Centrum Wiskunde & Informatica

Capturing the Laws of (Data) Nature

Hannes Mühleisen, Martin Kersten & Stefan Manegold

CIDR 2015

Statistical Model Fitting & DB?

Database

Stats

User gave me a model, let's see.

I am storing some data.

I need some of the observations to fit the model.

This other guy is reading some of my data.

Cool, the model seems to fit the data well!

Let's get some more data to validate the fit…

This other guy is reading some more of my data.

Amazing, model fit is validated.

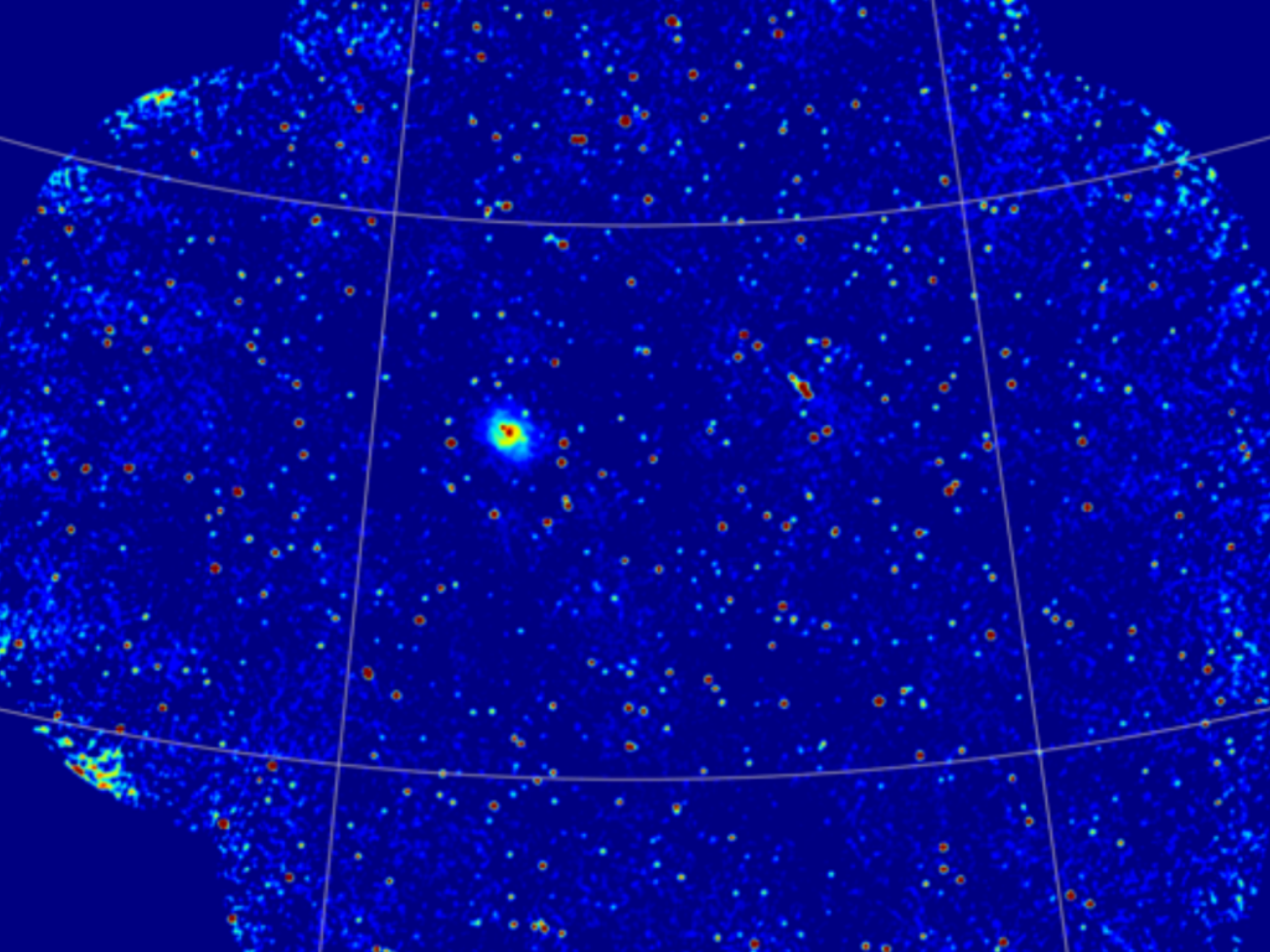
Beer!

I am storing some data.

The point?

- Everyone has models, **they encode our understanding of the world**
- Everyone has data to train/fit and validate a model
- So far, data management community has **ignored** these models
    - But they hold precious domain knowledge!

LOFAR Example



source	ngz	flux
1	0.15595	0.23159
1	0.12392	0.34782
1	0.14892	0.15927
1	0.14892	0.32261
1	0.15595	0.35390
1	0.14892	0.32784
1	0.12392	0.33068
1	0.18486	0.28135
1	0.18486	0.17872
1	0.15595	0.28813

Configuration

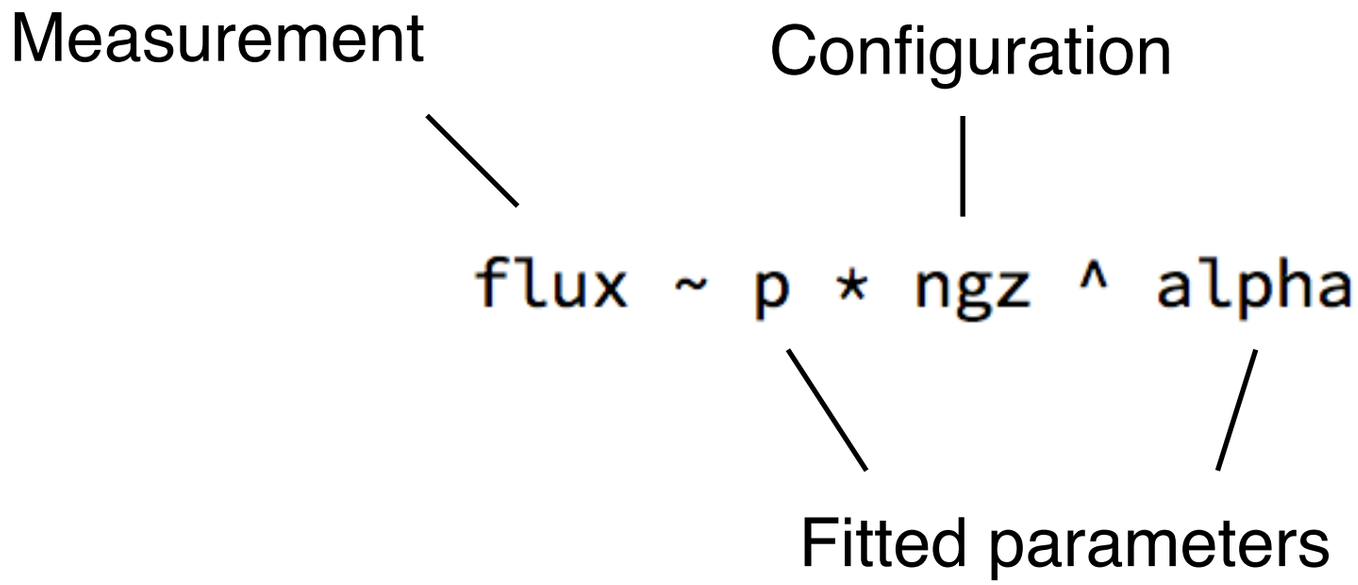
Measurement

Grouped by-source operation

Model!

Convergence Hints

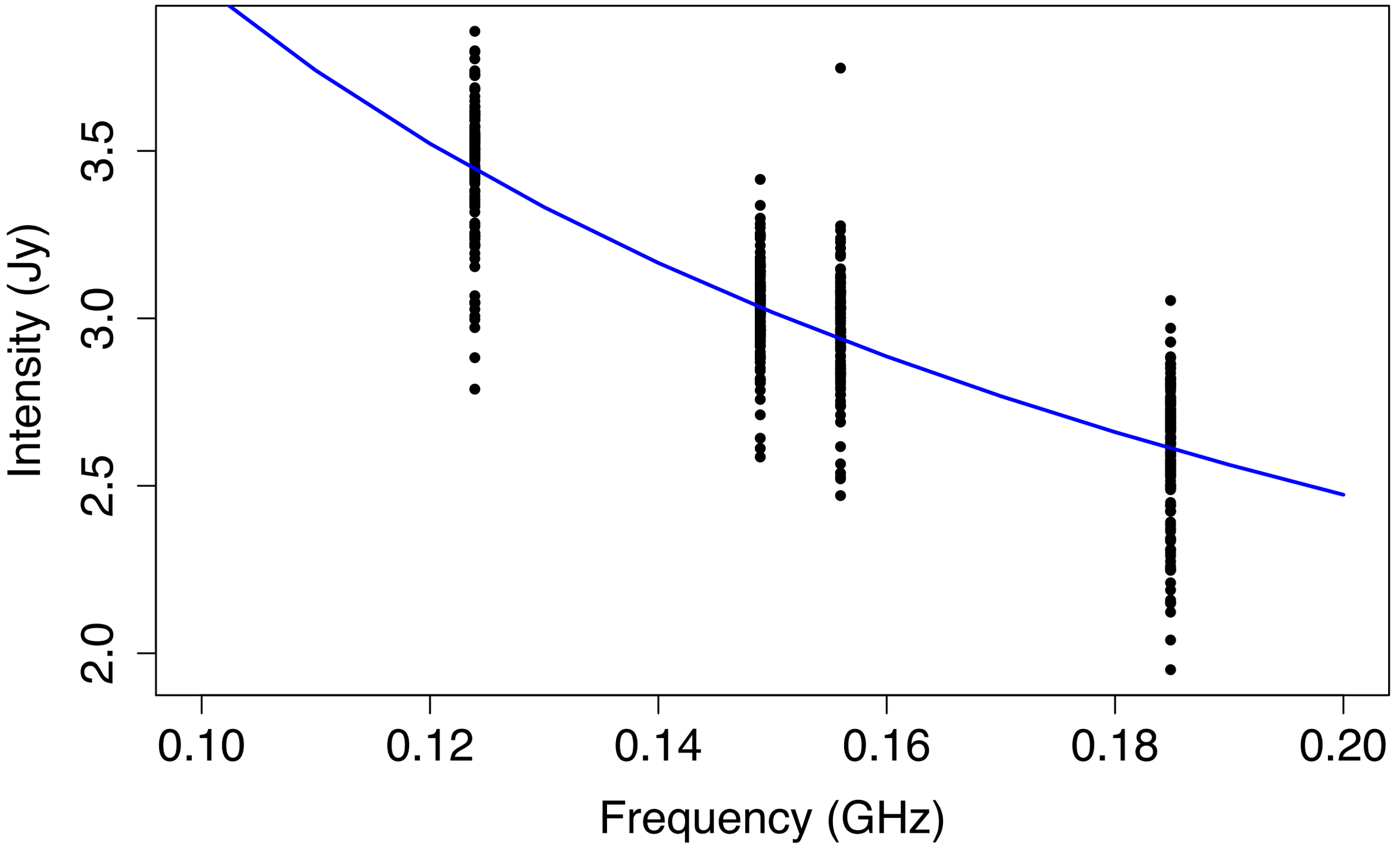
```
> alpha <- sapply(split(rsmdata, rsmdata$source),
+         function(df) {
+         tryCatch({
+                 fit <- nls(flux ~ p * ngz ^ alpha,
+                         data=df, start=list(alpha=-.7, p=0.8),
+                         control = list(maxiter = 500))
+                 coefs <- coef(fit)
+                 coefs[[1]]
+                 }, error=function(e) {return(as.numeric(NA))
+         })
+ })
```


Measurement
Configuration
flux ~ p * ngz ^ alpha
Fitted parameters

Source	Wavelength ν	Intensity I
1	0.1559555	0.2315911
1	0.1239243	0.3478159
1	0.1489243	0.1592717
[1,452,821 more rows]		

$\Rightarrow$

Source	Spectral Index α	Constant p	Residual SE
1	-0.7183309	0.06257838	0.006559710
2	-0.8932245	0.07195620	0.008007786
3	-0.7880774	0.56190180	0.016778232
[35,681 more rows]			



source=17562, alpha=-0.692, p=0.812

Exploit!

```
flux ~ p * ngz ^ alpha
```

Model to function conversion (automatic)

```
flux <- function(p, ngz, alpha) { p * ngz ^ alpha }
```

Move to DB (automatic)

```
CREATE FUNCTION flux3()
RETURNS TABLE (source integer, nu double, flux DOUBLE) LANGUAGE R {
    r <- data.frame(... # model params
    r$flux <-  r$p * r$nu ^ r$alpha
    return(r[c("source","nu","flux")])
};
```

```
sql>select flux from flux3() where source=17562 and nu=0.156;
+--------------------------+
| flux                     |
+==========================+
|       2.9370666853479666 |
+--------------------------+
```

Approximate Answer with zero IO*

But…

- What do we do if model parameters are not specified in the query?
  - Sample data?
- Given multiple parameters, it is far from certain that all combinations of values are allowed in the model.
  - Construct filter?

“Semantic” Compression

	Flux	Flux Residuals	Ratio
ORIG	11,665,408	11,665,408	0%
GZIP	4,331,782	3,748,872	86%
BZIP2	3,341,574	2,752,044	82%
XZ	2,887,584	2,727,144	94%

Drop residuals = lossy compression =

Data & Model Changes

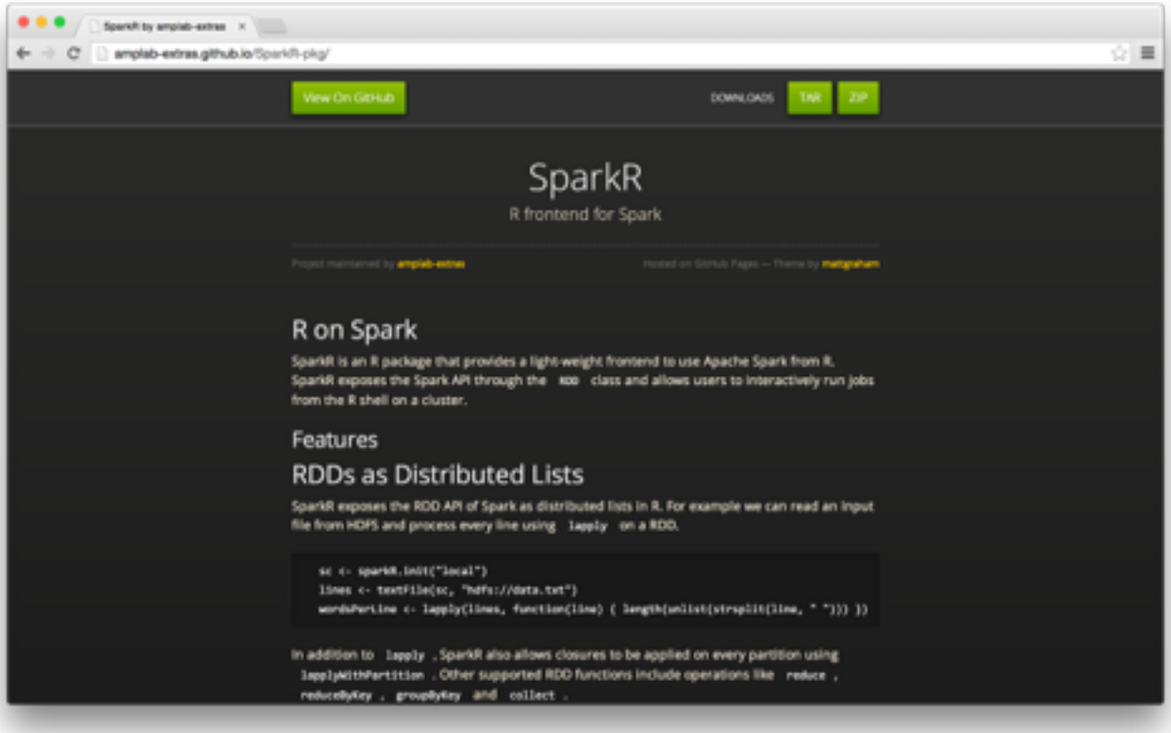
- What should we do if the user gives us a better model?
  - Recompressing could be very expensive
  - Threshold for improvement?
- Changes in the data affect the model quality, too
  - Switch models?
  - Constant Monitoring?

Multiple, partial or grouped

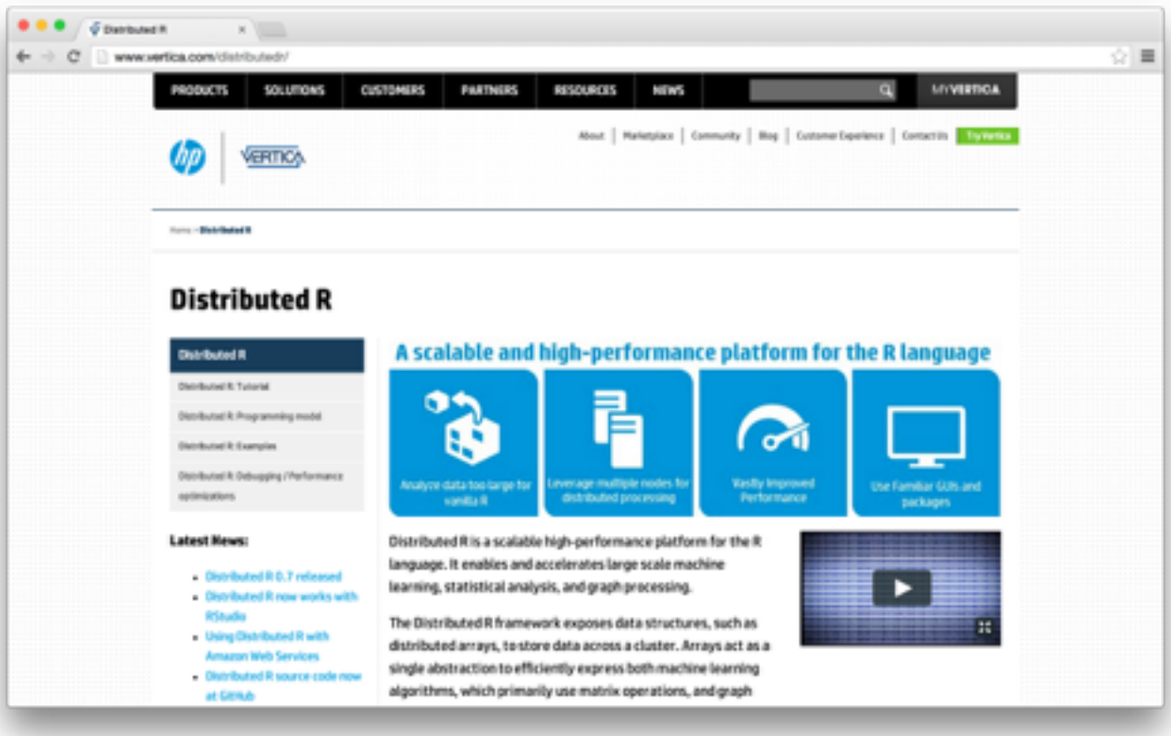
- There could be many models for a table with overlapping parameters
  - Which one to pick?
- Models do not have to cover the entire table/column
  - “Patching”?
- Models could be fitted on aggregation results
  - Keep group counts?

How do we get our hands on Models?


www.oracle.com/technetwork/database/database-technologies/r/r-enterprise/overview/index.html
ORACLE
Sign In/Register Help Country Communities I am a... I want to... Search
Products Solutions Downloads Store Support Training Partners About OTN
Oracle Technology Network > Database > Database Technology Index > R > R Enterprise > Overview
Overview Downloads Documentation Community Learn More
Oracle R Enterprise
Performance, Scalability, and Ease of Production Deployment
News Flash: Oracle R Enterprise 1.4.1 is now generally available!
New: Automated Trading Strategies using R
Oracle R Enterprise, a component of the Oracle Advanced Analytics Option, makes the open source R statistical programming language and environment ready for the enterprise and big data. Designed for problems involving large amounts of data, Oracle R Enterprise integrates R with Oracle Database. R users can run R commands and scripts for statistical and graphical analyses on data stored in the Oracle Database. R users can develop, refine, and deploy R scripts that leverage the parallelism and scalability of the database to automate data analysis. Data analysts and data scientists can run R packages and develop and operationalize R scripts for analytical applications in one step—without having to learn SQL. Oracle R Enterprise performs function pushdown for in-database execution of base R and popular R packages. Because it runs as an embedded component of Oracle Database, Oracle R Enterprise can run any R package either by function pushdown or via
COLLABORATE 15
REGISTRATION NOW OPEN
WEBCAST
The Future of the Database—Product Introduction
WATCH NOW
Get the Latest Oracle Database

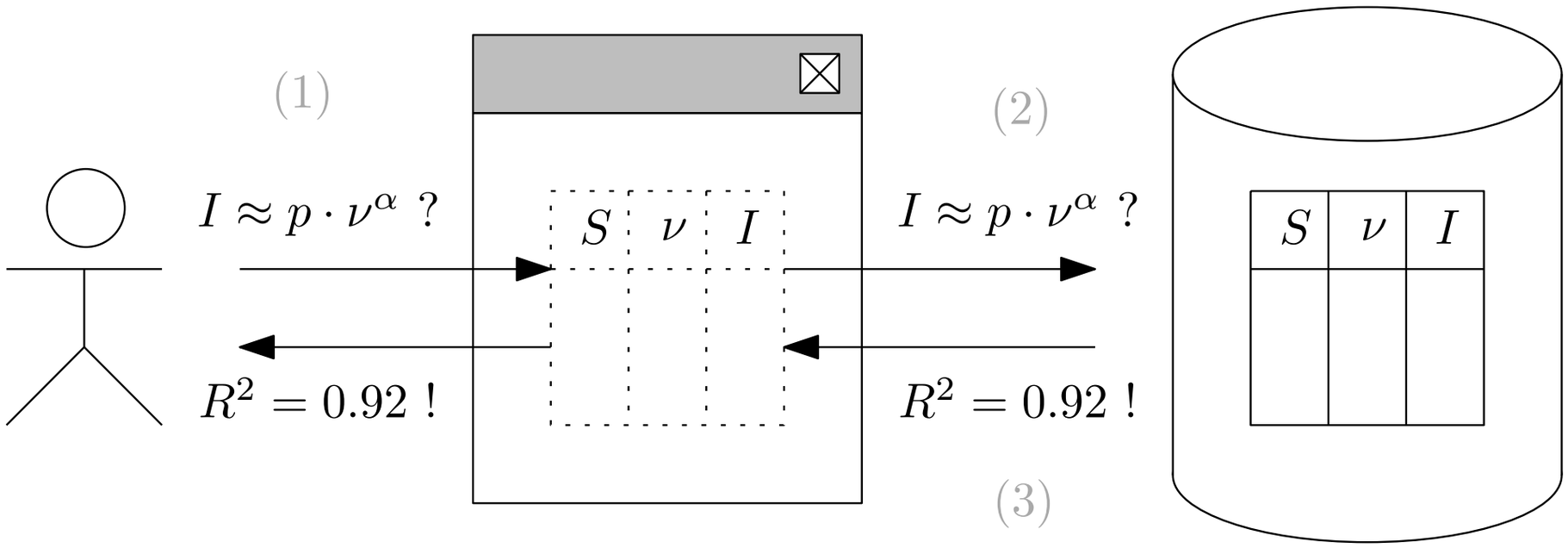

amplab-extras.github.io/SparkR-pkg/
View On GitHub
DOWNLOADS TAR ZIP
SparkR
R frontend for Spark
R on Spark
SparkR is an R package that provides a light-weight frontend to use Apache Spark from R. SparkR exposes the Spark API through the RDD class and allows users to interactively run jobs from the R shell on a cluster.
Features
RDDs as Distributed Lists
SparkR exposes the RDD API of Spark as distributed lists in R. For example we can read an input file from HDFS and process every line using lapply on a RDD.
sc <- sparkR.init("local")
lines <- textFile(sc, "hdfs://data.txt")
wordsPerLine <- lapply(lines, function(line) { length(unlist(strsplit(line, " "))) })
In addition to lapply , SparkR also allows closures to be applied on every partition using lapplyWithPartition . Other supported RDD functions include operations like reduce , reduceByKey , groupByKey and collect .


help.sap.com/hana/sap_hana_r_integration_guide_en.pdf
PUBLIC
SAP HANA Platform SPS 09
Document Version: 1.0 - 2014-11-26
SAP HANA R Integration Guide


www.vertica.com/distributedr/
PRODUCTS SOLUTIONS CUSTOMERS PARTNERS RESOURCES NEWS MYVERTICA
About Marketplace Community Blog Customer Experience Contact Us Try Vertica
Distributed R
Distributed R
A scalable and high-performance platform for the R language
Analyze data too large for vanilla R
Leverage multiple nodes for distributed processing
Verify Improved Performance
Use Familiar GUIs and packages
Latest News:
Distributed R 0.7 released
Distributed R now works with RStudio
Using Distributed R with Amazon Web Services
Distributed R source code now at GitHub
Distributed R is a scalable high-performance platform for the R language. It enables and accelerates large scale machine learning, statistical analysis, and graph processing.
The Distributed R framework exposes data structures, such as distributed arrays, to store data across a cluster. Arrays act as a single abstraction to efficiently express both machine learning algorithms, which primarily use matrix operations, and graph

Integrate & Intercept

- Integrate model fitting infrastructure into data management system.
    - Also: **Huge** performance benefits for analysts!
- Intercept model fitting and validation operations by the user and store the model for later use.
    - Storage format: Model code + Parameters


(1)
$I \approx p \cdot \nu^{\alpha}$?
S ν I
$R^2 = 0.92$!
(2)
$I \approx p \cdot \nu^{\alpha}$?
S ν I
$R^2 = 0.92$!
(3)
(4)
$S = 42, \nu = 0.14, I =?$
$I = 3.0 \pm 0.05$!
(5)
S p α
$I \approx p \cdot \nu^{\alpha}$

CWI
Centrum Wiskunde & Informatica

"Essentially, all models are wrong,
but some are useful."

George E. P. Box

Questions?

http://hannes.muehleisen.org
@hfmuehleisen