

Turn up the Heat! A Tutorial for the MAGMA R-package.

Julian Urban, Markus Feuchter, Vsevolod Scherrer, Moritz Breit, Franzis Preckel

Summary

Balancing quasi-experimental field research for effects of covariates is fundamental for drawing causal inference. Propensity Score Matching deals with this issue but current techniques are restricted to binary treatment variables. Moreover, they provide several solutions without providing a comprehensive framework on choosing the best model. The MAGMA-package addresses these restrictions by offering nearest neighbor matching for two to four groups. It also includes the option to match data of a 2x2 design. In addition, MAGMA includes a framework for evaluating the post-matching balance. This vignette is a tutorial on MAGMA. It demonstrates the main MAGMA functions on an included simulated dataset. In total, this vignette includes a two-group, a three-group, and a 2x2-design which is approximately equivalent to a four-group example. These three examples are conducted in a standard matching process, as well as with exact matching.

Introduction

Field research as well as observational or non-randomized studies have to deal with a lack of experimental control. One method to control post-hoc for effects of covariates is propensity score matching (PSM; Rosenbaum & Rubin, 1983, 1985). By balancing for the effects of covariates PSM facilitates causal inference. Since causal inference is fundamental in science, PSM is used in many scientific fields such as in organizational (e.g., Li, 2013), educational (e.g., Powell et al., 2020), social science (e.g., Thoemmes & Kim, 2011), and medical research (e.g., Austin, 2014).

In PSM we build statistical pairs of individuals of – as example – a treatment group and a control group based on a distance in a specific variable – the propensity score (for a review, see Stuart, 2010). The propensity score itself expresses the conditional probability of belonging to a group (e.g., the treatment group) given a set of covariates (Rosenbaum & Rubin, 1985). Although a plethora of matching techniques exist, a commonly applied method is nearest neighbor matching (Austin, 2014; Jacovidis, 2017). Thus, by matching a treated individual to its non-treated nearest neighbor PSM controls for the effects of covariates.

The usage of PSM is, nonetheless, restricted due to the following three reasons. First, current techniques are restricted to binary treatment variables (for first approaches, see Imai & van Dyk, 2004; McCaffrey et al., 2013). Second, researchers must successively extract and compare several matching solutions by hand, instead of having access to a full unambiguous display. Third, there is no comprehensive framework for evaluating the quality of matching solution, especially when multiple groups are involved. We addressed these issues with our MAGMA package. For an extensive overview of MAGMA and its balance framework see Preregistration (Feuchter et al., 2023).

The aims of this vignette were to provide a tutorial for the MAGMA R-package. This includes examples for all currently available group scenarios, namely two, three, and four groups as well as 2x2 designs. We conducted these examples for standard and exact matching. Moreover, this tutorial contains examples on how to report matching relevant statistics. Lastly, we show how to compute and report balance of PSMs using our developed framework.

The MAGMA Simulated Dataset

We used a simulated data set for all examples. This simulated dataset contains 800 cases and 14 variables. It is available after installing the MAGMA package. Associations between the variables were modeled to increase the usability and comprehensibility of this tutorial, and are, therefore, not representative of real-life phenomena. For information about this simulated dataset use `?MAGMA_sim_data`. Below, is a brief look at the data and its structure. Note that the last three variables are propensity scores estimated using the `twang` package (Ridgeway et al., 2015). For a detailed overview and tutorial on `twang`, see (Ridgeway et al., 2015). The propensity scores serve as distance indicators for all examples.

```
str(MAGMA_sim_data)
#> 'data.frame': 800 obs. of 14 variables:
#> $ ID              : int 1 2 3 4 5 6 7 8 9 10 ...
#> $ sex             : int 1 0 1 1 1 0 0 0 1 0 ...
#> $ gifted_support  : int 1 1 1 1 1 1 0 1 0 1 ...
#> $ teacher_ability_rating: int 3 2 2 1 3 3 2 2 1 1 ...
#> $ enrichment       : int 0 1 0 1 0 0 0 0 0 1 ...
#> $ parents_academic: int 1 1 1 0 1 0 0 1 0 0 ...
#> $ GPA_school       : num 4.73 4.41 3.23 2.88 2.98 ...
#> $ IQ_score          : num 111.6 97.8 120.1 97.9 74.4 ...
#> $ Motivation        : num 5.36 4.41 3.57 4.34 3.54 ...
#> $ college_GPA       : num 4.01 3.41 2.73 4.02 4.09 ...
#> $ support_enrichment: int 3 4 3 4 3 3 1 3 1 4 ...
#> $ ps_tar            : num 0.0657 0.1085 0.1899 0.4261 0.4782 ...
#> $ ps_2x2             : num 0.3 0.32 0.21 0.237 0.571 ...
#> $ ps_gifted          : num 0.431 0.364 0.626 0.353 0.187 ...
```

Two-Group Example

This is a fictitious two-group example. The independent variable (i.e., the treatment) is `gifted_support`. We are interested in how receiving giftedness support affects the college GPA. However, other variables, such as intelligence are associated with both giftedness support and college GPA. This is because gifted students were chosen to participate in this support program based on their previous achievement or intelligence test results. Therefore, we aim to reduce the effects of these covariates by matching. For this example, the covariates are *high school GPA*, *intelligence*, *motivation*, *sex*, and whether at least one parent has an *academic background*. Note that MAGMA can only consider metric and binary variables for balance estimation.

```
covariates_gifted <- c("GPA_school",
                         "IQ_score",
                         "Motivation",
                         "parents_academic",
                         "sex")
```

With this definition, we can estimate descriptive statistics, standardized mean differences, and initial unbalance over all cases in our dataset. To do this we can use the functions `MAGMA_desc` and `initial_unbalance`. The first function – `MAGMA_desc` – computes sample size, mean, and standard deviation for the overall sample and the two groups. Moreover, it estimates the standardized mean difference (i.e., Cohen's d) between the two groups. This function needs the dataset (`MAGMA_sim_data`), the grouping variable (`gifted_support`), and all binary and metric covariates of interest (`covariates_gifted`). The function results in a table that contains these statistics for each covariate.

The second function – `initial_unbalance` – estimates the four balance criteria of our balance framework. These criteria are *Pillai's Trace*, *d-ratio*, *mean g*, and *adjusted d-ratio*. For more information, see Feuchter et al. (2022). The function `initial_unbalance` indicates the unbalance in the sample before matching.

Therefore, it is necessary to define the dataset (`MAGMA_sim_data`), the grouping variable (`gifted_support`), and all binary and metric covariates of interest (`covariates_gifted`). This results in a vector of length four - the four balance criteria.

```
#Estimate overall and group specific descriptive statistics and Cohen's d
descs_gifted_pre <- MAGMA_desc(Data = MAGMA_sim_data,
                                 group = "gifted_support",
                                 covariates = covariates_gifted,
                                 filename = "stats_gifted_pre.docx")

descs_gifted_pre %>%
  purrr::set_names(c("Overall N", "Overall Mean", "Overall SD",
                    "No Support N", "No Support Mean", "No Support SD",
                    "Support N", "Support Mean", "Support SD",
                    "d"))

#>          Overall N Overall Mean Overall SD No Support N No Support Mean
#> gifted_support      800     0.38     0.49      495     0.00
#> GPA_school          800     3.29     0.93      495     3.13
#> IQ_score             800    100.51    15.00      495    96.32
#> Motivation           800     3.82     0.89      495     3.81
#> parents_academic    800     0.40     0.49      495     0.35
#> sex                  800     0.50     0.50      495     0.50
#>          No Support SD Support N Support Mean Support SD      d
#> gifted_support        0.00     305     1.00     0.00 -Inf
#> GPA_school            0.91     305     3.56     0.89 -0.48
#> IQ_score              14.20    305    107.30    13.75 -0.78
#> Motivation            0.90     305     3.83     0.88 -0.02
#> parents_academic     0.48     305     0.48     0.50 -0.27
#> sex                   0.50     305     0.49     0.50   0.02

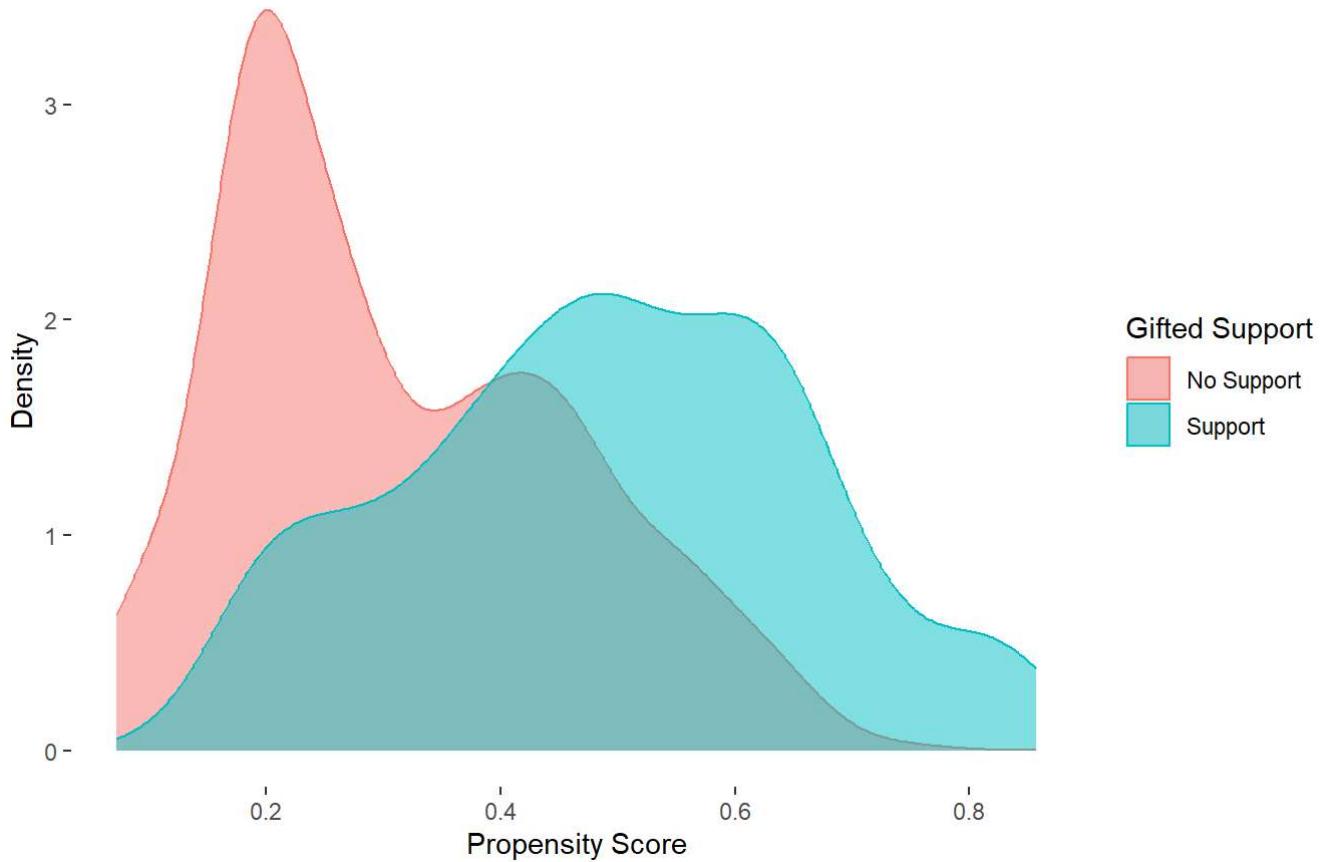
#Estimating the four balance criteria
unbalance_gifted <- initial_unbalance(Data = MAGMA_sim_data,
                                         group = "gifted_support",
                                         covariates = covariates_gifted)

#> [1] "mean g was computed using random effects meta-analysis with metafor."
unbalance_gifted
#>          Pillai's Trace d-ratio Mean g adj. d-ratio
#> Unbalance       0.13     0.4    0.31        0.4
```

As may be expected from the descriptive statistics, the cases that received giftedness support differed substantially from the cases that did not receive giftedness support on at least some variables. Before we start matching, we check the possibility of eligible matches. Therefore, we estimate and plot the common support in propensity score via the kernel-density (Pastore et al., 2022) and the function `density_overlap`. This function requires the dataset (`MAGMA_sim_data`), the variable for that the overlap should be computed (`ps_gifted`), and the group variable (`gifted_support`). The further arguments of the functions define aesthetics of the plot, namely the name of the variable (`variable_name`), the name of the group (`group_name`), and the labels of the respective groups (`group_labels`).

```
#Density overlap in Propensity scores for gifted before matching
Density_overlap(Data = MAGMA_sim_data,
                 variable = "ps_gifted",
                 group = "gifted_support",
                 variable_name = "Propensity Score",
                 group_labels = c("No Support", "Support"),
                 group_name = "Gifted Support")
```

Density Plot for Propensity Score



```
#>      OV  
#> 0.6162573
```

As we can see, there is substantial common support between the two groups. After these initial checks, we can conduct the matching. This process is based on the respective propensity score, `ps_gifted`. We first address standard matching and then with exact matching.

Standard Matching

For standard nearest neighbor matching we use the function `MAGMA`. This function has four arguments. The first argument specifies the dataset (`MAGMA_sim_data`). The second argument specifies the name of the grouping variable as a character (`"gifted_support"`). The third argument specifies the name of the distance variable/propensity score as a character (`"ps_gifted"`). The fourth argument is optional. Since MAGMA matching has some computational load, it includes the option of parallel computation. By default, only one CPU core is used. The usage of more cores may reduce the computation time of matching. While this reduction is of only restricted usage in two-group matching, it becomes more helpful for more group matching. If parallel computation is desired, an integer larger than one can be specified for the `cores` argument. If the specified value exceeds the number of available cores, MAGMA will set this argument to the maximum number of available cores.

Applying this function results in a dataset that extends the original dataset by three variables. The first, `weight`, indicates whether this case was matched or not. The second, `step`, indicates the iterative step in which a case was matched. More specifically, the cases with the smallest distance are matched first and, thus, receive `1` in the `step` variable. The matched cases with the second smallest distance receive the `2` and so on. The third variable, `distance` indicates the distance of the matched cases.

```

#Conducting matching for gifted support
MAGMA_sim_data_gifted <- MAGMA(Data = MAGMA_sim_data,
                                    group = "gifted_support",
                                    dist = "ps_gifted",
                                    cores = 2)

#> input correctly identified
#> Distance computation finished. Starting matching.
#> matching complete!
str(MAGMA_sim_data_gifted)
#> 'data.frame': 800 obs. of 17 variables:
#> $ ID : int 1 2 3 4 5 6 7 8 9 10 ...
#> $ sex : int 1 0 1 1 1 0 0 0 1 0 ...
#> $ gifted_support : int 1 1 1 1 1 0 1 0 1 ...
#> $ teacher_ability_rating: int 3 2 2 1 3 3 2 2 1 1 ...
#> $ enrichment : int 0 1 0 1 0 0 0 0 0 1 ...
#> $ parents_academic : int 1 1 1 0 1 0 0 1 0 0 ...
#> $ GPA_school : num 4.73 4.41 3.23 2.88 2.98 ...
#> $ IQ_score : num 111.6 97.8 120.1 97.9 74.4 ...
#> $ Motivation : num 5.36 4.41 3.57 4.34 3.54 ...
#> $ college_GPA : num 4.01 3.41 2.73 4.02 4.09 ...
#> $ support_enrichment : int 3 4 3 4 3 3 1 3 1 4 ...
#> $ ps_tar : num 0.0657 0.1085 0.1899 0.4261 0.4782 ...
#> $ ps_2x2 : num 0.3 0.32 0.21 0.237 0.571 ...
#> $ ps_gifted : num 0.431 0.364 0.626 0.353 0.187 ...
#> $ step : num 56 140 249 117 52 231 266 304 NA 29 ...
#> $ weight : num 1 1 1 1 1 1 1 NA 1 ...
#> $ distance : num 4.55e-08 3.88e-06 5.06e-01 1.43e-06 3.27e-08 ...

```

Balance estimation

MAGMA matches cases iteratively until all cases of the smallest group are matched. Thus, it is necessary to find the optimal sample size after matching. To do so, `Balance_MAGMA` estimates the four balance criteria for each possible sample size. Note that MAGMA uses a lower limit of at least $n = 20$ per group. To use the function, you need to specify at least your dataset (`MAGMA_sim_data_gifted`), the name of the grouping variable as a character (`"gifted_support"`), and your covariates as a character vector (`covariates_gifted`). Moreover, `Balance_MAGMA` needs the name of the variable that indicates the iterative `step` in which a case was matched. Since the `MAGMA` function names this variable `"step"` this name is the default of the `step` argument.

`Balance_MAGMA` returns a list of length four. In the case of a univariate grouping variable, this list contains three numeric vectors and one list of length two. The three vectors comprise Pillai's Trace, mean g , and adjusted d -ratio over the iteratively increased sample size. The list of length two includes one numeric vector, d -ratio, and one matrix. The matrix gives all pairwise d 's for all covariates. Both d -ratio and pairwise d 's are also estimated iteratively over the increased sample size. The example below estimates the balance for the matching and shows the balance criteria and pairwise effects for a group sample size of $n = 100$.

This function is augmented by the function `Balance_extract`. This function extracts the Balance criteria ('effects = FALSE') or the pairwise effects (effects = TRUE) for a specific sample size per group (`samplesize`) from the results of `Balance_MAGMA`. It returns a vector containing the balance criteria or the pairwise effects respectively.

```

#Estimating the four balance criteria iteratively over possible sample sizes
Balance_gifted <- Balance_MAGMA(Data = MAGMA_sim_data_gifted,
                                    group = "gifted_support",
                                    covariates = covariates_gifted,
                                    step = "step")
#> Start estimating Pillai's Trace.
#> Pillai's Trace finished. Starting to compute d-ratio.
#> d-ratio finished. Starting to compute mean-g.
#> [1] "mean g was computed using random effects meta-analysis with metafor."
#> mean-g finished. Starting to compute adjusted d-ratio.
#> finished balance estimation

#Extracting balance criteria for 100 cases per group
Balance_100_criteria <- Balance_extract(Balance = Balance_gifted,
                                         samplesize = 100,
                                         effects = FALSE)
Balance_100_criteria
#> Pillai's Trace      d-ratio      mean g   adj. d-ratio
#>          0.02        1.00        0.10        0.74

#Extracting pairwise effects for 100 cases per group
Balance_100_effects <- Balance_extract(Balance = Balance_gifted,
                                         samplesize = 100,
                                         effects = TRUE)
Balance_100_effects
#>      GPA_school      IQ_score      Motivation_parents_academic
#>          0.07        0.20        0.01        0.12
#>      sex
#>          0.10

```

Balance Evaluation

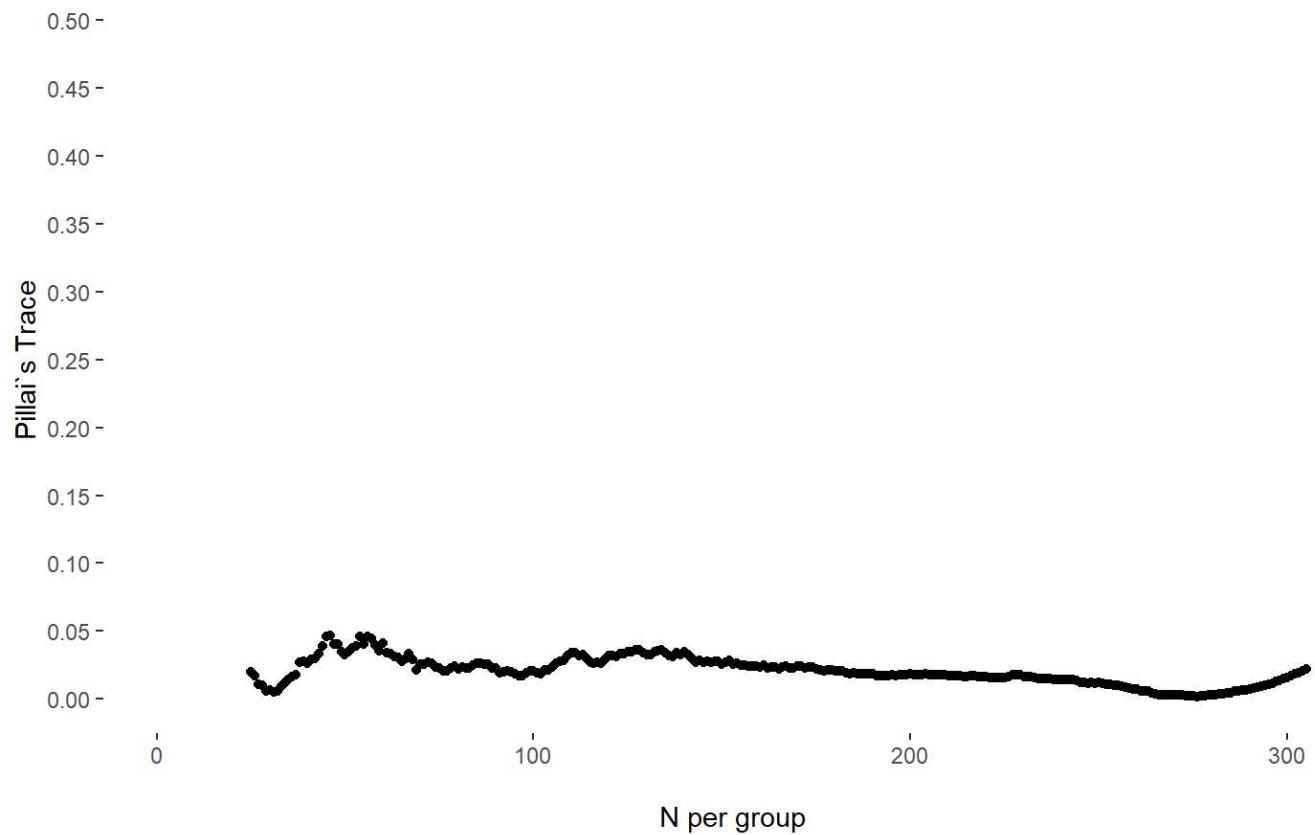
After balance estimation the “optimal” sample size to extract remains veiled. Which sample size is “optimal” strongly depends on the specific research aim. For example, the sample size requirements of some statistical procedures may narrow the selection, or as may the need to eliminate a specific pairwise effect. We therefore encourage to adapt the selection of optimal sample size to the specific research aim.

A first indicator of such an optimal model is the trend of the balance criteria across the iteratively increased sample size. The function `Plot_MAGMA` visualizes this trend. The one mandatory argument, `Balance`, specifies the object of the balance estimation results. Note that this object must be the result of a `Balance_MAGMA` function. Per default, `Plot_MAGMA` creates plots for all four balance criteria. However, the `criterion` argument enables the plotting of specific balance criteria. Note that running this code will result in a warning. This warning is a result of the specified minimum sample size for balance estimation and can be ignored.

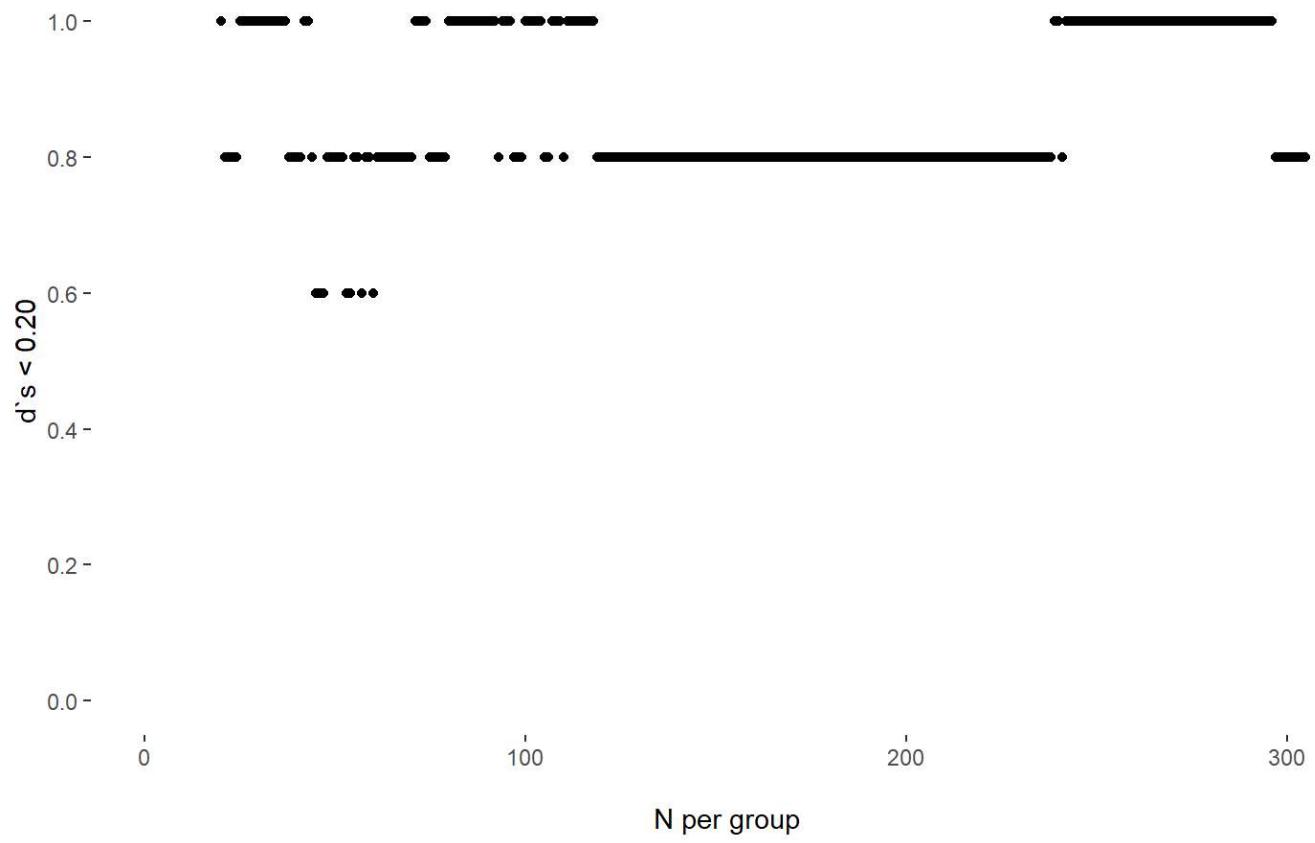
The trend of these plots depends strongly on the data. Nonetheless, they can be a good indicator of breakpoint in the sample size-balance associations. `Plot_MAGMA` uses fixed limits for the y-axis. This ensures comparability of these plots and prevents misinterpretation due to a restricted y-axis range. Both *d*-ratio and adjusted *d*-ratio vary between 0 and 1. Although Pillai’s Trace has theoretically the same range, we set the limits to 0 and 0.5. Larger values are unlikely and indicate a poor balance. A stricter limit might restrict a comprehensive trend evaluation. For mean *g* we set the limits to 0 and 1. The lower limit 0 corresponds to the theoretical lower limit. The rationale for the upper limit is comparable to that of Pillai’s Trace.

```
#Plotting balance trend over sample size
Plot_MAGMA(Balance = Balance_gifted,
            criterion = c("Pillai", "d_ratio", "mean_g", "Adj_d_ratio"))
```

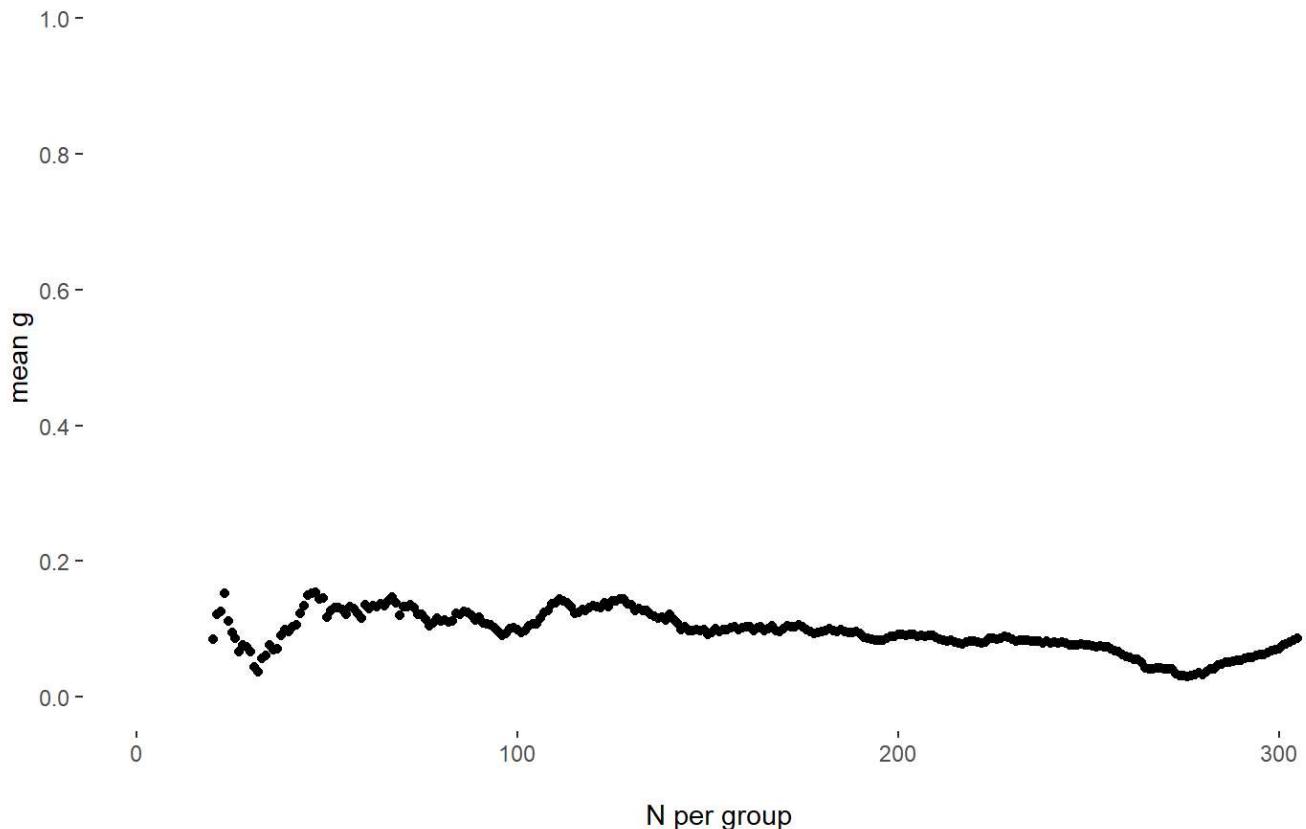
Pillai's Trace values for different sample sizes



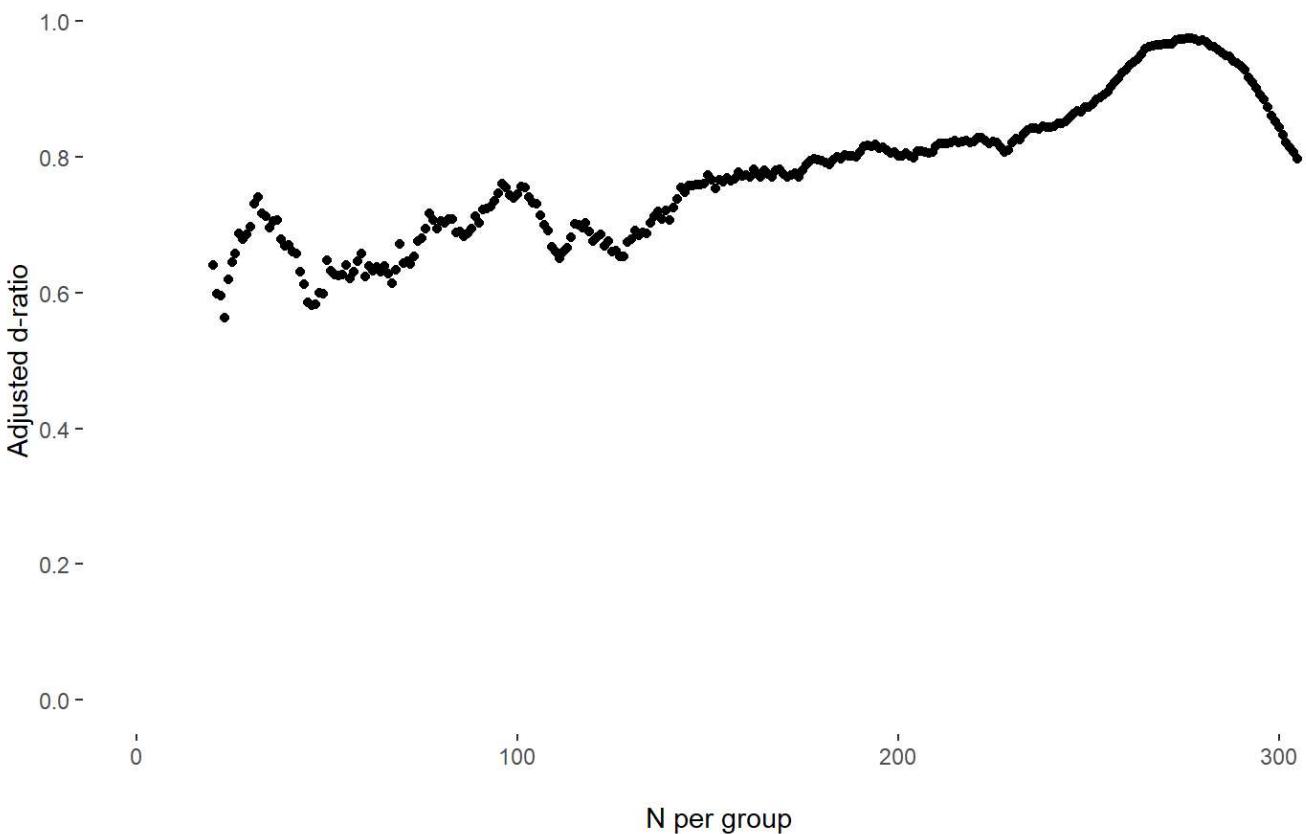
Cohen's d < .20` for different sample sizes



Mean effect for different sample sizes



Adjusted d-ratio for different sample sizes



Besides the trend over sample size, the absolutely best values for each criterion are of interest. The function `Table_MAGMA` extracts these values and returns them in a table in an extra file. More specifically it returns a 4x5 table. In each of the four rows one balance criterion has its optimal value. The five columns present the sample size per group for the matching solution where the respective criterion is optimal, and the four balance criteria values for this sample size per group. In addition to creating a file with a table, the table is printed to the console, too.

Similar to `Plot_MAGMA` the first argument `Balance` of `Table-MAGMA` specifies the object of the balance estimation results. The second argument `filename` specifies the desired name of the extra file containing the table. This argument is a character.

```
Table_MAGMA(Balance = Balance_gifted,
             filename = "Balance_gifted.docx")
#> Balance Table successfully created!
#> # A tibble: 4 × 6
#>   Criterion_optimized Pillai_Trace d_ratio mean_g adjusted_d_ratio n_per_group
#>   <chr>                <dbl>    <dbl>    <dbl>    <dbl>      <int>
#> 1 Best Pillai            0        1     0.03     0.97      276
#> 2 Best mean g           0        1     0.03     0.97      276
#> 3 adjusted d-ratio       0        1     0.03     0.97      276
#> 4 Best d-ratio          0.01    1     0.06     0.88      296
```

Post-matching statistics

After the decision for a final matching model, we suggest reporting the descriptive statistics and pairwise effects of the selected model. To do this, we can use the same function as we used for reporting pre-matching statistics (`MAGMA_desc`). Additional to the parameters we specified above (`Data`, `covariates`, and `group`), we need to name the step variable (`step_var`) and final sample size per group (`step_num`). As above, this function results in a table with descriptive statistics and pairwise effects (Cohen's d) of all covariates.

```
#Computing descriptive statistics and pairwise effects for 100 cases per group
descs_gifted_post <- MAGMA_desc(Data = MAGMA_sim_data_gifted,
                                   group = "gifted_support",
                                   covariates = covariates_gifted,
                                   step_num = 100,
                                   step_var = "step",
                                   filename = "stats_gifted_post.docx")

#Displaying the table with defined column names
descs_gifted_post %>%
  purrr::set_names(c("Overall N", "Overall Mean", "Overall SD",
                    "No Support N", "No Support Mean", "No Support SD",
                    "Support N", "Support Mean", "Support SD",
                    "d"))

#> Overall N Overall Mean Overall SD No Support N No Support Mean
#> gifted_support    200      0.50      0.50      100      0.00
#> GPA_school       200      3.38      0.76      100      3.41
#> IQ_score          200     102.82     12.50      100     104.05
#> Motivation        200      3.87      0.84      100      3.87
#> parents_academic 200      0.42      0.49      100      0.39
#> sex                200      0.47      0.50      100      0.49
#> No Support SD Support N Support Mean Support SD      d
#> gifted_support      0.00     100      1.00      0.00 -Inf
#> GPA_school         0.76     100      3.36      0.77  0.07
#> IQ_score            12.13    100     101.59     12.80  0.20
#> Motivation          0.84     100      3.88      0.84 -0.01
#> parents_academic   0.49     100      0.45      0.50 -0.12
#> sex                 0.50     100      0.44      0.50  0.10
```

Exact Matching

Unlike in standard matching, only cases sharing the same value on a defined variable can be matched in exact matching. In this example, the variable for exact matching is `enrichment`. This means that cases that participated in afternoon enrichment can only be matched with other cases that participated, and non-participants can only be matched with other non-participants. Reasons for choosing exact matching instead of standard matching may be, for example, a nested data structure. Except for this preselection of possible matches, the process of matching stays the same. The main difference is that we need to use `MAGMA_exact` instead of `MAGMA` to match the data. This function has an additional argument, where the name of the exact variable must be defined as a character ("enrichment"). Note that the initial unbalance, descriptive statistics, and density overlap are the same as for standard matching.

```
MAGMA_sim_data_gifted_exact <- MAGMA_exact(Data = MAGMA_sim_data,
                                              group = "gifted_support",
                                              dist = "ps_gifted",
                                              exact = "enrichment",
                                              cores = 2)

#> input correctly identified
#> matching complete!
str(MAGMA_sim_data_gifted_exact)
#> 'data.frame': 800 obs. of 17 variables:
#> $ ID              : int 1 2 3 4 5 6 7 8 9 10 ...
#> $ sex             : int 1 0 1 1 1 0 0 0 1 0 ...
#> $ gifted_support  : int 1 1 1 1 1 0 1 0 1 ...
#> $ teacher_ability_rating: int 3 2 2 1 3 3 2 2 1 1 ...
#> $ enrichment       : int 0 1 0 1 0 0 0 0 0 1 ...
#> $ parents_academic : int 1 1 1 0 1 0 0 1 0 0 ...
#> $ GPA_school       : num 4.73 4.41 3.23 2.88 2.98 ...
#> $ IQ_score          : num 111.6 97.8 120.1 97.9 74.4 ...
#> $ Motivation        : num 5.36 4.41 3.57 4.34 3.54 ...
#> $ college_GPA       : num 4.01 3.41 2.73 4.02 4.09 ...
#> $ support_enrichment: int 3 4 3 4 3 3 1 3 1 4 ...
#> $ ps_tar            : num 0.0657 0.1085 0.1899 0.4261 0.4782 ...
#> $ ps_2x2             : num 0.3 0.32 0.21 0.237 0.571 ...
#> $ ps_gifted          : num 0.431 0.364 0.626 0.353 0.187 ...
#> $ step               : num 156 136 125 167 32 230 265 303 NA 18 ...
#> $ weight              : num 1 1 1 1 1 1 1 NA 1 ...
#> $ distance            : num 3.92e-05 1.91e-05 1.10e-05 9.41e-05 3.27e-08 ...
```

Balance Estimation and Visualization

The steps of balance estimation, visualization, and post matching statistics are the same as for standard matching. Therefore, we summarize these steps. The only change we need to consider is the different data frame that contains the result of the exact matching.

```

Balance_gifted_exact <- Balance_MAGMA(Data = MAGMA_sim_data_gifted_exact,
                                         group = "gifted_support",
                                         covariates = covariates_gifted,
                                         step = "step")
#> Start estimating Pillai's Trace.
#> Pillai's Trace finsihed. Starting to compute d-ratio.
#> d-ratio finsihed. Starting to compute mean-g.
#> [1] "mean g was computed using random effects meta-analysis with metafor."
#> mean-g finsihed. Starting to compute adjusted d-ratio.
#> finsihed balance estimation

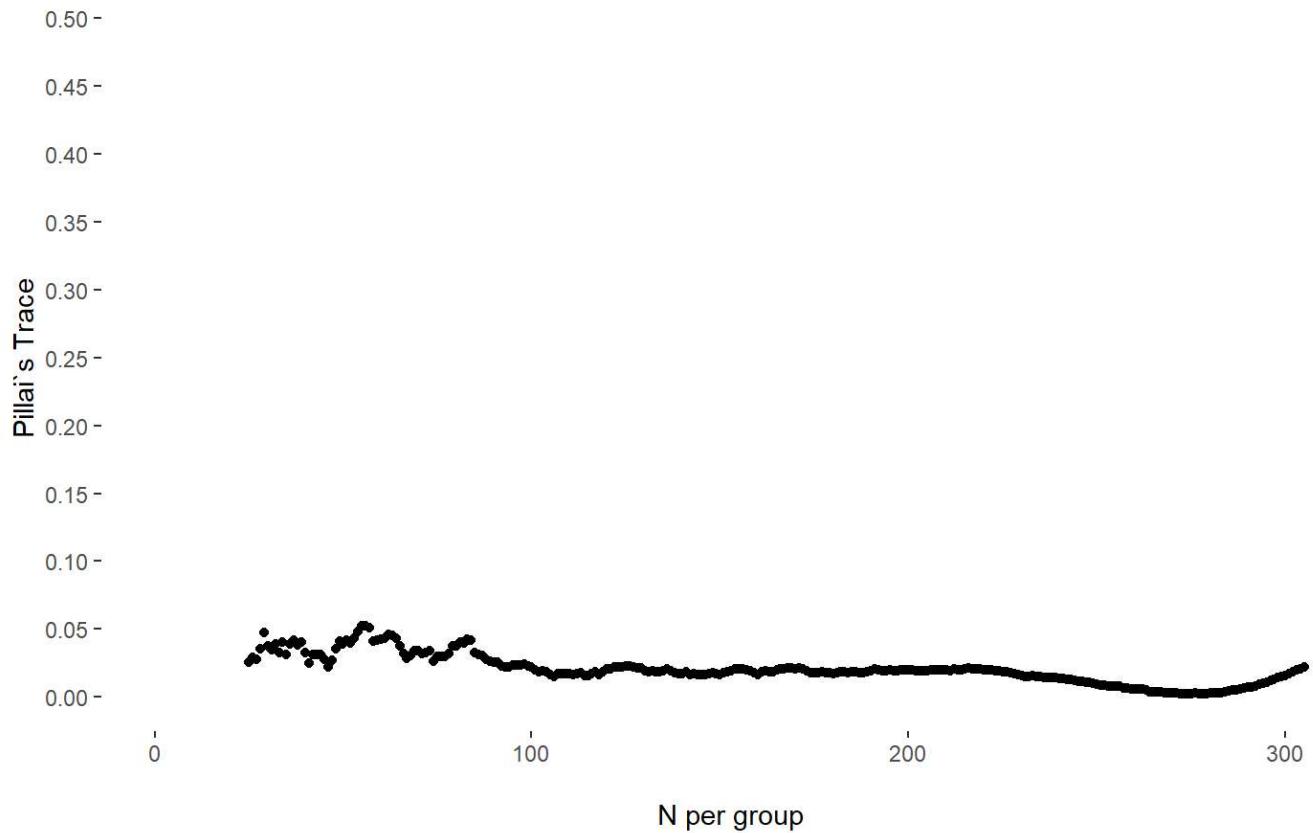
#Extracting balance criteria for 100 cases per group
Balance_100_criteria_exact <- Balance_extract(Balance = Balance_gifted_exact,
                                                samplesize = 100,
                                                effects = FALSE)
Balance_100_criteria_exact
#> Pillai's Trace      d-ratio      mean g    adj. d-ratio
#>          0.02        1.00        0.12        0.70

#Extracting pairwise effects for 100 cases per group
Balance_100_effects_exact <- Balance_extract(Balance = Balance_gifted_exact,
                                               samplesize = 100,
                                               effects = TRUE)
Balance_100_effects_exact
#>      GPA_school      IQ_score      Motivation_parents_academic
#>          0.16        0.17        0.05        0.16
#>      sex
#>          0.06

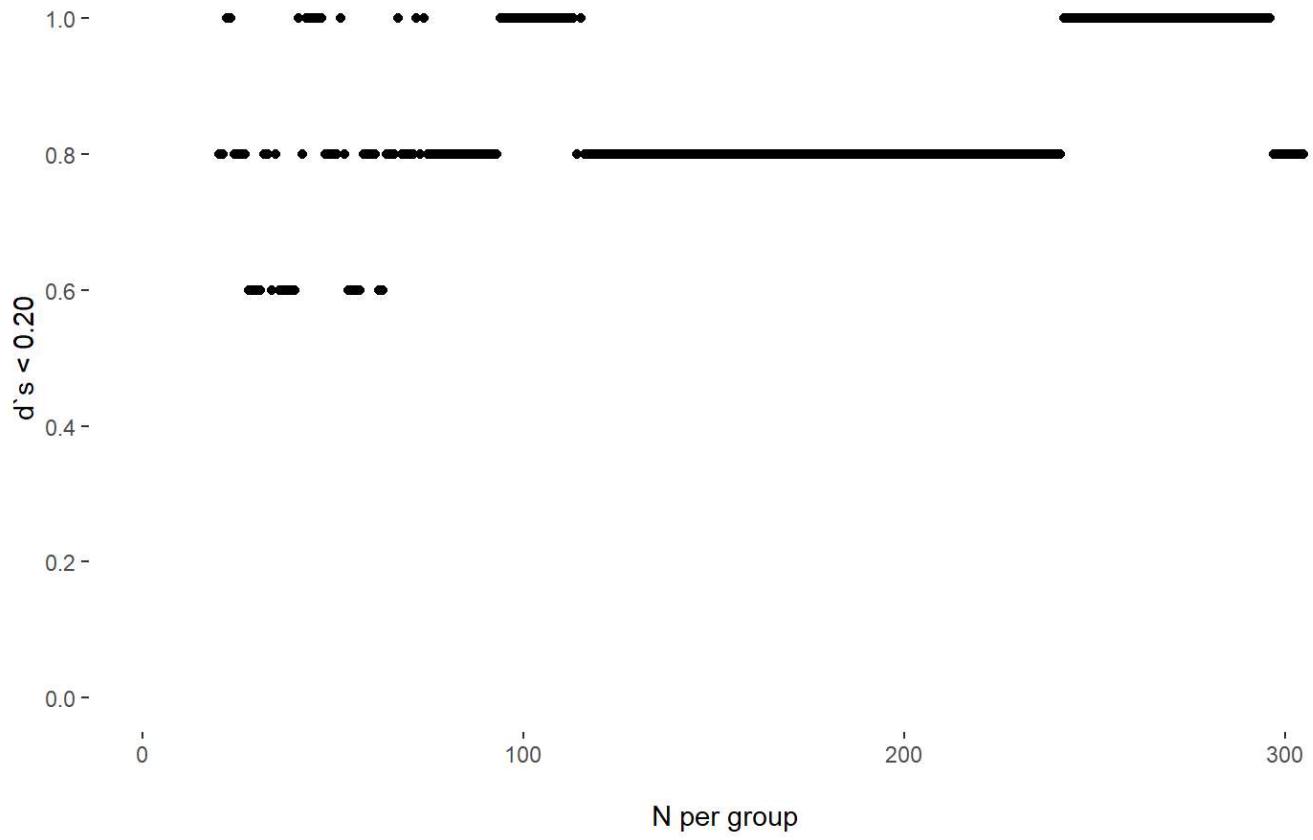
#Ploting trend over increasing sample size
Plot_MAGMA(Balance = Balance_gifted_exact,
            criterion = c("Pillai", "d_ratio", "mean_g", "Adj_d_ratio"))

```

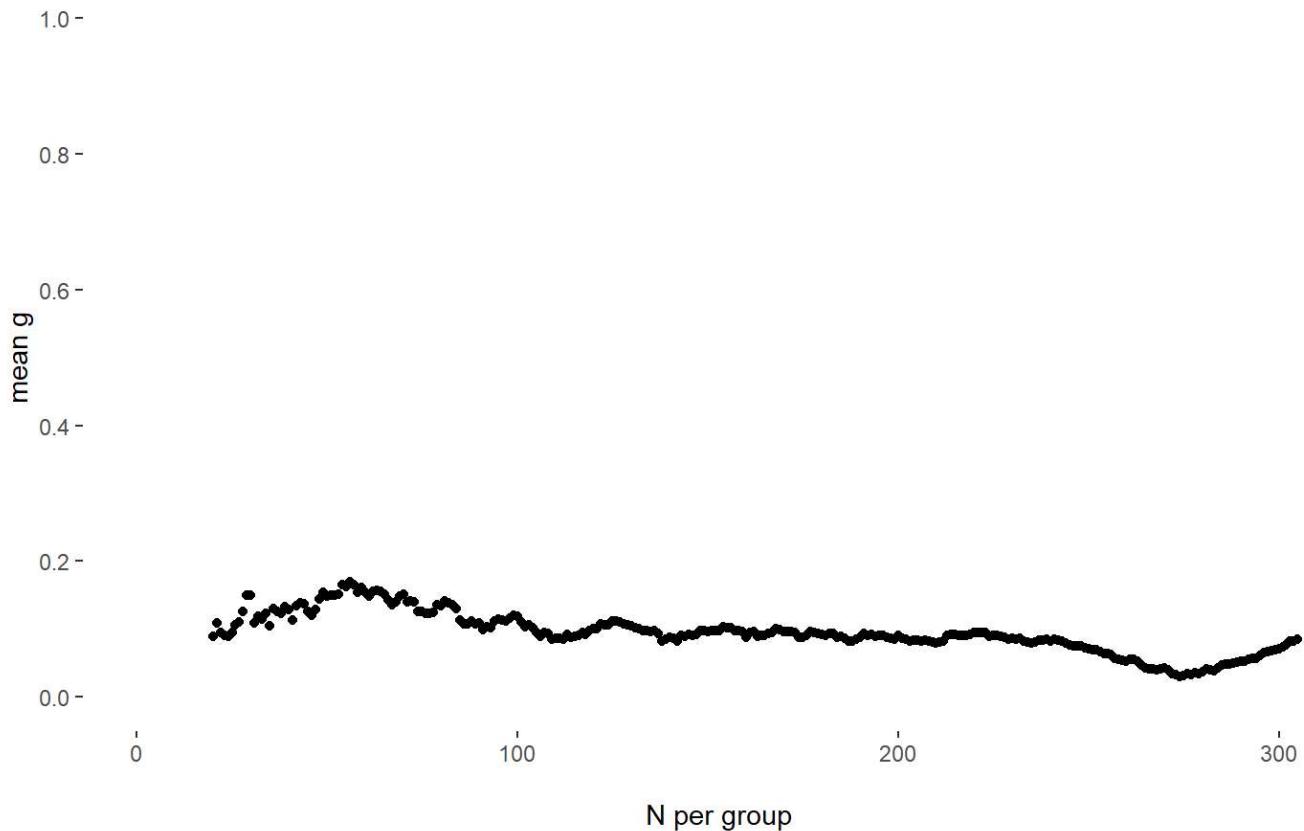
Pillai's Trace values for different sample sizes



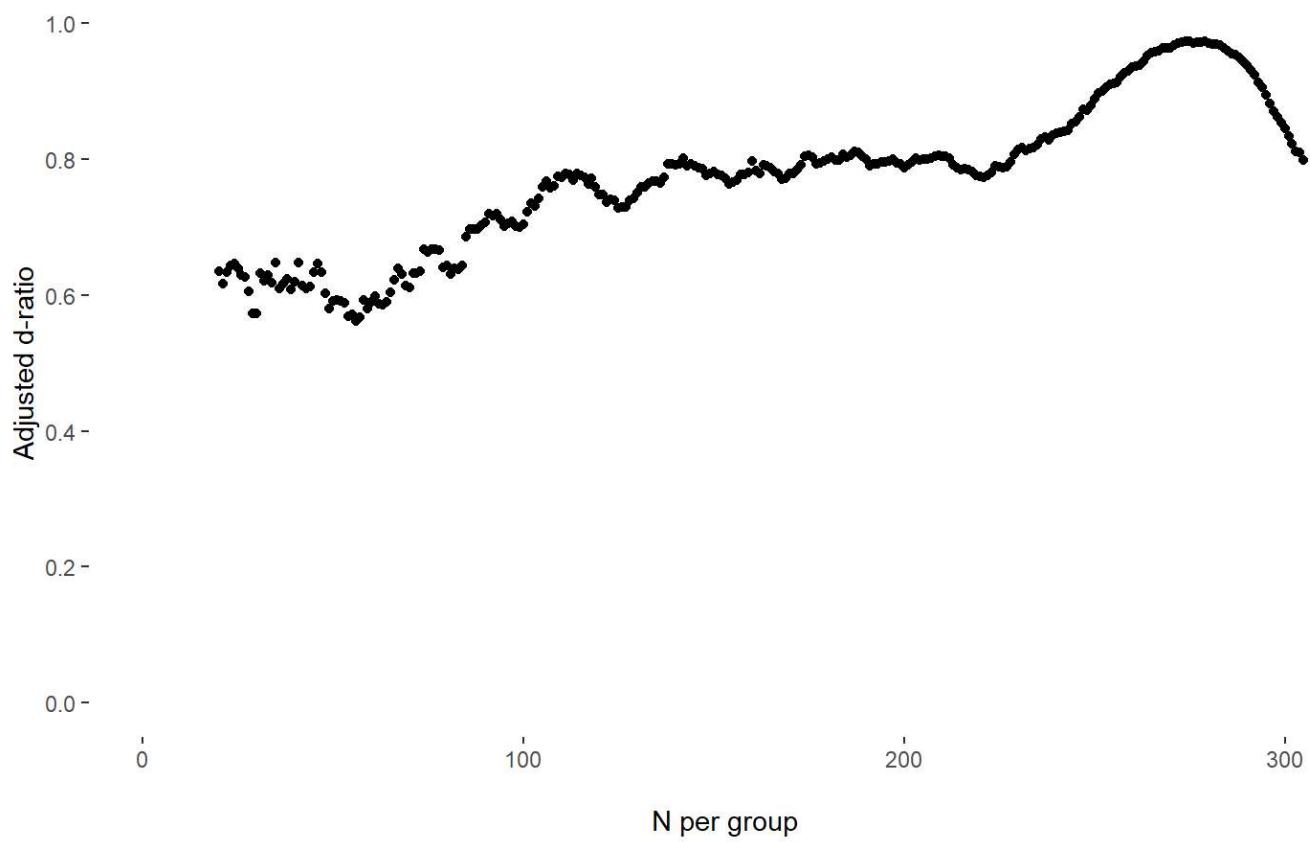
Cohen's d < .20` for different sample sizes



Mean effect for different sample sizes



Adjusted d-ratio for different sample sizes



```

#Creating table
Table_MAGMA(Balance = Balance_gifted_exact,
             filename = "Balance_gifted_exact.docx")
#> Balance Table successfully created!
#> # A tibble: 4 × 6
#>   Criterion_optimized Pillai_Trace d_ratio mean_g adjusted_d_ratio n_per_group
#>   <chr>                <dbl>    <dbl>    <dbl>    <dbl>      <int>
#> 1 Best mean g          0        1    0.03     0.97      274
#> 2 Best Pillai           0        1    0.03     0.97      279
#> 3 adjusted d-ratio     0        1    0.03     0.97      279
#> 4 Best d-ratio         0.01    1    0.07     0.88      296

#Computing descriptive statistics and pairwise effects for 100 cases per group
descs_gifted_post_exact <- MAGMA_desc(Data = MAGMA_sim_data_gifted_exact,
                                         group = "gifted_support",
                                         covariates = covariates_gifted,
                                         step_num = 100,
                                         step_var = "step",
                                         filename = "stats_gifted_post_exact.docx")

#Displaying the table with defined column names
descs_gifted_post_exact %>%
  purrr::set_names(c("Overall N", "Overall Mean", "Overall SD",
                    "No Support N", "No Support Mean", "No Support SD",
                    "Support N", "Support Mean", "Support SD",
                    "d"))
#> Overall N Overall Mean Overall SD No Support N No Support Mean
#> gifted_support    200      0.50      0.50      100      0.00
#> GPA_school        200      3.39      0.79      100      3.45
#> IQ_score          200     101.70     13.25      100     102.79
#> Motivation         200      3.88      0.88      100      3.90
#> parents_academic  200      0.38      0.49      100      0.34
#> sex                 200      0.47      0.50      100      0.48
#> No Support SD Support N Support Mean Support SD      d
#> gifted_support     0.00     100      1.00      0.00 -Inf
#> GPA_school          0.79     100      3.32      0.80  0.16
#> IQ_score            13.24     100     100.61     13.23  0.17
#> Motivation          0.92     100      3.86      0.85  0.05
#> parents_academic   0.48     100      0.42      0.50 -0.16
#> sex                  0.50     100      0.45      0.50  0.06

```

Three-Group Example

This is a fictitious three-group example. The independent variable (i.e., the treatment) is `teacher_ability_rating`. We are interested in how teacher-rated student ability, which ranged from below average (BA) to average (A) and above average (AA), affects the college GPA. While the process of matching and balance evaluation is the same as in the two-group example, the grouping variable (`teacher_ability_rating`), and consequently the distance variable/propensity score changes (`ps_tar`). Moreover, we slightly change the covariates by substituting `sex` through `gifted_support`.

```
covariates_tar <- c("GPA_school",
                     "IQ_score",
                     "Motivation",
                     "parents_academic",
                     "gifted_support")
```

Due to these changes in grouping variable and covariates, we need to estimate the initial unbalance, descriptive statistics, and area of common support for teacher-rated ability. Note that due to having three groups, the estimation of mean g slightly changes. However, this only affects the backend computation, while the code and the display of results are not affected. For more details, see Feuchter et al. (2022).

```

#Computing descriptive statistics and all pairwise effects for three groups
descs_tar_pre <- MAGMA_desc(Data = MAGMA_sim_data,
                             group = "teacher_ability_rating",
                             covariates = covariates_tar,
                             filename = "stats_tar_pre.docx")

descs_tar_pre %>%
  purrr::set_names(c("Overall N", "Overall Mean", "Overall SD",
                    "BA N", "BA Support Mean", "BA Support SD",
                    "A N", "A Mean", "A SD",
                    "AA N", "AA Mean", "AA SD",
                    "d BA-A", "d BA-AA", "d A-AA"))

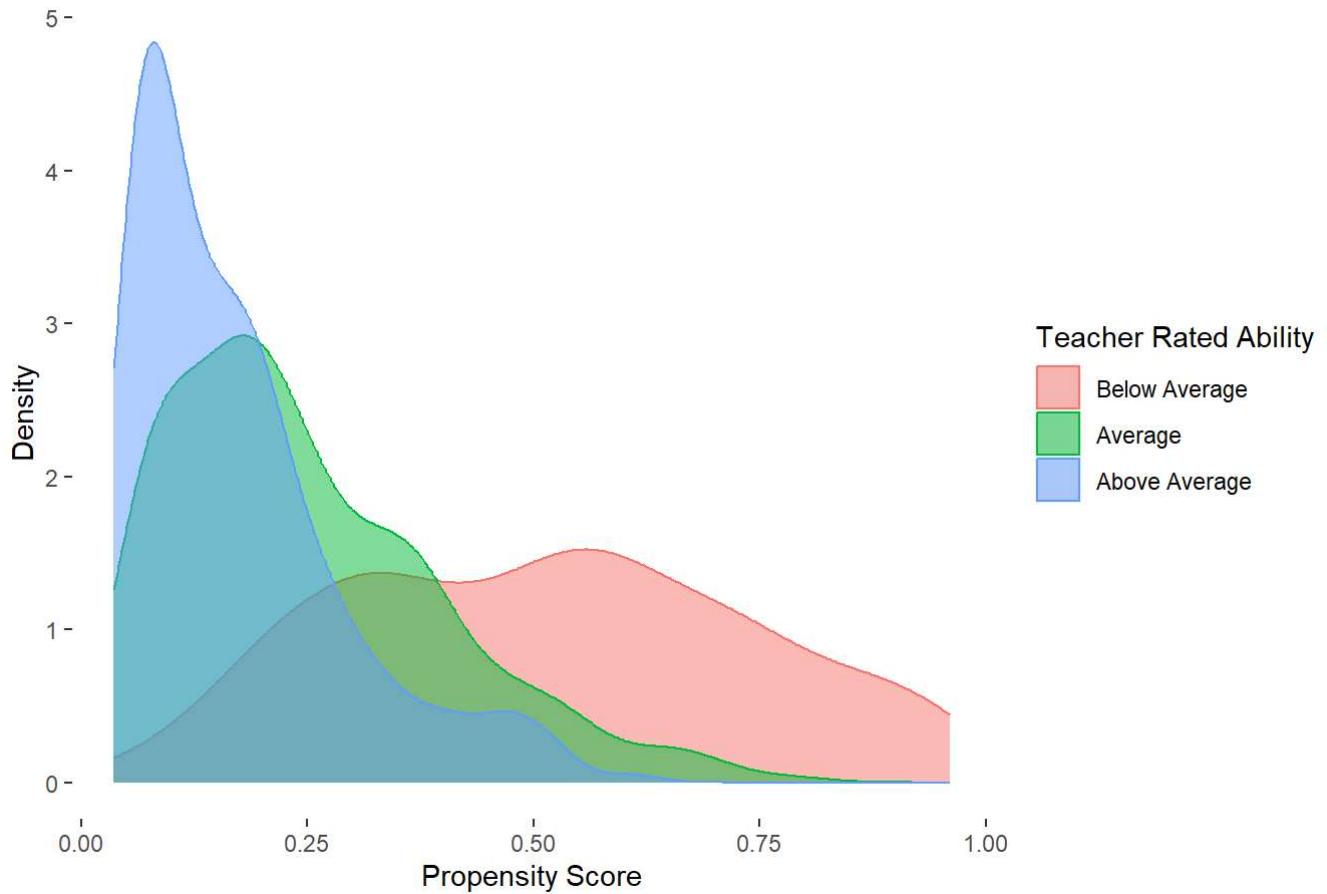
#> Overall N Overall Mean Overall SD BA N BA Support Mean
#> teacher_ability_rating     800      2.02      0.80  243      1.00
#> GPA_school                 800      3.29      0.93  243      2.76
#> IQ_score                   800    100.51     15.00  243    92.93
#> Motivation                  800      3.82      0.89  243      3.77
#> parents_academic            800      0.40      0.49  243      0.39
#> gifted_support                800      0.38      0.49  243      0.35
#> BA Support SD A N A Mean   A SD AA N AA Mean AA SD d BA-A
#> teacher_ability_rating      0.00 294    2.00  0.00  263    3.00  0.00 -Inf
#> GPA_school                  0.82 294    3.33  0.85  263    3.73  0.86 -0.68
#> IQ_score                     14.63 294  101.20 12.84  263  106.73 14.54 -0.61
#> Motivation                   0.88 294    3.72  0.88  263    3.98  0.90  0.06
#> parents_academic             0.49 294    0.38  0.49  263    0.42  0.49  0.02
#> gifted_support                 0.48 294    0.40  0.49  263    0.39  0.49 -0.10
#> d BA-AA d A-AA
#> teacher_ability_rating     -Inf  -Inf
#> GPA_school                  -1.16 -0.47
#> IQ_score                     -0.95 -0.41
#> Motivation                   -0.24 -0.29
#> parents_academic              -0.06 -0.08
#> gifted_support                 -0.08  0.02

#Estimating and printing initial imbalance for teacher rated ability
unbalance_tar <- initial_unbalance(Data = MAGMA_sim_data,
                                       group = "teacher_ability_rating",
                                       covariates = covariates_tar)
#> [1] "mean g was computed using robust variance meta-analysis with robumeta."
unbalance_tar
#> Pillai's Trace d-ratio Mean g adj. d-ratio
#> Unbalance        0.23    0.47    0.35        0.47

#Estimating and plotting density overlap in teacher rated ability propensity score
Density_overlap(Data = MAGMA_sim_data,
                 variable = "ps_tar",
                 group = "teacher_ability_rating",
                 variable_name = "Propensity Score",
                 group_labels = c("Below Average", "Average", "Above Average"),
                 group_name = "Teacher Rated Ability")

```

Density Plot for Propensity Score



```
#>                                     OV          OVPairs.Below Average-Average  
#>                                     0.7387758           0.5164831  
#> OVPairs.Below Average-Above Average          OVPairs.Average-Above Average  
#>                                     0.3483822           0.7658717
```

Standard Matching

After this check of initial unbalance, descriptive statistics, and common support, we can conduct the standard matching. As mentioned above, we need to adapt the grouping variable and the distance variable/propensity score for this example.

```

MAGMA_sim_data_tar <- MAGMA(Data = MAGMA_sim_data,
                                group = "teacher_ability_rating",
                                dist = "ps_tar",
                                cores = 2)
#> input correctly identified
#> Distance computation finished. Starting matching
#> matching complete!
str(MAGMA_sim_data_tar)
#> 'data.frame': 800 obs. of 17 variables:
#> $ ID : int 1 2 3 4 5 6 7 8 9 10 ...
#> $ sex : int 1 0 1 1 1 0 0 0 1 0 ...
#> $ gifted_support : int 1 1 1 1 1 0 1 0 1 ...
#> $ teacher_ability_rating: int 3 2 2 1 3 3 2 2 1 1 ...
#> $ enrichment : int 0 1 0 1 0 0 0 0 0 1 ...
#> $ parents_academic : int 1 1 1 0 1 0 0 1 0 0 ...
#> $ GPA_school : num 4.73 4.41 3.23 2.88 2.98 ...
#> $ IQ_score : num 111.6 97.8 120.1 97.9 74.4 ...
#> $ Motivation : num 5.36 4.41 3.57 4.34 3.54 ...
#> $ college_GPA : num 4.01 3.41 2.73 4.02 4.09 ...
#> $ support_enrichment : int 3 4 3 4 3 3 1 3 1 4 ...
#> $ ps_tar : num 0.0657 0.1085 0.1899 0.4261 0.4782 ...
#> $ ps_2x2 : num 0.3 0.32 0.21 0.237 0.571 ...
#> $ ps_gifted : num 0.431 0.364 0.626 0.353 0.187 ...
#> $ step : num 222 237 28 106 66 62 182 NA 236 38 ...
#> $ weight : num 1 1 1 1 1 1 NA 1 1 ...
#> $ distance : num 2.53 3.35 2.96e-06 1.88e-01 5.22e-05 ...

```

Balance Estimation and Visualization

We only need to change the specific object names, covariates, grouping variable, and the filename for the table. Otherwise, the process of balance estimation and visualization remains the same.

```

Balance_tar <- Balance_MAGMA(Data = MAGMA_sim_data_tar,
                               group = "teacher_ability_rating",
                               covariates = covariates_tar,
                               step = "step")
#> Start estimating Pillai's Trace.
#> Pillai's Trace finsihed. Starting to compute d-ratio.
#> d-ratio finsihed. Starting to compute mean-g.
#> [1] "mean g was computed using robust variance meta-analysis with robumeta."
#> mean-g finsihed. Starting to compute adjusted d-ratio.
#> finsihed balance estimation

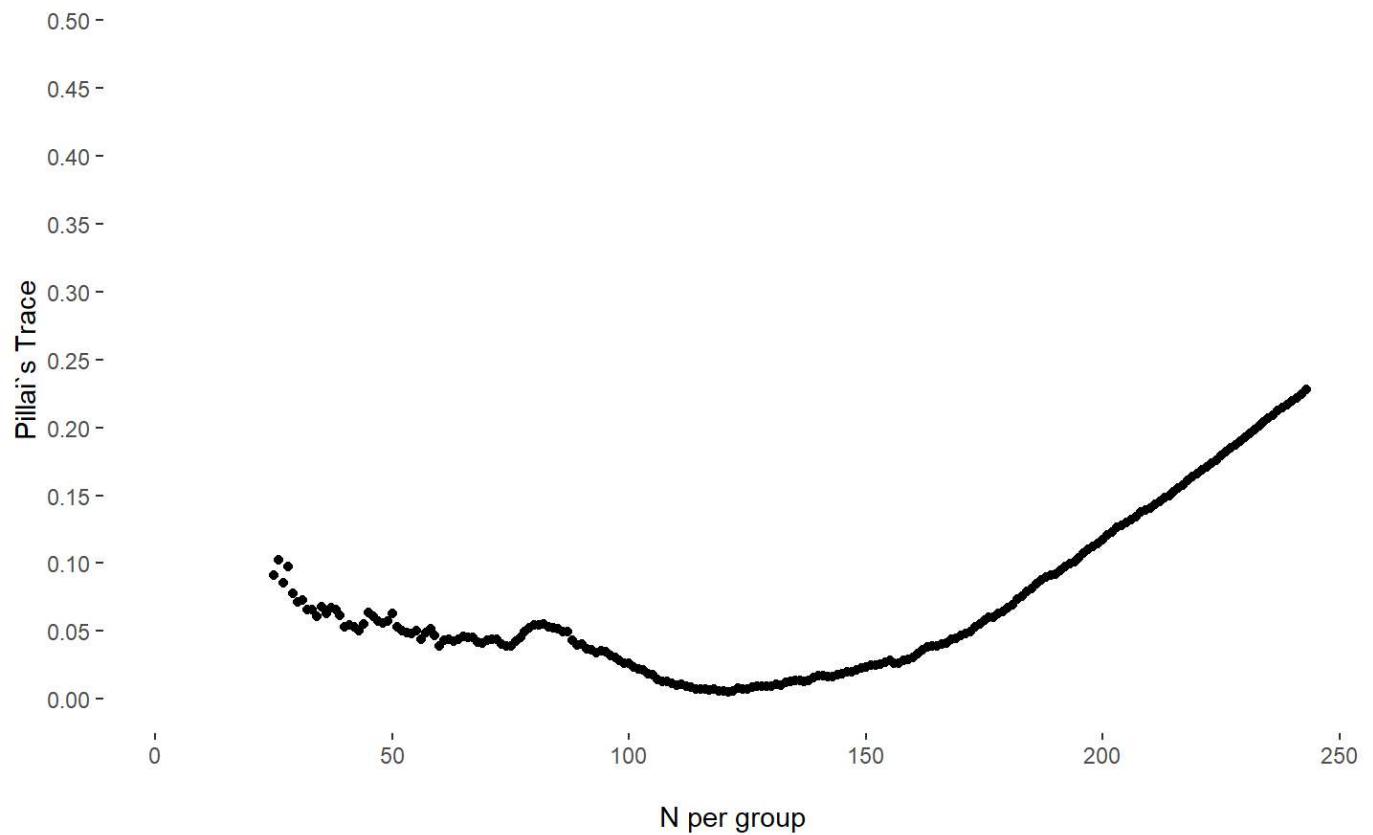
#Balance criteria for a 100 cases per group
Balance_100_tar_criteria <- Balance_extract(Balance = Balance_tar,
                                              samplesize = 100,
                                              effects = FALSE)
Balance_100_tar_criteria
#> Pillai's Trace      d-ratio      mean g    adj. d-ratio
#>          0.03        0.67        0.14        0.64

#Extracting pairwise effects for 100 cases per group
Balance_100_tar_effects <- Balance_extract(Balance = Balance_tar,
                                              samplesize = 100,
                                              effects = TRUE)
Balance_100_tar_effects
#>      GPA_school_1_2      IQ_score_1_2      Motivation_1_2
#>          0.32            0.22            0.20
#> parents_academic_1_2  gifted_support_1_2  GPA_school_1_3
#>          0.04            0.00            0.37
#>      IQ_score_1_3      Motivation_1_3  parents_academic_1_3
#>          0.23            0.07            0.14
#>  gifted_support_1_3  GPA_school_2_3  IQ_score_2_3
#>          0.04            0.03            0.05
#> Motivation_2_3  parents_academic_2_3  gifted_support_2_3
#>          0.26            0.10            0.04

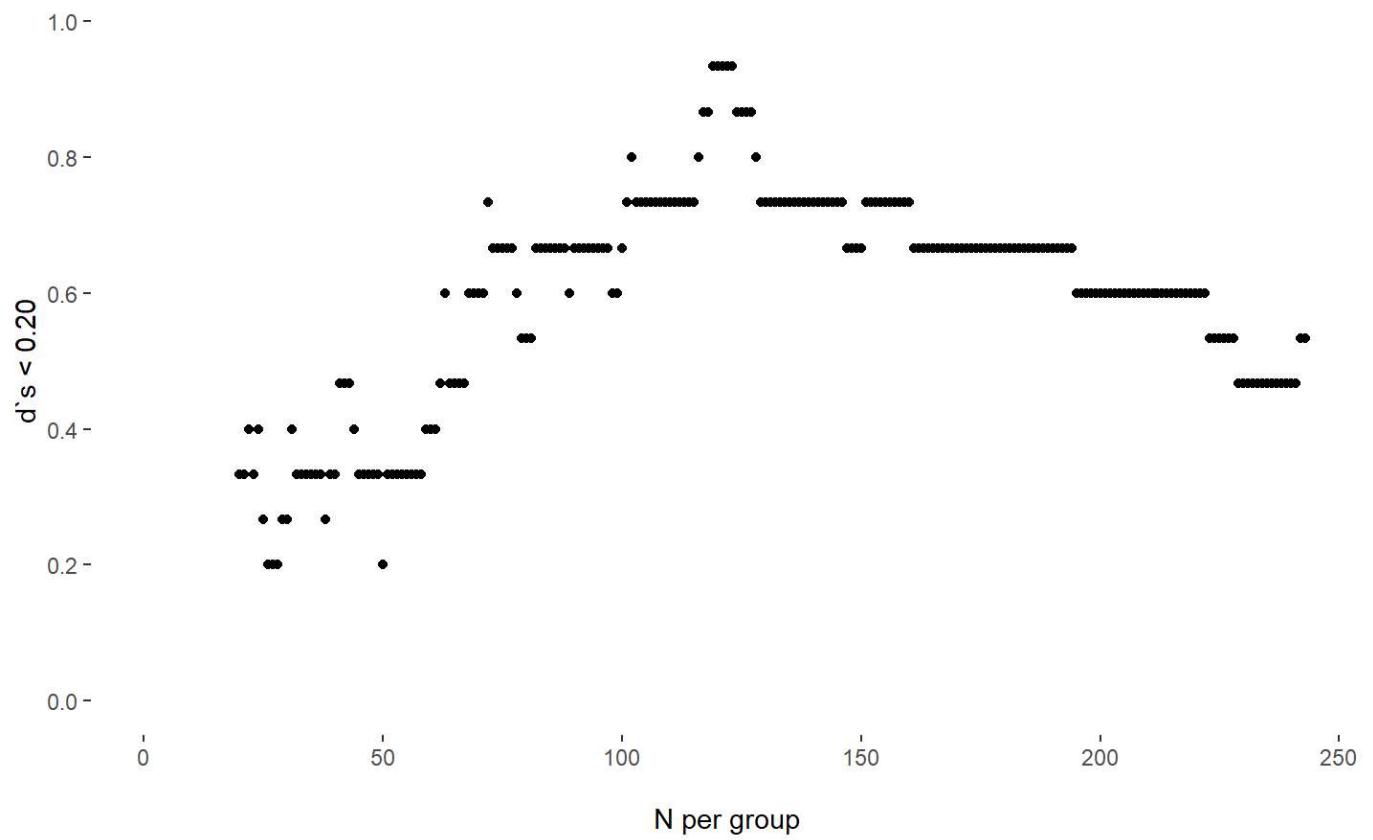
#Ploting trend over increasing samle size
Plot_MAGMA(Balance = Balance_tar,
            criterion = c("Pillai", "d_ratio", "mean_g", "Adj_d_ratio"))

```

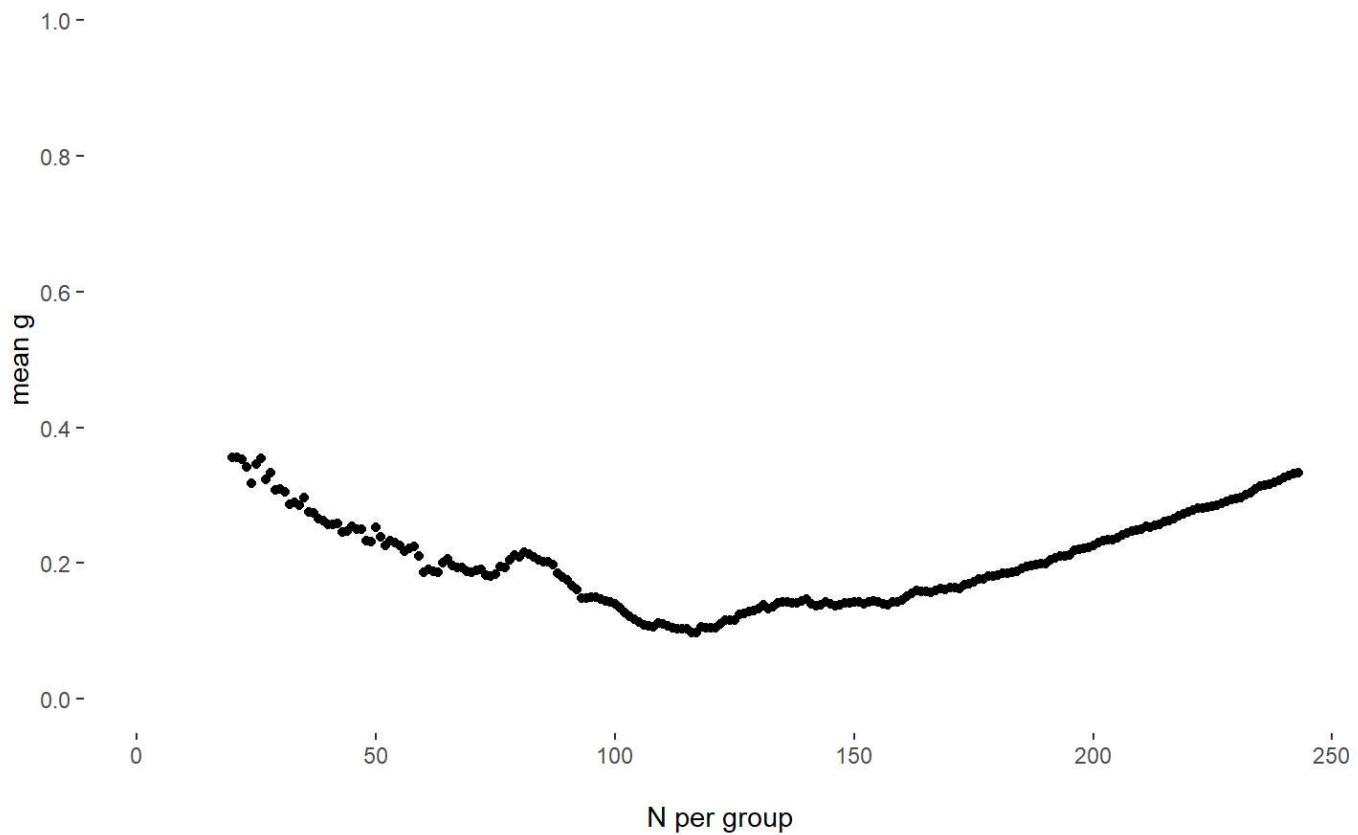
Pillai's Trace values for different sample sizes



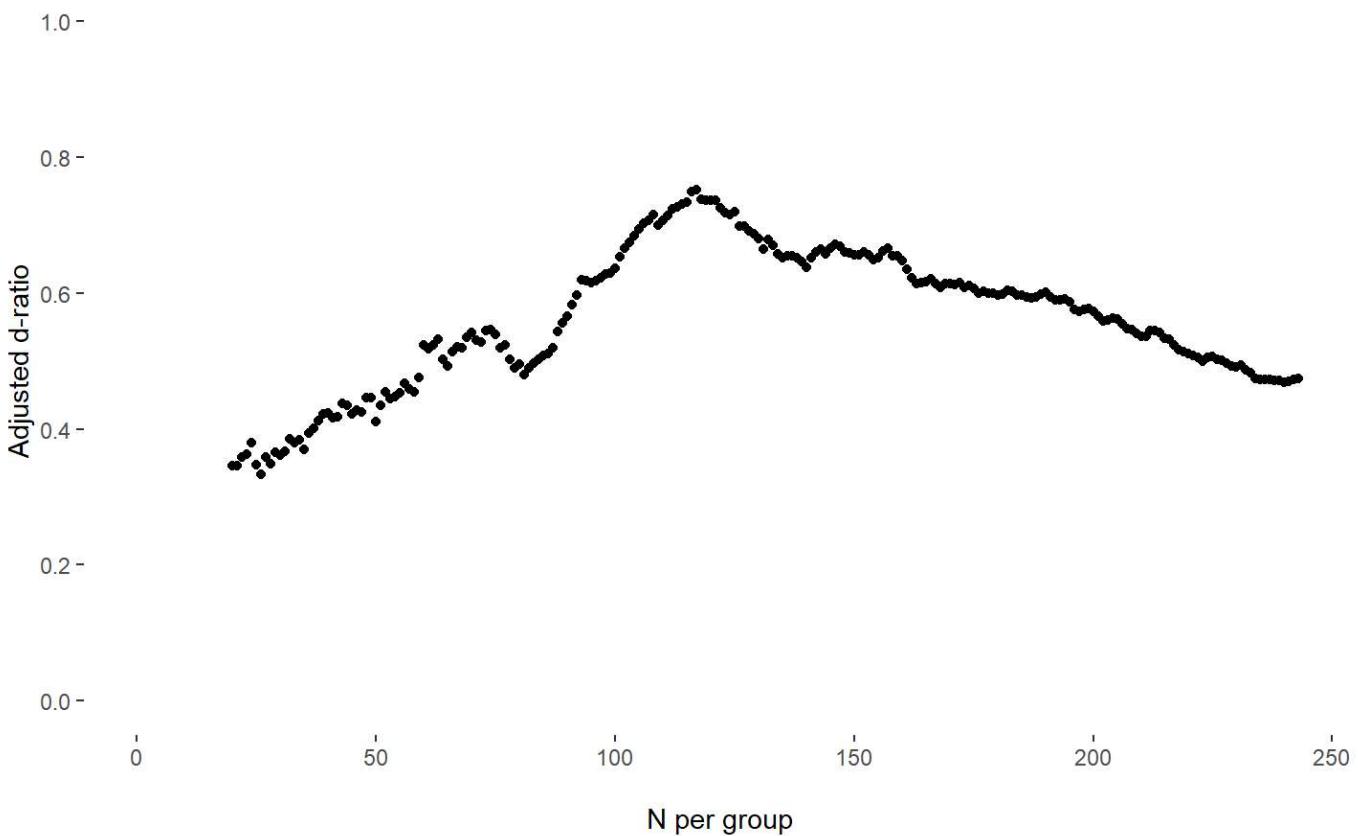
Cohen's d < .20` for different sample sizes



Mean effect for different sample sizes



Adjusted d-ratio for different sample sizes



```

#Creating table
Table_MAGMA(Balance = Balance_tar,
             filename = "Balance_tar.docx")
#> Balance Table successfully created!
#> # A tibble: 4 × 6
#>   Criterion_optimized Pillai_Trace d_ratio mean_g adjusted_d_ratio n_per_group
#>   <chr>                  <dbl>    <dbl>    <dbl>            <dbl>      <int>
#> 1 Best mean g           0.01     0.8      0.1            0.75       116
#> 2 adjusted d-ratio      0.01     0.87     0.1            0.75       117
#> 3 Best Pillai           0.01     0.93     0.11           0.74       121
#> 4 Best d-ratio          0.01     0.93     0.12           0.72       123

```

Post-matching statistics

Again, after selecting a final model we can report the post-matching descriptive statistics. Therefore, we need to adapt our arguments to this three group example.

```

#Computing descriptive statistics and pairwise effects for 100 cases per group
descs_tar_post <- MAGMA_desc(Data = MAGMA_sim_data_tar,
                                group = "teacher_ability_rating",
                                covariates = covariates_tar,
                                step_num = 100,
                                step_var = "step",
                                filename = "stats_tar_post.docx")

#Displaying the table with defined column names
descs_tar_post %>%
  purrr::set_names(c("Overall N", "Overall Mean", "Overall SD",
                     "BA N", "BA Support Mean", "BA Support SD",
                     "A N", "A Mean", "A SD",
                     "AA N", "AA Mean", "AA SD",
                     "d BA-A", "d BA-AA", "d A-AA"))

#>          Overall N Overall Mean Overall SD BA N BA Support Mean
#> teacher_ability_rating      300       2.00       0.82  100       1.00
#> GPA_school                  300       3.21       0.66  100       3.36
#> IQ_score                     300      99.29      12.99  100     101.19
#> Motivation                   300       3.86       0.87  100       3.89
#> parents_academic            300       0.39       0.49  100       0.42
#> gifted_support                300       0.41       0.49  100       0.42
#>          BA Support SD A N A Mean   A SD AA N AA Mean AA SD d BA-A
#> teacher_ability_rating      0.00 100    2.00  0.00  100    3.00  0.00 -Inf
#> GPA_school                  0.68 100    3.14  0.70  100    3.12  0.59  0.32
#> IQ_score                     11.71 100   98.67 11.74  100   98.02 15.13  0.22
#> Motivation                   0.86 100    3.73  0.82  100    3.96  0.93  0.19
#> parents_academic            0.50 100    0.40  0.49  100    0.35  0.48  0.04
#> gifted_support                0.50 100    0.42  0.50  100    0.40  0.49  0.00
#>          d BA-AA d A-AA
#> teacher_ability_rating     -Inf   -Inf
#> GPA_school                  0.38   0.03
#> IQ_score                     0.24   0.05
#> Motivation                  -0.08  -0.26
#> parents_academic            0.14   0.10
#> gifted_support                 0.04   0.04

```

Exact Matching

We use the same covariates as we used for the standard three-group matching. The only change is the use of `sex` as exact variable. Thus, girls will only be matched to girls, and boys will only be matched to other boys across the three groups. Initial unbalance, descriptive statistics, and common support are the same as for standard matching. Attached you find the whole matching, balance estimation, and visualization process.


```

Balance_100_tar_criteria_exact
#> Pillai's Trace      d-ratio      mean g   adj. d-ratio
#>      0.03           0.67        0.16       0.61

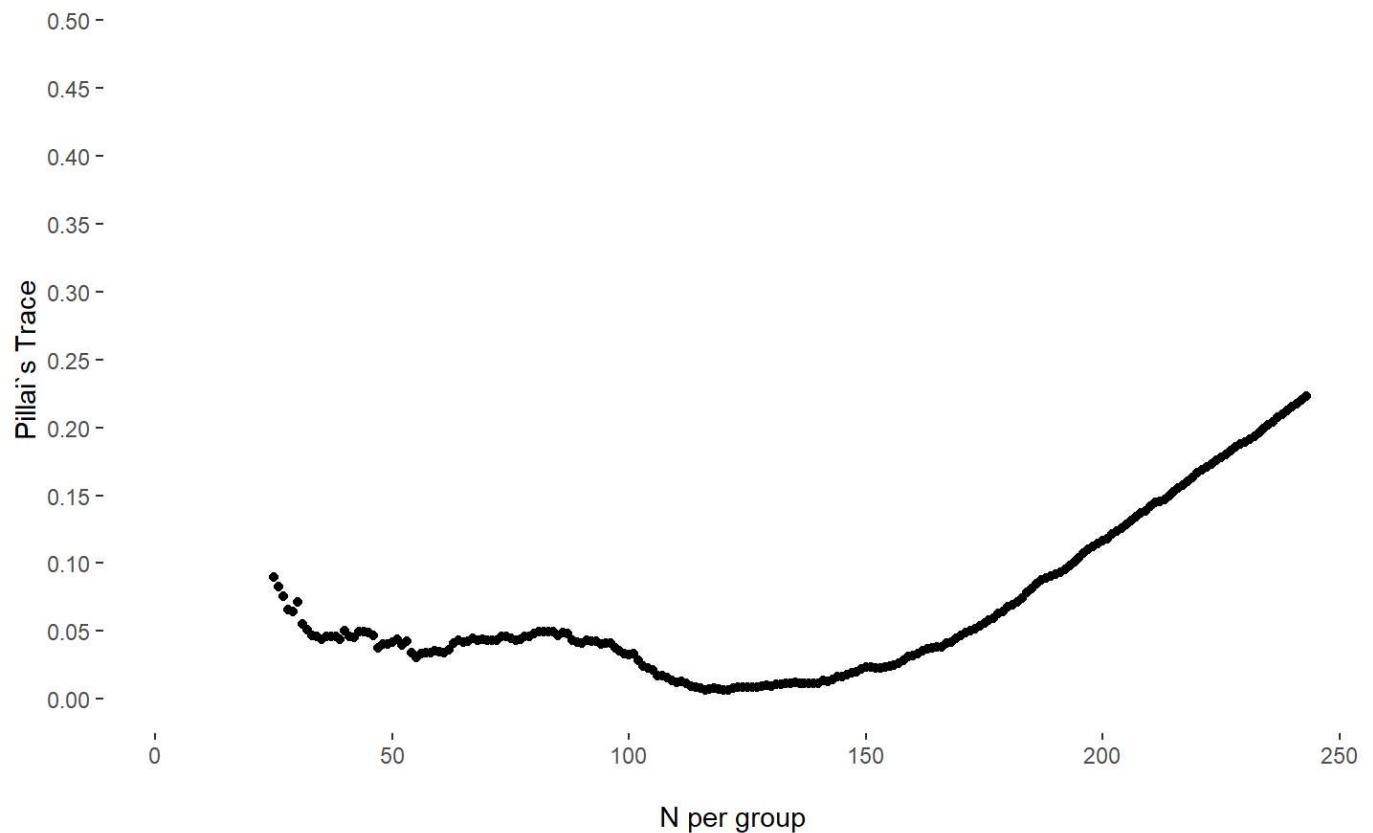
#Extracting pairwise effects for 100 cases per group
Balance_100_tar_effects_exact <- Balance_extract(Balance = Balance_tar_exact,
                                                    samplesize = 100,
                                                    effects = TRUE)

Balance_100_tar_effects_exact
#>      GPA_school_1_2      IQ_score_1_2      Motivation_1_2
#>      0.36                0.34                0.10
#> parents_academic_1_2   gifted_support_1_2   GPA_school_1_3
#>      0.12                0.12                0.37
#>      IQ_score_1_3        Motivation_1_3   parents_academic_1_3
#>      0.24                0.14                0.12
#>      gifted_support_1_3  GPA_school_2_3   IQ_score_2_3
#>      0.04                0.00                0.05
#>      Motivation_2_3     parents_academic_2_3  gifted_support_2_3
#>      0.24                0.00                0.08

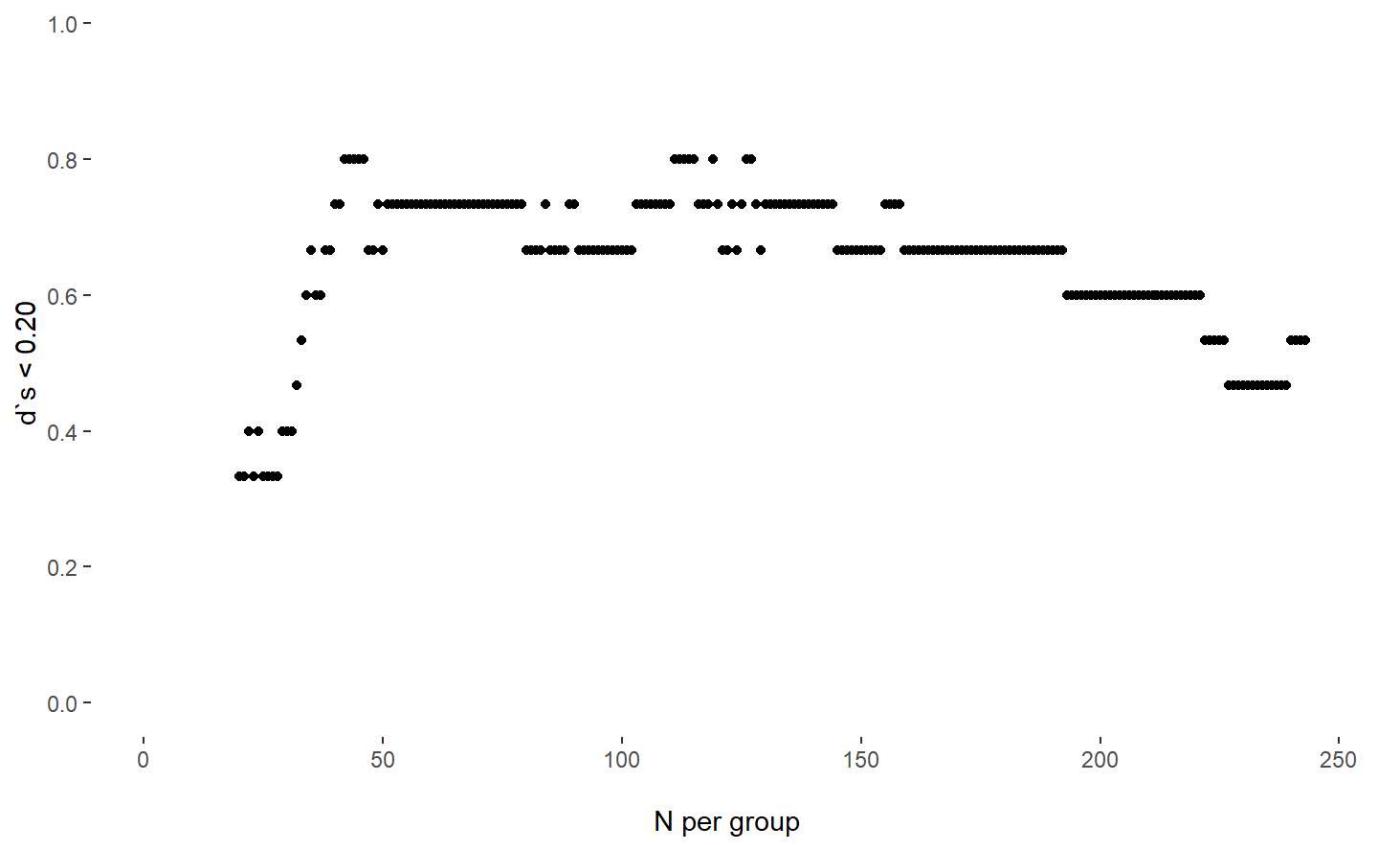
#Ploting trend over increasing samle size
Plot_MAGMA(Balance = Balance_tar_exact,
            criterion = c("Pillai", "d_ratio", "mean_g", "Adj_d_ratio")) #Could be omitted

```

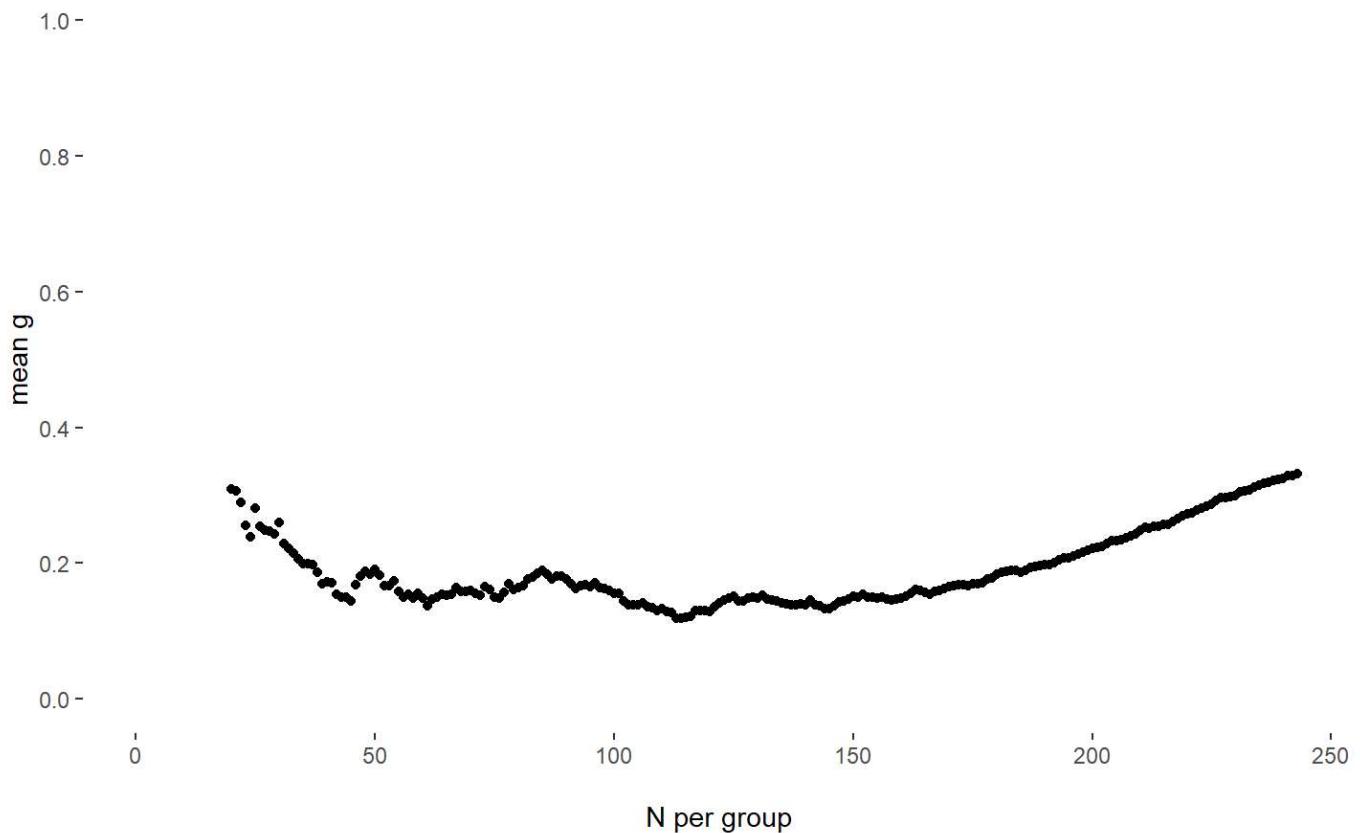
Pillai's Trace values for different sample sizes



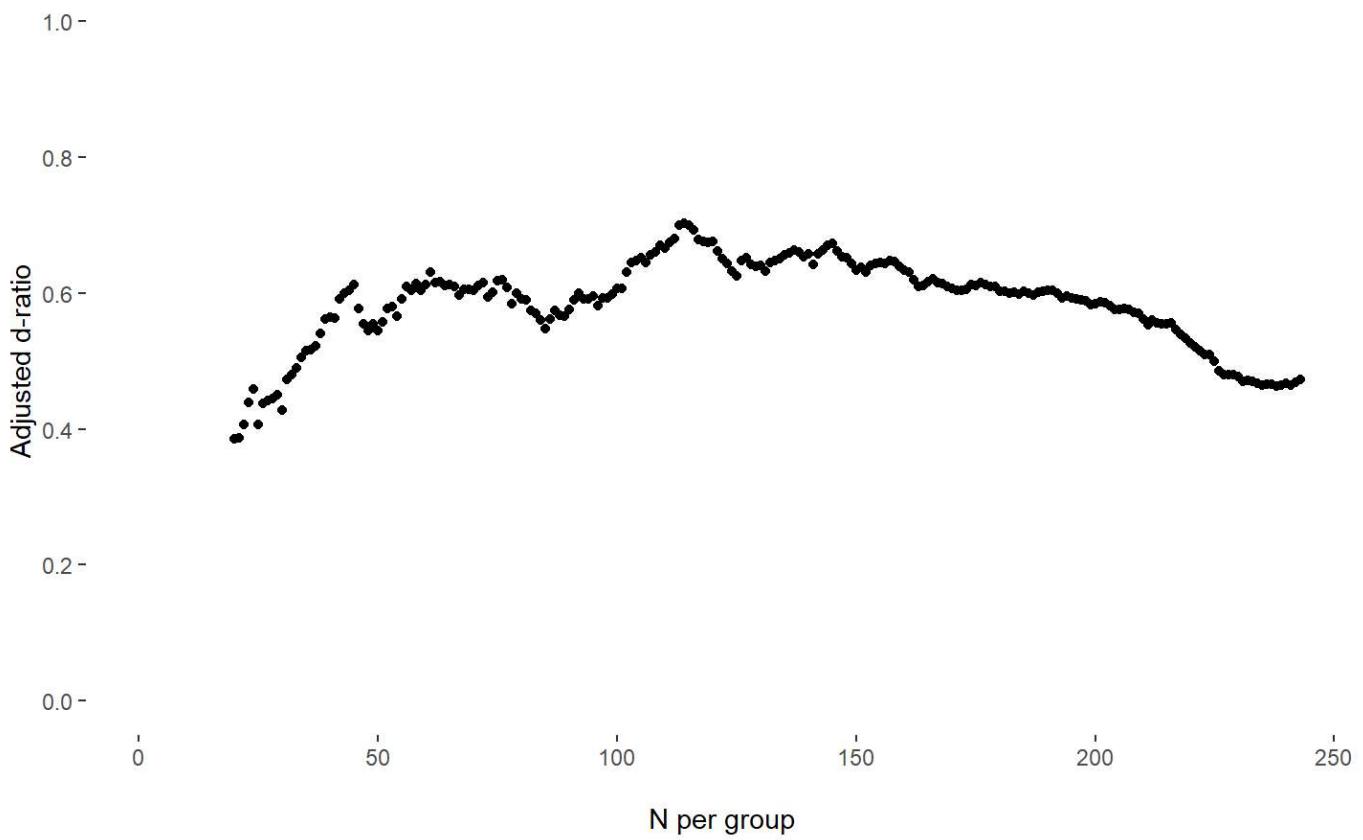
Cohen's d < .20` for different sample sizes



Mean effect for different sample sizes



Adjusted d-ratio for different sample sizes



```

#Creating table
Table_MAGMA(Balance = Balance_tar_exact,
             filename = "Balance_tar_exact.docx")
#> Balance Table successfully created!
#> # A tibble: 4 × 6
#>   Criterion_optimized Pillai Trace d_ratio mean_g adjusted_d_ratio n_per_group
#>   <chr>                <dbl>    <dbl>    <dbl>      <dbl>      <int>
#> 1 Best mean g          0.01     0.8     0.12       0.7       113
#> 2 adjusted d-ratio     0.01     0.8     0.12       0.7       114
#> 3 Best Pillai          0.01     0.73    0.12       0.69      116
#> 4 Best d-ratio         0.01     0.8     0.14       0.65      127

#Computing descriptive statistics and pairwise effects for 100 cases per group
descs_tar_post_exact <- MAGMA_desc(Data = MAGMA_sim_data_tar_exact,
                                     group = "teacher_ability_rating",
                                     covariates = covariates_tar,
                                     step_num = 100,
                                     step_var = "step",
                                     filename = "stats_tar_post_exact.docx")

#Displaying the table with defined column names
descs_tar_post_exact %>%
  purrr::set_names(c("Overall N", "Overall Mean", "Overall SD",
                    "BA N", "BA Support Mean", "BA Support SD",
                    "A N", "A Mean", "A SD",
                    "AA N", "AA Mean", "AA SD",
                    "d BA-A", "d BA-AA", "d A-AA"))
#> 
#> Overall N Overall Mean Overall SD BA N BA Support Mean
#> teacher_ability_rating 300      2.00      0.82 100      1.00
#> GPA_school            300      3.21      0.65 100      3.37
#> IQ_score               300      98.95     13.11 100     101.40
#> Motivation             300      3.83      0.91 100      3.82
#> parents_academic      300      0.38      0.49 100      0.42
#> gifted_support         300      0.39      0.49 100      0.42
#> 
#> BA Support SD A N A Mean A SD AA N AA Mean AA SD d BA-A
#> teacher_ability_rating 0.00 100  2.00  0.00 100  3.00  0.00 -Inf
#> GPA_school             0.68 100  3.13  0.64 100  3.13  0.60  0.37
#> IQ_score                1.93 100  97.36 11.53 100  98.07 15.30  0.35
#> Motivation              0.91 100  3.73  0.91 100  3.95  0.90  0.10
#> parents_academic       0.50 100  0.36  0.48 100  0.36  0.48  0.12
#> gifted_support          0.50 100  0.36  0.48 100  0.40  0.49  0.12
#> 
#> d BA-AA d A-AA
#> teacher_ability_rating -Inf -Inf
#> GPA_school             0.38  0.00
#> IQ_score                0.24 -0.05
#> Motivation              -0.14 -0.24
#> parents_academic       0.12  0.00
#> gifted_support          0.04 -0.08

```

2x2/four-group Example

In this fictitious 2x2-design example the variables `gifted_support` and `enrichment` are the independent (i.e., the treatment) variables. We are interested in how receiving giftedness support and the participation in afternoon enrichment affect the college GPA. The process of matching and balance evaluation is the same as in the previous examples. The grouping variable consists of two variables, so we use a character vector as input for the grouping variable (`c("gifted_support", "enrichment")`). Consequently, we also adapt the distance variable/propensity score (`ps_2x2`). Moreover, we use the same covariates as in the two-group example. Note that this 2x2 matching is nearly equivalent to a four-group matching using a multinomial variable indicating the four cells of the 2x2 group design. The multinomial variable for this example is `support_enrichment`. Using this variable is – regarding the code – similar to the three group examples, wherefore we focus on the 2x2 design here.

Major distinctions between such a multinomial four-group matching and a 2x2 matching occur only in the balance estimation. Because of these changes in grouping variable and covariates, we need to estimate the initial unbalance for the two independent variables `gifted_support` and `enrichment`. The result of the `initial_unbalance` function changes slightly in this two-factorial design. Pillai's Trace is estimated separately for the two main effects and the interaction. As can be seen, *d*-ratio, mean *g*, and adjusted *d*-ratio are the same for a 2x2 or a four-group matching. Changes occur only for Pillai's Trace. In a first step, we define the covariates and estimate initial unbalance, descriptive statistics, and area of common support.

```

#Defining the covariates
covariates_2x2 <- c("GPA_school",
                     "IQ_score",
                     "Motivation",
                     "parents_academic",
                     "sex")

#Computing descriptive statistics and all pairwise effects
descs_2x2_pre <- MAGMA_desc(Data = MAGMA_sim_data,
                               group = c("gifted_support", "enrichment"),
                               covariates = covariates_2x2,
                               filename = "stats_2x2_pre.docx")
#> 2x2 groups are represented as 4 groups.

descs_2x2_pre %>%
  purrr::set_names(c("Overall N", "Overall Mean", "Overall SD",
                    "Sup & No En N", "Sup & No En Mean",
                    "Sup & No En SD",
                    "Sup & En N", "Sup & En Mean", "Sup & En SD",
                    "No Sup & No En N", "No Sup & No En Mean", "No Sup & No En SD",
                    "No Sup & En N", "No Sup & En Mean", "No Sup & En SD",
                    "d YesNo-YesYes", "d YesNo-NoNo", "d YesNo-NoYes",
                    "d YesYes-NoNo", "d YesYes-YNoYes",
                    "d NoNo-NoYes"))

#> Overall N Overall Mean Overall SD Sup & No En N
#> group_Long     800      2.63      1.07      175
#> GPA_school     800      3.29      0.93      175
#> IQ_score       800    100.51     15.00      175
#> Motivation     800      3.82      0.89      175
#> parents_academic 800      0.40      0.49      175
#> sex             800      0.50      0.50      175
#>                 Sup & No En Mean Sup & No En SD Sup & En N Sup & En Mean
#> group_Long        1.00      0.00      130      2.00
#> GPA_school        3.44      0.88      130      3.71
#> IQ_score        105.09     14.09      130    110.26
#> Motivation        3.79      0.90      130      3.89
#> parents_academic 0.43      0.50      130      0.53
#> sex              0.49      0.50      130      0.48
#>                 Sup & En SD No Sup & No En N No Sup & No En Mean
#> group_Long        0.00      309      3.00
#> GPA_school        0.88      309      3.01
#> IQ_score        12.74      309    94.27
#> Motivation        0.86      309      3.75
#> parents_academic 0.50      309      0.37
#> sex              0.50      309      0.50
#>                 No Sup & No En SD No Sup & En N No Sup & En Mean
#> group_Long        0.00      186      4.00
#> GPA_school        0.93      186      3.31
#> IQ_score        13.30      186    99.74
#> Motivation        0.89      186      3.92
#> parents_academic 0.48      186      0.32
#> sex              0.50      186      0.51
#>                 No Sup & En SD d YesNo-YesYes d YesNo-NoNo d YesNo-NoYes
#> group_Long        0.00      -Inf      -Inf      -Inf
#> GPA_school        0.86     -0.31      0.47      0.15

```

```

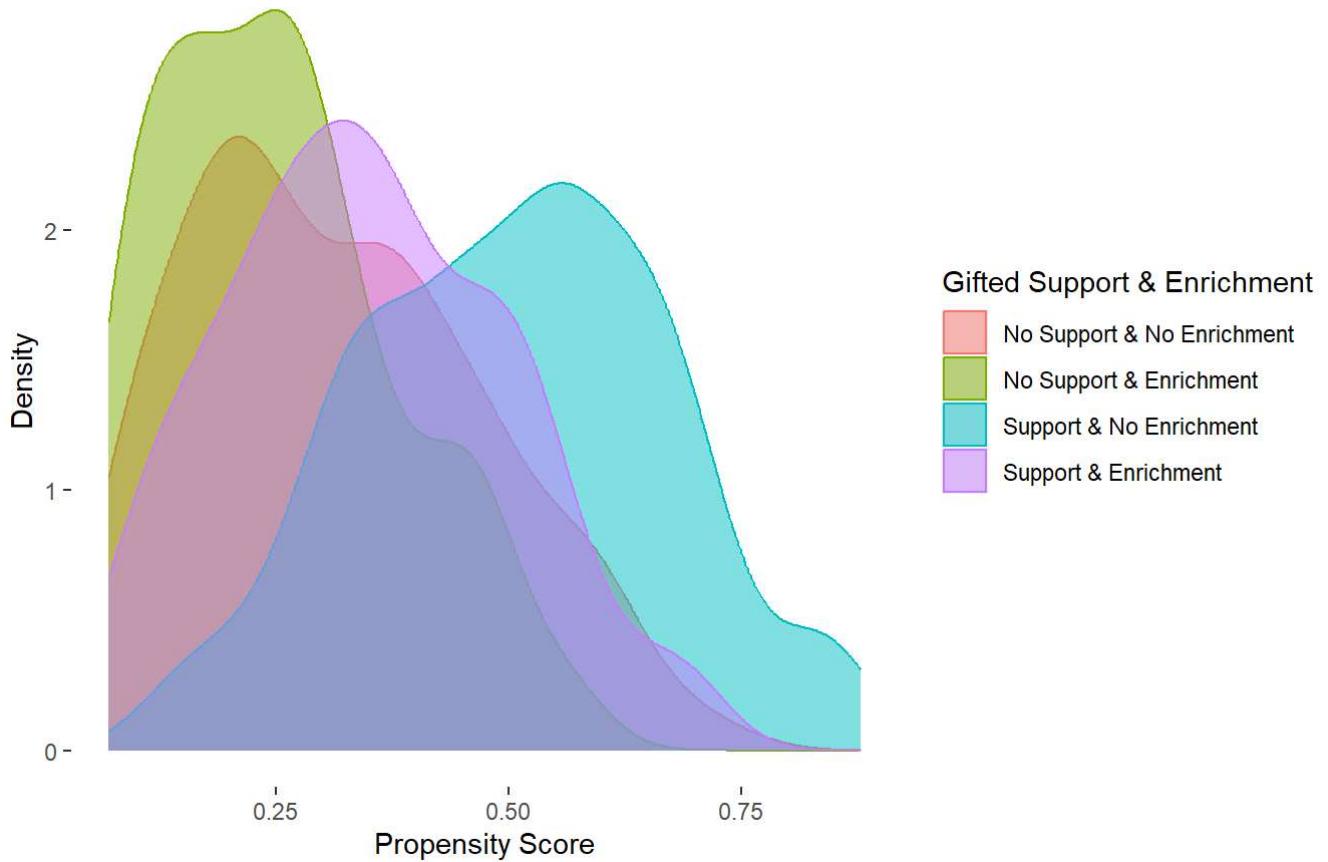
#> IQ_score           15.01        -0.38        0.80        0.37
#> Motivation         0.90        -0.11        0.04       -0.14
#> parents_academic  0.47        -0.20        0.12        0.23
#> sex                0.50        0.02       -0.02       -0.04
#>                  d YesYes-NoNo d YesYes-YNoYes d NoNo-NoYes
#> group_Long          -Inf        -Inf        -Inf
#> GPA_school          0.77        0.46       -0.33
#> IQ_score            1.22        0.75       -0.39
#> Motivation          0.16        -0.03       -0.19
#> parents_academic   0.33        0.44        0.11
#> sex                -0.04       -0.06       -0.02

#Estimating initial unbalance
unbalance_2x2 <- initial_unbalance(Data = MAGMA_sim_data,
                                       group = c("gifted_support", "enrichment"),
                                       covariates = covariates_2x2)
#> [1] "mean g was computed using robust variance meta-analysis with robumeta."
unbalance_2x2
#>      Pillai's Trace gifted_support Pillai's Trace enrichment
#> Unbalance             0.13              0.05
#>      Pillai's Trace IA d-ratio Mean g adj. d-ratio
#> Unbalance            0.01    0.53   0.29      0.53

#Estimating and plotting density overlap in gifted support & enrichment propensity score
Density_overlap(Data = MAGMA_sim_data,
                 variable = "ps_2x2",
                 group = c("gifted_support", "enrichment"),
                 variable_name = "Propensity Score",
                 group_labels = c("No Support & No Enrichment",
                                 "No Support & Enrichment",
                                 "Support & No Enrichment",
                                 "Support & Enrichment"),
                 group_name = "Gifted Support & Enrichment")
#> 2x2 groups are represented as 4 groups.

```

Density Plot for Propensity Score



```
#>                                     OV
#>                                     0.8692442
#> OVPairs.No Support & No Enrichment-No Support & Enrichment
#>                                     0.8264343
#> OVPairs.No Support & No Enrichment-Support & No Enrichment
#>                                     0.6062432
#> OVPairs.No Support & No Enrichment-Support & Enrichment
#>                                     0.8951293
#> OVPairs.No Support & Enrichment-Support & No Enrichment
#>                                     0.4478171
#> OVPairs.No Support & Enrichment-Support & Enrichment
#>                                     0.7450427
#> OVPairs.Support & No Enrichment-Support & Enrichment
#>                                     0.6573491
```

Standard Matching

After this check of initial unbalance, descriptive statistics, and area of common support we can conduct the matching. As mentioned above, we need to adapt the grouping variable and the distance variable/propensity score for this example. Moreover, conducting a 2x2/four-group matching has some computational load. If a RAM threshold is exceeded, MAGMA computes quasi-systematic matching. Again, this only affects the computational backend and does not affect the user application. Note that in the case of a multifactorial matching, MAGMA returns a fourth variable. This fourth variable is a multinomial version of the two independent variables (group_long).

```

MAGMA_sim_data_2x2 <- MAGMA(Data = MAGMA_sim_data,
                                group = c("gifted_support", "enrichment"),
                                dist = "ps_2x2",
                                cores = 2)
#> input correctly identified
#> Large Number of groups with Large group sizes. Computing quasi-systematic matching. Cores
were reduced to 2 to simplify node communication despite high RAM usage.
#> matching complete!
str(MAGMA_sim_data_2x2)
#> 'data.frame':   800 obs. of  18 variables:
#> $ ID                  : int  1 2 3 4 5 6 7 8 9 10 ...
#> $ sex                 : int  1 0 1 1 1 0 0 0 1 0 ...
#> $ gifted_support      : int  1 1 1 1 1 1 0 1 0 1 ...
#> $ teacher_ability_rating: int  3 2 2 1 3 3 2 2 1 1 ...
#> $ enrichment           : int  0 1 0 1 0 0 0 0 0 1 ...
#> $ parents_academic    : int  1 1 1 0 1 0 0 1 0 0 ...
#> $ GPA_school           : num  4.73 4.41 3.23 2.88 2.98 ...
#> $ IQ_score              : num  111.6 97.8 120.1 97.9 74.4 ...
#> $ Motivation            : num  5.36 4.41 3.57 4.34 3.54 ...
#> $ college_GPA           : num  4.01 3.41 2.73 4.02 4.09 ...
#> $ support_enrichment    : int  3 4 3 4 3 3 1 3 1 4 ...
#> $ ps_tar                : num  0.0657 0.1085 0.1899 0.4261 0.4782 ...
#> $ ps_2x2                : num  0.3 0.32 0.21 0.237 0.571 ...
#> $ ps_gifted              : num  0.431 0.364 0.626 0.353 0.187 ...
#> $ group_long             : num  1 2 1 2 1 1 3 1 3 2 ...
#> $ step                  : num  71 41 35 19 NA 12 NA NA NA 105 ...
#> $ weight                 : num  1 1 1 1 NA 1 NA NA NA 1 ...
#> $ distance               : num  3.61e-03 2.15e-04 1.24e-04 4.69e-05 NA ...

```

Balance Estimation and Visualization

The only difference to the former examples in the process is, that the grouping variable is a character vector of length two. As a result of this, `Plot_MAGMA` displays three Pillai's Trace plots corresponding to the two main effects and the interaction.

```

Balance_2x2 <- Balance_MAGMA(Data = MAGMA_sim_data_2x2,
                                group = c("gifted_support", "enrichment"),
                                covariates = covariates_2x2,
                                step = "step")
#> Start estimating Pillai's Trace.
#> Pillai's Trace finsihed. Starting to compute d-ratio.
#> d-ratio finsihed. Starting to compute mean-g.
#> [1] "mean g was computed using robust variance meta-analysis with robumeta."
#> mean-g finsihed. Starting to compute adjusted d-ratio.
#> finsihed balance estimation

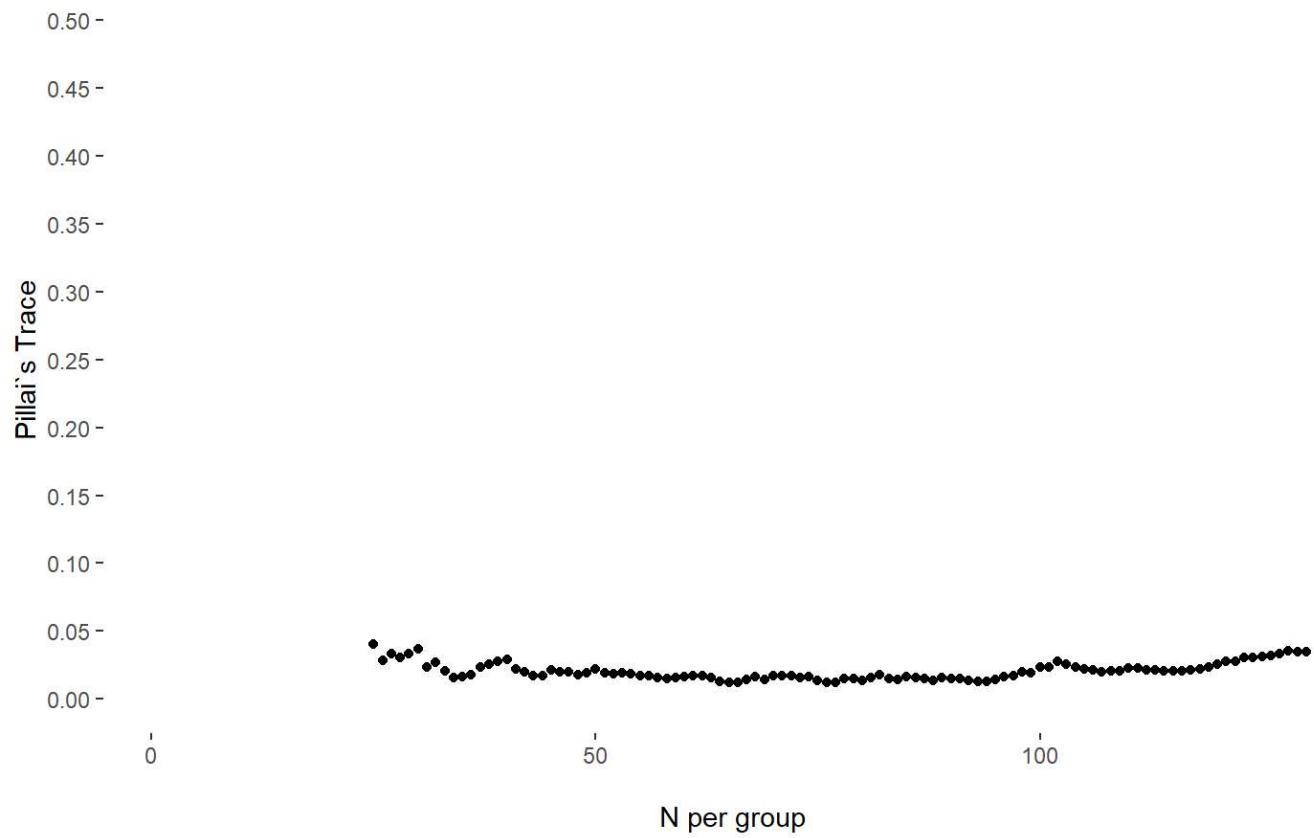
#Balance criteria for a 100 cases per group
Balance_100_2x2_criteria <- Balance_extract(Balance = Balance_2x2,
                                              samplesize = 100,
                                              effects = FALSE)
Balance_100_2x2_criteria
#> Pillai's Trace ME 1 Pillai's Trace ME 2 Pillai's Trace IA d-ratio
#>          0.02          0.01          0.02          0.73
#>      mean g     adj. d-ratio
#>          0.16          0.60

#Extracting pairwise effects for 100 cases per group
Balance_100_2x2_effects <- Balance_extract(Balance = Balance_2x2,
                                              samplesize = 100,
                                              effects = TRUE)
Balance_100_2x2_effects
#>      GPA_school_1_2      IQ_score_1_2      Motivation_1_2
#>          0.17          0.12          0.17
#> parents_academic_1_2      sex_1_2      GPA_school_1_3
#>          0.22          0.06          0.10
#>      IQ_score_1_3      Motivation_1_3 parents_academic_1_3
#>          0.03          0.21          0.10
#>      sex_1_3      GPA_school_1_4      IQ_score_1_4
#>          0.12          0.01          0.27
#>      Motivation_1_4 parents_academic_1_4      sex_1_4
#>          0.34          0.19          0.06
#>      GPA_school_2_3      IQ_score_2_3      Motivation_2_3
#>          0.07          0.09          0.03
#> parents_academic_2_3      sex_2_3      GPA_school_2_4
#>          0.12          0.18          0.19
#>      IQ_score_2_4      Motivation_2_4 parents_academic_2_4
#>          0.38          0.17          0.41
#>      sex_2_4      GPA_school_3_4      IQ_score_3_4
#>          0.12          0.12          0.29
#>      Motivation_3_4 parents_academic_3_4      sex_3_4
#>          0.14          0.29          0.06

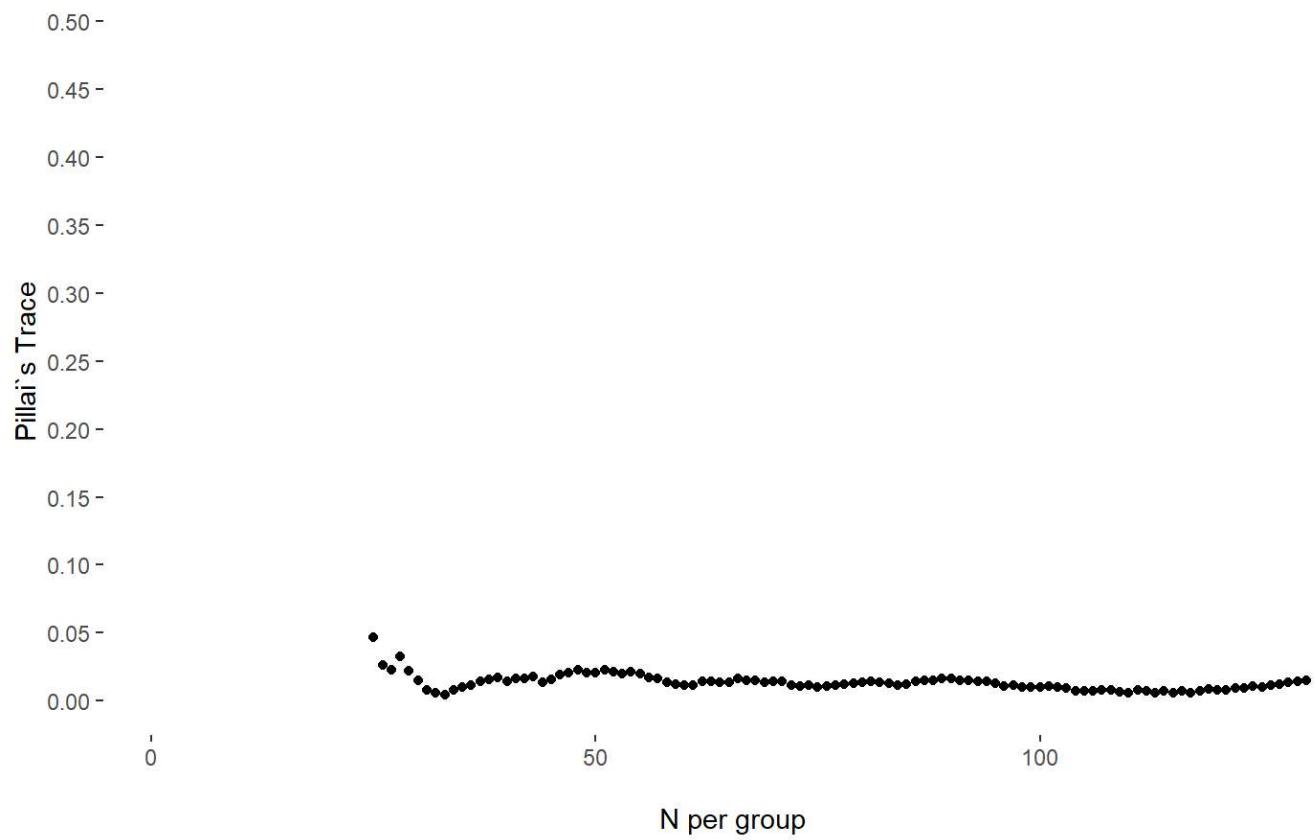
#Ploting trend over increasing samle size
Plot_MAGMA(Balance = Balance_2x2,
            criterion = c("Pillai", "d_ratio", "mean_g", "Adj_d_ratio"))

```

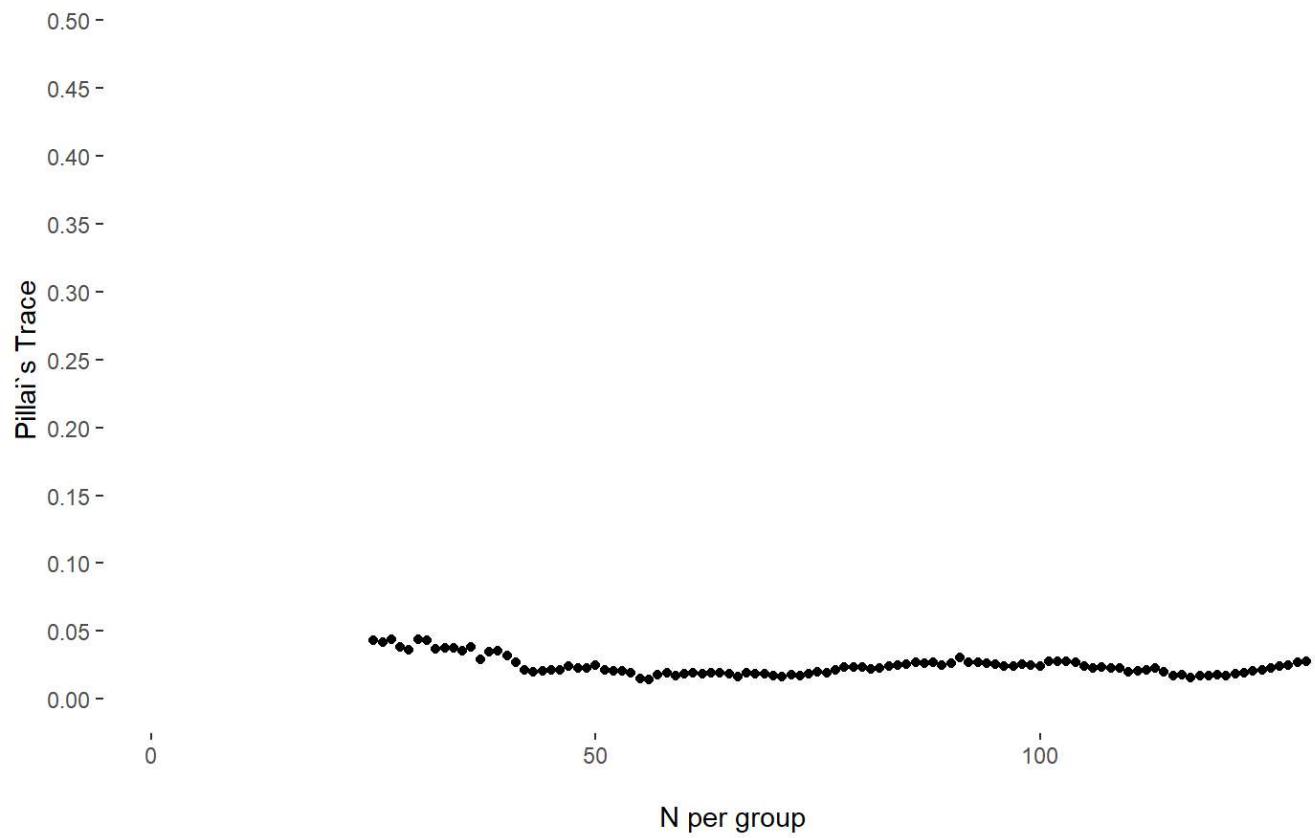
Pillai's Trace values for different sample sizes



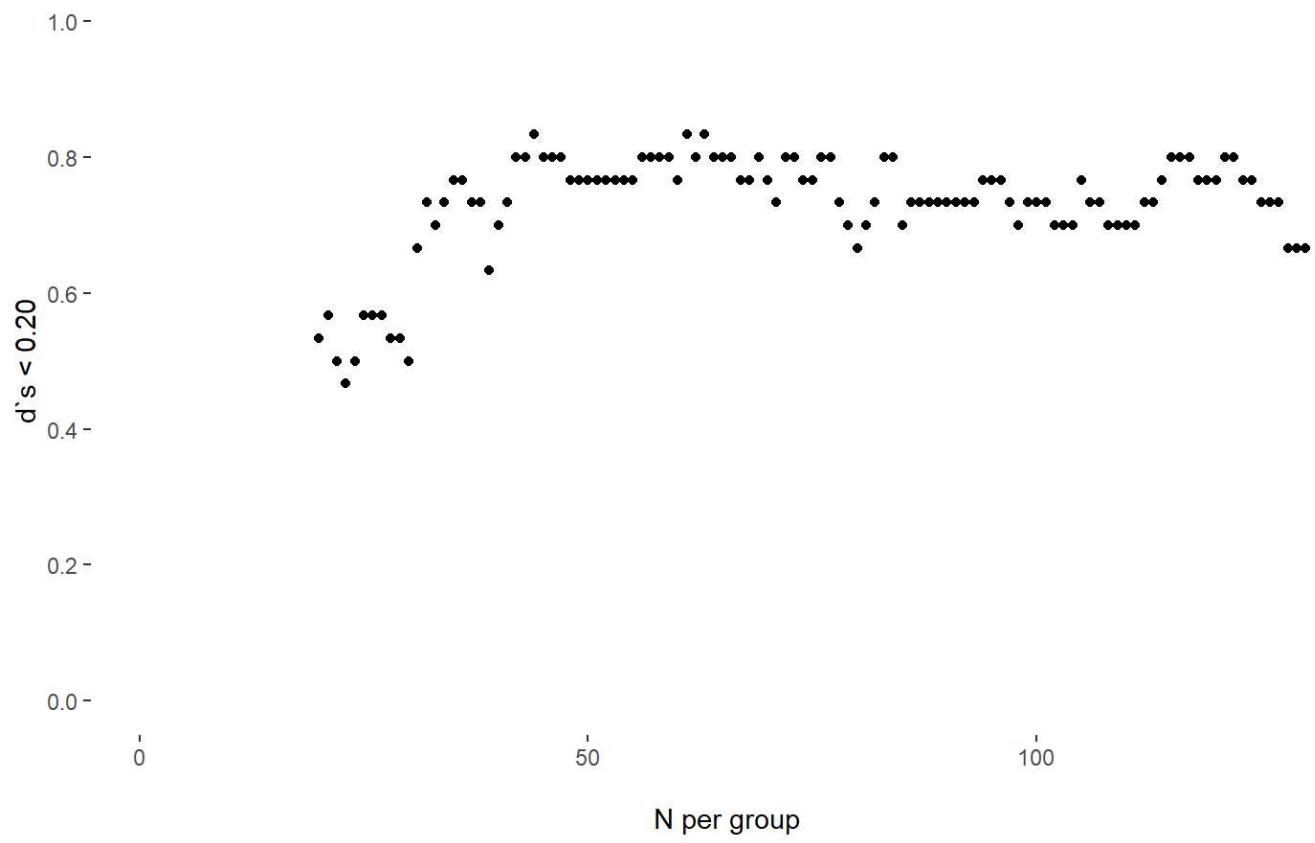
Pillai's Trace values for different sample sizes



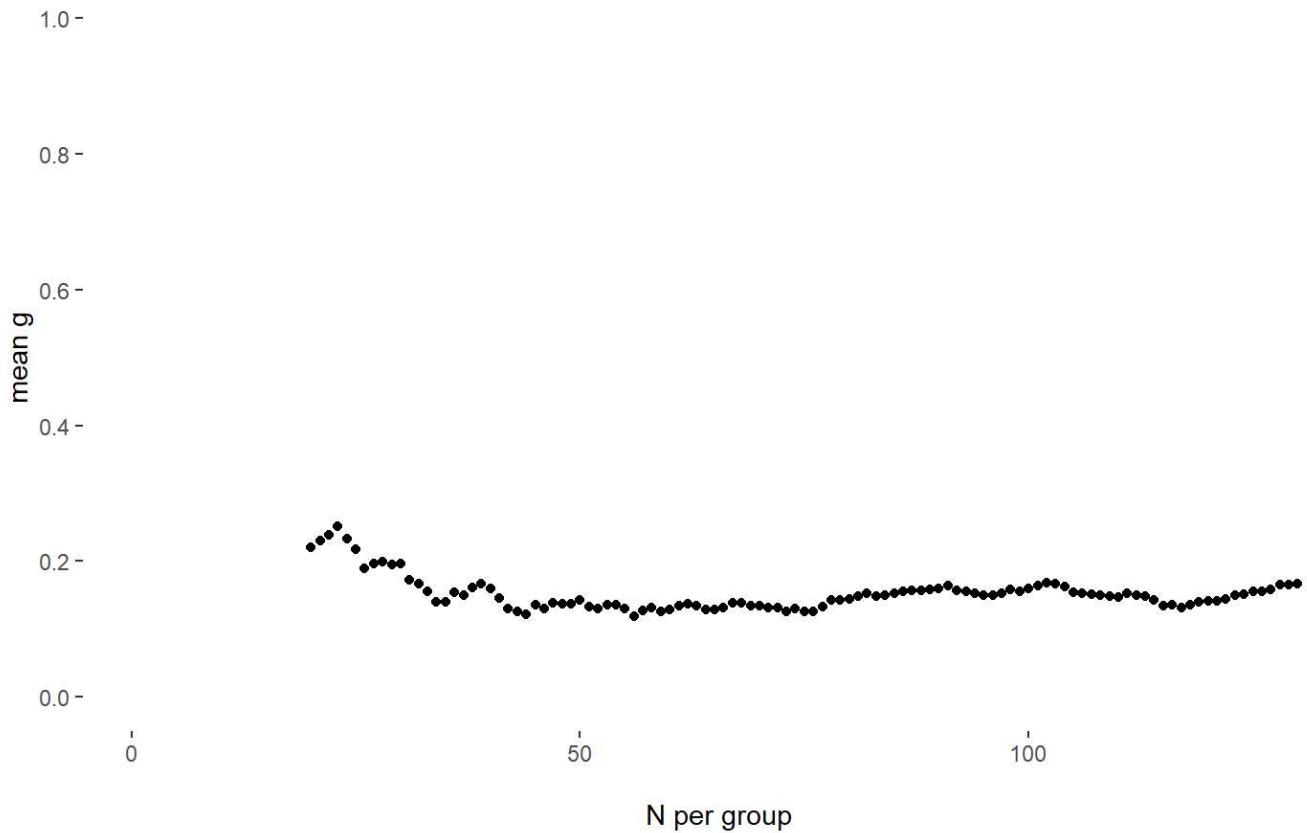
Pillai's Trace values for different sample sizes



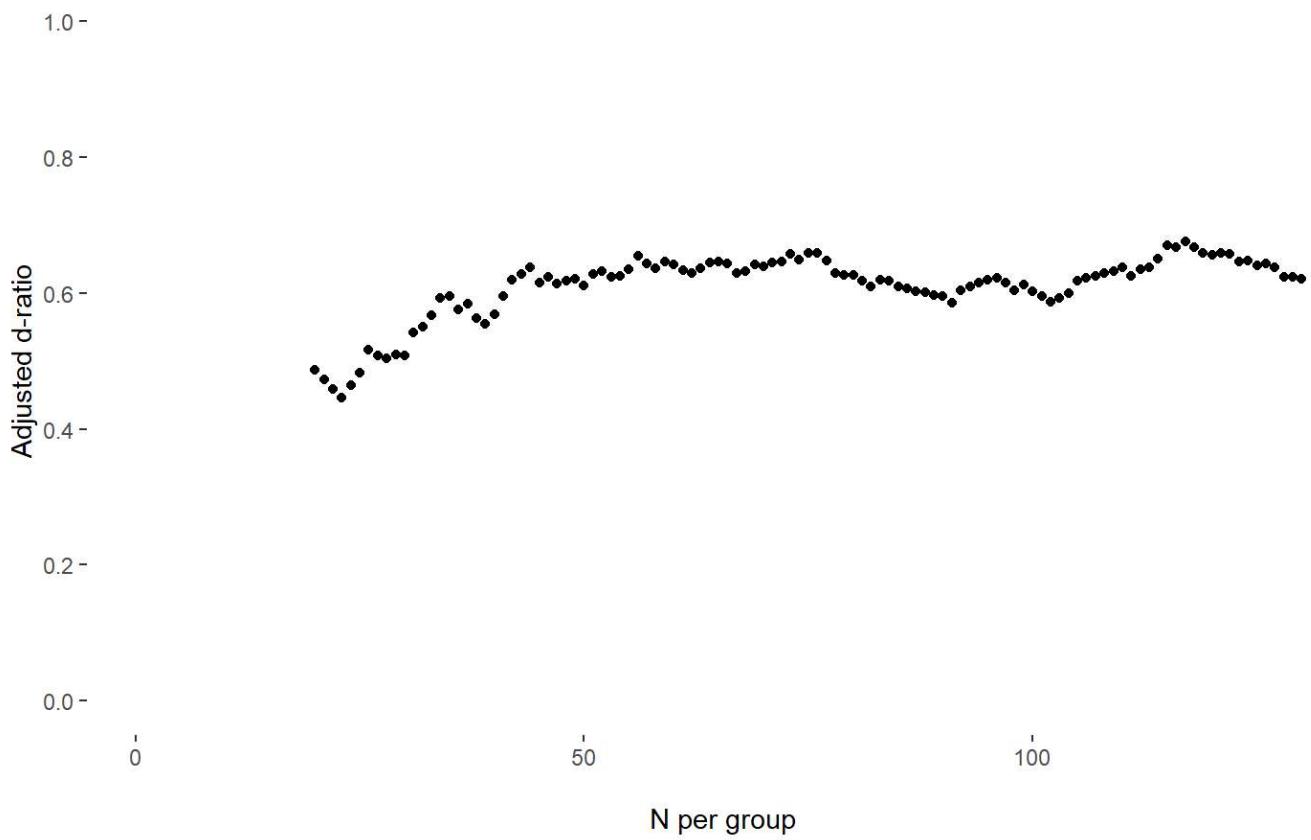
Cohen's d < .20` for different sample sizes



Mean effect for different sample sizes



Adjusted d-ratio for different sample sizes



```

#Creating table
Table_MAGMA(Balance = Balance_2x2,
             filename = "Balance_2x2.docx")
#> Balance Table successfully created!
#> # A tibble: 6 × 8
#>   Criterion_optimized Pillai_Trace_ME1 Pillai_Trace_ME2 Pillai_Trace_IA d_ratio
#>   <chr>                <dbl>            <dbl>            <dbl>            <dbl>
#> 1 Best Pillai ME 2      0.02              0                 0.04            0.7
#> 2 Best Pillai IA        0.02              0.02              0.01            0.8
#> 3 Best mean g          0.02              0.02              0.01            0.8
#> 4 Best d-ratio          0.02              0.01              0.02            0.833
#> 5 Best Pillai ME 1      0.01              0.01              0.02            0.8
#> 6 adjusted d-ratio       0.02              0.01              0.02            0.8
#> # i 3 more variables: mean_g <dbl>, adjusted_d_ratio <dbl>, n_per_group <int>

#Computing descriptive statistics and all pairwise effects after matching
descs_2x2_post <- MAGMA_desc(Data = MAGMA_sim_data_2x2,
                               group = c("gifted_support", "enrichment"),
                               covariates = covariates_2x2,
                               step_num = 100,
                               step_var = "step",
                               filename = "stats_post_pre.docx")
#> 2x2 groups are represented as 4 groups.

descs_2x2_post %>%
  purrr::set_names(c("Overall N", "Overall Mean", "Overall SD",
                    "Sup & No En N", "Sup & No En Mean",
                    "Sup & No En SD",
                    "Sup & En N", "Sup & En Mean", "Sup & En SD",
                    "No Sup & No En N", "No Sup & No En Mean", "No Sup & No En SD",
                    "No Sup & En N", "No Sup & En Mean", "No Sup & En SD",
                    "d YesNo-YesYes", "d YesNo-NoNo", "d YesNo-NoYes",
                    "d YesYes-NoNo", "d YesYes-YNoYes",
                    "d NoNo-NoYes"))

#> Overall N Overall Mean Overall SD Sup & No En N
#> group_Long      400     2.50    1.12    100
#> GPA_school      400     3.52    0.86    100
#> IQ_score         400    104.15   11.20    100
#> Motivation       400     3.90    0.85    100
#> parents_academic 400     0.44    0.50    100
#> sex               400     0.52    0.50    100
#> Sup & No En Mean Sup & No En SD Sup & En N Sup & En Mean
#> group_Long        1.00     0.00    100     2.00
#> GPA_school        3.46     0.90    100     3.61
#> IQ_score          104.55   10.59    100    105.81
#> Motivation         3.74     0.88    100     3.89
#> parents_academic  0.42     0.50    100     0.53
#> sex                 0.50     0.50    100     0.47
#> Sup & En SD No Sup & No En N No Sup & No En Mean
#> group_Long        0.00     100      3.00
#> GPA_school        0.86     100      3.55
#> IQ_score          10.29    100    104.85
#> Motivation         0.85     100      3.92
#> parents_academic  0.50     100      0.47

```

```

#> sex          0.50          100          0.56
#>              No Sup & No En SD No Sup & En N No Sup & En Mean
#> group_Long      0.00        100        4.00
#> GPA_school      0.89        100        3.46
#> IQ_score         10.79       100     101.40
#> Motivation       0.83        100        4.03
#> parents_academic 0.50        100        0.33
#> sex             0.50        100        0.53
#>              No Sup & En SD d YesNo-YesYes d YesNo-NoNo d YesNo-NoYes
#> group_Long      0.00        -Inf       -Inf       -Inf
#> GPA_school      0.78        -0.17      -0.10      0.00
#> IQ_score         12.66       -0.12      -0.03      0.27
#> Motivation       0.85        -0.17      -0.21      -0.34
#> parents_academic 0.47        -0.22      -0.10      0.19
#> sex             0.50        0.06      -0.12      -0.06
#>              d YesYes-NoNo d YesYes-YNoYes d NoNo-NoYes
#> group_Long      -Inf       -Inf       -Inf
#> GPA_school      0.07        0.18       0.11
#> IQ_score         0.09        0.38       0.29
#> Motivation      -0.04       -0.17      -0.13
#> parents_academic 0.12        0.41       0.29
#> sex             -0.18      -0.12       0.06

```

Exact Matching

We use the same covariates as we used for the standard 2x2 matching. The only change constitutes the use of `teacher_ability_rating` as exact variable. Below is the whole matching, balance estimation, and visualization process. Initial unbalance, descriptive statistics, and area of common support are the same as for standard matching.

```

MAGMA_sim_data_2x2_exact <- MAGMA_exact(Data = MAGMA_sim_data,
                                         group = c("gifted_support", "enrichment"),
                                         dist = "ps_2x2",
                                         exact = "teacher_ability_rating",
                                         cores = 2)

#> input correctly identified
#> matching complete!
str(MAGMA_sim_data_2x2_exact)
#> 'data.frame': 520 obs. of 18 variables:
#> $ ID : int 1 2 3 4 5 6 10 13 14 16 ...
#> $ sex : int 1 0 1 1 1 0 0 1 0 1 ...
#> $ gifted_support : int 1 1 1 1 1 1 1 0 0 0 ...
#> $ teacher_ability_rating: int 3 2 2 1 3 3 1 3 2 1 ...
#> $ enrichment : int 0 1 0 1 0 0 1 0 0 1 ...
#> $ parents_academic : int 1 1 1 0 1 0 0 0 0 0 ...
#> $ GPA_school : num 4.73 4.41 3.23 2.88 2.98 ...
#> $ IQ_score : num 111.6 97.8 120.1 97.9 74.4 ...
#> $ Motivation : num 5.36 4.41 3.57 4.34 3.54 ...
#> $ college_GPA : num 4.01 3.41 2.73 4.02 4.09 ...
#> $ support_enrichment : int 3 4 3 4 3 3 4 1 1 2 ...
#> $ ps_tar : num 0.0657 0.1085 0.1899 0.4261 0.4782 ...
#> $ ps_2x2 : num 0.3 0.32 0.21 0.237 0.571 ...
#> $ ps_gifted : num 0.431 0.364 0.626 0.353 0.187 ...
#> $ group_Long : num 1 2 1 2 1 1 2 3 3 4 ...
#> $ step : num 58 7 116 88 48 19 62 101 82 114 ...
#> $ weight : num 1 1 1 1 1 1 1 1 1 ...
#> $ distance : num 1.50e-03 1.76e-05 5.02e-01 9.73e-02 6.57e-04 ...

```

#Estimating Balance

```
Balance_2x2_exact <- Balance_MAGMA(Data = MAGMA_sim_data_2x2_exact,  
                                     group = c("gifted_support", "enrichment"),  
                                     covariates = covariates_2x2,  
                                     step = "step") #Not necessary to define here
```

#> Start estimating Pillai's Trace.

#> Pillai's Trace finished. Starting to compute d-ratio.

#> *d-ratio* finished. Starting to compute mean-*q*.

#> [1] "mean a was computed using robust variance meta-analysis with robumeta."

#> mean-a finished. Starting to compute adjusted d-ratio.

#> finished balance estimation

#Balance criteria for a 100 cases per group

Balance 100 2x2 criteria exact

```
#> Pillai's Trace ME 1 Pillai's Trace ME 2 Pillai's Trace IA d-ratio
#>          0.01          0.01          0.02          0.77
#>      mean g      adj. d-ratio
#>          0.12          0.69
```

#Extracting pairwise effects for 100 cases per group

```

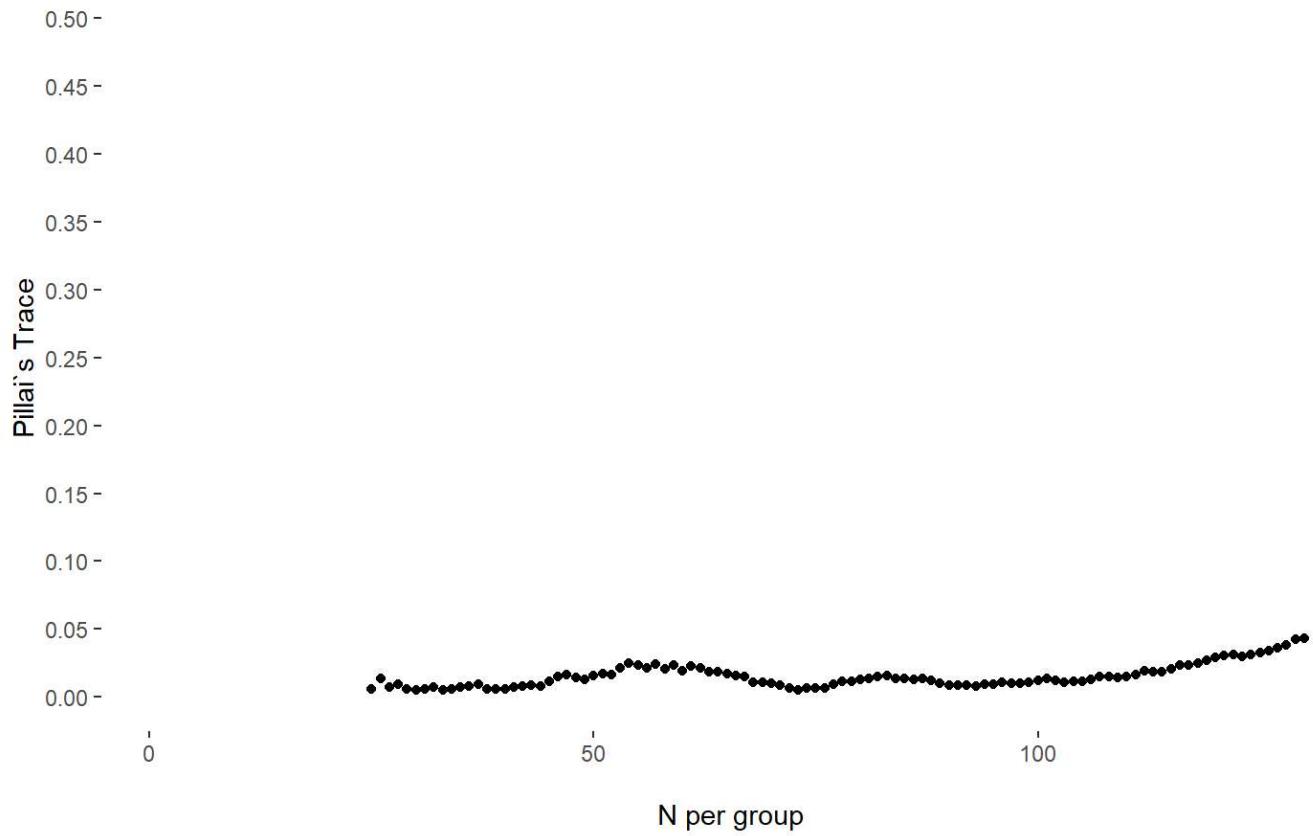
      effects = TRUE)

Balance_100_2x2_effects_exact
#>      GPA_school_1_2      IQ_score_1_2      Motivation_1_2
#>      0.12                  0.16                  0.17
#> parents_academic_1_2      sex_1_2      GPA_school_1_3
#>      0.04                  0.10                  0.10
#>      IQ_score_1_3      Motivation_1_3 parents_academic_1_3
#>      0.06                  0.23                  0.08
#>      sex_1_3      GPA_school_1_4      IQ_score_1_4
#>      0.02                  0.01                  0.18
#>      Motivation_1_4 parents_academic_1_4      sex_1_4
#>      0.21                  0.22                  0.02
#>      GPA_school_2_3      IQ_score_2_3      Motivation_2_3
#>      0.02                  0.10                  0.05
#> parents_academic_2_3      sex_2_3      GPA_school_2_4
#>      0.04                  0.12                  0.14
#>      IQ_score_2_4      Motivation_2_4 parents_academic_2_4
#>      0.34                  0.04                  0.26
#>      sex_2_4      GPA_school_3_4      IQ_score_3_4
#>      0.12                  0.12                  0.24
#>      Motivation_3_4 parents_academic_3_4      sex_3_4
#>      0.01                  0.30                  0.00

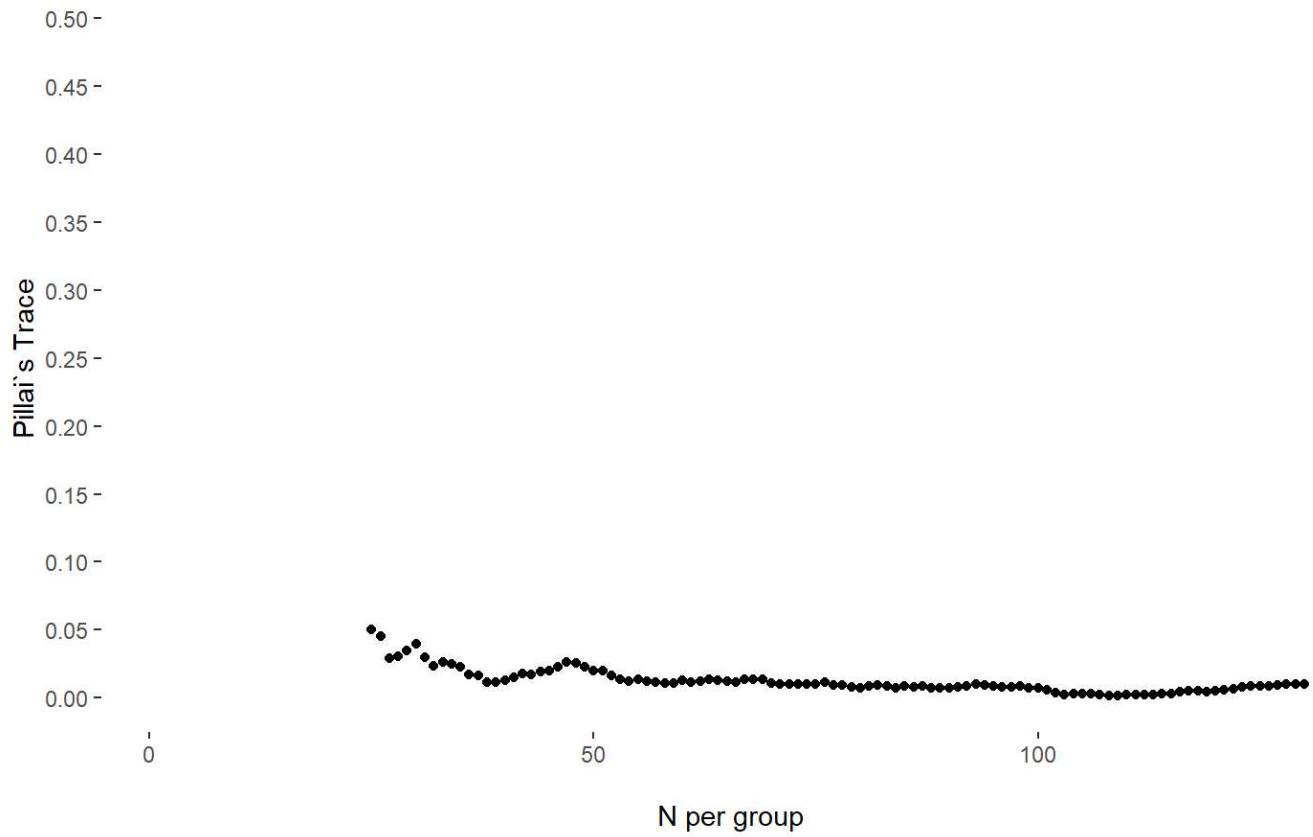
#Ploting trend over increasing sample size
Plot_MAGMA(Balance = Balance_2x2_exact,
            criterion = c("Pillai", "d_ratio", "mean_g", "Adj_d_ratio")) #Could be omitted

```

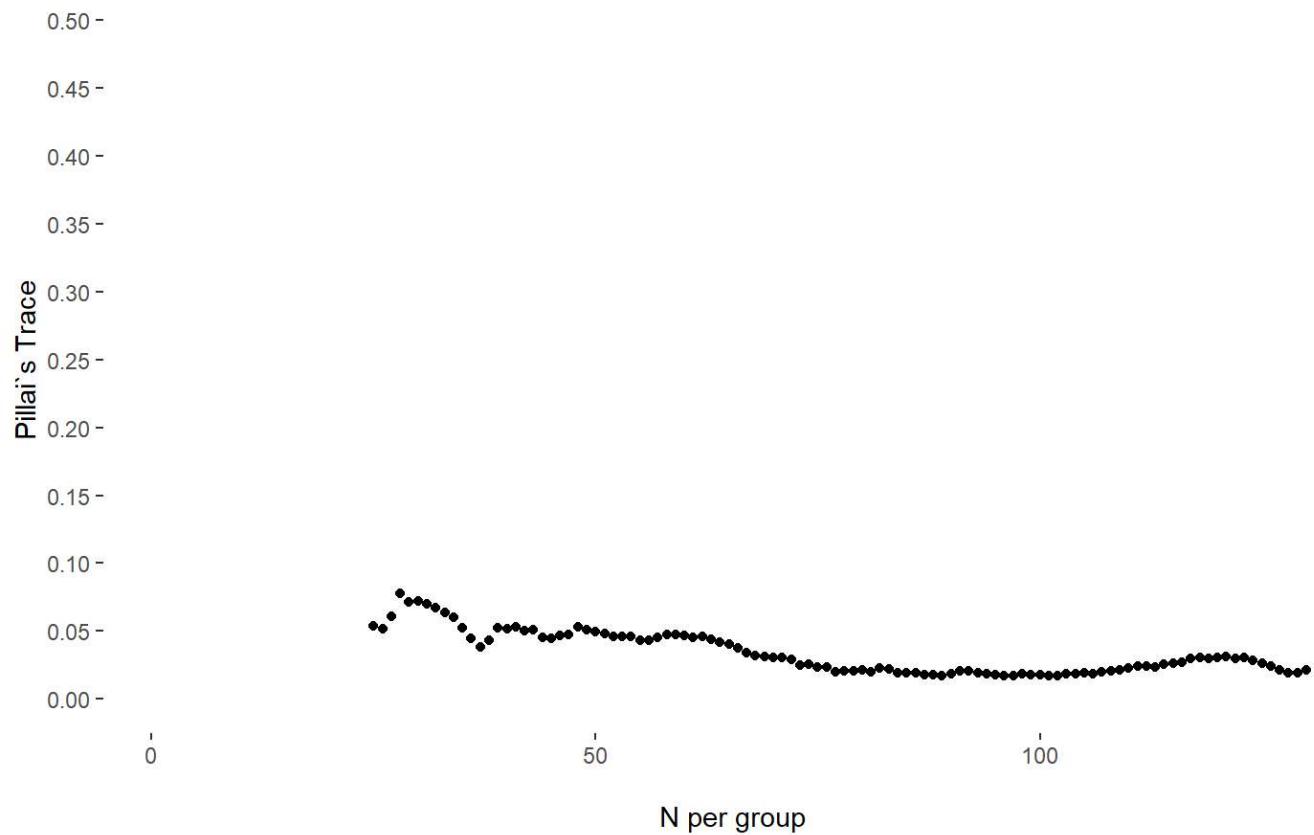
Pillai's Trace values for different sample sizes



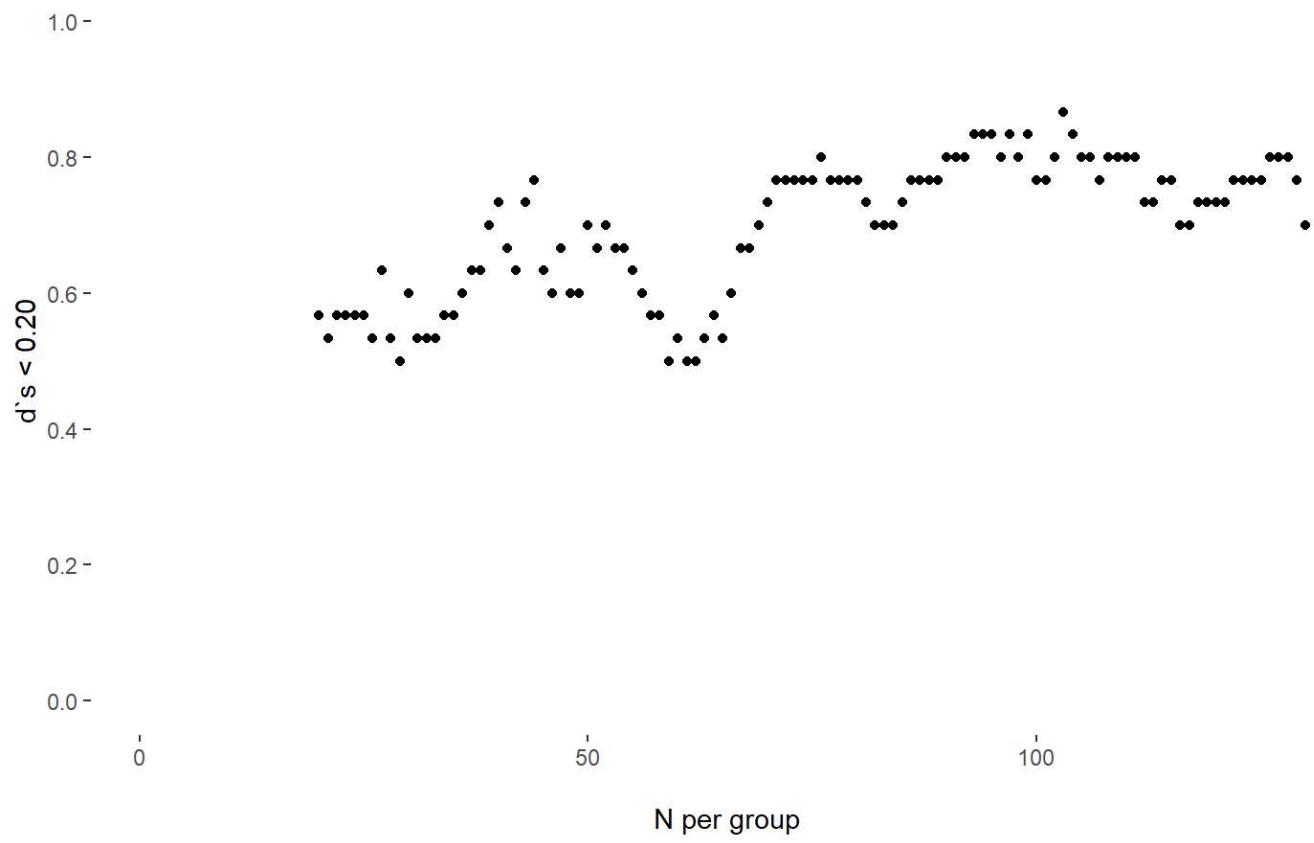
Pillai's Trace values for different sample sizes



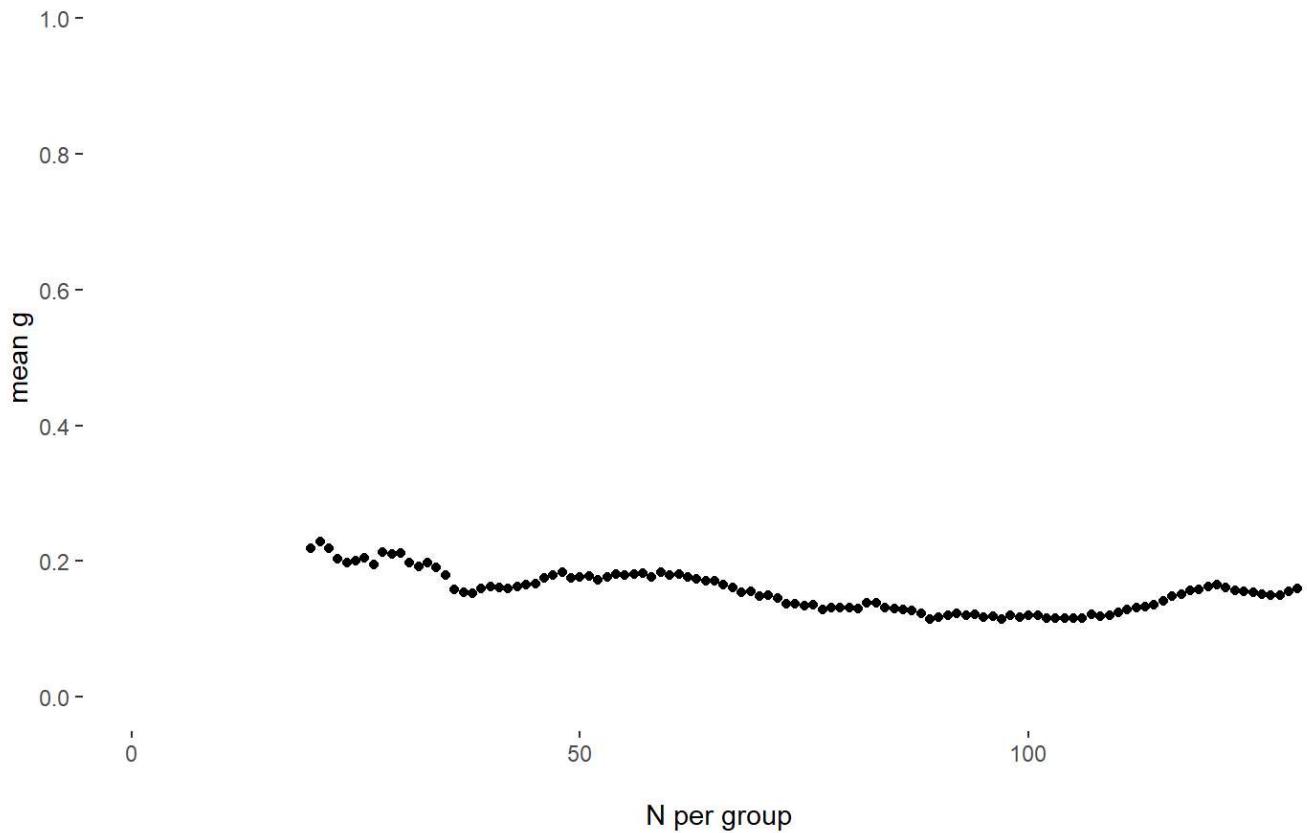
Pillai's Trace values for different sample sizes



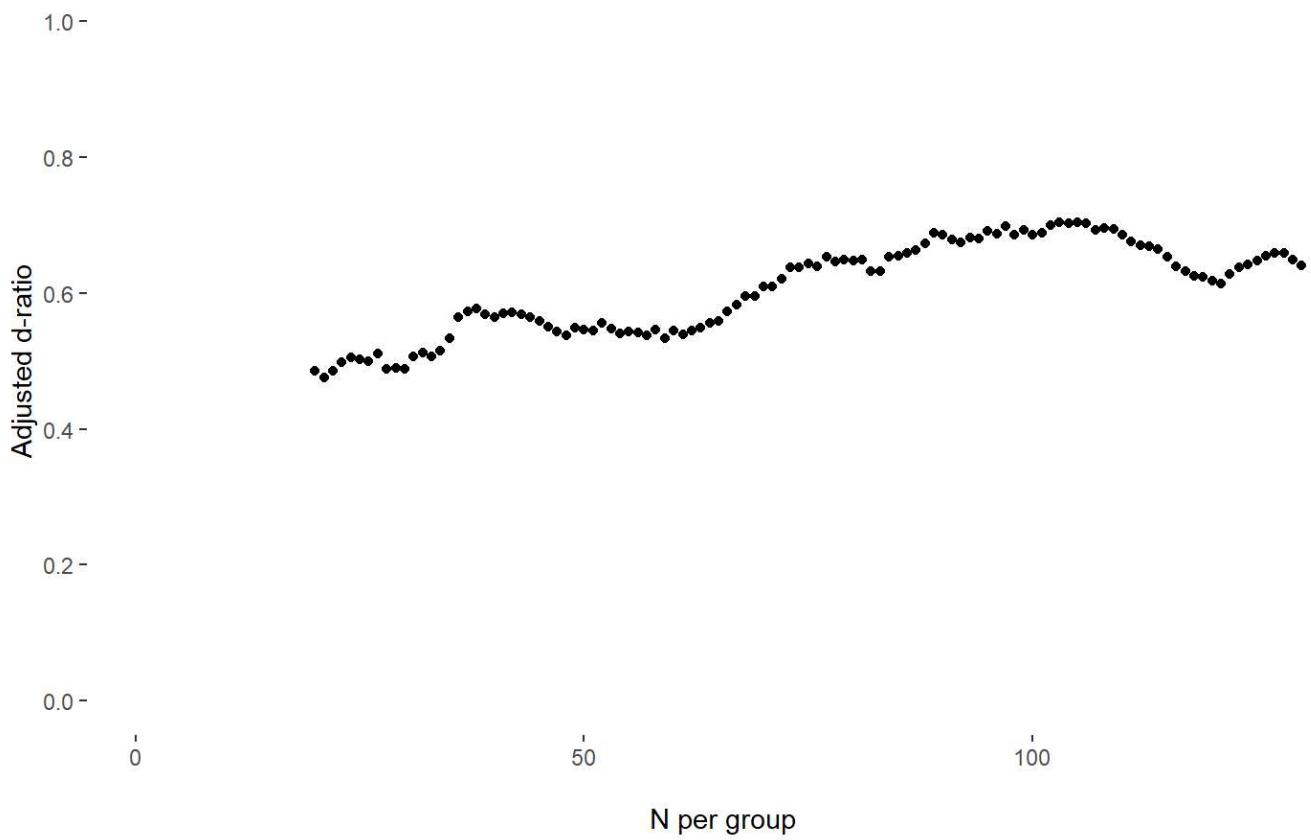
Cohen's d < .20` for different sample sizes



Mean effect for different sample sizes



Adjusted d-ratio for different sample sizes



```

#Creating table
Table_MAGMA(Balance = Balance_2x2_exact,
             filename = "Balance_2x2_exact.docx")
#> Balance Table successfully created!
#> # A tibble: 6 × 8
#>   Criterion_optimized Pillai_Trace_ME1 Pillai_Trace_ME2 Pillai_Trace_IA d_ratio
#>   <chr>                <dbl>            <dbl>            <dbl>            <dbl>
#> 1 Best Pillai ME 1      0                 0.04            0.07            0.6
#> 2 Best Pillai IA       0.01              0.01            0.02            0.833
#> 3 Best mean g          0.01              0.01            0.02            0.833
#> 4 Best d-ratio         0.01              0                 0.02            0.867
#> 5 adjusted d-ratio     0.01              0                 0.02            0.8
#> 6 Best Pillai ME 2      0.01              0                 0.02            0.8
#> # i 3 more variables: mean_g <dbl>, adjusted_d_ratio <dbl>, n_per_group <int>

#Computing descriptive statistics and all pairwise effects post matching
descs_2x2_post <- MAGMA_desc(Data = MAGMA_sim_data_2x2_exact,
                               group = c("gifted_support", "enrichment"),
                               covariates = covariates_2x2,
                               step_num = 100,
                               step_var = "step",
                               filename = "stats_2x2_post.docx")
#> 2x2 groups are represented as 4 groups.

descs_2x2_post %>%
  purrr::set_names(c("Overall N", "Overall Mean", "Overall SD",
                     "Sup & No En N", "Sup & No En Mean",
                     "Sup & No En SD",
                     "Sup & En N", "Sup & En Mean", "Sup & En SD",
                     "No Sup & No En N", "No Sup & No En Mean", "No Sup & No En SD",
                     "No Sup & En N", "No Sup & En Mean", "No Sup & En SD",
                     "d YesNo-YesYes", "d YesNo-NoNo", "d YesNo-NoYes",
                     "d YesYes-NoNo", "d YesYes-YNoYes",
                     "d NoNo-NoYes"))

#> Overall N Overall Mean Overall SD Sup & No En N
#> group_Long        400      2.50      1.12      100
#> GPA_school        400      3.54      0.86      100
#> IQ_score          400    104.14     11.18      100
#> Motivation        400      3.90      0.85      100
#> parents_academic 400      0.46      0.50      100
#> sex                400      0.50      0.50      100
#> Sup & No En Mean Sup & No En SD Sup & En N Sup & En Mean
#> group_Long        1.00      0.00      100      2.00
#> GPA_school        3.50      0.92      100      3.60
#> IQ_score          104.05    11.57      100    105.81
#> Motivation        3.77      0.86      100      3.92
#> parents_academic 0.47      0.50      100      0.49
#> sex                0.51      0.50      100      0.46
#> Sup & En SD No Sup & No En N No Sup & No En Mean
#> group_Long        0.00      100      3.00
#> GPA_school        0.86      100      3.59
#> IQ_score          10.35     100    104.74
#> Motivation        0.85      100      3.96
#> parents_academic 0.50      100      0.51

```

#> sex	0.50	100	0.52
#>	No Sup & No En SD	No Sup & En N	No Sup & En Mean
#> group_Long	0.00	100	4.00
#> GPA_school	0.89	100	3.49
#> IQ_score	10.56	100	101.97
#> Motivation	0.84	100	3.95
#> parents_academic	0.50	100	0.36
#> sex	0.50	100	0.52
#>	No Sup & En SD d YesNo-YesYes d YesNo-NoNo d YesNo-NoYes		
#> group_Long	0.00	-Inf	-Inf
#> GPA_school	0.76	-0.11	-0.10
#> IQ_score	11.99	-0.16	-0.06
#> Motivation	0.85	-0.18	-0.22
#> parents_academic	0.48	-0.04	-0.08
#> sex	0.50	0.10	-0.02
#>	d YesYes-NoNo d YesYes-YNoYes d NoNo-NoYes		
#> group_Long	-Inf	-Inf	-Inf
#> GPA_school	0.01	0.14	0.12
#> IQ_score	0.10	0.34	0.25
#> Motivation	-0.05	-0.04	0.01
#> parents_academic	-0.04	0.27	0.31
#> sex	-0.12	-0.12	0.00

References

- Austin, P. C. (2014). A comparison of 12 algorithms for matching on the propensity score. *Statistics in Medicine*, 33(6), 1057–1069. <https://doi.org/10.1002/sim.6004> (<https://doi.org/10.1002/sim.6004>)
- Feuchter, M. D., Urban, J., Scherrer, V., Breit, M., & Preckel, F. (2022). Introduction and Demonstration of the Many-Group Matching (MAGMA)-Algorithm: Matching Solutions for Two or More Groups. <https://doi.org/10.17605/OSF.IO/AEDXB> (<https://doi.org/10.17605/OSF.IO/AEDXB>)
- Imai, K., & van Dyk, D. A. (2004). Causal Inference With General Treatment Regimes. *Journal of the American Statistical Association*, 99(467), 854–866. <https://doi.org/10.1198/016214504000001187> (<https://doi.org/10.1198/016214504000001187>)
- Jacovidis, J. N. (2017). Evaluating the performance of propensity score matching methods: A simulation study [Doctoral Dissertation]. James Madison University.
- Li, M. (2013). Using the Propensity Score Method to Estimate Causal Effects. *Organizational Research Methods*, 16(2), 188–226. <https://doi.org/10.1177/1094428112447816> (<https://doi.org/10.1177/1094428112447816>)
- McCaffrey, D. F [Daniel F.], Griffin, B. A [Beth Ann], Almirall, D., Slaughter, M. E., Ramchand, R., & Burgette, L. F [Lane F.] (2013). A tutorial on propensity score estimation for multiple treatments using generalized boosted models. *Statistics in Medicine*, 32(19), 3388–3414. <https://doi.org/10.1002/sim.5753> (<https://doi.org/10.1002/sim.5753>)
- Pastore M, Loro PAD, Mingione M, Calcagni' A (2022). *overlapping: Estimation of Overlapping in Empirical Distributions*. R package version 2.1. <https://CRAN.R-project.org/package=overlapping> (<https://CRAN.R-project.org/package=overlapping>)
- Powell, M. G., Hull, D. M., & Beaujean, A. A. (2020). Propensity Score Matching for Education Data: Worked Examples. *The Journal of Experimental Education*, 88(1), 145–164. <https://doi.org/10.1080/00220973.2018.1541850> (<https://doi.org/10.1080/00220973.2018.1541850>)

Ridgeway, G., McCaffrey, D. F [D. F.], Morral, A. R., Cefalu, M., Burgette, L. F [L. F.], Pane, J. D., & Griffin, B. A [B. A.]. (2015). Toolkit for Weighting and Analysis of Nonequivalent Groups: A tutorial for the twang package. Rand.

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1), 41–55. <https://doi.org/10.1093/biomet/70.1.41> (<https://doi.org/10.1093/biomet/70.1.41>)

Rosenbaum, P. R., & Rubin, D. B. (1985). Constructing a Control Group Using Multivariate Matched Sampling Methods That Incorporate the Propensity Score. *The American Statistician*, 39(1), 33–38.

Stuart, E. A. (2010). Matching methods for causal inference: A review and a look forward. *Statistical Science : A Review Journal of the Institute of Mathematical Statistics*, 25(1), 1–21. <https://doi.org/10.1214/09-STS313> (<https://doi.org/10.1214/09-STS313>)

Thoemmes, F. J., & Kim, E. S. (2011). A Systematic Review of Propensity Score Methods in the Social Sciences. *Multivariate Behavioral Research*, 46(1), 90–118. <https://doi.org/10.1080/00273171.2011.540475> (<https://doi.org/10.1080/00273171.2011.540475>)