Scaling R to New Heights with Oracle Database Hands-On Lab

Mark Hornick
Oracle Advanced Analytics

Tim Vlamis
Vlamis Software Solutions

Alex Ardel
Oracle Advanced Analytics

January 27, 2016

BIWA SUMMIT 2016
The Oracle Big Data + Analytics User Conference
January 26–28, 2016
Oracle Spatial Summit + YesSQL Summit

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.

ORACLE

HOL – topics

- **Accessing data**
  - Data in a flat file (CSV)
  - Data in database tables
  - Dynamic query for ore.frame
  - Datastores
- **Exploring data**
  - Statistics
  - Visualizations
- **Preparing the data**
  - Recode, Binning, Normalize
  - Outlier treatment
  - Sampling
- **Model building and Scoring**
  - Attribute Importance
  - SVM
  - randomForest
  - Many models (ore.groupApply multi-column)
- **Model viewing in Oracle Data Miner**
- **Scoring using R models**
- **Solution deployment**
  - In-database scoring using embedded R
  - Table, Image, XML
  - Using SQL to invoke R scripts
  - Sharing R Scripts

ORACLE

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

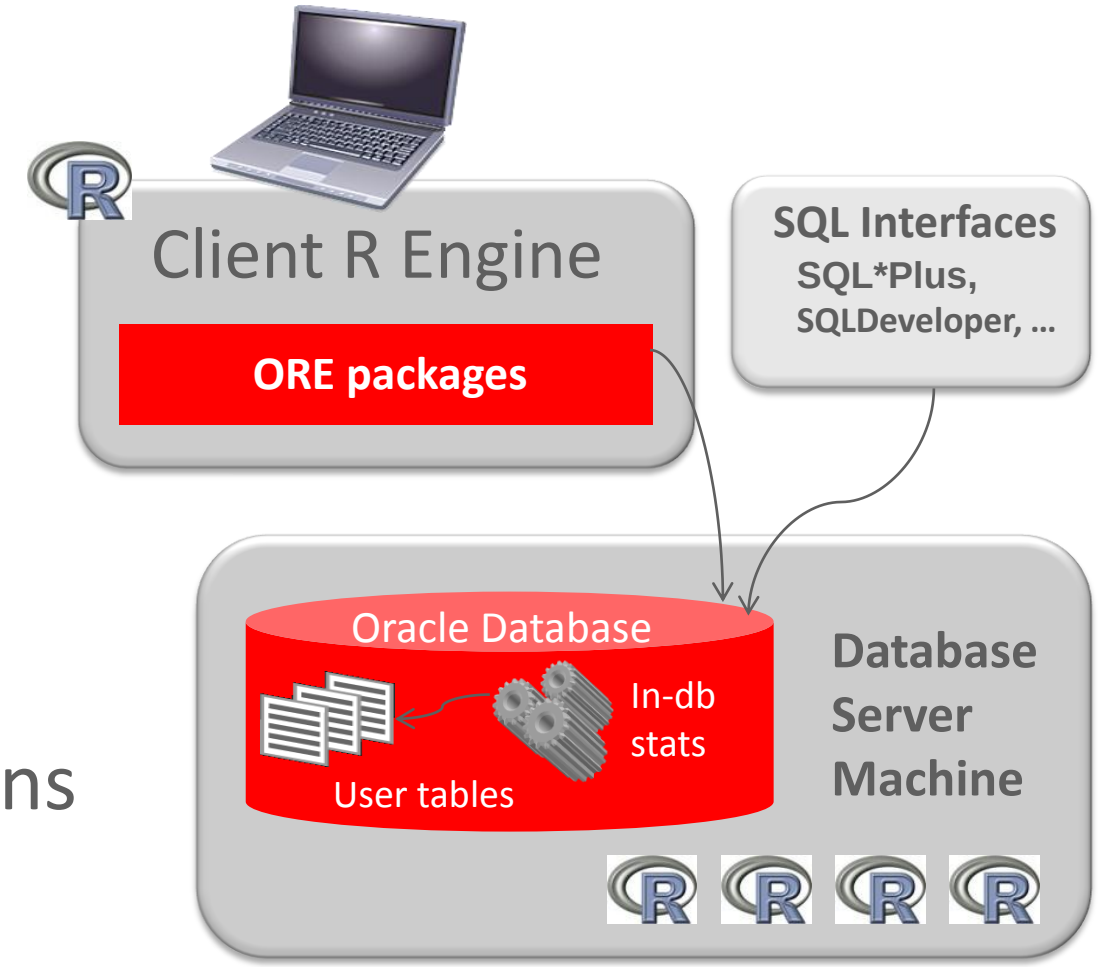
- Maintain and ensure data security, backup, and recovery using existing processes
- Store, access, manage, and track analytics objects (models, scripts, workflows, data) in Oracle Database

ORACLE

Scaling R to Big Data

***Oracle R Enterprise**, part of **Oracle Advanced Analytics** option*

- 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



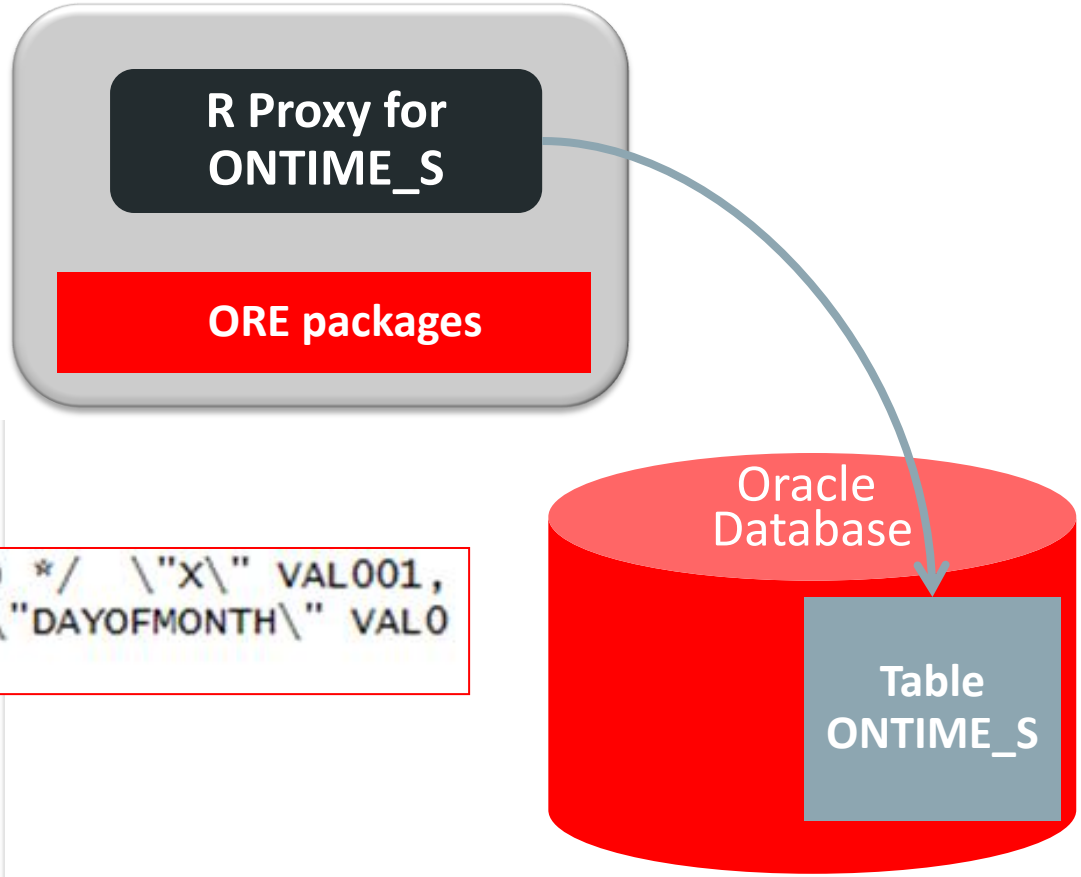
Oracle R Enterprise – *three key areas*

- Transparency Layer
  - Standard R syntax to manipulate data in-database via proxy objects (ore.frame)
  - Overloads R functions and translates functionality to SQL
  - Rich set of transformations, statistical functions, visualizations
- Predictive Analytics
  - In-database ODM algorithms exposed via R interface with auto data preparation
  - ORE-specific parallel, distributed algorithms
- Embedded R Execution
  - Invoke user-defined R functions from R and SQL at database server machine
  - Use CRAN packages in user-defined R functions
  - Execute in data-parallel and task-parallel manner to leverage powerful machines like Exadata
  - Return structured data, images, XML via SQL invocation for easy integration with applications
  - R Script Repository and Datastore as R object repository

Proxy Object – ore.frame

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

```
> str(ONTIME_S)
'data.frame':   219932 obs. of  27 variables:
Formal class 'ore.frame' [package "OREbase"] with 12 slots
  ..@ .Data    : list()
  ..@ dataQry  : Named chr "( select /*+ no_merge(t) */  \"X\" VAL001,
\"YEAR\" VAL002,\"MONTH\" VAL003,\"MONTH2\" VAL004,\"DAYOFMONTH\" VAL0
05,\"DAYOFMONTH"| __truncated__
  ..@ dataObj  : chr "384_3"
  ..@ desc     :'data.frame':   27 obs. of  2 variables:
  .. ..$ name  : chr  "X" "YEAR" "MONTH" "MONTH2" ...
  .. ..$ Sclass: chr  "numeric" "numeric" "numeric" "factor" ...
  ..@ sqlName  : chr
  ..@ sqlValue : chr  "\"X\"" "\"YEAR\"" "\"MONTH\"" "\"MONTH2\"" ...
  ..@ sqlTable : chr "\"RQUSER\".\"ONTIME_S\""
  ..@ sqlPred  : chr ""
  ..@ extRef   : list()
  ..@ names    : chr
  ..@ row.names: int
  ..@ .S3Class : chr "data.frame"
```



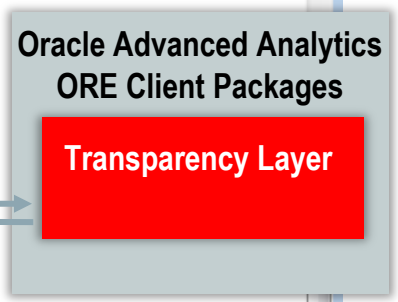
Scalability through proxies with function overloading

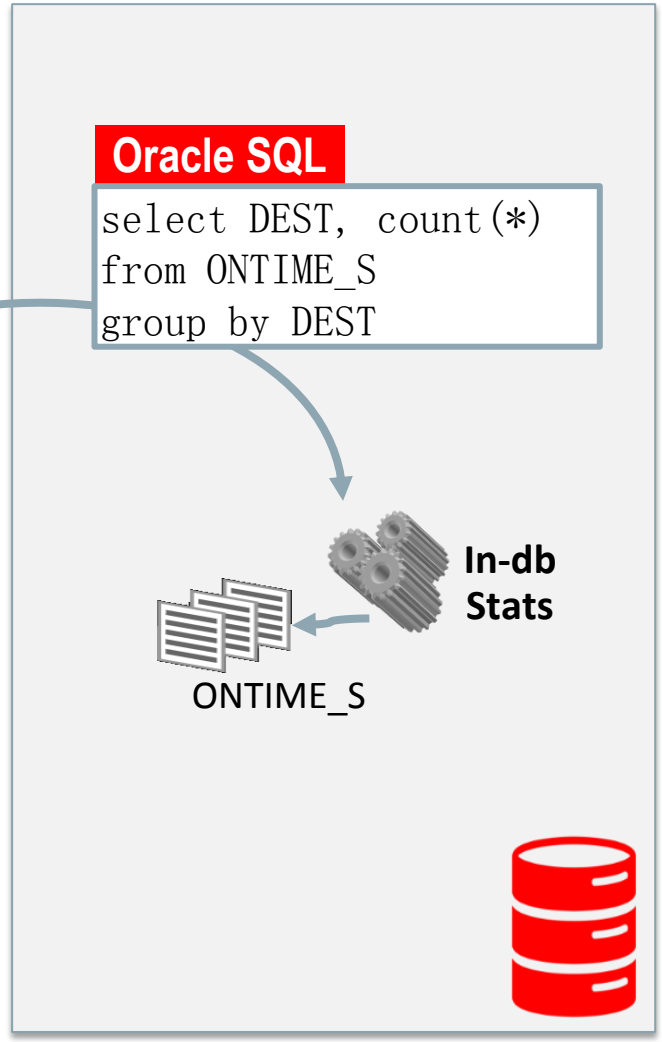
In-database aggregation – no data movement

R Console

Oracle Distribution of R version 3.2.0 (--) -- "Full of Ingredients"

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

> class(aggdata)
[1] "ore.frame"
attr(,"package")
[1] "OREbase"
> head(aggdata)
   Group.1    x
1  ABE      237
2  ABI       34
3  ABQ     1357
4  ABY       10
5  ACK        3
6  ACT       33
```



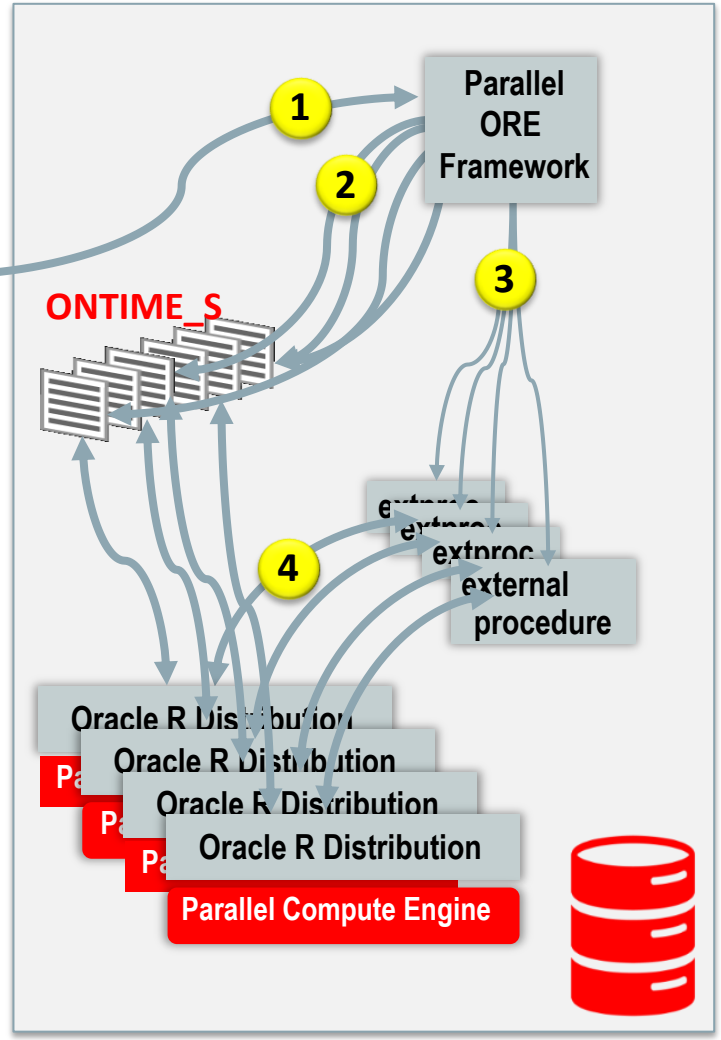


Scalable Machine Learning Algorithms

ORE parallel distributed model (e.g., Linear Regression) using embedded R engines

R Console

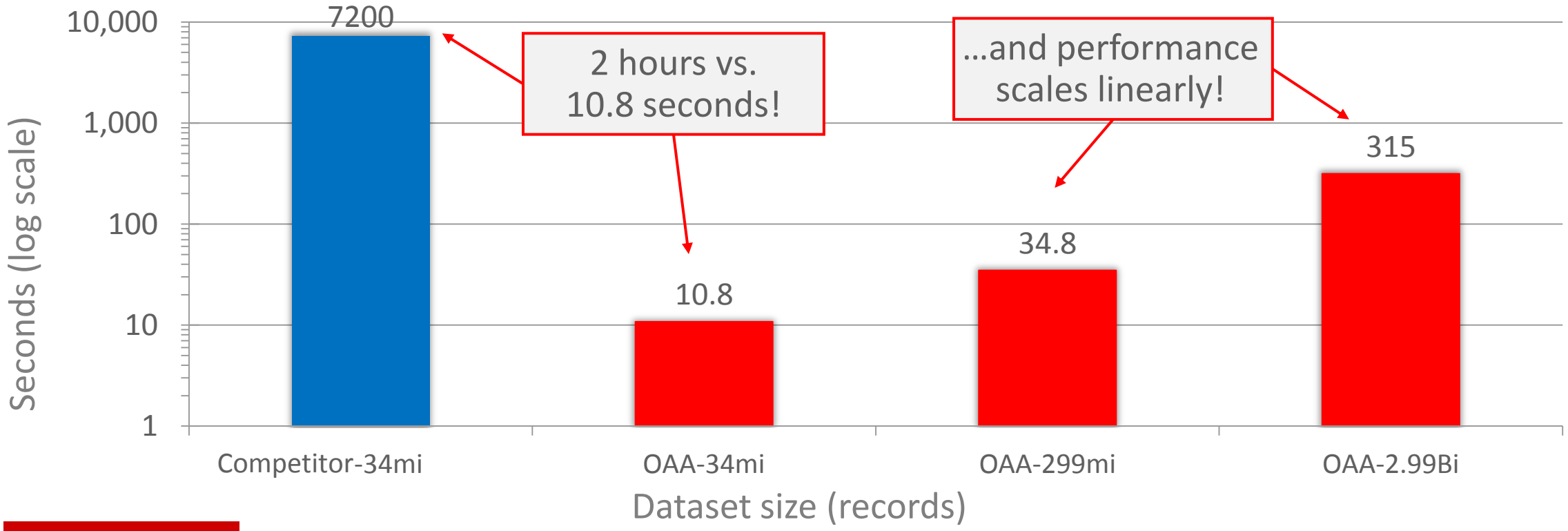
```
Oracle Distribution of R version 3.2.0  (--) -- "Full of Ingredients"

> 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 ***
---

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
```



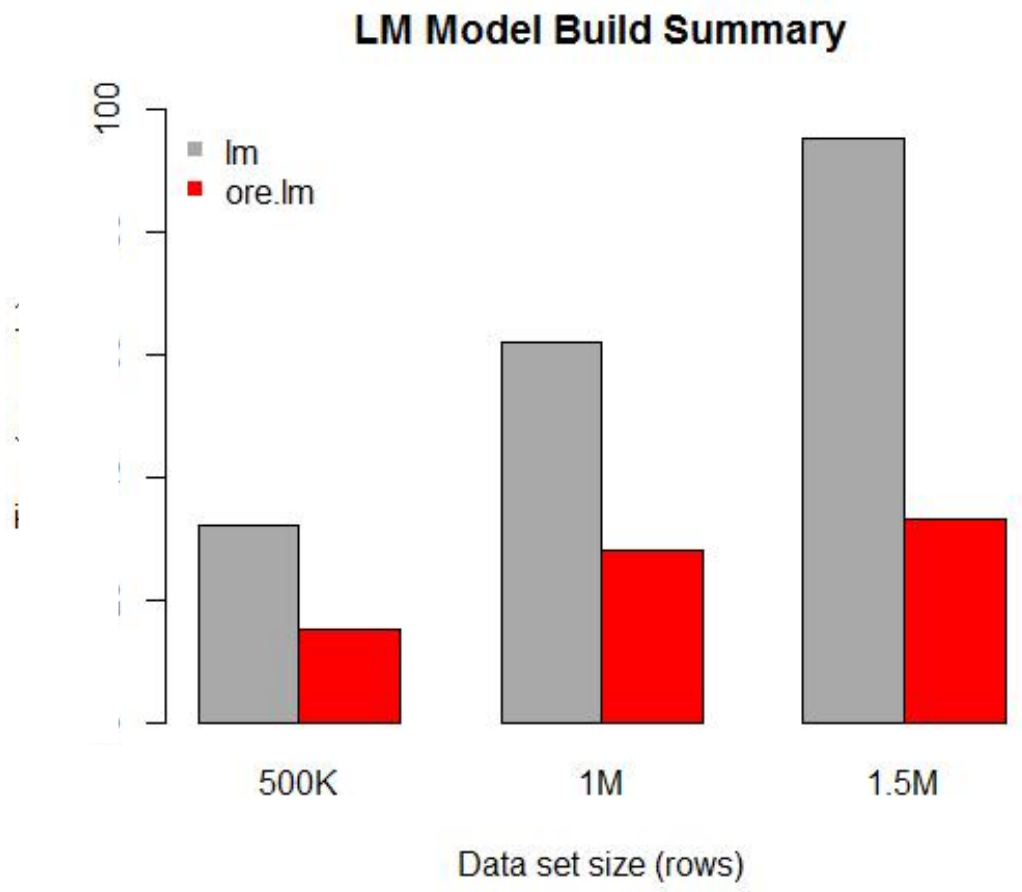
Oracle Advanced Analytics on Exadata X3-2 Half-Rack

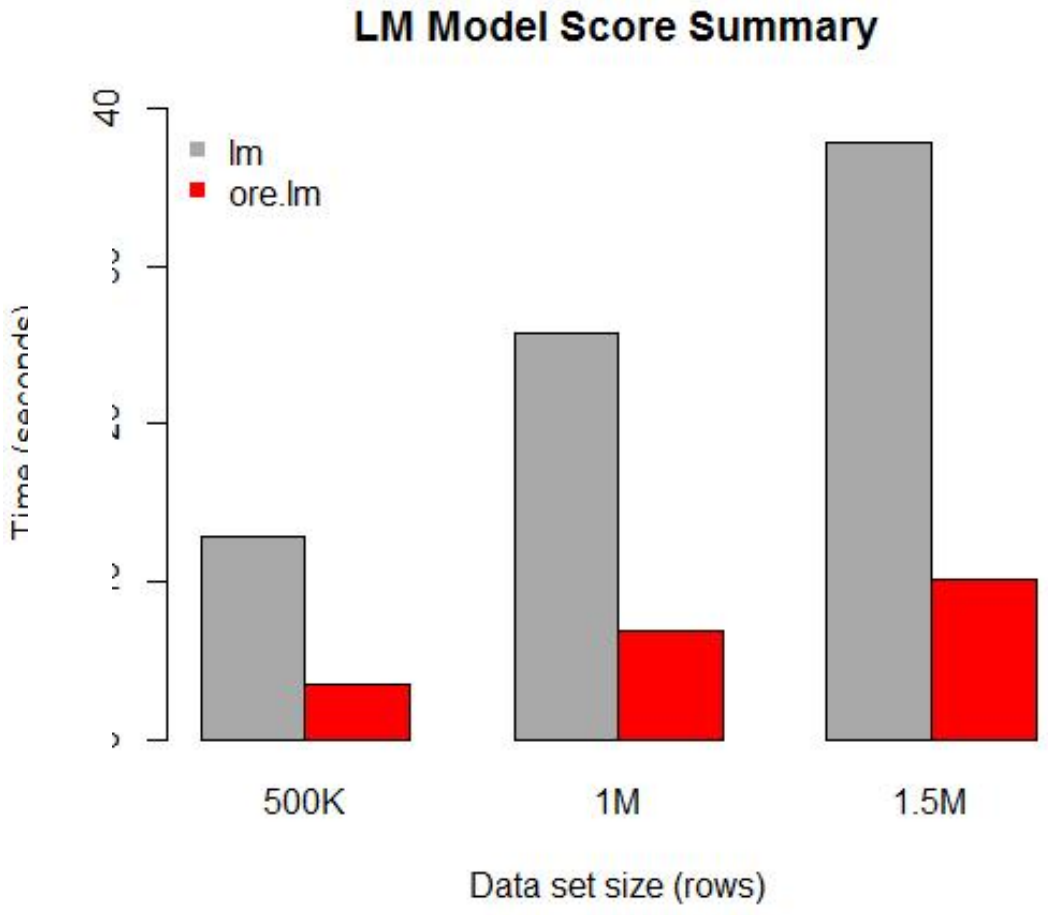
Scalability of distributed ore.lm() Linear Regression

Regression Model using 30 numeric variables: Competitor running on a Server connected via network to the same Exadata box took two hours for ETL+Model build on 34mi records.

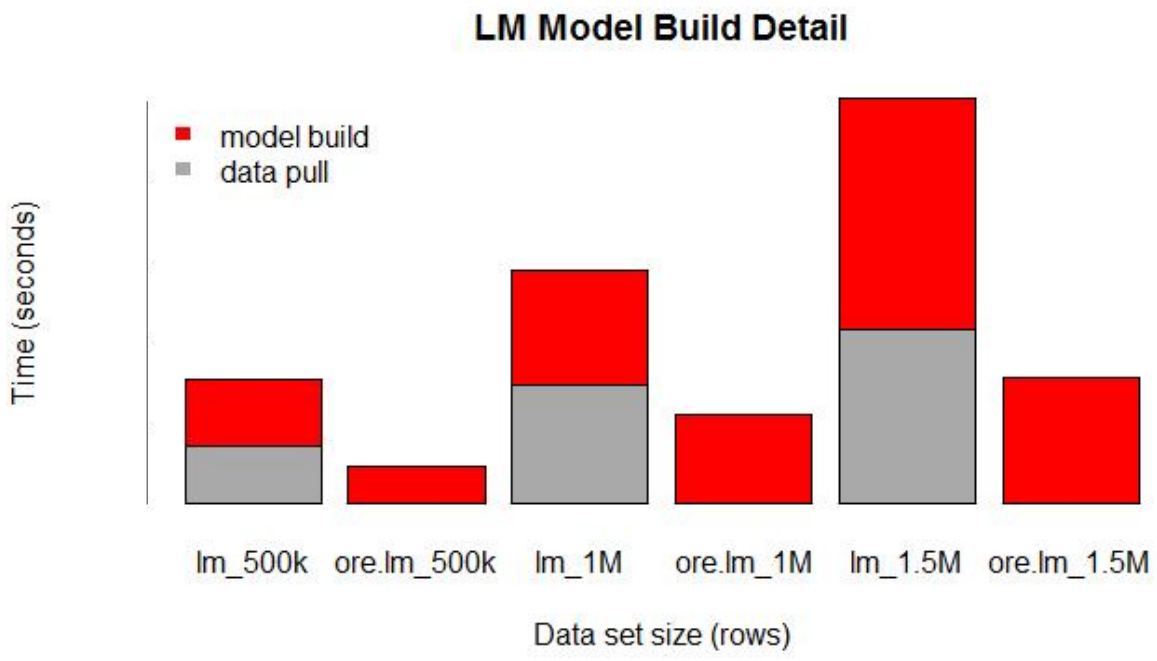


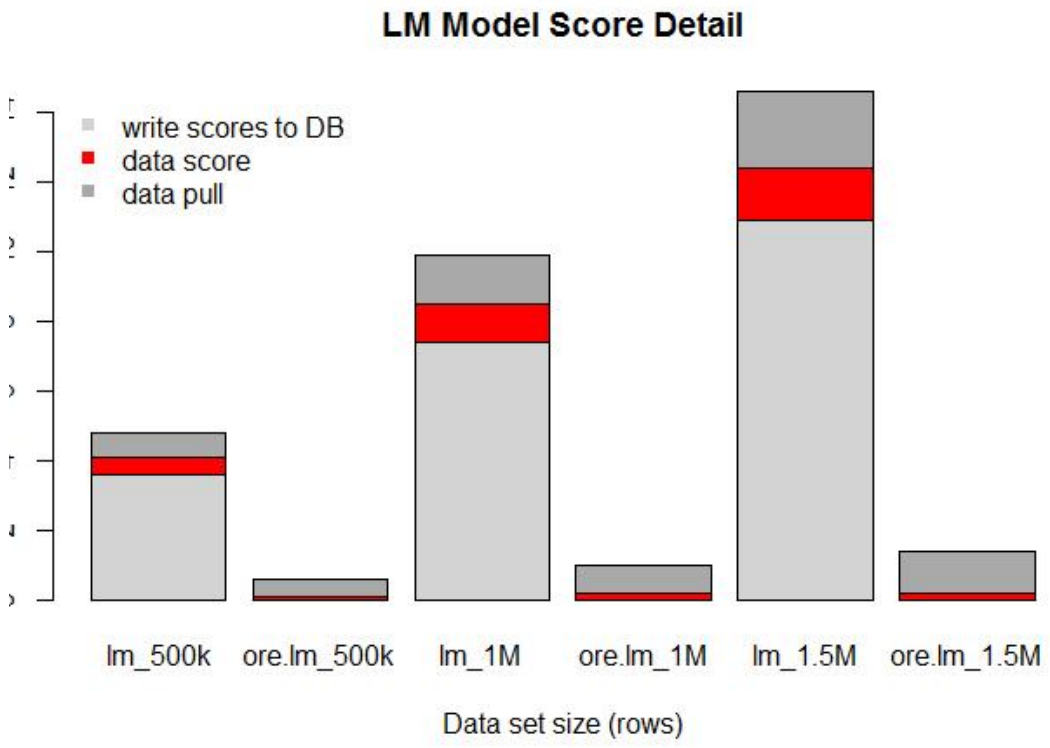
In-database Performance Advantage





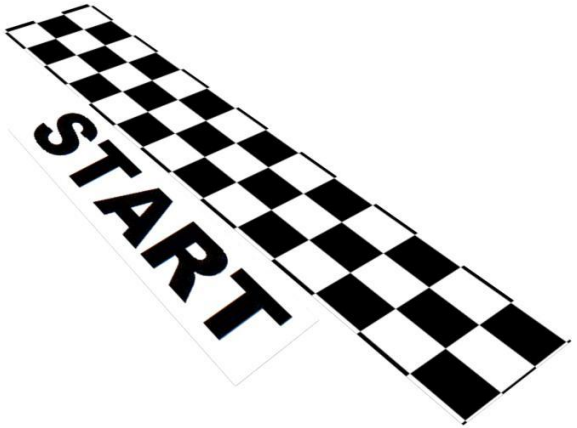
In-database Performance Advantage





Production Deployment of R through SQL

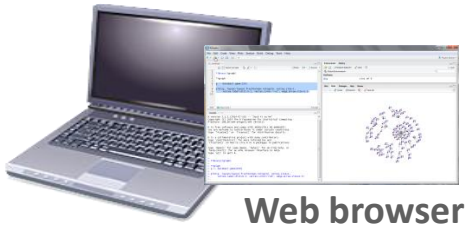
- Load R function into Oracle Database from R or SQL
- From SQL
  - Return images as PNG BLOB column
  - Return data.frame content as database table
  - Return XML with image and data.frame content
- Invoke same function from R



Connect to the ORE HOL Instance

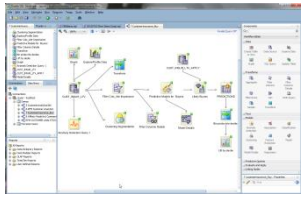
Oracle R Enterprise Cloud Deployment Architecture

Instructor Demonstration Environment

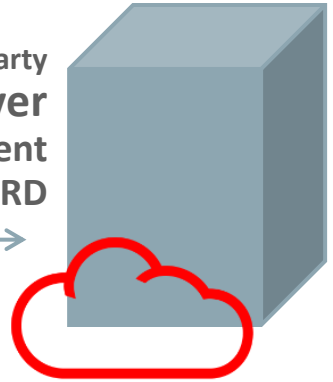


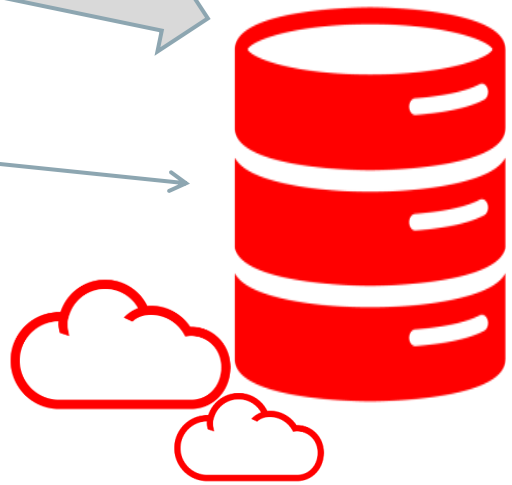
Web browser

Third-Party
RStudio Server
with ORE Client
Open Source R *or* ORD



Oracle Data Miner
via SQL Developer





R Script Repository

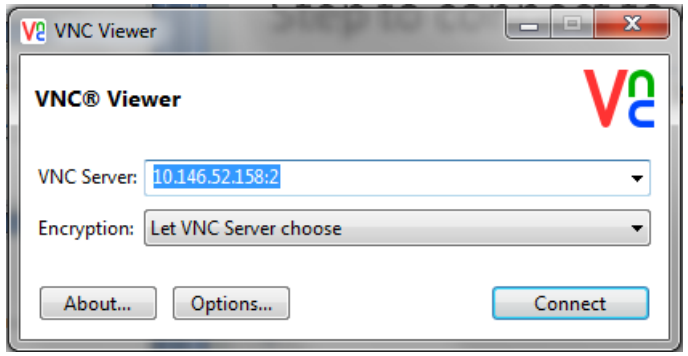
R Object Datastore

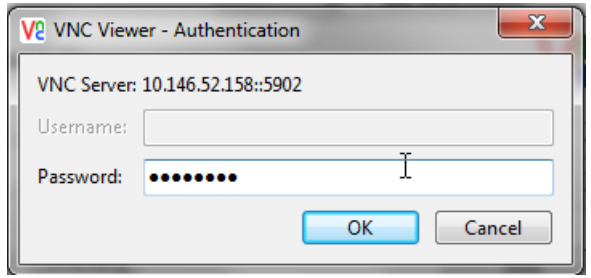
Oracle Database
OAA (ORE)
Oracle R Distribution (ORD)

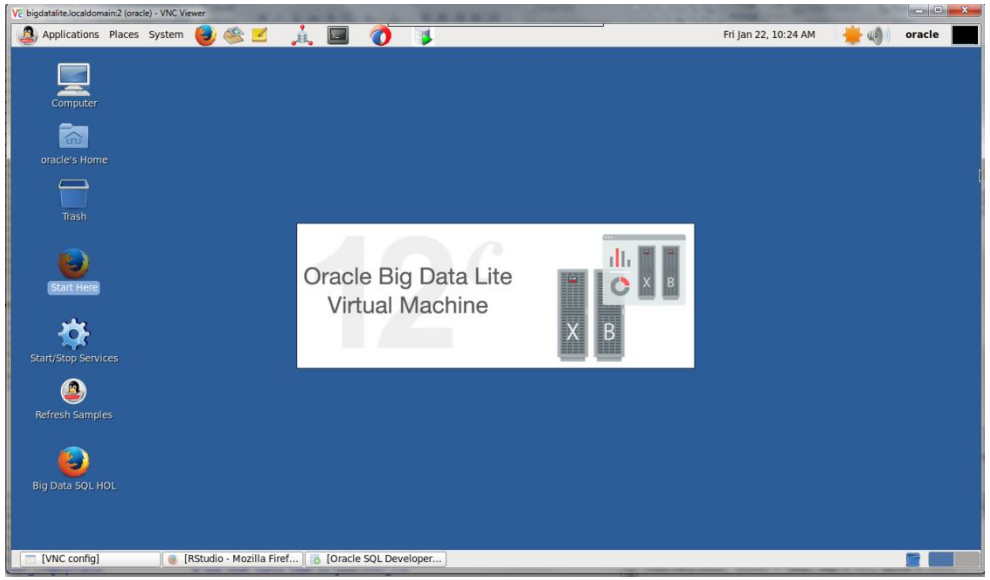
Steps to connect to ORE HOL Instance and set up

Student Environment

- Using VNC, use the IP address provided with port :2, password welcome1
  - Example:



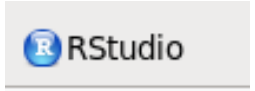




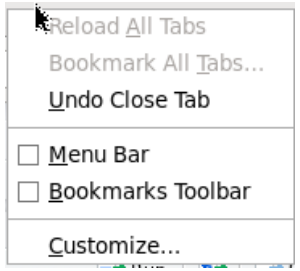
Steps to connect to ORE HOL Instance and set up

Student Environment

- Double click “Start Here” Firefox icon 

- In a new tab, click RStudio 

and sign in 

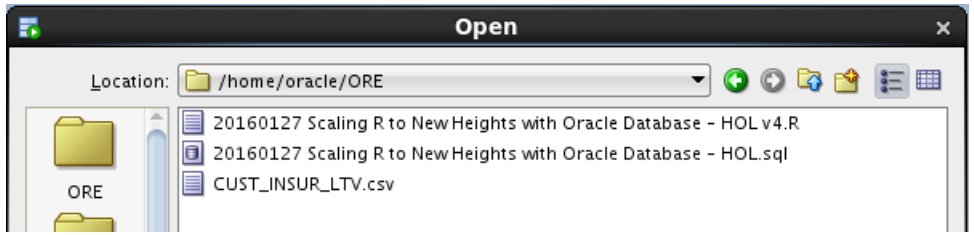
- Right click tab bar to turn off menu bar and bookmarks toolbar 

Steps to connect to ORE HOL Instance and set up

Student Environment

- Click SQL Developer icon  at top of screen

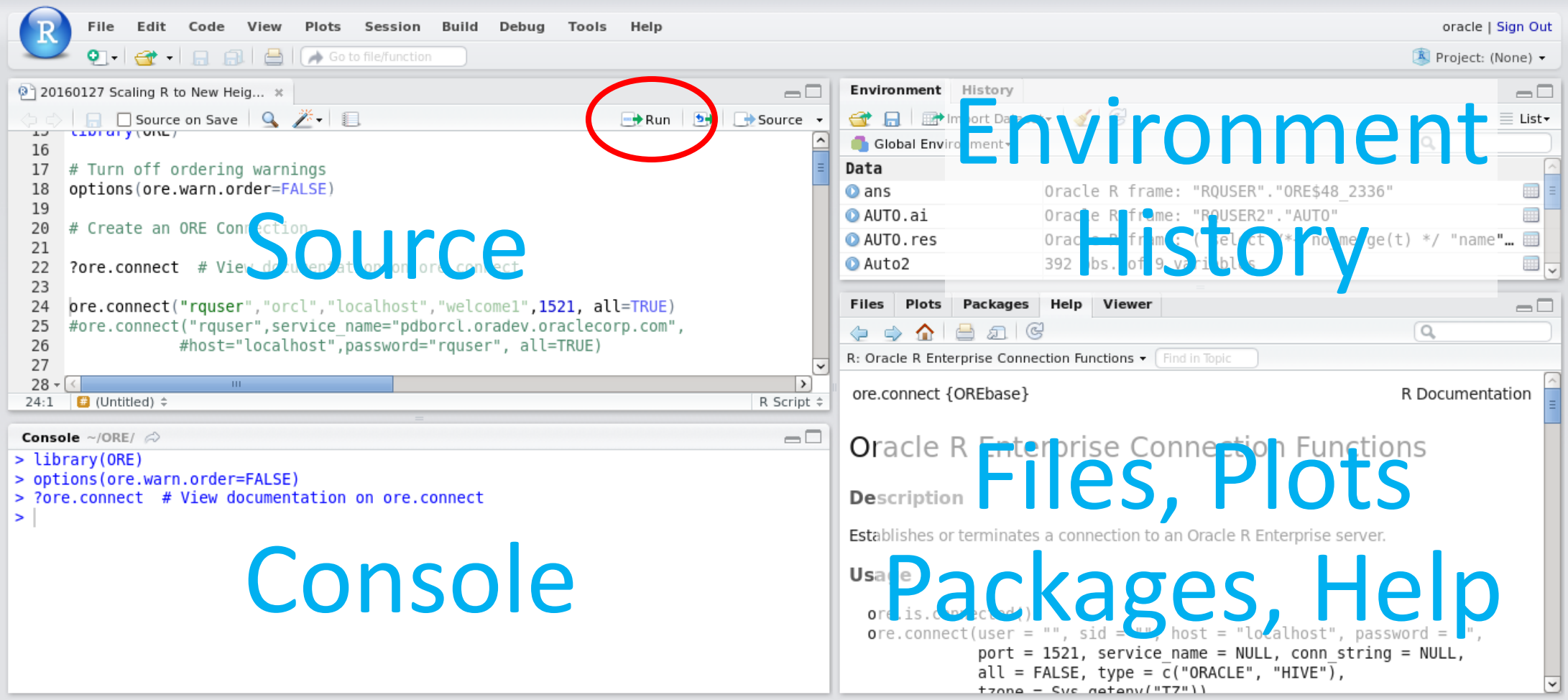
- Click File->Open, or this icon  and select the .sql file



- Go back to the browser and the RStudio interface
- In RStudio, click 

or CTRL-Enter to run one line or selected set of lines

RStudio Interface



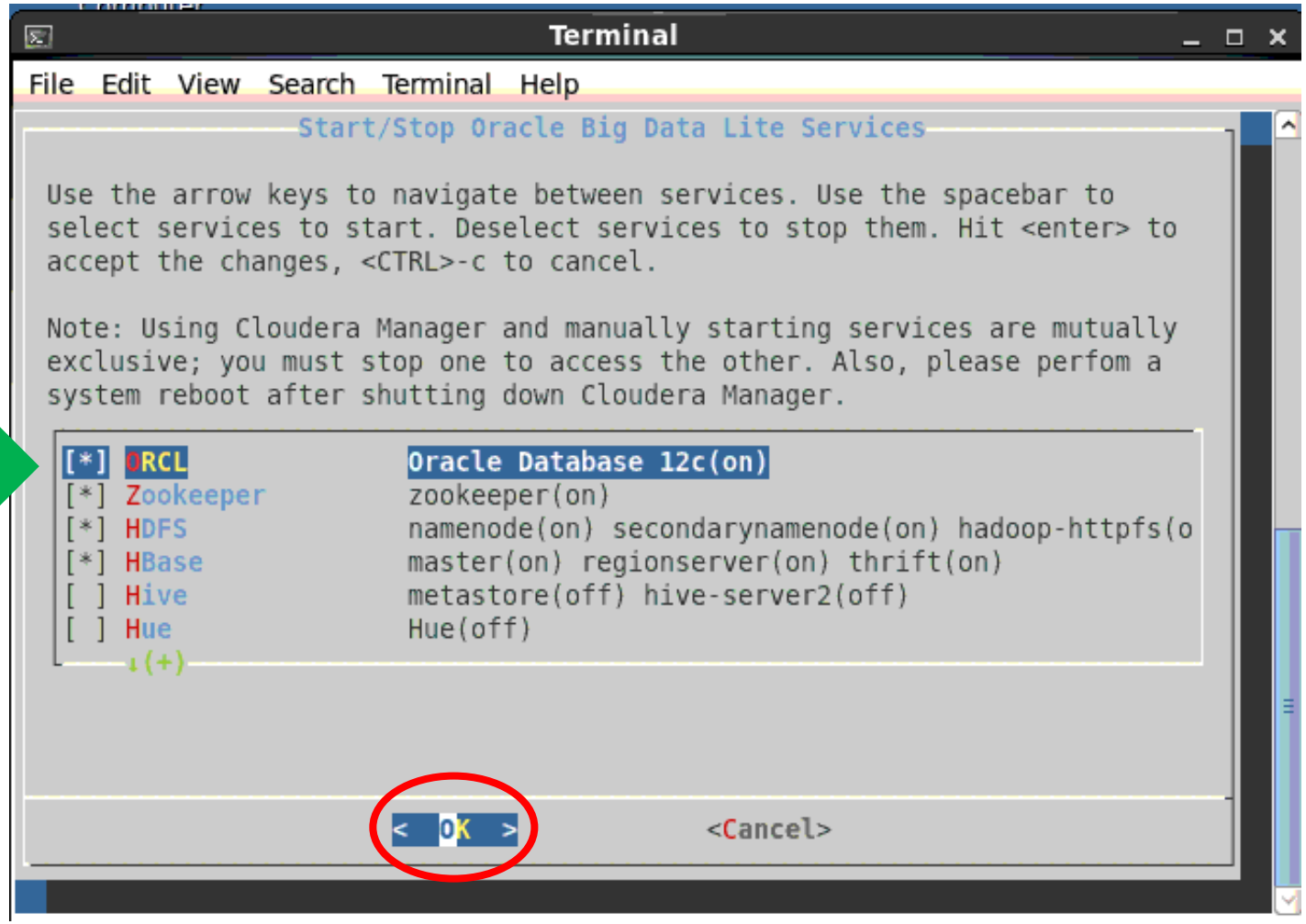
Create an ORE connection

Load ORE library and connect to Oracle Database

```
library(ORE)
?ore.connect
ore.connect("rquser",
            service_name="pdborcl.oradev.oraclecorp.com",
            host="localhost",
            password="rquser",
            all=TRUE)
# If error, start up the database on desktop
```

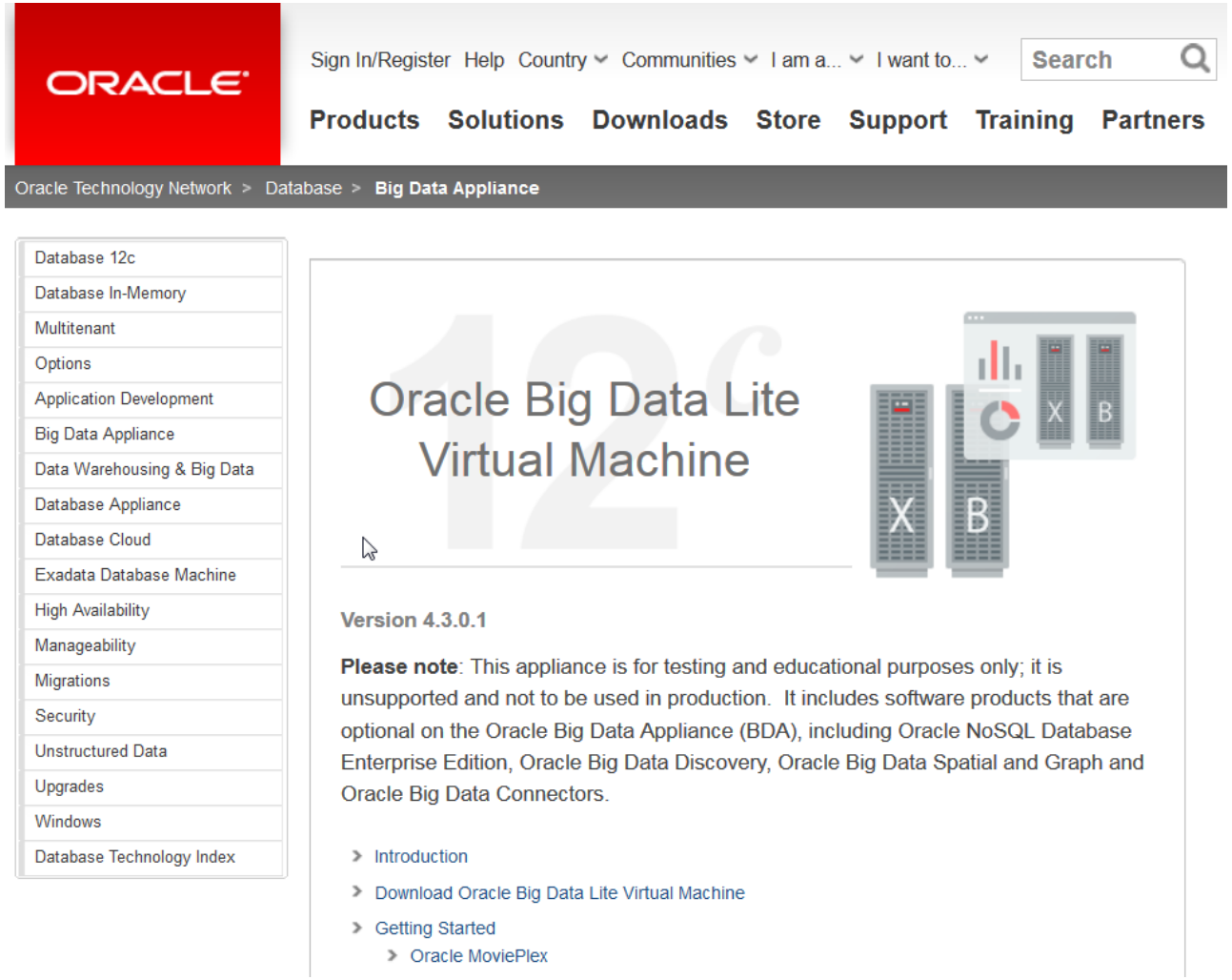
Start up Oracle Database 12c

Click to get *



You are using…
Big Data Lite 4.3.0

- Oracle Big Data Lite Virtual Machine provides an integrated environment to help you get started with the Oracle Big Data platform
- Many Oracle Big Data platform components have been installed and configured - allowing you to begin using the system right away
- http://www.oracle.com/technetwork/database/bigdata-appliance/oracle-bigdatalite-2104726.html
- You can download this for your own use

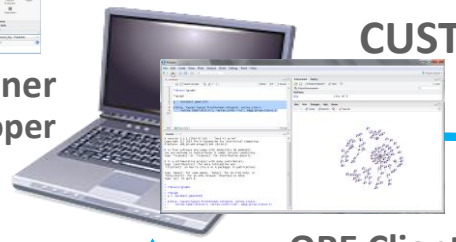


Accessing Data

Loading Data From a Flat File
Accessing Database Tables
Create ore.frame from query

Accessing Data

Oracle Data Miner
via SQL Developer



ORE Client

COLLEGE
AUTO
CUST_LTV_HIGH
DAT

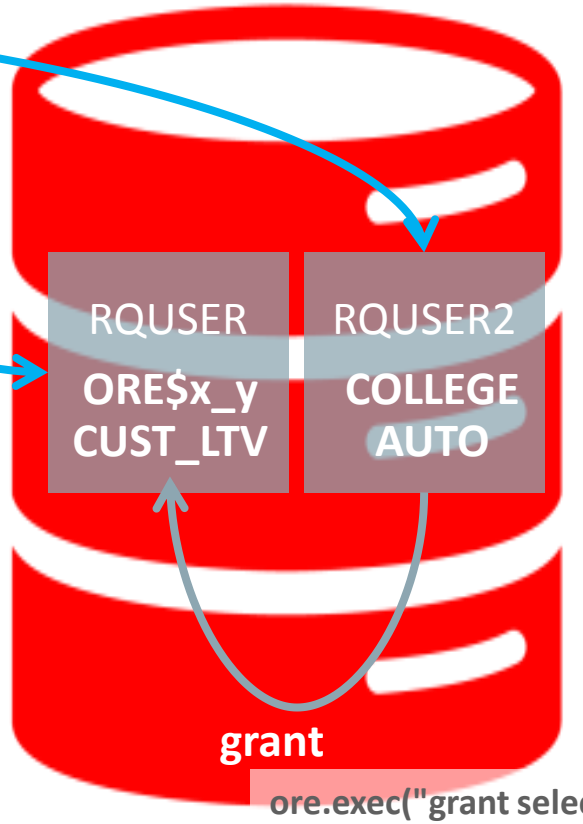
ore.create(...)

ore.sync(table=c("COLLEGE","AUTO"),
schema="RQUSER2")
ore.create(dat,table="CUST_LTV")

DAT <- ore.push(dat)
ore.attach(schema="RQUSER2")

dat <- read.csv(...)

CUST_INSUR_LTV.csv



grant

ore.exec("grant select on college to rquser")
ore.exec("grant select on auto to rquser")

Acessing Shared Datastores

Datastore – granting access

Client 1

USArrests df

ore.save(USArrests, name="ds_3", grantable=TRUE)

mtcars df

ore.save(mtcars, name="ds_2", grantable=TRUE)

iris df

ore.save(iris, name="ds_1")

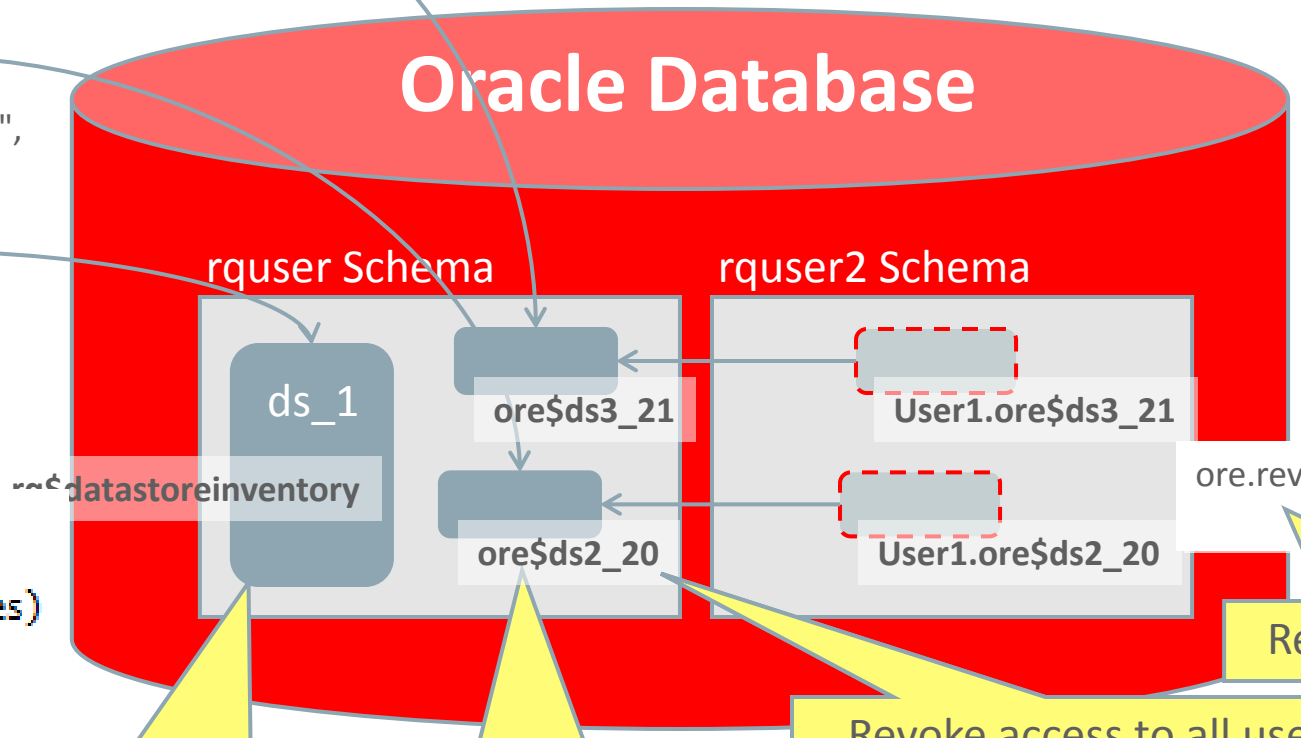
ore.datastore(type="grant")

```
> ore.datastore(type="grant")
[1] datastore.name grantee
<0 rows> (or 0-length row.names)
```

Client 2

ore.datastore(type="all")

```
> ore.datastore(type="all")[,1:3]
  owner datastore.name object.count
1 RQUSER           ds_2            1
2 RQUSER           ds_3            1
```



ore.revoke(name="ds_3", type="datastore", user="rquser2")

Revoke access to RQUSER2

Revoke access to all users

ore.revoke(name="ds_2", type="datastore", user="NULL")

Private Datastores
(schema local)

Shared Datastores
one table per datastore

Exploring Data

Scalability through proxies with function overloading

In-database summary – no data movement, only summary statistics

R Console

Oracle Distribution of R version 3.2.0 (--) -- "Full of Ingredients"

```
> summary(CUST_LTV[,c(1:5,30,31)])
```

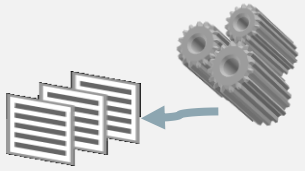
Oracle Advanced Analytics
ORE Client Packages

Transparency Layer

```
> summary(CUST_LTV[,c(1:5,30,31)])      # overloaded summary computes in-database
    CUST_ID          FIRST            LAST           STATE           REGION
 CU1     :     1   LACY     :   20   DICKENS :   16   NY      :5003   NorthEast:5469
 CU10    :     1   LAURENCE:   20   DUNCAN  :   16   CA      :3302   West      :4347
 CU100   :     1   LOGAN    :   20   SNYDER  :   16   MI      :2676   Midwest   :3348
 CU1000  :     1   ROBERTO  :   20   BAGGETT :   15   FL      : 613   South     :1143
 CU10000:      1   BRANDON  :   19   ALTMAN  :   14   NV      : 545   Southwest:1035
 CU10001:      1   (Other)  :15242   BROWNING:   14   MN      : 492
 (Other):15336   NA's     :    1   (Other) :15251   (Other):2711
      LTV            LTV_BIN
 Min.   :     0   HIGH     :7381
 1st Qu.:18657   MEDIUM   :5049
 Median :23054   VERY HIGH:1466
 Mean   :22267   LOW      :1446
 3rd Qu.:26104
 Max.   :47502
```

Oracle SQL

```
select ...
from CUST_LTV
```



In-db Stats

CUST_LTV

Scalability through proxies with function overloading

In-database histogram – no data movement, only summary statistics to construct graph

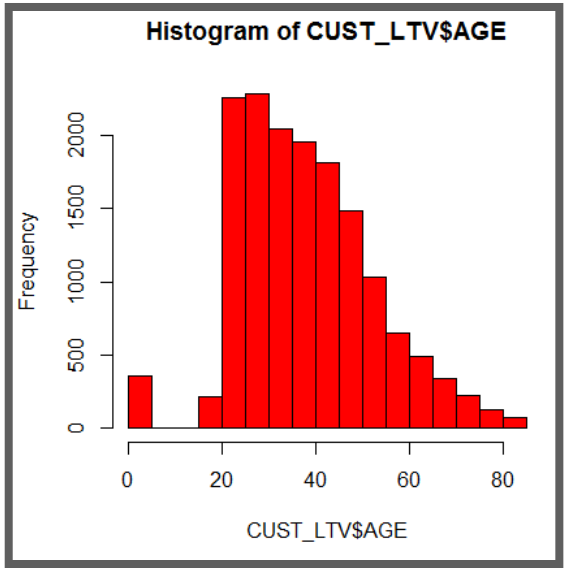
R Console

Oracle Distribution of R version 3.2.0 (--) -- "Full of Ingredients"

```
> hist(CUST_LTV$AGE, col="red")
```

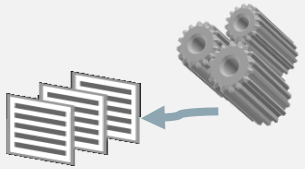
Oracle Advanced Analytics
ORE Client Packages

Transparency Layer



Oracle SQL

```
-- compute hist
-- statistics
select ...
from CUST_LTV
```



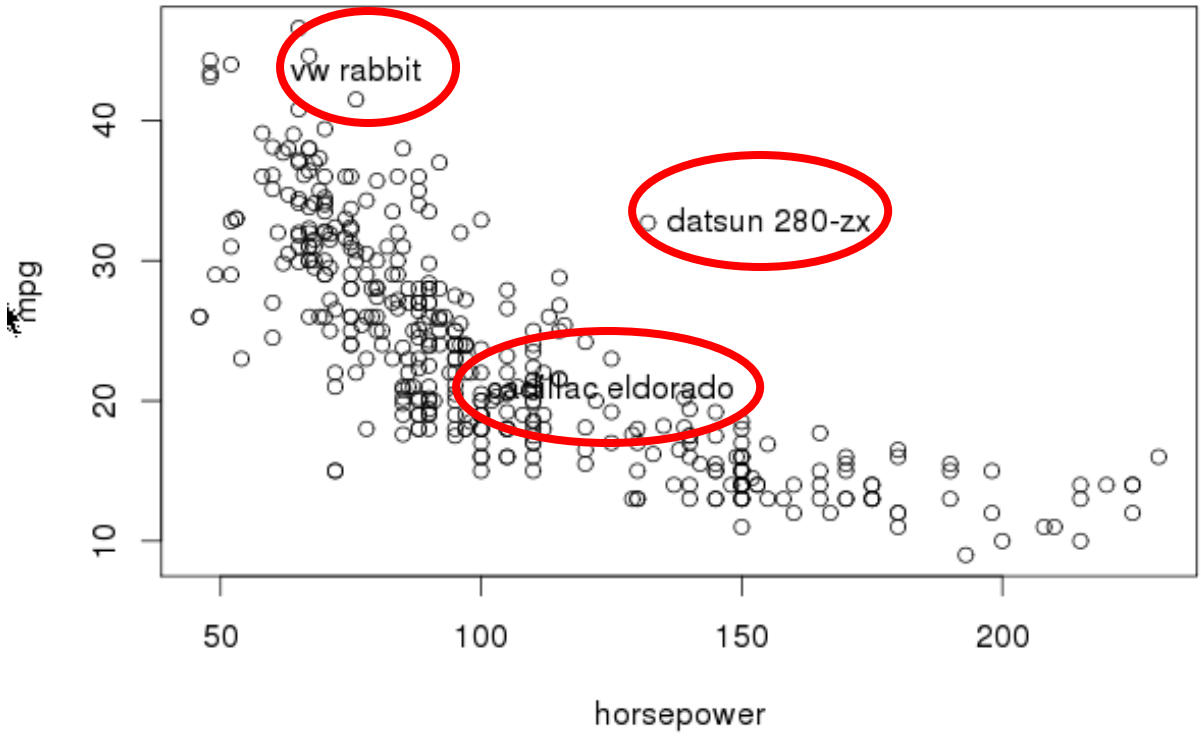
In-db Stats

CUST_LTV

Hint with identify()

- Identify allows you to click on points in the graph and get a label and index
- After clicking on multiple points, hit ESC

```
> with(AUTO, plot(horsepower ,mpg))
> with(AUTO, identify (horsepower ,mpg ,name))
[1] 101 112 135
>
```



Preparing Data

Recode using function definition

```
hasChildren_fmt <- function (x) {
  ifelse(x=='0', 'No',
  ifelse(x=='1', 'Yes','unknown'))
}

#-- recode the values by invoking the function and
#--    reassigning to the same variable
CUST_LTV$HAS_CHILDREN <- hasChildren_fmt(CUST_LTV$HAS_CHILDREN)
```

Recode using transform()

```
AUTO <- transform(AUTO,
                  origin2 = ifelse(origin==1,"American",
                            ifelse(origin==2,"European",
                            ifelse(origin==3,"Japanese","unknown"))))
```

Not in script.
Try it yourself!

Recode and categorical binning using ore.recode

```
AUTO$origin3 <-  ore.recode(AUTO$origin,
                            c("1", "2", "3"),
                            c("American", "European", "Japanese"),
                            "unknown")
table(AUTO$origin3)
```

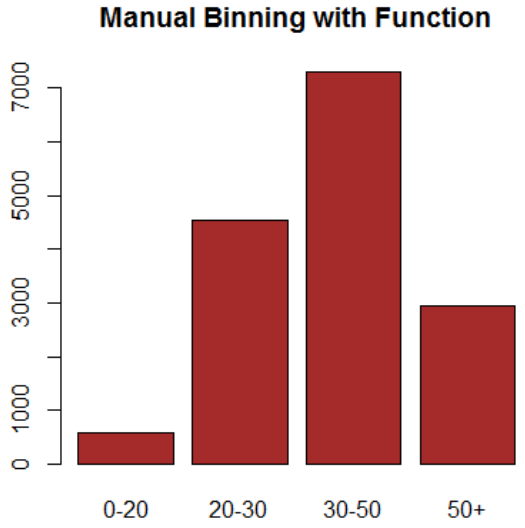
```
> table(AUTO$origin3)

American European Japanese
     245       68       79
```

```
AUTO$origin4 <-  ore.recode(AUTO$origin,
                            c("1", "2", "3"),
                            c("American", "Foreign", "Foreign"),
                            "unknown")
table(AUTO$origin4)
```

```
> table(AUTO$origin4)

American  Foreign
     245      147
```

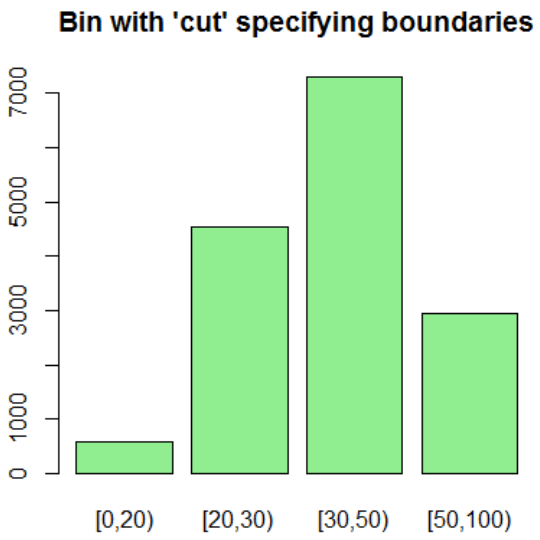
Bin data specifying explicit boundaries

```
CUST_LTV$AGE_BIN <- with(CUST_LTV, ifelse(AGE < 20, "0-20",
                                   ifelse(AGE < 30, "20-30",
                                   ifelse(AGE < 50, "30-50",
                                   ifelse(AGE >=50,"50+","unknown")))))
barplot(table(CUST_LTV$AGE_BIN),col="brown",
        main="Manual Binning with Function")
```



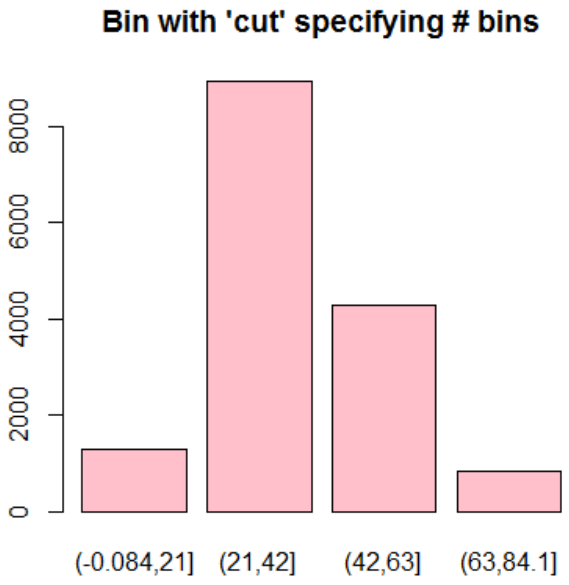
Bin data specifying explicit boundaries using ‘cut’

```
CUST_LTV$AGE_BIN3 <- cut(CUST_LTV$AGE,c(0,20,30,50,100),right=FALSE)

barplot(table(CUST_LTV$AGE_BIN3),col="lightgreen",
        main="Bin with 'cut' specifying boundaries")
```



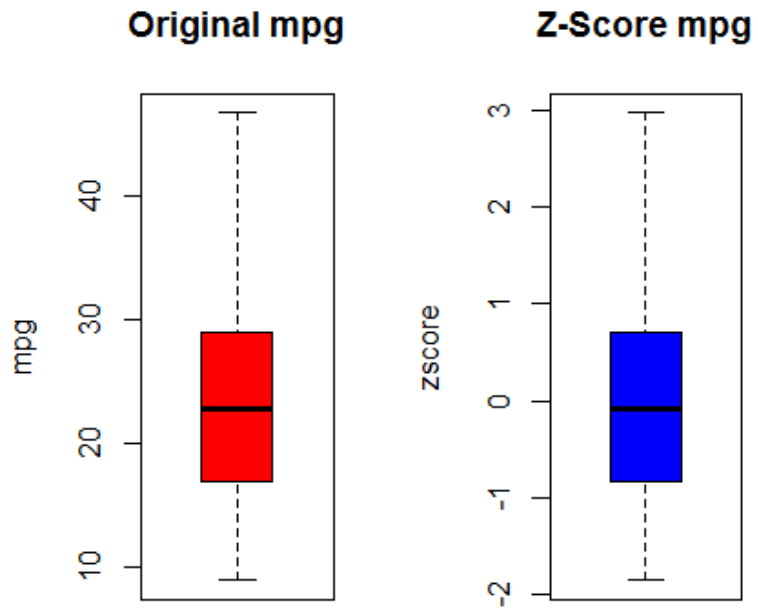
Bin data using number of bins with ‘cut’

```
CUST_LTV$AGE_BIN2 <- cut(CUST_LTV$AGE,4)
barplot(table(CUST_LTV$AGE_BIN2),col="pink",
        main="Bin with 'cut' specifying # bins")
```



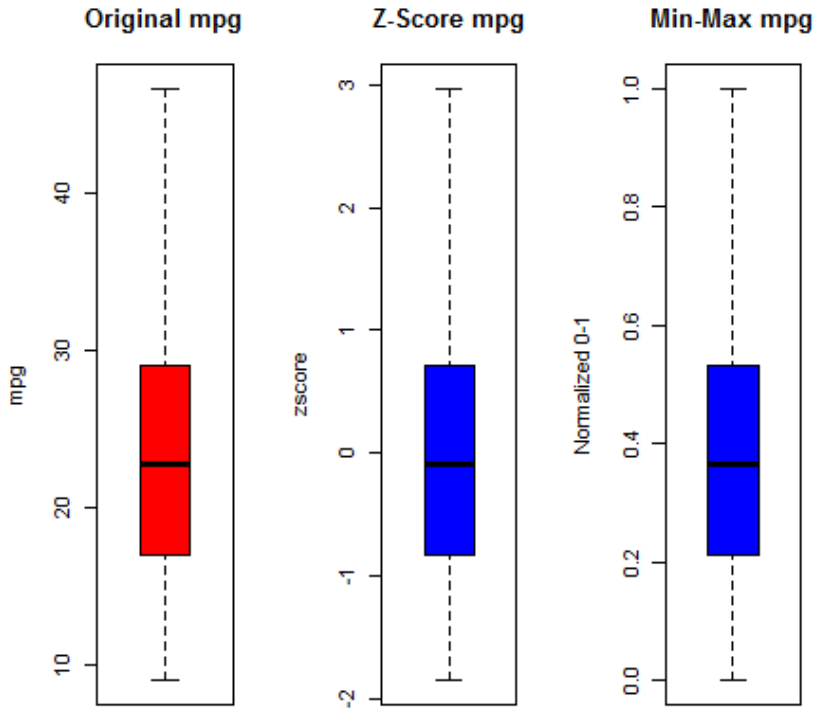
Normalize data using z-score

```
zscore <- function(x) {
  (x-mean(x,na.rm=TRUE))/sd(x,na.rm=TRUE)
}
AUTO$mpg_zscore   <- zscore(AUTO$mpg)
par(mfrow = c(1, 2))
boxplot(AUTO$mpg, main="Original mpg", ylab="mpg", col="red")

#-- notice the distribution is the same, but scale is different
boxplot(AUTO$mpg_zscore, main="Z-Score mpg", ylab="zscore", col="blue")
par(mfrow = c(1, 1))
```



Normalize data using min-max

```
minMaxNormalize <- function(x) {
  mn <- min(x)
  mx <- max(x)
  y <- (x-mn)/(mx-mn)
  y
}
#-- invoke function to normalize mpg
AUTO$mpg_minMaxNorm   <- minMaxNormalize(AUTO$mpg)
par(mfrow = c(1, 3))
boxplot(AUTO$mpg,main="Original mpg",ylab="mpg",col="red")
boxplot(AUTO$mpg_zscore, main="Z-Score mpg",ylab="zscore",col="blue")
#-- notice the distribution is the same, but again, scale is different, between 0 & 1
boxplot(AUTO$mpg_minMaxNorm, main="Min-Max mpg",ylab="Normalized 0-1",col="blue")
par(mfrow = c(1, 1))
```



Outlier Treatment – define function that sets outliers to NA

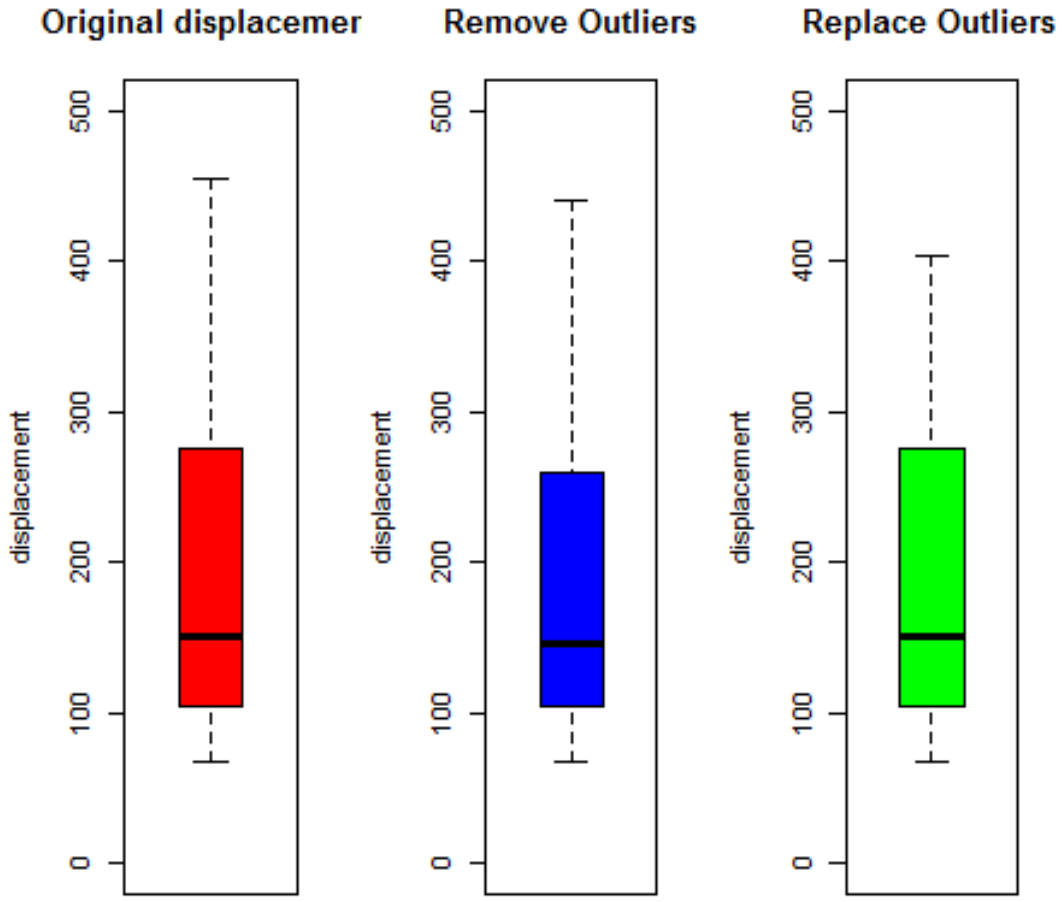
```
removeOutliers <- function(x, multiplier = 1.5, na.rm = TRUE, ...) {
  qnt <- quantile(x, probs=c(.25, .75), na.rm = na.rm, ...)
  H <- multiplier * IQR(x, na.rm = na.rm)
  y <- x
  y[x < (qnt[1] - H)] <- NA
  y[x > (qnt[2] + H)] <- NA
  y
}

#-- invoke function to remove outliers
AUTO$displacementRMO <- removeOutliers(AUTO$displacement,1)
```

Outlier Treatment – define function set to 2*sd from mean

```
capOutliers <- function(x, num.sd = 2, na.rm = TRUE, ...) {
  myMean <- mean(x,na.rm=na.rm)
  mySD   <- sd(x,na.rm=na.rm)
  maxLimit <- myMean + num.sd*mySD
  minLimit <- myMean - num.sd*mySD   y <- x
  y[x < minLimit] <- minLimit
  y[x > maxLimit] <- maxLimit
  y
}

#-- replace outliers
AUTO$displacementRPO <- capOutliers(AUTO$displacement)
```

Outlier Treatment – Compare results

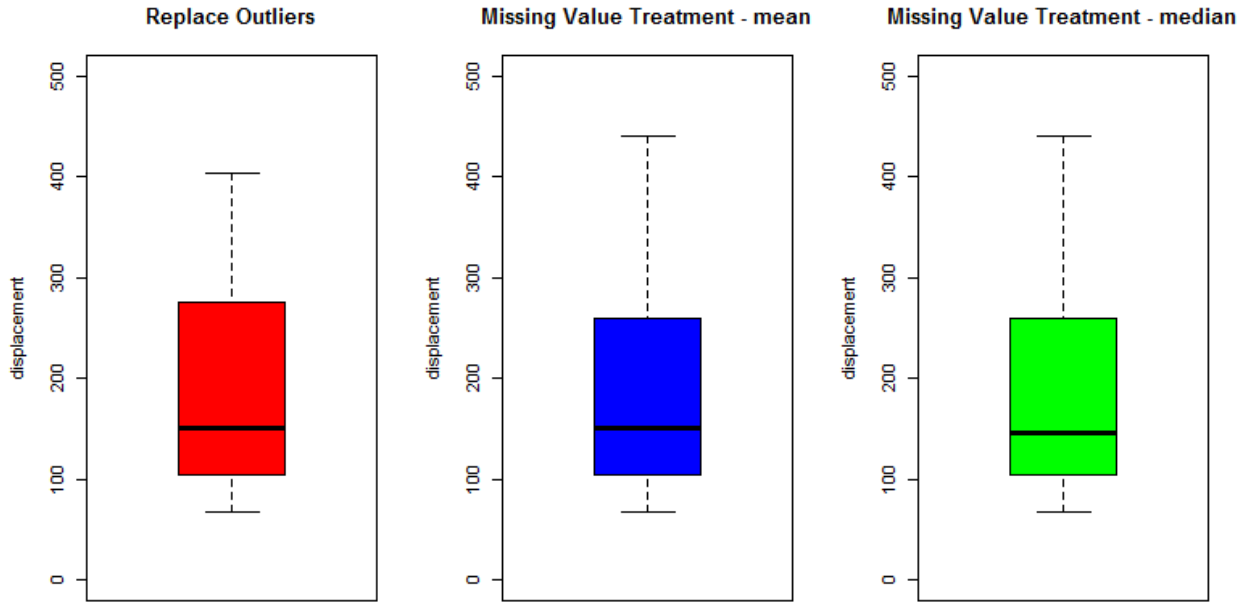
```
par(mfrow = c(1, 3))
boxplot(AUTO$displacement,
  main="Original displacement",
  ylab="displacement",col="red",ylim=c(0,500))
boxplot(AUTO$displacementRMO,
  main="Remove Outliers",
  ylab="displacement",col="blue",ylim=c(0,500))
boxplot(AUTO$displacementRPO,
  main="Replace Outliers",
  ylab="displacement",col="green",ylim=c(0,500))
par(mfrow = c(1, 1))
```



Missing Value Treatment –assign mean to missing values

```
missingValueTeatment <- function (x, value=mean(x,na.rm=TRUE)) {
  y <- ifelse(is.na(x), value,x)
  y
}
#-- treat missing values
AUTO$displacementMVT <- missingValueTeatment(AUTO$displacementRMO)
AUTO$displacementMVT2 <- missingValueTeatment(AUTO$displacementRMO,
     value=median(AUTO$displacementRMO,na.rm=TRUE))
```

```
par(mfrow = c(1, 3))
boxplot(AUTO$displacementRPO, main="Replace Outliers",
        ylab="displacement",col="red",ylim=c(0,500))
boxplot(AUTO$displacementMVT,
        main="Missing Value Treatment-mean",
        ylab="displacement",col="blue",ylim=c(0,500))
boxplot(AUTO$displacementMVT2,
       main="Missing Value Treatment-median",
        ylab="displacement",col="green",ylim=c(0,500))
par(mfrow = c(1, 1))
```



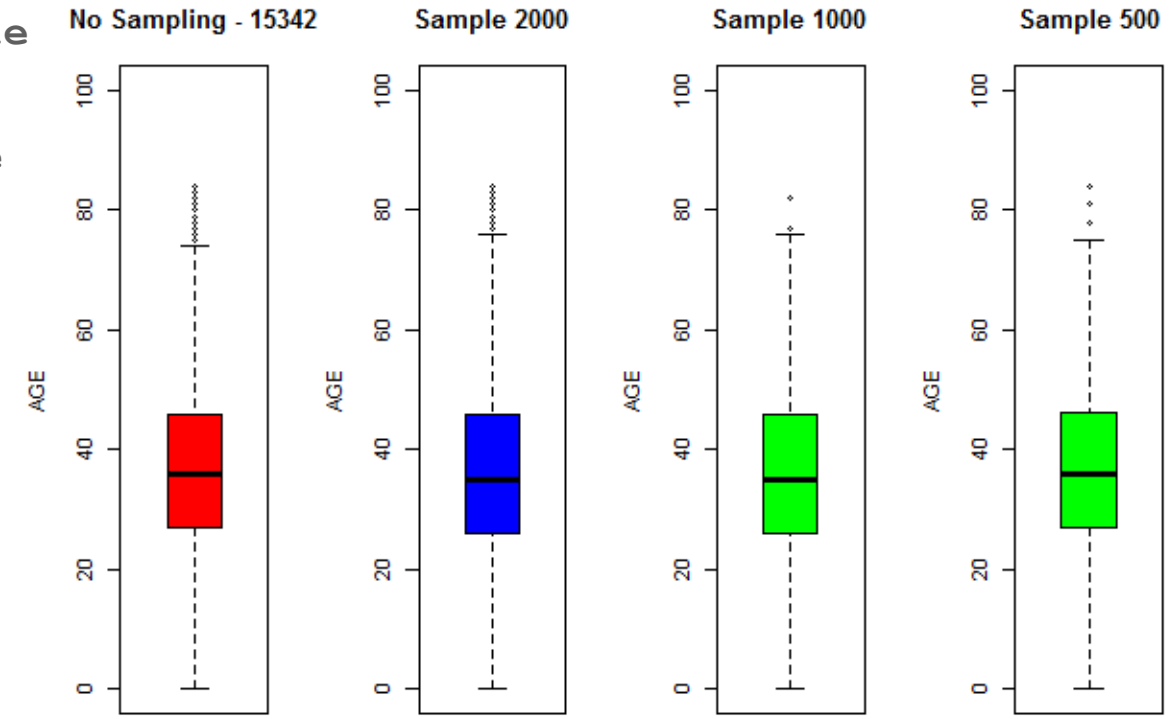
Sampling

Sampling

```
set.seed(1)                                    # make example repeatable
N <- nrow(CUST_LTV)
sampleSize <- 2000
row.names(CUST_LTV) <- CUST_LTV$CUST_ID        # assign row.names
simpleRandomSample2000 <- CUST_LTV[sample(N, sampleSize), ,drop=FALSE]

#-- take a few samples and compare distribution
simpleRandomSample1000 <- CUST_LTV[sample(N, 1000), ,drop=FALSE]
simpleRandomSample500 <- CUST_LTV[sample(N, 500), ,drop=FALSE]
```

Sampling – compare results

```
par(mfrow = c(1, 4))

boxplot(CUST_LTV$AGE, main=paste("No Sampling -",N),ylab="AGE",col="red",ylim=c(0,100))

boxplot(simpleRandomSample2000$AGE, main="Sample
  2000",ylab="AGE",col="blue",ylim=c(0,100))

boxplot(simpleRandomSample1000$AGE, main="Sample
  1000",ylab="AGE",col="green",ylim=c(0,100))

boxplot(simpleRandomSample500$AGE, main="Sample
  500",ylab="AGE",col="green",ylim=c(0,100))

par(mfrow = c(1, 1))
```



Split Sampling – train and test

```
N <- nrow(CUST_LTV)
trainPct <- 60
ind <- sample(1:N,trainPct*N/100)            # get indices for samples
group <- as.integer(1:N %in% ind)            # generate logical vector for selection

row.names(CUST_LTV) <- CUST_LTV$CUST_ID      # assign row names to enable row indexing
CUST_LTV.train <- CUST_LTV[group==FALSE,]    # select the train records
CUST_LTV.test <- CUST_LTV[group==TRUE,]      # select the test records

#-- compare total number of rows in source, with sum of train and test
nrow(CUST_LTV)
nrow(CUST_LTV.train) + nrow(CUST_LTV.test)
```

Model Building and Scoring

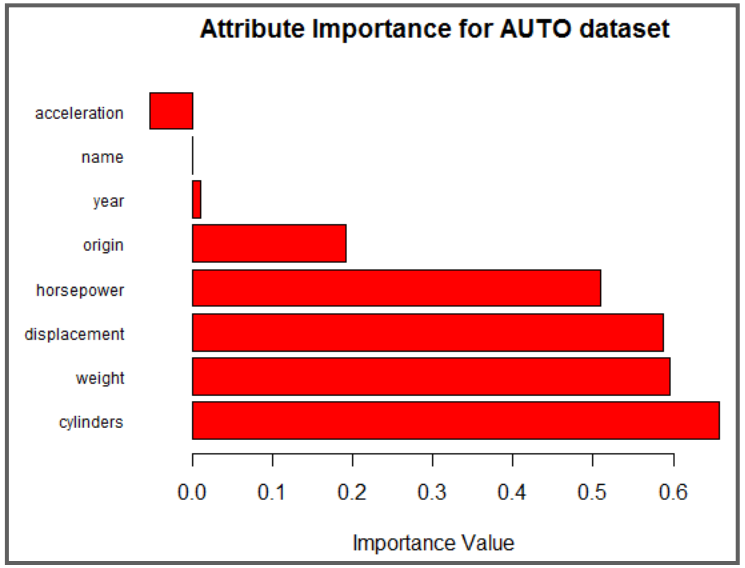
R: Scalable Machine Learning Models

Invoke in-database Data Mining algorithm (e.g., Attribute Importance)

R Console

Oracle Distribution of R version 3.2.0 (--) -- "Full of Ingredients"

```
> res <- ore.odmAI(mpg ~ ., AUTO)
> res
> barplot(...)
Call:
ore.odmAI(formula = mpg ~ ., data = AUTO)

Importance:
               importance rank
cylinders      0.657608467    1
weight         0.595553055    2
displacement   0.586937157    3
horsepower     0.509443482    4
origin         0.191069576    5
year           0.009964736    6
name           0.000000000    7
acceleration  -0.053014510    8
```

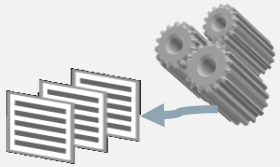


Oracle Advanced Analytics
ORE Client Packages

Predictive Analytics

Oracle PL/SQL

```
BEGIN
DBMS_DATA_MINING.CREATE_MODEL(
 model_name => ' ORE$x_y' ,
 mining_function =>
dbms_data_mining.attribute_impo
rtance...
```



In-db Mining Model

Build and Score Models

- Single model
  - Attribute Importance
  - Regression - SVM
  - Classification – SVM, Decision Tree, randomForest

Build multiple models in parallel

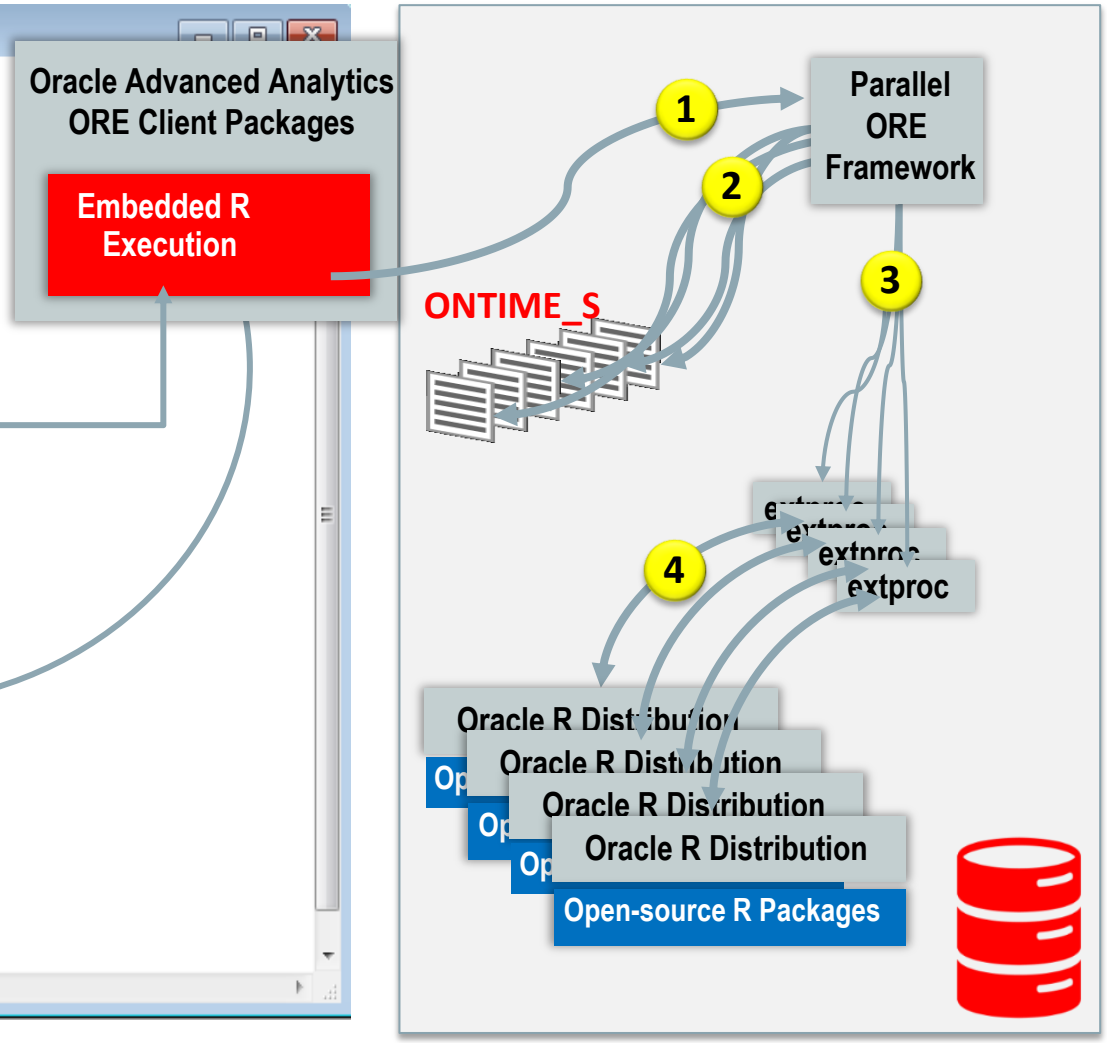
R: Embedded R for using open-source R algorithms

Server execution via open source R package: Embedded R + parallel group-by ore.groupApply()

R Console

Oracle Distribution of R version 3.2.0 (--) -- "Full of Ingredients"

```
> res <- ore.groupApply(DAT, INDEX = DAT$UNIQUECARRIER,
          function(df) {
             if(nrow(df) == 0)  NULL
               else coef(lm(ARRDELAY ~ DEPDELAY+DISTANCE, data=df))
             },
          parallel = 4)
> res
$AA
 (Intercept)     DEPDELAY     DISTANCE
-0.083722576  1.028404550 -0.001306912
$DL
  (Intercept)      DEPDELAY      DISTANCE
 3.4479245070  0.6630714546 -0.0008018419
$NW
 (Intercept)     DEPDELAY     DISTANCE
 0.659437522  1.007833579 -0.001454464
$UA
 (Intercept)     DEPDELAY     DISTANCE
-0.205089807  1.018052889 -0.001078487
```

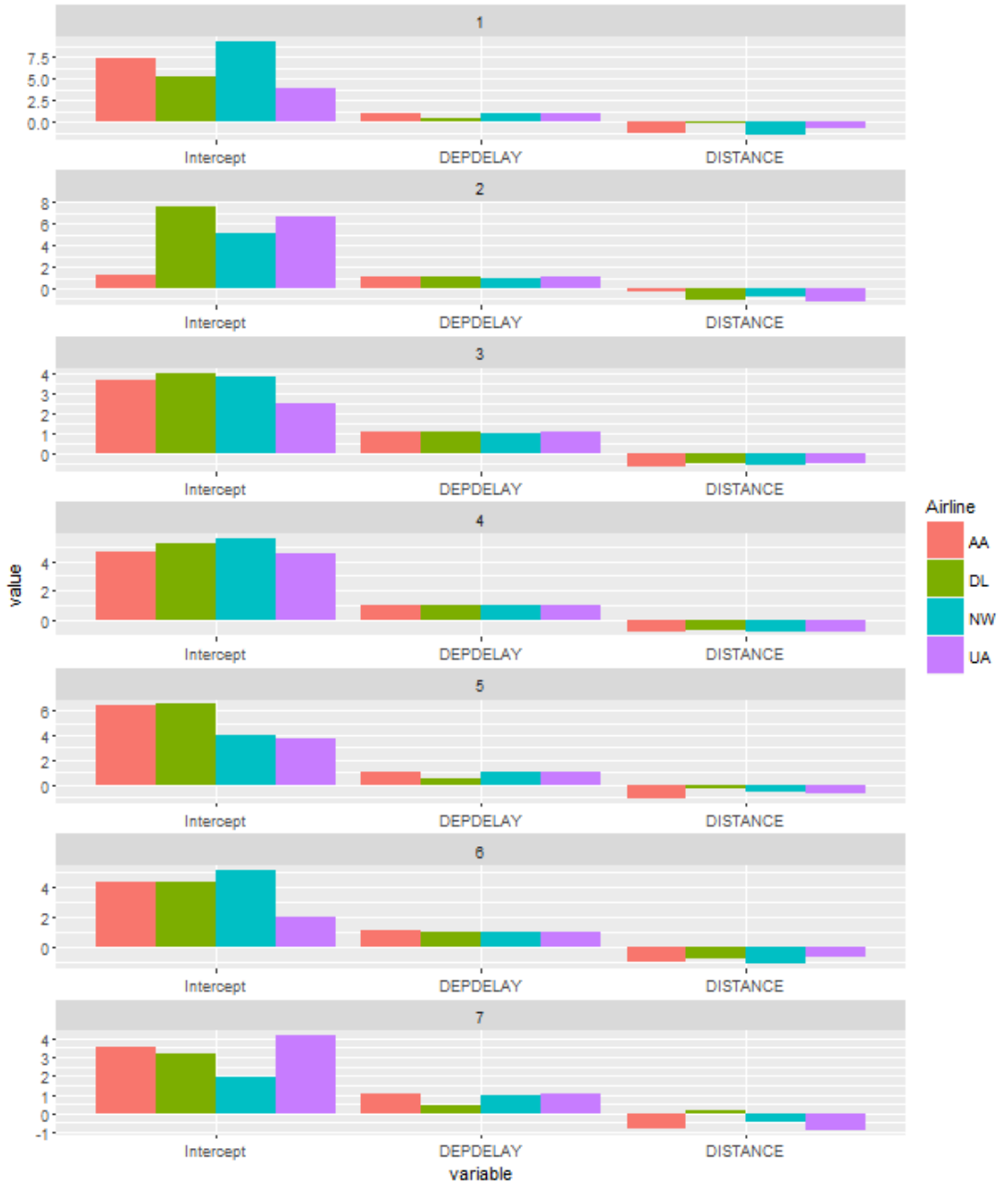


Group Apply with multiple columns

```
res <- ore.groupApply(DAT,
                      INDEX = DAT[,c(1,2)],
                      function(df) {
                        if(nrow(df) == 0)
                          NULL
                        else {
                          cc <- coef(lm(ARRDELAY ~ DEPDELAY+log(DISTANCE),data=df))
                          df <- data.frame(Day=df[1,1],
                                           Airline=df[1,2],
                                           Intercept=cc[1],
                                           DEPDELAY=cc[2],
                                           DISTANCE=cc[3])
                          row.names(df) <- NULL
                          df
                        }
                      },
                      parallel = 4)
```

Reshape data to facilitate plotting using ggplot

```
library(reshape2)
library(ggplot2)

#-- pull data to client since melt not overloaded
res2 <- melt(ore.pull(res), id=c("Day","Airline"))
head(res2)

#-- notice differences in intercept for certain
#--    airlines on certain days
#-- notice distance has inverse relationship to
#--    arrival delay -- making up time

ggplot(data=res2,
       aes(x=variable, y=value, fill=Airline)) +
       geom_bar(stat="identity",
                position=position_dodge()) +
       facet_wrap(~Day, ncol=1, scales="free")
```

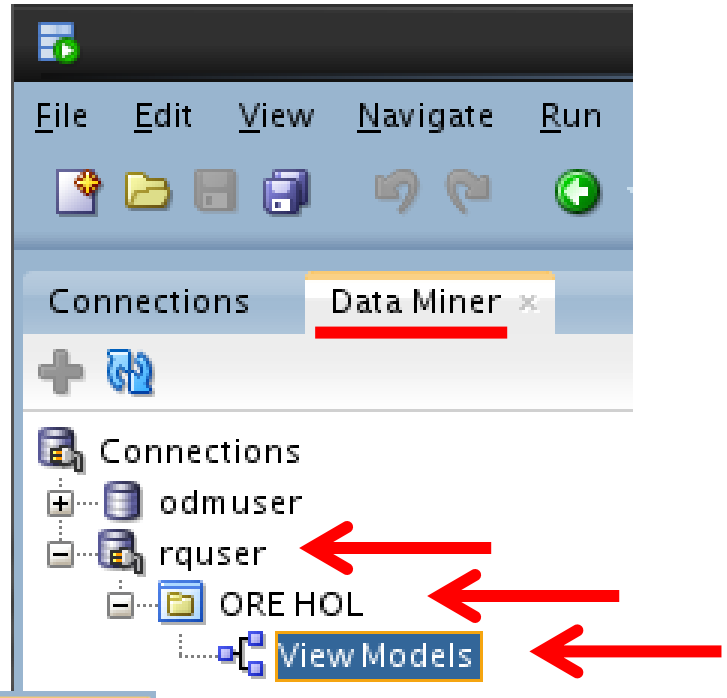


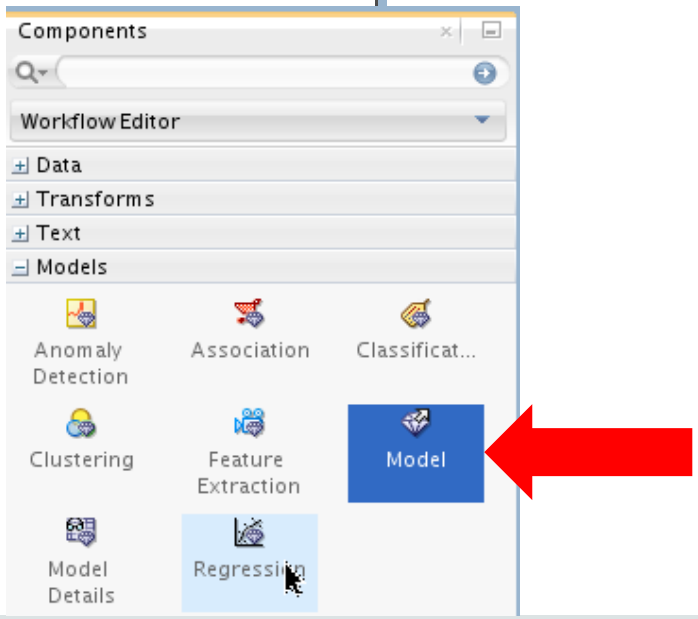
View Models in Oracle Data Miner

Viewing ODM models in ODMr

- Support Vector Machine Model
- Decision Tree Model

- Go to SQL Developer and navigate to this location
- Double click “View Models” workflow
- Select “Model” from components
- Click on canvas and select model(s)





Select the model, click OK, right click “View Models”

Oracle SQL Developer : rquser - localhost/ORE Demos/Model Viewing

File Edit View Navigate Run Diagram Team Tools Window Help

Model Viewing - Structure

Model

Start Page | 20150914 Short Demo Script.sql | ORE-HelloWorld.sql | Model Viewing

100% Parallel Query Off

Connect
Run
Force Run
Edit...
View Models
Generate Apply Chain
Show Event Log
Validate Parents

CLAS_DT_1_15

Edit Model Selection

Model Function: Classification

Values...

Data Type: VARCHAR2

Available Compatible Models

Name	Function:	Algorithm	Target	Data Type	Creation Date
CLAS_DT_1_7	Classification	Decision Tree	BUY_INSUR...	VARCHAR2	2015-05-26
CLAS_GLM_1_15	Classification	Generalized Linea...	BUY_INSUR...	VARCHAR2	2015-05-26
CLAS_GLM_1_7	Classification	Generalized Linea...	BUY_INSUR...	VARCHAR2	2015-05-26
CLAS_GLM_5_...	Classification	Generalized Linea...	BUY_INSUR...	VARCHAR2	2015-05-26
CLAS_GLM_5_...	Classification	Generalized Linea...	BUY_INSUR...	VARCHAR2	2015-05-26
CLAS_NB_1_15	Classification	Naive Bayes	BUY_INSUR...	VARCHAR2	2015-05-26
CLAS_NB_1_7	Classification	Naive Bayes	BUY_INSUR...	VARCHAR2	2015-05-26
CLAS_SVM_1_15	Classification	Support Vector M...	BUY_INSUR...	VARCHAR2	2015-05-26
CLAS_SVM_1_7	Classification	Support Vector M...	BUY_INSUR...	VARCHAR2	2015-05-26
CLAS_SVM_5_...	Classification	Support Vector M...	BUY_INSUR...	VARCHAR2	2015-05-26
CLAS_SVM_5_...	Classification	Support Vector M...	BUY_INSUR...	VARCHAR2	2015-05-26

Selected Models

Name	Algorithm
CLAS_DT_1_15	Decision Tree

Include Models from Other Schemas

Help OK Cancel

Connections | Data Mine

Connections
rquser - localhost
Analyst Demo
Demos
ORE Demos
Churn
Model Viewi
SLINSTANCE2

Components

Workflow Editor
Data
Transforms
Text
Models
Anomaly Detection
Association
Classification
Clustering
Feature Extraction
Model
Model Details
Regression
Predictive Queries
Evaluate and Apply
Linking Nodes

Model - Properties

Find

Models

Function:

Models

Name | Algorith

rquser - localhost/ORE Demos/Model Viewing

SVM Model

Oracle SQL Developer

File Edit View Navigate Run Team Tools Window Help

Connections | Data Miner

- Connections
  - odmuser
    - Big Data SQL
      - HDFS and DB Purchase Rec Model
    - Electronics R Us
      - Electronics R Us Target Cust
    - Movie Similarity
      - Text_Mining_and_Seg_Sample
  - rquser
    - ORE HOL
      - *View Models*
        - Model

Movie Similarity - Structure

No Structure

20160127 Scaling R to New Heights with Oracle Database - HOL.sql | *View Models* | ORE$67_1973

Coefficients | Compare | Settings

Target Value: MEDIUM

☑ Sort by absolute value

Query

Fetch Size: 1,000

Coefficients: 129 out of 129

Attribute

Attribute	Value	Coefficient
SALARY		-3.62397282
N_OF_DEPENDENTS		2.68779921
HOUSE_OWNERSHIP		-2.17526363
N_MORTGAGES		-2.17526363
<Intercept>		1.92351040
HAS_CHILDREN	Yes	0.91623391
HAS_CHILDREN	No	-0.91623391
MORTGAGE_AMOUNT		-0.74718671
PROFESSION	PROF-67	-0.63608742
TIME_AS_CUSTOMER		0.61923446
PROFESSION	PROF-45	0.57221627
PROFESSION	PROF-64	0.45628399
PROFESSION	PROF-39	-0.43514339
CREDIT_BALANCE		-0.42150830
PROFESSION	PROF-10	-0.40903611
CAR_OWNERSHIP		-0.40535047
BANK_FUNDS		-0.38713798
PROFESSION	DBA	0.37205652
PROFESSION	PROF-16	-0.37022650
PROFESSION	PROF-44	0.37022650
PROFESSION	PROF-47	-0.37022650
PROFESSION	PROF-49	-0.37022650
PROFESSION	PROF-53	-0.37022650

odmuser/Movie Similarity

Model Scoring

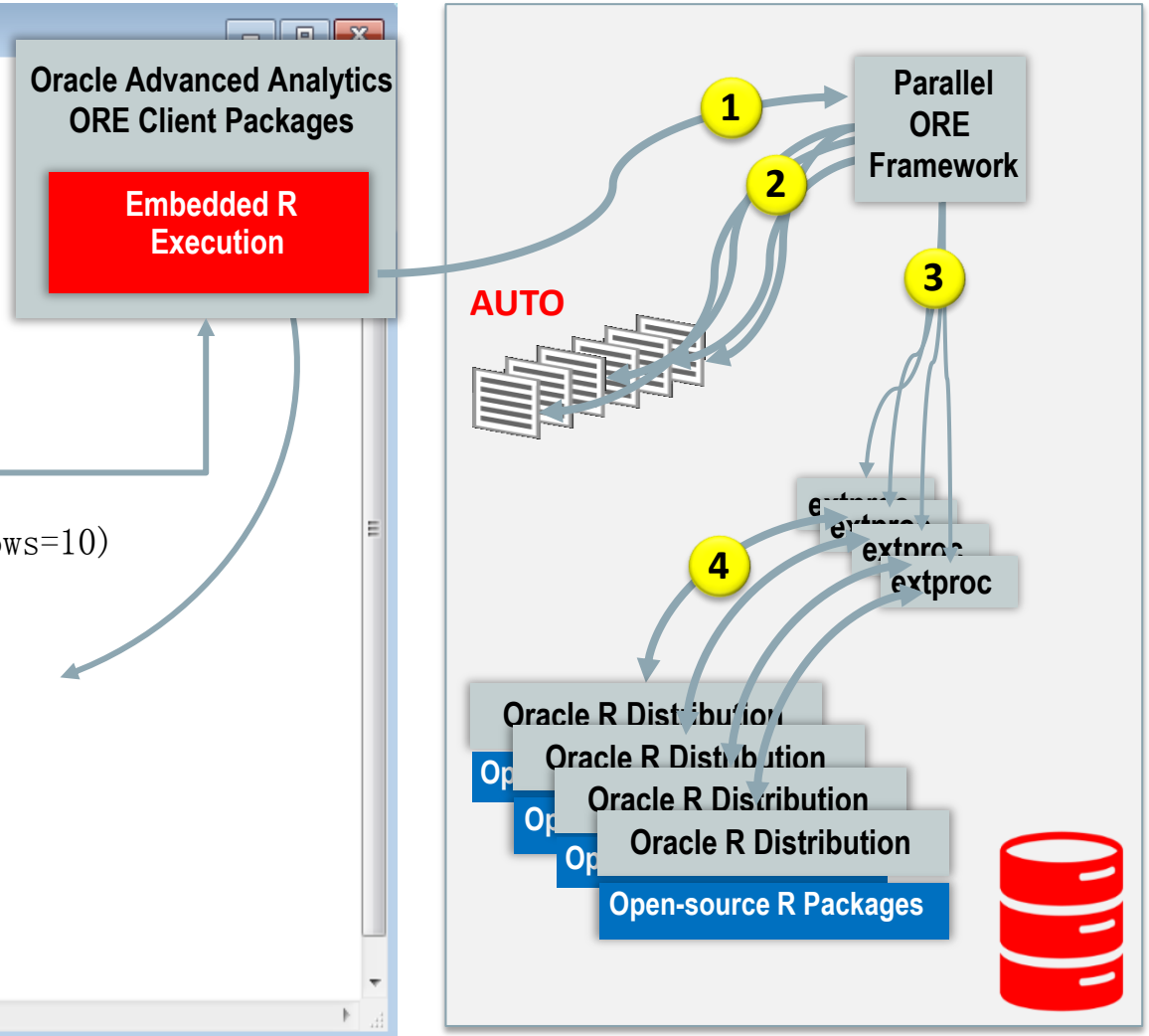
R: Embedded R for scoring open source R models

Server execution of open source R model: Embedded R + parallel ore.rowApply()

R Console

Oracle Distribution of R version 3.2.0 (--) -- "Full of Ingredients"

```
> cyl.mod.nb <- naiveBayes(cylinders ~ ., data = Auto2[,1:8])
> scoreNBmodel <- function(dat, mod) {
  library(e1071)
  dat$cylinders <- factor(dat$cylinders)
  dat$year      <- factor(dat$year)
  dat$origin    <- factor(dat$origin)
  dat$PRED      <- predict(mod, newdata = dat,type="class")
  dat
}
> res <- ore.rowApply(AUTO[,1:8], scoreNBmodel, mod = cyl.mod.nb, rows=10)
> res[[1]]                # view result
    mpg cylinders displacement horsepower weight acceleration year origin PRED
1  16.5         8          350        180   4380         12.1   76      1    8
2  13.0         8          350        145   4055         12.0   76      1    8
3  13.0         8          302        130   3870         15.0   76      1    8
4  13.0         8          318        150   3755         14.0   76      1    8
5  31.5         4           98         68   2045         18.5   77      3    4
6  30.0         4          111         80   2155         14.8   77      1    4
7  36.0         4           79         58   1825         18.6   77      2    4
8  25.5         4          122         96   2300         15.5   77      1    4
9  33.5         4           85         70   1945         16.8   77      3    4
10 17.5         8          305        145   3880         12.5   77      1    8
```



Factors and Embedded R Execution

- Original data.frame may contain factor variables
- Embedded R execution passes factors into user-defined function as chr
- Function must cast these factors for proper behavior, especially for model building

- Examples guide you through this process

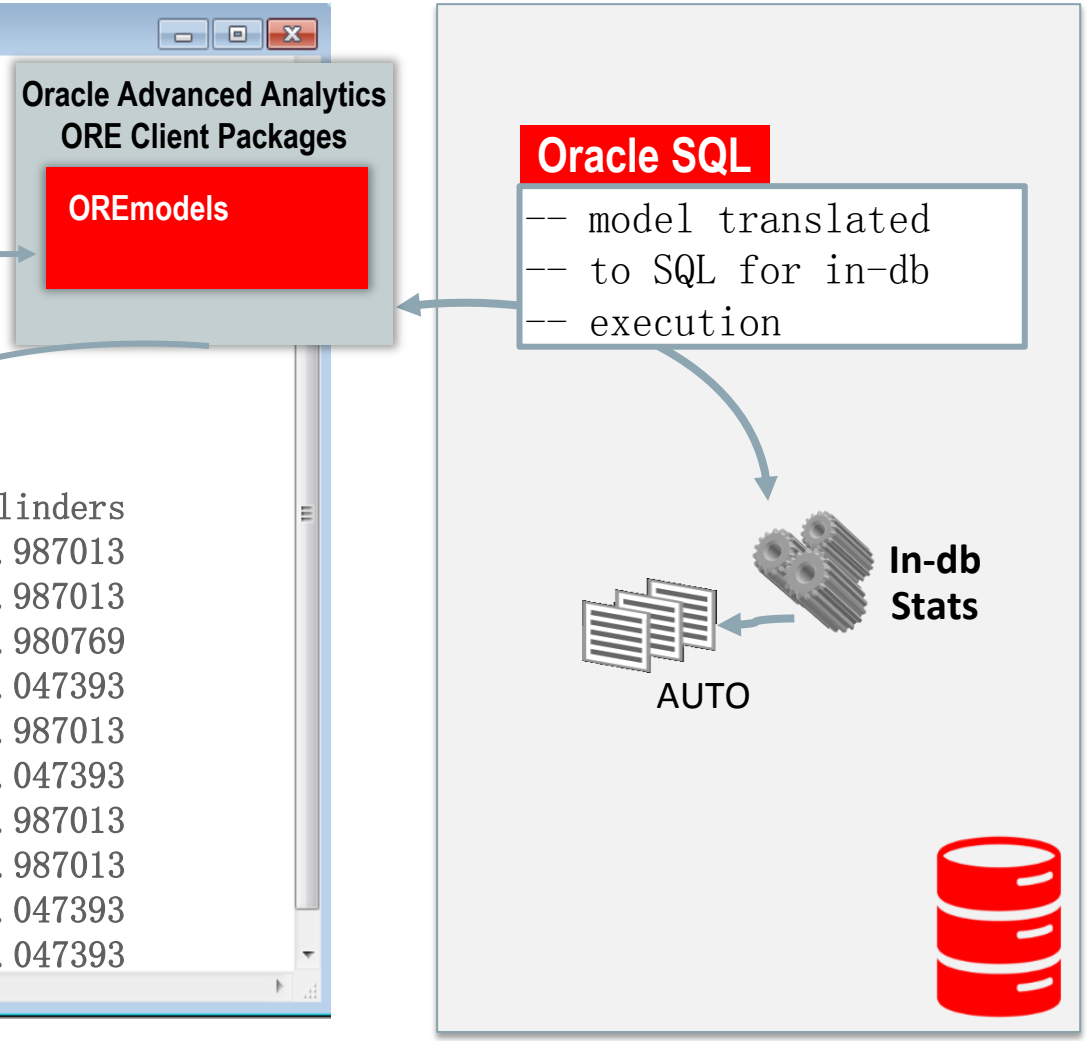
In-database R model scoring using ore.predict

R model translated to SQL – no data movement, results as ore.frame

R Console

Oracle Distribution of R version 3.2.0 (--) -- "Full of Ingredients"

```
> cyl.rpart.mod <- rpart(cylinders ~ .-name, data = Auto)
> AUTO.res <- ore.frame(AUTO[,c("name","cylinders")],
                        ore.predict(cyl.rpart.mod, AUTO))
> names(AUTO.res) <- c("name","cylinders","pred.cylinders")

> AUTO.res[sample(1:nrow(AUTO.res),10),]
                                        name cylinders pred.cylinders
buick century limited          buick century limited         6       5.987013
buick century special          buick century special         6       5.987013
buick lesabre custom            buick lesabre custom         8       7.980769
buick skylark limited          buick skylark limited         4       4.047393
chevrolet nova custom          chevrolet nova custom         6       5.987013
chrysler lebaron medallion chrysler lebaron medallion        4       4.047393
plymouth duster                      plymouth duster         6       5.987013
plymouth volare                      plymouth volare         6       5.987013
volkswagen super beetle       volkswagen super beetle        4       4.047393
volvo 145e (sw)                      volvo 145e (sw)         4       4.047393
```



Solution Deployment

Revise in-db scoring for deployment with R model

```
scoreNBmodel <- function(dat, mod) {
  library(e1071)
  dat$cylinders <- factor(dat$cylinders)
  dat$year <- factor(dat$year)
  dat$origin <- factor(dat$origin)
  dat$PRED <- predict(mod, newdata = dat,type="class")
  dat
}
```

```
#-- store script in R Script Repository
ore.scriptCreate("scoreNBmodel",scoreNBmodel)

#-- verify function works from R with ore.rowApply
res3 <- ore.rowApply(AUTO[,1:8],FUN.NAME="scoreNBmodel",
                     mod = cyl.mod.nb,  FUN.VALUE = res[[1]],rows=10)
head(res3)

#-- since can't pass a model in SQL, need to store in datastore
ore.delete("CYL_NB_MODEL_1")   # clean up if datastore exists from previous execution
ore.save(cyl.mod.nb,name="CYL_NB_MODEL_1")
ore.datastore()
```

Revise in-db scoring for deployment with R model

```
#-- revise function to load from datastore name
scoreNBmodel2 <- function(dat, dsname) {
  library(e1071)
  dat$cylinders <- factor(dat$cylinders)
  dat$year      <- factor(dat$year)
  dat$origin    <- factor(dat$origin)
  ore.load(dsname)
  dat$PRED      <- predict(cyl.mod.nb, newdata = dat, type="class")
  dat
}
#-- store revised script in R Script Repository
ore.scriptCreate("scoreNBmodel2", scoreNBmodel2)
```

Revise in-db scoring for deployment with R model

```
#-- verify function works from R with ore.rowApply
res4 <- ore.rowApply(
  AUTO[,1:8],
  FUN.NAME = "scoreNBmodel2",
  dsname = "CYL_NB_MODEL_1",
  FUN.VALUE = res[[1]],
  rows=10,
  ore.connect=TRUE)

head(res4)
```

```
> head(res4)
   mpg cylinders displacement horsepower weight acceleration year origin PRED
1 34.1         4           86         65   1975         15.2   79      3    4
2 35.7         4           98         80   1915         14.4   79      1    4
3 27.4         4          121         80   2670         15.0   79      1    4
4 25.4         5          183         77   3530         20.1   79      2    6
5 23.0         8          350        125   3900         17.4   79      1    8
6 27.2         4          141         71   3190         24.8   79      2    4
```

Execute R script from SQL

```
select 1 A, 2 B from dual; -- returns one row with values and column names specified


select *
from   table(rqRowEval(
            cursor(SELECT "mpg", "cylinders", "displacement", "horsepower",
                          "weight", "acceleration", "year", "origin"
                   FROM RQUSER2.AUTO), -- input data cursor
            cursor(SELECT 1 "ore.connect",'CYL_NB_MODEL_1' "dsname"
                   FROM dual), -- arguments to ORE and user-defined R function
            'SELECT 1 mpg, ''a'' cylinders, 1 displacement, 1 horsepower, 1 weight, 1
  acceleration, ''aa'' year, ''a'' origin, ''a'' PRED from dual', -- output format
            10,  -- num rows per execution
            'scoreNBmodel2'));  -- Function name in R Script Repository
```

Oracle Advanced Analytics: Embedded R Execution

SQL interface – generate image, table, or XML string for graphic output

Oracle PL/SQL

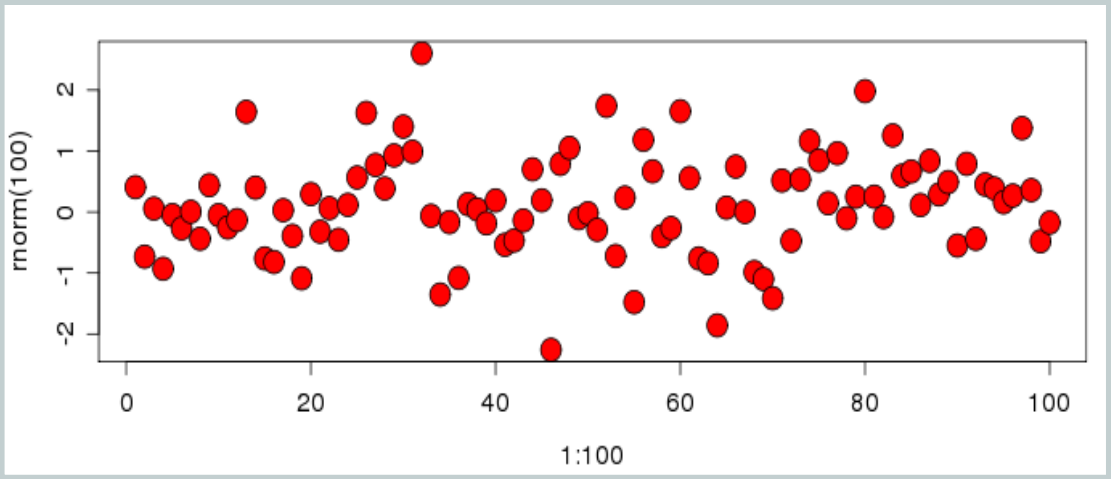
```
begin
  sys.rqScriptCreate('RandomRedDots',
 'function(){
      res <- 1:10
      plot( 1:100, rnorm(100), pch = 21,
            bg = "red", cex = 2 )
      data.frame(id=res,val=res/10)
      }');
end;
```

R Language

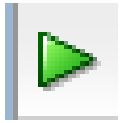
Oracle SQL

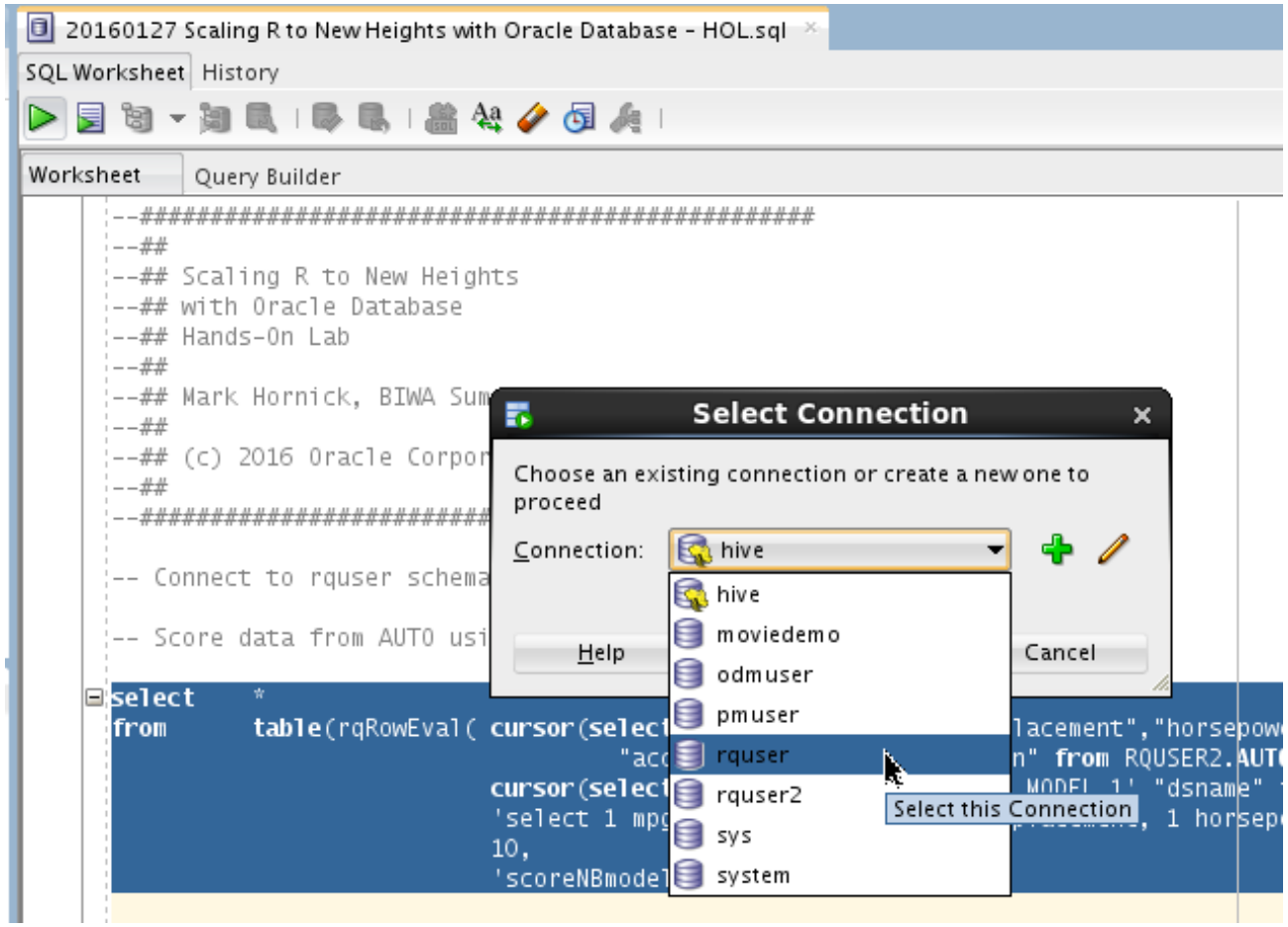
```
select value
from   table(rqEval(NULL,'PNG','RandomRedDots'));

select value
from   table(rqEval(NULL,
  'select 1 id, 1 val from dual'','RandomRedDots'));

select value
from   table(rqEval(NULL,'XML','RandomRedDots'));
```

- R script output is often dynamic – not conforming to pre-defined structure
- R apps generate stats, new data, graphics
- Example
  - Plot 100 random numbers, return image
  - Return a vector with values 1 to 10
  - Return image and data as XML



Invoking SQL in SQL Developer

- Highlight first select statement and click 
- Prompted to select connection
  - Choose ‘rquser’ and click OK



Sharing R Scripts

Create / List / Load / Grant R Scripts

```
ore.scriptCreate("privateFunction",
                 function(data, formula, ...) lm(formula, data, ...))
# create a global R script available to any user
ore.scriptCreate("globalFunction",
                 function(data, formula, ...) glm(formula=formula, data=data, ...),
                 global = TRUE)
ore.scriptList()                                # list locally available functions
ore.scriptList(pattern="Function",  type="all") # list global functions as well

# load an R script to an R function object
ore.scriptLoad(name="privateFunction")
ore.scriptLoad(name="globalFunction", newname="privateFunction2")

# grant and revoke R script read privilege to and from public
ore.grant(name = "privateFunction", type = "rqscript")
ore.scriptList(type="grant")  # no longer private!
```

What functions can RQUSER2 see

```
# Connect to RQUSER2 to see function
ore.connect("rquser2",service_name="pdborcl.oradev.oraclecorp.com",
            host="localhost",password="rquser2", all=TRUE)

ore.scriptList()
ore.scriptList(pattern="Function", type="all")

# Invoke shared private function to build lm model on iris data
ore.doEval(FUN.NAME="privateFunction",FUN.OWNER="RQUSER",
           formula=Sepal.Length~., data=iris)

ore.scriptLoad("privateFunction",owner="RQUSER")
privateFunction   # view function body
```

RQUSER revokes privilege

```
# Reconnect to RQUSER to revoke privileges
ore.disconnect()
ore.connect("rquser",service_name="pdborcl.oradev.oraclecorp.com",
            host="localhost",password="rquser", all=TRUE)

ore.revoke(name = "privateFunction", type = "rqscript")
ore.scriptList(type="grant")
```

Clean up R Script Repository

```
# Reconnect to RQUSER to clean up
ore.disconnect()
ore.connect("rquser",service_name="pdborcl.oradev.oraclecorp.com",
            host="localhost",password="rquser", all=TRUE)

# drop an R script
ore.scriptList(type="all")[,1:2]

ore.scriptDrop("privateFunction")
ore.scriptDrop("globalFunction", global=TRUE)
ore.scriptList(type="all")[,1:2]
```

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

ORACLE

ORACLE®