THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Democratizing AI with sparklyr

@javierluraschi
RStudio PBC - OSS

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

RStudio AI

RStudio is a Public Benefit Corporation supporting R. The RStudio AI team is focused on making AI accessible to the Data Science community.

Spark TensorFlow PyTorch mlflow

Credit: rstudio.com, blogs.rstudio.com/ai

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Artificial Intelligence

Artificial Intelligence is a rapid-growing field with significant transformational impact to many industries and civilization at large.


ImageNet Classification with Deep Convolutional Neural Networks

Modern AI starts with the deep learning breakthrough from AlexNet and more recently with achievements like DeepMind's AlphaGo.

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

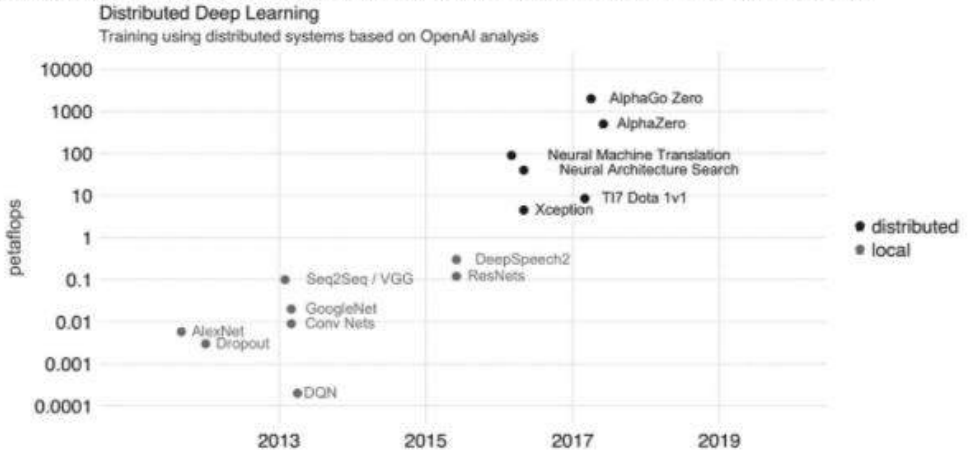
Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Artificial Intelligence

Deep Learning is a key discipline of modern AI, but Reinforcement Learning, statistical analysis and distributed computing are as relevant today.



Credit: therinspark.com

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Frameworks

Apache Spark, TensorFlow, PyTorch, MLflow, Horovod, and so on, are arguably required frameworks to build AI

HOROVOD PyTorch mlflow

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

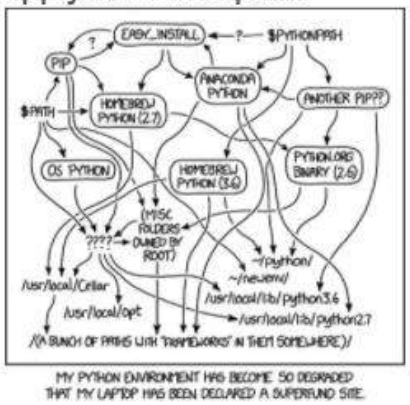
Javier Luraschi
Software Engineer
RStudio

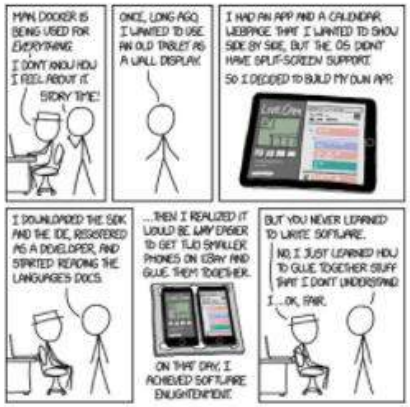
Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Today

AI is still hard, requires highly-qualified teams to set up the infrastructure and successfully apply AI techniques.





Installing and using say: TensorFlow, Spark and MLflow, is still hard.

Credit: xkcd.com

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

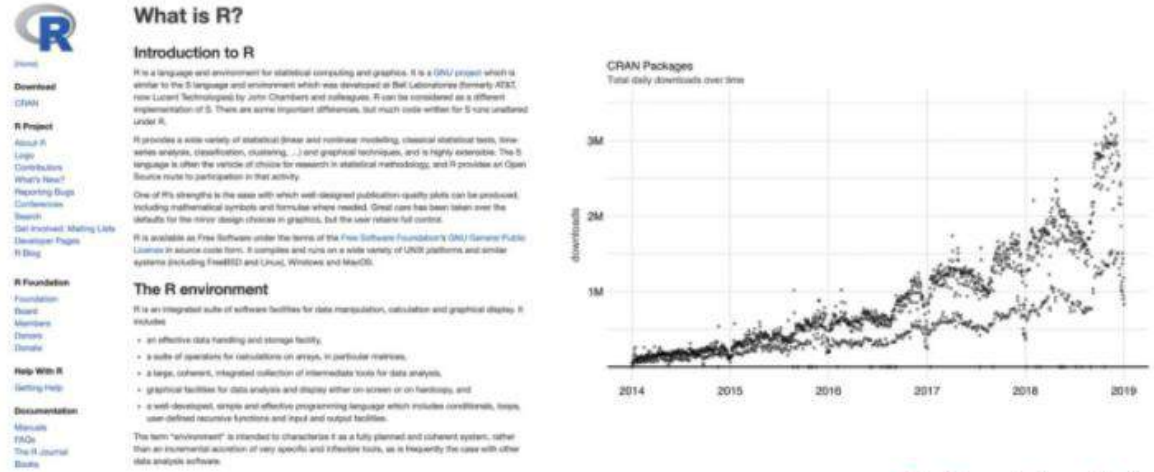
Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

R

R is a language and environment for statistical computing and graphics, with emphasis on making Data Science accessible to everyone and growing!



Credit: r-project.org, therinspark.com

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

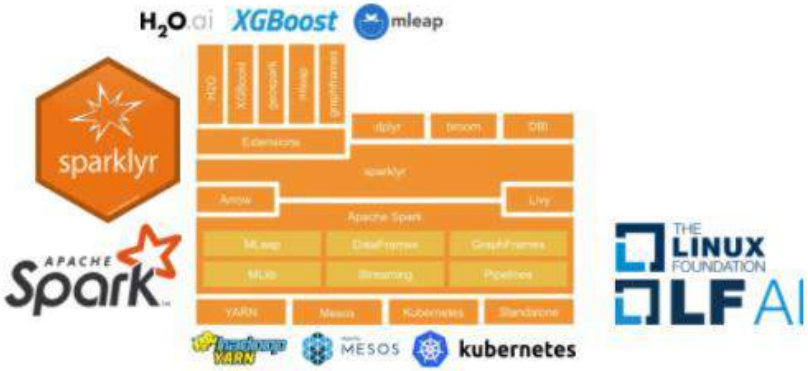
Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Sparklyr

Sparklyr is an open-source and modern interface to scale data science and machine learning workflows using Apache Spark™ and R.



Sparklyr joins the Linux Foundation under LF AI in 2020.

Credit: sparklyr.ai, lfai.foundation

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Ease of use

Sparklyr focuses on ease of use. You can install sparklyr and dependencies like Spark with one line of code, even in Windows:

```
# Install Spark
install.packages("sparklyr")
sparklyr::spark_install()
```

You can also then connect to local or remote Spark clusters with a line of code:

```
# Connect to Spark
sc <- sparklyr::spark_connect(master = "local")
```

Credit: sparklyr.ai

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Comprehensive

```
spark_install()                                  # Install local Spark
sc <- spark_connect(master = "local")            # Connect to Spark cluster

cars <- spark_read_csv(sc, "cars", "input/")     # Read data in Spark

summarize(cars, n = n())                         # Count records with dplyr
dbGetQuery(sc, "SELECT count(*) FROM cars")      # Count records with DBI

ml_linear_regression(cars, mpg ~ wt + cyl)       # Perform linear regression

ml_pipeline(sc) %>%                              # Define Spark pipeline
  ft_r_formula(mpg ~ wt + cyl) %>%               # Add formula transformation
  ml_linear_regression()                         # Add model to pipeline

spark_context(sc) %>% invoke("version")          # Extend sparklyr with Scala
spark_apply(cars, nrow)                          # Extend sparklyr with R

stream_read_csv(sc, "input/") %>%                # Define Spark stream
  filter(mpg > 30) %>%                           # Add dplyr transformation
  stream_write_json("output/")                   # Start processing stream
```

Credit: sparklyr.ai

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

Go to file/function Addins 2020-06-29

Source

Console Terminal Jobs

Terminal 1 ~/RStudio/talks/2020-06-29

```
Javiers-MacBook-Pro:2020-06-29 javierluraschi$
```

Environment History Connections Git

Import Dataset List

Global Environment

Environment is empty

Files Plots Packages Help Viewer

New Folder Delete Rename More

Home > RStudio > talks > 2020-06-29

Name	Size
..	
.gitignore	43 B
2020-06-29.Rproj	205 B
derby.log	779 B
input	
logs	
ossummit - sparklyr.pdf	6.9 MB
output	
sparklyr-ossumit.nb.html	771.1 KB
sparklyr-ossummit.Rmd	1.9 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

Go to file/function | Addins | 2020-06-29

Source

Console Terminal Jobs

Terminal 1 (busy) ~/RStudio/talks/2020-06-29

```
Javiers-MacBook-Pro:2020-06-29 javierluraschi$ R

R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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.

> library(sparklyr)
```

Environment History Connections Git

Import Dataset | List

Global Environment

Environment is empty

Files Plots Packages Help Viewer

New Folder | Delete | Rename | More

Home > RStudio > talks > 2020-06-29

Name	Size
..	
.gitignore	43 B
2020-06-29.Rproj	205 B
derby.log	779 B
input	
logs	
ossummit - sparklyr.pdf	6.9 MB
output	
sparklyr-ossumit.nb.html	771.1 KB
sparklyr-ossummit.Rmd	1.9 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd*

```
---
title: "R Notebook"
output: html_notebook
---

```{r echo=FALSE, message=FALSE}
library(sparklyr)                          # Load sparklyr
library(dplyr)                             # Load dplyr
library(DBI)                               # Load DBI
dir.create("input")                        # Create cars folder
write.csv(mtcars, "input/cars.csv")        # Write data in R
ggplot2::theme_set(ggplot2::theme_minimal()) # Make plot pretty
```

```{r}
spark_install()                            # Install local Spark
sc <- spark_connect(master = "local")      # Connect to Spark cluster
```

```{r class.source='fragment'}
cars <- spark_read_csv(sc, "cars", "input/")  # Read data in Spark
```

```{r class.source='fragment'}
summarize(cars, n = n())                   # Count records with dplyr
dbGetQuery(sc, "SELECT count(*) FROM cars")  # Count records with DBI
```

```{r}
library(ggplot2)
sample_frac(cars, 0.9) %>%                 # Sample dataset into R
  ggplot(aes(x = wt, y = mpg, shape = cyl)) +  # Use ggplot2 to plot
  geom_point() + scale_shape_identity()    # A scatter plot
```

```{r}
ml_linear_regression(cars, mpg ~ wt + cyl)  # Perform linear regression
```
```

Environment is empty

Name	Size
.gitignore	43 B
2020-06-29.Rproj	205 B
derby.log	779 B
input	
logs	
ossummit - sparklyr.pdf	6.9 MB
output	
sparklyr-ossumit.nb.html	771.1 KB
sparklyr-ossummit.Rmd	1.9 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd*

```
library(sparklyr)                          # Load sparklyr
library(dplyr)                             # Load dplyr
library(DBI)                               # Load DBI
dir.create("input")                        # Create cars folder
write.csv(mtcars, "input/cars.csv")        # Write data in R
ggplot2::theme_set(ggplot2::theme_minimal())  # Make plot pretty
```

```
'input' already exists
```

```{r}
spark_install()                            # Install local Spark
sc <- spark_connect(master = "local")      # Connect to Spark cluster
```

```{r class.source='fragment'}
cars <- spark_read_csv(sc, "cars", "input/")  # Read data in Spark
```

```{r class.source='fragment'}
summarize(cars, n = n())                   # Count records with dplyr
dbGetQuery(sc, "SELECT count(*) FROM cars")  # Count records with DBI
```

```{r}
library(ggplot2)
sample_frac(cars, 0.9) %>%                 # Sample dataset into R
  ggplot(aes(x = wt, y = mpg, shape = cyl)) +  # Use ggplot2 to plot
  geom_point() + scale_shape_identity()    # A scatter plot
```

```{r}
ml_linear_regression(cars, mpg ~ wt + cyl)  # Perform linear regression
```

```{r class.source='fragment'}
ml_pipeline(sc) %>%                        # Define Spark pipeline
  ft_r_formula(mpg ~ wt + cyl) %>%         # Add formula transformation
```

9:60 Chunk 1 R Markdown

Console

Environment History Connections Git

Global Environment

Environment is empty

Files Plots Packages Help Viewer

Home > RStudio > talks > 2020-06-29

Name	Size
..	
.gitignore	43 B
2020-06-29.Rproj	205 B
derby.log	779 B
input	
logs	
ossummit - sparklyr.pdf	6.9 MB
output	
sparklyr-ossumit.nb.html	771.1 KB
sparklyr-ossummit.Rmd	1.9 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd*

```
library(sparklyr)                          # Load sparklyr
library(dplyr)                             # Load dplyr
library(DBI)                               # Load DBI
dir.create("input")                        # Create cars folder
write.csv(mtcars, "input/cars.csv")        # Write data in R
ggplot2::theme_set(ggplot2::theme_minimal())  # Make plot pretty
```

```{r}
spark_install()                            # Install local Spark
sc <- spark_connect(master = "local")      # Connect to Spark cluster
```

Spark 2.4.3 for Hadoop 2.7 or later already installed.

```{r class.source='fragment'}
cars <- spark_read_csv(sc, "cars", "input/")   # Read data in Spark
```

```{r class.source='fragment'}
summarize(cars, n = n())                   # Count records with dplyr
dbGetQuery(sc, "SELECT count(*) FROM cars")   # Count records with DBI
```

```{r}
library(ggplot2)
sample_frac(cars, 0.9) %>%                 # Sample dataset into R
  ggplot(aes(x = wt, y = mpg, shape = cyl)) +   # Use ggplot2 to plot
  geom_point() + scale_shape_identity()    # A scatter plot
```

```{r}
ml_linear_regression(cars, mpg ~ wt + cyl)   # Perform linear regression
```

```{r class.source='fragment'}
ml_pipeline(sc) %>%                        # Define Spark pipeline
  ft_r_formula(mpg ~ wt + cyl) %>%         # Add formula transformation
```

16:1 Chunk 2 R Markdown

Console

Environment History Connections Git

Global Environment

Environment is empty

Files Plots Packages Help Viewer

Home > RStudio > talks > 2020-06-29

Name	Size
.gitignore	43 B
2020-06-29.Rproj	205 B
derby.log	779 B
input	
logs	
ossummit - sparklyr.pdf	6.9 MB
output	
sparklyr-ossumit.nb.html	771.1 KB
sparklyr-ossummit.Rmd	1.9 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd*

```
library(sparklyr)                          # Load sparklyr
library(dplyr)                             # Load dplyr
library(DBI)                               # Load DBI
dir.create("input")                        # Create cars folder
write.csv(mtcars, "input/cars.csv")        # Write data in R
ggplot2::theme_set(ggplot2::theme_minimal()) # Make plot pretty
```

```{r}
spark_install()                            # Install local Spark
sc <- spark_connect(master = "local")      # Connect to Spark cluster
```

Spark 2.4.3 for Hadoop 2.7 or later already installed.

```{r class.source='fragment'}
cars <- spark_read_csv(sc, "cars", "input/")  # Read data in Spark
```

```{r class.source='fragment'}
summarize(cars, n = n())                   # Count records with dplyr
dbGetQuery(sc, "SELECT count(*) FROM cars")  # Count records with DBI
```

```{r}
library(ggplot2)
sample_frac(cars, 0.9) %>%                 # Sample dataset into R
  ggplot(aes(x = wt, y = mpg, shape = cyl)) +  # Use ggplot2 to plot
  geom_point() + scale_shape_identity()    # A scatter plot
```

```{r}
ml_linear_regression(cars, mpg ~ wt + cyl)  # Perform linear regression
```

```{r class.source='fragment'}
ml_pipeline(sc) %>%                        # Define Spark pipeline
  ft_r_formula(mpg ~ wt + cyl) %>%         # Add formula transformation
```

Environment is empty

Name	Size
.gitignore	43 B
2020-06-29.Rproj	205 B
derby.log	779 B
input	
logs	
ossummit - sparklyr.pdf	6.9 MB
output	
sparklyr-ossumit.nb.html	771.1 KB
sparklyr-ossummit.Rmd	1.9 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd*

```r
library(sparklyr)                          # Load sparklyr
library(dplyr)                             # Load dplyr
library(DBI)                               # Load DBI
dir.create("input")                        # Create cars folder
write.csv(mtcars, "input/cars.csv")        # Write data in R
ggplot2::theme_set(ggplot2::theme_minimal()) # Make plot pretty
```

```{r}
spark_install()                            # Install local Spark
sc <- spark_connect(master = "local")      # Connect to Spark cluster
```

* Using Spark: 2.4.3

```{r class.source='fragment'}
cars <- spark_read_csv(sc, "cars", "input/")  # Read data in Spark
```

```{r class.source='fragment'}
summarize(cars, n = n())                   # Count records with dplyr
dbGetQuery(sc, "SELECT count(*) FROM cars")  # Count records with DBI
```

```{r}
library(ggplot2)
sample_frac(cars, 0.9) %>%                 # Sample dataset into R
  ggplot(aes(x = wt, y = mpg, shape = cyl)) +  # Use ggplot2 to plot
  geom_point() + scale_shape_identity()    # A scatter plot
```

```{r}
ml_linear_regression(cars, mpg ~ wt + cyl)  # Perform linear regression
```

```{r class.source='fragment'}
ml_pipeline(sc) %>%                        # Define Spark pipeline
  ft_r_formula(mpg ~ wt + cyl) %>%         # Add formula transformation
```

Environment History Connections Git

local Spark

cars

Files Plots Packages Help Viewer

~/home/RStudio/talks/2020-06-29/input

Name	Size
..	
cars.csv	1.7 KB

Console

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

Go to file/function Addins 2020-06-29

sparklyr-ossummit.Rmd* cars

(Displaying up to 1,000 records)

	_c0	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
6	Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
11	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
12	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
13	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
14	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
15	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
16	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
17	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
18	Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
19	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
20	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
21	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
22	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
23	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
24	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
25	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
26	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1

Showing 1 to 28 of 32 entries, 12 total columns

Console

Environment History Connections Git

Spark Log SQL Help

local Spark

cars

```
_c0 : chr "Mazda RX4" "Mazda RX"..
mpg : num 21 21 22.8 21.4 18.7
cyl : int 6 6 4 6 8
disp : num 160 160 108 258 360
hp : int 110 110 93 110 175
drat : num 3.9 3.9 3.85 3.08 3.15
wt : num 2.62 2.88 2.32 3.21 3.44
qsec : num 16.5 17 18.6 19.4 17
vs : int 0 0 1 1 0
am : int 1 1 1 0 0
```

Files Plots Packages Help Viewer

New Folder Delete Rename More

Home > RStudio > talks > 2020-06-29 > input

Name	Size
..	
cars.csv	1.7 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars

```r
library(sparklyr)                          # Load sparklyr
library(dplyr)                             # Load dplyr
library(DBI)                               # Load DBI
dir.create("input")                        # Create cars folder
write.csv(mtcars, "input/cars.csv")        # Write data in R
ggplot2::theme_set(ggplot2::theme_minimal())  # Make plot pretty
```

```{r}
spark_install()                            # Install local Spark
sc <- spark_connect(master = "local")      # Connect to Spark cluster
```

* Using Spark: 2.4.3

```{r class.source='fragment'}
cars <- spark_read_csv(sc, "cars", "input/")   # Read data in Spark
```

```{r class.source='fragment'}
summarize(cars, n = n())                   # Count records with dplyr
dbGetQuery(sc, "SELECT count(*) FROM cars")  # Count records with DBI
```

```{r}
library(ggplot2)
sample_frac(cars, 0.9) %>%                 # Sample dataset into R
  ggplot(aes(x = wt, y = mpg, shape = cyl)) +  # Use ggplot2 to plot
  geom_point() + scale_shape_identity()    # A scatter plot
```

```{r}
ml_linear_regression(cars, mpg ~ wt + cyl)  # Perform linear regression
```

```{r class.source='fragment'}
ml_pipeline(sc) %>%                        # Define Spark pipeline
  ft_r_formula(mpg ~ wt + cyl) %>%         # Add formula transformation
```

21:1 Chunk 3 R Markdown

Console

Environment | History | Connections | Git

Spark | Log | SQL | Help

local Spark

```
cars
  _c0 : chr "Mazda RX4" "Mazda RX"..
  mpg : num 21 21 22.8 21.4 18.7
  cyl : int 6 6 4 6 8
  disp : num 160 160 108 258 360
  hp : int 110 110 93 110 175
  drat : num 3.9 3.9 3.85 3.08 3.15
  wt : num 2.62 2.88 2.32 3.21 3.44
  qsec : num 16.5 17 18.6 19.4 17
  vs : int 0 0 1 1 0
  am : int 1 1 1 0 0
```

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More

Home > RStudio > talks > 2020-06-29 > input

Name	Size
..	
cars.csv	1.7 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars

```r
library(sparklyr)                          # Load sparklyr
library(dplyr)                             # Load dplyr
library(DBI)                               # Load DBI
dir.create("input")                        # Create cars folder
write.csv(mtcars, "input/cars.csv")        # Write data in R
ggplot2::theme_set(ggplot2::theme_minimal())  # Make plot pretty
```

```{r}
spark_install()                            # Install local Spark
sc <- spark_connect(master = "local")      # Connect to Spark cluster
```

```
* Using Spark: 2.4.3
```

```{r class.source='fragment'}
cars <- spark_read_csv(sc, "cars", "input/")   # Read data in Spark
```

```{r class.source='fragment'}
summarize(cars, n = n())                   # Count records with dplyr
dbGetQuery(sc, "SELECT count(*) FROM cars")   # Count records with DBI
```

dbGetQuery(conn, statement, ...)

n
<dbl>
32

1 row

```{r}
library(ggplot2)
sample_frac(cars, 0.9) %>%                 # Sample dataset into R
  ggplot(aes(x = wt, y = mpg, shape = cyl)) +   # Use ggplot2 to plot
  geom_point() + scale_shape_identity()    # A scatter plot
```

26:1 Chunk 4 R Markdown

Console

Environment | History | Connections | Git

Spark | Log | SQL | Help

local Spark

```
cars
  _c0 : chr "Mazda RX4" "Mazda RX"..
  mpg : num 21 21 22.8 21.4 18.7
  cyl : int 6 6 4 6 8
  disp : num 160 160 108 258 360
  hp : int 110 110 93 110 175
  drat : num 3.9 3.9 3.85 3.08 3.15
  wt : num 2.62 2.88 2.32 3.21 3.44
  qsec : num 16.5 17 18.6 19.4 17
  vs : int 0 0 1 1 0
  am : int 1 1 1 0 0
```

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More

Home > RStudio > talks > 2020-06-29 > input

Name	Size
..	
cars.csv	1.7 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars

```
* Using Spark: 2.4.3

```{r class.source='fragment'}
cars <- spark_read_csv(sc, "cars", "input/")     # Read data in Spark
```

```{r class.source='fragment'}
summarize(cars, n = n())                         # Count records with dplyr
dbGetQuery(sc, "SELECT count(*) FROM cars")      # Count records with DBI
```

count(1) <dbl>
32

1 row

```{r}
library(ggplot2)
sample_frac(cars, 0.9) %>%                       # Sample dataset into R
  ggplot(aes(x = wt, y = mpg, shape = cyl)) +    # Use ggplot2 to plot
  geom_point() + scale_shape_identity()          # A scatter plot
```

```{r}
ml_linear_regression(cars, mpg ~ wt + cyl)       # Perform linear regression
```

```{r class.source='fragment'}
ml_pipeline(sc) %>%                              # Define Spark pipeline
  ft_r_formula(mpg ~ wt + cyl) %>%               # Add formula transformation
  ml_linear_regression()                         # Add model to pipeline
```

```{r class.source='fragment'}
```
```

30:9 Chunk 5 R Markdown

Console

Environment | History | Connections | Git

local Spark

```
cars
  _c0 : chr "Mazda RX4" "Mazda RX"..
  mpg : num 21 21 22.8 21.4 18.7
  cyl : int 6 6 4 6 8
  disp : num 160 160 108 258 360
  hp : int 110 110 93 110 175
  drat : num 3.9 3.9 3.85 3.08 3.15
  wt : num 2.62 2.88 2.32 3.21 3.44
  qsec : num 16.5 17 18.6 19.4 17
  vs : int 0 0 1 1 0
  am : int 1 1 1 0 0
```

Files | Plots | Packages | Help | Viewer

New Folder Delete Rename More

home > RStudio > talks > 2020-06-29 > input

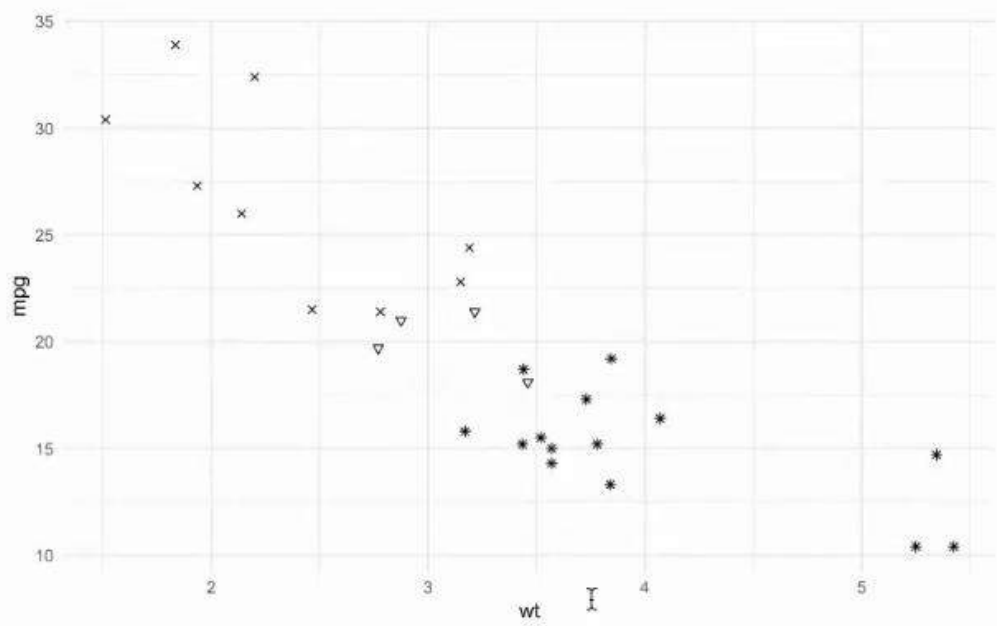
Name	Size
..	
cars.csv	1.7 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars

```r
library(ggplot2)
sample_frac(cars, 0.9) %>%                    # Sample dataset into R
  ggplot(aes(x = wt, y = mpg, shape = cyl)) +  # Use ggplot2 to plot
  geom_point() + scale_shape_identity()        # A scatter plot
```




```{r}
ml_linear_regression(cars, mpg ~ wt + cyl)      # Perform linear regression
```

```{r class.source='fragment'}
```

32:33 Chunk 5 R Markdown

Console

Environment | History | Connections | Git

Spark | Log | SQL | Help

local Spark

cars

```
_c0 : chr "Mazda RX4" "Mazda RX"..
mpg : num 21 21 22.8 21.4 18.7
cyl : int 6 6 4 6 8
disp : num 160 160 108 258 360
hp : int 110 110 93 110 175
drat : num 3.9 3.9 3.85 3.08 3.15
wt : num 2.62 2.88 2.32 3.21 3.44
qsec : num 16.5 17 18.6 19.4 17
vs : int 0 0 1 1 0
am : int 1 1 1 0 0
```

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More

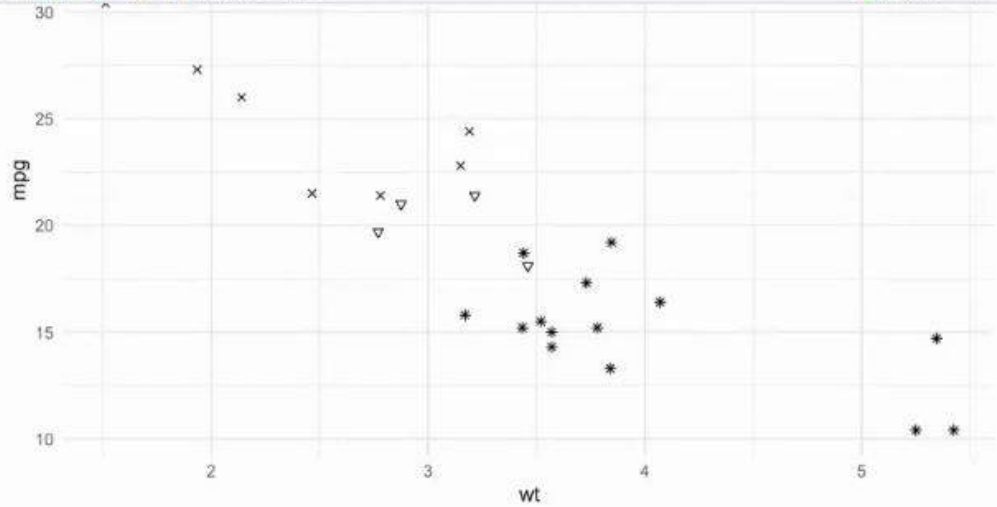
Home > RStudio > talks > 2020-06-29 > input

Name	Size
..	
cars.csv	1.7 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars




```
35
36 ```{r}
37 ml_linear_regression(cars, mpg ~ wt + cyl)        # Perform linear regression
38 ```
39
40 ```{r class.source='fragment'}
41 ml_pipeline(sc) %>%                               # Define Spark pipeline
42   ft_r_formula(mpg ~ wt + cyl) %>%                # Add formula transformation
43   ml_linear_regression()                          # Add model to pipeline
44 ```
45
46 ```{r class.source='fragment'}
47 spark_context(sc) %>% invoke("version")           # Extend sparklyr with Scala
48 spark_apply(cars, nrow)                           # Extend sparklyr with R
49 ```
50
51 ```{r class.source='fragment'}
```

37:1 Chunk 6 R Markdown

Console

Environment History Connections Git

local Spark

```
cars
  _c0 : chr "Mazda RX4" "Mazda RX"..
  mpg : num 21 21 22.8 21.4 18.7
  cyl : int 6 6 4 6 8
  disp : num 160 160 108 258 360
  hp : int 110 110 93 110 175
  drat : num 3.9 3.9 3.85 3.08 3.15
  wt : num 2.62 2.88 2.32 3.21 3.44
  qsec : num 16.5 17 18.6 19.4 17
  vs : int 0 0 1 1 0
  am : int 1 1 1 0 0
```

Files Plots Packages Help Viewer

New Folder Delete Rename More

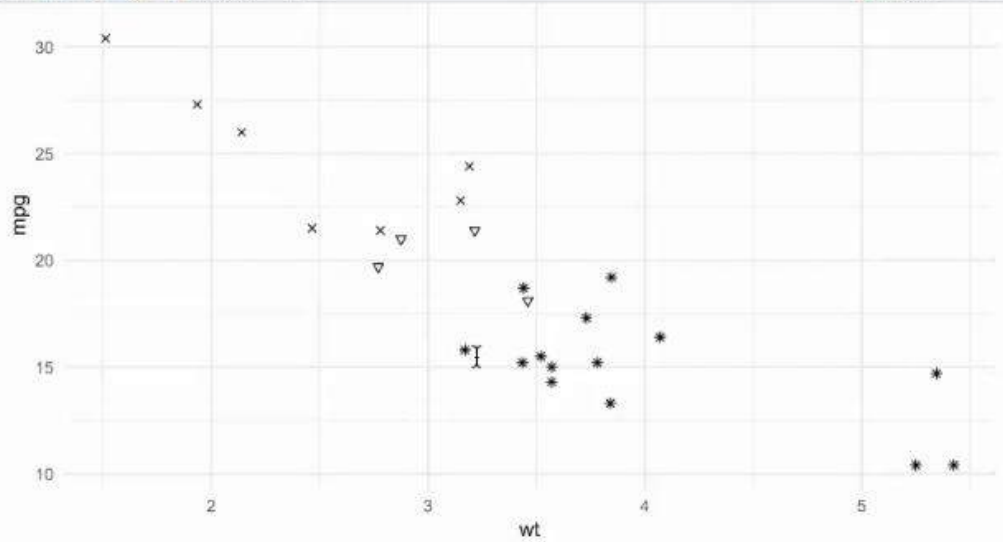
Home > RStudio > talks > 2020-06-29 > input

Name	Size
..	
cars.csv	1.7 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars




```{r}
ml_linear_regression(cars, mpg ~ wt + cyl)       # Perform linear regression
```

```
Formula: mpg ~ wt + cyl

Coefficients:
(Intercept)          wt         cyl
  39.686261   -3.190972   -1.507795
```

```{r class.source='fragment'}
ml_pipeline(sc) %>%                          # Define Spark pipeline
  ft_r_formula(mpg ~ wt + cyl) %>%           # Add formula transformation
```

Environment | History | Connections | Git

local | Spark

```
cars
  _c0 : chr "Mazda RX4" "Mazda RX"..
  mpg : num 21 21 22.8 21.4 18.7
  cyl : int 6 6 4 6 8
  disp : num 160 160 108 258 360
  hp : int 110 110 93 110 175
  drat : num 3.9 3.9 3.85 3.08 3.15
  wt : num 2.62 2.88 2.32 3.21 3.44
  qsec : num 16.5 17 18.6 19.4 17
  vs : int 0 0 1 1 0
  am : int 1 1 1 0 0
```

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More

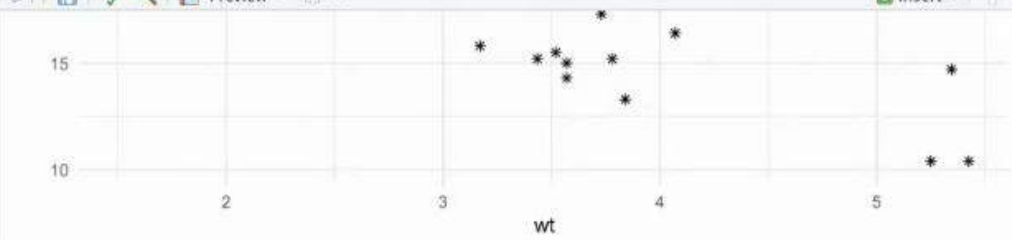
home > RStudio > talks > 2020-06-29 > input

Name	Size
..	
cars.csv	1.7 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars




```{r}
ml_linear_regression(cars, mpg ~ wt + cyl)       # Perform linear regression
```

```
Formula: mpg ~ wt + cyl

Coefficients:
(Intercept)          wt         cyl
  39.686261   -3.190972   -1.507795
```

```{r class.source='fragment'}
ml_pipeline(sc) %>%                              # Define Spark pipeline
  ft_r_formula(mpg ~ wt + cyl) %>%               # Add formula transformation
  ml_linear_regression()                         # Add model to pipeline
```

```{r class.source='fragment'}
spark_context(sc) %>% invoke("version")          # Extend sparklyr with Scala
spark_apply(cars, nrow)                          # Extend sparklyr with R
```

```{r class.source='fragment'}
stream_read_csv(sc, "input/") %>%                # Define Spark stream
  filter(mpg > 30) %>%                           # Add dplyr transformation
  stream_write_json("output/")                   # Start processing stream
```

43:24 Chunk 7 R Markdown

Console

Environment | History | Connections | Git

local Spark

```
cars
  _c0 : chr "Mazda RX4" "Mazda RX"..
  mpg : num 21 21 22.8 21.4 18.7
  cyl : int 6 6 4 6 8
  disp : num 160 160 108 258 360
  hp : int 110 110 93 110 175
  drat : num 3.9 3.9 3.85 3.08 3.15
  wt : num 2.62 2.88 2.32 3.21 3.44
  qsec : num 16.5 17 18.6 19.4 17
  vs : int 0 0 1 1 0
  am : int 1 1 1 0 0
```

Files | Plots | Packages | Help | Viewer

New Folder Delete Rename More

Home > RStudio > talks > 2020-06-29 > input

Name	Size
..	
cars.csv	1.7 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars

```
Stages
|--1 RFormula (Estimator)
|    <r_formula_a0c8793e1402>
|     (Parameters -- Column Names)
|      features_col: features
|      label_col: label
|     (Parameters)
|      force_index_label: FALSE
|      formula: mpg ~ wt + cyl
|      handle_invalid: error
|      stringIndexerOrderType: frequencyDesc
|--2 LinearRegression (Estimator)
|    <linear_regression_a0c85a93cd7d>
|     (Parameters -- Column Names)
|      features_col: features
|      label_col: label
|      prediction_col: prediction
|     (Parameters)
|      aggregation_depth: 2
|      elastic_net_param: 0
|      epsilon: 1.35
|      fit_intercept: TRUE
|      loss: squaredError
|      max_iter: 100
|      reg_param: 0
|      solver: auto
|      standardization: TRUE
|      tol: 1e-06
```

```{r class.source='fragment'}
spark_context(sc) %>% invoke("version")          # Extend sparklyr with Scala
spark_apply(cars, nrow)                          # Extend sparklyr with R
```

```{r class.source='fragment'}
stream_read_csv(sc, "input/") %>%                # Define Spark stream
  filter(mpg > 30) %>%                           # Add dplyr transformation
  stream_write_json("output/")                   # Start processing stream
```

43:24 Chunk 7 R Markdown

Console

Environment History Connections Git

local Spark

```
cars
  _c0 : chr "Mazda RX4" "Mazda RX"..
  mpg : num 21 21 22.8 21.4 18.7
  cyl : int 6 6 4 6 8
  disp : num 160 160 108 258 360
  hp : int 110 110 93 110 175
  drat : num 3.9 3.9 3.85 3.08 3.15
  wt : num 2.62 2.88 2.32 3.21 3.44
  qsec : num 16.5 17 18.6 19.4 17
  vs : int 0 0 1 1 0
  am : int 1 1 1 0 0
```

Files Plots Packages Help Viewer

New Folder Delete Rename More

Home > RStudio > talks > 2020-06-29 > input

Name	Size
..	
cars.csv	1.7 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars

```
45
46 ```{r class.source='fragment'}
47 spark_context(sc) %>% invoke("version")          # Extend sparklyr with Scala
48 spark_apply(cars, hrow)                           # Extend sparklyr with R
49 ```
```

R Console | tbl_sql 0 x 1

result <int>
32

1 row

```
50
51 ```{r class.source='fragment'}
52 stream_read_csv(sc, "input/") %>%                # Define Spark stream
53   filter(mpg > 30) %>%                            # Add dplyr transformation
54   stream_write_json("output/")                    # Start processing stream
55 ```
```

48:19 Chunk 8 R Markdown

Console

Environment | History | Connections | Git

local Spark

```
cars
  _c0 : chr "Mazda RX4" "Mazda RX"..
  mpg : num 21 21 22.8 21.4 18.7
  cyl : int 6 6 4 6 8
  disp : num 160 160 108 258 360
  hp : int 110 110 93 110 175
  drat : num 3.9 3.9 3.85 3.08 3.15
  wt : num 2.62 2.88 2.32 3.21 3.44
  qsec : num 16.5 17 18.6 19.4 17
  vs : int 0 0 1 1 0
  am : int 1 1 1 0 0
```

Files | Plots | Packages | Help | Viewer

New Folder Delete Rename More

Home > RStudio > talks > 2020-06-29 > input

Name	Size
..	
cars.csv	1.7 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars

```
  |     standardization: TRUE
  |     tol: 1e-06
```

```{r class.source='fragment'}
spark_context(sc) %>% invoke("version")          # Extend sparklyr with Scala
spark_apply(cars, nrow)                          # Extend sparklyr with R
```

R Console

tbl_sql
0 x 1

result <int>
32

1 row

stream_read_csv(sc, path, name = NULL, header = TRUE, columns = NULL, delimiter = ",", quote = "\"", escape = "\\", charset = "UTF-8", null_value stream_read_csv = list(), ...)

```{r class.source='fragment'}
stream_read_csv(sc, "input/") %>%                # Define Spark stream
  filter(mpg > 30) %>%                           # Add dplyr transformation
  stream_write_json("output/")                   # Start processing stream
```

54:75 Chunk 9 R Markdown

Console

Environment History Connections Git

local Spark

cars
```
_c0 : chr "Mazda RX4" "Mazda RX"..
mpg : num 21 21 22.8 21.4 18.7
cyl : int 6 6 4 6 8
disp : num 160 160 108 258 360
hp : int 110 110 93 110 175
drat : num 3.9 3.9 3.85 3.08 3.15
wt : num 2.62 2.88 2.32 3.21 3.44
qsec : num 16.5 17 18.6 19.4 17
vs : int 0 0 1 1 0
am : int 1 1 1 0 0
```

Files Plots Packages Help Viewer

New Folder Delete Rename More

home > RStudio > talks > 2020-06-29 > input

Name	Size
..	
cars.csv	1.7 KB

SCREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars

```
|       aggregation_depth: 2
|       elastic_net_param: 0
|       epsilon: 1.35
|       fit_intercept: TRUE
|       loss: squaredError
|       max_iter: 100
|       reg_param: 0
|       solver: auto
|       standardization: TRUE
|       tol: 1e-06
```

```{r class.source='fragment'}
spark_context(sc) %>% invoke("version")          # Extend sparklyr with Scala
spark_apply(cars, nrow)                          # Extend sparklyr with R
```

R Console

tbl_sql
0 x 1

result <int>
32

1 row

```{r class.source='fragment'}
stream_read_csv(sc, "input/") %>%                # Define Spark stream
  filter(mpg > 30) %>%                           # Add dplyr transformation
  stream_write_json("output/")                   # Start processing stream
```

```
Stream: 0f7d97ab-5f8c-41b7-918a-757ed41253a6
Status: Waiting for next trigger
Active: TRUE
```

52:15 Chunk 9 R Markdown

Console

Environment History Connections Git

Spark Log SQL Help

local Spark

```
cars
  _c0 : chr "Mazda RX4" "Mazda RX"..
  mpg : num 21 21 22.8 21.4 18.7
  cyl : int 6 6 4 6 8
  disp : num 160 160 108 258 360
  hp : int 110 110 93 110 175
  drat : num 3.9 3.9 3.85 3.08 3.15
  wt : num 2.62 2.88 2.32 3.21 3.44
  qsec : num 16.5 17 18.6 19.4 17
  vs : int 0 0 1 1 0
  am : int 1 1 1 0 0
```

Files Plots Packages Help Viewer

New Folder Delete Rename More

ome > RStudio > talks > 2020-06-29 > output

Name	Size
..	
_spark_metadata	
checkpoints	
part-00000-32b4e5ec-48ae-4...	514 B

Lobby Sessions Sponsor Showcase Networking Experiences Fun & Games Info Profile

CREEN SHARE

~/RStudio/talks/2020-06-29 - master - RStudio

sparklyr-ossummit.Rmd* | cars

```
|       aggregation_depth: 2
|       elastic_net_param: 0
|       epsilon: 1.35
|       fit_intercept: TRUE
|       loss: squaredError
|       max_iter: 100
|       reg_param: 0
|       solver: auto
|       standardization: TRUE
|       tol: 1e-06
```

```
45
46 ```{r class.source='fragment'}
47 spark_context(sc) %>% invoke("version")          # Extend sparklyr with Scala
48 spark_apply(cars, nrow)                          # Extend sparklyr with R
49 ```
```

R Console

tbl_sql
0 x 1

result
<int>
32

1 row

stream_read_csv(sc, path, name = NULL, header = TRUE, columns = NULL, delimiter = ",", quote = "\"", escape = "\\", charset = "UTF-8", null_value
stream_read_csv = list(), ...)

```
50
51 ```{r class.source='fragment'}
52 stream_read_csv(sc, "input/") %>%                # Define Spark stream
53   filter(mpg > 30) %>%                           # Add dplyr transformation
54   stream_write_json("output/")                   # Start processing stream
55 ```
```

```
Stream: 0f7d97ab-5f8c-41b7-918a-757ed41253a6
Status: Waiting for next trigger
Active: TRUE
```

52:1 Chunk 9 R Markdown

Console

Environment History Connections Git

Spark Log SQL Help

local Spark

```
cars
  _c0 : chr "Mazda RX4" "Mazda RX"..
  mpg : num 21 21 22.8 21.4 18.7
  cyl : int 6 6 4 6 8
  disp : num 160 160 108 258 360
  hp : int 110 110 93 110 175
  drat : num 3.9 3.9 3.85 3.08 3.15
  wt : num 2.62 2.88 2.32 3.21 3.44
  qsec : num 16.5 17 18.6 19.4 17
  vs : int 0 0 1 1 0
  am : int 1 1 1 0 0
```

Files Plots Packages Help Viewer

New Folder Delete Rename More

ome > RStudio > talks > 2020-06-29 > output

Name	Size
..	
_spark_metadata	
checkpoints	
part-00000-32b4e5ec-48ae-4...	514 B

attendees on the OSS+ELC Slack Channel!

powered by Intrado

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Rich in Functionality

But you can also use sparklyr to train complex models using frameworks like TensorFlow and Spark.

```
library(sparklyr)
sc <- spark_connect(master = "local|yarn|etc")

# partition dataset
sdf_len(sc, 3, repartition = 3) %>%
  spark_apply(function(df, barrier) {
      library(tensorflow)
      library(keras)

      # define configuration from barrier
      Sys.setenv(TF_CONFIG = "")

      # define strategy and model
      strategy <- MultiWorkerMirroredStrategy()
      with (strategy$scope(), {
        model <- keras_model_sequential() # %>% ...
        model %>% compile()
      })

      # fit and retrieve model
      model %>% fit()
  }, barrier = TRUE) %>% collect()
```

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Sponsors and Users

Sparklyr is support by all major cloud providers, is sponsored by Databricks, Qubole and RStudio, serves many users and is hosted within LF AI.

amazon web services · Azure · CLOUDERA · databricks · Google Cloud · Qubole · RStudio

databricks · Qubole · RStudio

:ZA:51° · EQUIFAX · EXPRESS SCRIPTS · IDEXX · Ketchbrook Analytics · NETFLIX

THE LINUX FOUNDATION · LF AI

Credit: sparklyr.ai

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Ecosystem

Sparklyr makes use of Apache Spark, MLlib, MLeap, Apache Livy and Apache Arrow from R, then pipelines can be used to export to Python or Java.

Local development is encouraged before running analysis in Spark clusters.

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

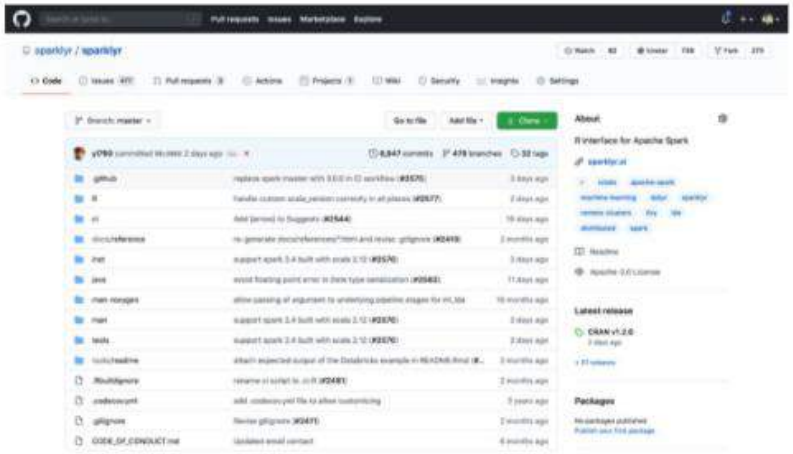
Javier Luraschi
Software Engineer
RStudio

Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

SLIDES

Contributing

We have dozens of major features requested and hundreds of issues, connect with us at github.com/sparklyr!



THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Javier Luraschi
Software Engineer
RStudio

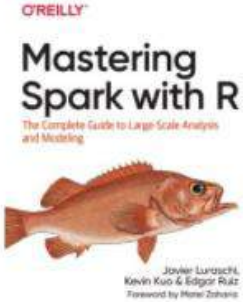
Javier is the author of "Mastering Spark with R", sparklyr, mlflow, pins and many other R packages for deep learning and data science. He holds a double degree in Math and Software Engineer and decades of industry experience with a focus on data analysis. He currently works in RStudio and previously in Microsoft Research and SAP.

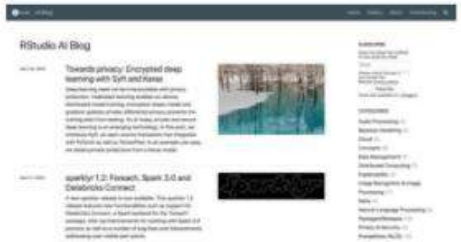
SLIDES

Resources

Thank you! Please learn more at sparklyr.ai, Mastering Spark with R, or thr RStudio AI blog.

sparklyr.ai





blogs.rstudio.com/ai