|Road_AP_Updated|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|

Zeppelin

Road_AP_Updated ▷ ⛶ 📖 ◆ 🗐 ⬇ 🗔   🗑   🕐   ⌨ ⚙ 🔒 default ▾

READY ▷ ⛶ 📖 ⚙

```
// Imports
import org.apache.spark.sql.functions._
import org.joda.time.format.DateTimeFormat

import org.apache.spark.sql.functions._
import org.joda.time.format.DateTimeFormat
```

READY ▷ ⛶ 📖 ⚙

```
// Load data - adjust the path to the location of your data
val inputPath =  "C:/Users/apandey/Downloads/Road/traffic_feb_june"
val roadtraffic = sqlContext.read
        .format("com.databricks.spark.csv")
        .option("header", "true") // Use first line of all files as header
        .option("delimiter", ",")
        .option("inferSchema", "true") // Automatically infer data types
        .load(inputPath)


inputPath: String = C:/Users/apandey/Downloads/Road/traffic_feb_june
roadtraffic: org.apache.spark.sql.DataFrame = [status: string, avgMeasuredTime: int ... 7 more fields]
```
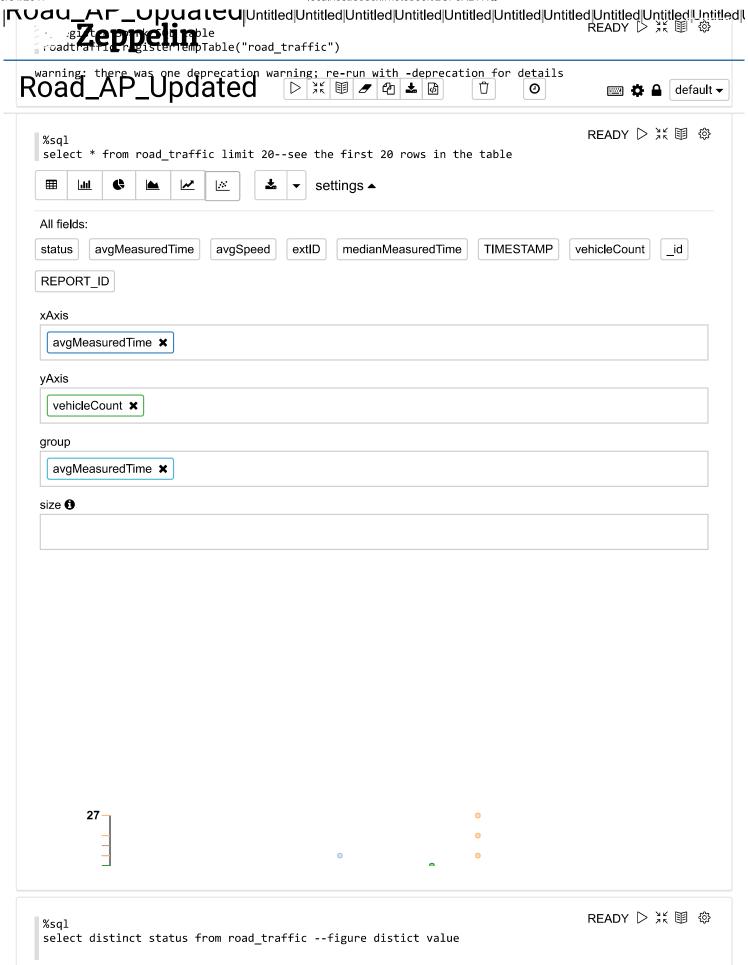

```
roadtraffic.show()
```
READY ▷ ⛶ 📖 ⚙

```
|   OK|        146|     90| 672|          146|2014-02-13 11:35:...|        20|190453  ↴
|  158624|
|   OK|        155|     84| 672|          155|2014-02-13 11:40:...|        20|190902
|  158624|
|   OK|        149|     88| 672|          149|2014-02-13 11:45:...|        25|191351
|  158624|
|   OK|        146|     90| 672|          146|2014-02-13 11:50:...|        23|191800
|  158624|
|   OK|        148|     88| 672|          148|2014-02-13 11:55:...|        22|192249
|  158624|
|   OK|        150|     87| 672|          150|2014-02-13 12:00:...|        21|192698
|  158624|
|   OK|        155|     84| 672|          155|2014-02-13 12:05:...|        19|193147
|  158624|
|   OK|        148|     88| 672|          148|2014-02-13 12:10:...|        19|193596
|  158624|
```

⌃

READY ▷ ⛶ 📖 ⚙

```
%spark
roadtraffic.toDF().registerTempTable("road_traffic")

warning: there was one deprecation warning; re-run with -deprecation for details
```

Road_AP_Updated

Road_AP_Updated|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|U

Zeppelin

                                                                    READY ▷ ⌇⌇ 📖 ⚙

```
roadtraffic.registerTempTable("road_traffic")
```

warning: there was one deprecation warning; re-run with -deprecation for details

Road_AP_Updated   ▷ ⌇⌇ 📖 🗒 🗐 ⬇ ⟨⟩   🗑   🕐   ⌨ ⚙ 🔒 | default ▾

```
%sql
select * from road_traffic limit 20--see the first 20 rows in the table
```

| ⊞ | ᯤ | ◕ | ⛰ | ⤳ | ⣿ | | ⬇ ▾ | settings ▲

All fields:

| status | avgMeasuredTime | avgSpeed | extID | medianMeasuredTime | TIMESTAMP | vehicleCount | _id |

| REPORT_ID |

xAxis

| avgMeasuredTime ✖ |

yAxis

| vehicleCount ✖ |

group

| avgMeasuredTime ✖ |

size ❶

27 —
    —
    —
    —

```
%sql
select distinct status from road_traffic --figure distict value
```

Road_AP_Updated

Road_AP_Updated Untitled Untitled Untitled Untitled Untitled Untitled Untitled Untitled Untitled U

Zeppelin

status

Road_AP_Updated

OK

```
%sql
select distinct extID from road_traffic -- figure distinct value
```

READY

extID

633

623

737

858

897

1,025

808

883

```
%sql
select count(eID) from (select distinct extID as eID from road_traffic)
```

READY

count(eID)

Road_AP_Updated

Zeppelin

Road_AP_Updated

```
%sql
select count(_id) from road_traffic --count # of rows
```

READY

count(_id)

13,577,132

```
%sql select vehicleCount as NumberOfVehiclesPassed, avgSpeed as AverageSpeed from road_traffic
```

READY

settings ▲

☐ **force Y to 0**

☐ **show line chart with focus**

All fields:

NumberOfVehiclesPassed   AverageSpeed

Keys

AverageSpeed ✖

Road_AP_Updated

Zeppelin

Road_AP_Updated

Groups

NumberOfVehiclesPassed ✖

Values

NumberOfVehiclesPassed SUM ✖

● 0 ● 1 ● 2 ● 3 ● 4 ● 5 ● 6 ● 7 ● 8 ● 9 ● 10 ● 11 ● 12 ● 13 ● 14 ● 15 ●
● 37 ● 38

Results are limited by 1000.

```
%pyspark
road_df = sqlContext.read.format("com.databricks.spark.csv").option("header","true").load("/Users/apar
```

READY ▷

```
%pyspark
road_df.show()
```

READY ▷

Road_AP_Updated

Road_AP_Updated |Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|U

Zeppelin

Road_AP_Updated

OK	146	90	672	146	2014-02-13T11:35:00	20	190453	
						158624	
OK	155	84	672	155	2014-02-13T11:40:00	20	190902	
						158624	
OK	149	88	672	149	2014-02-13T11:45:00	25	191351	
						158624	
OK	146	90	672	146	2014-02-13T11:50:00	23	191800	
						158624	
OK	148	88	672	148	2014-02-13T11:55:00	22	192249	
						158624	
OK	150	87	672	150	2014-02-13T12:00:00	21	192698	
						158624	
OK	155	84	672	155	2014-02-13T12:05:00	19	193147	
						158624	
OK	148	88	672	148	2014-02-13T12:10:00	19	193596	
						158624	
OK	145	90	672	145	2014-02-13T12:15:00	19	194045	

FINISHED

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

Took 41 sec. Last updated by anonymous at March 31 2017, 5:50:05 PM. (outdated)

FINISHED

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

Took 52 sec. Last updated by anonymous at March 31 2017, 5:51:54 PM. (outdated)

FINISHED

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

Took 1 sec. Last updated by anonymous at March 31 2017, 5:52:07 PM. (outdated)

FINISHED

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

```
status              13577132
avgMeasuredTime     13577132
avgSpeed            13577132
extID               13577132
medianMeasuredTime  13577132
TIMESTAMP           13577132
vehicleCount        13577132
_id                 13577132
REPORT_ID           13577132
```

Road_AP_Updated

Zeppelin

Road_AP_Updated   ▷ ⧉ 📖 🧽 🗐 📥 ⟨/⟩   🗑   ⏱   FINISHED ⌨ ⚙ 🔒 default ▾

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

	status	avgMeasuredTime	avgSpeed	extID	medianMeasuredTime	\
31791	OK	1839	2	623	1839	
31792	OK	1839	2	623	1839	
31793	OK	1839	2	623	1839	
31794	OK	1839	2	623	1839	
31795	OK	1839	2	623	1839	

	TIMESTAMP	vehicleCount	_id	REPORT_ID
31791	2014-06-09T05:00:00	0	14351211	210199
31792	2014-06-09T05:05:00	0	14351660	210199
31793	2014-06-09T05:10:00	0	14352109	210199
31794	2014-06-09T05:15:00	0	14352558	210199
31795	2014-06-09T05:35:00	0	14353756	210199

Took 0 sec. Last updated by anonymous at March 31 2017, 5:53:47 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 30 sec. Last updated by anonymous at March 31 2017, 5:54:32 PM. (outdated)

FINISHED ▷ ⧉ 📖 ⚙

```
%pyspark
roadtraffic[-5:]
roadtraffic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13577132 entries, 0 to 31795
Data columns (total 13 columns):
status              object
avgMeasuredTime     int64
avgSpeed            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: 1.4+ GB
```

Took 27 sec. Last updated by anonymous at March 31 2017, 5:55:52 PM. (outdated)

Road_AP_Updated

```
get_stats(group): return {'min' : group.min(), 'max' : group.max(), 'count' : group.count(), 'mean' :
grouped_avgspeed_byhour = roadtraffic['avgSpeed'].groupby(roadtraffic['hour'])
grouped_avgspeed_byhour.apply(get_stats).unstack()
grouped_vehicleCount_byhour = roadtraffic['vehicleCount'].groupby(roadtraffic['hour'])
grouped_vehicleCount_byhour.apply(get_stats).unstack()
grouped_avgMeasuredTime = roadtraffic['avgMeasuredTime'].groupby(roadtraffic['hour'])
grouped_avgMeasuredTime.apply(get_stats).unstack()
grouped_vehicle_bymonth =  roadtraffic['vehicleCount'].groupby(roadtraffic['months'])
grouped_vehicle_bymonth.apply(get_stats).unstack()
grouped_avgspeed_bymonth =  roadtraffic['avgSpeed'].groupby(roadtraffic['months'])
grouped_avgspeed_bymonth.apply(get_stats).unstack()
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
```

Took 23 sec. Last updated by anonymous at March 31 2017, 5:57:54 PM. (outdated)

FINISHED ▷ ⋇ 📖 ⚙

```
%pyspark
grouped = roadtraffic.groupby('hour')
get_wavg = lambda g: np.average(g['avgSpeed'], weights=g['vehicleCount'])
grouped.apply(get_wavg)
```

```
07     49.013946
08     51.244284
09     50.738650
10     50.583301
11     50.388740
12     50.007054
13     48.396417
14     47.570227
15     50.165976
16     52.974816
17     54.435287
18     55.424367
19     55.808125
20     55.425116
21     55.664074
22     56.718735
23     58.465512
dtype: float64
```

Took 38 sec. Last updated by anonymous at March 31 2017, 6:01:11 PM. (outdated)

FINISHED ▷ ⋇ 📖 ⚙

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

status	avgMeasuredTime	avgSpeed	extID	medianMeasuredTim
OK	141	93	672	141

Road_AP_Updated | Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|Untitled|U

Zeppelin

| OK | 146 | 90 | 672 | 146 |

Road_AP_Updated ▷ ⌐ 📖 🐾 🗐 ⭳ 🗎 🗑672 🕐 155 ⌨ ⚙ 🔒 default ▾

OK	149	88	672	149
OK	146	90	672	146
OK	148	88	672	148
OK	150	87	672	150

Took 2 sec. Last updated by anonymous at March 31 2017, 6:06:03 PM. (outdated)

FINISHED ▷ ⌐ 📖 ⚙

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

⊞ | 📊 | 🥧 | 📈 | 📉 | ፨   ⭳ ▾

count(_id)

13,577,132

Took 10 sec. Last updated by anonymous at March 31 2017, 6:08:23 PM. (outdated)

FINISHED ▷ ⌐ 📖 ⚙

```sql
%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
```

⊞ | 📊 | 🥧 | 📈 | 📉 | ፨   ⭳ ▾

count(eID)

449

Zeppelin

Road_AP_Updated

Took 18 sec. Last updated by anonymous at March 31 2017, 6:09:13 PM. (outdated)

FINISHED

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

avg(avgSpeed)

47.75767
48.09759
48.25598
48.19696
46.57352
43.14289
40.83283
41.48458

Took 29 sec. Last updated by anonymous at March 31 2017, 6:09:54 PM. (outdated)

READY