




S3 bucket screen shots

 Services ▾ Resource Groups ▾   saidheerajreddy.bhumireddy@i... ▾


Amazon S3 > newyork-tickets


Overview


Properties

Permissions


Management

 Type a prefix and press Enter to search. Press ESC to clear.








 Upload

 Create folder

More ▾

US West (Oregon) 

Viewing 1 to 3

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

EC2 Management Console | S3 Management Console | EMR - AWS Console

https://s3.console.aws.amazon.com/s3/buckets/n

Search

aws Services Resource Groups raghav.galhotra@iitb.net @ 6...

Amazon S3 > nypt

Overview Properties Permissions Management

Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder More

US West (Oregon)

Viewing 1 to 3

<input type="checkbox"/>	Name	Last modified	Size	Storage class
<input type="checkbox"/>	Parking_Violations_Issued_-_Fiscal...	Jul 9, 2018 11:12:23 PM GMT+0530	2.7 GB	Standard
<input type="checkbox"/>	Parking_Violations_Issued_-_Fiscal...	Jul 9, 2018 11:16:28 PM GMT+0530	2.0 GB	Standard
<input type="checkbox"/>	Parking_Violations_Issued_-_Fiscal...	Jul 9, 2018 11:18:54 PM GMT+0530	1.9 GB	Standard

Operations 0 In progress 1 Success 0 Error

Firefox: Taxi | cookies | Welcome | S3 Mana | NYC Park | HTTP req | wget can | Error HT | https://s3 | Download

Secure https://s3.console.aws.amazon.com/s3/buckets/pragyarohilla17?region=us-west-2&tab=overview

aws Services Resource Groups pragya.rohilla@iitb.net @ 077... Global Support

Amazon S3 > pragyarohilla17

Overview Properties Permissions Management

Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder More

US West (Oregon)

Viewing 1 to 6

<input type="checkbox"/>	Name	Last modified	Size	Storage class
<input type="checkbox"/>	j-1180DQ2SF0X7K	--	--	--
<input type="checkbox"/>	j-1YZQYU699GLIA	--	--	--
<input type="checkbox"/>	j-3U3Y7VA12K19	--	--	--
<input type="checkbox"/>	Parking_Violations_Issued_-_Fiscal_Year_2015.csv	Jul 15, 2018 8:08:02 PM GMT+0530	2.7 GB	Standard
<input type="checkbox"/>	Parking_Violations_Issued_-_Fiscal_Year_2016.csv	Jul 15, 2018 8:07:23 PM GMT+0530	2.0 GB	Standard
<input type="checkbox"/>	Parking_Violations_Issued_-_Fiscal_Year_2017.csv	Jul 15, 2018 8:06:31 PM GMT+0530	1.9 GB	Standard

Amazon S3 > anurag-iitb / taxi


Overview

 Type a prefix and press Enter to search. Press ESC to clear.

 Upload

 Create folder

More ▾

US East (Ohio) 

Viewing 1 to 3				
<input type="checkbox"/>	Name	Last modified	Size	Storage class
<input type="checkbox"/>	Parking_Violations_Issued_-_Fiscal_Year_2015.csv	Jul 10, 2018 1:57:53 PM GMT+0530	2.7 GB	Standard
<input type="checkbox"/>	Parking_Violations_Issued_-_Fiscal_Year_2016.csv	Jul 10, 2018 2:19:05 PM GMT+0530	2.0 GB	Standard
<input type="checkbox"/>	Parking_Violations_Issued_-_Fiscal_Year_2017.csv	Jul 10, 2018 2:52:07 PM GMT+0530	1.9 GB	Standard
Viewing 1 to 3				

```

1 #####
2 # loading the spark R library and initiating the spark session
3 library(SparkR)
4 sparkR.session(master = "local")
5 #####
6
7 #####
8 # loading the data into respective SparkDataFrames and also creating their temp
  tables to run the sql queries
9
10 d_2015 <- read.df("s3://nypt/Parking_Violations_Issued_-_Fiscal_Year_2015.csv",
11 source = "csv", inferSchema = "true", header = "true")
12 d_2016 <- read.df("s3://nypt/Parking_Violations_Issued_-_Fiscal_Year_2016.csv",
13 source = "csv", inferSchema = "true", header = "true")
14 d_2017 <- read.df("s3://nypt/Parking_Violations_Issued_-_Fiscal_Year_2017.csv",
15 source = "csv", inferSchema = "true", header = "true")
16 #####
17 head(d_2015)
18
19 nrow(d_2015)
20 #11809233
21
22 ncol(d_2015)
23 #51
24
25 str(d_2015)
26 # $ Summons Number : chr "8002531292" "8015318440" "7611181981"
27 "7445908067" "7037692864" "7704791394"
28 # $ Plate ID : chr "EPC5238" "5298MD" "FYW2775" "GWE1987"
29 "T671196C" "JJF6834"
30 # $ Registration State : chr "NY" "NY" "NY" "NY" "NY" "PA"
31 # $ Plate Type : chr "PAS" "COM" "PAS" "PAS" "PAS" "PAS"
32 # $ Issue Date : chr "10/01/2014" "03/06/2015" "07/28/2014"
33 "04/13/2015" "05/19/2015" "11/20/2014"
34 # $ Violation Code : chr "21" "14" "46" "19" "19" "21"
35 # $ Vehicle Body Type : chr "SUBN" "VAN" "SUBN" "4DSD" "4DSD" "4DSD"
36 # $ Vehicle Make : chr "CHEVR" "FRUEH" "SUBAR" "LEXUS" "CHRY"
37 "NISSA"
38 # $ Issuing Agency : chr "T" "T" "T" "T" "T" "T"
39 # $ Street Code1 : chr "20390" "27790" "8130" "59990" "36090"
40 "74230"
41 # $ Street Code2 : chr "29890" "19550" "5430" "16540" "10410"
42 "37980"
43 # $ Street Code3 : chr "31490" "19570" "5580" "16790" "24690"
44 "38030"
45 # $ Vehicle Expiration Date : chr "01/01/20150111 12:00:00 PM"
46 "01/01/88888888 12:00:00 PM" "01/01/20160524 12:0
47 # $ Violation Location : chr "0007" "0025" "0072" "102" "0028" "0067"
48 # $ Violation Precinct : chr "7" "25" "72" "102" "28" "67"
49 # $ Issuer Precinct : chr "7" "25" "72" "102" "28" "67"
50 # $ Issuer Code : chr "345454" "333386" "331845" "355669"
51 "341248" "357104"
52 # $ Issuer Command : chr "T800" "T103" "T302" "T402" "T103" "T302"
53 # $ Issuer Squad : chr "A2" "B" "L" "D" "X" "A"
54 # $ Violation Time : chr "0011A" "0942A" "1020A" "0318P" "0410P"
55 "0839A"
56 # $ Time First Observed : chr "NA" "NA" "NA" "NA" "NA" "NA"
57 # $ Violation County : chr "NY" "NY" "K" "Q" "NY" "K"
58 # $ Violation In Front Of Or Opposite : chr "F" "F" "F" "F" "F" "F"
59 # $ House Number : chr "133" "1916" "184" "120-20" "66" "1013"
60 # $ Street Name : chr "Essex St" "Park Ave" "31st St" "Queens
61 Blvd" "W 116th St" "Rutland Rd"
62 # $ Intersecting Street : chr "NA" "NA" "NA" "NA" "NA" "NA"
63 # $ Date First Observed : chr "01/05/0001 12:00:00 PM" "01/05/0001
64 12:00:00 PM" "01/05/0001 12:00:00 PM" "01
65 # $ Law Section : chr "408" "408" "408" "408" "408" "408"
66 # $ Sub Division : chr "d1" "c" "f1" "c3" "c3" "d1"
67 # $ Violation Legal Code : chr "NA" "NA" "NA" "NA" "NA" "NA"
68 # $ Days Parking In Effect : chr "Y Y Y" "YYYYY" "NA" "YYYYY" "YYYYYYYY" "Y"

```

```

54 # $ From Hours In Effect : chr "1200A" "0700A" "NA" "0300P" "NA" "0830A"
55 # $ To Hours In Effect : chr "0300A" "1000A" "NA" "1000P" "NA" "0900A"
56 # $ Vehicle Color : chr "BL" "BROWN" "BLACK" "GY" "BLACK" "WHITE"
57 # $ Unregistered Vehicle? : chr "NA" "NA" "NA" "NA" "NA" "NA"
58 # $ Vehicle Year : chr "2005" "0" "2010" "2015" "0" "0"
59 # $ Meter Number : chr "NA" "NA" "NA" "NA" "NA" "NA"
60 # $ Feet From Curb : chr "0" "0" "0" "0" "0" "0"
61 # $ Violation Post Code : chr "A 77" "CC3" "J 32" "01 4" "19 7" "C 32"
62 # $ Violation Description : chr "21-No Parking (street clean)" "14-No
Standing" "46A-Double Parking (Non-COM)"
63 # $ No Standing or Stopping Violation: chr "NA" "NA" "NA" "NA" "NA" "NA"
64 # $ Hydrant Violation : chr "NA" "NA" "NA" "NA" "NA" "NA"
65 # $ Double Parking Violation : chr "NA" "NA" "NA" "NA" "NA" "NA"
66 # $ Latitude : chr "NA" "NA" "NA" "NA" "NA" "NA"
67 # $ Longitude : chr "NA" "NA" "NA" "NA" "NA" "NA"
68 # $ Community Board : chr "NA" "NA" "NA" "NA" "NA" "NA"
69 # $ Community Council : chr "NA" "NA" "NA" "NA" "NA" "NA"
70 # $ Census Tract : chr "NA" "NA" "NA" "NA" "NA" "NA"
71 # $ BIN : chr "NA" "NA" "NA" "NA" "NA" "NA"
72 # $ BBL : chr "NA" "NA" "NA" "NA" "NA" "NA"
73 # $ NTA : chr "NA" "NA" "NA" "NA" "NA"
"NA"
74
75
76 printSchema(d_2015)
77 |-- Summons Number: long (nullable = true)
78 |-- Plate ID: string (nullable = true)
79 |-- Registration State: string (nullable = true)
80 |-- Plate Type: string (nullable = true)
81 |-- Issue Date: string (nullable = true)
82 |-- Violation Code: integer (nullable = true)
83 |-- Vehicle Body Type: string (nullable = true)
84 |-- Vehicle Make: string (nullable = true)
85 |-- Issuing Agency: string (nullable = true)
86 |-- Street Code1: integer (nullable = true)
87 |-- Street Code2: integer (nullable = true)
88 |-- Street Code3: integer (nullable = true)
89 |-- Vehicle Expiration Date: string (nullable = true)
90 |-- Violation Location: integer (nullable = true)
91 |-- Violation Precinct: integer (nullable = true)
92 |-- Issuer Precinct: integer (nullable = true)
93 |-- Issuer Code: integer (nullable = true)
94 |-- Issuer Command: string (nullable = true)
95 |-- Issuer Squad: string (nullable = true)
96 |-- Violation Time: string (nullable = true)
97 |-- Time First Observed: string (nullable = true)
98 |-- Violation County: string (nullable = true)
99 |-- Violation In Front Of Or Opposite: string (nullable = true)
100 |-- House Number: string (nullable = true)
101 |-- Street Name: string (nullable = true)
102 |-- Intersecting Street: string (nullable = true)
103 |-- Date First Observed: string (nullable = true)
104 |-- Law Section: integer (nullable = true)
105 |-- Sub Division: string (nullable = true)
106 |-- Violation Legal Code: string (nullable = true)
107 |-- Days Parking In Effect : string (nullable = true)
108 |-- From Hours In Effect: string (nullable = true)
109 |-- To Hours In Effect: string (nullable = true)
110 |-- Vehicle Color: string (nullable = true)
111 |-- Unregistered Vehicle?: integer (nullable = true)
112 |-- Vehicle Year: integer (nullable = true)
113 |-- Meter Number: string (nullable = true)
114 |-- Feet From Curb: integer (nullable = true)
115 |-- Violation Post Code: string (nullable = true)
116 |-- Violation Description: string (nullable = true)
117 |-- No Standing or Stopping Violation: string (nullable = true)
118 |-- Hydrant Violation: string (nullable = true)
119 |-- Double Parking Violation: string (nullable = true)
120 |-- Latitude: string (nullable = true)
121 |-- Longitude: string (nullable = true)
122 |-- Community Board: string (nullable = true)
123 |-- Community Council : string (nullable = true)
124 |-- Census Tract: string (nullable = true)

```

```

125 |-- BIN: string (nullable = true)
126 |-- BBL: string (nullable = true)
127 |-- NTA: string (nullable = true)
128
129 #####
130 createOrReplaceTempView(d_2015, "data_2015")
131 #####
132 #Following are the columns of interest
133 #`Summons Number`,`Registration State`,`Issue Date`,`Violation Code`,`Vehicle Body
134 #`Type`,`Vehicle Make`,`Violation Location`,`Violation Precinct`,
135 #`Issuer Precinct`,`Issuer Code`,`Violation Time`,`House Number`,`Street Name`,`Law
136 #`Section`,`Sub Division`,`Days Parking In Effect`,
137 #`From Hours In Effect`,`To Hours In Effect`
138 #####
139 # Filtering the data containing the columns of interest
140 selected_2015 <- SparkR::sql("select `Summons Number`,`Registration State`,`Issue
141 #`Date`,`Violation Code`,`Vehicle Body Type`,`Vehicle Make`,`Violation
142 #`Location`,`Violation Precinct`,`Issuer Precinct`,`Issuer Code`,`Violation
143 #`Time`,`House Number`,`Street Name`,`Law Section`,`Sub Division`,`Days Parking In
144 #`Effect`,`From Hours In Effect`,`To Hours In Effect` from data_2015")
145 #####
146 createOrReplaceTempView(selected_2015, "data_2015")
147 #####
148 ##Examine the data.
149
150 ##1. Q1 Find total number of tickets for each year.
151 ticket_count_2015 <- SparkR::sql("select count(distinct(`Summons Number`)) from
152 data_2015")
153 head(ticket_count_2015)
154 #10951256
155 head(summarize(select(selected_2015,selected_2015$`Summons Number`), count =
156 countDistinct(selected_2015$`Summons Number`)))
157 #10951256
158 #####
159 # this suggests that there are some duplicate values present in the Summons Number
160 field
161 #####
162 ##2. Q2 Find out how many unique states the cars which got parking tickets came from.
163 unique_states_2015 <- SparkR::sql("select count(distinct(`Registration State`)) from
164 data_2015")
165 head(unique_states_2015)
166 #69 different states does these cars belong to
167 head(summarize(select(selected_2015,selected_2015$`Registration State`), count =
168 countDistinct(selected_2015$`Registration State`)))
169 #69
170 #####
171 ##3. Q3 Some parking tickets don't have addresses on them, which is cause for
172 concern. Find out how many such tickets there are.
173 head(count(where(selected_2015, ((isNull(selected_2015$`House Number`) &
174 isNull(selected_2015$`Street Name`))|((selected_2015$`House Number` == "") &
175 (selected_2015$`Street Name` == ""))))))
176 #4413 records don't have the valid address in them
177 #####
178 #Performing some more quality checks and preparing the final filtered dataset for
179 analysis
180 head(count(where(selected_2015, (isNull(selected_2015$`Summons
181 Number`))|((selected_2015$`Summons Number` == ""))))
182 #0

```

```

171 head(count(where(selected_2015, (isNull(selected_2015$`Registration
172 State`))|(selected_2015$`Registration State` == ""))))
173 #0
174 head(count(where(selected_2015, (isNull(selected_2015$`Issue
175 Date`))|(selected_2015$`Issue Date` == ""))))
176 #0
177 head(count(where(selected_2015, (isNull(selected_2015$`Violation
178 Code`))|(selected_2015$`Violation Code` == ""))))
179 #0
180 head(count(where(selected_2015, (isNull(selected_2015$`Vehicle Body
181 Type`))|(selected_2015$`Vehicle Body Type` == ""))))
182 #45747
183 head(count(where(selected_2015, (isNull(selected_2015$`Vehicle
184 Make`))|(selected_2015$`Vehicle Make` == ""))))
185 #75517
186 head(count(where(selected_2015, (isNull(selected_2015$`Violation
187 Location`))|(selected_2015$`Violation Location` == ""))))
188 #1799170
189 head(count(where(selected_2015, (isNull(selected_2015$`Violation
190 Precinct`))|(selected_2015$`Violation Precinct` == ""))))
191 #0
192 head(count(where(selected_2015, (isNull(selected_2015$`Issuer
193 Precinct`))|(selected_2015$`Issuer Precinct` == ""))))
194 #0
195 head(count(where(selected_2015, (isNull(selected_2015$`Issuer
196 Code`))|(selected_2015$`Issuer Code` == ""))))
197 #0
198 head(count(where(selected_2015, (isNull(selected_2015$`Violation
199 Time`))|(selected_2015$`Violation Time` == ""))))
200 #1715
201 head(count(where(selected_2015, (isNull(selected_2015$`Days Parking In Effect
202 `))|(selected_2015$`Days Parking In Effect ` == ""))))
203 #2838555
204 head(count(where(selected_2015, (isNull(selected_2015$`From Hours In
205 Effect`))|(selected_2015$`From Hours In Effect` == ""))))
206 #5186602
207 head(count(where(selected_2015, (isNull(selected_2015$`To Hours In
208 Effect`))|(selected_2015$`To Hours In Effect` == ""))))
209 #5186602
210 #####
211 #####
212 # we found that there are 4413 records in the house number and street names which
213 needs to be excluded
214 # also there are 1715 records on which violation time is not present and those
215 should be excluded as well
216 # there are some duplicate values in the summons number field, we should remove them
217 as well
218
219 selected_2015 <- dropDuplicates(selected_2015, "Summons Number")
220
221 selected_2015 <- filter(selected_2015, ((isNotNull(selected_2015$`House Number`) |
222 isNotNull(selected_2015$`Street Name`))|((selected_2015$`House Number` != "") |
223 (selected_2015$`Street Name` != ""))))
224
225 selected_2015 <- filter(selected_2015, (isNotNull(selected_2015$`Violation
226 Time`))|(selected_2015$`Violation Time` != ""))
227 createOrReplaceTempView(selected_2015, "data_2015")
228 #####
229 #####
230 #####
231 #####
232 ##Aggregation tasks
233 ##1. Q1 How often does each violation code occur? (frequency of violation codes -
234 find the top 5)
235 v_code_count_2015 <- summarize(groupBy(selected_2015, selected_2015$`Violation
236 Code`), count = n(selected_2015$`Violation Code`))
237 head(arrange(v_code_count_2015, desc(v_code_count_2015$count)))
238
239 vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from
240 data_2015_1 group by `Violation Code` order by cnt desc limit 5")
241 head(vio_code_count)
242 # Violation Code cnt
243 #1 21 1501128

```

```

219 #2 38 1324529
220 #3 14 924113
221 #4 36 761571
222 #5 37 746229
223 #6 7 662209
224 #####
225 # The most common violation code is 21
226 #####
227 #####
228 ##2. Q2 How often does each vehicle body type get a parking ticket? How about the
vehicle make? (find the top 5 for both)
229 vbc_2015 <- summarize(groupBy(selected_2015,selected_2015$`Vehicle Body Type`),count
= n(selected_2015$`Vehicle Body Type`))
230 head(arrange(vbc_2015, desc(vbc_2015$count)))
231
232 v_body_count <- SparkR::sql("select `Vehicle Body Type`, count(*) as cnt from
data_2015 group by `Vehicle Body Type` order by cnt desc limit 5")
233 head(v_body_count)
234
235 #Vehicle Body Type Count
236 #1 SUBN 3450976
237 #2 4DSD 3102383
238 #3 VAN 1604777
239 #4 DELV 840097
240 #5 SDN 452714
241 #6 2DSD 296919
242
243 vmc_2015 <- summarize(groupBy(selected_2015,selected_2015$`Vehicle Make`),count =
n(selected_2015$`Vehicle Make`))
244 head(arrange(vmc_2015, desc(vmc_2015$count)))
245
246 v_make_count <- SparkR::sql("select `Vehicle Make`, count(*) as cnt from data_2015
group by `Vehicle Make` order by cnt desc limit 5")
247 head(v_body_count)
248
249 # Vehicle Make count
250 #1 FORD 1416869
251 #2 TOYOT 1123165
252 #3 HONDA 1017711
253 #4 NISSA 837301
254 #5 CHEVR 836165
255 #6 FRUEH 408150
256 #####
257 #####
258 ##3. Q3 A precinct is a police station that has a certain zone of the city under its
command. Find the (5 highest) frequencies of:
259 #a. Violating Precincts (this is the precinct of the zone where the violation
occurred)
260 vio_pre_2015 <- summarize(groupBy(selected_2015,selected_2015$`Violation
Precinct`),count = n(selected_2015$`Violation Precinct`))
261 head(arrange(vio_pre_2015, desc(vio_pre_2015$count)))
262
263 vio_pre_sql_2015 <- SparkR::sql("select `Violation Precinct`, count(*) as cnt from
data_2015 group by `Violation Precinct` order by cnt desc limit 5")
264 head(vio_pre_sql_2015)
265 #Violation Precinct count
266 #1 0 1630789
267 #2 19 559682
268 #3 18 400845
269 #4 14 384563
270 #5 1 307766
271 #6 114 300519
272
273 #b. Issuing Precincts (this is the precinct that issued the ticket)
274 iss_pre_2015 <- summarize(groupBy(selected_2015,selected_2015$`Issuer
Precinct`),count = n(selected_2015$`Issuer Precinct`))
275 head(arrange(iss_pre_2015, desc(iss_pre_2015$count)))
276

```



```

277 iss_pre_sql_2015 <- SparkR::sql("select `Issuer Precinct`, count(*) as cnt from
278 data_2015 group by `Issuer Precinct` order by cnt desc limit 5")
279 head(iss_pre_sql_2015)
280
281 #Issuer Precinct    count
282 #1                  0    1831810
283 #2                  19    544924
284 #3                  18    391464
285 #4                  14    369692
286 #5                   1    298562
287 #6                 114    295574
288
289 ##4. Q4 Find the violation code frequency across 3 precincts which have issued the
290 most number of tickets - do these precinct zones have an exceptionally high
291 frequency of certain violation codes? Are these codes common across precincts?
292
293 vio_pre_code_2015 <- SparkR::sql("select `Issuer Precinct`,`Violation Code`,
294 count(*) as cnt from data_2015 group by `Issuer Precinct`,`Violation Code` order by
295 cnt desc limit 10")
296 head(vio_pre_code_2015)
297
298 #    Issuer Precinct Violation Code    cnt
299 #1                0          36    683945
300 #2                0           7    604657
301 #3                0           5    166188
302 #4                0          21    156417
303 #5               18          14    112977
304 #6               19          38     83720
305
306 ##5. You'd want to find out the properties of parking violations across different
307 times of the day:
308 ##a. The Violation Time field is specified in a strange format. Find a way to make
309 this into a time attribute that you can use to divide into groups.
310
311 #Violation Time : chr "0011A" "0942A" "1020A" "0318P" "0410P" "0839A"
312 #From summary of data we see that Violation time is stored as characters having
313 alphabets A and P denoting AM and PM probably
314
315
316
317 ##b. Find a way to deal with missing values, if any.
318
319 selected_2015$hr <- substr(selected_2015$`Violation Time`, 1, 2)
320 selected_2015$ampm <- substr(selected_2015$`Violation Time`, 6, 6)
321
322 selected_2015$vt_bin <- ifelse(selected_2015$hr != 12 & selected_2015$ampm == "P",
323 selected_2015$hr + 12, selected_2015$hr)
324
325
326
327 createOrReplaceTempView(selected_2015,"data_2015")
328
329
330 ##c. Divide 24 hours into 6 equal discrete bins of time. The intervals you choose
331 are at your discretion.
332 selected_2015_hr <- SparkR::sql("select `Violation Code`, \
333 CASE WHEN (vt_bin >= 4 and vt_bin < 8) THEN 'early_morning'\
334 WHEN (vt_bin >= 8 and vt_bin < 12) THEN 'morning'\
335 WHEN (vt_bin >= 12 and vt_bin < 16) THEN 'after_noon'\
336 WHEN (vt_bin >= 16 and vt_bin < 20) THEN 'evening'\
337 WHEN (vt_bin >= 20 and vt_bin < 24) THEN 'night'\
338 ELSE 'late_night' END as time_group FROM data_2015")
339 createOrReplaceTempView(selected_2015_hr,"selected_2015_hr")
340
341 ##For each of these groups, find the 3 most commonly occurring violations
342 # For early_morning
343 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr
344 where time_group = 'early_morning' group by `Violation Code` order by cnt desc limit
345 3"))
346
347 #Violation Code    cnt
348 #14              134335
349 #21              106782
350 #40               91336

```

```

338
339 # For Morning
340 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr
where time_group = 'morning' group by `Violation Code` order by cnt desc limit 3"))
341 #Violation Code      cnt
342 #21                  1191837
343 #38                  449046
344 #36                  360365
345
346 # For after_noon
347 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr
where time_group = 'after_noon' group by `Violation Code` order by cnt desc limit 3"))
348 #Violation Code      cnt
349 #38                  568324
350 #37                  417605
351 #36                  323526
352
353 # For evening
354 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr
where time_group = 'evening' group by `Violation Code` order by cnt desc limit 3"))
355 #Violation Code      cnt
356 #38                  241317
357 #37                  175785
358 #7                   168888
359
360 # For night
361 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr
where time_group = 'night' group by `Violation Code` order by cnt desc limit 3"))
362 #Violation Code      cnt
363 #7                   81981
364 #38                  62414
365 #14                  45791
366
367 # For late_night
368 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr
where time_group = 'late_night' group by `Violation Code` order by cnt desc limit 3"))
369 #Violation Code      cnt
370 #21                  63571
371 #40                  36485
372 #78                  34806
373
374
375 most_vio_2015 <- SparkR::sql("select time_group,`Violation Code`,count(*) as cnt
from selected_2015_hr group by time_group,`Violation Code` order by cnt desc")
376
377 head(most_vio_2015)
378
379 ##d. Now, try another direction. For the 3 most commonly occurring violation codes,
find the most common times of day (in terms of the bins from the previous part)
380 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2015_hr
group by `Violation Code` order by cnt desc limit 3"))
381 #Violation Code      cnt
382 #21                  1501128
383 #38                  1324529
384 #14                  924113
385 #Three most common Violation Code are 21, 38 and 14
386 #The most common times of the day for these codes
387 head(SparkR::sql("Select time_group, count(*) as cnt from selected_2015_hr where
`Violation Code` IN (21,38,14) group by time_group order by cnt desc"))
388 ## time_group      cnt
389 # morning          1938451
390 # after_noon        977239
391 # evening           390627
392 # early_morning     243967
393 # night             108792
394 # late_night        90694
395
396 ##6. Let's try and find some seasonality in this data
397 ##a. First, divide the year into some number of seasons, and find frequencies of
tickets for each season.
398 seasons_2015 <- SparkR::sql("select `Violation Code`, \
399 CASE WHEN (month(`Issue Date`) >= 1 and month(`Issue Date`) <=
3) THEN 'Q1\'

```

```

400         WHEN (month(`Issue Date`) >= 4 and month(`Issue Date`) <= 6)
401             THEN 'Q2'\
402         WHEN (month(`Issue Date`) >= 7 and month(`Issue Date`) <= 9) THEN
403             'Q3'\
404         ELSE 'Q4' END as season FROM data_2015")
405
406 createOrReplaceTempView(seasons_2015,"seasons_2015")
407 ##b. Then, find the 3 most common violations for each of these season
408 vio_seas_2015 <- SparkR::sql("select season,`Violation Code`,count(*) as cnt from
409 seasons_2015 group by season,`Violation Code` order by cnt desc")
410 head(vio_seas_2015)
411 #    season Violation Code    cnt
412 #1      Q4              21  1501128
413 #2      Q4              38  1324529
414 #3      Q4              14   924113
415 #4      Q4              36   761571
416 #5      Q4              37   746229
417 #6      Q4               7   662209
418
419 ##7. The fines collected from all the parking violation constitute a revenue source
420 for the NYC police department. Let's take an example of estimating that for the 3
421 most commonly occurring codes.
422 ##a. Find total occurrences of the 3 most common violation codes
423 vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from
424 data_2015 group by `Violation Code` order by cnt desc limit 5")
425 head(vio_code_count)
426 #Code    Count
427 #21      1382405
428 #38      1231003
429 #14      865822
430 #37      696895
431 #36      683945
432
433 ##b. Then, search the internet for NYC parking violation code fines. You will find a
434 website (on the nyc.gov URL) that lists these fines. They're divided into two
435 categories, one for the highest-density locations of the city, the other for the
436 rest of the city. For simplicity, take an average of the two.
437 #Code    Average
438 #21      55
439 #38      50
440 #14      115
441 #37      50
442 #36      50
443
444 ##c. Using this information, find the total amount collected for all of the fines.
445 State the code which has the highest total collection.
446 #Code    Total Collection
447 #21      76032275
448 #38      61550150
449 #14      129873300
450 #37      34844750
451 #36      34197250
452
453 ##d. What can you intuitively infer from these findings?
454 # For 2015, we can infer that although Code #21 and #38 sees highest number of fines
455 but the average cost is highest for code #14 which draws the highest collection
456 among aall the fines.
457
458 #####
459 #####
460 #Analyses for 2016 data
461 #####
462 #####
463 head(d_2016)
464
465 nrow(d_2016)
466 #10626899
467
468 ncol(d_2016)
469 #51
470
471 str(d_2016)

```

```

459 # $ Summons Number : num 1363745270 1363745293 1363745438
1363745475 1363745487 1363745517
460 # $ Plate ID : chr "GGY6450" "KXD355" "JCK7576" "GYK7658"
"GMT8141" "GYK3760"
461 # $ Registration State : chr "99" "SC" "PA" "NY" "NY" "NY"
462 # $ Plate Type : chr "PAS" "PAS" "PAS" "OMS" "PAS" "PAS"
463 # $ Issue Date : chr "07/09/2016" "07/09/2016" "07/09/2016"
"07/09/2016" "07/09/2016" "07/09/2016"
464 # $ Violation Code : int 46 21 21 21 21 21
465 # $ Vehicle Body Type : chr "SDN" "SUBN" "SDN" "SUBN" "P-U" "SUBN"
466 # $ Vehicle Make : chr "HONDA" "CHEVR" "ME/BE" "NISSA" "LINCO"
"HONDA"
467 # $ Issuing Agency : chr "P" "P" "P" "P" "P" "P"
468 # $ Street Code1 : int 0 55730 42730 58130 58130 46730
469 # $ Street Code2 : int 40404 67030 26730 18630 67030 58730
470 # $ Street Code3 : int 40404 58730 26830 67030 58730 85730
471 # $ Vehicle Expiration Date : int 20170602 20160288 0 0 20160206 20160709
472 # $ Violation Location : int 74 79 79 79 79 79
473 # $ Violation Precinct : int 74 79 79 79 79 79
474 # $ Issuer Precinct : int 301 301 0 301 301 301
475 # $ Issuer Code : int 358160 358160 358114 358114 358114 358114
476 # $ Issuer Command : chr "T301" "T301" "TEBN" "T301" "T301" "T301"
477 # $ Issuer Squad : chr "0000" "0000" "0000" "0000" "0000" "0000"
478 # $ Violation Time : chr "1037A" "1206P" "0820A" "0918A" "0925A"
"0948A"
479 # $ Time First Observed : chr "NA" "NA" "NA" "NA" "NA" "NA"
480 # $ Violation County : chr "K" "K" "K" "K" "K" "K"
481 # $ Violation In Front Of Or Opposite: chr "F" "F" "F" "F" "F" "F"
482 # $ House Number : chr "142" "331" "1087" "207" "237" "201"
483 # $ Street Name : chr "MACDOUNGH ST" "LEXINGTON AVE" "FULTON
ST" "MADISON ST" "MADISON ST" "HALSEY S
484 # $ Intersecting Street : chr "NA" "NA" "NA" "NA" "NA" "NA"
485 # $ Date First Observed : int 0 0 0 0 0 0
486 # $ Law Section : int 408 408 408 408 408 408
487 # $ Sub Division : chr "D1" "F1" "D1" "D1" "D1" "D1"
488 # $ Violation Legal Code : chr "NA" "NA" "NA" "NA" "NA" "NA"
489 # $ Days Parking In Effect : chr "BBBBBBB" "YBBYBBB" "YBBYBBB" "YBBYBBB"
"YBBYBBB" "YBBYBBB"
490 # $ From Hours In Effect : chr "ALL" "1100A" "0800A" "0900A" "0900A"
"0900A"
491 # $ To Hours In Effect : chr "ALL" "1230P" "0930A" "1030" "1030A"
"1030A"
492 # $ Vehicle Color : chr "WHITE" "RED" "WHITE" "BK" "BLK" "OTHER"
493 # $ Unregistered Vehicle? : chr "0" "0" "0" "0" "0" "0"
494 # $ Vehicle Year : int 2010 0 0 2016 2006 2006
495 # $ Meter Number : chr "-" "-" "-" "-" "-" "-"
496 # $ Feet From Curb : int 0 0 0 0 0 0
497 # $ Violation Post Code : chr "NA" "NA" "NA" "NA" "NA" "NA"
498 # $ Violation Description : chr "NA" "NA" "NA" "NA" "NA" "NA"
499 # $ No Standing or Stopping Violation: chr "NA" "NA" "NA" "NA" "NA" "NA"
500 # $ Hydrant Violation : chr "NA" "NA" "NA" "NA" "NA" "NA"
501 # $ Double Parking Violation : chr "NA" "NA" "NA" "NA" "NA" "NA"
502 # $ Latitude : chr "NA" "NA" "NA" "NA" "NA" "NA"
503 # $ Longitude : chr "NA" "NA" "NA" "NA" "NA" "NA"
504 # $ Community Board : chr "NA" "NA" "NA" "NA" "NA" "NA"
505 # $ Community Council : chr "NA" "NA" "NA" "NA" "NA" "NA"
506 # $ Census Tract : chr "NA" "NA" "NA" "NA" "NA" "NA"
507 # $ BIN : chr "NA" "NA" "NA" "NA" "NA" "NA"
508 # $ BBL : chr "NA" "NA" "NA" "NA" "NA" "NA"
509 # $ NTA : chr "NA" "NA" "NA" "NA" "NA"
"NA"

```

```

510
511
512 printSchema(d_2016)
513 |-- Summons Number: long (nullable = true)
514 |-- Plate ID: string (nullable = true)
515 |-- Registration State: string (nullable = true)
516 |-- Plate Type: string (nullable = true)
517 |-- Issue Date: string (nullable = true)
518 |-- Violation Code: integer (nullable = true)
519 |-- Vehicle Body Type: string (nullable = true)
520 |-- Vehicle Make: string (nullable = true)
521 |-- Issuing Agency: string (nullable = true)

```

```

522 |-- Street Code1: integer (nullable = true)
523 |-- Street Code2: integer (nullable = true)
524 |-- Street Code3: integer (nullable = true)
525 |-- Vehicle Expiration Date: integer (nullable = true)
526 |-- Violation Location: integer (nullable = true)
527 |-- Violation Precinct: integer (nullable = true)
528 |-- Issuer Precinct: integer (nullable = true)
529 |-- Issuer Code: integer (nullable = true)
530 |-- Issuer Command: string (nullable = true)
531 |-- Issuer Squad: string (nullable = true)
532 |-- Violation Time: string (nullable = true)
533 |-- Time First Observed: string (nullable = true)
534 |-- Violation County: string (nullable = true)
535 |-- Violation In Front Of Or Opposite: string (nullable = true)
536 |-- House Number: string (nullable = true)
537 |-- Street Name: string (nullable = true)
538 |-- Intersecting Street: string (nullable = true)
539 |-- Date First Observed: integer (nullable = true)
540 |-- Law Section: integer (nullable = true)
541 |-- Sub Division: string (nullable = true)
542 |-- Violation Legal Code: string (nullable = true)
543 |-- Days Parking In Effect : string (nullable = true)
544 |-- From Hours In Effect: string (nullable = true)
545 |-- To Hours In Effect: string (nullable = true)
546 |-- Vehicle Color: string (nullable = true)
547 |-- Unregistered Vehicle?: string (nullable = true)
548 |-- Vehicle Year: integer (nullable = true)
549 |-- Meter Number: string (nullable = true)
550 |-- Feet From Curb: integer (nullable = true)
551 |-- Violation Post Code: string (nullable = true)
552 |-- Violation Description: string (nullable = true)
553 |-- No Standing or Stopping Violation: string (nullable = true)
554 |-- Hydrant Violation: string (nullable = true)
555 |-- Double Parking Violation: string (nullable = true)
556 |-- Latitude: string (nullable = true)
557 |-- Longitude: string (nullable = true)
558 |-- Community Board: string (nullable = true)
559 |-- Community Council : string (nullable = true)
560 |-- Census Tract: string (nullable = true)
561 |-- BIN: string (nullable = true)
562 |-- BBL: string (nullable = true)
563 |-- NTA: string (nullable = true)
564
565 #####
566 #####
567 createOrReplaceTempView(d_2016, "data_2016")
568 #####
569 #####
570 #Following are the columns of interest
571 #`Summons Number`,`Registration State`,`Issue Date`,`Violation Code`,`Vehicle Body
572 Type`,`Vehicle Make`,`Violation Location`,`Violation Precinct`,
573 #`Issuer Precinct`,`Issuer Code`,`Violation Time`,`House Number`,`Street Name`,`Law
574 Section`,`Sub Division`,`Days Parking In Effect`,
575 #`From Hours In Effect`,`To Hours In Effect`
576 #####
577 #####
578 # Filtering the data containing the columns of interest
579 selected_2016 <- SparkR::sql("select `Summons Number`,`Registration State`,`Issue
580 Date`,`Violation Code`,`Vehicle Body Type`,`Vehicle Make`,`Violation
581 Location`,`Violation Precinct`,`Issuer Precinct`,`Issuer Code`,`Violation
582 Time`,`House Number`,`Street Name`,`Law Section`,`Sub Division`,`Days Parking In
583 Effect`,`From Hours In Effect`,`To Hours In Effect` from data_2016")
584 #####
585 #####
586 createOrReplaceTempView(selected_2016, "data_2016")
587 #####
588 #####
589 ##Examine the data.
590
591 ##1. Q1 Find total number of tickets for each year.
592 ticket_count_2016 <- SparkR::sql("select count(distinct(`Summons Number`)) from
593 data_2016")

```

```

583 head(ticket_count_2016)
584 #10626899
585 head(summarize(select(selected_2016,selected_2016$`Summons Number`), count =
countDistinct(selected_2016$`Summons Number`)))
586 #10626899
587
588 #####
589 # this suggests that there are some duplicate values present in the Summons Number
field
590 #####
591 #####
592 ##2. Q2 Find out how many unique states the cars which got parking tickets came from.
593 unique_states_2016 <- SparkR::sql("select count(distinct(`Registration State`)) from
data_2016")
594 head(unique_states_2016)
595 #68 different states does these cars belong to
596 head(summarize(select(selected_2016,selected_2016$`Registration State`), count =
countDistinct(selected_2016$`Registration State`)))
597 #68
598 #####
599 #####
600 ##3. Q3 Some parking tickets don't have addresses on them, which is cause for
concern. Find out how many such tickets there are.
601 head(count(where(selected_2016, ((isNull(selected_2016$`House Number`) &
isNull(selected_2016$`Street Name`))|((selected_2016$`House Number` == "") &
(selected_2016$`Street Name` == ""))))))
602 #6462 records don't have the valid address in them
603 #####
604 #Performing some more quality checks and preparing the final filtered dataset for
analysis
605 head(count(where(selected_2016, (isNull(selected_2016$`Summons
Number`))|(selected_2016$`Summons Number` == ""))))
606 #0
607 head(count(where(selected_2016, (isNull(selected_2016$`Registration
State`))|(selected_2016$`Registration State` == ""))))
608 #0
609 head(count(where(selected_2016, (isNull(selected_2016$`Issue
Date`))|(selected_2016$`Issue Date` == ""))))
610 #0
611 head(count(where(selected_2016, (isNull(selected_2016$`Violation
Code`))|(selected_2016$`Violation Code` == ""))))
612 #0
613 head(count(where(selected_2016, (isNull(selected_2016$`Vehicle Body
Type`))|(selected_2016$`Vehicle Body Type` == ""))))
614 #39277
615 head(count(where(selected_2016, (isNull(selected_2016$`Vehicle
Make`))|(selected_2016$`Vehicle Make` == ""))))
616 #63583
617 head(count(where(selected_2016, (isNull(selected_2016$`Violation
Location`))|(selected_2016$`Violation Location` == ""))))
618 #1868656
619 head(count(where(selected_2016, (isNull(selected_2016$`Violation
Precinct`))|(selected_2016$`Violation Precinct` == ""))))
620 #1
621 head(count(where(selected_2016, (isNull(selected_2016$`Issuer
Precinct`))|(selected_2016$`Issuer Precinct` == ""))))
622 #1
623 head(count(where(selected_2016, (isNull(selected_2016$`Issuer
Code`))|(selected_2016$`Issuer Code` == ""))))
624 #1
625 head(count(where(selected_2016, (isNull(selected_2016$`Violation
Time`))|(selected_2016$`Violation Time` == ""))))
626 #4280
627 head(count(where(selected_2016, (isNull(selected_2016$`Days Parking In Effect
`))|(selected_2016$`Days Parking In Effect ` == ""))))
628 #2867416
629 head(count(where(selected_2016, (isNull(selected_2016$`From Hours In

```

```

Effect`))|(selected_2016$`From Hours In Effect` == "")))
630 #4976147
631 head(count(where(selected_2016, (isNull(selected_2016$`To Hours In
Effect`))|(selected_2016$`To Hours In Effect` == ""))))
632 #4976147
633 #####
634 # we found that there are 6462 records in the house number and street names which
needs to be excluded
635 # also there are 4280 records on which violation time is not present and those
should be excluded as well
636 # there are some duplicate values in the summons number field, we should remove them
as well
637
638 selected_2016 <- dropDuplicates(selected_2016, "Summons Number")
639
640 selected_2016 <- filter(selected_2016, ((isNotNull(selected_2016$`House Number`) |
isNotNull(selected_2016$`Street Name`))|((selected_2016$`House Number` != "") |
(selected_2016$`Street Name` != ""))))
641
642 selected_2016 <- filter(selected_2016, (isNotNull(selected_2016$`Violation
Time`))|(selected_2016$`Violation Time` != ""))
643 createOrReplaceTempView(selected_2016,"data_2016")
644 #####
645 #####
646 ##Aggregation tasks
647 ##1. Q1 How often does each violation code occur? (frequency of violation codes -
find the top 5)
648 v_code_count_2016 <- summarize(groupBy(selected_2016,selected_2016$`Violation
Code`),count = n(selected_2016$`Violation Code`))
649 head(arrange(v_code_count_2016, desc(v_code_count_2016$count)))
650
651 #####vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from
data_2016_1 group by `Violation Code` order by cnt desc limit 5")
652 #####head(vio_code_count)
653 # Violation Code cnt
654 #1 21 1530427
655 #2 36 1253511
656 #3 38 1143394
657 #4 14 874901
658 #5 37 686460
659 #6 20 610599
660 #####
661 # The most common violation code is 21
662 #####
663 #####
664 ##2. Q2 How often does each vehicle body type get a parking ticket? How about the
vehicle make? (find the top 5 for both)
665 vbc_2016 <- summarize(groupBy(selected_2016,selected_2016$`Vehicle Body Type`),count
= n(selected_2016$`Vehicle Body Type`))
666 head(arrange(vbc_2016, desc(vbc_2016$count)))
667
668 v_body_count <- SparkR::sql("select `Vehicle Body Type`, count(*) as cnt from
data_2016 group by `Vehicle Body Type` order by cnt desc limit 5")
669 head(v_body_count)
670
671 #Vehicle Body Type Count
672 #1 SUBN 3450976
673 #2 4DSD 3102383
674 #3 VAN 1604777
675 #4 DELV 840097
676 #5 SDN 452714
677 #6 2DSD 296919
678
679 vmc_2016 <- summarize(groupBy(selected_2016,selected_2016$`Vehicle Make`),count =
n(selected_2016$`Vehicle Make`))
680 head(arrange(vmc_2016, desc(vmc_2016$count)))
681

```

```

682 v_make_count <- SparkR::sql("select `Vehicle Make`, count(*) as cnt from data_2016
683 group by `Vehicle Make` order by cnt desc limit 5")
684 head(v_body_count)
685 #Vehicle Body Type    count
686 #1                SUBN 3463919
687 #2                4DSD 2991385
688 #3                VAN 1517704
689 #4                DELV 754966
690 #5                SDN 422240
691 #6                2DSD 276375
692 #####
693 #####
694 ##3. Q3 A precinct is a police station that has a certain zone of the city under its
695 command. Find the (5 highest) frequencies of:
696 #a. Violating Precincts (this is the precinct of the zone where the violation
697 occurred)
698 vio_pre_2016 <- summarize(groupBy(selected_2016,selected_2016$`Violation
699 Precinct`),count = n(selected_2016$`Violation Precinct`))
700 head(arrange(vio_pre_2016, desc(vio_pre_2016$count)))
701 # Violation Precinct    count
702 #1                0 1867301
703 #2                19 554325
704 #3                18 331615
705 #4                14 324389
706 #5                1 303745
707 #6                114 291235
708
709 #b. Issuing Precincts (this is the precinct that issued the ticket)
710 iss_pre_2016 <- summarize(groupBy(selected_2016,selected_2016$`Issuer
711 Precinct`),count = n(selected_2016$`Issuer Precinct`))
712 head(arrange(iss_pre_2016, desc(iss_pre_2016$count)))
713 iss_pre_sql_2016 <- SparkR::sql("select `Issuer Precinct`, count(*) as cnt from
714 data_2016 group by `Issuer Precinct` order by cnt desc limit 5")
715 head(iss_pre_sql_2016)
716 # Issuer Precinct    count
717 #1                0 2138264
718 #2                19 540458
719 #3                18 323058
720 #4                14 315241
721 #5                1 294899
722 #6                114 286835
723
724 ##4. Q4 Find the violation code frequency across 3 precincts which have issued the
725 most number of tickets - do these precinct zones have an exceptionally high
726 frequency of certain violation codes? Are these codes common across precincts?
727 df1 <- groupBy(selected_2016, selected_2016$`Violation Code`)
728 df2 <- agg(df1, precinct = n_distinct(selected_2016$`Issuer Precinct`), count =
729 n(selected_2016$`Violation Code`))
730 head(df2)
731 vio_pre_code_2016 <- SparkR::sql("select `Issuer Precinct`,`Violation Code`,
732 count(*) as cnt from data_2016 group by `Issuer Precinct`,`Violation Code` order by
733 cnt desc limit 10")
734 head(vio_pre_code_2016)
735 # Violation Code precinct    count
736 #1                31        86 139082
737 #2                85        96 27921
738 #3                65        48 126
739 #4                53       118 31676
740 #5                78       159 60532
741 #6                34        17 32
742
743 ##5. You'd want to find out the properties of parking violations across different
744 times of the day:

```



```

740 ##a. The Violation Time field is specified in a strange format. Find a way to make
741 this into a time attribute that you can use to divide into groups.
742 #Violation Time : chr "0011A" "0942A" "1020A" "0318P" "0410P" "0839A"
743 #From summary of data we see that Violation time is stored as characters having
744 alphabets A and P denoting AM and PM probably
745
746
747 ##b. Find a way to deal with missing values, if any.
748
749 selected_2016$hr <- substr(selected_2016$`Violation Time`, 1, 2)
750 selected_2016$ampm <- substr(selected_2016$`Violation Time`, 6, 6)
751
752 selected_2016$vt_bin <- ifelse(selected_2016$hr != 12 & selected_2016$ampm == "P",
753 selected_2016$hr + 12, selected_2016$hr)
754
755 #selected_2016_hr <- dapplyCollect (
756 #dfl,
757 #function(x) {
758 #hr <- as.numeric(substr(x$`Violation Time`, 1, 2))
759 #ampm <- substr(x$`Violation Time`, 5, 5)
760 #x <- cbind(x, "hr" = hr)
761 #x <- cbind(x, "ampm" = ampm)
762 #})
763
764 #selected_2016_hr$hr <- ifelse(selected_2016_hr$hr != 12 & selected_2016_hr$ampm ==
765 "P", selected_2016_hr$hr + 12, selected_2016_hr$hr)
766 #selected_2016_hr$`Violation Time` <- NULL
767 #selected_2016_hr$ampm <- NULL
768
769 createOrReplaceTempView(selected_2016,"data_2016")
770
771 ##c. Divide 24 hours into 6 equal discrete bins of time. The intervals you choose
772 are at your discretion.
773 selected_2016_hr <- SparkR::sql("select `Violation Code`, \
774 CASE WHEN (vt_bin >= 4 and vt_bin < 8) THEN 'early_morning'\
775 WHEN (vt_bin >= 8 and vt_bin < 12) THEN 'morning'\
776 WHEN (vt_bin >= 12 and vt_bin < 16) THEN 'after_noon'\
777 WHEN (vt_bin >= 16 and vt_bin < 20) THEN 'evening'\
778 WHEN (vt_bin >= 20 and vt_bin < 24) THEN 'night'\
779 ELSE 'late_night' END as time_group FROM data_2016")
780 createOrReplaceTempView(selected_2016_hr,"selected_2016_hr")
781
782 ##For each of these groups, find the 3 most commonly occurring violations
783 # For early_morning
784 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr
785 where time_group = 'early_morning' group by `Violation Code` order by cnt desc limit
786 3"))
787 # Violation Code cnt
788 #1 14 140033
789 #2 21 113985
790 #3 40 91680
791
792 # For Morning
793 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr
794 where time_group = 'morning' group by `Violation Code` order by cnt desc limit 3"))
795 #Violation Code cnt
796 #1 21 1209001
797 #2 36 586791
798 #3 38 388080
799
800 # For after_noon
801 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr
802 where time_group = 'after_noon' group by `Violation Code` order by cnt desc limit 3"))
803 #Violation Code cnt
804 #38 568324
805 #37 417605
806 #36 323526
807
808 # For evening

```

```

804 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr
805 # Violation Code cnt
806 #1 38 211262
807 #2 37 161647
808 #3 14 134917
809
810 # For night
811 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr
812 # Violation Code cnt
813 #1 7 60924
814 #2 38 53173
815 #3 40 44952
816
817 # For late_night
818 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr
819 # Violation Code cnt
820 #1 21 67818
821 #2 40 37256
822 #3 78 29451
823
824
825 most_vio_2016 <- SparkR::sql("select time_group,`Violation Code`,count(*) as cnt
826 from selected_2016_hr group by time_group,`Violation Code` order by cnt desc")
827 head(most_vio_2016)
828 # time_group Violation Code cnt
829 #1 morning 21 1209001
830 #2 morning 36 586791
831 #3 after_noon 36 545717
832 #4 after_noon 38 488347
833 #5 morning 38 388080
834 #6 after_noon 37 383352
835
836 ##d. Now, try another direction. For the 3 most commonly occurring violation codes,
837 find the most common times of day (in terms of the bins from the previous part)
838 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2016_hr
839 #Violation Code cnt
840 #1 21 1530427
841 #2 36 1253511
842 #3 38 1143394
843 #Three most common Violation Code are 21, 38 and 14
844 #The most common times of the day for these codes
845 head(SparkR::sql("Select time_group, count(*) as cnt from selected_2016_hr where
846 `Violation Code` IN (21,38,14) group by time_group order by cnt desc"))
847 time_group cnt
848 1 morning 1873228
849 2 after_noon 877934
850 3 evening 346780
851 4 early_morning 256227
852 5 night 97801
853 6 late_night 96752
854
855 ##6. Let's try and find some seasonality in this data
856 ##a. First, divide the year into some number of seasons, and find frequencies of
857 tickets for each season.
858 `Issue Date`
859 seasons_2016 <- SparkR::sql("select `Violation Code`, \
860 CASE WHEN (month(`Issue Date`) >= 1 and month(`Issue Date`) <=
861 3) THEN 'Q1'\
862 WHEN (month(`Issue Date`) >= 4 and month(`Issue Date`) <= 6)
863 THEN 'Q2'\
864 WHEN (month(`Issue Date`) >= 7 and month(`Issue Date`) <= 9) THEN
865 'Q3'\
866 ELSE 'Q4' END as season FROM data_2016")
867
868 ##b. Then, find the 3 most common violations for each of these season
869 createOrReplaceTempView(seasons_2016,"seasons_2016")
870 vio_seas_2016 <- SparkR::sql("select season,`Violation Code`,count(*) as cnt from
871 seasons_2016 group by season,`Violation Code` order by cnt desc")

```

```

865 head(vio_seas_2016)
866 #   season Violation Code      cnt
867 #1      Q4              21 1530427
868 #2      Q4              36 1253511
869 #3      Q4              38 1143394
870 #4      Q4              14  874901
871 #5      Q4              37  686460
872 #6      Q4              20  610599
873 ##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.
874 ##a. Find total occurrences of the 3 most common violation codes
875 vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from
    data_2016 group by `Violation Code` order by cnt desc limit 5")
876 head(vio_code_count)
877 # Violation Code      cnt
878 #1              21 1530427
879 #2              36 1253511
880 #3              38 1143394
881 #4              14  874901
882 #5              37  686460
883
884 ##b. Then, search the internet for NYC parking violation code fines. You will find a
    website (on the nyc.gov URL) that lists these fines. They're divided into two
    categories, one for the highest-density locations of the city, the other for the
    rest of the city. For simplicity, take an average of the two.
885 #Code      Average
886 #21         55
887 #36         50
888 #38         50
889 #14        115
890 #37         50
891
892 ##c. Using this information, find the total amount collected for all of the fines.
    State the code which has the highest total collection.
893 #Code      Total Collection
894 #21        84173485
895 #36        62675550
896 #38        57169700
897 #14        100613615
898 #37        34323000
899
900 ##d. What can you intuitively infer from these findings?
901 # For 2016, we can infer that although Code #21 and #36 sees highest number of fines
    but the average cost is highest for code #14 which draws the highest collection
    among all the fines.
902 #####
    #####
903 #Analyses for 2017 data
904 #####
    #####
905 head(d_2017)
906
907 nrow(d_2017)
908 #10803028
909 ncol(d_2017)
910 #43
911
912 str(d_2017)
913 # $ Summons Number      : num  5092469481 5092451658 4006265037
    8478629828 7868300310 5096917368
914 # $ Plate ID            : chr  "GZH7067" "GZH7067" "FZX9232" "66623ME"
    "37033JV" "FZD8593"
915 # $ Registration State  : chr  "NY" "NY" "NY" "NY" "NY" "NY"
916 # $ Plate Type          : chr  "PAS" "PAS" "PAS" "COM" "COM" "PAS"
917 # $ Issue Date          : chr  "07/10/2016" "07/08/2016" "08/23/2016"
    "06/14/2017" "11/21/2016" "06/13/2017"
918 # $ Violation Code      : int  7 7 5 47 69 7
919 # $ Vehicle Body Type   : chr  "SUBN" "SUBN" "SUBN" "REFG" "DELV" "SUBN"
920 # $ Vehicle Make        : chr  "TOYOT" "TOYOT" "FORD" "MITSU" "INTER"
    "ME/BE"
921 # $ Issuing Agency      : chr  "V" "V" "V" "T" "T" "V"
922 # $ Street Code1       : int  0 0 0 10610 10510 0

```

```

923 # $ Street Code2 : int 0 0 0 34330 34310 0
924 # $ Street Code3 : int 0 0 0 34350 34330 0
925 # $ Vehicle Expiration Date : int 0 0 0 20180630 20170228 0
926 # $ Violation Location : int NA NA NA 14 13 NA
927 # $ Violation Precinct : int 0 0 0 14 13 0
928 # $ Issuer Precinct : int 0 0 0 14 13 0
929 # $ Issuer Code : int 0 0 0 359594 364832 0
930 # $ Issuer Command : chr "NA" "NA" "NA" "T102" "T102" "NA"
931 # $ Issuer Squad : chr "NA" "NA" "NA" "J" "M" "NA"
932 # $ Violation Time : chr "0143A" "0400P" "0233P" "1120A" "0555P"
    "0852P"
933 # $ Time First Observed : chr "NA" "NA" "NA" "NA" "NA" "NA"
934 # $ Violation County : chr "BX" "BX" "BX" "NY" "NY" "QN"
935 # $ Violation In Front Of Or Opposite: chr "NA" "NA" "NA" "O" "F" "NA"
936 # $ House Number : chr "NA" "NA" "NA" "330" "799" "NA"
937 # $ Street Name : chr "ALLERTON AVE (W/B) @" "ALLERTON AVE (W/B)
    @" "SB WEBSTER AVE @ E 1" "7th Ave"
938 # $ Intersecting Street : chr "BARNES AVE" "BARNES AVE" "94TH ST" "NA"
    "NA" "@ MARATHON PKWY"
939 # $ Date First Observed : int 0 0 0 0 0 0
940 # $ Law Section : int 1111 1111 1111 408 408 1111
941 # $ Sub Division : chr "D" "D" "C" "l2" "h1" "D"
942 # $ Violation Legal Code : chr "T" "T" "T" "NA" "NA" "T"
943 # $ Days Parking In Effect : chr "NA" "NA" "NA" "Y" "Y" "NA"
944 # $ From Hours In Effect : chr "NA" "NA" "NA" "0700A" "0700A" "NA"
945 # $ To Hours In Effect : chr "NA" "NA" "NA" "0700P" "0700P" "NA"
946 # $ Vehicle Color : chr "GY" "GY" "BK" "WH" "WHITE" "WH"
947 # $ Unregistered Vehicle? : int NA NA NA NA NA NA
948 # $ Vehicle Year : int 2001 2001 2004 2007 2007 2012
949 # $ Meter Number : chr "NA" "NA" "NA" "NA" "NA" "NA"
950 # $ Feet From Curb : int 0 0 0 0 0 0
951 # $ Violation Post Code : chr "NA" "NA" "NA" "04" "31 6" "NA"
952 # $ Violation Description : chr "FAILURE TO STOP AT RED LIGHT" "FAILURE TO
    STOP AT RED LIGHT" "BUS LANE VIOLAT
953 # $ No Standing or Stopping Violation: chr "NA" "NA" "NA" "NA" "NA" "NA"
954 # $ Hydrant Violation : chr "NA" "NA" "NA" "NA" "NA" "NA"
955 # $ Double Parking Violation : chr "NA" "NA" "NA" "NA" "NA" "NA"
956 #####
957 createOrReplaceTempView(d_2017, "data_2017")
958 #####
959 # Filtering the data containing the columns of interest
960 selected_2017 <- SparkR::sql("select `Summons Number`,`Registration State`,`Issue
    Date`,`Violation Code`,`Vehicle Body Type`,`Vehicle Make`,`Violation
    Location`,`Violation Precinct`,`Issuer Precinct`,`Issuer Code`,`Violation
    Time`,`House Number`,`Street Name`,`Law Section`,`Sub Division`,`Days Parking In
    Effect`,`From Hours In Effect`,`To Hours In Effect` from data_2017")
961 createOrReplaceTempView(selected_2017, "data_2017")
962 #####
963 ##Examine the data.
964
965 ##1. Q1 Find total number of tickets for each year.
966 ticket_count_2017 <- SparkR::sql("select count(distinct(`Summons Number`)) from
    data_2017")
967 head(ticket_count_2017)
968 head(summarize(select(selected_2017,selected_2017$`Summons Number`), count =
    countDistinct(selected_2017$`Summons Number`)))
969 ##10803028
970 #####
971 ##2. Q2 Find out how many unique states the cars which got parking tickets came from.
972 unique_states_2017 <- SparkR::sql("select count(distinct(`Registration State`)) from
    data_2017")
973 head(unique_states_2017)
974 head(summarize(select(selected_2017,selected_2017$`Registration State`), count =
    countDistinct(selected_2017$`Registration State`)))
975 #67
976 #####
977 ##3. Q3 Some parking tickets don't have addresses on them, which is cause for
    concern. Find out how many such tickets there are.

```

```

978 head(count(where(selected_2017, ((isNull(selected_2017$`House Number`) &
  isNull(selected_2017$`Street Name`))|((selected_2017$`House Number` == "") &
  (selected_2017$`Street Name` == ""))))))
979 #2683
980 #####
981 #Performing some more quality checks and preparing the final filtered dataset for
  analysis
982 head(count(where(selected_2017, (isNull(selected_2017$`Summons
  Number`))|(selected_2017$`Summons Number` == ""))))
983 #0
984 head(count(where(selected_2017, (isNull(selected_2017$`Registration
  State`))|(selected_2017$`Registration State` == ""))))
985 #0
986 head(count(where(selected_2017, (isNull(selected_2017$`Issue
  Date`))|(selected_2017$`Issue Date` == ""))))
987 #0
988 head(count(where(selected_2017, (isNull(selected_2017$`Violation
  Code`))|(selected_2017$`Violation Code` == ""))))
989 #0
990 head(count(where(selected_2017, (isNull(selected_2017$`Vehicle Body
  Type`))|(selected_2017$`Vehicle Body Type` == ""))))
991 #42697
992 head(count(where(selected_2017, (isNull(selected_2017$`Vehicle
  Make`))|(selected_2017$`Vehicle Make` == ""))))
993 #73048
994 head(count(where(selected_2017, (isNull(selected_2017$`Violation
  Location`))|(selected_2017$`Violation Location` == ""))))
995 #2072400
996 head(count(where(selected_2017, (isNull(selected_2017$`Violation
  Precinct`))|(selected_2017$`Violation Precinct` == ""))))
997 #0
998 head(count(where(selected_2017, (isNull(selected_2017$`Issuer
  Precinct`))|(selected_2017$`Issuer Precinct` == ""))))
999 #0
1000 head(count(where(selected_2017, (isNull(selected_2017$`Issuer
  Code`))|(selected_2017$`Issuer Code` == ""))))
1001 #0
1002 head(count(where(selected_2017, (isNull(selected_2017$`Violation
  Time`))|(selected_2017$`Violation Time` == ""))))
1003 #63
1004 head(count(where(selected_2017, (isNull(selected_2017$`Days Parking In Effect
  `))|(selected_2017$`Days Parking In Effect ` == ""))))
1005 #2712416
1006 head(count(where(selected_2017, (isNull(selected_2017$`From Hours In
  Effect`))|(selected_2017$`From Hours In Effect` == ""))))
1007 #5450946
1008 head(count(where(selected_2017, (isNull(selected_2017$`To Hours In
  Effect`))|(selected_2017$`To Hours In Effect` == ""))))
1009 #5450943
1010 #####
1011
1012
1013 selected_2017 <- dropDuplicates(selected_2017, "Summons Number")
1014
1015 selected_2017 <- filter(selected_2017, ((isNotNull(selected_2017$`House Number`) |
  isNotNull(selected_2017$`Street Name`))|((selected_2017$`House Number` != "") |
  (selected_2017$`Street Name` != ""))))
1016
1017 selected_2017 <- filter(selected_2017, (isNotNull(selected_2017$`Violation
  Time`))|(selected_2017$`Violation Time` != ""))
1018 createOrReplaceTempView(selected_2017,"data_2017")
1019 #####
1020 ##Aggregation tasks
1021 ##1. Q1 How often does each violation code occur? (frequency of violation codes -
  find the top 5)
1022 v_code_count_2017 <- summarize(groupBy(selected_2017,selected_2017$`Violation
  Code`),count = n(selected_2017$`Violation Code`))
1023 head(arrange(v_code_count_2017, desc(v_code_count_2017$count)))
1024
1025 vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from

```

```

data_2017_1 group by `Violation Code` order by cnt desc limit 5")
1026 head(vio_code_count)
1027 # Violation Code cnt
1028 #1 21 1528184
1029 #2 36 1400614
1030 #3 38 1062063
1031 #4 14 893125
1032 #5 20 618466
1033 #6 46 599778
1034 #####
#####
1035 ##2. Q2 How often does each vehicle body type get a parking ticket? How about the
vehicle make? (find the top 5 for both)
1036 vbc_2017 <- summarize(groupBy(selected_2017,selected_2017$`Vehicle Body Type`),count
= n(selected_2017$`Vehicle Body Type`))
1037 head(arrange(vbc_2017, desc(vbc_2017$count)))
1038
1039 v_body_count <- SparkR::sql("select `Vehicle Body Type`, count(*) as cnt from
data_2017 group by `Vehicle Body Type` order by cnt desc limit 5")
1040 head(v_body_count)
1041
1042 #Vehicle Body Type Count
1043 #1 SUBN 3719191
1044 #2 4DSD 3081839
1045 #3 VAN 1411708
1046 #4 DELV 687139
1047 #5 SDN 437603
1048 #6 2DSD 274362
1049
1050 vmc_2017 <- summarize(groupBy(selected_2017,selected_2017$`Vehicle Make`),count =
n(selected_2017$`Vehicle Make`))
1051 head(arrange(vmc_2017, desc(vmc_2017$count)))
1052
1053 v_make_count <- SparkR::sql("select `Vehicle Make`, count(*) as cnt from data_2017
group by `Vehicle Make` order by cnt desc limit 5")
1054 head(v_body_count)
1055
1056 # Vehicle Make count
1057 #1 FORD 1280743
1058 #2 TOYOT 1211222
1059 #3 HONDA 1079024
1060 #4 NISSA 918433
1061 #5 CHEVR 714510
1062 #6 FRUEH 429090
1063 #####
#####
1064 ##3. Q3 A precinct is a police station that has a certain zone of the city under its
command. Find the (5 highest) frequencies of:
1065 #a. Violating Precincts (this is the precinct of the zone where the violation
occurred)
1066 vio_pre_2017 <- summarize(groupBy(selected_2017,selected_2017$`Violation
Precinct`),count = n(selected_2017$`Violation Precinct`))
1067 head(arrange(vio_pre_2017, desc(vio_pre_2017$count)))
1068
1069 vio_pre_sql_2017 <- SparkR::sql("select `Violation Precinct`, count(*) as cnt from
data_2017 group by `Violation Precinct` order by cnt desc limit 5")
1070 head(vio_pre_sql_2017)
1071 #Violation Precinct count
1072 #1 0 2071293
1073 #2 19 535633
1074 #3 14 352413
1075 #4 1 331752
1076 #5 18 306882
1077 #6 114 296482
1078
1079 b. Issuing Precincts (this is the precinct that issued the ticket)
1080 iss_pre_2017 <- summarize(groupBy(selected_2017,selected_2017$`Issuer
Precinct`),count = n(selected_2017$`Issuer Precinct`))
1081 head(arrange(iss_pre_2017, desc(iss_pre_2017$count)))
1082
1083 iss_pre_sql_2017 <- SparkR::sql("select `Issuer Precinct`, count(*) as cnt from
data_2017 group by `Issuer Precinct` order by cnt desc limit 5")
1084 head(iss_pre_sql_2017)

```

```

1085
1086 #Issuer Precinct      count
1087 #1                    0    2387057
1088 #2                    19    521491
1089 #3                    14    344942
1090 #4                    1    321129
1091 #5                    18    296532
1092 #6                   114    289921
1093 #####
1094 ##4. Q4 Find the violation code frequency across 3 precincts which have issued the
most number of tickets - do these precinct zones have an exceptionally high
frequency of certain violation codes? Are these codes common across precincts?

1095
1096 vio_pre_code_2017 <- SparkR::sql("select `Issuer Precinct`,`Violation Code`,
count(*) as cnt from data_2017 group by `Issuer Precinct`,`Violation Code` order by
cnt desc limit 10")
1097 head(vio_pre_code_2017)
1098
1099 #   Issuer Precinct Violation Code      cnt
1100 #1                0             36    1400614
1101 #2                0              7    516389
1102 #3                0             21    268249
1103 #4                0              5    145642
1104 #5               18             14    91477
1105 #6               19             46    86373
1106 #####
1107 ##5. You'd want to find out the properties of parking violations across different
times of the day:
1108 ##a. The Violation Time field is specified in a strange format. Find a way to make
this into a time attribute that you can use to divide into groups.
1109
1110 #Violation Time : chr "0011A" "0942A" "1020A" "0318P" "0410P" "0839A"
1111 #From summary of data we see that Violation time is stored as characters having
alphabets A and P denoting AM and PM probably
1112
1113
1114 ##b. Find a way to deal with missing values, if any.
1115
1116 selected_2017$hr <- substr(selected_2017$`Violation Time`, 1, 2)
1117 selected_2017$ampm <- substr(selected_2017$`Violation Time`, 6, 6)
1118
1119 selected_2017$vt_bin <- ifelse(selected_2017$hr != 12 & selected_2017$ampm == "P",
selected_2017$hr + 12, selected_2017$hr)
1120
1121 createOrReplaceTempView(selected_2017,"data_2017")
1122
1123 ##c. Divide 24 hours into 6 equal discrete bins of time. The intervals you choose
are at your discretion.
1124 selected_2017_hr <- SparkR::sql("select `Violation Code`, \
1125                                CASE WHEN (vt_bin >= 4 and vt_bin < 8) THEN 'early_morning'\
1126                                WHEN (vt_bin >= 8 and vt_bin < 12) THEN 'morning'\
1127                                WHEN (vt_bin >= 12 and vt_bin < 16) THEN 'after_noon'\
1128                                WHEN (vt_bin >= 16 and vt_bin < 20) THEN 'evening'\
1129                                WHEN (vt_bin >= 20 and vt_bin < 24) THEN 'night'\
1130                                ELSE 'late_night' END as time_group FROM data_2017")
1131
1132 createOrReplaceTempView(selected_2017_hr,"selected_2017_hr")
1133
1134
1135 ##For each of these groups, find the 3 most commonly occurring violations
1136 # For early_morning
1137 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr
where time_group = 'early_morning' group by `Violation Code` order by cnt desc limit
3"))
1138 #Violation Code      cnt
1139 #1                14    141214
1140 #2                21    119414
1141 #3                40    112158
1142
1143 # For Morning
1144 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr

```

```

where time_group = 'morning' group by `Violation Code` order by cnt desc limit 3"))
1145 #Violation Code cnt
1146 #1 21 1182416
1147 #2 36 751422
1148 #3 38 346409
1149
1150 # For after_noon
1151 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr
where time_group = 'after_noon' group by `Violation Code` order by cnt desc limit 3"))
1152 #Violation Code cnt
1153 #1 36 588395
1154 #2 38 462765
1155 #3 37 337045
1156
1157 # For evening
1158 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr
where time_group = 'evening' group by `Violation Code` order by cnt desc limit 3"))
1159 #Violation Code cnt
1160 #1 38 203203
1161 #2 37 145773
1162 #3 14 144704
1163
1164 # For night
1165 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr
where time_group = 'night' group by `Violation Code` order by cnt desc limit 3"))
1166 #Violation Code cnt
1167 #1 7 65593
1168 #2 38 47025
1169 #3 14 44755
1170
1171 # For late_night
1172 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr
where time_group = 'late_night' group by `Violation Code` order by cnt desc limit 3"))
1173 #Violation Code cnt
1174 #1 21 73170
1175 #2 40 45942
1176 #3 14 29310
1177
1178 most_vio_2017 <- SparkR::sql("select time_group,`Violation Code`,count(*) as cnt
from selected_2017_hr group by time_group,`Violation Code` order by cnt desc")
1179
1180 head(most_vio_2017)
1181
1182 ##d. Now, try another direction. For the 3 most commonly occurring violation codes,
find the most common times of day (in terms of the bins from the previous part)
1183 head(SparkR::sql("Select `Violation Code`, count(*) as cnt from selected_2017_hr
group by `Violation Code` order by cnt desc limit 3"))
1184 #Violation Code cnt
1185 #1 21 1528184
1186 #2 36 1400614
1187 #3 38 1062063
1188
1189 #Three most common Violation Code are
1190 #The most common times of the day for these codes
1191 head(SparkR::sql("Select time_group, count(*) as cnt from selected_2017_hr where
`Violation Code` IN (21,38,14) group by time_group order by cnt desc"))
1192 # time_group cnt
1193 #1 morning 1803002
1194 #2 after_noon 874002
1195 #3 evening 348456
1196 #4 early_morning 262923
1197 #5 late_night 102846
1198 #6 night 92143
1199 #####
#####
1200 ##6. Let's try and find some seasonality in this data
1201 ##a. First, divide the year into some number of seasons, and find frequencies of
tickets for each season.
1202 seasons_2017 <- SparkR::sql("select `Violation Code`, \
1203 CASE WHEN (month(`Issue Date`) >= 1 and month(`Issue Date`) <=
3) THEN 'Q1'\
1204 WHEN (month(`Issue Date`) >= 4 and month(`Issue Date`) <= 6)
THEN 'Q2'\

```



```

1205         WHEN (month(`Issue Date`) >= 7 and month(`Issue Date`) <= 9) THEN
1206             'Q3'\
1207         ELSE 'Q4' END as season FROM data_2017")
1208 createOrReplaceTempView(seasons_2017,"seasons_2017")
1209 ##b. Then, find the 3 most common violations for each of these season
1210 vio_seas_2017 <- SparkR::sql("select season,`Violation Code`,count(*) as cnt from
seasons_2017 group by season,`Violation Code` order by cnt desc")
1211 head(vio_seas_2017)
1212 #    season Violation Code      cnt
1213 #1      Q4             21    1528184
1214 #2      Q4             36    1400614
1215 #3      Q4             38    1062063
1216 #4      Q4             14     893125
1217 #5      Q4             20     618466
1218 #6      Q4             46     599778
1219
1220 ##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.
1221 ##a. Find total occurrences of the 3 most common violation codes
1222 vio_code_count <- SparkR::sql("select `Violation Code`, count(*) as cnt from
data_2017 group by `Violation Code` order by cnt desc limit 5")
1223 head(vio_code_count)
1224 #    Violation Code      cnt
1225 #1              21    1528184
1226 #2              36    1400614
1227 #3              38    1062063
1228 #4              14     893125
1229 #5              20     618466
1230
1231 ##b. Then, search the internet for NYC parking violation code fines. You will find a
website (on the nyc.gov URL) that lists these fines. They're divided into two
categories, one for the highest-density locations of the city, the other for the
rest of the city. For simplicity, take an average of the two.
1232 #Code    Average
1233 #21      55
1234 #36      50
1235 #38      50
1236 #14     115
1237 #20     62.5
1238
1239 ##c. Using this information, find the total amount collected for all of the fines.
State the code which has the highest total collection.
1240 #Code    Total Collection
1241 #21     84050120
1242 #36     70030700
1243 #38     53103150
1244 #14    102709375
1245 #20    38654125
1246
1247 ##d. What can you intuitively infer from these findings?
1248 For 2017, we can infer that although Code #21 and #36 sees highest number of fines
but the average cost is highest for code #14 which draws the highest collection
among aall the fines.

```