PyData
Bratislava

GAP
DATA
INSTITUTE

Scalable R

(R <- Slovakia Meetup #2, Nervosa)

19. 6. 2017

Michal Marušan, #RSlovakia #PyDataBA

Scalable R

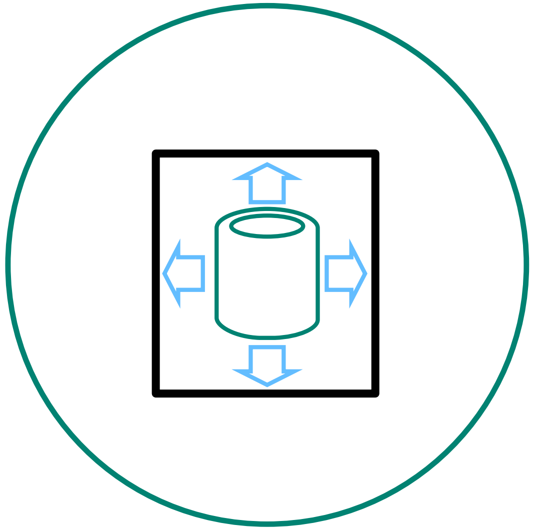
R Meetup

Michal Marusan @ 2017

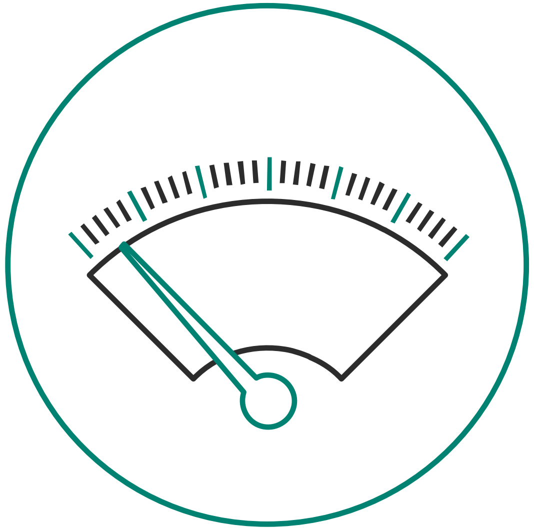
~~What is R?~~

• How many of you use/know R?	100%
• How many of you get data from database?	80%
• How many of you have to deal with R limitations?	60%
• How many of you know/use Hadoop/Spark?	50%

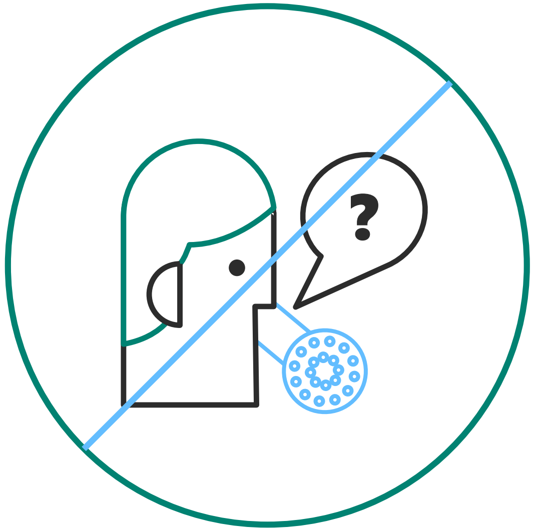
Challenges posed by open source R



Limited Data Scale in Memory



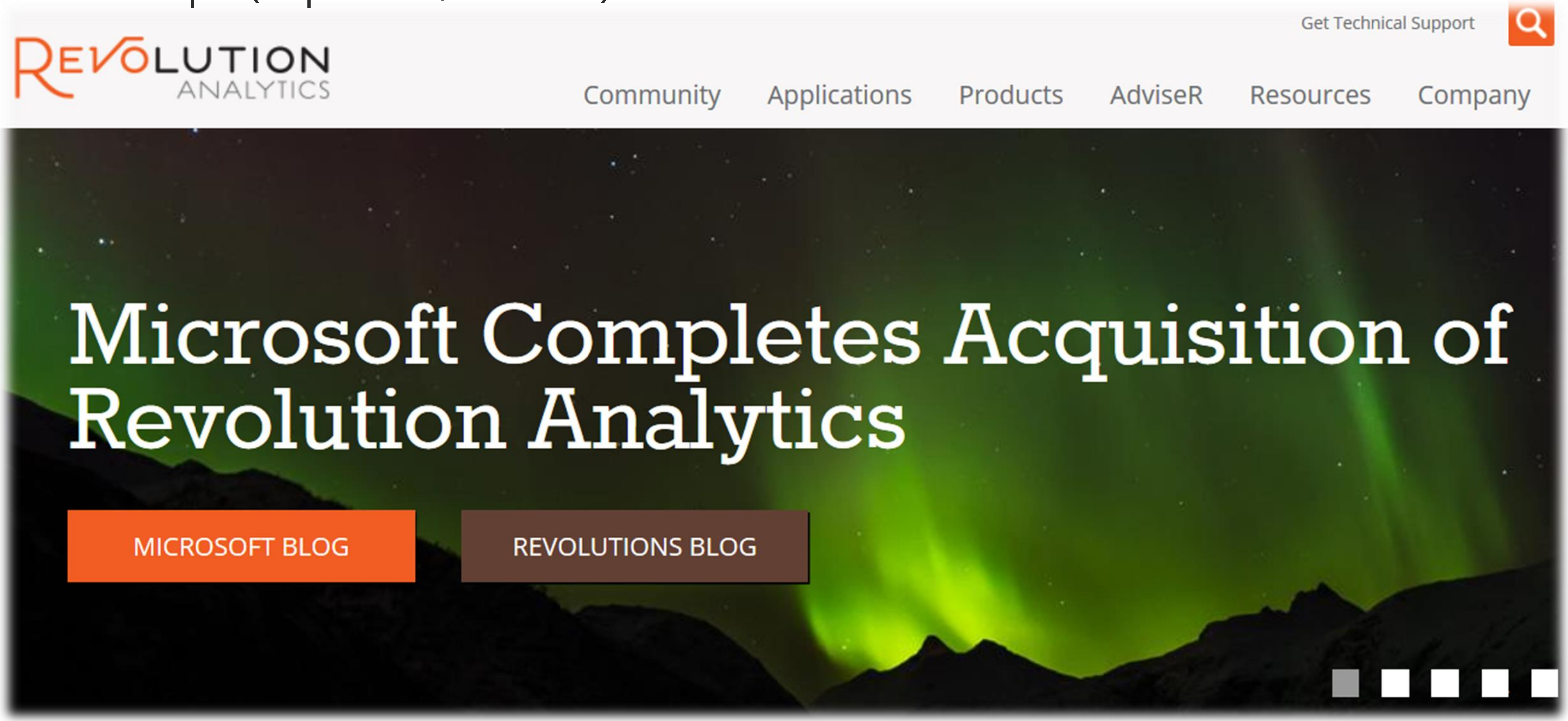
Single-threaded / Not parallelized



Lack of Commercial Support

Complex Deployment Processes Production

First step (April 6, 2015)



Get Technical Support

Community Applications Products AdviseR Resources Company

Microsoft Completes Acquisition of Revolution Analytics

Microsoft R products

Microsoft R Open / Client

- Free and open source R distribution
- Enhanced and distributed by Microsoft

SQL Server R Services

- Built in Advanced Analytics and Stand Alone Server Capability
- Leverages the Benefits of SQL 2016 Enterprise Edition

Microsoft R Server

- Microsoft R Server for Redhat Linux
- Microsoft R Server for SUSE Linux
- Microsoft R Server for Teradata DB
- Microsoft R Server for Hadoop on Redhat

how to tackle scale R?

multi-thread / parallel / memory

Microsoft R Server

MRS extends open-source R to allow:

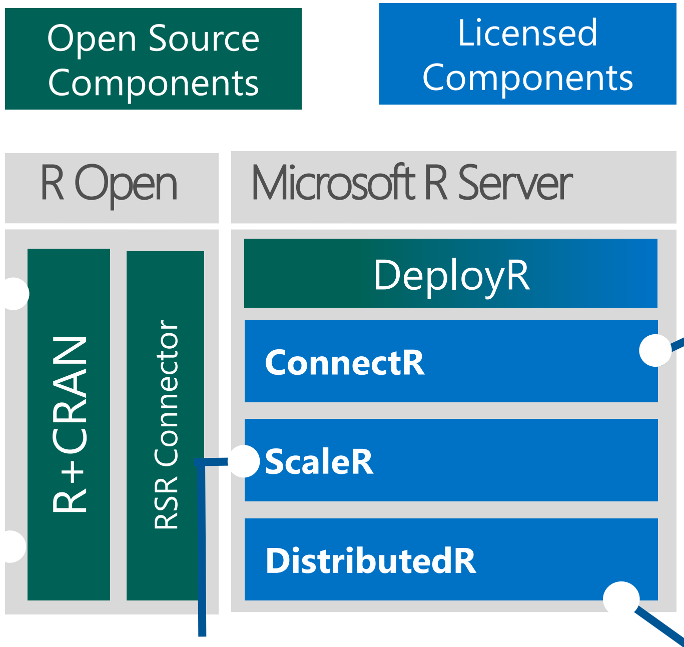
- Multi-threading
  - Matrix operations, linear algebra, and many other math operations run on all available cores
- Parallel processing
  - ScaleR functions utilize all available resources, local or distributed
- On-disk data storage
  - RAM limitations lifted

MRS Components

Platforms

R+CRAN

- Open source R interpreter
  - R 3.1.2
- Freely-available huge range of R algorithms
- Algorithms callable by RevoR
- Embeddable in R scripts
- 100% Compatible with existing R scripts, functions and packages



Open Source Components

Licensed Components

R Open

Microsoft R Server

R+CRAN

RSR Connector

DeployR

ConnectR

ScaleR

DistributedR

RevoR

- Performance enhanced R interpreter
- Based on open source R
- Adds high-performance math library to speed up linear algebra functions

ScaleR

- Ready-to-Use high-performance big data big analytics
- Fully-parallelized analytics
- Data prep & data distillation
- Descriptive statistics & statistical tests
- Range of predictive functions
- User tools for distributing customized R algorithms across nodes
- Wide data sets supported – thousands of variables

ConnectR

- High-speed & direct connectors

Available for:

- High-performance XDF
- SAS, SPSS, delimited & fixed format text data files
- Hadoop HDFS (text & XDF)
- Teradata Database & Aster
- EDWs and ADWs
- ODBC

DistributedR

- Distributed computing framework
- Delivers cross-platform portability

https://info.microsoft.com/rs/157-GQE-382/images/EN-WBNR-Slidedeck-Using-Microsoft-

CRAN, MRO, MRS Comparison

	R	Microsoft R Open	Microsoft R Server
Datasize	In-memory	In-memory	**In-Memory or Disk Based**
Speed of Analysis	Single threaded*	Multi-threaded*	**Multi-threaded, parallel processing 1:N servers**
Support	Community	Community	**Community + Commercial**
Analytic Breadth & Depth	7500+ innovative analytic packages	7500+ innovative analytic packages	**7500+ innovative packages + commercial parallel high-speed functions**
License	Open Source	Open Source	**Commercial license. Supported release with indemnity**

Open source R

```
mydata <-   read.csv("http://www.ats.ucla.edu/stat/data/binary.csv")

mylogit <-     glm(admit ~ gre + gpa + rank, data = mydata,
                   family = "binomial")
```

R Server

Switch functions

```
mydata <-   RxTextData(“/data/binary.csv”, fileSystem = hdfsFS)

mylogit <-     rxLogit(admit ~ gre + gpa + rank, data = mydata)
```

Demo

GLM

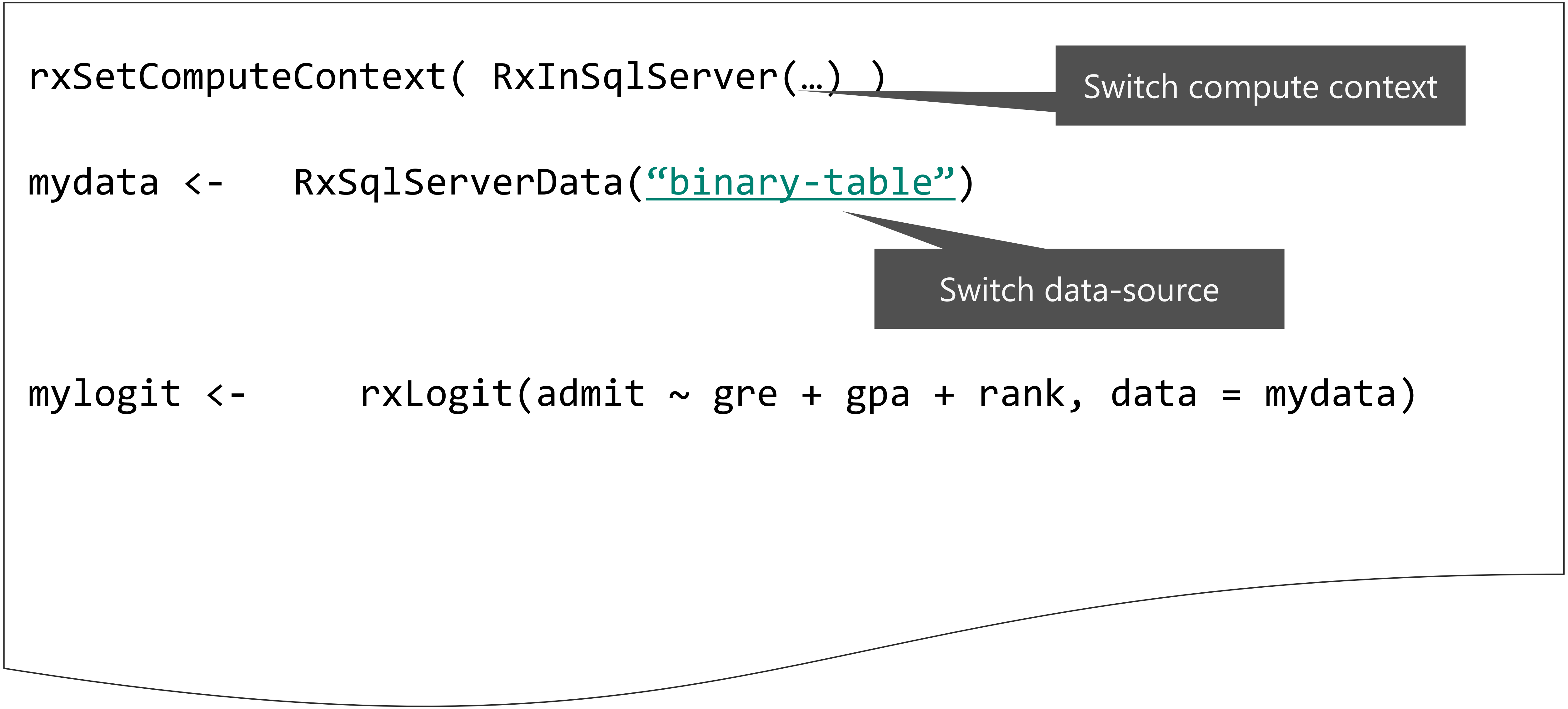
R Server parallelized by Spark

```
rxSetComputeContext( RxSpark(…) )

mydata <-   RxTextData(“/data/binary.csv”, fileSystem = hdfsFS)


mylogit <-     rxLogit(admit ~ gre + gpa + rank, data = mydata)
```

Switch compute context

Switch data-source

R Server parallelized by SQL Server

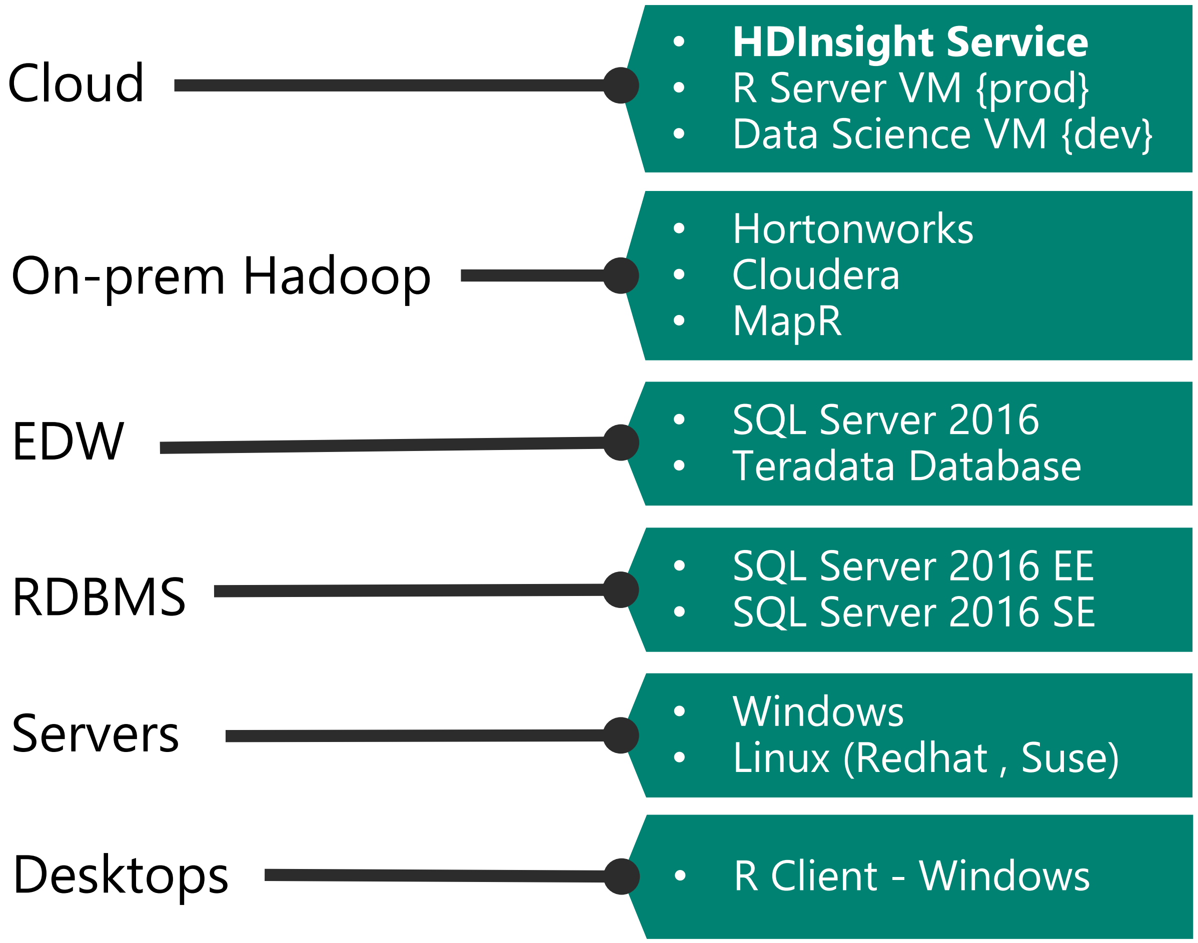


```
rxSetComputeContext( RxInSqlServer(…) )

mydata <-   RxSqlServerData(“binary-table”)


mylogit <-     rxLogit(admit ~ gre + gpa + rank, data = mydata)
```

Switch compute context

Switch data-source

R Server platform options



Cloud

- **HDInsight Service**
- R Server VM {prod}
- Data Science VM {dev}

On-prem Hadoop

- Hortonworks
- Cloudera
- MapR

EDW

- SQL Server 2016
- Teradata Database

RDBMS

- SQL Server 2016 EE
- SQL Server 2016 SE

Servers

- Windows
- Linux (Redhat , Suse)

Desktops

- R Client - Windows

Write once – deploy anywhere

Demo
WODA
rxComputeContext

High-Performance Compute & Analytics

HPC – “Many Models”

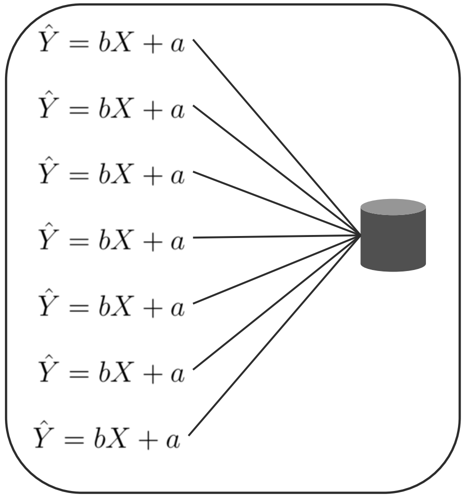
Characteristics

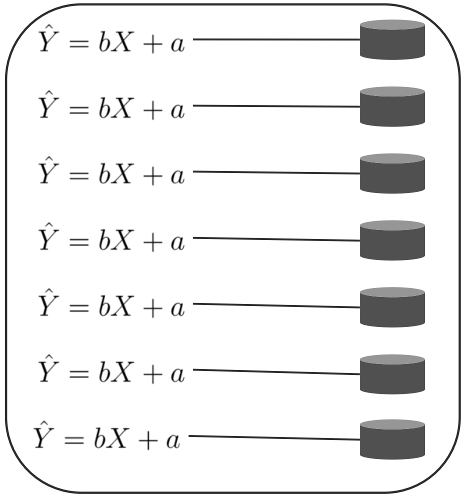
- Model validation
- Parameter sweep/optimisation
- Simulation
- Prediction at item/ group
- Embarrassingly parallel
- Small data-set per *“fn”*
- Parellise a *for* loop

Requirements

- Same *“fn”* many times, in parallel
- Any open-source *“fn”*
- Any home-grown *“fn”*

Common





HPA – “Big Model” and/or “Big Data”

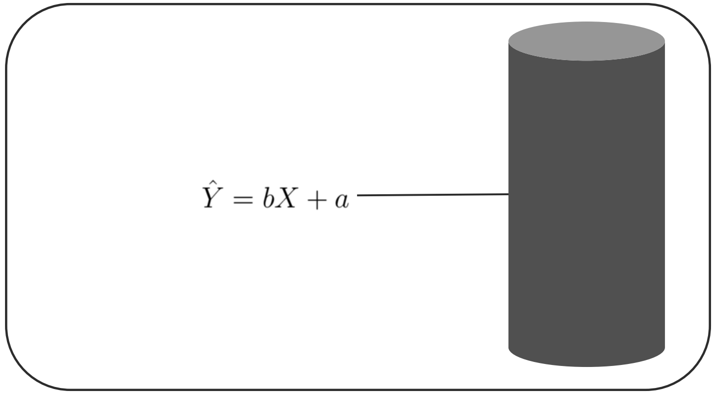
Characteristics

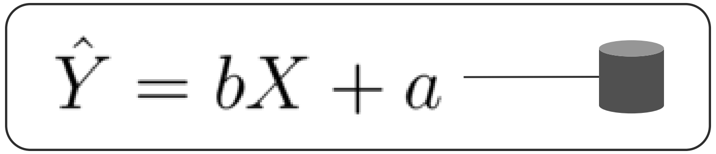
- Complex model formula/terms
- Curse of dimensionality
- High degree of freedom
- Wide Datasets
- Deep Datasets

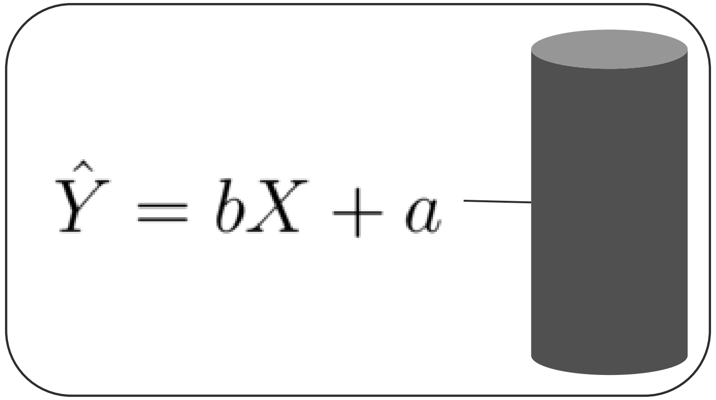
Requirements

- Split, parallelise & combine
- Specialised/custom *“fn”*

Not Common







Demo

rxExec

scale R for the enterprise use:

how do we operationalise R?

Some key operationalisation questions

How can we embed R-based analytics into business applications like CRM?

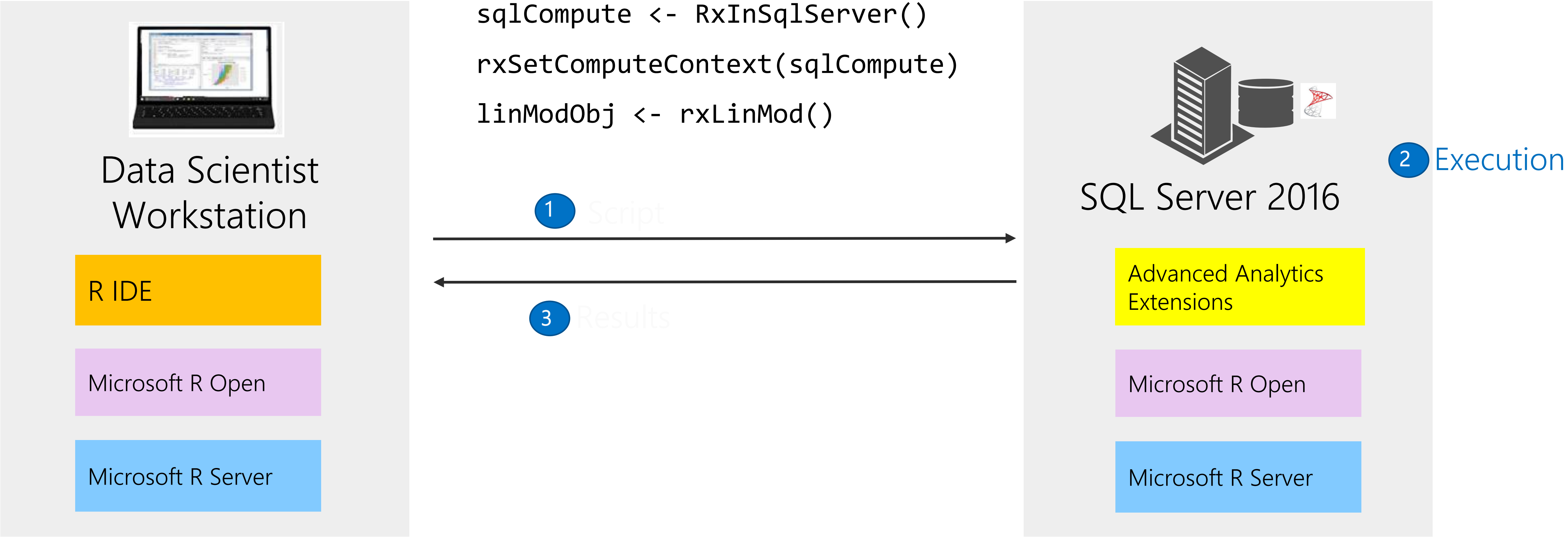
How can we build R-based analytics into BI dashboards and deploy quickly?

How can we maintain data security for R users?

How do we allow data scientists to share code and version control?

Model Development (R Users)

Working from R IDE on a local workstation, execute an R script that runs in-database on remote SQL server, and get the results back.



```
sqlCompute <- RxInSqlServer()
rxSetComputeContext(sqlCompute)
linModObj <- rxLinMod()
```

Data Scientist Workstation

R IDE

Microsoft R Open

Microsoft R Server

1 Script

3 Results

SQL Server 2016

2 Execution

Advanced Analytics Extensions

Microsoft R Open

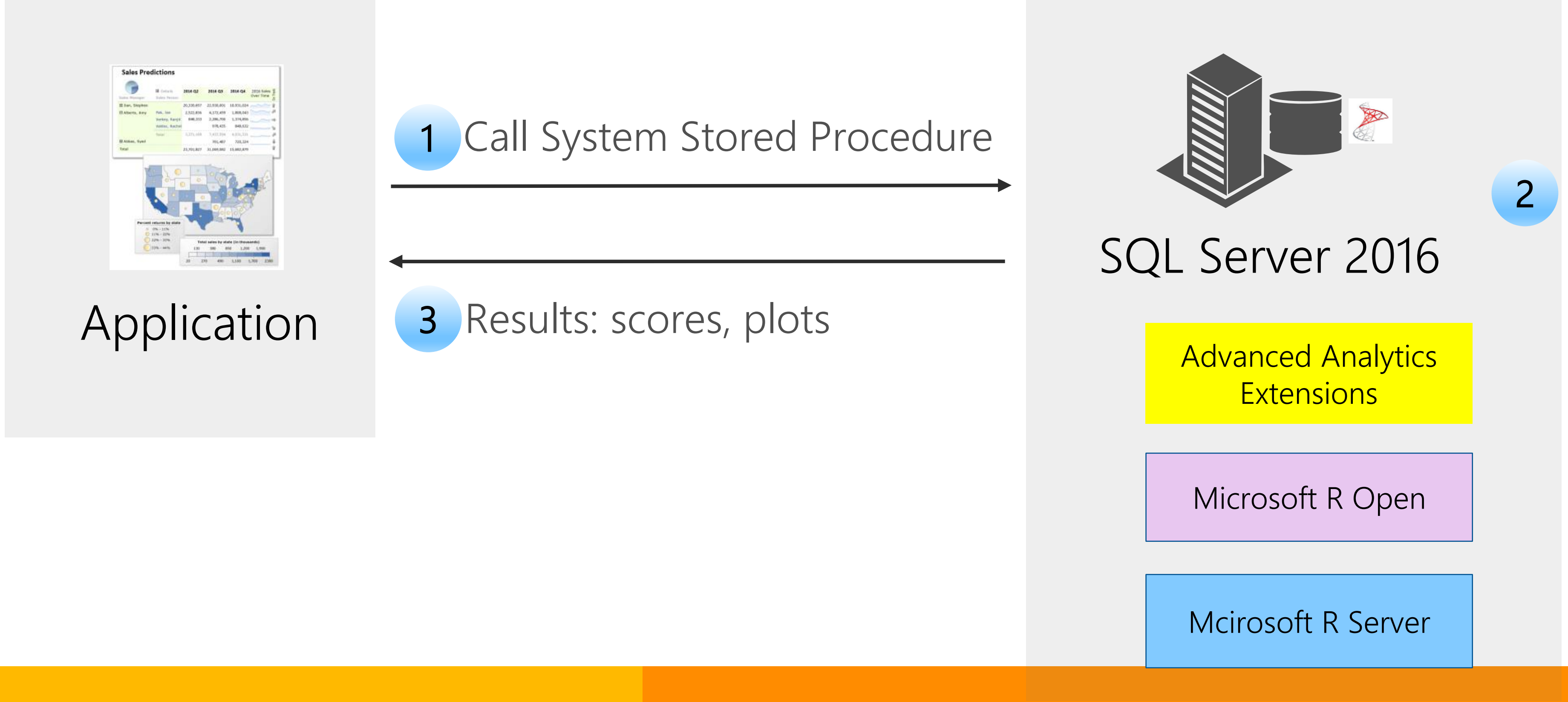
Microsoft R Server

Model Operationalization with SQL R Services

R Code->T-SQL Stored Proc

Call a T-SQL System Stored Procedure to generate features and train (or retrain) the model

Call a T-SQL System Stored Procedure from my application and have it trigger R script execution in-database to predict on new dataset. Results are then returned to my application (predictions, plots).



The stored procedure contains R code and executes in-database

```
exec sp_execute_external_script
    @language = ‘R’
,   @script =
-- R code --
```

R script usage from SQL Server

Original R script:

```
IrisPredict <- function(data, model){
 library(e1071)
 predicted_species <- predict(model, data)
 return(predicted_species)
}


library(RODBC)
conn <- odbcConnect("MySqlAzure", uid = myUser, pwd =
myPassword);
Iris_data <-sqlFetch(conn, "Iris_Data");
Iris_model <-sqlQuery(conn, "select model from my_iris_model");
IrisPredict (Iris_data, model);
```

Calling R script from SQL Server:

```
/* Input table schema */
create table Iris_Data (name varchar(100), length int, width int);
/* Model table schema */
create table my_iris_model (model varbinary(max));

declare @iris_model varbinary(max) = (select model from
my_iris_model);
exec sp_execute_external_script
  @language = 'R'
, @script = '
IrisPredict <- function(data, model){
 library(e1071)
 predicted_species <- predict(model, data)
 return(predicted_species)
}
IrisPredict(input_data_1, model);
'
, @parallel = default
, @input_data_1 = N'select * from Iris_Data'
, @params = N'@model varbinary(max)'
, @model = @iris_model
with result sets ((name varchar(100), length int, width int
, species varchar(30)));
```

Values highlighted in yellow are SQL queries embedded in the original R script

Values highlighted in aqua are R variables that bind to SQL variables by name

Why In-Database Analytics with SQL 2016 & R?

Leverage Full Capability of R:

- Rich Statistical, Visualization & Predictive Analytics
- A Large and Growing Skill Base

… including Microsoft R Servers Big Data Capabilities:

- Scalable Computation
- Scalable Data Size

… all Running In-Database:

- Divide Work Between Data Scientists and Data Engineers
- Reduce Data Duplication
- Reduce Data Movement

… While Protecting Information:

- Eliminate Data Movement & Unnecessary Copying
- Leverage Database Data Protections

SQL Server 2016 Enterprise Edition

Copyright Microsoft Corporation. All rights reserved.

What’s new?

R Server 9.1

SQL Server Machine Learning Services

What's new in R Server 9.1

Scalable R based machine learning where the data lives

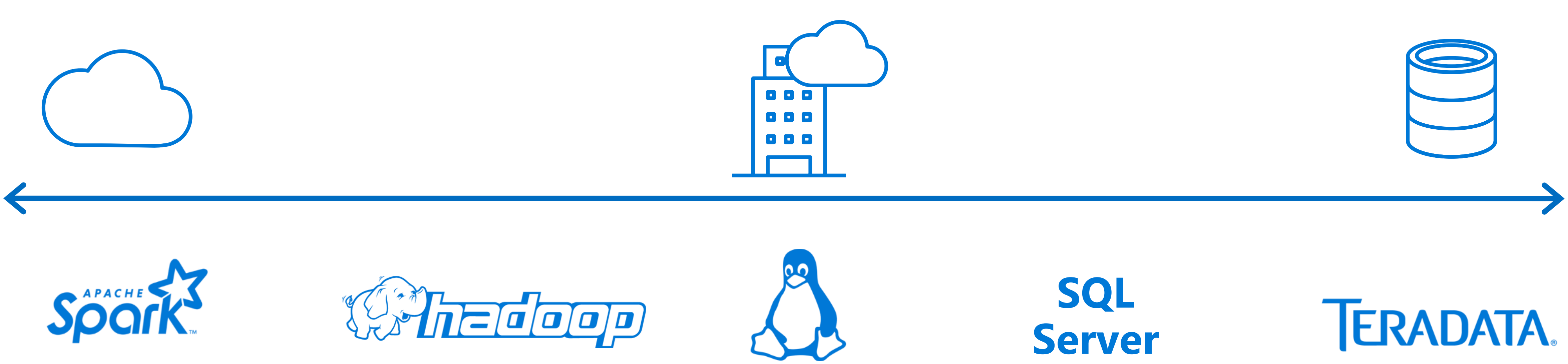
Best of Microsoft Innovation

Hyper scale distributed algorithms
Pre-trained cognitive models
High performance pleasingly parallel support
GPU powered Deep Learning

+

Best of Open Source

CRAN + Bioconductor
SparklyR + H20
Spark ETL
Sparrk SQL



Pre-trained Cognitive Models

Microsoft R Server 9.1 brings pre-trained cognitive models that accelerate time to value and can be re-trained with your data and optimized for your business.

Sentiment Analysis

Enables you to assess the sentiment of an English sentence/paragraph with just a few lines of code.

Image Featurizer

Derive up to 5,000 features on a given image, and use that to compare similarity between two images.

This model can be used in healthcare, research and quality control

For more information on Microsoft ML Libraries please visit: MicrosoftML Library

Optimized Algorithm for Pleasingly Parallel

One of the most popular advanced analytics use cases is Pleasingly Parallel (also known as Embarrassingly or Perfectly Parallel) where clients run massively parallel computations on partitions that are grouped by one or more attributes.

These embarrassingly parallel use cases are common across industries:

- Life sciences simulations to identify the best drug for a given situation
- custom transformation, enrichment and featurization
- Portfolio analysis to identify the right investment for each portfolio
- Utilities to forecast energy consumption for each cohort
- Shipping to forecast demand for various container types

For detailed best practices visit: R Server Blog on embarrassingly-parallel

Enterprise-Grade Operationalization

Easy, secure, and high-performance operationalization is essential for Tier-1 enterprises, at scale, to derive maximum value from their analytics investments. Microsoft R Server 9.1 release continues strengthening the power of operationalization.

- **Real time web services**: realize 10X to 100X boost in scoring performance, scoring speeds at <10ms. Currently on Windows platform; other platforms will be supported soon.

- **Role Based Access Control**: enables admins to control who can publish, update, delete or consume web services

- **Asynchronous batch processing**: speed up the scoring performance for the web services with large input data sets and long-running jobs

- **Asynchronous remote execution**: run scripts in background mode on a remote server, without having to wait for the job to complete

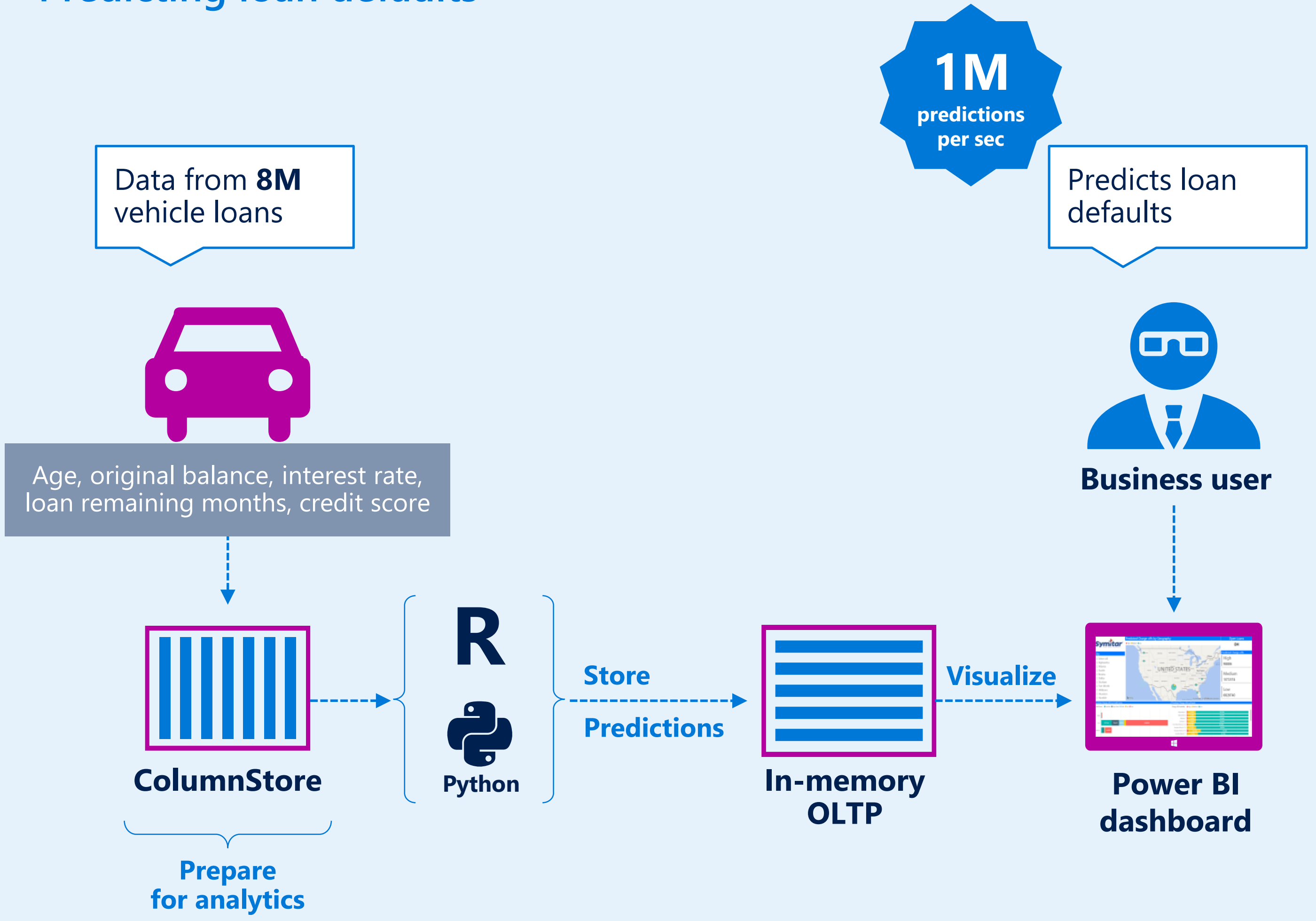
- **Dynamic scaling of operationalization grid with Azure VMs**: easily spin up a set of R Server VMs in Azure, configure them as a grid for operationalization, and scale it up and down based on CPU / Memory usage

What’s new in SQL Server Machine Learning Services

(previously called SQL Server R Services)

Even faster in-database analytics with SQL Server 2017

Predicting loan defaults



1M predictions per sec

Data from **8M** vehicle loans

Predicts loan defaults

Age, original balance, interest rate, loan remaining months, credit score

Business user

ColumnStore

Prepare for analytics

R

Python

Store Predictions

In-memory OLTP

Visualize

Power BI dashboard

- Support for **R and now Python languages**
- **80+ of the most popular Python packages** are parallelized and scalable to help tackle big data
- Enables users to **work with preferred tools** and **push intelligence to where data lives**
- Advanced **machine learning algorithms** with GPUs

SQL Server Machine Learning Services – R specific enhancements

Real-time scoring

- Real-time scoring supported on models trained using RevoScaleR and MicrosoftML algorithms & transforms.
- SQL Server understands these models natively and scores inputs without the need of R interpreter and overhead delivering significantly better performance, assured reliability, lower resource consumption.

Flexible R package management

- Updated RevoScaleR package enables users to install, uninstall and manage packages on SQL Server without administrative access to the SQL Server machine.
- Data scientists and other non-admin users can install packages in databases, user or group scope.
- Updated rxSyncPackages API to ensure that the user-installed packages are not lost if SQL Server node goes down or if the database is migrated.
- The list of packages and the permissions is maintained in a server table and this API ensures that the required packages are installed on the file system

Wrap Up

MRS extends open-source R to allow:

- Multi-threading
  - Matrix operations, linear algebra, and many other math operations run on all available cores
- Parallel processing
  - ScaleR functions utilize all available resources (rxExec)
  - local or distributed (locally parallel, Hadoop, Spark, SQL Server, Teradata)
- On-disk data storage
  - RAM limitations lifted -> XDF, XDFd

Q&A?
Thank you!

Michal.Marusan@Microsoft.com