```
{"paragraphs":[{"config":{"colWidth":12,"graph":
{"mode":"table","height":300,"optionOpen":false,"keys":[],"values":
[],"groups":[],"scatter":{}},"enabled":true,"editorMode":"ace/mode/
scala"},"settings":{"params":{},"forms":
{}},"jobName":"paragraph_1485473841970_1854599217","id":"20170126-1837
21_1096716925","dateCreated":"2017-01-26T18:37:21-0500","status":"FINI
SHED","progressUpdateIntervalMs":500,"$$hashKey":"object:
2079","text":"import org.apache.spark.streaming._\nimport
org.apache.spark.streaming.twitter._\nimport
org.apache.spark.storage.StorageLevel\nimport scala.io.Source\nimport
scala.collection.mutable.HashMap\nimport java.io.File\nimport
org.apache.log4j.Logger\nimport org.apache.log4j.Level\nimport
sys.process.stringSeqToProcess
\n","dateUpdated":"2017-01-26T18:38:11-0500","dateFinished":"2017-01-2
6T18:38:31-0500","dateStarted":"2017-01-26T18:38:11-0500","result":
{"code":"SUCCESS","type":"TEXT","msg":"\nimport
org.apache.spark.streaming._\n\nimport
org.apache.spark.streaming.twitter._\n\nimport
org.apache.spark.storage.StorageLevel\n\nimport scala.io.Source\n
\nimport scala.collection.mutable.HashMap\n\nimport java.io.File\n
\nimport org.apache.log4j.Logger\n\nimport org.apache.log4j.Level\n
\nimport sys.process.stringSeqToProcess\n"},"focus":true},{"config":
{"colWidth":12,"graph":{"mode":"table","height":
300,"optionOpen":false,"keys":[],"values":[],"groups":[],"scatter":
{}},"enabled":true},"settings":{"params":{},"forms":
{}},"jobName":"paragraph_1485473891775_-29454851","id":"20170126-18381
1_117383871","dateCreated":"2017-01-26T18:38:11-0500","status":"FINISH
ED","progressUpdateIntervalMs":500,"focus":true,"$$hashKey":"object:
2156","dateUpdated":"2017-01-26T18:39:08-0500","dateFinished":"2017-01
-26T18:39:10-0500","dateStarted":"2017-01-26T18:39:08-0500","result":
{"code":"SUCCESS","type":"TEXT","msg":"\nconfigureTwitterCredentials:
(apiKey: String, apiSecret: String, accessToken: String,
accessTokenSecret: String)Unit\n"},"text":"def
configureTwitterCredentials(apiKey: String, apiSecret: String,
accessToken: String, accessTokenSecret: String) {\n  val configs = new
HashMap[String, String] ++= Seq(\n    \"apiKey\" -> apiKey,
\"apiSecret\" -> apiSecret, \"accessToken\" -> accessToken,
\"accessTokenSecret\" -> accessTokenSecret)\n  println(\"Configuring
Twitter OAuth\")\n  configs.foreach{ case(key, value) =>\n    if
(value.trim.isEmpty) {\n      throw new Exception(\"Error setting
authentication - value for \" + key + \" not set\")\n    }\n    val
fullKey = \"twitter4j.oauth.\" + key.replace(\"api\", \"consumer\")\n
System.setProperty(fullKey, value.trim)\n    println(\"\\tProperty \"
+ fullKey + \" set as [\" + value.trim + \"]\")\n  }\n  println()
\n}"},{"config":{"colWidth":12,"graph":{"mode":"table","height":
300,"optionOpen":false,"keys":[],"values":[],"groups":[],"scatter":
{}},"enabled":true},"settings":{"params":{},"forms":
{}},"jobName":"paragraph_1485473948434_-1562401738","id":"20170126-183
908_1503131705","dateCreated":"2017-01-26T18:39:08-0500","status":"FIN
ISHED","progressUpdateIntervalMs":500,"focus":true,"$
```

```
$hashKey":"object:
2239","dateUpdated":"2017-01-26T18:40:19-0500","dateFinished":"2017-01
-26T18:40:24-0500","dateStarted":"2017-01-26T18:40:19-0500","result":
{"code":"INCOMPLETE","type":"TEXT","msg":"\napiKey: String =
xxxxxxxxxxxxxxxxxxxxxxxxx\n\napiSecret: String =
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n\naccessToken:
String = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n
\naccessTokenSecret: String =
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\nConfiguring Twitter
OAuth\n\tProperty twitter4j.oauth.consumerKey set as
[xxxxxxxxxxxxxxxxxxxxxxxxx]\n\tProperty twitter4j.oauth.accessToken
set as [xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]\n
\tProperty twitter4j.oauth.consumerSecret set as
[xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]\n\tProperty
twitter4j.oauth.accessTokenSecret set as
[xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]\n\n\nimport
org.apache.spark.streaming.twitter._\n\nssc:
org.apache.spark.streaming.StreamingContext =
org.apache.spark.streaming.StreamingContext@24773261\n\ntweets:
org.apache.spark.streaming.dstream.ReceiverInputDStream[twitter4j.Stat
us] = org.apache.spark.streaming.twitter.TwitterInputDStream@33b7833e
\n\ntwt: org.apache.spark.streaming.dstream.DStream[twitter4j.Status]
= org.apache.spark.streaming.dstream.WindowedDStream@e72e97f\n
\ndefined class Tweet\nIncomplete expression"},"text":"val apiKey =
\"xxxxxxxxxxxxxxxxxxxxxxxxx\"\nval apiSecret =
\"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\"\nval
accessToken = \"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
\"\nval accessTokenSecret =
\"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
\"\nconfigureTwitterCredentials(apiKey, apiSecret, accessToken,
accessTokenSecret)\n\nimport org.apache.spark.streaming.twitter._\nval
ssc = new StreamingContext(sc, Seconds(2))\nval tweets =
TwitterUtils.createStream(ssc, None)\nval twt =
tweets.window(Seconds(60))\n\ncase class Tweet(createdAt:Long,
text:String)\ntwt.map(status=>\n
Tweet(status.getCreatedAt().getTime()/1000, status.getText())
\n).foreachRDD(rdd=>"},{"config":{"colWidth":12,"graph":
{"mode":"table","height":300,"optionOpen":false,"keys":[],"values":
[],"groups":[],"scatter":{}},"enabled":true},"settings":{"params":
{},"forms":
{}},"jobName":"paragraph_1485474019719_484755215","id":"20170126-18401
9_260620869","dateCreated":"2017-01-26T18:40:19-0500","status":"ERROR"
,"progressUpdateIntervalMs":500,"focus":true,"$$hashKey":"object:
2318","dateUpdated":"2017-01-26T18:41:40-0500","dateFinished":"2017-01
-26T18:41:43-0500","dateStarted":"2017-01-26T18:41:40-0500","result":
{"code":"ERROR","type":"TEXT","msg":"\napiKey: String =
xxxxxxxxxxxxxxxxxxxxxxxxx\n\napiSecret: String =
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n\naccessToken:
String = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n
\naccessTokenSecret: String =
```

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\nConfiguring Twitter
OAuth\n\tProperty twitter4j.oauth.consumerKey set as
[xxxxxxxxxxxxxxxxxxxxxxxxx]\n\tProperty twitter4j.oauth.accessToken
set as [xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]\n
\tProperty twitter4j.oauth.consumerSecret set as
[xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]\n\tProperty
twitter4j.oauth.accessTokenSecret set as
[xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]\n\n\nimport
org.apache.spark.streaming.twitter._\n\nssc:
org.apache.spark.streaming.StreamingContext =
org.apache.spark.streaming.StreamingContext@6395e666\n\ntweets:
org.apache.spark.streaming.dstream.ReceiverInputDStream[twitter4j.Stat
us] = org.apache.spark.streaming.twitter.TwitterInputDStream@1bbdccea
\n\ntwt: org.apache.spark.streaming.dstream.DStream[twitter4j.Status]
= org.apache.spark.streaming.dstream.WindowedDStream@5c55b235\n
\ndefined class Tweet\n\n\n\n<console>:64: error: value
registerAsTable is not a member of org.apache.spark.sql.DataFrame\n
rdd.toDF().registerAsTable(\"tweets\")\n                      ^
\n"},"text":"val apiKey = \"xxxxxxxxxxxxxxxxxxxxxxxxx\"\nval apiSecret
= \"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\"\nval
accessToken = \"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
\"\nval accessTokenSecret =
\"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
\"\nconfigureTwitterCredentials(apiKey, apiSecret, accessToken,
accessTokenSecret)\n\nimport org.apache.spark.streaming.twitter._\nval
ssc = new StreamingContext(sc, Seconds(2))\nval tweets =
TwitterUtils.createStream(ssc, None)\nval twt =
tweets.window(Seconds(60))\n\ncase class Tweet(createdAt:Long,
text:String)\ntwt.map(status=>\n
Tweet(status.getCreatedAt().getTime()/1000, status.getText())
\n).foreachRDD(rdd=>\n  // Below line works only in spark 1.3.0.\n  //
For spark 1.1.x and spark 1.2.x,\n  // use
rdd.registerTempTable(\"tweets\") instead.\n
rdd.toDF().registerAsTable(\"tweets\")\n)\n\ntwt.print\n
\nssc.start()"},{"config":{"colWidth":12,"graph":
{"mode":"table","height":300,"optionOpen":false,"keys":[],"values":
[],"groups":[],"scatter":{}},"enabled":true},"settings":{"params":
{},"forms":
{}},"jobName":"paragraph_1485474089735_-749211129","id":"20170126-1841
29_1514311497","dateCreated":"2017-01-26T18:41:29-0500","status":"FINI
SHED","progressUpdateIntervalMs":500,"focus":true,"$$hashKey":"object:
2400","dateUpdated":"2017-01-26T18:42:41-0500","dateFinished":"2017-01
-26T18:42:43-0500","dateStarted":"2017-01-26T18:42:41-0500","result":
{"code":"SUCCESS","type":"TEXT","msg":"\napiKey: String =
xxxxxxxxxxxxxxxxxxxxxxxxx\n\napiSecret: String =
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n\naccessToken:
String = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\n
\naccessTokenSecret: String =
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\nConfiguring Twitter
OAuth\n\tProperty twitter4j.oauth.consumerKey set as

```
[xxxxxxxxxxxxxxxxxxxxxxxxx]\n\tProperty twitter4j.oauth.accessToken
set as [xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]\n
\tProperty twitter4j.oauth.consumerSecret set as
[xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]\n\tProperty
twitter4j.oauth.accessTokenSecret set as
[xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]\n\n"},"text":"val
apiKey = \"xxxxxxxxxxxxxxxxxxxxxxxxx\"\nval apiSecret =
\"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\"\nval
accessToken = \"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
\"\nval accessTokenSecret =
\"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
\"\nconfigureTwitterCredentials(apiKey, apiSecret, accessToken,
accessTokenSecret)"},{"config":{"colWidth":12,"graph":
{"mode":"table","height":300,"optionOpen":false,"keys":[],"values":
[],"groups":[],"scatter":{}},"enabled":true},"settings":{"params":
{},"forms":
{}},"jobName":"paragraph_1485474161604_-780501672","id":"20170126-1842
41_1907372591","dateCreated":"2017-01-26T18:42:41-0500","status":"ERRO
R","progressUpdateIntervalMs":500,"focus":true,"$$hashKey":"object:
2495","dateUpdated":"2017-01-26T18:54:57-0500","dateFinished":"2017-01
-26T18:54:59-0500","dateStarted":"2017-01-26T18:54:57-0500","result":
{"code":"ERROR","type":"TEXT","msg":"\nimport
org.apache.spark.streaming.twitter._\n\nssc:
org.apache.spark.streaming.StreamingContext =
org.apache.spark.streaming.StreamingContext@54132ef6\n\ntweets:
org.apache.spark.streaming.dstream.ReceiverInputDStream[twitter4j.Stat
us] =
org.apache.spark.streaming.twitter.TwitterInputDStream@21c90271\n
\ntwt: org.apache.spark.streaming.dstream.DStream[twitter4j.Status] =
org.apache.spark.streaming.dstream.WindowedDStream@1f79ac69\n\ndefined
class Tweet\n\n\n\n<console>:73: error: value registerAsTable is not a
member of org.apache.spark.sql.DataFrame\n
rdd.toDF().registerAsTable(\"tweets\")\n                    ^
\n"},"text":"import org.apache.spark.streaming.twitter._\nval ssc =
new StreamingContext(sc, Seconds(2))\nval tweets =
TwitterUtils.createStream(ssc, None)\nval twt =
tweets.window(Seconds(60))\n\ncase class Tweet(createdAt:Long,
text:String)\ntwt.map(status=>\n
Tweet(status.getCreatedAt().getTime()/1000, status.getText())
\n).foreachRDD(rdd=>\n  // Below line works only in spark 1.3.0.\n  //
For spark 1.1.x and spark 1.2.x,\n  // use
rdd.registerTempTable(\"tweets\") instead.\n
rdd.toDF().registerAsTable(\"tweets\")\n)\n\ntwt.print\n\nssc.start()
\n"},{"config":{"colWidth":12,"graph":{"mode":"table","height":
300,"optionOpen":false,"keys":[],"values":[],"groups":[],"scatter":
{}},"enabled":true},"settings":{"params":{},"forms":
{}},"jobName":"paragraph_1485474857772_-2052255486","id":"20170126-185
417_124550543","dateCreated":"2017-01-26T18:54:17-0500","status":"ERRO
R","progressUpdateIntervalMs":500,"focus":true,"$$hashKey":"object:
2658","dateUpdated":"2017-01-26T18:58:48-0500","dateFinished":"2017-01
```

```
-26T18:58:50-0500","dateStarted":"2017-01-26T18:58:48-0500","result":
{"code":"ERROR","type":"TEXT","msg":"\nimport
org.apache.spark.streaming.twitter._\n\nssc:
org.apache.spark.streaming.StreamingContext =
org.apache.spark.streaming.StreamingContext@121a8e4\n\ntweets:
org.apache.spark.streaming.dstream.ReceiverInputDStream[twitter4j.Stat
us] =
org.apache.spark.streaming.twitter.TwitterInputDStream@54bd2156\n
\ntwt: org.apache.spark.streaming.dstream.DStream[twitter4j.Status] =
org.apache.spark.streaming.dstream.WindowedDStream@67792e92\n\ndefined
class Tweet\n\n\n\n<console>:4: error: ')' expected but double literal
found.\n   Below line works only in spark 1.3.0.\n
^\n"},"text":"import org.apache.spark.streaming.twitter._\nval ssc =
new StreamingContext(sc, Seconds(2))\nval tweets =
TwitterUtils.createStream(ssc, None)\nval twt =
tweets.window(Seconds(60))\n\ncase class Tweet(createdAt:Long,
text:String)\ntwt.map(status=>\n
Tweet(status.getCreatedAt().getTime()/1000, status.getText())
\n).foreachRDD(rdd=>\n   Below line works only in spark 1.3.0.\n   For
spark 1.1.x and spark 1.2.x,\n   use rdd.registerTempTable(\"tweets\")
instead.\n  rdd.toDF().registerTempTable(\"tweets\"))\n\ntwt.print\n
\nssc.start()"},{"config":{"colWidth":12,"graph":
{"mode":"table","height":300,"optionOpen":false,"keys":[],"values":
[],"groups":[],"scatter":{}},"enabled":true},"settings":{"params":
{},"forms":
{}},"jobName":"paragraph_1485474985378_1567613763","id":"20170126-1856
25_373519891","dateCreated":"2017-01-26T18:56:25-0500","status":"FINIS
HED","progressUpdateIntervalMs":500,"focus":true,"$$hashKey":"object:
2753","dateUpdated":"2017-01-26T19:00:06-0500","dateFinished":"2017-01
-26T19:00:11-0500","dateStarted":"2017-01-26T19:00:06-0500","result":
{"code":"SUCCESS","type":"TEXT","msg":"\nimport
org.apache.spark.streaming.twitter._\n\nssc:
org.apache.spark.streaming.StreamingContext =
org.apache.spark.streaming.StreamingContext@6f9bb576\n\ntweets:
org.apache.spark.streaming.dstream.ReceiverInputDStream[twitter4j.Stat
us] = org.apache.spark.streaming.twitter.TwitterInputDStream@47c450f\n
\ntwt: org.apache.spark.streaming.dstream.DStream[twitter4j.Status] =
org.apache.spark.streaming.dstream.WindowedDStream@69e0b341\n\ndefined
class Tweet\n\nwarning: there was one deprecation warning; re-run with
-deprecation for details\n"},"text":"import
org.apache.spark.streaming.twitter._\nval ssc = new
StreamingContext(sc, Seconds(2))\nval tweets =
TwitterUtils.createStream(ssc, None)\nval twt =
tweets.window(Seconds(60))\n\ncase class Tweet(createdAt:Long,
text:String)\ntwt.map(status=>\n
Tweet(status.getCreatedAt().getTime()/1000, status.getText())
\n).foreachRDD(rdd=>\n \n  rdd.toDF().registerTempTable(\"tweets\")\n)
\n\ntwt.print\n\nssc.start()"},{"config":{"colWidth":12,"graph":
{"mode":"table","height":300,"optionOpen":false,"keys":[],"values":
[],"groups":[],"scatter":{}},"enabled":true},"settings":{"params":
```

```
{},"forms":
{}},"jobName":"paragraph_1485475206431_-854290262","id":"20170126-1900
06_2017286573","dateCreated":"2017-01-26T19:00:06-0500","status":"FINI
SHED","progressUpdateIntervalMs":500,"focus":true,"$$hashKey":"object:
2882","dateUpdated":"2017-01-26T19:04:54-0500","dateFinished":"2017-01
-26T19:04:55-0500","dateStarted":"2017-01-26T19:04:54-0500","result":
{"code":"SUCCESS","type":"TABLE","msg":"createdAt\ttext
\n","comment":"","msgTable":[],"columnNames":
[{"name":"createdAt","index":0,"aggr":"sum"},{"name":"text","index":
1,"aggr":"sum"}],"rows":[]},"text":"%sql select * from tweets where
text like '%girl%' limit 10"},{"config":{"colWidth":12,"graph":
{"mode":"table","height":300,"optionOpen":false,"keys":[],"values":
[],"groups":[],"scatter":{}},"enabled":true},"settings":{"params":
{},"forms":
{}},"jobName":"paragraph_1485475494438_261311338","id":"20170126-19045
4_105792299","dateCreated":"2017-01-26T19:04:54-0500","status":"FINISH
ED","progressUpdateIntervalMs":500,"focus":true,"$$hashKey":"object:
2970","dateUpdated":"2017-01-26T19:05:16-0500","dateFinished":"2017-01
-26T19:05:19-0500","dateStarted":"2017-01-26T19:05:16-0500","result":
{"code":"SUCCESS","type":"TABLE","msg":"createdAt
\tcount(1)\n","comment":"","msgTable":[],"columnNames":
[{"name":"createdAt","index":0,"aggr":"sum"},
{"name":"count(1)","index":1,"aggr":"sum"}],"rows":[]},"text":"%sql
select createdAt, count(1) from tweets group by createdAt order by
createdAt"},{"config":{"colWidth":12,"graph":{"mode":"table","height":
300,"optionOpen":false,"keys":[],"values":[],"groups":[],"scatter":
{}},"enabled":true},"settings":{"params":{},"forms":
{}},"jobName":"paragraph_1485475516553_-1136450768","id":"20170126-190
516_273038672","dateCreated":"2017-01-26T19:05:16-0500","status":"FINI
SHED","progressUpdateIntervalMs":500,"focus":true,"$$hashKey":"object:
3061","dateUpdated":"2017-01-26T19:06:54-0500","dateFinished":"2017-01
-26T19:06:56-0500","dateStarted":"2017-01-26T19:06:54-0500","result":
{"code":"SUCCESS","type":"TEXT","msg":"\nsentiment: (s: String)String
\n\nres16: org.apache.spark.sql.expressions.UserDefinedFunction =
UserDefinedFunction(<function1>,StringType,Some(List(StringType)))
\n"},"text":"def sentiment(s:String) : String = {\n    val positive =
Array(\"like\", \"love\", \"good\", \"great\", \"happy\", \"cool\",
\"the\", \"one\", \"that\")\n    val negative = Array(\"hate\", \"bad
\", \"stupid\", \"is\")\n\n    var st = 0;\n\n    val words =
s.split(\" \")    \n    positive.foreach(p =>\n        words.foreach(w
=>\n            if(p==w) st = st+1\n        )\n    )\n\n
negative.foreach(p=>\n        words.foreach(w=>\n            if(p==w)
st = st-1\n        )\n    )\n    if(st>0)\n        \"positivie\"\n
else if(st<0)\n        \"negative\"\n    else\n        \"neutral\"\n}
\nsqlc.udf.register(\"sentiment\", sentiment _)"},{"config":
{"colWidth":12,"graph":{"mode":"stackedAreaChart","height":
300,"optionOpen":false,"keys":[{"name":"UDF(text)","index":
0,"aggr":"sum"}],"values":[{"name":"count(1)","index":
1,"aggr":"sum"}],"groups":[],"scatter":{"xAxis":
{"name":"UDF(text)","index":0,"aggr":"sum"},"yAxis":
```

```
{"name":"count(1)","index":
1,"aggr":"sum"}}},"enabled":true},"settings":{"params":{},"forms":
{}},"jobName":"paragraph_1485475614345_392326823","id":"20170126-19065
4_226249904","dateCreated":"2017-01-26T19:06:54-0500","status":"FINISH
ED","progressUpdateIntervalMs":500,"focus":true,"$$hashKey":"object:
3169","dateUpdated":"2017-01-26T19:08:02-0500","dateFinished":"2017-01
-26T19:07:31-0500","dateStarted":"2017-01-26T19:07:30-0500","result":
{"code":"SUCCESS","type":"TABLE","msg":"UDF(text)
\tcount(1)\n","comment":"","msgTable":[],"columnNames":
[{"name":"UDF(text)","index":0,"aggr":"sum"},
{"name":"count(1)","index":1,"aggr":"sum"}],"rows":[]},"text":"%sql
select sentiment(text), count(1) from tweets where text like '%girl%'
group by sentiment(text)"},{"config":{"colWidth":12,"graph":
{"mode":"table","height":300,"optionOpen":false,"keys":[],"values":
[],"groups":[],"scatter":{}},"enabled":true},"settings":{"params":
{},"forms":
{}},"jobName":"paragraph_1485475643088_1087598733","id":"20170126-1907
23_247183887","dateCreated":"2017-01-26T19:07:23-0500","status":"READY
","progressUpdateIntervalMs":500,"focus":true,"$$hashKey":"object:
3248","dateUpdated":"2017-01-26T19:07:33-0500","text":""}],"name":"str
eaming","id":"2CAEZAX8P","angularObjects":{"2C6YZDZJ1:shared_process":
[],"2C9TGJH1M:shared_process":[],"2C5YQDCFT:shared_process":
[],"2C7JAUPCG:shared_process":[],"2C7Z11XQ5:shared_process":
[],"2C8YPPR9H:shared_process":[],"2C697T4A7:shared_process":
[],"2C9UM6M8B:shared_process":[],"2C7D9483F:2CAEZAX8P":
[],"2C8E9DEQ3:shared_process":[],"2C5YUFRQ3:shared_process":
[],"2C7ZSJBKN:shared_process":[],"2C8E11KXS:shared_process":
[],"2C8WKZ1QA:shared_process":[],"2C95F29PA:shared_process":
[],"2C7WJA2Z6:shared_process":[],"2C8ZRKM3C:shared_process":
[],"2C6WCZ7CA:shared_process":[]},"config":
{"looknfeel":"default"},"info":{}}
```