ESTIMATING THE EFFECT OF TRANEXAMIC ACID ON HEAD TRAUMATIZED PATIENTS WITH MATCHING METHODS

Charlotte Gallezot, Gauthier Guinet, Chiara Régniez
Alodie Boissonnet, Barnabé Mas, Samy Alsharani
Ecole Polytechnique
surname.name@polytechnique.edu

December 17, 2019

ABSTRACT

The goal of our project is to study the causal effect of tranexamic acid using on headtraumatized patients, using a dataset provided by the AP-HP, gathering medical information on victims of major trauma. In observational studies such as ours, one of the major difficulties is to deal with the absence of a proper control group. Indeed, contrary to randomized experiment, there is a bias in the administration of the treatment.
To cope with this issue, we focused on the use of matching methods. In (1), we introduce the associated statistical framework. In (2), we describe three of the main techniques used in matching: coarsened exact matching, cardinality matching and propensity matching. As little statistical evidence of these techniques' efficiency is known, we designed and implemented experiments to provide empirical results. We generated several synthetic datasets with various underlying complexity and compared our methods on these datasets. In (3), we expose the methodology and results of this experiment. Eventually we applied our methods on the traumabase after pre-processing it. This gave us an estimation of the treatment effect, that we present in (4). Finally, as the use of traumabase implied missing data imputation, we studied the robustness to missing data of our three matching methods in (5).

***K*eywords** Causal Inference · Matching Methods · Robustness in Missing Values Models · Traumabase

1 Introduction

1.1 Matching motivations

Major trauma such as head injuries or hemorrhage are a real public health challenge. It is the third cause of death in France and the third cause of disability. In the case of head injuries, a common complication that often leads to death is intracranial bleeding. Tranexamic acid has revealed to be a potential treatment to slow the bleeding. If proven to be efficient, it could thus contribute to reduce the number of death by head injuries.
In this project we try to assess the effect of tranexamic acid on patients who suffer from head injuries. To do so, we use the traumabase, an AP-HP (Paris' hospitals) database which gathers information on head trauma victims. In such observational studies, one major issue is to deal with bias. Indeed, contrary to randomized experiments we do not have a proper control group, and there is a bias in the attribution of the treatment. To cope with this issue, we will use matching techniques: every treated unit will be paired to one (or more) non-treated unit(s) with similar observable characteristics against whom the effect of the treatment can be assessed.
Matching can be defined as any method that "strategically subsamples" a dataset, with the aim of balancing observable covariate distributions in the treated and control groups such that both groups share an equal probability of treatment. An important distinction from other statistical approaches is that matching is only the first step in a two-step process to estimate the treatment effect. It prepares the data for further statistical analysis, but it is not a stand-alone estimation technique in and of itself. Matching is followed by difference-in-average estimation (assuming sufficient covariate coverage), linear regression, or any other modeling method.

1.2 Statistical Framework and Assumptions

We consider a model with n independent and identically distributed samples (X_i, Y_i, W_i), were X_i are observed variables, W_i the treatment variable ($W_i = 1$ if i is treated and $W_i = 0$ otherwise), $Y_i(w)$ the outcome for individual i if he is treated ($w = 1, Y_i(1)$) or if he is not treated ($w = 0, Y_i(0)$), and $Y_i(w) \in [0, 1]$. We assume :

$$Y_i = W_i Y_i(1) + (1 - W_i)Y_i(0) \quad \text{(SUTVA hypothesis)}.$$

What we wish to estimate with our database is the effect of tranexamic acid on head traumatized patient. This causal effect for an individual i can be defined as follows: $\Delta_i = Y_i(1) - Y_i(0)$.

As a patient can not be treated and not treated at the same time, Δ_i can never be observed. We want to estimate the average treatment effect (ATE) : $t = \mathbb{E}[Y_i(1) - Y_i(0)]$ or the average treatment effect on the treated (ATT) : $ATT = \mathbb{E}[Y_i(1) - Y_i(0)|W_i = 1]$. Depending on the context, one of these two estimators might be preferred. The ATE computes the outcome had everyone gotten treated minus the outcome had nobody gotten treated. The ATT computes the difference between the outcome of the treated and the outcome had the treated been not treated. The ATT is more interesting in certain cases where the not treated population is not concerned in anyway by the treatment. For instance, we do not care about the outcome for cancer-free people had they received chemotherapy. In our case, the ATE is more relevant since the clean database gathers information only on head-traumatized people.

One way to solve this problem is to conduct randomized controlled trials. In this kind of experiment, we have the following hypothesis:

$$W_i \perp\!\!\!\perp \{Y_i(0), Y_i(1)\} \quad \text{(Random Treatment Assignment)}$$

And thus, the difference in means estimator :

$$\hat{\tau}_{DM} = \frac{1}{n_1} \sum_{W_1=1} Y_i - \frac{1}{n_0} \sum_{W_1=0} Y_i$$

is unbiased and $\sqrt{n}$-consistent.

We face there a second problem. In our case, the treatment is not assigned randomly. It generates a bias and we can not use this estimator without preliminary work. To deal with this kind of situation, various methods have been developed. We will focus here on matching methods.

Matching methods' goal is to reduce this bias to estimate the ATE or the ATT. It is based on the unconfoundedness assumption (thrid assumption), that treatment assignment is random conditionally on X_i :

$$\{Y_i(0), Y_i(1)\} \perp\!\!\!\perp W_i | X_i = x, \quad \forall x \in \mathcal{X}.$$

The idea is that under this assumption, we can come down to the RCT (randomized controlled trials) situation if we consider the right group of individuals. In a group formed of individuals with similar covariates, we will be able to assume that the treatment is assigned randomly. To compute the tratment after matching we used the difference in means estimator described above.

2 Presentation of Matching Methods

2.1 Propensity Score Matching

Propensity score matching is the most used matching techniques. The propensity score represents the probability someone has to undergo the treatment depending on his characteristics. It is defined as follows:

$$e(x) = \mathbb{P}(W_i = 1 | X_i = x) \quad \forall x \in \mathcal{X}.$$

We assume $0 < e(x) < 1$ (overlap assumption). It means that everyone has a chance to get the treatment and no one is sure to be treated. And it has the following property, under unconfoundedness, $e(x)$ satisfies :

$$\{Y_i(0), Y_i(1)\} \perp\!\!\!\perp W_i | e(X_i).$$

Rosenbaum and Rubin showed in 1983 [1] that thanks to this property, matching on the propensity score was a way to eliminate the bias due to the covariates. However this is only relevant if the treatment is not given on the basis of other covariates that we do not observe. Moreover, the individuals compared have to be sufficiently alike or the comparison will not mean anything.

The propensity score is often unknown, and it has to be estimated. Hence the quality of the propensity score matching depends on the quality of the estimation. Indeed, the unconfoundedness property is only true for the real propensity score. Different techniques can be used to estimate the propensity score. A simple logistic regression can be applied, but we can also use more sophisticated machine learning algorithms such as random forests or neural networks.

Once it is estimated the ATE estimator can be computed the following way :

$$\hat{Y}_i(0) = \begin{cases} Y_i & \text{if } W_i = 0; \\ \frac{1}{M} \sum_{j \in \mathcal{J}_M(i)} Y_j & \text{if } W_i = 1. \end{cases}$$

$$\hat{Y}_i(1) = \begin{cases} \frac{1}{M} \sum_{j \in \mathcal{J}_M(i)} Y_j & \text{if } W_i = 0; \\ Y_i & \text{if } W_i = 1. \end{cases}$$

$$\hat{\tau}_M^m = \frac{1}{N} \sum_{i=1}^{N} (\hat{Y}_i(1) - \hat{Y}_i(0)),$$

with $\mathcal{J}_M(i)$ the indexes of the M nearest individuals in terms of estimated propensity score. If the matching is done with replacement, the same individual can be used as a match several times. If the matching is done without replacement, the order of the matching has to be defined.

A caliper can be imposed on the covariates distance as well. With such a caliper, two individuals can only be matched if the distance between their covariates is smaller than the caliper. According to P.C. Austin [2], the best way to proceed is:

- matching in random order;
- without replacement;
- imposing a caliper of 0.2 times the standard deviation of the propensity score.

2.2 Coarsened Exact Matching (CEM)

2.2.1 Concept

The most common matching technique, Propensity Score Matching, however, is slow and quite difficult to apply. Coarsened Exact Matching offers an alternative solution, which is faster and easier to understand. CEM starts by transforming continuous variables into categorical variables. This technique in the machine learning is often referred to as discretization. After 'filling' the bins, control units within each bin are weighted to equal the number of treated units in that stratum. Strata without at least one treated and one control are weighted at zero, and thus pruned from the data set. More precisely, the CEM algorithm then involves three steps :

- Temporarily coarsen each control variable in X (covariates) according to user-defined cutpoints, or with CEM's automatic binning algorithm, for the purposes of matching. For example, years of education might be coarsened into grade school, middle school, high school, college, graduate school.
- Sort all units into strata, each of which has the same values of the coarsened X.
- Prune from the data set the units in any stratum that do not include at least one treated

The 2 mains problems that can be encountered using coarsened exact matching are:

- Matching can be completely off if the wrong variables are chosen. Example: There may be dramatic differences between male / female members, if matching does not consider gender, then the matching may never work.
- If the right variables are chosen, but the coarsening is too loose. Example: Age could be binned into 1 or 0 depending on if a member is $\geq$ 50 years old or < 50 years old — for some studies that might be appropriate, but working on a geriatric study, almost everyone will be $\geq$ 50 years old, and this coarsening strategy is inappropriate and too lose.

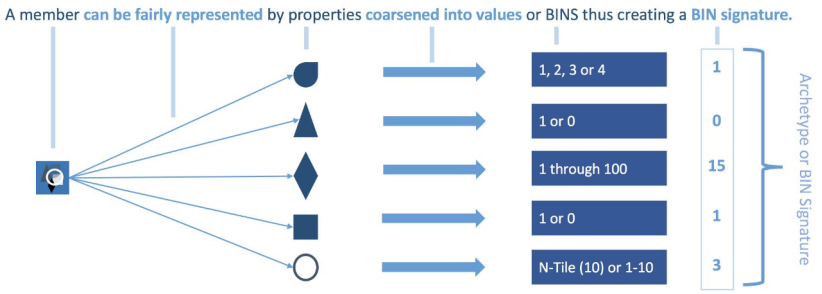

Figure 1: CEM Presentation [Source:https://medium.com/@devmotivation/cem]

2.2.2 Library

The CEM method was mainly studied by Garry King who designed a R package, `cem`, containing all the required functions to carry on coarsened exact matching. The coarsening can be made according to user-defined cutpoints, or CEM's automatic binning algorithm. Then the matching among people within the selected sample is achieved with the `att` function also in the `cem` package.

2.3 Cardinality Matching

2.3.1 Concept

Quite recent, Cardinality Matching's methods are based on a simple principle: match the largest number of individuals respecting balance constraints between covariates. This meets a twofold objective:

- reduce the bias with the balance constraint between the covariates;
- reduce variance by taking as many individuals as possible.

We set $m_{tc} = 1$ if the treated person t and the untreated one c (c for control) are matched ans 0 otherwise. Cardinality matching is therefore the same as solving the following integer linear problem:

$$\begin{aligned}
\underset{m}{\text{maximize}} \sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} m_{tc}, \\
m_{tc} \in \{0,1\}, \\
\sum_{t\in\mathcal{T}} m_{tc} \leq 1, \quad c \in \mathcal{C}, \\
\sum_{c\in\mathcal{C}} m_{tc} \leq 1, \quad t \in \mathcal{T}, \\
\left|\sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} m_{tc}(f_k(x_{tp}) - f_k(x_{cp}))\right| \leq \varepsilon_p \sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} m_{tc}.
\end{aligned}$$

f_k is a function chosen by the user that transforms the covariates x_p. Likewise ε_p is a threshold chosen by the user for each covariate. With both these parameters, the user can choose to enforce maximum levels of standardized differences in means but also balance between higher moments of the covariate distribution. Finally, with $\varepsilon_p = 0$, one may ensure exact matching on a specific covariate. This is particularly useful as it makes cardinality matching able to handle both quantitative and categorical covariates.
More precisely, exact matching is achieved with the constraint:

$$\sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} m_{tc}\mathbf{1}_{\{x_{tp}=b\}\cap\{x_{cp}\neq b\}} = 0, \quad b \in \mathcal{B}.$$

And ensuring that the differences in means are below the threshold ε_p is achieved with:

$$\left|\frac{\sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} m_{tc}x_{tp}}{\sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} m_{tc}} - \frac{\sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} m_{tc}x_{cp}}{\sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} m_{tc}}\right| \leq \varepsilon_p.$$

Cardinality matching is more precisely done in three steps:

1. The user first chooses and specifies the covariates balance rules;
2. Then the PLNE is solved using available solver. This brings balance between covariates;
3. Then among the people selected by the PLNE, a new matching minimzes the total sum of the covariates distance between matched people. This steps reduced the heterogenity between matched people.

2.3.2 Alternative methods

Moreover, alternative cardinality matching techniques can allow *1 to many* matching. This means that a single treated people can be matched to several other control people (the other way round can also be imposed but is less frequently mentioned as there is often more control people than treated). Only one constraint is changed to allow the treated to be matched with several people.
For instance to do a cardinality matching with *1 to K* matching one has to solve:

$$\begin{aligned}
&\underset{m}{\text{maximize}} \sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} m_{tc},\\
&m_{tc} \in \{0,1\},\\
&\sum_{c\in\mathcal{C}} m_{tc} \leq K, \quad t \in \mathcal{T},\\
&\sum_{t\in\mathcal{T}} m_{tc} \leq 1, \quad c \in \mathcal{C},\\
&\left|\sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} m_{tc}(f_k(x_{tp}) - f_k(x_{cp}))\right| \leq \varepsilon_p \sum_{t\in\mathcal{T}}\sum_{c\in\mathcal{C}} m_{tc}.
\end{aligned}$$

2.3.3 Library

The Cardinality Matching method was mainly studied by *Zubizarreta* who designed a R package, `designmatch`, containing all the required functions to carry on cardinality matching. The sample selection (step 2 of the cardinality matching process) is achieved with the `cardmatch` function. Then the new matching among people in the selected sample (step 3 of the cardinality matching process) is achieved with the `distmatch` function also in the `designmatch` package.

2.3.4 Implementation

We used the *glpk* solver easily and freely accesible on the internet. If one wanted to increase the speed of the solver, using solvers like *cplex* or *gurobi* could improve the speed of the matching methods. Both are free for academic or research purposes but *cplex* requires access to the data used which was not an option for the analysis on the AP-HP database.

When testing the matching methods on the generated datasets, we used 2 covariate balance rules. We enforced exact matching on categorical covariates (ie used a 0 threshold). For the quantitative covariates we compared the results with 2 threshold: 0.1 and 0.01.

Moreover the last step of the cardinality matching method (rematching in the selected sample) can be done in several ways. We decided to adopt the most general approach which does not require any previous knowledge on the structure of the dataset: we rematched all the selected people to diminish the sum of the euclidian distance between each pair of matched people:

$$\begin{aligned}
&\min \sum_{t\in\overline{\tau},c\in\overline{C}} m_{tc}||t-c||,\\
&\forall t \in \overline{\tau}, \quad \sum_{c\in\overline{C}} m_{tc} = 1,\\
&\forall c \in \overline{C}, \quad \sum_{t\in\overline{\tau}} m_{tc} = 1,
\end{aligned}$$

$$m_{tc} \in \{0, 1\}$$

where $\overline{\tau}$ is the set of treated people selected after first step of matching, and $\overline{C}$ the set of non-treated people selected after first step of matching.

3 Comparing Matching Methods

3.1 Methodology

One of the major drawbacks of matching methods is that they often lack theoretical results highlighting their performances. As our target was to evaluate the treatment effect on a real database we wanted to be able to know more about the characteristics of our matching methods to analyze our final results. Therefore, we decided to implement the methodology of Zubizarreta [3] which compared some matching methods on several generated datasets. Our contribution was to add more matching methods than the original paper as well as to propose new scenarios. We adapted his framework to missing values models as this was a major issue in our case. We also had to reimplement his approach, by making some choices presented below

We generated three kinds of datasets based on three different models. Each of them consisted of 250 treated people and 250 control people. We created 8 covariates *X1, X2, X3, X4, X5, X6, X7, X8*, an continuous outcome *Y* and a treament value *Treat* indicating if a person had received the treatment. $Treat = 1$ meaning that the person has been treated and $Treat = 0$ meaning the opposite . In order to assess the efficency of matching methods to balance treated and control group, we fixed the true standardized difference in means for each covariate between those 2 groups at 0.2.

Among the covariates:

- *X1, X2* were common Bernouilli, independent from all the other one and from the other covariates.
- *X3, X4* were rare Bernouilli, with same independence assumptions.
- *X5, X6, X7, X8* were generated following a multivariate normal distribution. The main difference between our scenarios were the parameters of this distribution.

From a global point of view, the datasets have increasing complexity in covariate repartition:

1. For the first kind of dataset (*Dataset 1*), all the covariates are independant with the same variance, $\sigma^2 = 1$, but with different means, $\mu = 0.5$ for the treated and $\mu = 0.3$ for the control group.
2. For the second kind of dataset (*Dataset 2*), all the covariates are independant but they have different variance and different means. For the treated $\mu = 0.5$ and $\sigma^2 = 1.2$ and for the control group $\mu = 0.2737$ and $\sigma^2 = 1$.
3. For the third kind of dataset (*Dataset 3*), the covariates of the control group are still independant but those in the treated group aren't. For the treated $\mu = 0.5$ and $\sigma^2 = 1.2$ and for the control group $\mu = 0.2737$ and $\sigma^2 = 1$. The covariance matrix in the treated group was fixed at:
$$Cov(\begin{pmatrix} X5 \\ X6 \\ X7 \\ X8 \end{pmatrix}) = \begin{pmatrix} 1.2 & 0.5 & 0.8 & 0.8 \\ 0.5 & 1.2 & 0.1 & 0.8 \\ 0.8 & 0.1 & 1.2 & 0.1 \\ 0.8 & 0.8 & 0.1 & 1.2 \end{pmatrix}$$

In addition of testing our matching methods on different data structures, we also tested different relationships *Y* (the outcome) and its covariates *X1, X2, X3, X4, X5, X6, X7, X8*. More precisely we generated the *Y* with the following relationship with $\epsilon = \mathcal{N}(0, 4)$

$$Y = f(X) + Treat + \epsilon$$

This means that the treatment effect has a value of one (because *Treat* is either equal to 1 or to 0). In this model we tested three kinds of relationships between *Y* and *X1, X2, X3, X4, X5, X6, X7, X8* with three f function:

- Linear, $f_1(x) = 3.5x_1 + 4.5x_3 + 1.5x_5 + 2.5x_7$,
- Additive, $f_2(x) = 3.5x_1 + 4.5x_3 + 1.5x_5 + 2.5x_7 + 2.5sign(x_1)|x_1|^{1/2} + 5.5x_3^2$
- Additive with interactions, $f_3(x) = 3.5x_1 + 4.5x_3 + 1.5x_5 + 2.5x_7 + 2.5sign(x_1)|x_1|^{1/2} + 5.5x_3^2 + 2.5x_3x_7 - 4.5|x_1x_3^3|$

Consequently, we generated three kinds of datasets with three causal relationships between the condition of the people and their covariates. Our analysis focused on comparing the estimation of the treatment effect achieved by the different matching methods on these 9 scenarios. To do so, we generated 500 times the dataset corresponding to each scenario, applied the matching techniques, computed the treatment effect with the difference in means estimator and compared the results.

The metrics on which we focused to compare the efficiency of matching techniques were :

- the treatment's effect estimation
- the covariates balance
- for each covariate, the ratio between the variance of the treated and the variance of the control group among the matched people
- the Kolmogorov-Smirnov distance (the uniform norm between the cumulative distribution functions)
- the ratio of the number of selected people on the total number of people

3.2 Results on simulation

As said above, our goal was to compare the matching methods in scenarios of increasing complexity. This complexity is percepted both through the impact of covariates in treatment effect and through the covariates distribution. After some preliminaries simulations to choose only the most efficent methods, we decided to focus on the following:

- Coarsened Exact Matching (with automatic binning algorithm)
- Propensity Matching using Logistic Regression
- Propensity Matching using Random Forest
- Cardinality matching with threshold 0.1
- Cardinality matching with threshold 0.01

For each scenario, we created 500 synthetic datasets and estimated the main results of our matching methods with Monte Carlo techniques. Those implementations took us around 3 to 5 hours per scenarios.

The main results we could deduce were:

- Imbalance reduction:
  - As expected, matching methods do reduce the difference in mean, or in other words, the balance between the distribution in control and treatment groups. Such results were noticed in the difference in means but also in the ratio of variance and the K-S distance between the two distributions. Therefore, to use a matching method appears to be a good preliminary tool.
  - With just this goal in mind, cardinality matching is definitely the most efficient technique, followed by coarsened exact matching, then propensity matching. This is in part explained by the fact that cardinality matching is built to reduce this imbalance whereas it is only a consequence in the case of CEM or propensity matching.
  - Concerning the number of observation dropped, the results are strongly correlated with the previous observation: cardinality matching drops around 40% of observations, CEM around 12% and propensity around 2%. Thus, the effect on balance reduction can be counterbalanced by problems for ATT estimated due to an important drop of observations.
  - For qualitative covariates (X1 and X3), in our case (low-dimension datasets), CEM and Cardinality matching perform exact matching between control and treatment. As there are few of them, it doesn't appear as a problem but leads to a drop of many values in the case of cardinality. The Traumabase doesn't have many qualitative covariates in our case and they are all converted in quantitative after FAMD so this effect is not to be considered but shouldn't be underestimated in other scenarios. Propensity Matching with random Forest seems to handle difficultly those covariates. It is perhaps because of the small number of observations in our datasets as we do not observe this phenomenon for propensity matching with logistic regression. We do not observe great differences between the different scenarios.
  - For quantitative covariates (X5 and X7), cardinality matching with threshold 0.01 is definitely the most efficient method, followed by the others who behave quite similarly (except random forest again for the same problem). Cardinality balance is much more concentrated in both case.

Diff. in means		Dataset 1			Dataset 2			Dataset 3		
		Linear	Additive	Interrac.	Linear	Additive	Interrac.	Linear	Additive	Interrac.
X1	Before matching	28.9	20.1	23.9	23.5	22.1	16.7	23.7	17.8	21.6
	CEM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Prop logit	8.83	7.43	4.88	6.55	8.02	10.0	5.24	7.59	11.1
	Prop RF	15.0	12.9	22.7	14.9	13.1	11.4	15.8	16.0	15.2
	Card 0.1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Card 0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
X3	Before matching	18.3	20.1	23.6	12.4	17.2	19.3	17.5	15.4	18.5
	CEM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Prop logit	7.54	3.99	9.65	6.57	4.76	8.02	7.04	5.41	6.57
	Prop RF	12.2	13.0	8.87	14.6	8.99	11.2	10.3	15.7	16.0
	Card 0.1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Card 0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
X5	Before matching	24.2	25.0	16.4	19.9	22.2	21.3	22.0	20.8	23.7
	CEM	9.95	7.47	8.75	6.13	8.46	6.72	8.94	7.51	7.48
	Prop logit	3.44	5.34	11.5	5.45	4.62	2.39	6.83	4.87	7.76
	Prop RF	16.0	15.4	15.5	14.6	6.30	13.1	12.3	16.8	16.8
	Card 0.1	8.26	9.04	7.59	7.95	7.56	7.39	8.63	9.00	8.57
	Card 0.01	0.73	0.72	0.92	0.64	0.61	0.59	0.78	0.72	0.54
X7	Before matching	25.4	29.0	14.8	22.4	25.2	20.3	15.6	20.7	21.9
	CEM	8.26	2.97	8.46	4.75	4.84	5.59	9.54	16.3	6.51
	Prop logit	4.11	7.74	4.53	6.24	7.87	7.69	5.35	5.33	4.94
	Prop RF	14.4	17.8	5.74	9.26	11.6	9.14	9.49	12.9	19.6
	Card 0.1	9.54	9.61	9.19	8.19	8.68	8.24	8.24	8.87	9.42
	Card 0.01	0.87	0.75	0.77	0.69	0.62	0.67	0.68	0.66	0.54

Table 1: Difference in means (in %)

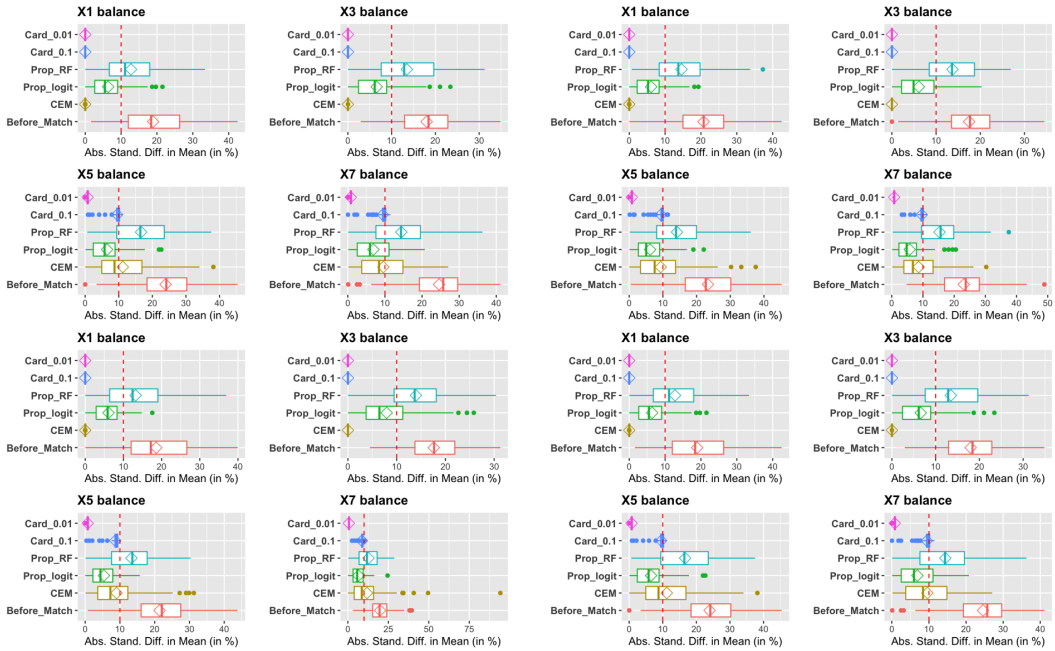

Figure 2: Plots of Diff. in Means for scenario 1 (*Dataset 1* with f_1), scenario 3 (*Dataset 3* with f_1), scenario 6 (*Dataset 2* with f_3) and scenario 9 (*Dataset 3* with f_3)

- ATT estimation:
  - As expected, all matching methods do improve estimated ATT in all scenarios. This is quite a strong and positive result. On the other hand, the simple difference in means estimator (without matching) has pretty bad results in all scenarios so the benchmark is quite easy to beat.
  - CEM performs well in all type of scenarios, with good approximation of real treatment effect in mean but a high variance that make it hard to use in real "one shot" situation, such as for the traumabase. For the more complex scenarios (8 and 9), it is the only matching method with good results. This robustness is perhaps linked to the simplicity of it model, as suggested by G. King in his paper. Nonetheless, it isn't a good choice because of the variance problem.
  - Propensity matching with logistic regression has great results for simpler scenarios (1 to 6), with a low variance. Such strong properties are surely the reason of its popularity in real life. Moreover, it is appropriated for high dimensions, whereas cardinality and coarsened matching aren't has they use simpler distances. The choice of random forest to estimate propensity score is not pertinent in our case as we don't have enough observations. It consistently underperform logistic regression.
  - Cardinality matching has also great results in scenarios 1 to 6. The tradoffs to consider when choosing the threshold is computation time vs precision as it appears that reducing the threshold consistently improves ATT estimation. To compare it with propensity matching (as they behave very similarly here), their performances are mostly similar in scenarios 3 and 4 and 7. For the rest, cardinality is slightly closer to real ATT.
  - In conclusion, using a matching method is definitly a good choice when estimating causal effect. Cardinality and propensity matching appear to be the most efficient and reliable (low variance) algorithms. Nevertheless, when the complexity of the treatment effect or of covariate distribution is too high, these methods aren't really usefull (eventhough they beat benchmark).

Mean ATT	Dataset 1			Dataset 2			Dataset 3		
	Linear	Additive	Interrac.	Linear	Additive	Interrac.	Linear	Additive	Interrac.
Before matching	2.53	2.34	-2.1	3.01	2.96	-2.7	2.32	2.96	-4.1
CEM	1.23	1.31	0.95	1.38	1.45	1.28	1.04	1.25	1.11
Prop logit	1.17	1.25	1.07	0.89	0.35	0.49	1.06	2.96	-2.3
Prop RF	2.01	-2.4	-0.9	0.33	0.24	2.03	1.98	2.96	-3.7
Card 0.1	1.31	1.29	0.23	0.35	0.56	1.47	1.66	2.96	-2.2
Card 0.01	1.05	1.13	0.88	1.21	1.13	1.06	1.09	0.12	-1.6

Table 2: ATT estimation

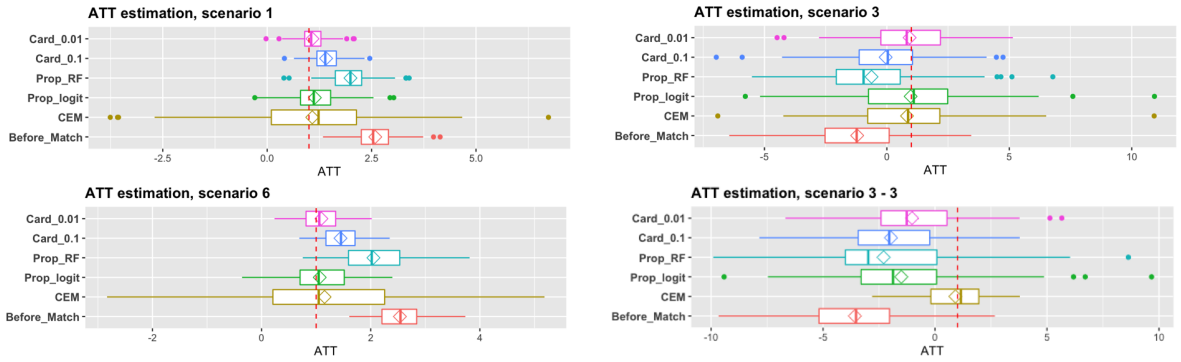

Figure 3: ATT Estimation for scenario 1 (*Dataset 1* with f_1), scenario 3 (*Dataset 3* with f_1), scenario 6 (*Dataset 2* with f_3) and scenario 9 (*Dataset 3* with f_3)

3.2.1 Propensity matching

For the propensity score method, we will look more closely at whether the quality of the propensity score estimation has an impact on the result or not. Quite clearly, the estimation of the propensity score with regression is more efficient than with random forest.

Considering estimation with logistic regression, it can be said at the end of these tests that for the same dataset, whatever the covariate effect complexity is, we will obtain approximately the same precision for the ATT/ATE.

For example, the scenario 6 (mid dataset complexity, high covariate effect complexity), the tests have shown that the propensity method with logistic regression gave a good approximation of the result. However, the linear regression model was not suitable as the 4 graphs below confirm.

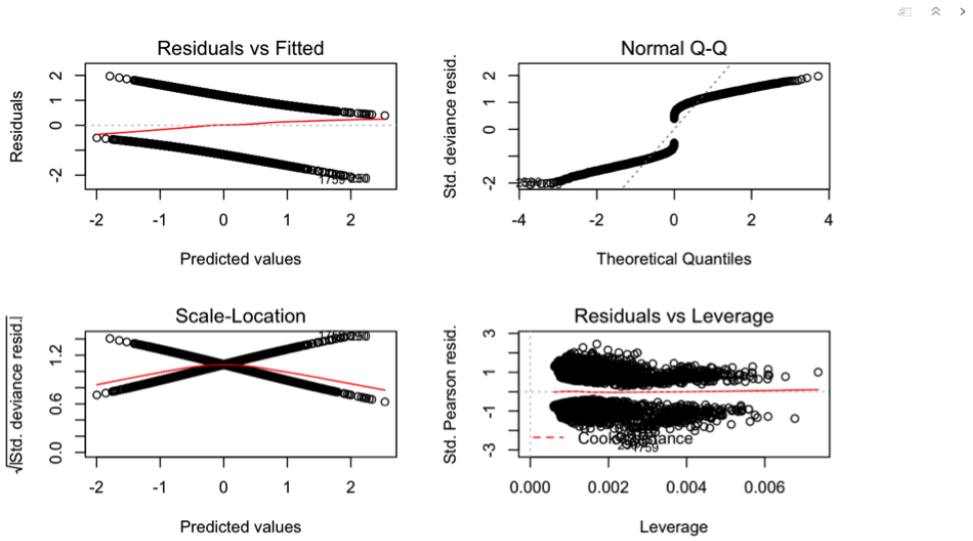

Figure 4: Diagnostic plot scenario 6 Logistic Regression Propensity Score

Thus, it can be concluded in the case of propensity score matching that the quality of the estimate is not essential, but that this model only works if the dataset has low correlations between its covariates. The most important thing is that a good propensity score matching is a balancing tool: it must make it possible to balance the distribution of the variables chosen in the two groups.

Another interesting question in the propensity method is the impact of a caliper, of the replacement possibility and of the number of neighbors considered during the matching process. We study these three questions under Scenario 1.

- Caliper: Tests show that a caliper considerably increases the quality of the estimate. This seems logical since two individuals with the same propensity score can however be very different. The caliper thus prevents two very different individuals with similar propensity scores from being matched.
- Replacement: The tests show that the quality of the ATT estimation is better with replacement.
- k-NN: The tests show that the quality of the ATT estimation is better considering only one neighbor.

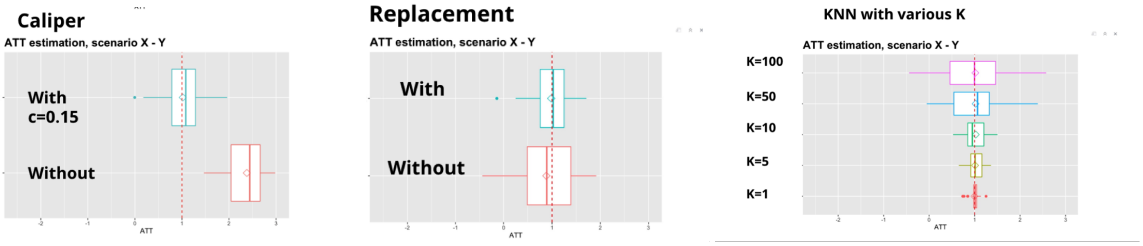

Figure 5: ATT estimation with caliper, replacement and k-NN for different k

4 Application to the Traumabase

4.1 Pre-processing of the database

After studying and comparing different matching methods, it is necessary to work on the database to extract the variables of interest, and format them for matching methods. The database under study gathers about 20,000 patients, for whom more than 270 characteristics were collected. We focus our analysis on about 9,000 who suffered from a head trauma and seventeen covariates, identified as the most relevant ones to match patients and assess the effect of a treatment.

However, before isolating these variables, two difficulties must be overcome : many data are missing and the dataset is made of mixed data. First, our dataset has a large amount of missing data. Notably, more than 30% of values of all patients are missing for one of the selected covariates, and if we only work on the patients who suffered from a head injury, there are sometimes nearly 40% of missing values.

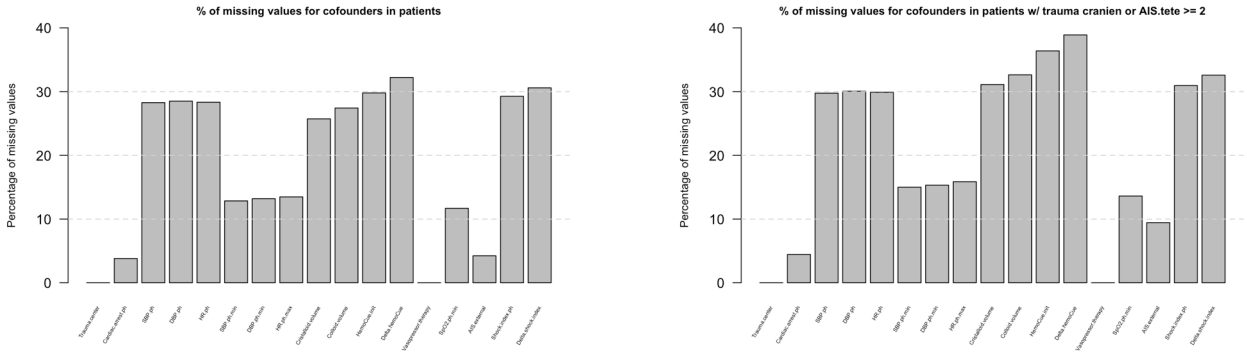

Figure 6: Proportion of missing data in the Traumabase for the 17 cofounders

To solve this issue, we imputed missing values with the R package `missMDA`, and specifically, with the `imputeFAMD` method. This method has the advantage of optimizing the value of missing data when a Principle Component Analysis (PCA) is realized afterwards.

However we work with mixed data (quantitative and categorical variables) and most matching methods can't handle non-binary categorical variables. For this reason, we have to carry out a PCA, to project qualitative variables into a new coordinate database. The `imputeFAMD` method was therefore the best one to pre-process our dataset. The PCA algorithm returns a new dataset with only quantitative variables, perfectly suitable for matching methods.

This pre-processing is applied to all patients to obtain the most accurate results possible. We then selected patients who suffered from a head injury to conduct our matching and our analysis on the effect of the treatment. We also wanted to study smaller groups of patients, according to the intensity or the type of trauma they suffered from. However, these groups contains at most two treated patients. Therefore, we do not have enough patients to match to draw relevant conclusions from these groups.

4.2 Implementation of Matching Methods on the Traumabase

We then implemented the different algorithms on the Traumabase. The results found are given below. The confidence intervals were constructed using bootstrapping algorithms.

Methods	Estimated ATT	IC 95%	Nb dropped
Propensity Logistic Reg	0.06	[0.0597, 0.0995]	53
Propensity Random Forest	0.09	[0.0969, 0.1664]	39
Propensity Multinomial	0.08	[0.0590, 0.1016]	53
CEM	0.12	[-0.0277, 0.4400]	3528
Cardinality Matching	0.10	[0.0208, 0.16201]	7571

Table 3: ATT Estimation on the Traumabase

To analyze the results obtained on the trauma database using our tests on the simulated datasets, it is necessary to estimate the dataset complexity. First, since we have just performed a PCA, our 26 covariates are independent from one another. Moreover, as we can see below, the difference in means between covariates is relatively small (<0.5 for 24 out of 26). However, we also notice with the graph on the right that the difference in variance ratios is more significant. So we have a mid dataset complexity.

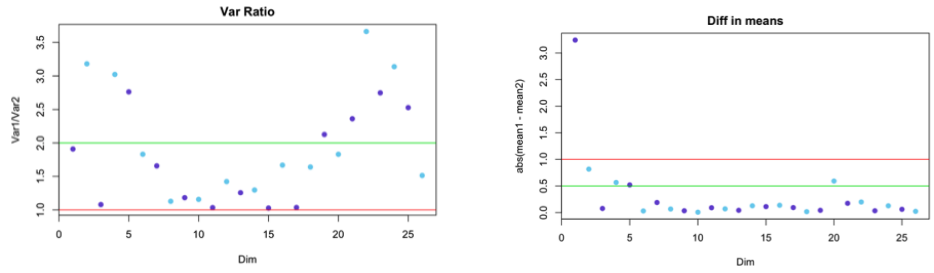

Figure 7: Variance ratio and difference in means for each covariate

We will then detail what we can deduce from these matching methods on the traumabase starting with propensity matching.

4.2.1 Propensity with logistic regression

We find an ATT=0,06 with this method. From the distributional Balance diagram for Y, we can see that before matching, among the ones who had survive from traumatic head injury, most had taken the tramexamic acid. After matching, there is always more people among the surviving patients who had taken the drug but the difference is really lower. **It is thus clear that the treatment was not given randomly**. It can be assumed that patients treated with this drug were more likely to survive even before taking the drug. This seems consistent with the reading of the *CRASH-3* paper [4], which states that this drug is given only to patients with brain injury for less than 3 hours.

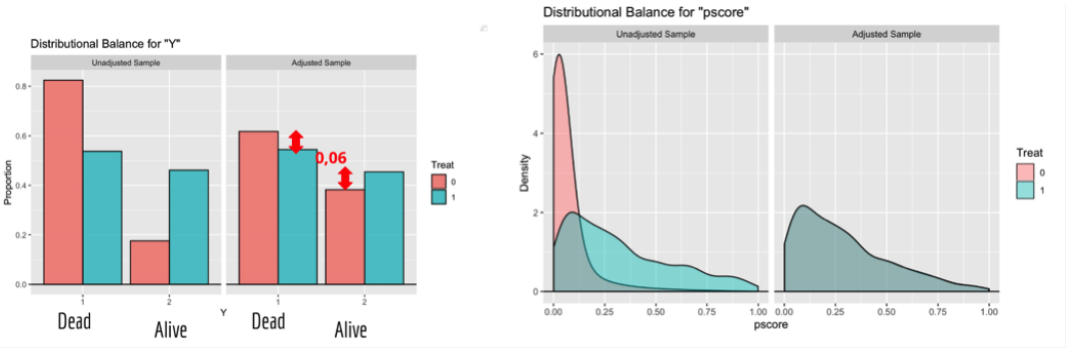

Figure 8: Distributional balance for Y and for the propensity score before and after matching

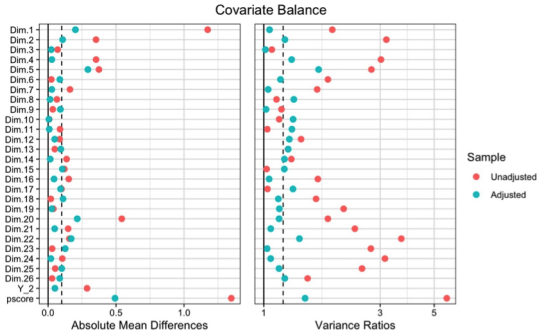

Figure 9: Covariates balance and variance ratios before and after matching

We notice from the two graphs above that the matching on the propensity score is almost perfect and that the individuals matched have similar characteristics (difference in means, variance ratio in blue) and much more similar than before the matching (blue dots before pink dots).

In addition, even if the covariate effect complexity is not linear (which is the case), the quality of the matching will be good, as shown by the simulations before.

4.2.2 With random forest

We also tried to estimate the propensity score using random forests.

Compared to the logistic regression estimation model using the true propensity score model, Random Forests had an additional advantage in producing unbiased estimated standard error and correct statistical inference of the average treatment effect.

We got these graphs below:

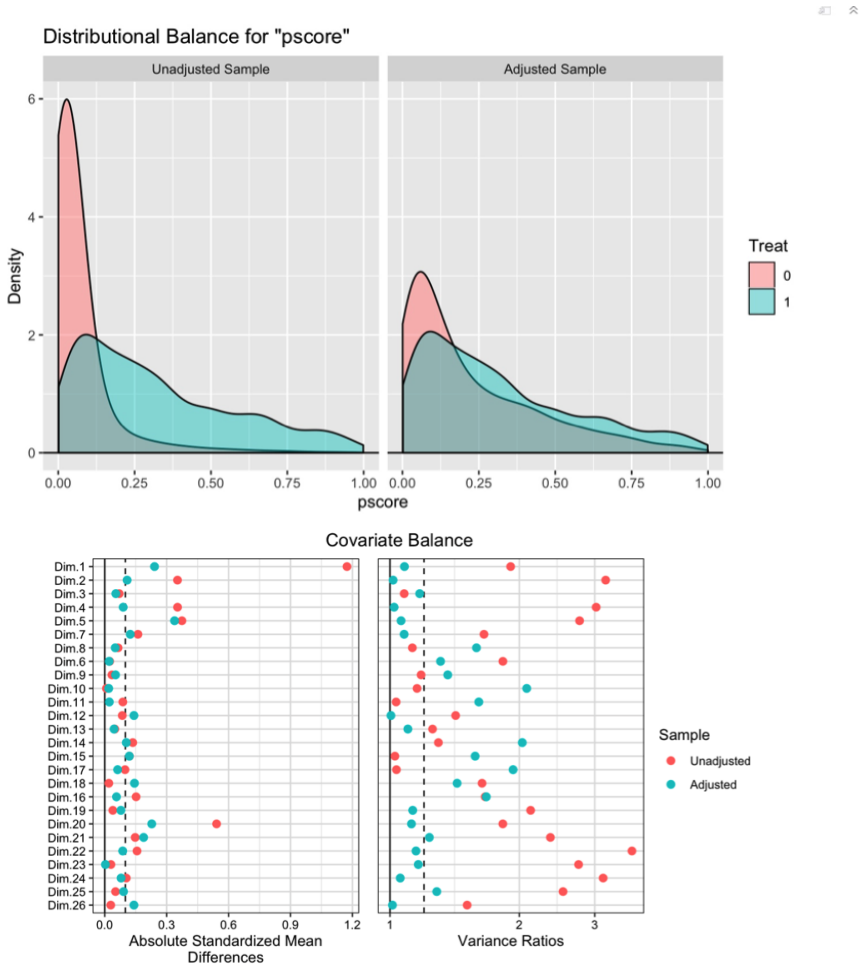

Figure 10: Covariate balance and distributional balance for Y

We notice here that the matching on the propensity score is much less accurate than previously. Moreover, the covariates after matching are still not very well balanced. Thus, we can deduce that the method with random forests is less precise than with a simple logistic regression.

4.2.3 General conclusion

In the light of our study on the traumabase underlying complexity, we can analyze our results. We have seen in the previous section that propensity score and cardinality work best on low to mid complexity dataset. Furthermore CEM

paradoxically works best on complex dataset and is less acurrate on simple dataset.
We observe that the results gave by propensity score and cardinality matching are rather close while CEM gives a wider interval. It is coherent with our previous analysis and we decide not to take CEM results into account.
Both propensity and cardinality matching give us a ATT between 0.02 and 0.16.

5 Robustness to Missing Data

After all this study done on different matching methods, a point still has to be examine. For now, we compared our methods with simulated datasets, that do not have any missing values. However, we have seen previously that the Traumabase we wanted to study contains lots of missing values (sometimes, almost 40% for some covariates). Therefore, it is worthwhile to analyse the effect of missing values and their imputation on matching.

5.1 Different methods to generate missing values

To study the effect of missing values on matching method, we first wanted to create an incomplete dataset. To do so, we used the previous algorithm to simulate datasets of variable complexity, to which we deleted some values. Different methods can be used to remove these values, and we examined two of them : Missing Completely At Random (MCAR) and Missing At Random (MAR).

With the first method, Missing Completely At Random, the probability that an observation is missing is independent of the variables and observations, which means that values are removed completely randomly among all the dataset:

$$\mathbb{P}_R(R|X^{obs}, X^{mis}; \phi) = \mathbb{P}_R(R), \quad \forall \phi.$$

where ϕ configures the missingness distribution $\mathbb{P}_R$.

When applying this method on a dataset simulated with `dataset_3` and `f3`, we obtain the following result. Values in red are the ones missing.

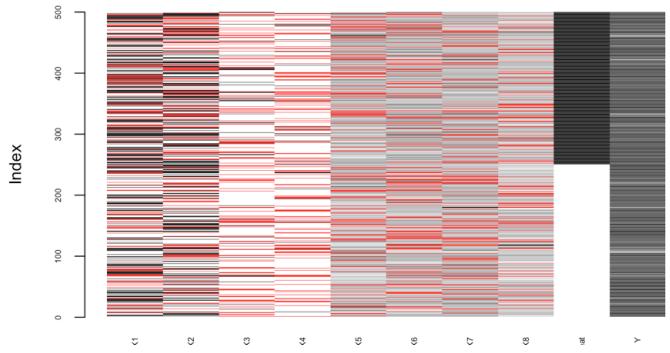

Figure 11: Dataset obtained with the MCAR method

As for the second method to generate missing values, Missing At Random (MAR), the probability that an observation is missing only depends on the observed data X^{obs}.

$$\mathbb{P}_R(R|X^{obs}, X^{mis}; \phi) = \mathbb{P}_R(R|X^{obs}; \phi), \quad \forall \phi, \forall X^{mis}.$$

where ϕ configures the missingness distribution $\mathbb{P}_R$.

When applying this method on a dataset simulated with `dataset_3` and `f3`, we obtain the result below.

We now want to study the impact of these two mechanisms on matching methods. For this purpose, we simulated datasets with missing values generated by one of the two approaches, and we used the same imputation method (here, *imputeFAMD* from the package MissMDA) to complete datasets. We then considered the variance of datasets, and we

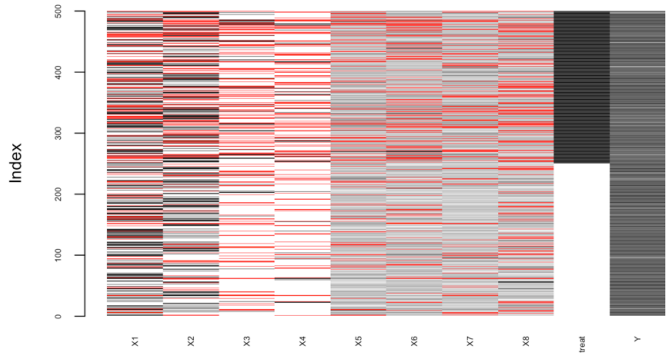

Figure 12: Dataset obtained with the MAR method

compared them to before matching, and to a situation without missing values initially. This enabled us to isolate the effect of removing values on the variance after matching.

We noticed then that the MCAR method tends to increase more the variance before matching. But both methods have similar results on the whole. While the MCAR mechanism seems to have fewer effects with the propensity score using logistic regression method, the MAR one has fewer effects with the propensity score using Random Forest method.

Models		Prop logit			Prop RF		
		No missing values	MAR	MCAR	No missing values	MAR	MCAR
Dataset 3	f3	10.95	17.1	9.36	9.7	2.96	16.46
	f2	1.32	1.66	1.24	1.38	2.57	1.28
	f1	0.21	0.55	0.17	0.51	0.35	0.49
Dataset 2	f3	19.26	10.49	17.46	27.9	7.42	18.91
	f2	1.48	1.42	3.96	0.91	3.19	3.26
	f1	0.26	0.23	0.43	0.3	0.56	0.29
Dataset 1	f3	9.92	12.35	17.18	3.58	4.44	3.66
	f2	0.22	2.62	2.55	0.93	2	1.21
	f1	0.28	0.3	0.23	0.48	0.46	0.16

Table 4: Variance of datasets obtained after matching according to MCAR or MAR methods

5.2 Different methods to impute missing values

We then went on to study four methods to impute missing values : deletion, mean, `Amelia` and `imputeFAMD`. The first one, deletion, consists in removing all rows in our database where some values are missing. This mechanisms leads to a complete database, but many meaningful values are lost, which significantly increases the variance within the dataset. In the case of the traumabase, as some variables have almost 40% of missing values, we can't consider deleting so much data.

The second method is to compute the mean of non null values of a variable, and to assign this mean value to missing values of the same column. Thus, the mean of all variables of the database is preserved and it does not have notable negative effect afterwards.

$$\text{if } X_{ij} = \text{NA}, \quad \text{then } X'_{ij} = \frac{\sum_{i=1}^{n} \mathbf{1}_{\{X_{ij} \neq \text{NA}\}} X_{ij}}{\sum_{i=1}^{n} \mathbf{1}_{\{X_{ij} \neq \text{NA}\}}}$$

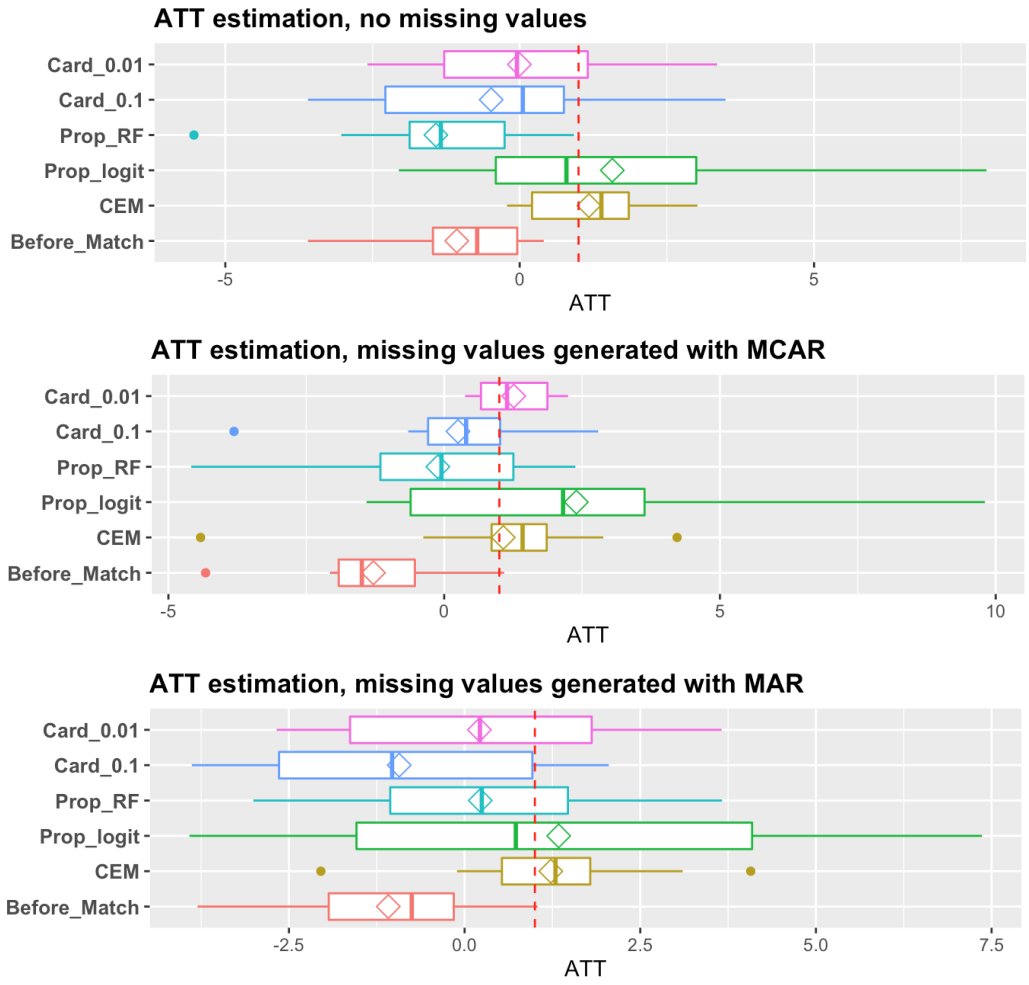

Figure 13: Comparison of methods to generate missing values

Then, Amelia is a very common used method to impute missing values. It uses a technique called *multiple imputation* and it relies on a bootstrapping and Expectation-Maximization (EM) algorithm. Indeed, it creates a bootstrapped version of the original data, estimates the sufficient statistics by EM on bootstrapped samples and then imputes the missing values of the original data using estimated sufficient statistics. It repeats this process m times to produce the m complete datasets where the observed values are the same and the unobserved values are drawn from their posterior distributions.

To compute the best value for missing ones, let $\bar{q}$ denote the average of the m separate estimates q_j $(j = 1, ..., m)$:

$$\bar{q} = \frac{1}{m} \sum_{j=1}^{m} q_j.$$

If $SE(q_j)^2$ is the estimated variance of q_j from the dataset j and $S_q^2 = \frac{\sum_{j=1}^{m} (q_j - \bar{q})^2}{m-1}$ is the sample variance across the m point estimates, we want to minimize the standard error of the multiple imputation point estimate :

$$SE(q)^2 = \frac{1}{m} \sum_{j=1}^{m} SE(q_j)^2 + S_q^2(1 + 1/m).$$

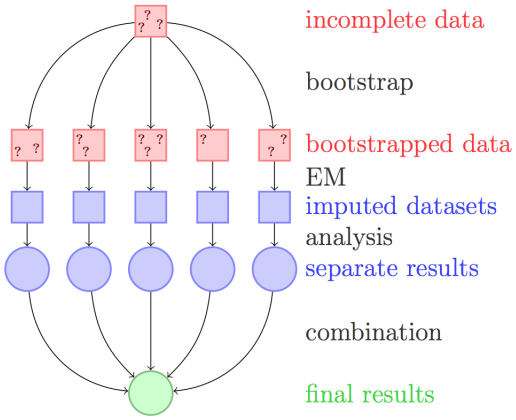

Figure 14: Explanatory drawing of the principle of bootstrapping and EM of the Amelia method

Even if Amelia is the most widespread method to impute missing values, we did not found it to be the best one with the matching methods we used. It has a larger effect on the variance of datasets, which affects the quality of the estimate.

Finally, we studied the imputeFAMD method from `missMDA` package. This method uses an iterative Factorial Analysis for Mixed Data (FAMD) algorithm to impute missing entries, which enables it to process both quantitative and categorical variables. Indeed, the latter are coded using an indicator matrix of dummy variables. Then, the algorithm assigns to missing values the mean of the variable for the continuous variables and the proportion of the category for each category using the non-missing entries. It then performs FAMD on the complete dataset and re-estimates the parameters to improve Principal Analysis Component result. The algorithm iterates these steps of estimation and imputation of missing values until convergence. Therefore, with this method, we obtain a dataset that allows us to have the best results with the FAMD algorithm after.

To compare these four previous methods, we worked on the dataset 3 built with the function f3, and we generated missing values with the MAR mechanism. The table below represents the variance of datasets obtained after generating missing values, imputing these missing values and matching individuals. We then considered the variance of datasets, and we compared them to before matching, and to a situation without missing values initially. This enabled us to isolate the effect of imputation methods on the variance after matching.

	No missing values	Delete	Mean	Amelia	ImputeFAMD
Before matching	9.01	25.71	4.36	3.35	3.08
CEM	2.91	N/A	3.02	4.01	0.57
Prop logit	10.95	55.93	4.56	7.65	9.36
Prop RF	9.7	37.51	8.73	5.4	2.96
Card 0.1	4.64	38.72	2.64	4.54	3.62
Card 0.01	5.67	35.15	2.24	6.81	2.79

Table 5: Variance of datasets obtained after matching, according to imputation methods

As our Traumabase gathers quantitative and qualitative values, we need to carry out a FAMD. For this reason, the best method to use to fill missing values in the Traumabase is the imputeFAMD one. We then conduct our researches about the impact of MCAR and MAR methods, by using this method of imputation for missing values.

5.3 Most robust methods to missing values

Once the previous study done, we decided to compare matching methods on datasets, after generating missing values with the MAR method and imputing them with the `imputeFAMD` method. We then compared the variance of datasets

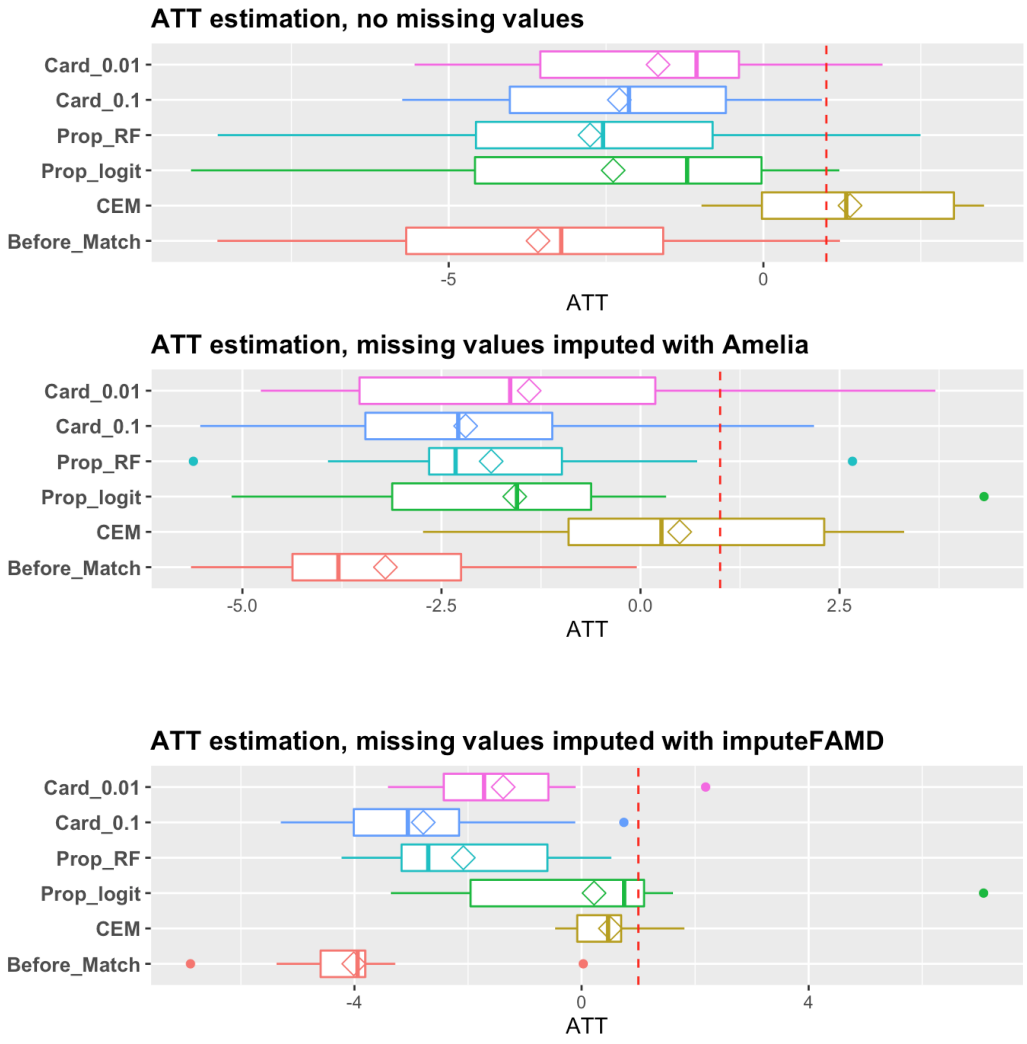

Figure 15: Comparison of methods to impute missing values

after matching for all methods. We noticed that the Coarsened Exact Matching and the Cardinality Matching (with a threshold of 0.01 or 0.1) are the two best ones. They tend to increase less the variance of the datasets.

		Before matching	CEM	Prop logit	Prop RF	Card 0.1	Card 0.01
Dataset 3	f3	3.08	0.57	9.36	2.96	3.62	2.79
	f2	0.59	4.82	1.24	1.28	1.27	1.27
	f1	0.36	0.26	0.17	0.49	0.13	0.17
Dataset 2	f3	13.02	2.91	17.46	18.91	19.38	19.78
	f2	1.2	2.96	3.96	3.26	2.13	2.44
	f1	0.37	1.05	0.43	0.29	0.21	0.19
Dataset 1	f3	2.61	2.82	17.18	3.66	4.45	4.57
	f2	0.94	2.56	2.55	1.21	0.68	0.62
	f1	0.18	0.9	0.23	0.16	0.1	0.03

Table 6: Variance of datasets obtained after matching according to matching methods

Conclusion

The study we led came to several conclusions. First we evaluated the efficiency of our three matching methods in various contexts. We proved that propensity score matching was the most relevant methods when dealing with a simple or mid complexity dataset. Cardinality matching is a bit more robust to complexity than propensity score matching. It works well on mid size datasets. In the case of a more complex dataset coarsened exact matching appears to be very interesting. When it comes to dealing with missing values, coarsened exact matching and cardinality matching must be preferred.
Then we assessed the effect of tranexamic acid. The propensity score matching methods, which are the most relevant in the traumabase frame, gives us a treatment effect on survival rate between 0.06 and 0.16. It means that on average, on ten head injuries victims, one more would survive if they were all administrated tranexamic acid. These results are coherent with the one found in 2019 by the Crash-3 study [4].

References

[1] P.R Rosenbaum and D.B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70:41–55, 1983.

[2] P.C. Austin. Optimal caliper widths for propensity-score matching when estimating difference in means and difference in proportions in observational studies. *Pharmaceutical statistics*, 10(2):150-161, 2011.

[3] M. Resa and J.R. Zubizarreta. Evaluation of subset matching methods and forms of covariate balance. *Statistics in Medicine*, 2016.

[4] Crash 3 trial collaborators. Effects tranexamic acid on death, disability, vascular occlusive events and other morbidities in patients with acute traumatic brain injury: a randomised placebo-controlled trial. *OA*, 2019.

[5] Teresa Alves de Sousa and Imke Mayer. How to simulate missing values? *https://rmisstastic.netlify.com/how-to/generate/misssimul*, 8 August 2019.

6 Annex

6.1 Comparison of Matching Methods

Drop Ratio	Dataset 1			Dataset 2			Dataset 3		
	Linear	Additive	Interrac.	Linear	Additive	Interrac.	Linear	Additive	Interrac.
Before matching	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CEM	0.14	0.12	0.13	0.13	0.13	0.15	0.13	0.12	0.14
Prop logit	0.02	0.02	0.03	0.03	0.02	0.03	0.03	0.02	0.03
Prop RF	0.01	0.00	0.01	001	0.00	0.00	0.02	0.00	0.01
Card 0.1	0.41	0.41	0.42	0.41	0.42	0.42	0.42	0.41	0.42
Card 0.01	0.40	0.40	0.40	0.41	0.41	0.41	0.40	0.41	0.41

Table 7: Drop Ratio Estimation

Var. Ratio		Dataset 1			Dataset 2			Dataset 3		
		Linear	Additive	Interrac.	Linear	Additive	Interrac.	Linear	Additive	Interrac.
X1	Before matching	1.08	1.04	1.04	1.04	1.03	1.04	1.04	1.06	1.03
	CEM	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.01	1.00
	Prop logit	1.00	0.55	1.01	1.01	1.00	1.00	1.01	1.00	1.01
	Prop RF	1.03	1.02	1.02	1.01	1.01	1.02	1.03	1.02	1.03
	Card 0.1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Card 0.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X3	Before matching	2.11	17.1	2.36	2.6	2.34	2.05	2.48	1.76	1.82
	CEM	1.03	1.01	1.02	1.02	1.01	1.01	1.02	1.02	1.01
	Prop logit	1.27	1.09	0.97	1.13	0.96	0.97	0.81	1.14	0.85
	Prop RF	1.44	1.45	1.61	1.69	2.37	1.68	1.50	1.63	1.85
	Card 0.1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Card 0.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X5	Before matching	1.01	1.02	1.04	1.49	1.52	1.44	1.19	1.33	1.09
	CEM	1.05	1.08	1.02	0.98	1.15	1.20	0.87	1.04	0.94
	Prop logit	1.10	1.06	1.09	1.44	1.63	1.26	1.17	1.29	1.24
	Prop RF	1.12	0.95	1.06	1.38	1.40	1.27	1.17	1.31	1.02
	Card 0.1	1.01	0.97	1.03	1.49	1.51	1.41	1.24	1.30	1.11
	Card 0.01	0.99	0.98	1.09	1.52	1.50	1.41	1.21	1.30	1.13
X7	Before matching	0.93	1.01	1.08	1.47	1.44	1.51	1.10	1.28	1.28
	CEM	0.94	1.06	1.05	1.23	1.04	0.99	1.10	1.00	1.16
	Prop logit	0.91	1.07	1.01	1.47	1.42	1.63	1.09	1.28	1.34
	Prop RF	0.94	0.88	1.14	1.29	1.24	1.44	1.10	1.20	1.19
	Card 0.1	0.93	1.00	1.12	1.46	1.47	1.52	1.09	1.28	1.27
	Card 0.01	9.93	1.03	1.14	1.45	1.49	1.52	1.07	1.29	1.25

Table 8: Var. Ratio

K-S distance		Dataset 1			Dataset 2			Dataset 3		
		Linear	Additive	Interrac.	Linear	Additive	Interrac.	Linear	Additive	Interrac.
X1	Before matching	0.10	0.10	0.02	0.29	0.05	0.10	0.08	0.01	0.17
	CEM	0.09	0.09	0.03	0.28	0.05	0.10	0.85	0.01	0.18
	Prop logit	0.41	0.62	0.55	0.66	0.62	0.55	0.80	0.38	0.65
	Prop RF	0.18	0.30	0.08	0.37	0.49	0.29	0.42	0.17	0.38
	Card 0.1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Card 0.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X3	Before matching	0.13	0.13	0.34	0.01	0.04	0.06	0.10	0.37	0.03
	CEM	0.13	0.13	0.34	0.01	0.04	0.06	0.09	0.37	0.04
	Prop logit	0.54	0.73	0.74	0.55	0.35	0.44	0.80	0.38	0.40
	Prop RF	0.29	0.23	0.21	0.25	0.25	0.08	0.42	0.25	0.10
	Card 0.1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Card 0.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X5	Before matching	0.37	0.12	0.05	0.04	0.02	0.02	0.03	0.37	0.06
	CEM	0.36	0.12	0.05	0.05	0.01	0.02	0.04	0.37	0.07
	Prop logit	0.22	0.44	0.38	0.17	0.13	0.18	0.05	0.25	0.28
	Prop RF	0.16	0.10	0.33	0.10	0.14	0.12	0.22	0.29	0.09
	Card 0.1	0.57	0.26	0.48	0.16	0.23	0.35	0.28	1.00	0.40
	Card 0.01	0.81	0.89	0.91	0.31	0.47	0.62	0.46	1.00	0.75
X7	Before matching	0.21	0.27	0.17	0.13	0.07	0.04	0.03	0.07	0.06
	CEM	0.22	0.27	0.17	0.12	0.07	0.07	0.61	0.08	0.7
	Prop logit	0.15	0.28	0.34	0.51	0.03	0.16	0.26	0.24	0.28
	Prop RF	0.11	0.19	0.28	0.20	0.14	0.19	0.21	0.32	0.09
	Card 0.1	0.36	0.53	0.43	0.30	0.22	0.23	0.52	0.53	0.75
	Card 0.01	0.60	0.78	0.73	0.36	0.44	0.49	0.99	1.00	1.00

Table 9: K-S Distance