Estimating the average treatment effect using the cluster hierarchy and merge post-stratification method

by

Kingsley Darko

B.S., Kwame Nkrumah University of Science and Technology, 2018

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Approved by:

Major Professor
Michael Higgins

Copyright

Abstract

Randomized experiments help reduce bias in estimates of the average treatment effect by ensuring that confounders have the same distribution across treatment groups. However, some randomizations can still have imbalances on important confounders, which can lead to inaccurate estimates. Post-stratification is one method for correcting these imbalances to improve estimates. In post-stratification, we form groups of units, called strata, and estimate the overall treatment effect by taking a weighted average of treatment effects within each stratum. In practice, strata are formed based on the values of the confounders. We examine the ad-hoc post-stratification method, where we form groups of units so that every group has at least one treated and control unit. A sufficient condition for the unbiasedness of post-stratification estimators is treatment assignment symmetry—that conditioned on the number of treated units within each stratum, each treatment assignment is equally likely. However, ensuring that each stratum has at least one treatment status often violates assignment symmetry and leads to biased estimates. This report considers a new method for forming strata— cluster hierarchy and merge post-stratification (CHAMP)—that ensures that each treatment status is represented within each stratum and satisfies a weaker form of assignment symmetry required for unbiased estimation. We perform a simulation study to compare CHAMP post-stratification with ad-hoc methods for forming strata. We show that CHAMP post-stratification successfully eliminates bias while ensuring small standard errors of post-stratification estimators. Finally, we apply our method to the Study to Understand Prognoses and Preferences for Outcomes and Risks and Treatments (SUPPORT) dataset to assess the efficacy of right heart catheterization in the initial care of critically ill patients.

Table of Contents

List of Figures vi

List of Tables vii

1 Introduction 1

1.1 Post-stratification 3

1.1.1 Stratification 3

1.1.2 Description of post-stratification 3

1.1.3 Steps of Post-stratification 4

1.2 Literature Review on Post-stratification 5

1.3 Neyman-Rubin Causal Model 7

2 Assumptions and Estimators 10

2.1 Estimation of Treatment Effects under Post-stratification 10

2.2 Performance of Estimators 13

3 Ad-hoc Post-stratification and CHAMP Post-stratification 15

3.1 Methods of Post-stratification 15

3.2 Ad-hoc Post-stratification 15

3.2.1 Left-to-Right Ad-hoc Post-stratification 16

3.2.2 Right-to-Left Ad-hoc Post-stratification 16

3.2.3 Ends-to-Center Ad-hoc Post-stratification 17

3.2.4 Illustrative Numerical Examples of Ad-hoc Post-stratification 18

3.3 Cluster Hierarchy and Merge Post-stratification 21
3.4 Simulations and Results 24
3.5 Special Cases 26
3.6 Discussion on the Simulations and Cases 31
3.7 Right Heart Catheterization (RHC) 31
3.7.1 Description of SUPPORT Data 32
3.8 CHAMP Estimator on RHC 33
3.9 Discussion 34

4 Conclusion 35
4.1 Future Work 36

Bibliography 37

A R code for Ad-hoc Post-Stratification Estimator 41

B R-Code for CHAMP Post-Stratification 56

List of Figures

1.1 Stratification 4

1.2 Post-stratification 5

3.1 Left to Right Ad-hoc Post-stratification 17

3.2 Right to Left Ad-hoc Post-stratification 18

3.3 Ends-to-Center Ad-hoc Post-stratification 19

3.4 CHAMP Post-stratification 23

List of Tables

1.1 Only one potential outcome for each unit can be observed. The other potential outcome is missing. The unit's treatment effect, τ_i, is not estimable because of the missing potential outcomes. However, the average treatment effect, τ, can be estimated, for example, using the difference in sample means, $\hat{\tau} = \frac{1}{4}\{(y_1(1) + y_2(1) + y_3(1) + y_4(1)\} - \frac{1}{4}\{(y_5(0) + y_6(0) + y_7(0) + y_8(0)\}$. Here, $\square$ represents the unobserved potential outcomes. 8

3.1 Units in the potential outcomes under treatment are greater than the corresponding units under control. We have unbiased estimates for this set of potential outcomes and the estimates are the same for all the types of the ad-hoc post-stratification. 19

3.2 The potential outcomes under treatment are the same for the corresponding outcomes under control. There is not much difference with this set of potential outcomes. 20

3.3 The potential outcomes under treatment are the same for the corresponding outcomes under control. We can see some disparities in the estimates and they are as a result of the direction of the outcomes. 20

3.4 Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate. 24

3.5 Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate. 25

3.6 Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the Ends-to-Center gives us a minimum bias. The CHAMP has an unbiased estimate. 25

3.7 Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate. 26

3.8 Special Cases 26

3.9 Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate. 27

3.10 Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. The above ad-hoc post-stratification provided estimates with smaller bias than the ends to center ad-hoc post-stratification. The CHAMP has an unbiased estimate. 27

3.11 Left-to-Right ad-hoc post-stratification estimator provided estimates with smaller bias than the ends to center ad-hoc post-stratification. The CHAMP has an unbiased estimate. 28

3.12 Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. The above ad-hoc post-stratification provided estimates with smaller bias than the ends to center ad-hoc post-stratification. The CHAMP has an unbiased estimate. 28
3.13 Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimates, and gave us minimum bias. The CHAMP has an unbiased estimate. 29
3.14 Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate. 29
3.15 Left-to-Right Ad-hoc post-stratification gives us a minimum bias. The CHAMP has an unbiased estimate. 30
3.16 Right to Left ad-hoc post-stratification provides us with a minimum bias. The CHAMP has an unbiased estimate. 30
3.17 Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate. 31
3.18 The means in the treatment groups were significantly different from zero with a p-value of <2.2e-16, indicating an imbalance 34

Chapter 1

Introduction

The causal effect of an action is the difference in responses or outcomes when a unit receives treatment and when it does not. There may be some independent variables whose values may influence outcomes and can be correlated with treatment—these are known as confounders (Bauman et al., 2002). For example, in assessing a regimen for treating coronavirus, where old people tend to have more severe cases of the coronavirus than young people, age is a confounding variable. Confounders can cause bias in estimates when there are imbalances in treatment groups.

Randomization has become an increasingly important tool most statisticians and analysts use to make inferences inference and estimate treatment effects since the time (Fisher, 1936) introduced the idea. A randomized experiment is a process where treatments are randomized to units,and can ensure that the distribution of confounders is the same across treatment groups.

However, imbalances on confounders can still occur even when treatment is randomized across units.One way of reducing imbalances in the design of the experiment is statistical blocking. This is a process where researchers partition a sample of units into disjoint sets of units called blocks (Higgins et al., 2016). After experimenters have stratified units, they randomize treatments to the units within their predefined blocks (Miratrix et al., 2013).

The process of blocking removes variability of treatment effect estimates (Krzywinski and Altman, 2014). There are instances where blocking may not be feasible, because blocking is a technique that is done before randomization. This implies blocking lacks foresight and there is no blocking after randomization (Miratrix et al., 2013). Mostly, researchers rely on covariates adjustments after randomization in instances where blocking cannot be performed (Keele et al., 2010; Pocock et al., 2002). Post-stratification is another way to perform an adjustment.

Post-stratification is a process where a researcher stratifies units based on confounders and finds a weighted average of the strata-level treatment effect estimates. Post-stratification is used to adjust for treatment effect estimates after randomization is performed. Post-stratification is similar to blocking in the sense that both improve precision. Also, if the number of treated units in each stratum are predefined and are fixed, then the estimate of the treatment effects by blocking will be the same as the treatment effects estimate by post-stratification (Miratrix et al., 2013). When post-stratification is performed properly, imbalance between treatment groups can be mitigated, leading to increased precision in treatment effect estimates.

Post-stratification estimators require each treatment status to be represented within each stratum. Unfortunately, current methods for enforcing this condition leads to a violation of the *assignment symmetry*—that conditional on the number of each treatment status within each stratum, each randomization is equally likely to occur (Miratrix et al., 2013) This can inject bias into the post-stratification estimator.

In this report, we investigate some ad-hoc methods for forming strata and measure their bias, standard errors and MSE under various models of response. We then perform a simulation study to compare the ad-hoc methods to a proposed method for stratification, called Cluster Hierarchy and Merge Post-stratification (CHAMP), that is designed to eliminate the bias that is present in the ad-hoc methods. We give evidence that CHAMP removes the bias in post-stratification estimators while ensuring competitive standard errors when compared

to the ad-hoc methods.

1.1 Post-stratification

Post-stratification is a process where a researcher stratifies units based on confounders and finds a weighted average of the strata-level treatment effect estimates (Miratrix et al., 2013). Post-stratification of an experiment is similar to stratification of a sample, so we first begin with a description of stratification.

1.1.1 Stratification

Stratification is a sampling technique where a population is divided into sub-populations called strata and samples are drawn independently across strata (Miratrix et al., 2013).

Stratification provides samples that represent major subgroups of the population and improves precision of estimators. There are individual responses of people which are expected to vary with age and similar factors but are not available at the time of stratification prior to sampling. After sampling, estimates of the population totals in every category are obtained from known census totals if the individual units are classified according to the categorical covariates. The aggregate of these estimates in every category produces the overall population estimate.

We divide population of N units into H homogeneous groups called strata. Each of the h^{th} stratum would consist of N_h units. We draw a sample of size n_h from the N_h units (Singh, 2003). Figure 1.1 gives a visual depiction of stratification.

1.1.2 Description of post-stratification

Post-stratification in simple terms is stratification after performing a sample or experiment. Post-stratification is a way of factoring in covariates to improve estimates in a way that looks like stratification.

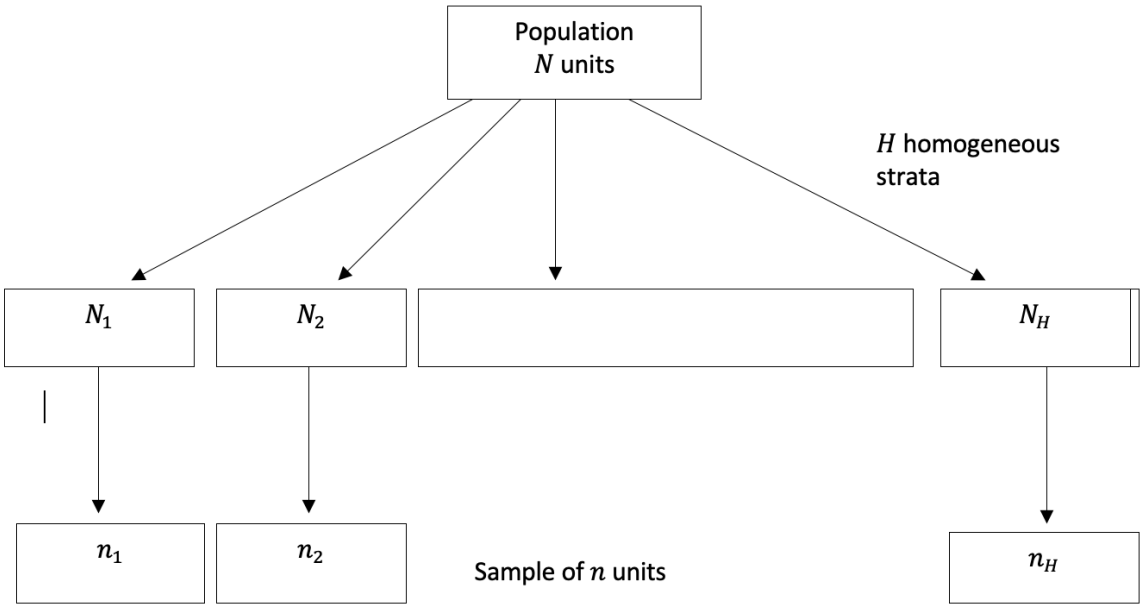


Figure 1.1: Stratification

1.1.3 Steps of Post-stratification

Post-stratification is performed as follows. We take a sample of n units from the population of N units using a chosen sampling technique. We divide the population into H strata, where units within each stratum have similar values of confounding covariates. The values of N_h, where $h = 1, 2, ..., H$ and $\sum_{h=1}^{H} N_h = N$ may or may not be known. We select each sample unit and place them in the h^{th} stratum based on the covariates such that $\sum_{h=1}^{H} n_h = n$, where $h = 1, 2, ..., H$ (Singh, 2003). Figure 1.2 gives a visual depiction of stratification.

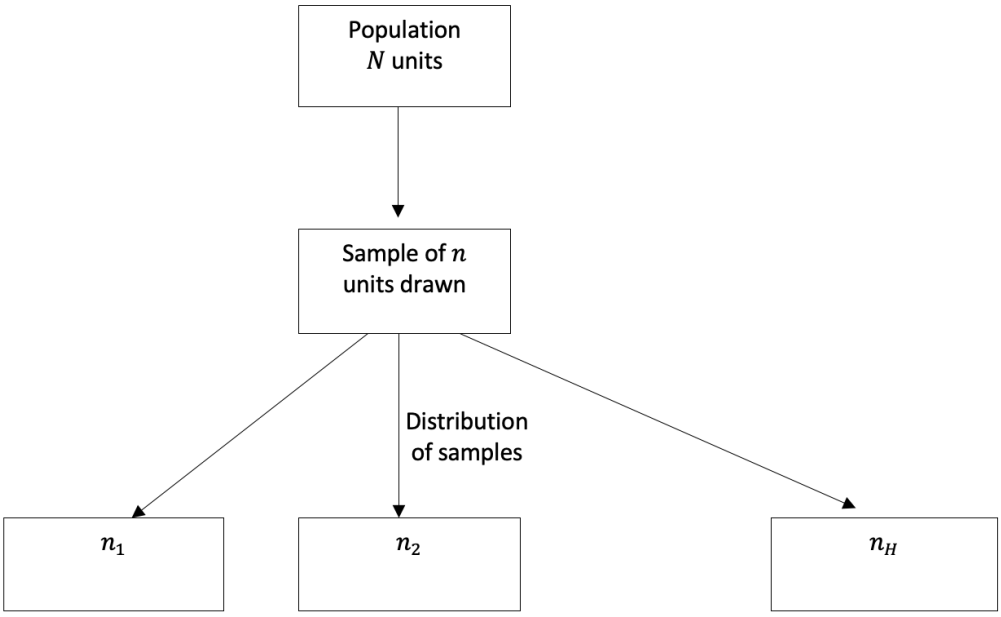


Figure 1.2: Post-stratification

1.2 Literature Review on Post-stratification

We now review some papers applying post-stratification methods.

(Valliant, 1993) pointed out that in complex large-scale surveys, particularly household surveys, post-stratification is a sampling technique that is used to improve the efficiency of estimators. Although inferences are better when they are conditioned on the achieved post-stratum sizes, he used a design-based conditional theory to make inferences from post-stratified samples. Much concentration was given on the properties of several commonly used variance estimators to determine whether they estimate the conditional variance of the post-stratified estimator of the finite population model. He did a simulation study of the theory using a fixed, finite population of 10,841 persons included in September 1988 Current Population Survey (CPS). Weekly wages and hours worked per week were considered as the variables. 2886 geographic segments which composed of about 4 neighbouring households were included in the study. Eight post-strata were formed based off age, race and sex. A two-stage stratified sample was used in which segments were selected as the first-stage

units and persons as the second-stage units. In both cases, the strata had same number of households. Within each stratum, segments were systematically selected and a random sample of four persons was selected without replacement in each segment. It was concluded that post-stratification is an important sampling technique that helps in reducing variance and also reducing bias of an estimator.

(Little, 1993) developed a Bayesian model-based theory for post-stratification instead of the conventional method of randomization where inference is based on sampling distribution whilst holding population values fixed. He talked about the difference between stratification and post-stratification, *i.e.*, stratified sampling is limited to variables that are known for survey units before data is collected whereas post-stratification combines data collected in the survey with aggregate data on the population from other sources. He used an example to make that difference clear. For instance, a demographic survey generally cannot stratify on age, because the ages of individuals are unavailable until after the interview is conducted. However, the population age distribution may be available in aggregate form from census data. This is when post-stratification comes into play. Post-stratification classifies the sample by age groups and weights individuals in each group up to the sub-population total. He talks about a usual estimator of $\bar{Y}$ as the post-stratified mean or weighted mean. Let Z denote the post-stratifying variable, let Y denote a survey variable, and consider inference for the finite population mean $\bar{Y} = \sum_h \mathcal{P}_h \bar{Y}_h$, where Y_h is the mean in post-stratum. Suppose that a simple random sample of size n is selected,r of which respond to the survey and Let n_h and $r_h \leq n_h$ denote the number sampled and the number responding in post-stratum h. Write $\mathcal{P} = \{\mathcal{P}_1, .., \mathcal{P}_H\}$, $n = \{n_l, ..., n_H\}$, and $r = \{r_i, ...r_H\}$. Then P and r are assumed known.

$$\bar{y}_{ps} = \sum_{h=1}^{H} \mathcal{P}_h \bar{y}_h = \frac{1}{r} \sum_{i=1}^{r} w_i y_i,$$

where y_i is the value of Y for respondent i, $\bar{y}_h$ is the respondent sample mean in post-stratum h, and $w_i = r\frac{\mathcal{P}_h}{r_h}$ if $z_i = h$; that is, case i belongs to stratum h.

1.3 Neyman-Rubin Causal Model

Randomized experiments are regarded as the "gold standard" for performing causal inference. In designing an experiment, Fisher mentioned that randomization of treatments in experiments is the basis for causal inference (Box, 1980; Fisher, 1992). Neyman also recognized the works of Fisher and developed an important model between the relationship of treatment assignment and causal inference, popularly known as the Neyman-Rubin Causal Model (NRCM).

The Neyman-Rubin non-parametric model of potential outcomes, which serves as the basis of our estimation method, has become an increasingly popular model that has received a lot of attention in the fields of Statistics (Rosenbaum, 2002); (Holland, 1986); (Rubin, 1974, 2006), Medicine (Christakis and Iwashyna, 2003); (Rubin, 1997), Political Science (Bowers and Hansen, 2005); (Sekhon, 2004, 2009); (Imai, 2005), Economics (Dehejia and Wahba, 1999, 2002); (Abadie and Imbens, 2006), and Sociology (Smith, 1997); (Morgan and Harding, 2006).

Assume we have n experimental units, and treatments are completely randomized to the n units. We denote T_i as the treatment assignment indicator for unit i, thus $T_i = 1$ if unit i receives treatment, and $T_i = 0$ if unit i receives control.

We assume the Neyman-Rubin model (Holland, 1986; Rubin, 1974) of response. We consider $y_i(1) \in \mathbb{R}$ to be unit i's outcome if it were treated, and $y_i(0)$ to be its outcome if it were given control. We call these the *potential outcomes* of unit i. For each unit, we observe either $y_i(1)$ or $y_i(0)$ depending on whether we treat it or not. We also assume that the potential outcomes are not random. The treatment effect on unit i is defined as

$$\tau_i = y_i(1) - y_i(0). \tag{1.1}$$

The average treatment effect (ATE) is defined as

$$\tau = \frac{1}{n}\sum_i \{y_i(1) - y_i(0)\} = \frac{1}{n}\sum_i \tau_i \qquad i = 1, ..., n. \tag{1.2}$$

Note that, even though unit level treatment effects are not estimable, we can still estimate the ATE, for example, with the difference in sample means. See Table 1.1 for details.

Illustration						
Unit	Potential			Observed		
i	Treatment	Control	τ_i	T_i	Y_i	τ_i
1	$y_1(1)$	$y_1(0)$	$y_1(1)$ - $y_1(0)$	1	$y_1(1)$	$y_1(1)$- □
2	$y_2(1)$	$y_2(0)$	$y_2(1)$ - $y_2(0)$	1	$y_2(1)$	$y_2(1)$- □
3	$y_3(1)$	$y_3(0)$	$y_3(1)$ - $y_3(0)$	1	$y_3(1)$	$y_3(1)$- □
4	$y_4(1)$	$y_4(0)$	$y_4(1)$ - $y_4(0)$	1	$y_4(1)$	$y_4(1)$- □
5	$y_5(1)$	$y_5(0)$	$y_5(1)$ - $y_5(0)$	0	$y_1(0)$	□ - $y_1(0)$
6	$y_6(1)$	$y_6(0)$	$y_6(1)$ - $y_6(0)$	0	$y_2(0)$	□ - $y_2(0)$
7	$y_7(1)$	$y_7(0)$	$y_7(1)$ - $y_7(0)$	0	$y_3(0)$	□ - $y_3(0)$
8	$y_8(1)$	$y_8(0)$	$y_8(1)$ - $y_8(0)$	0	$y_4(0)$	□ - $y_4(0)$
ATE	$\tau = \frac{1}{8}\{(y_1(1) - y_1(0)) + ... + (y_8(1) - y_8(0))\}$			$\hat{\tau} = \frac{1}{4}\{(y_1(1) + ... + y_4(1)\} - \frac{1}{4}\{(y_5(0) + ... + y_8(0)\}$		

Table 1.1: Only one potential outcome for each unit can be observed. The other potential outcome is missing. The unit's treatment effect, τ_i, is not estimable because of the missing potential outcomes. However, the average treatment effect, τ, can be estimated, for example, using the difference in sample means, $\hat{\tau} = \frac{1}{4}\{(y_1(1) + y_2(1) + y_3(1) + y_4(1)\} - \frac{1}{4}\{(y_5(0) + y_6(0) + y_7(0) + y_8(0)\}$. Here, □ represents the unobserved potential outcomes.

If we assign treatments to units, we cannot observe both treatment and control for a particular unit. This means that if we treat a unit, we would know the outcome under the treatment, but would not know its outcome if it were not treated and vice-versa. This is the fundamental problem of causal inference (Holland, 1986). The hypothetical unobserved outcome for a unit is called the counterfactual. In causal inference, we find estimates of the treatment effects by using available outcomes to estimate counterfactual.

The NRCM states that the observed response Y_i only depends on the potential outcomes of i and the treatment given to unit i,

$$Y_i = T_i y_i(1) + (1 - T_i) y_i(0), \tag{1.3}$$

where the potential outcomes $y_i(1)$ and $y_i(0)$ are fixed, but Y_i is random because of the

treatment assignment indicator T_i.

The NRCM does not need any distributional assumptions. However, it should be noted that randomization does not give us a certainty that the observation on one unit should not be affected by the particular assignment of treatments to other units (Cox, 1958). Thus, inherent in the Neyman Rubin Model of response is the stable unit treatment value assumption (SUTVA). SUTVA implies that treatments applied to one unit do not affect the outcome for another unit (Rubin, 1978). SUTVA is also called the "no interference" assumption between units.

Chapter 2

Assumptions and Estimators

2.1 Estimation of Treatment Effects under Post-stratification

Define $b_i \in \mathcal{B}$ to be the covariate which are observed for all units and are unaffected by treatment. Post-stratification can greatly reduce variation in the proportions of the units treated over using a simple difference-in-means estimator.

The classic unadjusted estimator $\hat{\tau}_{sd}$ (Miratrix et al., 2013) is the observed simple difference in the means of the treatment and control groups and it is given by:

$$\hat{\tau}_{sd} = \sum_{i=1}^{n} \frac{T_i}{W(1)} Y_i - \sum_{i=1}^{n} \frac{(1-T_i)}{W(0)} Y_i. \tag{2.1}$$

We know the quantities T_i and Y_i from Equation 1.3, but $W(1) = \sum_i T_i$ is the total number of treated units, $W(0)$ is total control and $W(1) + W(0) = n$. The strata that are defined by the levels of b have stratum-specific ATE_k (Miratrix et al., 2013):

$$\tau_k \equiv \frac{1}{n_k} \sum_{i:b_i=k} \{y_i(1) - y_i(0)\} \qquad k = 1, ..., K, \tag{2.2}$$

where n_k is the number of units in stratum k . The overall ATE can then be expressed as a weighted average of these $ATE_k s$:

$$\tau = \sum_{k=1}^{K} \frac{n_k}{n} \tau_k. \tag{2.3}$$

The simple difference estimator can be used to estimate $ATE_k s$ for each stratum k:

$$\hat{\tau}_k = \sum_{i:b_i=k} \frac{T_i}{W_k(1)} y_i(1) - \sum_{i:b_i=k} \frac{(1-T_i)}{W_k(0)} y_i(0), \tag{2.4}$$

where $W_k(1)$ and $W_k(0)$ are the number of treated and control units in stratum k respectively.

A post-stratification estimator is an appropriately weighted estimate of these strata level estimates:

$$\hat{\tau}_{ps} = \sum_{k \in \mathcal{B}} \frac{n_k}{n} \hat{\tau}_k. \tag{2.5}$$

For post-stratification estimators to estimate the ATE unbiasedly, it is sufficient for the randomization of treatment to units to satisfy the *treatment assignment symmetry assumption*. (Miratrix et al., 2013) defined that a randomization is assignment symmetric if it satisfies the following two assumptions:

Assumption 1. *Equiprobable treatment assignment patterns: all* $\binom{n_k}{W_k(1)}$ *ways to treat* $W_k(1)$ *units in stratum* k *are equiprobable, given* $W_k(1)$.

Assumption 2. *Independent treatment assignment patterns: for all strata* j *and* k, *with* $j \neq k$, *the treatment assignment pattern in stratum* j *is independent of the treatment assignment pattern in stratum* k, *given* $W_j(1)$ *and* $W_k(1)$.

Theorem 1. *The strata level estimators* $\hat{\tau}_k$ *are unbiased, i.e.*

$$\mathbb{E}[\hat{\tau}_k] = \tau_k \qquad\qquad k = 1, ..., K.$$

Proof.

$$\begin{aligned}
\mathbb{E}[\hat{\tau}_k] &= \mathbb{E}[\sum_{i:b_i=k} \frac{T_i}{W_k(1)} y_i(1) - \sum_{i:b_i=k} \frac{(1-T_i)}{W_k(0)} y_i(0)] \\
&= \mathbb{E}[\sum_{i:b_i=k} \frac{T_i}{W_k(1)} y_i(1)] - \mathbb{E}[\sum_{i:b_i=k} \frac{(1-T_i)}{W_k(0)} y_i(0)] \\
&= \sum_{i:b_i=k} \mathbb{E}[\frac{T_i}{W_k(1)}] y_i(1) - \sum_{i:b_i=k} \mathbb{E}[\frac{(1-T_i)}{W_k(0)}] y_i(0) \\
\text{Note that } \mathbb{E}[\frac{T_i}{W_k(1)}] &= \mathbb{E}[\mathbb{E}[\frac{T_i}{W_k(1)} \mid W_k(1)]] = \mathbb{E}[\frac{W_k(1)}{W_k(1) * n_k}] = \frac{1}{n_k} \\
\text{Similarly } \mathbb{E}[\frac{1-T_i}{W_k(0)}] &= \frac{1}{n_k} \\
\text{So } \mathbb{E}[\hat{\tau}_k] &= \sum_{i:b_i=k} \frac{1}{n_k} y_i(1) - \sum_{i:b_i=k} \frac{1}{n_k} y_i(0) \\
&= \tau
\end{aligned}$$

□

Corollary 1. *The unadjusted simple difference estimator $\hat{\tau}_{sd}$ is*

$$\mathbb{E}[\hat{\tau}_{sd}] = \tau$$

Theorem 2. *The post-stratification estimator $\hat{\tau}_{ps}$ is unbiased, i.e.*

$$\mathbb{E}[\hat{\tau}_{ps}] = \tau \qquad\qquad k = 1, ..., K.$$

Proof.

$$\text{From Equation 2.5 }, \mathbb{E}[\hat{\tau}_{ps}] = \mathbb{E}[\sum_{k\in\mathcal{B}} \frac{n_k}{n}\hat{\tau}_k]$$
$$= \sum_{k\in\mathcal{B}} \frac{n_k}{n}\mathbb{E}[\hat{\tau}_k]$$
$$\text{From Theorem 1, } \mathbb{E}[\hat{\tau}_k] = \tau_k$$
$$\mathbb{E}[\hat{\tau}_{ps}] = \sum_{k\in\mathcal{B}} \frac{n_k}{n}\tau_k$$
$$\text{From Equation 2.3, } \mathbb{E}[\hat{\tau}_{ps}] = \tau.$$

□

2.2 Performance of Estimators

We will judge the performance of the estimators based on these statistical properties. Suppose there is an estimator $\hat{\beta}$ of parameter β.

Definition 1. *(Bias)*

The bias of an estimator $\hat{\beta}$ is

$$bias(\hat{\beta}) = \mathbb{E}(\hat{\beta}) - \beta. \tag{2.6}$$

The estimator $\hat{\beta}$ is unbiased for β if bias($\hat{\beta}$)=0.

Definition 2. *(Variance)*

The variance of an estimator $\hat{\beta}$ is

$$var(\hat{\beta}) = \mathbb{E}(\hat{\beta} - \mathbb{E}(\hat{\beta}))^2. \tag{2.7}$$

The standard error of $\hat{\beta}$ is $\sqrt{var(\hat{\beta})}$.

Definition 3. *(Mean squared error)*

The mean squared error of an estimator $\hat{\beta}$ *is*

$$MSE(\hat{\beta}) = \mathbb{E}(\hat{\beta} - \beta)^2 = var(\hat{\beta}) + bias(\hat{\beta})^2. \tag{2.8}$$

Bias is basically the differences between the expected value of a measurement and the true value of the parameter being estimated. Unlike bias, the magnitude of precision is solely contingent on the estimated (or observed) values and is entirely independent of the actual value. Precision measures the variance of an estimation technique (West, 1999). Bias and precision in tandem define the performance of an estimator. If an estimator is more biased and less precise, its overall ability is worst to make an accurate point estimation. Accuracy is the total distance between the observed value and the true value (Bainbridge, 1985). We use the standard deviation to measure precision and the Mean squared error to measure the estimators accuracy. The standard deviation is used in lieu of the variance, because standard deviation is on the same scale as the mean and directly comparable.

Chapter 3

Ad-hoc Post-stratification and CHAMP Post-stratification

3.1 Methods of Post-stratification

In post-stratification, units with similar values of the confounder are grouped together before performing estimation. In this report, we will evaluate different ways of performing this post-stratification. We will consider Ad-hoc post-stratification and CHAMP Post-stratification.

3.2 Ad-hoc Post-stratification

It is infeasible to ensure that randomness is contained within the set of assignments if we stratify after randomization. If we form strata before treatments are randomized, we can perform statistical blocking. This means that, post-stratification will be a very useful ingredient in estimating treatment effects when we stratify after randomization. Thus, in practice, it is often not possible for stratification and randomization to satisfy the assignment symmetry assumption necessary for unbiased estimation.

We now detail a few *ad-hoc* methods for which strata can be formed given the randomization. These methods will ensure that each stratum contains one of each treatment

condition.

1. With ad-hoc post-stratification, we form groups of units so that each group has at least one treated and one control unit.

2. We group units with similar values of categorical covariates.

3. In the case of one confounding covariate, we can sort the units based on values of this covariate. We then start from one end of the covariate space and add units to the stratum until it has at least one treated and one control unit.

4. Continue adding units to groups until there are no more units. If the last group does not contain one unit from each treatment condition, add it to the second-to-last group.

3.2.1 Left-to-Right Ad-hoc Post-stratification

In left-to-right post-stratification, we start from the smallest value of the covariate and add units to strata in increasing order of the covariate. Figure 3.1 demonstrates a left-to-right post-stratification on eight units. Left-to-right post-stratification proceeds as follows.

1. We start from the left end of the covariate space and add units from the smallest to the largest to a group until group has at least one treated and one control unit.

2. Continue adding units to groups until there are no more units. If the last group does not contain one unit from each treatment condition, add it to the second-to-last group.

3.2.2 Right-to-Left Ad-hoc Post-stratification

In right-to-left post-stratification, we start from the largest value of the covariate and add units to strata in decreasing order of the covariate. Figure 3.2 demonstrates a left-to-right post-stratification on eight units. Right-to-left post-stratification proceeds as follows.

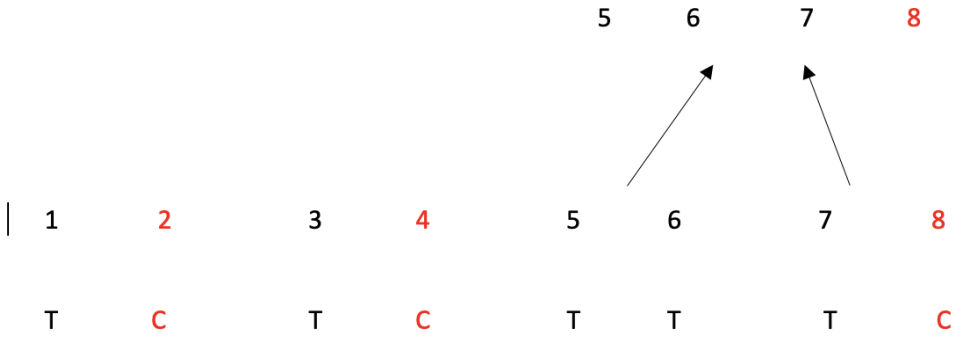


Figure 3.1: Left to Right Ad-hoc Post-stratification

1. We start from the right end of the covariate space and add units from the largest to the smallest to a group until group has at least one treated and one control unit.

2. Continue adding units to groups until there are no more units. If the last group does not contain one unit from each treatment condition, add it to the second-to-last group. In Figure 3.2, unit 1 was the remainder, so we add it to the group containing units 2 and 3.

3.2.3 Ends-to-Center Ad-hoc Post-stratification

In ends-to-center post-stratification, we alternate between left-to-right post-stratification and right-to-left post-stratification, working our way towards the center of the covariate distribution. Figure 3.3 demonstrates a ends-to-center post-stratification on eight units. Ends-to-center post-stratification proceeds as follows.

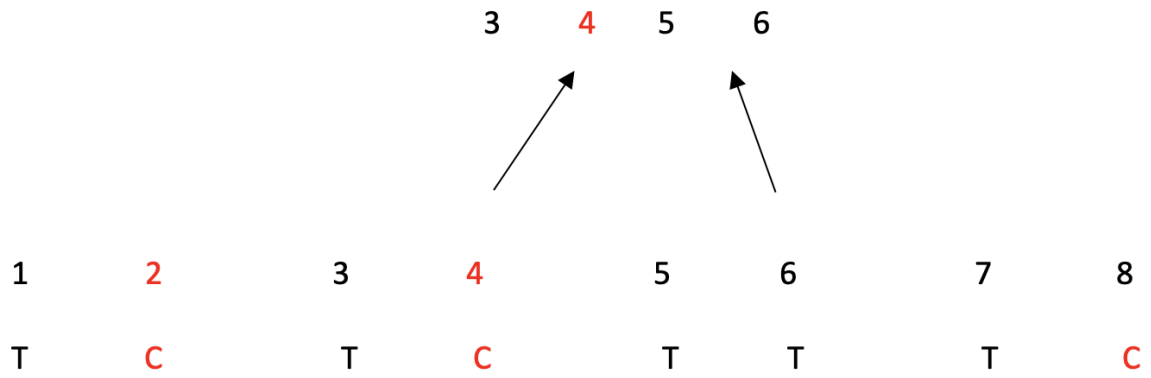


Figure 3.2: Right to Left Ad-hoc Post-stratification

1. We start from the ends of the covariate space and add units to a group until that group has at least one treated and one control unit.

2. Continue adding units to groups until there are no more units. If the last group does not contain one unit from each treatment condition, add it to the second-to-last group. In Figure 3.3, units 5 and 6 were the remainder, so we add it to the group containing units 3 and 4.

3.2.4 Illustrative Numerical Examples of Ad-hoc Post-stratification

We now consider the performance of these ad-hoc estimators under various scenarios. For each example, we define the potential outcomes under treated and control where these potential outcomes are specified for each of the n units under study.

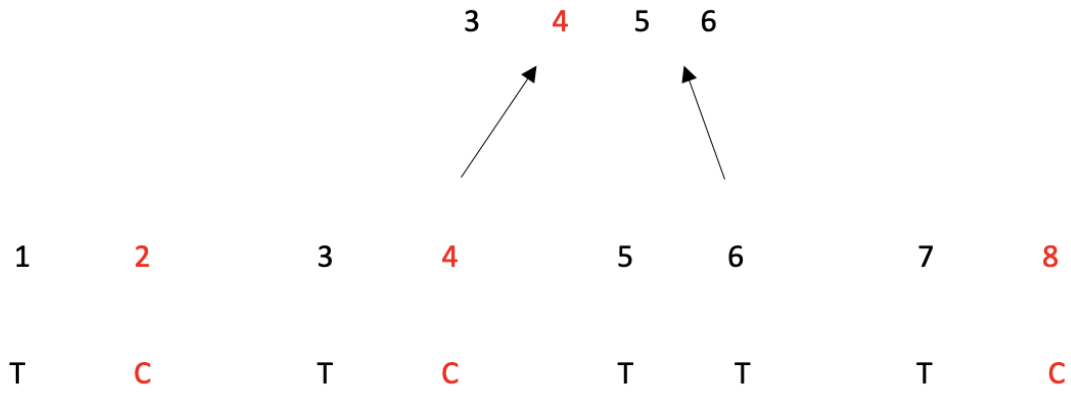


Figure 3.3: Ends-to-Center Ad-hoc Post-stratification

For example,...

Example 1: We consider every unit in the potential outcomes to be the same, but units in the potential outcomes under treatment are greater than the corresponding units under control. For this simple case, post-stratification estimators should be able to estimate the ATE unbiasedly, and this is true for the three ad-hoc estimators considered.

Consider y_1 = (4,4...,4); y_0 = (1,1...,1); $n = 6$, and $W(1) = 3$.

ATE = 3

Estimator	n	$Estimate$	$StandardError$	$Bias$
Adhoc:Left	6	3	0	0
Adhoc:Right	6	3	0	0
Adhoc:Center	6	3	0	0

Table 3.1: Units in the potential outcomes under treatment are greater than the corresponding units under control. We have unbiased estimates for this set of potential outcomes and the estimates are the same for all the types of the ad-hoc post-stratification.

Example 2: We consider every unit in the potential outcomes to be the same, and units in the potential outcomes under treatment are the same for the corresponding units under control. There are no differences in the estimates and the deviations are the same for each ad-hoc estimator.

Consider y_1 = (1,2...,6); y_0 = (1,2...,6); $n = 6$, and $W(1) = 3$.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$
Adhoc:Left	6	0	1.301	0
Adhoc:Right	6	0	1.301	0
Adhoc:Center	6	0	1.301	0

Table 3.2: The potential outcomes under treatment are the same for the corresponding outcomes under control. There is not much difference with this set of potential outcomes.

Example 3: We consider every unit in the potential outcomes under treatment to be in ascending order of magnitude and the corresponding units under control to be in descending order. There were some slight differences in the estimate. Also, the standard errors were the same.

Consider y_1 = (1,2...,6); y_0 = (6,5,...,1); $n = 6$, and $W(1) = 3$.

ATE = 0

Estimator	n	$Estimate$	SD	$Bias$
Adhoc:Left	6	0.075	0.160	0.075
Adhoc:Right	6	-0.075	0.160	-0.075
Adhoc:Center	6	0.075	0.160	0.075

Table 3.3: The potential outcomes under treatment are the same for the corresponding outcomes under control. We can see some disparities in the estimates and they are as a result of the direction of the outcomes.

3.3 Cluster Hierarchy and Merge Post-stratification

A sufficient condition for unbiasedness of the post-stratification estimator, the strata level estimators, and the simple difference in means estimator is treatment assignment symmetry. This assumption ensures that

$$\mathbb{E}\left[\frac{T_i}{W_k(1)}\right] = \frac{1}{n_k} \qquad k = 1, ..., K, \tag{3.1}$$

Equation (3.1) is also sufficient for unbiased post-stratification estimators.

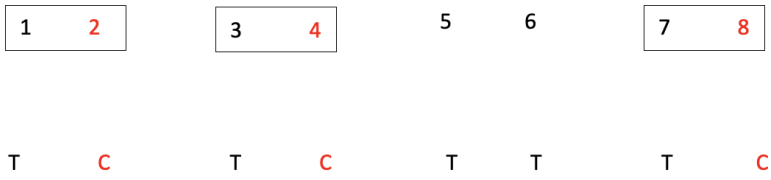
We propose a new post-stratification procedure called Cluster Hierarchy and Merge Post-stratification (CHAMP) that is designed to satisfy (3.1) without assignment symmetry. Thus, this proposed method should allow for unbiased estimation of the average treatment effect. CHAMP post-stratification proceeds as follows.

1. Construct the following cluster hierarchy.
   (a) Stratify the population into k_1 clusters of equal size, where n is divisible by k_1. Call these the level 1 strata.
   (b) Stratify these clusters into k_2 clusters of equal size, where k_1 is divisible by k_2. Call these the level 2 strata.
   (c) Continue this stratification into clusters until you are left with one strata containing all n units.
2. Form the following post-stratification.
   (a) Begin with the stratification into k_1 clusters.
   (b) If a level 1 stratum s_1 does not have at least one treated and control unit, find the level 2 stratum s_2 that contains s_1 and merge all units belonging to s_2 into a single stratum.

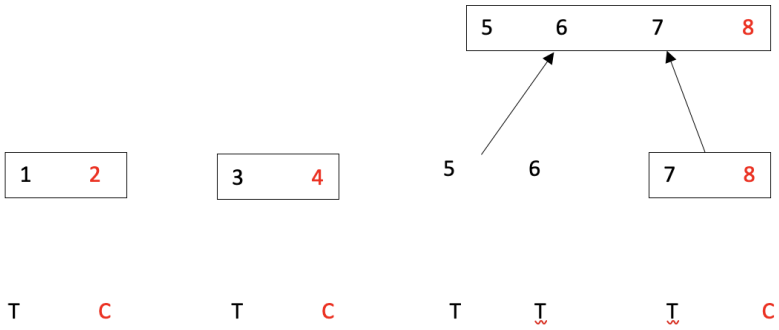
(c) Continue this merging until all strata in the post-stratification contain at least one treated and one control unit.

To ensure that post-stratification improves estimation precision, the cluster hierarchy should be built to ensure that units with similar values of confounding covariates belong to the same cluster. For example, in the case of one confounding covariate, units can be sorted in increasing order of that covariate and stratified with respect to that ordering as in the ad-hoc methods.

Also note that we can require strata to have arbitrarily many treated and control units. However, by ensuring at least one treated and one control unit belongs to each strata, we are guaranteed that the post-stratification estimator is well defined.



(a) One of the clusters does not contain at least one treated and control,i.e. We do not merge units 5 and 6 since both of them are treated.



(b) We will join units 5 and 6 to the closest neighbours which are 7 and 8

Figure 3.4: CHAMP Post-stratification

3.4 Simulations and Results

We run a simulation of 10000 units. We choose sample of sizes 8, 16, 100 and 500. We consider equal number of treated and control units in each case. We will also consider every unit in the potential outcomes under treatment to be in ascending order of magnitude and the corresponding units under control to be in descending order. For every estimator, we will the estimates, the standard errors, the bias and the Mean Squared Error (MSE).

Simulation 1:

Consider y_1 = (1,2...,8); y_0 = (8,7,...,1); $n = 8$, and $W(1) = 4$.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	8	0.153	0.057	0.153	0.080
Adhoc:Right	8	-0.153	0.348	-0.153	0.080
Adhoc:Center	8	0.068	0.350	0.068	0.072
Simple Difference	8	0.00	0.00	0.00	0.00
CHAMP	8	0	0.356	0	0.127

Table 3.4: Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate.

Simulation 2:

Consider y_1 = (1,2...,16); y_0 = (16,15,...,1); $n = 16$, and $W(1) = 8$.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	16	0.38	0.35	0.38	0.26
Adhoc:Right	16	-0.38	0.35	-0.38	0.26
Adhoc:Center	16	0.05	0.38	0.05	0.14
Simple Difference	16	0.00	0.00	0.00	0.00
CHAMP	16	0.00	0.53	0.00	0.28

Table 3.5: Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate.

Simulation 3:

Consider y_1 = (1,2...,100); y_0 = (100,99,...,1); $n = 100$, and $W(1) = 50$.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	100	0.74	0.23	0.74	0.60
Adhoc:Right	100	-0.74	0.23	-0.74	0.60
Adhoc:Center	100	0.01	0.23	0.01	0.08
Simple Difference	100	0.00	0.00	0.00	0.00
CHAMP	100	0.00	0.27	0.00	0.07

Table 3.6: Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the Ends-to-Center gives us a minimum bias. The CHAMP has an unbiased estimate.

Simulation 4:

Consider y_1 = (1,2...,500); y_0 = (500,499,...,1); n = 500, and $W(1)$ = 250.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	500	0.81	0.11	0.81	0.67
Adhoc:Right	500	-0.81	0.11	-0.81	0.67
Adhoc:Center	500	0.00	0.13	0.00	0.02
Simple Difference	500	0.00	0.00	0.00	0.00
CHAMP	500	0.00	0.11	0.00	0.01

Table 3.7: Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate.

3.5 Special Cases

We consider cases where the potential outcomes under the treatments vary either at the ends or in the middle whilst keeping the potential outcomes under the control constant at 0 for every unit. We choose a sample size of 16 units, where equal number of treated and control units. The goal is to know where the bias is.

Special Cases to Consider		
Cases	Potential	
	Treatment	Control
1	-20,-10,-5,-2,-1,-1,0,0,0,0,1,1,2,5,10,20	0,0,0,...,0
2	0,0,-1,1,-2,5,10,-20,20,-10,-5,2,1,-1,0,0	0,0,0,...,0
3	-8,-4,2,2,2,2,2,2,2,2,2,2,2,2,-4,-8	0,0,0,...,0
4	2,2,2,2,2,2,-4,-8,-8,-4,2,2,2,2,2,2	0,0,0,...,0
5	0,0,-1,-1,-2,-5,-10,-20,0,0,1,1,2,5,10,20	0,0,0,...,0
6	0,0,-1,-1,-2,-5,-10,-20,20,10,5,2,1,1,0,0	0,0,0,...,0
7	-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,150	0,0,0,...,0
8	150,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10	0,0,0,...,0
9	2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32	32,30,28,26,24,22,20,18,16,14,12,10,8,6,4,2

Table 3.8: Special Cases

Case 1: Consider y_1 = (-20,-10,-5,-2,-1,-1,0,0,0,0,1,1,2,5,10,20);

y_0 = (0,0,...,0); $n = 16$, and $W(1) = 8$.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	16	0.31	1.58	0.31	2.60
Adhoc:Right	16	-0.31	1.58	-0.31	2.60
Adhoc:Center	16	0.01	1.39	0.01	1.94
Simple Difference	16	0.00	2.10	0.00	4.43
CHAMP	16	0.00	1.84	0.00	3.38

Table 3.9: Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate.

Case 2:

Consider y_1 = (0,0,-1,1,-2,5,10,-20,20,-10,-5,2,1,-1,0,0);

y_0 = (0,0,...,0); $n = 16$, and $W(1) = 8$.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	16	-0.01	2.66	-0.01	7.04
Adhoc:Right	16	-0.01	2.63	-0.01	7.04
Adhoc:Center	16	-0.07	2.90	-0.00	8.43
Simple Difference	16	0.00	2.10	0.00	4.43
CHAMP	16	0.00	2.64	0.00	6.94

Table 3.10: Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. The above ad-hoc post-stratification provided estimates with smaller bias than the ends to center ad-hoc post-stratification. The CHAMP has an unbiased estimate.

Case 3:

Consider y_1 = (-8,-4,2,2,2,2,2,2,2,2,2,2,2,2,2,-4,-8); y_0 = (0,0,...,0); $n = 16$, and $W(1) = 8$.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	16	0.12	0.95	0.13	0.92
Adhoc:Right	16	0.13	0.95	0.13	0.92
Adhoc:Center	16	0.27	0.73	0.27	0.61
Simple Difference	16	0.00	0.93	0.00	0.87
CHAMP	16	0.00	1.07	0.00	1.14

Table 3.11: Left-to-Right ad-hoc post-stratification estimator provided estimates with smaller bias than the ends to center ad-hoc post-stratification. The CHAMP has an unbiased estimate.

Case 4: Consider y_1 = (2,2,2,2,2,2,-4,-8,-8,-4,2,2,2,2,2,2);

y_0 = (0,0,...,0); $n = 16$, and $W(1) = 8$.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	16	-0.03	0.96	-0.03	0.83
Adhoc:Right	16	-0.03	0.96	-0.03	0.83
Adhoc:Center	16	-0.16	0.99	-0.16	0.91
Simple Difference	16	0.00	0.93	0.00	0.87
CHAMP	16	0.00	1.07	0.00	1.14

Table 3.12: Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. The above ad-hoc post-stratification provided estimates with smaller bias than the ends to center ad-hoc post-stratification. The CHAMP has an unbiased estimate.

Case 5:

Consider y_1 = (0,0,-1,1,-2,5,10,-20,20,-10,-5,2,1,-1,0,0);

y_0 = (0,0,...,0); $n = 16$, and $W(1) = 8$.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	16	-0.01	2.65	-0.01	7.04
Adhoc:Right	16	0.01	2.65	0.01	7.04
Adhoc:Center	16	-0.07	2.90	-0.07	8.43
Simple Difference	16	0.00	2.10	0.00	4.43
CHAMP	16	0.00	2.64	0.00	6.95

Table 3.13: Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimates, and gave us minimum bias. The CHAMP has an unbiased estimate.

Case 6:

Consider y_1 = (0,0,-1,-1,-2,-5,-10,-20,20,10,5,2,1,1,0,0);

y_0 = (0,0,...,0); $n = 16$, and $W(1) = 8$.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	16	0.04	2.44	0.04	5.97
Adhoc:Right	16	-0.04	2.44	-0.04	5.97
Adhoc:Center	16	-0.02	2.76	-0.02	7.60
Simple Difference	16	0.00	2.10	0.00	4.43
CHAMP	16	0.00	1.84	0.00	3.39

Table 3.14: Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate.

Case 7:

Consider y_1 = (-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,150);

y_0 = (0,0,...,0); n = 16, and $W(1)$ = 8.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	16	0.09	11.85	0.09	140.51
Adhoc:Right	16	-1.37	8.89	-1.37	80.90
Adhoc:Center	16	-1.36	8.92	-1.36	81.47
Simple Difference	16	0.00	10	0.00	100
CHAMP	16	0.00	11.70	0.00	137

Table 3.15: Left-to-Right Ad-hoc post-stratification gives us a minimum bias. The CHAMP has an unbiased estimate.

Case 8:

Consider y_1 = (150,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10);

y_0 = (0,0,...,0); n = 16, and $W(1)$ = 8.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	16	-1.37	8.89	-1.37	80.90
Adhoc:Right	16	0.09	11.85	0.09	140.51
Adhoc:Center	16	-1.37	8.90	-1.37	80.90
Simple Difference	16	0.00	10	0.00	100
CHAMP	16	0.00	11.7	0.00	137

Table 3.16: Right to Left ad-hoc post-stratification provides us with a minimum bias. The CHAMP has an unbiased estimate.

Case 9:

Consider y_1 = (2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32);

y_0 = (32,30,28,26,24,22,20,18,16,14,12,10,8,6,4,2); $n = 16$; $W(1) = 8$.

ATE = 0

Estimator	n	$Estimate$	$StandardError$	$Bias$	MSE
Adhoc:Left	16	0.75	0.70	0.75	1.05
Adhoc:Right	16	-0.75	0.70	-0.75	1.05
Adhoc:Center	16	0.10	0.75	0.10	0.57
Simple Difference	16	0.00	0.00	0.00	0.00
CHAMP	16	0.00	0.79	0.00	0.63

Table 3.17: Left-to-Right and Right-to-Left Ad-hoc post-stratifications have the same standard error and same absolute value of the estimate. Unlike the above ad-hoc post-stratification, the ends to center gives us a minimum bias. The CHAMP has an unbiased estimate.

3.6 Discussion on the Simulations and Cases

CHAMP post-stratification estimator provided us with unbiased estimates. It proved to be a better estimate. However, it was difficult to see what was causing the bias or its location. Hence,considering cases with different potential outcomes, where we vary the numbers at the ends and at the center of the potential outcomes helped to determine location of the bias. For instance, Tables 3.15 and 3.16, bias was heavy at the tails while tables 3.13 and 3.14, bias was heavy at the center.

3.7 Right Heart Catheterization (RHC)

The Right Heart Catheterization (RHC) was introduced 30 years ago and is widely used to actively study patients who were critically ill in the Intensive Care Unit(ICU). Previous studies provide no standard or limited information on assessing its clinical effectiveness or cost-effectiveness. In an observational study with data from Study to Understand Prognoses

and Preferences for Outcomes and Risks and Treatments (SUPPORT) where 5735 patients were considered, 2184 patients were managed with RHC in the first 24 hours and 3551 patients were managed without RHC. The responses considered were the time it took patients to survive, the length of time patients stayed at the hospital and ICU, costs incurred at the hospital, and the intensity of care. They measured the rate of survival up to day 180. It was observed that patients managed with RHC had 180-day survival rates as well as higher hospital costs than patients managed without RHC (Connors et al., 1996).

Miratrix et al. considered the implementation of post-stratification in adjusting treatment effect estimates in randomized experiments. In a Pulmonary Artery Catheterization(PAC) or RHC trial, they selected 1013 subjects. 506 subjects were assigned to treatment and 507 subjects were assigned to control. The response variable was the Quality Adjusted Life years, which implies higher values for Quality Adjusted Life years would mean longer life. Covariate imbalance in randomized controlled trials in predicting the probability of death could be due to too many fluctuations in the data or the initial health of the patients. If we group patients based on their initial health using post-stratification procedure, the covariate imbalance can be solved. We expect patients with better initial health to have higher quality of life after the procedure than those with worse health. Miratrix et al. estimated the treatment effects within the resulting strata and averaged them appropriately. Post-stratification corrected the bias that were as a result of covariate imbalance and decreased variance of treatment effect estimates.

3.7.1 Description of SUPPORT Data

The Study to Understand Prognoses and Preferences for Outcomes and Risks and Treatments (SUPPORT) was a 5-medical center study of decision making and responses of adult patients who were critically ill and hospitalized. The five centers were: Beth Israel Hospital, Boston, Mass; Duke University Medical Center, Durham, NC; Metro-Health Medical Center, Cleveland, Ohio; St Joseph's Hospital, Marshfield, WI; and University of California Medical

Center, Los Angeles. The coordination of the study was performed by George Washington University, Washington, DC, and the statistical analysis was performed at Duke University. The disease classifications were acute respiratory failure (ARF), chronic obstructive pulmonary disease (COPD), congestive heart failure (CHF), cirrhosis, nontraumatic coma, colon cancer metastatic to the liver, non-small cell cancer of the lung (stage III or IV), and multiorgan system failure(MOSF) with malignancy or sepsis.

The SUPPORT data excluded the following factors: age less than 18 years, death or discharge within 48 hours, inability to speak English, acute psychiatric disorders, pregnancy, acquired immunodeficiency syndrome (AIDS), acute burns, and head trauma or other trauma (unless acute respiratory failure or MOSF developed later). The design for the SUPPORT data contained two phases. Phase I was a prospective observational study, and Phase II was a cluster randomized control trial. In Phase I, 4301 patients enrolled from June 1989 to June 1991. In Phase II, 4804 patients enrolled from January 1992 to January 1994. Admitted or transferred patients to the ICU within the first 24 hours were allowed to partake in the study. Hence, a study population of 5735 patients was considered. An APACHE score—which provides a measure of initial health—was measured on each patient upon admission. Of note, the patients included in the SUPPORT study were more likely to receive RHC if they had worse initial health. This suggests that a difference-in-means estimate without adjustment would be biased towards the ineffectiveness of RHC.

3.8 CHAMP Estimator on RHC

The RHC dataset that we worked on contains information on 5735 subjects with 63 variables. We applied the CHAMP post-stratification estimator on this dataset. The CHAMP post-stratification has 3 levels of clusters. The first level contains clusters of 5 units; the second level contains clusters comprised of 31 levels 1 clusters; the third level contains the entire dataset.

We post-stratified on the APACHE initial health score, i.e. we sorted on APACHE, and divided into groups based on this sorted score. We had an estimate of 0.008 using the CHAMP post-stratification estimator. Imbalances in the estimates arise due to the fact that the difference in means estimate would be biased towards the ineffectiveness of the RHC.

Group	Unadjusted Mean	P-value
Treatment Control	60.73901 50.93354	<2.2e-16

Table 3.18: The means in the treatment groups were significantly different from zero with a p-value of <2.2e-16, indicating an imbalance

3.9 Discussion

The CHAMP post-stratification estimator is a better estimator in determining the unbiasedness of treatment effect estimates. The CHAMP estimator provided unbiased estimates in all the simulations and cases considered. Where the ad-hoc estimators provided unbiased estimates, the CHAMP estimator had lower standard errors and mean squared error(MSE). CHAMP also increases precision. The simple difference in means estimator gave us unbiased estimates with low standard errors. This is as a result of how we chose the number of units under treatment and control. Thus, due to the symmetry in our models, it makes the difference in means better.

Chapter 4

Conclusion

In this report, we used the Neyman Rubin model of response to obtain estimators that will provide unbiased estimates of the average treatment effect. We examined the ad-hoc post-stratification method, where we form groups of units so that every group has at least one treated and control unit. We focused on the left-to-right, right-to-left and ends-to-center ad-hoc post-stratifications. Also, we presented our estimation method called Cluster Hierarchy and Merge Post-stratification(CHAMP). We compared the ad-hoc post-stratification method with the CHAMP method. We found that CHAMP post-stratification estimator always provides unbiased estimates of the Average Treatment Effects. Where the ad-hoc post-stratification provided unbiased estimates, the CHAMP post-stratification estimates had a lower standard deviation. This shows that CHAMP post-stratification estimator is a better estimator relative to the ad-hoc post-stratification estimator. CHAMP post-stratification successfully eliminates bias while ensuring small standard errors of post-stratification estimators. In all cases of unbiasedness, CHAMP post-stratification estimator improves precision relative to ad-hoc post-stratification estimator. We applied our method to the Right Heart Catheterization dataset. We post-stratified on the APACHE score, but had different estimates and that was due to the imbalance in initial health between treatment groups on APACHE score.

4.1 Future Work

Probing deeper, the results in this report also provide a strong foundation for future work in the estimation of average treatment effects using CHAMP. One area of future work is proving the theoretical properties rigorously about the unbiasedness and standard errors of the CHAMP estimator. Another area is in finding rules for dividing units when there are no small prime divisors of the number of units.

Bibliography

A Abadie and GW Imbens. On the failure of the bootstrap for matching estimators issued in june. Technical report, NBER Technical Working Paper, 2006.

Timothy R Bainbridge. The committee on standards-precision and bias. *ASTM Standardization News*, 13(1):44–46, 1985.

Adrian E Bauman, James F Sallis, David A Dzewaltowski, and Neville Owen. Toward a better understanding of the influences on physical activity: the role of determinants, correlates, causal variables, mediators, moderators, and confounders. *American journal of preventive medicine*, 23(2):5–14, 2002.

Jake Bowers and Ben Hansen. Attributing effects to a get-out-the-vote campaign using full matching and randomization inference. In *Prepared for presentation at the Annual Meeting of the Midwestern Political Science Association*. Citeseer, 2005.

Joan Fisher Box. Ra fisher and the design of experiments, 1922–1926. *The American Statistician*, 34(1):1–7, 1980.

Nicholas A Christakis and Theodore J Iwashyna. The health impact of health care on families: a matched cohort study of hospice use by decedents and mortality outcomes in surviving, widowed spouses. *Social science & medicine*, 57(3):465–475, 2003.

Alfred F Connors, Theodore Speroff, Neal V Dawson, Charles Thomas, Frank E Harrell, Douglas Wagner, Norman Desbiens, Lee Goldman, Albert W Wu, Robert M Califf, et al. The effectiveness of right heart catheterization in the initial care of critically iii patients. *Jama*, 276(11):889–897, 1996.

David Roxbee Cox. Planning of experiments. 1958.

Rajeev H Dehejia and Sadek Wahba. Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American statistical Association*, 94 (448):1053–1062, 1999.

Rajeev H Dehejia and Sadek Wahba. Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and statistics*, 84(1):151–161, 2002.

Ronald A Fisher. The arrangement of field experiments. In *Breakthroughs in statistics*, pages 82–91. Springer, 1992.

Ronald Aylmer Fisher. Design of experiments. *Br Med J*, 1(3923):554–554, 1936.

Michael J Higgins, Fredrik Sävje, and Jasjeet S Sekhon. Improving massive experiments with threshold blocking. *Proceedings of the National Academy of Sciences*, 113(27):7369–7376, 2016.

Paul W Holland. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960, 1986.

Kosuke Imai. Do get-out-the-vote calls reduce turnout? the importance of statistical methods for field experiments. *American Political Science Review*, 99(2):283–300, 2005.

Luke Keele, Ismail K White, and Corrine McConnaughy. Adjusting experimental data: Models versus design. In *APSA 2010 Annual Meeting Paper*, 2010.

Martin Krzywinski and Naomi Altman. Points of significance: Analysis of variance and blocking. *nature methods*, 11(7):699–700, 2014.

Roderick JA Little. Post-stratification: a modeler's perspective. *Journal of the American Statistical Association*, 88(423):1001–1012, 1993.

Luke W Miratrix, Jasjeet S Sekhon, and Bin Yu. Adjusting treatment effect estimates by post-stratification in randomized experiments. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(2):369–396, 2013.

Stephen L Morgan and David J Harding. Matching estimators of causal effects: Prospects and pitfalls in theory and practice. *Sociological methods & research*, 35(1):3–60, 2006.

Stuart J Pocock, Susan E Assmann, Laura E Enos, and Linda E Kasten. Subgroup analysis, covariate adjustment and baseline comparisons in clinical trial reporting: current practiceand problems. *Statistics in medicine*, 21(19):2917–2930, 2002.

Paul R Rosenbaum. Overt bias in observational studies. In *Observational studies*, pages 71–104. Springer, 2002.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Donald B Rubin. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, pages 34–58, 1978.

Donald B Rubin. Estimating causal effects from large data sets using propensity scores. *Annals of internal medicine*, 127(8_Part_2):757–763, 1997.

Donald B Rubin. *Matched sampling for causal effects*. Cambridge University Press, 2006.

Jasjeet S Sekhon. Quality meets quantity: Case studies, conditional probability, and counterfactuals. *Perspectives on Politics*, 2(2):281–293, 2004.

Jasjeet S Sekhon. Opiates for the matches: Matching methods for causal inference. *Annual Review of Political Science*, 12:487–508, 2009.

Sarjinder Singh. *Advanced Sampling Theory With Applications: How Michael"" Selected"" Amy*, volume 2. Springer Science & Business Media, 2003.

Herbert L Smith. 6. matching with multiple controls to estimate treatment effects in observational studies. *Sociological methodology*, 27(1):325–353, 1997.

Richard Valliant. Poststratification and conditional variance estimation. *Journal of the American Statistical Association*, 88(421):89–96, 1993.

Mark J West. Stereological methods for estimating the total number of neurons and synapses: issues of precision and bias. *Trends in neurosciences*, 22(2):51–61, 1999.

Appendix A

R code for Ad-hoc Post-Stratification Estimator

```
    #For a given 0-1 randomization, get left-to-right ad-hoc stratification.
ltradhoc = function(randomiz){

  #Store randomization into a variable called myrand for manipulation
  myrand = randomiz

  #Store all indices in a variable called tempstrat for manipulation
  tempstrat = 1:length(randomiz)

  #Create a list to store the stratification
  mystrat = list()

  #Boolean to determine whether or not to continue
  docont = TRUE
```

```
#Count the number of entries in mystrat
count = 0
while(docont){

  #Get which are 0's and which are 1's
  which0 = which(myrand == 0)
  which1 = which(myrand == 1)

  #If either which0 or which1 is empty, finish out mystrat and break.
  if(length(which0) == 0 | length(which1) == 0){

    #Add the rest of the indices to the last entry of mystrat
    mystrat[[count]] = c(mystrat[[count]], tempstrat)

    #Prepare to end for loop
    docont = FALSE
  }

  #Otherwise
  else{
    #Adding a stratum.  Increase the count
    count = count + 1

    #Make a stratum comprised of all entries leading to the first 0 or 1.
    endstrat = max(min(which0), min(which1))
    addstrat = tempstrat[1:endstrat]
```

```
      #Add addstrat to mystrat
      mystrat[[count]] = addstrat

      #Remove entries from myrand and tempstrat
      myrand = myrand[-(1:endstrat)]
      tempstrat = tempstrat[-(1:endstrat)]

    }

  }

  #Output mystrat
  mystrat
}



#Then run lapply to get your result.

#For a given 0-1 randomization, get right-to-left ad-hoc stratification.
rtladhoc = function(randomiz){

  #Store randomization into a variable called myrand for manipulation
  myrand = randomiz

  #Store all indices in a variable called tempstrat for manipulation
  tempstrat = 1:length(randomiz)
```

```
#Create a list to store the stratification
mystrat = list()

#Boolean to determine whether or not to continue
docont = TRUE

#Count the number of entries in mystrat
count = 0
while(docont){

  #Get which are 0's and which are 1's
  which0 = which(myrand == 0)
  which1 = which(myrand == 1)

  #If either which0 or which1 is empty, finish out mystrat and break.
  if(length(which0) == 0 | length(which1) == 0){

    #Add the rest of the indices to the last entry of mystrat
    #Sort it to make it look a little cleaner
    mystrat[[count]] = sort(c(mystrat[[count]], tempstrat))

    #Prepare to end for loop
    docont = FALSE
  }

  #Otherwise
  else{
```

```
      #Adding a stratum.  Increase the count
      count = count + 1

      #Make a stratum comprised of all entries leading to the first 0 or 1
      #coming from the right side.
      startstrat = min(max(which0), max(which1))
      endstrat = length(myrand)
      addstrat = tempstrat[startstrat:endstrat]

      #Add addstrat to mystrat
      mystrat[[count]] = addstrat

      #Remove entries from myrand and tempstrat
      myrand = myrand[-(startstrat:endstrat)]
      tempstrat = tempstrat[-(startstrat:endstrat)]

    }

  }

  #Output mystrat
  mystrat
}



#For a given 0-1 randomization, get end to center ad-hoc stratification.
```

```
etcadhoc = function(randomiz){

  #Store randomization into a variable called myrand for manipulation
  myrand = randomiz

  #Store all indices in a variable called tempstrat for manipulation
  tempstrat = 1:length(randomiz)

  #Create a list to store the stratification
  mystrat = list()

  #Boolean to determine whether or not to continue
  docont = TRUE

  #Count the number of entries in mystrat
  count = 0
  while(docont){

    #Get which are 0's and which are 1's
    which0 = which(myrand == 0)
    which1 = which(myrand == 1)

    #If either which0 or which1 is empty, finish out mystrat and break.
    if(length(which0) == 0 | length(which1) == 0){

      #Add the rest of the indices to the last entry of mystrat
      #Sort it to make it look a little cleaner
```

```
  mystrat[[count]] = sort(c(mystrat[[count]], tempstrat))

  #Prepare to end for loop
  docont = FALSE
}

#Otherwise
else{

  #If count is even, do left to right
  if(count%%2 == 0){

    #Adding a stratum.  Increase the count
    count = count + 1

    #Make a stratum comprised of all entries leading to the first 0 or 1.
    endstrat = max(min(which0), min(which1))
    addstrat = tempstrat[1:endstrat]

    #Add addstrat to mystrat
    mystrat[[count]] = addstrat

    #Remove entries from myrand and tempstrat
    myrand = myrand[-(1:endstrat)]
    tempstrat = tempstrat[-(1:endstrat)]

  }
```

```
      #If count is odd, do right to left
      else{
        #Adding a stratum.  Increase the count
        count = count + 1

        #Make a stratum comprised of all entries leading to the first 0 or 1
        #coming from the right side.
        startstrat = min(max(which0), max(which1))
        endstrat = length(myrand)
        addstrat = tempstrat[startstrat:endstrat]

        #Add addstrat to mystrat
        mystrat[[count]] = addstrat

        #Remove entries from myrand and tempstrat
        myrand = myrand[-(startstrat:endstrat)]
        tempstrat = tempstrat[-(startstrat:endstrat)]
      }
    }

  }

  #Output mystrat
  mystrat
}
```

```
#Pick Treated and total number of units
myn = 16
myt = 8


##Use this when myn<=20
#Get all combinations of choosing myt units from myn
allcombs = combn(myn,myt)

#Get all 0-1 randomizations
allrands = apply(allcombs, 2, function(x, myn){
  assign = rep(0,myn)
  assign[x] = 1
  assign
}, myn = myn)


##Use this if myn>20
#If simulated data
set.seed(1094)
allrands = replicate(10000, sample(c(rep(1,myt),rep(0,myn-myt))))

n = length(allrands[1,])


#Get all left to right stratifications:
aa = apply(allrands, 2, ltradhoc)
```

```
#Get all right to left stratifications:
bb = apply(allrands, 2, rtladhoc)

#Get all ends to center stratifications
cc = apply(allrands, 2, etcadhoc)

# #MIKE: If you need to extend to an even larger set of nc and nt, use simulation
# nc = 50
# nt = 50
#
# #Numrands probably needs to be large (on the order of 100,000)
# numrands = 200
# samprands = replicate(numrands, sample(c( rep(0,nc), rep(1,nt) ) ) )
#
# #Get all left to right stratifications:
# aa1 = apply(samprands, 2, ltradhoc)
#
# #Get all right to left stratifications:
# bb1 = apply(samprands, 2, rtladhoc)
#
# #Get all ends to center stratifications:
# cc1 = apply(samprands, 2, etcadhoc)
```

```
GetStratamean<-function(y1,y0){
  y1rands<-t(allrands*y1)
  y0rands <- t((1-allrands)*y0)
  Stratest = rep(0,n)
  for (i in 1:n){
    aa_i<-aa[[i]]
    y1mean<-numeric()
    y0mean<-numeric()
    StrataMean_vec<-numeric()
    for(k in 1:length(aa_i)){
      y1mean[k]<-sum(y1rands[i,aa_i[[k]]])/sum(allrands[aa_i[[k]],i] == 1)
      y0mean[k]<-sum(y0rands[i,aa_i[[k]]])/sum(allrands[aa_i[[k]],i] == 0)
      StrataMean_vec[k] <-y1mean[k] - y0mean[k]



      stratSize = length(aa_i[[k]])
      numUnits = length(allrands[,i])
      StrataMean_vec[k] = StrataMean_vec[k]*stratSize/numUnits
      #MIKE: Stratum weighting on StrataMean_vec[k]


    }
    #MIKE: At this point, you need to store your estimate.
    #MIKE: set stratest[i] = sum(StrataMean_vec)
    print(StrataMean_vec)
```

```
    print(aa_i)
    print(numUnits)
    print(allrands[,i])
    Stratest[i] = sum(StrataMean_vec)
    #cat("Row",i,": ",Stratest,"\n")

  }
  TrueATE  = mean(y1-y0)


  cat("ExpectedEstimate(Overall ATE)",":",mean(Stratest),"TrueATE",":",TrueATE,"Variance

  Stratest
}


GetStratamean1<-function(y1,y0){
  y1rands<-t(allrands*y1)
  y0rands <- t((1-allrands)*y0)

  Stratestj = rep(0,n)
  for (j in 1:n){
    bb_j<-bb[[j]]
    y1mean<-numeric()
    y0mean<-numeric()
    StrataMean_vecj<-numeric()
    for(b in 1:length(bb_j)){
```

```
      y1mean[b]<-sum(y1rands[j,bb_j[[b]]])/sum(allrands[bb_j[[b]],j] == 1)
      y0mean[b]<-sum(y0rands[j,bb_j[[b]]])/sum(allrands[bb_j[[b]],j] == 0)
      StrataMean_vecj[b] <-y1mean[b] - y0mean[b]


      stratSize = length(bb_j[[b]])
      numUnits = length(allrands[,j])
      StrataMean_vecj[b] = StrataMean_vecj[b]*stratSize/numUnits
      #MIKE: Stratum weighting on StrataMean_vec[k]
    }

    Stratestj[j] = sum(StrataMean_vecj)
    #cat("Row",i,": ",Stratest,"\n")
  }
  TrueATE  = mean(y1-y0)

  cat("ExpectedEstimate(Overall ATE)",":",mean(Stratestj),"TrueATE",":",TrueATE,"Varianc
  Stratestj

}


GetStratamean2<-function(y1,y0){

  y1rands<-t(allrands*y1)
  y0rands <- t((1-allrands)*y0)
  Stratestz = rep(0,n)
```

```
for (z in 1:n){
  cc_z<-cc[[z]]
  y1mean<-numeric()
  y0mean<-numeric()
  StrataMean_vecz<-numeric()

  for(c in 1:length(cc_z)){


    y1mean[c]<-sum(y1rands[z,cc_z[[c]]])/sum(allrands[cc_z[[c]],z] == 1)
    y0mean[c]<-sum(y0rands[z,cc_z[[c]]])/sum(allrands[cc_z[[c]],z] == 0)
    StrataMean_vecz[c] <-y1mean[c] - y0mean[c]


    stratSize = length(cc_z[[c]])
    numUnits = length(allrands[,z])
    StrataMean_vecz[c] = StrataMean_vecz[c]*stratSize/numUnits
    #MIKE: Stratum weighting on StrataMean_vec[k]

  }

  Stratestz[z] = sum(StrataMean_vecz)

}

TrueATE  = mean(y1-y0)
```

```
  cat("ExpectedEstimate(Overall ATE)",":",mean(Stratestz),"TrueATE",":",TrueATE,"Varianc
  Stratestz

}
```

Appendix B

R-Code for CHAMP Post-Stratification

```
#CHAMP POSTRATIFICATION EXACT


champPost = function(randomiz){
  if(length(randomiz) == 6){
    tiers = matrix(0,6,2)
    tiers[,1] = ceiling(1:6/3)
    tiers[,2] = 2 + ceiling(1:6/6)



    #Keeps track of which tier we're on.
    whichtier = rep(0,6)


    #Keeps track of which block we're on.
    whichblock = rep(0,6)
```

```
#Which units have not been assigned to a tier yet
slunits = rep(0,6)

#Check Hierarchy
for(i in 1:2){

  #Need to break?
  if(length(slunits) == sum(slunits)){
    break
  }

  #Which blocks are bad?
  remblocks = tiers[which(slunits == 0),i]

  #which units belong to those blocks?
  consunits = which(tiers[,i] %in% remblocks)

  #Check mean in that tier
  testagg = aggregate(randomiz[consunits], by = list(tiers[consunits,i]), mean)

  #Blocks that meet the guidelines
  goodblocks = testagg[(testagg[,2] > 0 & testagg[,2] < 1),1]

  #Which units correspond to the good tiers now
  goodunits = which(tiers[,i] %in% goodblocks)
```

```
  #What tier do they belong to
  whichtier[goodunits] = i

  #Which block do they belong to
  whichblock[goodunits] = tiers[goodunits,i]

  #Now we have a 1 here.
  slunits[goodunits] = 1
}
}else if(length(randomiz) == 8){
  tiers = matrix(0,8,3)
  tiers[,1] = ceiling(1:8/2)
  tiers[,2] = 4 + ceiling(1:8/4)
  tiers[,3] = 6 + ceiling(1:8/8)

  #Keeps track of which tier we're on.
  whichtier = rep(0,8)

  #Keeps track of which block we're on.
  whichblock = rep(0,8)

  #Which units have not been assigned to a tier yet
  slunits = rep(0,8)

  #Check Hierarchy
  for(i in 1:3){
```

```
#Need to break?
if(length(slunits) == sum(slunits)){
  break
}

#Which blocks are bad?
remblocks = tiers[which(slunits == 0),i]

#which units belong to those blocks?
consunits = which(tiers[,i] %in% remblocks)

#Check mean in that tier
testagg = aggregate(randomiz[consunits], by = list(tiers[consunits,i]), mean)

#Blocks that meet the guidelines
goodblocks = testagg[(testagg[,2] > 0 & testagg[,2] < 1),1]

#Which units correspond to the good tiers now
goodunits = which(tiers[,i] %in% goodblocks)

#What tier do they belong to
whichtier[goodunits] = i

#Which block do they belong to
whichblock[goodunits] = tiers[goodunits,i]

#Now we have a 1 here.
```

```
    slunits[goodunits] = 1
  }

}else if(length(randomiz) == 10){
  tiers = matrix(0,10,3)
  tiers[,1] = ceiling(1:10/2)
  tiers[,2] = 5 + ceiling(1:10/5)
  tiers[,3] = 7 + ceiling(1:10/10)

  #Keeps track of which tier we're on.
  whichtier = rep(0,10)

  #Keeps track of which block we're on.
  whichblock = rep(0,10)

  #Which units have not been assigned to a tier yet
  slunits = rep(0,10)

  #Check Hierarchy
  for(i in 1:3){

    #Need to break?
    if(length(slunits) == sum(slunits)){
      break
    }

    #Which blocks are bad?
```

```
    remblocks = tiers[which(slunits == 0),i]

    #which units belong to those blocks?
    consunits = which(tiers[,i] %in% remblocks)

    #Check mean in that tier
    testagg = aggregate(randomiz[consunits], by = list(tiers[consunits,i]), mean)

    #Blocks that meet the guidelines
    goodblocks = testagg[(testagg[,2] > 0 & testagg[,2] < 1),1]

    #Which units correspond to the good tiers now
    goodunits = which(tiers[,i] %in% goodblocks)

    #What tier do they belong to
    whichtier[goodunits] = i

    #Which block do they belong to
    whichblock[goodunits] = tiers[goodunits,i]

    #Now we have a 1 here.
    slunits[goodunits] = 1
  }


} else if(length(randomiz) == 16){
```

```
tiers = matrix(0,16,4)
tiers[,1] = ceiling(1:16/2)
tiers[,2] = 8 + ceiling(1:16/4)
tiers[,3] = 12 + ceiling(1:16/8)
tiers[,4] = 14 + ceiling(1:16/16)

#Keeps track of which tier we're on.
whichtier = rep(0,16)

#Keeps track of which block we're on.
whichblock = rep(0,16)

#Which units have not been assigned to a tier yet
slunits = rep(0,16)

#Check Hierarchy
for(i in 1:4){

  #Need to break?
  if(length(slunits) == sum(slunits)){
    break
  }

  #Which blocks are bad?
  remblocks = tiers[which(slunits == 0),i]

  #which units belong to those blocks?
```

```
    consunits = which(tiers[,i] %in% remblocks)

    #Check mean in that tier
    testagg = aggregate(randomiz[consunits], by = list(tiers[consunits,i]), mean)

    #Blocks that meet the guidelines
    goodblocks = testagg[(testagg[,2] > 0 & testagg[,2] < 1),1]

    #Which units correspond to the good tiers now
    goodunits = which(tiers[,i] %in% goodblocks)

    #What tier do they belong to
    whichtier[goodunits] = i

    #Which block do they belong to
    whichblock[goodunits] = tiers[goodunits,i]

    #Now we have a 1 here.
    slunits[goodunits] = 1
  }

} else if(length(randomiz) == 100){
  tiers = matrix(0,100,4)
  tiers[,1] = ceiling(1:100/5)
  tiers[,2] = 20 + ceiling(1:100/10)
  tiers[,3] = 30 + ceiling(1:100/20)
  tiers[,4] = 35 + ceiling(1:100/100)
```

```
#Keeps track of which tier we're on.
whichtier = rep(0,100)

#Keeps track of which block we're on.
whichblock = rep(0,100)

#Which units have not been assigned to a tier yet
slunits = rep(0,100)

#Check Hierarchy
for(i in 1:4){

  #Need to break?
  if(length(slunits) == sum(slunits)){
    break
  }

  #Which blocks are bad?
  remblocks = tiers[which(slunits == 0),i]

  #which units belong to those blocks?
  consunits = which(tiers[,i] %in% remblocks)

  #Check mean in that tier
  testagg = aggregate(randomiz[consunits], by = list(tiers[consunits,i]), mean)
```

```
    #Blocks that meet the guidelines
    goodblocks = testagg[(testagg[,2] > 0 & testagg[,2] < 1),1]

    #Which units correspond to the good tiers now
    goodunits = which(tiers[,i] %in% goodblocks)

    #What tier do they belong to
    whichtier[goodunits] = i

    #Which block do they belong to
    whichblock[goodunits] = tiers[goodunits,i]

    #Now we have a 1 here.
    slunits[goodunits] = 1
  }

} else if(length(randomiz) == 500){
  tiers = matrix(0,500,5)
  tiers[,1] = ceiling(1:500/5)
  tiers[,2] = 100 + ceiling(1:500/10)
  tiers[,3] = 150 + ceiling(1:500/20)
  tiers[,4] = 175 + ceiling(1:500/100)
  tiers[,5] = 180 + ceiling(1:500/500)

  #Keeps track of which tier we're on.
  whichtier = rep(0,500)
```

```
#Keeps track of which block we're on.
whichblock = rep(0,500)

#Which units have not been assigned to a tier yet
slunits = rep(0,500)

#Check Hierarchy
for(i in 1:5){

  #Need to break?
  if(length(slunits) == sum(slunits)){
    break
  }

  #Which blocks are bad?
  remblocks = tiers[which(slunits == 0),i]

  #which units belong to those blocks?
  consunits = which(tiers[,i] %in% remblocks)

  #Check mean in that tier
  testagg = aggregate(randomiz[consunits], by = list(tiers[consunits,i]), mean)

  #Blocks that meet the guidelines
  goodblocks = testagg[(testagg[,2] > 0 & testagg[,2] < 1),1]

  #Which units correspond to the good tiers now
```

```
      goodunits = which(tiers[,i] %in% goodblocks)

      #What tier do they belong to
      whichtier[goodunits] = i

      #Which block do they belong to
      whichblock[goodunits] = tiers[goodunits,i]

      #Now we have a 1 here.
      slunits[goodunits] = 1
    }

  }
else if(length(randomiz) == 5735){
  tiers = matrix(0,5735,3)
  tiers[,1] = ceiling(1:5735/5)
  tiers[,2] = 1147 + ceiling(1:5735/155)
  tiers[,3] = 1184 + ceiling(1:5735/5735)

  #Keeps track of which tier we're on.
  whichtier = rep(0,5735)

  #Keeps track of which block we're on.
  whichblock = rep(0,5735)

  #Which units have not been assigned to a tier yet
  slunits = rep(0,5735)
```

```
#Check Hierarchy
for(i in 1:3){

  #Need to break?
  if(length(slunits) == sum(slunits)){
    break
  }

  #Which blocks are bad?
  remblocks = tiers[which(slunits == 0),i]

  #which units belong to those blocks?
  consunits = which(tiers[,i] %in% remblocks)

  #Check mean in that tier
  testagg = aggregate(randomiz[consunits], by = list(tiers[consunits,i]), mean)

  #Blocks that meet the guidelines
  goodblocks = testagg[(testagg[,2] > 0 & testagg[,2] < 1),1]

  #Which units correspond to the good tiers now
  goodunits = which(tiers[,i] %in% goodblocks)

  #What tier do they belong to
  whichtier[goodunits] = i
```

```
      #Which block do they belong to
      whichblock[goodunits] = tiers[goodunits,i]

      #Now we have a 1 here.
      slunits[goodunits] = 1
    }
    } else {
      stop("Not correct length")
    }
  whichblock
}

#Estimation given a poststratification
#Get estimate
estChampStrat = function(potout1,potout0, randomiz, champStrat){
  #Get Which Treated
  whichtrt = which(randomiz == 1)

  #Get Which Control
  whichcon = which(randomiz == 0)

  #Size of the block
  blocklength = aggregate(champStrat, by = list(champStrat), length)[,2]

  #Treatment mean by block
  trtres =  aggregate(potout1[whichtrt], by = list(champStrat[whichtrt]), mean)[,2]
```

```
  #Control mean by block
  conres =  aggregate(potout0[whichcon], by = list(champStrat[whichcon]), mean)[,2]

  #Standard post-stratified estimator
  est = sum((trtres - conres)*blocklength)/length(potout1)

  #Output
  est
}

#Function for stratifying and estimating
ChampStratAndEst = function(randomiz, potout1, potout0){

  #Get stratification
  champStrat = champPost(randomiz)
  print(champStrat)

  #Get estimate
  est = estChampStrat(potout1, potout0, randomiz, champStrat)

  est


}


#Pick Treated and total number of units
```

```
myn = myn
myt = myt

potout1 = potout1
potout0 = potout2



## For myn <=16, use this
#Get all combinations of choosing myt units from myn
allcombs = combn(myn,myt)

#Get all 0-1 randomizations
allrands = apply(allcombs, 2, function(x, myn){
  assign = rep(0,myn)
  assign[x] = 1
  assign
}, myn = myn)



## For myn>16, use this simulation
set.seed(1094)

allrands = replicate(10000, sample(c(rep(1,myt),rep(0,myn-myt))))



#Get all poststratification estimates
allests = apply(allrands, 2, ChampStratAndEst, potout1 = potout1, potout0 = potout0)
```

```
#Expected value
myev = mean(allests)

#Variance
myvar = mean((allests-myev)^2)

#Standard error
mysd = sqrt(myvar)

TrueATE = 1

#MSE
MSE = myvar +(myev - TrueATE)^2

#Output
c(myev, myvar, mysd,MSE)
```