehicle Color Unregistered Vehicle? Vehicle
Year Meter Number Feet From Curb
#1               <NA>            GY                   NA
2001         <NA>              0
#2               <NA>            GY                   NA
2001         <NA>              0
#3               <NA>            BK                   NA
2004         <NA>              0
#4              0700P            WH                   NA
2007         <NA>              0
#5              0700P         WHITE                   NA
2007         <NA>              0
#6               <NA>            WH                   NA
2012         <NA>              0
#Violation Post Code         Violation Description No Standing or
Stopping Violation Hydrant Violation
#1                <NA>  FAILURE TO STOP AT RED LIGHT
<NA>              <NA>
#2                <NA>  FAILURE TO STOP AT RED LIGHT
```

```
<NA>              <NA>
#3                <NA>            BUS LANE VIOLATION
<NA>              <NA>
#4                  04         47-Double PKG-Midtown
<NA>              <NA>
#5                31 6 69-Failure to Disp Muni Recpt
<NA>              <NA>
#6                <NA>  FAILURE TO STOP AT RED LIGHT
<NA>              <NA>
#Double Parking Violation
#1                     <NA>
#2                     <NA>
#3                     <NA>
#4                     <NA>
#5                     <NA>
#6                     <NA>

printSchema(nyc_parking_tickets_raw_data_2017)
# root
#|-- 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?: 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)
#
-----------------------------------------------------------------------
----------------------------------
# --                                  Analysis for 2017
--
#
-----------------------------------------------------------------------
----------------------------------

# Rename Issue Date and Summons Number columns
nyc_parking_tickets_raw_data_2017 <-
withColumnRenamed(nyc_parking_tickets_raw_data_2017, "Issue Date",
"Issue_Date")
nyc_parking_tickets_raw_data_2017 <-
withColumnRenamed(nyc_parking_tickets_raw_data_2017, "Summons
Number", "Summons_Number")

# Create a Temp View of Parking Tickets 2017 Data Frame for
performing SQL operations
createOrReplaceTempView(nyc_parking_tickets_raw_data_2017,
"nyc_parking_tickets_2017_df_view")

# Check if there are any Data Quality Issues?

# 1, Does the file contain parking tickets only issued in 2017?
recs_by_issue_date <- SparkR::sql("select
distinct(substring(Issue_Date, -4)) as Year_Of_Issue, count(*) as
Total from nyc_parking_tickets_2017_df_view group by Year_Of_Issue
order by Year_Of_Issue")
showDF(recs_by_issue_date, 100, FALSE)
# Although the file says 2017 but it can be seen that there are
parking tickets issued from other years
# such as 1985, 1986. 1988, 1991, 2000 ... 2010, 2011, 2012, 2017
# There are 5368391 records of 2016 year and 5431918 of 2017 year.

#
-----------------------------------------------------------------------
----------------------------------
# Assumption:
# As this analysis is for the year 2017, parking tickets only
pertaining to 2017 are considered
#
-----------------------------------------------------------------------
----------------------------------

# Filter out parking tickets issued from other years and only retain
```

```
for year 2017
nyc_parking_tickets_only_2017 <- SparkR::sql("select * from
nyc_parking_tickets_2017_df_view where substring(Issue_Date, -4) =
2017")

head(nyc_parking_tickets_only_2017)
#Summons_Number Plate ID Registration State Plate Type Issue_Date
Violation Code Vehicle Body Type Vehicle Make
#1     8478629828  66623ME                NY        COM 06/14/2017
47              REFG        MITSU
#2     5096917368  FZD8593                NY        PAS 06/13/2017
7               SUBN        ME/BE
#3     1407740258  2513JMG                NY        COM 01/11/2017
78              DELV        FRUEH
#4     1413656420 T672371C                NY        PAS 02/04/2017
40              TAXI        TOYOT
#5     8480309064  51771JW                NY        COM 01/26/2017
64               VAN        INTER
#6     1416638830   GLP367                NY        PAS 04/30/2017
20              SUBN        DODGE
#Issuing Agency Street Code1 Street Code2 Street Code3 Vehicle
Expiration Date Violation Location
#1              T        10610        34330        34350
20180630                14
#2              V            0            0            0
0                 NA
#3              P            0        40404        40404
20161130               106
#4              P        59630        73470        82230
20170531                73
#5              T        17850        10210        10110
88888888                17
#6              P        17650        10110        10010
20180304                17
#Violation Precinct Issuer Precinct Issuer Code Issuer Command
Issuer Squad Violation Time Time First Observed
#1                 14              14      359594           T102
J          1120A               <NA>
#2                  0               0           0           <NA>
<NA>          0852P               <NA>
#3                106             106      960979           0106
0000          0015A               <NA>
#4                 73              73      960758           0073
0000          0525A               <NA>
#5                 17              17      363557           T102
L          0256P               <NA>
#6                 17              17      940179           0017
0000          1232A               <NA>
# Violation County Violation In Front Of Or Opposite House Number
Street Name Intersecting Street
#1               NY                                  O          330
7th Ave                <NA>
#2               QN                             <NA>         <NA>
NORTHERN BLVD (E/B)     @ MARATHON PKWY
```

```
#3                Q                                  <NA>          126
ST 115 AVE                 <NA>
#4                K                                     F          279
MCDOUGAL ST                <NA>
#5               NY                                     F          204
E 43rd St                <NA>
#6               NY                                     O          330
E 33 ST                <NA>
#  Date First Observed Law Section Sub Division Violation Legal Code
Days Parking In Effect     From Hours In Effect
#1                   0         408           l2                  <NA>
Y                0700A
#2                   0        1111            D                     T
<NA>                 <NA>
#3                   0         408           E2                  <NA>
BBBBBBB                  ALL
#4                   0         408           F1                  <NA>
BBBBBBB                  ALL
#5                   0         408           C8                  <NA>
YYYYYYY                 <NA>
#6                   0         408           E2                  <NA>
YYYYYYY                1200A
#To Hours In Effect Vehicle Color Unregistered Vehicle? Vehicle Year
Meter Number Feet From Curb
#1              0700P            WH                    NA
2007         <NA>              0
#2               <NA>            WH                    NA
2012         <NA>              0
#3                ALL         WHITE                    0
2015            -              0
#4                ALL           BLK                    0
2015            -              0
#5                <NA>         BROWN                   NA
2007         <NA>              0
#6              1159P            BLK                    0
2009            -              0
#Violation Post Code         Violation Description No Standing or
Stopping Violation Hydrant Violation
#1                  04         47-Double PKG-Midtown
<NA>              <NA>
#2                <NA>  FAILURE TO STOP AT RED LIGHT
<NA>              <NA>
#3                <NA>                          <NA>
<NA>              <NA>
#4                <NA>                          <NA>
<NA>              <NA>
#5                  06 64-No STD Ex Con/DPL, D/S Dec
<NA>              <NA>
#6                <NA>                          <NA>
<NA>              <NA>
#Double Parking Violation
#1                   <NA>
#2                   <NA>
#3                   <NA>
```

```
#4                    <NA>
#5                    <NA>
# 6                    <NA>

nrow(nyc_parking_tickets_only_2017)
# 5431918

# Create a Temp of Parking Tickets with Only 2017 records
createOrReplaceTempView(nyc_parking_tickets_only_2017,
"nyc_parking_tickets_only_2017_df_view")

# 2. Check if all tickets have a Summons Number
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_only_2017_df_view where Summons_Number is null
or Summons_Number in ('', 'NA')"))
#Count
# 0
# All records have a Summons Number for parking ticket

# 3. Are there any duplicate parking tickets i.e duplicate Summons
Number
head(SparkR::sql("select Summons_Number as Summons_Number,
Issue_date as Issue_Date, count(*) as Count from
nyc_parking_tickets_only_2017_df_view group by Summons_Number,
Issue_Date having count(*) > 1"))

# There were no duplicate parking tickets issued in the same year
2017

#
---------------------------------------------------------------------
-----------------------------------
# Assumption:
# Duplicate parking tickets are in the dataset by mistake. These
will be removed for further analysis.
#
---------------------------------------------------------------------
-----------------------------------

nyc_parking_tickets_2017 <- nyc_parking_tickets_only_2017

# Rename Registration State to Registration_State
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Registration State",
"Registration_State")
# Rename Plate ID to Plate_ID
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Plate ID", "Plate_ID")
# Rename Violation Code to Violation_Code
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Violation Code",
"Violation_Code")

# Create a Temp of Parking Tickets with duplicate records removed
```

```
createOrReplaceTempView(nyc_parking_tickets_2017,
"nyc_parking_tickets_2017_df_view")


# 4. Are there missing values for Issue Date?
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2017_df_view where Issue_Date is null or
Issue_Date in ('', 'NA')"))
# Count
# 0
# All parking tickets have an issue date and no rows have a missing
value

# 5. Is Registration State of vehicle missing in any parking
tickets?
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2017_df_view where Registration_State is null or
Registration_State in ('', 'NA')"))
# Count
# 0
# All parking tickets have a Registration State of car

# 6. Is Plate ID of vehicle missing in any parking tickets?
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2017_df_view where Plate_ID is null or Plate_ID
in ('', 'NA')"))
# Count
# 64
# There were 64 parking tickets that did not have Plate ID of a
vehicle
# As the number is insignificant these rows are retained

# 7. Is Violation Code missing in any parking tickets?
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2017_df_view where Violation_Code is null or
Violation_Code in ('', 'NA')"))
# Count
#  0
# All parking tickets have Violation Code

#
-----------------------------------------------------------------------
-----------------------------------
#                                Questions to be answered in the
analysis for 2017
#
-----------------------------------------------------------------------
-----------------------------------

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

# 1. Find total number of tickets for the year
```

```
head(SparkR::sql("select count(*) from
nyc_parking_tickets_2017_df_view"))
# 5373971

# 2. Find out how many unique states the cars which got parking
tickets came from
head(SparkR::sql("select count(distinct(Registration_State)) as
Count from nyc_parking_tickets_2017_df_view"))
# Count
# 65
# Cars from 65 states received parking tickets in 2017
states_df <- SparkR::sql("select distinct(Registration_State) from
nyc_parking_tickets_2017_df_view")
showDF(states_df, 100, FALSE)
#+------------------+
# |Registration_State|
# +------------------+
#|AZ                |
#|SC                |
#|NS                |
#|LA                |
#|MN                |
#|NJ                |
#|DC                |
#|OR                |
#|99                |
#|VA                |
#|RI                |
#|KY                |
#|WY                |
#|BC                |
#|NH                |
#|MI                |
#|GV                |
#|NV                |
#|QB                |
#|WI                |
#|ID                |
#|CA                |
#|CT                |
#|NE                |
#|MT                |
#|NC                |
#|VT                |
#|MD                |
#|DE                |
#|MO                |
#|IL                |
#|ME                |
#|MB                |
#|ND                |
#|WA                |
#|MS                |
#|AL                |
```

```
#|IN                |
#|OH                |
#|TN                |
#|IA                |
#|NM                |
#|PA                |
#|SD                |
#|FO                |
#|NY                |
#|ON                |
#|SK                |
#|AB                |
#|PE                |
#|TX                |
#|WV                |
#|GA                |
#|MA                |
#|KS                |
#|FL                |
#|CO                |
#|AK                |
#|AR                |
#|NB                |
#|OK                |
#|PR                |
#|UT                |
#|DP                |
#|HI                |
#  +------------------+
# Dataset shows that cars from 49 States of USA, 15 States of Canada
received parking tickets.
# There was 1 state with value 99

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

#
-----------------------------------------------------------------
----------------------------------
# Assumption:
# Address can be of two types,
# 1. Address where the violation occurred and
# 2. Address where the vehicle is registered
#
-----------------------------------------------------------------
----------------------------------

# 1. Address where violation occurred
# Rename Street Code1 to Street_Code1
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Street Code1",
"Street_Code1")
# Rename Street Code2 to Street_Code2
```

```
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Street Code2",
"Street_Code2")
# Rename Street Code3 to Street_Code3
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Street Code3",
"Street_Code3")
# Rename Violation Location to Violation_Location
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Violation Location",
"Violation_Location")
# Rename Intersecting Street to Intersecting_Street
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Intersecting Street",
"Intersecting_Street")
# Rename Violation Post Code to Violation_Post_Code
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Violation Post Code",
"Violation_Post_Code")

# 2. Address where the vehicle is registered
# Rename House Number to House_Number
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "House Number",
"House_Number")
# Rename Street Name to Street_Name
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Street Name",
"Street_Name")

createOrReplaceTempView(nyc_parking_tickets_2017,
"nyc_parking_tickets_clean_2017_df_view")

# Parking tickets with missing Address where the violation occurred
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_clean_2017_df_view where Street_Code1 is null or
Street_Code2 is null or Street_Code3 is null or
                 Violation_Location is null or Intersecting_Street
is null or Violation_Post_Code is null"))
# Count
# 4772670
# There were 4772670 parking tickets that were missing address where
violation occurred

# Parking tickets with missing Address of where the vehicle is
registered
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_clean_2017_df_view where House_Number is null or
House_Number in ('', 'NA') or Street_Name is null or Street_Name in
('', 'NA')"))
# Count
# 1029420
# There were 1029420 parking tickets that either does not have a
House Number or missing a Street Name
```

```
# So, a total of 5802090 parking tickets had incomplete address

# ----------------------------#
#      Aggregation tasks      #
# ----------------------------#

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

violation_code_counts_2017 <-
summarize(groupBy(nyc_parking_tickets_2017,
nyc_parking_tickets_2017$Violation_Code), Count =
n(nyc_parking_tickets_2017$Violation_Code))
head(arrange(violation_code_counts_2017,
desc(violation_code_counts_2017$count)), n = 5)
#   Violation_Code  Count
#             21 768087
#             36 662765
#             38 542079
#             14 476664
#             20 319646
# Top 5 commonly occurring violation codes were 21, 36, 38, 14 and
20

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

# Rename Vehicle Body Type to Vehicle_Body_Type
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Vehicle Body Type",
"Vehicle_Body_Type")
# Rename Vehicle Make to Vehicle_Make
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Vehicle Make",
"Vehicle_Make")

vehicle_body_type_counts_2017 <-
summarize(groupBy(nyc_parking_tickets_2017,
nyc_parking_tickets_2017$Vehicle_Body_Type),
                                          Count =
n(nyc_parking_tickets_2017$Vehicle_Body_Type))
head(arrange(vehicle_body_type_counts_2017,
desc(vehicle_body_type_counts_2017$count)), n = 5)
#Vehicle_Body_Type   Count
#              SUBN 1883954
#              4DSD 1547312
#               VAN  724029
#              DELV  358984
#               SDN  194197
# Suburban, 4 Door Sedan and Vans were the vehicle types that
received maximum parking tickets

vehicle_make_counts_2017 <-
```

```
summarize(groupBy(nyc_parking_tickets_2017,
nyc_parking_tickets_2017$Vehicle_Make),
                                        Count =
n(nyc_parking_tickets_2017$Vehicle_Make))
head(arrange(vehicle_make_counts_2017,
desc(vehicle_make_counts_2017$count)), n = 5)
#Vehicle_Make  Count
#         FORD 636844
#        TOYOT 605291
#        HONDA 538884
#        NISSA 462017
#        CHEVR 356032
# FORD, TOYOTA and HONDA vehicles received the most number of
parking tickets.

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

# Renaming Violation Precinct to Violation_Precinct
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Violation Precinct",
"Violation_Precinct")
# Renaming Issuer Precinct to Issuer_Precinct
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Issuer Precinct",
"Issuer_Precinct")

violation_precinct_counts_2017 <-
summarize(groupBy(nyc_parking_tickets_2017,
nyc_parking_tickets_2017$Violation_Precinct),
                                            Count =
n(nyc_parking_tickets_2017$Violation_Precinct))
head(arrange(violation_precinct_counts_2017,
desc(violation_precinct_counts_2017$count)), n = 5)

#Violation_Precinct  Count
#                  0 925596
#                 19 274445
#                 14 203553
#                  1 174702
#                 18 169131

#
-----------------------------------------------------------------
-----------------------------------
# Assumption:
# Precinct 0 is not a valid zone and does not appear in the NYPD
precincts list available on
# https://www1.nyc.gov/site/nypd/bureaus/patrol/precincts-
```

```
landing.page
# It could be that Precinct 0 refers to an incorrect value. So,
ignoring Precinct 0 although it has the
# highest count
#
-----------------------------------------------------------------
---------------------------------
# Zone 19 has the next maximum number of parking tickets. The 19th
Precinct command serves the Upper East Side of Manhattan.
# Zone 14 is Manhattan Midtown South
# Zone 1  is Manhattan Ericson Palace
# Zone 18 is Manhattan Midtown North
# Zones in Manhattan (Upper East, Midtown North, South and Ericson
Palace) have had the maximum number of parking tickets issued in
2015

issuer_precinct_counts_2017 <-
summarize(groupBy(nyc_parking_tickets_2017,
nyc_parking_tickets_2017$Issuer_Precinct),
                                          Count =
n(nyc_parking_tickets_2017$Issuer_Precinct))
head(arrange(issuer_precinct_counts_2017,
desc(issuer_precinct_counts_2017$count)), n = 5)
#Issuer_Precinct   Count
#               0 1078406
#              19  266961
#              14  200495
#               1  168740
#              18  162994
# Ignoring Issuer Precinct 0 as it appears to be an invalid valid
# Police Stations of Manhattan (Upper East, Midtown North, South and
Ericson Palace) have issued the most number of
# parking tickets in 2017

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

# Renaming Violation Code to Violation_Code
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Violation Code",
"Violation_Code")

createOrReplaceTempView(nyc_parking_tickets_2017,
"nyc_parking_tickets_2017_df_view")

# Violcation codes for precincts (19,14,1) that have issued the most
number of tickets
violation_codes_for_issuer_precincts <- SparkR::sql("select
Issuer_Precinct, Violation_Code, count(*) as Count from
nyc_parking_tickets_2017_df_view
                                                    where
```

```
Issuer_Precinct in (19,14,1) group by Issuer_Precinct,
Violation_Code")

head(arrange(violation_codes_for_issuer_precincts,desc(violation_cod
es_for_issuer_precincts$count)), n = 10)

#  Issuer_Precinct Violation_Code Count
#               19             46 48445
#               14             14 45036
#                1            14 38354
#               19             38 36386
#               19             37 36056
#               14             69 30464
#               19             14 29797
#               19             21 28415
#               14             31 22555
#                1             16 19081
# Precinct Zone 19 had the highest frequency of violation code 46
# Precinct Zone 14 had violation code 14 as the most occurring
# Precinct Zone 1 had violation code 14 as the most frequently
occurring
# Yes, the violation code 14 occurs commonly across Precincts

common_violation_codes <- SparkR::sql("select Violation_Code,
count(*) as Count from nyc_parking_tickets_2017_df_view
                                      where Issuer_Precinct in
(19,14,1) group by Violation_Code")

head(arrange(common_violation_codes,desc(common_violation_codes
$count)), n = 5)
#Violation_Code  Count
# 14             113187
# 46             68869
# 38             48190
# 37             43782
# 69             39046

# Violation code 14 has a very high frequency
# Violation codes 14, 46 and 38 are the top 3 most commonly occuring
violation codes in Zones 19, 14 and 1

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

```
# Renaming Violation Time to Violation_Time
nyc_parking_tickets_2017 <-
withColumnRenamed(nyc_parking_tickets_2017, "Violation Time",
"Violation_Time")

createOrReplaceTempView(nyc_parking_tickets_2017,
"nyc_parking_tickets_2017_df_view")

# Determine if there are any missing values for Violation_Time
head(SparkR::sql("select count(*) from
nyc_parking_tickets_2017_df_view where Violation_Time is null or
Violation_Time in ('na', '')"))
# 16 parking tickets issued have missing Violation Time
# There are very few records i.e 16 out of 5431918 with missing
Violation Time.
# So, ignoring these records from analysis as the number is
insignificant

# Check for Time consistency
head(SparkR::sql("select Violation_Time from
nyc_parking_tickets_2017_df_view where substring(Violation_Time, 1,
2) = '12' and substring(Violation_Time, -1) = 'A'"))
#Violation_Time
#1232A
#1255A
#1215A
#1221A
#1240A
#1245A
# There are many parking tickets that have time recorded with 12:nn
AM hours. These records will be binned
# along with 00 AM hours.

head(SparkR::sql("select Violation_Time from
nyc_parking_tickets_2017_df_view where substring(Violation_Time, 1,
2) = '03' and substring(Violation_Time, -1) = 'P'"))
#Violation_Time
#0334P
#0348P
#0338P
#0353P
#0300P
#0311P
# It can be seen that a proper 24 Hour Time convention was not been
followed. So, care must be taken whilst binning.

# Are there any records with Violation Time length greater than or
lesser than 5
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2017_df_view where length(Violation_Time) > '5'
or length(Violation_Time) < '5'"))
# Count
# 6
```

```
head(SparkR::sql("select Violation_Time from
nyc_parking_tickets_2017_df_view where length(Violation_Time) > '5'
or length(Violation_Time) < '5'"))
#Violation_Time
#0557
#0855
#0515
#0316
#0651
#1037
# There are 6 rows with invalid time length i.e <5 or >5 and these
records will be excluded

# Are there any records with Violation Time not in A or P
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2017_df_view where
upper(substring(Violation_Time, -1)) not in ('A', 'P')"))
# Count
# 8
# There are 8 rows with invalid time length i.e are neither A or P
and these records will be excluded


# Are there any records with Violation Time not in the 24 hour time
window
head(SparkR::sql("select count(*) as Count from
nyc_parking_tickets_2017_df_view where substring(Violation_Time, 1,
2) not in
('00','01','02','03','04','05','06','07','08','09','10','11','12','1
3','14','15','16','17','18','19','20','21','22','23')"))
# Count
#  44
# There were 44 parking tickets that had invalid time and these
records will be excluded

# Create 6 bins of 24 hour time period
time_bins_sql_2017 <- "select case when substring(Violation_Time,
1,2) in ('00','01','02','03','12') and
upper(substring(Violation_Time,-1)) = 'A' then 'Bin 1'
when substring(Violation_Time,1,2) in ('04','05','06','07') and
upper(substring(Violation_Time,-1)) = 'A' then 'Bin 2'
when substring(Violation_Time,1,2) in ('08','09','10','11') and
upper(substring(Violation_Time,-1)) = 'A' then 'Bin 3'
when substring(Violation_Time,1,2) in
('12','13','14','15','00','01','02','03') and
upper(substring(Violation_Time,-1)) = 'P' then 'Bin 4'
when substring(Violation_Time,1,2) in
('16','17','18','19','04','05','06','07') and
upper(substring(Violation_Time,-1)) = 'P' then 'Bin 5'
when substring(Violation_Time,1,2) in
('20','21','22','23','08','09','10','11') and
upper(substring(Violation_Time,-1)) = 'P' then 'Bin 6'
else null
end as Violation_time_bin, Violation_Code, Violation_Time
```

```
from nyc_parking_tickets_2017_df_view where Violation_Time is not
null and
substring(Violation_Time, 1, 2) in
('00','01','02','03','04','05','06','07','08','09','10','11','12','1
3','14','15','16','17','18','19','20','21','22','23')"

violations_time_bins_2017 <- SparkR::sql(time_bins_sql_2017)
head(violations_time_bins_2017)
# Violation_time_bin  Violation_Code Violation_Time
#Bin 3             47          1120A
#Bin 6              7          0852P
#Bin 1             78          0015A
#Bin 2             40          0525A
#Bin 4             64          0256P
#Bin 1             20          1232A

createOrReplaceTempView(violations_time_bins_2017,
"violations_time_bins_2017_df_view")

violation_code_count_in_time_bins_2017 <- SparkR::sql("select
Violation_time_bin, Violation_Code, count(*) Count from
violations_time_bins_2017_df_view
                                                    group by
Violation_time_bin, Violation_Code")

# Use Collect action to get results in df to driver node for faster
aggregation
violation_code_count_coll_2017 <-
SparkR::collect(violation_code_count_in_time_bins_2017)

getTop3ViolationCodesInTimeBins <- function(bin) {
  dplyr::filter(violation_code_count_coll_2017, Violation_time_bin
== bin) %>% dplyr::arrange(desc(Count)) %>% head(n = 3)
}

# Get top 3 Violation Codes in Bin 1 ('00','01','02','03','12') AM
getTop3ViolationCodesInTimeBins('Bin 1')
# Violation_time_bin Violation_Code Count
#Bin 1               21            36957
#Bin 1               40            25866
#Bin 1               78            15528

# Stopping closer to 15 feet of fire hydrant is common during very
early mornings.

# Get top 3 Violation Codes in Bin 2 ('04','05','06','07') AM
getTop3ViolationCodesInTimeBins('Bin 2')
# Violation_time_bin Violation_Code Count
# Bin 2              14             74114
# Bin 2              40             60652
# Bin 2              21             57897

# Get top 3 Violation Codes in Bin 3 ('08','09','10','11') AM
getTop3ViolationCodesInTimeBins('Bin 3')
```

```
# Violation_time_bin Violation_Code Count
# Bin 3              21             598069
# Bin 3              36             348165
# Bin 3              38             176570
# Violations are high between 8 and 11 AM. No parking where not
allowed, Failing to show a receipt and exceeding allowed time
# were the most common reasons

# Get top 3 Violation Codes in Bin 4
('12','13','14','15','00','01','02','03') PM
getTop3ViolationCodesInTimeBins('Bin 4')
# Violation_time_bin Violation_Code Count
# Bin 4              36             286284
# Bin 4              38             240721
# Bin 4              37             167026

# Get top 3 Violation Codes in Bin 5
('16','17','18','19','04','05','06','07') PM
getTop3ViolationCodesInTimeBins('Bin 5')
# Violation_time_bin Violation_Code Count
# Bin 5              38             102855
# Bin 5              14             75902
# Bin 5              37             70345
# Violations are the highest between 4 and 7 PM
# Parking in excess of the allowed time or failing to show a receipt
and parking where it is not allowed
# were the most common reasons for receiving parking ticket during
these hours

# Get top 3 Violation Codes in Bin 6
('20','21','22','23','08','09','10','11') PM
getTop3ViolationCodesInTimeBins('Bin 6')
# Violation_time_bin Violation_Code Count
# Bin 6              7              26293
# Bin 6              40             22338
# Bin 6              14             21045
# Going through red light at an intersection is the most common
violation after 10 PM.

# Three most commonly occurring Violation codes
most_popular_violation_codes_2017 <-
summarize(groupBy(violations_time_bins_2017,
violations_time_bins_2017$Violation_Code),
                                              Count =
n(violations_time_bins_2017$Violation_Code))
head(arrange(most_popular_violation_codes_2017,
desc(most_popular_violation_codes_2017$Count)), n = 3)
# Violation_Code  Count
# 21              768065
# 36              662795
# 38              542078
# Violation codes 21, 36 and 38 are the top 3 commonly occurring
violations
```

```
# Get time bins of most commonly occuring violations
filtered_violation_codes_2017 <-
dplyr::filter(violation_code_count_coll_2017,
violation_code_count_coll_2017$Violation_Code %in% c(21,38,36))
dplyr::arrange(dplyr::summarise(dplyr::group_by(filtered_violation_c
odes_2017, filtered_violation_codes_2017$Violation_time_bin),
Violation_count=sum(Count)), desc(Violation_count))  %>% head(n=3)
# Violation_time_bin Violation_count
# Bin 3                       1122804
# Bin 4                       601701
# Bin 5                       116650
# Most commonly occurring violations 21, 38 and 36 are in the bins
3, 4 and 5 which means that
# these violations occur between times 8am and 7pm
# No parking where not allowed, Failing to show receipt and
Exceeding allowed time all occur during the day and evenings

# 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

# Let us divide the year into 4 quarters representing 4 seasons
# For simplicity we shall use convention 1 to 3 months for Spring, 4
to 6 as Summer, 7 to 9 as Autumn and
# 10 to 12 as Winter

season_bins_sql_2017 <- "select case when substring(Issue_Date,1,2)
in ('01','02','03') then 'Bin 1'
when substring(Issue_Date,1,2) in ('04','05','06') then 'Bin 2'
when substring(Issue_Date,1,2) in ('07','08','09') then 'Bin 3'
when substring(Issue_Date,1,2) in ('10','11','12') then 'Bin 4'
else null
end as Violation_season_bin, Violation_Code
from nyc_parking_tickets_2017_df_view where Issue_Date is not null
or Issue_Date not in ('NA', '')"

violations_season_bins_2017 <- SparkR::sql(season_bins_sql_2017)

createOrReplaceTempView(violations_season_bins_2017,
"violations_season_bins_2017_df_view")

violation_code_count_in_season_bins_2017 <- SparkR::sql("select
Violation_season_bin, Violation_Code, count(*) Count from
violations_season_bins_2017_df_view
                                                        group by
Violation_season_bin, Violation_Code")

# Use Collect action to get results in df to driver node for faster
aggregation
violation_code_count_season_coll_2017 <-
SparkR::collect(violation_code_count_in_season_bins_2017)
```

```
getTop3ViolationCodesInSeasonBins <- function(bin) {
  dplyr::filter(violation_code_count_season_coll_2017,
Violation_season_bin == bin) %>% dplyr::arrange(desc(Count)) %>%
head(n = 3)
}

# Get top 3 Violation Codes in Season Bin 1
getTop3ViolationCodesInSeasonBins('Bin 1')
# Violation_season_bin Violation_Code  Count
# Bin 1                21              373874
# Bin 1                36              348240
# Bin 1                38              287000
# Stopping closer to 15 feet of fire hydrant are the common reasons
for receiving parking tickets

# Get top 3 Violation Codes in Season Bin 2
getTop3ViolationCodesInSeasonBins('Bin 2')
# Violation_season_bin Violation_Code  Count
# Bin 2                21             393885
# Bin 2                36             314525
# Bin 2                38             255064
# Failing to show parking ticket, exceeding time limit and parking
where not allowed are the common reasons for receiving parking
tickets

# Get top 3 Violation Codes in Season Bin 3
getTop3ViolationCodesInSeasonBins('Bin 3')
# Violation_season_bin Violation_Code  Count
# Bin 3                21              228
# Bin 3                46              219
# Bin 3                40              109
# There are very few records in Bin 3 i.e for Season 3

# Get top 3 Violation Codes in Season Bin 4
# Violation_season_bin Violation_Code  Count
# Bin 4                40              219
# Bin 4                46              121
# Bin 4                21              100
getTop3ViolationCodesInSeasonBins('Bin 4')
# There are very few records in Bin 4 i.e no records for Season 4

# 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.
#    Find total occurrences of the 3 most common violation codes
#    Then, search the internet for NYC parking violation code fines.
You will find a website (on the nyc.gov URL) that lists these fines.
They’re divided into two categories, one for the highest-density
locations of the city, the other for the rest of the city. For
simplicity, take an average of the two.
#    Using this information, find the total amount collected for all
of the fines. State the code which has the highest total collection.
#    What can you intuitively infer from these findings?
```

```
# Three most commonly occurring Violation codes
most_common_violation_codes_2017 <-
summarize(groupBy(nyc_parking_tickets_2017,
nyc_parking_tickets_2017$Violation_Code),
                                              Count =
n(nyc_parking_tickets_2017$Violation_Code))
head(arrange(most_common_violation_codes_2017,
desc(most_common_violation_codes_2017$Count)), n = 5)
# Violation_Code  Count
# 21              720902
# 38              663904
# 14              466488
# 36              406249
# 37              373229
# Violation codes 21, 38 and 14 are the top 3 commonly occurring
violations

# Define a dataframe that has a specific fine for each Violation
Code from 0 to 100
# Source for Violation Code and Fines is https://www1.nyc.gov/site/
finance/vehicles/services-violation-codes.page
# Average fine has been used from two columns "Manhattan 96th St. &
below" and "All Other Areas"
# Where there are no values "NA" is used
all_violation_codes_2017 <- c(0:100)
all_avg_fines_2017 <-
c("NA","515",    "515",   "515",   "115",   "115",   "390",   "50",
"115",   "115",   "115",   "115",   "95",    "115",   "115",   "NA",
"95",    "95",    "115",   "115",   "62.5",  "55",    "60",    "62.5",
"62.5",  "115",   "115",   "180",   "95",    "515",   "515",   "115",
"50",    "50",    "50",    "50",    "50",    "50",    "50",    "62.5",
"115",   "NA",    "50",    "50",    "50",    "115",   "115",   "115",
"115",   "95",    "115",   "115",   "115",   "115",   "NA",    "115",
"115",   "65",    "55",    "115",   "55",    "55",    "55",    "95",
"95",    "95",    "55",    "165",   "65",    "65",    "65",    "65",
"65",    "65",    "65",    "65",    "NA",    "55",    "65",    "115",
"55",    "95",    "115",   "65",    "55",    "65",    "115",   "NA",
"NA",    "115",   "NA",    "55",    "55",    "65",    "100",   "NA",
"95",    "55",    "95",    "NA",    "NA")
fines_for_violation_codes_df_2017 <-
data.frame(all_violation_codes_2017, all_avg_fines_2017)
names(fines_for_violation_codes_df_2017) <- c("Violation_Code",
"Average_Fine")

# Merge Fine with Common Violation Codes dataframe
fines_for_violation_codes_spark_df_2017 <-
as.DataFrame(fines_for_violation_codes_df_2017)
total_collection_2017 <- drop(join(most_common_violation_codes_2017,
fines_for_violation_codes_spark_df_2017,
most_common_violation_codes_2017$Violation_Code ==
fines_for_violation_codes_spark_df_2017$Violation_Code),
fines_for_violation_codes_spark_df_2017$Violation_Code)
head(total_collection_2017)
```

```
# Violation_Code Count          Average_Fine
#  31            80593           115
#  85            9316            65
#  65            36              95
#  53            19488          115
#  78            26776           65
#  34            11              50

# Total fine for each Violation Code
total_collection_2017$TotalFine <- total_collection_2017$Count *
total_collection_2017$Average_Fine
head(total_collection_2017)
# Violation_Code Count          Average_Fine    TotalFine
#  31            80593           115             9268195
#  85            9316            65              605540
#  65            36              95              2470
#  53            19488           115             2241120
#  78            26776           65              1740440
#  34            11              50              550

createOrReplaceTempView(total_collection_2017,
"total_collection_2017_df_view")

# Total amount collected from fines for all Violation Codes
head(SparkR::sql("select sum(TotalFine) as Total_Amount from
total_collection_2017_df_view"))
# Total_Amount
# 407266465
# Total amount collected from all violations is $407266465

# Violation code that has the highest collection
head(arrange(total_collection_2017,
desc(total_collection_2017$TotalFine)), n = 3)
# Violation_Code  Count     Average_Fine TotalFine
# 14              476664    115          54816360
# 21              768087    55           42244785
# 46              312330    55           35917950
# Violation code 14 has the highest total collection of $54816360

#
-----------------------------------------------------------------
-----------------------------------
#                       Inferences for Parking Violations in New
York City for 2017
#
-----------------------------------------------------------------
-----------------------------------
#
# 1. Top 3 most commonly occurring violation codes were 21, 36 and
38
# 2. Top 3 reasons for parking violations are
#      a. No parking where parking is not allowed by sign, Parking
in excess of the allowed time or
#      b. Exceeding the posted speed limit in or near a designated
```

school zone
c Failing to show a receipt or tag in the windshield.Drivers get a 5-minute grace period past the expired time on Muni-Meter receipts
3. Suburban, 4 Door Sedan and Vans were the vehicle types that received maximum parking tickets
4. FORD, TOYOTA and HONDA vehicles received the most number of parking tickets
5. Zones in Manhattan (Upper East, Midtown North, South and Ericson Palace) have had the maximum number of parking tickets issued in 2015
6. Police Stations of Manhattan (Upper East, Midtown North, South and Ericson Palace) have issued the most number of parking tickets
7. Violations were the highest between 8 and 11 AM. No parking where not allowed, Failing to show a receipt and exceeding allowed time
are the most common reasons
8. Violations were high between 12 and 3 PM. Exceeding the speed limit or failing to show a receipt and parking where it is not allowed
are the most common reasons for receiving parking ticket during these hours
9. Going through red light at an intersection was the most common violation after 10 PM.
10. Stopping closer to 15 feet of fire hydrant was also common during late nights and early mornings.
11. Most common violations all round the year were,
a. Failing to show parking ticket,
b. Parking where not allowed and
c. Exceeding time limit
12. Total fine of $407266465 was collected from all violation codes
13. Violation code 14 (Standing or parking where standing is not allowed by sign, street marking or; traffic control device)
collected the most fine
14. Even though the total count of violation code 14 was lesser than code 21,
revenue collected was more because the fine levied for code 14 was higher than the other two codes.