Learning R Series
Session 1: Introduction to Oracle's R Technologies and Oracle R Enterprise 1.3
Mark Hornick, Senior Manager, Development
Oracle Advanced Analytics
<table>
<thead>
<tr>
<th>Session</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session 1</td>
<td>Introduction to Oracle's R Technologies and Oracle R Enterprise 1.3</td>
</tr>
<tr>
<td>Session 2</td>
<td>Oracle R Enterprise 1.3 Transparency Layer</td>
</tr>
<tr>
<td>Session 3</td>
<td>Oracle R Enterprise 1.3 Embedded R Execution</td>
</tr>
<tr>
<td>Session 4</td>
<td>Oracle R Enterprise 1.3 Predictive Analytics</td>
</tr>
<tr>
<td>Session 5</td>
<td>Oracle R Enterprise 1.3 Integrating R Results and Images with OBIEE Dashboards</td>
</tr>
<tr>
<td>Session 6</td>
<td>Oracle R Connector for Hadoop 2.0 New features and Use Cases</td>
</tr>
</tbody>
</table>
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remain at the sole discretion of Oracle.
Topics

• Introduction
  – R
  – Oracle’s R Strategy
  – Oracle R Enterprise overview

• New features in Oracle R Enterprise 1.3

• Analytics Example and Scenario

• Oracle Advanced Analytics Option

• Summary
What is R?

- R is an Open Source scripting language and environment for statistical computing and graphics

- Started in 1994 as an Alternative to SAS, SPSS & Other proprietary Statistical Environments

- The R environment
  - R is an integrated suite of software facilities for data manipulation, calculation and graphical display

- Around 2 million R users worldwide
  - Widely taught in Universities
  - Many Corporate Analysts and Data Scientists know and use R

- Thousands of open sources packages to enhance productivity such as:
  - Bioinformatics with R
  - Spatial Statistics with R
  - Financial Market Analysis with R
  - Linear and Non Linear Modeling
CRAN Task View – Machine Learning & Statistical Learning

ahaz
arules
BayesTree
Boruta
BPHO
bst
caret
CORElearn
CoxBoost
Cubist
e1071 (core)
earth
elasticnet
ElemStatLearn
evtree
gafit
GAMBoost
gamboostLSS
gbmv
gbm (core)
glmnet
glmpath
GMMBoost
gplasso
hda
ipred
kernlab (core)
klaR
lars
lasso2
LiblineaR
LogicForest
LogicReg
longRPart
mboost (core)
mvpart
ncvreg
nnet (core)
oblique.tree
obliqueRF
pamr
party
partykit
penalized
penalizedSVM
predbayescor
quantregForest
randomForest (core)
randomSurvivalForest
rattle
rda
rdetools
REEMtree
relaxo
rgenoud
rgp
rminer
ROCR
rpart (core)
rpartOrdinal
RPMM
RSNNS
RWeka
sda
SDDA
spam
tgp
tree
TWIX
varSelRF
Why statisticians/data analysts use R

R is a statistics language similar to Base SAS or SPSS statistics

R environment is ..

- Powerful
- Extensible
- Graphical
- Extensive statistics
- OOTB functionality with many ‘knobs’ but smart defaults
- Ease of installation and use
- **Free**

http://cran.r-project.org/
Third Party Open Source IDEs, e.g., RStudio

Traditional R and Database Interaction

- Paradigm shift: R \(\rightarrow\) SQL \(\rightarrow\) R
- R memory limitation – data size, call-by-value
- R single threaded
- Access latency, backup, recovery, security…?
- Ad hoc script execution
Oracle R Enterprise enhances open source R

- Analyze and manipulate data in Oracle Database through R, transparently
- Execute R scripts through the database with data and task parallelism
- Use in-database Predictive Analytics algorithms seamlessly through R
- Scoring R models in the database
- R scripts integrated into SQL language dynamically
- Integrate R into the IT software stack
Oracle’s R Strategic Offerings

*Deliver enterprise-level advanced analytics based on R environment*

- **Oracle R Enterprise**
  - Transparent access to database-resident data from R
  - Embedded R script execution through database managed R engines with SQL language integration
  - Statistics engine

- **Oracle R Distribution**
  - Free download, pre-installed on Oracle Big Data Appliance, bundled with Oracle Linux
  - Enterprise support for customers of Oracle R Enterprise, Big Data Appliance, and Oracle Linux
  - Enhanced linear algebra performance using Intel, AMD, or Solaris libraries

- **ROracle**
  - Open source Oracle *database interface driver* for R based on OCI
  - Maintainer is Oracle – rebuilt from the ground up
  - Optimizations and bug fixes made available to open source community

- **Oracle R Connector for Hadoop**
  - R interface to Oracle Hadoop Cluster on BDA
  - Access and manipulate data in HDFS, database, and file system
  - Write MapReduce functions using R and execute through natural R interface
  - Leverage several native Hadoop-based analytic techniques that are part of ORCH package

©2012 Oracle – All Rights Reserved
Oracle R Distribution

Ability to dynamically load
- Intel Math Kernel Library (MKL)
- AMD Core Math Library (ACML)
- Solaris Sun Performance Library

Oracle Support

- Improve scalability at client and database for embedded R execution
- Enhanced linear algebra performance using Intel’s MKL, AMD’s ACML, and Sun Performance Library for Solaris
- Enterprise support for customers of Oracle Advanced Analytics option, Big Data Appliance, and Oracle Linux
- Free download
- Oracle to contribute bug fixes and enhancements to open source R
Oracle R Enterprise

Function push-down – data transformation & statistics

No changes to the user experience

Scale to large data sets

Embed in operational systems

Development | Production | Consumption
OBIEE Dashboard

Parameterized data selection and graph customization
OBIEE Dashboard
Leverage open source R packages

...to be explored in more detail in Session 5 on OBIEE Integration
Collaborative Execution Model

1. R Engine
   - R-SQL Transparency Framework intercepts R functions for scalable in-database execution
   - Interactive display of graphical results and flow control as in standard R
   - Submit entire R scripts for execution by Oracle Database

2. Oracle Database
   - Database Compute Engine
     - Scale to large datasets
     - Leverage database SQL parallelism
     - Leverage in-database statistical and data mining capabilities
   - Post processing of results

3. R Engine(s) managed by Oracle DB
   - Database manages multiple R engines for database-managed parallelism
   - Efficient parallel data transfer to spawned R engines to emulate map-reduce style algorithms and applications
   - Enables “lights-out” execution of R scripts

Collaborative execution with in-database R engine

Analytic techniques not available in-database
Target Environment with ORE

- Eliminate memory constraint with client R engine
- Execute R scripts at database server machine for scalability and performance
- Execute R scripts in \textit{data parallel} or \textit{task parallel} with database spawned and controlled R engines
- Get maximum value from your Oracle Database
- Get even better performance with Exadata
- Enable integration and management through SQL
Oracle R Enterprise – Packages and R Engines

* ORD available on Linux, AIX, Solaris, SPARC platforms
aggdata <- aggregate(ONTIME_S$DEST, 
                       by = list(ONTIME_S$DEST), 
                       FUN = length)

class(aggdata)
head(aggdata)

select DEST, count(*)
from ONTIME_S
group by DEST
Transparency Layer
Overloads graphics functions for in-database statistics

```r
ontime <- ONTIME_S
delay <- ontime$ARRDELAY
dayofweek <- ontime$DAYOFWEEK
bd <- split(delay, dayofweek)
boxplot(bd, notch = TRUE, col = "red", cex = 0.5,
  outline = FALSE, axes = FALSE,
  main = "Airline Flight Delay by Day of Week",
  ylab = "Delay (minutes)", xlab = "Day of Week")
axis(1, at=1:7, labels=c("Monday", "Tuesday",
  "Wednesday", "Thursday",
  "Friday", "Saturday", "Sunday"))
axis(2)
```

...to be explored in more detail in Session 2 on Transparency Layer
Embedded R Execution – R Interface

Data parallel in-database execution

Also includes
- ore.doEval
- ore.tableApply
- ore.rowApply
- ore.indexApply

```r
modList <- ore.groupApply(
  X=ONTIME_S,
  INDEX=ONTIME_S$DEST,
  function(dat) {
    lm(ARRDELAY ~ DISTANCE + DEPDELAY, dat)
  });
modList_local <- ore.pull(modList)
summary(modList_local$BOS) # return model for BOS
```

...to be explored in more detail in Session 3 on Embedded R Execution
Embedded R Execution – SQL Interface
For model build and batch scoring

begin
  sys.rqScriptDrop('Example2');
  sys.rqScriptCreate('Example2',
    'function(dat, datastore_name) {
      mod <- lm(ARRDELAY ~ DISTANCE + DEPDELAY, dat)
      ore.delete(datastore_name)
      ore.save(mod, name=datastore_name)
    }');
end;
/

select *
from table(rqTableEval(
  cursor(select ARRDELAY, DISTANCE, DEPDELAY
    from ontime_s,
    cursor(select 1 "ore.connect",
      'myDatastore' as "datastore_name"
      from dual),
    'XML',
    'Example2'))));

begin
  sys.rqScriptCreate('Example3',
    'function(dat, datastore_name) {
      ore.load(datastore_name)
      prd <- predict(mod, newdata=dat)
      prd[as.integer(rownames(prd))] <- prd
      res <- cbind(dat, PRED = prd)
      res}
    }');
end;
/

select *
from table(rqTableEval(
  cursor(select ARRDELAY, DISTANCE, DEPDELAY
    from ontime_s
    where year = 2003
    and month = 5
    and dayofmonth = 2),
    cursor(select 1 "ore.connect",
      'myDatastore' as "datastore_name" from dual),
    'select ARRDELAY, DISTANCE, DEPDELAY, 1 PRED from ontime_s',
    'Example3'))
order by 1, 2, 3;
Embedded R Execution – SQL Interface

rqTableEval + datastore for model building

SQL Interface

SQL Developer

Oracle Database

User tables

Datastore

myDatastore (mod)

rq*Apply () interface

DB R Engine

ORE

extproc

©2012 Oracle – All Rights Reserved
Statistics Engine

**Example Features**

- **Special Functions**
  - Gamma function
  - Natural logarithm of the Gamma function
  - Digamma function
  - Trigamma function
  - Error function
  - Complementary error function

- **Tests**
  - Chi-square, McNemar, Bowker
  - Simple and weighted kappas
  - Cochran-Mantel-Haenzel correlation
  - Cramer's V
  - Binomial, KS, t, F, Wilcox

- **Base SAS equivalents**
  - Freq, Summary, Sort
  - Rank, Corr, Univariate

- **Density, Probability, and Quantile Functions**
  - Beta distribution
  - Binomial distribution
  - Cauchy distribution
  - Chi-square distribution
  - Exponential distribution
  - F-distribution
  - Gamma distribution
  - Geometric distribution
  - Log Normal distribution
  - Logistic distribution
  - Negative Binomial distribution
  - Normal distribution
  - Poisson distribution
  - Sign Rank distribution
  - Student's t distribution
  - Uniform distribution
  - Weibull distribution
  - Density Function
  - Probability Function
  - Quantile
Oracle R Enterprise

Main components

- **Transparency Layer**
  - Work solely from R for data preparation, analysis, and visualization
  - Use database as compute engine with query optimization and parallelism
  - Eliminates need to manage flat file data – complexity, backup, recovery, security
  - Eliminates R memory constraints so you can handle bigger data
  - No knowledge of SQL required

- **Embedded R Execution**
  - *Roll your own* techniques in R and execute closer to database data
  - Leverage CRAN open source packages
  - Lights-out execution for integrated operationalizing R scripts via SQL interface
  - Leverage user-defined, dba-controlled, and database-managed, data parallel R execution
  - Combine with benefits of Transparency Layer and Statistics Engine capabilities
  - Enables integration of structured and graph results with OBIEE dashboards and BIP documents

- **Statistics Engine**
  - Enable standard and advanced statistics for in-database execution
  - Provide in-database scoring of R models
New features in Oracle R Enterprise 1.3
Oracle R Enterprise 1.3 – Themes

• Big Data

• Time Series Analytics

• Rapid Application Deployment

• Certification for R version 2.15.1
Support for Big Data Analytics

• Exadata storage tier scoring for R models with the new ORE package **OREpredict**

• Comprehensive in-database sampling techniques

• New ORE package, **OREdm**, for high performance in-database predictive algorithms from Oracle Data Mining
Exadata storage tier scoring for R models

• Fastest way to operationalize R-based models for scoring in Oracle Database

• Go from model to SQL scoring in one step
  – No dependencies on PMML or any other plugins

• R packages supported out-of-the-box include
  – glm, glm.nb, hclust, kmeans, lm, multinom, nnet, rpart

• Models can be managed in-database using ORE datastore

…to be explored in more detail in Session 4 on Predictive Analytics
High performance in-database sampling Techniques

- Simple random sampling
- Split data sampling
- Systematic sampling
- Stratified sampling
- Cluster sampling
- Quota sampling
- Accidental sampling

...to be explored in more detail in Session 2 on Transparency Layer
Example: Bag of Little Bootstraps

Approach big data analysis by first randomly partitioning a data set into subsets that can be analyzed using in-memory R algorithms and then aggregating the results from those partitions

- Assign a random partition number to each observation as a derived column (a relational view)
  
  ```r
  x = myTable
  nrowX <- nrow(x)
  x$partition <- sample(rep(1:k, each = nrowX/k, length.out = nrowX), replace = TRUE)
  ```

- Generate the bootstraps in-database and efficiently pass data to R engines
  
  ```r
  results <- ore.groupApply(x, x$partition, function(y) {...}, parallel = TRUE)
  ```

- Build multiple models and aggregate results via voting or averaging
The “Bagging” Concept

Data for Building → Data Samples

- S1
- S2
- ... Sn

Build Models

- S1 → M1 → P1
- S2 → M2 → P2
- ... Sn → Mn → Pn

Individual Model Predictions

Voting or Averaging

Final Prediction

Data to Score
“Bagging” Execution Model

Two options: client-controlled and database-controlled

Client R Engine

Multiple invocations of `ore.lm` to Oracle Database

Client R Engine

Single invocation to Oracle Database using `ore.indexApply`
High performance in-database predictive techniques available through ORE packages

Parallel, distributed, in-database execution

- SVM
- GLM
- k-Means clustering
- Naïve Bayes
- Decision Trees
- Attribute Importance
- Neural Networks
- Stepwise Linear Regression

OREedm

OREeda
Example using OREdm functions

Highlighting Support Vector Machine algorithm

x <- seq(0.1, 5, by = 0.02)
y <- log(x) + rnorm(x, sd = 0.2)
dat <- ore.push(data.frame(x=x, y=y))

# Regression
svm.mod <- ore.odmSVM(y~x,dat,"regression",
kernel.function="linear")
summary(svm.mod)
coef(svm.mod)
svm.res <- predict(svm.mod,dat,supplemental.cols="x")
head(svm.res,6)

m <- mtcars
m$gear <- as.factor(m$gear)
m$cyl <- as.factor(m$cyl)
m$vs <- as.factor(m$vs)
m$ID <- 1:nrow(m)
MTCARS <- ore.push(m)

# Classification
svm.mod <- ore.odmSVM(gear ~ .-ID, MTCARS,"classification")
summary(svm.mod)
coef(svm.mod)
svm.res <- predict (svm.mod, MTCARS,"gear")
with(svm.res, table(gear,PREDICTION))  # generate confusion matrix

# Anomaly Detection
svm.mod <- ore.odmSVM(~ .-ID, MTCARS,"anomaly.detection")
summary(svm.mod)
svm.res <- predict (svm.mod, MTCARS, "ID")
head(svm.res)
table(svm.res$PREDICTION)

…to be explored in more detail in Session 4 on Predictive Analytics
Time Series Analysis

Motivation

• Time series data is widely prevalent
  – Stock / trading data
  – Sales data
  – Employment data

• Need to understand trends, seasonable effects, residuals
Time Series Analysis

- Aggregation and moving window analysis of large time series data
- Equivalent functionality from popular R packages for data preparation available in-database

CRAN Task View: Time Series Analysis

Maintainer: Rob J. Hyndman
Contact: Rob@Hyndman.co.nz
Version: 2012-10-07

Base R ships with a list of functionality useful for time series, as well as in the stats package. This is complemented by many packages on CRAN, which are briefly summarized below. There is also a considerable overlap between the tools for time series and those in the Econometrics and Finance task views. The packages in this view can be roughly structured into the following topics. If you think that some package is missing from the list, please let us know.

Basics
- Infrastructure: Base R contains substantial infrastructure for representing and analyzing time series data. The fundamental class is \texttt{ts} that can represent regular time series (using numeric time stamps). Hence, it is particularly well-suited for annual, monthly, quarterly, daily, etc.
- Modeling: Methods for analyzing and modeling time series include ARIMA models in \texttt{arima}, \texttt{AR} and \texttt{VAR} models in \texttt{ar}, structural models in \texttt{StructTS}, visualization in \texttt{vars} (plotting), prediction functions in \texttt{forecast}, classical decomposition in \texttt{decomp}, STL decomposition in \texttt{stl}, moving averages and autoregressive filter functions in \texttt{filter}, and basic Holt-Winters forecasting in \texttt{HoltWinters}.

Time Series Classes
- As mentioned above, \texttt{ts} is the basic class for regularly spaced time series using numeric time stamps.
- The \texttt{zoo} package provides infrastructure for regularly and irregularly spaced time series using arbitrary classes for the time stamps (i.e., allowing all classes from the previous sentence). It is designed to be as consistent as possible with \texttt{ts}. Coercion from and to \texttt{zoo} is available for all other classes mentioned in this section.
- The package \texttt{xts} is based on \texttt{zoo} and provides uniform handling of R's different time-based data classes.
- Various packages implement irregular time series based on \texttt{POSIXt} time stamps, intended especially for financial applications. These include \texttt{xts} from \texttt{xts}, \texttt{zoo}, \texttt{rugarch}, and \texttt{tsibble} from \texttt{tsibble}.
- The class \texttt{tsibble} in \texttt{tsibble} (previously \texttt{xts}) implements time series with \texttt{tsibble} time stamps.
- The package \texttt{dtseries} implements time series with \texttt{dt} time stamps.
- The package \texttt{tsibble} contains infrastructure for setting time frames in different formats.

Forecasting and Univariate Modeling
- The \texttt{forecast} package provides a class and methods for univariate time series forecasts, and provides many functions implementing different forecasting models including all those in the stats package.
- \texttt{Exponential smoothing} (\texttt{ets}) in \texttt{stats} provides some basic univariate models with partial optimization, \texttt{stlm} from \texttt{forecast} provides a larger set of models and facilities with full optimization.
- \texttt{Auto-regressive models} (\texttt{ar}) in \texttt{stats} (with model selection), \texttt{ggARFIMA} for subset AR models, and \texttt{ar} for periodic autoregressive time series models.
- \texttt{ARIMA models} (\texttt{ar}) in \texttt{stats} is the basic function for ARIMA, SARIMA, ARMAX, and subset ARIMA models. It is enhanced in the \texttt{forecast} package along with \texttt{auto.arima} for automatic order selection. \texttt{arma} in the \texttt{tseries} package provides different algorithms for ARMA and subset ARIMA models. \texttt{fitARMA} implements a fast MLE algorithm for ARMA models. Some facilities for fractional differenced ARFIMA models are provided in the \texttt{fracdiff} package. \texttt{aRMS} adds estimation, diagnostics and forecasting for ARFIMA models. \texttt{armaFit} from the \texttt{arima} package is an interface for ARIMA and ARFIMA models. Package \texttt{arma} contains functionality for generalized ARIMA time series simulation. The \texttt{mets} package handles multivariate AR(1) with seasonal processes.
- \texttt{GARCH models} (\texttt{garch}) from \texttt{tseries} fits basic \texttt{GARCH} models, \texttt{rugarch} from \texttt{rugarch} implements ARIMA models with a wide class of GARCH innovations. \texttt{bbricks} estimates a Bayesian \texttt{GARCH}(1,1) model with innovations. \texttt{rugarch} implements Generalized Orthogonal GARCH (GO-GARCH) models. The \texttt{R}-Forge project \texttt{rfgarch} aims to provide a flexible and sub-GARCH modelling and testing environment including additive and multi-variate GARCH packages. Its \texttt{webarch} has extensive information and examples.
- \texttt{Miscellaneous}: \texttt{tseries} contains methods for linear time series analysis. \texttt{BATS} for Bayesian analysis of dynamic linear models. \texttt{timseries} for time series analysis and control.
- \texttt{Forecast} for bias-corrected forecasting and bootstrap prediction intervals for autoregressive time series.

Resampling
- \texttt{Bootstraping:} The \texttt{boot} package provides function \texttt{tsboot} for time series bootstrapping, including block bootstrap with several variants. \texttt{tsbootstrap} from \texttt{ts} provides fast stationary and block bootstrapping. Maximum entropy bootstrap for time series is available in \texttt{metboot}.
Support for Time Series Data

• Support for Oracle data types
  – DATE,TIMESTAMP
  – TIMESTAMP WITH TIME ZONE
  – TIMESTAMP WITH LOCAL TIME ZONE

• Analytic capabilities
  – Date arithmetic, Aggregations & Percentiles
  – Moving window calculations:
    - ore.rollmax
    - ore.rollmean
    - ore.rollmin
    - ore.rollsd
    - ore.rollsum
    - ore.rollvar
    - ore.rollsd

…to be explored in more detail in Session 2 on Transparency Layer
Rapid Application Deployment

*Motivation and enabling features*

• Streamline and simplify application deployment
  – Avoid data staging, movement, and latency

• Increase data security

• Embed ORE into application backends and web UI infrastructures

• Allow applications to integrate with ORE to leverage:
  – Execution of R in-database via R-to-SQL transparency layer
  – In-database high performance predictive techniques in concert with R algorithms
  – R integration into SQL language
  – Persistence of R objects in Oracle Database
  – In-database scoring using models from R algorithms
Rapid Application Deployment

**Benefits**

- Database is the server managing instances of R in-database
- Data and task parallel execution of R scripts in Oracle Database
- Use cases include
  - Bag of Little Bootstraps
  - Partitioned model builds
  - Simulations and backtesting
- Resource utilization of R instances automatically managed by Oracle Database
- R models and objects stored securely in database-managed R datastore
- No additional packages (like Rserve) or maintenance required
Using ORE with CRAN package and visualization

Data preparation using ORE, movie recommendations using \{arules\}

```r
MF <- MOVIE_FACT[,c("CUST_ID","MOVIE_ID","ACTIVITY_ID")]
MV <- MOVIE[,c("MOVIE_ID","TITLE")]

transData <- merge(MF[MF$ACTIVITY_ID==2,], MV,
                    by="MOVIE_ID")
transData <- ore.pull(transData[,c("CUST_ID","TITLE")])
transData <-
    data.frame(CUST_ID=as.factor(transData$CUST_ID),
                TITLE=as.factor(transData$TITLE))

library(arules)
trans.movie <- as(split(transData[,"TITLE"],
                       transData[,"CUST_ID"],
                       "transactions")

assocRules <-
    apriori(trans.movie,
            parameter=list(minlen=2,
                           maxlen=2,
                           support=0.05,
                           confidence=0.1))

inspect(sort(assocRules,by="support")[1:25])
plot(sort(assocRules,by="support")[1:50],
     method="graph",
     interactive=TRUE,
     control=list(type="items"))
```
Results

R> assocRules <- apriori(trans.movie,
+   parameter=list(minlen=2,
+                 maxlen=2,
+                 support=0.05,
+                 confidence=0.1))

parameter specification:
  confidence minval smax arem aval originalSupport support minlen maxlen target ext
  0.1    0.1    1    none    FALSE        TRUE    0.05    2    2    rules    FALSE

algorithmic control:
  filter tree heap memopt load sort verbose
  0.1    TRUE    TRUE    FALSE    TRUE    2    TRUE

apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09)    (c) 1996-2004 Christian Borgelt
set item appearances ...[9 item(s)] done [0.00s].
set transactions ...[970 item(s), 4427 transaction(s)] done [0.06s].
sorting and recoding items ...[429 item(s)] done [0.00s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 done [0.02s].
writing ...[25590 rule(s)] done [0.06s].
creating S4 object ... done [0.01s].
R> inspect(sort(assocRules,by="support")[1:25])

   lhs                  rhs                          support confidence   lift
1  {Candyman}   {The Time Machine}   0.2256607  0.8560411    2.472981
2  {The Time Machine}   {Candyman}   0.2256607  0.7494374    2.472981
3  {Memento}   {The Time Machine}   0.2236277  0.8790968    2.946120
4  {The Time Machine}   {Memento}   0.2236277  0.7426857    2.946120
5  {Memento}   {Candyman}   0.2175288  0.8629932    3.273413
6  {Candyman}   {Memento}   0.2175288  0.8251280    3.273413
7  {American Beauty}   {The Time Machine}   0.2157217  0.8858990    2.942144
8  {The Time Machine}   {American Beauty}   0.2157217  0.7151011    2.942144
ORE as framework for Model Building and Scoring

Workflow example

**Analysis**
- Data Preparation (filter, transform)
- Exploratory Data Analysis
- Sample data and split in train and test
- Build and test models in parallel with `ore.indexApply`
- Select best model and save in database `‘datastore’` object
- Load and test model from datastore for scoring new data

**Development**
- Code the build methodology in R script repository
- Code the scoring methodology in R script repository
- Invoke build and scoring R functions using `ore.*Apply`

**Production**
- Schedule build and score as nightly jobs for execution

...to be explored in more detail in Session 3 on Embedded R Execution
Oracle Advanced Analytics Option
Oracle Advanced Analytics Option

Fastest Way to Deliver Scalable Enterprise-wide Predictive Analytics

• Powerful
  – Combination of in-database predictive algorithms and open source R algorithms
  – Accessible via SQL, PL/SQL, R and database APIs
  – Scalable, parallel in-database execution of R language

• Easy to Use
  – Range of GUI and IDE options for business users to data scientists

• Enterprise-wide
  – Integrated feature of the Oracle Database available via SQL
    • R is integrated into SQL
  – Seamless support for enterprise analytical applications and BI environments
Oracle Advanced Analytics Value Proposition

Value Proposition
- Fastest path from data to insights
- Fastest analytical development
- Fastest in-database scoring engine on the planet
- Flexible deployment options for analytics
- Lowest TCO by eliminating data duplication
- Secure, Scalable and Manageable

Data remains in the Database
- Automated data preparation for select analytics
- Scalable distributed-parallel implementation of machine learning techniques in-database
- Scalable R leveraging database computational engine

Flexible interface options – R, SQL, IDE, GUI
- Fastest and most Flexible analytic deployment options
- Can import 3rd party models

Oracle Advanced Analytics

Traditional Analytics

Data Import

Model “Scoring”

Data Preparation and Transformation

Model Building

Data Preparation and Transformation

Data Extraction

Savings

Hours, Days or Weeks

Secs, Mins or Hours

Source Data

Dataset in Work Area

Analytic Processing

Process Output

Target
Customer Loyalty - Solution Summary

Managing customer loyalty is a key component of customer experience management in retail, telecommunications, and consumer markets. It starts with making use of the mountains of data available about each customer, their shopping patterns, and assessing long term value of each customer, funding effective marketing campaigns that target customers most likely to respond to offers, and determining that next best profitable action. Customer Loyalty management drives over 500B dollars in revenue worldwide.

- Build brand loyalty
- Accelerate predictive model build through deployment leveraging every customer interaction and transaction available to you
- Quickly identify profitable customers and create effective marketing campaigns
Lifetime Customer Loyalty with Oracle Advanced Analytics

Customer Problem
• It is expensive to acquire new customers or lose existing ones.
• Assessing long term value of existing customers and finding ways to retain and convert at-risk customers into profitable ones is a challenge.
• Improving customer loyalty is about that unique individual customer insight to be able to offer the right product/service at the right time. It’s the ability to predict what influences repeat shopping behavior at the right cost.
• Issues: 1) Very large data volumes, 2) Too many scenarios to model, 3) Operationalization of resulting models into production.

Power Positions
• Easily work with billions of transactions from points of sale, 10s of thousands of products and 100s of millions of customers in-place in Oracle Database where the data reside.
• Thousands of unique attributes about each consumer/household.
• Readily incorporate unstructured data such as social networks and review feedback into analysis.
• Lowest TCO and fastest path to enterprise-wide analytics deployment.

Unique Capabilities
• Powerful combination of in-database predictive algorithms and open source R algorithms.
• Range of GUI and IDE options for business users to data scientists.
• Rapid transition of models from development to operationalization.

Benefits
• Get started immediately with data in the database.
• Sub-second query response at very large data volumes to allow rapid data preparation.
• Scalable parallel distributed predictive algorithms.
• Range of interface options that facilitate business-IT collaboration.
• Leverages Enterprise-class infrastructure.
Typical volumetrics at retailer

• 3.2 Billion transactions
  – 120 million transactions bought a specific product
  – Understand co-occurrence of products across transactions to determine likelihood of 2 products bought together

• 19 million households
  – Segment households based on demographic data and purchase behavior
## Big Data Scenarios with Database Data

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyze 100 million households that carry loyalty card to find out what the most influential factors that drive purchase behavior of products in one group are</td>
<td>From start to model ready state: 25 minutes</td>
</tr>
<tr>
<td>Identify households that consumed a specific product from a 5 billion transactions data set Eliminate those households with an aggregated spend of less than x dollars Segment the remaining households into 30 groups What describes each segment and how does that relate to the business?</td>
<td>Start to finish: 5 minutes</td>
</tr>
<tr>
<td>What products tend to be bought together? Analyze 5 billion POS transactions to identify subsets of products bought together Use this as basis to identify next best offer for each of 100 million households in each product category</td>
<td>Start to Finish: 4 minutes</td>
</tr>
<tr>
<td>Analyze 150 million orders in the last month to build a fraud detection model</td>
<td>Start to Finish: 8 minutes</td>
</tr>
</tbody>
</table>
Summary

- R-to-SQL transparency improves user efficiency by allowing use of R directly against database data
- ORE enables R users to leverage in-database analytical techniques
- Open source R packages can be leveraged in combination with database-managed data and task parallel execution
- ORE provides a framework for sophisticated model building and data scoring
- R integration into the SQL language enables integration into IT software stack
- Oracle redistributes R and provides Enterprise support
Resources

• Blog:  https://blogs.oracle.com/R/

• Forum:  https://forums.oracle.com/forums/forum.jspa?forumID=1397

• Oracle R Distribution:  

• ROracle:  
  http://cran.r-project.org/web/packages/ROracle

• Oracle R Enterprise:  
  http://www.oracle.com/technetwork/database/options/advanced-analytics/r-enterprise

• Oracle R Connector for Hadoop:  