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

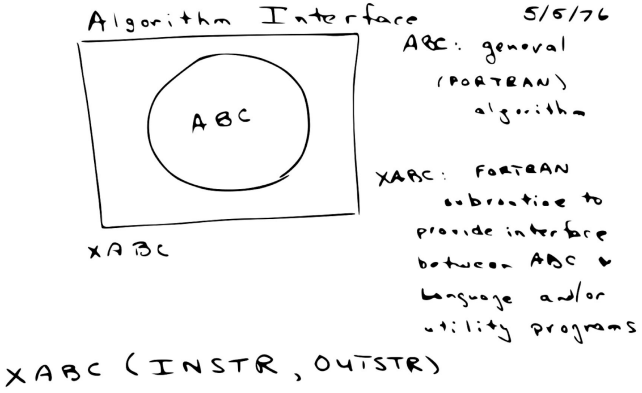
 javierluraschi

javier@rstudio.com

Intro to R

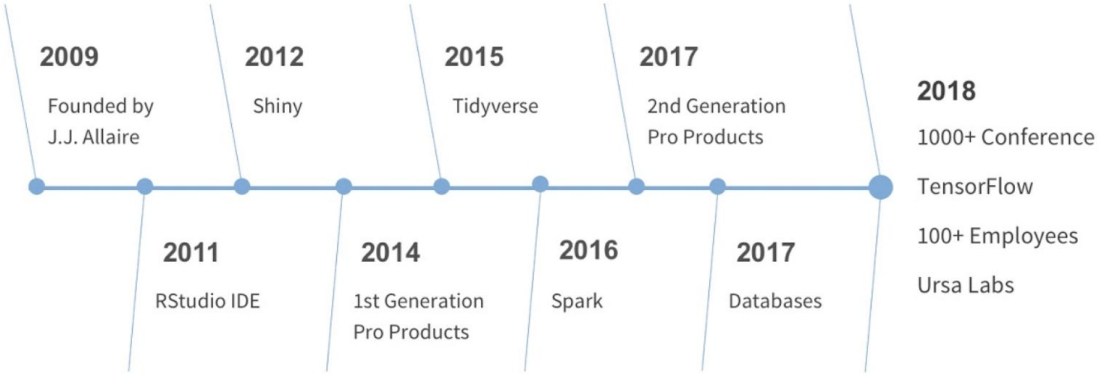
Josh Wills
@josh_wills

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.

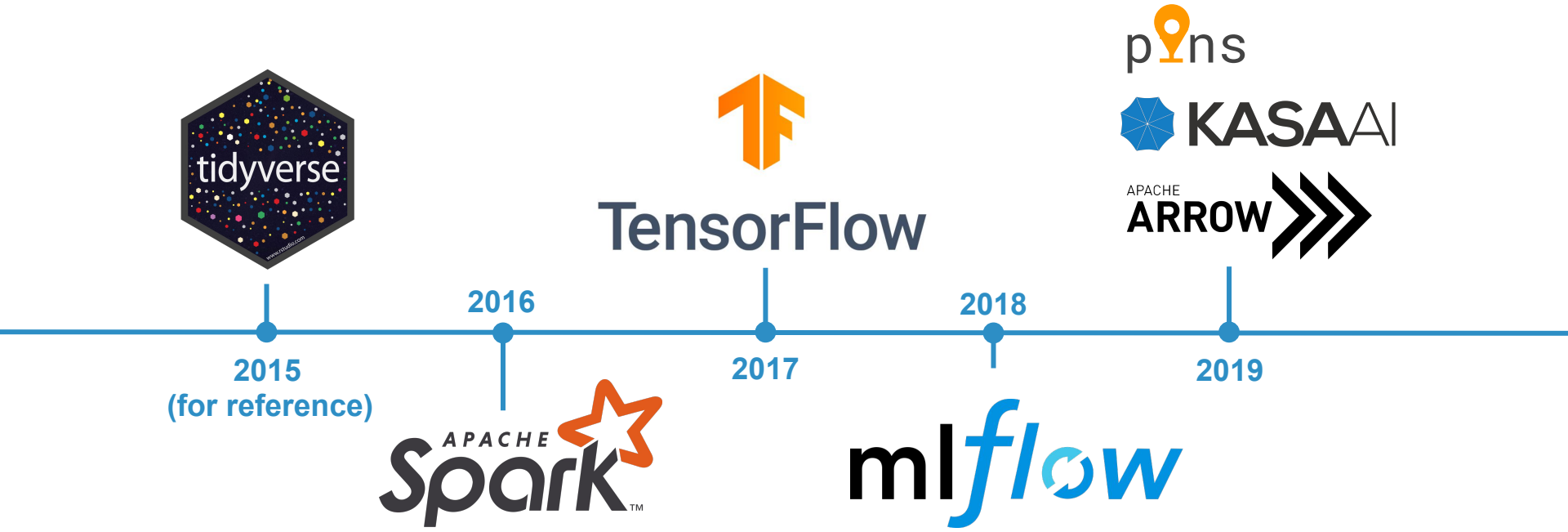
Daniel Falbel
@dfalbel

Sigrid Keydana
@zkajdan

Kevin Kuo
@kevinykuo

Javier Luraschi
@javierluraschi

Multiverse - Timeline



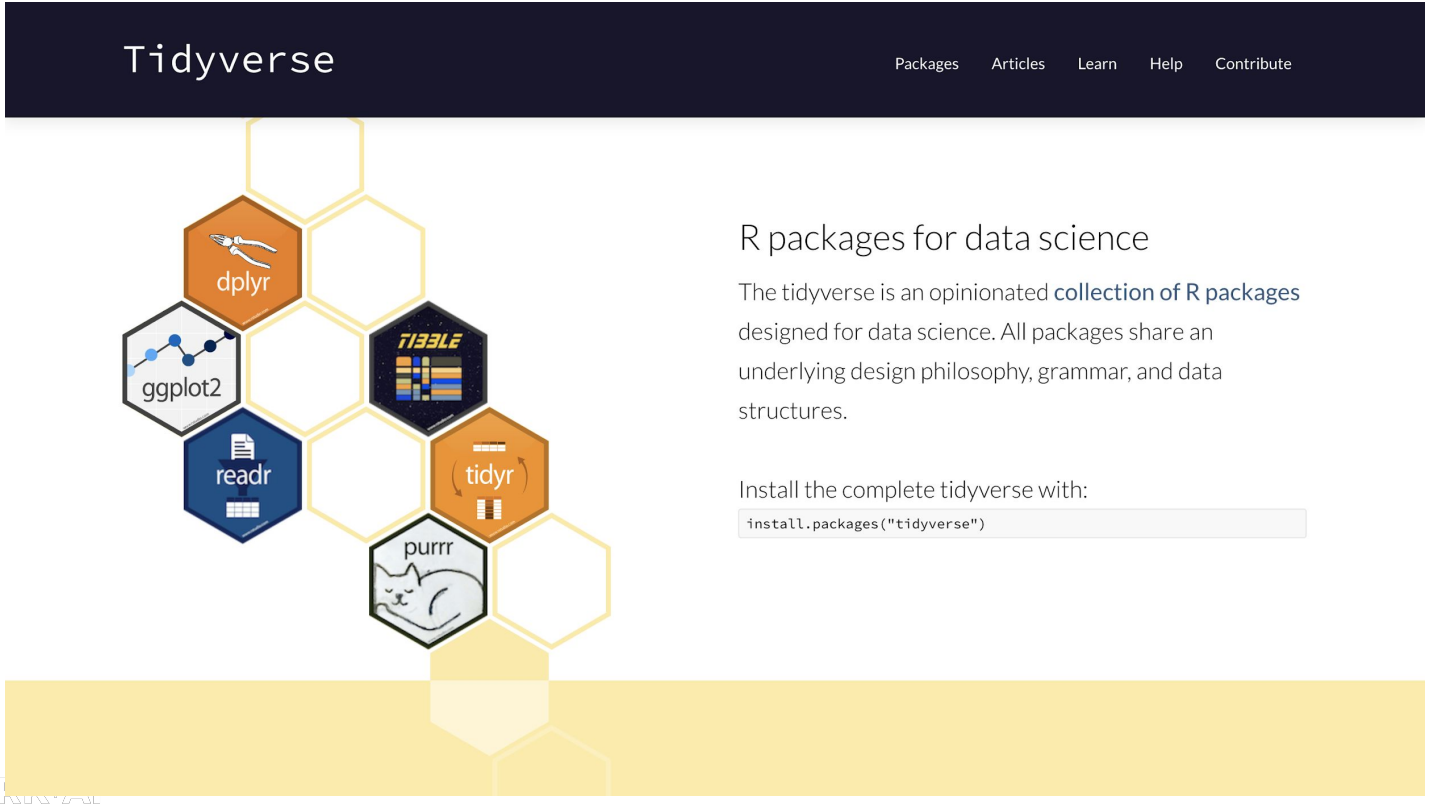

tidyverse
www.rstudio.com

Tidyverse - Modern R

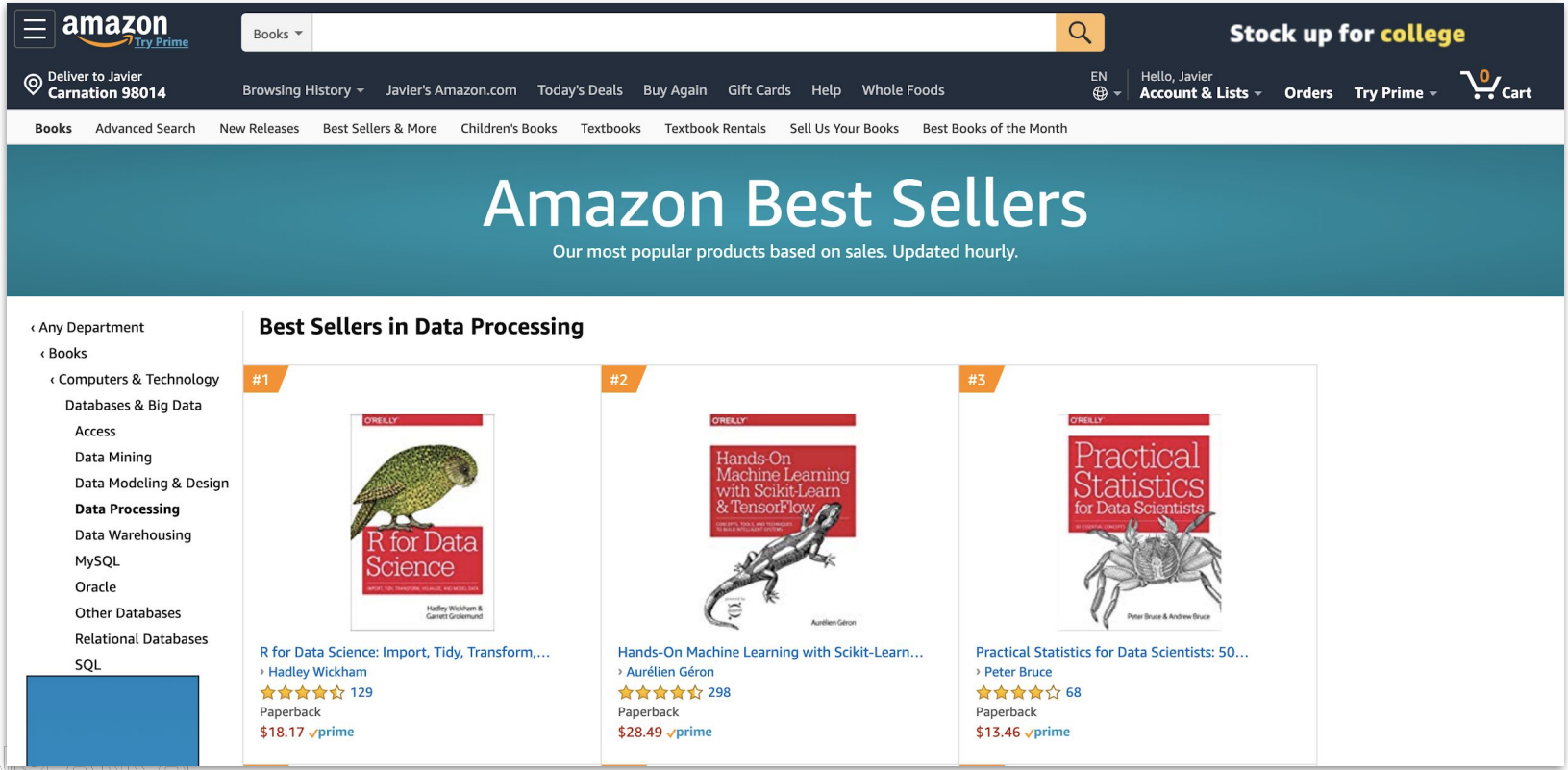
The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.


```
library(tidyverse)

flights %>%

  group_by(month, day) %>%

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

  filter(count > 1000)
```

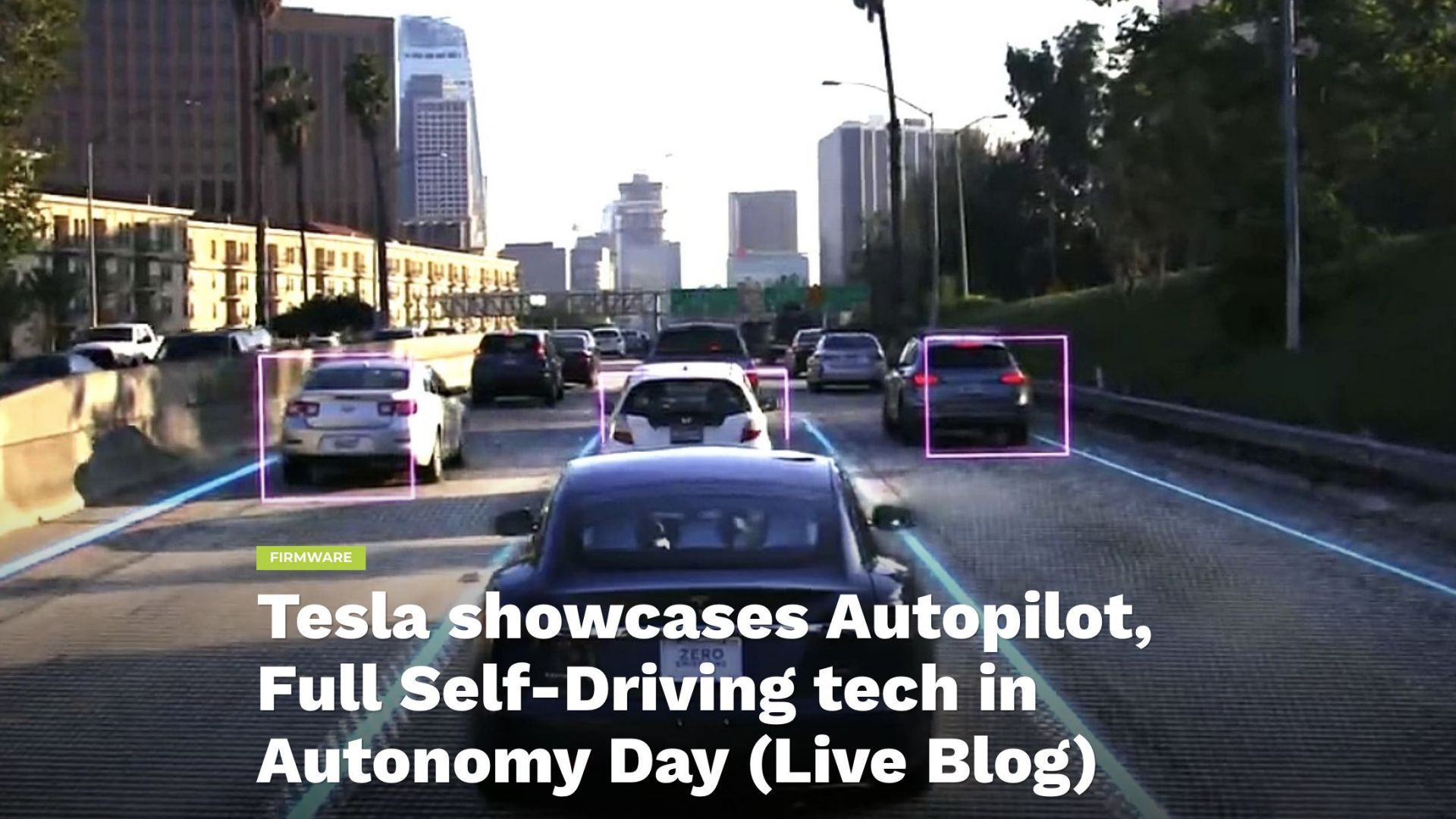
Tidyverse - Modern R



Tidyverse - Modern R



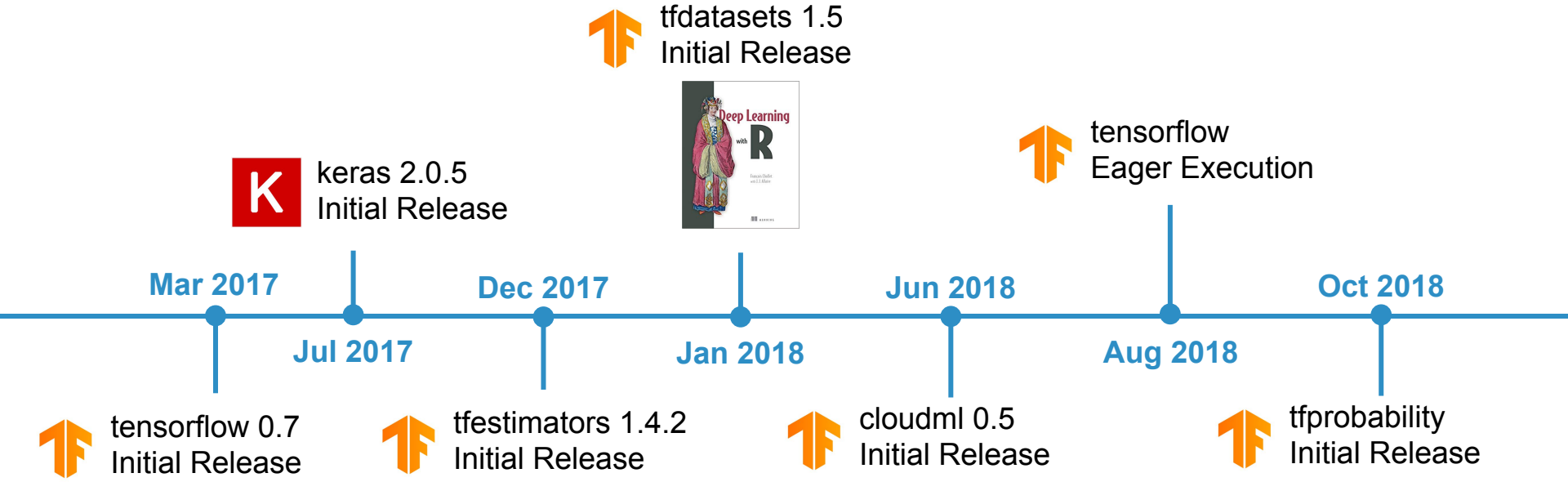
TensorFlow



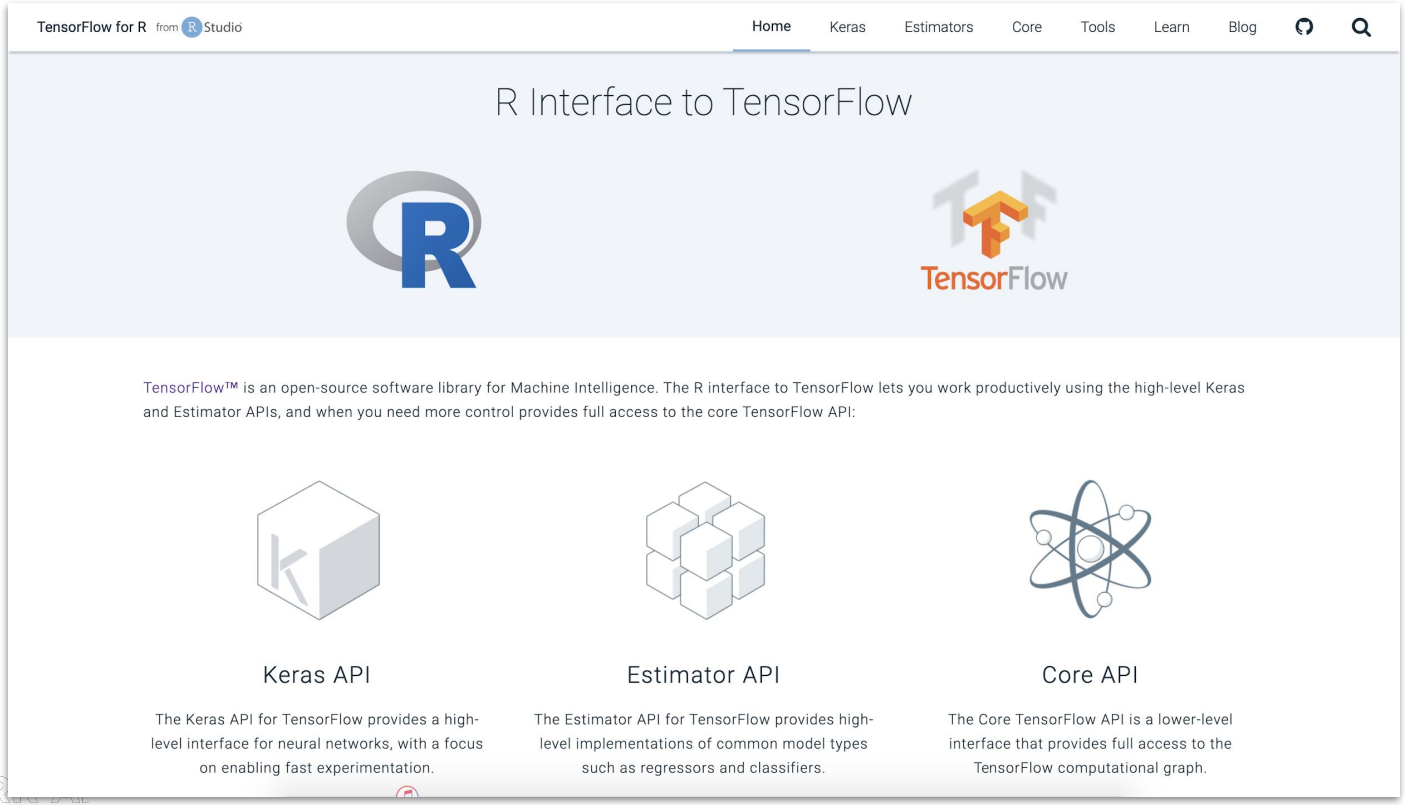
FIRMWARE

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

TensorFlow with R - Timeline



tensorflow.rstudio.com



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()
```

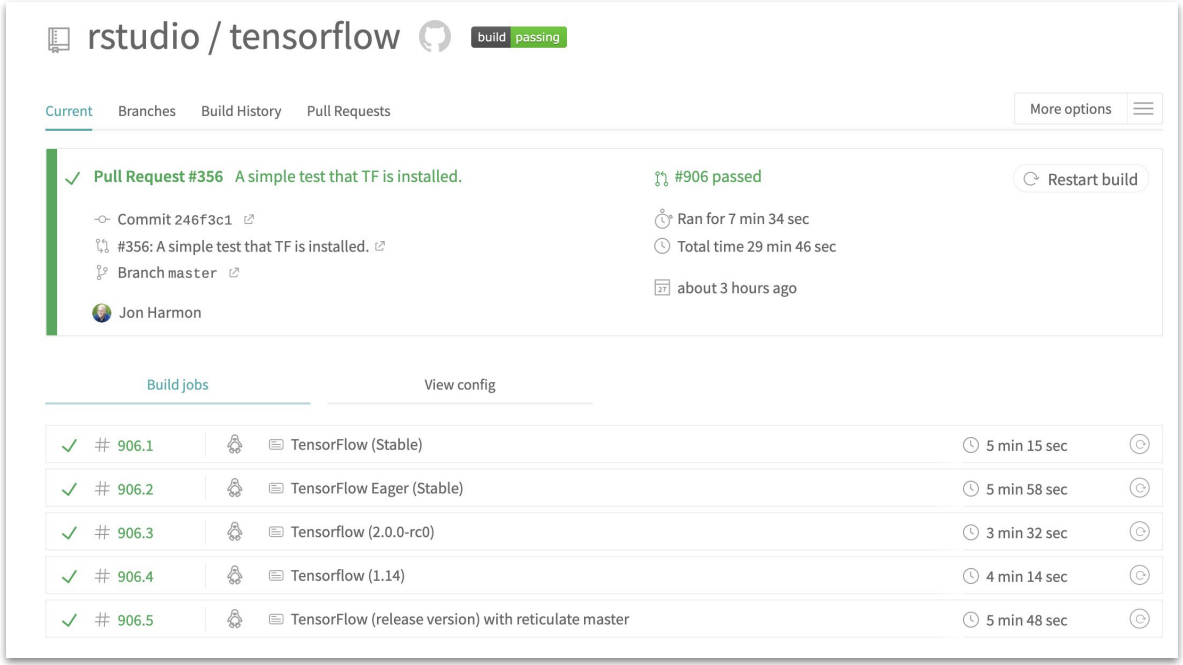
TensorFlow - New? - tfprobability

Combine probabilistic models and deep learning on modern hardware

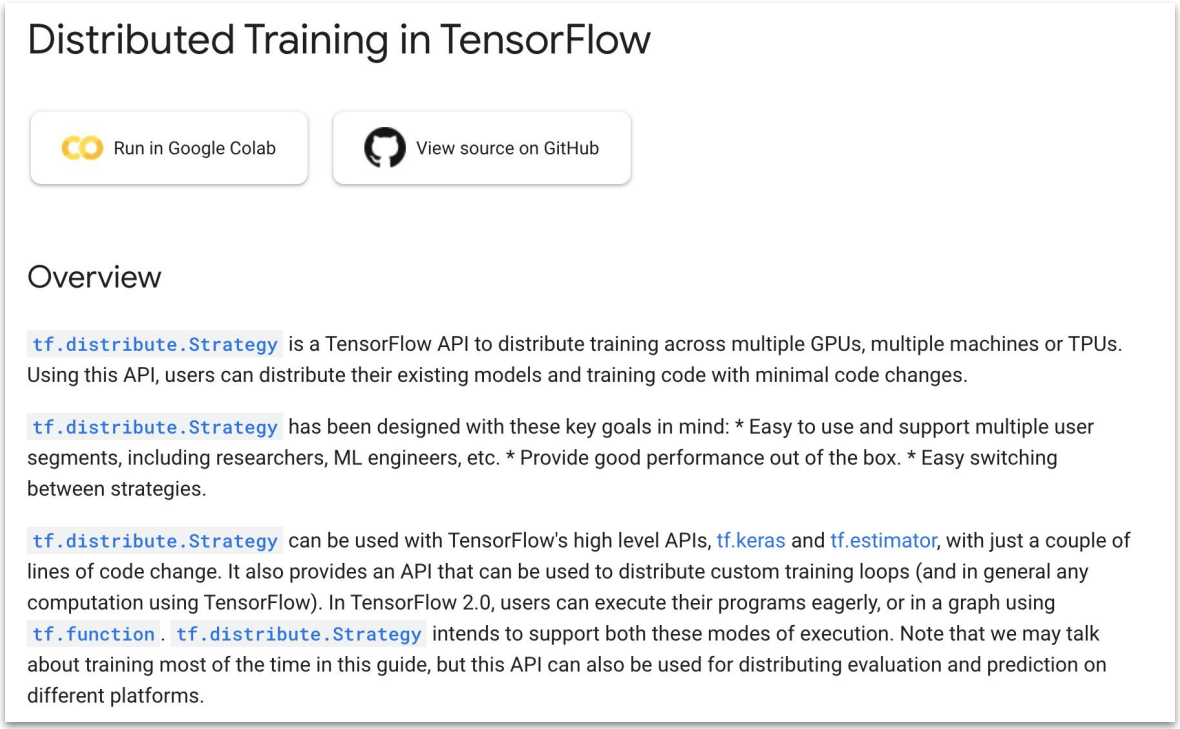
```
# create a binomial distribution with n = 7 and p = 0.3
d <- tfd_binomial(total_count = 7, probs = 0.3)
# compute mean
d %>% tfd_mean()
# compute variance
d %>% tfd_variance()
# compute probability
d %>% tfd_prob(2.3)
```

github.com/rstudio/tfprobability

TensorFlow - What's next? TF 2.0



TensorFlow - Next? - Distributed



Distributed Training in TensorFlow

Run in Google Colab

View source on GitHub

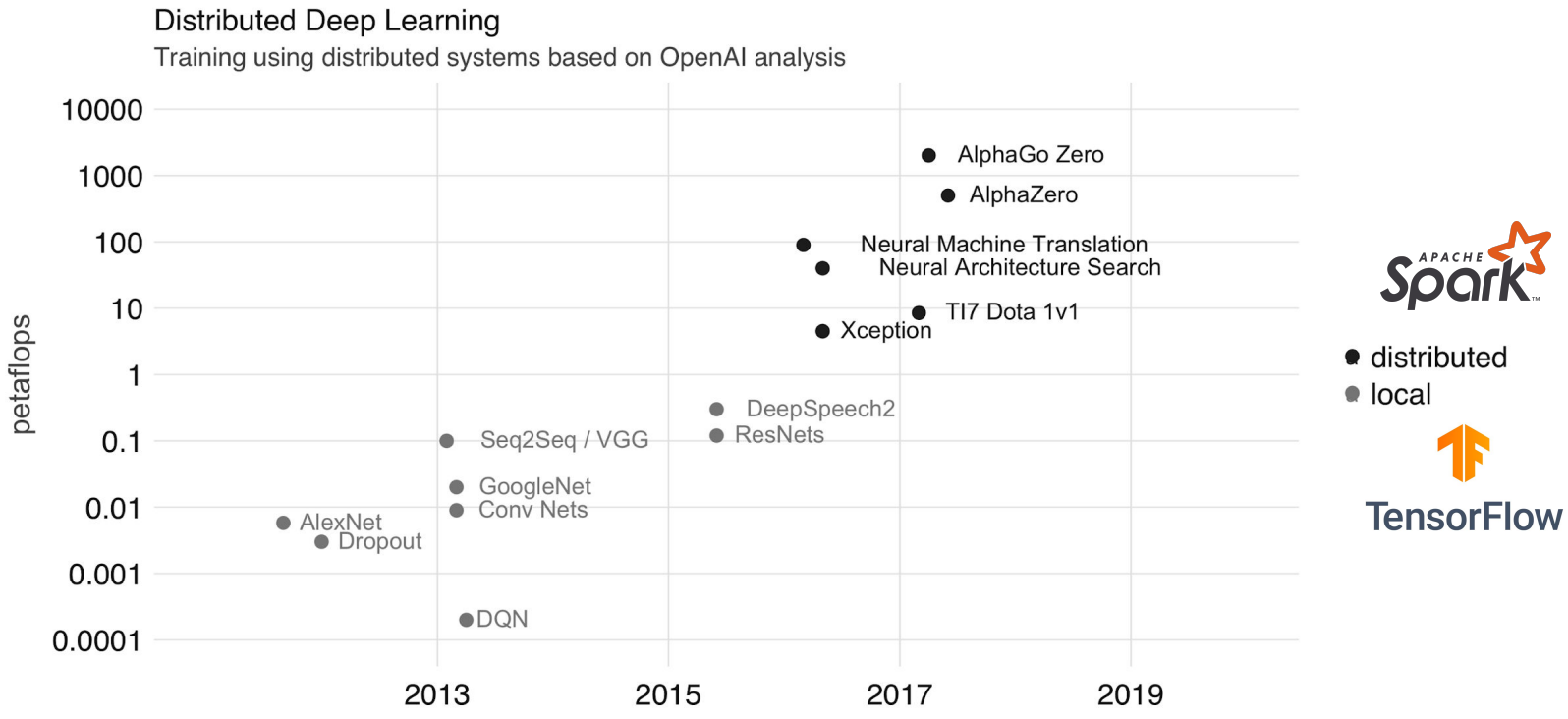
Overview

`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.

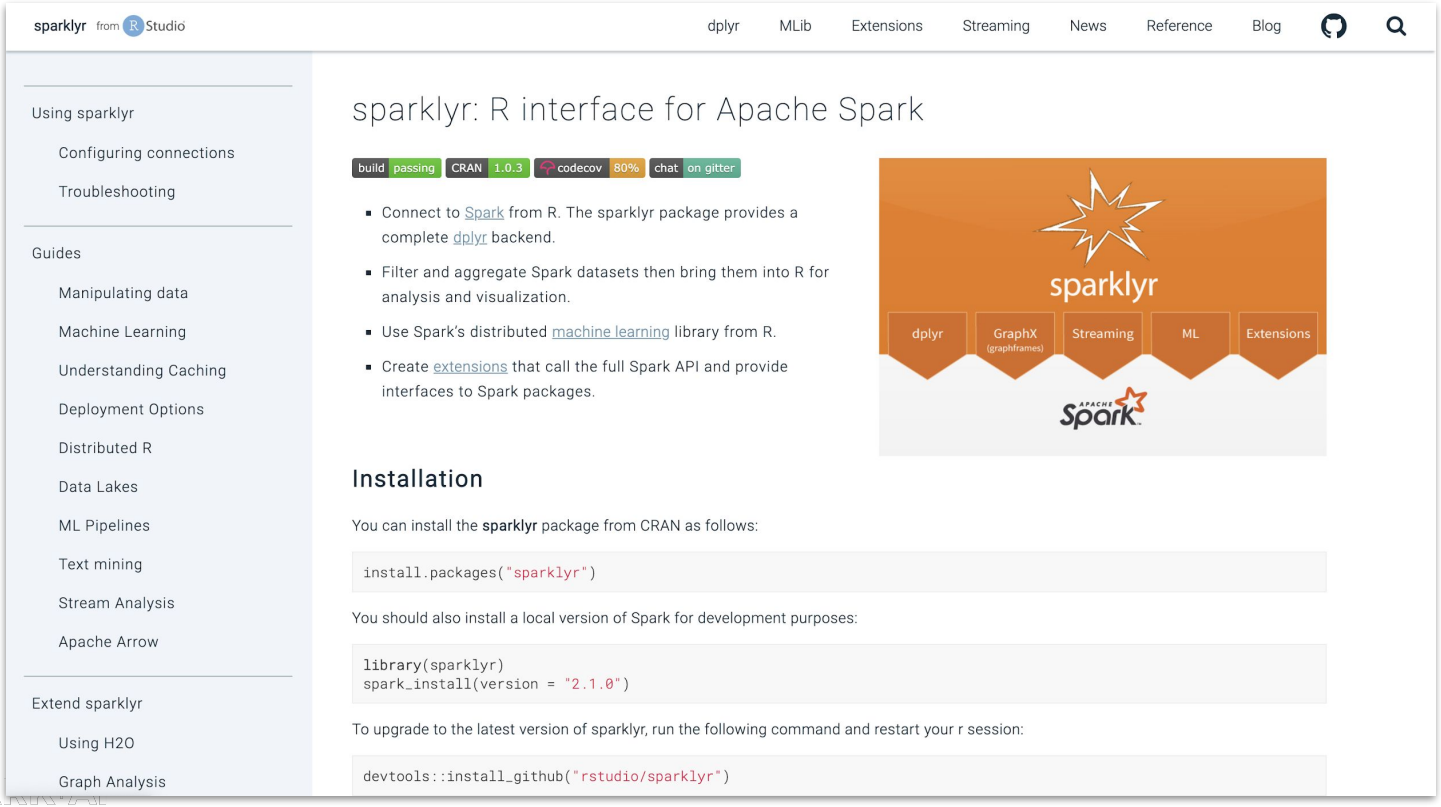
`tf.distribute.Strategy` has been designed with these key goals in mind: * Easy to use and support multiple user segments, including researchers, ML engineers, etc. * Provide good performance out of the box. * Easy switching between strategies.

`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



spark.rstudio.com



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

Sep 2016 — sparklyr 0.4
R interface for Apache Spark.

Jan 2017 — sparklyr 0.5
Livy and dplyr improvements.

Jul 2017 — sparklyr 0.6
Distributed R and external sources.

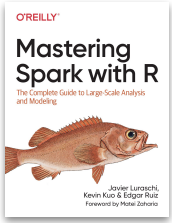
Jan 2018 — sparklyr 0.7
Spark Pipelines and Machine Learning.

May 2018 — sparklyr 0.8
Production pipelines and graphs.

Oct 2018 — sparklyr 0.9
Streams and Kubernetes.

Mar 2019 — sparklyr 1.0
Arrow, XGBoost, Broom and TFRecords

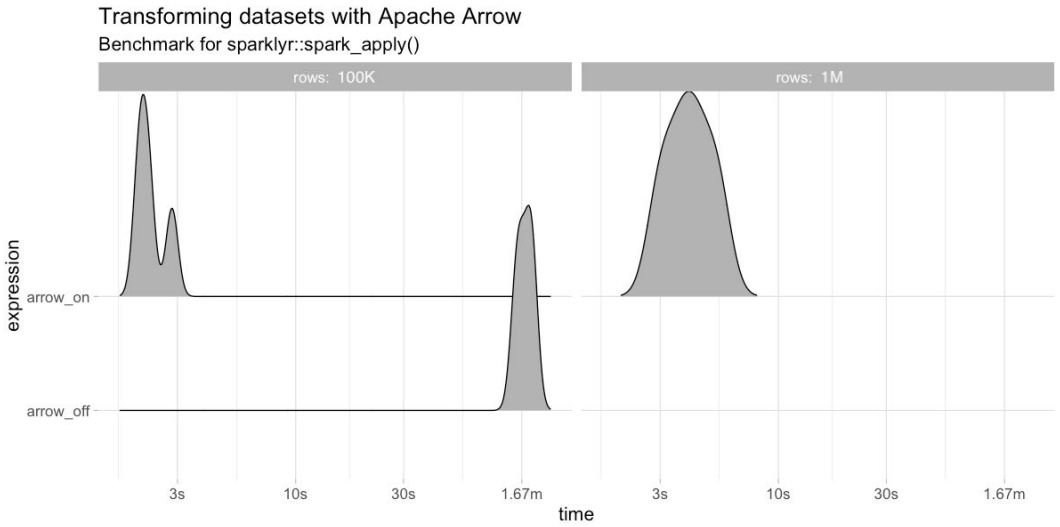
Oct 2019



Spark - What's new?

APACHE ARROW

```
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
```

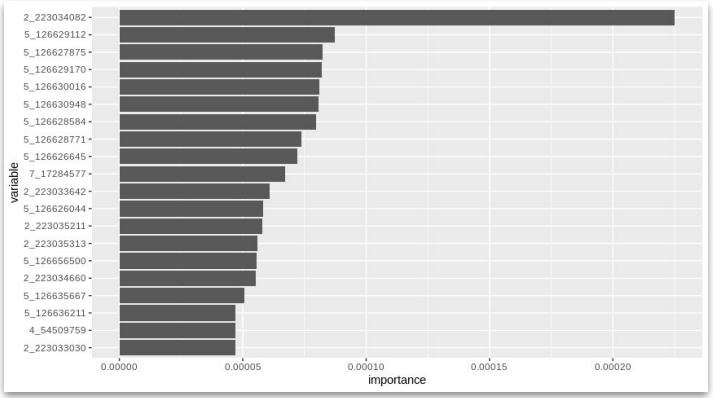
Spark - New? - TF Records


```
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

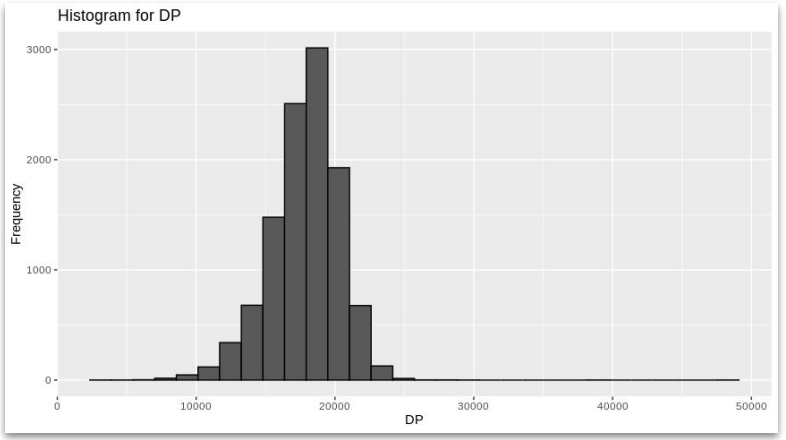
```
library(sparklyr)
library(variantspark)

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

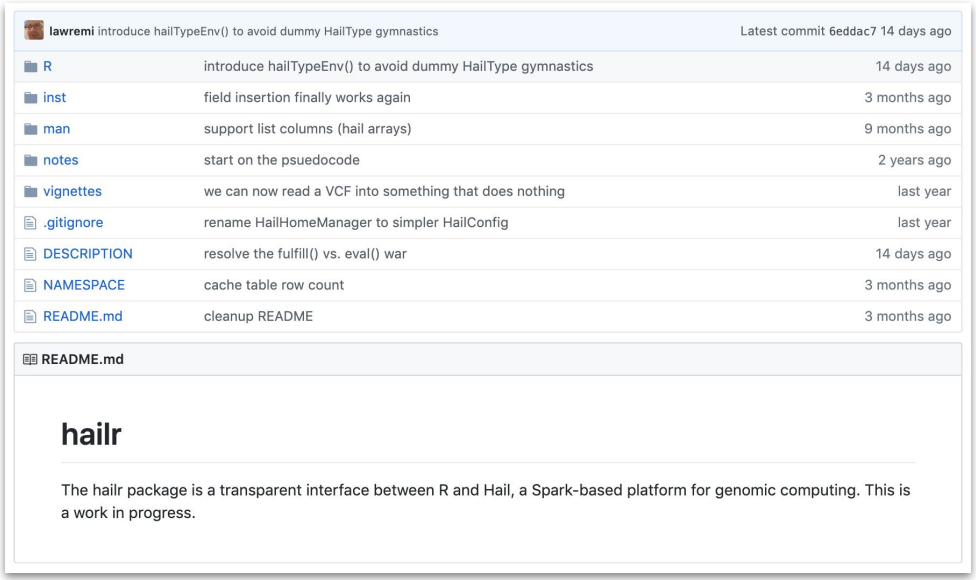
github.com/r-spark/sparkhail by Samuel Macêdo

```
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/lawremi/hailr by Michael Lawrence



lawremi 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

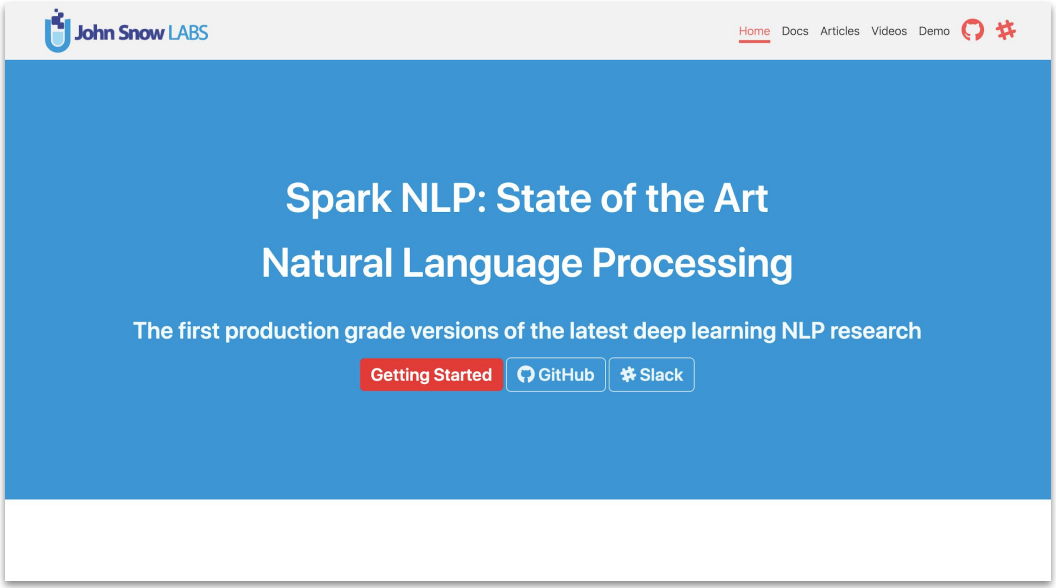
README.md

hailr

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

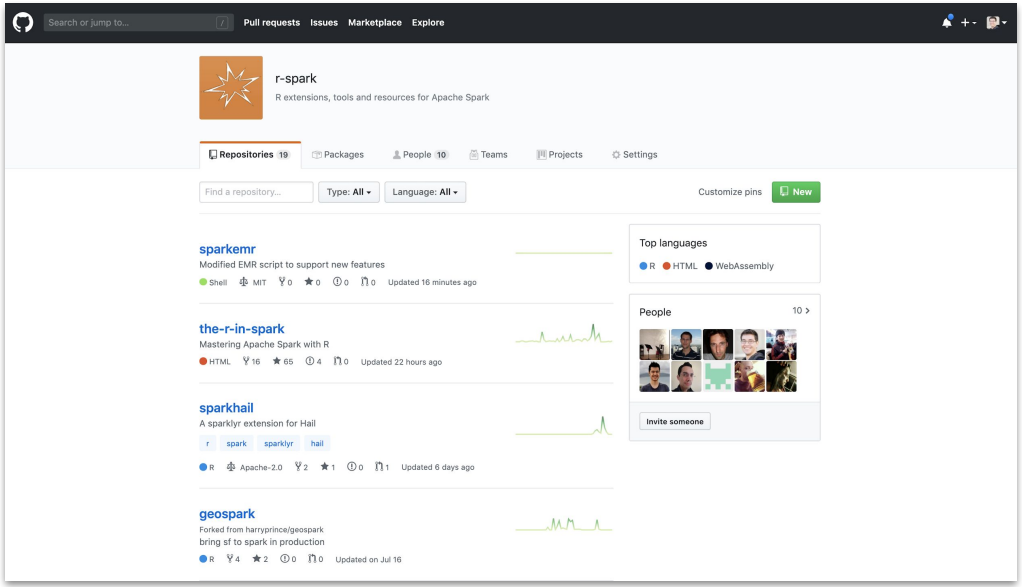
Spark - What's next? - NLP

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



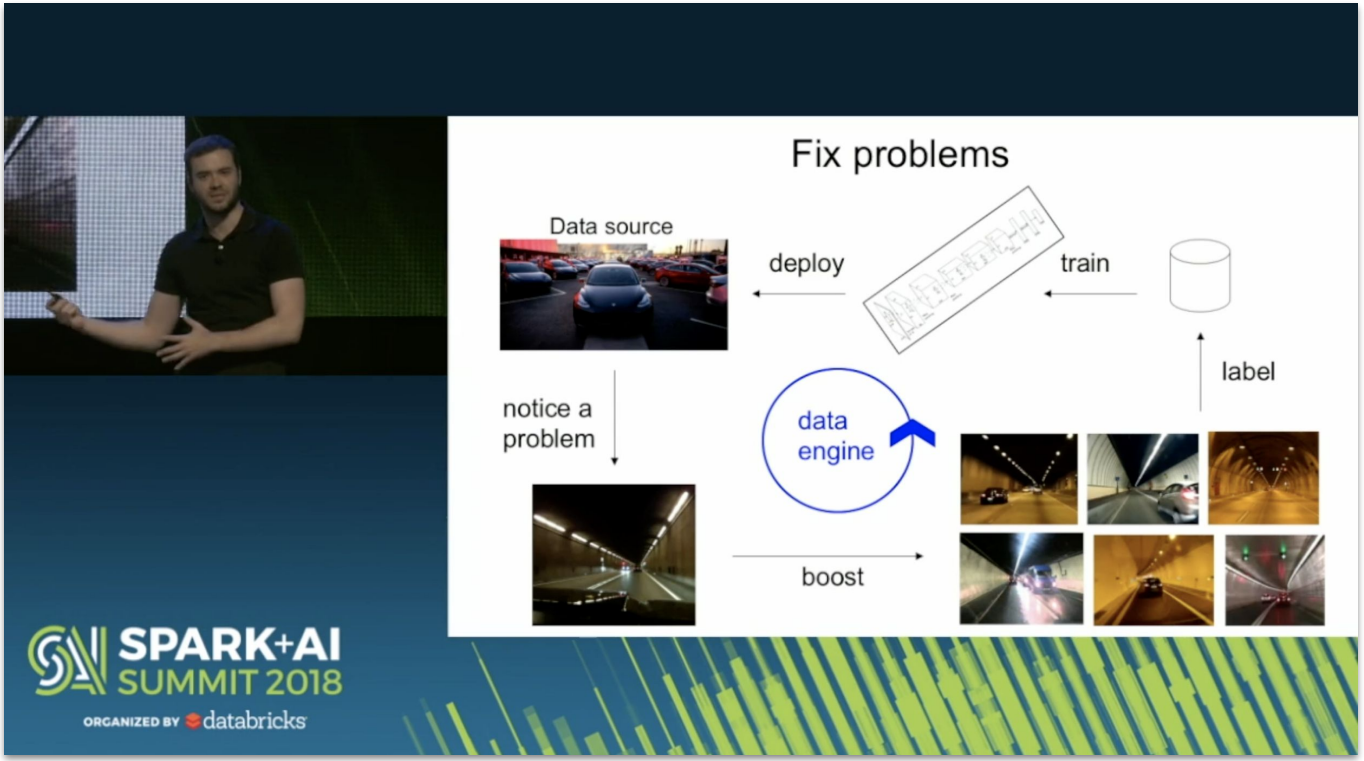
Spark - What's next? - GitHub

New github.com/r-spark organization.

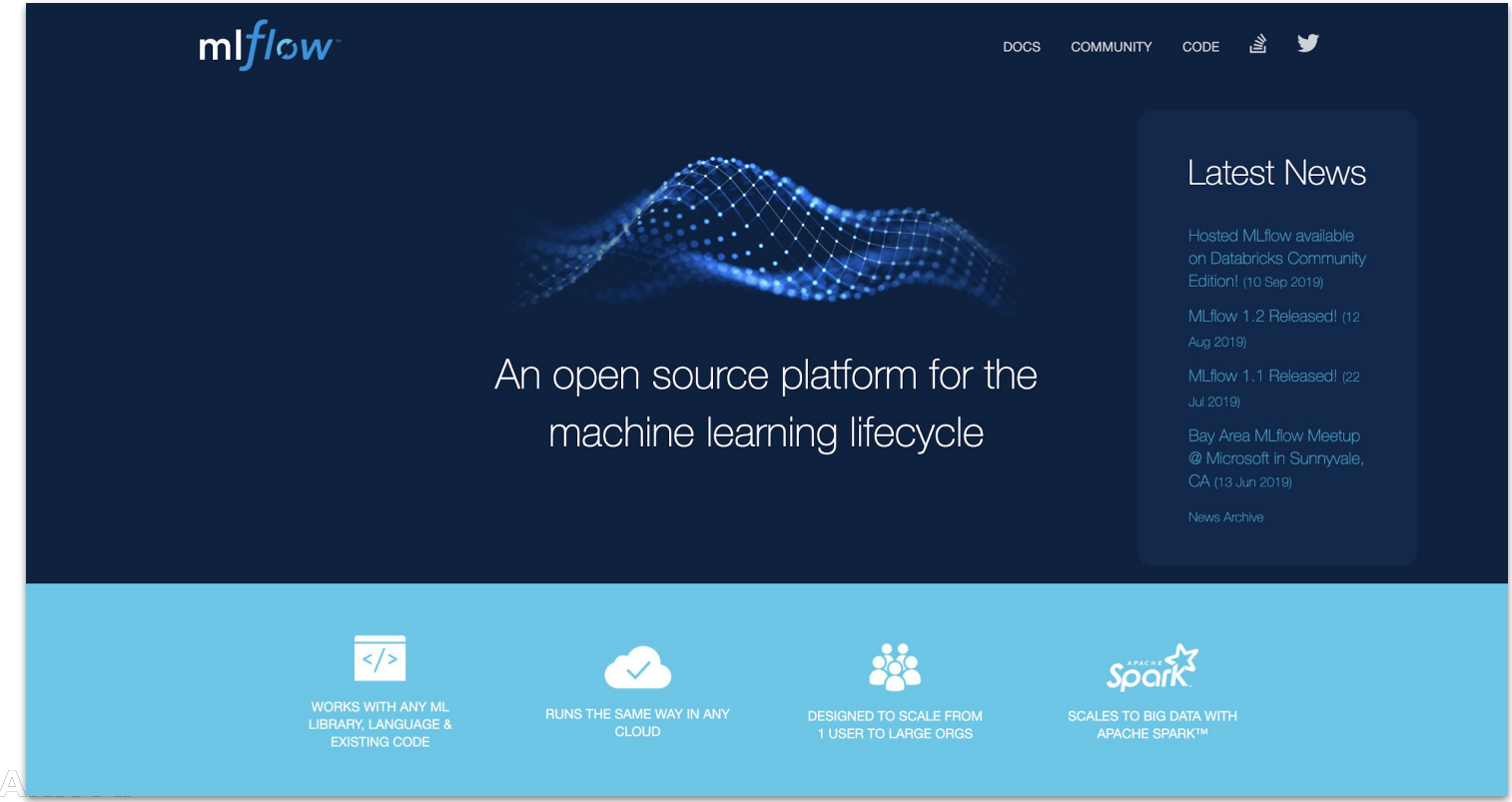



mlflow

MLflow - Software 2.0



MLflow - Intro



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, forge, 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.

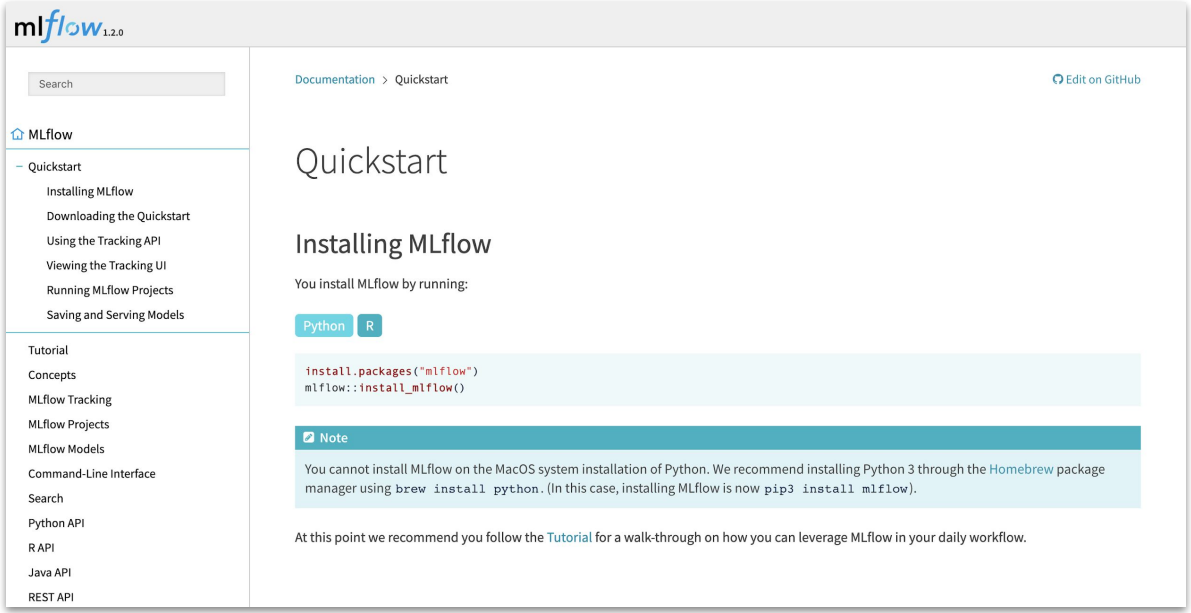
Index of /src/contrib/Archive/mlflow

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

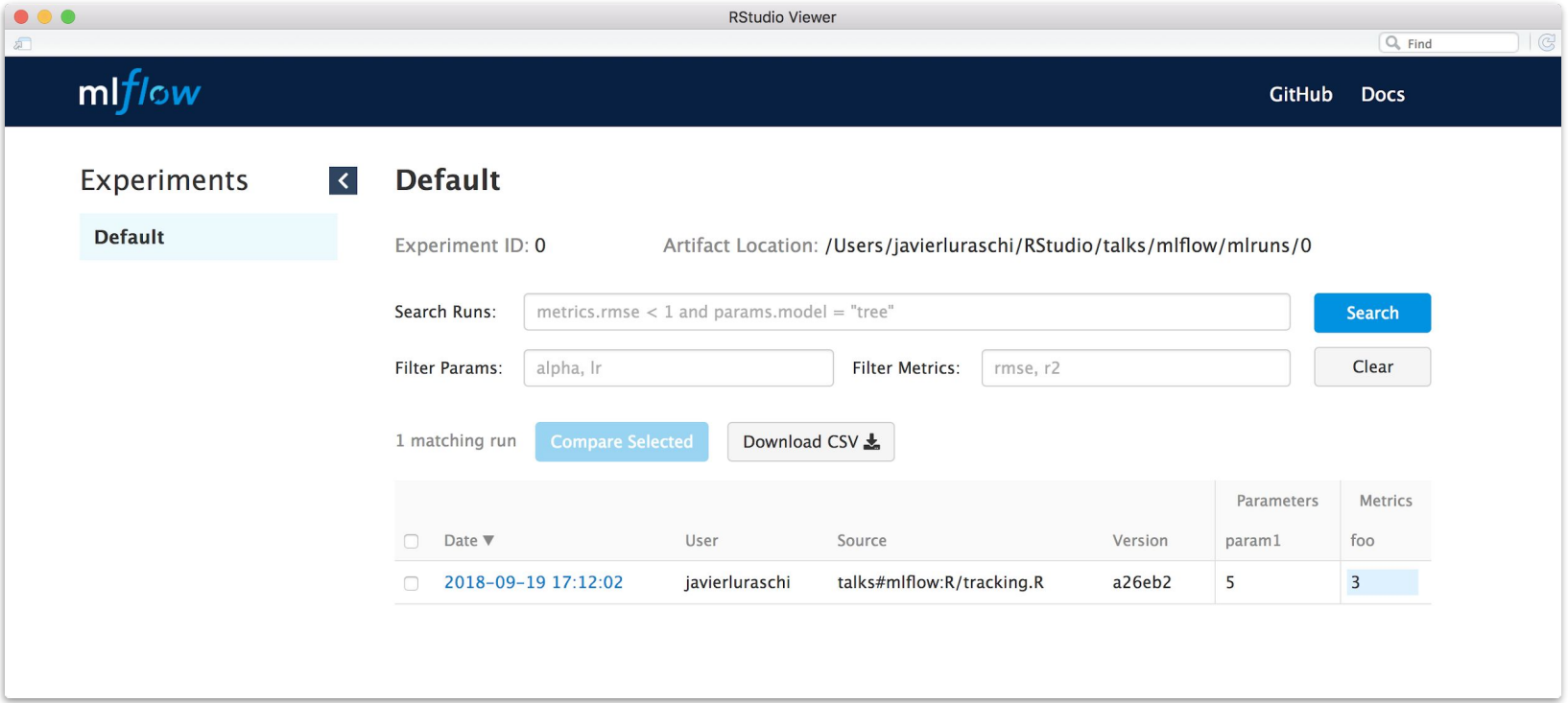
mlflow.org/docs/latest/index.html



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
Find
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
Parameters
Metrics
Date ▼
User
Source
Version
param1
foo
2018-09-19 17:12:02
javierluraschi
talks#mlflow:R/tracking.R
a26eb2
5
3

MLflow - What's next?

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

Thanks!