

Conditional and Marginal Adjustment

Ashley I Naimi

October 2026

Contents

1	Introduction	2
2	Collapsibility and Noncollapsibility: Basic Concepts	2
3	Noncollapsibility in a Regression Context	5
3.1	Unadjusted Regression Models	6
3.2	Conditionally Adjusted Model	7
3.3	Marginally Adjusted Model	7
4	Marginally and Conditionally Adjusted Regression: NHEFS	8
4.1	Conditional Adjustment	8
4.2	Marginal Adjustment	12
5	Marginal and Conditional Adjustment with Interactions	18
6	Marginal Standardization with a Stratified Outcome Model	22
6.1	Interaction Terms: Review	22
6.2	Exposure Stratification and Interaction	22
7	Propensity Scores and Inverse Probability Weighting	27
7.1	Propensity Score Methods	27
7.2	Inverse Probability Weighting: Intuition	30
7.3	Inverse Probability Weighting in Practice	32
8	Conclusion	38
	Bonus Material	39
	Combining Coefficients for a Conditional Contrast	40
	IP Weighting for the ATT and ATU	42

1 Introduction

This week we cover concepts related to collapsibility and non-collapsibility of three measures of association: the odds ratio, the risk ratio, and the risk difference. We consider non-collapsibility in a regression context, and use this as a background to discuss the distinctions between marginal and conditional regression adjustment.

We then turn to the interpretation of marginally and conditionally adjusted association measures, the role that interactions play in these settings, and a few ways to obtain a marginally adjusted estimate, including a few versions of marginal standardization, and inverse probability weighting.

In week 5, we split the right hand side of a regression model into a target function (e.g., the exposure term coded for interpretation), and a nuisance function (the confounder adjustment portion of the model, coded for performance of the target function). This week is about what the target function returns.

When we read the exposure coefficient from an adjusted model, we get an association that is *conditionally* adjusted. What this means specifically can depend on a few things. First, if exposure-confounder interactions are included in the model, then we have to interpret the “exposure effect” relative to some specific confounder level.¹ Second, for some link functions, whether or not there are interactions in the model, yield conditionally adjusted results that are different than marginally adjusted effects.

When we instead average the model's predictions over covariates, we get an association that is *marginal* with respect to them. For some association measures these two numbers coincide. For the odds ratio they do not (even when the covariates are not confounders).

We explain these concepts here.

What to read: Daniel et al. Making apples from oranges: Comparing noncollapsible effect estimators and their standard errors after adjustment for different covariate sets. *Biometrical J* 2021;63:528-557.

¹ In the bonus material at the end of these notes, I show you how you can combine coefficients for different confounder levels.

2 Collapsibility and Noncollapsibility: Basic Concepts

Consider the following data adapted from [Greenland \(2005\)](#):

```
#table 1 from greenland
d <- data.frame(
```

```

z = c(1,1,1,1,0,0,0,0),
x = c(1,1,0,0,1,1,0,0),
y = c(1,0,1,0,1,0,1,0),
n = c(200,50,150,100,100,150,50,200)
)

d <- d %>%
  uncount(n)

```

These data lead to the following table of risks and association contrasts:

	Z = 1	Z = 0	Overall
$P(Y = 1)$ (Overall)	0.700	0.300	0.500
$P(Y = 1 X = 1)$	0.800	0.400	0.600
$P(Y = 1 X = 0)$	0.600	0.200	0.400
Risk Difference (RD)	0.200	0.200	0.200
Risk Ratio (RR)	1.333	2.000	1.500
Odds Ratio (OR)	2.667	2.667	2.250

This Table shows some important properties of the risk difference, risk ratio, and odds ratio. First, let's ensure we understand its elements. Table 1 shows the proportions of the outcome Y stratified by Z and overall (Marginal) for each level of X . Importantly, the data in this Table were generated from a model where Z did not affect the exposure X . The only causal relations are between the exposure X and the outcome Y , and the covariate Z and the outcome Y .

These proportions are used to then compute the risk differences, risk ratios, and odds ratios for the relation between X and Y overall, and within strata of Z .

Notice how these results differ between the Z strata and overall:

- The risk difference is 0.2 across stratum and overall. This is true even though the absolute risks differ across strata and overall. This is a general phenomenon associated with the risk difference. That is, across strata of *a variable that is not causally related to or associated with the exposure* the

risk difference is constant and equal to the overall risk difference. For this reason, we say that the **risk difference is strictly collapsible**.

- The risk ratio is 1.33 for the $Z = 1$ stratum and 2.00 for $Z = 0$. Furthermore, overall, the risk ratio is 1.5. While these three risk ratios are different, they are related in an important way. Note that the weighted average of the stratum specific risk ratios equal the overall risk ratio:

$$\begin{aligned} & \frac{P(Z = 1)P(Y = 1 \mid X = 1, Z = 1) + P(Z = 0)P(Y = 1 \mid X = 1, Z = 0)}{P(Z = 1)P(Y = 1 \mid X = 0, Z = 1) + P(Z = 0)P(Y = 1 \mid X = 0, Z = 0)} \\ &= \frac{0.5(0.80) + 0.5(0.40)}{0.5(0.60) + 0.5(0.20)} \\ &= 1.5 \end{aligned}$$

Thus, the information that is contained in the overall risk ratio is available in the stratum specific risk ratios, but must be transformed appropriately. For this reason, we say that the **risk ratio is collapsible**, because there is a function (weighted average) that allows us to “collapse” the stratum specific estimates to yield the overall estimate.

- The odds ratio is 2.67 in both strata created by Z . However, the overall odds ratio is less than the stratum specific odds ratios, with a value of 2.25. Thus, there is no weighted combination of the stratum specific estimates that yield the overall estimate. In fact, the stratum specific odds ratios seem to be capturing something completely different from the overall odds ratio. For this reason, we say that the **odds ratio is noncollapsible**.

Note also the direction: the stratum specific odds ratio is greater than the overall or marginally adjusted odds ratio. This turns out to be part of a general property: stratum specific odds ratios are always greater than or equal to the overall odds ratio.

In practice, we tend not to make a difference between strict collapsibility and collapsibility. Generally, we tend to classify risk differences and ratios as collapsible, and the odds ratio as non-collapsible. Additionally, there other measures of association that are non-collapsible. These include hazard ratios, rate ratios, and, depending on some nuanced circumstances, hazard, rate, and odds differences ([Daniel et al., 2021](#)).

3 Noncollapsibility in a Regression Context

In a regression context, there are some additional considerations that we'll demonstrate. First, note that the terminology that we used above referred to "stratum specific" and "overall". These are still relevant, but in a regression context, we can introduce distinctions between unadjusted, conditionally adjusted, and marginally adjusted associations.

To see these, consider the following data adapted from Greenland ([Greenland, 2005](#)). They are the two strata of the previous section with 500 people in each: Z affects Y but not X , the unexposed risks are 0.2 and 0.6, and the odds ratio is 2.67 in both strata.

```
# table 1 from greenland
d <- data.frame(
  z = c(1, 1, 1, 1, 0, 0, 0, 0),
  x = c(1, 1, 0, 0, 1, 1, 0, 0),
  y = c(1, 0, 1, 0, 1, 0, 1, 0),
  n = c(200, 50, 150, 100, 100, 150, 50, 200)
)

d <- d %>% uncount(n)
```

If we situate these data in a regression context, we can compute a range of different associations. An unadjusted association comes from a model regressing Y on X that does not include any other variables (in this case, Z) in the model.

A conditionally adjusted association comes from a model that regresses Y on X and Z , where the association measure is obtained by reading the output of the model summary, i.e., the estimated regression coefficient from the model.

A marginally adjusted association comes from a model that also regresses Y on X and Z , but where we obtain the association by averaging (or marginalizing) this association over the distribution of Z , in effect taking a weighted average in this case.²

There are situations where, completely independent of concerns regarding mediation, confounding, selection, or information bias, each of these three

² We'll soon see that there are a few ways to construct marginally adjusted estimates, including this outcome modeling approach that one might call marginal standardization, model based standardization, or the "g formula" for a time-fixed exposure, as well as inverse probability weighting and some double robust methods.

types of associations will be equivalent. Similarly, there are situations independent of the same concerns where these associations will be different.

3.1 Unadjusted Regression Models

Using these data, let's fit an unadjusted logistic regression model as:

```
mod_unadjusted <- glm(y ~ x, data = d,
                      family = binomial("logit"))
```

We can compute the association of interest by taking the exponent of the coefficient for the exposure from this model:

```
exp(summary(mod_unadjusted)$coefficients[2, 1])
```

```
## [1] 2.25
```

We can also compute the marginal association by taking the average of the predictions from this model under exposed and unexposed settings and constructing an odds ratio with them:

```
mu1 <- mean(predict(mod_unadjusted,
                    newdata = transform(d, x = 1),
                    type = "response"))
mu0 <- mean(predict(mod_unadjusted,
                    newdata = transform(d, x = 0),
                    type = "response"))

marg_OR <- (mu1 / (1 - mu1)) / (mu0 / (1 - mu0))

marg_OR
```

```
## [1] 2.25
```

Note that, in this case where there are no other variables in the model, the conditional and marginal odds ratio are equivalent.

3.2 Conditionally Adjusted Model

We can also adjust for Z in two ways. The first is in a conditionally adjusted model. To do this, we simply add Z to the unadjusted model, and then again read the coefficient for the exposure from the model:

```
mod_adjusted <- glm(y ~ x + z, data = d,
                    family = binomial("logit"))

exp(summary(mod_adjusted)$coefficients[2, 1])
```

```
## [1] 2.666667
```

Note that in this conditionally adjusted regression model, the odds ratio is 2.67, which is larger than the unadjusted coefficient. This is true even though Z is not a confounder, collider, or mediator (it is causally associated with the outcome, but not with the exposure), yielding a conditionally adjusted odds ratio that is larger than the unadjusted coefficient.

Note also that the estimates we get from these regression models are identical to the values we built the data with. This is due mainly to the simplicity of our example: with one binary exposure and one binary covariate, the model with a main effect for each is saturated except for the product term, and the data have a common stratum specific odds ratio, so nothing is lost.

This is noncollapsibility of the odds ratio.

3.3 Marginally Adjusted Model

What happens if we adjust our model for Z , and then deploy a marginally adjusted approach to compute the odds ratio? We could deploy marginal adjustment by taking predictions from this model under $X = 0$ and $X = 1$, and then contrasting the risks that arise:

```
# generate predictions and then average them
mu1 <- mean(predict(mod_adjusted,
                    newdata = transform(d, x = 1),
                    type = "response"))
```

```
mu0 <- mean(predict(mod_adjusted,
                    newdata = transform(d, x = 0),
                    type = "response"))

# compute OR based on these averages
marg_OR <- (mu1 / (1 - mu1)) / (mu0 / (1 - mu0))

marg_OR
```

```
## [1] 2.25
```

We can see that, even though our regression model includes the variable Z , the marginally adjusted odds ratio is once again 2.25. Note again that this marginally adjusted odds ratio can't be constructed from the coefficient of the same logistic regression model, which is 2.67.



Note: Conditional versus Adjusted

Unadjusted is often used as if synonymous with marginal, and adjusted as synonymous with conditional. Daniel et al. argue that these pairs should be kept apart: conditional and marginal describe the *estimand*, the quantity we are after, while adjusted and unadjusted describe the *analysis* (Daniel et al., 2021). The model in this section that includes Z is an adjusted analysis in both cases: reading its coefficient targets a conditional estimand, and averaging its predictions targets a marginal one. A marginally adjusted estimate is therefore not an unadjusted estimate, and in an observational study it is no more confounded than the conditional estimate from the same model. This is also why the labels we use in this course, conditionally adjusted and marginally adjusted, both carry the word adjusted.

4 Marginally and Conditionally Adjusted Regression: NHEFS

4.1 Conditional Adjustment

Let's see how some of these concepts play out in a more realistic setting. We'll focus on a few things related to fitting marginally and conditionally adjusted regression models to data. First, we'll explore the impact of non-collapsibility similar to what we did in the simpler example above. Next, we'll talk about the interpretation of parameters from marginally and conditionally adjusted

models. Finally, we'll discuss some benefits and tradeoffs in more complex settings, such as when there are exposure confounder interactions present in the model.

Suppose we wanted to use the NHEFS data to estimate the confounder adjusted effect of quitting smoking on weight change. Suppose further that we were interested specifically in whether an individual's weight change was greater than the median value in the data. The data, the outcome, and the adjustment set are those of Weeks 5 through 7: the 1566 people with a weight measurement at both visits, with marital status recoded so that the separated are pooled with the divorced and the one person of unknown status joins the married.

```
nhefs <- read_csv(here("data", "nhefs.csv")) %>%
  filter(!is.na(wt82_71)) %>%
  mutate(wt_delta = as.numeric(wt82_71 > median(wt82_71)),
         marital = ifelse(marital == 6, 5, marital),
         marital = ifelse(marital == 8, 2, marital))

# quick view of data
dim(nhefs)
```

```
## [1] 1566 65
```

```
table(nhefs$qsmk)
```

```
##
```

```
## 0 1
```

```
## 1163 403
```

A typical approach would be to use a generalized linear model to regress the indicator of weight change against the indicator of whether the individual quit smoking. We'd use this model to extract (i.e., read off) the coefficient from the model:

```
# modelForm is a regression argument that one can use to
# regress the outcome (wt_delta) against the exposure
# (qsmk) and selected confounders.
```

```
modelForm <- wt_delta ~ qsmk + age + sex + race +
  factor(exercise) + factor(marital) + smokeintensity
```

```
# This model can be used to quantify a conditionally
# adjusted odds ratio with correct standard error
```

```
modelOR <- glm(modelForm, data = nhefs,
  family = binomial("logit"))
```

```
# An example with broom::tidy
tidy(modelOR)[2, ]
```

```
## # A tibble: 1 x 5
##   term estimate std.error statistic    p.value
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>
## 1 qsmk      0.674      0.124      5.44 0.0000000533
```

```
exp(tidy(modelOR)[2, 2])
```

```
## # A tibble: 1 x 1
##   estimate
##   <dbl>
## 1      1.96
```

We can use a similar approach to obtain estimates of the risk ratio and risk difference. To compute the risk ratio, we adjust the link function and, as we saw in Weeks 6 and 7, the distribution: the log-binomial model of Week 6 needed starting values to converge, and the Poisson family with a log link and a robust variance is the usual workaround (Zou, 2004; Naimi and Whitcomb, 2020). The model-based standard error of this Poisson model is wrong for a binary outcome, which is why we pair it with the HC3 sandwich estimator from Week 7:

```

# This model can be used to quantify a conditionally
# adjusted risk ratio; the model-based standard error is
# INCORRECT, so we use the robust variance from Week 7
modelRR <- glm(modelForm, data = nhefs,
               family = poisson("log"))

round(coeftest(modelRR,
               vcov. = vcovHC(modelRR, type = "HC3"))[2, ], 4)

```

```

##      Estimate Std. Error    z value    Pr(>|z|)
##      0.3041     0.0524     5.8074     0.0000

```

```
exp(coef(modelRR)[2])
```

```

##      qsmk
## 1.355458

```

To compute the risk difference, we use a linear risk model, the parameters of which can be estimated with either ordinary least squares or maximum likelihood estimation. Its constant variance assumption is also wrong for a binary outcome, so again we use the robust variance:

```

# This model can be used to quantify a conditionally
# adjusted risk difference; using the lm() function fits
# with OLS
modelRD <- lm(modelForm, data = nhefs)

# using the glm() function with gaussian and identity fits
# with MLE. check for yourself: same results
# modelRD <- glm(modelForm, data = nhefs,
#               family = gaussian(link = "identity"))

round(coeftest(modelRD,
               vcov. = vcovHC(modelRD, type = "HC3"))[2, ], 4)

```

```
##      Estimate Std. Error    t value    Pr(>|t|)
##      0.1587     0.0284     5.5820     0.0000
```

Let's quickly summarize these results from our conditionally adjusted regression models:

Contrast	Conditional Estimate
Odds Ratio	1.963
Risk Ratio	1.355
Risk Difference	0.159

4.2 Marginal Adjustment

Another approach to obtaining association measures from GLMs is to use marginal standardization, which is equivalent to g computation (aka the parametric g formula) when the exposure is measured at a single time point (Naimi et al., 2017; Snowden et al., 2011). This process gives us marginally adjusted association measures. There are several approaches we can use, but the easiest will be to start with fitting a single logistic model, regressing the binary outcome against the exposure and all confounder variables, and then generating predictions from this model. To obtain standard errors, the entire procedure is usually bootstrapped.

Here is some code to implement this marginal standardization in the NHEFS data:

```
# Regress the outcome against the exposure and confounders
ms_model <- glm(modelForm, data = nhefs,
                family = binomial("logit"))

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

c(OR = marg_stand_OR, RR = marg_stand_RR,
  RD = marg_stand_RD)

```

```

##          OR          RR          RD
## 1.8994323 1.3442877 0.1580764

```

The bootstrap of Week 7 gives us confidence intervals. The function returns all three contrasts, and we ask `boot.ci()` for the bias-corrected and accelerated interval of each:

```

bootfunc <- function(data, index){
  boot_dat <- data[index, ]
  ms_model <- glm(modelForm, data = boot_dat,
    family = binomial("logit"))
  mu1 <- predict(ms_model,
    newdata = transform(boot_dat, qsmk = 1),
    type = "response")
  mu0 <- predict(ms_model,
    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)

c(marg_stand_RD_, marg_stand_RR_, marg_stand_OR_)
}

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

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

boot_RD

```

```

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, type = "bca", index = 1)
##
## Intervals :
## Level      BCa
## 95%      ( 0.1058, 0.2189 )
## Calculations and Intervals on Original Scale

```

```
boot_RR
```

```

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, type = "bca", index = 2)
##
## Intervals :

```

```
## Level          BCa
## 95%   ( 1.222,  1.494 )
## Calculations and Intervals on Original Scale

boot_OR

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, type = "bca", index = 3)
##
## Intervals :
## Level          BCa
## 95%   ( 1.527,  2.450 )
## Calculations and Intervals on Original Scale
```

This marginal standardization approach can yield an estimate of the marginally adjusted odds ratio, risk ratio, and risk difference, and the bootstrap yields a correct standard error. Importantly, there are some subtle distinctions between marginally and conditionally adjusted effects that make their interpretation different.

Collapsibility / non-collapsibility can cause some differences to occur with the odds ratio. Here's the summary of all our contrasts and we can see how non-collapsibility of the odds ratio is influencing our estimates.

Contrast	Conditional Estimate	Marginal Estimate
Odds Ratio	1.963	1.899
Risk Ratio	1.355	1.344
Risk Difference	0.159	0.158

We can see here that the estimated odds ratios are different, but the risk ratios and risk differences are pretty much the same. Note, however, that the risk ratio and risk difference columns come from different models: the conditional values are read from a log-link and an identity-link model, while the marginal values are standardized from the logistic model, so their near agreement reflects that the three models describe these data similarly, as we saw in Week 6. The cleaner demonstration of collapsibility is within a single

model. Standardizing the Poisson model returns its own coefficient exactly, as does standardizing the linear model, whereas standardizing the logistic model does not:

```
standardize <- function(model, data = nhfs){
  mu1 <- predict(model,
                  newdata = transform(data, qsmk = 1),
                  type = "response")
  mu0 <- predict(model,
                  newdata = transform(data, qsmk = 0),
                  type = "response")
  c(OR = (mean(mu1) / mean(1 - mu1)) /
      (mean(mu0) / mean(1 - mu0)),
    RR = mean(mu1) / mean(mu0),
    RD = mean(mu1) - mean(mu0))
}
```

```
# log link: standardized RR versus exp(coefficient)
c(standardize(modelRR) ["RR"], exp(coef(modelRR) [2]))
```

```
##      RR      qsmk
## 1.355458 1.355458
```

```
# identity link: standardized RD versus coefficient
c(standardize(modelRD) ["RD"], coef(modelRD) [2])
```

```
##      RD      qsmk
## 0.1586763 0.1586763
```

```
# logit link: standardized OR versus exp(coefficient)
c(standardize(modelOR) ["OR"], exp(coef(modelOR) [2]))
```

```
##      OR      qsmk
## 1.899432 1.962751
```

This is the CCF at work. With a log link and no interaction, every person's exposed prediction is $\exp(\hat{\beta}_1)$ times their unexposed prediction, so the ratio of

averages is the common ratio; with an identity link, every person's difference is $\hat{\beta}_1$, so the average difference is the common difference. With a logit link, the person-level odds ratios are all $\exp(\hat{\beta}_1)$, but the odds ratio of the averaged risks is not.

If there were interactions present in the model between the exposure and the confounders, things get a little messy, and we address this scenario next.

The fact that we have an outcome with a prevalence of 50% makes it so that the odds ratio is not a good approximation to the risk ratio, and non-collapsibility is in full effect.

A natural follow up question is regarding how to interpret the conditionally versus marginally adjusted odds ratio? If they are different, does this mean that they have a different interpretation? The answer is generally yes, but the specific interpretations can get confusing, at least for the conditionally adjusted OR.

Interpreting the marginally adjusted odds ratio is easier, and aligns quite well with the code we used, particularly if the assumptions for causal inference hold. If these assumptions (Weeks 1 and 4) hold, we can interpret this measure as the odds of greater than median weight change that would be observed if everyone in the population quit smoking, relative to the odds that would be observed if no one quit smoking. If the causal assumptions don't hold, or we're not interested in a causal interpretation per se, we could say that the marginally adjusted odds ratio captures the odds of greater than median weight change among those who quit smoking versus those who didn't, standardized to the distribution of confounders in the sample.

Conceptually, what we get when we use marginal standardization is individual level probability estimates, averaged over all individuals in the sample, and then formulated into an odds ratio. That is, we get predictions under different "quit smoking" scenarios for each individual, average those predictions, and construct an odds ratio. What about the conditionally adjusted contrast we estimated above? Conceptually, we can think of this measure as getting individual level predictions for each "quit smoking" scenario, constructing an odds ratio using these predictions, and then averaging these individual level odds ratios. Because the odds ratio is a non-collapsible measure, the odds ratio of averaged individual level predictions is not the same as the average of individual odds ratios from individual level predictions.³

³ This is Jensen's inequality again: the two orders of operation agree only when the CCF is linear.

But what does this mean for us as applied researchers? Which one should we choose to report? Arguments can be made for both, but my preference is to report marginally adjusted associations. These can be, arguably, more easily aligned to population-level interventions, and in more complex settings (such as we will see below), can be much easier to construct. Furthermore, even when we are interested in conditionally adjusted associations, such as the association among those with certain characteristics, one can argue that marginally adjusted associations within levels of this variable are easier to interpret.⁴

Overall, for non-collapsible measures of association, marginal adjustment solves a lot of interpretational problems.

⁴ A corollary from this week's reading concerns randomized trials. In a trial, adjustment for baseline covariates is not needed for validity, and it is often said that adjustment costs precision because the adjusted (conditional) log odds ratio has a



Note: Marginal Standardization, the g Formula, and Software

Week 4 wrote the g formula for a point exposure as $E(Y^x) = \sum_z E(Y \mid X = x, Z = z)P(Z = z)$ (Robins, 1986; Hernán and Robins, 2020). Marginal standardization is this formula with the conditional mean replaced by a fitted regression model and the distribution of Z replaced by the empirical distribution of the sample, which is why it is also called model based standardization, the parametric g formula, or g computation (Naimi et al., 2017; Snowden et al., 2011). The predicted risks can be averaged over any reference distribution, not only the sample's; different choices correspond to different target populations (Muller and MacLehose, 2014). In this week's reading, the same estimator appears as Zhang's method for recovering a marginal odds ratio from a conditional logistic model (Zhang, 2009; Daniel et al., 2021), and it is what the `margins` command in Stata and the `marginaleffects` package in R compute.

5 Marginal and Conditional Adjustment with Interactions

There is another scenario where the use of marginal adjustment is preferable to the use of conditional adjustment, and this is when there are important interactions between the exposure and other variables in the model, but we are not necessarily interested in reporting all of those interactions, but are rather interested in the association in the whole population (i.e., the marginal association). The no-interaction model above assumes a constant conditional odds ratio for quitting across levels of all of the other variables in the model. Let's consider a new logistic regression model with two interactions that we deem are important:

```

modellInt <- wt_delta ~ qsmk + age + sex + race +
  factor(exercise) + factor(marital) + smokeintensity +
  qsmk:sex + qsmk:smokeintensity

conditional_model <- glm(modellInt, data = nhefs,
  family = binomial("logit"))

round(summary(conditional_model)$coefficients, 4)

```

##	Estimate	Std. Error	z value
## (Intercept)	1.5087	0.2743	5.5005
## qsmk	0.6437	0.2874	2.2400
## age	-0.0366	0.0048	-7.6851
## sex	0.0224	0.1268	0.1768
## race	-0.1903	0.1626	-1.1706
## factor(exercise)1	-0.0812	0.1458	-0.5571
## factor(exercise)2	-0.2288	0.1511	-1.5140
## factor(marital)3	0.2551	0.2398	1.0638
## factor(marital)4	-0.0140	0.2241	-0.0625
## factor(marital)5	-0.0623	0.1837	-0.3392
## smokeintensity	0.0022	0.0055	0.4077
## qsmk:sex	0.1103	0.2533	0.4355
## qsmk:smokeintensity	-0.0011	0.0103	-0.1066
##	Pr(> z)		
## (Intercept)	0.0000		
## qsmk	0.0251		
## age	0.0000		
## sex	0.8597		
## race	0.2418		
## factor(exercise)1	0.5775		
## factor(exercise)2	0.1300		
## factor(marital)3	0.2874		
## factor(marital)4	0.9502		
## factor(marital)5	0.7344		
## smokeintensity	0.6835		

```
## qsmk:sex          0.6632
## qsmk:smokeintensity 0.9151
```

If we wanted to compute the association between quitting smoking and greater than median weight change in the entire population, we could do it with the coefficients from this model by combining them appropriately, but it would be a considerable undertaking (especially since we'd have to integrate over the distribution of the continuous smoking intensity measurement).

Fortunately, we could do this easily with marginal standardization (the bonus material at the end of these notes shows what the conditional route requires even for a single binary modifier). We do it here for the risk difference, but the same can be done for odds and risk ratios:

```
# Regress the outcome against the confounders and
# interactions
ms_model <- glm(modelInt, data = nhefs,
                family = binomial("logit"))

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

# Marginally adjusted risk difference
marg_stand_RD <- mean(mu1) - mean(mu0)

# Using the bootstrap to obtain confidence intervals for
# the marginally adjusted risk difference.
bootfunc <- function(data, index){
  boot_dat <- data[index, ]
  ms_model <- glm(modelInt, data = boot_dat,
                  family = binomial("logit"))
```

```

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

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

boot_RD <- boot.ci(boot_res, type = "bca")

marg_stand_RD

```

```
## [1] 0.1587901
```

```
boot_RD
```

```

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, type = "bca")
##
## Intervals :
## Level      BCa
## 95%      ( 0.1053, 0.2192 )
## Calculations and Intervals on Original Scale

```

6 Marginal Standardization with a Stratified Outcome Model

6.1 Interaction Terms: Review

One last consideration is how we might include more interactions between our exposure and all the variables in our model, without actually having to include the interactions manually. This is a useful trick to be aware of, particularly because some more advanced modeling approaches (e.g., some programs fitting double robust estimators) rely on this technique. Therefore, it's useful for you to know how to deploy it, and why it works.

Let's review some basics first. Suppose we fit the following linear regression model:

$$E(Y | A, W) = \alpha_0 + \alpha_1 A + \alpha_2 W + \alpha_3 AW$$

This model allows us to estimate an association between A and Y that differs for different values of W because of the interaction between A and W . For example, if A is binary, the $W = w$ specific association between A and Y is:

$$E(Y | A = 1, W = w) - E(Y | A = 0, W = w) = (\alpha_1 + \alpha_3 w)$$

Basically, we can think of this as allowing for a different parameter depending on the value of W . Continuing the example, let's assume that $\alpha_1 = 0.2$ and $\alpha_3 = -0.05$. We can define a new parameter:

$$\phi(w) = E(Y | A = 1, W = w) - E(Y | A = 0, W = w)$$

So we have $\phi(w = 0.5) = (0.2 - 0.05 \times 0.5) = 0.175$.

6.2 Exposure Stratification and Interaction

Now, imagine a scenario where instead of having a single α_1 and α_3 that we can use to capture the association between A and Y , we have two A specific coefficient values for the W variable. For instance, we might have a model stratified by the exposure A yielding four different parameters:

$$E(Y | A = 1, W) = \beta_0 + \beta_1 W$$

and

$$E(Y \mid A = 0, W) = \gamma_0 + \gamma_1 W$$

From these two models we could compute the $W = w$ specific association between A and Y as:

$$E(Y \mid A = 1, W = w) - E(Y \mid A = 0, W = w) = (\beta_0 + \beta_1 w) - (\gamma_0 + \gamma_1 w)$$

Continuing the example, let's assume that $\beta_0 = 0.4$ and $\beta_1 = 0.05$, and $\gamma_0 = 0.275$ and $\gamma_1 = -0.05$. We again can define a new parameter: $\phi(w = 0.5) = (0.4 + 0.05 \times 0.5) - (0.275 - 0.05 \times 0.5) = 0.175$, which is the same as the previous estimate.

However, notice that because we are using all the parameters in each stratified model, we can actually just compute the predictions and take the difference between those predictions. We'd get the same result for all individuals whose value of $W = 0.5$.

This latter procedure, based on the two stratified models, can be generalized to a host of different variables. In R, we can deploy this procedure as follows:

```
# 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 formula
# 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.
modelStrat <- wt_delta ~ age + sex + race +
  factor(exercise) + factor(marital) + smokeintensity

# Regress the outcome against the confounders among the
# unexposed (model0) and then among the exposed (model1)
model0 <- glm(modelStrat, data = subset(nhefs, qsmk == 0),
```

```

    family = binomial("logit"))
model11 <- glm(modelStrat, 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(model11, newdata = nhefs, type = "response")
mu0 <- predict(model10, 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
bootfunc <- function(data, index){
    boot_dat <- data[index, ]
    model0 <- glm(modelStrat,
        data = subset(boot_dat, qsmk == 0),
        family = binomial("logit"))
    model1 <- glm(modelStrat,
        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)

c(marg_stand_RD_, marg_stand_RR_, marg_stand_OR_)
}

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

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

c(OR = marg_stand_OR, RR = marg_stand_RR,
  RD = marg_stand_RD)

```

```

##          OR          RR          RD
## 1.9458337 1.3561060 0.1637278

```

```
boot_RD
```

```

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, type = "bca", index = 1)
##
## Intervals :
## Level      BCa
## 95%      ( 0.1078,  0.2234 )
## Calculations and Intervals on Original Scale

```

```
boot_RR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, type = "bca", index = 2)
##
## Intervals :
## Level      BCa
## 95%      ( 1.228,  1.506 )
## Calculations and Intervals on Original Scale
```

```
boot_OR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, type = "bca", index = 3)
##
## Intervals :
## Level      BCa
## 95%      ( 1.546,  2.517 )
## Calculations and Intervals on Original Scale
```

The difference between this code and our simple example with just four parameters is that, instead of generating predictions for only those individuals with certain values of the confounding variables, we generate predictions for each individual in our sample, and then average over the predictions.

There's a cautionary note that should be considered here. This procedure works well under two related conditions. The first is that we have sufficient data to be able to fit the stratified models well. For instance, suppose we had 10,000 individuals and an exposure with a prevalence of 50%. This means that we have 5,000 observations to fit both models. Unless we have a ridiculously large number of confounders, we would likely be able to fit these models

well. However, if we had only 100 individuals, splitting this sample across the exposure would likely yield too few observations to fit the models well.

Another consideration is the exposure prevalence. If there is a large imbalance in the exposure, it may be that one stratum has too few observations to fit a stratified model well, whereas the other can fit the model quite well. In our NHEFS data, this is an important consideration:

```
prop.table(table(nhefs$qsmk))
```

```
##
##           0           1
## 0.7426564 0.2573436
```

```
table(nhefs$qsmk)
```

```
##
##      0      1
## 1163   403
```

These tables show that there are 403 individuals (26%) available to fit the model among the exposed in these data.

7 Propensity Scores and Inverse Probability Weighting

7.1 Propensity Score Methods

So far, we've focused on obtaining quantitative estimates of the average treatment effect using an outcome modeling approach. In this approach, one regresses the outcome against the exposure and confounders. This works for the conditionally adjusted estimator, where we read the coefficient for the exposure from the regression model, or the marginally adjusted estimator (e.g., using marginal standardization). The basic formulation of this outcome modeling approach can be depicted heuristically using the DAG in panel (b) of the figure below.

In this figure, the open back-door path from X to Y is blocked by conditioning on C . This leads to a conditionally adjusted estimate of the exposure

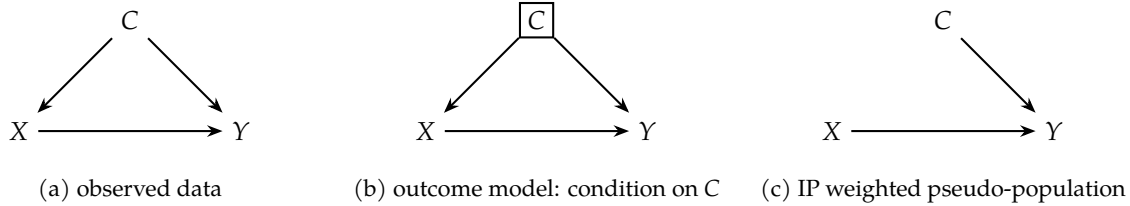


Figure 1: Directed acyclic graphs depicting confounder adjustment. (a) The observed data, in which the covariate C affects both the exposure X and the outcome Y , so the back-door path from X to Y through C is open. (b) Outcome model adjustment conditions on C (box), which blocks the back-door path and renders X d-separated from Y given C ; the conditionally and marginally adjusted estimates of this lecture both come from this model. (c) The pseudo-population created by inverse probability weighting, in which C no longer affects X , so the crude X to Y contrast in the weighted data is unconfounded.

effect. One can marginalize over the distribution of C to get a marginally adjusted estimate from a model for the outcome. For a binary outcome, such estimates may be obtained on the risk difference, risk ratio, or odds ratio scales, or one may obtain a marginally adjusted estimate of the cumulative risk function that would be observed if everyone were exposed or unexposed.

On the other hand, we may want to obtain an adjusted estimate of the exposure effect by modeling the exposure. This may be the case when the event under study (e.g., mortality, birth defect, heart attack) is very rare in the sample, and there are a large number of confounders that we must adjust for. In this case, it may be preferable to limit the number of confounders in the outcome model, and adjust for confounding by modeling the exposure. This can be accomplished using propensity score methods.

The propensity score was first defined by Rosenbaum and Rubin ([Rosenbaum and Rubin, 1983](#)) as the conditional probability of receiving the treatment (or, equivalently, of being exposed). The true propensity score is defined as

$$e(C) = P(X = 1 \mid C)$$

This propensity score is typically known in a randomized trial. It's important to distinguish this true PS from the estimated PS:

$$\hat{e}(C) = \hat{P}(X = 1 \mid C)$$

This estimated PS can be fit using a parametric model, in which case, we might write:⁵

$$\hat{e}(C; \alpha) = \hat{P}(X = 1 \mid C) = \text{expit}(\alpha C)$$

For reasons that are not entirely obvious, it is usually better to use the estimated PS, even when the true PS is known ([Robins et al., 1992](#); [Henmi and Eguchi, 2004](#)). For example, in the context of a randomized trial, the true

⁵ Recall, the $\text{expit}(a)$ function is the inverse of the $\text{logit}(a)$ function, defined as $1/[1 + \exp(-a)]$.

propensity score is actually known exactly, and is equal to the probability of receiving treatment. However, the performance of a propensity score based estimator (in terms of statistical efficiency, which determines, among other things, the size of the standard errors) will be better after using an estimated propensity score, even in this setting.

If the set of conditioning variables consists of the relevant confounding variables, the propensity score can be used to invoke conditional exchangeability. To see why, consider the case where the probability of being exposed is conditional on two binary confounders:

$$e(C) = P(X = 1 | C) = \text{expit}(\alpha_0 + \alpha_1 C_1 + \alpha_2 C_2 + \alpha_3 C_1 C_2)$$

Consider that, for two binary confounders, there are four possible joint confounder levels:

C_1	C_2	$P(X = 1 C)$
0	0	$\text{expit}(\alpha_0)$
1	0	$\text{expit}(\alpha_0 + \alpha_1)$
0	1	$\text{expit}(\alpha_0 + \alpha_2)$
1	1	$\text{expit}(\alpha_0 + \alpha_1 + \alpha_2 + \alpha_3)$

Notice also that there are four parameters in the above model, one parameter for each level. This implies that, for this simple example, there is a unique propensity score value for each unique confounder level. In other words, the one-dimensional propensity score in this example contains all the information available in the two-dimensional set of confounders. We can thus reduce the complexity of the set of confounders to a single variable, and adjust for this variable instead of the confounders. For example:

$$E[Y | X, e(C)] = \beta_0 + \beta_1 X + \beta_c e(C)$$

Here $e(C)$ is a proxy for all the variables C . Thus, we can use $e(C)$ to replace C in an outcome regression model, with the caveat that the coefficient β_1 is then conditional on the propensity score, which for a noncollapsible measure is yet another conditional estimand, different from the one conditional on C itself (Daniel et al., 2021).

As a univariate proxy for the multivariate set of confounders, we can use the propensity score to adjust for confounding using a number of different

techniques. These techniques include regression adjustment, propensity score matching, stratification, and weighting. In this lecture, we focus on the creation and use of IP-weights. The reason for this emphasis is that (i) in epidemiology, the most commonly used PS technique is inverse probability weighting, and so it is important to know how to implement; and (ii) weighting targets the same marginal estimand as marginal standardization, so the two approaches can be compared directly.

7.2 Inverse Probability Weighting: Intuition

The most commonly employed propensity score adjustment technique is inverse probability weighting. The simple heuristic often used to describe the way IP-weighting works is that, when applied to data, they yield a “pseudo-population” where there is no longer an effect of the confounder on the exposure. The causal structure of the variables in this new pseudo-population is depicted in panel (c) of the figure above.

The weights for each individual needed to create this pseudo-population are defined as the inverse of the probability of receiving their observed exposure. Let’s consider the following simple example to explain why this works. In this example, there are 20 observations, with one binary confounder (50:50 split) and one binary exposure. Let’s suppose the probability that the exposure $x = 1$ is 0.2 for those with $c = 0$ and 0.9 for those with $c = 1$. Under these scenarios, we’d have exposure and confounder data that looks like this:

```
c <- c(0, 0, 1, 1)
x <- c(1, 0, 1, 0)
n <- c(2, 8, 9, 1)
```

```
tibble(x, c, n)
```

```
## # A tibble: 4 x 3
##       x     c     n
##   <dbl> <dbl> <dbl>
## 1     1     0     2
## 2     0     0     8
## 3     1     1     9
```

```
## 4      0      1      1
```

Let's focus on the stratum of individuals with $c = 0$. In this stratum, there are 2 exposed individuals, and 8 unexposed individuals. Now let's ask what these data should look like if we were able to implement an ideal (marginally) randomized trial with the probability of treatment being 50% for all individuals. We would expect that within the stratum of individuals with $c = 0$, there would be 5 exposed and 5 unexposed individuals. Inverse probability weighting seeks to accomplish this balance.

Consider that the probability of the observed exposure among those with $c = 0$ is 0.2 for those who were actually exposed, and 0.8 for those who were unexposed. The inverse probability weight is thus $\frac{1}{0.2} = 5$ for the exposed in this confounder stratum, and $\frac{1}{0.8} = 1.25$ for the unexposed in this confounder stratum.

This suggests that, in their contribution to the overall analysis, the two exposed individuals in the $c = 0$ stratum would each receive a weight of 5, while the eight unexposed individuals in the $c = 0$ stratum would each receive a weight of 1.25. Under these conditions, we would have a re-balanced set of observations in the $c = 0$ stratum of:

$$\text{Exposed Observations: } 5 \times 2 = 10$$

$$\text{Unexposed Observations: } 8 \times 1.25 = 10$$

In effect, our inverse probability weighting strategy made it such that we now have an equal number of exposed and unexposed observations within the $c = 0$ stratum. The pseudo-population is twice the size of the stratum, but what matters is that exposure is now balanced within it, and the same happens in the $c = 1$ stratum. In these weighted data, we can now compute the difference in the outcome among the exposed and unexposed individuals (if we had it) to obtain an estimate of our average treatment effect. Note that this is standardization to the total population: the weighted exposed group and the weighted unexposed group each have the covariate distribution of the whole sample, which is why IP weighting delivers a marginal contrast ([Sato and Matsuyama, 2003](#); [Hernán and Robins, 2006](#)).

7.3 Inverse Probability Weighting in Practice

Simply taking the inverse of the probability of the observed exposure, while valid, is not the usual strategy for implementing inverse probability weights. In practice, one will often use stabilized inverse probability weights, potentially with some degree of “truncation” or, more accurately, trimming of the weights to reduce the impact of potential outliers in the weights.⁶

Furthermore, it’s important to note that “data” are often not weighted, but rather the contribution that each individual in the sample makes to the estimating function (e.g., likelihood, estimating equation, or other function used to find parameters). This is important in that one must choose a fitting algorithm that allows for this type of weighting.

To start, the simplest type of weight used in practice is the stabilized inverse probability weight. These are often defined as:

$$sw = \begin{cases} \frac{P(X = 1)}{P(X = 1 | C)} & \text{if } X = 1 \\ \frac{P(X = 0)}{P(X = 0 | C)} & \text{if } X = 0 \end{cases}$$

but are sometimes written more succinctly as:⁷

$$sw = \frac{f(X)}{f(X | C)}$$

Let’s use the NHEFS data again to construct the stabilized weights. We fit the PS model to the data, with the same confounders as the outcome models above, and construct the weights:

```
# create the propensity score in the dataset
psForm <- qsmk ~ age + sex + race + factor(exercise) +
  factor(marital) + smokeintensity

ps_model <- glm(psForm, data = nhefs,
  family = binomial("logit"))

nhefs$propensity_score <- ps_model$fitted.values

# stabilized inverse probability weights
```

⁶ In contrast to our usage of the word “truncation”, which refers to the removal of observations from the dataset, researchers will often refer to “truncating” the weights, which sets the largest value to be equal to the 99th or 95th percentile values. This is more accurately referred to as “trimming” the weights, since no truncation is occurring.

⁷ This formulation is unusual, since $f(\cdot)$ represents the probability density function, which is usually taken for a specific realization of the random variable. However, in this case, because the weights are defined as a function of the observed exposure status, the argument in the operator is the observed data (denoted with capital letter), as opposed to some specific realization.

```

nhefs$sw <- (mean(nhefs$qsmk) / nhefs$propensity_score) *
  nhefs$qsmk +
  ((1 - mean(nhefs$qsmk)) / (1 - nhefs$propensity_score)) *
  (1 - nhefs$qsmk)

summary(nhefs$sw)

```

```

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.4215  0.8859  0.9713  0.9991  1.0781  4.2208

```

```

nhefs %>%
  select(wt_delta, qsmk, sex, age, propensity_score, sw) %>%
  print(n = 5)

```

```

## # A tibble: 1,566 x 6
##   wt_delta qsmk  sex  age propensity_score  sw
##   <dbl> <dbl> <dbl> <dbl>          <dbl> <dbl>
## 1      0     0    0   42            0.155 0.879
## 2      1     0    0   36            0.234 0.970
## 3      1     0    1   56            0.147 0.870
## 4      1     0    0   68            0.360 1.16
## 5      1     0    0   40            0.320 1.09
## # i 1,561 more rows

```

As we can see from the output above, the stabilized weights are, in fact, well behaved, with a mean of one and a max value that is small. We can then fit the weighted crude model to obtain the risk difference, and, with the log and logit links, the risk ratio and odds ratio:

```

model_RD_weighted <- glm(wt_delta ~ qsmk, data = nhefs,
  weights = sw,
  family = quasibinomial("identity"))

summary(model_RD_weighted)$coefficients

```

```
##           Estimate Std. Error  t value
## (Intercept) 0.4602361 0.01462014 31.47959
## qsmk        0.1545945 0.02833959  5.45507
##           Pr(>|t|)
## (Intercept) 6.732231e-169
## qsmk        5.682334e-08
```

```
model_RR_weighted <- glm(wt_delta ~ qsmk, data = nhefs,
                          weights = sw,
                          family = quasipoisson("log"))

model_OR_weighted <- glm(wt_delta ~ qsmk, data = nhefs,
                          weights = sw,
                          family = quasibinomial("logit"))

ipw_est <- c(OR = exp(coef(model_OR_weighted)[2]),
             RR = exp(coef(model_RR_weighted)[2]),
             RD = coef(model_RD_weighted)[2])

ipw_est
```

```
##   OR.qsmk   RR.qsmk   RD.qsmk
## 1.8720903 1.3359025 0.1545945
```

Note the `quasibinomial` and `quasipoisson` families: with non-integer weights, the `binomial` family issues a warning about non-integer successes, while the quasi families fit the same mean model without complaint. Their model-based standard errors are not the ones we will use anyway.

At times, the mean and max of the stabilized weights are sub-optimal, in that the mean may not be one, or the max may be too large for comfort. More commonly, researchers will “trim” the weights to avoid problems induced by very large weights. The procedure for doing this requires simply replacing all weight values greater than a certain percentile with their percentile values:

```
quantile(nhefs$sw, 0.99)
```

```
##      99%
```

```
## 1.762022
```

```
nhefs <- dhefs %>%
  mutate(sw_trim = if_else(sw > quantile(sw, 0.99),
                             quantile(sw, 0.99), sw))
```

```
summary(nhefs$sw)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.4215  0.8859  0.9713  0.9991  1.0781  4.2208
```

```
summary(nhefs$sw_trim)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.4215  0.8859  0.9713  0.9941  1.0781  1.7620
```

When trimming weights, it's often best to start at the highest percentile (e.g., 99th percentile) and work your way down until the mean of the weights is close enough to 1 and the max weight is not too large. This strategy of starting with the highest percentile allows us to avoid the brunt of the potential bias that results from trimming. In fact, the key idea behind weight trimming is to optimize the bias-variance tradeoff being made (Cole and Hernán, 2008).

A final note that is important with these weighted approaches is to consider how to estimate the standard errors. In fact, the model-based standard errors are no longer valid when weighting is used. One must instead use the robust variance estimator, or the bootstrap. The `coeftest` function in the `lmtest` package, which relies on the `sandwich` function without making an explicit call to the `sandwich` package, gives the robust version in a clean format:

```
coeftest(model_RD_weighted,
         vcov. = vcovHC(model_RD_weighted, type = "HC0"))
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.460236   0.014722 31.2614 < 2.2e-16 ***
## qsmk        0.154594   0.029945  5.1625 2.436e-07 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

One can then construct CIs in the standard way using the estimated standard error in the output above. However, this estimator does not account for the fact that the propensity score model is estimated. Because the propensity score was estimated by maximum likelihood, the robust standard error that treats the weights as known is conservative for the average treatment effect ([Robins et al., 1992](#); [Lunceford and Davidian, 2004](#)). Better would be to use a bootstrap procedure that resamples data, fits the propensity score, and estimates the weighted association measure:

```
boot_ipw <- function(data, index){
  boot_dat <- data[index, ]
  ps <- glm(psForm, data = boot_dat,
            family = binomial("logit"))$fitted.values
  boot_dat$sw <- (mean(boot_dat$qsmk) / ps) *
    boot_dat$qsmk +
    ((1 - mean(boot_dat$qsmk)) / (1 - ps)) *
    (1 - boot_dat$qsmk)
  mod <- glm(wt_delta ~ qsmk, data = boot_dat,
            weights = sw,
            family = quasibinomial("identity"))
  coef(mod)[2]
}

set.seed(123)
boot_res <- boot(nhefs, boot_ipw, R = 2000)
```

```
round(c(robust_SE = sqrt(vcovHC(model_RD_weighted,
                               type = "HCO")[2, 2]),
       bootstrap_SD = sd(boot_res$t)), 4)
```

```
##      robust_SE bootstrap_SD
##      0.0299      0.0302
```

```
boot.ci(boot_res, type = "bca")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, type = "bca")
##
## Intervals :
## Level      BCa
## 95%      ( 0.0945, 0.2124 )
## Calculations and Intervals on Original Scale
```

We now have three adjusted estimates of each contrast from the same confounder set: the conditional estimate read from the outcome model, the marginal estimate standardized from the outcome model, and the marginal estimate from the weighted exposure model.

Contrast	Conditional	Standardized	IP weighted
Odds Ratio	1.963	1.899	1.872
Risk Ratio	1.355	1.344	1.336
Risk Difference	0.159	0.158	0.155

Three adjusted estimates from one confounder set. The standardized and IP weighted estimates are both marginal and target the same estimand, and they agree closely here; the conditional odds ratio is further from the null than either, as noncollapsibility predicts, while the conditional risk ratio and risk difference are close to their marginal counterparts.


Note: Standardization or Weighting?

Both routes deliver a marginally adjusted estimate from the same confounder set, and both require the identification assumptions of Weeks 1 and 4; what differs is which nuisance function is modeled. Marginal standardization models the outcome given exposure and confounders, and will be biased if that model is misspecified. IP weighting models the exposure given confounders, and will be biased if the propensity score model is misspecified. Weighting is attractive when the outcome is rare and the confounders many, since the exposure model then has far more events to work with. In observational data, standardization is in general the more efficient of the two ([Daniel et al., 2021](#)), but efficiency is of little comfort if the outcome model is wrong. Double robust estimators, which we touch on later in the course, combine both nuisance models so that the estimate is consistent if either one is correct.

8 Conclusion

Estimate	What is modeled	How the contrast is obtained	Standard error
Conditionally adjusted	outcome given exposure and confounders	read the exposure coefficient	model-based (if the variance function is right) or robust
Marginally adjusted, standardization	outcome given exposure and confounders	average predictions under $X = 1$ and $X = 0$, then contrast	bootstrap the whole procedure
Marginally adjusted, IP weighting	exposure given confounders	crude contrast in the weighted data	robust (conservative) or bootstrap including the PS fit

Three ways to adjust. The first two share an outcome model and differ only in what is done with it; the third models the exposure instead. The two marginal routes target the same estimand.

- Non-collapsibility can occur in settings where we'd like to estimate an exposure-outcome association in the presence of a covariate that is associated with the outcome.
- The risk difference and the risk ratio are collapsible measures of association: in the presence of a non-confounding cause of the outcome, the overall measure is a weighted average of the stratum specific measures, and equals them when they are homogeneous. The characteristic collapsibility function $g_v(p) = f^{-1}\{f(p) + v\}$ is linear for the identity and log links, which is why.
- The odds ratio is a non-collapsible measure of association: its CCF bends, and in the presence of a non-confounding cause of the outcome, the stratum specific odds ratios are further from the null than the overall odds ratio.

- In a regression context, we can obtain marginally or conditionally adjusted risk differences, risk ratios, or odds ratios. Conditional and marginal describe the estimand; adjusted and unadjusted describe the analysis.
- Conditionally adjusted association measures can be obtained by reading coefficients from the regression model fit.
- Marginally adjusted association measures can be obtained by obtaining predictions from the model under different exposure settings, and then averaging and contrasting those predictions, or by weighting the data by the inverse probability of the observed exposure and contrasting the weighted outcomes.
- Collapsibility can cause the conditionally adjusted and marginally adjusted odds ratios to differ, but not conditionally versus marginally adjusted risk ratios or risk differences from the same model.
- In addition to collapsibility or non-collapsibility, conditionally and marginally adjusted association measures can differ in the presence of exposure-covariate interactions.
- When interest lies in the overall association, fitting marginally adjusted associations can facilitate interpretation.
- The bootstrap can easily be deployed to obtain uncertainty measures such as confidence intervals when marginal standardization or IP weighting is used; for IP weighting, the robust variance that treats the weights as known is a conservative alternative.
- Incorporating interactions between the exposure and all covariates in the model can easily be accomplished using a stratified modeling approach. However, care is needed to ensure appropriate sample sizes are present in both exposure strata.

Bonus Material

The two subsections below extend the main text. The first shows what the conditional route requires when the target function includes an interaction; the second shows how the inverse probability weights can be redefined to target the treated or the untreated rather than the whole population.

Combining Coefficients for a Conditional Contrast

The interaction section obtained a population contrast from a model with two interactions by marginal standardization. It is worth seeing what the conditional route requires in the simplest interaction setting, a single binary modifier, since this situation arises often. Suppose we are fitting a linear risk model to estimate the risk difference, and we have to include an interaction between the exposure of interest and sex:

$$E(Y | X, C) = \beta_0 + \beta_1 X + \beta_2 C + \beta_3 XC + \dots$$

```
mod <- lm(wt_delta ~ qsmk + sex + qsmk:sex + age + race +
          factor(exercise) + factor(marital) +
          smokeintensity, data = nhefs)

# we'll need robust SEs:
V_hc3 <- vcovHC(mod, type = "HC3")

mod_res <- coeftest(mod, vcov. = V_hc3)

mod_res[c("qsmk", "sex", "qsmk:sex"), ]
```

```
##           Estimate Std. Error   t value    Pr(>|t|)
## qsmk      0.145701893 0.03951200  3.6875354 0.0002342281
## sex       0.004666793 0.03013864  0.1548442 0.8769642871
## qsmk:sex  0.027427302 0.05597427  0.4899983 0.6242042358
```

Based on this model, we might interpret the stratum specific `qsmk` effects separately. The risk difference among individuals classified as male (`sex = 0`) is simply the coefficient for `qsmk` in the model. The risk difference among individuals classified as female (`sex = 1`) is the coefficient for `qsmk` in the model plus the coefficient for `qsmk:sex`. We can compute these by name, or with a contrast vector that picks out and sums the relevant coefficients:

```
# by name
rd_sex0 <- mod_res["qsmk", 1]
```

```
rd_sex1 <- mod_res["qsmk", 1] + mod_res["qsmk:sex", 1]

# by contrast vector
cn <- names(coef(mod))
contrast0 <- as.numeric(cn == "qsmk")
contrast1 <- as.numeric(cn %in% c("qsmk", "qsmk:sex"))

c(rd_sex0, coef(mod) %*% contrast0)
```

```
## [1] 0.1457019 0.1457019
```

```
c(rd_sex1, coef(mod) %*% contrast1)
```

```
## [1] 0.1731292 0.1731292
```

The next question is how to obtain the correct standard errors for both of these estimates. The standard error for the `qsmk` coefficient can be read directly from the robust output. For the combined parameter, we have to note the following variance equation:

$$\text{Var}(A + B) = \text{Var}(A) + \text{Var}(B) + 2\text{Cov}(A, B)$$

Because of this equation, we can't simply use the output of the `coeftest` function, since we do not have the covariances available in this output. However, the covariances are available from the variance-covariance matrix, whose diagonal entries are the variances of the coefficients and whose off-diagonal entries are their covariances (Week 7). We can combine the requisite pieces according to the equation above, or use matrix multiplication with the contrast vector, which generalizes to any number of coefficients:

```
se_sex1_eq <- sqrt(V_hc3["qsmk", "qsmk"] +
                  V_hc3["qsmk:sex", "qsmk:sex"] +
                  2 * V_hc3["qsmk", "qsmk:sex"])

se_sex0 <- sqrt(t(contrast0) %*% V_hc3 %*% contrast0)
```

```
se_sex1 <- sqrt(t(contrast1) %*% V_hc3 %*% contrast1)

c(se_sex1_eq, se_sex1)
```

```
## [1] 0.04030054 0.04030054
```

```
rbind(`Sex = 0` = c(RD = rd_sex0,
                    LCL = rd_sex0 - 1.96 * se_sex0,
                    UCL = rd_sex0 + 1.96 * se_sex0),
      `Sex = 1` = c(RD = rd_sex1,
                    LCL = rd_sex1 - 1.96 * se_sex1,
                    UCL = rd_sex1 + 1.96 * se_sex1)) %>%
  round(3)
```

```
##           RD    LCL    UCL
## Sex = 0 0.146 0.068 0.223
## Sex = 1 0.173 0.094 0.252
```

That was one binary modifier, on a collapsible scale, and we obtained two stratum specific contrasts rather than one population contrast. With two interactions, one of them with a continuous variable, there is no single conditional contrast to read off at all, which is why the main text turned to marginal standardization.

IP Weighting for the ATT and ATU

One can use an inverse probability weighted estimator to compute the treatment effect on the treated and untreated. The key is to use weights that target these effects instead of the ATE. Different weights must be used.

Inverse probability weights for the treatment effect on the treated are defined as:

$$w_{ATT} = \begin{cases} 1 & \text{if } X = 1 \\ \frac{P(X = 1 | C)}{P(X = 0 | C)} & \text{if } X = 0 \end{cases}$$

The treated keep a weight of one, and the untreated are reweighted so that their covariate distribution matches that of the treated. Inverse probability weights for the treatment effect on the untreated reverse the roles:⁸

$$w_{ATU} = \begin{cases} \frac{P(X=0|C)}{P(X=1|C)} & \text{if } X=1 \\ 1 & \text{if } X=0 \end{cases}$$

These estimators can be deployed in R as follows:

```
ps <- nhefs$propensity_score

# inverse probability weights for the ATT
nhefs$w_att <- nhefs$qsmk +
  (ps / (1 - ps)) * (1 - nhefs$qsmk)

# inverse probability weights for the ATU
nhefs$w_atu <- ((1 - ps) / ps) * nhefs$qsmk +
  (1 - nhefs$qsmk)

model_att <- glm(wt_delta ~ qsmk, data = nhefs,
  weights = w_att,
  family = quasibinomial("identity"))

coeftest(model_att,
  vcov. = vcovHC(model_att, type = "HCO"))

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.440993   0.016234  27.165 < 2.2e-16 ***
## qsmk        0.154540   0.029347   5.266 1.394e-07 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

⁸ A different set of weights, $1 - P(X = 1 | C)$ for the exposed and $P(X = 1 | C)$ for the unexposed, is sometimes confused with the ATU weights. These are the overlap weights, which target the effect in the population with the most covariate overlap between exposed and unexposed, a different estimand again (Li et al., 2018).

```

model_atu <- glm(wt_delta ~ qsmk, data = nhefs,
                 weights = w_atu,
                 family = quasibinomial("identity"))

coeftest(model_atu,
          vcov. = vcovHC(model_atu, type = "HCO"))

##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.466896   0.014629  31.915 < 2.2e-16 ***
## qsmk        0.154645   0.031160   4.963 6.942e-07 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The three weighted risk differences are nearly identical in these data, which is what we expect when the exposure effect does not vary much across the covariates: the ATE, ATT, and ATU differ only through effect heterogeneity across the covariate distributions of the exposed and unexposed. The same machinery extends to categorical and continuous exposures, for which the propensity score becomes a predicted probability from a multinomial model, or a density, of the observed exposure level ([Naimi et al., 2014](#)).

References

- S. R. Cole and M. A. Hernán. Constructing inverse probability weights for marginal structural models. *Am J Epidemiol*, 168(6):656–64, 2008.
- Rhian Daniel, Jingjing Zhang, and Daniel Farewell. Making apples from oranges: Comparing noncollapsible effect estimators and their standard errors after adjustment for different covariate sets. *Biometrical Journal*, 63(3):528–557, 2021. DOI: 10.1002/bimj.201900297.
- Sander Greenland. Collapsibility. In Mitchell H. Gail and Jacques Bénichou, editors, *Encyclopedia of Epidemiologic Methods*. John Wiley & Sons, Ltd, 2005.
- Masayuki Henmi and Shinto Eguchi. A paradox concerning nuisance parameters and projected estimating functions. *Biometrika*, 91(4):929–941, 12 2004.
- M. A. Hernán and JM Robins. *Causal Inference*. Chapman & Hall/CRC, Boca Raton, FL, 2020.
- Miguel A Hernán and James M Robins. Estimating causal effects from epidemiological data. *J Epidemiol Community Health*, 60(7):578–586, 2006.
- Fan Li, Kari Lock Morgan, and Alan M Zaslavsky. Balancing covariates via propensity score weighting. *J Am Stat Assoc*, 113(521):390–400, 2018. DOI: 10.1080/01621459.2016.1260466.
- Jared K. Lunceford and Marie Davidian. Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study. *Statistics in Medicine*, 23(19):2937–2960, 2022/03/22 2004. DOI: <https://doi.org/10.1002/sim.1903>. URL <https://doi.org/10.1002/sim.1903>.
- Clemma J Muller and Richard F MacLehose. Estimating predicted probabilities from logistic regression: different methods correspond to different target populations. *Int J Epidemiol*, 43(3):962–970, Jun 2014. DOI: 10.1093/ije/dyu029.

- Ashley I Naimi and Brian W Whitcomb. Estimating risk ratios and risk differences using regression. *Am J Epidemiol*, 189(6):508–510, 2020. doi: 10.1093/aje/kwaa044.
- Ashley I Naimi, Stephen R Cole, and Edward H Kennedy. An Introduction to G Methods. *Int J Epidemiol*, 46(2):756–762, 2017.
- Ashley Isaac Naimi, Erica E M Moodie, N Auger, and Jay S Kaufman. Constructing Inverse Probability Weights for Continuous Exposures: A Comparison of Methods. *Epidemiol*, 25(2):292–299, 2014.
- J. M. Robins. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical Modelling*, 7:1393–1512, 1986.
- James M Robins, Steven D Mark, and Whitney K Newey. Estimating exposure effects by modelling the expectation of exposure conditional on confounders. *Biometrics*, 48(2):479–495, 1992.
- Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.
- Tosiya Sato and Yutaka Matsuyama. Marginal structural models as a tool for standardization. *Epidemiol*, 14(6):680–86, 2003.
- Jonathan M. Snowden, Sherri Rose, and Kathleen M. Mortimer. Implementation of g-computation on a simulated data set: Demonstration of a causal inference technique. *Am J Epidemiol*, 173(7):731–738, 2011.
- Elizabeth J. Williamson, Andrew Forbes, and Ian R. White. Variance reduction in randomised trials by inverse probability weighting using the propensity score. *Statistics in Medicine*, 33(5):721–737, 2014.
- Zhiwei Zhang. Estimating a marginal causal odds ratio subject to confounding. *Communications in Statistics - Theory and Methods*, 38(3):309–321, 2009. doi: 10.1080/03610920802200076.
- Guangyong Zou. A modified poisson regression approach to prospective studies with binary data. *Am J Epidemiol*, 159(7):702–706, Apr 2004.