dataExploration

```
// val github = img()
// github.url(new java.net.URL("https://assets-cdn.github.com/images/modules/logos_page
// github
```

Took: 3 seconds 49 milliseconds, at 2017-5-5 1:8

Import libraries

```
import org.apache.spark._
import org.apache.spark.SparkContext._
import org.apache.spark.rdd._
import org.apache.spark.sql.SparkSession

import org.apache.spark.sql.types.{DataType}
import org.apache.spark.sql.types.IntegerType
import org.apache.spark.sql.types.FloatType
import org.apache.spark.sql.types.LongType

val spark = SparkSession.builder().getOrCreate()
import spark.implicits._
```

Took: 4 seconds 648 milliseconds, at 2017-5-5 1:8

Import train set

```
val trainRaw = spark.read
                    .format("csv")
                    .option("header", "true")
                    .option("mode", "DROPMALFORMED")
                    .csv("/Users/Albert/Data Science/Kaggle/Facebook/train.csv")

trainRaw.show()
```

Took: 7 seconds 956 milliseconds, at 2017-5-5 1:8

Count the number of imported rows and check the DataFrame schema

```
trainRaw.count()
```

29118021

Took: 41 seconds 733 milliseconds, at 2017-5-5 1:9

```
trainRaw.printSchema()
```

Took: 2 seconds 449 milliseconds, at 2017-5-5 1:9

Recast columns to appropriate data types before summarising

```
val train = trainRaw.select(trainRaw("row_id").cast(IntegerType).as("row_id"),
                            trainRaw("x").cast(FloatType).as("x"),
                            trainRaw("y").cast(FloatType).as("y"),
                            trainRaw("accuracy").cast(IntegerType).as("accuracy"),
                            trainRaw("time").cast(IntegerType).as("time"),
                            round(trainRaw("time").cast(IntegerType), -3).as("timeRound
                            trainRaw("place_id").cast(LongType).as("place_id")
                           )

train.printSchema()
```

Took: 2 seconds 894 milliseconds, at 2017-5-5 1:9

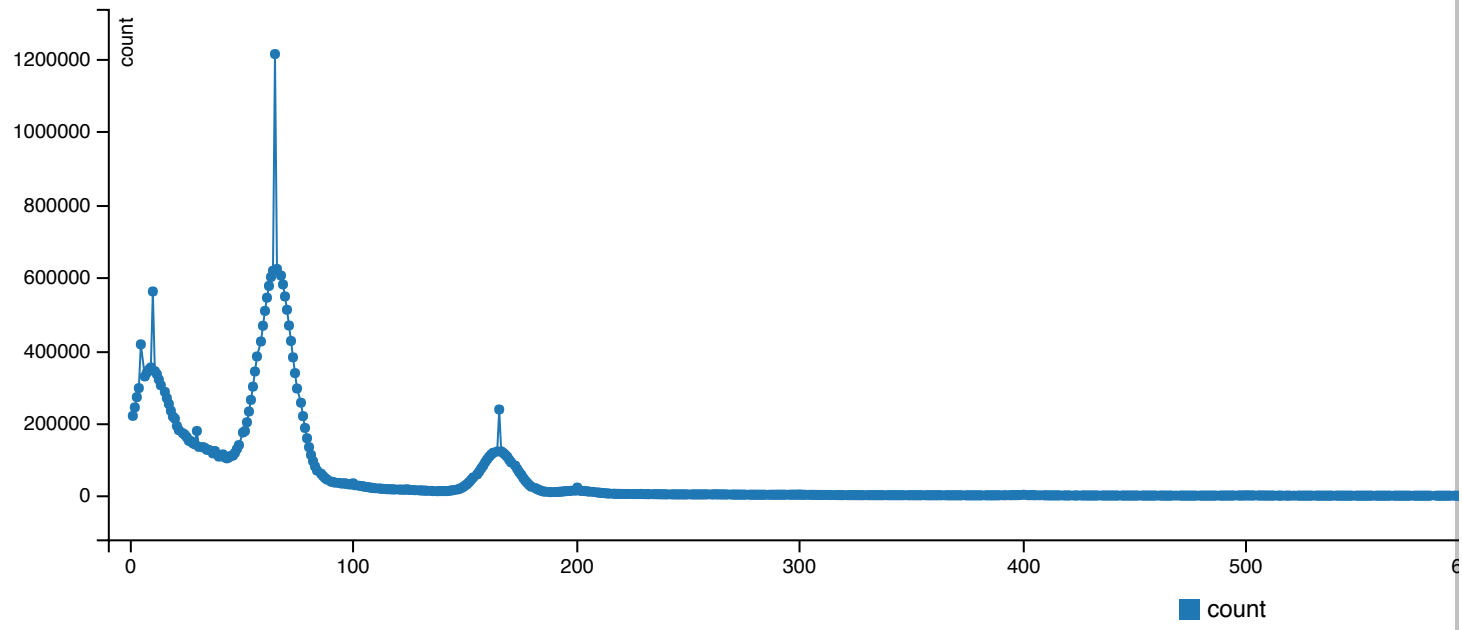
```
val trainAccHist = train.groupBy("accuracy").count()
val trainTimeHist = train.groupBy("time").count()
val trainTimeRoundedHist = train.groupBy("timeRounded").count()
val trainPlaceHist = train.groupBy("place_id").count()

val trainXAvgAcc = train.groupBy("x").agg(mean(train("accuracy")))
val trainYAvgAcc = train.groupBy("y").agg(mean(train("accuracy")))
```

Took: 2 seconds 46 milliseconds, at 2017-5-5 1:9

Plot charts

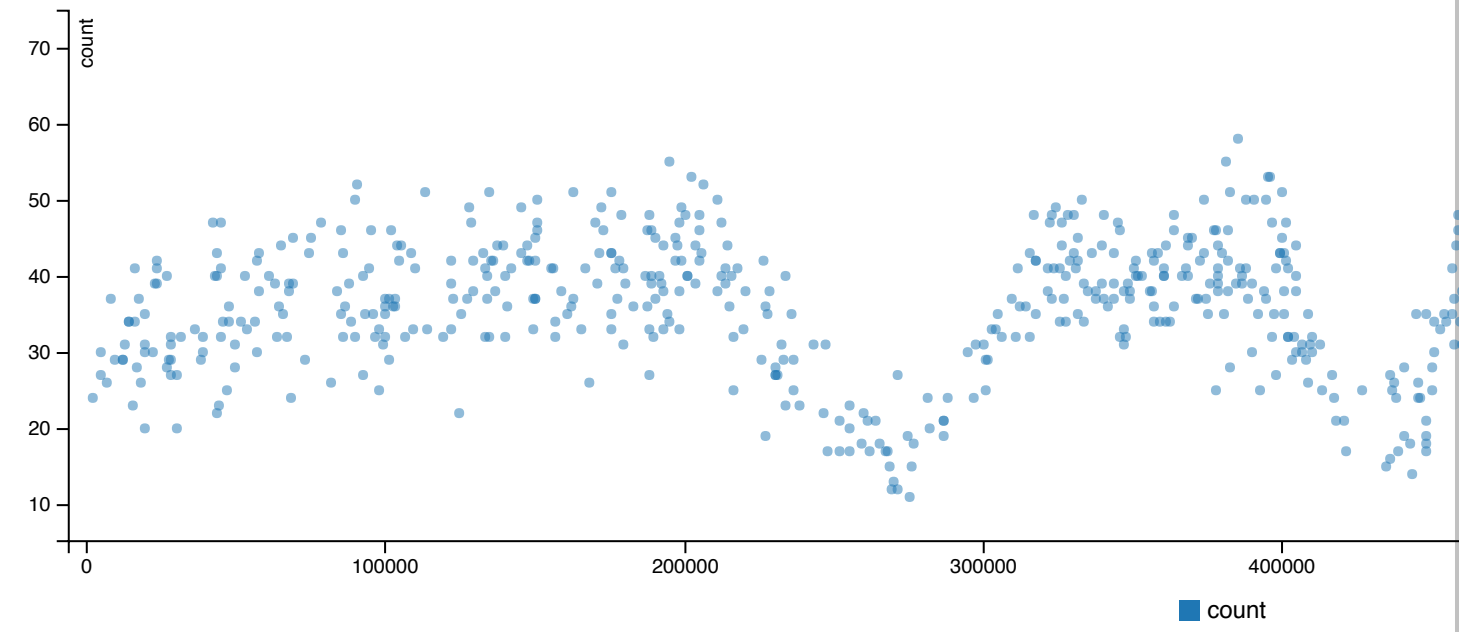
```
trainAccHist.collect()
```

1025 entries total (Warning: randomly sampled 1000 entries)



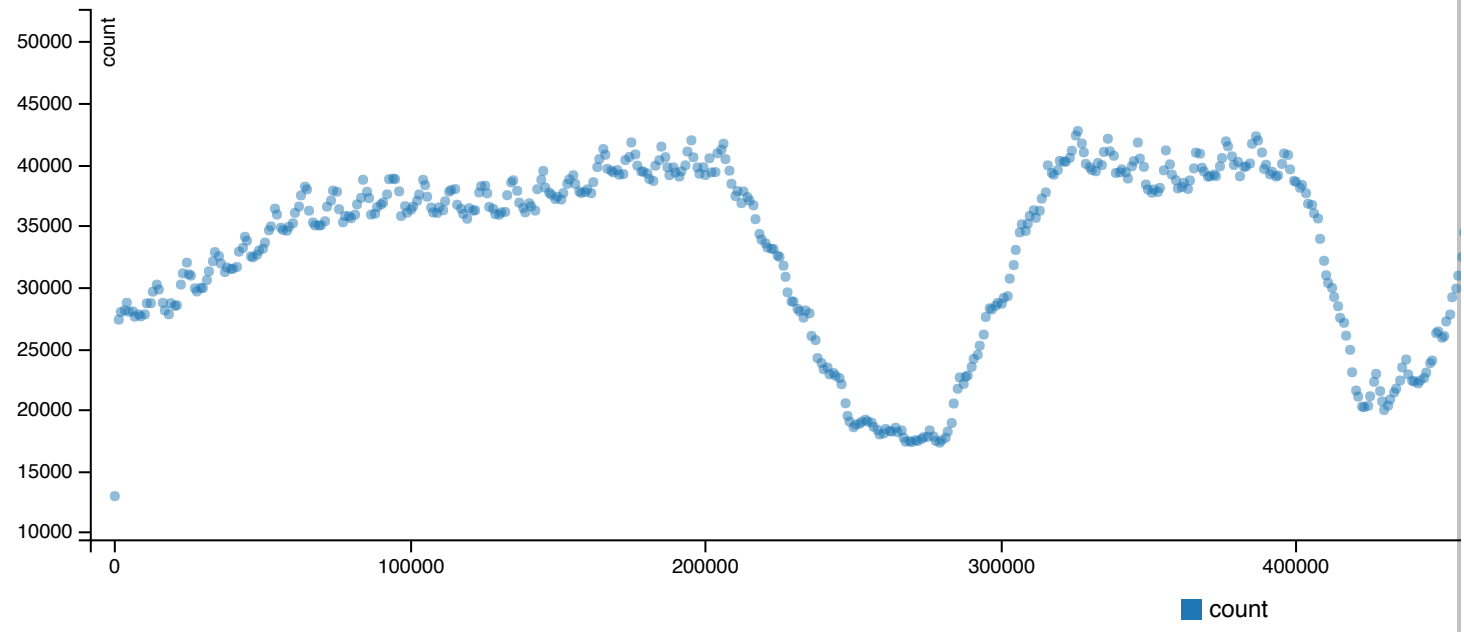

```
trainTimeHist.collect()
```

786239 entries total (Warning: randomly sampled 1000 entries)



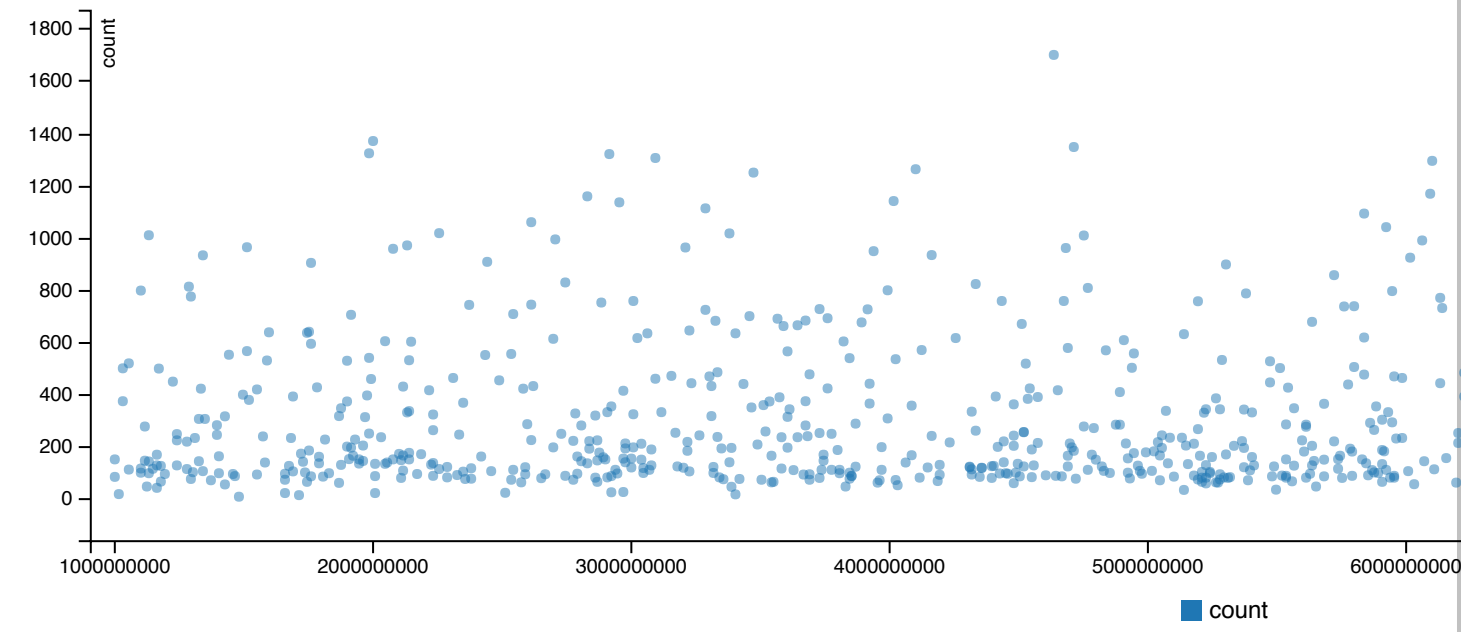
```
trainTimeRoundedHist.collect()
```

787 entries total



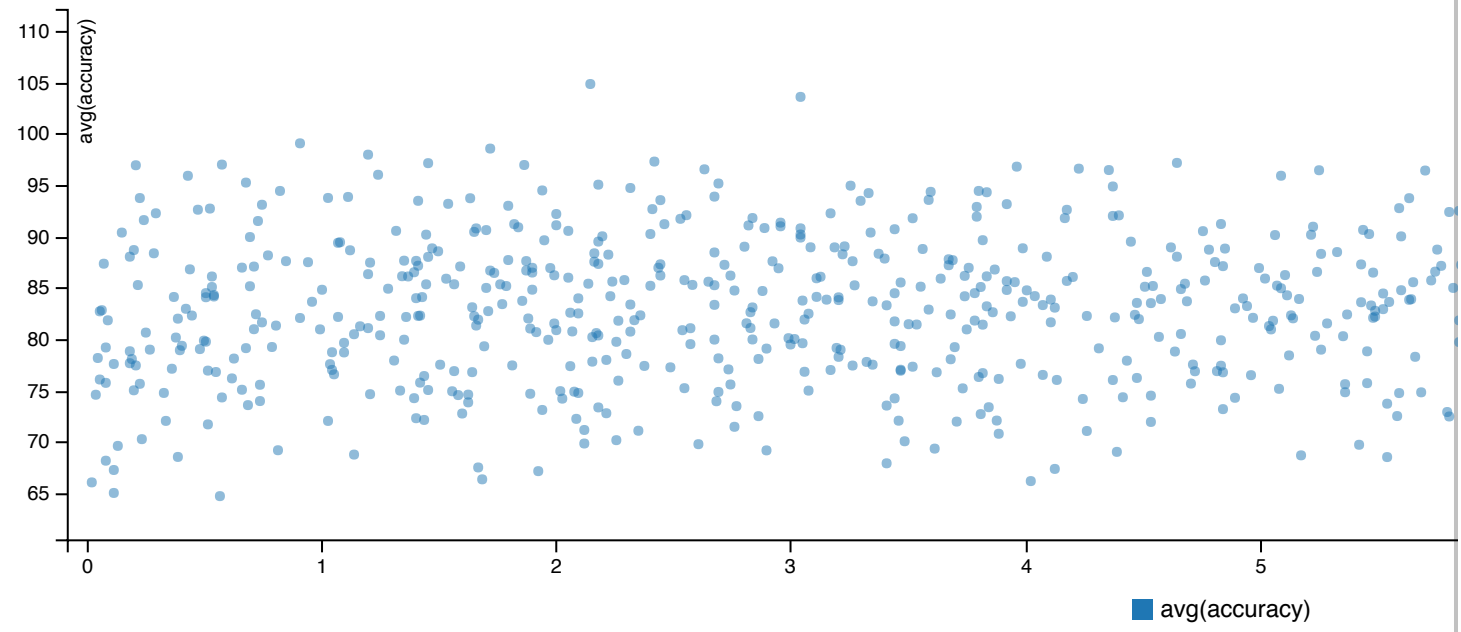

```
trainPlaceHist.collect()
```

108390 entries total (Warning: randomly sampled 1000 entries)



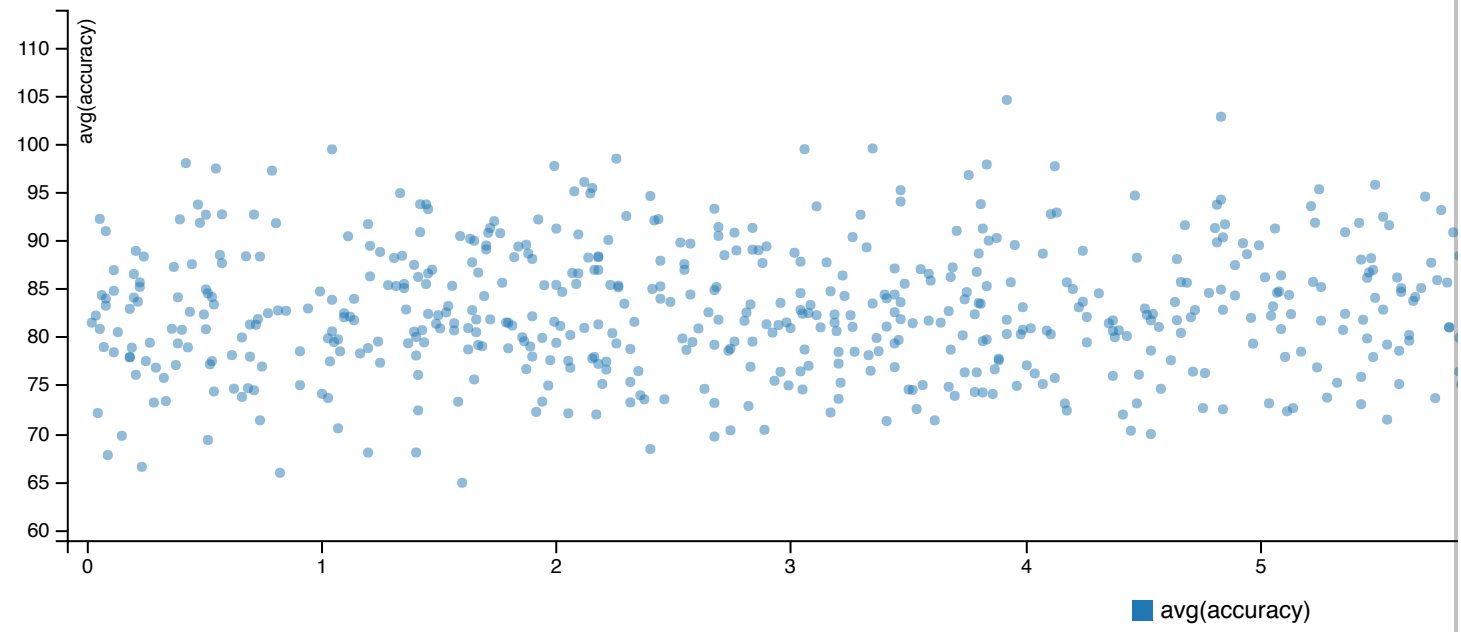
```
trainXAvgAcc.collect()
```

100001 entries total (Warning: randomly sampled 1000 entries)




```
trainYAvgAcc.collect()
```

100001 entries total (Warning: randomly sampled 1000 entries)



Build: | **buildTime**-*Mon Jan 30 17:32:36 UTC 2017* | **formattedShaVersion**-*0.7.0-c955e71d0204599035f603109527e679aa3bd570* | **sbtVersion**-*0.13.8* | **scalaVersion**-*2.11.8* | **sparkNotebookVersion**-*0.7.0* | **hadoopVersion**-*2.7.3* | **jets3tVersion**-*0.7.1* | **jlineDef**-*(jline,2.12)* | **sparkVersion**-*2.1.0* | **withHive**-*true* |.