

QuickStart: Oracle Statistics

Release 10gR2 Charlie.berger@oracle.com February 2006

Statistical Analysis of Lymphoma

Oracle Release 10g added support for a range of statistical functions that help generate better and more useful information.

Know More

Oracle's Statistics help to perform descriptive statistics, hypothesis testing, correlation analysis, and other statistical functions on data that reside in the Oracle Database.

Do More

With Oracle's embedded statistical functions, you can extract more information without costly materialization or extraction of the data to special statistical packages.

Spend Less

Oracle's statistical functions are included for FREE in the Oracle SE & EE Editions of the Oracle 10g Release 1 and 2 Database. Thus, the total cost of ownership is dramatically less than the other major competitors.

```
Command Prompt - sqlplus cberger/cberger@ora10gr2
SQL>
SQL>
SQL> SELECT GENDER,
2 stats_one_way_anova(TREATMENT_PLAN,
3 SIZE_REDUCTION,'F_RATIO') f_ratio,
4 stats_one_way_anova(TREATMENT_PLAN,
5 SIZE_REDUCTION,'SIG') p_value, AVG(SIZE_REDUCTION)
6 FROM CBERGER.LYMPHOMA
7 GROUP BY GENDER ORDER BY 1;

GENDER
-----
F_RATIO    P_VALUE  AVG(SIZE_REDUCTION)
-----
0
10.8492925 4.6240E-07          1.31785714
1
23.43483  1.6280E-14          1.3617801
```

Data Summarization and Descriptive Statistics

1. Medical researchers want to statistically analyze LYMPHOMA data about cancer patients, demographic information, laboratory results, surgical data, and medical treatment and outcomes data. They want to summarize the data and run some statistical tests to find the factors most associated with patients that survive lymphoma. They also want to run some other statistical analyses on some other life and health sciences data.
2. Using LYMPHOMA data, lets take a look at the median value of AGE.

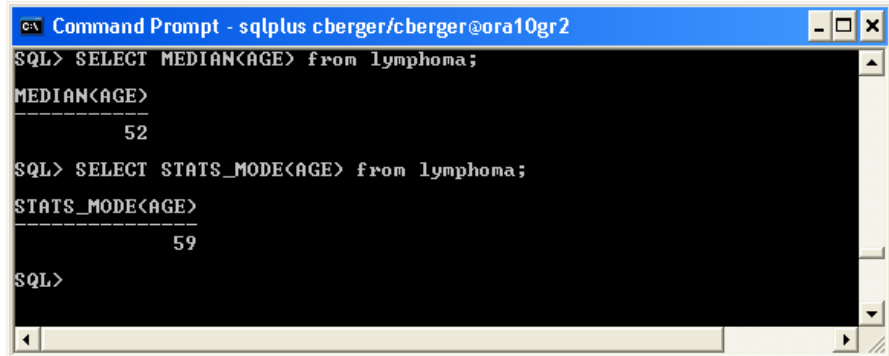
SQL> SELECT MEDIAN(AGE) from LYMPHOMA;

```
Command Prompt - sqlplus cberger/cberger@ora10gr2
SQL> SELECT MEDIAN(AGE) from lymphoma;

MEDIAN(AGE)
-----
          52

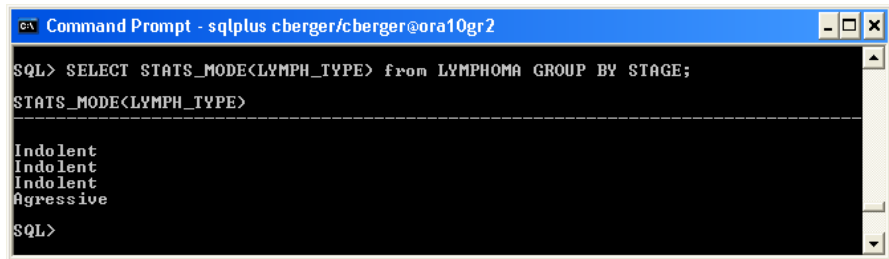
SQL> _
```

3. Let's see what the statistical mode of AGE is:
SQL> SELECT STATS_MODE(AGE) from LYMPHOMA;



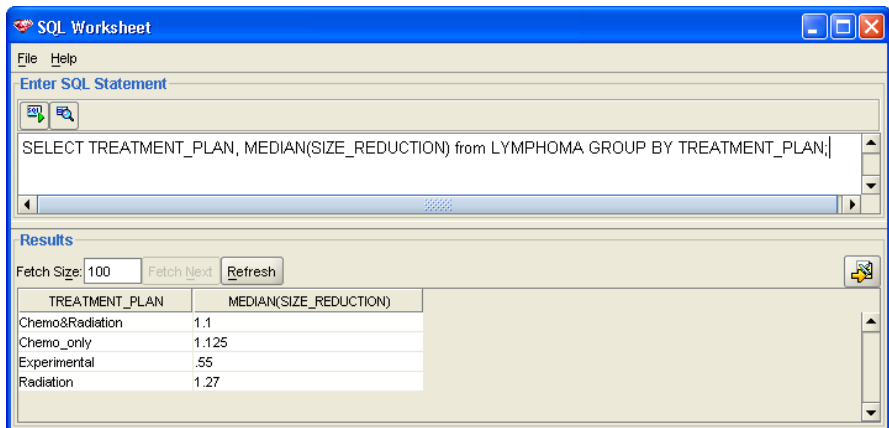
```
Command Prompt - sqlplus cberger/cberger@ora10gr2
SQL> SELECT MEDIAN(AGE) from Lymphoma;
MEDIAN(AGE)
-----
          52
SQL> SELECT STATS_MODE(AGE) from Lymphoma;
STATS_MODE(AGE)
-----
          59
SQL>
```

4. Using the *Group By* clause, let's do that again:
SQL> SELECT STATS_MODE(LYMPH_TYPE) from LYMPHOMA GROUP BY STAGE;



```
Command Prompt - sqlplus cberger/cberger@ora10gr2
SQL> SELECT STATS_MODE(LYMPH_TYPE) from LYMPHOMA GROUP BY STAGE;
STATS_MODE(LYMPH_TYPE)
-----
Indolent
Indolent
Indolent
Agressive
SQL>
```

5. Using the SQL Worksheet provide more formatting support for the following GROUP BY query:
SQL> SELECT TREATMENT_PLAN, MEDIAN(SIZE_REDUCTION) from LYMPHOMA GROUP BY TREATMENT_PLAN;



```
SQL Worksheet
File Help
Enter SQL Statement
SELECT TREATMENT_PLAN, MEDIAN(SIZE_REDUCTION) from LYMPHOMA GROUP BY TREATMENT_PLAN;
Results
Fetch Size: 100 Fetch Next Refresh
TREATMENT_PLAN  MEDIAN(SIZE_REDUCTION)
-----
Chemo&Radiation  1.1
Chemo_only       1.125
Experimental     .55
Radiation        1.27
```

Oracle's statistical functions can be used to automatically analyze data and be included within automated "analytical pipelines".

6. The **Summarize DBMS_STATS_SUMMARIZE** PL/SQL procedure summarizes all the basic descriptive statistics for an attribute or attributes.

SQL>

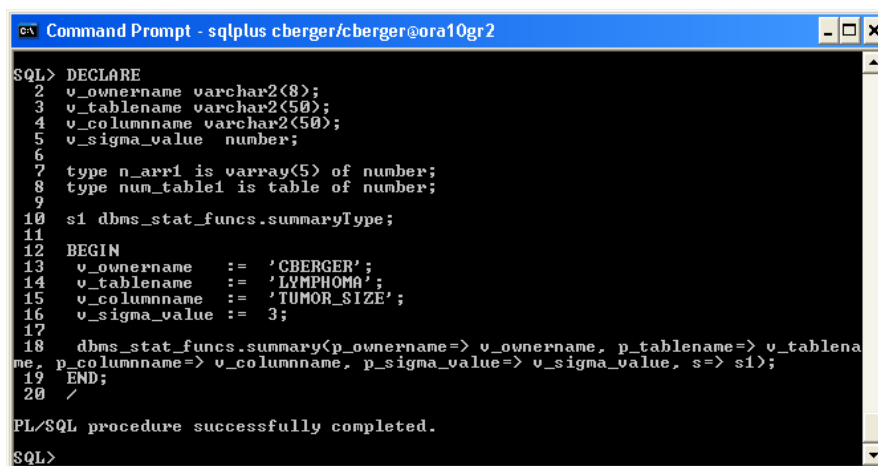
```
DECLARE
v_ownershipname varchar2(8);
v_tablename varchar2(50);
v_columnname varchar2(50);
v_sigma_value number;

type n_arr1 is varray(5) of number;
type num_table1 is table of number;

s1 dbms_stat_funcs.summaryType;

BEGIN
v_ownershipname := 'OLSUG';
v_tablename := 'LYMPHOMA';
v_columnname := 'SIZE_TUMOR_MM';
v_sigma_value := 3;

dbms_stat_funcs.summary(p_ownershipname=>
v_ownershipname, p_tablename=> v_tablename,
p_columnname=> v_columnname, p_sigma_value=>
v_sigma_value, s=> s1);
END;
/
```



```
Command Prompt - sqlplus cberger/cberger@ora10gr2
SQL> DECLARE
2 v_ownershipname varchar2(8);
3 v_tablename varchar2(50);
4 v_columnname varchar2(50);
5 v_sigma_value number;
6
7 type n_arr1 is varray(5) of number;
8 type num_table1 is table of number;
9
10 s1 dbms_stat_funcs.summaryType;
11
12 BEGIN
13 v_ownershipname := 'CBERGER';
14 v_tablename := 'LYMPHOMA';
15 v_columnname := 'TUMOR_SIZE';
16 v_sigma_value := 3;
17
18 dbms_stat_funcs.summary(p_ownershipname=> v_ownershipname, p_tablename=> v_tablename,
19 p_columnname=> v_columnname, p_sigma_value=> v_sigma_value, s=> s1);
20 END;
21 /
PL/SQL procedure successfully completed.
SQL>
```

View the results through an SQL query.

SQL>

```
set echo off
connect OLSUG/OLSUG
```

```

set serveroutput on
set echo on
declare
s DBMS_STAT_FUNCS.SummaryType;
begin
DBMS_STAT_FUNCS.SUMMARY('OLSUG','LYMPHOMA','ADM_PULS
E',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;
/

```

```

Command Prompt - sqlplus cberger/cberger
SUMMARY STATISTICS
Count: 2500
Min: 47
Max: 168
Range: 121
Mean: 82
Mode Count: 1
Mode: 80
Variance: 244
Stddev: 16
Quantile 5: 60
Quantile 25: 70
Median: 80
Quantile 75: 91
Quantile 95: 109
Extreme Count: 8
Extremes: 168
Top 3: 168,168,168
Bottom 3: 47,47,47

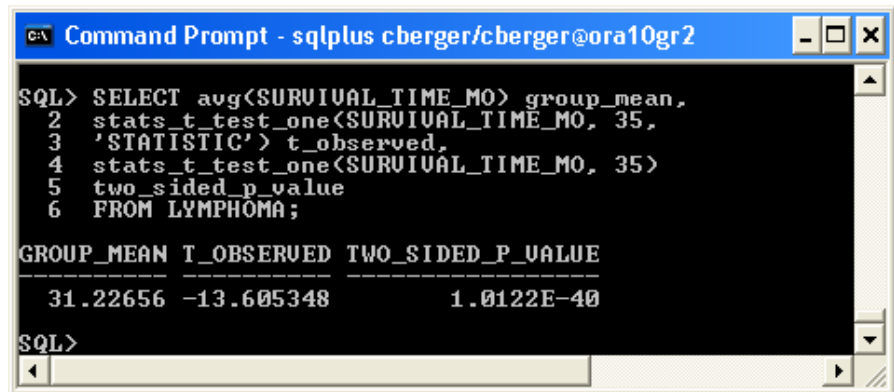
PL/SQL procedure successfully completed.
SQL>

```

Hypothesis Testing T-TEST

7. Let's compare whether the average survival time for Lymphoma patients is equal to 35 months.

```
SQL>
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;
```



```
CA Command Prompt - sqlplus cberger/cberger@ora10gr2
SQL> SELECT avg(SURVIVAL_TIME_MO) group_mean,
2 stats_t_test_one(SURVIVAL_TIME_MO, 35,
3 'STATISTIC') t_observed,
4 stats_t_test_one(SURVIVAL_TIME_MO, 35)
5 two_sided_p_value
6 FROM LYMPHOMA;

GROUP_MEAN T_OBSERVED TWO_SIDED_P_VALUE
-----
31.22656 -13.605348 1.0122E-40
SQL>
```

8. Now, using the PIGLETS3 data, let's compare whether two different diets have a significant impact on pig weight over time.

```
SQL>
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 OLSUG.PIGLETS3
GROUP BY ROLLUP(DIET)
ORDER BY 5 ASC;
```

```

c:\ Command Prompt - sqlplus cberger/cberger@ora10gr2
SQL> SELECT substr(diet,1,1) as diet, avg(LOGWT3) logwt3_mean,
2 avg(LOGWT8) logwt8_mean,
3 stats_t_test_paired(LOGWT3, LOGWT8, 'STATISTIC') t_observed,
4 stats_t_test_paired(LOGWT3, LOGWT8) two_sided_p_value
5 FROM CBERGER.PIGLET$3
6 GROUP BY ROLLUP(DIET)
7 ORDER BY 5 ASC;

```

D	LOGWT3_MEAN	LOGWT8_MEAN	T_OBSERVED	TWO_SIDED_P_VALUE
	.721463	1.607227	-51.043742	2.1343E-12
1	.759226	1.654972	-40.845666	.000002147
2	.6837	1.559482	-30.503204	6.8812E-06

```

SQL>

```

Independent Samples T-Test (Pooled Variances) Example

- This next example demonstrates (*using a non life sciences example*) compares the mean of amount sold between men and women using the SH Common Schema data that ships with the Oracle Database.

```

SQL>
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') 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;

```

```

C:\ Command Prompt - sqlplus cberger/cberger@ora10gr2
SQL> SELECT substr(cust_income_level,1,22) income_level,
2         avg(decode(cust_gender,'M',amount_sold,null))
3         sold_to_men,
4         avg(decode(cust_gender,'F',amount_sold,null))
5         sold_to_women,
6         stats_t_test_indep(cust_gender, amount_sold,
7         'STATISTIC') t_observed,
8         stats_t_test_indep(cust_gender, amount_sold)
9         two_sided_p_value
10        FROM sh.customers c, sh.sales s
11        WHERE c.cust_id=s.cust_id
12        GROUP BY rollup(cust_income_level)
13        ORDER BY 1;

```

INCOME_LEVEL	SOLD_TO_MEN	SOLD_TO_WOMEN	T_OBSERVED	TWO_SIDED_P_VALUE
A: Below 30,000	105.28349	99.4281447	-1.9880629	.046811482
B: 30,000 - 49,999	102.59651	109.829642	3.04330875	.002341053
C: 50,000 - 69,999	105.627588	110.127931	2.36148671	.018204221
D: 70,000 - 89,999	106.630299	110.47287	2.28496443	.022316997
E: 90,000 - 109,999	103.396741	101.610416	-1.2544577	.209677823
F: 110,000 - 129,999	106.76476	105.981312	-.60444998	.545545304
G: 130,000 - 149,999	108.872532	107.31377	-.85298245	.393671218
H: 150,000 - 169,999	110.987258	107.152191	-1.9062363	.056622983
I: 170,000 - 189,999	102.808238	107.43556	2.18477851	.028908566
J: 190,000 - 249,999	108.040564	115.343356	2.58313425	.009794516
K: 250,000 - 299,999	112.377993	108.196097	-1.4107871	.158316973
INCOME_LEVEL	SOLD_TO_MEN	SOLD_TO_WOMEN	T_OBSERVED	TWO_SIDED_P_VALUE
L: 300,000 and above	120.970235	112.216342	-2.0642868	.039003862
	107.121845	113.80441	.686144393	.492670059
	106.663769	107.276386	1.08013499	.280082357

```

14 rows selected.
SQL>

```

Hypothesis Testing F-TEST Example

10. This query compares the distribution of SIZE_TUMOR of men to women.

```

SQL> 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') f_statistic,
stats_f_test(GENDER, SIZE_TUMOR_MM)
two_sided_p_value
FROM OLSUG.LYMPHOMA;

```

```

C:\ Command Prompt - sqlplus cberger/cberger@ora10gr2
SQL> SELECT variance(decode(GENDER,'0',
2 SIZE_TUMOR_MM,null)) var_tumor_men,
3 variance(decode(GENDER,'1',
4 SIZE_TUMOR_MM,null)) var_tumor_women,
5 stats_f_test(GENDER, SIZE_TUMOR_MM,
6 'STATISTIC') f_statistic, stats_f_test(GENDER, SIZE_TUMOR_MM)
7 two_sided_p_value
8 FROM CBERGER.LYMPHOMA;

```

VAR_TUMOR_MEN	VAR_TUMOR_WOMEN	F_STATISTIC	TWO_SIDED_P_VALUE
1661682.34	2519391.83	.65955693	3.7514E-12

```

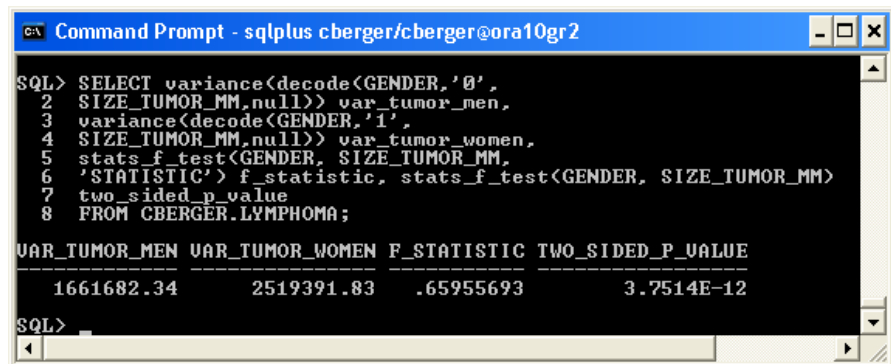
SQL>

```

Hypothesis Testing ONE-WAY ANOVA

11. This query compares the SIZE_REDUCTION between TREATMENT_PLANS using a One-Way ANOVA and returns one-way ANOVA significance and splits this on a per-gender basis

```
SQL> 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 OLSUG.LYMPHOMA  
GROUP BY LYMPH_TYPE ORDER BY 1;
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt - sqlplus cberger/cberger@ora10gr2". The window contains the following SQL query and its output:

```
SQL> SELECT variance(decode(GENDER,'0',  
2 SIZE_TUMOR_MM,null)) var_tumor_men,  
3 variance(decode(GENDER,'1',  
4 SIZE_TUMOR_MM,null)) var_tumor_women,  
5 stats_f_test(GENDER, SIZE_TUMOR_MM,  
6 'STATISTIC') f_statistic, stats_f_test(GENDER, SIZE_TUMOR_MM)  
7 two_sided_p_value  
8 FROM CBERGER.LYMPHOMA;
```

VAR_TUMOR_MEN	VAR_TUMOR_WOMEN	F_STATISTIC	TWO_SIDED_P_VALUE
1661682.34	2519391.83	.65955693	3.7514E-12

The prompt ends with "SQL>".

Correlation Tests

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

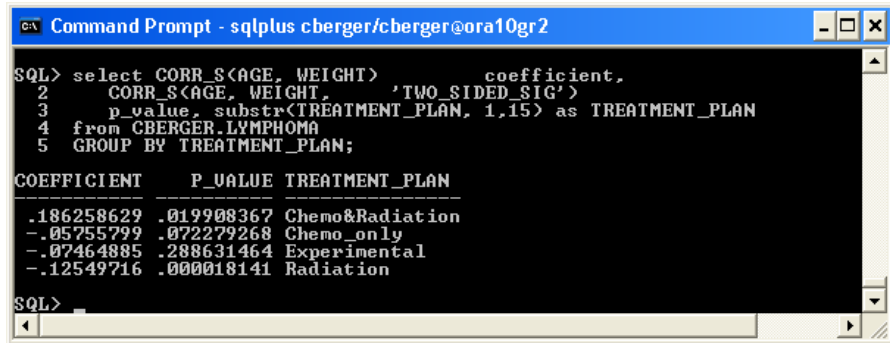
-1 indicates a perfect inverse relationship

0 indicates no relationship

The following query determines whether there is a correlation between the AGE and WEIGHT of people, using Spearman's correlation:

```
SQL>  
select CORR_S(AGE, WEIGHT)  
       coefficient, CORR_S(AGE, WEIGHT, 'TWO_SIDED_SIG')  
       p_value, substr(TREATMENT_PLAN, 1,15)  
       as TREATMENT_PLAN  
from OLSUG.LYMPHOMA
```

GROUP BY TREATMENT_PLAN;



```
Command Prompt - sqlplus cberger/cberger@ora10gr2

SQL> select CORR_S(AGE, WEIGHT) coefficient,
2      CORR_S(AGE, WEIGHT, 'TWO_SIDED_SIG')
3      p_value, substr(TREATMENT_PLAN, 1,15) as TREATMENT_PLAN
4 from CBERGER.LYMPHOMA
5 GROUP BY TREATMENT_PLAN;

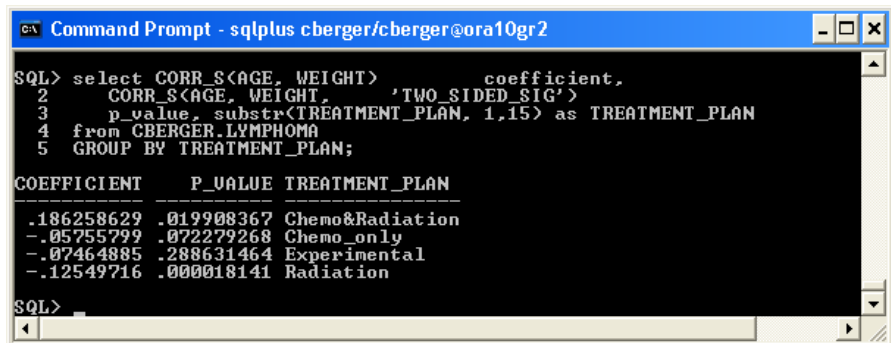
COEFFICIENT      P_VALUE TREATMENT_PLAN
-----
.186258629      .019908367 Chemo&Radiation
-.05755799      .072279268 Chemo_only
-.07464885      .288631464 Experimental
-.12549716      .000018141 Radiation

SQL>
```

Cross Tabulation Statistics

12. This query analyzes the strength of the association between TREATMENT_PLAN and GENDER Grouped By LYMPH_TYPE using a cross tabulation and returns the observed p_value and phi coefficient significance:

```
SQL> 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 OLSUG.LYMPHOMA
GROUP BY LYMPH_TYPE ORDER BY 1;
```



```
Command Prompt - sqlplus cberger/cberger@ora10gr2

SQL> select CORR_S(AGE, WEIGHT) coefficient,
2      CORR_S(AGE, WEIGHT, 'TWO_SIDED_SIG')
3      p_value, substr(TREATMENT_PLAN, 1,15) as TREATMENT_PLAN
4 from CBERGER.LYMPHOMA
5 GROUP BY TREATMENT_PLAN;

COEFFICIENT      P_VALUE TREATMENT_PLAN
-----
.186258629      .019908367 Chemo&Radiation
-.05755799      .072279268 Chemo_only
-.07464885      .288631464 Experimental
-.12549716      .000018141 Radiation

SQL>
```

13. Congratulations! You successfully used Oracle's statistical functions.

For more information, go to

OTN Oracle 10g R2 Documentation
<http://www.oracle.com/pls/db102/homepage>

http://download-west.oracle.com/docs/cd/B19306_01/appdev.102/b14258/d_stat_f.htm