Combinability in meta-analysis of multi-treatment observational studies

Assessing and minimising covariate imbalance

A. L. Termaat

Delft University of Technology

Combinability in meta-analysis of multi-treatment observational studies

Assessing and minimising covariate imbalance

by

A. L. Termaat

Supervisor:	M. Vittorietti
Graduation Committee:	M. Vittorietti & G.F. Nane
Project Duration:	April 2025 - June 2025
Faculty:	EEMCS, Delft

Cover: AI-generated image by ChatGPT with prompt: “A modern, abstract 3D line chart with 3 smooth, wavy curves in different colors (blue, red, orange). The lines are rendered as glossy, translucent ribbons, appearing to float in a minimalistic light gray environment. Thin vertical lines connect the curves to the base, giving a sense of depth and structure. The overall mood is clean, futuristic, and elegant, with soft lighting and subtle shadows. Each line should be one color and not go over in another. Moreover, the aspect ratio should be 2:3 portrait.”

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Laymen's summary

Meta-analysis is a powerful method to combine multiple independent studies answering a common research question. To determine a causal treatment effect in a meta-analysis it is important to look at the characteristics of the participants, like age or diseases, also known as covariates. This is especially important in observational studies, where the covariate is not guaranteed to be balanced between treatment groups. This makes it difficult to determine a causal effect of the treatment, since the covariate may affect the outcome. In this thesis, the covariate imbalance between more than two treatment groups is assessed using so-called multi-sample test statistics, from which it is determined whether the imbalance is too large to consider the studies combinable. If this imbalance is too large, a balancing procedure is proposed to make the studies more combinable for meta-analysis. This is done by discarding some treatment groups that particularly hinder combinability. In simulated datasets, the result is an improvement of the covariate imbalance by discarding groups.

Summary

Meta-analysis is a powerful method to combine the treatment effects of multiple independent studies answering a common research question. To determine a causal treatment effect in a meta-analysis it is important to look at the characteristics of the participants, like age or diseases, also known as covariates. In the case of experimental studies, participants are randomly allocated to treatment groups. As a result, the covariate is in expectation equally distributed between the treatment groups. In observational studies, no randomisation has occurred and thus, the covariate imbalance between treatment groups may be more profound. This makes it difficult to determine a causal effect of the treatment, since the covariate may affect the outcome. Therefore, it is important to balance these covariates between the treatment, especially for observational studies. The condition when this balance is present is called combinability.

In this thesis, the covariate imbalance between treatment groups is assessed using five multi-sample test statistics. These assessment methods are based on the comparison of the empirical cumulative distribution functions of the covariate between the meta-arms, which are the collections of the similar treatment groups. Then, a permutation test is used to determine whether the covariate imbalance is significant, as assessed by the multi-sample test statistics. This is done by computing a distribution of the multi-sample test statistics under the null hypothesis that there is no covariate imbalance between the meta-arms. If the observed multi-sample test statistics are significantly large, then combinability is not satisfied.

Subsequently, a balancing procedure is proposed to minimise the covariate imbalance if combinability is not satisfied. This balancing procedure works by discarding a selection of treatment groups from the meta-analysis, such that the multi-sample test statistics indicate that the covariate imbalance is no longer significant. The result is a more combinable set of treatment groups that can be used for the purposes of meta-analysis. Finally, a simulation study of the balancing procedure is done for three and four treatment groups. In these simulations, the treatment groups are simulated with different underlying distributions, such that in theory the covariate imbalance is significant. These simulations seem to indicate that the more treatment groups there are, the more groups need to be discarded before the covariate imbalance is no longer significant. This is explained by the fact that the initial covariate imbalance is larger if there are more treatment groups. On average, in the case of three treatment groups, nearly a fifth of groups needs to discarded, while in in the case of four treatment groups, roughly a third of groups needs to be discarded. Finally the use of five multi-sample test statistics in the balancing procedure result in a sizeable overlap of groups that are discarded.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
RCT	Randomized Clinical Trial
OBS	Observational Study
ECDF	Empirical Cumulative Distribution Function
MNDF	Monotonically Non-Decreasing Function

Symbols

Symbol	Definition
n	Number of studies included in meta-analysis.
g	Number of treatment groups.
X	Matrix with entry X_{ij} representing the mean covariate value of treatment group j in study i.
P	Matrix with entry P_{ij} representing the number of participants in treatment group j in study i.
D	The domain of covariate values in the data.
$\tilde{F}_j$	The ECDF of the covariate of interest in meta-arm j.
F_{mean}	The pointwise mean values of the ECDFs of the meta-arms.
F_{median}	The pointwise median values of the ECDFs of the meta-arms.
$\tilde{F}_{\text{joint}}$	The ECDF of the joint sample of the meta-arms.
F_{min}	The pointwise minimum values of the ECDFs of the meta-arms.
F_{max}	The pointwise maximum values of the ECDFs of the meta-arms.
T_{FG}	The Wasserstein metric between two MNDFs F and G.
T_{pairwise}	Multi-sample test statistic based on pairwise Wasserstein metric of the meta-arms.
T_{mean}	Multi-sample test statistic based on the Wasserstein metrics between F_{mean} and the meta-arms.
T_{median}	Multi-sample test statistic based on the Wasserstein metrics between F_{median} and the meta-arms.
T_{joint}	Multi-sample test statistic based on the Wasserstein metrics between $\tilde{F}_{\text{joint}}$ and the meta-arms.
$T_{\text{min-max}}$	Wasserstein metric between F_{min} and F_{max}.
$T_{1-\alpha}$	$100(1-\alpha)\%$ quantile of the null distribution of the multi-sample test statistic from the permutation test.

1

Introduction

The method of meta-analysis combines the treatment effects of multiple independent studies answering a common research question. The result is that meta-analyses have more statistical power than single studies [4]. This makes them particularly useful when combining smaller studies that individually lack power to detect a significant effect, but in the context of meta-analysis may reach significance. This allows meta-analyses to give more reliable results by detecting small effect sizes or rarely occuring effects. Consequently, they can be particularly useful in the case of rare diseases. Due to the increased range of values in participant characteristics, the treatment effect can also be extended to larger populations. Therefore, they are an important instrument in the literature in answering causal questions about the effect of a specific treatment. Hence, they are widely applied to determine public health policy and shape guidelines.

One of the main goals of meta-analysis is to answer causal questions about the treatment effect. To do this, it is important that the outcome is solely a result of the treatment. Therefore, other factors should be excluded as possible confounding factors. Factors that may also affect the outcome are also called covariates. Examples of covariates include the age, gender, socioeconomic status and lifestyle factors of participants and the dosage of treatment and duration of the study. To determine a causal effect, it is vital that these covariates are similarly distributed between the treatment groups. For example, consider that one wants to establish a causal effect of smoking on lung cancer. If the group of non-smokers consists of participants younger than 30 years and the group of smokers consists of participants older than 50 years, it becomes hard to tell whether the smoking or the age of the participants is leading to lung cancer.

In the case of experimental studies, like Randomised Clinical Trials (RCTs), the participants are randomly allocated to the treatment groups. As a consequence, the extent of covariate imbalances between treatment groups is limited. In the smoking example, the young and old participants would be randomised between the group of non-smokers and group of smokers. Due to ethical considerations, however, it is argued that such experimental studies should not be undertaken [9]. Therefore, observational studies (OBSs) can play a vital role in such contexts. In OBSs, the participants are not randomly allocated to treatment groups. As a result, the extent of covariate imbalances between treatment groups may be severe.

In the context of meta-analysis, the condition where this covariate balance is present between treatment groups and studies, is referred to as *combinability.* This term can be described as “the extent to which separate studies measure approximately the same thing” [5]. There are two main types of combinability: *basic* combinability and *marginal* combinablity. The first refers to the comparison between the collection of similar types of treatment groups in the meta-analysis, which are called meta-arms. The second type refers to the comparison of subsets of studies with different characteristics [1]. In order to have a meta-analysis of good scientific quality, it is necessary that both types of combinability are satisfied. In this thesis, only basic combinability is considered.

When experimental studies, like RCTs, are combined in meta-analysis, the degree of covariate imbalances between the meta-arms will be limited, since the individual studies are already relatively balanced. However, the large cumulative effect of small imbalances could still lead to a violation of basic combinability [1]. In contrast to RCTs, individual OBSs can already have a significant imbalance between treatment groups. Thus, when OBSs are combined in a meta-analysis, the resulting covariate imbalance between the meta-arms may be substantially larger than in meta-analyses of experimental studies.

Therefore, a preemptive balancing procedure is necessary, in particular for meta-analyses of OBSs. Different techniques of balancing procedures have been proposed. The current state-of-the-art method is that of *propensity score* [8]. This method is generally used to balance a single observational study. In this thesis, the focus is not on balancing a single observational study, but instead, on balancing a meta-analysis of OBSs. However, the method of *propensity score* works best when data is known for each participant, which is generally unavailable in meta-analysis.

Instead, a new preemptive balancing procedure for meta-analyses of OBSs is proposed in this thesis, based on the comparison of empirical cumulative distribution functions of a particular coviarate between the meta-arms. This procedure only considers one covariate of interest. The key idea is that single observational studies may not be balanced individually, but a meta-analysis may become balanced when carefully selecting which treatment groups of the studies are included.

The outline of this thesis is as follows: in Chapter 2 the general framework and structure of the data in context of meta-analyses of observational studies is laid out. In Chapter 3 the assessment of the covariate balance is undertaken. First, previous work in the literature is considered for experimental studies with two treatment groups. This inspires the proposal of an extension to the assessment of covariate imbalance for multi-treatment observational studies. Five different methods are introduced to assess the covariate imbalance in the multi-treatment case. These methods are based on the comparison of empirical cumulative distribution functions of the covariate of interest in each meta-arm. Then, a permutation test is used to determine whether the covariate imbalance is significant as measured by the assessment methods. Subsequently, in Chapter 4 a balancing procedure is proposed that minimises the covariate imbalance by creating a smaller selection of groups to be included in the meta-analysis. This is done by discarding some treatment groups of some studies that are "hindering" combinability based on one of the assessment methods. Lastly, in Chapter 5 a conclusion is drawn and further areas of research are discussed.

The source code used for the assessment method, balancing procedure and graphs in this thesis is found in Appendix A in the programming language *R*.

2 Framework

In the meta-analysis literature, there exist different types of *combinability*. In this thesis, only *basic combinability* as explained by Aiello, Attanasio, and Tinè [1], is considered. This is satisfied if a balance in covariates between the treatment groups is present after combining the studies into a meta-analysis. That is, the values of the coviarate are roughly equally distributed between the treatment groups. Henceforth, when referring to *combinability*, the *basic* type is implied.

As mentioned in Chapter 1, in this thesis, a new preemptive balancing procedure is proposed for a meta-analysis of multi-treatment observational studies (OBSs) that considers one covariate of interest. First, five assessment methods of combinability are proposed. These assessment methods are based on the comparison of the Empirical Cumulative Distribution Functions (ECDFs) of the covariate of interest. Subsequently, a balancing procedure is introduced that aims to create a balanced meta-analysis by discarding treatment groups of the included OBSs. A similar balancing method has been proposed by Attanasio, Aiello, and Tinè [2], but for the case of RCTs with one control and one experimental group. In that case, however, full studies are discarded and not treatment groups. The reason why, in the case of OBSs, it is allowed to discard treatment groups is explained below. If the assessment method concludes that combinability is already satisfied, then no balancing procedure is necessary. As previously stated, in this thesis only one covariate of interest is considered. More specifically, only participant-level variables such as age or comorbidity are considered.

An important goal of a meta-analysis is to answer causal questions about a specific treatment effect. Thus, a pool of studies is collected which all approximately investigate this treatment effect. For the purposes of this thesis, it is assumed that each study contains exactly the same number of treatment groups and that the treatment groups are identically defined in each study. In the context of meta-analysis, only the following data on each treatment group in each study is generally known:

1. The mean value of the covariate of interest;
2. The standard deviation of the covariate of interest;
3. The number of participants.
4. The treatment effect (only for RCTs)

If, however, more data is available, for example on the individual participant level, more accurate techniques may be developed and/or used than proposed in this thesis.

The reason why it is allowed to discard treatment groups instead of discarding full studies, lies in the difference between OBSs and experimental studies, such as RCTs. In RCTs, this should not be done, since information would be lost on the causal treatment effect of that particular treatment group. However, in the case of OBSs this information is not known, as stated by the fourth item mentioned above. A key difference between OBSs and RCTs is that in OBSs the balance between treatment groups is not guaranteed, since no randomisation has occurred in OBSs. Thus, the treatment groups are not actually predefined. The treatment groups are instead determined by the participants and a causal effect of the treatment cannot be inferred [9]. Using meta-analysis of OBSs, the idea is to combine treatment groups of different studies in order to achieve balance overall. Moreover, in the balancing procedure of this thesis, when discarding part of a study, only treatment groups in full are discarded.

In the context of meta-analysis, a meta-arm is defined as the collection of all similar treatment groups [1]. Consider the illustrative example of Figure 2.1 with three meta-arms: the meta-control-arm, the meta-experimental-one-arm and the meta-experimental-two-arm. The meta-control-arm, for example, contains all control groups in the n studies included in this example. This is an example of multi-treatment studies with three treatment groups.

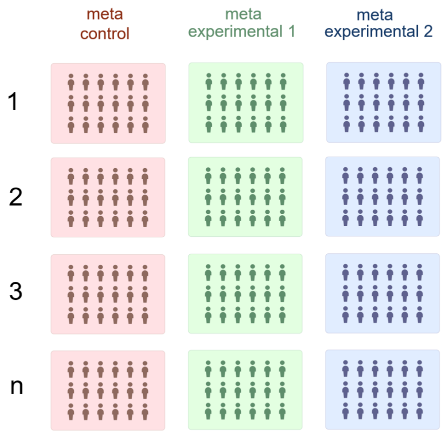


Figure 2.1: An example of a meta-analysis including three meta-arms indicated by the different colours. They all consists of the n similar treatment groups.

Let g be the number of treatment groups and let n be the number of studies. For each treatment group in each study the mean covariate value and the number of participants are known. These values are recorded in the matrices X and P, respectively. X_{ij} represents the mean covariate value of treatment group j in study i and P_{ij} represents the number of participants in treatment group j in study i.

In Table 2.1 and Table 2.2 an illustrative example dataset consisting of X and P is shown to highlight the structure of the data. This example contains $n = 25$ studies and $g = 3$ treatment groups. This example is used a few more times throughout the thesis.

	$X_{\cdot 1}$	$X_{\cdot 2}$	$X_{\cdot 3}$
1	-0.288	0.106	3.598
2	1.727	1.864	2.320
3	0.167	1.393	4.505
4	0.296	1.354	3.821
⋮	⋮	⋮	⋮
25	-0.558	2.139	3.617

Table 2.1: An illustrative example of the mean covariate values in each treatment group in each study.

	$P_{\cdot 1}$	$P_{\cdot 2}$	$P_{\cdot 3}$
1	899	191	820
2	934	306	649
3	689	647	335
4	213	764	460
⋮	⋮	⋮	⋮
25	815	687	409

Table 2.2: An illustrative example of the number of participants in each treatment group in each study.

The domain D is taken as the smallest real interval containing all covariate values in X. For example, if the covariate of interest is *age* in years, D could be $D = [20, 90]$, depending on the ages of the participants. Another possibility for the covariate of interest could be *proportion of participants with diabetes*, in which case $D \subseteq [0, 1]$.

In the next chapter five assessment methods of the covariate imbalance are proposed based on the comparison of ECDFs of the covariate in each meta-arm. First, the case of $g = 2$ treatment groups is considered, before making an extension to $g > 2$ treatment groups.

3

Assessing covariate imbalance

In this chapter, five test statistics are introduced to a priori assess the degree of *basic combinability* in meta-analyses of OBSs. As mentioned in Chapter 1, this occurs when a balance in the covariate of interest is present after combining the studies into a meta-analysis. Thus, *basic combinability* is satisfied if there are no meaningful differences in covariate value between meta-arms. In this chapter, a comparison between Empirical Cumulative Distribution Functions (ECDFs) is used to quantify the difference in covariate values between the meta-arms. This method is inspired by Aiello, Attanasio, and Tinè [1], where the covariate imbalance is assessed in the case of two treatment groups by comparing the ECDFs of the meta-arms. Below, this method is extended to multiple treatment groups. Five different test statistics are introduced in Section 3.1 that assess the covariate imbalance. Subsequently, in Section 3.2 a permutation test is introduced to determine whether the covariate imbalance is significant. These five test statistics and the permutation test are used in the balancing procedures later on in order to minimise the covariate imbalance in Chapter 4.

Let $t \in D$ be an arbitrary covariate value, where D is the domain of the covariate value. The ECDF of the column vector $X_{\cdot j}$ with corresponding weights $P_{\cdot j}$, denoted $\tilde{F}_j$, is given by Formula 3.1. The tilde indicates the fact that it represents empirical data. In essence, this is a weighted version of the classical definition of an ECDF, where the weights are determined by the number of participant in each treatment group.

$$\tilde{F}_j(t) = \frac{\sum_{i=1}^{n} P_{ij} \cdot \mathbb{1}_{\{X_{ij} \leq t\}}}{\sum_{i=1}^{n} P_{ij}} \tag{3.1}$$

$\tilde{F}_j$ represents the distribution of the covariate in a particular meta-arm j. From Formula 3.1 the ECDF value in each point $t \in D$ for each meta-arm can be computed. For the illustrative data of Table 2.1 and Table 2.2 with 25 studies and 3 treatment groups, the result is shown in Figure 3.1.

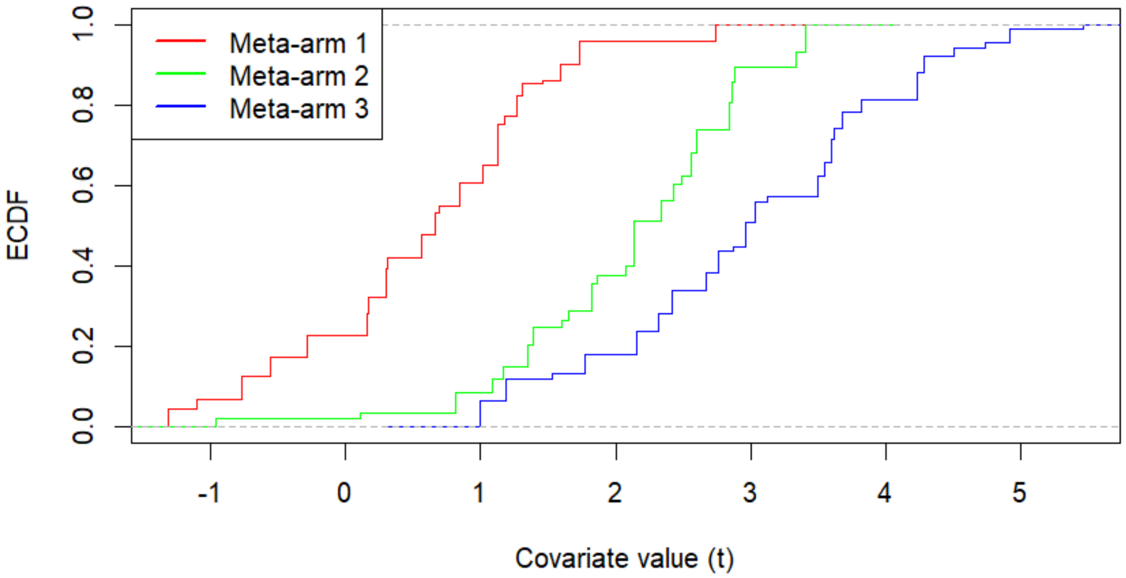


Figure 3.1: The ECDFs of the covariate value for illustrative data with $n = 25$ and $g = 3$, where each colour represents the ECDF of a particular meta-arm.

Recall that *combinability* is satisfied if there are no meaningful differences in the covariate value between meta-arms. In terms of ECDFs, this is satisfied whenever the ECDFs are "similar enough". Then, the distribution of the covariate is "similar" across the meta-arms and combinability is satisfied. Naturally, this raises the question when g ECDFs are "similar".

First, consider the case of $g = 2$ meta-arms. This case has been extensively covered by Aiello, Attanasio, and Tinè [1]. They address this issue for RCTs with one control and one treatment group. They propose the use of a nonparametric two-sample test to determine basic combinability between two ECDFs. For this, they apply the Kolmogorov-Smirnov test and the Anderson-Darling test with a correction for ties, that is, observations in the data with identical covariate value. These tests reject the null hypothesis that the underlying, true distributions are identical in all meta-arms if the resulting p-value is below a certain significant level. If the null is not rejected, the meta-arms are considered similar enough and hence, combinability is satisfied between these two meta-arms.

This could be a useful testing method to a priori assess the combinability or covariate imbalance of a meta-analysis, for both RCTs and OBSs. However, using p-values as the maximisation target in a balancing procedure may not yield the best results. In that case, the p-value could even be misleading, since it is susceptive to the sample size of the meta-arms. Discarding a selection of groups from the meta-analysis automatically reduces the sample size, since fewer participants are included. Then, a change in p-value could be caused by both the discarding of groups "hindering" combinability and the reduction in the number of participants. In the worst case, a group could be discarded, whose removal does not "significantly" alter the shape of the ECDFs of the meta-arms, but whose sample size is large. Then, a relatively large reduction in p-value would occur entirely due to the sample size reduction, even though the meta-arms still have a similar distribution w.r.t. the covariate. Hence, useful data would be discarded and combinability may not even be improved, despite the p-value increasing. Instead, the test statistic of a non-parametric test is used in the balancing procedure.

Inspired by the method proposed by Aiello, Attanasio, and Tinè [1], in this thesis an extension is proposed to multi-treatment observational studies. As previously stated, in this thesis, the assessment of combinability is not based on the p-value but on the test statistic from a non-parametric test. This test statistic can be thought of as the "distance measure" or "amount of difference" between two monotonic non-decreasing functions and is used to assess the covariate imbalance between meta-arms. The reason why monotonic non-decreasing functions are used and not ECDFs will be clear later, however, note that ECDFs are monotonically non-decreasing functions. A definition of this test statistic is given by Definition 3.1.

Definition 3.1 (Two-sample test statistic T_{FG})**.** *Let F and G be two monotonically non-decreasing functions. The statistic T_{FG} is defined as Wasserstein distance metric between F and G, that is,*

$$T_{FG} = \int_{t \in D} |F(t) - G(t)| \; dt$$

.

In this thesis, the Wasserstein distance metric, also known as Earth Mover's distance or Kantorovich–Rubinstein metric[7] is used to measure the difference between two monotonically non-decreasing functions, such as ECDFs . This metric is the area between two monotonically non-decreasing functions. The reason for choosing the Wasserstein metric is two-fold:

1. The Wasserstein metric is not affected by ties in the data, which are frequent in meta-analyses. Tests such as the Kolmogorov Smirnov test and Anderson-Darling test are affected by ties, and therefore Aiello, Attanasio, and Tinè [1] introduced perturbations functions to the data. However, such perturbation functions require assumptions on the data, which is not necessarily guaranteed to correctly reflect the actual data. To circumvent this issue, a test shall be used that is not affected by ties. The Wasserstein metric fullfills this criterion.
2. The Wasserstein metric is sensitive to the shape of the distribution, which means that horizontal differences in the ECDFs have meaning. Thus, it detects a difference between a situation where the covariate values in the samples lie close together and where they lie further apart, even though the vertical differences in ECDFs may be identical. To illustrate this issue, consider three sets of covariate observations $(10, 20, 30)$, $(5, 15, 25)$, and $(9, 19, 29)$ from which the corresponding ECDFs F, G_1 and G_2, respectively, are computed and shown in Figure 3.2 . Intuitively, G_2 is more "similar" to F than G_1 is to F and it is desired that the test statistic reflects this. This is true for the Wasserstein metric, but not, for example, for the Kolmogorov-Smirnov test statistic and Anderson-Darling test statistic.

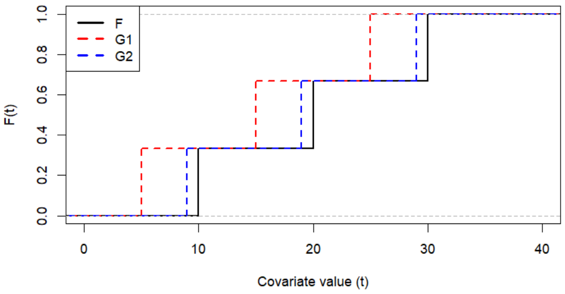


Figure 3.2: An example of ECDFs F,G_1 and G_2 corresponding to covariate observations $(10, 20, 30)$, $(5, 15, 25)$, and $(9, 19, 29)$, respectively.

However, the Definition of 3.1 can be adapted to use any test statistic, not just the Wasserstein metric. Other examples may include the Kolmogorov Smirnov test, the Anderson-Darling test, the Cramér–von Mises criterion and the Kuiper's test.

Having established an assessment method for the covariate imbalance for $g = 2$ treatment groups, an extension of this assessment can now be made for $g > 2$ treatment groups. Multiple paths are possible here. In this thesis, five different multi-sample test statistics are explored. These five multi-sample test statistics are introduced in the next section and measure the covariate imbalance between the meta-arms when $g > 2$. These are referred to as multi-sample test statistics, as opposed to the two-sample statistic from Definition 3.1.

3.1. Multi-sample test statistics

In this section five multi-sample test statistics are introduced that measure the covariate imbalance between the meta-arms. More precisely, these test statistics measure or quantify the “distance” between the ECDFs of the g meta-arms and thus they give a measure for the covariate imbalance. The lower this statistic is, the more combinable the meta-arms are. If the test statistic equals zero, then the meta-arms are perfectly balanced with respect to the covariate.

The first of these multi-sample test statistics is the *pairwise* statistic. The pairwise statistic is determined by calculating the two-sample test statistics $T_{\tilde{F}_{j_1}\tilde{F}_{j_2}}$ of each pairwise combination of ECDFs of the meta-arms. Subsequently, the maximum of all combinations is taken as the pairwise statistic, since the maximum represents the largest imbalance in any pair of the meta-arms. This is summarised in Definition 3.2.

Definition 3.2 (T_{pairwise})**.** $T_{pairwise} = \max\{T_{\tilde{F}_{j_1}\tilde{F}_{j_2}} | j_1, j_2 \in \{1, 2, \ldots, g\}\}$

The intuitive idea is that if the meta-arms are combinable, then the covariate imbalance between all meta-arm pairs should be small.

Secondly, the *mean* statistic is defined by Definition 3.3. First, the monotonic non-decreasing function F_{mean} is computed as in Equation 3.2. It computes the pointwise mean value of the g ECDFs of the meta-arms in a given covariate value $t \in D$. Subsequently, the maximum is taken of the two-sample statistics between F_{mean} and the ECDF of each meta-arm.

$$F_{\text{mean}}(t) = \text{mean}(\tilde{F}_j(t) | j \in \{1, 2, \ldots, g\}) \tag{3.2}$$

Definition 3.3 (T_{mean})**.** $T_{mean} = \max\{T_{\tilde{F}_j F_{mean}} | j \in \{1, 2, \ldots, g\}\}$

Thirdly, the *median* statistic, is given by Definition 3.4. First, the monotonic non-decreasing function F_{median} is computed as in Equation 3.3. It computes the pointwise median value of the g ECDFs of the meta-arms in a given covariate value $t \in D$. Subsequently, the maximum is taken of the two-sample statistics between F_{median} and the ECDF of each meta-arm.

$$F_{\text{median}}(t) = \text{median}(\tilde{F}_j(t) | j \in \{1, 2, \ldots, g\}) \tag{3.3}$$

Definition 3.4 (T_{median})**.** $T_{median} = \max\{T_{\tilde{F}_j F_{median}} | j \in \{1, 2, \ldots, g\}\}$

The fourth statistic is the joint statistic. First, the joint sample is created by combining all covariate values across the meta-arms. Subsequently, the ECDF of this joint sample is computed in each covariate value $t \in D$ as determined by Equation 3.4. This is in fact an ECDF, since the joint sample can be considered empirical data. Then, the maximum is taken of the two-sample statistics between $\tilde{F}_{\text{joint}}$ and the ECDF of each meta-arm.

$$\tilde{F}_{\text{joint}}(t) = \frac{\sum_{j=1}^{g}\sum_{i=1}^{n} P_{ij} \cdot \mathbb{1}_{\{X_{ij} \leq t\}}}{\sum_{j=1}^{g}\sum_{i=1}^{n} P_{ij}} \tag{3.4}$$

Definition 3.5 (T_{joint})**.** $T_{joint} = \max\{T_{\tilde{F}_j \tilde{F}_{joint}} | j \in \{1, 2, \ldots, g\}\}$

The intuitive idea behind the mean, median and joint multi-sample test statistics, is that if all meta-arms are combinable, then each individual meta-arm should be “similar” to some “average” measure of the meta-arms. The F_{mean}, F_{median} and $\tilde{F}_{\text{joint}}$ are suggested as measures for this.

The fifth and final statistic, the *Min-Max* statistic, is based on two monotonically non-decreasing functions of the pointwise minimum and maximum values of the ECDFs of the meta-arms. These minimum and maximum values, denoted F_{min} and F_{max}, respectively, are defined by Formulas 3.5 and 3.6, where $t \in D$.

$$F_{\text{min}}(t) = \min\{\tilde{F}_j(t) | j \in \{1, 2, \ldots, g\}\} \tag{3.5}$$

$$F_{\text{max}}(t) = \max\{\tilde{F}_j(t) | j \in \{1, 2, \ldots, g\}\} \tag{3.6}$$

Then, the *Min-Max* statistic is defined by Definition 3.6.

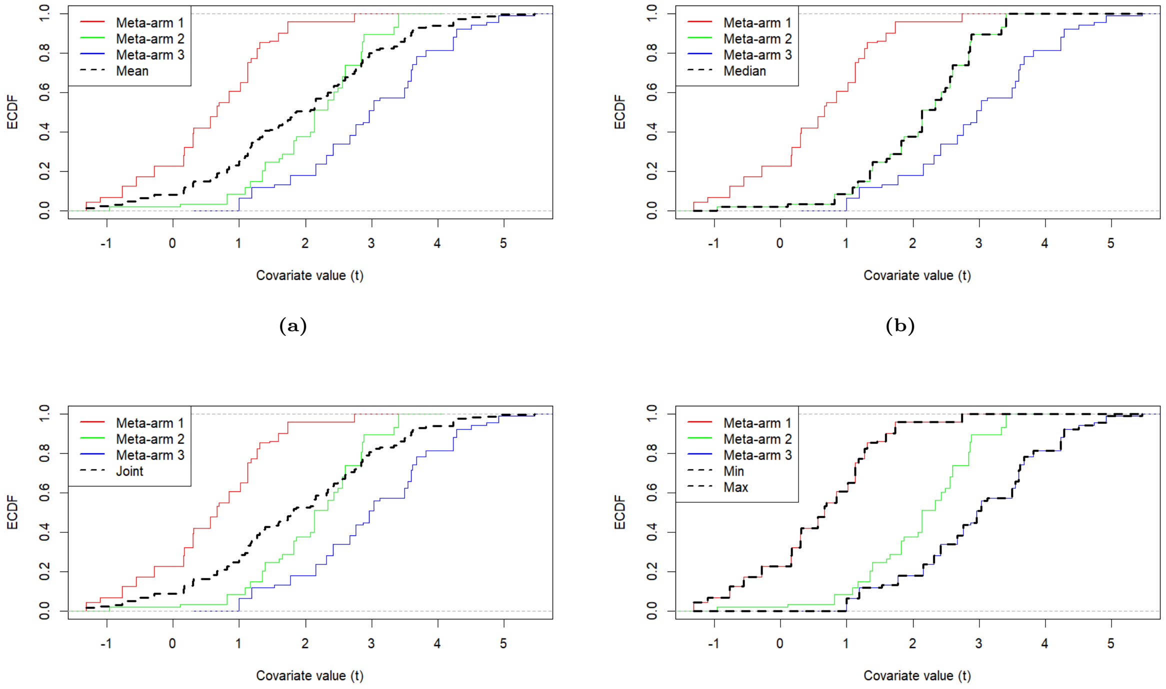
Definition 3.6 ($T_{\text{min-max}}$)**.** $T_{\textit{min-max}} = T_{F_{\text{min}} F_{\text{max}}}$

The intuitive idea here, is that if the maximum and minimum vertical extents of the ECDFs of the meta-arms are "similar", then the meta-arms themselves should also be "similar enough".

Note that, despite F_{mean}, F_{median}, F_{min} and F_{max} being drawn from ECDFs, they are not actually ECDFs themselves, since their distributions do not stem from empirical data and thus, they are referred to as monotonically non-decreasing functions (MNDFs). They are treated the same as ECDFs, however, in the sense that they satisfy the following:

1. F is defined on D and not on $\mathbb{R} \setminus D$.
2. F is non-decreasing;
3. $\lim_{x \uparrow \sup(D)} F(x) = 1$ and $\lim_{x \downarrow \inf(D)} F(x) = 0$, and;
4. F is right-continuous

In Figure 3.3 the MNDFs F_{mean} **(a)**, F_{median} **(b)**, $\tilde{F}_{\text{joint}}$ **(c)** and F_{min} and F_{max} **(d)** are plotted along with the ECDFs of the meta-arms of the illustrative example of Table 2.1 and Table 2.2.



(a)

(b)

(c)

(d)

Figure 3.3: The ECDFs of the meta-arms, in the coloured lines, of the illustrative example of Table 2.1 and Table 2.2, along with the MNDFs F_{mean} **(a)**, F_{median} **(b)**, $\tilde{F}_{\text{joint}}$ **(c)** and F_{min} and F_{max} **(d)** in the dashed black lines.

In fact, it can be shown that $T_{\text{min-max}} \geq T_{\text{pairwise}}$, $T_{\text{pairwise}} \geq T_{\text{mean}}$, $T_{\text{pairwise}} \geq T_{\text{median}}$ and $T_{\text{pairwise}} \geq T_{\text{joint}} \quad \forall g \geq 2$. Proposition 1 gives a proof for the first inequality, but the proof is similar for the others.

Proposition 1. $\forall g \geq 2: \quad T_{\textit{min-max}} \geq T_{\textit{pairwise}}$

Proof. Let $g \geq 2$, $t \in D$ and let $\tilde{F}_1, \tilde{F}_2, \dots \tilde{F}_g$ be the ECDFs of the covariate corresponding to meta-arms $1, 2, \dots, g$, respectively. By construction, $F_{\min}(t) \leq \tilde{F}_j(t) \leq F_{\max}(t) \quad \forall j \in \{1, 2, \dots, g\}$ and thus $|F_{\min}(t) - F_{\max}(t)| \geq |\tilde{F}_{j_1}(t) - \tilde{F}_{j_2}(t)| \quad \forall j_1, j_2 \in \{1, 2, \dots, g\}$.

Now, consider the Wasserstein metric of two MNDFs F and G:

$$T_{FG} = \int_{t \in D} |F(t) - G(t)| \, \mathrm{d}t$$

Then, substituting F and G for $F_{\min}$ and $F_{\max}$ yields that

$$\begin{aligned} T_{\text{min-max}} = T_{F_{\min}, F_{\max}} &= \int_{t \in D} |F_{\min}(t) - F_{\max}(t)| \, \mathrm{d}t \\ &\geq \max_{j_1, j_2} \int_{t \in D} |\tilde{F}_{j_1}(t) - \tilde{F}_{j_2}(t)| \, \mathrm{d}t = \max_{j_1, j_2} T_{\tilde{F}_{j_1} \tilde{F}_{j_2}} = T_{\text{pairwise}} \end{aligned}$$

□

The proofs of the other inequalities are based on the fact that for F_{mean}, F_{median} and $\tilde{F}_{\text{joint}}$, there is at least one meta-arm whose ECDF value is larger, and one whose ECDF value is smaller, at every $t \in D$. For F_{mean} and F_{median} this follows directly from the definitions, but for $\tilde{F}_{\text{joint}}$ this may not be directly obvious. Hence, a quick proof of this fact is given below.

Proof. Let $g \geq 2$. To show: $\forall t \in D: \quad \exists j$ s.t. $\tilde{F}_{\text{joint}}(t) \leq \tilde{F}_j(t)$. Let $t \in D$. Suppose this not the case, thus $\tilde{F}_{\text{joint}}(t) > \tilde{F}_j(t) \quad \forall j \in \{1, 2, \dots, g\}$. Then, by definition of $\tilde{F}_{\text{joint}}$ and $\tilde{F}_j$ it follows that

$$\frac{\sum_{j=1}^{g} \sum_{i=1}^{n} P_{ij} \cdot \mathbb{1}_{\{X_{ij} \leq t\}}}{\sum_{j=1}^{g} \sum_{i=1}^{n} P_{ij}} > \frac{\sum_{i=1}^{n} P_{ij} \cdot \mathbb{1}_{\{X_{ij} \leq t\}}}{\sum_{i=1}^{n} P_{ij}} \quad \forall j$$

$$\Longrightarrow \frac{\sum_{j=1}^{g} \sum_{i=1}^{n} P_{ij} \cdot \mathbb{1}_{\{X_{ij} \leq t\}}}{\sum_{j=1}^{g} \sum_{i=1}^{n} P_{ij}} \cdot \sum_{i=1}^{n} P_{ij} > \sum_{i=1}^{n} P_{ij} \cdot \mathbb{1}_{\{X_{ij} \leq t\}} \quad \forall j$$

Then. summing these inequalities over $j = 1, 2 \dots, g$ together yields

$$\frac{\sum_{j=1}^{g} \sum_{i=1}^{n} P_{ij} \cdot \mathbb{1}_{\{X_{ij} \leq t\}}}{\sum_{j=1}^{g} \sum_{i=1}^{n} P_{ij}} \cdot \sum_{j=1}^{g} \sum_{i=1}^{n} P_{ij} > \sum_{j=1}^{g} \sum_{i=1}^{n} P_{ij} \cdot \mathbb{1}_{\{X_{ij} \leq t\}}$$

$$\Longrightarrow \sum_{j=1}^{g} \sum_{i=1}^{n} P_{ij} \cdot \mathbb{1}_{\{X_{ij} \leq t\}} > \sum_{j=1}^{g} \sum_{i=1}^{n} P_{ij} \cdot \mathbb{1}_{\{X_{ij} \leq t\}} \qquad \Rightarrow\Leftarrow$$

Hence, $\exists j$ s.t. $\tilde{F}_{\text{joint}}(t) \leq \tilde{F}_j(t) \quad \forall t \in D$. The case of $\tilde{F}_j(t) \leq \tilde{F}_{\text{joint}}(t)$ is nearly identical. □

In terms of computations, the $T_{\text{min-max}}$ may have an advantage, since it requires only one Wasserstein metric to be calculated, whereas T_{mean}, T_{median} and T_{joint} require g Wasserstein metrics to be calculated and T_{pairwise} requires $\binom{g}{2}$ calculations of a Wasserstein metric.

In the next section, a permutation test is used to determine whether the covariate imbalance between the meta-arms is significant based on the multi-sample test statistics.

3.2. Permutation test

In the previous section of this chapter, five multi-sample test statistics were introduced to assess the covariate imbalance between the g meta-arms based on the comparison of ECDFs of the covariate. In this section, a permutation test is used to determine whether the covariate imbalance between the meta-arms is significant in a particular dataset. This permutation test can be applied to any of the multi-sample test statistics.

Under the null hypothesis, the underlying distributions of the meta-arms are identical. Hence, the treatment group labels are interchangeable. This allows for a permutation test where a large number of datasets is sampled, with replacement, from the original dataset. Subsequently, the multi-sample test statistics can be recomputed for each of these datasets. This yields a null distribution of the multi-sample test statistics for the original dataset. In essence, this null distribution shows which range of values of the multi-sample test statistic is expected or likely to occur under the null. The details of this permutation test can be found in Algorithm 1.

Algorithm 1 Permutation test for X and P

```
Compute multi-sample test statistic $T^{obs}$ from $X$ and $P$
Define $\alpha$ as the significance level
Set $N_{\text{boots}} = 500$
for m in $\{1, 2, \dots, N_{\text{boots}}\}$ do
    Resample $X'$ with replacement from $X$ and determine $P'$ such that the new covariate value $X'_{ij}$ correspond to the original weights attributed to that value in $P$.
    Compute multi-sample test statistic $T_m$ for $X'$ and $P'$
Define $T_{1-\alpha}$ as the $100(1-\alpha)\%$ quantile of $(T_1, T_2, \dots, T_{N_{\text{boots}}})$
Define $p = \frac{1}{N_{\text{boots}}} \sum_{m=1}^{N_{\text{boots}}} \mathbb{1}_{\{T_m \geq T\}}$
```

The resulting $100(1-\alpha)\%$ quantile of the null distribution of the multi-sample test statistics can then be used to determine whether the covariate imbalance is significant in the original dataset. The null is rejected if the observed multi-sample test statistic in the original dataset $T^{obs} \geq T_{1-\alpha}$ at significance level α. Equivalently, the null is rejected if the p-value obtained by Algorithm 1 is smaller than significance level α.

As an example, this permutation test is applied to the dataset of Table 2.1 and Table 2.2. The observed multi-sample test statistic in this illustrative dataset T^{obs}, the 95% quantile $T_{0.95}$ of the null distribution of the multi-sample test statistics and the p-values as determined by the permutation test are presented in Table 3.1 for each of the five multi-sample test statistics. As an example, the histogram of the null distribution of T_{pairwise} is given in Figure 3.4, for this illustrative dataset.

	T^{obs}	$T_{0.95}$	p-value
T_{pairwise}	2.404	1.037	0.000
T_{mean}	1.313	0.603	0.000
T_{median}	1.536	0.711	0.000
T_{joint}	1.258	0.592	0.000
$T_{\text{min-max}}$	2.404	1.156	0.000

Table 3.1: T^{obs} in the illustrative dataset, $T_{0.95}$ and the p-value as determined by permutation test for the five multi-sample test statistics.

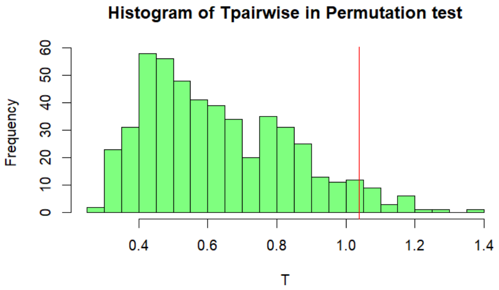


Figure 3.4: Histogram of the null distribution of T_{pairwise} determined by the permutation test of the illustrative dataset. The vertical red line indicates $T_{0.95}$ for T_{pairwise}.

In Table 3.1 each observed multi-sample test statistic is larger than $T_{0.95}$. Therefore, the null is rejected and thus, the covariate imbalance between the meta-arms is significant in this example. Moreover, the p-values are equal to zero in this illustrative example for all multi-sample test statistics.

In the next chapter, a balancing procedure is proposed to minimise the covariate imbalance by discarding some groups. In this balancing procedure $T_{1-\alpha}$ from the permutation test is used to determine when this procedure stops.

4

Minimising covariate imbalance

In this chapter a balancing procedure is introduced. The aim of this procedure is to minimise the covariate imbalance between meta-arms by selectively discarding groups from studies. A group refers to a single treatment group from a single study. This reduces the amount of data used, but may result in a more combinable selection of studies and groups for the purposes of meta-analysis. Therefore, it is desired to discard as few groups as necessary. In the balancing procedure, each of the multi-sample test statistics from Chapter 3 can be used as the measure of covariate imbalance between the g meta-arms and is the objective that is minimised.

In Section 4.1 the balancing procedure is introduced and applied to an illustrative dataset. In Section 4.2 a simulation of the balancing procedure is performed under the alternative hypothesis that the meta-arms have different underlying distributions. Lastly, in Section 4.3 other balancing procedures are discussed that were considered, but ultimately abandoned.

4.1. Balancing procedure

In this section a balancing procedure is proposed that repeatedly discards one group until a stopping condition is satisfied, or when one meta-arm consist of only a single group. In each iteration, the discarded group is selected as the group whose discarding results in the smallest multi-sample test statistic. I.e. it temporarily leaves out one of the non-discarded groups and then measures the resulting multi-sample test statistic. Subsequently, it discards the group for which the lowest multi-sample test statistic has occurred. It repeats this process with the non-discarded groups until the stopping condition is satisfied, or when one meta-arm consist of only a single group.

However, a blind focus on minimising the multi-sample test statistic may not be ideal, since discarding more groups means less available data to investigate the treatment effect in the meta-analysis. Thus, it is desired to discard no more groups than necessary. One way to do this, is by means of the stopping condition. Before discarding any group in the balancing procedure, $T_{1-\alpha}$ is determined from Algorithm 1 from Section 3.2. Since $T_{1-\alpha}$ is determined under the null that all meta-arms stem from the same underlying distribution, $T_{1-\alpha}$ can be considered as an upper bound for an acceptable level of measured covariate imbalance under the null. The stopping condition is then satisfied if the multi-sample test statistic falls below this threshold $T_{1-\alpha}$. Then, the balancing procedure stops and yields the multi-sample test statistic T^{BP} and number of discarded groups at the stopping point as determined by the application of the balancing procedure.

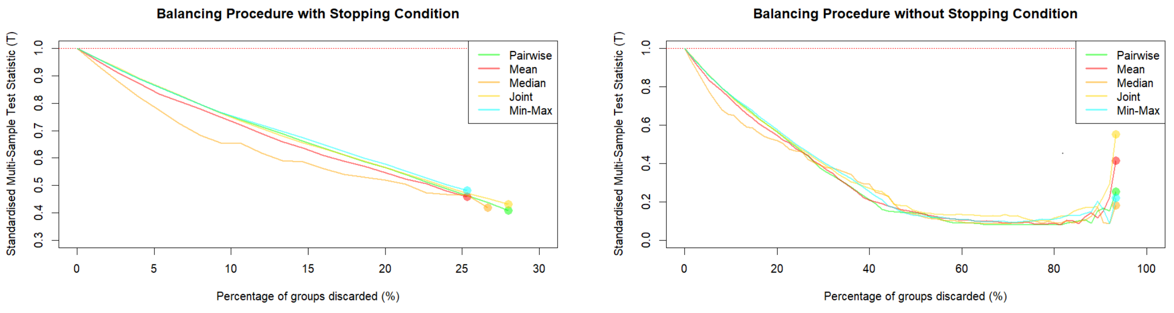
The balancing procedure is given in detail in Algorithm 2.

Algorithm 2 Balancing Procedure

```
Define $k$ as iteration number, with initially $k = 1$.
Define $T_0$ as multi-sample test statistic w.r.t. initial dataset consisting of $X_0$ and $P_0$.
Determine $T_{1-\alpha}$ by Algorithm 1
while $X_{k-1}$ and $P_{k-1}$ contain at least one group in each meta-arm do
    for group $a$ in included groups in $X_{k-1}$ and $P_{k-1}$ do
        $X_{temp} = X_{k-1}$
        $P_{temp} = P_{k-1}$
        Discard group $a$ from $X_{temp}$ and $P_{temp}$
        Calculate multi-sample test statistic w.r.t. updated $X_{temp}$ and $P_{temp}$
        Denote $T_{-a}$ as the resulting multi-sample test statistic when group $a$ is discarded
    Denote $a_{\min} = \arg\min_a T_{-a}$
    Set $T_k = T_{-a_{\min}}$
    if $T_k \leq T_{1-\alpha}$ then
        Stop
    Discard group $a_{\min}$ from $X_{k-1}$ and $P_{k-1}$ and denote as $X_k$ and $P_k$
    Update $k = k + 1$.
Denote $T^{BP} = T_k$ as the multi-sample test statistic at the stopping point.
```

In all figures showing results of applying the balancing procedure, the values of the multi-sample test statistics are standardised to the initial value of the multi-sample test statistics, that is, before discarding any groups. Thus, all multi-sample test statistics are initially equal to 1, even though their non-standardised values are not equal. This standardisation is done to visually compare the different multi-sample test statistics. Only the relative change in multi-sample test statistic, as caused by the balancing procedure, is meaningful. The multi-sample test statistics are different methods to measure the covariate imbalance. Hence, size differences between the different non-standardised multi-sample test statistics do not represent a difference in covariate imbalance or combinability, only a difference in measurement. Moreover, the number of discarded groups is denoted in percentages. Note that this standardisation of the results is done after the balancing procedure. In the balancing procedure, the non-standardised values are used.

As an example, the balancing procedure of Algorithm 2 is applied to the illustrative example, with $n = 25$ and $g = 3$, of Table 2.1 and Table 2.2 of Chapter 2 with and without stopping condition. The resulting graphs are shown in Figure 4.1 with **(a)** and without **(b)** stopping condition. Note that the scales of these graphs are different.



(a) **(b)**

Figure 4.1: The standardised multi-sample test statistic and the corresponding percentage of discarded groups at each iteration in the balancing procedure with **(a)** and without **(b)** stopping condition applied to the illustrative example of Figure 3.1 with $n = 25$ and $g = 3$ for each multi-sample test statistic. The stopping points are indicated by the points at the end of each curve. The different multi-sample test statistics are indicated by the different colours.

In the graphs of Figure 4.1, each curve represents the use of one multi-sample test statistic in the balancing procedure. The value of the multi-sample test statistic is measured and plotted at each percentage of discarded groups corresponding to the iterations of the balancing procedure. The points of each curve represent the stopping points in the balancing procedure. The dashed horizontal red line indicates the value of the initial test statistic. Thus, if the multi-sample test statistic is below this line, then the covariate imbalance between the meta-arms is decreased by discarding groups.

The first thing to note is that the balancing procedure with stopping condition **(a)** yields an almost strict decrease of the multi-sample test statistic until the stopping point. In the balancing procedure without stopping condition **(b)** the multi-sample test statistic decreases before plateauing and then increasing again. Clearly, at some point it is no longer optimal to continue discarding groups in the balancing procedure without stopping condition. This highlights the effect of the stopping condition. By stopping when the multi-sample test statistic is below the threshold $T_{1-\alpha}$, the potentially unnecessary discarding of groups is prevented.

At the right side of Figure 4.1 **(b)**, the multi-sample test statistics tend to increase again with large fluctuations. This is explained by the fact that when fewer non-discarded groups remain, any discarding of an additional group has a larger relative influence on the shape of the ECDFs of the meta-arms. Hence, discarding a group may then drastically change the multi-sample test statistics.

The non-standardised multi-sample test statistics initially and at the stopping point of the balancing procedure as well as the percentage of discarded groups and the relative reduction of the multi-sample test statistic of this illustrative example, with stopping condition, can be found in Table 4.1.

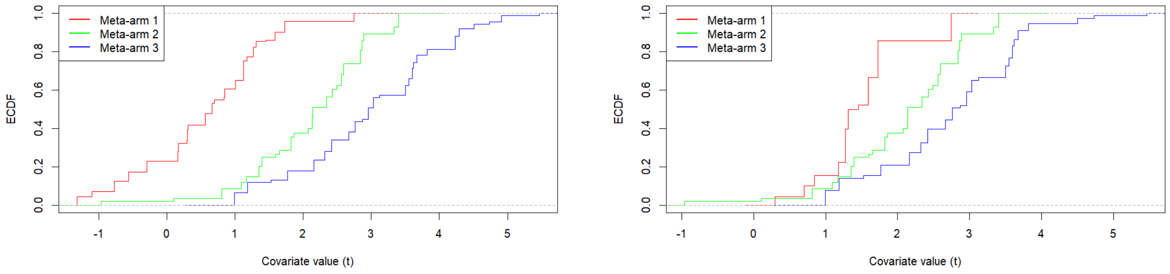
	Percentage of discarded groups	T_0	T^{BP}	Reduction in multi-sample test statistic
T_{pairwise}	28.0%	2.404	0.981	59.2%
T_{mean}	25.3%	1.313	0.603	54.1%
T_{median}	26.7%	1.536	0.645	58.0%
T_{joint}	28.0%	1.258	0.541	57.0%
$T_{\text{min-max}}$	25.3%	2.404	1.156	51.9%

Table 4.1: The non-standardised multi-sample test statistic before (T_0) and after (T^{BP}) applying the balancing procedure, the reduction in multi-sample test statistic in percentages and the corresponding percentage of discarded groups in the illustrative example of Table 2.1 and Table 2.2.

From Table 4.1 it follows that the differences in the reduction of the multi-sample test statistic and the differences in the percentage of discarded groups are relatively modest between the different multi-sample test statistics. In this example, T_{mean} and $T_{\text{min-max}}$ resulted in the lowest percentage of discarded groups, 25.3%, while T_{pairwise} resulted in the largest reduction in the multi-sample test statistic, 59.2%.

A careful consideration must be made though between minimising the number of discarded groups and minimising the multi-sample test statistic. However, since the stopping condition is only satisfied if the multi-sample test statistic is below $T_{1-\alpha}$, the result of using each multi-sample test statistic in Table 4.1 could be considered equally balanced. In that case, the result of T_{mean} or $T_{\text{min-max}}$ may be considered the “best” in this example, since they discard the fewest number of groups.

Taking the result of T_{mean}, one can determine the corresponding dataset after applying the balancing procedure to this example. The resulting ECDFs of the meta-arms of the dataset before and after this balancing procedure with T_{mean} are plotted in Figure 4.2 **(a)** and **(b)**, respectively.



(a)

(b)

Figure 4.2: The ECDFs of the meta-arms of the illustrative example of Table 2.1 and Table 2.2, before **(a)** and after **(b)** applying the balancing procedure with T_{mean}, resulting in discarding 25.3% of groups and reducing T_{mean} by 54.1%.

From the datasets corresponding to Figures 4.2 **(a)** and **(b)** one can compute each multi-sample test statistics and determine a p-value by permutation test for each multi-sample test statistic as detailed in Algorithm 1. The resulting p-values are shown in Table 4.2.

	Before **(a)**	After **(b)**
T_{pairwise}	0.000	0.008
T_{mean}	0.000	0.010
T_{median}	0.000	0.014
T_{joint}	0.000	0.008
$T_{\text{min-max}}$	0.000	0.012

Table 4.2: The p-values in $[0, 1]$ computed by permutation test of Algorithm 1 in the dataset of Figure 4.2 before **(a)** and after **(b)** applying the balancing procedure with T_{mean}.

At significance level $\alpha = 0.05$, all reject the null hypothesis that the underlying distribution is the same in each meta-arm. Thus, the covariate imbalance is still significant before and after applying the balancing procedure in this example. Hence, combinability is not satisfied, but it is improved.

4.2. Simulation study of balancing procedure

In the previous section, a balancing procedure was introduced to make a selection of groups that is more combinable in context of meta-analysis. This is done by discarding groups one by one such that the multi-sample test statistic is minimised. The five multi-sample test statistics of Section 3.1 are used for this purpose. In this section, to understand the behaviour and accuracy of the balancing procedure under the alternative hypothesis, a simulation study is performed of this balancing procedure. Under the alternative hypothesis, the distribution of the covariate is not identically distributed in each meta-arm.

Two simulations are performed, one with $g = 3$ and one with $g = 4$. Each simulation consists of 100 sample datasets. In each sample dataset the number of studies is set at $n = 25$ and $g = 3$ or $g = 4$. The sample datasets each consist of covariate value matrix X and number of participants matrix P, which represent the mean covariate value and number of participants of the groups. Thus, X and P are of shape $25 \times g$. In each sample dataset, these are generated by the following properties:

- $X_{ij} \sim N(j, 1)$
- $P_{ij} \sim U\{100, 1000\}$

Here, $N(\mu, \sigma^2)$ represents a normal distribution with mean μ and standard deviation σ and $U\{a, b\}$ indicates a discrete uniform distribution of integers $\{a, a+1, \ldots, b-1, b\}$. Note that X has a different distribution in each meta-arm, meaning this simulation is performed under the alternative hypothesis, where the meta-arms do not have the same underlying distribution. As a result, the samples in the simulation have, in theory, a substantial covariate imbalance before applying the balancing procedure.

From X and P, the ECDFs of the g meta-arms from Definition 3.1 are computed and then, the MNDFs of F_{pairwise}, F_{mean}, F_{median}, $\tilde{F}_{\text{joint}}$, F_{min} and F_{max} from Section 3.1 can be computed.

Subsequently, the balancing procedure of Algorithm 2 is applied to each sample dataset in the simulation. For each sample dataset, the multi-sample test statistics over the iterations of the procedure are obtained. Each iteration naturally corresponds to a certain number of discarded groups. These values are again standardised in each sample dataset such that the initial multi-sample test statistic equals 1 and the number of discarded groups is expressed in percentages.

For the simulation of $g = 3$, these results are shown per multi-sample test statistic in Figure 4.3, representing T_{pairwise} **(a)**, T_{mean} **(c)**, T_{median} **(e)**, T_{joint} **(g)** and $T_{\text{min-max}}$ **(i)**. For the simulation of $g = 4$, these results are also shown in Figure 4.3, representing T_{pairwise} **(b)**, T_{mean} **(d)**, T_{median} **(g)**, T_{joint} **(h)** and $T_{\text{min-max}}$ **(j)**. Note that the scale of the graphs is different between the simulation of $g = 3$ and $g = 4$. In each graph in Figure 4.3, each line represents one sample dataset to which the balancing procedure is applied. A circular point represents a stopping point of one sample dataset.

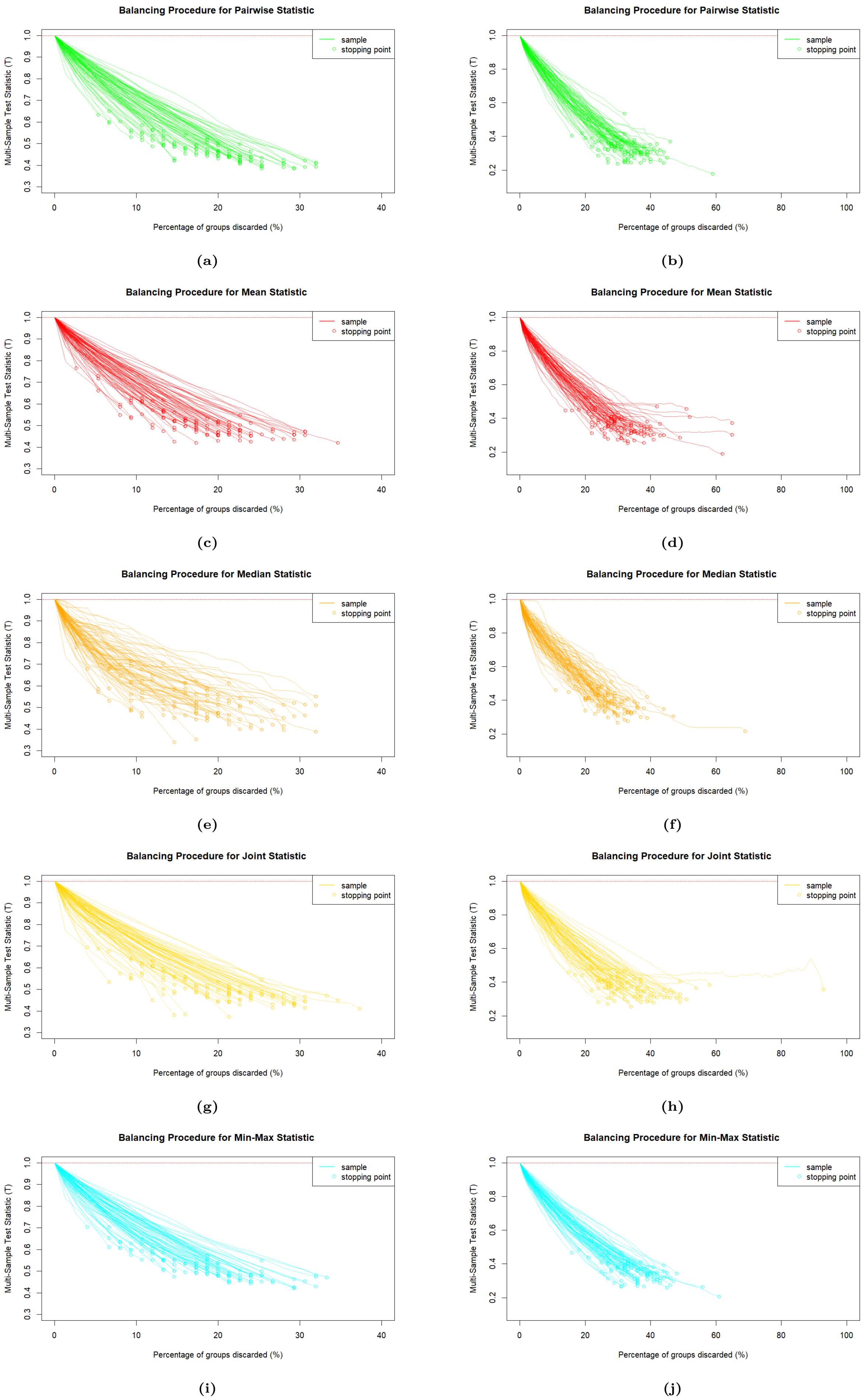
The graphs in Figure 4.3 all show a clearly decrease in the multi-sample test statistic. In the case of $g = 4$ there are a few outliers where the balancing procedure discards substantially more groups than in the other sample datasets. Note, for example, the outlier in the case of T_{joint} and $g = 4$ **(h)**, where more than 90% of groups is discarded. In fact, this was the only case in the simulations where the stopping condition was not met and instead the procedure stopped since one meta-arm consisted of only one group.

Note, that in the case of T_{median} the lines are less smooth and instead resemble a more "twisting" or "zigzagging" motion. This may be caused by the fact that the in $g = 3$, F_{median} is equal to one of the ECDFs of the meta-arms. Thus, if the ECDFs are altered by the discarding of groups, F_{median} may change less smoothly than other MNDFs such as F_{mean} and $\tilde{F}_{\text{joint}}$. However, this also happens in the case of $g = 4$, where the median value essentially becomes a mean between the middle ECDFs of the meta-arms.

Moreover, the multi-sample test statistics and the number of discarded groups at the stopping points are determined in each sample dataset of both simulations. These values across all sample datasets are then combined, from which the mean, 2.5% and 97.5% quantiles of the multi-sample test statistic at the stopping points are computed. The range of values between the 2.5% and 97.5% quantile is then denoted as the 95% range of values. In Table 4.3 and Table 4.4, these values at the stopping points are shown for the simulation of $g = 3$ and $g = 4$, respectively

	Percentage of discarded groups		Reduction in multi-sample test statistic	
	Mean	95% value range	Mean	95% value range
T_{pairwise}	18.8%	[6.7%, 31.4%]	52.9%	[39.8%, 61.1%]
T_{mean}	17.9%	[5.3%, 30.7%]	48.7%	[31.0%, 57.5%]
T_{median}	16.9%	[3.3%, 31.4%]	46.7%	[25.2%, 60.8%]
T_{joint}	19.6%	[6.0%, 32.1%]	49.8%	[31.7%, 60.3%]
$T_{\text{min-max}}$	18.4%	[6.7%, 32.0%]	47.9%	[34.9%, 57.1%]

Table 4.3: The mean and 95% range of values of the percentage of discarded groups and reduction in multi-sample test statistic at the stopping points in the simulation for $g = 3$.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

Figure 4.3: The multi-sample test statistic and corresponding percentage of discarded groups at each iteration in the balancing procedure in the simulation of 100 sample datasets with $n = 25$. The figures on the left indicate the simulati0n with $g = 3$ and on the right with $g = 4$. Each curve and its stopping point indicated by the circle represent one sample dataset in the simulation

	Percentage of discarded groups		Reduction in multi-sample test statistic	
	Mean	95% value range	Mean	95% value range
T_{pairwise}	32.6%	[20.5%, 44.5%]	67.2%	[54.2%, 75.8%]
T_{mean}	32.3%	[19.0%, 57.2%]	64.6%	[53.2%, 74.2%]
T_{median}	27.3%	[15.0%, 41.6%]	60.3%	[45.2%, 72.1%]
T_{joint}	34.2%	[19.5%, 52.6%]	63.5%	[51.0%, 72.2%]
$T_{\text{min-max}}$	35.1%	[21.5%, 47.5%]	64.4%	[52.2%, 73.6%]

Table 4.4: The mean and 95% range of values of the percentage of discarded groups and reduction in multi-sample test statistic at the stopping points in the simulation for $g = 4$.

From Table 4.3 and Table 4.4 it follows that for both simulations using T_{median} in the balancing procedure yields on average the lowest percentage of discarded groups (16.9% and 27.3%), while T_{pairwise} yields the largest reduction in the multi-sample test statistic (52.9% and 67.2%). This is also reflected in the corresponding 95% range of values. However, the differences between the percentage of discarded groups and the differences between the reduction in multi-sample test statistic are modest between the five multi-sample test statistics.

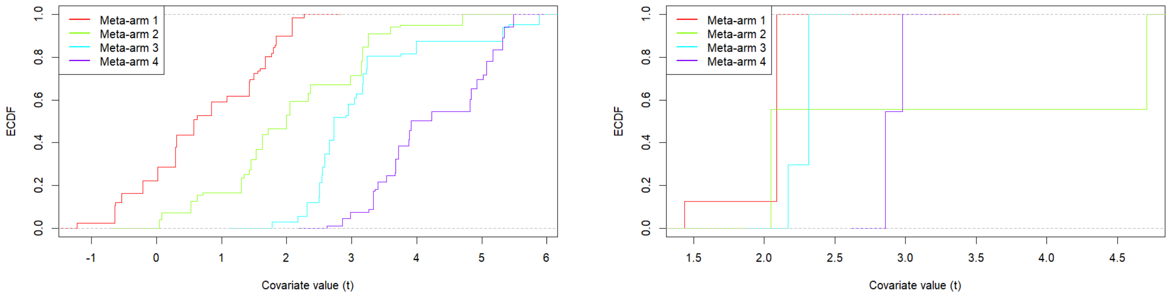
Moreover, for $g = 4$ the percentage of discarded groups and the reduction in the multi-sample test statistics is substantially larger than for $g = 3$. This may imply that in the case of $g = 4$, more groups need to be discarded such that a lower multi-sample test statistic is reached, before the stopping condition is satisfied. Thus, it takes a larger reduction in multi-sample test statistic until this value reaches the corresponding $T_{0.95}$ quantile of the null distribution of the multi-sample test statistics. This may point to the fact that, for $g = 4$, the meta-arms have a larger covariate imbalance to begin with.

Below, three examples of the balancing procedure are shown corresponding to datasets from the simulation. Note that the scales are not the same in these figures.

First, the ECDFs of the meta-arms of the sample dataset where the most groups were discarded (93%) in Figure 4.3 **(h)** (T_{joint} with $g = 4$) is shown in Figure 4.4 before **(a)** and after **(b)** applying the balancing procedure with T_{joint}. In this case, the balancing procedure does not appear to give satisfactory results, as nearly all groups are discarded. However, applying the balancing procedure with T_{pairwise}, T_{mean}, T_{median} and $T_{\text{min-max}}$ leads to 38%, 43%, 27% and 42% of groups being discarded, respectively. Thus, this is a particular case where the use of T_{joint} was unfruitful.

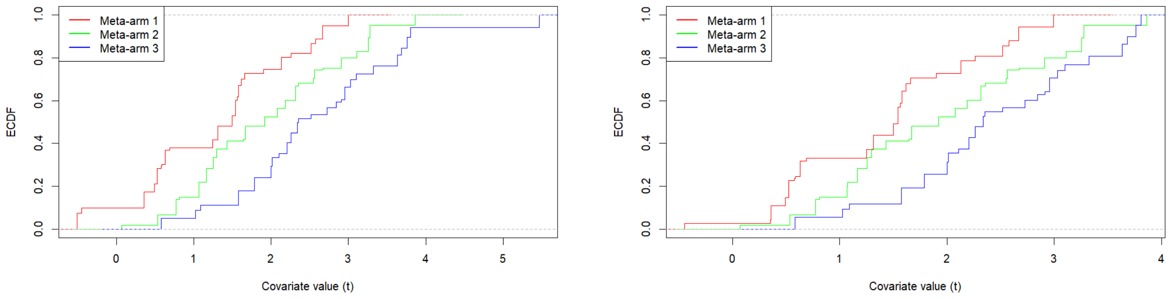
Secondly, the ECDFs of the meta-arms of the sample dataset that discarded the fewest groups (4%) in Figure 4.3 **(i)** ($T_{\text{min-max}}$ with $g = 3$) is shown in Figure 4.5 before **(a)** and after **(b)** applying the balancing procedure with $T_{\text{min-max}}$. There is not a lot of change between the before and after image, since only 4% of groups are discarded. Note that it appears that the groups are discarded with the most extreme covariate, since the “tails” of the ECDFs are reduced.

Thirdly, the ECDFs of the meta-arms of the first sample dataset in the simulation with $g = 4$ is shown in Figure 4.6 before **(a)** and after **(b)** applying the balancing procedure with T_{pairwise}. Note that the ECDFs are shifted towards the covariate range where each meta-arm contains groups with covariate in that range.



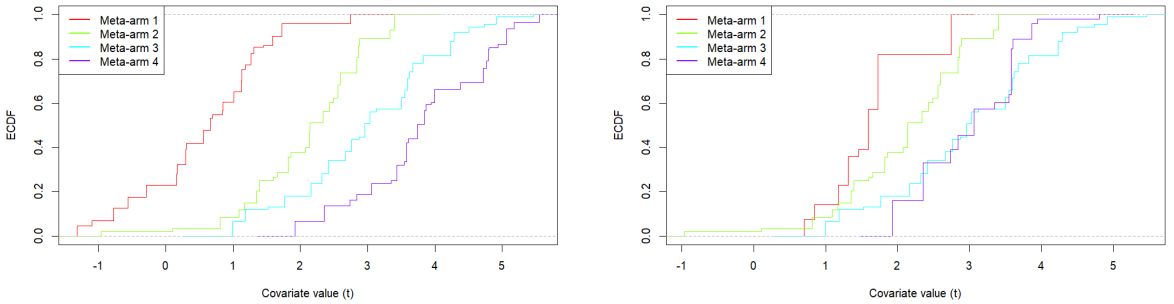
(a) (b)

Figure 4.4: An example of ECDFs of the meta-arms before **(a)** and after **(b)** applying the balancing procedure with T_{joint}, resulting in discarding 93% of groups and reducing T_{mean} by 64.4%.



(a) (b)

Figure 4.5: An example of ECDFs of the meta-arms before **(a)** and after **(b)** applying the balancing procedure with $T_{\text{min-max}}$, resulting in discarding 4% of groups and reducing T_{mean} by 29.5%.



(a) (b)

Figure 4.6: An example of ECDFs of the meta-arms before **(a)** and after **(b)** applying the balancing procedure with T_{pairwise}, resulting in discarding 30% of groups and reducing T_{mean} by 60.5%.

Lastly, a natural question of the balancing procedure, is whether the same groups are discarded when using different multi-sample test statistics. To investigate this, the overlap of groups between every combination of two multi-sample test statistics is computed in each sample dataset of the simulations. The overlap is the number of discarded groups, not percentage, that two balancing procedures have both discarded at the stopping point. Subsequently, the mean of these overlaps can be taken over all sample datasets. The resulting mean overlaps are presented in Table 4.5 and Table 4.6, for the simulation of

$g = 3$ and $g = 4$, respectively. Recall that there were 75 and 100 groups, respectively, in total in each sample dataset. Note that the overlap of a balancing procedure using a a particular test statistic with itself is just all groups it discards. This is represented by the main diagonal.

	Pairwise	Mean	Median	Joint	Min-Max
Pairwise	14.1	10.6	8.9	10.8	12.2
Mean		13.5	9.5	10.3	11.2
Median			12.7	9.1	9.2
Joint				14.7	10.9
Min-Max					13.8

Table 4.5: The mean overlaps of each combination of two multi-sample test statistics, in the simulation with $g = 3$.

	Pairwise	Mean	Median	Joint	Min-Max
Pairwise	32.6	28.0	25.2	25.9	28.1
Mean		32.3	25.0	26.7	28.1
Median			27.3	23.0	24.7
Joint				34.2	28.0
Min-Max					35.1

Table 4.6: The mean overlaps of each combination of two multi-sample test statistics, in the simulation with $g = 4$.

From Table 4.5 one can conclude that the smallest overlap in the simulation of $g = 3$ occurred between T_{median} and T_{joint} at 9.1 groups overlap on average, while they individually discarded 12.7 and 14.7 groups on average, respectively. In the simulation of $g = 4$, the smallest overlap also occurred between T_{median} and T_{joint} at 23.0 groups overlap on average, while individually they discarded 27.3 and 34.2 groups on average, respectively. In most combinations, the mean overlap is relatively large, thus the use of different multi-sample test statistics in the balancing procedure result in quite some overlap between the selection of groups that are discarded.

4.3. Abandoned balancing procedures

As a side note, the balancing procedure presented in Algorithm 2 was not the only procedure developed and investigated for this thesis. However, other balancing procedures that were considered were not effective at minimising the covariate imbalance. In essence, they tried to find a criterium that would yield the group whose discarding results in the lowest multi-sample test statistic. That would be the same group as in the balancing procedure in Algorithm 2, but may result in an algorithm with faster computation time. The considered balancing procedures were the following:

- **Histogram method**:
  This method is based on an estimator of the density of the covariate for each meta-arm, considering the number of participants in each group as weights. In this method, the histogram was used as an estimator of the density. Similar to ECDFs, if the meta-arms have the same underlying distribution of the covariate, then the density of the covariate should be the same in each meta-arm. This fact is used to determine which group is discarded in each iteration. To determine this group, the method first determined the “bin” in the histogram with the largest frequency difference between the meta-arms. The contribution to the covariate imbalance would then be considered the greatest at the covariate values of that bin and hence, the group is discarded that decreases the frequency difference in this bin the most. This procedure then repeats the same steps, but without the discarded group and continues until one meta-arm has only one group left or the stopping condition is satisfied.
- **Adapted histogram method**:
  This method is similar to the histogram method, but with an adapted “histogram”. First, it constructs a grid of all unique covariate values in X. The goal is to find the point on the grid where the largest covariate imbalance between the meta-arms occurs. This is done by, for each

point on the grid, determining the number of groups in each meta-arm that have covariate value "close" to this point. In this method, "close" meant that the difference between the value on the axis and the covariate value of a group was smaller or equal than the mean step size between the grid values. Then, it determined at which covariate value on the grid the difference between the meta-arms is largest in the number of groups that are "close". Subsequently, the group is discarded such that this difference decreases the most. This procedure then repeats the same steps, but without the discarded group and continues until one meta-arm has only one group left or the stopping condition is satisfied.

- **Maximum height method**:
  This method determines the covariate value on the grid for which the largest vertical difference between the ECDFs of the meta-arms occurs. Then, a group is discarded with that covariate value. This groups must also belong to the meta-arm with the largest ECDF value at that covariate value. This procedure then repeats the same steps, but without the discarded group and continues until one meta-arm has only one group left or the stopping condition is satisfied.
- **Difference-to-measure method**:
  This method is similar to the maximum height method, but instead does not use the maximum vertical height, but the largest vertical height between the ECDFs of the meta-arms and a measure MNDF. This measure MNDF is taken as either F_{mean}, F_{median} or $\tilde{F}_{\text{joint}}$. Then, a group is discarded in the same way as the maximum height method. This procedure then repeats the same steps, but without the discarded group and continues until one meta-arm has only one group left or the stopping condition is satisfied.

However, these procedures did not produce fruitful results in the sense that they did not decrease the multi-sample test statistics with certainty. They would not give the same result as the balancing procedure in Algorithm 2 and instead, the multi-sample test statistic would on average barely decrease, or even increase. In fact, these balancing procedures seemed to almost arbitrarily determine which group is discarded at each iteration. The presumed reason for this is three-fold:

1. Firstly, in the case of the Maximum height method and the Difference-to-measure method, it is uncertain whether these criteria accurately determine the group that is causing the most covariate imbalance and thus, hindering combinability. The problem arises from the nature of ECDFs. If at a certain covariate value two ECDFs have a large vertical difference, then that does not mean that this vertical difference is caused at this value of the covariate. Instead, it could be caused at any covariate value smaller than this value, since ECDFs are cumulative. A way to circumvent this issue may be to look at, for example, histograms instead, since these are not cumulative. However, the Histogram method was not fruitful either.
2. Secondly, the Histogram method puts all covariate values into distinct bins. Thus, two groups that are "close" in covariate value, could have their covariate value sorted into separate bins. Subsequently, this method could make two errors. First, it could consider the frequency difference in a particular bin as the largest, even though this difference may be reduced when including groups "close" in covariate values but with covariates sorted into different bins. Secondly, in a particular bin it could consider the frequency difference to be small, even though this difference may be increased when including groups "close" in covariate values but with covariates sorted into different bins. Moreover, the Histogram method is very dependent on the size of the bins. Perhaps choosing a more appropriate bin size may improve this method. To prevent these problems, the Adapted histogram method was developed, but it was also unfruitful.
3. Lastly, the Histogram method and Adapted histogram method are limited by the number of participants in each group, since only full groups and thus, a fixed number of participants can be discarded. This means that a large covariate imbalance for a certain bin, can only be reduced by discarding a fixed amount of frequency. Hence, this discarding could even result in an increased covariate imbalance which has been reversed between the meta-arms, because too much frequency has been discarded.

These reasons may explain why these alternative balancing procedures were unfruitful. Hence, they were abandoned.

5

Conclusion & discussion

Meta-analysis is a powerful and useful method to combine the treatment effects of multiple independent studies. In order to answer causal questions about the treatment effect, it is crucial that combinability is ensured. In this thesis, the basic type of combinability was studied, which refers to the covariate imbalance between meta-arms. If the meta-arms are not balanced with respect to a particular covariate, then the covariate cannot be excluded as a confounding factor. This problem is of even greater importance in the case of observational studies, since they are inherently less balanced between treatment groups.

Therefore, in Chapter 3 five multi-sample test statistic are proposed to assess the covariate imbalance between meta-arms. This assessment is based on the ECDFs of the meta-arms with respect to the covariate. In the case of $g = 2$ treatment groups, the Wasserstein metric is used as the two-sample test statistic measuring the covariate imbalance between two MDNFs. Subsequently, five extensions are made to the multi-treatment case determined by taking the maximum of one or more two-sample test statistics. These multi-sample test statistics are T_{pairwise}, T_{mean}, T_{median}, T_{joint} and $T_{\text{min-max}}$ and represent the covariate imbalance in the multi-treatment case.

To determine the significance of the covariate imbalance, a permutation test is used. This permutation test yields a null distribution of the multi-sample test statistics and the corresponding $100(1-\alpha)\%$ quantile $T_{1-\alpha}$. If the observed multi-sample test statistic is larger than this value, then the covariate imbalance is significant and a balancing procedure is required. This value is used in the balancing procedure as stopping condition.

In Chapter 4 a balancing procedure is introduced that aims to minimise the covariate imbalance, as can be measured by any of the multi-sample test statistics. This is done by discarding groups one by one based on whichever group's discarding results in the lowest multi-sample test statistic. This is continued until the multi-sample test statistic is below $T_{1-\alpha}$, or at least one meta-arm contains only a single group.

Subsequently, a simulation study for each multi-sample test statistic is performed of the balancing procedure, for $g = 3$ and $g = 4$. These simulations all consisted of 100 sample datasets with $n = 25$. For $g = 3$, the balancing procedure resulted on average in a 18.8%, 17.9%, 16.9%, 19.6% and 18.4% of groups being discarded yielding a reduction in the multi-sample test statistic of 52.9%, 48.7%, 46.7%, 49.8% and 47.9%, respectively, for the multi-sample statistics T_{pairwise}, T_{mean}, T_{median}, T_{joint} and $T_{\text{min-max}}$, respectively. For $g = 4$, the balancing procedure resulted on average in a 32.6%, 32.3%, 27.3%, 34.2% and 35.1% of groups being discarded yielding a reduction in the multi-sample test statistic of 67.2%, 64.6%, 60.3%, 63.5% and 64.4%, respectively. Thus, in the case of $g = 4$ a larger reduction in the multi-sample test statistic is needed until the multi-sample test statistic is below $T_{1-\alpha}$. Hence, more groups need to be discarded to reach this reduction. This may indicate that the initial covariate imbalance in the simulation of $g = 4$ is substantially larger than in the simulation of $g = 3$.

Using T_{median} in the balancing procedure yielded the lowest percentage of discarded groups, while the

T_{pairwise} statistic resulted in the highest reduction of the multi-sample test statistic value, in both simulation of g. Then, T_{median} may be considered the "best", since it discards the fewest groups. The reduction in multi-sample test statistic is less relevant, since the stopping condition at $T_{1-\alpha}$ may ideally guarantee combinability. However, in practice, combinability is not necessarily satisfied after the balancing procedure, but it is improved.

A way to determine which multi-sample test statistic yields the "best" result, may be to reassess the covariate imbalance after the balancing procedure. Then, by permutation test a p-value can be obtained using each multi-sample test statistic. The "best" result may then be obtained by the multi-sample test statistic whichever results in the fewest groups being discarded, but has $p > \alpha$ for all multi-sample test statistics in the reassessment of the covariate imbalance.

Moreover, in the simulation study it turned out that the balancing procedures, with different multi-sample test statistic, result in quite some overlap of groups being discarded. However, it would be interesting to investigate this further. Maybe the intersection of groups discarded by all the different multi-sample test statistics in the balancing procedure, can be used as a starting point. One discards all these groups and then assesses the covariate imbalance and if need be, apply the balancing procedure from there.

In addition, some further areas of research may include:

- The two-sample test statistic used in this thesis is the Wasserstein metric. However, other two-sample test statistic can be explored as well, such as the Kolmogorv-Smirnov test, the Anderson-Darling test, the Cramér-von Mises test and the Kuiper's test.
- The multi-sample test statistics introduced in this thesis are just some possible ways to make an extension to the multi-treatment case. Of course, other multi-sample test statistics may be used or defined to extend to the multiple treatment case. For example, a multi-sample Anderson-Darling test statistic may be used, as proposed by Scholz and and [10].
- Moreover, the methods in this thesis consider only a single covariate. However, multiple covariates may be of interest, therefore an extension could be made to the multivariate case. This could be extended by considering multivariate ECDFs and testing similarity using multivariate two-sample tests, such as proposed by Justel, Peña, and Zamar [6] and Baringhaus and Franz [3].
- The proposed balancing procedure in this thesis discards groups one by one. This makes it near-sighted, in the sense that, at every iteration, it only considers the discarding of one group. It does not consider the fact that more groups may be discarded at later iterations. Thus, there could be a "better" combination of groups than is found by discarding one by one. However, the balancing procedure of this thesis might be used to determine an upper bound for the number of groups that needs to be discarded to find an optimum.
- Another issue of the balancing procedure is that it only discards groups. After having applied the balancing procedure, it could be that one group could be re-added without significantly increasing the covariate imbalance. A solution could be to allow the possibility of restoring discarded groups in the balancing procedure. Thus, at each iteration, the possibility of restoring any discarded group is considered as well and unless this leads, for example, to a larger multi-sample test statistic, a discarded group is restored.
- The stopping condition in this thesis is based on the null distribution of the multi-sample test statistics as determined by the permutation test. The balancing procedure stops if the multi-sample test statistic is below the $100(1-\alpha)\%$ quantile of the null distribution of the multi-sample test statistics. However, this null distribution is computed before the balancing procedure and therefore is valid for the initial dataset before discarding groups. The stopping condition could be improved by recomputing the null distribution of the multi-sample test statistics at each iteration in the balancing procedure, and then stopping if the null is rejected.
- The result of the balancing procedure is not necessarily balanced between the meta-arms. A dataset can have significant covariate imbalance before and after the balancing procedure. One way to reach a balanced result, could be to repeatedly perform the balancing procedure until the end result is in fact balanced. However, it is unsure whether every dataset can even be made balanced by the balancing procedure.

- In the simulation of the balancing procedure, the meta-arms were simulated using different underlying normal distributions with mean j and standard deviation 1 for meta-arm j. However, the first and last meta-arm may then barely have any overlap in values. This is especially true if g increases. It may be interesting to perform this simulation with different underlying distributions, that may or may not have more overlap in values between the meta-arms.
- Instead of a permutation test to determine the null distribution of the multi-sample test statistics, one may also perform a Monte Carlo simulation under the null hypothesis that the underlying distributions are the same in each meta-arm. For example, the covariate in each meta-arm could be distributed according to a uniform distribution on $[0, 1]$. However, this method works best when the assumed underlying distribution is representative of the structure of the real data.
- Lastly, the standard deviation of the covariate in each treatment in each study is generally known in meta-analyses. However, it was not used in this thesis. Perhaps, the standard deviation can be used to introduce uncertainty in the ECDFs of the meta-arms. One could also generate a large number of slightly different datasets than the original dataset. Each dataset may, for example, be generated as a normal distribution with parameters set to the mean covariate value in each treatment group and the standard deviation in that treatment group. Then, the multi-sample test statistics can be calculated over all these datasets and give a sense of variability in the assessment of the covariate imbalance.

References

[1] Fabio Aiello, Massimo Attanasio, and Fabio Tinè. “Assessing covariate imbalance in meta-analysis studies”. In: *Statistics in Medicine* 30.22 (2011), pp. 2671–2682. DOI: https://doi.org/10.1002/sim.4311. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.4311. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4311.

[2] Massimo Attanasio, Fabio Aiello, and Fabio Tinè. “A statistical method for removing unbalanced trials with multiple covariates in meta-analysis”. In: *PLoS ONE* 18.12 (Dec. 2023), e0295332. DOI: 10.1371/journal.pone.0295332. URL: https://doi.org/10.1371/journal.pone.0295332.

[3] L. Baringhaus and C. Franz. “On a new multivariate two-sample test”. In: *Journal of Multivariate Analysis* 88.1 (2004), pp. 190–206. ISSN: 0047-259X. DOI: https://doi.org/10.1016/S0047-259X(03)00079-4. URL: https://www.sciencedirect.com/science/article/pii/S0047259X03000794.

[4] Nancy G Berman and Robert A Parker. “Meta-analysis: Neither quick nor easy”. In: *BMC Medical Research Methodology* 2.1 (Aug. 2002). ISSN: 1471-2288. DOI: 10.1186/1471-2288-2-10. URL: http://dx.doi.org/10.1186/1471-2288-2-10.

[5] Thomas C. Chalmers et al. “Meta-analysis of clinical trials as a scientific discipline. I: Control of bias and comparison with large co-operative trials”. In: *Statistics in Medicine* 6.3 (1987), pp. 315–325. DOI: https://doi.org/10.1002/sim.4780060320. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.4780060320. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4780060320.

[6] Ana Justel, Daniel Peña, and Rubén Zamar. “A multivariate Kolmogorov-Smirnov test of goodness of fit”. In: *Statistics & Probability Letters* 35.3 (1997), pp. 251–259. ISSN: 0167-7152. DOI: https://doi.org/10.1016/S0167-7152(97)00020-5. URL: https://www.sciencedirect.com/science/article/pii/S0167715297000205.

[7] A. Lipp and P. Vermeesch. “Short communication: The Wasserstein distance as a dissimilarity metric for comparing detrital age spectra and other geological distributions”. In: *Geochronology* 5.1 (2023), pp. 263–270. DOI: 10.5194/gchron-5-263-2023. URL: https://gchron.copernicus.org/articles/5/263/2023/.

[8] P.R. Rosenbaum and D.B. Rubin. “The central role of the propensity score in observational studies for causal effects.” In: *Biometrika* 70.1 (1983), pp. 41–55.

[9] William F. Rosenberger and John M. Lachin. *Randomization in clinical trials.* Wiley-Interscience, July 2002.

[10] F. W. Scholz and M. A. Stephens and. “K-Sample Anderson–Darling Tests”. In: *Journal of the American Statistical Association* 82.399 (1987), pp. 918–924. DOI: 10.1080/01621459.1987.10478517. eprint: https://doi.org/10.1080/01621459.1987.10478517. URL: https://doi.org/10.1080/01621459.1987.10478517.

A

Source Code (R)

The source code below, in the programming language *R*, is the code used to produce the results and graphs in this thesis. In Section A.1 a list of functions is coded to be used in Section A.2, where the graphs, permutation test, balancing procedure and simulation are coded.

A.1. Functions

```r

# This function generates fictitious data with g treatment groups and S studies under the null hypothesis.

generate_data = function(S, g) {

  # Lists of variable names
  Cov_list = paste0("CovVal", 1:g)
  Pat_list = paste0("PatNum", 1:g)

  # Distributions
  for (i in 1:g) {

    CovVals = runif(S)
    PatVals = sample(100:1000,S)

    assign( Cov_list[i], CovVals )
    assign( Pat_list[i], PatVals )
  }


  # Combine into data frame
  df_Cov = do.call(cbind, mget(Cov_list))
  df_Pat = do.call(cbind, mget(Pat_list))
  df     = data.frame(cbind(df_Cov,df_Pat))

  # Add two more columns
  df["TotalPatNum"] = apply(df_Pat, 1, sum, na.rm = TRUE)
  df["Study"]       = 1:S

  return(df)
}

# This function generates a dataset under the alternative hypothesis, with the covariates being distributed according
# to a normal(j,1) in meta arm j

generate_data_H1 = function(S, g) {

  # Lists of variable names
  Cov_list = paste0("CovVal", 1:g)
  Pat_list = paste0("PatNum", 1:g)

  # Distributions
  for (i in 1:g) {

    CovVals = rnorm(S,mean=i,sd=1)
    PatVals = sample(100:1000,S)

    assign( Cov_list[i], CovVals )
    assign( Pat_list[i], PatVals )
  }

  # Combine into data frame
  df_Cov = do.call(cbind, mget(Cov_list))
  df_Pat = do.call(cbind, mget(Pat_list))
  df     = data.frame(cbind(df_Cov,df_Pat))

  # Add two more columns
  df["TotalPatNum"] = apply(df_Pat, 1, sum, na.rm = TRUE)
  df["Study"]       = 1:S

  return(df)
}

# This function creates the names and values of the ECDFs for a given dataset df.

create_ECDFs = function(df, g) {

  ECDF_names = paste0("ECDF", 1:g)
  ECDF_list = list()

  for (group in 1:g) {

    vals    = na.omit(df[, group])
    weights = na.omit(df[, (g + group)])

    ECDF_list[[ECDF_names[group]]] = ecdf(rep(vals,weights  ) )
  }
  return(list("List" = ECDF_list, "Names" = ECDF_names) )
}


# This function plots the ECDFs for a given dataset df.

plot_ECDFs = function(df, g, selected_groups= 1:g, lty = 1) {

  # Plotting ECDFs
  plot(NULL, xlab = "Covariate␣value␣(t)", ylab = "ECDF",  main="",
       xlim = c(min(df[,1:g],na.rm = TRUE),max(df[,1:g],na.rm = TRUE)), ylim = c(0,1) )

  ECDFs = create_ECDFs(df, g)

  colours = rainbow(g)
  for (i in selected_groups) {
    ECDF_func = ECDFs$List[[ECDFs$Names[i]]]
    lines(ECDF_func, verticals = TRUE, do.points = FALSE, col = colours[i], lty = lty)
  }
  legend(x = "topleft",
         legend = c(paste0("Meta-arm␣", 1:g)),
         lty = 1,
         col = colours,
         lwd = 2)
}


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

# This function determines the multi-sample test statistics in a particular dataset.

AreSimilar2 = function(df, g) {

  result    = data.frame()
  ECDF_vals = measures_ecdf(df, g)

  # Pairwise
  res_pairwise   = c()
  names_pairwise = paste0("ECDF",1:g)

  for (i in 1:(g-1)) {
    for (j in (i+1):g) {

      name = paste(paste0("ECDF", i), "vs", paste0("ECDF", j))

      res_pairwise = c(res_pairwise, wass_stat_ecdf(ECDF_vals,names_pairwise[i],names_pairwise[j]) )
    }
  }
  result["pairwise","statistic"] = max(res_pairwise)


  #Compared to measure
  measure_names = c("mean","median","joint")

  for (measure_name in measure_names) {

    res_measure    = c()

    for (j in 1:g) {
      name           = paste(paste0("ECDF", j), paste0("vs␣", measure_name))

      res_measure = c(res_measure, wass_stat_ecdf(ECDF_vals, measure_name, names_pairwise[j]))
    }
    result[measure_name,"statistic"] = max(res_measure)
  }


  # Min-Max
  result["Min␣vs␣Max","statistic"] = wass_stat_ecdf(ECDF_vals, "MIN", "MAX")


  return(result)
}


# This function resamples the dataset with replacement according to the permutation test in Section 3.2

resample_groups = function(df,g) {

  X = unlist(df[,1:g], use.names = FALSE)
  P = unlist(df[,(g+1):(g+g)], use.names = FALSE)

  groups = data.frame("X" = X, "P" = P)
  L      = length(X)

  new_indices = sample(1:L, nrow(df)*g, replace = TRUE)
  new_X       = X[new_indices]
  new_P       = P[new_indices]


  new_df = data.frame(split(new_X,1:g), split(new_P,1:g))
  new_df["TotalPatNum"] = apply(new_df[,(g+1):(g+g)],1,sum)
  new_df["Study"]       = 1:nrow(df)

  colnames(new_df) = c(paste0("CovVal",1:g),paste0("PatNum",1:g),"TotalPatNum","Study")

  return(new_df)
}

# This function computes the p value by the permutation test in Section 3.2

p_value_permutation = function(df, g) {

  T_observed = AreSimilar2(df,g)$statistic
  multistats = permutation_stats(df, g)

  counts     = sapply(1:5, function(k) {length(which(multistats[,k]>=T_observed[k]))})

  return(counts/nrow(multistats))

}

# This function gives the null distribution of the stats by the permutation test of Section 3.2

permutation_stats = function(df, g) {

  multistats = data.frame()

  N_boots = 500
  for (m in 1:N_boots) {

    set.seed(314+m)
    df_resampled = resample_groups(df,3)
    result       = AreSimilar2(df_resampled,3)
    multistats   = rbind(multistats, result$statistic)

  }
  colnames(multistats) = c("Pairwise","Mean","Median","Joint","Min-Max")

  return(multistats)
}

# This function gives the 100(1-alpha)% quantile of the null distribution of the stats by permutation test from #Section 3.2
permutation_quantiles = function(df, g, alpha = 0.05) {

  multistats = permutation_stats(df,g)

  quantiles = apply(multistats,2,function(vals) {quantile(vals,probs = 1-alpha)})
  return(quantiles)
}

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



# This function finds Fmean, Fmedian, Fjoint, Fmin and Fmax

measures_ecdf = function(df, g) {

  joint_sample = na.omit(unlist(df[,1:g], use.names = FALSE))
  joint_partic = na.omit(unlist(df[,(g+1):(g+g)], use.names = FALSE))
  joint        = rep(joint_sample, joint_partic)

  jointje = na.omit(c(apply(df[,1:g],1,c)))
  axis    = sort(unique((jointje)))

  # Calculate F1 to Fg
  ECDFs     = create_ECDFs(df, g)
  ECDF_vals = data.frame("Covariate␣Value" = axis)

  for (i in 1:g) {

    ECDF_vals[ECDFs$Names[i]] = ECDFs$List[[i]](axis)
  }

  # Determine F_measure
  ECDF_vals[,"mean"]   = apply(ECDF_vals[2:(1+g)], 1, mean)
  ECDF_vals[,"median"] = apply(ECDF_vals[2:(1+g)], 1, median)
  ECDF_vals[,"joint"]  = ecdf(joint)(axis)
  # Determine Fmin and Fmax
  ECDF_vals[,"MIN"]    = apply(ECDF_vals[2:(1+g)], 1, min)
  ECDF_vals[,"MAX"]     = apply(ECDF_vals[2:(1+g)], 1, max)


  return(ECDF_vals)
}


# This function computes the Wasserstein metric for any combination of MNDFs.
# Input "ECDF1", ..., "ECDFg", "mean", "median", "joint", "MIN" or "MAX" for a and b

wass_stat_ecdf = function(ECDF_vals, a = "ECDF1", b = "ECDF2") {

  axis = ECDF_vals$Covariate.Value

  stat = 0
  for (k in 1:(length(axis)-1)) {

    width = axis[k+1] - axis[k]
    height = ECDF_vals[k,a]-ECDF_vals[k,b]

    stat = stat + width * abs(height)
  }

  return(stat)
}

# This function computes Tpairwise

stat_pairwise = function(df,g) {

  joint_sample = unlist(df[,1:g], use.names = FALSE)

  jointje = na.omit(c(apply(df[,1:g],1,c)))
  axis    = sort(unique((jointje)))

  # Calculate F1 to Fg
  ECDFs     = create_ECDFs(df, g)
  ECDF_vals = data.frame("Covariate␣Value" = axis)

  for (i in 1:g) {

    ECDF_vals[ECDFs$Names[i]] = ECDFs$List[[i]](axis)
  }
  res             = c()
  for (i in 1:(g-1)) {
    for (j in (i+1):g) {

      res = c(res, wass_stat_ecdf(ECDF_vals,paste0("ECDF",i),paste0("ECDF",j)) )
    }
  }
  return( max(res) )


}

# This function computes Tmean

stat_mean = function(df,g) {

  joint_sample = unlist(df[,1:g], use.names = FALSE)

  jointje = na.omit(c(apply(df[,1:g],1,c)))
  axis    = sort(unique((jointje)))

  # Calculate F1 to Fg
  ECDFs     = create_ECDFs(df, g)
  ECDF_vals = data.frame("Covariate␣Value" = axis)

  for (i in 1:g) {

    ECDF_vals[ECDFs$Names[i]] = ECDFs$List[[i]](axis)
  }

  # Determine F_mean
  ECDF_vals[,"mean"]   = apply(ECDF_vals[2:(1+g)], 1, mean)

  res = c()

  for (j in 1:g) {
    res = c(res, wass_stat_ecdf(ECDF_vals, "mean", paste0("ECDF",j) ) )
  }
  return( max(res) )
}

# This function computes Tmedian

stat_median = function(df,g) {

  joint_sample = unlist(df[,1:g], use.names = FALSE)

  jointje = na.omit(c(apply(df[,1:g],1,c)))
  axis    = sort(unique((jointje)))

  # Calculate F1 to Fg
  ECDFs     = create_ECDFs(df, g)
  ECDF_vals = data.frame("Covariate␣Value" = axis)

  for (i in 1:g) {

    ECDF_vals[ECDFs$Names[i]] = ECDFs$List[[i]](axis)
  }

  # Determine F_mean
  ECDF_vals[,"median"]   = apply(ECDF_vals[2:(1+g)], 1, median)

  res = c()

  for (j in 1:g) {
    res = c(res, wass_stat_ecdf(ECDF_vals, "median", paste0("ECDF",j) ) )
  }
  return( max(res) )
}

# This function computes Tjoint

stat_joint = function(df,g) {

  joint_sample = unlist(df[,1:g], use.names = FALSE)

  jointje = na.omit(c(apply(df[,1:g],1,c)))
  axis    = sort(unique((jointje)))

  # Calculate F1 to Fg
  ECDFs     = create_ECDFs(df, g)
  ECDF_vals = data.frame("Covariate␣Value" = axis)

  for (i in 1:g) {

    ECDF_vals[ECDFs$Names[i]] = ECDFs$List[[i]](axis)
  }

  # Determine F_mean
  ECDF_vals[,"joint"]  = ecdf(joint_sample)(axis)

  res = c()

  for (j in 1:g) {
    res = c(res, wass_stat_ecdf(ECDF_vals, "joint", paste0("ECDF",j) ) )
  }
  return( max(res) )
}

```

```
# This function computes Tminmax

stat_minmax = function(df,g) {

  joint_sample = unlist(df[,1:g], use.names = FALSE)

  jointje = na.omit(c(apply(df[,1:g],1,c)))
  axis    = sort(unique((jointje)))

  # Calculate F1 to Fg
  ECDFs     = create_ECDFs(df, g)
  ECDF_vals = data.frame("Covariate␣Value" = axis)

  for (i in 1:g) {

    ECDF_vals[ECDFs$Names[i]] = ECDFs$List[[i]](axis)
  }

  # Determine F_mean
  ECDF_vals[,"MIN"]    = apply(ECDF_vals[2:(1+g)], 1, min)
  ECDF_vals[,"MAX"]    = apply(ECDF_vals[2:(1+g)], 1, max)

  res = wass_stat_ecdf(ECDF_vals, "MIN","MAX" )

  return( res )
}



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


# This function checks if each meta-arm has at least one group left.
all_groups_have_data  = function(df,g) {

  counts_NA      = apply(df[,1:g], 2, function(col) {sum(is.na(col))} )
  groups_not_NA  = nrow(df)-counts_NA > 1
  res            = floor(sum(groups_not_NA)/g)

  return(res == 1)
}

# This function is the balancing procedure and repeatedly discards group until the statistic is below threshold = quantile of null distribution from permutation test
# Set stop = FALSE to turn off stopping condition
# Set p = TRUE to receive p values at stopping point
# Set stats = TRUE to receive removed groups and corresponding stats at each iteration
# Set print = TRUE to plot the ECDFs of meta-arms at stopping point
# Input "stat_pairwise", "stat_mean", "stat_median", "stat_joint" or "stat_minmax" for quant

remove_groups = function(df, g, threshold, stop = TRUE, quant = "stat_minmax", print = FALSE, p = FALSE, stats = FALSE) {

  result = data.frame("study"     = NA,
                      "group"     = NA,
                      "statistic" = get(quant)(df,g),
                      "removed"   = 0                 )


  df_remain = df
  df_min    = df
  min_stat  = 10^10

  while ( all_groups_have_data(df_remain,g) ) {

    new_row = RemoveGroupBrute(df_remain, g, quant)

    df_temp                                                         = df_remain
    df_temp[which(df_temp$Study==new_row$study), new_row$group]     = NA
    df_temp[which(df_temp$Study==new_row$study), (g+new_row$group)] = NA
    new_row["removed"] = tail(result$removed,1) + 1
```

```r


    df_remain = df_temp
    result    = rbind(result, new_row)
    if ( stop & new_row$statistic < threshold ) {break}

    if ( new_row$statistic < min_stat ) {
      df_min   = df_temp
      min_stat = new_row$statistic
    }

  }

  if (print) {plot_ECDFs(df_min, g)}
  if (p) {print(p_value_permutation(df_min,g))}
  if (stats) {print(AreSimilar2(df_min,g))}

  return(result)
}



# This function determines which group to exclude in each iteration.

RemoveGroupBrute = function(df ,g, quant) {

  lowest_stat = 10^10

  for ( study in 1:nrow(df) ) {
    for ( group in 1:g ) {

      if ( is.na(df[study,group]) ) {
        next
      }

      df_temp                   = df
      df_temp[study, group]     = NA
      df_temp[study, (g+group)] = NA

      stat = get(quant)(df_temp,g)

      if ( stat < lowest_stat ) {
        best_group  = group
        corr_study  = df$Study[study]
        lowest_stat = stat
      }
    }
  }
  result = list("study"     = corr_study,
                "group"     = best_group,
                "statistic" = lowest_stat)
  return(result)
}

# This function standardises the result of the balancing procedure

standardise_group_result = function(result, n, g) {

  initial_stat     = result$statistic[1]
  result$statistic = result$statistic/initial_stat
  result$removed   = result$removed/(n*g)*100


  return(result)
}



# This function returns the overlap between two sets of groups

overlap_checker = function(groups1, groups2) {

  # Remove first NA NA
  groups1 = groups1[2:length(groups1)]
  groups2 = groups2[2:length(groups2)]


  intersection = intersect(groups1, groups2)

  return(length(intersection))
}
```

A.2. Permutation, simulations and graphs

This part of the source code contains the code producing the ECDF graphs in Chapter 3, performing the permutation test of Section 3.2 and the balancing procedure and graphs of Chapter 4 using the functions of Section A.1.

```

# This part plots the simple example highlighting the reason for why the Wasserstein distance is used.

source("functions.R")

set.seed(314)
S = 25
g = 3

df = generate_data_H1(S, g)

plot_ECDFs(df, g)

black = c(10,20,30)
blue =  c( 9,19,29)
red  =  c( 5,15,25)

plot(ecdf(black),col="black", verticals = TRUE, do.points = F,lwd = 2,main="",xlab="Covariate␣value␣(t)",ylab="F(t)")
lines(ecdf(red) , col="red" ,verticals = TRUE, do.points = F, lwd = 2, lty = 2)
lines(ecdf(blue), col="blue",verticals = TRUE, do.points = F, lwd = 2, lty = 2)

legend( x = "topleft",
        legend = c("F","G1","G2"),
        lty = c(1,2,2),
        col = c("black","red","blue"),
        lwd = 3 )

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

# This part is used to plot the ECDFs and MNDFs in the thesis.


set.seed(314)
S = 25
g = 3

df = generate_data_H1(S,g)



plot_ECDFs(df,g)
names = c("Mean","Median","Joint", "Min","Max")
x = measures_ecdf(df,g)

#lines(x$Covariate.Value,x$mean,   type = "s", col = "black",    lwd = 3, lty = 2)
#lines(x$Covariate.Value,x$median, type = "s", col = "black", lwd = 3, lty = 2)
#lines(x$Covariate.Value,x$joint,  type = "s", col = "black",   lwd = 3, lty = 2)
lines(x$Covariate.Value,x$MIN,  type = "s", col = "black", lwd = 3, lty = 2)
lines(x$Covariate.Value,x$MAX,  type = "s", col = "black", lwd = 3, lty = 2)

legend(x = "topleft",
         legend = c(paste0("Meta-arm␣", 1:g),names[4:5]),
         lty = c(rep(1,g),rep(2,2)),
         col = c(rainbow(g),rep("black",2)),
         lwd = 2)


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

# This part determines the table and plot of the permutation test in Section 3.2

set.seed(314)
df = generate_data_H1(25,3)

multistats = data.frame()


M = 10^3
for (m in 1:M) {

  set.seed(314+m)
  df_resampled = resample_groups(df,3)
  result       = AreSimilar2(df_resampled,3)
  multistats   = rbind(multistats, result$statistic)

}
colnames(multistats) = c("Pairwise","Mean","Median","Joint","Min-Max")


names_lower = c("pairwise","mean","median","joint","min-max")
for (k in 1:5) {

  hist(multistats[,k], 20, col = colors[k],xlab = "T",
       main=paste0("Histogram␣of␣T",names_lower[k], "␣in␣Permutation␣test"))
  val = quantile(multistats[,k],probs = .95)
  abline(v = val, col="red")
}

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

# This part is used to apply the balancing procedure to the example in Section 4.1

N = 100
g = 3
n = 25

set.seed(314)
df = generate_data_H1(25, 3)


quants = c("stat_pairwise","stat_mean","stat_median","stat_joint","stat_minmax")


plot(NULL, xlim = c(0,30), ylim = c(0.3,1),
     xlab = "Percentage␣of␣groups␣discarded␣(%)", ylab = "Standardised␣Multi-Sample␣Test␣Statistic␣(T)",
     main=paste0("Balancing␣Procedure␣with␣Stopping␣Condition"))
abline(h=1, col="red",lty=3)

stats_5       = c()
axis_5        = c()
min_removed_5 = c()
min_stat_5    = c()

quantiles95 = permutation_quantiles(df,g)

for (k in 1:5) {

  result       = remove_groups(df, 3, quantiles95[k], stop = TRUE, quant = quants[k])
  standardised = standardise_group_result(result, 25, 3)

  min_loc      = which.min(standardised$statistic)
  min_row      = standardised[min_loc,]
  last_row     = tail(standardised, n = 1)

  stats_5       = c( stats_5, standardised$statistic )
  axis_5        = c( axis_5 , standardised$removed   )
  min_removed_5 = c( min_removed_5, min_row$removed        )
  min_stat_5    = c( min_stat_5, min_row$statistic      )


  groups = paste0(standardised[1:min_loc,1],"␣",standardised[1:min_loc,2])
  assign(paste0("groups_",quants[k],"_example"), groups)

  lines(standardised$removed, standardised$statistic, col=colors[k],type="l",lwd=2)
  lines(last_row$removed,last_row$statistic, type = "p", col = colors[k],lwd=6)
}
legend(x = "topright",
       legend = c(names),
       lty = 1,
       col = colors,
       lwd = 3 )


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

# This part generates the datasets of the simulation for g = 3 and g = 4 of Section 4.2

for (index in 1:N) {
  set.seed(314*index)
  dfi = generate_data_H1(25,3)
  assign(paste0("df_",index),dfi)
}

for (index in 1:N) {
  set.seed(314*index)
  dfi = generate_data_H1(25,4)
  assign(paste0("df4_",index),dfi)
}

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

# This part performs the simulation in Section 4.2
# It is better to do this for each k = 1,2,3,4,5 separately,
#since it takes some time to run.

# Change g to 3, and df_4 to df_ t obtain the results for g = 3

g = 4
for (k in 1:5) {

  quant = quants[k]

  stats = c()
  axis = c()
  min_removed  = c()
  min_stat = c()

  for (index in 1:N) {

    print(index)

    data        = get(paste0("df4_",index))
    quantiles95 = permutation_quantiles(data,g)
    quant       = quants[k]

    result       = remove_groups(data, g, quantiles95[k], stop = TRUE, quant = quants[k])
    standardised = standardise_group_result(result, 25, g)

    min_loc      = which.min(standardised$statistic)
    min_row      = standardised[min_loc,]

    stats       = c( stats, standardised$statistic )
    axis        = c( axis , standardised$removed   )
    min_removed = c( min_removed, min_row$removed        )
    min_stat    = c( min_stat, min_row$statistic      )



    groups = paste0(standardised[1:min_loc,1],"␣",standardised[1:min_loc,2])
    assign(paste0("groups4_",quant,index), groups)

    #lines(standardised$removed, standardised$statistic, col=colors[k],type="l",lwd=2)
    #lines(min_row$removed,min_row$statistic, type = "p", col = colors[k],lwd=5)

  }
  assign(paste0("stats4_",quant),stats)
  assign(paste0("axis4_" ,quant), axis)
  assign(paste0("min_sta4t_",quant),min_stat)
  assign(paste0("min_removed4_",quant),min_removed)

}

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

# This part plots the results of the simulation

trans = 1/2
colors = c( rgb(0,   1,0, trans),
            rgb(1,   0,0, trans),
            rgb(1,0.65,0, trans),
            rgb(1,0.84,0, trans),
            rgb(0,   1,1, trans) )


names = c("Pairwise","Mean","Median","Joint","Min-Max")
for (i in 1:5 ) {

  plot(NULL, xlim = c(0,40), ylim = c(0.3,1),
       xlab = "Percentage␣of␣groups␣discarded␣(%)", ylab = "Multi-Sample␣Test␣Statistic␣(T)",
       main=paste0("Balancing␣Procedure␣for␣",names[i], "␣Statistic␣"))
  abline(h=1, col="red",lty=3)

  quant = quants[i]

  stats       = get(paste0("stats_"        ,quant))
  axis        = get(paste0("axis_"         ,quant))
  min_removed = get(paste0("min_removed_"  ,quant))
  min_stat    = get(paste0("min_stat_"     ,quant))

  # Adding mean line
  #mean_line = aggregate(stats ~ axis,  FUN = mean)
  #lines(mean_line$axis, mean_line$stats, col = colors[i], lwd = 2)
  splitted_indices = which(axis == 0)
  for ( j in 1:(length(splitted_indices)-1) ) {

    line_stats = stats[splitted_indices[j]:(splitted_indices[j+1]-1)]
    line_axis  =  axis[splitted_indices[j]:(splitted_indices[j+1]-1)]

    lines(line_axis, line_stats, col = colors[i], lwd = 1)
  }
  j = length(splitted_indices)
  line_stats = stats[splitted_indices[j]:length(stats) ]
  line_axis  =  axis[splitted_indices[j]:length(axis)  ]

  lines(line_axis, line_stats, col = colors[i], lwd = 1)


  #Adding "CI"
  #x_axis = 0:max(stops)
  #CI     = aggregate(stats ~ axis, FUN = function(val){quantile(val,probs = c(.025,.975))})
  #polygon(c(x_axis, rev(x_axis)),
          #c(CI[,2][,1], rev(CI[,2][,2])),
          #col = colors_CI95[i], border = NA)
  #CI50     = aggregate(stats ~ axis, FUN = function(val){quantile(val,probs = c(.25,.75))})
  #polygon(c(x_axis, rev(x_axis)),
          #c(CI50[,2][,1], rev(CI50[,2][,2])),
          #col = colors_CI95[i], border = NA)

  #Adding stopping points
  lines(min_removed, min_stat, type = "p", col = colors[i], lwd = 2)

  #hist(stops, col = colors[i], breaks = 10)
  #abline(v = mean(stops),col="black",lty = 2)

  #length(which(final <  1) )/N

  legend(x = "topright",
         legend = c("sample","stopping␣point"),
         lty = c(1,NA),
         pch=c(NA,1), merge=FALSE,
         col = c(colors[i],colors[i]),
         lwd = c(3, 2) )
}

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



# This part makes the table of the values of the multi-sample test stats and discarded percentage of groups in the simulation at the stopping points.

mat = matrix(NA, nrow = 5, ncol = 4)

for (k in 1:5) {

  quant = quants[k]

  stops    = get(paste0("min_removed_"  ,quant))
  final    = get(paste0("min_stat_"     ,quant))

  95_stops = round(quantile(stops, probs =c(.025,.975)),digits = 1)
  95_final = round(100*quantile(final, probs =c(.025,.975)),digits = 1)

  mat[k,] = c(round(mean(stops),digits = 1),
              paste0("[",95_stops[1],",␣",95_stops[2],"]"),

              round(100-mean(100*final),digits = 3),
              paste0("[",100-95_final[2],",␣",100-95_final[1],"]") )

}
mat

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

# This part determines the overlap between the balancing procedures using different multisamp test stats

overlapping           = data.frame(matrix("",nrow=5,ncol=5))
colnames(overlapping) = names
rownames(overlapping) = names

# Checking overlap
for (k1 in 1:5) {
  for (k2 in k1:5) {

    overlaps = c()

    stat1 = paste0("groups_",quants[k1])
    stat2 = paste0("groups_",quants[k2])

    #stops1 = get(paste0("min_removed_",quants[k1]))
    #stops2 = get(paste0("min_removed__",quants[k2]))

    for (index in 1:N) {


      groups1_index = get(paste0(stat1,index))
      groups2_index = get(paste0(stat2,index))



      val          = overlap_checker(groups1_index, groups2_index)
      overlaps     = c(overlaps, val)
    }
    mean_overlap = mean(overlaps)

    overlapping[k1,k2] = round(mean_overlap, digits = 1)
  }
}
overlapping

##############################################################################
```