Comparative Effectiveness and Personalized Medicine Research Using Real-World Data

Thomas Debray

2023-05-03

Table of contents

1 Preface

Thomas Debray (Smart Data Analysis and Statistics B.V.)

About this book

This book provides practical guidance for estimating the effectiveness of treatments in real-world populations. It explains how real-world data can directly be used or combined with other data sources to derive overall and individualized estimates of treatment effect. The book explains statistical methods for implementing bias adjustments, conducting evidence synthesis and individualizing treatment effect, whilst also providing illustrative examples and supporting software. The chapters and contents of the book are written by leading experts, with a track record in the generation and/or evaluation of real-world evidence.

This book is intended as a pivotal textbook for statisticians, epidemiologists, methodologists, regulators and/or regulatory scientists considering, undertaking or appraising the real-world evidence of treatment effectiveness. It covers key concepts and stages to derive and evaluate treatment effect estimates for entire populations and specific individuals. The book offers a conceptual framework towards estimating treatment effects at both the population and individualized level, where modelling methods may include traditional regression-based and machine learning methods.

Motivation

Although randomized clinical trials traditionally form the cornerstone of comparative effectiveness research, there is a growing demand to consider evidence from "real-world data" (RWD) in clinical decision-making. These data are often available from observational cohort studies, administrative databases, and patient registries, and may offer additional insights into the comparative effectiveness and safety of treatments. Yet, the analysis of RWD and the evaluation of real-world evidence face many operational and methodological challenges.

In this book, we aim to address three current needs. First, this book will offer the guidance that is currently lacking on assessing the quality of RWD and on implementing appropriate

statistical methods to reduce bias of single study estimates of treatment effects. Second, this book will provide researchers with advanced approaches to pooling estimates from multiple non-randomized studies for which traditional evidence synthesis methods are not suitable. Finally, to answer the growing need to translate average estimates of treatment effects to individualized clinical decision-making, this book will present recent methods for more tailored approaches where patient characteristics are used to derive their individualized prognosis and treatment benefit.

This book aims to explain key principles and state-of-the-art methods for deriving treatment effects in entire populations and specific individuals using RWD. It will not only discuss statistical theory by key experts in the field; it will also provide illustrative examples and practical guidance for implementation in R. In short, the book aims to prepare a new generation of researchers who wish to generate and integrate evidence from both randomized and non-randomized data sources to investigate the real-world effectiveness of treatments in populations and individual patients.

Contents

The book is divided into six sections:

1. **Introduction**. This section introduces the relevance of real-world data for conducting comparative effectiveness research, and discusses various concerns regarding their use.
2. **Principles of treatment effect estimation using real-world data**. In this section, we discuss key principles of treatment effect estimation in non-randomized data sources. We explain methods to adjust for confounding (including propensity score analysis and disease risk score analysis) and missing data when estimating the treatment effect for a specific (sub)population.
3. **Principles of evidence synthesis**. In this section, we discuss statistical methods for estimating the treatment effect using (individual participant and/or aggregate) data from multiple studies. To this purpose, key principles of meta-analysis are introduced and explained, including the standard fixed effect and random effects meta-analysis models, methods for individual patient data (IPD) meta-analysis, methods for network meta-analysis, and methods for data-driven and tailored bias adjustment.
4. **Advanced modelling issues for dealing with additional bias in both randomized and non-randomized data sources**. In this section, we discuss advanced statistical and machine learning methods for dealing with time-varying confounding, informative visit schedules, and measurement error.
5. **Individualizing treatment effects for personalized medicine**. In this section, we discuss statistical methods to estimate and evaluate individualized treatment effects.
6. Closing

2 Validity control and quality assessment of real-world data and real-world evidence

Christina ReadThomas Debray (Smart Data Analysis and Statistics B.V.)

```
library(readxl)
library(robvis)
```

The quality of real-world data is often suboptimal and can therefore lead to bias when generating real-world evidence (RWE). In this chapter, we will introduce key quality concerns of RWD, including their accuracy, completeness, and timeliness. Subsequently, we will discuss which steps can be taken to assess the quality of RWD, and determine their fitness for use. The chapter will also introduce directed acyclic graphs to explain how the analysis of RWD may be affected by different types of bias. We will put particular focus on confounding bias, selection bias, and information bias, and explain how these biases can be addressed by referring to specific chapters from the book. Finally, the chapter presents common quality appraisal tools that can be used to assess the quality of real-world evidence (for instance when conducting a systematic review).

2.1 Example code

A risk of bias assessment was conducted in the COVID-NMA review. We can create a summary table of risk of bias assessment and produce a traffic light plot as follows:

```
Risk_of_Bias <- read_excel("resources/RoB-covid.xlsx")

#creation of traffic light plot
trafficlight_rob <- rob_traffic_light(data = Risk_of_Bias, tool = "ROB2")
trafficlight_rob
```


Risk of bias domains
D1
D2
D3
D4
D5
Overall
Study
Abd−Elsalam et al
Ader
Beigel (ACTT−1)
Cao et al
Cavalcanti et al
Criner*
Davoudi−Monfared
Horby (RECOVERY; DEXA)
Horby (RECOVERY; LOPI)
Horby (RECOVERY; TOCI)
Islam^
Lyngbakken et al
Mahajan
Pan (WHO Solidarity; HCQ)
Pan (WHO Solidarity; INTB)
Pan (WHO Solidarity; LOPI)
Pan (WHO Solidarity; REM)
Spinner
Wang
Broman
Declerq
Gordon (REMAP−CAP)
Hermine
Rohmani
Rosas (COVACTA)
Rutgers
Salama(EMPACTA)
Salvarani*
Soin (COVINTOC)
Stone*
Tomazini
Talaschian
Ulrich et al
Veiga
Domains:
D1: Bias arising from the randomization process.
D2: Bias due to deviations from intended intervention.
D3: Bias due to missing outcome data.
D4: Bias in measurement of the outcome.
D5: Bias in selection of the reported result.
Judgement
High
Some concerns
Low

Version info

This chapter was rendered using the following version of R and its packages:

```
R version 4.2.3 (2023-03-15 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19045)

Matrix products: default

locale:
[1] LC_COLLATE=Dutch_Netherlands.utf8  LC_CTYPE=Dutch_Netherlands.utf8
[3] LC_MONETARY=Dutch_Netherlands.utf8 LC_NUMERIC=C
[5] LC_TIME=Dutch_Netherlands.utf8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] robvis_0.3.0.900 readxl_1.4.3

loaded via a namespace (and not attached):
 [1] cellranger_1.1.0  pillar_1.9.0      compiler_4.2.3    tools_4.2.3
 [5] digest_0.6.31     jsonlite_1.8.7    evaluate_0.23     lifecycle_1.0.4
 [9] tibble_3.2.1      gtable_0.3.4      pkgconfig_2.0.3   rlang_1.1.1
[13] cli_3.6.1         rstudioapi_0.15.0 yaml_2.3.7        xfun_0.39
[17] fastmap_1.1.1     withr_2.5.2       stringr_1.5.1     dplyr_1.1.2
[21] knitr_1.45        generics_0.1.3    vctrs_0.6.3       grid_4.2.3
[25] tidyselect_1.2.0  glue_1.6.2        R6_2.5.1          fansi_1.0.4
[29] rmarkdown_2.25    farver_2.1.1      tidyr_1.3.0       purrr_1.0.1
[33] ggplot2_3.4.4     magrittr_2.0.3    scales_1.2.1      codetools_0.2-19
[37] htmltools_0.5.5   colorspace_2.1-0  utf8_1.2.3        stringi_1.7.12
[41] munsell_0.5.0
```

3 Confounding adjustment using propensity score methods

Tammy Jiang (Biogen)
Thomas Debray (Smart Data Analysis and Statistics B.V.)

3.1 Introduction

The purpose of this document is to provide example R code that demonstrates how to estimate the propensity score and implement matching, stratification, weighting, and regression adjustment for the continuous propensity score. In this example using simulated data, we have two disease modifying therapies (DMT1 and DMT0) and the outcome is the number of post-treatment multiple sclerosis relapses during follow-up. We will estimate the average treatment effect in the treated (ATT) using propensity score matching, stratification, and weighting. We will estimate the average treatment effect in the population (ATE) using regression adjustment for the continuous propensity score. The treatment effects can be interpreted as annualized relapse rate ratios (ARR).

We consider an example dataset with the following characteristics:

```
head(dat)
```

```
   age female prevDMTefficacy premedicalcost numSymptoms prerelapse_num
1:  50      1            None        3899.61           1              1
2:  51      0            None        9580.51           1              0
3:  56      0            None        4785.89           1              0
4:  44      1            None        8696.80           1              1
5:  63      0            None        2588.03           1              0
6:  28      1            None        5435.57           1              0
   treatment y      years      Iscore
1:      DMT1 0 1.78507871 Moderate A1
2:      DMT1 0 0.01368925     High A1
```

```
3:      DMT1 2 3.25530459     High A1
4:      DMT1 2 5.73853525     Neutral
5:      DMT1 0 1.31143053     High A1
6:      DMT1 0 0.59137577 Moderate A0
```

3.2 Comparing baseline characteristics

- `DMT1` is the treatment group and `DMT0` is the control group
- `prevDMTefficacy` is previous DMT efficacy (none, low efficacy, and medium/high efficacy)
- `prerelapse_num` is the number of previous MS relapses

	DMT0	DMT1
n	2300	7700
age (mean (SD))	51.39 (8.32)	44.25 (9.79)
female = 1 (%)	1671 (72.65)	5915 (76.82)
prevDMTefficacy (%)		
None	1247 (54.22)	3171 (41.18)
Low_efficacy	261 (11.35)	858 (11.14)
Medium_high_efficacy	792 (34.43)	3671 (47.68)
prerelapse_num (mean (SD))	0.39 (0.62)	0.46 (0.68)

3.3 Estimating the propensity score

3.3.1 Logistic regression

We sought to restore balance in the distribution of baseline covariates in patients treated with DMT1 (index treatment) and DMT0 (control tratment). We fit a multivariable logistic regression model in which treatment was regressed on baseline characteristics including age, sex, previous DMT efficacy, and previous number of relapses.

```r
# Fit logistic regression model
ps.model <- glm(treatment ~ age + female + prevDMTefficacy + prerelapse_num,
                data = dat, family = binomial())

# Summary of logistic regression model
summary(ps.model)
```

```
Call:
glm(formula = treatment ~ age + female + prevDMTefficacy + prerelapse_num,
    family = binomial(), data = dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7949   0.2585   0.5220   0.7478   1.5033

Coefficients:
                                   Estimate Std. Error z value Pr(>|z|)
(Intercept)                        4.809473   0.157127  30.609  < 2e-16 ***
age                               -0.086708   0.002996 -28.939  < 2e-16 ***
female1                            0.253611   0.057664   4.398 1.09e-05 ***
prevDMTefficacyLow_efficacy        0.310394   0.083022   3.739 0.000185 ***
prevDMTefficacyMedium_high_efficacy 0.660266   0.054393  12.139  < 2e-16 ***
prerelapse_num                     0.156318   0.039288   3.979 6.93e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 10786  on 9999  degrees of freedom
Residual deviance:  9597  on 9994  degrees of freedom
AIC: 9609

Number of Fisher Scoring iterations: 5
```

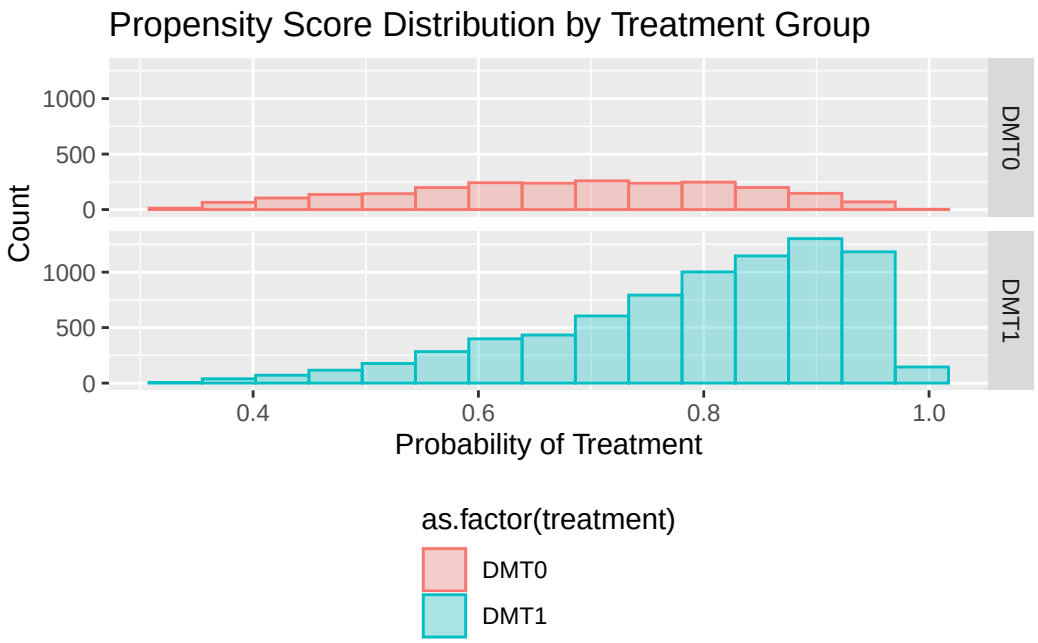
```r
# Extract propensity scores
dat$ps <- predict(ps.model, data = dat, type = "response")
```

3.3.2 Assessing overlap

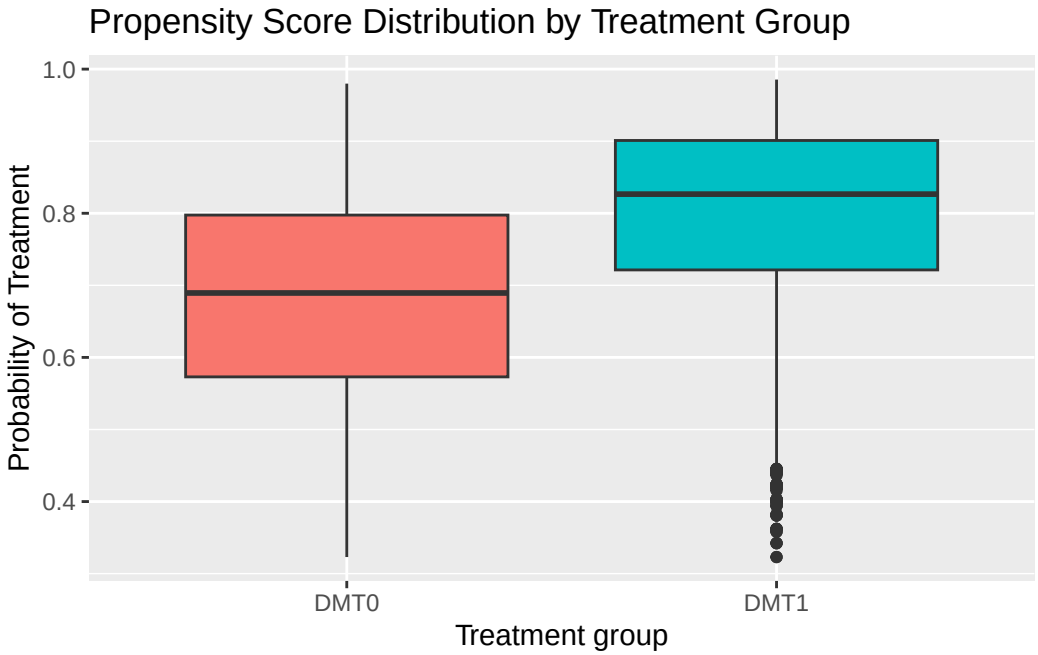
We examined the degree of overlap in the distribution of propensity scores across treatment groups using histograms and side-by-side box plots.

```r
# Histogram
ggplot(dat, aes(x = ps, fill = as.factor(treatment), color = as.factor(treatment))) +
  geom_histogram(alpha = 0.3, position='identity', bins = 15) +
  facet_grid(as.factor(treatment) ~ .) +
```

```
  xlab("Probability of Treatment") +
  ylab("Count") +
  ggtitle("Propensity Score Distribution by Treatment Group") +
  theme(legend.position = "bottom", legend.direction = "vertical")
```




```
# Side-by-side box plots
ggplot(dat, aes(x=as.factor(treatment), y=ps, fill=as.factor(treatment))) +
  geom_boxplot() +
  ggtitle("Propensity Score Distribution by Treatment Group") +
  ylab("Probability of Treatment") +
  xlab("Treatment group") +
  theme(legend.position = "none")
```




```r
# Distribution of propensity scores by treatment groups
summary(dat$ps[dat$treatment == "DMT1"])
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.3230  0.7214  0.8265  0.7970  0.9010  0.9854
```

```r
summary(dat$ps[dat$treatment == "DMT0"])
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.3230  0.5730  0.6894  0.6795  0.7975  0.9799
```

3.4 Propensity score matching

3.4.1 1:1 Optimal full matching without replacement

```
library(MatchIt)

# Use MatchIt package for PS matching
opt <- matchit(treatment ~ age + female + prevDMTefficacy + prerelapse_num,
               data = dat,
               method = "full",
               estimand = "ATT")

opt
```

```
A matchit object
 - method: Optimal full matching
 - distance: Propensity score
             - estimated with logistic regression
 - number of obs.: 10000 (original), 10000 (matched)
 - target estimand: ATT
 - covariates: age, female, prevDMTefficacy, prerelapse_num
```

3.4.2 Assess balance after matching

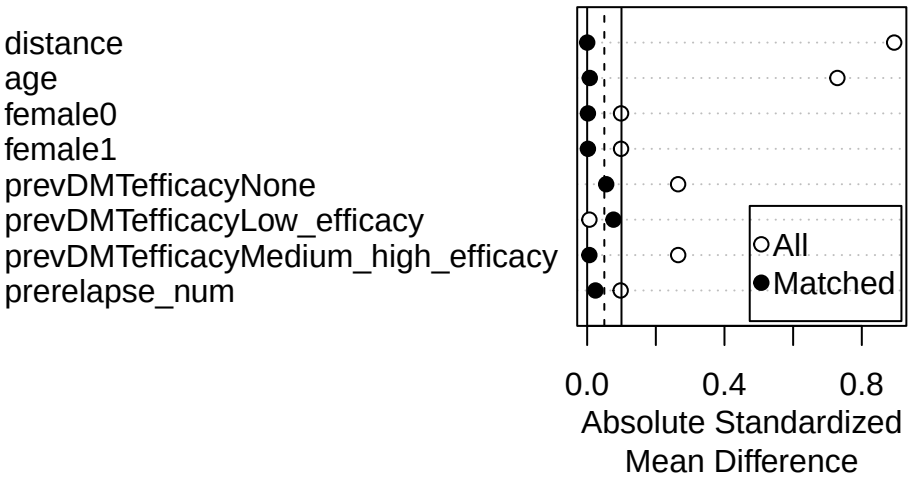
```
summary(opt)
```

```
Call:
matchit(formula = treatment ~ age + female + prevDMTefficacy +
    prerelapse_num, data = dat, method = "full", estimand = "ATT")

Summary of Balance for All Data:
                                  Means Treated Means Control Std. Mean Diff.
distance                             0.7970        0.6795          0.8943
age                                 44.2496       51.3883         -0.7289
female0                              0.2318        0.2735         -0.0987
female1                              0.7682        0.7265          0.0987
prevDMTefficacyNone                  0.4118        0.5422         -0.2649
prevDMTefficacyLow_efficacy          0.1114        0.1135         -0.0065
```

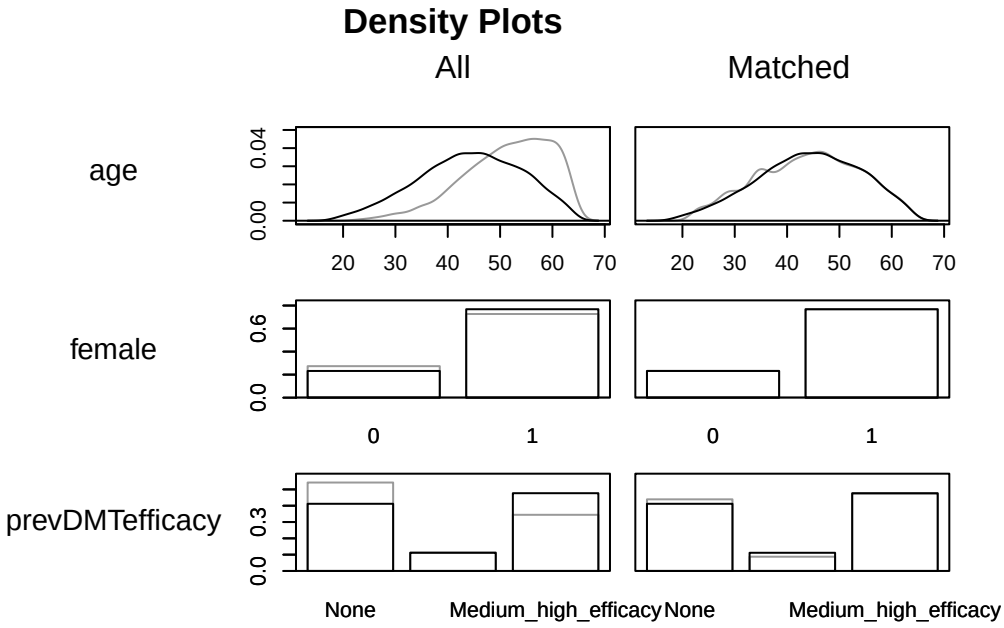
```
prevDMTefficacyMedium_high_efficacy        0.4768        0.3443          0.2651
prerelapse_num                             0.4595        0.3930          0.0976
                                  Var. Ratio eCDF Mean eCDF Max
distance                              0.7873    0.1917   0.3379
age                                   1.3868    0.1519   0.3085
female0                                    .    0.0417   0.0417
female1                                    .    0.0417   0.0417
prevDMTefficacyNone                        .    0.1304   0.1304
prevDMTefficacyLow_efficacy                .    0.0020   0.0020
prevDMTefficacyMedium_high_efficacy        .    0.1324   0.1324
prerelapse_num                        1.1990    0.0133   0.0383

Summary of Balance for Matched Data:
                                  Means Treated Means Control Std. Mean Diff.
distance                             0.7970        0.7970          0.0001
age                                 44.2496       44.1749          0.0076
female0                              0.2318        0.2327         -0.0022
female1                              0.7682        0.7673          0.0022
prevDMTefficacyNone                  0.4118        0.4391         -0.0555
prevDMTefficacyLow_efficacy          0.1114        0.0873          0.0766
prevDMTefficacyMedium_high_efficacy  0.4768        0.4736          0.0064
prerelapse_num                       0.4595        0.4759         -0.0241
                                  Var. Ratio eCDF Mean eCDF Max
distance                              0.9968    0.0007   0.0100
age                                   0.9979    0.0047   0.0300
female0                                    .    0.0009   0.0009
female1                                    .    0.0009   0.0009
prevDMTefficacyNone                        .    0.0273   0.0273
prevDMTefficacyLow_efficacy                .    0.0241   0.0241
prevDMTefficacyMedium_high_efficacy        .    0.0032   0.0032
prerelapse_num                        1.0230    0.0060   0.0229
                                  Std. Pair Dist.
distance                                  0.0012
age                                       0.1158
female0                                   0.2334
female1                                   0.2334
prevDMTefficacyNone                       0.2183
prevDMTefficacyLow_efficacy               0.3742
prevDMTefficacyMedium_high_efficacy       0.3758
prerelapse_num                            0.3690

Sample Sizes:
              Control Treated
```

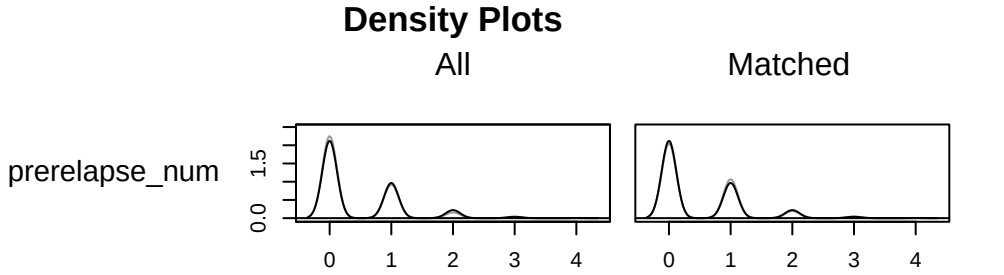
```
All           2300.      7700
Matched (ESS)  253.16    7700
Matched       2300.      7700
Unmatched        0.         0
Discarded        0.         0
```

```r
plot(summary(opt))
```




```r
# black line is treated group, grey line is control group
plot(opt, type = "density", which.xs = vars)
```


Density Plots
All
Matched
age
0.00
0.04
20
30
40
50
60
70
female
0.0
0.6
0
1
prevDMTefficacy
0.0
0.3
None
Medium_high_efficacy


Density Plots
All
Matched
prerelapse_num
0.0
1.5
0
1
2
3
4

3.4.3 Estimating the ATT

We can estimate the ATT in the matched sample using Poisson regression in which the number of post-treatment relapses is regressed on treatment status and follow-up time for each patient (captured by the variable `years`). More details are provided at https://cran.r-project.org/web/packages/MatchIt/vignettes/estimating-effects.html.

```
# Matched data
matched.data <- match.data(opt)

# Poisson regression model
opt.fit <- glm(y ~ treatment + offset(log(years)),
               family = poisson(link = "log"),
               data = matched.data,
               weights = weights)

# Treatment effect estimation
opt.comp <- comparisons(opt.fit,
                        variables = "treatment",
                        vcov = ~subclass,
                        newdata = subset(matched.data, treatment == "DMT1"),
                        wts = "weights",
                        transform_pre = "ratio")
```

```
Warning: The `transform_pre` argument is deprecated. Use `comparison` instead.
```

```
opt.comp |> tidy()
```

```
Warning: The `transform_pre` argument is deprecated. Use `comparison` instead.
```

```
# A tibble: 1 x 8
  term      contrast    estimate std.error statistic  p.value conf.low conf.high
  <chr>     <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
1 treatment mean(DMT1)~    0.772     0.105      7.37 1.68e-13    0.566     0.977
```

As indicated in the summary output above, the annualized relapse rate ratio for DMT1 vs DMT0 among patients treated with DMT0 (ATT) is given as 0.77 with a 95% confidence interval ranging from 0.57 to 0.98.

3.5 Propensity score stratification

3.5.1 Divide sample into quintiles of propensity scores

We will form five mutually exclusive groups of the estimated propensity score.

```
# Divide the PS scores into five strata of roughly equal size
breaks <- quantile(dat$ps, probs = seq(0, 1, by = 0.2))

dat <- dat %>% mutate(ps.strata = cut(ps,
                                      breaks = breaks,
                                      labels = seq(1:5),
                                      include.lowest = TRUE))

# Number of patients in each stratum
table(dat$ps.strata)
```

```
   1    2    3    4    5
2002 2015 1991 1997 1995
```

3.5.2 Assess balance within each propensity score stratum

Within each propensity score stratum, treated and control patients should have similar values of the propensity score and the distribution of baseline covariates should be approximately balanced between treatment groups.

3.5.2.1 Propensity Score Stratum #1

```
tab1.strata1 <- CreateTableOne(vars, data = dat %>% filter(ps.strata == 1),
                               factorVars = c("female", "prevDMTefficacy"),
                               strata = "treatment",
                               test = FALSE)

tab1.strata1.print <- print(tab1.strata1,
                            catDigits = 2,
                            contDigits = 2,
                            smd = TRUE)
```

```
tab1.strata1.print
```

	DMT0	DMT1	SMD
n	901	1101	
age (mean (SD))	58.38 (3.67)	57.45 (3.73)	0.251
female = 1 (%)	605 (67.15)	775 (70.39)	0.070
prevDMTefficacy (%)			0.056
None	650 (72.14)	771 (70.03)	
Low_efficacy	106 (11.76)	130 (11.81)	
Medium_high_efficacy	145 (16.09)	200 (18.17)	
prerelapse_num (mean (SD))	0.29 (0.53)	0.33 (0.56)	0.074

3.5.2.2 Propensity Score Stratum #2

```
tab1.strata2 <- CreateTableOne(vars, data = dat %>% filter(ps.strata == 2),
                               factorVars = c("female", "prevDMTefficacy"),
                               strata = "treatment", test = FALSE)

tab1.strata2.print <- print(tab1.strata2, catDigits = 2, contDigits = 2,
                            smd = TRUE)
```

	DMT0	DMT1	SMD
n	617	1398	
age (mean (SD))	52.18 (4.35)	51.97 (4.22)	0.049
female = 1 (%)	458 (74.23)	1048 (74.96)	0.017
prevDMTefficacy (%)			0.054
None	292 (47.33)	624 (44.64)	
Low_efficacy	69 (11.18)	162 (11.59)	
Medium_high_efficacy	256 (41.49)	612 (43.78)	
prerelapse_num (mean (SD))	0.40 (0.64)	0.41 (0.66)	0.004

3.5.2.3 Propensity Score Stratum #3

```
tab1.strata3 <- CreateTableOne(vars, data = dat %>% filter(ps.strata == 3),
                               factorVars = c("female", "prevDMTefficacy"),
                               strata = "treatment", test = FALSE)
```

```
tab1.strata3.print <- print(tab1.strata3, catDigits = 2, contDigits = 2,
                            smd = TRUE)
```

	DMT0	DMT1	SMD
n	392	1599	
age (mean (SD))	46.73 (4.06)	46.36 (4.08)	0.092
female = 1 (%)	305 (77.81)	1193 (74.61)	0.075
prevDMTefficacy (%)			0.041
None	168 (42.86)	687 (42.96)	
Low_efficacy	52 (13.27)	191 (11.94)	
Medium_high_efficacy	172 (43.88)	721 (45.09)	
prerelapse_num (mean (SD))	0.49 (0.68)	0.47 (0.66)	0.031

3.5.2.4 Propensity Score Stratum #4

```
tab1.strata4 <- CreateTableOne(vars, data = dat %>% filter(ps.strata == 4),
                               factorVars = c("female", "prevDMTefficacy"),
                               strata = "treatment", test = FALSE)

tab1.strata4.print <- print(tab1.strata4, catDigits = 2, contDigits = 2,
                            smd = TRUE)
```

	DMT0	DMT1	SMD
n	269	1728	
age (mean (SD))	41.07 (4.11)	40.88 (4.29)	0.046
female = 1 (%)	203 (75.46)	1356 (78.47)	0.071
prevDMTefficacy (%)			0.084
None	105 (39.03)	634 (36.69)	
Low_efficacy	22 (8.18)	181 (10.47)	
Medium_high_efficacy	142 (52.79)	913 (52.84)	
prerelapse_num (mean (SD))	0.50 (0.69)	0.51 (0.71)	0.012

3.5.2.5 Propensity Score Stratum #5

```
tab1.strata5 <- CreateTableOne(vars, data = dat %>% filter(ps.strata == 5),
                               factorVars = c("female", "prevDMTefficacy"),
                               strata = "treatment", test = FALSE)

tab1.strata5.print <- print(tab1.strata5, catDigits = 2, contDigits = 2,
                            smd = TRUE)
```

	DMT0	DMT1	SMD
n	121	1874	
age (mean (SD))	33.26 (4.95)	32.04 (5.58)	0.233
female = 1 (%)	100 (82.64)	1543 (82.34)	0.008
prevDMTefficacy (%)			0.050
None	32 (26.45)	455 (24.28)	
Low_efficacy	12 (9.92)	194 (10.35)	
Medium_high_efficacy	77 (63.64)	1225 (65.37)	
prerelapse_num (mean (SD))	0.52 (0.66)	0.52 (0.73)	0.004

3.5.3 Estimating and pooling of stratum-specific treatment effects

The overall ATT across strata can be estimated by weighting stratum-specific estimates by the proportion of treated patients in each stratum over all treated patients in the sample.

We first define a function `att.strata.function()` to calculate stratum-specific estimates of the treatment effect:

```
att.strata.function <- function(data, stratum, confint = TRUE) {

  fit <- glm("y ~ treatment + offset(log(years))",
      family = poisson(link = "log"),
      data = data %>% filter(ps.strata == stratum))

  arr <- round(as.numeric(exp(coef(fit)["treatmentDMT1"])), digits = 3)
  ll <- ul <- NA

  if (confint) {
    ll <- round(exp(confint(fit))["treatmentDMT1",1], digits = 3)
    ul <- round(exp(confint(fit))["treatmentDMT1",2], digits = 3)
  }
```

```
  return(c("stratum" = stratum,
           "arr" = arr,
           "ci_lower"  = ll,
           "ci_upper"  = ul))
}

arr.strata <- as.data.frame(t(sapply(1:5, att.strata.function, data = dat)))
arr.strata
```

```
  stratum   arr ci_lower ci_upper
1       1 0.904    0.760    1.076
2       2 0.822    0.696    0.975
3       3 0.798    0.666    0.961
4       4 0.716    0.587    0.881
5       5 0.589    0.463    0.761
```

Subsequently, we define a function `weights.strata.function()` to calculate the weights for each stratum. The weight is the proportion of treated patients in each stratum over all treated patients in the sample:

```
weights.strata.function <- function(data, stratum) {
  n_DMT1_stratum <- nrow(data %>% filter(ps.strata == stratum & treatment == "DMT1"))
  n_DMT1_all <- nrow(data %>% filter(treatment == "DMT1"))
  weight <- n_DMT1_stratum/n_DMT1_all
  return(c("stratum" = stratum, "weight" = weight))
}

weights.strata <- as.data.frame(t(sapply(1:5, weights.strata.function, data = dat)))
weights.strata
```

```
  stratum    weight
1       1 0.1429870
2       2 0.1815584
3       3 0.2076623
4       4 0.2244156
5       5 0.2433766
```

```
# Create table with ARRs and weights for each PS stratum
arr.weights.merged <- merge(arr.strata, weights.strata, by = "stratum")
```

```r
# Calculate the weighted ARR for each stratum
arr.weights.merged <- arr.weights.merged %>%
  mutate(weighted.arr = as.numeric(arr) * weight)

# Sum the weighted ARRs across strata to get the overall ATT
sum(arr.weights.merged$weighted.arr)
```

```
[1] 0.7482462
```

We now define a new function `ps.stratification.bootstrap()` that integrates estimation of the ATT and the PS weights for bootstrapping purposes:

```r
ps.stratification.bootstrap <- function(data, inds) {
  d <- data[inds,]

  d$ps.strata <- cut(d$ps,
                     breaks = c(quantile(dat$ps, probs = seq(0, 1, by = 0.2))),
                     labels = seq(5),
                     include.lowest = TRUE)

  arr.strata <- as.data.frame(t(sapply(1:5, att.strata.function,
                                       data = d, confint = FALSE)))

  weights.strata <- as.data.frame(t(sapply(1:5, weights.strata.function, data = d)))

  return(arr.strata$arr[1] * weights.strata$weight[1] +
           arr.strata$arr[2] * weights.strata$weight[2] +
           arr.strata$arr[3] * weights.strata$weight[3] +
           arr.strata$arr[4] * weights.strata$weight[4] +
           arr.strata$arr[5] * weights.strata$weight[5])
}
```

We can now estimate the treatment effect and its confidence interval using the bootstrap procedure:

```r
library(boot)

set.seed(1854)
arr.stratification.boot <- boot(data = dat,
                                statistic = ps.stratification.bootstrap,
                                R = 1000)
```

We can summarize the bootstrap samples as follows:

```
# Bootstrap estimate of the ARR
median(arr.stratification.boot$t)
```

```
[1] 0.7558609
```

```
# Bootstrap 95% CI of the ARR
boot.ci(arr.stratification.boot, conf = 0.95, type = "perc")
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = arr.stratification.boot, conf = 0.95, type = "perc")

Intervals :
Level     Percentile
95%   ( 0.6833,  0.8366 )
Calculations and Intervals on Original Scale
```

3.6 Propensity score weighting

3.6.1 Calculate propensity score weights for ATT

Propensity score weighting reweights the study sample to generate an artificial population (i.e., pseudo-population) in which the covariates are no longer associated with treatment, thereby removing confounding by measured covariates. For the ATT, the weight for all treated patients is set to one. Conversely, the weight for patients in the control group is set to the propensity score divided by one minus the propensity score, that is, (PS/(1 − PS)). We estimated stabilized weights to address extreme weights.

```
library(WeightIt)

w.out <- weightit(treatment ~ age + female + prevDMTefficacy + prerelapse_num,
                  data = dat,
                  method = "ps",
                  estimand = "ATT")
                  #stabilize = TRUE)
```

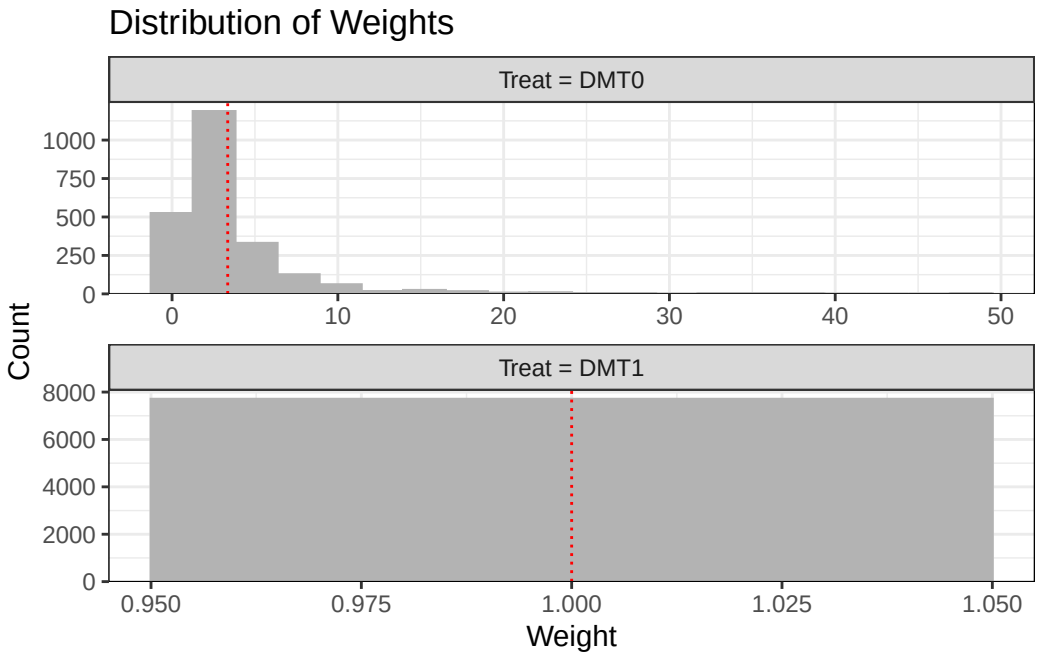
```
w.out
```

```
A weightit object
 - method: "glm" (propensity score weighting with GLM)
 - number of obs.: 10000
 - sampling weights: none
 - treatment: 2-category
 - estimand: ATT (focal: DMT1)
 - covariates: age, female, prevDMTefficacy, prerelapse_num
```

```
summary(w.out)
```

```
                Summary of weights

- Weight ranges:

        Min                                   Max
DMT0 0.4772 |---------------------------| 48.6856
DMT1 1.0000  ||                            1.0000

- Units with the 5 most extreme weights by group:

         9492    8836    6544    9610    4729
 DMT0 32.1027 32.1027 34.3126 38.1817 48.6856
            8       7       4       2       1
 DMT1       1       1       1       1       1

- Weight statistics:

     Coef of Var   MAD Entropy # Zeros
DMT0       1.098 0.673   0.383       0
DMT1       0.000 0.000  -0.000       0

- Effective Sample Sizes:

              DMT0 DMT1
Unweighted 2300.   7700
Weighted   1043.16 7700
```

```
plot(summary(w.out))
```



3.6.2 Assess balance in the weighted sample

```
bal.tab(w.out, stats = c("m", "v"), thresholds = c(m = .05))
```

```
Balance Measures
                                        Type Diff.Adj     M.Threshold
prop.score                          Distance  -0.0045 Balanced, <0.05
age                                  Contin.   0.0054 Balanced, <0.05
female                                Binary   0.0005 Balanced, <0.05
prevDMTefficacy_None                  Binary  -0.0003 Balanced, <0.05
prevDMTefficacy_Low_efficacy          Binary   0.0023 Balanced, <0.05
prevDMTefficacy_Medium_high_efficacy  Binary  -0.0020 Balanced, <0.05
prerelapse_num                       Contin.  -0.0034 Balanced, <0.05
                                    V.Ratio.Adj
prop.score                               0.9926
age                                      1.0102
female                                        .
prevDMTefficacy_None                          .
```

```
prevDMTefficacy_Low_efficacy                    .
prevDMTefficacy_Medium_high_efficacy            .
prerelapse_num                                1.0941

Balance tally for mean differences
                    count
Balanced, <0.05         7
Not Balanced, >0.05     0

Variable with the greatest mean difference
 Variable Diff.Adj     M.Threshold
      age   0.0054 Balanced, <0.05

Effective sample sizes
              DMT0 DMT1
Unadjusted 2300.   7700
Adjusted   1043.16 7700
```

3.6.3 Estimate the ATT

One way to estimate the ATT is to use the survey package. The function `svyglm()` generates model-robust (Horvitz-Thompson-type) standard errors by default, and thus does not require additional adjustments.

```r
library(survey)

weighted.data <- svydesign(ids = ~1, data = dat, weights = ~w.out$weights)

weighted.fit <- svyglm(y ~ treatment + offset(log(years)),
                       family = poisson(link = "log"),
                       design = weighted.data)

exp(coef(weighted.fit)["treatmentDMT1"])
```

```
treatmentDMT1
    0.7083381
```

```r
exp(confint(weighted.fit))["treatmentDMT1",]
```

```
    2.5 %    97.5 %
0.6245507 0.8033662
```

As indicated above, propensity score weighting yielded an ATT estimate of 0.71 (95% CI: 0.62; 0.8).

An alternative approach is to use `glm()` to estimate the treatment effect and calculate robust standard errors.

```
# Alternative way to estimate treatment effect
weighted.fit2 <- glm(y ~ treatment + offset(log(years)),
                     family = poisson(link = "log"),
                     data = dat,
                     weights = w.out$weights)

# Extract the estimated ARR
exp(coef(weighted.fit2))["treatmentDMT1"]
```

```
treatmentDMT1
    0.7083381
```

```
# Calculate robust standard error and p-value of the log ARR
coeftest(weighted.fit2, vcov. = vcovHC)["treatmentDMT1",]
```

```
     Estimate    Std. Error       z value      Pr(>|z|)
-3.448337e-01  6.442745e-02 -5.352280e+00  8.685284e-08
```

```
# Derive 95% confidence interval of the ARR
exp(lmtest::coefci(weighted.fit2,
       level = 0.95, # 95% confidence interval
       vcov. = vcovHC)["treatmentDMT1",])
```

```
    2.5 %    97.5 %
0.6243094 0.8036767
```

Using this approach, the ATT estimate was 0.71 (95% CI: 0.62; 0.8).

3.7 Regression adjustment for the propensity score for the ATE

In this approach, a regression model is fitted to describe the observed outcome as a function of the received treatment and the estimated propensity score:

```r
ps.reg.fit <- glm(y ~ treatment + ps + offset(log(years)),
                  family = poisson(link = "log"),
                  data = dat)

summary(ps.reg.fit)
```

```
Call:
glm(formula = y ~ treatment + ps + offset(log(years)), family = poisson(link = "log"),
    data = dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0160  -0.7336  -0.4441  -0.1352   4.2634

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.99585    0.10359 -19.266  < 2e-16 ***
treatmentDMT1 -0.25598    0.04431  -5.777 7.60e-09 ***
ps             1.07521    0.13878   7.748 9.36e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 7514.7  on 9999  degrees of freedom
Residual deviance: 7443.0  on 9997  degrees of freedom
AIC: 12378

Number of Fisher Scoring iterations: 6
```

```r
# ATE
exp(coef(ps.reg.fit))["treatmentDMT1"]
```

```
treatmentDMT1
    0.7741606
```

```
Waiting for profiling to be done...
Waiting for profiling to be done...
```

Bootstrapped confidence intervals can be obtained as follows:

```r
# Function to bootstrap for 95% CIs
ps.reg.bootstrap <- function(data, inds) {
  d <- data[inds,]

  fit <- glm(y ~ treatment + ps + offset(log(years)),
             family = poisson(link = "log"),
             data = d)

  return(exp(coef(fit))["treatmentDMT1"])
}

set.seed(1854)

# Generate 1000 bootstrap replicates
arr.boot <- boot(dat, statistic = ps.reg.bootstrap, R = 1000)

# Extract the median annualized relapse rate across 1000 bootstrap replicates
median(arr.boot$t)
```

```
[1] 0.7750426
```

```r
# Bootstrap 95% CI of the ARR
boot.ci(arr.boot, conf = 0.95, type = "perc")
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = arr.boot, conf = 0.95, type = "perc")

Intervals :
Level     Percentile
95%   ( 0.7007,  0.8547 )
Calculations and Intervals on Original Scale
```

3.8 Overview

Method	Estimand	Estimate	95% CI (lower)	95% CI (upper)
Optimal full matching	ATT	0.7715105	0.5663852	0.9766358
Propensity score stratification	ATT	0.7482462	NA	NA
Propensity score stratification (with bootstrapping)	ATT	0.7558609	0.6832955	0.8365798
Propensity score weighting	ATT	0.7083381	0.6245507	0.8033662
Propensity score weighting (robust SE)	ATT	0.7083381	0.6243094	0.8036767
PS regression adjustment	ATE	0.7741606	0.7101080	0.8448218
PS regression adjustment (bootstrapping)	ATE	0.7750426	0.7006837	0.8546834

Version info

This chapter was rendered using the following version of R and its packages:

```
R version 4.2.3 (2023-03-15 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19045)

Matrix products: default

locale:
[1] LC_COLLATE=Dutch_Netherlands.utf8  LC_CTYPE=Dutch_Netherlands.utf8
[3] LC_MONETARY=Dutch_Netherlands.utf8 LC_NUMERIC=C
[5] LC_TIME=Dutch_Netherlands.utf8

attached base packages:
[1] grid      stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
 [1] WeightIt_0.14.2        boot_1.3-28.1          MatchIt_4.5.5
 [4] sandwich_3.0-2         truncnorm_1.0-9        tableone_0.13.2
 [7] survey_4.2-1           survival_3.5-5         Matrix_1.5-4.1
[10] MASS_7.3-60            marginaleffects_0.15.0 lmtest_0.9-40
[13] zoo_1.8-12             knitr_1.45             ggplot2_3.4.4
[16] data.table_1.14.8      cobalt_4.5.2           dplyr_1.1.2

loaded via a namespace (and not attached):
 [1] tidyselect_1.2.0  xfun_0.39         mitools_2.4       haven_2.5.2
```

```
 [5] splines_4.2.3     lattice_0.21-8    labelled_2.12.0   colorspace_2.1-0
 [9] vctrs_0.6.3       generics_0.1.3    htmltools_0.5.5   yaml_2.3.7
[13] utf8_1.2.3        rlang_1.1.1       e1071_1.7-13      pillar_1.9.0
[17] glue_1.6.2        withr_2.5.2       DBI_1.1.3         lifecycle_1.0.4
[21] munsell_0.5.0     gtable_0.3.4      codetools_0.2-19  evaluate_0.23
[25] labeling_0.4.3    forcats_1.0.0     fastmap_1.1.1     class_7.3-22
[29] fansi_1.0.4       optmatch_0.10.7   Rcpp_1.0.10       checkmate_2.3.0
[33] backports_1.4.1   scales_1.2.1      jsonlite_1.8.7    farver_2.1.1
[37] chk_0.9.1         hms_1.1.3         digest_0.6.31     insight_0.19.6
[41] cli_3.6.1         tools_4.2.3       magrittr_2.0.3    proxy_0.4-27
[45] tibble_3.2.1      crayon_1.5.2      pkgconfig_2.0.3   rlemon_0.2.1
[49] rmarkdown_2.25    rstudioapi_0.15.0 R6_2.5.1          compiler_4.2.3
```

References

4 Effect Modification Analysis within the Propensity score Framework

Mohammad Ehsanul Karim (University of British Columbia)

Observational comparative effectiveness studies often adopt propensity score analysis to adjust for confounding. Although this approach is relatively straightforward to implement, careful thought is needed when treatment effect heterogeneity is present. This chapter illustrates the estimation of subgroup-specific treatment effects using (traditional) covariate adjustment methods, propensity score matching, propensity score weighting, propensity score stratification, and covariate adjustment using propensity scores.

4.1 Simulation

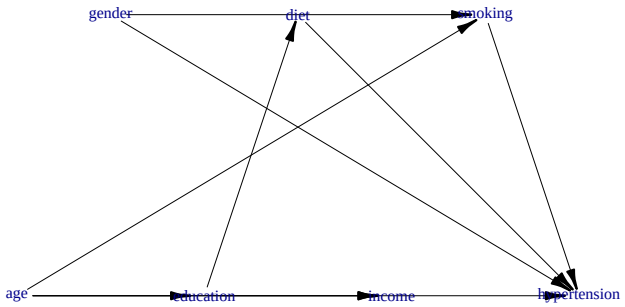
We will use the following data-generation model:

```
require(simcausal)
D <- DAG.empty()
D <- D +
  node("age", distr = "rnorm",
       mean = 2, sd = 4) +
  node("gender", distr = "rbern",
       prob = plogis(4)) +
  node("education", distr = "rbern",
       prob = plogis(3 + 5 * age)) +
  node("diet", distr = "rbern",
       prob = plogis(1 - 3 * education)) +
  node("income", distr = "rbern",
       prob = plogis(2 - 5 * education - 4 * age)) +
  node("smoking", distr = "rbern",
       prob = plogis(1 + 1.2 * gender + 2 * age)) +
  node("hypertension", distr = "rbern",
```

```
        prob = plogis(1 + log(3) * diet +
                        log(1.3) * age +
                        log(3.5) * smoking +
                        log(0.5) * gender))
Dset <- set.DAG(D)
```

Below is the diagram, with pink lines representing open backdoor path.

```
using the following vertex attributes:

NAdarkbluenone100.50

using the following edge attributes:

black0.210.60.5
```



We can now generate an example dataset:

```
Obs.Data <- sim(DAG = Dset, n = 50000, rndseed = 123)
Obs.Data$smoking <- as.character(Obs.Data$smoking)
Obs.Data$income <- as.factor(Obs.Data$income)
```

```r
Obs.Data$income <- relevel(Obs.Data$income, ref = "1")
```

Sample data from the hypothetical example of association between hypertension and smoking, where other variables such as income, age [centered], gender, education and diet also plays a role in the data generation process.

	age	gender	education	diet	income	smoking	hypertension
34901	12.29	1	1	1	0	1	1
149	10.40	1	1	0	0	1	1
10060	2.99	1	1	0	0	1	0
22220	-4.31	0	0	0	1	0	1
9979	-6.44	0	0	0	1	0	1

4.2 Covariate adjustment

4.2.1 Interaction approach

Below, we estimate a logistic regression model to assess whether the effect of smoking (the exposure) on hypertension is modified by income levels. This model considers the following variables:

- Outcome: `hypertension`
- Exposure variables: `smoking` and `income`
- Confounders: `age` and `gender`

```r
require(jtools)

fit.w.em <- glm(hypertension ~ smoking * income + age + gender,
                family = binomial(link = "logit"),
                data = Obs.Data)

results.model <- summ(fit.w.em, exp = TRUE)
```

Results indicate that the interaction between smoking status and income level is statistically significant ($p = 0.02$).

If we expand previous model to adjust for an additional confounder `education`, we have:

```r
fit.w.int <- glm(hypertension ~ smoking * income + age + gender + education,
                 family = binomial(link = "logit"),
```

	exp(Est.)	2.5%	97.5%	z val.	p
(Intercept)	5.46	4.37	6.82	14.97	0.00
smoking1	2.93	2.60	3.30	17.69	0.00
income0	0.48	0.41	0.57	-8.28	0.00
age	1.29	1.27	1.31	36.77	0.00
gender	0.54	0.43	0.67	-5.55	0.00
smoking1:income0	1.27	1.04	1.56	2.33	0.02

```
                data = Obs.Data)

results.int.model <- summ(fit.w.int, exp = TRUE)
```

	exp(Est.)	2.5%	97.5%	z val.	p
(Intercept)	5.69	4.56	7.11	15.31	0.00
smoking1	3.35	2.95	3.79	18.85	0.00
income0	1.09	0.85	1.40	0.68	0.49
age	1.30	1.28	1.32	37.32	0.00
gender	0.54	0.43	0.67	-5.58	0.00
education	0.42	0.35	0.51	-8.87	0.00
smoking1:income0	1.10	0.90	1.35	0.93	0.35

The interaction term between income and smoking is no longer statistically significant (p = 0.35).

We can generate a summary report from aforementioned effect modification analysis.

```
library(interactionR)

em.object <- interactionR(fit.w.em,
                          exposure_names = c("income0", "smoking1"),
                          ci.type = "mover", ci.level = 0.95,
                          em = TRUE, recode = FALSE)
```

The table below depicts the adjusted odds ratios for income levels (`high = 0`, and `low = 1`). The variables `CI.ll` and `CI.ul` depict the lower and upper limits of the 95 percent confidence intervals, `OR11` = $OR_{A=1,M=1}$, `OR10` = $OR_{A=1}$, `OR01` = $OR_{M=1}$ and `OR00` captures the reference.

Similarly, for the analysis adjusting for an additional confounder `education`, we have:

Table 4.1: Summary report from an interaction analysis when investigating association between two exposure variables (smoking and income) and hypertension.

Measures	Estimates	CI.ll	CI.ul
OR00	1.00	NA	NA
OR01	2.93	2.60	3.30
OR10	0.48	0.41	0.57
OR11	1.80	1.63	1.98
OR(smoking1 on outcome [income0==0]	2.93	2.60	3.30
OR(smoking1 on outcome [income0==1]	3.72	3.14	4.41
Multiplicative scale	1.27	1.04	1.56
RERI	-0.61	-0.98	-0.29

Table 4.2: Summary report from an interaction analysis when investigating association between two exposure variables (smoking and income) and hypertension.

Measures	Estimates	CI.ll	CI.ul
OR00	1.00	NA	NA
OR01	1.09	0.85	1.40
OR10	3.35	2.95	3.79
OR11	4.02	3.29	4.92
OR(income0 on outcome [smoking1==0]	1.09	0.85	1.40
OR(income0 on outcome [smoking1==1]	1.20	1.00	1.45
OR(smoking1 on outcome [income0==0]	3.35	2.95	3.79
OR(smoking1 on outcome [income0==1]	3.69	3.11	4.37
Multiplicative scale	1.10	0.90	1.35
RERI	0.59	0.03	1.27
AP	0.15	0.00	0.26
SI	1.24	1.01	1.53

```
# test run with additive model
Obs.Data$smoking <- as.numeric(as.character(Obs.Data$smoking))
Obs.Data$income <- as.numeric(as.character(Obs.Data$income))
fit.w.int.add <- glm(hypertension ~ smoking * income + age + gender + education,
                     family = gaussian(link = "identity"), data = Obs.Data)
interactions::sim_slopes(fit.w.int.add,
                         pred = smoking,
                         modx = income,
                         exp = TRUE,
```

```
                          robust = TRUE,
                          confint = TRUE,
                          data = Obs.Dat)
```

```
JOHNSON-NEYMAN INTERVAL

When income is INSIDE the interval [-3.27, 16.87], the slope of smoking is
p < .05.

Note: The range of observed values of income is [0.00, 1.00]

SIMPLE SLOPES ANALYSIS

Slope of smoking when income = 0.00 (0):

  Est.   S.E.   2.5%   97.5%   t val.      p
------ ------ ------ ------- -------- ------
  0.25   0.02   1.24    1.34    12.76   0.00

Slope of smoking when income = 1.00 (1):

  Est.   S.E.   2.5%   97.5%   t val.      p
------ ------ ------ ------- -------- ------
  0.28   0.01   1.30    1.34    34.53   0.00
```

4.2.2 Stratification

This approach involves estimating a regression model in different strata of the discrete effect modifier `income`:

```
# Estimate the prognostic effect of smoking in low income individuals
fit.income1 <- glm(hypertension ~ smoking + age + gender,
            family = binomial(link = "logit"),
            data = subset(Obs.Data, income == 1))

# Estimate the prognostic effect of smoking in high income individuals
fit.income0 <- glm(hypertension ~ smoking + age + gender,
            family = binomial(link = "logit"),
            data = subset(Obs.Data, income == 0))
```

The table below summarizes the adjusted odds ratios for smoking across the different income levels (`low = 1`, and `high = 0`) as obtained using the stratified approach.

Value of income	Estimate	2.5 %	97.5 %	z value	p value
1	3.07	2.71	3.47	17.65	0
0	3.59	3.02	4.26	14.57	0

Note that we can obtain the same results by estimating a regression model with an interaction term between the modifier and all covariates:

```
fit.all.int <- glm(hypertension ~ income * (smoking + age + gender),
                   family = binomial(link = "logit"), data = Obs.Data)

# Odds ratio for smoking in individuals with low income
exp(coef(fit.all.int)["smoking"])
```

```
smoking
3.59026
```

```
# Odds ratio for smoking in individuals with high income
exp(coef(fit.all.int)["smoking"] + coef(fit.all.int)["income:smoking"])
```

```
 smoking
3.066878
```

4.3 Propensity score matching

4.3.1 Stratification with exact matching within subgroups

We simulate another example dataset using aforementioned DAG, but restrict the sample size to 5000 individuals to reduce computational burden.

```
set.seed(123)
Obs.Data <- sim(DAG = Dset, n = 5000, rndseed = 123)
```

We first estimate the propensity of smoking in the high-income group (`income == 0`):

```
require(MatchIt)

match.income.0 <- matchit(smoking ~ age + gender,
                          data = subset(Obs.Data, income == 0),
                          method = "full", distance = "glm", link = "logit")
data.income.0 <- match.data(match.income.0)
```

Below, we draw a sample from the high-income group based on the hypothetical example of an association between hypertension and smoking. Here age [centered], gender, education, and diet are covariates.

```
            age gender education diet income smoking hypertension  distance
657   6.0810120      0         1    1      0       1            1 0.9999874
4932  1.6109860      1         1    0      0       1            0 0.9943155
252  -0.2475055      1         1    1      0       0            1 0.8525107
2693 -0.2511048      1         1    0      0       1            1 0.8516785
1646 -0.2836155      1         0    1      0       1            1 0.8439843
        weights subclass
657  1.00000000       33
4932 1.00000000       33
252  0.04944134       23
2693 1.00000000       23
1646 1.00000000        4
```

Now, we do the same for the low-income group (`income == 1`):

```
match.income.1 <- matchit(smoking ~ age + gender,
                          data = subset(Obs.Data, income == 1),
                          method = "full", distance = "glm", link = "logit")
data.income.1 <- match.data(match.income.1)
```

We estimated the exposure effect from a weighted outcome model for the matched data. While the `weights` are essential for estimating the point estimate from the outcome model, the `subclass` variable assists in calculating the robust variance of the exposure effect estimate.

```
# Treatment effect estimation
fit.income.0 <- glm(hypertension ~ smoking + age + gender,
                    data = data.income.0, weights = weights,
                    family = quasibinomial("logit"))
fit.income.1 <- glm(hypertension ~ smoking + age + gender,
                    data = data.income.1, weights = weights,
                    family = quasibinomial("logit"))
```

```r
# Robust variance calculation
fit.nexp.adj.res1 <- summ(fit.income.1,
                          robust = TRUE,
                          cluster = "subclass",
                          confint = TRUE)
fit.nexp.adj.res0 <- summ(fit.income.0,
                          robust = TRUE,
                          cluster = "subclass",
                          confint = TRUE)
```

Table 4.3: Subgroup-specific treatment effect estimates (expressed in log-OR) from the hypothetical example using the stratified approach.

Value of income	Est.	2.5%	97.5%	t val.	p
0	3.76	-43.15	50.66	0.16	0.88
1	1.36	0.90	1.81	5.78	0.00

4.3.2 Joint approach without exact matching within subgroups

Here, entire cohort data is used to estimate the propensity scores, and the effect modifier `income` is considered as a covariate in the propensity score model:

```r
ps.formula <- as.formula("smoking ~ age + gender + income")
match.obj.j <- matchit(ps.formula, data = Obs.Data,
                       method = "full",
                       distance = "glm",
                       link = "logit")
match.data.j <- match.data(match.obj.j)
```

```r
fit.joint.no.exact <- glm(hypertension ~ smoking*income + age + gender,
                          data = match.data.j,
                          weights = weights,
                          family = binomial("logit"))

nem.nexp.adj.res <- interactions::sim_slopes(fit.joint.no.exact,
                                             pred = smoking,
                                             modx = income,
                                             robust = "HC1",
                                             cluster = "subclass",
```

```
                                                johnson_neyman = TRUE,
                                                confint = TRUE,
                                                data = match.data.j)
```

Table 4.4: Subgroup-specific treatment effect estimates (expressed in log-OR) from the hypothetical example using the joint approach.

Value of income	Est.	S.E.	2.5%	97.5%	z val.	p
0	3.77	1.02	1.77	5.76	3.70	0
1	1.34	0.28	0.79	1.89	4.78	0

4.3.3 Joint approach with exact matching within subgroups

We specify the moderator variable's name in the `exact` argument of the `matchit` function.

```
ps.formula.no.mod <- as.formula("smoking ~ age + gender")
match.obj.js <- matchit(ps.formula.no.mod, data = Obs.Data,
                        method = "full", distance = "glm",link = "logit",
                        exact = "income")
match.data.js <- match.data(match.obj.js)
fit.joint.exact <- glm(hypertension ~ smoking*income + age + gender,
                       data = match.data.js, weights = weights,
                       family = binomial("logit"))
js.nexp.adj.res <- interactions::sim_slopes(fit.joint.exact,
                                            pred = smoking,
                                            modx = income, # effect modifier
                                            robust = "HC1",
                                            cluster = "subclass",
                                            johnson_neyman = FALSE,
                                            confint = TRUE,
                                            data = match.data.js)
```

Table 4.5: Subgroup-specific exposure effect estimates (expressed in log-OR) from the hypothetical example using the Joint model, separate matching approach.

Value of income	Est.	S.E.	2.5%	97.5%	z val.	p
0	3.81	1.02	1.81	5.80	3.74	0
1	1.34	0.28	0.79	1.89	4.76	0

4.3.4 Interaction approach without exact matching within subgroups

Analysts incorporate relevant moderator-covariate interactions into the propensity score model that align with biological plausibility. For instance, in the case study we considered an interaction between age (a covariate) and income (a moderator), but did not include other interactions terms.

```
ps.formula.with.int <- formula("smoking ~ age*income + gender")
match.obj.i <- matchit(ps.formula.with.int, data = Obs.Data,
                       method = "full", distance = "glm",link = "logit")
match.data.i <- match.data(match.obj.i)
fit.int.no.exact <- glm(hypertension ~ smoking*income + age + gender,
                        data = match.data.i, weights = weights,
                        family = binomial("logit"))
i.nexp.adj.res <- interactions::sim_slopes(fit.int.no.exact,
                                           pred = smoking,
                                           modx = income,
                                           robust = "HC1",
                                           cluster = "subclass",
                                           johnson_neyman = FALSE,
                                           confint = TRUE,
                                           data = match.data.i)
```

Table 4.6: Subgroup-specific exposure effect estimates (expressed in log-OR) from the hypothetical example using the interaction approach.

Value of income	Est.	S.E.	2.5%	97.5%	z val.	p
0	3.77	1.02	1.77	5.77	3.70	0
1	1.35	0.28	0.80	1.91	4.77	0

4.3.5 Interaction approach with exact matching within subgroups

This method bears resemblance to the interaction approach for propensity score estimation. However, when it comes to matching, researchers match within each moderator subgroup.

```
match.obj.is <- matchit(ps.formula.with.int, data = Obs.Data,
                        method = "full", distance = "glm",link = "logit",
                        exact = "income")
match.data.is <- match.data(match.obj.is)
fit.int.exact <- glm(hypertension ~ smoking*income + age + gender,
```

```
                    data = match.data.is, weights = weights,
                    family = binomial("logit"))
is.nexp.adj.res <- interactions::sim_slopes(fit.int.exact,
                                            pred = smoking,
                                            modx = income,
                                            robust = "HC1",
                                            cluster = "subclass",
                                            johnson_neyman = FALSE,
                                            confint = TRUE,
                                            data = match.data.is)
```

Table 4.7: Subgroup-specific exposure effect estimates (expressed in log-OR) from the hypothetical example using the interaction model, separate matching approach.

Value of income	Est.	S.E.	2.5%	97.5%	z val.	p
0	3.75	1.02	1.75	5.75	3.68	0
1	1.37	0.28	0.82	1.92	4.88	0

4.4 Propensity Score Weighting

4.4.1 Common model

This approach adds confounder-moderator interactions in the common weight model.

```
require(WeightIt)
library(survey)
library(interactions)
W.out <- weightit(ps.formula.with.int,
                  data = Obs.Data,
                  method = "ps",
                  estimand = "ATT")
d.w <- svydesign(~1, weights = W.out$weights, data = Obs.Data)
fit2w <- svyglm(hypertension ~ smoking*income, design = d.w,
                family = quasibinomial("logit"))
w.nexp.adj.res <- interactions::sim_slopes(fit2w,
                                           pred = smoking,
                                           modx = income,
                                           confint = TRUE)
```

Table 4.8: Subgroup-specific exposure effect estimates (expressed in log-OR) from the hypothetical example using the weighting approach.

Value of income	Est.	S.E.	2.5%	97.5%	t val.	p
0	2.66	0.63	1.42	3.89	4.23	0
1	1.32	0.25	0.83	1.82	5.24	0

We can adjust previous analysis model to adopt stabilized weights for the propensity score (`stabilize = TRUE`):

```
W.out.st <- weightit(ps.formula.with.int, data = Obs.Data,
                     method = "ps",
                     estimand = "ATT",
                     stabilize = TRUE)
d.sw <- svydesign(~1, weights = W.out.st$weights, data = Obs.Data)
fit2sw <- svyglm(hypertension ~ smoking*income + age + gender,
                 design = d.sw,
                 family = binomial("logit"))
ws.nexp.adj.res <- interactions::sim_slopes(fit2sw,
                                            pred = smoking,
                                            modx = income,
                                            confint = TRUE)
```

Table 4.9: Subgroup-specific exposure effect estimates (expressed in log-OR) from the hypothetical example using stabilized propensity score weights.

Value of income	Est.	S.E.	2.5%	97.5%	t val.	p
0	2.27	0.73	0.84	3.69	3.12	0
1	1.32	0.25	0.83	1.82	5.23	0

4.4.2 Separate models

Propensity score weighting approach with weights estimated separately from each subgroup:

```
ps.formula.with.no.int <- formula("smoking ~ age + gender")
W.out1 <- weightit(ps.formula.with.int,
                   data = subset(Obs.Data, income == 1),
                   method = "ps",
                   estimand = "ATT")
```

```r
trimmed.weight.1.percent1 <- trim(W.out1$weights,
                                  at = 1, lower = TRUE)
```

Table 4.10: Weight summaries before and after truncation.

Weight	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Raw weights	0	0.01	0.11	0.45	1	11.69
1% truncated weights	0	0.01	0.11	0.44	1	7.61

```r
# Outcome model for income = 1
d.w1 <- svydesign(~1, weights = trimmed.weight.1.percent1,
                  data = subset(Obs.Data, income == 1))

fit2unadj1 <- svyglm(hypertension ~ smoking,
                     design = d.w1,
                     family = binomial("logit"))

# weight model for income = 0
W.out0 <- weightit(ps.formula, data = subset(Obs.Data, income == 0),
                   method = "ps", estimand = "ATT")
trimmed.weight.1.percent0 <- trim(W.out0$weights, at = 1, lower = TRUE)

# Outcome model for income = 0
d.w0 <- svydesign(~1, weights = trimmed.weight.1.percent0,
                  data = subset(Obs.Data, income == 0))

fit2unadj0 <- svyglm(hypertension ~ smoking,
                     design = d.w0,
                     family = binomial("logit"))

fit.exp.adj.res1 <- summ(fit2unadj1, confint = TRUE)
fit.exp.adj.res0 <- summ(fit2unadj0, confint = TRUE)
```

Table 4.11: Subgroup-specific exposure effect estimates (expressed in log-OR) from the hypothetical example using the propensity score weighting approach (Separate weight models).

Value of income	Est.	2.5%	97.5%	t val.	p
0	2.21	1.27	3.15	4.60	0
1	1.34	0.85	1.83	5.36	0

4.4.3 Weights from the subgroup balancing propensity scores

Subgroup balancing propensity scores for propensity score weighting:

```
w.out <- weightit(smoking ~ age + gender + income,
                  data = Obs.Data,
                  method = "ps",
                  estimand = "ATT")

w.out.sb <- sbps(w.out, moderator = "income")
d.w.sb <- svydesign(~1, weights = w.out.sb$weights, data = Obs.Data)

fit2unadj.sb <- svyglm(hypertension ~ smoking*income,
                       design = d.w.sb,
                       family = binomial("logit"))

sb.w.nexp.adj.res <- interactions::sim_slopes(fit2unadj.sb,
                                              pred = smoking,
                                              modx = income,
                                              confint = TRUE,
                                              johnson_neyman = FALSE)
```

Table 4.12: Subgroup-specific exposure effect estimates (expressed in log-OR) from the hypothetical example using the subgroup balancing weighting approach.

Value of income	Est.	S.E.	2.5%	97.5%	t val.	p
0	2.68	0.64	1.44	3.92	4.22	0
1	1.32	0.25	0.82	1.82	5.22	0

4.5 Covariate adjustment for the propensity score

4.5.1 As continuous covariate

An implementation of propensity scores as a continuous covariate in the outcome model:

```
# Separate models for each subgroup

# For subgroup income = 1
Obs.Data$ps[Obs.Data$income == 1] <- glm(ps.formula,
```

```
                                          data = subset(Obs.Data, income == 1),
                                          family = "binomial")$fitted.values

fit2adj1 <- glm(hypertension ~ smoking + age + gender,
                family = binomial("logit"),
                data = subset(Obs.Data, income == 1))

# For subgroup income = 0
Obs.Data$ps[Obs.Data$income == 0] <- glm(ps.formula,
                                          data = subset(Obs.Data, income == 0),
                                          family = "binomial")$fitted.values

fit2adj0 <- glm(hypertension ~ smoking + age + gender,
                family = binomial("logit"),
                data = subset(Obs.Data, income == 0))

fit.nexp.adj.res1 <- summ(fit2adj1, robust = TRUE, confint = TRUE)
fit.nexp.adj.res0 <- summ(fit2adj0, robust = TRUE, confint = TRUE)
```

Table 4.13: Subgroup-specific exposure effect estimates (expressed in log-OR) from the hypothetical example using Propensity Score as a covariate adjustment approach (considering separate models for each subgroup).

Value of income	Est.	2.5%	97.5%	z val.	p
0	1.16	0.56	1.75	3.83	0
1	1.37	0.96	1.77	6.61	0

```
# Common model
Obs.Data$ps <- glm(ps.formula.with.int, data = Obs.Data,
                   family = "binomial")$fitted.values
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
fit2adjc <- glm(hypertension ~ smoking*income + age + gender + ps,
                family = binomial("logit"),
                data = Obs.Data)

c.nexp.adj.res <- interactions::sim_slopes(fit2adjc,
                                           pred = smoking,
```

```
                                        modx = income,
                                        confint = TRUE,
                                        data = Obs.Data)
```

Table 4.14: Subgroup-specific exposure effect estimates (expressed in log-OR) from the hypothetical example using Propensity Score as a covariate adjustment approach (considering a common model).

Value of income	Est.	S.E.	2.5%	97.5%	z val.	p
0	1.17	0.29	0.61	1.74	4.07	0
1	1.43	0.23	0.98	1.87	6.30	0

4.5.2 As quantiles

The propensity scores as a categorical covariate, broken by quintiles, in the outcome model.

```
Obs.Data$ps <- glm(ps.formula.with.int,
                   data = Obs.Data,
                   family = "binomial")$fitted.values
quintiles <- quantile(Obs.Data$ps,
                      prob = seq(from = 0, to = 1, by = 0.2),
                      na.rm = T)
Obs.Data$psq <- cut(Obs.Data$ps, breaks = quintiles,
                    labels = seq(1,5), include.lowest = T)
Obs.Data$psq <- as.factor(Obs.Data$psq)

fit2adjq <- glm(hypertension ~ (smoking*psq)*income,
                family = binomial("logit"),
                data = Obs.Data)

cq.nexp.adj.res <- interactions::sim_slopes(fit2adjq,
                                            pred = smoking,
                                            modx = income,
                                            confint = TRUE,
                                            data = Obs.Data)
```

Table 4.15: Subgroup-specific exposure effect estimates (expressed in log-OR) from the hypothetical example using Propensity Score as a covariate adjustment approach (as quintiles).

Value of income	Est.	S.E.	2.5%	97.5%	z val.	p
0	3.08	0.63	1.85	4.32	4.91	0
1	2.60	0.47	1.68	3.51	5.56	0

4.6 Propensity Score Stratification

Here is an implementation of propensity score stratification approach by using the marginal mean weighting through stratification (MMWS):

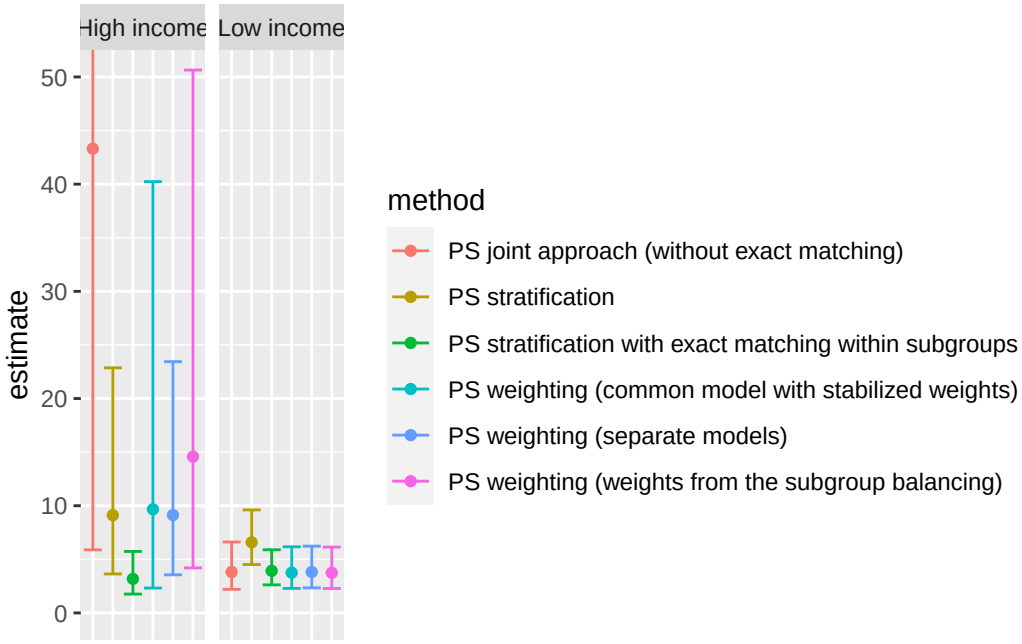
```
match.obj <- matchit(ps.formula, data = Obs.Data,
                     method = "subclass", subclass = 3,
                     estimand = "ATT", min.n = 10)

data.subclass <- match.data(match.obj)

subclass.fit <- glm(hypertension ~ smoking*income,
                    family = binomial("logit"),
                    data = data.subclass,
                    weights = weights)

subclass.nexp.adj.res <- interactions::sim_slopes(subclass.fit,
                                                  pred = smoking,
                                                  modx = income,
                                                  confint = TRUE,
                                                  robust = "HC3",
                                                  johnson_neyman = FALSE,
                                                  data = data.subclass)
```

Table 4.16: Subgroup-specific exposure effect estimates (expressed in log-OR) from the hypothetical example using propensity score stratification approach.

Value of income	Est.	S.E.	2.5%	97.5%	z val.	p
0	2.21	0.47	1.29	3.13	4.71	0
1	1.89	0.19	1.51	2.26	9.78	0

4.7 Summary

The marginal odds ratios for `smoking` are summarized below



Version info

This chapter was rendered using the following version of R and its packages:

```
R version 4.2.3 (2023-03-15 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19045)

Matrix products: default

locale:
[1] LC_COLLATE=Dutch_Netherlands.utf8  LC_CTYPE=Dutch_Netherlands.utf8
[3] LC_MONETARY=Dutch_Netherlands.utf8 LC_NUMERIC=C
[5] LC_TIME=Dutch_Netherlands.utf8

attached base packages:
[1] grid      stats     graphics  grDevices utils     datasets  methods
```

```
[8] base

other attached packages:
 [1] interactions_1.1.5 survey_4.2-1       survival_3.5-5     Matrix_1.5-4.1
 [5] interactionR_0.1.6 simcausal_0.5.6    scales_1.2.1       ggplot2_3.4.4
 [9] xtable_1.8-4       dplyr_1.1.2        kableExtra_1.3.4   knitr_1.45
[13] cowplot_1.1.1      broom_1.0.5        MatchIt_4.5.5      jtools_2.2.2
[17] sandwich_3.0-2     lmtest_0.9-40      zoo_1.8-12         optmatch_0.10.7
[21] WeightIt_0.14.2    cobalt_4.5.2       table1_1.4.3

loaded via a namespace (and not attached):
 [1] fontquiver_0.2.1        webshot_0.5.5           httr_1.4.7
 [4] tools_4.2.3             backports_1.4.1         utf8_1.2.3
 [7] R6_2.5.1                DBI_1.1.3               colorspace_2.1-0
[10] withr_2.5.2             tidyselect_1.2.0        curl_5.1.0
[13] compiler_4.2.3          textshaping_0.3.7       cli_3.6.1
[16] rvest_1.0.3             expm_0.999-7            flextable_0.9.4
[19] xml2_1.3.4              officer_0.6.3           fontBitstreamVera_0.1.1
[22] labeling_0.4.3          mvtnorm_1.2-3           askpass_1.2.0
[25] systemfonts_1.0.4       stringr_1.5.1           digest_0.6.31
[28] rmarkdown_2.25          svglite_2.1.1           gfonts_0.2.0
[31] pkgconfig_2.0.3         htmltools_0.5.5         fastmap_1.1.1
[34] rlang_1.1.1             rstudioapi_0.15.0       httpcode_0.3.0
[37] shiny_1.8.0             farver_2.1.1            generics_0.1.3
[40] jsonlite_1.8.7          car_3.1-2               zip_2.3.0
[43] magrittr_2.0.3          Formula_1.2-5           Rcpp_1.0.10
[46] munsell_0.5.0           fansi_1.0.4             abind_1.4-5
[49] gdtools_0.3.4           lifecycle_1.0.4         chk_0.9.1
[52] stringi_1.7.12          yaml_2.3.7              carData_3.0-5
[55] promises_1.2.1          crayon_1.5.2            lattice_0.21-8
[58] splines_4.2.3           pander_0.6.5            pillar_1.9.0
[61] uuid_1.1-1              igraph_1.5.1            codetools_0.2-19
[64] crul_1.4.0              glue_1.6.2              evaluate_0.23
[67] msm_1.7                 mitools_2.4             fontLiberation_0.1.0
[70] data.table_1.14.8       vctrs_0.6.3             httpuv_1.6.12
[73] gtable_0.3.4            openssl_2.1.1           purrr_1.0.1
[76] tidyr_1.3.0             assertthat_0.2.1        xfun_0.39
[79] mime_0.12               later_1.3.1             ragg_1.2.6
[82] viridisLite_0.4.2       tibble_3.2.1            ellipsis_0.3.2
[85] rlemon_0.2.1
```

References

5 Dealing with missing data

Johanna Munoz (Julius Center for Health Sciences and Primary Care)
Thomas Debray (Smart Data Analysis and Statistics B.V.)

5.1 Main Analysis

The main objective of this analysis is to assess whether the number of episodes (y) occurring within specific time periods (years) differs between the treatment groups (1: DMF and 0: TERI). To address potential confounding factors, the researchers consider variables such as patient age, the log of premedical cost (`logPremedicalcost`), previous DMT efficacy (`prevDMTefficacy`), and the number of episodes in previous relapses (prerelapseNum).

When estimating treatment effects from observational data, an assumption is made that the patient populations in both treatment groups are as similar as possible. Various methods for balancing data across treatment groups are proposed, including matching, inverse propensity weighting, stratification, and regression adjustment.

In this case, the focus is specifically on the matching method, which offers advantages over regression adjustment by potentially alleviating issues related to model mis-specification. This includes addressing non-linear relationships between certain confounders and the outcome variable and accounting for treatment effects that may depend on specific confounders (treatment-confounder interaction terms). Propensity scores are used to match subjects in the treatment groups.

Moreover, intentionally introducing incomplete covariate variables in this example adds complexity to the propensity score estimation. Depending on the propensity score estimation technique employed, it may be necessary to incorporate an imputation step. For instance, logistic regression estimation requires complete data for all observations, while XGBoost is robust to missing data [**?**].

To estimate marginal treatment effects, the g-computation method is employed [**?**]. This method involves specifying a model for the outcome dependent on the treatment and covariates. The potential outcomes, i.e., the predicted values of the outcome on treatment (y_i^1) and control

(y_i^0) for each sample unit i, are estimated. The marginal treatment effect is then calculated by contrasting the averaged estimated potential outcomes.

In this example, we consider the estimation of comparative treatment effects in the absence of treatment-effect heterogeneity.

5.2 Estimation workflow

The proposed workflow consists of the following steps:

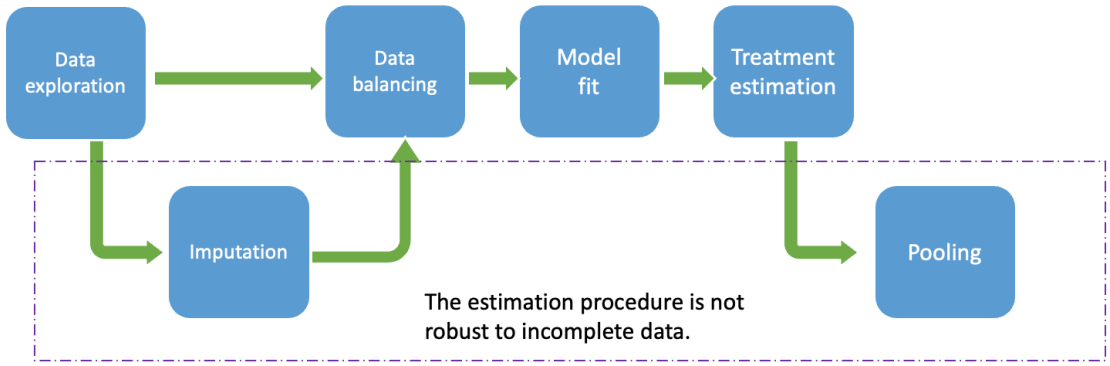


Figure 5.1: Estimation Workflow

1. **Data Exploration:** In this step, we examine the observed data to comprehend the variables within the dataset. Our primary focus lies on identifying missing patterns and relationships among observed variables, including missing indicator variables and others. This exploration aids in discerning the most plausible missing mechanisms and suitable imputation techniques. Additionally, field experts' insights may be incorporated to enhance understanding of the missing process, potentially considering MNAR assumptions.
2. **Imputation:** It is essential to evaluate whether the imputation procedure is necessary or if simpler methods, such as complete case analysis, are more suitable. In case imputation procedures are required, selecting plausible imputation methods that align with the main model analysis is crucial. This involves choosing individual imputation methods for each incomplete variable, determining the predictor variables on the imputation model. Pre_imputation (where imputation values can be deterministically derived from other variables) and Post-imputation (e.g.ensuring imputed values fall within a reasonable range) steps may also considered.
3. **Data Balancing:** Several methods, including PS matching or inverse weighting propensity score, can be utilized. It is required to evaluate the balance, which could be done

via visual inspection.(eg.cobalt package). In this example, we estimate propensity scores using logistic regression. For most balancing procedures in R, counterparts specifically designed for imputed datasets are available, such as those in the matchthem R package, which includes PS matching and IPW as done in the matchit R package.

4. **Model Fit:** : It is fit a model to predict the outcomes for each sample unit under each possible treatment value (DMF and TERI), as predictors include the treatment and optionally the baseline covariates and also the propensity score.
5. **Treatment Estimation & Pooling:** For simplicity in this tutorial, we will use the comparison functions from the R **matchingmethods** package [**?**], which can be used for completed data and also from outputs from the imputation process. In the last case, internally the functions calculate the treatment effects on each imputed dataset and pool the estimates using Rubin's Rules.

Let's start by preparing the R environment. All the functions used in this tutorial can be found in the resource file `functions.r`.

```
# Load the required packages and additional functions
source("resources/chapter 09/functions.r")
```

5.3 Homogeneous Treatment Effect

In this example, we focus on estimating comparative treatment effects in the absence of heterogeneous treatment effects (HTE).

5.3.1 Generating an Observational Dataset

We can simulate an observational dataset of $N = 3000$ patients as follows:

```
data_hom <- generate_data(n = 3000, seed = 1234)
```

The **generate_data()** function allows the specification of various treatment effect options, easily adjustable by modifying the beta parameter. In this instance, we assume a consistent treatment effect across the entire target population. This dataset currently contains no missing values.

The simulated dataset comprises two treatment groups with variations in baseline characteristics. For example, the figure below illustrates baseline imbalances in covariates such as age.

We can calculate the treatment effect on the complete observed dataset. To do this, we start by balancing the dataset using Propensity Score matching. In this case, the propensity score

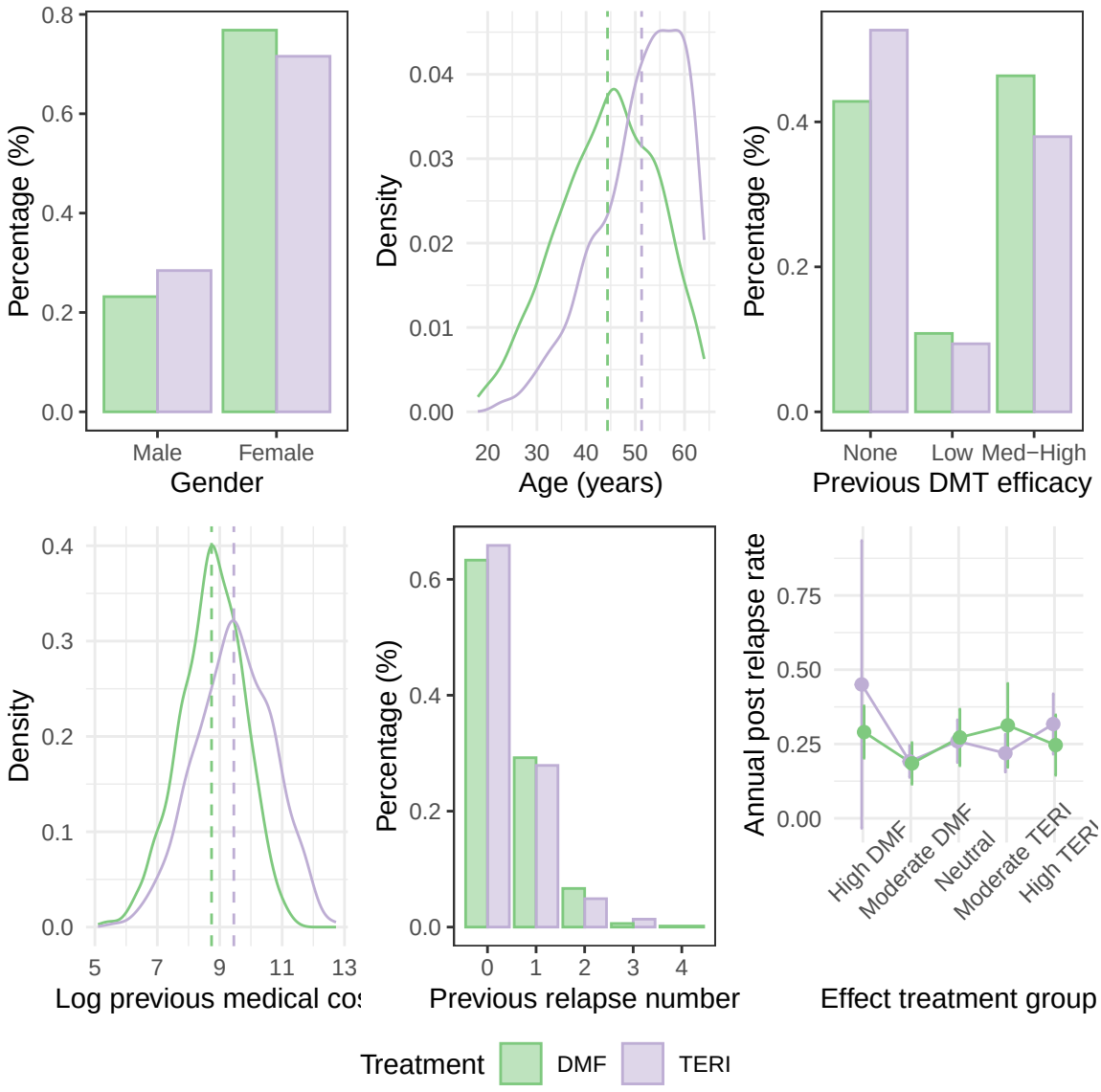


Figure 5.2: Figure 1:Distribution of confounders and outcome variable