Sys.setenv(SPARK\_HOME = "/usr/local/spark")

library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK\_HOME"),"R","lib")))

sparkR.session(master = 'local')

df <- read.df("hdfs:///common\_folder/nyc\_parking", source = "csv",

inferSchema = "true", header = "true")

df<-dropDuplicates(df,'Summons Number')

head(df)

createOrReplaceTempView(df, "df\_view")

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

#Finding total number of tickets for each year.

ticket\_15<- SparkR::sql("SELECT COUNT(DISTINCT `Summons Number`) FROM df\_view WHERE `Issue Date` like '%2015' ")

head(ticket\_15)

#count(Summons Number)

# 10903411

#total number of tickets for 2015 is 10903411.

ticket\_16<- SparkR::sql("SELECT COUNT(DISTINCT `Summons Number`) FROM df\_view WHERE `Issue Date` like '%2016' ")

head(ticket\_16)

#total number of tickets for 2016 is 10241012

ticket\_17<- SparkR::sql("SELECT COUNT(DISTINCT `Summons Number`) FROM df\_view WHERE `Issue Date` like '%2017' ")

head(ticket\_17)

#total number of tickets for 2017 is 5433018

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

state<-SparkR::sql("SELECT COUNT(DISTINCT `Registration State`) FROM df\_view")

state<-as.data.frame(state)

state

# 69 states

states<-SparkR::sql("SELECT COUNT(\*), `Registration State` FROM df\_view group by `Registration State` ")

states<-as.data.frame(states)

states

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

ad<-SparkR::sql("SELECT COUNT(`Violation In Front Of Or Opposite`) AS ADD\_COUNT , COUNT(\*) AS TOTAL FROM df\_view")

head(ad)

#ADD\_COUNT TOTAL

# 26414277 32156308

No\_add<- 32156308-26414277

head(No\_add)

#5742031 parking tickets do not have address where there is no address

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

#top 5 violation codes

code<-SparkR::sql("SELECT COUNT(`Violation Code`) as number,`Violation Code` FROM df\_view group by `Violation Code` order by number desc")

head(code)

#number Violation Code

#4528048 21

#3513464 38

#3395137 36

# 2678477 14

# 2020773 37

# 1836030 20

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

#violation time analysis

#preparing Violation Time column as timestamp

df$M <- "M"

df$`Violation Time`<-concat(df$`Violation Time`,df$M)

df<- drop(df, c("M"))

#converting to Violation Time to timestamp

df$`Violation Time`<-to\_timestamp(x = df$`Violation Time`, format = "hhmma")

#extracting hour from Violation Time

df$hour<-hour(df$`Violation Time`)

head(df)

createOrReplaceTempView(df, "df\_view")

# Binning into 6 equal discrete bins of time

#we will group the missing values in Violation Time as a separate bin called NA.

bins <- sql("SELECT hour,`Violation Code`, \

CASE WHEN hour <= 4 THEN 1\

WHEN (hour > 4 and hour <= 8) THEN 2\

WHEN (hour > 8 and hour <= 12) THEN 3\

WHEN (hour > 12 and hour <= 16) THEN 4\

WHEN (hour > 16 and hour <= 20) THEN 5\

WHEN (hour > 20 and hour <= 24) THEN 6\

ELSE 'NA' END as bin\_number FROM df\_view")

head(bins)

createOrReplaceTempView(bins, "bins")

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

#3 top violation codes for different bin hour intervals

first<-SparkR::sql("SELECT COUNT(\*) as number,`Violation Code` FROM bins where bin\_number==1 group by `Violation Code` order by number desc")

head(first)

#number Violation Code

# 142713 21

# 134211 40

# 99539 7

#Top three violation codes for 1st bin of violation time which is between 0 to 4 AM in morning are 21,40 and 7.

second<-SparkR::sql("SELECT COUNT(\*) as number,`Violation Code` FROM bins where bin\_number== 2 group by `Violation Code` order by number desc")

head(second)

#number Violation Code

#1470236 21

# 619494 14

# 485997 36

# 374662 20

# 366386 40

# 262716 71

#Top three violation codes for 2nd bin of violation time is 21,14 and 36.

third<-SparkR::sql("SELECT COUNT(\*) as number,`Violation Code` FROM bins where bin\_number== 3 group by `Violation Code` order by number desc")

head(third)

#number Violation Code

# 2822679 21

#1850435 36

#1362962 38

#Top three violation codes for 3rd bin of violation time is 21,36 and 38.

fourth<-SparkR::sql("SELECT COUNT(\*) as number,`Violation Code` FROM bins where bin\_number==4 group by `Violation Code` order by number desc ")

head(fourth)

#number Violation Code

# 1453078 38

#1123997 37

#1030903 36

# 828822 14

#632502 20

# 571717 46

#Top three violation codes for 4th bin of violation time is 38,37 and 36.

fifth<-SparkR::sql("SELECT COUNT(\*) as number,`Violation Code` FROM bins where bin\_number==5 group by `Violation Code` order by number desc")

head(fifth)

#number Violation Code

#417677 38

#356060 7

#265841 37

#Top three violation codes for 5th bin of violation time is 38,7 and 37.

sixth<-SparkR::sql("SELECT COUNT(\*) as number,`Violation Code` FROM bins where bin\_number==6 group by `Violation Code` order by number desc")

head(sixth)

#number Violation Code

#140935 7

#102056 40

# 100088 14

#Top three violation codes for 6th bin of violation time is 40,7 and 14.

#making a dummy variable to calculate missing values of violation time column.

dummy<-SparkR::sql("SELECT COUNT(\*) as number FROM bins where bin\_number=='NA' ")

head(dummy)

# 185415 values are missing in violation time column.