```
*** Indexes also don't particularly like outlier values as they prevent
space efficient 9010 splits

*** First create a simple table with a primary key on the ID column


SQL> CREATE TABLE outlier2 (id NUMBER CONSTRAINT outlier2_pk_i PRIMARY KEY,
name VARCHAR2(20));

Table created.


*** Now populate the table with monotonically increasing values for the ID
column


SQL> INSERT INTO outlier2 SELECT rownum, 'RADIOHEAD' FROM dual CONNECT BY
LEVEL <= 1000000;

1000000 rows created.

SQL> COMMIT;

Commit complete.


*** Let's see just how compact and efficient our index is on the primary
key


SQL> ANALYZE INDEX outlier2_pk_i VALIDATE STRUCTURE;

Index analyzed.

SQL> SELECT LF_BLKS, BTREE_SPACE, PCT_USED FROM INDEX_STATS;

   LF_BLKS BTREE_SPACE   PCT_USED
---------- ----------- ----------
      1875    15032128        100


*** As expected, the index is entirely used with no free space as Oracle
has been generating 90-10 splits,
*** leaving behind totally full index leaf blocks. Perfect !!

----------------------

*** Now let’s repeat the demo, but this time introduce an outlier value
before inserting in all the data


SQL> CREATE TABLE outlier3 (id NUMBER CONSTRAINT outlier3_pk_i PRIMARY KEY,
name VARCHAR2(20));

Table created.


*** Here's the outlier value, 9999999999999 which is going to be way way
outside the normal range of values
*** and unfortunately, the maximum value in the table now and always moving
forward ...
```

```
SQL> INSERT INTO outlier3 VALUES (9999999999999, 'DAVID BOWIE');

1 row created.

SQL> COMMIT;

Commit complete.


*** Now when we insert our rows, none of these values are ever the maximum
value
*** Therefore as a leaf block is filled, a 50-50 block split is performed


SQL> INSERT INTO outlier3 SELECT rownum, 'RADIOHEAD' FROM dual CONNECT BY
LEVEL <= 1000000;

1000000 rows created.

SQL> COMMIT;

Commit complete.


*** The impact on the space efficiency of the index is significant ...


SQL> ANALYZE INDEX outlier3_pk_i VALIDATE STRUCTURE;

Index analyzed.

SQL> SELECT LF_BLKS, BTREE_SPACE, PCT_USED FROM INDEX_STATS;

   LF_BLKS BTREE_SPACE   PCT_USED
---------- ----------- ----------
      3681    29528320         51



*** Where previously we had a fully utilised index structure, now we are
only using 51% of the index
*** The other 49% is totally wasted and redundant as this space cannot be
used be subsequent inserts
*** Unless subsequent deletes totally empty an index leaf block and the
block is placed again on the freelist ...
```