Package 'QuantileGradeR'

February 6, 2017

Title Quantile-Adjusted Restaurant Grading

Version 0.1.1

Date 2017-02-06

Author Zoe Ashwood <zashwood@law.stanford.edu>,
Becky Elias <Becky.Elias@kingcounty.gov>,
Daniel E. Ho <dho@law.stanford.edu>

Maintainer Zoe Ashwood <zashwood@law.stanford.edu>

Description Implementation of the food safety restaurant grading system adopted by Public Health - Seattle & King County (see Ashwood, Z.C., Elias, B., and Ho. D.E. ``Improving the Reliability of Food Safety Disclosure: A Quantile Adjusted Restaurant Grading System for Seattle-King County'' (working paper)). As reported in the accompanying paper, this package allows jurisdictions to easily implement refinements that address common challenges with unadjusted grading systems. First, in contrast to unadjusted grading, where the most recent single routine inspection is the primary determinant of a grade, grading inputs are allowed to be flexible. For instance, it is straightforward to base the grade on average inspection scores across multiple inspection cycles. Second, the package can identify quantile cutoffs by inputting substantively meaningful regulatory thresholds (e.g., the proportion of establishments receiving sufficient violation points to warrant a return visit). Third, the quantile adjustment equalizes the proportion of establishments in a flexible number of grading categories (e.g., A/B/C) across areas (e.g., ZIP codes, inspector areas) to account for inspector differences. Fourth, the package implements a refined quantile adjustment that addresses two limitations with the stats::quantile() function when applied to inspection score datasets with large numbers of score ties. The quantile adjustment algorithm iterates over quantiles until, over all restaurants in all areas, grading proportions are within a tolerance of desired global proportions. In addition the package allows a modified definition of ``quantile'' from ``Nearest Rank''. Instead of requiring that at least $p[1]\%$ of restaurants receive the top grade and at least $(p[1]+p[2])\%$ of restaurants receive the top or second best grade for quantiles p, the algorithm searches for cutoffs so that as close as possible $p[1]\%$ of restaurants receive the top grade, and as close as possible to $p[2]\%$ of restaurants receive the second top grade.

URL http://www.kingcounty.gov/depts/health/environmental-health/food-safety/inspection-system/food-safety-rating.aspx

Depends R (>= 3.2.3)

License GPL (>= 2)

LazyData TRUE

RoxygenNote 5.0.1

Imports stats

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-06 21:22:48

R topics documented:

---

findCutoffs	*Find Cutoff Values.*

Description

findCutoffs applies a quantile adjustment to inspection scores within a jurisdiction's subunits
(e.g. ZIP codes) and creates a data frame of cutoff values to be used for grading restaurants or other
inspected entities.

Usage

```
findCutoffs(X, z, gamma, resolve.ties = TRUE, restaurant.tol = 10,
  max.iterations = 20)
```

Arguments

X	Numeric matrix of size n x p, where n is the number is restaurants to be graded and p is the number of inspections to be used in grade assignment. Entry X[i,j] represents the inspection score for the ith restaurant in the jth most recent inspection.
z	Character vector of length n representing ZIP codes (or other subunits within a jurisdiction). z[i] is the ZIP code corresponding to the restaurant with inspection scores in row i of X.
gamma	Numeric vector representing absolute grade cutoffs. Entries in gamma should be increasing, with gamma[1] <= gamma[2] etc (this is related to the "Warning" section and larger scores being associated with higher risk).

resolve.ties   Boolean value that determines the definition of quantile to be used after optimal quantiles have been found with the percentileSeek function. See Modes below, as well as Appendix J of Ho, D.E., Ashwood, Z.C., and Elias, B. "Improving the Reliability of Food Safety Disclosure: A Quantile Adjusted Restaurant Grading System for Seattle-King County".

restaurant.tol   An integer indicating the maximum difference in the number of restaurants in a grading category between the unadjusted and adjusted grading algorithms (for the top length(gamma) grading categories).

max.iterations   The maximum number of iterations that the iterative algorithm (carried out by the internal percentileSeek function) should run in order to find optimal quantiles for ZIP cutoffs. The iterative algorithm is described in more detail below.

Details

In our documentation, we use the language "ZIP code" and "restaurant", however, our grading algorithm and our code can be applied to grade other inspected entities; and quantile cutoffs can be sought in subunits of a jurisdiction that are not ZIP codes. For example, it may make sense to search for quantile cutoffs in an inspector's allocated inspection area or within a census tract. We chose to work with ZIP codes in our work because area assignments for inspectors in King County (WA) tend to be single or multiple ZIP codes, and we desired to assign grades based on how a restaurant's scores compare to other restaurants assessed by the same inspector. We could have calculated quantile cutoffs in an inspector's allocated area, but inspector areas are not always contiguous. Because food choices are generally local, ZIP codes offer a transparent and meaningful basis for consumers to distinguish establishments. Where "ZIP code" is referenced, please read "ZIP code or other subunit of a jurisdiction" and "restaurant" should read "restaurant or other entity to be graded".

findCutoffs takes in a vector of cutoff scores, gamma, a matrix of restaurants' scores, X, and a vector corresponding to restaurants' ZIP codes, z, and outputs a data frame of cutoff scores to be used in the gradeAllBus function to assign grades to restaurants. findCutoffs first carries out "unadjusted grading" and compares restaurants' most recent routine inspection scores to the raw cutoff scores contained in gamma and assigns initial grades to restaurants. Grade proportions in this scheme are then used as initial quantiles to find quantile cutoffs in each ZIP code (or quantile cutoffs accommodating for the presence of score ties in the ZIP code, depending on the value of resolve.ties; see the Modes section). Restaurants are then graded with the ZIP code quantile cutoffs, and grading proportions are compared with grading proportions from the unadjusted system. Quantiles are iterated over one at a time (by the internal percentileSeek function, which uses a binary search root finding method) until grading proportions with ZIP code quantile cutoffs are within a certain tolerance (as determined by restaurant.tol) of the unadjusted grading proportions. This iterative step is important because of the discrete nature of the inspection score distribution, and the existence of large numbers of restaurants with the same inspection scores.

The returned ZIP code cutoff data frame has one row for each unique ZIP code and has (length(gamma)+1) columns, corresponding to one column for the ZIP code name, and (length(gamma)) cutoff scores separating the (length(gamma)+1) grading categories. Across each ZIP code's row, cutoff scores increase and we assume, as in the King County (WA) case, that greater risk is associated with larger inspection scores. (If scores are decreasing in risk, users should transform inspection scores with a simple function such as $f(score) = -score$ before using any of the functions in QuantileGradeR.)

Modes

When `resolve.ties = TRUE`, in order to calculate quantile cutoffs in a ZIP code, we alter the definition of quantile from the usual "Nearest Rank" definition and use the "Quantile Adjustment (with Ties Resolution)" definition that is discussed in Appendix J of Ho, D.E., Ashwood, Z.C., and Elias, B. "Improving the Reliability of Food Safety Disclosure: A Quantile Adjusted Restaurant Grading System for Seattle-King County" (working paper). In particular, once we have found the optimal set of quantiles to be applied across ZIP codes, p, with the `percentileSeek` function, instead of returning (for B/C cutoffs, for example) the scores in each ZIP code that result in *at least* (p[2] x 100)% of restaurants in the ZIP code scoring less than or equal to these cutoffs, the mode `resolve.ties = TRUE` takes into account the ties that exist in ZIP codes. Returned scores for A/B cutoffs are those that result in the *closest* percentage of restaurants in the ZIP code scoring less than or equal to the A/B cutoff to the desired percentage, (p[1] x 100)%. Similarly, B/C cutoffs are the scores in the ZIP code that result in the *closest* percentage of restaurants in the ZIP code scoring less than or equal to the B/C cutoff and more than the A/B cutoff to the desired percentage, ((p[2] - p[1]) x 100)%.

When `resolve.ties = FALSE`, we use the usual "Nearest Rank" definition of quantile when applying the optimal quantiles, p, across ZIP codes.

Warning

`findCutoffs` will produce cutoff scores even for ZIP codes with only one restaurant: situations in which a quantile adjustment shouldn't be used. It is the job of the user to ensure that, if using the `findCutoffs` function, it makes sense to do so. This may involve only performing the quantile adjustment on larger ZIP codes and providing absolute cutoff points for smaller ZIP codes, or may involve aggregating smaller ZIP codes into a larger geographical unit and then performing the quantile adjustment on the larger area (the latter approach is the one we adopted).

As mentioned previously, `findCutoffs` was created for an inspection system that associates greater risk with larger inspection scores. If the inspection system of interest associates greater risk with reduced scores, it will be neccessary to perform a transformation of the scores matrix before utilizing the `findCutoffs` function. However a simple function such as `f(score) = - score` would perform the necessary transformation.

Examples

```
## ==== Quantile-Adjusted Grading =====
## ZIP Code Cutoffs
# In King County, meaningful scores in the inspection system are 0 and 30:
# more than 50% of restaurants score 0 points in a single inspection round,
# and 30 is the highest score that a restaurant can be assigned before it is
# subject to a return inspection, hence these values form our gamma vector.
# The output dataframe, zipcode.cutoffs.df, has ten rows and three columns: one
# row for every unique ZIP code in zips.kc, one column for the ZIP name, the
# second column for the A/B cutoff (Gamma.A) and the third column for the B/C
# cutoff (Gamma.B).

 zipcode.cutoffs.df <- findCutoffs(X.kc, zips.kc, gamma = c(0, 30))

## ==== Traditional Grading Systems ====
```

```
## ZIP Code Cutoffs
# Traditional (unadjusted) restaurant grading systems use the same cutoff scores
# for all ZIP codes. To allow comparison, an unadjusted ZIP code cutoff frame
# for King County is generated by the internal createCutoffsDF function:

 unadj.cutoffs.df <- createCutoffsDF(X.kc, zips.kc, gamma = c(0, 30), type = "unadj")
```

gradeAllBus	*Grade Businesses.*

Description

`gradeAllBus` takes in a vector of business inspection scores, business ZIP codes and a data frame of ZIP code cutoff scores (generated by the `findCutoffs` function) and returns a vector of business grades.

Usage

```
gradeAllBus(scores, z, zip.cutoffs)
```

Arguments

scores	Numeric vector of length n, where n is the number is restaurants to be graded. Each entry is the inspection score for one business.
z	Character vector of length n, where each entry is the ZIP code (or other geographic area) of a business. The order of businesses in z is the same as the order of businesses in scores.
zip.cutoffs	A dataframe with the first column containing all of the ZIP codes in z and later columns containing cutoff scores for each ZIP code for grade classification. Cutoff scores for each ZIP code should be ordered from lowest score in column 2 (representing the cutoff for the best grade) to the largest cutoff score in the final column (representing the cutoff inspection score for the second worst grade). This dataframe will most likely have been generated by the `findCutoffs` function.

Details

As explained in the `findCutoffs` documentation, we use the language "ZIP code" and "restaurant", however, our grading algorithm can be applied to grade other inspected entities. As with `findCutoffs`, where "ZIP code" is referenced, please read "ZIP code or other subunit of a jurisdiction" and "restaurant" should read "restaurant or other entity to be graded".

`gradeAllBus` takes a vector of inspection scores (one score for each restaurant: the score can be a mean across multiple inspections or the result of a single inspection), a vector of ZIP codes and a dataframe of ZIP code cutoffs (most likely generated by the `findCutoffs` function). It compares each restaurant's inspection score to cutoff scores in the restaurant's ZIP code. It finds the smallest cutoff score in the restaurant's ZIP code that the restaurant's inspection score is less than or equal to - let's say this is the (letter.index)th cutoff score - and returns the (letter.index)th letter of

the alphabet as the grade for the restaurant. The returned vector of grades maintains the order of businesses in vector inputs scores and in z).

Value

A character vector of length n, with each entry corresponding to the grade that the restaurant received.

Examples

```
## ===== Quantile-Adjusted Grading =====
## ZIP Code Cutoffs (see findCutoffs documentation for an explanation of how
## these are calculated)

 zipcode.cutoffs.df <- findCutoffs(X.kc, zips.kc, gamma = c(0, 30))

## In King County, we use a restaurant's mean inspection score over the last
## four inspections for grading (see Ho, D.E.,
## Ashwood, Z.C., and Elias, B. "Improving the Reliability of Food Safety
## Disclosure: A Quantile Adjusted Restaurant Grading System for Seattle-King
## County" (working paper)). Calculate these mean scores:

 mean.scores <- rowMeans(X.kc, na.rm = TRUE)

## We then use the mean scores and the zipcode.cutoffs.df dataframe to perform
## grading:

 adj.grades <- gradeAllBus(mean.scores, zips.kc, zipcode.cutoffs.df)


## ===== Traditional Grading Systems =====
## For comparison, calculate grades as if we had used a traditional grading
## system in King County, with 0 and 30 as the A/B and B/C cutoffs for all ZIP
## codes.

## Cutoffs:

 unadj.cutoffs.df <- createCutoffsDF(X.kc, zips.kc, gamma = c(0, 30), type = "unadj")

## Grades (traditional grading systems only use the most recent inspection score
## for grading):

 unadj.grades <- gradeAllBus(scores = X.kc[,c(1)], zips.kc, zip.cutoffs = unadj.cutoffs.df)


## ===== Comparison: Quantile-Adjusted Grading and Traditional Grading ===
## Proportion of restaurants in each grading category varies dramatically
## between ZIPs in traditional compared to quantile-adjusted grading; these
## differences do not reflect sanitation differences, but rather differences in
## stringency across inpectors (see: Ho, D.E., Ashwood, Z.C., and Elias, B.
```

```
## "Improving the Reliability of Food Safety Disclosure: A Quantile Adjusted
## Restaurant Grading System for Seattle-King County" (working paper)).
## Tabulate restaurants in each ZIP code in each grading category and then
## divide by total number of restaurants in each ZIP to obtain proportions.
## Proportions are rounded to 2 decimal places.

## Traditional Grading

 foo1 <- round(table(zips.kc, unadj.grades)/apply(table(unadj.grades, zips.kc), 2, sum), 2)

## Quantile-Adjusted Grading

 foo2 <- round(table(zips.kc, adj.grades)/apply(table(adj.grades, zips.kc), 2, sum), 2)
```

X.kc   *Example Inspection Scores Matrix.*

Description

A small dataset of inspection scores.

Usage

```
X.kc
```

Format

A matrix with 4 columns and ~1500 rows, where each row represents one business and each column is one inspection cycle. X.kc[i,j] represents the inspection score for the ith restaurant in the jth most recent inspection.

Details

X.kc contains restaurant inspection information from 11 randomly chosen ZIP codes in the King County (WA) jurisdiction. Establishments and ZIP codes are masked. Inspection information is limited to the 01-01-2012 to 03-25-2016 time period.

---

zips.kc   *Example ZIP Code Vector.*

Description

A vector of ZIP codes.

Usage

```
zips.kc
```

Format

A character vector with a length that matches the number of rows of X.kc (i.e. zips.kc has ~1500 elements). Each entry represents the ZIP code of one business.

Details

zips.kc[i] represents the ZIP code for the restaurant represented in the ith row of the X.kc inspection scores matrix. ZIP codes in zips.kc have the format "zip.j" where j is an integer between 1 and 11, i.e., ZIP codes are masked. In this masking step, we also demonstrate that our functions can be applied not solely over character vectors of real ZIP codes, but any vector of character strings representing the same facet for all restaurants can be used in the grading process.

Index