RSutdio对接FusionInsight Spark

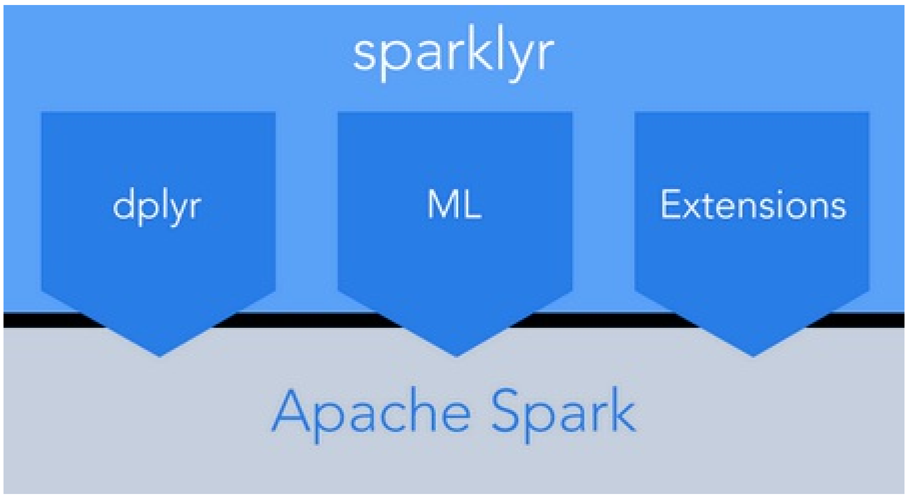
适用场景

> R-3.4.1 ↔ FusionInsight V100R002C60U10
>
> R-3.4.1 ↔ FusionInsight V100R002C70SPC100

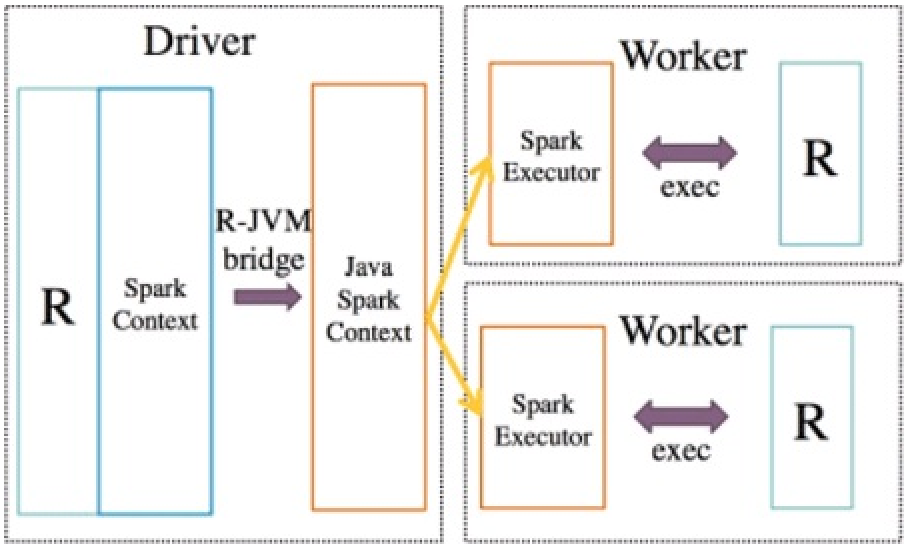
对接方式

RStudio与Spark集成有两种方式：

- 通过RStudio官方发布的sparklyr与Spark进行集成



- 通过Apache Spark社区发布的SparkR进行集成



本文档包含了两种方式对接的步骤, 相关对接步骤如下：

- 安装R
- 安装RStudio Server
- 安装FusionInsight客户端
- 使用SparkR与RStudio集成进行分析
  - 在RStudio中使用SparkR进行数据分析
- 使用RStudio Sparklyr和Spark集成进行分析
  - 使用sparklyr结合spark进行数据分析babynames数据集
  - 使用sparklyr结合spark进行数据分析航空公司飞行数据(必须配套Spark2x)

安装R

由于Spark的Executor上也需要执行R，所以除了在RStudio的节点上安装R以外，所有FusionInsight集群节点上也要安装同版本的R，安装步骤如下：

> 本文使用的RStudio节点为Redhat7.1，FusionInsight集群节点为Redhat6.6

- 配置redhat的yum源，国内可以配置aliyun的源或者163的源
- 配置EPEL的源
- 安装R-3.4.1

配置aliyun的源

- 配置好Redhat7.1的yum源

```
cd ~
rpm -qa|grep yum|xargs rpm -e --nodeps
rpm -qa|grep python-urlgrabber|xargs rpm -e --nodeps
wget https://mirrors.aliyun.com/centos/7/os/x86_64/Packages/yum-metadata-parser-1.1.4-10.el7.x86_64.rpm
wget https://mirrors.aliyun.com/centos/7/os/x86_64/Packages/yum-3.4.3-150.el7.centos.noarch.rpm
wget https://mirrors.aliyun.com/centos/7/os/x86_64/Packages/yum-rhn-plugin-2.0.1-6.el7.noarch.rpm
wget https://mirrors.aliyun.com/centos/7/os/x86_64/Packages/yum-plugin-fastestmirror-1.1.31-40.el7.noarch.rpm
wget https://mirrors.aliyun.com/centos/7/os/x86_64/Packages/python-urlgrabber-3.10-8.el7.noarch.rpm
rpm -ivh *.rpm
cd /etc/yum.repos.d/
wget https://mirrors.aliyun.com/repo/Centos-7.repo
sed -i 's/$releasever/7/g' /etc/yum.repos.d/Centos-7.repo
yum clean
yum makecache
```

配置163的源

- 配置好Redhat6.6的yum源

```
cd ~
rpm -aq | grep yum | xargs rpm -e --nodeps
wget http://mirrors.163.com/centos/6/os/x86_64/Packages/python-iniparse-0.3.1-2.1.el6.noarch.rpm
wget http://mirrors.163.com/centos/6/os/x86_64/Packages/yum-metadata-parser-1.1.2-16.el6.x86_64.rpm
wget http://mirrors.163.com/centos/6/os/x86_64/Packages/yum-3.2.29-81.el6.centos.noarch.rpm
wget http://mirrors.163.com/centos/6/os/x86_64/Packages/yum-plugin-fastestmirror-1.1.30-40.el6.noarch.rpm
wget http://mirrors.163.com/centos/6/os/x86_64/Packages/python-urlgrabber-3.9.1-11.el6.noarch.rpm
rpm -ivh python-iniparse-0.3.1-2.1.el6.noarch.rpm
rpm -ivh yum-metadata-parser-1.1.2-16.el6.x86_64.rpm
rpm -U python-urlgrabber-3.9.1-11.el6.noarch.rpm
rpm -ivh yum-3.2.29-81.el6.centos.noarch.rpm yum-plugin-fastestmirror-1.1.30-40.el6.noarch.rpm
cd /etc/yum.repos.d/
wget http://mirrors.163.com/.help/CentOS6-Base-163.repo
sed -i 's/$releasever/6/g' /etc/yum.repos.d/CentOS6-Base-163.repo
yum clean all
yum makecache
```

配置EPEL的源

- 安装EPEL源

  Redhat 6.x 使用下面命令安装

  ```
  rpm -Uvh https://mirrors.tuna.tsinghua.edu.cn/epel//6/x86_64/epel-release-6-8.noarch.rpm
  ```

  Redhat 7.x 使用下面命令安装

  ```
  rpm -Uvh https://mirrors.tuna.tsinghua.edu.cn/epel//7/x86_64/e/epel-release-7-10.noarch.rpm
  ```

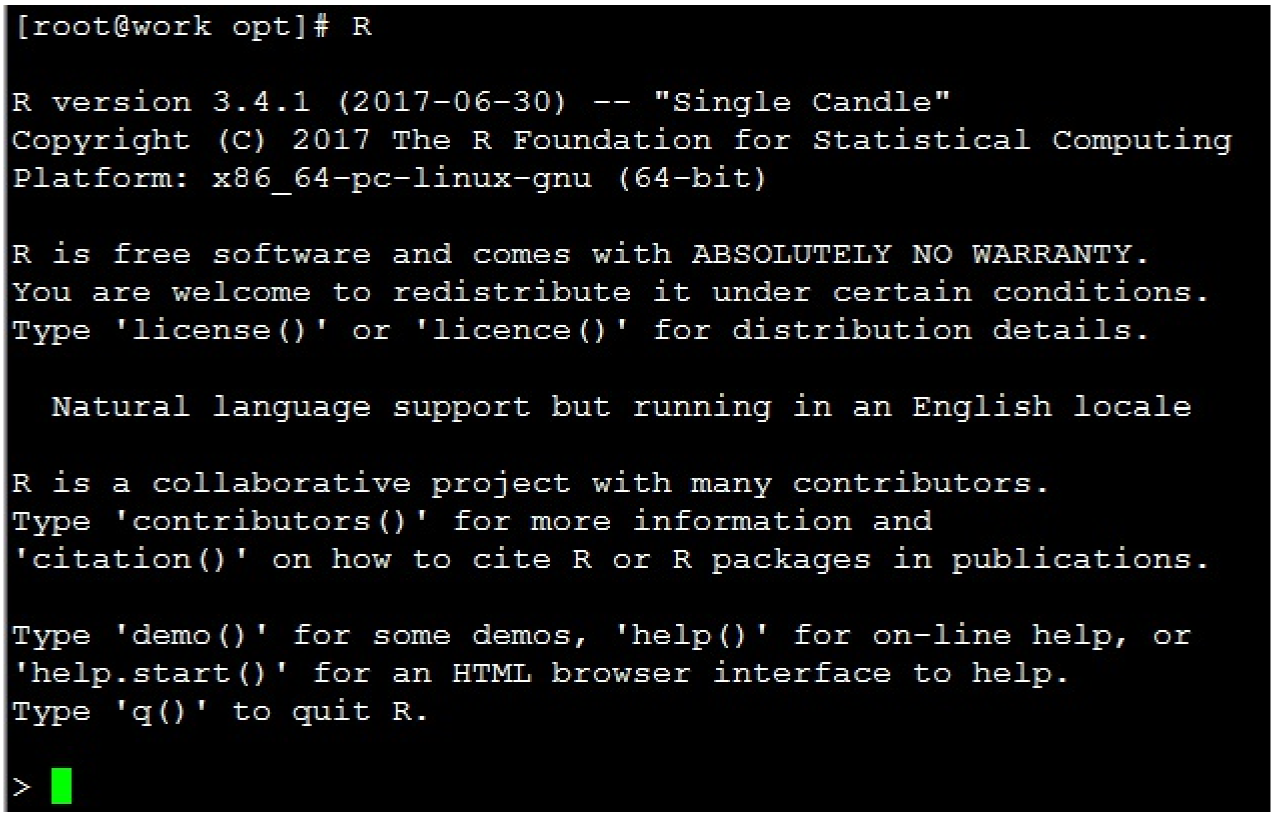
- 更新cache

  ```
  yum clean all
  yum makecache
  ```

安装R-3.4.1

- 执行 `yum install R` 安装R的相关的包
- 执行 `R`，检查R是否可用

正常启动如下图所示



```
[root@work opt]# R

R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

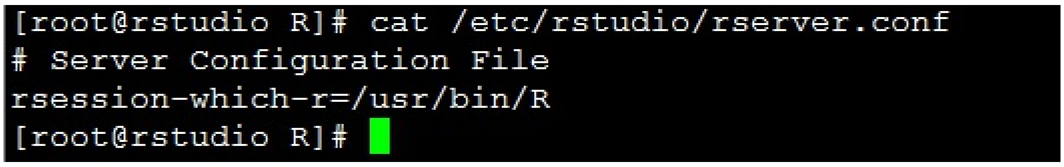
安装RStudio Server

- 下载并安装RStudio Server

```
wget https://download2.rstudio.org/rstudio-server-rhel-1.0.153-x86_64.rpm
yum install --nogpgcheck rstudio-server-rhel-1.0.153-x86_64.rpm
```

- 使用 `vi /etc/rstudio/rserver.conf` 修改RStudio的配置文件，指定RStudio Server使用的R的路径

```
rsession-which-r=/usr/bin/R
```

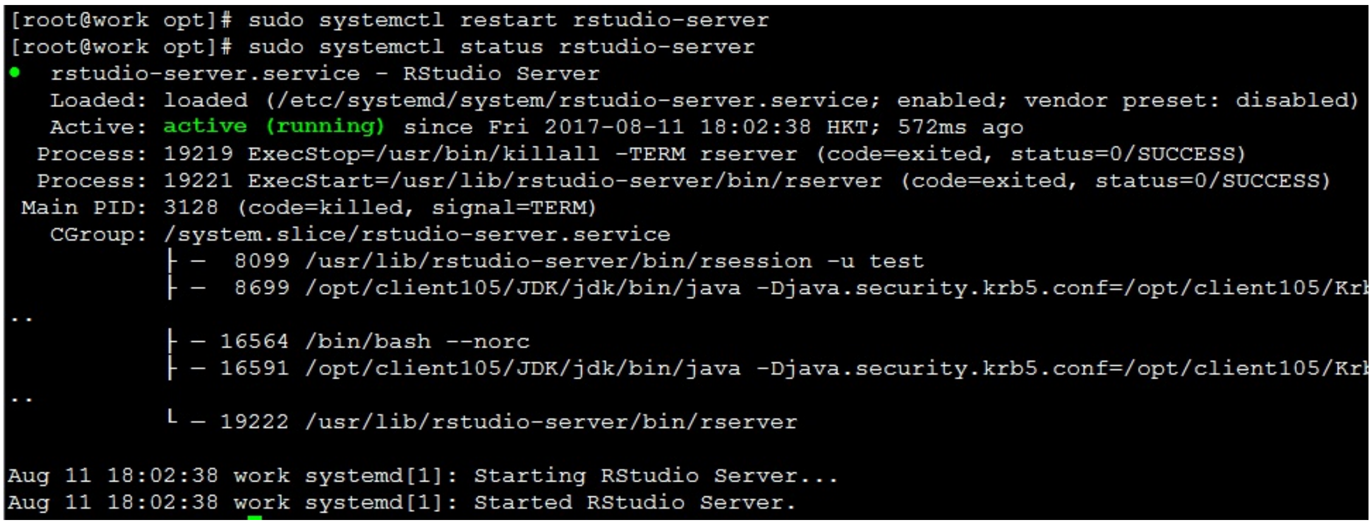


```
[root@rstudio R]# cat /etc/rstudio/rserver.conf
# Server Configuration File
rsession-which-r=/usr/bin/R
[root@rstudio R]#
```

- 重启rstudio-server后，查看服务是否正常

```
sudo systemctl restart rstudio-server
sudo systemctl status rstudio-server
```

- 服务正常启动如下



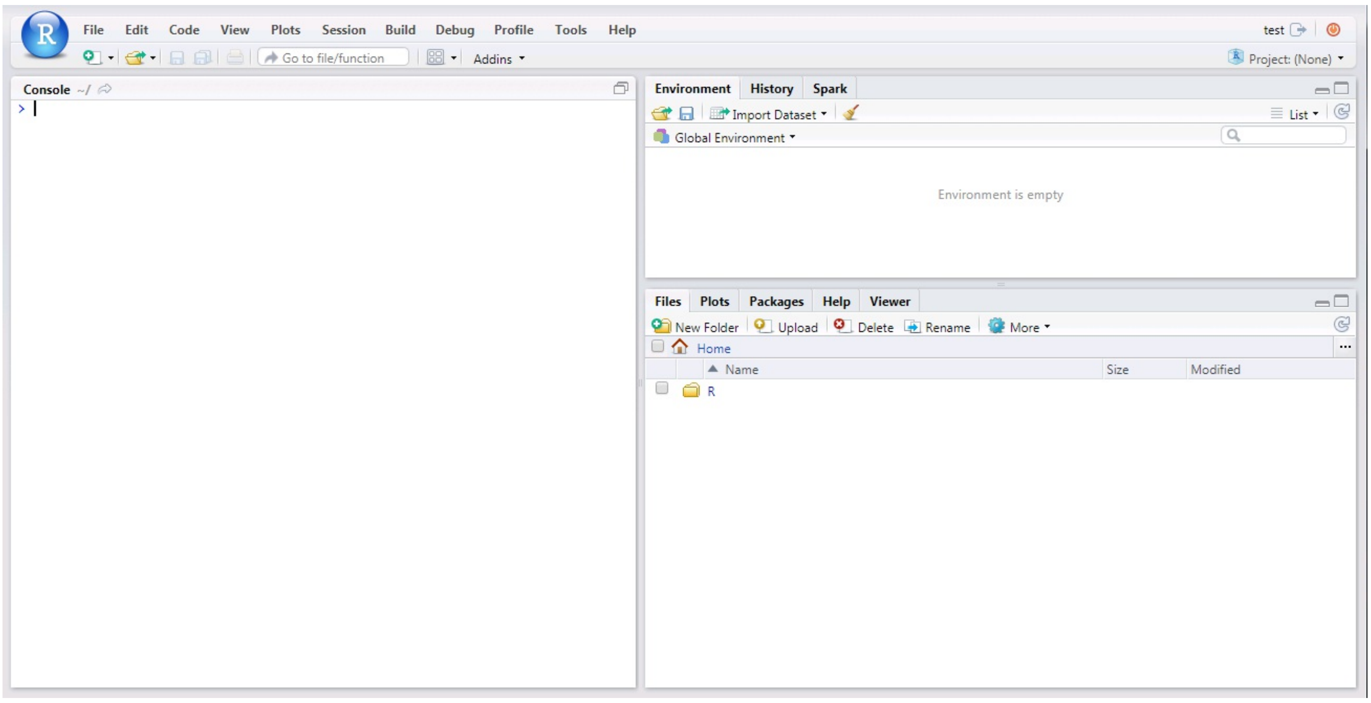
```
[root@work opt]# sudo systemctl restart rstudio-server
[root@work opt]# sudo systemctl status rstudio-server
● rstudio-server.service - RStudio Server
   Loaded: loaded (/etc/systemd/system/rstudio-server.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2017-08-11 18:02:38 HKT; 572ms ago
  Process: 19219 ExecStop=/usr/bin/killall -TERM rserver (code=exited, status=0/SUCCESS)
  Process: 19221 ExecStart=/usr/lib/rstudio-server/bin/rserver (code=exited, status=0/SUCCESS)
 Main PID: 3128 (code=killed, signal=TERM)
   CGroup: /system.slice/rstudio-server.service
           ├─ 8099 /usr/lib/rstudio-server/bin/rsession -u test
           ├─ 8699 /opt/client105/JDK/jdk/bin/java -Djava.security.krb5.conf=/opt/client105/Kr
..
           ├─16564 /bin/bash --norc
           ├─16591 /opt/client105/JDK/jdk/bin/java -Djava.security.krb5.conf=/opt/client105/Kr
..
           └─19222 /usr/lib/rstudio-server/bin/rserver

Aug 11 18:02:38 work systemd[1]: Starting RStudio Server...
Aug 11 18:02:38 work systemd[1]: Started RStudio Server.
```

- 由于RStudio Server不允许使用root用户登陆，需要新建一个普通用户用于Web界面的登陆

```
useradd -d /home/test -m test
passwd test
```

- 用户新建完成后，关闭防火墙，然后使用本机ip:8787端口访问RStudio Server，使用新建的test用户登陆即可进入RStudio的Web开发界面

```
sudo systemctl stop firewalld
```


File Edit Code View Plots Session Build Debug Profile Tools Help
test
Go to file/function
Addins
Project: (None)
Console ~/
Environment History Spark
Import Dataset
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder Upload Delete Rename More
Home
Name Size Modified
R

安装FusionInsight客户端

- 登录FusionInsight Manager系统，单击 **服务管理** ，在菜单栏中单击 **下载客户端**，客户端类型勾选 **完整客户端**，是否在集群的节点中生成客户端文件选择否
- 使用WinSCP工具将下载下来的软件包上传到Linux服务器的目录，例如 `/tmp/client`
- 切换到新建的test用户

```
su test
```

- 解压软件包。进入安装包所在目录，例如 `/tmp/client` 。执行如下命令解压安装包到本地目录

```
cd /tmp/client
tar -xvf FusionInsight_V100R002C60U20_Services_Client.tar
tar -xvf FusionInsight_V100R002C60U20_Services_ClientConfig.tar
```

- 进入安装包所在目录，执行如下命令安装客户端到指定目录（绝对路径），例如安装到 `/home/test/hadoopclient` 目录

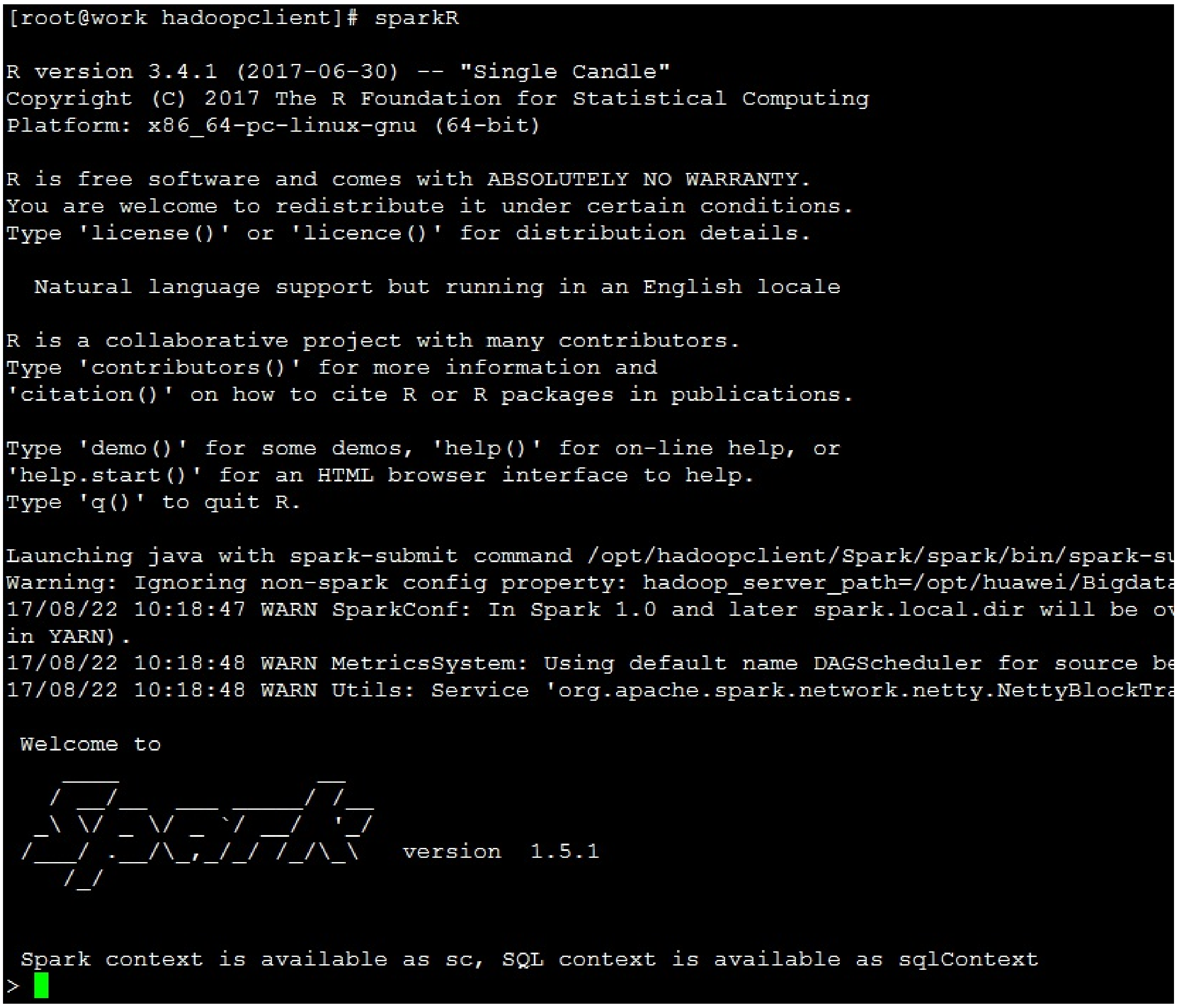
  ```
  cd /opt/tmp/FusionInsight_V100R002C60U20_Services_ClientConfig
  ./install.sh /home/test/hadoopclient
  ```

- 客户端将被安装到 `/home/test/hadoopclient` 目录中
- 检查客户端节点与FusionInsight集群时间同步（差距不能超过5分钟）
- 检查SparkR是否可用

  使用sparkuser进行Kerberos认证(sparkuser为FusionInsight中创建的拥有Spark访问权限的人机用户)

  ```
  cd /home/test/hadoopclient
  source bigdata_env
  kinit sparkuser
  ```

  执行 `sparkR` 启动SparkR, 正常启动出现以下界面



```
[root@work hadoopclient]# sparkR

R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Launching java with spark-submit command /opt/hadoopclient/Spark/spark/bin/spark-su
Warning: Ignoring non-spark config property: hadoop_server_path=/opt/huawei/Bigdata
17/08/22 10:18:47 WARN SparkConf: In Spark 1.0 and later spark.local.dir will be ov
in YARN).
17/08/22 10:18:48 WARN MetricsSystem: Using default name DAGScheduler for source be
17/08/22 10:18:48 WARN Utils: Service 'org.apache.spark.network.netty.NettyBlockTra

 Welcome to
    ____              __
   / __/__  ___ _____/ /__
  _\ \/ _ \/ _ `/ __/  '_/
 /___/ .__/\_,_/_/ /_/\_\   version  1.5.1
    /_/


 Spark context is available as sc, SQL context is available as sqlContext
>
```

使用SparkR与RStudio集成进行分析

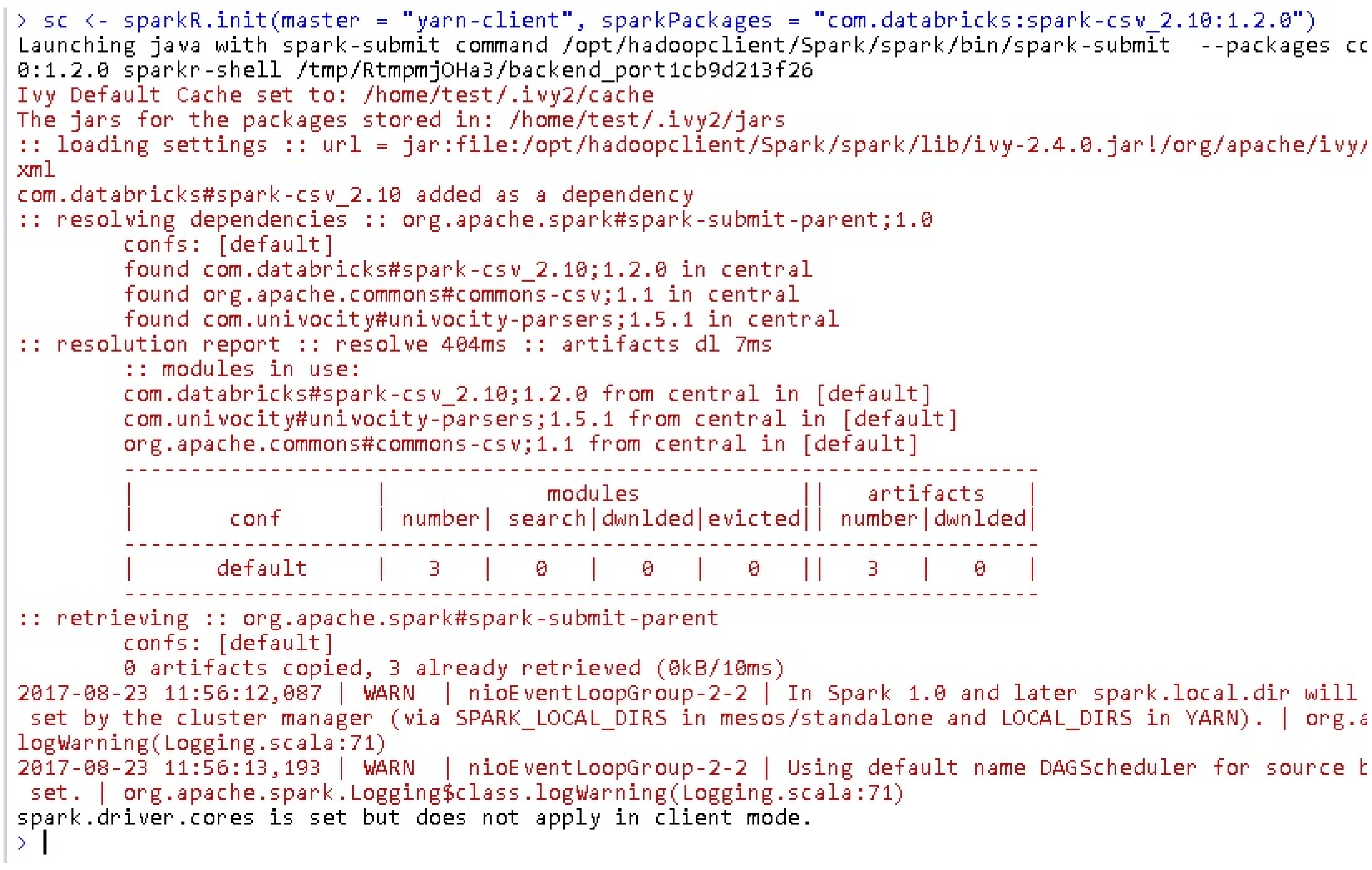
- 使用新建的用户登陆即可进入RStudio的Web开发界面
- 选择 **Tools** 菜单下的 **Shell** 进入登陆用户的shell进行kerberos认证

  ```
  cd /home/test/hadoopclient
  source bigdata_env
  kinit sparkuser
  ```

- 在RStudio界面中配置环境变量，初始化SparkR

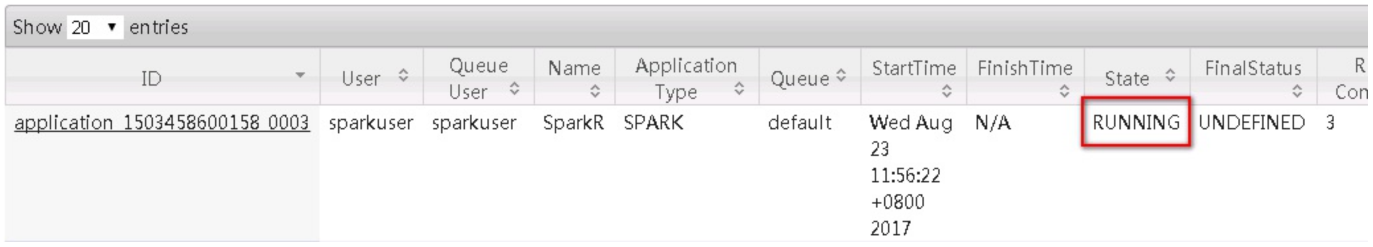
```
Sys.setenv("SPARKR_SUBMIT_ARGS"="--master yarn-client --num-executors 1 sparkr-shell")
Sys.setenv(SPARK_HOME="/home/test/hadoopclient/Spark/spark")
Sys.setenv(JAVA_HOME="/home/test/hadoopclient/JDK/jdk")
.libPaths(c(file.path(Sys.getenv("SPARK_HOME"), "R","lib"), .libPaths()))
library(SparkR)
sc <- sparkR.init(master = "yarn-client", sparkPackages = "com.databricks:spark-csv_2.10:1.2.0")
sqlContext <- sparkRSQL.init(sc)
```

- 初始化成功后如下图



```
> sc <- sparkR.init(master = "yarn-client", sparkPackages = "com.databricks:spark-csv_2.10:1.2.0")
Launching java with spark-submit command /opt/hadoopclient/Spark/spark/bin/spark-submit  --packages cc
0:1.2.0 sparkr-shell /tmp/RtmpmjOHa3/backend_port1cb9d213f26
Ivy Default Cache set to: /home/test/.ivy2/cache
The jars for the packages stored in: /home/test/.ivy2/jars
:: loading settings :: url = jar:file:/opt/hadoopclient/Spark/spark/lib/ivy-2.4.0.jar!/org/apache/ivy/
xml
com.databricks#spark-csv_2.10 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent;1.0
        confs: [default]
        found com.databricks#spark-csv_2.10;1.2.0 in central
        found org.apache.commons#commons-csv;1.1 in central
        found com.univocity#univocity-parsers;1.5.1 in central
:: resolution report :: resolve 404ms :: artifacts dl 7ms
        :: modules in use:
        com.databricks#spark-csv_2.10;1.2.0 from central in [default]
        com.univocity#univocity-parsers;1.5.1 from central in [default]
        org.apache.commons#commons-csv;1.1 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   3   |   0   |   0   |   0   ||   3   |   0   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent
        confs: [default]
        0 artifacts copied, 3 already retrieved (0kB/10ms)
2017-08-23 11:56:12,087 | WARN  | nioEventLoopGroup-2-2 | In Spark 1.0 and later spark.local.dir will
 set by the cluster manager (via SPARK_LOCAL_DIRS in mesos/standalone and LOCAL_DIRS in YARN). | org.a
logWarning(Logging.scala:71)
2017-08-23 11:56:13,193 | WARN  | nioEventLoopGroup-2-2 | Using default name DAGScheduler for source b
 set. | org.apache.spark.Logging$class.logWarning(Logging.scala:71)
spark.driver.cores is set but does not apply in client mode.
> |
```

在Yarn的ResourceManager界面可以看到sparkuser在集群启动了一个SparkR的应用



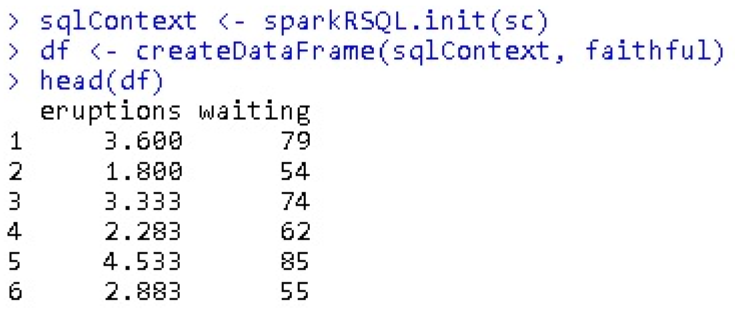
Show 20 ▾ entries

ID	User	Queue User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	R Con
application_1503458600158_0003	sparkuser	sparkuser	SparkR	SPARK	default	Wed Aug 23 11:56:22 +0800 2017	N/A	RUNNING	UNDEFINED	3

在RStudio中使用SparkR进行数据分析

- R DataFrame 转化为SparkR DataFrame

```
df <- createDataFrame(sqlContext, faithful)
head(df)
```



```
> sqlContext <- sparkRSQL.init(sc)
> df <- createDataFrame(sqlContext, faithful)
> head(df)
  eruptions waiting
1     3.600      79
2     1.800      54
3     3.333      74
4     2.283      62
5     4.533      85
6     2.883      55
```

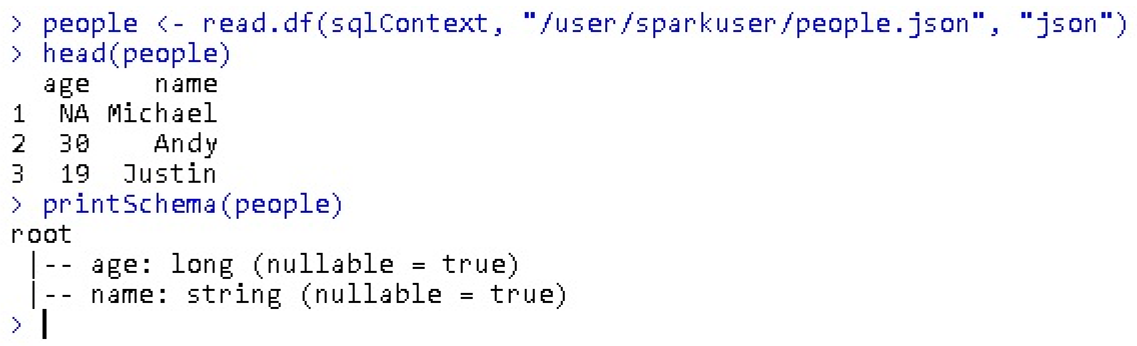
- 通过JSON文件加载数据进行分析处理

  将测试数据put到HDFS中

  ```
  wget https://raw.githubusercontent.com/eBay/Spark/master/examples/src/main/resources/people.json
  hdfs dfs -put people.json /user/sparkuser/
  ```

  执行文件加载分析

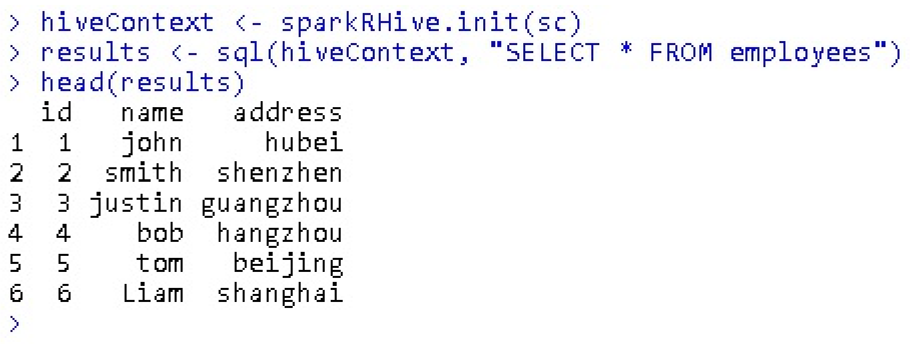
  ```
  people <- read.df(sqlContext, "/user/sparkuser/people.json", "json")
  head(people)
  printSchema(people)
  ```



```
> people <- read.df(sqlContext, "/user/sparkuser/people.json", "json")
> head(people)
  age    name
1  NA Michael
2  30    Andy
3  19  Justin
> printSchema(people)
root
 |-- age: long (nullable = true)
 |-- name: string (nullable = true)
> 
```

- 从Hive表中加载数据进行分析

  ```
  hiveContext <- sparkRHive.init(sc)
  results <- sql(hiveContext, "SELECT * FROM employees")
  head(results)
  ```

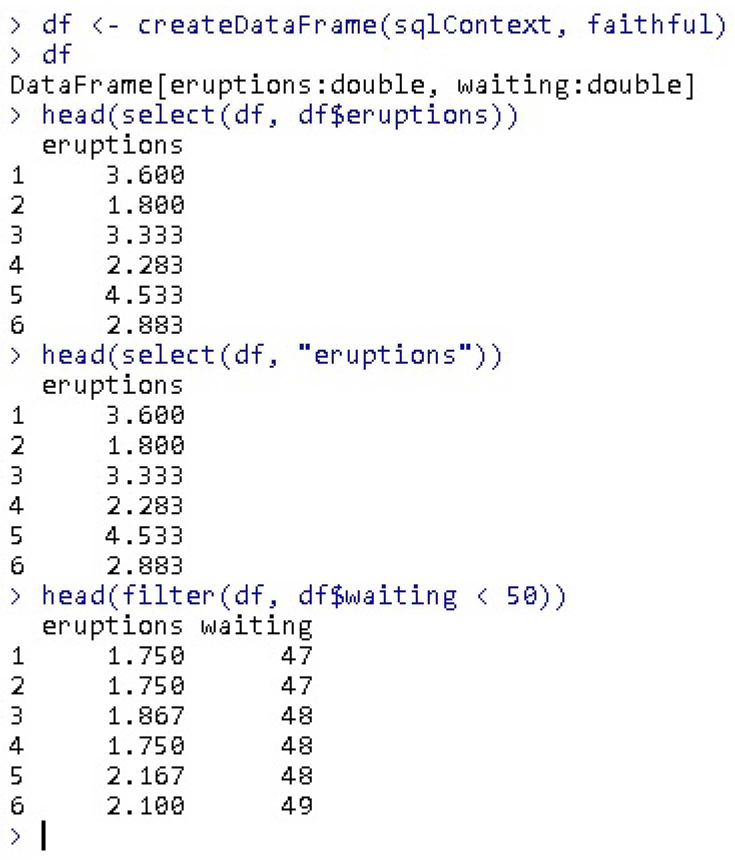


```
> hiveContext <- sparkRHive.init(sc)
> results <- sql(hiveContext, "SELECT * FROM employees")
> head(results)
  id   name   address
1  1   john     hubei
2  2  smith  shenzhen
3  3 justin guangzhou
4  4    bob  hangzhou
5  5    tom   beijing
6  6   Liam  shanghai
> 
```

- DataFrame Operations

  - Selecting rows, columns

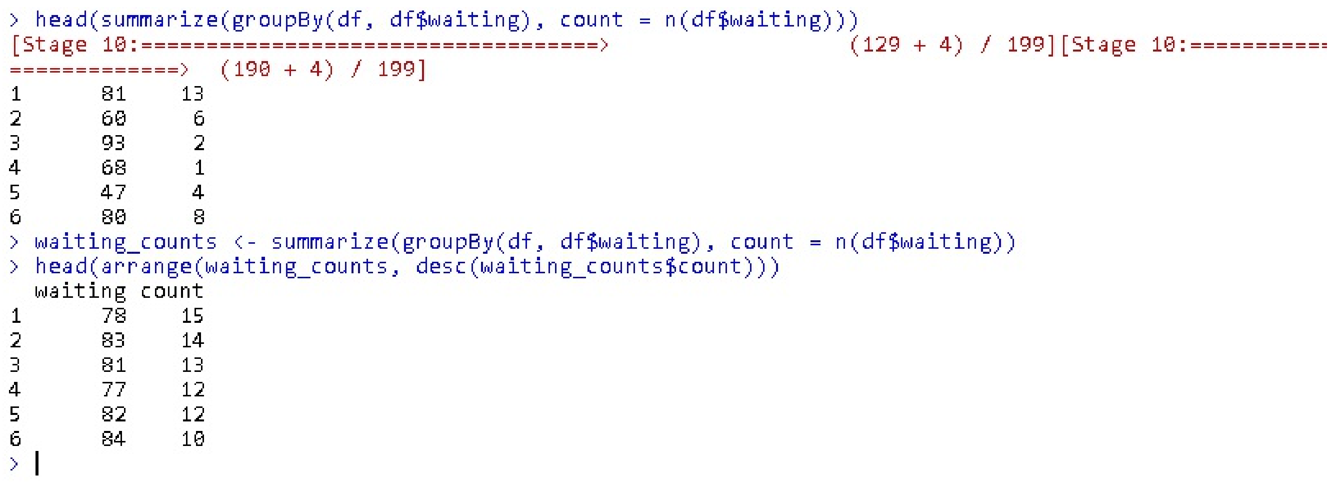
    ```
    df <- createDataFrame(sqlContext, faithful)
    df
    head(select(df, df$eruptions))
    head(select(df, "eruptions"))
    head(filter(df, df$waiting < 50))
    ```



```
> df <- createDataFrame(sqlContext, faithful)
> df
DataFrame[eruptions:double, waiting:double]
> head(select(df, df$eruptions))
  eruptions
1     3.600
2     1.800
3     3.333
4     2.283
5     4.533
6     2.883
> head(select(df, "eruptions"))
  eruptions
1     3.600
2     1.800
3     3.333
4     2.283
5     4.533
6     2.883
> head(filter(df, df$waiting < 50))
  eruptions waiting
1     1.750      47
2     1.750      47
3     1.867      48
4     1.750      48
5     2.167      48
6     2.100      49
> 
```

- Grouping, Aggregation

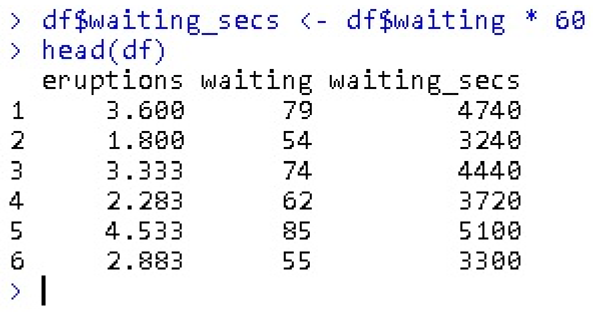
```
head(summarize(groupBy(df, df$waiting), count = n(df$waiting)))
waiting_counts <- summarize(groupBy(df, df$waiting), count = n(df$waiting))
head(arrange(waiting_counts, desc(waiting_counts$count)))
```



```
> head(summarize(groupBy(df, df$waiting), count = n(df$waiting)))
[Stage 10:===================================>                  (129 + 4) / 199][Stage 10:==========
=============>  (190 + 4) / 199]
1      81     13
2      60      6
3      93      2
4      68      1
5      47      4
6      80      8
> waiting_counts <- summarize(groupBy(df, df$waiting), count = n(df$waiting))
> head(arrange(waiting_counts, desc(waiting_counts$count)))
  waiting count
1      78    15
2      83    14
3      81    13
4      77    12
5      82    12
6      84    10
> 
```

- Operating on Columns

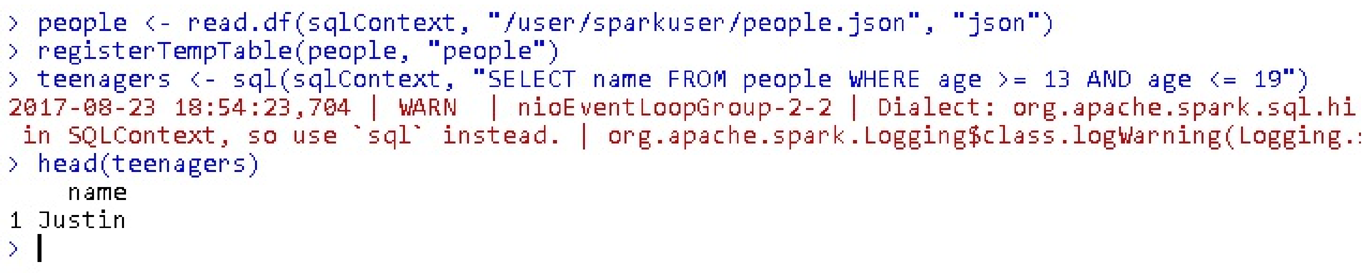
```
df$waiting_secs <- df$waiting * 60
head(df)
```



```
> df$waiting_secs <- df$waiting * 60
> head(df)
  eruptions waiting waiting_secs
1     3.600      79         4740
2     1.800      54         3240
3     3.333      74         4440
4     2.283      62         3720
5     4.533      85         5100
6     2.883      55         3300
> 
```

- Running SQL Queries from SparkR

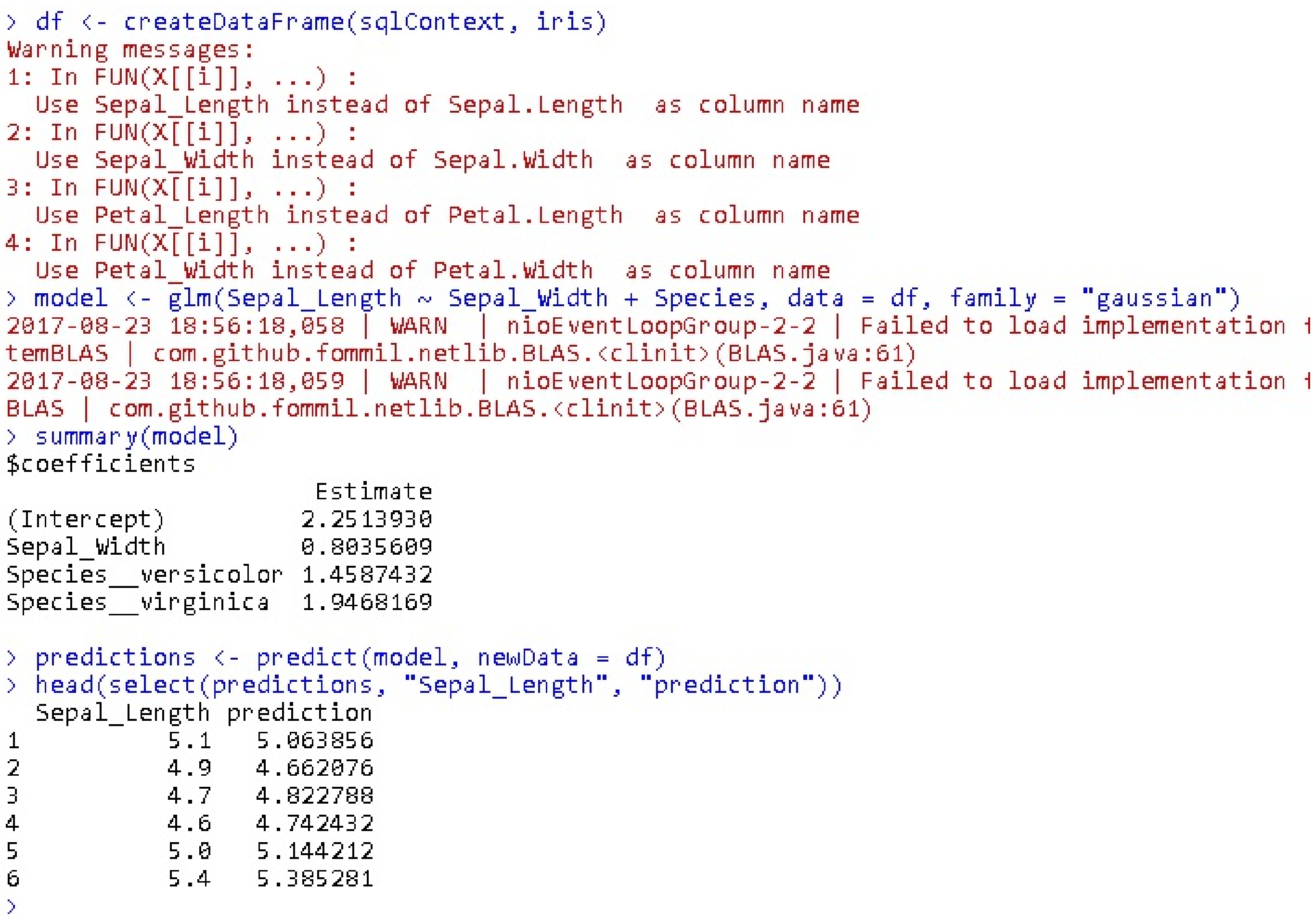
```
people <- read.df(sqlContext, "/user/sparkuser/people.json", "json")
registerTempTable(people, "people")
teenagers <- sql(sqlContext, "SELECT name FROM people WHERE age >= 13 AND age <= 19")
head(teenagers)
```



```
> people <- read.df(sqlContext, "/user/sparkuser/people.json", "json")
> registerTempTable(people, "people")
> teenagers <- sql(sqlContext, "SELECT name FROM people WHERE age >= 13 AND age <= 19")
2017-08-23 18:54:23,704 | WARN  | nioEventLoopGroup-2-2 | Dialect: org.apache.spark.sql.hi
 in SQLContext, so use `sql` instead. | org.apache.spark.Logging$class.logWarning(Logging.:
> head(teenagers)
    name
1 Justin
> 
```

- Machine Learning

```
df <- createDataFrame(sqlContext, iris)
model <- glm(Sepal_Length ~ Sepal_Width + Species, data = df, family = "gaussian")
summary(model)
predictions <- predict(model, newData = df)
head(select(predictions, "Sepal_Length", "prediction"))
```



```
> df <- createDataFrame(sqlContext, iris)
Warning messages:
1: In FUN(X[[i]], ...) :
  Use Sepal_Length instead of Sepal.Length  as column name
2: In FUN(X[[i]], ...) :
  Use Sepal_Width instead of Sepal.Width  as column name
3: In FUN(X[[i]], ...) :
  Use Petal_Length instead of Petal.Length  as column name
4: In FUN(X[[i]], ...) :
  Use Petal_Width instead of Petal.Width  as column name
> model <- glm(Sepal_Length ~ Sepal_Width + Species, data = df, family = "gaussian")
2017-08-23 18:56:18,058 | WARN  | nioEventLoopGroup-2-2 | Failed to load implementation f
temBLAS | com.github.fommil.netlib.BLAS.<clinit>(BLAS.java:61)
2017-08-23 18:56:18,059 | WARN  | nioEventLoopGroup-2-2 | Failed to load implementation f
BLAS | com.github.fommil.netlib.BLAS.<clinit>(BLAS.java:61)
> summary(model)
$coefficients
                     Estimate
(Intercept)         2.2513930
Sepal_Width         0.8035609
Species__versicolor 1.4587432
Species__virginica  1.9468169

> predictions <- predict(model, newData = df)
> head(select(predictions, "Sepal_Length", "prediction"))
  Sepal_Length prediction
1          5.1   5.063856
2          4.9   4.662076
3          4.7   4.822788
4          4.6   4.742432
5          5.0   5.144212
6          5.4   5.385281
> 
```

使用RStudio Sparklyr和Spark集成进行分析

- 选择 **Tools** 菜单下的 **Shell** 进入登陆用户的shell进行kerberos认证

```
cd /home/test/hadoopclient
source bigdata_env
kinit sparkuser
```

- 在RStudio中执行下面的命令，安装所需的library

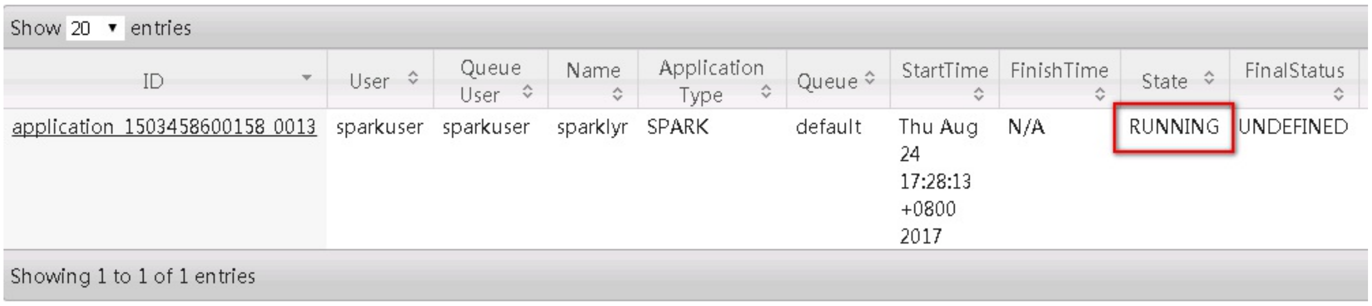
```
install.packages("sparklyr")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("babynames")
install.packages("dygraphs")
install.packages("rbokeh")
```

- 通过spark_connect连接spark集群

```
library(sparklyr)
library(dplyr)
library(ggplot2)
options(bitmapType = 'cairo')
Sys.setenv(JAVA_HOME="/home/test/hadoopclient/JDK/jdk")
Sys.setenv(SPARK_HOME="/home/test/hadoopclient/Spark2x/spark")
Sys.setenv(SPARK_HOME_VERSION="2.1.0")
sc <- spark_connect(master =  "yarn-client", version = "2.1.0", spark_home = "/home/test/hadoopclient/Spark2x/spark")
```

> 这里如果SPARK_HOME指向/home/test/hadoopclient/Spark/spark，同时设置version为1.6.1，则会对接上1.5.1的Spark
>
> sparklyr官方支持是1.6.1以上的Spark，这里强制指定version为1.6.1，主要功能均正常，部分Spark1.6.1支持而1.5.1不支持的特性执行会失败

启动成功后，在FusionInsgiht的Yarn的ResourceManager页面可以看到sparklyr的任务已经启动

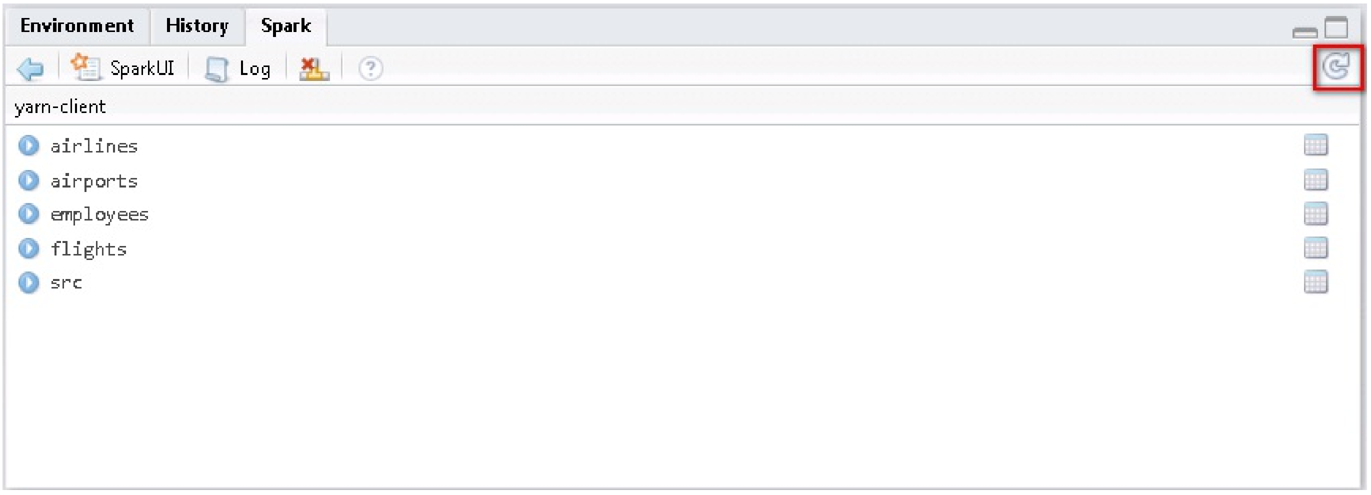


Show 20 entries

ID	User	Queue User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus
application_1503458600158_0013	sparkuser	sparkuser	sparklyr	SPARK	default	Thu Aug 24 17:28:13 +0800 2017	N/A	RUNNING	UNDEFINED

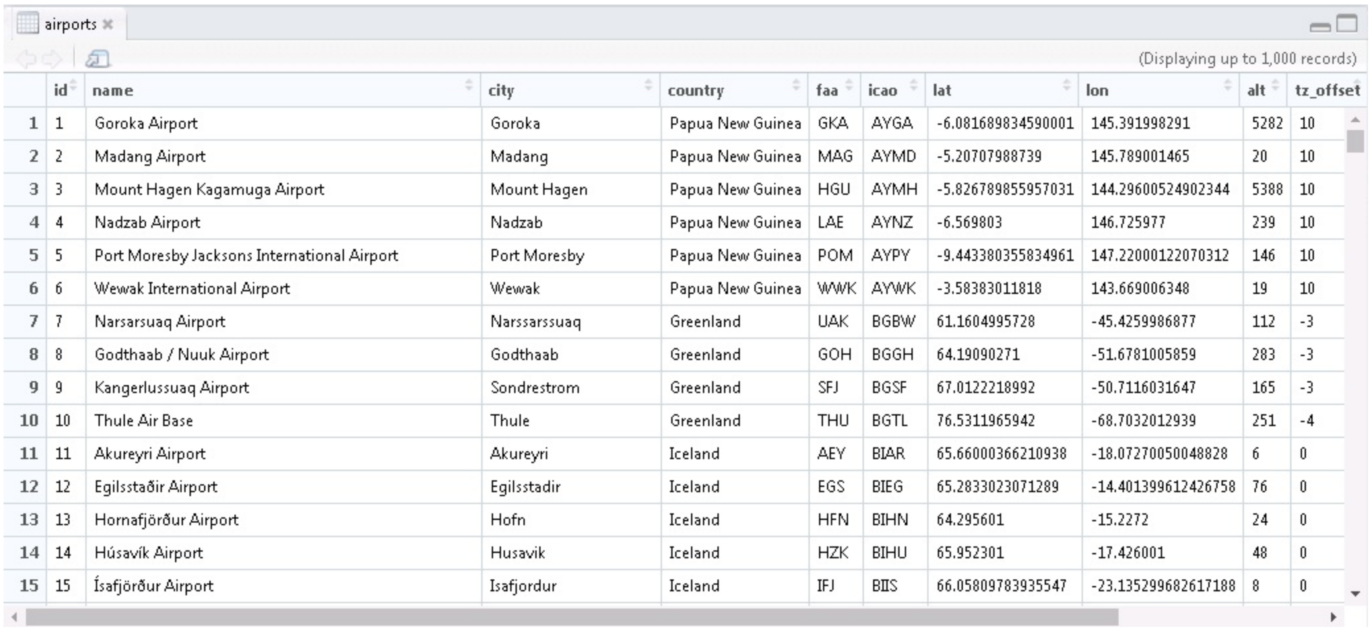
Showing 1 to 1 of 1 entries

在RStudio的Spark面板刷新一下，可以看到所有hive的表



```
airlines
airports
employees
flights
src
```

选择hive表右边的数据图表可以预览表中的数据


airports

(Displaying up to 1,000 records)

	id	name	city	country	faa	icao	lat	lon	alt	tz_offset
1	1	Goroka Airport	Goroka	Papua New Guinea	GKA	AYGA	-6.081689834590001	145.391998291	5282	10
2	2	Madang Airport	Madang	Papua New Guinea	MAG	AYMD	-5.20707988739	145.789001465	20	10
3	3	Mount Hagen Kagamuga Airport	Mount Hagen	Papua New Guinea	HGU	AYMH	-5.826789855957031	144.29600524902344	5388	10
4	4	Nadzab Airport	Nadzab	Papua New Guinea	LAE	AYNZ	-6.569803	146.725977	239	10
5	5	Port Moresby Jacksons International Airport	Port Moresby	Papua New Guinea	POM	AYPY	-9.443380355834961	147.22000122070312	146	10
6	6	Wewak International Airport	Wewak	Papua New Guinea	WWK	AYWK	-3.58383011818	143.669006348	19	10
7	7	Narsarsuaq Airport	Narssarssuaq	Greenland	UAK	BGBW	61.1604995728	-45.4259986877	112	-3
8	8	Godthaab / Nuuk Airport	Godthaab	Greenland	GOH	BGGH	64.19090271	-51.6781005859	283	-3
9	9	Kangerlussuaq Airport	Sondrestrom	Greenland	SFJ	BGSF	67.0122218992	-50.7116031647	165	-3
10	10	Thule Air Base	Thule	Greenland	THU	BGTL	76.5311965942	-68.7032012939	251	-4
11	11	Akureyri Airport	Akureyri	Iceland	AEY	BIAR	65.66000366210938	-18.07270050048828	6	0
12	12	Egilsstaðir Airport	Egilsstadir	Iceland	EGS	BIEG	65.2833023071289	-14.401399612426758	76	0
13	13	Hornafjörður Airport	Hofn	Iceland	HFN	BIHN	64.295601	-15.2272	24	0
14	14	Húsavík Airport	Husavik	Iceland	HZK	BIHU	65.952301	-17.426001	48	0
15	15	Ísafjörður Airport	Isafjordur	Iceland	IFJ	BIIS	66.05809783935547	-23.135299682617188	8	0

使用sparklyr结合spark进行数据分析babynames数据集

Use dplyr syntax to write Apache Spark SQL queries. Use select, where, group by, joins, and window functions in Aparche Spark SQL.

Setup

```
library(sparklyr)
library(dplyr)
library(babynames)
library(ggplot2)
library(dygraphs)
library(rbokeh)
knitr::opts_chunk$set(message = FALSE, warning = FALSE)
```

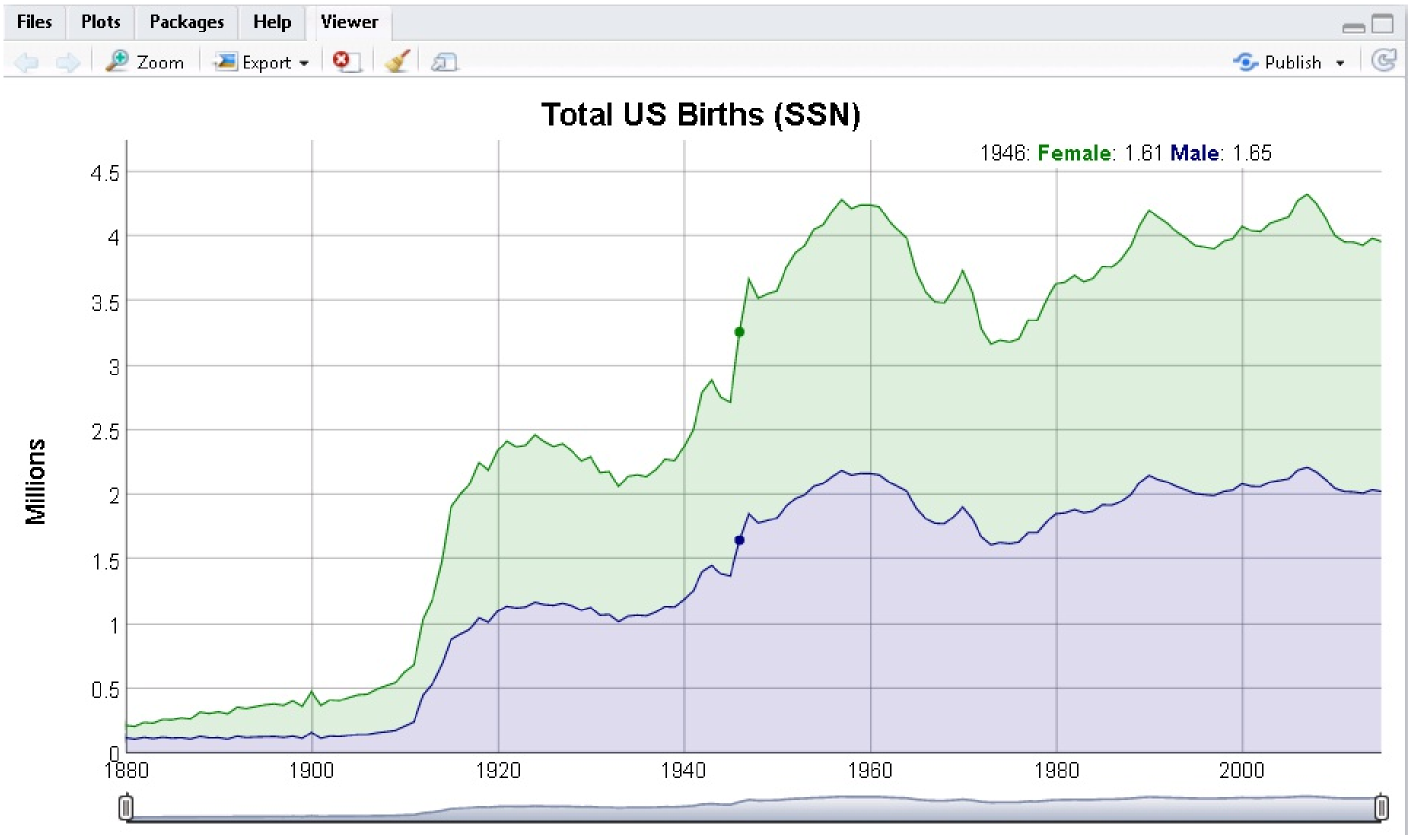
Connect to Spark

```
options(bitmapType = 'cairo')
Sys.setenv(JAVA_HOME="/home/test/hadoopclient/JDK/jdk")
Sys.setenv(SPARK_HOME="/home/test/hadoopclient/Spark/spark")
Sys.setenv(SPARK_HOME_VERSION="1.6.1")
sc <- spark_connect(master =  "yarn-client", version = "1.6.1", spark_home = "/home/test/hadoopclient/Spark/spark")
```

Total US births

Plot total US births recorded from the Social Security Administration.

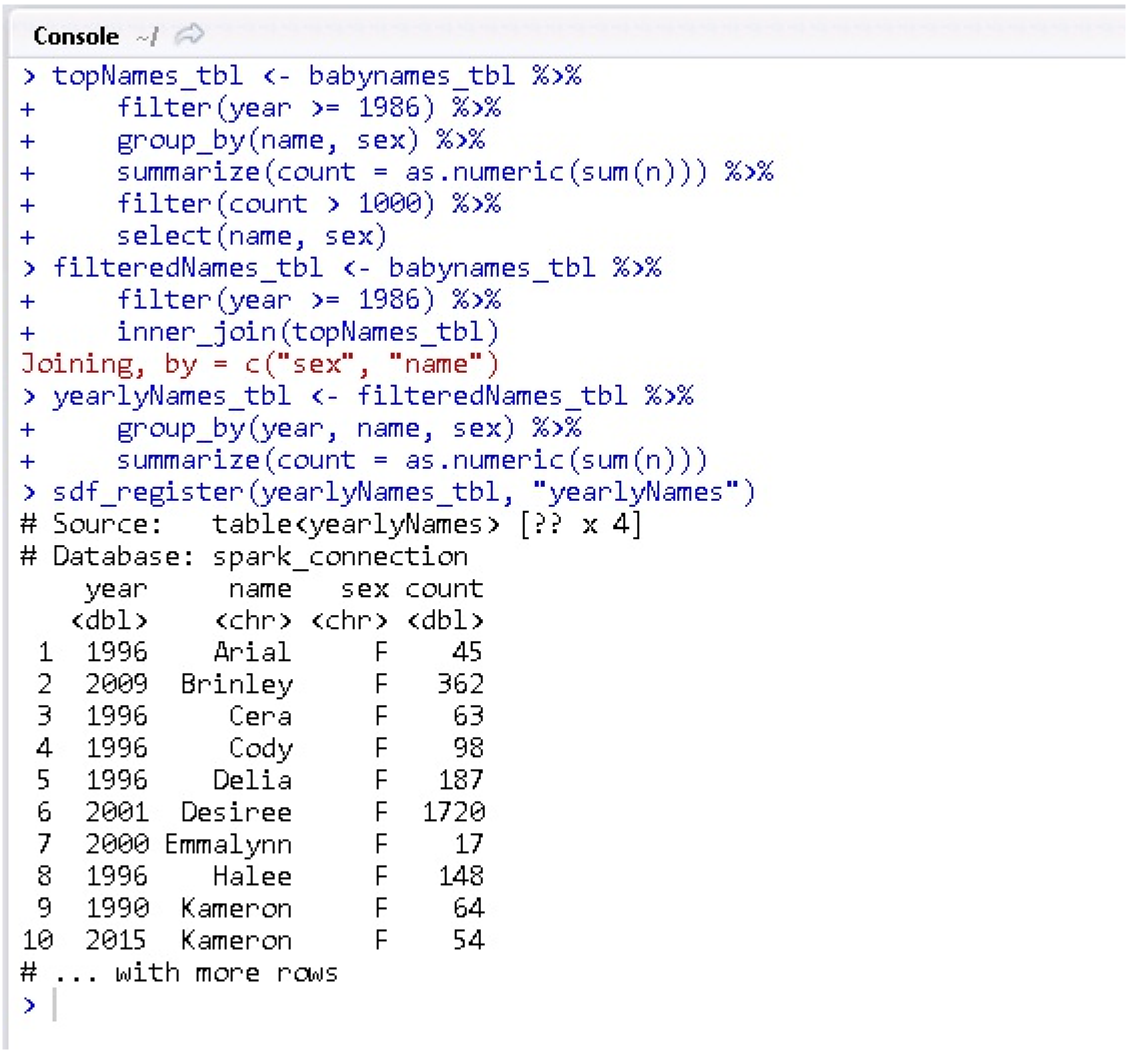
```
birthsYearly <- applicants_tbl %>%
  mutate(male = ifelse(sex == "M", n_all, 0), female = ifelse(sex == "F", n_all, 0)) %>%
  group_by(year) %>%
  summarize(Male = sum(male) / 1000000, Female = sum(female) / 1000000) %>%
  arrange(year) %>%
  collect
birthsYearly %>%
  dygraph(main = "Total US Births (SSN)", ylab = "Millions") %>%
  dySeries("Female") %>%
  dySeries("Male") %>%
  dyOptions(stackedGraph = TRUE) %>%
  dyRangeSelector(height = 20)
```



Aggregate data by name

Use Spark SQL to create a look up table. Register and cache the look up table in Spark for future queries.

```
topNames_tbl <- babynames_tbl %>%
  filter(year >= 1986) %>%
  group_by(name, sex) %>%
  summarize(count = as.numeric(sum(n))) %>%
  filter(count > 1000) %>%
  select(name, sex)
filteredNames_tbl <- babynames_tbl %>%
  filter(year >= 1986) %>%
  inner_join(topNames_tbl)
yearlyNames_tbl <- filteredNames_tbl %>%
  group_by(year, name, sex) %>%
  summarize(count = as.numeric(sum(n)))
sdf_register(yearlyNames_tbl, "yearlyNames")
```



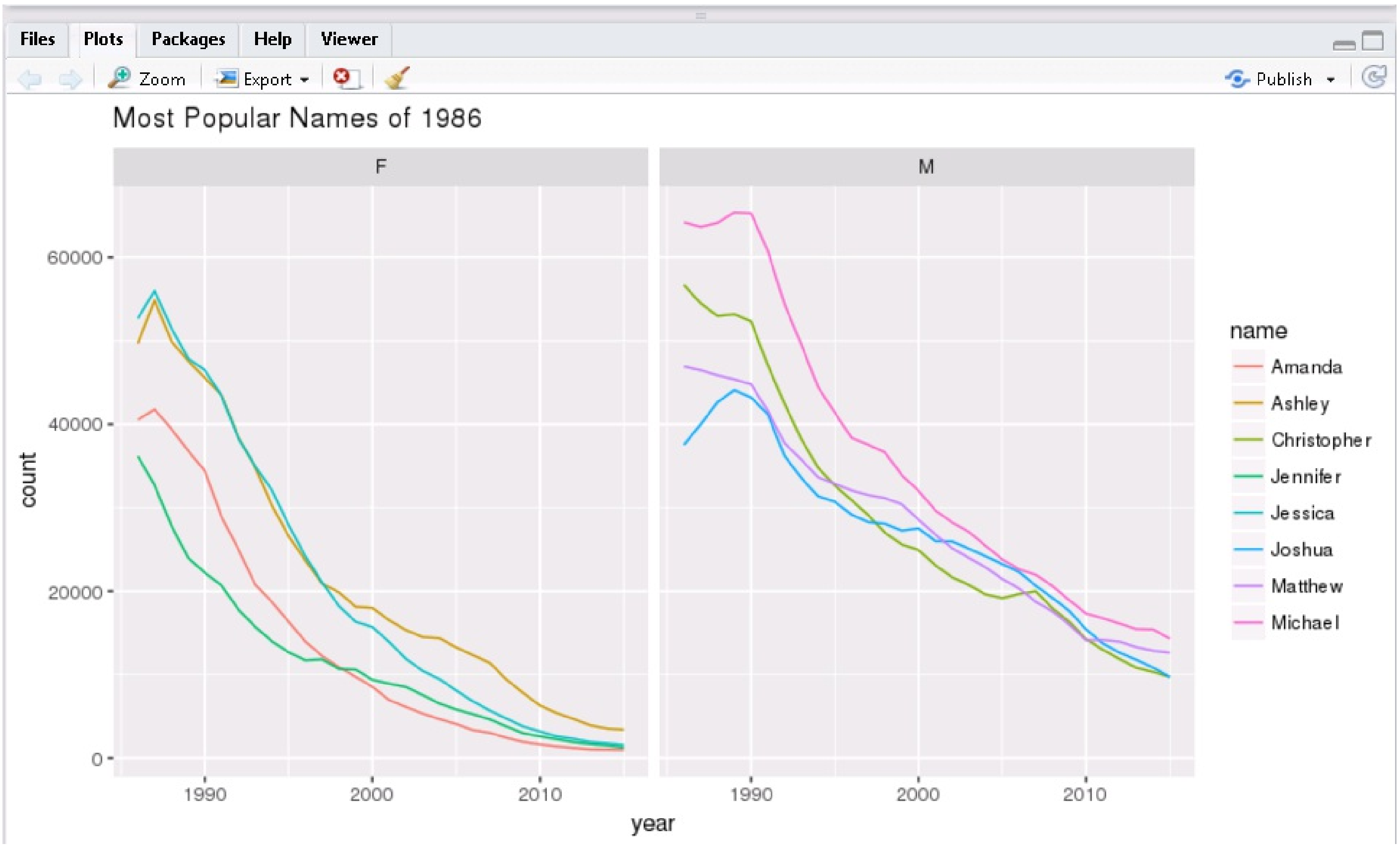
```
> topNames_tbl <- babynames_tbl %>%
+     filter(year >= 1986) %>%
+     group_by(name, sex) %>%
+     summarize(count = as.numeric(sum(n))) %>%
+     filter(count > 1000) %>%
+     select(name, sex)
> filteredNames_tbl <- babynames_tbl %>%
+     filter(year >= 1986) %>%
+     inner_join(topNames_tbl)
Joining, by = c("sex", "name")
> yearlyNames_tbl <- filteredNames_tbl %>%
+     group_by(year, name, sex) %>%
+     summarize(count = as.numeric(sum(n)))
> sdf_register(yearlyNames_tbl, "yearlyNames")
# Source:   table<yearlyNames> [?? x 4]
# Database: spark_connection
    year     name   sex count
   <dbl>    <chr> <chr> <dbl>
 1  1996    Arial     F    45
 2  2009  Brinley     F   362
 3  1996     Cera     F    63
 4  1996     Cody     F    98
 5  1996    Delia     F   187
 6  2001  Desiree     F  1720
 7  2000 Emmalynn     F    17
 8  1996    Halee     F   148
 9  1990  Kameron     F    64
10  2015  Kameron     F    54
# ... with more rows
>
```

```
tbl_cache(sc, "yearlyNames")
```

Most popular names (1986)

Identify the top 5 male and female names from 1986. Visualize the popularity trend over time.

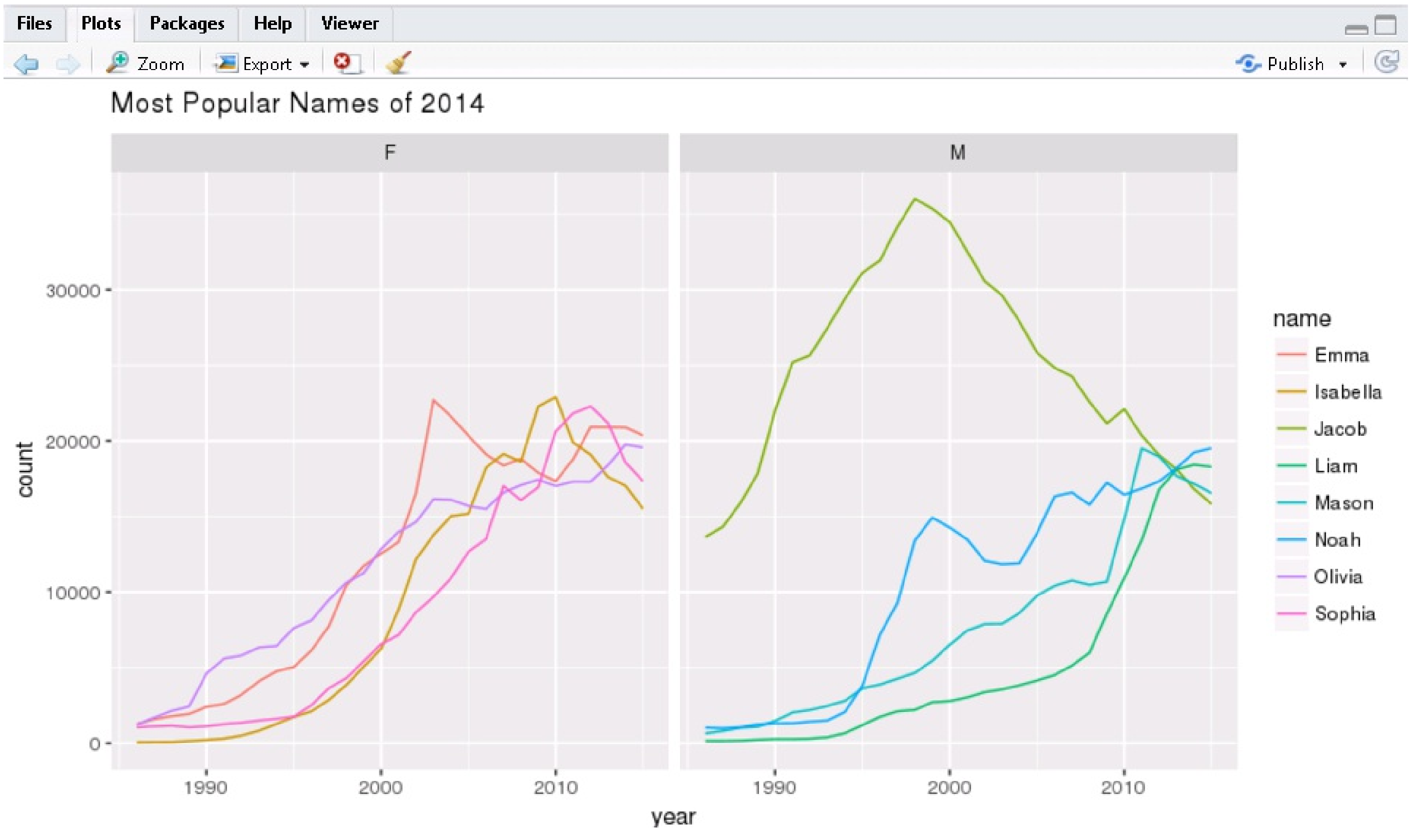
```
topNames1986_tbl <- yearlyNames_tbl %>%
  filter(year == 1986) %>%
  group_by(name, sex) %>%
  summarize(count = sum(count)) %>%
  group_by(sex) %>%
  mutate(rank = min_rank(desc(count))) %>%
  filter(rank < 5) %>%
  arrange(sex, rank) %>%
  select(name, sex, rank) %>%
  sdf_register("topNames1986")
tbl_cache(sc, "topNames1986")
topNames1986Yearly <- yearlyNames_tbl %>%
  inner_join(select(topNames1986_tbl, sex, name)) %>%
  collect
ggplot(topNames1986Yearly, aes(year, count, color=name)) +
  facet_grid(~sex) +
  geom_line() +
  ggtitle("Most Popular Names of 1986")
```



Most popular names (2014)

Identify the top 5 male and female names from 2014. Visualize the popularity trend over time.

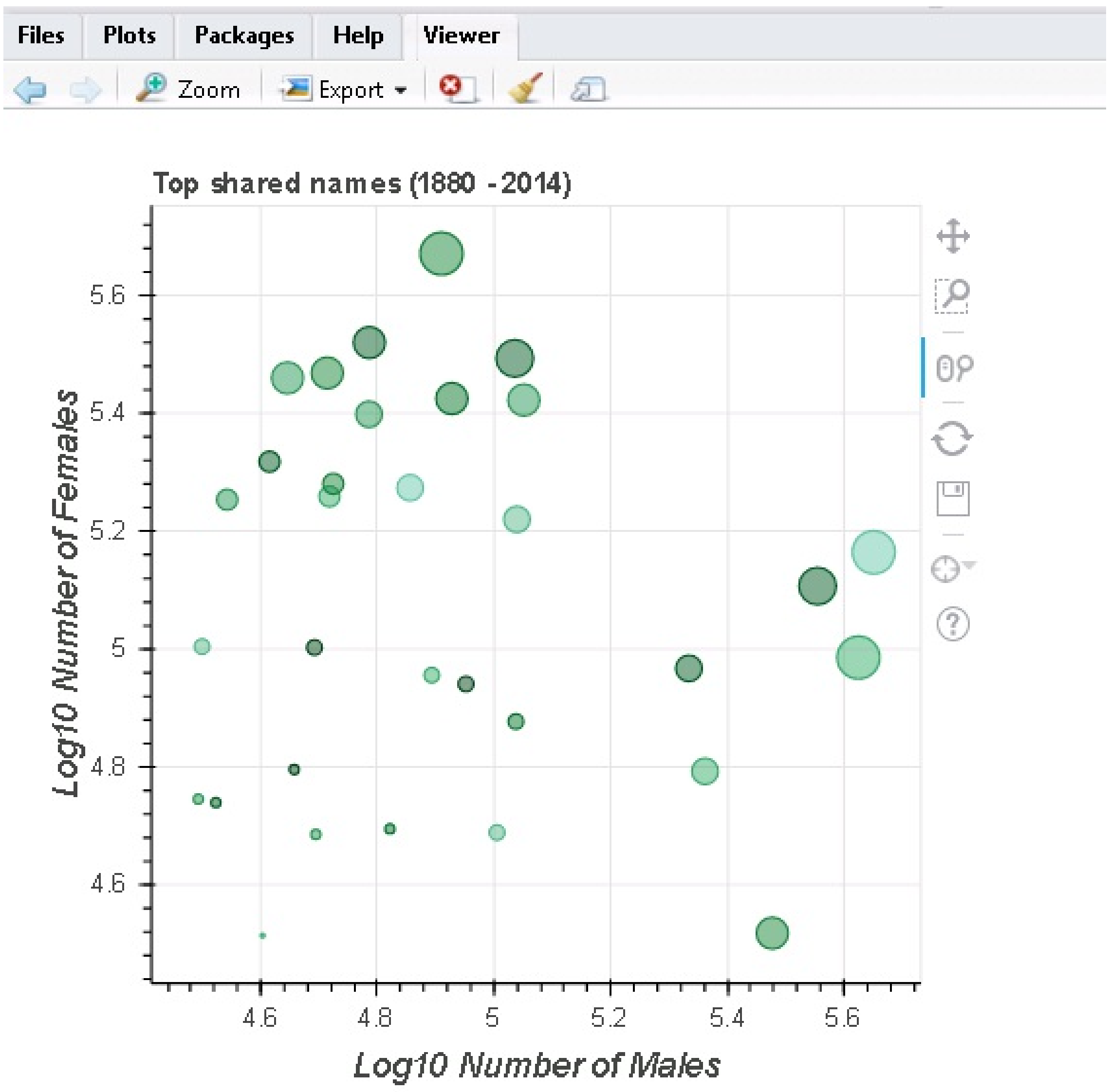
```
topNames2014_tbl <- yearlyNames_tbl %>%
  filter(year == 2014) %>%
  group_by(name, sex) %>%
  summarize(count = sum(count)) %>%
  group_by(sex) %>%
  mutate(rank = min_rank(desc(count))) %>%
  filter(rank < 5) %>%
  arrange(sex, rank) %>%
  select(name, sex, rank) %>%
  sdf_register("topNames2014")
tbl_cache(sc, "topNames2014")
topNames2014Yearly <- yearlyNames_tbl %>%
  inner_join(select(topNames2014_tbl, sex, name)) %>%
  collect
ggplot(topNames2014Yearly, aes(year, count, color=name)) +
  facet_grid(~sex) +
  geom_line() +
  ggtitle("Most Popular Names of 2014")
```



Shared names

Visualize the most popular names that are shared by both males and females.

```
sharedName <- babynames_tbl %>%
  mutate(male = ifelse(sex == "M", n, 0), female = ifelse(sex == "F", n, 0)) %>%
  group_by(name) %>%
  summarize(Male = as.numeric(sum(male)),
            Female = as.numeric(sum(female)),
            count = as.numeric(sum(n)),
            AvgYear = round(as.numeric(sum(year * n) / sum(n)),0)) %>%
  filter(Male > 30000 & Female > 30000) %>%
  collect
figure(width = NULL, height = NULL,
       xlab = "Log10 Number of Males",
       ylab = "Log10 Number of Females",
       title = "Top shared names (1880 - 2014)") %>%
  ly_points(log10(Male), log10(Female), data = sharedName,
            color = AvgYear, size = scale(sqrt(count)),
            hover = list(name, Male, Female, AvgYear), legend = FALSE)
```



使用sparklyr结合spark进行数据分析航空公司飞行数据(必须配套Spark2x)

> **Train a linear model** step will failed in Spark 1.5.1, because Spark 1.5.1 does not support the coefficients method for linear model output

Is there evidence to suggest that some airline carriers make up time in flight? This analysis predicts time gained in flight by airline carrier.

Connect to spark2x

```
library(sparklyr)
library(dplyr)
library(ggplot2)
options(bitmapType = 'cairo')
Sys.setenv(JAVA_HOME="/home/test/hadoopclient/JDK/jdk")
Sys.setenv(SPARK_HOME="/home/test/hadoopclient/Spark2x/spark")
Sys.setenv(SPARK_HOME_VERSION="2.1.0")
sc <- spark_connect(master =  "yarn-client", version = "2.1.0", spark_home = "/home/test/hadoopclient/Spark2x/spark")
```

Cache the tables into memory

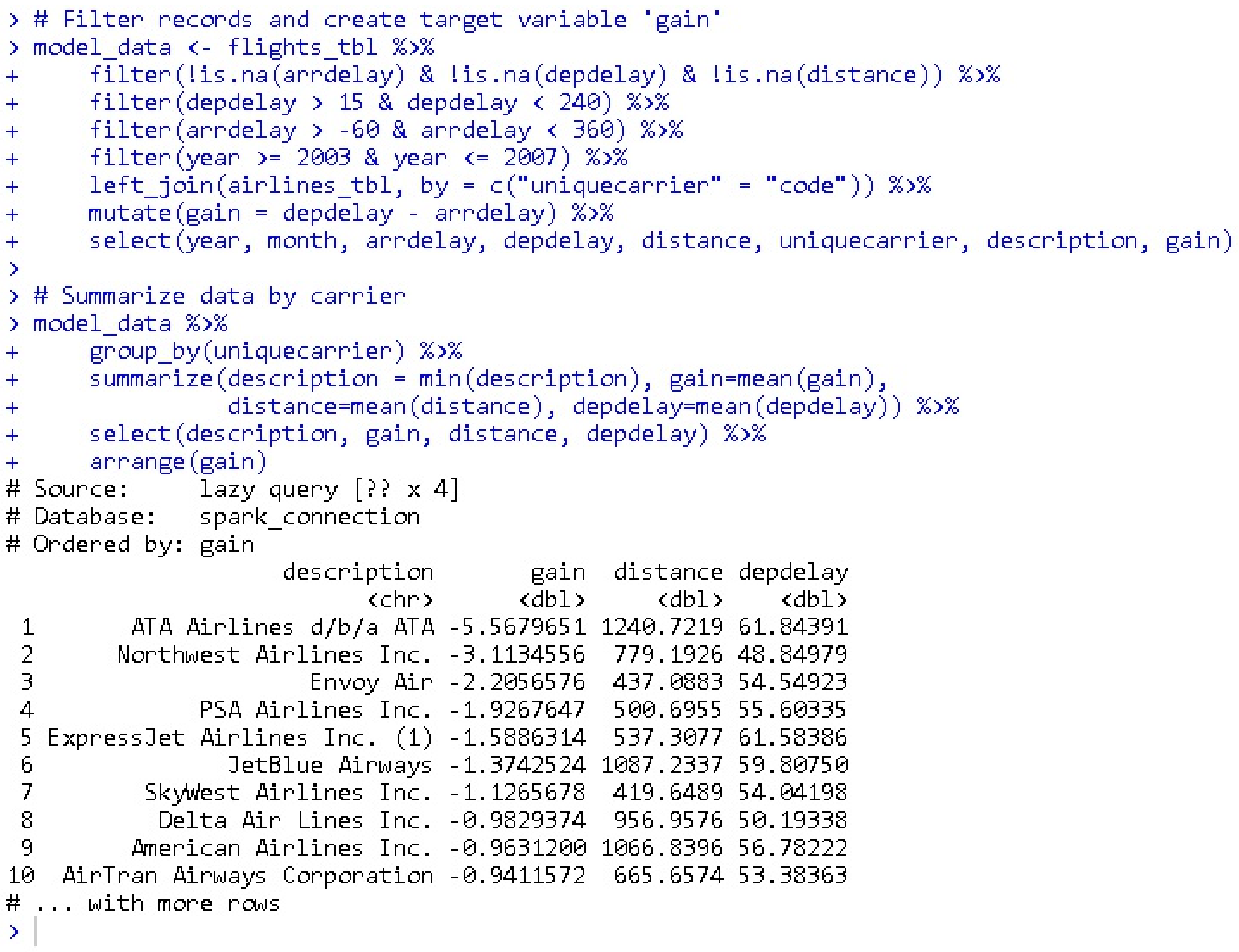
Use `tbl_cache` to load the flights table into memory. Caching tables will make analysis much faster. Create a dplyr reference to the Spark DataFrame.

```
# Cache flights Hive table into Spark
tbl_cache(sc, 'flights')
flights_tbl <- tbl(sc, 'flights')

# Cache airlines Hive table into Spark
tbl_cache(sc, 'airlines')
airlines_tbl <- tbl(sc, 'airlines')

# Cache airports Hive table into Spark
tbl_cache(sc, 'airports')
airports_tbl <- tbl(sc, 'airports')
```

Create a model data set

Filter the data to contain only the records to be used in the fitted model. Join carrier descriptions for reference. Create a new variable called gain which represents the amount of time gained (or lost) in flight.

```
# Filter records and create target variable 'gain'
model_data <- flights_tbl %>%
filter(!is.na(arrdelay) & !is.na(depdelay) & !is.na(distance)) %>%
filter(depdelay > 15 & depdelay < 240) %>%
filter(arrdelay > -60 & arrdelay < 360) %>%
filter(year >= 2003 & year <= 2007) %>%
left_join(airlines_tbl, by = c("uniquecarrier" = "code")) %>%
mutate(gain = depdelay - arrdelay) %>%
select(year, month, arrdelay, depdelay, distance, uniquecarrier, description, gain)

# Summarize data by carrier
model_data %>%
group_by(uniquecarrier) %>%
summarize(description = min(description), gain=mean(gain),
          distance=mean(distance), depdelay=mean(depdelay)) %>%
select(description, gain, distance, depdelay) %>%
arrange(gain)
```



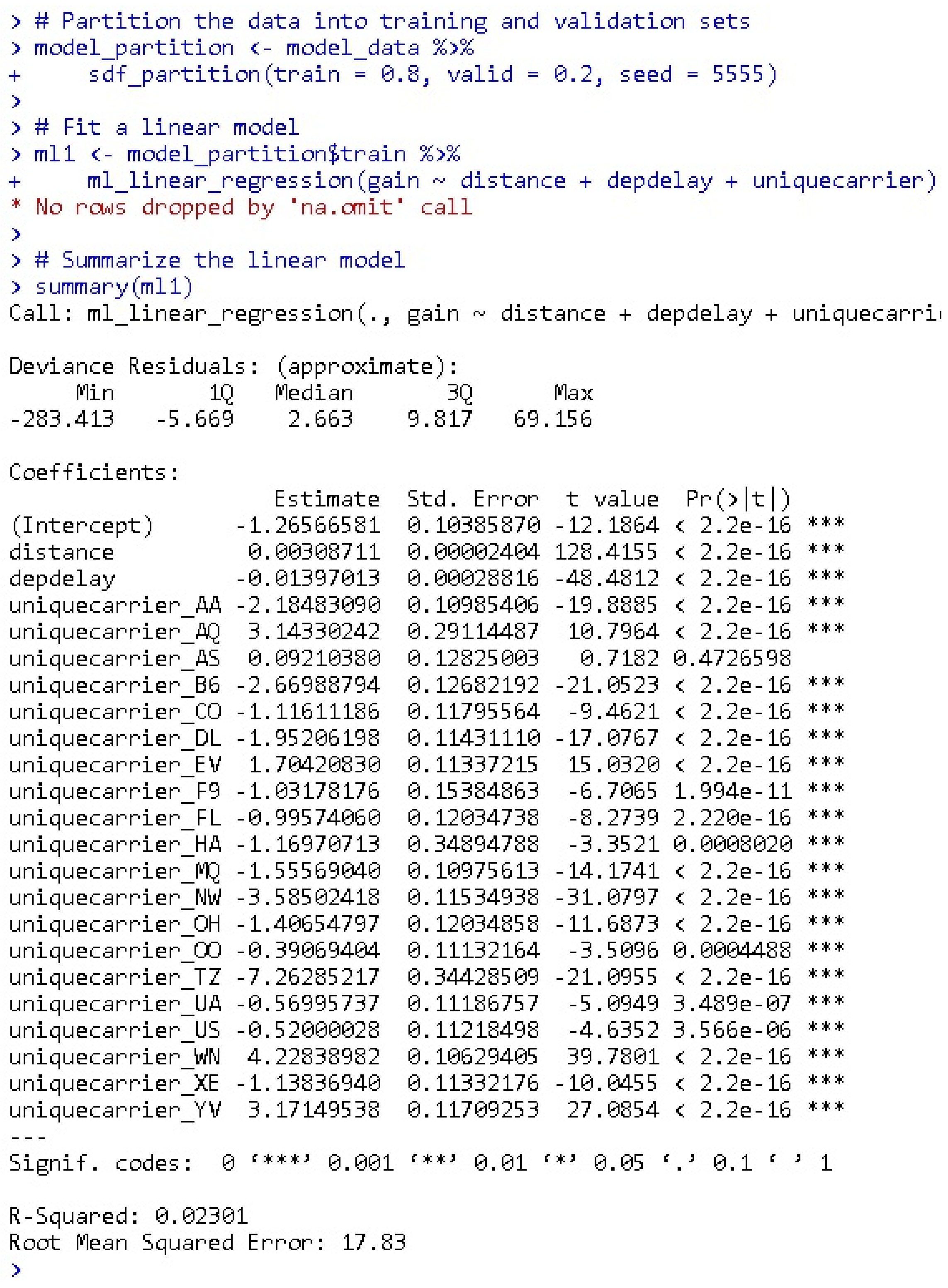
```
> # Filter records and create target variable 'gain'
> model_data <- flights_tbl %>%
+     filter(!is.na(arrdelay) & !is.na(depdelay) & !is.na(distance)) %>%
+     filter(depdelay > 15 & depdelay < 240) %>%
+     filter(arrdelay > -60 & arrdelay < 360) %>%
+     filter(year >= 2003 & year <= 2007) %>%
+     left_join(airlines_tbl, by = c("uniquecarrier" = "code")) %>%
+     mutate(gain = depdelay - arrdelay) %>%
+     select(year, month, arrdelay, depdelay, distance, uniquecarrier, description, gain)
>
> # Summarize data by carrier
> model_data %>%
+     group_by(uniquecarrier) %>%
+     summarize(description = min(description), gain=mean(gain),
+               distance=mean(distance), depdelay=mean(depdelay)) %>%
+     select(description, gain, distance, depdelay) %>%
+     arrange(gain)
# Source:     lazy query [?? x 4]
# Database:   spark_connection
# Ordered by: gain
                  description       gain  distance depdelay
                        <chr>      <dbl>     <dbl>    <dbl>
 1      ATA Airlines d/b/a ATA -5.5679651 1240.7219 61.84391
 2     Northwest Airlines Inc. -3.1134556  779.1926 48.84979
 3                   Envoy Air -2.2056576  437.0883 54.54923
 4           PSA Airlines Inc. -1.9267647  500.6955 55.60335
 5 ExpressJet Airlines Inc. (1) -1.5886314  537.3077 61.58386
 6             JetBlue Airways -1.3742524 1087.2337 59.80750
 7       SkyWest Airlines Inc. -1.1265678  419.6489 54.04198
 8        Delta Air Lines Inc. -0.9829374  956.9576 50.19338
 9      American Airlines Inc. -0.9631200 1066.8396 56.78222
10  AirTran Airways Corporation -0.9411572  665.6574 53.38363
# ... with more rows
>
```

Train a linear model

Predict time gained or lost in flight as a function of distance, departure delay, and airline carrier.

```
# Partition the data into training and validation sets
model_partition <- model_data %>%
sdf_partition(train = 0.8, valid = 0.2, seed = 5555)

# Fit a linear model
ml1 <- model_partition$train %>%
ml_linear_regression(gain ~ distance + depdelay + uniquecarrier)

# Summarize the linear model
```

```
summary(ml1)
```

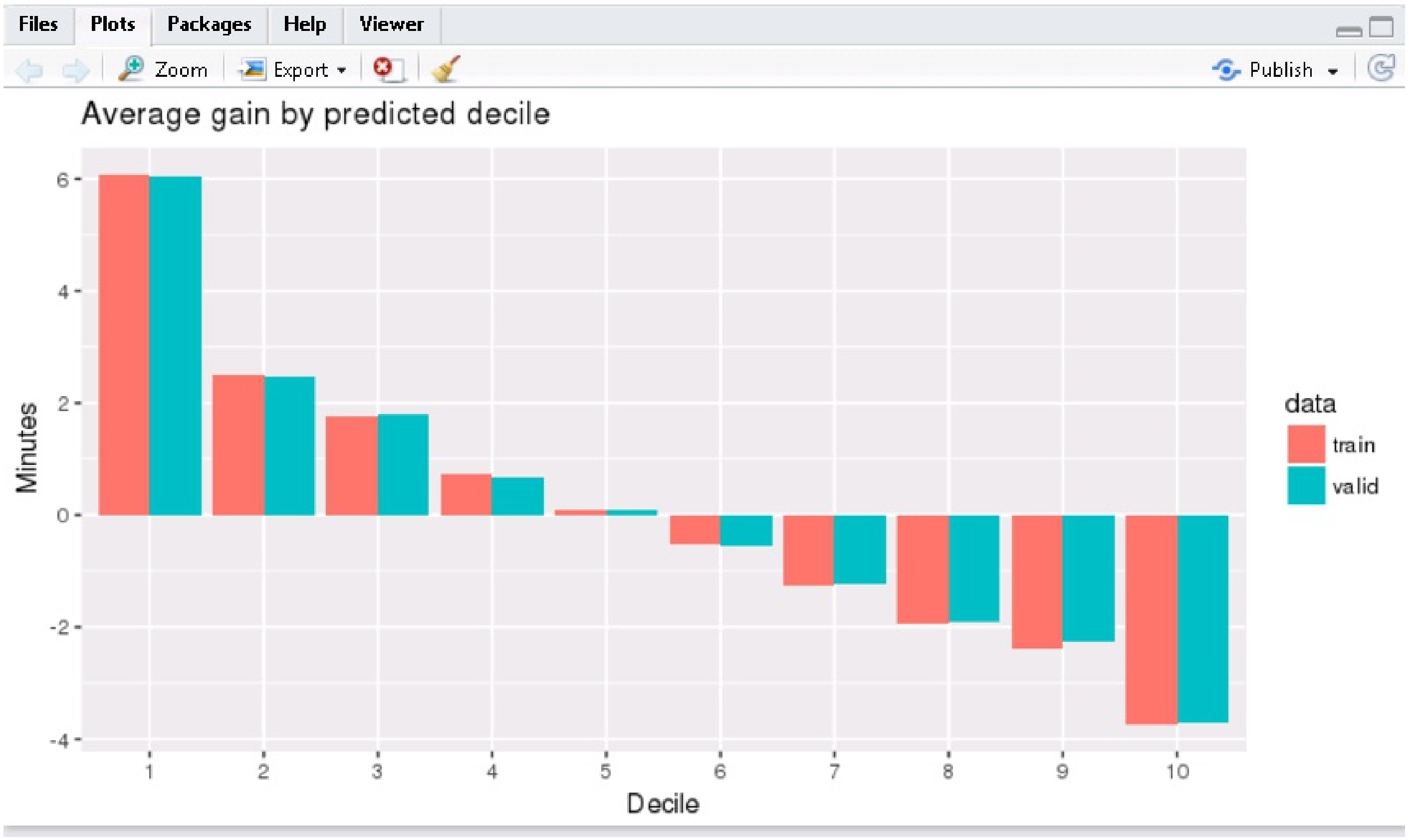


```
> # Partition the data into training and validation sets
> model_partition <- model_data %>%
+     sdf_partition(train = 0.8, valid = 0.2, seed = 5555)
>
> # Fit a linear model
> ml1 <- model_partition$train %>%
+     ml_linear_regression(gain ~ distance + depdelay + uniquecarrier)
* No rows dropped by 'na.omit' call
>
> # Summarize the linear model
> summary(ml1)
Call: ml_linear_regression(., gain ~ distance + depdelay + uniquecarri

Deviance Residuals: (approximate):
     Min       1Q   Median       3Q      Max
-283.413   -5.669    2.663    9.817   69.156

Coefficients:
                    Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)      -1.26566581  0.10385870 -12.1864 < 2.2e-16 ***
distance          0.00308711  0.00002404 128.4155 < 2.2e-16 ***
depdelay         -0.01397013  0.00028816 -48.4812 < 2.2e-16 ***
uniquecarrier_AA -2.18483090  0.10985406 -19.8885 < 2.2e-16 ***
uniquecarrier_AQ  3.14330242  0.29114487  10.7964 < 2.2e-16 ***
uniquecarrier_AS  0.09210380  0.12825003   0.7182 0.4726598
uniquecarrier_B6 -2.66988794  0.12682192 -21.0523 < 2.2e-16 ***
uniquecarrier_CO -1.11611186  0.11795564  -9.4621 < 2.2e-16 ***
uniquecarrier_DL -1.95206198  0.11431110 -17.0767 < 2.2e-16 ***
uniquecarrier_EV  1.70420830  0.11337215  15.0320 < 2.2e-16 ***
uniquecarrier_F9 -1.03178176  0.15384863  -6.7065 1.994e-11 ***
uniquecarrier_FL -0.99574060  0.12034738  -8.2739 2.220e-16 ***
uniquecarrier_HA -1.16970713  0.34894788  -3.3521 0.0008020 ***
uniquecarrier_MQ -1.55569040  0.10975613 -14.1741 < 2.2e-16 ***
uniquecarrier_NW -3.58502418  0.11534938 -31.0797 < 2.2e-16 ***
uniquecarrier_OH -1.40654797  0.12034858 -11.6873 < 2.2e-16 ***
uniquecarrier_OO -0.39069404  0.11132164  -3.5096 0.0004488 ***
uniquecarrier_TZ -7.26285217  0.34428509 -21.0955 < 2.2e-16 ***
uniquecarrier_UA -0.56995737  0.11186757  -5.0949 3.489e-07 ***
uniquecarrier_US -0.52000028  0.11218498  -4.6352 3.566e-06 ***
uniquecarrier_WN  4.22838982  0.10629405  39.7801 < 2.2e-16 ***
uniquecarrier_XE -1.13836940  0.11332176 -10.0455 < 2.2e-16 ***
uniquecarrier_YV  3.17149538  0.11709253  27.0854 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-Squared: 0.02301
Root Mean Squared Error: 17.83
>
```

** Assess model performance**

Compare the model performance using the validation data.

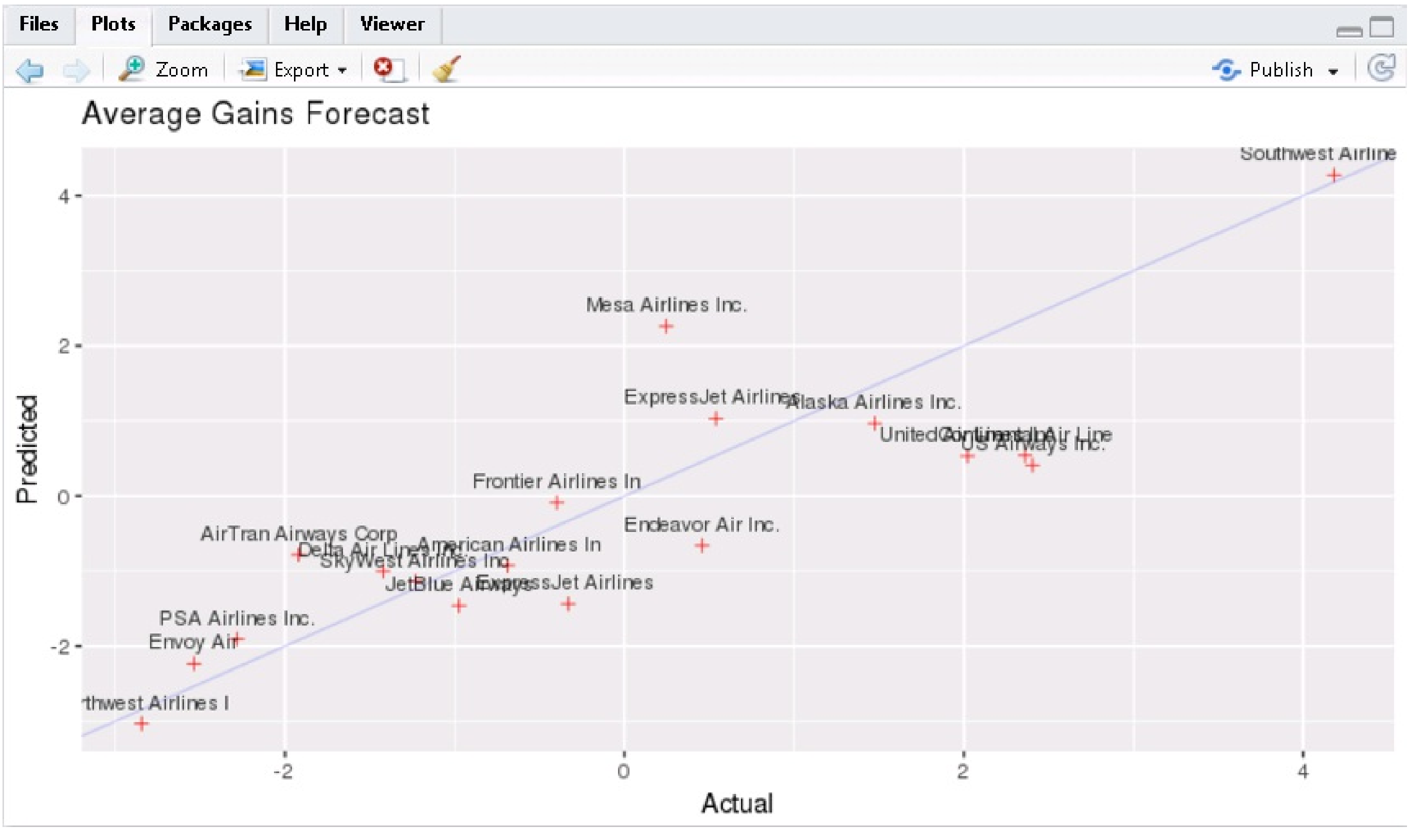
```
# Calculate average gains by predicted decile
model_deciles <- lapply(model_partition, function(x) {
  sdf_predict(ml1, x) %>%
    mutate(decile = ntile(desc(prediction), 10)) %>%
    group_by(decile) %>%
    summarize(gain = mean(gain)) %>%
    select(decile, gain) %>%
    collect()
})

# Create a summary dataset for plotting
deciles <- rbind(
  data.frame(data = 'train', model_deciles$train),
  data.frame(data = 'valid', model_deciles$valid),
  make.row.names = FALSE
)

# Plot average gains by predicted decile
deciles %>%
  ggplot(aes(factor(decile), gain, fill = data)) +
  geom_bar(stat = 'identity', position = 'dodge') +
  labs(title = 'Average gain by predicted decile', x = 'Decile', y = 'Minutes')
```



Visualize predictions

Compare actual gains to predicted gains for an out of time sample.

```
# Select data from an out of time sample
data_2008 <- flights_tbl %>%
  filter(!is.na(arrdelay) & !is.na(depdelay) & !is.na(distance)) %>%
  filter(depdelay > 15 & depdelay < 240) %>%
  filter(arrdelay > -60 & arrdelay < 360) %>%
  filter(year == 2008) %>%
  left_join(airlines_tbl, by = c("uniquecarrier" = "code")) %>%
  mutate(gain = depdelay - arrdelay) %>%
  select(year, month, arrdelay, depdelay, distance, uniquecarrier, description, gain, origin,dest)

# Summarize data by carrier
carrier <- sdf_predict(ml1, data_2008) %>%
  group_by(description) %>%
  summarize(gain = mean(gain), prediction = mean(prediction), freq = n()) %>%
  filter(freq > 10000) %>%
  collect

# Plot actual gains and predicted gains by airline carrier
ggplot(carrier, aes(gain, prediction)) +
  geom_point(alpha = 0.75, color = 'red', shape = 3) +
  geom_abline(intercept = 0, slope = 1, alpha = 0.15, color = 'blue') +
  geom_text(aes(label = substr(description, 1, 20)), size = 3, alpha = 0.75, vjust = -1) +
  labs(title='Average Gains Forecast', x = 'Actual', y = 'Predicted')
```



Some carriers make up more time than others in flight, but the differences are relatively small. The average time gains between the best and worst airlines is only six minutes. The best predictor of time gained is not carrier but flight distance. The biggest gains were associated with the longest flights.

FAQ

FusionInsight集群不允许访问网络，如何安装R

- 在集群外同版本的Redhat版本下按照配置EPEL的源安装R进行操作，最后一步不要执行 `yum install R`
- 执行 `yum install yum-utils` 安装yumdownloader
- 执行 `yumdownloader R --resolve --destdir=/tmp/packages` 把所有的rpm安装包下载到 `/tmp/packages` 中
- 将 `/tmp/packages` 中的所有rpm包复制到集群每个节点的 `/tmp/packages` 中
- 切换到集群每个节点的 `/tmp/packages` 中，执行 `yum localinstall *.rpm` 完成安装

安装sparklyr报错configuration failed for package 'openssl'

- 操作系统需要执行 `yum install openssl-devel` 安装openssl-devel

如何获取本文中使用sparklyr分析的源数据

- 执行以下shell脚本获取待分析的数据

```
# Make download directory
mkdir /tmp/flights

# Download flight data by year
for i in {2006..2008}
  do
    echo "$(date) $i Download"
    fnam=$i.csv.bz2
    wget -O /tmp/flights/$fnam http://stat-computing.org/dataexpo/2009/$fnam
    echo "$(date) $i Unzip"
    bunzip2 /tmp/flights/$fnam
  done

# Download airline carrier data
wget --no-check-certificate -O /tmp/airlines.csv http://www.transtats.bts.gov/Download_Lookup.asp?Lookup=L_UNIQUE_CARRIERS

# Download airports data
wget --no-check-certificate -O /tmp/airports.csv https://raw.githubusercontent.com/jpatokal/openflights/master/data/airports.dat
```

- 将下载下来的/tmp/flights目录以及/tmp/airlines.csv，/tmp/airports.csv文件上传到HDFS的/user/sparkuser目录中，然后在Hive中创建三张表，将数据加载到对应的表中

```
  hdfs dfs -mkdir /user/sparkuser/flights
  hdfs dfs -put flights/* /user/sparkuser/flights/
  hdfs dfs -put airlines.csv /user/sparkuser/
  hdfs dfs -put airports.csv /user/sparkuser/
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS flights
(
  year int,
  month int,
  dayofmonth int,
  dayofweek int,
  deptime int,
  crsdeptime int,
  arrtime int,
  crsarrtime int,
  uniquecarrier string,
  flightnum int,
  tailnum string,
  actualelapsedtime int,
  crselapsedtime int,
  airtime string,
  arrdelay int,
  depdelay int,
  origin string,
  dest string,
  distance int,
  taxiin string,
  taxiout string,
  cancelled int,
  cancellationcode string,
  diverted int,
  carrierdelay string,
  weatherdelay string,
  nasdelay string,
  securitydelay string,
  lateaircraftdelay string
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
TBLPROPERTIES("skip.header.line.count"="1");
```

```
LOAD DATA INPATH '/user/sparkuser/flights/2006.csv' INTO TABLE flights;
LOAD DATA INPATH '/user/sparkuser/flights/2007.csv' INTO TABLE flights;
LOAD DATA INPATH '/user/sparkuser/flights/2008.csv' INTO TABLE flights;
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS airlines
(
Code string,
Description string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES
(
"separatorChar" = '\,',
"quoteChar"     = '\"'
)
STORED AS TEXTFILE
tblproperties("skip.header.line.count"="1");
```

```
LOAD DATA INPATH '/user/sparkuser/airlines.csv' INTO TABLE airlines;
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS airports
(
id string,
name string,
city string,
country string,
faa string,
icao string,
lat double,
lon double,
alt int,
tz_offset double,
dst string,
tz_name string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES
(
"separatorChar" = '\,',
"quoteChar"     = '\"'
)
```

```
STORED AS TEXTFILE;
```

```
LOAD DATA INPATH '/user/sparkuser/airports.csv' INTO TABLE airports;
```