Supplimentary materials for ‘High-dimensional propensity score and its machine learning extensions in residual confounding control’

Appendix Tables

Appendix Table 1: List of Journals that have cited the 2009 Epidemiology Article by Schneeweiss et al. at least 10 times

Journal Name	**# of Articles Cited the Schneeweiss et al. 2009 Article**
Pharmacoepidemiology And Drug Safety	58
Clinical Pharmacology And Therapeutics	23
American Journal Of Epidemiology	22
Epidemiology	19
Statistics In Medicine	16
BMJ Online	15
BMJ Open	15
Drug Safety	15
Diabetes Obesity And Metabolism	13
Journal Of Clinical Epidemiology	12
Biometrics	11
JAMA Network Open	11
Statistical Methods In Medical Research	11
International Journal Of Cardiology	10
Plos One	10

The list was generated based on information from Scopus up to the present date (Oct 22, 2023)

A total of 791 articles have cited the Schneeweiss et al. 2009 article, the majority of which are from health sciences journals.

Appendix Table 2: Comparison of effect estimates using different high-dimensional propensity score weighting methods, as well as crude and conventional propensity score weighting methods based on the NHANES data.

Method	OR[a]	Beta-coef	coef-SE[a]	coef-CI (2.5%)[a]	coef-CI (97.5%)	p-value
Crude (no adjustment)	2.08	0.73	0.05	0.63	0.84	< 2e-16
PS (no proxies)[c]	1.98	0.68	0.04	0.61	0.76	< 2e-16
hdPS[d]	1.52	0.42	0.04	0.35	0.49	< 2e-16
Pure LASSO[c]	1.51	0.41	0.04	0.34	0.49	< 2e-16
Hybrid[d](hdPS, then LASSO)	1.55	0.44	0.04	0.36	0.51	< 2e-16
Super learner[d] (GLM, LASSO, MARS)[b]	1.60	0.47	0.04	0.39	0.54	< 2e-16
TMLE[d] (GLM, LASSO, MARS in SL)[b]	1.57	0.45	0.05	0.34	0.55	< 2e-16
TMLE[d](only GLM in SL)[b]	1.55	0.44	0.06	0.33	0.55	2.7e-15

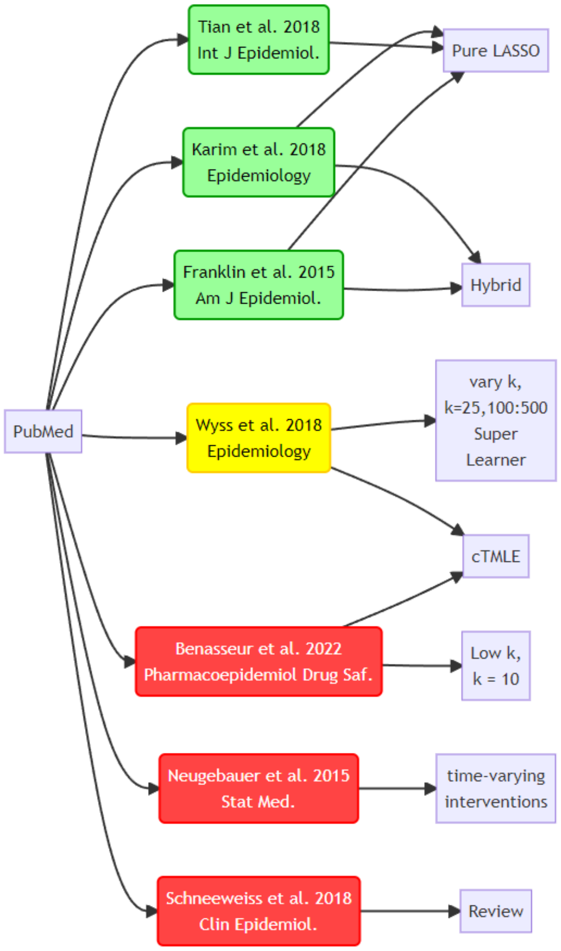
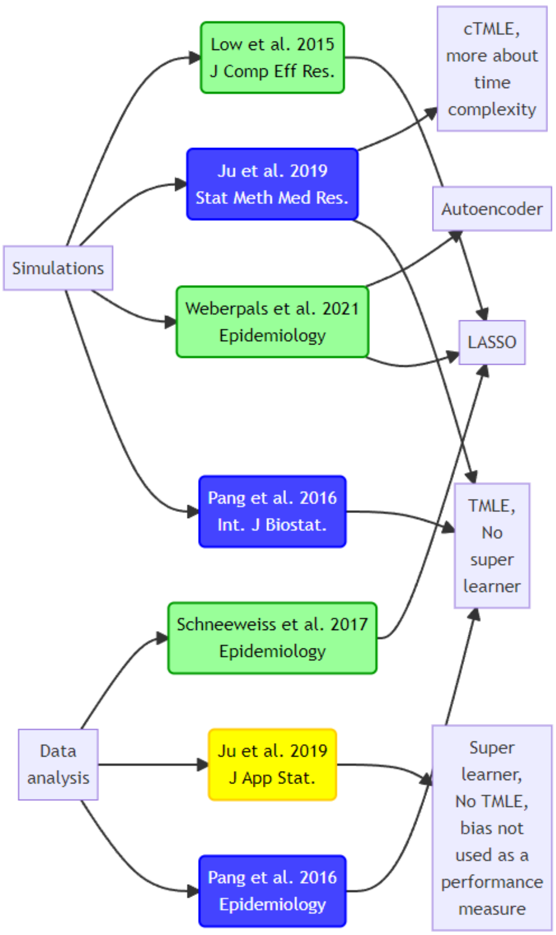
[a] OR: Odds Ratio, SE: Standard Error, CI: confidence interval.

[b] GLM: Generalized Linear Model or logistic regression, LASSO: Least Absolute Shrinkage and Selection Operator, MARS: Multivariate Adaptive Regression Splines, SL: Super Learner.

[c] PS: Propensity score weighting approach, without recurrence of hdPS variables. Only includes investigator-specified covariates.

[d] hdPS: High-dimensional propensity score weighting approach, includes investigator-specified covariates as well as proxies selected by respective methods.

Appendix Table 3: Selecting articles that discussed expansion of hdPS algorithm in the machine learning or double robust direction

From PubMed[a]	From elsewhere[b]
	

PubMed
Tian et al. 2018 Int J Epidemiol.
Karim et al. 2018 Epidemiology
Franklin et al. 2015 Am J Epidemiol.
Wyss et al. 2018 Epidemiology
Benasseur et al. 2022 Pharmacoepidemiol Drug Saf.
Neugebauer et al. 2015 Stat Med.
Schneeweiss et al. 2018 Clin Epidemiol.
Pure LASSO
Hybrid
vary k, k=25,100:500 Super Learner
cTMLE
Low k, k = 10
time-varying interventions
Review

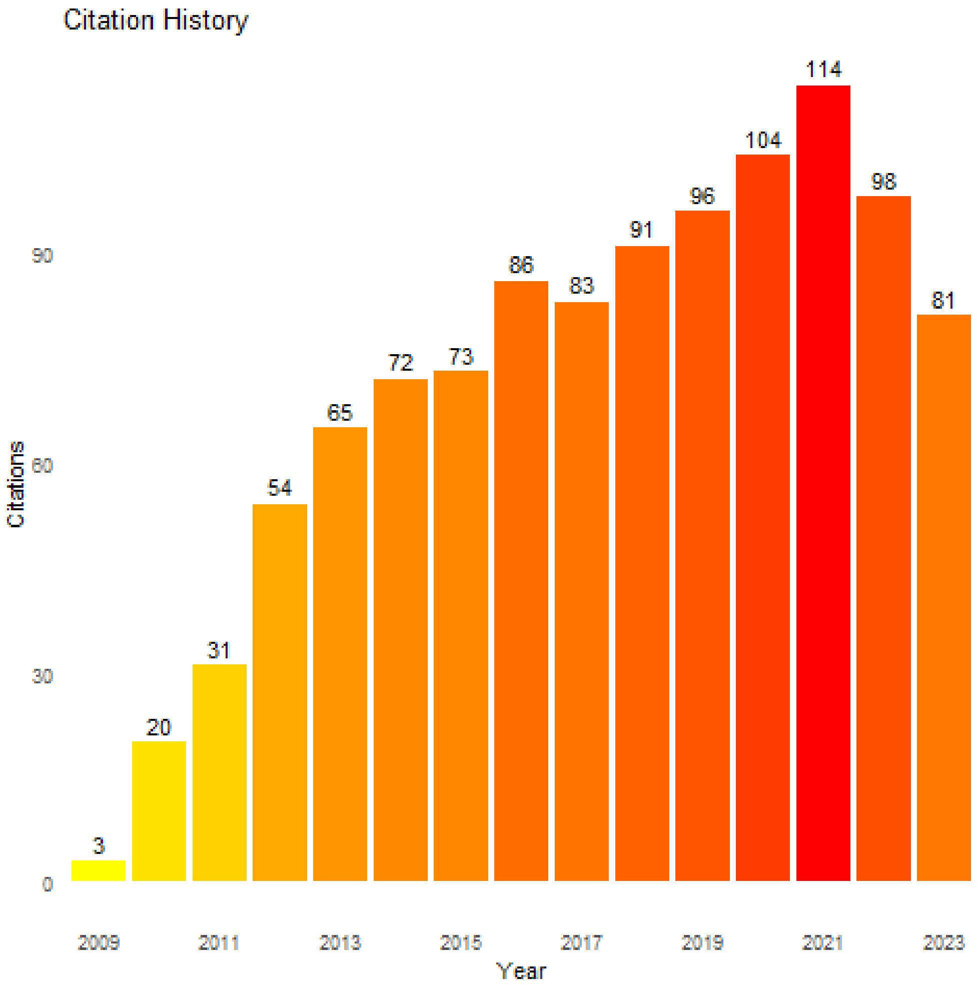
Simulations
Data analysis
Low et al. 2015 J Comp Eff Res.
Ju et al. 2019 Stat Meth Med Res.
Weberpals et al. 2021 Epidemiology
Pang et al. 2016 Int. J Biostat.
Schneeweiss et al. 2017 Epidemiology
Ju et al. 2019 J App Stat.
Pang et al. 2016 Epidemiology
cTMLE, more about time complexity
Autoencoder
LASSO
TMLE, No super learner
Super learner, No TMLE, bias not used as a performance measure

LASSO: Least Absolute Shrinkage and Selection Operator; TMLE: Targeted Maximum Likelihood Estimation; cTMLE: Collaborative Targeted Maximum Likelihood Estimation.

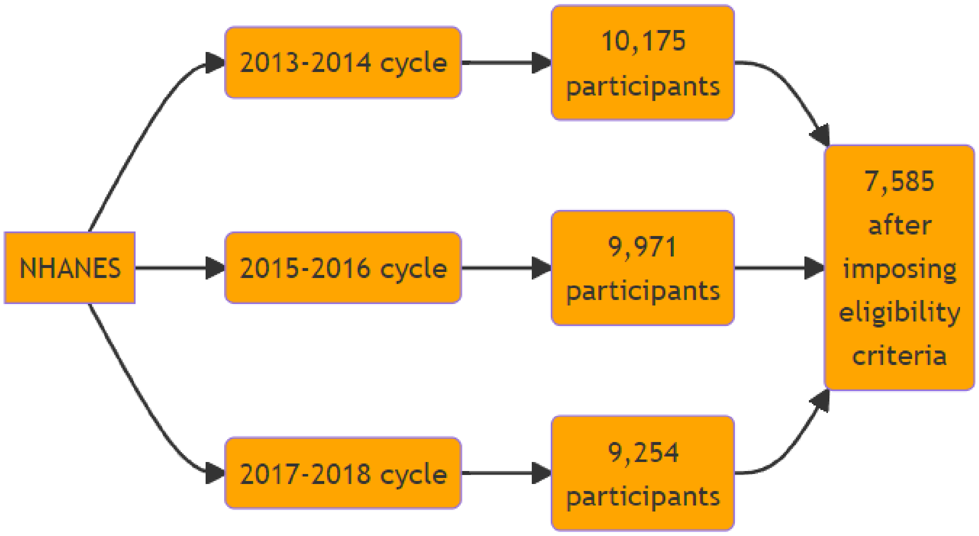
[a] PubMed search keywords from April 2023: (("plasmode"[Title/Abstract]) OR ("simulation"[Title/Abstract])) AND ("high-dimensional propensity"[Title/Abstract])

[b] References and citations based on the articles obtained from Pubmed.

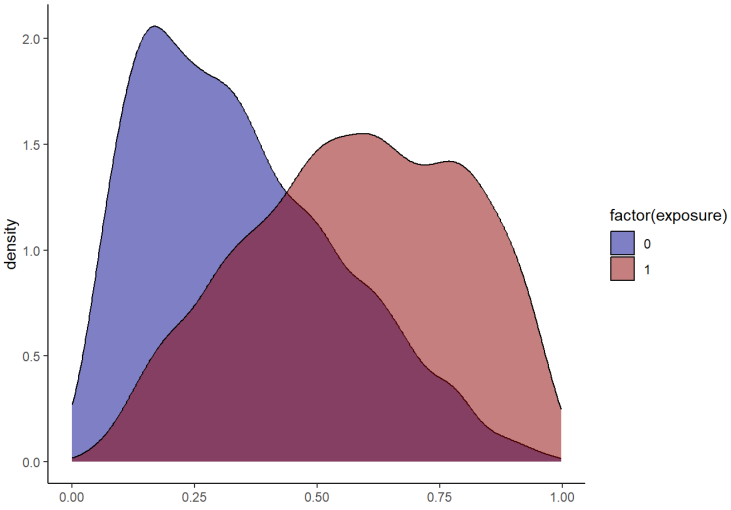
Appendix Figures



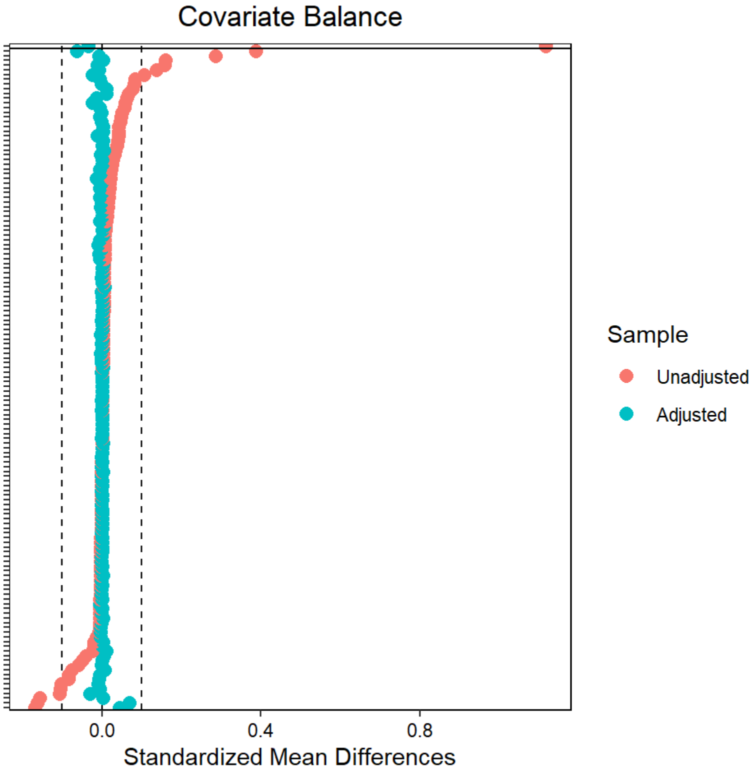
Appendix Figure 1: Citation history for Schneeweiss et al. (2009) up to November 14, 2023



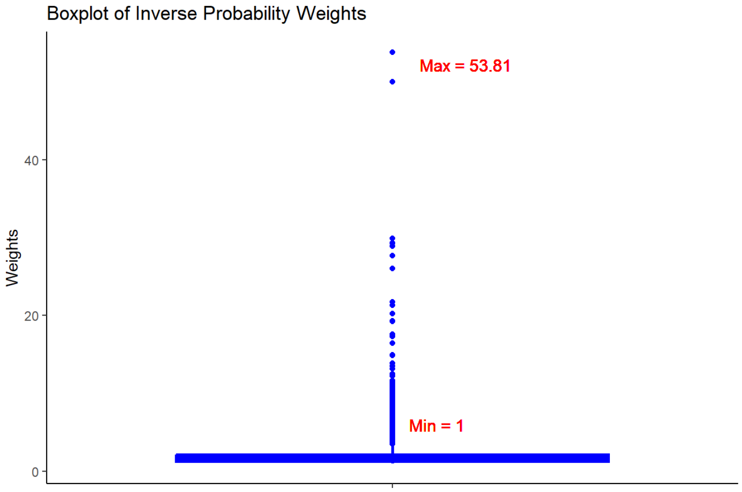
Appendix Figure 2: Sample sizes after incorporating the eligibility criteria stated in Step 0.



Appendix Figure 3: Overlap of propensity score for the exposed and unexposed group in the NHANES dataset. Horizontal axis representes the estimated propensity scores.



Appendix Figure 4: Balance of the covaiates (investigator-specified and hdPS) in the NHANES dataset. Judging by the Standardized mean difference (SMD) at 0.1 cut-off point, all covariates seem to be within satisfactory balance.



Appendix Figure 5: Boxplot of calculated inverse probability of exposure weights in the NHANES dataset. For a sample size of 7,585 patients, the maximum weight of 53.81 does not seem extreme.

Appendix R code chunks

As illustrated in R Code chunk 1, specific ICD-10-CM codes are removed from the dataset `dat.proxy.long`, which contains proxy information only. Table 4 provides examples of ICD-10-CM codes that are closely associated with the exposure (obesity) and outcome (diabetes) in this study and are thus excluded to avoid duplication.

R Code Chunk 1: Omitting duplicating information (3 digit granularity)

```
dat.proxy.long <- subset(dat.proxy.long,
                         icd10 != ”E66”)
                         # Overweight and obesity
dat.proxy.long <- subset(dat.proxy.long,
                         icd10 != ”O24”)
                         # Gestational diabetes mellitus
dat.proxy.long <- subset(dat.proxy.long,
                         icd10 != ”E10”)
                         # Type 1 diabetes mellitus
dat.proxy.long <- subset(dat.proxy.long,
                         icd10 != ”E11”)
                         # Type 2 diabetes mellitus
```

In the following code, `dfx` is the merged data combining `dat.proxy.long` and the analytic data that includes the investigator-specific covariates. `domain`, `icd10` and `idx` are the columns indicating the domain name, the associated codes and the patient ID respectively. The candidate empirical baseline covariates resulting from this process are saved in the long format in the `out1` object.

R Code Chunk 2: Generating candidate empirical baseline covariates based on their prevalence

```
require(autoCovariateSelection)
step1 <- get_candidate_covariates(df = dfx,
             domainVarname = ”domain”,
             eventCodeVarname = ”icd10”,
             patientIdVarname = ”idx”,
             patientIdVector = patientIds,
             min_num_patients = 20)
out1 <- step1[[”covars_data”]]
```

The following code saves the binary recurrence covariates in the wide format in the `out2` object.

R Code Chunk 3: Generating three recurrence covariates from the candidate empirical baseline covariates

```
step2 <- get_recurrence_covariates(df = out1,
             patientIdVarname = ”idx”,
             eventCodeVarname = ”icd10”,
             patientIdVector = patientIds)
out2 <- step2[[”recurrence_data”]]
```

In the following code, hdps.dim includes the prioritised proxy covariates that we then merge with the analytical data with the investigator-specified covariates.

R Code Chunk 4: Generating the prioritised covariates

```
out3 <- get_prioritised_covariates(df = out2,
            patientIdVarname = "idx",
            exposureVector = basetable$exposure,
            outcomeVector = basetable$outcome,
            patientIdVector = patientIds,
            k = 100)
sorted_values <- sort(out3[["multiplicative_bias"]],
                      decreasing = TRUE)
hdps.dim <- out3[["autoselected_covariate_df"]]
```

In the following code, the hdps.data is the merged data that includes the prioritized proxy covariates along with the analytical data with the investigator-specified covariates. Two new columns, exposure and outcome, are created in the hdps.data dataframe. Names of the prioritized proxy covariates are extracted into proxy.list.sel, and we create the propensity score model formula with exposure as the dependent variable, and all investigator-specified and prioritized proxy covariates on the right-hand side. We use the WeightIt package to facilitate the computation of the propensity score and inverse probability weights.

R Code Chunk 5: Propensity score model fitting for hdPS

```
hdps.data[["exposure"]] <- as.numeric(I(
                            hdps.data[["obese"]]=='Yes'))
hdps.data[["outcome"]] <- as.numeric(I(
                            hdps.data[["diabetes"]]=='Yes'))
proxy.list.sel <- names(out3[[
                    "autoselected_covariate_df"]][,-1])
proxyform <- paste0(proxy.list.sel, collapse = "+")
covform <- paste0(investigator.specified.covariates,
                  collapse = "+")
rhsformula <- paste0(c(covform, proxyform),
                  collapse = "+")
ps.formula <- as.formula(paste0("exposure",
                  "~", rhsformula))
require(WeightIt)
W.out <- weightit(ps.formula,
                  data = hdps.data,
                  estimand = "ATE",
                  method = "ps")
hdps.data[["ps"]] <- W.out[["ps"]]
hdps.data[["w"]] <- W.out[["weights"]]
```

The following code illustrates the estimation of association measures (odds ratio and risk difference) through inverse probability of weighting using the hdPS approach.

R Code Chunk 6: Estimation of association measures from hdPS

```r
# Estimating OR
out.formula <- as.formula(paste0("outcome",
                     "~", "exposure"))
fit <- glm(out.formula,
            data = hdps.data,
            weights = W.out$weights,
            family= binomial(link = "logit"))
fit.summary <- summary(fit)$coef["exposure",
                                  c("Estimate",
                                    "Std.␣Error",
                                    "Pr(>|z|)")]
fit.ci <- confint(fit, level = 0.95)["exposure", ]

# Estimating RD
fit <- glm(out.formula,
            data = hdps.data,
            weights = W.out$weights,
            family= gaussian(link = "identity"))
fit.summary <- summary(fit)$coef["exposure",
                                  c("Estimate",
                                    "Std.␣Error",
                                    "Pr(>|t|)")]
fit.summary[2] <- sqrt(sandwich::sandwich(fit)[2,2])
require(lmtest)
conf.int <- confint(fit, "exposure", level = 0.95, method = "hc1")
fit.summary <- c(fit.summary, conf.int)
round(fit.summary, 2)
```

The following code show example of fitting a conventional propensity score model.

R Code Chunk 7: Conventional propensity score model fitting

```r
covform <- paste0(investigator.specified.covariates,
                    collapse = "+")
ps.formula <- as.formula(paste0("exposure", "~", covform))
W.out <- weightit(ps.formula,
                    data = hdps.data,
                    estimand = "ATE",
                    method = "ps")
fit <- glm(out.formula,
            data = hdps.data,
            weights = W.out$weights,
            family= binomial(link = "logit"))
```

In the following code, lambda is a hyperparameter or tuning parameter in LASSO regression that controls the amount of shrinkage applied to the coefficients. Hyperparameters in machine learning encompass configurations, which influence the model's learning process and its performance on tasks. Cross-validation (we chose 5 folds in this code) can be often used to find the optimal parameter that minimizes prediction error on a validation dataset.

R Code Chunk 8: Proxy selection based on LASSO

```
proxy.list <- names(step2[["recurrence_data"]]
covarsTfull <- c(investigator.specified.covariates,
                  proxy.list)
Y.form <- as.formula(paste0(c("outcome~␣exposure",
                              covarsTfull),
                              collapse = "+") )
covar.mat <- model.matrix(Y.form, data = hdps.data)[,-1]
lasso.fit<-glmnet::cv.glmnet(y = hdps.data[["outcome"]],
                             x = covar.mat,
                             type.measure='mse',
                             family="binomial",
                             alpha = 1,
                             nfolds = 5)
coef.fit<-coef(lasso.fit,s='lambda.min',exact=TRUE)
sel.variables<-row.names(coef.fit)[
                which(as.numeric(coef.fit)!=0)]
proxy.list.sel.ml <- proxy.list[proxy.list %in%
                sel.variables]
covform <- paste0(investigator.specified.covariates,
                  collapse = "+")
proxyform <- paste0(proxy.list.sel.ml, collapse = "+")
rhsformula <- paste0(c(covform, proxyform), collapse = "+")
ps.formula <- as.formula(paste0("exposure",
                  "~", rhsformula))
```

The following code shows how to obtain estimates from hdPS approach through a hybrid framework.

R Code Chunk 9: Refining hdPS proxy selection based on LASSO

```r
proxy.list <- names(out3$autoselected_covariate_df[,-1])
covarsTfull <- c(investigator.specified.covariates,
                  proxy.list)
Y.form <- as.formula(paste0(c("outcome~␣exposure",
                              covarsTfull),
                              collapse = "+") )
covar.mat <- model.matrix(Y.form, data = hdps.data)[,-1]
lasso.fit<-glmnet::cv.glmnet(y = hdps.data$outcome,
                             x = covar.mat,
                             type.measure='mse',
                             family="binomial",
                             alpha = 1,
                             nfolds = 5)
coef.fit<-coef(lasso.fit,s='lambda.min',exact=TRUE)
sel.variables<-row.names(coef.fit)[
                which(as.numeric(coef.fit)!=0)]
proxy.list.sel.hybrid <- proxy.list[proxy.list %in%
                           sel.variables]
proxyform <- paste0(proxy.list.sel.hybrid,
                    collapse = "+")
rhsformula <- paste0(c(covform, proxyform),
                    collapse = "+")
ps.formula <- as.formula(paste0("exposure",
                    "~", rhsformula))
```

The following code shows how to obtain estimates from hdPS approach through a super learning framework.

R Code Chunk 10: Using Super learner to build the propnsity score model

```r
proxy.list <- names(step2[["recurrence_data"]]
covform <- paste0(investigator.specified.covariates,
                  collapse = "+")
proxyform <- paste0(proxy.list, collapse = "+")
rhsformula <- paste0(c(covform, proxyform), collapse = "+")
ps.formula <- as.formula(paste0("exposure",
                  "~", rhsformula))
W.out <- weightit(ps.formula,
                  data = hdps.data,
                  estimand = "ATE",
                  method = "super",
                  SL.library = c("SL.glm",
                                 "SL.glmnet",
                                 "SL.earth"))
```

The following code shows how to obtain estimates from hdPS approach through a TMLE framework.

R Code Chunk 11: Using Super learner to build the propnsity score model within the TMLE framework

```
SL.library = c(”SL.glm”, ”SL.glmnet”,”SL.earth”)
proxy.list <- names(step2[[”recurrence_data”]]
ObsData.noYA <- hdps.data[,
                c(investigator.specified.covariates,
                proxy.list)]
tmle.fit <- tmle::tmle(Y = hdps.data[[”outcome”]],
                       A = hdps.data[[”exposure”]],
                       W = ObsData.noYA,
                       family = ”binomial”,
                       V = 3,
                       Q.SL.library = SL.library,
                       g1W = W.out[[”ps”]])
```