Politecnico di Milano

Department of Mathematics

Ph.D. School in

Mathematical Models and Methods in Engineering

Mining Large Administrative Databases: Efficient and Scalable Algorithms for Statistical Modeling

Doctoral Dissertation of:

Francesco Grossetti

Advisor:

Prof. Anna Maria Paganoni

Co-advisor:

Dr. Francesca Ieva

Ph.D. Coordinator:

Prof. Irene Sabadini

XXIX Cycle

È dall'ironia che comincia la libertà
[cit. Victor Hugo, La Légende des siècles]

This page has been intentionally left blank

Table of Contents

Abstract

Healthcare administrative databases are becoming more and more important and reliable sources of clinical and epidemiological information. The present work marks the first Italian attempt which focuses on the acquisition, management and study of several data sources in the form of administrative databases regarding the Heart Failure pathology. All the data used in this thesis have been extracted from the administrative data warehouse of Lombardy Region, a region located in the northern part of Italy whose capital is Milan. One of the main goal of the present work is to identify, extrapolate and build a unique and consistent data structure to be used for statistical and research purposes. The administrative databases are conceived as repositories which are able to store many information but typically for managerial aims. This work is a step forward in moving the focus from a descriptive stand point of view to an inferential one. To achieve this goal, a great effort has been dedicated to the development of efficient algorithms, some of them have been finalized into a `R` package called `msmtools`. Moreover, this work studies the hospital admission-readmission process in order to explore the Heart Failure patient's epidemiology and to profile the health service utilization over time. We also investigate variations in patient care according to geographic area, socio-demographic characteristics as well as other administrative and clinical variables. The heterogeneity of the different data sources is fundamental to better characterize the disease progression and to possibly identify what are the main determinants of a hospital admission, readmission and death in patients with Heart Failure.

Riassunto

I database amministrativi sanitari stanno diventando fonti di informazioni cliniche ed epidemiologiche sempre più importanti e affidabili. Questo lavoro segna il primo tentativo italiano che si concentra sulla acquisizione, la gestione e lo studio di diverse fonti di dati amministrativi con focus su pazienti affetti da scompenso cardiaco. Tutti i dati utilizzati in questa tesi sono stati estratti dalla banca dati di Regione Lombardia. Uno dei principali obiettivi di questo lavoro è quello di identificare, estrapolare e costruire una struttura di dati unica e coerente che possa essere utilizzata a fini statistici e di ricerca. I database amministrativi sono concepiti come repository in grado di memorizzare molte informazioni, tuttavia per scopi tipicamente gestionali. Questo lavoro si propone di spostare l'attenzione da il classico approccio descrittivo ad uno inferenziale. A tal fine, un notevole sforzo è stato dedicato allo sviluppo di algoritmi efficienti, alcuni dei quali hanno portato allo sviluppo di un pacchetto `R` chiamato `msmtools`. Inoltre, questo lavoro si è concentrato sullo studio del processo di ammissione-riammissione ospedaliera al fine di esplorare l'epidemiologia della patologia e di profilare l'utilizzo dei servizi sanitari nel corso del tempo, ma anche di studiare variazioni nella cura del paziente in base alla zona geografica, alle caratteristiche socio-demografiche, nonché ad altre variabili cliniche. L'eterogeneità delle diverse fonti di dati risulta fondamentale per caratterizzare al meglio la progressione della malattia e possibilmente identificare quali sono i principali determinanti di un ricovero ospedaliero, di riammissione e di morte nei pazienti con insufficienza cardiaca.

Acknowledgements

The Heart Failure (HF) data project "*Utilization of regional health service databases for evaluating epidemiology, short and medium-term outcome, and process indexes in patients hospitalized for heart failure*" has been funded by the Italian Ministry of Health (RF2009-1483329) and by the Region of Lombardy which owns the property of all the data described in this work and which are not publicly available.

Beside the main topic and work of this thesis, several side projects have been carried out in parallel during the past three years. The different works have been carried out in collaboration with different clinicians, physiotherapists, biologists belonging to the *Scientific Institute of Lumezzane - Fondazione Salvatore Maugeri, IRCCS* and with *MOXOFF - Mathematics for Innovation*, a spinoff of the Modeling and Scientific Computing (MOX) laboratory at Politecnico di Milano.

My sincerest thanks go to my advisor Anna Paganoni for her invaluable guidance and unstopable support throughout the project. I also thank Francesca Ieva for her hints. Special thanks go to my friends and office mates Nicholas Tarabelloni, Stefano Pagani and Mattia Tamellini who contributed, each in his own spectacular way, to shape the PhD period into something which is going to last as one of my most beautiful memories. I am also grateful to Alice Parodi who undoubtedly belongs to the fellowship as well.

List of Acronyms

Chapter 1

Introduction

The present work is the first Italian attempt which focuses on the acquisition, management and study of several data sources regarding the HF pathology collected by the public healthcare system of Lombardy Region in Italy. We make use of such databases extracted from the Region's data warehouse in order to investigate several aspects. One of the main goal of the present work is to identify, extrapolate and build a unique and consistent data structure to be used for statistical and research purposes. As we will describe later on, administrative databases are conceived as repositories which are able to store many information but typically for managerial aims. This work tries to move the focus from a descriptive stand point of view to an inferential one. Moreover, this work studies the hospital admission-readmission process in order to explore the HF patient's epidemiology and to profile the health service utilization over time. We also investigate variations in patient care according to geographic area, socio-demographic characteristics as well as other administrative and clinical variables. The heterogeneity of the different data sources is fundamental to better characterize the disease progression and to possibly identify what are the main determinants of a hospital admission, readmission and death in patients with HF.

1.1 Structure of the Thesis

The thesis is organized as follows:

Chapter 1.

After a brief introduction to explain the project under which this work has been realized, some references on the HF pathology are provided. In particular, we discuss some of the key characteristics of the disease in order to understand its complex pattern and unpredictable evolution with time as well as its issues when clinicians need to diagnose it.

Chapter 2.

This chapter introduces the reader to the data sources under the context of Administrative Data (AD). In particular, we discuss the necessary steps in order to define a database which could sum up several characteristics related both to the patient and his/her clinical behaviour. We also present each single dataset, the contents as well as the structure which will be of fundamental importance.

Chapter 3.

This chapter is related to the computational challenges faced in the present work. In particular, we discuss the memory allocation and optimization problems, we present and describe the roadmap which we have built in order to achieve a single and usable database for all the subsequent analyses. Finally, a new piece of software in the form of a `R` package called `msmtools` is presented. We also discuss its development process by going into details of the core algorithms used to build the package as well as some of its functionalities.

Chapter 4.

This chapter introduces all the statistical models adopted in this thesis. In particular, we discuss the analysis of longitudinal data through the use of survival analysis and the multi-state models family by introducing some key quantities

which will be estimated later on. We also introduce and explain the multi-state model adopted and discuss the structure of the covariates.

Chapter 5.
This chapter focuses on the results obtained by fitting the models introduced in chapter 4. We discuss the effect of the covariates on the different transitions and we compare the results by using graphical tools. We also evaluate their performances with respect to the observed behaviour.

Chapter 6.
This chapter summarizes what has been done and achieved in the present thesis and also what can be further developed in future works.

Appendix A. This chapter presents all the side projects which have been carried out while working on HF data.

All the developed procedures and analyses carried out in the present work have made use of the `R` programming language version 2.15.3, 3.0.0 and higher [R Core Team, 2016]. Several packages have also been used: they are cited as the work progresses.

1.2 The Chronic Heart Failure Pathology

HF is a clinical syndrome characterized by typical symptoms (e.g. breathlessness, ankle swelling and fatigue) that may be accompanied by signs (e.g. elevated jugular venous pressure, pulmonary crackles and peripheral oedema) caused by a structural and/or functional cardiac abnormality, resulting in a reduced cardiac output and/or elevated intracardiac pressures at rest or during stress [Ponikowski et al., 2015]. In general, HF tends to evolve into a chronic condition thus getting the Chronic Heart Failure (CHF) label. CHF is a complex, heterogeneous clinical syndrome which often requires urgent and continuing therapy [Maggioni, 2015]. The current definition of

CHF restricts itself to stages at which clinical symptoms are apparent. Before any of these symptoms arise, patients are substantially asymptomatic, though with abnormal structural and functional cardiac conditions. For instance, they can present systolic or diastolic Left Ventricular (LV) dysfunctional. Recognition of these precursors is the key of a successful treatment which if starts at the precursor stage may reduce mortality in patients with asymptomatic systolic LV dysfunction [Wang et al., 2003], [Solvd Investigators, 1992]. Demonstration of an underlying cardiac cause is central to the diagnosis of CHF. This is usually a myocardial abnormality causing systolic and/or diastolic ventricular dysfunction. However, abnormalities of the valves, pericardium, endocardium, heart rhythm and conduction can also cause CHF (and more than one abnormality is often present). Identification of the underlying cardiac problem is crucial for therapeutic reasons, as the precise pathology determines the specific treatment used.

CHF is a major public health problem with relevant socioeconomic impact. It is quite complicated to put a number on the prevalence of the disease because it mainly depends on the definition applied. In general, it is approximately $1-2\%$ of the adult population in developed countries, rising to $\geq 10\%$ among people over 70 years of age. Among people over 65 years of age presenting to primary care with breathlessness on exertion, one in six will have unrecognized CHF [Mosterd and Hoes, 2007; Bleumink et al., 2004; Redfield et al., 2003; Ceia et al., 2002]. The lifetime risk of CHF at age 55 years is 33% for men and 28% for women [Bleumink et al., 2004]. Data on temporal trends based on hospitalized patients suggest that the incidence of CHF may be decreasing [Gerber et al., 2015; Owan et al., 2006].

Over the last 30 years, improvements in treatments and their implementation have improved survival and reduced the hospitalization rate. The most recent European data (ESC-CHF pilot study) demonstrate that 12-months all-cause mortality rates for hospitalized and stable/ambulatory CHF patients were 17% and 7%, respectively, and the 12-months hospitalization rates were 44% and 32%, respectively [Maggioni et al., 2013]. In patients with CHF (both hospitalized and ambulatory), most deaths are due

to cardiovascular causes, mainly sudden death and worsening HF. Hospitalizations are often due to non-cardiovascular causes. Hospitalization for cardiovascular causes did not change from 2000 to 2010, whereas those with non-cardiovascular causes increased [Gerber et al., 2015; Bottle et al., 2014; Cowie, 2003].

Many patients will have several different pathologies, cardiovascular and non-cardiovascular, that conspire to cause HF. Identification of these diverse pathologies should be part of the diagnostic workup, as they may offer specific therapeutic opportunities. Co-morbidities are of great importance in CHF since they can interfere with the diagnostic process of CHF [Hawkins et al., 2013; Blondé-Cynober et al., 2011], aggravate CHF symptoms and further impair quality of life [Enjuanes et al., 2014; Hawkins et al., 2013]. They can contribute to the burden of hospitalizations and mortality [Braunstein et al., 2003], as the main cause of readmissions at 1 and 3 months [Muzzarelli et al., 2010], affect the use of treatments for CHF [Reddel et al., 2015], drugs used to treat co-morbidities may cause worsening CHF [Eschenhagen et al., 2011], interaction between drugs used to treat CHF and those used to treat occurrence of side effects. A correct management of co-morbidities is a key component of the holistic care of patients with CHF.

Chapter 2

The Administrative Data

In this chapter, we are going to describe what is to be considered an Administrative Database (AD) and we will focus on a particular set of databases stored by the Lombardy Region (Italy).

In recent years, ADs have become a reliable source of multiple types of information. Their main purpose is to record almost any type of contact between a subject and a complex system. This subject could be anything in between a private citizen, a bank, a firm and all their relative interactions subscriptions. For example, administrative records are maintained to regulate the flow of goods and people across borders, to respond to the legal requirements of registering particular events such as births and deaths, and to administer benefits, such as pensions, or obligations, such a taxation (both for individuals and businesses). As such, the records are collected with a specific decision-making purpose in mind which makes the identity of the unit, corresponding to a given record, absolutely crucial. Furthermore, all the information are collected primarily for administrative purposes thus not with any research or statistical goal. In this work we are going to use ADs with the specific aim of carrying out measures of several quantities of interest through a statistical approach.

As always, there exists a tradeoff when looking at advantages and disadvantages. Let us report some of the most common and acknowledged advantages. ADs are typically very large, covering sample of individuals and time periods which are, in principle, impossible to achieve both financially and logistically through common survey methods. Alongside cost savings, the scope of ADs can be viewed also under a research purpose label. Other advantages include relieving the burden on survey respondents and providing data on individuals who would not normally respond to more classical surveys such questionnaires and they can be linked one to another to build up powerful research resources. They are collected for operational purposes and therefore no additional costs of collection are required. Moreover, the recording process is not intrusive to target population and is carried out routinely and automatically on a time basis which strongly depends on the type of data itself. Sometimes, the update process is continuous. Moreover, they provide historical information and everything is built up on strong consistence, particularly if they are part of national systems. Despite some criticism, they go under rigorous quality checks an they typically cover 100% of population interest so that they can be considered remarkably reliable even at small area level. Beside all these interesting and very powerful aspects, we can not forget that ADs present some disadvantages too which we report in the following. All the information collected is restricted to data required for administrative purposes, thus limited to users of services and administrative definitions. Despite the huge amount of data, sometimes proxy indicators have to be used. A general lack of contextual and/or background information can occur. One of the issue that affects more these data is the presence of missing or erroneous data and the lack of quality controls for those variables which are appointed to be less important to the administrator for whatever reason. This translates in outdated details of some sort. Finally, access for researchers is dependent on support of data providers and, sometimes, it is not of immediate acquisition.

2.1 The Chronic Heart Failure Database

In this work, we will deal with several large ADs focused on the CHF pathology from different point of view. Under this framework, the subjects of our data will be citizens entered the Italian National Public Health System under different circumstances. The databases have been built and are currently managed entirely by Lombardy Region (Italy). They are part of its data warehouse infrastructure which Lombardy Region uses to store several types of databases belonging to specific archives of the Sistema Informativo Socio Sanitario (SISS). Among others, there are archives of infectious diseases, vaccinations, rehabilitative assistance, pharmacological assistance, hospital assistance, and so on so forth.

All the tables are organized following the very common star structure. Such a schema presents some advantages like denormalized tables [Shin and Sanders, 2006; Sanders and Shin, 2001]. This translates to simpler and more performing queries, but also to fast aggregation thanks to specific algorithms, like grouping, which can remarkably improve the reading capabilities of a database. This comes at the cost of having an update process which can be tricky due to the introduction of some errors if strong quality checks are not well built. The star structure contains multiple fact tables which are linked all together through a linkage key. The linkage key provides uniqueness of records and thus allows immediate identification of an event. The algorithm adopted is the *deterministic* or *rules-based record linkage*. Of course, there exist alternative methods like the *probabilistic record linkage*. What kind of linkage method to employ in a given situation really depends on a variety of factors, some of which are scientific and some of which are more subjective. In a very general framework, we can affirm that for information rich scenarios, where direct identifiers are available and of good quality, deterministic methods have been recommended [Howe et al., 2007]. In these scenarios, these methods are easier to implement, easier to interpret, and more effective. In information poor scenarios, for instance where direct identifiers are unavailable, and/or data are of poor quality, probabilistic methods consistently outperform deterministic ones even if they typically require a longer

implementation time. We highlight the work of Gomatam et al. [2002] for an empirical comparison between these two methods. Anyway, beyond these broad guidelines, the final decision strongly depends on the ultimate goals of the study.

This work focuses on three tables within the Regional Healthcare System which track and store hospital admissions data as well as pharmacological and outpatient cares prescriptions. In particular, to best of our knowledge, this will be the first consistent Italian attempt to build and use all these information jointly together. In the following subsections we are going to describe the different datasets in details as well as the process which brought to their construction.

2.1.1 Building the Minimal Database

Several steps must be taken in order to build an exploitable database which gathers all the information we are requiring in just one accessible place. Querying the data warehouse is a complex procedure which undergoes different limitations, for instance the amount of follow-up years one can require or the number of patients for which one would like to download information. Beside these reasons which are more related to the interrogation operations theirself, the data acquisition has been splitted in two parts. The former was used to identify those patients who have been admitted to a given hospital with a confirmed diagnose of CHF and to collect all their hospital admissions for this cause. The latter was used to depict the care process by mean of collecting not only their hospital admissions and related information, but also all their drug prescriptions and outpatient cares.

The raw databases which have been queried directly to Lombardy Region were originally five. In this work we will work with three of them. Most of the available information in the data come from a specific informative flux which is base on the Hospital Discharge Paper (i.e. Scheda di Dimissione Ospedaliera (SDO)). This is the tool which collects all the information associated with all the admission events occurred in public and private hospitals located in the region. The SDO was initially thought as a pure administrative tool with relative purposes. In Table 2.1, we show

some of the fields which are typically recorded in the SDOs. In the next subsections, we will reprise them in the context of the databases variables description.

Field	Description
Gender	Patient's gender
Age	Patient's age
Group	Primary diagnose group
Comorbidity	Flag if any comorbidity is diagnosed within the hospital admission

TABLE 2.1: Examples of fields recorded in the SDO.

Thanks to its valuable and rich information it is now a fundamental implement on which many activities are based. For instance, the care program activity, hospital assistance monitoring and different levels of clinical-epidemiological analyses. The type of information which we can find in the SDO are various. Among others, we find some registry information, hospitalization characteristics like which hospital and diagnose are associated with the event, what type of hospitalization, date of admission, clinical characteristic of the patient thus leading to specific diagnostic procedures. Beside the information inside the SDOs, we managed two more data flows related to drug prescriptions coming directly from the hundreds of pharmacies distributed across the region as well as the outpatient cares taken by patients during their care process.

As we can imagine, the process of collecting and storing all these information is amazingly complex. To have an idea of the size of these data we report some descriptives of the original databases:

Hospital Admissions Database:

- number of records: 2,622,802;
- number of variables: 165;

- time window: 2000-2012;
- size: 6,6 GB.

Drug Prescriptions Database:

- number of records: 35,858,388;
- number of variables: 51;
- time window: 2006-2012;
- size: 26 GB.

Outpatient Cares Database:

- number of records: 128,777,598;
- number of variables: 70;
- time window: 2000-2012;
- size: 122 GB.

The total number of records is more than 167 millions which are generated by more than 370 thousands patients.

Defining the selection criteria for patients who are affected by a given pathology is very challenging due to the high risk of introducing strong biases in the entire analysis process. Also, the definition of what is an heart failure is not intuitive and requires some workarounds. All the assumptions and the selection criteria adopted are explained in the methodological work by Mazzali et al. [2016]. To clarify some of the key points, we sum up two fundamental concepts below. The former is related to the definition of what is an hospitalization caused by CHF and the latter to the definition of what is a CHF event and an incident event. Moreover, we also report the flow chart of the quering process as show in Mazzali et al. [2016] in Figure 2.1

There are many studies which tried to define a set of criteria related to the CHF pathology and they are all based on the information collected in the SDO

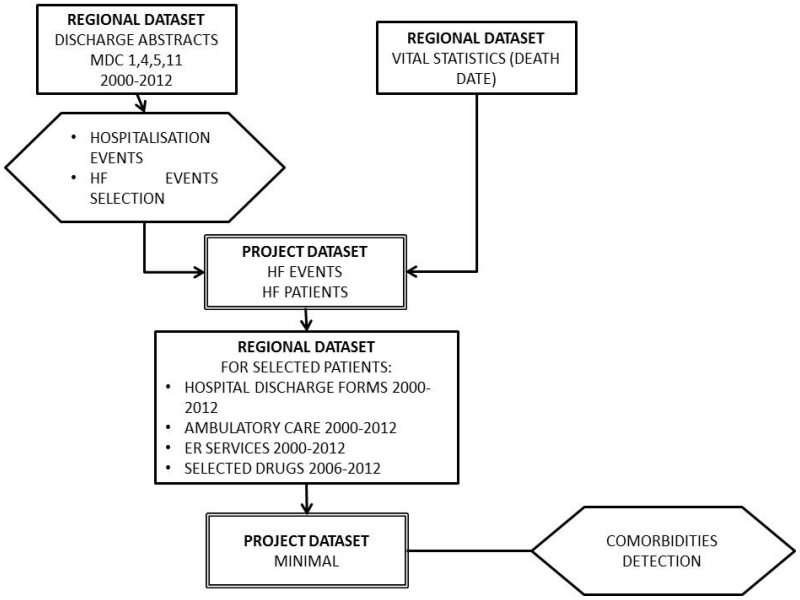

FIGURE 2.1: Data processing steps to build the project dataset as reported in Mazzali et al. [2016]

which every single hospital fills in and for each patient who has been admitted into that structure [Goff et al., 2000; Lee et al., 2005; Saczynski et al., 2012; Schultz et al., 2013; Zarrinkoub et al., 2013]. These type of information are coded through the International Classification of Diseases - Clinical Modification (ICD-CM) World Health Organization [2015]. In particular, these data use the 9^{th} revision, so ICD9-CM, with the Italian version dated 1997 Ministero della Salute [1997]. The Agency of Healthcare Research and Quality (AHRQ) proposed an indicator to evaluate intra-hospital mortality for heart failure [AHRQ, 2015]. The Centers for Medicare and Medicaid Services (CMS) developed a different model with aggregated diagnosis codes into Hierarchical Condition Categories (HCC) [Pope et al., 2011].

The data warehouse which collects information through the SDOs stores all the occurred events in the whole region. It often happens that a given patient is admitted into a hospital and then transferred into a different structure in order to receive the

best cares. For instance, a given structure may provide specific rehabilitation cares. For these cases, the database records two separated events which actually might refer to a unique care event. This happens because the databases are built upon single events, but as we have seen, this is not the optimal structure. Hence, the very first restructuring operation upon data had the goal to merge all the subsequent hospital admissions for a given patient into a single record. The rational was simply to collapse all the records which have no delay in between. That is, if a subsequent readmission occurred the same day of the previous discharge then these two records have been merged together. Moreover, this unique event is appointed as a *CHF event* if and only if at least one of the two admissions had the relative AHRQ and/or CMS-HCC codes, as described in Mazzali et al. [2016].

The raw databases have undergone a strong preprocessing procedure which aimed at minimizing errors and at aggregating several sparse and very granular information. Moreover, a restructuring procedure aimed at building a unique table in which each row is well identified with respect to the original event. At this stage, the database contains 51,565,258 events which are generated by 369,389 patients. This table collects in the same place hospital admissions, pharmacological and outpatient cares events.

Finally, we identified the very first admission caused by CHF called *incident event*. As pointed out earlier, the hospitalizations database covers the time window 2000 - 2012. An incident event is so if no other events occurred in the previous five years [Mazzali et al., 2013]. The final time window considered is 2006 - 2012 in order to have time consistency throughout the three databases, to respect the first admission assumption and, last, because the recording system of the pharmacological database prior 2006 was quiet different from the other two sources. This shrinkage caused a further decrease in the number of records and patients which now are 20,293,118 generated by 187,520 patients. We will go in details over the software procedures which allowed us to create the final dataset in Chapter 3.

All the aforementioned procedures have been carried out on the original unprocessed databases as they got out from the data warehouse. Besides other reasons not directly

related to this work, one of the main aim was to define and build a unique database which provided a clear identification of the event type for each row. In Subsections 2.1.2, 2.1.3 and 2.1.4 we describe the different information when we focus on a particular event like hospital admissions, drug prescriptions and outpatient cares, respectively. In Subsection 2.1.5 we provide a description of the data structure which is shared between all the databases.

2.1.2 The Hospital Admissions Database

In this subsection we describe the structure and the information which are strictly related to what is inside the SDOs and is referred to the process of admitting a patient into a given hospital within the Lombardy Region. These type of events will be treated as *guidance events*. This means that, for each hospital admission we will try to compute and link all the other information which are not coming directly from this table. This particular operation will be discussed later on, but will be fundamental for the purposes of these analyses.

Of all the more than 20 millions events, 583,345 ($\sim 3\%$) are hospital admissions. The available information is multiple. There are few variables which are shared through the data sources and are: patient's age and gender, censoring/death flag, closing date/date of death. All the patients are labelled into four different groups which are based on the diagnose which caused the a hospital admission. The interested reader can find more about patients' regrouping in Mazzali et al. [2016].

In Tables 2.2 and 2.3, we provide a description of all the original and computed variables, respectively, as they appear in the final version of the dataset, that is after all the cleaning and preprocessing operations. Column "Variable Name" represents the name of the variable inside the databases, column "Description" provides a brief explanation of the variable, column "Class" identifies the variable type as the software reads the datasets. The type "int" refers to an integer variable, "char" to a string, "date" to the date format as read by `R`, and "difftime" to the difference between two dates in the "date" format.

Variable Name	Description	Class
COD_REG	Original patient's unique identification ID	int
gender	Patient's gender	char
age	Patient's age	int
group	Primary diagnose group	char
ASL_RESIDENZA	Identification code of Italian Azienda Sanitaria Locale (ASL) (Local Health Board)	int
IDosp	Hospital identification ID	char
labelOUT	Patient's censoring/death flag	char
dateOUT	Either patient's censoring date/date of death	date
dateADM	Admission date	date
year_discharge	Year of discharge	int
LOS	Length of Stay in days	difftime
riab	Binary flag which marks if admission is in rehabilitation	int
ti	Binary flag which marks if admission is in intensive therapy	int
cardiochir	Binary flag which marks if patient went through heart surgery	int
ICD	Binary flag which marks if patient has received an Implantable Cardioverter Defribillator	int
CABG	Binary flag which marks if patient went through a Coronary Artery Bypass Surgery	int
charlson	Charlson comorbidity scores[1]	int
metastatic	Binary flag which marks the presence of metastasis as a comorbidity	int
chf	Binary flag which marks the presence of CHF as a comorbidity	int

[1]For a detailed description of the scores, see the works of Charlson et al. [1987] and Gagne et al. [2011]

`dementia`	Binary flag which marks the presence of dementia as a comorbidity	int
`renal`	Binary flag which marks the presence of renal related issues as a comorbidity	int
`wtloss`	Binary flag which marks the presence of weight loss as a comorbidity	int
`alcohol`	Binary flag which marks the presence of CHF as a comorbidity	int
`chf`	Binary flag which marks the presence of alcoholic issues as a comorbidity	int
`hemiplegia`	Binary flag which marks the presence of hemiplegia as a comorbidity	int
`tumor`	Binary flag which marks the presence of tumors as a comorbidity	int
`arrhythmia`	Binary flag which marks the presence of arrhythmia as a comorbidity	int
`pulmonarydz`	Binary flag which marks the presence of one or more pulmonary diseases as a comorbidity	int
`coagulopathy`	Binary flag which marks the presence of coagulopathy as a comorbidity	int
`compdiabetes`	Binary flag which marks the presence of diabetes as a comorbidity	int
`anemia`	Binary flag which marks the presence of anemia as a comorbidity	int
`electrolytes`	Binary flag which marks the presence of electrolytes related issues as a comorbidity	int
`liver`	Binary flag which marks the presence of liver issues as a comorbidity	int
`pvd`	Binary flag which marks the presence of peripheral vascular disease as a comorbidity	int

psychosis	Binary flag which marks the presence of psychosis as a comorbidity	int
pulmcirc	Binary flag which marks the presence of pulmonary circulation issues as a comorbidity	int
hivaids	Binary flag which marks the presence of HIV/AIDS as a comorbidity	int
hypertension	Binary flag which marks the presence of hypertension as a comorbidity	int

TABLE 2.2: Hospital Admission Database original variables.

Variable Name	Description	Class
ID	Patient's unique identification ID after preprocessing	int
adm_number	Number of hospital admissions per patient	int
labelOUT.2	Patient's specific status at the end of the study	char
dateDISCHARGE	Discharge date	date
exposure	Time in days since the patient had his/her first record in the database	difftime
timetoREADM	Time in days to the next event	difftime
timeADMtoOUT	Time in days between admission and the end of the study	difftime
timeDISCHARGEtoOUT	Time in days between discharge and the end of the study	difftime
DEATH_intraH	Binary flag which marks if death occurred inside the hospital	int
DEATH	Binary flag of death	int
n_com	Total number of comorbidities flagged	int
n_pro	Total number of surgical procedures registered	int

TABLE 2.3: Hospital Admission Database computed variables.

2.1.3 The Drug Prescriptions Database

This database deals with the pharmacological treatment undertaken by given a patient through the Anatomical Therapeutic Chemical (ATC) classification system[2]. This is by far the most common and best known coding systems for drugs. It operates at five different levels of details where the first indicates the anatomical main group and the fifth the chemical substance contained in the drug. The regional data warehouse is able to collect a huge number of information spanning through different families of drugs. In this work, we will focus only on those related to the cardiological system. In particular, we do register drugs of the following families:

- **A:** alimentary tract and metabolism;
- **B:** blood and blood forming organs;
- **C:** cardiovascular system;
- **N:** nervous system.

Whenever a patient buys a drug, the associated ATC code is added to the dataset and linked to the patient. Of all the more than 20 millions events, 11,238,019 ($\sim 55\%$) are given by drug prescriptions.

In Table 2.4, we provide a description of all the variables as they appear in the final version of the dataset, that is after all the cleaning and preprocessing operations.

Variable Name	Description	Class
`ID`	Patient's unique identification ID after preprocessing	int
`COD_REG`	Original patient's unique identification ID	int

[2]For further information regarding ATC codes, refer to the following link `http://www.whocc.no/atc_ddd_index/`

qta_pharma	Drug coverage in days according to the Defined Daily Dose (DDD)	int
A10	Number of registered drugs used in diabetes	int
B01	Number of registered antithrombotic agents	int
B03	Number of registered antianemic preparations	int
C01	Number of registered cardiac therapy	int
C02	Number of registered antihypertensives	int
C03	Number of registered diuretics	int
C04	Number of registered peripheral vasodilators	int
C05	Number of registered vasoprotectives	int
C07	Number of registered beta blocking agents	int
C08	Number of registered calcium channel blockers	int
C09	Number of registered agents acting on the renin-angiotensin system	int
N02	Number of registered analgesics	int

TABLE 2.4: Drug Prescriptions Database variables.

It is worth saying that though the system is able to track all the way down to the fifth ATC level, we decided that such a level of detail could be unfeasible during the modeling phase. After several discussions with the clinicians involved in the project, we came up with the variables reported in Table 2.4 which are a cutoff at the second level.

2.1.4 The Outpatient Cares Database

The third database regards the outpatient cares in the form of outpatient appointments which the patient set after a prescription has been delivered. Similarly to the pharmacological database, this one registers all the cares a patient undertakes. The classification system tracks multiple things including main discharge diagnosis, secondary diagnoses, surgeries and diagnostic and/or therapeutic procedures. Of all

the more than 20 millions events, 8,471,754 ($\sim 42\%$) are given by outpatient cares.

Again, here we focus on the number of cardiological outpatient cares as reported in Table 2.5.

Variable Name	Description	Class
`ID`	Patient's unique identification ID after preprocessing	int
`COD_REG`	Original patient's unique identification ID	int
`qta_pa`	Total number of outpatient cares taken	int
`eco`	Number of echocardiography	int
`visita_cardio`	Number of cardiological doctor's visits	int
`visit_contr`	Number of control doctor's visits	int
`test`	Number of stress tests	int
`controllo_device`	Number of device checks (i.e. peacemaker)	int
`ecg_dinamico`	Number of dynamic electrocardiograms	int
`monit_ecg`	Number of electrocardiogram monitorings	int
`ecg`	Number of electrocardiograms	int
`riab_card`	Number of cardiological rehabilitation	int

TABLE 2.5: Outpatient Cares Database variables.

2.1.5 The Longitudinal Structure

The most important feature that the aforementioned databases share is their structure which is called *longitudinal*. Longitudinal data can be defined as data resulting from the observation of subjects (human beings, animals, organizations, societies, countries) on a number of variables (health status, employment status, arithmetic skills, financial situation) over time [Bijleveld et al., 1998]. This definition implies the notion of *repeated measures*, i.e. the observations are collected on a certain number of occasions for each statistical units. The pattern of these occasion is not defined in general so that records can occur randomly. We do speak of longitudinal data

whenever we have observed more than once or whenever the number of observation points or measurement occasions (T) is greater than one. The number of subjects may vary from one to many as well as the number of variables involved. Thus, replication over time distinguishes longitudinal research ($T > 1$) from cross-sectional research ($T = 1$) [Bijleveld et al., 1998]. In the latter, we measure just one single outcome for each individual in the study. There are plenty of advantages when using longitudinal data: for instance, they can separate cohort and age effects by being able to identify and track little changes over time within individuals (aging-effect) from differences among people in their baseline levels (cohort-effects). There are many other pros and cons in using such data and they have already been widely discussed in the literature. Let us remind that a big advantage is that administrative data with the discussed structure are population based and almost costless. This is key point since trying to collect data in the longitudinal structure without the management and hardware given by data warehouses which do that for governments/public system would be economically unfeasible. For instance, we do have to ensure that the same subjects can be measured repeatedly over the course of often many years. On the other hand, particular caution must be taken when dealing with these type of data because their quality and reliability are not alway as good as expected, due, for instance, to data imputation issues (Nguyen and Barshes [2010], Grimes [2010], Hoover et al. [2011], Gavrielov-Yusim and Friger [2013], Ieva et al. [2014], Mazzali and Duca [2015]).

There exist different ways of collecting longitudinal data. The first one is the most common and natural way of tracking information that is prospectively, thus following subjects forward in time. We say natural because if we think about a data warehouse which collects data on a daily basis regarding healthcare events, we intuitively think of that as a process which goes forward in time for each patient who is registered. If it is the case, it is also possible to collect data retrospectively, by extracting multiple measurements on each person from historical records. For sake of clarity, these historical data could have been, with a high chance, administrative data previously recorded in the prospective way.

Chapter 3

Software Development for Data Preparation

A very consistent part of this thesis regards writing codes. In this chapter, we are going to present and discuss all the steps which we have taken in order to read, manage, develop and analyze all the databases.

`R` is a powerful tool for data mining/wrangling but caution must be taken when very large datasets need to be processed. We will discuss some of the key issues related to memory management and code optimization in Section 3.1 and we will present a new `R` package specifically developed for data wrangling longitudinal datasets in Section 3.3. Before going ahead with the present chapter, let us point out that all the procedures have run always on a single-core per time. All the process of making efficient codes by addressing the different issues we are going to discuss below have been thought and built to be used on a consumer laptop and not on a cluster or in any of the parallel paradigms. All the lines of code presented share the same printing style. In particular, any line beginning with a single `#` is a comment. A double `##` represents a code output. When none of these symbols are printed out, then that line is a piece of code which has to be executed.

3.1 Memory Management and Code Optimization

One of the greatest feature of R is its interactivity and ease of use. R is a high-level scripting language which implies that we do have to face some trade-offs with performances. The main purpose of R is to make data analysis and statistics more accessible to users, but at the same time it was not specifically designed to make the life of a computer easier. There are several approaches that we may consider when describing this language and one of the most important is how it uses memory.

It is well known that every time a certain operation is requested, whether arithmetical or procedural, something happens behind the scenes and we do not have complete control on it. Specifically, R often works (i.e. almost always) with a *call-by-value* schema as opposite to a way more efficient *call-by-reference* one. To clarify this concept, consider the following few lines of code in which we define a vector of integers a and we trace its memory address after very simple operations. The memory address, represented below by a sequence of numbers and letters between [], is the memory logical position in which the vector is temporarily stored. The memory is called Random Access Memory (RAM) and it is responsible to manage all the information that have to be read or written in a very fast way.

```
# defining the integer vector a
a = 1:10
# activating R memory tracing on a
tracemem( a )
## [1] [0x7fbeeb48ee88]
# assigning a to b
b = a
# modifying an element of b
b[ 1 ] = 1
## tracemem[0x7fbeeb48ee88 -> 0x7fbeeb3b8ad8]
## tracemem][0x7fbeeb3b8ad8 -> 0x7fbeebee1948]
```

As we can see, every time we define a new object in the `R` environment, we associate a specific memory block to it. When we add a new object, `b` in this case, we can see that no other memory cells are occupied, hence `R` is passing the value of `a` to `b` by reference. This operation comes at no cost and will be fundamental as we will discuss later on in this chapter. As soon as we change even a single element of `b`, we can see that the original memory cell is modified to a new one which is then edited again to the final block in which we find the updated version of `b`. Editing `b` is done here by value which means that the program is physically (i.e. through memory cells) passing the value of `a` and then modify just one of its element. It is evident that the whole operation that we are carrying out in just one line of code is not the best we can do for several reasons. The first and most important one is that we are doing copies which consume memory. `R` does copies, which here take the name of *demand copies* or just *internal copies*, all the time, so this is something that we must be warned of especially when dealing with large datasets.

The same exact behaviour occurs even when the common `data.frame` structure is used. In the following example, we show that if we need to repeat, for whatever reason, a given operation for n times, then the internal copies come into play n times by assigning at each iteration a new memory cell. Below, we build a `data.frame` with 10 rows and 3 columns and we want to assign 0 to all the rows in the first column whose value is below 0.5. It is a very simple filter and assign procedure that we take 3 consecutive times by using the `for` loop. As we can see, for each time new memory cells are involved.

```
# defining a test data.frame object
df = data.frame( a = rnorm( 10 ), b = rnorm( 10 ), c = rnorm( 10 ) )
tracemem( df )
## [1] [0x7fe061ae7a48]
# checking for any memory variation
for ( i in 1:3 ) {
  cat ( "i =", i, "\n" )
```

```
  df[ df$a < 0.5, 1 ] = 0
}
## i = 1
## tracemem[0x7fe061ae7a48 -> 0x7fe06194b890]
## tracemem[0x7fe06194b890 -> 0x7fe0658f5ba8]
## tracemem[0x7fe0658f5ba8 -> 0x7fe0658f7688]
## i = 2
## tracemem[0x7fe0658f7688 -> 0x7fe0658fe1b0]
## tracemem[0x7fe0658fe1b0 -> 0x7fe0658fe480]
## tracemem[0x7fe0658fe480 -> 0x7fe0658f96e0]
## i = 3
## tracemem[0x7fe0658f96e0 -> 0x7fe0658f9848]
## tracemem[0x7fe0658f9848 -> 0x7fe0639ca0c8]
## tracemem[0x7fe0639ca0c8 -> 0x7fe0634af568]
```

3.1.1 Call-by-reference: the `data.table` package

The above example is totally unrealistic because it makes no sense to loop over such an operation, but it is very useful though to understand what happens when we code in `R` . The best solution that we can think of is to convert every operation to a *call-by-reference* operation. Luckily, there exists a well known package called `data.table` [Dowle et al., 2014] which addresses this particular issue and which does many more things. For detailed explanations of the available functionalities, we remind to the GitHub repository at `https://github.com/Rdatatable/data.table`. The most important feature of `data.table` is that it does not make any copy whatsoever, unless explicitly requested. To show this, we run again the previous example under the `data.table` environment. As we can see, `tracemem()` is working on the `dt`, but the original memory cell has not been changed.

```
# loading data.table package
library( data.table )
# defining a test data.table object
dt = data.table( a = rnorm( 10 ), b = rnorm( 10 ), c = rnorm( 10 ) )
tracemem( dt )
## [1] [0x7fe067613000]
# checking for any memory variation
for ( i in 1:3 ) {
  cat ( "i =", i, "\n" )
  dt[ a < 0.5, a := 0 ]
}
## i = 1
## i = 2
## i = 3
```

This is a very particular approach if we think at how the R ecosystem works. This little difference, which is actually not even visible throughout the code, is at the basis of writing efficient codes and has been extensively used throughout the present work.

3.2 A Roadmap Through Complexity

The goals of the following procedures are multiple, but they can be substantially gathered into three main steps. In the first place, we want to be able to import and manage multiple large datasets efficiently and we want to do this by exploiting a suitable data structure with correct information in it. We also want to be able to identify and work with a selection of specific events, for instance just hospitalizations. This is described in Subsection 3.2.1. In the second place, we want to completely restructure the original data such that each information related to drug prescriptions and outpatient cares is well associated to a specific hospital admission. The reason and the computational procedure are explained in Subsection 3.2.2. In the end, we

want to reshape the classic longitudinal structure into a new form specifically created to run a set of statistical models. This is briefly explained in Subsection 3.2.3 and then discussed with greater details in Section 3.3. All of these waypoints are fundamentals and necessary to safely navigate through such a complex system, though they hide different levels of computational challenges and issues.

3.2.1 The Raw Data: Import and Events Generation

The very first step consists of importing the raw data. Several initial preprocessing operations have been carried out on these data in order to achieve the structure and the minimal that we have described in Section 2.1.1. The data came as `csv` files splitted up into the seven years of the follow-up.

Each file contains the longitudinal information of all the three events: hospitalizations, drug prescriptions and outpatient cares. Besides the consistent work of taking care of errors and mismatches, the strength of these data format is their simplicity in understanding which type of event we are dealing with. For sake of clarity, in Table 3.1, we show a simplified version of the data structure in which we can identify the different events by simply looking at the variable `tipo_prest` such that 41 is an hospital admission, 30 is a drug prescription and 21 is an outpatient care. We also show the level of detail that we are able to manage by looking at variable `class_prest`. This is the variable which contains the exact information related to a given event. If we look at Table 3.1, we can see that the first row corresponds to an hospital admission. Through `class_prest`, we recognize the cause of the admission which in this case is related to a Chronic Obstructive Pulmonary Disease (COPD). The same occurs when the event is a drug prescription and an outpatient care. The second and fourth rows tell us that the patient bought an antithrombotic agents and took a respiratory exam, respectively.

row id	ID	tipo_prest	class_prest
1	1	41	0127_COPD
2	1	30	B01AC06
3	1	30	B01AC06
4	1	41	0131_Adlt_resp_fl
5	1	30	C07AB07
6	1	21	0261_-_ALTROINGENERE
7	1	30	C03DA03
8	1	21	0121_-_MEDICINAFISICAERIA
9	1	21	0021_-_CARDIOLOGIA
10	1	21	0264_-_VISITADICONTROLLO

TABLE 3.1: Simplified version of a raw data file structure.

After the importing operations have been concluded correctly, we have run some consistency checks related to data format. We noticed many incongruences on columns which potentially may affect the subsequent operations. For instance, all date type variables have been cast to character so that arithmetical operations were totally impossible. Hence, the second step has been a complete scan of all the variable classes in each file and where mismatches popped out, we have cast them back to the appropriate class. Once all the different databases have been correctly loaded in the environment, a fast binding procedure has been carried out in order to aggregate them into one single dataset of 20,293,118 rows each of which is a tracked event coded as reported in 3.1. This whole operation is very efficient and it takes only few seconds to complete despite the amount of information to be processed and aligned. In the name of labelling each object, from now on we will refer to this single dataset as `HF_DATA`.

From the `HF_DATA` structure, briefly reported in Table 3.1, we have built the three specific event-type databases which we call `SDO`, `PHARMA` and `OUTP` for hospital admissions, drug prescriptions and outpatient cares events, respectively. At this point,

we can compute all the secondary custom quantities such as the Length of Stay in a hospital, the time to the next admission, or the amount of a specific chemical substance bought, or a given cardiological outpatient care taken. For instance, all the extrapolation procedures of the lower levels of details of the ATC codes have been done at this stage.

All the previous operations had three precise aims. The first one was to generate a single longitudinal database which contained all the event types ordered by patient ID and by time of event. The second one was to split again this database, but by event's type and not by event's year (this was the original way of splitting the raw data). There is also a more practical reason and it is related to the already mentioned concept of optimization. Dealing with a single and large dataset and run several complex operations on it could be hard to manage on a single machine. It is much more convenient and efficient to work with independent tables of data. The third one was to compute the most part of custom quantities which will be then used in the modeling part.

3.2.2 Defining the Sample and Aligning Information

This is the core of the whole preprocessing part. The procedure has only two aims: the first one is to define the final sample of eligible patients who will be selected for the subsequent analyses. The second one is to align the information associated with drug prescriptions and outpatient cares to the hospital admissions events.

Before going in the details of the two operations, let us point out a key fact. In order to achieve our objectives, we need to introduce a hierarchy on our data. This means that from now on, we consider the `SDO` events as our main events to which we attach more information which come from the `PHARMA` and `OUTP` events. This has consequences on several following procedures and analyses starting from the information alignment. As we will see later on, we are now focusing more on the hospital admissions process and its evolution with time than all the rest. We are assuming that all the other information could help us in understanding and eventually

modeling this process. Hence, hereafter each hospitalization will be labelled *main event*, while both a drug prescription and an outpatient care will be labelled *secondary events*.

So, back to the first procedure, with this in mind we want to identify all the patients in the three databases who are eligible for the analysis and we want to do that in order to exploit the full potential of the administrative data. In Table 3.2, we report for each database (first column) the size in terms of the number of patients (second column), the number of events they have generated (third column) and the total number of variables in each structure (fourth column).

Type of events	n. patients	n. events	n. variables
Hospital Admissions - `SDO`	187,514[1]	583,345	46
Drug Prescriptions - `PHARMA`	155,254	11,238,019	12
Outpatient Cares - `OUTP`	153,660	8,471,754	20

TABLE 3.2: Different sizes of the `SDO`s, `PHARMA`s and `OUTP`s.

As we can see, despite the amount of recorded events are not strictly related to the number of patients. The `SDO`s outnumber the other two and this is effectively a issue since for 32,260 and 33,854 patients we would not have any information associated with `PHARMA` or `OUTP`, respectively. Our best option is to have at least one drug prescription and one outpatient care for each `SDO` patient, hence we cut out all the patients who do not belong to the intersection of the three samples. Again, this operation is carried out very efficiently by exploiting the design of `INTERSECT` operator under the SQL[2] programming language which is way faster than the related base function and which is implemented in the `data.table` package. The final sample counts 144,933 patients

[1] Six patients were in the end rejected due to sevaral inconsistencies which we could not address.

[2] For further details, we refer to the following link: `https://db.apache.org/derby/papers/Intersect-design.html`

for whom we have all the information related to the three types of events. This step defines the ultimate version of `HF_DATA` on which we will run all the next procedures.

By now, nothing has changed to the original data format. For each patient selected, we do have all his/her events whether they are `SDO`s, `PHARMA`s, or `OUTP`s. The next big jump we need to take in order to satisfy the concept that `SDO`s are the main events, is to realign `HF_DATA` such that the longitudinal structure is kept, but the information regarding the secondary events would be moved and shifted next to the main events and not below. Let us explain this a little bit further with an example. Consider a patient with 10 events: 3 are main events occurred at given points in time, the other 7 are secondary events such that 4 are given by drug prescriptions and 3 by outpatient cares. We report them in Table 3.3. We would also like to highlight that missingness is present, but is due to the fact the `Length Of Stay (LOS)` is a variable which is strictly related to `SDO`s only.

row_id	ID	tipo_prest	LOS	date_of_event	class_prest
1	1	41	10	01/01/2006	0133_Oth_low_resp
2	1	30	NA	01/15/2006	C07AB07
3	1	30	NA	01/15/2006	B01AA03
4	1	41	10	02/01/2006	0127_COPD
5	1	30	NA	02/20/2006	C03DA03
6	1	21	NA	02/25/2006	0251_-_UROLOGIA
7	1	41	20	03/01/2006	0106_Dysrhythmia
8	1	21	NA	03/23/2006	0021_-_CARDIOLOGIA
9	1	21	NA	03/28/2006	0132_-_DIALISI
10	1	30	NA	03/31/2006	C09AA01

TABLE 3.3: Example of the `HF_DATA` before the alignment procedure. From left to right, we report the row id, the patient ID, the event type, the Length of Stay in a hospital, the date of the event and the associated code for identification.

The goal of this procedure consists in rearranging the information by shrinking the database in order to have only the hospital admissions by rows and the secondary events to be seated next to them as shown in Table 3.4. For demonstration purposes, we simplified some part of information. We do not have anymore the fifth level of ATC codes for chemicals, but only the families. For instance B01 are antithrombotic agents, C03 are the diuretics, C07 are beta blocking agents, C09 are agents acting on the reninangiotensin system. The same happens for the outpatient cares where now we register only the general branches where CARDIO, DIALY and URO collect all the cardiological, dialysis and urological related cares.

row_id	ID	LOS	date	B01	C03	C07	C09	CARDIO	DIALY	URO
1	1	10	01/01/2006	1	0	1	0	0	0	0
2	1	10	02/01/2006	0	1	0	0	0	0	1
3	1	20	03/01/2006	0	0	0	1	1	1	0

TABLE 3.4: The aligned version of HF_DATA in which each row is a hospital admission. All the information related to the main event are self contained in the associated row.

As we can see, we reduce the complexity along the longitudinal dimension and in the meantime we explode it in the other direction as we show in Figure 3.1. The structure is well defined and is divided into three blocks of information. The first block reflects the original HF_DATA and corresponds to the SDOs. The second and the third blocks are instead aligned to the hospitalization rows thus increasing the knowledge around that particular event. Main events here act as attractors of all the other types of events so that the evolution of the process is now really associated with the hospitalization pattern of a given patient.

To achieve this data format, a great effort has been dedicated to the identification of the secondary events between the main ones. Doing so, allows us to know what happens between two subsequent admissions. That is, we can track if a patient follows

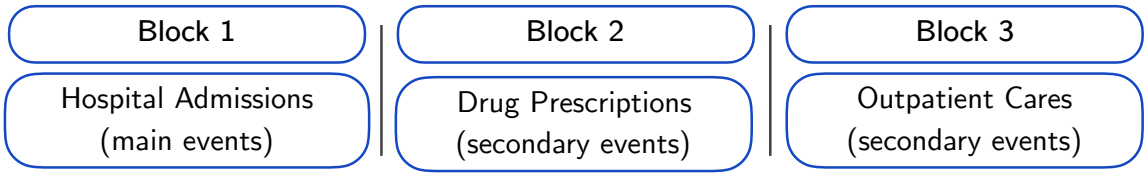

FIGURE 3.1: Schematic of the aligned version of `HF_DATA`. Now it is clear how the secondary events work as contributors to the whole set of information around a hospital admission event.

doctor's recommendations appropriately and his/her willingness to adhere to therapy is consistent throughout the period.

From a computational point of view, the alignment procedure presents some glitches which must be handled with caution. By going into details, we can analyze and split the algorithm in two sub routines.
We define a primary key in order to quickly order the database and we choose it to be patient's ID. Due to the nature of the longitudinal structure, here the primary key does not identify unique rows, but a set of multiple observations generated by the same ID. Once the groups, or the first level of aggregation, have been identified, we iteratively proceed to spot all the continuous sequences of secondary events which occur between two main events. Each row has been marked with an identification counter for each type of event to facilitate the reordering. It is worth to notice that this step comes at high computational costs. The introduction of a key, which not only stores the grouping structure, but also order the database very quickly is a an important optimization step which reduced the computational time considerably. After this first step, we have been able to compute the different indexes which track each sequence of event. We have also computed a more general index which focuses on main events in order to track their continuous sequences. All of them have been added to `HF_DATA` and we report their structure for just one patient in Table 3.5.

row_id	ID	tipo_prest	hosp.id	hosp	pharm	pa
1	1	41	1	1	NA	NA
2	1	30	1	NA	1	NA
3	1	30	1	NA	2	NA
4	1	41	2	2	NA	NA
5	1	30	2	NA	3	NA
6	1	21	2	NA	NA	1
7	1	30	2	NA	4	NA
8	1	21	2	NA	NA	2
9	1	21	2	NA	NA	3
10	1	21	2	NA	NA	4
11	1	21	2	NA	NA	5
12	1	30	2	NA	5	NA
13	1	30	2	NA	6	NA
14	1	21	2	NA	NA	6
15	1	21	2	NA	NA	7
16	1	21	2	NA	NA	8
17	1	21	2	NA	NA	9

TABLE 3.5: Custom indexing structure in `HF_DATA` for the first two main events of a given patient. We can notice how the indexes are able to track all the continuous sequences of each event type.

This table contains several information. We have `hosp.id` which tracks whether the given secondary event is related to the just occurred main event. For instance, we can see how it repeats 1 for three times. This means that the patient has had a hospital admission, given in the first row and for which we have `hosp` equals to 1, followed by two secondary events, given by in the second and third row and for which we have `pharm` equals to 1 and 2. In this case, all the secondary events are drug prescriptions

so that is why we impute a NA in `pa`. Similarly, we can look at the events related to the second hospital admission and seeing that there are 13 secondary events after the main one. Four of them are drug prescriptions while the remaining 9 are outpatient cares. Again we notice how the general index `hosp.id` marks this all sequence with a 2.

The second step of the algorithm takes care of excluding main events with a `LOS` of zero and of physically aligning the data in the structure as shown in Figure 3.1. A `LOS` of zero represents the so called *day-hospital* events for which we have no interest at this early stage of the analysis. The realignment procedure has been carried out by exploiting again the fast joins method. Specifically a triple inner join has been used to efficiently achieve the goal.

3.2.3 Reshaping the Data

All the previous operations served to create a single database whose focus is on hospitalizations. Moreover, each event is enriched with information coming from multiple sources and the structure is once again longitudinal. The very final step of our roadmap through complexity is to apply to the database a brand new reshaping procedure which will modify the data structure into a new one much more suitable for a specific set of statistical models called multi-state models. We will discuss with greater details the technicalities of the procedure in Section 3.3 where we introduce and explain the functionalities of the new `R` package we have developed. Multi-state models and related concepts will be treated extensively later on in Chapter 4. In order to fully understand the remaining part of this chapter, let us think of a multi-state model as a method to study the movement of an individual between a define set of states. States can be patient's condition or, in our case, the physical location of a patient inside or outside a hospital plus the state which identifies the death of the patient.

3.3 msmtools: Building Augmented Data in R

As we already explain, most of the effort of this work has been focused to the development of routines with special attention to efficiency and speed. While most of them, though the majority is automatic, were very data dependent, the whole process of data reshaping could have been generalized to whatever database with certain and well defined characteristics.

In this Section, we are introducing a new R package called msmtools [Grossetti, 2016] which has been developed by the author of this thesis. msmtools introduces a fast and general method for restructuring classical longitudinal datasets into an enhanced shape called *augmented*. The reasons for this are multiple as discussed in the next subsection.

3.3.1 Motivations

The development of new software is always challenging and requires to have a clear picture of what the purposes, advantages and possible glitches might be. There are three main motivations which have driven the effort.

For what concerns this work, the main question to address has been typically the following: how can we make things go faster without loosing efficiency and generality? The answer is of course not trivial and if one is asking something about reproducibility too, then these are all good reasons to start thinking about writing a consistence peace of code which tries to address them all. When working with R, the first thing which comes in mind is a package.

Besides the philosophical reasons and code ethics, the real deal here is that up to now no methods exist to restructure time to event data of the format described previously which satisfy certain requisites in order to run a multi-state model through the msm package [Jackson, 2011]. So, the main reason is the lack of a computational tool which could address this problem.

Seen the typical size of these days databases, a second fundamental motivation

is related to code efficiency and speed. `msmtools` has been so developed with a special attention to these two apects. All the functions are based on the well known `data.table` package. Non-standard-evaluation has been exploited in order to reduce the typing effort of the user [Wickham, 2014, 2015]. Lists are widely used because of their flexibility and their efficiency. We also wanted to build a tool which could take a very large dataset and process it quickly and without any computational glitch especially when using a consumer machine instead of cluster or distributed computing. As a result, `msmtools` can process around 50 millions patients in less than an hour on a single processor.

The third motivation is more related the applicability of the software. Building a general data wrangling code is very hard. `msmtools` can instead process any longitudinal dataset with certain prerequisites and output a very general augmented format which exploits all the information passed to the software. We are going to discuss this in details in Subsection 3.3.2.

`msmtools` is available for download and installation from CRAN, the Comprehensive R Archive Network. The current stable version is 1.2. To install and load it, type the following:

```
install.packages( "msmtools" )
library( msmtools )
```

`msmtools` requires R version 3.0 or higher to run. It is also compatible with Microsoft R Open 3.2.3 or higher available at `https://mran.microsoft.com/open/`. Development releases of the package are available on the GitHub repository `https://github.com/contefranz/msmtools`. In order to ensure their stability and consistency, a continuous integration service named *Travis-CI*[3] has been configured. Travis-CI specifically works to build and test the package hosted on GitHub.

[3]The service is open source and is available at: `https://travis-ci.org`

To install the latest release (possibly in development), run the following code:

```
devtools::install_github( "contefranz/msmtools" )
```

msmtools comes with three functions:

- augment(): the main function of the package. augment() processes a longitudinal dataset to produce an augmented version;
- prevplot(): a graphical Goodness of Fit (GoF) tool which plots observed and expected prevalences given a multi-state model as discussed in Subsection 4.4;
- survplot(): a second graphical GoF tool which plot fitted and empirical survival curves given a multi-state model as discussed in Subsection 4.4. It also compute the data on which plots are built.

In the next subsection, we are going to describe accurately the most important one.

3.3.2 The Function augment()

Let us recall some of the key feature of a longitudinal data structure. First of all, it can be built upon several different type of observations. For instance, bank transactions, online purchases, volcanic eruptions, hospital admissions, and so on so forth. Among the many characteristics which qualify a dataset to be labelled as longitudinal, there is one which, in this context, tops all the others and it is whether observations are time distributed or not. For instance, if we can identify a starting time and an ending time of a given observation, then we deal with data in which each observation (row) has a well defined time length that is known and exact. If we cannot identify any, then we deal with data in which observations are just points in time and for which we do not know their duration. It is now clear that a dataset regarding hospital admissions falls under the first category, while bank transactions fall under the second one. In fact, the former typically provides the dates of admission in and

discharge from a hospital, while the latter provides just the exact moment in time at which the transaction occurred. This difference is rather fundamental. `msmtools` has been developed to work exclusively with the first type of data. That means, if observations are just points in time, then the package is of no help at all.

There exist a bunch of reasons why the classic longitudinal structure is not enough when specific statistical analyses need to be run, particularly when dealing multi-state models. Because this class of models focus on finding the probability of a subject, a patient in our case, of being in a given state at a particular point in time, a first observation could be that we are not able to infer anything about this state. The data we are managing provides us information about the occurrence of specific events, like hospital admissions. Looking at single rows does not suggest us if a patient is inside a hospital or outside of it nor if he/she is dead or still alive. We clearly need a new structure in which we exploit the longitudinal format to extrapolate new *hidden* information in order to make them usable from the very beginning.

The workhorse of `msmtools` is the function `augment()` which takes a longitudinal dataset with exact starting and ending times where each row represents an observation of a given event of interest and reshapes it to produce an augmented version where each row is a specific transition between two states. `augment()` takes several arguments: a longitudinal dataset of classes `data.table` or `data.frame`, a `data_key` which identifies the subject, an optional event counter with `n_events`, a `pattern` which provides the subject' status at the end of time, a starting and ending times as well as a censoring time/death time given by `t_start`, `t_end` and `t_cens` or `t_death`, respectively. By default, `augment()` builds a pre-defined vector of states (i.e. where a patient is at a given time). The user can pass a custom vector through the argument `state`.

To illustrate how the function works, let's consider the following example which is based on the synthetic dataset `hosp`, available with `msmtools`. `hosp` mimics the behaviour of a cohort of 10 patients. These are observed anytime they enter and get out from a hospital. Each subject has his/her own admission pattern both from a time scale point of view, and from the number of events recorded. The dataset counts

53 rows and 12 variables. For a detailed description of the data, we remind to the package vignette and to the help of the dataset itself.

So, let us consider a simplified version of hosp. We extract only the first two patients, reducing the sample to 17 rows, and 8 variables out of 12 as depicted in Table 3.6. These data formats are very common when dealing with observational studies, or with chronic disease monitoring and with hospital admissions recording. In general, they are a well established system where to collect information in.

row	ID	adm_number	gender	age	label_2	dateIN	dateOUT	dateCENS
1:	1	1	F	83	dead	2008-11-30	2008-12-12	2011-04-28
2:	1	2	F	83	dead	2009-01-26	2009-02-16	2011-04-28
3:	1	3	F	83	dead	2009-05-13	2009-05-15	2011-04-28
4:	1	4	F	83	dead	2009-05-20	2009-05-25	2011-04-28
5:	1	5	F	83	dead	2009-06-12	2009-06-16	2011-04-28
6:	1	6	F	83	dead	2009-06-20	2009-06-25	2011-04-28
7:	1	7	F	83	dead	2009-07-17	2009-07-22	2011-04-28
8:	1	8	F	84	dead	2010-04-15	2010-04-20	2011-04-28
9:	1	9	F	84	dead	2010-10-11	2010-10-14	2011-04-28
10:	1	10	F	85	dead	2011-01-14	2011-01-17	2011-04-28
11:	1	11	F	85	dead	2011-04-27	2011-04-28	2011-04-28
12:	2	1	F	99	alive	2007-09-17	2007-09-27	2012-12-31
13:	2	2	F	100	alive	2009-04-09	2009-04-17	2012-12-31
14:	2	3	F	103	alive	2012-04-16	2012-04-20	2012-12-31
15:	2	4	F	103	alive	2012-04-24	2012-05-19	2012-12-31
16:	2	5	F	103	alive	2012-05-20	2012-05-25	2012-12-31
17:	2	6	F	103	alive	2012-08-19	2012-08-21	2012-12-31

TABLE 3.6: Longitudinal structure for the first two patients of dataset hosp with the following quantities: ID is the subject, adm_number is a progressive event counter, gender is the patient's gender, age is the patient's age in years, label_2 is the patient's condition at the end of the study or at his/her last observation, dateIN, dateOUT and dateCENS are the admission, discharge and censoring / death times respectively.

A call to `augment()` would look like this:

```
hosp_augmented = augment( data = hosp, data_key = subj,
                          n_events = adm_number,
                          pattern = label_2,
                          state = list("IN", "OUT", "DEAD"),
                          t_start = dateIN, t_end = dateOUT,
                          t_cens = dateCENS )
```

The augmented data are reported in Table 3.7. Despite the fact that not the same variables have been reported because of layout concerns, two things come up at first sight. In the first place, the number of rows is more than doubled. We now have 35 observations against the initial 17. In the second place, new variables have been created by the function and we report their description below:

- `status`: a status flag which gives the patient's condition at a given point in time. `augment()` automatically and quickly checks whether argument `pattern` has 2 or 3 unique values and computes the correct structure of a given subject. The variable is cast as character;
- `status_num`: the corresponding integer version of `status`;
- `n_status`: a mix of `status` and `n_events` cast as character. `n_status` comes into play when a model on the progression of the process is intended;
- `augmented`: the new timing variable for the process when looking at transitions. If `t_augmented` is missing, then `augment()` creates `augmented` by default. The function looks directly to `t_start` and `t_end` to build it and thus it inherits their class. In particular, if `t_start` is a date format, then `augment()` computes a new variable cast as integer and names it `augmented_int`. If `t_start` is a *difftime* format, then `augment()` computes a new variable as numeric and names it `augmented_num`;

row	ID	adm_number	gender	age	label_2	augmented	status	n_status
1:	1	1	F	83	dead	2008-11-30	IN	1 IN
2:	1	1	F	83	dead	2008-12-12	OUT	1 OUT
3:	1	2	F	83	dead	2009-01-26	IN	2 IN
4:	1	2	F	83	dead	2009-02-16	OUT	2 OUT
5:	1	3	F	83	dead	2009-05-13	IN	3 IN
6:	1	3	F	83	dead	2009-05-15	OUT	3 OUT
7:	1	4	F	83	dead	2009-05-20	IN	4 IN
8:	1	4	F	83	dead	2009-05-25	OUT	4 OUT
9:	1	5	F	83	dead	2009-06-12	IN	5 IN
10:	1	5	F	83	dead	2009-06-16	OUT	5 OUT
11:	1	6	F	83	dead	2009-06-20	IN	6 IN
12:	1	6	F	83	dead	2009-06-25	OUT	6 OUT
13:	1	7	F	83	dead	2009-07-17	IN	7 IN
14:	1	7	F	83	dead	2009-07-22	OUT	7 OUT
15:	1	8	F	84	dead	2010-04-15	IN	8 IN
16:	1	8	F	84	dead	2010-04-20	OUT	8 OUT
17:	1	9	F	84	dead	2010-10-11	IN	9 IN
18:	1	9	F	84	dead	2010-10-14	OUT	9 OUT
19:	1	10	F	85	dead	2011-01-14	IN	10 IN
20:	1	10	F	85	dead	2011-01-17	OUT	10 OUT
21:	1	11	F	85	dead	2011-04-27	IN	11 IN
22:	1	11	F	85	dead	2011-04-28	DEAD	DEAD
23:	2	1	F	99	alive	2007-09-17	IN	1 IN
24:	2	1	F	99	alive	2007-09-27	OUT	1 OUT
25:	2	2	F	100	alive	2009-04-09	IN	2 IN
26:	2	2	F	100	alive	2009-04-17	OUT	2 OUT
27:	2	3	F	103	alive	2012-04-16	IN	3 IN
28:	2	3	F	103	alive	2012-04-20	OUT	3 OUT
29:	2	4	F	103	alive	2012-04-24	IN	4 IN
30:	2	4	F	103	alive	2012-05-19	OUT	4 OUT
31:	2	5	F	103	alive	2012-05-20	IN	5 IN
32:	2	5	F	103	alive	2012-05-25	OUT	5 OUT
33:	2	6	F	103	alive	2012-08-19	IN	6 IN

34:	2	6	F	103	alive	2012-08-21	OUT	6 OUT
35:	2	6	F	103	alive	2012-08-21	OUT	6 OUT

TABLE 3.7: Augmented structure for the first two patients of dataset `hosp`. New variables added: `augmented` is the new time variable of the process, `status` is the patient' status flag for a given transition, `n_status` is a mix of `adm_number` and `status`.

As we can see, now for each row we have the patient' state occupied at a specific transition. In this example, we passed to `state` the default argument which translates in having three possible states: if a patient is inside a hospital, then a status IN is associated; if a patient is outside a hospital, then a status OUT is associated and when a patient dies, then a status DEAD is associated.

Given the complexity and the size of the data, which can be both very high in principle, building a subject specific status flag marking his/her condition at a given time point, could be tricky and computationally intensive. At the end of the study, so at the censoring time, a subject, in general, can be alive, dead inside a given transition if death occurs within `t_start` and `t_end`, or outside a given transition if death occurs otherwise. After n events, the corresponding flag sequence is given by $2n + 1$ for subjects who are alive and dead outside a transition, while it is just $2n$ for subjects who died inside of it. Let us consider an individual with three events. His/her status combinations will be as follows:

ALIVE: IN - OUT | IN - OUT | IN - OUT | OUT[4];

DEAD OUT: IN - OUT | IN - OUT | IN - OUT | DEAD;

DEAD IN: IN - OUT | IN - OUT | IN - DEAD.

[4]We fictitiously indicate the last state in which we see a subject.

augment() processes each patient's condition as an independent object, in this a case a list. It efficiently stores the necessary memory for them and then it fills each chunk iteratively.

3.3.3 Further Notes on augment()

The main and only aim of augment() is data wrangling. When dealing with complex structures like longitudinal data, it is really important to introduce some rules which help the user to not fail when using the function. Here we discuss some of these rules which are mandatory in order to get a correct workflow with augment(). Moreover, we will describe some of the under the hood procedures which the function runs behind the scenes.

There are some arguments which are fundamental. They are pattern and state. pattern must contain the condition of a given subject at the end of the study. That is, how the subject is found at the censoring time. Because this peculiar structure is very common when dealing with hospital admissions, the algorithm of augment() takes this framework as a reference. So, what does this mean? pattern can be either an integer, a factor or a character. Suppose we have it as an integer. augment() accepts only a pattern variable with 2 or 3 unique values (i.e. running length(unique(pattern)) must return 2 or 3). Now, suppose we provide a variable with 3 unique values. They must be 0, 1, and 2, nothing different than that. Let us explain this concept furthermore below:

Case 1. integer:

- pattern = 0: subject is alive at the censoring time;
- pattern = 1: subject is dead during a transition;
- pattern = 2: subject is dead out of a transition.

Case 2. factor:

- `pattern = alive` this is the first level of the factor and corresponds to `pattern = 0` when integer;
- `pattern = dead in`: this is the second level of the factor and corresponds to `pattern = 1` when integer;
- `pattern = dead out`: this is the third level of the factor and corresponds to `pattern = 2` when integer.

Case 3. character:

- the unique values must be in alphabetical order to resemble the pattern of the integer and factor cases.

In case one passes to `pattern` a variable which contains only 2 unique values, `augment()` automatically detects if the unit has an absorbing state occurred inside or outside a given transition. From version 1.2 and higher, this is not anymore a computational issue, but we suggest to always provide 3 unique values in order to exploit the most efficient part of the code. Everything else different from what described above will inevitably produce wrong behaviour of `augment()` with incorrect results.

The second important argument is `state`. This is passed as a list and contains the status flags which will be used to compute all the status variables for the process. The length of `state` is 3, no less, no more and comes with a default given by: `state = list( "IN", "OUT" , "DEAD" )`. The order is important here too. The status flags must be passed such that the first one (IN) represents the first state (i.e., being inside a hospital), the second one (OUT) represents the second state (i.e., being outside a hospital), and the third one (DEAD) represents the absorbing state (i.e., being dead inside or outside a hospital). Again, this is typical of hospital admissions data, where a patient can enter a hospital, can be discharged from it, or can die. As we have already see, the DEAD status is reached no matter if the subject has died inside or

outside a transition (i.e. in our case, inside or outside the hospital). One may need a higher level of complexity when specifying the states of a subject.

`augment()` by default takes a very simple status structure given by 3 different values specified in argument `state`. In general, this is enough to define a multi-state model, but one may require a more complex structure. Let us consider again the dataset `hosp` for the 3rd, 4th, 5th, and 6th patients with the structure reported in Table 3.8.

row	subj	adm_number	rehab	it	rehab_it	dateIN	dateOUT	dateCENS
1:	3	1	0	0	df	2012-09-18	2012-09-27	2012-12-31
2:	3	2	0	1	it	2012-11-28	2012-12-15	2012-12-31
3:	3	3	1	0	rehab	2012-12-18	2012-12-28	2012-12-31
4:	4	1	0	0	df	2008-08-13	2008-09-20	2012-12-31
5:	4	2	0	0	df	2012-03-18	2012-03-19	2012-12-31
6:	4	3	0	1	it	2012-07-02	2012-07-20	2012-12-31
7:	5	1	0	0	df	2006-02-09	2006-02-25	2008-04-16
8:	6	1	0	0	df	2009-03-05	2009-03-16	2010-12-19
9:	6	2	0	0	df	2009-07-06	2009-07-20	2010-12-19
10:	6	3	0	0	df	2010-11-17	2010-11-23	2010-12-19
11:	6	4	0	0	df	2010-12-05	2010-12-19	2010-12-19

TABLE 3.8: Data for 3rd, 4th, 5th and 6th patient in the dataset `hosp`. Beside the already described variables there are: `rehab` and `it` which mark if the admission is in rehabilitation or in intensive therapy units, respectively. `rehab_it` is a combination of the first two and provides both the information in just one place.

As we can see, we have two variables which take into account the type of hospital admission. `rehab` marks a rehabilitation admission while `it` marks an intensive therapy one. They are both binary and integer variables, so one can compose them to get something which is informative and, at the same time, usable in the context of "making a status". We then created the variable `rehab_it` which marks all the

information in one place and it is a character. We can pass `rehab_it` to the argument `more_status` to tell `augment()` to add these information into a new structure. Now, it is important to remember that `augment()` introduces some rules when you require to compute a more complex status structure. As we can see from the dataset, many values of `rehab_it` are set to *df*. This stands for "default" and when `augment()` finds it, it just computes the default status we already passed to argument `state` (i.e. in this case, it can be `'IN'`, `'OUT'`, or `'DEAD'`). The argument `more_status` always looks for the value *df*, hence whenever we need to specify a default transition, we also need to be sure to label it with this value.
So, if we run the following code, we build an augmented dataset with some more information regarding status structure:

```
hosp_complex = augment( data = hosp, data_key = subj,
                        n_events = adm_number,
                        pattern = label_2,
                        state = list( "IN", "OUT", "DEAD" ),
                        t_start = dateIN, t_end = dateOUT,
                        t_cens = dateCENS,
                        more_status = rehab_it )
```

In Table 3.9, we report the augmented data when the argument `more_status` is passed to `augment()`. The function computes new variables as follows:

- `status_exp`: is the direct expansion of status and the variable you passed to `more_status`, which in this case is `rehab_it`. The function composes them by pasting a '_' in between;
- `status_exp_num`: the corresponding integer version of `status_exp`;
- `n_status_exp`: similar to what has been done before, `augment()` mixes information coming from the current expanded status and the number of admissions to provide the time evolution of the process.

row	subj	adm_number	rehab_it	augmented	status	status_exp	n_status_exp
1:	3	1	df	2012-09-18	IN	df_IN	1 df_IN
2:	3	1	df	2012-09-27	OUT	df_OUT	1 df_OUT
3:	3	2	it	2012-11-28	IN	it_IN	2 it_IN
4:	3	2	it	2012-12-15	OUT	it_OUT	2 it_OUT
5:	3	3	rehab	2012-12-18	IN	rehab_IN	3 rehab_IN
6:	3	3	rehab	2012-12-28	OUT	rehab_OUT	3 rehab_OUT
7:	3	3	rehab	2012-12-28	OUT	rehab_OUT	3 rehab_OUT
8:	4	1	df	2008-08-13	IN	df_IN	1 df_IN
9:	4	1	df	2008-09-20	OUT	df_OUT	1 df_OUT
10:	4	2	df	2012-03-18	IN	df_IN	2 df_IN
11:	4	2	df	2012-03-19	OUT	df_OUT	2 df_OUT
12:	4	3	it	2012-07-02	IN	it_IN	3 it_IN
13:	4	3	it	2012-07-20	OUT	it_OUT	3 it_OUT
14:	4	3	it	2012-07-20	OUT	it_OUT	3 it_OUT
15:	5	1	df	2006-02-09	IN	df_IN	1 df_IN
16:	5	1	df	2006-02-25	OUT	df_OUT	1 df_OUT
17:	5	1	df	2008-04-16	DEAD	DEAD	DEAD
18:	6	1	df	2009-03-05	IN	df_IN	1 df_IN
19:	6	1	df	2009-03-16	OUT	df_OUT	1 df_OUT
20:	6	2	df	2009-07-06	IN	df_IN	2 df_IN
21:	6	2	df	2009-07-20	OUT	df_OUT	2 df_OUT
22:	6	3	df	2010-11-17	IN	df_IN	3 df_IN
23:	6	3	df	2010-11-23	OUT	df_OUT	3 df_OUT
24:	6	4	df	2010-12-05	IN	df_IN	4 df_IN
25:	6	4	df	2010-12-19	DEAD	DEAD	DEAD

TABLE 3.9: Augmented data when a complex structure is required. New variables are: `status_exp` which mixes the information coming from status and argument `more_status`. `n_status_exp` mimics the behaviour of `n_status`.

`augment()` takes care of other two things before reshaping the data. In the first place, it checks whether the hospital admissions process is monotonic. In the second place, it scans several function arguments looking for the presence of missing data. In order to get the data processed, a monotonic increasing process needs to be ensured. `augment()` checks this both in case `n_events` is missing or not. The data are efficiently ordered through the `setkey()` function with `data_key` set as the primary key and `t_start` as the secondary key. Then it checks the monotonicity of `n_events` and if it fails, it stops with error and returns the subjects for whom the condition is not met. If `n_events` is missing, then `augment()` internally computes the progression number with the name `n_events` and runs the same procedure. This whole procedure is carried out with very little memory overheads thus it is very fast and efficient despite the dataset is large. The search for missing data is now optional and by default it is not run anymore. In fact, from version 1.2 the argument `check_NA` has been introduced and set to `FALSE`. This is due because `augment()` has been developed just for restructuring data. This procedure is computationally intensive and could cause strong memory overheads. When dealing with very highly dimensional datasets, this becomes mostly unfeasible. We then suggest to perform all these types of checks before running `augment()`. If one really wants to run this procedure, can set `check_NA = TRUE` and the detection will be performed over `data_key`, `n_events`, `pattern`, `t_start` and `t_end`. Beware, that no missing imputation or deletion is carried out. If any missing value is found, then `augment()` stops with error asking to fix the problem.

As already said in Subsection 3.3.1, `msmtools` has been developed to be fast and efficient. In Figure 3.2, we show the results of two simulation runs of the function `augment()`. The procedure has been carried out on the GIGAT cluster available at MOX (Department of Mathematics – Politecnico di Milano) which is made of 160 cores splitted into 5 nodes with a Xeon E4610-v2 CPU and 1.2 TB of RAM. The simulations used just a single core. Each point in the plot represents a new simulated longitudinal dataset which needs to be converted into the augmented format. In

bottom the x-axis we report the number of input rows, which correspond to our hospital admission events. In the left y-axis, we can read the running time of the reshaping procedure. In the top x-axis, we can read the related number of output rows obtained after the dataset has been converted into the augmented format. Finally, in the right y-axis, we can see the number of patients for each simulated dataset. So, what we like to highlight is that `augment()` is able to process almost the whole Italian population in about an hour. Also, the computational complexity is quasi-linear, which is a remarkable result.

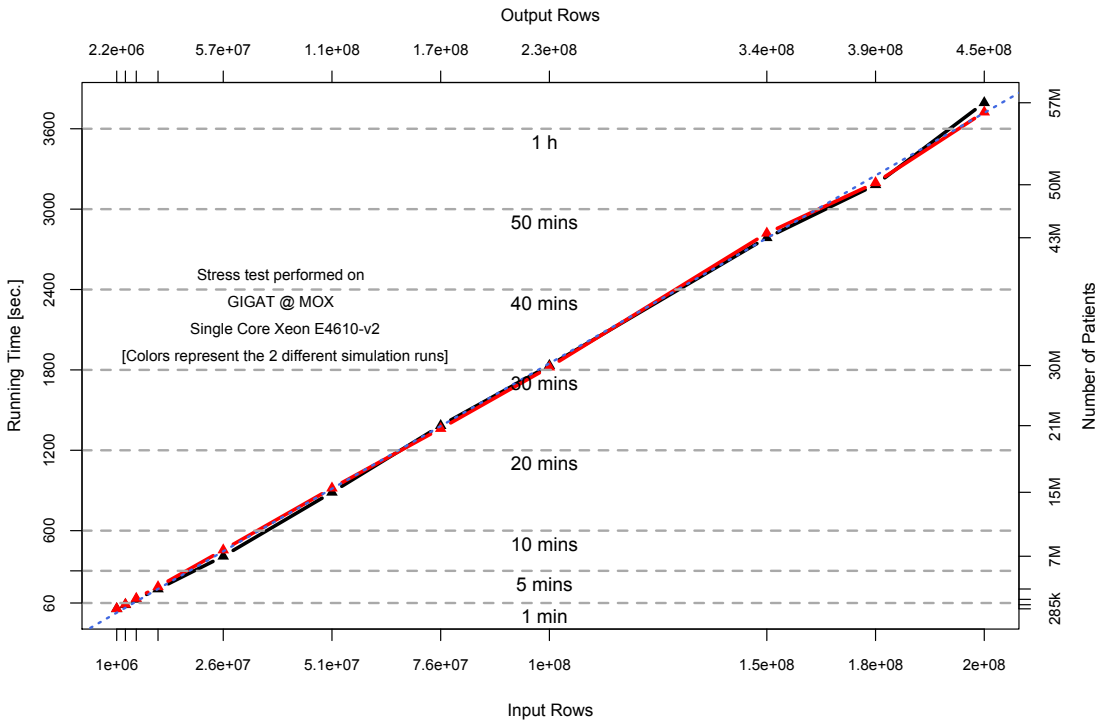

FIGURE 3.2: GIGAT simulations obtained with `augment()` for an increasing number of patients.

3.3.4 In Development

What has been described above is available as a stable release. There are though some parts of the package which are still under development. Some of them are merely related to the bug fixing process, while some others are brand new functions which introduce more features into `msmtools`.

One of the biggest issue when dealing with multi-state models is the occurrence of the same event, hence with the same status, in the exact same point in time. This is strictly related to the definition of the time scale of the process under study, which in our case is given by days. For instance, it is not possible to have two admissions on the same day for the same patient. This would cause an error and the model would simply not run. To overcome this problem, we can either delete all the events of the patients who showed the issue or we can perform little time transformations only on the critical rows (single events). We call this second approach the *jittering algorithm*. The two possible solutions will be wrapped up into a `R` function which allows the user to choose the rough solution or the more elegant one given the fact that the idea behind is very different. While the first approach is very simplistic and has the only aim at deleting patients, the jittering algorithm has instead the purpose of retaining the highest number of units. At the present time, the algorithm is in alpha-testing, so for this work we will adopt the rough procedure which deletes around 30% of patients. Up to now the jittering algorithm is able to preserve around 90% of the patients. This is mainly due to erratic behaviours of the patient's structures which are therefore hard to recognize and fix. Being able to intercept these peculiar patterns in order to minimize the loss of patients is at the core objective of the ongoing development.

Despite the very fast processing time of the current `msmtools` version, another great improvement in speed might be gained by using `C` code for the most demanding tasks. For instance, we know that there are some bottlenecks in the `augment()` function like the pattern recognition based on argument `pattern` when this is of length 2 or when the algorithm looks for defining the dimension of targets for the reshaping. There has been a lot of improvement both in speed and memory optimization, but the first one is

a major problem which can only be bypassed using a compiled programming language like C is. To fulfill this goal, we use the Rcpp package [Eddelbuettel, 2013; Eddelbuettel et al., 2011] which makes really easier the integration between C++ code with existing R one. In particular, all the implementation of loops and some consistency checks will moved to compiled code.

Another desirable achievement would be the reimplementation of all the loops using multicore computations. Though this makes no sense for small datasets, and in fact it is not even recommended for our dataset because the number of events is too small, it could be very powerful when the number events goes really up high. This is the case where there are wide margins of speed improvements. This is in the very alpha-testing phase and up to now we are implementing the parallel backend using the following packages: parallel, doParallel [Analytics and Weston, 2015a], and foreach [Analytics and Weston, 2015b].

There are many other minor general improvements and among them it is worth to cite the integration of ggplot2 grammar of graphics [Wickham, 2009] in support of the plots computed by the functions survplot() and prevplot().

Chapter 4

The Statistical Models

In this chapter we discuss the set of statistical models we use to analyze the dataset we have prepared in the previous chapter. We are particularly interested in modeling the movement of a patient between a given set of states. We are also interested in estimating the risk associated with the movement and the relative survival distribution. Survival analysis and, more in general, multi-state models help answering these questions. We provide a brief description of the two approaches in Sections 4.1 and 4.2.

4.1 Survival Analysis

Survival analysis involves the consideration of the occurrence of an event and particularly, the time between a starting point and an ending point. In our case, we can use this approach to estimate the time between an admission into a hospital (starting point) and the death of a patient (ending point).

One of the most important aspect to keep in mind when dealing with this type of analysis and data is the fact that there is no assurance that all the individuals experience the event of interest (i.e. death) before the time of the study ends. This has some consequences like the impossibility to compute the full survival times for

these individuals. A situation like the one described is called *censoring* for which we discuss some details in the next subsection.

4.1.1 Censoring and Truncation

Whenever we are dealing with time to event data, there is a certain percentage of censored data. Together with the concept of censoring is the *truncation* which is typically due to study design and it is thus deliberate.

Right censoring: occurs when a subject leaves the study or the study just ends before an event occurs. For example, consider some patients in a clinical trial which studies the effect of a given treatment for some cardiac pathology with a closing time of 5 years. All the patients who did not show up any expected sign after the treatment are labelled as censored. Typically, the censoring is independent of the survival time. There two types of independent censoring:

- Type I: occurs when the dropout is completely random and/or the time of the end of the study is fixed and thus no event of interest has occurred before it;
- Type II: occurs when the study ends with a set of subjects experience a fixed number of events.

Left censoring: is when the event of interest has already occurred before the actual enrollment in the study. In this work we do not consider this case since it just does not happen.

Right truncation: occurs when the entire study population has already experienced the event of interest.

Left truncation: occurs when the subjects have been at risk before entering the study. For example, consider the case of life insurance policy holders where the

study starts on a fixed date and the event of interest is the age at the time of death.

In this work, we assume type I right censoring. This is also called *administrative censoring* and 91,403 patients ($\sim 64\%$) registered in our database are of this kind. The hypothesis of independent right censoring holds since the censoring mechanism is due to an administrative choice and does not affect neither the disease progression nor the estimate of risk of future events [Putter et al., 2007; Andersen and Keiding, 2002].

4.1.2 Survival and Hazard Functions

The analysis of survival data requires special methods due to the particular set of data and to their characteristics. For example, data of this type are rarely normally distributed but tend to be strongly skewed with the vast majority of events occurring in early times.

In principle, survival data can be described and modeled in terms of two related probabilities: *survival* and *hazard*. The survival probability $S(t)$, also called the survival function (s.f.), is the probability that an individual survives from the origin of time to a specified future time t. $S(t)$ contains crucial summary information from time to event data and its values describe directly the survival experience of a study cohort. The hazard probability, also called the hazard function (h.f.), is usually denoted with $h(t)$ or $\lambda(t)$, the notation used in this work. $\lambda(t)$ represents the probability that an individual who is the study and thus under observation at a time t has an event at that time. In other words, it is the instantaneous risk, or event rate, for an individual who has survived to time t. This is the opposite information provided by $S(t)$ which focuses on not having an event. $\lambda(t)$ is quite important since it can shed some light on the conditional failure rates and thus it helps when we need to specify a survival model.

Even if the two functions provide different information, there is a defined relationship between them which is given by the following equation:

$$\lambda(t) = -\frac{\mathrm{d}}{\mathrm{d}t}\left[\log S(t)\right]. \tag{4.1}$$

This equation has a major consequence which is that if either $S(t)$ or $\lambda(t)$ is known, then the other one is automatically determined. In other words, either functions can be the basis of a survival analysis and their estimation methods are discussed in Subsection 4.1.3.

4.1.3 Non-parametric Estimators

In this subsection, we briefly discuss two non-parametric estimators for the s.f. and h.f. in the univariate case. $S(t)$ can be estimated from observed survival times, both censored and uncensored, using the so called Kaplan-Meier (KM) method [Kaplan and Meier, 1958], also called product-limit method. Let us consider k individuals who have events in the follow-up period at distinct times $t_1 < t_2 < \ldots < t_k$. Because events are assumed to occur independently of one another, the probabilities of surviving from one interval to the next one can be multiplied together to give the *cumulative survival probability*. Put another way, the probability of being alive at time t_j, given by $S(t_j)$, is calculated from $S(t_{j-1})$ which is the probability of being alive at the time t_{j-1}, so:

$$S(t_j) = S(t_{j-1})\left(1 - \frac{d_j}{n_j}\right), \tag{4.2}$$

where d_j is the number of events at time t_j and n_j is the number of individuals alive immediately before t_j. Here $t_0 = 0$ and $S(0) = 1$. Between two events, $S(t)$ is constant which leads to compute an estimated step function whose values change only at the exact times of the events. Every individual contributes information to this function as long as they are known to be event-free. The estimator then takes the following form for right-continuous processes:

$$\widehat{S}(t) = \prod_{t_j \leq t} \frac{(n_j - d_j)}{n_j}, \tag{4.3}$$

with an approximated estimated variance given by Greenwood's formula:

$$\widehat{Var}[\widehat{S}(t)] = \widehat{S}(t)^2 \sum_{t_j \leq t} \frac{d_j}{(n_j - d_j)}. \tag{4.4}$$

The KM estimator supports the computation of a Confidence Interval (CI). The plot of the KM survival curve is a useful inspection tool which provides a summary of the data.

Conversely to what discussed for $S(t)$, there is no immediate way to estimate $\lambda(t)$. It is instead more convenient to estimate a quantity called Cumulative Hazard Function (c.h.f.) denoted by $\Lambda(t)$ defined as follows:

$$\Lambda(t) = \int_0^t \lambda(\tau) \mathrm{d}\tau, \tag{4.5}$$

which represents the area under the hazard curve between times 0 and t. $\Lambda(t)$ has again a relation with $S(t)$ given by:

$$\Lambda(t) = -\log\left[S(t)\right], \tag{4.6}$$

but the interpretation is harder. A possible way to interpret this quantity is to think it as the number of events that would be expected for each individual by time t. A non-parametric method to estimate $\Lambda(t)$ is the Nelson-Aalen (NA) estimator [Hosmer Jr and Lemeshow, 1999] and use the result to derive an estimate of $\lambda(t)$ through a kernel smoother to the increments [Ramlau-Hansen, 1983]. The NA estimator for the cumulative hazard rate is defined as:

$$\widehat{A}(t) = \sum_{t_j \leq t} \frac{d_j}{n_j}, \tag{4.7}$$

where n_j is the number of individuals at risk just prior to time t_j. Hence, the estimator is an increasing right-continuous step function with increments d_j/n_j at observed failure times. Another estimation method is suggested in Cox [1979] and it is based on order statistics. As we will discuss later on in Subsection 4.1.6, another approach

is to assume that the survival times follow a specific probability distribution. Before tackling this problem, we need to built a statistical model which is able to operate in a multivariate framework.

Below in Subsections 4.1.4 and 4.1.5 we introduce the two families of regression models for survival data.

4.1.4 The Semi-parametric Proportional Hazard Model

The main advantage of building a statistical model is that it allows the survival to be assessed with respect to different variables simultaneously. Moreover, it is able to provide an estimate of the effect of each of the factors involved. As for the dataset under study, it is common to have several known quantities which can potentially affect the outcome. As for the linear regression model, we make use of covariates to improve the assessment of the response. Covariates may be registered and measured at the time of entry to the study or at each time step corresponding to an event. Covariates can be of different form. For example, they can be continuous (i.e. patient's age), binary (i.e. patient's gender), unordered or ordered categorical (i.e. pathology group and some performance status indicators).

There exist several Proportional Hazard (PH) models for modeling survival data. Here, we briefly discuss one of the most common: the Cox Proportional Model (CPM) [Cox, 1972]. CPM consists of a multivariate survival analysis regression model which provides a statistical assessment between the event incidence as expressed by the h.f. $\lambda(t)$ and a set of p known covariates. Mathematically, the model has the following form:

$$\lambda(t) = \lambda_{(0)}(t) \exp\left[\boldsymbol{\beta}(t)^T \mathbf{z}(t)\right], \tag{4.8}$$

where $\boldsymbol{\beta}(t)$ is the ith covariate effect vector and $\mathbf{z}(t)$ is the covariate vector. $\lambda_{(0)}(t)$ is the *baseline hazard function* which represents the value of the hazard when no covariates intervene. The semi-parametric form of this model is due to the fact that

$\lambda_{(0)}(t)$ is estimated non-parametrically through the NA estimator while the covariates effect through the Maximum Likelihood (ML) method. This version of the CPM is also know as semi-parametric CPM (spCPM). The diffuse dependence on time is the general form of the equation which will be simplified later on by introducing some constraints. In principle, every part of the CPM can vary with time.

The model is composed such that $\lambda_{(0)}(t)$ is the intercept with the covariates that act through a multiplicative effect on $\lambda(t)$. This is the assumption PH in the CPM and implies that the hazard of the event of interest in any given group is a constant multiple of the hazard in any other. In practice, the different hazard curves should be proportional and thus they should not cross each other. PH also implies that the coefficient $\exp[\beta_i]$ is expressed in the form of Hazard Ratio (HR) between two individuals whose values of the covariate $z_i(t)$ differ by one unit when all the other covariates are held constant. Because the CPM models the h.f. with equation 4.8, this is equivalent to say that:

$$\mathrm{HR}_i = \log\left[\frac{\lambda^i(t)}{\lambda^i_{(0)}(t)}\right] = \sum_i \beta_i(t) z_i(t). \tag{4.9}$$

Hence, a unit increase in the covariate $z_i(t)$ is associated with a β_i increase in the log hazard rate. In other words, if the HR > 1, then for a unitary increase in the ith covariate, the associated risk of event increases too and the survival length decreases. We speak of a positively associated covariate with the h.f..

4.1.5 The Accelerated Failure Time Model

A second method to model survival data is represented by the so called Accelerated Failure Time (AFT) models. This class of models assumes that the passage of time can be slowed down or speeded up with respect to a given set of covariates. In other words, the main focus here is the survival time of an individual which can be shortened or extended depending on the covariates. If we consider a group of individual with the covariate vector $\mathbf{z}(t) = (z_1, z_2, \ldots, z_p)$, then we can write the model using the s.f.

as follows:

$$\widetilde{S}(t) = S(\phi\, t), \tag{4.10}$$

where ϕ is an *acceleration factor* which depends on the covariates in the usual way:

$$\phi = \exp\left[(\beta_1 z_1, \beta_2 z_2, \ldots, \beta_p z_p)\right]. \tag{4.11}$$

4.1.6 The Parametric Approach

As we mentioned very briefly in Subsection 4.1.3, it is possible to estimate the h.f. with a fully-parametric approach. There exist several models under this approach many of them assume PH. In this work, we will also treat the AFT models [Kalbfleisch and Prentice, 2011; Wei, 1992] which conversely to PHs assume an accelerating (decelerating) effect of the covariate on the hazard. We will refer to these models with parametric Proportional Hazard (pPH) and parametric Accelerated Failure Time (pAFT) models which are totally similar to spCPMs from a concept and interpretation point of views. The only difference dwells of course in how we estimate $\lambda_{(0)}(t)$ and in the general form of the regression equation. In a parametric framework, the baseline hazard is assumed to follow a specific probability distribution belonging to a parametric family. Beside this, all the interpretations on the nature of the model coefficients and relations hold.

The main assumption here is is that the survival time T follows a probability distribution with density function $f(t)$ such that:

$$S(t) = \text{Prob}(T > t) = \int_t^\infty f(\tau)\mathrm{d}\tau. \tag{4.12}$$

We deal now with three functions $S(t)$, $\lambda(t)$ and $f(t)$ which are linked together through the following relation:

$$f(t) = \lambda(t)S(t), \tag{4.13}$$

and it is easy to show that:

$$\lambda(t)S(t) = -\frac{\frac{\mathrm{d}}{\mathrm{d}t}S(t)}{S(t)}S(t) = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\infty}^{t} f(\tau)\mathrm{d}\tau = f(t). \tag{4.14}$$

Equation (4.14) implies that specifying one of the three key functions specifies the other two.

The general form of a parametric survival regression model is as follows:

$$\lambda(t) = \lambda_{(0)}^{\mathrm{par}}(t, \boldsymbol{\theta})\, r(\boldsymbol{\beta}(t), \mathbf{z}(t)), \tag{4.15}$$

where $\lambda_{(0)}^{\mathrm{par}}(t, \boldsymbol{\theta})$ is the parametric baseline hazard function and $r(\boldsymbol{\beta}(t), \mathbf{z}(t))$ is a relative risk function describing the effect of the covariates. In the present work, we assume r to follow an exponential such that we can rewrite equation 4.15 as follows:

$$\lambda(t) = \lambda_{(0)}^{\mathrm{par}}(t, \boldsymbol{\theta})\, \exp\left[\boldsymbol{\beta}(t)^T \mathbf{z}(t)\right]. \tag{4.16}$$

In Subsection 4.3.1, we will discuss the implementation of such a model using a specific probability distribution.

4.2 Multi-state Models

Another approach to model survival type data is to consider the movement of an individual through different conditions or states. In this work, we focus our attention to a specific family of these models called Multi-State Models (MSM). These constitute a very smart approach to analyzing categorical longitudinal data. As we described in Chapter 3, the main event is given by a hospital admission and through `msmtools` we have increased the amount of information self contained in each event into single transition. From this point of view, a patient is seen as a subject who can move from different state representing his/her hospital admission status.

In general, in MSMs with irregular observation times, the movement between states is governed by a continuous time stochastic process. This process $(X(t), t \in [0, \tau])$

occupies at any time one of a set of finite discrete states called state space given by $\mathcal{S} = \{1, \dots R\}$ and with right continuous paths such that $X(t+) = X(t)$ [Hougaard, 1999; Castañeda and Gerritse, 2010]. t is the observation time and assumes values in $[0, \tau]$ or $[0, \tau)$ with $\tau \leq +\infty$.

Multi-state models have been successfully used in several medical applications in which the main output was stages or levels of a disease. In our case, this corresponds to register a hospital admission, discharge or death. It is common to define a MSM by its matrix of *transition intensities* $Q(t, \mathcal{F}_t)$, where $\mathcal{F}_t$ is the history, or *filtration*, of the stochastic process up to time t. In fact, an individual can move to one particular state from a starting one according to a defined set of transition intensities $q_{rs}(t)$ with $r, s \in \mathcal{S}$. These intensities, or hazards, represent the instantaneous risk an individual has of moving from state r to state s and characterize the process. In general, hazard intensities functions can be written in the following form[1]:

$$q_{rs}(t|\mathcal{F}_t) = \lim_{\delta t \to 0} \frac{\mathrm{P}\left(X(t+\delta t) = s | X(t) = r, \mathcal{F}_t\right)}{\delta t}. \tag{4.17}$$

Similarly, we can consider the *transition probabilities* matrix of being in each state at a fixed time in the future as follows:

$$p_{rs}(t_1, t_2, \mathcal{F}_{t_1}) = \mathrm{Prob}(X(t_2 + \delta t) = s \,|\, X(t_1) = r, \mathcal{F}_{t_1}). \tag{4.18}$$

Equation 4.17 defines the h.f. in the context of MSMs. This form is rather general and in the following sections we are going to discuss some of the assumptions which can be considered in order to simplify the approach both from a statistical and a computational point of view. The starting point is to use semi- and fully-parametric approaches by exploiting equations (4.8) and (4.15) for CPMs and pAFT models, respectively. For instance, we describe what is a general Markov model, a time-homogeneous Markov model and a semi-Markov model. In later sections, we provide

[1] In the context of MSMs, we use the letter q to indicate the h.f. in place of λ to be more consistent with the transition intensities matrix $Q(t)$. The meaning and interpretation remain the same of equations written in Section 4.1.

a brief appraisal of the different types of model structures available. We will also point out pros and cons of each approach and will show what type of structure we adopted for the analyses.

4.2.1 Full Markov Model

One of the most powerful assumption one can incorporate in a model is that the future evolution of the process is entirely described and thus depends only on the current state of the ongoing process. This is call *Markov property* and in terms of transition intensities implies the following:

$$q_{rs}(t, \mathbf{z}(t)) = \lim_{\delta t \to 0} \frac{\mathrm{P}\left(X(t+\delta t) = s | X(t) = r\right)}{\delta t}, \tag{4.19}$$

hence the transition intensities actually vary with time but do not depend anymore on the filtration $\mathcal{F}_t$. One of the peculiar advantages which this approach brings in, is the ability to strongly simplify the likelihood of the system by expressing it as a product of transition intensities:

$$L = \prod_{i=0}^{N-1} p_{x_i, x_{i+1}}(t_i, t_{i+1}), \tag{4.20}$$

where the index i represents the number of observed transitions up to $N-1$. The transition probability matrix $P(t)$ for a full Markov Model (fMM) satisfies the forward Kolmogorov equations [Cox and Miller, 1977]:

$$\frac{\mathrm{d}P(t_1, t)}{\mathrm{d}t} = P(t_1, t)Q(t), \tag{4.21}$$

according to an initial condition $P(t_1, t) = I$ where $P(t_1, t)$ is the matrix with (r, s) entry given by $p_{rs}(t_1, t)$. For the large majority of $Q(t)$, the Kolmogorov equations define a system of non-linear differential equations which cannot typically be solved analytically. Numerical iterations are then required in order to solve the system. For

most of the models used in this work, the algorithm of Broyden-Fletcher-Goldfarb-Shanno (BFGS) has been adopted [Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970].

Note that in this case, the observation time of each event is known and exact. Adopting this framework allows us to estimate the probability of transitioning from one state to another (i.e. admission, discharge or death), with respect to patient characteristics as well as his/her medical treatments which are all summarized in the covariate vector ($\mathbf{z}(t)$). In the present work, we assume the process to be a time-homogeneous Markov process. This implies that the future trajectory of the process depends only on the current state of the process and not on the previous history and q_{rs} is taken as constant in time. A state can be absorbing, meaning that no transition exits from this state. In our case-study, death is the absorbing state. It is possible to evaluate the probability of no events during a period and also the number and types of events. In order to do so, we must consider the transition probabilities matrix of being in each state at a fixed time in the future. Since the process is time-homogeneous, the (r, s) entry of matrix $P(t)$ depends only the time interval length. The transition probability is given by:

$$p_{rs}(t) = \text{Prob}(X(t + \delta t) = s \,|\, X(t) = r). \tag{4.22}$$

The likelihood for the model is computed from the transition matrix, using the matrix exponential algorithm because no analytical form is available for the model we are going to consider later on. Consider a patient i who had n_i transitions occurred at the exact times $(t_{i1}, \ldots t_{in_i})$ with corresponding states $(s(t_{i1}), \ldots s(t_{in_i}))$. No other transitions between the observed ones are allowed. The contribution to the likelihood for the time interval (t_j, t_{j+1}) is then given by:

$$L_{ij} = \exp\left[q_{s(t_j)s(t_j)}(t_{j+1} - t_j)\right] q_{s(t_j)s(t_j+1)}, \tag{4.23}$$

because we are assuming that a patient stays in state $s(t_j)$ throughout the time interval between t_j and t_{j+1} when he/she goes in state $s(t_{j+1})$ precisely at time t_{j+1}.

Equation (4.23) accounts for the contribution of these two states. The full likelihood L is given by the the product of all the terms L_{ij} and depends on the intensity matrix $Q(t)$ such that:

$$L = \prod_{i=1}^{K} \prod_{j=0}^{T} L_{ij}, \tag{4.24}$$

where K is the number of patients under study and T is the follow-up time.

4.2.2 Time-homogeneous Markov Models

A second strong assumption is to consider the process to be time-homogeneous, thus leading to a time-homogeneous Markov Model (thMM). In words, this means that the transition intensities are constants and independent of time t. A process of this form has therefore the further property:

$$Q(t) = Q_0, \quad \forall t \tag{4.25}$$

where Q_0 is some constant matrix. Equation (4.25) has a strong implication: the sojourn time within a given state defined as the amount of time spent in that state (i.e. the number of days a patient stays in a hospital) is exponentially distributed with a rate parameter $\sum_{s \neq r} q_{rs}$ where q_{rs} is the entry in place (r, s) of the transition matrix Q_0.

If equation (4.25) is a reasonable assumption, then the transition probabilities only depend on the length of the interval between t_1 and t_2 and not on t_1 itself. This allows us to rewrite equation (4.21) in the following form:

$$\frac{\mathrm{d}P(t_1, t)}{\mathrm{d}t} = P(t_1, t)Q_0, \tag{4.26}$$

according to an initial condition $P(0) = I$. The solution of this equation is:

$$P(t) = \exp\left(tQ_0\right), \tag{4.27}$$

which is typically computed to matrix exponential algorithm. Parameters estimation is carried out through ML numerical optimization using BFGS.

4.2.3 Semi-Markov Models

A less restrictive, and possibly more realistic, assumption is to consider the sojourn times dependence on the history of the process only through the present state and the time since entry of that state. The resulting MSM forms a sequence of embedded Markov models, called a *Markov renewal model* [Dabrowska et al., 1994; Prentice et al., 1981; Gill, 1980; Lagakos et al., 1978], or also a semi-Markov Model (sMM). In the present thesis, we will make use of the above mentioned types of approaches.

4.2.4 Structure of Multi-state Models

When modeling the movement of a subject throughout a set of states, things can become oddly complicated. It is therefore important to define a structure which the MSM has to follow. We have already introduced the transition matrix $Q(t)$ in Section 4.2 which manages which transitions are permitted. We now introduce the concepts of *transient* and *absorbing* state. The former defines a state which a subject can reach at some point in time, but which can leave too. The latter defines a state for which a subject can only enter once. By using these two types of states, it is possible to build several different schemas which help to understand the undergoing process.

Unidirectional and Progressive Models

The simplest structure we can think of consists of a single chain of states $\mathcal{S} = \{1, \ldots, R\}$ where the progression from the initial state is always sequential till the final state. This is called *unidirectional model* and in Figure 4.1 we show a representation of it in which the last state R is an absorbing state which is indicated by A.
A simple survival model in which the outcome is given by two states, alive or dead, is an example of unidirectional model.

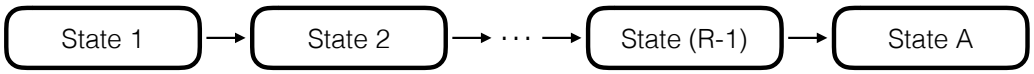

FIGURE 4.1: Scheme of a unidirectional MSM.

These types of models belong to a wider class of models called *progressive*. Progressive models are made up by two or more chains and the progression is again sequential. From the initial state, a subject can move to one of the available chains and then it continues the process till the absorbing state in the exact same way as in unidirectional models. We show a scheme of progressive model in Figure 4.2.

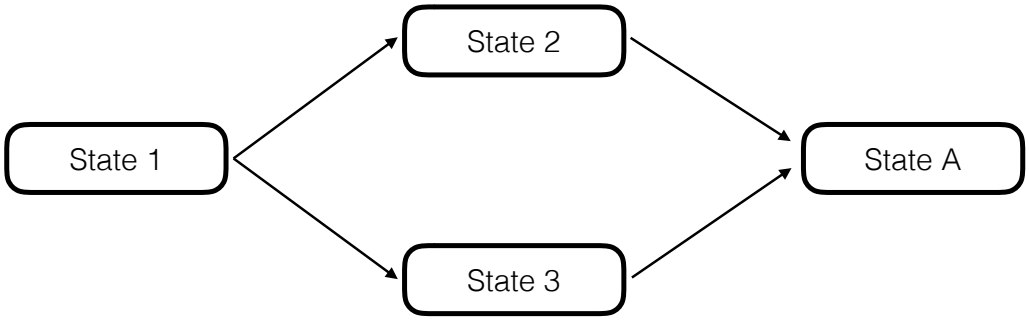

FIGURE 4.2: Scheme of a progressive MSM.

Bi-directional Models

What seems restrictive in previous structures is that there is no possible way back from a state. Once the subject has moved from a given state, it must follow the chain no matter what. Though this is not necessary unrealistic, it may be not the best option for certain problems. Here is where *bi-directional models* come into play. They allow transitions in either directions for transient states, or for a subset. A common example is the recovery from a chronic disease. A patient starts in a healthy state, then it gets diseased and eventually it can recover and get back to a healthy state again. We show a scheme of this model in Figure 4.3.

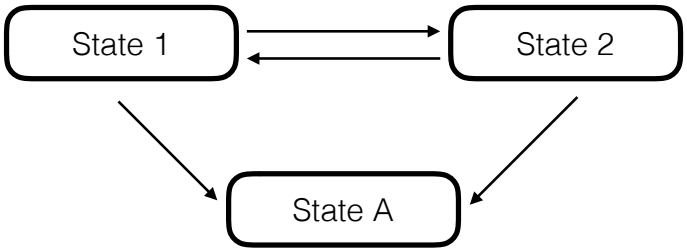

FIGURE 4.3: Scheme of a bi-directional MSM.

For instance, a bi-directional model will be used to analyze the hospital admissions process, as we will depict later on.

There can be cases in which no absorbing states are present either because they are not physically achievable nor because they are not part of the process under study. A very simple example is given by a bi-directional model in which all states are transient as sowed in Figure 4.4.

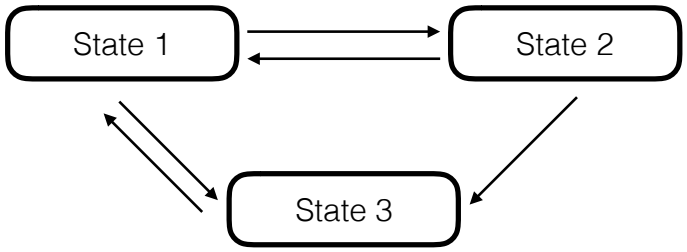

FIGURE 4.4: Scheme of a bi-directional recurrent MSM.

4.2.5 Observation Pattern

When collecting data, there can be basically two different approaches:

- *balanced observation:* this scheme assumes that all subjects are observed at a series of pre-defined times $t_1, \ldots, t_n$. In its simplest case, called *regular balanced observation* all the gap times between events are equally spaced. Observations

then take place at t, $2t, \ldots,$ nt. This type of scheme is commonly used in clinical trials, such the randomized controlled trial;

- *irregular observation:* this scheme does not assume any pre-defined time structure so that each subject can have its own. The observation times get a dependence on the subject i as follows $t_{1i}, \ldots, t_{n_i i}$. Our data belong to this category since the hospital admissions process cannot predetermined.

4.3 The 3-state Model

The main goal of this work is to study the hospital admission-readmission process given some patient's characteristics. This translates in studying the movement of a patient throughout different states defined in the database which we have built using `msmtools` package.

The structure of the model is bi-directional with $\mathcal{S} = \{1, \ldots, 3\}$ as showed in Figure 4.5. The model accounts for the process of admissions and readmissions into a given hospital and for the absorbing state of death. In other words, a patient is observed at first because he/she is admitted to a hospital. Here, he/she gains the status IN. Once the patient is in hospital, he/she can experience two different events. If the patient is discharged, then he/she moves to the state OUT. If he/she dies in hospital, then he/she moves to the absorbing state DEAD. Because the model is bi-directional, once the patient is in the state OUT, he/she can experience two different events once again: the former is being admitted back in a hospital thus moving to the state IN, the latter is dying outside the hospital thus moving to the absorbing state DEAD. Intuitively, this loop generates a wide variety of admission-readmission patterns. We will discuss some details of its distribution in Chapter 5.

Though the model structure is simple and the number of states is small, our main goal is to implement a robust statistical approach which allows us to investigate the potential effect of the brand new covariates. As far as we know, this is the very first Italian attempt to integrate the pharmacological and outpatient cares information

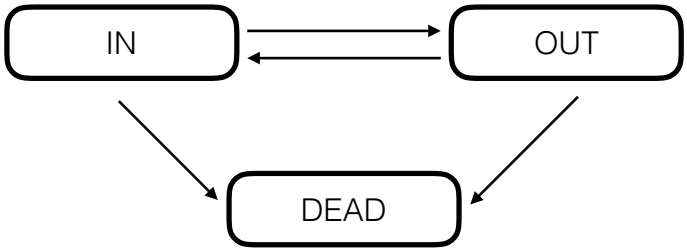

FIGURE 4.5: Scheme of the 3-state bi-directional MSM used for the assessment of the admission-readmission process.

in the evaluation of hospital admissions process. In particular, we are interested in investigating if and how much those covariates affect the transition probabilities of being readmitted into hospital or of dying outside a hospital [Grossetti et al., 2016].

Furthermore, covariates are used such that each of them intervene in a specific transition. In particular, we consider the information associated with a hospitalization to be more effective for those transitions which depart from the state IN. Conversely, all the cares which a patient may take have been considered to be more effective for transitions departing from the state OUT. The model structure with defined groups of covariates is showed in Figure 4.6.

The model represented in Figure 4.6 will be used under different assumptions. In the first place we consider a time-homogeneous fMM in which the h.f. is modeled with CPM through equation (4.8). In the second place we consider a fully-parametric approach by assuming a probability distribution for the baseline hazard $q_{rs}^{(0)}$ through equation (4.15).

In order to compute the hazards, we need to define the associated intensity matrix $Q(t) = Q_0$ which provides all the possible instantaneous transition rates. For our model, Q_0 is a 3×3 matrix with the following form:

$$Q_0 = \begin{pmatrix} 0 & q_{IN \to OUT} & q_{IN \to DEAD} \\ q_{OUT \to IN} & 0 & q_{OUT \to DEAD} \\ 0 & 0 & 0 \end{pmatrix}, \tag{4.28}$$

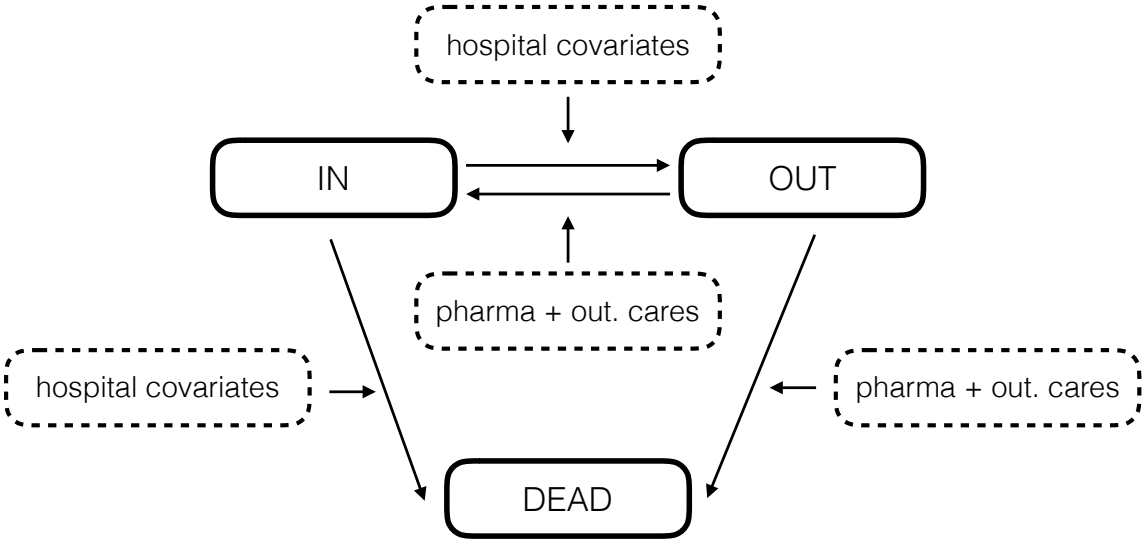

FIGURE 4.6: Scheme of the 3-state bi-directional MSM with the groups of covariates highlighted.

where 0 means that there is no possibility of a direct transition between the given two states.

In Table 4.1, we report the sets of covariates we adopted to run the models for each transition. The selection of these factors has been made jointly with clinicians in order to retain both the statistically significant covariates as well as the new variables. As we will discuss in the next chapter, we do expect a substantial impact of the new covariates on the admission-readmission probability. Models implementation and estimation has been carried out through the `R`'s `survival` [Therneau, 2015; Therneau and Grambsch, 2000], `mstate` de Wreede et al. [2011], `msm` [Jackson, 2011] and `flexsurv` [Jackson, 2016] packages for CPMs and pAFT models, respectively.

Transitions	Covariates
IN → OUT IN → DEAD	`age`, `gender`, `charlson`, `rehab`, `it`, `n_pro`, `n_com`
OUT → IN OUT → DEAD	`age`, `gender`, `charlson`, `LOS`, `C07`, `C09`, `sum_pa`

TABLE 4.1: Transition specific covariates used in the model of Figure 4.6 where `age` is the age in years a patient, `gender` indicates whether a patient is a man or a woman, `charlson` is the comorbidity score given by the Charlson index, `rehab` and `it` are binary flags indicating the passage in rehabilitation and/or intensive therapy units, respectively, `n_pro` and `n_com` give the number of comorbidities and of surgical procedures, respectively, `LOS` is the length of stay in hospital, `C07`, `C09` and `sum_pa` report the number of beta blocking agents, ACE-inhibitors agents and of cardiological outpatient cares registered between two subsequent events, respectively.

4.3.1 The parametric 3-state model

We have selected two distributions for the baseline hazard: the exponential and the Weibull leading to a PH model and to an AFT model, respectively. Through the function `flexsurvreg()` we obtain the estimates for the distributions parameters.

The Probability Density Function (PDF) of the exponential is:

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0, \end{cases} \tag{4.29}$$

where λ is the rate parameter which represents the event rate of the process. The higher λ is, the steeper the density is leading to a greater rate of event.

The PDF of the Weibull is:

$$f(x;\beta,k) = \begin{cases} k\beta\left(x\beta\right)^{k-1} e^{-(x\beta)^k} & x \geq 0, \\ 0 & x < 0, \end{cases} \tag{4.30}$$

where k and β are parameters called *shape* and *scale* parameters, respectively. Both are strictly greater than zero and need to be estimated. In particular, the shape parameter is the quantity which defines three possible regimes in the AFT mode:

- $k < 1$: the failure rate decreases over time. The number of observed events gets smaller over time, so that most of them occur at early stages;
- $k = 1$: the failure rate is constant over time. The Weibull reduces to the exponential;
- $k > 1$: the failure rate increases over time. The number of observe events gets larger over time.

4.4 Graphical Goodness of Fit Tools

Assessing the validity of a multi-state model is not straightforward. There are plenty of different GoF techniques and approaches to face this problem and in general, there exist two main categories under which to regroup them: the former considers informal approaches such graphical comparisons, the latter defines more formal statistical tests. In this work, we are going to use informal GoF tools only in order to better compare different methodological approach. Moreover, no statistical tests exist for panel-type data in which death times are known and exact.

Here we briefly discuss two graphical methods for assessing a multi-state model. In cases where the model has an absorbing state for which the exact time of entry is known (i.e. DEAD in our case), a common approach is to compare the KM survival curve with the estimate survival function which we call $\widehat{S}(t)$. This is a very simple, yet effective, tool to highlight any departure from the Markov model by considering the degree of disagreement between the two curves. The idea behind this, is to consider all the patients starting their process in the same state (i.e. IN in our case) at the beginning of the study. We assume the starting time to be zero and all the subjects move toward the defined absorbing state. If model assumptions are correct, we should not observe a disagreement between empirical and estimated survival curves. What is non-trivial here is to determine the threshold within which we consider a model to be acceptable or not.

There are basically two ways of visualizing the comparison: the first one reports the confidence bands around $\widehat{S}(t)$, the second one reports bands around the KM curve. In both methods, we are seeking to minimize the disagreement between the two point-wise curves and we consider the confidence bands as the common acceptance limits for which discard or not discard the multi-state model. While the second approach is very simple and is typically calculated using a normal approximation like Greenwood's formula [Jackson, 2000], in general the confidence limit tends to be a little broader than what requested. Bands around $\widehat{S}(t)$ are straightforward in case of semi-Markov or fully-parametric models, but are not easily computable for bi-directional

Markov models because closed form expressions for the transitions probabilities are not available. It is however possible to differentiate the associated matrix exponential such that we obtain an expression for the derivates in term of partial derivatives and the eigenvalues and eigenvectors of the intensity matrix $Q(t)$. Hence, confidence intervals is computed by simulating a given number of random vectors from the asymptotic multivariate normal distribution implied by the maximum likelihood estimates and covariance matrix of the log transition intensities and covariate effects. Then, for each replicate the resulting transition probability matrix is computed. The described approach is discussed in Mandel [2013] and can be computationally intensive because intervals must be calculated at a series of times.

For Markov models, another well known tool is based on the comparison between observed state occupancies at a fixed set of times and expected ones. The work of Gentleman et al. [1994] discusses the method and its applicability. An indication of where the data depart from the model is achieved by comparing the observed count $O_{rs}(t_1, t_2)$ with the expected count $E_{rs}(t_1, t_2)$ for individuals in the state r at time t_1 and in state s at time t_2 through the following quantity:

$$M_{rs} = \frac{[O_{rs}(t_1, t_2) - E_{rs}(t_1, t_2)]^2}{E_{rs}(t_1, t_2)}. \tag{4.31}$$

An ideal situation would see $M_{rs} \to 0$ for any given point in time. Any deviance from zero, implies a worsening of the model performances. However, determining whether a deviation is statistically significant is currently impossible because no formal tests are available yet. The main reason is related to the totally custom choice of time knots at which interpolate the observed data. To estimate prevalence, the differential form of the Aalen-Johansen (AJ) estimator [Aalen and Johansen, 1978] can be used as follows:

$$\hat{p}_{1s}(t + \delta t) = \sum_{r=1}^{R} \hat{p}_{1r} \frac{\mathrm{d}N_{rs}(t)}{Y_r(t)}, \tag{4.32}$$

where $Y_r(t)$ is the number of subjects under observation at time t in state r, $\mathrm{d}N_{rs}(t)$

is the number of transitions from state r to s in the time interval $(t, t + \delta t)$, and $\hat{p}_{1r}$ is the corresponding 1, r value of the transition probability matrix. Actually, the AJ estimator can be seen as a finite product of matrices. When right-censoring is assumed, then this estimator is more efficient than considering just the proportion of those subjects who are still under observation in each defined state[2].

[2]This is also known as the *simple moment estimator*.

Chapter 5

Results

In this chapter we present the results of the statistical models. After some model assessment, we will provide some interpretations. We also present some descriptive statistics which helps in understanding heterogeneity of the patient's population.

5.1 Descriptives

We do possess several information regarding each patient. In order to have an idea of the data composition, we focus our attention on few quantities. For these, we take a snapshot by providing histograms and other significant summaries.

As we have described in Subsection 3.3.4, the final sample of 144,933 patients has been downsized to 101,821 in order to solve the same event at the same time issue. The descriptives are then computed on this sample size.

A total of 36,109 (35.46%) patients died for any cause, 16,153 (15.86%) of which died during a hospital admission. Age at first admission ranged from 18 to 106 years for women with a mean (SD) of 79.4 (10.4) years. Men's age ranged from 18 to 104 years with a mean (SD) of 73.6 (11.9) years. Age at last discharge ranged from 18 to 106 years for women with a mean (SD) of 80.6 (10.4) years. Men ranged from 18 to 106 years with a mean (SD) of 75.0 (11.9) years. In Figure 5.1, we show

comparison boxplots of age for women and men at first admission and at last discharge, respectively. Wilcoxon rank sum tests confirm a significant difference in the gender distribution for both cases ($p \leq 0.001$).

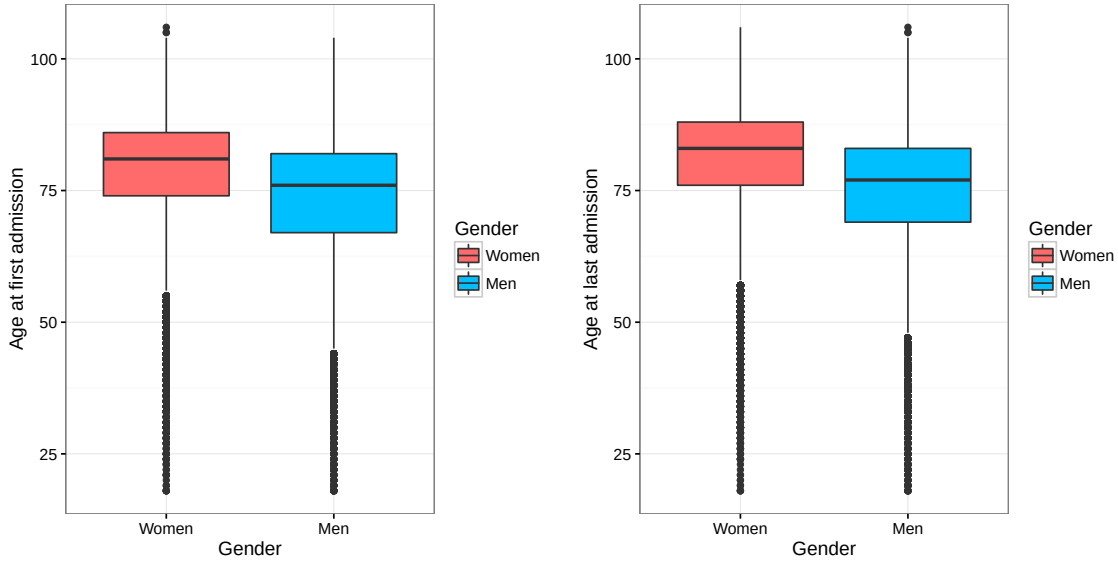

FIGURE 5.1: Boxplots of ages at first (left panel) and at last admission (right panel) grouped by gender.

The average number (SD) of hospital admissions for women is 2.5 (1.95) with a range of 1 - 34, while for men is 2.8 (2.25) with the same range. In Figure 5.2, we show comparison boxplots of the number of events for women and men. Wilcoxon rank sum tests confirm a significant difference in the gender distribution for both cases ($p \leq 0.001$).

The LOS ranges from 1 to 159 days for women with a mean (SD) of 10.4 days (6.5). Men's LOS ranges from 1 to 148 days with a mean (SD) of 9.3 days (6.3). Wilcoxon rank sum test confirms a significant difference in the LOS distribution ($p \leq 0.001$) with respect to gender. In Figure 5.3, we show a comparison boxplots of LOS.

In Table 5.1, we report the number of transitions and their percentage with respect to the total. Dependence has been assessed through the Pearson's χ^2-test ($p \leq 0.001$).

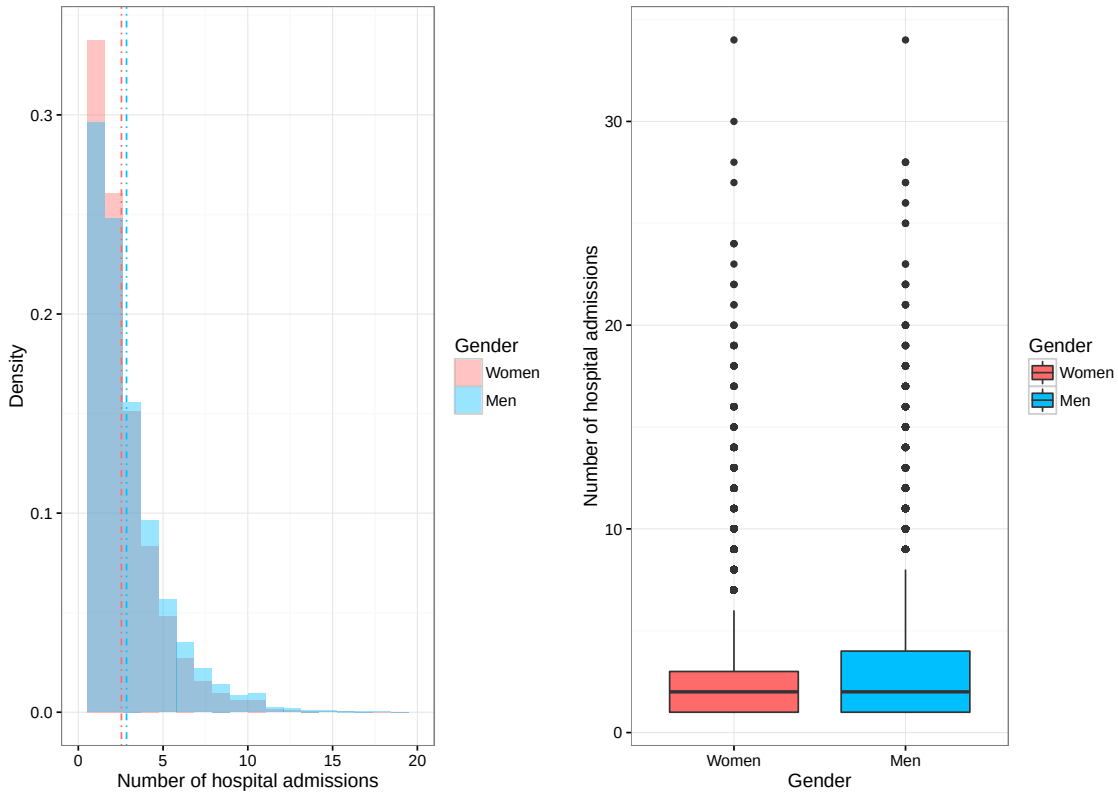

FIGURE 5.2: Histograms (left panel) and boxplots (right panel) of the number of hospital admissions grouped by gender. The dashed-dotted lines in the histogram panel mark the average values.

In Table 5.2, we report the number (percentage) of diagnosed comorbidities per patient with respect to gender. Column ">3" counts all the patients who have more than 3 diagnosed comorbidities. We confirm a significant difference between the two distributions, according to a Wilcoxon rank sum test ($p \leq 0.001$). In Table 5.3, we report the number (percentage) of surgical procedures a patient experienced with respect to gender. Column ">3" counts all the patients who have undergone more than 3 surgical procedures. We confirm a significant difference between the two distributions, according to a Wilcoxon rank sum test ($p \leq 0.001$). Much of this difference is represented by having just one procedure or having no procedures at all.

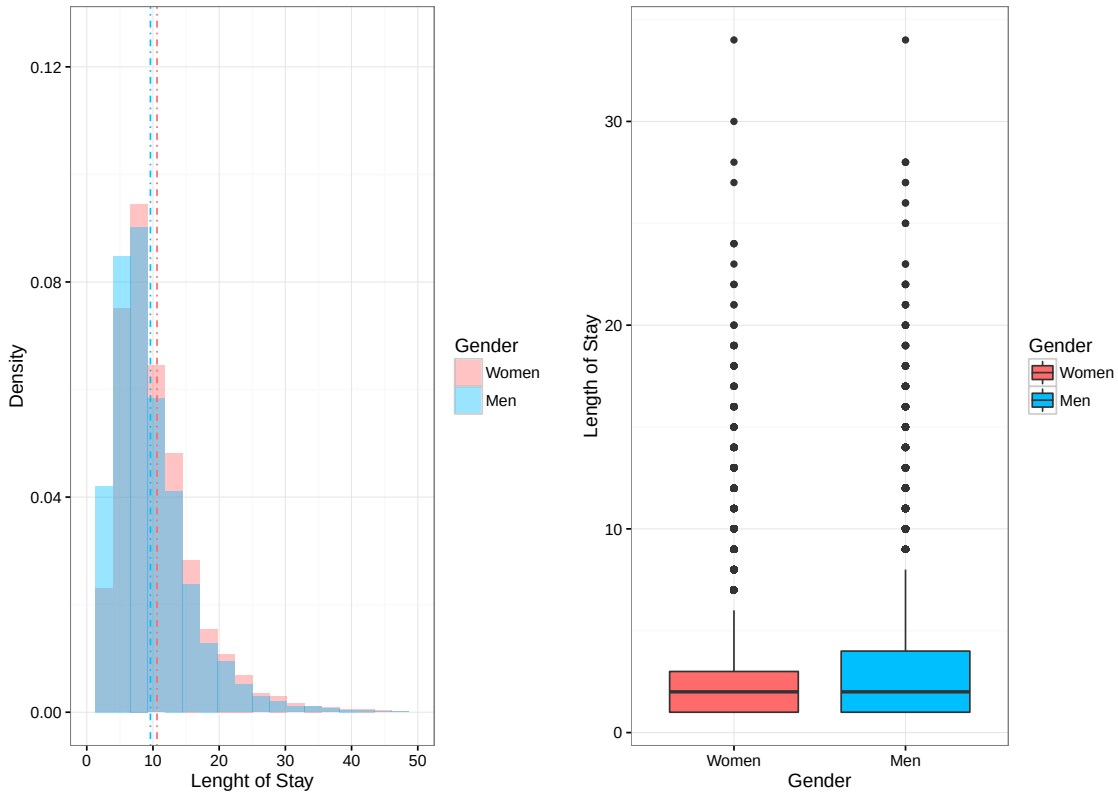

FIGURE 5.3: Histograms (left panel) and boxplots (right panel) of Length of Stay grouped by gender. The dashed-dotted lines in the histogram panel mark the average values.

We see how men count more procedures than women.

		To		
		IN	OUT	DEAD
From	IN	-	258,330 (40.71%)	16,153 (2.55%)
	OUT	172,662 (27.21%)	65,712 (10.35%)	19,956 (3.21%)

TABLE 5.1: Number of transitions (and percentage) recorded according to the transition matrix Q_0.

Gender		Diagnosed Comorbidities				
		0	1	2	3	> 3
Women	# com.	117	9,521	16,055	12,241	12,711
	[%]	0.23	18.80	31.70	24.17	25.10
Men	# com.	125	9,456	14,788	11,952	14,865
	[%]	0.24	18.47	28.89	23.35	29.04

TABLE 5.2: Number of diagnosed comorbidities with respect to gender. The last column on the right counts all the patients with more than 3 comorbidities.

Gender		Number of Surgical Procedures				
		0	1	2	3	> 3
Women	# proc.	45,757	4,710	167	9	2
	[%]	90.35	9.30	0.33	0.02	0
Men	# proc.	40,425	10,230	516	15	0
	[%]	78.98	19.99	1.01	0.03	0

TABLE 5.3: Number of surgical procedures with respect to gender. The last column on the right counts all the patients who experienced more than 3 surgical procedures.

In Figure 5.4, we show the proportion of patients, grouped by gender, who take or not take any beta blocking agents, ACE-inhibitors, and cardiological outpatient cares. For instance we see that among women, 28,003 (55.3%) did take beta blocking agents while 22,638 (44.7%) did not. Among men, 31,983 (62.5%) did take beta blocking agents while 19,197 (37.5%) did not. For ACE-inhibitors, 12,992 (25.7%) women did take it while 37,649 (74.3%) did not. For men, 40,355 (78.8%) did take it while 10,825 (21.2%) did not. Among the cardiological outpatient cares, 36,207 (71.5%) women did take any cares while 14,434 (28.5%) did not. Among men, 42,009 (82.1%) did take them while 9,171 (17.9%) did not. We can see how men consistently take more drugs and outpatient cares with respect to women. One of the possible explanation of this behaviour could be a general more severe condition of the male population in the sample. Though we are awared of the gender bias against women with cardiovascular disease in terms of access to revascularization. Moreover, men are younger than women (i.e. they generally die sooner) and they could have a slightly different aetiology.

In Figure 5.5, we show the drill down on the drug prescriptions for patients who did not take anything at all (label *Nothing* in the plot), did take only beta blocking agents (label *C07* in the plot), did take only ACE-inhibitors (label *C09* in the plot), and did take both of them at least once (label *Both* in the plot). Data are shown grouped by gender. 22,782 (45.0%) women did take both, 5,221 (10.3%) did take beta blocking agents only, 14,867 (29.4%) ACE-inhibitors only, while 7,771 (15.3%) women did not take any drugs. Among men, 27,567 (53.9%) did take both, 4,416 (8.6%) did take beta blocking agents only, 12,788 (25.0%) ACE-inhibitors only, 27,567 (53.9%), while 6,409 (12.5%) did not take any drugs. As we can see, ACE-inhibitors seem to be the preferred treatment in terms of prescriptions. Though we do not possess exact data related to therapy compliance but only of puprescription purchasing, we can think of these results as a compliance proxy. It seems that men are slightly less compliance when a single therapy is prescribed, but better perform women when both drugs are suggested.

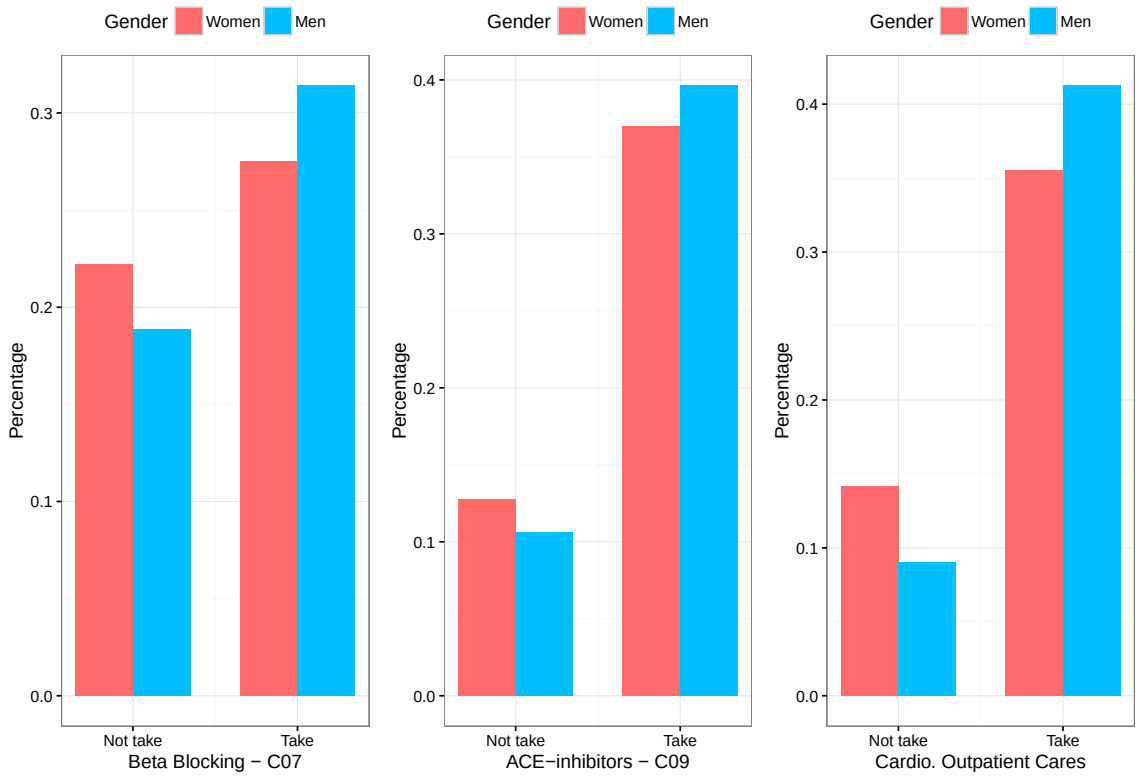

FIGURE 5.4: Barplots of the percentage of patients who take any beta blocking (C07) or ACE-inhibitors (C09) (left and center panel, respectively) and cardiological outpatient cares (right panel) grouped by gender.

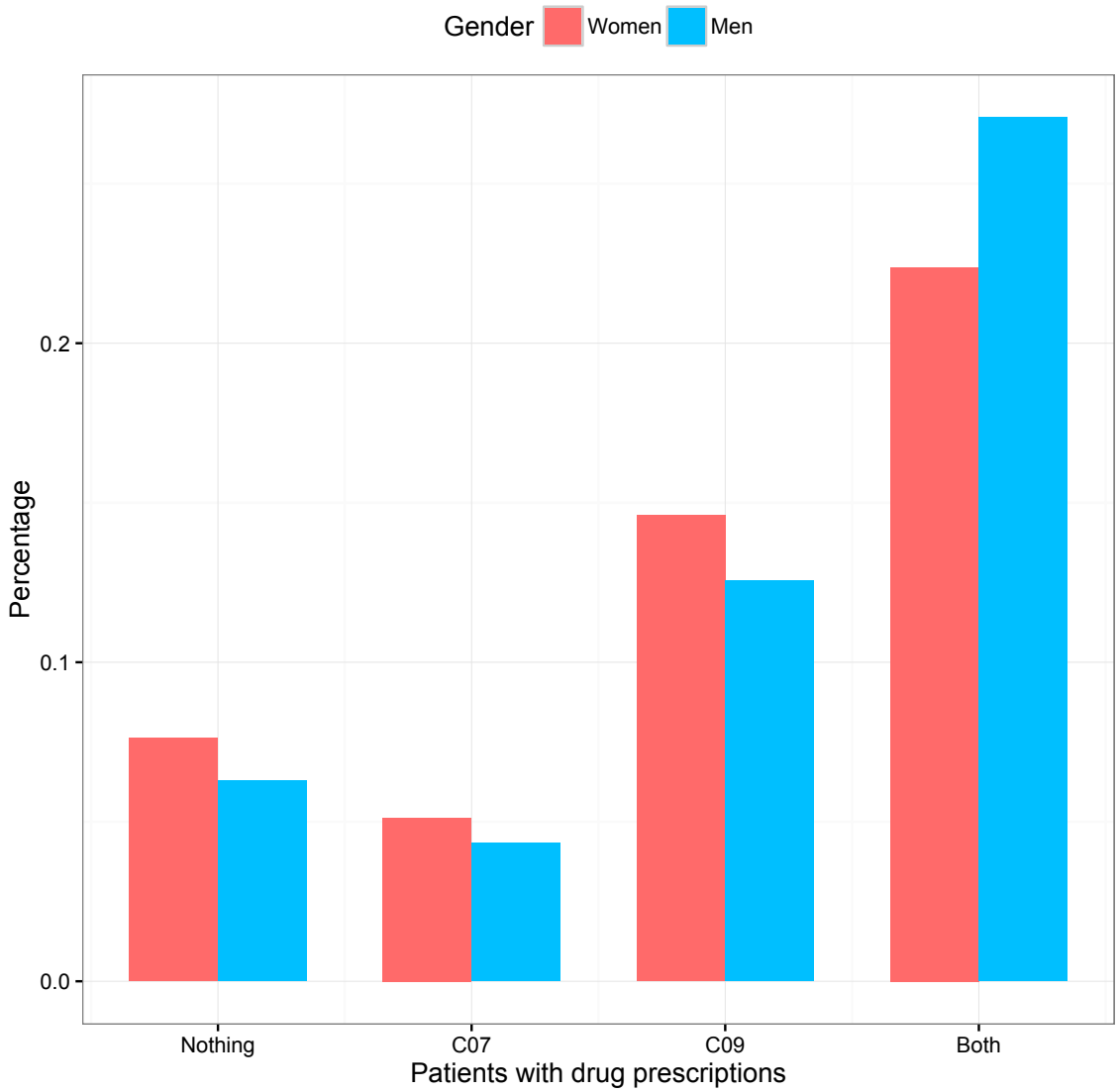

FIGURE 5.5: Barplot of the percentage of drug prescriptions. "Nothing" represents patients who did not take C07 nor C09, "C07" and "C09" patients who did take beta blocking agents or ACE-inhibitors only, respectively, "Both" patients who did take both C07 and C09. Data are grouped by gender.

5.2 Model Results

In this section we present and discuss the results of the different statistical models we have adopted. In particular, in Subsections 5.2.1, 5.2.2, and 5.2.3, we show the main outputs of the application of a full-, semi-Markov and fully-parametric models, respectively. The three different approaches have been carried out using different `R` packages as standalone tools or in one of their combination. We highlight the most important ones which are: `survival` [Therneau, 2015; Therneau and Grambsch, 2000], `msm` [Jackson, 2011], `mstate` [de Wreede et al., 2011], and `flexsurv` [Jackson, 2016].

The last two approaches require a further data reshaping procedure in order to transform augmented or longitudinal data into a suitable form. We have used the `msm` function `msm2Suv()` to carry out the procedure which takes a longitudinal dataset and restructures it such that both observed and censored transitions are visible and available at a glance. To clarify the concept, in Tables 5.4 and 5.5 we report, for a given patient with few selected covariates, the augmented format as computed by `augment()` and this new shape, respectively. As we can see, we still have multiple observations for a single patient, but we also have several more related information. They are:

- `from` and `to`: the departing and arrival status coded as in the original transition matrix Q_0;
- `trans`: the corresponding transition number as coded in Q_0;
- `Tstart` and `Tstop`: the starting and ending time of a given transition;
- `time`: the clock-reset time scale of the event;
- `obs/cens`: a flag which marks an observed transition (1) or a censored one (0).

For a detailed description of the data structure and its applications, we remind to the following two works of de Wreede et al. [2011] and Putter et al. [2007].

row	ID	dateAUGMENTED	age	gender	status
1:	35	2010-06-10	48	Man	IN
2:	35	2010-06-17	48	Man	OUT
3:	35	2010-12-20	49	Man	IN
4:	35	2010-12-22	49	Man	OUT
5:	35	2011-06-07	49	Man	IN
6:	35	2011-06-10	49	Man	OUT
7:	35	2011-10-08	49	Man	DEAD

TABLE 5.4: Example of the augmented representation of the events for patient 35 as computed by the function `augment()` in `msmtools`.

For the semi-Markov and fully-parametric approaches, we are going to use the structure as shown in Table 5.5 as well as the clock-reset time variable to mark the process events. For the semi-Markov model only, a further step is mandatory if transition specific covariates are considered. We thus use the function `expand.covs()` in the package `mstate` which takes the new data structure and expands the required covariates so that a transition specific model can be run. In Table 5.6, we show the expanded data structure in which `age` has been converted into a transition specific covariate. Due to the fact that we have a model with four possible transitions, the covariate is splitted accordingly being `age.1` the value of patient's age in the transition 1, coded by `trans = 1`, which corresponds to the movement IN $\rightarrow$ OUT. Hence, `age.2`, `age.3`, and `age.4`, are the corresponding age values for transition 2, 3, and 4, respectively.

row	ID	from	to	trans	Tstart	Tstop	time	obs/cens	age	gender
1:	35	1	2	1	2010-06-10	2010-06-17	7	1	48	Man
2:	35	1	3	2	2010-06-10	2010-06-17	7	0	48	Man
3:	35	2	1	3	2010-06-17	2010-12-20	186	1	48	Man
4:	35	2	3	4	2010-06-17	2010-12-20	186	0	48	Man

5:	35	1	2	1	2010-12-20	2010-12-22	2	1	49	Man
6:	35	1	3	2	2010-12-20	2010-12-22	2	0	49	Man
7:	35	2	1	3	2010-12-22	2011-06-07	167	1	49	Man
8:	35	2	3	4	2010-12-22	2011-06-07	167	0	49	Man
9:	35	1	2	1	2011-06-07	2011-06-10	3	1	49	Man
10:	35	1	3	2	2011-06-07	2011-06-10	3	0	49	Man
11:	35	2	1	3	2011-06-10	2011-10-08	120	0	49	Man
12:	35	2	3	4	2011-06-10	2011-10-08	120	1	49	Man

TABLE 5.5: Example of the longitudinal representation of the events for patient 35 as computed by the function `msm2Surv()` in `msm`.

row	ID	from	to	trans	Tstart	Tstop	age.1	age.2	age.3	age.4
1:	35	1	2	1	2010-06-10	2010-06-17	48	0	0	0
2:	35	1	3	2	2010-06-10	2010-06-17	0	48	0	0
3:	35	2	1	3	2010-06-17	2010-12-20	0	0	48	0
4:	35	2	3	4	2010-06-17	2010-12-20	0	0	0	48
5:	35	1	2	1	2010-12-20	2010-12-22	49	0	0	0
6:	35	1	3	2	2010-12-20	2010-12-22	0	49	0	0
7:	35	2	1	3	2010-12-22	2011-06-07	0	0	49	0
8:	35	2	3	4	2010-12-22	2011-06-07	0	0	0	49
9:	35	1	2	1	2011-06-07	2011-06-10	49	0	0	0
10:	35	1	3	2	2011-06-07	2011-06-10	0	49	0	0
11:	35	2	1	3	2011-06-10	2011-10-08	0	0	49	0
12:	35	2	3	4	2011-06-10	2011-10-08	0	0	0	49

TABLE 5.6: Example of the expanded `age` for patient 35 as computed by the function `expand.covs()` in `mstate`.

5.2.1 Full Markov Model Results

The parameters estimation is carried out through the function `msm()` in the package `msm` which is applied to the data described in Table 5.4. In Table 5.7, we report the hazard ratios and the relative 95% confidence intervals for the covariates which are defined for all the transitions allowed in the model as specified by the transition matrix Q_0. They are `age`, `gender` and `charlson`. The `gender`'s values are reported for men with respect to women. With an increase in the age of a patient, the risk of being discharged from (HR = 0.9901 [0.9898; 0.9905]) or readmitted to a hospital (HR = 0.9890 [0.9886; 0.9894]) decreases. At the same time, we observe the natural effect of aging when looking at transitions to death for which the hazard ratios are greater than one. Being a men increases the risk of all the transitions to occur. An increase of the Charlson index `charlson` increases the risk of dying both inside and outside a hospital (HR = 1.2205 [1.2081; 1.2330] and HR = 1.2422 [1.2338; 1.2508], respectively). It protracts the sojourn in the hospital (HR = 0.9317 [0.9286; 0.9347]) and also increases the chances of a hospital readmission (HR = 1.0579 [1.0550; 1.0609]).

Transition	`age`	`gender` (men)	`charlson`
IN → OUT	0.9901 [0.9898; 0.9905]	1.0990 [1.0903; 1.1078]	0.9317 [0.9286; 0.9347]
IN → DEAD	1.0576 [1.0555; 1.0597]	1.0755 [1.0415; 1.1105]	1.2205 [1.2081; 1.2330]
OUT → IN	0.9885 [0.9881; 0.9889]	1.1550 [1.1437; 1.1664]	1.0579 [1.0550; 1.0609]
OUT → DEAD	1.0507 [1.0489; 1.0526]	1.0833 [1.0522; 1.1154]	1.2422 [1.2338; 1.2508]

TABLE 5.7: Hazard Ratios and 95% confidence intervals for `age`, `gender`, and `charlson` as computed by the full-Markov model. The `gender` covariate refers to the men with respect to women.

In Table 5.8, we report the hazard ratios for the covariates which are defined only for transitions departing from the state IN. They are, `rehab`, `it`, `n_com` and `n_pro`.

Transition	rehab	it	n_com
IN → OUT	0.4113 [0.4032; 0.4195]	0.6459 [0.6368; 0.6552]	0.9923 [0.9883; 0.9963]
IN → DEAD	0.4906 [0.4500; 0.5349]	1.5097 [1.4433; 1.5790]	1.0713 [1.0560; 1.0868]
Transition	n_pro		
IN → OUT	0.8879 [0.8742; 0.9017]		
IN → DEAD	2.9423 [2.8350; 3.0537]		

TABLE 5.8: Hazard Ratios and 95% confidence intervals for `rehab`, `it`, `n_com`, and `n_pro` as computed by the full-Markov model. The `gender` covariate refers to the men with respect to women.

Here several considerations can be done. The passage through the rehabilitation unit decreases the chances of both transitions. A possible explanation could be that a rehabilitative admission takes longer care time, so a patient is likely to stay more days in a hospital (HR = 0.4113 [0.4032; 0.4195]). These cares also help the patient to diminish the risk of death (HR = 0.4906 [0.4500; 0.5349]). The passage through the intensive therapy unit acts the same way as the rehabilitative admission (HR = 0.6459 [0.6368; 0.6552]), but increases the chances of dying (HR = 1.5097 [1.4433; 1.5790]). This is something we can expect since a patient who undergoes any intensive therapy surgery has a high probability of being in severe conditions. The number of comorbidities and surgical procedures affect the risk of having a transition in the same way. Both decrease the chances of being discharged from hospital sooner (HR = 0.9923 [0.9883; 0.9963]; and HR = 0.8879 [0.8742; 0.9017], respectively) but increase the probability of dying. In particular, it seems that the surgical procedure have a stronger impact (HR = 2.9423 [2.8350; 3.0537]) on the mortality of a patient compared to the comorbidities (HR = 1.0713 [1.0560; 1.0868]).

In Table 5.9, we report the hazard ratios for the covariates which are defined only for transitions departing from the state OUT. They are LOS, `C07`, `C09`, and `sum_PA`.

Transition	LOS	C07	C09
OUT → IN	0.9992 [0.9986; 0.9998]	0.9618 [0.9608; 0.9628]	0.9427 [0.9420; 0.9435]
OUT → DEAD	1.0191 [1.0178; 1.0205]	0.9402 [0.9361; 0.9444]	0.9185 [0.9155; 0.9215]
Transition	sum_PA		
OUT → IN	0.9752 [0.9746; 0.9757]		
OUT → DEAD	0.9362 [0.9331; 0.9394]		

TABLE 5.9: Hazard Ratios and 95% confidence intervals for LOS, C07, C09, and sum_PA as computed by the full-Markov model.

Here we observe a general decrease in the risk of having a transition, no matter which it is except for an increase in the LOS which also increases the risk of dying outside the hospital. The new covariates coming from the pharmacological and outpatient cares seem to be effective in the prevention of patient's death. In particular, the acquisition of agents acting on the renin-angiotensin system, in the form of ACE-inhibitors (C09), seems to be very effective.

Through the function `survplot()` in the package `msmtools`, we are able to build specific survival curves for a given patient's profile. That is, for a given set of covariates, we compute the fitted survival curves as estimated by `msm()`. To give some examples, we show a series of plots in which we focus on the effects of a single covariate of interest. For all the other model's covariates, we consider their mean values. In Figures 5.6 and 5.7, we can see how different increments in the number of comorbidities and surgical procedures, respectively, change the survival function. In particular, it seems again that procedures stronger affect the outcome.

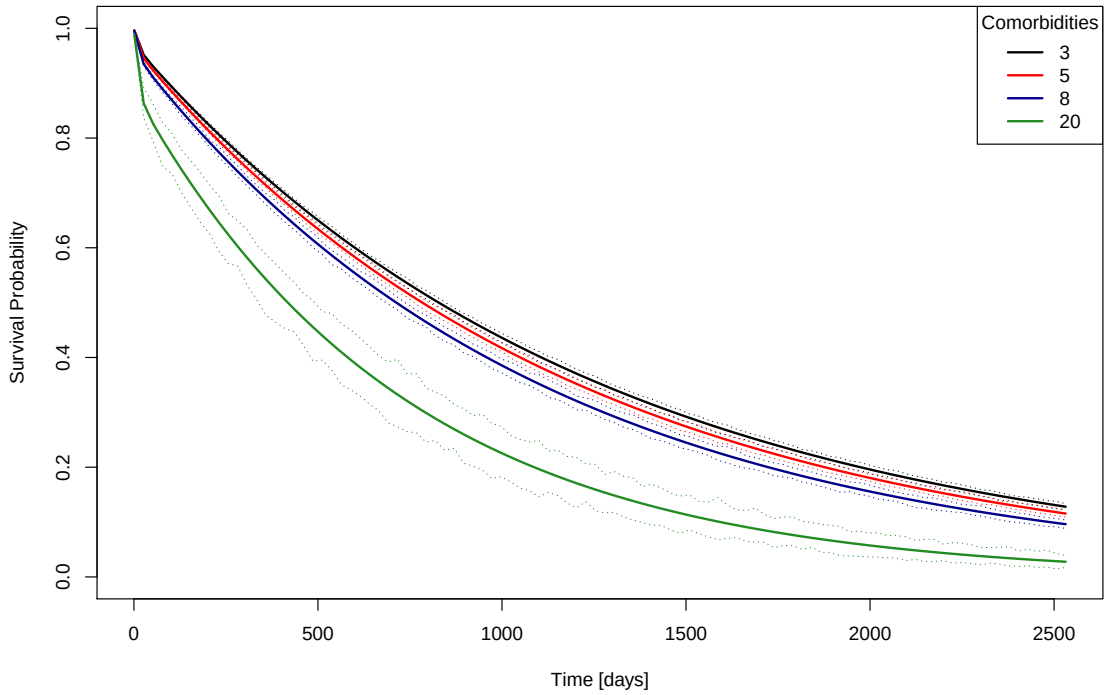

FIGURE 5.6: Estimated survival curves for the transition IN → DEAD for changing values of the number of comorbidities. `n_com` is set to 3 (black curve), 5 (red curve), 8 (blue curve), 20 (green curve). For all the other covariates, the mean is taken. 95% confidence intervals are also plotted.

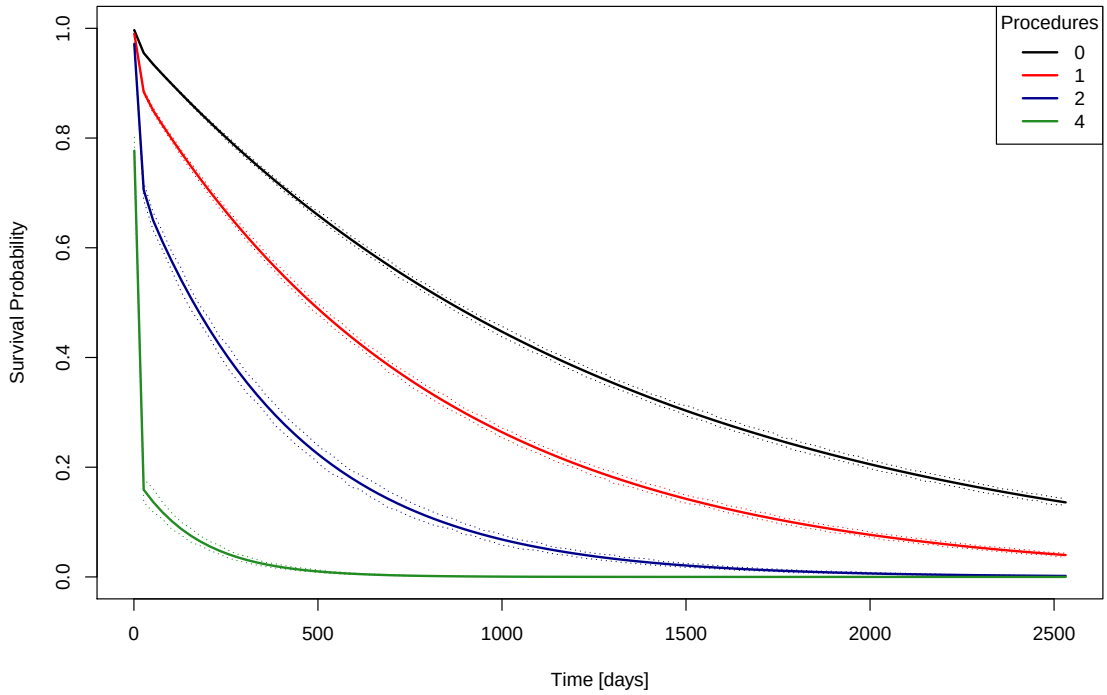

FIGURE 5.7: Estimated survival curves for the transition IN → DEAD for changing values of the number of procedures. `n_pro` is set to 0 (black curve), 1 (red curve), 2 (blue curve), 4 (green curve). For all the other covariates, the mean is taken. 95% confidence intervals are also plotted.

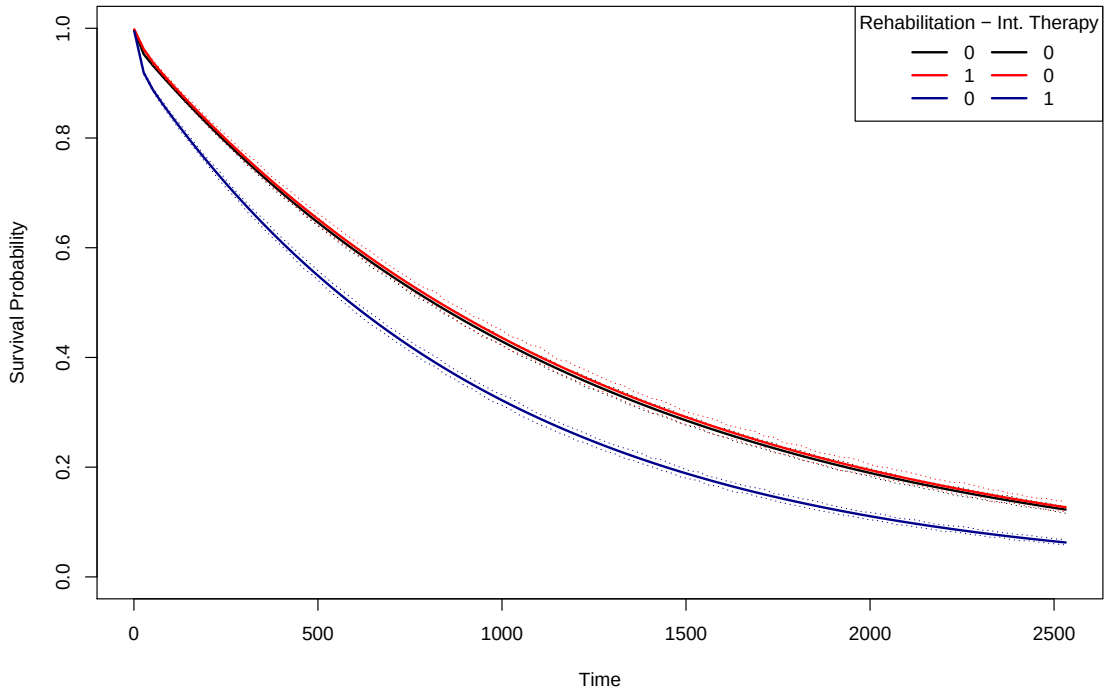

FIGURE 5.8: Estimated survival curves for the transition IN → DEAD for for changing values in the rehabilitation and intensive therapy flags. Absence of rehabilitation and intensive therapy cares (black curve), rehabilitation passage only (red curve), intensive therapy passage only (blue curve). The black and red curve collapse one on another even if the rehabilitation curve has a slightly more positive effect.

5.2.2 Semi-Markov Model Results

Semi-Markov models relax the Markov assumption by including, in the definition of the hazard function, the sojourn time in the previous state. The parameters estimation is carried out through the function `coxph()` in the package `survival` which is applied to the data described in Table 5.6. The covariate structure is the same of the full Markov model. As we can see from table 5.10, 5.11, and 5.12, the model behaviour is the same of the full Markov one. The relation between the covariates and the risk associated with a given transition is thus retained. Even the intensities in the hazard ratios are pretty similar.

Transition	`age`	`gender` (men)	`charlson`
IN → OUT	0.9878 [0.9874; 0.9881]	1.1272 [1.1182; 1.1362]	0.9148 [0.9118; 0.9178]
IN → DEAD	1.0585 [1.0564; 1.0605]	1.0740 [1.0401; 1.1089]	1.2178 [1.2055; 1.2303]
OUT → IN	0.9893 [0.9889; 0.9898]	1.1482 [1.1370; 1.1596]	1.0515 [1.0486; 1.0545]
OUT → DEAD	1.0530 [1.0511; 1.0549]	1.0743 [1.0434; 1.1061]	1.2322 [1.2238; 1.2406]

TABLE 5.10: Hazard Ratios and 95% confidence intervals for `age`, `gender`, and `charlson` as computed by the semi-Markov model. The `gender` covariate refers to the men with respect to women.

Transition	`rehab`	`it`	`n_com`
IN → OUT	0.3372 [0.3305; 0.3440]	0.5850 [0.5766; 0.5934]	0.9905 [0.9865; 0.9945]
IN → DEAD	0.4182 [0.3820; 0.4579]	1.5238 [1.4569; 1.5938]	1.0697 [1.0544; 1.0852]
Transition	`n_pro`		
IN → OUT	0.8872 [0.8736; 0.9011]		
IN → DEAD	2.8043 [2.7016; 2.9108]		

TABLE 5.11: Hazard Ratios and 95% confidence intervals for `rehab`, `it`, `n_com`, and `n_pro` as computed by the semi-Markov model. The `gender` covariate refers to the men with respect to women.

Transition	LOS	C07	C09
OUT → IN	0.9992 [0.9986; 0.9997]	0.9639 [0.9629; 0.9649]	0.9454 [0.9446; 0.9461]
OUT → DEAD	1.0189 [1.0176; 1.0203]	0.9469 [0.9430; 0.9509]	0.9244 [0.9215; 0.9273]
Transition	sum_PA		
OUT → IN	0.9769 [0.9763; 0.9775]		
OUT → DEAD	0.9414 [0.9383; 0.9445]		

TABLE 5.12: Hazard Ratios and 95% confidence intervals for LOS, C07, C09, and sum_PA as computed by the semi-Markov model. The gender covariate refers to the men with respect to women.

5.2.3 Fully Parametric Models Results

In this subsection, we present the results obtained by assuming a probability distribution for the baseline hazard. We have run two different models: the former assumes an exponential baseline, the latter a Weibull one.

Exponential Baseline

In Table 5.13, we report the estimated rate parameter for the exponential distribution assumed for the baseline hazard. We observe how the transitions related purely to admissions and readmissions ones have a rate parameter way greater than those related to death. This means that the decay rate of the survival function is faster with respect to admissions and readmissions than the one of deaths.

In Tables 5.14, 5.15, and 5.16 we report the model results for the different groups of covariates. The considerations discussed in Subsection 5.2.1 still hold here. Hence, we observe the strong effects of intensive therapy and surgical procedures. We confirm again the positive effects, in terms of enhancing survival chances, of pharmacological

and outpatient cares covariates. Despite these similar results, the shape of the estimated survival curves has remarkably improved by better mimic the empirical ones.

Transition	`rate`
IN → OUT	0.2395886 [0.2325853; 0.2468027]
IN → DEAD	0.0000290 [0.0000245; 0.0000344]
OUT → IN	0.0143289 [0.0138367; 0.0148386]
OUT → DEAD	0.0000098 [0.0000083; 0.0000114]

TABLE 5.13: Estimates and 95% confidence intervals for `rate` parameter of the baseline distribution as computed by the fully-parametric model with an exponential baseline.

Transition	`age`	`gender` (men)	`charlson`
IN → OUT	0.9903 [0.9900; 0.9907]	1.1022 [1.0934; 1.1110]	0.9321 [0.9291; 0.9352]
IN → DEAD	1.0570 [1.0549; 1.0591]	1.0720 [1.0382; 1.1069]	1.2196 [1.2073; 1.2321]
OUT → IN	0.9885 [0.9881; 0.9889]	1.1548 [1.1435; 1.1662]	1.0579 [1.0550; 1.0609]
OUT → DEAD	1.0502 [1.0483; 1.0520]	1.0773 [1.0464; 1.1091]	1.2415 [1.2331; 1.2500]

TABLE 5.14: Hazard Ratios and 95% confidence intervals for `age`, `gender`, and `charlson` as computed by the fully-parametric model with an exponential baseline. The `gender` covariate refers to the men with respect to women.

Transition	rehab	it	n_com
IN → OUT	0.4185 [0.4103; 0.4268]	0.6596 [0.6503; 0.6690]	0.9892 [0.9852; 0.9932]
IN → DEAD	0.4527 [0.4140; 0.4951]	1.5632 [1.4954; 1.6341]	1.0720 [1.0567; 1.0875]
Transition	n_pro		
IN → OUT	0.8645 [0.8511; 0.8782]		
IN → DEAD	2.8595 [2.7547; 2.9682]		

TABLE 5.15: Hazard Ratios and 95% confidence intervals for `rehab`, `it`, `n_com`, and `n_pro` as computed by fully-parametric model with an exponential baseline.

Transition	LOS	C07	C09
OUT → IN	0.9992 [0.9986; 0.9998]	0.9617 [0.9607; 0.9627]	0.9427 [0.9420; 0.9435]
OUT → DEAD	1.0190 [1.0177; 1.0204]	0.9407 [0.9366; 0.9448]	0.9188 [0.9158; 0.9218]
Transition	sum_PA		
OUT → IN	0.9752 [0.9746; 0.9757]		
OUT → DEAD	0.9367 [0.9336; 0.9398]		

TABLE 5.16: Hazard Ratios and 95% confidence intervals for `LOS`, `C07`, `C09`, and `sum_PA` as computed by the fully-parametric model with an exponential baseline.

Weibull Baseline

In Table 5.17, we report the estimated `shape` and `scale` parameters for the Weibull distribution assumed for the baseline hazard. In particular, since this is an AFT model, the `shape` parameter reflects the acceleration (deceleration) of the time flowing. In other words, if the parameter is > 1, then the time to transition would be reduced and viceversa. We can see that there is a speed up for transitions departing from state IN and a slow down for transitions departing from state OUT.

In Tables 5.18, 5.19, and 5.20, we report the hazard ratios estimated by the fully-parametric model in which the baseline hazard is distributed as a Weibull.

Transition	`shape`	`scale`
IN → OUT	1.3115 [1.3075; 1.3156]	6.3205 [6.1734; 6.4710]
IN → DEAD	1.0429 [1.0299; 1.0561]	24852.1760 [20530.8124; 30083.1082]
OUT → IN	0.9106 [0.9073; 0.9138]	63.2889 [60.8855; 65.7872]
OUT → DEAD	0.8612 [0.8523; 0.8702]	330141.7708 [269078.4219; 405062.5393]

TABLE 5.17: Estimates and 95% confidence intervals for `shape` and `scale` parameters as computed by the `flexsurvreg()` under a fully-parametric model with a weibull baseline.

Transition	`age`	`gender` (men)	`charlson`
IN → OUT	1.0051 [1.0048; 1.0054]	0.8973 [0.8918; 0.9028]	1.0769 [1.0742; 1.0797]
IN → DEAD	0.9482 [0.9463; 0.9501]	0.9331 [0.9049; 0.9622]	0.8279 [0.8196; 0.8364]
OUT → IN	1.0122 [1.0118; 1.0127]	0.8669 [0.8576; 0.8763]	0.9467 [0.9439; 0.9496]
OUT → DEAD	0.9423 [0.9402; 0.9444]	0.9422 [0.9109; 0.9746]	0.7879 [0.7815; 0.7942]

TABLE 5.18: Hazard Ratios and 95% confidence intervals for `age`, `gender`, and `charlson` as computed by the fully-parametric model with a Weibull baseline. The `gender` covariate refers to the men with respect to women.

Transition	rehab	it	n_com
IN → OUT	2.7120 [2.6676; 2.7570]	1.3283 [1.3144; 1.3425]	1.0133 [1.0101; 1.0164]
IN → DEAD	2.2040 [2.0230; 2.4012]	0.6608 [0.6326; 0.6903]	0.9358 [0.9229; 0.9488]
Transition	n_pro		
IN → OUT	1.7448 [1.7171; 1.7729]		
IN → DEAD	0.3671 [0.3532; 0.3815]		

TABLE 5.19: Hazard Ratios and 95% confidence intervals for `rehab`, `it`, `n_com`, and `n_pro` as computed by fully-parametric model with a Weibull baseline.

Transition	LOS	C07	C09
OUT → IN	1.0009 [1.0003; 1.0015]	1.0408 [1.0396; 1.0420]	1.0625 [1.0615; 1.0634]
OUT → DEAD	0.9784 [0.9769; 0.9799]	1.0676 [1.0621; 1.0731]	1.0943 [1.0901; 1.0985]
Transition	sum_PA		
OUT → IN	1.0257 [1.0251; 1.0263]		
OUT → DEAD	1.0715 [1.0674; 1.0756]		

TABLE 5.20: Hazard Ratios and 95% confidence intervals for `LOS`, `C07`, `C09`, and `sum_PA` as computed by the fully-parametric model with a Weibull baseline.

In Figures 5.9 and 5.10, we show a comparison between the observed survival curve, computed with the Kaplan-Meier estimator, and the estimated curves by the different models. Figure 5.9 refers to transitions departing from state IN. That is, for the discharge and for the death inside the hospital transitions. The time scale is reduced to 50 days due to an average `LOS` of 10 days which translates in the fact that almost every patient has already moved from the starting state after this period of time. Figure 5.10 refers to transitions departing from state OUT. That is, for the readmission into hospital and death outside of it. Even if, in this case, it is not true

that within 50 days every patient has died inside a hospital, the plot is consistent. The observed and the estimated curves simply keep on moving far apart till end of the follow-up time. The agreement between the estimated curve and the observed one is a proxy of the model performance. The curve computed by the semi-Markov is shown in black, the one computed by the exponential and Weibull models in red and green, respectively.

Consider the semi-Markov model. We can clearly observe a lack of fit for all the transitions. The model is not able to capture the observed pattern and this heavily affects the estimation of the survival probabilities at any time. Even when the sojourn time in the previous state is included so that the Markov assumption is relaxed, does not seem to sufficiently compensate. We still need to introduce new assumptions in order to improve the fit and hence obtaining a more valuable and reliable estimations.

Consider now the exponential model. We can observe a general improvement in the fit with respect to the semi-Markov approach. In particular, for the discharge and readmission to hospital transitions (see Figure 5.9), now the model correctly intercepts the observed behaviour so that we do not observe a total tumble at less than 5 days of `LOS`. The transition to death in hospital still suffers, even if we are now able to mimic the observed pattern for almost 20 days. The transition to death outside the hospital (see Figure 5.10) is not critical anymore in the sense that we do not have two almost parallel curves. Now we observe a certain amount of overlapping, particularly in the early times.

Finally, consider the Weibull model. With respect to the exponential baseline model, we do observe a further improvement in the discharge and in the death outside the hospital transitions (see Figure 5.9 and 5.10, respectively). For the former, now the model not only mimics the observed shape, but it also intercepts the empirical curve from the first day of follow-up time. For the latter, we observe an improved overlap between the two curves for almost all the follow-up time.

In order to quantify the deviation of the fitted survival curves from the empirical one, we compute a point-wise error $\delta(t)$ conditioned to the current state as given by:

$$\delta(t) = \delta_i(t) = |O_i(t) - E_i(t)|, \tag{5.1}$$

where i marks the current state, $O_i(t)$ is the value of the Kaplan-Meier curve at the time point t for the i-th state and $E_i(t)$ is the relative estimated values of the survival curve given by each model. In Figures 5.11 and 5.12, we show the results for transition departing from state IN and OUT, respectively. The optimum is given when $\delta(t) = 0$ and it is represented by the dotted-dashed grey line over zero. Any deviation from this, can be interpreted as a deviation of the fitted survival curve from the empirical one. It is clear how the semi-Markov model is the worst case with an error which is consistently higher than any other approach. Conversely, the Weibull model seems to outperform others, particularly in the transition to death from outside the hospital.

After the above considerations, we deem the Weibull model to be on top of all the others.

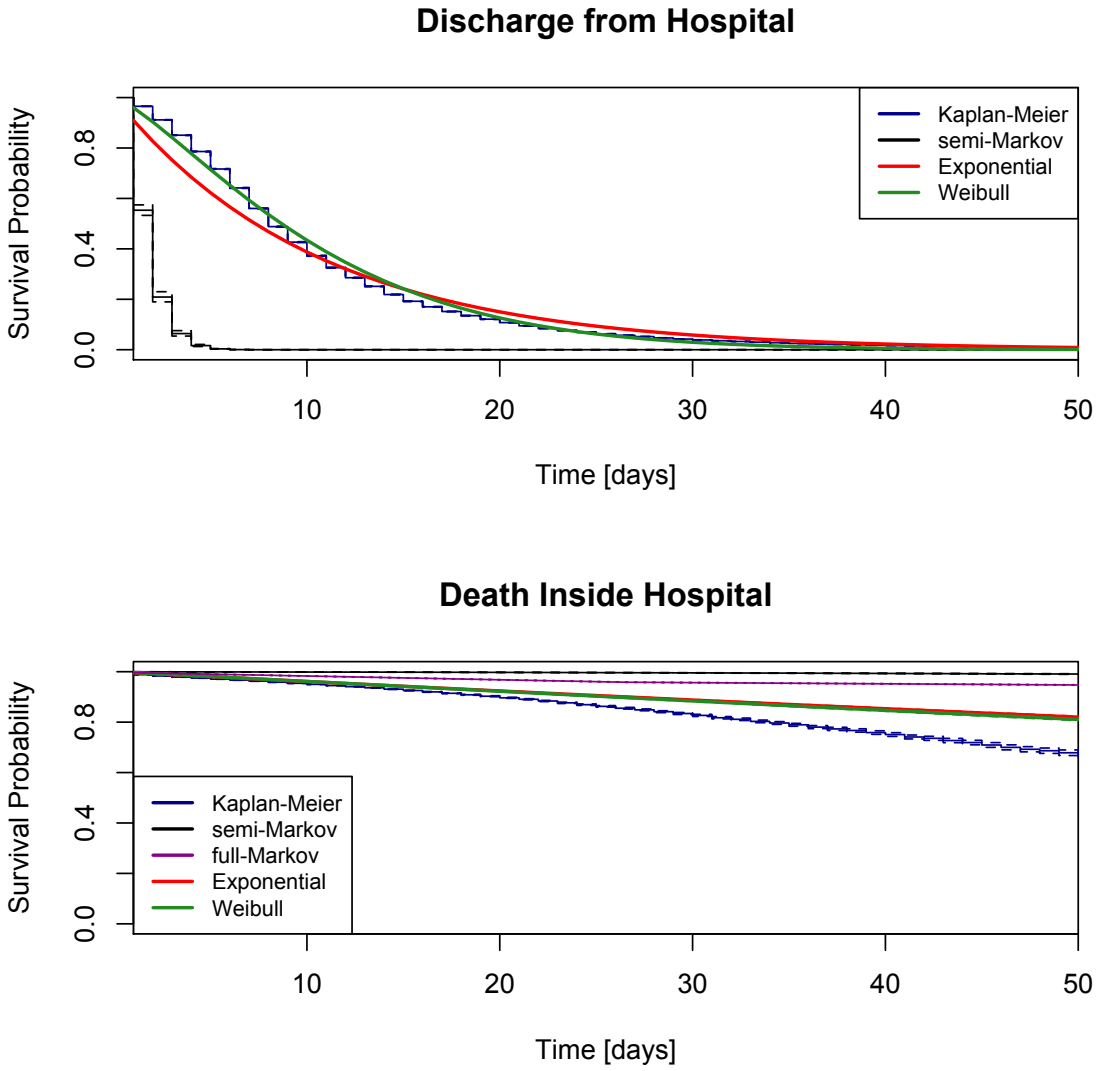

FIGURE 5.9: Comparison between the empirical survival curve (blue) with the estimated ones through a semi-Markov (black), full-Markov (magenta), exponential (red), and Weibull (green) models. Top panel refers to the transition from IN to OUT. Bottom panel refers to the transition from IN to DEAD.

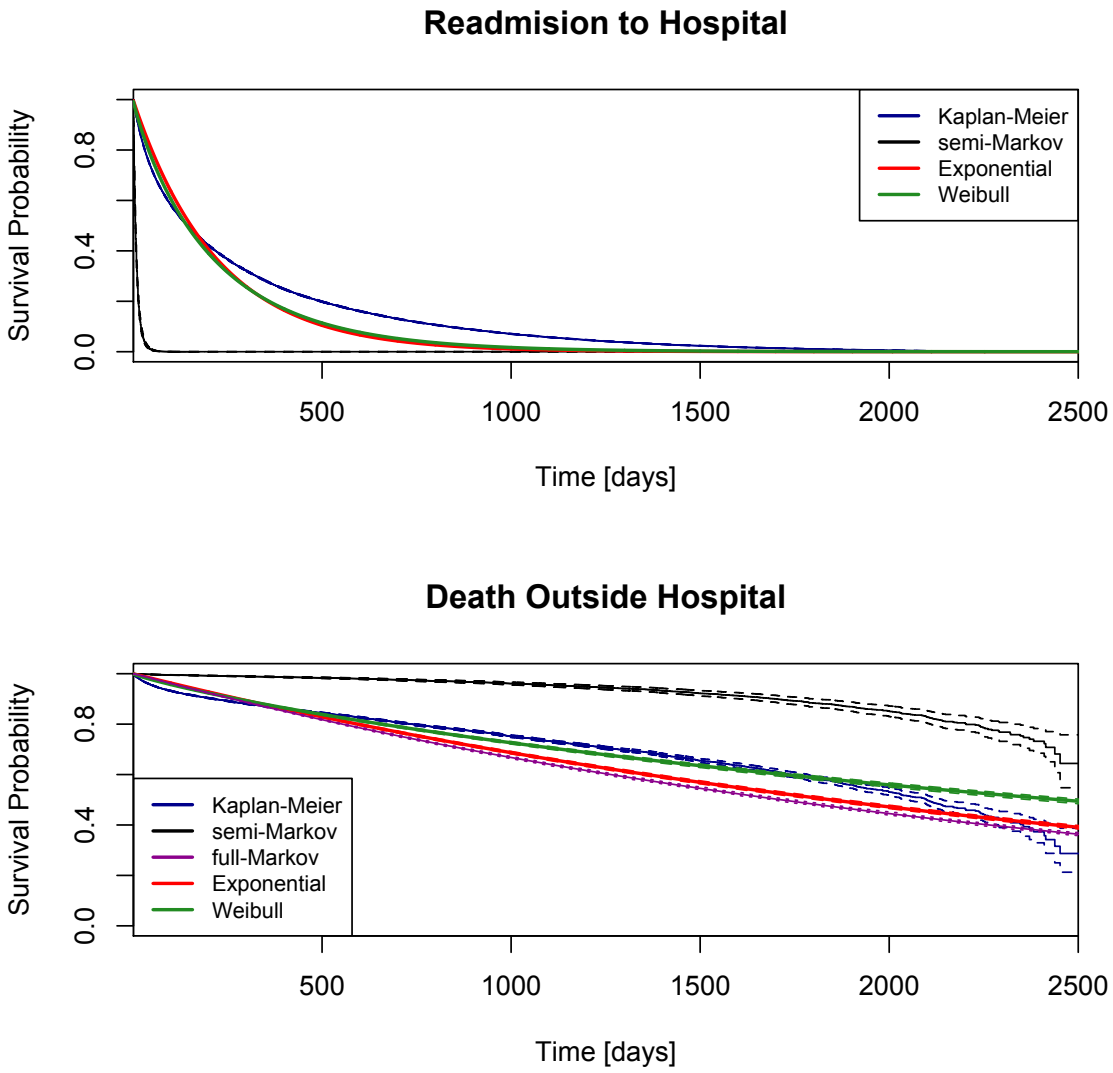

FIGURE 5.10: Comparison between the empirical survival curve (blue) with the estimated ones through a semi-Markov (black), full-Markov (magenta), exponential (red), and Weibull (green) models. Top panel refers to the transition from OUT to IN. Bottom panel refers to the transition from OUT to DEAD.

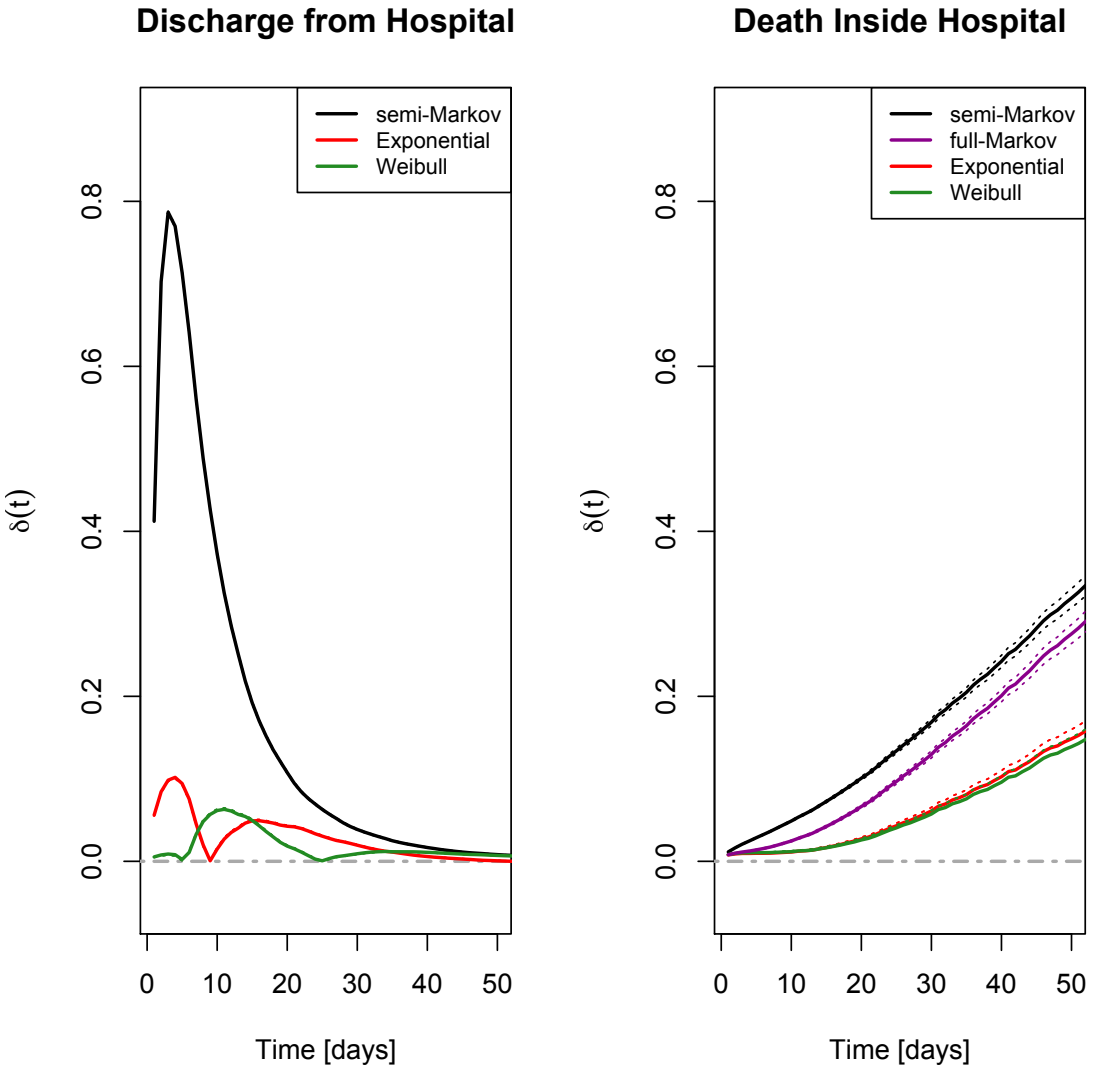

FIGURE 5.11: Models performances with respect to the empirical survival curve. Left panel refers to transition from IN to OUT. Right panel refers to transition from IN to DEAD. The dotted-dashed grey line represents the optimum. Semi-Markov error is reported in black, full-Markov in magenta, exponential in red, and Weibull in green.

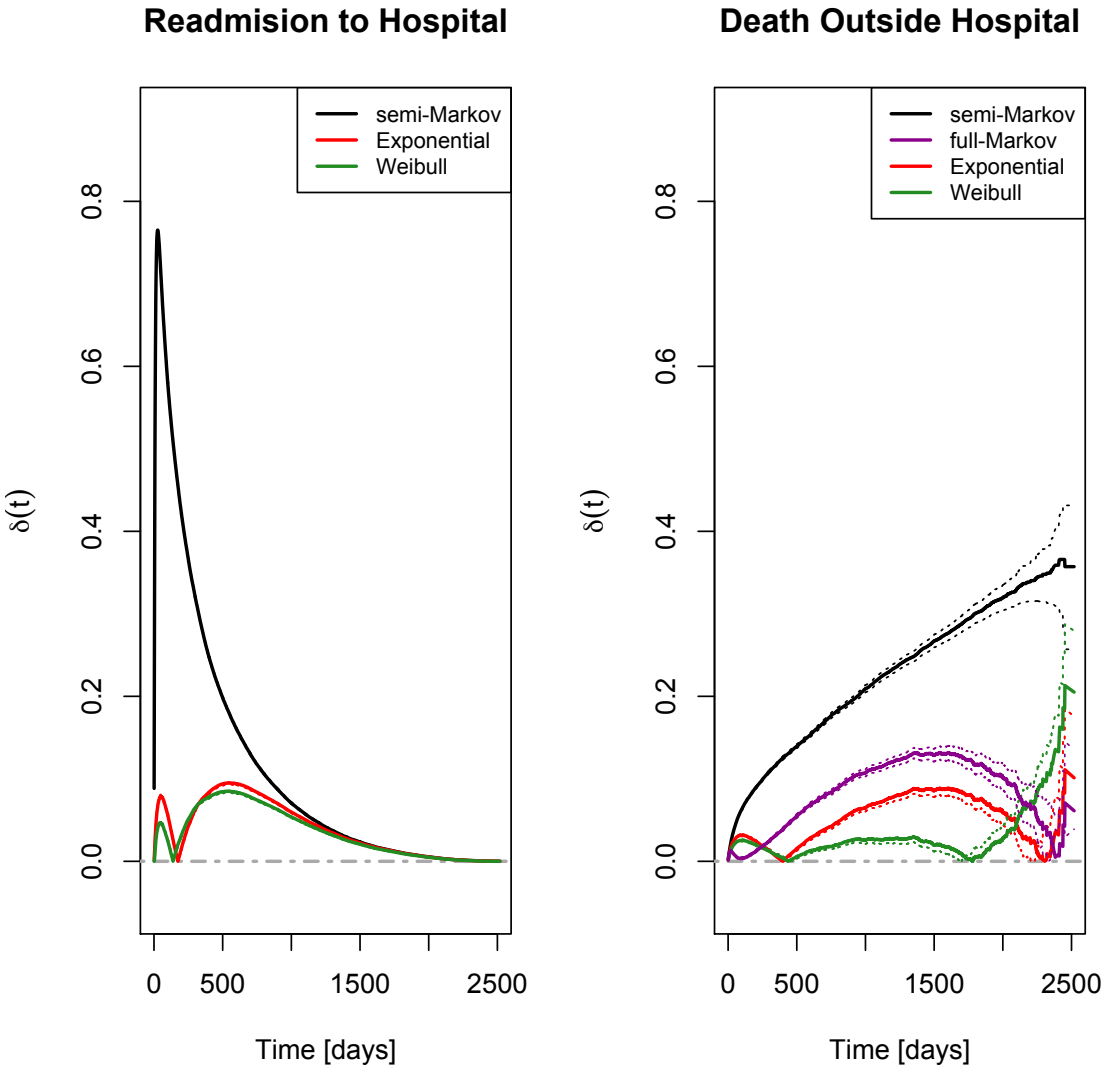

FIGURE 5.12: Models performances with respect to the empirical survival curve. Left panel refers to transition from OUT to IN. Right panel refers to transition from OUT to DEAD. The dotted-dashed grey line represents the optimum. Semi-Markov error is reported in black, full-Markov in magenta, exponential in red, and Weibull in green.

Chapter 6

Conclusions

This work has focused on the management and analysis of multiple healthcare administrative databases for which Lombardy Region (Italy) has granted us the access to. The present work marks the first Italian attempt which focuses on the acquisition, management and study of several data sources regarding the HF pathology collected by the public healthcare system of Lombardy Region in Italy. The main goal has been to define a consistent and semi-automatic procedure to import multiple highly dimensional and complex databases and to process their structure such that subsequent statistical analyses could be performed. The original and unprocessed databases count 370 thousands patients who generate more than 167 millions events and for the first time in Italy, we have been able to exploit the information coming from the drug prescriptions and outpatient cares histories to model the hospitalization pattern of the patient. This allowed us to to move the focus from a descriptive stand point of view to an inferential one. Moreover, this work studied the hospital admission-readmission process using different statistical approaches and assumptions. This allowed us to explore the HF patient's epidemiology and to profile the health service utilization over time. We also investigate variations in patient care according to geographic area, socio-demographic characteristics as well as other administrative and clinical variables.

During this thesis, there have been different steps in the workflow which marked the advancement. Most of the effort consisted in the development of procedures in order to import and process the data efficiently using `R`. In particular, we have been able to import and work with all the datasets in a single environment on a single machine. This has been achieved by widely using the call-by-reference paradigm, by avoiding internal copies of environment objects and by consistently adopting the most efficient method or objects during the coding phase.

The first part has focused on the data preprocessing and preparation such that all the datasets could be read with the correct longitudinal structure. Each information has been scanned and checked in order to avoid inconsistencies and to correct any issue due to errors occurred in the original imputation phase.

We have developed from scratch a new `R` package called `msmtools` [Grossetti, 2016] with the specific aim of modifying the longitudinal structure of the data into an enhanced version which we called augmented. This new format has been introduced to facilitate the modeling of such data under a multi-state models framework using the package `msm` [Jackson, 2011]. `msmtools` has several important characteristics. Among them, we highlight its speed, efficiency and generality. As we have described in Section 3.3, the function `augment()` is able to process almost 60 millions patients in less than an hour by running just in single core mode. This is very important because it allows the reshaping of very highly dimensional data in a reasonable amount of computational time. Moreover, the package is also very efficient in terms of memory consumption, so that a large dataset can be processed on a single consumer machine. At last, `msmtools` can work with any longitudinal dataset which satisfies certain prerequisites like the identification of time located events. It is also able to introduce even more general status types according to a combination of multiple information. For instance, we may want to extend the information of being inside or outside a hospital with more complex and general conditions like the type of hospital admission. This can be achieved seamlessly with internal parameter of `augment()`. The package also comes with two graphical tools which provide informal Goodness of Fit tests like

a comparison between the empirical and estimated survival curves as well as observed and estimated prevalences. Moreover, `survplot()` can predict the survival curves generated by a specific patient's profile through the model computed by `msm()`. It can also return the related data very efficiently so that for each provided profile it is possible to store point-wise estimation of the survival function.

The statistical analyses have used both survival and multi-state models. In particular, we are interested in estimating the probability that a given patient moves between different states. We have defined three different states which mark the patient when inside the hospital (state IN), outside the hospital (state OUT) and when the patient dies (state DEAD). The first two transitions are transient, while the third defines the absorbing state. We have build a bi-directional illness-death model in which a patient can move back and forth through the IN and OUT states. We have then four possible transitions defined by the transition matrix Q_0:

$$Q_0 = \begin{pmatrix} 0 & q_{IN \to OUT} & q_{IN \to DEAD} \\ q_{OUT \to IN} & 0 & q_{OUT \to DEAD} \\ 0 & 0 & 0 \end{pmatrix}, \tag{6.1}$$

For this structure, we have made different assumptions which lead to four different models. In particular, all the models are considered to be time-homogeneous, so that the transition matrix is assumed constant with time. Then we have assumed a Markov process for the first model implying that the future evolution of the process depends only on the current state (see Subsections 4.2.1 and 5.2.1 for theoretical definition and for results, respectively). In other words, we do not consider anymore the full history of the process, but we deem that the last state provides sufficient information for the estimation. The model is assessed through the function `msm()` in the homonymous package and it has been run on the augmented database as computed by the function `augment()` in the package `msmtools`. A second model, called semi-Markov, relaxes the Markov assumption by including the sojourn time length in the last state into the estimation (see Subsections 4.2.3 and 5.2.2 for theoretical

definition and for results, respectively). The model is assessed through the function `coxph()` in the package `survival` [Therneau, 2015] over the data structure computed by the functions `msm2Surv()` and `expand.covs()` in the packages `msm` and `mstate` [de Wreede et al., 2011], respectively. These first two models fall in the semi-parametric approach since both of them are based on the Cox regression model which does not assume any distribution for the baseline hazard function which is then estimated non-parametrically. The third and fourth models, instead, assume a probability distribution for the baseline. In particular, we have considered an exponential and a Weibull models (see Subsections 4.1.6 and 4.3.1 for theoretical definition and 5.2.3 for results, respectively). Both are assessed through the function `flexsurvreg()` in the package `flexsurv` [Jackson, 2016] and are run with over the same data structure as the semi-Markov model.

All the models have made use of the covariates `age`, `gender`, and `charlson` over all the transitions; `rehab`, `it`, `n_com`, and `n_pro` over the transitions departing from the state IN; `LOS`, `C07`, `C09`, and `sum_pa` over the transitions departing from the state OUT.

In terms of parameters estimation, the overall behaviour of the four models is comparable. First of all, we have confirmed a clear impact of the brand new set of covariates which provide drug prescriptions and outpatient cares history. Pursuing a correct therapy, both in terms of drug prescriptions and outpatient cares, positively affects the general condition of a patient. They decrease the risk of hospital readmission and death with all the hazard ratios being smaller than 1. In particular, ACE-inhibitors have a slightly higher prevention rate with respect to beta blocking agents. Even if we do not possess the exact data related to prescribed drugs consumption and the outpatient cares taken, this result can be considered as a therapy compliance proxy. A deeper study of these types of covariate is fundamental to investigate more characteristics like the time duration of drug prescriptions. For instance, a correct acceptance of therapy thus a prolonged compliance with time, might suggest an even more positive effect over the transitions to death. Being a man increased the general

risk of transitioning and the natural aging effect is present and positively affects the probability of dying. Also the Charlson index correctly reflects the patient's condition with an increasing risk for higher index values. The rehabilitation and intensive therapy units passage decrease the chances of being discharged. Both the mentioned cares are typically more delicate and require a longer sojourn in the hospital and at the same time, they help the patient by improving his/her general conditions. If a rehabilitative admission positively acts also over a transition to death, this is not true when a patient is admitted in the intensive therapy unit. This is something we might consider acceptable since a patient who undergoes this type of therapies is typically in more severe conditions. The number of comorbidities and surgical procedures behave the same way. Both prolong the sojourn time in the hospital, that is the chances of being discharged from hospital are lower, but increase the probability of dying. In particular, having any of the surgical procedures has an impact almost three times stronger than having any comorbidity. For what concern the covariates adopted to model transitions departing from state OUT, we have observed a general decrease in the risk. The only exception is given by the `LOS` which comprehensibly increases the risk of dying.

Of the four adopted models, the one which better captures the patient's process is the fully-parametric one with a Weibull baseline. In particular, this is very effective in intercepting the observed risk of being discharged from hospital as well as the risk of dying outside of it. Moreover, the error is consistently lower than that of the other model, except for moderate time intervals.

This work is a first attempt at modeling the process of hospital admissions using the drug prescriptions and outpatient cares histories. From a statistical point of view, the models implemented form a first framework onto which building their development. This can be achieved in several different ways. For instance, an extension of the covariates structure by including more information related to different dugs and outpatient cares which, up to now, are restricted to cardiological ones. A change in the statistical approach could also be considered. For instance, we may include

Hidden Markov Models and Point Processes. From a computational point of view, there is still room for the development of `msmtools` as discussed in Subsection 3.3.4. Moreover, some effort could be dedicated to the development of a package which acts as a wrapper of several others from a data restructuring point of view. Also, some plotting methods have been build based on how the different estimation functions internally work. Some ideas have already been formalized, but still a consistent shape has not been achieved yet.

Appendix A

Side Projects

Beside the main topic and work of this thesis, several side projects have been carried out in parallel during the past three years. The works described in Sections A.1, A.2, and A.3 have been carried out in collaboration with different clinicians, physiotherapists, biologists belonging to the *Scientific Institute of Lumezzane - Fondazione Salvatore Maugeri, IRCCS* while the work described in Section A.4 is in collaboration with *MOXOFF - Mathematics for Innovation*, a spinoff of the MOX laboratory at Politecnico di Milano. In this chapter, we are going to present them and briefly discuss their primary goals.

A.1 Effects of Tele Assistance

Over the last years, the interest in clinical interventions like Tele Assistance (TA) has considerably increased [Goldstein and O'Hoski, 2014]. In particular, TA applied to chronic diseases is consider to be a game changer in terms of patient's and costs management.

The present work focuses on the evaluation of the effects of addition of long-term TA to patients affected with hypercapnic Coronary Obstructive Pulmonary Disease (COPD). In particular, the study investigates what are the benefits in case

Non-Invasive Ventilation (NIV) is adopted. This is a retrospective analysis of data regarding hypercapnic COPD affected patients under long-term TA. The data have been collected as a randomized study [Vitacca et al., 2009]. Patients have been evaluated for at least 4 years including long-term TA with and without home night NIV and with at least one hospital admission for respiratory illness in the previous years before randomization of the original study. We considered three exclusion criteria:

1. patients already randomized or with no working home phone, with a nursing home residence or with no caregiver to facilitate phone contacts;

2. no COPD diagnosis;

3. presence of tracheostomy.

Those patients eligible for the study, have been admitted to hospital to begin a 4-week rehabilitation plan. Long-term Oxygen Therapy (OT) has been prescribed for all patients according to the Italian Guidelines [Murgia et al., 2004] and home NIV has been prescribed as well at least one year before the hospital admission, when available.

Several baseline data have been recorded for all patients. Primary outcome measures are the time to the first exacerbation and hospitalization in the following 12 months after discharge and the 12-months survival probability. KM method has been used to assess first exacerbation and hospitalization as well as death. These quantities have been also studied through a Cox PH regression model.

This study shows the usefulness of adding TA to long-term care plans in hypercapnic COPD patients under long-term OT with or without night NIV. For a detailed discussion of the results, we refer to the work of Vitacca et al. [2016].

A.2 Recovery After Rotator Cuff Repair

This work aims to verify if the concomitant biceps surgery prejudices the shoulder functionality during the short-term period in rotator cuff repair patients. Rotator cuff tear is a common cause of disability of the shoulder and, among several diseases of the upper limbs, is considered to be the one with higher associated costs whether from a medical, surgical, insurance and management point of views. Surgical treatment and repair of chronic condition is thus indicated when conservative treatment fails. The goals of rotator cuff tear surgery are to decrease pain, improve functionality, and prevent subsequent extension of the defect [Pai and Lawson, 2001]. This approach seems to be effective and to provide very good results in 93.3% of cases [Redziniak et al., 2009].

The present work has been conceived as a prospective longitudinal study carried out on 101 patients who underwent surgery for rotator cuff repair. Observations occurred at the admission (T_0), at the end of the post-surgical rehabilitation period (T_1) and then after six months from the surgery (T_2). Among the 101 patients, 25 underwent rotator cuff repair and additional tendon biceps surgery (ABS group) while 76 rotator cuff repair only (RCR group).

Final score, efficiency and effectiveness in Constant Scores have been considered as outcome measures [Shah et al., 1990]. Several other quantities have been recorded. Among others: the comorbidity index of Cumulative Illness Rating Scale (CIRS) [Parmelee et al., 1995], the Range of Motion (ROM) of shoulder, the University of California at Los Angeles (UCLA) shoulder score [Ellman et al., 1986], the Constant Scale and Pain [Constant and Murley, 1987; DeLoach et al., 1998].

A 2-way mixed ANOVA has been implemented to investigate the effect of time progression in interaction with patient's group for these variables: Constant Score, Pain, ROM, and UCLA. The magnitude of effects, where significative, has been assessed through the η^2 and Cohen's d [Cohen, 1969] effect sizes.

Of the initial 101 patients, 8 have been lost at 6 months post-surgery so the final analyses have been carried out on a sample of 93 patients. The sample size has been

deemed adequate to highlight significant differences between groups in Constant Final score with a power $\gamma = 0.85$ at a significance level of $\alpha = 0.05$.

In this work we have been able to point out that both RCR and ABS patients increased Constant Scores at T_1 and at T_2 with respect to T_0, though ABS ones showed, intuitively, lower scores than RCR. In general, poorer functionality has been observed in ABS group in the different outcomes meaning that ABS affects the rehabilitation program thus slowing down functional recovery of patients. Biceps surgery seems to be an important predictor of the shoulder functionality both at T_1 and at T_2. In literature, no evidence of this has been found.

Further details can be found in the work of Gialanella et al. [2016].

A.3 Bioelectrical Impedance Analysis

The measurement of body composition in terms of Fat-Free Mass (FFM) and Fat Mass (FM) is of strategic importance in nutrition assessment. There exist a wide variety of techniques all of which are considered standards for their reliability. Typically, techniques such in vivo neutron activation, isotopic dilution and hydrostatic weighing are considerably expensive, they require a high level of expertise and involve the patient in terms of cooperation time. A different approach to specifically evaluate FFM and Total Body Water (TBW) is to measure the electrical properties of biological tissues. The technique is called Bioelectrical Impedance Analysis (BIA) and presents several advantages over the other methods. Among others, we highlight its safeness, portability, ease of use and cost-friendliness. A good review of the working principles and applications can be found in the work of Kushner [1992].

For the purpose of the present study, let us recall some concepts of BIA. It is possible to measure the variations the amount of blood flowing through an organ by tracking the electrical changes in the section of the body under study. The impedance of a isotropic conductor can be written as follows:

$$Z = \rho \frac{L}{A}, \tag{A.1}$$

where Z is the impedance [ohm], ρ is the specific resistivity of the medium [ohm $\cdot$ cm], L is the conductor length [cm] and A is the cross-sectional area [cm^2]. Equation (A.1) can be rearranged to explicitly solve for the volume V [cm^3] by multiplying everything for L/L as follows:

$$V = \rho \frac{L^2}{Z}. \tag{A.2}$$

Solving equation (A.2), that is the electrical volume, allows to measure the FFM and TBW. Moreover, there is a direct relation between impedance and resistance which helps us out to use them interchangeably. In general, BIA is used as a tool to describe the impedance of a patient through its components given by the resistance R and the reactance X of the conducting substance. This allows us to write the following equation:

$$Z = R + X = R + (X_L + X_C)\,. \tag{A.3}$$

where X_L is the inductance and X_C is the capacitance. Total body impedance is then a combination of resistance and reactance across biological tissues.

The study counts a total of 408 patients with different pathologies. The full analysis will be carried out on two groups of patients: the former is made of 166 patients affected by CHF and the latter of 64 patients affected by cardiomyopathy for a total of 230 patients. We have registered several types of data like haematochemicals, functional, and of course BIA data.

The aims of this work are multiple. The main objective is to describe the two population through a classification based on the BIA data, that is based on reactance and resistance data. At the present stage, a *k-means* algorithm has been implemented [Hartigan and Wong, 1979]. A simulation study has also been included to specifically evaluate the performance of the algorithm. The following quality indexes are adopted:

the *silhouette* index [Rousseeuw, 1987] which provides a graphical tool to interpret and validate the cluster analysis, the *Caliński-Harabasz* index [Caliński and Harabasz, 1974], and the *C-index* [Hubert and Schultz, 1976]. The procedure is interactive and fast and consists in running a k-means multiple times by changing the initial random seed at each iteration and the number of cluster given in input. The simulation is iteratively repeated for at most 1000 times with the number of clusters ranging from 2 to at most 20. We always obtain a consistent general behaviour with an increasing factor of fluctuation around the mean as the number of clusters grow.

To detect the number of cluster which best catches the variance, we use the aforementioned indexes. For each of them, we either maximize or minimize an objective function. In particular, for the silhouette index we look for the maximum of the following function, as derived from the original paper of Rousseeuw:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad \text{with} -1 \leq s(i) \leq 1, \tag{A.4}$$

where $a(i)$ is the average dissimilarity of the i-th element to all other objects of cluster A, $b(i)$ is the minimum of the average length from i to all the objects in cluster C such that $b(i) = \min_{C \neq A} d(i, C)$.

The Caliński-Harabasz index is the ratio of the between-cluster and within-cluster dispersion. It should look familiar as this is actually the F-value[1] of a one-way ANOVA with K representing the number of factor levels. The criterion has proven to work well in many situations. It also also shows a consistent robustness across the varying number of clusters. Moreover, it is not affected by a low number of clusters in contrast with the Duda and Hart index as discussed in the work of [Milligan and Cooper, 1985]. We show below the form of the index below:

$$\mathcal{CH} = \arg\max \left\{ \frac{\text{BGSS}/(K-1)}{\text{WGSS}/(N-K)} \right\} = \arg\max \left\{ \frac{N-K}{K-1} \frac{\text{BGSS}}{\text{WGSS}} \right\}, \tag{A.5}$$

[1] Often it is also called pseudo F-statistic due to its similarity with the statistic used in the more classical F-test.

where BGSS is the between-group dispersion defined as the dispersion of the barycenters G^k of each cluster with respect to the barycenter G of the whole set of data. It is given by the trace of the between-group scatter matrix $\mathrm{BG} = B^t B$ as follows:

$$\mathrm{BGSS} = Tr(\mathrm{BG}) = \sum_{k=1}^{K} n_k \|G^{\{k\}} - G\|^2, \tag{A.6}$$

where WGSS is the within-group dispersion. For each cluster C_k we introduce the within-group scatter matrix $\mathrm{WG}^{\{k\}}$. If $\mu^{\{k\}}$ designates the barycenter of the observations in cluster k and $X^{\{k\}}$ is the matrix formed by the centered vectors $v_j^{\{k\}} = V_j^{\{k\}} - \mu_j^{\{k\}}$, then the scatter matrix is defined by $\mathrm{WG}^{\{k\}} = X^{t,\{k\}} X^{\{k\}}$. WGSS is defined as the sum of the trace of the scatter matrix as follows:

$$\mathrm{WGSS} = \sum_{k=0}^{K} Tr(\mathrm{WG}^{\{k\}}) = \sum_{k=0}^{K} \sum_{i \in I_k} \|M_i^{\{k\}} - G^{\{k\}}\|^2. \tag{A.7}$$

Finally, the C-index is defined as:

$$C = \arg\min \left\{ \frac{S - S_{min}}{S_{max} - S_{min}} \right\}, \tag{A.8}$$

where S is the sum of distances over all pairs of objects which form the same cluster, n is the number of those pairs and S_{min} is the sum of the n smallest distances if all pairs of objects are considered. Likewise S_{max} is the sum of the n largest distances out of all pairs. The C-index is limited to the interval $[0, 1]$ and we want to take its minimum.

A second aim is to study the correlation of the Human Serum Albumin (HSA) with mortality. HSA is a protein which is very abundant in human blood plasma and is fundamental in the regulation of blood plasma colloid osmotic pressure. It acts as a carrier protein for a wide range of endogenous molecules including hormones, fatty acids, and metabolites, as well as exogenous drugs. It is known that an inverse

correlation between the HSA concentration and the mortality risk does exist [Alderson et al., 2003] in patients affected by acute and chronic illness. The study tries to provide a snapshot of the protein metabolism impairment in CHF patients referred to a rehabilitative center to perform cardiac rehabilitation. The 166 CHF patients have been selected according to different HSA values which identify three different sub-populations: a normal one, a sarcopenic one (i.e patients are affected by sarcopenia which is a degenerative loss of skeletal muscle mass), and a cachectic one (i.e. patients affected by cachexia which is a loss of weight, muscle atrophy as well as a loss of appetite).

At the present stage, the simulation study has been correctly implemented and the assessment step needs to be taken. The study of HSA correlations is currently ongoing and requires a further preprocessing step due to the high impact of missing data in the dataset.

A.4 Customer Churn in a No-Profit Setting

The analysis of the customer base is very challenging and requires a robust set of tools in order to obtain consistent results. The world of customer base analysis, which in the end becomes in the capacity of evaluating whether a customer is profitable or not, is amazingly widespread. A very solid definition defines a profitable customer as "a person, household, or company whose revenues over time exceed, by an acceptable amount, the company costs of attracting, selling, and servicing that customer" [Kotler and Armstrong, 2010]. This excess is called Customer Lifetime Value (CLV) and nowadays is a pillar in many marketing and budgeting decisions within companies.

The study of the relation between a customer and a firm can be approached in different ways. For instance, in this project we adopt the general framework we have depicted in Chapter 4. We are interested in modeling the behaviour of a customer along time in the context of no-profit associations. In this framework, an individual starts a subscription with a given association which imposes the payment of recurrent

fee. The payment frequency depends on the type of contract and can be monthly, quarterly, every six month or yearly. In principle, a correct behaviour is given by no interruptions in the payment flow. Of course, this happens rarely because, for whatever reason, a payment or even a series of payments can be skipped. The individual is now in a grey and misty situation because we do not know if he/she intends to renew, actually reprise, the subscription or if he/she has completely abandoned it.

One of the main interest of no-profit associations is to maintain their donors, especially the ones with at least one subscription. The aim is to compute a probability of churn for each donor or cluster of them at a given point in time and highlight the ones that show a risk of leaving the association. Moreover, the projects also aims at implementing all the statistical models into a web-based platform which helps the associations in managing the donors and the relative fundraising campaigns.

We have analyzed multiple datasets all in the longitudinal format. The main dataset has 1,226,429 observations with 42 variables though several preprocessing procedures have been carried out. For instance, we focus on the relation between a given donor and his/her donations' pattern by selecting just one and only one subscription which registered the maximum number of donations. Moreover, we have selected only women and men thus excluding families, foundations and firms. We have considered only donor with a registered date of birth and we have excluded all the events with time inconsistencies like the date of the end of the subscription occurring before the subscription began. The final dataset consists of 619,794 events generated by 30,707 donors.

MSMs help in the assessment of the movement between possible intermediate states, beside the active and the absorbing ones. The model is bi-directional with $\mathcal{S} = \{1, \ldots, 3\}$. The states are:

- active: the donor is following the correct payment schedule and the amount of money is greater than zero;
- waiting: when a donor fails to respect the payment schedule, reaches this state;

- inactive: if a donor signs the end of the subscription, then enters in the absorbing state.

For this model, we assume a time-homogeneous full Markov process and we estimate the transition probabilities using the `msm` package. Covariates are transition specific and each transition is defined by a semi-parametric Cox model.

Up to now, we are able to produce a survival curve for each donor's profile easily and in very fast way for a specific association. The project is at its first lights and there is still much to do, both in terms of database management, variables selection and model improvements. For instance, a general data import and variables creation is under development. This will ensure the possibility to independently work with different associations whose databases have different structures. The variables selection process will be finalized once all the databases will be well defined. The statistical models will also be modified with respect to the new set of variables. However, the general guideline is to exploit the work of this thesis and adapt the models accordingly.

References

Aalen, O. O. and Johansen, S. (1978). An empirical transition matrix for non-homogeneous markov chains based on censored observations. *Scandinavian Journal of Statistics*, pages 141–150.

AHRQ (2015). *Quality Indicators, Heart Failure Mortality Rate, Technical Specifications, version 5.0, 2015;*. `http://www.qualityindicators.ahrq.gov/Downloads/Modules/IQI/V50/TechSpecs/IQI_16_Heart_Failure_Mortality_Rate.pdf`.

Alderson, P., Bunn, F., Lefebvre, C., Li, W., Li, L., Roberts, I., and Schierhout, G. (2003). Human albumin solution for resuscitation and volume expansion in critically ill patients. *The Cochrane database of systematic reviews*, (4):CD001208–CD001208.

Analytics, R. and Weston, S. (2015a). doparallel: Foreach parallel adaptor for the parallel package. *R package version 1.0.10*, 1(8).

Analytics, R. and Weston, S. (2015b). Foreach: provides foreach looping construct for r. *R package version 1.4.3*, 1(3).

Andersen, P. and Keiding, N. (2002). Multi-state models for event history analysis. *Statistical methods in medical research*, 11(2):91–115.

Bijleveld, C. C., Leo, J. T., Leo, J., Mooijaart, A., Van Der Van Der, W. A., Van Der Leeden, R., Van Der Burg, E., et al. (1998). *Longitudinal data analysis: Designs, models and methods*. Sage.

Bleumink, G. S., Knetsch, A. M., Sturkenboom, M. C., Straus, S. M., Hofman, A., Deckers, J. W., Witteman, J. C., and Stricker, B. H. C. (2004). Quantifying the heart failure epidemic: prevalence, incidence rate, lifetime risk and prognosis of heart failure. *European heart journal*, 25(18):1614–1619.

Blondé-Cynober, F., Morineau, G., Estrugo, B., Fillie, E., Aussel, C., and Vincent, J.-P. (2011). Diagnostic and prognostic value of brain natriuretic peptide (bnp) concentrations in very elderly heart disease patients: specific geriatric cut-off and impacts of age, gender, renal dysfunction, and nutritional status. *Archives of gerontology and geriatrics*, 52(1):106–110.

Bottle, A., Aylin, P., and Bell, D. (2014). Effect of the readmission primary diagnosis and time interval in heart failure patients: analysis of english administrative data. *European journal of heart failure*, 16(8):846–853.

Braunstein, J. B., Anderson, G. F., Gerstenblith, G., Weller, W., Niefeld, M., Herbert, R., and Wu, A. W. (2003). Noncardiac comorbidity increases preventable hospitalizations and mortality among medicare beneficiaries with chronic heart failure. *Journal of the American College of Cardiology*, 42(7):1226–1233.

Broyden, C. G. (1970). The convergence of a class of double-rank minimization algorithms 2. the new algorithm. *IMA Journal of Applied Mathematics*, 6(3):222–231.

Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Castañeda, J. and Gerritse, B. (2010). Appraisal of several methods to model time to multiple events per subject: Modelling time to hospitalizations and death. *Revista Colombiana de Estadística*, 33(1):43–61.

Ceia, F., Fonseca, C., Mota, T., Morais, H., Matias, F., Sousa, A., and Oliveira, A. G.

(2002). Prevalence of chronic heart failure in southwestern europe: the epica study. *European journal of heart failure*, 4(4):531–539.

Charlson, M. E., Pompei, P., Ales, K. L., and MacKenzie, C. R. (1987). A new method of classifying prognostic comorbidity in longitudinal studies: development and validation. *Journal of chronic diseases*, 40(5):373–383.

Cohen, J. (1969). Statistical power analysis for the behavioral sciences. *New York Academic Press*.

Constant, C. and Murley, A. (1987). A clinical method of functional assessment of the shoulder. *Clinical orthopaedics and related research*, 214:160–164.

Cowie, M. (2003). Estimating prognosis in heart failure: time for a better approach. *Heart*, 89(6):587–588.

Cox, D. (1979). A note on the graphical analysis of survival data. *Biometrika*, 66(1):188–190.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 187–220.

Cox, D. R. and Miller, H. D. (1977). *The theory of stochastic processes*, volume 134. CRC Press.

Dabrowska, D. M., Sun, G.-W., and Horowitz, M. M. (1994). Cox regression in a markov renewal model: an application to the analysis of bone marrow transplant data. *Journal of the American Statistical Association*, 89(427):867–877.

de Wreede, L., Fiocco, M., Putter, H., et al. (2011). mstate: an r package for the analysis of competing risks and multi-state models. *Journal of Statistical Software*, 38(7):1–30.

DeLoach, L. J., Higgins, M. S., Caplan, A. B., and Stiff, J. L. (1998). The visual analog scale in the immediate postoperative period: intrasubject variability and correlation with a numeric scale. *Anesthesia & Analgesia*, 86(1):102–106.

Dowle, M., Short, T., Lianoglou, S., with contributions from R. Saporta, A. S., and Antonyan, E. (2014). *data.table: Extension of data.frame*. `http://CRAN.R-project.org/package=data.table`.

Eddelbuettel, D. (2013). *Seamless R and C++ integration with Rcpp*. Springer.

Eddelbuettel, D., François, R., Allaire, J., Chambers, J., Bates, D., and Ushey, K. (2011). Rcpp: Seamless r and c++ integration. *Journal of Statistical Software*, 40(8):1–18.

Ellman, H., Hanker, G., and Bayer, M. (1986). Repair of the rotator cuff. end-result study of factors influencing reconstruction. *J Bone Joint Surg Am*, 68(8):1136–1144.

Enjuanes, C., Klip, I. T., Bruguera, J., Cladellas, M., Ponikowski, P., Banasiak, W., Van Veldhuisen, D. J., Van Der Meer, P., Jankowska, E. A., and Comín-Colet, J. (2014). Iron deficiency and health-related quality of life in chronic heart failure: results from a multicenter european study. *International journal of cardiology*, 174(2):268–275.

Eschenhagen, T., Force, T., Ewer, M. S., Keulenaer, G. W., Suter, T. M., Anker, S. D., Avkiran, M., Azambuja, E., Balligand, J.-L., Brutsaert, D. L., et al. (2011). Cardiovascular side effects of cancer therapies: a position statement from the heart failure association of the european society of cardiology. *European journal of heart failure*, 13(1):1–10.

Fletcher, R. (1970). A new approach to variable metric algorithms. *The computer journal*, 13(3):317–322.

Gagne, J. J., Glynn, R. J., Avorn, J., Levin, R., and Schneeweiss, S. (2011). A combined comorbidity score predicted mortality in elderly patients better than existing scores. *Journal of clinical epidemiology*, 64(7):749–759.

Gavrielov-Yusim, N. and Friger, M. (2013). Use of administrative medical databases in population-based research. *Journal of epidemiology and community health*, pages jech–2013.

Gentleman, R., Lawless, J., Lindsey, J., and Yan, P. (1994). Multi-state markov models for analysing incomplete disease history data with illustrations for hiv disease. *Statistics in medicine*, 13(8):805–821.

Gerber, Y., Weston, S. A., Redfield, M. M., Chamberlain, A. M., Manemann, S. M., Jiang, R., Killian, J. M., and Roger, V. L. (2015). A contemporary appraisal of the heart failure epidemic in olmsted county, minnesota, 2000 to 2010. *JAMA internal medicine*, 175(6):996–1004.

Gialanella, B., Grossetti, F., Mazza, M., and Danna, L. (2016). Functional recovery after rotator cuff repair: the role of biceps surgery. *Journal of Sport Rehabilitation (in press)*.

Gill, R. D. (1980). Nonparametric estimation based on censored observations of a markov renewal process. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 53(1):97–116.

Goff, D. C., Pandey, D. K., Chan, F. A., Ortiz, C., and Nichaman, M. Z. (2000). Congestive heart failure in the united states: is there more than meets the i (cd code)? the corpus christi heart project. *Archives of internal medicine*, 160(2):197–202.

Goldfarb, D. (1970). A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26.

Goldstein, R. S. and O'Hoski, S. (2014). Telemedicine in copd: time to pause. *CHEST Journal*, 145(5):945–949.

Gomatam, S., Carter, R., Ariet, M., and Mitchell, G. (2002). An empirical comparison of record linkage procedures. *Statistics in medicine*, 21(10):1485–1496.

Grimes, D. A. (2010). Epidemiologic research using administrative databases: garbage in, garbage out. *Obstetrics and Gynecology*, 116(5):1018–1019.

Grossetti, F. (2016). Building augmented data for multi-state models: the msmtools package (in progress).

Grossetti, F., Ieva, F., and Paganoni, A. (2016). A multi-state approach to patients affected by chronic heart failure: the value added by administrative data (in review). *Health Care Management Science*.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

Hawkins, N. M., Virani, S., and Ceconi, C. (2013). Heart failure and chronic obstructive pulmonary disease: the challenges facing physicians and health services. *European heart journal*, 34(36):2795–2807.

Hoover, K., Tao, G., Kent, C., and Aral, S. (2011). Epidemiologic research using administrative databases: garbage in, garbage out. *Obstetrics and Gynecology*, 117(3):729–730.

Hosmer Jr, D. W. and Lemeshow, S. (1999). Applied survival analysis: Regression modelling of time to event data (1999).

Hougaard, P. (1999). Multi-state models: a review. *Lifetime data analysis*, 5(3):239–264.

Howe, H. L., Lake, A. J., and Shen, T. (2007). Method to assess identifiability in electronic data files. *American Journal of Epidemiology*, 165(5):597–601.

Hubert, L. and Schultz, J. (1976). Quadratic assignment as a general data analysis strategy. *British journal of mathematical and statistical psychology*, 29(2):190–241.

Ieva, F., Gale, C., and Sharples, L. (2014). Contemporary roles of registries in clinical cardiology: when do we need randomized trials? *Expert review of cardiovascular therapy*, 12(12):1383–1386.

Investigators, S. (1992). Effect of enalapril on mortality and the development of heart failure in asymptomatic patients with reduced left ventricular ejection fractions. *N Engl j Med*, 1992(327):685–691.

Jackson, C. (2000). *Statistical models for the latent progression of chronic diseases using serial biomarkers*. PhD thesis, University of Cambridge.

Jackson, C. (2011). Multi-state models for panel data: the `msm` package for R. *Journal of Statistical Software*, 38(8):1–29.

Jackson, C. (2016). flexsurv: a platform for parametric survival modelling in r. *Journal of Statistical Software*, 70(8):1–33.

Kalbfleisch, J. D. and Prentice, R. L. (2011). *The statistical analysis of failure time data*, volume 360. John Wiley & Sons.

Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481.

Kotler, P. and Armstrong, G. (2010). *Principles of marketing*. pearson education.

Kushner, R. F. (1992). Bioelectrical impedance analysis: a review of principles and applications. *J Am Coll Nutr*, 11(2):199–209.

Lagakos, S. W., Sommer, C. J., and Zelen, M. (1978). Semi-markov models for partially censored data. *Biometrika*, 65(2):311–317.

Lee, D. S., Donovan, L., Austin, P. C., Gong, Y., Liu, P. P., Rouleau, J. L., and Tu, J. V. (2005). Comparison of coding of heart failure and comorbidities in administrative and clinical data for use in outcomes research. *Medical care*, 43(2):182–188.

Maggioni, A. (2015). Epidemiology of heart failure in europe. *Heart failure clinics*, 11(4):625–635.

Maggioni, A. P., Dahlström, U., Filippatos, G., Chioncel, O., Leiro, M. C., Drozdz, J., Fruhwald, F., Gullestad, L., Logeart, D., Fabbri, G., et al. (2013). Eurobservational research programme: regional differences and 1-year follow-up results of the heart failure pilot survey (esc-hf pilot). *European journal of heart failure*, 15(7):808–817.

Mandel, M. (2013). Simulation-based confidence intervals for functions with complicated derivatives. *The American Statistician*, 67(2):76–81.

Mazzali, C. and Duca, P. (2015). Use of administrative data in healthcare research. *Internal and emergency medicine*, pages 1–8.

Mazzali, C., Paganoni, A., Ieva, F., Masella, C., Maistrello, M., Agostoni, O., Scalvini, S., and Frigerio, M. (2016). Methodological issues on the use of administrative data in healthcare research: the case of heart failure hospitalizations in lombardy region, 2000 to 2012. *BMC Health Services Research*, 16(1):1.

Mazzali, C., Severgnini, B., Maistrello, M., Barbieri, P., and Marzegalli, M. (2013). Heart diseases registries based on healthcare databases. In *New Diagnostic, Therapeutic and Organizational Strategies for Acute Coronary Syndromes Patients*, pages 25–46. Springer.

Milligan, G. W. and Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179.

Ministero della Salute (1997). *ICD9-CM Italian Version*. `http://www.salute.gov.it`.

Mosterd, A. and Hoes, A. W. (2007). Clinical epidemiology of heart failure. *Heart*, 93(9):1137–1146.

Murgia, A., Scano, G., Palange, P., Corrado, A., Gigliotti, F., Bellone, A., Clini, E., DEL GRUPPO, N. A. A. N., DI STUDIO RIABILITAZIONE, R., AUGUSTYNEN, A., et al. (2004). Linee guida per la ossigenoterapia a lungo termine (otlt) aggiornamento anno 2004. *Rassegna di Patologia dell'Apparato Respiratorio*, 19:206–219.

Muzzarelli, S., Leibundgut, G., Maeder, M. T., Rickli, H., Handschin, R., Gutmann, M., Jeker, U., Buser, P., Pfisterer, M., Brunner-La Rocca, H.-P., et al. (2010). Predictors of early readmission or death in elderly patients with heart failure. *American heart journal*, 160(2):308–314.

Nguyen, L. L. and Barshes, N. R. (2010). Analysis of large databases in vascular surgery. *Journal of vascular surgery*, 52(3):768–774.

Owan, T. E., Hodge, D. O., Herges, R. M., Jacobsen, S. J., Roger, V. L., and Redfield, M. M. (2006). Trends in prevalence and outcome of heart failure with preserved ejection fraction. *New England Journal of Medicine*, 355(3):251–259.

Pai, V. S. and Lawson, D. A. (2001). Rotator cuff repair in a district hospital setting: outcomes and analysis of prognostic factors. *Journal of Shoulder and Elbow Surgery*, 10(3):236–241.

Parmelee, P. A., Thuras, P. D., Katz, I. R., and Lawton, M. P. (1995). Validation of the cumulative illness rating scale in a geriatric residential population. *Journal of the American Geriatrics Society*, 43(2):130–137.

Ponikowski, P., Voors, A. A., Anker, S. D., Bueno, H., Cleland, J. G., Coats, A. J., Falk, V., González-Juanatey, J. R., Harjola, V.-P., Jankowska, E. A., et al. (2015).

2016 esc guidelines for the diagnosis and treatment of acute and chronic heart failure. *European heart journal*, page ehw128.

Pope, G., Kautter, J., Ingber, M., Freeman, S., Sekar, R., and C., N. (2011). Evaluation of the cms-hcc risk adjustment model - final report. Technical report, Centers for Medicare and Medicaid Services, `https://www.cms.gov/Medicare/Health-Plans/MedicareAdvtgSpecRateStats/downloads/evaluation_risk_adj_model_2011.pdf`.

Prentice, R. L., Williams, B. J., and Peterson, A. V. (1981). On the regression analysis of multivariate failure time data. *Biometrika*, 68(2):373–379.

Putter, H., Fiocco, M., Geskus, R., et al. (2007). Tutorial in biostatistics: competing risks and multi-state models. *Statistics in medicine*, 26(11):2389.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, `http://www.r-project.org/` edition.

Ramlau-Hansen, H. (1983). Smoothing counting process intensities by means of kernel functions. *The Annals of Statistics*, pages 453–466.

Reddel, H. K., Bateman, E. D., Becker, A., Boulet, L.-P., Cruz, A. A., Drazen, J. M., Haahtela, T., Hurd, S. S., Inoue, H., de Jongste, J. C., et al. (2015). A summary of the new gina strategy: a roadmap to asthma control. *European Respiratory Journal*, 46(3):622–639.

Redfield, M. M., Jacobsen, S. J., Burnett Jr, J. C., Mahoney, D. W., Bailey, K. R., and Rodeheffer, R. J. (2003). Burden of systolic and diastolic ventricular dysfunction in the community: appreciating the scope of the heart failure epidemic. *Jama*, 289(2):194–202.

Redziniak, D. E., Hart, J., Turman, K., Treme, G., Hart, J., Lunardini, D., Miller, M. D., and Diduch, D. R. (2009). Arthroscopic rotator cuff repair using the opus

knotless suture anchor fixation system. *The American journal of sports medicine*, 37(6):1106–1110.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Saczynski, J. S., Andrade, S. E., Harrold, L. R., Tjia, J., Cutrona, S. L., Dodd, K. S., Goldberg, R. J., and Gurwitz, J. H. (2012). A systematic review of validated methods for identifying heart failure using administrative data. *Pharmacoepidemiology and drug safety*, 21(S1):129–140.

Sanders, G. L. and Shin, S. (2001). Denormalization effects on performance of rdbms. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, pages 9–pp. IEEE.

Schultz, S., Rothwell, D., Chen, Z., and Tu, K. (2013). Identifying cases of congestive heart failure from administrative data: a validation study using primary care patient records. *Chronic diseases and injuries in Canada*, 33(3).

Shah, S., Vanclay, F., and Cooper, B. (1990). Efficiency, effectiveness, and duration of stroke rehabilitation. *Stroke*, 21(2):241–246.

Shanno, D. F. (1970). Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656.

Shin, S. K. and Sanders, G. L. (2006). Denormalization strategies for data retrieval from data warehouses. *Decision Support Systems*, 42(1):267–282.

Therneau, T. M. (2015). *A Package for Survival Analysis in S*. version 2.38, http://CRAN.R-project.org/package=survival.

Therneau, T. M. and Grambsch, P. M. (2000). *Modeling survival data: extending the Cox model*. Springer Science & Business Media.

Vitacca, M., Bianchi, L., Guerra, A., Fracchia, C., Spanevello, A., Balbi, B., and Scalvini, S. (2009). Tele-assistance in chronic respiratory failure patients: a randomised clinical trial. *European Respiratory Journal*, 33(2):411–418.

Vitacca, M., Paneroni, M., Grossetti, F., and Ambrosino, N. (2016). Is there any additional effect of tele-assistance on long-term care programmes in hypercapnic copd patients? a retrospective study. *COPD: Journal of Chronic Obstructive Pulmonary Disease*, pages 1–7.

Wang, T. J., Evans, J. C., Benjamin, E. J., Levy, D., LeRoy, E. C., and Vasan, R. S. (2003). Natural history of asymptomatic left ventricular systolic dysfunction in the community. *Circulation*, 108(8):977–982.

Wei, L. (1992). The accelerated failure time model: a useful alternative to the cox regression model in survival analysis. *Statistics in medicine*, 11(14-15):1871–1879.

Wickham, H. (2009). *ggplot2: elegant graphics for data analysis*. Springer Science & Business Media.

Wickham, H. (2014). *Advanced R*. CRC Press.

Wickham, H. (2015). *R packages*. "O'Reilly Media, Inc.".

World Health Organization (2015). *The International Classification of Diseases system used to classify the different type of diagnosis*. `http://www.who.int/classifications/icd/en/`.

Zarrinkoub, R., Wettermark, B., Wändell, P., Mejhert, M., Szulkin, R., Ljunggren, G., and Kahan, T. (2013). The epidemiology of heart failure, based on data for 2.1 million inhabitants in sweden. *European journal of heart failure*, 15(9):995–1002.

List of Tables

List of Figures