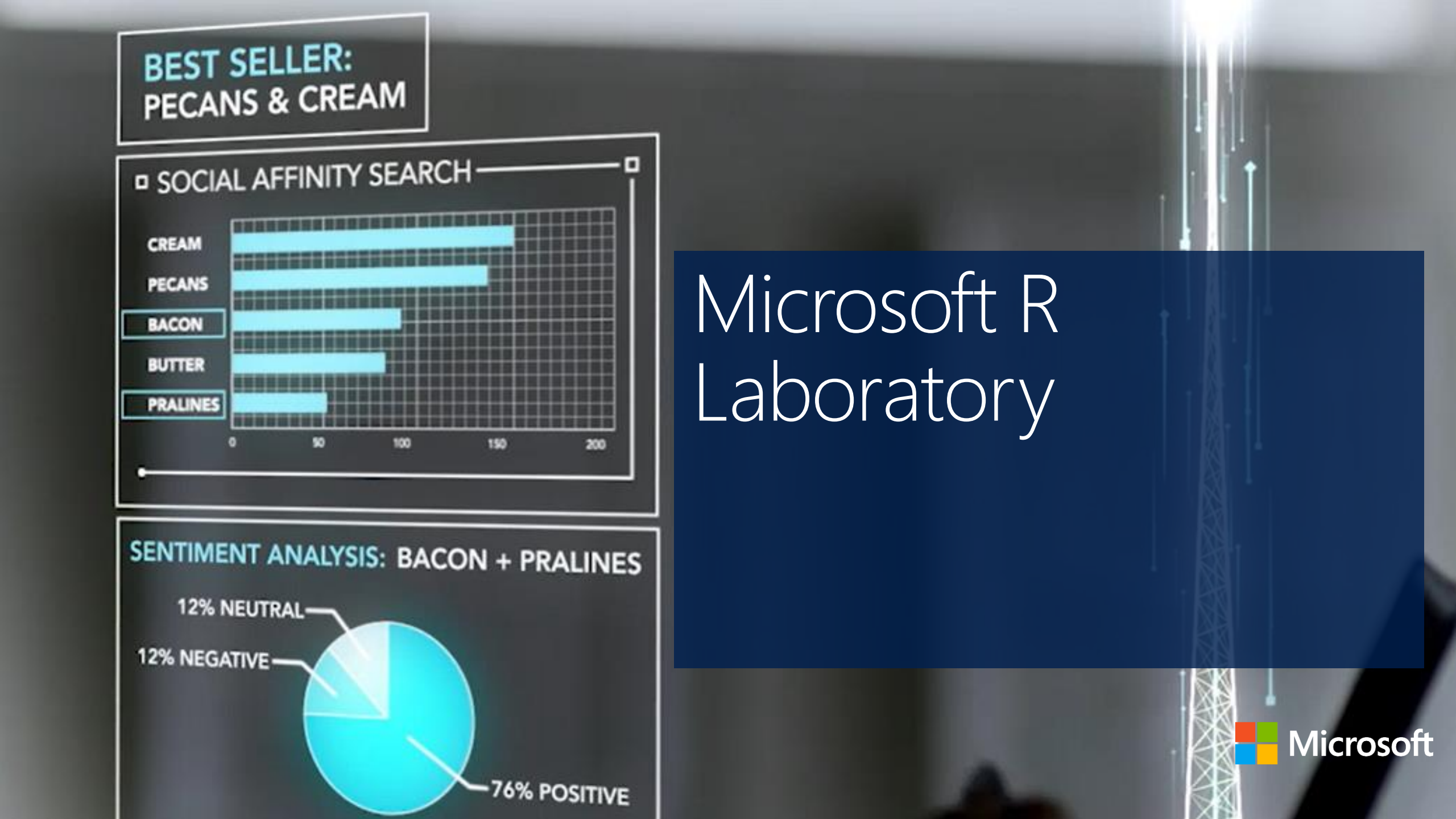


Microsoft R Laboratory

Microsoft

Agenda – Day 1

Who	When	What	How
All	09:30 – 09:45	Coffee, Introductions, Connectivity !	
Instructors	09:45 – 11:00	Microsoft R Server (MRS)	Presentation
You	11:00 – 12:00	Lab 01 : Introduction to Microsoft R Server	Lab
All	12:00 – 13:00	< LUNCH >	
You	13:00 – 14:30	Lab 02 : Data Cleansing & Management with MRS	Lab
You	14:30 – 15:30	Lab 03 : Building Predictive Models with MRS	Lab
All	15:30 – 15:45	< BREAK >	
You	15:45 – 17:00	Lab 04 : Free Lab with MRS	Lab
All	17:00 – 17:15	Wrap-Up : Questions and Answers	Discussion

Agenda – Day 2

Who	When	What	How
Instructors	09:00 – 09:30	R Deployment options	Chalk & Talk
You	09:30 – 11:00	Lab 05: Operationalizing R with Azure Machine Learning	Lab
You	11:00 – 12:30	Lab 08: SQL Server R Services	Lab
All	12:30 – 13:30	< LUNCH >	
You	13:30 – 14:00	Microsoft R Server on Hadoop	Presentation
You	14:00 – 16:00	Lab 07 : Getting started with MRS on HDInsight (Spark)	Lab
All	16:00 – 16:30	Wrap up: Questions and Answers	Discussion

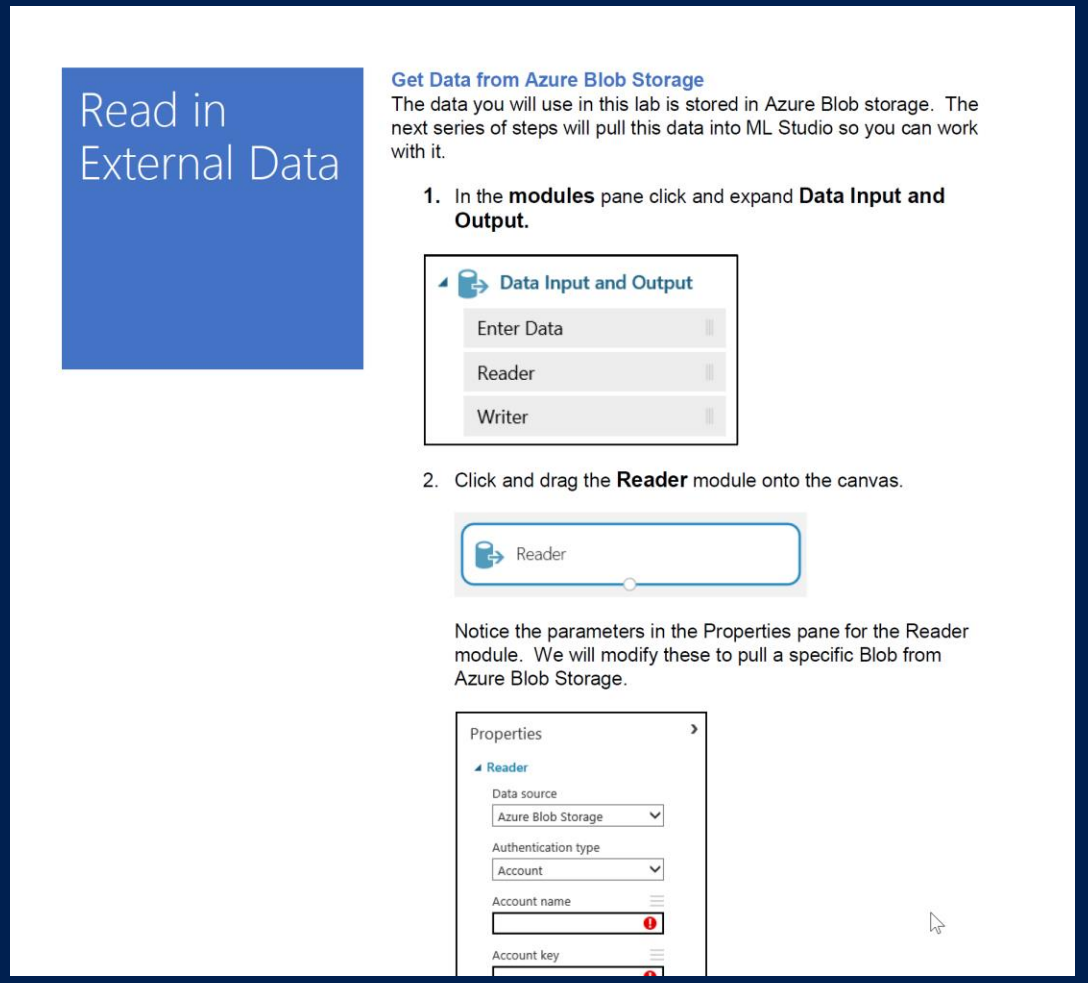
Lab Content

In your Data Science VM go to the following web URL in IE:

https://aka.ms/rlab

Getting the best out of the labs

- Worksheet format
- Follow the instructions
- Explore!
- Ask questions as you are working through
- http://aka.ms/aafellows - Linked in Group



Objectives

- This is NOT a data science course or an introductory R programming course
- An awareness of the Microsoft R Server
- Get over the initial hurdles of
  - Thinking about big data
  - Scaling R
  - Working with R and Hadoop
  - Spark
  - Web services (Azure ML and DeployR)
- Learn how to operationalize analytics using the right components of the technology stack.

Microsoft R Server (MRS)

Introduction

Microsoft

What is R?

Language Platform

- A programming language for statistics, analytics, and data science
- A data visualization framework
- Provided as Open Source

Community

- Used by 2.5M+ data scientists, statisticians and analysts
- Taught in most university statistics programs
- New and recent graduates prefer it
- Active and thriving user groups across the world

Ecosystem

- CRAN: 8000+ freely available algorithms, test data and evaluation
- Many of these are applicable to big data if scaled

R's popularity continues to outpace alternatives

2015 2014

Language Rank	Types	Spectrum Ranking	Spectrum Ranking
1. Java		100.0	100.0
2. C		99.9	99.3
3. C++		99.4	95.5
4. Python		96.5	93.5
5. C#		91.3	92.4
6. R		84.8	84.8
7. PHP		84.5	84.5
8. JavaScript		83.0	78.9
9. Ruby		76.2	74.3
10. Matlab		72.4	72.8

IEEE Spectrum July 2015

Tool Use for Data Science
O'Reilly Data Science Survey 2014
(max=80%)

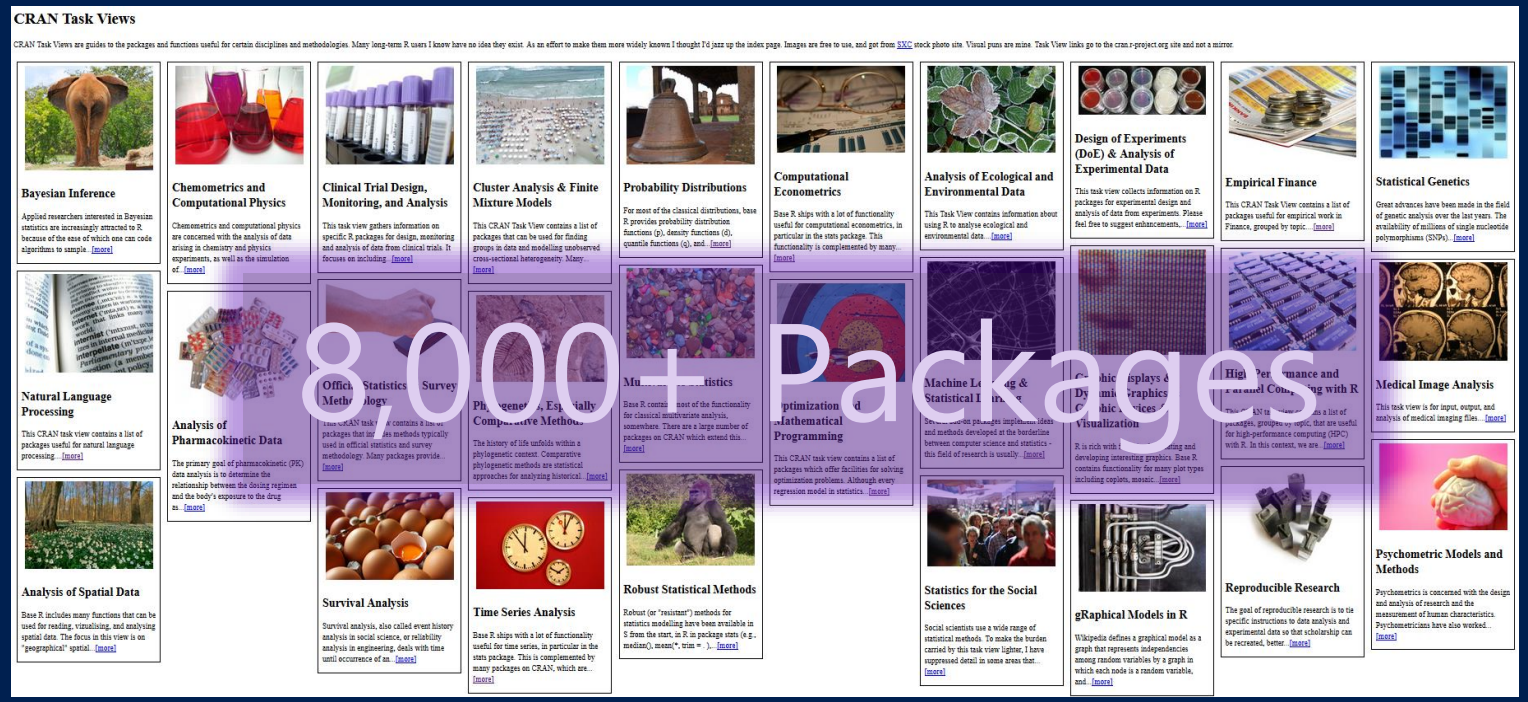
Standing on the Shoulders of Giants

A Vast Community of R Users Share Rich Repositories of Pre-Built Solutions

CRAN

The Comprehensive R Archive Network

Resources For All Fields of Analysis



8,000+ Packages

Microsoft R product suite

Microsoft R Open

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

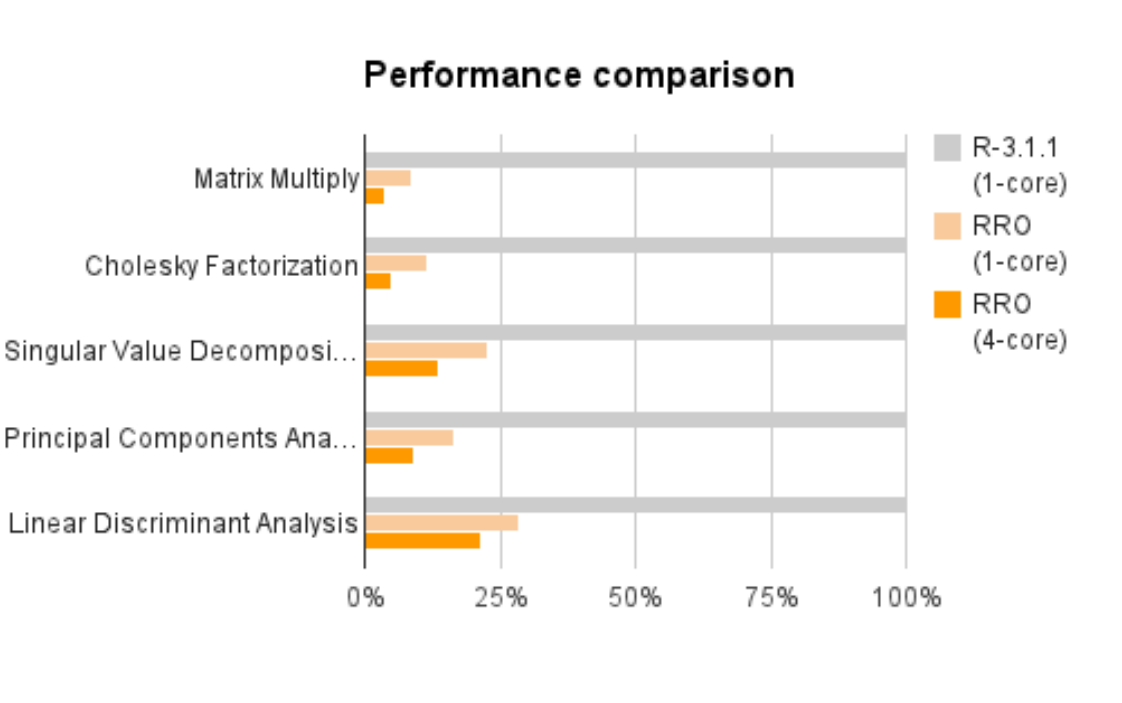
Microsoft R Server

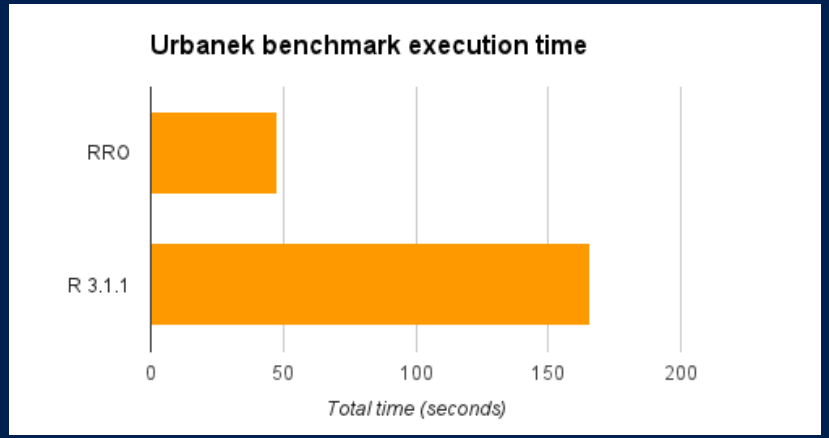
- Secure, Scalable and Supported Distribution of R
- With commercial components created by Microsoft

Microsoft R Open

- Enhanced Open Source R distribution
  - Based on the latest Open Source R (3.2.4)
  - Built, tested and distributed by Microsoft
  - Enhanced by Intel MKL Library to speed up linear algebra functions
- Compatible with all R-related software
  - CRAN packages, RStudio, third-party R integrations, …
- Revolutions Open-Source R packages
  - Reproducible R Toolkit – Checkpoint , miniCRAN
  - ParallelR – parallelise execution via 'foreach' loop
  - RHadoop – rhdfs, rhbase, ravro, rmr2, plyrmr
  - AzureML – read/write data to AzureML, publish R code as ML API
- MRAN website mran.revolutionanalytics.com
  - Enhanced documentation and learning resources
  - Discover 8000 free add-on R packages
- Open source (GPLv2 license) - 100% free to download, use and share

CRAN R compared to Microsoft R Open

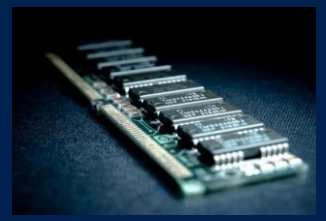




- Matrix calculation – upto 27x faster
- Matrix functions – upto 16x faster
- Programation – 0x faster

- More efficient and multi-threaded math computation.
- Benefits math intensive processing.
- No benefit to program logic and data transform

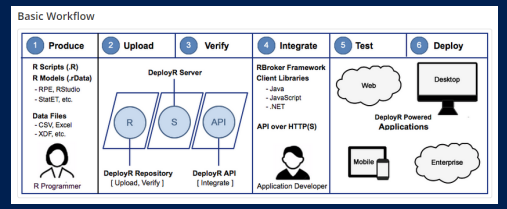
Enterprise use of open source R



R needs data in memory to start a computation*

R is single threaded*

R requires skilled resource to scale out



Model deployment/ integration to business application

Enterprise looking for a commercially supported version of R

*Open source R solutions are not joined up and can be complex to implement

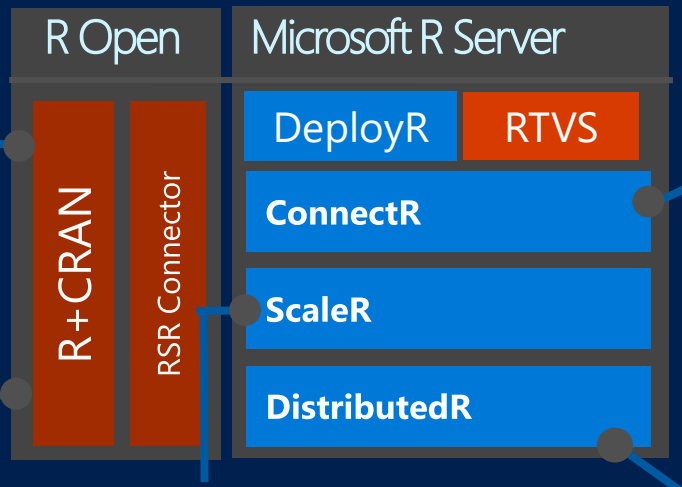
The Microsoft R Server Platform

R+CRAN

- Open source R interpreter
  - R 3.2.4
- Freely-available huge range of R algorithms
- Algorithms callable by MRO
- Embeddable in R scripts
- 100% Compatible with existing R scripts, functions and packages

MRO

- 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

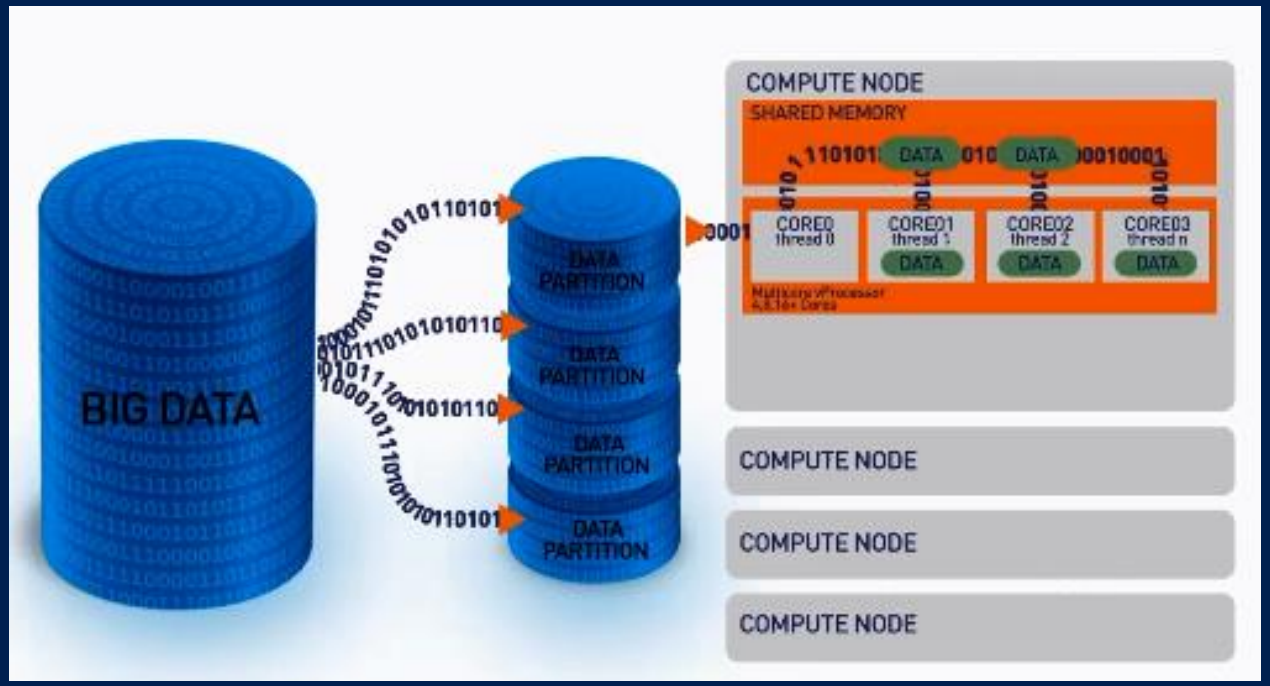
DistributedR

- Distributed computing framework
- Delivers cross-platform portability

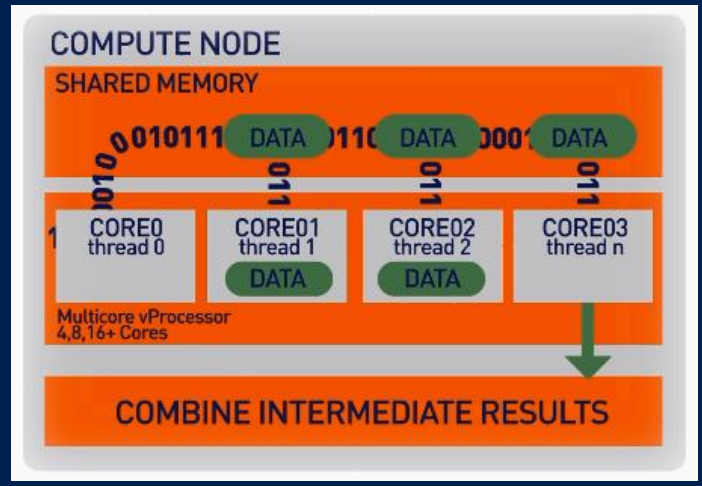
CRAN, MRO, MRS Comparison

		**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	8000+ innovative analytic packages	8000+ innovative analytic packages	**8000+ innovative packages + commercial parallel high-speed functions**
Licence	Open Source	Open Source	**Commercial license. Supported release with indemnity**

ScaleR – Parallel + "Big Data"



Our ScaleR algorithms work inside multiple cores / nodes in parallel at high speed



Stream data in to RAM in blocks. "Big Data" can be any data size. We handle Megabytes to Gigabytes to Terabytes…

XDF file format is optimised to work with the ScaleR library and significantly speeds up iterative algorithm processing.

Interim results are collected and combined analytically to produce the output on the entire data set

Scale R – Parallelized Algorithms & Functions

Data Preparation

- Data import – Delimited, Fixed, SAS, SPSS, OBDC
- Variable creation & transformation
- Recode variables
- Factor variables
- Missing value handling
- Sort, Merge, Split
- Aggregate by category (means, sums)

Descriptive Statistics

- Min / Max, Mean, Median (approx.)
- Quantiles (approx.)
- Standard Deviation
- Variance
- Correlation
- Covariance
- Sum of Squares (cross product matrix for set variables)
- Pairwise Cross tabs
- Risk Ratio & Odds Ratio
- Cross-Tabulation of Data (standard tables & long form)
- Marginal Summaries of Cross Tabulations

Statistical Tests

- Chi Square Test
- Kendall Rank Correlation
- Fisher's Exact Test
- Student's t-Test

Sampling

- Subsample (observations & variables)
- Random Sampling

Predictive Models

- Sum of Squares (cross product matrix for set variables)
- Multiple Linear Regression
- Generalized Linear Models (GLM) exponential family distributions: binomial, Gaussian, inverse Gaussian, Poisson, Tweedie. Standard link functions: cauchit, identity, log, logit, probit. User defined distributions & link functions.
- Covariance & Correlation Matrices
- Logistic Regression
- Classification & Regression Trees
- Predictions/scoring for models
- Residuals for all models

Variable Selection

- Stepwise Regression

Simulation

- Simulation (e.g. Monte Carlo)
- Parallel Random Number Generation

Cluster Analysis

- K-Means

Classification

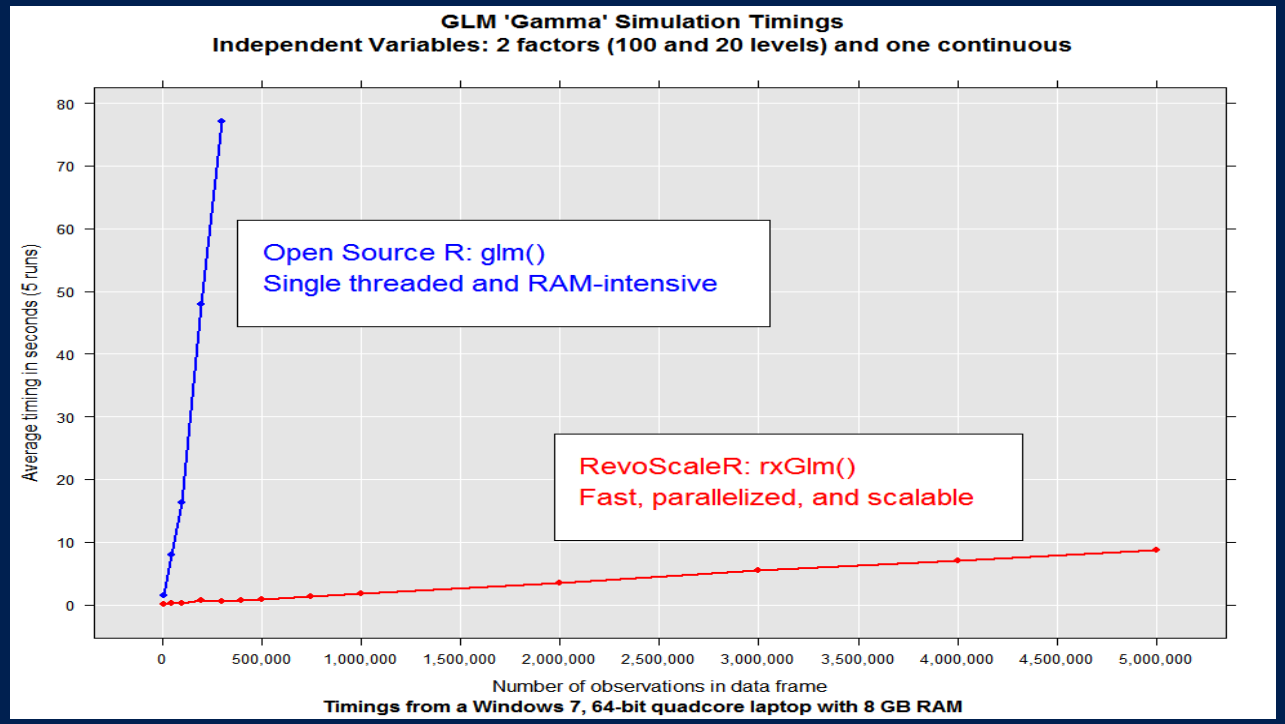
- Decision Trees
- Decision Forest
- Gradient Boosted Decision Trees
- Naïve Bayes

Combination

- rxDataStep
- rxExec
- PEMA API

ScaleR - Performance comparison

Microsoft R Server has no data size limits in relation to size of available RAM. When open source R operates on data sets that exceed RAM it will fail. In contrast Microsoft R Server scales linearly well beyond RAM limits and parallel algorithms are much faster.



File Name	Compressed File Size (MB)	No. Rows	Open Source R (secs)	Revolution R (secs)
Tiny	0.3	1,235	0.00	0.05
V. Small	0.4	12,353	0.21	0.05
Small	1.3	123,534	0.03	0.03
Medium	10.7	1,235,349	1.94	0.08
Large	104.5	12,353,496	60.69	0.42
Big (full)	12,960.0	123,534,969	Memory!	4.89
V.Big	25,919.7	247,069,938	Memory!	9.49
Huge	51,840.2	494,139,876	Memory!	18.92

- US flight data for 20 years
- Linear Regression on Arrival Delay
- Run on 4 core laptop, 16GB RAM and 500GB SSD

Compute Contexts

Microsoft

Write Once Deploy Anywhere

ScaleR functions can run in-Hadoop or in-Database without any functional R recoding

Local Parallel – Linux or Windows

```
# SETUP LINUX ENVIRONMENT VARIABLES
rxSetComputeContext("localpar")

# CREATE LINUX, DIRECTORY AND FILE OBJECTS
linuxFS <- RxNativeFileSystem()

AirlineDataSet <-
RxXdfData("AirlineDemoSmall.xdf", fileSystem =
linuxFS)
```

In – Hadoop

```
### SETUP HADOOP ENVIRONMENT VARIABLES
myHadoopCluster <- RxHadoopMR()

### HADOOP COMPUTE CONTEXT USING HDFS
rxSetComputeContext(myHadoopCluster)

### CREATE HDFS, DIRECTORY AND FILE OBJECTS
hdfsFS <- RxHdfsFileSystem()
 AirlineDataSet <-
 RxXdfData("AirlineDemoSmall.xdf",
    fileSystem = hdfsFS)
```

SQL Server

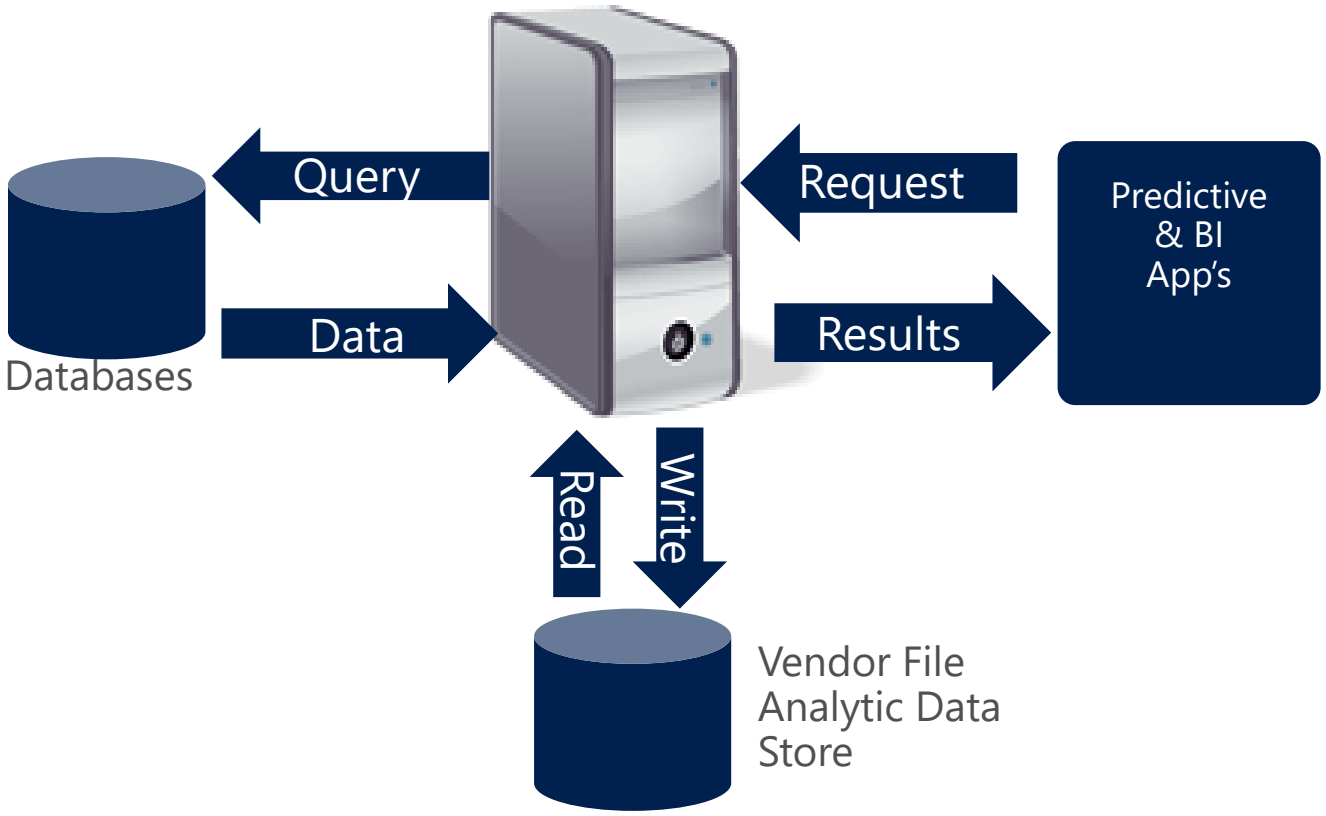
```
# SETUP SQLSERVER ENVIRONMENT VARIABLES
mySqlServer <- RxInSqlServer()

# SQL SERVER COMPUTE CONTEXT AND TABLE REF
rxSetComputeContext(mySqlServer)

AirlineDataSet <-
RxSqlServerData(table="AirlineDemoSmall")
```

R script – does not need to change to run across different platforms

```
### ANALYTICAL PROCESSING ###
### Statistical Summary of the data
    rxSummary(~ArrDelay+DayOfWeek, data= AirlineDataSet, reportProgress=1)

### CrossTab the data
    rxCrossTabs(ArrDelay ~ DayOfWeek, data= AirlineDataSet, means=T)

### Linear Model and plot
    hdfsXdfArrLateLinMod   <- rxLinMod(ArrDelay ~ DayOfWeek + 0 , data = AirlineDataSet)
    plot(hdfsXdfArrLateLinMod$coefficients)
```

The Challenge of Traditional Predictive Analytic Approach

- Users pull data to separate analytics server
- 'ETL' on the data – repeated effort
- Store data locally - avoid data movement latency, transformations,
- Poor data governance and management practices
- Model deployment requires re-coding to SQL or other
- Data locked in proprietary formats, unreadable from other tools



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

Two supported data scientist scenarios

Run R script

- Use your preferred R IDE
- Set compute context to SQL Server
- Use RevoScaleR rx functions
- Wrap open-source R functions within rxExec for execution on SQL Server

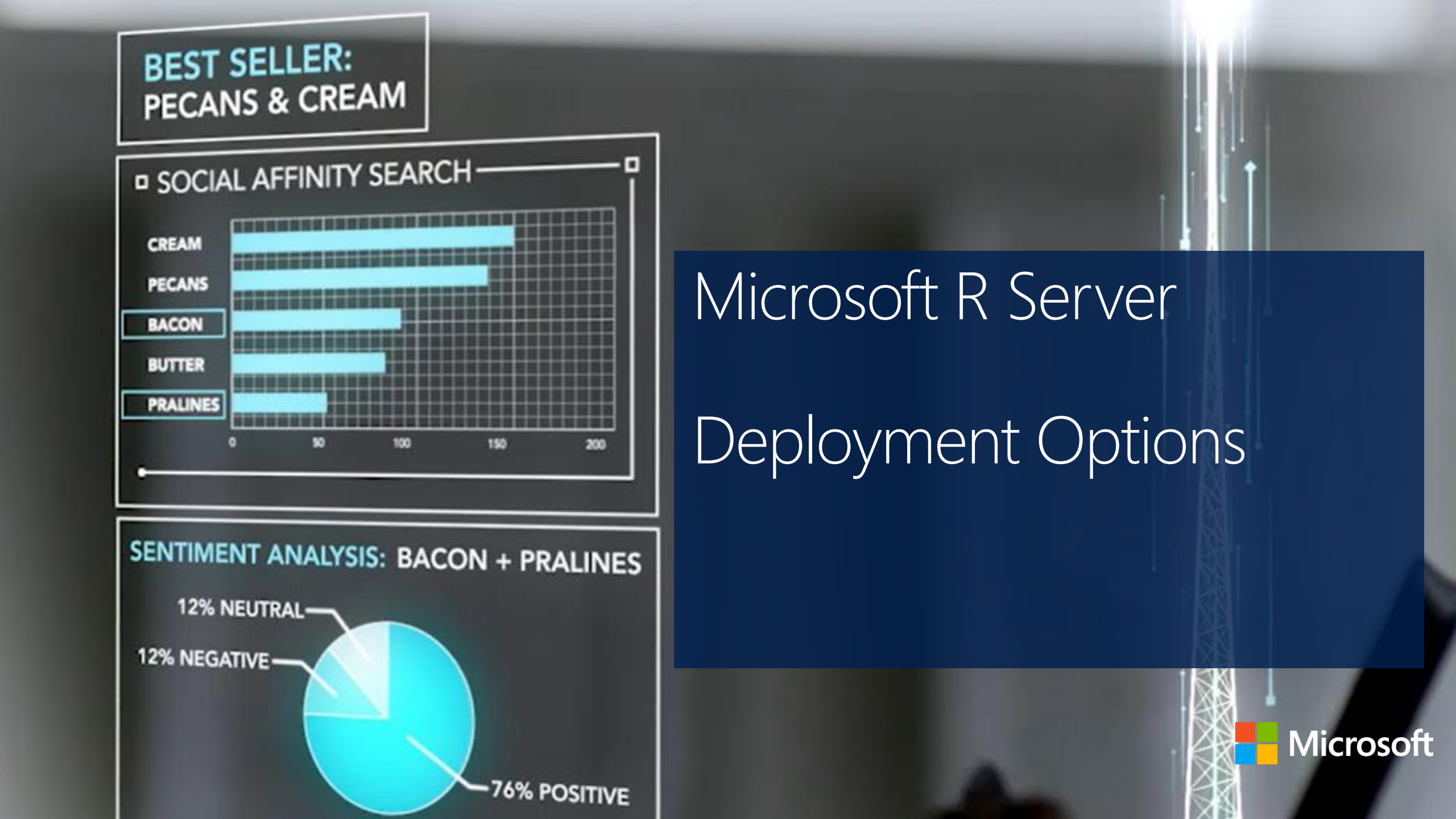
Create SQL query

- Create stored procedure
- Embedded R Language support
- Execute directly in SSMS query

Features of SQLServer R Services

- Define/exploit SQL transformation in a data source and pipeline into ScaleR functionality
- Utilises full parallelism of SQL and ScaleR for fine-grained parallelism
  - Parallel task execution via rxExec
- Processing platform flexibility – change compute-context and/or data source
  - In-database for large datasets
  - Local data exchange support
  - ODBC for small datasets
- Embed, execute and operationalise R within T-SQL
  - Caveat: R session created per stored procedure call. Latency! Good for big data chunks.
  - Data limited to data-frame passed to/from R from SQL engine

Microsoft



Microsoft R Server

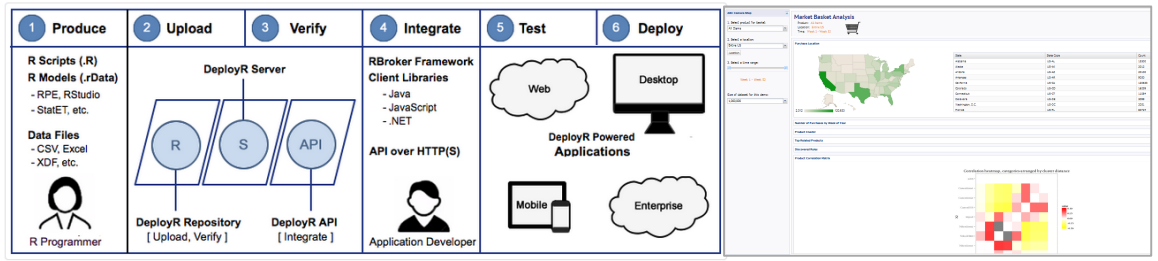
Deployment Options

Microsoft

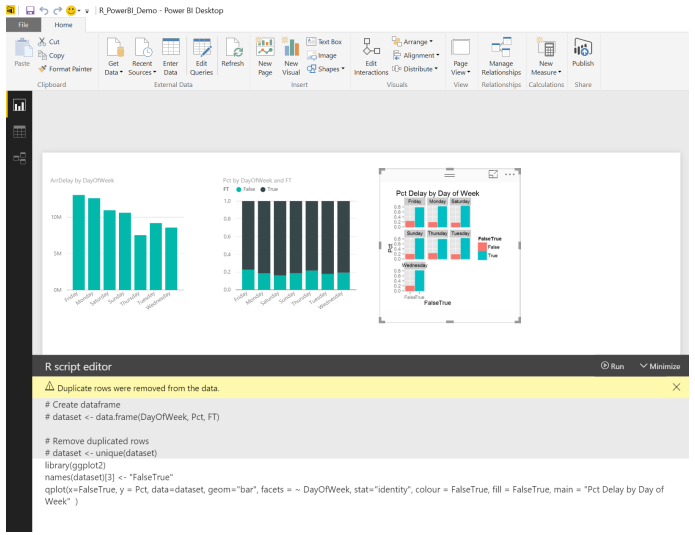
Deployment Acceleration

DeployR { part of Microsoft R Server }

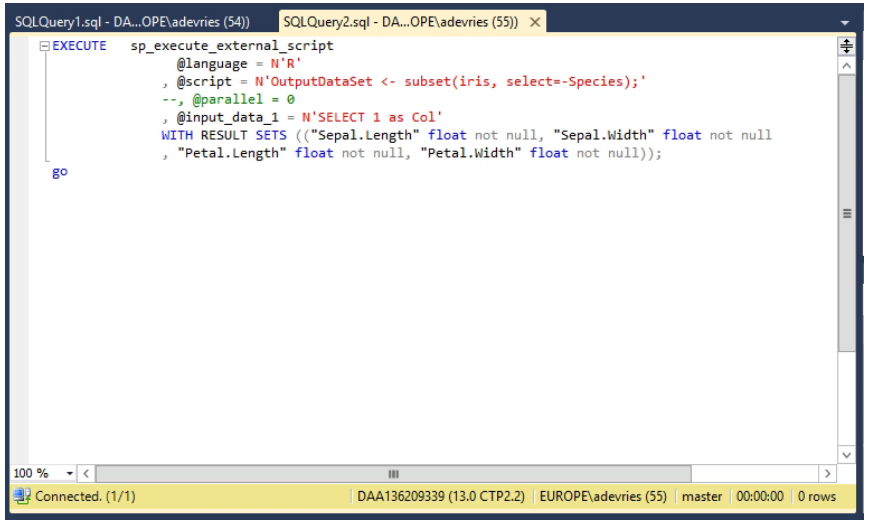
Basic Workflow



Deploy in PowerBI – R Integration



Deploy in SQL Server Stored Procedure



Deploy to Azure

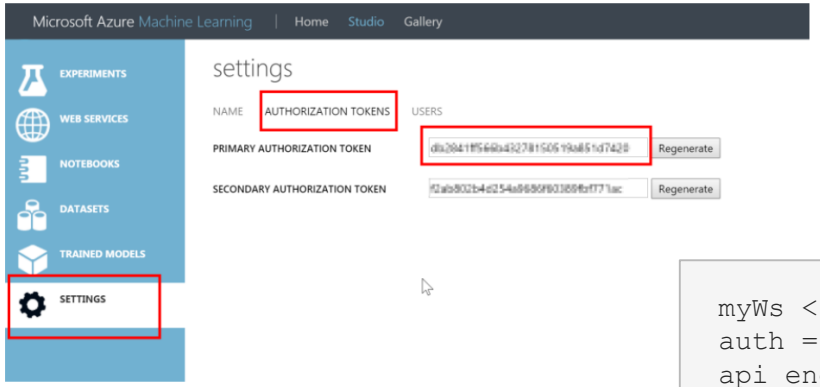
AzureML R package

```
api <- publishWebService(
  ws,
  fun = add,
  name = "aalab-silly",
  inputSchema = list(
    x = "numeric",
    y = "numeric"
  ),
  outputSchema = list(
    ans = "numeric"
  )
)

api
```

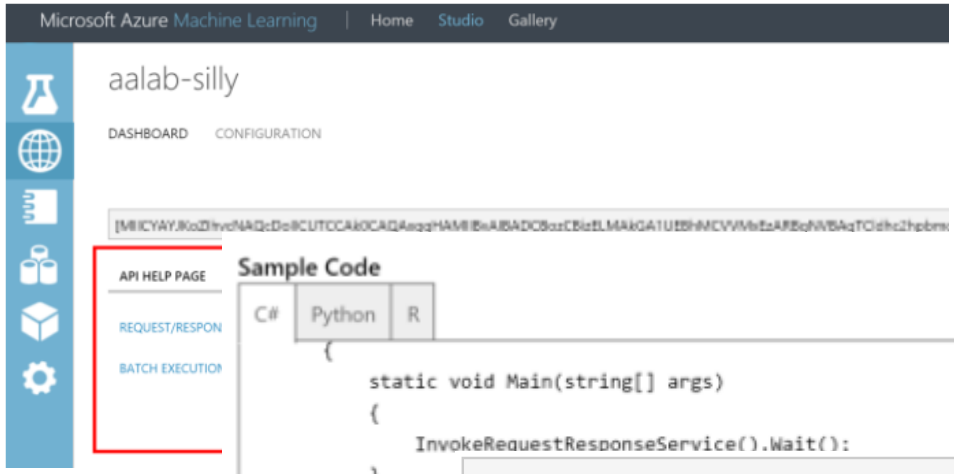
Microsoft

AzureML R Package - Interact & Publish R to AzureML

- Capture workspace & authorisation token
- Create workspace object in R
- Define and publish an R function to AzureML
- Consume web-service e.g. C#, R, Excel etc




```
myWs <- workspace(id = "WORKSPACE ID",
auth = "AUTH KEY",
api_endpoint =
"https://europewest.studio.azureml.net",
management_endpoint =
"https://europewest.management.azureml.net")
```

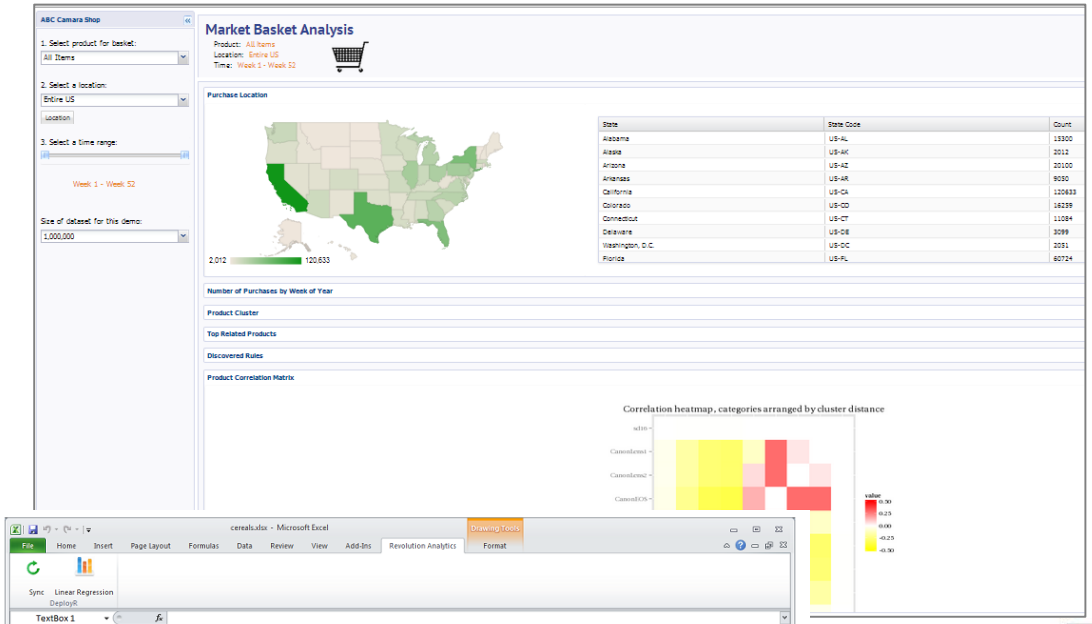



```
df <- data.frame(
  x = 1:5,
  y = 6:10
)
s <- services(ws, name = "aalab-silly")
s <- tail(s, 1) # use the last published function, in case of duplica
te function names
ep <- endpoints(ws, s)
consume(ep, df)
```

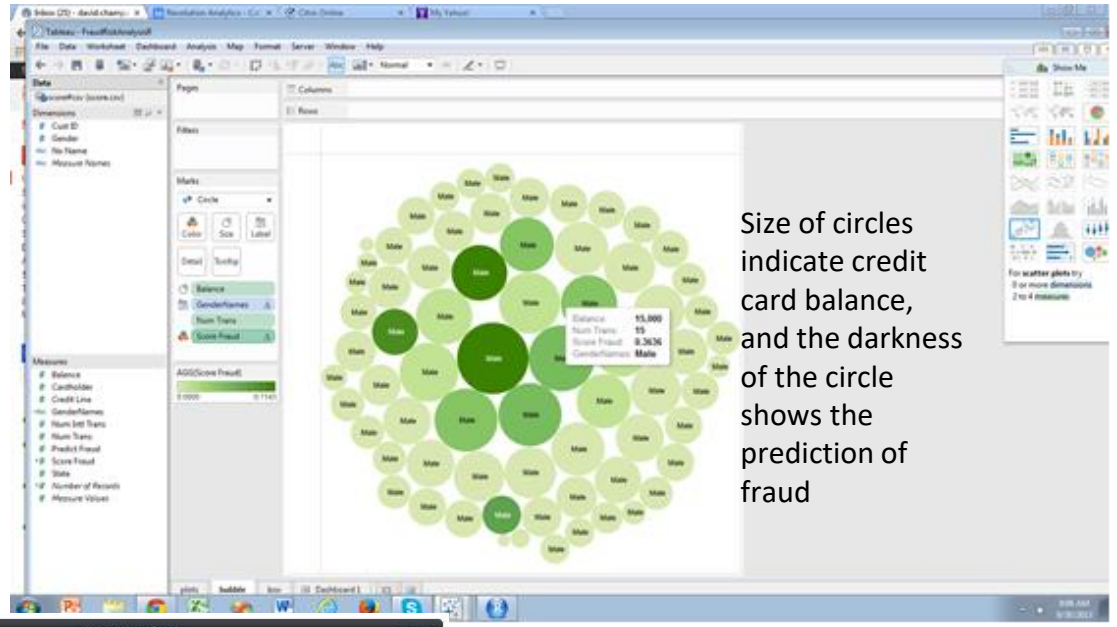
```
api <- publishWebService(
  ws,
  fun = add,
  name = "aalab-silly",
  inputSchema = list(
    x = "numeric",
    y = "numeric"
  ),
  outputSchema = list(
    ans = "numeric"
  )
)

api
```

DeployR: example R as a service for BI / web apps

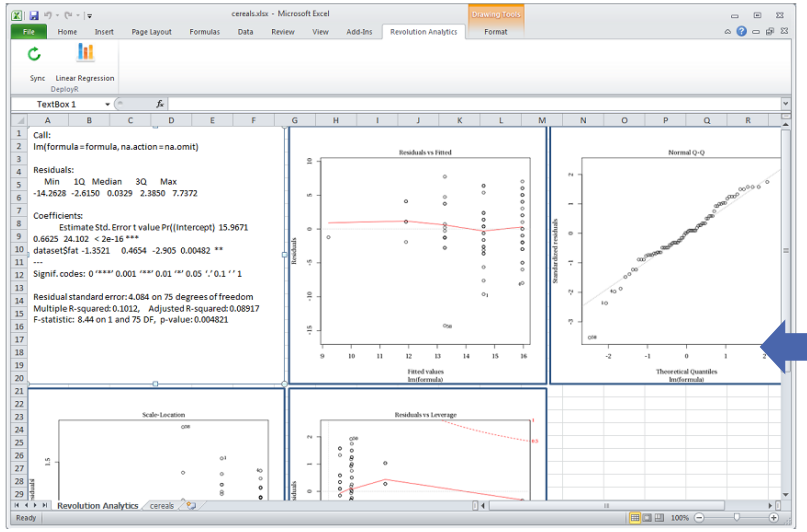
Example: Market Basket Analysis in HTML tool



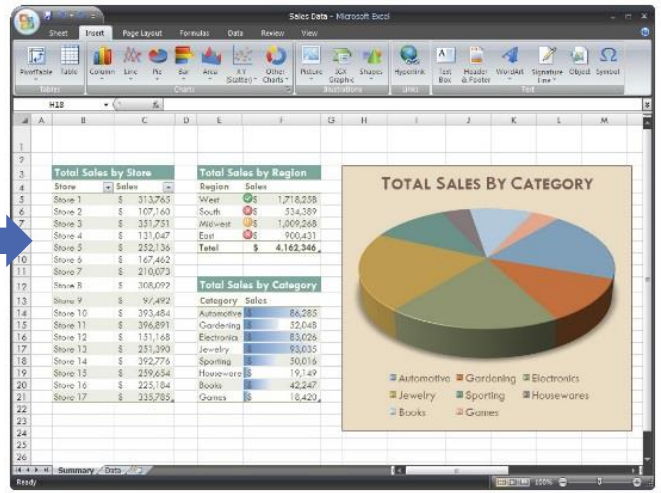
Example: fraud analytics deployed to BI tool



Size of circles indicate credit card balance, and the darkness of the circle shows the prediction of fraud



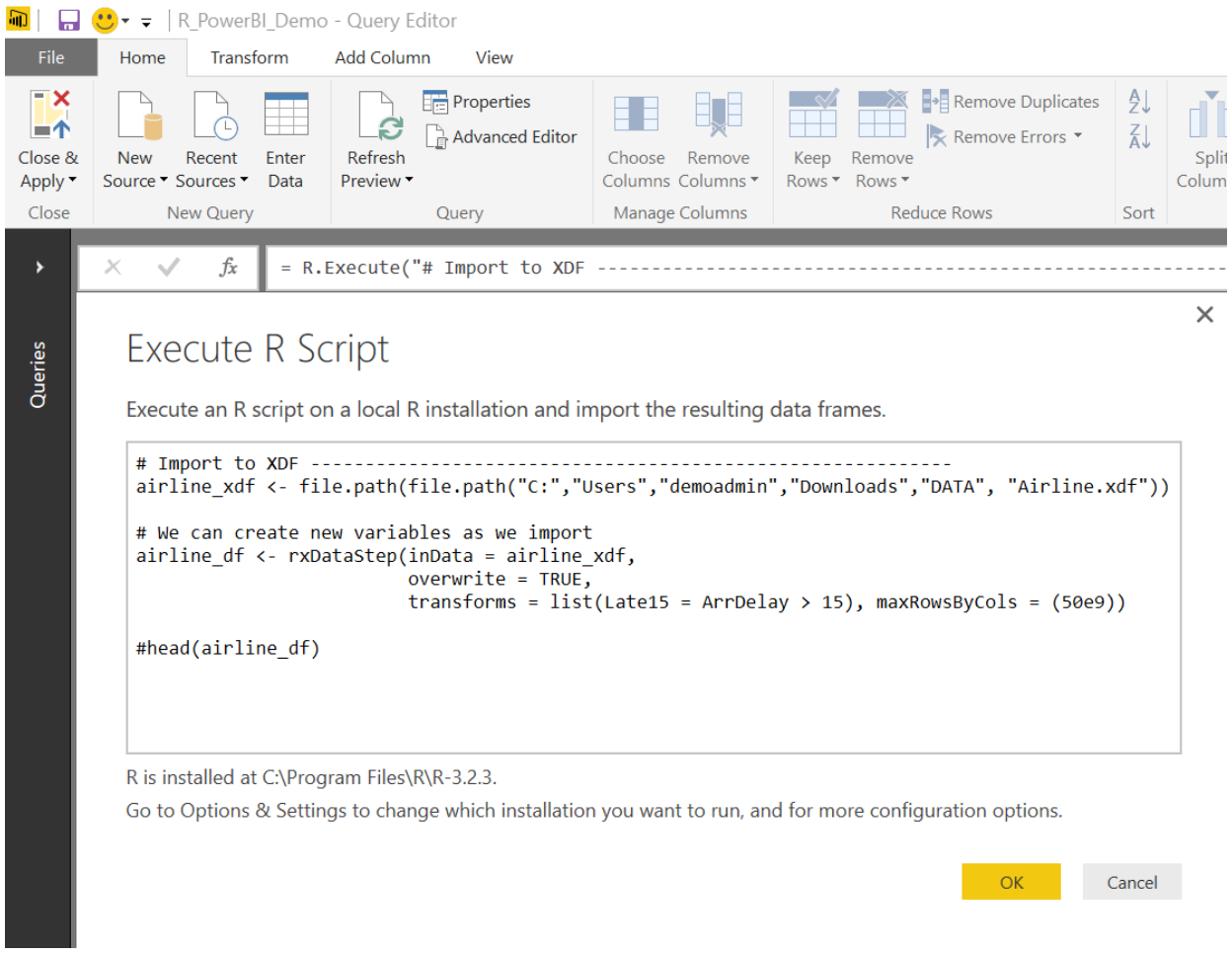
Example: integration with Excel



Microsoft

PowerBI - R Integration

Execute R Scripts to create PowerBI data-sources



Use R Visualisations directly in PowerBI