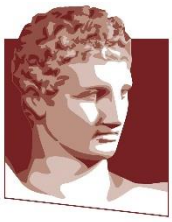
https://doi.org/10.26219/heal.aueb.5509




# ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

## Department of Statistics


## M.Sc. in Statistics

**Master Dissertation**

**«** *Application of Machine Learning methods in baseline covariate adjustment to improve the efficiency of clinical trials* **»**


**Dimitrios Moliotis**

**Supervisor: Prof Vasileios Vasdekis**


**Athens, August 2020**

https://doi.org/10.26219/heal.aueb.5509

# Contents

https://doi.org/10.26219/heal.aueb.5509

https://doi.org/10.26219/heal.aueb.5509

# List of Tables



# List of Figures

https://doi.org/10.26219/heal.aueb.5509

# Acknowledgments

I would like to express my sincere appreciation to each and every one of the tutors of this Master's program, for providing me their valuable guidance, knowledge and support throughout my studies. Especially, I would like to express my sincere gratitude to my supervisor, professor Vasileios Vasdekis, for taking the time to help me, and provide me the right tools in order to successfully complete my dissertation.

https://doi.org/10.26219/heal.aueb.5509

# Abstract

This dissertation thesis is about understanding how to improve the efficiency of a clinical trial study, by performing various estimator augmentation techniques. At first, a few basic notions are discussed regarding the setup and the framework of a typical clinical trial study. Afterwards, we exhibit a few different estimator augmentation techniques, as they appear in the international bibliography. These methods are primarily based on a semiparametric theory. Later in our thesis, we perform an analysis on a real-world example (**ACTG175** study), using these various methods, in order to examine whether there is a difference in the means between the different treatment groups. Concluding, we perform a simulation analysis, to examine the performance of the methods, as well as to answer whether the increase in efficiency is valid, and should we consider it. The augmentation techniques which we are going to present vary between complexity from a simple linear model, to non-parametric machine learning algorithms like Random Forest, and finally we showcase an ensemble method which combines many of the other algorithms, which is based on the **SuperLearner** library of the R Programming Language.

https://doi.org/10.26219/heal.aueb.5509

# Περίληψη

Η διατριβή αυτή έχει σκοπό να δείξει πως μπορούμε να βελτιώσουμε την αποδοτικότητα μίας μελέτης κλινικής δοκιμής. Αρχικά, εισάγουμε κάποιες βασικές έννοιες σχετικά με το γενικότερο πλαίσιο μια τυπικής κλινικής δοκιμής. Έπειτα, παρουσιάζουμε μερικές τεχνικές που χρησιμοποιούνται με σκοπό την βελτίωση ενός εκτιμητή, όπως αυτές φαίνονται στην διεθνή βιβλιογραφία. Οι μέθοδοι αυτές βασίζονται κυρίως στην ημι-παραμετρική θεωρία (semiparametric theory). Στην συνέχεια της διατριβής, πρόκειται να αναλύσουμε δεδομένα τα οποία βασίζονται σε μια πραγματική κλινική δοκιμή (**ACTG175** study), όπου χρησιμοποιώντας αυτές τις μεθόδους θα εξετάσουμε εάν υπάρχουν διαφορές μεταξύ των θεραπειών. Καταλήγοντας, θα διεξάγουμε μία ανάλυση προσομοίωσης, με σκοπό να εξετάσουμε την απόδοση και συμπεριφορά των διαφόρων τεχνικών, καθώς επίσης και να απαντήσουμε στο ερώτημα του εάν υπάρχει βελτίωση στην αποδοτικότητα, την οποία θα πρέπει να λάβουμε υπόψιν. Οι τεχνικές βελτίωσης των εκτιμητών, τις οποίες πρόκειται να παρουσιάσουμε διαφέρουν ως προς την πολυπλοκότητά τους, από τα απλά γραμμικά μοντέλα, σε μη παραμετρικούς αλγορίθμους μηχανικής μάθησης όπως ο Random Forest. Τέλος, θα δείξουμε και μια μέθοδο ensemble, η οποία συνδυάζει πολλές από τις παραπάνω μεθόδους και βασίζεται στη βιβλιοθήκη **SuperLearner** της R γλώσσας προγραμματισμού.

https://doi.org/10.26219/heal.aueb.5509

# 1 Introduction

The primary objective of a typical randomized clinical trial study is to compare two or more different treatments, regarding an outcome of clinical interest. Usually, along with the outcome and the treatment indicator, many baseline covariates may be recorded for each subject. Such covariates include, but are not limited to demographical characteristics, prior medical history, physiological characteristics and more importantly, baseline measurements of the outcome. (Zhang, et al., 2008).

The analysis of such data is primarily based only on the outcome and the treatment indicator. However, there are many examples where the auxiliary baseline covariates have a high association with the outcome. As an example, AIDs Clinical trial group 175 (ACTG175), which is study on randomized HIV – infected patients, and the effect of 4 different treatments, (Hammer, et al., 1996) found that there is a high association between the outcome and the baseline measure of it, as well as the prior antiretroviral history of the subjects and the outcome. In these cases, we can improve the precision and efficiency of such trials, by "adjusting" our estimators based on these relationships (Zhang, et al., 2008).

Regarding the nature of the outcome of interest, there are different ways to identify the treatment effect between the groups. Such methods may include the difference in the means, for continuous response, the Log-odds ratio in the case of binary response. For our analysis and based on the available data, we will focus on continuous response where our main point of interest is to examine the difference in the mean between the treatment groups.

The special case of the difference of the treatment means, is heavily researched among the international bibliography, with authors proposing a great variety of methods for "adjusting" or "augmenting" this estimator. In detail, based on the semiparametric theory, a methodology for augmenting the estimators is presented by (Tsiatis, et al., 2008) (Zhang, et al., 2008) (Yang & Tsiatis, 2001), who showed that while making only a few assumptions, we can use the available data more efficiently.

The augmentation methods which we are going to take into consideration include the ANCOVA model and the use of linear regression models (including stepwise procedures), as presented in (Tsiatis, et al., 2008), the lasso regularization method, similar to (Tian, et al., 2012) and various

https://doi.org/10.26219/heal.aueb.5509

machine learning algorithms, including recursive partitioning, Random Forest and the use of an ensemble method following the super learning principle, considering the indirect approach of (Zhang & Ma, 2019).

In this thesis, we will present some of these methods regarding augmentation techniques in order to improve the efficiency, by utilizing baseline covariates (chapter 2.4). Afterward, we will perform an analysis on a real-world example dataset, using these methods (chapter 3). Finally, we will exhibit the result of a simulation analysis which aims to compare these augmentation techniques (chapter 4) . For the simulation, we will perform a bootstrap -like procedure (Efron & Tibshirani, 1986), where we consider the initial dataset as a population, from which we draw random samples.

# 2 Methodology

In this chapter, we will provide some basic background information, regarding the framework of typical clinical trials, as well as a brief presentation regarding the statistical theories and methods which we are going to use in our analysis.

## 2.1 Introduction to Clinical Trials

Clinical trials are considered one of the best experimental approaches to compare two or more medical treatments. The main goal of a randomized clinical trial, is to assess the effectiveness of an intervention (treatment), as well as to identify possible harms that may arise, as a result of this intervention (Friedman, et al., 2015). A widely accepted definition of a clinical trial as presented by (Friedman, et al., 2015) is "*A prospective study comparing the effects and value of intervention (s) against a control in human beings".* In this context, each study subject – participant must be followed forward in time, from a well-defined point.

In addition, a clinical trial refers to an experiment that it is being performed, in the sense that there must be implemented one or more *intervention* techniques (in contrast to observational studies). Such intervention techniques may be the use of therapeutic - experimental drugs, medical devices, surgical procedures, or approaches, which aim to change an *outcome* (response variable). The essence of the analysis of the clinical trial data, is to compare the effects of these interventions

https://doi.org/10.26219/heal.aueb.5509

with respect to an important outcome variable (response). Depending on the context of the study, as well as the nature of this outcome variable, we can measure the effect of each treatment using various estimators such as a difference in the mean (continuous response variable), an odds ratio (binary response variable), or a difference in survival probability / mean restricted survival time (survival response variable). To estimate such effect measures, there is a big variety of estimators to be considered, among them there are also some empirical counterparts which are being estimated (e.g. difference in the sample mean).

Typically, in a clinical trial study the comparison of the effect measure is perform between two or more different groups of subjects, out of which one is the *control* group and the others are the *treatment* groups on which the intervention is being performed. Usually, the control method regards to the best available method (drug, therapy, procedure) that is the standard method to treat similar cases. In addition, there are many cases where this standard method is not available, or there are ethical complications in the use of it, then the control group receive a of a "*no active"* treatment (Friedman, et al., 2015). This means that the members of the control group will receive a placebo, or in more extreme cases, no treatment at all.

At the starting point of the clinical trial, the groups of the study must be similar to each other. The reason behind this to avoid cases where the differences in the outcomes are attributed to other (possible unknown) reasons, besides the intervention. There are many different techniques in order to best select the control and treatment groups, but something that is common amongst most of them is that there is some way of *randomization* process that it is being executed. Essentially, the randomization process is the way by which each one of the participants of the study, are assigned to the different groups of control and treatment (same chance).

The role that randomization plays in clinical trials has been discussed by a great number of authors including (Friedman, et al., 2015) who have presented a great deal of information in Chapters 5 and 6. The basic idea behind randomization is (a) that the treatments are compared under similar circumstances and (b) randomization allows the use of probability theory to evaluate whether the difference between the groups is likely to be due to change (Armitage, 1982).

Another important aspect of the clinical trials is the number of participants that will participate, i.e. *sample size.* The calculation of the sample size plays an important role in the design of the study. It is a basic step during the planning phase, as without a proper sample size, the study lacks

https://doi.org/10.26219/heal.aueb.5509

the statistical power to detect the effects of the intervention (Friedman, et al., 2015). On the other hand, there is a great deal of studies that do not have large enough samples size. As noted by (Moher, et al., 1994) regarding negative result trials, if a trial has insufficient power, a clinical important effect can be ignored. As a result to this event, there is great number of international bibliography that aims to improve and highlight the importance of both the determination of sample size, as well as the scientifically accurate reporting of it (Moher, et al., 1994). There are greatly available formulas and equations in order to best select the sample size. Regarding the nature of the response variable (continuous, binary, survival time), and the desired power of the experiment to detect the relevant differences, there are many ways for the researchers to estimate the sample size. As an example, a simple formula proposed by (Lachin, 1981) that expresses the basic relationship between the power and the sample size is:

$$\sqrt{N}|\mu_1 - \mu_0| = Z_\alpha \sigma_0 + Z_\beta \sigma_1$$

Where $N$ is the sample size, $\sigma$ is the standard deviation of the individual measurements, $Z_\beta$ is the power of the experiment.

The measures of various important outcome variables at this starting point refers to the *baseline* for each participant.

## 2.2 Baseline Information

In clinical trials, baseline refers as the status of a participant patient, before the start of the intervention (Friedman, et al., 2015). Baseline data measurements can consist of numerical data like the weight of an individual, as well as categorical data like the classification of study subjects into different categories based on various factors such as the presence of a prior drug history. In order to measure the baseline data one can use a great variety of tools like questionnaires, laboratory tests and interviews.

The 4 basic uses of baseline data, as described by (Friedman, et al., 2015) are:

1. Description of trial participants: refers to the need to determine to which population the findings of the clinical trial study apply.

https://doi.org/10.26219/heal.aueb.5509

2. Baseline Comparability: refers to the need to evaluate whether the study groups were comparable before the start of the intervention (i.e. right randomization).
3. Controlling for imbalances in the Analysis: refers to the need to "balance out" prognostic factors that are not controlled in the randomization process.
4. Subgrouping: refers to the need to analyze data on the basis of a specific subgroup that may benefit more / less from the intervention.

Finally, the amount of data which are collected at baseline, depends on the purpose for which the data will be used, as well as the nature of the trial itself (Friedman, et al., 2015).

## 2.3 Framework

In this section we will briefly present the framework of a typical randomized clinical trial, which we will use as a base for the next of this thesis. For the purpose of this thesis, we will consider a typical randomized clinical trial which aims to identify the *treatment effect*, comparing 2 separate groups, as described earlier by *control* and *treatment.* To this context the point of interest is the analysis of a *continuous response outcome*. It is worth mentioning that the methods that we will explore in our thesis, have a wide application amongst other types of response outcomes like binary and survival treatment.

Let denote $Y$ the outcome of the clinical trial, which is continuous and on which we will perform the analysis. Moreover let $T$ denote the treatment indicator where $T = 1$ refers to the experimental intervention treatment (treatment) with probability $\pi$ , and $T = 0$ refers to the control treatment (control) with probability $1 - \pi$. In addition, let $\boldsymbol{X}$ denote a vector of baseline covariates. As we discussed earlier, $\boldsymbol{X}$ may include a baseline measurement of $Y$, as well as additional quantitative and qualitative characteristics, which were records at the starting point of the study, and without any treatment assigned earlier. Note that due to the randomization process $T$ and $\boldsymbol{X}$ are independent of each other, denoted by $T \perp \boldsymbol{X}$. In that respect, for a clinical trial with $n$ participants, we can express the observed data as $(Y_i, T_i, \boldsymbol{X}_i)$ and $i = 1, \dots, n$, independent and identically distributed (denoted as *iid*) across $i$. In addition, let $\theta$ denote the marginal effect measure of $T = 1$ versus $T = 0$, which is the main interest of the analysis of the clinical trial data.

https://doi.org/10.26219/heal.aueb.5509

Under this framework, a common example for $\theta$ based on clinical considerations, is the difference in the group means noted:

$$\theta = \mu_1 - \mu_0 \tag{1}$$

Where $\mu_t = E(Y \mid T = t),\ t = 0, 1.$

The main question that arises regarding our problem, is how we can estimate $\theta$ in a consistent and an efficient way.

An initial thought is to estimate the mean difference:

$$\theta_0 = E(Y \mid T = 1) - E(Y \mid T = 0) \tag{2}$$

Another way to represent this problem is by:

$$E(Y \mid T) = \beta_1 + \beta_2 I(T = 1) \tag{3}$$

Where $\beta_1 = E(Y \mid T = 0)$, and the comparison of the difference in means of each treatment is represented directly by $\beta_2$, and $I(\cdot)$ is the indicator function. Moreover, the treatment effect $\beta_2$ is defined as the *unconditional effect* of treatment relative to control.

A natural choice for an estimator like described above is the simple *unadjusted estimator:*

$$\tilde{\theta} = \bar{Y_1} - \bar{Y_0}$$

$$\tilde{\theta} = \frac{1}{n}\sum_{i=1}^{n} \frac{(T_i - \pi)Y_i}{\pi(1-\pi)} \tag{4}$$

Which is a consistent estimate for $\theta_0$, and $\pi = P(T = 1)$

The standard error of the unadjusted estimator can be estimated as follows:

$$SE_{\hat{\theta}} = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_0^2}{n_0}} \tag{5}$$

Under this scope, the hypothesis testing setting of a typical clinical trial with a continuous outcome is conceptualized as:

$$H_0 : \mu_1 = \mu_0$$

$$H_1 : \mu_1 \neq \mu_0$$

https://doi.org/10.26219/heal.aueb.5509

Note that this is a two-tailed test meaning that the null hypothesis that there is not a significant difference in means between the groups, will be rejected if the difference is either too big, or too small. For this analysis, one can conduct a *two-sample t-test* in order to proceed to the statistical inference of the clinical trial data. It is important to notice that using this method, one aims to estimate the treatment effect using only the responses at the end of the study, meaning that it does not use any other information at baseline.

## 2.4 Covariate Adjustment Methods

The unadjusted estimator method provides an *unbiased* and *consistent* estimate of the treatment effect, although it is a widely accepted method among the international bibliography, many authors have proposed other methods in order to obtain a more *efficient* estimator for $\theta_0$ by adjusting the estimator with the use of baseline covariates.

### 2.4.1 ANCOVA

One common method for covariate adjustment of the treatment effect on continuous response variables in the *Analysis of Covariance* (ANCOVA). Using the ANCOVA method or MANCOVA since we are referring to the *Multivariate Analysis of Covariance* (covariates), we can test whether there are any statistically significant differences between two (or more) groups regarding the dependent variable, after *adjusting* for some covariates (Yang & Tsiatis, 2001). In this case, the effect of the treatment is defined as *conditional* to the population which shares the same values of covariates $\boldsymbol{X}$ :

$$\theta_x = E(Y \mid T = 1, \boldsymbol{X} = \boldsymbol{x}) - E(Y \mid T = 0, \boldsymbol{X} = \boldsymbol{x}) \tag{6}$$

In our case of continuous response $Y$, we can easily estimate $\theta_x$ by postulating a linear regression model:

$$E(Y \mid T, \boldsymbol{X}) = \gamma_0 + \gamma_X^T \boldsymbol{X} + \beta_t T \tag{7}$$

Using the above models, we can obtain the OLS estimator for $\theta = \beta$.

https://doi.org/10.26219/heal.aueb.5509

It is important to note that part of the ANCOVA analysis method is to check carefully whether the assumption are met, since in any other case, the results of the ANCOVA can be misleading and not valid (i.e. bigger variance than expected).

The major assumptions of ANCOVA are

1. The relationship between the dependent variable (response) and the independent variables (covariates) must be linear (*linearity*).
2. All comparison groups have the same variance (*homogeneity of variances*)
3. The residual errors are approximately normally distributed
4. The residual errors have the same variance (*homoscedasticity)*

In case that all the above assumptions are valid, then using the ANCOVA method we can an consistent estimate for $\theta_0$.

In addition, it is important to notice that there are two different types of analysis of covariance models, noted ANCOVA I and ANCOVA II respectively (Yang & Tsiatis, 2001). The main difference of these two models is that in the case of ANCOVA I (one-way) we model the response $Y_{2i}$ on the treatment indicator $T_i$ and the baseline response $Y_{1i}$, while in the case of ANCOVA II (two-way) we model the response on the treatment indicator, the baseline response as well as an interaction term between treatment indicator and baseline response. Regarding these two estimators, (Yang & Tsiatis, 2001) showed that both of these estimator are more efficient when compared to the unadjusted estimator. Moreover, using an extensive simulation analysis, the ANCOVA II was slightly more efficient than ANCOVA I, while the method of *Generalized Estimating Equations* estimator (GEE) was the best of all the others.

### 2.4.2 Augmented Estimator

As we discussed, the ANCOVA method provides a more efficient way to find an estimator for the difference in the means, by utilizing some baseline covariate information. The drawbacks of these methods are that there are some restrictive conditions as we mentioned earlier that have to be met.

Another approach which aims to find an estimator under mild restrictive conditions, have been proposed by a great deal of authors (Tsiatis, et al., 2008) (Zhang, et al., 2008) (Leon, et al., 2003).

https://doi.org/10.26219/heal.aueb.5509

Firstly, we do not make any assumption about the join distribution of $(Y_i, T_i, \boldsymbol{X}_i)$, $i = 1, \dots, n$ such as normality, common variance etc. The only assumption which is the basis for further development is, as we mentioned earlier, that $T$ and $\boldsymbol{X}$ are independent $(T \perp \boldsymbol{X})$. Using the semiparametric theory, we only require that for a *large* sample size $n$, the class of all consistent and asymptotically normal estimators $\hat{\theta}$ can be demonstrated by:

$$\sqrt{n}\left(\hat{\theta} - \theta\right) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \psi(T_i, Y_i) + o_p(1) \tag{8}$$

where $\psi$ is the influence function. One may notice that the influence function $\psi$ does not involve any information about the baseline covariates, which means that it is independent of $\boldsymbol{X}_i$.

For an estimator as defined by (4) the influence function has the following formula:

$$\psi(T, Y) = \frac{T(Y - \mu_1)}{\pi} - \frac{(1 - T)(Y - \mu_0)}{1 - \pi} \tag{9}$$

Bases on the semiparametric theory, (Tsiatis, et al., 2008) and (Tian, et al., 2012) showed than an augmented estimator for $\theta$ can be obtained by:

$$\hat{\theta}(\alpha) = \tilde{\theta} - \frac{1}{n} \sum_{i=1}^{n} (T_i - \pi)\alpha(\boldsymbol{X}_i) \tag{10}$$

Where $\alpha(\boldsymbol{X})$ is any arbitrary function of $\boldsymbol{X}$ for which $E\{\alpha(\boldsymbol{X})\}^2 < \infty$.

In this context, one can view the $\alpha(\boldsymbol{X})$ as *augmenting* the simple unadjusted estimator. Due to the randomization process $(T \perp \boldsymbol{X})$, (Tsiatis, et al., 2008) showed in the Appendix, that the augmentation term converges in probability to zero. As a result, the augmented estimator $\hat{\theta}(\alpha)$ is consistent for $\theta$ for any function $\alpha(\boldsymbol{X})$.

In addition, when $\alpha^{(0)}(\boldsymbol{X}_i) = \alpha^{(1)}(\boldsymbol{X}_i) \equiv 0$, then the augmented estimator in (10) reduces to the simple unadjusted estimator $\tilde{\theta}$, which is the sample difference as we discussed earlier.

Furthermore, (Tsiatis, et al., 2008) showed that the ANCOVA I estimator can be expressed asymptotically to an estimator of the above form (10), which means that we can view that the

https://doi.org/10.26219/heal.aueb.5509

ANCOVA I estimator is consistent and asymptotically normal for $\theta$, without the need of restrictive conditions. On the same time, this result holds even if the model of ANCOVA (7) is not the correct representation of the true regression relationship. They demonstrated that ANCOVA I estimator is asymptotically equivalent to the form (10) when:

$$\alpha^{(0)}(\boldsymbol{X}_i) = \alpha^{(1)}(\boldsymbol{X}_i) = \sum_{XY}^{T} \sum_{XX}^{-1} \boldsymbol{X}_i \qquad (11)$$

Where $\sum_{XY}$ is the covariance between X and Y and $\sum_{XX}$ is the covariance matrix of *X*.

Correspondingly to these results, the also showed that the ANCOVA II estimator can be viewed asymptotically as in the form of (10) and when the $\pi = 0.5$ both of the ANCOVA estimator are equivalent. In any other case, the ANCOVA II estimator has a smaller asymptotical variance.

### 2.4.3 Defining Optimal Estimator

Having defined the form of an augmented estimator, the next step is to find an *optimal* augmented estimator, where the optimality can be expressed as the estimator with the *smaller variance*.

As showed by (Leon, et al., 2003), among all the estimators which are equal or asymptotically equivalent to an expression like (10), the estimator with the smallest variance can be obtained when $\alpha(\boldsymbol{X}_i)$ is assumed to be a linear function like:

$$\alpha^{(t)}(\boldsymbol{X}_i) = E(Y_i \mid T_i = t, \boldsymbol{X}_i), \quad t = 0,1 \qquad (12)$$

In this context, the *optimal* estimator can be derived from the true regression relationship of $Y$ on $\boldsymbol{X}$ for each treatment. This means that one can consider different functions of $\boldsymbol{X}$ for each treatment, respectively. This result suggests that our goal is to construct separate models for the $E(Y \mid T = t, \boldsymbol{X}),\ t = 0,1$ that describe the best possible true regression relationship. If the models do represent the true mean relationship for each treatment, then the resulting estimator will have the smallest asymptotic variance and will improve the simple unadjusted estimator. On the other hand, if the models are not correctly specified, then we will have larger variance than the *optimal*, but our estimators will also be consistent and asymptotically normal. As (Tsiatis, et al., 2008) noted, all estimators that are asymptotically equivalent to an expression in the form of (10) are

https://doi.org/10.26219/heal.aueb.5509

members of the *semiparametric* class, because we have no assumption regarding the distribution of $Y$, yet they are consistent and asymptotically normal.

In order to find the models that best describe the *true* relationship of $Y$ and $\boldsymbol{X}$, we can use various model selection procedures such as stepwise selection as used by (Tsiatis, et al., 2008) or penalized methods (lasso) as used by (Tian, et al., 2012).

### 2.4.4 Learning Optimal Estimator

Having defined the form of the *optimal* estimator, a logical question that arises is how can we estimate the optimal function $a_{opt}$. As (Zhang & Ma, 2019) pointed, in reality, "the optimal function is unknown and must be estimated from the data". As a result, there is great flexibility regarding the use of modeling methods to best obtain an efficient estimator for $\theta$. Under some mild regularity conditions, (Zhang & Ma, 2019) showed that:

$$\sqrt{n}\left(\hat{\theta}(\hat{a}) - \theta\right) = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\{\psi(T_i, Y_i) - (T_i - \pi)\,\alpha^*(\boldsymbol{X}_i)\} + o_p(1) \qquad (13)$$

Where $\hat{a}$ is a generic estimator with probability limit $\alpha^*$, although $\alpha^*$ is not equal to $a_{opt}$. Based on this asymptotic result, one can find different ways to estimate $\hat{a}$. A straightforward approach that was used by (Zhang & Ma, 2019), is to apply different machine learning algorithms in order to find the best $\hat{a}$.

Following the expression of (12), (Zhang & Ma, 2019) suggested that we can obtain an estimate of $a_{opt}$, by using the regression:

$$a(\boldsymbol{X}) = \eta(1, \boldsymbol{x}) - \eta(0, \boldsymbol{x}) \qquad (14)$$

where: $\eta(t, \boldsymbol{x}) = E\{\psi(T, Y) | \, T = t, \boldsymbol{X} = \boldsymbol{x}\}$.

Due to the fact that this regression function is complicated as the $\psi(T, Y)$ is not fully observed and depends on unknown parameters, as for example in (9), where there are involved the unknown parameters $\mu_1$ and $\mu_0$. One possible solution to this problem, is to obtain an estimate $\hat{\psi}(T, Y)$ by using the empirical estimates of $\mu_1$ and $\mu_0$, as demonstrated in (Tian, et al., 2012). Then we can

https://doi.org/10.26219/heal.aueb.5509

use any machine learning algorithm to obtain an estimate of $\eta(t, \boldsymbol{x})$, which we will use to form the augmented estimator. In order to estimate $\hat{\eta}(t, \boldsymbol{x})$ we can use various machine learning algorithms treating $\hat{\psi}(T, Y)$ as response. Once we have obtained $\hat{\eta}(t, \boldsymbol{x})$, the corresponding estimate of the augmentation term can be estimated by:

$$\hat{a}(\boldsymbol{X}) = \hat{\eta}(1, \boldsymbol{x}) - \hat{\eta}(0, \boldsymbol{x}) \tag{15}$$

This method was characterized as an *indirect* method (Zhang & Ma, 2019), since we do not know in reality if this method of obtaining the estimator $\hat{\theta}(\hat{a})$ will attain the minimum asymptotic variance of all the estimator of this form. As an alternative, they proposed the use of a more *direct* method, where in order to estimate efficiently $\theta$, the best way is to minimize the asymptotic variance of $\hat{\theta}(\alpha)$

$$E\{\psi(T, Y) - (T - \pi)\, \alpha(\boldsymbol{X})\}^2 \tag{16}$$

Which means, we need to choose an $\hat{a}$ to minimize the sum of squares of the sample version of the above form, as follows:

$$\sum_{i=1}^{n} \left\{\hat{\psi}(T_i, Y_i) - (T_i - \pi)\, \alpha(\boldsymbol{X}_i)\right\}^2 \tag{17}$$

Another similar example has been presented by (Tian, et al., 2012) where $\alpha$ is assumed to be linear with the *lasso* penalty. One important advantage of this direct approach is that it is better aligned with the objective of estimating $\theta$ efficiently. As pointed by (Zhang & Ma, 2019), in the direct approach, if (17) is minimized over a class of function that include $\alpha(\boldsymbol{x}) \equiv 0$ , then the $\hat{\theta}(\hat{\alpha}_{dir})$ is at least as efficient as the unadjusted estimator.

Finally, various authors, including (Tian, et al., 2012) and (Zhang & Ma, 2019), they have highlighted that when the estimator of interest $(\theta_0)$ is the mean difference, we don't need to estimate the influence function, and we could estimate $E(Y|T_i, Y_i)$ using a machine learning algorithm and substitute the estimate into (9).

https://doi.org/10.26219/heal.aueb.5509

### 2.4.5 Super-learning principle

In order to take the augmentation method a step further, (Zhang & Ma, 2019) proposed the use of an *ensemble method* algorithm. The basic idea behind ensemble learning is to build a prediction model by combining the strengths of a collection of simpler base models (Hastie, et al., 2009). Since we do not know beforehand which machine learning algorithm will perform better, we can combine all the candidate methods into a super learner. (Mark, et al., 2007). In essence, the super learner algorithm combines various base learners, and by using a V-Fold Cross Validation procedure to find the minimum risk predictor, it assigns weights to each base learner.

The three basic components of a super learning algorithm are:

1. The elements in a library:
2. The loss function
3. The function to combine the predictors

As the authors of this algorithm suggested, we need to think about the specific prediction problem when we choose the library. If most of the learners are poor in performance, so will the Super Learner also be. Moreover, for out specific case, since our main interest will be the difference in the mean of a continuous response, a natural choice is the squared error loss.

### 2.4.6 Variance Estimation

For an estimator that satisfies (13), the asymptotic variance can be consistently estimated by:

$$\frac{1}{n}\sum_{i=1}^{n}\left\{\hat{\psi}(T_i, Y_i) - \ (T_i - \pi)\ \hat{\alpha}(\boldsymbol{X}_i)\right\}^2 \qquad (18)$$

Although, (Zhang & Ma, 2019) highlighted that the above variance estimator may suffer from a downward resubstitution bias, since the dataset which is used to fit the predictive model, is also used in assessing the performance of this model. In order to cope with this problem, they proposed the use of a cross-validation procedure where:

https://doi.org/10.26219/heal.aueb.5509

$$\frac{1}{n}\sum_{i=1}^{n}\left\{\hat{\psi}^{(-j_i)}(T_i, Y_i) - \ (T_i - \pi)\ \hat{\alpha}_k{}^{(-j_i)}(\boldsymbol{X}_i)\right\}^2 \qquad (19)$$

# 3 Application to the ACTG175 dataset

The aim of this chapter is to demonstrate the application of the various methods, which were presented earlier, in a dataset derived from a real clinical trial. First, we will present an exploratory data analysis of the dataset and then we will compare the various methods based on the ACTG175 dataset.

## 3.1 ACTG175 trial study

For the scope of our analysis, we will use a freely available clinical trial dataset, named ACTG175. The dataset can be easily accessed via the R library "`speff2trial`".

The dataset refers to a clinical trial study which aimed to evaluate different treatments, in adults with human immunodeficiency virus type 1 (HIV-1) whose CD4 cell counts were from 200 to 500 per cubic millimeter. In detail, ACTG 175 was to a randomized, double-blind, placebo-controlled trial to compare the different treatments: 600 mg of zidovudine; 600 mg of zidovudine plus 400 mg of didanosine; 600 mg of zidovudine plus 2.25 mg of zalcitabine; or 400 mg of didanosine in HIV-1–infected subjects with CD4 cell counts between 200 and 500 per cubic millimeter. The primary study end point was a >50 percent decline in the CD4 cell count, an event indicating progression to the acquired immunodeficiency syndrome (AIDS), or death (Hammer, et al., 1996).

The results of the study were that treatment with zidovudine plus didanosine, zidovudine plus zalcitabine, or didanosine alone slows the progression of HIV disease, and all of them are superior to treatment with zidovudine alone. On the other hand, it seems that there was not any significant difference between the three treatments. In addition, they found that antiretroviral therapy can improve survival in patients with CD4 cells below 500 per cubic millimeter.

Based on the results of the original study, and following the example of (Tsiatis, et al., 2008), we will consider 2 different treatment groups: zidovudine monotherapy (ZDV) with $n_0 = 532$ patients, and all the other groups combined $n_1 = 1607$ patients ($\pi = 0.75$), on which we will perform our analysis.

https://doi.org/10.26219/heal.aueb.5509

## 3.2 Exploratory Data Analysis of ACTG175

As we have already mentioned, we focus on analysis the differences between the means of the outcome variables. In the case of ACTG175 dataset, the outcome variable is the CD4 cell count per cubic millimeter (cells/mm$^3$) at $20 \pm 5$ weeks (cd420).

Our dataset consists of 2139 patients, out of which *532* were on zidovudine only (arms = 0), *522* on zidovudine and didanosine (arms = 1), *524* on zidovudine and zalcitabine (arms =2) and *561* on didanosine only treatment (arms = 3). Due to fact that previous studies did not find any statistical significance between the 3 treatments, (Hammer, et al., 1996), and following (Tsiatis, et al., 2008) example, we will consider two treatment groups: zidovudine monotherapy (ZDV) with $n_0 = 532$ patients, and all the other groups combined $n_1 = 1607$ patients. Considering the ZDV treatment as the control group, the proportion of patients under the other treatments is ~75%.

In addition to the outcome variable, we have information about the CD4 cell count at the baseline level (cd40). Other continuous baseline covariates that are available from the dataset that we may include in our analysis are CD8 cell count (cells/mm$^3$, *cd80*), age (years, *age*), weight (kg, *wtkg*), Karnofksy [1]Performance Scale Index (scale of 0-100, *karnof*). Regarding indicator variables, there are hemophilia (*hemo*), homosexual activity (*homo*), race (0 = white, 1 = non-white, *race*), history of intravenous drug use (*drugs*), gender (0 = female, 1 = male, *gender*), antiretroviral history (0 = naive, 1 = experienced, *str2*) and symptomatic indicator (0 = asymptomatic, 1 = symptomatic, *symptom*).

Below, we present a table with descriptive statistics of the quantitative covariates, for each treatment group:

[1] The Karnofsky Performance Scale Index allows patients to be classified as to their functional impairment. The lower the Karnofsky score, the worse the survival for most serious illnesses

https://doi.org/10.26219/heal.aueb.5509

**Table 1 Descriptive Statistics of Continuous Baseline Covariates**

| | Control N = 532 | | | | | Treatment N = 1607 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ***Mean*** | ***SD*** | ***Median*** | ***Skew*** | ***Kurtosis*** | ***Mean*** | ***SD*** | ***Median*** | ***Skew*** | ***Kurtosis*** |
| *age* | 35.23 | 8.85 | 34 | 0.71 | 1.19 | 35.26 | 8.66 | 34 | 0.62 | 0.88 |
| *wtkg* | 76.06 | 13.22 | 74.6 | 0.85 | 2.23 | 74.82 | 13.27 | 74.16 | 0.66 | 2.2 |
| *karnof* | 95.43 | 5.98 | 100 | -1.14 | 1.32 | 95.45 | 5.88 | 100 | -0.98 | 0.45 |
| *cd40* | 353.2 | 114.11 | 346 | 0.57 | 0.42 | 349.61 | 120.04 | 339 | 0.81 | 2.16 |
| *cd80* | 987.25 | 475.22 | 881 | 1.83 | 6.11 | 986.42 | 481.98 | 896 | 1.7 | 6.15 |

Since the assign of each patient on the treatment group was random, we see that the covariates are independent from the group, and there are not any visible differences among the groups.

In addition, we present a table with the proportions of each binary covariate.

**Table 2 Proportions of Binary Baseline Covariates per treatment group**

| | Control N = 532 | | Treatment N = 1607 | | ***Total*** | |
|---|---|---|---|---|---|---|
| *Hemotherapy (yes)* | 42 | 7.89% | 138 | 8.59% | 180 | 8.42% |
| *Homosexuality (yes)* | 341 | 64.10% | 1073 | 66.77% | 1414 | 66.11% |
| *Intravenous Drugs (yes)* | 63 | 11.84% | 218 | 13.57% | 281 | 13.14% |
| *Race (non - white)* | 156 | 29.32% | 461 | 28.69% | 617 | 28.85% |
| *Gender (male)* | 432 | 81.20% | 1339 | 83.32% | 1771 | 82.80% |
| *Antiretroviral History (experienced)* | 309 | 58.08% | 944 | 58.74% | 1253 | 58.58% |
| *Symptomatic Status (symptomatic)* | 89 | 16.73% | 281 | 17.49% | 370 | 17.30% |

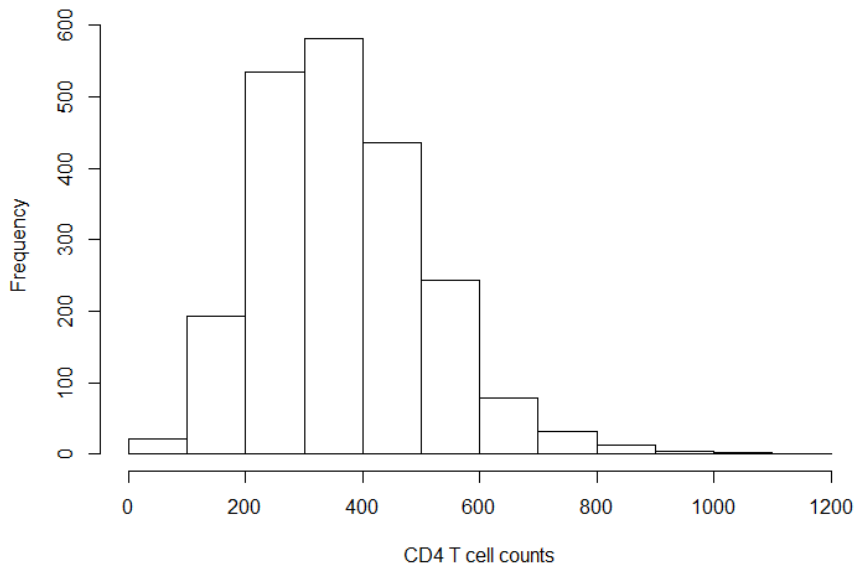
Again, as we expect due to randomization, there is not any significant difference among the groups.

Regarding the response variable CD4 (cd420), upon a visual investigation of the histogram, there are early signs that the data are not normally distributed as we see below (Figure 1):

Based on a quick visual analysis, it seems that the response variable is positive skewed (right skew), while normality tests do reject the normality assumption. (Shapiro's Normality test p-value < 2.2e-16).

https://doi.org/10.26219/heal.aueb.5509

**Figure 1 Histogram of response outcome CD4 cell count at 20 ± 5 weeks**




In addition, we present a table with some summary statistics of the response variable:

**Table 3 Summary Statistics of Response variable outcome CD4 cell count at 20 ± 5 weeks**

| Data | N | Mean | SD | Median | Min | Max | Skew | Kurtosis |
|---|---|---|---|---|---|---|---|---|
| *All* | 2139 | 371.31 | 144.63 | 353 | 49 | 1119 | 0.73 | 1.06 |
| *Treatment* | 1607 | 382.95 | 147.08 | 364 | 52 | 1119 | 0.76 | 1.07 |
| *Control* | 532 | 336.14 | 130.96 | 330.5 | 49 | 909 | 0.49 | 0.46 |

Using the mean estimation of the above table (Table 3), and based on the equation 4, we can estimate the unadjusted estimator as we mentioned earlier as:

$$\tilde{\theta}_{unadjusted} = 46.81$$

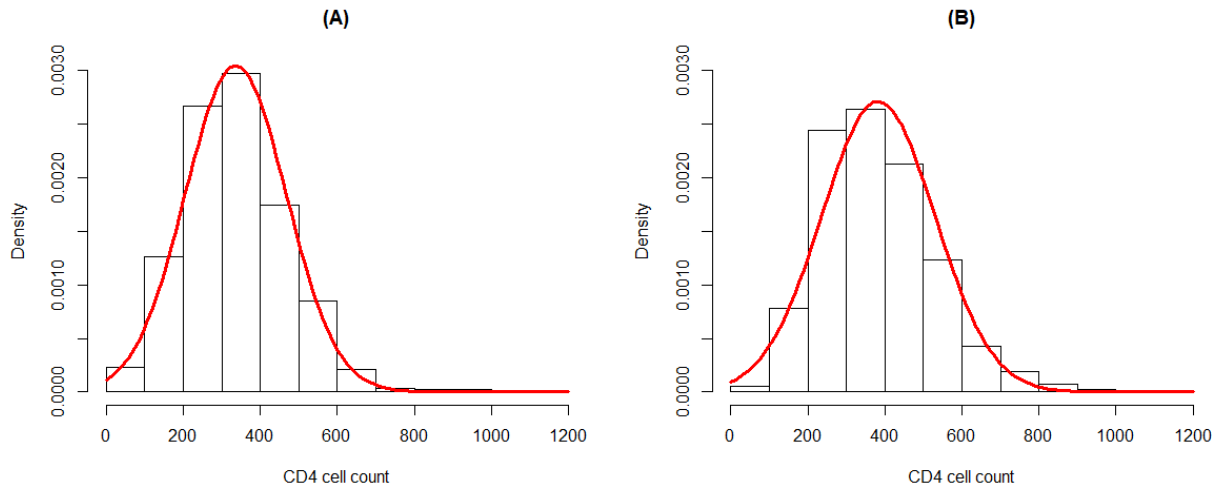
We will investigate this result later in our analysis, as this unadjusted estimator forms the basis on which we will have to compare all the other covariate adjustment methods.

It is important to notice that when CD4 cell count outcomes exhibit severe skewed distributions, they are often analyzed on transformed scales (e.g. Logarithmic). In our case, although we have rejected the normality hypothesis, since the estimators are semiparametric, they are consisted and asymptotically normal regardless of the true distribution of the data. Moreover, following (Tsiatis, et al., 2008) implementation, we will not transform the data in any way, as skewness is not severe,

https://doi.org/10.26219/heal.aueb.5509

while the original scale is also more readily interpretable, as well as consistent with the real-world applications.

**Figure 2 Histograms of CD4 counts at 20 ± 5 weeks: (A) ZDV monotherapy group. (B) All other treatments**




## 3.3 Estimator for ACTG175 dataset

In this subchapter, we will presently briefly each method that we used during our analysis of the ACTG175 dataset, and then we will present and compare the different estimators, as well as their relative efficiency gains, when compared to the simple unadjusted estimator.

### 3.3.1 Unadjusted Estimator

Firstly, we begin our analysis by estimating the simple unadjusted estimator using the equation (4), which is no other than the difference of the sample means:

$$\tilde{\theta}_{unadjusted} = 46.81 \qquad se = 6.76$$

This estimator forms the basis of our analysis, which we will try to improve by leveraging baseline covariate information. Using hypothesis testing based on this simple unadjusted estimator, there are strong evidence of treatment difference. In fact, we reject the null hypothesis that both treatments (control and treatment) are equal. On one hand, without making any other assumptions, one can easily reach to a conclusion, based solely on this estimator. On the other hand, one may

https://doi.org/10.26219/heal.aueb.5509

wish to further improve the efficiency of such clinical trials, where in order to do that, there is a need of take the baseline covariates into account. As we will highlight later in this analysis, there is an actual window for improvement, as there are associations between the covariates and the response, which are not taken into account by the simple unadjusted estimator.

### 3.3.2 ANCOVA Method

The second method in order to estimate the adjusted estimator is to use the ANCOVA method, which we will estimate by postulating a linear model[2], using equation 7. One important distinction is that, because of the covariate adjustment, the effect of the treatment is defined *conditionally* to the population which shares the same covariate characteristics. In addition, we present two different estimators:

$$\tilde{\theta}_{ANCOVA} = 49.69 \qquad se = 5.647$$

for which we used all baseline covariates, and

$$\tilde{\theta}_{ANCOVA\ backward} = 49.76 \qquad se = 5.633$$

for which we modeled all the baseline covariates but proceeded to a stepwise backward selection[3]. If we compare these estimators with the unadjusted one, it is already clear that there is a huge relative gain in efficiency. One explanation is that there is high correlation between CD4 cell counts at the baseline (cd40) and the CD4 cell at 20 ± 5 weeks (correlation coefficient = 0.642 for control and 0.577 for treatment group). This association cannot be captured by the unadjusted estimator, and as a result it exhibits larger standard errors that the other estimators. On the other hand, by taking our analysis a step further, we see that some of the assumptions of the ANCOVA method are not valid. Upon visual investigation of the normal Q-Q plot of the residuals of each model, we see that there is strong evidence that the normality assumption is violated.

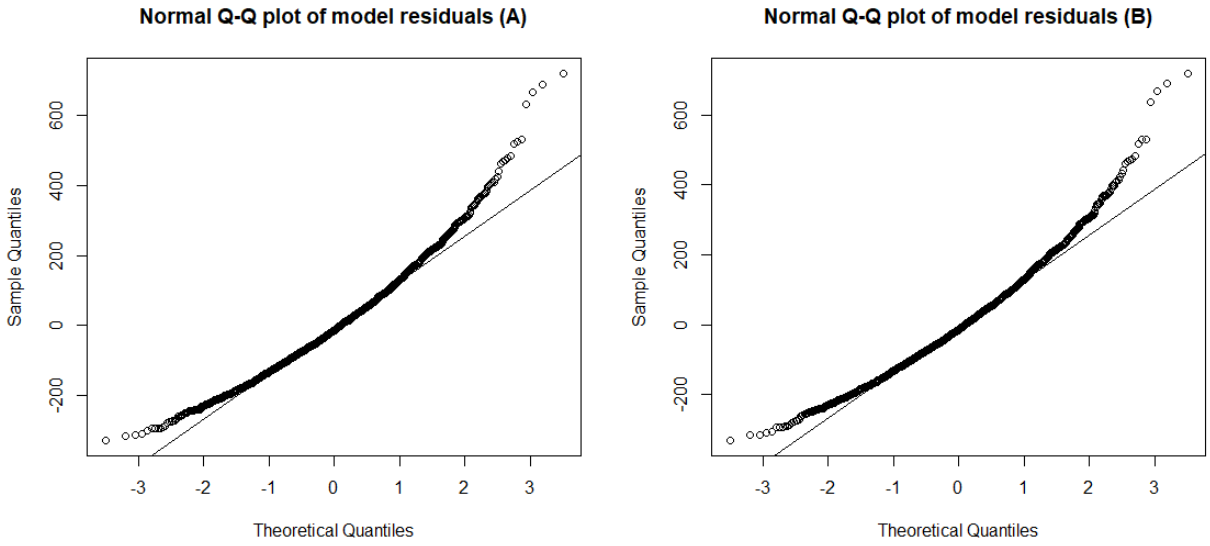
The normal QQ plots suggest that the model residuals are not normally distributed, as there is evidence of fatter tails. This fact is also confirmed by using normality hypothesis test[4], which in

[2] Linear Models were fitted using `lm()` function in R
[3] For more information about the model specifications view appendix
[4] The hypothesis test for normality of residuals was performed using `shapiro.test()` function in R

https://doi.org/10.26219/heal.aueb.5509

this case also rejects the null hypothesis of normality (Shapiro-Wilk normality test p-value < 2.2e-16 for both models).

**Figure 3 Normal QQ plot of residuals, (A) ANCOVA, (B) Backward Step ANCOVA**




Finally, for completeness purposes, we examined whether the assumption of homogeneity of variances between the groups of each model is valid. Again, using Bartlett [5]test of homogeneity of variances, we reject the null, as there is strong evidence that the variances are not equal between the treatment groups (Bartlett test of homogeneity of variances p-value = 0.0006 for ANCOVA and p-value = 0.0005 for Backward ANCOVA model).

### 3.3.3 Augmented Estimator – Linear Models

The next step of baseline covariate adjustment analysis, is to follow (Tsiatis, et al., 2008) implementation, where initially we split the data for each treatment group, and then we fit separate linear models on each group. In order to find a model that is as close as possible to the true, we used mainly stepwise[6] methods. In addition, we fitted models including all terms, including squared terms or two-way interactions of the baseline covariates (Harrell, 2015).

The steps of our analysis were the following:

[5] The hypothesis test for homogeneity of variance was performed using `bartlett.test()` function in `R`
[6] The stepwise methods was performed using `step()` function in `R`.

https://doi.org/10.26219/heal.aueb.5509

1. Follow a separate model for each treatment group.
2. Estimate the predictions of each different on the complete data, to obtain $\alpha(\boldsymbol{X}_i)$.
3. Estimate the augmented estimator using equation 10.

During our analysis, we selected 3 different cases during the model fitting. First, we fitted a separate linear model for each treatment group, where we included all terms ($\tilde{\theta}_{all}$). In the second case, based on the above models, we incorporated stepwise methods, to further improve our estimates ($\tilde{\theta}_{Backward1}$). Finally, we considered separate models, for which we included all terms, all two-way interactions, and the squared terms of the continuous variables $\tilde{\theta}_{Backward2}$). The visual inspection of the previous models suggested that maybe the association is not linear.

It is important to notice that all of the previously mentioned linear models, reject the assumptions of linear OLS regression (i.e. normality assumption of model residuals is rejected). This may not pose a serious problem, as various authors (Tsiatis, et al., 2008) (Zhang, et al., 2008) have already showed that the augmented estimator are asymptotically normal and consistent, even when these models are not correctly specified. In that case, the estimator may have larger variance than the optimal augmentation technique.

Below we present results of the different estimators:

**Table 4 Estimates of θ for ACTG175 data using Linear Models**

| Estimator | Estimate | Standard Error | Relative Efficiency |
|---|---|---|---|
| *Unadjusted* | 46.810 | 6.760 | 1 |
| *LM - All terms* | 49.818 | 5.101 | 1.755 |
| *Backward - 1* | 49.89 | 5.118 | 1.744 |
| *Backward - 2* | 53.647 | 4.896 | 1.906 |

Based on the above estimates, it is easy to see the immediate improvement by incorporating a baseline covariate adjustment. We calculated the relative efficiency of an estimator based as

$$Relative\ Efficiency\ = \frac{(SE\ of\ unadjusted\ estimator)^2}{(SE\ of\ augmented\ estimator)^2}$$

All of the proposed methods perform a lot better than the unadjusted estimator. Since, the interpretation of the model is not of primary concern, in a clinical trial setting, there are clear advantages of fitting a model like *Backward – 2,* where despite its complexity (84 terms before

https://doi.org/10.26219/heal.aueb.5509

stepwise), we have big improvements. Out of all such models, *Backward-2* seems to perform better, as it has the higher relative efficiency (1.906).

### 3.3.4 Augmented Estimator – GLMNET

The next method that we used on order to augment the simple estimator is to consider the two different regularization methods, ridge and lasso procedures (Hastie, et al., 2009). We followed the same approach as for the previous models regarding fitting separate models for each treatment group. In addition, due to the fact that the ridge and lasso procedures[7] tend to produce biased estimators, previous authors (Tian, et al., 2012) suggested the use of a k-Fold Cross Validation method (10 fold and 20 fold). In addition, the loss function of the model fitting is based on Mean Square Error (MSE). For reproducibility purposes, we used the internal R function `set.seed(1)`, in order to use the same cross validation split of the data for each different method. Finally, we fitted[8] all models for both all terms and all terms plus their interactions separately. Although, the full specified models (84 terms) tended to perform worse than the other models, we present their results for completeness.

Below we present results of the different estimators:

**Table 5 Estimates of θ for ACTG175 data using GLMnet**

| Estimator | Estimate | Standard Error | Relative Efficiency |
|---|---|---|---|
| *Unadjusted* | 46.810 | 6.760 | 1 |
| *Lasso – 10 folds – all terms* | 49.579 | 5.118 | 1.744 |
| *Lasso – 20 folds – all terms* | 49.542 | 5.114 | 1.747 |
| *Ridge – 10 folds – all terms* | 49.637 | 5.108 | 1.751 |
| *Ridge – 20 folds – all terms* | 49.637 | 5.108 | 1.751 |
| | | | |
| *Lasso – 20 folds – full model* | 48.926 | 5.323 | 1.61 |
| *Ridge – 10 folds – full model* | 48.076 | 5.711 | 1.40 |

As we mentioned earlier, the models which are fully specified and contain all terms, all continuous squared terms, and all the two-way interactions, seems to underperform, so we see no value of fitting these complex models. On the other hand, it seems that by increasing the number of folds

[7] More info on `GLMNET` package in https://cran.r-project.org/web/packages/glmnet/glmnet.pdf
[8] We fitted the models with `SuperLearner` function, using the `GLMNET` model wrappers.

https://doi.org/10.26219/heal.aueb.5509

in cross validation from 10 to 20, we gain minimal improvement (ridge has the same estimates). Finally, out of all models, ridge regression seems to outperform all the others, since it has the higher relative efficiency gain.

### 3.3.5 Augmented Estimator – Rpart

Our next step of baseline covariate adjustment, we incorporated a Recursive Partitioning method, using regression decision trees, via the `Rpart`[9] R package. The essential idea behind tree-based methods, is that they partition the data (feature space) into smaller subgroups and then fit a simple model in each one (Hastie, et al., 2009). The partitioning is achieved by successive binary partitions (recursive partitioning) based on the different predictors. The constant to predict is based on the mean response of all observations that fall in that subgroup. In order to find the best binary partition, we used the Mean Squared Error (mse).

One of the key advantages of this method, is that the resulting recursive binary tree is easily interpretable, as a single tree can fully describe the feature space (Hastie, et al., 2009). This interpretability can be viewed by the way a tree stratifies the population of interest into different strata, a technique which is frequently used by medical experts.

For this method, we followed the same procedure and fitted a separate model for each treatment group, considering all covariates.

**Table 6 Estimates of θ for ACTG175 data using Rpart**

| Estimator | Estimate | Standard Error | Relative Efficiency |
|---|---|---|---|
| *Unadjusted* | 46.810 | 6.760 | 1 |
| *Rpart – all terms* | 52.133 | 4.982 | 1.84 |

For the recursive partitioning method, we considered the default settings, as they are configured in `SuperLearner` package. The number of minimum observations that must exists in a tree node was set to 20 (`minsplit`), while the complexity parameter (`minsplit`) was set to 0.01. The complexity parameter is used to save computing time by pruning off splits that are not worthwhile. In order to consider a split, the overall fit must be improved by a factor of cp = 0.01.

[9] More info on `Rpart` package in https://cran.r-project.org/web/packages/rpart/rpart.pdf

https://doi.org/10.26219/heal.aueb.5509

Once again, it is straightforward to notice that the augmentation procedure of recursive partitioning offers a big improvement when compared to unadjusted estimator. As we already mentioned, there are some clear advantages of this methods, such as the model fitting is fast and the results are easily interpretable. As an example, we present the decision trees graphs of each model for each different treatment group.

**Figure 4 Decision Tree of Rpart model for control group**

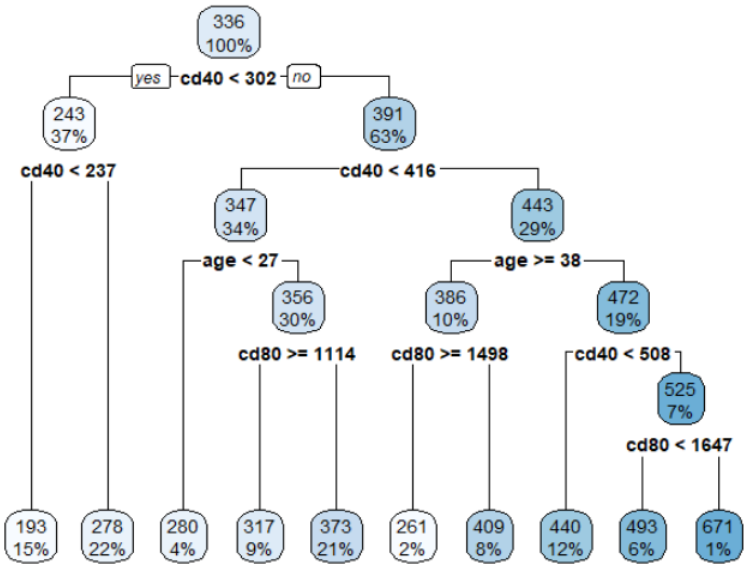
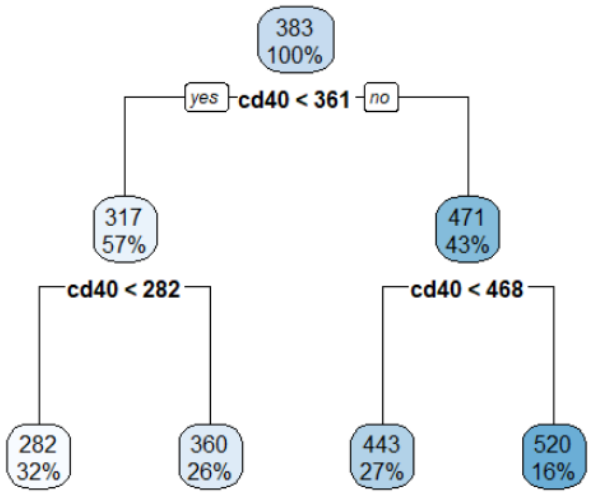



**Figure 5 Decision Tree of Rpart model for treatment group**



https://doi.org/10.26219/heal.aueb.5509

Each node shows the predicted value and the percentage of observations in the node. We will not get into much detail regarding the analysis of these graphs. Our focus is just to point the importance of covariate CD4 cell count at the baseline (cd40), in the analysis. As we highlighted previously, there is high association between this covariate and the response, which the unadjusted estimator fails to take into consideration, while the regression trees succeed in doing so.

### 3.3.6 Augmented Estimator – Random Forest

For our next method of augmenting the simple unadjusted estimator using baseline covariates, we used the logical extension of decision trees, the Random Forest method, based on the `randomForest` R package[10]. Random forest is a very popular method, which consists of a modification of bagging techniques.

In random forest, we build a large collection of decision trees and then we average them. As (Hastie, et al., 2009) already mentioned, the idea is to average many approximately unbiased models, which are noisy, in order to reduce the variance. In addition, the bias of bagged trees, is the same as that of individual trees, and there can be an improvement through variance reduction (Hastie, et al., 2009).

For this method, we fitted 6 different models in total. We experimented with 3 different settings regarding the number of trees that the forest consists of (`ntree` = { 100, 500, 1000} ). In addition, we varied the number of variables to randomly sample as candidates for each split (`mtry` = {3,4}).

**Table 7 Estimates of θ for ACTG175 data using Random Forest**

| Estimator | Estimate | Standard Error | Relative Efficiency |
|---|---|---|---|
| *Unadjusted* | 46.810 | 6.760 | 1 |
| *Random Forest, 4 variables, 100 trees* | 52.045 | 3.690 | 3.354 |
| *Random Forest, 4 variables, 500 trees* | 52.272 | 3.67 | 3.392 |
| *Random Forest, 4 variables, 1000 trees* | 52.275 | 3.65 | 3.416 |
| | | | |
| *Random Forest, 3 variables, 100 trees* | 52.515 | 3.84 | 3.097 |
| *Random Forest, 3 variables, 100 trees* | 51.894 | 3.832 | 3.110 |
| *Random Forest, 3 variables, 100 trees* | 52.024 | 3.824 | 3.125 |

[10] More info on `randomForest` package in https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

https://doi.org/10.26219/heal.aueb.5509

As we expect, by increasing the number of trees that consist a forest, we improve our results. In doing so, there is a big increase in computation time. In addition, we see that the by decreasing the number of variables to consider (mtry), we get worse estimators.

We are going to test whether these estimators are biased, later on our analysis, during the simulation chapter. Taking into consideration all the above results, as well as the limitations our computer hardware, we chose to include only the random forest model, where the number of variables to consider mtry = 4, and the number of trees, ntrees = 100.

### 3.3.7 Augmented Estimator – Super Learner

For our final augmentation, we used a combination of all the best methods we mentioned earlier, based an ensemble algorithm, using `SuperLearner`[11] R package. Following the example of (Zhang & Ma, 2019), we created a superlearner using the models: lm with all the covariates without interaction and squared terms, lasso 10-folds, ridge 10-folds, recursive partitioning, random forest mtry=4 and ntrees =100. We decided on these models based on both the relative efficiency measure, as well as the computation limitations. In addition, the library includes a trivial predictor based on the sample mean to create an "intercept" for the super learner model. Moreover, the super learner is implemented using a 5-fold cross-validation procedure.

Based on the results of the fitted models, the super learner algorithm set the Ridge regression coefficient as zero for each treatment group. The results of the super learner models are the following:

| **Method** | **Risk** | | **Coefficient** | |
|---|---|---|---|---|
| | **Control** | **Treatment** | **Control** | **Treatment** |
| *LM* | 9819.203 | 13905.38 | 0.5210 | 0.6326 |
| *Lasso* | 9793.657 | 13896.21 | 0.0436 | 0.0444 |
| *Ridge* | 9801.553 | 13904.41 | 0.0000 | 0.0000 |
| *Rpart* | 10525.527 | 15134.52 | 0.2495 | 0.0595 |
| *Random Forest* | 10151.232 | 14459.57 | 0.1859 | 0.2635 |

The estimates of the super learner method will be presented below as part of the numerical results

[11] More info on `SuperLearner` package in https://cran.r-project.org/web/packages/SuperLearner/SuperLearner.pdf

https://doi.org/10.26219/heal.aueb.5509

## 3.4 Numerical Results

Here we present a summary of our analysis, based on the best estimators of each method, including the super learner. As we expect, the efficiency gain by incorporating baseline covariates is straightforward, as all the models performed better than the unadjusted estimator.

**Table 8 Summary of Estimates of θ for ACTG175 data**

| Estimator | Estimate | Standard Error | Relative Efficiency |
|---|---|---|---|
| *Unadjusted* | 46.810 | 6.760 | 1 |
| *LM - All terms* | 49.818 | 5.101 | 1.755 |
| *Backward - 2* | 53.647 | 4.896 | 1.906 |
| *Lasso – 10 folds – all terms* | 49.579 | 5.118 | 1.744 |
| *Ridge – 10 folds – all terms* | 49.637 | 5.108 | 1.751 |
| *Rpart – all terms* | 52.133 | 4.982 | 1.84 |
| *Random Forest, 4 variables, 100 trees* | 52.045 | 3.690 | 3.354 |
| *Super Learner* | 51.02 | 4.68 | 2.07 |

As we expect, the super learner performs better almost most of the time. Except the case of random forest, which has the highest efficiency gain, we see that the super learner is the second best. On possible explanation may be that Random Forest estimators tend to have larger bias, as the primary concern is the minimization of variance. In order to test this, we will present a simulation analysis.

In addition, one may perform statistical inference by calculating the Wald Confidence Intervals based on the standard errors of the estimators. In addition, taking the super learner or the Backward 2 methods that have relative efficiency $\approx 2$, this has a real world application as by using these augmentation techniques, it allows us to maintain the same level of precision, by reducing the sample size in half.

One important notice, is that although our response CD4 is referring to count data, which means that we could fit Poisson or Log-Linear models, but since the data have a great range (Min =49, Max = 1119), as well as large variance (VAR = 20919.26) relative to the mean (Mean = 371.30), Linear Models seems to be more suitable. This is in accordance to other authors (Tsiatis, et al., 2008), which used linear models, for these data, as well.

Next in our analysis, is to perform a simulation to test the performance of these estimators.

https://doi.org/10.26219/heal.aueb.5509

# 4 Simulation Study

The aim of this simulation study is to empirically evaluate and access the different methods for covariate adjustment which described earlier. In order to make the simulation study relevant to the ACTG175 study, we will take the support set of the simulation to be the collection of observed values $(\boldsymbol{X}, Y)$.

## 4.1 Data Generation

Our data simulation generating algorithm is based on the bootstrap technique (Efron & Tibshirani, 1986). In detail, we will use a non-parametric bootstrap method, since we do not know the form of $F$ distribution, we estimate the empirical version $\hat{F}$ which is obtained from our data.

The simulation setup which we are going to use is similar to the one used by (Zhang & Ma, 2019). First, we consider a population of patients with potential outcomes $Y(0) \equiv Y(1)$ and $\boldsymbol{X}$ be a vector of baseline covariates. Let $F$ denote the joint distribution of $(\boldsymbol{X}, Y)$, and $F$ is a discrete distribution with support $\{(\boldsymbol{x}_i, y_i),\ i = 1, \dots, N\}$.

It follows that:

$$F = \sum_{i=1}^{N} p_i \delta_i$$

Where $p_i \geq 0$ and $\sum_{i=1}^{N} p_i = 1$. Moreover, $\delta_i$ denotes a point mass of 1 at $(\boldsymbol{x}_i, y_i)$.

We now consider a random sample of size $n$ from a randomized clinical trial which is based on the above population. We now have $(\boldsymbol{X}_i, Y_i)$, $i = 1, \dots, n$ which are the baseline covariate and outcome values in our sample. As a result, $(\boldsymbol{X}_i, Y_i)$ are independent and identically distributed (i.i.d) according to $F$. Finally, we have $T_i$ denote the randomized treatment indicator for the $i$th subject in our sample, where

$$T \sim Bernoulli\ (\pi)\ \ and\ \ T \perp \boldsymbol{X}$$

with success probability $\pi$ independent of $\boldsymbol{X}$.

https://doi.org/10.26219/heal.aueb.5509

## 4.2 Design of Experiment

To best select the number of simulations denoted as $n_{sim}$ , there are various suggestions among the international bibliography such the following, as noted by (Morris, et al., 2019)

$$Monte\ Carlo\ SE\ (Bias) = \sqrt{\frac{Var(\hat{\theta})}{n_{sim}}}$$

Upon taking into consideration both the Monte Carlo standard error, as well as the hardware limitations and the execution time, in our example we will be performing $n_{sim} = 1000$ simulations.

Moreover, regarding the sample size of each simulation, denoted as $n_{obs}$ , we choose it to be $n_{obs} = 400$ which is a common practice around clinical trials.

In addition, we will test the performance of the different methods that were mentioned earlier, using success probability for the treatment assignment π, where we will set $\pi = 0.5$ which is more common practice amongst clinical trials.

## 4.3 Performance Measures

For each of the proposed methods, we will estimate some performance measures, in order to be able to best compare the augmentation techniques with each other. As (Morris, et al., 2019) have proposed, when the primary point of interest is regarding an estimator, some of the basic measures to be compared and reported are Bias and Coverage Probability. For our analysis, we will use the following equations during the estimation procedure:

$$Bias = \frac{1}{n_{sim}} \sum_{i=1}^{n_{sim}} \tilde{\theta}_i - \theta \qquad (20)$$

$$Monte\ Calro\ SE\ of\ Bias = \frac{1}{n_{sim}(n_{sim}-1)} \sum_{i=1}^{n_{sim}} \left(\tilde{\theta}_i - \bar{\theta}\right)^2 \qquad (21)$$

https://doi.org/10.26219/heal.aueb.5509

$$Empirical\ SE\ =\ \sqrt{\frac{1}{n_{sim}-1}\sum_{i=1}^{n_{sim}}\left(\tilde{\theta}_i-\bar{\theta}\right)^2} \tag{22}$$

$$Coverage\ =\ \frac{1}{n_{sim}}\sum_{i=1}^{n_{sim}}1\left(\tilde{\theta}_{low,i}\ \leq\theta\ \leq\ \tilde{\theta}_{upp,i}\right)^2 \tag{23}$$

In order to estimate the coverage, we need to calculate the upper and lower limits of the estimates. In doing do, we will use the equation of bootstrap confidence interval, since we expect that our estimators are approximately normal, as we mentioned earlier. The equation for the confidence intervals is:

$$Simple\ Bootstrap\ CI\ =\ \left(\tilde{\theta}-Z_{1-a/2}\ se_B\left(\tilde{\theta}\right),\quad \tilde{\theta}-Z_{a/2}\ se_B\left(\tilde{\theta}\right)\right) \tag{24}$$

Where $Z_a$ denotes the a-th quantile of the standard normal distribution. In our case, we will construct a 95% (a=5%) confidence interval.

## 4.4 Numerical Results

In this chapter, we present the numerical result of the simulation analysis for the augmentation algorithms we presented earlier. To compare the methods, our main points of interest are the bias of each methods, as well as the relative efficiency (RE), and the coverage probability (CP) compared to the unadjusted estimator.

**Table 9 Simulation Results of augmentation techniques for π = 0.50**

| Estimator | Bias | SE of bias | Emp SE | RE | CP |
|---|---|---|---|---|---|
| *Unadjusted* | 0.122 | 0.442 | 13.997 | 1.000 | 0.948 |
| *LM - All terms* | 3.025 | 0.349 | 11.046 | 1.606 | 0.936 |
| *Lasso – 10 folds – all terms* | 2.651 | 0.347 | 10.982 | 1.624 | 0.939 |
| *Ridge – 10 folds – all terms* | 2.750 | 0.349 | 11.056 | 1.603 | 0.933 |
| *Rpart – all terms* | 4.190 | 0.359 | 11.364 | 1.517 | 0.927 |
| *Random Forest, mtry = 4, ntree= 100* | 3.781 | 0.347 | 10.999 | 1.619 | 0.938 |
| *Super Learner* | 3.569 | 0.484 | 10.822 | 1.672 | 0.938 |

https://doi.org/10.26219/heal.aueb.5509

As we expected, all the augmentation techniques produce biased estimates, while the unadjusted estimator has low bias compared to the other methods. In the other hand, we notice that all augmentation techniques are at least 1.5 more efficient that the unadjusted estimator as they have smaller Empirical Standard Errors (Emp SE), while the Super Learner is the more efficient one. As (Zhang & Ma, 2019) noted, the super learner performs better (or no worse) than the candidate Algorithms

On possible solution to the reduction of bias is to perform another external cross-validation as showcased by (Zhang & Ma, 2019), which we avoided for our analysis due to complexity reasons. In addition, these methods seem to be somewhat sensitive to the specific dataset, as the prediction algorithm can be varied in many ways. Perhaps, we could also consider another simulation, one that is more aligned with the nature of this specific dataset.

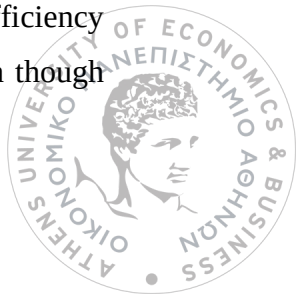
# 5 Conclusion – Remarks

To conclude this thesis, we reviewed some covariate adjustment techniques which aim to improve the efficiency of a randomized clinical trial study, by incorporating auxiliary baseline covariates. For these methods to hold, we appeal to the semi-parametric theory as presented by (Yang & Tsiatis, 2001) (Tsiatis, et al., 2008), where we estimate the simple unadjusted estimator and by using an arbitrary function $\alpha(\boldsymbol{X}_i)$, we leveraged the baseline information and augment the estimator.

The various augmentation techniques we used in our analysis include the ANCOVA (Yang & Tsiatis, 2001), linear models using stepwise procedures (Tsiatis, et al., 2008), as well as various machine learning algorithms like recursive partitioning and Random Forest (Zhang & Ma, 2019). Ultimately, we combined the above prediction methods to construct a super learner ensemble algorithm (Zhang & Ma, 2019).

We examined the use of the augmentation technique on a real example dataset, based on an AIDS Clinical Trial (ACTG175). We saw that all these augmentation techniques offered a great efficiency boost, when compared to the unadjusted estimator. Using the relative efficiency measure, the best method was the Random Forest, followed by the super learner. Even though





https://doi.org/10.26219/heal.aueb.5509

these models are a bit complex by nature, we found out that there are actual efficiency gains, so they should be considered during the analysis of a clinical trial data.

Finally, we examined the performance of these covariate adjustment methods by performing a simulation analysis, which was relevant to the original dataset ACTG175. During this simulation we found out that all of these techniques suffer from high bias, compared to the unadjusted estimator. On the other hand, the relative efficiency of these techniques suggested that we could achieve at least 1.5 gain in efficiency by leveraging baseline covariates information. As a final suggestion / remark, we should point out that, one could use other models (Poisson, Log-Linear or Negative Binomial) to better capture the true nature of the data, since we are referring to count data.

https://doi.org/10.26219/heal.aueb.5509

# 6 References

Armitage, P., 1982. The Role of Randomization in Clinical Trials. *Statistics in Medicine* , Volume 1, pp. 345-352.

Efron, B. & Tibshirani, R., 1986. Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy. *Statistical Science,* 1(1), pp. 54-77.

Friedman, L. M. et al., 2015. *Fundamentals of Clinical Trials.* 5th ed. Switzerland: Springer International Publishing Switzerland.

Hammer, S. et al., 1996. A Trial Comparing Nucleoside Monotherapy with Combination Therapy in HIV-Infected Adults with CD4 Cell Counts from 200 to 500 per Cubic Millimeter. *The New England Journal of Medicine,* 335(15), pp. 1081-1090.

Harrell, F. J. E., 2015. *Regression Modeling Strategies with Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis.* s.l.:Springer, Cham.

Hastie, T., Tibshirani, R. & Friedman, J., 2009. *The Elements of Statistical Learning Data Mining, Inference, and Prediction.* 2nd ed. New York: Springer, New York,.

Lachin, J. M., 1981. Introduction to Sample Size Determination and Power Analysis for Clinical trials. *Controlled Clinical Trials,* 2(2), pp. 93-113.

Leon, S., Tsiatis, A. & Davidian, M., 2003. Semiparametric estimation of treatment effect in a pretest - posttest study. *Biometrics,* 59(4), pp. 1046 - 1055.

Mark, J., van der Laan, Polley, E. C. & Hubbard, A. E., 2007. Super Learner. *Statistical Applications in Genetics and Molecular Biology,* 6(1).

Moher, D., Dulberg, C. & Wells, G., 1994. Statistical power, sample size, and their reporting in randomized controlled trials.. *JAMA,* 272(2), pp. 122-124.

Morris, T. P., White, I. R. & Crowther, M. J., 2019. Using simulation studies to evaluate statistical methods. *Statistics in Medicine,* pp. 2074-2102.

https://doi.org/10.26219/heal.aueb.5509

Pocock, S. J., Assmann, S. E., Enos, L. E. & Kasten, L. E., 2002. Subgroup analysis, covariate adjustment and baseline comparisons in clinical trial reporting: current practice and problems. *Statistics in Medicine,* 21(19), pp. 2917-2930.

Robert, C. P. & Casella, G., 2010. *Introduction Monte Carlo Methods with R.* New York: Springer-Verlag New York.

Tian, L., Cai, T., Zhao, L. & Wei, L.-J., 2012. On the covariate-adjusted estimation for an overall treatment difference with data from a randomized comparative clinical trial. *Biostatistics,* 13(2), pp. 256-273.

Tsiatis, A. A., Davidian, M., Zhang, M. & Lu, X., 2008. Covariate adjustment for two-sample treatment comparisons in randomized clinical trials: A principled yet flexible approach. *Statist Med,* 27(23), pp. 4658-4677.

White, I. R., 2010. simsum: Analyses of simulation studies including Monte Carlo error. *The Stata Journal,* 10(3), pp. 369-385.

Yang, L. & Tsiatis, A. A., 2001. Efficiency Study of Estimators for a Treatment Effect in a Pretest-Posttest Tri. *The American Statistician,* 55(4), pp. 314-321.

Zhang, M., Tsiatis, A. A. & Davidian, M., 2008. Improving efficiency of inferences in randomized clinical trials using auxiliary covariates. *Biometrics,* 64(3), pp. 707-715.

Zhang, Z. & Ma, S., 2019. Machine learning methods for leveraging baseline covariate information to improve the efficiency of clinical trials. *Statistics in Medicine,* pp. 1703-1714.

https://doi.org/10.26219/heal.aueb.5509

# 7 APPENDIX

## 7.1 Fitted Models

**Table 10 ANCOVA Model of CD4 cell counts**

| | cd420 | | | | | |
|---|---|---|---|---|---|---|
| *Predictors* | *Estimates* | *std. Error* | *CI* | *Statistic* | *p* | *df* |
| (Intercept) | 67.16 | 45.05 | -21.19 – 155.51 | 1.49 | 0.136 | 2125.00 |
| cd40 | 0.70 | 0.02 | 0.65 – 0.74 | 32.31 | **<0.001** | 2125.00 |
| cd80 | -0.02 | 0.01 | -0.03 – -0.01 | -3.66 | **<0.001** | 2125.00 |
| age | -0.36 | 0.29 | -0.94 – 0.21 | -1.23 | 0.217 | 2125.00 |
| wtkg | -0.04 | 0.19 | -0.42 – 0.34 | -0.19 | 0.852 | 2125.00 |
| karnof | 1.00 | 0.42 | 0.17 – 1.83 | 2.37 | **0.018** | 2125.00 |
| hemo [1] | -27.79 | 11.68 | -50.70 – -4.88 | -2.38 | **0.017** | 2125.00 |
| homo [1] | -0.39 | 8.54 | -17.13 – 16.36 | -0.05 | 0.964 | 2125.00 |
| drugs [1] | 7.50 | 7.60 | -7.40 – 22.40 | 0.99 | 0.324 | 2125.00 |
| race [1] | -19.51 | 5.85 | -30.99 – -8.04 | -3.33 | **0.001** | 2125.00 |
| gender [1] | -4.35 | 9.60 | -23.18 – 14.49 | -0.45 | 0.651 | 2125.00 |
| str2 [1] | -40.55 | 5.11 | -50.57 – -30.53 | -7.94 | **<0.001** | 2125.00 |
| symptom [1] | -18.46 | 6.60 | -31.40 – -5.52 | -2.80 | **0.005** | 2125.00 |
| treat [1] | 49.69 | 5.65 | 38.62 – 60.77 | 8.80 | **<0.001** | 2125.00 |
| Observations | 2139 | | | | | |
| $R^2$ / $R^2$ adjusted | 0.397 / 0.393 | | | | | |

**Table 11 ANCOVA Model - Backward Selection of CD4 cell count**

| | cd420 | | | | | |
|---|---|---|---|---|---|---|
| *Predictors* | *Estimates* | *std. Error* | *CI* | *Statistic* | *p* | *df* |
| (Intercept) | 47.50 | 41.18 | -33.26 – 128.27 | 1.15 | 0.249 | 2130.00 |

https://doi.org/10.26219/heal.aueb.5509

| | | | | | | |
|---|---|---|---|---|---|---|
| cd40 | 0.70 | 0.02 | 0.66 – 0.74 | 32.50 | <**0.001** | 2130.00 |
| cd80 | -0.02 | 0.01 | -0.03 – -0.01 | -3.84 | <**0.001** | 2130.00 |
| karnof | 1.00 | 0.42 | 0.18 – 1.82 | 2.40 | **0.017** | 2130.00 |
| hemo [1] | -26.02 | 8.92 | -43.51 – -8.53 | -2.92 | **0.004** | 2130.00 |
| race [1] | -17.03 | 5.42 | -27.66 – -6.39 | -3.14 | **0.002** | 2130.00 |
| str2 [1] | -40.66 | 5.06 | -50.57 – -30.74 | -8.04 | <**0.001** | 2130.00 |
| symptom [1] | -18.44 | 6.58 | -31.34 – -5.54 | -2.80 | **0.005** | 2130.00 |
| treat [1] | 49.76 | 5.63 | 38.72 – 60.81 | 8.83 | <**0.001** | 2130.00 |
| Observations | 2139 | | | | | |
| $R^2$ / $R^2$ adjusted | 0.396 / 0.394 | | | | | |

## 7.2 Session Info

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] tibble_3.0.4        rpart.plot_3.0.9    lattice_0.20-38
##  [4] SuperLearner_2.0-26 nnls_1.4            sjPlot_2.8.6
##  [7] randomForest_4.6-14 rpart_4.1-15        glmnet_4.0-2
## [10] Matrix_1.2-18       car_3.0-10          carData_3.0-4
## [13] speff2trial_1.0.4   survival_3.1-8      leaps_3.1
##
## loaded via a namespace (and not attached):
##  [1] tidyr_1.1.2       splines_3.6.2     foreach_1.5.1     modelr_0.1.8
##  [5] statmod_1.4.35    cellranger_1.1.0  yaml_2.2.1        bayestestR_0.7.5
##  [9] pillar_1.4.7      backports_1.2.0   glue_1.4.2        digest_0.6.27
## [13] minqa_1.2.4       colorspace_2.0-0  htmltools_0.5.0   pkgconfig_2.0.3
## [17] broom_0.7.2       haven_2.3.1       purrr_0.3.4       xtable_1.8-4
## [21] mvtnorm_1.1-1     scales_1.1.1      openxlsx_4.2.3    rio_0.5.16
## [25] lme4_1.1-25       emmeans_1.5.2-1   generics_0.1.0    ggplot2_3.3.2
```

https://doi.org/10.26219/heal.aueb.5509

```
## [29] sjlabelled_1.1.7  ellipsis_0.3.1    magrittr_2.0.1    crayon_1.3.4
## [33] effectsize_0.4.0  readxl_1.3.1      estimability_1.3  evaluate_0.14
## [37] nlme_3.1-142      MASS_7.3-51.4     forcats_0.5.0     foreign_0.8-72
## [41] tools_3.6.2       data.table_1.13.2 hms_0.5.3         lifecycle_0.2.0
## [45] stringr_1.4.0     munsell_0.5.0     zip_2.1.1         ggeffects_0.16.0
## [49] compiler_3.6.2    rlang_0.4.9       grid_3.6.2        nloptr_1.2.2.2
## [53] iterators_1.0.13  parameters_0.9.0  rmarkdown_2.5     boot_1.3-23
## [57] gtable_0.3.0      codetools_0.2-16  abind_1.4-5       sjstats_0.18.0
## [61] curl_4.3          sjmisc_2.8.5      R6_2.5.0          knitr_1.30
## [65] dplyr_1.0.2       performance_0.5.1 insight_0.11.0    shape_1.4.5
## [69] stringi_1.5.3     Rcpp_1.0.5        vctrs_0.3.5       tidyselect_1.1.0
## [73] xfun_0.19
```

## 7.3 Libraries

```
library("speff2trial")
library("car")
library("glmnet")
library("rpart")
library("randomForest")
library("sjPlot")
library("SuperLearner")
library('lattice')
library("rpart.plot")
library("tibble")
```

## 7.4 Custom Functions

```
#calculate empirical influence function estimates for difference in means
infl.fun <- function (response, treatment){
  m0 <- mean(response[treatment == 0])
  m1 <- mean(response[treatment == 1])
  p  <- length(treatment[treatment == 1])/length(treatment)
  s  <- (treatment*(response - m1))/p - ((1-treatment)*(response - m0))/(1-p)
  return(s)
}

infl.fun2 <- function (response, treatment){
  p  <- length(treatment[treatment == 1])/length(treatment)
  m0 <- sum(( (1-treatment) * response) / ( (1-p)* length(treatment)))
  m1 <- sum(( treatment * response) / ( p * length(treatment)))
  s  <- (treatment*(response - m1))/p - ((1-treatment)*(response - m0))/(1-p)
  return(s)
}


#Estimete the augmented estimator, standard error and relative efficiency
theta_aug <- function( m_control, m_treat, data)
{
  pred_c <- predict(m_control, data[,-c(2,14)])
```

https://doi.org/10.26219/heal.aueb.5509

```
  pred_t <- predict(m_treat, data[,-c(2,14)])
  ax<-0
  if(class(m_control)== "SuperLearner"){
    pred_t <-pred_t$pred
    pred_c <-pred_c$pred
    ax<-pred_t-pred_c
  }

  n0 <- length(data$cd420[data$treat ==0])
  n1 <- length(data$cd420[data$treat ==1])
  prop <- n1/(n0+n1)
  spred_c <- (1/n0)*pred_c
  spred_t <- (1/n1)*pred_t
  meany_c <- mean(data$cd420[data$treat == 0])
  meany_t <- mean(data$cd420[data$treat == 1])
  meanpred_c <- mean(pred_c)
  meanpred_t <- mean(pred_t)
  t0 <- meany_c - meanpred_c
  t1 <- meany_t - meany_t
  treat_indicator <- as.numeric(levels(data$treat))[data$treat]
  #treat_indicator <- data$treat
  theta_est <- meany_t - meany_c - sum( (treat_indicator - prop) * (spred_c + spred_t
) )
  theta_var <- sum((( (1/n1)*treat_indicator - (1/n0)*(1-treat_indicator) ) * data$cd
420 - 1/(n0+n1)*theta_est - (treat_indicator - prop)*( spred_c + spred_t ) - (treat_i
ndicator - prop)*(((1/n0)*t0) + ((1/n1)*t1)) )^2 )

  seun <- 6.760197 # sqrt(sd(cd420[treat == 1])^2/length(cd420[treat == 1]) + sd(cd42
0[treat == 0])^2/length(cd420[treat == 0]))
  rel_ef<- (seun^2)/theta_var
  ret <- list(theta_est, sqrt(theta_var), (rel_ef))
  names(ret) <- c('theta', 'SE theta', 'Rel Eff')
  return(ret)
}


#estimate variance of augmented theta
aug.var <-function (response, data){
  treatment<- data$treat
  p <- length(treatment[treatment == 1])/length(treatment)
  n <- length(treatment)
  s <- sum( (infl.fun(response, treatment) - (treatment-p)*aug.fun(...))^2)/n
  return(s)
}


#
theta_aug2 <- function( m_control, m_treat, data)
{
  pred_c <- predict(m_control, data[,-c(1,2)])

  pred_t <- predict(m_treatment, data[,-c(1,2)])
  ax<-0
  if(class(m_control)== "SuperLearner"){
    pred_t <-pred_t$pred
```



https://doi.org/10.26219/heal.aueb.5509

```
    pred_c <-pred_c$pred
    ax<-pred_t-pred_c
  }
  n0 <- length(data$cd420[data$treat ==0])
  n1 <- length(data$cd420[data$treat ==1])
  prop <- n1/(n0+n1)
  spred_c <- (1/n0)*pred_c
  spred_t <- (1/n1)*pred_t
  meany_c <- mean(data$cd420[data$treat == 0])
  meany_t <- mean(data$cd420[data$treat == 1])

  meanpred_c <- mean(pred_c)
  meanpred_t <- mean(pred_t)
  t0 <- meany_c - meanpred_c
  t1 <- meany_t - meany_t
  treat_indicator <- as.numeric(data$treat)
  theta_est <- meany_t - meany_c - sum( (treat_indicator - prop) * (spred_c + spred_t
) )
  theta_var <- sum((( (1/n1)*treat_indicator - (1/n0)*(1-treat_indicator) ) * data$cd
420 - 1/(n0+n1)*theta_est - (treat_indicator - prop)*( spred_c + spred_t ) - (treat_i
ndicator - prop)*(((1/n0)*t0) + ((1/n1)*t1)) )^2 )

  seun <- 6.760197 # sqrt(sd(cd420[treat == 1])^2/length(cd420[treat == 1]) + sd(cd42
0[treat == 0])^2/length(cd420[treat == 0]))
  rel_ef<- (seun^2) / theta_var
  ret <- list(theta_est, sqrt(theta_var), (rel_ef))
  names(ret) <- c('theta', 'SE theta', 'Rel eff')
  return(ret)
}



#Fit Super learner models for specific wrapper - all linear terms
model_fitting <-function(data, wrapper = 'SL.glm')
{
  set.seed(1)
  m_control <- SuperLearner(data$cd420[data$treat == 0], data[data$treat == 0, -c(2,1
4)],  SL.library = wrapper )
  set.seed(1)
  m_treatment <- SuperLearner(data$cd420[data$treat == 1], data[data$treat == 1, -c(2
,14)],  SL.library = wrapper )

  theta <- theta_aug(m_control, m_treatment, data)
  return(theta)
}


#Fit Super learner models for specific wrapper - full specified data
model_fitting2 <-function(data, wrapper = 'SL.glm')
{
  mm <- model.matrix(cd420~.^2 + I(cd40^2) + I(cd80^2) + I(age^2) + I(wtkg^2) + I(kar
nof^2), data = data[,-14])
  set.seed(1)
  m_control <- SuperLearner(data$cd420[data$treat == 0], mm[data$treat == 0, -c(1,14)
],  SL.library = wrapper )
```

https://doi.org/10.26219/heal.aueb.5509

```
  set.seed(1)
  m_treatment <- SuperLearner(data$cd420[data$treat == 1], mm[data$treat == 1, -c(1,1
4)],  SL.library = wrapper )

  theta <- theta_aug2(m_control, m_treatment, data.frame(cbind(cd420, treat, mm[,-c(1
)])))
  return(theta)
}


##Super learner wrappers - GLMNETS
SL.rf.glmnet_lasso1 = function(...) {
  SL.glmnet(..., alpha = 1, loss = 'mse',  nfolds = 10  )
}

SL.rf.glmnet_lasso2 = function(...) {
  SL.glmnet(..., alpha = 1, loss = 'mse',  nfolds = 20  )
}

SL.rf.glmnet_ridge1 = function(...) {
  SL.glmnet(..., alpha = 0,  loss = 'mse',  nfolds = 10   )
}

SL.rf.glmnet_ridge2 = function(...) {
  SL.glmnet(..., alpha = 0,  loss = 'mse',  nfolds = 20  )
}




#Super learner wrappers - Random Forest
SL.rf.rforest.4.100 = function(...) {
  SL.randomForest(..., mtry = 4 , ntree =100 )
}

SL.rf.rforest.4.500 = function(...) {
  SL.randomForest(..., mtry = 4 , ntree =500 )
}

SL.rf.rforest.4.1000 = function(...) {
  SL.randomForest(..., mtry = 4 , ntree =1000 )
}


SL.rf.rforest.3.100 = function(...) {
  SL.randomForest(..., mtry = 3 , ntree =100 )
}

SL.rf.rforest.3.500 = function(...) {
  SL.randomForest(..., mtry = 3 , ntree =500 )
}

SL.rf.rforest.3.1000 = function(...) {
  SL.randomForest(..., mtry = 3 , ntree =1000 )
}
```



https://doi.org/10.26219/heal.aueb.5509

```
#Fitt models for simulated data
simul<- function (N_sim  = 100, N_obs = 400, method = 'all', data, theta = 46.8105, s
e_theta = 6.760197){

  if (method == "lm"){
    f_wrapper <- "SL.lm"
  } else if ( method == "lasso"){
    f_wrapper <- "SL.rf.glmnet_lasso1"
  } else if (method == 'ridge') {
    f_wrapper <- "SL.rf.glmnet_ridge1"
  } else if (method == "rpart") {
    f_wrapper <- "SL.rpart"
  } else if (method == "forest") {
    f_wrapper <- "SL.rf.rforest.4.100"
  } else {
    f_wrapper <- c("SL.lm", "SL.rf.glmnet_lasso1", "SL.rf.glmnet_ridge1" , "SL.rpart"
, "SL.rf.rforest.4.100" )
  }

  sim_measures <- data.frame(theta_stim = numeric(N_sim), se_stim = numeric(N_sim), n
1_sim = numeric(N_sim), n0_sim = numeric(N_sim))

  set.seed(100)
  for (i in 1:N_sim) {

    temp <- model_fitting(data[data$nsim == i, -c( 15,16)], wrapper = f_wrapper )

    n1_sim <- sum(data[(data$nsim == i),]$treat==1)
    n0_sim <- N_obs - n1_sim
    sim_measures[i, ] <- cbind(temp$theta, temp$`SE theta`, n1_sim, n0_sim)
  }


  bias <-mean(sim_measures$theta_stim) - theta
  se_bias <- sqrt ( sum((sim_measures$theta_stim - mean(sim_measures$theta_stim))^2)/
(N_sim* (N_sim - 1 )))
  B_se<- sqrt ( sum((sim_measures$theta_stim - mean(sim_measures$theta_stim))^2)/( (N
_sim - 1 )))


  CI <- data.frame(theta_low = (sim_measures$theta_stim + qnorm(0.025) *  B_se),
                   theta_high = (sim_measures$theta_stim + qnorm(0.975)* B_se) )

  #coverage probability
  cp <- mean((theta >= CI$theta_low & theta <= CI$theta_high))

  ret <- list(bias, se_bias, B_se, cp)
  names(ret) <- c('Bias', 'SE Bias',"EMP SE", "CP")
  return(ret)

}
```



https://doi.org/10.26219/heal.aueb.5509

## 7.5 R – Data Simulation

```
########## DATASET GENERATION ############

#Constants
N_sim <- 1000 #number of simulations
N_obs <- 400 #number of observations per simulation

#get indexes of treatment and control
g_treat <- which(newdata2$treat ==1 )
g_control <- which(newdata2$treat == 0)


set.seed(100)

#create treatment indicator
treat_ind <- rbinom(n = N_obs, size = 1, prob = 0.75)

#create index
dt <- c(sample(g_treat, size = sum(treat_ind == 1), replace = TRUE), sample(g_control
, size = sum(treat_ind == 0), replace = TRUE))

#Append data from original dataset based in index
df_0.75 <- data.frame( cd40 = newdata2$cd40[dt],        cd420 = newdata2$cd420[dt],
cd80 = newdata2$cd80[dt],
                       age = newdata2$age[dt],          wtkg = newdata2$wtkg
[dt],        karnof = newdata2$karnof[dt],
                       hemo = as.numeric(levels(newdata2$hemo[dt]))[newdata2
$hemo[dt]],
                       homo = as.numeric(levels(newdata2$homo[dt]))[newdata2
$homo[dt]],
                       drugs = as.numeric(levels(newdata2$drugs[dt]))[newdat
a2$drugs[dt]],
                       race = as.numeric(levels(newdata2$race[dt]))[newdata2
$race[dt]],
                       gender = as.numeric(levels(newdata2$gender[dt]))[newd
ata2$gender[dt]],
                       str2 = as.numeric(levels(newdata2$str2[dt]))[newdata2
$str2[dt]],
                       symptom = as.numeric(levels(newdata2$symptom[dt]))[ne
wdata2$symptom[dt]],
                       treat = as.numeric(levels(newdata2$treat[dt]))[newdat
a2$treat[dt]],      which = dt,
                       nsim = 1)

#begin simulation
for (i in 2:N_sim) {

  #create treatment indicator
  treat_ind <- rbinom(n = N_obs, size = 1, prob = 0.75)
```



https://doi.org/10.26219/heal.aueb.5509

```
  #create index
  dt <- c(sample(g_treat, size = sum(treat_ind == 1), replace = TRUE), sample(g_contr
ol, size = sum(treat_ind == 0), replace = TRUE))

  #Append data from original dataset based in index
  df_0.75 <- df_0.75 %>% add_row( cd40 = newdata2$cd40[dt],        cd420 = newdata2$c
d420[dt],      cd80 = newdata2$cd80[dt],
                                  age = newdata2$age[dt],          wtkg = newdata2$wt
kg[dt],        karnof = newdata2$karnof[dt],
                                  hemo = as.numeric(levels(newdata2$hemo[dt]))[newdat
a2$hemo[dt]],
                                  homo = as.numeric(levels(newdata2$homo[dt]))[newdat
a2$homo[dt]],
                                  drugs = as.numeric(levels(newdata2$drugs[dt]))[newd
ata2$drugs[dt]],
                                  race = as.numeric(levels(newdata2$race[dt]))[newdat
a2$race[dt]],
                                  gender = as.numeric(levels(newdata2$gender[dt]))[ne
wdata2$gender[dt]],
                                  str2 = as.numeric(levels(newdata2$str2[dt]))[newdat
a2$str2[dt]],
                                  symptom = as.numeric(levels(newdata2$symptom[dt]))[
newdata2$symptom[dt]],
                                  treat = as.numeric(levels(newdata2$treat[dt]))[newd
ata2$treat[dt]],      which = dt,
                                  nsim = i)

}

#proportion = 0.5
set.seed(100)
treat_ind <- rbinom(n = N_obs, size = 1, prob = 0.50)
dt <- c(sample(g_treat, size = sum(treat_ind == 1), replace = TRUE), sample(g_control
, size = sum(treat_ind == 0), replace = TRUE))

df_0.50 <- cbind(newdata2[dt,], which = dt, nsim = 1)

for (i in 2:N_sim) {
  treat_ind <- rbinom(n = N_obs, size = 1, prob = 0.50)
  dt <- c(sample(g_treat, size = sum(treat_ind == 1), replace = TRUE), sample(g_contr
ol, size = sum(treat_ind == 0), replace = TRUE))

  df_0.50 <- df_0.50 %>% add_row( cd40 = newdata2$cd40[dt],        cd420 = newdata2$c
d420[dt],      cd80 = newdata2$cd80[dt],
                                  age = newdata2$age[dt],          wtkg = newdata2$wt
kg[dt],        karnof = newdata2$karnof[dt],
                                  hemo = newdata2$hemo[dt],        homo = newdata2$ho
mo[dt],        drugs = newdata2$drugs[dt],
                                  race = newdata2$race[dt],        gender = newdata2$
gender[dt],    str2 = newdata2$str2[dt],
                                  symptom = newdata2$symptom[dt],  treat = newdata2$t
reat[dt],      which = dt,
                                  nsim = i)
}
```



https://doi.org/10.26219/heal.aueb.5509

## 7.6 Model Fitting – Misc

```
#unadjusted
mean(cd420[treat == 1]) - mean(cd420[treat == 0])
sqrt(sd(cd420[treat == 1])^2/length(cd420[treat == 1]) + sd(cd420[treat == 0])^2/leng
th(cd420[treat == 0]))

#ancova1
mancova <-lm(cd420 ~ . -cd40, data=newdata2)
summary(mancova)
shapiro.test(mancova$residuals)
tab_model(mancova)
tab_model(mancova, show.stat = TRUE, show.se = TRUE, show.df = TRUE)
plot(mancova)
bartlett.test(mancova$residuals~newdata2$treat)

#ancova2
mancova2 <-lm(cd420 ~ . -cd40, data=newdata2)
mancova2 <-step(mancova2)
summary(mancova2)
shapiro.test(mancova2$residuals)

#All
m_control <-lm(cd420 ~ ., data = newdata2[newdata2$treat == 0, -14])
summary(m_control)
shapiro.test(m_control$residuals)

m_treat <-lm(cd420 ~ ., data = newdata2[newdata2$treat == 1, -14])
summary(m_treat)
shapiro.test(m_treat$residuals)

theta_aug(m_control, m_treat, newdata2)

#Backward

m_for1_control <-lm(cd420 ~ ., data = newdata2[newdata2$treat == 0, -14])
m_for1_control <- step(m_for1_control)
summary(m_for1_control)
shapiro.test(m_for1_control$residuals)

m_for1_treat <-lm(cd420 ~ ., data = newdata2[newdata2$treat == 1, -14])
m_for1_treat <- step(m_for1_treat)
summary(m_for1_treat)
shapiro.test(m_for1_treat$residuals)

theta_aug(m_for1_control, m_for1_treat, newdata2)

#Forward2
m_for2_control <-lm(cd420 ~ .^2  + I(cd40^2) + I(cd80^2) + I(age^2) + I(wtkg^2) + I(k
arnof^2), data = newdata[newdata$treat == 0, -14])
m_for2_control <- step(m_for2_control)
summary(m_for2_control)

shapiro.test(m_for2_control$residuals)
```



https://doi.org/10.26219/heal.aueb.5509

```
m_for2_treat <- lm(cd420 ~ .^2 + I(cd40^2) + I(cd80^2) + I(age^2) + I(wtkg^2) + I(kar
nof^2), data = newdata[newdata$treat == 1, -14])
m_for2_treat <- step(m_for2_treat)
summary(m_for2_treat)
shapiro.test(m_for2_treat$residuals)

theta_aug(m_for2_control, m_for2_treat, newdata2)
```

```
#glmnet on all terms
model_fitting(newdata2, "SL.rf.glmnet_lasso1" )
model_fitting(newdata2, "SL.rf.glmnet_lasso2" )
model_fitting(newdata2, "SL.rf.glmnet_ridge1" )
model_fitting(newdata2, "SL.rf.glmnet_ridge2" )

#glmnet on full models
model_fitting2(newdata2, "SL.rf.glmnet_lasso1" )
model_fitting2(newdata2, "SL.rf.glmnet_lasso2" )
model_fitting2(newdata2, "SL.rf.glmnet_ridge1" )
model_fitting2(newdata2, "SL.rf.glmnet_ridge2" )

options(warn=0)

model_fitting(newdata2, "SL.rpart")

df75<-df_0.75[df_0.75$nsim]

set.seed(1)
m_control <- SuperLearner(data$cd420[data$treat == 0], newdata2[data$treat == 0, -c(1
,14)],  SL.library = "SL.rpart" )
set.seed(1)
m<-rpart(cd420~., data= newdata[treat == 1, -c(14)])

rpart.plot(m)

#random forest
model_fitting(newdata2, "SL.rf.rforest.4.100" )
model_fitting(newdata2, "SL.rf.rforest.4.500" )
model_fitting(newdata2, "SL.rf.rforest.4.1000" )

model_fitting(newdata2, "SL.rf.rforest.3.100" )
model_fitting(newdata2, "SL.rf.rforest.3.500" )
model_fitting(newdata2, "SL.rf.rforest.3.1000" )
```