New Data Lake/DataSci/Tutorial 2 U...

DataSci Tutorial 2: Using R with Zeppelin

FINISHED

This tutorial was built for BDCS-CE version 17.4.1 as part of the Data Science Acceleration User Journey: here (https://oracle.github.io/learning-library/workshops/journey3-data-science/). Questions and feedback about the tutorial: david.bayard@oracle.com (mailto:david.bayard@oracle.com)

```
Be sure you previously ran the Tutorial "Setup R, SparkR, RStudio Server".
```

This tutorial provides some examples of using R and SparkR in Zeppelin notebooks. It will show:

- How to query a hive table from R
- How to read data directly from the Object Store
- How to convert a R data.frame into a Spark Temporary Table and query it with SparkSQL
- Machine Learning with R and Spark
- Save results back to the Object Store

Took 0 sec. Last updated by anonymous at November 16 2017, 3:02:16 PM.

About R and Zeppelin

READY

Zeppelin includes an interpreter that is integrated with R and SparkR. You can find some details about Zeppelin and R here (https://zeppelin.apache.org/docs/0.7.0/interpreter/r.html). You can find some details about SparkR here (https://spark.apache.org/docs/2.1.0/sparkr.html).

These examples focuses on using R to work with Spark features like DataFrames. As the SparkR documentation writes, “A SparkDataFrame is a distributed collection of data organized into named columns. It is conceptually equivalent to a table in a relational database or a data frame in R, but with richer optimizations under the hood. SparkDataFrames can be constructed from a wide array of sources such as: structured data files, tables in Hive, external databases, or existing local R data frames”. This tutorial will demonstrate some of these capabilities.

Example of R querying Hive

READY

This example shows how to use R to query the bike_trips hive table via SparkR features.

SparkR to query a Hive table

FINISHED

```
%r
results <- sql("SELECT * from bike_trips")
head(results)
```

```
tripduration           starttime            stoptime startstationid
1          528 2016-12-01 00:00:04 2016-12-01 00:08:52            499
2          218 2016-12-01 00:00:28 2016-12-01 00:04:06           3418
3          399 2016-12-01 00:00:39 2016-12-01 00:07:19            297
4          254 2016-12-01 00:00:44 2016-12-01 00:04:59            405
5         1805 2016-12-01 00:00:54 2016-12-01 00:31:00            279
6          483 2016-12-01 00:01:13 2016-12-01 00:09:17            245
               startstationname startstationlatitude startstationlongitude
1             Broadway &amp; W 60 St          40.76915505          -73.98191841
2  Plaza St West &amp; Flatbush Ave           40.6750207          -73.97111473
3                E 15 St &amp; 3 Ave            40.734232            -73.986923
4 Washington St &amp; Gansevoort St            40.739323            -74.008119
5           Peck Slip &amp; Front St            40.707873             -74.00167
6    Myrtle Ave &amp; St Edwards St          40.69327018          -73.97703874
  endstationid                endstationname endstationlatitude
1          228               E 48 St &amp; 3 Ave      40.7546011026
2         3358           Garfield Pl &amp; 8 Ave         40.6711978
3          345               W 13 St &amp; 6 Ave      40.73649403
```

Took 9 sec. Last updated by anonymous at November 16 2017, 3:02:35 PM. (outdated)

FINISHED

```
%r
# let's see what kind of class our results are...
results
# It is a SparkDataFrame
```

```
SparkDataFrame[tripduration:int, starttime:timestamp, stoptime:timestamp, startstationid:string, startstationname:string, startstationlatitude:string, startstationlongitude:string, end
stationid:string, endstationname:string, endstationlatitude:string, endstationlongitude:string, bikeid:int, usertype:string, birthyear:int, gender:int]
```

Took 0 sec. Last updated by anonymous at November 16 2017, 3:03:41 PM.

READY

Example of reading a CSV from Object Store

This example shows SparkR features to read a CSV from Object Store via Spark's DataSources mechanisms

FINISHED

```
%r
biketrips <- read.df("swift://journeyC.default/citibike/raw/201612-citibike-tripdata.csv", "csv", header = "true", inferSchema = "true", na.strings = "NA")
head(biketrips)
```

```
Trip Duration          Start Time           Stop Time Start Station ID
1           528 2016-12-01 00:00:04 2016-12-01 00:08:52              499
2           218 2016-12-01 00:00:28 2016-12-01 00:04:06             3418
3           399 2016-12-01 00:00:39 2016-12-01 00:07:19              297
4           254 2016-12-01 00:00:44 2016-12-01 00:04:59              405
5          1805 2016-12-01 00:00:54 2016-12-01 00:31:00              279
6           483 2016-12-01 00:01:13 2016-12-01 00:09:17              245
             Start Station Name Start Station Latitude
1             Broadway &amp; W 60 St               40.76916
2  Plaza St West &amp; Flatbush Ave               40.67502
3                E 15 St &amp; 3 Ave               40.73423
```

```
4 Washington St &amp; Gansevoort St               40.73932
5         Peck Slip &amp; Front St               40.70787
6    Myrtle Ave &amp; St Edwards St               40.69327
  Start Station Longitude End Station ID              End Station Name
1               -73.98192            228               E 48 St &amp; 3 Ave
2               -73.97111           3358           Garfield Pl &amp; 8 Ave
3               -73.98692            345               W 13 St &amp; 6 Ave
4               -74.00812            358 Christopher St &amp; Greenwich St
5               -74.00167            279          Peck Slip &amp; Front St
6               -73.97704            372     Franklin Ave &amp; Myrtle Ave
  End Station Latitude End Station Longitude Bike ID  User Type Birth Year
1             40.75460             -73.97188   26931 Subscriber       1964
2             40.67120             -73.97484   27122 Subscriber       1955
3             40.73649             -73.99704   19352 Subscriber       1985
4             40.73292             -74.00711   20015 Subscriber       1982
5             40.70787             -74.00167   23148 Subscriber       1989
6             40.69453             -73.95809   16140 Subscriber       1986
```

Took 24 sec. Last updated by anonymous at November 16 2017, 3:04:12 PM.

Example of making a R dataframe into a SparkSQL table

READY

Here is an example of converting a R data.frame into a Spark DataFrame and registering it as a Spark SQL table. You can more examples like this here (https://rpubs.com/wendyu/sparkr).

Load an R data frame

FINISHED

```
%r
data(iris)
iris
```

```
Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1           5.1         3.5          1.4         0.2     setosa
2           4.9         3.0          1.4         0.2     setosa
3           4.7         3.2          1.3         0.2     setosa
4           4.6         3.1          1.5         0.2     setosa
5           5.0         3.6          1.4         0.2     setosa
6           5.4         3.9          1.7         0.4     setosa
7           4.6         3.4          1.4         0.3     setosa
8           5.0         3.4          1.5         0.2     setosa
9           4.4         2.9          1.4         0.2     setosa
10          4.9         3.1          1.5         0.1     setosa
11          5.4         3.7          1.5         0.2     setosa
12          4.8         3.4          1.6         0.2     setosa
13          4.8         3.0          1.4         0.1     setosa
14          4.3         3.0          1.1         0.1     setosa
15          5.8         4.0          1.2         0.2     setosa
16          5.7         4.4          1.5         0.4     setosa
17          5.4         3.9          1.3         0.4     setosa
18          5.1         3.5          1.4         0.3     setosa
19          5.7         3.8          1.7         0.3     setosa
20          5.1         3.8          1.5         0.3     setosa
21          5.4         3.4          1.7         0.2     setosa
```

```
22          5.1         3.7          1.5          0.4     setosa
23          4.6         3.6          1.0          0.2     setosa
24          5.1         3.3          1.7          0.5     setosa
25          4.8         3.4          1.9          0.2     setosa
26          5.0         3.0          1.6          0.2     setosa
27          5.0         3.4          1.6          0.4     setosa
28          5.2         3.5          1.5          0.2     setosa
29          5.2         3.4          1.4          0.2     setosa
30          4.7         3.2          1.6          0.2     setosa
31          4.8         3.1          1.6          0.2     setosa
32          5.4         3.4          1.5          0.4     setosa
33          5.2         4.1          1.5          0.1     setosa
34          5.5         4.2          1.4          0.2     setosa
35          4.9         3.1          1.5          0.2     setosa
36          5.0         3.2          1.2          0.2     setosa
37          5.5         3.5          1.3          0.2     setosa
38          4.9         3.6          1.4          0.1     setosa
39          4.4         3.0          1.3          0.2     setosa
40          5.1         3.4          1.5          0.2     setosa
41          5.0         3.5          1.3          0.3     setosa
42          4.5         2.3          1.3          0.3     setosa
43          4.4         3.2          1.3          0.2     setosa
44          5.0         3.5          1.6          0.6     setosa
45          5.1         3.8          1.9          0.4     setosa
46          4.8         3.0          1.4          0.3     setosa
47          5.1         3.8          1.6          0.2     setosa
48          4.6         3.2          1.4          0.2     setosa
49          5.3         3.7          1.5          0.2     setosa
50          5.0         3.3          1.4          0.2     setosa
51          7.0         3.2          4.7          1.4 versicolor
52          6.4         3.2          4.5          1.5 versicolor
53          6.9         3.1          4.9          1.5 versicolor
54          5.5         2.3          4.0          1.3 versicolor
55          6.5         2.8          4.6          1.5 versicolor
56          5.7         2.8          4.5          1.3 versicolor
57          6.3         3.3          4.7          1.6 versicolor
58          4.9         2.4          3.3          1.0 versicolor
59          6.6         2.9          4.6          1.3 versicolor
60          5.2         2.7          3.9          1.4 versicolor
61          5.0         2.0          3.5          1.0 versicolor
62          5.9         3.0          4.2          1.5 versicolor
63          6.0         2.2          4.0          1.0 versicolor
64          6.1         2.9          4.7          1.4 versicolor
65          5.6         2.9          3.6          1.3 versicolor
66          6.7         3.1          4.4          1.4 versicolor
67          5.6         3.0          4.5          1.5 versicolor
68          5.8         2.7          4.1          1.0 versicolor
69          6.2         2.2          4.5          1.5 versicolor
70          5.6         2.5          3.9          1.1 versicolor
71          5.9         3.2          4.8          1.8 versicolor
72          6.1         2.8          4.0          1.3 versicolor
73          6.3         2.5          4.9          1.5 versicolor
74          6.1         2.8          4.7          1.2 versicolor
75          6.4         2.9          4.3          1.3 versicolor
76          6.6         3.0          4.4          1.4 versicolor
```

```
77           6.8         2.8          4.8         1.4 versicolor
78           6.7         3.0          5.0         1.7 versicolor
79           6.0         2.9          4.5         1.5 versicolor
80           5.7         2.6          3.5         1.0 versicolor
81           5.5         2.4          3.8         1.1 versicolor
82           5.5         2.4          3.7         1.0 versicolor
83           5.8         2.7          3.9         1.2 versicolor
84           6.0         2.7          5.1         1.6 versicolor
85           5.4         3.0          4.5         1.5 versicolor
86           6.0         3.4          4.5         1.6 versicolor
87           6.7         3.1          4.7         1.5 versicolor
88           6.3         2.3          4.4         1.3 versicolor
89           5.6         3.0          4.1         1.3 versicolor
90           5.5         2.5          4.0         1.3 versicolor
91           5.5         2.6          4.4         1.2 versicolor
92           6.1         3.0          4.6         1.4 versicolor
93           5.8         2.6          4.0         1.2 versicolor
94           5.0         2.3          3.3         1.0 versicolor
95           5.6         2.7          4.2         1.3 versicolor
96           5.7         3.0          4.2         1.2 versicolor
97           5.7         2.9          4.2         1.3 versicolor
98           6.2         2.9          4.3         1.3 versicolor
99           5.1         2.5          3.0         1.1 versicolor
100          5.7         2.8          4.1         1.3 versicolor
101          6.3         3.3          6.0         2.5  virginica
102          5.8         2.7          5.1         1.9  virginica
103          7.1         3.0          5.9         2.1  virginica
104          6.3         2.9          5.6         1.8  virginica
105          6.5         3.0          5.8         2.2  virginica
106          7.6         3.0          6.6         2.1  virginica
107          4.9         2.5          4.5         1.7  virginica
108          7.3         2.9          6.3         1.8  virginica
109          6.7         2.5          5.8         1.8  virginica
110          7.2         3.6          6.1         2.5  virginica
111          6.5         3.2          5.1         2.0  virginica
112          6.4         2.7          5.3         1.9  virginica
113          6.8         3.0          5.5         2.1  virginica
114          5.7         2.5          5.0         2.0  virginica
115          5.8         2.8          5.1         2.4  virginica
116          6.4         3.2          5.3         2.3  virginica
117          6.5         3.0          5.5         1.8  virginica
118          7.7         3.8          6.7         2.2  virginica
119          7.7         2.6          6.9         2.3  virginica
120          6.0         2.2          5.0         1.5  virginica
121          6.9         3.2          5.7         2.3  virginica
122          5.6         2.8          4.9         2.0  virginica
123          7.7         2.8          6.7         2.0  virginica
124          6.3         2.7          4.9         1.8  virginica
125          6.7         3.3          5.7         2.1  virginica
126          7.2         3.2          6.0         1.8  virginica
127          6.2         2.8          4.8         1.8  virginica
128          6.1         3.0          4.9         1.8  virginica
129          6.4         2.8          5.6         2.1  virginica
130          7.2         3.0          5.8         1.6  virginica
131          7.4         2.8          6.1         1.9  virginica
```

```
132          7.9         3.8          6.4         2.0  virginica
133          6.4         2.8          5.6         2.2  virginica
134          6.3         2.8          5.1         1.5  virginica
135          6.1         2.6          5.6         1.4  virginica
136          7.7         3.0          6.1         2.3  virginica
137          6.3         3.4          5.6         2.4  virginica
138          6.4         3.1          5.5         1.8  virginica
139          6.0         3.0          4.8         1.8  virginica
140          6.9         3.1          5.4         2.1  virginica
141          6.7         3.1          5.6         2.4  virginica
142          6.9         3.1          5.1         2.3  virginica
143          5.8         2.7          5.1         1.9  virginica
144          6.8         3.2          5.9         2.3  virginica
145          6.7         3.3          5.7         2.5  virginica
146          6.7         3.0          5.2         2.3  virginica
147          6.3         2.5          5.0         1.9  virginica
148          6.5         3.0          5.2         2.0  virginica
```

Took 0 sec. Last updated by anonymous at November 16 2017, 3:05:24 PM.

SparkR code to register an R dataframe as a SparkSQL table

FINISHED

```
%r
irisDF <- as.DataFrame(iris)
registerTempTable(irisDF,"iris")
irisDF
```

```
SparkDataFrame[Sepal_Length:double, Sepal_Width:double, Petal_Length:double, Petal_Width:double, Species:string]
```

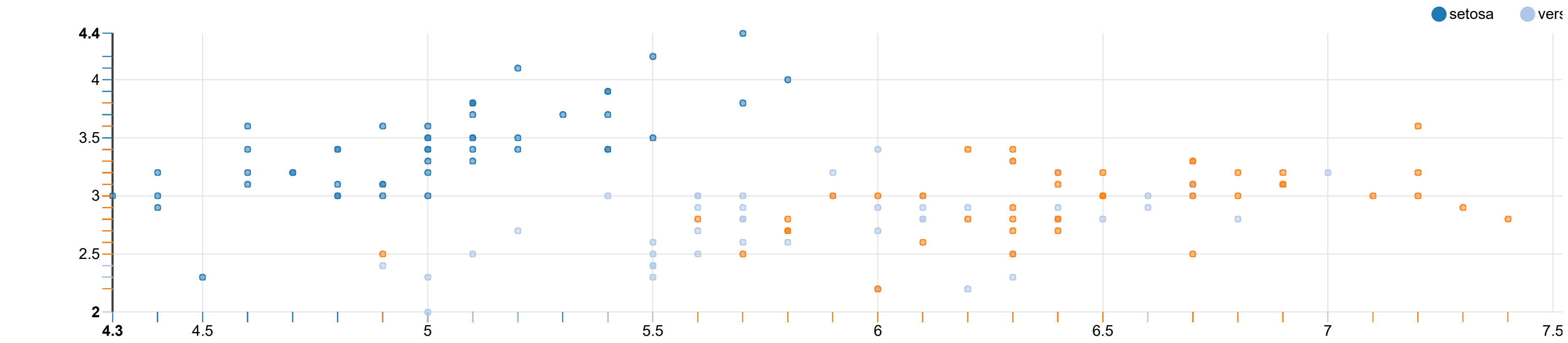
Took 4 sec. Last updated by anonymous at November 16 2017, 3:05:31 PM.

SparkSQL querying R data

FINISHED

```
%sql
select * from iris
```

settings ▾



Took 3 sec. Last updated by anonymous at November 16 2017, 3:05:42 PM.

Machine Learning with R and Spark

READY

This example shows running a Spark machine learning algorithm - Generalized Linear Model (glm).

We will use our citibike data and model tripduration based on age and gender.

SparkR code to build generalized linear model (GLM) of tripduration based on gender and age

FINISHED

```
%r
ageGender  <- sql("SELECT tripduration, (2016-birthyear) age, gender from bike_trips")
training <- dropna(ageGender)

model <- glm(tripduration ~ age + gender,
    family = "gaussian", data = training)
summary(model)
```

```
Deviance Residuals:
(Note: These are approximate quantiles with relative error &lt;= 0.01)
    Min       1Q   Median       3Q      Max
   -850     -389     -199      117  3472328
Coefficients:
             Estimate  Std. Error  t value  Pr(&gt;|t|)
(Intercept)  529.52    34.083      15.536   0
age          2.7453    0.62725     4.3768   1.2048e-05
gender       66.934    17.791      3.7623   0.00016837
(Dispersion parameter for gaussian family taken to be 43709180)
Null deviance: 3.3766e+13  on 772487  degrees of freedom
Residual deviance: 3.3765e+13  on 772485  degrees of freedom
AIC: 15782665
Number of Fisher Scoring iterations: 1
```

Took 42 sec. Last updated by anonymous at November 16 2017, 3:06:31 PM.

Check our predictions (not so good)

FINISHED

```
%r
fitted <- predict(model, training)
registerTempTable(fitted,"fitted")
compare <- sql("select prediction, tripduration, age, gender from fitted")
head(compare)
```

```
prediction tripduration age gender
1   739.2069          528  52      1
2   763.9147          218  61      1
3   681.5551          399  31      1
4   689.7911          254  34      1
5   670.5738         1805  27      1
6   678.8098          483  30      1
```

Took 6 sec. Last updated by anonymous at November 16 2017, 3:08:06 PM.

SparkSQL to view predictions

FINISHED

```
%sql
select prediction, tripduration, gender, age from fitted
limit 100
```

prediction	tripduration	gender	age
739.2068577724133	528	1	52
763.9147413320832	218	1	61
681.5551294665167	399	1	31
689.7910906530734	254	1	34
670.5738478844412	1805	1	27
678.8098090709979	483	1	30
739.2068577724133	1114	1	52
766.660061727602	356	1	62
678.8098090709979	298	1	30

Took 0 sec. Last updated by anonymous at November 16 2017, 3:08:12 PM.

Example of writing a DataFrame back to Object Store

READY

The follow shows an example of writing a DataFrame back to the Object Store. We use the write.df method from SparkR. It supports multiple source types (csv, json, parquet, etc).

R code to write our predictions back to the Object Store

FINISHED

```
%r
# Since we know the resulting file is small, we will do a repartition command to force Spark to write the output as a single file.  This is an optional step.
fitted_singlepartition <- repartition(fitted,1)
write.df(fitted_singlepartition, "swift://journeyC.default/citibike/results/201612-fitted-projections", source="csv", mode="overwrite")
```

Took 58 sec. Last updated by anonymous at November 16 2017, 3:09:19 PM.

Explore the contents of the Object Store

FINISHED

```
%sh
# this command will show you the contents of the Object Store that were just written
hadoop fs -ls swift://journeyC.default/citibike/results/201612-fitted-projections
```

```
Found 2 items
drw-rw-rw-   -          0 2017-11-16 20:09 swift://journeyC.default/citibike/results/201612-fitted-projections/_SUCCESS
-rw-rw-rw-   1   25628332 2017-11-16 20:09 swift://journeyC.default/citibike/results/201612-fitted-projections/part-00000-a5622ae8-77ba-4e47-936f-c17890875927.csv
```

Took 14 sec. Last updated by anonymous at November 16 2017, 3:11:34 PM. (outdated)

Change Log

FINISHED

November 16, 2017 - Confirmed it works with 17.4.1

September 12, 2017 - Confirmed it works with 17.3.5

August 13, 2017 - Confirmed it works with BDCSCE 17.3.3-20

August 11, 2017 - First version

Took 0 sec. Last updated by anonymous at November 16 2017, 3:08:53 PM.

```
%md
```

READY