OPEN
DATA
SCIENCE
CONFERENCE

SAN FRANCISCO | 2015

#ODSC

@OPENDATASCI

Apache Drill & Zeppelin: Two Promising Tools You've Never Heard Of

Charles Givre

@cgivre, givre_charles@bah.com

Who am I?

Charles Givre @cgivre
givre_charles@bah.com

Booz | Allen | Hamilton
100 YEARS

Apache Zeppelin

#ODSC

@OPENDATASCI

#ODSC

@OPENDATASCI

Drill is the first schema free SQL Engine

@OPENDATASCI

Drill allows you to query self-describing data wherever it is, using standard SQL.

Drill allows you to query **self-describing data** wherever it is, using standard SQL.

Drill allows you to query **self-describing data** **wherever it is**, using **standard SQL**.

Open Source

Flexible and Extensible

Scale

+

=

#ODSC

@OPENDATASCI

#ODSC

@OPENDATASCI


```
SELECT <fields>
FROM dfs.strata.`file1.csv.` f1
JOIN dfs.strata.`file2.json` f2 ON
f1.id = f2.id
WHERE <something is true>
GROUP BY f1.field1
ORDER BY f2.name DESC
```

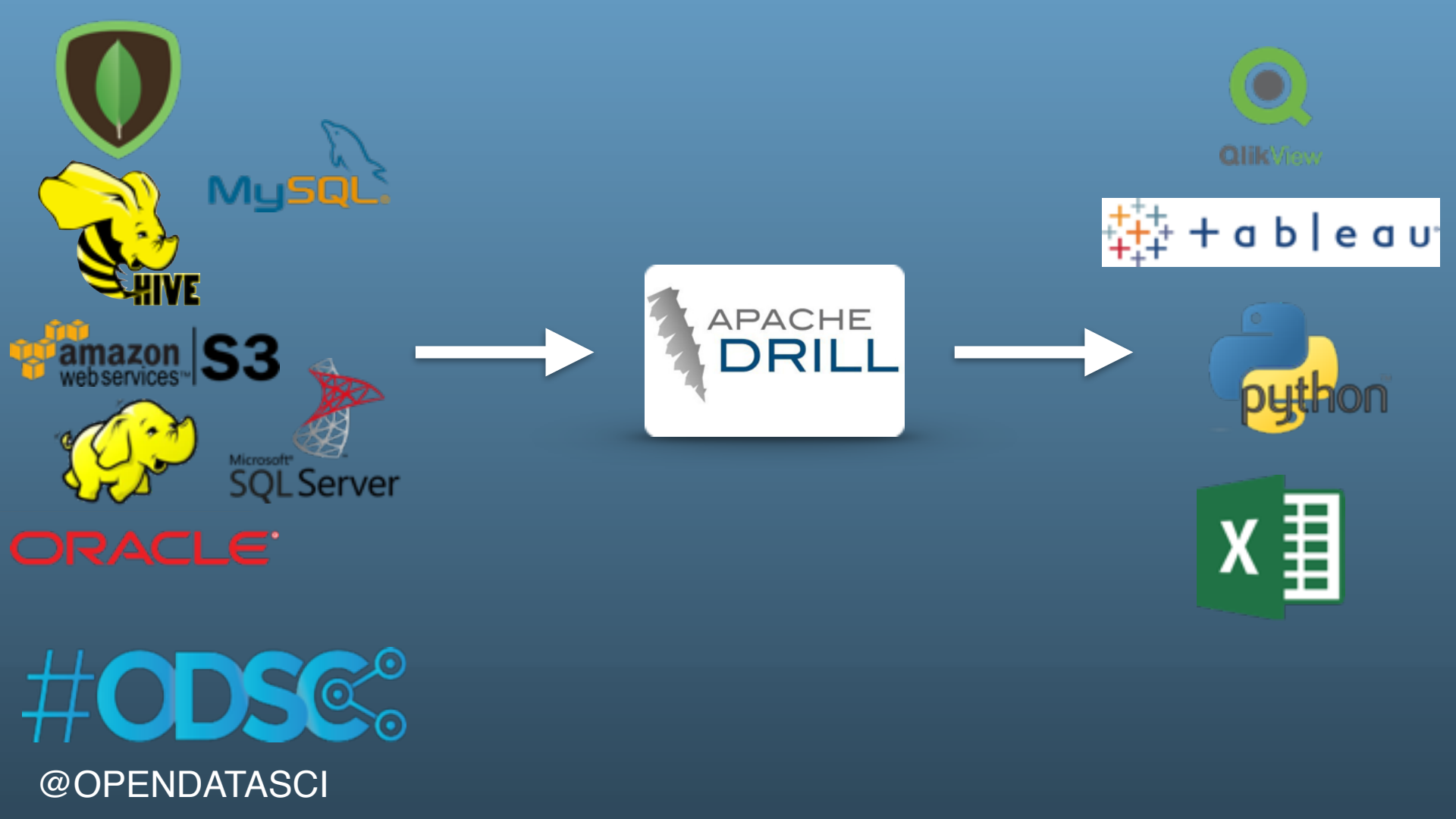
#ODSC

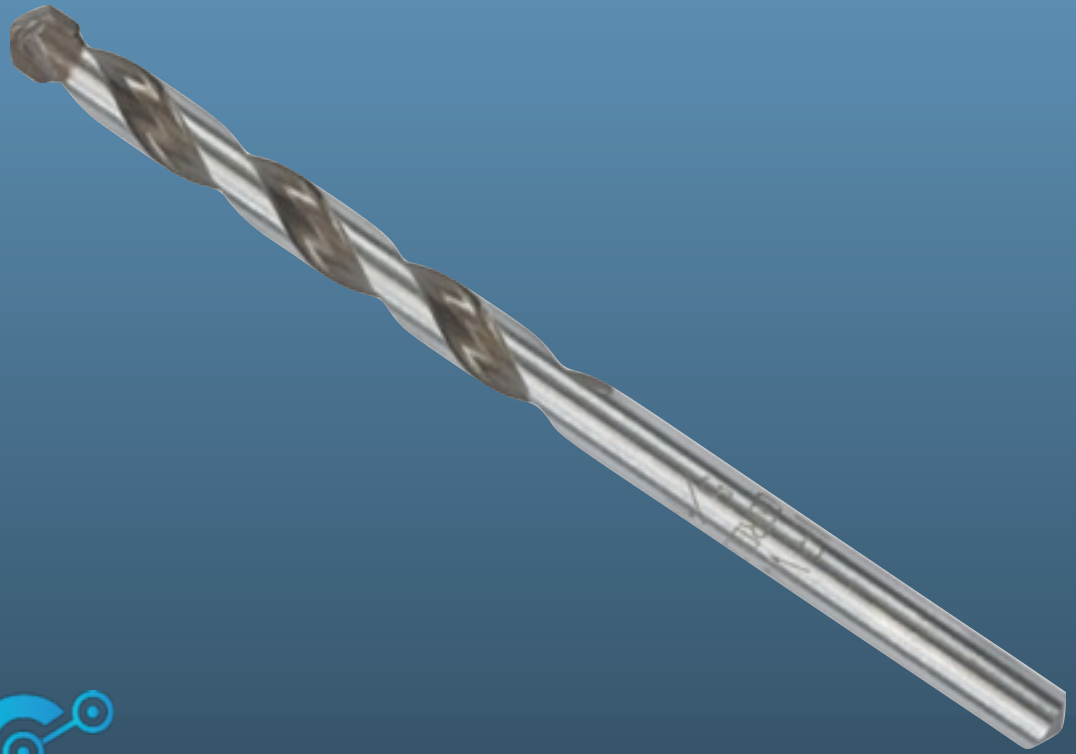
@OPENDATASCI

ETL

#ODSC

@OPENDATASCI

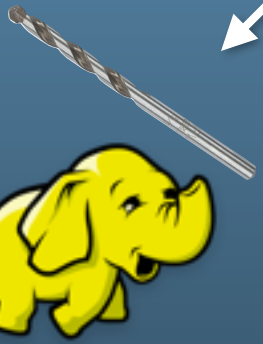

MySQL
HIVE
amazon web services
S3
Microsoft SQL Server
ORACLE
APACHE DRILL
QlikView
tableau
python
X
#ODSC
@OPENDATASCI



#ODSC

@OPENDATASCI

APACHE DRILL



#ODSC

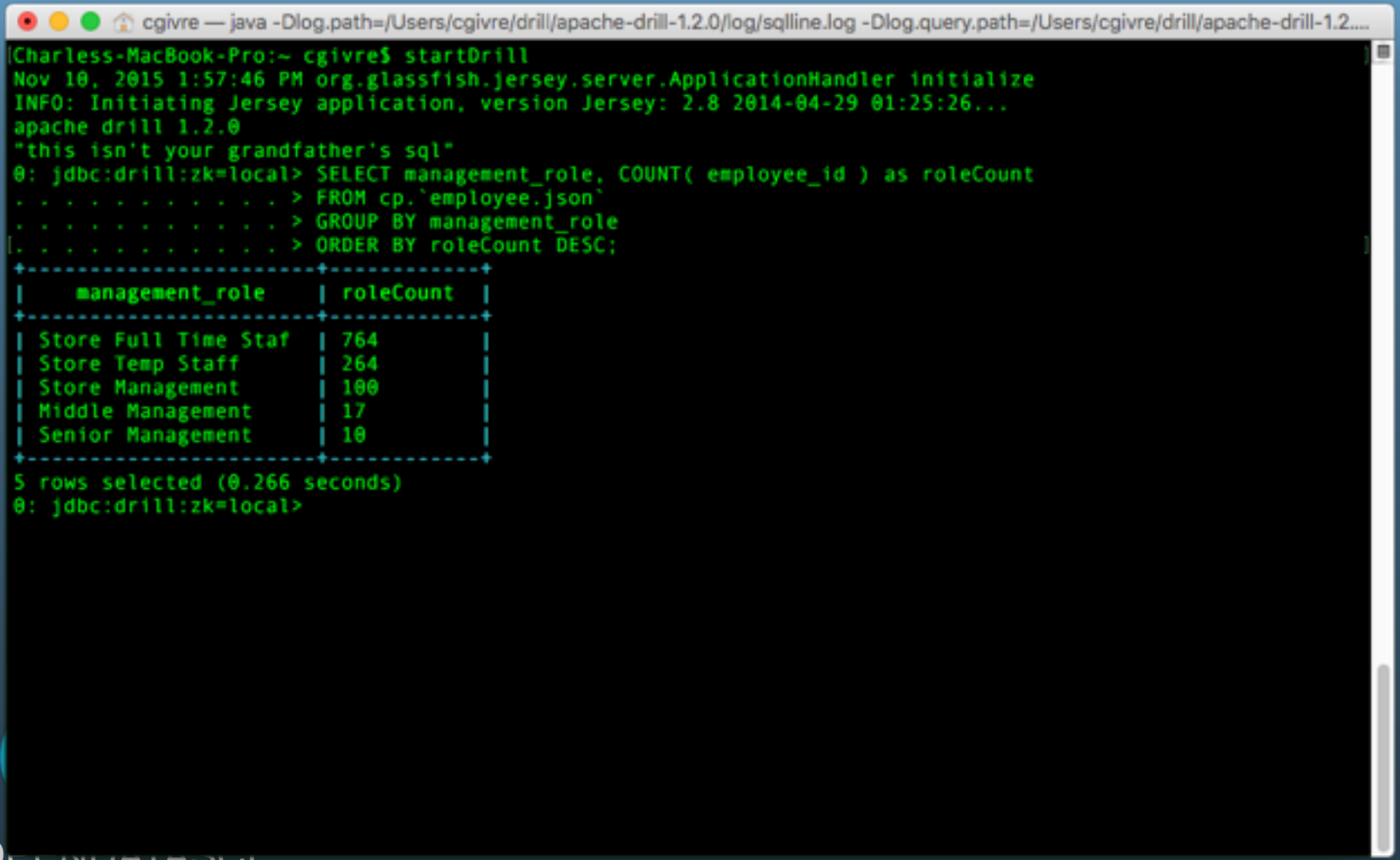
@OPENDATASCI

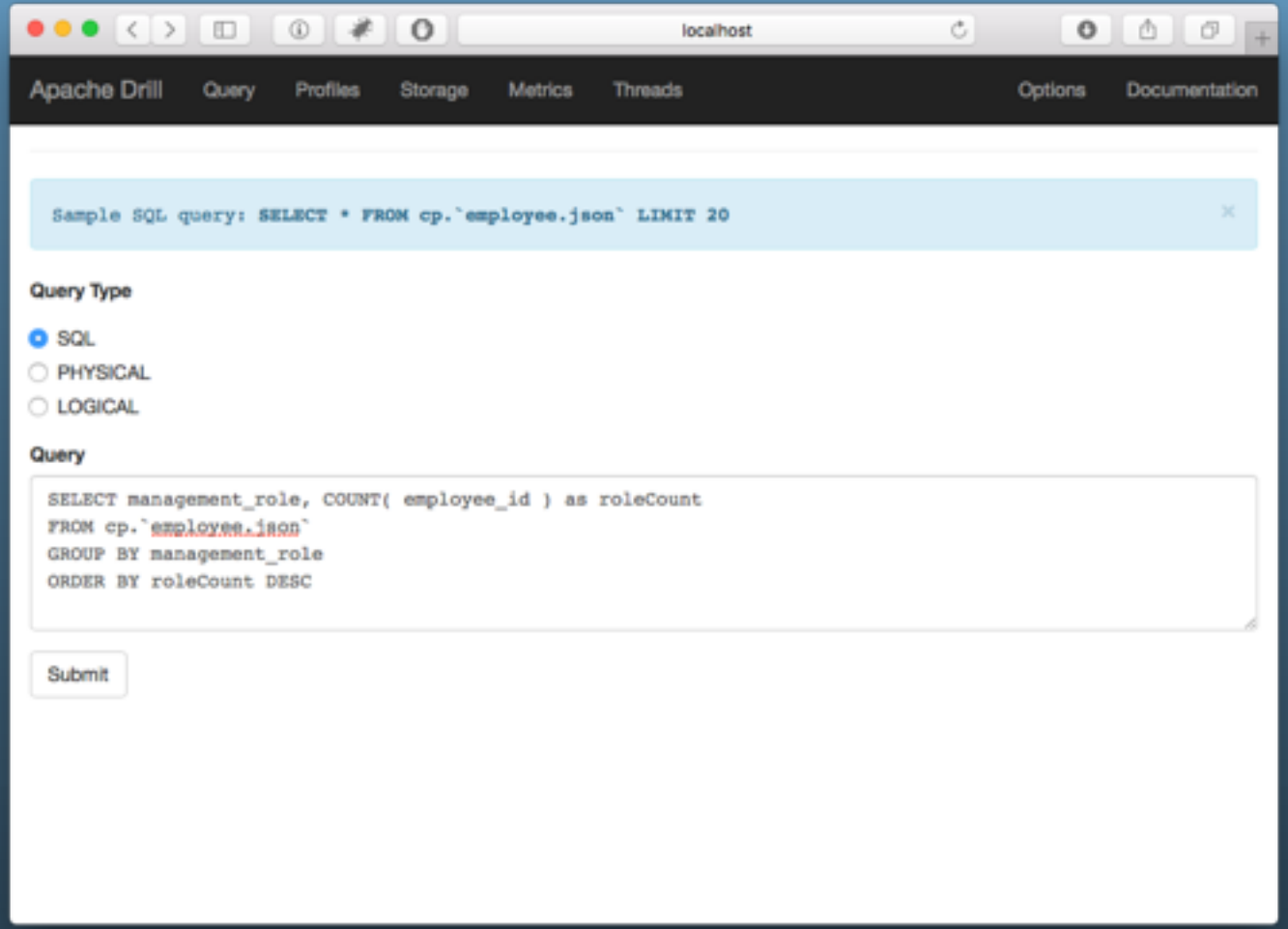
```
{    “employee_id":1,
    "full_name":"Sheri    Nowmer”,
    “first_name":"Sheri",
    “last_name”:”Nowmer",
     …
    "management_role":"Senior Management”
}
```

#ODSC

@OPENDATASCI

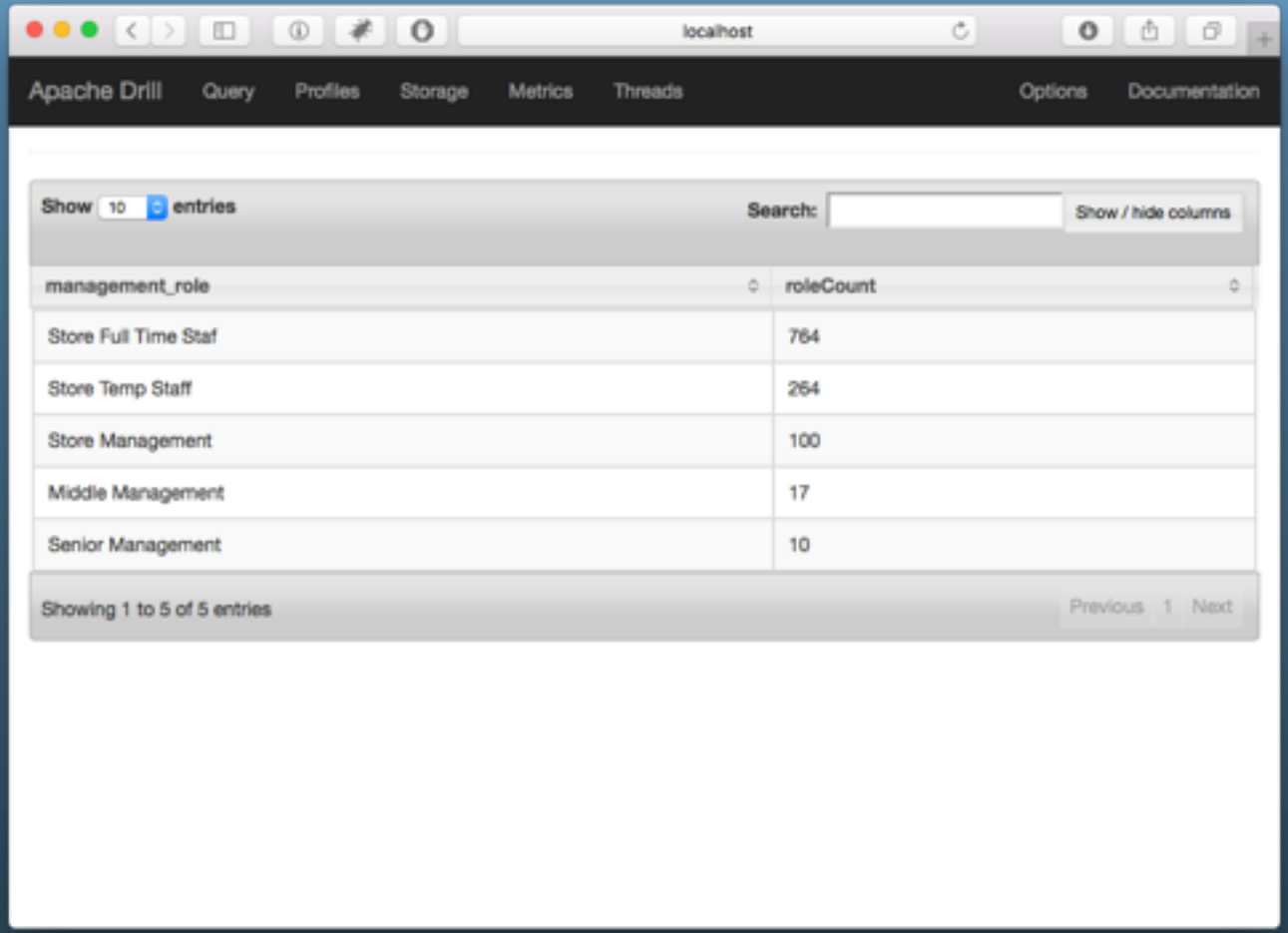
```
SELECT management_role, COUNT( employee_id ) as roleCount
FROM cp.`employee.json`
GROUP BY management_role
ORDER BY roleCount DESC
```


cgivre — java -Dlog.path=/Users/cgivre/drill/apache-drill-1.2.0/log/sqlline.log -Dlog.query.path=/Users/cgivre/drill/apache-drill-1.2....
Charless-MacBook-Pro:~ cgivre$ startDrill
Nov 10, 2015 1:57:46 PM org.glassfish.jersey.server.ApplicationHandler initialize
INFO: Initiating Jersey application, version Jersey: 2.8 2014-04-29 01:25:26...
apache drill 1.2.0
"this isn't your grandfather's sql"
0: jdbc:drill:zk=local> SELECT management_role, COUNT(employee_id) as roleCount
. . . . . > FROM cp.`employee.json`
. . . . . > GROUP BY management_role
. . . . . > ORDER BY roleCount DESC;
management_role	roleCount
Store Full Time Staf	764
Store Temp Staff	264
Store Management	100
Middle Management	17
Senior Management	10
5 rows selected (0.266 seconds)
0: jdbc:drill:zk=local>



#ODSC

@OPENDATASCI



#ODSC

@OPENDATASCI

```
import requests
import pandas as pd
import json

url = "http://localhost:8047/query.json"
employee_query = """SELECT management_role, COUNT( employee_id ) as roleCount
FROM cp.`employee.json`
GROUP BY management_role
ORDER BY roleCount DESC"""

data = {"queryType" : "SQL", "query": employee_query }
data_json = json.dumps(data)
headers = {'Content-type': ‘application/json'}

response = requests.post(url, data=data_json, headers=headers)
df = pd.DataFrame( response.json()['rows'] )
print( df.head() )
```

@OPENDATASCI

	management_role	roleCount
0	Store Full Time Staf	764
1	Store Temp Staff	264
2	Store Management	100
3	Middle Management	17
4	Senior Management	10

http://bit.ly/1MJZ7QX : Accessing Drill via REST
http://bit.ly/1GWbNIq: Accessing Drill via ODBC

Performance

#ODSC

@OPENDATASCI

	Drill	SQL-on-Hadoop (Hive, Impala, etc.)
Use case	Self-service, in-situ, SQL-based analytics	Data warehouse offload
Data sources	Hadoop, NoSQL, cloud storage (including multiple instances)	A single Hadoop cluster
Data model	Schema-free JSON (like MongoDB)	Relational
User experience	Point-and-query	Ingest data → define schemas → query
Deployment model	Standalone service or co-located with Hadoop or NoSQL	Co-located with Hadoop
Data management	Self-service	IT-driven
SQL	ANSI SQL	SQL-like
1.0 availability	Q2 2015	Q2 2013 or earlier

https://drill.apache.org/faq/

https://drill.apache.org

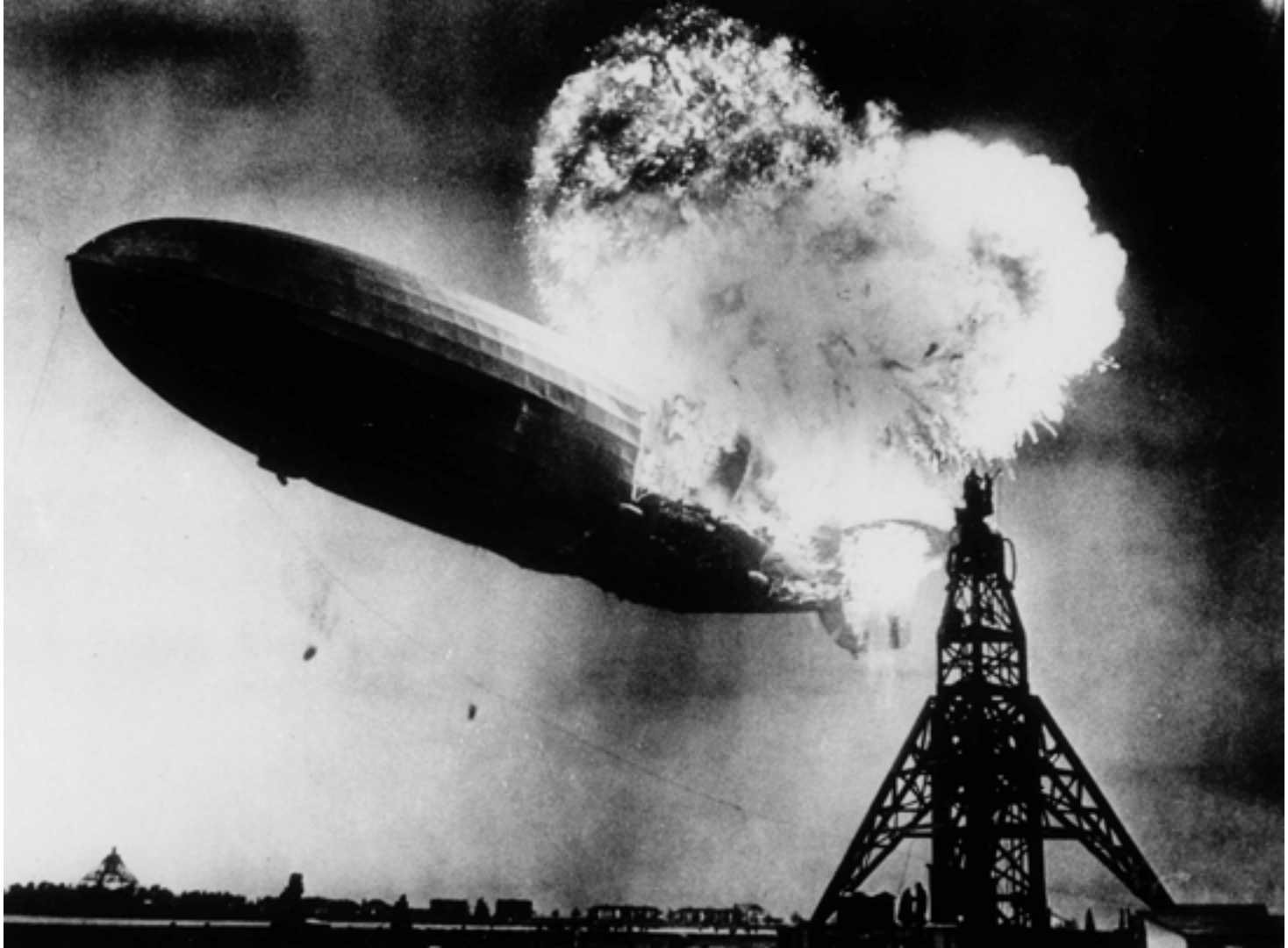
@OPENDATASCI

Apache Zeppelin

#ODSC

@OPENDATASCI

What is it?



A web based notebook that enables interactive data analytics

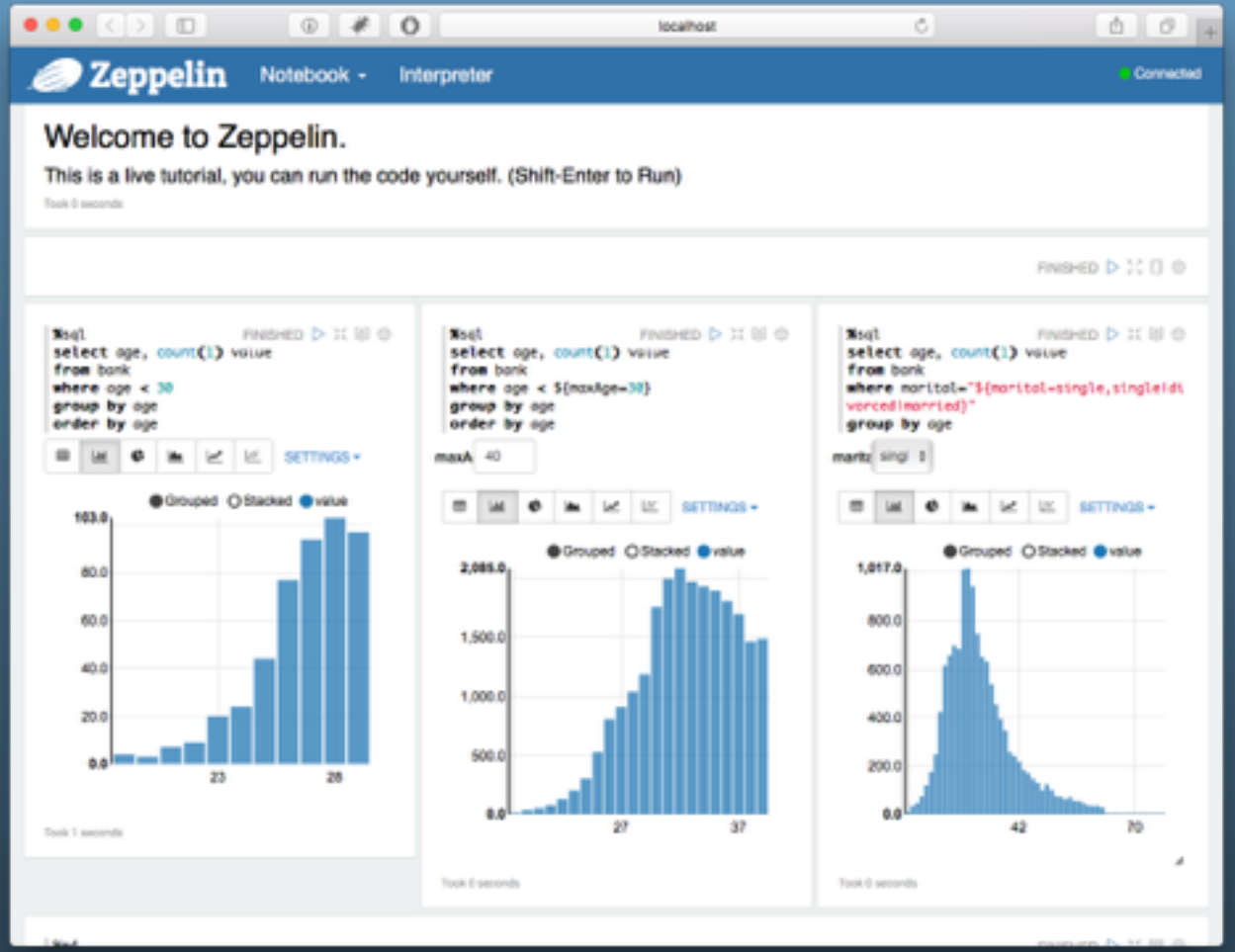
@OPENDATASCI

Built in integration with 

@OPENDATASCI

Zeppelin makes exploring data easy

@OPENDATASCI



#ODSC

@OPENDATASCI

```
val bankText = sc.textFile("/Users/cgivre/Downloads/bank/bank-full.csv")

case class Bank(age:Integer, job:String, marital : String, education : String, balance :
Integer)

val bank = bankText.map(s=>s.split(";")).filter(s=>s(0)!="\"age\"").map(
    s=>Bank(s(0).toInt,
            s(1).replaceAll("\"", ""),
            s(2).replaceAll("\"", ""),
            s(3).replaceAll("\"", ""),
            s(5).replaceAll("\"", "").toInt
        )
)

bank.toDF().registerTempTable("bank")
```

@OPENDATASCI

```
%sql
SELECT marital, COUNT( 1 ) AS statusCount
FROM bank
GROUP BY marital
ORDER BY statusCount DESC
```

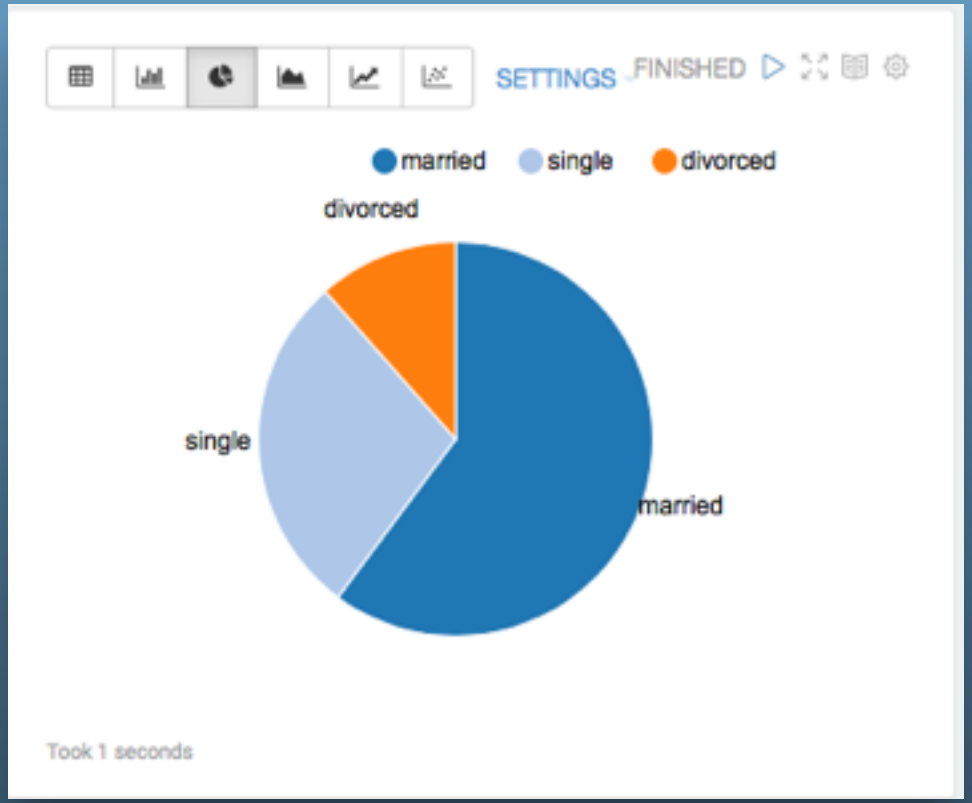
```
%sql
SELECT marital, COUNT( 1 ) AS statusCount
FROM bank
GROUP BY marital
ORDER BY statusCount DESC
```

FINISHED

marital	statusCount
married	27,214
single	12,790
divorced	5,207

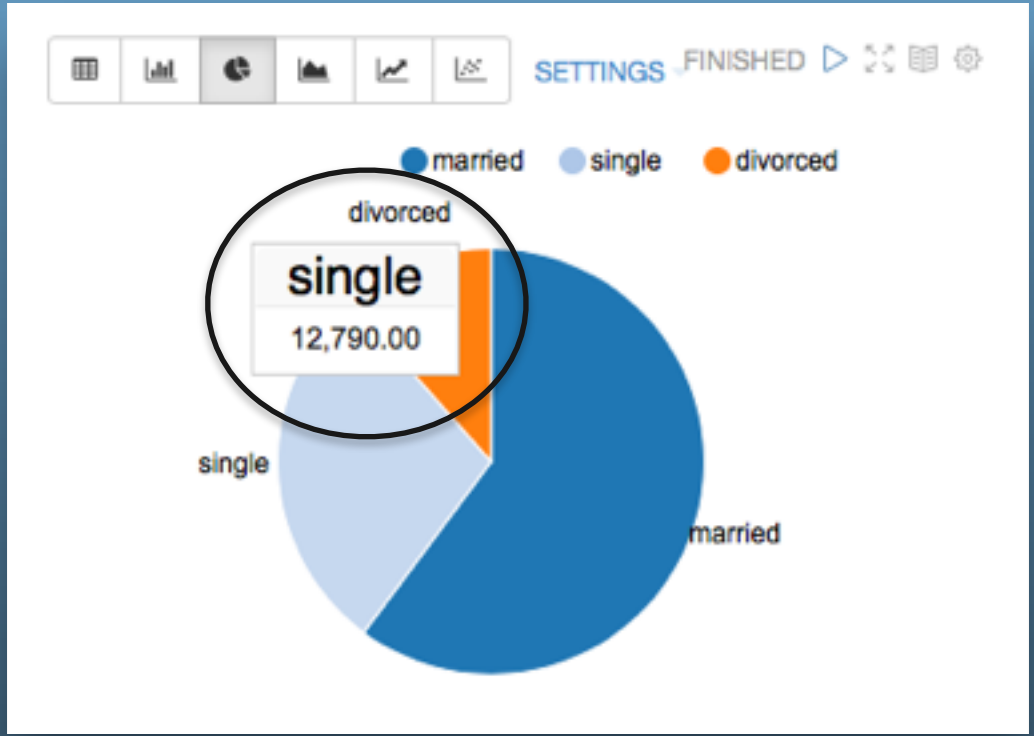
#ODSC

@OPENDATASCI



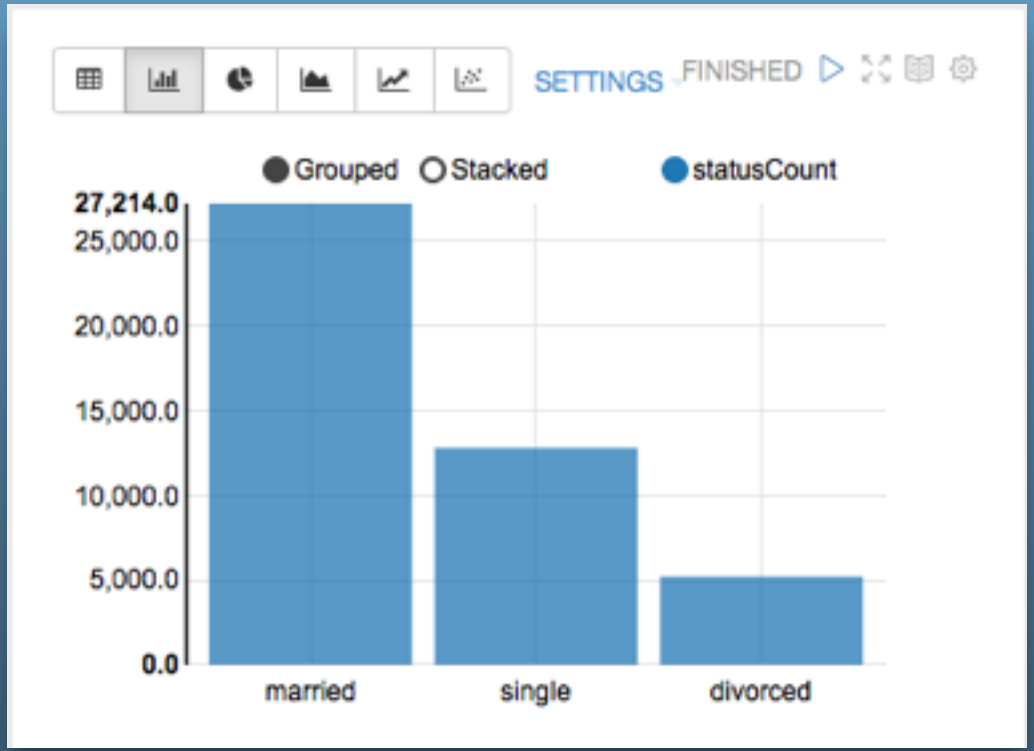
#ODSC

@OPENDATASCI



#ODSC

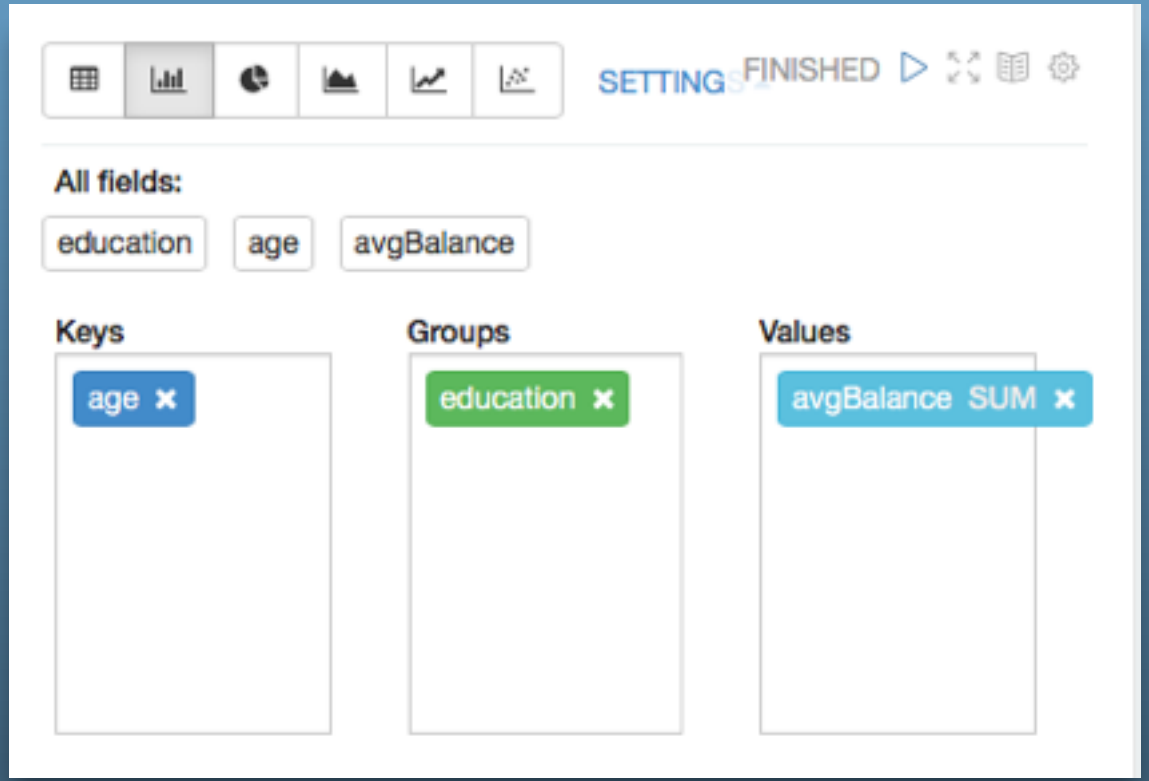
@OPENDATASCI



#ODSC

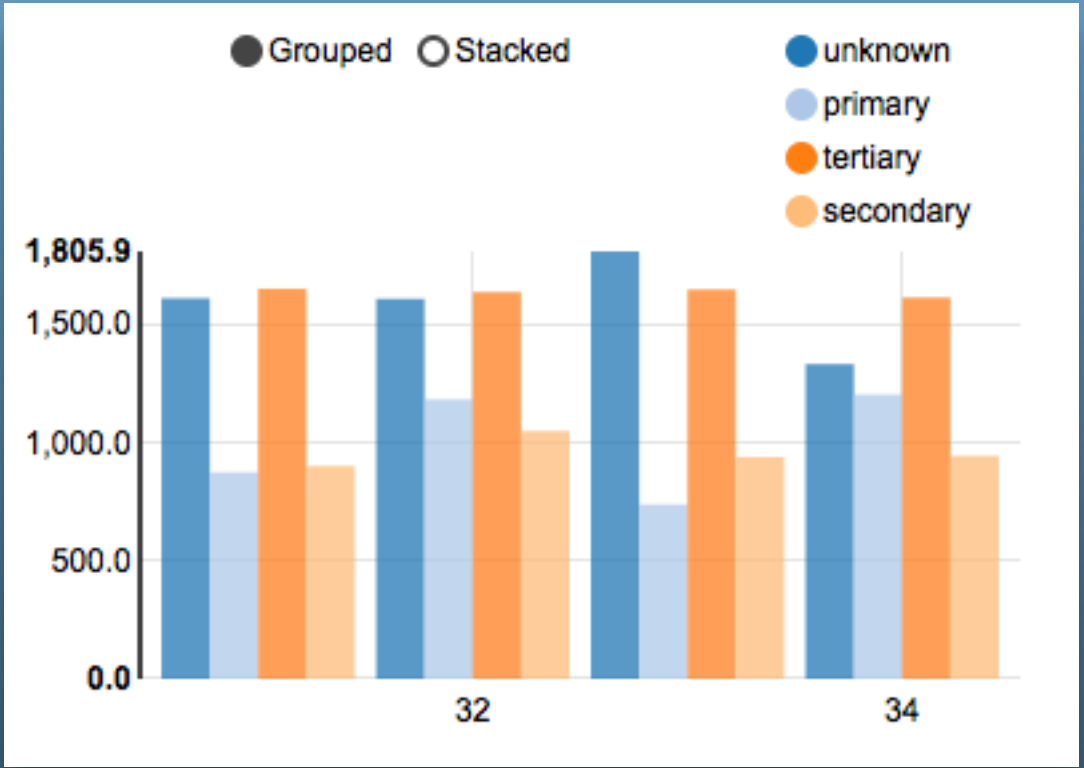
@OPENDATASCI

```
%sql
SELECT education, age, AVG( balance) as avgBalance
FROM bank
WHERE age > 20 AND age < 35
GROUP BY age, education
ORDER BY age ASC
```



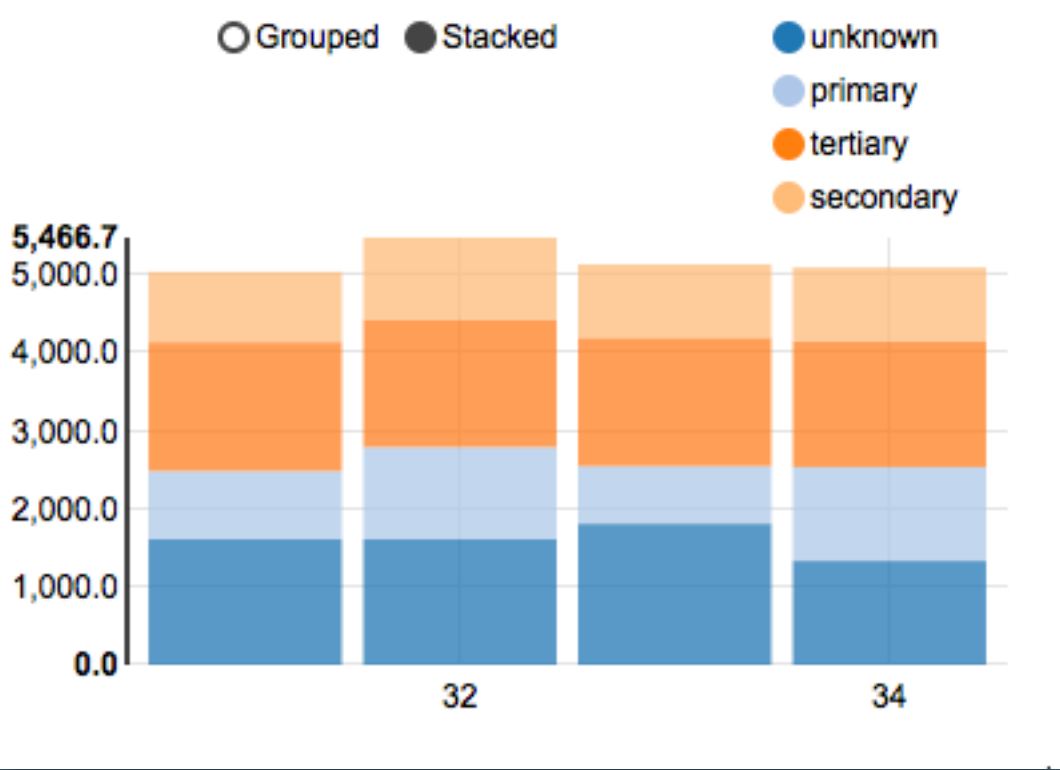
#ODSC

@OPENDATASCI



#ODSC

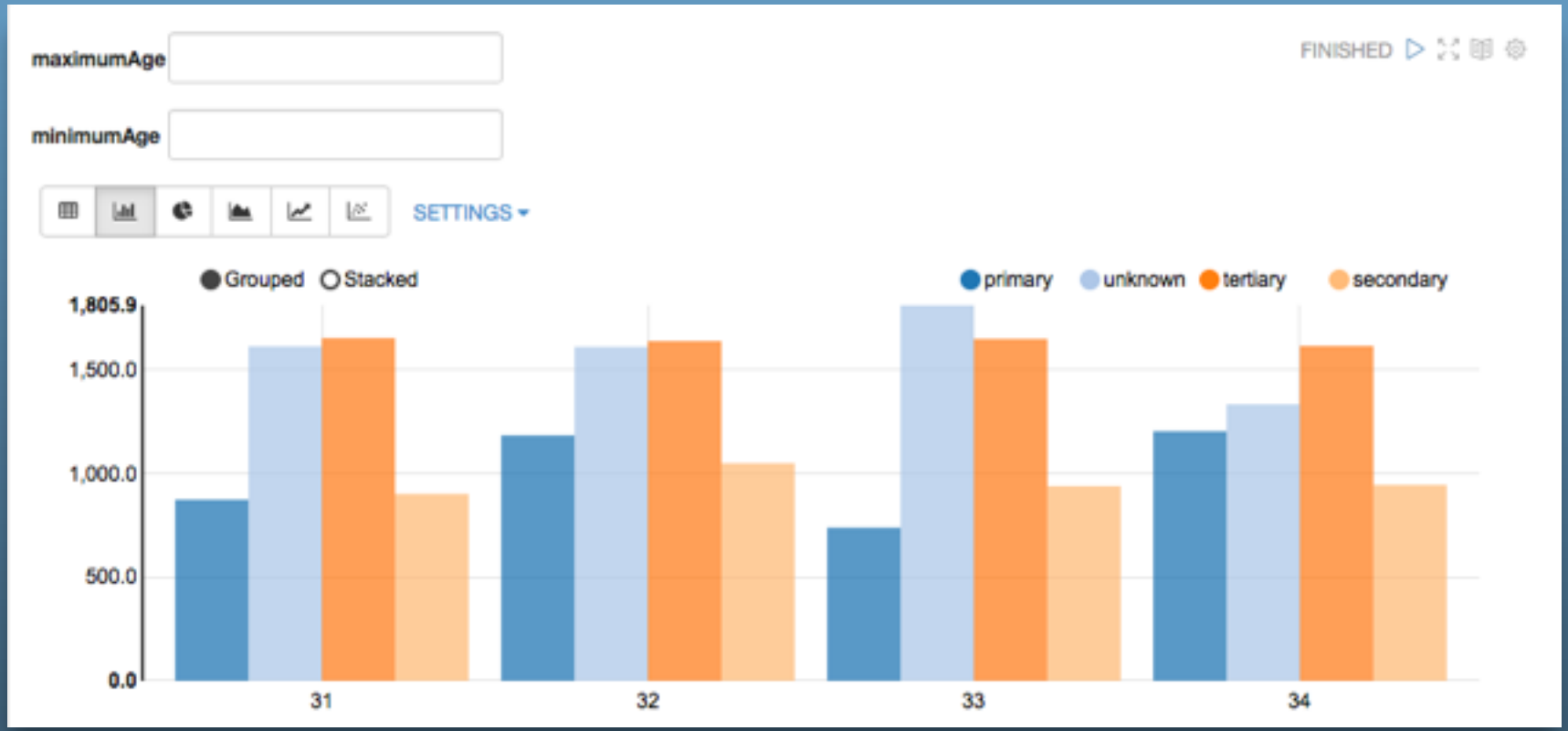
@OPENDATASCI



#ODSC

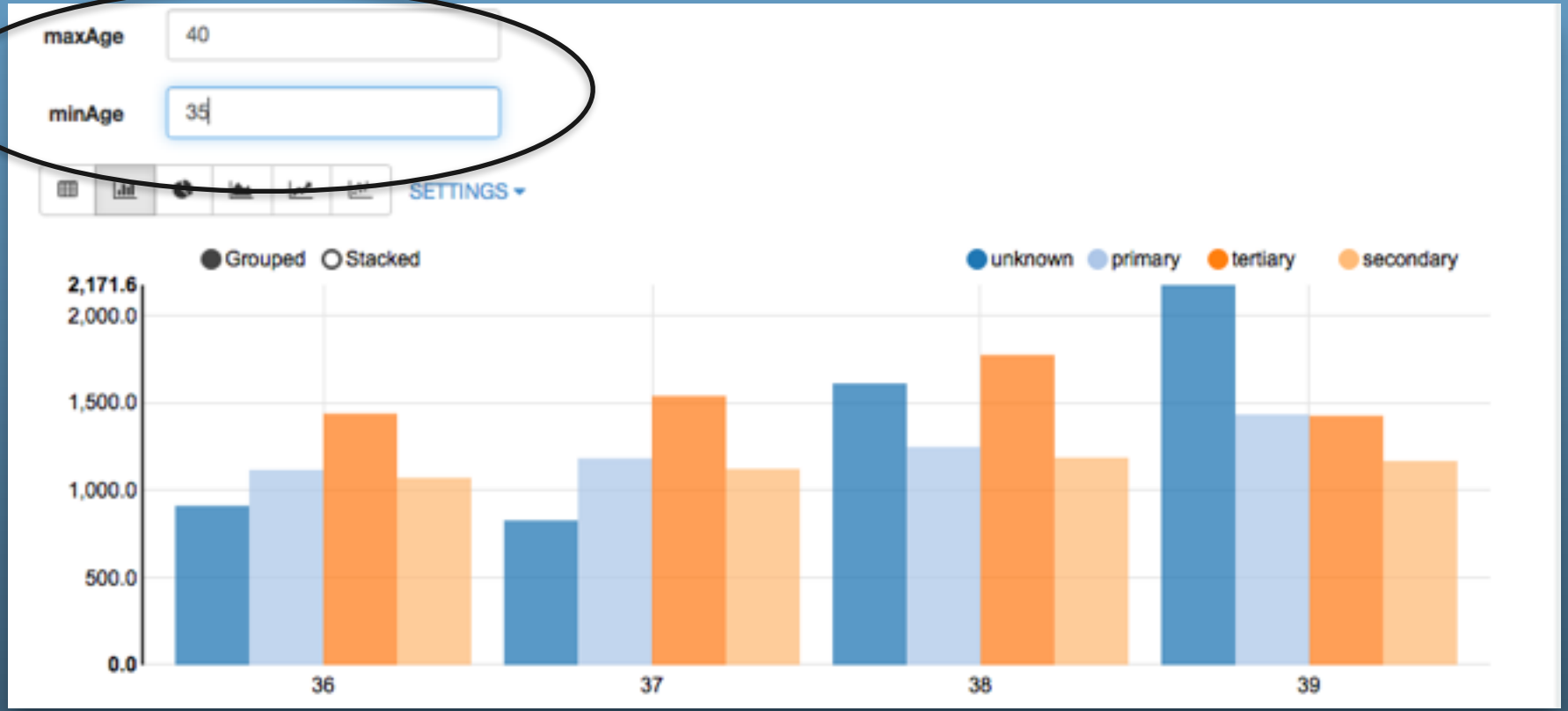
@OPENDATASCI

```
%sql
SELECT education, age, AVG( balance) as avgBalance
FROM bank
WHERE age > ${minimumAge=30} AND age < ${maximumAge=35}
GROUP BY age, education
ORDER BY age ASC
```



#ODSC

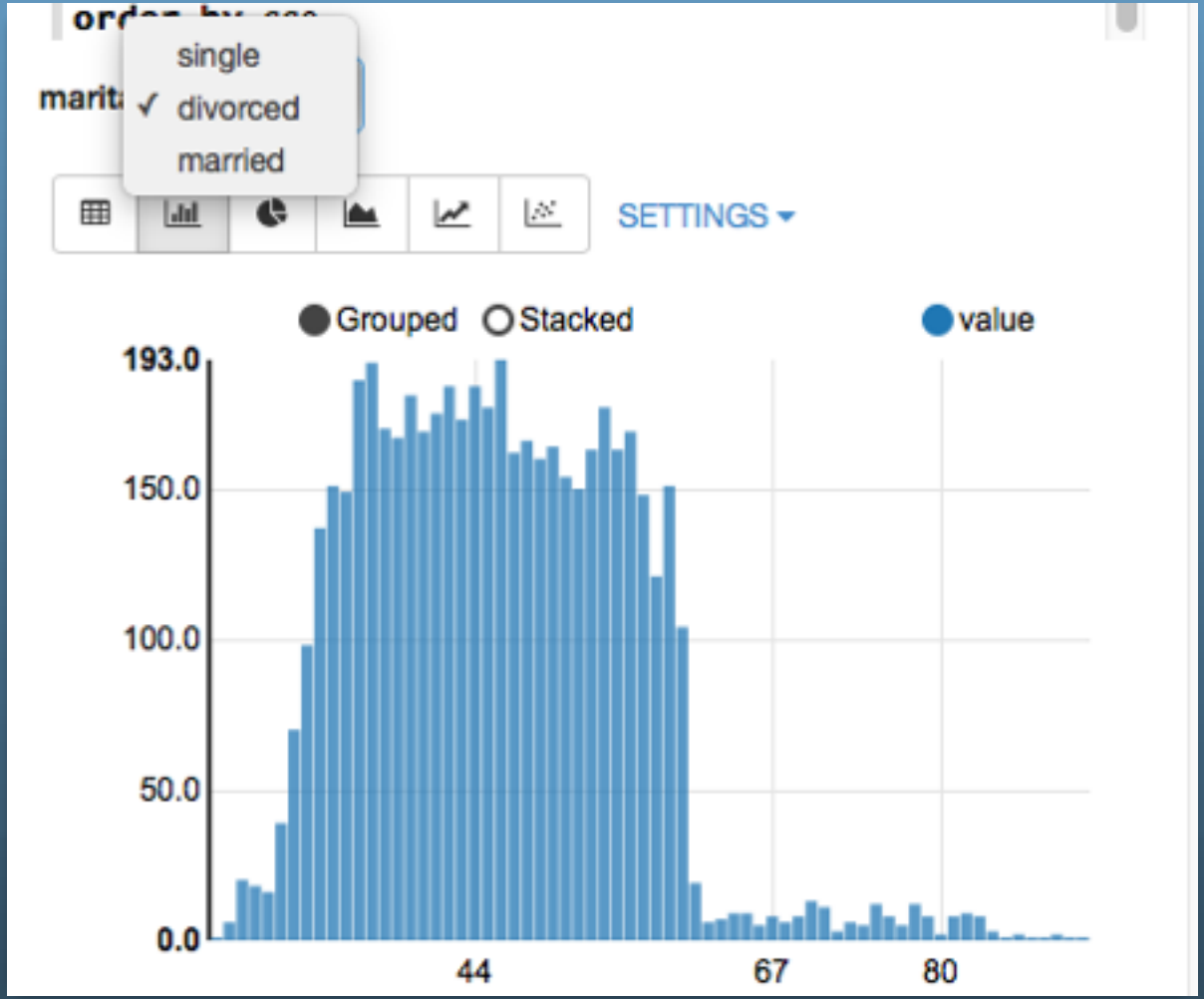
@OPENDATASCI



#ODSC

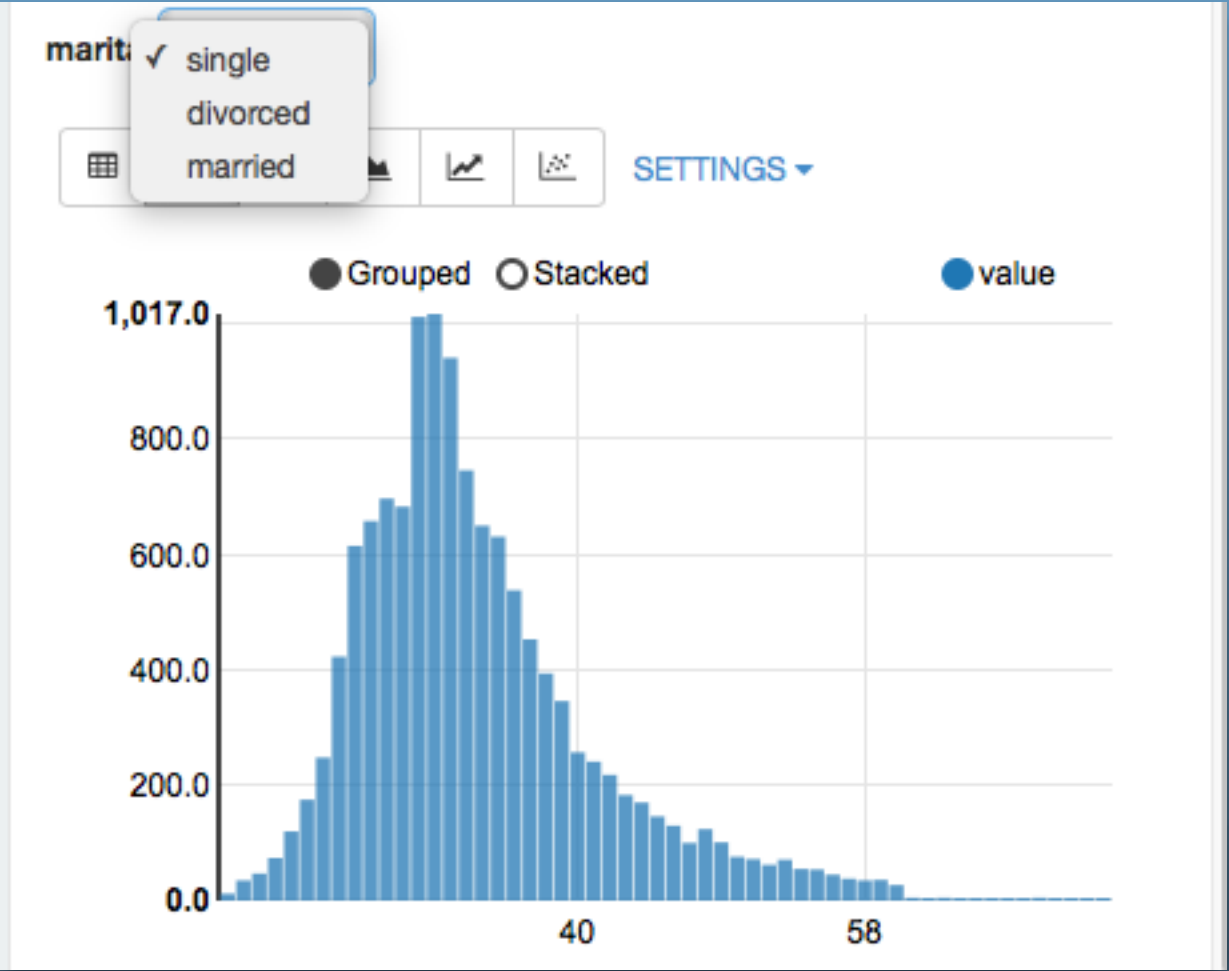
@OPENDATASCI

```
%sql
SELECT age, count(1) value
FROM bank
WHERE marital="${marital=single,single|divorced|married}"
GROUP BY age
ORDER BY age
```



#ODSC

@OPENDATASCI



#ODSC

@OPENDATASCI

https://zeppelin.incubator.apache.org/

@OPENDATASCI

Questions?

givre_charles@bah.com

@cgivre

#ODSC

@OPENDATASCI