

Assignment 2 Road Traffic...



default ▾

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

// Load data - adjust the path to the location of your data
val inputPath = "/Users/Kapil/Downloads/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)

import org.apache.spark.sql.functions._
import org.joda.time.format.DateTimeFormat
inputPath: String = /Users/Kapil/Downloads/traffic_feb_june
roadtraffic: org.apache.spark.sql.DataFrame = [status: string, avgMeasuredTime: int ... 7 more fields]
```

FINISHED ▶ ⌵ 📖 ⚙️

```
//Register as spark SQL table
roadtraffic.registerTempTable("road_traffic")

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





FINISHED ▶ ⌵ 📖 ⚙️

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





FINISHED ▶ ⌵ 📖 ⚙️







status	avgMeasuredTime	avgSpeed	extID	medianMeasuredTime	TIMESTAMP
OK	141	93	672	141	2014-02-13 11:30:00
OK	146	90	672	146	2014-02-13 11:35:00
OK	155	84	672	155	2014-02-13 11:40:00
OK	149	88	672	149	2014-02-13 11:45:00
OK	146	90	672	146	2014-02-13 11:50:00
OK	148	88	672	148	2014-02-13 11:55:00
OK	150	87	672	150	2014-02-13 12:00:00
OK	155	84	672	155	2014-02-13 12:05:00

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



%sql FINISHED    
 select distinct status from road_tra
 --to see what distinct values are in



%sql FINISHED    
 select distinct extID from road_tra
 --to see what distinct values are in



count(_id)

13,577,132

status

OK

eID

610

611

612

613

614

615

616

617

%sql FINISHED ▶ ⌵ 📖 ⚙️
select count(eID) from (select distinct extID as eID from
--to see total number of distinct values in extID column



count(eID)

449

%sql FINISHED ▶ ⌵ 📖 ⚙️
select count(rID) from (select distinct REPORT_ID as rID ·
REPORT_ID)
--to see total number of distinct values in REPORT_ID col



count(rID)

449

READY ▶ ⌵ 📖 ⚙️