Apache Zeppelin使用入门指南：编程

使用Apache Zeppelin

编译和启动完Zeppelin相关的进程之后，我们就可以来使用Zeppelin了。我们进入到https://www.iteblog.com:8080页面，我们可以在页面上直接操作Zeppelin，依次选择Notebook->Create new note，然后会弹出一个对话框，我们在Note Name里面随便输入一个名字，这里我输入iteblog，然后点击Create Note就可以创建一个新的Notebook了。我们可以在新建的Notebook里面输入相关的代码进行测试：

```
sc.version
sqlc

res26: String = 1.5.2
res27: org.apache.spark.sql.SQLContext = org.apache.spark.sql.SQLContext@48806d6c
```

和Spark Shell一样，Zeppelin会初始化好SparkContext和SQLContext对象，分别命名为sc和sqlc，我们可以直接在里面使用到它。接下来我们来在Zeppelin里面加载HDFS上面的数据，如下：

```
sc.textFile("hdfs://www.iteblog.com/tmp/json").count

res29: Long = 200
Took 0 seconds (outdated)
```

我们再来使用sqlc对象读取上面的json文件来创建一个DataFrame：

```
val profilesJsonRdd =sqlc.jsonFile("hdfs://www.iteblog.com/tmp/json")
val profileDF=profilesJsonRdd.toDF()
val iteblog = profileDF.selectExpr("_id", "address", "age", "email")
iteblog.show()
profileDF.registerTempTable("profiles")

profilesJsonRdd: org.apache.spark.sql.DataFrame = [_id: string, about: string, address: string, a
ge: bigint, company: string, email: string, eyeColor: string, favoriteFruit: string, gender: string,
name: string, phone: string, registered: string, tags: array&lt;string&gt;]
profileDF: org.apache.spark.sql.DataFrame = [_id: string, about: string, address: string, age: bigi
nt, company: string, email: string, eyeColor: string, favoriteFruit: string, gender: string, name: s
tring, phone: string, registered: string, tags: array&lt;string&gt;]
root
 |-- _id: string (nullable = true)
 |-- about: string (nullable = true)
 |-- address: string (nullable = true)
 |-- age: long (nullable = true)
 |-- company: string (nullable = true)
 |-- email: string (nullable = true)
 |-- eyeColor: string (nullable = true)
 |-- favoriteFruit: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- name: string (nullable = true)
 |-- phone: string (nullable = true)
 |-- registered: string (nullable = true)
 |-- tags: array (nullable = true)
 |    |-- element: string (containsNull = true)
```

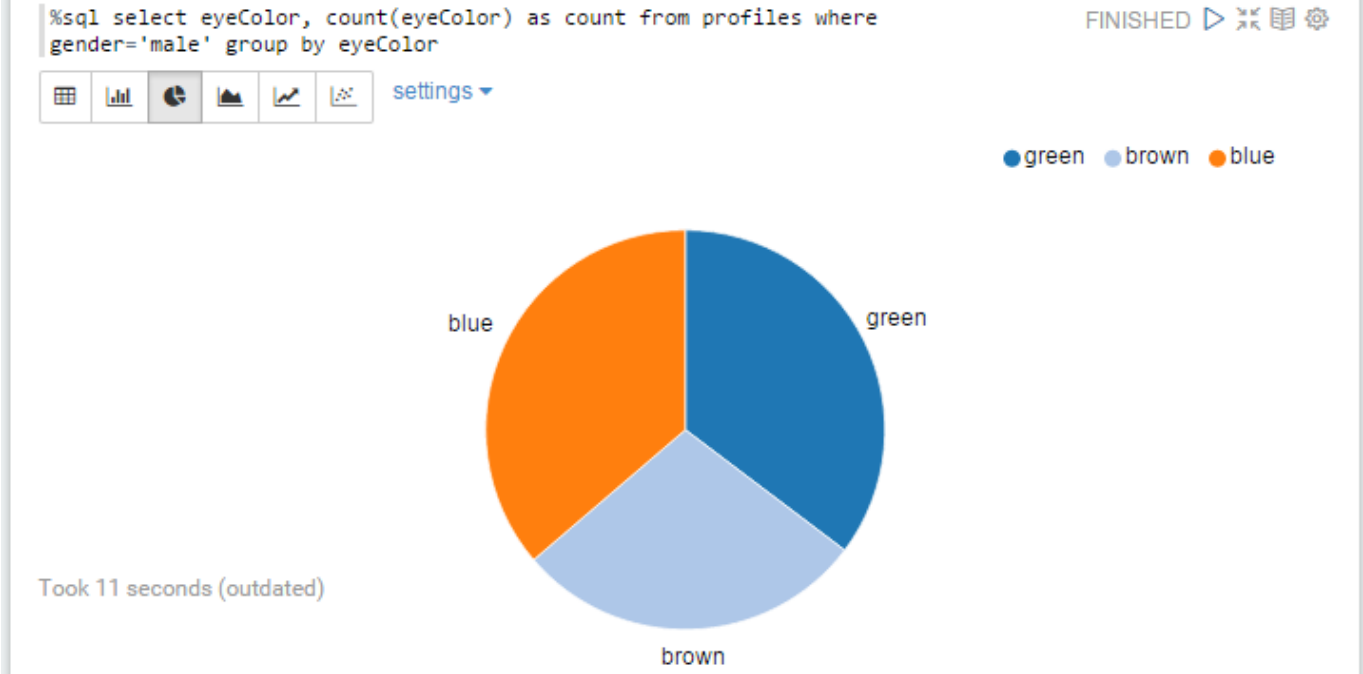
```
iteblog: org.apache.spark.sql.DataFrame = [_id: string, address: string, age: bigint, email: string
]
+--------------------+--------------------+---+--------------------+
|                 _id|             address|age|               email|
+--------------------+--------------------+---+--------------------+
|55578ccb0cc5b350d...|694 Oriental Cour...| 30|tracynguyen@endip...|
|55578ccb6975c4e2a...|267 Amber Street,...| 23|leannagarrett@war...|
|55578ccb33399a615...|243 Bridgewater S...| 24|blairwhite@imperi...|
|55578ccb0f1d5ab09...|647 Loring Avenue...| 24|andrearay@beadzza...|
|55578ccb591a45d4e...|721 Bijou Avenue,...| 27|penningtongilbert...|
|55578ccb9f0cd20c4...|694 Llama Court, ...| 21|shelleyburns@pyra...|
|55578ccb8d0accc28...|498 Perry Terrace...| 40|nicolefigueroa@ed...|
|55578ccbd682cca21...|243 Stillwell Ave...| 32|galealvarado@sing...|
|55578ccb0d9025ddd...|649 Beard Street,...| 36|melindaparker@fur...|
|55578ccb5be70de0d...|972 Marconi Place...| 36|byerscarson@digia...|
|55578ccbc5a1050a5...|483 Hanson Place,...| 31|kristiemckinney@a...|
|55578ccb07fa02369...|540 Woodpoint Roa...| 40|salazarburks@micr...|
|55578ccb809e55bf0...|442 Ainslie Stree...| 32|hopkinspatterson@...|
|55578ccb204ff8ee6...|444 Argyle Road, ...| 23|maysrosario@imkan...|
|55578ccb4b062fc61...|571 Sunnyside Ave...| 38|atkinshancock@hel...|
|55578ccba5ff361a9...|385 Meeker Avenue...| 40|edwinarobertson@s...|
|55578ccb386940ac3...|936 Cheever Place...| 37|elsienoel@fleetmi...|
|55578ccbfc41ff7fe...|406 Lake Place, M...| 36|mirandamarsh@even...|
|55578ccbfa6b6c300...|364 Metropolitan ...| 31|sharronmcconnell@...|
|55578ccbdd6650d81...|113 Applegate Cou...| 29|mcdowellwelch@eur...|
+--------------------+--------------------+---+--------------------+
only showing top 20 rows

Took 1 seconds
```

需要注意的是，因为Zeppelin会自动地初始化好SparkContext和SQLContext，所以我们不能再显示地创建SparkContext和SQLContext。如果你显示地创建了SQLContext，而且使用它来注册一张临时表，当你下一次使用这个临时表的时候将会出现以下的异常信息：

```
no such table List ([iteblog])
```

下面我们来使用上面注册的临时表，Zeppelin自带了SQL Interpreter，所以我们可以直接在上面编写SQL语句：

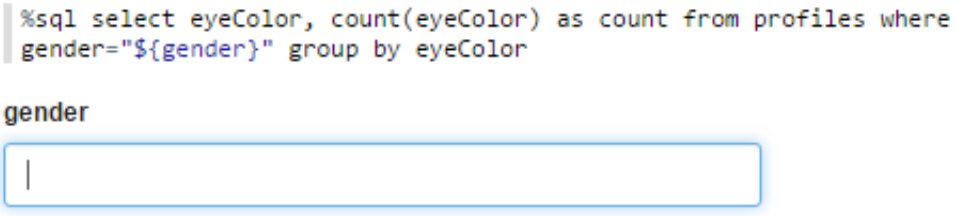
%sql select eyeColor, count(eyeColor) as count from profiles where gender='male' group by eyeColor



运行上面的SQL我们就可以得到图形化显示的结果，而且我们可以根据自己的需要选择饼型、条型、表格，线型等方式展现我们需要的结果！上面的SQL已经我们已经将查询的gender写死成male了，其实我们可以将这个值设置成参数的形式，然后我们可以在页面上输入相关的查询参数：

%sql select eyeColor, count(eyeColor) as count from profiles where gender='male' group by eyeColor

然后我们运行这个sql，我们可以看到下图的运行结果：

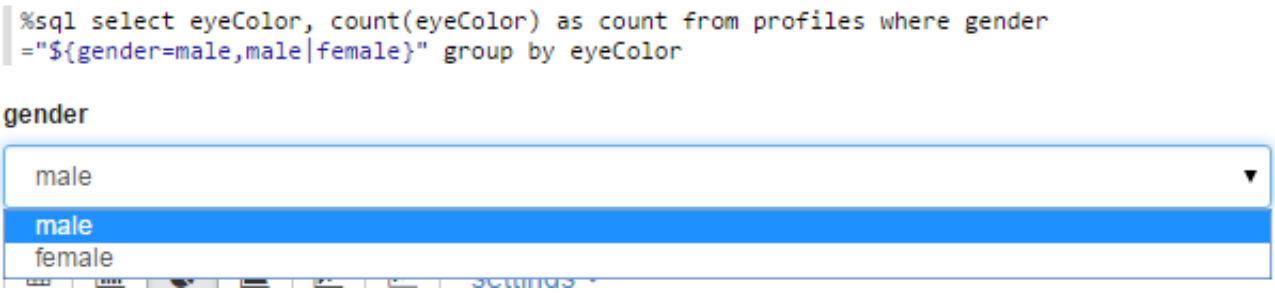


可以看出这里出现了一个文本框gender，我们可以输入需要查询的条件比如：male，然后再运行

就可以得到上面sql一样的结果。大家可能看到了，文本框里面是没有输入限制的，我们可以随便输入数据，而且你也不清楚到底有几种值可以输入。值得高兴的是，我们可以将这个值设置成只固定可选的：

%sql select eyeColor, count(eyeColor) as count from profiles where gender ="${gender=male,male|female}" group by eyeColor

这里限制了gender的值只能选择male和female，而且默认male是选中的，如下：



有时候我们需要在SQL中使用自定义的函数，我们可以直接在Zeppelin中定义函数，然后在SQL使用它，如下：

```
def ageGroup(age: Long) = {
   val buckets = Array("0-10", "11-20", "20-30", "31-40",
    "41-50", "51-60", "61-70", "71-80", "81-90", "91-100", ">100")
   buckets(math.min((age.toInt - 1) / 10, buckets.length - 1))
}

ageGroup: (age: Long)String
```

为了能够在Spark SQL中使用这个函数，我们必须先注册这个函数：

```
sqlc.udf.register("ageGroup", (age:Long)=>ageGroup(age.toInt))

res44: org.apache.spark.sql.UserDefinedFunction = UserDefinedFunction(<function1>,StringType,List())
```

然后我们就可以在Spark SQL中使用这个自定义函数：

```
%sql select ageGroup(age) as iteblogAgeGroup,
count(1) as total
from profiles
where gender='${gender=male,male|female}' group by ageGroup(age)
order by iteblogAgeGroup
```

运行的结果如下：