Example run maicplus

Introduction

This package describes the steps required to perform a matching-adjusted indirect comparison (MAIC) analysis using the `maicplus` package in R where the endpoint of interest is either time-to-event (e.g. overall survival) or binary (e.g. objective tumor response).

The methods described in this document are based on those originally described by Signorovitch et al. 2010 and described in the National Institute for Health and Care Excellence (NICE) Decision Support Unit (DSU) Technical Support Document (TSD) 18. [signorovitch2010; phillippo2016a]

MAIC methods are often required when:

- There is no common comparator treatment to link a clinical trial of a new intervention to clinical trials of other treatments in a given disease area. For example if the only study of a new intervention is a single arm trial with no control group. This is commonly referred to as an unanchored MAIC.
- A common comparator is available to link a clinical trial of a new intervention to a clinical trial of one other treatment in a given disease area but there are substantial differences in patient demographic or disease characteristics that are believed to be treatment effect modifiers. This is commonly referred to as an anchored MAIC.

The premise of MAIC methods is to adjust for between-trial differences in patient demographic or disease characteristics at baseline. When a common treatment comparator or 'linked network' are unavailable, a MAIC assumes that differences between absolute outcomes that would be observed in each trial are entirely explained by imbalances in prognostic variables and treatment effect modifiers. Prognostic variables are those that are predictive of disease outcomes, independent of the treatment received. For example, older patients may have increased risk of death compared to younger patients. Treatment effect modifiers are those variables that influence the relative effect of one treatment compared to another. For example patients with a better performance status may experience a larger treatment benefit than those with a worse performance status. Under this assumption, every prognostic variable and every treatment effect modifier that is imbalanced between the two studies must be available. This assumption is generally considered very difficult to meet. [phillippo2016a] There are several ways of identifying prognostic variables/treatment effect modifiers to be used in the MAIC analyses, some of which include:

- Clinical expertise (when available to a project)
- Published papers/previous submissions (what has been identified in the disease area previously)
- Univariable/multivariable regression analyses to identify which covariates have a significant effect on the outcome
- Subgroup analyses of clinical trials may identify interactions between patient characteristics and the relative treatment effect

Theory behind MAIC

We will briefly go over the theory behind MAIC. For detailed information, see Signorovitch et al. 2010.

Let us define t_i to be the treatment patient i received. We assume $t_i = 0$ if the patient received intervention (IPD) and $t_i = 1$ if the patient received comparator treatment. The causal effect of treatment $T = 0$ vs $T = 1$ on the mean of the outcome Y can be estimated as below

$$\frac{\sum_{i=1}^{n} y_i(1-t_i)w_i}{\sum_{i=1}^{n}(1-t_i)w_i} - \bar{y}_1$$

where $w_i = \frac{Pr(T_i=1|x_i)}{Pr(T_i=0|x_i)}$ is the odds that patient i received treatment $T = 1$ vs $T = 0$ (i.e. enrolls in aggregate data study vs IPD study) given baseline characteristics x_i. Thus, the patients receiving $T = 0$ are re-weighted to match the distribution of patients receiving $T = 1$. Note that this causal effect would be the case when the outcome Y is continuous. If the outcome is binary, Y would be a proportion and we would use a link function such as logit to give us the causal effect in an odds ratio scale. As in propensity score methods, we may assume w_i to follow logistic regression form

$$w_i = exp(x_i^T \beta)$$

However, in order to estimate β, we cannot use maximum likelihood approach because we do not have IPD for both trials. Instead, we use method of moments. We estimate β such that the weighted averages of the covariates in the IPD exactly matches the aggregate data averages. Mathematically speaking, we want to estimate β such that:

$$0 = \frac{\sum_{i=1}^{n} x_i exp(x_i^T \hat{\beta})}{\sum_{i=1}^{n} exp(x_i^T \hat{\beta})} - \bar{x}_{agg}$$

If the x_i contains all confounders and the logistic regression for w_i is correctly specified, we obtain a consistent estimate of the causal effect of intervention vs comparator treatment. Above equation is equivalent to

$$0 = \sum_{i=1}^{n}(x_i - \bar{x}_{agg})exp(x_i^T \hat{\beta})$$

We could transform transform IPD by subtracting the aggregate data means (this is why centering is needed when preprocessing).

$$0 = \sum_{i=1}^{n} x_i exp(x_i^T \hat{\beta})$$

Note that this is the first derivative of

$$Q(\beta) = \sum_{i=1}^{n} exp(x_i^T \hat{\beta})$$

which has second derivative

$$Q''(\beta) = \sum_{i=1}^{n} x_i x_i^T exp(x_i^T \hat{\beta})$$

Since $Q''(\beta)$ is positive-definite for all β, $Q(\beta)$ is convex and any finite solution from the equation is unique and corresponds to the global minimum of $Q(\beta)$. Thus, we can use optimization methods to calculate β.

Example scenario

We present an unanchored MAIC of two treatments in lung cancer. The two endpoints being compared are overall survival (a time to event outcome) and objective response (a binary outcome). The data available are:

- Individual patient data from a single arm study
- Aggregate summary data for the comparator study
- Psuedo patient data from the comparator study. This is not required for the matching process but is needed to derive the relative treatment effects between the internal treatment and comparator treatment.

Preprocessing

Package load

```r
# change directory
setwd("~/GitHub/maicplus")
devtools::load_all()

# devtools::install_github('hta-pharma/maicplus') library(maicplus)

library(dplyr)  # this is used for data merging/cleaning. Package itself does not depend on dplyr

library(clubSandwich)  # For robust standard error in logistic regression
library(sandwich)

library(survminer)  # for ggsurvplot
library(ggplot2)  # for ggplot functions
library(boot)  # for bootstrapping
```

Preprocessing IPD

In this example scenario, age, sex, the Eastern Cooperative Oncology Group (ECOG) performance status, smoking status, and number of previous treatments have been identified as imbalanced prognostic variables/treatment effect modifiers.

This example reads in and combines data from three standard simulated data sets (adsl, adrs and adtte) which are saved as '.csv' files.

```r
adsl <- read.csv(system.file("extdata", "adsl.csv", package = "maicplus", mustWork = TRUE))
adrs <- read.csv(system.file("extdata", "adrs.csv", package = "maicplus", mustWork = TRUE))
adtte <- read.csv(system.file("extdata", "adtte.csv", package = "maicplus", mustWork = TRUE))

# Data containing the matching variables
adsl <- adsl %>%
    mutate(SEX_MALE = ifelse(SEX == "Male", 1, 0)) %>%
    mutate(AGE_SQUARED = AGE^2)

# Could use built-in function for dummizing variables adsl <- dummize_ipd(adsl,
```

```r
# dummize_cols=c('SEX'), dummize_ref_level=c('Female'))

# Response data
adrs <- adrs %>%
    filter(PARAM == "Response") %>%
    transmute(USUBJID, ARM, RESPONSE = AVAL)

# Time to event data (overall survival)
adtte <- adtte %>%
    filter(PARAMCD == "OS") %>%
    mutate(EVENT = 1 - CNSR) %>%
    transmute(USUBJID, ARM, TIME = AVAL, EVENT)

# Combine all ipd data
ipd <- adsl %>%
    full_join(adrs, by = c("USUBJID", "ARM")) %>%
    full_join(adtte, by = c("USUBJID", "ARM"))
head(ipd)
```

```
##   X USUBJID ARM AGE    SEX SMOKE ECOG0 N_PR_THER SEX_MALE AGE_SQUARED RESPONSE
## 1 1       1   A  45   Male     0     0         4        1        2025        0
## 2 2       2   A  71   Male     0     0         3        1        5041        1
## 3 3       3   A  58   Male     1     1         2        1        3364        1
## 4 4       4   A  48 Female     0     1         4        0        2304        1
## 5 5       5   A  69   Male     0     1         4        1        4761        0
## 6 6       6   A  48 Female     0     1         4        0        2304        0
##       TIME EVENT
## 1 281.5195     0
## 2 500.0000     0
## 3 304.6406     0
## 4 102.4731     0
## 5 101.6632     0
## 6 237.0593     1
```

Preprocessing aggregate data

There are two ways of specifying aggregate data. One approach is to read in aggregate data using an excel spreadsheet. In the spreadsheet, possible variable types include mean, median, or standard deviation for continuous variables and count or proportion for binary variables. The naming should be followed by these suffixes accordingly: _COUNT, _MEAN, _MEDIAN, _SD, _PROP. Then, `process_agd` will convert the count into proportions.

Other way is to define data frame of aggregate data in R. If you do it this way, _COUNT prefix should not be specified and only proportion is allowed for binary variables. Other suffix names would be the same as the first approach.

Possible missingness in the binary variables should be accounted for by subtracting the denominator by the missing count i.e. proportion = count / (N - missing).

```r
# Through excel spreadsheet target_pop <-
# read.csv(system.file('extdata','aggregate_data_example_1.csv', package =
# 'maicplus', mustWork = TRUE)) agd <- process_agd(target_pop)
```

```r
# Second approach by defining a data frame in R
agd <- data.frame(STUDY = "Lung study", ARM = "Total", N = 300, AGE_MEAN = 51, AGE_MEDIAN = 49,
    AGE_SD = 3.25, SEX_MALE_PROP = 147/300, ECOG0_PROP = 0.4, SMOKE_PROP = 58/(300 -
        5), N_PR_THER_MEDIAN = 2)
```

Preprocessing aggregate data

```r
#### prepare data
ipd_centered <- center_ipd(ipd = ipd, agd = agd)
head(ipd_centered)
```

```
##   X USUBJID ARM AGE    SEX SMOKE ECOG0 N_PR_THER SEX_MALE AGE_SQUARED RESPONSE
## 1 1       1   A  45   Male     0     0         4        1        2025        0
## 2 2       2   A  71   Male     0     0         3        1        5041        1
## 3 3       3   A  58   Male     1     1         2        1        3364        1
## 4 4       4   A  48 Female     0     1         4        0        2304        1
## 5 5       5   A  69   Male     0     1         4        1        4761        0
## 6 6       6   A  48 Female     0     1         4        0        2304        0
##       TIME EVENT AGE_CENTERED AGE_MEDIAN_CENTERED AGE_SQUARED_CENTERED
## 1 281.5195     0           -6                -0.5            -586.5625
## 2 500.0000     0           20                 0.5            2429.4375
## 3 304.6406     0            7                 0.5             752.4375
## 4 102.4731     0           -3                -0.5            -307.5625
## 5 101.6632     0           18                 0.5            2149.4375
## 6 237.0593     1           -3                -0.5            -307.5625
##   SEX_MALE_CENTERED ECOG0_CENTERED SMOKE_CENTERED N_PR_THER_MEDIAN_CENTERED
## 1              0.51           -0.4     -0.1966102                       0.5
## 2              0.51           -0.4     -0.1966102                       0.5
## 3              0.51            0.6      0.8033898                      -0.5
## 4             -0.49            0.6     -0.1966102                       0.5
## 5              0.51            0.6     -0.1966102                       0.5
## 6             -0.49            0.6     -0.1966102                       0.5
```

How to handle standard deviation aggregate summary

As described by Phillippo et al. 2016, balancing on both mean and standard deviation for continuous variables (where possible) may be considered in some cases. If a standard deviation is provided in the comparator population, preprocessing is done so that in the target population, $E(X^2)$ is calculated using the variance formula $Var(X) = E(X^2) - E(X)^2$. This $E(X^2)$ in the target population is matched with the IPD level data, which is why X^2 was calculated during the preprocessing stage of IPD.

How to handle median aggregate summary

If a median is provided, IPD is preprocessed to categorize the variable into a binary variable. All the values in the IPD that are higher than the comparator population median is assigned a value of 1. Conversely, all values that are lower are assigned a value of 0. Comparator population median is replaced by 0.5 to adjust to the categorization in the IPD data. The newly created IPD binary variable is matched so that the proportion is 0.5.

Calculating weights

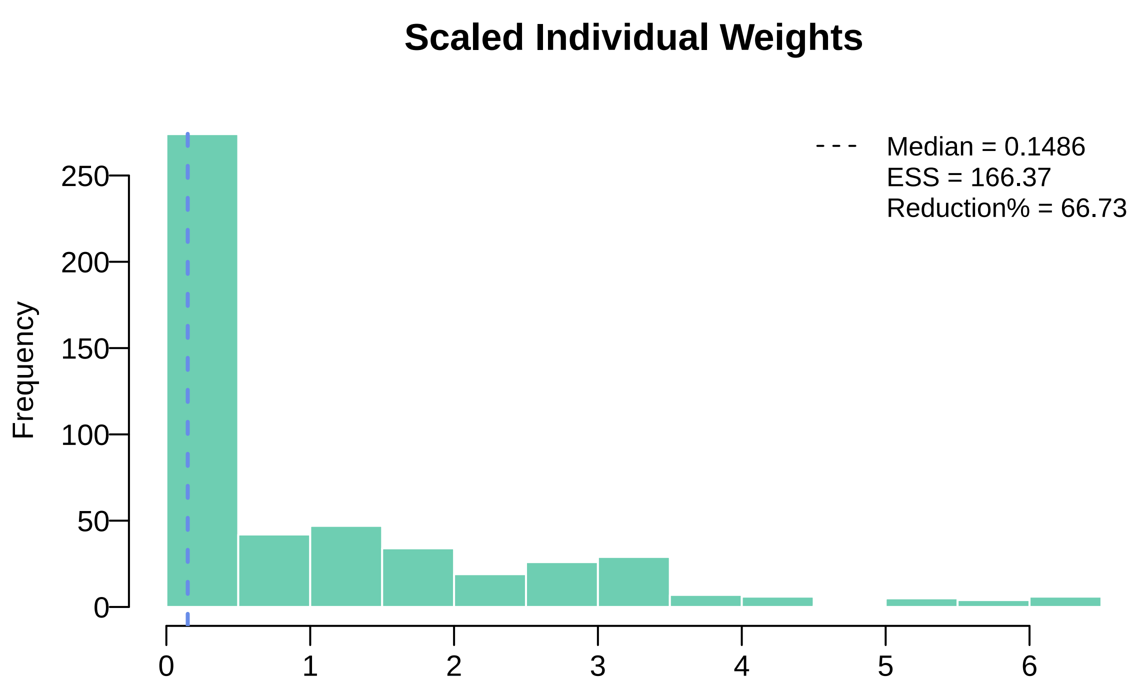
```
# list variables that are going to be used to match
centered_colnames <- c("AGE", "AGE_SQUARED", "SEX_MALE", "ECOGO", "SMOKE", "N_PR_THER_MEDIAN")
centered_colnames <- paste0(centered_colnames, "_CENTERED")

weighted_data <- estimate_weights(data = ipd_centered, centered_colnames = centered_colnames)
```

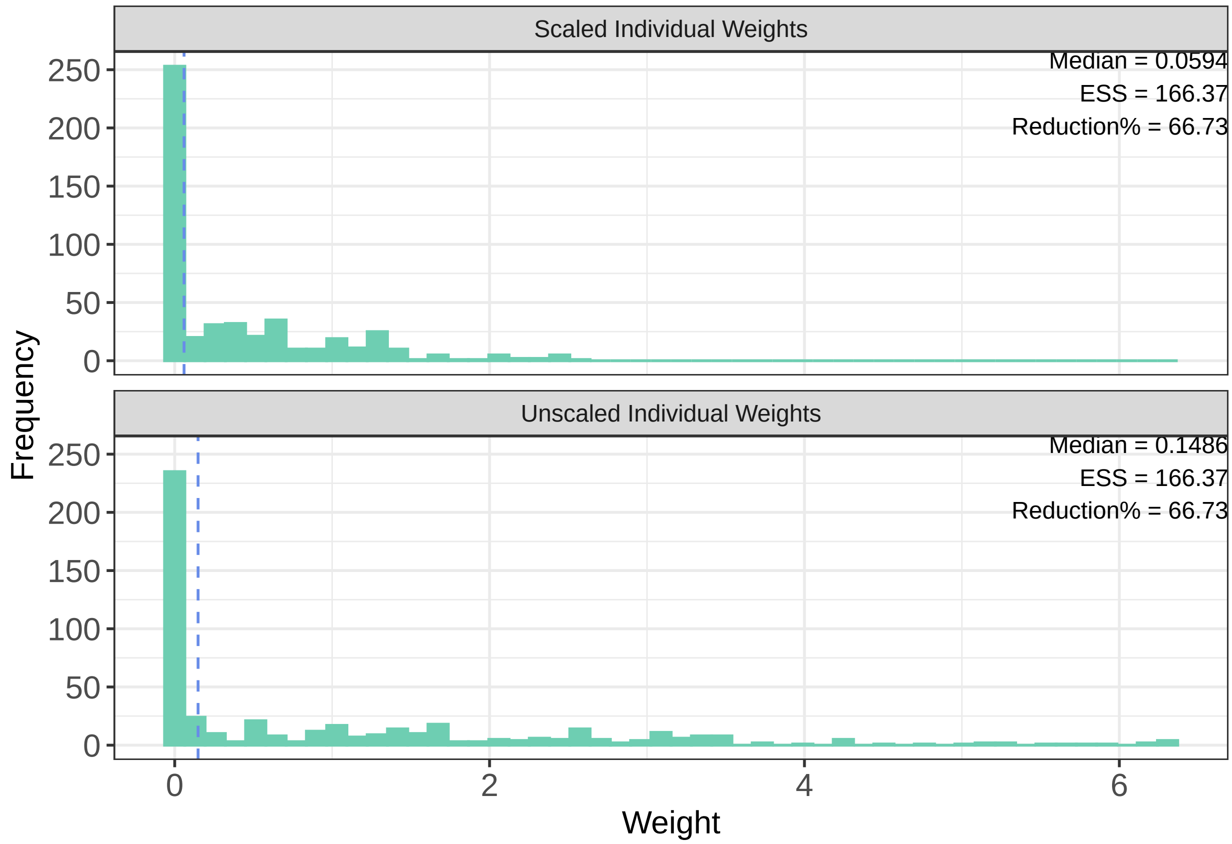
```
## initial  value 500.000000
## iter  10 value 215.753747
## iter  20 value 199.844445
## final  value 199.842237
## converged
```

```
# Alternatively, you can specify the numeric column locations for
# centered_colnames weighted_data <- estimate_weights(ipd_centered,
# centered_colnames = c(14, 16:20))

# Two options to plot weights plot using base R or ggplot
plot(weighted_data)
```




```
plot(weighted_data, ggplot = TRUE)
```



Another check after the weights are calculated is to look at how the weighted covariates match with the aggregate data summary.

```
outdata <- check_weights(weighted_data, agd)
outdata
```

```
##   covariate match_stat internal_trial internal_trial_after_weighted
## 1       AGE       Mean         59.850                         51.00
## 2       AGE         SD          9.011                          3.25
## 3  SEX_MALE       Prop          0.380                          0.49
## 4     ECOG0       Prop          0.410                          0.40
## 5     SMOKE       Prop          0.320                          0.20
## 6 N_PR_THER     Median          3.000                          2.00
##   external_trial sum_centered_IPD_with_weights
## 1          51.00                        0.0001
## 2           3.25                        0.0125
## 3           0.49                        0.0000
## 4           0.40                        0.0000
## 5           0.20                        0.0000
## 6           2.00                        0.0000
```

Time to event analysis

We first need to combine internal IPD data with pseudo comparator IPD. To obtain pseudo comparator IPD, we would digitize Kaplan Meier curves from the comparator study.

```r
pseudo_ipd <- read.csv(system.file("extdata", "psuedo_IPD.csv", package = "maicplus",
    mustWork = TRUE))
ipd_matched <- weighted_data$data

# Need to specify pseudo_ipd ARM if not specified
pseudo_ipd$ARM <- "B"  #Need to specify ARM for pseudo ipd

# Need to specify weights for pseudo_ipd which is 1
pseudo_ipd$weights <- 1

# make sure pseudo_ipd name has same name compared with ipd_matched for time,
# event, arm
colnames(pseudo_ipd) <- c("TIME", "EVENT", "ARM", "weights")

combined_data_tte <- rbind(ipd_matched[, colnames(pseudo_ipd)], pseudo_ipd)
combined_data_tte
```

```
##            TIME EVENT ARM      weights
## 1   281.5194863     0   A 6.884028e-01
## 2   500.0000000     0   A 5.669105e-08
## 3   304.6405555     0   A 1.146355e-01
## 4   102.4731386     0   A 1.178307e+00
## 5   101.6631927     0   A 1.605203e-06
## 6   237.0593233     1   A 1.178307e+00
## 7   337.3663345     0   A 7.020419e-01
## 8   180.6996302     1   A 1.076517e-02
## 9   156.3530945     0   A 4.281036e-01
## 10  126.0500878     0   A 1.794858e-01
## 11    2.9969434     0   A 1.887615e-03
## 12  189.3613424     0   A 1.315199e+00
## 13  137.1939184     1   A 4.172768e-01
## 14    0.4303082     1   A 2.829799e-03
## 15  238.4480442     0   A 2.447849e-01
## 16   64.6872731     0   A 9.795904e-05
## 17   62.6283981     0   A 1.831411e-11
## 18   61.8689337     1   A 8.304939e-01
## 19  172.9990508     0   A 3.727682e-01
## 20  500.0000000     0   A 1.320416e+00
## 21  404.9665031     1   A 4.664021e-01
## 22    8.6078129     1   A 1.222983e-03
## 23    2.6752029     1   A 4.966403e-01
## 24   37.1862175     1   A 1.146355e-01
## 25   72.3471050     1   A 1.409553e+00
## 26  149.0394051     0   A 1.452911e-02
## 27  271.3474961     0   A 1.375440e+00
## 28  132.7882079     0   A 5.493376e-01
## 29  124.8525835     1   A 6.432655e-05
## 30  115.1863723     0   A 2.321810e-01
## 31  391.6005114     0   A 2.344667e+00
## 32   53.9968261     0   A 2.180362e-02
## 33  125.7206051     0   A 4.878947e-05
## 34  500.0000000     0   A 6.854063e-01
## 35  313.1573766     1   A 3.446106e-02
```

```
## 36  198.6208194      1    A 6.991433e-11
## 37  168.8770094      1    A 9.458694e-10
## 38   97.4835043      1    A 2.690322e-10
## 39  120.7169260      0    A 1.314668e+00
## 40   87.4381573      1    A 6.724802e-02
## 41  117.0809444      1    A 1.431773e-02
## 42    7.4378994      1    A 4.590016e-01
## 43   53.9061795      0    A 5.227680e-10
## 44   42.4329399      0    A 6.346509e-06
## 45   22.8324762      1    A 1.228729e+00
## 46  152.6837825      0    A 3.311161e-08
## 47    5.1546502      1    A 3.325765e-07
## 48   13.8523925      0    A 2.527960e-03
## 49  317.5266710      0    A 2.085457e+00
## 50  222.4007682      0    A 1.403734e-10
## 51    7.2337488      1    A 6.755100e-01
## 52   60.2611923      0    A 1.854286e-06
## 53  176.9610656      1    A 5.079003e-01
## 54   12.6413833      1    A 1.074351e-11
## 55   63.5048707      0    A 8.304939e-01
## 56  329.3894821      0    A 6.906663e-01
## 57   54.6861744      0    A 5.844263e-09
## 58  187.3284384      0    A 2.069241e-01
## 59  108.2199961      0    A 1.024216e+00
## 60   16.0028888      0    A 2.000419e-01
## 61   59.6044136      0    A 1.797056e-09
## 62  176.6590842      1    A 1.633499e-02
## 63   49.6663522      0    A 3.284876e-11
## 64   87.2997415      1    A 4.649248e-07
## 65  326.3333469      1    A 9.256083e-01
## 66  173.9223130      1    A 2.403360e-10
## 67   59.3667629      0    A 3.706514e-08
## 68  500.0000000      0    A 6.791004e-01
## 69  101.2849073      0    A 6.374256e-06
## 70  500.0000000      0    A 2.334461e+00
## 71   12.7575511      1    A 3.311289e-07
## 72  170.4704706      0    A 4.020763e-01
## 73  379.0575377      0    A 9.296550e-01
## 74  295.6642687      0    A 2.321810e-01
## 75  180.0471671      1    A 1.375440e+00
## 76   88.3276166      0    A 3.723019e-06
## 77  230.9483853      1    A 5.016332e-02
## 78   19.5843537      0    A 1.993864e-01
## 79   67.6855902      0    A 2.177232e-01
## 80  187.2930109      1    A 1.472234e+00
## 81  205.1596244      0    A 3.601956e-03
## 82  133.9676829      0    A 8.268788e-01
## 83   99.3645367      0    A 4.185769e-06
## 84  152.3036359      1    A 2.229472e+00
## 85  158.5194710      1    A 4.994496e-02
## 86  203.4020672      1    A 5.913878e-01
## 87  336.8420868      1    A 7.925421e-07
## 88   25.6490953      0    A 1.950976e-07
## 89  352.4383084      0    A 9.919157e-09
```

```
## 90   33.8799882     0    A 6.725696e-01
## 91   15.8860250     0    A 1.105577e+00
## 92  245.3458498     1    A 9.375528e-07
## 93  425.2838139     0    A 9.795904e-05
## 94   28.5374060     0    A 1.797056e-09
## 95   94.0876845     1    A 4.690489e-10
## 96  109.4524597     0    A 5.706485e-03
## 97  313.3722027     0    A 5.517393e-01
## 98   50.3150662     1    A 1.294041e-05
## 99   52.4467296     1    A 3.841122e-01
## 100  10.2701408     0    A 6.619996e-01
## 101   6.8633253     0    A 3.711455e-01
## 102  11.1663130     1    A 1.213868e-01
## 103 208.5440529     0    A 3.727682e-01
## 104 453.8755271     0    A 8.460560e-06
## 105  13.7869996     0    A 8.460560e-06
## 106 350.2207221     1    A 7.792145e-11
## 107  89.6167692     1    A 2.885507e-09
## 108   0.2046028     0    A 9.458694e-10
## 109  30.8717178     1    A 2.186750e-01
## 110 280.3909651     0    A 4.153338e-07
## 111  84.7166775     0    A 9.364378e-01
## 112   8.3998372     0    A 1.409871e-10
## 113 135.1019149     0    A 3.739296e-06
## 114 126.7874323     0    A 1.699809e+00
## 115 174.5461325     1    A 1.427743e-06
## 116 161.5846889     1    A 7.104283e-06
## 117  66.0390115     0    A 2.065127e-01
## 118 104.5408470     0    A 6.409077e-04
## 119 209.6796639     1    A 5.644428e-08
## 120   1.2808132     0    A 9.375528e-07
## 121  45.4720260     0    A 3.178209e-04
## 122  43.6389358     0    A 5.394206e-04
## 123  13.1747627     0    A 1.052832e+00
## 124 204.8322250     1    A 2.403360e-10
## 125  76.2621398     1    A 2.520641e+00
## 126 179.5550296     0    A 7.104283e-06
## 127 153.0189691     0    A 4.994496e-02
## 128   2.0300958     0    A 6.235207e-02
## 129 168.2281259     1    A 1.649156e-08
## 130  55.0706664     0    A 1.605203e-06
## 131  31.4891944     0    A 4.033983e-01
## 132 168.4294215     0    A 2.495671e+00
## 133 213.0554394     1    A 3.807629e-04
## 134  58.1487885     1    A 3.821334e-01
## 135 139.0211403     0    A 1.223381e+00
## 136 104.1193509     0    A 6.991433e-11
## 137  46.1566080     0    A 1.846066e-08
## 138 158.4800477     0    A 3.401490e-04
## 139 295.1617767     1    A 1.524147e-05
## 140  42.5864914     0    A 1.052623e+00
## 141 127.9743928     0    A 1.950976e-07
## 142  34.0826279     0    A 5.818823e-09
## 143  93.3136053     0    A 9.476914e-01
```

```
## 144  39.6943191      0    A 1.854286e-06
## 145 418.5750209      1    A 3.807629e-04
## 146 108.9082910      0    A 9.674732e-08
## 147 105.0762188      1    A 2.065127e-01
## 148  56.6565761      1    A 4.185769e-06
## 149  80.5380440      1    A 7.745887e-01
## 150 142.4839370      0    A 1.074351e-11
## 151 189.7859562      1    A 6.820694e-01
## 152  44.5140716      0    A 5.016332e-02
## 153  30.8273271      1    A 2.002581e-01
## 154  10.2380585      1    A 4.690489e-10
## 155 214.0049455      0    A 3.722719e-08
## 156  12.6075825      0    A 9.458694e-10
## 157 185.7438835      0    A 1.216751e-01
## 158 210.5303223      0    A 6.949310e-02
## 159  35.8691523      1    A 2.157072e-11
## 160  28.2886213      0    A 3.401490e-04
## 161  42.1096005      1    A 8.859415e-01
## 162 132.0808389      0    A 2.103799e-03
## 163 158.7392013      0    A 1.409871e-10
## 164 208.5095081      1    A 6.409077e-04
## 165 124.8827926      1    A 1.223381e+00
## 166  66.0733700      0    A 4.051619e-01
## 167  18.0631542      0    A 4.033983e-01
## 168  35.7879285      0    A 1.048041e+00
## 169  35.6976157      1    A 3.824275e-04
## 170 344.6852715      0    A 1.001509e+00
## 171 130.7031601      0    A 9.919157e-09
## 172  75.4758943      1    A 9.597560e-12
## 173  79.9076509      0    A 1.942484e-07
## 174   3.8751813      0    A 4.944784e-01
## 175  56.6804001      0    A 2.241689e-01
## 176  88.0871919      0    A 5.559392e-01
## 177  40.1599769      0    A 1.329157e+00
## 178  11.3527364      1    A 6.342725e-03
## 179   8.7321192      0    A 2.971024e-07
## 180  81.6143745      1    A 1.216751e-01
## 181  48.3479226      0    A 1.314668e+00
## 182  62.6601429      0    A 1.699809e+00
## 183 187.8449785      0    A 2.392899e-10
## 184 114.1935322      1    A 1.222983e-03
## 185 500.0000000      0    A 2.024093e+00
## 186 151.8747627      1    A 9.458694e-10
## 187  43.0372554      0    A 2.736030e-01
## 188  58.9000652      0    A 3.325637e-08
## 189  44.1549084      1    A 1.084392e-01
## 190  39.9270162      1    A 6.066646e-01
## 191  10.3717562      1    A 1.433985e-06
## 192  25.5091635      1    A 6.961000e-11
## 193 500.0000000      0    A 6.235207e-02
## 194 108.9450103      0    A 1.707240e+00
## 195 101.7250316      0    A 1.334968e+00
## 196  56.4060157      0    A 4.590016e-01
## 197 129.6704036      1    A 1.279054e-02
```

```
## 198  47.0546355      0    A 1.302170e+00
## 199 169.4244480      1    A 2.527960e-03
## 200  20.9988640      0    A 3.078529e-02
## 201   6.5182421      0    A 2.065127e-01
## 202  66.0441764      0    A 1.431773e-02
## 203 329.8200403      0    A 4.020763e-01
## 204 130.9970045      1    A 5.874464e-02
## 205  32.8199139      0    A 6.007505e-02
## 206 111.9904588      1    A 2.598163e-05
## 207 259.3829166      0    A 6.457357e-01
## 208  48.0823337      0    A 3.538358e-06
## 209 373.8745493      0    A 4.944784e-01
## 210 500.0000000      0    A 5.417789e-04
## 211  25.5794699      1    A 2.430073e-02
## 212 362.2067038      1    A 1.003731e+00
## 213  49.2037680      1    A 2.447849e-01
## 214  12.3885075      0    A 1.079671e-01
## 215  71.2672138      0    A 1.228729e+00
## 216   3.3942755      1    A 1.237583e+00
## 217  34.0987472      0    A 2.929895e-02
## 218  91.9824456      1    A 1.452911e-02
## 219 320.7036093      1    A 3.325637e-08
## 220 178.5031223      0    A 7.712169e-01
## 221 116.0163469      0    A 1.100765e+00
## 222 134.5847123      0    A 6.345997e-08
## 223 246.1299544      0    A 1.409553e+00
## 224 132.8690671      0    A 3.401490e-04
## 225  25.3781343      0    A 9.971498e-01
## 226 180.2837518      0    A 3.603700e-01
## 227  12.8976681      1    A 1.105577e+00
## 228  48.7539111      1    A 3.723019e-06
## 229 254.5446271      1    A 2.157072e-11
## 230   8.3675931      1    A 1.223381e+00
## 231  30.4157094      1    A 1.186619e-10
## 232  32.3195134      0    A 5.746520e-05
## 233   5.8283381      1    A 1.403734e-10
## 234 409.5208072      0    A 5.706485e-03
## 235  80.7172811      0    A 9.971498e-01
## 236  29.5736758      0    A 5.706485e-03
## 237  88.3243550      1    A 8.642782e-08
## 238  16.6777043      1    A 1.452911e-02
## 239 201.8084701      1    A 9.919157e-09
## 240 310.0655345      0    A 6.991433e-11
## 241   4.0793628      0    A 4.118352e-01
## 242   1.2688996      1    A 8.670747e-01
## 243  33.3162513      0    A 2.233649e-04
## 244  70.2602449      0    A 2.678611e-10
## 245   5.4278376      0    A 2.430073e-02
## 246  50.1248807      0    A 3.841122e-01
## 247  85.1873869      0    A 3.416361e-04
## 248 464.9161960      1    A 2.898122e-09
## 249  26.3512119      0    A 1.838030e-08
## 250  23.2462453      1    A 3.416361e-04
## 251  54.6123914      0    A 2.311704e-01
```

```
## 252 204.7131962      0    A 1.024216e+00
## 253 221.0842591      1    A 4.966403e-01
## 254 126.8931381      0    A 1.649156e-08
## 255   6.8416124      1    A 1.584752e+00
## 256  39.5099966      0    A 6.854063e-01
## 257 303.3503769      0    A 2.251778e+00
## 258 103.3599337      0    A 7.591163e-06
## 259  73.0774123      0    A 4.918829e-09
## 260  15.2886098      0    A 4.500843e-01
## 261 144.5084460      1    A 9.555783e-12
## 262 227.9757161      0    A 3.416361e-04
## 263 233.3291179      0    A 3.294064e-02
## 264  10.9352310      0    A 2.166503e-11
## 265 284.9961410      0    A 1.459263e-02
## 266 119.6814391      0    A 6.066646e-01
## 267 153.7818843      0    A 1.356103e-01
## 268  41.5788383      0    A 2.069241e-01
## 269  17.1922456      0    A 7.080060e-07
## 270  12.1850070      0    A 2.484808e+00
## 271 301.7347889      0    A 3.841122e-01
## 272  93.7104093      0    A 1.052832e+00
## 273 414.9602508      1    A 1.228729e+00
## 274 203.2206009      0    A 6.961000e-11
## 275  81.0661957      0    A 4.643719e-01
## 276 135.8094446      1    A 5.888135e-01
## 277 113.1377720      0    A 3.401490e-04
## 278 406.0771660      0    A 8.605161e-08
## 279 136.1992022      1    A 3.189901e-05
## 280 110.6752035      0    A 2.112996e-03
## 281 282.6318510      0    A 1.431773e-02
## 282 245.5501089      0    A 3.807629e-04
## 283 101.3714715      0    A 1.605374e-09
## 284  14.6109023      1    A 3.062712e-01
## 285   3.2894058      0    A 2.009165e-01
## 286  20.4779808      0    A 4.167548e-06
## 287   0.3321626      1    A 6.991433e-11
## 288 178.9681939      0    A 1.524147e-05
## 289  27.2088851      0    A 1.101348e-04
## 290 211.4006422      0    A 1.052623e+00
## 291  30.7223985      1    A 6.820694e-01
## 292  60.0808303      0    A 1.626389e-02
## 293  99.6052279      0    A 9.416517e-07
## 294  86.9340820      0    A 1.362031e-01
## 295 191.1076682      1    A 2.241689e-01
## 296 311.8106687      1    A 1.369453e+00
## 297  55.6752080      1    A 7.591163e-06
## 298  20.9775017      1    A 3.217755e-03
## 299  89.1750066      0    A 1.315199e+00
## 300  39.8458329      0    A 4.185769e-06
## 301 204.6085827      0    A 1.902766e+00
## 302  34.5820833      0    A 3.586277e-03
## 303  37.8814246      0    A 2.074156e-01
## 304  88.6145506      0    A 1.950976e-07
## 305  98.5382896      1    A 3.603700e-01
```

```
## 306 237.8774307      1    A 6.906663e-01
## 307  76.6524844      1    A 1.279054e-02
## 308  91.8647559      0    A 9.919157e-09
## 309  51.6937890      0    A 2.484808e+00
## 310  29.5678669      0    A 9.962523e-09
## 311  71.2587985      1    A 6.409077e-04
## 312 145.2570428      1    A 1.058269e-01
## 313 249.9233069      0    A 1.935414e-11
## 314 131.4908055      1    A 1.926989e-11
## 315 312.7128096      1    A 2.392899e-10
## 316  22.6567004      1    A 1.478074e+00
## 317  96.7897325      0    A 5.517393e-01
## 318 103.3858997      1    A 5.888135e-01
## 319  34.5806822      1    A 6.725696e-01
## 320  56.7837124      0    A 1.699809e+00
## 321 311.0184998      0    A 8.633004e-01
## 322  63.9452407      0    A 5.097806e-03
## 323 173.7507167      0    A 6.547835e-01
## 324   4.6245131      0    A 3.619456e-01
## 325  52.2469200      1    A 1.048041e+00
## 326  93.2408592      0    A 1.329157e+00
## 327 101.6701105      0    A 6.176170e-01
## 328 247.7760192      0    A 8.375493e-07
## 329  43.3221132      0    A 1.048041e+00
## 330 130.9639163      0    A 5.843128e-01
## 331  89.7755695      1    A 5.016332e-02
## 332  37.7626359      1    A 1.571340e-10
## 333 236.1039880      0    A 4.020763e-01
## 334   5.5735189      0    A 5.394206e-04
## 335  15.9789179      0    A 1.074351e-11
## 336 219.5385758      0    A 5.120093e-03
## 337 122.1766793      0    A 5.669105e-08
## 338  85.2629108      1    A 5.843128e-01
## 339  72.1670941      1    A 9.971498e-01
## 340  49.3435304      1    A 8.728055e-03
## 341  84.3199651      0    A 8.375493e-07
## 342 102.0646836      0    A 5.818823e-09
## 343  99.9203054      0    A 1.273487e-02
## 344 105.9591180      0    A 1.409553e+00
## 345 341.3657643      0    A 2.311704e-01
## 346  61.1170188      1    A 1.192574e+00
## 347 273.1257002      0    A 1.228729e+00
## 348  83.7537322      0    A 1.302170e+00
## 349 287.8308610      0    A 7.228367e-01
## 350  85.1705400      0    A 9.962523e-09
## 351 500.0000000      0    A 6.374256e-06
## 352  60.6304562      0    A 1.228729e+00
## 353 405.1024439      0    A 4.038341e-01
## 354  10.3618326      0    A 1.228729e+00
## 355  13.8586577      0    A 3.807629e-04
## 356 222.7575312      0    A 6.143110e-01
## 357  12.9398765      0    A 2.231931e-01
## 358  71.8214219      0    A 1.840480e-01
## 359  28.0451023      1    A 9.435662e-01
```

```
## 360 217.0832170      1    A 1.995398e-04
## 361  30.4968240      1    A 4.172768e-01
## 362  37.0950110      0    A 6.176170e-01
## 363  15.8151577      0    A 4.154604e-01
## 364   9.1691822      0    A 6.755100e-01
## 365 128.8321170      1    A 9.375528e-07
## 366  68.7223284      0    A 1.713589e-05
## 367  23.2247571      0    A 9.364378e-01
## 368 500.0000000      0    A 1.121090e+00
## 369  12.0051551      1    A 6.034572e-01
## 370  70.5995378      0    A 5.669105e-08
## 371 128.1287128      0    A 7.104283e-06
## 372 173.8815563      0    A 1.375440e+00
## 373 147.5909236      1    A 1.307863e+00
## 374 106.7080111      0    A 1.112491e-04
## 375  72.1695415      1    A 5.517393e-01
## 376 100.0319960      0    A 1.707240e+00
## 377  21.5799492      1    A 2.243415e-04
## 378 129.3533244      0    A 3.244159e-09
## 379   1.9459751      1    A 2.430073e-02
## 380  50.8566442      1    A 9.417521e-10
## 381 169.4422522      1    A 2.829799e-03
## 382 141.6016337      0    A 1.024216e+00
## 383 500.0000000      0    A 2.495671e+00
## 384  67.3416893      1    A 2.784573e-02
## 385 293.6815759      1    A 4.051619e-01
## 386 170.5776237      1    A 9.375528e-07
## 387  74.4971040      0    A 6.724802e-02
## 388 252.7425757      0    A 6.754202e-02
## 389 370.3157123      1    A 1.116210e+00
## 390  18.4767328      0    A 8.268788e-01
## 391 242.3122311      0    A 1.184626e-01
## 392 108.6286686      0    A 8.304939e-01
## 393 135.1588000      1    A 1.459263e-02
## 394  13.4000154      1    A 9.458694e-10
## 395  50.6666237      1    A 4.994496e-02
## 396 181.3283312      0    A 3.299237e-11
## 397  13.0073322      0    A 2.004122e-04
## 398 357.7553493      0    A 6.346509e-06
## 399 180.9009377      1    A 5.517393e-01
## 400 242.1178978      0    A 2.992495e-01
## 401  60.2893293      0    A 4.500843e-01
## 402  93.4905863      0    A 1.649220e-07
## 403  52.7997845      1    A 4.051619e-01
## 404 350.2654152      0    A 5.120093e-03
## 405 241.1774392      0    A 2.094575e+00
## 406   8.7683064      0    A 1.192574e+00
## 407 330.7110893      0    A 1.713589e-05
## 408  48.4040249      0    A 7.300688e-01
## 409  77.9263829      0    A 2.066653e-06
## 410  16.1304526      0    A 7.745887e-01
## 411   9.3282041      0    A 3.416361e-04
## 412 164.7372721      1    A 6.149286e-01
## 413 366.6874542      1    A 5.559392e-01
```

```
## 414  38.5668907      0    A 3.311161e-08
## 415  21.8873529      0    A 1.786562e+00
## 416  20.9492488      0    A 8.670747e-01
## 417 174.2766212      1    A 1.636065e-11
## 418  57.1719156      0    A 9.971498e-01
## 419  41.6267738      1    A 6.550478e-01
## 420  15.8302898      1    A 3.164374e-04
## 421  86.8743175      1    A 1.995398e-04
## 422 350.3644114      0    A 2.243415e-04
## 423 241.5604544      1    A 1.452911e-02
## 424 318.7833017      1    A 8.642782e-08
## 425 359.1658176      0    A 1.302170e+00
## 426  14.1516358      1    A 4.500843e-01
## 427 104.8671198      0    A 9.555783e-12
## 428 158.0674152      0    A 5.888135e-01
## 429  53.4114635      1    A 2.971024e-07
## 430  52.6613128      1    A 4.669574e-07
## 431 402.2578995      0    A 6.381179e-04
## 432   2.4121713      0    A 1.846215e-06
## 433 189.4981770      1    A 1.100765e+00
## 434   4.2866692      0    A 6.949310e-02
## 435 256.4404196      0    A 2.177232e-01
## 436 452.7208595      1    A 1.409553e+00
## 437  31.7437313      0    A 3.807629e-04
## 438 370.0496553      0    A 3.807629e-04
## 439 299.0591141      0    A 3.401490e-04
## 440 134.9211996      0    A 1.024216e+00
## 441  34.0150132      1    A 6.409077e-04
## 442  36.3408479      1    A 2.447849e-01
## 443 133.8500312      0    A 1.713589e-05
## 444 332.3420058      0    A 6.342725e-03
## 445  23.7400288      0    A 6.169967e-01
## 446 343.7001138      1    A 2.032942e+00
## 447 222.9677203      0    A 7.826211e-11
## 448  76.9672003      1    A 3.693166e-11
## 449 389.5516777      1    A 2.321810e-01
## 450 174.7162274      0    A 2.447849e-01
## 451 363.8263817      1    A 5.493376e-01
## 452  19.6747055      1    A 4.944784e-01
## 453  19.9823259      0    A 1.636065e-11
## 454  10.2292442      0    A 2.243415e-04
## 455 326.7511055      0    A 2.004122e-04
## 456  20.0770245      1    A 2.074156e-01
## 457  36.0369583      1    A 3.325637e-08
## 458 184.0466008      1    A 5.669105e-08
## 459  76.6013852      1    A 5.270811e-02
## 460 207.4070734      0    A 5.101208e-01
## 461 154.3107802      0    A 1.307863e+00
## 462 106.6420898      1    A 4.051619e-01
## 463 205.0363326      1    A 5.746520e-05
## 464  20.9126041      1    A 3.078529e-02
## 465  18.6336849      0    A 5.197152e-01
## 466 136.3800132      1    A 3.739296e-06
## 467 316.3884586      0    A 2.186750e-01
```

```
## 468 141.3017821      0    A 3.217755e-03
## 469  13.2112969      0    A 9.770186e-03
## 470  99.6448581      0    A 1.742876e-07
## 471 135.2069709      1    A 6.149286e-01
## 472  61.6891086      0    A 3.059637e-01
## 473 101.1286638      0    A 1.100765e+00
## 474 166.2426128      1    A 1.307863e+00
## 475  79.8316941      0    A 3.349799e-01
## 476 234.7546470      0    A 2.032942e+00
## 477  94.7052270      0    A 6.345997e-08
## 478 168.1161018      1    A 6.547835e-01
## 479 146.5126322      0    A 1.048249e+00
## 480 500.0000000      0    A 6.034572e-01
## 481 130.1803275      0    A 1.735290e-07
## 482  39.2364700      1    A 4.185769e-06
## 483  67.4707314      1    A 2.509669e+00
## 484  89.4307290      1    A 4.020763e-01
## 485 101.4314849      0    A 6.346509e-06
## 486 245.8857517      0    A 6.876598e-01
## 487 117.1680451      1    A 1.189805e-01
## 488 143.3116393      1    A 1.232196e+00
## 489 382.6979071      1    A 6.342725e-03
## 490  33.8249243      0    A 3.294064e-02
## 491 157.0907872      0    A 1.846215e-06
## 492  15.7821636      1    A 1.448549e-05
## 493  87.0820950      1    A 3.804700e-01
## 494 190.6912802      1    A 3.706514e-08
## 495  49.8173522      0    A 1.052832e+00
## 496 215.4027340      1    A 5.844263e-09
## 497 111.1416159      1    A 5.219874e-01
## 498  17.0752816      1    A 1.846215e-06
## 499  98.2021703      1    A 4.020763e-01
## 500  97.5095606      1    A 6.754202e-02
## 501  20.2311676      1    B 1.000000e+00
## 502  28.7679537      1    B 1.000000e+00
## 503  41.0662129      0    B 1.000000e+00
## 504   0.8492261      1    B 1.000000e+00
## 505   9.0521882      0    B 1.000000e+00
## 506   3.4450075      1    B 1.000000e+00
## 507  88.7049039      0    B 1.000000e+00
## 508  75.5205919      0    B 1.000000e+00
## 509 102.0480708      0    B 1.000000e+00
## 510  61.1832674      0    B 1.000000e+00
## 511  44.6248777      1    B 1.000000e+00
## 512 417.7060892      0    B 1.000000e+00
## 513 188.0853029      1    B 1.000000e+00
## 514 135.4086771      1    B 1.000000e+00
## 515  14.4693075      1    B 1.000000e+00
## 516  19.2886739      1    B 1.000000e+00
## 517 253.9413828      0    B 1.000000e+00
## 518  18.8265045      1    B 1.000000e+00
## 519  42.5311133      0    B 1.000000e+00
## 520 203.2845550      0    B 1.000000e+00
## 521  32.2845542      1    B 1.000000e+00
```

```
## 522   3.0685160      1    B 1.000000e+00
## 523  53.3559798      1    B 1.000000e+00
## 524  69.5240787      0    B 1.000000e+00
## 525  38.8236166      1    B 1.000000e+00
## 526 204.6022754      0    B 1.000000e+00
## 527 184.5103590      0    B 1.000000e+00
## 528  83.5830935      0    B 1.000000e+00
## 529 114.7067767      1    B 1.000000e+00
## 530  31.4008382      1    B 1.000000e+00
## 531  65.7530859      0    B 1.000000e+00
## 532 163.6438166      1    B 1.000000e+00
## 533 103.4746398      0    B 1.000000e+00
## 534 115.6529216      0    B 1.000000e+00
## 535 119.8663712      1    B 1.000000e+00
## 536  39.7824824      0    B 1.000000e+00
## 537  26.2260987      1    B 1.000000e+00
## 538 168.2133266      1    B 1.000000e+00
## 539  77.4276997      0    B 1.000000e+00
## 540 113.1300571      0    B 1.000000e+00
## 541 341.8226476      0    B 1.000000e+00
## 542   4.6203917      1    B 1.000000e+00
## 543  76.1546928      1    B 1.000000e+00
## 544  31.3239259      1    B 1.000000e+00
## 545 139.8137748      1    B 1.000000e+00
## 546  26.2748075      1    B 1.000000e+00
## 547  78.8443008      0    B 1.000000e+00
## 548 150.1996343      0    B 1.000000e+00
## 549   7.5020884      1    B 1.000000e+00
## 550  27.0121949      1    B 1.000000e+00
## 551 157.8603589      1    B 1.000000e+00
## 552 196.4246807      0    B 1.000000e+00
## 553  21.6935437      0    B 1.000000e+00
## 554 140.2541753      1    B 1.000000e+00
## 555  12.1075238      0    B 1.000000e+00
## 556 107.3694875      1    B 1.000000e+00
## 557  37.8779261      1    B 1.000000e+00
## 558 121.0960158      0    B 1.000000e+00
## 559  20.3595135      1    B 1.000000e+00
## 560  41.0376828      0    B 1.000000e+00
## 561  27.4099098      1    B 1.000000e+00
## 562  14.2204744      1    B 1.000000e+00
## 563  49.0816396      1    B 1.000000e+00
## 564  91.8865159      1    B 1.000000e+00
## 565  20.8240812      0    B 1.000000e+00
## 566  19.7480647      1    B 1.000000e+00
## 567  13.4869944      0    B 1.000000e+00
## 568 297.0789867      1    B 1.000000e+00
## 569  35.3155408      1    B 1.000000e+00
## 570  61.3301293      1    B 1.000000e+00
## 571 236.3826316      0    B 1.000000e+00
## 572  37.6872463      1    B 1.000000e+00
## 573  12.4915818      0    B 1.000000e+00
## 574 117.1516860      1    B 1.000000e+00
## 575  74.0347874      1    B 1.000000e+00
```

```
## 576  31.8360820      1    B 1.000000e+00
## 577   6.1533491      0    B 1.000000e+00
## 578  58.6084245      1    B 1.000000e+00
## 579   3.7608198      0    B 1.000000e+00
## 580  32.1776180      1    B 1.000000e+00
## 581 123.7513692      1    B 1.000000e+00
## 582 191.6321564      1    B 1.000000e+00
## 583  49.7949898      0    B 1.000000e+00
## 584  95.9720009      0    B 1.000000e+00
## 585  35.3873079      0    B 1.000000e+00
## 586  93.7520019      1    B 1.000000e+00
## 587   8.2221920      1    B 1.000000e+00
## 588  69.4935935      1    B 1.000000e+00
## 589 115.3801046      0    B 1.000000e+00
## 590 201.1106058      0    B 1.000000e+00
## 591 101.8512269      1    B 1.000000e+00
## 592  16.3430735      0    B 1.000000e+00
## 593  73.2620925      1    B 1.000000e+00
## 594  20.9476736      0    B 1.000000e+00
## 595  13.2074189      1    B 1.000000e+00
## 596 133.1594500      0    B 1.000000e+00
## 597   6.7140516      1    B 1.000000e+00
## 598   2.0758651      0    B 1.000000e+00
## 599 155.7676131      1    B 1.000000e+00
## 600 324.6217046      0    B 1.000000e+00
## 601  57.1030595      1    B 1.000000e+00
## 602   3.6143976      1    B 1.000000e+00
## 603  35.3912534      0    B 1.000000e+00
## 604  48.2754867      1    B 1.000000e+00
## 605  24.2416986      0    B 1.000000e+00
## 606   3.2412819      1    B 1.000000e+00
## 607  77.1943697      0    B 1.000000e+00
## 608 401.8506175      1    B 1.000000e+00
## 609  52.2920989      0    B 1.000000e+00
## 610  14.2519825      0    B 1.000000e+00
## 611 195.5002523      1    B 1.000000e+00
## 612   3.1712902      1    B 1.000000e+00
## 613  58.4270646      1    B 1.000000e+00
## 614  18.5735263      1    B 1.000000e+00
## 615  59.6546380      1    B 1.000000e+00
## 616   6.4806603      1    B 1.000000e+00
## 617  75.1656036      1    B 1.000000e+00
## 618   1.7198339      1    B 1.000000e+00
## 619  84.2445319      0    B 1.000000e+00
## 620  18.3219669      0    B 1.000000e+00
## 621  27.3660973      0    B 1.000000e+00
## 622  23.8855711      0    B 1.000000e+00
## 623  68.8229804      1    B 1.000000e+00
## 624  18.9849356      1    B 1.000000e+00
## 625  37.3202892      1    B 1.000000e+00
## 626  48.0896160      1    B 1.000000e+00
## 627  78.4347848      1    B 1.000000e+00
## 628 128.6077718      0    B 1.000000e+00
## 629  91.3637782      1    B 1.000000e+00
```

```
## 630  56.9334980      1    B 1.000000e+00
## 631  13.9279249      1    B 1.000000e+00
## 632   3.0113133      1    B 1.000000e+00
## 633  53.6840931      0    B 1.000000e+00
## 634 109.4392228      1    B 1.000000e+00
## 635  27.8727602      0    B 1.000000e+00
## 636  31.3737925      1    B 1.000000e+00
## 637  36.6028967      0    B 1.000000e+00
## 638  34.0285658      1    B 1.000000e+00
## 639  49.9062966      1    B 1.000000e+00
## 640  47.1405277      0    B 1.000000e+00
## 641 101.0785894      1    B 1.000000e+00
## 642  93.6487147      1    B 1.000000e+00
## 643  20.8436242      1    B 1.000000e+00
## 644   3.4515079      1    B 1.000000e+00
## 645  29.8875161      1    B 1.000000e+00
## 646  63.3996256      0    B 1.000000e+00
## 647 230.9729773      1    B 1.000000e+00
## 648 364.3259341      1    B 1.000000e+00
## 649  12.2323488      1    B 1.000000e+00
## 650 159.8315468      0    B 1.000000e+00
## 651 258.5498424      1    B 1.000000e+00
## 652  24.9889490      0    B 1.000000e+00
## 653  63.0700888      1    B 1.000000e+00
## 654 320.4212314      1    B 1.000000e+00
## 655  90.6121802      1    B 1.000000e+00
## 656 160.7520774      0    B 1.000000e+00
## 657  28.8516511      0    B 1.000000e+00
## 658   8.7919587      0    B 1.000000e+00
## 659  27.1385849      1    B 1.000000e+00
## 660  38.2870764      1    B 1.000000e+00
## 661   7.4210626      0    B 1.000000e+00
## 662  13.6943879      1    B 1.000000e+00
## 663 139.8042160      1    B 1.000000e+00
## 664 170.6566002      0    B 1.000000e+00
## 665 114.2058151      1    B 1.000000e+00
## 666 172.4208727      0    B 1.000000e+00
## 667  39.0678457      1    B 1.000000e+00
## 668  16.0775600      0    B 1.000000e+00
## 669  62.4040822      1    B 1.000000e+00
## 670  95.9224128      1    B 1.000000e+00
## 671  31.0753223      1    B 1.000000e+00
## 672   7.3859528      1    B 1.000000e+00
## 673  13.5637730      0    B 1.000000e+00
## 674  25.3284722      0    B 1.000000e+00
## 675 215.0981459      1    B 1.000000e+00
## 676  13.9126408      1    B 1.000000e+00
## 677 220.6392416      1    B 1.000000e+00
## 678  82.5469181      0    B 1.000000e+00
## 679  10.3249249      0    B 1.000000e+00
## 680 114.4152727      0    B 1.000000e+00
## 681  67.4200893      1    B 1.000000e+00
## 682  16.5377021      1    B 1.000000e+00
## 683  21.4476172      0    B 1.000000e+00
```

```
## 684 163.3827108      1    B 1.000000e+00
## 685 114.8396735      0    B 1.000000e+00
## 686  33.7408843      1    B 1.000000e+00
## 687  72.9266499      1    B 1.000000e+00
## 688  85.2637815      1    B 1.000000e+00
## 689  12.7049720      0    B 1.000000e+00
## 690  33.5083467      1    B 1.000000e+00
## 691  64.9965467      1    B 1.000000e+00
## 692  62.4816013      1    B 1.000000e+00
## 693  32.6268562      1    B 1.000000e+00
## 694 230.5868701      1    B 1.000000e+00
## 695  40.2919970      0    B 1.000000e+00
## 696  38.4803842      1    B 1.000000e+00
## 697 116.5770227      1    B 1.000000e+00
## 698  11.9051673      1    B 1.000000e+00
## 699  34.7955872      1    B 1.000000e+00
## 700  48.1972482      1    B 1.000000e+00
## 701  84.6593786      0    B 1.000000e+00
## 702 210.0548161      0    B 1.000000e+00
## 703  32.3201275      0    B 1.000000e+00
## 704  47.8891543      1    B 1.000000e+00
## 705  94.4459064      1    B 1.000000e+00
## 706  27.1427844      1    B 1.000000e+00
## 707  75.4571413      0    B 1.000000e+00
## 708  13.0554772      1    B 1.000000e+00
## 709   1.4030134      1    B 1.000000e+00
## 710 150.6582394      1    B 1.000000e+00
## 711  80.4921978      0    B 1.000000e+00
## 712 102.1683344      1    B 1.000000e+00
## 713  44.9419892      0    B 1.000000e+00
## 714 232.5020784      0    B 1.000000e+00
## 715   2.7979480      0    B 1.000000e+00
## 716  64.5608166      0    B 1.000000e+00
## 717  67.5251343      1    B 1.000000e+00
## 718 129.1025209      1    B 1.000000e+00
## 719 500.0000000      0    B 1.000000e+00
## 720  19.5837178      0    B 1.000000e+00
## 721  70.1227875      0    B 1.000000e+00
## 722  22.5998941      1    B 1.000000e+00
## 723  83.5853505      1    B 1.000000e+00
## 724   8.6680839      1    B 1.000000e+00
## 725 129.7187580      1    B 1.000000e+00
## 726  19.3144386      0    B 1.000000e+00
## 727 333.8879947      1    B 1.000000e+00
## 728  51.2082623      1    B 1.000000e+00
## 729  46.6101451      0    B 1.000000e+00
## 730  73.6926698      1    B 1.000000e+00
## 731  34.3307012      0    B 1.000000e+00
## 732  76.4192775      1    B 1.000000e+00
## 733   2.9805455      0    B 1.000000e+00
## 734 127.4336784      0    B 1.000000e+00
## 735   3.4211613      0    B 1.000000e+00
## 736  54.9236065      1    B 1.000000e+00
## 737 294.1386975      1    B 1.000000e+00
```

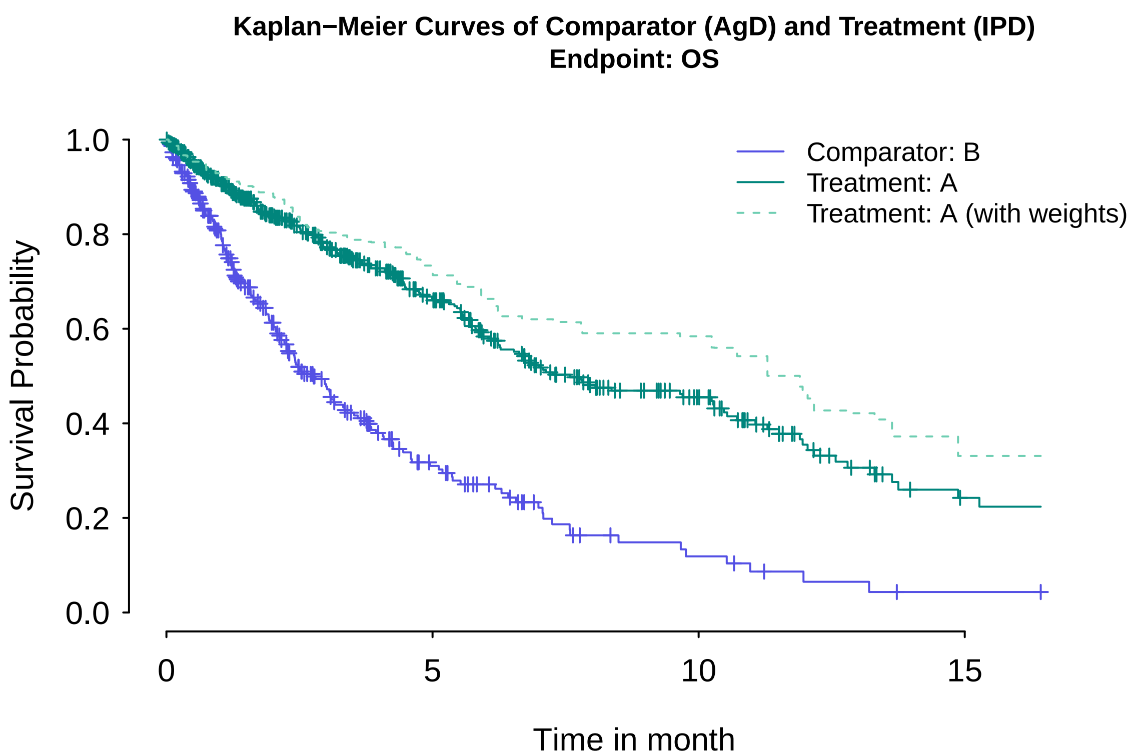
```
## 738  15.2915139      0   B 1.000000e+00
## 739  56.8481500      1   B 1.000000e+00
## 740  55.3458512      0   B 1.000000e+00
## 741   7.7622930      1   B 1.000000e+00
## 742  93.8619589      0   B 1.000000e+00
## 743 177.5091814      0   B 1.000000e+00
## 744  70.2628625      0   B 1.000000e+00
## 745  41.2949253      1   B 1.000000e+00
## 746 129.7250459      1   B 1.000000e+00
## 747   5.5211413      0   B 1.000000e+00
## 748  38.4697339      1   B 1.000000e+00
## 749  38.4278985      0   B 1.000000e+00
## 750  37.9046463      1   B 1.000000e+00
## 751  29.7580207      0   B 1.000000e+00
## 752  37.3557166      0   B 1.000000e+00
## 753   2.3393342      1   B 1.000000e+00
## 754 116.2388631      0   B 1.000000e+00
## 755  37.1140694      1   B 1.000000e+00
## 756  93.2119835      1   B 1.000000e+00
## 757  16.7345687      0   B 1.000000e+00
## 758  28.7863941      0   B 1.000000e+00
## 759  17.0540489      1   B 1.000000e+00
## 760  60.2527239      0   B 1.000000e+00
## 761  56.7171853      0   B 1.000000e+00
## 762  22.2428646      1   B 1.000000e+00
## 763  75.5904943      0   B 1.000000e+00
## 764  32.8579123      1   B 1.000000e+00
## 765 105.5323827      0   B 1.000000e+00
## 766 124.2150196      1   B 1.000000e+00
## 767  90.5420118      1   B 1.000000e+00
## 768  55.0154610      1   B 1.000000e+00
## 769 215.5597419      1   B 1.000000e+00
## 770   5.5773040      1   B 1.000000e+00
## 771 144.2521772      0   B 1.000000e+00
## 772  69.0486551      1   B 1.000000e+00
## 773 212.7464896      1   B 1.000000e+00
## 774 128.9828791      0   B 1.000000e+00
## 775  26.6127303      1   B 1.000000e+00
## 776  12.2191633      1   B 1.000000e+00
## 777  18.6707015      0   B 1.000000e+00
## 778 175.4929594      0   B 1.000000e+00
## 779  40.8238759      1   B 1.000000e+00
## 780  69.7451992      1   B 1.000000e+00
## 781  47.8510476      0   B 1.000000e+00
## 782  18.9679808      0   B 1.000000e+00
## 783  61.2068279      1   B 1.000000e+00
## 784  44.8460398      1   B 1.000000e+00
## 785 199.8426419      1   B 1.000000e+00
## 786  58.7730149      1   B 1.000000e+00
## 787  64.6149696      1   B 1.000000e+00
## 788  38.9656553      0   B 1.000000e+00
## 789  17.7346078      1   B 1.000000e+00
## 790  64.0457781      1   B 1.000000e+00
## 791  40.1446851      1   B 1.000000e+00
```

```
## 792  73.5593273      1    B 1.000000e+00
## 793  68.5812580      0    B 1.000000e+00
## 794  96.2560394      1    B 1.000000e+00
## 795  10.5896188      1    B 1.000000e+00
## 796 111.0420553      0    B 1.000000e+00
## 797  22.4233793      1    B 1.000000e+00
## 798   9.0396962      1    B 1.000000e+00
## 799 143.5751793      0    B 1.000000e+00
## 800  15.1259399      0    B 1.000000e+00
```

Report 1: Kaplan-Meier plot

```r
kmobj <- survfit(Surv(TIME, EVENT) ~ ARM, combined_data_tte, conf.type = "log-log")
kmobj_adj <- survfit(Surv(TIME, EVENT) ~ ARM, combined_data_tte, weights = combined_data_tte$weights,
    conf.type = "log-log")

par(cex.main = 0.85)
km_plot(kmobj, kmobj_adj, time_scale = "month", trt = "A", trt_ext = "B", endpoint_name = "OS")
```



There is also a ggplot option for Kaplan-Meier curves using `survminer` R package.

```r
# km_plot2(combined_data_tte, trt = 'A', trt_ext = 'B', censor = TRUE,
# risk.table = TRUE)
```

Report 2: Analysis table (Cox model) before and after matching, incl Median Survival Time

We can then fit a cox regression model using the combined dataset. For the weight adjusted cox regression, we fit the model with robust standard errors. Along with the hazard ratios, we can also find median survival time using `medSurv_makeup` function. Then, `report_table` function nicely combines the information together and create a result table.

```
# Fit a Cox model with/without weights to estimate the HR
unweighted_cox <- coxph(Surv(TIME, EVENT == 1) ~ ARM, data = combined_data_tte)
weighted_cox <- coxph(Surv(TIME, EVENT == 1) ~ ARM, data = combined_data_tte, weights = combined_data_t
    robust = TRUE)

# Derive median survival time
medSurv <- medSurv_makeup(kmobj, legend = "before matching", time_scale = "day")
medSurv_adj <- medSurv_makeup(kmobj_adj, legend = "after matching", time_scale = "day")
medSurv_out <- rbind(medSurv, medSurv_adj)
medSurv_out
```

```
##   treatment            type records    n.max  n.start    events    rmean
## 1     ARM=A before matching     500 500.0000 500.0000 190.00000 265.1012
## 2     ARM=B before matching     300 300.0000 300.0000 178.00000 130.9893
## 3     ARM=A  after matching     500 199.8422 199.8422  66.84953 307.7223
## 4     ARM=B  after matching     300 300.0000 300.0000 178.00000 130.9893
##   se(rmean)    median   0.95LCL  0.95UCL
## 1  10.80981 230.94839 191.10767 313.1574
## 2  10.24910  83.58535  68.82298 101.0786
## 3  16.71338 362.20670 237.05932 452.7209
## 4  10.24910  83.58535  68.82298 101.0786
```

```
rbind(report_table(unweighted_cox, medSurv, tag = paste0("Before/", "Overall survival")),
    report_table(weighted_cox, medSurv_adj, tag = paste0("After/", "Overall survival")))
```

```
##                   Matching treatment     N n.events(%)     median[95% CI]
## 2  Before/Overall survival     ARM=B 300.0   178(59.3)  83.6[ 68.8;101.1]
## 1  Before/Overall survival     ARM=A 500.0   190(38.0) 230.9[191.1;313.2]
## 21  After/Overall survival     ARM=B 300.0   178(59.3)  83.6[ 68.8;101.1]
## 11  After/Overall survival     ARM=A 199.8  66.8(33.5) 362.2[237.1;452.7]
##          HR[95% CI] WaldTest
## 2  2.67[2.16;3.29]   <0.001
## 1
## 21 3.46[2.53;4.74]   <0.001
## 11
```

Report 3: Bootstrap result

```
set.seed(1)
HR_bootstraps <- boot(data = ipd_centered, statistic = bootstrap_HR, centered_colnames = centered_colna
    pseudo_ipd = pseudo_ipd, model = Surv(TIME, EVENT == 1) ~ ARM, ref_treat = "B",
    R = 1000)
```

```r
# Median of the bootstrap samples
HR_median <- median(HR_bootstraps$t)

# Bootstrap CI - Percentile CI
boot_ci_HR <- boot.ci(boot.out = HR_bootstraps, index = 1, type = "perc")

# Bootstrap CI - BCa CI
boot_ci_HR_BCA <- boot.ci(boot.out = HR_bootstraps, index = 1, type = "bca")

HR_median
```

```
## [1] 0.2858165
```

```r
boot_ci_HR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = HR_bootstraps, type = "perc", index = 1)
##
## Intervals :
## Level     Percentile
## 95%   ( 0.2236,  0.3689 )
## Calculations and Intervals on Original Scale
```

```r
boot_ci_HR_BCA
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = HR_bootstraps, type = "bca", index = 1)
##
## Intervals :
## Level       BCa
## 95%   ( 0.2296,  0.3789 )
## Calculations and Intervals on Original Scale
```

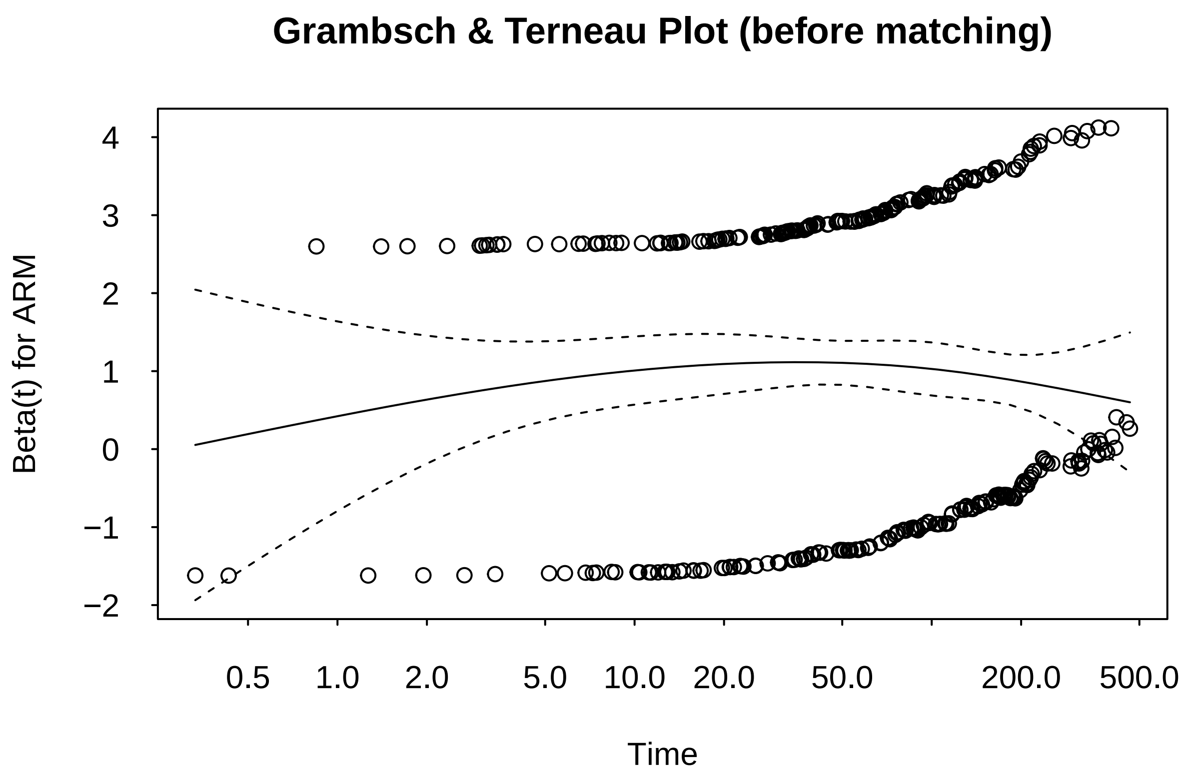
Report 4: Diagnosis Plot

```r
# grambsch & theaneu ph test
coxdiag <- cox.zph(unweighted_cox, global = F, transform = "log")
coxdiag_adj <- cox.zph(weighted_cox, global = F, transform = "log")

coxdiag
```

```
##       chisq df    p
## ARM 0.00996  1 0.92
```

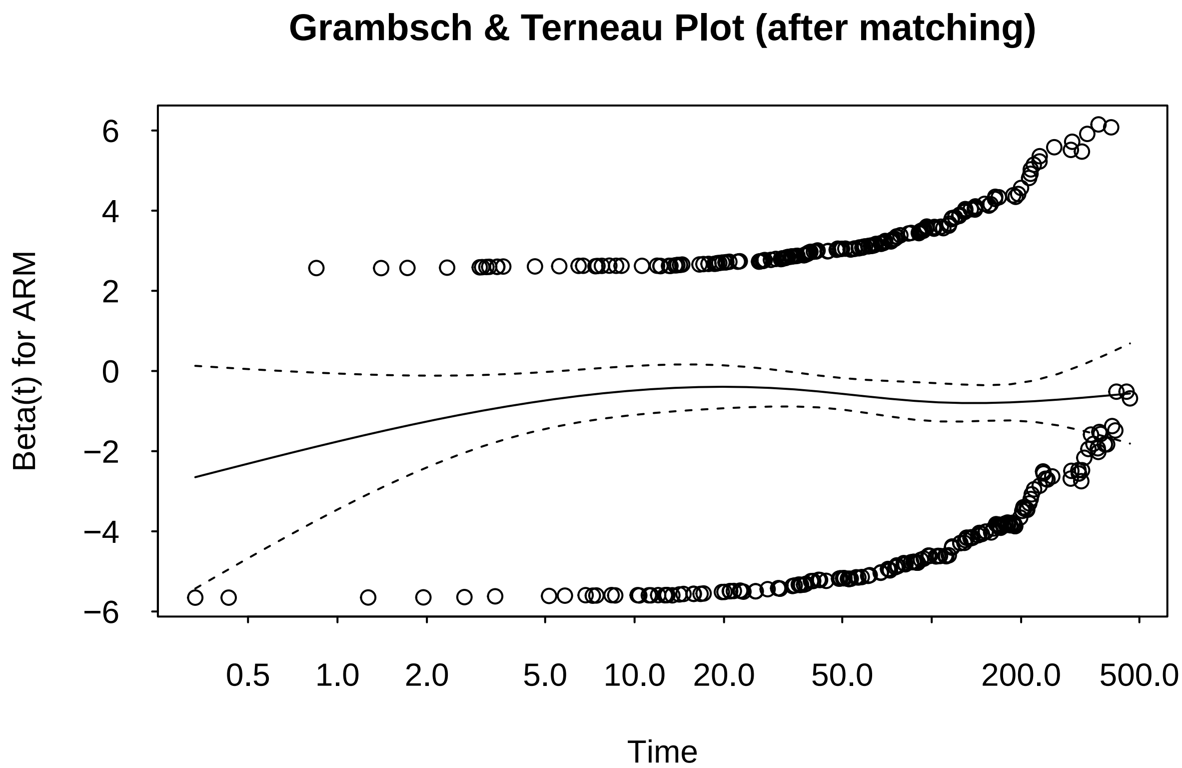
```r
par(mfrow = c(1, 1), tcl = -0.15)
plot(coxdiag, yaxt = "n", main = "Grambsch & Terneau Plot (before matching)")
axis(2, las = 1)
```



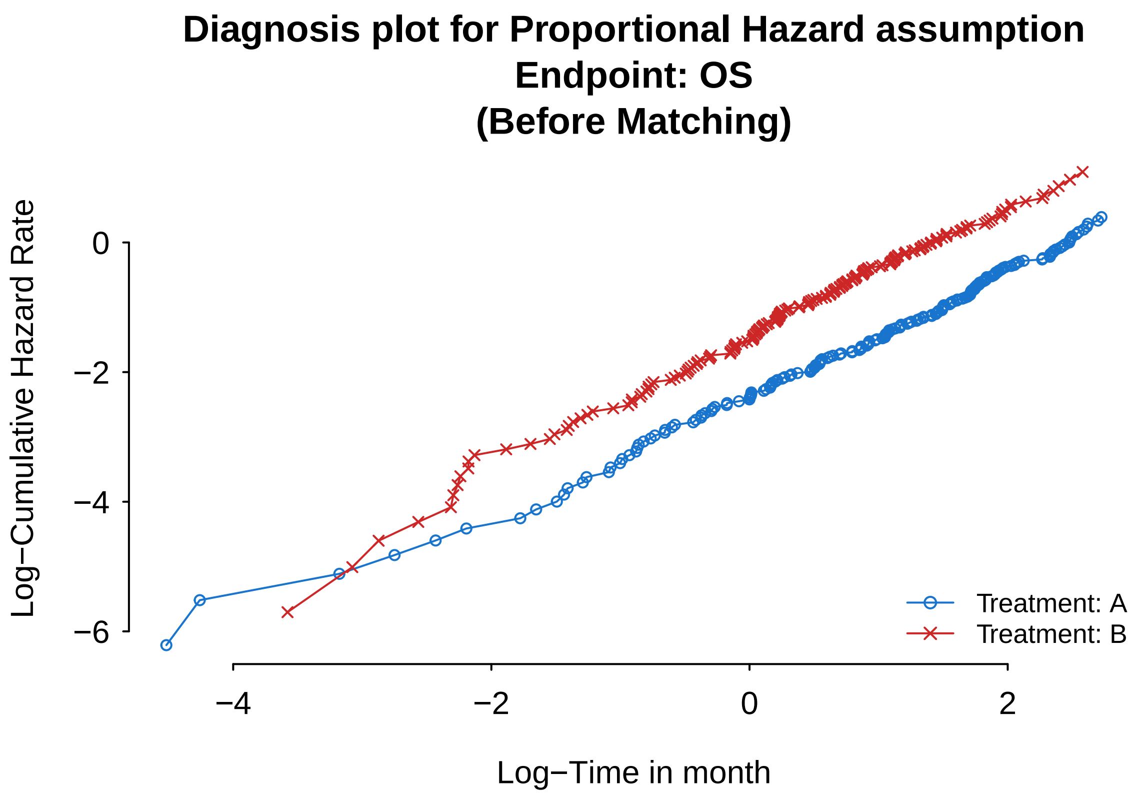

```r
coxdiag_adj
```

```
##       chisq df    p
## ARM 0.0438  1 0.83
```

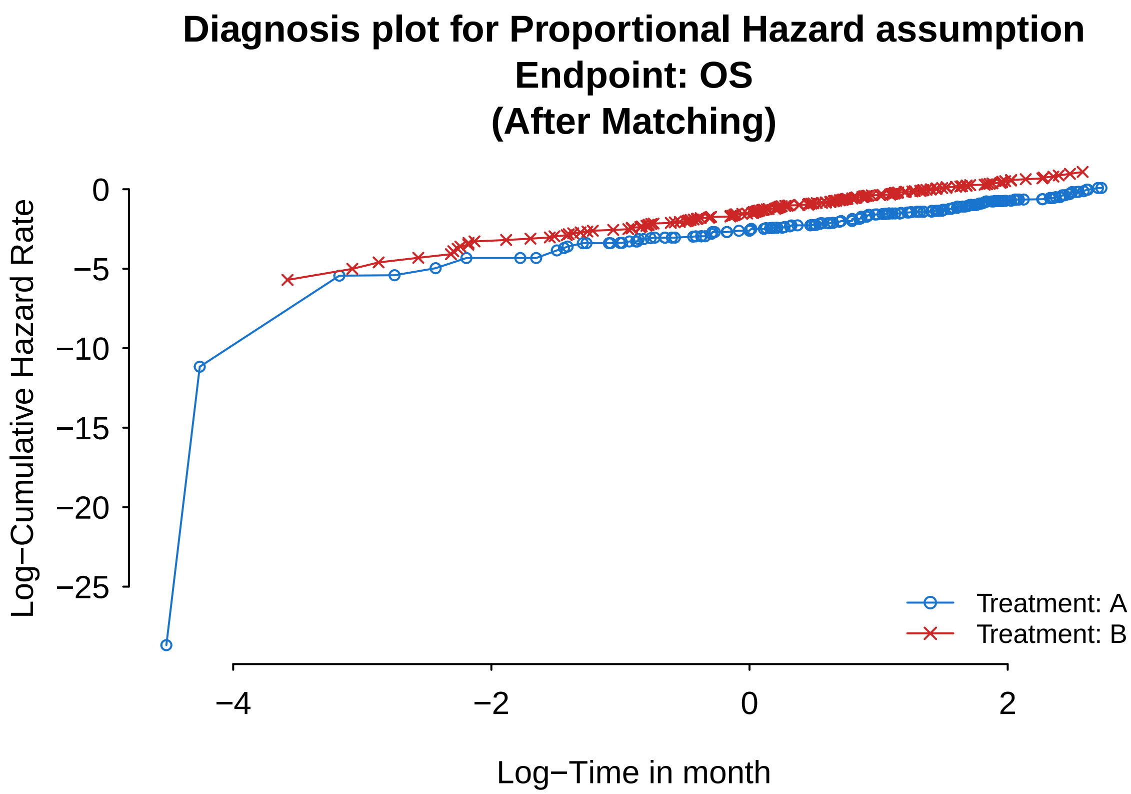
```r
par(mfrow = c(1, 1), tcl = -0.15)
plot(coxdiag_adj, yaxt = "n", main = "Grambsch & Terneau Plot (after matching)")
axis(2, las = 1)
```



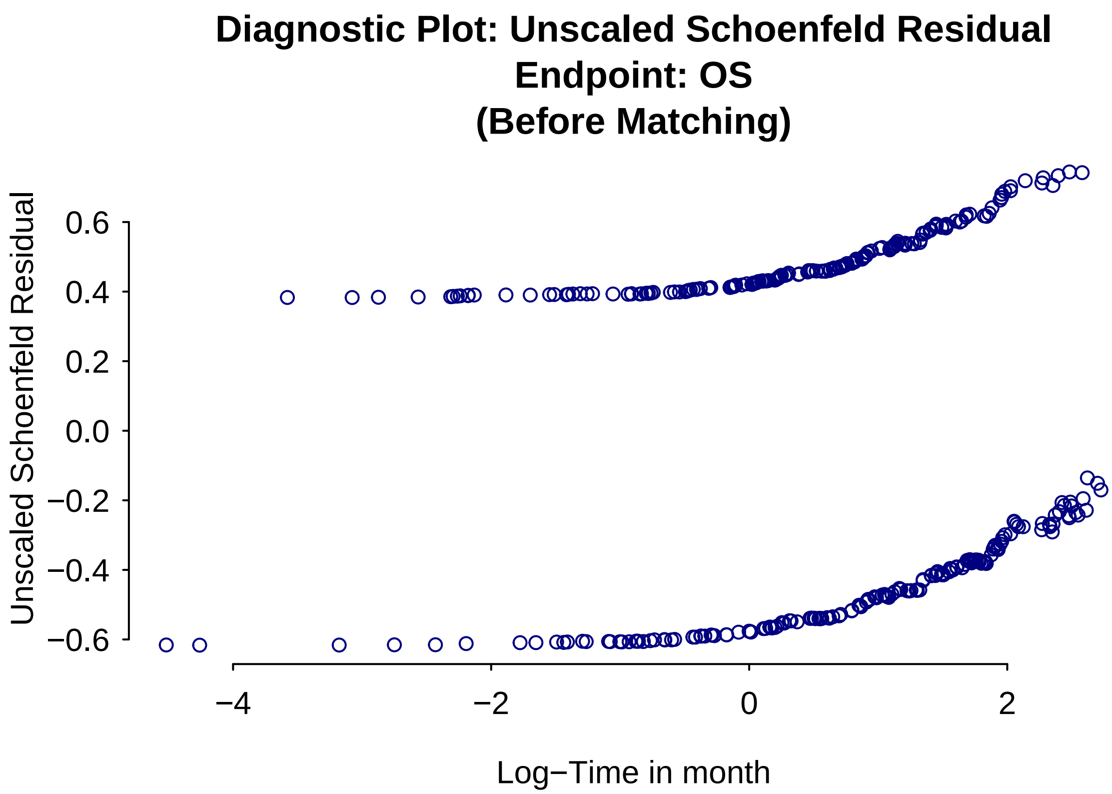

```
# log-cumulative hazard plot
log_cum_haz_plot(kmobj, time_scale = "month", log_time = TRUE, endpoint_name = "OS",
    subtitle = "(Before Matching)")
```



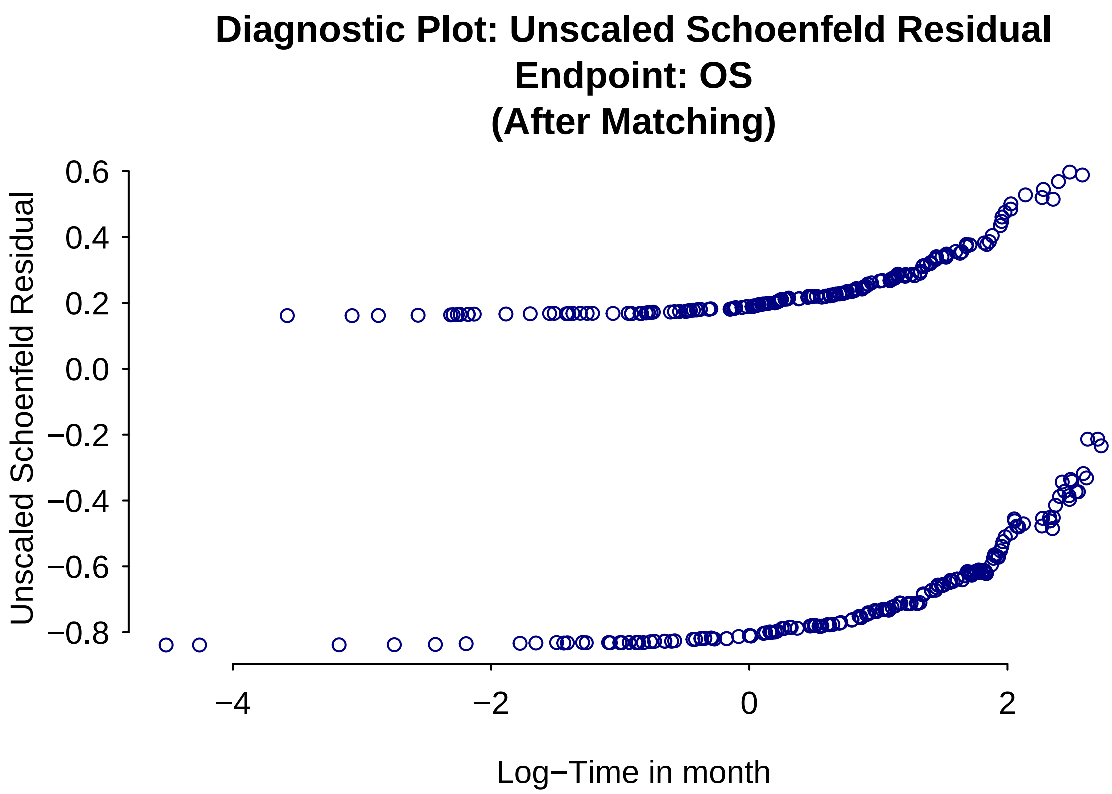

```
log_cum_haz_plot(kmobj_adj, time_scale = "month", log_time = TRUE, endpoint_name = "OS",
    subtitle = "(After Matching)")
```




```r
# schoenfeld residual plot
resid_plot(unweighted_cox, time_scale = "month", log_time = TRUE, endpoint_name = "OS",
    subtitle = "(Before Matching)")
```




```r
resid_plot(weighted_cox, time_scale = "month", log_time = TRUE, endpoint_name = "OS",
    subtitle = "(After Matching)")
```



Analysis using a built-in wrapper

One can do all this analysis in a wrapper

```
# put in wrapper code
```

Binary outcome analysis (TODO)

```
# Simulate response data based on the known proportion of responders
comparator_prop_events <- 0.4

# Calculate number with event. Use round() to ensure we end up with a whole
# number of people. number without an event = Total N - number with event to
# ensure we keep the same number of patients
n_with_event <- round(agd$N * comparator_prop_events, digits = 0)
comparator_binary <- data.frame(RESPONSE = c(rep(1, n_with_event), rep(0, agd$N -
    n_with_event)))
comparator_binary$ARM <- "B"  # need to specify ARM for comparator data
ipd_matched <- weighted_data$data

combined_data_binary <- merge_two_data(comparator_binary, ipd_matched, internal_response_name = "RESPONS
```

```r
unweighted_OR <- glm(formula = RESPONSE ~ ARM, family = binomial(link = "logit"),
    data = combined_data_binary)

# Log odds ratio
log_OR_CI <- cbind(coef(unweighted_OR), confint.default(unweighted_OR, level = 0.95))[2,
    ]

# Odds ratio
OR_CI <- exp(log_OR_CI)
names(OR_CI) <- c("OR", "OR_low_CI", "OR_upp_CI")
OR_CI

# Fit a logistic regression model with weights to estimate the weighted OR
weighted_OR <- suppressWarnings(glm(formula = RESPONSE ~ ARM, family = binomial(link = "logit"),
    data = combined_data_binary, weight = weights))

# Weighted log odds ratio
log_OR_CI_wtd <- cbind(coef(weighted_OR), confint.default(weighted_OR, level = 0.95))[2,
    ]

# Weighted odds ratio
OR_CI_wtd <- exp(log_OR_CI_wtd)
names(OR_CI_wtd) <- c("OR", "OR_low_CI", "OR_upp_CI")
OR_CI_wtd

# Robust standard error
vmod <- clubSandwich::vcovCR(weighted_OR, cluster = 1:dim(combined_data_binary)[1],
    type = "CR2")
coef_res <- clubSandwich::conf_int(weighted_OR, vmod, coef = 2)

OR_CI_robust <- exp(with(coef_res, c(beta, CI_L, CI_U)))
names(OR_CI_robust) <- c("Estimate", "Lower 95% CI", "Upper 95% CI")
OR_CI_robust

# Using sandwich package
V.sw <- sandwich::vcovHC(weighted_OR)  #white's estimator
SD <- sqrt(V.sw[2, 2])
Estimate <- coef(weighted_OR)[2]
OR_CI_robust2 <- exp(c(Estimate, Estimate - 1.96 * SD, Estimate + 1.96 * SD))
names(OR_CI_robust2) <- c("Estimate", "Lower 95% CI", "Upper 95% CI")
OR_CI_robust2
```