clustcurv: An R Package for Determining Groups in Multiple Curves

by Nora M. Villanueva, Marta Sestelo, Luis Meira-Machado and Javier Roca-Pardiñas

Abstract In many situations, it could be interesting to ascertain whether groups of curves can be performed, especially when confronted with a considerable number of curves. This paper introduces an R package, known as **clustcurv**, for determining clusters of curves with an automatic selection of their number. The package can be used for determining groups in multiple survival curves as well as for multiple regression curves. Moreover, it can be used with large numbers of curves. An illustration of the use of **clustcurv** is provided, using both real data examples and artificial data.

Keywords: multiple curves, number of groups, nonparametric, survival analysis, regression models, cluster

1 Introduction

A problem often encountered in many fields is the comparison of several populations through specific curves. Typical examples, considered by a number of authors, are given by the comparison of survival curves in survival analysis, children growth curves in pediatrics, or the comparison of regression curves in regression analysis. In many of these studies, it is very common to compare a large number of curves between groups, and methods of summarizing and extracting relevant information are necessary. A common approach is to look for a partition of the sample into a number of groups in such a way that curves in the same group are as alike as possible but as distinct as possible from those in other groups. This process is also known as curve clustering. A hypothesis test can be used to ascertain that the curves in the same group are equal. A fundamental and difficult problem in clustering curves is the estimation of the number of clusters in a dataset.

Traditionally, the comparison of these functions is performed using parametric models through the comparison of the resulting model parameters. This approach, however, requires the specification of the parametric model, which is often difficult and may be considered a disadvantage. Several nonparametric methods have been proposed in the literature to compare multiple curves. In the area of survival analysis, for example, several nonparametric methods have been proposed to test for the equality of survival curves for censored data. The most well-known and widely used to test the null hypothesis of no difference in survival between two or more independent groups was proposed by Mantel (1966). An alternative test that is often used is the Peto & Peto (Peto and Peto, 1972) modification of the Gehan-Wilcoxon test (Gehan, 1965). Several other variations of the log-rank test statistic using weights on each event time have been proposed in the literature (Tarone and Ware, 1977; Harrington and Fleming, 1982; Fleming et al., 1987) as well as other procedures to compare these survival curves based on different measures, as can be the medians (Chen and Zhang, 2016). There exists an extensive literature on curve comparison in the framework of regression analysis. In this context, several nonparametric tests have been proposed to test the equality of the mean functions, $H_0 : m_1 = \cdots = m_j$. Hall and Hart (1990) proposed a bootstrap test, while Härdle and Mammen (1993) Hardle and Marron (1990) suggested a semiparametric approach based on kernel smoothing. Other relevant papers on this topic are King et al. (1991) , Delgado (1993) , Kulasekera (1995) , Young and Bowman (1995), Dette and Neumeyer (2001), Pardo-Fernández et al. (2007) , Srihera and Stute (2010) , among others. A good review on this topic can be seen in the paper by Neumeyer and Dette (2003).

When the null hypothesis of equality of curves is rejected, leading to the clear conclusion that at least one curve is different, it can be interesting to ascertain whether curves can be grouped or if all these curves are different from each other. In this setting, one naïve approach would be to perform pairwise comparisons. In this line are the papers by Rosenblatt (1975), González-Manteiga and Cao (1993), Härdle and Mammen (1993), Dette and Neumeyer (2001) who proposed alternative tests of the null hypothesis of equality of curves obtained from pairwise comparisons of the estimators of the regression functions. A similar statistic was also considered by King et al. (1991). Pairwise comparisons between group levels with corrections for multiple testing are also possible in the framework of survival analysis. Among others, this can be achieved with the `pairwise_survdiff` of the package **survminer** (Kassambara et al., 2019). However, in any case, this approach would lead to a large number of comparisons without the possibility of determining groups with similar curves. Results for such a test can tell us that all combinations are different, or just one pair. When the number of curves to be compared increase, so does the difficulty of interpretation.

For partitioning a given set of curves into a set of *K* groups of curves (i.e., *K* clusters), Villanueva et al. (2019a) propose an adaptation of the *K*-means methodology. *K*-means is probably the most

commonly used clustering method for splitting a dataset into a set of *K* groups with a very simple and fast algorithm. Furthermore, it can efficiently deal with very large data sets. One potential disadvantage of *K*-means clustering is that it requires the number of clusters to be pre-specified. A method is proposed by Villanueva et al. (2019a) to determine the number of clusters.

The development of **clustcurv** R package has been motivated by recent contributions that account for these problems, in particular, the methods proposed by Villanueva et al. (2019a) to determine groups in multiple survival curves and those introduced by Villanueva et al. (2019b) in the framework of regression curves. The **clustcurv** R package attempts to answer the following two questions: (i) given a potential large sample of curves, what is the best value for the number of clusters? (ii) What is the best subdivision of the sample curves into a given number of *K* clusters? To facilitate the task of selecting the optimal number of clusters as well as the composition of the clusters, it is essential to have software for implementing the proposed methods in an environment that researchers will find user-friendly and easily understandable. We believe that our package can answer this aim by providing several user-friendly functions. The package **clustcurv** is freely available from the Comprehensive R Archive Network (CRAN) at `https://cran.r-project.org/web/packages/clustcurv`

Three data sets were chosen for illustration of the software usage with real data. The first two datasets show the applicability of the proposed methods for obtaining clusters of survival curves. These applications were chosen to solve two real problems in the study of recurrence of breast cancer patients and survival of myeloma cancer. To illustrate the package usage in the regression context we used real data from a Barnacle's Growth study conducted in Galicia, Spain. Simulated data were also used to illustrate the package capabilities in a more complicated scenario.

The remainder of the paper is structured as follows: Section 2 briefly reviews methods for selecting the number of clusters and the nonparametric test used; Section 3 explains the use of the main functions and methods of **clustcurv**; Section 4 gives an illustration of the practical application of the package using real and simulated data; and finally, the last section concludes with a discussion and possible future extensions of the package.

2 An overview of the methodology

In this section, we briefly review the methodological background of the **clustcurv** package. As it solves problems addressed in the field of survival analysis and regression analysis, firstly, the notation and the nonparametric estimation procedures for both contexts are exposed. Then, the procedure for determining groups of curves is explained in detail, considering a general framework which includes the aforementioned contexts. Briefly, our procedure is described as follows. First, the *J* curves are estimated by nonparametric estimators. Second, given a number of *K* groups, the optimal possible assignment of *J* curves into *K* groups is chosen by means of a heuristic algorithm. Third, the optimal number of groups is determined using an automatic bootstrap-based testing procedure.

Notation and estimation procedure in the survival context

We will assume the *J*-sample general random censorship model where observations are made on n_j individuals from population $j(j = 1, \ldots, J)$. Denote $n = \sum_{j=1}^{J} n_j$ and suppose that the observations from the *n* individuals are mutually independent. Let T_{ij} be an event time corresponding to an event measured from the start of the follow-up of the *i*-th subject $(i = 1, \ldots, n_j)$ in the sample *j*, and assume that T_{ij} is observed subject to a (univariate) random right-censoring variable C_{ij} assumed to be independent of T_{ij}. Due to the censoring, rather than T_{ij}, we observe $(\widetilde{T}_{ij}, \Delta_{ij})$, where $\widetilde{T}_{ij} = \min(T_{ij}, C_{ij})$, $\Delta_{ij} = I(T_{ij} \leq C_{ij})$, where $I(\cdot)$ is the indicator function.

Since the censoring time is assumed to be independent of the process, the survival functions, $S_j(t) = P(T_j > t)$, may be consistently estimated by the Kaplan-Meier estimator (Kaplan and Meier, 1958) based on the $(\widetilde{T}_j, \Delta_j)$. The Kaplan-Meier estimator or the Product-Limit estimator is a nonparametric method frequently used to estimate survival for censored data. Let $t_1 < t_2 < \ldots < t_{m_j}$, $m_j \leq n_j$ denote the distinct ordered failure times from population j $(j = 1, \ldots, J)$, and let d_u be the number of events from population j at time t_u. Then, the Kaplan-Meier estimator of survival (for population j) is

$$\widehat{S}_j(t) = \prod_{u:t_u \leq t} \left(1 - \frac{d_u}{R_j(t_u)}\right),$$

where $R_j(t) = \sum_{i=1}^{n_j} I(\widetilde{T}_{ij} \geq t)$ denote the number of individuals at risk just before time *t*, among individuals from population *j*. The Kaplan-Meier estimate is a step function with jumps at event times. The size of the steps depends on the number of events and the number of individuals at risk at the

corresponding time. Under this setup, we will be interested to determine clusters in multiple survival curves.

Notation and estimation procedure in the regression context

Let (X_j, Y_j) be J independent random vectors, and assume that they satisfy the following nonparametric regression models, for $j = 1, \ldots, J$,

$$Y_j = m_j(X_j) + \varepsilon_j, \tag{1}$$

where the error variable ε_j has mean zero, and $m_j(X_j) = E(Y_j | X_j)$ is the unknown regression function. We do not make any assumptions about the error distribution.

The regression models in (1) can be estimated using several approaches, such as methods based on regression splines (de Boor, 2001), Bayesian approaches (Lang and Brezger, 2004), or local polynomial kernel smoothers (Wand and Jones, 1995; Fan and Gijbels, 1996). In this package, local linear kernel smoothers, as implemented in the **npregfast** package, are used.

Determining groups of nonparametric curves

As noted earlier, several authors have proposed different methods that can be used to compare estimates of nonparametric functions of multiple samples. The null hypothesis is that all the curves have identical functions, $H_0 : \mathcal{F}_1 = \ldots = \mathcal{F}_J$. However, if this hypothesis is rejected, there are no available procedures that let determine groups among these curves, that is, to assess if the levels $\{1, \ldots, J\}$ can be grouped in K groups $(\mathcal{G}_1, \ldots, \mathcal{G}_K)$ with $K < J$, so that $\mathcal{F}_i = \mathcal{F}_j$ for all $i, j \in \mathcal{G}_k$, for each $k = 1, \ldots, K$. Note that $(\mathcal{G}_1, \ldots, \mathcal{G}_K)$ must be a partition of $\{1, \ldots, J\}$, and therefore must satisfy the following conditions:

$$\mathcal{G}_1 \cup \ldots \cup \mathcal{G}_K = \{1, \ldots, J\} \quad \text{and} \quad \mathcal{G}_i \cap \mathcal{G}_j = \varnothing, \quad \forall i \neq j \in \{1, \ldots, K\}. \tag{2}$$

We propose a procedure to test, for a given number K, the null hypothesis $H_0(K)$ that at least one partition exists $(\mathcal{G}_1, \ldots, \mathcal{G}_K)$ so that all the conditions above are verified. The alternative hypothesis $H_1(K)$ is that for any $(\mathcal{G}_1, \ldots, \mathcal{G}_K)$, exists at least a group $\mathcal{G}_k$ in which $\mathcal{F}_i \neq \mathcal{F}_j$ for some $i, j \in \mathcal{G}_k$.

The cited testing procedure is based on the J-dimensional process

$$\widehat{\mathbf{U}}(z) = (\widehat{U}_1(z), \widehat{U}_2(z), \ldots, \widehat{U}_J(z))^t,$$

where, for $j = 1, \ldots, J$,

$$\widehat{U}_j(z) = \sum_{k=1}^{K} [\widehat{\mathcal{F}}_j(z) - \widehat{\mathcal{C}}_k(z)] \, I_{\{j \in \mathcal{G}_k\}},$$

and $\widehat{\mathcal{C}}_k$ is the pooled nonparametric estimate based on the combined $\mathcal{G}_k$-partition sample.

The following test statistics were considered in order to test $H_0(K)$: a Cramér-von Mises type statistic

$$D_{CM} = \min_{\mathcal{G}_1, \ldots, \mathcal{G}_K} \sum_{j=1}^{J} \int_R \widehat{U}_j^2(z) dz,$$

and a modification of it based on the L_1 norm proposed in the Kolmogorov-Smirnov test statistic

$$D_{KS} = \min_{\mathcal{G}_1, \ldots, \mathcal{G}_K} \sum_{j=1}^{J} \int_R |\widehat{U}_j(z)| dz,$$

where R is the support of the lifetime distribution or the support of the independent variable in case of survival or regression, respectively.

In order to approximate the minimizers involved in the test statistics, we propose the use of clustering algorithms. Particularly, in the case of D_{CM}, defined in terms of the L_2-distance, we propose the use of the K-means (Macqueen, 1967). However, for obtaining the values of D_{KS}, defined in this case in terms of the L_1-norm, a variation of the K-means where instead of calculating the mean for each group to determine its centroid, it calculates the median, the k-medians —suggested by Macqueen (1967) and developed by Kaufman and Rousseeuw (1990)— would be more appropriate. In both cases, the carried-out procedure is equivalent: the functions $\mathcal{F}_j$ ($j = 1, \ldots, J$) have to be estimated in a common grid of size Q leading to a matrix of (J x Q) dimension, where each row corresponds

with the estimates of the j curve in the Q positions of the grid. Then, this matrix will be the input of both heuristic methods, K-means and K-medians, and from these, the "best" partition $(\mathcal{G}_1, \ldots, \mathcal{G}_K)$ is obtained.

Finally, the decision rule based on D consists of rejecting the null hypothesis if D is larger than the $(1-\alpha)$-percentile obtained under the null hypothesis. To approximate the distributions of the test statistic under the null hypothesis, resampling methods such as the bootstrap introduced by Efron (1979) can be applied.

The testing procedure used here involves the following steps:

1. Using the original sample, for $j = 1, \ldots, J$ and $i = 1, \ldots, n_j$, estimate the functions $\mathcal{F}_j$ in a nonparametric way and in a common grid using each sample separately. Then, using the proposed algorithms, obtain the "best" partition $(\mathcal{G}_1, \ldots, \mathcal{G}_K)$. With it, obtain the estimated curves $\widehat{\mathcal{C}}_k$ using a pooled nonparametric estimator based on the combined partition samples (i.e., the estimator obtained by applying the nonparametric estimator to the combined partition samples).

2. Obtain the D value as explained before.

3. Draw bootstrap samples using a bootstrap procedure. In the survival context, follow step 3.(a), and in the regression context, follow step 3.(b):

    (a) For $b = 1, \ldots, B$ (e.g., $B = 1000$), and for each $j \in \mathcal{G}_k$, draw $(\widetilde{T}_{1j}^{*b}, \Delta_{1j}^{*b}), (\widetilde{T}_{2j}^{*b}, \Delta_{2j}^{*b}), \ldots, (\widetilde{T}_{n_j j}^{*b}, \Delta_{n_j j}^{*b})$ by independent sampling n_j times with replacement from the empirical distribution function, $\widehat{F}_k$, putting mass n_k^{-1} ($n_k = \sum_{j=1}^J n_j I_{\{j \in \mathcal{G}_k\}}$) at each point $(\widetilde{T}_{ij}, \Delta_{ij})$, with $j \in \mathcal{G}_k$. Note that this procedure is a pooled bootstrap, i.e., bootstrap from the pooled-combined partition sample given by the null hypothesis $H_0(K)$.

    (b) For $b = 1, \ldots, B$, and for each $j \in \mathcal{G}_k$, draw $\left\{ \left(X_{i1}, Y_{i1}^{*b}\right) \right\}_{i=1}^{n_1}, \ldots, \left\{ \left(X_{iJ}, Y_{iJ}^{*b}\right) \right\}_{i=1}^{n_J}$, where

$$Y_{ij}^{*b} = \sum_{k=1}^K \widehat{\mathcal{C}}_k(X_{ij}) I_{\{j \in \mathcal{G}_k\}} + \widehat{\varepsilon}_{ij} W_i^{*b}$$

being $\widehat{\varepsilon}_{ij}$ the null errors under the $H_0(K)$ obtained as

$$\widehat{\varepsilon}_{ij} = \sum_{k=1}^K \left(Y_{ij} - \widehat{\mathcal{C}}_k(X_{ij})\right) I_{\{j \in \mathcal{G}_k\}},$$

and the variables $W_1^{*b}, \ldots, W_n^{*b}$ are independent for the observed sample and i.i.d. with $E(W_i^{*b}) = 0$, $Var(W_i^{*b}) = 1$, and third moment equals to 1. A common choice is to consider a binary variable with probabilities $P\{W_i^{*b} = (1 - \sqrt{5})/2\} = (5 + \sqrt{5})/10$ and $P\{W_i^{*b} = (1 + \sqrt{5})/2\} = (5 - \sqrt{5})/10$, which corresponds to the *golden section*. Note that we have used the wild bootstrap here (Wu, 1986; Liu, 1988; Mammen, 1993) because this method is valid both for homoscedastic and for heteroscedastic models where the variance of the error is a function of the covariate.

4. Let D^{*b} be the test statistic obtained from the bootstrap samples after applying step 1 and 2 to the cited bootstrap samples.

Since in step 3 the bootstrap resamples are constructed under the null hypothesis of K groups, this mechanism approximates the distribution of the test statistic under the null hypothesis. If we denote $D^{*(b)}$ for the order statistics of the values $D^{*1}, \ldots, D^{*B}$ obtained in step 4, then $D^{*([(1-\alpha)B])}$ approximates the $(1-\alpha)$-quantile of the distribution of D under the null hypothesis.

It is important to highlight that repeating this procedure, testing $H_0(K)$, from $K = 1$ onwards until a certain null hypothesis is not rejected allows us to determine the number of K groups automatically. Note that however, that unlike the previous test decision, this latter one is not statistically significant (strong evidences for rejecting the null hypothesis are not given). The whole procedure is briefly described step by step in Algorithm 1.

Finally, note that, under survival and regression scenarios, the proposed procedure for the determination of groups in multiple curves may be translated as a test of multiple hypotheses where a set of K p-values corresponding to the K null hypotheses, $H_0(1), H_0(2), \ldots, H_0(K)$ are given. Even though several methods have been proposed to deal with this problem (see, e.g., Dudoit and van der Laan (2008) for an introduction to this area), there are still open challenges because there is no information about the minimum number of tests needed to apply these techniques. In any case, we have decided to propose a possible approach to apply some of these well-known techniques as Bonferroni, Holm

---

Algorithm 1: k-nonparametric curves algorithm

---

1. With the original sample, for $j = 1, \ldots, J$ and $i = 1, \ldots, n_j$, and using the nonparametric estimator obtain $\widehat{\mathcal{F}}_j$.

2. Initialize with $K = 1$ and test $H_0(K)$:

    2.1 Obtain the "best" partition $\mathcal{G}_1, \ldots, \mathcal{G}_K$ by means of the k-means or k-medians algorithm.

    2.2 For $k = 1, \ldots, K$, estimate $\widehat{\mathcal{C}}_k$ and retrieve the test statistic D.

    2.3 Generate B bootstrap samples and calculate D^{*b}, for $b = 1, \ldots, B$.

    2.4 **if** $D > D^{*(1-\alpha)}$ **then**
    >   reject $H_0(K)$
    > $K = K + 1$
    >   go back to 2.1
    > **else**
    >   accept $H_0(K)$
    > **end**

3. The number K of groups of equal nonparametric curves is determined.

(Holm, 1979), etc. As the problem is still open, we feel that the final user must be able to decide to apply them by means of an argument included in the functions of the package. The challenge in the present context is that the number of hypotheses that are going to be tested is unknown in advance. In order to solve this we propose that, after having increased K in the algorithm, the null hypothesis for "smaller K's" has to be re-tested simultaneously with $H_0(K)$.

3   Package structure and functionality

The **clustcurv** package is a shortcut for "clustering curves" for being this its major functionality: to provide a procedure that allows users to determine groups of multiple curves with an automatic selection of their number. The package enables both survival and regression curves to be grouped, and it is designed along lines similarly into both contexts. In addition, in view of the high computational cost entailed in these methods, parallelization techniques are included to become feasible and efficient in real situations.

The functions within the **clustcurv** package are described in Table 1. Briefly, there are two main types of functionalities: (i) to determine groups of multiple curves with the automatic selection of their number with `regclustcurves` or `survclustcurves` functions and (ii) to determine groups of curves, given a number K, with `kregcurves` or `ksurvcurves` functions. The S3 object obtained from whatever previous functions is the argument required as input for `autoplot`, which returns a graphical output based on **ggplot2** package. Numerical summaries of the fitted objects can be obtained by using `print` or `summary`.

Since the two most important functions in this package are `survclustcurv` and `regclustcurv`, the arguments of these functions are shown in Table 2. Note that the `ksurvcurves` `kregcurves` functions are just a simplified version of the previous two. Users can automatically obtain the optimal number of groups of multiple curves by means of `survclustcurves` and `regclustcurves`. Nevertheless, in those situations where the user knows in advance the number of groups, it is possible to obtain the assignment of the curves into the corresponding group, by means of the function `ksurvcurves` or `kregcurves`. In both functions, a common argument is the algorithm, which returns the best assignments of the curves into the groups to which they belong. At the moment, the algorithms to solve this optimization problem can be K-means or K-medians, through the argument `algorithm = 'kmeans'` or `algorithm = 'kmedians'`.

Furthermore, in order to address the high computational burden, the functions `survclustcurves`, `regclustcurves`, `ksurvcurves` and `kregcurves` have been programmed in parallel to compute the bootstrap-based testing procedure. The input command required for the use of parallelization is `cluster = TRUE`. The number of cores for parallel execution is fixed using the number of CPU cores on the current host minus one unless it is specified by the user (`ncores = NULL`). Then,

registerDoParallel of the **doParallel** package is used to register the parallel backend. The parallel computation is performed by the foreach function of **foreach** package.

Function	Description
survclustcurves	Main function for determining groups of multiple survival curves and selecting automatically the optimal number of them.
regclustcurves	Main function for determining groups of multiple regression curves ecting automatically the optimal number of them.
ksurvcurves	Main function for determining groups of survival curves, given a number of groups *K*.
kregcurves	Main function for determining groups of regression curves, given a number of groups *K*.
summary	Method of the generic summary function for kcurves and clustcurves objects (both survival and regression context), which returns a short summary.
print	Method of the generic print function for kcurves and clustcurves objects, which prints out some key components.
autoplot	Visualisation of clustcurves and kcurves objects with **ggplot2** (Wickham et al., 2019) graphics. Provides the plots for the estimated non-parametric curves grouped by color (optional) and their centroids (mean curve of the curves pertaining to the same group).

Table 1: Summary of functions in the **clustcurv** package.

4 Illustrative examples

In this section, we illustrate the use of **clustcurv** package using some real and simulated data. In the case of the survival context, the proposed methods were applied to the German breast cancer data included in the **condSURV** package and to the multiple myeloma data freely available as part of the **survminer** package. For the regression analysis, the **clustcurv** package includes a data set called barnacle5 with measurements of rostro-carinal length and dry weight of barnacles collected from five sites of Galicia (northwest of Spain). Additionally, in order to show the behaviour of the method in a more complicated scenario, an example with simulated data is also provided.

Application to German Breast Cancer Study Data

In this study, a total of 686 patients with primary node-positive breast cancer were recruited between July 1984 and December 1989, and 16 variables were measured such as the age of the patient (age), menopausal status (menopause), hormonal therapy (hormone), tumour size (size, in mm), tumor grade (grade), and the number of positive nodes (nodes). In addition to these and other variables, the recurrence-free survival time (rectime, in days) and the corresponding censoring indicator (0 – censored, 1 – event) were also recorded.

We will use these data to illustrate the package capabilities to build clusters of survival curves based on the covariate nodes (grouped from 1 to > 13). An excerpt of the data.frame with one row per patient is shown below:

```
> library(condSURV)
> library(clustcurv)
> data(gbcsCS)
> head(gbcsCS[, c(5:10, 13, 14)])
  age menopause hormone size grade nodes rectime censrec
1  38         1       1   18     3     5    1337       1
2  52         1       1   20     1     1    1420       1
3  47         1       1   30     2     1    1279       1
4  40         1       1   24     1     3     148       0
5  64         2       2   19     2     1    1863       0
6  49         2       2   56     1     3    1933       0
```

The first three patients have developed a recurrence shown by censrec variable equals to 1, unlike the following three, which take the value of 0. This variable, along with the other two, rectime and nodes, will be taken into account for applying the methods described in Section 2.2. The number of

	survclustcurves() arguments
`time`	A vector with variable of interest, i.e. survival time.
`status`	A vector with censoring indicator of the survival time of the process; 0 if the total time is censored and 1 otherwise.
`x`	A vector with categorical variable indicating the population to which the observations belongs.
`kvector`	A vector specifying the number of groups of curves to be checking. By default it is NULL.
`kbin`	Size of the grid over which the survival functions are to be estimated.
`nboot`	Number of bootstrap repeats.
`algorithm`	A character string specifying which clustering algorithm is used, i.e., *K*-means (`'kmeans'`) or *K*-medians (`'kmedians'`).
`alpha`	A numeric value, particularly, the signification level of the hypothesis test.
`cluster`	A logical value. If `TRUE` (default) the code is parallelized. Note that there are cases without enough repetitions (e.g., a low number of initial variables) that R will gain in performance through serial computation. R takes time to distribute tasks across the processors also it will need time for binding them all together later on. Therefore, if the time for distributing and gathering pieces together is greater than the time needed for single-thread computing, it could be better not to parallelize.
`ncores`	An integer value specifying the number of cores to be used in the parallelized procedure. If `NULL`, the number of cores to be used is equal to the number of cores of the machine -1.
`seed`	Seed to be used in the procedure.
`multiple`	A logical value. If TRUE (not default), the resulted pvalues are adjusted by using one of several methods for multiple comparisons.
`multiple.method`	Correction method: `'bonferroni'`, `'holm'`, `'hochberg'`, `'hommel'`, `'BH'`, `'BY'`
	regclustcurves() arguments
`y`	A vector with variable of interest, i.e. response variable.
`x`	A vector with independent variable.
`z`	A vector with categorical variable indicating the population to which the observations belongs.
`kvector`	A vector specifying the number of groups of curves to be checking. By default it is NULL.
`kbin`	Size of the grid over which the survival functions are to be estimated.
`h`	The kernel bandwidth smoothing parameter.
`nboot`	Number of bootstrap repeats.
`algorithm`	A character string specifying which clustering algorithm is used, i.e., *K*-means (`'kmeans'`) or *K*-medians (`'kmedians'`).
`alpha`	A numeric value, particularly, the signification level of the hypothesis test.
`cluster`	A logical value. If `TRUE` (default) the code is parallelized. Note that there are cases without enough repetitions (e.g., a low number of initial variables) that R will gain in performance through serial computation. R takes time to distribute tasks across the processors also it will need time for binding them all together later on. Therefore, if the time for distributing and gathering pieces together is greater than the time needed for single-thread computing, it could be better not to parallelize.
`ncores`	An integer value specifying the number of cores to be used in the parallelized procedure. If `NULL`, the number of cores to be used is equal to the number of cores of the machine -1.
`seed`	Seed to be used in the procedure.
`multiple`	A logical value. If TRUE (not default), the resulted pvalues are adjusted by using one of several methods for multiple comparisons.
`multiple.method`	Correction method: `'bonferroni'`, `'holm'`, `'hochberg'`, `'hommel'`, `'BH'`, `'BY'`

Table 2: Arguments of `survclustcurves` and `regclustcurves`

positive nodes has been grouped from 1 to > 13 because of its low numbers onwards. Below, the steps for this preprocessed are shown:

```
> table(gbcsCS$nodes)

  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
187 110  79  57  41  33  36  20  20  19  15  13  11   3   5   8   5   5   5   3   1
 23  24  26  30  33  35  36  38  51
  1   2   1   1   1   1   1   1   1

> gbcsCS[gbcsCS$nodes > 13,'nodes'] <- 14
> gbcsCS$nodes <- factor(gbcsCS$nodes)
> levels(gbcsCS$nodes)[14]<- '>13'
> table(gbcsCS$nodes)

  1   2   3   4   5   6   7   8   9  10  11  12  13 >13
187 110  79  57  41  33  36  20  20  19  15  13  11  45
```

Estimates of the survival curves are obtained using the survclustcurves function. This function allows determining groups using the optimization algorithm *K*-means or *K*-medians. The function will verify if data has been introduced correctly and will create a 'clustcurves' object. The first three arguments must be introduced, where time is a vector with event-times, status for their corresponding indicator statuses, and x is the categorical covariate.

As we mentioned, note that the proposed procedure may deal with the problem of testing multiple hypotheses, particularly relevant when the categorical variable has many levels. Thus, if the user wants to apply some correction, it is possible to specify multiple = TRUE and select some of the well-known techniques such as Bonferroni, Holm, etc., by means of the argument multiple.method.

The output of this function is the assignment of the survival curves to the group to which they belong and an automatic selection of their number. The following input commands provide an example of this output using the *K*-medians algorithm:

```
> fit.gbcs <- survclustcurves(time = gbcsCS$rectime, status = gbcsCS$censrec,
                x = gbcsCS$nodes, nboot = 500, seed = 300716, algorithm = 'kmedians',
                cluster = TRUE)
Checking 1 cluster...
Checking 2 clusters...
Checking 3 clusters...

Finally, there are 3 clusters.
> summary(fit.gbcs)

Call:
survclustcurves(time = gbcsCS$rectime, status = gbcsCS$censrec,
    x = gbcsCS$nodes, nboot = 500, algorithm = "kmedians", cluster = TRUE,
    seed = 300716)

Clustering curves in 3 groups

Number of observations:  640
Cluster method:  kmedians

Factor's levels:
 [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"  "12"  "13"
[14] ">13"

Clustering factor's levels:
 [1] 1 1 1 3 3 3 3 2 3 2 2 2 2 2

Testing procedure:
  H0   Tvalue pvalue
1  1 95.68626  0.000
2  2 56.03966  0.018
3  3 33.63386  0.830

Available components:
```

```
[1] "num_groups" "table"      "levels"    "cluster"   "centers"   "curves"
[7] "method"     "data"       "algorithm" "call"
```

The graphical representation of the fit can be easily obtained with the function autoplot. Specifying the argument groups_by_color = FALSE, the estimated survival curves for each level of the factor nodes by means of the Kaplan-Meier estimator can be drawn. The assignment of the curves to the three groups can be observed in Figure 1 simply typing groups_by_color = TRUE. As expected, the survival of patients can be influenced by the number of lymph nodes. The patients' recurrence time rises with the decrease of lymph nodes. Note that having 3 or fewer positive nodes seems to be related to higher recurrence-free probabilities. Patients with 9 or more positive nodes are more likely to develop a recurrence. The group of patients with 8 positive nodes was assigned to the group with highest recurrence probabilities. Though this was unexpected, further analysis confirm the poor and unexpected behavior.
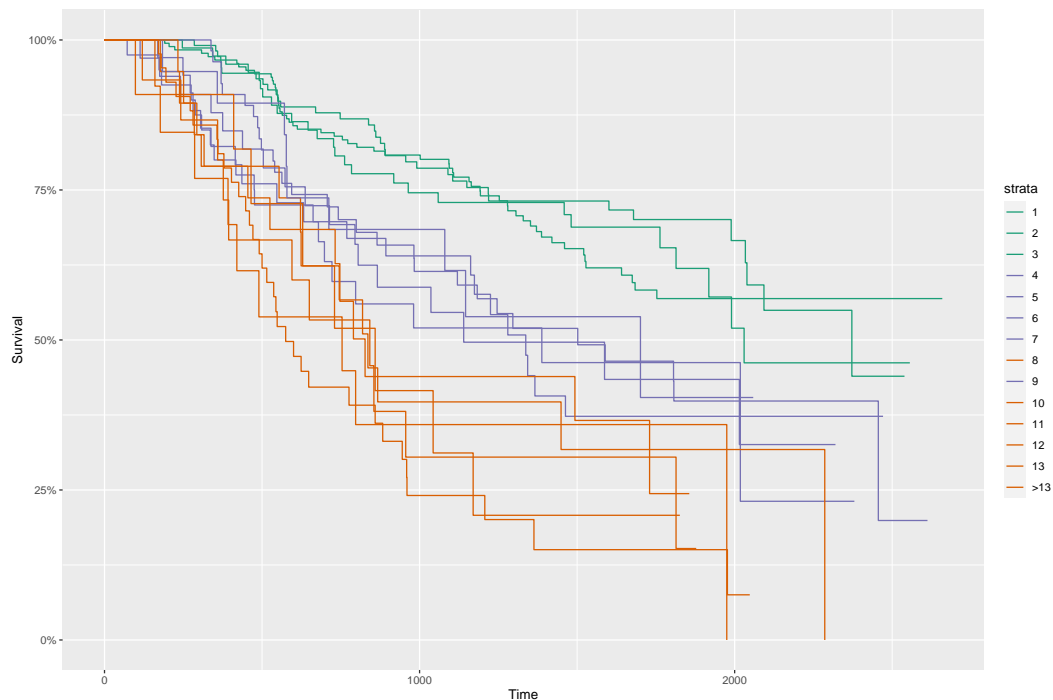
Figure 1: Estimated survival curves for each of the levels of the variable nodes. A specific color is assigned for each curve according to the group to which it belongs using the *K*-medians algorithm (in this case, three groups, *K* = 3).

Equivalently, the following piece of code shows the input commands and the results obtained with the algorithm = 'kmeans'. However, the number of groups and the assignments are different from those obtained with the 'kmedians'. Although this situation is not so common, in some real applications, it can happen.

```
> fit.gbcs2 <- survclustcurves(time = gbcsCS$rectime, status = gbcsCS$censrec,
                x = gbcsCS$nodes, nboot = 500, seed = 300716, algorithm = 'kmeans',
                cluster = TRUE)
Checking 1 cluster...
Checking 2 clusters...

Finally, there are 2 clusters.


> fit.gbcs2

Call:
survclustcurves(time = gbcsCS$rectime, status = gbcsCS$censrec,
    x = gbcsCS$nodes, nboot = 500, algorithm = "kmeans", cluster = TRUE,
    seed = 300716)

Clustering curves in 2 groups
```

```
Number of observations:  607
Cluster method:  kmeans
```

The corresponding plot is shown in Figure 2. Note that having 9 or more positive nodes seems to be related to a lower recurrence-free survival than having 9 or less, with the exception of the survival curve for those patients with 8 positive nodes, which was assigned to the group with highest recurrence probabilities.
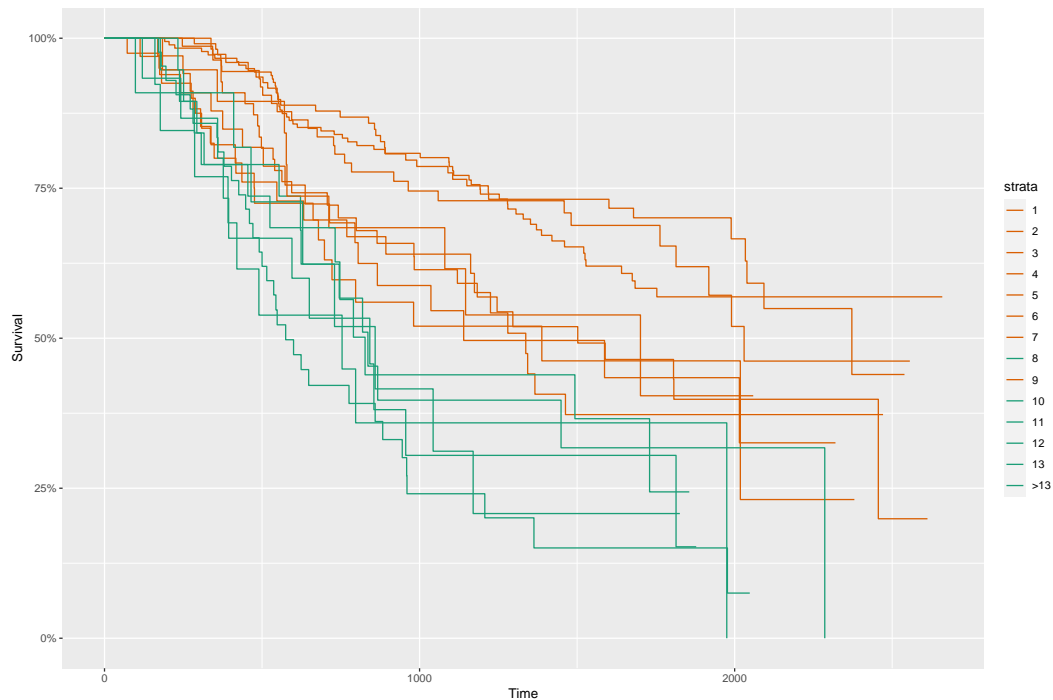
Figure 2: Estimated survival curves for each of the levels of the variable nodes. A specific color is assigned for each curve according to the group to which it belongs using the *K*-means algorithm (in this case, two groups, *K* = 2).

It is important to highlight that given a fixed value of *K*, one may also be interested in determining the group to which each survival function belongs. This is possible by means of the ksurvcurves function by considering, for example, the argument k = 3.

```
> ksurvcurves(time = gbcsCS$rectime, status = gbcsCS$censrec, x = gbcsCS$nodes,
                    seed = 300716, algorithm = 'kmedians', k = 3)

Call:
ksurvcurves(time = gbcsCS$rectime, status = gbcsCS$censrec, x = gbcsCS$nodes,
    k = 3, algorithm = "kmedians", seed = 300716)

Clustering curves in 3 groups

Number of observations:  640
Cluster method:  kmedians
```

More information related to the output above can be obtained running the summary function.

Application to Multiple Myeloma Study Data

In this case, a study of the survival in patients with multiple myeloma (MM) cancer was conducted. 256 individuals were included from the start of the follow-up to whom were analyzed and collected 16 variables. This data set is freely available in the **survminer** package. Below, it is shown the first rows of the data.frame with columns such as treatment (treatment), life state indicator (event; censored – 0; 1 – dead), survival time (time, in months), among others.

```
> library(survminer)
> data(myeloma)
> head(myeloma[,1:5])
          molecular_group chr1q21_status treatment event  time
GSM50986      Cyclin D-1       3 copies      TT2      0 69.24
GSM50988      Cyclin D-2       2 copies      TT2      0 66.43
GSM50989           MMSET       2 copies      TT2      0 66.50
GSM50990           MMSET       3 copies      TT2      1 42.67
GSM50991             MAF           <NA>      TT2      0 65.00
GSM50992     Hyperdiploid       2 copies      TT2      0 65.20
```

In this example, it is interesting to analyze if the survival in patients with MM disease is the same for the different molecular subgroups. If there is an effect of the molecular subgroups on the survival, future therapies that might exploit molecular insights should lead to an improvement in outcome for patients with these types of disease (Zhan et al., 2006).

Below, a summary of the results of the survclustcurves function obtained with time, event, and molecular_group as input variables and for both kmedians and kmeans algorithms are shown.

```
> fit.mye <- survclustcurves(time = myeloma$time, status = myeloma$event,
                        x = myeloma$molecular_group, nboot = 500, seed = 300716,
                        algorithm = 'kmedians', cluster = TRUE)
Checking 1 cluster...
Checking 2 clusters...

Finally, there are 2 clusters.
> summary(fit.mye)

Call:
survclustcurves(time = myeloma$time, status = myeloma$event,
    x = myeloma$molecular_group, nboot = 500, algorithm = "kmedians",
    cluster = TRUE, seed = 300716)

Clustering curves in 2 groups

Number of observations:  248
Cluster method:  kmedians

Factor's levels:
[1] "Cyclin D-1"      "Cyclin D-2"       "Hyperdiploid"     "Low bone disease"
[5] "MAF"             "MMSET"            "Proliferation"

Clustering factor's levels:
[1] 1 1 1 1 1 2 2

Testing procedure:
  H0   Tvalue pvalue
1  1 31.31603  0.026
2  2 14.94269  0.682

Available components:
 [1] "num_groups" "table"      "levels"     "cluster"    "centers"    "curves"
 [7] "method"     "data"       "algorithm"  "call"

> fit.mye2 <- survclustcurves(time = myeloma$time, status = myeloma$event,
                        x = myeloma$molecular_group, nboot = 500, seed = 300716,
                        algorithm = 'kmeans', cluster = TRUE)
Checking 1 cluster...
Checking 2 clusters...

Finally, there are 2 clusters.
```

```
> summary(fit.mye2)

Call:
survclustcurves(time = myeloma$time, status = myeloma$event,
    x = myeloma$molecular_group, nboot = 500, algorithm = "kmeans",
    cluster = TRUE, seed = 300716)

Clustering curves in 2 groups

Number of observations:  248
Cluster method:  kmeans

Factor's levels:
[1] "Cyclin D-1"       "Cyclin D-2"       "Hyperdiploid"      "Low bone disease"
[5] "MAF"              "MMSET"            "Proliferation"

Clustering factor's levels:
[1] 1 1 1 1 1 2 2

Testing procedure:
  H0   Tvalue pvalue
1  1 4.500272  0.032
2  2 1.108812  0.730

Available components:
 [1] "num_groups" "table"      "levels"     "cluster"    "centers"    "curves"
 [7] "method"     "data"       "algorithm"  "call"
```

When comparing the results obtained through the two methods (kmeans, kmedians), it is seen that the obtained number of clusters is the same (2 groups), even the assignment of the curves to the groups.

In particular, results obtained reveal that MMSET level and Proliferation level are associated with a high-risk or damage on the lifetime, while MAF, Low bone disease, Hyperdiploid, Cycline D-1, and Cycline D-2 have higher survival probabilities. This is observed in the plot shown in Figure 3, which can be obtained using the following input command:

```
> autoplot(fit.mye, groups_by_color = TRUE)
```

Application to Barnacle's Growth Study Data

This study was conducted on the Atlantic coast of Galicia (Northwest Spain), which consists of an approximately 1000km long shoreline with extensive rocky stretches exposed to tidal surges and wave action that are settled by the *Pollicipes pollicipes* (Gmelin, 1789) populations targeted for study. A total of 5000 specimens were collected from five sites of the region's Atlantic coastline and corresponded to the stretches of coast where this species is harvested: Punta do Mouro, Punta Lens, Punta de la Barca, Punta del Boy and Punta del Alba. Two biometric variables of each specimen were measured: RC (Rostro-carinal length, maximum distance across the capitulum between the ends of the rostral and carinal plates) and DW (Dry Weight). This data set (barnacle5) is available in the **clustcurv** package. The idea of this study is to know the relation between RC and DW variables along the coast, i.e., to analyze if the barnacle's growth is similar in all locations or by contrast, if it is possible to detect geographical differentiation in growth. A sample of the dataset is shown as follow:

```
> data("barnacle5")
> head(barnacle5)
    DW   RC    F
1 0.52 12.0 laxe
2 1.46 18.9 laxe
3 0.05  6.4 laxe
4 0.17  9.4 laxe
5 0.05  6.2 laxe
6 0.41 12.2 laxe
```
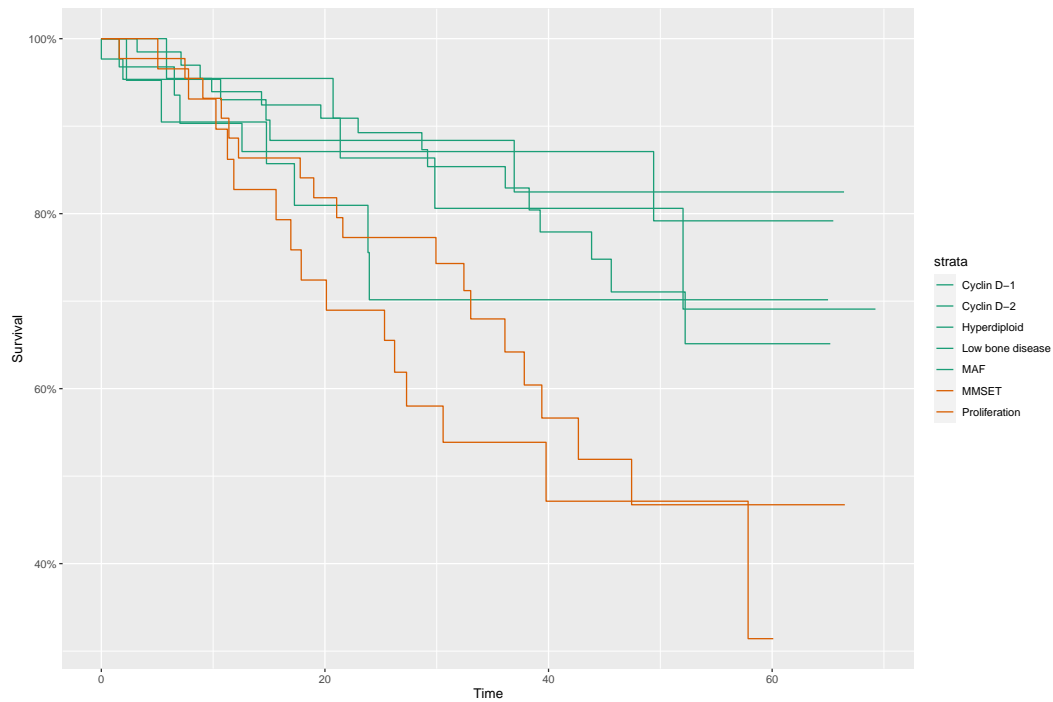
Figure 3: Estimated survival curves for each of the levels of the variable `molecular group`. A specific color is assigned for each curve according to the group to which it belongs using the *K*-medians algorithm, two groups, *K* = 2.

For each location (F), nonparametric regression curves were estimated to modeling the dependence between RC and DW. In order to determine groups, we used the proposed methodology in Subsection 2.2.2. Through executing the next piece of code, the following results can be obtained: one estimated curve was attributed to the first group (Punta Lens), two estimated curves were assigned to group 2 (Punta de la Barca and Punta del Boy), and the other two belong to group 3 (Laxe do Mouro and Punta del Alba) (Figure 4). In this example, the `regclustcurves` function was used with `algorithm = 'kmeans'` and the input variables y, x, z.

```
> fit.bar <- regclustcurves(y = barnacle5$DW, x = barnacle5$RC, z = barnacle5$F,
              nboot = 500, seed = 300716, algorithm = 'kmeans', cluster = TRUE)

Checking 1 cluster...
Checking 2 clusters...
Checking 3 clusters...

Finally, there are 3 clusters.

> summary(fit.bar)

Call:
regclustcurves(y = barnacle5$DW, x = barnacle5$RC, z = barnacle5$F,
    nboot = 500, algorithm = "kmeans", cluster = TRUE, seed = 300716)

Clustering curves in 3 groups

Number of observations:  5000
Cluster method:  kmeans

Factor's levels:
[1] "laxe"  "lens"  "barca" "boy"   "alba"

Clustering factor's levels:
[1] 2 1 3 3 2
```

```
Testing procedure:
  H0     Tvalue pvalue
1  1 0.94353014  0.000
2  2 0.15463483  0.034
3  3 0.02348982  0.422

Available components:
 [1] "num_groups" "table"      "levels"    "cluster"   "centers"   "curves"
 [7] "method"     "data"       "algorithm" "call"
```

As can be seen, Figure 4 obtained using the following input command. The specimens from Punta de la Barca and Punta del Boy have similar morphology, wide and short. This is due to these zones present similar oceanographic characteristic, such as exposed rocky shore to waves and highly articulated. Unlike, the barnacles collected from Laxe do Moure and Punta del Alba are narrow and long because they are less exposed locations. Finally, Punta Lens is an intermediate coast, alternating sections of steep coast with large sand.

```
> autoplot(fit.bar, groups_by_color = TRUE)
```
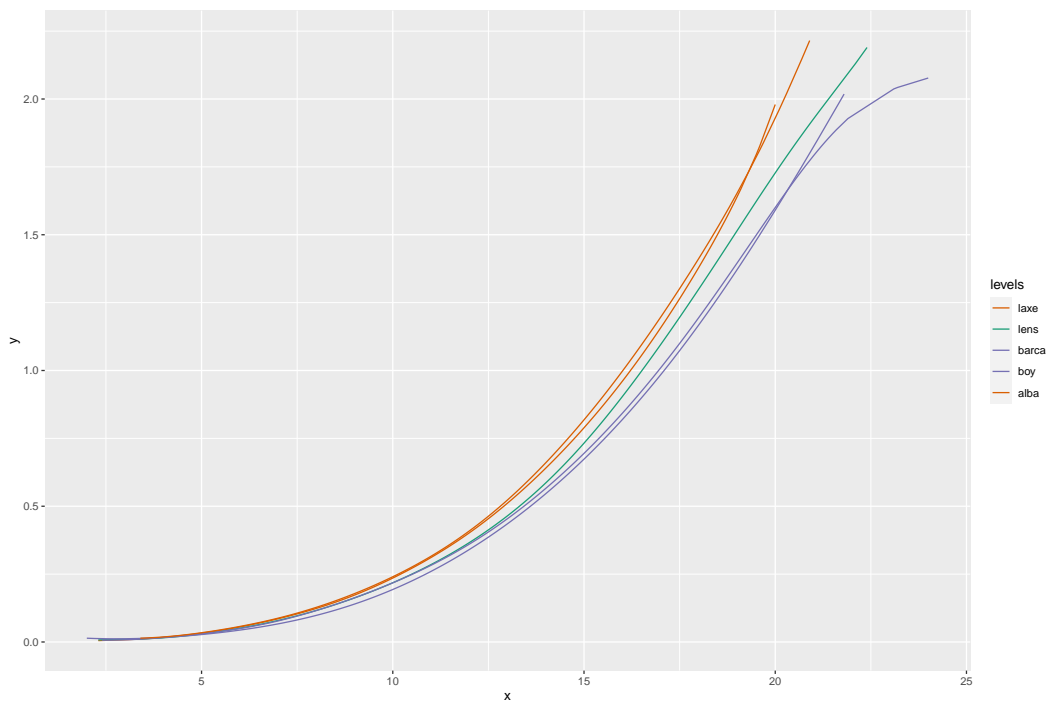

Figure 4: Estimated regression curves for each of the levels of the factor. A specific color is assigned for each curve according to the group to which it belongs using the *K*-means algorithm (in this case, three groups, *K* = 3).

Application to simulated data

Finally, this subsection reports the capabilities of the **clustcurv** package in a more complicated simulated scenario. We consider the regression models given in (1) for $j = 1, \ldots, 30$, with

$$
m_j(X_j) = \begin{cases}
X_j + 1 & \text{if} \quad j \le 5 \\
X_j^2 + 1 & \text{if} \quad 5 < j \le 10 \\
2 \sin(2\,X_j) & \text{if} \quad 10 < j \le 15 \\
2 \sin(X_j) & \text{if} \quad 15 < j \le 20 \\
2 \sin(X_j) + a\,\mathrm{e}^{X_j} & \text{if} \quad 20 < j \le 25 \\
1 & \text{if} \quad j > 25,
\end{cases}
\tag{3}
$$

where a is a real constant, X_j is the explanatory covariate drawn from a uniform distribution $[-2, 2]$, and ε_j is the error distributed in accordance to a $N(0, \sigma_j(x))$. We have considered the heteroscedastic scenario where $\sigma_j(x) = 0.5 + 0.05 m_j(x) \cdot N(0, 1.5)$.

We explore the methodology considering the null hypothesis $H_0(5)$ of assignment of the m_j curves into five groups ($K = 5$). To show the performance of our procedure, two values were considered for a, 0 and 0.4. It should be noted that the value $a = 0$ corresponds to the null hypothesis, while if $a = 0.4$ the number of groups is six. Particularly, we have defined an unbalanced scenario with unequal sample sizes for each j curve, particularly, $(n_1, n_2, \ldots, n_J) \sim \text{Multinomial}(n; p_1, p_2, \ldots, p_J)$ being $p_j = p_j^* / \sum_{j=1}^{J} p_j^*$, with p_j^* randomly drawn from $\{1, 1.5, 2, 2.5, 3\}$ and $n = 5000$. Note that we propose this procedure for generating the n_j in order to obtain a completely unbalanced study.

The code for the generation of this dataset with $a = 0$ can be found below:

```
> m <- function(x, j) {
    y <- numeric(length(x))
    y[j <= 5] <- x[j <= 5] + 1
    y[j > 5 & j <= 10] <- x[j > 5 & j <= 10] ^ 2 + 1
    y[j > 10 & j <= 15] <- 2 * sin(2 * x[j > 10 & j <= 15]) #- 4
    y[j > 15 & j <= 20] <- 2 * sin(x[j > 15 & j <= 20])
    y[j > 20 & j <= 25] <- 2 * sin(x[j > 20 & j <= 25]) + a * exp(x[j > 20 & j <= 25])
    y[j > 25] <- 1
    return(y)
 }


> seed <- 300716
> set.seed(seed)
> n <- 5000
> a <- 0.0
> x <- runif(n, -2, 2)
> prob <- sample(c(1, 1.5, 2, 2.5, 3), 30, replace = TRUE)
> prob <- prob/sum(prob)
> f <- sample(1:30, n, replace = TRUE, prob = prob)
> N <- length(unique(f))
> error <- rnorm(n,0,1.5)
> y <- m(x, f) + (0.5 + 0.05 * m(x, f)) * error
> data <- data.frame(x, y, f)
```

In order to determine groups of the generated curves, the user has to execute the next piece of code. As expected, when $a = 0$, the number of groups selected is five.

```
> fit.sim <- regclustcurves(x = data$x, y = data$y,  z = data$f, nboot = 500,
                    algorithm = 'kmedians', cluster = TRUE, seed = 300716)
Checking 1 cluster...
Checking 2 clusters...
Checking 3 clusters...
Checking 4 clusters...
Checking 5 clusters...

Finally, there are 5 clusters.
> fit.sim

Call:
regclustcurves(y = data$y, x = data$x, z = data$f, nboot = 500,
    algorithm = "kmedians", cluster = TRUE, seed = 300716)

Clustering curves in 5 groups

Number of observations:  5000
Cluster method:  kmedians
> autoplot(fit.sim, groups_by_colour = TRUE, centers = TRUE)
```

Additionally, for different values of a ($a > 0$), our procedure should determine 6 groups. For instance, for $a = 0.4$, it selects the true number of groups ($K = 6$) typing the commands below:

```
> seed <- 300716
> set.seed(seed)
> n <- 5000
> a <- 0.4
> x <- runif(n, -2, 2)
```

```
> prob <- sample(c(1, 1.5, 2, 2.5, 3), 30, replace = TRUE)
> prob <- prob/sum(prob)
> f <- sample(1:30, n, replace = TRUE, prob = prob)
> N <- length(unique(f))
> error <- rnorm(n,0,1.5)
> y <- m(x, f) + (0.5 + 0.05 * m(x, f)) * error
> data2 <- data.frame(x, y, f)
> fit.sim2 <- regclustcurves(x = data2$x, y = data2$y,  nboot = 500, seed = 300716,
                             z = data$f, algorithm = 'kmedians', cluster = TRUE)
Checking 1 cluster...
Checking 2 clusters...
Checking 3 clusters...
Checking 4 clusters...
Checking 5 clusters...
Checking 6 clusters...

Finally, there are 6 clusters.
> fit.sim2

Call:
regclustcurves(y = data2$y, x = data2$x, z = data$f, nboot = 500,
    algorithm = "kmedians", cluster = TRUE, seed = 300716)

Clustering curves in 6 groups

Number of observations:  5000
Cluster method:  kmedians
> autoplot(fit.sim2, groups_by_colour = TRUE, centers = TRUE)
```

Figures 5 and 6 show the results with the simulated data with $a = 0$ and $a = 0.4$, respectively. In this situation, the true number of groups is equal to 5 and 6. As can be appreciated, our method seems to perform reasonably well for both values of a. For $a = 0$, the null hypothesis $H_0(5)$ is accepted, curves assigned to each group are plotted with the same color. In the case of $a = 0.4$, the null hypothesis $H_0(6)$ is accepted. Therefore, there are 6 groups of regression curves. Note that in both plots, the centroids are colored in black because in the autoplot function, the argument centers = TRUE.

## Conclusion and further extensions of the R package

This paper discussed the implementation of some methods developed for determining groups of multiple nonparametric curves in the R package **clustcurv**. In particular, the methods proposed are focused on the framework of regression analysis and in the framework of survival analysis. In the context of survival analysis, we restrict ourselves to survival curves. Hopefully, future versions of the package will extend the methodology to determine groups in risk functions, cumulative hazard curves, or density functions. The current version of the package implements two optimization algorithms, the well-known K-means and K-medians. It can be interesting to let the user choose far from those, such as Means-Shift or K-medoids algorithms.

Acknowledgements

Bibliography

Z. Chen and G. Zhang. Comparing survival curves based on medians. *BMC Medical Research Methodology*, 16(1):33, 2016. ISSN 1471-2288. doi: 10.1186/s12874-016-0133-3. URL http://dx.doi.org/10.1186/s12874-016-0133-3. [p164]
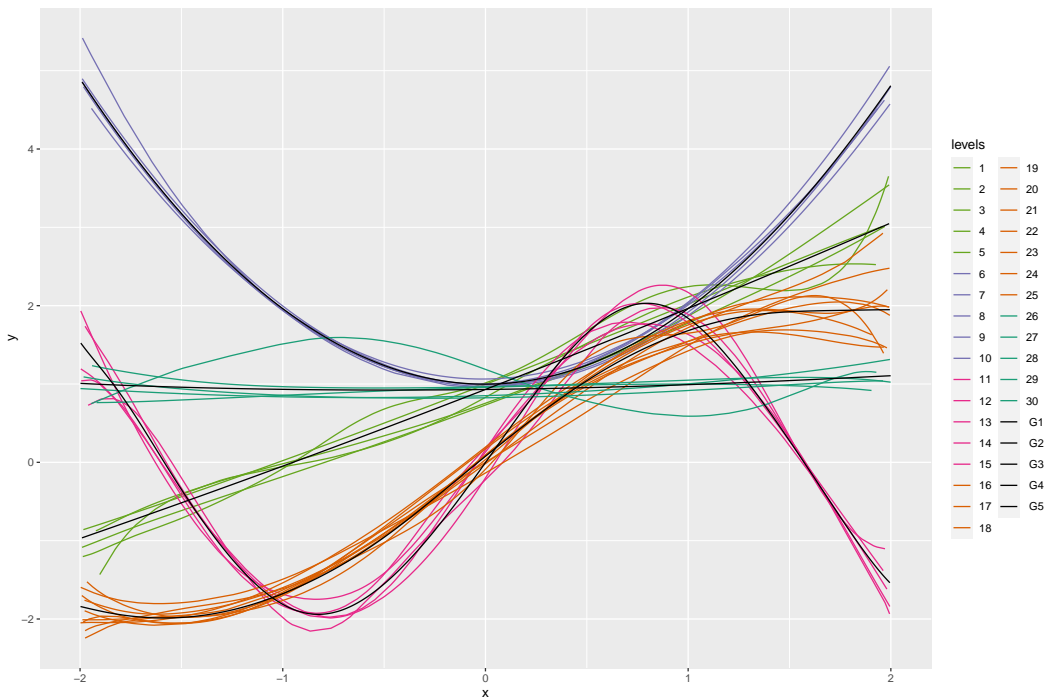
Figure 5: Estimated regression curves for each of the levels of the variable f with $a = 0$. A specific color is assigned for each curve according to the group to which it belongs using the *K*-means algorithm (in this case, five groups, $K = 5$). Black curves are the centroids of each group.
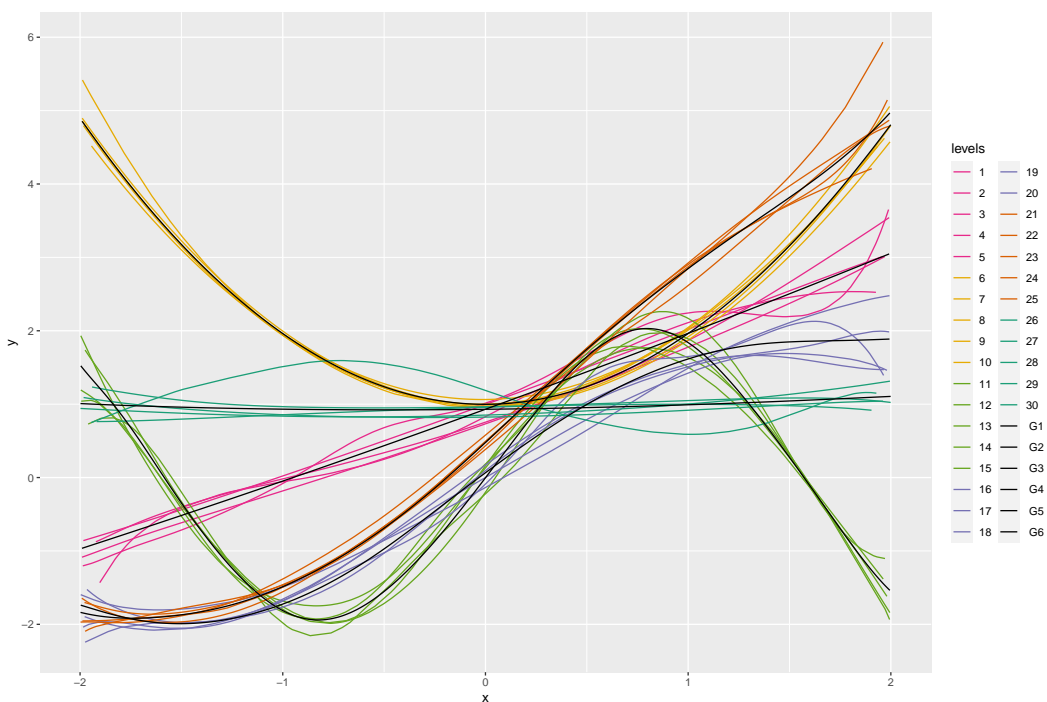
Figure 6: Estimated regression curves for each of the levels of the variable f with $a = 0.4$. A specific color is assigned for each curve according to the group to which it belongs using the *K*-means algorithm (in this case, six groups, $K = 6$). Black curves are the centroids of each group.

C. A. de Boor. *A Practical Guide to Splines*. Springer Verlag, New York, 2001. [p166]

M. A. Delgado. Testing the equality of nonparametric regression curves. *Statistics and Probability Letters*, 17:199–204, June 1993. [p164]

D. Dette and N. Neumeyer. Nonparametric analysis of covariance. *The Annals of Statistics*, 29:1361–1400, 2001. [p164]

S. Dudoit and M. J. van der Laan. *Multiple Testing Procedures with Applications to Genomics*. Springer Series in Statistics. Springer, New York, 2008. [p167]

B. Efron. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 7:1–26, 1979. [p167]

J. Fan and I. Gijbels. *Local polynomial modelling and its applications*. Number 66 in Monographs on statistics and applied probability series. Chapman & Hall, 1996. [p166]

T. R. Fleming, D. P. Harrington, and M. O'Sullivan. Supremum versions of the log-rank and generalized wilcoxon statistics. *Journal of the American Statistical Association*, 82(397):312–320, 1987. ISSN 01621459. URL http://www.jstor.org/stable/2289169. [p164]

E. A. Gehan. A generalized wilcoxon test for comparing arbitrarily singly censored samples. *Biometrika*, 52:203–223, 1965. ISSN 00063444. URL http://www.jstor.org/stable/2333825. [p164]

W. González-Manteiga and R. Cao. Testing the hypothesis of a general linear model using nonparametric regression estimation. *Test*, 2(1):223–249, 1993. [p164]

P. Hall and J. D. Hart. Bootstrap test for difference between means in nonparametric regression. *Journal of the American Statistical Association*, 85(412):1039–1049, Dec. 1990. [p164]

W. Härdle and E. Mammen. Testing parametric versus nonparametric regression. *Annals of Statistics*, 21:1926–1947, 1993. [p164]

D. P. Harrington and T. R. Fleming. A class of rank test procedures for censored survival data. *Biometrika*, 69(3):553, 1982. doi: 10.1093/biomet/69.3.553. URL +http://dx.doi.org/10.1093/biomet/69.3.553. [p164]

S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6: 65–70, 1979. [p168]

E. L. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53:457–481, 1958. [p165]

A. Kassambara, M. Kosinski, P. Biecek, and S. Fabian. survminer: Drawing Survival Curves using 'ggplot2'. R package version 0.4.6, 2019. URL https://cran.r-project.org/web/packages/survminer. [p164]

L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley, 1990. [p166]

E. King, J. D. Hart, and T. E. Wehrly. Testing the equality of two regression curves using linear smoothers. *Statistics and Probability Letters*, 12(3):239–247, 1991. ISSN 0167-7152. [p164]

K. B. Kulasekera. Comparison of regression curves using quasi-residuals. *Journal of the American Statistical Association*, 90(431):1085–1093, 1995. ISSN 01621459. [p164]

S. Lang and A. Brezger. Bayesian p-splines. *Journal of Computational and Graphical Statistics*, 13:183–212, 2004. [p166]

R. Y. Liu. Bootstrap Procedures under some Non-I.I.D. Models. *The Annals of Statistics*, 16(4):1696–1708, 1988. URL http://www.jstor.org/stable/2241788. [p167]

J. B. Macqueen. *Some methods of classification and analysis of multivariate observations*, volume 1. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (Univ. of Calif. Press), 1967. [p166]

E. Mammen. Bootstrap and Wild Bootstrap for High Dimensional Linear Models. *The Annals of Statistics*, 21(1):255–285, 1993. URL http://www.jstor.org/stable/3035590. [p167]

N. Mantel. Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer Chemotherapy Reports*, 50(3):163–170, 1966. [p164]

N. Neumeyer and H. Dette. Nonparametric comparison of regression curves: An empirical process approach. *The Annals of Statistics*, 31(3):880–920, 2003. ISSN 00905364. [p164]

J. C. Pardo-Fernández, I. Van Keilegom, and W. González-Manteiga. Testing for the equality of k regression curves. *Statistica Sinica*, 17:1115–1137, 2007. [p164]

R. Peto and J. Peto. Asymptotically efficient rank invariant test procedures (with discussion). *Journal of the Royal Statistical Society, Series A*, 135:185–206, 1972. [p164]

M. Rosenblatt. A quadratic measure of deviation of two-dimensional density estimates and a test of independence. *The Annals of Statistics*, 3(1):1–14, 01 1975. [p164]

R. Srihera and W. Stute. Nonparametric comparison of regression functions. *Journal of Multivariate Analysis*, 101:2039–2059, October 2010. ISSN 0047-259X. [p164]

R. E. Tarone and J. Ware. On distribution-free tests for equality of survival distribution. *Biometrika*, 64: 156–160, 1977. [p164]

N. M. Villanueva, M. Sestelo, and L. Meira-Machado. A Method for Determining Groups in Multiple Survival Curves. *Statistics in Medicine*, 38:366–377, 2019a. [p164, 165]

N. M. Villanueva, M. Sestelo, C. Ordoñez, and J. Roca-Pardiñas. An Approach to Determine Groups of Multiple Regression Curves. 2019b. [p165]

M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman & Hall: London, 1995. [p166]

H. Wickham, W. Chang, L. Henry, T. L. Pedersen, K. Takahashi, C. Wilke, K. Woo, and H. Yutani. ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics. R package version 3.2.1, 2019. URL https://cran.r-project.org/web/packages/ggplot2. [p169]

C. F. J. Wu. Jackknife, Bootstrap and other resampling methods in regression analysis. *The Annals of Statistics*, 14(4):1261–1295, 1986. doi: 10.2307/2241454. URL http://dx.doi.org/10.2307/2241454. [p167]

S. G. Young and A. W. Bowman. Nonparametric analysis of covariance. *Biometrics*, 51:920–931, 1995. [p164]

F. Zhan, Y. Huang, S. Colla, J. P. Stewart, I. Hanamura, S. Gupta, J. Epstein, S. Yaccoby, J. Sawyer, B. Burington, E. Anaissie, K. Hollmig, M. Pineda-Roman, G. Tricot, F. van Rhee, R. Walker, M. Zangari, J. Crowley, B. Barlogie, and J. Shaughnessy, John D. The molecular classification of multiple myeloma. *Blood*, 108(6):2020–2028, 09 2006. ISSN 0006-4971. doi: 10.1182/blood-2005-11-013458. [p174]

Nora M. Villanueva
Department of Statistics and OR,
SiDOR research group & CINBIO
University of Vigo, Spain
ORCiD: 0000-0001-8085-2745
http://noramvillanueva.github.io
nmvillanueva@uvigo.es

Marta Sestelo
Department of Statistics and OR,
SiDOR research group & CINBIO
University of Vigo, Spain
ORCiD: 0000-0003-4284-6509
http://sestelo.github.io
sestelo@uvigo.es

Luís Meira-Machado
Department of Mathematics
Centre of Mathematics
University of Minho, Portugal
lmachado@math.uminho.pt

Javier Roca-Pardiñas
Department of Statistics and OR,
SiDOR research group & CINBIO
University of Vigo, Spain
roca@uvigo.es