NERDERY®

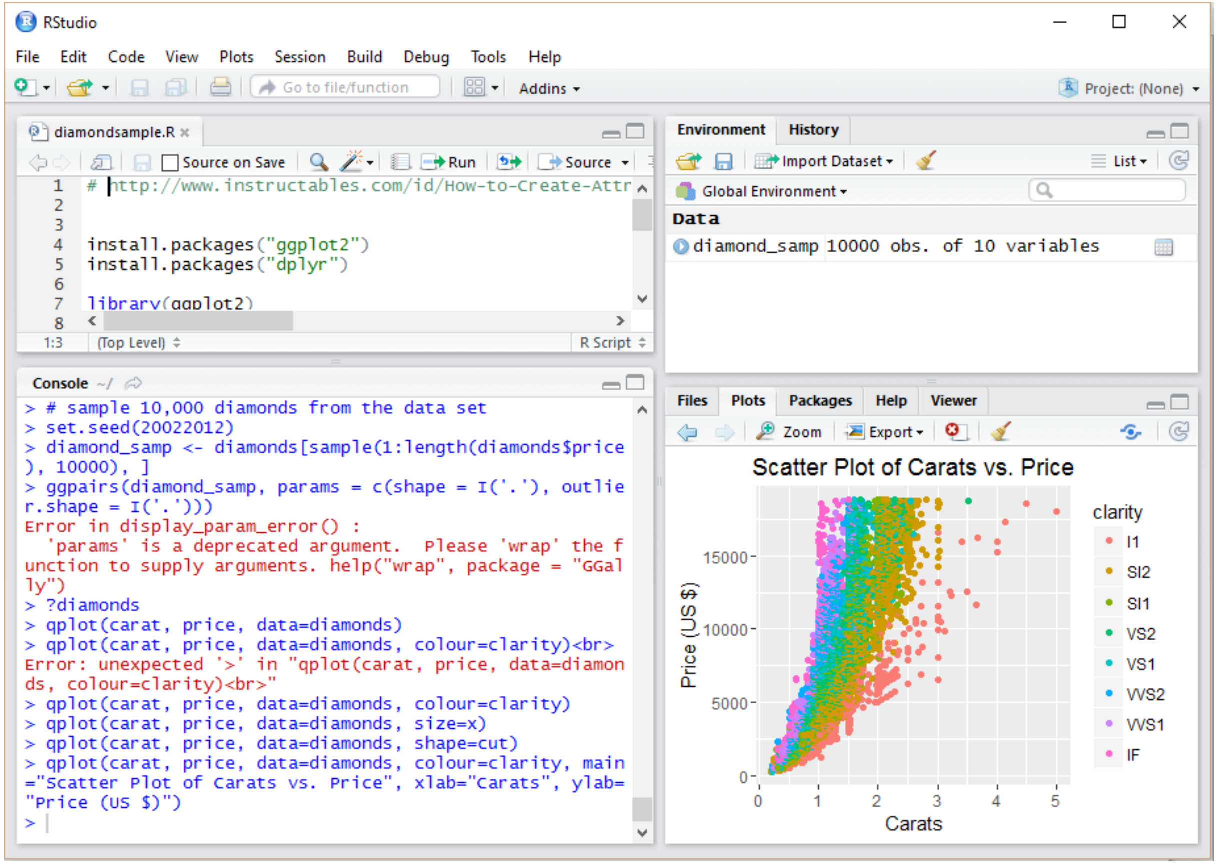
R Studio Server on Amazon EMR

Chad Dvoracek

Brandon Veber

RStudio

- **Leading IDE for R**
- **Integrated console**
- **Code completion**
- **Syntax highlighting editor**
- **Direct code execution**
- **Plotting, history, debugging**
- **Open source**



```
# http://www.instructables.com/id/How-to-Create-Attr


install.packages("ggplot2")
install.packages("dplyr")

library(ggplot2)
```

Data

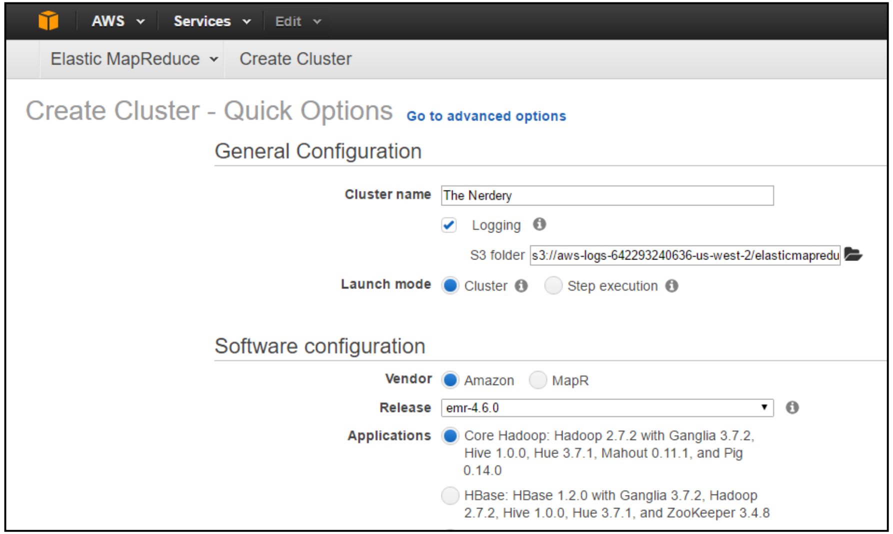
diamond_samp 10000 obs. of 10 variables

```
> # sample 10,000 diamonds from the data set
> set.seed(20022012)
> diamond_samp <- diamonds[sample(1:length(diamonds$price
), 10000), ]
> ggpairs(diamond_samp, params = c(shape = I('.'), outlie
r.shape = I('.')))
Error in display_param_error() :
  'params' is a deprecated argument.  Please 'wrap' the f
unction to supply arguments. help("wrap", package = "GGal
ly")
> ?diamonds
> qplot(carat, price, data=diamonds)
> qplot(carat, price, data=diamonds, colour=clarity)<br>
Error: unexpected '>' in "qplot(carat, price, data=diamon
ds, colour=clarity)<br>"
> qplot(carat, price, data=diamonds, colour=clarity)
> qplot(carat, price, data=diamonds, size=x)
> qplot(carat, price, data=diamonds, shape=cut)
> qplot(carat, price, data=diamonds, colour=clarity, main
="Scatter Plot of Carats vs. Price", xlab="Carats", ylab=
"Price (US $)")
> 
```

EMR (Elastic MapReduce)

Managed Hadoop Framework

- Easy to use
- Quick set up
- Low cost (Spot Instance)
- Elastic
- Flexible



Create Cluster - Quick Options Go to advanced options

General Configuration

Cluster name: The Nerdery

☑ Logging

S3 folder: s3://aws-logs-642293240636-us-west-2/elasticmapredu

Launch mode: ● Cluster ○ Step execution

Software configuration

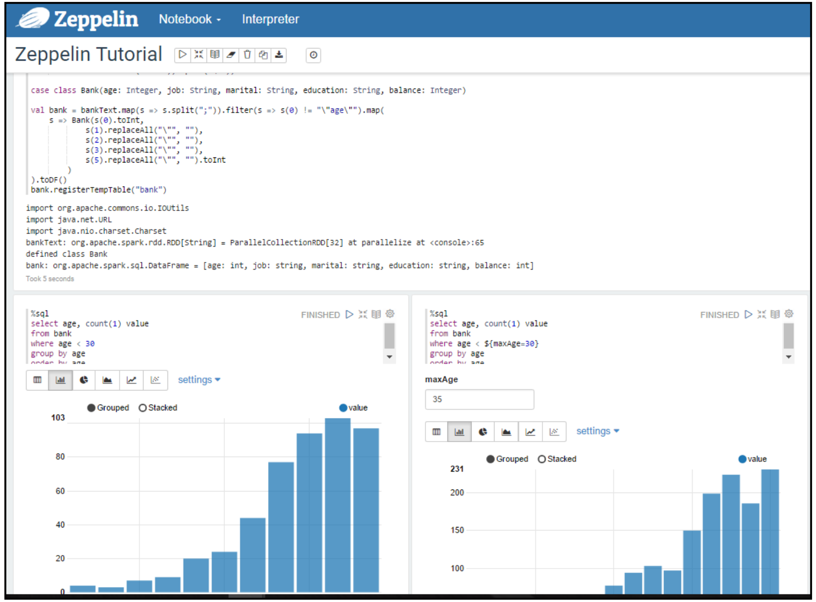
Vendor: ● Amazon ○ MapR

Release: emr-4.6.0

Applications:
● Core Hadoop: Hadoop 2.7.2 with Ganglia 3.7.2, Hive 1.0.0, Hue 3.7.1, Mahout 0.11.1, and Pig 0.14.0
○ HBase: HBase 1.2.0 with Ganglia 3.7.2, Hadoop 2.7.2, Hive 1.0.0, Hue 3.7.1, and ZooKeeper 3.4.8

GUI Interactive Environments

Amazon EMR

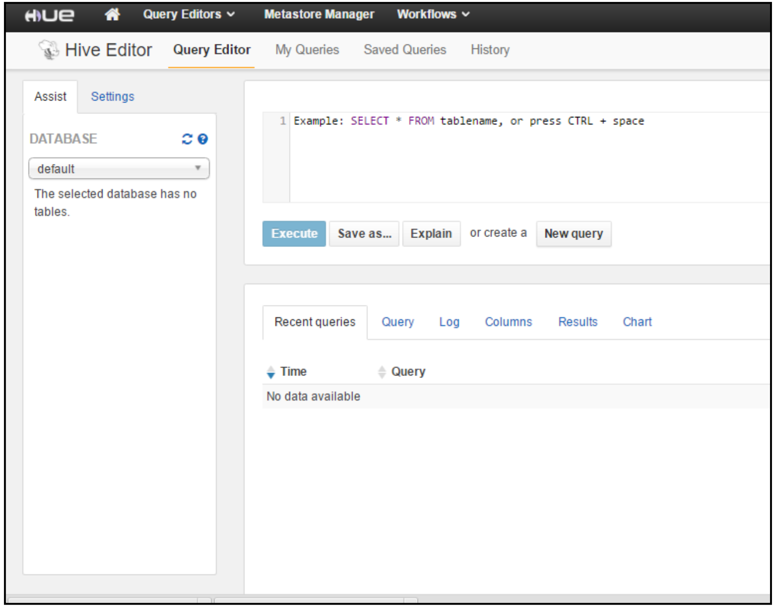


```
case class Bank(age: Integer, job: String, marital: String, education: String, balance: Integer)

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

```
import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
bankText: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[32] at parallelize at <console>:65
defined class Bank
bank: org.apache.spark.sql.DataFrame = [age: int, job: string, marital: string, education: string, balance: int]
Took 5 seconds
```

```
%sql
select age, count(1) value
from bank
where age < 30
group by age
order by age
```

```
%sql
select age, count(1) value
from bank
where age < ${maxAge=30}
group by age
order by age
```



default

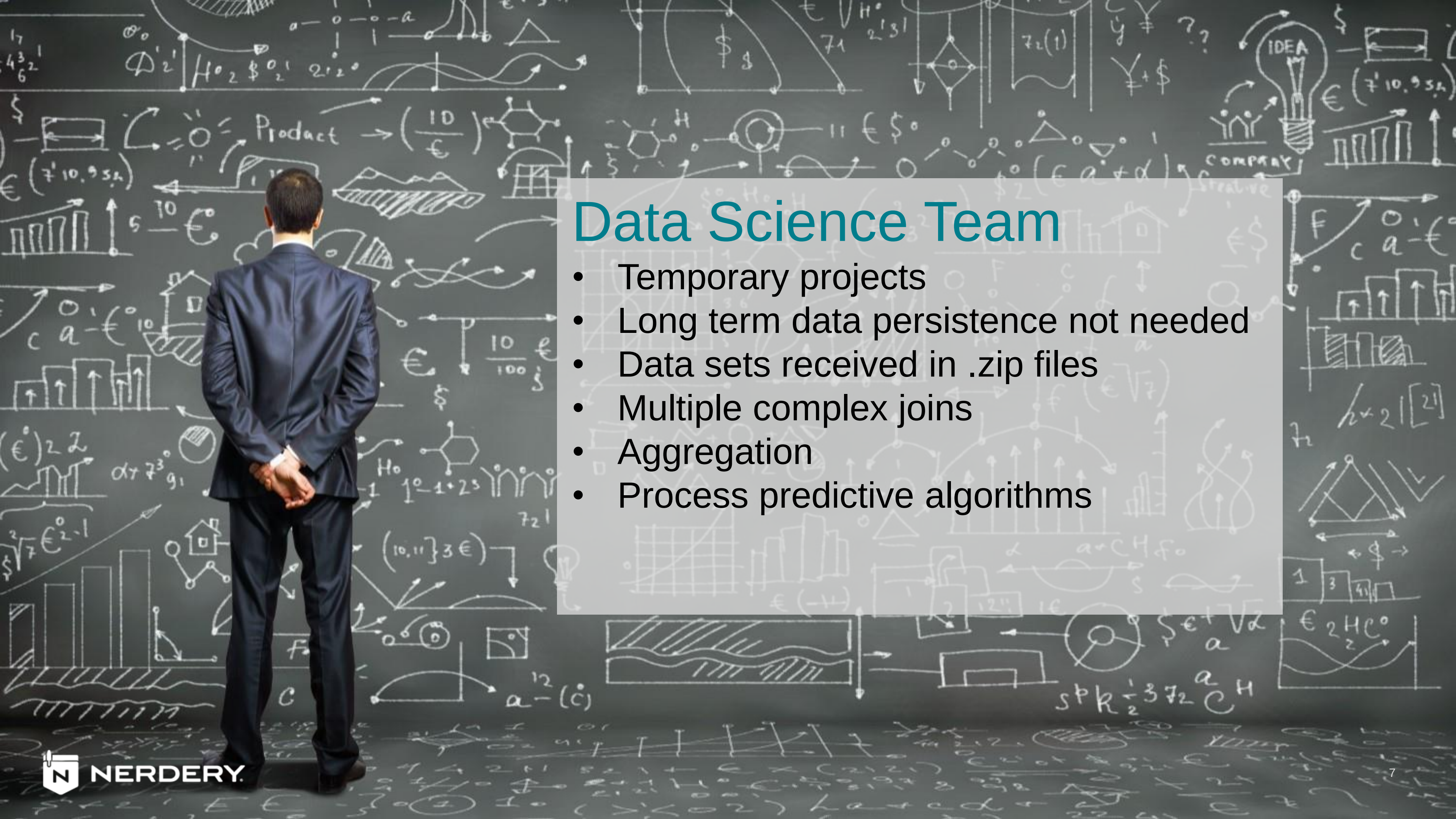
The selected database has no tables.

```
Example: SELECT * FROM tablename, or press CTRL + space
```

No data available

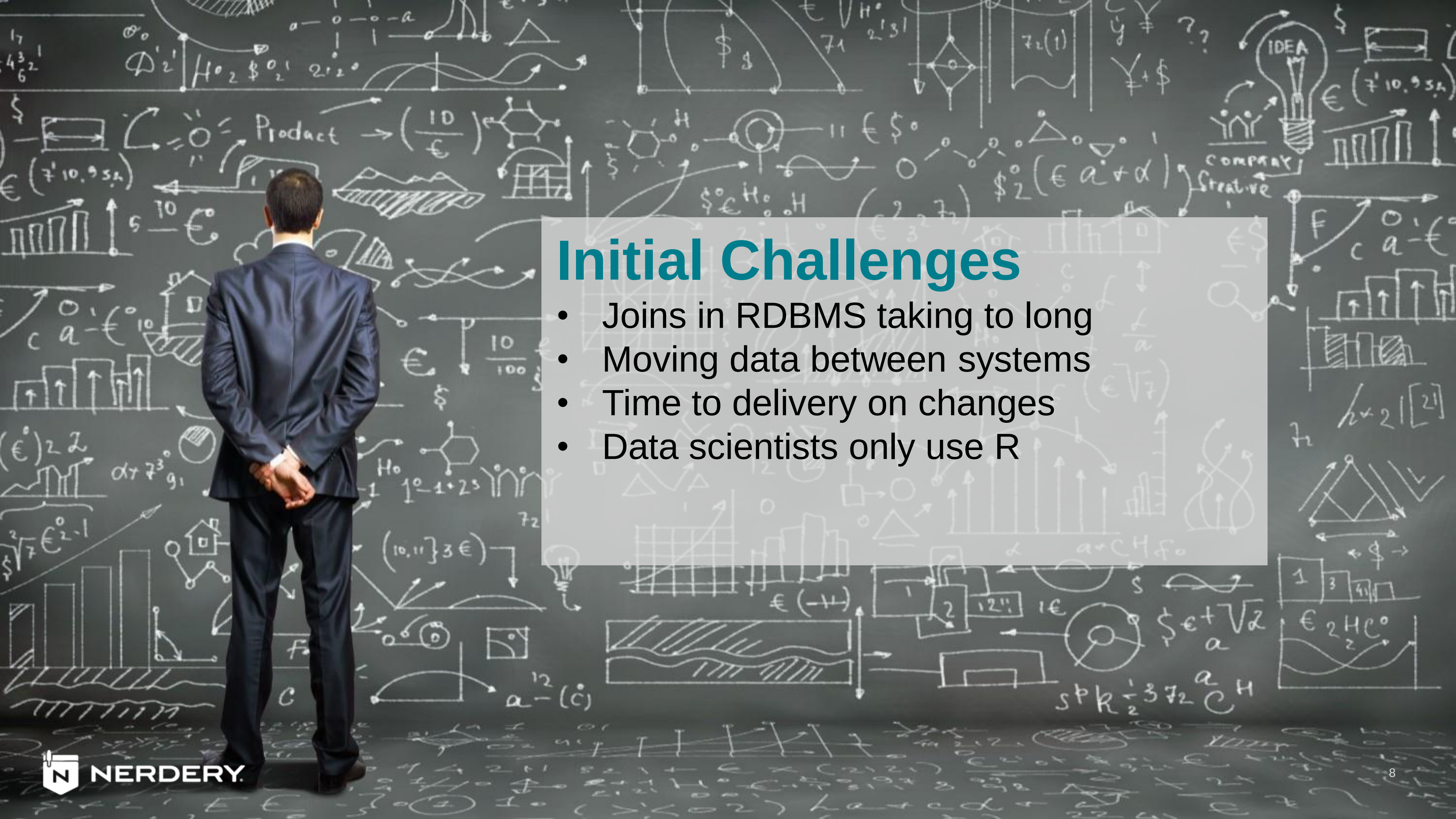
Case Study

Speeding data deliver to data scientists.



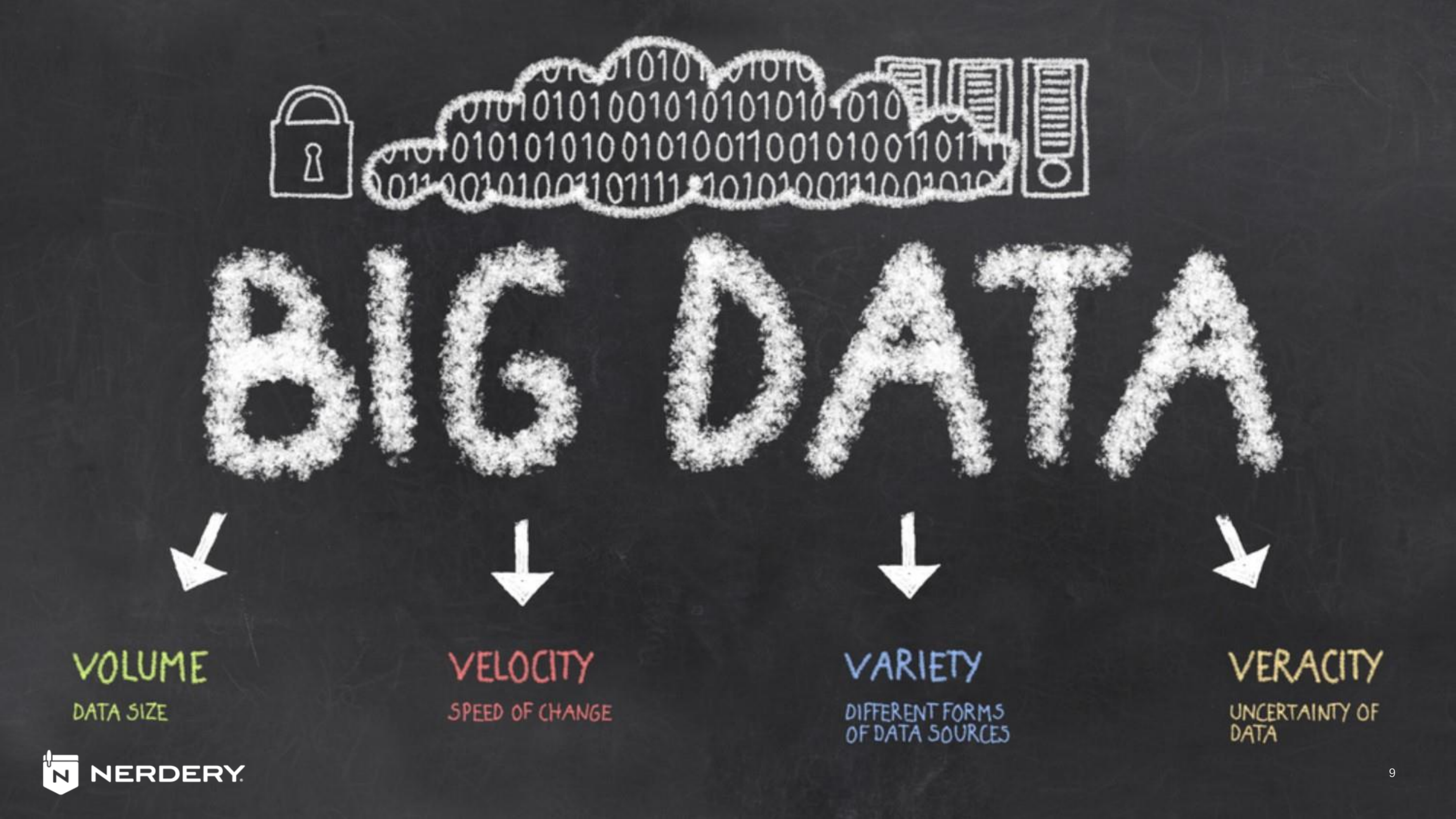
Data Science Team

- Temporary projects
- Long term data persistence not needed
- Data sets received in .zip files
- Multiple complex joins
- Aggregation
- Process predictive algorithms



Initial Challenges

- Joins in RDBMS taking to long
- Moving data between systems
- Time to delivery on changes
- Data scientists only use R


BIG DATA
VOLUME
DATA SIZE
VELOCITY
SPEED OF CHANGE
VARIETY
DIFFERENT FORMS OF DATA SOURCES
VERACITY
UNCERTAINTY OF DATA

The Hadoop ecosystem can run in Amazon EMR

MapReduce
AVRO
hadoop HDFS
HIVE
FLUME
Spark
APACHE DRILL
R
APACHE Phoenix
chukwa
TEZ
presto
cascading
MAPR
accumulo
APACHE HBASE
mahout
gradle

Case Study: Results

- ✓ **Reduced time for data delivery**

  Moving to EMR allowed for faster processing and preparing of the data. By incorporating automating data load scripts and utilizing Hive for batch processing and Spark for in memory processing it allowed the data scientist to focus on solutions rather than time constraints.

- ✓ **Confidence utilizing Big Data systems**

  By providing assistance in set up and training for data processing it allowed the data engineering team the ability to gain confidence in preparing data on distributed systems.

- ✓ **Process change**

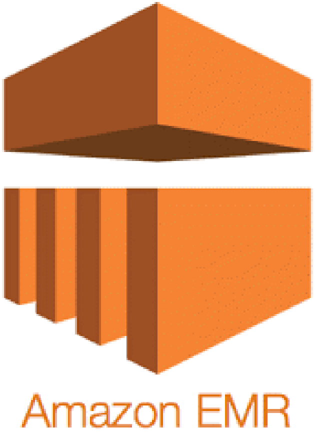
  Having the option to transform data in distributed systems provided an opportunity to re-think the process and time needed for data and solution delivery.

- ✓ **Future considerations**

  With the data scientist working primarily in R, exploring R Studio Server and SparkR may be a logical next step in improving the teams workflow.

Bootstrap Problems

+



= ?

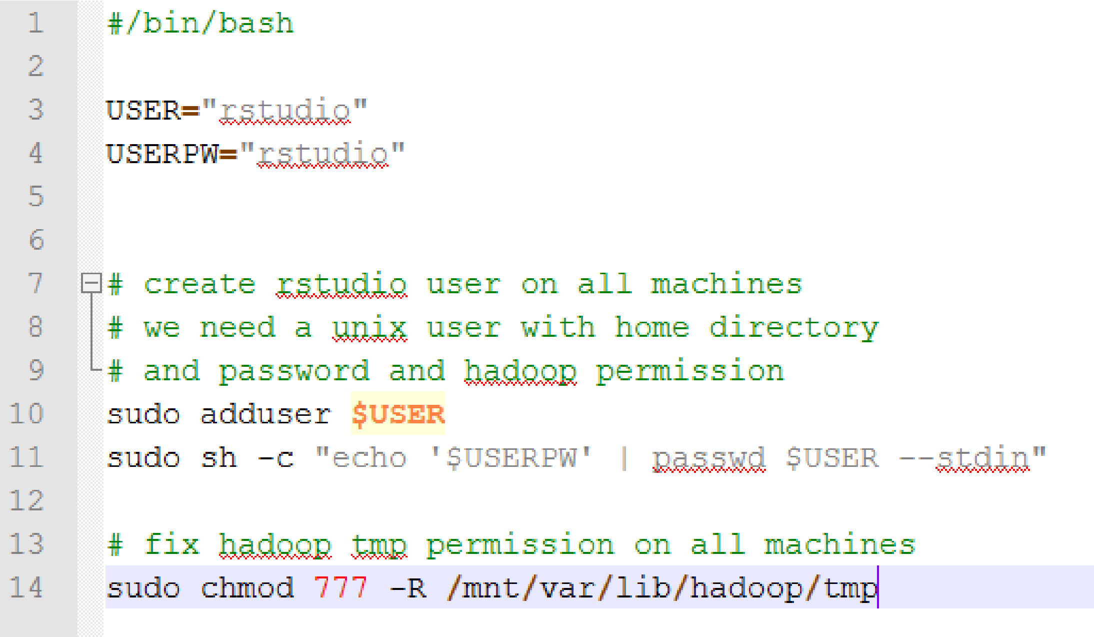
What about Spark R?

Spark™
R
Amazon EMR

EMR Set Up

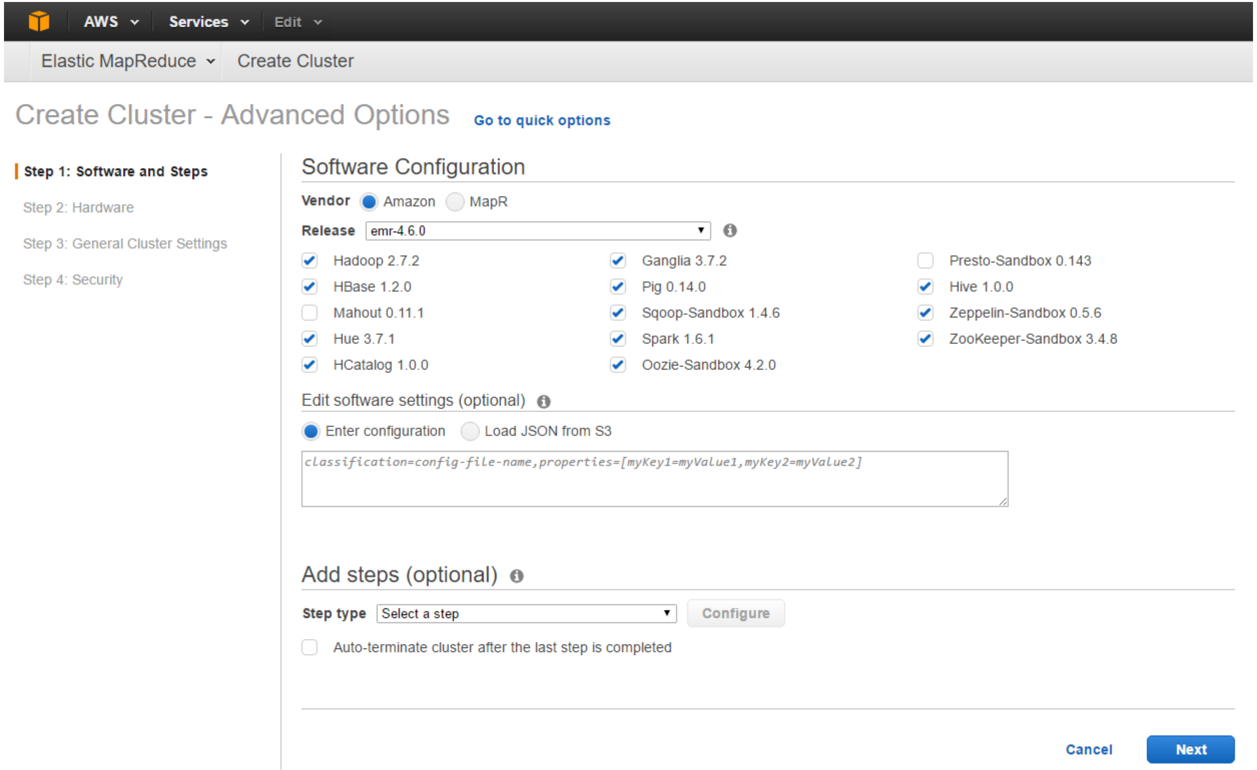
Quick & Simple

Simple Start Up Script



```bash
#/bin/bash

USER="rstudio"
USERPW="rstudio"


# create rstudio user on all machines
# we need a unix user with home directory
# and password and hadoop permission
sudo adduser $USER
sudo sh -c "echo '$USERPW' | passwd $USER --stdin"

# fix hadoop tmp permission on all machines
sudo chmod 777 -R /mnt/var/lib/hadoop/tmp
```

EMR Configuration: Step 1



Create Cluster - Advanced Options

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

Software Configuration

Vendor ● Amazon ○ MapR

Release emr-4.6.0

☑ Hadoop 2.7.2
☑ HBase 1.2.0
□ Mahout 0.11.1
☑ Hue 3.7.1
☑ HCatalog 1.0.0

☑ Ganglia 3.7.2
☑ Pig 0.14.0
☑ Sqoop-Sandbox 1.4.6
☑ Spark 1.6.1
☑ Oozie-Sandbox 4.2.0

□ Presto-Sandbox 0.143
☑ Hive 1.0.0
☑ Zeppelin-Sandbox 0.5.6
☑ ZooKeeper-Sandbox 3.4.8

Edit software settings (optional)

● Enter configuration ○ Load JSON from S3

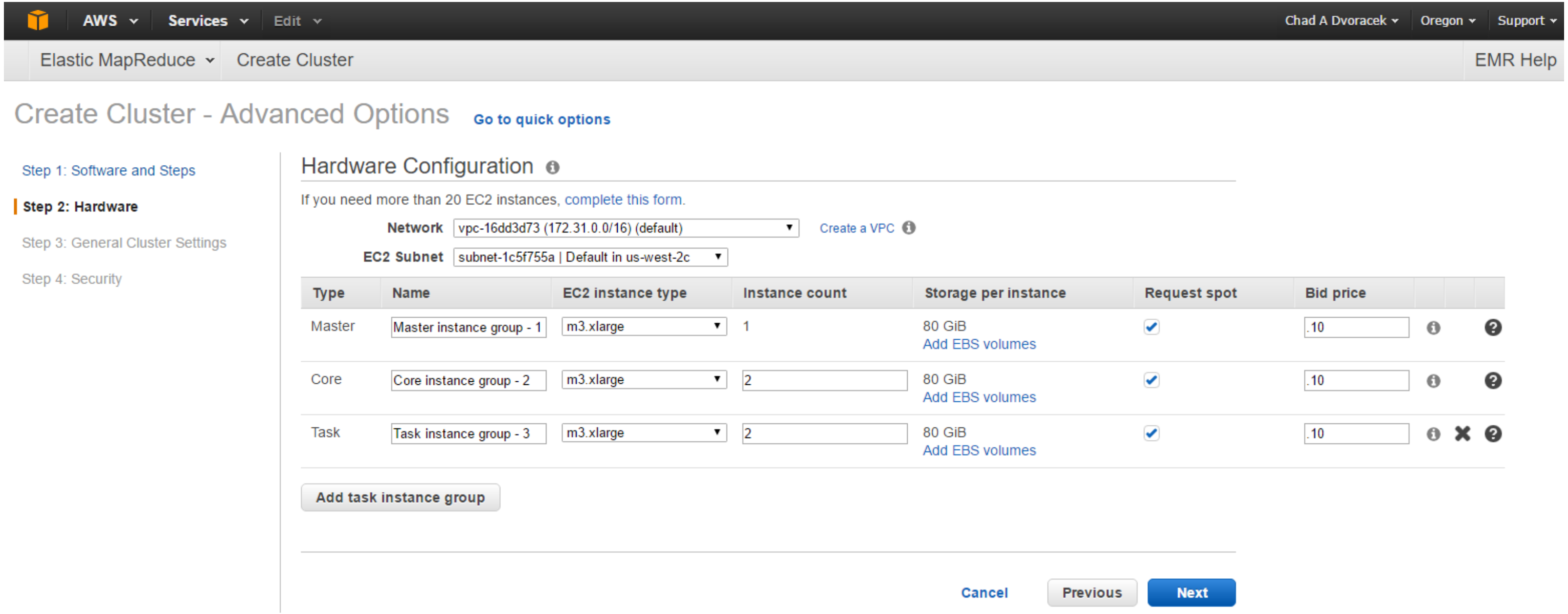
```
classification=config-file-name,properties=[myKey1=myValue1,myKey2=myValue2]
```

Add steps (optional)

Step type Select a step

□ Auto-terminate cluster after the last step is completed

EMR Configuration: Step 2



Create Cluster - Advanced Options

Hardware Configuration

If you need more than 20 EC2 instances, complete this form.

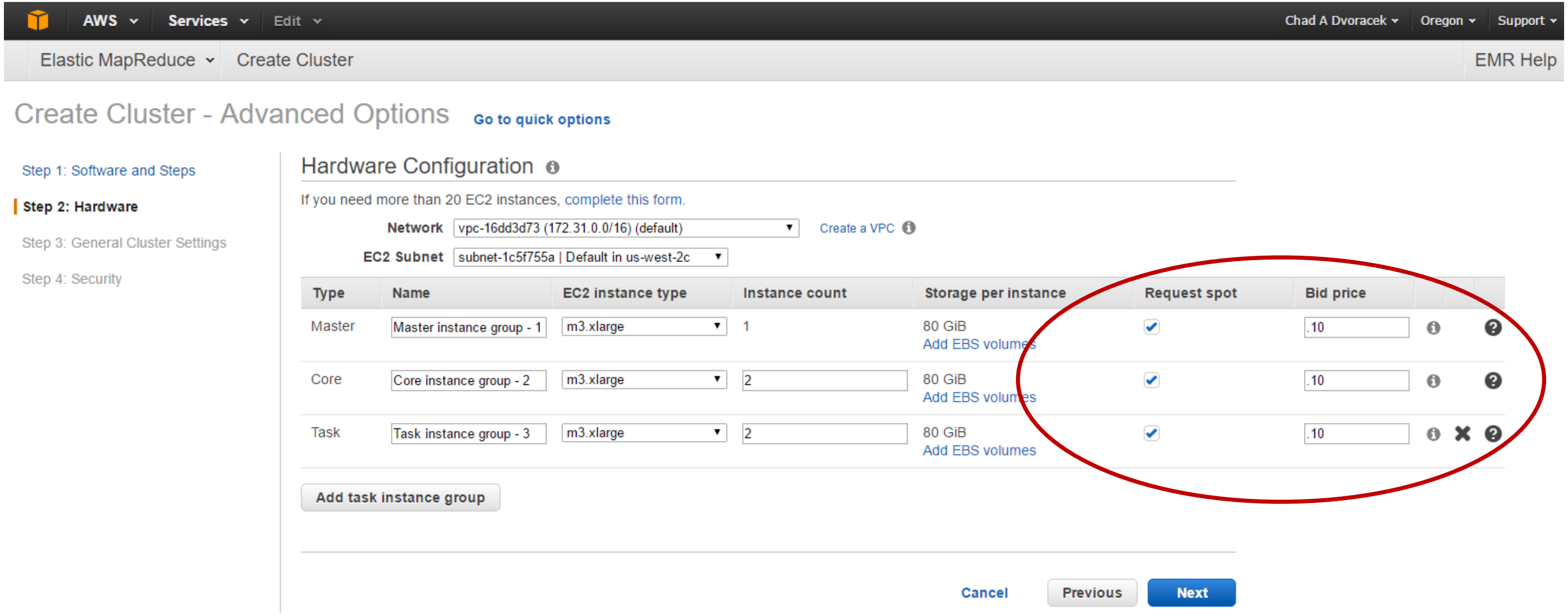
Network vpc-16dd3d73 (172.31.0.0/16) (default)

EC2 Subnet subnet-1c5f755a | Default in us-west-2c

Type	Name	EC2 instance type	Instance count	Storage per instance	Request spot	Bid price
Master	Master instance group - 1	m3.xlarge	1	80 GiB Add EBS volumes	☑	.10
Core	Core instance group - 2	m3.xlarge	2	80 GiB Add EBS volumes	☑	.10
Task	Task instance group - 3	m3.xlarge	2	80 GiB Add EBS volumes	☑	.10

NERDERY

EMR Configuration: Step 2



Create Cluster - Advanced Options **Go to quick options**

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

Hardware Configuration

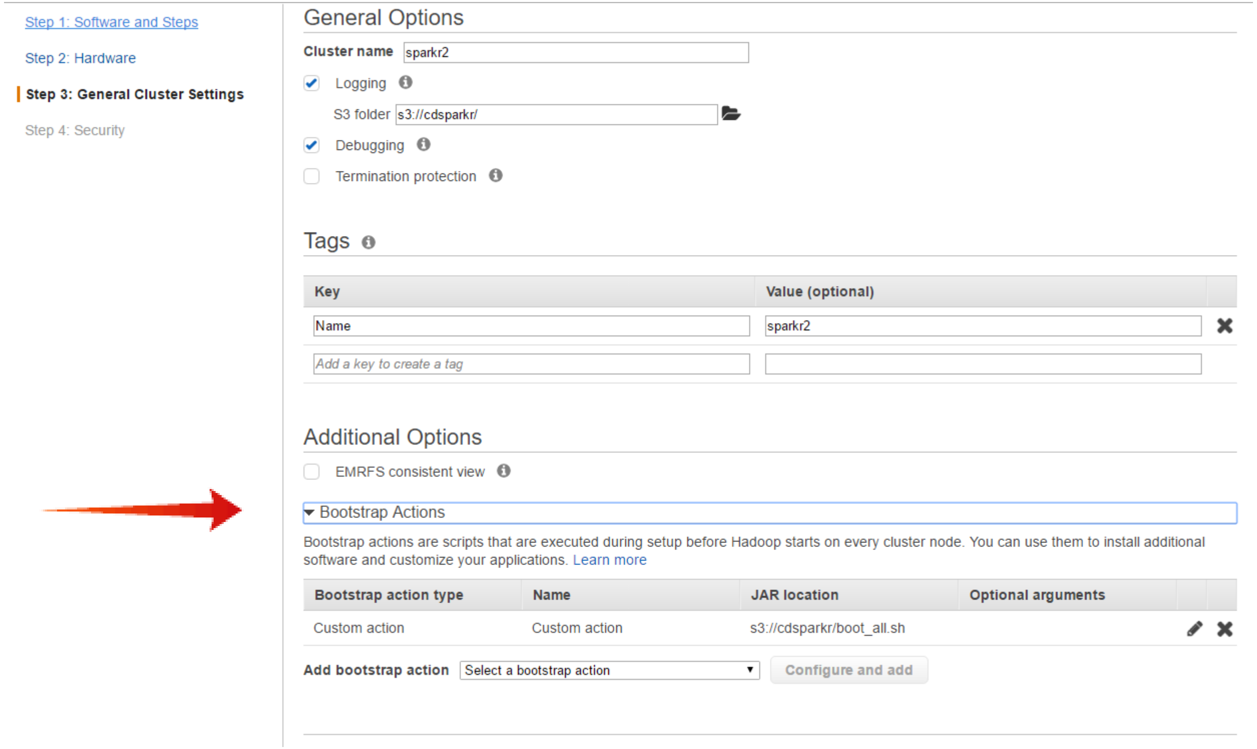
If you need more than 20 EC2 instances, complete this form.

Network vpc-16dd3d73 (172.31.0.0/16) (default) Create a VPC

EC2 Subnet subnet-1c5f755a | Default in us-west-2c

Type	Name	EC2 instance type	Instance count	Storage per instance	Request spot	Bid price
Master	Master instance group - 1	m3.xlarge	1	80 GiB Add EBS volumes	☑	.10
Core	Core instance group - 2	m3.xlarge	2	80 GiB Add EBS volumes	☑	.10
Task	Task instance group - 3	m3.xlarge	2	80 GiB Add EBS volumes	☑	.10

EMR Configuration: Step 3



Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

General Options

Cluster name sparkr2

☑ Logging

S3 folder s3://cdsparkr/

☑ Debugging

□ Termination protection

Tags

Key	Value (optional)	
Name	sparkr2	✖
Add a key to create a tag		

Additional Options

□ EMRFS consistent view

▾ Bootstrap Actions

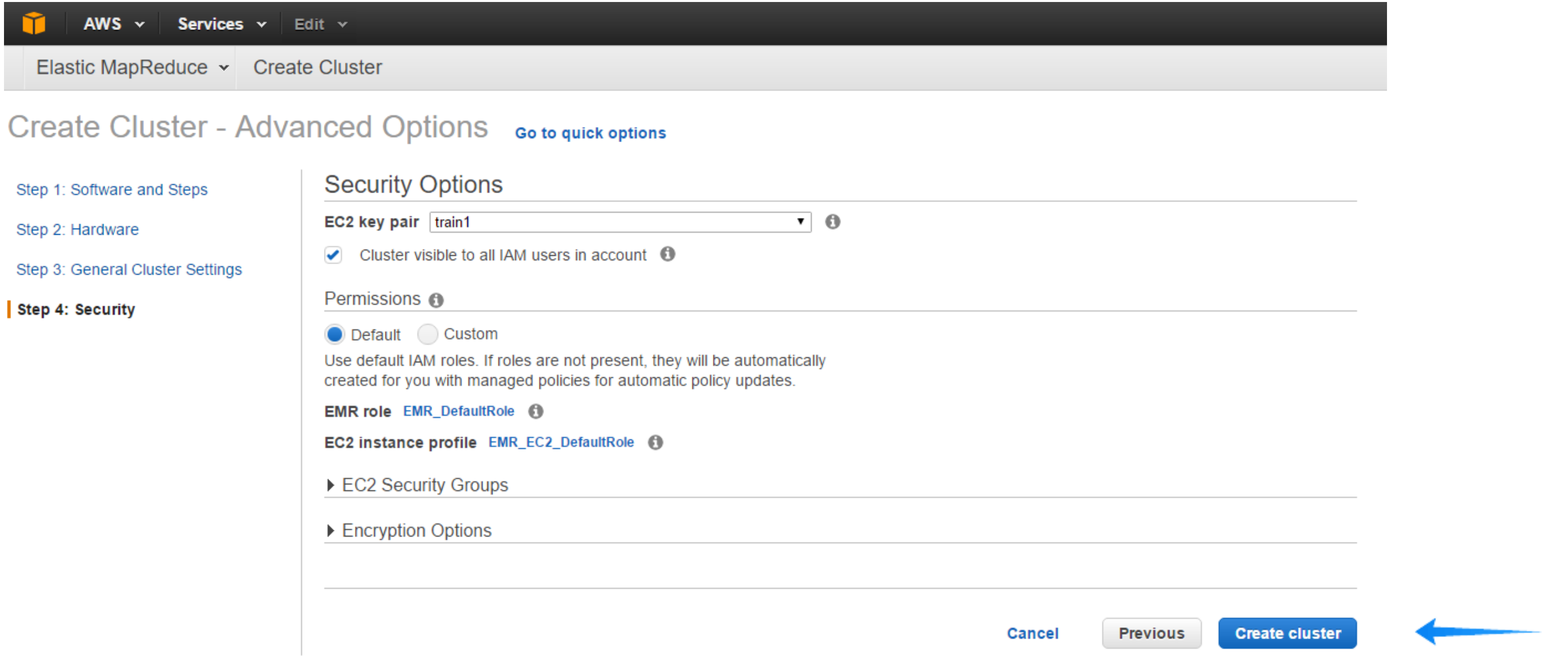
Bootstrap actions are scripts that are executed during setup before Hadoop starts on every cluster node. You can use them to install additional software and customize your applications. Learn more

Bootstrap action type	Name	JAR location	Optional arguments	
Custom action	Custom action	s3://cdsparkr/boot_all.sh		✎ ✖

Add bootstrap action Select a bootstrap action ▾ Configure and add

NERDERY

EMR Configuration: Step 4



Create Cluster - Advanced Options

Security Options

EC2 key pair: train1

☑ Cluster visible to all IAM users in account

Permissions

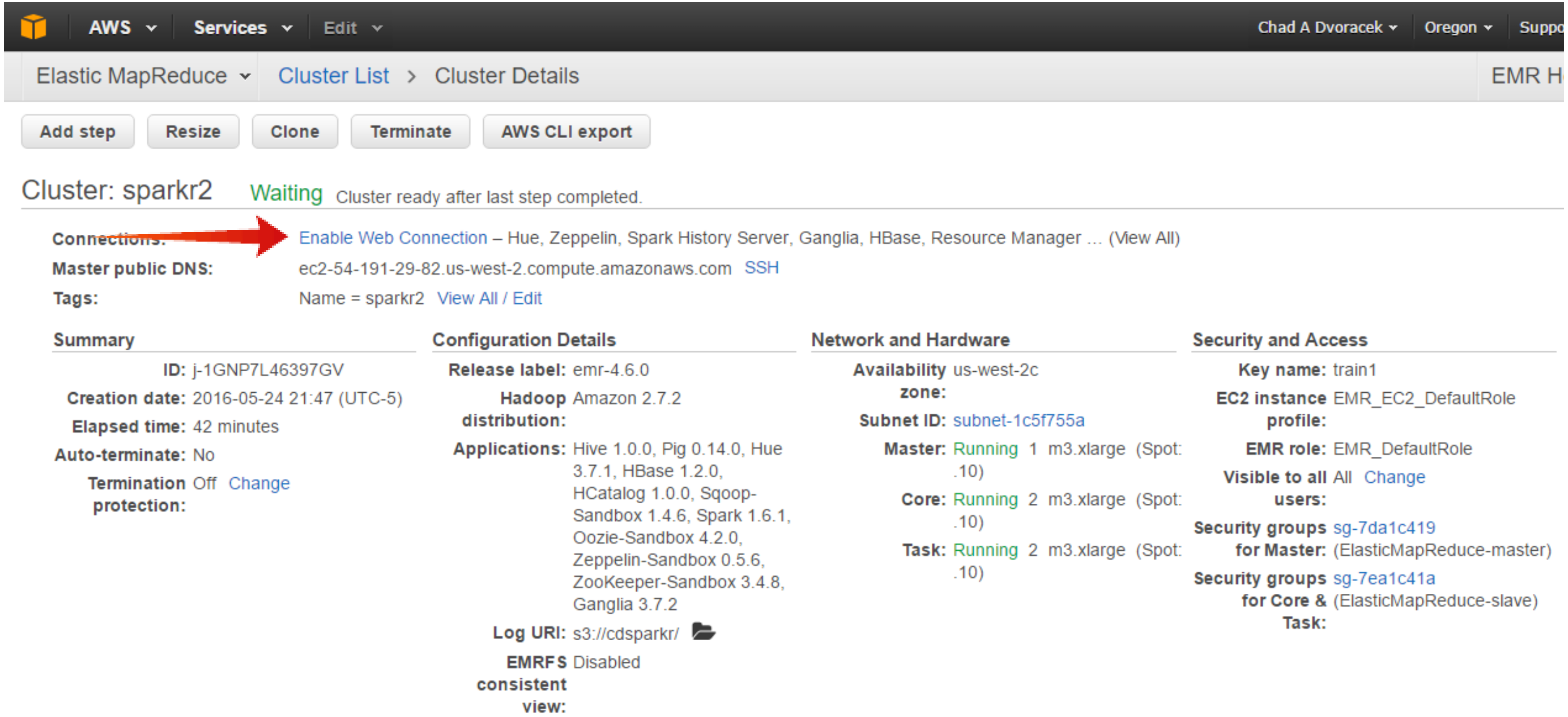
● Default ○ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role: EMR_DefaultRole

EC2 instance profile: EMR_EC2_DefaultRole

EMR Web Connection



Cluster: sparkr2 Waiting Cluster ready after last step completed.

Connections:	Enable Web Connection – Hue, Zeppelin, Spark History Server, Ganglia, HBase, Resource Manager ... (View All)
Master public DNS:	ec2-54-191-29-82.us-west-2.compute.amazonaws.com SSH
Tags:	Name = sparkr2 View All / Edit

Summary

ID: j-1GNP7L46397GV

Creation date: 2016-05-24 21:47 (UTC-5)

Elapsed time: 42 minutes

Auto-terminate: No

Termination protection: Off Change

Configuration Details

Release label: emr-4.6.0

Hadoop distribution: Amazon 2.7.2

Applications: Hive 1.0.0, Pig 0.14.0, Hue 3.7.1, HBase 1.2.0, HCatalog 1.0.0, Sqoop-Sandbox 1.4.6, Spark 1.6.1, Oozie-Sandbox 4.2.0, Zeppelin-Sandbox 0.5.6, ZooKeeper-Sandbox 3.4.8, Ganglia 3.7.2

Log URI: s3://cdsparkr/

EMRFS consistent view: Disabled

Network and Hardware

Availability zone: us-west-2c

Subnet ID: subnet-1c5f755a

Master: Running 1 m3.xlarge (Spot: .10)

Core: Running 2 m3.xlarge (Spot: .10)

Task: Running 2 m3.xlarge (Spot: .10)

Security and Access

Key name: train1

EC2 instance profile: EMR_EC2_DefaultRole

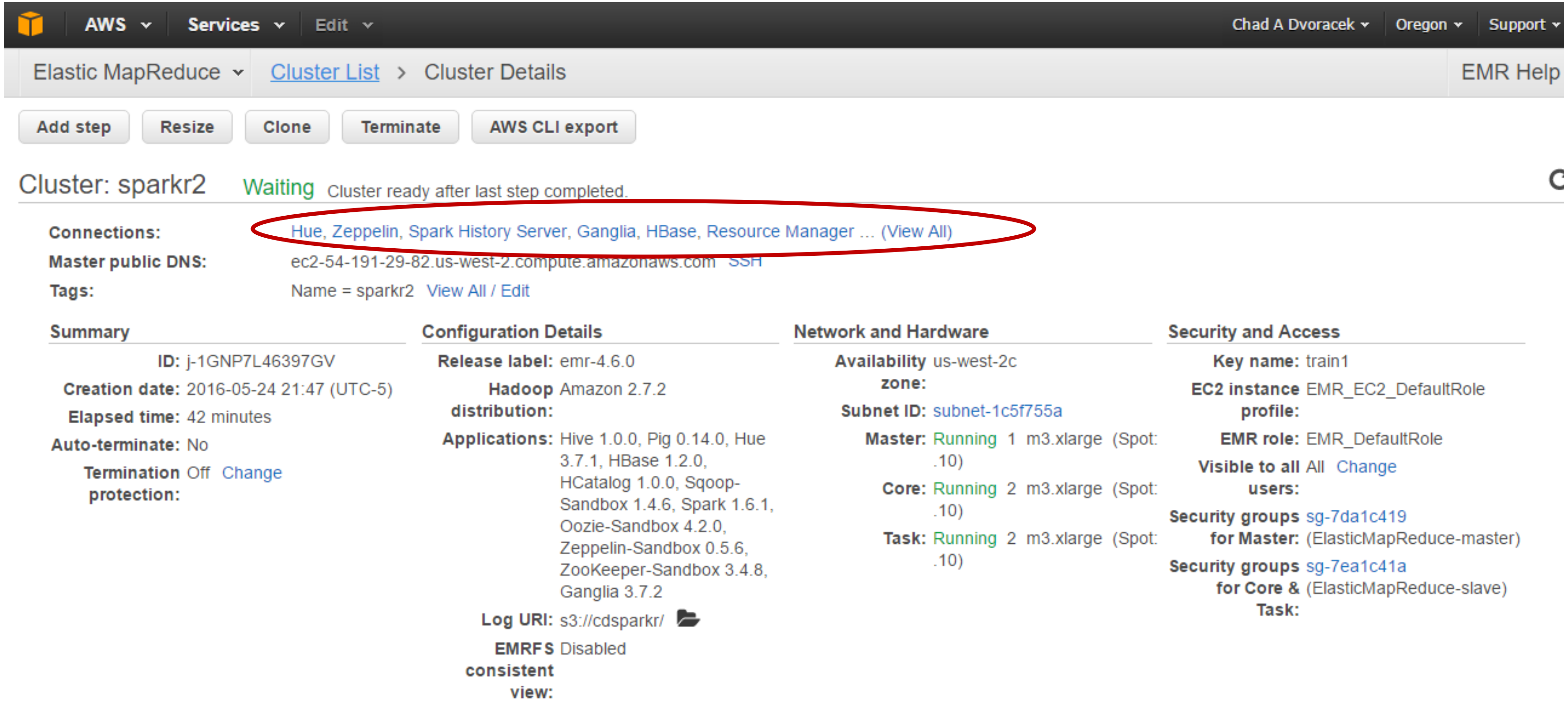
EMR role: EMR_DefaultRole

Visible to all users: All Change

Security groups for Master: sg-7da1c419 (ElasticMapReduce-master)

Security groups for Core & Task: sg-7ea1c41a (ElasticMapReduce-slave)

EMR Web Connection



Cluster: sparkr2 Waiting Cluster ready after last step completed.

Connections: Hue, Zeppelin, Spark History Server, Ganglia, HBase, Resource Manager ... (View All)

Master public DNS: ec2-54-191-29-82.us-west-2.compute.amazonaws.com SSH

Tags: Name = sparkr2 View All / Edit

Summary

ID: j-1GNP7L46397GV

Creation date: 2016-05-24 21:47 (UTC-5)

Elapsed time: 42 minutes

Auto-terminate: No

Termination protection: Off Change

Configuration Details

Release label: emr-4.6.0

Hadoop distribution: Amazon 2.7.2

Applications: Hive 1.0.0, Pig 0.14.0, Hue 3.7.1, HBase 1.2.0, HCatalog 1.0.0, Sqoop-Sandbox 1.4.6, Spark 1.6.1, Oozie-Sandbox 4.2.0, Zeppelin-Sandbox 0.5.6, ZooKeeper-Sandbox 3.4.8, Ganglia 3.7.2

Log URI: s3://cdsparkr/

EMRFS consistent view: Disabled

Network and Hardware

Availability zone: us-west-2c

Subnet ID: subnet-1c5f755a

Master: Running 1 m3.xlarge (Spot: .10)

Core: Running 2 m3.xlarge (Spot: .10)

Task: Running 2 m3.xlarge (Spot: .10)

Security and Access

Key name: train1

EC2 instance profile: EMR_EC2_DefaultRole

EMR role: EMR_DefaultRole

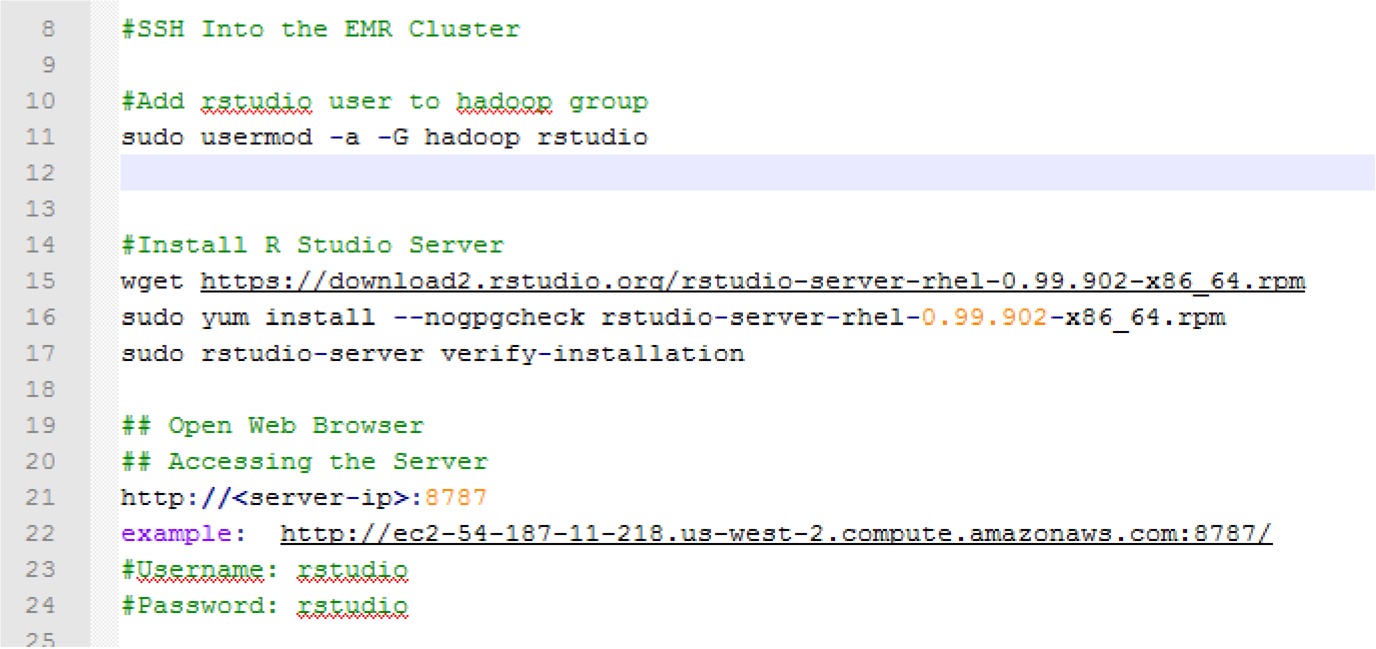
Visible to all users: All Change

Security groups for Master: sg-7da1c419 (ElasticMapReduce-master)

Security groups for Core & Task: sg-7ea1c41a (ElasticMapReduce-slave)

NERDERY

Final Step

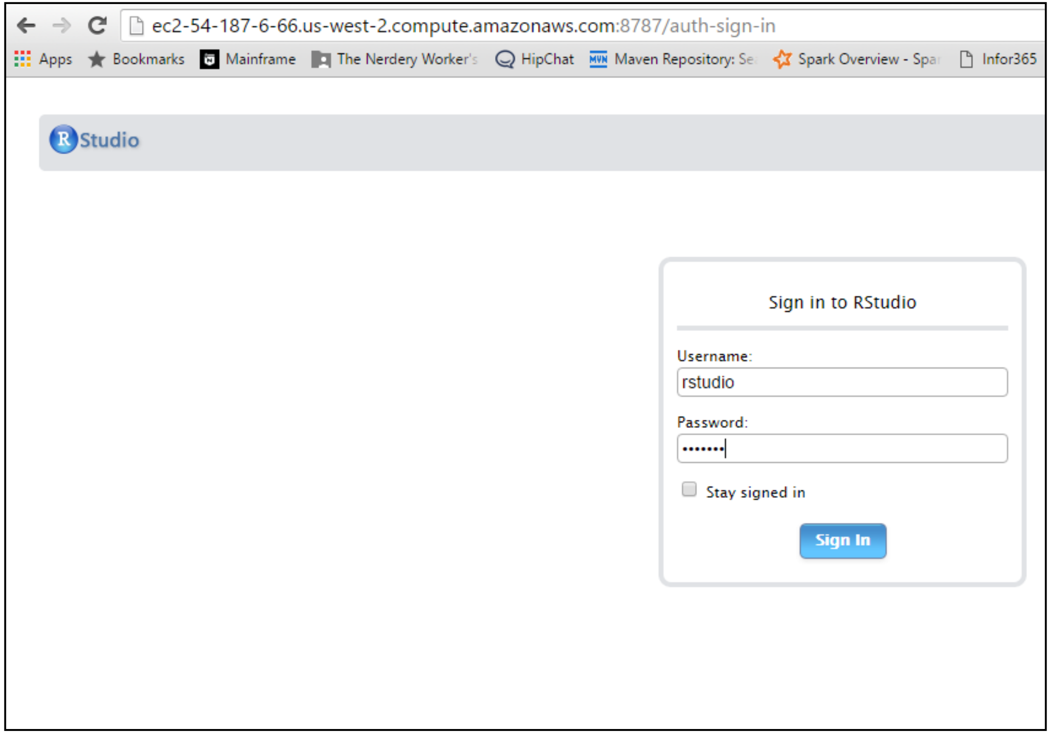


```
#SSH Into the EMR Cluster

#Add rstudio user to hadoop group
sudo usermod -a -G hadoop rstudio


#Install R Studio Server
wget https://download2.rstudio.org/rstudio-server-rhel-0.99.902-x86_64.rpm
sudo yum install --nogpgcheck rstudio-server-rhel-0.99.902-x86_64.rpm
sudo rstudio-server verify-installation

## Open Web Browser
## Accessing the Server
http://<server-ip>:8787
example:  http://ec2-54-187-11-218.us-west-2.compute.amazonaws.com:8787/
#Username: rstudio
#Password: rstudio
```

R Studio Server

R and SparkR

R Studio Server: Sign In



ec2-54-187-6-66.us-west-2.compute.amazonaws.com:8787/auth-sign-in

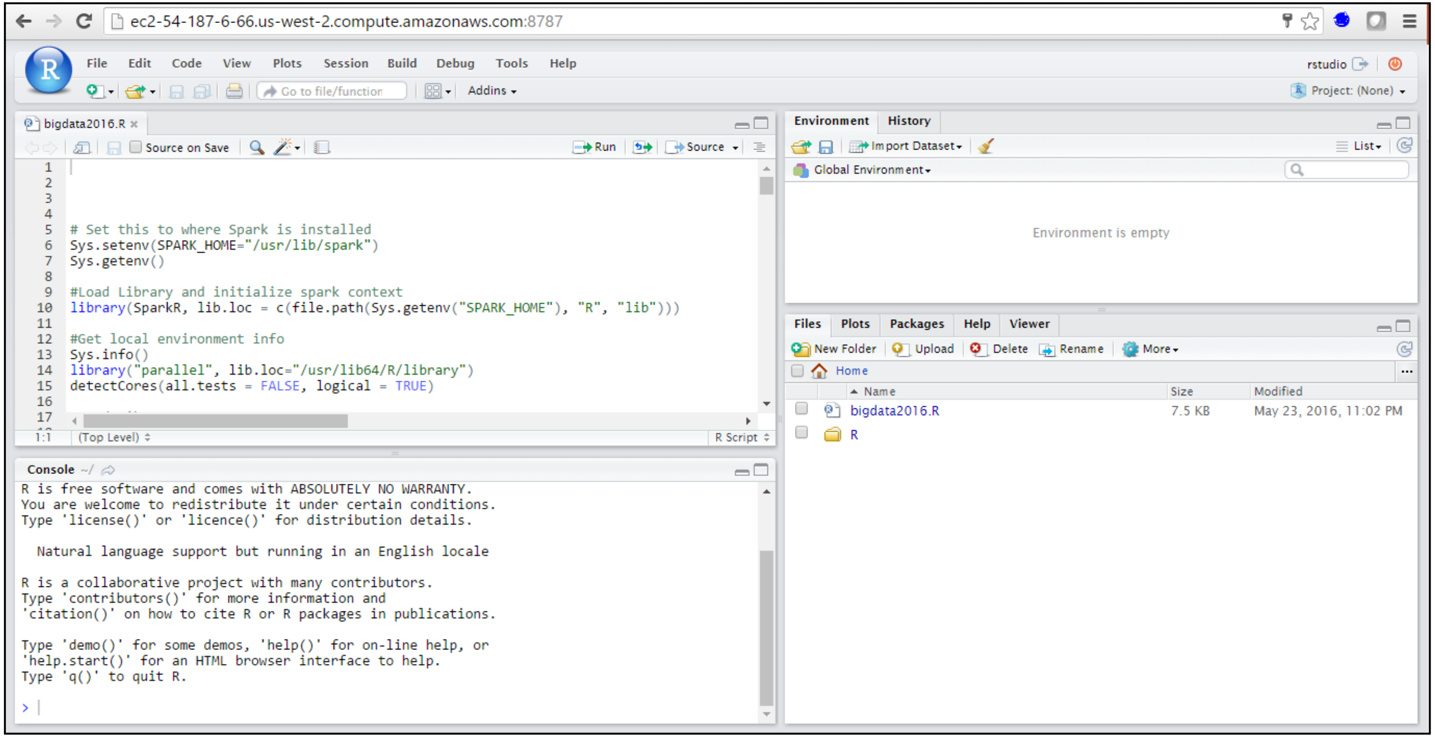
Sign in to RStudio

Username: rstudio

Password: •••••••

□ Stay signed in

R Studio Server



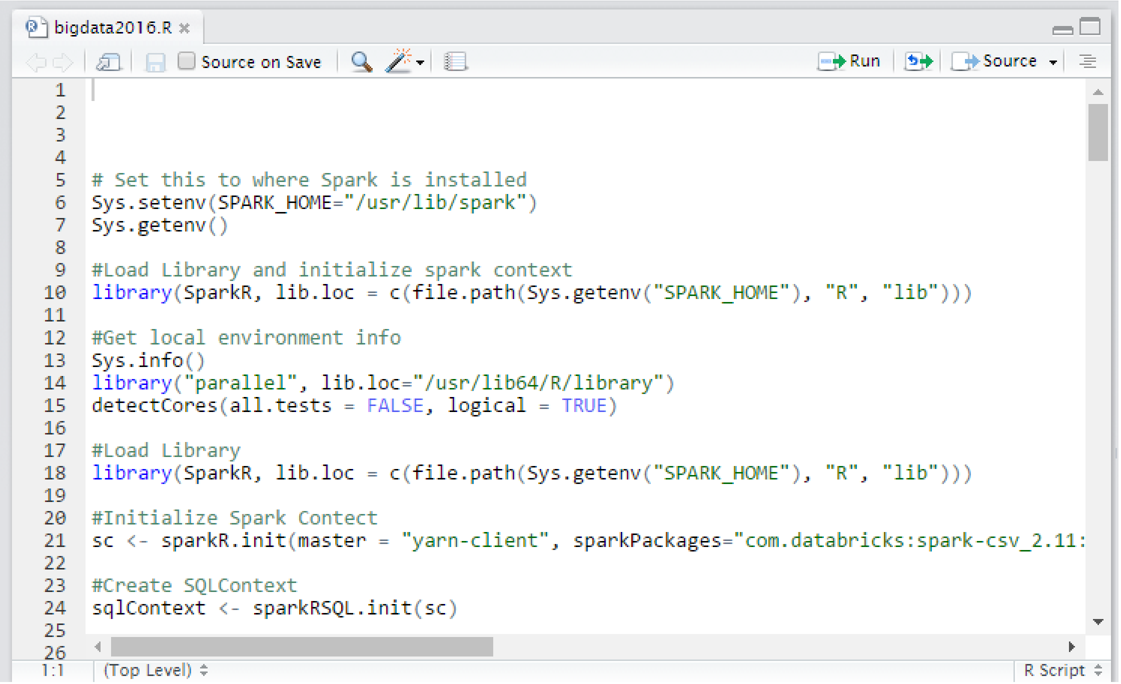
```



# Set this to where Spark is installed
Sys.setenv(SPARK_HOME="/usr/lib/spark")
Sys.getenv()

#Load Library and initialize spark context
library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib")))

#Get local environment info
Sys.info()
library("parallel", lib.loc="/usr/lib64/R/library")
detectCores(all.tests = FALSE, logical = TRUE)
```

Environment is empty

Name	Size	Modified
bigdata2016.R	7.5 KB	May 23, 2016, 11:02 PM
R		

```
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.

>
```

Environment Set Up

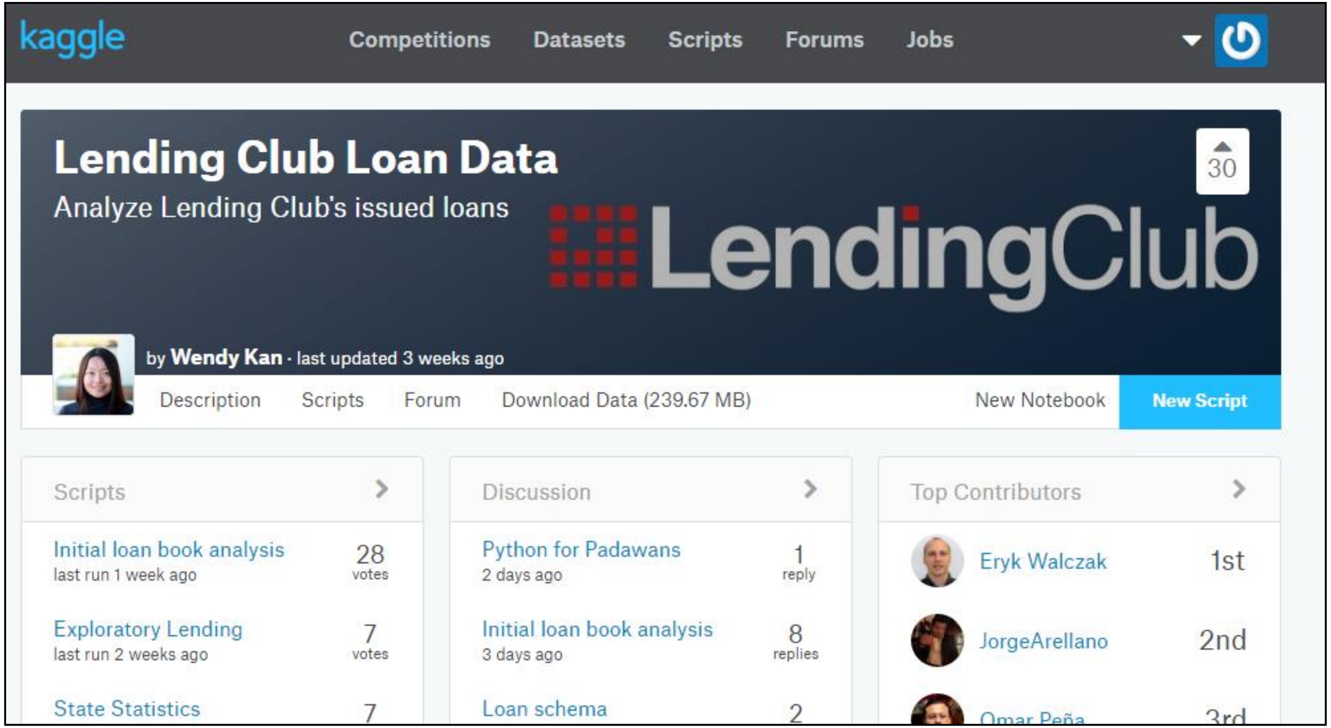


```r




# Set this to where Spark is installed
Sys.setenv(SPARK_HOME="/usr/lib/spark")
Sys.getenv()

#Load Library and initialize spark context
library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib")))

#Get local environment info
Sys.info()
library("parallel", lib.loc="/usr/lib64/R/library")
detectCores(all.tests = FALSE, logical = TRUE)

#Load Library
library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib")))

#Initialize Spark Contect
sc <- sparkR.init(master = "yarn-client", sparkPackages="com.databricks:spark-csv_2.11:

#Create SQLContext
sqlContext <- sparkRSQL.init(sc)
```

Data Analysis

NERDERY

Data Set



Lending Club Loan Data

Analyze Lending Club's issued loans

by Wendy Kan · last updated 3 weeks ago

Description Scripts Forum Download Data (239.67 MB) New Notebook New Script

Scripts

Initial loan book analysis last run 1 week ago	28 votes
Exploratory Lending last run 2 weeks ago	7 votes
State Statistics	7

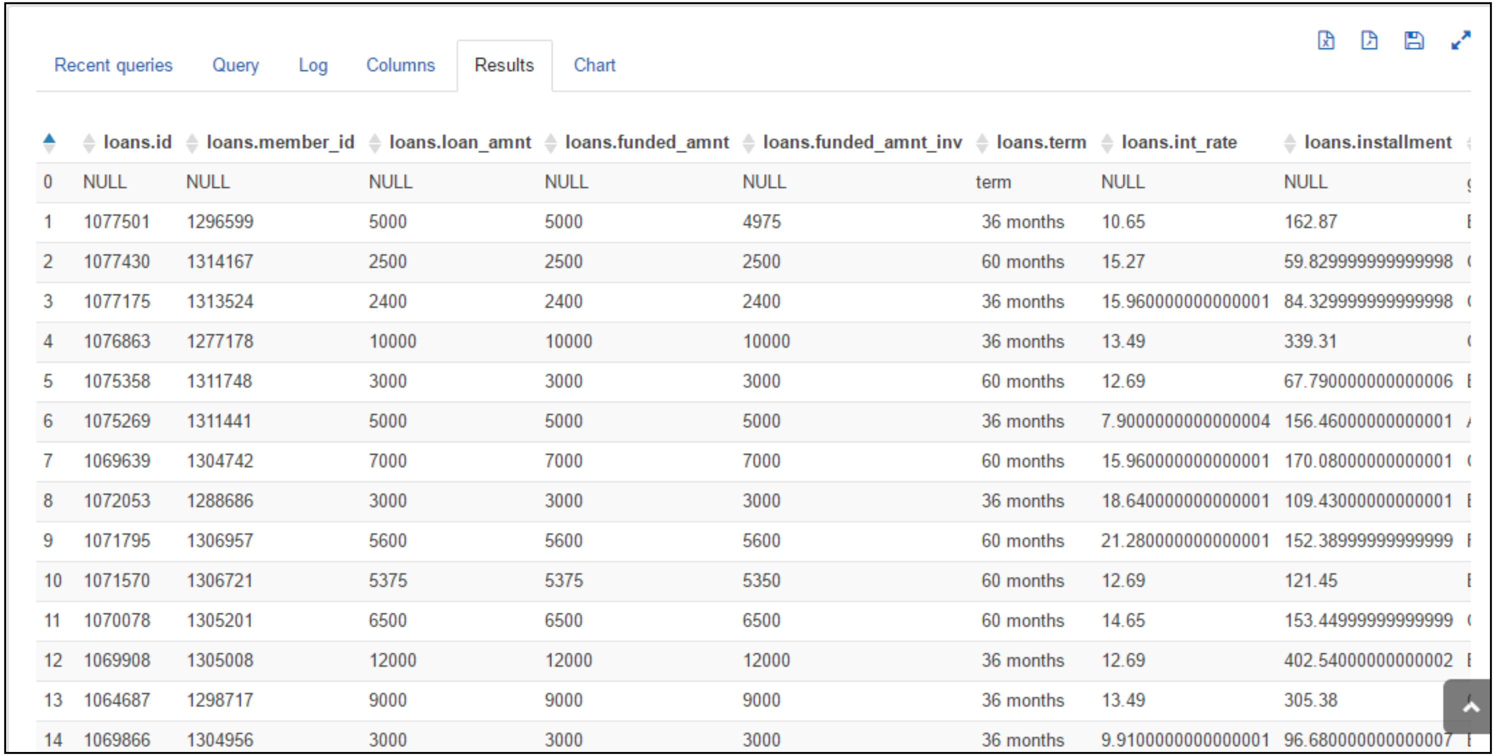
Discussion

Python for Padawans 2 days ago	1 reply
Initial loan book analysis 3 days ago	8 replies
Loan schema	2

Top Contributors

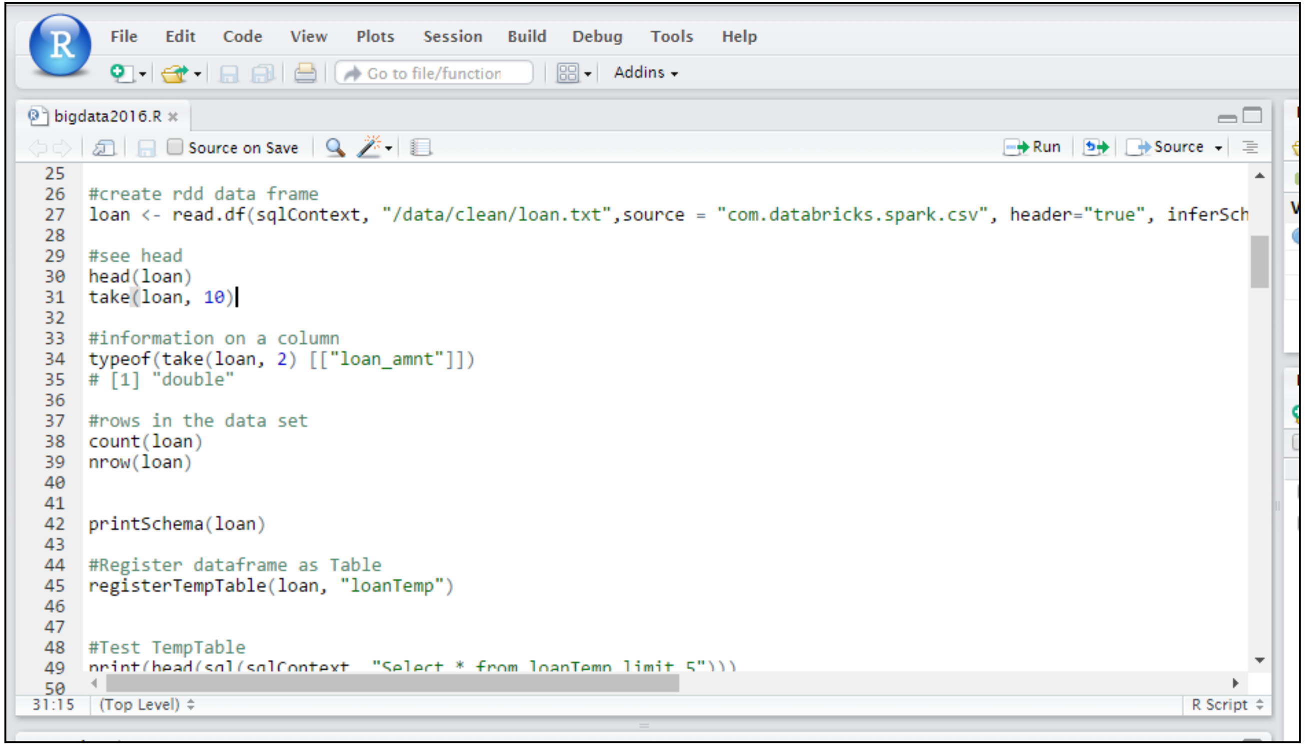
Eryk Walczak	1st
JorgeArellano	2nd
Omar Peña	3rd

Data Set



	loans.id	loans.member_id	loans.loan_amnt	loans.funded_amnt	loans.funded_amnt_inv	loans.term	loans.int_rate	loans.installment
0	NULL	NULL	NULL	NULL	NULL	term	NULL	NULL
1	1077501	1296599	5000	5000	4975	36 months	10.65	162.87
2	1077430	1314167	2500	2500	2500	60 months	15.27	59.829999999999998
3	1077175	1313524	2400	2400	2400	36 months	15.960000000000001	84.329999999999998
4	1076863	1277178	10000	10000	10000	36 months	13.49	339.31
5	1075358	1311748	3000	3000	3000	60 months	12.69	67.790000000000006
6	1075269	1311441	5000	5000	5000	36 months	7.9000000000000004	156.46000000000001
7	1069639	1304742	7000	7000	7000	60 months	15.960000000000001	170.08000000000001
8	1072053	1288686	3000	3000	3000	36 months	18.640000000000001	109.43000000000001
9	1071795	1306957	5600	5600	5600	60 months	21.280000000000001	152.38999999999999
10	1071570	1306721	5375	5375	5350	60 months	12.69	121.45
11	1070078	1305201	6500	6500	6500	60 months	14.65	153.44999999999999
12	1069908	1305008	12000	12000	12000	36 months	12.69	402.54000000000002
13	1064687	1298717	9000	9000	9000	36 months	13.49	305.38
14	1069866	1304956	3000	3000	3000	36 months	9.9100000000000001	96.680000000000007

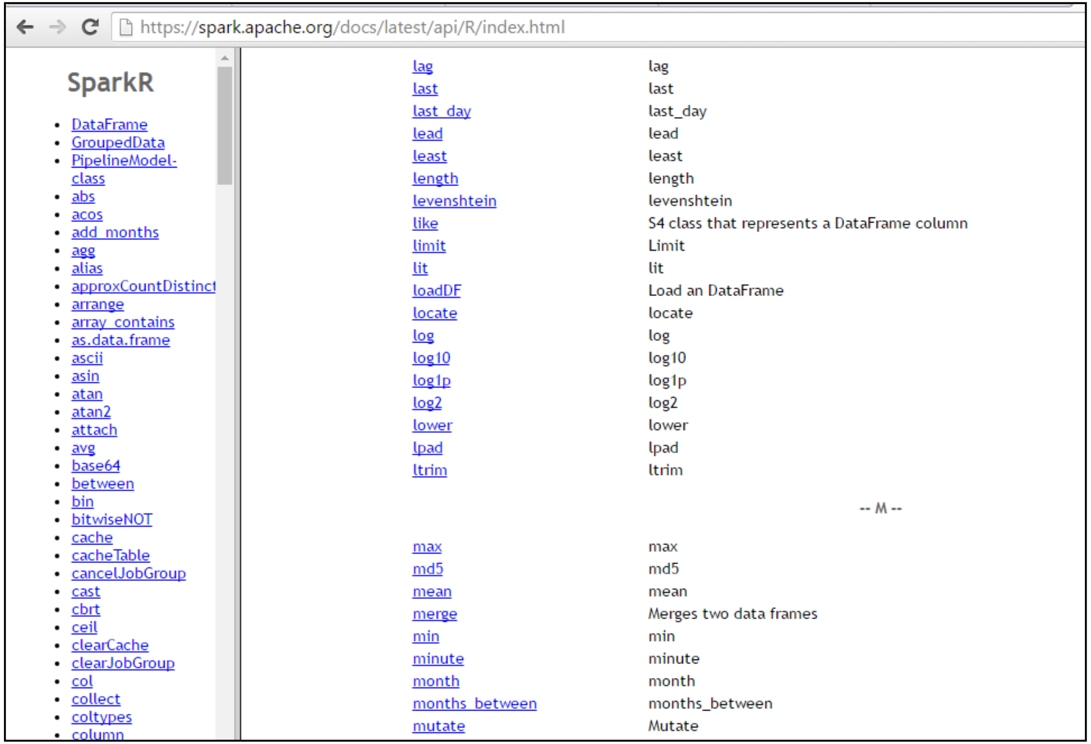
Create RDD Data Frame



```
#create rdd data frame
loan <- read.df(sqlContext, "/data/clean/loan.txt",source = "com.databricks.spark.csv", header="true", inferSch

#see head
head(loan)
take(loan, 10)

#information on a column
typeof(take(loan, 2) [["loan_amnt"]])
# [1] "double"

#rows in the data set
count(loan)
nrow(loan)


printSchema(loan)

#Register dataframe as Table
registerTempTable(loan, "loanTemp")


#Test TempTable
print(head(sql(sqlContext, "Select * from loanTemp limit 5")))
```

NERDERY

SparkR API



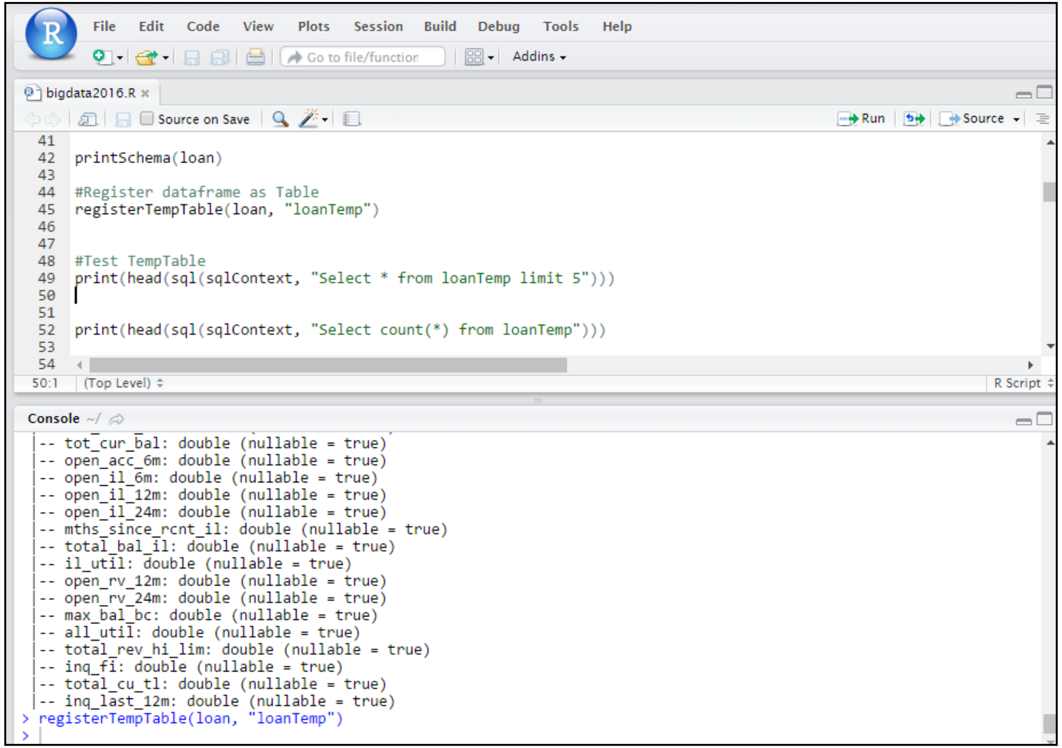
https://spark.apache.org/docs/latest/api/R/index.html

SparkR

- DataFrame
- GroupedData
- PipelineModel-class
- abs
- acos
- add_months
- agg
- alias
- approxCountDistinct
- arrange
- array_contains
- as.data.frame
- ascii
- asin
- atan
- atan2
- attach
- avg
- base64
- between
- bin
- bitwiseNOT
- cache
- cacheTable
- cancelJobGroup
- cast
- cbrt
- ceil
- clearCache
- clearJobGroup
- col
- collect
- coltypes
- column

lag	lag
last	last
last_day	last_day
lead	lead
least	least
length	length
levenshtein	levenshtein
like	S4 class that represents a DataFrame column
limit	Limit
lit	lit
loadDF	Load an DataFrame
locate	locate
log	log
log10	log10
log1p	log1p
log2	log2
lower	lower
lpad	lpad
ltrim	ltrim
	-- M --
max	max
md5	md5
mean	mean
merge	Merges two data frames
min	min
minute	minute
month	month
months_between	months_between
mutate	Mutate

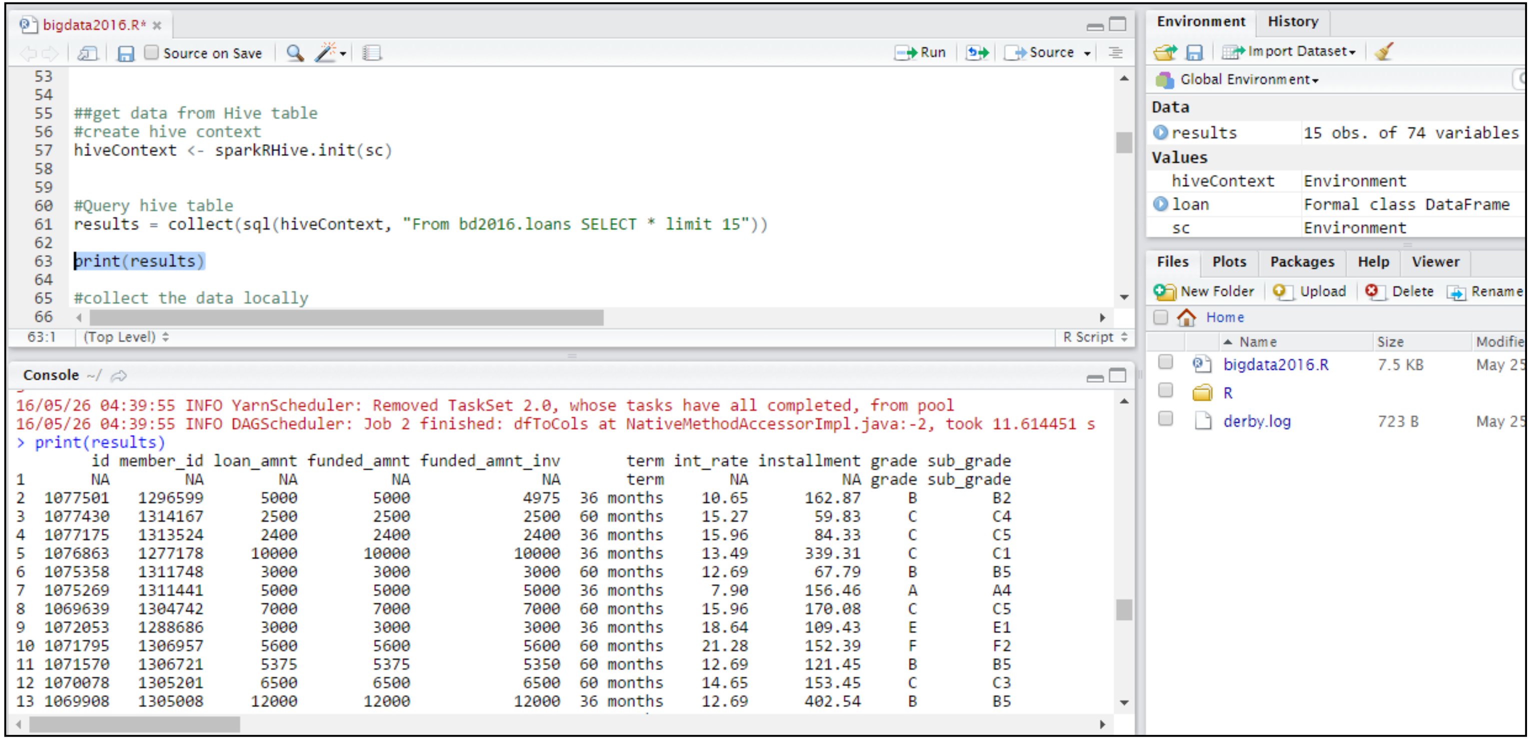
sqlContext



```r

printSchema(loan)

#Register dataframe as Table
registerTempTable(loan, "loanTemp")


#Test TempTable
print(head(sql(sqlContext, "Select * from loanTemp limit 5")))


print(head(sql(sqlContext, "Select count(*) from loanTemp")))

```

```
 |-- tot_cur_bal: double (nullable = true)
 |-- open_acc_6m: double (nullable = true)
 |-- open_il_6m: double (nullable = true)
 |-- open_il_12m: double (nullable = true)
 |-- open_il_24m: double (nullable = true)
 |-- mths_since_rcnt_il: double (nullable = true)
 |-- total_bal_il: double (nullable = true)
 |-- il_util: double (nullable = true)
 |-- open_rv_12m: double (nullable = true)
 |-- open_rv_24m: double (nullable = true)
 |-- max_bal_bc: double (nullable = true)
 |-- all_util: double (nullable = true)
 |-- total_rev_hi_lim: double (nullable = true)
 |-- inq_fi: double (nullable = true)
 |-- total_cu_tl: double (nullable = true)
 |-- inq_last_12m: double (nullable = true)
> registerTempTable(loan, "loanTemp")
> 
```

Access Hive



```r


##get data from Hive table
#create hive context
hiveContext <- sparkRHive.init(sc)


#Query hive table
results = collect(sql(hiveContext, "From bd2016.loans SELECT * limit 15"))

print(results)

#collect the data locally

```

```
16/05/26 04:39:55 INFO YarnScheduler: Removed TaskSet 2.0, whose tasks have all completed, from pool
16/05/26 04:39:55 INFO DAGScheduler: Job 2 finished: dfToCols at NativeMethodAccessorImpl.java:-2, took 11.614451 s
> print(results)
        id member_id loan_amnt funded_amnt funded_amnt_inv       term int_rate installment grade sub_grade
1       NA        NA        NA          NA              NA       term       NA          NA grade sub_grade
2  1077501   1296599      5000        5000            4975  36 months    10.65      162.87     B        B2
3  1077430   1314167      2500        2500            2500  60 months    15.27       59.83     C        C4
4  1077175   1313524      2400        2400            2400  36 months    15.96       84.33     C        C5
5  1076863   1277178     10000       10000           10000  36 months    13.49      339.31     C        C1
6  1075358   1311748      3000        3000            3000  60 months    12.69       67.79     B        B5
7  1075269   1311441      5000        5000            5000  36 months     7.90      156.46     A        A4
8  1069639   1304742      7000        7000            7000  60 months    15.96      170.08     C        C5
9  1072053   1288686      3000        3000            3000  36 months    18.64      109.43     E        E1
10 1071795   1306957      5600        5600            5600  60 months    21.28      152.39     F        F2
11 1071570   1306721      5375        5375            5350  60 months    12.69      121.45     B        B5
12 1070078   1305201      6500        6500            6500  60 months    14.65      153.45     C        C3
13 1069908   1305008     12000       12000           12000  36 months    12.69      402.54     B        B5
```

Data

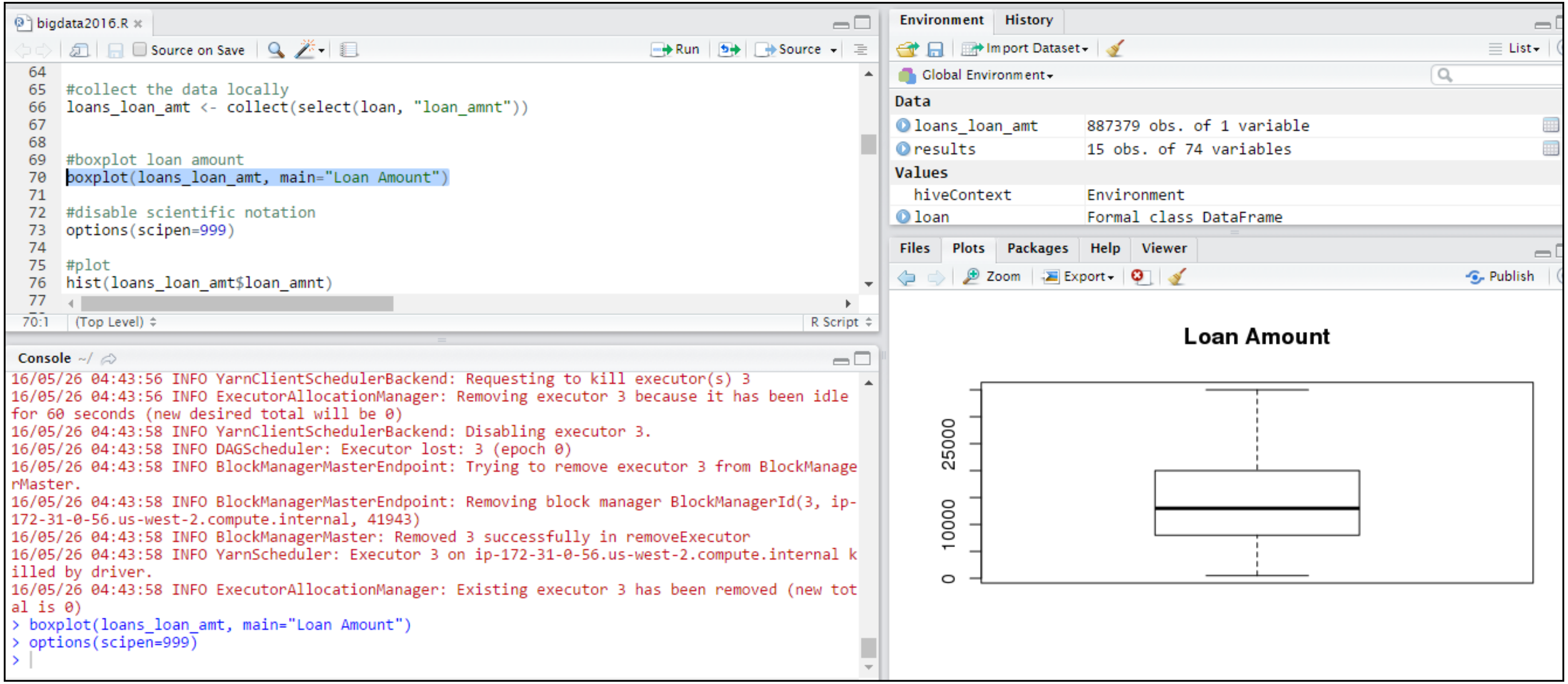
results	15 obs. of 74 variables

Values

hiveContext	Environment
loan	Formal class DataFrame
sc	Environment

Name	Size	Modifie
bigdata2016.R	7.5 KB	May 25
R		
derby.log	723 B	May 25

Full Power of R

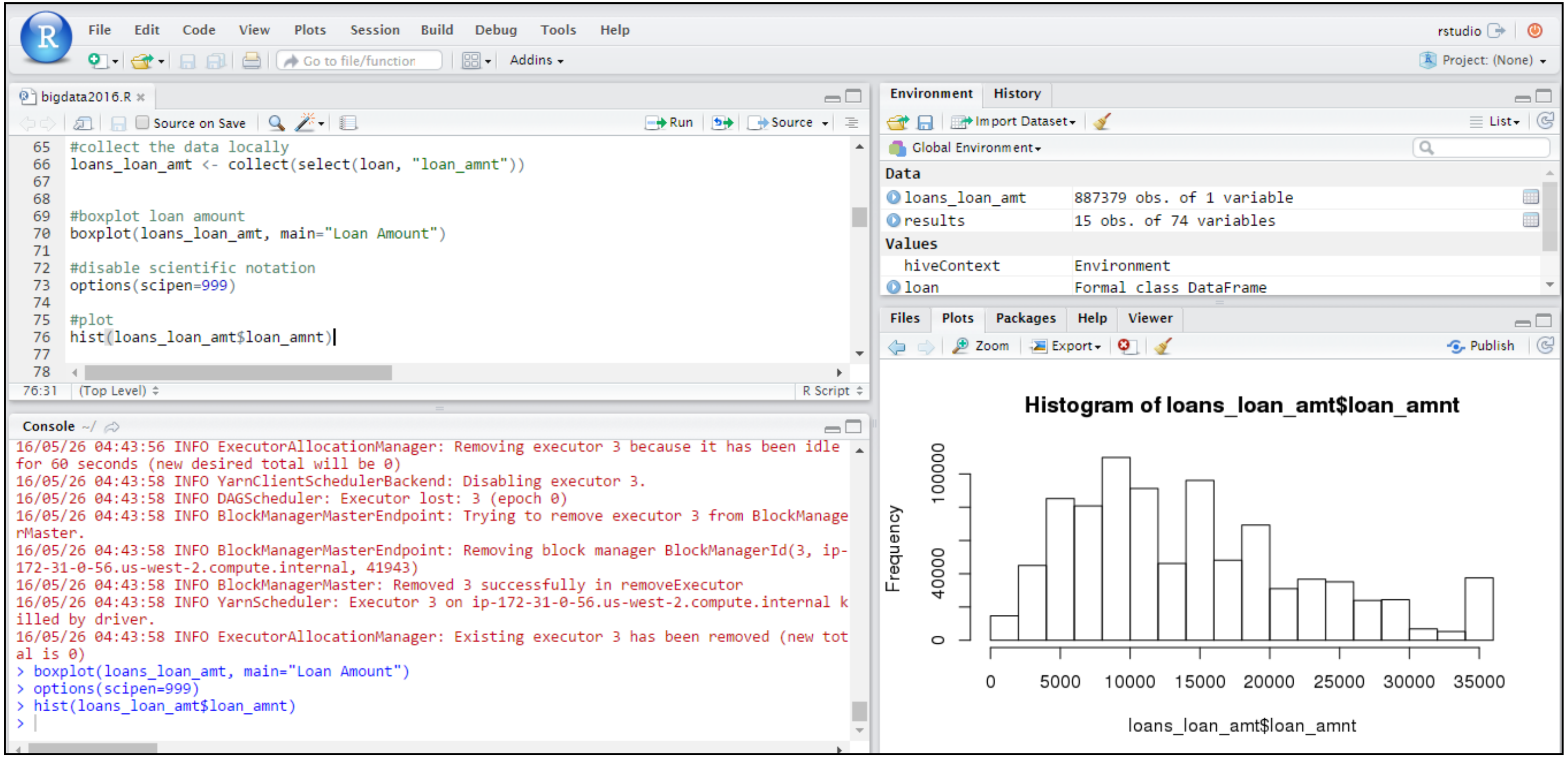


```r

#collect the data locally
loans_loan_amt <- collect(select(loan, "loan_amnt"))


#boxplot loan amount
boxplot(loans_loan_amt, main="Loan Amount")

#disable scientific notation
options(scipen=999)

#plot
hist(loans_loan_amt$loan_amnt)

```

```
16/05/26 04:43:56 INFO YarnClientSchedulerBackend: Requesting to kill executor(s) 3
16/05/26 04:43:56 INFO ExecutorAllocationManager: Removing executor 3 because it has been idle
for 60 seconds (new desired total will be 0)
16/05/26 04:43:58 INFO YarnClientSchedulerBackend: Disabling executor 3.
16/05/26 04:43:58 INFO DAGScheduler: Executor lost: 3 (epoch 0)
16/05/26 04:43:58 INFO BlockManagerMasterEndpoint: Trying to remove executor 3 from BlockManage
rMaster.
16/05/26 04:43:58 INFO BlockManagerMasterEndpoint: Removing block manager BlockManagerId(3, ip-
172-31-0-56.us-west-2.compute.internal, 41943)
16/05/26 04:43:58 INFO BlockManagerMaster: Removed 3 successfully in removeExecutor
16/05/26 04:43:58 INFO YarnScheduler: Executor 3 on ip-172-31-0-56.us-west-2.compute.internal k
illed by driver.
16/05/26 04:43:58 INFO ExecutorAllocationManager: Existing executor 3 has been removed (new tot
al is 0)
> boxplot(loans_loan_amt, main="Loan Amount")
> options(scipen=999)
> 
```

Data	
loans_loan_amt	887379 obs. of 1 variable
results	15 obs. of 74 variables
Values	
hiveContext	Environment
loan	Formal class DataFrame

Full Power of R

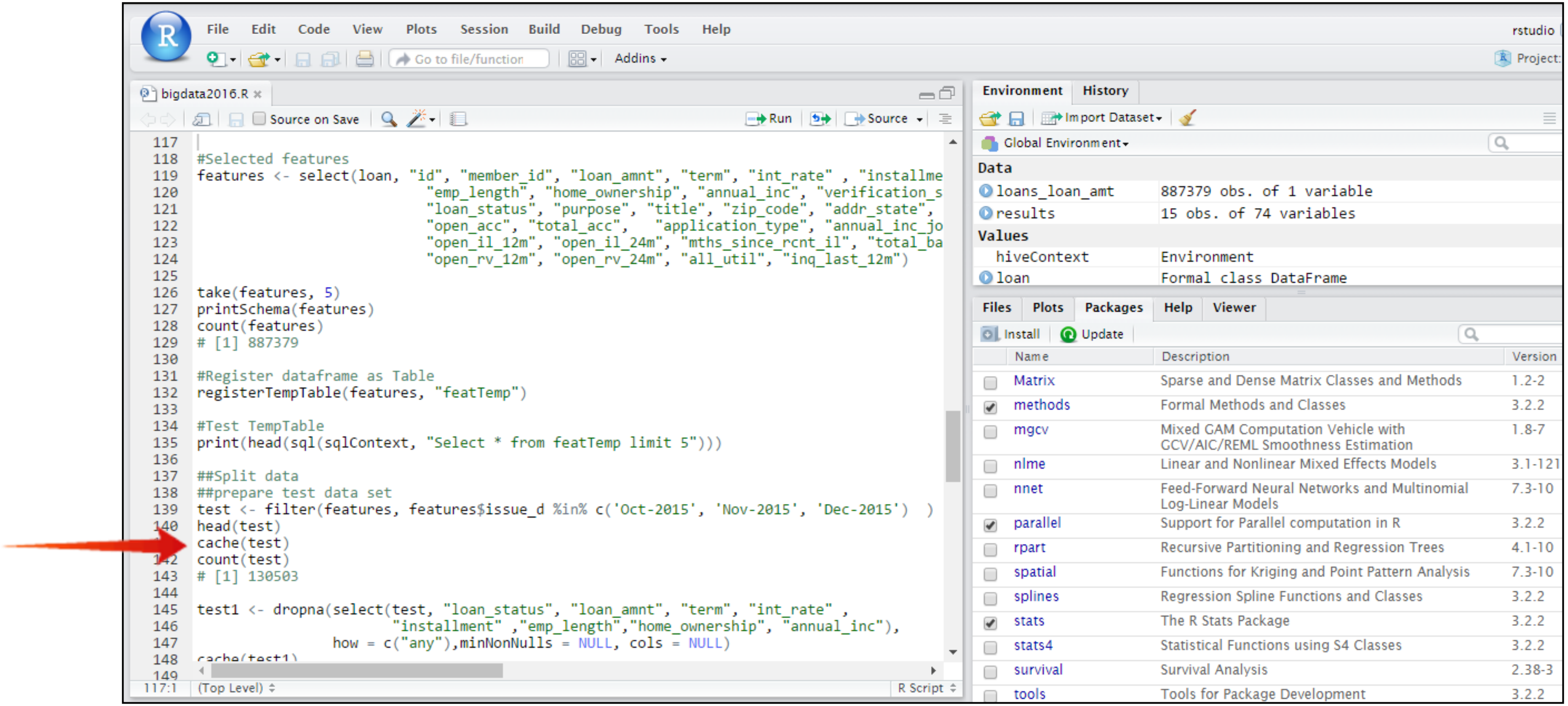


```r
#collect the data locally
loans_loan_amt <- collect(select(loan, "loan_amnt"))


#boxplot loan amount
boxplot(loans_loan_amt, main="Loan Amount")

#disable scientific notation
options(scipen=999)

#plot
hist(loans_loan_amt$loan_amnt)


```

```
16/05/26 04:43:56 INFO ExecutorAllocationManager: Removing executor 3 because it has been idle
for 60 seconds (new desired total will be 0)
16/05/26 04:43:58 INFO YarnClientSchedulerBackend: Disabling executor 3.
16/05/26 04:43:58 INFO DAGScheduler: Executor lost: 3 (epoch 0)
16/05/26 04:43:58 INFO BlockManagerMasterEndpoint: Trying to remove executor 3 from BlockManage
rMaster.
16/05/26 04:43:58 INFO BlockManagerMasterEndpoint: Removing block manager BlockManagerId(3, ip-
172-31-0-56.us-west-2.compute.internal, 41943)
16/05/26 04:43:58 INFO BlockManagerMaster: Removed 3 successfully in removeExecutor
16/05/26 04:43:58 INFO YarnScheduler: Executor 3 on ip-172-31-0-56.us-west-2.compute.internal k
illed by driver.
16/05/26 04:43:58 INFO ExecutorAllocationManager: Existing executor 3 has been removed (new tot
al is 0)
> boxplot(loans_loan_amt, main="Loan Amount")
> options(scipen=999)
> hist(loans_loan_amt$loan_amnt)
> 
```

```
Data
loans_loan_amt    887379 obs. of 1 variable
results           15 obs. of 74 variables
Values
hiveContext       Environment
loan              Formal class DataFrame
```

SparkR: In Memory Processing



```

#Selected features
features <- select(loan, "id", "member_id", "loan_amnt", "term", "int_rate" , "installme
                          "emp_length", "home_ownership", "annual_inc", "verification_s
                          "loan_status", "purpose", "title", "zip_code", "addr_state",
                          "open_acc", "total_acc",   "application_type", "annual_inc_jo
                          "open_il_12m", "open_il_24m", "mths_since_rcnt_il", "total_ba
                          "open_rv_12m", "open_rv_24m", "all_util", "inq_last_12m")

take(features, 5)
printSchema(features)
count(features)
# [1] 887379

#Register dataframe as Table
registerTempTable(features, "featTemp")

#Test TempTable
print(head(sql(sqlContext, "Select * from featTemp limit 5")))

##Split data
##prepare test data set
test <- filter(features, features$issue_d %in% c('Oct-2015', 'Nov-2015', 'Dec-2015')  )
head(test)
cache(test)
count(test)
# [1] 130503

test1 <- dropna(select(test, "loan_status", "loan_amnt", "term", "int_rate" ,
                       "installment" ,"emp_length","home_ownership", "annual_inc"),
                how = c("any"),minNonNulls = NULL, cols = NULL)
cache(test1)
```

Data

loans_loan_amt	887379 obs. of 1 variable
results	15 obs. of 74 variables

Values

hiveContext	Environment
loan	Formal class DataFrame

	Name	Description	Version
□	Matrix	Sparse and Dense Matrix Classes and Methods	1.2-2
☑	methods	Formal Methods and Classes	3.2.2
□	mgcv	Mixed GAM Computation Vehicle with GCV/AIC/REML Smoothness Estimation	1.8-7
□	nlme	Linear and Nonlinear Mixed Effects Models	3.1-121
□	nnet	Feed-Forward Neural Networks and Multinomial Log-Linear Models	7.3-10
☑	parallel	Support for Parallel computation in R	3.2.2
□	rpart	Recursive Partitioning and Regression Trees	4.1-10
□	spatial	Functions for Kriging and Point Pattern Analysis	7.3-10
□	splines	Regression Spline Functions and Classes	3.2.2
☑	stats	The R Stats Package	3.2.2
□	stats4	Statistical Functions using S4 Classes	3.2.2
□	survival	Survival Analysis	2.38-3
□	tools	Tools for Package Development	3.2.2

NERDERY

Machine Learning

SparkR

Limitations

- Machine Learning limited to glm
- Distributed processing constrained by existing API.

Future

✓ Databricks

✓ Alteryx

Machine Learning in R*

1. **e1071** Functions for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, naive Bayes classifier etc *(142479 downloads)*
2. **rpart** Recursive Partitioning and Regression Trees. (135390)
3. **igraph** A collection of network analysis tools. (122930)
4. **nnet** Feed-forward Neural Networks and Multinomial Log-Linear Models. (108298)
5. **randomForest** Breiman and Cutler's random forests for classification and regression. (105375)
6. **caret** package (short for Classification And Regression Training) is a set of functions that attempt to streamline the process for creating predictive models. (87151)
7. **kernlab** Kernel-based Machine Learning Lab. (62064)
8. **glmnet** Lasso and elastic-net regularized generalized linear models. (56948)
9. **ROCR** Visualizing the performance of scoring classifiers. (51323)
10. **gbm** Generalized Boosted Regression Models. (44760)
11. **party** A Laboratory for Recursive Partitioning. (43290)
12. **arules** Mining Association Rules and Frequent Itemsets. (39654)
13. **tree** Classification and regression trees. (27882)
14. **klaR** Classification and visualization. (27828)
15. **RWeka** R/Weka interface. (26973)
16. **ipred** Improved Predictors. (22358)
17. **lars** Least Angle Regression, Lasso and Forward Stagewise. (19691)
18. **earth** Multivariate Adaptive Regression Spline Models. (15901)
19. **CORElearn** Classification, regression, feature evaluation and ordinal evaluation. (13856)
20. **mboost** Model-Based Boosting. (13078)

* KDnuggets Top 20 R Machine Learning Packages and Data Science Packages.
Source: http://www.kdnuggets.com/2015/06/top-20-r-machine-learning-packages.html

Machine Learning in Spark

- Classification and regression
  - linear models (SVMs, logistic regression, linear regression)
  - naive Bayes
  - decision trees
  - ensembles of trees (Random Forests and Gradient-Boosted Trees)
  - isotonic regression
- Collaborative filtering
  - alternating least squares (ALS)
- Clustering
  - k-means
  - Gaussian mixture
  - power iteration clustering (PIC)
  - latent Dirichlet allocation (LDA)
  - bisecting k-means
  - streaming k-means
- Dimensionality reduction
  - singular value decomposition (SVD)
  - principal component analysis (PCA)
- Feature extraction and transformation
- Frequent pattern mining
  - FP-growth
  - association rules
  - PrefixSpan
- Evaluation metrics
- PMML model export
- Optimization (developer)
  - stochastic gradient descent
  - limited-memory BFGS (L-BFGS)

Resources

Slides: 🐦 @the_nerdery

Code: https://github.com/thenerdery/SparkRTalk

Questions?

NERDERY

Contact

The Nerdery
info@nerdery.com
(877) 664.6373