Introduction to Marginal Standardization with R

Ashley I Naimi

September 2024

Contents

1 Marginally Adjusted Regression Models

1.1 Basic Marginal Standardization for a Continuous Outcome

Another approach to obtaining mean differences, risk differences, and risk ratios from GLMs is to use marginal standardization (Naimi et al., 2017). This process can be implemented by fitting a single model, regressing the outcome against the exposure and all confounder variables. But instead of reading the coefficients the model, one can obtain odds ratios, risk ratios, or risk differences by using this model to generate predicted risks for each individual under "exposed" and "unexposed" scenarios in the dataset. To obtain standard errors, the entire procedure must be bootstrapped.

Here is some code to implement this marginal standardization in the NHEFS data for the association between quitting smoking and weight change:

```
pacman::p_load(tidyverse,
               here,
               broom,
               boot)

nhefs <- read_csv(here("data","analytic_data.csv"))

nhefs <- nhefs %>%
  mutate(wt_delta = as.numeric(wt82_71 > median(wt82_71)))

#' Regress the outcome against the confounders with interaction
model_MD <- glm(wt82_71 ~ qsmk + sex + age + race + income + map,
                data = nhefs,
                family = gaussian("identity"))

##' Generate predictions for everyone in the sample to obtain
##' unexposed (mu0 predictions) and exposed (mu1 predictions) risks.
mu1 <- predict(model_MD,newdata=transform(nhefs,qsmk=1),type="response")
mu0 <- predict(model_MD,newdata=transform(nhefs,qsmk=0),type="response")
```

```
#' Marginally adjusted mean difference
marg_stand_MD <- mean(mu1) - mean(mu0)
```

1.2 The Bootstrap for Basic Marginal Standardization

To get the standard error or confidence intervals for the marginally standardized mean difference above, we can use the bootstrap. This entails taking R resamples from our data, with replacement, estimating the standardized effect over and over again, and then computing variability from the distribution of these standardized effects. We can do this here using the `boot` package:

```
#' Using the bootstrap to obtain confidence intervals for the marginally adjusted
#' risk ratio and risk difference.
bootfunc <- function(data,index){

  boot_dat <- data[index,]

  model_MD_ <- glm(wt82_71 ~ qsmk + sex + age + race + income + map,
                   data = boot_dat,
                   family = gaussian("identity"))

  mu1_ <- predict(model_MD_,newdata=transform(boot_dat,qsmk=1),type="response")
  mu0_ <- predict(model_MD_,newdata=transform(boot_dat,qsmk=0),type="response")

  #' Marginally adjusted mean difference
  res <- mean(mu1_) - mean(mu0_)
  return(res)
}

#' Run the boot function. Set a seed to obtain reproducibility
set.seed(123)
boot_res <- boot(nhefs,bootfunc,R=2000)

boot_MD <- boot.ci(boot_res)
```

```
marg_stand_MD
```

```
## [1] 2.832019
```

```
boot_MD
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.835,  3.812 )   ( 1.866,  3.793 )
##
## Level     Percentile            BCa
## 95%   ( 1.871,  3.798 )   ( 1.862,  3.794 )
## Calculations and Intervals on Original Scale
```

1.3 Basic Marginal Standardization for a Binary Outcome

We can do the same thing to estimate the association between quitting smoking and greater than median weight change:

```
#' Regress the outcome against the confounders with interaction
model_OR <- glm(wt_delta ~ qsmk + sex + age + race + income + map,
                data = nhefs,
                family = binomial("logit"))

##' Generate predictions for everyone in the sample to obtain
##' unexposed (mu0 predictions) and exposed (mu1 predictions) risks.
mu1 <- predict(model_OR,newdata=transform(nhefs,qsmk=1),type="response")
mu0 <- predict(model_OR,newdata=transform(nhefs,qsmk=0),type="response")
```

```r
#' Marginally adjusted odds ratio
marg_stand_OR <- (mean(mu1)/mean(1-mu1))/(mean(mu0)/mean(1-mu0))
#' Marginally adjusted risk ratio
marg_stand_RR <- mean(mu1)/mean(mu0)
#' Marginally adjusted risk difference
marg_stand_RD <- mean(mu1)-mean(mu0)

#' Using the bootstrap to obtain confidence intervals for the marginally adjusted
#' risk ratio and risk difference.
bootfunc <- function(data,index){

  boot_dat <- data[index,]

  model_OR_ <- glm(wt_delta ~ qsmk + sex + age + race + income + map,
                data = boot_dat,
                family = binomial("logit"))

  mu1 <- predict(model_OR_,newdata=transform(boot_dat,qsmk=1),type="response")
  mu0 <- predict(model_OR_,newdata=transform(boot_dat,qsmk=0),type="response")

  marg_stand_OR_ <- (mean(mu1)/mean(1-mu1))/(mean(mu0)/mean(1-mu0))
  marg_stand_RR_ <- mean(mu1)/mean(mu0)
  marg_stand_RD_ <- mean(mu1)-mean(mu0)

  res <- c(marg_stand_RD_,marg_stand_RR_,marg_stand_OR_)

  return(res)
}

#' Run the boot function. Set a seed to obtain reproducibility
set.seed(123)
boot_res <- boot(nhefs,bootfunc,R=2000)

boot_RD <- boot.ci(boot_res,index=1)
```

```
boot_RR <- boot.ci(boot_res,index=2)
boot_OR <- boot.ci(boot_res,index=3)


marg_stand_OR
```

```
## [1] 1.774049
```

```
marg_stand_RR
```

```
## [1] 1.304308
```

```
marg_stand_RD
```

```
## [1] 0.141587
```

```
boot_RD
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 1)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 0.0840,  0.1984 )   ( 0.0851,  0.1999 )
##
## Level     Percentile            BCa
## 95%   ( 0.0833,  0.1981 )   ( 0.0832,  0.1976 )
## Calculations and Intervals on Original Scale
```

```
boot_RR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
```

```
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 2)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.169,  1.436 )   ( 1.168,  1.437 )
##
## Level     Percentile            BCa
## 95%   ( 1.172,  1.441 )   ( 1.170,  1.439 )
## Calculations and Intervals on Original Scale
```

```
boot_OR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 3)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.328,  2.183 )   ( 1.295,  2.149 )
##
## Level     Percentile            BCa
## 95%   ( 1.399,  2.253 )   ( 1.397,  2.246 )
## Calculations and Intervals on Original Scale
```

1.4 Marginal Standardization with Stratified Outcome Models

This marginal standardization approach yields an estimate of the average outcome difference between exposed an unexposed. However, in this case, it assumes a constant effect of qsmk on weight change across levels of all of the other variables in the model. This constant effect assumption might be true, but if one wanted to account for potential interactions between the exposure

and all of the confounders in the model, there is an easy way.

We call this the "stratified modeling approach."

This stratified modeling approach avoids the exposure effect homogeneity assumption across levels of all the confounders. In effect, the approach fits a separate model for each exposure stratum. To obtain predictions under the "exposed" scenario, we use the model fit to the exposed individuals to generate predicted outcomes in the entire sample. To obtain predictions under the "unexposed" scenario, we repeat the same procedure, but with the model fit among the unexposed. One can then average the risks obtained under each exposure scenario, and take their difference and ratio to obtain the risk differences and ratios of interest.

```
#' Marginal Standardization
##' To avoid assuming no interaction between
##' quitting smoking and any of the other variables
##' in the model, we subset modeling among
##' exposed/unexposed. This code removes qsmk from the model,
##' which will allow us to regress the outcome
##' against the confounders among the exposed and
##' the unexposed separately. Doing so will allow us
##' to account for any potential exposure-covariate interactions
##' that may be present.

#' Regress the outcome against the confounders
#' among the unexposed (model0) and then among the exposed (model1)
model0 <- glm(wt_delta ~ sex + age + race + income + map,
              data=subset(nhefs,qsmk==0),
              family=binomial("logit"))
model1 <- glm(wt_delta ~ sex + age + race + income + map,
              data=subset(nhefs,qsmk==1),
              family=binomial("logit"))

##' Generate predictions for everyone in the sample using the model fit to only the
##' unexposed (mu0 predictions) and only the exposed (mu1 predictions).
mu1 <- predict(model1,newdata=nhefs,type="response")
```

```
mu0 <- predict(model0,newdata=nhefs,type="response")

#' Marginally adjusted odds ratio
marg_stand_OR <- (mean(mu1)/mean(1-mu1))/(mean(mu0)/mean(1-mu0))
#' Marginally adjusted risk ratio
marg_stand_RR <- mean(mu1)/mean(mu0)
#' Marginally adjusted risk difference
marg_stand_RD <- mean(mu1)-mean(mu0)

#' Using the bootstrap to obtain confidence intervals for the marginally adjusted
#' risk ratio and risk difference.
bootfunc <- function(data,index){
  boot_dat <- data[index,]
  model0 <- glm(wt_delta ~ sex + age + race + income + map,
                data=subset(boot_dat,qsmk==0),
                family=binomial("logit"))
  model1 <- glm(wt_delta ~ sex + age + race + income + map,
                data=subset(boot_dat,qsmk==1),
                family=binomial("logit"))

  mu1 <- predict(model1,newdata=boot_dat,type="response")
  mu0 <- predict(model0,newdata=boot_dat,type="response")

  marg_stand_OR_ <- (mean(mu1)/mean(1-mu1))/(mean(mu0)/mean(1-mu0))
  marg_stand_RR_ <- mean(mu1)/mean(mu0)
  marg_stand_RD_ <- mean(mu1)-mean(mu0)
  res <- c(marg_stand_RD_,marg_stand_RR_,marg_stand_OR_)
  return(res)
}

#' Run the boot function. Set a seed to obtain reproducibility
set.seed(123)
boot_res <- boot(nhefs,bootfunc,R=2000)
```

```
boot_RD <- boot.ci(boot_res,index=1)
boot_RR <- boot.ci(boot_res,index=2)
boot_OR <- boot.ci(boot_res,index=3)

marg_stand_OR
```

```
## [1] 1.793597
```

```
marg_stand_RR
```

```
## [1] 1.309761
```

```
marg_stand_RD
```

```
## [1] 0.1441892
```

```
boot_RD
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 1)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 0.0860,  0.2017 )   ( 0.0884,  0.2049 )
##
## Level     Percentile            BCa
## 95%   ( 0.0835,  0.2000 )   ( 0.0819,  0.1991 )
## Calculations and Intervals on Original Scale
```

```
boot_RR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 2)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.173,  1.442 )   ( 1.175,  1.448 )
##
## Level     Percentile            BCa
## 95%   ( 1.171,  1.445 )   ( 1.169,  1.444 )
## Calculations and Intervals on Original Scale
```

```
boot_OR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 3)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.337,  2.212 )   ( 1.313,  2.188 )
##
## Level     Percentile            BCa
## 95%   ( 1.400,  2.274 )   ( 1.391,  2.263 )
## Calculations and Intervals on Original Scale
```

When predicted risks are estimated using a logistic model, relying on marginal standardization will not result in probability estimates outside the bounds [0, 1]. And because the robust variance estimator is not required, model-based standardization will not be as affected by small sample sizes. However, the bootstrap is more computationally demanding than alternative variance estimators, which may pose problems in larger datasets.

References

Ashley I Naimi, Stephen R Cole, and Edward H Kennedy. An Introduction to G
Methods. *Int J Epidemiol*, 46(2):756–62, 2017.