PRACTICAL AND THEORETICAL ISSUES IN RANDOMIZATION

By

Chantal Belley, B.Sc.

December 2004

A Thesis

submitted to the School of Graduate Studies and Research

in partial fulfillment of the requirements

for the degree of

Master of Science in Biostatistics[1]

---

[1] The M.Sc. Program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Your file Votre référence
ISBN: 0-494-00775-3
Our file Notre référence
ISBN: 0-494-00775-3

Abstract

The objective of this thesis is to acknowledge some of the different randomized trial designs and analytical approaches, as well as specific issues related to these. The thesis can be divided into two parts. The first part focuses on the validity of using randomization tests and tests based on meta-analysis techniques in the context of cluster randomized trials with binary outcomes. Reasons for making statistical inferences based on a randomization model rather than on a population model are discussed. Also, the theory of randomization tests and tests based on meta-analysis techniques is reviewed. The validity of using such tests is evaluated for various study designs through simulation of size. Finally, application of the theory is made to the CHAT program. In the second part of the thesis, the appropriateness of using multiple personal digital assistants (PDA) per centre, each with pre-loaded blocked randomization schemes, to allocate participants in multi-centre randomized trials is evaluated. The probabilities of allocation imbalance and their potential effects on power are evaluated by simulation.

Acknowledgements

I would like to thank my supervisor, Nicholas Barrowman, for giving so generously of his time to guide and teach me, not only academically but also in my secular and personal endeavors. You've been a mentor and friend and I hope to have the chance to work with you again. I would also like to thank Larry Chambers of the Élizabeth Bruyère Research Institute for providing me with data for this thesis and Keith O'Rourke for his assistance in its application. Thanks also to Isabelle Gaboury, Khaled El Emam, David Moher for their assistance in conjunction with the PDA work, as well as Dr. Kwan Chan for his permission to use the recruitment numbers (at the first year's interim analysis) for Astronomer.

I want to thank my husband, Keith Oliver, for being so incredibly supportive and encouraging. You always encourage me to be the best I can and you definitely make me a better person. Thank you to my family for their unwavering support.

I'd also like to acknowledge the financial support provided by the Natural Sciences and Engineering Research Council.

Finally, I'd like to thank my dear friends, Dino, Eunice, Accalia and Isabella for distracting me from my studies.

Dedication

To my husband, Keith.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The randomized control trial (RCT) is considered by many to be the gold standard for evidence about the effectiveness of health interventions.[1] It is defined as an epidemiologic experiment whereby subjects are randomly allocated to groups which will or will not receive an intervention.[2] The effects of the intervention are then determined through comparison of the outcomes in each group. Major achievements of randomization include the ability to establish statistical significance of results, to make causal inferences on treatment effects, to balance known and unknown prognostic factors and also facilitate blinding of the observers and/or subjects. Nevertheless, there remains some controversy over various aspects of randomized trials. For example, Abel and Koch dispute some of the above-mentioned achievements of randomization stating that "randomization is often credited with advantages that it does not possess or confer", namely that of providing a basis for causal inferences on treatment effects by means of hypothesis testing.[3]

The objective of this thesis is not to delve into a philosophical debate on the use of randomization, but rather to acknowledge some of the different randomized trial designs and analytical approaches, as well as specific issues related to these. In this introductory chapter, we give an overview of some of the key design elements and

analytical approaches that will be discussed in more detail in other chapters of the thesis.

1.1 Design

1.1.1 Cluster randomization

In community intervention trials as well as in an increasing number of clinical trials, it is necessary to randomize groups of subjects to treatment, rather than each subject individually. Examples of groupings, which we shall call clusters, include communities, physician practices, hospitals, schools, and families. The majority of such cluster randomization trials will use one of the three following randomization designs: completely randomized, stratified, or pair-matched.[4] In a completely randomized design, clusters are allocated to intervention groups completely at random, without matching or stratification. This is only to be used when there is a sufficiently large number of clusters to be randomized so as to ensure adequate balancing of potential confounders, and subsequently sufficient power to detect treatment effects. However, imbalance may also occur when the clusters are large (thousands) if there is between-group variability of characteristics.[5] If the number of clusters is small or there is a need to ensure allocation balance for specific factors, stratified or matched are designs are used. There is much literature on the advantages and disadvantages of using stratification and matching.[6,7,8]

Additional reasons for adopting a cluster randomization strategy include 1) political feasibility and ethical considerations; 2) reduction of costs; or 3) when the intervention will affect all members of the group.[1,9] To illustrate, suppose a trial is carried out to assess the efficacy of a physician-based health campaign promoting smoking cessation. Due to ethical considerations, it would not be reasonable to expect a physician to give this beneficial treatment to only selected patients. If the intervention requires complex treatment, it may be cost efficient to treat selected

groups of patients rather than set up additional facilities to treat patients within each group. Also, it would be impossible to guarantee that those patients allocated to the control group would not be affected by the intervention and vice-versa. It is very plausible that a patient in the treatment group communicate with a patient in the control group, in the physicians' waiting area for example, which could result in a change of conduct on the part of either one of the patients. This would seriously compromise the ability to detect treatment effects.

Just as the randomized control trial (RCT) is considered the gold standard in health studies when it is possible to allocate individual participants, the cluster or group randomized trial (GRT) is the gold standard when it is necessary to allocate identifiable groups.[1] However, randomizing groups rather than subjects has important consequences for analysis and interpretation. The main concern is that individuals within a group are more likely to be homogenous and as a result, there may be greater heterogeneity among the groups, which must be accounted for in the analysis.[1,10,11] Thus, from a purely statistical viewpoint, cluster randomization is less efficient because of the need to incorporate a possible between-cluster variability in addition to the within cluster variability. Another concern related clustering has to do with the number of clusters needed to ensure adequate power to obtain statistically significant measures of treatment effects. On the other hand, cluster randomization may be the only option in situations where costs or feasibility of mounting a large trial are prohibitive, when a complete list of subjects is not available or when, as in the example described above, all members of a group need to receive the same treatment.

1.1.2 Blocked randomization

Blocked randomization is typically used to ensure that there will be nearly equal numbers of participants in each treatment arm and also to ensure a balance of important covariates across the treatment groups. Another feature of blocking is that is permits the concealment of treatment allocation from the observer and/or the

subject. Though some may feel that too much emphasis is placed on having equal group sizes to ensure sufficient power to perform final analyses,[12] a non-negligible advantage of forcing allocation balance is that it will increase the power at the time of interim analysis.

There are however situations in which the use of a blocking strategy can still lead to substantial allocation imbalance. This issue is examined in the context of a multi-centre trial where competitive recruitment of participants is undertaken, both at the centre and study level. Typically, in order to ensure a specified distribution of participants between treatment and control groups, each centre will randomize patients based on a centralized blocking system. For example, for each participant, the investigator will access the centre's centralized system to determine the treatment group to which they are to be assigned. Thus, at any given time, the maximum imbalance in allocation at the centre is one half the block size. Imbalance at the study level will be one half the block size times the number of centres in the trial. Hallstrom et al. warn that although there is no imbalance at the centre or the strata level, it is possible to have extreme allocation imbalances at the study level.[7] This can be problematic because the allocation imbalance reduces the probability of achieving balance in the important covariates across treatment groups. However, in the context under investigation, randomization of patients is accomplished using multiple personal digital assistants (PDA) per centre, each with pre-loaded blocking sequences. Based on this method a certain degree of allocation imbalance is inevitable for each centre. Implications of the allocation imbalances at the centre and study levels need to be assessed as this method of randomization participants in multi-centre trials becomes more popular.

1.2 Analysis

In broad terms, one's choice of analysis method is based on whether or not we are willing to make parametric assumptions about our data and the population from which

it came. This thesis discusses classical and frequentist modes of inference. Though Bayesian inference provides another interesting and very different way of justifying statistical inference, it is beyond the scope of this thesis to provide a review of this method. We shall now give a brief overview of both the classical and frequentist models.

1.2.1 Population Model

The classical model, also known as the population model, was formally proposed by Neyman and Pearson in 1928.[13] Inference under the population model is based on the premise that samples have been randomly selected from a study population that follows a defined theoretical frequency-distribution. The validity of statistical inferences made based on the population model depends on the two following key assumptions: firstly, the study samples have been selected randomly from large populations which follow theoretical frequency distributions; secondly, the sampled populations are normal in form and all have the same variance.

1.2.2 Randomization model

In biomedical research, sampling units are rarely selected by taking random samples from defined populations, whether these units be clusters or individuals. Rather, they are often selected through a non-random process, and are in turn randomized to experimental groups. For instance, in a trial measuring the effectiveness of a medication on patients with a particular rare disease, the sample population will often be constructed of individuals presenting to the hospital or clinic with the disease in question. These patients in turn would be randomized to either a treatment or a control group. In cases such as this, participants are not randomly selected from a larger defined population of patients with the disease. Thus, it is not strictly valid to use classical *t*- or *F*- tests to analyse the experimental results from these populations.[13,14,15,16] Other assumptions related to inference based on the

population model may not be met and are difficult to assess in medical trials where sample sizes tend to be small.

Unlike model-based inference, inference under the randomization model makes no assumptions about the manner in which the samples have been selected. The only necessary criterion is that treatment assignment is randomly performed.

1.2.3 Hypothesis testing and Type I and II errors.

In this thesis we discuss tests under the null hypothesis of no intervention effect, or in other words that intervention does not have an effect on the measured outcome. We'll see in subsequent chapters that the choice of the null hypothesis can actually have important consequences. Under the population and randomization models, it is possible to evaluate two types of error in statistical inference. Type I error refers to the risk of falsely rejecting the null hypothesis, whereas Type II error refers to the risk of falsely accepting the null hypothesis. This latter type of error is closely related to the power of a study to detect a treatment effect and consequently reject the null hypothesis. Researchers must balance the control of these two types of error with the needs of the study. In biomedical research, a false positive inference could result in the promotion of a worthless therapy, and thus there is often more effort put forth to control Type I error. In Chapters 2 to 4 we discuss the validity of tests of the null hypothesis based on their size. Size is defined as the likelihood of a significant treatment effect under the null hypothesis of no treatment effect. It is akin to the Type I error rate for the null hypothesis of no treatment effect.

1.3 Thesis outline

The main focus of this thesis is to examine select design and analytical issues surrounding randomized control trials. The thesis can be divided into two parts. The first part focuses on the validity of using randomization tests and tests based on meta-analysis techniques in the context of cluster randomized trials. Chapter 2 begins with

a discussion of the reasons for making statistical inferences based on a randomization model rather than on a population model. In this chapter we review aspects of Gail's paper "On the design considerations and randomization-based inference for community intervention trials".[9] First the reasons for the use of permutation tests are explained, followed by a review of the theory behind permutation testing to measure treatment effect in both matched and unmatched cluster-randomized intervention trials. The usefulness of permutation tests is examined for sub-optimal study designs through simulation. In some cases Gail's results are affected by imprecision that we have been able to improve upon due to advances in computing power. We also provide additional details on some of the methodological aspects and present simulation results on cases not covered by Gail.

Meta-analysis is typically used to locate, select, and combine a collection of analytic results on a research question in order to integrate the findings. The meta-analytical approach to analyzing cluster-randomized data has been put forth fairly recently, and is the focus of Chapter 3. In this chapter, we review Thompson's proposal to use meta-analytical techniques, as outlined in his paper entitled "The Design and analysis of paired cluster randomized trials: An application of meta-analysis techniques".[17] Whereas Thompson focused on application of random effects modeling to continuous outcomes in matched cluster randomized trials, we present the theory of applying random and fixed effects modeling to a binary outcome. In addition, Thompson focused on estimation, whereas we focused on testing for treatment effects.

Chapter 4 describes the Community Hypertension Assessment Trial (CHAT), which was a program carried out by randomizing family physician practices, where a blood-pressure monitoring strategy was implemented in some of the practices while the other practices maintained their standard approach.[18] The patients in these practices were grouped or clustered within each practice. Permutation tests as well as meta-analytic techniques were applied to the data and the results are compared.

In the second part of the second part of the thesis, we evaluate the appropriateness of using multiple personal digital assistants (PDA) per centre, each with pre-loaded blocked randomization schemes, in multi-centre randomized trials. The use of PDAs in randomization is becoming increasingly attractive and widespread. Due to the dynamic nature of the recruitment process, one in which speed of allocation is of the essence, investigators are seeking to devise a system where PDAs can immediately randomize a new participant without having to access a centralized randomization server. To achieve this, there have been proposals to implement blocked randomization at the PDA level rather than at the level of the centre. In chapter 5, we investigate treatment allocation imbalance that can occur when the blocking is done at the PDA level, both within the centres and at the study level, and discuss the implications on the analysis and inference. The probabilities of imbalance and their potential effects on power are evaluated by simulation under conditions where the number of centres, the number of PDAs and the number of recruits are varied. Note however that although sample size and power are mentioned in the context of design and analytical issues, a detailed discussion on these topics is beyond the scope of this thesis. The benefits of a non-centralized block-randomized design compared to simple randomization are also discussed. Though the issue of allocation imbalance has been investigated extensively, it has not been investigated in the context of multiple PDAs with pre-loaded blocked randomization schemes where a certain level of allocation imbalance is inevitable.

Chapter 2

Randomization-based Inference for Cluster RCTs

This chapter provides an overview of the randomization model and presents the theory for hypothesis testing in both matched and unmatched cluster randomized trials based on the paper by Gail et al. entitled "On design considerations and randomization-based inference for community intervention trials".[9] This chapter also provides additional details on some of the methodological aspects and present simulation results on cases not covered by Gail. A concerted effort was made to explain some of the surprising results in the balanced design scenarios. Finally, the validity of inference based on the randomization model in situations with severe treatment allocation imbalance is discussed.

2.1 Overview

R.A Fisher proposed the randomization model in the early 1930s. However it was E. J. G. Pitman (1938) who showed that randomization tests could be performed without random sampling. In general terms, a randomization test is a permutation test that is based on random assignment to treatment groups. The premise of this model is

that a sample, regardless of the means by which it was obtained, is randomized to two or more experimental groups. Thus, unlike the Population model, under a randomization model, no assumptions are made about the theoretical frequency distribution of a larger population. In addition, assumptions regarding random sampling, normality, homogeneity of variance and of study characteristics are not necessary. Rather, the frequency distribution is determined exactly by randomization of the sample values.

The first use of a randomization test in a GRT was in the Community Intervention Trial for Smoking Cessation (COMMIT).[19,20,21] Gail et al. have since published a paper that discusses the validity of randomization tests in matched and unmatched GRTs based on Type I error rates when testing the null hypothesis of no treatment effect on average.[9] Lets now review the methodology of randomization testing based on this paper.

In the case of independent (unmatched) groups, all possible permutations of the experimental results are carried out. When analyzing the results of a paired-cluster randomized trial, all possible ways of reassigning treatment allocation within the pairs are considered. These two permutation methods are explained in greater detail further on in this chapter. When repeated measurements are made on the experimental units, all possible permutations of the values for each unit are carried out. In all these cases, the outcome of interest is evaluated for each permutation. The permutation results determine a frequency distribution from which it is possible to do significance testing as well as estimation.

The null hypothesis (H_0) that treatments have no differential effects on the treatment groups is tested by calculating the probability of obtaining the observed outcome or one more extreme.

$$P = \frac{No.\ of\ the\ same\ or\ more\ extreme\ outcomes\ as\ that\ observed}{Total\ no.of\ possible\ outcomes}$$

Here P refers to the probability contained in the tails of the frequency distribution, and is used as a guide in accepting or rejecting H_0. Typically, a value of P that is less than 5% provides some evidence that the null hypothesis is not true. If it is less than 1%, it provides strong evidence that the null hypothesis is not true.[14]

A significant challenge to performing permutation tests is the computational power required. The number of permutations increases rapidly as the number of units and groups increase. For instance, with two independent groups of size n_1 and n_2, there are $(n_1+n_2)!/[(n_1)!(n_2)!]$ possible permutations of the sample data. In the case where clusters have been paired, the permutational distribution of the outcome is obtained by considering all 2^J possible ways of reassigning treatment allocation within the pairs, where J is the number of pairs. In some cases, instead of performing an exact permutation test, which is based on all possible permutations, tests are performed on a Monte Carlo random sample of all permutations. The effects of sampling on precision have been found to be minimal and can be calculated through estimation of confidence intervals for P.[9]

Literature has established that, under conditions of severe treatment imbalance, permutation tests under the 'strong' null hypothesis of no treatment effect in *any* cluster, have proper size. Size is defined as the likelihood of a significant treatment effect under the null hypothesis of no treatment effect. It is akin to the Type I error rate for the null hypothesis of no treatment effect. However, often in community intervention trials, the interest does not lie so much in testing the 'strong' null hypothesis of no treatment effect in any community. A community treatment normally would not be implemented if it did not result in an improvement in the *average* community outcome, even though improvement was seen in some clusters. Rather, testing of the 'weak' null hypothesis that treatment on average has no effect is favored. This chapter will describe randomization tests for both the unmatched and matched study designs and present simulation studies under the 'weak' null hypothesis of no treatment effect on average.

2.2 Unmatched Designs

In an unmatched design, J_1 out of J_1+J_2 clusters are randomly selected for intervention, and the J_2 remaining ones serve as control clusters. Let $Y_{ijk}(t)$ denote the response of cohort member k in cluster j on treatment i at time t for $i = 1, 2$; $j = 1, 2, \ldots, J_i$; $k = 1, 2, \ldots, K_{ij}$; and $t = t_0, t_T$. Here $i = 1$ corresponds to intervention and $i = 2$ to control, t_0 corresponds to baseline and t_T to post-intervention. Also, J_i represents the number of clusters in treatment group i. Usually it is not possible to study each cluster member, and so treatment effects are measured on K_{ij} of the n_{ij} members of cluster (i, j).

Let W_{ijk} be a scalar function of $Y_{ijk}(t)$ that summarizes the health effects on cohort member (i,j,k) at t. For example, if we are interested in the change in a characteristic over the course of the trial, we would define $W_{ijk} = Y_{ijk}(t_T) - Y_{ijk}(t_0)$. On the other hand, defining $W_{ijk} = Y_{ijk}(t_T)$ will give us information on the health response at the end of the trial. W_{ijk} can be defined as either a continuous or a discrete measure of change in the outcome of interest. A general model for W_{ijk} is given by

$$W_{ijk} = \mu + d_{ij} + \varepsilon_{ijk} \quad \text{where } \varepsilon_{ijk} \sim \mathrm{N}(0, \sigma^2) \text{ and } d_{ij} \sim \mathrm{N}(\delta_i, \phi_i)$$

Here the variance σ^2 associated with ε_{ijk} represents individual variation within the cluster and d_{ij} represents the effect of treatment i in cluster j. The assumption is made that ε_{ijk} and d_{ij} are mutually independent, and that δ_2, the mean effect of no intervention, is 0. Given K_{ij} for each cluster,

$$W_{ij.} = \frac{1}{K_{ij}} \sum_k W_{ijk}$$

is an unbiased estimate of the expected outcome in cluster (i, j). The variance of $W_{ij.}$ is given by $\mathrm{var}(W_{ij.}) = \phi_i + \sigma^2 / K_{ij}$. The global measure of treatment effect is

$$U = \sum_{j=1}^{J_1} W_{1j.} / J_1 - \sum_{j=1}^{J_2} W_{2j.} / J_2 = W_{1..} - W_{2..}$$

2.2.1 Testing the Null Hypothesis of No Intervention Effect on Average

Under the general model $W_{ijk} = \mu + d_{ij} + \varepsilon_{ijk}$, where $\varepsilon_{ijk} \sim \mathrm{N}(0, \sigma^2)$ and $d_{ij} \sim \mathrm{N}(\delta_i, \phi_i)$, the null hypothesis of no intervention effect on average is defined as $H_0 : \delta_1 = 0$. Based on this definition of the null hypothesis, a treatment effect is measured only as a significant difference in the mean of the d_{1j} , δ_1, rather than in their variance, ϕ_1. Thus, if an intervention only has an effect on the variance of the d_{1j} and not on the mean, no treatment effect will be detected by H_0.

2.2.2 Permutation test

Under the null hypothesis of no treatment effect, we can reallocate the clusters to the treatments and the outcome should not be affected. The permutational distribution of the global treatment effect U is obtained by measuring U for each of the $Z = (J_1+J_2)!/[(J_1!)(J_2!)]$ possible ways of assigning the clusters to treatment and control. Figure 2.1 illustrates how the permutational distribution of U is obtained for the case where $J_1 = J_2 = 2$ (balanced design) and the clusters (1, 2, 3, 4) are randomized to treatments A and B. For the purposes of this example, suppose the values of U have been sorted from the smallest, $U_{(1)}$, to the largest, $U_{(6)}$.

Figure 2.1 Permutations and treatment effects for a balanced unmatched study design with 4 clusters and 2 treatment arms.

Clusters				Treatment effect
1	2	3	4	
A	A	B	B	$U_{(3)}$
A	B	A	B	$U_{(6)}$
A	B	B	A	$U_{(2)}$
B	B	A	A	$U_{(1)}$
B	A	B	A	$U_{(5)}$
B	A	A	B	$U_{(4)}$
W_{i1}	W_{i2}	W_{i3}	W_{i4}	

For a one-sided permutation test, if the observed value of U from our study sample, say U^*, is greater than or equal to the one corresponding to the (1-α) quantile of the permutational frequency distribution, then we reject the null hypothesis. In other words, the observed value of U suggests that intervention (i = 1) leads to larger values of the outcome than does control (i = 2). To illustrate using figure 2.1, a one-sided test at the α = 1/6 level would result in the rejection of H_0 if $U^* >= U_{(6)}$. Similarly, for a two-sided permutation test, we reject H_0 if U^* is either less than or equal to the $U_{(i)}$ corresponding to the α/2 quantile or greater than or equal to the (1-α/2) quantile of the permutational distribution.

One issue related to the discreteness of the permutation distribution of U concerns the choice of a desired significance level for testing the null hypothesis. When the number of clusters is small, it is possible that there is no value of U corresponding to the desired significance level. Consider a trial where there are 4 clusters allocated to treatment A and 5 clusters allocated to control B. In this scenario, assuming there are no ties (duplicate values) there are thus (4+5)!/[(4!)(5!)] = 126 possible permutations and values of U. For a one-sided test, we would reject H_0 at the (1- α) level if the observed U was amongst the 6 highest values of the permutational distribution since 6/126=0.047 and 7/126 = 0.055. As the number of clusters increases, this issue becomes negligible.

2.2.3 Unpaired *t*-test

The unpaired t-test on $J_1 + J_2 - 2$ degrees of freedom can provide a very good approximation to the permutation test under the strong null hypothesis of no treatment effect in any of the intervention groups.[22] The unpaired t-statistic is given by

$$t = U\left\{\frac{(J_1-1)s_1^2 + (J_2-1)s_2^2}{(J_1+J_2-2)}\left(\frac{1}{J_1}+\frac{1}{J_2}\right)\right\}^{-1/2}$$

where $s_i^2 = \frac{1}{(J_i - 1)} \sum_j (W_{ij.} - W_{i..})^2$

If the value of t is larger than the (1-α)-quantile of the t-distribution, the null hypothesis is rejected.

2.2.4 Normal Approximation test

For situations where J_1 and J_2 are large, the normal approximation test can also yield a valid test of H_0. In these cases, the permutational distribution of U will be approximately normal with mean 0 and variance ω given by

$$\omega = \frac{(J_1 + J_2)}{J_1 J_2 (J_1 + J_2 - 1)} \sum_{i,j} (W_{ij.} - W_{...})^2$$

A value of the test statistic $z = U\omega^{-1/2}$ leads to a rejection of the null hypothesis when it is greater than the p-value at the specified (1-α)-quantile.

2.2.5 Validity of Testing the Null Hypothesis of No Intervention Effect on Average

In the ideal study design where there is an even distribution of clusters to each treatment arm and a large number of clusters in the study, the randomization test, unpaired t-test and the normal approximation all yield valid tests for H_0. However, this is not usually the case, and as a result, it may be that one or more of the above-mentioned tests will not be valid for testing the weak null hypothesis of no intervention effect on average. For example, as mentioned previously, due to the discreteness of the permutation distribution, permutation tests on studies with small sample sizes may have a size that exceeds the nominal α level, making the permutation test inappropriate. However, this is not an issue with the *t*-test and normal approximation test as these assume the distribution of U is continuous.

It may also happen that there is an imbalance in the number of clusters assigned to the different treatment arms, or that there are unequal variances in the cluster-level summary statistics ($W_{ij.}$). Unequal variances can arise in many different ways and depending on the magnitude of these differences, can have a negative impact on the validity of hypothesis tests. For example if the intervention (i=1) is effective in some communities but deleterious in others, the mean effect of the intervention would be null however the variance would be increased. As a result $\text{var}(W_{1j.}) > \text{var}(W_{2j.})$. Also, if there is significantly more loss to follow-up in the intervention group than in the control group such that $K_{1j} < K_{2j}$, then $\text{var}(W_{1j.})$ will tend to be greater than $\text{var}(W_{2j.})$. If, on the other hand, $K_{1j} > K_{2j}$, then $\text{var}(W_{1j.})$ will tend to be smaller than $\text{var}(W_{2j.})$. However, as can be seen from the formula for $\text{var}(W_{ij.}) = \phi_i + \sigma^2/K_{ij}$, the imbalance in the cluster sample sizes would have to be severe to result in a significant difference in variances.

Hence, in cases where there are unequal variances in the cluster-level summary statistics, large imbalance in the sample distribution or limited sample sizes, the ability of the randomization test, unpaired t-test, and the normal approximation to yield valid tests for H_0. may be severely compromised. Simulations were used to determine just how misleading these tests can be in such situations.

2.2.6 Simulation exercise

Simulations were run, using R software, to determine the size of the i) permutation test, ii) unpaired t-test and iii) normal approximation test under less than optimal conditions. Using the general model $W_{ijk} = \mu + d_{ij} + \varepsilon_{ijk}$, simulations were performed for the parameters $\mu = \delta_1 = \delta_2 = 0$, $\text{var}(W_{2j.}) = 1$, $\text{var}(W_{2j.}) = \psi$, for $\psi = \frac{1}{5}, \frac{1}{3}, \frac{1}{2}, 1, 2, 3, 5$ and various combinations of values for J_1 and J_2 (see Appendix A.1 for the program).

Firstly, the simulation program randomly generates the cluster-level outcomes, $W_{ij\cdot}$, based on an assumed distribution (normal, t), with a specified variance ψ. The permutational distribution of treatment effect, U, is constructed by generating all the possible outcomes through random assignment of the clusters to the treatment arms in all cases except where $J_1 = 60$ and $J_2 = 30$ where, due to computational constraints, the distribution is constructed from a random sample of 100,000 independent reallocations of the $W_{ij\cdot}$ to treatments. The program then assesses whether the observed U is in the one-sided 5% rejection area of the permutational distribution of U. This process is undertaken 10,000 times. Tables 2.1 and 2.2 present the number of times out of 10,000 the treatment effect using a permutation test, a normal approximation test (z-test) as well as an unpaired t-test, falls in the rejection area when the $W_{ij\cdot}$ follow a normal distribution and a t-distribution respectively.

Estimates that fall outside of the 95% confidence interval for the nominal size (CI = [457, 543]), are considered significantly different from nominal size at the 5% level. The interval was calculated as follows:

$$CI = n \times \left(p \pm 1.96 \sqrt{\frac{p(1-p)}{n}} \right),$$

where n=10,000 is the number of times the simulation was performed and $p = 0.5$ is the Bernoulli probability of success for unit k in cluster (i, j).

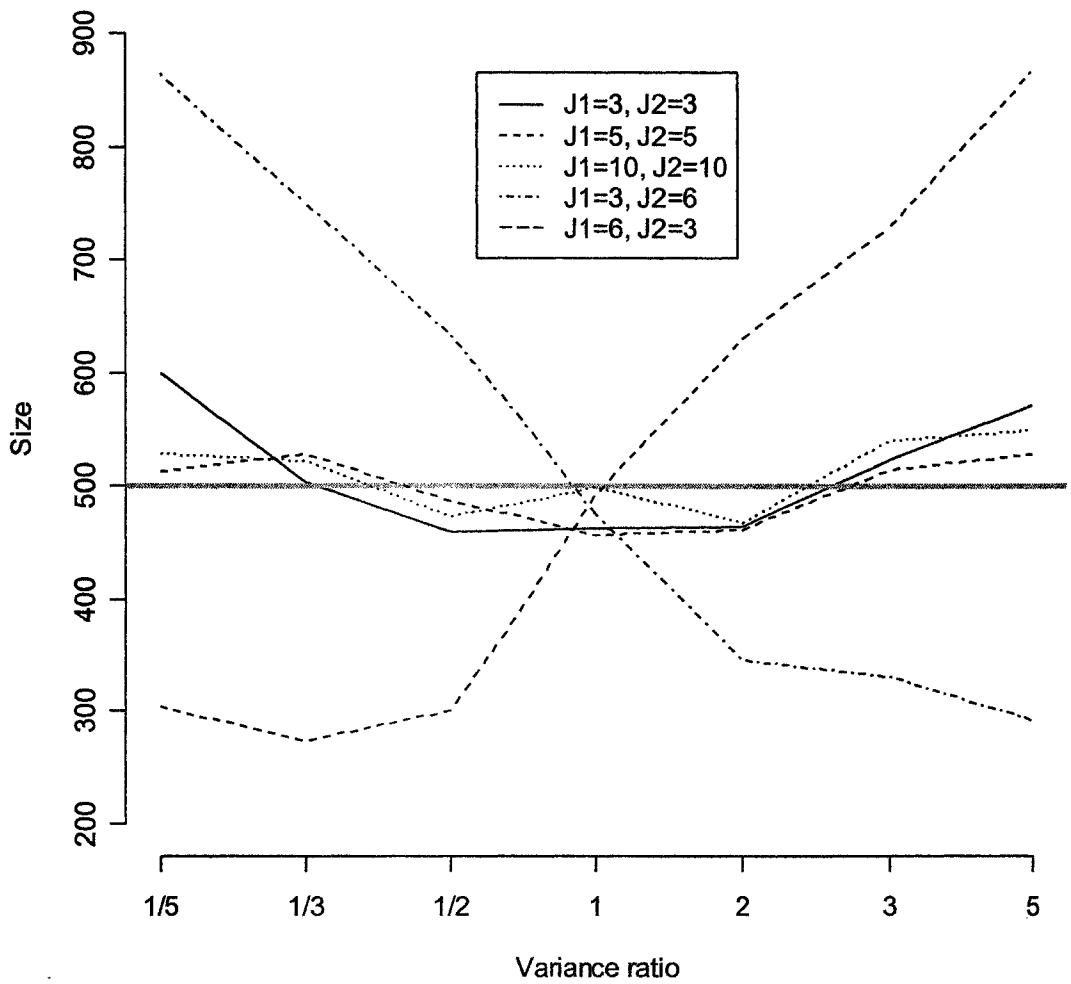
The simulation results indicate that with balanced designs the permutation tests are for the most part close to nominal sizes, even when the number of clusters is small (J_1=J_2=3) and there are unequal variances in the cluster-level summary statistics. Notice how the size for all three tests is greater for extreme values of ψ but near nominal levels for ψ =1. As J_1 and J_2 increase this pattern diminishes and the sizes become quite stable across the values of ψ .

Table 2.1 Estimated size times 10,000 of the one-sided permutation test (upper), unpaired *t*-test (middle) and normal approximation test (bottom) for combinations of variance ratios $\psi = \text{var}(W_{2j.})/\text{var}(W_{1j.})$ and number of clusters J_1 and J_2. The $W_{ij.}$ follow a normal distribution with mean 0 and $\text{var}(W_{1j.})=1$, $\text{var}(W_{2j.})= \psi$.

Number of Clusters		ψ						
Intervention	Control							
J_1	J_2	1/5	1/2	1/3	1	2	3	5
3	3	600	503	460	463	464	523	571
		688	565	517	501	553	598	642
		608	533	479	485	499	543	570
5	5	513	528	487	457	462	514	528
		610	586	526	493	493	543	634
		551	551	507	500	504	547	546
10	10	528	522	474	499	468	540	548
		553	527	513	476	503	512	544
		529	523	477	504	471	545	553
3	6	864	749	632	475	346	331	292
		1320	1058	769	497	333	277	228
		1068	883	714	523	361	321	263
6	3	304	274	300	493	630	728	866
		242	261	311	541	780	1030	1313
		279	285	308	516	712	865	1034

For unbalanced designs where $J_1 < J_2$, the sizes of all three tests are well above nominal levels when the treatment increases the variance, i.e. $\psi < 1$. Similarly, when $J_1 > J_2$ and $\psi > 1$ the sizes of all three tests are well above nominal levels. Conversely, the sizes are well below nominal levels when either $J_1 < J_2$ and $\psi > 1$, or when $J_1 > J_2$ and $\psi < 1$. The trends associated with the test sizes across the increasing values of ψ for the various combinations of treatment allocation are presented in Figure 2.2.

Figure 2.2 Trends in estimated size times 10,000 of the one-sided permutation test for values of $\psi = \text{var}(W_{2j.})/\text{var}(W_{1j.})$ ranging from 1/5 to 5.



Since in practice treatment effects are rarely normally distributed but rather tend to be skewed, the simulation was also run under the assumption that the $W_{ij.}$ follow a t-distribution with 20 degrees of freedom. The results are presented below in Table 2.2.

Table 2.2 Estimated size times 10,000 of the one-sided permutation test (upper), unpaired t-test (middle) and normal approximation test (bottom) for combinations of variance ratios $\psi = \text{var}(W_{2j.})/\text{var}(W_{1j.})$ and number of clusters J_1 and J_2. The $W_{ij.}$ follow a t-distribution with 20 degrees of freedom.

Number of clusters		ψ						
Intervention J1	Control J2	1/5	1/2	1/3	1	2	3	5
3	3	554	562	526	480	476	524	573
		592	551	518	520	499	554	681
		542	561	513	469	491	509	576
5	5	530	518	478	427	527	497	559
		575	575	520	504	547	559	654
		564	564	493	464	552	520	591
10	10	538	513	527	531	484	517	539
		588	479	515	522	500	520	567
		544	519	533	537	490	518	545
3	6	865	788	622	406	330	304	286
		1314	1021	830	440	321	265	238
		1024	924	701	452	326	296	269
6	3	279	302	334	497	649	726	856
		210	281	316	517	766	976	1333
		256	283	331	539	732	842	1036

Comparison of Tables 2.1 and 2.2 shows that, regardless of whether the $W_{ij.}$ are approximately normally distributed or follow a skewed t-distribution with 20 degrees of freedom, the size of permutation, t- and normal approximation tests tend to be similar and the sizes all approach nominal levels if $\psi = 1$ or if the design is balanced. In the context of community intervention trials, these results highlight the importance of a balanced design when testing the 'weak' null hypothesis of no treatment effect on average. Another way to express it is to say that the permutation test will yield nominal sizes unless there is a combination of large allocation imbalance to the treatment and control groups and large differences in the variances for these groups. Let us now consider pair-matched designs.

2.3 Pair-matched Designs

Pair-matching is a very common strategy in epidemiological research. The popularity of this method is based on the concept that it takes into account potentially important confounders. If the matching factors are well selected, pair matching may result in an increase in power compared to an unmatched design. Popular matching factors for GRTs are cluster size and geographic area. Disadvantages of pair-matching include the uncertainty in the choice of matching criteria, and possible confounding of the effect of intervention due to the natural variation between two paired clusters.[4,11]

In pair-matched designs, cluster $(1, j)$ is paired with cluster $(2, j)$ based on specified characteristics, where (i, j) refers to the combination of treatment (i) and pair (j). Let Y_{ijk} denote the response of cohort member k in pair j on treatment i at time t for $i = 1,2; j = 1,2, \ldots J; k = 1,2, \ldots K_{ij}$; and $t = t_0, t_T$. Here $i = 1$ corresponds to intervention and $i = 2$ to control, t_0 corresponds to baseline and t_T to post-intervention. Following the same reasoning as in the previous section on unmatched designs, W_{ijk} is a scalar function of $Y_{ijk}(t)$ that summarizes the health effects on cohort member (i,j,k) for t. A general model for W_{ijk} is given by

$$W_{ijk} = \mu + \alpha_j + d_{ij} + \varepsilon_{ijk}$$

where α_j is a fixed pair effect, $\varepsilon_{ijk} \sim \mathrm{N}(0, \sigma^2)$ and $d_{ij} \sim \mathrm{N}(\delta_i, \phi_i)$ is a random treatment effect. Given K_{ij} for each cluster,

$$W_{ij.} = \frac{1}{K_{ij}} \sum_k W_{ijk}$$

is an unbiased estimate of the expected cohort response. Treatment effect in each pair j is then calculated as $U_j = W_{1j.} - W_{2j.} = d_{1j} - d_{2j} + \varepsilon_{1j.} - \varepsilon_{2j.}$. The global measure of treatment effect U is the average of all the U_j across all J pairs:

$$U = \frac{1}{J}\sum_{j=1}^{J} U_j$$

2.3.1 Testing the Null Hypothesis of No Intervention Effect on Average

By design, there is no allocation imbalance in the numbers of clusters assigned to treatment or control when pair-matching is used. Hence, the tests under a pair-matched design are more robust than those under an unbalanced unmatched design. Using the general model for W_{ijk} described above, the 'weak' null hypothesis of no intervention effect on average is defined as $H_0 : \delta_1 = \delta_2$, which is equivalent to $H_0 : \mathrm{E}(U_j) = 0$. In addition, the $U_j = W_{1j.} - W_{2j.} = d_{1j} - d_{2j} + \varepsilon_{1j.} - \varepsilon_{2j.}$ are symmetrically distributed about $\delta_1 - \delta_2$ and they are independent under the following three assumptions[9]:

1. $\{d_{ij}\}$ and $\{\varepsilon_{ijk}\}$ are independent;
2. d_{1j} and d_{2j} have the same distribution or are symmetrically distributed;
3. $\varepsilon_{1j.}$ and $\varepsilon_{2j.}$ have the same distribution or are symmetrically distributed about a common location.

2.3.2 Permutation test

The permutational distribution of the outcome U is generated through reallocation of the treatments within each pair all 2^J possible ways. Table 2.3 illustrates how the permutation of treatment allocation is carried out for a case where there are 2 pairs of clusters and two treatment options. Outcomes are measured as a difference in risk. Let clusters 1 and 2 be paired, as well as 3 and 4, and let A represent intervention and B control. The 2^2 permutations and outcomes are then given as follows:

Table 2.3 Permutations of treatment allocation for a balanced matched study design with 2 pairs of clusters (j) and 2 treatment arms (i), A and B.

Cluster pairs (j)						Global treatment effect
$j=1$			$j=2$			
1	2	Treatment effect	3	4	Treatment effect	
A	B	U_1	A	B	U_2	$(U_1+U_2)/2$
A	B	U_1	B	A	$-U_2$	$(U_1-U_2)/2$
B	A	$-U_1$	A	B	U_2	$-(U_1-U_2)/2$
B	A	$-U_1$	B	A	$-U_2$	$-(U_1+U_2)/2$
W_{i1}	W_{i1}	$W_{A1}-W_{B1}$	W_{i2}	W_{i2}	$W_{A2}-W_{B2}$	U

From this table we can see that each U_j will have two values. As a result, the permutational distribution of U is perfectly symmetrical. By fixing a significance level α for our critical region, we can test the null hypothesis based on our observed treatment effect, as described in the section on unmatched designs. In the case of pair-matched studies, due to the discreteness of the permutation distribution of U, a trial needs at least 5 pairs in order to have a critical region with a significance level less than or equal to 0.05 ($1/2^4 = 0.0625$; $1/2^5 = 0.03125$). In the case where there are 5 pairs of clusters, only the largest value leads to rejection of the null hypothesis at the 0.03125 level. Thus, one would expect the size of the permutation test to be very close to its nominal size 0.03125.

2.3.3 Paired t-test

The paired t statistic is given by

$$t = U\left\{s^2/J\right\}^{-1/2}$$

where $s^2 = \frac{1}{(J-1)}\sum_j (U_j - U_\cdot)^2$

If the value of t is larger than the (1-α)-quantile of the t-distribution, the null hypothesis is rejected. Typically, one would look to significance tables with J-1

degrees of freedom to determine the significance of the t statistic. Efron's work suggests that under symmetry conditions, this test will produce excellent approximation to the permutational distribution of U.[23] However, Edgington states that "in the absence of random sampling, use of the t table to determine significance is valid only to the extent that the significance given by the t table approximates that given by the randomization test procedure."[14] As a result, he suggests deriving a theoretical distribution for t by calculating its value for each of the 2^J possible permutations and determining the probability of the observed t statistic.

2.3.4 Validity of Testing the Null Hypothesis of No Intervention Effect on Average

In section 2.2.5 we discussed how unequal variances in the cluster-level summary statistics (W_{ij}) can compromise the validity of tests for the null hypothesis of no treatment effect on average in the case of unmatched permutation tests. This is not so much of an issue for the pair-matched design by virtue of the fact that the U_j are symmetrically distributed around about $\delta_1 - \delta_2$. For instance, say d_{1j} and d_{2j} are both normally distributed however they have different variances. Under the assumptions presented in section 2.3.1, namely that the ε_{ijk} are symmetrically distributed about a common point, the U_j are symmetrically distributed about $\delta_1 - \delta_2$. Thus, a permutation test based on $H_0 : \delta_1 = \delta_2$ will have nominal size.

However, though the pair-matched design is more robust than the unmatched design, there are situations that make the use of permutation tests and t-tests invalid. For instance, if the intervention causes the distribution of U_j to become skewed or if there is a large imbalance in the number of subjects in the intervention and control clusters, then the paired permutation test may not be close to nominal size under H_0. To illustrate, lets say W_{ijk} is a Bernoulli variate with a low associated probability of success $p = 0.1$. In addition the intervention cluster is composed of K_{1j}=15 subjects, whereas the control cluster is composed of $K_{2j} = 150$ subjects. Thus, $W_{1j.}$ and $W_{2j.}$ would have different distributions with mean p and variance $p(1\text{-}p)/K_{ij}$ (0.006 and

0.0006 respectively) that are skewed toward the left. Since $W_{2j.}$ is based on a much larger number of units, its distribution is more symmetrical around 0.1. As a result, $U_j = W_{1j.} - W_{2j.}$ will also be skewed. However, since the null hypothesis is solely based on the mean of the distribution of U_j and not on its variance, a treatment effect will not be detected. In order to assess the effect of unbalanced sample sizes and rare events, simulations were undertaken which are discussed in the next section.

2.3.5 Simulation exercise

Size for binary outcomes using permutation tests and *t*-tests were assessed by means of simulation using R software for combinations of the number of pairs, $J = 5$, 10, 20, 40; the cluster sizes, K_{ij}; and the probability of a successful outcome, $p = 0.1$, 0.5. The simulation program (see Appendix A.2) randomly generates study data for J pairs of clusters according to the parameters K_{ij} and p, and calculates a global treatment effect. Since the outcome for each subject in the cluster is a Bernoulli variate with probability of success p, the number of successes for a cluster of size K_{ij} follows a Binomial distribution with parameters (K_{ij}, p). The permutational distribution of treatment effect is constructed by generating all the possible permutations, or a random subset thereof, and calculating the global measure of effect for each of these. Due to the computational limitations of permuting the outcomes, a complete set of possible permutations was generated when 2^J was less than 100,000. Otherwise, 100,000 permutations were randomly generated. The program then assesses whether the observed treatment effect is in the one-sided 5% rejection area of the permutational distribution. This process was undertaken 10,000 times. Table 2.4 presents the number of times out of 10,000 the observed outcome was in the one-sided 5% rejection area.

Due to some surprising results in the case where there are 5 pairs, exact calculations of size were performed. Whereas Gail et al. briefly summarize how the exact calculations were done, what follows is a more detailed explanation. As discussed in

the previous section, for a permutation test with 5 pairs, only the largest of the 32 (2^5) possible values for U resulting from permutations of the data leads to rejection at the 0.03125 level. The largest value of

$$U = \frac{1}{J}\sum_j \left(W_{1j.} - W_{2j.}\right)$$

occurs when $W_{1j.} > W_{2j.}$ for all j = 1, ...,5. Therefore, under the assumption that all pairs are identical,

$$\begin{aligned} P(W_{1j.} \overset{\forall j}{>} W_{2j.}) &= P(X/K_{1j} > Y/K_{2j})^5 \\ &= \left(\sum_y P(X/K_{1j} > y/K_{2j} \mid Y = y)P(Y = y)\right)^5 \\ &\overset{X,Y \text{ indep}}{=} \left(\sum_y P(X/K_{1j} > y/K_{2j})P(Y = y)\right)^5 \end{aligned}$$

where $X \sim \text{binomial}(p, K_{1j})$ and $Y \sim \text{binomial}(p, K_{2j})$. The exact calculation is incorporated into the table for J=5.

Table 2.4 Estimated size times 10,000 of a one-sided permutation test (upper) and paired t-test (middle) and exact calculation (bottom) for $W_{ij.}$ following a binomial distribution with parameters K_{ij} and p.

Cluster sizes		Number of pairs, J							
Intervention	Control	5		10		20		40	
K_{1j}	K_{2j}	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$
100	9	287	324	753	533	740	517	648	487
		630	515	626	525	607	504	521	490
		320	313						
100	29	373	324	601	511	612	490	543	532
		520	520	508	527	553	480	503	495
		405	313						
100	49	249	308	555	520	559	497	511	497
		510	506	495	494	509	500	492	511
		274	313						
100	99	188	334	463	552	511	510	499	522
		489	507	496	527	504	473	500	548
		211	313						
400	39	356	306	637	505	603	479	550	498

Cluster sizes		Number of pairs, J							
Intervention	Control	5		10		20		40	
K_{1j}	K_{2j}	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$
		522	504	548	480	479	472	514	549
		376	313						
400	119	342	333	569	464	545	486	577	531
		529	486	515	487	497	522	485	511
		353	313						
9	100	260	322	298	525	342	528	338	489
		630	544	598	540	522	480	518	511
		305	313						
29	100	221	305	380	464	400	529	466	496
		537	511	499	509	498	518	498	539
		238	313						
49	100	303	275	478	512	460	527	432	503
		476	532	505	530	510	520	501	496
		355	313						
99	100	425	332	516	437	496	503	498	512
		516	532	503	509	502	497	489	533
		450	313						
39	400	220	325	359	528	400	454	450	502
		537	504	502	516	506	492	498	539
		258	313						
119	400	256	294	464	492	472	468	459	487
		480	460	479	521	491	476	472	522
		276	313						
		119							
9	9	1	649	676	612	554	535	444	513
		392	490	480	519	534	489	479	493
		42	112						
29	29	638	459	629	547	558	514	562	512
		521	510	506	497	500	499	500	512
		118	180						
39	39	626	484	608	535	532	526	513	536
		517	500	495	503	498	513	487	508
		137	195						
49	49	571	490	582	578	583	482	486	524
		516	488	490	543	523	463	480	518
		151	206						
100	100	452	434	541	536	513	538	519	507
		492	547	530	502	500	551	487	479
		190	234						

Cluster sizes		Number of pairs, J							
Intervention	Control	5		10		20		40	
K_{1j}	K_{2j}	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$	$p = 0.1$	$p = 0.5$
200	200	384	397	543	515	485	501	477	547
		475	528	465	452	481	541	475	535
		221	255						
300	300	439	383	526	527	513	540	491	521
		503	512	530	469	494	523	479	566
		236	265						
400	400	402	362	556	532	526	481	477	523
		507	476	521	502	481	513	512	507
		246	271						
Nominal size									
Permutation test		312	312	498	498	500	500	500	500
Paired t-test		500	500	500	500	500	500	500	500

The paired t-test tends to exceed nominal levels when K_{1j} is much larger than K_{2j} when $p = 0.1$. However, the extent with which the tests exceed the nominal levels decreases with an increase in the sample size as well as an increase in the value of J. The extent to which the nominal levels are exceeded also diminishes as the value of J increases. This is in accord with the central limit theorem which states that the size of paired t-tests reaches nominal levels as the number of pairs increases asymptotically. When K_{1j} is much less than K_{2j}, the size of the permutation test is below nominal levels.

For unbalanced designs with $p = 0.1$, the paired permutation test generally exceeds nominal levels when $K_{1j} = 100$ and $K_{2j} = 9, 29$. However, for the case $K_{1j} = 400$ and $K_{2j} = 39$, even though there is a large difference in sample sizes, the size of the permutation test is near nominal levels. This indicates that it is the combination of sample size differences and small sample size that produces test results well above nominal levels when the probability of success is less than 0.5. In reality such a large imbalance in sample sizes should not occur as a study would not be designed in such a way and it is unlikely that a study would have such an extreme differential loss to follow-up in the treatment groups. For $p = 0.5$, the permutation test yields sizes near

nominal levels, even with important sample size imbalance. This is because of the symmetrical distribution of the U_j.

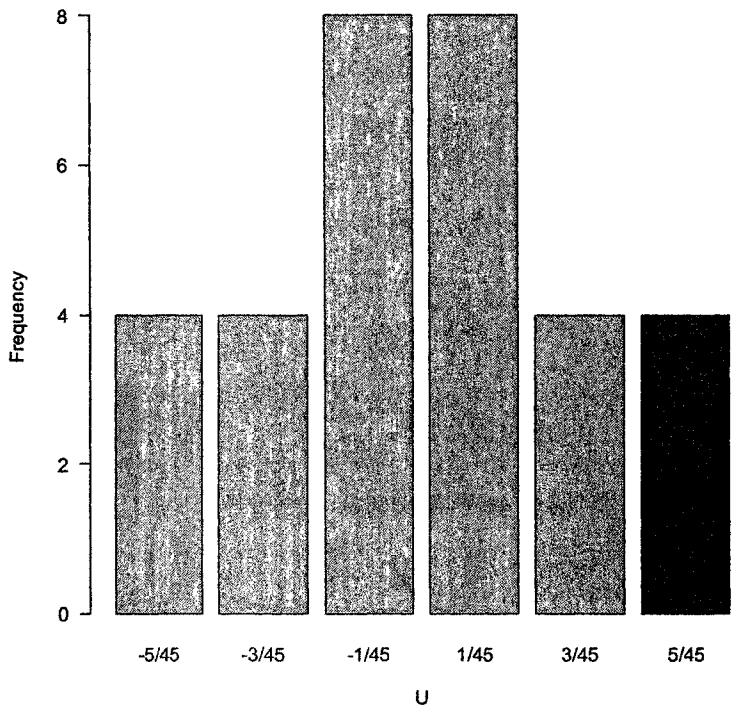
For balanced designs where $J = 5$, the size of the permutation test is consistently well above nominal levels, and is not consistent with the exact calculations. One explanation for this is that the simulation is based on the assumption that all possible permutations of the outcomes result in unique values of U. In other words, the permutational-distribution of U is assumed to be composed of 2^J unique values. This, however, is rarely (never really) the case. The proportion of duplicate values of U is increased when the number of clusters is small and the probability of success is small (i.e. $J = 5$ and $p = 0.1$). To illustrate, consider the case where $K_{1j} = K_{2j} = 9$, $J = 5$ and $p = 0.1$. The number of successes for each pair is randomly generated (n_{1j}, n_{2j}) and is as follows: (1,1), (2,0), (1,0), (0,2), (1,1). Thus, in the first pair, one event was observed in treatment group 1, and one event was also observed in treatment group 2. Note that due to the fact there are two pairs with equal numbers of events in each treatment group, the number of unique permutations is reduced from 2^5 to 2^3. As a result, the maximum possible number of unique values of U is also reduced to 2^3. Table 2.5 presents the unique permutations along with their corresponding values of treatment effect for each pair (U_j)and across all pairs (U). The unique values of U are shaded.

Table 2.5 Values of U_j and U for the unique permutations (I to VIII) resulting from the random generating of events for the case $K_{1j} = K_{2j} = 9$, $J = 5$ and $p = 0.1$.

Permutation No.	Pairs (J)					Treatment
U_j	j=1	j=2	j=3	j=4	j=5	effect (U)
I	(1,1)	(2,0)	(1,0)	(0,2)	(1,1)	
U_j	0	2/9	1/9	-2/9	0	1/45
II	(1,1)	(2,0)	(1,0)	(2,0)	(1,1)	
U_j	0	2/9	1/9	2/9	0	5/45
III	(1,1)	(2,0)	(0,1)	(0,2)	(1,1)	
U_j	0	2/9	-1/9	-2/9	0	-1/45
IV	(1,1)	(2,0)	(0,1)	(2,0)	(1,1)	
U_j	0	2/9	-1/9	2/9	0	3/45
V	(1,1)	(0,2)	(1,0)	(0,2)	(1,1)	
U_j	0	-2/9	1/9	-2/9	0	-3/45
VI	(1,1)	(0,2)	(1,0)	(2,0)	(1,1)	
U_j	0	-2/9	1/9	2/9	0	1/45
VII	(1,1)	(0,2)	(0,1)	(0,2)	(1,1)	
U_j	0	-2/9	-1/9	-2/9	0	-5/45
VIII	(1,1)	(0,2)	(0,1)	(2,0)	(1,1)	
U_j	0	-2/9	-1/9	2/9	0	-1/45

In the case presented, there are only 6 unique values of U that result from the permutation of treatment assignments, a far cry from the expected 32. The problem with ties in values of U is that it makes it impossible to specify the level of a permutation test with any certainty. Based on the assumption that there are no ties and using a rejection criteria of 5%, with 5 pairs of clusters, only an observed value corresponding to the largest in the permutational distribution of U would lead to rejection of the null hypothesis. The following figure help us visualize how this criteria performs in the case under study. Figure 2.3 shows the distribution of U, and the rejection area (in black) when a level of 5% is used.

Figure 2.3 Histogram showing the distribution of the values of treatment effect U for the case $K_{1j} = K_{2j} = 9$, . $J = 5$ and $p = 0.1$.



In the example at hand, rejection of the null when the observed treatment effect corresponds to the largest value in the permutational distribution leads to rejection 12.5% (4/32) of the time rather than the planned 3.125% (1/32) of the time. This helps to explain why the test size was so high (1191) in Table 2.4 for the case $K_{1j} = K_{2j} = 9$ with p=0.1. This simple example illustrates the danger of using permutation tests in situations where there may be many ties in the values of U. Ties occurs with greater probability in balanced designs, when the number of pairs is very small and events occur with either very low or very high probability. Interestingly, the prevalence of duplicate values of U when the number of pairs is small can be high even though the size of the clusters may be quite large. This is seen in Table 2.4 where the size for the permutation test when $J = 5$; $K_{1j} = K_{2j} = 100$ was well above both the nominal size and the exact size calculation. In conclusion, the assumption that there are no ties in

the distribution of U can be very misleading when conducting a permutation test, as it could result in the detection of a treatment effect that is not there.

2.4 Summary

In this chapter, we presented an overview of the theory for randomization tests on binary outcome data from both matched and unmatched cluster randomized trials where the outcome is measured as a difference in proportions. Simulations have shown that for unmatched trial designs, the permutation test will yield nominal sizes unless there is a combination of large allocation imbalance to the treatment and control groups and large differences in the variances for these groups. In the case of matched designs, the size of permutation tests may deviate substantially from nominal sizes under the following combination of factors: 1) the number of pairs is too small, 2) there is consistently a large sample imbalance in the treatment and control groups and 3) the individual response data is skewed, such as when the events are either rare or very common. Even when there is no imbalance in the treatment allocation, it is very possible that the size of permutation tests not attain nominal levels when the number of cluster is too small, the cluster sizes are also small and the response data is skewed.

In the next chapter, we will investigate the usefulness of applying meta-analytical techniques, which rely on parametric assumptions, to the analysis of pair-matched studies of binary outcomes.

Chapter 3

Meta-analytical Approach to Cluster RCTs

In this chapter we will present an overview of meta-analysis and discuss how meta-analytical techniques may be applied to the analysis of paired-cluster randomized trials with a binary outcome, based on the paper by Thompson et al., "The design and analysis of paired cluster randomized trials: An application of meta-analysis techniques"[17]. Through simulations comparable to those presented in Chapter 2, the effect of severe treatment imbalance on test size are presented and evaluated. The chapter concludes with a discussion of possible explanations for some surprising results.

3.1 Overview

Meta-analysis involves locating, selecting, and combining a collection of analytic results on a research question in order to integrate the findings. This type of analysis is of great interest in medical and epidemiological research as there are often multiple published studies on a topic, which on their own may not give clear evidence of intervention effect, however combined may give added weight to a conclusion. The

first step is to calculate a summary statistic for each study, and then to pool the data as a weighted average to obtain an overall treatment effect.

The main concern when pooling information from different studies arises from the diversity of study designs. Some may be highly controlled randomized experiments whereas others are much less controlled and perhaps not even randomized. Differences in sample sizes and population characteristics result in heterogeneous sampling errors. To take these into account, approaches have been devised that allow for treatment effects to vary across studies.

In clinical trials where pairs of clusters are matched and subsequently randomized to either treatment or control, it is vital that the between-cluster component of variation be taken into consideration in sample size calculations and the analysis strategy. For this purpose, Thompson et al. proposed the use of techniques developed for meta-analysis in which treatment effects are allowed to vary across studies. By application of these techniques, it is possible to take into consideration intra-cluster variability and derive confidence intervals for the overall treatment effect.

There are several statistical meta-analytical techniques that have been developed to combine individual study results. Specifically, three techniques will be expounded in this chapter. The first two techniques are based on a fixed effects model, which assumes that the true treatment effect is the same for all studies. The difference lies in the method for estimating the true effect of intervention. We will consider the inverse variance method as well as the Mantel-Haenszel method. The third technique is based on the random effects model which allows the true effect to vary from one study to the other.

3.2 Model Structure - Notation

In a paired cluster randomized trial, each pair can be thought of as a study for which it is possible to evaluate an effect size. The information from each pair can

then be combined using meta-analytic techniques to evaluate the presence of an overall effect. Thus, consider a grouping of k paired clusters, where the j^{th} has an estimated effect size Y_j and a true effect size of θ_j. A general model is given by

$$Y_j = \theta_j + e_j \quad \text{where } e_j \overset{d}{=} \mathrm{N}(0, \sigma_j^2), \quad j = 1, 2, ..., k$$

The estimate of effect size Y_j for each pair can be any measure of effect as long as the assumption that Y_j is normally distributed with mean θ_j and variance σ_j^2 is appropriate. For example, Y_j can represent the effect of intervention in pair j as measured by a difference in means, or a log-odds ratio. This chapter specifically considers binary outcomes where treatment effect is measured as a difference in the means.

Let W_{ijk} represent a Bernoulli variable with a probability of success p. In addition, suppose all W_{ijk} have same probability p of success. In this case,

$$\sum_k W_{ijk} \sim \text{Binomial}(n_{ijk}, p), \text{ with } \mathrm{E}[W_{ij.}] = p \text{ and } \mathrm{Var}(W_{ij.}) = \frac{p(1-p)}{n_{ij}},$$

where $W_{ij.}$ is the average of the W_{ijk} in treatment i, pair j. By defining the treatment effect as a risk difference, we obtain

$$Y_j = \sum W_{2j.} - \sum W_{1j.}, \text{ with } \mathrm{E}[Y_j] = 0 \text{ and } \mathrm{Var}(Y_j) = p(1-p)\left(\frac{1}{n_{1j}} + \frac{1}{n_{2j}}\right).$$

For both the fixed and random effects models, the overall estimate of treatment effect can be written as a weighted average of the Y_j.

$$\hat{\mu} = \frac{\sum \hat{\omega}_j Y_j}{\sum \hat{\omega}_j}.$$

What differentiates the fixed model from the random model is the calculation method of the weight $\hat{\omega}_j$.

3.3 Testing the Null Hypothesis of No Treatment Effect On Average

When treatment effect is measured as a risk difference, the strong null hypothesis of no treatment effect in any pair is defined as H_0: $\theta_j = 0$. Hence, absence of treatment effect measured as a risk difference in a pair would be evidenced by the fact that the confidence interval contained 0. The weaker null hypothesis of no treatment effect on average is defined as H_0: $\mu = 0$. In this case, absence of treatment effect measured as a risk difference in a pair would be evidenced by the fact that the confidence interval for the global treatment effect contained 0.

3.4 Fixed Effects Model

A fixed effects model assumes that the treatment effect $\theta_j = \mu$ for all $j = 1, 2, ..., k$, or in other words that it is constant across each pair of clusters. Hence, by writing Y_j as the effect of intervention in pair j and μ as the overall effect of intervention, then

$$Y_j = \mu + e_j \quad \text{where } e_j \overset{d}{=} \mathrm{N}(0, \sigma_j^2), \quad j = 1, 2, ..., k .$$

As seen in a preceding section, the estimator of μ is a weighted average of the Y_j. There are many different methods for calculating the weights, $\hat{\omega}_j$. We will review two of these: the first is calculated using the inverse of the variance of Y_j, while the second employs a constant weight.

3.4.1 Inverse Variance

Under this model, the optimal weights are calculated using the inverse variance method, where $\omega_j = 1/\sigma_j^2$. Since in practice we don't know the value of σ_i^2, an estimated variance based on the observed study values is used to estimate μ and $\mathrm{var}(\hat{\mu})$. Thus, our estimators are

$$\hat{\mu} = \frac{\sum \hat{\omega}_j Y_j}{\sum \hat{\omega}_j} = \sum \frac{Y_j}{\hat{\sigma}_j^{\,2}} \Big/ \sum \frac{1}{\hat{\sigma}_j^{\,2}} \quad \text{and} \quad \hat{\text{var}}(\hat{\mu}) = \frac{1}{\sum \hat{\omega}_j} = \left(\sum \frac{1}{\hat{\sigma}_j^{\,2}} \right)^{-1}$$

These estimators are then used to calculate confidence intervals for the global treatment effect and subsequently determine the significance of observed effect.

3.4.2 Mantel-Haenszel

The Mantel-Haenszel method of calculating weights is more robust when data are sparse, in terms of low event rates or trial size. However, this method can only be used with binary outcome data. For randomized control trials with 2 treatment arms and a binary outcome, the results can be presented the following way:

Table 3.1 2x2 table of counts

Pair *j*	**Event**	**No event**	**Group size**
Intervention ($i = 1$)	a_j	b_j	n_{1j}
Control ($i = 2$)	c_j	d_j	n_{2j}

When treatment effect is calculated as the risk difference, the weight ω_j is given by

$$\omega_j = \frac{n_{1j} n_{2j}}{N_j},$$

where n_{1j} and n_{2j} are the number of participants in treatment group 1 and 2 respectively, and N_j is the total number of participants in the study ($N_j = n_{1j} + n_{2j}$). As a result, the estimates of μ and var($\hat{\mu}$) are given by

$$\hat{\mu} = \sum \frac{n_{1j} n_{2j}}{N_j} Y_j \Big/ \sum \frac{n_{1j} n_{2j}}{N_j} \text{ and } \text{var}(\hat{\mu}) = \Gamma / \text{K}^2, \text{ where}$$

$$\Gamma = \sum_i \left(\frac{a_i b_i n_{2i}^3 + c_i d_i n_{1i}^3}{n_{1i} n_{2i} N_i^2} \right) \text{ and } \text{K} = \sum_i \left(\frac{n_{1i} n_{2i}}{N_i} \right)$$

These estimators are then used to calculate confidence intervals for the global treatment effect and subsequently determine the significance of observed effect.

3.5 Random Effects Model

The random effects model allows pairs to vary both in the estimated effect and the true effect. The main assumption is that the true effects θ_j are normally distributed about the overall effect μ. By writing Y_j as the effect of intervention in pair j and μ as the overall effect of intervention, then

$$Y_j = \theta_j + e_j \quad \text{where } e_j \overset{d}{=} \mathrm{N}(0, \hat{\sigma}_j^{\,2})$$

and

$$\theta_j = \mu + \varepsilon_j \quad \text{where } \varepsilon_j \overset{d}{=} \mathrm{N}(0, \tau^2)$$

This is a two-stage hierarchical random effects model, where the error terms e_j and ε_j are assumed to be independent, and with a variance between pairings expressed as τ^2. Thus, the random effects model can also be written as

$$Y_i = \mu + \varepsilon_j + e_j \quad \text{where } \varepsilon_j \overset{d}{=} \mathrm{N}(0, \tau^2) \text{ and } e_j \overset{d}{=} \mathrm{N}(0, \hat{\sigma}_j^{\,2})$$

and subsequently,

$$Y_j \overset{d}{=} \mathrm{N}(\mu, \hat{\sigma}_j^2 + \tau^2)$$

In cases where there is heterogeneity between pairs, it is recommended to use the random effects model in lieu of the fixed effects model.[24,25] Estimates of μ and $\mathrm{var}(\hat{\mu})$ are obtained in much the same way as for the fixed effects model. Hence we define $\hat{\omega}_j = 1/(\hat{\sigma}_j^{\,2} + \tau^2)$, giving

$$\hat{\mu} = \frac{\sum \hat{\omega}_j Y_j}{\sum \hat{\omega}_j} = \sum \frac{Y_j}{(\hat{\sigma}_j^{\,2} + \tau^2)} \Big/ \sum \frac{1}{(\hat{\sigma}_j^{\,2} + \tau^2)}$$

with variance

$$\hat{\text{var}}(\hat{\mu}) = \frac{1}{\sum \hat{\omega}_j} = \left(\sum \frac{1}{(\hat{\sigma}_j^{\,2} + \tau^2)} \right)^{-1}$$

Once again, in practice the value of τ^2 is not known, and must therefore be estimated in order to obtain the desired estimate of overall effect. Methods of estimating τ^2 include the moments based estimator of DerSimonian and Laird,[24] as well as confidence intervals based on likelihood methods using the maximum and profile likelihood functions. For the purposes of the simulations, DerSimonian and Laird's estimate was applied. Details on this estimate are provided in the next section.

There are some limitations associated with the random effects model. Firstly, the proper application of the model rests on the validity of the multiple normality assumptions. The main assumption associated with application of the random effects model is that the true effects θ_j within each cluster pair are normally distributed about the overall effect μ, implying that the random effects ε_j are normally distributed. This assumption is difficult to validate and is questionable when the number of paired-clusters is small. Also, the assumption that the Y_j are approximately normally distributed around θ_i is reasonable when the Y_i are based on large numbers of subjects. Another limitation is related to the confidence intervals for overall treatment effect. These are based on an estimate of the variance of the true effect sizes, $\hat{\tau}^2$, which results in confidence intervals that are more narrow than they should be. As a result, test sizes are decreased.

3.5.1 Estimation of τ^2 – DerSimonian and Laird

Heterogeneity between clusters can be tested using the following statistic developed by Cochran

$$Q_{\hat{\omega}} = \sum \hat{\omega}_j (Y_j - \hat{\mu})^2, \quad \text{where } \hat{\omega}_j = 1/\hat{\sigma}_j^2$$

which is approximately Chisquare with k-1 degrees of freedom. The DerSimonian and Laird estimate is obtained by equating the observed value of $Q_{\hat{\omega}}$ with an estimate of its expected value. The expected value of $Q_{\hat{\omega}}$ $(q_{\hat{\omega}})$ is

$$q_{\hat{\omega}} = k - 1 + \tau^2 \left(\sum \hat{\omega}_j - \frac{\sum \hat{\omega}_j^2}{\sum \hat{\omega}_j} \right)$$

Substituting t for τ^2 and solving for t we obtain

$$t = \frac{q_{\hat{\omega}} - (k-1)}{\sum \hat{\omega}_j - \sum \hat{\omega}_j^2 / \sum \hat{\omega}_j}.$$

By definition, the value of t must be positive, thus since it is possible to obtain a negative value using the above formula, we must define the DerSimonian and Laird truncated estimator as follows

$$\hat{\tau}^2 = \begin{cases} t & \text{if } t > 0 \\ 0 & \text{if } t \leq 0 \end{cases}$$

The value of $\hat{\tau}^2$ is then used to calculate the estimates $\hat{\mu}$ and $\widehat{\text{var}}(\hat{\mu})$.

3.6 Continuity correction

Conducting meta-analyses on studies with sparse event rates can often lead to computational difficulties because of the higher probability of zero events in one or both of the treatment arms. Zero events in an arm lead to an inability to calculate the weight associated with the information from that study. A variety of continuity corrections have been proposed to remedy the problem and their pitfalls are well discussed.[26] For the purposes of this thesis, a constant continuity correction of 0.5 was applied in situations where one of the treatment arms had zero events. In other words, if there happened to be a zero event in one of the treatment arms, 0.5 was added to a_j, b_j, c_j and d_j.

3.7 Simulation

A simulation study was undertaken to estimate the size of hypothesis tests using the two fixed effects models described above, as well as a random effects model. For comparability with the paired permutation test size estimates presented in Chapter 2, the simulations were run using the same values for the cluster sizes (K_{ij}={(100,9), (100,29), (100,49), (100,99), (400,39), (400,119), (9,100), (29,100), (49,100), (99,100), (39,400), (119, 400), (9,9), (29,29), (39,39), (49,49), (100,100), (200,200), (300,300), (400,400)}), for the number of pairings (J=(5, 10, 20, 40)) as well as the Bernoulli probabilities (p={0.1, 0.5}). It was assumed that all the pairings in a study would have the same cluster-size ratio. Ten thousand meta-analyses were simulated for each set of parameter values. The R code for this simulation can be found in Appendix A.3.

The first step in the simulation program was to randomly generate the number of events for each pair of practices based on a Binomial distribution with mean K_{ij}* p and variance $p(1\text{-}p)/K_{ij}$. Next, the treatment effect for each pair was calculated as the difference in the proportions of event rates. Continuity corrections were applied to alleviate the computational difficulties that arise from zero events. Zero events lead to a null variance which in turn makes the weights based on an inverse variance impossible to calculate. For each of the three models, overall treatment effects were calculated along with an associated 95% confidence interval.

Since the data for the simulation were generated under the null hypothesis of no treatment effect, a failure to accept H_0 results in Type 1 error. Also, under the null hypothesis , one would expect the risk difference to be close to 0. Thus, test sizes at the alpha = 5% level were determined over 10,000 simulations by counting the frequency with which the 95 per cent confidence intervals did not contain 0.

Table 3.2 Estimated size times 10,000 of a one-sided test using a fixed effects model with inverse variance weights (upper), random effects model (2nd from top), Mantel-Haenszel approach (3rd from top) and the Mantal-Haenszel with continuity correction (bottom) for $W_{ij.}$ following a binomial distribution with parameters K_{ij} and p.

Cluster sizes		Number of pairings, J							
Intervention	Control	5		10		20		40	
K_{1j}	K_{2j}	p=0.1	p=0.5	p=0.1	p=0.5	p=0.1	p=0.5	p=0.1	p=0.5
100	9	56	1554	109	1625	251	1748	657	1789
		54	688	108	708	251	682	657	692
		941	706	817	674	695	598	664	633
		170	699	254	670	419	594	875	631
100	29	1208	617	1865	665	3160	669	5150	664
		773	415	1225	457	2242	501	4088	500
		558	527	544	556	556	553	552	541
		503	527	492	556	517	553	507	541
100	49	790	560	1006	584	1352	540	2062	571
		543	406	652	445	906	431	1481	454
		511	514	531	545	476	497	508	520
		507	514	529	545	472	497	505	520
100	99	536	545	521	545	540	534	525	561
		388	396	412	412	435	425	448	462
		501	509	488	530	504	506	518	531
		501	509	488	530	504	506	518	531
400	39	2206	626	3384	672	5262	642	7741	674
		1087	411	1780	469	3179	461	5548	514
		606	541	536	577	540	524	550	544
		578	541	515	577	512	524	532	544
400	119	791	527	890	542	1274	550	1858	522
		518	394	556	409	813	425	1315	425
		556	507	521	524	537	525	519	496
		556	507	521	524	537	525	519	496
9	100	45	1530	85	1658	256	1823	712	1839
		44	678	85	685	256	743	712	700
		901	677	776	647	698	657	671	619
		163	673	241	645	423	655	864	616
29	100	1225	649	1837	639	3035	659	5108	688
		830	443	1233	452	2144	460	4014	522
		581	557	568	534	544	514	539	557
		523	557	522	534	497	514	510	557
49	100	796	569	933	570	1362	591	2090	580

Cluster sizes		Number of pairings, J							
Intervention	Control	5		10		20		40	
K_{1i}	K_{2i}	p=0.1	p=0.5	p=0.1	p=0.5	p=0.1	p=0.5	p=0.1	p=0.5
		497	401	613	429	928	466	1493	455
		527	501	510	521	541	544	537	519
		522	501	508	521	535	544	531	519
99	100	604	489	492	532	528	556	519	540
		436	349	390	390	430	443	426	429
		559	449	486	508	491	531	503	511
		559	449	486	508	491	531	503	511
39	400	2250	664	3468	667	5340	635	7671	632
		1133	458	1806	461	3158	477	5419	492
		631	563	560	534	528	519	530	517
		606	563	535	534	503	519	510	517
119	400	803	566	908	542	1252	534	1926	499
		516	410	557	410	820	411	1360	421
		547	549	482	519	510	506	544	477
		547	549	482	519	510	506	544	477
9	9	133	1033	87	1053	73	1083	96	1137
		126	581	84	567	71	587	96	660
		683	719	623	609	656	636	663	682
		267	717	259	601	262	634	303	682
29	29	446	619	434	640	393	632	411	613
		365	438	367	454	337	469	357	456
		528	531	541	543	501	566	525	513
		494	531	521	543	486	566	510	513
39	39	510	581	481	602	493	618	460	619
		405	407	406	450	413	476	380	491
		528	525	531	553	527	527	473	554
		521	525	527	553	521	527	469	554
49	49	525	548	542	571	536	553	550	619
		405	392	443	412	436	428	478	496
		511	513	521	515	501	520	527	577
		510	513	520	515	499	520	526	577
100	100	563	582	540	556	585	536	529	566
		420	444	414	400	471	422	421	483
		537	571	526	531	548	508	516	547
		537	571	526	531	548	508	516	547
119	119	497	527	529	538	538	521	511	511
		366	407	428	428	438	417	425	428
		460	523	507	528	518	491	501	488
		460	523	507	528	518	491	501	488

Cluster sizes		Number of pairings, J							
Intervention	Control	5		10		20		40	
K_{1j}	K_{2j}	p=0.1	p=0.5	p=0.1	p=0.5	p=0.1	p=0.5	p=0.1	p=0.5
200	200	529	487	545	535	515	533	541	501
		391	366	425	417	426	440	455	430
		526	486	524	534	504	521	522	497
		526	486	524	534	504	521	522	497
300	300	506	549	522	520	496	497	529	508
		380	412	403	385	414	408	451	414
		486	548	495	520	485	483	538	500
		486	548	495	520	485	483	538	500
400	400	536	536	518	515	496	524	527	485
		395	413	400	394	416	416	446	416
		526	536	512	513	488	523	523	470
		526	536	512	513	488	523	523	470

When the design is balanced (i.e. $K_{1j} \approx K_{2j}$), and the number of pairs is not too small (i.e. $J > 5$), the test sizes are close to the nominal level of 500. Test sizes associated with the random effects approach were the smallest. This is because the confidence intervals for $\hat{\mu}$ are wider with a random than for a fixed effects model with inverse variance weights. Further, as the Mantel-Haenszel estimate is a conservative one, the confidence intervals are even wider than those for the fixed effects.

For the unbalanced design where K_{1j} and K_{2j} were very different, it is evident that the meta-analysis approach to analyzing cluster randomized data did not work well. The test sizes were extremely erratic, ranging from 45 to 7741 for the fixed effects model with inverse variance weights, and from 44 to 5548 for the random effects model. There does not seem to be any particular trend for the test sizes. For example, for the case where $K_{1j} = 9$ and $K_{2j} = 100$, when $J = 5$ and $p = 0.1$, the size for the fixed (inverse variance) and random effects models are 45 and 44 respectively when a correction of 0.5 is applied. However, if we increase K_{1j} to 29, the sizes become astronomical! (1225 inverse variance fixed, 830 random). The Mantel-Haenszel method yielded test sizes that are much more appropriate except for the case where

K_{1j} =9 and K_{2j} = 100 (and vice versa) and p = 0.1. Possible explanations for the erratic behaviour of the test sizes were investigated and the findings for the fixed effects model with J = 5, K_{1j} = 9, K_{2j} = 100 and p = 0.1 are now presented.

The main question we sought to answer was "how was it that only 45 out of 10,000 simulated confidence intervals for global treatment effect did not contain 0". The first step was to take a closer look at the simulated sequences of event rates when J = 5 and K_{1j} = 9. Each simulated sequences consists of 5 elements, one for each pair in the meta-analysis. Each of these elements is a randomly generated number of events, ranging from 0 to K_{1j} = 9, based on an event probability p = 0.1. For example, possible sequences are (1,1,0,0,0), (0,2,1,0,1) and so on. The most common numbers of events will be 0, 1, and 2 since, based on the binomial approximation, the expected mean K_{ij}*p = 9*0.1 = 0.9. A total of 10,000 sequences were simulated in order to obtain a frequency distribution for the sequences. Note that the order in which the events occur is not important, but rather it is the number of 0 events, 1 events, 2 events and so on. The twenty sequences that occurred with the highest frequency are presented in Table 3.3.

The next step was to calculate the confidence intervals based on these twenty sequences. For the control clusters, it was assumed that the number of events perfectly reflected the theoretical event p = 0.1, as opposed to a random generating of events with a variation component. Hence, since K_{2j} = 100, all 10,000 sequences were (10, 10, 10, 10, 10). The treatment effect and associated confidence intervals were calculated for the cases both with and without the continuity correction factor of 0.5. Table 3.3 presents the results.

Table 3.3 Sequences that occurred with highest frequency out of 10,000 random generations when $K_{1j} = 9$, $J = 5$ and $p = 0.1$. Treatment effects and confidence intervals are based on $K_{2j} = 100$. Sequences for K_{2j} were set to (10, 10, 10, 10, 10).

K_{1j}		Uncorrected		Corrected	
Sequence	Freq.	Treatment effect	95% CI	Treatment effect	95% CI
21100	1087	-0.09	(-0.13, -0.05)	-0.02	(-0.10, 0.06)
11100	862	-0.09	(-0.13, -0.05)	-0.03	(-0.11, 0.05)
21000	860	-0.09	(-0.13, -0.06)	-0.03	(-0.11, 0.04)
11000	851	-0.09	(-0.13, -0.06)	-0.04	(-0.11, 0.04)
21110	777	-0.07	(-0.12, -0.02)	0.00	(-0.09, 0.09)
22100	531	-0.09	(-0.13, -0.05)	-0.01	(-0.10, 0.07)
22110	520	-0.07	(-0.12, -0.02)	0.01	(-0.08, 0.10)
10000	441	-0.10	(-0.13, -0.07)	-0.05	(-0.12, 0.02)
11110	441	-0.07	(-0.13, -0.02)	-0.01	(-0.1, 0.08)
31100	312	-0.09	(-0.13, -0.05)	-0.01	(-0.1, 0.07)
32110	280	-0.07	(-0.12, -0.02)	0.02	(-0.08, 0.11)
32100	277	-0.09	(-0.13, -0.05)	-0.01	(-0.09, 0.08)
31000	206	-0.09	(-0.13, -0.06)	-0.03	(-0.11, 0.05)
31110	201	-0.07	(-0.12, -0.02)	0.01	(-0.09, 0.10)
21111	190	0.03	(-0.07, 0.13)	0.03	(-0.07, 0.13)
20000	187	-0.10	(-0.13, -0.07)	-0.04	(-0.11, 0.03)
22111	185	0.04	(-0.06, 0.15)	0.04	(-0.06, 0.15)
22000	170	-0.09	(-0.13, -0.06)	-0.03	(-0.10, 0.05)
22210	135	-0.07	(-0.12, -0.01)	0.02	(-0.07, 0.12)
32210	124	-0.07	(-0.12, -0.01)	0.03	(-0.07, 0.13)

For the corrected case, almost all the confidence intervals include 0, which explains the ridiculously low size estimate of 45 for $K_{1j} = 9$, $K_{2j} = 100$, $J = 5$ and $p = 0.1$ in Table 3.2. On the other hand, without the correction factor, a large proportion of all the CI do not include 0. Thus, an estimate of size without the continuity correction would be unreasonably high (not shown).

The process was also repeated for the case where $K_{1j} = 29$. In this case, for both the corrected and the uncorrected cases, a large proportion of confidence intervals did not

include 0, which explains why our size estimates were so high in table 3.2. It does not however provide any direction on how these erratic behaviours can be controlled.

3.8 Summary

In this chapter we presented an overview of meta-analysis and discussed how meta-analytical techniques may be applied to the analysis of paired-cluster randomized trials with a binary outcome where treatment effect is measured as a difference in the proportion of outcomes in both treatment groups. Simulation results of the application of these methods have shown that though meta-analysis techniques were appropriate in some cases for continuous outcomes, as demonstrated by Thompson, they are not appropriate for use with binary outcomes, especially when there is a chance that there will be systematic imbalance in the sizes of paired clusters.

In the next chapter, we will apply both a randomization test and a test based on meta-analysis techniques to the CHAT data.

Chapter 4

Application to CHAT

The objective of this chapter is to assess the validity of applying a permutation test and/or a test based on meta-analytic techniques to the data from the Community Hypertension Assessment Trial (CHAT), a multi-centre matched cluster randomized trial. Simulations are run for both methodologies based on the CHAT study parameters to assess their validity as measured by test sizes. All results are presented in Table 4.1 at the end of this chapter.

4.1 The CHAT program

4.1.1 Overview

The CHAT program is a community-based program designed to promote health and prevent cardiovascular disease among older adults in the community. More specifically, the Program is designed to improve the cardiovascular health of older adult patients by identifying previously unidentified patients with high blood pressure and by improving blood pressure monitoring and management for patients diagnosed with high blood pressure. The Program is designed as a loop. First, physicians suggest that their patients attend blood pressure sessions. Subsequently, patients attend the

sessions and the information obtained at these sessions is sent back to the physicians, who, in turn, follow-up with their patients.

4.1.2 Procedures

Family physicians (FP) participating in the Program are paired and randomized to either intervention or control. All eligible patients are identified and a random sample of 55 charts in each practice are abstracted by registered nurses or experienced health record abstractors. Chart abstractors are asked to record numerous blood pressures and answer questions related to the presence or absence of a diagnosis of diabetes, heart disease, stroke/transient ischemic attack/cerebrovascular accident, and target organ damage (nephropathy or retinopathy). Family physicians in the intervention group invite, using personalized invitations, their eligible patients to attend community pharmacy blood pressure measurement sessions operated by volunteer peer health educators (trained by public health nurses). During the sessions, volunteer peer health educators measure and record the patients' blood pressure using accurate automated devices; they also help participants complete a cardiovascular risk profile and suggest participants see pharmacists or their family physicians, when appropriate. With the participants' permission, volunteer coordinators forward all blood pressure readings and risk profiles to a computerized database, which subsequently sends the information to family physicians by fax (or mail) in an effort to augment in-office records and encourage physician follow-up of their patients. A copy of the blood pressure readings and cardiovascular risk factor information is given to each participant and to their regular pharmacist. One year later, the chart abstractors return to the physicians' practices to audit the same charts. The intent is to evaluate if the physicians responded to the intervention as measured by a change in the proportion of their patients with a BP reading in the last 12 months.

4.1.3 Physicians

The selection of FP practices is done by drawing a random sample of 100 family physicians in Hamilton and Ottawa from the Southam Medical Database in Hamilton and a local database with the Ottawa Health Research Institute. Eligible physicians are those who have a non-academic, full time, regular family practice in terms of size and case mix, and who are able to provide a roster of patients aged 65 and over. Physicians are excluded if they worked in walk-in clinics, emergency departments, or if they are about to retire or work part-time. Physicians are also excluded if they have fewer than 50 patients who are over 65 or if they have a specialized practice profile. Starting from the top of the randomized list, physician recruiters contact the family physicians to participate in the Program until the sample size of 14 physicians in each city has been met. Due to time constraints, pairing of the practices is done prior to establishing the complete list of the 14 participating physicians. For example, once the first 4 physicians are identified, they are paired according to size of practice (pair1: 2 largest, pair2: 2 smallest). Subsequently, within each pair, the practices are randomized to either treatment or control.

4.1.4 Patients

Participants in the study are community-dwelling patients aged 65 years and older, who have visited their family physician at least once in the previous 12 months, who have medical records in the practice, and who are mobile to attend blood pressure clinics. Participants are screened based on eligibility criteria detailed in the health record extraction manual and patients are excluded if they are deceased, terminally ill, have transferred to another practice, are not mobile, are scheduled to have surgery within the next 60 days, or have significant cognitive impairment. Patients are also excluded if they do not speak English or French and do not have a family member who can interpret, or if they have a personal or family situation that prevents them from participating.

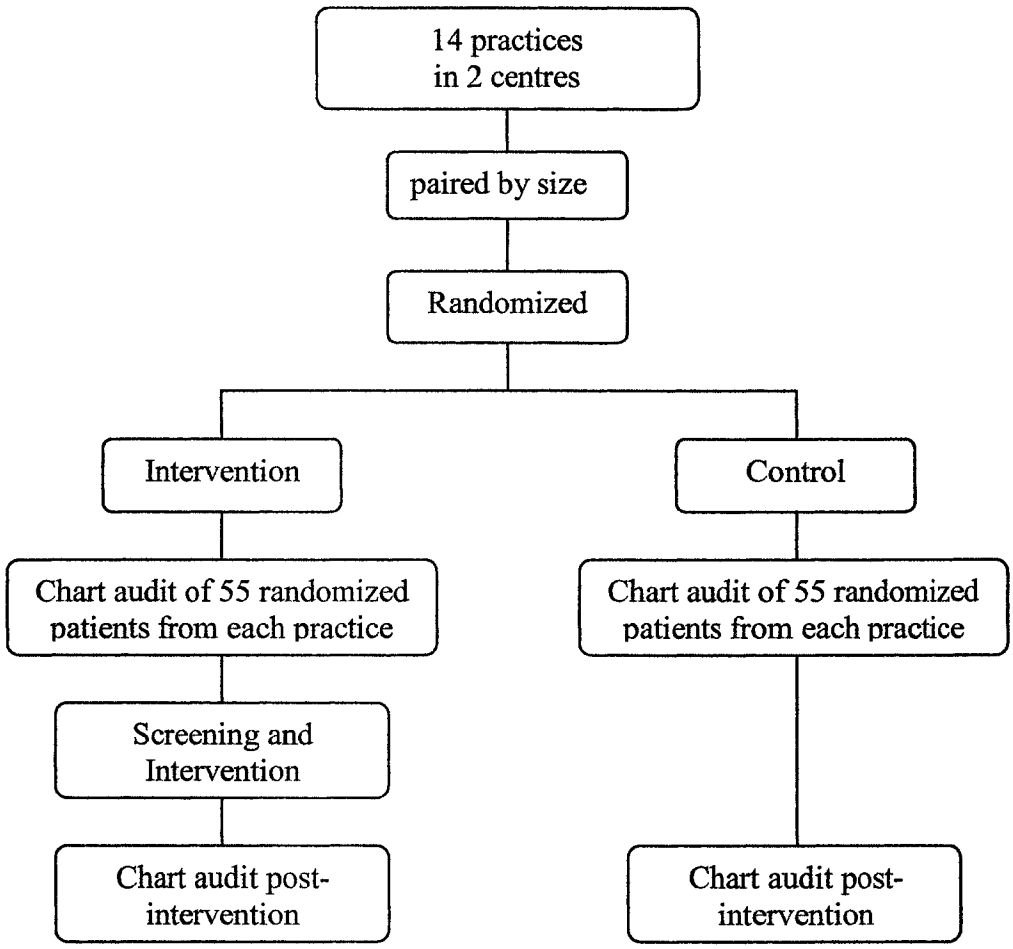


Figure 4.1 The design of the CHAT study

4.1.5 Primary Outcome

The primary objective of the CHAT program is to evaluate if the physicians responded to the intervention, as measured by a change in the proportion of patients with a BP reading in the last 12 months at the practice level from pre- to post-intervention. Since at the time of the writing of this thesis only the baseline data was available, the tested outcome was a simple proportion of participants with a BP

reading in the last 12 months. In addition, due to the desire for the analysts to be blinded to the treatment allocation until the final analysis stage, the 28 practices were paired randomly and then randomized to either treatment or control.

4.2 Permutation tests

The randomization procedure used for CHAT is equivalent to that of a pair-matched design, whereby family practice $(1, j)$ is paired with practice $(2, j)$ based on size of practice and timing of study entry, where (i, j) refers to the combination of treatment (i) and pair (j). Treatment effects are measured as the difference between the pairs : $U_j = W_{1j.} - W_{2j.}$

Let $Y_{ijk}(t)$ denote the response of cohort member k in family practice pair j on treatment i at time t for $i = 1, 2$; $j = 1, 2, \ldots, J$; $k = 1, 2, \ldots, K_{ij}$; and $t = t_0, t_T$. Here $i = 1$ corresponds to intervention and $i = 2$ to control, t_0 corresponds to baseline and t_T to post-intervention. Let W_{ijk} be a scalar function of $Y_{ijk}(t)$ that summarizes the health effects on cohort member (i, j, k) for t. In our case, $W_{ijk} = Y_{ijk}(t_0)$, where $Y_{ijk}(t_0)$ is a binary indicator of whether cohort member (i, j, k) had a BP reading at baseline, t_0.

$$W_{ijk} = \begin{cases} 0 & \text{no BP readings in the past 12 months} \\ 1 & \text{at least one BP reading in the past 12 months} \end{cases}$$

Given K_{ij} for each family practice,

$$W_{ij.} = \frac{1}{K_{ij}} \sum_k W_{ijk}$$

represents the proportion of participants in practice pair j on treatment i with a BP reading in the last 12 months. Treatment effect in each pair is then calculated as $U_j = W_{1j.} - W_{2j.}$ and the global measure of treatment effect U is the average of all the U_j across the J pairs. A permutation test on the CHAT data leads to rejection of the hypothesis that there is no treatment effect on average.

4.3 Meta-analytical techniques

Using the meta-analytical approach, each pair of family practices is treated as a separate study. Thus, the first step is to calculate the effect sizes Y_i for each of the 14 pairs. Let p_{ij} represent the proportion of patients in treatment group i and in pair j with a BP reading in the past 12 months. Then our measure of treatment effect Y_j can be defined as the difference between the treatment and control groups in the proportions of patients with readings,

$$Y_j = p_{1j} - p_{2j}$$

The global measure of treatment effect U is then calculated as the weighted average of the Y_j. As was the case with the randomization test, application of the meta-analysis techniques to the data results in the rejection of the hypothesis that there is no treatment effect. The R code to apply permutations tests and tests based on fixed and random model meta-analysis is included in Appendix A.6.

4.4 Simulation

Under the null hypothesis of no treatment effect on average, size associated with both a permutation test and a test based on meta-analytical techniques was estimated by simulation for the CHAT study parameters. These programs are found in Appendix A.4 and A.5 respectively. The study parameters, presented in Table 4.1, are number of clusters (28), number of pairs (14) and cluster size for both treatment and control groups (55). In order to randomly generate event rates (blood pressure readings in the past 12 months) for each family physician practice, knowledge of the distribution of $Y_{ijk}(t)$ was required. This proportion was calculated from the CHAT study data as the mean of all the practice level proportions (0.9).

Table 4.1 CHAT parameters and simulation results

Design Parameters	
Number of clusters	28
Pairs	14
Participants per cluster	55
Probability of success	0.90
Simulation results	
Test	Size
Randomization	
permutation test	574
t-test approximation	514
Meta-analytic approach	
Fixed effects	
inverse variance	549
Mantel Haenszel	548
Random effects	453
Nominal size	500

The simulation results for the randomization tests are consistent with those in Table 2.4 of Chapter 2. Based on Table 2.4 one would expect the size of a permutation and t-test to be close to nominal levels when event rates are high, there are more than 10 pairs and the cluster sizes are over 50. The sizes for the fixed and random effects meta-analytic tests are also very much in line with what we would expect based on Table 3.2 in Chapter 3.

4.5 Summary

We have shown that in the context of the CHAT study, tests based on the randomization model and meta-analytical techniques both have sizes that approach nominal levels. In the case of the randomization test, the size is close to the nominal

levels by virtue of the balanced design and the reasonable number of clusters. As for the test based on the meta-analytical techniques, the nominal levels were attained because the study was balanced, the number of pairs was suitably large, and the cluster sizes were also sufficiently large. In fact, the study characteristics represent the ideal combination of factors to ensure Type I errors that are close to the specified levels.

In the next chapter, we will discuss the issue of allocation imbalance resulting from non-centralized block randomization.

Chapter 5

Allocation issues with PDAs in multi-centre block-randomized trials

The objective of this chapter is to clarify the issues related to allocation imbalance resulting from non-centralized blocked randomization. We focus on the use of multiple personal digital assistants (PDA) per centre each with pre-loaded blocked randomization sequences. A comparison is made of the probability of imbalance at the centre and study levels for blocked and non-blocked randomization schemes and investigate. The effect of these imbalances on power, not only at the final analysis stage but also at the time of interim analysis, is discussed. Recommendations on the practicality of these methods are made.

5.1 Introduction

In multi-centre trials where randomization in blocks is employed to allocate participants to treatment arms, each centre typically follows its own randomly generated blocked sequence. Each block contains a pre-specified number and proportion of treatment assignments. This ensures a specific distribution of patients to the treatment and control arms for each of the centres, and for the study overall.

Before a participant can be randomized to the study, the recruiter must first obtain the blinded allocation assignment from the centre's centralized randomization system. However, due to the dynamic nature of the recruitment process, one in which speed of allocation is of the essence, investigators are seeking to devise a system where each recruiter can immediately randomize a new participant without having to access a centralized randomization server. To achieve this, there have been proposals for blocked randomization at the level of the recruiter through use of a personal digital assistant (PDA) rather than at the level of the centre.

PDAs have been used in centralized recruitment as they are portable, inexpensive and allocation concealment is almost assured. The use of PDAs is likely to increase as evidenced by Martin's findings that PDA use among physicians has increased from 19% in 2001, to 28% in 2002 and to 33% in 2003.[27,28,29] However, connecting a PDA to a centralized randomization server is problematic because 1) the server may go down, 2) synchronizing with a desktop PC is time consuming and not always feasible at the time when a treatment allocation is required, 3) synchronizing over a wireless network may be preferable but wireless networks are not always available in hospital settings, wireless-enabled PDAs are more expensive and they do not alleviate the problem of servers going down. Thus, preloading allocation sequences into PDAs seems an attractive alternative.

5.2 Allocation Proportions

It is generally recommended that randomization be blocked and stratified by centre if more than one centre is involved.[30,31,32,33] This is to ensure that balance will be obtained, not only in the number of treatment assignments in each group, but also, for known and unknown confounders associated with each centre, reducing bias and increasing power to detect a treatment effect. However, blocking is not recommended when there is a high chance of incomplete blocks within one or more strata as the analysis is complicated by the existence of an intra-block correlation.[34] In addition,

large allocation imbalances can significantly reduce the power of a study to detect treatment effects.[35] The effect of treatment allocation imbalance on power is discussed in more detail in the next section. Loss of power due to imbalance is an important factor to consider in trials where interim analyses are planned. The importance of performing interim analyses in trials where the effects of intervention or lack thereof can result in harm to participants has been well documented.[36,37] Hence, it is vital to ensure that allocation at interim will be such that there is adequate power to detect treatment effects.

In the case of non-central blocked randomization schemes, varying degrees of imbalance are expected depending on the combination of factors such as the number of participants, the number of PDAs, and the block sizes. Figures 6.1a and 6.1b illustrate the imbalances that may occur both in the centre's treatment allocation, as well as for the study as a whole when each PDA works from its own individual blocked randomization schemes. Suppose there are 2 treatments, A and B, and blocks of size 4 are used. In addition, competitive recruitment is practiced within each centre, i.e. each PDA will be used to randomize as many participants as possible until the total number of participants for that centre has been satisfied. The imbalance arises from the fact that some blocks may be incomplete. For each PDA, the maximum imbalance is one half of the block size,[34,38] therefore the maximum imbalance for each centre is one half the block size times the number of PDAs used by the centre. Therefore, the imbalance at the study level is capped at one half the block size times the total number of PDAs across all centres. In Figure 6.1a, the allocation imbalances for the centres results in an extreme imbalance for the study. In contrast, Figure 6.1b illustrates how the imbalances at the centre level can cancel out such that there is no imbalance in the total number of treatment allocations.

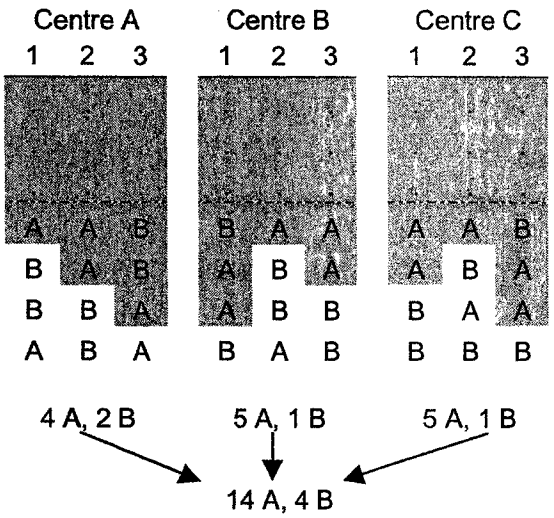


Figure 5.1a Imbalance at both centre and study level, 3 PDA per centre

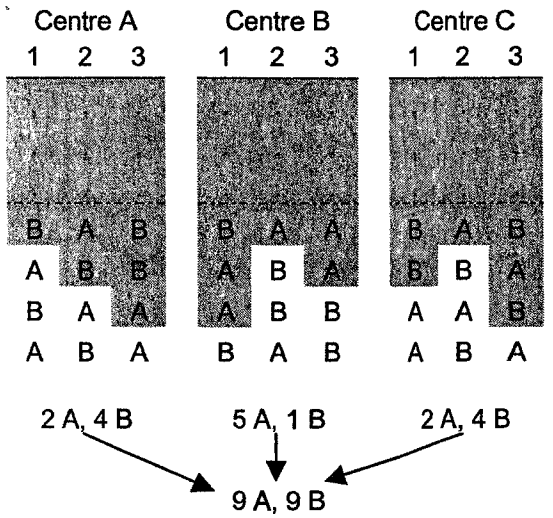


Figure 5.1b Imbalance at centre level only, 3 PDA per centre

5.3 Effects of Imbalance on Power

Lachin has shown that although power is usually maximized with equal treatment allocation, the effects of imbalance on power are trivial unless the imbalances are substantial, on the order of 0.6 or 0.7 to one of the two groups.[35] Both Piantadosi[38] and Lachin[39] describe how the probabilities of imbalance greater than a specified limit, r, can be estimated for complete randomization using the large sample approximation to the binomial. After n assignments, the number of assignments to treatment A, n_a, follows a binomial distribution with $p = 1/2$ and sample size n. Under the condition $np > 5$, n_a is approximately normally distributed with mean $np = n/2$ and variance $np(1\text{-}p) = n/4$. Thus the probability of imbalance greater than r is calculated as follows:

$$\mathrm{P}\left(n(1-r) > n_a > nr\right) = 2\mathrm{P}\left(n_a > nr\right)$$

$$= 2\mathrm{P}\left(\frac{n_a - np}{\sqrt{np(1-p)}} > \frac{nr - np}{\sqrt{np(1-p)}}\right)$$

$$\cong 2\left(1-\Phi\left(\frac{nr-np}{\sqrt{np(1-p)}}\right)\right)$$

$$= 2\Phi\left(-\frac{nr-np}{\sqrt{np(1-p)}}\right)$$

$$= 2\Phi\left(-2(r-0.5)\sqrt{n}\right)$$

Figure 5.2 presents the probabilities of obtaining levels of imbalance greater than 0.55, 0.60 and 0.70 as a function of sample size.

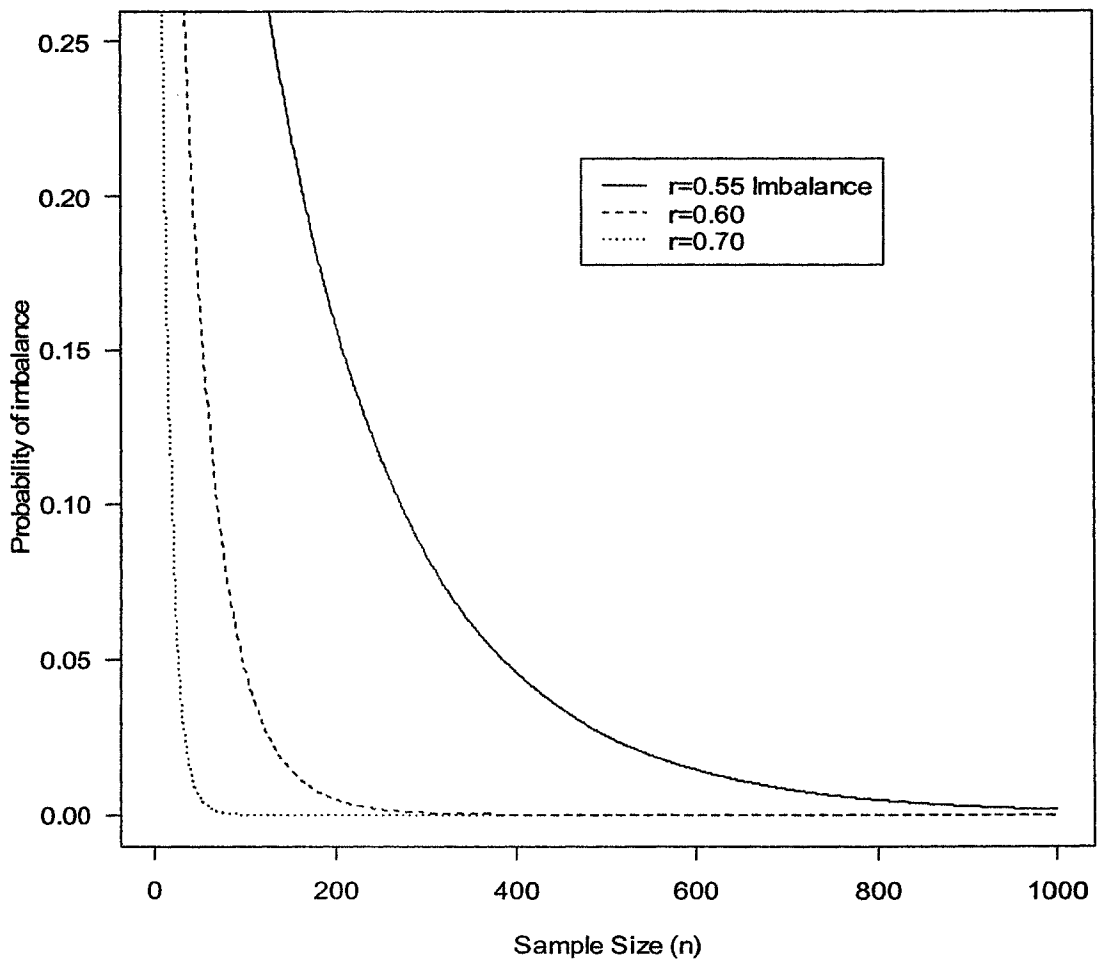


Figure. 5.2 Probability of treatment imbalances for complete randomization as a function of sample size for imbalances $\max(n_a,n_b)/n \geq r$ and $r = 0.55$, 0.6, and 0.7

This figure shows that the probability of an imbalance r decreases as the sample size increases. Also, the probability of imbalance greater than 0.70 is less than 0.05 for sample sizes greater than 30. It is equally unlikely for $r = 0.60$ with sample size greater than 100, and for $r = 0.55$ with sample size greater than 400.

Imbalance resulting from complete randomization is always greater than imbalance one can obtain using a blocking strategy. As the sample size increases, the proportions of imbalance from complete randomization decrease and approach those associated with blocked randomization. The worst case scenario for imbalance using a blocking design is that there are as many blocking sequences as there are participants, and that each sequence randomizes a maximum of one participant. This however is equivalent to performing a simple random sampling where each participant has a pre-specified probability of being assigned to either treatment A or B.

5.4 Simulating probability of imbalance

Having established that extreme imbalances could occur, the probabilities of imbalance were assessed by simulation for a range of values of the number of centres, the number of PDAs and the number of participants. The simulation was based on the assumption that there are two treatment arms, intervention and control; there should be an equal distribution of participants in each arm; each recruiter practices competitive recruitment, i.e. each recruiter will randomize as many patients as possible until the total number of participants for that centre has been satisfied; each centre practices competitive recruitment, i.e. each centre will randomize as many participants as possible until the total number of participants for the study has been satisfied; each centre has the same number of PDAs. Imbalance was assessed for two different blocking strategies: a fixed block size of 4 and a random block size of 4 or 6.

The simulation can be broken into four major steps. The first step involves randomly assigning a number of participants to each centre. This is achieved by randomly assigning N participants to n centres with specified probability. The second step

involves randomly creating block patterns for each PDA based on pre-specified block size and blocking strategy. The length of each of these patterns will be such that it would be possible for one recruiter to randomize all the patients for his centre. The third step is to randomly select the number of participants recruited by each recruiter. This is achieved by randomly assigning N participants to n PDAs with equal probability. Finally, the fourth step measures the proportion of allocation imbalance for three levels of seriousness (55%, 60%, 70%) resulting from the blocked randomization as well as complete randomization (SRS).

Table 5.1 presents the percentages of imbalance for blocked randomization (block size of 4) for various combinations of the number of centres, the number of PDAs and the number of participants per centre, based on a specified level of seriousness. For this scenario, each centre recruits the same number of participants. Ideally, allocation percentages would be near 50%. Larger imbalances occur in situations in which there are small numbers of participants. More specifically, an increase in imbalance is related to one of more of the following: 1) a decrease in the number of study centres, 2) an increase in the number of PDAs, and 3) a decrease in the number of study participants. These are very similar to the results obtained with a random block size randomization (size= 4 or 6, not shown). The results for the 1 centre case indicate the level of allocation imbalance that can occur at the centre level. As expected, this is much greater than the imbalance one would expect at the study level where multiple centres are involved. Issues related to centre-level imbalance are described later. As figure 5.1b intimated, by increasing the number of centres, the within-centre imbalances tend to cancel out. The results from table 5.1 also demonstrate the pitfall of increasing the number of PDAs. Increasing the number of PDAs unnecessarily will result in a substantial increase in imbalance.

Table 5.1 Percentages of treatment imbalances for blocked randomization (block size = 4) as a function of the number of centres, the number of PDAs per centre and the number of participants per centre, for imbalances $\max(n_a,n_b)/n \geq r$ and r = 0.55, 0.6, and 0.7. (n=total sample size)

Number of Centres	PDAs per Centre	Participants per Centre: 4				8				12				16				20				24			
		0.55	0.60	0.70	n	0.55	0.60	0.70	n	0.55	0.60	0.70	n	0.55	0.60	0.70	n	0.55	0.60	0.70	n	0.55	0.60	0.70	n
1	1	0	0	0	(4)	0	0	0	(8)	0	0	0	(12)	0	0	0	(16)	0	0	0	(20)	0	0	0	(24)
1	2	43	44	45	(4)	37	35	1	(8)	38	1	0	(12)	40	1	0	(16)	1	0	0	(20)	2	0	0	(24)
1	3	50	52	55	(4)	52	53	4	(8)	52	5	0	(12)	50	5	0	(16)	4	0	0	(20)	5	0	0	(24)
1	4	55	55	55	(4)	60	58	10	(8)	58	9	0	(12)	54	8	0	(16)	9	0	0	(20)	9	1	0	(24)
5	1	0	0	0	(20)	0	0	0	(40)	0	0	0	(60)	0	0	0	(80)	0	0	0	(100)	0	0	0	(120)
5	2	34	11	0	(20)	7	0	0	(40)	2	0	0	(60)	0	0	0	(80)	0	0	0	(100)	0	0	0	(120)
5	3	43	16	1	(20)	17	1	0	(40)	3	0	0	(60)	1	0	0	(80)	0	0	0	(100)	0	0	0	(120)
5	4	46	20	2	(20)	25	3	0	(40)	8	0	0	(60)	3	0	0	(80)	1	0	0	(100)	0	0	0	(120)
15	1	0	0	0	(60)	0	0	0	(120)	0	0	0	(180)	0	0	0	(240)	0	0	0	(300)	0	0	0	(360)
15	2	21	2	0	(60)	1	0	0	(120)	0	0	0	(180)	0	0	0	(240)	0	0	0	(300)	0	0	0	(360)
15	3	27	4	0	(60)	5	0	0	(120)	0	0	0	(180)	0	0	0	(240)	0	0	0	(300)	0	0	0	(360)
15	4	31	5	0	(60)	7	0	0	(120)	1	0	0	(180)	0	0	0	(240)	0	0	0	(300)	0	0	0	(360)
25	1	0	0	0	(100)	0	0	0	(200)	0	0	0	(300)	0	0	0	(400)	0	0	0	(500)	0	0	0	(600)
25	2	12	0	0	(100)	0	0	0	(200)	0	0	0	(300)	0	0	0	(400)	0	0	0	(500)	0	0	0	(600)
25	3	16	2	0	(100)	1	0	0	(200)	0	0	0	(300)	0	0	0	(400)	0	0	0	(500)	0	0	0	(600)
25	4	21	2	0	(100)	4	0	0	(200)	0	0	0	(300)	0	0	0	(400)	0	0	0	(500)	0	0	0	(600)

Based on Lachin's guidelines regarding acceptable levels of imbalance so as not to compromise the power of the statistical analyses, we would want to avoid planning our study such that the final or interim analyses would land in the shaded areas of Table 5.1. For these cases, the probability of an imbalance greater than or equal to 0.60 is non-negligible. Thus, depending of the moral and ethical issues involved in the trial, interim sample sizes that fall into these categories should be avoided.

Figures 5.3a and 5.3b are based on the data from Table 5.1 and were produced to help visualize the effect of the number of centres and the number of PDAs on the probability of obtaining a treatment imbalance $\max(n_a,n_b)/n \geq 0.55$ with the PDA level blocking. The proportions of imbalance decrease as the number of centres increase, and they increases as the number of PDAs increases, represented by each line. Comparing figure 5.2a with figure 5.2b shows how increasing the number of participants to be recruited by each centre also results in a decrease in the allocation imbalance. For example, if a study expects to only be able to recruit 8 participants per centre because of the nature of the inclusion criteria and plan to make available 4 PDAs, they will need to involve more than 25 centres to "ensure" imbalance does not exceed 5%. However, if the number of participants per centre can be increased, the number of centres needed drops dramatically. Figure 5.2b shows that with 16 participants per centre, only 5 centres would be needed to decrease the proportion of imbalance to below 5%.

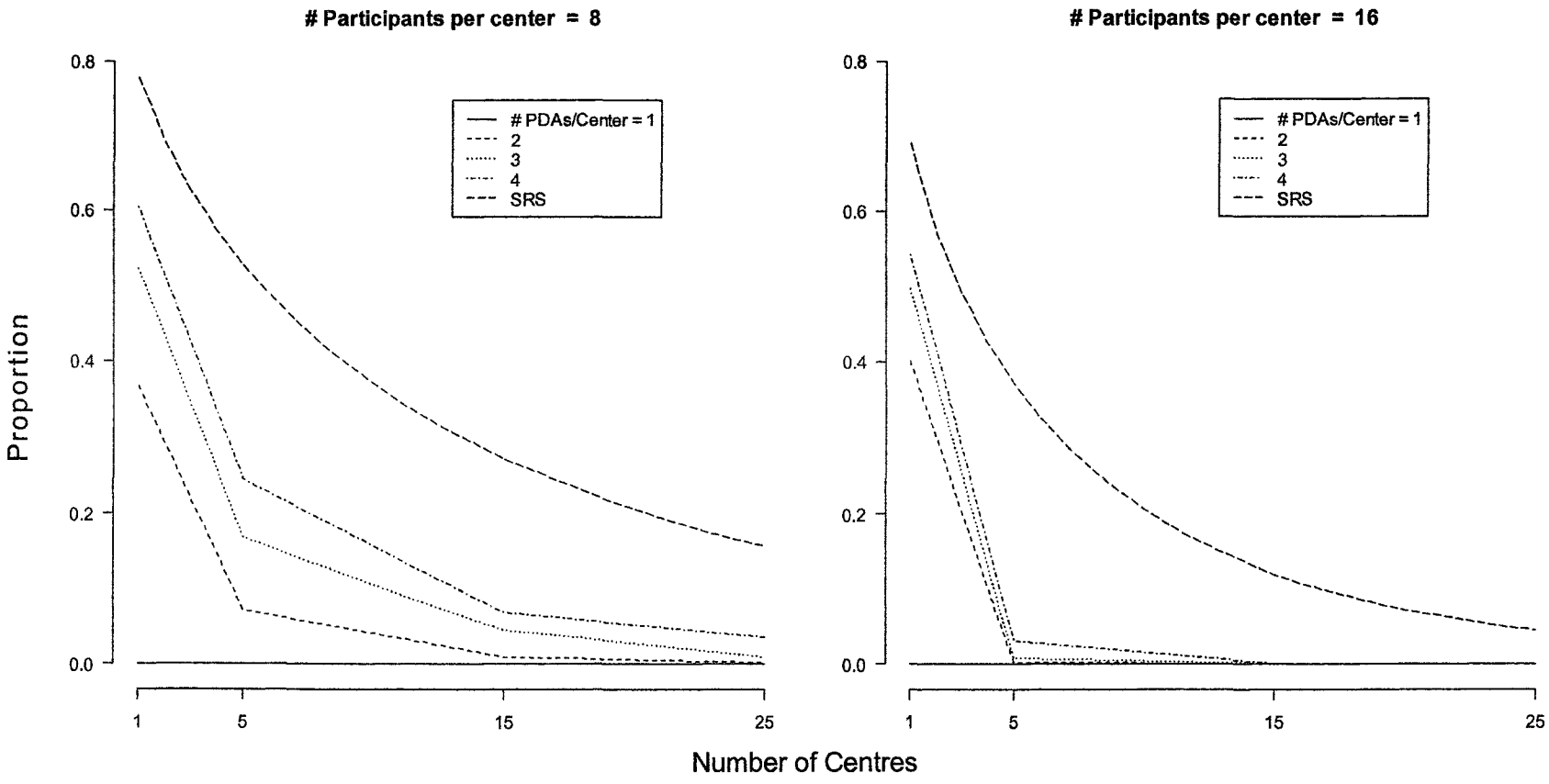


Figure 5.3a

Figure 5.3b

Figures 5.3a & 5.3b Probability of treatment imbalances for blocked randomization and for complete randomization (SRS) at the PDA level as a function of the number of centres, the number of participants and the number of PDA for imbalances $\max(n_a,n_b)/n \geq 0.55$. Based on 1000 repetitions.

5.4.1 Astronomer study data

To study the probability of imbalance in an actual setting at the time of an interim analysis, recruitment numbers at the first yearly interim were obtained for the Astronomer study. The Astronomer study is a multi-centre trial involving 19 hospitals. Each centre practiced competitive recruitment and used one PDA to randomize participants to one of two treatments arms. The total expected sample size is 442. At interim, a total of 60 participants had been enrolled. For each of the 19 centres, the number of allocations to each of the 2 treatment arms are provided in table 5.2. Though only one PDA was used per centre, we are interested in the distribution of recruitment across centres so as to simulate possible imbalances had there been more than one PDA per centre.

Table 5.2 Treatment allocations for the Astronomer study data

	Total		Group A		Group B	
Centre	n	%	n_a	%	n_b	%
1	15	25	8	53	7	47
2	3	5	2	67	1	33
3	3	5	1	33	2	67
4	5	8	2	40	3	60
5	5	8	3	60	2	40
6	2	3	0	0	2	100
7	2	3	1	50	1	50
12	3	5	2	67	1	33
14	12	20	6	50	6	50
15	1	2	0	0	1	100
16	2	3	1	50	1	50
17	2	3	0	0	2	100
19	5	8	2	40	3	60
Total	60		28		32	

The probabilities of imbalance were assessed by simulation based on the Astronomer recruitment distribution among the centres for a range of values of the number of centres, the number of PDAs and the total number of participants (see Appendix A.7).

Thus, a distribution was favored whereby 10% of the centres recruited 50% of the sample, 40% of the centres did not recruit any participants, and the remaining 50% of the centres had the same probability of recruiting. The simulation was based on much the same assumptions as for the first simulation, namely there are two treatment arms, intervention and control; there should be an equal distribution of participants in each arm; each recruiter practices competitive recruitment, i.e. each recruiter will randomize as many patients as possible until the total number of participants for that centre has been satisfied; each centre has the same number of PDAs. Tables 5.3 and 5.4 present the simulation results based on the Astronomer recruitment figures using a fixed block size of 4 and a random block size of 4 and 6 respectively. Percentages of imbalance were calculated for various combinations of the number of centres, the number of PDAs and the total number of participants per centre, based on a specified level of seriousness.

Both Tables 5.3 and 5.4 show that for a fixed sample size, the allocation imbalance increases as the number of centres increases and as the number of PDAs increases. For studies where 15 or more centres are participating and there is a sample size of 30 or less at interim, statistical analyses could be compromised by lack of power. Also note that compared to simple random sampling, blocked randomization yields a much lower probability of imbalance. However, it is unlikely a study would have such a design, even when dealing with rare events. A design using random block sizes yields slightly higher proportions of imbalance than a fixed block design.

Table 5.3 Percentages of treatment imbalance for blocked randomization and simple random sampling (SRS). Used fixed block sizes (4) and competitive recruitment between centres. Proportions are a function of the number of centres, the number of recruiters per centre and the number of participants, for imbalances $\max(n_a,n_b)/n \geq r$ and r = 0.55, 0.6, and 0.70.

Number of Centres	PDAs per Centre	Total Sample Size														
		30			60			90			120			150		
		0.55	0.60	0.70	0.55	0.60	0.70	0.55	0.60	0.70	0.55	0.60	0.70	0.55	0.60	0.70
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	4	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	2	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	3	27	1	0	1	0	0	0	0	0	0	0	0	0	0	0
5	4	32	3	0	3	0	0	0	0	0	0	0	0	0	0	0
15	1	27	1	0	1	0	0	0	0	0	0	0	0	0	0	0
15	2	40	4	0	10	0	0	2	0	0	0	0	0	0	0	0
15	3	47	8	0	14	1	0	7	0	0	1	0	0	0	0	0
15	4	48	12	0	17	1	0	9	0	0	2	0	0	1	0	0
25	1	38	6	0	6	0	0	2	0	0	0	0	0	0	0	0
25	2	47	11	0	16	1	0	10	0	0	2	0	0	0	0	0
25	3	52	12	0	19	1	0	13	0	0	5	0	0	2	0	0
25	4	49	13	1	21	2	0	17	1	0	6	0	0	4	0	0
SRS		58	27	3	44	12	0	34	6	0	27	3	0	22	1	0

Table 5.4 Percentages of treatment imbalance for blocked randomization and simple random sampling (SRS). Used random block sizes (4,6) and competitive recruitment between centres. Proportions are a function of the number of centres, the number of recruiters per centre and the number of participants, for imbalances $\max(n_a,n_b)/n \geq r$ and $r = 0.55$, 0.6, and 0.70.

Number of Centres	PDAs per Centre	Total Sample Size														
		30			60			90			120			150		
		0.55	0.60	0.70	0.55	0.60	0.70	0.55	0.60	0.70	0.55	0.60	0.70	0.55	0.60	0.70
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	3	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	4	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	2	23	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	3	29	2	0	3	0	0	0	0	0	0	0	0	0	0	0
5	4	34	2	0	5	0	0	2	0	0	0	0	0	0	0	0
15	1	35	2	0	3	0	0	0	0	0	0	0	0	0	0	0
15	2	44	9	0	11	1	0	6	0	0	1	0	0	0	0	0
15	3	49	9	0	16	1	0	10	0	0	2	0	0	1	0	0
15	4	50	13	0	22	1	0	12	0	0	4	0	0	2	0	0
25	1	44	5	0	12	0	0	4	0	0	0	0	0	0	0	0
25	2	49	11	0	19	1	0	14	0	0	2	0	0	2	0	0
25	3	50	14	0	22	2	0	17	0	0	6	0	0	3	0	0
25	4	52	16	0	28	4	0	19	1	0	8	0	0	7	0	0
SRS		58	27	3	44	12	0	34	6	0	27	3	0	22	1	0

An additional simulation was also conducted, as a subset to the one described above (see Appendix A.8). The purpose of this simulation was to assess the possible imbalances that could have occurred for the Astronomer study had its participating centres used more than one PDA for the recruitment. Thus, simulation was undertaken for different numbers of PDAs with the total sample size fixed at 60 and using the actual distribution of participants in each centre. The percentages of imbalance presented in table 5.5 are based on the actual recruitment pattern of the Astronomer study. Imbalance that would seriously compromise the analysis arises with low probability. Even with as many as 4 PDAs per centre, the probabilities of an imbalance greater than or equal to 0.60 remain below 5%. A random block size results in slightly higher imbalance.

Table 5.5 Simulated percentages of treatment imbalance based on Astronomer recruitment patterns for blocked randomization and simple random sampling (SRS) as a function of the number of PDA's per centre, for imbalances $\max(n_a,n_b)/n \geq r$ and r = 0.55, 0.6, and 0.7. (N=60, number of centres=19)

Number of PDAs	Percent of imbalance					
	Fixed block size (4)			Random block size (4,6)		
	0.55	0.60	0.70	0.55	0.60	0.70
1	6	0	0	8	0	0
2	15	1	0	19	1	0
3	20	1	0	22	2	0
4	21	2	0	23	4	0
8	26	5	0	30	6	0
10	28	5	0	31	6	0
15	33	7	0	35	8	0
SRS	44	12	0	44	12	0

5.5 Imbalance at the centre level

One respect in which multiple PDAs might cause problems is that in a multi-centre study, increasing the number of PDAs will increase the chances of imbalance at the centre level, particularly at the interim analysis, when the number of participants randomized is small. Imbalance at the centre level could be a problem for several reasons. For instance, it could introduce a bias in the comparison of treatment groups due to covariate imbalance. Imbalance can be especially problematic when the imbalance in a centre is complete, i.e. all patients are randomized to the same treatment group. In this case, the centre effect and the treatment*centre interaction cannot be separated, which can make it impossible to determine exactly what's going on in a given centre. For example, suppose there are 4 patients in a centre and because 2 PDAs are used at that centre, all 4 patients happen to be randomized to the treatment group (with a single PDA and a block size of 4, this couldn't happen). If all 4 patients were to experience an event, it wouldn't be possible to separate the effect of treatment at that centre (the treatment*centre interaction) from the effect of the patient mix at that centre (the centre effect).

The frequency of complete imbalance was obtained from the same simulation which produced the results presented in table 5.5 (Appendix A.8). Whereas table 5.5 presented the probability of imbalance, table 5.6 shows what percentage of time the imbalance was complete. Figures for fixed and random block size randomization, as well as for complete randomization are shown. For example, according to table 5.5, the simulated probability of a treatment imbalance greater than or equal to 0.60 was 2% for the scenario where 4 PDAs were used per centre as well as a fixed block size of 4. From table 5.6, we learn that an estimated 26% of these cases had complete imbalance. For the case where only one PDA with pre-allocated blocking sequences of size 4 is used in each of 19 centres, one can expect 18% of centres to have allocated all their participants to only one of the treatment arms. The percentages are

similar for the scenario where a random block-size design is employed. As the number of PDAs increases, the percentages of complete imbalance reach 30%, which is the percentage one can expect with a simple random sampling scheme.

Table 5.6 Simulated percentages of complete treatment imbalance based on Astronomer recruitment patterns for blocked randomization and simple random sampling (SRS) as a function of the number of PDA's per centre. (N=60, number of centres=19)

Number of PDA	Fixed block size (4)	Random block size (4, 6)
1	18	20
2	24	25
3	25	26
4	26	27
8	28	28
10	29	29
15	29	29
SRS	30	30

5.6 Blocked vs. complete randomization

A secondary goal of the above simulations is to determine if a blocked randomization technique is to be preferred over a non-blocked randomization in this situation. As we have seen, even when there does not appear to be significant treatment allocation imbalance at the study level when a blocking strategy is employed, there may be large imbalances at the centre level. Thus, it is of interest to examine whether a blocking strategy is appropriate, or even beneficial, in this case. Note that in figures 5.3a and 5.3b described earlier, the line representing imbalance for simple randomization (SRS) is always well above those for the blocked randomization schemes. As the number of PDAs increases, the lines slowly move up vertically to meet up with the SRS probabilities of imbalance. Further to this, the histograms in figures 5.4a and b allow a comparison of the distributions of imbalance when participants are randomized to one of two treatment arms using three different

randomization methods: fixed block (size=4) randomization (white), random block (size=4 or 6) randomization (grey), and simple unblocked randomization (black).

Ideally, the distribution would be tightly concentrated around the 0.5 column, meaning that participants are equally distributed in the two treatment groups. As the number of participants and the number of centres increases, the distribution does in fact tighten around 0.5. The distributions for the fixed and the random block scenarios are very similar. However, simple randomization produces proportions of allocation that deviate much more from the equal treatment mark of 0.5, especially with smaller sample sizes.

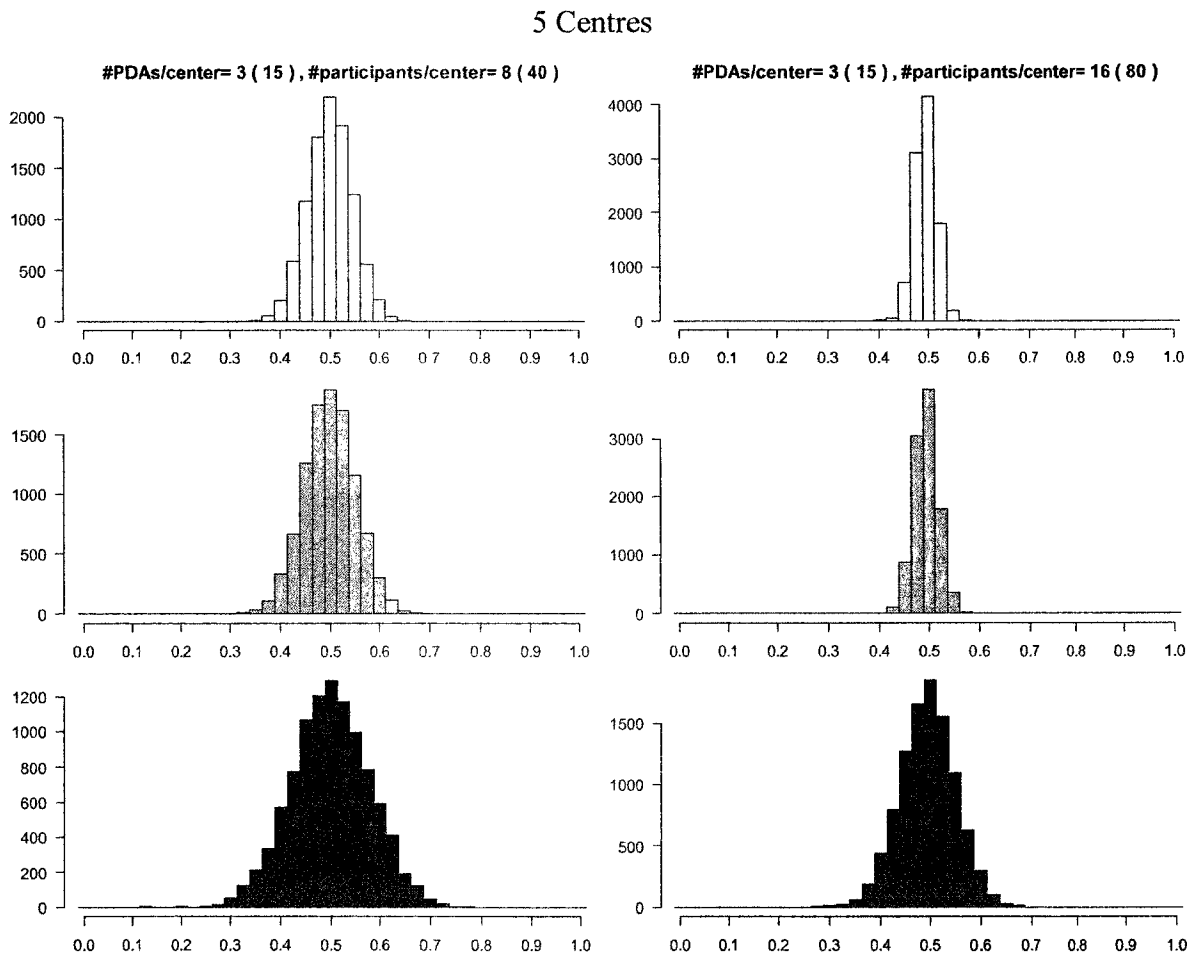


Figure 5.4a Simulation of treatment allocation with 5 centres by the number of PDAs and the number of participants. Based on 10,000 repetitions.
White : fixed size blocked randomization (4); Grey : random size blocked randomization (4,6); Black : complete randomization

15 Centres

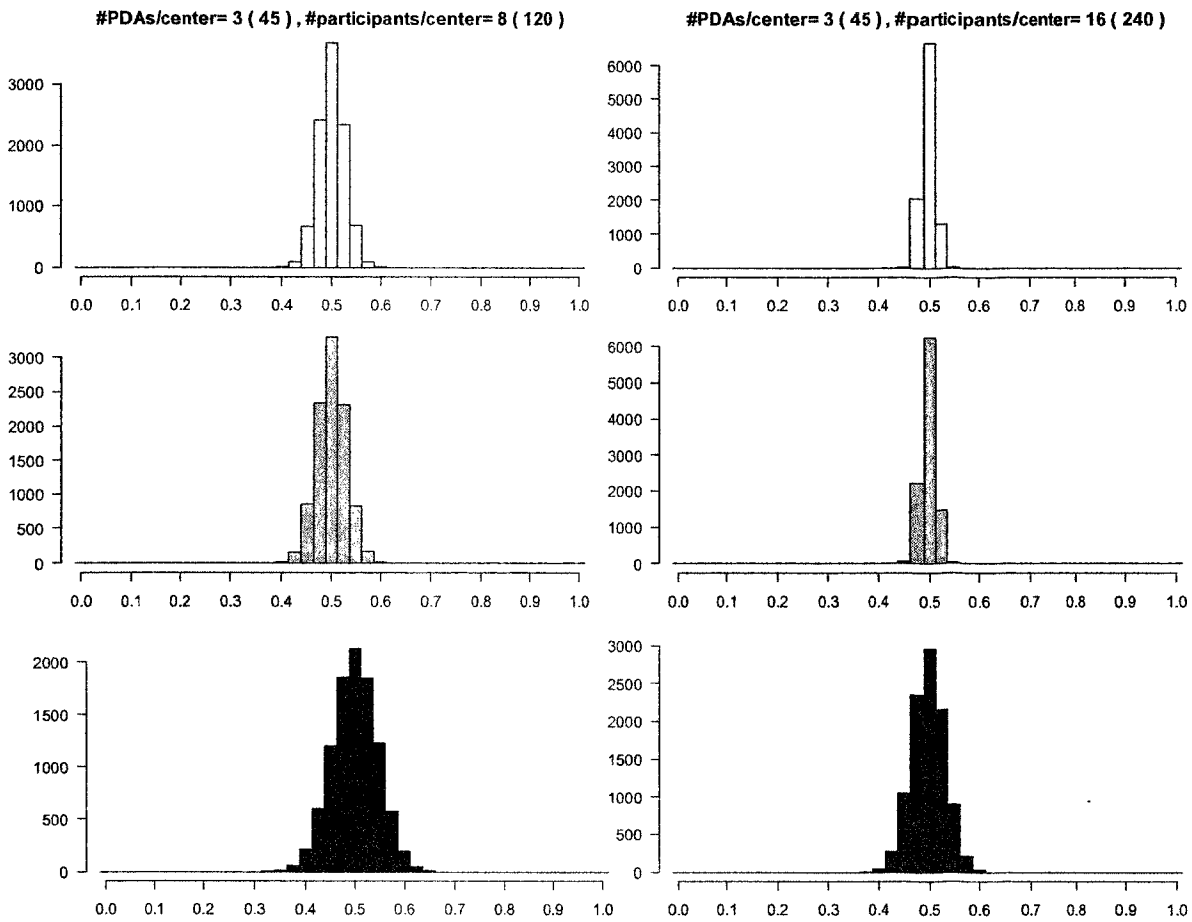


Figure 5.4b Simulation of treatment allocation for 15 centres, by the number of PDAs and the number of participants. Based on 10,000 repetitions.

White : fixed size blocked randomization (4); Grey : random size blocked randomization (4,6); Black : complete randomization

Thus, from a purely statistical point of view, blocked randomization strategies are superior to simple randomization even when incomplete blocks are anticipated. However, are there ethical issues related to the inability to pre-specify an allocation ratio? In other words, are there ethical considerations involved with the inability to provide the potential participant with a concrete probability of receiving treatment?

5.7 Conclusions

We have shown that in multi-centre trials, preloading allocation sequences into multiple PDAs can result in treatment imbalances at the study level, and especially at the centre level, depending on such factors as the number of centres, the number of PDAs and the number of participants at the time of analysis, be it interim or end of trial. However, provided the sample sizes are adequate, in most cases it is appropriate to randomize using PDAs with pre-loaded allocation sequences. Based on Lachin's findings, the aggregate allocation imbalance resulting from randomization at the PDA level is unlikely to have a large impact on the power of the trial to detect treatment differences.

The issue of imbalance at the interim analysis stage must be taken into consideration when deciding on the recruitment strategy and sample size goals. It is not sufficient to look at the total expected sample size when determining if the probabilities of imbalance are acceptable for the needs of the study. Even though the final allocation proportions may ensure adequate power, this may not be the case for interim analyses. When planning interim analyses, sample size calculations should take this fact into account.

In multi centre trials, imbalance within centres can be problematic, as it can confound the treatment effect and the centre effect. Based on the Astronomer data, the probability of complete imbalance was very high, between 18 and 30 percent depending on the number of PDAs used in the randomization process. Investigators need to decide whether this is acceptable given their study and potential treatment risks for the participants.

A general guideline is to keep the number of PDAs to a minimum as imbalance increases substantially with each additional unit. Limiting the number of centres is also key in reducing potential for imbalance.

Chapter 6

Concluding Remarks

In this thesis we have discussed how the choice of randomization scheme in clinical trials has important implications for their design and analysis. In chapters 2 and 3, we described the strengths and weaknesses associated with permutation and meta-analytical model-based inference in the context of cluster randomized clinical trials of rare events. It was determined through simulation that inference based on permutation tests in situations with allocation imbalance and rare event rates are likely to be valid as long as the sample sizes are sufficiently large to compensate for the discreteness of the randomization frequency distribution. The findings were very different when the model-based meta-analytical approach was evaluated. It was evident that meta-analysis tests for binary outcomes where risk differences are measured are not appropriate if there is a possibility of systematic allocation imbalance. Based on these results, it would be of interest to revisit the use of meta-analytical techniques in group randomized trials where different outcomes are measured, such as the odds ratio and relative risk.

The second portion of the thesis provided an evaluation of the appropriateness of using multiple PDAs to randomize participants to a multi-center RCT. It was determined that treatment imbalances resulting from the use of multiple PDAs would

rarely be severe enough to have a negative effect on inference at the end of the study. However, through the course of this evaluation, we identified some key issues that merit further consideration. For instance, ensuring a trial has sufficient power at interim to allow an accurate measure of the possibly deleterious effects of the treatment is complicated by the potential for allocation imbalance. In addition, though severe allocation imbalance at the level of the study is unlikely, it is probable that some centers experience heavy imbalance. This can confound treatment and center effects, making it difficult if not impossible to determine the true effect of the treatment. Thus, we can see how an increased understanding of the effects of allocation imbalance on the validity of inference would be highly beneficial.

In conclusion, the randomized control trial is considered by many the gold standard in health studies. As a result, some mistakenly assume that inference based on RCTs will be valid regardless of the randomization scheme. In this thesis we have shown the need for investigators to be vigilant when allocation imbalance is a possibility, whether it be at the study or the center level. Identifying the potential for allocation imbalance is a vital consideration in selecting the randomization scheme in any RCT.

Appendix A

R Programs

A.1. Simulation program for randomization-based inference with unmatched designs

```
#=============================================================================#
#                                                                             #
# Randomization-based Inference for Community Intervention Trials             #
#  with an unmatched design                                                   #
#                                                                             #
#  Based on Gail et al. paper :                                               #
#   "On design considerations and randomization-based inference for           #
#    community intervention trials"                                           #
#                                                                             #
#                                                                             #
#  Objective : Simulate the test sizes from performing permutation tests,     #
#              t-tests and normal approximation tests for a variety of        #
#              variance ratios                                                #
#                                                                             #
#=============================================================================#
#                                                                             #
# Key variable definitions:                                                   #
#    Ji   number of clusters in treatment i                                   #
#    Wij  outcome in treatment i, cluster j.                                  #
#    VarWi variance in the treatment group = i                                #
#    Uj   treatment effect in pair j                                          #
#                                                                             #
# Details/concepts:                                                           #
#    Measurability   What is the probability the subject has BP readings?     #
#    Measure         Does the subject have BP readings?  Y/N (1/0)            #
#    treatment.effect                                                         #
#      .Measure      How much did treatment affect the Measurability?         #
#                                                                             #
#=============================================================================#
```

```
# For debugging purposes                                                    #
#============================================================================#
InitShowStatus <- function() {
  showstatus.characters <<- 0
  showstatus.width <<- options()$width-5
}

ShowStatus <- function(message) {
  if (showstatus.characters > showstatus.width) {
    cat("\n")
       showstatus.characters <<-0
   }
   cat(message)
   flush.console()
   showstatus.characters <<- showstatus.characters+nchar(message)
}

# Choose a subset among the given values in v.
# This is how we pair the clusters in all choose(n,r) ways
#============================================================================#
Subsets1 <-  function(n, r, v = 1:n, set = TRUE) {

  if(r < 0 || r > n) stop("invalid r for this n")
  if(set) {
    v <- unique(sort(v))
    if (length(v) < n) stop("too few different elements")
  }
  v0 <- vector(mode(v), 0)

  sub <- function(n, r, v) { ## Inner workhorse
    if(r == 0) {
      v0
    }
    else {
      if(r == n) {
        matrix(v, 1, n)
      }
      else {
        if(choose(n,r) < 100000) {
          rbind(cbind(v[1], Recall(n-1, r-1, v[-1])),Recall(n-1, r, v[-1]))
        }
        else {
          A <- matrix(0,100000,r)
          for (i in 1:100000)  {
            A[i,] <- sample(n,r,replace=F)
          }
          A
        }
      }
    }
  }
  sub(n, r, v[1:n])
}

# The Simulation function calculates the estimated size for three tests:   #
# permutation test, t-test, and normal approximation test, based on        #
# values of J1, J2 and the variance ratio,                                 #
#============================================================================#
SizeSim <- function(numsimul,J1,J2,VarW1,VarW2,alpha,Ws,CW,reject) {
```

```
# Initializing vectors and matrices for the simulation
#-------------------------------------------------------------------------#
count.perm <- 0
count.ttest <- 0
count.norm <- 0

position <- 0
I1 <- matrix(1,J1,1)
I2 <- matrix(1,J2,1)

InitShowStatus()

# Simulation                                                              #
#-------------------------------------------------------------------------#
for (z in 1:numsimul) {


  # Generate study data for each cluster using a specified                #
  # distribution, normal or t-, and with specified variances              #
  #-------------------------------------------------------------------------#
  W1j <- matrix(1,1,J1)  ## treatment 1 (row=cluster)
  W2j <- matrix(1,1,J2)  ## treatment 2 " "

    # Normal distribution
    #-------------------------------------------------------------------------#
#   W1j <- rnorm(J1,sd=sqrt(VarW1))
#   W2j <- rnorm(J2,sd=sqrt(VarW2))

    # t distribution
    #-------------------------------------------------------------------------#
    r <- 20
    mean <- 0
    W1j <- sqrt(VarW1*(r-2)/2)*rt(J1,r) + mean
    W2j <- sqrt(VarW2*(r-2)/2)*rt(J2,r) + mean

  # Outcomes at the treatment level
  #-------------------------------------------------------------------------#
  values <- c(W1j, W2j)

  W1 <- sum(W1j)/J1
  W2 <- sum(W2j)/J2

  W <- (sum(W1j) + sum(W2j))/(J1+J2) # used in normal approximation

  # Calculate our original U
  #-------------------------------------------------------------------------#
  U <- W1 - W2

  # One-sided permuation test
  #-------------------------------------------------------------------------#
    # Perform all possible permutations of the Wij
    #-------------------------------------------------------------------------#
    if (J1 < 30 & J2 < 30) {
      W1js <- matrix(values[Ws],ncol=J1,nrow=nrow(Ws))
      W2js <- matrix(values[CW],ncol=J2,nrow=nrow(CW))
    }
    else {
      if (J1 >= 30 & J2 >= 30) {
        W1js <- matrix(NA,ncol=J1,nrow=nrow(Ws))
        W2js <- matrix(NA,ncol=J2,nrow=nrow(Ws))
```

```
      for (i in 1:nrow(Ws))  {
        W1js[i,] <- as.matrix(values[Ws[i,]])
        W2js[i,] <- as.matrix(values[CW[i,]])
      }
    }
    else {
      if (J1 >= 30)  {
        W1js <- matrix(NA,ncol=J1,nrow=nrow(Ws))
        W2js <- matrix(values[CW],ncol=J2,nrow=nrow(CW))

        for (i in 1:nrow(Ws))  {
          W1js[i,] <- as.matrix(values[Ws[i,]])
        }
      }
      else {
        if (J2 >= 30) {
          W1js <- matrix(values[Ws],ncol=J1,nrow=nrow(Ws))
          W2js <- matrix(NA,ncol=J2,nrow=nrow(Ws))

          for (i in 1:nrow(Ws))  {
            W2js[i,] <- as.matrix(values[CW[i,]])
          }
        }
      }
    }
  }

  W1s <- (W1js %*% I1)/J1
  W2s <- (W2js %*% I2)/J2

  # Calculate U for each permutation of the Wijs
  #-------------------------------------------------------------------------#
  Us <- c(W1s-W2s)

  # Is U in the rejection zone?
  #-------------------------------------------------------------------------#
  Usorted <- sort(Us, decreasing=TRUE)
  Ureject <- Usorted[reject]
  count.perm <- (U-Ureject >= 0) + count.perm

# unpaired t-test
#-----------------------------------------------------------------------------#
  # unpaired t-statistic
  #-------------------------------------------------------------------------#
  Var1 <- (J1-1)^(-1) * sum((W1j-W1)^2)
  Var2 <- (J2-1)^(-1) * sum((W2j-W2)^2)
  t <- U * ((((J1-1)*Var1 + (J2-1)*Var2)/(J1+J2-2))*(1/J1 + 1/J2))^(-1/2)

  # Is t in the rejection zone?
  #-------------------------------------------------------------------------#
  ttest.statistic <- qt(1-alpha/2,J1+J2-2)
  count.ttest <- (abs(t) - ttest.statistic > 0) + count.ttest

# normal approximation
#-----------------------------------------------------------------------------#
  # Variance of the finite population of cluster responses
  #-------------------------------------------------------------------------#
  Var.p <- sum((c(W1j,W2j)-W)^2)/(J1+J2)
```

```
      # permutational distrib of U as approx normal, with mean 0 and variance :
      #-------------------------------------------------------------------#
      omega <- (Var.p*(J1+J2)^2)/(J1*J2*(J1+J2-1))

      # normal approximate deviate
      #-------------------------------------------------------------------#
      z.norm <- U * omega^(-1/2)

      # Is Z in the rejection zone?
      #-------------------------------------------------------------------#
      z.test.statistic <- qnorm(1-alpha)
      count.norm <- (z.norm - z.test.statistic > 0) + count.norm

    # print a message every 100 iterations
    #-------------------------------------------------------------------#
    if ((z %% 100)==0) {
      ShowStatus(paste(" ",z))
      ShowStatus(paste(" count.perm=",count.perm," count.ttest=",count.ttest,"
       count.norm=",count.norm))
    }
  } # end of simulation
  cat("\n")  # newline

list(count.perm,count.ttest,count.norm)
}

# Determines the rejection area for the permutation test
#=====================================================================#
Reject <- function(J1,J2,alpha) {
  n <- choose(J1+J2, J1)
  if (n > 100000) {
    reject <- floor(100000*alpha)
  }
  else {
    reject <- floor(alpha*n)
  }
}

# Define the variables
#=====================================================================#

  # Values of PSI=VarW2/VarW1
  #-------------------------------------------------------------------#
  VARW1  <- rep(1,7)
  VARW2  <- c(1/5,1/3,1/2,1,2,3,5)
  psi    <- VARW2
  psilen <- length(psi)

  # Values of J1 and J2
  #-------------------------------------------------------------------#

  J      <- matrix(0,5,2)
  J[1,]  <- c(3,3)
  J[2,]  <- c(5,5)
  J[3,]  <- c(3,6)
  J[4,]  <- c(6,3)
  J[5,]  <- c(10,10)
  Jlen   <- dim(J)[1]
```

```
# Calls the simulation function for all combinations of PSI and J1 and J2
# Creates table for each test (permutation, t- and normal approximation)
#=============================================================================#
SizeTab <- function(NUMSIMUL, ALPHA, J, VARW1, VARW2, psi) {

  Jlen   <- dim(J)[1]
  psilen <- length(psi)

  SIZE.perm  <- matrix(0,Jlen,psilen)
  SIZE.ttest <- matrix(0,Jlen,psilen)
  SIZE.norm  <- matrix(0,Jlen,psilen)
  EXCEEDS    <- matrix(0,Jlen*3,psilen)

  for (j in 1:Jlen) {
    j1 <- J[j,1]
    j2 <- J[j,2]

    # Permutations of treatment allocation
    #-------------------------------------------------------------------------#
      # w gets treatment
      w <- Subsets1(j1+j2,j1)

      # The [c]omplement of [w] gets control
      cw <- matrix(NA,nrow=nrow(w),ncol=j2)
      for (i in 1:dim(cw)[1])  {
        cw[i,] <- (1:(j1+j2))[-w[i,]]
      }

    Rejection criteria
    #-------------------------------------------------------------------------#
    REJECT <- Reject(j1,j2,ALPHA)

    # Call SizeSim for every combination of J1, J2 and PSI
    #-------------------------------------------------------------------------#
    for (i in 1:psilen) {
      cat(date()," j1=",j1," j2=",j2," psi=",VARW2[i],"\n",sep="")
      flush.console()

      varW1 <- VARW1[i]
      varW2 <- VARW2[i]
      out <- SizeSim(NUMSIMUL,j1,j2,varW1,varW2,ALPHA,w,cw,REJECT)
      SIZE.perm[j,i] <- out[[1]]
      SIZE.ttest[j,i] <- out[[2]]
      SIZE.norm[j,i] <- out[[3]]

    }
  }
  print("Finished")
  print(date())
list(SIZE.perm, SIZE.ttest, SIZE.norm)
}

A <- SizeTab(NUMSIMUL=10000, ALPHA=0.05, J, VARW1, VARW2, psi)


# Create final table that contains all three test results
#=============================================================================#
SIZE.perm <- A[[1]]
SIZE.ttest <- A[[2]]
SIZE.norm <- A[[3]]
```

```
SIZE <- matrix(NA,nrow=Jlen*3,ncol=psilen)
SIZE <- rbind(SIZE.perm[1,],SIZE.ttest[1,],SIZE.norm[1,],
              SIZE.perm[2,],SIZE.ttest[2,],SIZE.norm[2,],
              SIZE.perm[3,],SIZE.ttest[3,],SIZE.norm[3,],
              SIZE.perm[4,],SIZE.ttest[4,],SIZE.norm[4,],
              SIZE.perm[5,],SIZE.ttest[5,],SIZE.norm[5,])

# Calculate nominal size
#=============================================================================#
NUMSIMUL <- 10000
nominal.size <- NUMSIMUL*(ALPHA)
a <- nominal.size/NUMSIMUL
CIlower <- NUMSIMUL*(a-(qnorm(1-ALPHA/2)*sqrt(a*(1-a)/NUMSIMUL)))
CIupper <- NUMSIMUL*(a+(qnorm(1-ALPHA/2)*sqrt(a*(1-a)/NUMSIMUL)))
nominal.CI <- matrix(c(CIlower,CIupper),nrow=1,ncol=2)

# Create table that indicates if estimate exceeds upper nominal CI bound
#=============================================================================#
  # Does SIZE exceed upper CI bound for nominal size?
  EXCEEDS <-(SIZE > nominal.CI[1,2])

# Formatting
#=============================================================================#
Jmatrix <- rbind(t(matrix(rep(J[1,],3),nrow=2)),t(matrix(rep(J[2,],3),nrow=2)),
                 t(matrix(rep(J[3,],3),nrow=2)),t(matrix(rep(J[4,],3),nrow=2)),
                 t(matrix(rep(J[5,],3),nrow=2)))

unmatched.table <- cbind(Jmatrix,SIZE)
dimnames(unmatched.table) <- list(NULL,c("J1","J2","1/5","1/3","1/2",
                                         "1","2","3","5"))

dimnames(nominal.CI) <- list(NULL,c("Lower","Upper"))

unmatched.exceeds <- cbind(Jmatrix,EXCEEDS)
dimnames(unmatched.exceeds) <- list(NULL,c("J1","J2","1/5","1/3","1/2",
                                           "1","2","3","5"))

unmatched.table
nominal.size
nominal.CI
unmatched.exceeds


# Graph the results
#=============================================================================#
Graphs <- function(psi,table) {
    # all rows from the permutation test
  for (i in 1:Jlen)  {
    if (i==1) {
      plot(psi,table[i,],xlim=range(psi),ylim=range(table),type="l",
           main="Estimated size times 10000 of the one-sided permutation test
       \n
                 for various variance ratios",
           xlab="Variance Ratios",ylab="Size")
    }
    else {
     lines(psi,table[i,],lty=i)
    }
  }
```

```
  legend(locator(n=1),legend=c("J1=3,J2=3","J1=5,J2=5","J1=3,J2=6","J1=6,J2=3",
                               "J1=10,J2=10"),lty=1:J1en)
}

Graphs(psi,SIZE.perm)
```

A.2. Simulation program for randomization-based inference with matched paired designs

```
#=============================================================================#
#                                                                             #
# Randomization-based Inference for Community Intervention Trials             #
#  with a matched paired design                                               #
#                                                                             #
#  Based on Gail et al. paper :                                               #
#   "On design considerations and randomization-based inference for           #
#    community intervention trials"                                           #
#                                                                             #
#                                                                             #
#  Objective : Simulate the test sizes from performing permutation tests     #
#              and t-tests for a variety of imput parameters                  #
#                                                                             #
#=============================================================================#
#                                                                             #
# Key variable definitions:                                                   #
#    J    number of pairs                                                     #
#    Kij  number of subjects in treatment i, pair j                           #
#    Wij  outcome in treatment i, pair j. A combination of the Yijk.     #
#    Uj   treatment effect in pair j                                          #
#                                                                             #
# Details/concepts:                                                           #
#    Measurability   What is the probability the subject has BP readings?  #
#    Measure         Does the subject have BP readings?  Y/N (1/0)          #
#    treatment.effect                                                         #
#      .Measure      How much did treatment affect the Measurability?       #
#                                                                             #
#=============================================================================#

# For debugging purposes                                                      #
#=============================================================================#
InitShowStatus <- function() {
  showstatus.characters <<- 0
  showstatus.width <<- options()$width-5
}

ShowStatus <- function(message) {
  if (showstatus.characters > showstatus.width) {
    cat("\n")
       showstatus.characters <<-0
   }
   cat(message)
   flush.console()
   showstatus.characters <<- showstatus.characters+nchar(message)
}


# Permuting the outcomes                                                      #
#=============================================================================#
# Constructs a full matrix of all possibilities if 2^J < 100,000,            #
# else creates a matrix of 100000 randomly generated possibilites             #
```

```
#=============================================================================#
Permute <- function(k) {
  if (k==0) {
    NULL
  }
  else {
    if (2^k < 100000) {
      rbind(cbind(1,Recall(k-1)),cbind(-1,Recall(k-1)))
    }
    else {
      A <- matrix(0,100000,k)
        for (reps in 1:100000) {
          A[reps,] <- sample(c(1,-1),k,replace=T,prob=c(0.5,0.5))
        }
    A
    }
  }
}

# The Simulation function calculates U and its plausibility based on all   #
# possible permutations of the outcomes a total of numsimul times.        #
# Produces an estimate of size of the tests                                #
#=============================================================================#
SizeSim <- function(numsimul,J,K1j,K2j,p,alpha,permute,reject) {

  InitShowStatus()

  # Vector of cluster sizes for treatment 1 and 2
  K1 <- rep(K1j,J)
  K2 <- rep(K2j,J)

  # Definitions of vectors used for test of significance
  U <- rep(0,numsimul)
  count.perm <- count.ttest <- position <- 0

  # Simulation                                                             #
  #-------------------------------------------------------------------------#
  for (z in 1:numsimul) {

    # Create study data for each practice                                  #
    # Proportions of measurement are based on SMART probabilities          #
    #-----------------------------------------------------------------------#
      # Calculate the Wij at the cluster level
      W1    <- matrix(1,J,1)
      W2    <- matrix(1,J,1)

      for (j in  1:J) {
        W1[j]    <- rbinom(1,K1[j],p)/K1[j]
        W2[j]    <- rbinom(1,K2[j],p)/K2[j]
      }

      # Calculate our original U
      #---------------------------------------------------------------------#
      Uj <- W1 - W2
      U  <- sum(Uj)/J

      # Permutations of the outcomes Wij to the 2J clusters
      #---------------------------------------------------------------------#
      Us <- (permute %*% Uj)/J
```

```
      # Permutation test. Is U in the rejection zone?
      #-------------------------------------------------------------------#
      Usorted <- c(sort(Us, decreasing=TRUE))
      Ureject <- Usorted[reject]
      if (!(all(c(Uj,U)==0)) {
            count.perm <- (U-Ureject >= 0) + count.perm
      }

    # paired t-test
    #-------------------------------------------------------------------#
      # paired t-statistic
      #-------------------------------------------------------------------#
      if (all(c(Uj,U)==0) {
          t <- 0 }
      else {
          Var.t <- (J-1)^(-1) * sum((Uj-U)^2)
          t <- U * (Var.t / J)^(-1/2)
      }

      # Is t in the rejection zone?  This is a one-sided
      #-------------------------------------------------------------------#
        ttest.statistic <- qt(1-alpha/2,J-1)
#       ttest.statistic <- qt(1-alpha,J-1)
        count.ttest <- (abs(t) - ttest.statistic > 0) + count.ttest

    # print a message every 1000 iterations
    #-------------------------------------------------------------------#
    if ((z %% 1000)==0) {
      ShowStatus(paste("",z))
      ShowStatus(paste(" count=",count.perm))
    }
  } # end of simulation
  cat("\n")  # newline

  list(count.perm, count.ttest)
}


# Determines the rejection area for the permutation test
#=====================================================================#
Reject <- function(N,alpha) {
  n <- 2^N
  if (n > 100000) {
    reject <- 100000*alpha
  }
  else {
    reject <- floor(alpha*n)
  }
reject
}

# Define the variables
#=====================================================================#
  # Number of [pairs]
  #-------------------------------------------------------------------#
  J    <- c(5,10,20,40)
  Jlen <- length(J)

  # Number of subjects[Kij] in the treatment(i=1) and control(i=2) groups
  #-------------------------------------------------------------------#
```

```
  KIJ      <- matrix(0,20,2)
  KIJ[1,]  <- c(100,9)
  KIJ[2,]  <- c(100,29)
  KIJ[3,]  <- c(100,49)
  KIJ[4,]  <- c(100,99)
  KIJ[5,]  <- c(400,39)
  KIJ[6,]  <- c(400,119)
  KIJ[7,]  <- c(9,100)
  KIJ[8,]  <- c(29, 100)
  KIJ[9,]  <- c(49,100)
  KIJ[10,] <- c(99, 100)
  KIJ[11,] <- c(39, 400)
  KIJ[12,] <- c(119, 400)
  KIJ[13,] <- c(9,9)
  KIJ[14,] <- c(29,29)
  KIJ[15,] <- c(39,39)
  KIJ[16,] <- c(49,49)
  KIJ[17,] <- c(100,100)
  KIJ[18,] <- c(200,200)
  KIJ[19,] <- c(300,300)
  KIJ[20,] <- c(400,400)
  KIJlen   <- dim(KIJ)[1]


# Calls the simulation function for all combinations of J and Kij
# Creates table for each test (permutation, t-) for combinations of Kij
#=============================================================================#
SizeTab <- function(NSIMUL,P,ALPHA,pairs,pairslen, Kij, Kijlen) {

  SIZE.perm <- SIZE.ttest <- SIZE.exact <- matrix(NA,Kijlen,pairslen)

  for (i in 1:pairslen) {
    PAIRS <- pairs[i]
    NUMSIMUL <- NSIMUL
    PERMUTE <- Permute(PAIRS)

    for (j in 1:Kijlen) {
      K1J <- Kij[j,1]
      K2J <- Kij[j,2]
      REJECT <- Reject(PAIRS,ALPHA)

      cat("Time Start=",date()," PAIRS=",PAIRS," K1j=",K1J,"
       K2j=",K2J,"\n",sep="")
      flush.console()

      SIZE.perm[j,i]  <-
       SizeSim(NUMSIMUL,PAIRS,K1J,K2J,P,ALPHA,PERMUTE,REJECT)[[1]]
      SIZE.ttest[j,i] <-
       SizeSim(NUMSIMUL,PAIRS,K1J,K2J,P,ALPHA,PERMUTE,REJECT)[[2]]
      SIZE.exact[j,i] <-
       sum(pbinom((K1J/K2J)*0:K2J,prob=P,size=K1J,lower.tail=FALSE)*
                          dbinom(0:K2J,prob=P,size=K2J))^PAIRS
    }
  cat("Time end=",date())
  flush.console()
  }

  # Pretty up the output
  perm    <- cbind(Kij,SIZE.perm)
  ttest   <- cbind(Kij,SIZE.ttest)
```

```
  exact    <- cbind(Kij,SIZE.exact)

list(perm,ttest,exact)
}


p.1<-SizeTab(NSIMUL=10000,P=0.1,ALPHA=0.05, J, Jlen, KIJ, KIJlen)
p.5<-SizeTab(NSIMUL=10000,P=0.5,ALPHA=0.05, J, Jlen, KIJ, KIJlen)


# Create final table that contains all three test results
#=============================================================================#
SIZE.perm <- cbind(p.1[[1]][,1:3],p.5[[1]][,3],
                   p.1[[1]][,4],  p.5[[1]][,4],
                   p.1[[1]][,5],  p.5[[1]][,5],
                   p.1[[1]][,6],  p.5[[1]][,6])

SIZE.ttest<- cbind(p.1[[2]][,1:3],p.5[[2]][,3],
                   p.1[[2]][,4],  p.5[[2]][,4],
                   p.1[[2]][,5],  p.5[[2]][,5],
                   p.1[[2]][,6],  p.5[[2]][,6])

SIZE.exact<- cbind(p.1[[3]][,1:3],p.5[[3]][,3],
                   p.1[[3]][,4],  p.5[[3]][,4],
                   p.1[[3]][,5],  p.5[[3]][,5],
                   p.1[[3]][,6],  p.5[[3]][,6])

SIZE <- rbind(SIZE.perm[1,],SIZE.ttest[1,],SIZE.exact[1,],
              SIZE.perm[2,],SIZE.ttest[2,],SIZE.exact[2,],
              SIZE.perm[3,],SIZE.ttest[3,],SIZE.exact[3,],
              SIZE.perm[4,],SIZE.ttest[4,],SIZE.exact[4,],
              SIZE.perm[5,],SIZE.ttest[5,],SIZE.exact[5,],
              SIZE.perm[6,],SIZE.ttest[6,],SIZE.exact[6,],
              SIZE.perm[7,],SIZE.ttest[7,],SIZE.exact[7,],
              SIZE.perm[8,],SIZE.ttest[8,],SIZE.exact[8,],
              SIZE.perm[9,],SIZE.ttest[9,],SIZE.exact[9,],
              SIZE.perm[10,],SIZE.ttest[10,],SIZE.exact[10,],
              SIZE.perm[11,],SIZE.ttest[11,],SIZE.exact[11,],
              SIZE.perm[12,],SIZE.ttest[12,],SIZE.exact[12,],
              SIZE.perm[13,],SIZE.ttest[13,],SIZE.exact[13,],
              SIZE.perm[14,],SIZE.ttest[14,],SIZE.exact[14,],
              SIZE.perm[15,],SIZE.ttest[15,],SIZE.exact[15,],
              SIZE.perm[16,],SIZE.ttest[16,],SIZE.exact[16,],
              SIZE.perm[17,],SIZE.ttest[17,],SIZE.exact[17,],
              SIZE.perm[18,],SIZE.ttest[18,],SIZE.exact[18,],
              SIZE.perm[19,],SIZE.ttest[19,],SIZE.exact[19,],
              SIZE.perm[20,],SIZE.ttest[20,],SIZE.exact[20,])

dimnames(SIZE) <- list(NULL,c("K1j","K2j","J=5,p=0.1","J=5,p=0.5",
                                          "J=10,p=0.1","J=10,p=0.5",
                                          "J=20,p=0.1","J=20,p=0.5",
                                          "J=40,p=0.1","J=40,p=0.5",))


# Calculate nominal size
#=============================================================================#
NUMSIMUL <- 10000
nomsize.perm  <- rep(NUMSIMUL*floor(0.05*2^pairs)/2^pairs, each=2)
nomsize.ttest <- rep(NUMSIMUL*0.05,length(pairs)*2)
```

```
# Create table to indicates if estimate exceeds nominal size by >= 20%
#=============================================================================#
EXCEEDS.perm  <- t(t(SIZE.perm)  >= 1.2*nomsize.perm)
EXCEEDS.ttest <- t(t(SIZE.ttest) >= 1.2*nomsize.ttest)

# Graph the results
#=============================================================================#
Graphs <- function(pairs, table) {

  for (i in 1:Kijlen)  {
    if (i==1) {
      plot(pairs,table[i,],xlim=range(pairs),ylim=range(table),type="l",
           main="Estimated size times 10,000 of the one-sided permutation test
       \n
                  for Bernoulli probabilities p=0.1 and p=0.5",
           xlab="Number of pairs",ylab="Size")
    }
    else {
      lines(pairs,table[i,],lty=i)
    }
  }
  legend(locator(n=1),

       legend=c("K1j=100,K2j=9","K1j=100,K2j=29","K1j=100,K2j=49","K1j=100,K2j=9
       9",
              "K1j=400,K2j=39","K1j=400,K2j=119","K1j=9,  K2j=100","K1j=29,
       K2j=100",
              "K1j=49, K2j=100","K1j=99, K2j=100","K1j=39,
       K2j=400","K1j=119,K2j=440"),
         lty=1:Kijlen)
}
Graphs(pairs, SIZE.perm)
```

A.3.Simulation program for inference based on meta-analysis

```
#=============================================================================#
#                                                                             #
# Inference based on meta-analytical techniques                               #
#                                                                             #
#  Based on Thompson et al. paper :                                           #
#   "The design and analysis of paired cluster randomized trials:             #
#    An application of meta-anlysis techniques"                               #
#                                                                             #
#  Theory for estimation of the variance of the true effect sizes             #
#  based on paper by Brockwell et al. :                                       #
#    "A comparison of statistical methods for meta-analysis                   #
#                                                                             #
#  Objective : Simulate the coverage probabilities resulting from             #
#              application of fixed and random effect meta-analysis           #
#              methods                                                        #
#                                                                             #
#=============================================================================#
#                                                                             #
# Key variable definitions:                                                   #
#    J    number of pairs                                                     #
#    Kij  number of subjects in treatment i, pair j                           #
#    Xij  outcome in treatment i, pair j. A combination of the Yijk.     #
#    dj   treatment effect in pair j                                          #
#    wj   weights                                                             #
#                                                                             #
#=============================================================================#


# For debugging purposes                                                      #
#=============================================================================#

InitShowStatus <- function() {
  showstatus.characters <<- 0
  showstatus.width <<- options()$width-5
}

ShowStatus <- function(message) {
  if (showstatus.characters > showstatus.width) {
    cat("\n")
       showstatus.characters <<-0
   }
   cat(message)
   flush.console()
   showstatus.characters <<- showstatus.characters+nchar(message)
}


# The Simul function calculates global treatment effects based on             #
# application of fixed and random meta-analytical techniques as well as      #
# corresponding coverage probabilities                                        #
#=============================================================================#
```

```
Simul <- function(Numsimul,p,n.1j,n.2j,k) {

  count.fixed <- count.MH <- count.m.MH <- count.random <- 0

  mX.1j <- mn.1j <- rep(NA,k) # [m]odified vectors
  mX.2j <- mn.2j <- rep(NA,k) # [m]odified vectors

  # Simulation                                                              #
  #-------------------------------------------------------------------------#
  for (z in 1:Numsimul) {
    # print a message every 1000 iterations
    if ((z %% 1000)==0) ShowStatus(paste("",z))

    # Create study data for each practice                                   #
    # Since outcomes are binary with probability p for each individual      #
    #  in a practice, the number of events in each practice follows a       #
    #  binomial distribution                                                #
    #-------------------------------------------------------------------------#
    X.1j <- rbinom(k,n.1j,p)
    X.2j <- rbinom(k,n.2j,p)

      # Continuity correction
      # For each index j, if there is a zero in any of the 4 cells of the  #
      # implicit 2 x 2 table then add 0.5 to all 4 cells                    #
      #-------------------------------------------------------------------------#
      zero <- X.1j==0 | X.2j==0 | X.1j==n.1j | X.2j==n.2j

      mX.1j[!zero]  <- X.1j[!zero]
      mX.1j[zero] <- X.1j[zero] + 0.5

      mn.1j[!zero]  <- n.1j[!zero]
      mn.1j[zero] <- n.1j[zero] + 1

      mX.2j[!zero]  <- X.2j[!zero]
      mX.2j[zero] <- X.2j[zero] + 0.5

      mn.2j[!zero]  <- n.2j[!zero]
      mn.2j[zero] <- n.2j[zero] + 1

    # proportion of BP readings in each practice
    #-------------------------------------------------------------------------#
    p.1j <- X.1j/n.1j
    p.2j <- X.2j/n.2j
    mp.1j <- mX.1j/mn.1j
    mp.2j <- mX.2j/mn.2j

    # treatment effect measured as difference in proportions
    #-------------------------------------------------------------------------#
    dj <- p.1j - p.2j
    mdj <- mp.1j - mp.2j

    # FIXED EFFECTS
    #-------------------------------------------------------------------------#
    sigma2 <- mp.1j*(1-mp.1j)/mn.1j + mp.2j*(1-mp.2j)/mn.2j
    wj <- 1/sigma2

      # Overall effect (weighted average of the dj across strata)
      #-------------------------------------------------------------------------#
      d.w <- sum(wj*mdj)/sum(wj)
```

```
      # Cochran's Q  to measure heterogeneity
      #---------------------------------------------------------------------#
      qw <- sum(wj*(mdj-d.w)^2)

      # Estimate of tau (Brockwell)  or 2*sigma2B (Thompson)
      #---------------------------------------------------------------------#
      t <- (qw - (k-1))/(sum(wj) - (sum(wj^2))/sum(wj))
      tau2 <- t*(t>=0)

      # Risk Difference weights
      #---------------------------------------------------------------------#
      wj.MH <- n.1j*n.2j/(n.1j+n.2j)
      d.w.MH <- sum(wj.MH*dj)/sum(wj.MH)
      a <- X.1j
      b <- n.1j - a
      c <- X.2j
      d <- n.2j - c
      var.d.w.MH <- sum((a*b*n.2j^3 +
c*d*n.1j^3)/(n.1j*n.2j*(n.1j+n.2j)^2))/(sum(n.1j*n.2j/(n.1j+n.2j)))^2

        # with continuity correction
        #-------------------------------------------------------------------#
        mwj.MH <- mn.1j*mn.2j/(mn.1j + mn.2j)
        md.w.MH <- sum(mwj.MH*mdj)/sum(mwj.MH)
        a <- mX.1j
        b <- mn.1j - a
        c <- mX.2j
        d <- mn.2j - c
        var.md.w.MH <- sum((a*b*mn.2j^3 +
c*d*mn.1j^3)/(mn.1j*mn.2j*(mn.1j+mn.2j)^2))/(sum(mn.1j*mn.2j/(mn.1j+mn.2j)))^2


      # CI with Mantel Haenszel methods
      #---------------------------------------------------------------------#
      lo.MH <- d.w.MH - 1.96*sqrt(var.d.w.MH)
      hi.MH <- d.w.MH + 1.96*sqrt(var.d.w.MH)
      # with continuity correction
      lo.m.MH <- md.w.MH - 1.96*sqrt(var.md.w.MH)
      hi.m.MH <- md.w.MH + 1.96*sqrt(var.md.w.MH)

      # CI with inverse variance fixed effects
      #---------------------------------------------------------------------#
      lo.fixed <- d.w - 1.96*sqrt(1/sum(wj))
      hi.fixed <- d.w + 1.96*sqrt(1/sum(wj))

    # RANDOM EFFECTS
    #-----------------------------------------------------------------------#
      # Estimate of var(dj)
      #---------------------------------------------------------------------#
      var.dj <- tau2 + sigma2

      # weights (wt), estimator of dj (effect of intervention) and
      # variance of dj
      #---------------------------------------------------------------------#
      w <- var.dj^(-1)

      # estimator of dj
      #---------------------------------------------------------------------#
      dw <- sum(w*mdj)/sum(w)
```

```
      var.dw <- (sum(w))^(-1)

      # CI with random effects
      #-------------------------------------------------------------------------#
      lo.random <- dw - 1.96*sqrt(var.dw)
      hi.random <- dw + 1.96*sqrt(var.dw)

    # CALCULATE COUNTS
    # If the CI contains 0, increase count
    #-------------------------------------------------------------------------#
    count.MH <- (!(lo.MH < 0 && hi.MH >0)) + count.MH
    count.m.MH <- (!(lo.m.MH < 0 && hi.m.MH >0)) + count.m.MH
    count.fixed <- (!(lo.fixed < 0 && hi.fixed >0)) + count.fixed
    count.random <- (!(lo.random < 0 && hi.random >0)) + count.random

   } # end for z

list(count.MH, count.m.MH, count.fixed, count.random)
} # end simul


# The sim function calls the Simul function for a variety of variable
# combination
#-----------------------------------------------------------------------------#
sim <- function(NSIMUL) {
  InitShowStatus()

  # Define the variables

  # Number of [pairs]
  #-------------------------------------------------------------------------#
  pairs <- c(5,10,20,40)
  pairs.len <- length(pairs)

  # Number of subjects[K] in treatment arm[i] and pair[j]
  #-------------------------------------------------------------------------#
  K1j <- c(100,100,100,100,400,400,9,29,49,99,39,119,9,29,39,49,99,119,
           200,300,400)
  K2j <- c(9,29,49,99,39,119,100,100,100,100,400,400,9,29,39,49,99,119,
           200,300,400)
  Kij.len <- length(K1j)

  # Level for the t-test
  #-------------------------------------------------------------------------#
  alpha <- 0.05

  # Initialize result tables
  #-------------------------------------------------------------------------#
  SIZE.MH <- SIZE.mMH <- SIZE.fixed <- SIZE.random <-
             matrix(NA,ncol=(pairs.len*2),nrow=Kij.len)

  # The workhorse
  #-------------------------------------------------------------------------#
  for (j in 1:pairs.len){
    for (i in 1:Kij.len) {

      n1j <- rep(K1j[i],pairs[j])
      n2j <- rep(K2j[i],pairs[j])
      PAIRS <- pairs[j]
```

```
      set.seed(rs)

      p.1 <- Simul(Numsimul=NSIMUL,p=0.1,n1j,n2j,PAIRS)

      set.seed(rs)

      p.5 <- Simul(Numsimul=NSIMUL,p=0.5,n1j,n2j,PAIRS)

      SIZE.MH[i,(2*j - 1)] <- p.1[[1]]
      SIZE.MH[i,(2*j)]     <- p.5[[1]]

      SIZE.mMH[i,(2*j - 1)] <- p.1[[2]]
      SIZE.mMH[i,(2*j)]     <- p.5[[2]]

      SIZE.fixed[i,(2*j - 1)] <- p.1[[3]]
      SIZE.fixed[i,(2*j)]     <- p.5[[3]]

      SIZE.random[i,(2*j - 1)] <- p.1[[4]]
      SIZE.random[i,(2*j)]     <- p.5[[4]]

  } # end for i
} # end for j
cat("\n")

list(SIZE.MH, SIZE.mMH, SIZE.fixed, SIZE.random)
}

A <- sim(10000)
```

A.4. Simulation program for matched randomization-based inference based on CHAT

```
#=============================================================================#
#                                                                             #
# Randomization-based Inference for the CHAT Community Intervention Trials #
#                                                                             #
#  Based on Gail et al. paper :                                               #
#   "On design considerations and randomization-based inference for           #
#    community intervention trials"                                           #
#                                                                             #
#                                                                             #
#  Objective : Simulate the test sizes from performing a permutation test  #
#              and a t-test based on CHAT design                              #
#                                                                             #
#=============================================================================#
#                                                                             #
# Key variable definitions:                                                   #
#    J    number of pairs                                                     #
#    Kij  number of subjects in treatment i, pair j                           #
#    Wij  outcome in treatment i, pair j. A combination of the Yijk.     #
#    Uj   treatment effect in pair j                                          #
#                                                                             #
# Details/concepts:                                                           #
#    Measurability   What is the probability the subject has BP readings?  #
#    Measure         Does the subject have BP readings?  Y/N (1/0)           #
#    treatment.effect                                                         #
#      .Measure      How much did treatment affect the Measurability?       #
#                                                                             #
# Outcomes of interest:                                                       #
#    1) Difference in proportion of patients with BP reading in the         #
#       past 12 months                                                        #
#=============================================================================#


# For debugging purposes                                                      #
#=============================================================================#
InitShowStatus <- function() {
  showstatus.characters <<- 0
  showstatus.width <<- options()$width-5
}

ShowStatus <- function(message) {
  if (showstatus.characters > showstatus.width) {
    cat("\n")
       showstatus.characters <<-0
   }
   cat(message)
   flush.console()
   showstatus.characters <<- showstatus.characters+nchar(message)
}

# Permuting the outcomes                                                      #
#=============================================================================#
```

```
# Method A constructs a full matrix of all possibilities if 2^J < 100,000, #
# else creates a matrix of 100000 randomly generated possibilites          #
#===========================================================================#
Permute <- function(k) {
  if (k==0) {
    NULL
  }
  else {
    if (2^k < 100000) {
      rbind(cbind(1,Recall(k-1)),cbind(-1,Recall(k-1)))
    }
    else {
      A <- matrix(0,100000,k)
        for (reps in 1:100000) {
          A[reps,] <- sample(c(1,-1),k,replace=T,prob=c(0.5,0.5))
        }
    A
    }
  }
}

# Determining the value of M.mu that will allow us to randomly generate    #
#  data with measurement probabilities similar to the SMART data           #
#===========================================================================#
SMART.values <- function(Smart,K1j,K2j) {
  val <- 2
  continue <- 1

  while (continue == 1) {
    M <- rep(NA,10000)
    for (i in 1:10000) {
      M.latent <- rnorm(K1j+K2j,mean=val,sd=.1)
      m <- runif(110) < exp(M.latent)/(1+exp(M.latent))
      M[i] <- mean(m)
    }

    res <- mean(M)
    if (res < (Smart-0.005)) val <- val + .1
    if (res > (Smart+0.005)) val <- val - .1
    if (res > (Smart-0.005) & res < (Smart+0.005)) continue <- 0
  }

  res <- mean(M)
  list(val)
}

# The Simulation function calculates U and its plausibility based on all   #
# possible permutations of the outcomes a total of numsimul times          #
#===========================================================================#
SizeSim <- function(numsimul, J, K1j, K2j, p.M, alpha, permute, reject)
  {
   InitShowStatus()

  # Vector of community sizes for treatment 1 and 2
  K1 <- rep(K1j,J)
  K2 <- rep(K2j,J)

  # Definitions of vectors used for test of significance
  U <- rep(0,numsimul)
  count.perm <- count.ttest <- count.t.perm <- 0
```

```
  # Simulation                                                              #
  #-------------------------------------------------------------------------#
  for (z in 1:numsimul) {

    # Create study data for each practice                                   #
    # Proportions of measurement are based on SMART probabilities           #
    #-------------------------------------------------------------------------#
    W1    <- matrix(1,J,1)
    W2    <- matrix(1,J,1)

    for (j in  1:J) {
      W1[j]    <- rbinom(1,K1j,p.M)/K1j
      W2[j]    <- rbinom(1,K2j,p.M)/K2j
    }

    # Calculate our original U
    #-------------------------------------------------------------------------#
    Uj <- W1 - W2
    U  <- sum(Uj)/J

    # Permutations of the outcomes Wij to the 2J clusters
    #-------------------------------------------------------------------------#
    Us <- (permute %*% Uj)/J
#   Us <- randomization(Uj)/J # works only for J<=20)

    # Is U in the rejection zone
    #-------------------------------------------------------------------------#
    Usorted <- c(sort(Us, decreasing=TRUE))
    Ureject <- Usorted[reject]
    if (!(all(c(Uj,U)==0))) {
          count.perm <- (U-Ureject >= 0) + count.perm
    }

    # paired t-test
    #-------------------------------------------------------------------------#
      # paired t-statistic
      #-------------------------------------------------------------------------#
      if (all(c(Uj,U)==0)) {
          t <- 0
      } else {
          Var.t <- (J-1)^(-1) * sum((Uj-U)^2)
          t <- U * (Var.t / J)^(-1/2)
      }

      # Is t in the rejection zone according to distribution table?
      #-------------------------------------------------------------------------#
        ttest.statistic <- qt(1-alpha/2,J-1)
#       ttest.statistic <- qt(1-alpha,J-1)
        count.ttest <- (abs(t) - ttest.statistic > 0) + count.ttest

    # print a message every 1000 iterations
    #-------------------------------------------------------------------------#
    if ((z %% 1000)==0) {
      ShowStatus(paste("",z))
      ShowStatus(paste(" count=",count.perm))
    }
  } # end of simulation
  cat("\n")  # newline
```

```
  list(count.perm, count.ttest)
}


# Determines the rejection area for the permutation test
#=============================================================================#
Reject <- function(N,alpha) {
  n <- 2^N
  if (n > 100000) {
    reject <- 100000*alpha
  }
  else {
    reject <- floor(alpha*n)
  }
reject
}


# Define the variables
#=============================================================================#
  # Number of [pairs]
  #-----------------------------------------------------------------------#
  pairs <-14

  # Number of subjects[Kij] in the treatment(i=1) and control(i=2) groups
  #-----------------------------------------------------------------------#
  K1J <- 55
  K2J <- 55

  # [P]robability of [m]easurement. Based on SMART study data
  #-----------------------------------------------------------------------#
  P.m <- 0.9

  # Level for the t-test
  #-----------------------------------------------------------------------#
  alpha <- 0.05

  # Set the random seed (optional)
  #-----------------------------------------------------------------------#
  rs <- c(1, 764832617, 646225089)

  set.seed(rs)


SizeSim(numsimul=10000,pairs,K1j,K2j,P.m,alpha,permute=Permute(pairs),reject=Re
       ject(pairs,alpha))
```

A.5. Simulation program for meta-analysis techniques applied to CHAT

```
#=============================================================================#
#                                                                             #
# Inference based on meta-analytical techniques                               #
#  Applied to the CHAT Community Intervention Trial                           #
#                                                                             #
#  Based on Thompson et al. paper :                                           #
#   "The design and analysis of paired cluster randomized trials:             #
#    An application of meta-anlysis techniques"                               #
#                                                                             #
#  Theory for estimation of the variance of the true effect sizes             #
#  based on paper by Brockwell et al. :                                       #
#    "A comparison of statistical methods for meta-analysis                   #
#                                                                             #
#  Objective : Simulate the coverage probabilities resulting from             #
#              application of fixed and random effect meta-analysis           #
#              methods to CHAT data                                           #
#                                                                             #
#=============================================================================#
#                                                                             #
# Key variable definitions:                                                   #
#    J    number of pairs                                                     #
#    Kij  number of subjects in treatment i, pair j                           #
#    Xij  outcome in treatment i, pair j. A combination of the Yijk.     #
#    dj   treatment effect in pair j                                          #
#    wj   weights                                                             #
#                                                                             #
# Details/concepts:                                                           #
#    Measurability   What is the probability the subject has BP readings?  #
#    Measure         Does the subject have BP readings?  Y/N (1/0)           #
#    treatment.effect                                                         #
#      .Measure      How much did treatment affect the Measurability?       #
#                                                                             #
# Outcomes of interest:                                                       #
#    1) Difference in proportion of patients with BP reading in the           #
#       past 12 months                                                        #
#=============================================================================#


# For debugging purposes                                                      #
#=============================================================================#
InitShowStatus <- function() {
  showstatus.characters <<- 0
  showstatus.width <<- options()$width-5
}

ShowStatus <- function(message) {
  if (showstatus.characters > showstatus.width) {
    cat("\n")
       showstatus.characters <<-0
   }
   cat(message)
```

```
    flush.console()
    showstatus.characters <<- showstatus.characters+nchar(message)
}


# The Simul function calculates global treatment effects based on          #
# application of fixed and random meta-analytical techniques as well as    #
# corresponding coverage probabilities                                     #
#==========================================================================#
Simul <- function(Numsimul, p.M, K.1j, K.2j, J) {

  count.fixed <- count.MH <- count.m.MH <- count.random <- 0
  n.1j <- rep(K.1j, J)
  n.2j <- rep(K.2j, J)

  mX.1j <- mn.1j <- rep(NA,J) # [m]odified vectors
  mX.2j <- mn.2j <- rep(NA,J) # [m]odified vectors

  # Simulation                                                             #
  #-------------------------------------------------------------------------#
  for (z in 1:Numsimul) {
    # print a message every 1000 iterations
    if ((z %% 1000)==0) ShowStatus(paste("",z))

    # Create study data for each practice                                  #
    # [P]robabilities of [m]easurement are based on CHAT baseline data     #
    #-------------------------------------------------------------------------#
    X.1j <- rbinom(J,n.1j,p.M) # num.patients with BP in treatment practices
    X.2j <- rbinom(J,n.2j,p.M) # num.patients with BP in control practices

      # Continuity correction
      # For each index j, if there is a zero in any of the 4 cells of the  #
      # implicit 2 x 2 table then add 0.5 to all 4 cells                   #
      #-------------------------------------------------------------------------#
      zero <- X.1j==0 | X.2j==0 | X.1j==n.1j | X.2j==n.2j

      mX.1j[!zero] <- X.1j[!zero]
      mX.1j[zero]  <- X.1j[zero] + 0.5

      mn.1j[!zero] <- n.1j[!zero]
      mn.1j[zero]  <- n.1j[zero] + 1

      mX.2j[!zero] <- X.2j[!zero]
      mX.2j[zero]  <- X.2j[zero] + 0.5

      mn.2j[!zero] <- n.2j[!zero]
      mn.2j[zero]  <- n.2j[zero] + 1

    # proportion of BP readings in each practice
    #-------------------------------------------------------------------------#
    p.1j  <- X.1j/n.1j
    p.2j  <- X.2j/n.2j
    mp.1j <- mX.1j/mn.1j
    mp.2j <- mX.2j/mn.2j

    # treatment effect measured as difference in proportions
    #-------------------------------------------------------------------------#
    dj  <- p.1j - p.2j
    mdj <- mp.1j - mp.2j
```

```
# FIXED EFFECTS
#-------------------------------------------------------------------------#
sigma2 <- mp.1j*(1-mp.1j)/mn.1j + mp.2j*(1-mp.2j)/mn.2j
wj <- 1/sigma2

  # Overall effect (weighted average of the dj across strata)
  #-----------------------------------------------------------------------#
  d.w <- sum(wj*mdj)/sum(wj)

  # Cochran's Q  to measure heterogeneity
  #-----------------------------------------------------------------------#
  qw <- sum(wj*(mdj-d.w)^2)

  # Estimate of tau (Brockwell)  or 2*sigma2B (Thompson)
  #-----------------------------------------------------------------------#
  t <- (qw - (J-1))/(sum(wj) - (sum(wj^2))/sum(wj))
  tau2 <- t*(t>=0)

  # Risk Difference weights
  #-----------------------------------------------------------------------#
  wj.MH <- n.1j*n.2j/(n.1j+n.2j)
  d.w.MH <- sum(wj.MH*dj)/sum(wj.MH)
  a <- X.1j
  b <- n.1j - a
  c <- X.2j
  d <- n.2j - c
  var.d.w.MH <- sum((a*b*n.2j^3 + c*d*n.1j^3)/(n.1j*n.2j*(n.1j+n.2j)^2))/
                (sum(n.1j*n.2j/(n.1j+n.2j)))^2

    # with continuity correction
    #-----------------------------------------------------------------------#
    mwj.MH <- mn.1j*mn.2j/(mn.1j + mn.2j)
    md.w.MH <- sum(mwj.MH*mdj)/sum(mwj.MH)
    a <- mX.1j
    b <- mn.1j - a
    c <- mX.2j
    d <- mn.2j - c
    var.md.w.MH <- sum((a*b*mn.2j^3 +
   c*d*mn.1j^3)/(mn.1j*mn.2j*(mn.1j+mn.2j)^2))/
                   (sum(mn.1j*mn.2j/(mn.1j+mn.2j)))^2

  # CI with Mantel Haenszel methods
  #-----------------------------------------------------------------------#
  lo.MH <- d.w.MH - 1.96*sqrt(var.d.w.MH)
  hi.MH <- d.w.MH + 1.96*sqrt(var.d.w.MH)
  # with continuity correction
  lo.m.MH <- md.w.MH - 1.96*sqrt(var.md.w.MH)
  hi.m.MH <- md.w.MH + 1.96*sqrt(var.md.w.MH)

  # CI with inverse weights fixed effects
  #-----------------------------------------------------------------------#
  lo.fixed <- d.w - 1.96*sqrt(1/sum(wj))
  hi.fixed <- d.w + 1.96*sqrt(1/sum(wj))

# RANDOM EFFECTS
#-------------------------------------------------------------------------#
  # Estimate of var(dj)
  #-----------------------------------------------------------------------#
  var.dj <- tau2 + sigma2
```

```
      # weights (wt), estimator of dj (effect of intervention) and
      # variance of dj
      #-------------------------------------------------------------------#
      w <- var.dj^(-1)

      # estimator of dj
      #-------------------------------------------------------------------#
      dw <- sum(w*mdj)/sum(w)
      var.dw <- (sum(w))^(-1)

      # CI with random effects
      #-------------------------------------------------------------------#
      lo.random <- dw - 1.96*sqrt(var.dw)
      hi.random <- dw + 1.96*sqrt(var.dw)


    # CALCULATE COUNTS
    # If the CI doesn't contain 0, increase count
    #-------------------------------------------------------------------#
    count.MH <- (!(lo.MH < 0 && hi.MH >0)) + count.MH
    count.m.MH <- (!(lo.m.MH < 0 && hi.m.MH >0)) + count.m.MH
    count.fixed <- (!(lo.fixed < 0 && hi.fixed >0)) + count.fixed
    count.random <- (!(lo.random < 0 && hi.random >0)) + count.random

  } # end for z

list(count.MH, count.m.MH, count.fixed, count.random)
} # end simul


# Define the variables
  # Number of [pairs]
  #-------------------------------------------------------------------#
  pairs <-14

  # Number of subjects[Kij] in the treatment(i=1) and control(i=2) groups
  #-------------------------------------------------------------------#
  K1j <- 55
  K2j <- 55

  # [P]robability of [m]easurement. Based on SMART study data
  #-------------------------------------------------------------------#
  P.m <- 0.9

  # Level for the t-test
  #-------------------------------------------------------------------#
  alpha <- 0.05

  # Set the random seed (optional)
  #-------------------------------------------------------------------#
  rs <- c(1, 764832617, 646225089)

  set.seed(rs)

  A <- Simul(Numsimul=10000, P.m, K1j, K2j, pairs)
```

A.6. Program for application of permutation and meta-analysis to CHAT data

```
#=============================================================================#
# Apply permutation-based and meta-analysis techniques to the CHAT data
#=============================================================================#

# Set the random seed (optional)
#-----------------------------------------------------------------------------#
rs <- c(1,934715370, 1082990829)
set.seed(rs)

# Define the variables
  # Number of [pairs]
  #---------------------------------------------------------------------------#
   pairs <-14

  # Level for the rejection areas
  #---------------------------------------------------------------------------#
  alpha <- 0.05

# Read in the CHAT data and add practice identifier
#-----------------------------------------------------------------------------#
xx<-read.table("C:\\base.txt",sep="\t",header=T)
xx$practice<-sort(rep(1:28,55))[-1407]

# randomly pair the practices and assign to treatment or control
#-----------------------------------------------------------------------------#
x <- c(1:28)
a <- runif(28,0,1)
t <- x[a<median(a)]
c <- x[a>median(a)]

# Split xx into tt for treatment group, and cc for controls
#-----------------------------------------------------------------------------#
attach(xx)
tt <- rbind(xx[practice==t[1],],xx[practice==t[2],],
xx[practice==t[3],],xx[practice==t[4],],
xx[practice==t[5],],xx[practice==t[6],],
xx[practice==t[7],],xx[practice==t[8],],
xx[practice==t[9],],xx[practice==t[10],],
xx[practice==t[11],],xx[practice==t[12],],
xx[practice==t[13],],xx[practice==t[14],])

cc <- rbind(xx[practice==c[1],],xx[practice==c[2],],
xx[practice==c[3],],xx[practice==c[4],],
xx[practice==c[5],],xx[practice==c[6],],
xx[practice==c[7],],xx[practice==c[8],],
xx[practice==c[9],],xx[practice==c[10],],
xx[practice==c[11],],xx[practice==c[12],],
xx[practice==c[13],],xx[practice==c[14],])

detach(xx)
```

```
# Outcomes for the treatment group
#-------------------------------------------------------------------------#
attach(tt)

  # Number of patients with BP reading in past 12 months
  #-------------------------------------------------------------------------#
  BPrec <- abs(bprecany-1) # no BP=0, at leat 1 BP=1
  X.1j <- tapply(BPrec,practice,sum)

  # Number of patients in each practice
  #-------------------------------------------------------------------------#
  n.1j <- tapply(BPrec,practice,length)

  # Proportion in each practice with a BP reading in the past 12 months
  #-------------------------------------------------------------------------#
  p.1j <- tapply(BPrec,practice,mean)
  mean(p.1j)

detach(tt)

# Outcomes for the control group
#-------------------------------------------------------------------------#
attach(cc)

  # Number of patients with BP reading in past 12 months
  #-------------------------------------------------------------------------#
  BPrec <- abs(bprecany-1)
  X.2j <- tapply(BPrec,practice,sum)

  # Number of patients in each practice
  #-------------------------------------------------------------------------#
  n.2j <- tapply(BPrec,practice,length)

  # Proportion in each practice with a BP reading in the past 12 months
  #-------------------------------------------------------------------------#
  p.2j <- tapply(BPrec,practice,mean)
  mean(p.2j)

detach(cc)

# Treatment effect measured as difference in proportions
#-------------------------------------------------------------------------#
dj  <- p.1j - p.2j


#=========================================================================#
# Meta-analysis techniques
#=========================================================================#
  # FIXED EFFECTS
  #-------------------------------------------------------------------------#
  sigma2 <- p.1j*(1-p.1j)/n.1j + p.2j*(1-p.2j)/n.2j
  wj <- 1/sigma2

    # Overall effect (weighted average of the dj across strata)
    #-------------------------------------------------------------------------#
    d.w <- sum(wj*dj)/sum(wj)

    # Cochran's Q  to measure heterogeneity
    #-------------------------------------------------------------------------#
    qw <- sum(wj*(dj-d.w)^2)
```

```
    # Estimate of tau (Brockwell)  or 2*sigma2B (Thompson)
    #-------------------------------------------------------------------------#
    t <- (qw - (pairs-1))/(sum(wj) - (sum(wj^2))/sum(wj))
    tau2 <- t*(t>=0)

    # Risk Difference weights
    #-------------------------------------------------------------------------#
    wj.MH <- n.1j*n.2j/(n.1j+n.2j)
    d.w.MH <- sum(wj.MH*dj)/sum(wj.MH)
    a <- X.1j
    b <- n.1j - a
    c <- X.2j
    d <- n.2j - c
    var.d.w.MH <- sum((a*b*n.2j^3 + c*d*n.1j^3)/(n.1j*n.2j*(n.1j+n.2j)^2))/
                  (sum(n.1j*n.2j/(n.1j+n.2j)))^2

    # CI with Mantel Haenszel methods
    #-------------------------------------------------------------------------#
    lo.MH <- d.w.MH - 1.96*sqrt(var.d.w.MH)
    hi.MH <- d.w.MH + 1.96*sqrt(var.d.w.MH)
    lo.MH;hi.MH

    # CI with only fixed effects
    #-------------------------------------------------------------------------#
    lo.fixed <- d.w - 1.96*sqrt(1/sum(wj))
    hi.fixed <- d.w + 1.96*sqrt(1/sum(wj))
    lo.fixed;hi.fixed

  # RANDOM EFFECTS
  #-------------------------------------------------------------------------#
    # Estimate of var(dj)
    #-------------------------------------------------------------------------#
      var.dj <- tau2 + sigma2

    # weights (wt), estimator of dj (effect of intervention) and
    # variance of dj
    #-------------------------------------------------------------------------#
    w <- var.dj^(-1)

    # estimator of dj
    #-------------------------------------------------------------------------#
    dw <- sum(w*dj)/sum(w)
    var.dw <- (sum(w))^(-1)

    # CI with random effects
    #-------------------------------------------------------------------------#
    lo.random <- dw - 1.96*sqrt(var.dw)
    hi.random <- dw + 1.96*sqrt(var.dw)
    lo.random;hi.random


  # DETERMINE WHETHER CI CONTAINS 0. IF so, reject H0
  #-------------------------------------------------------------------------#
    MH <- (!(lo.MH < 0 && hi.MH >0))
    fixed <- (!(lo.fixed < 0 && hi.fixed >0))
    random <- (!(lo.random < 0 && hi.random >0))


#===========================================================================#
```

```
# Permutation-based inference
#=============================================================================#

# Reset the random seed (optional)
#-----------------------------------------------------------------------------#
rs <- c(1,934715370, 1082990829)
set.seed(rs)

  # Function to permute the outcomes
  #-----------------------------------------------------------------------------#
  # Constructs a full matrix of all possibilities if 2^J < 100,000,          #
  # else creates a matrix of 100000 randomly generated possibilites           #
  #-----------------------------------------------------------------------------#
  Permute <- function(k) {
    if (k==0) {
      NULL
    }
    else {
      if (2^k < 100000) {
        rbind(cbind(1,Recall(k-1)),cbind(-1,Recall(k-1)))
      }
      else {
        A <- matrix(0,100000,k)
          for (reps in 1:100000) {
            A[reps,] <- sample(c(1,-1),k,replace=T,prob=c(0.5,0.5))
          }
      A
      }
    }
  }

  # Function to determine the rejection criteria for U                        #
  #-----------------------------------------------------------------------------#

  #  One sided rejection area                                                 #
  #-----------------------------------------------------------------------------#
  Reject1 <- function(J,alpha) {
    n <- 2^J
    if (n > 100000) {
      reject <- 100000*alpha
    }
    else {
      reject <- floor(alpha*n)
    }
  reject
  }

  #  Two sided rejection area                                                 #
  #-----------------------------------------------------------------------------#
  Reject2 <- function(J,alpha) {
    n <- 2^J
    if (n > 100000) {
      reject <- c(floor(100000*0.5*alpha),ceiling(100000*(1-0.5*alpha)))
    }
    else {
      reject <- c(floor(0.5*alpha*n),ceiling((1-0.5*alpha)*n))
    }
  reject
  }
```

```
  # Calculate our original U
  #-------------------------------------------------------------------------#
    U  <- sum(dj)/pairs
    U

    # Permutations of the outcomes Wij to the 2J clusters
    #-------------------------------------------------------------------------#
    Us <- (Permute(pairs) %*% dj)/pairs

    # Permutation test. Is U in the rejection zone?
    #-------------------------------------------------------------------------#
    reject <- Reject1(pairs, alpha)
    Usorted <- c(sort(Us, decreasing=TRUE))
    Ureject <- Usorted[reject]
    if (!(all(c(dj,U)==0))) {
      REJECT.perm <- (U-Ureject >= 0)
    }
    REJECT.perm

    # paired t-test
      # paired t-statistic
      #-----------------------------------------------------------------------#
      if (all(c(dj,U)==0)) {
          t <- 0
      } else {
          Var.t <- (pairs-1)^(-1) * sum((dj-U)^2)
          t <- U * (Var.t / pairs)^(-1/2)
      }

      # Is t in the rejection zone according to distribution table?
      #-----------------------------------------------------------------------#
        ttest.statistic <- qt(1-alpha/2,pairs-1)
#       ttest.statistic <- qt(1-alpha,pairs-1)
        REJECT.ttest <- (abs(t) - ttest.statistic > 0)
        REJECT.ttest
```

A.7. Simulation program for the PDA study, competitive recruitment between centres

```
#=============================================================================#
#                                                                             #
# Simulate the imbalance that can occur in treatment allocation when          #
# a blocked design at the recruiter level is used instead of a                #
# blocked design at the centre level.                                        #
#                                                                             #
# Use recruitment proportions from Astronomer data to simulate                #
#  allocation from competitive recruitment among centres.                    #
#                                                                             #
# "Simulation" performs the simulation                                        #
#                                                                             #
# "Results" produces tabular output of simulation results                     #
#                                                                             #
# "Format.table" sorts the table according to specifications                  #
#                                                                             #
#=============================================================================#
#                                                                             #
# Key variable definitions:                                                   #
#    ncentres    [n]umber of [centres]                                        #
#    nrecruiters [n]umber of [recruiters] per centre                          #
#    nrecruits   [n]umber of [recruits] per centre                            #
#    block.type  Can be fixed (4) or random (4,6).                           #
#    l.serious   [l]ower bound for serious imbalance                          #
#    u.serious   [u]pper bound for serious imbalance                          #
#=============================================================================#

# For debugging purposes                                                      #
#=============================================================================#
InitShowStatus <- function() {
  showstatus.characters <<- 0
  showstatus.width <<- options()$width-5
}

ShowStatus <- function(message) {
  if (showstatus.characters > showstatus.width) {
    cat("\n")
       showstatus.characters <<-0
   }
   cat(message)
   flush.console()
   showstatus.characters <<- showstatus.characters+nchar(message)
}


# Sum.n.M function randomly assigns N objects to n categories.               #
# This will provide us with the number of recruits/recruiter                  #
#=============================================================================#
Sum.n.N <- function(n, N) {
  count <- rep(NA,n)
  x <- sample(c(1:n),N,replace=TRUE)

  for (i in 1:n) {
```

```
    count[i] <- sum(x==i)
  }
count
}

# Skewed.Sum.n.M function assigns N objects to n categories based on Astronomer
# proportions of recruitment. This is used to provide us with the number of
# recruits/centre
#=============================================================================#
Skewed.Sum.n.N <- function(n, N) {
  count <- rep(NA,n)
  x <- sample(c(1:n),N,replace=TRUE,
              prob=c(rep(0.95,ceiling(0.1*n)),
                     rep(0.001,ceiling(0.28*n)),
                     rep(0.2,(n-(ceiling(0.1*n)+ceiling(0.3*n))))))

  for (i in 1:n) {
    count[i] <- sum(x==i)
  }
count
}

  # Test that Skewed.Sum.n.N gives proportions of recruitment similar to the
  # Astronomer data : (15,12,0,0,0,0,0,0,5,5,5,3,3,3,2,2,2,2,1)
    test <- matrix(NA,nrow=10000,ncol=19)
    for (j in 1:10000) {
      test[j,] <- Skewed.Sum.n.N(19,60)
    }
    apply(test,2,mean)
#   [1] 13.8549 13.8325  0.0161  0.0143  0.0154  0.0147  0.0140  0.0149  2.9478
# [10]  2.9120  2.9284  2.9188  2.9262  2.9331  2.9510  2.9059  2.9190  2.9539
# [19]  2.9271



# Simulation 1) randomly selects number of recruits per centre                #
#            2) randomly creates block patterns for each recruiter            #
#            3) randomly selects the number of recruits for each recruiter    #
#            4) measures the proportion of allocation to treatment A (=! B)   #
#=============================================================================#
Simulation <- function(nsimul, block.type, ncentres, nrecruiters, n.recruits) {

  # Initializing of the output data matrix                                    #
  #---------------------------------------------------------------------------#
  Simul.data <- matrix(NA,nrow=nsimul,ncol=3)

  for (z in 1:nsimul) {
    # Randomly assign a number of recruited patients to each centre           #
    #-------------------------------------------------------------------------#
    if (ncentres > 1) {
      nrecruits <- Skewed.Sum.n.N(n=ncentres,N=n.recruits)}
    else nrecruits <- n.recruits

    # treatments assigned to the nrecruits for each centre(rows)              #
    #-------------------------------------------------------------------------#
    recruits.centre <- matrix(NA,nrow=ncentres,ncol=max(nrecruits))

    # number of A(col1), B(col2) and total(col3) for each centre(rows)        #
    #-------------------------------------------------------------------------#
```

```
    recruited.centre <- matrix(NA,nrow=ncentres,ncol=3)

    # Each centre
    for (i in 1:ncentres) {
      if (nrecruits[i] == 0)
         recruits.centre[i,] <- rep(0,max(nrecruits))
      else {

      blocks.recruiters <- matrix(NA,nrow=nrecruits[i],ncol=nrecruiters)

      # For each recruiter, create a sequence of randomized blocks
      # Allow for the possibility that one recruiter did all the work
      #-------------------------------------------------------------------------#
      for (j in 1:nrecruiters) {

        # [length] of the blocked sequence
        length <- 0
        # number of blocks created
        nblocks <- 0

        while (nrecruits[i] - length > 0) {
          if (block.type == "fixed")
            block.size <- 4
          if (block.type == "unblocked")
            block.size <- 1
          if (block.type == "random")
            block.size <- sample(c(4,6),1,replace=T,prob=c(0.5,0.5))

          length <- block.size + length
          nblocks <- nblocks + 1

          if (nblocks == 1)
            blocks.recruiter <- sample(c(rep(0,ceiling(block.size/2)),
                                         rep(1,ceiling(block.size/2))),
                                       block.size,replace=F)
          else
            blocks.recruiter <-
c(blocks.recruiter,sample(c(rep(0,ceiling(block.size/2)),

rep(1,ceiling(block.size/2))),block.size,replace=F))

        # Have to truncate if larger than nrecruits                          #
        #-------------------------------------------------------------------#
        blocks.recruiters[,j] <- blocks.recruiter[1:nrecruits[i]]
        }  # end while

      }  # end j for nrecruiters


      # Randomly assign a number of recruited patients to each recruiter,    #
      # based on the Astronomer centre recruitement proportions              #
      #-------------------------------------------------------------------#
      randomized.patients <- Sum.n.N(n=nrecruiters,N=nrecruits[i])

      # Based on the random assignment above, select the appropriate units   #
      #  for each recruiter                                                  #
      #-------------------------------------------------------------------#
      a <- 0
      for (j in 1:nrecruiters) {
        if (randomized.patients[j] > 0) {
```

```
          a <- a + 1
          if (a==1)
            recruited <- blocks.recruiters[1:randomized.patients[j],j]
          else
            recruited <-
c(recruited,blocks.recruiters[1:randomized.patients[j],j])
        }
      }

      recruits.centre[i,] <- c(recruited,rep(0,max(nrecruits)-nrecruits[i]))
      }

      # Determine allocation to treatment A and B in each centre             #
      #-------------------------------------------------------------------#
      recruited.centre[i,2] <- sum(recruits.centre[i,])           # number of B
      recruited.centre[i,1] <- nrecruits[i]-recruited.centre[i,2] # number of A
      recruited.centre[i,3] <- sum(recruited.centre[i,1:2])

    } # end i for centres

    # matrix with 3 columns, col1:number of A, col2:number of B, col3:total  #
    #  one row per simulation. Did this to calculate variances and range    #
    #-------------------------------------------------------------------#
    recruited.all.centres <- cbind(sum(recruited.centre[,1]),
                                   sum(recruited.centre[,2]),
                                   sum(recruited.centre[,3]))

    Simul.data[z,] <- recruited.all.centres
  } # end nsimul
Simul.data
}  ## end Simulation


# Results produces tabular output of simulation results                      #
# Returns settings and tables for graphical functions                        #
#============================================================================#
Results <- function(NSIMUL, BLOCK.TYPE, l.serious, u.serious) {

  # Define settings                                                          #
  #-------------------------------------------------------------------#
    N.centres <- c(1,5,15,25)               # Number of centres
    n1 <- length(N.centres)

    N.recruiters <- c(1,2,3,4)              # Number of recruiters per centre
    n2 <- length(N.recruiters)

    N.recruits <- c(30,60,90,120,150)       # Number of recruits per centre
    n3 <- length(N.recruits)

  settings <- expand.grid(N.centres,N.recruiters,N.recruits)

  # Run simulation for each combination of settings                          #
  #-------------------------------------------------------------------#
  pcnt<-pcnt.serious<-rep(NA,n1*n2*n3)

  for (i in 1:(n1*n2*n3))  {
    Simul.data <-
Simulation(NSIMUL,BLOCK.TYPE,settings[i,1],settings[i,2],settings[i,3])
```

```
    # Measure allocation imbalance based on pre-defined limits                #
    #-------------------------------------------------------------------------#
    pcnt.simul.A <- Simul.data[,1]/Simul.data[,3] #  Proportion of A's for each
                                                     simulation(rows)
    pcnt.simul.B <- Simul.data[,2]/Simul.data[,3] #  Proportion of B's for each
                                                     simulation(rows)
    pcnt[i] <- sum(Simul.data[,1])/sum(Simul.data[,3]) # proportion of A's
                                                     across all simulations

    #  proportion of serious imbalance across all NSIMUL simulations          #
    pcnt.serious[i] <-
                  mean(apply(cbind(pcnt.simul.A,pcnt.simul.B),1,max)>u.serious)
  }

  pcnt <- cbind(settings,pcnt)
  pcnt.serious <- cbind(settings,pcnt.serious)

  list(NSIMUL, BLOCK.TYPE, N.centres, N.recruiters, N.recruits, #1-5
       u.serious,l.serious,                                     #6,7
       pcnt,                                                    #8
       pcnt.serious,                                            #9
      )
}

X.fixed.55 <- Results(1000,"fixed",0.45,0.55)
X.fixed.60 <- Results(1000,"fixed",0.40,0.60)
X.fixed.70 <- Results(1000,"fixed",0.30,0.70)
X.random.55 <- Results(1000,"random",0.45,0.55)
X.random.60 <- Results(1000,"random",0.40,0.60)
X.random.70 <- Results(1000,"random",0.30,0.70)


# Sorts the Table according to specifications for graphing                    #
#-----------------------------------------------------------------------------#
Format.tables <- function(values,z,...) {
  column.values <- sort(unique(z))
  o <- order(...)
  m <- matrix(values[o],ncol=length(column.values),byrow=TRUE,
    dimnames=list(NULL,column.values))
  left <- unique(cbind(...)[o,])
  cbind(left,m)
}


# Put table in proper format                                                  #
#-----------------------------------------------------------------------------#
Format.tables(X.fixed.55[[9]][,4],X.fixed.55[[9]][,3],X.fixed.55[[9]][,1],X.fix
ed.55[[9]][,2])
Format.tables(X.fixed.60[[9]][,4],X.fixed.60[[9]][,3],X.fixed.60[[9]][,1],X.fix
ed.60[[9]][,2])
Format.tables(X.fixed.70[[9]][,4],X.fixed.70[[9]][,3],X.fixed.70[[9]][,1],X.fix
ed.70[[9]][,2])

Format.tables(X.random.55[[9]][,4],X.random.55[[9]][,3],X.random.55[[9]][,1],X.
random.55[[9]][,2])
Format.tables(X.random.60[[9]][,4],X.random.60[[9]][,3],X.random.60[[9]][,1],X.
random.60[[9]][,2])
Format.tables(X.random.70[[9]][,4],X.random.70[[9]][,3],X.random.70[[9]][,1],X.
random.70[[9]][,2])
```

A.8. Simulations of size and complete imbalance based on Astronomer recruitment pattern

```
#=============================================================================#
#                                                                             #
# Based on Astronomer data. Using competitive recruitement between centres. #
# Simulate the imbalance that can occur in treatment allocation when          #
# a blocked design at the recruiter level is used instead of a                #
# blocked design at the centre level.                                        #
#                                                                             #
#=============================================================================#
#                                                                             #
# Key variable definitions:                                                   #
#    ncentres     [n]umber of [centres]                                       #
#    nrecruiters [n]umber of [recruiters] per centre                          #
#    nrecruits    [n]umber of [recruits] per centre                           #
#    block.size   [size] of randomization [block]s. Can be fixed or random.  #
#    block.size.method [Method] of selecting [block.size] = fixed or random   #
#    nblocks      [n]umber of randomization [blocks]                          #
#                  if block.size is fixed, nblocks= nrecruits/block.size      #
#                  else if random (4,6), nblocks in                           #
#                                        ceiling[nrecruits/6,nrecruits/4]     #
#                                                                             #
#=============================================================================#

# For debugging purposes                                                      #
#=============================================================================#
InitShowStatus <- function() {
  showstatus.characters <<- 0
  showstatus.width <<- options()$width-5
}

ShowStatus <- function(message) {
  if (showstatus.characters > showstatus.width) {
    cat("\n")
       showstatus.characters <<-0
   }
   cat(message)
   flush.console()
   showstatus.characters <<- showstatus.characters+nchar(message)
}


# Sum.n.M function randomly assigns N objects to n categories.               #
# This will provide us with the number of recruits/recruiter                  #
#=============================================================================#
Sum.n.N <- function(n, N) {
  count <- rep(NA,n)
  x <- sample(c(1:n),N,replace=TRUE)

  for (i in 1:n) {
    count[i] <- sum(x==i)
  }
count
```

```
}

# Simulation 1)randomly creates block patterns for each recruiter            #
#            2)randomly selects the number of recruits for each recruiter    #
#            3)measures the proportion of allocation to treatment A (=! B)   #
#=============================================================================#

Simulation <- function(nsimul, block.type, ncentres, nrecruiters, nrecruits) {

  # Initializing of the output data matrix                                   #
  #---------------------------------------------------------------------------#
  Simul.data <- complete.imbalance <- matrix(NA,nrow=nsimul,ncol=3)

  for (z in 1:nsimul) {

    # treatments assigned to the nrecruits for each centre(rows)             #
    #---------------------------------------------------------------------------#
    recruits.centre <- matrix(NA,nrow=ncentres,ncol=max(nrecruits))

    # number of A(col1), B(col2) and total(col3) for each centre(rows)       #
    #---------------------------------------------------------------------------#
    recruited.centre <- matrix(NA,nrow=ncentres,ncol=3)

    #Each centre
    for (i in 1:ncentres) {
      if (nrecruits[i] == 0)
         recruits.centre[i,] <- rep(0,max(nrecruits))
      else {

      blocks.recruiters <- matrix(NA,nrow=nrecruits[i],ncol=nrecruiters)

      # For each recruiter, create a sequence of randomized blocks
      # Allow for the possibility that one recruiter did all the work
      #-------------------------------------------------------------------------#
      for (j in 1:nrecruiters) {

        # [length] of the blocked sequence
        length <- 0
        # number of blocks created
        nblocks <- 0

        while (nrecruits[i] - length > 0) {
          if (block.type == "fixed")
            block.size <- 4
          if (block.type == "unblocked")
            block.size <- 1
          if (block.type == "random")
            block.size <- sample(c(4,6),1,replace=T,prob=c(0.5,0.5))

          length <- block.size + length
          nblocks <- nblocks + 1

          if (nblocks == 1)
            blocks.recruiter <- sample(c(rep(0,ceiling(block.size/2)),
                                         rep(1,ceiling(block.size/2))),
                                       block.size,replace=F)
          else
            blocks.recruiter <-
c(blocks.recruiter,sample(c(rep(0,ceiling(block.size/2)),
```

```
rep(1,ceiling(block.size/2))),block.size,replace=F))

        # Have to truncate if larger than nrecruits                          #
        #-------------------------------------------------------------------#
        blocks.recruiters[,j] <- blocks.recruiter[1:nrecruits[i]]
        }  # end while

      }  # end j for nrecruiters


      # Randomly assign a number of recruited patients to each recruiter,     #
      # based on the Astronomer centre recruitement proportions               #
      #-------------------------------------------------------------------#
      randomized.patients <- Sum.n.N(n=nrecruiters,N=nrecruits[i])

      # Based on the random assignment above, select the appropriate units    #
      #  for each recruiter                                                   #
      #-------------------------------------------------------------------#
      a <- 0
      for (j in 1:nrecruiters) {
        if (randomized.patients[j] > 0) {
          a <- a + 1
          if (a==1)
            recruited <- blocks.recruiters[1:randomized.patients[j],j]
          else
            recruited <-
c(recruited,blocks.recruiters[1:randomized.patients[j],j])
        }
      }

      recruits.centre[i,] <- c(recruited,rep(0,max(nrecruits)-nrecruits[i]))
      }

      # Determine allocation to treatment A and B in each centre
#
      #-------------------------------------------------------------------
#
      recruited.centre[i,2] <- sum(recruits.centre[i,])           # number of B
      recruited.centre[i,1] <- nrecruits[i]-recruited.centre[i,2] # number of A
      recruited.centre[i,3] <- sum(recruited.centre[i,1:2])

    } # end i for centres

    # Of the centres with participants, what proportion had 100% imbalance
    #   i.e. either treatment of control arm had no participants
    #-------------------------------------------------------------------#
      # Keep only the centres where there are participants
      tmp <- recruited.centre[!recruited.centre[,3]==0,byrow=T]
      denom <- nrow(tmp)
      num <- nrow(matrix(tmp[tmp[,1]==0,byrow=T],ncol=3))+
             nrow(matrix(tmp[tmp[,2]==0,byrow=T],ncol=3))
      complete.imbalance[z,] <- c(num,denom,num/denom)


    # matrix with 3 columns, col1:number of A, col2:number of B, col3:total   #
    #  one row per simulation. Did this to calculate variances and range     #
    #-------------------------------------------------------------------#
    recruited.all.centres <- cbind(sum(recruited.centre[,1]),
                                   sum(recruited.centre[,2]),
```

```
                                    sum(recruited.centre[,3]))

    Simul.data[z,] <- recruited.all.centres
  } # end nsimul
#recruited.centre
list(Simul.data,complete.imbalance)
}  ## end Simulation

N.recruits <- c(15,3,3,5,5,2,2,0,0,0,0,3,0,12,1,2,2,0,5)
Simulation(nsimul=10,block.type="fixed",ncentres=length(N.recruits),
           nrecruiters=3,N.recruits)


# Results produces tabular output of simulation results                    #
# Returns settings and tables for graphical functions                      #
#==========================================================================#
Results <- function(NSIMUL, BLOCK.TYPE, l.serious, u.serious) {

  # Define settings                                                        #
  #------------------------------------------------------------------------#
  # Number of recruits per centre
    Astronomer <- c(15,3,3,5,5,2,2,0,0,0,0,3,0,12,1,2,2,0,5)
    N.recruits <- Astronomer

    N.centres <- length(N.recruits)    # Number of centres
    n1 <- length(N.centres)

    N.recruiters <- c(1,2,3,4,8)       # Number of recruiters per centre
    n2 <- length(N.recruiters)

    settings <- expand.grid(N.centres,N.recruiters)

  # Run simulation for each combination of settings                        #
  #------------------------------------------------------------------------#
  pcnt <- pcnt.serious <- complete.imbalance <- rep(NA,n1*n2)

  for (i in 1:(n1*n2))  {
    Sim <- Simulation(NSIMUL,BLOCK.TYPE,settings[i,1],settings[i,2],N.recruits)
    Simul.data <- Sim[[1]]
    c.imbalance <- Sim[[2]]

    # Measure allocation imbalance based on pre-defined limits             #
    #----------------------------------------------------------------------#
    pcnt.simul.A <- Simul.data[,1]/Simul.data[,3] #  Proportion of A's for each
                                                     simulation(rows)
    pcnt.simul.B <- Simul.data[,2]/Simul.data[,3] #  Proportion of B's for each
                                                     simulation(rows)
    pcnt[i] <- sum(Simul.data[,1])/sum(Simul.data[,3]) # proportion of A's
                                                     across all simulations
    complete.imbalance[i] <- sum(c.imbalance[,1])/sum(c.imbalance[,2]) #
                                                     Proportion of complete
                                                     imbalance

    #  proportion of serious imbalance across all NSIMUL simulations      #
    #----------------------------------------------------------------------#
    tmp <- cbind(pcnt.simul.A,pcnt.simul.B)
    pcnt.serious[i] <-
              mean(apply(cbind(pcnt.simul.A,pcnt.simul.B),1,max)>u.serious)

  }
```

```
  pcnt <- cbind(settings,pcnt)
  pcnt.serious <- cbind(settings,pcnt.serious)
  complete.imbalance <- cbind(settings,complete.imbalance)

  list(NSIMUL, BLOCK.TYPE, N.centres, N.recruiters, N.recruits, #1-5
       u.serious,l.serious,                                     #6,7
       pcnt,                                                    #8
       pcnt.serious,                                            #9
       complete.imbalance                                       #10
      )
}

X.fixed.55 <- Results(10000,"fixed",0.45,0.55)
X.fixed.60 <- Results(10000,"fixed",0.40,0.60)
X.fixed.70 <- Results(10000,"fixed",0.30,0.70)

table.fixed <- cbind(X.fixed.55[[9]],X.fixed.60[[9]][,3],X.fixed.70[[9]][,3])
names(table.fixed) <- c("Centres","Palms","0.55", "0.60", "0.70")

X.random.55 <- Results(10000,"random",0.45,0.55)
X.random.60 <- Results(10000,"random",0.40,0.60)
X.random.70 <- Results(10000,"random",0.30,0.70)

table.random<-cbind(X.random.55[[9]],X.random.60[[9]][,3],X.random.70[[9]][,3])
names(table.random) <- c("Centres","Palms","0.55", "0.60", "0.70")


# Compare with SRS sampling imbalance
#-------------------------------------------------------------------------#
  #Probability of imbalance for simple randomization
  P.imbalance <- function(r,n) {
    2 * pnorm(-2*(r-0.5)*sqrt(n))
  }

  P.imbalance(0.55,60)
[1] 0.438578

  P.imbalance(0.60,60)
[1] 0.1213353

  P.imbalance(0.70,60)
[1] 0.001945774
```

Bibliography

Reference List

(1) Murray DM. Design and analysis of group-randomized trials. New York: Oxford University Press; 1998.

(2) Last JM. A Dictionary of Epidemiology. 4th ed. New York: Oxford; 2001.

(3) Abel U, Koch A. The role of randomization in clinical studies: myths and beliefs. J Clin Epidemiol 1999 Jun;52(6):487-97.

(4) Klar N, Donner A. The merits of matching in community intervention trials: a cautionary tale. Stat Med 1997 Aug 15;16(15):1753-64.

(5) Moulton L. Covariate-based constrained randomization of group-randomized trials. Clin Trials 2004;1:297-305.

(6) Kernan WN, Viscoli CM, Makuch RW, Brass LM, Horwitz RI. Stratified randomization for clinical trials. J Clin Epidemiol 1999 Jan;52(1):19-26.

(7) Hallstrom A, Davis K. Imbalance in treatment assignments in stratified blocked randomization. Control Clin Trials 1988 Dec;9(4):375-82.

(8) Feng Z, Diehr P, Peterson A, McLerran D. Selected statistical issues in group randomized trials. Annu Rev Public Health 2001;22:167-87.

(9) Gail MH, Mark SD, Carroll RJ, Green SB, Pee D. On design considerations and randomization-based inference for community intervention trials. Stat Med 1996 Jun 15;15(11):1069-92.

(10) Donner A, Klar N. Design and Analysis of Cluster Randomiation Trials in Health Research. London: Arnold; 2000.

(11) Donner A, Klar N. Pitfalls of and controversies in cluster randomization trials. Am J Public Health 2004 Mar;94(3):416-22.

(12) Schulz KF, Grimes DA. Unequal group sizes in randomised trials: guarding against guessing. Lancet 2002 Mar 16;359:966-70.

(13) Ludbrook J, Dudley H. Why Permutation Tests are Superior to *t* and *F* Tests in Biomedical Research. Am Stat 1998;52(2):127-32.

(14) Edgington ES. Randomization Tests. 3rd ed. New York: Marcel Dekker; 1995.

(15) Ludbrook J. Advantages of permutation (randomization) tests in clinical and experimental pharmacology and physiology. Clin Exp Pharmacol Physiol 1994 Sep;21(9):673-86.

(16) Manly BFJ. Randomization, Bootstrap and Monte Carlo Methods in Biology. 2nd ed. London: Chapman & Hall; 1997.

(17) Thompson SG, Pyke SD, Hardy RJ. The design and analysis of paired cluster randomized trials: an application of meta-analysis techniques. Stat Med 1997 Sep 30;16(18):2063-79.

(18) Chambers LW, Kaczorowski J, Levitt C, Karwalajtys T, McDonough B, Lewis J. Blood pressure self-monitoring in pharmacies. Building on existing resources. Can Fam Physician 2002 Oct;48:1594.

(19) Gail MH, Byar DP, Pechacek TF, Corle DK. Aspects of statistical design for the Community Intervention Trial for Smoking Cessation (COMMIT). Control Clin Trials 1992 Feb;13(1):6-21.

(20) Community intervention trial for smoking cessation (COMMIT): II. Changes in adult cigarette smoking prevalence. Am J Public Health 1995 Feb;85(2):193-200.

(21) Community Intervention Trial for Smoking Cessation (COMMIT): I. cohort results from a four-year community intervention. Am J Public Health 1995 Feb;85(2):183-92.

(22) Pitman EJG. Significance tests which may be applied to samples from any population. III. The analysis of variance tests. Biometrika 1938;29:322-35.

(23) Efron B. Student's t-test under symmetry conditions. JASA 1969;64:1278-302.

(24) DerSimonian R, Laird N. Meta-analysis in clinical trials. Control Clin Trials 1986 Sep;7(3):177-88.

(25) Deeks J, Altman D, Bradburn M. Statistical methods for examining heterogeneity and combining results from several studies in meta-analysis. In: Egger M, Davey Smith G, Altman D, editors. Systematic Reviews in Health Care: Meta-Analysis in Context. 2nd ed. London: BMJ Publication Group; 2001. p. 285-311.

(26) Sweeting J, Sutton J, Lambert C. What to add to nothing? Use and avoidance of continuity corrections in meta-analysis of sparse data. Stat Med 2004 May 15;23(9):1351-75.

(27) Martin S. Computer use by Canada's physicians approaches 90% mark. Can Med Assoc J 2001;(165):632.

(28) Martin S. MD's computer, PDA use on the upswing. Can Med Assoc J 2002;(167):794.

(29) Martin S. More than half of MD's under age 35 now using PDA's. Can Med Assoc J 2003;9(169):952.

(30) Friedman LM, Furberg CD, DeMets DL. Fundamentals of Clinical Trials. New York: Springer; 1988.

(31) Lachin JM, Matts JP, Wei LJ. Randomization in clinical trials: conclusions and recommendations. Control Clin Trials 1988 Dec;9(4):365-74.

(32) Fleiss JL. The Design and Analysis of Clinical Experiments. New York: Wiley; 1986.

(33) Schulz KF, Grimes DA. Generation of allocation sequences in randomised trials: chance, not choice. Lancet 2002 Feb 9;359(9305):515-9.

(34) Matts JP, Lachin JM. Properties of permuted-block randomization in clinical trials. Control Clin Trials 1988 Dec;9(4):327-44.

(35) Lachin JM. Statistical properties of randomization in clinical trials. Control Clin Trials 1988 Dec;9(4):289-311.

(36) Geller NL, Pocock SJ. Interim analyses in randomized clinical trials: ramifications and guidelines for practitioners. Biometrics 1987 Mar;43(1):213-23.

(37) Jennison C, Turnbull B. Group sequential methods to clinical trials. Chapman & Hall/CRC Press; 1999.

(38) Piantadosi S. Clinical Trials: A Methodologic Perspective. New York: Wiley; 1997.

(39) Lachin JM. Properties of simple randomization in clinical trials. Control Clin Trials 1988 Dec;9(4):312-26.