POLITECNICO DI TORINO
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Matematica

Tesi di laurea magistrale

Statistical methods to analyse registry data in a comparative setting

Relatori:
Prof. Mauro Gasparini
Dott.ssa Gaëlle Saint-Hilary

Candidata:
Margherita Annaratone

Dicembre 2018

Contents

Abstract

In rare diseases, randomised controlled trials are not always feasible for ethical reasons or because the required number of patients is too large. In this case, the use of single arm trials (all patients in the treatment group) is preferred, and the control group may be taken from historical data (e.g., registries).

Patients from registries are selected so that they respect the same inclusion/exclusion criteria present in the clinical trial. However, in the absence of randomisation, treatment and control groups may still have differences in baseline characteristics. It is necessary to take into account these differences in order to avoid, or limit, a bias in the treatment effect estimation.

There are several statistical methods to achieve this purpose, and we focus here on propensity score (PS) methods. The PS is a score summarising patients' baseline characteristics, and it can be used to select similar patients belonging to treatment and control arms. We describe the following PS methods: matching, inverse probability of treatment weighting and stratification.

An application is presented using a fictive, but realistic, example where the response variable is a time-to-event endpoint. On this example, the results from the PS methods are compared to those obtained with a naïve analysis (without any adjustment for baseline characteristics) and to those obtained with a covariate adjustment method, in which baseline covariates are simply included in the survival analysis.

The naïve approach results in a biased treatment effect estimation, compared to covariate adjustment and PS methods, with or without "doubly robust" approach (i.e., relevant covariates are included in the survival analysis), that provide almost

the same, unbiased, results.

However, in general, relying on only one single method of analysis can be too misleading and several approaches should be used as sensitivity analyses to check the robustness of the results.

Chapter 1

Historical controls in clinical trials

In this chapter we give a short description of clinical trials, with particular attention to historical control trials. We explain the meaning of historical data - in particular, registry data, with their main features, challenges and possible improvements. In the last subsection, two examples of registries related to rare diseases are mentioned.

1.1 Clinical trials

Clinical trials are experiments on human participants done in clinical research to answer specific questions related to new treatments like, for example, drugs or medical devices. The main endpoints in clinical trials refer to the efficacy and the safety of a new treatment, generally, with respect to placebo or standard of care. Clinical trials are designed so that results can be reproduced and validated by health authorities, before drugs or medical devices are commercialised [1]. The effect of a new treatment for a single patient is given by the difference between what happened to the patient due to the treatment and what would have happened to him without the treatment [4]. In order to do this, clinical trials usually have two populations of patients, a group is treated with the experimental cure (**treatment**) and the other one is treated with placebo, standard of care or even nothing (**control**). The way treatment and control groups are established lead to different type of clinical trials: here, we focus on **randomized controlled** and **historical controlled** trials.

1.1.1 Randomized controlled trials

In randomized trials, patients are randomly allocated to the treatment and control groups. After the allocation, the patients from the two arms are followed in the same way and the only difference is the treatment they receive.
Randomization is considered to be the gold standard for clinical trials, because it minimizes the allocation bias, balancing for both measured and unmeasured prognostic factors (the characteristics of a patient, that can be used to estimate the chance of recovery from a disease). Furthermore, in the case of blinded clinical trials, blinding the identity of treatments from investigators and participants is made easier by randomization [2].

1.1.2 Historical controlled trials

With the term "historical control" we mean a group of subjects external to the study, taken from a different population with respect to the one used in the clinical trial. Historical controls are often used when giving a placebo would be unethical and in the case of rare diseases. In the latter case, only few patients could be included in the trial, so generally all the subjects are enrolled in the treatment group (**single arm trial**) and the control arm is a historical one. **Historical data**, the data related to this external control, are taken from previous trials or natural history studies, like **registries**, either retrospectively (i.e., extraction of data from medical records for further analysis), or - not very common - prospectively (i.e., ad hoc studies to get the required information). The main advantage in using historical controls is that more resources can be allocated to the treatment group resulting in a reduction in costs and in a more accurate point estimation of endpoints [3].

One of the biggest concerns in historical controls is that the external control group should be chosen in such a way that the trial's endpoints are comparable, otherwise large selection bias may be introduced in the analysis. Moreover, if treatment and historical control arms have great differences in covariates, this may inevitably affect the treatment effect estimation.

The following are the main guidelines that must be followed to use historical data as a control arm [5, 6]:

- the course of the analysed disease has to be well documented in both the two arms, with particular attention to the covariates that influence the outcome of the illness. The historical data can be observed at an earlier time and/or at another institution, but the characteristics related to demographics, baseline status and concomitant therapy have to be available;
- the endpoints have to be objective and there has to be a relevant difference in the outcome between treatment and control groups;
- the treatment and control populations have to be similar in demographics and disease characteristics; all the measurements must have been taken in a similar setting and in a similar manner together with timing and methodology.

The simplest ways to include historical data as a control group are:

- dynamic borrowing with test-then-pool: according to the result of a statistical test to assess whether the treatment and external control populations are different, if the two arms are equal enough in baseline characteristics then the historical data are used for the control group;
- pooled: historical data are pooled together with the data from the treatment group, without checking for equality between the two populations.

The purpose of this thesis is to describe complex statistical methods to include historical data as a control arm - even though they may have differences in baseline covariates with the treatment group - permitting to avoid, or limit, a bias in the treatment effect estimate.

1.2 Registries

1.2.1 Definition and differences from clinical trials

A patient registry, by definition, is "an organised system that uses observational methods to collect uniformed data to evaluate specific outcomes for a population characterised by a particular disease and that serves a predetermined scientific, clinical or policy purpose" [7]. Given its observational nature, this kind of studies allow to get real-world data for a specific disease, i.e. its typical clinical features, its differences in phenotype, its natural history. Registries are helpful because they increase the knowledge about a certain illness, so that researchers can develop new treatments based on the trends found among patients in the registry. As in clinical trials, in registries patients may be treated (standard of care or not) or untreated. There are different types of registries according to how their populations are defined: for example, disease registries include patients with the same diagnosis, product registries gather individuals exposed to the same biopharmaceutical products or medical devices. Furthermore, it is worthwhile to stress the importance of rare disease registries in supporting the development of treatment protocols and therapies for those illnesses whose information is poor. This kind of registries need to be updated several times whenever knowledge increases or treatments become available and they have to monitor follow-up rates to determine whether or not there are medical events.

Registries collect information voluntarily about a certain disease, while clinical trial data are obtained in a more restrictive and controlled setting whose purpose is to determine if the treatment satisfies safety and efficacy conditions such that it can be prescribed to the public [8]. As a result, the main differences between registries and clinical trials are related to the treatment of patients and inclusion/exclusion criteria.

1.2.2 Patient population

In registries, doctors decide whether or not a subject has to be treated, and inclusion/exclusion criteria allow to include a wider population than in clinical trials, in which the treatment assignment is generally randomized.

Furthermore, in clinical trials, due to the great amount of tests performed on patients, a higher number of observed parameters is available rather than in registries, in which only the covariates required for their purpose are present.

A registry is created to better understand the characteristics of a **target population**, but, obviously only a subgroup of it is considered - the **accessible population** - because of a series of inclusion/exclusion criteria: these criteria, that are mostly geographic, demographic, disease-specific and temporal, may cause a lack of representativeness in the accessible population.

A further restriction is given by the sampling scheme of the registry, leading to the so-called **intended population**. This subgroup comes from the fact that often for convenience only certain patients are included in the registry: as an example, this is the case when, for simplicity, only the patients who come to the clinic on a certain week day are involved in the study.

The **actual population** contains all the subjects in the intended population that take part in the registry and that consistently go to follow-up visits.

Finally, the **analytic population** represents the true population in the registry: few patients belonging to the actual population are excluded because they may have particular attributes that are not meaningful for the analysis [9].

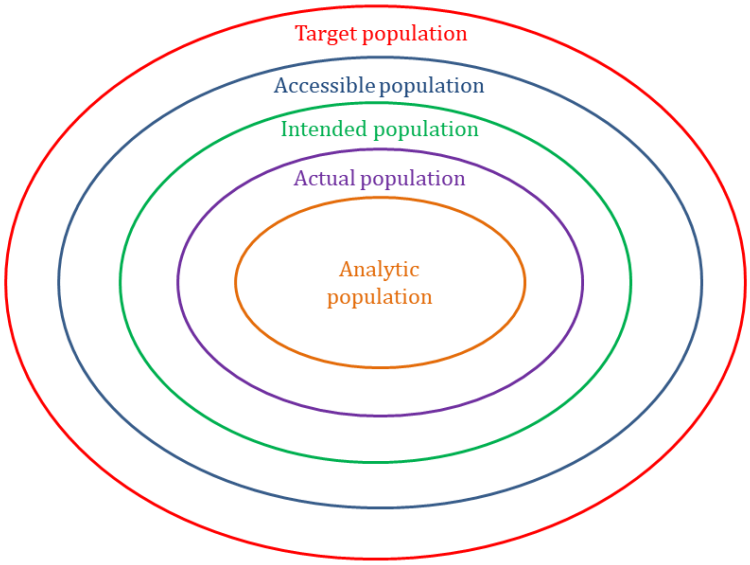


Figure 1.1: All the different populations contained in the target population.

1.2.3 Observed covariates and data completeness

In general, in registries, the following covariates should be provided:

- **demographics**: gender, age, ethnicity, disease status, past history of significant medical conditions, ...;
- **treatment**: therapeutic treatment (e.g., medication or surgery), behavioral factors (e.g., diet, smoking habits);
- **endpoints**: parameters related to safety and effectiveness. Examples are survival and disease recurrence for effectiveness, adverse events for safety;
- **time**: date of starting a treatment and follow-up visits.

One recurrent issue related to registries is the missingness of such variables for certain patients. **Missing data** include variables that are "non reported" or the reported value is not understandable, parameters that are unavailable or missing, covariates with inconsistent values (e.g., at the same time two different values for the same covariate) or out-of-range values (e.g., year of birth 1800). Several methods

have been proposed to handle missing data, and their appropriateness/feasibility depends on the context and the type of missing data [9, 10, 12]. More details are provided in Appendix A.

1.2.4 Main sources of registries and future improvements

Pharmaceutical companies and other organisations are interested in collecting data to understand the value of specific treatments or the trend of a certain disease; here is a list of the main sources of registries [11]:

- **national health systems** generally use data to improve the quality of care and the clinical outcomes;
- **payers** gather information in order to understand the quality and cost of care;
- **pharmaceutical** and **medical device companies** create registries including patients from previous trials in order to use them for historical controls in the case of future trials.

Since historical data are an important resource in drug development, especially in the case of rare diseases, pharmaceutical industries can get access to these real world data, under the condition that the patients' privacy and the confidentiality are maintained.
A great innovation may be the introduction of a common data model (**standardisation**) for registries so that registries related to the same disease can be collected on a platform that allows to look at patients across different sources.
Creating registries require a large amount of resources and time. At hospitals, registries require manual data entry that inevitably affect the quality and the completeness of the data. A good improvement can be automating the procedure of data collection as much as possible, exploiting the support given by patients' electronic health records.

The achievement of registry data automation and standardisation will surely facilitate the collection of essential information to improve clinical care and patient

research.

1.2.5 Examples of rare disease registries

Table 2.1 presents two examples of registries related to rare diseases [9].

NAME	SPONSOR	NUMBER OF SITES	DESCRIPTION	CHALLENGE
The National Amyotrophic Lateral Sclerosis (ALS) Registry	U.S. Department of Health and Human Services and Agency for Toxic Substances and Disease Registry	All U.S. 50 States	Its purpose is to quantify the incidence and prevalence of ALS in the United States. Demographics are available for each patient, together with potential risk factors for the illness.	Combining two data sources (national administrative databases and a secure Web portal where patients can self-enroll) to get the registry database, eliminating duplicate patients.
The Cystic Fibrosis Foundation (CFF) Patient Registry	Cystic Fibrosis Foundation	110 CFF-accredited care centers in the United States	It collects demographics data for patients with cystic fibrosis: clinical visits, hospitalizations, care episodes, morbidity, mortality and treatment outcomes.	Transferring the existing CFF registry to a Web platform. In order to assess the integrity of this data migration, patients were asked to identify and fix errors related to their records in the new registry.

Table 1.1: Two examples of rare disease registries [9].

Chapter 2

Statistical methods to use historical controls

There is currently a large amount of clinical data available, like natural history data or those coming from previous clinical trials, that can be used in new clinical trials in order to achieve different purposes: reducing sample sizes, getting better estimations and comparing an experimental treatment to historical control(s) when it is not feasible to include a control group in clinical trials. This is particularly true in the case of rare diseases, where only few patients are enrolled in trials. Clearly such objectives can be reached only if there is sufficient consistency between historical data and current data.

In this chapter, we will examine several statistical methods that can be used to implement historical controls in clinical trials. The aim of these methods is assuring comparability between treatment and control arms, in order to get a balanced dataset, in a way that takes into account the differences in baseline characteristics between the two groups.

2.1 Why balancing for baseline characteristics?

Randomised trials are conducted in a way that treatment allocation is not confounded with either measured or unmeasured baseline characteristics; this is not

true when dealing with historical controls.

A **confounding** factor is a variable that influences two other variables, "questioning their causal relationship": in our case, a confounder is a baseline covariate that interferes with treatment status (i.e., which treatment patients receive) and treatment outcome (see Figure 2.1) [13].

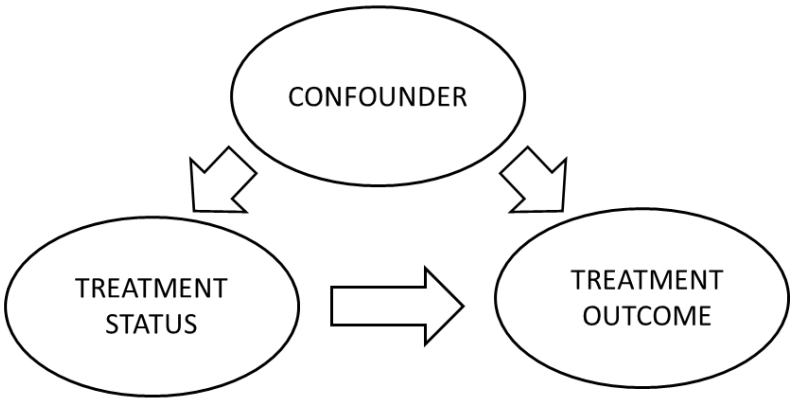


Figure 2.1: Model of a confounding factor.

Confounding factors lead to a biased estimation of the treatment effect, so, in order to get an estimation as close as possible to the truth, particular statistical methods should be used. The following example gives a clear idea of confounding.

Suppose to test a new drug for a certain disease on a population of 100 patients, 50 men and 50 women. The new drug is given to the women while the men get the placebo. At the end of the test period, a higher number of women recover from the illness. In this case, gender is the confounding factor and it is impossible to say whether the drug was effective regardless the patients' sex: perhaps, this result is due to peculiar women's characteristics.

2.2 Statistical methods

2.2.1 Naïve method

The treatment effect estimation is computed without taking into account the presence of confounding factors among baseline characteristics. As a result, the estimation can be biased. We will compare its results to those of the methods handling confounding factors described hereafter.

2.2.2 Logistic regression

A **generalised linear model** is used when we want to "predict" a response variable Y according to a linear combination of n observed covariates $X = (x_1, ..., x_n)$:

$$\eta(X) = \beta X = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n.$$

Suppose that $\mathbb{E}[Y] = \mu$, each generalised model is defined by

- a link function $g(\mu) = \eta(X)$;
- the distribution of Y that belongs to an exponential family with probability density function $f(y\,;\theta,\phi) = \exp(\frac{y\cdot\theta - b(\theta)}{a(\phi)} + c(y,\phi))$, where
  - $\theta = g(\mu)$ is the **location parameter**
  - ϕ is the **scale parameter**
  - a, b, c are known functions that depend on the model.

A **Bernoulli model** is a generalised model where the response variable Y is Bernoulli-distributed, $Y \sim Bernoulli(\mu)$, where μ is the parameter of interest. We need to find θ, ϕ, a, b and c. Knowing that the probability density function of a Bernoulli is

$f(y\,;\,\mu) = \mu^y(1-\mu)^{(1-y)}$ (where $y \in \{0,1\}$); we can equate $f(y\,;\,\mu)$ and $f(y\,;\,\theta,\phi)$:

$$\begin{aligned}
\exp\left(\frac{y\cdot\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right) &= \mu^y(1-\mu)^{(1-y)} \\
\frac{y\cdot\theta - b(\theta)}{a(\phi)} + c(y,\phi) &= \ln\mu^y + \ln(1-\mu)^{(1-y)} \\
\frac{y\cdot\theta}{a(\phi)} - \frac{b(\theta)}{a(\phi)} + c(y,\phi) &= y\cdot\ln\mu + (1-y)\cdot\ln(1-\mu) \\
y\cdot\underbrace{\frac{\theta}{a(\phi)}}_{(1)} - \frac{b(\theta)}{a(\phi)} + c(y,\phi) &= y\cdot\underbrace{\ln\left(\frac{\mu}{1-\mu}\right)}_{(2)} + \ln(1-\mu)
\end{aligned}$$

Looking at (1) and (2), we can put $\theta = \ln\left(\frac{\mu}{1-\mu}\right)$, $\phi = 1$ and $a(\phi) = 1$. As a consequence, $c(y,\phi) = 0$ and, since $\mu = \dfrac{e^\theta}{1+e^\theta}$, we get $b(\theta) = \ln(1+e^\theta)$.
Finally, we have

$$\theta = \ln\left(\frac{\mu}{1-\mu}\right) = \text{logit}(\mu) = \beta X. \tag{2.1}$$

From (2.1), it is possible to obtain $f(y\,;\,\mu)$ as a function of β

$$f(y\,;\,\beta) = \left(\frac{e^{\beta X}}{1+e^{\beta X}}\right)^y \left(\frac{1}{1+e^{\beta X}}\right)^{1-y}.$$

An estimation of β, $\hat{\beta}$, can be evaluated by **maximum likelihood**. By definition, in the case of discrete probability distributions, the likelihood of β given the value of y is equal to the probability of observing y, given the parameters in β; the likelihood can be defined as follows:

$$\mathcal{L}(\beta\,;\,y) = f(y\,;\,\beta).$$

Given a set of m independent observations, the likelihood of all the set of observations is

$$\mathcal{L}(\beta\,;\,y_1,...,y_m) = \prod_{i=1}^{m}\mathcal{L}(\beta\,;\,y_i) = \prod_{i=1}^{m} f(y_i\,;\,\beta).$$

It is more convenient to use the log-likelihood

$$\ln\mathcal{L}(\beta\,;\,y_1,...,y_m) = \sum_{i=1}^{m}\ln f(y_i\,;\,\beta);$$

then, $\hat{\beta}$ is the vector that maximises $\ln \mathcal{L}(\beta\,;\, y_1, ..., y_m)$:

$$\hat{\beta} \;=\; \text{argmax}_{\beta}\; \ln \mathcal{L}(\beta\,;\, y_1, ..., y_m).$$

Explicitly, $\hat{\beta}$ is the vector that satisfies the following conditions:

- $\mathbb{E}\left[\dfrac{\partial \ln \mathcal{L}(\beta\,;\, y_1, ..., y_m)}{\partial \beta}\right] \;=\; 0$
- $\mathbb{E}\left[\dfrac{\partial^2 \ln \mathcal{L}(\beta\,;\, y_1, ..., y_m)}{\partial \beta^2}\right] \;>\; 0$ (maximality condition)

2.2.3 Propensity score based methods

In the case of non-randomized trials, patients are usually assigned to treatment and control groups based on their characteristics. As a result, the probability of being included in the treatment arm does not depend on a pre-defined randomization ratio, as in randomized trials, but it depends on underlying factors in the patient population. In order to reduce the bias in the treatment estimation that can arise, statistical methods could be used to obtain a balanced dataset with comparable groups, at least for the observed patients' variables. One possible method to achieve this purpose is the **propensity score analysis**.

The propensity score (PS) is the probability that a patient will receive the treatment given a set of observed covariates that summarize what is known about the patient prior to treatment assignment. In randomized trials this probability is equal to the randomization ratio (e.g., 50%) for each patient, while in non-randomized ones this probability can be estimated by a logistic regression model (see next section). In this particular case, the dependent variable is the allocated treatment while the independent ones are the patient observed characteristics before treatment; in the end, the logistic regression output will be the probability of being treated, given a set of observed variables [14, 15].

The choice of the variables that should be included in the model is one major issue

of such analyses. First of all, only covariates that are measured at baseline (e.g., demographics, diagnosis date, number of previous treatments) and not post-baseline should be included in the model because the latter ones can be affected by the treatment received. As a consequence, the disease experts and the statisticians should decide among four main sets of variables to be included in the logistic regression: all measured covariates (associated or not to the treatment assignment and/or the outcome), all the covariates that are associated with treatment assignment, those variables that affect the outcome (possible confounders), those variables that affect both the outcome and treatment assignment (true confounders). It was shown that including possible or true confounders in the PS model results in a more balanced design and a better treatment estimation but classifying variables as confounders may be challenging. In any case, most measured variables affect both treatment assignment and the outcome, so often all observed characteristics can be included in the logistic regression without concerns [16].

Finally, problems may arise in the case of variables that affect both treatment assignment and outcome but they are not baseline characteristics: this is the case of covariates related to temporal periods; depending on the time when a patient is enrolled, this may increase his probability to receive a certain (new or old) treatment. For example, in the case of a clinical trial that compares a newer treatment to an older one, patients enrolled in an earlier period are more likely to get the older treatment than the newer one.

Once the PS is estimated for each patient, it can be used to create comparable groups with respect to baseline covariates among treated and untreated subjects; this is particularly useful in the case of single arm trials when, in order to assess the treatment effect, the control group comes from historical data.

There are mainly three PS-based methods adequate for this purpose:

- propensity score matching;
- inverse probability of treatment weighting (IPTW) estimation;

- stratification.

2.2.4 How to estimate the PS

In order to estimate the PS for all the patients, a **logistic regression** is performed. First of all, we have to specify our parameter of interest, PS, the probability of being in the treated arm, and then we define its **odds**

$$ODDS_{PS} = \frac{PS}{1-PS},$$

the ratio between the probability of being in the treatment group and the one of being in the control arm. The logistic regression model is the following:

$$\text{logit}\,(PS) = \ln\left(ODDS_{PS}\right) = \ln\left(\frac{PS}{1-PS}\right) = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n,$$

where $x_1, ..., x_n$ are the baseline covariates and $\beta_0, ..., \beta_n$ are the parameters of the model.
$\hat{\beta}_0, ..., \hat{\beta}_n$, the estimations of the coefficients, are obtained by maximum likelihood; then, the PS is estimated to be

$$\widehat{PS} = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x_1+...+\hat{\beta}_n x_n}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 x_1+...+\hat{\beta}_n x_n}}.$$

For each patient, it is sufficient to substitute the values of his characteristics to $x_1, ..., x_n$ to get his PS estimation.

After the estimation of the PS for each patient, there can be two different scenarios:

- in the case of great unbalance in the observed baseline covariates between the two arms, there are extreme PS values; low PSs for controlled patients because their characteristics are very different from those of the individuals present in the treatment group, that have high PSs;

- in the other case, the patients belonging to both groups have values of PS not far from 0.5.

2.2.5 Propensity score matching

The aim of this method is creating pairs or small sets of patients that have similar values of PS. After that, the treatment effect is estimated on the dataset made up of all these small groups, by applying a model stratified per group or a conventional unmatched analysis [17].

The most common implementation of matching using PS is the **one-to-one** (1:1) in which every treated patient is associated to one patient from the control group with similar propensity score (see Figure 2.2).

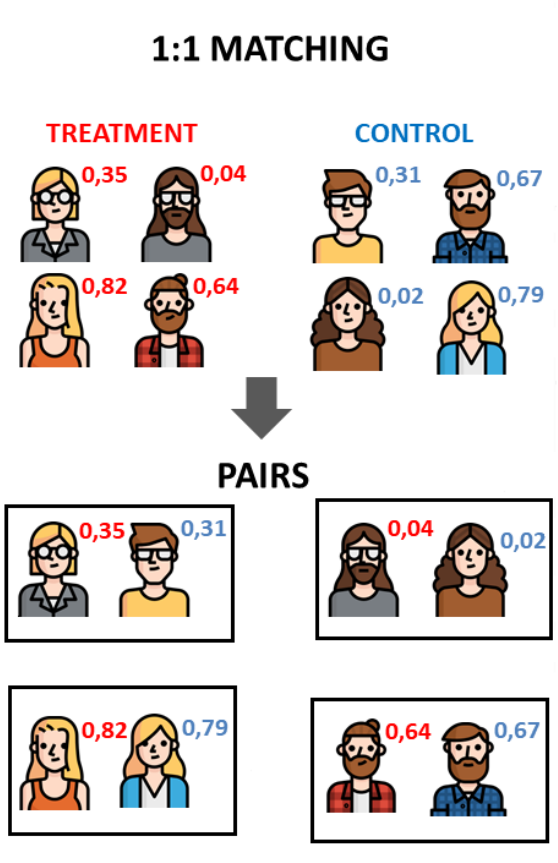


Figure 2.2: Example of 1:1 matching, the number next to each individual is his PS.

The criteria to determine whether a PS is close to another are essentially the following: **nearest neighbour matching** and **nearest neighbour matching within**

a caliper distance.

In both cases, similarity in PS is measured in terms of a distance:

$$d(A, B) = |PS_A - PS_B|,$$

where A is a patient belonging to the treated arm, while B is a patient from the control group; the smaller $d(A, B)$, the more similar A and B. With nearest neighbour matching, each treated subject is associated to a patient from the control group that has the closest PS to his, but there is no restriction on the distance between the two. There can be candidates for the same match that show the same propensity score, in that case one of them is chosen at random to match the pair.
In the case of matching with caliper, the absolute difference in PS between patients in the same pair has to be lower than a specific threshold; due to this, there can be treated or controlled individuals that cannot be matched to any others and they will be excluded from the balanced dataset. Although there is no method to determine the best caliper distance, some recommendations could be found in the literature: 0.1, to avoid pairing dissimilar individuals [17], or 0.2 of the standard deviation of the logit of the propensity score ($\log(\frac{PS}{1-PS})$) that was demonstrated to reduce bias [16].

Matching could be performed with or without **replacement**. There is matching without replacement whenever a patient from smallest group (either treatment or control group) that is included in a pair is no longer available to make pairs with other patients; on the other hand, there is replacement when patients can be part of multiple matched sets. In the case of matching without replacement, in the end, one gets a balanced dataset even though some patients can be excluded from the analysis because there are no longer any other individuals to match. Intuitively, this is not the best approach to use when the sample size is small because this implies discarding information that may be useful in estimating the treatment effect. However, the repetition of the same patient from the control group in multiple pairs compli-

cates the analysis as this should be taken into account in the variance estimation [16].

Another possible aspect to take into account is whether to have a **greedy** or an **optimal** approach for matching. With the first choice, an individual from the smallest arm is randomly selected and matched to one in the other group that has the PS closest to his. This step is repeated until all treated individuals are matched with the untreated ones or there are no patients left in the control group for matching (in case of matching without replacement). Such an approach is called greedy because, every time that a patient from the smallest group is selected, the patient from the other group that has the PS closer to his is chosen even though he/she may have a propensity score that is more similar to another patient. As opposite to the greedy matching, there is the optimal matching in which pairs are made so as to limit the overall difference across pairs. The pairs are the result of an optimisation model, a minimum cost bipartite matching problem, that can be formulated as a minimum cost flow problem [18]. Suppose that the patients are the nodes of a bipartite graph $\mathcal{G} = (\mathcal{T}, \mathcal{C}, \mathcal{E})$, where $\mathcal{T}$ and $\mathcal{C}$ are the treatment and control arms and $\mathcal{E} = \mathcal{T} \times \mathcal{C}$ represents the arcs between pairs; the optimal 1:1 matching is the solution of the following linear problem:

$$\min \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{C}} c_{ij} x_{ij} \tag{2.2}$$

such that

$$\sum_{i \in \mathcal{T}} x_{ij} = 1\,, \qquad \forall\, j \in \mathcal{C} \tag{2.3}$$

$$\sum_{j \in \mathcal{C}} x_{ij} = 1\,, \qquad \forall\, i \in \mathcal{T} \tag{2.4}$$

$$x_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are paired} \\ 0 & \text{otherwise} \end{cases}, \quad \forall\, j \in \mathcal{C}, \forall\, i \in \mathcal{T}. \tag{2.5}$$

The constraints, (2.3), (2.4), (2.5), indicate that each patient in the treatment arm is matched to exactly one in the control group; the coefficients c_{ij} are the differences betwen patients i's and j's PSs, that correspond to the costs of possible matches.

Finally, there is also the so-called **one-to-many** (1:M) implementation in which M patients from the biggest group are associated to an individual in other arm according to similarities in their PSs (see Figure 2.3, for an example of 1:2 matching); the value of M can be either fixed or not, but a reduction in bias is noticed when using a variable number for it. As a consequence, an improvement in the one-to-many approach is the **full** one: a treated subject can be matched to one or more untreated patients; otherwise, each untreated patient is associated to one or more treated ones [16].

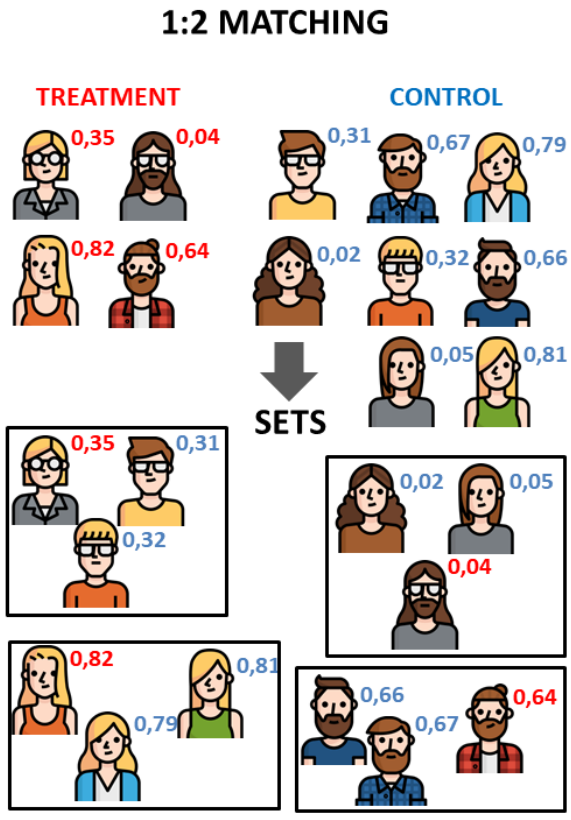


Figure 2.3: Example of 1:2 matching, the number next to each individual is his PS.

		DESCRIPTION	PROS	CONS
METHOD	GREEDY	Each patient in the smallest group is picked at random and matched with the closest one in the other group	Easy to perform in terms of time and computation	Patients may be matched even though a better match could be performed
	OPTIMAL	All matched pairs are made in such a way to minimize the overall difference in PS across all pairs	Improves the overall matching	It requires the solution of an optimization method, so it is time-consuming and needs long computations
	REPLACEMENT	Patients from the smallest group can be part of multiple pairs	More patients from the biggest group are included in the treatment effect analysis	The repetition of the same patients has to be taken into account in the variance estimation
PARAMETERS	CALIPER	The difference in PS has to be lower or equal than a certain value to have a match	Only "real neighbours" are matched	Some patients may remain unmatched
	1:1 OR 1:M	Each patient from the smallest group can be matched with either one or many patients in the other group	1:M : more patients from the biggest group are included in the treatment effect analysis	1:M : remarkable difference in the size of treatment and control groups 1:1 : we don't use all the information available.

Table 2.1: Summary of the main methods and parameters involved in the propensity score matching.

2.2.6 Inverse probability of treatment weighting (IPTW) estimation

In this method, each patient is assigned a weight (a **general weight**) such that for each combination of baseline characteristics, which corresponds to a certain PS value, the sums of contributions of all treated and control patients are equal, resulting in a balanced dataset between treatment groups. In particular, a given propensity score leads to a $\frac{1}{PS}$ weight for treated patients and a $\frac{1}{1-PS}$ weight for control ones. For example, suppose to have 50 patients with a propensity score equal to 0.4, that are divided as follows: 20 (40%) in the treatment group, while the remaining 30 (60%) in the control one. The IPTW assigns a weight of 2.5 $\left(\frac{1}{0.4}\right)$ to the 20 treated patients and a weight of 1.67$\left(\frac{1}{0.6}\right)$ to the remaining subjects in the control arm. Now, it is easy to see that the sum of weights within each group is 50 (20×2.5 for treated patients and 30×1.67 for control ones) [14]. So a direct consequence of

the IPTW implementation is the creation of a **pseudo-population** in which each combination of covariates results almost balanced between treatment and control groups.

IPTW aims at giving more importance (more "weight") to those patients that have unexpected PS values. Remembering that PS is the likelihood of receiving the treatment, given the values of certain covariates; unusual values are:

- treated patients with low PS → given their covariates, they should be part of the control arm;
- control patients with high PS → given their characteristics, they should have received the treatment.

In practice, patients with unexpected PS values are counted more than once in the pseudo-population as it is shown in Figure 2.4, in which it is easy to see how the distribution of PS between the two arms changes when applying IPTW general weights.

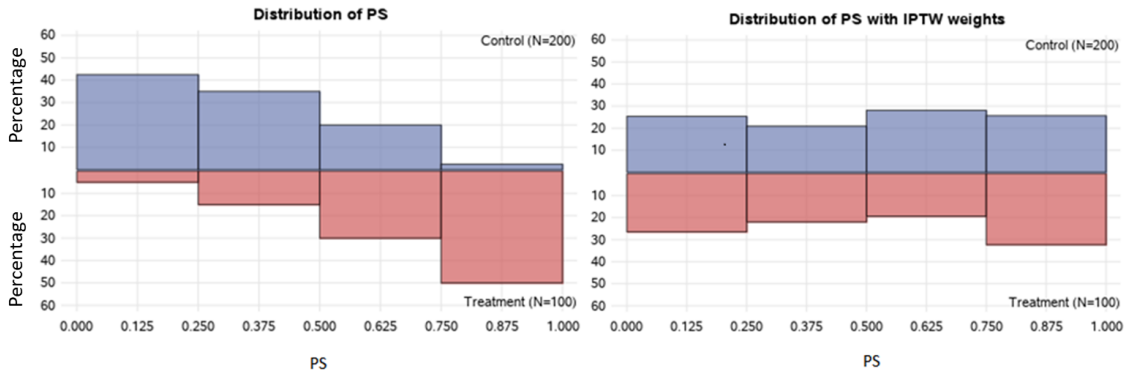


Figure 2.4: The two histograms show how a random distribution of PS can change applying IPTW general weights.

In the case of very high or low PSs, generally when sample sizes are small, weights may be disproportionally high or low (see Figure 2.5) causing an imprecise estimation of the treatment effect: possible solutions can be restricting the analysis to those patients that show homogeneous weights or using **stabilized weights**.

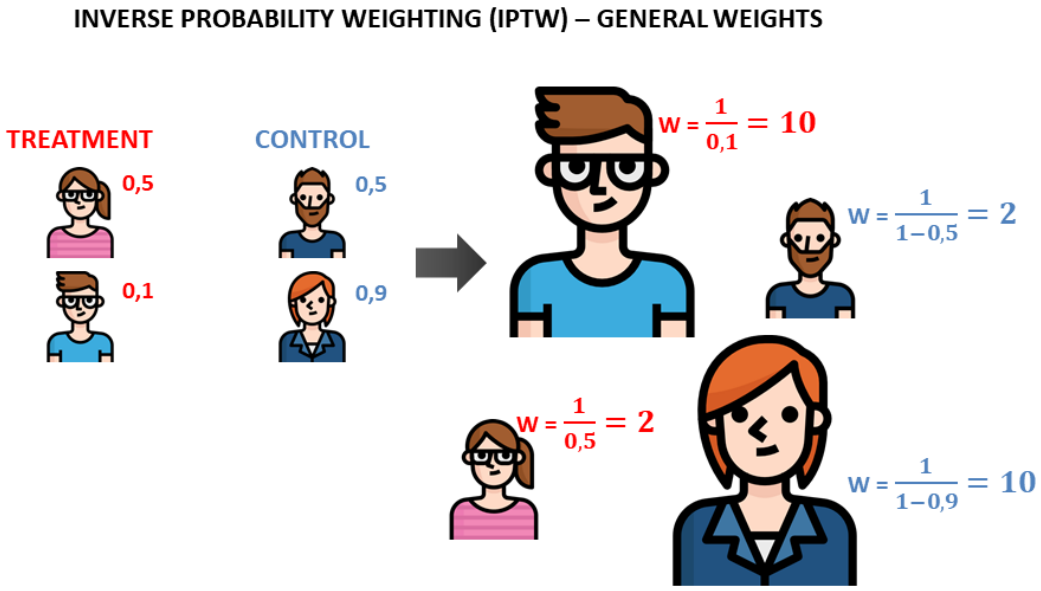


Figure 2.5: Example of IPTW general weights (w), the number next to each individual is his PS.

It is important to note that IPTW with general weights generates a pseudo population whose size is always greater than the sample one; the use of stabilized weights allow to have almost the same size for real and pseudo populations. Let us consider a simple example with just one binary covariate - but it can be extended to continuous and multilevel variables [19]: assume we have a sample size of N patients divided as follows,

Dummy variable	Treated subjects	Untreated subjects	Propensity score
0	n_{01}	n_{00}	PS_0
1	n_{11}	n_{10}	PS_1

Table 2.2: Example with a dummy variable

then, the size of the pseudo population is

$$N_w = \sum_{i=1}^{N} w_i = n_{01} \cdot \frac{1}{PS_0} + n_{00} \cdot \frac{1}{1 - PS_0} + n_{11} \cdot \frac{1}{PS_1} + n_{10} \cdot \frac{1}{1 - PS_1}. \quad (2.6)$$

With the notations presented in Table 2.2, the two propensity scores can be estimated using treated subjects' frequencies in each category of the binary variable:

$$\widehat{PS}_0 = \frac{n_{01}}{n_{01} + n_{00}}, \quad \widehat{PS}_1 = \frac{n_{11}}{n_{11} + n_{10}}.$$

Substituting in (2.6), one gets:

$$N_w = n_{01} + n_{00} + n_{01} + n_{00} + n_{11} + n_{10} + n_{11} + n_{10} = 2\,N. \tag{2.7}$$

Suppose, now, to use stabilized weights; they are defined as

$$w_i = \begin{cases} \frac{1-p}{1-PS_i}, & \text{if } i \text{ is a control patient} \\ \frac{p}{PS_i}, & \text{if } i \text{ is a treated patient} \end{cases}, \tag{2.8}$$

where p is the probability of treatment without any covariate. In this case, the estimation of p is the following

$$\hat{p} = \frac{n_{01} + n_{11}}{N}.$$

The size of the pseudo population - after substituting and simplifying- becomes equal to the sample one

$$N_w = n_{01} + n_{11} + n_{00} + n_{10} = N.$$

However, once the weights are estimated, no matter what approach is adopted, the whole dataset can be used in a weighted statistical analysis to determine the treatment effect.

2.2.7 Stratification

Stratification is similar to matching but without any exclusions of patients. The total dataset is divided into mutually exclusive groups (**strata**), based on the estimated PS: subjects from both arms are stratified in subsets that are defined by specific

thresholds in propensity score (see Figure 2.6 for an example). There are two possible ways to obtain strata [17]:

- **PS quantiles**, the PS range (minimum to maximum, obtained from the PSs in the dataset) is divided into quantiles and all the patients that have a PS that falls within a certain quantile are grouped together;
- **equal sized groups**, each group has the same number of patients.

PS quantiles lead to groups that can be very different in size; while, with equal sized groups, strata may include patients that are not very similar according to their baseline characteristics, because they have very different values of PS.
An important thing to take into account is that strata may be really heterogeneous in the number of treated and control patients; for example, if there is great unbalance in observed baseline characteristics between the two arms, it is possible that the stratum with the lowest values of PS includes only patients belonging to the control group, or vice versa with high values of PS, and in these cases we cannot conduct the analysis.

A question may arise regarding the number of strata to use for the analysis: five strata are commonly used but a higher number of strata can improve the balance of covariates between treatment groups [16]. In any case, the number of strata depends on the size of the dataset so, as a consequence, small datasets should have fewer groups than larger ones, otherwise the treatment evaluation will result in a poor analysis.

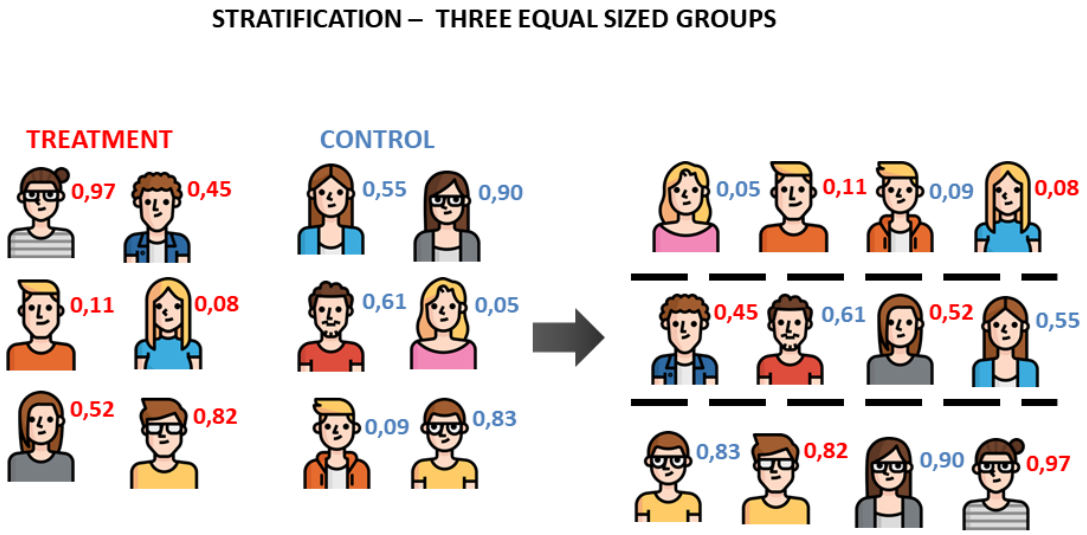


Figure 2.6: Example of three equal sized strata, the number next to each individual is his PS.

Within each stratum, treated and control patients should share almost the same characteristics because they have similar propensity scores. Estimations of the treatment effect are obtained between treated and control arms then, they can be pooled across groups, using meta-analytic approaches to get an overall common estimation. Usually these estimations are weighted by the number of subjects present in each stratum, so, if these groups have N individuals each, weights of $\frac{1}{N}$ are used when pooling the stratum-specific treatment estimations [16].

Furthermore, the strata obtained with this method can also be used in a stratified analysis to get the treatment effect estimation.

2.2.8 Assessing the balance of covariates after PS analysis

As mentioned previously, the purpose of propensity score modeling is balancing the patient characteristics between treatment and control groups; one possible way to determine the effective balance of covariates, after PS analysis, is computing the standardized differences for each observed characteristic [14], using the following

formulas

$$\frac{|\,\overline{x}_T - \overline{x}_C\,|}{\sqrt{\frac{s_T^2 + s_C^2}{2}}}, \tag{2.9}$$

for continuous variables, while for binary ones

$$\frac{|\,\hat{p}_T - \hat{p}_C\,|}{\sqrt{\frac{\hat{p}_T\,(1-\hat{p}_T) + \hat{p}_C\,(1-\hat{p}_C)}{2}}}\,. \tag{2.10}$$

In the above formulas, $\overline{x}_T$, s_T, $\overline{x}_C$ and s_C represent the sample mean and standard deviation of a given covariate in the treatment and control groups, respectively; while, $\hat{p}_T$ and $\hat{p}_C$ are the frequencies of a certain category for a given covariate in the treatment and control arms, respectively. These definitions can be used for PS matching and stratification, but for the IPTW estimation they are slightly different because weights have to be taken into account. First of all, the following quantities must be defined using the definition of w_i in Equation (2.8) (in section 2.2.6) :

$$\overline{x}^w = \frac{\sum_{i=1}^{N} w_i\, x_i}{\sum_{i=1}^{N} w_i},$$

the weighted sample mean, where x_i is the value of a given covariate for a certain patient i;

$$(s^w)^2 = \frac{\sum_{i=1}^{N} w_i\,(x_i - \overline{x}_w)^2}{\sum_{i=1}^{N} w_i\, -\, 1},$$

the weighted sample variance with frequency weights (weights are random variables);

$$\hat{p}^w = \frac{\sum_{i=1}^{N} w_i\, \mathbb{1}_{x_i = certain\,category}}{\sum_{i=1}^{N} w_i},$$

the weighted frequency of a certain category for a given variable.

Then (2.9) and (2.10) become, respectively:

$$\frac{|\,\overline{x}_T^w - \overline{x}_C^w\,|}{\sqrt{\frac{(s_T^w)^2 + (s_C^w)^2}{2}}}, \tag{2.11}$$

and

$$\frac{|\,\hat{p}_T^w - \hat{p}_C^w\,|}{\sqrt{\frac{\hat{p}_T^w\,(1-\hat{p}_T^w) + \hat{p}_C^w\,(1-\hat{p}_C^w)}{2}}}\,, \tag{2.12}$$

where the subscript T or C specifies that these statistics have to be calculated within the treatment and control arms, respectively.

The computation of such differences can be useful for descriptive purpose: they are not influenced by sample sizes and a difference is considered negligible if it is under a reasonable threshold, like 0.1 [16]. However, a more detailed analysis can be done; since matched patients share (more or less) the same propensity score - so they have almost the same values for the observed covariates- independently if they receive the experimental or the control treatment, not only the means or frequencies of the covariates should be the same between treatment and control arms but also their distributions. For this purpose, graphical methods like side-by-side boxplots and density plots can be used in the propensity score matched sample.

In the case of systematic differences between treated and control groups, the initial propensity score model can be improved, for example, including new variables, adding nonlinear terms or interactions between covariates, etc. In any case, the selection of variables to include in the logistic regression should be led by the objective of adding as many variables as required to have an effective model to assess balance between treatment groups.

In addition to this, sensitivity analyses may be conducted with subsets of covariates to assess the robustness of the results.

2.2.9 Covariate adjustment

With covariate adjustment, relevant baseline covariates are included in the model for the treatment effect estimation, in order to minimize their confounding effect. In particular, this method aims at balancing for baseline characteristics during the estimation of the treatment effect; while, PS methods try to reduce the effects of confounding on the dataset before the treatment effect estimation.

Generally, the common rule is having at least ten observations per covariate included in the model. When simple sizes are small, for example, in the case of rare diseases, if the number of available covariates is relatively high compared to the number of patients, covariate adjustment can result in an overfitted model.
A possible solution to this issue is using PS instead of baseline covariates in the model: by definition, PS is the outcome of a logistic regression on observed baseline covariates, so PS summarises all patients' characteristics into a single variable avoiding overfitting [17]. As a result, the treatment effect is estimated according to each patient's likelihood of receiving the treatment, given the values of their observed characteristics.

Chapter 3

Application

In this chapter, an example of application of the methods discussed in Chapter 2 is presented: after a description of the data and the treatment analysis used, the results obtained with the different methods are summarised in the different sections.

For confidentiality reasons, we present here a fictive, but realistic, case-study inspired by a real case. The data are simulated, and the disease and the treatments are not explicitly named. This application was performed with SAS 9.04 and R 3.5.1.

3.1 Data description

For the application, a dataset with 400 patients was created: 100 patients in the treatment arm and 300 patients in the control arm. The dataset is made up of 10 variables with the relative baseline values:

- **count**, a patient id;
- **trt**, equal to 1 if the patient is in the treatment arm, 0 otherwise;
- **sex**, 0 for females and 1 for males;
- **age** (continuous);
- **V1**, **V2** and **V5** are continuous baseline variables;

- **V3** and **V4** are discrete variables; V3 and V4 have values in $\{1, 2, 3, 4, 5\}$;
- **Event time** is the dependent variable and the primary endpoint. It is a time to event endpoint, expressed in days: i.e., the number of days before recovery or censoring occurs.

In Tables 3.1 and 3.2, descriptive statistics for those variables are presented, by treatment group

ARM	VARIABLE	MEAN	SAMPLE STANDARD DEVIATION	MIN	MAX
Control (N=300)	AGE	64.53	8.78	38.84	93.51
	V1	8.01	0.64	6.13	9.82
	V2	27.46	1.35	23.93	30.55
	V5	2059.08	512.22	822.89	3090.41
Treatment (N=100)	AGE	65.05	9.30	43.45	92.19
	V1	7.80	0.60	6.65	9.83
	V2	27.65	1.47	23.50	31.07
	V5	2146.77	533.33	955.19	3190.61

Table 3.1: Descriptive statistics for the continuous variables in the dataset, by treatment and control group.

ARM	VARIABLE	CATEGORY	n/N (%)
Control (N=300)	SEX	Female (0)	161/300 (53.67%)
		Male (1)	139/300 (46.33%)
	V3	1	4/300 (1.33%)
		2	67/300 (22.33%)
		3	163/300 (54.33%)
		4	64/300 (21.34%)
		5	2/300 (0.67%)
	V4	1	0/300 (0.00%)
		2	3/300 (1.00%)
		3	138/300 (46.00%)
		4	152/300 (50.67%)
		5	7/300 (2.33%)
Treatment (N=100)	SEX	Female (0)	42/100 (42.00%)
		Male (1)	58/100 (58.00%)
	V3	1	4/100 (4.00%)
		2	29/100 (29.00%)
		3	45/100 (45.00%)
		4	22/100 (22.00%)
		5	0/100 (0.00%)
	V4	1	4/100 (4.00%)
		2	45/100 (45.00%)
		3	44/100 (44.00%)
		4	7/100 (7.00%)
		5	0/100 (0.00%)

Table 3.2: Descriptive statistics for the categorical variables in the dataset, by treatment and control group.

The primary endpoint, the variable *Event time*, has,

- for the control group, median equal to 128.5 days (min=1 and max=684);
- for the treatment group, median equal to 102.5 days (min=1 and max=652).

If we look at variable *V4*, we see that the statistics are quite different between the two arms, meaning that this variable can be a **true confounder** if it also influences the treatment outcome.

3.2 Survival analysis

In order to assess the treatment effect, a survival analysis has been applied on a time-to-event dataset. In describing this dataset, it is worthwhile to give a description of the survival models used in this application.

Survival analysis can be used to assess the efficacy of a treatment compared to another. Suppose we have treatment and control arms, like in our case; both groups are followed-up to check whether patients have recovered/relapsed from the disease (for simplicity, we consider recovery as the event under examination). We define a **survival time** T_{ij}^* as the time from the enrollment in the trial until recovery occurs for patient i, with $T_{ij}^* \overset{i.i.d}{\sim} f_j(t)$ for each patient and $j = 1, 2$, the treatment and the control, respectively, to specify that the two treatments are supposed to lead to different probability density functions. For each treatment j, the survival analysis aims at estimating

$$S_j(t) = \mathbb{P}(T_j^* > t) = 1 - F_j(t),$$

the survival function of $T_j^* \sim f_j(t)$, the probability that a patient recovers after time t with treatment j. The **hazard function**, the probability that an individual recovers at time t, conditional on being observed without a recovery up to that time, for each treatment, is

$$\lambda_j(t) = \lim_{h \to 0^+} \frac{\mathbb{P}(t \leq T_j^* < t + h \,|\, T_j^* \geq t)}{h}$$

applying the definition of conditional probability

$$\lambda_j(t) \;=\; \lim_{h\to 0^+} \frac{\mathbb{P}\left(t \le T_j^* < t+h\right)}{h \;\cdot\; \mathbb{P}(T_j^* \ge t)} \;=\; \lim_{h\to 0^+} \frac{F_j\left(t+h\right) - F_j(t)}{h \;\cdot\; S_j(t)}$$

from the definition of difference quotient, we get the derivative of $F_j(t)$; so the result is

$$\lambda_j(t) \;=\; \frac{f_j(t)}{S_j(t)},$$

where $f_j(t)$ is the derivative of $F_j(t)$. As a consequence, the **cumulative hazard function** is defined as follows

$$\Lambda_j(t) \;=\; \int_0^t \lambda_j(s)\, ds.$$

3.2.1 Kaplan-Maier curve

If we consider a very short period of time h, $\lambda_j(t)$ can be thought of being constant over h, then it is possible to approximate:

$$\Lambda_j(t+h) - \Lambda_j(t) \;\approx\; \lambda_j(t) \cdot h \;=\; \mathbb{P}\left(t \le T_j^* < t+h \,|\, T_j^* \ge t\right); \tag{3.1}$$

In an empirical way, (3.1) can be approximated to

$$\frac{R_j(t+h) \;-\; R_j(t)}{N_j(t)},$$

where $R_j(t+h)$ and $R_j(t)$ are the number of recoveries measured, respectively, at the instants $t+h$ and t, while $N_j(t)$ is the number of people that have not had a recovery yet just before time t.

Under this consideration, if the interval $[t; t+h)$ is small such that there is only one recovery, it is possible to give the Nelson-Aalen estimator for the cumulative hazard function:

$$\hat{\Lambda}_j(t) \;=\; \sum_{k\,:\,t_k \le t} \frac{R_j(t_k)}{N_j(t_k)}, \tag{3.2}$$

where $t_1, t_2, ..., t_n$ are the times at which recoveries are shown [20].

In order to get the estimation of the survival function $S_j(t)$ for treatment j, a "plug-in" approach can be used [21]. Suppose that the interval $[0; t]$ is divided in sub intervals, identified by $t_1, t_2, ..., t_n$ that are the time at which recoveries occur:

$$S_j(t) = \mathbb{P}(T_j^* > t \,|\, T_j^* > t-1) \cdot \mathbb{P}(T_j^* > t-1) = (1-\mathbb{P}(T_j^* \leq t \,|\, T_j^* > t-1)) \cdot \mathbb{P}(T_j^* > t-1) =$$

$$= q_j(t) \cdot S_j(t-1). \tag{3.3}$$

By recursion, (3.3) can be written as:

$$S_j(t) = q_j(t) \cdot q_j(t-1)...q_j(0), \tag{3.4}$$

where $q_j(0) = 1 - \mathbb{P}(T_j^* = 0 \,|\, T_j^* > -1) = 1 - \mathbb{P}(T_j^* = 0)$. In particular, $q_j(t)$, with $t \in \{0, t_1, ...\}$ can be rewritten using a generic term in the sum (3.2):

$$q_j(t_k) = 1 - \mathbb{P}(T_j^* \leq t_k \,|\, T_j^* > t_{k-1}) = 1 - \frac{R_j(t_k)}{N_j(t_k)}. \tag{3.5}$$

Finally, plugging into (3.4) the result in (3.5), the **Kaplan-Meier estimation** of the survival function for treatment j is obatined

$$\hat{S}_j(t) = \prod_{k\,:\,t_k \leq t} \left(1 - \frac{R_j(t_k)}{N_j(t_k)}\right). \tag{3.6}$$

In order to compute the confidence intervals for the estimated $\hat{S}_j(t)$, it is necessary to apply a transformation to $\hat{S}(t)$, otherwise values bigger than 1 or lower than 0 may be obtained for lower and upper bounds.
We have decided to use a **loglog transformation** $L(S(t)) = \ln(-\ln(S(t)))$ [23], so that the α-confidence interval for L is $[\hat{L} - A; \hat{L} + A]$. Starting from this interval,

we can get the one for $S(t)$ [23]:

$$\begin{aligned}
\ln(-\ln(\hat{S}(t))) - A &\leq \ln(-\ln(\hat{S}(t))) + A \\
e^{\ln(-\ln(\hat{S}(t)))-A} &\leq e^{\ln(-\ln(\hat{S}(t)))+A} \\
e^{-A} \cdot (-\ln(\hat{S}(t))) &\leq e^{A} \cdot (-\ln(\hat{S}(t))) \\
e^{A} \cdot \ln(\hat{S}(t)) &\leq e^{-A} \cdot \ln(\hat{S}(t)) \\
\ln(\hat{S}(t))^{e^{A}} &\leq \ln(\hat{S}(t))^{e^{-A}} \\
(\hat{S}(t))^{e^{A}} &\leq (\hat{S}(t))^{e^{-A}}
\end{aligned}$$

Furthermore, $A = z_{1-\frac{\alpha}{2}} \cdot \dfrac{S(L(t))}{\sqrt{n}}$, where $z_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quantile of a standard normal distribution, $S(L(t))$ is the sample standard deviation of $L(t)$ and n the sample size. Knowing from the Greenwood's formula [23] the survival sample variance

$$S^2(S_j(t)) = [\hat{S}(t)_j]^2 \cdot \sum_{k\,:\,t_k \leq t} \frac{R_j(t_k)}{(N_j(t_k) - R_j(t_k))N_j(t_k)},$$

applying the delta method we get

$$S^2(L(t)) = S^2(\ln(-\ln(S(t)))) = \frac{1}{\ln^2(\hat{S}(t))} \cdot \sum_{k\,:\,t_k \leq t} \frac{R_j(t_k)}{(N_j(t_k) - R_j(t_k))N_j(t_k)}$$

In the end the $(1-\alpha)$**-confidence interval of** $S(t)$ is

$$[(\hat{S}(t))^{\exp\left(z_{1-\frac{\alpha}{2}} \cdot \frac{S(L(t))}{\sqrt{n}}\right)} ; (\hat{S}(t))^{\exp\left(-z_{1-\frac{\alpha}{2}} \cdot \frac{S(L(t))}{\sqrt{n}}\right)}]$$

An important thing to stress is that some patients may be lost to follow-up during the study period and they have not experienced a recovery or they recover after the end of the study period (**right censoring**), so we need to define a new variable C_i that is the censoring time for patient i. So, instead of observing just the survival times, we have the pair (T_{ij}, δ_i) for each subject i with treatment j:

$$\begin{cases} T_{ij} = \min\left(T_{ij}^{*}\ ,\ C_i\right) \\ \delta_i = \mathbb{1}_{\{T_{ij}^{*} \leq C_i\}} \end{cases},$$

where δ_i describes patient i's status, 0 if he is censored or 1 if he recovers before censoring. The results obtained before are the same, with the substitution of T^*_{ij} with T_{ij}. The censoring only affects $N_j(t_k)$ in (3.6) because if there are censoring times between t_{k-1} and t_k, they reduce the size of the population that has not had a recovery just before time t_k.

In the case of a weighted analysis, there is also a **weighted Kaplan-Meier estimator** [22], that is simply the estimator (3.6), adjusted with the patients' weights w_{ij}, related to treatment j . Remembering that for each patient we have (T_{ij}, δ_i), (3.6) becomes:

$$\hat{S}^w_j(t) = \prod_{k\,:\,t_k \le t} \left(1 - \frac{R^w_j(t_k)}{N^w_j(t_k)}\right),$$

where $R^w_j(t_k) = \sum_{i\,:\,T_{ij}=t_k} w_{ij}\,\delta_i$ and $N^w_j(t_k) = \sum_{i\,:\,T_{ij}\ge t_k} w_{ij}$.

The survival sample variance in this case is $S^2(S^w_j(t)) = [\hat{S}^w_j(t)]^2 \cdot \sum_{k\,:\,t_k \le t} \frac{1 - s_{kj}}{M_{kj} \cdot s_{kj}}$,

where $M_{kj} = \dfrac{\left(\sum_{i:T_{ij}\ge t_k} w_{ij}\right)^2}{\sum_{i:T_{ij}\ge t_k} w^2_{ij}}$ and $s_{kj} = 1 - \frac{R^w_j(t_k)}{N^w_j(t_k)}$.

In an analogous way to what has been done for the unweighted survival function, it is possible to obtain the $(1-\alpha)$**-confidence interval of the weighted** $S(t)$.

3.2.2 Log-rank test

To check whether the covariates in the model are statistically significant, a non-parametric **Log-rank test** could be performed [29]. Before defining the log-rank statistics, we must define the following quantities; let $k = 1, 2, ..$ be the distinct times at which events occur in both arms, then

- N_{kT} and N_{kC} are the number of patients that have not had an event yet or been censored at the beginning of period k, respectively, in the treatment and

control groups ($N_k = N_{kT} + N_{kC}$);

- O_{kT} and O_{kC} are the number of observed recoveries in the treatment and control arms, respectively, at time k ($O_k = O_{kT} + O_{kC}$).

Under the null hypothesis that the two arms have the same hazard functions, O_{kj} is hypergeometrically distributed with parameters N_k, N_{kj} and O_k, where $j = T, C$. The expected value and variance of this distribution are

$$E_{kj} = \frac{O_k}{N_k} N_{kj} \quad \text{and} \quad V_k = \frac{O_k \frac{N_{kj}}{N_k} \left(1 - \frac{N_{kj}}{N_k}\right) (N_k - O_k)}{N_k - 1}.$$

Choosing $j = T$ (analogously, for $j = C$), we define the Log-rank statistic:

$$\chi^2_{stat} = \frac{\left(\sum_k (O_{kT} - E_{kT})\right)^2}{\sum_k V_k};$$

then, its p-value is calculated with respect to a chi-square distribution with one degree of freedom.

3.2.3 Cox Model

We consider, now, a semiparametric model, the **Cox model** that allows to evaluate the effect of covariates on survival times. The Cox model relies on the assumption of **proportional hazards**:

$$\lambda(t \,|\, X) = \lambda_0(t) \cdot g(X),$$

where X is a vector of observed covariates, $g(X)$ is the parametric component of the hazard and $\lambda_0(t)$ is the **baseline hazard** that is never estimated but it has to be positive [24].

The hazard function in a Cox model is expressed in the following way

$$\lambda(t \,|\, X) = \lambda_0(t) \cdot e^{\beta' X} = \lambda_0(t) \cdot e^{\beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n},$$

where X is n-dimensional. It is easy to see that $\lambda_0(t)$ is the hazard function when X is the null vector. This way of expressing the hazard function makes easy to calculate the so-called **hazard ratios** for each covariate in the model. Suppose to have two patients with the same characteristics but one of the two is in the treatment group ($x_1 = 1$) while the other in the control arm ($x_1 = 0$); the hazard ratio of x_1 is

$$HR_{x_1} = \frac{\lambda_0(t) \cdot e^{\beta_1 1 + \beta_2 x_2 + \ldots + \beta_n x_n}}{\lambda_0(t) \cdot e^{\beta_1 0 + \beta_2 x_2 + \ldots + \beta_n x_n}} = e^{\beta_1} \tag{3.7}$$

The same concept may be extended to a generic variable x_i incremented of one. From (3.7), we get that the coefficients $\beta_{x_i} = \ln HR_{x_i}$. For each x_i, it is possible to have the following results:

- $HR_{x_i} = 1$, an increment in x_i does not affect the hazard function;
- $HR_{x_i} > 1$, an increment in x_i increases the hazard function, i.e. increases the risk/chance to have an event;
- $HR_{x_i} < 1$, an increment in x_i decreases the hazard function, i.e. decreases the risk/chance to have an event.

Now, we demonstrate how to estimate β parameters [25]. The probability that a patient j recovers at time T_j, given a vector of observed covariates X_j is

$$L_j(\beta) = \frac{\lambda(T_j \mid X_j)}{\sum_{k:T_k \geq T_j} \lambda(T_j \mid X_k)} = \frac{\lambda_0(T_j) \cdot e^{\beta' X_i}}{\sum_{k:T_k \geq T_j} \lambda_0(T_j) \cdot e^{\beta' X_k}} = \frac{e^{\beta' X_j}}{\sum_{k:T_k \geq T_j} e^{\beta' X_k}}. \tag{3.8}$$

This is a partial likelihood and β can be estimated without modeling the change of the hazard over time. Fixing the instant T_j and considering the subjects as statistically independent, the joint probability (another partial likelihood) of all the realised recoveries is

$$L(\beta) = \prod_{j:\delta_j = 1} L_j(\beta), \tag{3.9}$$

where $\delta_j = 1$ if the patient j recovers at time T_j.

In order to get an estimation of β parameters is sufficient to maximise the natural logarithm of (3.9):

$$\ln(L(\beta)) = \sum_{j:\delta_i=1} \left(\beta' X_j - \ln \sum_{k:T_k \geq T_j} e^{\beta' X_k} \right). \tag{3.10}$$

To get the $(1-\alpha)$**-confidence interval of** HR_{x_i}

$$[e^{\hat{L}_{x_i}} ; e^{\hat{U}_{x_i}}]$$

is necessary to compute

$$\hat{L}_{x_i} = \hat{\beta}_{x_i} - z_{1-\frac{\alpha}{2}} \cdot \frac{S(\hat{\beta}_{x_i})}{\sqrt{n}} \quad \text{and} \quad \hat{U}_{x_i} = \hat{\beta}_{x_i} + z_{1-\frac{\alpha}{2}} \cdot \frac{S(\hat{\beta}_{x_i})}{\sqrt{n}},$$

where n is the sample size and $S(\hat{\beta}_{x_i})$ is the sample standard deviation of β_{x_i} that can be obtained when maximising (3.10) to get the estimation of the coefficients.

In order to estimate the parameters and their relative sample variance in the case of a **weighted Cox model**, it is sufficient to use

$$L_j^w(\beta) = \left(\frac{e^{\beta' X_j}}{\sum_{k:T_k \geq T_j} w_k \cdot e^{\beta' X_k}} \right)^{w_j}$$

instead of (3.8) and repeat the same steps [26].

The assumption that all the patients share the same baseline hazard $\lambda_0(t)$ may be too strong; this consideration leads to the **stratified Cox model**. Given the study population divided into strata, each stratum shares the same baseline hazard, so that the hazard function becomes

$$\lambda_h(t \,|\, X) = \lambda_{0_h}(t) \cdot e^{\beta' X} = \lambda_{0_h}(t) \cdot e^{\beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n}$$

for each stratum h. The estimation of the coefficients in the model is more complex because the presence of strata has to be taken into account. The partial likelihoods (3.8) and (3.9) change respectively into:

$$L_{jh}(\beta) = \frac{e^{\beta' X_{jh}}}{\sum_{k:T_k \geq T_j} e^{\beta' X_{kh}}},$$

where we consider j-th patient in the h stratum, and

$$L(\beta) = \prod_h \prod_{j:\delta_j=1} L_{jh}(\beta).$$

After that, the computations are the same as before.

To check whether the covariates in the model are statistically significant, a Wald Chi-Square statistic can be computed for each variable i:

$$\chi_i^2 = \left(\frac{\hat{\beta}_i}{SE_i}\right)^2,$$

where SE_i is the standard error of variable i; then, its p-value is calculated with respect to a chi-square distribution with one degree of freedom [27].

3.3 Simulated dataset

The survival and censoring times (right censoring) have been simulated from an exponential distribution (parametric) [28], where its parameter and survival function are, respectively

$$\lambda(trt, V4) = \lambda = \frac{1}{720} \cdot e^{0.621 \cdot trt + 0.468 \cdot V4} \quad \text{and} \quad S(t, \lambda) = e^{-\lambda t}.$$

Given $S(t, \lambda) = u$ where $u \sim Uniform(0, 1)$, survival times are

$$T(u, \lambda) = -\frac{\ln u}{\lambda}.$$

As a consequence, $V4$ is a **true confounder** because it affects both treatment assignment and treatment outcome.

There are 24 censored patients (9 in the treatment group and 15 in the control arm) out of 400; the times are expressed in days and the study period lasts 2 years.

3.4 Naïve method and covariate adjustment

The whole dataset has been used to estimate two Kaplan-Meier curves, one for the treatment and one for the control (Figure 3.1).

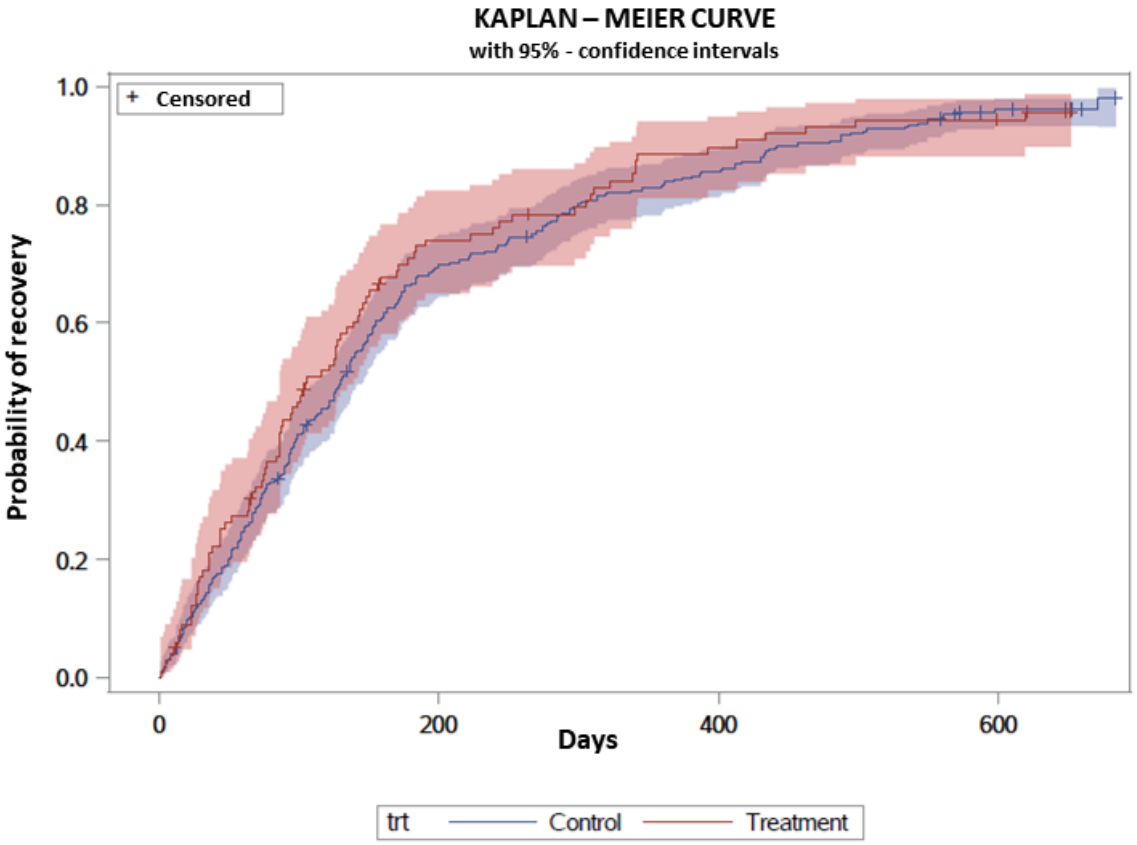


Figure 3.1: Kaplan - Meier curves for control and treatment groups on the whole dataset.

A Cox model with only the treatment as a variable permits to estimate the hazard ratio of the treatment

$$HR_{trt} = \frac{\lambda(t \mid trt = \text{"Treatment"})}{\lambda(t \mid trt = \text{"Control"})} = 1.107, \quad \text{p-value} = 0.3985$$

with $[0.874; 1.403]$ as the 95%-confidence interval of the HR_{trt}.

In order to reduce the confounding effect of baseline covariates, a simple approach is covariate adjustment: in the Cox model, we add also the baseline variables together with *trt* (in our case, the only confounder is *V4* but we pretend not to know it). The hazard ratio of the treatment obtained with this method is

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.857, \quad \text{p-value} < 0.0001$$

with $[1.363; 2.530]$ as the 95%-confidence interval of the HR_{trt}.

3.5 Propensity score based methods

3.5.1 Propensity score estimation

In order to estimate a propensity score for each patient, a logistic regression has been fitted using all the baseline variables in the dataset (*sex*, *age*, *V1*, *V2*, *V3*, *V4*, *V5*). Once the PS has been calculated for each patient, an important thing to do is comparing its distribution between the two arms. If there is a good overlap between the two distributions, this means that there is not a great unbalance among the variables used to estimate the PS (values of PSs close to 0.5); otherwise, at least one of the covariates included in the PS estimation has values very different between the two treatment groups (extreme values of PSs).
We expect to be in the latter case because variable *V4* has very different values between the two arms; in fact, looking at Figure 3.2, we see that a high percentage of controlled patients has low PSs, while most of the treated subjects have high PSs.

In the following subsections, we will show the results obtained by applying the different PS based methods. Since these methods modify the dataset in several ways (e.g, reducing its size, adding weights and stratifying it), for every method, we will estimate two new Kaplan-Meier curves, one for each treatment. After that, we will

use a Cox model with only treatment as a covariate to get the treatment hazard ratio (its confidence interval and p-value) and, then, we will adopt a "doubly robust" approach [17], adding also the variables included in the estimation of the PS to the Cox model. The "doubly robust" (DR) approach aims at further removing the confounding effect of covariates.

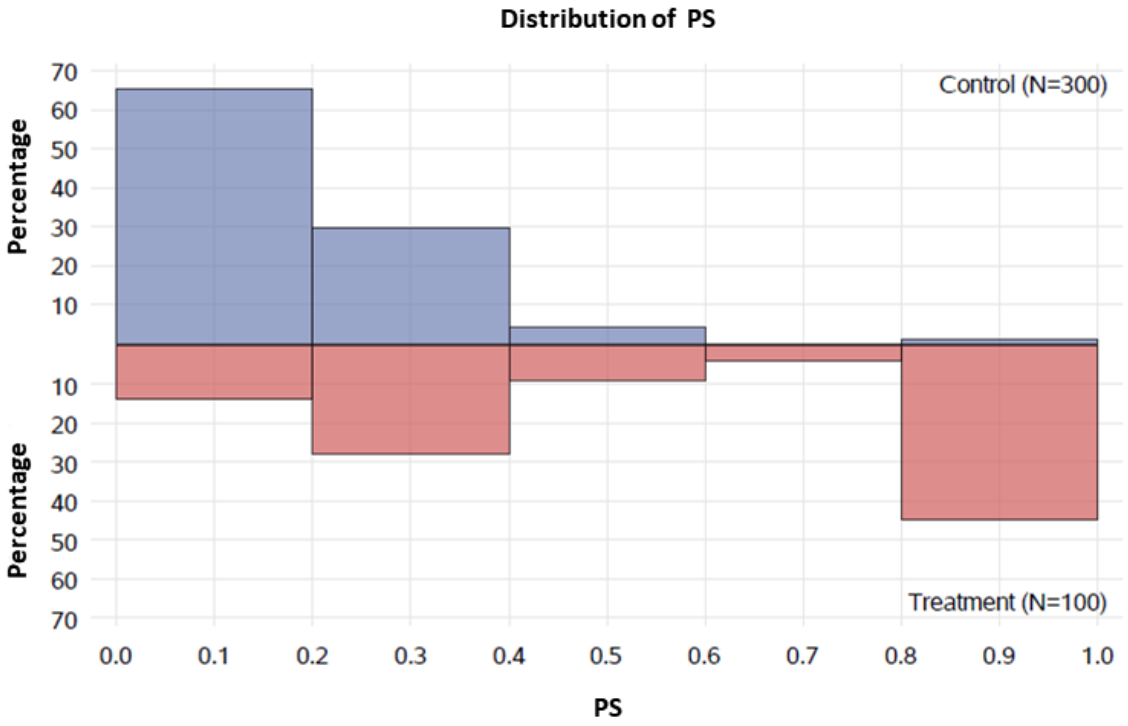


Figure 3.2: Distribution of the propensity score within each treatment arm.

3.5.2 Matching

Greedy 1:1

Fisrt of all, the PSs for all the patients in the dataset have to be computed. Starting from the smallest group (the treatment one), a treated patient is picked at random and he is matched to a subject in the control arm that has the PS closer to his (if multiple matches are available, one of them is randomly chosen). This step is repeated until all the patients in the treated arm are matched; since it is a 1:1 matching, each treated patient is matched to exactly one subject in the control

group. No limit was fixed on how big the difference in PS has to be between patients belonging to the same pair (very dissimilar patients can be matched, see Figure 3.3) and the number of pairs is equal to the size of the treatment arm, i.e, 100 pairs, for a total of 200 patients. There is no replacement, i.e. no reintroduction of the "already-matched" patients.

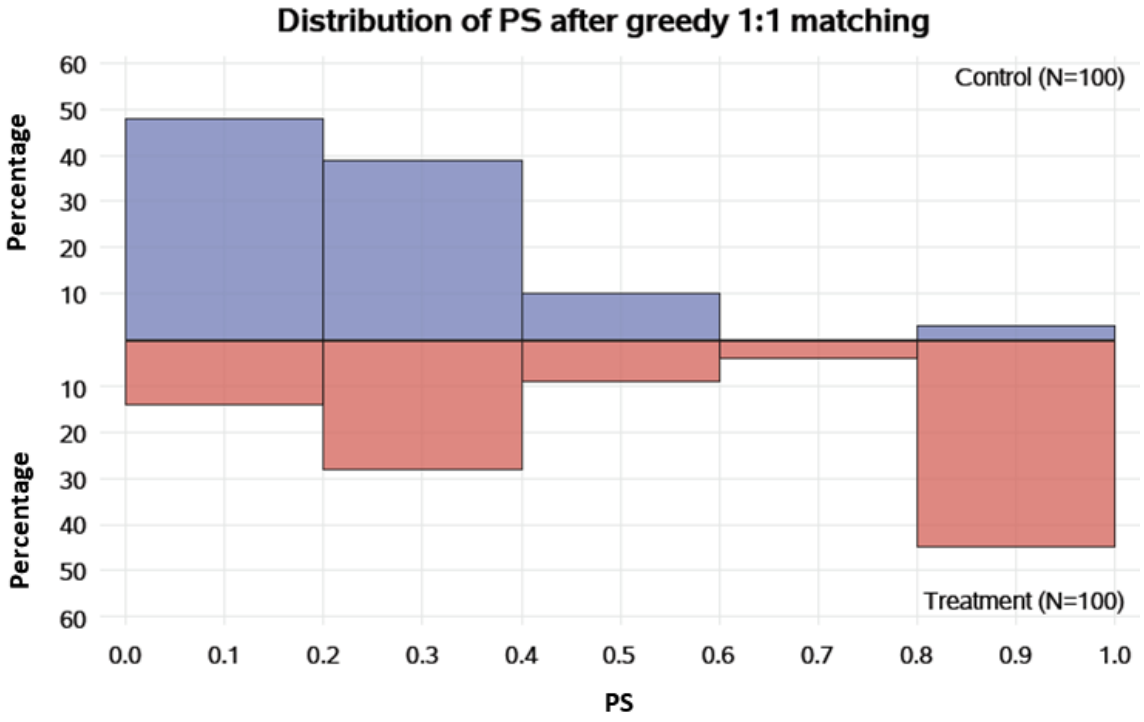


Figure 3.3: Distribution of the propensity score within each treatment arm after the greedy 1:1 matching.

Figure 3.4 shows the two Kaplan Meier curves.

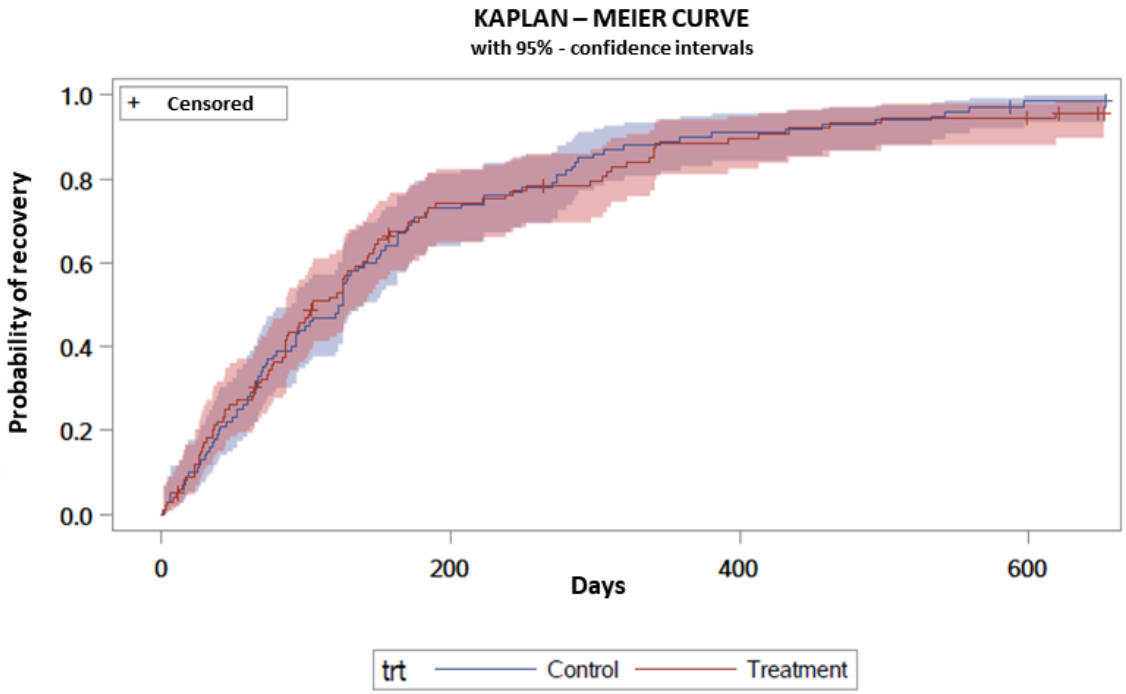


Figure 3.4: Kaplan - Meier curves for control and treatment groups on the greedy 1:1 matched dataset.

The results for the hazard ratios of the treatment with and without "doubly robust" approach are:

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 0.952, \quad \text{p-value} = 0.7385$$

with $[0.716; 1.268]$ as the 95%-confidence interval of the HR_{trt};

$$HR_{trt}^{DR} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.819, \quad \text{p-value} = 0.0009$$

with $[1.279; 2.589]$ as the 95%-confidence interval of the HR_{trt}^{DR}.

Greedy 1:1 with caliper

The method applied is the same as the previous one but matching is performed on a subset of the whole dataset. Because of a caliper set at 0.1, only the patients that share a distance smaller than or equal to 0.1 can be potential pairs. From Figure

3.5, we can see that there is a great reduction in the number of possible pairs: the columns on the left hand side of the red bar represent the percentage of the possible matches after the application of the caliper.

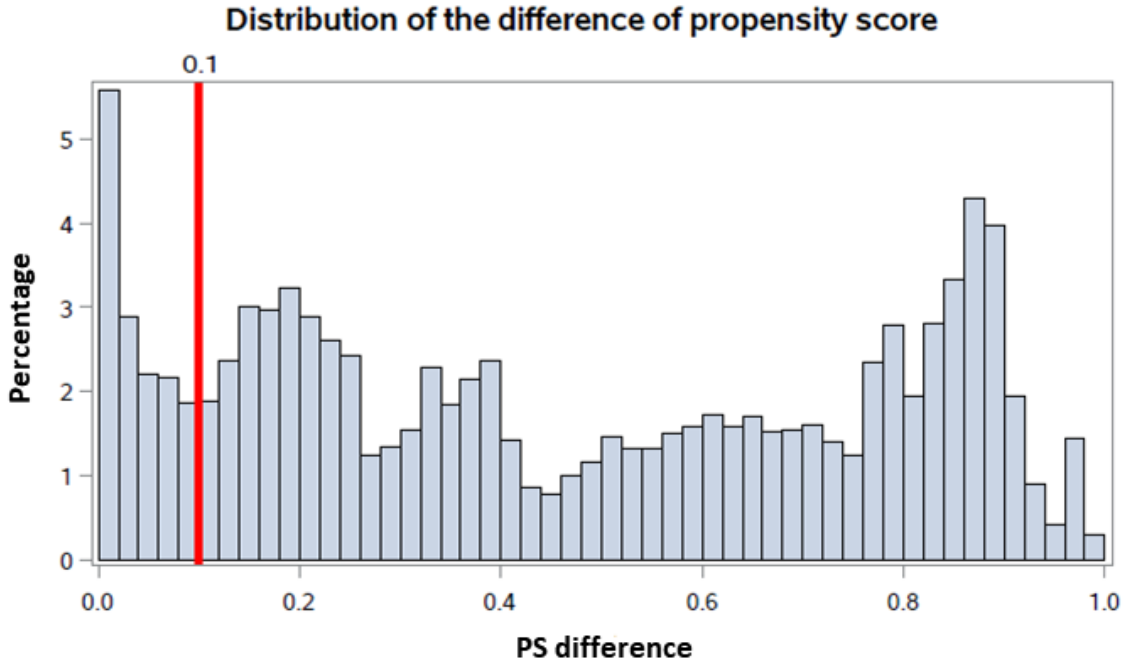


Figure 3.5: Distribution of the difference of PS among all the possible pairs in the dataset; the red bar indicates the value at which the caliper has been set.

After the application of the greedy 1:1 matching with caliper, we obtain 54 pairs for a total of 108 patients (the PS distribution in the matched dataset is in Figure 3.6).

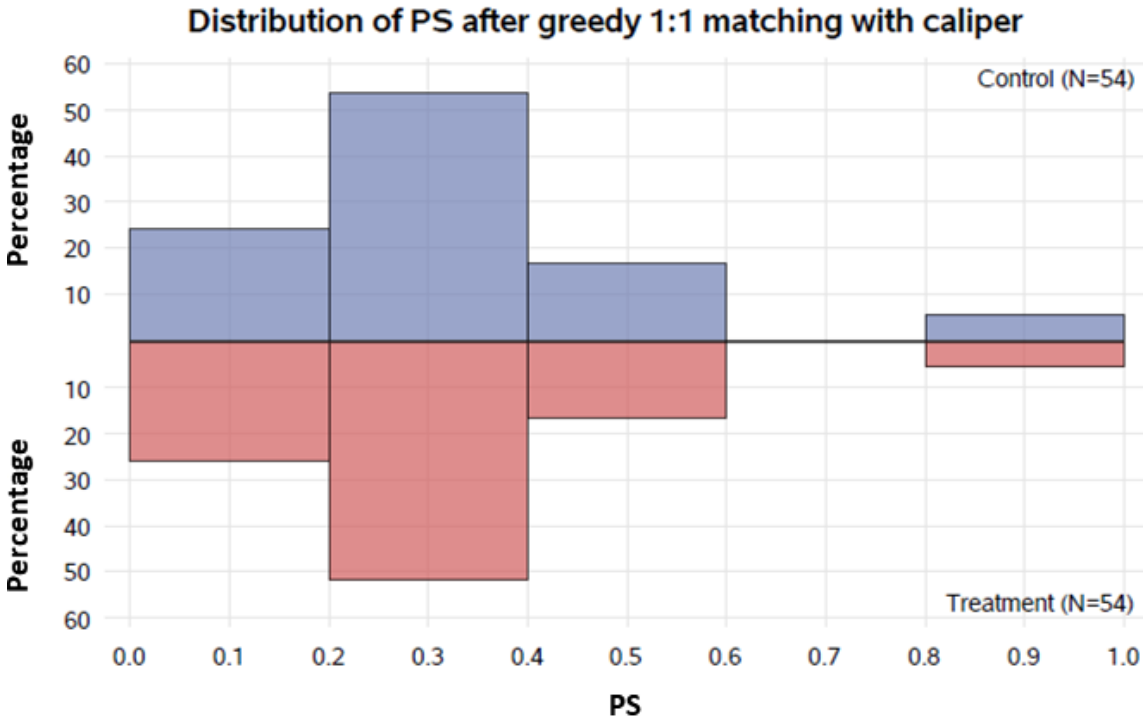


Figure 3.6: Distribution of the propensity score within each treatment arm after the greedy 1:1 matching with caliper.

Figure 3.7 shows the two Kaplan Meier curves.

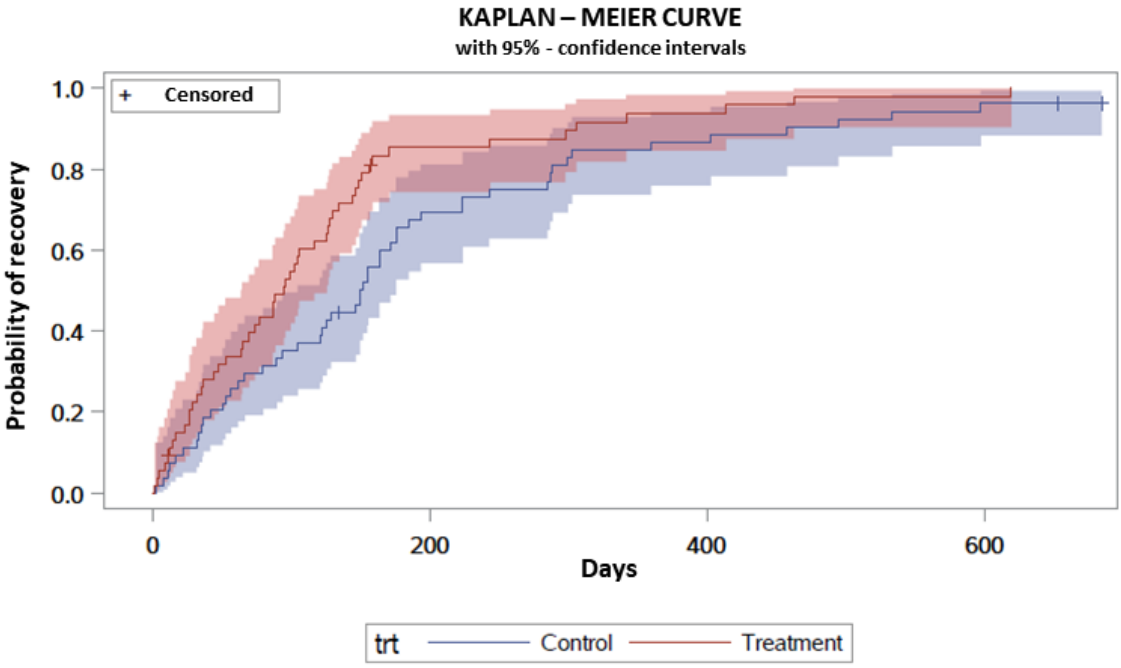


Figure 3.7: Kaplan - Meier curves for control and treatment groups on the greedy 1:1 with caliper matched dataset.

The results for the hazard ratios of the treatment with and without "doubly robust" approach are:

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.629, \quad \text{p-value} = 0.0148$$

with $[1.100; 2.412]$ as the 95%-confidence interval of the HR_{trt};

$$HR_{trt}^{DR} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 2.152, \quad \text{p-value} = 0.0004$$

with $[1.410; 3.284]$ as the 95%-confidence interval of the HR_{trt}^{DR}.

Optimal 1:1

All the pairs are determined to minimize the sum of the distances between pairs. For this purpose, the optimisation problem in Chapter 2 has to be solved: in R, the "optmatch" package uses the RELAX-IV minimum cost flow solver to solve it.

The number of pairs is 100 for a total of 200 patients, there is no reintroduction of the "already-matched" individuals (the PS distribution in the matched dataset is in Figure 3.8).

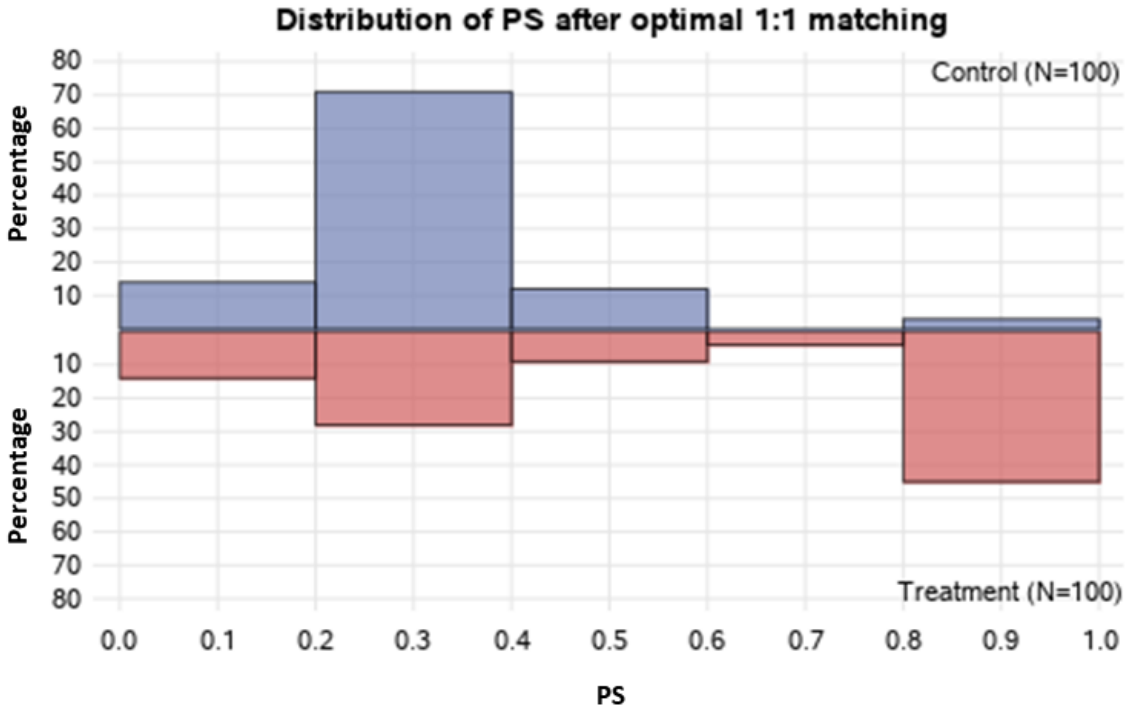


Figure 3.8: Distribution of the propensity score within each treatment arm after the optimal 1:1 matching.

Figure 3.9 shows the two Kaplan Meier curves.

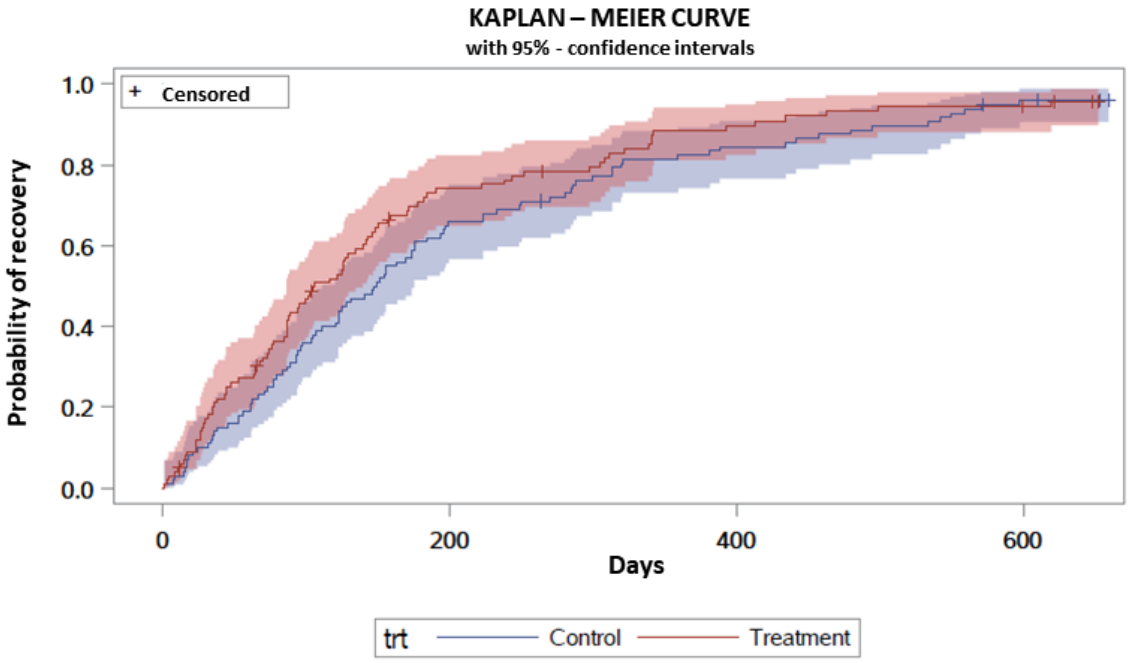


Figure 3.9: Kaplan - Meier curves for control and treatment groups on the optimal 1:1 matched dataset.

The results for the hazard ratios of the treatment with and without "doubly robust" approach are:

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.217, \quad \text{p-value} = 0.1826$$

with $[0.912; 1.624]$ as the 95%-confidence interval of the HR_{trt};

$$HR_{trt}^{DR} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.934, \quad \text{p-value} = 0.0001$$

with $[1.379; 2.713]$ as the 95%-confidence interval of the HR_{trt}^{DR}.

Optimal 1:1/2

The method applied is the same as before but this time each patient in the treatment arm can be matched to 1 or 2 individuals - according to the result of the optimisation problem - in the control group. In this way, a higher number of controlled patients is

included in the treatment effect analysis (the PS distribution in the matched dataset is in Figure 3.10).

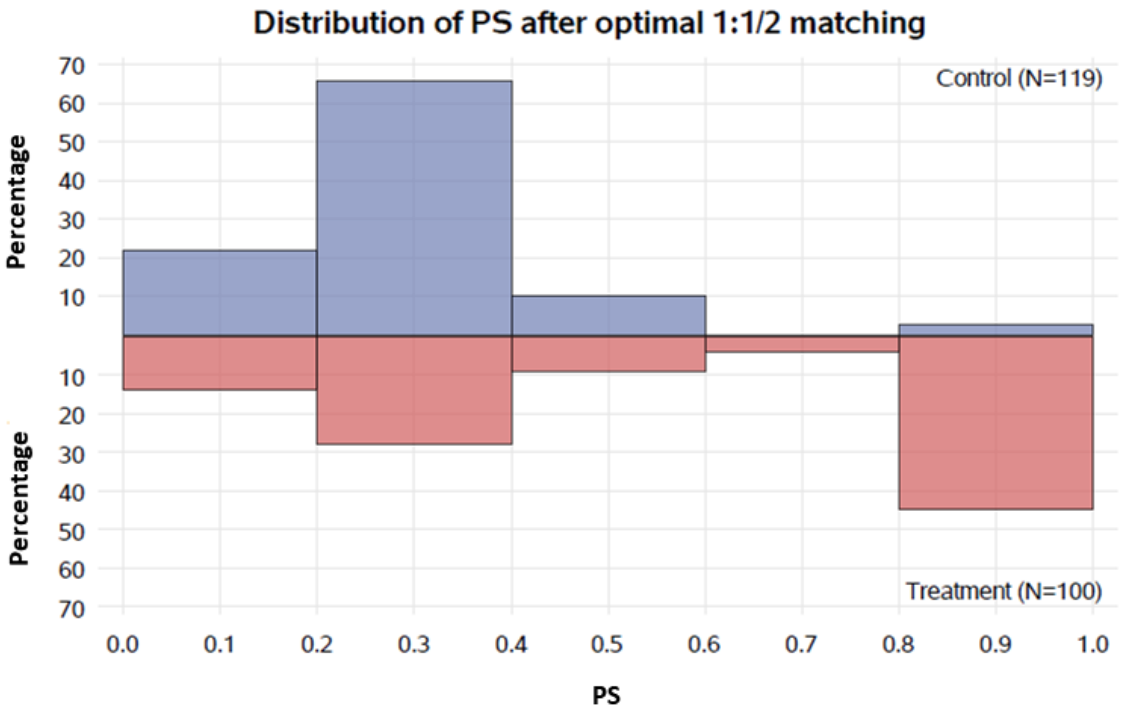


Figure 3.10: Distribution of the propensity score within each treatment arm after the optimal 1:1/2 matching.

Figure 3.11 shows the two Kaplan Meier curves.

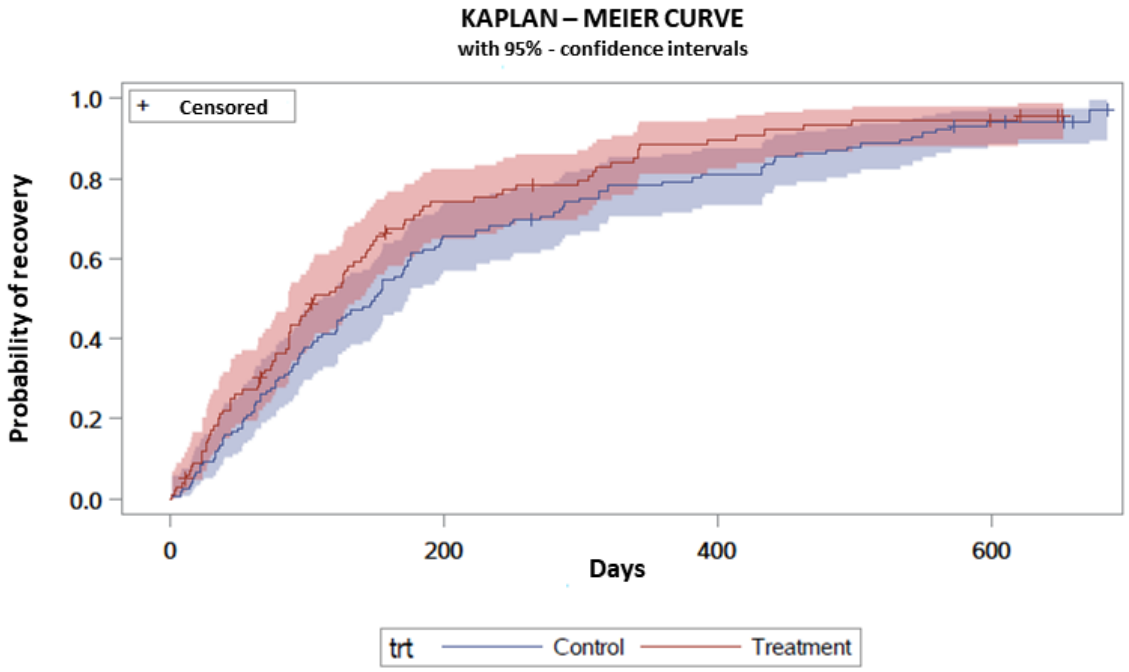


Figure 3.11: Kaplan - Meier curves for control and treatment groups on the optimal 1:2 matched dataset.

The results for the hazard ratios of the treatment with and without "doubly robust" approach are:

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.274, \quad \text{p-value} = 0.0880$$

with $[0.965; 1.684]$ as the 95%-confidence interval of the HR_{trt};

$$HR_{trt}^{DR} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 2.001, \quad \text{p-value} < 0.0001$$

with $[1.432; 2.796]$ as the 95%-confidence interval of the HR_{trt}^{DR}.

Description of the PS differences for all approaches

Table 3.3 shows the statistics of the difference in PS between among the different matched sets in the different matching approaches.

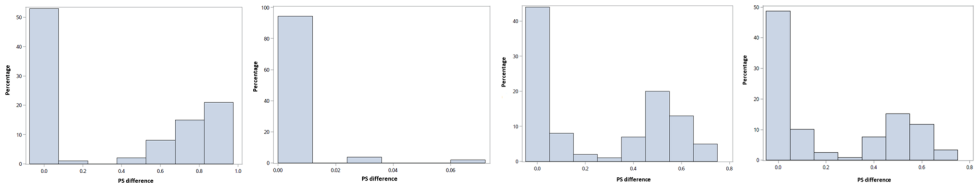
MATCHING	Greedy 1:1 without caliper	Greedy 1:1 with caliper = 0.1	Optimal 1:1	Optimal 1:1/2
MIN	< 0.0001	0.0002	< 0.0001	< 0.0001
MEAN	0.3618	0.0051	0.2602	0.2189
MEDIAN	0.0090	0.0023	0.0999	0.0612
MAX	0.9606	0.0684	0.7020	0.7291
STANDARD DEVIATION	0.3983	0.0104	0.2645	0.2536
DISTRIBUTION				

Table 3.3: Statistics of the difference in PS among matched sets in the different matching approaches.

3.5.3 Inverse probability of treatment weighting (IPTW)

After the computation of the PS for all the patients, we assign a (general or stabilised) weight to each individual, according to the arm he belongs to. These weights represent how many times an observation in the true dataset counts in the pseudo-population: the sum of all the weights gives the size of the pseudo-population.
After that, for each type of weights used, we obtain the weighted Kaplan-Meier curves for treatment and control arms and the hazard ratio of the treatment by a weighted Cox model, i.e. the same methods for the treatment effect estimation we have used so far, applied on the pseudo-population.

General weights

We have obtained a pseudo-population of about 849 individuals (462 in the treatment arm and 387 in the control group). The distribution of the PS between the two groups changes as showed in Figure 3.12.

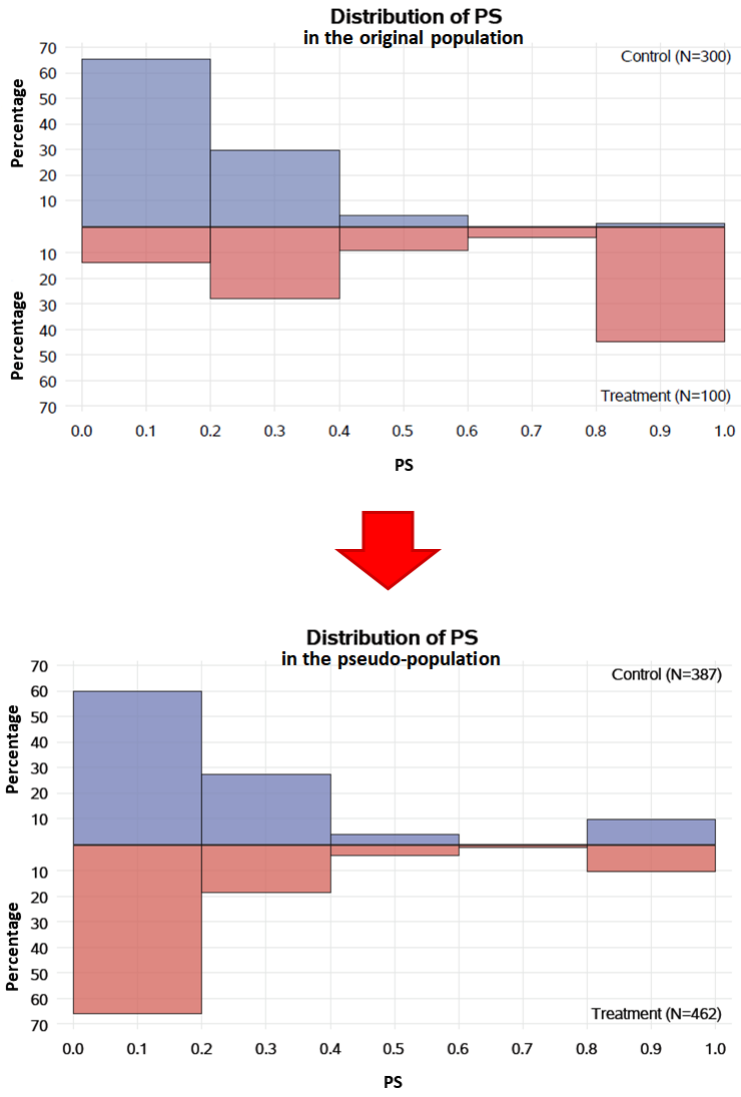


Figure 3.12: Change in the distribution of the PS between the two arms with IPTW general weights.

Figure 3.13 shows the two weighted Kaplan Meier curves.

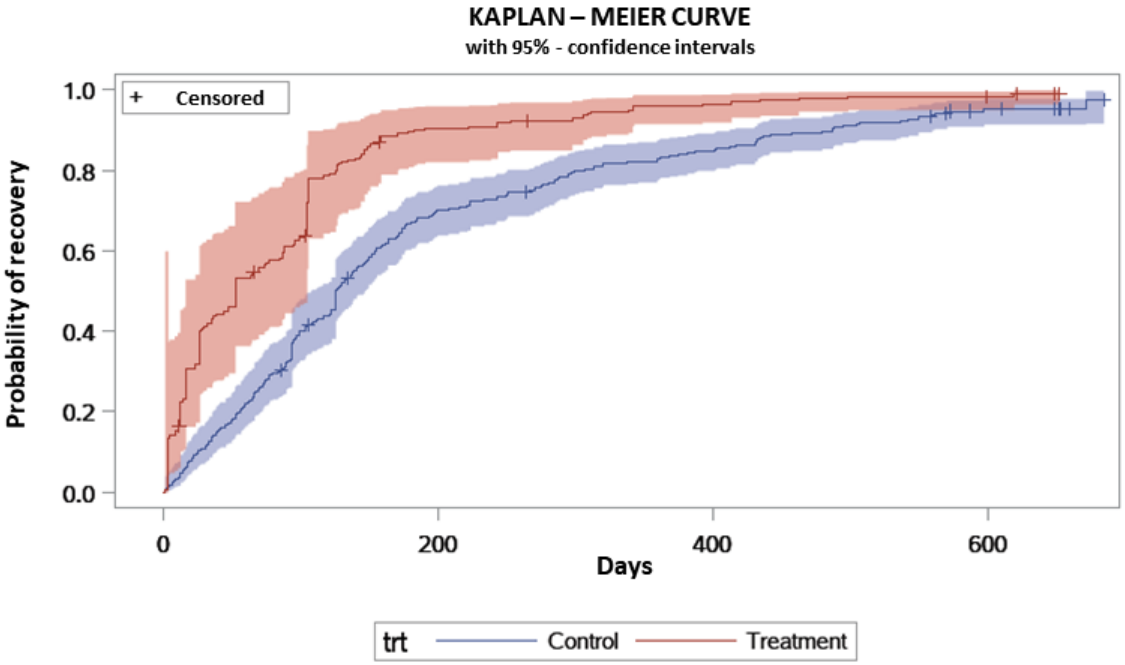


Figure 3.13: Weighted Kaplan - Meier curves for control and treatment groups on the dataset with general IPTW weights.

The results for the hazard ratios of the treatment with and without "doubly robust" approach are:

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.996, \quad \text{p-value} < 0.0001$$

with $[1.733; 2.300]$ as the 95%-confidence interval of the HR_{trt};

$$HR_{trt}^{DR} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 3.080, \quad \text{p-value} < 0.0001$$

with $[2.607; 3.638]$ as the 95%-confidence interval of the HR_{trt}^{DR}.

Stabilised weights

We have obtained a pseudo-population of about 406 individuals (116 in the treatment arm and 290 in the control group). The distribution of the PS between the two groups changes as showed in Figure 3.14.

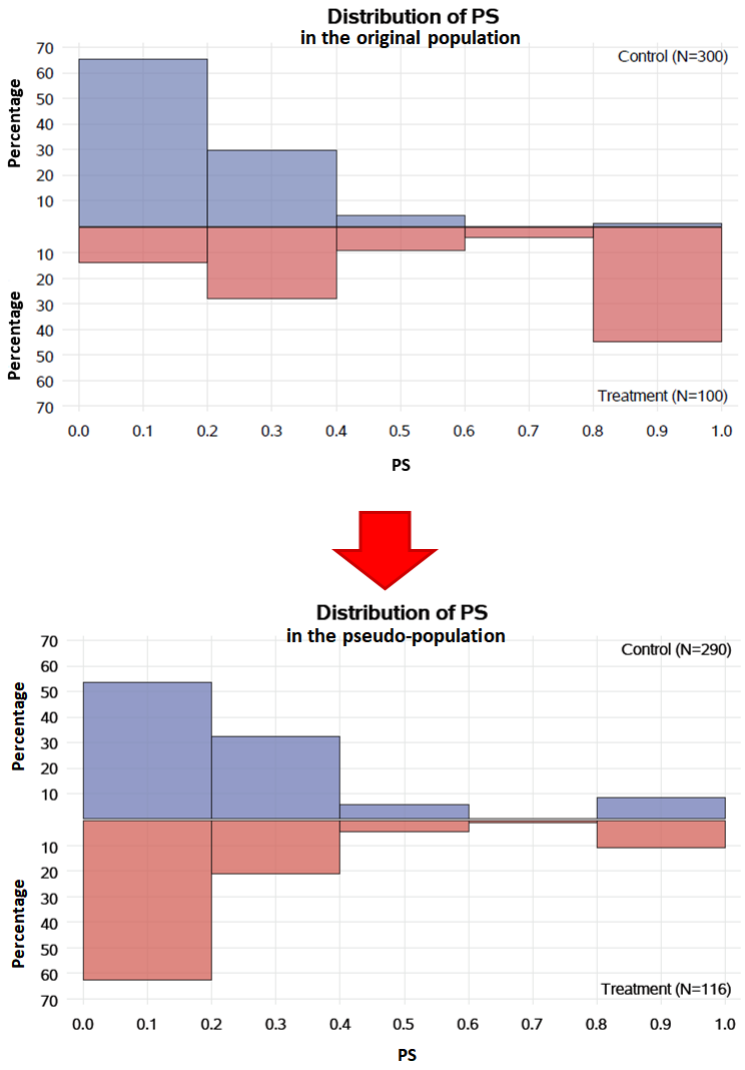


Figure 3.14: Change in the distribution of the PS between the two arms with IPTW stabilised weights.

Figure 3.15 shows the two weighted Kaplan Meier curves.

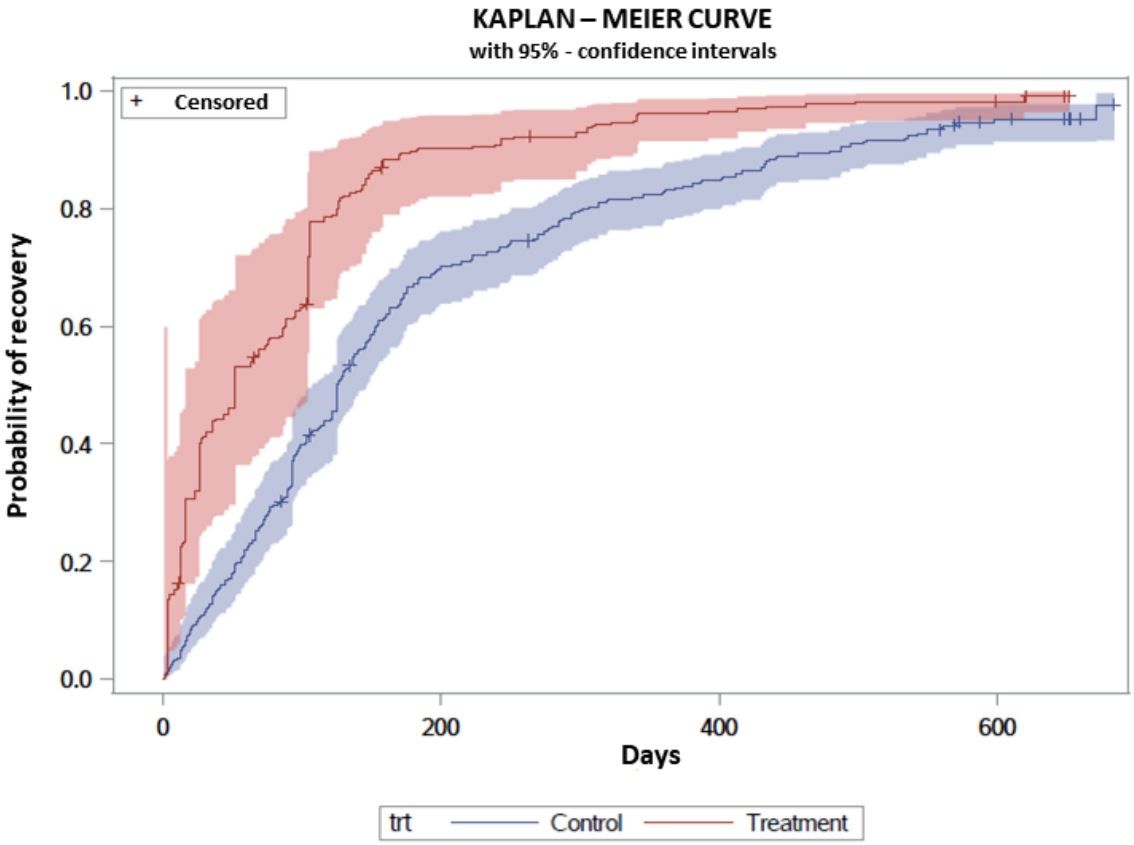


Figure 3.15: Weighted Kaplan - Meier curves for control and treatment groups on the dataset with stabilised IPTW weights.

The results for the hazard ratios of the treatment with and without "doubly robust" approach are:

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 2.094, \quad \text{p-value} < 0.0001$$

with $[1.674; 2.620]$ as the 95%-confidence interval of the HR_{trt};

$$HR_{trt}^{DR} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 2.630, \quad \text{p-value} < 0.0001$$

with $[2.060; 3.358]$ as the 95%-confidence interval of the HR_{trt}^{DR}.

3.5.4 Stratification

After the PS has been computed for each patient, the total dataset is ordered by PS and divided into mutually exclusive groups (strata) based on the PS. Then, a stratified analysis is performed to assess the treatment effect: the two Kaplan-Meier curves, one for each stratum and for both treatments, and a stratified Cox model.

The whole dataset was divided into three strata. Then, we chose between two types of strata: PS quantiles or equal sized groups. Since the PS distribution is unbalanced towards extreme values, strata of both types are inevitably heterogeneous in the number of treated and control patients (see Table 3.4).

	1ST STRATUM		2ND STRATUM		3RD STRATUM	
	CONTROL	TREATMENT	CONTROL	TREATMENT	CONTROL	TREATMENT
PS QUANTILES	263	30	34	21	3	49
EQUAL SIZED GROUPS	130	3	109	24	61	73

Table 3.4: Distribution of the patients belonging to the treatment and the control arms, according to the different types of stratification.

We decided to use PS quantiles because the three strata seem to be less heterogeneous in terms of treated and controlled patients. Figure 3.16 shows the distribution of the PS within each tertile with a stratification by PS quantiles.

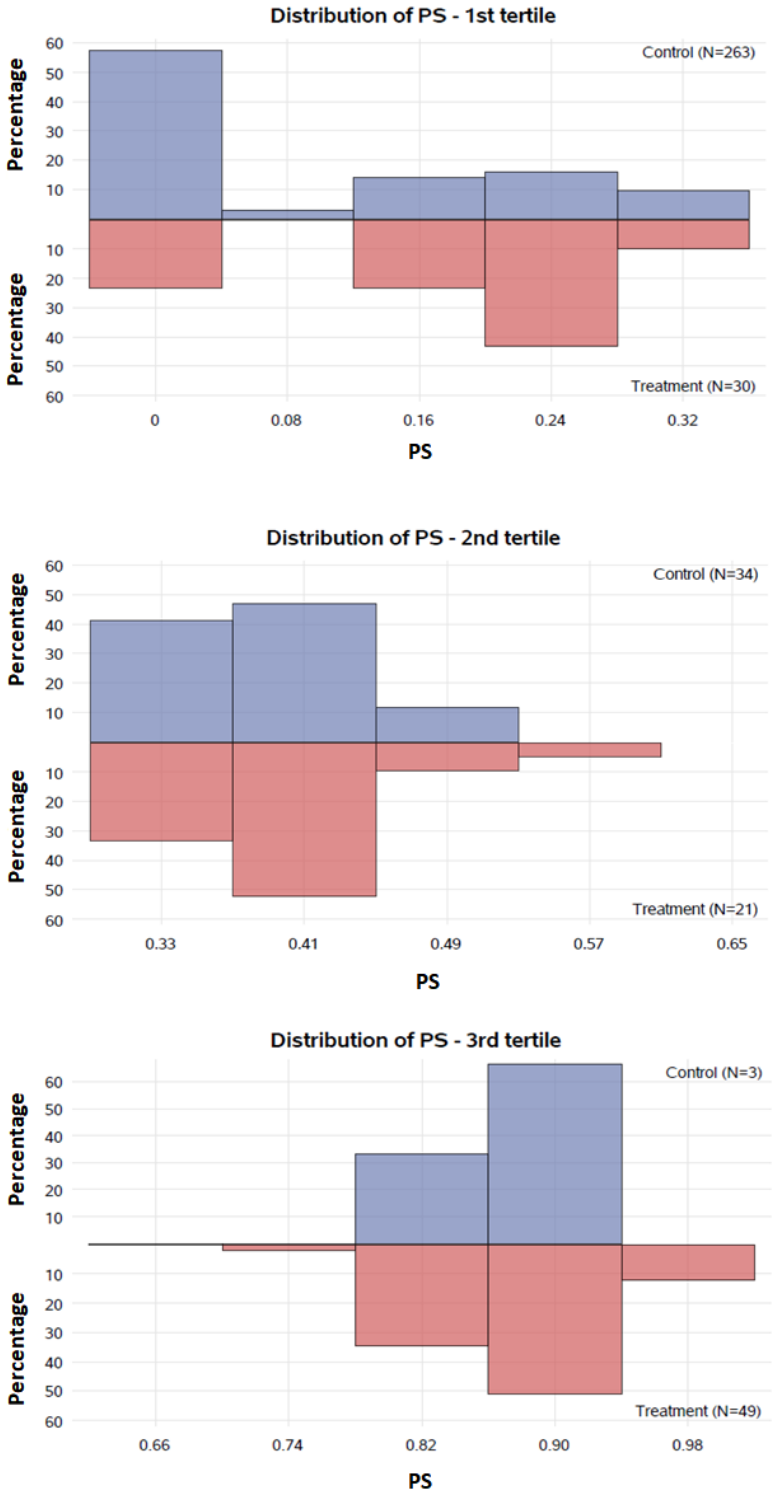


Figure 3.16: Distribution of the PS within each of the three PS quantiles.

For each tertile the two Kaplan Meier curves have been calculated; in particular, in the third tertile, there are only 3 controlled patients, so the estimation of the curve is not accurate (see Figure 3.17)

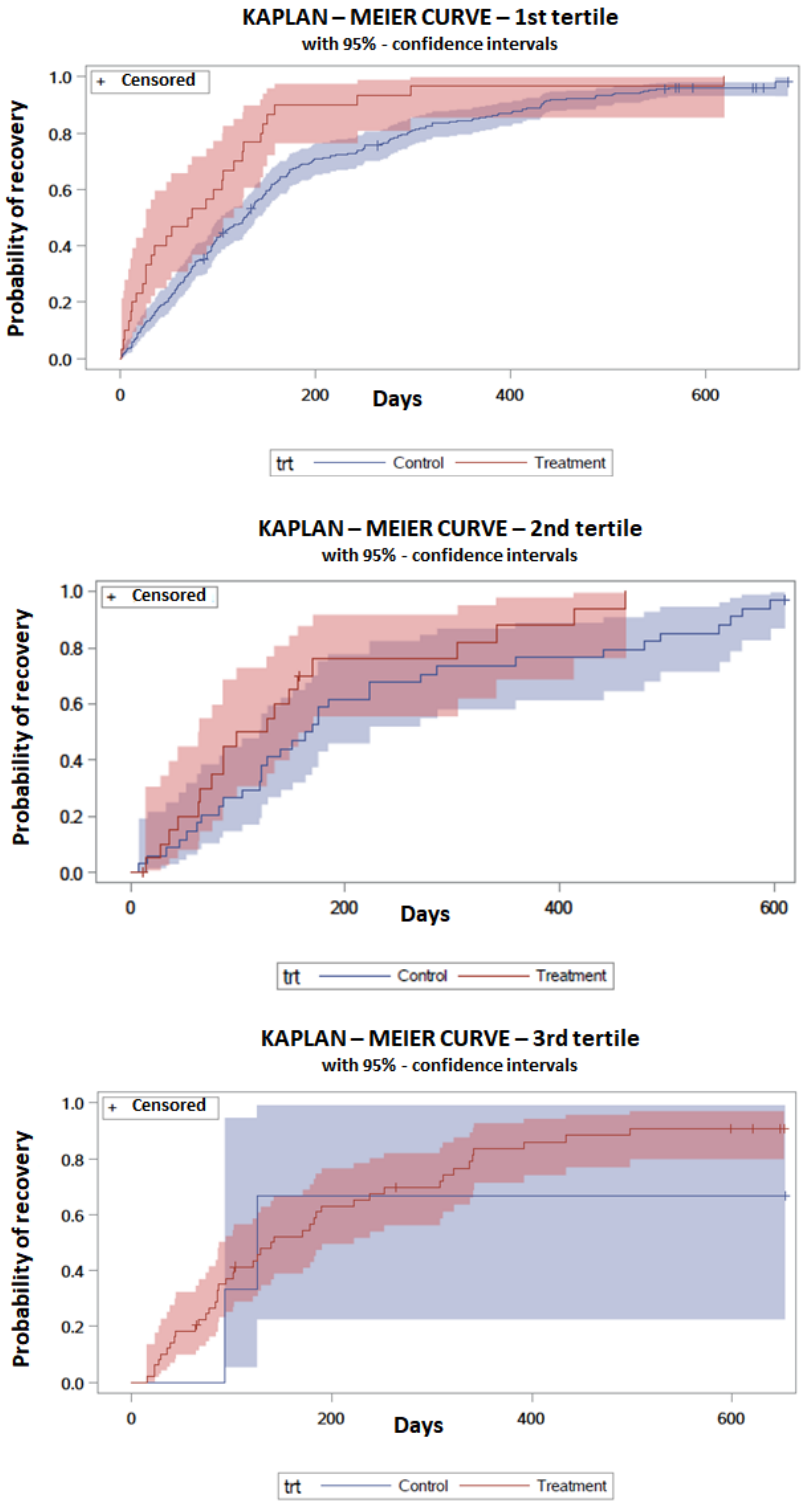


Figure 3.17: Kaplan - Meier curves for control and treatment groups on each of the three PS quantiles.

The results for the hazard ratios of the treatment with and without "doubly robust"

approach are:

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.798, \quad \text{p-value} = 0.0003$$

with $[1.312; 2.466]$ as the 95%-confidence interval of the HR_{trt};

$$HR_{trt}^{DR} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 2.102, \quad \text{p-value} = 0.0007$$

with $[1.501; 2.942]$ as the 95%-confidence interval of the HR_{trt}^{DR}.

3.5.5 Matching and pair-stratified Cox model

After applying a matching technique to the dataset, a "pair-stratified" Cox model can be applied. In this case, since each pair (or small matched set) is a stratum, there is a risk to overfit the model, because the Cox model implies that each stratum has its own baseline hazard function. The use of strata can relax the constraint that there is the same baseline hazard function for all the patients; but, if no caliper is applied in the matching method, this can lead to patients with really different baseline characteristics in the same pair (stratum) that share the same baseline hazard function. Furthermore, since strata are very small, there may be the case where all the patients within a stratum are censored, and so this stratum gives no contribution to the estimation of the parameters of the Cox model.
Nevertheless, we have applied a "pair-stratified" analysis after each of the matching technique used in the previous subsection and we have obtained the following results:

- **greedy 1:1**

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 0.902, \quad \text{p-value} = 0.6122$$

with $[0.606; 1.344]$ as the 95%-confidence interval of the HR_{trt};

$$HR_{trt}^{DR} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.917, \quad \text{p-value} = 0.0272$$

with $[1.076; 3.415]$ as the 95%-confidence interval of the HR_{trt}^{DR};

- **greedy 1:1 with caliper**

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.650, \quad \text{p-value} = 0.0772$$

with $[0.947; 2.875]$ as the 95%-confidence interval of the HR_{trt};

$$HR_{trt}^{DR} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.803, \quad \text{p-value} = 0.0525$$

with $[0.994; 3.272]$ as the 95%-confidence interval of the HR_{trt}^{DR};

- **optimal 1:1**

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.487, \quad \text{p-value} = 0.0553$$

with $[0.991; 2.232]$ as the 95%-confidence interval of the HR_{trt};

$$HR_{trt}^{DR} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 2.547, \quad \text{p-value} = 0.0032$$

with $[1.369; 4.738]$ as the 95%-confidence interval of the HR_{trt}^{DR};

- **optimal 1:1/2**

$$HR_{trt} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.267, \quad \text{p-value} = 0.2273$$

with $[0.863; 1.862]$ as the 95%-confidence interval of the HR_{trt};

$$HR_{trt}^{DR} = \frac{\lambda(t \,|\, trt = \text{"Treatment"})}{\lambda(t \,|\, trt = \text{"Control"})} = 1.904, \quad \text{p-value} = 0.0198$$

with $[1.108; 3.272]$ as the 95%-confidence interval of the HR_{trt}^{DR}.

Discussion and conclusion

This report has focused on the importance of using particular statistical methods, PS based methods and covariate adjustmen, when dealing with non-randomised trials, for example, in the case of rare diseases. When historical controls are used, in order not to have a treatment effect estimation biased by the possible confounding effect of baseline covariates, these methods have to be implemented.

The results obtained in the previous chapter are summarised in Figures 3.18 and 3.19.

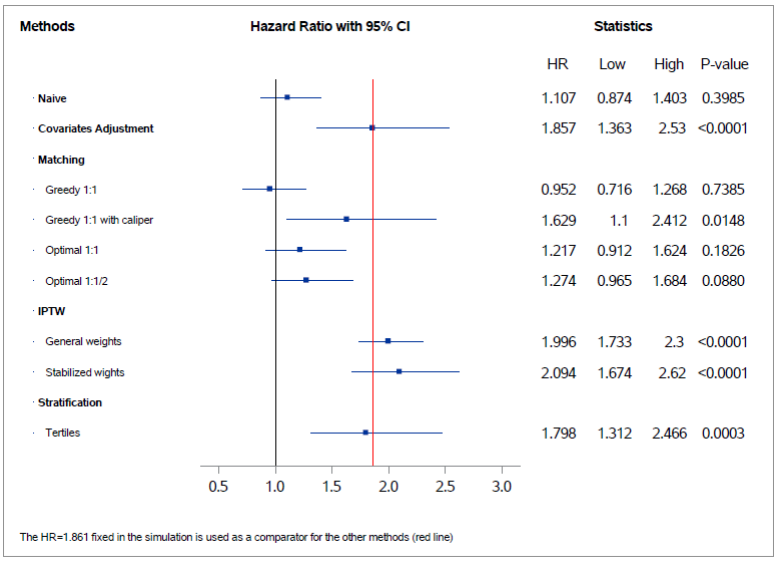


Figure 3.18: Forest plot that summarises the estimations of the hazard ratio of the treatment, its confidence interval and its p-value obtained by the different statistical methods considered in this report.

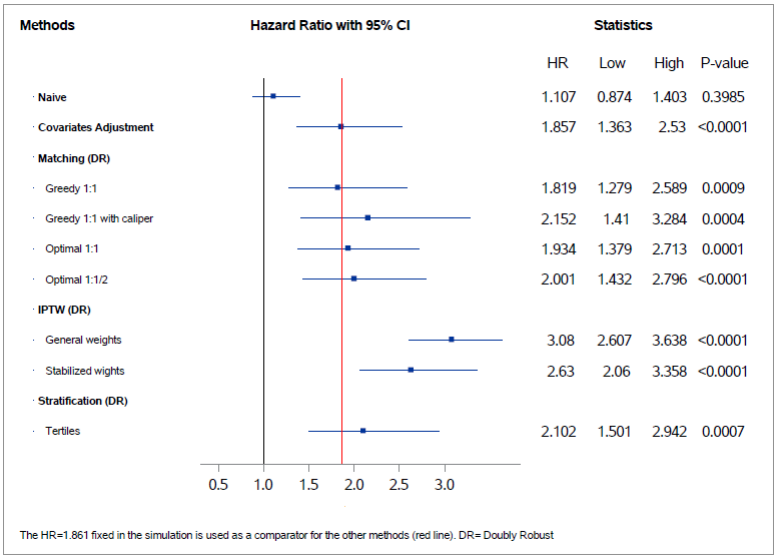


Figure 3.19: Forest plot that summarises the estimations of the hazard ratio of the treatment, its confidence interval and its p-value, obtained by the different statistical methods considered in this report with a doubly robust approach.

Looking at Figure 3.18, we see that covariate adjustment performs really well but this can be due to the parametric model used to simulate the confounding effect of $V4$.

Regarding the different matching techniques without caliper, there is not a great improvement with respect to the naïve method, meaning that confounding is still present. Our results confirm the importance of a caliper when a matching approach is applied, in order to get a treatment effect estimation that is closer to the truth. In the case of the greedy 1:1 matching with caliper, the confidence interval is wider than in the other matching techniques because the size of the matched dataset is reduced by the use of the caliper.

Good results are obtained also with IPTW and stratification. Both types of IPTW weights performed well but the general ones allowed to obtain a smaller confidence interval because of an inflated pseudo-population. As a consequence, stabilised weights are preferred because the width of the confidence interval reflects the size of the actual population.

Stratification performed well even though tertiles were heterogeneous. An increase in the number of strata may have given better results, but the choice of tertiles reflected the distribution of PS among patients (see Figure 3.2).

As expected, the "doubly robust" approach further reduced the confounding effect (see Figure 3.19) so that also the matching techniques without a caliper gave good results; but there is a worsening in the results obtained with IPTW because the weights modify the distribution of the PS, and so the one of all the baseline covariates that are included in the Cox model.

In conclusion, in order to understand which method is the best one, simulations on other scenarios should be done and, furthermore, an application on real data would allow to understand the real performance of these methods in reducing the effect of observed confounders. In any case, when registry data have to be used as control arm, we should not focus on only one method to reduce confounding but other methods should be applied as sensitivity analyses.

Appendices

Appendix A

Missing data

There are three categories of missing data:

- **missing completely at random (MCAR)** → missing data can be seen as a random sample of the total dataset, that is to say, they are not specific of a certain category of patients. In this case, missing data do not add bias but they only reduce the power of the study;
- **missing at random (MAR)** → missing data depend on known variables, so if these variables are considered in the analysis, the presence of missing data will not bias the analysis. As an example, the availability of the results of certain tests is related to what is covered by patients' health insurance, that is a known characteristic for each subject;
- **missing not at random (MNAR)** → missing data depend on unobserved covariates and they can introduce bias in the analysis. For example, suppose that certain patients affected by type 2 diabetes are hospitalized because of a high value of glycated hemoglobin, then they could not return for a scheduled follow-up visit at which glycated hemoglobin has to be measured. These probable missing values of glycated hemoglobin would be different from the other observed values because of the reason they are missing (hospitalization).

There are several ways to handle missingness but these methods depend on the kind of missing data we have to deal with, and it is not always easy to understand if they

belong to MCAR, MAR or MNAR.
The simplest strategy is using **complete-case analysis**: only the patients with complete data are used in the analysis and patients with missing records are discarded. This approach is sufficient to get consistent estimates of regression coefficients and valid inferences when missing data are MCAR or in the case when the missing variables do not depend on the response variable. The main disadvantages refer to the loss of valuable information when discarding patients with missing records and the decrease in efficiency when missingness affects several variables.

A different method is filling values with imputed values (**imputation**) and then analyse the complete dataset with traditional statistical methods. The most used approaches are **unconditional mean imputation** and **conditional-mean imputation**. In the first case, the missing values of a certain variable are replaced by the arithmetic mean of its observed records. As a result, the mean of the variable does not change, but there is a decrease in its variance and its covariance with other variables. On the other hand, this approach can bias regression coefficients and it can lead to invalid inferences.

The second approach aims at substituting the blank spaces left by missing data with predicted values obtained from a regression model. Each variable with missing records is regressed on the other covariates, using the complete available data. After that, the predicted values are used to complete the dataset before the analysis. In any case, even this strategy has negative aspects: imputed values are less variable than the real data because of the lack of the residual variation and the uncertainty of the estimates of the regression coefficients is not taken into account.

When we have to deal with MAR missing data, a suggested impute method is the so-called "**multiple imputation**". In order to take into account the uncertainty related to the missingness of certain values, this approach produces several complete datasets. In each dataset, the missing values are replaced by different imputed

values; after that, for each dataset, the parameters of interest and the standard errors are estimated. Finally, a common estimation of the parameters is got by averaging the single estimations mentioned before; standard errors are combined too, considering the variation among the estimates in the imputed datasets.

Appendix B

Code

B.1 R code

```
############################################################
# BASELINE COVARIATES AND TIME-TO-EVENT DATASET SIMULATION #
############################################################


##control=300 and treatment=100#


# CLEAN WORKING ENVIRONMENT ------------------------------------------
rm(list=setdiff(ls(), c()))


# WORKING DIRECTORY --------------------------------------------------
wd <- "C:/Users/635185/Desktop/Poli/Tesi␣magistrale"
setwd(wd)
```

```
# FIX THE SEED (once and for all)
    ------------------------------------------------------
set.seed(1234)
library(survival)
### GENERATION OF THE COVARIATES (including the treatment)


# total nb of patients
total=400

# init dataset
dataset=array(dim=c(total, 11))
colnames(dataset) = c("count", "trt", "sex", "age","V1","V2","V3","V4","V5"
    , "Event_time", "Censor")

### GENERATION OF THE COVARIATES
for(i in 1:total){
  dataset[i,1]=i

  # Treatment
  if (i<=100) {dataset[i,2]=1}
  else {dataset[i,2]=0}

  # Sex
  dataset[i,3]=rbinom(1, 1, 0.5)

  # Age0
  dataset[i,4]=rnorm(1, mean=65, sd=9)

  # Covariates V1, V2, V3
  dataset[i,5]=rnorm(1, mean=8, sd=0.6)
  dataset[i,6]=rnorm(1, mean=27.5, sd=1.4)
```

```
  dataset[i,7]=round(rnorm(1, mean=3, sd=0.7),digits = 0)

  # Covariate V4
  dataset[i,8]=round(rnorm(1, mean=2.57*dataset[i,2]+3.51*(1-dataset[i,2]),
       sd=0.5*dataset[i,2]+0.5*(1-dataset[i,2])),digits = 0)

  # Covariate V5
  dataset[i,9]=rnorm(1, mean=150+1500*(1-dataset[i,3])+2400*dataset[i,3],
      sd=300)
}

dataset=data.frame(dataset)

### GENERATION OF THE RESPONSE VARIABLE (Time to event, Weibull)
#lambda=1/720 * exp(-0.692*dataset$trt+0.4*dataset$V4)
lambda=1/720 * exp(0.4*dataset$trt+0.4*dataset$V4)
# lambda=1/720 * exp(-0.692*dataset$trt) # just to check, if there are no
    confounder, if the Cox model provides correct results

# Survival times
s = rexp(n=total, rate= lambda)

# Maximum follow-up times (for the administrative censor)
accrual = 170 # patients are recruited between 0 and 170 days (~ 0.5 year)
    after the start of the study
fup = 720 - 170 # patients are followed-up for a minimum of 550 days (~ 1.5
     year ) => end of the study after 2 years
tfup = runif(total) * accrual + fup

# Random uniform censor for 2% of the patients, lost to follow-up
lost_fup = rbinom(total, 1, 0.02)
tfup2 = runif(1)*s * lost_fup + 720 * (1-lost_fup)
```

```
# mean(tfup2!=s) # to check


# Censored survival times

patients.times.sim = pmin(s,tfup, tfup2);


# Censoring indicator

patients.cens.sim =(pmin(tfup, tfup2)>s)



dataset[, 10]=patients.times.sim

dataset[, 11]=patients.cens.sim


#Round days to integer number

dataset[,10]<-ceiling(dataset[,10])


for (i in 400){

  if (dataset[i,]$Censor=='FALSE') dataset[i,]$Censor=0

  else dataset[i,]$Censor=1

}

#Save dataset as an excel file

write.csv(dataset, "survival.csv")



########################################

# OPTIMAL MATCHING WITH PROPENSITY SCORE#

########################################


#clean environment

rm(list=setdiff(ls(),c()))


#working directory

wd<-'C:/Users/635185/Desktop/Poli/Tesi␣magistrale'
```

```
#load dataset
basedata<-read.csv(file="C:/Users/635185/Desktop/Poli/Tesi␣magistrale/
    survival.csv",header=TRUE, sep=",")

##PROPENSITY SCORE ESTIMATION##
psm<-glm(factor(trt)~factor(sex)+age+V1+V2+V3+V4+V5, family=binomial, data=
    basedata)

#create a matrix with the differences between patients' propensity scores
install.packages("optmatch")
library(optmatch)
diff <- match_on(psm, method="euclidean")
for (i in 1:100){
    for (j in 1:300) {
      diff[i,j]=abs(psm$fitted.values[i]-psm$fitted.values[100+j])
    }
}

##OPTIMAL 1:1 MATCHING##
matched<-pairmatch(diff,data=basedata)
print(matched, grouped=TRUE)
#to make sure observations are in the proper order:
all.equal(names(matched), row.names(basedata))
#give to each patient the number of the pair he belongs to
matched_db<-cbind(basedata, matches=matched)
#remove unmatched patients
matched_db <- na.omit(matched_db)
#export dataset with matchings
write.csv(matched_db, "C:/Users/635185/Desktop/Poli/Tesi␣magistrale/opt_
    pairs.csv")
```

```
##OPTIMAL 1:1/2 MATCHING##
matched_2<-fullmatch(diff,min.controls=1,max.controls=2,data=basedata,
    remove.unmatchables = FALSE)
print(matched_2, grouped=TRUE)
# to make sure observations are in the proper order:
all.equal(names(matched_2), row.names(basedata))
#give to each patient the number of the pair he belongs to
matched_db_2<-cbind(basedata, matches=matched_2)
#remove unmatched patients
matched_db_2<- na.omit(matched_db_2)
#export dataset with matchings
write.csv(matched_db_2, "C:/Users/635185/Desktop/Poli/Tesi␣magistrale/opt_
    pairs_2.csv")
```

B.2 SAS code

```
**********************
* GREEDY 1:1 MATCHING *
**********************
*Authors:  Yinpu Chen and Margherita Annaratone

/*formats*/
proc format;
        value   sexid 0 = 'F'
                      1 = 'M'
                      ;
        value   trtid 0 = 'Control'
                      1 = 'Treatment'
                      ;
run;
```

```
*****************************************************************;
*****************Import dataset(baseline+survival)**************;

PROC IMPORT OUT= WORK.survival
            DATAFILE= "/folders/myfolders/tesi/survival.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;

data survival;
set survival;
format sex sexid. trt trtid.;
drop var1;
run;

%let total_t=100;
%let total_c=300;
%let total=%sysevalf(&total_t+&total_c);


proc means data=survival;
class trt;
run;

*************************************************************;

*Kaplan - Meier without any adjustments;
proc lifetest data=survival atrisk plots=survival(failure cl);
```

```
          time event_time*censor(0);
          strata trt / test=logrank;
run;


*Cox model with Naive method;
proc phreg data=survival;
          class trt(ref='Control');
          model event_time*censor(0) = trt;
          hazardratio trt /cl=wald diff=ref;
run;


*The HR I set in the simulation;
proc phreg data=survival ;
          class trt(ref='Control');
          model event_time*censor(0) = trt v4;
          hazardratio trt /cl=wald diff=ref;
run;


*Cox model with Covariate adjustment;
proc phreg data=survival;
          class trt(ref='Control');
          model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5;
          hazardratio trt /cl=wald diff=ref;
run;

**************************PS computation**************;
```

```
proc logistic data=survival;
        class sex trt;
        model trt(event='Treatment') = sex age v1 v2 v3 v4 v5;
        output out = ps pred = ps xbeta = logit_ps;
run;


/*Histogram for PS distribution within the two arms*/

data histo;
set ps;
if trt=0 then c=ps;
else t=ps;
run;


%let gpath="C:\Users\635185\Desktop\Poli\Tesi magistrale";
%let dpi=200;
ods html close;
ods listing image_dpi=&dpi;
/*--MirrorHistograms-Vertical--*/
proc template;
  define statgraph MirrorHistogramVert;
    dynamic _binwidth;
begingraph;
entrytitle 'Distribution of PS';
layout lattice / columndatarange=union rowgutter=0;
columnaxes;
      columnaxis / display=(tickvalues) griddisplay=on
```

```
            linearopts=(tickvaluesequence=(start=0 end=1
            ↪ increment=0.1) tickvaluepriority=true);
    endcolumnaxes;
layout overlay / walldisplay=none xaxisopts=(griddisplay=on)
                        xaxisopts=(griddisplay=on)
                yaxisopts=(griddisplay=on display=(tickvalues label)
                   linearopts=(tickvaluesequence=(start=10 end=70
                   ↪ increment=10) tickvaluepriority=true));
  histogram c /  binstart=0.1 binwidth=_binwidth binaxis=false
            fillattrs=graphdata1 datatransparency=0.3;
              entry halign=right 'Control (N=300)' /  valign=top;
  endlayout;
layout overlay / walldisplay=none
                yaxisopts=(reverse=true griddisplay=on
                ↪ display=(tickvalues label)
                linearopts=(tickvaluesequence=(start=10 end=70
                ↪ increment=10) tickvaluepriority=true))
                xaxisopts=(griddisplay=on);
    histogram t /  binstart=0.1 binwidth=_binwidth binaxis=false
          fillattrs=graphdata2 datatransparency=0.3;
            entry halign=right 'Treatment (N=100)' /  valign=bottom;
   endlayout;
          endlayout;
        endgraph;
  end;
run;


ods graphics / reset width=5in height=3in imagename='histo_logistic';
proc sgrender data= work.histo template=MirrorHistogramVert;
```

```
  dynamic _binwidth=0.08;
run;

*************************************************************************;
*********************************1:1 greedy
↪  matching*********************************;
*the dataset is divided into 2 sets: according to the treatment;
data psc pst;
        set ps;
        if trt = 0 then output psc;
        if trt = 1 then output pst;
run;

proc sort data=psc;
by count;
run;

proc sort data=pst;
by count;
run;

%let seed=3;
*assign a random number to all t patients;
data pst_random;
   array isAssigned[1:&total_t] _temporary_(&total_t*0);
   set pst;
   do while(1);
   tid=ceil(&total_t*ranuni(&seed));
   if isAssigned[tid] then continue;
   isAssigned[tid]=1;
```

```
   leave;
   end;
run;



*create a table where all t patients are repeated for each c patient
 ↪  with their difference in ps-everything is ordered by
ps difference;
proc sql;
        create table matching_m as
        select c.count as csubid, c.ps as cps, t.count as tsubid,
         ↪  t.ps as tps, t.tid as tid, abs(tps-cps) as diff
        from psc c cross join pst_random t
        order by csubid, diff
        ;

*for each csubid I give a descending number (trank) to its tsubid
 ↪  match: trank=1 is given to the subid that is
closest in terms of ps to the csubid considered (smallest diff);
data ranking;
        set matching_m;
        by csubid diff;
        retain trank;
        if first.csubid then do;
                trank = 1;
        end;
        else trank = trank + 1;
run;
```

```
*reverse order, now everything is ordered by random tid number- I can
↪ have multiple 1s for the same tid;
proc sort data=ranking out=matching_con;
by tid;
run;

*Within each tid subgroup I assign a random number(cid) for all the c
↪ patients;
data matching_con;
array isAssigned[1:&total_c] _temporary_(&total_c*0);
set matching_con;
by tid;
do while(1);
cid=ceil(&total_c*ranuni(&seed));
if isAssigned[cid] then continue;
isAssigned[cid]=1;
if last.tid then do i=1 to dim(isAssigned);
isAssigned[i]=0;
end;
leave;
end;
drop i;
run;



*I order by tid closeness to c patients and I pick the first one- no
↪ ordered rows because I order by cid;
proc sort data=matching_con out=matched1;
        by tid trank cid;
```

```
*proc print data=matched1;
*run;

*for each tid I make the match picking at random one of those with
 ↪  trank=1;
*in the unmatched1 I have all those c patients that have trank=1 but
 ↪  were not taken because of their cid;
data matched1 unmatched1;
        set matched1;
        by tid trank cid;
        if first.tid and trank=1 then output matched1;
        else if trank = 1 then output unmatched1;
run;

proc print data=matched1;
proc print data=unmatched1;
run;

proc sort data=matching_con;
        by csubid;
run;

proc sort data=unmatched1;
        by csubid;
run;
proc sort data=matched1;
        by tsubid;

*I keep the c patients that weren't matched so that I can do another
 ↪  iteration of the same algorithm;
```

```
data matching2;
        merge matching_con unmatched1(keep=csubid in=inum);
        by csubid;
        if inum;
run;
proc print data=matching2;
run;

proc sort data=matching2;
        by tsubid;

*remove the t patients in the dataset that were matched;
data matching2;
        merge matching2 matched1 (in=inm1 keep=tsubid);
        by tsubid;
        if inm1 then delete;;
run;

*order by csubid and diff to restart the algorithm giving rank
 ↪  according to closeness;
proc sort data=matching2;
        by csubid diff;

proc print data=matching2;
run;

*second iteration of the previous algorithm;
data matching2;
        set matching2;
        by csubid diff;
```

```
        retain trank;
        if first.csubid then do;
                trank = 1;
        end;
        else trank = trank + 1;
run;

proc sort data=matching2 out=matched2;
        by tid trank cid;

data matched2 unmatched2;
        set matched2;
        by tid trank cid;
        if first.tid and trank=1 then output matched2;
        else if trank = 1 then output unmatched2;
run;

proc print data=matched2;
proc print data=unmatched2;
run;

proc sort data=unmatched2;
        by csubid;
run;
proc sort data=matched2;
        by tsubid;

data matching3;
        merge matching2 unmatched2(keep=csubid in=inum);
        by csubid;
```

```
        if inum;
run;

proc sort data=matching3;
        by tsubid;

data matching3;
        merge matching3 matched2(in=inm1 keep=tsubid);
        by tsubid;
        if inm1 then delete;;
run;

proc sort data=matching3;
        by csubid diff;
run;

/*--The same part of the code is iterated other 47 times--*/

*all pairs together;
data allmatch;
        retain pairno 0;
        set matched1 matched2 matched3 matched4 matched5 matched6
         ↪ matched7 matched8 matched9 matched10 matched11 matched12
         ↪ matched13
        matched14 matched15 matched16 matched17 matched18 matched19
         ↪ matched20 matched21 matched22 matched23 matched24
         ↪ matched25
        matched26 matched27 matched28 matched29 matched30 matched31
         ↪ matched32 matched33 matched34 matched35 matched36
         ↪ matched37 matched38
```

```
        matched39 matched40 matched41 matched42 matched43 matched44
        ↪ matched45 matched46 matched47 matched48 matched49;
        pairno = pairno+1;
        keep tsubid csubid pairno diff;
run;

*pairs statistics;
proc means data=allmatch min mean median max stddev;
var diff;
run;

proc sgplot data=allmatch;
histogram diff;
run;

*Pick up the matched pairs so that for each patient I have the number
↪ of the pair he belongs to;
proc sql;
        create table mbase as
        select b.*, m.pairno
        from survival b, allmatch m
        where count = m.tsubid or count = m.csubid
        ;

proc sql;
        create table after as
        select m.*, p.ps
        from mbase m, ps p
        where p.count = m.count
        ;
```

```
*PS distribution after greedy matching;

data histo;
set after;
if trt=0 then c=ps;
else t=ps;
run;



%let gpath="C:\Users\635185\Desktop\Poli\Tesi magistrale";
%let dpi=200;
ods html close;
ods listing image_dpi=&dpi;
/*--MirrorHistograms-Vertical--*/
proc template;
  define statgraph MirrorHistogramVert;
    dynamic _binwidth;
      begingraph;
      entrytitle 'Distribution of PS after greedy 1:1 matching';
      layout lattice / columndatarange=union rowgutter=0;
            columnaxes;
                  columnaxis / display=(tickvalues) griddisplay=on
                       linearopts=(tickvaluesequence=(start=0 end=1
                        ↪  increment=0.1) tickvaluepriority=true);
                endcolumnaxes;
          layout overlay / walldisplay=none
           ↪  xaxisopts=(griddisplay=on)
                                    xaxisopts=(griddisplay=on)
```

```
                        yaxisopts=(griddisplay=on
                         ↪ display=(tickvalues label)
                           linearopts=(tickvaluesequence=(start=10
                            ↪ end=60 increment=10)
                            ↪ tickvaluepriority=true));
            histogram c /  binstart=0.1 binwidth=_binwidth
             ↪ binaxis=false
                      fillattrs=graphdata1 datatransparency=0.3;
                        entry halign=right 'Control (N=100)' /
                         ↪ valign=top;
            endlayout;
          layout overlay / walldisplay=none
                        yaxisopts=(reverse=true griddisplay=on
                         ↪ display=(tickvalues label)
                        linearopts=(tickvaluesequence=(start=10
                         ↪ end=60 increment=10)
                         ↪ tickvaluepriority=true))
                        xaxisopts=(griddisplay=on);
              histogram t /  binstart=0.1 binwidth=_binwidth
               ↪ binaxis=false
                     fillattrs=graphdata2 datatransparency=0.3;
                       entry halign=right 'Treatment (N=100)' /
                        ↪ valign=bottom;
   endlayout;
          endlayout;
        endgraph;
  end;
run;
```

```
ods graphics / reset width=5in height=3in imagename='histo_logistic';
proc sgrender data= work.histo template=MirrorHistogramVert;
  dynamic _binwidth=0.08;
run;

*Kaplan - Meier after matching;
proc lifetest data=mbase atrisk plots=survival(failure cl);
          time event_time*censor(0);
          strata trt / test=logrank;
run;


*Cox model with 1:1 greedy matching with just trt;
proc phreg data=mbase;
          class trt(ref='Control');
          model event_time*censor(0) = trt;
          hazardratio trt /cl=wald diff=ref;
run;


*Cox model with 1:1 greedy matching with trt and covariates in PS
 ↪  (doubly robust);
proc phreg data=mbase;
          class sex trt(ref='Control');
          model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5;
          hazardratio trt /cl=wald diff=ref;
run;


*Cox model with 1:1 greedy matching with trt with pairs strata;
```

```
proc phreg data=mbase;
          class trt(ref='Control') pairno;
          model event_time*censor(0) = trt;
          strata pairno;
          hazardratio trt /cl=wald diff=ref;
run;

*Cox model with 1:1 greedy matching with trt and covariates in PS
 ↪  (doubly robust) with pairs strata;
proc phreg data=mbase;
          class sex trt(ref='Control') pairno;
          model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5;
          strata pairno;
          hazardratio trt /cl=wald diff=ref;
run;



**********************************
* GREEDY 1:1 MATCHING WITH CALIPER *
**********************************

/*formats*/
proc format;
        value   sexid 0 = 'F'
                      1 = 'M'
                      ;
        value   trtid 0 = 'Control'
                      1 = 'Treatment'
                      ;
run;
```

```
*****************************************************************;
*****************Import dataset(baseline+survival)**************;

PROC IMPORT OUT= WORK.survival
            DATAFILE= "/folders/myfolders/tesi/survival.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;

data survival;
set survival;
format sex sexid. trt trtid.;
drop var1;
run;



*****************************************************************;
*****************************************PS**********************;
proc logistic data=survival;
        class sex trt;
        model trt(event='Treatment') = sex age v1 v2 v3 v4 v5;
        output out = ps pred = ps xbeta = logit_ps;
run;
*caliper;
%let caliper=0.1;
%let total_t=100;
%let total_c=300;
```

```
%let total=%sysevalf(&total_t+&total_c);



*the dataset is divided into 2 sets: according to the treatment;
data psc pst;
        set ps;
        if trt = 0 then output psc;
        if trt = 1 then output pst;
run;

proc sort data=psc;
by count;
run;

proc sort data=pst;
by count;
run;

%let seed=3;
*assign a random number to all k patients;
data pst_random;
   array isAssigned[1:&total_t] _temporary_(&total_t*0);
   set pst;
   do while(1);
   tid=ceil(&total_t*ranuni(&seed));
   if isAssigned[tid] then continue;
   isAssigned[tid]=1;
   leave;
   end;
run;
```

```
*create a table where all k patients are repeated for each p patient
 ↪  with their difference in ps-everything is ordered by
ps difference;
proc sql;
        create table matching_m as
        select c.count as csubid, c.ps as cps, t.count as tsubid,
         ↪  t.ps as tps, t.tid as tid, abs(tps-cps) as diff
        from psc c cross join pst_random t
        order by csubid, diff
        ;

title 'Distribution of the difference of propensity score';

*ods pdf file="C:/Users/635185/Desktop/Poli/Tesi
 ↪  magistrale/output/histo_diff_cal.pdf";

proc sgplot data=matching_m;
  histogram diff;
  refline &caliper / axis = x label lineattrs=(color=CXFF0000
   ↪  thickness= 0.08cm);
run;
title;

*ods pdf close;

*remove pairs with a difference in PS higher than the caliper;
data matching_m;
```

```
set matching_m;
where diff<= &caliper;
run;

proc freq data=matching_m;
tables csubid;
run;

*for each csubid I give a descending number (trank) to its tsubid
 ↪  match: trank=1 is given to the subid that is
closest in terms of ps to the csubid considered (smallest diff);
data ranking;
        set matching_m;
        by csubid diff;
        retain trank;
        if first.csubid then do;
                trank = 1;
        end;
        else trank = trank + 1;
run;

*reverse order, now everything is ordered by random tid number- I can
 ↪  have multiple 1s for the same tid;
proc sort data=ranking out=matching_con;
by tid;
run;

*Within each tid subgroup I assign a random number(cid) for all the c
 ↪  patients;
data matching_con;
```

```
array isAssigned[1:&total_c] _temporary_(&total_c*0);
set matching_con;
by tid;
do while(1);
cid=ceil(&total_c*ranuni(&seed));
if isAssigned[cid] then continue;
isAssigned[cid]=1;
if last.tid then do i=1 to dim(isAssigned);
isAssigned[i]=0;
end;
leave;
end;
drop i;
run;



*I order by tid closeness to c patients and I pick the first one- no
 ↪  ordered rows because I order by cid;
proc sort data=matching_con out=matched1;
        by tid trank cid;

*proc print data=matched1;
*run;

*for each tid I make the match picking at random one of those with
 ↪  trank=1;
*in the unmatched1 I have all those c patients that have trank=1 but
 ↪  were not taken because of their cid;
data matched1 unmatched1;
        set matched1;
```

```
        by tid trank cid;
        if first.tid and trank=1 then output matched1;
        else if trank = 1 then output unmatched1;
run;

proc print data=matched1;
proc print data=unmatched1;
run;

proc sort data=matching_con;
        by csubid;
run;

proc sort data=unmatched1;
        by csubid;
run;
proc sort data=matched1;
        by tsubid;

*I keep the c patients that weren't matched so that I can do another
 ↪  iteration of the same algorithm;
data matching2;
        merge matching_con unmatched1(keep=csubid in=inum);
        by csubid;
        if inum;
run;
proc print data=matching2;
run;

proc sort data=matching2;
```

```
        by tsubid;

*remove the t patients in the dataset that were matched;
data matching2;
        merge matching2 matched1 (in=inm1 keep=tsubid);
        by tsubid;
        if inm1 then delete;;
run;

*order by csubid and diff to restart the algorithm giving rank
 ↪  according to closeness;
proc sort data=matching2;
        by csubid diff;

proc print data=matching2;
run;

*second iteration of the previous algorithm;
data matching2;
        set matching2;
        by csubid diff;
        retain trank;
        if first.csubid then do;
                trank = 1;
        end;
        else trank = trank + 1;
run;

proc sort data=matching2 out=matched2;
        by tid trank cid;
```

```
data matched2 unmatched2;
        set matched2;
        by tid trank cid;
        if first.tid and trank=1 then output matched2;
        else if trank = 1 then output unmatched2;
run;

proc print data=matched2;
proc print data=unmatched2;
run;

proc sort data=unmatched2;
        by csubid;
run;
proc sort data=matched2;
        by tsubid;

data matching3;
        merge matching2 unmatched2(keep=csubid in=inum);
        by csubid;
        if inum;
run;

proc sort data=matching3;
        by tsubid;

data matching3;
        merge matching3 matched2(in=inm1 keep=tsubid);
        by tsubid;
```

```
        if inm1 then delete;;
run;

proc sort data=matching3;
        by csubid diff;

*third iteration of the same algorithm;
data matching3;
        set matching3;
        by csubid diff;
        retain trank;
        if first.csubid then do;
                trank = 1;
        end;
        else trank = trank + 1;
run;

proc sort data=matching3 out=matched3;
        by tid trank cid;

data matched3 unmatched3;
        set matched3;
        by tid trank cid;
        if first.tid and trank=1 then output matched3;
        else if trank = 1 then output unmatched3;
run;

proc print data=matched3;
proc print data=unmatched3;
run;
```

```
proc sort data=unmatched3;
        by csubid;
run;
proc sort data=matched3;
        by tsubid;

data matching4;
        merge matching3 unmatched3(keep=csubid in=inum);
        by csubid;
        if inum;
run;

proc sort data=matching4;
        by tsubid;

data matching4;
        merge matching4 matched3(in=inm1 keep=tsubid);
        by tsubid;
        if inm1 then delete;;
run;

proc sort data=matching4;
        by csubid diff;

*all pairs together;
data allmatch;
        retain pairno 0;
        set matched1 matched2 matched3;
        pairno = pairno+1;
```

```
        keep tsubid csubid pairno diff;
run;

proc means data=allmatch min mean median max stddev;
var diff;
run;

proc sgplot data=allmatch;
histogram diff;
run;


*Pick up the matched pairs so that for each patient I have the number
↪  of the pair he belongs to;
proc sql;
        create table mbase as
        select b.*, m.pairno
        from survival b, allmatch m
        where count = m.tsubid or count = m.csubid
        ;

proc print data=mbase;
run;

proc sort data=mbase;
by count;
run;

proc sql;
        create table after as
```

```
        select m.*, p.ps
        from mbase m, ps p
        where p.count = m.count
        ;



proc sql;
select count(*)
from after
where trt=0;
run;
quit;

*PS distribution after greedy matching with caliper;

data histo;
set after;
if trt=0 then c=ps;
else t=ps;
run;



%let gpath="C:\Users\635185\Desktop\Poli\Tesi magistrale";
%let dpi=200;
ods html close;
ods listing image_dpi=&dpi;
/*--MirrorHistograms-Vertical--*/
proc template;
```

```
define statgraph MirrorHistogramVert;
  dynamic _binwidth;
    begingraph;
    entrytitle 'Distribution of PS after greedy 1:1 matching with
     ↪ caliper';
    layout lattice / columndatarange=union rowgutter=0;
          columnaxes;
                columnaxis / display=(tickvalues) griddisplay=on
                     linearopts=(tickvaluesequence=(start=0 end=1
                      ↪ increment=0.1) tickvaluepriority=true);
               endcolumnaxes;
        layout overlay / walldisplay=none
         ↪ xaxisopts=(griddisplay=on)
                                  xaxisopts=(griddisplay=on)
                        yaxisopts=(griddisplay=on
                         ↪ display=(tickvalues label)
                           linearopts=(tickvaluesequence=(start=10
                            ↪ end=60 increment=10)
                            ↪ tickvaluepriority=true));
            histogram c /  binstart=0.1 binwidth=_binwidth
             ↪ binaxis=false
                      fillattrs=graphdata1 datatransparency=0.3;
                        entry halign=right 'Control (N=54)' /
                         ↪ valign=top;
            endlayout;
        layout overlay / walldisplay=none
                        yaxisopts=(reverse=true griddisplay=on
                         ↪ display=(tickvalues label)
```

```
                              linearopts=(tickvaluesequence=(start=10
                               ↪  end=60 increment=10)
                               ↪  tickvaluepriority=true))
                              xaxisopts=(griddisplay=on);
                    histogram t /  binstart=0.1 binwidth=_binwidth
                     ↪  binaxis=false
                          fillattrs=graphdata2 datatransparency=0.3;
                            entry halign=right 'Treatment (N=54)' /
                             ↪  valign=bottom;
   endlayout;
          endlayout;
        endgraph;
  end;
run;


ods graphics / reset width=5in height=3in imagename='histo_logistic';
proc sgrender data= work.histo template=MirrorHistogramVert;
  dynamic _binwidth=0.08;
run;

*Kaplan - Meier;
proc lifetest data=mbase atrisk plots=survival(failure cl);
          time event_time*censor(0);
          strata trt / test=logrank;
run;


*Cox model with 1:1 greedy matching with caliper with just trt;
proc phreg data=mbase;
```

```
          class trt(ref='Control');
          model event_time*censor(0) = trt;
          hazardratio trt /cl=wald diff=ref;
run;


*Cox model with 1:1 greedy matching with caliper with trt and
↪  covariates in PS (doubly robust);
proc phreg data=mbase;
          class trt(ref='Control') sex;
          model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5;
          hazardratio trt/cl=wald diff=ref;
run;

*Cox model with 1:1 greedy matching with caliper with trt and pairs
↪  strata;
proc phreg data=mbase;
          class trt(ref='Control') pairno;
          model event_time*censor(0) = trt;
          strata pairno;
          hazardratio trt/cl=wald diff=ref;
run;

*Cox model with 1:1 greedy matching with caliper with trt and
↪  covariates in PS (doubly robust) with pairs strata;
proc phreg data=mbase;
          class trt(ref='Control') sex pairno;
          model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5;
          strata pairno;
          hazardratio trt/cl=wald diff=ref;
```

```
run;



**********************************
* OPTIMAL 1:1  AND 1:1/2 MATCHING *
**********************************
/*formats*/
proc format;
        value   sexid 0 = 'F'
                      1 = 'M'
                      ;
        value   trtid 0 = 'Control'
                      1 = 'Treatment'
                      ;
run;



*****************************************************************;
*****************Import dataset(baseline+survival)**************;

PROC IMPORT OUT= WORK.survival
            DATAFILE= "/folders/myfolders/tesi/survival.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;

data survival;
set survival;
format sex sexid. trt trtid.;
```

```
drop var1;
run;

%let total_t=100;
%let total_c=300;
%let total=%sysevalf(&total_t+&total_c);



proc logistic data=survival;
        class sex trt;
        model trt(event='Treatment') = sex age v1 v2 v3 v4 v5;
        output out = ps pred = ps xbeta = logit_ps;
run;


**********optimal matching 1:1**********;

PROC IMPORT OUT= WORK.opt_pairs
            DATAFILE= "/folders/myfolders/tesi/opt_pairs.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;


*import optimal pairs;
data opt_pairs;
set opt_pairs;
drop var1;
```

```
format sex sexid. trt trtid.;
run;

data ps;
        merge ps opt_pairs (in=inm1 keep=count matches);
        by count;
        if inm1;
run;

data psc pst;
        set ps;
        if trt = 0 then output psc;
        if trt = 1 then output pst;
run;

*pairs statistics;
proc sql;
        create table difference as
        select c.count as csubid, c.ps as cps, t.count as tsubid,
        ↪  t.ps as tps, abs(tps-cps) as diff
        from psc c, pst t
        where c.matches=t.matches
        ;
run;
quit;

proc means data=difference min mean median max stddev;
var diff;
run;
```

```
proc sgplot data=difference;
histogram diff;
run;



*PS distribution after matching;


data histo;
set ps;
if trt=0 then c=ps;
else t=ps;
run;




%let gpath="C:\Users\635185\Desktop\Poli\Tesi magistrale";
%let dpi=200;
ods html close;
ods listing image_dpi=&dpi;
/*--MirrorHistograms-Vertical--*/
proc template;
  define statgraph MirrorHistogramVert;
    dynamic _binwidth;
      begingraph;
      entrytitle 'Distribution of PS after optimal 1:1 matching';
      layout lattice / columndatarange=union rowgutter=0;
            columnaxes;
                  columnaxis / display=(tickvalues) griddisplay=on
                        linearopts=(tickvaluesequence=(start=0 end=1
                        ↪  increment=0.1) tickvaluepriority=true);
```

```
            endcolumnaxes;
        layout overlay / walldisplay=none
         ↪ xaxisopts=(griddisplay=on)
                                xaxisopts=(griddisplay=on)
                        yaxisopts=(griddisplay=on
                         ↪ display=(tickvalues label)
                           linearopts=(tickvaluesequence=(start=10
                            ↪ end=80 increment=10)
                            ↪ tickvaluepriority=true));
            histogram c /  binstart=0.1 binwidth=_binwidth
             ↪ binaxis=false
                      fillattrs=graphdata1 datatransparency=0.3;
                        entry halign=right 'Control (N=100)' /
                         ↪ valign=top;
            endlayout;
        layout overlay / walldisplay=none
                        yaxisopts=(reverse=true griddisplay=on
                         ↪ display=(tickvalues label)
                        linearopts=(tickvaluesequence=(start=10
                         ↪ end=80 increment=10)
                         ↪ tickvaluepriority=true))
                        xaxisopts=(griddisplay=on);
              histogram t /  binstart=0.1 binwidth=_binwidth
               ↪ binaxis=false
                     fillattrs=graphdata2 datatransparency=0.3;
                       entry halign=right 'Treatment (N=100)' /
                        ↪ valign=bottom;
endlayout;
        endlayout;
      endgraph;
```

```
  end;
run;


ods graphics / reset width=5in height=3in imagename='histo_logistic';
proc sgrender data= work.histo template=MirrorHistogramVert;
  dynamic _binwidth=0.08;
run;


*Kaplan - Meier without any ajustments;
proc lifetest data=opt_pairs atrisk plots=survival(failure cl);
          time event_time*censor(0);
          strata trt / test=logrank;
run;


*Cox model with 1:1 optimal matching with just trt;
proc phreg data=opt_pairs;
          class trt(ref='Control');
          model event_time*censor(0) = trt;
          hazardratio trt /cl=wald diff=ref;
run;


*Cox model with 1:1 optimal matching with trt and covariates in PS
↪  (doubly robust);
proc phreg data=opt_pairs;
          class trt(ref='Control') sex;
          model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5;
```

```
          hazardratio trt /cl=wald diff=ref;
run;

*Pair stratified Cox model with 1:1 optimal matching with trt;
proc phreg data=opt_pairs;
          class trt(ref='Control') matches;
          model event_time*censor(0) = trt;
          strata matches;
          hazardratio trt /cl=wald diff=ref;
run;

*Pair tratified Cox model with 1:1 optimal matching with trt and
 ↪  covariates in PS (doubly robust);
proc phreg data=opt_pairs;
          class trt(ref='Control') sex matches;
          model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5;
          strata matches;
          hazardratio trt /cl=wald diff=ref;
run;


**********optimal matching 1:1/2 **********;

PROC IMPORT OUT= WORK.opt_pairs
            DATAFILE= "/folders/myfolders/tesi/opt_pairs_2.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;
```

```
data opt_pairs;
set opt_pairs;
drop x;
format sex sexid. trt trtid.;
run;


data ps;
        merge ps opt_pairs (in=inm1 keep=count matches);
        by count;
        if inm1;
run;


data psc pst;
        set ps;
        if trt = 0 then output psc;
        if trt = 1 then output pst;
run;

*pairs statistics;
proc sql;
        create table difference as
        select c.matches as pair, c.count as csubid, c.ps as cps,
        ↪  t.count as tsubid, t.ps as tps, abs(tps-cps) as diff
        from psc c, pst t
        where c.matches=t.matches
        order by c.matches
        ;
```

```
run;
quit;

proc means data=difference min mean median max stddev;
var diff;
run;

proc sgplot data=difference;
histogram diff;
run;



*PS distribution after matching;

data histo;
set ps;
if trt=0 then c=ps;
else t=ps;
run;




%let gpath="C:\Users\635185\Desktop\Poli\Tesi magistrale";
%let dpi=200;
ods html close;
ods listing image_dpi=&dpi;
/*--MirrorHistograms-Vertical--*/
proc template;
  define statgraph MirrorHistogramVert;
    dynamic _binwidth;
```

```
begingraph;
entrytitle 'Distribution of PS after optimal 1:1/2 matching';
layout lattice / columndatarange=union rowgutter=0;
      columnaxes;
            columnaxis / display=(tickvalues) griddisplay=on
                 linearopts=(tickvaluesequence=(start=0 end=1
                  ↪ increment=0.1) tickvaluepriority=true);
          endcolumnaxes;
    layout overlay / walldisplay=none
     ↪ xaxisopts=(griddisplay=on)
                              xaxisopts=(griddisplay=on)
                    yaxisopts=(griddisplay=on
                     ↪ display=(tickvalues label)
                       linearopts=(tickvaluesequence=(start=10
                        ↪ end=70 increment=10)
                        ↪ tickvaluepriority=true));
        histogram c /  binstart=0.1 binwidth=_binwidth
         ↪ binaxis=false
                  fillattrs=graphdata1 datatransparency=0.3;
                    entry halign=right 'Control (N=119)' /
                     ↪ valign=top;
        endlayout;
    layout overlay / walldisplay=none
                    yaxisopts=(reverse=true griddisplay=on
                     ↪ display=(tickvalues label)
                    linearopts=(tickvaluesequence=(start=10
                     ↪ end=70 increment=10)
                     ↪ tickvaluepriority=true))
                    xaxisopts=(griddisplay=on);
```

```
                histogram t /  binstart=0.1 binwidth=_binwidth
                 ↪  binaxis=false
                      fillattrs=graphdata2 datatransparency=0.3;
                        entry halign=right 'Treatment (N=100)' /
                         ↪  valign=bottom;
   endlayout;
          endlayout;
        endgraph;
  end;
run;


ods graphics / reset width=5in height=3in imagename='histo_logistic';
proc sgrender data= work.histo template=MirrorHistogramVert;
  dynamic _binwidth=0.08;
run;



*Kaplan - Meier without any ajustments;
proc lifetest data=opt_pairs atrisk plots=survival(failure cl);
          time event_time*censor(0);
          strata trt / test=logrank;
run;


*Cox model with 1:1/2 optimal matching with caliper with just trt;
proc phreg data=opt_pairs;
          class trt(ref='Control');
          model event_time*censor(0) = trt;
```

```
          hazardratio trt /cl=wald diff=ref;
run;


*Cox model with 1:1/2 optimal matching with caliper with trt and
 ↪  covariates in PS (doubly robust);
proc phreg data=opt_pairs;
          class trt(ref='Control') sex;
          model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5;
          hazardratio trt /cl=wald diff=ref;
run;

*Pair stratified Cox model with 1:1/2 optimal matching with trt and
 ↪  covariates in PS (doubly robust;
proc phreg data=opt_pairs;
          class trt(ref='Control') sex matches;
          model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5;
          strata matches;
          hazardratio trt /cl=wald diff=ref;
run;

*Pair stratified Cox model with 1:1/2 optimal matching with caliper
 ↪  with trt;
proc phreg data=opt_pairs;
          class trt(ref='Control') matches;
          model event_time*censor(0) = trt;
          strata matches;
          hazardratio trt /cl=wald diff=ref;
run;
```

```
********
* IPTW *
********
/*formats*/
proc format;
        value   sexid 0 = 'F'
                      1 = 'M'
                      ;
        value   trtid 0 = 'Control'
                      1 = 'Treatment'
                      ;
run;

*****************************************************************;
*****************Import dataset(baseline+survival)**************;

PROC IMPORT OUT= WORK.survival
            DATAFILE= "/folders/myfolders/tesi/survival.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;

data survival;
set survival;
format sex sexid. trt trtid.;
drop var1;
run;
```

```
************************************************************;



****************************************PS*************************;
proc logistic data=survival;
        class sex trt;
        model trt(event='Treatment') = sex age v1 v2 v3 v4 v5;
        output out = ps pred = ps xbeta = logit_ps;
run;

*IPTW method;
*general weights;
data iptw_db;
set ps;
if trt=1 then iptw=1/ps;
else iptw=1/(1-ps);
run;

*size of the pseudo population;
proc sql;
select sum(iptw) as Pseudo
from iptw_db;
quit;

*stabilized weights;
proc sql;
select count(*) into :total
from ps;
quit;
```

```
%put &total;
%let t_patients=;
proc sql;
select count(*) into :t_patients
from ps
where trt=1;
quit;
%put &t_patients;

data iptw_db;
set iptw_db;
if trt=1 then iptws=iptw*(&t_patients/&total);
else iptws=iptw*(1-&t_patients/&total);
run;

*size of the pseudo population with stabilized weights;
proc sql;
select sum(iptws) as Pseudo_s
from iptw_db;
quit;

*number of treated patients in the pseudo-population;
proc sql;
select sum(iptw)
from iptw_db
where trt=1;
quit;
run;

*PS distribution with the different weights;
```

```
data frequencies;
set iptw_db;
f=iptws*100;
run;

data histo;
set iptw_db;
if trt=1 then t=ps;
else c=ps;
run;



%let gpath="/export/home/ma898685/Report";
%let dpi=200;
ods html close;
ods listing gpath=&gpath image_dpi=&dpi;
/*--MirrorHistograms-Vertical--*/
proc template;
  define statgraph MirrorHistogramVert;
    dynamic _binwidth;
      begingraph;
      entrytitle 'Distribution of PS';
      layout lattice / columndatarange=union rowgutter=0;
            columnaxes;
                  columnaxis / display=(tickvalues) griddisplay=on
                        linearopts=(tickvaluesequence=(start=0 end=1
                        ↪  increment=0.1) tickvaluepriority=true);
                endcolumnaxes;
          layout overlay / walldisplay=none
          ↪  xaxisopts=(griddisplay=on)
```

```
                                    xaxisopts=(griddisplay=on)
                            yaxisopts=(griddisplay=on
                             ↪  display=(tickvalues label)
                               linearopts=(tickvaluesequence=(start=10
                                ↪  end=70 increment=10)
                                ↪  tickvaluepriority=true));
                    histogram c / freq=iptw  binstart=0.1
                     ↪  binwidth=_binwidth binaxis=false
                              fillattrs=graphdata1 datatransparency=0.3;
                                entry halign=right 'Control (N=387)' /
                                 ↪  valign=top;
                    endlayout;
                layout overlay / walldisplay=none
                                 yaxisopts=(reverse=true griddisplay=on
                                  ↪  display=(tickvalues label)
                                 linearopts=(tickvaluesequence=(start=10
                                  ↪  end=70 increment=10)
                                  ↪  tickvaluepriority=true))
                                 xaxisopts=(griddisplay=on);
                      histogram t / freq=iptw binstart=0.1
                       ↪  binwidth=_binwidth binaxis=false
                            fillattrs=graphdata2 datatransparency=0.3;
                              entry halign=right 'Treatment (N=462)' /
                               ↪  valign=bottom;
    endlayout;
            endlayout;
          endgraph;
  end;
run;
```

```
ods graphics / reset width=5in height=3in imagename='histo_iptws';
proc sgrender data= work.histo template=MirrorHistogramVert;
  dynamic _binwidth=0.08;
run;

*****************************************iptws************************;
data histo;
set frequencies;
if trt=1 then t=ps;
else c=ps;
run;


%let gpath="/export/home/ma898685/Report";
%let dpi=200;
ods html close;
ods listing gpath=&gpath image_dpi=&dpi;
/*--MirrorHistograms-Vertical--*/
proc template;
  define statgraph MirrorHistogramVert;
    dynamic _binwidth;
      begingraph;
      entrytitle 'Distribution of PS';
      layout lattice / columndatarange=union rowgutter=0;
            columnaxes;
                  columnaxis / display=(tickvalues) griddisplay=on
                       linearopts=(tickvaluesequence=(start=0 end=1
                        ↪  increment=0.1) tickvaluepriority=true);
                endcolumnaxes;
```

```
        layout overlay / walldisplay=none
        ↪ xaxisopts=(griddisplay=on)
                                  xaxisopts=(griddisplay=on)
                          yaxisopts=(griddisplay=on
                          ↪ display=(tickvalues label)
                            linearopts=(tickvaluesequence=(start=10
                            ↪ end=70 increment=10)
                            ↪ tickvaluepriority=true));
            histogram c / freq=f  binstart=0.1 binwidth=_binwidth
            ↪ binaxis=false
                        fillattrs=graphdata1 datatransparency=0.3;
                          entry halign=right 'Control (N=290)' /
                          ↪ valign=top;
            endlayout;
        layout overlay / walldisplay=none
                          yaxisopts=(reverse=true griddisplay=on
                          ↪ display=(tickvalues label)
                          linearopts=(tickvaluesequence=(start=10
                          ↪ end=70 increment=10)
                          ↪ tickvaluepriority=true))
                          xaxisopts=(griddisplay=on);
              histogram t / freq=f binstart=0.1 binwidth=_binwidth
              ↪ binaxis=false
                    fillattrs=graphdata2 datatransparency=0.3;
                      entry halign=right 'Treatment (N=116)' /
                      ↪ valign=bottom;
 endlayout;
        endlayout;
      endgraph;
end;
```

```
run;


ods graphics / reset width=5in height=3in imagename='histo_iptws';
proc sgrender data= work.histo template=MirrorHistogramVert;
  dynamic _binwidth=0.08;
run;

**********IPTW;
*Kaplan - Meier;
ods graphics on;
ods pdf file="/folders/myfolders/tesi/iptw_KM.pdf";
proc lifetest data=iptw_db atrisk plots=survival(failure cl);
          strata trt;
          weight iptw;
          time event_time*censor(0);
run;
ods pdf close;
ods graphics off;

*Cox model with general IPTW and trt;
   proc phreg data=iptw_db;
          class trt(ref='Control');
          weight iptw;
          model event_time*censor(0) = trt;
          hazardratio trt /cl=wald diff=ref;

  run;

*Cox model with general IPTW with doubly robust approach;
```

```
  proc phreg data=iptw_db; *plots(overlay)=survival;
          class trt(ref='Control') sex;
          weight iptw;
          model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5;
          hazardratio trt /cl=wald diff=ref;
  run;

****************************************stabilized weights;
*Kaplan - Meier;
ods graphics on;
ods pdf file="/folders/myfolders/tesi/iptws_KM.pdf";
proc lifetest data=iptw_db atrisk plots=survival(failure cl);
          strata trt;
          weight iptws;
          time event_time*censor(0);
run;
ods pdf close;
ods graphics off;

*Cox model with stabilized IPTW and trt;
   proc phreg data=iptw_db plots=survival;
          class trt(ref='Control');
          weight iptws;
          model event_time*censor(0) = trt;
          hazardratio trt /cl=wald diff=ref;

  run;

*Cox model with stabilized IPTW with doubly robust approach;
  proc phreg data=iptw_db; *plots(overlay)=survival;
```

```
          class trt(ref='Control') sex;
          weight iptws;
          model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5 ;
          hazardratio trt /cl=wald diff=ref;
  run;

******************
* STRATIFICATION *
******************
/*formats*/
proc format;
        value   sexid 0 = 'F'
                      1 = 'M'
                      ;
        value   trtid 0 = 'Control'
                      1 = 'Treatment'
                      ;
        value   ttl 1 = 'First Tertile'
                    2 = 'Second Tertile'
                    3 = 'Third Tertile'
                    ;
run;

*****************************************************************;
*****************Import dataset(baseline+survival)**************;

PROC IMPORT OUT= WORK.survival
            DATAFILE= "/folders/myfolders/tesi/survival.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
```

```
     DATAROW=2;
RUN;

data base;
set survival;
format sex sexid. trt trtid.;
drop var1;
run;

%let total=400;
*************************************************************;

proc logistic data=base;
        class sex trt;
        model trt(event='Treatment') = sex age v1 v2 v3 v4 v5;
        output out = ps pred = ps xbeta = logit_ps;
run;




*************************************************************************choose
 ↪  tertiles;
proc sort data=ps;
by ps;
run;

*strata with equal sizes;
data tert;
        set ps;
        ttl = 1 + (_n_/&total > 1/3) + (_n_/&total > 2/3);
```

```
        format ttl ttl.;
        label ttl = 'tertiles';
run;



*propensity score divided in 3 quantiles ----> 3 groups with
 ↪  different size;
*select min and max of ps;
proc sql;
select min(ps) into :min_ps
from ps;
quit;
%put &min_ps;

proc sql;
select max(ps) into :max_ps
from ps;
quit;
%put &max_ps;

data tert;
        set tert;
        ttl_bis = 1 + (ps > &min_ps+(&max_ps-&min_ps)/3) + (ps >
         ↪  &min_ps+2*(&max_ps-&min_ps)/3);
        format ttl_bis ttl.;
        label ttl_bis = 'tertiles second type'
              ;
run;

proc print data=tert;
```

```
run;


data t1 t2 t3;
set tert;
if ttl_bis=1 then output t1;
else if ttl_bis=2 then output t2;
else output t3;
run;


*statistics per each tertile;
proc print data=t3;
run;


proc sql;
select count(*)
from t3
where trt=0;
run;


data histo;
set t3;
if trt=0 then c=ps;
else t=ps;
run;


%let gpath="C:\Users\635185\Desktop\Poli\Tesi magistrale";
%let dpi=200;
ods html close;
ods listing image_dpi=&dpi;
/*--MirrorHistograms-Vertical--*/
```

```
proc template;
  define statgraph MirrorHistogramVert;
    dynamic _binwidth;
      begingraph;
      entrytitle 'Distribution of PS - 3rd tertile';
      layout lattice / columndatarange=union rowgutter=0;
            columnaxes;
                  columnaxis / display=(tickvalues) griddisplay=on
                       linearopts=(tickvaluesequence=(start=0.66
                        ↪ end=0.98 increment=0.08)
                        ↪ tickvaluepriority=true);
                endcolumnaxes;
          layout overlay / walldisplay=none
           ↪ xaxisopts=(griddisplay=on)
                                    xaxisopts=(griddisplay=on)
                         yaxisopts=(griddisplay=on
                          ↪ display=(tickvalues label)
                            linearopts=(tickvaluesequence=(start=10
                             ↪ end=60 increment=10)
                             ↪ tickvaluepriority=true));
              histogram c /  binstart=0.66 binwidth=_binwidth
               ↪ binaxis=false
                        fillattrs=graphdata1 datatransparency=0.3;
                          entry halign=right 'Control (N=3)' /
                           ↪ valign=top;
              endlayout;
          layout overlay / walldisplay=none
                         yaxisopts=(reverse=true griddisplay=on
                          ↪ display=(tickvalues label)
```

```
                              linearopts=(tickvaluesequence=(start=10
                               ↪  end=60 increment=10)
                               ↪  tickvaluepriority=true))
                              xaxisopts=(griddisplay=on);
                    histogram t /  binstart=0.66 binwidth=_binwidth
                     ↪  binaxis=false
                          fillattrs=graphdata2 datatransparency=0.3;
                            entry halign=right 'Treatment (N=49)' /
                             ↪  valign=bottom;
   endlayout;
           endlayout;
         endgraph;
  end;
run;


ods graphics / reset width=5in height=3in imagename='histo_logistic';
proc sgrender data= work.histo template=MirrorHistogramVert;
  dynamic _binwidth=0.08;
run;

proc sort data=tert;
by count;
run;


ODS TRACE ON;

*Kaplan - Meier without any ajustments per each tertile;
proc lifetest data=t3 atrisk plots=survival(failure cl);
```

```
          time event_time*censor(0);
          strata trt/ test=logrank;
run;

ODS TRACE OFF;

*Stratified Cox model by quantiles;
proc phreg data=tert;
        class trt(ref='Control') ttl_bis;
        model event_time*censor(0) = trt;
        strata ttl_bis;
        hazardratio trt/cl=wald diff=ref;
run;
ods pdf close;



*Stratified Cox model by quantiles with doubly robust approach;
proc phreg data=tert;
        class trt(ref='Control')  ttl_bis sex;
        model event_time*censor(0) = trt sex age v1 v2 v3 v4 v5;
        strata ttl_bis;
        hazardratio trt/cl=wald diff=ref;
run;

****************
* FOREST PLOTS *
****************
* without doubly robust approach;

data forest;
```

```
  input Id Subgroup $3-27  HR  Low  High P$;
  zero=0; one=1;
  HR_lbl='HR';
  low_lbl='Low';
  high_lbl='High';
  p_lbl='P-value';
  ObsId=_n_;
  if count ne . then CountPct=put(count, 4.0) || "(" || put(percent,
  ↪  3.0) || ")";
  datalines;
1 Naive.....................1.107  0.874  1.403  0.3985
1 Covariates Adjustment....1.857  1.363  2.530  <0.0001
1 Matching..................      .      .      .
2 ..Greedy 1:1.............0.952  0.716  1.268  0.7385
2 ..Greedy 1:1 with caliper1.629  1.100  2.412  0.0148
2 ..Optimal 1:1............1.217  0.912  1.624  0.1826
2 ..Optimal 1:1/2..........1.274  0.965  1.684  0.0880
1 IPTW......................      .      .      .
2 ..General weights........1.996  1.733  2.300  <0.0001
2 ..Stabilized wights......2.094  1.674  2.620  <0.0001
1 Stratification............      .      .      .
2 ..Tertiles...............1.798  1.312  2.466  0.0003
;
run;
ods listing;

/*--Replace '.' in subgroup with blank--*/
data forest2;
  set forest;
  subgroup=translate(subgroup, ' ', '.');
```

```
  val=mod(_N_-1, 6);
  if val eq 1 or val eq 2 or val eq 3 then ref=obsid;

  /*--Separate Subgroup headers and obs into separate columns--*/
  if id=1 then do;
     heading=subgroup;
         subgroup='';
  end;
  run;

/*--Create font with smaller fonts for axis label, value and data--*/
proc template;
   define style listingSF;
      parent = Styles.Listing;
      style GraphFonts from GraphFonts
         "Fonts used in graph styles" /
         'GraphDataFont' = ("<sans-serif>, <MTsans-serif>",7pt)
         'GraphValueFont' = ("<sans-serif>, <MTsans-serif>",7pt)
         'GraphLabelFont' = ("<sans-serif>, <MTsans-serif>",7pt,
 ↪  bold);
;
   end;
run;

%let dpi=150;
ods listing style=listingSF gpath=&gpath image_dpi=&dpi;

/*--Define templage for Forest Plot--*/
/*--Template uses a Layout Lattice of 5 columns--*/
proc template;
```

```
 define statgraph Forest;
 dynamic  _bandcolor _headercolor _subgroupcolor;
   begingraph;
     layout lattice / columns=3 columnweights=(0.23 0.4 0.3);

       /*--Column headers--*/
       sidebar / align=top;
             layout lattice / rows=2 columns=3 columnweights=(0.18
↪  0.35 0.3)
          backgroundcolor=_headercolor opaque=true;
          entry textattrs=(size=8 weight=bold) halign=left
↪  "Methods";
          entry textattrs=(size=8 weight=bold) "Hazard Ratio with
↪  95% CI";
          entry halign=center textattrs=(size=8 weight=bold)
↪  "Statistics" ;
                  entry " ";
          entry " ";
                      entry " ";
          entry halign=center textattrs=(size=6) "Confidence
↪  interval";
                endlayout;
       endsidebar;

       /*--First Subgroup column, shows only the Y2 axis
↪  --*/
       /*--Use HighLow plot to place the heading and subgroup values
↪  as HighLabels--*/
       /*--Indenting is done by making the 2nd highlow bar 1 unit
↪  long            --*/
```

```
        /*--Highlow bar itself has thickness=0
↪  --*/
        layout overlay / walldisplay=none
              xaxisopts=(display=none linearopts=(viewmin=0
↪  viewmax=20))
              yaxisopts=(reverse=true display=none
↪  tickvalueattrs=(weight=bold));
          referenceline y=ref / lineattrs=(thickness=15
↪  color=_bandcolor);
          highlowplot y=obsid low=zero high=zero / highlabel=heading
↪  lineattrs=(thickness=0)
              labelattrs=(size=7 weight=bold);
          highlowplot y=obsid low=zero high=one / highlabel=subgroup
↪  lineattrs=(thickness=0);
        endlayout;


        /*--Second column showing Count and percent--
        layout overlay / xaxisopts=(display=none)
              yaxisopts=(reverse=true display=none)
↪  walldisplay=none;
                 referenceline y=ref / lineattrs=(thickness=15
↪  color=_bandcolor);
          scatterplot y=obsid x=zero / markercharacter=countpct
              markercharacterattrs=graphvaluetext;
               endlayout;                              */


        /*--Third column showing odds ratio graph--*/
        layout overlay / xaxisopts=( label=""
                         linearopts=(tickvaluepriority=true
                            tickvaluelist=(0.5 1.0 1.5 2.0 2.5 3.0)))
```

```
            yaxisopts=(reverse=true display=none)
↪ walldisplay=none;
               referenceline y=ref / lineattrs=(thickness=15
↪ color=_bandcolor);
      highlowplot y=obsid low=low high=high;
      scatterplot y=obsid x=hr /
↪ markerattrs=(symbol=squarefilled);
               referenceline x=1.861 /lineattrs=(color=red);
                referenceline x=1.00 /lineattrs=(color=black);

    endlayout;

    /*--Fourth column showing PCIGroup and Group columns--*/
    layout overlay / x2axisopts=(display=(tickvalues)
↪ offsetmin=0.15 offsetmax=0.15)
            yaxisopts=(reverse=true display=none)
↪ walldisplay=none;
               referenceline y=ref / lineattrs=(thickness=15
↪ color=_bandcolor);
      scatterplot y=obsid x=hr_lbl / markercharacter=hr xaxis=x2
            markercharacterattrs=graphvaluetext;
      scatterplot y=obsid x=low_lbl / markercharacter=low
↪ xaxis=x2
            markercharacterattrs=graphvaluetext;
      scatterplot y=obsid x=high_lbl / markercharacter=high
↪ xaxis=x2
            markercharacterattrs=graphvaluetext;
       scatterplot y=obsid x=p_lbl / markercharacter=p xaxis=x2
            markercharacterattrs=graphvaluetext;
             endlayout;
```

```
          endlayout;
        entryfootnote halign=left textattrs=(size=7)
           'The HR=1.861 fixed in the simulation is used as a
 ↪  comparator for the other methods (red line)';
        endgraph;
  end;
run;

/*--Render Forest Plot without horizontal bands--*/
ods graphics / reset width=7in height=5in
 ↪  imagename='Forest_HighLow_93';
proc sgrender data=Forest2 template=Forest;
dynamic _bandcolor='white' _headercolor='white';
run;

***************DOUBLY ROBUST;
data forest;
  input Id Subgroup $3-27  HR  Low  High P$;
  zero=0; one=1;
  HR_lbl='HR';
  low_lbl='Low';
  high_lbl='High';
  p_lbl='P-value';
  ObsId=_n_;
  if count ne . then CountPct=put(count, 4.0) || "(" || put(percent,
 ↪  3.0) || ")";
  datalines;
1 Naive.....................1.107  0.874  1.403  0.3985
1 Covariates Adjustment....1.857  1.363  2.530  <0.0001
1 Matching (DR).............      .      .      .
```

```
2 ..Greedy 1:1.............1.819  1.279  2.589  0.0009
2 ..Greedy 1:1 with caliper2.152  1.410  3.284  0.0004
2 ..Optimal 1:1............1.934  1.379  2.713  0.0001
2 ..Optimal 1:1/2..........2.001  1.432  2.796  <0.0001
1 IPTW (DR)..................      .      .      .
2 ..General weights........3.080  2.607  3.638  <0.0001
2 ..Stabilized wights......2.630  2.060  3.358  <0.0001
1 Stratification (DR).......      .      .      .
2 ..Tertiles...............2.102  1.501  2.942  0.0007
;
run;
ods listing;
/*proc print;run;*/

/*--Replace '.' in subgroup with blank--*/
data forest2;
  set forest;
  subgroup=translate(subgroup, ' ', '.');
  val=mod(_N_-1, 6);
  if val eq 1 or val eq 2 or val eq 3 then ref=obsid;

  /*--Separate Subgroup headers and obs into separate columns--*/
  if id=1 then do;
     heading=subgroup;
         subgroup='';
  end;
  run;
/*proc print;run;*/

/*--Create font with smaller fonts for axis label, value and data--*/
```

```
proc template;
   define style listingSF;
      parent = Styles.Listing;
      style GraphFonts from GraphFonts
         "Fonts used in graph styles" /
         'GraphDataFont' = ("<sans-serif>, <MTsans-serif>",7pt)
         'GraphValueFont' = ("<sans-serif>, <MTsans-serif>",7pt)
         'GraphLabelFont' = ("<sans-serif>, <MTsans-serif>",7pt,
 ↪  bold);
;
   end;
run;

%let dpi=150;
ods listing style=listingSF gpath=&gpath image_dpi=&dpi;

/*--Define templage for Forest Plot--*/
/*--Template uses a Layout Lattice of 5 columns--*/
proc template;
  define statgraph Forest;
  dynamic  _bandcolor _headercolor _subgroupcolor;
    begingraph;
      layout lattice / columns=3 columnweights=(0.23 0.4 0.3);

        /*--Column headers--*/
        sidebar / align=top;
              layout lattice / rows=2 columns=3 columnweights=(0.18
               ↪  0.35 0.3)
            backgroundcolor=_headercolor opaque=true;
```

```
    entry textattrs=(size=8 weight=bold) halign=left
     ↪ "Methods";
    entry textattrs=(size=8 weight=bold) "Hazard Ratio with
     ↪ 95% CI";
    entry halign=center textattrs=(size=8 weight=bold)
     ↪ "Statistics" ;
            entry " ";
    entry " ";
                entry " ";
    entry halign=center textattrs=(size=6) "Confidence
     ↪ interval";
          endlayout;
endsidebar;

/*--First Subgroup column, shows only the Y2 axis
 ↪ --*/
/*--Use HighLow plot to place the heading and subgroup values
 ↪ as HighLabels--*/
/*--Indenting is done by making the 2nd highlow bar 1 unit
 ↪ long            --*/
/*--Highlow bar itself has thickness=0
 ↪ --*/
layout overlay / walldisplay=none
       xaxisopts=(display=none linearopts=(viewmin=0
        ↪ viewmax=20))
       yaxisopts=(reverse=true display=none
        ↪ tickvalueattrs=(weight=bold));
  referenceline y=ref / lineattrs=(thickness=15
   ↪ color=_bandcolor);
```

```
        highlowplot y=obsid low=zero high=zero / highlabel=heading
        ↪ lineattrs=(thickness=0)
             labelattrs=(size=7 weight=bold);
        highlowplot y=obsid low=zero high=one / highlabel=subgroup
        ↪ lineattrs=(thickness=0);
      endlayout;

      /*--Second column showing Count and percent--
      layout overlay / xaxisopts=(display=none)
            yaxisopts=(reverse=true display=none)
↪ walldisplay=none;
              referenceline y=ref / lineattrs=(thickness=15
↪ color=_bandcolor);
        scatterplot y=obsid x=zero / markercharacter=countpct
            markercharacterattrs=graphvaluetext;
             endlayout;                              */

      /*--Third column showing odds ratio graph--*/
      layout overlay / xaxisopts=( label=""
                       linearopts=(tickvaluepriority=true
                          tickvaluelist=(0.5 1.0 1.5 2.0 2.5 3.0)))
            yaxisopts=(reverse=true display=none)
            ↪ walldisplay=none;
              referenceline y=ref / lineattrs=(thickness=15
              ↪ color=_bandcolor);
        highlowplot y=obsid low=low high=high;
        scatterplot y=obsid x=hr /
        ↪ markerattrs=(symbol=squarefilled);
              referenceline x=1.861 /lineattrs=(color=red);
              referenceline x=1 /lineattrs=(color=black);
```

```
        endlayout;

        /*--Fourth column showing PCIGroup and Group columns--*/
        layout overlay / x2axisopts=(display=(tickvalues)
         ↪  offsetmin=0.15 offsetmax=0.15)
               yaxisopts=(reverse=true display=none)
                ↪  walldisplay=none;
                  referenceline y=ref / lineattrs=(thickness=15
                   ↪  color=_bandcolor);
          scatterplot y=obsid x=hr_lbl / markercharacter=hr xaxis=x2
               markercharacterattrs=graphvaluetext;
          scatterplot y=obsid x=low_lbl / markercharacter=low
           ↪  xaxis=x2
               markercharacterattrs=graphvaluetext;
          scatterplot y=obsid x=high_lbl / markercharacter=high
           ↪  xaxis=x2
               markercharacterattrs=graphvaluetext;
           scatterplot y=obsid x=p_lbl / markercharacter=p xaxis=x2
               markercharacterattrs=graphvaluetext;
                endlayout;
          endlayout;
        entryfootnote halign=left textattrs=(size=7)
           'The HR=1.861 fixed in the simulation is used as a
            ↪  comparator for the other methods (red line). DR=
            ↪  Doubly Robust';
        endgraph;
  end;
run;

/*--Render Forest Plot without horizontal bands--*/
```

```
ods graphics / reset width=7in height=5in
↪  imagename='Forest_HighLow_93';
proc sgrender data=Forest2 template=Forest;
dynamic _bandcolor='white' _headercolor='white';
run;
```

List of Figures

List of Tables

Bibliography

[1] *"Clinical trial."* Wikipedia: The Free Encyclopedia. Wikimedia Foundation, 28 Nov 2018. Web. 11 Dec 2018. [https://en.wikipedia.org/wiki/Clinical_trial].

[2] *"Randomized controlled trial."* Wikipedia: The Free Encyclopedia. Wikimedia Foundation, 10 Nov 2018. Web. 11 Dec 2018. [https://en.wikipedia.org/wiki/Randomized_controlled_trial].

[3] Glen, Stephanie. *"Historical Controls."* Weblog entry. Statistics How To, Statistics for the rest of us!, 01 Mar 2017. Web. 11 Dec 2018. [https://www.statisticshowto.datasciencecentral.com/historical-controls/].

[4] Senn, Stephen. *"Statistical Issues in Drug Development."* Chichester: John Wiley & Sons Ltd, 2007. Print.

[5] *"Choice Of Control Group And Related Issues In Clinical Trials E10."* International Conference On Harmonisation Of Technical Requirements For Registration Of Pharmaceuticals For Human Use, 20 Jul 2000. Web. 11 Dec 2018. [https://www.ich.org/fileadmin/Public_Web_Site/ICH_Products/Guidelines/Efficacy/E10/Step4/E10_Guideline.pdf].

[6] *"Rare Diseases: Common Issues in Drug Development, Guidance for Industry."* U.S. Department of Health and Human Services, Food and Drug Administration, Center for Drug Evaluation and Research (CDER), Center for Biologics Evaluation and Research (CBER), August 2015. Web. 11 Dec 2018. [https://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/UCM458485.pdf].

[7] *"Patient registries vs. Clinical trials."* Eversheds Sutherland, 01 Apr 2013. Web. 11 Dec 2018. [https://www.eversheds-sutherland.com/global/en/what/publications/shownews.page?News=en/ireland/patient-registries-vs-clinical-trials-april-2013].

[8] *"List of Registries."* National Institutes of Health, 14 Nov 2018. Web. 11 Dec 2018. [https://www.nih.gov/health-information/nih-clinical-research-trials-you/list-registries].

[9] Outcome Sciences, Inc., A Quintiles Company. *"Registries for Evaluating Patient Outcomes: A User's Guide."* Vol.1, Cambridge, Massachussets: Michelle B. Leavy, M.P.H., 2014. Print.

[10] *"Missing data and multiple imputation."* Computing for the Social Sciences, Web. 11 Dec 2018. [https://cfss.uchicago.edu/persp014_missing_data.html].

[11] Kelly, Brian. *"Registries and the Future of Medicine."* Weblog entry. PharmExec.com, 16 May 2016. Web. 11 Dec 2018. [http://www.pharmexec.com/registries-and-future-medicine].

[12] *"Guideline on Missing Data in Confirmatory Clinical Trials."* European Medicines Agency, 2 Jul 2010. Web. 11 Dec 2018. [https://www.ema.europa.eu/documents/scientific-guideline/guideline-missing-data-confirmatory-clinical-trials_en.pdf].

[13] *"Confounding."* Wikipedia: The Free Encyclopedia. Wikimedia Foundation, 9 Nov 2018. Web. 11 Dec 2018. [https://en.wikipedia.org/wiki/Confounding#cite_note-Pearl_2009-1].

[14] Georg Heinze, Peter Jüni. *"An overview of the objectives of and the approaches to propensity score analyses."* European Heart Journal, Volume 32, Issue 14, 1 July 2011, Pages 1704–1708, [https://doi.org/10.1093/eurheartj/ehr031].

[15] Kuss Oliver, Blettner Maria, Börgermann Jochen. *"Propensity score: an alternative method of analyzing treatment effects—part 23 of a series on evaluation of scientific publications."* Dtsch Arztebl Int 2016, 113: 597–603. DOI: 10.3238/arztebl.2016.0597. [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5963493/pdf/Dtsch_Arztebl_Int-113-0597.pdf].

[16] Austin, Peter C. *"An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies."* Multivariate Behavioral Research, 09 Jun 2011, 46:3, 399-424, DOI: 10.1080/00273171.2011.568786. [https://www.tandfonline.com/doi/full/10.1080/00273171.2011.568786?scroll=top&needAccess=true].

[17] Elze Markus C., Gregson John, Baber Usman, Williamson Elizabeth, Sartori Samantha, Mehran Roxana, Nichols Melissa, Stone Gregg W., Pocock Stuart J. *"Comparison of Propensity Score Methods and Covariate Adjustment: Evaluation in 4 Cardiovascular Studies."*, Journal of the American College of Cardiology, Volume 69, Issue 3, 2017, Pages 345-357, DOI: 10.1016/j.jacc.2016.10.060. [https://www.sciencedirect.com/science/article/pii/S073510971637036X#!].

[18] Righini, Giovanni.*"Minimum cost bipartite matching - Complements of Operations Research."* Università degli Studi di Milano. [https://homes.di.unimi.it/righini/Didattica/OttimizzazioneCombinatoria/MaterialeOC/5%20-%20Min%20cost%20bipartite%20matching.pdf].

[19] Xu Stanley, Ross Colleen, Raebel Marsha, Shetterly Susan, Blanchette Christopher, Smith David. *"Use of Stabilized Inverse Propensity Scores as Weights to Directly Estimate Relative Risk and Its Confidence Intervals."*. Value in health : the journal of the International Society for Pharmacoeconomics and Outcomes Research, 2009. 13. 273-7. DOI: 10.1111/j.1524-4733.2009.00671.x. [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4351790/pdf/nihms-475004.pdf].

[20] Therneau Terry M., Grambsch Patricia M. *"Modeling Survival Data: Extending the Cox Model."* New York, Springer, 2000. Print.

[21] *"Kaplan–Meier estimator."* Wikipedia: The Free Encyclopedia. Wikimedia Foundation, 14 Aug 2018. Web. 11 Dec 2018. [https://en.wikipedia.org/wiki/Kaplan%E2%80%93Meier_estimator].

[22] Xie Jun, Liu Chaofeng. *"Adjusted Kaplan–Meier estimator and log-rank test with inverse probability of treatment weighting for survival data."* Statistics in Medicine, 2005. 24:3089–3110, Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/sim.2174. [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.135.1246&rep=rep1&type=pdf].

[23] *"Estimatig the Survival Function."* UC San Diego - Department of Mathematics. [http://www.math.ucsd.edu/~rxu/math284/slect2.pdf].

[24] *"Cox Proportional-Hazards Model."* STHDA - Statistical tools for high-throughput data analysis. [http://www.sthda.com/english/wiki/cox-proportional-hazards-model].

[25] *"Proportional hazards model."* Wikipedia: The Free Encyclopedia. Wikimedia Foundation, 15 Aug 2018. Web. 11 Dec 2018. [https://en.wikipedia.org/wiki/Proportional_hazards_model].

[26] Buchanan Ashley L., et al. *"Worth the weight: using inverse probability weighted Cox models in AIDS research."* AIDS research and human retroviruses vol. 30,12: 1170-7, 2014, DOI: 10.1089/aid.2014.0037. [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4250953/].

[27] *"The PHREG Procedure."* SAS/STAT(R) 9.3 User's Guide. [https://support.sas.com/documentation/cdl/en/statug/63962/HTML/default/viewer.htm#statug_phreg_sect042.htm].

[28] Crowther Michael J., Lambert Paul C. *"Simulating biologically plausible complex survival data."* Statistics in Medine, 2013. 32: 4118-4134. DOI: 10.1002/sim.5823 [https://lra.le.ac.uk/bitstream/2381/33061/2/Crowther2013%20(2).pdf].

[29] *"Logrank Test."* Wikipedia: The Free Encyclopedia. Wikimedia Foundation, 27 Nov 2018. Web. 11 Dec 2018. [https://en.wikipedia.org/wiki/Logrank_test].