

R in Spark, TensorFlow and other libraries that flow.

Javier Luraschi



javierluraschi

javier@rstudio.com

~~R in Spark, TensorFlow and
other libraries that flow.~~

Things you might need to build autonomous cars using R

Javier Luraschi



javierluraschi

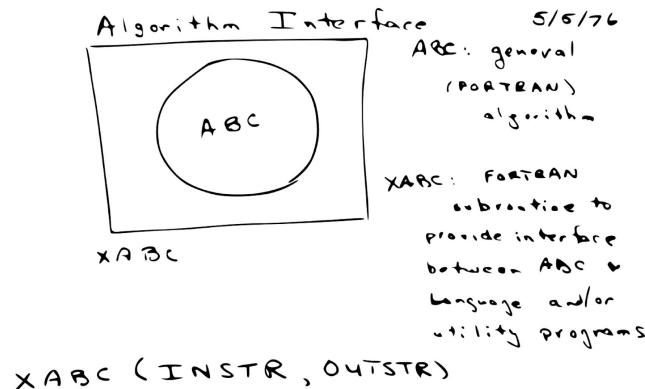
javier@rstudio.com

Intro to R



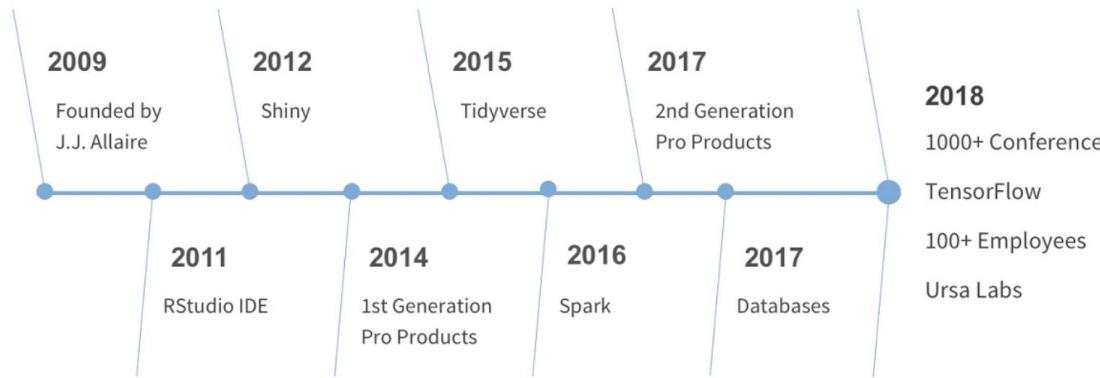
Josh Wills
@josh_wills

Data Scientist (n.): Person who is better at statistics than any software engineer and better at software engineering than any statistician.



"R is a programming language and free software environment for statistical computing and graphics."

About RStudio



RStudio's Multiverse Team

Authors of R packages to support Apache Spark, TensorFlow and MLflow.
Contributors to tidyverse and Apache Arrow.



Daniel Falbel
[@dfalbel](https://twitter.com/dfalbel)



Sigrid Keydana
[@zkajdan](https://twitter.com/zkajdan)



Kevin Kuo
[@kevinykuo](https://twitter.com/kevinykuo)



Javier Luraschi
[@javierluraschi](https://twitter.com/javierluraschi)

Multiverse - Timeline



2015
(for reference)



2016



TensorFlow

2017

mlflow

2018



2019



Tidyverse - Modern R

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.



```
library(tidyverse)

flights %>%

  group_by(month, day) %>%

  summarise(count = n(), avg_delay = mean(dep_delay)) %>%

  filter(count > 1000)
```

Tidyverse - Modern R

Tidyverse

Packages Articles Learn Help Contribute



R packages for data science

The tidyverse is an opinionated [collection of R packages](#) designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

Tidyverse - Modern R

amazon Try Prime Books ▾ Books ▾ Stock up for college

Deliver to Javier Carnation 98014 Browsing History ▾ Javier's Amazon.com Today's Deals Buy Again Gift Cards Help Whole Foods EN Hello, Javier Account & Lists ▾ Orders Try Prime ▾ 0 Cart

Books Advanced Search New Releases Best Sellers & More Children's Books Textbooks Textbook Rentals Sell Us Your Books Best Books of the Month

Amazon Best Sellers

Our most popular products based on sales. Updated hourly.

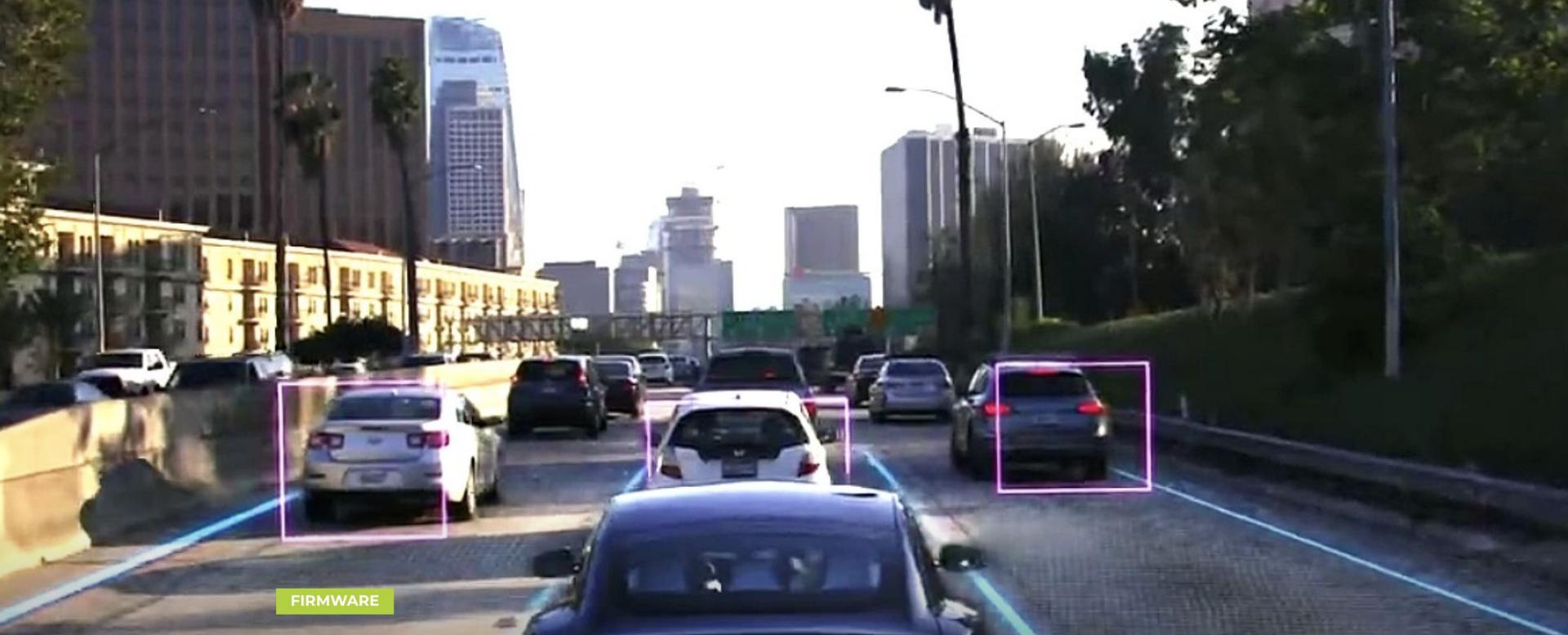
Any Department
Books
Computers & Technology
Databases & Big Data
Access
Data Mining
Data Modeling & Design
Data Processing
Data Warehousing
MySQL
Oracle
Other Databases
Relational Databases
SQL

Best Sellers in Data Processing

#1	#2	#3
R for Data Science	Hands-On Machine Learning with Scikit-Learn & TensorFlow	Practical Statistics for Data Scientists
Hadley Wickham & Garrett Grolemund	Aurélien Géron	Peter Bruce & Andrew Bruce
R for Data Science: Import, Tidy, Transform,... Hadley Wickham ★★★★★ 129 Paperback \$18.17 ✓prime	Hands-On Machine Learning with Scikit-Learn... Aurélien Géron ★★★★★ 298 Paperback \$28.49 ✓prime	Practical Statistics for Data Scientists: 50... Peter Bruce ★★★★★ 68 Paperback \$13.46 ✓prime



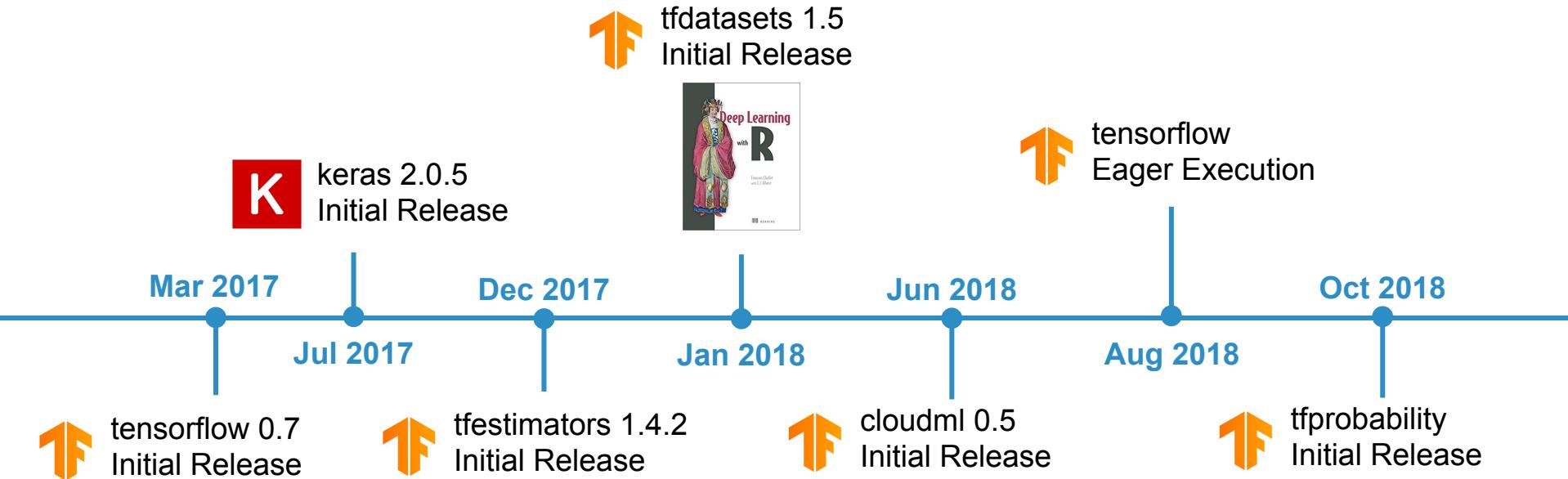
TensorFlow



FIRMWARE

Tesla showcases Autopilot, Full Self-Driving tech in Autonomy Day (Live Blog)

TensorFlow with R - Timeline



tensorflow.rstudio.com

TensorFlow for R from  Studio

Home Keras Estimators Core Tools Learn Blog  

R Interface to TensorFlow



TensorFlow™ is an open-source software library for Machine Intelligence. The R interface to TensorFlow lets you work productively using the high-level Keras and Estimator APIs, and when you need more control provides full access to the core TensorFlow API:



Keras API

The Keras API for TensorFlow provides a high-level interface for neural networks, with a focus on enabling fast experimentation.



Estimator API

The Estimator API for TensorFlow provides high-level implementations of common model types such as regressors and classifiers.



Core API

The Core TensorFlow API is a lower-level interface that provides full access to the TensorFlow computational graph.



Keras - Starting

```
library(keras)
install_keras()

mnist <- dataset_mnist()
x_train <- array_reshape(mnist$train$x, c(nrow(mnist$train$x), 784)) / 255
x_test <- array_reshape(mnist$test$x, c(nrow(mnist$test$x), 784)) / 255
y_train <- to_categorical(mnist$train$y, 10)
y_test <- to_categorical(mnist$test$y, 10)

model <- keras_model_sequential()
model %>%
  layer_dense(units = 256, activation = 'relu', input_shape = c(784)) %>%
  layer_dropout(rate = 0.4) %>%
  layer_dense(units = 128, activation = 'relu') %>%
  layer_dropout(rate = 0.3) %>%
  layer_dense(units = 10, activation = 'softmax')

model %>% compile(loss = 'categorical_crossentropy', optimizer = optimizer_rmsprop(),
metrics = c('accuracy'))
model %>% fit(x_train, y_train, epochs = 3, batch_size = 128, validation_split = 0.2)
```

TensorFlow - New? - tfdatasets

Feature specs

```
ft_spec <- training %>%
  select(-id) %>%
  feature_spec(target ~ .) %>%
  step_numeric_column(ends_with("bin")) %>%
  step_numeric_column(-ends_with("bin"),
    -ends_with("cat"),
    normalizer_fn = scaler_standard()) %>%
  step_categorical_column_with_vocabulary_list(ends_with("cat")) %>%
  step_embedding_column(ends_with("cat"),
    dimension = function(vocab_size) as.integer(sqrt(vocab_size) + 1)) %>%
  fit()
```

TensorFlow - New? - tfprobability

Combine probabilistic models and deep learning
on modern hardware

```
# create a binomial distribution with n = 7 and p = 0.3
d <- tfd_binomial(total_count = 7, probs = 0.3)
# compute mean
d %>% tfd_mean()
# compute variance
d %>% tfd_variance()
# compute probability
d %>% tfd_prob(2.3)
```

github.com/rstudio/tfprobability

TensorFlow - What's next? TF 2.0

 [rstudio / tensorflow](#)  build passing

[Current](#) [Branches](#) [Build History](#) [Pull Requests](#) More options 

✓ [Pull Request #356](#) A simple test that TF is installed.

⌚ Commit 246f3c1 ↗
🏷 #356: A simple test that TF is installed. ↗
/Branch master ↗

⌚ #906 passed
⌚ Ran for 7 min 34 sec
⌚ Total time 29 min 46 sec
⌚ about 3 hours ago

 Jon Harmon

[Build jobs](#) [View config](#)

#	Job Type	Description	Duration	Action
906.1	TensorFlow (Stable)	TensorFlow (Stable)	5 min 15 sec	
906.2	TensorFlow Eager (Stable)	TensorFlow Eager (Stable)	5 min 58 sec	
906.3	Tensorflow (2.0.0-rc0)	Tensorflow (2.0.0-rc0)	3 min 32 sec	
906.4	Tensorflow (1.14)	Tensorflow (1.14)	4 min 14 sec	
906.5	TensorFlow (release version) with reticulate master	TensorFlow (release version) with reticulate master	5 min 48 sec	

TensorFlow - Next? - Distributed

Distributed Training in TensorFlow



Run in Google Colab



View source on GitHub

Overview

`tf.distribute.Strategy` is a TensorFlow API to distribute training across multiple GPUs, multiple machines or TPUs. Using this API, users can distribute their existing models and training code with minimal code changes.

`tf.distribute.Strategy` has been designed with these key goals in mind:

- * Easy to use and support multiple user segments, including researchers, ML engineers, etc.
- * Provide good performance out of the box.
- * Easy switching between strategies.

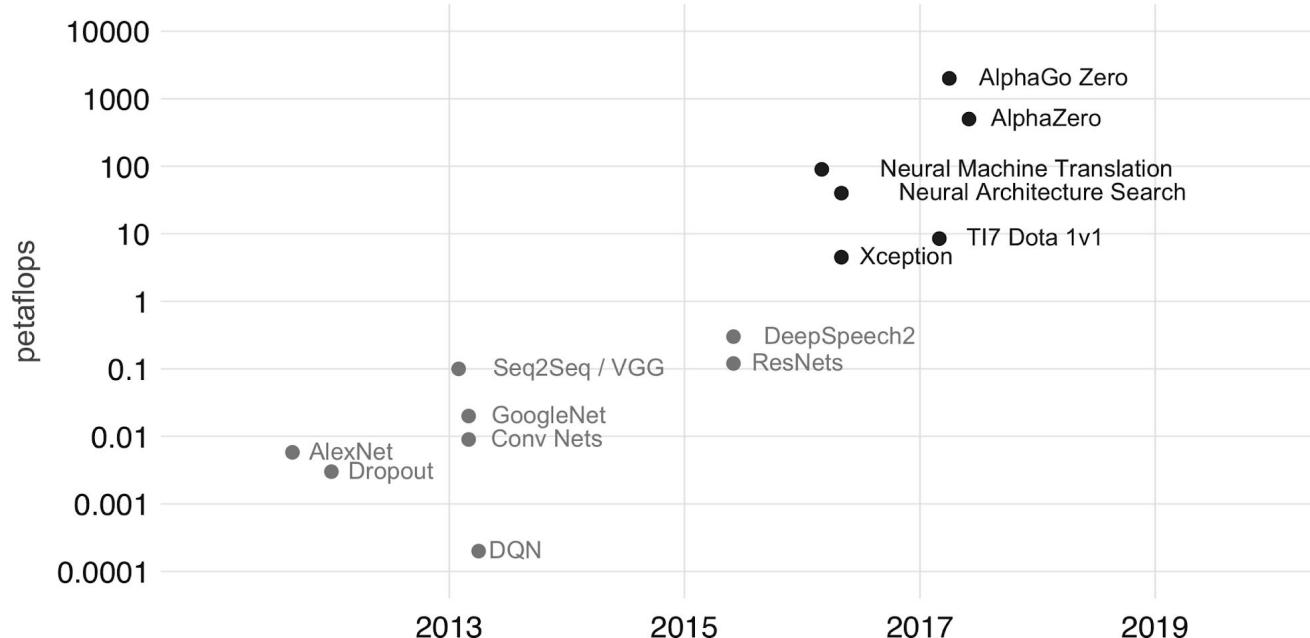
`tf.distribute.Strategy` can be used with TensorFlow's high level APIs, `tf.keras` and `tf.estimator`, with just a couple of lines of code change. It also provides an API that can be used to distribute custom training loops (and in general any computation using TensorFlow). In TensorFlow 2.0, users can execute their programs eagerly, or in a graph using `tf.function`. `tf.distribute.Strategy` intends to support both these modes of execution. Note that we may talk about training most of the time in this guide, but this API can also be used for distributing evaluation and prediction on different platforms.



Scaling Deep Learning

Distributed Deep Learning

Training using distributed systems based on OpenAI analysis



- distributed
- local



TensorFlow

spark.rstudio.com

sparklyr from R Studio

dplyr MLib Extensions Streaming News Reference Blog 🔍

Using sparklyr

- Configuring connections
- Troubleshooting

Guides

- Manipulating data
- Machine Learning
- Understanding Caching
- Deployment Options
- Distributed R
- Data Lakes
- ML Pipelines
- Text mining
- Stream Analysis
- Apache Arrow

Extend sparklyr

- Using H2O
- Graph Analysis

sparklyr: R interface for Apache Spark

build passing CRAN 1.0.3 codecov 80% chat on gitter

- Connect to [Spark](#) from R. The sparklyr package provides a complete [dplyr](#) backend.
- Filter and aggregate Spark datasets then bring them into R for analysis and visualization.
- Use Spark's distributed [machine learning](#) library from R.
- Create [extensions](#) that call the full Spark API and provide interfaces to Spark packages.



Installation

You can install the `sparklyr` package from CRAN as follows:

```
install.packages("sparklyr")
```

You should also install a local version of Spark for development purposes:

```
library(sparklyr)
spark_install(version = "2.1.0")
```

To upgrade to the latest version of `sparklyr`, run the following command and restart your r session:

```
devtools::install_github("rstudio/sparklyr")
```

Spark with R - Philosophy



```
library(sparklyr)

sc <- spark_connect(master = "local|yarn|mesos|spark|livy")

flights <- copy_to(sc, flights)
```



```
library(tidyverse)

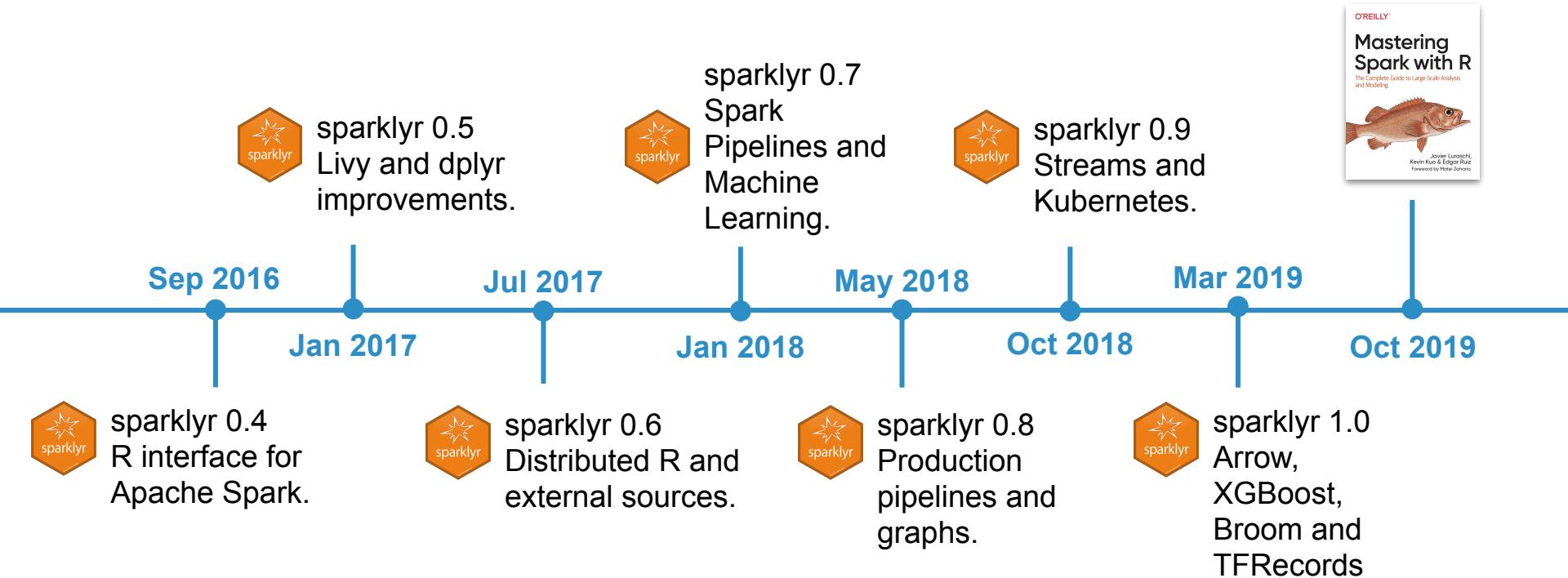
flights %>%

  group_by(month, day) %>%

  summarise(count = n(), avg_delay = mean(dep_delay, na.rm = TRUE)) %>%

  filter(count > 1000)
```

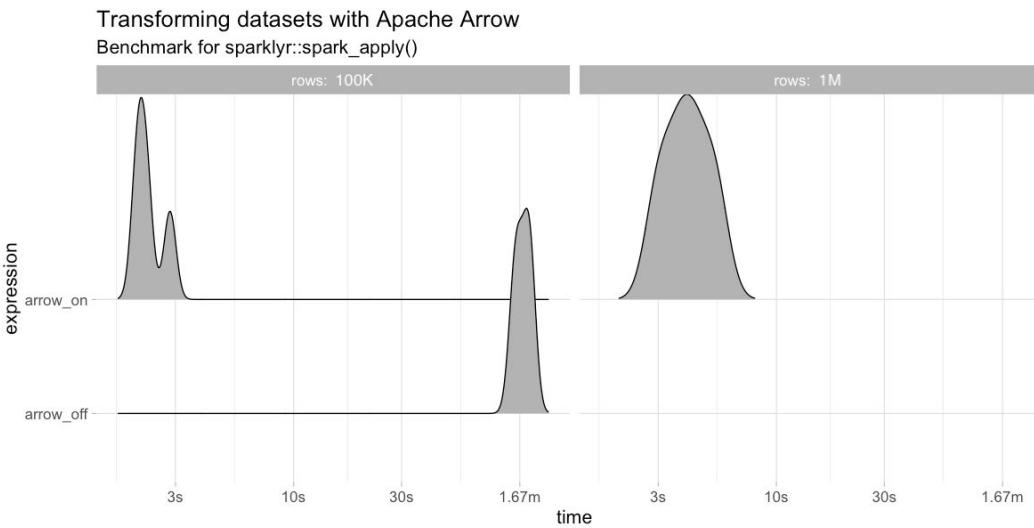
Spark with R - Timeline



Spark - What's new?



```
library(sparklyr)
library(arrow)
```



Spark - What's new? - XGBoost

```
library(sparkxgb)
iris <- copy_to(sc, iris)

xgb_model <- xgboost_classifier(iris, Species ~ ., num_class = 3, num_round = 50, max_depth = 4)

xgb_model %>% ml_predict(iris) %>%
  select(Species, predicted_label, starts_with("probability_")) %>% glimpse()
```

```
#> Observations: ???
#> Variables: 5
#> Database: spark_connection
#> $ Species           <chr> "setosa", "setosa", "setosa", "setosa", ...
#> $ predicted_label    <chr> "setosa", "setosa", "setosa", "setosa", ...
#> $ probability_versicolor <dbl> 0.003566429, 0.003564076, 0.003566429, 0...
#> $ probability_virginica   <dbl> 0.001423170, 0.002082058, 0.001423170, 0...
#> $ probability_setosa     <dbl> 0.9950104, 0.9943539, 0.9950104, 0.995010...
```

Spark - What's new? - Broom



```
movies <- data.frame(user = c(1, 2, 0, 1, 2, 0),
                      item = c(1, 1, 1, 2, 2, 0),
                      rating = c(3, 1, 2, 4, 5, 4))

copy_to(sc, movies) %>%
  ml_als(rating ~ user + item) %>%
  augment()
```

```
# Source: spark<?> [?? x 4]
  user item rating .prediction
  <dbl> <dbl> <dbl>      <dbl>
1     2     2     5       4.86
2     1     2     4       3.98
3     0     0     4       3.88
4     2     1     1       1.08
5     0     1     2       2.00
6     1     1     3       2.80
```



TensorFlow

Spark - New? - TF Records

```
library(sparktf)
library(sparklyr)

sc <- spark_connect(master = "local")

copy_to(sc, iris) %>%
  ft_string_indexer_model(
    "Species", "label",
    labels = c("setosa", "versicolor", "virginica")
  ) %>%
  spark_write_tfrecord(path = "tfrecord")
```

Spark - What's next? - Genomics

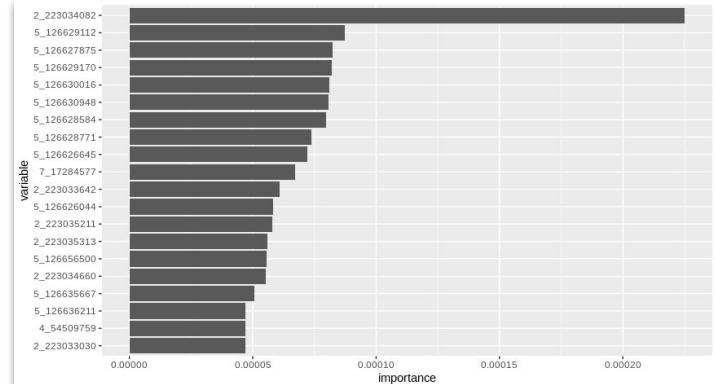
github.com/r-spark/variantspark by Samuel Macêdo

```
library(sparklyr)
library(variantspark)

sc <- spark_connect(master = "local")
vsc <- vs_connect(sc)

hipster_vcf <- vs_read_vcf(vsc, "inst/extdata/hipster.vcf.bz2")
hipster_labels <- vs_read_csv(vsc,
  "inst/extdata/hipster_labels.txt")
labels <- vs_read_labels(vsc, "inst/extdata/hipster_labels.txt")

vs_importance_analysis(vsc, hipster_vcf, labels, n_trees = 100)
```



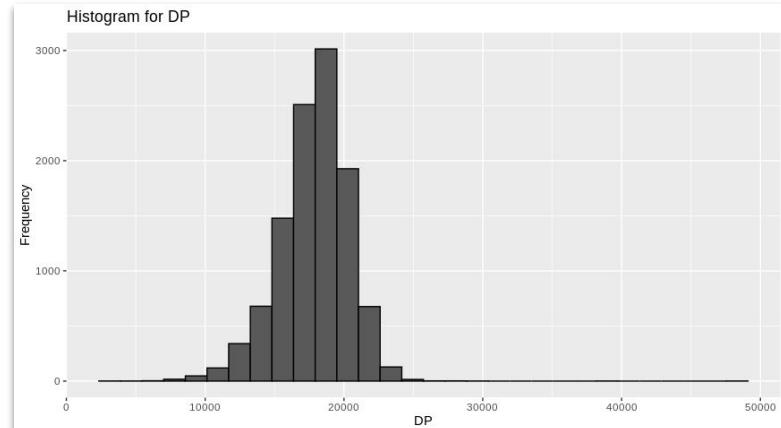
Spark - What's next? - Genomics

github.com/r-spark/sparkhail by Samuel Macêdo

```
library(sparkhail)
sc <- spark_connect(master = "local",
                     version = "2.4",
                     config = hail_config())

hl <- hail_context(sc)
mt <- hail_read_matrix(hl, system.file("extdata/1kg.mt",
                                         package = "sparkhail"))

hail_dataframe(mt)
```



Spark - What's next? - Genomics

github.com/lawreml/hailr by Michael Lawrence

The screenshot shows a GitHub repository page for the project "hailr". At the top, there is a list of recent commits from the user "lawreml". The commits are as follows:

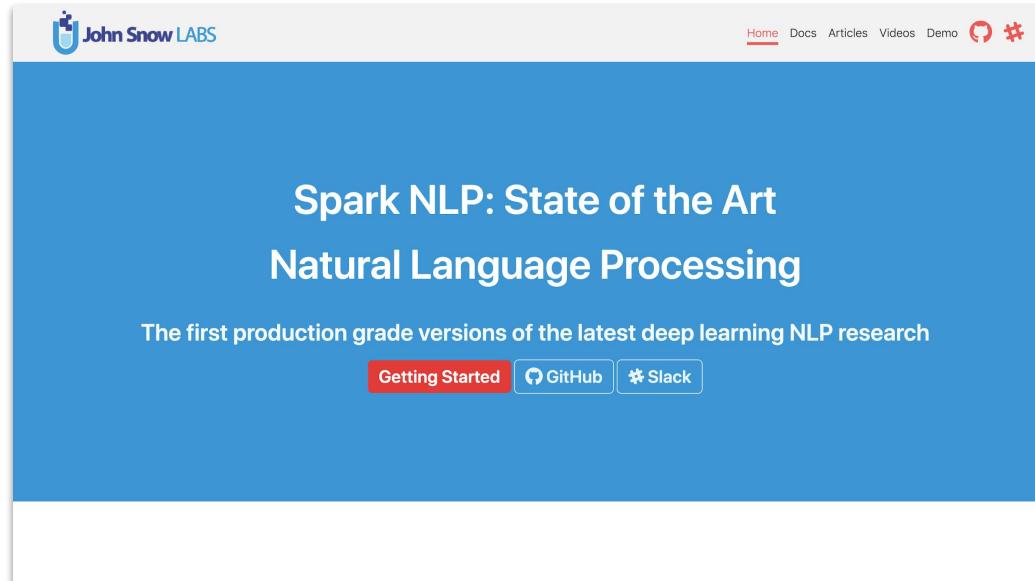
Commit	Message	Date
introduce hailTypeEnv() to avoid dummy HailType gymnastics	Latest commit 6eddac7	14 days ago
R	introduce hailTypeEnv() to avoid dummy HailType gymnastics	14 days ago
inst	field insertion finally works again	3 months ago
man	support list columns (hail arrays)	9 months ago
notes	start on the psuedocode	2 years ago
vignettes	we can now read a VCF into something that does nothing	last year
.gitignore	rename HailHomeManager to simpler HailConfig	last year
DESCRIPTION	resolve the fulfill() vs. eval() war	14 days ago
NAMESPACE	cache table row count	3 months ago
README.md	cleanup README	3 months ago

Below the commits, there is a section titled "hailr" which contains the following text:

The hailr package is a transparent interface between R and Hail, a Spark-based platform for genomic computing. This is a work in progress.

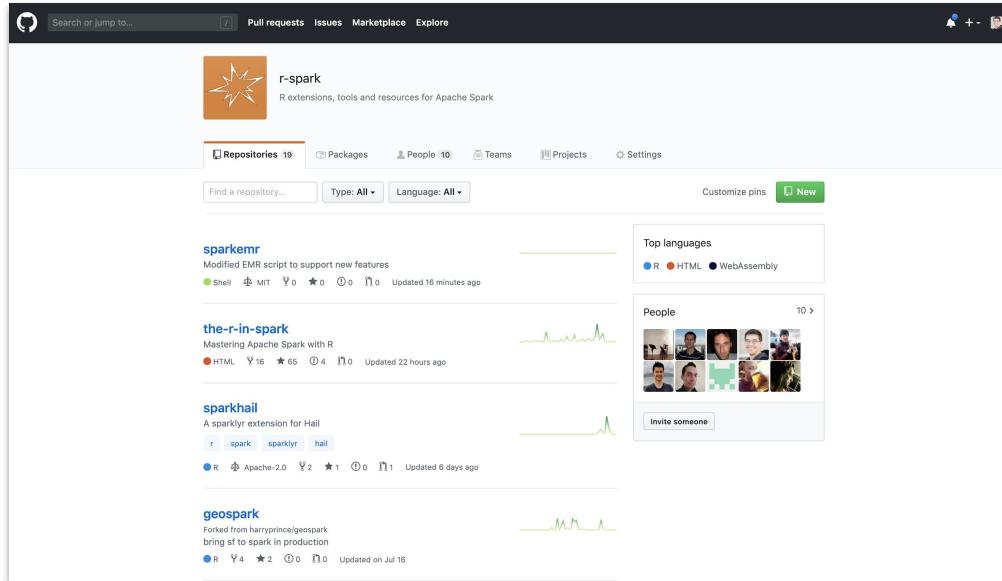
Spark - What's next? - NLP

github.com/r-spark/sparknlp by Kevin Kuo



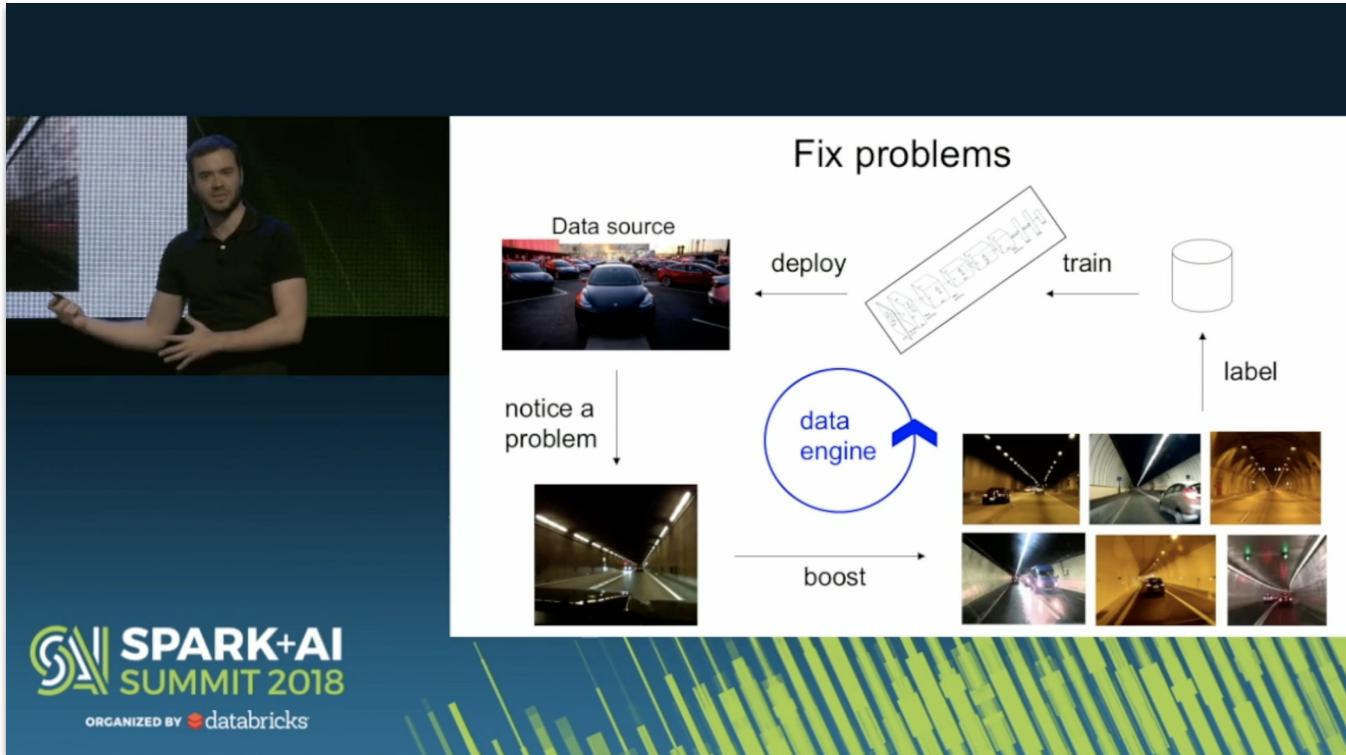
Spark - What's next? - GitHub

New github.com/r-spark organization.

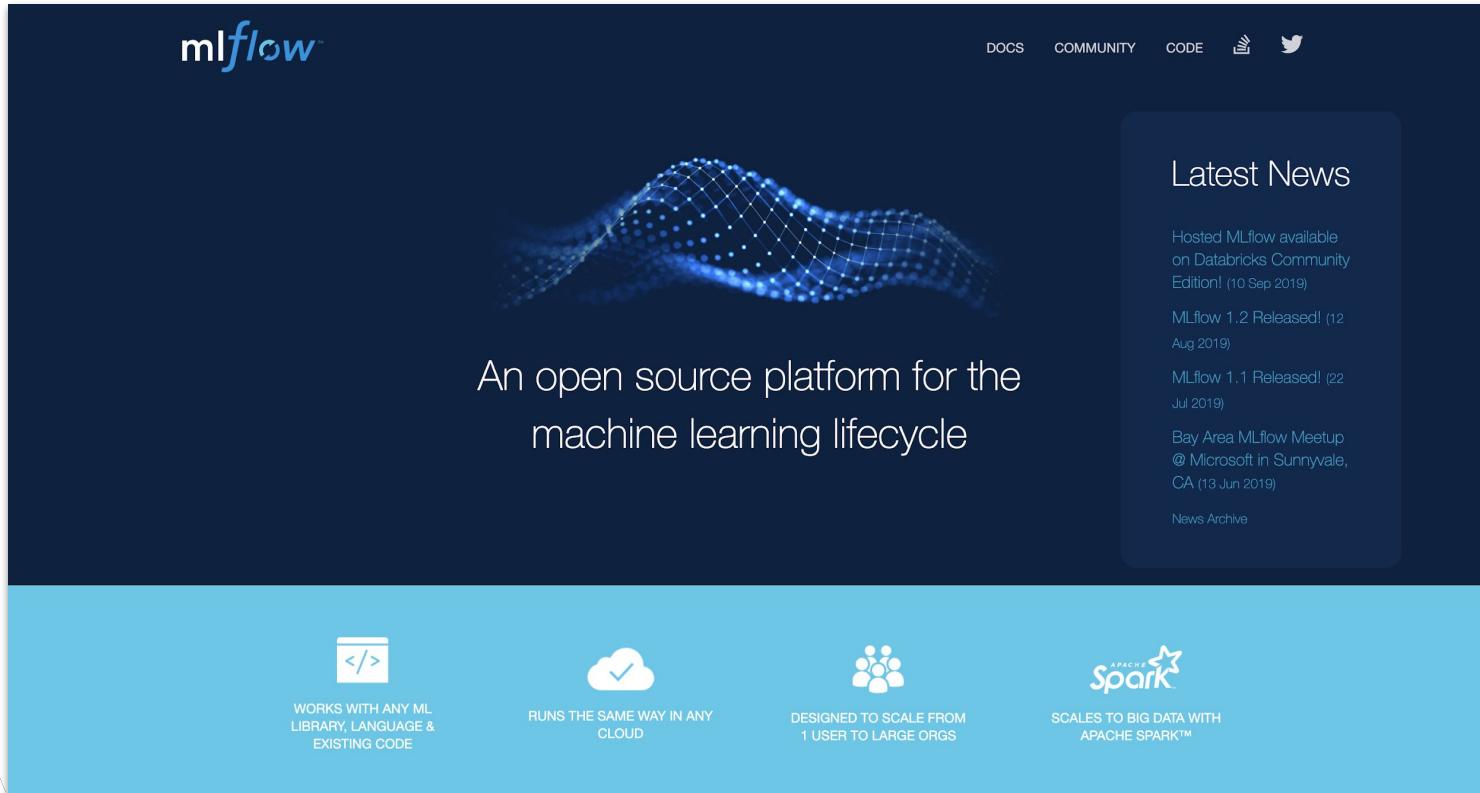


m|flow

MLflow - Software 2.0



MLflow - Intro



The screenshot shows the official MLflow website. At the top, there's a dark blue header with the "mlflow" logo on the left and navigation links for "DOCS", "COMMUNITY", "CODE", and social media icons for LinkedIn and Twitter. Below the header is a large dark blue background image featuring a glowing blue 3D wireframe mesh wave. In the center of this image, white text reads "An open source platform for the machine learning lifecycle". To the right is a dark rounded rectangle containing a "Latest News" section with four news items and a "News Archive" link. At the bottom, a light blue footer bar contains four icons with corresponding text: a code icon for "WORKS WITH ANY ML LIBRARY, LANGUAGE & EXISTING CODE", a cloud icon for "RUNS THE SAME WAY IN ANY CLOUD", a people icon for "DESIGNED TO SCALE FROM 1 USER TO LARGE ORGS", and an Apache Spark logo for "SCALES TO BIG DATA WITH APACHE SPARK™".

mlflow

DOCS COMMUNITY CODE

LinkedIn Twitter

An open source platform for the machine learning lifecycle

Latest News

Hosted MLflow available on Databricks Community Edition! (10 Sep 2019)

MLflow 1.2 Released! (12 Aug 2019)

MLflow 1.1 Released! (22 Jul 2019)

Bay Area MLflow Meetup @ Microsoft in Sunnyvale, CA (13 Jun 2019)

[News Archive](#)

 WORKS WITH ANY ML LIBRARY, LANGUAGE & EXISTING CODE

 RUNS THE SAME WAY IN ANY CLOUD

 DESIGNED TO SCALE FROM 1 USER TO LARGE ORGS

 SCALES TO BIG DATA WITH APACHE SPARK™

MLflow - Components

MLflow is an open source platform to manage the ML lifecycle, including experimentation, reproducibility and deployment. It currently offers three components:

MLflow Tracking

Record and query experiments: code, data, config, and results.

[Read more](#)

MLflow Projects

Packaging format for reproducible runs on any platform.

[Read more](#)

MLflow Models

General format for sending models to diverse deployment tools.

[Read more](#)

MLflow - Timeline

Available in CRAN since v0.7.0

```
mlflow: Interface to 'MLflow'

R interface to 'MLflow', open source platform for the complete machine learning life cycle, see <https://mlflow.org>. This package supports installing 'MLflow', tracking experiments, creating and running projects, and saving and serving models.

Version: 1.2.0
Depends: R (>= 3.3.0)
Imports: base64enc, forage, fs, git2r, httpuv, httr, ini, jsonlite, openssl, processx, reticulate, purrr, swagger, withr, xml2, yaml, rlang (>= 0.2.0), zeallot, tibble (>= 2.0.0), glue
Suggests: covr, carrier, keras, lintr, testthat
Published: 2019-08-19
Author: Matei Zaharia [aut, cre], Javier Luraschi [aut], Kevin Kuo [aut], RStudio [cph]
Maintainer: Matei Zaharia <matei at databricks.com>
BugReports: https://github.com/mlflow/mlflow/issues
License: Apache License 2.0
URL: https://github.com/mlflow/mlflow
NeedsCompilation: no
SystemRequirements: MLflow (https://www.mlflow.org/)
Materials: README
CRAN checks: mlflow results

Downloads:

Reference manual: mlflow.pdf
Package source: mlflow\_1.2.0.tar.gz
Windows binaries: r-devel: mlflow\_1.2.0.zip, r-release: mlflow\_1.2.0.zip, r-oldrel: mlflow\_1.2.0.zip
OS X binaries: r-release: mlflow\_1.2.0.tgz, r-oldrel: mlflow\_1.2.0.tgz
Old sources: mlflow archive

Linking:

Please use the canonical form https://CRAN.R-project.org/package=mlflow to link to this page.
```

Index of /src/contrib/Archive/mlflow

Name	Last modified	Size	Description
Parent Directory			
mlflow_0.7.0.tar.gz	2018-10-06 23:30	598K	
mlflow_0.8.0.tar.gz	2018-11-12 23:00	597K	
mlflow_0.9.0.1.tar.gz	2019-04-14 07:12	603K	
mlflow_0.9.0.tar.gz	2019-03-28 19:30	601K	
mlflow_1.0.0.tar.gz	2019-06-06 09:50	603K	
mlflow_1.1.0.tar.gz	2019-07-23 18:52	605K	

Apache Server at cran.r-project.org Port 443

MLflow - Docs

mlflow.org/docs/latest/index.html

The screenshot shows the MLflow documentation website at mlflow.org/docs/latest/index.html. The page title is "Quickstart". The left sidebar contains a navigation menu with sections like "MLflow", "Tutorial", "Concepts", "MLflow Tracking", "MLflow Projects", "MLflow Models", "Command-Line Interface", "Search", "Python API", "R API", "Java API", and "REST API". The main content area starts with the heading "Installing MLflow". It includes a code snippet for installing MLflow using Python and R:

```
install.packages("mlflow")
mlflow::install_mlflow()
```

Below the code, there is a note in a blue bar:

Note
You cannot install MLflow on the MacOS system installation of Python. We recommend installing Python 3 through the [Homebrew](#) package manager using `brew install python`. (In this case, installing MLflow is now `pip3 install mlflow`).

At the bottom, it says: "At this point we recommend you follow the [Tutorial](#) for a walk-through on how you can leverage MLflow in your daily workflow."

MLflow - Using

```
library(mlflow)
with(mlflow_start_run(), {
    # Log a parameter (key-value pair)
    mlflow_log_param("param1", 5)

    # Log a metric; metrics can be updated throughout the run
    mlflow_log_metric("foo", 1)

    # Log an artifact (output file)
    writeLines("Hello world!", "output.txt")
    mlflow_log_artifact("output.txt")
})
```

MLflow- UI

RStudio Viewer

mlflow GitHub Docs

Experiments Default

Default Experiment ID: 0 Artifact Location: /Users/javierluraschi/RStudio/talks/mlflow/mlruns/0

Search Runs: metrics.rmse < 1 and params.model = "tree" Search

Filter Params: alpha, lr Filter Metrics: rmse, r2 Clear

1 matching run Compare Selected Download CSV

	Date ▼	User	Source	Version	Parameters	Metrics
<input type="checkbox"/>	2018-09-19 17:12:02	javierluraschi	talks#mlflow:R/tracking.R	a26eb2	param1 foo	5 3

MLflow - What's next?

- renv (packrat successor)
- Cloud Deployment Targets
- Keras Autolog

Thanks!

