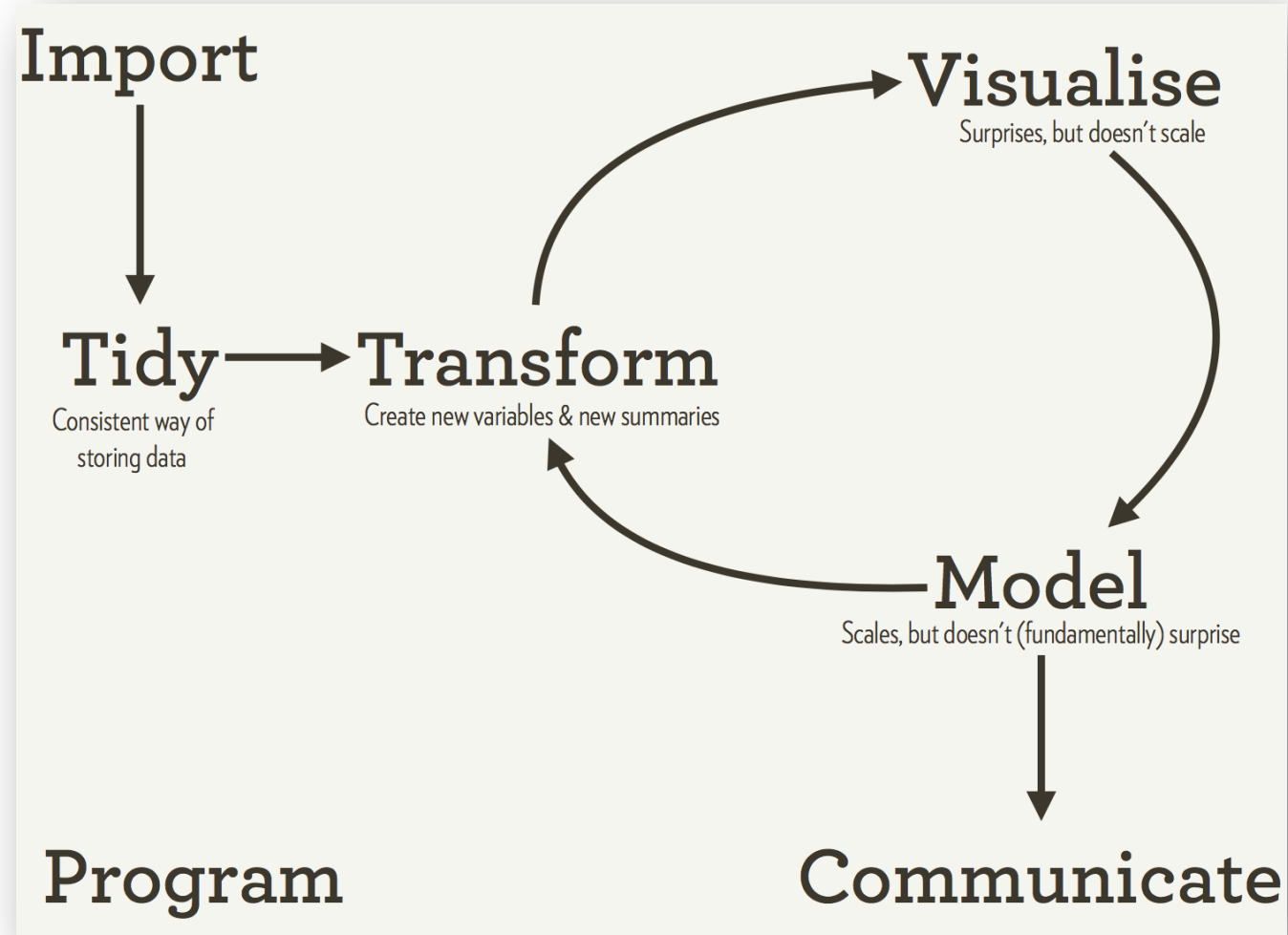




Modeling with R, TensorFlow, and Spark

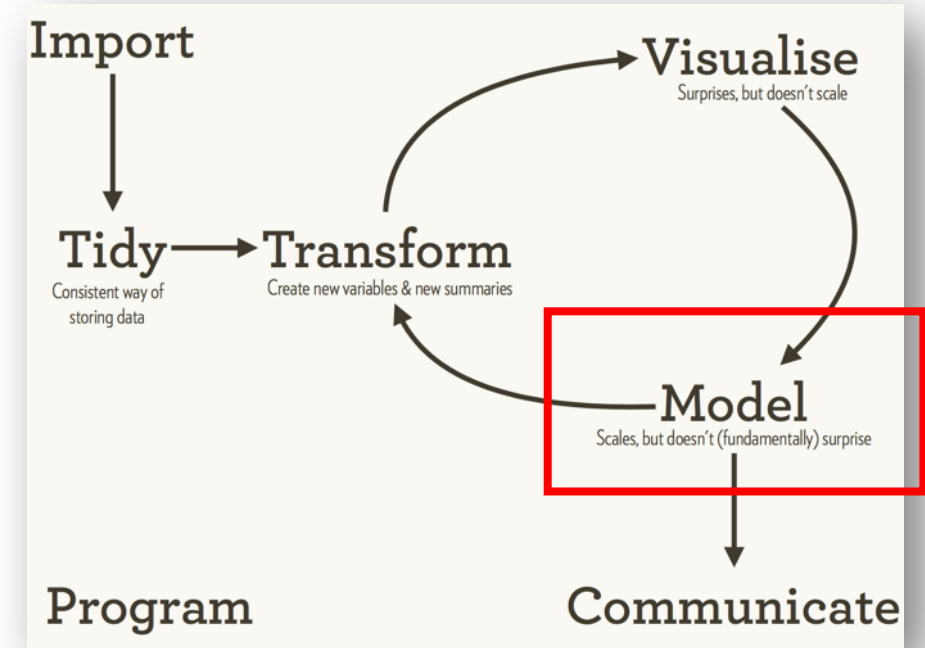
An Introduction to the Landscape for Machine Learning with R

How we think about data science



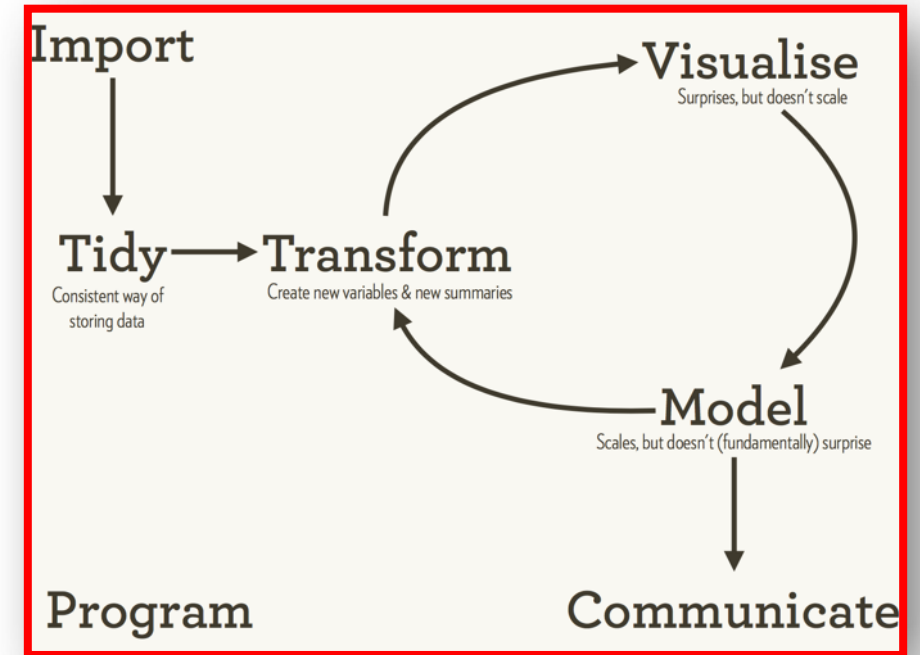
Models and methods

- Regression models
 - Logistic regression (stats)
 - Lasso and Elastic Net Regularized GLM (glmnet)
 - Gradient Boosting Machine (gbm)
- Convolutional neural networks
 - Tensorflow (tensorflow)
- Helpers
 - Keras (keras)
 - Classification and regression training (caret)



Workflow - Census Data

- Data wrangling
 - Tidy + Transformation
- Visualization
 - Exploratory data analysis
- Modeling
 - Logistic regression (stats)
 - Lasso and Elastic Net Regularized GLM (glmnet)
 - Gradient Boosting Machine (gbm & caret)
- Communicate
 - Flex Dashboard
 - Shiny



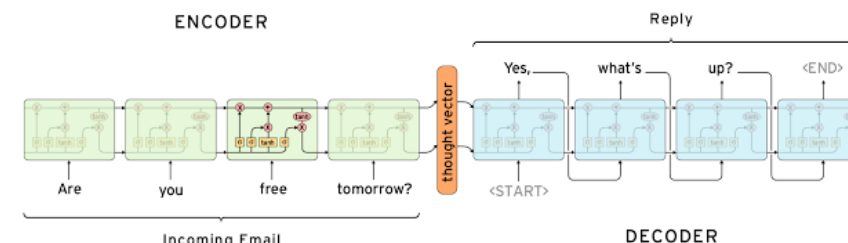
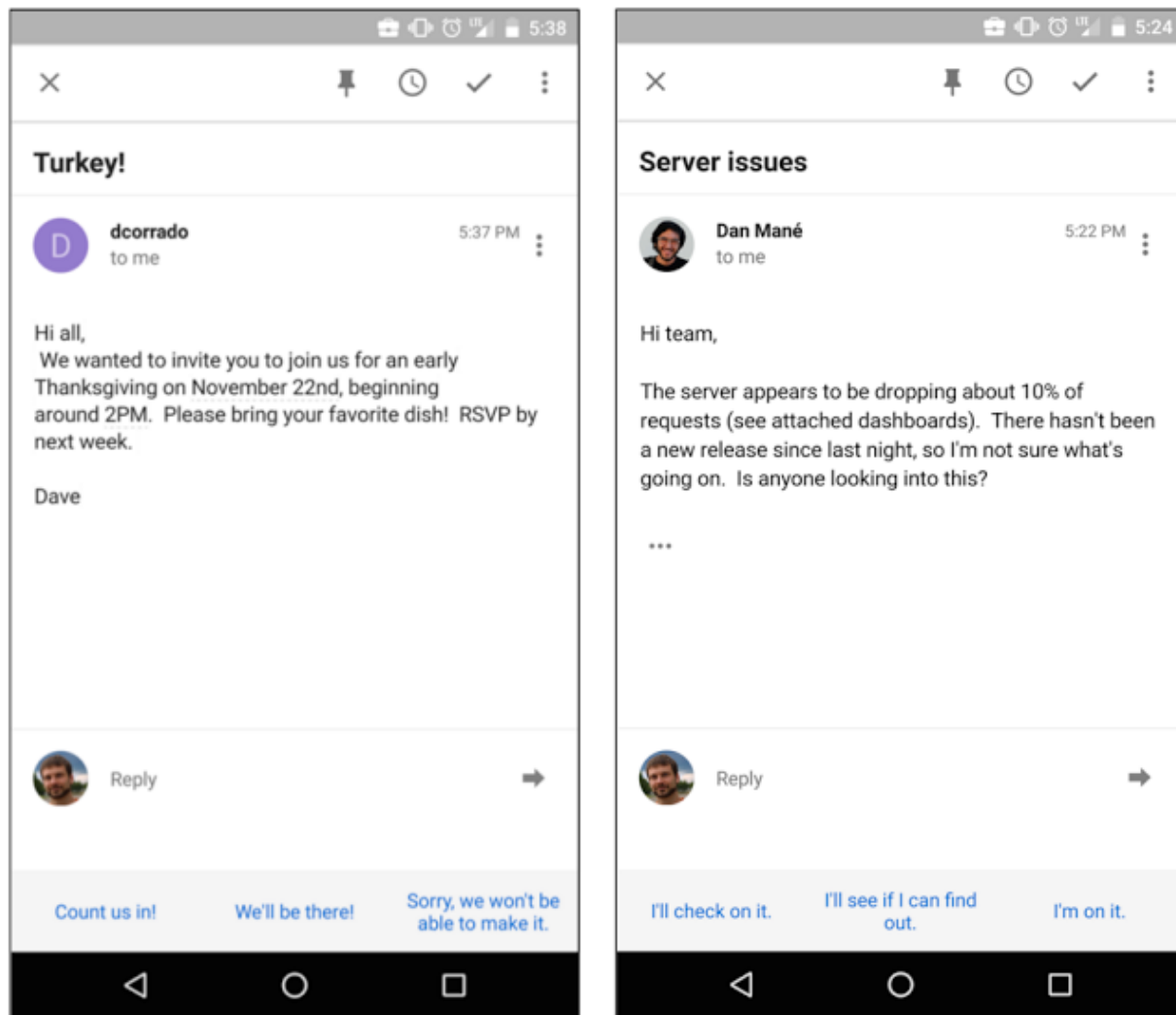
Census Data

- Prediction task is to determine whether a person makes over 50K a year in the year 1994.
- <https://archive.ics.uci.edu/ml/datasets/Census+Income>

	label	gender	native_country	education	education_num	occupation	workclass	marital_status	race	age_buckets
1	0	Male	United-States	Bachelors	13	Adm-clerical	State-gov	Never-married	White	(35,40]
2	0	Male	United-States	Bachelors	13	Exec-managerial	Self-emp-not-inc	Married-civ-spouse	White	(45,50]
3	0	Male	United-States	HS-grad	9	Handlers-cleaners	Private	Divorced	White	(35,40]
4	0	Male	United-States	11th	7	Handlers-cleaners	Private	Married-civ-spouse	Black	(50,55]
5	0	Female	Cuba	Bachelors	13	Prof-specialty	Private	Married-civ-spouse	Black	(25,30]
6	0	Female	United-States	Masters	14	Exec-managerial	Private	Married-civ-spouse	White	(35,40]
7	0	Female	Jamaica	9th	5	Other-service	Private	Married-spouse-absent	Black	(45,50]
8	1	Male	United-States	HS-grad	9	Exec-managerial	Self-emp-not-inc	Married-civ-spouse	White	(50,55]
9	1	Female	United-States	Masters	14	Prof-specialty	Private	Never-married	White	(30,35]
10	1	Male	United-States	Bachelors	13	Exec-managerial	Private	Married-civ-spouse	White	(40,45]
11	1	Male	United-States	Some-college	10	Exec-managerial	Private	Married-civ-spouse	Black	(35,40]
12	1	Male	India	Bachelors	13	Prof-specialty	State-gov	Married-civ-spouse	Asian-Pac-Islander	(25,30]
13	0	Female	United-States	Bachelors	13	Adm-clerical	Private	Never-married	White	(18,25]

Census Demo

TensorFlow



Smart Reply
Are you free tomorrow?
Yes, what's up?

<https://www.tensorflow.org/about/uses>

TensorFlow & RStudio

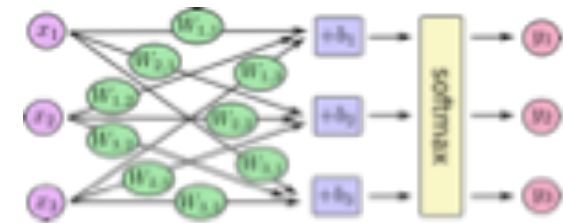
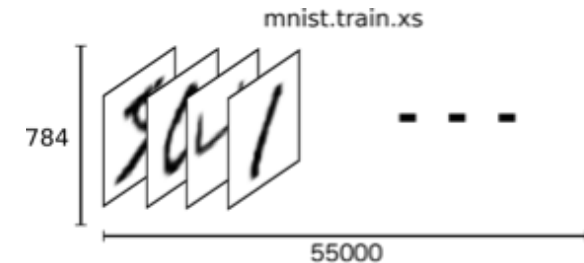
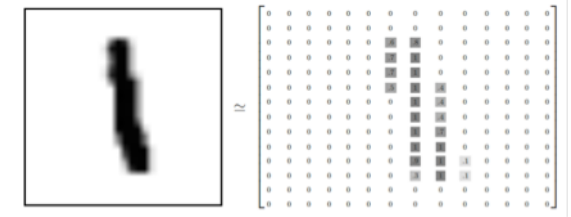


- tensorflow R package
- <https://tensorflow.rstudio.com/>
- Connects to TensorFlow API
- Runs in standalone or cluster mode
- Portable across servers and devices
- Works with Keras
- Easy to get started!



TensorFlow – MNIST Example

- Define your model
 - $y = \text{softmax}(Wx + b)$
- Set up the loss function
 - $\text{cross entropy} = -\sum(y * \log(\hat{y}))$
- Optimize using stochastic gradient descent
 - Enhanced by backpropagation



$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax} \left(\begin{bmatrix} W_{1,1} & W_{1,2} & W_{1,3} \\ W_{2,1} & W_{2,2} & W_{2,3} \\ W_{3,1} & W_{3,2} & W_{3,3} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$

TensorFlow Demo

Keras

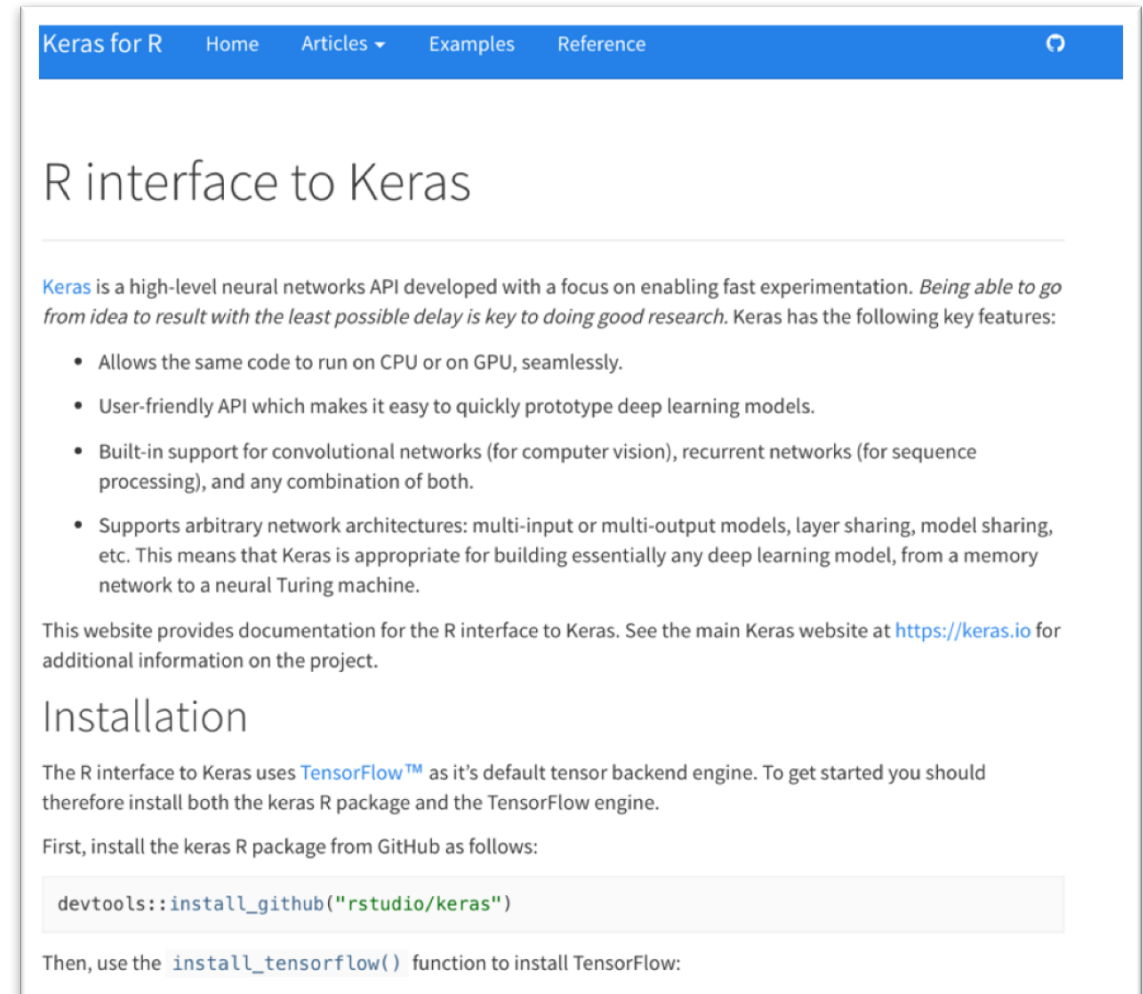


- Allows the **same code** to run on CPU or on GPU, seamlessly.
- **User-friendly** API which makes it easy to quickly prototype deep learning models.
- **Built-in support** for convolutional networks (for computer vision), recurrent networks (for sequence processing), and any combination of both.
- Supports arbitrary network architectures: multi-input or multi-output models, layer sharing, model sharing, etc. This means that Keras is appropriate for building essentially **any deep learning model**, from a memory network to a neural Turing machine.

Keras & RStudio



- keras R package
- <https://rstudio.github.io/keras>
- Uses friendly magrittr syntax
- Runs in standalone or cluster mode
- Uses tensorflow on the backend
- Portable across servers and devices
- Even easier to get started!

A screenshot of the 'Keras for R' website. The page has a blue header with navigation links: 'Keras for R', 'Home', 'Articles', 'Examples', and 'Reference'. The main heading is 'R interface to Keras'. Below this, a paragraph describes Keras as a high-level neural networks API. A bulleted list of features follows: allowing code to run on CPU or GPU, a user-friendly API for prototyping, built-in support for convolutional and recurrent networks, and support for arbitrary network architectures. A paragraph mentions that the website provides documentation for the R interface and points to the main Keras website. The 'Installation' section states that the R interface uses TensorFlow as its default backend and provides instructions to install the keras R package from GitHub using the 'devtools::install_github()' function. A code block shows the command: `devtools::install_github("rstudio/keras")`. The final instruction is to use the `install_tensorflow()` function to install TensorFlow.

Keras – Reuters Example

- Sequential model
- Layers
 - Densely connected NN
 - ReLU activation
 - Dropout regularization
 - Densely connected NN
 - Softmax activation
- Optimization
 - Categorical cross entropy
 - Adam
 - Accuracy
- Fit Model
- Assess Performance

Reuters Text Categorization data set (Reuters-21578) document

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="12981" NEWID="798">

<DATE> 2-MAR-1987 16:51:43.42</DATE>

<TOPICS><D>livestock</D><D>hog</D></TOPICS>

<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>

<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress kicks off tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC.

Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control and eradication program, the NPPC said.

A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter

⁵⁴
</BODY></TEXT></REUTERS>

Keras Demo



If you are investing in Spark, then there is nothing stopping you from using it with the full power of R.

Apache Spark

Fast and general engine for large-scale data processing

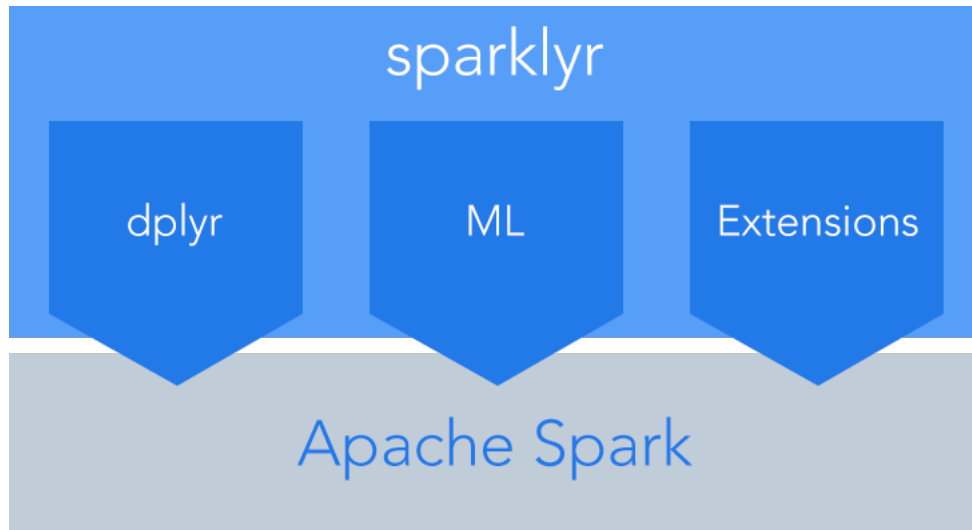
- *Can integrate with the Hadoop ecosystem*
- *Supports Spark SQL (HiveQL)*
- *Built-in machine learning*
- *Designed for performance*
- *Extensible*



The sparklyr package lets use Spark with R

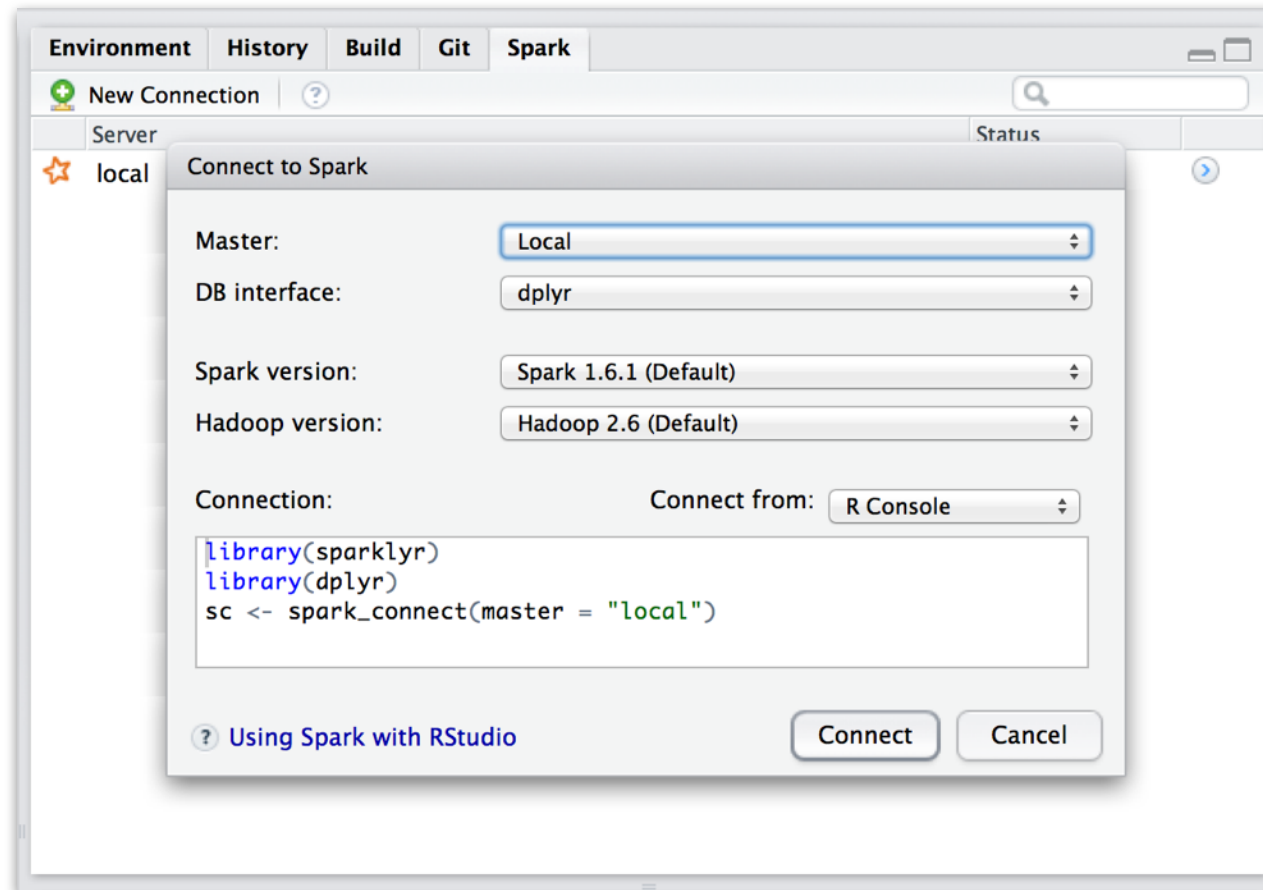


sparklyr has full support for



- dplyr syntax
- Spark ML
- Third party extensions
- *And the RStudio IDE*

Integrated with the RStudio IDE



dplyr backend

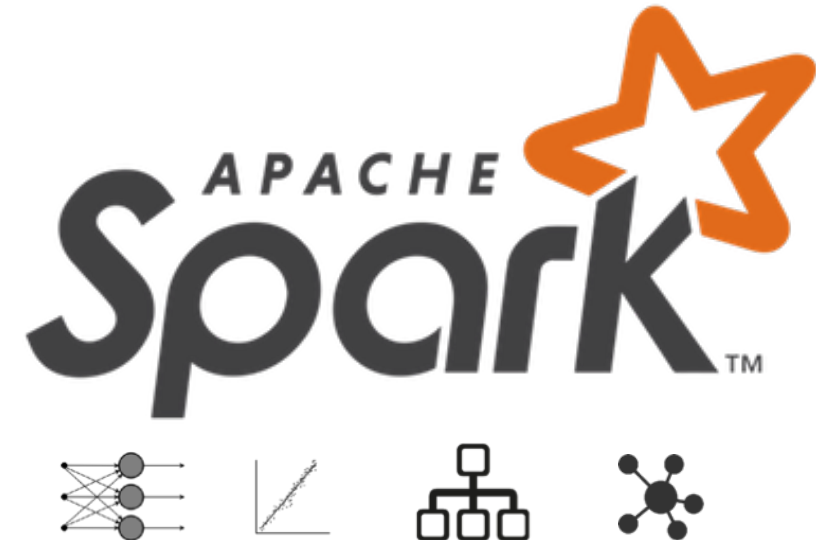


```
flights %>%  
  select(arr_delay) %>%  
  filter(distance > 100)
```

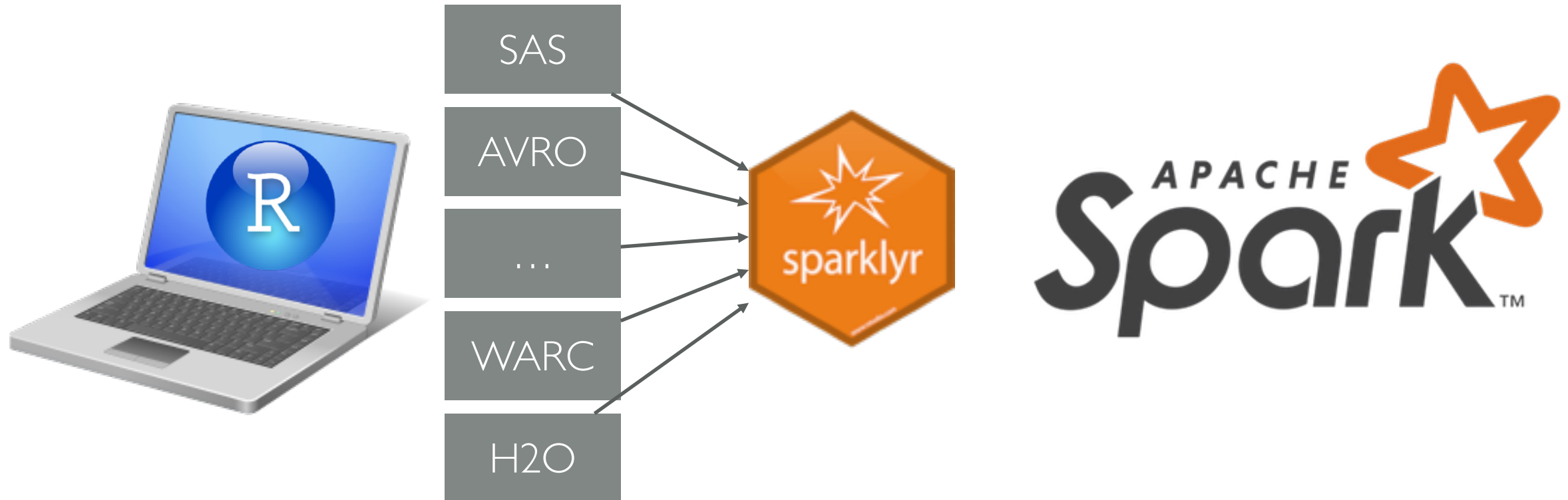


```
select arr_delay  
from flights  
where distance > 100
```

Analyze at scale with Spark ML



Create your own extensions



Run in local mode



Data Science in Spark with sparklyr
Chest Sheet

Data Science Toolchain with Spark + sparklyr

Import: Export as R Dataframe, Read a file, Read existing Hive table
Tidy: dplyr verbs, Direct Spark SQL (DB), SDF function (Scala API)
Transform: Transform function
Visualize: Collect data into R for plotting
Model: Spark MLlib, H2O Extension
Communicate: Collect data into R, Share plots, documents, and apps

Intro
sparklyr is an R interface for Apache Spark™. It provides a complete dplyr backend and the option to query directly using Spark SQL statements. With sparklyr, you can orchestrate distributed machine learning using either Spark's MLlib or H2O Sparkling Water.
Starting with version 1.0.0, RStudio Desktop, Server and Pro include integrated support for the sparklyr package. You can create and manage connections to Spark clusters and local Spark instances from inside the IDE.

Getting started

Local Mode
Easy setup, no cluster required
1. Install a local version of Spark: spark_install("2.0.1")
2. Open a connection: sc = spark_connect(master = "local")

On a YARN Managed Cluster
1. Install RStudio Server or RStudio Pro on one of the existing nodes, preferably an edge node.
2. Locate path to the cluster's Spark Home Directory, it normally is "/usr/lib/spark"
3. Open a connection: spark_connect(master="yarn-client", version = "2.0.1", spark_home = [Cluster's Spark path])

On a Mesos Managed Cluster
1. Install RStudio Server or Pro on one of the existing nodes.
2. Locate path to the cluster's Spark directory
3. Open a connection: spark_connect(master="mesos", version = "2.0.1", spark_home = [Cluster's Spark path])

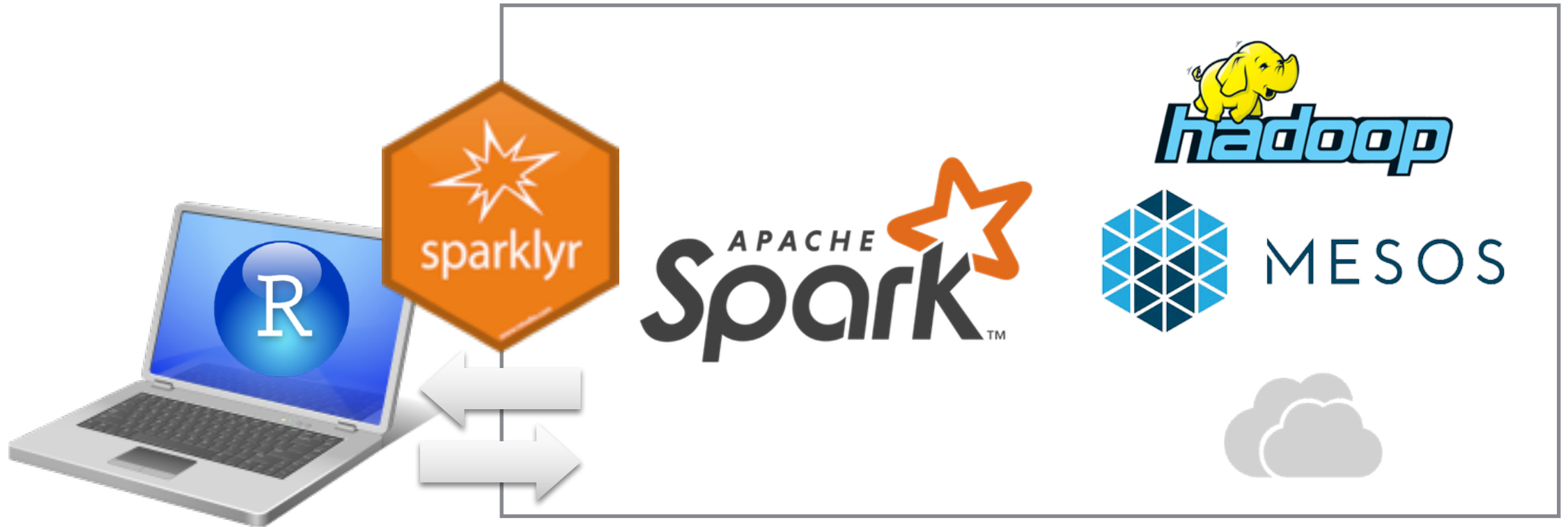
Using Livy (Experimental)
1. The Livy REST application should be running on the cluster.
2. Connect to the cluster: sc = spark_connect(master = "http://host:port", method = "livy")

Cluster Deployment
Managed Cluster: Driver Node, Cluster Manager, Worker Nodes
Stand Alone Cluster: Driver Node, Worker Nodes

Tuning Spark
Example Configuration: config = spark_config(), config\$spark.executor.cores = 2, config\$spark.executor.memory = "4G", sc = spark_connect(master = "yarn-client", config = config, version = "2.0.1")
Important Tuning Parameters with defaults: spark.prem.jvmopts, spark.prem.memory.kb
Important Tuning Parameters with defaults continued: spark.network.timeout, spark.executor.memory, spark.executor.cores, spark.executor.extraJavaOptions, spark.executor.instances, sparklyr.shell.executor-memory, sparklyr.shell.driver-memory

Using sparklyr
A brief example of a data analysis using Apache Spark, R and sparklyr in local mode
library(sparklyr), library(dplyr), library(ggplot2), library(json), library(readr), library(scales), library(tibble), library(waldo)
sc = spark_connect(master = "local")
spark_install("2.0.1")
importiris = emp_load(sc, iris, "spark_iris", overwrite = TRUE)
partition_iris = sdf_partition(importiris, training=0.5, testing=0.5)
sdf_register(partition_iris, c("spark_iris_training", "spark_iris_test"))
tidy_iris = tbl(sc, spark_iris_training) %>% select(Species, Petal.Length, Petal.Width)
model_iris = tidymodels::train_model(decision_tree, response=Species, features=c(Petal.Length, Petal.Width))
test_iris = tbl(sc, spark_iris_test)
pred_iris = sdf_predict(model_iris, test_iris) %>% collect()
inner_join(data.frame(prediction=pred_iris\$prediction, id=1:nrow(model_iris\$model_parameters\$labels)) %>% apply(test_iris\$Petal.Length, test_iris\$Petal.Width, FUN=guess, as.is=TRUE))
spark_disconnect(sc)

Connect to your cluster



For more information see: spark.rstudio.com

The screenshot shows a web browser window with the URL `spark.rstudio.com`. The page has a blue header with navigation links: `sparklyr`, `Home`, `dplyr`, `ML`, `Extensions`, `Deployment`, `Examples`, and `Reference`. The main content area is titled "sparklyr: R interface for Apache Spark". It features a list of bullet points describing the package's capabilities, a diagram showing the relationship between sparklyr, dplyr, ML, Extensions, and Apache Spark, and sections for installation, links, license, developers, and dev status.

sparklyr: R interface for Apache Spark

- 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 Sparks distributed [machine learning](#) library from R.
- Create [extensions](#) that call the full Spark API and provide interfaces to Spark packages.

The diagram shows a blue box labeled "sparklyr" at the top. Below it are three blue boxes labeled "dplyr", "ML", and "Extensions". These three boxes are connected by arrows pointing down to a larger blue box labeled "Apache Spark".

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 = "1.6.2")
```

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

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

If you use the RStudio IDE, you should also download the latest [preview release](#) of the IDE which includes several enhancements for interacting with Spark (see the [RStudio IDE](#) section below for more details).

Connecting to Spark

You can connect to both local instances of Spark as well as remote Spark clusters. Here we'll connect to a

Links

Download from CRAN at <https://cran.r-project.org/package=sparklyr>
Report a bug at <https://github.com/rstudio/sparklyr/issues>

License

Apache License 2.0 | file [LICENSE](#)

Developers

Javier Luraschi
Author, maintainer
Kevin Ushey
Author
JJ Allaire
Author
The Apache Software Foundation
Author, copyright holder
[All authors...](#)

Dev status

build passing
CRAN 0.5.4

sparklyr Demo