STAT6230 Term Paper

Xue Ming Wang(G20580112)

2023-12-16

Contents

1.Dataset

```
library(MatchIt)
library(tableone)
library(cobalt)
library(MatchIt)
library(tidyverse)

file_path <- "/Users/vivianwang/Desktop/heart_data.csv"

heart<- read.csv(file_path)

# Identify rows with missing data
missing_rows <- apply(heart, 1, function(row) any(is.na(row)))
rows_with_missing <- which(missing_rows)

# Remove rows with missing data
heart <- heart[!missing_rows, ]

# Display the cleaned dataset
head(heart)
```

```
##   index id   age gender height weight ap_hi ap_lo cholesterol gluc smoke alco
## 1     0  0 18393      2    168     62   110    80           1    1     0    0
## 2     1  1 20228      1    156     85   140    90           3    1     0    0
## 3     2  2 18857      1    165     64   130    70           3    1     0    0
## 4     3  3 17623      2    169     82   150   100           1    1     0    0
## 5     4  4 17474      1    156     56   100    60           1    1     0    0
## 6     5  8 21914      1    151     67   120    80           2    2     0    0
##   active cardio
## 1      1      0
## 2      1      1
## 3      0      1
## 4      1      1
## 5      0      0
## 6      0      0
```

2. Cleaned dataset and EDA

```r
heart$age <- round(heart$age / 365)
```

```r
heart$active = as.factor(heart$active)
heart$cardio = as.numeric(as.character(heart$cardio))

str(heart)
```

```
## 'data.frame':    70000 obs. of  14 variables:
##  $ index      : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ id         : int  0 1 2 3 4 8 9 12 13 14 ...
##  $ age        : num  50 55 52 48 48 60 61 62 48 54 ...
##  $ gender     : int  2 1 1 2 1 1 1 2 1 1 ...
##  $ height     : int  168 156 165 169 156 151 157 178 158 164 ...
##  $ weight     : num  62 85 64 82 56 67 93 95 71 68 ...
##  $ ap_hi      : int  110 140 130 150 100 120 130 130 110 110 ...
##  $ ap_lo      : int  80 90 70 100 60 80 80 90 70 60 ...
##  $ cholesterol: int  1 3 3 1 1 2 3 3 1 1 ...
##  $ gluc       : int  1 1 1 1 1 2 1 3 1 1 ...
##  $ smoke      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ alco       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ active     : Factor w/ 2 levels "0","1": 2 2 1 2 1 1 2 2 2 1 ...
##  $ cardio     : num  0 1 1 1 0 0 0 1 0 0 ...
```

```r
summary(heart)
```

```
##      index             id             age           gender         height
##  Min.   :    0   Min.   :    0   Min.   :30.0   Min.   :1.00   Min.   : 55
##  1st Qu.:17500   1st Qu.:25007   1st Qu.:48.0   1st Qu.:1.00   1st Qu.:159
##  Median :35000   Median :50002   Median :54.0   Median :1.00   Median :165
##  Mean   :35000   Mean   :49972   Mean   :53.3   Mean   :1.35   Mean   :164
##  3rd Qu.:52499   3rd Qu.:74889   3rd Qu.:58.0   3rd Qu.:2.00   3rd Qu.:170
##  Max.   :69999   Max.   :99999   Max.   :65.0   Max.   :2.00   Max.   :250
##      weight          ap_hi           ap_lo        cholesterol        gluc
##  Min.   : 10.0   Min.   : -150   Min.   :  -70   Min.   :1.00   Min.   :1.00
##  1st Qu.: 65.0   1st Qu.:  120   1st Qu.:   80   1st Qu.:1.00   1st Qu.:1.00
##  Median : 72.0   Median :  120   Median :   80   Median :1.00   Median :1.00
##  Mean   : 74.2   Mean   :  129   Mean   :   97   Mean   :1.37   Mean   :1.23
##  3rd Qu.: 82.0   3rd Qu.:  140   3rd Qu.:   90   3rd Qu.:2.00   3rd Qu.:1.00
##  Max.   :200.0   Max.   :16020   Max.   :11000   Max.   :3.00   Max.   :3.00
##      smoke            alco        active        cardio
##  Min.   :0.000   Min.   :0.000   0:13739   Min.   :0.0
##  1st Qu.:0.000   1st Qu.:0.000   1:56261   1st Qu.:0.0
##  Median :0.000   Median :0.000             Median :0.0
##  Mean   :0.088   Mean   :0.054             Mean   :0.5
##  3rd Qu.:0.000   3rd Qu.:0.000             3rd Qu.:1.0
##  Max.   :1.000   Max.   :1.000             Max.   :1.0
```

3. Sampling - Treatment variable = active (0/1)

```r
data_0 = heart[heart$active == 0, ]
data_1 = heart[heart$active == 1, ]

a = 0.8


set.seed(200)
sample_0 = data_0[sample(nrow(data_0), size = floor(a * nrow(data_0))), ]
sample_1 = data_1[sample(nrow(data_1), size = floor(a * nrow(data_1))), ]

sample_data = rbind(sample_0, sample_1)
summary(sample_data$active)
```

```
##     0     1
## 10991 45008
```

4. Estimate Propensity Score

logistic regression

```r
library(caTools)

# Estimating logistic regression
log_mod <- glm(active ~ age + gender + height + weight + ap_hi + ap_lo + cholesterol + gluc + smoke + a

# Print coefficients
summary(log_mod)
```

```
##
## Call:
## glm(formula = active ~ age + gender + height + weight + ap_hi +
##     ap_lo + cholesterol + gluc + smoke + alco, family = "binomial",
##     data = sample_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.20e+00   2.56e-01    8.60  < 2e-16 ***
## age         -3.58e-03   1.61e-03   -2.23  0.02578 *
## gender       1.17e-02   2.69e-02    0.43  0.66401
## height      -2.77e-03   1.57e-03   -1.77  0.07638 .
## weight      -3.12e-03   7.79e-04   -4.00  6.2e-05 ***
## ap_hi        6.76e-05   9.64e-05    0.70  0.48338
## ap_lo        1.35e-04   7.16e-05    1.89  0.05935 .
## cholesterol  6.34e-02   1.81e-02    3.51  0.00045 ***
## gluc        -5.60e-02   2.06e-02   -2.71  0.00665 **
## smoke        1.91e-01   4.42e-02    4.33  1.5e-05 ***
## alco         3.01e-01   5.57e-02    5.41  6.3e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 55460  on 55998  degrees of freedom
## Residual deviance: 55341  on 55988  degrees of freedom
## AIC: 55363
##
## Number of Fisher Scoring iterations: 4
```

```
# Calculate overall propensity score using all variables
sample_data$propensity_score <- predict(log_mod, type = "response")
# Assuming log_mod is your logistic regression model
# Assuming sample_data contains your data

# Example Columns calculate propensity scores
columns_of_interest <- c("age", "gender", "height", "weight", "ap_hi", "ap_lo", "cholesterol", "gluc",
summary(sample_data$propensity_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.718   0.794   0.801   0.804   0.810   0.954
```

propensity_score summary

```
summary(sample_data$propensity_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.718   0.794   0.801   0.804   0.810   0.954
```

```
# Logistic regression before matching
lmod_before_matching <- glm(cardio ~ active + age + gender + height + weight + ap_hi + ap_lo + choleste
coefficients_before_matching <- summary(lmod_before_matching)$coefficients[2, ]
conf_interval_before_matching <- confint(lmod_before_matching, parm = 2, level = 0.95)
```

5. Nearest Neighbor Matching

5.1 Propensity Score Matching

Method

```
library(MatchIt)

Nearest <- matchit(active ~ age + gender + height + weight + ap_hi + ap_lo + cholesterol + gluc + smoke
Nearest
```

```
## A matchit object
##  - method: 1:1 nearest neighbor matching without replacement
##  - distance: Propensity score
```

```
##              - estimated with logistic regression
##  - number of obs.: 55999 (original), 21982 (matched)
##  - target estimand: ATT
##  - covariates: age, gender, height, weight, ap_hi, ap_lo, cholesterol, gluc, smoke, alco
```

```
matchsum <- summary(Nearest, standardize = TRUE)$sum.matched
matched_summary <- data.frame(round(matchsum[,1:3], 3))

matched_summary
```

Summary statistics of matched groups

```
##             Means.Treated Means.Control Std..Mean.Diff.
## distance            0.828         0.802           1.428
## age                51.364        53.477          -0.312
## gender              1.461         1.344           0.245
## height            162.702       164.464          -0.216
## weight             69.072        74.673          -0.390
## ap_hi             135.774       127.965           0.052
## ap_lo             135.871        93.932           0.210
## cholesterol         1.670         1.355           0.462
## gluc                1.147         1.238          -0.158
## smoke               0.361         0.073           0.995
## alco                0.233         0.040           0.832
```

```
Nearest_matched <- match.data(Nearest)
```

Propensity Score Matching Model

```
summary(Nearest)
```

```
##
## Call:
## matchit(formula = active ~ age + gender + height + weight + ap_hi +
##     ap_lo + cholesterol + gluc + smoke + alco, data = sample_data,
##     method = "nearest")
##
## Summary of Balance for All Data:
##             Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance            0.804         0.802           0.113      1.203     0.029
## age                53.305        53.477          -0.025      0.995     0.006
## gender              1.351         1.344           0.015      1.010     0.004
## height            164.325       164.464          -0.017      0.968     0.002
## weight             74.084        74.673          -0.041      0.990     0.007
## ap_hi             128.946       127.965           0.006      1.470     0.003
## ap_lo              97.809        93.932           0.019      1.227     0.003
## cholesterol         1.370         1.355           0.022      1.055     0.005
## gluc                1.227         1.238          -0.019      0.973     0.004
```

```
## smoke               0.092         0.073           0.067          .     0.019
## alco                0.057         0.040           0.073          .     0.017
##             eCDF Max
## distance       0.045
## age            0.013
## gender         0.007
## height         0.018
## weight         0.021
## ap_hi          0.014
## ap_lo          0.011
## cholesterol    0.010
## gluc           0.009
## smoke          0.019
## alco           0.017
##
## Summary of Balance for Matched Data:
##             Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance            0.828         0.802           1.428      1.205     0.401
## age                51.364        53.477          -0.312      1.105     0.075
## gender              1.461         1.344           0.245      1.101     0.059
## height            162.702       164.464          -0.216      1.381     0.017
## weight             69.072        74.673          -0.390      0.982     0.069
## ap_hi             135.774       127.965           0.052      5.931     0.017
## ap_lo             135.871        93.932           0.210      4.592     0.022
## cholesterol         1.670         1.355           0.462      1.513     0.105
## gluc                1.147         1.238          -0.158      0.636     0.030
## smoke               0.361         0.073           0.995          .     0.288
## alco                0.233         0.040           0.832          .     0.193
##             eCDF Max Std. Pair Dist.
## distance       0.798           1.428
## age            0.130           1.219
## gender         0.117           1.011
## height         0.114           1.265
## weight         0.181           1.217
## ap_hi          0.059           0.197
## ap_lo          0.049           0.354
## cholesterol    0.198           1.121
## gluc           0.057           0.609
## smoke          0.288           1.076
## alco           0.193           0.848
##
## Sample Sizes:
##           Control Treated
## All         10991   45008
## Matched     10991   10991
## Unmatched       0   34017
## Discarded       0       0
```

```
summary_matched <- summary(Nearest)
```

5.2 Checking Balance - SMD

SMD value

```
if (!require("MatchIt")) {
  install.packages("MatchIt")
  library("MatchIt")
}

# Extract matched data
Nearest_matched <- match.data(Nearest)

# Calculate Standardized Mean Differences (SMDs) for covariates
smd <- function(var, treated, control) {
  (mean(treated) - mean(control)) / sqrt((var(treated) + var(control)) / 2)
}

# Variables to assess balance
variables <- c("age", "gender", "height", "weight", "ap_hi", "ap_lo", "cholesterol", "gluc", "smoke", "a

# Calculate SMDs for each variable
smd_results <- sapply(variables, function(var) {
  smd(Nearest_matched[[var]], Nearest_matched[Nearest_matched$active == 1, ][[var]], Nearest_matched[Nea
})

# Display SMDs
smd_results
```

```
##         age      gender      height      weight       ap_hi       ap_lo
##     -0.3033      0.2406     -0.1944     -0.3895      0.0336      0.1395
## cholesterol        gluc       smoke        alco
##      0.4229     -0.1720      0.7458      0.5856
```

Balance table

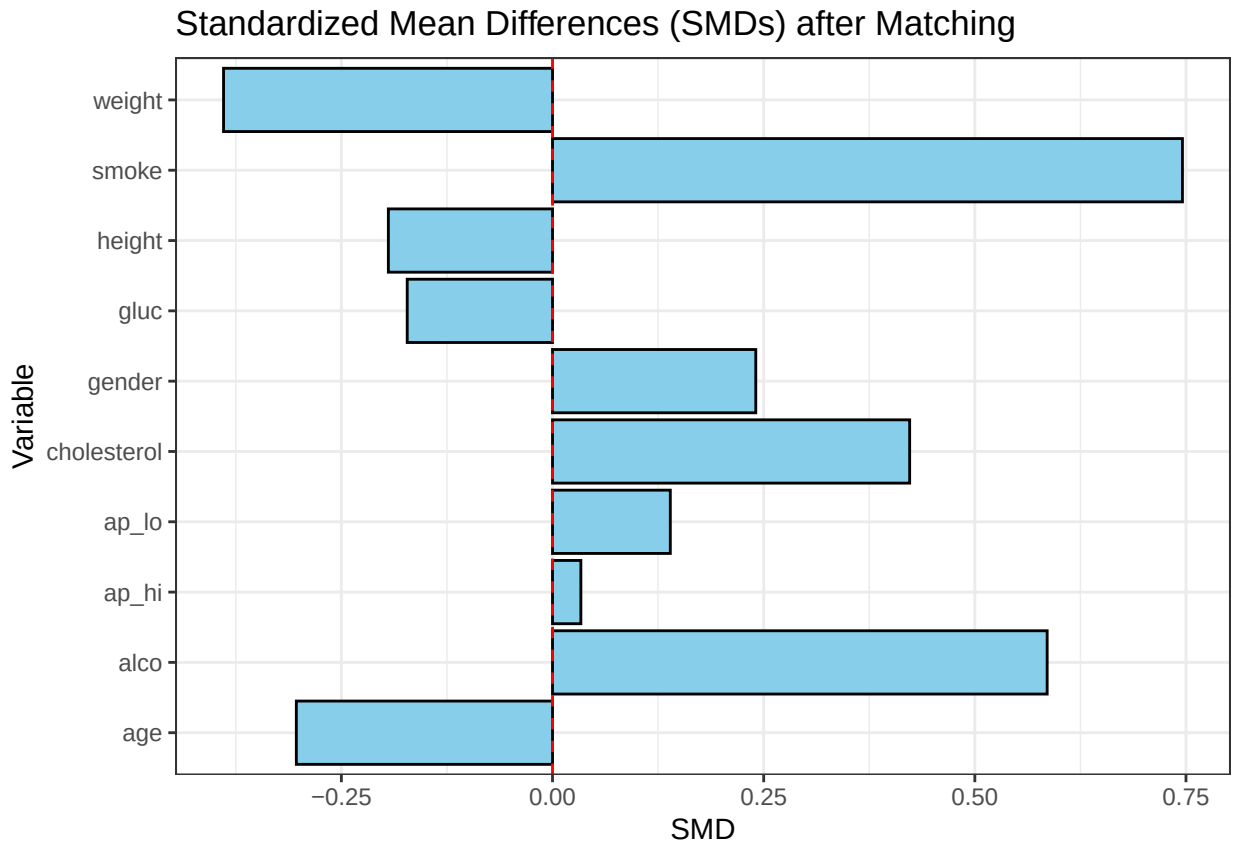
```
# Example data - replace this with your actual data
values <- c(-0.30331679, 0.24063699, -0.19435280, -0.38947625, 0.03356264, 0.13945329, 0.42285815, -0.1

# Function to determine balance based on the threshold (0.3 in this case)
check_balance <- function(value) {
  if (abs(value) <= 0.3) {
    return("Balance")
  } else {
    return("Not Balance")
  }
}

# Apply the function to each value and create a table
balance_table <- data.frame(
  Variable = c("age", "gender", "height", "weight", "ap_hi", "ap_lo", "cholesterol", "gluc", "smoke", "a
  Value = values,
```

```
  Balance_Status = sapply(values, check_balance)
)

balance_table
```

```
##       Variable   Value Balance_Status
## 1          age -0.3033    Not Balance
## 2       gender  0.2406        Balance
## 3       height -0.1944        Balance
## 4       weight -0.3895    Not Balance
## 5        ap_hi  0.0336        Balance
## 6        ap_lo  0.1395        Balance
## 7  cholesterol  0.4229    Not Balance
## 8         gluc -0.1720        Balance
## 9        smoke  0.7458    Not Balance
## 10        alco  0.5856    Not Balance
```

SMD Plot

```
# Plotting SMDs
library(ggplot2)
smd_df <- data.frame(Variable = names(smd_results), SMD = smd_results)
ggplot(smd_df, aes(x = Variable, y = SMD)) +
  geom_col(fill = "skyblue", color = "black") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  coord_flip() +
  labs(title = "Standardized Mean Differences (SMDs) after Matching",
       x = "Variable", y = "SMD") +
  theme_bw()
```

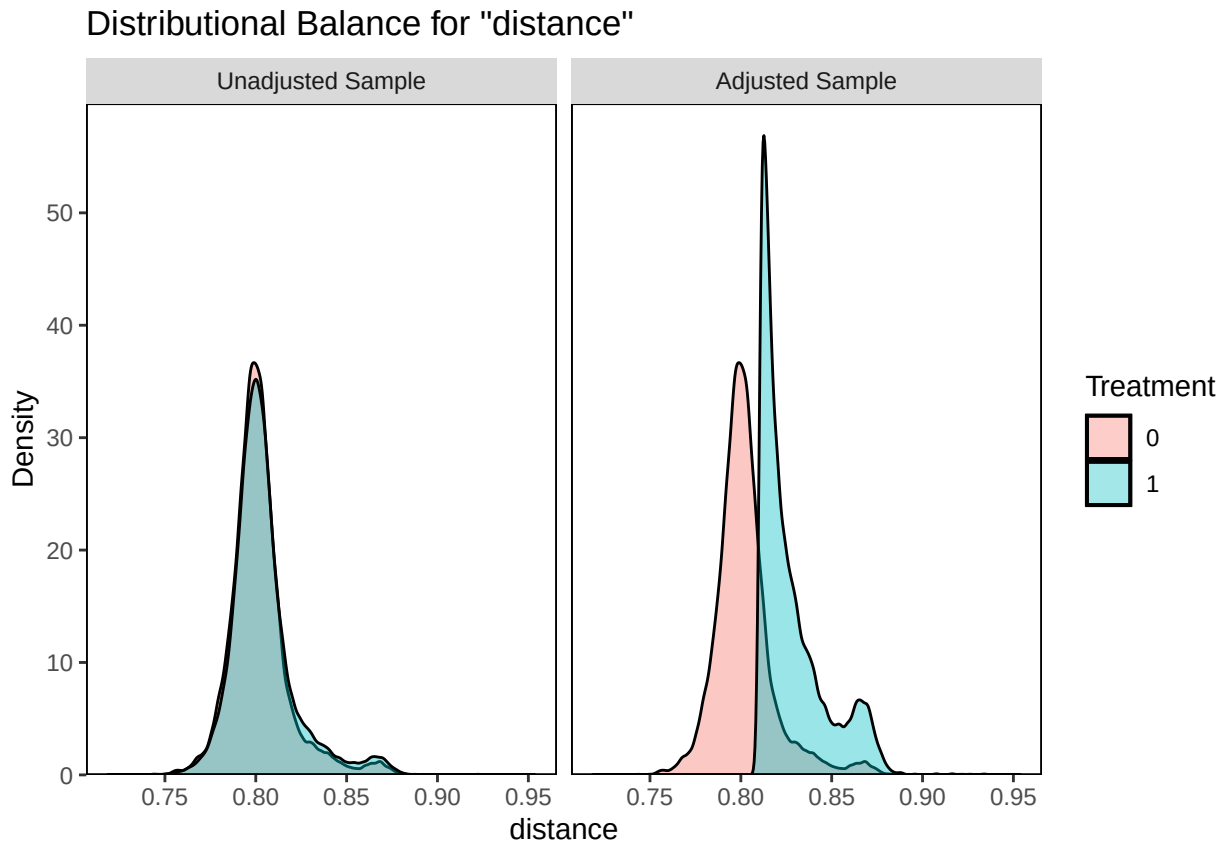


Balance plots

```
library(cobalt)

# Generate balance tables
bal_tab1 <- bal.tab(Nearest)
print(bal_tab1)
```

```
## Balance Measures
##                 Type Diff.Adj
## distance    Distance    1.428
## age          Contin.   -0.312
## gender_2      Binary    0.117
## height       Contin.   -0.216
## weight       Contin.   -0.390
## ap_hi        Contin.    0.052
## ap_lo        Contin.    0.210
## cholesterol  Contin.    0.462
## gluc         Contin.   -0.158
## smoke         Binary    0.288
## alco          Binary    0.193
##
## Sample sizes
##           Control Treated
## All         10991   45008
## Matched     10991   10991
## Unmatched       0   34017
```

```r
# Plot balance plots
bal_plot1 <- bal.plot(Nearest, which = "both")
print(bal_plot1)
```



Logistic model after matching

```r
# Logistic model after matching
lmod_after = glm(cardio~ active+ age + gender + height + weight + ap_hi + ap_lo + cholesterol + gluc + s
summary(lmod_after)$coefficients[2,]
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##   -0.01657    0.00794   -2.08754    0.03685
```

```r
confint(lmod_after,2,0.95)
```

```
##    2.5 %   97.5 %
## -0.03213 -0.00101
```

```r
summary(lmod_after)
```

```
##
```

```
## Call:
## glm(formula = cardio ~ active + age + gender + height + weight +
##     ap_hi + ap_lo + cholesterol + gluc + smoke + alco, data = Nearest_matched)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.37e-01   7.03e-02  -10.48  < 2e-16 ***
## active1     -1.66e-02   7.94e-03   -2.09    0.037 *
## age          1.45e-02   4.63e-04   31.28  < 2e-16 ***
## gender       1.43e-02   8.11e-03    1.77    0.077 .
## height      -8.49e-04   4.24e-04   -2.00    0.045 *
## weight       6.22e-03   2.43e-04   25.64  < 2e-16 ***
## ap_hi        7.17e-05   1.33e-05    5.37  7.8e-08 ***
## ap_lo        9.70e-05   1.04e-05    9.35  < 2e-16 ***
## cholesterol  1.41e-01   4.79e-03   29.48  < 2e-16 ***
## gluc        -4.01e-02   6.41e-03   -6.26  3.9e-10 ***
## smoke       -4.29e-02   9.56e-03   -4.49  7.2e-06 ***
## alco        -4.40e-02   9.97e-03   -4.42  1.0e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.211)
##
##     Null deviance: 5491.2  on 21981  degrees of freedom
## Residual deviance: 4637.6  on 21970  degrees of freedom
## AIC: 28204
##
## Number of Fisher Scoring iterations: 2
```

5.3 Analyzing Treatment Effects - ATT

ATT Table

```
#ATT
before_N_ATT=summary(lmod_before_matching)$coefficients[2,1]
after_N_ATT=summary(lmod_after)$coefficients[2,1]

# Extract SE for ATT before matching
before_N_ATT_SE = summary(lmod_before_matching)$coefficients[2, 2]

# Extract SE for ATT after matching
after_N_ATT_SE = summary(lmod_after)$coefficients[2, 2]


# Create a table
att_table <- data.frame(
  Method = c("Before Matching", "After Matching"),
  ATT = c(before_N_ATT, after_N_ATT),
  SE = c(before_N_ATT_SE, after_N_ATT_SE)
)

att_table
```

```
##            Method     ATT      SE
## 1 Before Matching -0.0422 0.00500
## 2  After Matching -0.0166 0.00794
```

5.4 T-test

```
match.matrix <- data.frame(Nearest$match.matrix)
T_index <- match(row.names(match.matrix), row.names(Nearest_matched))
C_index <- match(match.matrix$Nearest.match.matrix, row.names(Nearest_matched))
Tgroup <- Nearest_matched[T_index,]
Cgroup <- Nearest_matched[C_index,]

# Perform a paired t-test between treated and control groups
m1_test <- t.test(Tgroup$cardio, Cgroup$cardio, paired = TRUE)
m1_test
```

```
##
##  Paired t-test
##
## data:  Tgroup$cardio and Cgroup$cardio
## t = -7, df = 10990, p-value = 4e-12
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -0.0602 -0.0337
## sample estimates:
## mean difference
##         -0.0469
```

5.5 Result

```
#Results
active = data.frame(method = c("Logit before","PSM","Logit after"),est_ATT=c(summary(lmod_before_matchi
active
```

```
##         method est_ATT CI.lower CI.upper   Pvalue significant
## 1 Logit before -0.0422  -0.0520 -0.03237 3.33e-17         Yes
## 2          PSM -0.0469  -0.0602 -0.03369 4.14e-12         Yes
## 3  Logit after -0.0166  -0.0321 -0.00101 3.69e-02         Yes
```

6. Mahalanobis Distance Matching

6.1 Propensity Score Matching

```
covariates <- c("age", "gender", "height", "weight", "ap_hi", "ap_lo",
                "cholesterol", "gluc", "smoke", "alco")
tableone <- CreateTableOne(vars = covariates, strata = "active", data = heart)
print(tableone, nonnormal = covariates)
```

```
##                              Stratified by active
##                               0                      1
##   n                            13739                  56261
##   age (median [IQR])           54.00 [49.00, 59.00]   54.00 [48.00, 58.00]
##   gender (median [IQR])         1.00 [1.00, 2.00]      1.00 [1.00, 2.00]
##   height (median [IQR])       165.00 [159.00, 170.00] 165.00 [159.00, 170.00]
##   weight (median [IQR])        72.00 [65.00, 83.00]   72.00 [65.00, 82.00]
##   ap_hi (median [IQR])        120.00 [120.00, 140.00] 120.00 [120.00, 140.00]
##   ap_lo (median [IQR])         80.00 [80.00, 90.00]   80.00 [80.00, 90.00]
##   cholesterol (median [IQR])    1.00 [1.00, 1.00]      1.00 [1.00, 2.00]
##   gluc (median [IQR])           1.00 [1.00, 1.00]      1.00 [1.00, 1.00]
##   smoke (median [IQR])          0.00 [0.00, 0.00]      0.00 [0.00, 0.00]
##   alco (median [IQR])           0.00 [0.00, 0.00]      0.00 [0.00, 0.00]
##                              Stratified by active
##                               p      test
##   n
##   age (median [IQR])           0.008 nonnorm
##   gender (median [IQR])        0.121 nonnorm
##   height (median [IQR])        0.051 nonnorm
##   weight (median [IQR])       <0.001 nonnorm
##   ap_hi (median [IQR])         0.737 nonnorm
##   ap_lo (median [IQR])         0.368 nonnorm
##   cholesterol (median [IQR])   0.045 nonnorm
##   gluc (median [IQR])          0.023 nonnorm
##   smoke (median [IQR])        <0.001 nonnorm
##   alco (median [IQR])         <0.001 nonnorm
```

Estimate propensity scores

```r
# Estimate propensity scores
ps_model <- glm(active ~ age + gender + height + weight + ap_hi + ap_lo +
                cholesterol + gluc + smoke + alco + cardio, data = heart, family = "binomial")
heart$propensity_score <- predict(ps_model, type = "response")
```

Mahalanobis distance matching method

```r
mahalanobis<- matchit(active ~ propensity_score, data = heart, method = "nearest", ratio = 1)
mahalanobis
```

```
## A matchit object
##  - method: 1:1 nearest neighbor matching without replacement
##  - distance: Propensity score
##              - estimated with logistic regression
##  - number of obs.: 70000 (original), 27478 (matched)
##  - target estimand: ATT
##  - covariates: propensity_score
```

```r
matchsum2 <- summary(mahalanobis, standardize = TRUE)$sum.matched
matched_summary2 <- data.frame(round(matchsum2[,1:3], 3))

matched_summary2
```

Summary statistics of matched groups

```
##                   Means.Treated Means.Control Std..Mean.Diff.
## distance                  0.831         0.801            1.35
## propensity_score          0.831         0.801            1.38
```

```r
mahalanobis_matched <- match.data(mahalanobis)
```

```r
summary(mahalanobis)
```

```
##
## Call:
## matchit(formula = active ~ propensity_score, data = heart, method = "nearest",
##     ratio = 1)
##
## Summary of Balance for All Data:
##                   Means Treated Means Control Std. Mean Diff. Var. Ratio
## distance                  0.804         0.801           0.139       1.04
## propensity_score          0.804         0.801           0.138       1.06
##                   eCDF Mean eCDF Max
## distance              0.038    0.065
## propensity_score      0.038    0.065
##
## Summary of Balance for Matched Data:
##                   Means Treated Means Control Std. Mean Diff. Var. Ratio
## distance                  0.831         0.801            1.35      0.368
## propensity_score          0.831         0.801            1.38      0.503
##                   eCDF Mean eCDF Max Std. Pair Dist.
## distance              0.404    0.792            1.35
## propensity_score      0.404    0.792            1.38
##
## Sample Sizes:
##           Control Treated
## All         13739   56261
## Matched     13739   13739
## Unmatched       0   42522
## Discarded       0       0
```

```r
summary_matched2 <- summary(mahalanobis)
```

Logistic model after matching

```r
# Logistic model after matching
lmod_after2 = glm(cardio~ active+ age + gender + height + weight + ap_hi + ap_lo + cholesterol + gluc +
summary(lmod_after2)$coefficients[2,]
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##   -0.44265    0.00577  -76.76233    0.00000
```

```r
summary(lmod_after2)
```

```
##
## Call:
## glm(formula = cardio ~ active + age + gender + height + weight +
##     ap_hi + ap_lo + cholesterol + gluc + smoke + alco, data = mahalanobis_matched)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.34e-01   5.60e-02    2.39    0.017 *
## active1     -4.43e-01   5.77e-03  -76.76  < 2e-16 ***
## age          9.81e-03   3.65e-04   26.85  < 2e-16 ***
## gender       2.95e-02   6.28e-03    4.70  2.7e-06 ***
## height      -3.28e-03   3.39e-04   -9.69  < 2e-16 ***
## weight       3.02e-03   1.93e-04   15.66  < 2e-16 ***
## ap_hi        7.57e-05   1.00e-05    7.56  4.0e-14 ***
## ap_lo        1.09e-04   8.95e-06   12.19  < 2e-16 ***
## cholesterol  1.43e-01   4.03e-03   35.51  < 2e-16 ***
## gluc        -6.45e-02   4.92e-03  -13.10  < 2e-16 ***
## smoke        1.20e-01   7.69e-03   15.61  < 2e-16 ***
## alco         2.06e-01   8.10e-03   25.43  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.162)
##
##     Null deviance: 6236.1  on 27477  degrees of freedom
## Residual deviance: 4462.3  on 27466  degrees of freedom
## AIC: 28058
##
## Number of Fisher Scoring iterations: 2
```

6.2 Checking Balance - SMD

```r
if (!require("MatchIt")) {
  install.packages("MatchIt")
  library("MatchIt")
}

# Extract matched data
mahalanobis_matched <- match.data(mahalanobis)

# Calculate Standardized Mean Differences (SMDs) for covariates
```

```r
smd2 <- function(var, treated, control) {
  (mean(treated) - mean(control)) / sqrt((var(treated) + var(control)) / 2)
}

# Variables to assess balance
variables <- c("age", "gender", "height", "weight", "ap_hi", "ap_lo", "cholesterol", "gluc", "smoke", "a

# Calculate SMDs for each variable
smd2_results <- sapply(variables, function(var) {
  smd2(mahalanobis_matched[[var]], mahalanobis_matched[mahalanobis_matched$active == 1, ][[var]], mahala
})
```

SMD value

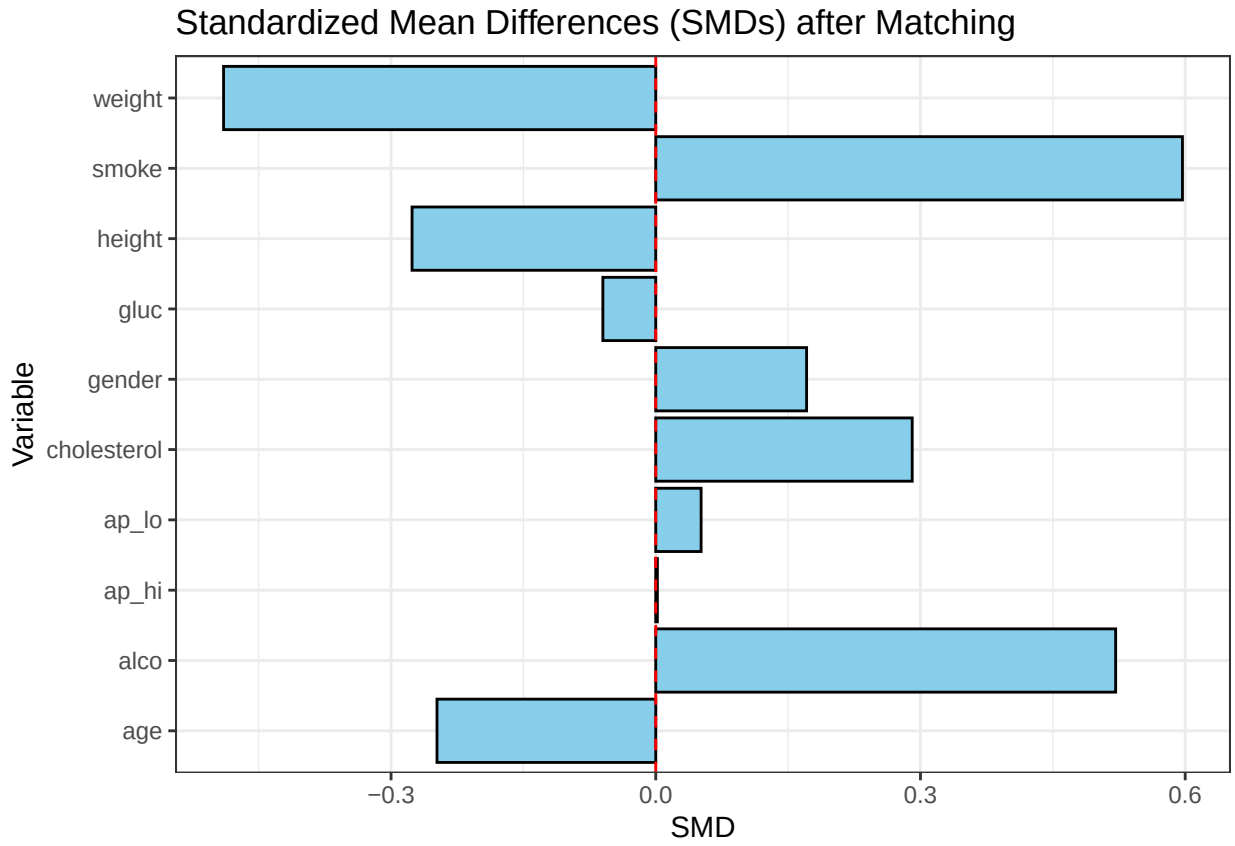
```r
# Display SMDs
smd2_results
```

```
##         age      gender      height      weight       ap_hi       ap_lo
##    -0.24793     0.17096    -0.27620    -0.48981     0.00183     0.05140
## cholesterol        gluc       smoke        alco
##     0.29065    -0.05994     0.59690     0.52125
```

```r
values <- c(-0.247934811, 0.170958004, -0.276199321, -0.489808888, 0.001829254, 0.051404742, 0.290654308

check_balance <- function(value) {
  if (abs(value) <= 0.3) {
    return("Balance")
  } else {
    return("Not Balance")
  }
}

# Apply the function to each value and create a table
balance_table <- data.frame(
  Variable = c("age", "gender", "height", "weight", "ap_hi", "ap_lo", "cholesterol", "gluc", "smoke", "a
  Value = values,
  Balance_Status = sapply(values, check_balance)
)

balance_table
```

```
##       Variable    Value Balance_Status
## 1          age -0.24793        Balance
## 2       gender  0.17096        Balance
## 3       height -0.27620        Balance
## 4       weight -0.48981    Not Balance
## 5        ap_hi  0.00183        Balance
## 6        ap_lo  0.05140        Balance
## 7  cholesterol  0.29065        Balance
## 8         gluc -0.05994        Balance
## 9        smoke  0.59690    Not Balance
## 10        alco  0.52125    Not Balance
```

```r
smd_df <- data.frame(Variable = names(smd2_results), SMD = smd2_results)
# Plotting SMDs
library(ggplot2)

ggplot(smd_df, aes(x = Variable, y = SMD)) +
  geom_col(fill = "skyblue", color = "black") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  coord_flip() +
  labs(title = "Standardized Mean Differences (SMDs) after Matching",
       x = "Variable", y = "SMD") +
  theme_bw()
```

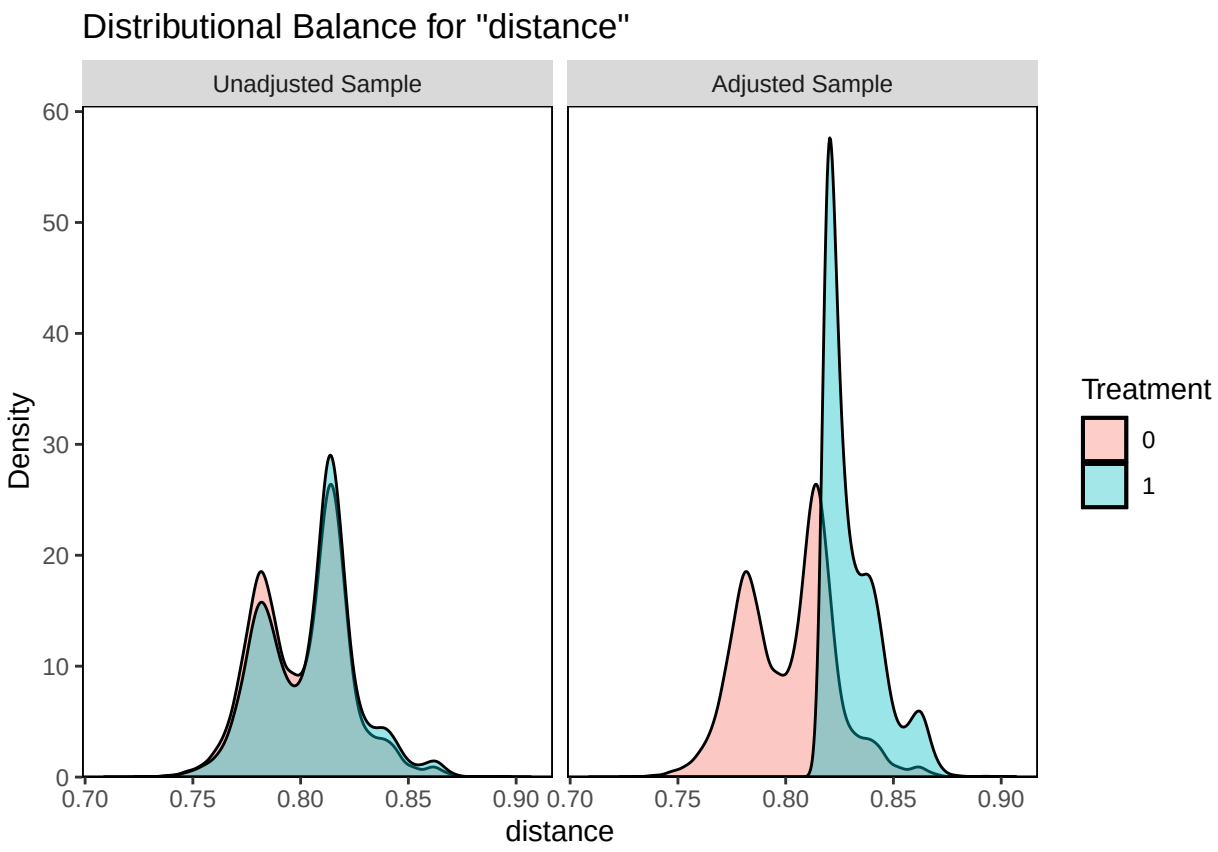



```r
library(cobalt)

# Generate balance tables
bal_tab2 <- bal.tab(mahalanobis)
print(bal_tab2)
```

```
## Balance Measures
##                      Type Diff.Adj
## distance         Distance     1.35
## propensity_score  Contin.     1.38
##
## Sample sizes
##           Control Treated
```

```
## All          13739   56261
## Matched      13739   13739
## Unmatched        0   42522
```

```
# Plot balance plots
bal_plot2 <- bal.plot(mahalanobis, which = "both")
print(bal_plot2)
```



6.3 Analyzing Treatment Effects - ATT

```
#ATT
before_M_ATT=summary(ps_model)$coefficients[2,1]
after_M_ATT=summary(lmod_after2)$coefficients[2,1]


# Extract SE for ATT before matching
before_M_ATT_SE = summary(ps_model)$coefficients[2, 2]


# Extract SE for ATT after matching
after_M_ATT_SE = summary(lmod_after2)$coefficients[2, 2]
```

```r
att_table <- data.frame(
  Method = c("Before Matching", "After Matching"),
  ATT = c(before_M_ATT, after_M_ATT),
  SE = c(before_M_ATT_SE, after_M_ATT_SE)
)

att_table
```

```
##            Method       ATT      SE
## 1 Before Matching -0.000781 0.00147
## 2  After Matching -0.442652 0.00577
```

6.4 T-test

```r
m2_test <- t.test(cardio ~ active, data =  mahalanobis_matched)
m2_test
```

```
##
##  Welch Two Sample t-test
##
## data:  cardio by active
## t = 71, df = 25248, p-value <2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  0.365 0.386
## sample estimates:
## mean in group 0 mean in group 1
##           0.536           0.161
```

6.5 Result

```r
result_mahalanobis <- data.frame(
  method = c("Logit before", "PSM", "Logit after"),
  est_ATT = c(summary(ps_model)$coefficients[2, 1], after_M_ATT, summary(lmod_after2)$coefficients[2, 1]
  CI.lower = c(confint(ps_model, 2, 0.95)[1], after_M_ATT - 1.96 * after_M_ATT_SE, confint(lmod_after2,
  CI.upper = c(confint(ps_model, 2, 0.95)[2], after_M_ATT + 1.96 * after_M_ATT_SE, confint(lmod_after2,
  Pvalue = c(summary(ps_model)$coefficients[2, 4], m2_test$p.value, summary(lmod_after2)$coefficients[2
  significant = c("No", "Yes", "Yes")
)

result_mahalanobis
```

```
##         method   est_ATT CI.lower CI.upper Pvalue significant
## 1 Logit before -0.000781 -0.00366   0.0021  0.595          No
## 2          PSM -0.442652 -0.45395  -0.4313  0.000         Yes
## 3  Logit after -0.442652 -0.45395  -0.4313  0.000         Yes
```