

NHISS Example



# Contents

<b>1</b>	<b>Baseline characteristic table</b>	<b>5</b>
1.1	Baseline characteristic table . . . . .	6
1.2	Baseline characteristic table_total . . . . .	7
<b>2</b>	<b>Multiple imputation</b>	<b>9</b>
2.1	The number of missing values . . . . .	9
2.2	Imputation for missing values . . . . .	10
<b>3</b>	<b>Propensity Score Matching</b>	<b>13</b>
3.1	Complete data version . . . . .	14
3.2	Balance check (Complete) . . . . .	15
3.3	Missing data version . . . . .	16
3.4	Balance check (Missing) . . . . .	17
<b>4</b>	<b>Propensity score weighting</b>	<b>19</b>
4.1	Complete data version . . . . .	19
4.2	Balance check (Complete) . . . . .	21
4.3	Missing data version . . . . .	21
4.4	Balance check (Missing) . . . . .	23
<b>5</b>	<b>Footnotes and citations</b>	<b>25</b>
5.1	Footnotes . . . . .	25
5.2	Citations . . . . .	25

<b>6</b>	<b>Blocks</b>	<b>27</b>
6.1	Equations . . . . .	27
6.2	Theorems and proofs . . . . .	27
6.3	Callout blocks . . . . .	27
<b>7</b>	<b>Sharing your book</b>	<b>29</b>
7.1	Publishing . . . . .	29
7.2	404 pages . . . . .	29
7.3	Metadata for sharing . . . . .	29

# Chapter 1

## Baseline characteristic table

Baseline tables show the characteristics of research subjects included in a study. A table characterizing baseline characteristics is so important that it's typically the first table that appears in any observational epidemiology (or clinical trial) manuscript, so it's commonly referred to as a "Table 1". The "Table 1" contain information about the mean and standard deviation(or median and IQR) for continue/scale variable, and proportion for categorical variable.

Baseline characteristic table should be created before imputaion, matching, or weighting.

---

Using data **final\_db**  
Outcome variable : **HTN**  
Follow-up period : **DATEDIFF**  
Exposure variable : **DM**  
Covariates : **Age, Sex, SES, Region, BMI, CCI, Comorbidities(Dyslipidemia, Ischemic heart disease)**

---

```
## load library
library(moonBook)
library(dplyr)
```

```
## load data
final_db <- read.csv('Data/final_db.csv', header=T)
```

```
## formula
formula.bc <- formula(DM ~ HTN + DATEDIFF + AGE + SEX + SES + REGION + BMI + CCI + DYS + IHD)
```

- Use **mytable()** function in **moonBook** package to create baseline characteristic tables.
  - method=1 : forces analysis as normal-distributed
  - method=3 : performs a Shapiro-Wilk test to decide between normal or non-normal

### 1.1 Baseline characteristic table

```
mytable(formula.bc, data=final_db, method=3)
```

Descriptive Statistics by 'DM'				
	0	1	p	
	(N=2356)	(N=118)		
HTN			0.000	
- 0	2215 (94.0%)	69 (58.5%)		
- 1	141 ( 6.0%)	49 (41.5%)		
DATEDIFF	1685.0 [835.5;2460.5]	963.5 [324.0;1690.0]	0.000	
AGE	36.0 [22.0;48.0]	58.0 [50.0;68.0]	0.000	
SEX			0.903	
- 1	1182 (50.2%)	58 (49.2%)		
- 2	1174 (49.8%)	60 (50.8%)		
SES			0.393	
- 1	668 (29.6%)	29 (25.2%)		
- 2	709 (31.4%)	34 (29.6%)		
- 3	883 (39.1%)	52 (45.2%)		
REGION			0.996	
- 1	1160 (49.5%)	58 (49.2%)		
- 2	489 (20.9%)	25 (21.2%)		
- 3	694 (29.6%)	35 (29.7%)		
BMI	23.1 [21.0;25.2]	24.3 [22.6;26.1]	0.013	
CCI			0.001	
- 0	1810 (76.8%)	80 (67.8%)		
- 1	433 (18.4%)	23 (19.5%)		
- 2	113 ( 4.8%)	15 (12.7%)		
DYS			0.000	
- 0	2285 (97.0%)	100 (84.7%)		
- 1	71 ( 3.0%)	18 (15.3%)		
IHD			0.476	

```
##      - 0          2340 (99.3%)          116 (98.3%)
##      - 1           16 ( 0.7%)           2 ( 1.7%)
## -----
```

## 1.2 Baseline characteristic table\_total

```
tot1 <- final_db %>% mutate(tmp=1)
tot2 <- final_db %>% mutate(tmp=2)
tot3 <- rbind(tot1,tot2)
```

```
mytable(tmp ~ HTN + DATEDIFF + AGE + SEX + SES + REGION + BMI + CCI + DYS + IHD, data=tot3, method="glm")
```

```
##
##      Descriptive Statistics by 'tmp'
## -----
##              1              2              p
##              (N=2474)      (N=2474)
## -----
## HTN                                     1.000
##   - 0          2284 (92.3%)          2284 (92.3%)
##   - 1           190 ( 7.7%)           190 ( 7.7%)
## DATEDIFF 1656.0 [811.0;2458.0] 1656.0 [811.0;2458.0] 1.000
## AGE       36.0 [22.0;50.0]      36.0 [22.0;50.0] 1.000
## SEX
##   - 1          1240 (50.1%)          1240 (50.1%)
##   - 2          1234 (49.9%)          1234 (49.9%)
## SES
##   - 1           697 (29.3%)          697 (29.3%)
##   - 2           743 (31.3%)          743 (31.3%)
##   - 3           935 (39.4%)          935 (39.4%)
## REGION
##   - 1          1218 (49.5%)          1218 (49.5%)
##   - 2           514 (20.9%)          514 (20.9%)
##   - 3           729 (29.6%)          729 (29.6%)
## BMI       23.2 [21.0;25.3]      23.2 [21.0;25.3] 1.000
## CCI
##   - 0          1890 (76.4%)          1890 (76.4%)
##   - 1           456 (18.4%)          456 (18.4%)
##   - 2           128 ( 5.2%)          128 ( 5.2%)
## DYS
##   - 0          2385 (96.4%)          2385 (96.4%)
##   - 1           89 ( 3.6%)           89 ( 3.6%)
## IHD
##                                     1.000
```

##	- 0	2456 (99.3%)	2456 (99.3%)
##	- 1	18 ( 0.7%)	18 ( 0.7%)
##	-----		



## Chapter 2

# Multiple imputation

Multiple imputation is a general approach to the problem of missing data. It aims to allow for the uncertainty about the missing data by creating several different plausible imputed data sets and appropriately combining results obtained from each of them.

Multiple imputation using chained equations (MICE) were performed to generate 10 imputed datasets. For the imputation model, predictive mean matching was used for continuous data and logistic regression was used for binary data.

---

Using data **final\_db**  
Outcome variable : **HTN**  
Follow-up period : **DATEDIFF**  
Exposure variable : **DM**  
Covariates : **Age, Sex, SES, Region, BMI, CCI, Comorbidities(Dyslipidemia, Ischemic heart disease)**

---

```
## load library
library(mice)
library(dplyr)

## load data
final_db <- read.csv('Data/final_db.csv', header=T)
```

### 2.1 The number of missing values

```
na_count <- function(data){
  num.na <- colSums(is.na(data))
}
```

```

per.na <- paste0(round(colSums(is.na(data))/nrow(data) *100,2),"%")

return(data.frame(missing=paste0(num.na,"(",per.na,")"),row.names = names(num.na)))
}

na_count(final_db)

```

```

##                missing
## RN_INDI          0(0%)
## DM               0(0%)
## INDEX_DT         0(0%)
## HTN              0(0%)
## FU_DT            0(0%)
## AGE              0(0%)
## SEX              0(0%)
## SES             99(4%)
## REGION          13(0.53%)
## BMI            1565(63.26%)
## CCI              0(0%)
## DYS              0(0%)
## IHD              0(0%)
## DATEDIFF         0(0%)

```

- Use **mice()** function in **mice** package to deal with missing data.
  - m=10 refers to the number of imputed datasets. Five is the default value.
  - Extract imputed data sets using **complete()** function

## 2.2 Imputation for missing values

```

## Exclude subject ID, index date before imputation
dat_mice <- final_db %>% select(-RN_INDI, -INDEX_DT, -FU_DT)
dat_imp <- mice(dat_mice, m=10, seed=1)

```

```

##
## iter imp variable
## 1 1 SES REGION BMI
## 1 2 SES REGION BMI
## 1 3 SES REGION BMI
## 1 4 SES REGION BMI
## 1 5 SES REGION BMI

```

```

## 1 6 SES REGION BMI
## 1 7 SES REGION BMI
## 1 8 SES REGION BMI
## 1 9 SES REGION BMI
## 1 10 SES REGION BMI
## 2 1 SES REGION BMI
## 2 2 SES REGION BMI
## 2 3 SES REGION BMI
## 2 4 SES REGION BMI
## 2 5 SES REGION BMI
## 2 6 SES REGION BMI
## 2 7 SES REGION BMI
## 2 8 SES REGION BMI
## 2 9 SES REGION BMI
## 2 10 SES REGION BMI
## 3 1 SES REGION BMI
## 3 2 SES REGION BMI
## 3 3 SES REGION BMI
## 3 4 SES REGION BMI
## 3 5 SES REGION BMI
## 3 6 SES REGION BMI
## 3 7 SES REGION BMI
## 3 8 SES REGION BMI
## 3 9 SES REGION BMI
## 3 10 SES REGION BMI
## 4 1 SES REGION BMI
## 4 2 SES REGION BMI
## 4 3 SES REGION BMI
## 4 4 SES REGION BMI
## 4 5 SES REGION BMI
## 4 6 SES REGION BMI
## 4 7 SES REGION BMI
## 4 8 SES REGION BMI
## 4 9 SES REGION BMI
## 4 10 SES REGION BMI
## 5 1 SES REGION BMI
## 5 2 SES REGION BMI
## 5 3 SES REGION BMI
## 5 4 SES REGION BMI
## 5 5 SES REGION BMI
## 5 6 SES REGION BMI
## 5 7 SES REGION BMI
## 5 8 SES REGION BMI
## 5 9 SES REGION BMI
## 5 10 SES REGION BMI

```

```
## Create 10 imputed data
for (i in 1:dat_imp$m){
  z <- assign(paste0('dat_imp',i),complete(dat_imp,i))
  assign(paste0('dat_imp',i),cbind(z,final_db %>% select(RN_INDI)))
}

## list of 10 imputed data
dat_imp_list <- list(dat_imp1,dat_imp2,dat_imp3,dat_imp4,dat_imp5,dat_imp6,dat_imp7,dat_imp8,dat_imp9,dat_imp10)

## Save multiple imputation result
save(dat_imp,file="Data/dat_imp.RData")
## Save list for imputed data
save(dat_imp_list,file="Data/dat_imp_list.RData")
```

## Chapter 3

# Propensity Score Matching

### Covariate balance

Covariate balance is the degree to which the distribution of covariates is similar across levels of the treatment.

SMD(Standardized Mean Difference) is the most widely used statistic for the assessment of balance after PSM.

SMD for continuous variables :

$$SMD = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{(S_1^2 + S_2^2)/2}}$$

- $\bar{X}_1$  and  $\bar{X}_2$  are sample mean for the treated and control groups.
- $S_1^2$  and  $S_2^2$  are sample variance for the treated and control groups.

SMD for binary variables :

$$SMD = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{[\hat{p}_1(1 - \hat{p}_1) + \hat{p}_2(1 - \hat{p}_2)]/2}}$$

- $\hat{p}_1$  and  $\hat{p}_2$  are prevalence of binary variables in the treated and control groups.

If the SMD after matching is less than **0.1**, it is determined that the difference by the covariates between the two groups is negligible.

---

Using list **dat\_imp\_list**  
 Outcome variable : **HTN**  
 Follow-up period : **DATEDIFF**  
 Exposure variable : **DM**  
 Covariates : **Age, Sex, SES, Region, BMI, CCI, Comorbidities(Dyslipidemia, Ischemic heart**

---

```
## load library
library(MatchIt)
library(dplyr)
source("cobalt_3.9.0.R")
```

```
## load data
load("Data/dat_imp_list.RData")
final_com <- read.csv('Data/final_com.csv', header=T)
```

```
## Formula
formula.mat <- formula(DM ~ AGE + SEX + SES + REGION + BMI + CCI + DYS + IHD)
```

- Use **matchit()** function in **MatchIt** package to create treatment and control groups balanced on included covariates.
  - method='nearest' : nearest neighbor matching on the propensity score
  - ratio=k : the number of controls matched to each treated unit for k:1 matching
  - caliper : Units whose propensity score difference is larger than the caliper will not be paired, and some treated units may therefore not receive a match.

### 3.1 Complete data version

1:5 nearest matching \_\_ caliper: 0.4

```
## Optimal caliper
ps <- glm(formula.mat,data=final_com, family = 'binomial')
ps$pscore<- predict(ps, type='link')
0.2*sd(ps$pscore)
```

```
## [1] 0.2122638
```

```

set.seed(1)
mat <- matchit(formula.mat, method = 'nearest', data=final_com, ratio=5, caliper=0.4)
matdat <- match.data(mat) %>% select(-subclass) # R subclass

# adding an matching index as a subclass to matdat and store as dat_mat
# as.numeric(tmp[,1:ratio]), rep(c(1:nrow(tmp)),ratio+1)
tmp <- na.omit(mat$match.matrix)
matid <- data.frame(rowid=c(as.numeric(rownames(tmp)), as.numeric(tmp[,1:5])), subclass=rep(c(1:nrow(tmp)),ratio+1))
matid$RN_INDI <- final_com$RN_INDI[matid$row]
dat_mat <- matdat %>% left_join(matid %>% select(RN_INDI,subclass),by = 'RN_INDI') %>% filter(is.na(subclass)==F)

## Save matching data
save(dat_mat,file="Data/dat_mat.RData")

```

## 3.2 Balance check (Complete)

```

bal.ch <- function(before_data, after_data, group){

  group<-deparse(substitute(group))

  # SMD before matching
  bal_check_un <- bal.tab.data.frame(before_data[covariates], treat=before_data[,group], binary="un")
  un <- abs(bal_check_un$Balance$Diff.Un)

  # SMD after matching
  bal_check_adj <- bal.tab.data.frame(after_data[covariates], treat=after_data[,group], binary="adj")
  adj <- abs(bal_check_adj$Balance$Diff.Un)

  bal.res <- data.frame(un=round(un,3),adj=round(adj,3))
  rownames(bal.res) <- rownames(bal_check_un$Balance)
  return(bal.res)
}

covariates <- c("AGE","SEX","SES","REGION","BMI","CCI","DYS","IHD")

bal.ch(before_matching_data, after_matching_data, group variable)

bal.ch(final_com, dat_mat, DM)

##          un    adj

```

```
## AGE      0.997 0.033
## SEX_2    0.029 0.017
## SES      0.159 0.201
## REGION   0.087 0.014
## BMI      0.273 0.147
## CCI      0.202 0.216
## DYS      0.519 0.199
## IHD      0.036 0.130
```

### 3.3 Missing data version

1:3 nearest matching \_\_ caliper: 0.3, 0.35, 0.4

```
## Optimal caliper
opt.clp <- c()
for (i in 1:length(dat_imp_list)){
  ps <- glm(formula.mat,data=dat_imp_list[[i]], family = 'binomial')
  ps$pscore<- predict(ps, type='link')
  opt.clp <- c(opt.clp, 0.2*sd(ps$pscore))
}
opt.clp; mean(opt.clp)
```

```
## [1] 0.2899617 0.2709918 0.2828702 0.2778456 0.2794823 0.2710618 0.2745410
## [8] 0.2706783 0.2788889 0.2695220
```

```
## [1] 0.2765843
```

```
caliper <- c(0.3,0.35,0.4)

for (i in 1:length(dat_imp_list)){
  for (j in caliper){
    set.seed(1)
    mat <- matchit(formula.mat, method = 'nearest', data=dat_imp_list[[i]], ratio=3, caliper=caliper[j])
    matdat <- match.data(mat) %>% select(-subclass) # R subclass

    # adding an matching index as a subclass to matdat and store as dat_mati_j
    # as.numeric(tmp[,1:ratio]), rep(c(1:nrow(tmp)),ratio+1)
    tmp <- na.omit(mat$match.matrix)
    matid <- data.frame(rowid=c(as.numeric(rownames(tmp)), as.numeric(tmp[,1:3])), subclass=rep(1, nrow(tmp)))
    matid$RN_INDI <- dat_imp_list[[i]]$RN_INDI[matid$rowid]
    assign(paste0('dat_mat',i,"_",j), matdat %>% left_join(matid %>% select(RN_INDI,subclass)))
  }
}
```



```
## list of 10 matched data
dat_mat_list_0.3 <- list(dat_mat1_0.3,dat_mat2_0.3,dat_mat3_0.3,dat_mat4_0.3,dat_mat5_0.3,dat_mat6_0.3,dat_mat7_0.3,dat_mat8_0.3,dat_mat9_0.3,dat_mat10_0.3)
dat_mat_list_0.35 <- list(dat_mat1_0.35,dat_mat2_0.35,dat_mat3_0.35,dat_mat4_0.35,dat_mat5_0.35,dat_mat6_0.35,dat_mat7_0.35,dat_mat8_0.35,dat_mat9_0.35,dat_mat10_0.35)
dat_mat_list_0.4 <- list(dat_mat1_0.4,dat_mat2_0.4,dat_mat3_0.4,dat_mat4_0.4,dat_mat5_0.4,dat_mat6_0.4,dat_mat7_0.4,dat_mat8_0.4,dat_mat9_0.4,dat_mat10_0.4)

## Save list for matched data
save(dat_mat_list_0.3,file="Data/dat_mat_list_0.3.RData")
save(dat_mat_list_0.35,file="Data/dat_mat_list_0.35.RData")
save(dat_mat_list_0.4,file="Data/dat_mat_list_0.4.RData")
```

### 3.4 Balance check (Missing)

```
bal.ch <- function(dat_imp_list, dat_mat_list, group){

  group<-deparse(substitute(group))

  # SMD before matching
  bal_check_un <- dat_imp_list %>% lapply(function(x){
    bal.tab.data.frame(x[covariates],
                       treat=x[,group], binary="std", s.d.denom = "pooled"))
  un <- sapply(bal_check_un, function(x) (abs(x$Balance$Diff.Un)))
  rownames(un) <- rownames(bal_check_un[[1]]$Balance)

  # SMD after matching
  bal_check_adj <- dat_mat_list %>% lapply(function(x){
    bal.tab.data.frame(x[covariates],
                       treat=x[,group], binary="std", s.d.denom = "pooled"))
  adj <- sapply(bal_check_adj, function(x) (abs(x$Balance$Diff.Un)))
  rownames(adj) <- rownames(bal_check_adj[[1]]$Balance)

  bal.res <- list(un=apply(un, 1, summary), adj=apply(adj, 1, summary))
  return(data.frame(un=round(bal.res$un[6,],3),adj=round(bal.res$adj[6,],3)))
}

covariates <- c("AGE","SEX","SES","REGION","BMI","CCI","DYS","IHD")
```

```
bal.ch(before_matching_list, after_matching_list, group variable)
```

```
bal.ch(dat_imp_list, dat_mat_list_0.3, DM)
```

```
##          un    adj
```

```
## AGE      1.450 0.041
## SEX_2    0.020 0.057
## SES      0.148 0.064
## REGION   0.006 0.105
## BMI      0.423 0.079
## CCI      0.267 0.095
## DYS      0.435 0.112
## IHD      0.094 0.077
```

```
bal.ch(dat_imp_list, dat_mat_list_0.35, DM)
```

```
##          un   adj
## AGE      1.450 0.040
## SEX_2    0.020 0.076
## SES      0.148 0.073
## REGION   0.006 0.094
## BMI      0.423 0.084
## CCI      0.267 0.082
## DYS      0.435 0.100
## IHD      0.094 0.101
```

```
bal.ch(dat_imp_list, dat_mat_list_0.4, DM)
```

```
##          un   adj
## AGE      1.450 0.037
## SEX_2    0.020 0.074
## SES      0.148 0.065
## REGION   0.006 0.094
## BMI      0.423 0.062
## CCI      0.267 0.077
## DYS      0.435 0.100
## IHD      0.094 0.100
```

## Chapter 4

# Propensity score weighting

---

Using data **final\_com** (complete data version)  
Using object **dat\_imp** (missin data version, imputaion result)  
Outcome variable : **HTN**  
Follow-up period : **DATEDIFF**  
Exposure variable : **DM**  
Covariates : **Age, Sex, SES, Region, BMI, CCI, Comorbidities(Dyslipidemia, Ischemic heart disease)**

---

```
## load library
library(dplyr)
source("cobalt_3.9.0.R")
```

```
## load data
final_com <- read.csv('Data/final_com.csv', header=T)
load("Data/dat_imp.RData")
```

```
## Formula
formula.wt <- formula(DM ~ AGE + SEX + SES + REGION + BMI + CCI + DYS + IHD)
```

### 4.1 Complete data version

- s.weights : stabilized weights
- trim.weights : trimming weights

```
## Propensity score function
ps_wt <- function(data, ps.formula, group, trt, psweight){

  modellist <- glm(formula=ps.formula, data=data, family="binomial", x=TRUE, y=TRUE)
  cov = data.frame(modellist$x[, -1])
  p.score <- modelfitted.values
  weights <- (data[,group]==trt)/p.score + (data[,group]!=trt)/(1-p.score)
  pt <- sum(data[,group]==trt)/nrow(cov)
  s.weights <- pt*(data[,group]==trt)/p.score + (1-pt)*(data[,group]!=trt)/(1-p.score)
  trim.p.score <- ifelse(p.score<0.01, 0.01, p.score)
  trim.p.score <- ifelse(p.score>0.99, 0.99, p.score)
  trim.weights <- (data[,group]==trt)/trim.p.score + (data[,group]!=trt)/(1-trim.p.score)

  ballist <- bal.tab.data.frame(cov, treat=modellist$y,
                                weights=get(paste0(psweight)), s.d.denom = "pooled", b
                                continuous="std", disp.means=TRUE, disp.sds=TRUE)

  return(list(modellist = modellist, weights=weights, s.weights=s.weights, trim.weights=trim.weights))
}
```

```
## Propensity score for imputed data sets
ps1 <- ps_wt(final_com, formula.wt, "DM", "1", "s.weights")
```

## Assuming "weighting". If not, specify with an argument to method.

```
## Complete weighted data
dat_wt <- final_com
dat_wt$weights <- ps1$s.weights
head(dat_wt)
```

```
##   RN_INDI DM   INDEX_DT HTN      FU_DT AGE SEX SES REGION  BMI CCI DYS IHD
## 1 1011725  0 2006-03-27   0 2013-01-01  36  1  1     2 23.7  0  0  0
## 2 1042143  0 2006-11-04   0 2013-01-01  56  1  3     2 21.9  0  0  0
## 3 1049401  1 2006-01-19   0 2013-01-01  66  1  2     3 27.1  0  0  0
## 4 1049841  0 2008-07-04   0 2013-01-01  48  2  3     3 21.0  0  0  0
## 5 1050697  0 2010-03-31   0 2013-01-01  50  1  3     3 24.9  1  0  0
## 6 1063538  0 2010-12-14   0 2013-01-01  50  2  1     1 19.3  0  0  0
##   DATEDIFF  weights
## 1      2472 0.9530470
## 2      2250 1.0084775
## 3      2539 0.3767263
## 4      1642 0.9677230
## 5      1007 0.9785016
## 6       749 0.9695923
```

```
## Save propensity score
save(dat_wt, file="Data/dat_wt.RData")
```

## 4.2 Balance check (Complete)

```
## Banlance check function
bal_wt<- function(obj){
  un <- round(abs(obj$ballist$Balance$Diff.Un),3)
  adj <- round(abs(obj$ballist$Balance$Diff.Adj),3)
  bal_df <- data.frame(un=un,adj=adj)
  rownames(bal_df) <- rownames(obj$ballist$Balance)

  return(bal_df)
}
```

```
## Balance check across imputed datasets
bal_wt(ps1)
```

```
##          un  adj
## AGE      0.997 0.374
## SEX_2    0.029 0.134
## SES      0.159 0.230
## REGION   0.087 0.000
## BMI      0.273 0.172
## CCI      0.202 0.212
## DYS      0.519 0.051
## IHD      0.036 0.073
```

## 4.3 Missing data version

- s.weights : stabilized weights
- trim.weights : trimming weights

```
## Propensity score function
ps_impute <- function(datasets, ps.formula, group, trt, psweight){
  modellist <- vector("list", 10)
  ballist <- vector("list", 10)
  weights <- s.weights <- trim.weights <- vector("list", 10)
  for(i in 1:10){
```

```
## Complete weighted data function
complete.wdata <- function(object, data, weights) {

  lapply(seq_len(data$m), function(j, object, data, weights) {
    out <- mice::complete(data, j)
    modelvars <- weights
    for( v in modelvars)
      out[[v]] <- object[[v]][[j]]

    out}, object = object, data=data, weights=weights)
}
```

[illegible]

```
## Assuming "weighting". If not, specify with an argument to method.
```

```
## Complete weighted data
```

```
data_wt <- complete.wdata(ps1, dat_imp, "s.weights")
head(data_wt[[1]])
```

```
##   DM HTN AGE SEX SES REGION  BMI CCI DYS IHD DATEDIFF s.weights
## 1  0   0  20  1  2      3 19.4  1  0  0    1001 0.9552116
## 2  0   0  46  1  2      1 23.9  0  0  0    2542 0.9918582
## 3  0   0  16  1  2      3 25.0  0  0  0    2542 0.9567726
## 4  0   0  36  1  1      2 23.7  0  0  0    2472 0.9672481
## 5  0   1  56  2  3      1 21.0  0  0  0    2449 1.0219381
## 6  0   0  20  2  2      1 16.4  1  0  0    1095 0.9547622
```

```
## Save propensity score
```

```
save(data_wt, file="Data/data_wt.RData")
```

## 4.4 Balance check (Missing)

```
## Banlance check function
```

```
bal.comb <- function(obj){
  unadj <- sapply(obj$ballist, function(x) (abs(x$Balance$Diff.Un)))
  adj <- sapply(obj$ballist, function(x) (abs(x$Balance$Diff.Adj)))
  rownames(unadj) <- rownames(adj) <- rownames(obj$ballist[[1]]$Balance)

  bal.res <- list(un=round(apply(unadj, 1, summary),3), adj=round(apply(adj, 1, summary),3))
  return(data.frame(un=round(bal.res$un[6,],3),adj=round(bal.res$adj[6,],3)))
}
```

```
## Balance check across imputed datasets
```

```
bal.comb(ps1)
```

```
##           un    adj
## AGE      1.450 0.743
## SEX_2    0.020 0.119
## SES      0.148 0.131
## REGION   0.006 0.093
## BMI      0.423 0.239
## CCI      0.267 0.156
## DYS      0.435 0.069
## IHD      0.094 0.059
```





## Chapter 5

# Footnotes and citations

### 5.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one <sup>1</sup>.

### 5.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2023] (check out the last code chunk in `index.Rmd` to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file `book.bib`). Note that the `.bib` files need to be listed in the `index.Rmd` with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: <https://rstudio.github.io/visual-markdown-editing/#/citations>

---

<sup>1</sup>This is a footnote.



## Chapter 6

# Blocks

### 6.1 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (6.1)$$

You may refer to using `\@ref{eq:binom}`, like see Equation (6.1).

### 6.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref{thm:tri}`, for example, check out this smart theorem 6.1.

**Theorem 6.1.** *For a right triangle, if  $c$  denotes the length of the hypotenuse and  $a$  and  $b$  denote the lengths of the **other** two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here <https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html>.

### 6.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: <https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html>



## Chapter 7

# Sharing your book

### 7.1 Publishing

HTML books can be published online, see: <https://bookdown.org/yihui/bookdown/publishing.html>

### 7.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

### 7.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book—all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

<https://pkgs.rstudio.com/bookdown/reference/gitbook.html>

Or use:

```
?bookdown::gitbook
```

# Bibliography

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL <http://yihui.org/knitr/>. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2023. URL <https://CRAN.R-project.org/package=bookdown>. R package version 0.32.