Road Traffic Dataset

Total lines of code 102.

FINISHED

Data loading and running of some SQL queries took between 1 and 2 minutes as the data set consists of more than 20 million rows.

Took 3 sec. Last updated by anonymous at March 31 2017, 1:26:15 PM.

```
%pyspark
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import glob,os
path =  "/Users/Kapil/Downloads/traffic_feb_june"
roadtraffic = pd.concat(map(pd.read_csv, glob.glob(os.path.join(path, "*.csv"))))
```

FINISHED

Took 1 min 10 sec. Last updated by anonymous at March 31 2017, 12:20:48 PM.

```
%pyspark
inputPath =  "/Users/Kapil/Downloads/traffic_feb_june"
roadtraffic2 = sqlContext.read.format("com.databricks.spark.csv").option("header", "true"
```

FINISHED

Took 1 min 30 sec. Last updated by anonymous at March 31 2017, 12:24:43 PM.

```
%pyspark
roadtraffic2.registerTempTable("road_traffic")
```

FINISHED

Took 1 sec. Last updated by anonymous at March 31 2017, 12:50:35 PM.

```
%pyspark
print(roadtraffic.count())
```

FINISHED

```
status                20713165
avgMeasuredTime       20713165
avgSpeed              20713165
extID                 20713165
medianMeasuredTime    20713165
TIMESTAMP             20713165
vehicleCount          20713165
_id                   20713165
REPORT_ID             20713165
dtype: int64
```

Took 11 sec. Last updated by anonymous at March 31 2017, 12:21:50 PM.

```
%pyspark
roadtraffic[-5:]
```

FINISHED

```
      status  avgMeasuredTime  avgSpeed  extID  medianMeasuredTime  \
16938     OK              112        36    623                 112
16939     OK              112        36    623                 112
16940     OK              112        36    623                 112
16941     OK              112        36    623                 112
16942     OK              112        36    623                 112
                 TIMESTAMP  vehicleCount       _id  REPORT_ID
16938  2014-09-30T23:35:00             0  28062086     210199
16939  2014-09-30T23:40:00             0  28062468     210199
16940  2014-09-30T23:45:00             0  28062917     210199
16941  2014-09-30T23:50:00             0  28063308     210199
16942  2014-09-30T23:55:00             0  28063757     210199
```

Took 0 sec. Last updated by anonymous at March 31 2017, 12:21:55 PM. (outdated)

FINISHED

```
%pyspark
import re
roadtraffic['hour'] = roadtraffic['TIMESTAMP'].str[11:13]
roadtraffic['minutes'] = roadtraffic['TIMESTAMP'].str[14:16]
roadtraffic['date'] = roadtraffic['TIMESTAMP'].str[0:10]
roadtraffic['months'] = roadtraffic['TIMESTAMP'].str[5:7]
```

Took 39 sec. Last updated by anonymous at March 31 2017, 12:36:24 PM.

FINISHED

```
%pyspark
roadtraffic[-5:]
```

```
      status  avgMeasuredTime  avgSpeed  extID  medianMeasuredTime  \
16938     OK              112        36    623                 112
16939     OK              112        36    623                 112
16940     OK              112        36    623                 112
16941     OK              112        36    623                 112
16942     OK              112        36    623                 112
                 TIMESTAMP  vehicleCount       _id  REPORT_ID hour minutes  \
16938  2014-09-30T23:35:00             0  28062086     210199   23      35
16939  2014-09-30T23:40:00             0  28062468     210199   23      40
16940  2014-09-30T23:45:00             0  28062917     210199   23      45
16941  2014-09-30T23:50:00             0  28063308     210199   23      50
16942  2014-09-30T23:55:00             0  28063757     210199   23      55
             date months
16938  2014-09-30     09
16939  2014-09-30     09
16940  2014-09-30     09
16941  2014-09-30     09
16942  2014-09-30     09
```

Took 0 sec. Last updated by anonymous at March 31 2017, 12:36:32 PM.

FINISHED

```
%pyspark
roadtraffic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20713165 entries, 0 to 16942
Data columns (total 13 columns):
status                object
avgMeasuredTime       int64
avgSpeed              int64
extID                 int64
extID                  int64
medianMeasuredTime     int64
TIMESTAMP              object
vehicleCount           int64
_id                    int64
REPORT_ID              int64
hour                   object
minutes                object
date                   object
months                 object
dtypes: int64(7), object(6)
memory usage: 2.2+ GB
```

Took 8 sec. Last updated by anonymous at March 31 2017, 12:36:45 PM.

FINISHED

```
%pyspark
def get_stats(group): return {'min': group.min(), 'max': group.max(), 'count': group.cour
    group.mean()}
grouped_avgspeed_byhour =  roadtraffic['avgSpeed'].groupby(roadtraffic['hour'])
grouped_avgspeed_byhour.apply(get_stats).unstack()
```

```
         count    max       mean  min
hour
00    781546.0  149.0  48.058822  0.0
01    887090.0  149.0  48.348357  0.0
02    904458.0  149.0  48.544265  0.0
03    900229.0  150.0  48.353697  0.0
04    865603.0  150.0  46.639626  0.0
05    846795.0  150.0  42.866610  0.0
06    867806.0  149.0  40.894512  0.0
07    894057.0  150.0  42.059450  0.0
08    895654.0  150.0  42.058984  0.0
09    885253.0  150.0  41.679704  0.0
10    878261.0  149.0  41.382687  0.0
11    894081.0  149.0  41.330556  0.0
12    896679.0  149.0  40.876565  0.0
13    898122.0  149.0  39.753598  0.0
```

Took 6 sec. Last updated by anonymous at March 31 2017, 12:40:34 PM. (outdated)

FINISHED

```
%pyspark
grouped_vehicleCount_byhour = roadtraffic['vehicleCount'].groupby(roadtraffic['hour'])
grouped_vehicleCount_byhour.apply(get_stats).unstack()
```

```
06    867806.0  121.0  5.720163  0.0
07    894057.0   99.0  5.069171  0.0
08    895654.0   79.0  5.036747  0.0
09    885253.0   65.0  5.164977  0.0
10    878261.0   67.0  5.303841  0.0
11    894081.0   77.0  5.357505  0.0

12    896679.0   90.0  5.627419  0.0
13    898122.0   97.0  6.022087  0.0
14    895562.0   94.0  6.050365  0.0
15    910856.0   85.0  4.940116  0.0
16    908012.0   74.0  3.505207  0.0
17    912939.0   71.0  2.405558  0.0
18    920914.0   68.0  1.797495  0.0
19    914818.0   78.0  1.529196  0.0
20    759371.0   86.0  1.253169  0.0
```

```
20     739371.0    86.0  1.232169  0.0
21     702923.0    85.0  0.782027  0.0
22     765726.0    58.0  0.398256  0.0
23     726410.0    67.0  0.249556  0.0
```

Took 4 sec. Last updated by anonymous at March 31 2017, 12:40:30 PM.

FINISHED

```
%pyspark
grouped_avgMeasuredTime = roadtraffic['avgMeasuredTime'].groupby(roadtraffic['hour'])
grouped_avgMeasuredTime.apply(get_stats).unstack()
```

```
          count     max        mean  min
hour
00     781546.0  3595.0   95.816972  0.0
01     887090.0  3587.0   95.751300  0.0
02     904458.0  3587.0   95.795096  0.0
03     900229.0  3587.0   96.949256  0.0
04     865603.0  3587.0  102.146635  0.0
05     846795.0  3587.0  114.061216  0.0
06     867806.0  3596.0  121.030937  0.0
```

Took 4 sec. Last updated by anonymous at March 31 2017, 12:39:16 PM. (outdated)

FINISHED

```
%pyspark
grouped_avgspeed_bymonth =  roadtraffic['avgSpeed'].groupby(roadtraffic['months'])
grouped_avgspeed_bymonth.apply(get_stats).unstack()
```

```
             count    max       mean  min
months
02       1910192.0  149.0  42.935319  0.0
03       3485620.0  150.0  43.568671  0.0
04       3705591.0  150.0  43.705196  0.0
05       3681921.0  150.0  44.117599  0.0
06        793808.0  149.0  43.076908  0.0
08       3608396.0  150.0  44.692306  0.0
09       3527637.0  150.0  44.316682  0.0
```

Took 3 sec. Last updated by anonymous at March 31 2017, 12:43:16 PM. (outdated)

FINISHED

```
%pyspark
grouped_vehicle_bymonth =  roadtraffic['vehicleCount'].groupby(roadtraffic['months'])
grouped_vehicle_bymonth.apply(get_stats).unstack()
```

```
             count    max      mean  min
months
02       1910192.0  111.0  3.380066  0.0
03       3485620.0   97.0  3.443799  0.0
04       3705591.0  111.0  2.854224  0.0
05       3681921.0  100.0  3.252956  0.0
06        793808.0  121.0  2.805493  0.0
08       3608396.0  107.0  3.151881  0.0
09       3527637.0  108.0  3.200058  0.0
```

Took 4 sec. Last updated by anonymous at March 31 2017, 12:45:06 PM. (outdated)

FINISHED

```
%pyspark
grouped_avgTime_bymonth =  roadtraffic['avgMeasuredTime'].groupby(roadtraffic['months'])
grouped_avgTime_bymonth.apply(get_stats).unstack()
```

```
             count    max        mean  min
months
02       1910192.0  3587.0  103.188559  0.0
03       3485620.0  3648.0  104.820731  0.0
04       3705591.0  3595.0  105.554581  0.0
05       3681921.0  3656.0  109.032780  0.0
06        793808.0  3456.0  104.187155  0.0
08       3608396.0  3585.0  107.636983  0.0
09       3527637.0  3572.0  109.661455  0.0
```

Took 4 sec. Last updated by anonymous at March 31 2017, 12:45:28 PM. (outdated)

```
%sql
select * from road_traffic limit 15 --see the first 15 rows in the table
```

FINISHED

status	avgMeasuredTime	avgSpeed	extID	medianM
OK	148	88	672	148
OK	145	90	672	145
OK	164	80	672	164
OK	166	79	672	166
OK	154	85	672	154
OK	150	87	672	150
OK	149	88	672	149

Took 1 sec. Last updated by anonymous at March 31 2017, 12:55:06 PM.

```
%sql
select count(_id)
from road_traffic --count total number of rows in the table
```

FINISHED

count(_id)
20713165

Took 18 sec. Last updated by anonymous at March 31 2017, 12:50:56 PM. (outdated)

```
%sql
select distinct status
from road_traffic--to see what distinct values are in status column
```

FINISHED

status
OK

Took 19 sec. Last updated by anonymous at March 31 2017, 12:53:08 PM. (outdated)

```
%sql
select distinct extID
from road_traffic
order by extID --to see what distinct values are in extID column
```

READY

eID
642
643
644
645
646

```
%sql
select count(eID)
from
(select distinct extID as eID
from road_traffic
group by extID) --to see total number of distinct values in extID column
```

FINISHED

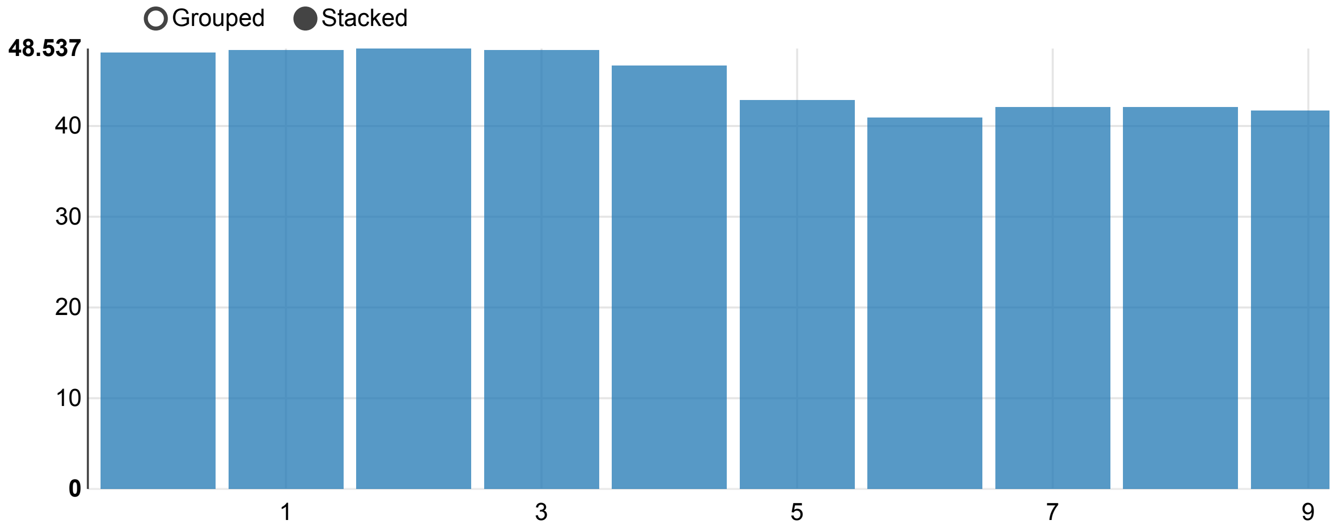
count(eID)
449

Took 18 sec. Last updated by anonymous at March 31 2017, 12:54:23 PM.

```
%sql
select avg(avgSpeed), hour(timestamp) as hour
from road_traffic
group by hour(timestamp)
order by hour(timestamp)
```

FINISHED

settings ▾



Took 1 min 35 sec. Last updated by anonymous at March 31 2017, 1:03:43 PM. (outdated)

```
%sql
select avgSpeed, vehicleCount
from road_traffic
```

FINISHED

settings ▴

All fields:

avgSpeed vehicleCount

xAxis

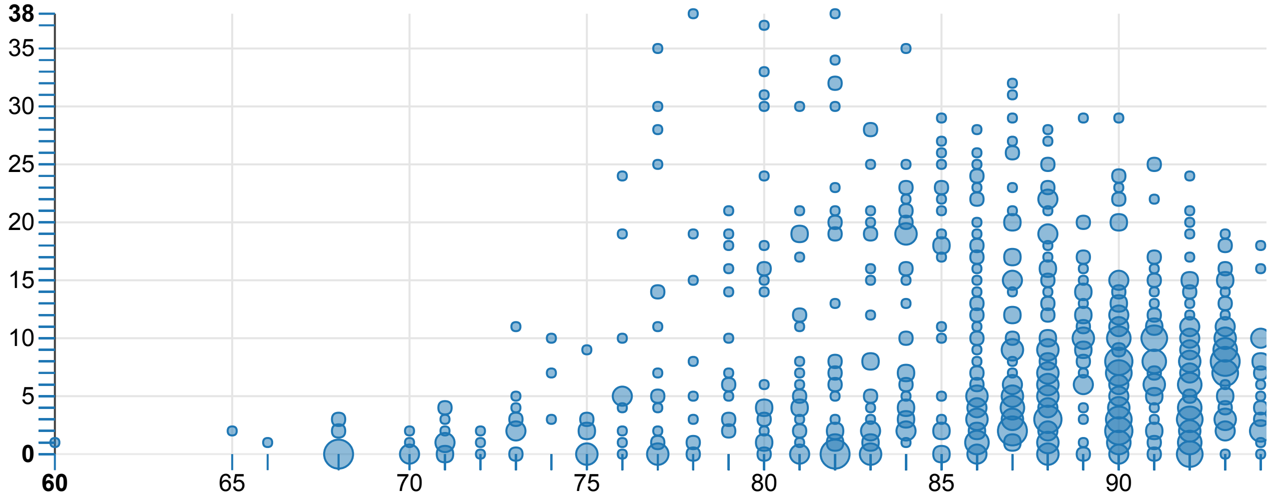
avgSpeed ✖

yAxis

vehicleCount ✖

group

size ℹ



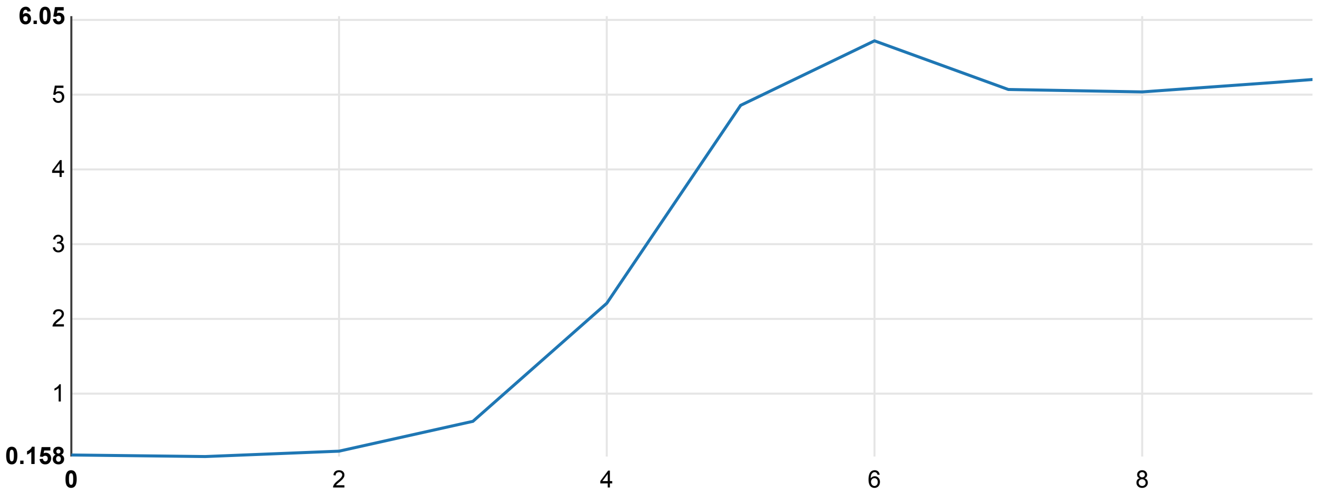
```
Results are limited by 1000.
```

Took 0 sec. Last updated by anonymous at March 31 2017, 1:01:10 PM. (outdated)

```sql
%sql
select avg(vehiclecount), hour(timestamp) as hour
from road_traffic
group by hour(timestamp)
order by hour(timestamp)
```

FINISHED

settings ▾

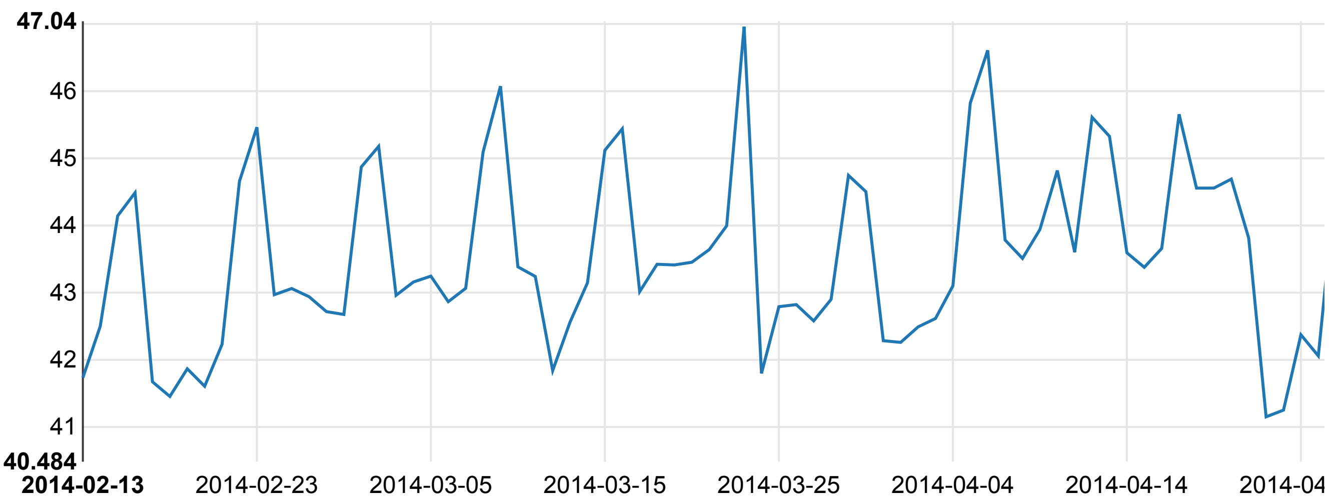


Took 2 min 59 sec. Last updated by anonymous at March 31 2017, 1:05:12 PM. (outdated)

```sql
%sql
select date(timestamp) date, avg(avgSpeed)
from road_traffic
group by date(timestamp)
order by date(timestamp)
```

FINISHED

settings



Took 1 min 30 sec. Last updated by anonymous at March 31 2017, 1:11:35 PM. (outdated)

```
%sql
select *
from road_traffic
order by road_traffic.timestamp
```

FINISHED

status	avgMeasuredTime	avgSpeed	extID	medianM
OK	101	32	1003	101
OK	117	40	1056	117
OK	73	53	1025	73
OK	62	46	817	62
OK	96	49	835	96
OK	0	0	893	0
OK	35	55	1012	35
OK	105	48	809	105
OK	74	55	805	74

```
Results are limited by 1000.
```

Took 1 min 43 sec. Last updated by anonymous at March 31 2017, 1:15:18 PM. (outdated)

```
%pyspark

avgSpeed_corr = lambda x: x.corrwith(x['avgSpeed'])
by_hour =roadtraffic.groupby(lambda x: x.hour)
```

ERROR