Safe Harbor Statement

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 remains at the sole discretion of Oracle.
Connect to the ORE HOL Instance
Oracle R Enterprise Cloud Deployment Architecture

Domain: biwa-nn.oraclepts.nl

Remote Desktop: oreuser1
Student 1

Remote Desktop: oreuser2
Student 2

Third-Party RStudio Server

Schemas rquser1 rquser2

PDB ORE1 PDB ORE2

PaaS
Oracle R Enterprise Cloud Deployment Architecture

- PaaS
- Third-Party RStudio Server
- Web browser
- Oracle Data Miner via SQL Developer
- oreuser1
- oreuser2

Oracle R Enterprise PDB ORE1 PDB ORE2

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |
Oracle R Enterprise Cloud Deployment Architecture

Oracle Database
OAA (ORE)
Oracle R Distribution (ORD)

R Script Repository
R Object Datastore

PDB OREn
MacOS users

• Install from the Apple App Store, not Microsoft website (old version)
• https://itunes.apple.com/us/app/microsoft-remote-desktop
Domains

*Use the domain from the signup sheet*

- biwa-\textit{nn}.oraclepts.nl
- Login as oreuser1
Connect to Remote Desktop

Student Environment

• Use the computer domain provided
• Login and Password
  – oreuser1 [or oreuser2]
  – Biwa2017
Double click on “RStudio Server” Firefox icon

[or]

Single click the firefox icon at the top.
Steps to connect to ORE HOL Instance and set up
*RStudio environment to access Oracle R Enterprise*

- Sign in with user ‘oreuser1’ [or] ‘oreuser2’ and password ‘Biwa2017’
- Click the open file icon
- Click the ‘ORE’ folder
- Select the file “Oracle Vignette – Variable Selection”
Steps to connect to ORE HOL Instance and set up SQL Developer for Oracle Data Miner

• Click SQL Developer icon at top

• Click File->Open, or this icon, click ‘ORE folder, and select the .sql file

• Go back to Firefox and the RStudio interface

• In RStudio, click or CTRL-Enter to run one line or selected set of lines
In SQL Developer

Go to ‘Connections’
right lick “OREn”
Select ‘Properties’
In SQL Developer

REMOVE all text after

i.e., .lab14.oraclecloud.internal
Back to RStudio

• Go back to Firefox and the RStudio interface
• In RStudio, click or CTRL-Enter to run one line or selected set of lines
Back to Firefox and R

Source

Console

Environment

History

Files, Plots

Packages, Help
Hands-on Lab Format and Content

• Follow along with instructor through first script, or go at your own pace
  – OAA Vignette – Variable Selection using Attribute Importance
  – OAA Vignette – Clustering
  – OAA Vignette – Regression
  – OREdplyr
  – Using R for Big Data AA and ML

• Click “run” on each line, or group of lines to execute

• Explore beyond the script if you’re comfortable with R

• Get online doc for any function you need help with ?functionName

Note: we will not cover all of these as a group
OAA Vignette – Variable Selection using Attribute Importance

• Learn the basics of interacting with R and ORE
• Create HOUSE dataset and table from a file
• Explore and prepare data using in-database execution from ORE Transparency Layer
• Visualize data using both overloaded functions and CRAN package, e.g., ggplot2
• Perform variable selection using in-database Attribute Importance function
• Use embedded R execution from R and SQL with ORE datastore and R script repository
• Attribute importance selection can be a step before classification or regression model building
OAA Vignette – Clustering

• Generate simple 2D data with 3 clusters in R and push to Oracle Database
• Build k-Means and O-Cluster models, assign clusters, and visualize results
• Use auto data set from ISLR package
• Build k-Means clustering model, assign clusters, and visualize results
• Explore clusters with a few statistics and 2D ggplot2 visualization
• Visualize clusters in 3D using plot3D
• Use Oracle Data Miner to build multiple models and visualize results
• Generate plots using ORE embedded R execution from both R and SQL
OAA Vignette – Regression

• Create HOUSE dataset and table from a file
• Explore and prepare data using in-database execution from ORE Transparency Layer
• Sample data into train and test sets using ORE row indexing
• Build a variety of models and score data using:
  – R: lm
  – ORE: ore.lm, ore.odmSVM, ore.odmGLM, ore.neural
• Use ORE embedded R execution to build one model per zipcode and store in datastore
Using OREdplyr

• Use the package nycflights13 and mtcars datasets and create database tables
• Explore basic operations of the overloaded dplyr functions in OREdplyr
  – These use the same API as dplyr, but accept ore.frame objects for in-database execution
  – Functions: select, rename, filter, arrange, distinct, mutate, transmute, summarise, slice, sample_n, tally
  – Stacking operations
  – Grouping with group_by
  – Chaining
• Contrast non-standard evaluation and standard evaluation
• Table joins
  – Functions: inner_join, left_join, right_join, full_join
Using R for Big Data AA and Machine Learning *(advanced - long)*

- Broader range of functionality of ORE
- Loading data and accessing across database schemas – granting access
- Accessing shared datastores
- Exploring data – statistics and visualization
- Preparing data – recode, bin, normalize, outlier treatment
- Sampling

- Model building and scoring
- Embedded R Execution – parallel building on partitioned data
- Viewing models in Oracle Data Miner
- In-database scoring using R models
- Solution deployment using embedded R execution with the R and SQL interfaces
- Sharing R scripts
ORE Introduction
Analytic Pain Points

• It takes too long to get my data or to get the ‘right’ data
• I can’t analyze or mine all of my data – it has to be sampled
• Putting analytics/predictive models and results into production is ad hoc and complex
• Recoding R or other models into SQL, C, or Java takes time and is error prone
• Our company is concerned about data security, backup and recovery
• We need to build 10s of thousands of models fast to meet business objectives

See the blog series at
https://blogs.oracle.com/R/entry/addressing_analytic_pain_points
Scaling R to Big Data

Immediate access to database and Hadoop data from R

• Eliminate need to request extracts from IT/DBA
• Process data where they reside – minimize or eliminate data movement – through data.frame proxies

Scalability and Performance

• Use parallel, distributed algorithms that scale to big data on Oracle Database
• Leverage powerful engineered systems to build models on billions of rows of data or millions of models in parallel from R

Ease of deployment

• Using Oracle Database, place R scripts immediately in production (no need to recode) via SQL
• Use production quality infrastructure without custom plumbing or extra complexity

Process support

• Maintain and ensure data security, backup, and recovery using existing processes
• Store, access, manage, and track analytics objects (models, scripts, workflows, data) in Oracle Database
Oracle R Enterprise
Part of Oracle Advanced Analytics option to Oracle Database

• Use Oracle Database as HPC environment
• Use in-database parallel and distributed machine learning algorithms
• Manage R scripts and R objects in Oracle Database
• Integrate R results into applications and dashboards via SQL
Oracle R Enterprise

Part of Oracle Advanced Analytics option to Oracle Database

- **Transparency layer**
  - Leverage proxy objects (ore.frames) - data remains in the database
  - Overload R functions that translate functionality to SQL
  - Use standard R syntax to manipulate database data

- **Parallel, distributed algorithms**
  - Scalability and performance
  - Exposes in-database algorithms from ODM
  - Additional R-based algorithms executing and database server

- **Embedded R execution**
  - Manage and invoke R scripts in Oracle Database
  - Data-parallel, task-parallel, and non-parallel execution
  - Use open source CRAN packages
### Predictive Analytics algorithms in-Database

#### Classification
- Decision Tree
- Logistic Regression
- Naïve Bayes
- Support Vector Machine
- RandomForest

#### Regression
- Linear Model
- Generalized Linear Model
- Multi-Layer Neural Networks
- Stepwise Linear Regression
- Support Vector Machine

#### Clustering
- Hierarchical k-Means
- Orthogonal Partitioning
- Expectation Maximization*

#### Attribute Importance
- Minimum Description Length
- Expectation Maximization*

#### Anomaly Detection
- 1 Class Support Vector Machine

#### Market Basket Analysis
- Apriori – Association Rules

#### Feature Extraction
- Nonnegative Matrix Factorization
- Principal Component Analysis
- Singular Value Decomposition
- Explicit Semantic Analysis*

#### Time Series
- Single Exponential Smoothing
- Double Exponential Smoothing

*New in ORE 1.5.1
**ODB 12.2 only**

---

*plus open source R packages for algorithms in combination with embedded R data- and task-parallel execution*
Proxy Object – ore.frame

• Inherits from data.frame
• Overloaded R functions translate functionality to SQL
• No data movement

```
> str(ONTIME_S)
'data.frame': 219032 obs. of 27 variables:
  ..@ dataQry : Named chr 
199 ' (select /**+ no_merge(t) */ "X" VAL001, "YEAR" VAL002, "MONTH" VAL003, "MONTH2" VAL004, "DAYOFMONTH" VAL05, "DAYOFMONTH" | __truncated__ 
  ..@ dataObj : chr "384_3"  
  ..@ desc : chr 'data.frame': 27 obs. of 2 variables:  
  ..$ name : chr "X" "YEAR" "MONTH" "MONTH2"  
  ..$ sclass : chr "numeric" "numeric" "numeric" "factor"  
  ..@ sqlName : chr  
  ..@ sqlValue : chr "" "" "" "" "" "" "" ""  
  ..@ sqlTable : chr 
  "RQUSER." "ONTIME_S"  
  ..@ sqlPred : chr ""  
  ..@ extRef : list()  
  ..@ names : chr  
  ..@ row.names : int  
  ..@ S3class : chr "data.frame"
```
Scalability through proxies with function overloading
In-database aggregation – no data movement

Oracle Distribution of R version 3.3.0 (-- "Supposedly Educational"

```r
aggdata <- aggregate(ONTIME_S$DEST,
  by = list(ONTIME_S$DEST),
  FUN = length)
```

```r
class(aggdata)
[1] "ore.frame"
attr(,"package")
[1] "OREbase"

head(aggdata)
```

<table>
<thead>
<tr>
<th>Group.1</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABE</td>
<td>237</td>
</tr>
<tr>
<td>ABI</td>
<td>34</td>
</tr>
<tr>
<td>ABQ</td>
<td>1357</td>
</tr>
<tr>
<td>ABY</td>
<td>10</td>
</tr>
<tr>
<td>ACK</td>
<td>3</td>
</tr>
<tr>
<td>ACT</td>
<td>33</td>
</tr>
</tbody>
</table>

Oracle SQL

```sql
select DEST, count(*)
from ONTIME_S
group by DEST
```
Scalable Machine Learning Algorithms
ORE parallel distributed model (e.g., Linear Regression) using embedded R engines

Oracle Distribution of R version 3.3.0 (-- "Supposedly Educational"

```r
> options(ore.parallel=4)
> lm_mod <- ore.lm(ARRDELAY ~ DISTANCE + DEPDELAY,
                    data=ONTIME_S)

> summary(lm_mod)
Call:
ore.lm(formula = ARRDELAY ~ DISTANCE + DEPDELAY, data = ONTIME_S)
Residuals:
     Min       1Q   Median       3Q      Max
-1462.45   -6.97  -1.36     5.07   925.08
Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            2.254e-01  5.197e-02   4.336  1.45e-05 ***
DISTANCE               -1.218e-03  5.803e-05  -20.979  < 2e-16 ***
DEPDELAY                9.625e-01  1.151e-03   836.289  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 14.73 on 215144 degrees of freedom
(4785 observations deleted due to missingness)
Multiple R-squared: 0.7647, Adjusted R-squared: 0.7647
F-statistic: 3.497e+05 on 2 and 215144 DF, p-value: < 2.2e-16
```

Copyright © 2017 Oracle and/or its affiliates. All rights reserved.
Linear Model Performance Comparison

- Predict “Total Revenue” of a customer based on 31 numeric variables as predictors, on 184 million records using SPARC T5-8, 4TB of RAM
- Data in an Oracle Database table

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Threads Used*</th>
<th>Memory required**</th>
<th>Time for Data Loading***</th>
<th>Time for Computation</th>
<th>Total</th>
<th>Relative Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open-Source R Linear Model (lm)</td>
<td>1</td>
<td>220Gb</td>
<td>1h3min</td>
<td>43min</td>
<td>1h46min</td>
<td>1x</td>
</tr>
<tr>
<td>Oracle R Enterprise lm (ore.lm)</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>42.8min</td>
<td>42.8min</td>
<td>2.47X</td>
</tr>
<tr>
<td>Oracle R Enterprise lm (ore.lm)</td>
<td>32</td>
<td>-</td>
<td>-</td>
<td>1min34s</td>
<td>1min34s</td>
<td>67.7X</td>
</tr>
<tr>
<td>Oracle R Enterprise lm (ore.lm)</td>
<td>64</td>
<td>-</td>
<td>-</td>
<td>57.97s</td>
<td>57.97s</td>
<td>110X</td>
</tr>
<tr>
<td>Oracle R Enterprise lm (ore.lm)</td>
<td>128</td>
<td>-</td>
<td>-</td>
<td>41.69s</td>
<td>41.69s</td>
<td>153X</td>
</tr>
</tbody>
</table>

*Open-source R lm() is single threaded
**Data moved into the R Session's memory, since open-source lm() requires all data to be in-memory
***How long it takes to load 40Gb of raw data into the open-source R Session's memory
IoT Use Case: Sensor Data Analysis

Massive Predictive Modeling

• Model each customer’s behavior and identify deviations in individual behavior and overall aggregate demand

• 200 thousand households, each with a utility “smart meter”

• 1 reading / meter / hr

• 200K x 8760 hrs / yr ➞ 1.752B readings

• 3 years worth of data ➞ 5.256B readings

• Each customer has 26280 readings

• If each model takes 10 seconds to build, ➞ 555.6 hrs (23.2 days)

...with 128 DOP ➞ 4.3 hrs
Scalable Sensor Data Analysis – Model Building

Smart meter scenario

f(dat, args, ...) {
  R Script
  build model
}

Oracle Database

Data

R Datastore

R Script Repository

f(dat, args, ...)

f(dat, args, ...)

f(dat, args, ...)

f(dat, args, ...)

R

Model c1

Model c2

Model ci

Model cn
Build models and store in database, partition on CUST_ID

```r
ore.groupApply (CUST_USAGE_DATA,
    CUST_USAGE_DATA$CUST_ID,
    function(dat, ds.name) {
      cust_id <- dat$CUST_ID[1]
      mod <- lm(Consumption ~ . -CUST_ID, dat)
      mod/effects <- mod$residuals <- mod$fitted.values <- NULL
      name <- paste("mod", cust_id,sep="")
      assign(name, mod)
      ds.name1 <- paste(ds.name,".",cust_id,sep="")
      ore.save(list=paste("mod",cust_id,sep=""), name=ds.name1, overwrite=TRUE)
      TRUE
    },
    ds.name="myDatastore", ore.connect=TRUE, parallel=TRUE
)
```
Production Deployment of R through SQL

• Load R function into Oracle Database from R or SQL

• From SQL
  – Return images as PNG BLOB column
  – Return data.frame content as database table
  – Return XML with image and data.frame content

• Invoke same function from R
Have fun and raise your hand if you need help
Learn More about Oracle’s Advanced Analytics R Technologies...

http://oracle.com/goto/R

R Technologies from Oracle
Bringing the Power of R to the Enterprise