MASTER'S DEGREE THESIS

Interuniversity Master in Statistics and Operations Research UPC-UB

Title: Testing Goodness-of-Fit of Parametric Survival Models for Right-Censored Data

Author: Mireia Besalú i Mayol

Advisor: Guadalupe Gómez Melis

Co-Advisor: Klaus Gerhard Langohr

Department: Estadística i Investigació Operativa

University: Universitat Politècnica de Catalunya-Universitat de Barcelona

Academic year: 2015-2016

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈNCIA DE CATALUNYA

FACULTAT DE MATEMÀTIQUES I ESTADÍSTICA

MASTER'S DEGREE THESIS

Testing Goodness-of-Fit of Parametric Survival Models for Right-Censored Data

Author:
Mireia BESALÚ MAYOL

Advisor:
Guadalupe Gómez Melis
Co-Advisor:
Klaus Gerhard Langohr

A thesis submitted in fulfilment of the requirements for the degree of Master degree

in the

Department Estadística i Investigació Operativa

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Abstract

Master Degree Thesis

Testing Goodness-of-Fit of Parametric Survival Models for Right-Censored Data

by Mireia BESALÚ MAYOL

The main goal of this work it is to present a review of the existing methods to deal with the goodness-of-fit for right-censored data. Goodness-of-fit tests are developed to assess whether a given distribution is suited to a data set. Literature on goodness-of-fit tests for right-censored data is scarce and scattered.

This master's degree thesis is divided into three different parts. The first part is devoted to review the bibliography of goodness-of-fit test for parametric models with right-censored data. We classify them according to the type of censoring and the methodology used, and we also propose a unified notation. The second part it focuses on the theoretic explanation of the Generalized Chi Squared test presented by Moore and Spruill (1975) [35] and Kim (1993) [27]. The first authors present a Chi-Squared test that included almost all the different known Chi-Squared tests for complete data. The second author extend the result to random right-censored data. Finally, the last part of the work presents an implementation in **R** of the Generalized Chi-Squared test for complete and right-censored data. We also have applied the above methods to some data sets and we have analyzed the results.

This work follows the master's degree thesis of Anna Febrer (2015) [17], who studied and implemented graphic methods to assess the goodness-of-fit for right-censored data. The aim it is to compile the work to create a local library in R for goodness-of-fit tests for right-censored data.

Keywords: Goodness-of-fit, right-censored, chi-squared, implementation

MSC2010: 62N01, 62N03

Resum

Treball final de Màster

Testing Goodness-of-Fit of Parametric Survival Models for Right-Censored Data

de Mireia BESALÚ MAYOL

L'objectiu principal d'aquest treball és presentar una revisió dels mètodes existents per tractar amb la bondat d'ajustament per dades censurades per la dreta. Els tests de bondat d'ajustament s'utilitzen per validar si una distribució donada s'ajusta a un conjunt de dades. La literatura de tests de bondat d'ajustament per dades censurades per la dreta és escassa i dispersa.

Aquest treball de màster està dividit en tres parts diferents. La primera part està dedicada a revisar la bibliografia de tests de bondat d'ajustament per models paramètrics amb dades censurades per la dreta. Hem classificat aquests tests d'acord al tipus de censura i a la metodologia utilitzada, i hem proposat una notació unificada. La segona part del treball està focalitzada en l'explicació teòrica del test de Chi-quadrat Generalitzat presentat per Moore i Spruill (1975) [35] i Kim (1993) [27]. Els primers autors presenten un test de Chi-quadrat que inclou quasi tots els tests de Chi-quadrat coneguts per a dades completes. El segon autor estén el resultat a dades censurades per la dreta. Finalment, en l'última part implementem a **R** el test Chi-quadrat Generalitzat per a dades completes i per a dades censurades per la dreta. També apliquem aquests mètodes a alguns conjunts de dades i n'analitzem els resultats.

Aquest treball segueix el treball de màster de l'Anna Febrer (2015) [17], que va estudiar i implementar mètodes gràfics per validar la bondat d'ajustament per dades censurades per la dreta. L'objectiu serà compilar aquest treball per crear una llibreria en R per tests de bondat d'ajustament per dades censurades per la dreta.

Paraules clau: Bondat d'ajustament, censura per la dreta, Chi-quadrat, implementació

MSC2010: 62N01, 62N03

Acknowledgements

A la Lupe i en Klaus, per ensenyar-me que tots els textos matemàtics, per complicats, tècnics i amb molta notació es poden entendre i explicar senzillament, encara que facin falta unes quantes iteracions.

A en Bernat, per ser-hi sempre, per recolzar-me i al mateix temps instar-me a superar-me, per fer-me pensar i anar més enllà amb cada raonament.

A l'Aina, perquè juntes podem amb el que sigui.

A tots els amics i companys que han estat sempre al meu costat i que m'han ajudat a continuar amb aquest màster en els moments més dificils. MOLTES GRÀCIES, sou els millors!

Contents

Chapter 1

Introduction

One of the main problems in statistics it is to describe the probability behavior of a sample of observations. Although there are many different choices our goal it is to find the distribution that fits best to the data. Our aim is to study the test procedures, with the objective of checking whether the chosen distribution is good enough; these tests are known as **goodness-of-fit tests**.

In survival analysis the outcome variable of interest is the time until an event occurs, we refer to it as **survival time**. Most of the times, this type of data is **censored**, that means we have information about survival time of the individual, but we don't know exactly the survival time. More precisely, the data can be **right censored**, when we know that the exact survival time becomes incomplete at the right side. Right censoring might be due to

- Loss to follow-up. We only have observed the individuals during part of the survival time.
- Drop-out. It occurs when we interrupt the treatment or observation of an individual, for example due to intolerance of the treatment.
- Study termination. In that case, the study ends before we can observe the event of interest.

This type of data often occurs in health sciences: time to death or time to relapse of a disease, in reliability studies: time until a machine part fails, or in social sciences: life times of elderly in particular social programs.

A proper definition of survival time T, a non-negative random variable, requires:

- An unambiguous time origin.
- A time scale we will use.
- A clear definition of the event of interest.

Our aim in this work is to review, discuss and implement, some goodness-of-fit test for complete and right censored data.

1.1 Outline

This work is divided in three main Chapters and four Appendices.

In Chapter 1, we introduce the main goal of the work as well as some unified notation for the other Chapters. The main part of this Chapter is devoted to review the literature

of goodness-of-fit tests for complete and right-censored data. The review is divided in sections to classify the different tests presented by the method used.

In Chapter 2, the Generalized Chi-Squared test by Moore and Spruill (1975) [35] for complete data and by Kim (1993) [27] for right-censored data it is explained in detail.

Finally in Chapter 3 we explain the use and the results of the implementation in **R** of the tests in Chapter 2. The implementation is done for complete and right censored data.

We conclude the work with the conclusions and the further work, and the appendices containing the hypotheses necessaries for the results in Chapter 2 and the code of the functions explained in Chapter 3.

1.2 Right-censored data and notation

When we have right censored data, maybe we do not have the survival time of the individual but we want to capture all the available information about the individual. For example, in a clinical trial if we finish a two years study, for all the individuals who do not present the disease at the end of the study we know the survival time (in months) will be in some point in the interval $(24, \infty)$. We always assume non-informative censoring, subjects who drop out of the study should do because of reasons unrelated to the study.

This fact brings us to define T as the survival time, a non-negative variable, that can either be continuous (taking values on $(0, \infty)$) or discrete (taking a finite set of values). But, we also need another non-negative variable to keep the information about the censoring. We will set C to denote the time to censoring, we suppose that the individual will stop at C if the event has not occurred before.

For each individual i we can define the survival time T_i and a censoring time C_i. We will consider the case that censoring is independent (non-informative), so T_i is independent of C_i. We will assume that $T_1, \ldots, T_n$ (n will be the number of individuals) are independent and identically distributed with unknown distribution function F. Then, what we will observe is a value Y_i which will be the minimum between T_i and C_i, $Y_i = \min(T_i, C_i)$ together with an indicator of censoring δ_i, which will be 1 if the Y_i is survival time and 0 if it is the time to censoring

$$\delta_i = \begin{cases} 1, & \text{if } T_i \leq C_i \\ 0, & \text{if } T_i > C_i. \end{cases}$$

The data will be usually presented in pairs (Y_i, δ_i), for $i = 1, \ldots, n$.

Not all the data present the same type of right censoring, in fact we can distinguish three different types of right censoring. Now, we will present them:

- **Type I:** This type of censoring is characterized by a fixed censoring time. We will call C_R the censoring time, the same for all the individuals. That implies, the number of observed events will be random, we do not know how many events have occurred until the end of the study. In this type of censoring we can distinguish two different kinds of censoring depending on the moment of entrance at the study.

  - **Fixed:** In this case, all the individuals enter at the study at the same time. So, we observe $(Y_i,\, \delta_i)$ for $i = 1, \ldots, n$

$$Y_i = \min(T_i,\, C_R) \qquad \text{and} \qquad \delta_i = \begin{cases} 1, & \text{if } T_i \leq C_R \\ 0, & \text{if } T_i > C_R. \end{cases}$$

  - **Generalized:** Now, we let the entrance time of the individuals be different, we denote it by $\mathcal{O}_i$, since the end of the study is fixed C_R, each individual will have a different censoring time $C_i = C_R - \mathcal{O}_i$. Usually in that case we re-scaled the survival time, to consider the same entrance time for all the individuals. T_i will be the re-scaled survival time. So with the previous considerations, we observe $(Y_i,\, \delta_i)$ for $i = 1, \ldots, n$

$$Y_i = \min(T_i,\, C_i) \qquad \text{and} \qquad \delta_i = \begin{cases} 1, & \text{if } T_i \leq C_i \\ 0, & \text{if } T_i > C_i. \end{cases}$$

- **Type II:** The main point of this kind of censoring is that the number of events we want to observe is fixed, $r < n$. The study ends when we have observed r events. So, if we consider $T_{(1)}, T_{(2)}, \ldots, T_{(n)}$ the ordered survival times, the random censoring time $C = T_{(r)}$, it is the same for all the individuals. So, we observe $(Y_i,\, \delta_i)$ for $i = 1, \ldots, n$

$$Y_i = \min(T_i,\, T_{(r)}) \qquad \text{and} \qquad \delta_i = \begin{cases} 1, & \text{if } T_i \leq T_{(r)} \\ 0, & \text{if } T_i > T_{(r)}. \end{cases}$$

- **Random censoring:** Here, the survival time and the censoring time are both random variables. As we have state before, $T_1, \ldots, T_n$ are independent and identically distributed with unknown distribution function F and in the same way $C_1, \ldots, C_n$ are independent and identically distributed with unknown distribution function G. Also, T_i and C_i for $i = 1, \ldots, n$ are independent. So, we observe $(Y_i,\, \delta_i)$ for $i = 1, \ldots, n$

$$Y_i = \min(T_i,\, C_i) \qquad \text{and} \qquad \delta_i = \begin{cases} 1, & \text{if } T_i \leq C_i \\ 0, & \text{if } T_i > C_i. \end{cases}$$

In the following table, we summarize the main differences between the different type of censoring.

	Censoring time	Num. observed events	Entrance moment
Type I	Fixed C_R	Random	At the same time (Fixed) It can be different (Gen.)
Type II	Random	Fixed $r < n$	At the same time
Random	Random	Random	It can be different

TABLE 1.1: Differences between the types of censoring.

1.3 State of the art

In this section we aim to review the existing work on goodness-of-fit, and we want to specially focus on the literature of goodness-of-fit tests for right censored data.

One of the main references that summarizes the major literature to 1986 on goodness-of-fit-tests, is the book of D'Agostino and Stephens [14]. This book is devoted to identify the major techniques behind the goodness-of-fit methods. As it is explained in this book, when we try to classify these techniques in different groups we can use two different criteria according to:

1. Tests for specific distributions,
2. Tests according to the used techniques.

When we talk about specific distributions, we usually think in the Normal, the Uniform, the Exponential or the Weibull distributions among many others. But, we depart from a situation where the data we want to analyze can follow any distribution. So, we have chosen the second criteria, since our goal is more general, there are many distributions that can fit the data and we look for tests that let us check any of these distributions.

The classification among the different possible techniques can be summarized in five main groups of techniques:

1. Tests based on graphical analysis,
2. Chi-squared type tests,
3. Tests based on the empirical distribution function (EDF) statistics,
4. Tests based on regression and correlation.
5. Other tests based on different techniques.

In what follows, we review these five types of techniques paying special attention to the ones that have been adapted to include right-censored data.

1.3.1 Tests based on graphical analysis

Graphical techniques are simple tools that can be implemented easily and can be used as an exploratory technique. This type of tools are less formal than the numeric ones, but they aid us to understand the distribution of the data.

One of the main advantages of the graphical techniques used to asses goodness-of-fit problems is that they extend naturally to censored samples.

There are different plots that are used to assess goodness-of-fit. In the work of Wilk and Gnanadesikan (1968) [56], we find the first reference on probability plots (P-P plots), where the distribution function of one distribution and the distribution function of a second distribution are plotted or quantile-quantile probability plots (Q-Q plots), where the quantiles of two different distributions are plotted.

In 1983, Michael [34] introduces the stabilized probability plot, a variation of the P-P plot where the variances of the plotted points are approximately equal. He also produce a new statistic analogous to the Kolmogorov- Smirnov (which we will present

later). Castro-Kuriss (2007) studies and extends the test of Michael to type I and II censored data, in her master thesis, and she also gives an extension of the stabilized plot [7].

These three graphics were presented for non-censored data but they can be applied to right-censored data. In fact, the main disadvantage of these plots, when the data have censored observations, is the plotted points are not evenly spread. To solve this problem, Waller and Turnbull (1992) propose the empirically rescaled plot [53].

Also, we have another graphic, the cumulative hazard plot. This plot is based on transforming the cumulative hazard function Λ in order to obtain a linear or a logarithmic function. One of the first references on this test based on graphical analysis for censored data is the paper of Nelson (1972) [41], where a hazard plotting method for the analysis of multiple right censored life data is presented.

Nair (1981) [40] considers two different non-parametric procedures for random censored data. One of these procedures is inverted to obtain confidence bands for the survival and cumulative hazard functions. From these bands he provides other bands associated with the percentage, quantile and hazard plots.

In all of these graphics we want to test the following null hypothesis

$$H_0 : F(\cdot) = F_0(\cdot|\theta), \tag{1.1}$$

where F is the unknown distribution of the event times of our data and $F_0(\cdot, \theta)$ is the theoretical distribution we want to fit. We summarize in Table 1.2 the plots presented above.

Type	Plot
P-P plot	$\widehat{F}_0(t)$ vs $\widehat{F}_{KM}(t)$
Q-Q plot	$\widehat{F}_0^{-1}(\widehat{F}_{KM}(t))$ vs t
Stabilized probability plot	$\frac{\pi}{2}\arcsin\left(\sqrt{\widehat{F}_0(t)}\right)$ vs $\frac{\pi}{2}\arcsin\left(\sqrt{\widehat{F}_{KM}(t)}\right)$
Empirically rescaled plot	$\widehat{F}_u(\widehat{F}_0^{-1}(\widehat{F}_{KM}(t)))$ vs $\widehat{F}_u(t)$
Cumulative hazard plot	Depends on the distribution

TABLE 1.2: Description of the plots

We define $\widehat{F}_0(t) := F_0(t, \widehat{\theta})$, $\widehat{F}_{KM}(t)$ is the Kaplan-Meier estimation of F, and finally $\widehat{F}_u$ is the empirical cumulative distribution of the points corresponding to uncensored observations.

These tests are implemented in **R** by Febrer (2015) [17] in her master's thesis.

Finally, to conclude this section we present the Table 1.3 that sum up all the references and the implementation done for graphical methods.

Name	P-P Plot	Q-Q Plot	Stab. P-P Plot	Hazard Plot	Rescaled plot
Reference (Complete)			[34]		
Implement.	[17]	[17]	[17]	[17]	
Type of Censoring	Random		I and II	Random	Random
Problems	Many censoring	Censored data	Many censoring		
Reference (Censored)	[40]	[40]	[7]	[40], [41]	[53]
Implement.	[17]	[17]		[17]	[17]

TABLE 1.3: References: Graphical methods

1.3.2 Chi-squared type tests

Chi-squared type tests, are in general less powerful than other classes of test of goodness-of-fit. Their main advantage is that they are in generally more applicable and can be used either for continuous or discrete data.

The usual form of the χ^2 statistic compares the observed (f_i) versus the expected (e_i) frequencies in each of the cells. The choice of the cells is straightforward when the data is discrete, and more complicated when we have continuous data, but in both cases the statistic can be expressed as

$$Q = \sum_{i=1}^{k} \frac{(f_i - e_i)^2}{e_i},$$

where k is the number of cells.

The most known χ^2 tests for complete data consider the simple null hypothesis

$$\begin{aligned} H_0 &: F = F_0 \\ H_1 &: F \neq F_0, \end{aligned}$$

where F is the unknown distribution of our data and F_0 the distribution we want to check.

Among these tests we have the Pearson χ^2 (1900) [45], the modified χ^2, the log likelihood ratio and the Freeman-Tukey tests. Moreover, Moore and Spruill (1975) [38], give a unified large-sample theory of general chi-squared tests. This class of general χ^2 tests, includes all the previous ones. They also work with composite null hypothesis. That means we want to test if the data follows a distribution belonging to a family of distributions $\mathcal{F}$

$$\begin{aligned} H_0 &: F \in \mathcal{F} \\ H_1 &: F \notin \mathcal{F}. \end{aligned}$$

If the distribution F depends on some parameters θ, then $\mathcal{F}$ is a parametric family of distributions and the hypotheses can be rewritten as:

$$H_0 : F(\cdot|\theta) \in \mathcal{F}_\theta$$
$$H_1 : F(\cdot|\theta) \notin \mathcal{F}_\theta.$$

We have, also, some results for this type of tests for censored data. For example: Mihalko and Moore (1980) [36] extend the applicability of chi-squared tests to data with Type II censoring. Arkritas (1988) [1] proposes chi-squared tests for testing goodness-of-fit when the data may or may not be subject to random censoring. He also considers the simple and compositive null hypotheses. Kim (1993) [27] follows the work of Moore and Spruill and extends the work to randomly right-censored data, he also compares his statistic with the one obtained by Arkritas. Habib and Thomas (1986) [24] give two Pearson-type goodness-of-fit test statistics for parametric families with randomly right-censored data.

Hjort (1990) [25] studies the convergence of the non-parametric Nelson-Aalen plot of the cumulative hazard rate with the estimated parametric cumulative hazard rate to construct a χ^2-type statistic for goodness-of-fit and Crámer-von-Mises and Kolmogorov-Smirnov tests.

For these tests we only found the Pearson chi-squared test implemented in **R**.

We will extend, in Chapter 2, the general chi-squared tests proposed by Moore and Spruill (1975) [38] and Kim (1993) [27], for complete and censored data respectively.

Table 1.4 summarizes the previous references.

Name	**Pearson χ^2**	**Gener. Pearson**	**Akritas**	**Others**
Reference (Complete)	[45]	[38]	[1]	
Implement.	**`chisq.test`** **`gofstat`** (fitdistplus)			
Type of censoring	1- Type I and II 2- Random	Random	Random	Random
Technique	Product-limit estimator	Product-limit estimator	Num. of uncens. obs. in the cell	Product limit MLE nuisance
Reference (Censored)	1- [36] 2- [10]	[27]	[1]	[24]
Implent.				

TABLE 1.4: References: χ^2 Test

1.3.3 Tests based on empirical distribution function statistics

Tests based on the empirical distribution function (EDF), are the most used when the distribution is completely specified. When we think about tests of goodness-of-fit based on the empirical distribution function, the Kolmogorov-Smirnov test (KS) (1933),

based in the supremum distance between our chosen distribution and the empirical one [28], comes to our mind. Related with this statistic, Kuiper (1960) [32] defines another statistic useful for observations on a circle. Also, the Crámer-von Mises statistic (C-vM) (1928) ([13], [37]) and the Anderson-Darling (A-D) (1954) [2], based in the L^2 distances, are well-known tests used to study the goodness-of-fit. A modification of the Crámer-von Mises originally when the probability is distributed on the circumference of a circle is defined by Watson (1961) ([54], [55]).

With the same hypothesis as in (1.1), the statistics of the three main tests presented above are

Name	Statistic
K-S	$\sup_t \lvert F_0(t) - \widehat{F}_n(t)\rvert$
C-vM	$n\int_{-\infty}^{\infty} (F_0(t) - \widehat{F}_n(t))^2 dt$
A-D	$n\int_{-\infty}^{\infty} \frac{(F_0(t) - \widehat{F}_n(t))^2}{F_0(t)(1-F_0(t))} dt$

TABLE 1.5: Statistics of the most important EDF tests.

where $\widehat{F}_n(t)$ is the empirical distribution function of the data and n the data sample size.

All these tests have been adapted to censored data by using the Kaplan-Meier estimate instead of the empirical distribution function in the formulas of the statistics. Barr and Davidson (1973) [4] give a modification of the Kolmogorov-Smirnov test for censored data of Type I and II. The new statistic converges to an asymptotic distribution, given by Koziol and Byar (1975) [30]. Later, Dufour and Maag (1978) [15] give useful formulas so that the asymptotic distributions could be used with finite samples. Fleming, O'Fallon, O'Brien and Harrigton (1980) [18] present a modified version for arbitrarily right-censored data. Also, Pettitt and Stephens (1976) [46], introduce versions of the Crámer-von-Mises, Watson and Anderson-Darling statistics for censored data of Type I and II. Smith and Bain [51] suggest another version of the Crámer-von-Mises statistic for Type II right-censored data from the Uniform distribution. The random censoring case is studied by Koziol and Green (1976) [31], they give the modified statistic and the asymptotic significance points of the statistics for various degrees of censoring. Finally, Koziol (1980) [29], give versions of Kolmogorov-Smirnov, Kuiper and Crámer-von-Mises statistics for random censored data.

In general, the limiting distributions of the statistics when there is random right censoring are unknown. That is because they considerably depend on the censoring degree and distribution of the censoring times. Some recent works have approached the limiting distributions by simulations, although the process of simulation requires time and computational cost. In [11] (2011) Chimitova, Liero and Vedernikova propose an application of the classical Kolmogorov-Smirnov, Cramer-von Mises and Anderson-Darling tests for a complete sample obtained from original censored sample by using randomization. Also Chimitova, Nikulin, Lemeshko and Tsivinskaya (2011) [12]

presents a work in the same direction. Finally, a recent work of Balakrishnan, Chimitova and Vedernikova (2015) [3] the modified Kolmogorov-Smirnov, Cramer-von Mises and Anderson-Darling tests as well as one χ^2 test and the tests based on pseudo-complete samples for Type II right-censored data and using the respective powers as a measure of comparison.

Tests for complete data based on the EDF can be extended to random right-censored data using a transformation so that the transformed censored sample behaves, under the null hypothesis, like a complete sample from the Uniform $(0, 1)$ distribution. Then, any standard goodness-of-fit test can be applied. See for example for Type II censored data Michael and Schucany (1979) [35] or Glen and Foote (2009) [20], where they also give test for uniformity based on the convolution for uniform random variables. There is also the possibility to transform the data to normal variables, for example for Type II censored data, Goldmann et al. (2015) [21].

About the implementation of these test, there exist a package **`fitdistrplus`** [33], which contains a function **`gofstat`**, that computes the Kolmogorov-Smirnov, Crámer von-Mises and Anderson-Darling statistics for complete data.

We summarize the main references of this section in the Table 1.6.

Name	**K-S**	**Kuiper**	**C-vM**	**A-D**	**C-vM mod (Watson)**
Reference (Complete)	[28]	[32]	[13] and [37]	[2]	[54] and [55]
Distance	Supremum	Supremum	L^2	L^2 distance	L^2
Implement.	**`ks.test`** (dgof) **`gofstat`** (fitdistrplus)		**`cvm.test`** (dgof) **`gofstat`**	**`gofstat`**	
Type of censoring	1- I and II 2- Random	Random	1- I and II 2- Random	I and II	I and II
Reference (Censored)	1- [4], [30], [15] 2- [29], [18]	[29], [18]	1- [46] 2- [29], [31]	[46]	[46]
Implent.	[17]				

TABLE 1.6: References: test EDF

1.3.4 Tests based on regression and correlation

The last type of tests of our classification are the tests based on regression and correlation, which rely basically in a graphical method. The ordered observed sample $Y_{(i)}$ is plotted in the vertical axis against a function of the position i, Z_i on the horizontal axis and a straight line is fitted to these points. If the test is based on statistics associated with this line, we will call it a regression test. Whenever the statistic used is the correlation between Y and Z, we will have a correlation test.

In the presence of censored data, the calculation of the correlation coefficient has to be adapted to the types of censored data. See for instance, Salinas et al. (2013) [50] for the Gumbel distribution with right censored data of Type II, Smith and Bain (1976)

[51] for complete and censored data or Chen (1984) [8] for random censored data and completely specified distribution function.

We could calculate the correlation coefficient only for the uncensored data, as is suggested in D'Agostino [14], but then we have to be careful about the bias and specially when we have a large percentage of censored data.

Grané (2012) [23] modifies the statistic introduced by Fortiana and Grané (2003) [19] to construct a goodness-of-fit test for censored sample of Type I and II, when the distribution function is fully specified. She uses an statistic based on Hoeffding's maximum correlation between the empirical distribution function and the hypothesized distribution. Grané and Strzalkowska-Kominiak (2014) [22] extend the result for general right-censored data.

Most of this type of tests assume that the distribution is completely specified, usually they cover the Uniform, the Normal or the Exponential distributions. For instance, Pettitt (1976) [47], gives modified versions of the Crámer-von-Mises, Watson and Anderson-Darling statistics for censored data for testing normality when the parameters μ and σ have to be estimated and Chen (1984) [9] presents a correlation statistic for the composite hypothesis of exponentially when the data is randomly censored.

For these type of tests it is difficult to find any implementation since they are too specific.

1.3.5 Tests based on other techniques

There also other types of tests based in different techniques.

An important set of tests are smooth tests, also known as Neymann's test, Neymann (1937) [42]. The main reference of these tests for complete data can be found in the book of Rayner (2009) [49]. The test is smooth because it is constructed to have a good power against the alternatives whose probability density functions depart slowly from the function stated in the null hypothesis. A more general result is given by Peña (1998) [44]. He presents a class of goodness-of-fit tests that extends Neyman's smooth test, but in a more suitable and natural formulation through hazard functions and it is applicable even when the available data is incomplete because of censoring or truncation.

Turnbull and Weiss (1978) [52] propose an omnibus test for a composite null hypothesis based on the likelihood ratio statistic with grouped data which are subject to random right censoring. Finally, the Kullback-Leibler information for measuring the distance between two distribution functions can be used as well. Some works can be found in Park and Shin (2014) [43] and Rad et al. (2011)[48].

We have found a package **`ddst`**, [5] that implements the smooth tests for complete data, but we do not find any references for censored data.

Table 1.7 summarizes the references presented in this section.

Name	Likelihood ratio	Smooth test	Kullback-Leibler
Reference (Complete)		[42], [49]	
Implement.		`package ddst`	
Type of censoring	Grouped data Random	II Left trunc.	I and II
Reference (Censored)	[52]	[44]	[43], [48]
Implement.			

TABLE 1.7: References: Others

Chapter 2

Generalized Chi-Squared Test

In this Chapter we present the results in Moore and Spruill (1975) [35] and Kim (1993) [27].

2.1 Chi-squared statistics

We start with the original Pearson χ^2 test for goodness of fit for a fixed distribution, [45].

We recall a simple version of the χ^2 test. Assume that $T_1, \ldots, T_n$ are independent, identically distributed (i.i.d) samples from an unknown distribution F. We wish to infer if this sample comes from a distribution F_0. So we can formulate the following hypothesis test:

$$\begin{aligned} H_0 &: \ F = F_0 \\ H_1 &: \ F \neq F_0. \end{aligned}$$

We proceed by dividing the support of the variable into M mutually exclusive classes. The test compares the observed frequencies of the outcomes that fall in these classes with the expected frequencies under the null hypothesis.

We denote by N_j the observed class frequencies, for $j = 1, \ldots, M$, and by $e_j = n \cdot p_j$, the expected class frequencies, where p_j is the probability under H_0 to fall in the class j, $j = 1, \ldots, M$.

Theorem 1 *If $T_1, \ldots, T_n$ are an i.i.d sample. The statistic*

$$Q = \sum_{j=1}^{M} \frac{(N_j - n \cdot p_j)^2}{n \cdot p_j},$$

has as its limiting distribution under H_0 the distribution χ^2_{M-1}.

In the previous Theorem we have considered a simple null hypothesis. Often we want to deal with composite null hypothesis, that means we want to check if our data follow a distribution belonging to a family of distributions. In that case, we can write the hypotheses as

$$\begin{aligned} H_0 &: \ F \in \mathcal{F} \\ H_1 &: \ F \notin \mathcal{F}. \end{aligned}$$

where $\mathcal{F}$ is a certain class of distribution functions.

Also, most of the times, F depends on some parameters θ (often a vector), and $\mathcal{F}$ is a parametric family of distributions denoted by $\mathcal{F}_\theta$. Then, we rewrite the hypotheses

$$H_0 : \ F(\cdot|\theta) \in \mathcal{F}_\theta$$
$$H_1 : \ F(\cdot|\theta) \ \notin \ \mathcal{F}_\theta.$$

In that case, our first problem is how to estimate the parameter θ. Depending on the estimator of θ used, the methods receive different names:

- We can use the maximum likelihood estimator (MLE) based on N_j, the observed class frequencies. In that case the resulting test is known as the Pearson-Fisher χ^2.
- We can use the MLE based on T_i, the original data. Now the resulting test is the Chernoff-Lehmann χ^2.

Both tests are included in the general class of χ^2 statistics which will be presented later.

However, when we need to estimate θ, some problems may appear:

- The statistic does not have a limiting null distribution.
- The limiting null distribution depends on the unknown true value of θ.

To try to solve that problem one of the solutions is to consider random cells. This type of cells will be presented in the next section.

2.1.1 Random cells

The random cells are a generalization of the mutually exclusive class defined in the Pearson χ^2 test. When we think of fixed cells or mutually exclusive classes we make a partition of the range of the variable a priori, before having our data. The meaning of random is just that the cells are functions of the data $T_1, \ldots, T_n$.

Example 1 *We study the time of relapse of a cancer (in months). If we work with fixed cells we can split the time into one year (12 months) intervals such as*

$$(0,\ 12],\ (12,\ 24],\ (24,\ 36],\ (36,\ +\infty).$$

If we consider random cells we are taking into account the data, and our boundaries could be the quantiles. In this case the random cells would be

$$(0,\ P_{25}],\ (P_{25},\ Me],\ (Me,\ P_{75}],\ (P_{75},\ +\infty),$$

where P_{25} is the 25th percentile, Me is the median and P_{75} is the 75th percentile of the data $T_1, \ldots, T_n$.

Our random cells are intervals in $\mathbb{R}$, and the main novelty is that the boundaries of the intervals depend on some summaries of the data, that means they would change from one data set to another.

We set φ the vector of all the statistics of the sample used to compute the boundaries. We set as r the number of statistics needed. So φ is defined in an open set $\Omega_2 \subset \mathbb{R}^r$.

We will denote by $I_j(\varphi) = (a_{j-1}(\varphi),\ a_j(\varphi)]$ the cells, where $j = 1, \ldots, M$ and $-\infty = a_0(\varphi) < a_1(\varphi) < \ldots < a_M(\varphi) = \infty$.

Some of the usual boundaries are:

- For $r = 1$, we take $\varphi = \bar{T}$ (mean of the sample). So $\Omega_2 = \{x : -\infty < x < \infty\} \subset \mathbb{R}$. We define $a_j(\varphi) = \bar{T} + b_j$, $j = 1, \ldots, M-1$, where b_j are some constants with $-\infty < b_1 < \ldots < b_{M-1} < \infty$.
- For $r = 2$, we take $\varphi = (\bar{T}, S_T)$, where S_T is the standard deviation of the sample. In that case, $\Omega_2 = \{(x, y) : -\infty < x < \infty, y > 0\} \subset \mathbb{R}^2$. We can define $a_j(\varphi) = \bar{T} + b_j \cdot S_Y$, $j = 1, \ldots, M-1$, where b_j are some constants as in the previous case.
- For $r = M - 1$, we can take $\varphi = (q_1, \ldots, q_{M-1})$, q_j, $j = 1, \ldots, M-1$, are the quantiles of the sample. Now, $\Omega_2 = \{(x_1, \ldots, x_{M-1}), -\infty < x_1 < \ldots, < x_{M-1} < \infty\} \subset \mathbb{R}^{M-1}$. We define $a_j(\varphi) = q_j$, $j = 1, \ldots, M-1$.

2.1.2 Hypothesis

The next aim is to state the hypothesis of the test we are presenting.

As we have said before, we observe a sample $T_1, T_2, \ldots$ of independent $\mathbb{R}^k$ random variables with distribution function $F(x|\theta, \eta)$.

We have divided the parameters in two groups:

- $\theta \in \Omega_1 \subset \mathbb{R}^m$, where Ω_1 is an open set, are the own parameters of the distribution we want to check.
- η take values over a neighborhood of $\eta_0 \in \mathbb{R}^p$ (often η_0 will be 0). This second set of parameters can be interpreted as the difference or distance between the distribution we want to check and some nuisance we are adding.

We can see the role of the two sets of parameters in the two following examples.

Example 2 *We want to check that our data come from a distribution $F(x|\theta)$, but it might be contaminated by a distribution $H(x)$. We define for $\eta \in [0, 1]$*

$$F(x|\theta, \eta) = (1 - \eta)F(x|\theta) + \eta H(x).$$

The null hypothesis will be that the data come from $F(x|\theta)$, and it will be equivalently written as $H_0 : \eta = 0$. In that case, $\eta = 0$ is equivalent to $T \sim F(x|\theta)$ and we have to check our hypothesis that the data come from a distribution $F(x|\theta)$.

Example 3 *We might assume that our data can come from a Weibull(α, β), but maybe there have been a translation with respect to the origin. In that situation $\theta = (\alpha, \beta)$ and $\eta = a$*

$$F(x|(\alpha, \beta), a) = 1 - \exp(-(\beta(t + a))^\alpha).$$

The null hypothesis in this case is that the data come from a Weibull with a translation a, we can write it as $H_0 : \eta = a$. If we do not reject H_0, we can assume that our data follow a Weibull(α, β) displaced a from the origin.

So in the general case, we will define a distribution function $F(x|\theta, \eta)$ and we want to check a fixed value of η. So, we will set our composite null hypothesis as

$$H_0 : \eta = \eta_0,$$

for some value η_0 fixed.

We denote by $F(x|\theta) = F(x|\theta, \eta_0)$, so under H_0, T_i has a distribution function in the family $F(x|\theta)$.

We will explore Pitman's method of sequential tests, where the alternative hypothesis is a family $H_n : \eta = \eta_n$, for $\eta_n = \eta_0 + n^{-\frac{1}{2}}\gamma$, $\gamma \in \mathbb{R}^p$ fixed. We can observe that H_0 holds when $\gamma = 0$. For n sufficiently large, η_n is in the neighborhood of η_0 where $F(x|\theta, \eta)$ is defined.

2.1.3 General class of statistics

Moore and Spruill (1975) [38] and Kim (1993) [27] consider a general class of χ^2 statistic, T_n that we can describe as a quadratic form with the observed minus the expected frequencies.

We recall that a quadratic form can be written as

$$v^T K v,$$

where K is a symmetric matrix and v a vector.

In our case,

$$T_n = V_n^T \cdot K \cdot V_n, \tag{2.1}$$

V_n will be in essence the difference between observed and expected frequencies. In particular, when $K = Id$, we will obtain the usual sum of squares associated to the χ^2 tests.

Since this point we do not differentiate the case of complete data from the case with censored data. Although the methodology is the same, the definitions become a bit different from that point so, we start now two different sections, one for each case.

2.2 Complete data

Main reference: Moore and Spruill [38]
Statistics: General class of χ^2 statistics (quadratic forms in the standardized cell frequencies).
Limitations: When the number of cells grows with the number n of observations at a rate faster than $\mathcal{O}(n^{\frac{1}{2}})$.

Our first objective is to define the vector V_n in the general class of statistics we have presented in (2.1). For that we need to compute the observed and the expected frequencies.

The observed frequency of the cell j is

$$N_{nj}(\varphi) = n \int_{I_j(\varphi)} dF_e, \tag{2.2}$$

that is the number of $T_1, \ldots, T_n$ falling in the cell $I_j(\varphi)$ where $j = 1, \ldots, M$ and F_e is the empirical distribution function. Moreover, the cell probability for this cell for

values $(\theta,\, \eta)$ is

$$p_j(\theta,\, \eta, \varphi) = \int_{I_j(\varphi)} dF(x|\theta,\, \eta). \tag{2.3}$$

In order to compute these probabilities under the null hypothesis since θ is an unknown parameter, we replace θ by an estimator $\theta_n = \theta_n(T_1,\, \ldots,\, T_n)$. The cells also depend on φ, the variable with the statistics depending on the data, so we state $\varphi_n = \varphi_n(T_1,\, \ldots,\, T_n)$.

So, under the null hypothesis the expected probabilities will be

$$p_j(\theta_n,\, \eta_0, \varphi_n) = \int_{I_j(\varphi_n)} dF(x|\theta_n). \tag{2.4}$$

Now, we can define the components of the M-vector $V_n(\theta,\, \eta,\, \varphi) = (\nu_{nj})_{j=1,\ldots,M}$ that compares the standardized difference between the observed and the expected frequencies. The jth component is

$$\nu_{nj}(\theta_n,\, \eta_0,\, \varphi_n) = \frac{N_{nj}(\varphi_n) - n \cdot p_j(\theta_n,\, \eta_0,\, \varphi_n)}{[n \cdot p_j(\theta_n,\, \eta_0,\, \varphi_n)]^{\frac{1}{2}}}. \tag{2.5}$$

We observe that ν_{nj} follows the same pattern of a χ^2-statistic.

So, we can rewrite our class of general χ^2 statistics from (2.1)

$$T_n = V_n'(\theta_n,\, \eta_0,\, \varphi_n) \cdot K(\theta_n,\, \eta_0,\, \varphi_n) \cdot V_n(\theta_n,\, \eta_0,\, \varphi_n), \tag{2.6}$$

where $K(\theta,\, \eta,\, \varphi)$ is a symmetric $M \times M$ matrix for each $(\theta,\, \varphi) \in \Omega_1 \times \Omega_2$, η fixed.

As a particular statistics included in this general form (2.6), when $K(\theta,\, \eta,\, \varphi) = I_M$, we have

- A particular case of T_n is found estimating θ by

$$\bar{\theta}_n = \{\arg\max_{\theta} \sum_{j=1}^{M} N_{nj}(\varphi_n) \log p_j(\theta,\, \eta_0,\, \varphi_n)\}. \tag{2.7}$$

  So, if we have $K(\theta,\, \eta,\, \varphi) = I_M$ and $\theta_n = \bar{\theta}_n$ this statistic is known as the Pearson-Fisher statistic.

- If we estimate θ using the MLE of θ

$$\widehat{\theta}_n = \{\arg\max_{\theta} \sum_{i=1}^{n} \log f(T_i|\theta)\},$$

  where $f(x|\theta)$ is the probability density function of the distribution function $F(x|\theta)$. This statistic is known as the Chernoff-Lehmann statistic.

Before we state the main results for this class of χ^2 statistics we present an easily computing example to clarify the definitions and notations presented above.

Example 4 *We consider the family of shifted geometric distributions with mass function*

$$f(x|p, c) = p^{x-c}(1-p), \qquad x = c,\, c+1,\, c+2,\, \ldots$$

We will denote $\theta = (p, c)$, so for this case $\Omega_1 = \{(p, c) : 0 \leq p \leq 1, -\infty < c < \infty\}$. Ω_1 is a subset of $\mathbb{R}^2$.

1. **Define our hypothesis.** *We want to check if our data comes from a translated geometric distribution or if there is some contamination. So, we consider*

$$F(x|\theta, \eta) = (1 - \eta)[1 - (1 - p)^{x-c}] + \eta H(x),$$

where $H(x)$ is a fixed distribution function and $\eta \in [0, 1]$.

In that case our hypotheses will be

$$H_0 : \eta = 0$$
$$H_1 : \eta \neq 0.$$

The null hypothesis states that the data comes from a translated geometric distribution.

2. **Data.** *We have an observable (complete) sample of i.i.d r.v $T_1, \ldots, T_n$.*

3. **Estimator for θ.** *We need to compute an estimation of θ. The MLE estimators for p and c are given by*

$$\widehat{c}_n = \min_{1 \leq i \leq n} T_i \qquad \widehat{p}_n = \frac{\bar{T} - \min_{1 \leq i \leq n} T_i}{\bar{T} - \min_{1 \leq i \leq n} T_i + 1}.$$

Since we have used the original data $T_1, \ldots, T_n$ to estimate our parameters and we will take $K(\theta, \eta, \varphi) = I_M$, we are working with the Chernoff-Lehmann statistic.

4. **Define the random cells**. *The cells will be $I_j = [a_{j-1}(\varphi), a_j(\varphi)]$, where $a_j(\varphi) = \varphi - \frac{1}{2} + b_j$, for $0 = b_0 < b_1 < \ldots < b_{M-1} < b_M = \infty$ integers and $j = 1, \ldots, M$.*

Since we think that our data comes from a translated geometric one possibility are might take $\varphi_n = \widehat{c}_n$.

5. **Compute T_n.** *The values of $N_{nj}(\varphi_n)$ (observed frequencies) will depend on the data. We can compute $p_j(\theta, \eta, \varphi)$*

$$p_j(\theta, \eta, \varphi) = \int_{\varphi - \frac{1}{2} + b_{j-1}}^{\varphi - \frac{1}{2} + b_j} dF(x|\theta, \eta) = (1 - \eta)(1 - p)^{\varphi - \frac{1}{2} - c}[(1 - p)^{b_{j-1}} - (1 - p)^{b_j}].$$

Then, under H_0

$$p_j = p_j(\widehat{\theta}, 0, \widehat{c}_n) = (1 - \widehat{p}_n)^{-\frac{1}{2}}[(1 - \widehat{p}_n)^{b_{j-1}} - (1 - \widehat{p}_n)^{b_j}].$$

So $\nu_{nj}(\widehat{\theta}_n, 0, \widehat{c}_n) = (np_j)^{-\frac{1}{2}}[N_{nj}(\widehat{c}_n) - np_j]$.

Finally, we can compute $T_n = \|V_n(\widehat{\theta}_n, \widehat{c}_n)\|^2$.

2.2.1 Main results

For the sake of readability the notation and the assumptions necessaries to state the following results will be discussed in the next section. The assumption are stated in Appendix A

We present in this section three main results and we start summarizing the essential ideas of the two theorems we state.

1. In Theorem 2 the limiting distributions of T_n under H_0 and under H_n are presented.
2. The second result, in Theorem 2, proves that the distribution of T_n does not depend on the true value θ_0.
3. Theorem 3 gives the same result as in 1 for three of the most used statistics included in the general class of statistics (2.6). These statistics will be presented later.

One of the main motivations for using random-cell statistics is to obtain statistics whose null distribution does not depend on the unknown parameter θ in location-scale cases.

So, in order to compute the limiting distribution of the statistics T_n and to prove that it does not depend on θ, we use as a main tool the weak convergence of the empiric distribution function on the unit cube E^k of $\mathbb{R}^k$.

The second part of this theorem is quite general, but note that it includes the result that under some assumptions, T_n is unchanged by linear transformations of the observations T_i.

Now, we can state the first result.

Theorem 2 *1. We assume some regularity conditions (A1-A5) with $\eta = \eta_0$ and $\gamma = 0$. We consider the test with hypotheses*

$$H_0 : \eta = \eta_0,$$
$$H_n : \eta = \eta_n,$$

and the statistic

$$T_n = V_n'(\theta_n,\, \eta_0,\, \varphi_n) \cdot K(\theta_n,\, \eta_0,\, \varphi_n) \cdot V_n(\theta_n,\, \eta_0,\, \varphi_n),$$

defined in (2.6).

Then under the null hypothesis, (θ_0, η_0), the statistic T_n has as its limiting distribution the distribution of

$$\sum_{j=1}^{M} \lambda_j \chi_{1j}^2,$$

where λ_j are the characteristic roots of Σ_0 (defined in the next section) and the χ_{1j}^2 are independent χ^2 variables of 1 degree of freedom.

If we add some more regularity conditions (A1-A6), T_n has as its limiting distribution under the alternative hypothesis, (θ_0, η_n), the distribution of

$$\sum_{\lambda_j \neq 0} \lambda_j \chi_{1j}^2(\nu_j^2/\lambda_j) + \sum_{\lambda_j = 0} \nu_j^2,$$

where $\chi_{1j}^2(\nu_j^2/\lambda_j)$ are independent non-central χ^2 variables of 1 degree of freedom and non-centrality parameter ν_j^2/λ_j and ν_j are the components of the M-vector $\nu = P'\mu_0$ where P is an orthogonal matrix such that $P'\Sigma_0 P$ is diagonal.

2. *Moreover, if the regularity assumptions B1-B4 are satisfied, the distribution of the statistic T_n does not depend on the true value θ_0.*

Finally, we can also rewrite these results for some particular χ^2 statistics as: the Pearson-Fisher (T_{1n}), Chernoff-Lehmann (T_{2n}) and Kambhampati's statistics (T_{3n}). We recall here the definitions

$$\begin{aligned} T_{1n} &= \|V_n(\bar{\theta}_n, \varphi_n)\|^2 \\ T_{2n} &= \|V_n(\widehat{\theta}_n, \varphi_n)\|^2 \\ T_{3n} &= V_n(\widehat{\theta}_n, \varphi_n)' \cdot Q(\widehat{\theta}_n, \varphi_n) \cdot V_n(\widehat{\theta}_n, \varphi_n), \end{aligned}$$

where,

$$Q(\widehat{\theta}_n, \varphi_n) = (I_M - B_n J_n^{-1} B_n')^{-1}, \qquad B_n = B(\widehat{\theta}_n, \varphi_n) \qquad J_n = J(\widehat{\theta}_n).$$

Clearly, as $n \to \infty$, $Q(\widehat{\theta}_n, \varphi_n)$ converges to $Q = (I_M - BJ^{-1}B')^{-1}$.

So now, we can obtain the limiting distributions for these particular cases.

Theorem 3 *When the regularity assupmtions C1, C2 and C3 hold, T_{1n} has the limiting distribution*

$$\begin{aligned} \chi^2_{M-m-1} \qquad & \textit{under} \ \ (\theta_0, \eta_0) \\ \chi^2_{M-m-1}(\|\mu_1\|^2) \qquad & \textit{under} \ \ (\theta_0, \eta_n). \end{aligned}$$

When the conditions C1, C2, C4, C5 and C6 hold, T_{2n} has the limiting distribution

$$\begin{aligned} \chi^2_{M-m-1} + \sum_{j=M-m}^{M-1} \lambda_j \chi^2_{1j} \qquad & \textit{under} \ \ (\theta_0, \eta_0) \\ \chi^2_{M-m-1}(\|\mu_2\|^2) + \sum_{j=M-m}^{M-1} \lambda_j \chi^2_{1j}(\nu_j^2/\lambda_j) \qquad & \textit{under} \ \ (\theta_0, \eta_n). \end{aligned}$$

Finally if the assumptions C1, C2, C4, C5 and C6 hold, T_{3n} has the limiting distribution

$$\begin{aligned} \chi^2_{M-1} \qquad & \textit{under} \ \ (\theta_0, \eta_0) \\ \chi^2_{M-1}(\|\mu_3\|^2 + \sum_{j=M-m}^{M-1} \nu_j^2/\lambda_j) \qquad & \textit{under} \ \ (\theta_0, \eta_n). \end{aligned}$$

Where we recall that m is the length of θ, λ_j, $j = M-m, \ldots, M-1$ is defined in the end of the next section, ν_j follows the definition as in Theorem 2 and μ_1, μ_2, μ_3 are the particular cases of μ defined in the next Section.

2.2.2 Discussion of the hypotheses

The block of assumptions (A) in Appendix A will be needed to find the distribution of T_n under the null hypothesis.

Most of the hypotheses in this block are regularity assumptions on p_j, some technical ones on the differences $\theta_n - \theta_0$ and $\varphi_n - \varphi_0$ and some others on the asymptotic behaviour

of the estimator θ_n and the statistic T_n under (θ_0, η_n). Finally, it is required for $F(x|\theta, \eta)$ to be continuous in x or to have mass points fixed for all (θ, η) and for $K(\theta, \eta)$ to be decomposed as a product of a matrix and its transposed.

Also, we need some additional notation to describe the limiting distribution of general χ^2 statistics of the form T_n. We define

$$\begin{aligned}
\mu &= [B_{12} - BA]\gamma \qquad \text{M-vector} \\
\mu_0 &= S'\mu \\
q' &= (p_1^{\frac{1}{2}}, \ldots, p_M^{\frac{1}{2}}) \\
\mathbb{1}_j(y) &\quad \text{indicator function of } I_j(\varphi_0) \\
W(y) &\quad \text{the M-vector with the } j\text{th component of } \frac{[\mathbb{1}_j(y) - p_j]}{p_j^{\frac{1}{2}}} \\
\Sigma &= I_M - qq' + BLB' - B \cdot E[h(Y)W(Y)'] - E[W(Y)h(Y)']B' \text{ MxM matrix} \\
\Sigma_0 &= S'\Sigma S
\end{aligned} \tag{2.8}$$

where A, h and L are as in A5, S as in A4, B is the $M \times m$ matrix that has (i, j)th entry $p_i^{-\frac{1}{2}} \frac{\partial p_i}{\partial \theta_j}$ and B_{12} is a $M \times p$ matrix that has (i, j)th entry $p_i^{-\frac{1}{2}} \frac{\partial p_i}{\partial \eta_j}$.

The block of assumptions (B) are necessary to show that the null distribution does not depend on the unknown parameter θ. Most of them are technical necessary conditions.

The last block of assumptions (C) are almost equivalent to the assumptions (A) for the particular statistics T_{1n}, T_{2n} and T_{3n} defined in the previous section. We can compare the assumptions in block (A) and in block (C). C1 is equivalent to A1, A2, A3 and A6. C3 is equivalent to A5 for the statistic T_{1n}. And in the same way C5 is equivalent to A5 for the statistic T_{2n}. Finally, C6 is necessary for T_{3n} to be well defined.

It only remains, to define $\lambda_{M-m}, \ldots, \lambda_{M-1}$ appearing in Theorem 3 as the m roots of the equation

$$|B'V - (1 - \lambda)J| = 0,$$

which always satisfy $0 \leq \lambda_j < 1$ and satisfy $0 < \lambda_j < 1$ when $J - B'B$ is positive defined.

2.3 Right-censored data

Main reference: Kim [27]
Statistics: General class of χ^2 statistics (quadratic forms in the standardized cell frequencies).
Limitations: The same as in complete data.
Type of censoring: Random right censoring.

In this section we want to extend the results presented in section 2.2 for right censored data. The scheme is practically the same but since the type of data is quite different we need to rewrite some of the notation and the results.

We recall from the Introduction that under a right random censoring model we will assume that the responses $T_1, \ldots, T_n$ are independent non-negative random variables with continuous parametric distribution $F(x|\theta, \eta)$ (we consider as we explain in the

Section 2.1.2, a distribution function with two type of parameters). The censoring variables $C_1, \ldots, C_n$ are also a non-negative random sample, independent from T_i's and with an unknown continuous distribution function $G(y)$. But in fact what we observe is $Y_i = \min(T_i, C_i)$ and $\delta_i = \mathbb{1}_{\{T_i \leq C_i\}}$ for $i = 1, \ldots, n$.

Our main goal is to define a class of general χ^2 statistics equivalent to (2.6) but for random right-censored data. The main problem when we have censored observations is that we can not use F_e (the empirical distribution function) as an estimator of F. Instead we will use $\widehat{F}_{KM}(t) = 1 - \widehat{S}_{KM}(t)$, where $\widehat{S}_{KM}(t)$ is the Kaplan-Meier estimator, Kaplan and Meier (1985) [26], given by

$$\widehat{S}_{KM}(t) = \begin{cases} 1, & \text{if } t < Y_{(1)} \\ \prod_{i:Y_{(i)} \leq t} \left(1 - \frac{d_i}{n_i}\right), & \text{if } t \geq Y_{(1)}. \end{cases}$$

where $Y_{(i)}$ corresponds to the ordered data ($Y_{(1)} \leq \ldots, \leq Y_{(n)}$), n_i the number of individuals that are at risk just before $Y_{(i)}$ and d_i the number of events observed at moment $Y_{(i)}$.

The observed frequencies are defined now by

$$N_{jn}(\varphi) = \int_{I_j(\varphi)} d\widehat{F}_{KM},$$

where N_{jn} follows the same idea shown in (2.2) where F_e has been replaced by $\widehat{F}_{KM}$.

Moreover, the expected cell probabilities and the standardized difference between observed and expected frequencies do not change from (2.4) and (2.5) respectively. We just take into account that to estimate the unknown parameter θ, this will be replaced by an estimator $\theta_n = \theta_n(Y_1, \ldots, Y_n, \delta_1, \ldots, \delta_n)$ and also the cells will depend on φ and this variable depends on the data $\varphi_n = \varphi_n(Y_1, \ldots, Y_n, \delta_1, \ldots, \delta_n)$.

We now define the general chi-square statistic, equivalent to the class defined in (2.6)

$$T_n = V_n'(\theta_m, \varphi_n) \cdot K_n \cdot V_n(\theta_m, \varphi_n), \tag{2.9}$$

where K_n is a non-negative definite, possibly random, symmetric $M \times M$ matrix converging to a fixed non-negative definite matrix K.

2.3.1 Main result

For right-censored data, we present just one result equivalent to the first part of Theorem 2. This theorem states the limiting distribution of T_n defined in (2.9) under H_0 and H_n.

As in the case of complete data, we have the block of assumptions (A), equivalent to the same block for complete data, but adapted to the introduction of right censoring data. We can find the hypotheses (A') in the Appendix A.

Before we state the main result, we need some new definitions. The values μ, μ_0 and Σ_0 are defined as in (2.8), but Σ has a quite different form

$$\Sigma = \Gamma + BLB' - BE[h(Y, \delta)W(Y, \delta)'] - E[W(Y, \delta)h(Y, \delta)']B',$$

where $\Gamma = E[WW']$ is the asymptotic covariance matrix of V_n, W has a equivalent definition for right-censored data to the one in (2.8). h and L are as in $A5'$ and $W(Y, \delta) = W(Y, \delta|\eta_0)$.

This theorem extends Theorem 2 to the censored data case.

Theorem 4 *Under some regularity assumptions (A1'-A5') with $\eta = \eta_0$ and $\gamma = 0$, then under (θ_0, η_0) the limiting distribution of T_n is the distribution of*

$$\sum_{j=1}^{k} \lambda_j \chi^2_{1j},$$

where λ_j are the characteristic roots of Σ_0 and the χ^2_{1j} are independent χ^2 random variables with one degree of freedom.

If Assumptions A1'-A6' hold, T_n has as its limiting distribution under (θ_0, η_n) the distribution of

$$\sum_{\lambda_j \neq 0} \lambda_j \chi^2_{1j}(\nu_j^2/\lambda_j) + \sum_{\lambda_j = 0} \nu_j^2,$$

where $\chi^2_{1j}(\nu_j^2/\lambda_j)$ are independent non-central χ^2 random variables with one degree of freedom and noncentrality parameter ν_j^2/λ_j, and ν_j are the components of the vector $\nu = P'\mu_0$, where P is an orthogonal matrix such that $P'\Sigma_0 P$ is a diagonal matrix with diagonal elements $\lambda_1, \ldots, \lambda_k$.

2.3.2 Akritas statistic and comparison

Another χ^2 type statistic that it is not included in the general class defined by (2.9) is the Akritas statistic defined in Akritas (1988) [1].

Akritas statistic

This statistic is defined for uncensored or censored data, so we consider that the data $T_1, \ldots, T_n$ follow a distribution function $F(x|\eta)$. We remember that $Y_i = \min(T_i, C_i)$ and we denote by $H(x)$ the distribution function of the variables Y_i.

We can estimate $H(x)$ by its empirical distribution function $H_e(x)$. But, we also can deduce, using that T_i and C_i are independent, that

$$1 - H(x) = (1 - F(x))(1 - G(x)),$$

where $G(x)$ is the distribution function of C_i.

We will consider as null hypothesis $H_0 : \eta = \eta_0$, but now it will be a simple hypothesis since $F(x)$ does not depend of any parameter.

Under the null hypothesis, we can estimate the distribution function $G(x)$ as

$$\tilde{G}(x) = 1 - \frac{1 - H_e(x)}{1 - F(x)}.$$

With this notation we now proceed to define the Akritas statistic. This statistic is defined for fixed cells, A_j, $j = 1, \ldots, M$.

- We denote by n_{1j} the number of uncensored observations in each cell

$$n_{1j} = \sum_{k=1}^{n} \mathbb{1}_{\{Y_k \in A_j,\, \delta_j = 1\}}.$$

- The expected probability in each cell will be

$$\pi_{1j}(\eta) = \int_{A_j} (1 - G(x)) dF(x|\eta),$$

  and we estimate $\pi_{1j}(\eta)$ estimating $G(x)$ by $\tilde{G}(x)$ as we have computed before, we will denote by $\tilde{\pi}_{1j}(\eta)$ this estimation.

- We compare the observed and the expected probabilities

$$\omega_{nj}(\eta) = \sqrt{n} \left(\frac{n_{1j}}{n} - \tilde{\pi}_{1j}(\eta) \right),$$

  for $j = 1, \ldots, M$. We denote by $W_n(\eta)$ the M-vector of the differences.

- We define a diagonal matrix $D_{\tilde{\pi}}(\eta)$, with diagonal elements $\tilde{\pi}_{1j}(\eta)$.

Now, the Akritas statistic for a simple hypothesis is

$$Q_A = W_n'(\eta)\, D_{\tilde{\pi}}(\eta)\, W_n'(\eta) = \sum_{j=1}^{M+1} \frac{(n_{1j} - n \cdot \tilde{\pi}_{1j}(\eta))^2}{n \cdot \tilde{\pi}_{1j}(\eta)}. \tag{2.10}$$

Which under the null hypothesis has asymptotically the distribution χ^2_{M+1}.

Comparing Akrita's and the Generalized Pearson tests for right-censored data

Kim [27] dedicates one section of his paper to compare the asymptotic performance of the Akritas statistic (2.10) and the generalized Pearson statistic for censored data (included in the class of the statistic defined in (2.9)).

We recall the definition on the generalized Pearson statistic for fixed cells

$$Q = n \sum_{i=1}^{M} \frac{(d_i - p_i)^2}{r_i q_i^2 q_{i-1}^2},$$

where,

$$\begin{aligned} d_i &= q_{i-1} \widehat{F}(a_i) - q_i \widehat{F}(a_{i-1}), \\ q_i &= 1 - F(a_i), \\ r_i &= \int_{A_i} \frac{dF(x)}{(1 - F(x))^2 (1 - G(x))} \end{aligned}$$

His conclusions of the comparison in the sense of Pitmann's efficiency are:

1. Neither the Pearson statistic nor the Akritas statistic dominates the other.
2. The results of the comparison depends on the number of censored data and the number of cells.

3. If there is no-censored data, Pearson statistic is superior.
4. For heavily censored data, Akritas statistic is superior.
5. He believes that with moderate censoring Pearson statistic is superior. For that case, he computes some numeric example, but the result it is not proved.

Chapter 3

Implementation

In this chapter we present the implementation of the tests explained in the Chapter 2 for complete and right-censored data. We have implemented three different functions: one for complete data and two for right-censored data. For complete data, we compute the statistic and the p-value of the test, for the distributions presented in Table 3.1. For right-censored data we have one function that compute the statistic for the same distributions and another one that computes also the statistic and the p-value just for some of the distributions.

Distributions	Survival Functions	Parameters
Weibull [Wei(α, β)]	$e^{-(\beta t)^{\alpha}}$	$t \geq 0$ and α, $\beta > 0$
Gumbel [Gum(μ, β)]	$e^{-e^{-\frac{t-\mu}{\beta}}}$	$t \in \mathbb{R}$ and $\mu \in \mathbb{R}$, $\beta > 0$
Normal [N(μ, β)]	$\int_t^{\infty} \frac{1}{\beta\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\beta^2}} dx$	$t \in \mathbb{R}$ and $\mu \in \mathbb{R}$, $\beta > 0$
Log-Normal [LN(μ, β)]	$\int_{\frac{\log(t)-\mu}{\beta}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$	$t \geq 0$ and $\mu \in \mathbb{R}$, $\beta > 0$
Logistic [Logis(μ, β)]	$\frac{e^{-\frac{t-\mu}{\beta}}}{1+e^{-\frac{t-\mu}{\beta}}}$	$t \in \mathbb{R}$ and $\mu \in \mathbb{R}$, $\beta > 0$
Log-Logistic [LLogis(α, β)]	$\frac{1}{1+\left(\frac{t}{\beta}\right)^{\alpha}}$	$t \geq 0$ and α, $\beta > 0$
Four-Paramter Beta [Beta(α, γ, a, b)]	$\frac{B(\alpha,\, \gamma) - B_{\frac{t-a}{b-a}}(\alpha,\, \gamma)}{B(\alpha,\, \gamma)}$	$t \in [a,\, b]$ and α, $\gamma > 0$
Exponential Power [ExpPow(α, β)]	$e^{1-e^{(\beta t)^{\alpha}}}$	$t > 0$ and α, $\beta > 0$
Exponentiated Weibull [ExpWei(α, γ, β)]	$1 - \left[1 - e^{-(\beta t)^{\alpha}}\right]^{\gamma}$	$t > 0$ and α, γ, $\beta > 0$

TABLE 3.1: Definition of the survival functions considered. The shape parameters will be denoted as α and γ, the location parameter will be μ and the scale parameter will be β.

3.1 Distributions

In the implementation of the tests we have worked with nine possible distributions.

We have chosen the same distributions as in Anna Febrer's master's thesis (2015) [17] since this master's thesis follows her work. We recall in Table 3.1 the survival functions of the distributions considered.

It has been used a unification of the parameters. The shape parameters will be denoted as α and γ, the location parameter will be μ and the scale parameter β.

The function $B(\cdot, \cdot)$, appearing in the survival function of the four-parameter Beta, is the Beta function and $B_t(\cdot, \cdot)$ the incomplete Beta function

$$\begin{aligned} B(\alpha, \gamma) &= \int_0^1 x^{\alpha-1}(1-x)^{\gamma-1}dx, \\ B_t(\alpha, \gamma) &= \int_0^t x^{\alpha-1}(1-x)^{\gamma-1}dx. \end{aligned}$$

3.2 Generalized Chi-Squared test for complete data - `Genchi` function

The **`Genchi`** function implements the test of Moore and Spruill [35] presented in Chapter 2 Section 2.2. This function can be a applied to a vector of complete data to test if the data comes from a fixed distribution.

3.2.1 How to use the function?

Description

```
GenChi (data, M, r = M-1, step = 1, distr, estim,
beta.limits=c(0,1), parameters = list(shape = NULL,
shape2 = NULL, location = NULL, scale = NULL), K =
diag(M), boot = FALSE)
```

Arguments

The **`Genchi`** function has ten input arguments, which correspond to

`data`	The vector of data studied.
`M`	A number indicating the number of cells that will be considered.
`r`	Boundaries of cells type. This argument can take the values **`1`** (for cells type mean plus a constant), **`2`** (for cells type mean plus a constant times the standard deviation) and **`M-1`** (for cells type quantiles). By default it is set to **`M-1`**.
`step`	A number corresponding to the distance between the constants for cells type **`1`** and **`2`**.

distr A string specifying the distribution to be tested, with possible values **'weibull'** for the Weibull distribution, **'gumbel'** for the Gumbel distribution, **'norm'** for the Normal distribution, **'lnorm'** for the Log-Normal distribution, **'logis'** for the Logistic distribution, **'loglogis'** for the Log-Logistic distribution, **'beta'** for the Beta distribution, **'expweibull'** for the exponentiated Weibull and **'exppower'** for the Exponential Power distribution.

estim A string specifying if the maximum likelihood estimation of the parameters of the distribution is computed with the observed data or the observed class frequencies. The possible values are **'orig'** for the observed data and **'obsfreq'** for the observed class frequencies.

beta.limits A two components vector corresponding to the lower and upper bounds of the Beta distribution. This argument is only required if the beta distribution is considered. By default, it is set to **c(0,1)**.

parameters A list of specifying the parameters of the theoretical distribution. By default, they are set to **NULL** and they will be computed with the maximum likelihood estimation. This argument is only considered if all parameters of the tested distribution are specified.

K A symmetric matrix used to compute the statistic. By default, it will be an identity matrix.

boot A logical value indicating in the case observed data frequencies if the p-value will be also computed using bootstrap. By default, will be **'FALSE'**.

Value

The function returns a list containing two vectors, one with the results of the test and the other with the estimations of the parameters. It also returns the distribution tested and if the estimation has done for the observed data or to the observed class frequencies.

test A vector containing the value of the observed statistic (**Estad**), the exact p-value when **estim='obsfreq'** (**p-value**) and the p-value computed using bootstrap when **estim='orig'** or **boot=T** in the other case (**p-value.boot**).

distr The distribution tested.

estim Type of data used to estimate the parameters.

param The values of the parameters of the tested distribution. If the user has set the parameters manually, these will be parameters returned otherwise will be computed by maximum likelihood estimation.

For example, if we simulated **rnorm(1000,25,4)** and we test this data against a Weibull distribution using the observed class frequencies to estimate the parameters, six cells and the quantiles to define the cells, the output of calling

```
set.seed(240)
x<-rnorm(1000,25,4)

GenChi(x,M=6,r=5,distr='weibull' estim='obsfreq',boot=T)
```

will be

```
$test
   Estad         p-value    p-value.boot
 1.659200e+01 8.574626e-04 7.500000e-03

$distr
[1] "weibull"

$estim
[1] "obsfreq"

$param
    shape     scale
  7.02592  26.386508
```

3.2.2 How does it work?

We present here the details of the implementation of the General Chi-Squared test introduced by Moore and Spruill [35].

Estimation of the distribution parameters

The estimation of the parameters is different for the case in which we use the original data (**'estim=orig'**) or for the case we want to use the observed class frequencies (**estim='obsfreq'**).

In the first case, we use the **fitdist** function from the package **fitdistrplus** [33] and we estimate the parameter by maximum likelihood.

In the second case we define the likelihood function for each distribution as in (2.7), which is in fact the product between the observed frequencies and the logarithm of the difference of the expected probabilities on the boundaries of each cell. Then, we simply maximize this function using the **mle2** function from the package **bbmle** [6]. It is also, a maximum likelihood estimatation but using the observed frequencies.

Computation of the cells

To compute the cells, there are three possibilities depending on the value **r** of the function.

The first option (**r=1**): The boundaries of the cells are of the form $b + mean(data)$, for some constants b. We have considered the sequence of constants b as $-\left(\frac{M}{2}+1\right)\cdot step, , \ldots, \frac{M}{2}\cdot step$, and by default we set **step=1**, so the length of the intervals will be 1. With this construction the mid interval will contain the mean of the data, so the intervals are centered around the mean.

The second option (**r=2**): The boundaries of the cells are of the form $mean(data) + b \cdot sd(data)$, for some constants b. The constants b are set in the same way as before, but now the default length of the interval will be $sd(data)$. Again the intervals are centered around the mean.

The last option (**r=M-1**): The boundaries of the cells are the quantiles. In that case, we obtain the same observed frequencies in all of the intervals.

Computation of the statistic

For computing the statistic, we need first the observed frequencies and the expected probabilities. Both of them are easily computed: the first one using the **cut** function once we have the cells and the second one by computing the difference of probabilities of the boundaries of the cells. For the distributions Exponentiated Weibull and Exponential Power we have had to define the distribution function since they are not implemented in **R**. See the code in Appendix B.

Computation of the p-value

When we want to compute the p-value, although we know the distribution of the statistic, we can only implement the computation of the exact p-value when the estimation is done with the observed class frequencies. In the other cases the distribution is a sum of chi-squared distributions and we can not compute exactly.

For that reason we have used bootstrap to compute the p-values. The function generates 2000 random samples with the distribution of the null hypothesis and the estimated parameters it has computed and for each sample the function computes the value of the statistic. After that the function obtains the probability of the statistic with the data using the empirical distribution of the statistics computes with the random samples. So, for **estim='orig'** we have compute the p-value using bootstrap and for **estim='obsfreq** we can compute the exact p-value or the p-value obtain using bootstrap.

3.2.3 Limitations

While implementing this method in **R** we came across with some limitations.

The first limitation comes when we try to define the bounds of the cells using the mean or the mean and the standard deviation (**r=1, r=2**) and also the number of cells we want to create is big. In some cases, if the data is too concentrated around the mean, the first and the last cells do not have observed values and it also can be that the expected probabilities are zero. In that case, the solution is to take a small **step**. When this value is small enough there is a partition to avoid the problem. In case we find this type of problem, the function print a warning message and computes the statistic by changing the parameter **r** to **M-1**.

The second limitation is related to the data. In some cases for the data it is impossible to fit certain distributions. For example, if the data takes values between -5 and 8, this data can not follow a Beta or a Weibull distribution. Also the function **fitdist** can not estimate the parameters to fit the desired distribution. We have taken into account the cases for the Log-Normal distribution and the Beta distribution, in the case that the data can not fit this distribution the function returns an error message.

3.2.4 Results

We present in this section some tables of results we have obtained simulating data and applying it to the **Genchi** function.

In the first table we just present the type of results we can obtain from this function. We simulate 1000 values of a Normal distribution and we have computed the statistics using 6 cells and the quantiles as bounds of the intervals.

Distrib./**estim**	**'orig'**	pval.boot	**'obsfreq'**	pvalue	pval.boot
Weibull	24.54	5e-04	13.92	0.003	0.024
Gumbel	58.16	0	20.91	1.1e-04	5.e-04
Normal	4.27	0.52	2.69	0.442	0.746
LNormal	12.63	0.026	4.91	0.179	0.439
Logistic	2.98	0.697	2.69	0.441	0.749
LLogistic	4.49	0.485	4.75	0.191	0.448
Beta	NA	NA	NA	NA	NA
ExpWei	4.57	0.488	10.02	0.007	0.074
ExpPower	121.05	0	41.13	6.1e-09	0

TABLE 3.3: Data: **rnorm(1000,25,4)**. Parameters: M=6, r=5. The colored cells indicates the reject H_0 cases.

We have compared the data with the nine distributions considered in the function and we also compute the statistic and the p-value estimating the parameters using the original data and the observed class frequencies.

In Table 3.3 we have colored the cases where we reject the null hypothesis. So, we can only discard that our simulated data follow a Weibull, a Gumbel and Exponential Power distribution.

Now, we try to compare the data distribution and the results of the test for each distribution. So we have simulated data ($n = 1000$) of each of the distributions considered and we have tested also for each distribution. We have obtained two different tables depending of the type of estimation of the parameters. The results are presented in Tables 3.4 and 3.5.

Dist/Data	Wei	Gumb	Nor	LNor	Logis	LLogis	Beta	ExpW	ExpP
Weibull	0.953	0	0.229	0	NA	0	0.525	0.016	0.048
Gumbel	0	0.955	0	0	0	0	0.080	0	0
Normal	0.670	0	0.996	0	0.240	0	0.035	0.922	0
LNormal	0	0.75	0.028	0.674	NA	0	0.003	0.335	0
Logistic	0.122	0	0.370	0	0.932	0	0.001	0.157	0
LLogistic	0	0.264	0.054	0.199	NA	0.644	0.001	0.091	0
Beta	NA	NA	NA	NA	NA	NA	0.916	0.846	0
ExpWei	0.96	0.989	0.587	NA	NA	0.004	0.663	0.938	0.919
ExpPow	0.001	0	0	NA	NA	NA	0	0	0.930

TABLE 3.4: Comparison Data and distribution. Estimation: Original data. Parameters: M=6, r=5. The colored cells indicates the reject H_0 cases.

Dist/Data	Wei	Gumb	Nor	LNor	Logis	LLogis	Beta	ExpW	ExpP
Weibull	0.777	0.002	0.192	NA	NA	0	0.474	0.014	0.208
	0.955	0.011	0.469	NA	NA	0	0.777	0.063	0.479
Gumbel	0	0.824	0	0	0	0	0.037	0	0
	0	0.974	0	0	0	0	0.140	0.001	0
Normal	0.476	0	0.932	0	0.447	0	0.055	0.684	0.002
	0.785	0	0.994	0	0.756	0	0.188	0.929	0.014
LNormal	0	0.640	0.040	0.529	NA	0.205	0.004	0.204	0
	0	0.898	0.141	0.821	NA	0.452	0.021	0.463	0
Logistic	0.15	0	0.385	0	0.678	0	0.011	0.229	0.001
	0.392	0	0.689	0	0.911	0	0.057	0.516	0.003
LLogistic	0	0.334	0.019	0.070	NA	0.370	0.003	0.105	0
	0	0.622	0.083	0.234	NA	0.682	0.015	0.296	0
Beta	NA	NA	NA	NA	NA	NA	0.799	0.603	0
	NA	NA	NA	NA	NA	NA	0.962	0.859	0.001
ExpWei	0.603	0.741	0.002	NA	0	0.273	0.759	0.410	0.697
	0.96	0.988	0.027	NA	0	0.758	0.99	0.880	0.978
ExpPow	0.052	0	0	NA	NA	0	0.005	0	0.733
	0.188	0	0	NA	NA	0	0	0	0.942

TABLE 3.5: Comparison Data and distribution. Estimation: Observed class frequencies. Parameters: M=6, r=5. The first p-value corresponds to the excat one and the second to bootstrap one. The colored cells indicates the reject H_0 cases.

The columns correspond to the distribution of the simulated data and the rows the tested distributions. The colored cells indicated the rejected distributions. In Table 3.5, there are two p-values, the first one corresponds to the exact p-value and the second one to the p-value computed using bootstrap. We can see that in most of the times the decision taking in account the first or the second are the same.

We observe that the results in both tables are very similar, although there are some minimal differences, most of them coincide. The main differences are when the data is Normal, Log-Logistic or Beta, for the other distributions the decision to take would be the same. If we look to the results by columns we can analyze the discarded distributions when the data are the distribution columns simulated. For example the most illustrative case is when the data are Exponential Power, we can only confuse them with a Weibull, an Exponentiated Weibull or the true distribution would be correct. On the other side regarding both tables, the most confusing case is when the data are an Exponentiated Weibull where we can only discard the Weibull, the Gumbel or the Exponential Power distribution.

Finally, the last two Tables 3.6 and 3.7 try to analize the differences between the possible values of the parameters **`M`**, **`r`** and **`step`** (when it is applicable). We have colored the cells where the test fails, that means do not reject when it has to and reject when it has not to.

Parameters/ **estim**	**'orig'**	pval.boot	**'obsfreq'**	pvalue	pval.boot
M=6 r=5	11.81	0.039	9.15	0.027	0.115
M=6 r=1	17.57	0.003	6.25	0.100	0.273
M=6 r=1 step=.5	19.53	0.004	9.11	0.028	0.104
M=6 r=2	13.28	0.022	13.50	0.004	0.025
M=6 r=2 step=.5	13.44	0.019	9.83	0.020	0.073
M=15 r=14	31.71	0.005	31.92	0.001	0.004
M=15 r=1	40.17	0.001	40.19	0	0
M=15 r=1 step=.5	30.21	0.009	27.67	0.006	0.016
M=15 r=2	NA	NA	NA	NA	NA
M=15 r=2 step=.5	40.07	0.014	41.24	0.000	0.012
M=25 r=24	35.22	0.082	35.81	0.032	0.083
M=25 r=1	54.32	0.006	54.85	0.000	0.003
M=25 r=1 step=.5	50.67	0.001	50.13	0.001	0.002
M=25 r=2	NA	NA	NA	NA	NA
M=25 r=2 step=.5	NA	NA	NA	NA	NA
M=25 r=2 step=.3	54.83	0.015	55.272	0.000	0.015

TABLE 3.6: Data: **rnorm(1000,25,4)** vs Weibull distribution. The colored cells indicate when we do not reject H_0

Parameters/ **estim**	**'orig'**	pval.boot	**'obsfreq'**	pvalue	pval.boot
M=6 r=5	2.85	0.727	2.97	0.397	0.701
M=6 r=1	7.57	0.194	5.49	0.139	0.361
M=6 r=1 step=.5	11.93	0.041	10.02	0.018	0.076
M=6 r=2	3.00	0.698	2.71	0.439	0.754
M=6 r=2 step=.5	4.85	0.445	4.80	0.187	0.445
M=15 r=14	11.02	0.692	10.94	0.534	0.718
M=15 r=1	13.73	0.487	13.72	0.319	0.493
M=15 r=1 step=.5	19.79	0.136	19.72	0.073	0.132
M=15 r=2	4.27	0.776	3.99	0.984	0.807
M=15 r=2 step=.5	10.26	0.722	10.23	0.596	0.742
M=25 r=24	12.20	0.980	12.18	0.953	0.976
M=25 r=1	24.58	0.415	24.53	0.320	0.419
M=25 r=1 step=.5	27.54	0.284	27.52	0.192	0.285
M=25 r=2	NA	NA	NA	NA	NA
M=25 r=2 step=.5	10.88	0.768	10.87	0.977	0.758

TABLE 3.7: Data: **rnorm(1000,25,4)** vs Normal distribution. The colored cells indicate when we reject H_0.

We can see that when we use the quantiles as a bounds of the cells there is not any problem of computation, and in the case that **r=1** or **r=2**, we have to adjust the step value to compute the statistic.

With this tables we can see that it is necessary to think carefully the number of cells used, since for values too small we obtain contradictory results. We can conjecture that there is a number of intervals that would be the optimal and it would be the related to the number of observations. For 1000 observations we can guess it will be around 15, since there is none colored cells for this value of **M**.

3.3 Generalized Chi-Squared test for right-censored data - `GenchiCensv1` function

The **`GenchiCensv1`** function implements the test of Kim [27]. This function computes the statistic for a vector of right-censored data to test if the the data follows a fixed distribution.

3.3.1 How to use the function?

Description

```
GenChiCensv1 (x, c, M, distr, estim, beta.limits=c(0,
1), parameters = list(shape = NULL, shape2 = NULL,
location = NULL, scale = NULL))
```

Arguments

The **`GenchiCensv1`** function has seven input arguments, which correspond to

`x`	The vector of data studied.
`c`	The vector indicating the censored observations.
`M`	A number indicating the number of cells that will be considered.
`distr`	A string specifying the distribution to be tested, with possible values **`'weibull'`** for the Weibull distribution, **`'gumbel'`** for the Gumbel distribution, **`'norm'`** for the Normal distribution, **`'lnorm'`** for the Log-Normal distribution, **`'logis'`** for the Logistic distribution, **`'loglogis'`** for the Log-Logistic distribution, **`'beta'`** for the Beta distribution, **`'expweibull'`** for the exponentiated Weibull and **`'exppower'`** for the Exponential Power distribution.
`estim`	A string specifying if the maximum likelihood estimation of the parameters of the distribution is computed with the observed data or the observed class frequencies. The possible values are **`'orig'`** for the observed data and **`'obsfreq'`** for the observed class frequencies.
`beta.limits`	A two components vector corresponding to the lower and upper bounds of the Beta distribution. This argument is only required if the beta distribution is considered. By default, it is set to **`c(0,1)`**.
`parameters`	A list of specifying the parameters of the theoretical distribution. By default, they are set to **`NULL`** and they will be computed with the maximum likelihood estimation. This argument is only considered if all parameters of the tested distribution are specified.

Value

The function returns the value of the statistic, the distribution tested, if the estimation has done for the observed data or to the observed class frequencies and the values of the estimated parameters.

Estad The value of the observed statistic.

distr The distribution tested.

estim Type of data used for estimate the parameters.

param The values of the parameters of the tested distribution. If the user has set the parameters manually, these will be parameters returned otherwise will be the maximum likelihood estimation.

3.3.2 How does it work?

We present here the details of the implementation of the General Chi-Squared test for Right-Censored data introduced by Kim [27].

Estimation of distribution parameters

The estimation of the parameters is different for the case we want to use the original data (**'estim=orig'**) or for the case we prefer the observed class frequencies (**estim='obsfreq'**).

In the first case, we use the **fitdistcens** function from the package **fitdistrplus** [33] and we estimate the parameters by maximum likelihood.

In the second case, we proceed in the same way as for complete data. We already considered the observed frequencies for the non-censored data, so we define the likelihood function as in (2.7). Then, we simply maximize this function using the **mle2** function from the package **bbmle** [6]. It is also a maximum likelihood estimate but using the observed frequencies.

Computation of the cells

Since we work with censored data, it does not make sense to compute the mean or the standard deviation. So, we will set as boundaries of the cells the quantiles.

We compute the **M** quantiles applying the **quantile** function to a **survfit** object that contains the data with the censored information and then we check if all the desired quantiles have been computed. If they are not we change the **M** value and consider only the ones that exists.

Computation of the statistic

For computing the statistic, we need first the observed frequencies and the expected probabilities. Both of them are easily computed. The first one uses the **cut** function once we have the cells and considers only the non-censored data. The second one is obtained by computing the difference of probabilities of the boundaries of the cells. For the distributions Exponentiated Weibull and Exponential Power we have had to define the distribution function since they are not implemented in **R**. See Appendix C.

3.3.3 Limitations

When we treat with censored data, we have some more limitations than in the complete data case.

First, we can notice that we do not have the function parameter **r**. The cells are always computed using the quantiles. Also, as we said before, we can not always compute all the quantiles needed, so in those cases we change the value of **M**, and the number of cells will depend on the number of quantiles that can be computed.

Related with the previous problem, it is not possible to have in the function argument **K**, since the number of cells can change, the dimension of the vectors containing the observed frequencies and the expected probabilities will change too and it will make incompatible the product with a matrix **K**.

The main limitation we have for this functions is that we can not compute the p-value using bootstrap. When we want to compute the p-value we need to generate a random sample assuming the null hypothesis, so in that case than means to generate random samples of Weibull, Gumbel, Normal... with censoring. There is no function in **R** with that implementation and we also have to assume a distribution for the censored-data. We have not considered this possibility in this function. Also, there is not the possibility of computing the exact p-value, since the distribution is for all the statistics the sum of χ^2 distributions.

3.3.4 Results

We present in this section the results obtained applying the function **GenchiCensv1** to two sets of data of the **survival** package.

We have considered the sets of data **aml** and **kidney**. The first ones contains the survival in patients with Acute Myelogenous Leukemia and the second ones, data on the recurrence times to infection, at the point of insertion of the catheter, for kidney patients using portable dialysis equipment. We present some statistics about the data considered and their survival functions.

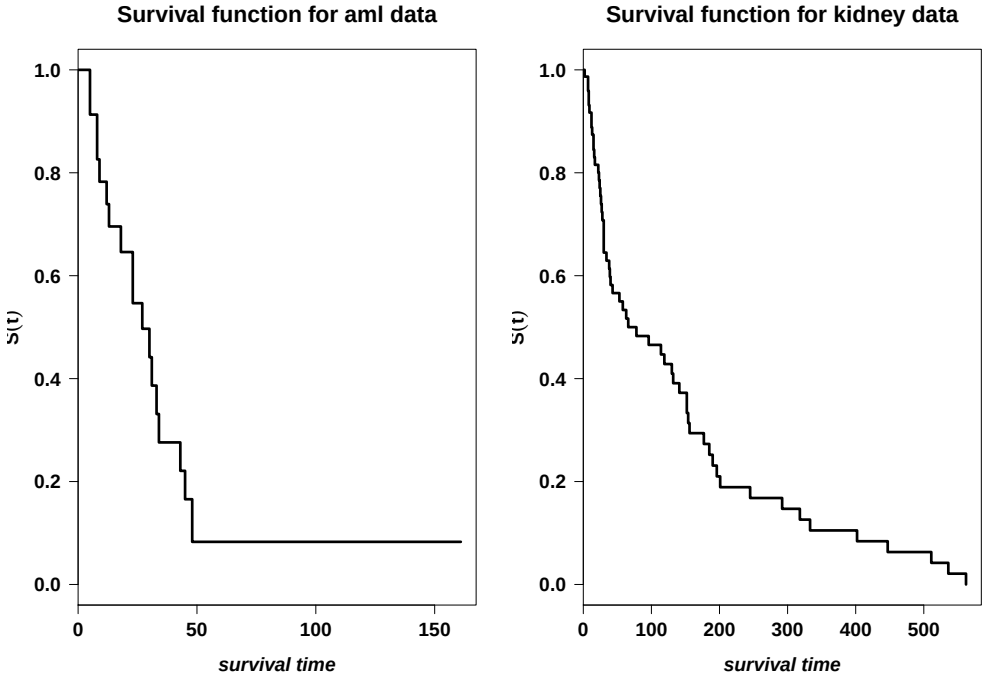

FIGURE 3.1: Survival functions of the **aml** and **kidney** data.

	aml	**kidney**
median	27	78
% of censored data	21.74	23.68

TABLE 3.8: Statistic about the **aml** and **kidney** data.

We have computed the statistics for the original data and for the observed frequencies with number of cells 6 and 12. The results are presented in Table 3.9 and Table 3.10. We have colored the smallest values of each category.

Distr/**estim**	**'orig'** (6)	**'obsfreq'** (6)	**'orig'** (12)	**'obsfreq'** (12)
Weibull	2.039	2.551	7.794	8.433
Gumbel	3.202	2.731	12.964	9.817
Normal	6.049	NA	15.128	NA
LNormal	1.555	3.879	7.210	9.805
Logistic	1.555	2.772	9.035	11.463
LLogistic	1.403	3.580	7.734	10.388
Beta	NA	NA	NA	NA
ExpWei	NA	2.009	NA	NA
ExpPower	NA	NA	NA	NA

TABLE 3.9: Results of the function **GenChiCensv1**. Data: **aml**. The number in parenthesis corresponds to the number of cells considered. The colored cells are the smallest values of each category

We can suspect for the Table 3.9 that **aml** data can follow a Weibull or a Log-Normal distribution since the values of the statistic are lower for these distributions in three of the four cases considered.

Distr/**estim**	**'orig'** (6)	**'obsfreq'** (6)	**'orig'** (12)	**'obsfreq'** (12)
Weibull	8.237	NA	21.836	NA
Gumbel	24.799	NA	61.956	NA
Normal	71.658	NA	112.635	NA
LNormal	7.141	7.985	16.606	18.519
Logistic	58.666	NA	102.621	NA
LLogistic	8.079	8.473	18.566	20.183
Beta	NA	NA	NA	NA
ExpWei	NA	NA	NA	NA
ExpPower	NA	NA	NA	NA

TABLE 3.10: Results of the function **GenChiCensv1**. Data: **kidney**. The number in parenthesis corresponds to the number of cells considered. The colored cells are the smallest values of each category

If we study the second set of data **kidney** we first notice that there are many cases where we can not estimate the parameters of the distribution, probably because the data does not fit the proposed distribution. On the other side all the different cases point in the same direction, that the data may be follow a Log-Normal of a Log-Logistic distribution since they the distributions with the lowest statistics in all the considered cases.

3.4 Generalized Chi-Squared test for right-censored data - `GenchiCensv2` function

The **`GenchiCensv2`** function implements the test of Kim [27]. This function computes the statistic and the p-value for a vector of right-censored data to test if the the time to event and the time to censoring follows fixed distributions. The interest of this function it is if we deal with random censoring.

3.4.1 How to use the function?

Description

```
GenChiCensv2(data, c, M, distr.data, distr.cens,
estim, parameters = list(shape = NULL, shape2 =
NULL, location = NULL, scale = NULL))
```

Arguments

The **`GenchiCensv2`** function has seven input arguments, which correspond to

`data`	The vector of data studied.
`c`	The vector indicating the censored observations.
`M`	A number indicating the number of cells that will be considered.
`distr.data`	A string specifying the time to event distribution to be tested, with possible values **`'weibull'`** for the Weibull distribution, **`'lnorm'`** for the Log-Normal distribution and **`'loglogis'`** for the Log-Logistic distribution.
`distr.cens`	A string indicating the time to censoring distribution to be tested, with possible values **`'weibull'`** for the Weibull distribution, **`'lnorm'`** for the Log-Normal distribution. **`'loglogis'`** for the Log-Logistic distribution and **`'unif'`** for the Uniform distribution.
`estim`	A string specifying if the maximum likelihood estimation of the parameters of the distribution is done with the observed data or the observed class frequencies. The possible values are **`'orig'`** for the observed data and **`'obsfreq'`** for the observed class frequencies.
`parameters`	A list of specifying the parameters of the theoretical distribution. By default, they are set to **`NULL`** and they will be estimated with the maximum likelihood estimate. This argument is only considered if all parameters of the tested distribution are specified.

Value

The function returns a list containing two vectors. One with the results of the test and the other with the estimation of the parameters. It also returns the distributions of the time to event and the time to censoring tested and if the estimation has been done for the observed data or to the observed class frequencies.

test	A vector containing the value of the observed statistic (**Estad**) and the p-value computed using bootstrap (**p-value.boot**).
distr.data	The time to event distribution tested.
distr.cens	The time to censoring distribution tested.
estim	Type of data used to estimate the parameters.
param	The values of the parameters of the tested distribution. If the user has set the parameters manually, these will be parameters returned otherwise will be the maximum likelihood estimation.

3.4.2 How does it work?

We present here the details of the implementation of the General Chi-Squared test for right-censored data introduced by Kim [27].

Estimation of distribution parameters

The estimation of the distribution parameters is done in the same way as in the previous function.

We only have to add, that in the same way we also need to estimate the parameters of the censoring distribution. This is needed in order to compute the p-value using bootstrap. Since we need to generate samples with the distribution in the null hypothesis we also need an estimation of the parameters.

Computation of the cells

In the computation of the cells there is no difference with the previous function. Everything is computed exactly in the same way.

Computation of the statistic

There is also no difference in the computation of the statistic in comparing with the previous function.

Computation of the p-value

This point is the main difference with the previous function. We have used the **survsim** package (2014) [39], to simulate random censored samples. This package allows to generate a simulation for a fixed distribution of the data and a fixed distribution of the censoring times.

The function generates 1000 random censored samples with the distribution of the null hypothesis and a fixed distribution for the censoring times. For each sample the function computes the value of the statistic. After that, the function obtains the probability of the statistic with the data using the empirical distribution of the statistics computed with the random samples.

The idea is the same as with the complete data, but now the functions generates random censored samples.

3.4.3 Limitations

All the limitations considered for the previous function are also applicable to this function and also we have to add some other ones specific to this one.

First of all, we have to notice that there are less distributions considered, but since they also have to be related to the distribution of the censoring times, that give us 12 possible cases to consider.

The main problem we have found is when the combination of the distribution of the data and the distribution of the censoring times gives us a random sample with to many censored data, for that type of sample it is impossible to compute the statistic because we only have one possible cell. For that combinations we can not do the test.

3.4.4 Illustrations

We have used the same data sets as in the previous function to check the possible distributions. We remember that we are working with the data sets **`aml`** and **`kidney`** from the package **`survival`**.

For the next tables, we have set NA^1 when the generate random samples used to compute the p-value have too much censoring times and we can not compute the observed statistic and NA^2 when we are not able to estimate the parameters of the distribution function.

The Tables 3.11 and 3.12, capture the results for the data in **`aml`**. In that case, we have considered 4 cells when the statistic is computed with the original data, because the data set contains 23 observations and 6 cells for the observed frequencies (with less cells there are more cases of NA).

Dist. Data/Dist. Cens	Weibull	LNormal	LLogistic	Uniform
Weibull	1	1	1	1
LNormal	NA^1	0.067	0.129	0.658
LLogistic	NA^1	0.587	0.003	0.541

TABLE 3.11: Data: **`aml`**. Estimation: Original Data. Parameter: M=4. The colored cells we do not reject the null hypothesis. NA^1 indicates too much censoring times, the statistic can not be computed.

Dist. Data/Dist. Cens	Weibull	LNormal	LLogistic	Uniform
Weibull	1	1	1	NA^2
LNormal	NA^1	0.051	0.345	NA^2
LLogistic	NA^1	0.515	0.157	NA^2

TABLE 3.12: Data: **`aml`**. Etimation: Observed class frequencies. Parameter: M=6. The colored cells we do not reject the null hypothesis. NA^1 indicates too much censoring times, the statistic can not be computed and NA^2 the parameters of the distribution function can not be estimate.

If we try to analyze the two previous tables, it is difficult to take a decision about the distribution of the data and the distribution of the censoring times. It seem plausible that the data follows a Weibull distribution or maybe a Log-Normal distribution. This results coincides with the results in Table 3.9

In fact, if we plot together the survival function of the data with a Weibull and a Log-Normal distribution it see that the two distributions can fit the data.

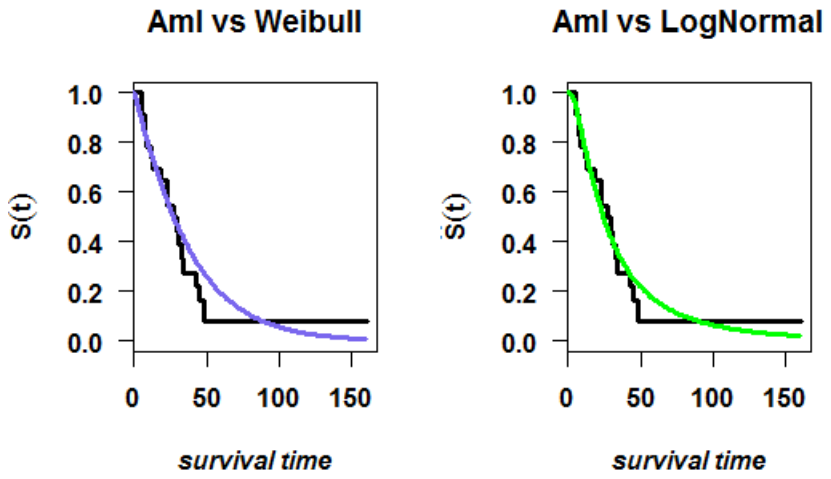

FIGURE 3.2: Survival function of the **aml** versus a Wei(1.097, 38.197) and versus a Log-Norm(3.194, 0.929).

We have estimate the parameters of the Weibull and the Log-Normal distribution by MLE using the original data.

We check now, the second set of data and we summarize the results in the Tables 3.13 and 3.14.

Dist. Data/Dist. Cens	Weibull	LNormal	LLogistic	Uniform
Weibull	1	1	1	1
LNormal	NA[1]	0.003	0.181	0.645
LLogistic	NA[1]	0.173	0.006	0.538

TABLE 3.13: Data: **kidney**. Estimation: Original Data. Parameter: M=6. The colored cells we do not reject the null hypothesis. NA[1] indicates too much censoring times, the statistic can not be computed.

Dist. Data/Dist. Cens	Weibull	LNormal	LLogistic	Uniform
Weibull	NA[2]	NA[2]	NA[2]	NA[2]
LNormal	NA[1]	0.005	0.000	NA[2]
LLogistic	NA[1]	0.013	0.004	NA[2]

TABLE 3.14: Data: **kidney**. Estimation: Observed class frequencies. Parameter: M=6. The colored cells we do not reject the null hypothesis. NA[1] indicates too much censoring times, the statistic can not be computed and NA[2] the parameters of the distribution function can not be estimate.

In that case, the Table 3.13 is very similar to the Table 3.11, so the conclusions would be similar, but Table 3.14 will contradict this conclusions as well as the results from the previous function Table 3.10.

To have a graphic point of view, we also have plotted the survival function together with the Log-Normal and the Log-Logistic distributions, using as the parameters the estimation by MLE with the original data.

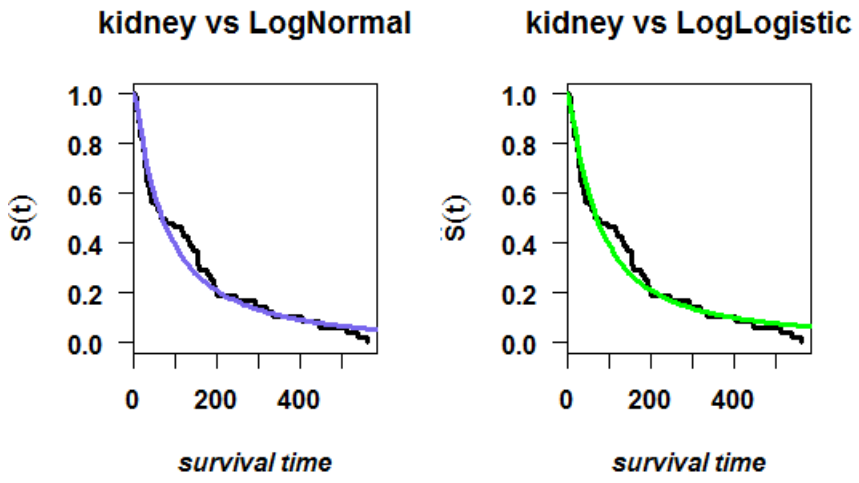

FIGURE 3.3: Survival function of the **`kidney`** versus a Log-Norm(4.224, 1.330) and versus a Log-Logistic(1.257, 70.032).

We can see that both distributions seems to fit properly the survival function, so the results in Table 3.14, makes us suspect that we have to change or the number of cells considered or the type of estimation of the parameters.

Chapter 4

Conclusions and further work

4.1 Conclusions

The work we have presented can be divided into three different parts: the bibliographically research about goodness-of-fit test for right-censored data, the theoretic explanation of the Generalized Chi-Squared test of Moore and Spruill (1975) [35] for complete data and Kim (1993)[27] for right-censored data, and finally the implementation of the results for both complete and right-censored data.

After an exhaustive research on goodness-of-fit tests for right-censored data, we have found that although there are many results on the topic they are particular cases (for fixed distributions or just for one type of data censoring) and disperse (there are some extensions of known tests for complete data to right-censored data and some new techniques just for censored data). There is not a clear reference or references for this type of tests. Moreover, if we are looking for an implementation of any of these methods in **R**, to the best of our knowledge, we have not found any package covering all.

Now, we focus on the Generalized Chi-Squared test for complete and censored data presented in Chapter 2 and their implementation in Chapter 3. These tests consider a composite null hypothesis, so the distribution under the null hypothesis is a parametric family of distributions. The results presented in references [35] and [27] are strong in the sense that they provide an easy computation of the statistic and the asymptotic distribution under the null hypothesis and this does not depend on the possible true value of the parameters of the distribution on the null hypothesis. The asymptotic distribution depends only on the number of cells considered and the number of parameters of the null distribution. The results presented are so general that included almost all the different known Chi-squared tests.

The next step it is to implement these methods, but the fact of having a lot of freedom to choose the cells becomes to be the main disadvantage for implementation. First of all, we have to choose the number of cells and then the method to compute the boundaries of the cells. The number of cells chosen by the user seems that it has to be related to the number of data. On the other hand, for the second choice there is a difference if we are dealing with the complete data implementation or the right-censored data. For complete data we have considered three different type of cells. One of them uses the quantiles, and seems to us the most robust method to choose the boundaries while the other two types of cells depend on the mean and on the mean and the standard deviation. These type of cells are centered on the mean so are not appropriate for non centered data. For right-censored data, we can only compute the cells boundaries using the quantiles, since the mean and the standard deviation can not be computed.

The computation of the p-value is another important limitation in the implementation. Although the distribution is known, this distribution is the sum of chi-squared distributions, and it becomes very difficult to compute the p-value exactly, we have used bootstrap to compute it. The p-values computed using bootstrap and the exact p-value computed in the case where it is possible are not the same, but most of the times they will bring us to take the same decision, there are only a few cases for p-values near 0.05 that give us contradictory results.

The results obtained for complete data, always bring us to not reject the true distribution, but also for the distributions we have checked, there is a group of distributions we do not reject when we would have to. So, there are some confusing distributions but it seems we always can be sure we do not discard the true distribution.

We have two different types of the results for right-censored data, we can see checking the graphics that the lowest values of the statistic gives us a good fit for the survival function. For the second sets of results obtained computing the p-value they are not conclusive, so it would be necessary to review the number of cells and the type of estimations to obtain more accurate results.

4.2 Further work

After working in the topic of goodness-of-fit for right-censored data one question comes to our mind: Is there a goodness-of-test that is the most preferable when we deal with right-censored data?

If someone asks us the same question for complete data, probably we will answer to use the **R** function `gofstat` of the package `fitdistrplus` [33]. In this function, if the data is continuous, a χ^2 test, the Kolmogorov-Smirnov test, the Crámer-von Mises test and the Anderson-Darling test are computed, so with that results you can be pretty confident about the decision you take. So, we think it will be useful to have a similar function for right-censored data. One function, that computes the three or four most important tests of goodness-of-fit and that can be an important point in order to make your conclusions about the data.

About the Generalized Chi-Squared test implemented in Chapter 3, there are in our opinion, too many decisions to be made for the researcher. So, it would be a nice work to establish a criteria or a set of recommendations about some points needed to apply the test. The main points would be:

- The number of cells. It is clear that the number of cells has to be related to the number of observations, or in right censored data, the number of non-censored data and the percentage of censoring data.
- The computation of the boundaries. Since the boundaries depends on the data, it seems important to study a little bit the structure of the data before making the choice of the bounds. Maybe the median could be used instead of the mean.
- The minimum number of data necessary to obtain significant results. The number of data will also take into account the percentage of censored data if it is the case.
- The best choice of the type of estimation of the parameters. Probably this choice will be related to the number of parameters or the form of the null distribution function.

There is further work to be done researching what can we state for small sets of data and what about results of goodness-of-fit for interval-censored data.

When we have data with too few observations and we can not apply any of the known test, is there any way to assess the goodness-of-fit?

Finally, we can open another topic, related to this, about goodness-of-fit results for interval censored data. Which type of results exists for interval censored data? Is it possible to extend the results of right-censored data to interval censoring?

The goodness-of-fit test for censored data it is still an open problem . There is some work to do, especially trying to summarize and to implement the existing results.

Appendix A

Hypothesis on χ^2 tests

In this Appendix we will provide the necessary assumptions to prove the results for the χ^2 tests presented in the Chapter 2.

A.1 Complete Data

The first block of assumptions is used for finding the distribution of T_n under the null hypothesis.

Assumptions (A):

A1. Under (θ_0, η_n), $\theta_n - \theta_0 = \mathcal{O}_p(n^{-\frac{1}{2}})$ and $\varphi_n - \varphi_0 = \mathcal{O}_p(1)$ for some $\theta_0 \in \Omega_1,\ \varphi_0 \in \Omega_2$. Every vertex $x(\varphi)$ of every cell $I_j(\varphi)$ is a continuous $\mathbb{R}^k$-valued function of φ in a neighborhood of φ_0.

A2. *Regularity assumption.* For each j, $p_j(\theta, \eta, \varphi)$ is continuous in (θ, η, φ) and continuously differentiable in (θ, η) in a neighborhood of $(\theta_0, \eta_0, \varphi_0)$. Moreover, $\sum_{j=1}^{M} p_j = 1$ and $p_j > 0$ for each j.

A3. $F(x) = F(x|\theta_0, \eta_0)$ is continuous at every vertex $x(\varphi_0)$ of every cell $I_j(\varphi_0)$. As $n \to \infty$, $\sup_x |F(x|\eta_n) - F(x)| \to 0$.

A4. $K(\theta, \varphi) = S(\theta, \varphi)S(\theta, \varphi)'$ for an $M \times M$ matrix $S(\theta, \varphi)$ with entries continuous in (θ, φ) at (θ_0, φ_0).

A5. Under (θ_0, η_n)

$$n^{\frac{1}{2}}(\theta_n - \theta_0) = n^{-\frac{1}{2}} \sum_{i=1}^{n} h(T_i, \eta_n) + A\gamma + \mathcal{O}_p(1)$$

for some $m \times p$ matrix A and measurable function $h(x, \eta)$ from $\mathbb{R}^k \times \mathbb{R}^p$ to $\mathbb{R}^m$ satisfying

$$\begin{aligned} E[h(y, \eta_n)|(\theta_0, \eta_n)] &= 0 \\ E[h(y, \eta_n)h(y, \eta_n)'|(\theta_0, \eta_n)] &= L(\eta_n) \end{aligned}$$

where $L(\eta_n)$ is a nnd $m \times m$ matrix converging to the finite nnd matrix $L = E[h(y)h(y)']$ as $n \to \infty$.

A6. The distribution function $F(x|\eta)$ possess a density function $f(x|\eta)$ with respect to a σ-finite dominating measure ν. As $n \to \infty$, $f(x|\eta_n) \to f(x|\eta_0)$ and $h(y, \eta_n) \to h(y)$ a.s (ν).

The second block of assumptions (B) are required to prove that the distribution of T_n under the null hypothesis do not depend on the parameter θ.

Assumptions (B):

B1. For $-\infty < \theta^{2j-1} < \infty$ and $\theta^{2j} > 0$, $j = 1, \ldots, k$,

$$F(y^1, \ldots, y^k|\theta) = F\left(\frac{y^1 - \theta^1}{\theta^2}, \ldots, \frac{y^k - \theta^{2k-1}}{\theta^{2k}}\right)$$

B2. If $Z = (Z^1, \ldots, Z^k)'$, where $Z^j = \alpha_j T_j + \beta_j$ for any $-\infty < \alpha_j < \infty$ and $\beta_j > 0$, $j = 1, \ldots, k$ then θ_n satisfies

$$\begin{aligned} \theta_n^{2j-1}(Z_1, \ldots, Z_n) &= \alpha_j \theta_n^{2j-1}(T_1, \ldots, T_n) + \beta_j \\ \theta_n^{2j}(Z_1, \ldots, Z_n) &= \alpha_j \theta_n^{2j}(T_1, \ldots, T_n) + \beta_j, \end{aligned}$$

B3. $r = m = 2k$ and each vertex $x(\varphi)$ and φ_n satisfy for $j = 1, \ldots, k$

$$x^j(\varphi_n(Z_1, \ldots, Z_n)) = \alpha_j x^j(\varphi_n(Z_1, \ldots, Z_n)) + \beta_j$$

B4. $K(\theta_n(Z_1, \ldots, Z_n), \varphi_n(Z_1, \ldots, Z_n)) = K(\theta_n(T_1, \ldots, T_n), \varphi_n(T_1, \ldots, T_n))$.

The third block of assumptions, are the particular conditions needed to prove the previous results for the Pearson-Fisher, Chernoff-Lehmann and Kambhampati's statistics.

Assumptions (C):

C1. A1, A2, A3 and A6 hold.

C2. $m \leq M$ and the matrix with entries $\frac{\partial p_i}{\partial \theta_j}$ has rank m.

C3. $n^{\frac{1}{2}}(\bar{\theta}_n - \theta_0) = (B'B)^{-1}B'V_n(\eta_n) + (B'B)^{-1}B'B_{12}\gamma + \mathcal{O}_p(1)$, holds.
This implies that $\bar{\theta}_n$ satisfies A5.

C4. $\log f(x|\theta, \eta)$ is differentiable with respect to (θ, η) at (θ_0, η_0). The matrix J is positive definite and J_{12} is finite.

C5. $n^{\frac{1}{2}}(\widehat{\theta}_n - \theta_0) = n^{-\frac{1}{2}} \sum_{i=1}^{n} J^{-1} \frac{\partial \log f(T_i|\eta_n)}{\partial \theta} + J^{-1}J_{12}\gamma + \mathcal{O}_p(1)$ holds.
So that implies $\widehat{\theta}_n$ satisfies A5.

C6. $J - B'B$ is positive definite.

Where J is the information matrix for $F(x|\theta)$ at θ_0

$$J = E\left[\left(\frac{\partial \log f}{\partial \theta}\right)\left(\frac{\partial \log f}{\partial \theta}\right)'\right],$$

and J_{12} is the $m \times p$ matrix

$$J = E\left[\left(\frac{\partial \log f}{\partial \theta}\right)\left(\frac{\partial \log f}{\partial \eta}\right)'\right].$$

A.2 Right censored data

This block of assumptions is equivalent to the block of assumptions (A) for complete data

Assumptions (A′):

A1′. Under (θ_0, η_n), $\theta_n - \theta_0 = \mathcal{O}_p(n^{-\frac{1}{2}})$ and $\varphi_n - \varphi_0 = \mathcal{O}_p(1)$ for some $\theta_0 \in \Omega_1,\ \varphi_0 \in \Omega_2$. The cell boundaries $a_i(\varphi)$ are real valued continuous functions of φ in a neighborhood of φ_0.

A2′. *Regularity assumption.* For each j, $p_j(\theta, \eta, \varphi)$ is continuous in (θ, η, φ) and continuously differentiable in (θ, η) in a neighborhood of $(\theta_0, \eta_0, \varphi_0)$. Moreover, $\sum_{j=1}^{M} p_j = 1$ and $p_j > 0$ for each j.

A3′. $F(x) = F(x|\theta_0, \eta_0)$ is continuous at every vertex $x(\varphi_0)$ of every cell $I_\sigma(\varphi_0)$. As $n \to \infty$, $\sup_x |F(x|\eta_n) - F(x)| \to 0$.

A4′. K_n is a non-negative definite, possibly random $M \times M$ matrix which converges to a fixed non-negative definite $M \times M$ matrix K as $n \to \infty$.

A5′. Under (θ_0, η_n)

$$n^{\frac{1}{2}}(\theta_n - \theta_0) = n^{-\frac{1}{2}} \sum_{i=1}^{n} h(T_i, \delta_i, \eta_n) + A\gamma + \mathcal{O}_p(1)$$

for some $m \times p$ matrix A and measurable function $h(x, \delta, \eta)$ from $\mathbb{R}^k \times \{0, 1\} \times \mathbb{R}^p$ to $\mathbb{R}^m$ satisfying

$$\begin{aligned} E[h(y, \delta, \eta_n)|(\theta_0, \eta_n)] &= 0 \\ E[h(y, \delta, \eta_n)h(y, \delta, \eta_n)'|(\theta_0, \eta_n)] &= L(\eta_n) \end{aligned}$$

where $L(\eta_n)$ is a nnd $m \times m$ matrix converging to the finite non-negative matrix $L = E[h(y, \delta)h(y, \delta)']$ as $n \to \infty$.

A6′. The distribution function $F(x|\eta)$ and $G(x)$ possess a density function $f(x|\eta)$ and $g(x)$ with respect to a σ-finite dominating measure ν. As $n \to \infty$, $f(x|\eta_n) \to f(x|\eta_0)$ and $h(y, \delta, \eta_n) \to h(y, \delta)$ a.s (ν).

Appendix B

GenChi code

```
GenChi <- function(data, M, r=M-1, step=1, distr, estim,
              beta.limits=c(0,1), parameters=list(shape=NULL,
              shape2 = NULL, location = NULL, scale = NULL),
              K=diag(M), boot=FALSE){

  if((data<0||data>1)& distr=='beta')
    stop('The values are not between 0 and 1, the beta
          distribution cannot fit the data')
  if(sum(data<0)>0 & distr=='lnorm')
    stop('There are negative values, the log-normal distribution
       cannot fit the data')

  require(FAdist)  # per la Gumbel
  require(eha)  # per la loglogis

  n <- length(data)
  a.beta <- beta.limits[1]
  b.beta <- beta.limits[2]

  est<-boot.fun(data,M,r,step,distr,estim,beta.limits,parameters)
  tn<-est$tn
  m<-est$m
  parameters<-est$param
  alpha <- parameters$shape
  gamma <- parameters$shape2
  mu <- parameters$location
  beta <- parameters$scale

  # Computation of the p-value
  pvalue <- NULL
  pvalue.boot <- NULL
  if (estim == 'obsfreq') {
    pvalue <- 1-pchisq(tn, M-m-1)
  }
  if (estim == 'orig'|| boot == T) {
    t<-numeric(2000)
    if(distr=="weibull"){
      for(i in 2:2000){
```

```
    rand<-rweibull(n, alpha,beta)
    t[i]<-boot.fun(rand, M=M, r=r, step=step, distr='weibull',
     parameters = list(shape=alpha, shape2 = NULL,location=
     NULL, scale = beta), K=diag(nrow = M, ncol = M))$tn
  }
}
if(distr=='gumbel'){
  for(i in 1:2000){
    rand<-rgumbel(n, beta, mu)
    t[i]<-boot.fun(rand, M=M, r=r, step=step, distr='gumbel',
     parameters = list(shape = NULL, shape2 = NULL,location=
     mu, scale = beta), K=diag(nrow = M, ncol = M))$tn
  }
  rm(dgumb,pos = ".GlobalEnv")
  rm(pgumb,pos = ".GlobalEnv")
}
if(distr=='norm'){
  for(i in 1:2000){
    rand<-rnorm(n, mu, beta )
    t[i]<-boot.fun(rand, M=M, r=r, step=step, distr='norm',
     parameters = list(shape = NULL, shape2 = NULL,location=
     mu, scale = beta), K=diag(nrow = M, ncol = M))$tn
  }
}
if(distr=='lnorm'){
  for(i in 1:2000){
    rand<-rlnorm(n, mu, beta )
    t[i]<-boot.fun(rand, M=M, r=r, step=step, distr='lnorm',
     parameters = list(shape = NULL, shape2 = NULL,location=
     mu, scale = beta), K=diag(nrow = M, ncol = M))$tn
  }
}
if(distr=='logis'){
  for(i in 1:2000){
    rand<-rlogis(n, mu,beta )
    t[i]<-boot.fun(rand, M=M, r=r, step=step, distr='logis',
     parameters = list(shape = NULL, shape2 = NULL,location=
     mu, scale = beta), K=diag(nrow = M, ncol = M))$tn
  }
}
if(distr=='loglogis'){
  for(i in 1:2000){
    rand<-rllogis(n, alpha, beta)
    t[i]<-boot.fun(rand, M=M, r=r, step=step, distr='loglogis',
     parameters = list(shape = alpha, shape2 = NULL,location=
     NULL, scale = beta), K=diag(nrow = M, ncol = M))$tn
  }
  rm(ploglogis,pos = ".GlobalEnv")
  rm(dloglogis,pos = ".GlobalEnv")
}
```

```
    if(distr=='beta'){
      for(i in 1:2000){
        rand<-(b.beta-a.beta)*rbeta(n, alpha,gamma)+a.beta
        t[i]<-boot.fun(rand, M=M, r=r, step=step, distr='beta',
         parameters=list(shape=alpha,shape2=gamma,location
         =NULL, scale = NULL), K=diag(nrow = M, ncol = M))$tn
      }
    }
    if(distr=='expweibull'){
      for(i in 1:2000){
        rand<-1/beta*(-log(1-runif(n)^(1/gamma)))^(1/alpha)
        t[i]<-boot.fun(rand,M=M,r=r,step=step,distr='expweibull',
         parameters=list(shape=alpha,shape2=gamma,location=
         NULL, scale = beta), K=diag(nrow = M, ncol = M))$tn
      }
      rm(pexpwei,pos = ".GlobalEnv")
      rm(dexpwei,pos = ".GlobalEnv")
    }
    if(distr=='exppower'){
      for(i in 1:2000){
        rand<-1/beta*(log(1-log(1-runif(n))))^(1/alpha)
        t[i]<-boot.fun(rand, M=M, r=r, step=step, distr='exppower',
         parameters=list(shape=alpha,shape2=NULL,location=
         NULL, scale = beta), K=diag(nrow = M, ncol = M))$tn
      }
      rm(pexppow,pos = ".GlobalEnv")
      rm(dexppow,pos = ".GlobalEnv")
    }
    pvalue.boot <- 1 - ecdf(t)(tn)
  }

# Results
  output <- list(test = c('Estad'= tn, 'p-value'= pvalue,
  "p-value.boot"=pvalue.boot),
              distr = distr, estim= estim,
              param = c(shape = alpha, shape2 = gamma,
                        location = mu, scale = beta))
  return(output)
}

# Function to compute the statistic

boot.fun <- function(data, M, r=M-1, step=1, distr, estim,
               beta.limits=c(0,1), parameters=list(shape=NULL,
               shape2 = NULL, location = NULL, scale = NULL),
               K=diag(M)) {

  require(fitdistrplus)
  require(bbmle)
  require(eha)  # per la loglogis
```

```
require(FAdist) # per la gumbel

n <- length(data)
alpha <- parameters$shape
gamma <- parameters$shape2
mu <- parameters$location
beta <- parameters$scale

# Compute the cells boundaries r=1, 2, M-1
i<-0
while(i!=2){
  q <- numeric(M+1)
  if (r == 1) {
    if (M%%2 == 0) {
      b <- seq(((-M/2+1)*step), ((M/2-1)*step), by=step)
      q[2:M] <- b+rep(mean(data), length(b))
      q[1] <- min(data)-M/2*step
      q[M+1] <- max(data)+M/2*step}
    else {
      b <- seq(((-as.integer(M/2)+1)*step), ((as.integer(M
               /2))*step), by=step)
      q[2:M] <- b+rep(mean(data), length(b))
      q[1] <- min(data)-M/2*step
      q[M+1] <- max(data)+M/2*step}
  }
  else {
    if (r == 2) {
      if (M%%2 == 0) {
        b <- seq(((-M/2+1)*step), ((M/2-1)*step), by=step)
        q[2:M] <- sd(data)*b+rep(mean(data), length(b))
        q[1] <- min(data)-M/2*sd(data)*step
        q[M+1] <- max(data)+M/2*sd(data)*step}
      else {
        b <- seq(((-as.integer(M/2)+1)*step), ((as.integer(M
             /2))*step), by=step)
        q[2:M] <- sd(data)*b+rep(mean(data), length(b))
        q[1] <- min(data)-M/2*sd(data)*step
        q[M+1] <- max(data)+M/2*sd(data)*step}
    }
    else {
      q <- unique(quantile(data, probs=seq(0, 1, 1/M)))}
  }
  if(is.element(0,q)){
    r<-M-1
    i<-i+1
    warning('The values of r and step are not compatible. The
     function will be executed for r=M-1')
  }
  else{
    i<-2
```

```
  }
}

# Compute the observed frequencies
q[M+1] <- q[M+1]+1
cells.cut = cut(data, q, right=FALSE)
obs.freq = as.vector(table(cells.cut))

# Determine the theoretical distribution and estimate
# its parameters
# Compute the expected probabilities

# Weibull distribution
if (distr == "weibull") {
  if (is.null(alpha) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdist(data, "weibull")
      alpha <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      weibull.funcmax <- function(alpha, beta) {
        -sum(obs.freq*log(pweibull(q[1:M+1],
         alpha, beta)-pweibull(q[1:M], alpha, beta)))
      }
      param <- mle2(weibull.funcmax, start = list(alpha =1,
       beta = 4), method="Nelder-Mead")
      alpha <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  m=2
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- pweibull(q[i+1], alpha, beta)-pweibull(q[i],
     alpha, beta)
  }
}

# Gumbel distribution
if(distr=="gumbel"){
  dgumb <<- function(x,mu,beta)
    1/beta*exp(-(x-mu)/beta)*exp(-exp(-(x-mu)/beta))
  pgumb <<- function(q,mu,beta) 1-exp(-exp(-(q-mu)/beta))
  if(is.null(mu) || is.null(beta)){
    if(estim=='orig'){
      param<-fitdist(data,"gumb",start=list(mu=median(data)-
      0.28*sd(data),beta=0.78*sd(data)))
      mu<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])}
```

```
    if(estim=='obsfreq'){
      gumbel.funcmax <- function(mu, beta){
        -sum(obs.freq*log(pgumbel(q[1:M+1],beta,mu)
        -pgumbel(q[1:M],beta,mu)))
      }
      param <- mle2(gumbel.funcmax,start = list(mu=median(data)
      -0.28*sd(data),beta=0.78*sd(data)), method="Nelder-Mead")
      mu<-unname(coef(param)[1])
      beta<-unname(coef(param)[2])}
  }
  m=2
  exp.prob<-numeric(M)
  for (i in 1:M){
    exp.prob[i]<-pgumbel(q[i+1],beta,mu)-pgumbel(q[i],beta,mu)
  }
}

# Normal distribution
if (distr == "norm") {
  if (is.null(mu) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdist(data, "norm")
      mu <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      norm.funcmax <- function(mu, beta) {
        -sum(obs.freq*log(pnorm(q[1:M+1], mu, beta)-pnorm(q[1:M],
         mu, beta)))
      }
      param <- mle2(norm.funcmax, start = list(mu = mean(data),
       beta = sd(data)), method="Nelder-Mead")
      mu <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  m=2
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- pnorm(q[i+1], mu, beta)-pnorm(q[i], mu, beta)
  }
}

# Log-normal distribution
if (distr == "lnorm") {
  if (is.null(mu) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdist(data, "lnorm")
      mu <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
```

```
    }
    if (estim == 'obsfreq') {
      lnorm.funcmax <- function(mu, beta) {
        -sum(obs.freq*log(plnorm(q[1:M+1],mu,beta)
        -plnorm(q[1:M], mu, beta)))
      }
      param <- mle2(lnorm.funcmax, start=list(mu=mean(data),
       beta = sd(data)), method="Nelder-Mead")
      mu <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  m=2
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- plnorm(q[i+1],mu,beta)-plnorm(q[i],mu,beta)
  }
}

# Logistic distribution
if (distr == "logis") {
  if (is.null(mu) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdist(data, "logis")
      mu <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      logis.funcmax <- function(mu, beta) {
        -sum(obs.freq*log(plogis(q[1:M+1],mu,beta)-plogis(q[1:M],
         mu, beta)))
      }
      param <- mle2(logis.funcmax, start = list(mu = mean(data),
       beta = 0.55*sd(data)), method="Nelder-Mead")
      mu <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  m=2
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- plogis(q[i+1],mu,beta)-plogis(q[i],mu,beta)
  }
}

# Log-Logistic distribution
if (distr == "loglogis") {
  dloglogis <<- function(x, alpha, beta) {
    alpha*beta^(-alpha)*x^(alpha-1)/(1+(x/beta)^alpha)^2
  }
```

```
  ploglogis <<- function(q, alpha, beta) 1/(1+(q/beta)^(-alpha))
  if (is.null(alpha) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdist(data, "loglogis", start=list(alpha=1,
       beta=median(data)))
      alpha <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      loglogis.funcmax <- function(alpha, beta) {
        -sum(obs.freq*log(pllogis(q[1:M+1], alpha, beta)-
        pllogis(q[1:M], alpha, beta)))
      }
      param <- mle2(loglogis.funcmax, start = list(alpha = 1,
          beta = median(data)), method="Nelder-Mead")
      alpha <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  m=2
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- pllogis(q[i+1], alpha, beta)-pllogis(q[i],
    alpha, beta)
  }
}

# Beta distribution
if (distr == "beta") {
  a.beta <- beta.limits[1]
  b.beta <- beta.limits[2]
  if (is.null(alpha) || is.null(gamma)) {
    if (estim == 'orig') {
      param <- fitdist((data-a.beta)/(b.beta-a.beta), "beta")
      alpha <- unname(param$estimate[1])
      gamma <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      beta.funcmax <- function(alpha, gamma) {
        -sum(obs.freq*log(pbeta((q[1:M+1]-a.beta)/(b.beta-a.beta),
         alpha, gamma)-pbeta((q[1:M]-a.beta)/(b.beta-a.beta),
         alpha, gamma)))
      }
      param <- mle2(beta.funcmax, start=list(alpha=1, gamma=1),
       method="Nelder-Mead")
      alpha <- unname(coef(param)[1])
      gamma <- unname(coef(param)[2])
    }
  }
  m=2
```

```
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- pbeta((q[i+1]-a.beta)/(b.beta-a.beta),
    alpha, gamma)-pbeta((q[i]-a.beta)/(b.beta-a.beta),
     alpha, gamma)
  }
}

# Exponentiated Weibull distribution
if(distr=="expweibull"){
  dexpwei <<- function(x,alpha,gamma,beta){
    gamma*alpha*beta^alpha*x^(alpha-1)*exp(-(beta*x)^alpha)*
      (1-exp(-(beta*x)^alpha))^(gamma-1)}
  pexpwei <<- function(q,alpha,gamma,beta) (1-exp(-(beta*q)
  ^alpha))^gamma
  if(is.null(alpha) || is.null(gamma) || is.null(beta)){
    if(estim=='orig'){
      param<-fitdist(data,"expwei",start=list(alpha=1,gamma=1,
      beta=1))
      alpha<-unname(param$estimate[1])
      gamma<-unname(param$estimate[2])
      beta<-unname(param$estimate[3])}
    if(estim=='obsfreq'){
      expwei.funcmax <- function(alpha,gamma,beta){
        -sum(obs.freq*log(pexpwei(q[1:M+1],alpha, gamma, beta)
        -pexpwei(q[1:M], alpha, gamma, beta)))
      }
      param <- mle2(expwei.funcmax,start = list(alpha=1,
       gamma=1, beta = 1), method="Nelder-Mead")
      alpha<-unname(coef(param)[1])
      gamma<-unname(coef(param)[2])
      beta<-unname(coef(param)[3])}
  }
  m=3
  exp.prob<-numeric(M)
  for (i in 1:M){
    exp.prob[i]<-pexpwei(q[+1],alpha,gamma,beta)
     -pexpwei(q[i],alpha,gamma,beta)
  }
}

# Exponential power
if (distr == "exppower") {
  dexppow <<- function(x, alpha, beta) {
    alpha*beta^alpha*x^(alpha-1)*exp((beta*x)^alpha)
    *exp(1-exp((beta*x)^alpha))
  }
  pexppow <<- function(q, alpha, beta)
    1-exp(1-exp((beta*q)^alpha))
  if (is.null(alpha) || is.null(beta)) {
```

```
      if (estim == 'orig') {
        param <- fitdist(data, "exppow", start=list(alpha=0.5,
         beta=0.5))
        alpha <- unname(param$estimate[1])
        beta <- unname(param$estimate[2])
      }
      if (estim == 'obsfreq') {
        exppow.funcmax <- function(alpha, beta) {
          -sum(obs.freq*log(pexppow(q[1:M+1], alpha, beta)-
          pexppow(q[1:M], alpha, beta)))
        }
        param <- mle2(exppow.funcmax, start = list(alpha =0.25,
         beta = .5), method="Nelder-Mead")
        alpha <- unname(coef(param)[1])
        beta <- unname(coef(param)[2])
      }
    }
    m=2
    exp.prob <- numeric(M)
    for (i in 1:M) {
      exp.prob[i] <- pexppow(q[i+1], alpha, beta)-pexppow(q[i],
       alpha, beta)
    }
  }

  # Computation of the Observed statistic
  if(is.element(0,exp.prob)){
    stop('The values of r and step are incompatibles,
    the statistic can not be computed.')
  }
  v <- (obs.freq-n*exp.prob)/sqrt(n*exp.prob)
  tn <- t(v)%*% K%*% v

  output <- list(tn= tn, m=m, param = list(shape = alpha,
   shape2 = gamma, location = mu, scale = beta))
}
```

Appendix C

GenChiCensv1 code

```
GenChiCensv1 <- function(x, c, M, distr, estim, beta.limits=c(0,1),
                parameters = list(shape = NULL,
                shape2 = NULL, location = NULL, scale = NULL)) {

  require(FAdist)  # per la Gumbel
  require(eha)  # per la loglogis
  require(fitdistrplus)
  require(bbmle)
  require(survival)

  n <- length(x)
  alpha <- parameters$shape
  gamma <- parameters$shape2
  mu <- parameters$location
  beta <- parameters$scale

  # censored data
  d<-data.frame(time=x, cens=c, count=rep(1,n))
  data<-data.frame(left=x,right=ifelse(c==1,x,NA))
  data2<-Surv(x, c)
  fit<-survfit(data2~1)

  # Compute the cells boundaries

  q1 <- unique(quantile(fit, probs=seq(0, 1, 1/M))$quantile)
  i<-1
  while(!is.na(q1[i]) && i!=(M+2)){
    i=i+1
  }
  if(i!=(M+2)){
    M<-i-2
    q<-q1[1:(i-1)]
    warning('The value of M has change due to the presence
        of NA quantiles')
  }
  else{
    q<-q1
  }
```

```
# Compute the observed frequencies
q[M+1] <- q[M+1]+1
cells.cut = cut(d$time[d$cens==1], q, right=FALSE)
obs.freq = as.vector(table(cells.cut))

# Determine the theoretical distribution and estimate
# its parameters
# Compute the expected probabilities

# Weibull distribution
if (distr == "weibull") {
  if (is.null(alpha) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdistcens(data, "weibull")
      alpha <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      weibull.funcmax <- function(alpha, beta) {
        -sum(obs.freq*log(pweibull(q[1:M+1], alpha, beta)
        -pweibull(q[1:M], alpha, beta)))
      }
      param <- mle2(weibull.funcmax, start = list(alpha =1,
      beta = 4), method="Nelder-Mead")
      alpha <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- pweibull(q[i+1], alpha, beta)
    -pweibull(q[i], alpha, beta)
  }
}

# Gumbel distribution
if(distr=="gumbel"){
  dgumb <<- function(x,mu,beta)
    1/beta*exp(-(x-mu)/beta)*exp(-exp(-(x-mu)/beta))
  pgumb <<- function(q,mu,beta) 1-exp(-exp(-(q-mu)/beta))
  if(is.null(mu) || is.null(beta)){
    if(estim=='orig'){
      param<-fitdistcens(data,"gumb",start=list(mu=2,beta=4))
      mu<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])}
    if(estim=='obsfreq'){
      gumbel.funcmax <- function(mu, beta){
        -sum(obs.freq*log(pgumbel(q[1:M+1], beta, mu)
        -pgumbel(q[1:M], beta, mu)))
```

```
      }
      param <- mle2(gumbel.funcmax,start = list(mu=2,beta=4),
      method="Nelder-Mead")
      mu<-unname(coef(param)[1])
      beta<-unname(coef(param)[2])}
  }
  exp.prob<-numeric(M)
  for (i in 1:M){
    exp.prob[i]<-pgumbel(q[i+1], beta, mu)
    -pgumbel(q[i], beta, mu)
  }
  rm(dgumb,pos = ".GlobalEnv")
  rm(pgumb,pos = ".GlobalEnv")
}

# Normal distribution
if (distr == "norm") {
  if (is.null(mu) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdistcens(data, "norm")
      mu <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      norm.funcmax <- function(mu, beta) {
        -sum(obs.freq*log(pnorm(q[1:M+1], mu, beta)
        -pnorm(q[1:M], mu, beta)))
      }
      param <- mle2(norm.funcmax, start = list(mu=4, beta=2),
       method="Nelder-Mead")
      mu <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- pnorm(q[i+1],mu,beta)-pnorm(q[i],mu,beta)
  }
}

# Log-normal distribution
if (distr == "lnorm") {
  if (is.null(mu) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdistcens(data, "lnorm")
      mu <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      lnorm.funcmax <- function(mu, beta) {
```

```
        -sum(obs.freq*log(plnorm(q[1:M+1], mu, beta)
        -plnorm(q[1:M], mu, beta)))
      }
      param <- mle2(lnorm.funcmax, start = list(mu=1, beta=1),
      method="Nelder-Mead")
      mu <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <-plnorm(q[i+1],mu,beta)-plnorm(q[i],mu,beta)
  }
}

# Logistic distribution
if (distr == "logis") {
  if (is.null(mu) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdistcens(data, "logis")
      mu <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      logis.funcmax <- function(mu, beta) {
        -sum(obs.freq*log(plogis(q[1:M+1], mu, beta)
        -plogis(q[1:M], mu, beta)))
      }
      param <- mle2(logis.funcmax, start = list(mu=1, beta=3),
       method="Nelder-Mead")
      mu <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- plogis(q[i+1],mu,beta)-plogis(q[i],mu,beta)
  }
}

# Log-Logistic distribution
if (distr == "loglogis") {
  dloglogis <<- function(x, alpha, beta) {
    alpha*beta^(-alpha)*x^(alpha-1)/(1+(x/beta)^alpha)^2
  }
  ploglogis <<- function(q, alpha, beta) 1/(1+(q/beta)^(-alpha))
  if (is.null(alpha) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdistcens(data, "loglogis", start=list(alpha=1,
       beta=2))
```

```
      alpha <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      loglogis.funcmax <- function(alpha, beta) {
        -sum(obs.freq*log(pllogis(q[1:M+1], alpha, beta)
        -pllogis(q[1:M], alpha, beta)))
      }
      param <- mle2(loglogis.funcmax, start = list(alpha = 1,
       beta = 2), method="Nelder-Mead")
      alpha <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- pllogis(q[i+1], alpha, beta)
    -pllogis(q[i], alpha, beta)
  }
  rm(ploglogis,pos = ".GlobalEnv")
  rm(dloglogis,pos = ".GlobalEnv")
}

# Beta distribution
if (distr == "beta") {
  a.beta <- beta.limits[1]
  b.beta <- beta.limits[2]
  if (is.null(alpha) || is.null(gamma)) {
    if (estim == 'orig') {
      param <- fitdistcens((data-a.beta)/(b.beta-a.beta),"beta")
      alpha <- unname(param$estimate[1])
      gamma <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      beta.funcmax <- function(alpha, gamma) {
        -sum(obs.freq*log(pbeta((q[1:M+1]-a.beta)/(b.beta-a.beta),
         alpha, gamma)-pbeta((q[1:M]-a.beta)/(b.beta-a.beta),
          alpha, gamma)))
      }
      param <- mle2(beta.funcmax, start = list(alpha=1,gamma=1),
      method="Nelder-Mead")
      alpha <- unname(coef(param)[1])
      gamma <- unname(coef(param)[2])
    }
  }
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- pbeta((q[i+1]-a.beta)/(b.beta-a.beta),
     alpha, gamma)-pbeta((q[i]-a.beta)/(b.beta-a.beta),
     alpha, gamma)
```

```
  }
}

# Exponentiated Weibull distribution
if(distr=="expweibull"){
  dexpwei <<- function(x,alpha,gamma,beta){
    gamma*alpha*beta^alpha*x^(alpha-1)*exp(-(beta*x)^alpha)*
      (1-exp(-(beta*x)^alpha))^(gamma-1)}
  pexpwei <<- function(q,alpha,gamma,beta)
  (1-exp(-(beta*q)^alpha))^gamma
  if(is.null(alpha) || is.null(gamma) || is.null(beta)){
    if(estim=='orig'){
      param<-fitdistcens(data,"expwei",start=list(alpha=1,
      gamma=1,beta=1))
      alpha<-unname(param$estimate[1])
      gamma<-unname(param$estimate[2])
      beta<-unname(param$estimate[3])}
    if(estim=='obsfreq'){
      expwei.funcmax <- function(alpha,gamma,beta){
        -sum(obs.freq*log(pexpwei(q[1:M+1],alpha, gamma, beta)
        -pexpwei(q[1:M], alpha, gamma, beta)))
      }
      param <- mle2(expwei.funcmax,start = list(alpha=1,
      gamma=1, beta =1), method="Nelder-Mead")
      alpha<-unname(coef(param)[1])
      gamma<-unname(coef(param)[2])
      beta<-unname(coef(param)[3])}
  }
  exp.prob<-numeric(M)
  for (i in 1:M){
    exp.prob[i]<-pexpwei(q[i+1],alpha,gamma,beta)-
    pexpwei(q[i],alpha,gamma,beta)
  }
  rm(pexpwei,pos = ".GlobalEnv")
  rm(dexpwei,pos = ".GlobalEnv")
}

# Exponential power
if (distr == "exppower") {
  dexppow <<- function(x, alpha, beta) {
    alpha*beta^alpha*x^(alpha-1)*exp((beta*x)^alpha)
    *exp(1-exp((beta*x)^alpha))
  }
  pexppow <<- function(q, alpha, beta)
    1-exp(1-exp((beta*q)^alpha))
  if (is.null(alpha) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdistcens(data, "exppow", start=list(alpha=1,
      beta=0.5))
      alpha <- unname(param$estimate[1])
```

```
        beta <- unname(param$estimate[2])
      }
      if (estim == 'obsfreq') {
        exppow.funcmax <- function(alpha, beta) {
          -sum(obs.freq*log(pexppow(q[1:M+1], alpha, beta)
          -pexppow(q[1:M], alpha, beta)))
        }
        param <- mle2(exppow.funcmax, start = list(alpha =0.5,
         beta = .5), method="Nelder-Mead")
        alpha <- unname(coef(param)[1])
        beta <- unname(coef(param)[2])
      }
    }
    exp.prob <- numeric(M)
    for (i in 1:M) {
      exp.prob[i] <- pexppow(q[i+1], alpha, beta)
      -pexppow(q[i], alpha, beta)
    }
    rm(pexppow,pos = ".GlobalEnv")
    rm(dexppow,pos = ".GlobalEnv")
  }

  # Computation of the Observed statistic
  if(is.element(0,exp.prob)){
    stop('The some of the expected probabilities are 0.')
  }
  v <- (obs.freq-n*exp.prob)/sqrt(n*exp.prob)
  tn <- t(v)%*% v

  # Results
  output <- list(Estad= tn, distr = distr, estim= estim,
  param = c(shape = alpha, shape2 = gamma,
                location = mu, scale = beta))
  return(output)
}
```

Appendix D

GenChiCensv2 code

```
GenChiCensv2 <- function(data, c, M, distr.data, distr.cens, estim,
                parameters = list(shape = NULL, shape2 = NULL,
                 location = NULL, scale = NULL)) {

  require(FAdist)  # per la Gumbel
  require(eha)  # per la loglogis
  require(survsim)

  n <- length(data)
  m<-max(data)

  est<-boot2.fun(data, c, M, distr.data, estim, parameters)
  tn<-est$tn
  M<-est$M

  # estimate parameters for the data distribution
  parameters<-est$param
  alpha <- parameters$shape
  gamma <- parameters$shape2
  mu <- parameters$location
  beta <- parameters$scale


  # estimate parameters for the censoring distribution
  c1<-c
  c1[c==0]<-1
  c1[c==1]<-0

  est.cens<-boot2.fun(data,c1,M, distr.cens, estim, parameters,
  cens=TRUE)
  parameters.cens<-est.cens$param
  alpha.cens <- parameters.cens$shape
  gamma.cens <- parameters.cens$shape2
  mu.cens <- parameters.cens$location
  beta.cens <- parameters.cens$scale

  # Computation of the p-value

  t<-numeric(1000)
```

```
if(distr.data=="weibull"){
  if(distr.cens=='weibull'){
    for(i in 1:1000){
      rand<-simple.surv.sim(n, m,  dist.ev='weibull',alpha,
      -log(beta),dist.cens='weibull',alpha.cens,-log(beta.cens))
      x<-as.vector(rand$stop)
      c<-as.vector(rand$status)
      t[i]<-boot2.fun(x,c, M=M, distr='weibull',estim, parameters
      =list(shape = alpha, shape2= NULL,location = NULL,
      scale = beta))$tn
    }}
  if(distr.cens=='lnorm'){
    for(i in 1:1000){
      rand<-simple.surv.sim(n, m,  dist.ev='weibull',alpha,
      -log(beta),dist.cens='lnorm',beta.cens,mu.cens)
      x<-as.vector(rand$stop)
      c<-as.vector(rand$status)
      t[i]<-boot2.fun(x,c, M=M, distr='weibull', estim,parameters
       = list(shape = alpha, shape2 = NULL,location = NULL,
        scale = beta))$tn
    }}
  if(distr.cens=='loglogis'){
    for(i in 1:1000){
      rand<-simple.surv.sim(n, m,  dist.ev='weibull',alpha,
      -log(beta),dist.cens='llogistic',1/alpha.cens,log(beta.cens))
      x<-as.vector(rand$stop)
      c<-as.vector(rand$status)
      t[i]<-boot2.fun(x,c, M=M, distr='weibull',estim, parameters =
       list(shape = alpha, shape2 = NULL,location = NULL,
       scale = beta))$tn
    }
    rm(ploglogis,pos = ".GlobalEnv")
    rm(dloglogis,pos = ".GlobalEnv")
    }
  if(distr.cens=='unif'){
    for(i in 1:1000){
      rand<-simple.surv.sim(n, m,  dist.ev='weibull',alpha,
      -log(beta), dist.cens='unif',gamma.cens,alpha.cens)
      x<-as.vector(rand$stop)
      c<-as.vector(rand$status)
      t[i]<-boot2.fun(x,c, M=M, distr='weibull',estim, parameters
      = list(shape = alpha, shape2 = NULL,location = NULL,
        scale = beta))$tn
    }}
  }

if(distr.data=='lnorm'){
  if(distr.cens=='weibull'){
    for(i in 1:1000){
      rand<-simple.surv.sim(n, m,  dist.ev='lnorm',beta,mu,
```

```
        dist.cens='weibull',alpha.cens,-log(beta.cens))
      x<-as.vector(rand$stop)
      c<-as.vector(rand$status)
      t[i]<-boot2.fun(x,c, M=M, distr='lnorm',estim, parameters
      =list(shape = alpha, shape2= NULL,location = NULL,
       scale = beta))$tn
    }}
  if(distr.cens=='lnorm'){
    for(i in 1:1000){
      rand<-simple.surv.sim(n, m,  dist.ev='lnorm',beta,mu,
       dist.cens='lnorm',beta.cens,mu.cens)
      x<-as.vector(rand$stop)
      c<-as.vector(rand$status)
      t[i]<-boot2.fun(x,c, M=M, distr='lnorm',estim, parameters
     = list(shape = alpha, shape2 = NULL,location = NULL,
      scale = beta))$tn
    }}
  if(distr.cens=='loglogis'){
    for(i in 1:1000){
      rand<-simple.surv.sim(n, m,  dist.ev='lnorm',beta,mu,
       dist.cens='llogistic',1/alpha.cens,log(beta.cens))
      x<-as.vector(rand$stop)
      c<-as.vector(rand$status)
      t[i]<-boot2.fun(x,c, M=M, distr='lnorm', estim,parameters
       =list(shape = alpha, shape2 = NULL,location = NULL,
       scale = beta))$tn
    }
    rm(ploglogis,pos = ".GlobalEnv")
    rm(dloglogis,pos = ".GlobalEnv")
    }
  if(distr.cens=='unif'){
    for(i in 1:1000){
      rand<-simple.surv.sim(n, m,  dist.ev='lnorm',beta,mu,
       dist.cens='unif',gamma.cens,alpha.cens)
      x<-as.vector(rand$stop)
      c<-as.vector(rand$status)
      t[i]<-boot2.fun(x,c, M=M, distr='lnorm', estim,parameters
      =list(shape = alpha, shape2 = NULL,location = NULL,
       scale = beta))$tn
    }}}

if(distr.data=='loglogis'){
  if(distr.cens=='weibull'){
    for(i in 1:1000){
      rand<-simple.surv.sim(n, m,  dist.ev='llogistic',1/alpha,
      log(beta), dist.cens='weibull',alpha.cens,-log(beta.cens))
      x<-as.vector(rand$stop)
      c<-as.vector(rand$status)
      t[i]<-boot2.fun(x,c, M=M, distr='loglogis',estim,parameters
       = list(shape = alpha, shape2= NULL,location = NULL,
```

```
          scale = beta))$tn
      }}
    if(distr.cens=='lnorm'){
      for(i in 1:1000){
        rand<-simple.surv.sim(n, m,  dist.ev='llogistic',1/alpha,
        log(beta), dist.cens='lnorm',beta.cens,mu.cens)
        x<-as.vector(rand$stop)
        c<-as.vector(rand$status)
        t[i]<-boot2.fun(x,c, M=M, distr='loglogis', estim,parameters
        = list(shape = alpha, shape2 = NULL,location = NULL,
         scale = beta))$tn
      }}
    if(distr.cens=='loglogis'){
      for(i in 1:1000){
        rand<-simple.surv.sim(n, m,  dist.ev='llogistic',1/alpha,
        log(beta), dist.cens='llogistic',1/alpha.cens,log(beta.cens))
        x<-as.vector(rand$stop)
        c<-as.vector(rand$status)
        t[i]<-boot2.fun(x,c, M=M, distr='loglogis',estim,parameters
         = list(shape = alpha, shape2 = NULL,location = NULL,
         scale = beta))$tn
      }}
    if(distr.cens=='unif'){
      for(i in 1:1000){
        rand<-simple.surv.sim(n, m,  dist.ev='llogistic',1/alpha,
        log(beta), dist.cens='unif',gamma.cens,alpha.cens)
        x<-as.vector(rand$stop)
        c<-as.vector(rand$status)
        t[i]<-boot2.fun(x,c, M=M, distr='loglogis',estim,parameters
         = list(shape = alpha, shape2 = NULL,location = NULL,
         scale = beta))$tn
      }}
    rm(ploglogis,pos = ".GlobalEnv")
    rm(dloglogis,pos = ".GlobalEnv")
  }


  pvalue.boot <- 1 - ecdf(t)(tn)

  # Results
  output <- list(test = c('Estad'= tn,  "p-value.boot"=pvalue.boot),
                 distr.data = distr.data, distr.cens=distr.cens,
                  estim= estim, param = c(shape = alpha, shape2
                   = gamma, location = mu, scale = beta))
  return(output)
}


boot2.fun <- function(x, c, M, distr, estim, parameters = list(
             shape = NULL, shape2= NULL, location = NULL, scale
```

```
           = NULL),cens=FALSE) {

require(fitdistrplus)
require(bbmle)
require(eha)  # per la loglogis
require(FAdist) # per la gumbel
require(survival)

n <- length(data)
alpha <- parameters$shape
gamma <- parameters$shape2
mu <- parameters$location
beta <- parameters$scale

# censored data
d<-data.frame(time=x, cens=c, count=rep(1,n))
data<-data.frame(left=x,right=ifelse(c==1,x,NA))
data2<-Surv(x, c)
fit<-survfit(data2~1)

# Compute the cells boundaries

q1 <- unique(quantile(fit, probs=seq(0, 1, 1/M))$quantile)
i<-1
while(!is.na(q1[i]) && i!=(M+2)){
  i=i+1
}
if(i<2){
  stop('Too much censored data, we can not compute
  the statistic')
}
if(i!=(M+2)){
  M<-(i-2)
  q<-q1[1:(i-1)]
  warning('The value of M has change due to the presence of
  NA quantiles')
}
else{
  q<-q1
}

# Compute the observed frequencies
q[M+1] <- q[M+1]+1
cells.cut = cut(d$time[d$cens==1], q, right=FALSE)
obs.freq = as.vector(table(cells.cut))

# Determine the theoretical distribution and estimate
# its parameters
# Compute the expected probabilities
```

```
# Weibull distribution
if (distr == "weibull") {
  if (is.null(alpha) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdistcens(data, "weibull")
      alpha <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      weibull.funcmax <- function(alpha, beta) {
        -sum(obs.freq*log(pweibull(q[1:M+1], alpha, beta)
        -pweibull(q[1:M], alpha, beta)))
      }
      param <- mle2(weibull.funcmax, start = list(alpha =1,
      beta = 4), method="Nelder-Mead")
      alpha <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- pweibull(q[i+1], alpha, beta)
    -pweibull(q[i], alpha, beta)
  }
}

# Log-normal distribution
if (distr == "lnorm") {
  if (is.null(mu) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdistcens(data, "lnorm")
      mu <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      lnorm.funcmax <- function(mu, beta) {
        -sum(obs.freq*log(plnorm(q[1:M+1], mu, beta)
        -plnorm(q[1:M], mu, beta)))
      }
      param <- mle2(lnorm.funcmax, start = list(mu = 1,
       beta = 2), method="Nelder-Mead")
      mu <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- plnorm(q[i+1], mu, beta)
    -plnorm(q[i], mu, beta)
  }
```

```
}

# Log-Logistic distribution
if (distr == "loglogis") {
  dloglogis <<- function(x, alpha, beta) {
    alpha*beta^(-alpha)*x^(alpha-1)/(1+(x/beta)^alpha)^2
  }
  ploglogis <<- function(q, alpha, beta) 1/(1+(q/beta)^(-alpha))
  if (is.null(alpha) || is.null(beta)) {
    if (estim == 'orig') {
      param <- fitdistcens(data, "loglogis", start=list(alpha=1,
      beta=2))
      alpha <- unname(param$estimate[1])
      beta <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      loglogis.funcmax <- function(alpha, beta) {
        -sum(obs.freq*log(pllogis(q[1:M+1], alpha, beta)
        -pllogis(q[1:M], alpha, beta)))
      }
      param <- mle2(loglogis.funcmax, start = list(alpha = 1,
       beta = 2), method="Nelder-Mead")
      alpha <- unname(coef(param)[1])
      beta <- unname(coef(param)[2])
    }
  }
  exp.prob <- numeric(M)
  for (i in 1:M) {
    exp.prob[i] <- pllogis(q[i+1], alpha, beta)
    -pllogis(q[i], alpha, beta)
  }
}

# Uniform distribution
if (distr == "unif") {
  if (is.null(alpha) || is.null(gamma)) {
    if (estim == 'orig') {
      param <- fitdistcens(data, "unif")
      alpha <- unname(param$estimate[1])
      gamma <- unname(param$estimate[2])
    }
    if (estim == 'obsfreq') {
      unif.funcmax <- function(alpha, gamma) {
        -sum(obs.freq*log(punif(q[1:M+1], alpha, gamma)
        -punif(q[1:M], alpha, gamma)))
      }
      param <- mle2(unif.funcmax, start = list(alpha = 0,
       gamma = 1), method="Nelder-Mead")
      alpha <- unname(coef(param)[1])
      gamma <- unname(coef(param)[2])
```

```
      }
    }
    exp.prob <- numeric(M)
    for (i in 1:M) {
      exp.prob[i] <- punif(q[i+1], alpha, gamma)
      -punif(q[i], alpha, gamma)
    }
  }

  # Computation of the Observed statistic
  if(is.element(0,exp.prob)){
    stop('The some of the expected probabilities are 0.')
  }
  tn<-NULL
  if(cens==FALSE){
    v <- (obs.freq-n*exp.prob)/sqrt(n*exp.prob)
    tn <- t(v)%*% v
  }

  output <- list(tn=tn, M=M, param=list(shape=alpha,
  shape2=gamma,location=mu, scale=beta))
}
```

Bibliography

[1] M. G. Akritas. "Pearson-Type Goodness-of-Fit Tests: The Univariate Case". In: *Journal of the American Statistical Association* 83.401 (1988), pp. 222–230.

[2] T. W Anderson and D. A. Darling. "A test of goodness of fit". In: *Journal of the American Statistical Association* 49.268 (1954), 765–769.

[3] N. Balakrishnan, E. Chimitova, and M. Vedernikova. "An Empirical Analysis of Some Nonparametric Goodness-of-Fit Tests for Censored Data". In: *Communications in Statistics - Simulation and Computation* 44.4 (2015), pp. 1101–1115.

[4] D. R. Barr and T. Davidson. "A Kolmogorov-Smirnov Test for Censored Samples". In: *Technometrics* 15.4 (1973), pp. 739–757.

[5] P. Biecek and T. Ledwina. *ddst: Data Driven Smooth Tests*. 2016. URL: http://CRAN.R-project.org/package=ddst.

[6] Ben Bolker and R Development Core Team. *bbmle: Tools for General Maximum Likelihood Estimation*. 2016. URL: http://CRAN.R-project.org/package=bbmle.

[7] C. Castro-Kuriss et al. "A New Goodness-of-Fit Test for Censored Data with an Application in Monitoring Processes". In: *Communications in Statistics - Simulation and Computation* 38.6 (2009), pp. 1161–1177.

[8] C. Chen. *Correlation-Type Goodness-of-Fit Tests for Randomly Censored Data*. Tech. rep. Standford, Department of Statistics, Dec. 1981.

[9] C. H. Chen. "A correlation goodness-of-fit test for randomly censored data". In: *Biometrika* 71.2 (1984), pp. 315–322.

[10] J. Chen. *Ph. D. dissertation: Goodness of fit tests under random censorship*. 1975.

[11] E. Chimitova, H. Liero, and M. Vedernikova. "Application of classical Kolmogorov, Cramer-von Mises-Smirnov and Anderson-Darling test for censored samples". In: *Proccedings of the International Workshop "Applied Methods of Statistical Analysis"* (2011), pp. 176–185.

[12] E. Chimitova et al. "Non-parametric goodness-of-fit tests for censored data". In: *Proccedings of the 7th International Conference on "Mathematical Methods in Reliability: Theory, Methods, Applications"* (2011), pp. 817–823.

[13] H. Cramér. "On the composition of elementary errors". In: *Scandinavian Actuarial Journal* 1928.1 (1928), pp. 13–74.

[14] R. B. D'Agostino. *Goodness-of-Fit-Techniques*. Statistics: A Series of Textbooks and Monographs. Taylor & Francis, 1986.

[15] R. Dufour and U. R. Maag. "Distribution Results for Modified Kolmogorov-Smirnov Statistics for Truncated or Censored". In: *Technometrics* 20.1 (1978), pp. 29–32.

[16] B. Epstein and M. Sobel. "Life testing". In: *Journal of the American Statistical association* 48.263 (1953), 486–502.

[17] A. Febrer Galvany. *Master's degree thesis: Analytical and Graphical Goodness of Fit Methods for Parametric Survival Models with Right-censored Data*. 2015.

[18] T. R. Fleming et al. "Modified Kolmogorov-Smirnov Test Procedures with Application to Arbitrarily Right-Censored Data". In: *Biometrics* 36.4 (1980), pp. 607–625.

[19] J. Fortiana and A. Grané. "Goodness-of-fit tests based on maximum correlations and their orthogonal decompositions". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65.1 (2003), pp. 115–126.

[20] A.G. Glen and B.L. Foote. "An Inference Methodology for Life Tests With Complete Samples or Type-II Right Censoring". In: *Reliability, IEEE Transactions on* 58.4 (2009), pp. 597–603.

[21] C. Goldmann, B. Klar, and S. G. Meintanis. "Data transformations and goodness-of-fit tests for type-II right censored samples". In: *Metrika* 78.1 (2015), pp. 59–83.

[22] A. Grané and E. Strzalkowska-Kominiak. *Goodness-of-fit test for randomly censored data based on maximum correlation*. DES - Working Papers. Statistics and Econometrics. WS. Universidad Carlos III de Madrid. Departamento de Estadística, 2014.

[23] A. Grané. "Exact goodness-of-fit tests for censored data". In: *Annals of the Institute of Statistical Mathematics* 64.6 (2012), pp. 1187–1203.

[24] M. G. Habib and D. R. Thomas. "Chi-Square Goodness-if-Fit Tests for Randomly Censored Data". In: *Ann. Statist.* 14.2 (1986), pp. 759–765.

[25] N. L. Hjort. "Goodness of Fit Tests in Models for Life History Data Based on Cumulative Hazard Rates". In: *Ann. Statist.* 18.3 (1990), pp. 1221–1258.

[26] E. L. Kaplan and P. Meier. "Nonparametric Estimation from Incomplete Observations". In: *Journal of the American Statistical Association* 53.282 (1958), pp. 457–481.

[27] J. H. Kim. "Chi-Square Goodness-of-Fit Tests for Randomly Censored Data". In: *Ann. Statist.* 21.3 (1993), pp. 1621–1639.

[28] A. N. Kolmogorov. "Sulla Determinazione Empirica di una Legge di Distribuzione". In: *Giornale dell'Istituto Italiano degli Attuari* 4 (1933), pp. 83–91.

[29] J. A. Koziol. "Goodness-of-fit tests for randomly censored data". In: *Biometrika* 67.3 (1980), pp. 693–696.

[30] J. A. Koziol and D. P. Byar. "Percentage Points of the Asymptotic Distributions of One and Two Sample K-S Statistics for Truncated or Censored Data". In: *Technometrics* 17.4 (1975), pp. 507–510.

[31] J. A. Koziol and S. B. Green. "A Crámer-von Mises Statistic for Randomly Censored Data". In: *Biometrika* 63.3 (1976), pp. 465–474.

[32] N. H. Kuiper. "Tests concerning random points on a circle". In: *Indagationes Mathematicae (Proceedings)* 63 (1960), pp. 38 –47.

[33] M. L.Delignette-Muller and C. Dutang. "fitdistrplus: An R Package for Fitting Distributions". In: *Journal of Statistical Software* 64.4 (2015), pp. 1–34.

[34] J. R. Michael. "The stabilized probability plot". In: *Biometrika* 70.1 (1983), pp. 11–17.

[35] J. R. Michael and W. R. Schucany. "A New Approach to Testing Goodness of Fit for Censored Samples". In: *Technometrics* 21.4 (1979), pp. 435–441.

[36] D.P. Mihalko and D.S. Moore. "Chi-Square tests of fit for type-II Censored-data". In: *Annals of Statistics* 8.3 (1980), 625–644.

[37] R. E. von Mises. "Wahrscheinlichkeit". In: *Statistik und Wahrheit. Julius Springer, Vienna, Austria.* (1928).

[38] D. S. Moore and M. C. Spruill. "Unified Large-Sample Theory of General Chi-Squared Statistics for Tests of Fit". In: *Ann. Statist.* 3.3 (1975), pp. 599–616.

[39] D. Moriña and A. Navarro. "The R Package survsim for the Simulation of Simple and Complex Survival Data". In: *Journal of Statistical Software* 59.2 (2014), pp. 1–20.

[40] V. N. Nair. "Plots and Tests for Goodness of Fit with Randomly Censored Data". In: *Biometrika* 68.1 (1981), pp. 99–103.

[41] W. Nelson. "Theory and Applications of Hazard Plotting for Censored Failure Data". In: *Technometrics* 14.4 (1972), pp. 945–966.

[42] J. Neyman. "»Smooth test» for goodness of fit". In: *Scandinavian Actuarial Journal* 1937.3-4 (1937), pp. 149–199.

[43] S. Park and M. Shin. "Kullback–Leibler information of a censored variable and its applications". In: *Statistics* 48.4 (2014), pp. 756–765.

[44] E. A. Peña. "Smooth Goodness-Of-Fit Tests for Composite Hypothesis in Hazard Based Models". In: *The Annals of Statistics* 26.5 (1998), pp. 1935–1971.

[45] K. Pearson. "X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling". In: *Philosophical Magazine Series 5* 50.302 (1900), pp. 157–175.

[46] A. Pettitt and M.A. Stephens. "Modified Cramer Von Mises statistics for censored data". In: *Biometrika* 63.2 (1976), 291–298.

[47] A. N. Pettitt. "Cramer-von Mises Statistics for Testing Normality with Censored Samples". In: *Biometrika* 63.3 (1976), pp. 475–481.

[48] A.H. Rad, F. Yousefzadeh, and N. Balakrishnan. "Goodness-of-Fit Test Based on Kullback-Leibler Information for Progressively Type-II Censored Data". In: *Reliability, IEEE Transactions on* 60.3 (2011), pp. 570–579.

[49] J. C. W. Rayner, O. Thas, and D. J. Best. *Smooth Tests of Goodness of Fit: Using R*. Statistics and Probability. Wiley, 2009.

[50] V. Salinas et al. "Goodness of Fit Tests for the Gumbel Distribution with Type II right Censored data". In: *Revista Colombiana de Estadística* 35.3 (2013), pp. 407–422.

[51] R. M. Smith and L. J. Bain. "Correlation type gogdness-of-fit statistics with censored sampling". In: *Communications in Statistics - Theory and Methods* 5.2 (1976), pp. 119–132.

[52] B. W. Turnbull and L. Weiss. "A Likelihood Ratio Statistic for Testing Goodness of Fit with Randomly Censored Data". In: *Biometrics* 34.3 (1978), pp. 367–375.

[53] L.A. Waller and B.W. Turnbull. "Probability plotting with censored data". In: *American Statistician* 46.1 (1992), 5–12.

[54] G.S. Watson. “Goodness-of-fit tests on a circle”. In: *Biometrika* 48.1-2 (1961), 109–114.

[55] G.S. Watson. “Goodness-of-fit tests on a circle .2.” In: *Biometrika* 49.1-2 (1962), 57–63.

[56] M. B. Wilk and R. Gnanadesikan. “Probability Plotting Methods for the Analysis of Data”. In: *Biometrika* 55.1 (1968), pp. 1–17.