Biost 578: Problem Set 2

Department of Biostatistics @ University of Washington

Alejandro Hernandez

Due Friday 3, May 2024

This practice is organized around the causal question "Does being physically active cause you to live longer?" We will practice the methods we have learned (optimal multivariate matching, outcome regression, IPW, and AIPW, balancing estimators) using data from NHANES I Epidemiologic Follow-up Study. We will also perform sensitivity analyses to gauge the robustness of the evidence to possible unmeasured confounding.

The dataset `nhanesi_class_dataset.csv` is posted on the course website. More detail of the data can be found in the paper by Davis et al. (1994), "Health behaviors and survival among middle aged and older men and women in the NHANES I Epidemiologic Follow-Up Study."

The NHANES I sample was interviewed in 1971 and followed for survival until 1992. Physical activity was measured in two variables: self-reported nonrecreational activity and self-reported recreational activity. We consider the treatment to be adults who reported themselves to be "quite inactive", both at work and at leisure, and we will compare them to controls who were quite active ("very active" in physical activity outside of recreation and "much" or "moderate" recreational activity). The treatment variable is physically.inactive. Following Davis et al. (1994), we excluded people who were quite ill at the time of the NHANES I survey. We included people aged between 45 and 74 at baseline, and excluded people who, prior to NHANES I, had heart failure, a heart attack, stroke, diabetes, polio or paralysis, a malignant tumor, or a fracture of the hip or spine.

The measured confounders are the following:

- sex
- smoking status (current smoker, former smoker or never smoker)
- income.poverty.ratio: ratio of household income to poverty line for the household size, where this variable is top coded (right censored) at 9.98 (i.e., if is greater than 9.98, it is coded as 9.98).
- age at time of interview
- race (white vs. non-white)
- education (8 years, 9-11 years, high school graduate but no college, some college, college graduate)
- working.during.last.three months - employed or not during the previous three months
- marital status
- alcohol consumption (never, 1 time per month, 1-4 times per month, 2+ times per week, just about everyday
- dietary adequacy (number of five nutrients - protein, calcium, iron, Vitamin A and Vitamin C - that were consumed at more than two thirds of the recommended dietary allowance)

The outcome of interest is years.lived.since.1971.up.to.1992, the number of years the person was alive between the interview in 1971 up until 1992 (the maximum value is 21 since followup ended in 1992).

1

`income.poverty.ratio` and `dietary.adequacy` have missing values (indicated by NA). Create indicator variables for whether `income.poverty.ratio` and `dietary.adequacy` have missing values and fill in the missing values with the mean of the observed values. [Note that education has a few missing values but Missing is already coded as a category for education].

2

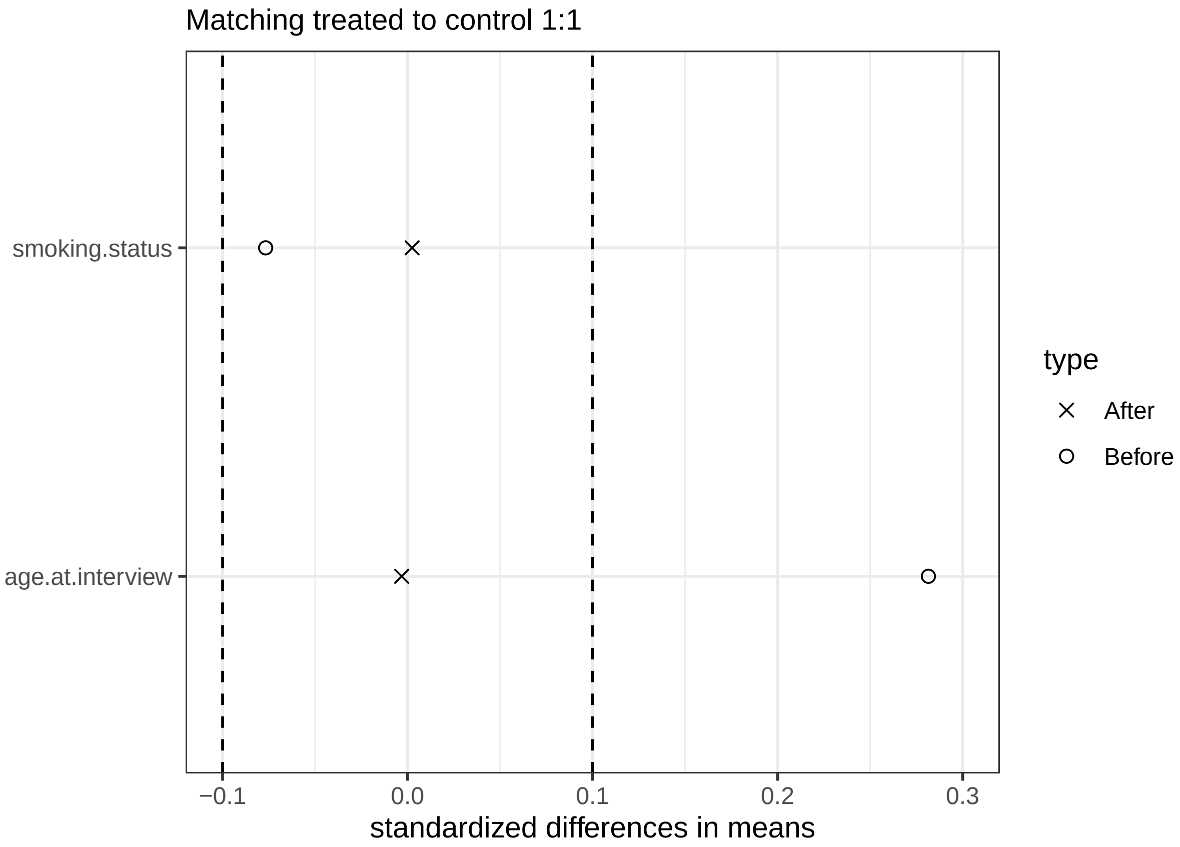
Fit a propensity score model adjusting for the confounders and the two missing indicators in Q1. To find a subset of the units with overlap, follow the procedure of Dehejia and Wahba (1999, Journal of American Statistical Association): exclude from further analysis any treated unit whose propensity score is greater than the maximum propensity score of the control units and exclude any control unit whose propensity score is less than the minimum propensity score of the treated units. How many (if any) units are excluded by this procedure?

7 participatns are removed by truncation of propensity scores.

3

Form optimal matched pairs using rank based Mahalanobis distance with a propensity score caliper, using the following prognostically important variables in the Mahalanobis distance – smoking status, sex and age at time of interview. Assess the balance on the confounders between the treated and control matched pairs. Compare it with the balance before matching in terms of the standardized differences (Lecture 2, page 22). Construct a Love plot. Generate a table 1 for the matched samples.

```
##                  stand.diff.before stand.diff.after
## sex                          -0.24                0
## smoking.status               -0.08                0
## age.at.interview              0.28                0
##                  treatmean.after controlmean.before controlmean.after
## sex                         1.42               1.54              1.42
## smoking.status              2.12               2.19              2.12
## age.at.interview           61.69              59.25             61.72
```



4

Consider matching 2 controls to each treated unit. Is there adequate balance to do so? If there is, consider matching 3 controls to each treated unit and decide if there is adequate balance to do so.

Matching treated and control units 1:1 gives us a subset with almost perfect balance across matching covariates. Matching 1:2 worsens age imbalance and matching 1:3 induces similar imbalance as the unmatched sample.

```
##                  stand.diff.before stand.diff.after
## sex                          -0.24            -0.02
## smoking.status               -0.08            -0.05
## age.at.interview              0.28             0.04
##                  treatmean.after controlmean.before controlmean.after
## sex                         1.42               1.54              1.43
## smoking.status              2.12               2.19              2.17
## age.at.interview           61.69              59.25             61.37
```

```
##                  stand.diff.before stand.diff.after
## sex                          -0.24            -0.20
## smoking.status               -0.08            -0.04
## age.at.interview              0.28             0.27
##                  treatmean.after controlmean.before controlmean.after
## sex                         1.42               1.54              1.52
## smoking.status              2.12              2.19              2.16
## age.at.interview           61.69             59.25             59.34
```

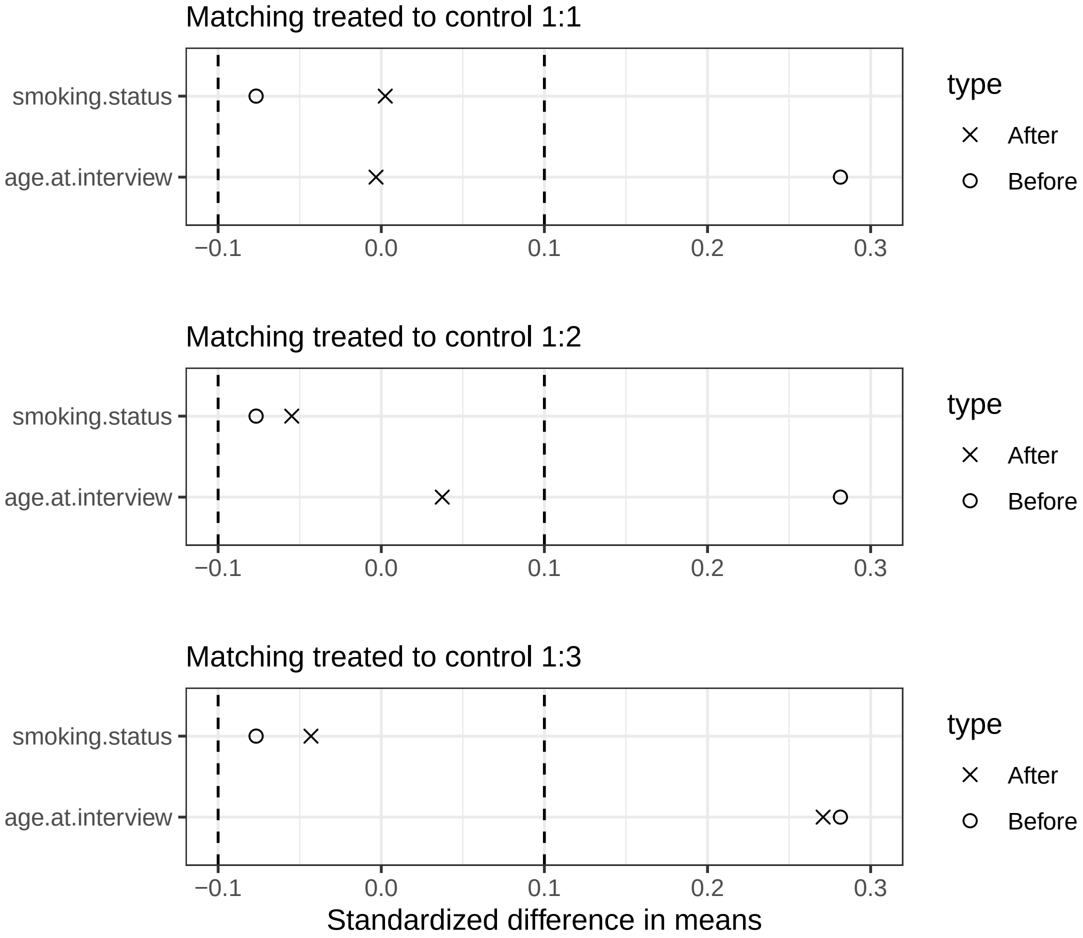


Figure 1: Love plots for covariate balance across matching

5

Which matching that you have considered do you feel is best? Justify your answer.

I most prefer matching 1:1, because it gives near perfect balance between the matched pairs.

6

Using the pair matching from Q3, find a point estimate and 95% confidence interval for the effect of being physically inactive compared to being physically active on `years.lived.since.1971.up.to.1992`. Using your chosen matching from Q5 [if it is not pair matching], find a point estimate and 95% confidence interval for the effect of being physically inactive compared to being physically active on `years.lived.since.1971.up.to.1992`.

My calculated 95% CI for the effect of being physically inactive on years lived since 1971 up to 1992 is [-2.603, -1.228].

```
##
##  Exact Wilcoxon signed rank test
##
## data:  nhanes_treated$years.lived.since.1971.up.to.1992 and nhanes_control$years.lived.since.1971.up
## V = 14543, p-value = 3.859e-12
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -4.5 -2.5
## sample estimates:
## (pseudo)median
##          -3.75
```

7

Can you think of any potential unmeasured confounders?

At this time I cannot.

8

Use one of your matching settings (e.g., your pair matching) to perform a sensitivity analysis. Up to what value of Γ is there still evidence that being physically active causes you to live longer?

With significance level of $\alpha = 0.5$, values up to $\Gamma = 1.85$ will lead us to believe being physically active causes individuals to live longer.

Worst-case p-value	Gamma
0.000	1.00
0.006	1.55
0.012	1.60
0.034	1.85
0.053	1.90

9

Apply three alternative methods on the data without matching: outcome regression, IPW, and AIPW. These three estimators and their standard errors can be directly obtained using the R package `CausalGAM`. Compare the point estimators and standard errors of these three methods to the matching method. Also apply the entropy balancing method to estimate the average treatment effect for treated using the R package `ebal`.

The table below lists results of three ATE estimators. Using the entropy balancing method to estimate the average treatment effect for treated, my result was 18.02.

```
## ####################################################################
## AIPW Estimator:
## --------------------------------------------------------------------
## Estimated ATE:  -1.82
##                      SE           z-statistic           Pr(>|z|)
## Emp. Sandwich      0.3178           -5.7277              < 0.0001
## Estim. Asymptotic  0.2849           -6.3885              < 0.0001
## Bootstrap          0.2998           -6.0707              < 0.0001
## ######################################################################
##
## ######################################################################
## IPW Estimator:
## ----------------------------------------------------------------------
## Estimated ATE:  -1.5613
##                      SE           z-statistic           Pr(>|z|)
## Estim. Asymptotic  0.2849           -5.4793              < 0.0001
## Bootstrap          0.3036           -5.1428              < 0.0001
## ######################################################################
##
## ######################################################################
## Regression Estimator:
## ----------------------------------------------------------------------
## Estimated ATE:  -1.8117
##                      SE           z-statistic           Pr(>|z|)
## Estim. Asymptotic  0.2849           -6.3594              < 0.0001
## Bootstrap          0.2837           -6.3863              < 0.0001
## ######################################################################
##
## ######################################################################
## General Information
## ----------------------------------------------------------------------
## Control Value:  physically.inactive = FALSE
## Treated Value:  physically.inactive = TRUE
## Number of Discarded Units:  0
## Number of Truncated Propensity Scores:  0
## Number of Treated Units Before Discards/Truncations:  465
## Number of Treated Units not Discarded/Truncated:      465
## Number of Control Units Before Discards/Truncations:  1479
## Number of Control Units not Discarded/Truncated:      1479
## ######################################################################
```

Table 2: ATE estimates and standard errors from bootstrap resamples (n = 501)

estimator	estimate	boostrap.se
Outcome regression	-18.117	0.284
IPW	-1.561	0.304
AIPW	-1.820	0.300

```
## Converged within tolerance
```

10

Conduct sensitivity analysis on IPW and AIPW estimator. Up to what value of Γ is there still evidence that being physically active causes you to live longer?

We will conclude that being physically active causes you to live longer up to values of $\Gamma = 1.2$ for the IPW estimator and $\Gamma = 1.4$ for the AIPW estimator. *Code in appendix will validate that 1.3 and 1.4 will generate CIs that cover zero, for their respective estimators..*

Bonus (ungraded)

Given the treatment A, observed outcome Y, and covariates X, under the assumption that $A \perp Y(0) \mid X$ and $P(A = 1 \mid X) < 1$, develop the IPW and AIPW identification formulas for the average treatment effect for the treated $E[Y(1) - Y(0) \mid A = 1]$.

Code Appendix

```r
# clear environment
rm(list=ls())

# load relevant packages
library(tidyverse)      # data manipulation
library(knitr)          # table format
library(gridExtra)      # plot format
library(DOS)            #
library(exactRankTests) #
library(CausalGAM)      # ATE estimators
library(ebal)           # entropy balancing for ATE
library(tableone)       # convenient tabling of matched sample


# setup options
knitr::opts_chunk$set(echo = F, warning = F)
options(knitr.kable.NA = '-')
labs = knitr::all_labels()
labs = labs[!labs %in% c("setup", "llm_appendix", "allcode")]
set.seed(0927)

# data loading
nhanes <- read.csv("nhanesi_class_dataset.csv")

# color selection
colors <- c("#FC600A", # dark orange
            "#C21460", # dark pink
            "#3F0000") # darker red
### -----------------------------------------------------------
### external code, from Ting Ye

library(caret)
library(ggplot2)
library(optmatch)

# This R code is a slight modification of code in Prof. Dylan Small's lecture
# It is used to construct rank based Mahalanobis distance with propensity score caliper


optmatch_caliper <- function(datatemp, nocontrols.per.match, ps.formula,
                             mahal.formula, calipersd = .5){
  # Comment about this code and subsequent matching code
  # There is assumed to be no missing data in the variables that go into the
  # propensity score model so that there is a propensity score for every variable in
  # the data frame.
  # Fit a propensity score using logistic regression with each covariate entering
  # linearly into the logistic link function
  # Put x=TRUE in order to have model object include design matrix
  propscore.model = glm(ps.formula, family = binomial, data = datatemp)

  # This model is to obtain model.matrix for mahalanobis distance.
```

```r
mahal.model = glm(mahal.formula, family = binomial, x = TRUE,
                  y = TRUE, data = datatemp)

datatemp$treated = mahal.model$y
datatemp$treatment = datatemp$treated
# Use the caret package to include all categories of categorical variables (i.e.,
# do not leave out one category) in X matrix
dmy = dummyVars(mahal.model$formula, data = datatemp)
Xmat = data.frame(predict(dmy, newdata = datatemp))
# Matrix of covariates to include in the Mahalanobis distance, for now include all
# covariates
Xmatmahal = Xmat
treated = datatemp$treated
datatemp$logit.ps = predict(propscore.model)


# Use Hansen (2009)'s rule for removing subjects who lack overlap
logit.propscore = datatemp$logit.ps
pooled.sd.logit.propscore = sqrt(var(logit.propscore[datatemp$treatment==1])/2 + var(logit.propscore[d
min.treated.logit.propscore = min(logit.propscore[datatemp$treatment==1])
max.control.logit.propscore = max(logit.propscore[datatemp$treatment==0])
# How many treated and control subjects lack overlap by Hansen's criterion
no.treated.lack.overlap = sum(logit.propscore[datatemp$treatment==1] > (max.control.logit.propscore +
no.control.lack.overlap = sum(logit.propscore[datatemp$treatment==0] < (min.treated.logit.propscore -
# If there are subjects who lack overlap, remove them from the datatemp dataset
datatemp.original = datatemp
datatemp.full = datatemp
Xmat.original = Xmat
Xmat.full = Xmat
if (no.treated.lack.overlap+no.control.lack.overlap > 0) {
  which.remove = which((logit.propscore > (max.control.logit.propscore + .5*pooled.sd.logit.propscore)
  datatemp = datatemp[-which.remove,]
  datatemp.full = rbind(datatemp,datatemp.original[which.remove,])
  Xmat = Xmat[-which.remove,]
  Xmat.full = rbind(Xmat,Xmat.original[which.remove,])
  Xmatmahal = Xmatmahal[-which.remove,]
}
# For the purposes of balance checking later, in datatemp.full, append
# the removed rows of datatemp to the end of datatemp

# Make the rownames in datatemp be 1:number of rows
rownames(datatemp) = seq(1, nrow(datatemp), 1)

# Function for computing
# rank based Mahalanobis distance.  Prevents an outlier from
# inflating the variance for a variable, thereby decreasing its importance.
# Also, the variances are not permitted to decrease as ties
# become more common, so that, for example, it is not more important
# to match on a rare binary variable than on a common binary variable
# z is a vector, length(z)=n, with z=1 for treated, z=0 for control
# X is a matrix with n rows containing variables in the distance

smahal=
  function(z,X){
    X<-as.matrix(X)
    n<-dim(X)[1]
    rownames(X)<-1:n
    k<-dim(X)[2]
    m<-sum(z)
    for (j in 1:k) X[,j]<-rank(X[,j])
    cv<-cov(X)
    vuntied<-var(1:n)
    rat<-sqrt(vuntied/diag(cv))
    cv<-diag(rat)%*%cv%*%diag(rat)
    out<-matrix(NA,m,n-m)
    Xc<-X[z==0,]
    Xt<-X[z==1,]
    rownames(out)<-rownames(X)[z==1]
    colnames(out)<-rownames(X)[z==0]
    library(MASS)
    icov<-ginv(cv)
    for (i in 1:m) out[i,]<-mahalanobis(Xc,Xt[i,],icov,inverted=T)
    out
  }

# Function for adding a propensity score caliper to a distance matrix dmat
# calipersd is the caliper in terms of standard deviation of the logit propensity scoe
addcaliper=function(dmat,z,logitp,calipersd=.5,penalty=1000){
  # Pooled within group standard devation
  sd.logitp=sqrt((sd(logitp[z==1])^2+sd(logitp[z==0])^2)/2)
  adif=abs(outer(logitp[z==1],logitp[z==0],"-"))
  adif=(adif-(calipersd*sd.logitp))*(adif>(calipersd*sd.logitp))
  dmat=dmat+adif*penalty
  dmat
}


# Rank based Mahalanobis distance
distmat=smahal(datatemp$treated,Xmatmahal)
# Add caliper
distmat=addcaliper(distmat,datatemp$treated,datatemp$logit.ps,calipersd=.5)


# Label the rows and columns of the distance matrix by the rownames in datatemp
rownames(distmat)=rownames(datatemp)[datatemp$treated==1]
colnames(distmat)=rownames(datatemp)[datatemp$treated==0]


matchvec=pairmatch(distmat,controls=nocontrols.per.match,data=datatemp)
datatemp$matchvec=matchvec
## Create a matrix saying which control units each treated unit is matched to
## Create vectors of the subject indices of the treatment units ordered by
## their matched set and corresponding control unit
treated.subject.index=rep(0,sum(treated==1))
matched.control.subject.index.mat=matrix(rep(0,nocontrols.per.match*length(treated.subject.index)),nc
matchedset.index=substr(matchvec,start=3,stop=10)
```

```
  matchedset.index.numeric=as.numeric(matchedset.index)

  for(i in 1:length(treated.subject.index)){
    matched.set.temp=which(matchedset.index.numeric==i)
    treated.temp.index=which(datatemp$treated[matched.set.temp]==1)
    treated.subject.index[i]=matched.set.temp[treated.temp.index]
    matched.control.subject.index.mat[i,]=matched.set.temp[-treated.temp.index]
  }
  matched.control.subject.index=matched.control.subject.index.mat

  Xmat.without.missing<-Xmat.full
  treatedmat=Xmat.without.missing[datatemp.full$treated==1,];
  # Standardized differences before matching
  controlmat.before=Xmat.without.missing[datatemp.full$treated==0,];
  controlmean.before=apply(controlmat.before,2,mean,na.rm=TRUE);

  treatmean=apply(treatedmat,2,mean,na.rm=TRUE);
  treatvar=apply(treatedmat,2,var,na.rm=TRUE);
  controlvar=apply(controlmat.before,2,var,na.rm=TRUE);
  stand.diff.before=(treatmean-controlmean.before)/sqrt((treatvar+controlvar)/2);

  treatmat.after=Xmat.without.missing[treated.subject.index,]
  controlmat.after=Xmat.without.missing[matched.control.subject.index,];
  controlmean.after=apply(controlmat.after,2,mean,na.rm=TRUE);
  treatmean.after=apply(treatmat.after,2,mean,na.rm=TRUE)
  stand.diff.after=(treatmean-controlmean.after)/sqrt((treatvar+controlvar)/2)

  res.stand.diff<-cbind(stand.diff.before,stand.diff.after)
  res.mean<-cbind(treatmean.after,controlmean.before,controlmean.after)
  print(round(res.stand.diff,2))
  print(round(res.mean,2))

  abs.stand.diff.before=stand.diff.before[-1]
  abs.stand.diff.after=stand.diff.after[-1]
  covariates=names(stand.diff.before[-1])
  plot.dataframe=data.frame(abs.stand.diff=c(abs.stand.diff.before,abs.stand.diff.after),covariates=rep
  p<-ggplot(plot.dataframe,aes(x=abs.stand.diff,y=covariates))+geom_point(size=2,aes(shape=type))+scale_
  return(list(p=p,datatemp=datatemp,treated.subject.index=treated.subject.index,
              matched.control.subject.index=matched.control.subject.index,
              res.stand.diff=res.stand.diff,res.mean=res.mean))
}
### ----------------------------------------------------------
### Problem 1

# count NAs present in each column
# colSums(is.na(nhanes))

# handle NAs
nhanes <- nhanes %>%
  # cast character variables to factors
  mutate_if(is.character, as.factor) %>%
  # create missing indicator variables
  mutate(income.poverty.ratio.missingind = ifelse(is.na(income.poverty.ratio),
```

```r
                                                  1, 0),
         dietary.adequacy.missingind = ifelse(is.na(dietary.adequacy),
                                              1, 0)) %>%
  # impute missing values with column mean
  mutate_all(~ifelse(is.na(.x), mean(.x, na.rm = T), .x))
### -----------------------------------------------------------
### Problem 2
ps_formula <- physically.inactive ~ sex + smoking.status +
  income.poverty.ratio + age.at.interview + race + education +
  working.last.three.months + married + alcohol.consumption + dietary.adequacy + income.poverty.ratio.mi

# fit a (logistic regression) propensity model
ps_model <- glm(ps_formula, family = "binomial", data = nhanes)
# calculate propensity scores and inverse probability weights
nhanes$ps.score = predict(ps_model, type = "response")

# truncate propensity scores that are below min score of treated or above max
# score of control
score_range <- nhanes %>%
  filter(!physically.inactive) %>%
  summarize(max = max(ps.score), min = min(ps.score))

nhanes <- nhanes %>%
  filter(score_range$min < ps.score & ps.score < score_range$max)
# removes 7 participants
### -----------------------------------------------------------
### Problem 3

# Remark: The function optmatch caliper in optmatch.R (posted on the course website) implements the mat

# Specify all the variables you want to use in the Mahalanobis distance
# on the right handside of ~
mahal_formula <- physically.inactive ~ sex + smoking.status + age.at.interview

# Perform paired matching
# Print standardized difference (both before and after matching) and the Love plot
match_res1 <- optmatch_caliper(nhanes,
                               nocontrols.per.match = 1, calipersd = 0.5,
                               ps.formula = ps_formula,
                               mahal.formula = mahal_formula)
match1_love <- match_res1$p +
  xlim(-.1, .3) +
  labs(subtitle = "Matching treated to control 1:1") +
  theme_bw()
match1_love

# Remark: The table one can be conveniently constructed using the tableone package in R, using the foll
nhanes_treated <- match_res1$datatemp[match_res1$treated.subject.index,]
nhanes_control <- match_res1$datatemp[match_res1$matched.control.subject.index,]
nhanes_matched <- rbind(nhanes_treated, nhanes_control)

tb1 <- tableone::CreateTableOne(vars = names(nhanes)[-c(1:2)],
                      strata = "physically.inactive",
                      data = nhanes_matched)
print(tb1, showAllLevels = T, smd = F, quote = F, noSpaces = T, printToggle = F)
# note this SMD is different than the SMD discussed in the lecture,
# the latter uses the pre-matching data to estimate the pooled standard error
### -----------------------------------------------------------
### Problem 4

# perform matching 1 treated to 2 controls
match_res2 <- optmatch_caliper(nhanes,
                               nocontrols.per.match = 2, calipersd = 0.5,
                               ps.formula = ps_formula,
                               mahal.formula = mahal_formula)
match2_love <- match_res2$p + xlim(-.1, .3) + theme_bw() +
  labs(subtitle = "Matching treated to control 1:2")

# perform matching 1 treated to 3 controls
match_res3 <- optmatch_caliper(nhanes,
                               nocontrols.per.match = 3, calipersd = 0.5,
                               ps.formula = ps_formula,
                               mahal.formula = mahal_formula)
match3_love <- match_res3$p + xlim(-.1, .3) + theme_bw() +
  labs(subtitle = "Matching treated to control 1:3")

# generate combination of all three love plots
gridExtra::grid.arrange(match1_love + xlab(""),
                        match2_love + xlab(""),
                        match3_love + xlab("Standardized difference in means"))
### -----------------------------------------------------------
### Problem 6

# Remark: For example, we can apply the Wilcoxon signed rank test, as follows
wilcox.exact(nhanes_treated$years.lived.since.1971.up.to.1992,
             nhanes_control$years.lived.since.1971.up.to.1992,
             paired = T, conf.int = T, exact = T)

# We can also fit a regression model controlling for the matched set indicators,
# as follows
matched.reg.model = lm(
  years.lived.since.1971.up.to.1992 ~ physically.inactive + matchvec + sex +
    smoking.status + income.poverty.ratio + age.at.interview + race +
    education + working.last.three.months + married + alcohol.consumption +
    dietary.adequacy + income.poverty.ratio.missingind +
    dietary.adequacy.missingind,
  data = nhanes_matched)

# coef(matched.reg.model)[2]      # Point estimate of treatment effect
# confint(matched.reg.model)[2,]  # Confidence interval
#    2.5 %    97.5 %
# -2.602726 -1.228433
# ### -----------------------------------------------------------
# ### Problem 6 Supplementary
#
# # Q: Sensitivity analysis: Do results change when matching 1:2?
```

```
#
# # implement 1 treated : 2 control matching
# nhanes_treated2 <- match_res2$datatemp[match_res2$treated.subject.index,]
# nhanes_control2 <- match_res2$datatemp[match_res2$matched.control.subject.index,]
# nhanes_matched2 <- rbind(nhanes_treated2, nhanes_control2)
#
# # Remark: For example, we can apply the Wilcoxon signed rank test, as follows
# wilcox.exact(nhanes_treated2$years.lived.since.1971.up.to.1992,
#              nhanes_control2$years.lived.since.1971.up.to.1992,
#              paired = T, conf.int = T, exact = T)
#
# # We can also fit a regression model controlling for the matched set indicators,
# # as follows
# matched.reg.model2 = lm(
#   years.lived.since.1971.up.to.1992 ~ physically.inactive + matchvec + sex +
#     smoking.status + income.poverty.ratio + age.at.interview + race +
#     education + working.last.three.months + married + alcohol.consumption +
#     dietary.adequacy + income.poverty.ratio.missingind +
#     dietary.adequacy.missingind,
#   data = nhanes_matched2)
#
# coef(matched.reg.model2)[2]      # Point estimate of treatment effect
# confint(matched.reg.model2)[2,]  # Confidence interval
### ----------------------------------------------------------
### Problem 8

# Below are some key codes:
diff = nhanes_treated$years.lived.since.1971.up.to.1992 - nhanes_control$years.lived.since.1971.up.to.19

data.frame("Worst-case p-value" = c(
  # pvalues
  senWilcox(diff, gamma = 1,
            conf.int = F, alpha = 0.05, alternative = "less")[[1]],
  senWilcox(diff, gamma = 1.7,
            conf.int=F, alpha = 0.05, alternative = "less")[[1]],
  senWilcox(diff, gamma = 1.75,
            conf.int=F, alpha = 0.05, alternative = "less")[[1]],
  senWilcox(diff, gamma = 1.85,
            conf.int=F, alpha = 0.05, alternative = "less")[[1]],
  senWilcox(diff, gamma = 1.9,
            conf.int=F, alpha = 0.05, alternative = "less")[[1]]),
  # corresponding gamma values
  gamma = c(1, 1.55, 1.6, 1.85, 1.9)) %>%
  knitr::kable(digits = 3, col.names = c("Worst-case p-value", "Gamma"))
### ----------------------------------------------------------
### Problem 9

# estimate ATE with outcome regression (g-computation), IPW, and AIPW
outcome_formula_t <- years.lived.since.1971.up.to.1992 ~ physically.inactive + sex + smoking.status + i

outcome_formula_c <- outcome_formula_t

CausalGAM::estimate.ATE(pscore.formula = ps_formula,
                        pscore.family = binomial,
                        outcome.formula.t = outcome_formula_t,
                        outcome.formula.c = outcome_formula_c,
                        outcome.family = gaussian,
                        treatment.var = "physically.inactive",
                        data = subset(nhanes, select = -c(X, ps.score)),
                        var.gam.plot = F)
# g-comp ATE estimate and bootstrap s.e.* : -1.8117 | 0.2837
# IPW    ATE estimate and bootstrap s.e.* : -1.5613 | 0.3036
# AIPW   ATE estimate and bootstrap s.e.* : -1.82   | 0.2998
# *bootstrap resamples size: n = 501
data.frame(estimator = c("Outcome regression", "IPW", "AIPW"),
           estimate = c(-18.117, -1.5613, -1.82),
           boostrap.se = c(0.2837, 0.3036, 0.2998)) %>%
  kable(digits = 3, caption = "ATE estimates and standard errors from bootstrap
        resamples (n = 501)")

# estimate ATE using weighting and the entropy balancing method
treatment <- as.numeric(nhanes$physically.inactive)
covariates <- subset(nhanes,
                     select = -c(X, physically.inactive, ps.score)) %>%
  mutate_if(is.logical, as.integer)

ebal_res <- ebal::ebalance(Treatment = treatment, X = covariates)
weighted_nhanes <- data.frame(ebal_res$co.xdata)

# estimate of ATE among treated
# mean(weighted_nhanes$years.lived.since.1971.up.to.1992)
# [1] 18.02167
# ### -----------------------------------------------------------
# ### Problem 10
#
# # Below are some key codes:
# library(devtools)
# install_github("qingyuanzhao/bootsens")
# library(bootsens)
#
# # IPW
# A <- nhanes$physically.inactive
# X <- model.matrix(glm(ps_formula, family = binomial, data = nhanes))
# X <- X[,-1]
# Y <- nhanes$years.lived.since.1971.up.to.1992
#
# ## IPW, assuming no unmeasured confounder (i.e. gamma = 0 or Gamma = e^0 = 1)
# extrema.os(A, X, Y) # point estimate
# bootsens.os(A, X, Y, parallel = F) # bootstrap confidence interval (CI)
#
# ## IPW, Sensitivity analysis (gamma = log(1.2), i.e. Gamma = 1.2)
# extrema.os(A, X, Y, gamma = log(1.2)) # point estimate
# bootsens.os(A, X, Y, gamma = log(1.2), parallel = F) # bootstrap CI
# bootsens.os(A, X, Y, gamma = log(1.3), parallel = F) # bootstrap CI
# # the IPW estimator is robust to Gamma = 1.2
#
```

```
# ## AIPW, assuming no unmeasured confounder (i.e. gamma = 0 or Gamma = e^0 = 1)
# extrema.os(A, X, Y, reg.adjust = T) # point estimate
# bootsens.os(A, X, Y, reg.adjust = T, parallel = F) # bootstrap CI
#
# ## AIPW, Sensitivity analysis (Gamma = exp(gamma))
# extrema.os(A, X, Y, reg.adjust = T,gamma = log(1.2)) # point estimate
# bootsens.os(A, X, Y, reg.adjust = T, gamma = log(1.2),parallel = F) # bootstrap CI
# bootsens.os(A, X, Y, reg.adjust = T, gamma = log(1.4),parallel = F) # bootstrap CI
# bootsens.os(A, X, Y, reg.adjust = T, gamma = log(1.5),parallel = F) # bootstrap CI
# # AIPW is robust to Gamma=1.4, because it has higher power compared to the IPW
```

End of document.