```
*** Start by creating a simple table ...


SQL> CREATE TABLE outlier (id NUMBER, name VARCHAR(20), outlier_date DATE);

Table created.



*** we now populate the table with a whole bunch of rows.
*** The date column is going to be the interesting one, initially it's being populated with a series
of uniformly distributed dates,
*** going back 10000 days


SQL> INSERT INTO outlier SELECT rownum, 'RADIOHEAD', SYSDATE-10000+MOD(rownum,10000) FROM dual
CONNECT BY LEVEL <= 1000000;

1000000 rows created.

SQL> COMMIT;

Commit complete.


*** We now create an index on the date column ...


SQL> CREATE INDEX outlier_date_i ON outlier(outlier_date);

Index created.

SQL> exec DBMS_STATS.GATHER_TABLE_STATS(ownname=> null, tabname=>'outlier', cascade=> TRUE,
estimate_percent=> null, method_opt=> 'FOR ALL COLUMNS SIZE 1');

PL/SQL procedure successfully completed.



*** The dates are perfectly distributed and go back to 12 June 1980 ...


SQL> SELECT MIN(outlier_date) "Min Date", MAX(outlier_date) "Max Date" FROM outlier;

Min Date  Max Date
--------- ---------
12/JUN/80 28/OCT/07


*** When we perform a simple select of everything greater than the start of the current day, we use
the index as expected.
*** Note the cardinality is perfect.


SQL> SELECT * FROM outlier WHERE outlier_date > '28/OCT/2007';

100 rows selected.

Elapsed: 00:00:00.03

Execution Plan
----------------------------------------------------------
Plan hash value: 4074660270
-------------------------------------------------------------------------------------------------
| Id  | Operation                   | Name            | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |                 |   100 |  2200 |      103(0)| 00:00:02 |
|   1 |  TABLE ACCESS BY INDEX ROWID| outlier         |   100 |  2200 |      103(0)| 00:00:02 |
|*  2 |   INDEX RANGE SCAN          | outlier_DATE_I  |   100 |       |        3(0)| 00:00:01 |
-------------------------------------------------------------------------------------------------


Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("outlier_DATE">TO_DATE('2007-10-28 00:00:00', 'yyyy-mm-dd hh24:mi:ss'))



Statistics
----------------------------------------------------------
          1  recursive calls
```

```
          0  db block gets
        110  consistent gets
          1  physical reads
          0  redo size
       2324  bytes sent via SQL*Net to client
        462  bytes received via SQL*Net from client
          8  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
        100  rows processed
```

*** Now let's introduce an outlier date, perhaps used by an application to denote some default characteristic or value
*** The value is 8 Jan 9999, David Bowie's birthday way way way in the future ...

```
SQL> INSERT INTO outlier VALUES (1000001, 'DAVID BOWIE', '08/JAN/9999');

1 row created.

SQL> COMMIT;

Commit complete.

SQL> exec DBMS_STATS.GATHER_TABLE_STATS(ownname=> null, tabname=>'outlier', cascade=> TRUE,
estimate_percent=> null, method_opt=> 'FOR ALL COLUMNS SIZE 1');

PL/SQL procedure successfully completed.
```

*** When we perform the same select as previous, remembering we have only added in one additional date,
*** Note that a full table scan is perform on the same data (plus one row) and that the rows estimate is way way off ...

```
SQL> SELECT * FROM outlier WHERE outlier_date > '28/OCT/2007';

101 rows selected.

Elapsed: 00:00:03.50

Execution Plan
----------------------------------------------------------
Plan hash value: 384055367

--------------------------------------------------------------------------------
| Id  | Operation         | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |          |   996K|    20M|   873   (3)| 00:00:11 |
|*  1 |  TABLE ACCESS FULL| OUTLIER  |   996K|    20M|   873   (3)| 00:00:11 |
--------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("outlier_DATE">TO_DATE('2007-10-28 00:00:00',
              'yyyy-mm-dd hh24:mi:ss'))


Statistics
----------------------------------------------------------
          1  recursive calls
          0  db block gets
       3877  consistent gets
       3865  physical reads
          0  redo size
       2355  bytes sent via SQL*Net to client
        462  bytes received via SQL*Net from client
          8  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
        101  rows processed
```

*** The outlier value has made the CBO go horribly wrong as the CBO basically uses the following formula in its calculation of the selectivity:

*** (max_value - predicate_value) / (max_value - min_value) = (days between 8 Jan 9999 and 28 Oct 2007) / (days between 8 Jan 9999 and 12 Jan 1980)

*** which equals a value very very close to 100% of all rows ...

*** So outlier values can cause all sorts of issues with the CBO.