SQL Statistical Functions

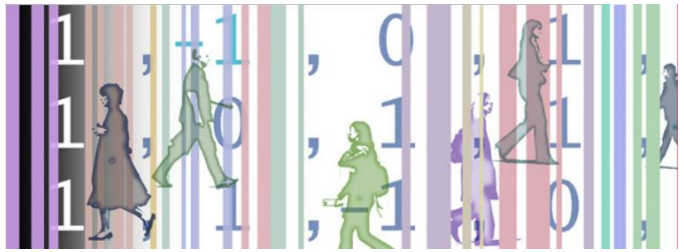
Make Big Data + Analytics Simple

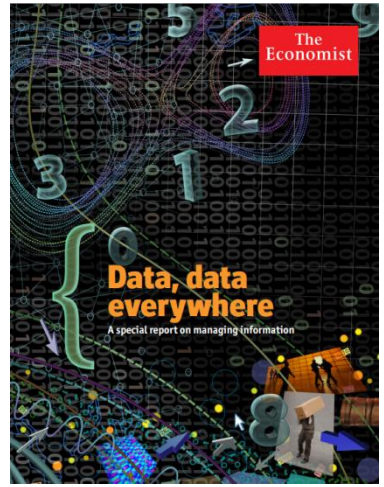
Charlie Berger, MS Engineering, MBA
Sr. Director Product Management, Data Mining and Advanced Analytics
charlie.berger@oracle.com www.twitter.com/CharlieDataMine

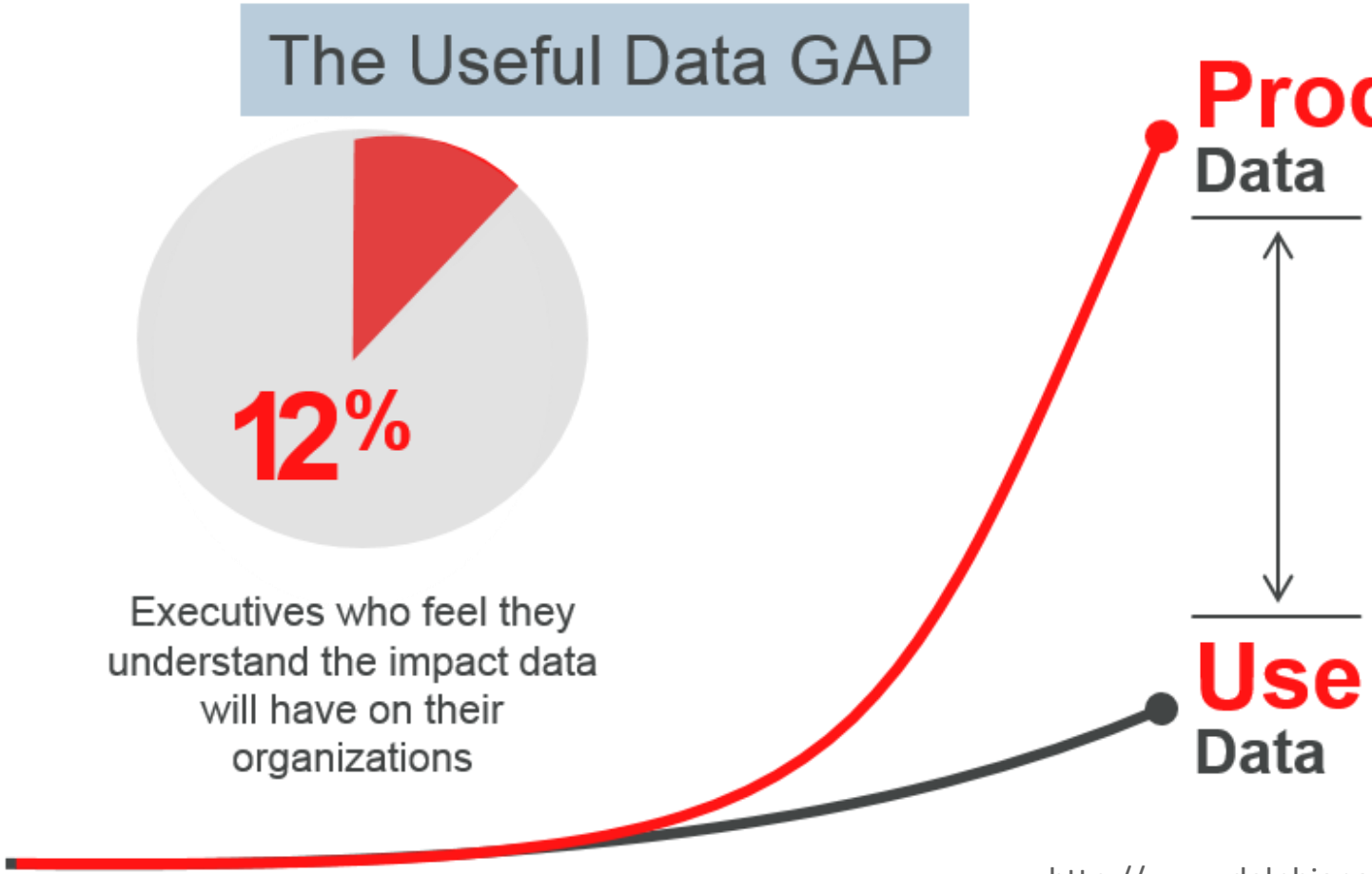
ORACLE

Data, data everywhere

Growth of Data Exponentially Greater than Growth of Data Analysts!







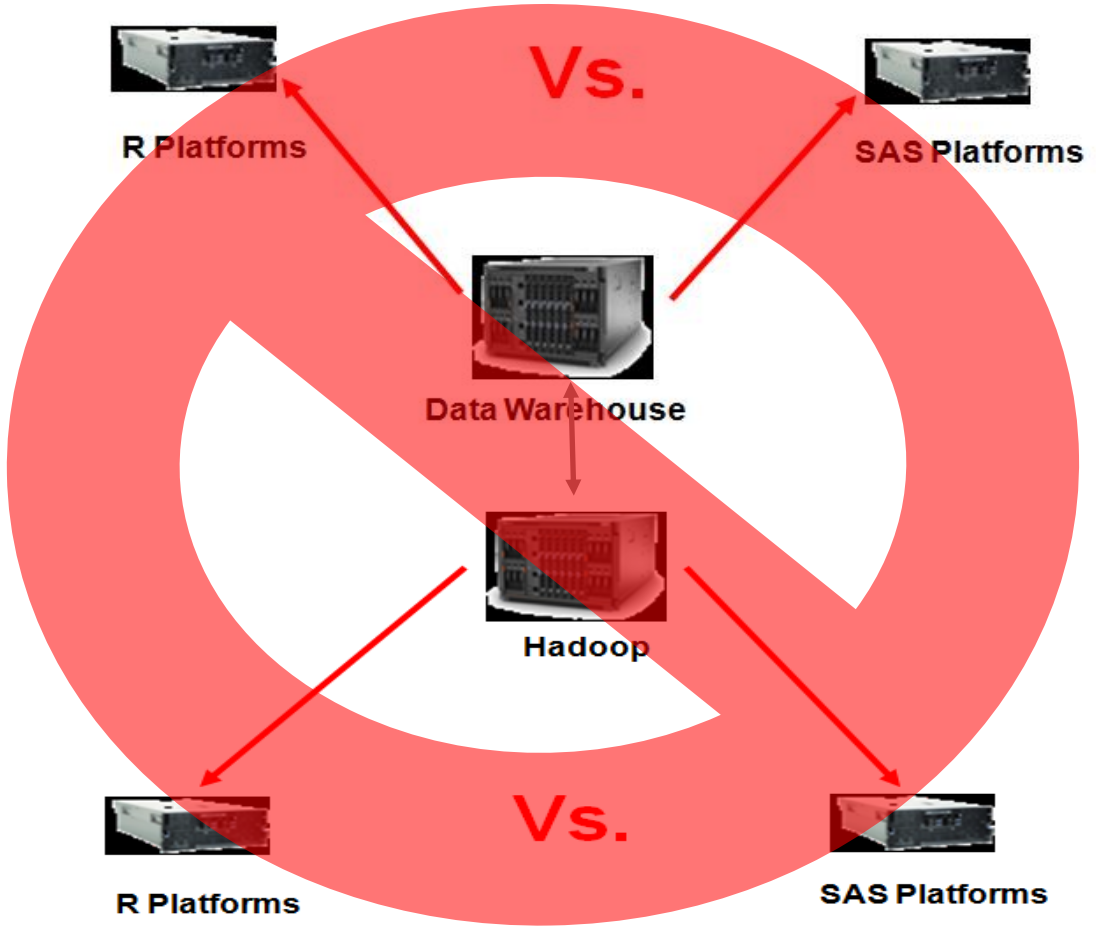
Data Analysis platforms requirements:

- Be extremely powerful and handle large data volumes
- Be easy to learn
- Be highly automated & enable deployment

http://www.delphianalytics.net/more-data-than-analysts-the-real-big-data-problem/
http://uk.emc.com/collateral/analyst-reports/ar-the-economist-data-data-everywhere.pdf

Analytics + Data Warehouse + Hadoop

- Platform Sprawl
  - More Duplicated Data
  - More Data Movement Latency
  - More Security challenges
  - More Duplicated Storage
  - More Duplicated Backups
  - More Duplicated Systems
  - More Space and Power



Oracle Advanced Analytics Database Evolution

1998

- *7 Data Mining "Partners"*

1999

- *Oracle acquires Thinking Machine Corp's dev. team + "Darwin" data mining software*

2002

- *Oracle Data Mining 9.2i launched – 2 algorithms (NB and AR) via Java API*

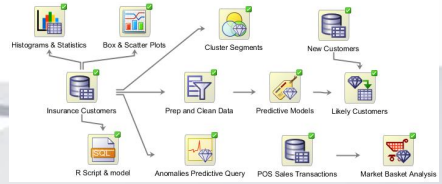
2004

- *Oracle Data Mining 10gR2 SQL - 7 new SQL dm algorithms and new Oracle Data Miner "Classic" wizards driven GUI*
- *SQL statistical functions introduced*

2005

- *ODM 11g & 11gR2 adds AutoDataPrep (ADP), text mining, perf. improvements*
- *SQLDEV/Oracle Data Miner 3.2 "work flow" GUI launched*
- *Integration with "R" and introduction/addition of Oracle R Enterprise*
- *Product renamed "Oracle Advanced Analytics (ODM + ORE)*

2008



2011

- *New algorithms (EM, PCA, SVD)*
- *Predictive Queries*
- *SQLDEV/Oracle Data Miner 4.0 SQL script generation and SQL Query node (R integration)*
- *OAA/ORE 1.3 + 1.4 adds NN, Stepwise, scalable R algorithms*
- *Oracle Adv. Analytics for Hadoop Connector launched with scalable BDA algorithms*

2014

You Can Think of Oracle Advanced Analytics Like This…

Traditional SQL

- "Human-driven" queries
- Domain expertise
- Any "*rules*" must be defined and managed

SQL Queries

- `SELECT`
- `DISTINCT`
- `AGGREGATE`
- `WHERE`
- `AND OR`
- `GROUP BY`
- `ORDER BY`
- `RANK`

+

SQL Statistical Functions - SQL & R

- Automated knowledge discovery, model building and deployment
- Domain expertise to assemble the "*right*" data to mine/analyze

Statistical SQL "Verbs"

- `MEAN, STDEV`
- `MEDIAN`
- `SUMMARY`
- `CORRELATE`
- `FIT`
- `COMPARE`
- `ANOVA`

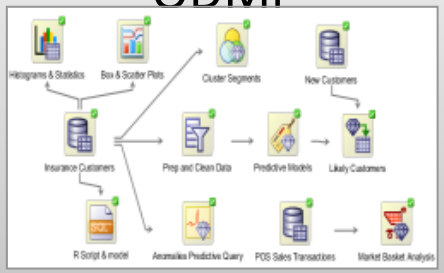
Oracle Advanced Analytics Database Architecture

Multi-lingual Component of Oracle Database—SQL, SQL Dev/ODMr GUI, R

Users

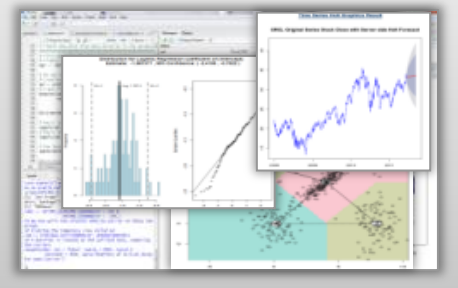
Data & Business Analysts

SQL Developer/ ODMr



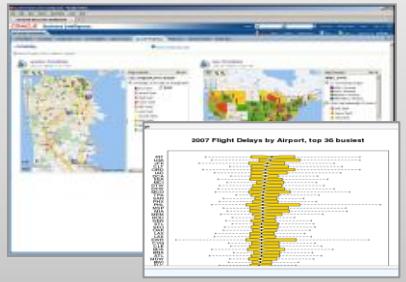
R programmers

R Client



Business Analysts/Mgrs

OBIEE



Domain End Users

Applications

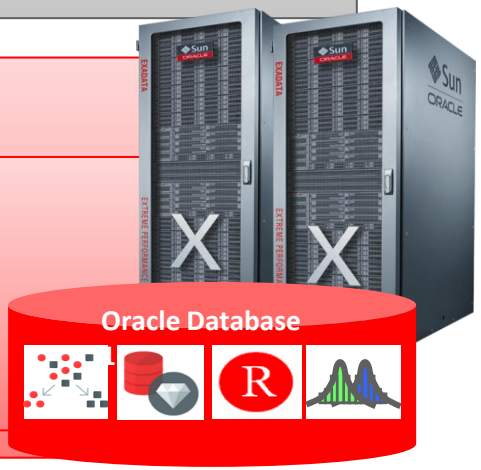
Platform

Oracle Database Enterprise Edition

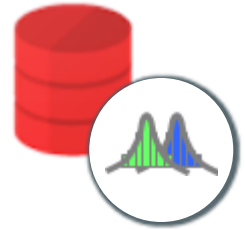
Oracle Advanced Analytics - Database Option

SQL Data Mining & Analytic Functions + R Integration

for Scalable, Distributed, Parallel in-Database ML Execution



Vision



- Big Data + Analytic Platform for the Era of Big Data and Cloud
  - **Make Big Data + Analytics Discovery *Simple***
    - Any data size, on any computer infrastructure
    - Any variety of data (structured, unstructured, transactional, geospatial), in any combination
  - **Make Big Data + Analytics Deployment *Simple***
    - As a service, as a platform, as an application

Oracle Advanced Analytics Database Option

Wide Range of In-Database Data Mining and Statistical Functions

- **Data Understanding & Visualization**
  - Summary & Descriptive Statistics
  - Histograms, scatter plots, box plots, bar charts
  - R graphics: 3-D plots, link plots, special R graph types
  - Cross tabulations
  - Tests for Correlations (t-test, Pearson's, ANOVA)
  - Selected Base SAS equivalents
- **Data Selection, Preparation and Transformations**
  - Joins, Tables, Views, Data Selection, Data Filter, SQL time windows, Multiple schemas
  - Sampling techniques
  - Re-coding, Missing values
  - Aggregations
  - Spatial data
  - SQL Patterns
  - R to SQL transparency and push down
- **Classification Models**
  - Logistic Regression (GLM)
  - Naive Bayes
  - Decision Trees
  - Support Vector Machines (SVM)
  - Neural Networks (NNs)
- **Regression Models**
  - Multiple Regression (GLM)
  - Support Vector Machines
- **Clustering**
  - Hierarchical K-means
  - Orthogonal Partitioning
  - Expectation Maximization
- **Anomaly Detection**
  - Special case Support Vector Machine (1-Class SVM)
- **Associations / Market Basket Analysis**
  - A Priori algorithm
- **Feature Selection and Reduction**
  - Attribute Importance (Minimum Description Length)
  - Principal Components Analysis (PCA)
  - Non-negative Matrix Factorization
  - Singular Vector Decomposition
- **Text Mining**
  - Most OAA algorithms support unstructured data (i.e. customer comments, email, abstracts, etc.)
- **Transactional & Spatial Data**
  - All OAA algorithms support transactional data (i.e. purchase transactions, repeated measures over time, distances from location, time spent in area A, B, C, etc.)
- **R packages—ability to run open source**
  - Broad range of R CRAN packages can be run as part of database process via R to SQL transparency and/or via Embedded R mode

ORACLE

* included free in every Oracle Database

Independent Samples T-Test

(Pooled Variances)

- Query compares the mean of AMOUNT_SOLD between MEN and WOMEN within CUST_INCOME_LEVEL ranges. Returns observed t value and its related two-sided significance

```
SELECT substr(cust_income_level,1,22) income_level,
  avg(decode(cust_gender,'M',amount_sold,null)) sold_to_men,
  avg(decode(cust_gender,'F',amount_sold,null)) sold_to_women,
  stats_t_test_indep(cust_gender, amount_sold, 'STATISTIC','F')
  t_observed,
  stats_t_test_indep(cust_gender, amount_sold) two_sided_p_value
FROM sh.customers c, sh.sales s
WHERE c.cust_id=s.cust_id
GROUP BY rollup(cust_income_level)
  ORDER BY 1;
```

SQL Plus

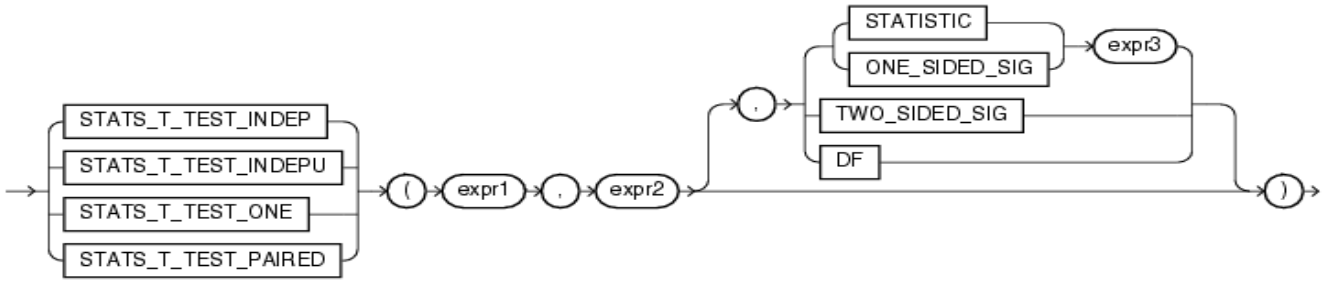
STATS_T_TEST_*

The t-test functions are:

- STATS_T_TEST_ONE: A one-sample t-test
- STATS_T_TEST_PAIRED: A two-sample, paired t-test (also known as a crossed t-test)
- STATS_T_TEST_INDEP: A t-test of two independent groups with the same variance (pooled variances)
- STATS_T_TEST_INDEPU: A t-test of two independent groups with unequal variance (unpooled variances)

Syntax

***stats_t_test*::=**



Description of the illustration stats_t_test.gif

Purpose

The t-test measures the significance of a difference of means. You can use it to compare the means of two groups or the means of one group with a constant. The one-sample and two-sample STATS_T_TEST_* functions take three arguments: two expressions and a return value of type VARCHAR2. The functions return one number, determined by the value of the third argument. If you omit the third argument, the default is TWO_SIDED_SIG. The meaning of the return values is shown in Table 5-9.

Table 5-9 STATS_T_TEST_* Return Values

Return Value	Meaning
STATISTIC	The observed value of *t*
DF	Degree of freedom

Hide Navigation

t-test

Search

This book Entire library

Quick Lookup

Advanced Search • Master Book List • Master Index • Master Glossary • Error Messages

Main Categories

- Installation
- Getting Started
- Administration
- Application Development
- Grid Computing
- High Availability
- Data Warehousing
- Content Management and Unstructured Data
- Information Integration
- Security
- Favorites

This Page [0 comments, Leave a comment]

- STATS_T_TEST_*
  - STATS_T_TEST_ONE
  - STATS_T_TEST_PAIRED
  - STATS_T_TEST_INDEP and STATS_T_TEST_INDEPU

This Book

Oracle® Database SQL Language Reference 11g Release 2 (11.2)

New and changed books:
HTML documents PDF files

STATS_T_TEST_ONE

In the STATS_T_TEST_ONE function, *expr1* is the sample and *expr2* is the constant mean against which the sample mean is compared. For this *t*-test onl optional; the constant mean defaults to 0. This function obtains the value of *t* by dividing the difference between the sample mean and the known mean by th error of the mean (rather than the standard error of the difference of the means, as for STATS_T_TEST_PAIRED).

STATS_T_TEST_ONE Example
The following example determines the significance of the difference between the average list price and the constant value 60:

```
SELECT AVG(prod_list_price) group_mean,
       STATS_T_TEST_ONE(prod_list_price, 60, 'STATISTIC') t_observed,
       STATS_T_TEST_ONE(prod_list_price, 60) two_sided_p_value
  FROM sh.products;

GROUP_MEAN T_OBSERVED TWO_SIDED_P_VALUE
---------- ---------- ------------------
139.545556 2.32107746         .023158537
```

STATS_T_TEST_PAIRED

In the STATS_T_TEST_PAIRED function, *expr1* and *expr2* are the two samples whose means are being compared. This function obtains the value of *t* by dividing the difference between the sample means by the standard error of the difference of the means (rather than the standard error of the mean, as for STATS_T_TEST_ONE).

STATS_T_TEST_INDEP and STATS_T_TEST_INDEPU

In the STATS_T_TEST_INDEP and STATS_T_TEST_INDEPU functions, *expr1* is the grouping column and *expr2* is the sample of values. The pooled variances version (STATS_T_TEST_INDEP) tests whether the means are the same or different for two distributions that have similar variances. The unpooled variances version (STATS_T_TEST_INDEPU) tests whether the means are the same or different even if the two distributions are known to have significantly different variances.

Before using these functions, it is advisable to determine whether the variances of the samples are significantly different. If they are, then the data may come from distributions with different shapes, and the difference of the means may not be very useful. You can perform an *F* test to determine the difference of the variances. If they are not significantly different, use STATS_T_TEST_INDEP. If they are significantly different, use STATS_T_TEST_INDEPU. Refer to STATS_F_TEST for information on performing an *F* test.

STATS_T_TEST_INDEP Example
The following example determines the significance of the difference between the average sales to men and women where the distributions are assumed to have similar (pooled) variances:

```
SELECT SUBSTR(cust_income_level, 1, 22) income_level,
       AVG(DECODE(cust_gender, 'M', amount_sold, null)) sold_to_men,
       AVG(DECODE(cust_gender, 'F', amount_sold, null)) sold_to_women,
       STATS_T_TEST_INDEP(cust_gender, amount_sold, 'STATISTIC', 'F') t_observed,
       STATS_T_TEST_INDEP(cust_gender, amount_sold) two_sided_p_value
  FROM sh.customers c, sh.sales s
  WHERE c.cust_id = s.cust_id
  GROUP BY ROLLUP(cust_income_level)
  ORDER BY income_level, sold_to_men, sold_to_women, t_observed;
```

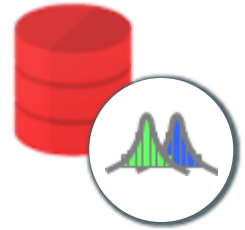
Split Lot A/B Offer testing

In-Database SQL t-test

- Offer "A" to one population and "B" to another
- Over time period "t" calculate median purchase amounts of customers receiving offer A & B
- Perform t-test to compare
- If statistically significantly better results achieved from one offer over another, offer everyone higher performing offer

DBMS_STAT_FUNCS Package

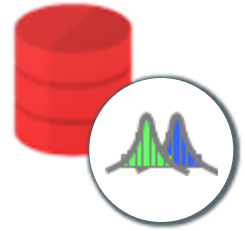
SUMMARY procedure



- SUMMARY procedure is summarize a numerical column (ADM_PULSE); the summary is returned as record of type summaryType

```
set echo off
connect CBERGER/CBERGER
set serveroutput on
set echo on
declare
  s DBMS_STAT_FUNCS.SummaryType;
begin
  DBMS_STAT_FUNCS.SUMMARY('CBERGER','LYMPHOMA','ADM_PULSE',3,s);
  dbms_output.put_line('SUMMARY STATISTICS');
  dbms_output.put_line('Count:     '||s.count);
  dbms_output.put_line('Min:'||s.min);
  dbms_output.put_line('Max:       '||s.max);
  dbms_output.put_line('Range:     '||s.range);
  dbms_output.put_line('Mean:      '||round(s.mean));
  dbms_output.put_line('Mode Count:  '||s.cmode.count);
  dbms_output.put_line('Mode:            '||s.cmode(1));
  dbms_output.put_line('Variance:'||round(s.variance));
  dbms_output.put_line('Stddev:          '||round(s.stddev));
  dbms_output.put_line('Quantile 5       '||s.quantile_5);
  dbms_output.put_line('Quantile 25      '||s.quantile_25);
  dbms_output.put_line('Median           '||s.median);
  dbms_output.put_line('Quantile 75      '||s.quantile_75);
  dbms_output.put_line('Quantile 95      '||s.quantile_95);
  dbms_output.put_line('Extreme Count:       '||s.extreme_values.count);
  dbms_output.put_line('Extremes:        '||s.extreme_values(1));
  dbms_output.put_line('Top 3:           '||s.top_5_values(1)||','||s.top_5_values(2)||','||s.top_5_values(3));
  dbms_output.put_line('Bottom 3:        '||s.bottom_5_values(5)||','||s.bottom_5_values(4)||','||s.bottom_5_values(3));
end;
/
```

One-Sample T-Test



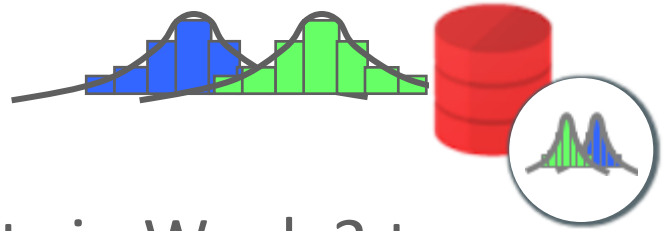
- Query compares the mean of SURVIVAL_TIME to the assumed value of 35:

```
SELECT avg(SURVIVAL_TIME_MO) group_mean,
stats_t_test_one(SURVIVAL_TIME_MO, 35,
'STATISTIC') t_observed,
stats_t_test_one(SURVIVAL_TIME_MO, 35)
two_sided_p_value
FROM LYMPHOMA;
```

- Returns the observed t value and its related two-sided significance

SQL Worksheet

Paired Samples T-Test



- Query compares the mean of LOGWT for Pig Weights in Week 3 to Week 8, grouped by Diet:

```
SELECT substr(diet,1,1) as diet, avg(LOGWT3) logwt3_mean,
avg(LOGWT8) logwt8_mean,
stats_t_test_paired(LOGWT3, LOGWT8,'STATISTIC') t_observed,
stats_t_test_paired(LOGWT3, LOGWT8) two_sided_p_value
FROM CBERGER.PIGLETS3
GROUP BY ROLLUP(DIET)
ORDER BY 5 ASC;
```

- Returns the observed t value and its related two-sided significance

SQL Worksheet

F-Test

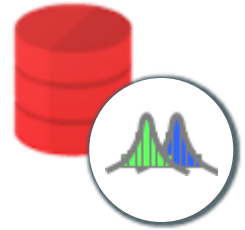
Query compares the variance in the SIZE_TUMOR between MALES and FEMALES

```
SELECT variance(decode(GENDER,'0', SIZE_TUMOR_MM, null))
var_tumor_men,
   variance(decode(GENDER,'1', SIZE_TUMOR_MM,null)) var_tumor_women,
   stats_f_test(GENDER, SIZE_TUMOR_MM, 'STATISTIC', '1') f_statistic,
   stats_f_test(GENDER, SIZE_TUMOR_MM) two_sided_p_value
 FROM DMUSER.LYMPHOMA;
```

- Returns observed f value and two-sided significance

SQL Worksheet

F-Test



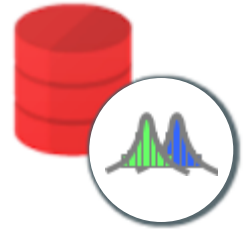
- Query compares the variance in the SIZE_TUMOR between males and females Grouped By GENDER

```
SELECT GENDER,
     stats_one_way_anova(TREATMENT_PLAN,
     SIZE_REDUCTION,'F_RATIO') f_ratio,
     stats_one_way_anova(TREATMENT_PLAN,
     SIZE_REDUCTION,'SIG') p_value, AVG(SIZE_REDUCTION)
     FROM CBERGER.LYMPHOMA
GROUP BY GENDER ORDER BY GENDER;
```

- Returns observed f value and two-sided significance

SQL Worksheet

One-Way ANOVA

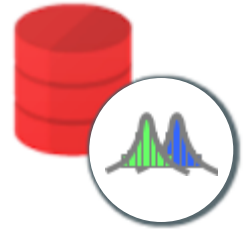


- Query compares the average SIZE_REDUCTION within different TREATMENT_PLANS Grouped By LYMPH_TYPE:

```
SELECT LYMPH_TYPE,
  stats_one_way_anova(TREATMENT_PLAN,
  SIZE_REDUCTION,'F_RATIO') f_ratio,
  stats_one_way_anova(TREATMENT_PLAN,
  SIZE_REDUCTION,'SIG') p_value
  FROM DMUSER.LYMPHOMA
GROUP BY LYMPH_TYPE ORDER BY 1;
```

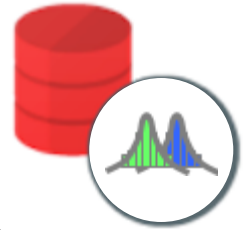
- Returns one-way ANOVA significance and split by LYMPH_TYPE

Hypothesis Testing
Nonparametric



- Nonparametric tests are used when certain assumptions about the data are questionable.
- This may include the difference between samples that are not normally distributed.
- All tests involving ordinal scales (in which data is ranked) are nonparametric.
- Nonparametric tests supported in Oracle Database 10g:
  - Binomial test
  - Wilcoxon Signed Ranks test
  - Mann-Whitney test
  - Kolmogorov-Smirnov test

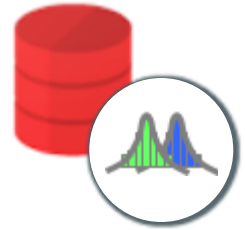
Customer Example



- "..Our experience suggests that Oracle Statistics and Data Mining features can reduce development effort of analytical systems by an order of magnitude."
  - Sumeet Muju, Senior Member of Professional Staff, SRA International (SRA supports NIH bioinformatics development projects)

```
248 rows selected.

SQL> select peak_id peak,avg(decode(E.sample_group,'CNS',s.intensity,null)) avg_CNS, avg(decode(E.sa
mple_group,'ND',s.intensity,null)) avg_ND, stats_ks_test(E.sample_group,s.intensity,'STATISTIC') ks_
stat, stats_ks_test(E.sample_group,s.intensity) ks_p_value,  stats_t_test_indep(E.sample_group,s.int
ensity) t_test_p_value, avg(subs_mass) AVG_MASS from exp_descriptor E, celd_spectrum s where E.exp_i
d = s.exp_id and E.chip_id = s.chip_id and E.spot_number = s.spot_number and (sample_group = 'CNS' o
r sample_group='ND')  Group By peak_id order by  stats_t_test_indep(E.sample_group,s.intensity);

      PEAK    AVG_CNS     AVG_ND    KS_STAT KS_P_VALUE T_TEST_P_VALUE   AVG_MASS
---------- ---------- ---------- ---------- ---------- --------------- ----------
       178  1.3314339 2.17817187 .673333333 7.2556E-16     2.6544E-17 5952.91674
       181  5.0996028 7.89194275 .626666667 8.4480E-14     1.4848E-14  6075.9581
       180 2.27649538 3.47917519 .606666667 5.8453E-13     8.8539E-14 6055.14643
       182 1.82166302 2.70982458 .586666667 3.7986E-12     1.7684E-13 6093.52256
       112 1.43756807 .415726202          .6 1.0984E-12     4.5081E-13 4033.66603
       179 .470304995  .71366692 .546666667 1.3289E-10     6.0678E-13 5976.18528
       162  .32065549 .488111947 .606666667 5.8453E-13     6.3174E-13 5384.91078
       176 1.71447936 2.70554235 .553333333 7.4775E-11     1.7747E-12 5914.53224
       185 .336895407 .472142857         .55 9.9772E-11     1.9222E-12 6260.71013
       186 .401995708 .562915017 .506666667 3.6175E-09     2.1445E-12 6281.69466
       177  2.3623861 3.80160199 .586666667 3.7986E-12     4.1808E-12 5933.83033
```

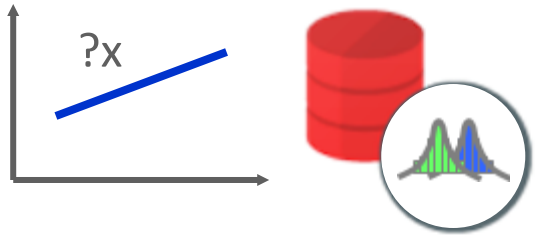
Correlation Functions



- The CORR_S and CORR_K functions support nonparametric or rank correlation (finding correlations between expressions that are ordinal scaled).
- Correlation coefficients take on a value ranging from –1 to 1, where:
  - 1 indicates a perfect relationship
  - –1 indicates a perfect inverse relationship
  - 0 indicates no relationship
- The following query determines whether there is a correlation between the AGE and WEIGHT of people, using Spearman's correlation:

```
select CORR_S(AGE, WEIGHT)
    coefficient,
  CORR_S(AGE, WEIGHT,
    'TWO_SIDED_SIG')
  p_value, substr(TREATMENT_PLAN,
  1,15) as TREATMENT_PLAN
from DMUSER.LYMPHOMA
GROUP BY TREATMENT_PLAN;
```

Correlation Functions



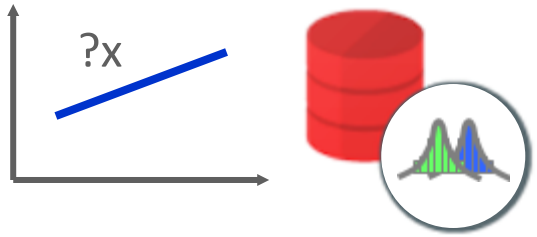
- Procedure to find correlation coefficients for all attributes vs. all attributes

```
-- Procedure for creating a correlation table
-- Parameters:
-- p_in_table  - name of the input data table
-- p_out_table - name of the output table
-- p_type      - correlation type ('P' - Pearson - default
--                                 'S' - Spearman's rho
--                                 'K' - Kendall's tau-b)
-- p_compact   - output type (1 - compact output, triangular matrix, default
--                            0 - full matrix)
--
-- Usage:
-- 1) Uses Pearson correlation and compact output, saves results to OUTTAB
-- DROP TABLE OUTTAB PURGE;
-- BEGIN
--   mcorr('EMP','OUTTAB');
-- END;
-- /
-- SELECT * FROM outtab ORDER BY col1, col2;
--
-- 2) Uses Spearman's rho and full matrix output, saves results to OUTTAB
```

```
DROP TABLE OUTTAB PURGE;
BEGIN
  mcorr('CUST_INSUR_LTV','OUTTAB');
 END;
/
```

```
select * from outtab order by correlation desc;
```

Correlation Functions

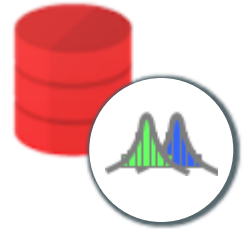



```
DROP TABLE OUTTAB PURGE;
BEGIN
  mcorr('CUST_INSUR_LTV','OUTTAB');
 END;
/
```

```
select * from outtab order by
correlation desc;
```

```
-- DROP TABLE OUTTAB PURGE;
-- BEGIN
--   mcorr('EMP','OUTTAB','S',0);
-- END;
-- /
-- SELECT * FROM outtab ORDER BY col1, col2;
--

CREATE OR REPLACE PROCEDURE mcorr(p_in_table   VARCHAR2,
                        p_out_table  VARCHAR2,
                        p_type       VARCHAR2 DEFAULT 'P',
                        p_compact    NUMBER DEFAULT 1) AS
 TYPE Char_Tab    IS TABLE OF VARCHAR2(30);
 v_col_names     Char_Tab;
 v_stmt          VARCHAR2(4000);
 v_stmt1         VARCHAR2(4000);
 v_corr          NUMBER;
 v_corr_str      VARCHAR2(6) := 'CORR';
BEGIN
 IF (p_type ='S') THEN
  v_corr_str := 'CORR_S';
 ELSE
  IF (p_type = 'K') THEN
   v_corr_str := 'CORR_K';
  END IF;
 END IF;

 -- get list of columns
 v_stmt := 'SELECT column_name FROM user_tab_columns ' ||
        'WHERE data_type = ''NUMBER'' AND ' ||
           ' table_name = ''' || p_in_table || '''';
 EXECUTE IMMEDIATE v_stmt BULK COLLECT INTO v_col_names;

 -- create output table
 v_stmt := 'CREATE TABLE ' || p_out_table ||
        '(col1 VARCHAR2(30), col2 VARCHAR2(30), correlation NUMBER)';
 EXECUTE IMMEDIATE v_stmt;

 -- compute correlation and insert into output table
 v_stmt:='INSERT INTO ' ||  p_out_table ||
       ' (col1, col2, correlation) VALUES(:v1, :v2, :v3)';

 FOR i IN 1..v_col_names.count LOOP
  EXECUTE IMMEDIATE v_stmt using v_col_names(i), v_col_names(i), 1.0;
  FOR j IN (i+1)..v_col_names.count LOOP
   v_stmt1 :='SELECT ' || v_corr_str || '(' ||
            v_col_names(i) || ',' ||
            v_col_names(j) || ') ' ||
           'FROM ' || p_in_table;
   EXECUTE IMMEDIATE v_stmt1 INTO v_corr;
   EXECUTE IMMEDIATE v_stmt using v_col_names(i), v_col_names(j), v_corr;
   IF (p_compact = 0) THEN
    EXECUTE IMMEDIATE v_stmt using v_col_names(j), v_col_names(i), v_corr;
   END IF;
  END LOOP;
 END LOOP;
END;
/
SHOW ERRORS;
```

Cross Tabulations



- This query analyzes the strength of the association between TREATMENT_PLAN and GENDER Grouped By LYMPH_TYPE using a cross tabulation:

```
SELECT LYMPH_TYPE,
stats_crosstab(GENDER, TREATMENT_PLAN,
   'CHISQ_OBS') chi_squared,
   stats_crosstab(GENDER, TREATMENT_PLAN,
   'CHISQ_SIG') p_value,
   stats_crosstab(GENDER, TREATMENT_PLAN,
   'PHI_COEFFICIENT') phi_coefficient
FROM CBERGER.LYMPHOMA
 GROUP BY LYMPH_TYPE ORDER BY 1;
```

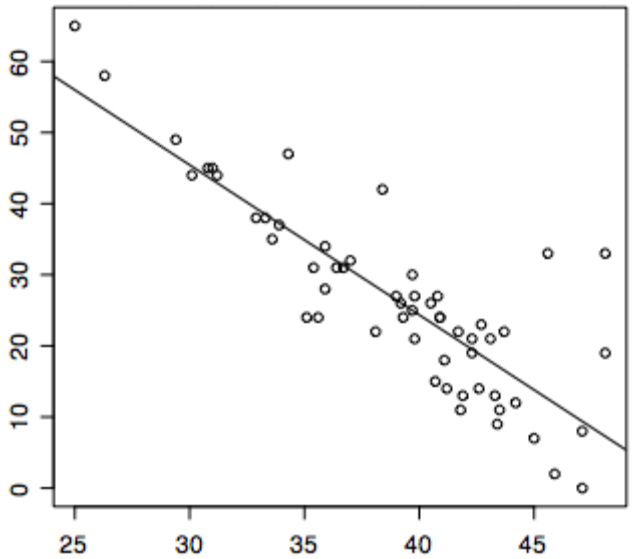
- Returns the observed p_value and phi coefficient significance:

Cross Tabulations

- STATS_CROSSTAB function takes as arguments two expressions (the two variables being analyzed) and a value that determines which test to perform. These values include the following:
  - CHISQ_OBS (observed value of chi-squared)
  - CHISQ_SIG (significance of observed chi-squared)
  - CHISQ_DF (degree of freedom for chi-squared)
  - PHI_COEFFICIENT (phi coefficient)
  - CRAMERS_V (Cramer's V statistic)
  - CONT_COEFFICIENT (contingency coefficient)
  - COHENS_K (Cohen's kappa)
- Function returns all values as specified by the third argument (default is CHISQ_SIG)

Linear Regression

Quantity Sold



Price

Excerpted from Rob Rolek, BIWA TechCast presentation “Lies, Damned Lies and SQL Statistical Functions”, rolekr@tusc.com

Linear Regressions

Multiple Simple Linear Regression Fits

```
SELECT
s.channel_id,
REGR_SLOPE(s.quantity_sold, p.prod_list_price)  SLOPE ,
REGR_INTERCEPT(s.quantity_sold, p.prod_list_price)  INTCPT ,
REGR_R2(s.quantity_sold, p.prod_list_price)  RSQR ,
REGR_COUNT(s.quantity_sold, p.prod_list_price)  COUNT ,
REGR_AVGX(s.quantity_sold, p.prod_list_price)  AVGLISTP ,
REGR_AVGY(s.quantity_sold, p.prod_list_price)  AVGQSOLD
FROM  sales s, products p
WHERE s.prod_id=p.prod_id AND
p.prod_category='Men'  AND
s.time_id=to_DATE('10-OCT-2000')
GROUP BY s.channel_id;
```

C	SLOPE	INTCPT	RSQR	COUNT	AVGLISTP	AVGQSOLD
C	-.03529838	16.4548382	.217277422	17	87.8764706	13.3529412
I	-.0108044	13.3082392	.028398018	43	116.77907	12.0465116
P	-.01729665	11.3634927	.026191191	33	80.5818182	9.96969697
S	-.01277499	13.488506	.000473089	71	52.571831	12.8169014
T	-.01026734	5.01019929	.064283727	21	75.2	4.23809524

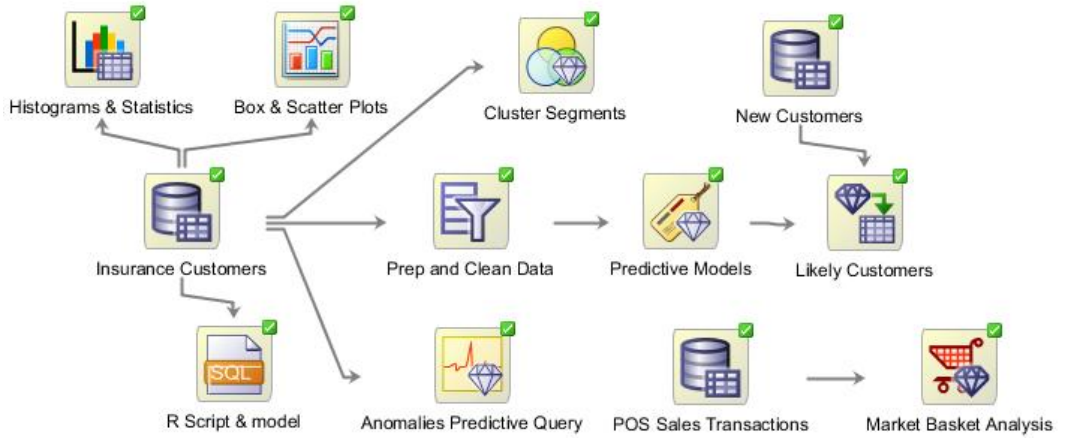
Excerpted from Rob Rolek, BIWA TechCast presentation “Lies, Damned Lies and SQL Statistical Functions”, rolekr@tusc.com

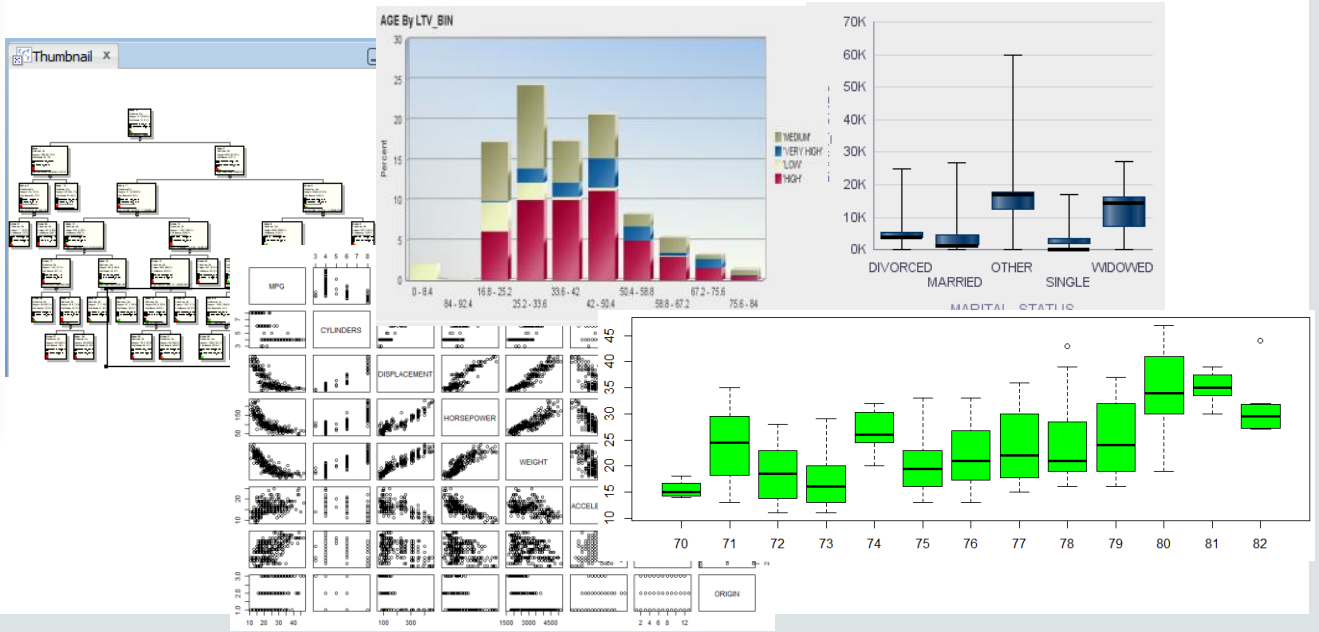
Oracle Advanced Analytics Database Option

Fastest Way to Deliver Scalable Enterprise-wide Predictive Analytics

Key Features

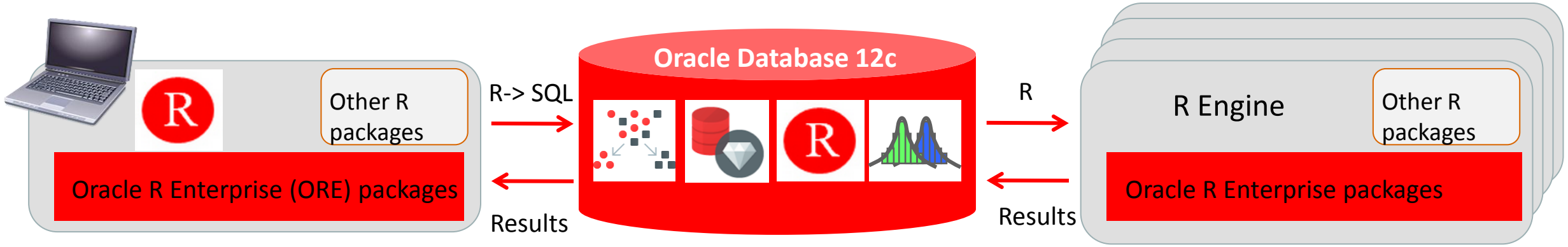
- Scalable in-database data mining algorithms and R integration
- Powerful predictive analytics and deployment platform
- Drag and drop workflow, R and SQL APIs
- Data analysts, data scientists & developers
- Enables enterprise predictive analytics applications





Oracle Advanced Analytics

How Oracle R Enterprise Compute Engines Work



1 R-> SQL Transparency "Push-Down"

- R language for interaction with the database
- R-SQL Transparency Framework overloads R functions for scalable in-database execution
- Function overload for data selection, manipulation and transforms
- Interactive display of graphical results and flow control as in standard R
- Submit user-defined R functions for execution at database server under control of Oracle Database

2 In-Database Adv Analytical SQL Functions

- 15+ Powerful data mining algorithms (regression, clustering, AR, DT, etc._
- Run Oracle Data Mining SQL data mining functioning (ORE.odmSVM, ORE.odmDT, etc.)
- Speak "R" but executes as proprietary in-database SQL functions—machine learning algorithms and statistical functions
- Leverage database strengths: SQL parallelism, scale to large datasets, security
- Access big data in Database and Hadoop via SQL, R, and Big Data SQL

3 Embedded R Package Callouts

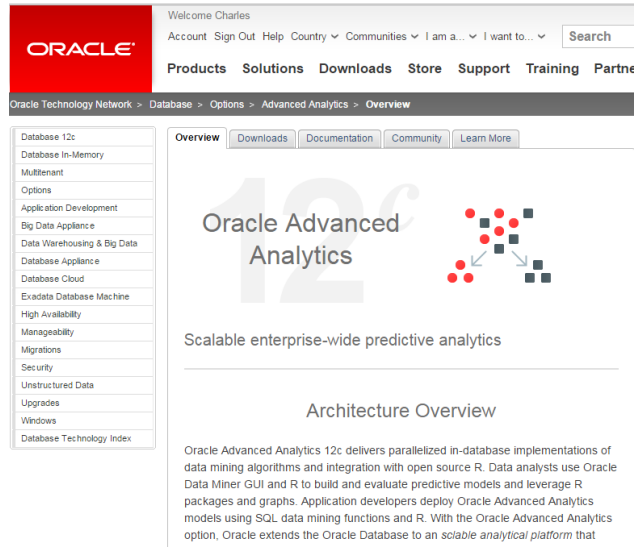
- R Engine(s) spawned by Oracle DB for database-managed parallelism
- ore.groupApply high performance scoring
- Efficient data transfer to spawned R engines
- Emulate map-reduce style algorithms and applications
- Enables production deployment and automated execution of R scripts

ORACLE

Getting started

OAA Links and Resources

- **Oracle Advanced Analytics Overview:**
  - **OAA presentation—** Big Data Analytics in Oracle Database 12c With **Oracle Advanced Analytics** & Big Data SQL
  - **Big Data Analytics with Oracle Advanced Analytics: Making Big Data and Analytics Simple white paper** on OTN
  - *Oracle Internal* OAA Product Management Wiki and Workspace
- ***YouTube* recorded OAA Presentations and Demos:**
  - **Oracle Advanced Analytics and Data Mining at the YouTube Movies**
    (6 + OAA "live" Demos on ODM'r 4.0 New Features, Retail, Fraud, Loyalty, Overview, etc.)
- **Getting Started:**
  - Link to **Getting Started w/ ODM blog entry**
  - Link to New OAA/Oracle Data Mining 2-Day Instructor Led Oracle University course.
  - Link to **OAA/Oracle Data Mining 4.0 Oracle by Examples (free) Tutorials** on OTN
  - Take a Free Test Drive of Oracle Advanced Analytics (Oracle Data Miner GUI) on the Amazon Cloud
  - Link to **OAA/Oracle R Enterprise (free) Tutorial Series** on OTN
- **Additional Resources:**
  - Oracle Advanced Analytics Option on OTN page
  - **OAA/Oracle Data Mining on OTN** page, ODM Documentation & ODM Blog
  - **OAA/Oracle R Enterprise page on OTN** page, ORE Documentation & ORE Blog
  - Oracle SQL based Basic Statistical functions on OTN
  - BIWA Summit'16, Jan 26-28, 2016 – Oracle Big Data & Analytics User Conference @ Oracle HQ Conference Center



BIWA SUMMIT 2016

The Oracle Big Data + Analytics User Conference
January 26-28, 2016
Including Oracle Spatial Summit

Home | Abstract Submission | Sponsorship | Hotel and Travel | Registration Pricing | Registration

BIWA SIG
Better Information Better Results
www.oraclebiwa.org

ORACLE SPATIAL AND GRAPH SPECIAL INTEREST GROUP (SIG)

NoCOUG — Northern California Oracle Users Group

Publicity

- Oracle Business Analytics Newsletter
- DB Insider Dec 2014
- Oracle Magazine
- Latest BIWA SIG Blog Entry
- Jeff Shauer Blog Entry
- Daily BIWA Newsletter
- Email to BIWA members
- Real Time BI Webcast
- Oracle Events Calendar
- Oracle ACE Newsletter
- DB Insider Jan 2015 with Spatial Summit
- Lots of other emails

January 26-28, 2016

Oracle Conference Center at Oracle HQ Campus, Redwood Shores, CA

- **Hands-on-Labs**
- **Customer stories, told by the customers**
- **Educational sessions by Practitioners and Direct from Developers**
- **Oracle Keynote presentations**
- **Presentations covering: Advanced Analytics, Big Data, Business Intelligence, Cloud, Data Warehousing and Integration, Spatial and Graph, SQL**
- **Networking with product management and development professionals**

ORACLE


ORACLE®