



NERDERY®

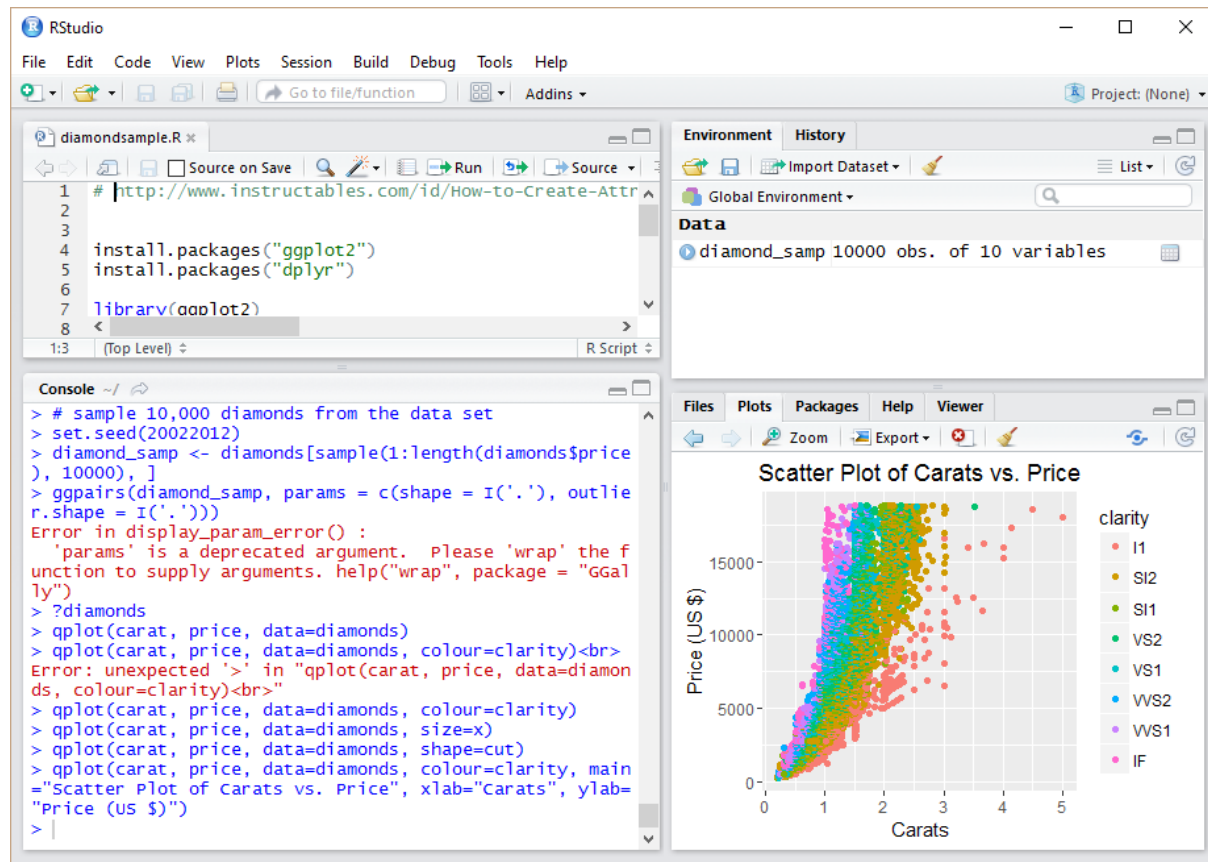
R Studio Server on Amazon EMR

Chad Dvoracek

Brandon Veber



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





EMR (Elastic MapReduce)

Managed Hadoop Framework

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

AWS Services Edit

Elastic MapReduce Create Cluster

Create Cluster - Quick Options [Go to advanced options](#)

General Configuration

Cluster name

☒ Logging ⓘ

S3 folder

Launch mode ☒ Cluster ⓘ ☐ Step execution ⓘ

Software configuration

Vendor ☒ Amazon ☐ MapR

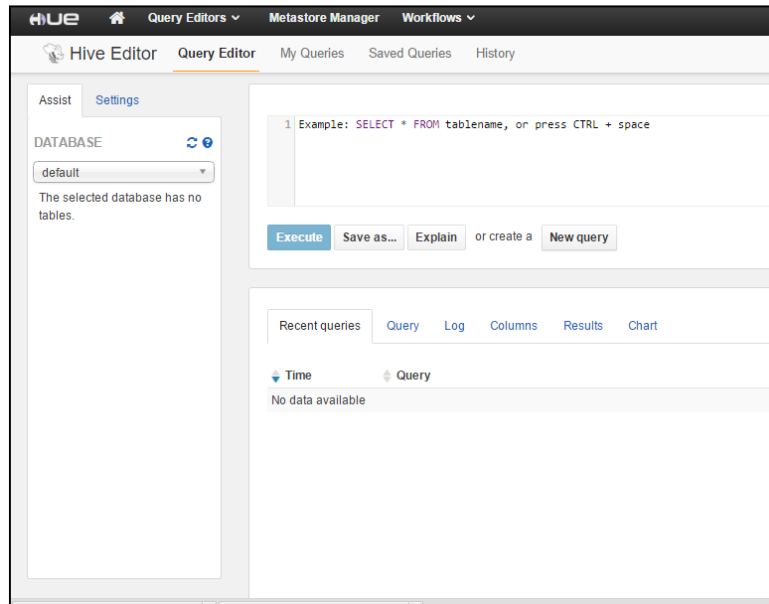
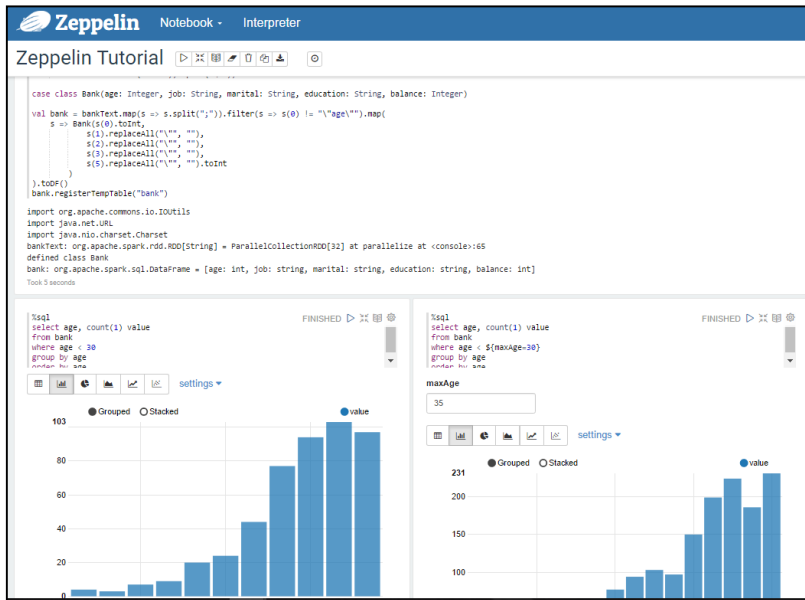
Release ⓘ

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



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 too 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



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?




EMR Set Up

Quick & Simple

Simple Start Up Script

```
1  #/bin/bash
2
3  USER="rstudio"
4  USERPW="rstudio"
5
6
7  # create rstudio user on all machines
8  # we need a unix user with home directory
9  # and password and hadoop permission
10 sudo adduser $USER
11 sudo sh -c "echo '$USERPW' | passwd $USER --stdin"
12
13 # fix hadoop tmp permission on all machines
14 sudo chmod 777 -R /mnt/var/lib/hadoop/tmp
```


EMR Configuration: Step 1

 AWS

Services

Edit

Elastic MapReduce

Create Cluster

Create Cluster - Advanced Options [Go to quick options](#)

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

Software Configuration

Vendor ☒ Amazon ☐ MapR

Release

☒ 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

Configure

☐ Auto-terminate cluster after the last step is completed


Cancel

Next

 NERDERY

16

EMR Configuration: Step 2

 AWS

Services

Edit

Chad A DvoracekOregonSupport

Elastic MapReduceCreate ClusterEMR Help

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 [Create a VPC ⓘ](#)

EC2 Subnet

Type	Name	EC2 instance type	Instance count	Storage per instance	Request spot	Bid price		
Master	<input type="text" value="Master instance group - 1"/>	<input type="text" value="m3.xlarge"/>	<input type="text" value="1"/>	80 GiB Add EBS volumes	<input checked="" type="checkbox"/>	<input type="text" value=".10"/>	ⓘ	?
Core	<input type="text" value="Core instance group - 2"/>	<input type="text" value="m3.xlarge"/>	<input type="text" value="2"/>	80 GiB Add EBS volumes	<input checked="" type="checkbox"/>	<input type="text" value=".10"/>	ⓘ	?
Task	<input type="text" value="Task instance group - 3"/>	<input type="text" value="m3.xlarge"/>	<input type="text" value="2"/>	80 GiB Add EBS volumes	<input checked="" type="checkbox"/>	<input type="text" value=".10"/>	ⓘ	✕ ?


Add task instance group

Cancel

Previous

Next

EMR Configuration: Step 2

 AWS

Services

Edit

Chad A DvoracekOregonSupport

Elastic MapReduceCreate ClusterEMR Help

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 [Create a VPC ⓘ](#)

EC2 Subnet

Type	Name	EC2 instance type	Instance count	Storage per instance	Request spot	Bid price		
Master	<input type="text" value="Master instance group - 1"/>	<input type="text" value="m3.xlarge"/>	<input type="text" value="1"/>	80 GiB Add EBS volumes	<input checked="" type="checkbox"/>	<input type="text" value=".10"/>	ⓘ	?
Core	<input type="text" value="Core instance group - 2"/>	<input type="text" value="m3.xlarge"/>	<input type="text" value="2"/>	80 GiB Add EBS volumes	<input checked="" type="checkbox"/>	<input type="text" value=".10"/>	ⓘ	?
Task	<input type="text" value="Task instance group - 3"/>	<input type="text" value="m3.xlarge"/>	<input type="text" value="2"/>	80 GiB Add EBS volumes	<input checked="" type="checkbox"/>	<input type="text" value=".10"/>	ⓘ	✕ ?

Add task instance group

Cancel

Previous

Next

EMR Configuration: Step 3

[Step 1: Software and Steps](#)

[Step 2: Hardware](#)

Step 3: General Cluster Settings

[Step 4: Security](#)

General Options

Cluster name

☒ Logging ⓘ

S3 folder

☒ Debugging ⓘ

☐ Termination protection ⓘ

Tags ⓘ

Key	Value (optional)	
Name	sparkr2	✕
<i>Add a key to create a tag</i>		

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

EMR Configuration: Step 4



AWS

Services

Edit

Elastic MapReduce>Create Cluster

Create Cluster - Advanced Options [Go to quick options](#)

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

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

▶ EC2 Security Groups


▶ Encryption Options

Cancel

Previous

Create cluster


EMR Web Connection

 **AWS** ▾ **Services** ▾ **Edit** ▾ Chad A Dvoracek ▾ Oregon ▾ Support

Elastic MapReduce ▾ [Cluster List](#) > Cluster Details EMR H


[Add step](#) [Resize](#) [Clone](#) [Terminate](#) [AWS CLI export](#)

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	Configuration Details	Network and Hardware	Security and Access
ID: j-1GNP7L46397GV	Release label: emr-4.6.0	Availability zone: us-west-2c	Key name: train1
Creation date: 2016-05-24 21:47 (UTC-5)	Hadoop distribution: Amazon 2.7.2	Subnet ID: subnet-1c5f755a	EC2 instance profile: EMR_EC2_DefaultRole
Elapsed time: 42 minutes	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	Master: Running 1 m3.xlarge (Spot: .10)	EMR role: EMR_DefaultRole
Auto-terminate: No		Core: Running 2 m3.xlarge (Spot: .10)	Visible to all users: Change
Termination protection: Off Change		Task: Running 2 m3.xlarge (Spot: .10)	Security groups for Master: sg-7da1c419 (ElasticMapReduce-master)
	Log URI: s3://cdsparkr/ 		Security groups for Core & Task: sg-7ea1c41a (ElasticMapReduce-slave)
	EMRFS consistent view: Disabled		

EMR Web Connection

AWS Services Edit Chad A Dvoracek Oregon Support

Elastic MapReduce Cluster List Cluster Details EMR Help

Add step Resize Clone Terminate AWS CLI export

Cluster: sparkr2 **Waiting** Cluster ready after last step completed.

C

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: Disabled

consistent view:

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)

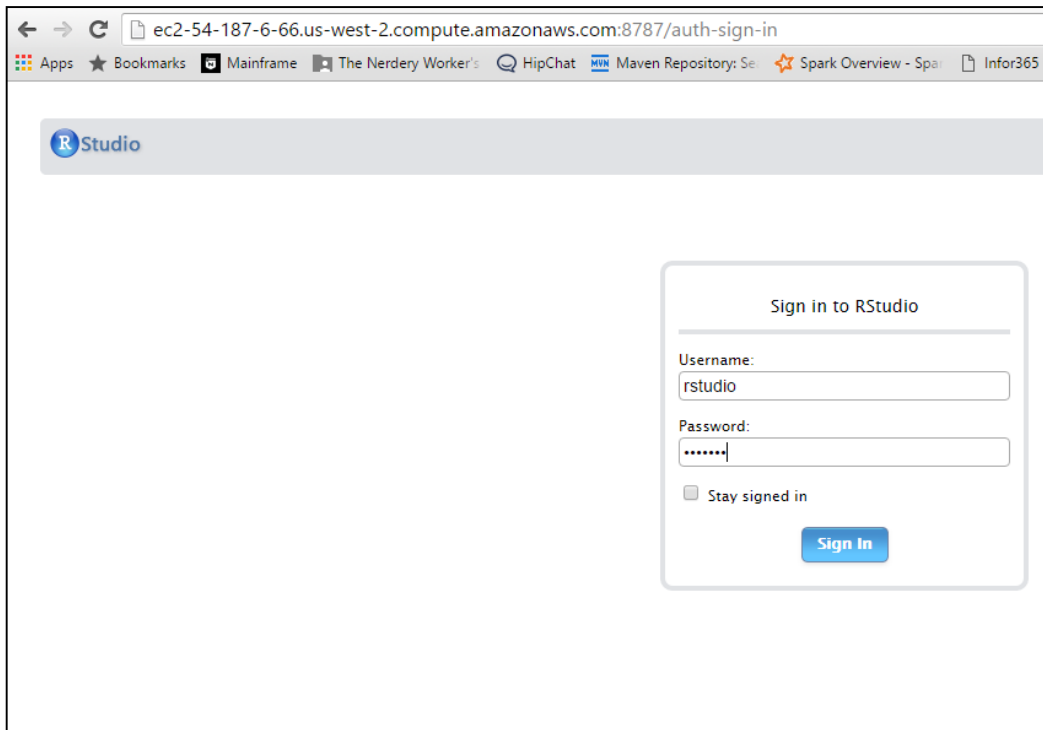
Final Step

```
8 #SSH Into the EMR Cluster
9
10 #Add rstudio user to hadoop group
11 sudo usermod -a -G hadoop rstudio
12
13
14 #Install R Studio Server
15 wget https://download2.rstudio.org/rstudio-server-rhel-0.99.902-x86_64.rpm
16 sudo yum install --nogpgcheck rstudio-server-rhel-0.99.902-x86_64.rpm
17 sudo rstudio-server verify-installation
18
19 ## Open Web Browser
20 ## Accessing the Server
21 http://<server-ip>:8787
22 example: http://ec2-54-187-11-218.us-west-2.compute.amazonaws.com:8787/
23 #Username: rstudio
24 #Password: rstudio
25
```

R Studio Server

R and SparkR

R Studio Server: Sign In



A screenshot of a web browser displaying the RStudio Server sign-in page. The browser's address bar shows the URL `ec2-54-187-6-66.us-west-2.compute.amazonaws.com:8787/auth-sign-in`. The browser's bookmark bar includes links for 'Apps', 'Bookmarks', 'Mainframe', 'The Nerder Worker's', 'HipChat', 'Maven Repository: Se', 'Spark Overview - Spa', and 'Infor365'. The page header features the RStudio logo. The main content area contains a sign-in form titled 'Sign in to RStudio'. The form has two input fields: 'Username:' with the value 'rstudio' and 'Password:' with masked characters '.....'. Below the password field is a checkbox labeled 'Stay signed in'. A blue 'Sign In' button is positioned at the bottom right of the form.

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

Apps ★ Bookmarks Mainframe The Nerder Worker's HipChat Maven Repository: Se Spark Overview - Spa Infor365

RStudio

Sign in to RStudio

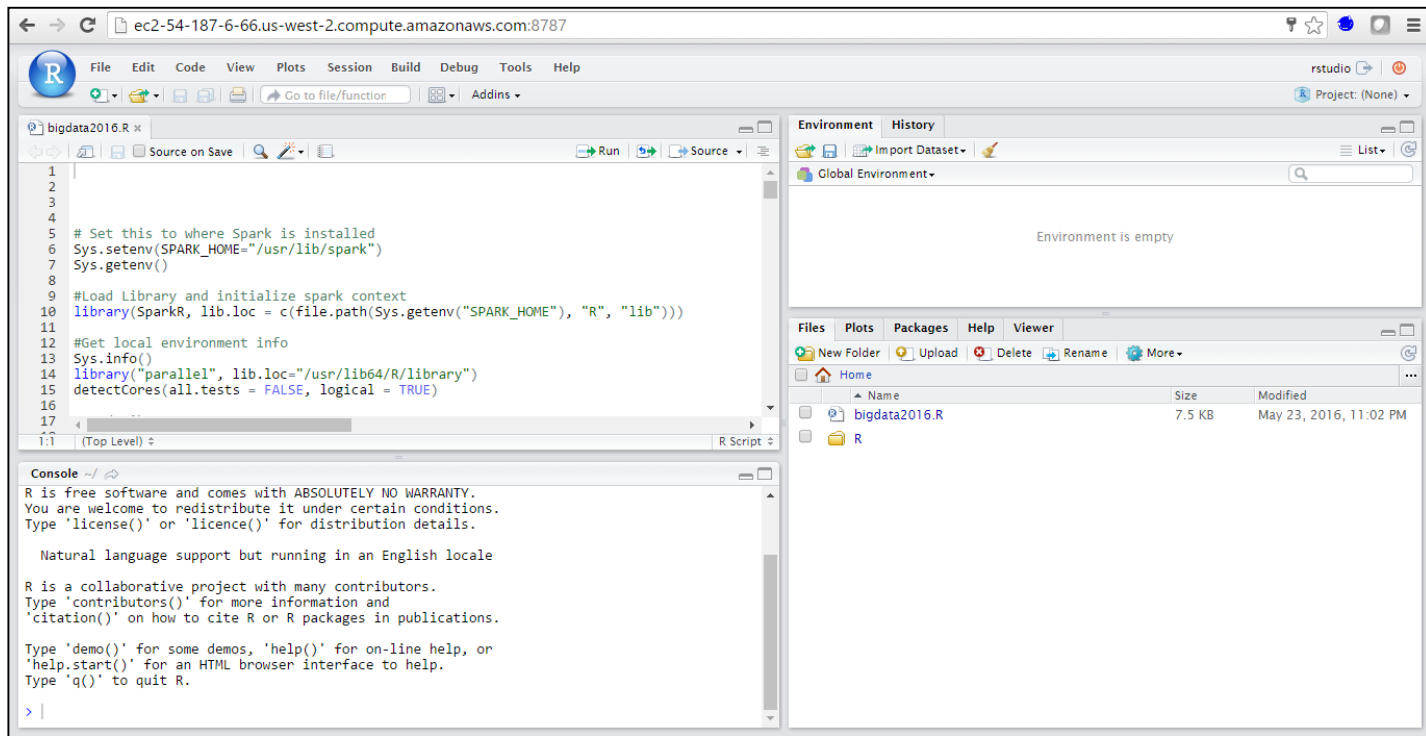
Username:
rstudio

Password:
.....

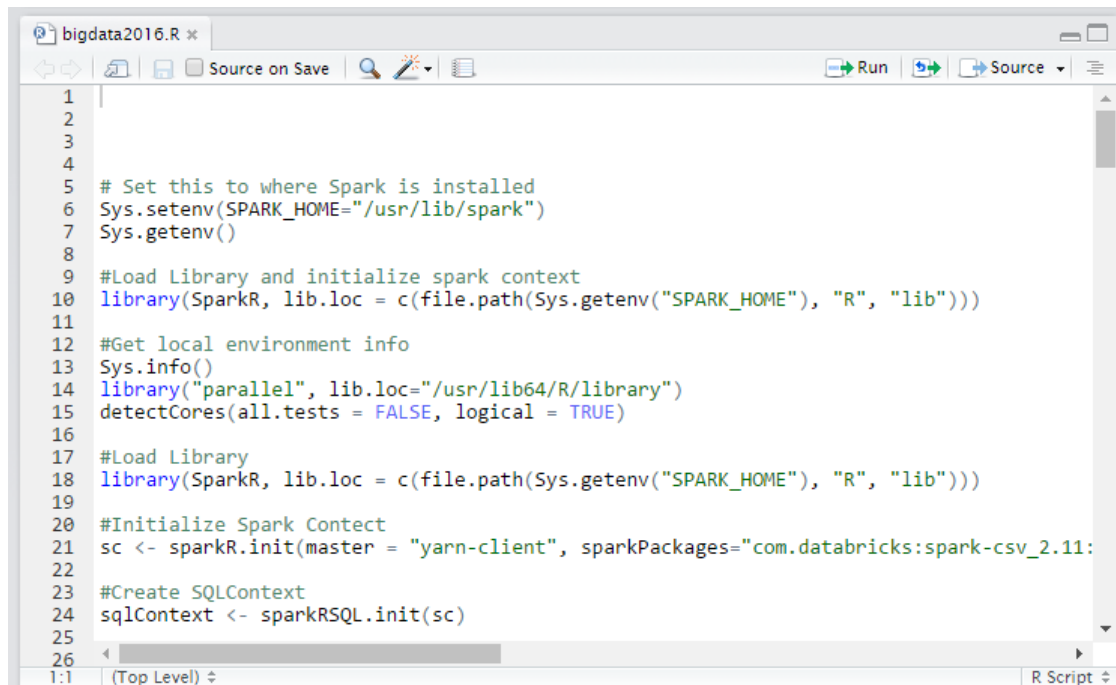
☐ Stay signed in

Sign In

R Studio Server



Environment Set Up



```
1  
2  
3  
4  
5 # Set this to where Spark is installed  
6 Sys.setenv(SPARK_HOME="/usr/lib/spark")  
7 Sys.getenv()  
8  
9 #Load Library and initialize spark context  
10 library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib")))  
11  
12 #Get local environment info  
13 Sys.info()  
14 library("parallel", lib.loc="/usr/lib64/R/library")  
15 detectCores(all.tests = FALSE, logical = TRUE)  
16  
17 #Load Library  
18 library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib")))  
19  
20 #Initialize Spark Context  
21 sc <- sparkR.init(master = "yarn-client", sparkPackages="com.databricks:spark-csv_2.11:  
22  
23 #Create SQLContext  
24 sqlContext <- sparkRSQL.init(sc)  
25  
26  
1:1 | (Top Level) ↕ R Script ↕
```

new strong performance was achieved by the year



Data Analysis

Data Set

The screenshot shows the Kaggle website interface for the 'Lending Club Loan Data' dataset. The header includes the Kaggle logo and navigation links for Competitions, Datasets, Scripts, Forums, and Jobs. The dataset title 'Lending Club Loan Data' is prominently displayed, along with the subtitle 'Analyze Lending Club's issued loans' and the Lending Club logo. The dataset is credited to Wendy Kan, last updated 3 weeks ago. Below the title, there are tabs for Description, Scripts, Forum, Download Data (239.67 MB), New Notebook, and New Script. The main content area is divided into three sections: Scripts, Discussion, and Top Contributors. The Scripts section lists 'Initial loan book analysis' (28 votes), 'Exploratory Lending' (7 votes), and 'State Statistics' (7 votes). The Discussion section lists 'Python for Padawans' (1 reply), 'Initial loan book analysis' (8 replies), and 'Loan schema' (2 replies). The Top Contributors section lists Eryk Walczak (1st), Jorge Arellano (2nd), and Omar Peña (3rd).

Lending Club Loan Data
Analyze Lending Club's issued loans

by **Wendy Kan** · last updated 3 weeks ago

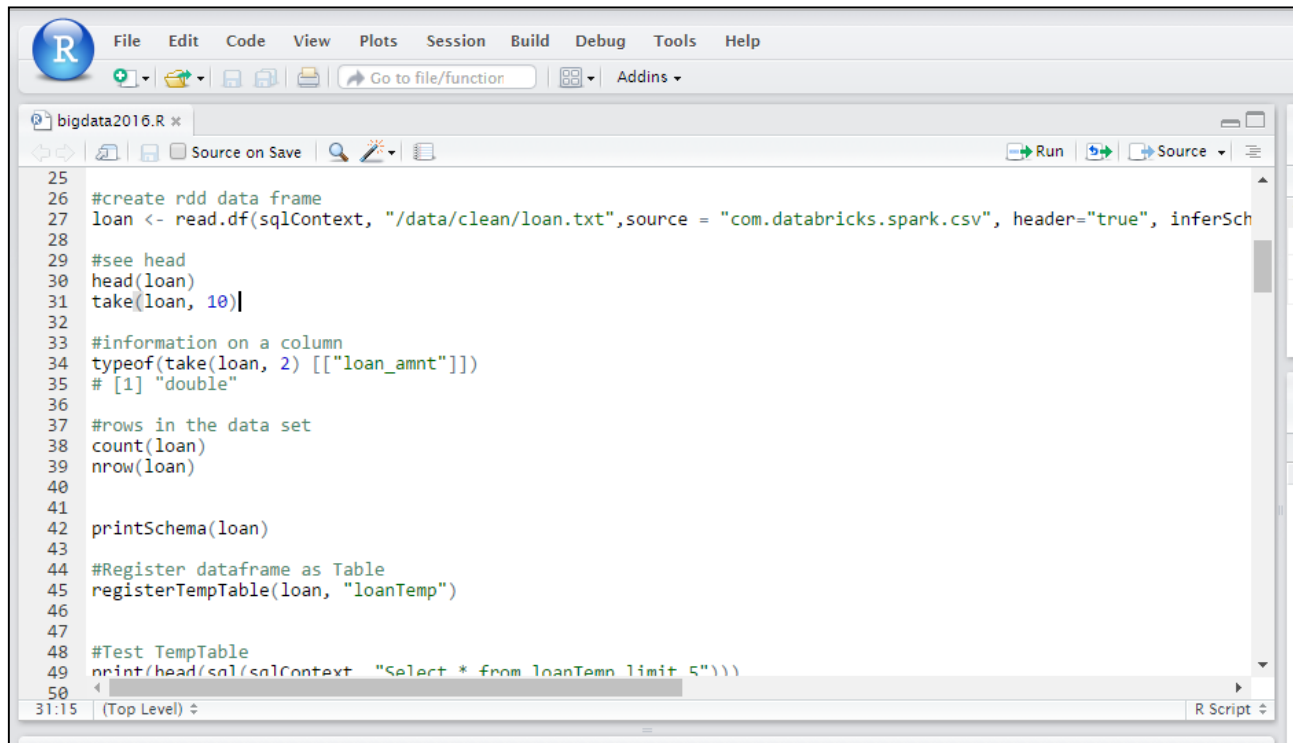
Description Scripts Forum Download Data (239.67 MB) New Notebook **New Script**

Scripts	Discussion	Top Contributors
Initial loan book analysis last run 1 week ago 28 votes	Python for Padawans 2 days ago 1 reply	Eryk Walczak 1st
Exploratory Lending last run 2 weeks ago 7 votes	Initial loan book analysis 3 days ago 8 replies	Jorge Arellano 2nd
State Statistics 7 votes	Loan schema 2 replies	Omar Peña 3rd

Data Set

Recent queries Query Log Columns Results Chart									
	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	

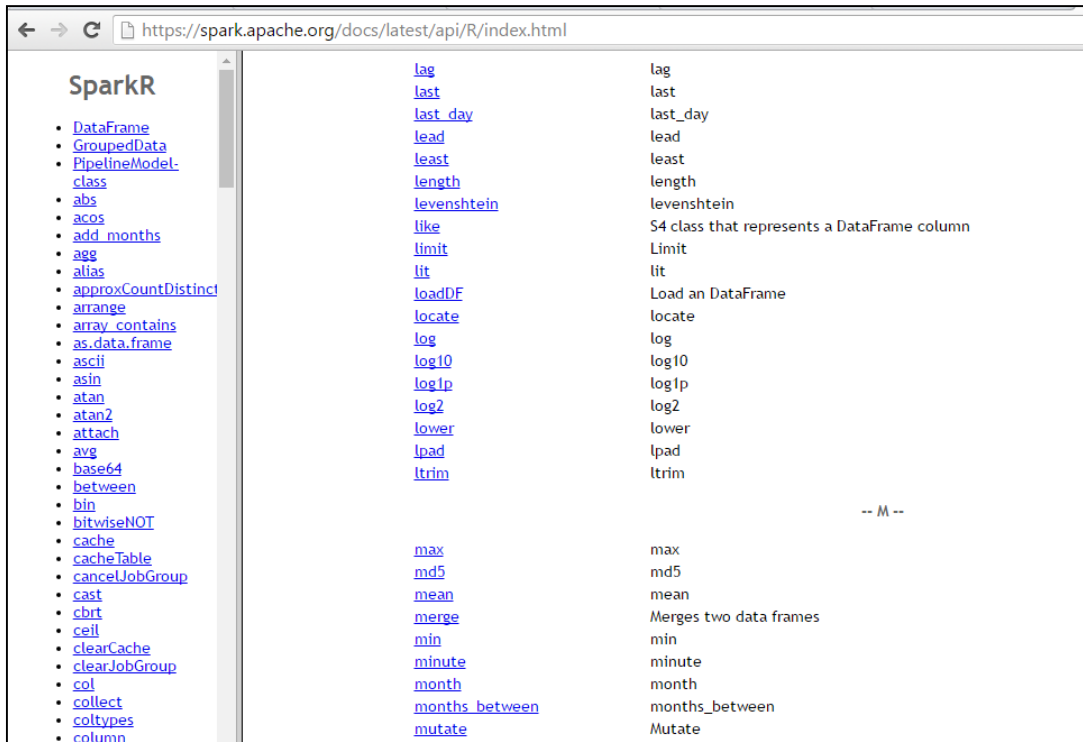
Create RDD Data Frame



The screenshot shows the RStudio IDE with a script titled 'bigdata2016.R'. The script contains the following R code:

```
25  
26 #create rdd data frame  
27 loan <- read.df(sqlContext, "/data/clean/loan.txt", source = "com.databricks.spark.csv", header="true", inferSch  
28  
29 #see head  
30 head(loan)  
31 take(loan, 10)|  
32  
33 #information on a column  
34 typeof(take(loan, 2)[["loan_amnt"]])  
35 # [1] "double"  
36  
37 #rows in the data set  
38 count(loan)  
39 nrow(loan)  
40  
41  
42 printSchema(loan)  
43  
44 #Register dataframe as Table  
45 registerTempTable(loan, "loanTemp")  
46  
47  
48 #Test TempTable  
49 print(head(sql(sqlContext, "Select * from loanTemp limit 5")))  
50  
31:15 | (Top Level) ↕
```

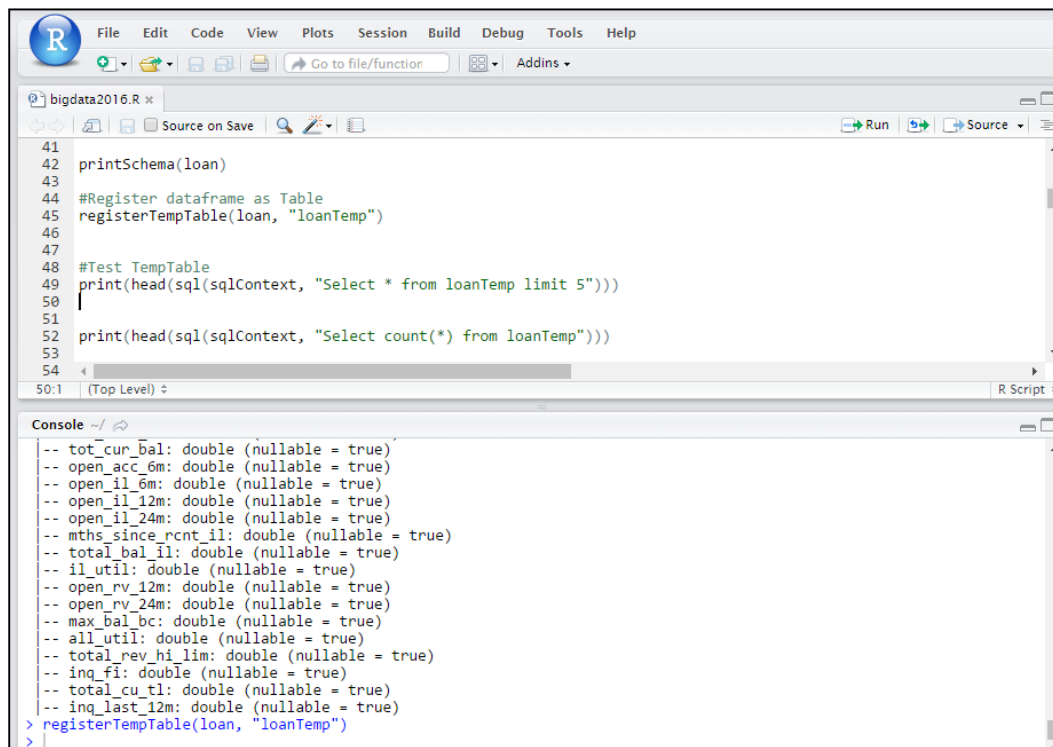
SparkR API



A screenshot of a web browser displaying the SparkR API index page. The browser's address bar shows the URL `https://spark.apache.org/docs/latest/api/R/index.html`. The page has a sidebar on the left titled "SparkR" containing a list of API categories and functions. The main content area displays a grid of API entries, each with a function name and a brief description.

SparkR		
<ul style="list-style-type: none">• 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• cell• clearCache• clearJobGroup• col• collect• coltypes• column	<ul style="list-style-type: none">• lag• last• last_day• lead• least• length• levenshtein• like• limit• lit• loadDF• locate• log• log10• log1p• log2• lower• lpad• ltrim	<ul style="list-style-type: none">• lag• last• last_day• lead• least• length• levenshtein• S4 class that represents a DataFrame column• Limit• lit• Load an DataFrame• locate• log• log10• log1p• log2• lower• lpad• ltrim

sqlContext



The screenshot shows the RStudio IDE with a script editor and a console. The script editor contains R code that interacts with a Spark SQL context. The console shows the output of the code, which is a list of column names and their data types for a table named 'loanTemp'.

```
41  
42 printSchema(loan)  
43  
44 #Register dataframe as Table  
45 registerTempTable(loan, "loanTemp")  
46  
47  
48 #Test TempTable  
49 print(head(sql(sqlContext, "Select * from loanTemp limit 5")))  
50  
51  
52 print(head(sql(sqlContext, "Select count(*) from loanTemp")))  
53  
54
```

```
-- 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

The screenshot shows the RStudio IDE with a script editor, a console, and an environment pane.

Script Editor:

```
53  
54  
55 ##get data from Hive table  
56 #create hive context  
57 hiveContext <- sparkRHive.init(sc)  
58  
59  
60 #Query hive table  
61 results = collect(sql(hiveContext, "From bd2016.loans SELECT * limit 15"))  
62  
63 print(results)  
64  
65 #collect the data locally  
66
```

Console:

```
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)
```

Environment Pane:

- Global Environment**
- Data**
 - results**: 15 obs. of 74 variables
 - hiveContext**: Environment
 - loan**: Formal class DataFrame
 - sc**: Environment
- Files**: New Folder, Upload, Delete, Rename
- Home**
- Table View**

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

Table View:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade
1	NA	NA	NA	NA	NA	term	int_rate	installment	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

Full Power of R

The screenshot displays the RStudio environment with the following components:

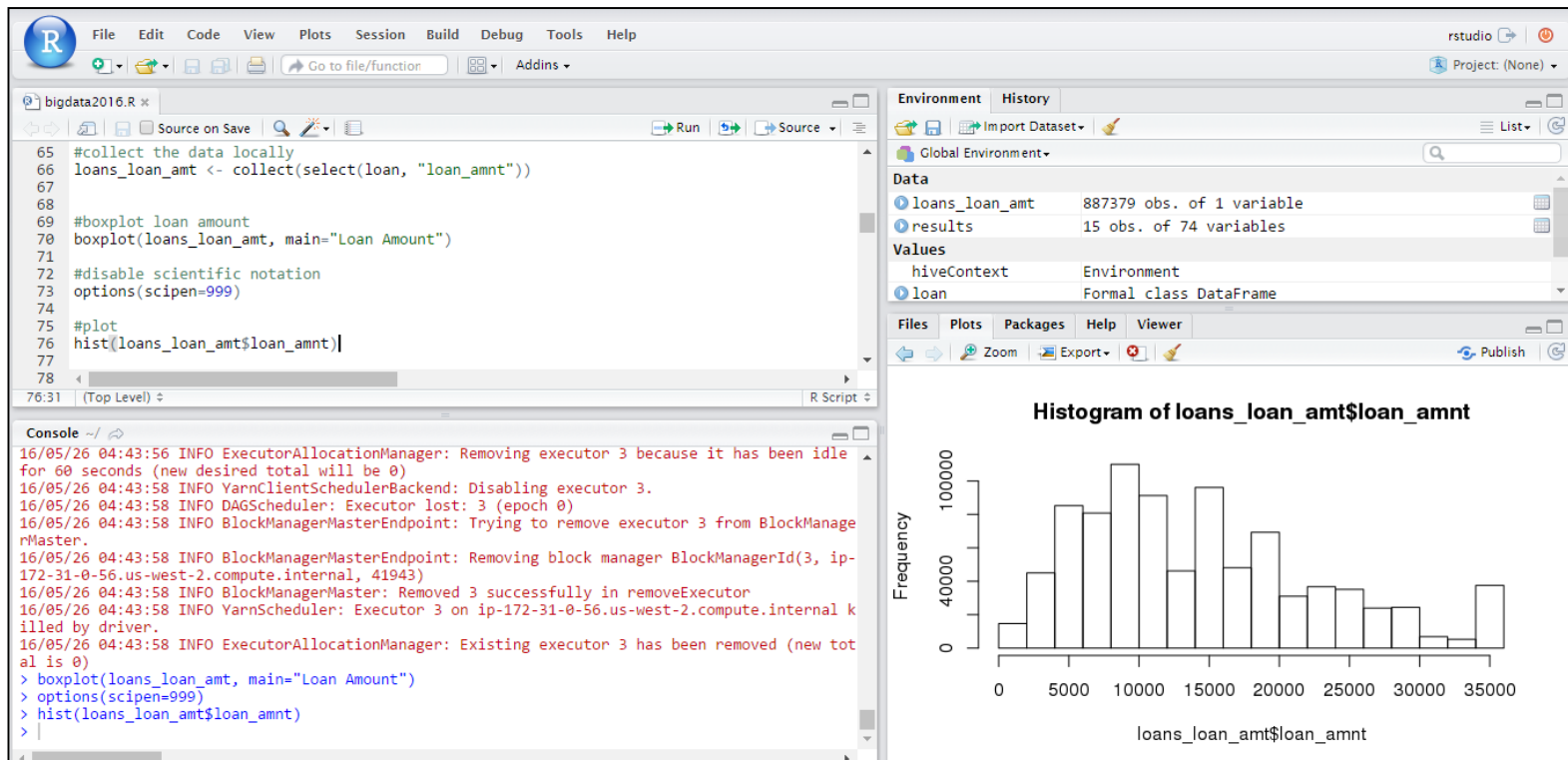
- Source Editor:** Contains R code for data collection and visualization.

```
64  
65 #collect the data locally  
66 loans_loan_amt <- collect(select(loan, "loan_amnt"))  
67  
68  
69 #boxplot loan amount  
70 boxplot(loans_loan_amt, main="Loan Amount")  
71  
72 #disable scientific notation  
73 options(scipen=999)  
74  
75 #plot  
76 hist(loans_loan_amt$loan_amnt)  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```
- Console:** Displays system logs and the execution of the boxplot command.

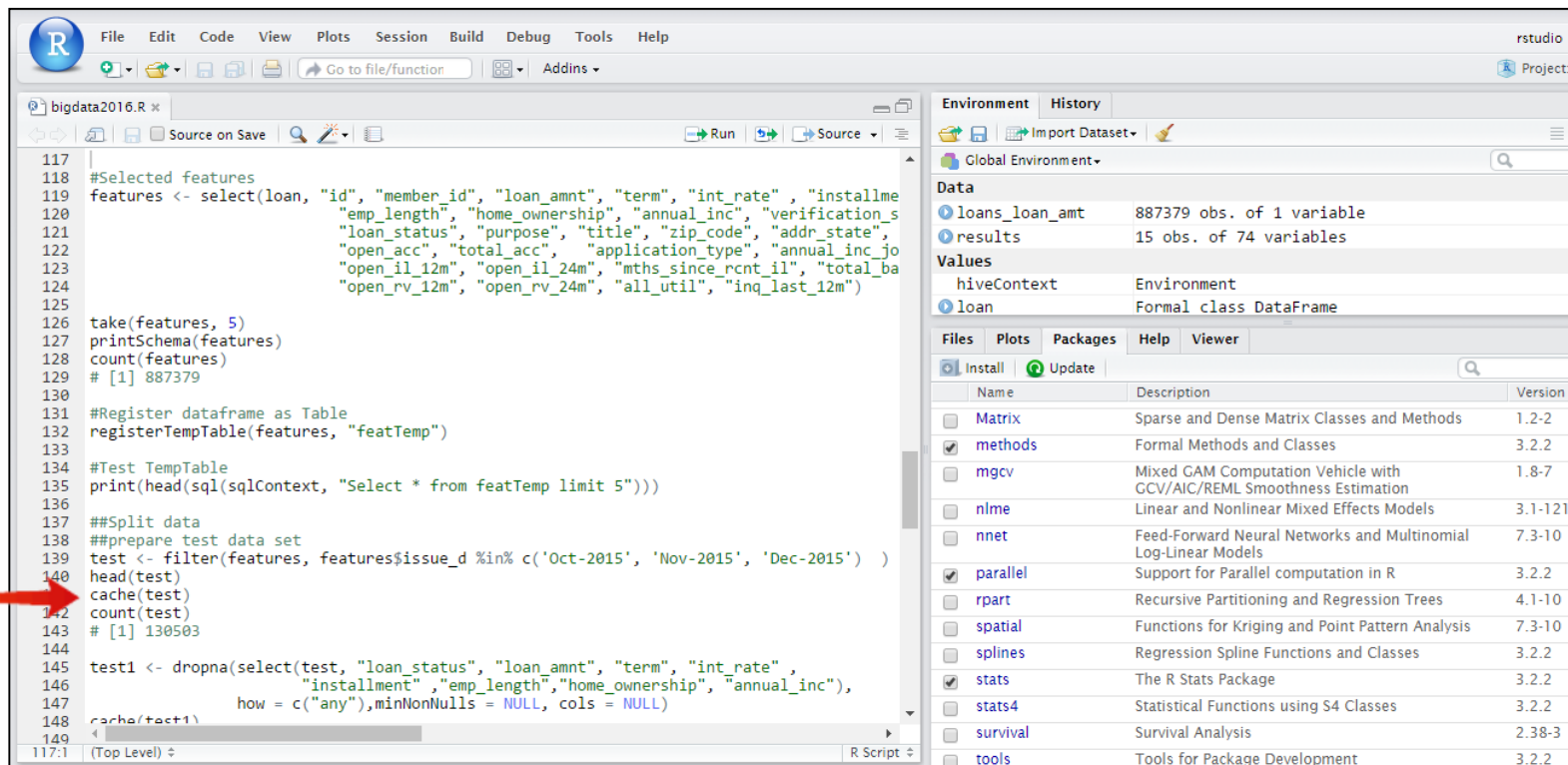
```
16/05/26 04:43:56 INFO YarnClientSchedulerBackend: Requesting to kill executor(s) 3  
16/05/26 04:43:56 INFO ExecutorAllocationManager: Removing executor 3 because it has been idle  
for 60 seconds (new desired total will be 0)  
16/05/26 04:43:58 INFO YarnClientSchedulerBackend: Disabling executor 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 BlockManag  
erMaster.  
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)  
>
```
- Environment:** Shows the loaded data objects.

Object	Class	Attributes
loans_loan_amt	Environment	887379 obs. of 1 variable
results	Formal class DataFrame	15 obs. of 74 variables
- Plots:** Displays a boxplot titled "Loan Amount". The y-axis ranges from 0 to 25,000. The boxplot shows the median, quartiles, and range of the loan amounts.

Full Power of R



SparkR: In Memory Processing



The screenshot shows the RStudio IDE with a script file named `bigdata2016.R`. The code defines a set of features, registers a temporary table, and filters data. A red arrow points to line 140, which is `cache(test)`. The Environment pane on the right shows the `loans_loan_amt` variable with 887379 observations and the `results` variable with 15 observations. The Packages pane shows a list of installed and available packages.

```
117 |
118 | #Selected features
119 | features <- select(loans, "id", "member_id", "loan_amnt", "term", "int_rate", "installme
120 |   "emp_length", "home_ownership", "annual_inc", "verification_s
121 |   "loan_status", "purpose", "title", "zip_code", "addr_state",
122 |   "open_acc", "total_acc", "application_type", "annual_inc_jo
123 |   "open_il_12m", "open_il_24m", "mths_since_rcnt_il", "total_ba
124 |   "open_rv_12m", "open_rv_24m", "all_util", "inq_last_12m")
125 |
126 | take(features, 5)
127 | printSchema(features)
128 | count(features)
129 | # [1] 887379
130 |
131 | #Register dataframe as Table
132 | registerTempTable(features, "featTemp")
133 |
134 | #Test TempTable
135 | print(head(sql(sqlContext, "Select * from featTemp limit 5")))
136 |
137 | ##Split data
138 | ##prepare test data set
139 | test <- filter(features, features$issue_d %in% c('Oct-2015', 'Nov-2015', 'Dec-2015') )
140 | head(test)
141 | cache(test)
142 | count(test)
143 | # [1] 130503
144 |
145 | test1 <- dropna(select(test, "loan_status", "loan_amnt", "term", "int_rate",
146 |   "installment", "emp_length", "home_ownership", "annual_inc"),
147 |   how = c("any"), minNonNulls = NULL, cols = NULL)
148 | cache(test1)
149 |
117:1 (Top Level) ↕
```

Environment

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

Packages

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



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