**Score Partition Summary Description**

This code partitions observations into categories defined by numeric boundaries that optimize a metric. Observations are REAL numbers at the interval or ratio level. Each partition boundary assignment set, defined as a category, has an associated value called a “score” for that partitioning. The scoring mechanism is also a “metric” for evaluating a partition. Our code finds observation boundaries that maximize that score or metric.

The Python code described here uses “maximum R2” as a partitioning score value, and finds a partition with boundaries that maximize that R2 value across all partitions. A brute-force search is used to find the maximum score from all partitions evaluated.

**History**

The original “simplified” version of this code came from statistician Alan Bostrom in the late 1980’s, while working at Iameter, and was written In Visual Basic. It was later translated into C++ and somewhat optimized for production use. It was translated to Java in the late 90’s for a similar problem. This is the first Python version.

**Constraints**

A category may not be “empty”. That is, partitions without observations are not considered in the search. If multiple partitions have the same maximum score, only the first is reported.

**Computational Equivalences used in the Metric Evaluation**

Categories are evaluated with respect to a “metric” to define the best selection. This version of our code uses the “maximum R2” criterion as the scoring metric. The R2 is computed using these equivalences:

*SS Total = SS Within + SS Between*

*SSWithin = SSErrors = SSResidual*

*SSBetween = SSExplained*

The SS values represent sums of squared score values. These equivalences are based on the ANOVA model where the sums of squares around the means of within categories and sums of squares between the means of categories is used as a measure of the amount of variation “explained” by the ANOVA model.

ETA-SQUARED - <https://www.theanalysisfactor.com/effect-size/>.