G Computation

Ashley I. Naimi, PhD

Outline

G Computation

- Some Preliminaries
- Model Based Standardization
- The Effect of Smoking on Weight Change
- Time-Varying Confounding

Some Preliminaries

In this section, we will illustrate implementation of the parametric g formula using four examples with simulated and empirical data. The first will be a very simple setting with one exposure, one confounder, and one outcome. This example will demonstrate model-based standardization, which is essentially what the parametric g formula does with complex longitudinal data. However, the data from the first example are neither complex nor longitudinal.

The second example will be similar to the first, but slightly more complicated because we will use real data to estimate the impact of smoking on high blood pressure.

The third example will be identical to the first, except the exposure will be measured twice (time-varying). It will also include a time-varying confounder measured once, but that creates a feedback loop between the first and second exposure measurement. This is the simplest complex longitudinal data scenario in which one can implement the g formula, and we will use it to emphasize core concepts.

In the first two examples, we will establish a series of procedures to implement the g formula in a wide range of settings. Specifically, we will discuss problem setup, implementation, validation, and interpretation. The setup stage is about what you need to write down and organize to implement the parametric g formula. In the implementation stage, I will show you what models you need to fit based on the setup. After fitting these models, we need to evaluate quality (validation stage). Finally we must interpret in light of the assumptions we covered in the previous section.

The parametric g formula is the first of three "g" methods developed by James Robins beginning in the mid-1980s. The other g methods are: g estimation of a structural nested model, and inverse probability weighted marginal structural models.

Inverse probability weighted marginal structural models consist of two important parts: the marginal structural model, which is a model for potential outcomes (structural) averaged over the entire population (marginal). Inverse proabaility weights are a tool that enable estimation of the MSM parameters (e.g., weighted least squares or weighted maximum likelihood).

G estimation of a structural nested model also consist of two parts: the structural nested model, which is a model for a contrast of potential outcomes (structural) within levels of past time-varying and baseline covariates (nested). G estimation is an **estimator** that takes advantage of the independence between the potential outcomes and the observed expsoure (i.e., exchangeability) to solve for the parameters of a SNM.

Marginal structural and structural nested models target very different estimands. As we will see, the g formula is simply an equation that links potential outcomes to observed data (i.e., outcomes, exposures, confounders). It can be used to target the quantities defined in either marginal structural or structural nested models. As it turns out, if we are willing to model each of the terms in the (potentially lengthy) equation, can also use it to estimate the effects quantified by these models.

Example 1: Model-Based Standardization

Let's start with a simple simulated example, and presume it represents data to answer questions about the effect of treatment for HIV on CD4 count. The causal diagram representing this scenario is depicted in Figure 3.

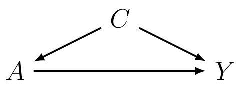


Figure 1: Causal diagram representing the relation between anti-retroviral treatment (A), HIV viral load just prior to treatment (C), and CD4 count measured at the end of follow-up (Y).

Table 1 presents data from this simulated observational cohort study ($A = 1$ for treated, $A = 0$ otherwise).

Table 1: Example data illustrating the number of subjects (N) within each possible combination of treatment (A) and HIV viral load (C). The outcome column (Y) corresponds to the mean of Y within levels of A and C.

C	A	Y	N
0	0	94.3	344052
0	1	119.2	154568
1	0	130.6	154560
1	1	155.7	346820

The CD4 outcome in Table 1 is summarized (averaged) over the participants at each level of the treatments and covariate. Becuase the continuous outcome is summarized over each treatment × covariate level, we cannot estimate standard errors but will rather focus on estimating the parameter of interest. We will analyze these data using **model-based standardization, which is equivalent to the parametric g formula in a time-fixed exposure setting.**

Setup

We first start with the **setup**, where we define our estimand, order our variables causally, write down our models, and "tie" them together into the g formula. In this simple setting, our estimand of interest is the marginal average causal effect on the difference scale:

$$E(Y^{a=1} - Y^{a=0})$$

This estimand tells us that we need to quantify two outcome averages: one that would be observed if everyone were exposed, and one if everyone were unexposed.

Next, we examine our causal diagram to order our variables causally. The causal sequence of variables is: C (first), A (second), and Y (third). To see why, note that in Figure 3 there are no variables that cause C, A is caused by C, and Y is caused by both A and C. Becuase of this, A cannot come before C (an effect cannot precede its cause), nor can Y come before A or C. The causal ordering of our variables is therefore C, A, and Y.

We then write down models for each variable.[1] How do we know which models to specify? We regress each variable against everything that comes before it.

Variable	Model
Y	$E(Y \mid A, C) = \alpha_0 + \alpha_1 A + \alpha_2 C$
A	$P(A \mid C) = \text{expit}(\beta_0 + \beta_1 C)$
C	$P(C) = \text{expit}(\gamma_0)$

However, we must ensure that we do not break the **cardinal rule: do not adjust for the future.**

Finally, we tie each of these models together to give us a precursor to the g formula. To do this, we invoke the law of total proba-

[1] Recall: The "expit" function is the inverse of the logit: $\text{expit}(a) = 1/[(1 + \exp(-a))]$.

bility, which states that the $P(A) = \sum_B P(A \mid B)P(B)$. This allows us to "average over" a conditional to obtain a marginal. In our case, the relevant conditional is the regression model for the outcome, and we have to average over the distributions of A and C:

$$E(Y) = \sum_A \sum_C E(Y \mid A, C)P(A \mid C)P(C)$$

To obtain the g formula from this expression, we replace all instances of A with $A = a$ and remove $P(A \mid C)$

$$E(Y^a) = \sum_C E(Y \mid A = a, C)P(C)$$

which holds under our identifiability assumptions.

Implementation and Validation

We're now ready for **implementation**. Suppose we wanted to estimate the unconditional (i.e., marginal) mean outcome in the sample. There are two ways we can do this. The easy way would be to simply take the average in the sample:

```
## CODE SET 2
# arrange into long data
C<-c(0,0,1,1);A<-c(0,1,0,1);Y<-c(94.3,119.2,130.6,155.7)
N<-c(344052,154568,154560,346820)
D<-NULL
for(i in 1:4){
  d<-data.frame(cbind(rep(C[i],N[i]),rep(A[i],N[i]),rep(Y[i],N[i])))
  D<-rbind(D,d)
}
names(D)<-c("C","A","Y")
# take the mean of Y
mean(D$Y)
```

```
## [1] 125.054
```

```
## END CODE SET 2
```

But we could also compute the marginal mean using the law of total probability. To do this, we can estimate our models using the data, and then predict from each in sequence:

```
## CODE SET 3
# fit models
mC<-glm(C~1,data=D,family=binomial("logit"))
mA<-glm(A~C,data=D,family=binomial("logit"))
mY<-glm(Y~A+C,data=D,family=gaussian("identity"))

## obtain predictions
# obtain C predictions
pC<-predict(mC,type="response")
# use predicted C to obtain predicted A
pA<-predict(mA,newdata=data.frame(C=pC),type="response")
# use predicted A and C to obtain predicted Y
pY<-predict(mY,newdata=data.frame(A=pA,C=pC),type="response")

# compute marginal mean of predicted Y
mean(pY)

## [1] 125.0584

## END CODE SET 3
```

The key is that C is predicted, then A is predicted using the C predictions, and then Y is predicted using the A and C predictions.

Side Note: To see why this works, suppose we're interested in the marginal (i.e., averaged over C) mean of Y if $A = 0$, and let's assume for illustrative purposes that $P(C = 1) = 0.2$ (it's not in our example):

$$\begin{aligned} E(Y \mid A = 0) &= \textstyle\sum_C E(Y \mid A = 0, C)P(C) \\ &= E(Y \mid A = 0, C = 0)P(C = 0) + E(Y \mid A = 0, C = 1)P(C = 1) \\ &= \alpha_0 \times 0.8 + (\alpha_0 + \alpha_2) \times 0.2 \end{aligned}$$

Note that, in the second line of the above, $E(Y \mid A = 0, C = 0)$ and $E(Y \mid A = 0, C = 1)$ are just the averages of Y among those with $A = 0, C = 0$ and $A = 0, C = 1$, respectively. We can therefore replace these with the parameters from our model. In a dataset of 100 people with $A = 0$, $\sim$ 80 would have $C = 0$ and $\sim$ 20 would have $C = 1$. Among those 100, the true average outcome for those with $C = 0$ would be α_0, and the true average outcome for those with $C = 1$ would be $\alpha_0 + \alpha_2$. Therefore, the average of Y among these 100 people with $A = 0$ would be precisely the weighted combination of averages that we need: $\alpha_0 \times 0.8 + (\alpha_0 + \alpha_2) \times 0.2$. This is why we can use our data and/or predictions to implement the law of total probability.

Back to our original example, we have two versions of our outcome: the actual data (Y) and the predictions based on our models (pY). The mean of both these versions is the same: 125.0. This **validation** step tells us that our models are doing a decent job at recreating the averages that result from our actual data generating mechanisms.

Continuing with our **implementation**, we can also use this code to predict Y if $A = 1$ for everyone or if $A = 0$ for everyone. We must just replace "A=pA" with "A=1" and "A=0" in the last line of code that yields the predictions we want. Replacing "A=pA" with "A=a" is tantamount to replacing all instances of A in the above equations with $A = a$, and removing the $P(A \mid C)$ term:

```
## CODE SET 4
# for A=1
pY_1<-predict(mY,newdata=data.frame(A=1,C=pC),type="response")
mY_1<-mean(pY_1)

#for A=0
pY_0<-predict(mY,newdata=data.frame(A=0,C=pC),type="response")
mY_0<-mean(pY_0)
## END CODE SET 4
```

The difference between these two means of interest is 25, which we must **interpret**.

Interpretation

The basic question is whether we can interpret this difference as the causal effect of ART on CD4 count. To do this, we must refer back to the set of assumptions discussed in the section on identifiability. For counterfactual consistency, we must ask two key questions: 1) how many different ways are there to assign someone to ART?; and 2) will these different assignment mechanisms lead to different outcomes? Suppose, for instance, that 1/2 of the sample took ART with ibuprofen. Suppose further that ibuprofen reduces the efficacy of ART. We then have a situation where counterfactual consistency may be violated, becuase assigning someone to ART (without ibuprofen) will not lead to the same effect that was quantified in our study. If we assume that all of the different ways in which one can take ART will not really lead to different outcomes, we can assume counterfactual consistency.

For interference, we must ask whether giving someone ART will affect the CD4 count of another person. In this case, it seems reasonable to assume no such interference occurs. Exchangeability is something we often consider in epidemiology, and requires no uncontrolled confounding, information, or selection bias.

Becuase of the small number of variables in this example, correct model specification is not likely to pose any problems. If, for example, an interaction between A and C in the model for Y is required, our model would be mis-specified. With a small number of categorical variables, we can saturate all the models to estimate things nonparametrically. However, this is often not possible when there are many categorical confounders, or any continuous confounders.

Finally, for positivity, we must ask whether there are exposed and unexposed individuals in each confounder level. In our simple setting, it is easy to verify this with a 2×2 table:

```
table(D$A, D$C)
```

```
##
##          0      1
##   0 344052 154560
##   1 154568 346820
```

Becuase there are no empty cells in this table, we can assume positivity is met. Additionally, becuase we are willing to make all these identifiability assumptions, we infer that the causal effect of A on Y is 25.

Example 2: Effect of Quitting Smoking on Weight Gain (NEHFS)

Let's apply the same reasoning to a real dataset. We'll load the same NEHFS data we used in the section on models:

```
aa <- read_csv("./nhefs.csv")
# original sample size
nrow(aa)
```

```
## [1] 1746
```

```
a <- aa %>% select(seqn, qsmk, smkintensity82_71, smokeintensity, active, exercise, wt82_71,
    sbp, dbp, hbp, hf, ht, hbpmed, sex, age, hf, race, income, marital, school, asthma, bronch,
    diabetes)
a$hbp_71 <- a$hbp


a <- a %>% na.omit()


a$delta <- as.numeric(a$wt82_71 > 0)
```

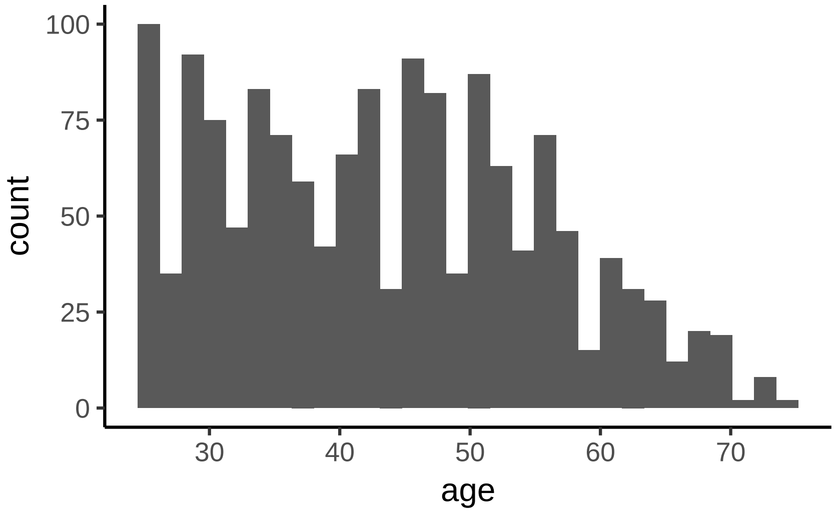
Let's again assume that the relevant confounders are: `sex`, `age`, `race`, `income`, `marital`, `school`, `active`, `hf`, `hbpmed`, `asthma`, `bronch`, `smokeintensity`, `exercise`, `diabetes`, and `hbp_71`.

```
a <- a %>% select(delta, qsmk, sex, age, race, income, marital, school, active, hf, hbpmed,
    asthma, bronch, smokeintensity, exercise, diabetes, hbp_71)
```

One important question to address is how adjust for continuous variables, or categorical variables with many levels. For example,

```
# age distribution
ggplot(a, aes(x = age)) + geom_histogram()
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



```
# income levels
table(a$income)
```

```
##
##  11  12  13  14  15  16  17  18  19  20  21  22
##  27  55  61  58  74  71  61 269 394 212 110  84
```

In typical settings, continuous variables would be fitted using polynomial or spline functions, while multi-level categorical variables would be further categorized by judicious selection of thresholds. In this course, we will forego these complications, and categorize each variable rather hastily (recall, we are ignoring the numerically coded missing data).

```
a$smokeintensity <- as.numeric(a$smokeintensity > median(a$smokeintensity))
a$age <- as.numeric(a$age > median(a$age))
a$exercise <- as.numeric(a$exercise > 0)
a$income <- as.numeric(a$income > median(a$income))
a$marital <- as.numeric(a$marital > median(a$marital))
a$school <- as.numeric(a$school > median(a$school))
a$active <- as.numeric(a$active > 0)
a$hbpmed <- as.numeric(a$hbpmed > 0)
a$smokeintensity <- as.numeric(a$smokeintensity > median(a$smokeintensity))
a$exercise <- as.numeric(a$exercise > 0)
a$diabetes <- as.numeric(a$diabetes > 0)
a$hbp_71 <- as.numeric(a$hbp_71 > 0)
```

Setup

We first start with the **setup**: define the estimand, order the variables causally, write down the models, "tie" them together into the g formula. Again, in this simple setting, our estimand of interest is the marginal average causal effect on the difference scale:

$$E(Y^{a=1} - Y^{a=0})$$

And again, this estimand tells us we need to quantify the same two outcome averages.

Even in this complex setting, the ordering of the variables is relatively straightforward. This is becuase we actually don't have to order any of the baseline confounders, which consists of all confounders. The causal ordering of our variables is therefore baseline confounders, quitting smoking, and weight change.

Instead of writing down a model for each baseline confounder, we can model their joint distribution empirically. What this means is we just have to model the exposure[2] So the only models we need are:

Variable	Model
Y	$E(Y \mid A, C) = \alpha_0 + \alpha_1 A + \alpha_2 \mathbf{C}$
A	$P(A = 1 \mid C) = \text{expit}(\beta_0 + \beta_1 C)$

Still, we must not break the **cardinal rule: do not adjust for the future.**

Finally, we again tie each of these models together to give us a pre-cursor to the g formula:

$$E(Y) = \sum_A \sum_C E(Y \mid A, C) P(A \mid C) P(C)$$

Again, to obtain the g formula from this expression, we replace all instances of A with $A = a$ and remove $P(A \mid C)$

$$E(Y^a) = \sum_C E(Y \mid A = a, C) P(C)$$

which holds under our identifiability assumptions.

We can fit these models fairly easily:

```
model_A <- glm(qsmk ~ sex + age + race + income + marital + school + active + hf + hbpmed +
```

[2] In principle, we only have to model the exposure to generate the natural course, not to estimate the effect of interest. However, because the natural course is a critical verification step, it should be done every time.

```
    asthma + bronch + smokeintensity + exercise + diabetes + hbp_71, data = a, family = binomial("logit"))
summary(model_A)
```

```
##
## Call:
## glm(formula = qsmk ~ sex + age + race + income + marital + school +
##     active + hf + hbpmed + asthma + bronch + smokeintensity +
##     exercise + diabetes + hbp_71, family = binomial("logit"),
##     data = a)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2413  -0.7884  -0.6664  -0.4343   2.2404
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.3409     0.1942  -6.903 5.09e-12 ***
## sex             -0.2968     0.1304  -2.275  0.02288 *
## age              0.3889     0.1280   3.039  0.00237 **
## race            -0.6628     0.2171  -3.053  0.00227 **
## income          -0.1965     0.1524  -1.290  0.19722
## marital         -0.2327     0.1646  -1.413  0.15757
## school           0.4293     0.1607   2.672  0.00755 **
## active           0.1671     0.1314   1.272  0.20329
## hf               0.1056     0.8505   0.124  0.90121
## hbpmed           0.1607     0.3999   0.402  0.68783
## asthma           0.3852     0.2817   1.367  0.17148
## bronch          -0.1082     0.2265  -0.478  0.63268
## smokeintensity  -0.2864     0.1454  -1.970  0.04884 *
## exercise         0.2753     0.1707   1.613  0.10679
## diabetes        -0.4228     0.3255  -1.299  0.19399
## hbp_71           0.2143     0.2841   0.754  0.45068
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1644.5  on 1475  degrees of freedom
## Residual deviance: 1596.3  on 1460  degrees of freedom
## AIC: 1628.3
##
## Number of Fisher Scoring iterations: 4
```

```
model_Y <- glm(delta ~ qsmk + sex + age + race + income + marital + school + active + hf +
    hbpmed + asthma + bronch + smokeintensity + exercise + diabetes + hbp_71, data = a, family = binomial(
summary(model_Y)
```

```
##
## Call:
## glm(formula = delta ~ qsmk + sex + age + race + income + marital +
##     school + active + hf + hbpmed + asthma + bronch + smokeintensity +
##     exercise + diabetes + hbp_71, family = binomial("logit"),
##     data = a)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0741  -1.2385   0.7275   0.8802   1.4810
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.077644   0.181006   5.954 2.62e-09 ***
## qsmk            0.575806   0.140980   4.084 4.42e-05 ***
## sex             0.115711   0.121413   0.953   0.3406
## age            -0.808872   0.119493  -6.769 1.30e-11 ***
## race           -0.082986   0.175659  -0.472   0.6366
## income          0.332082   0.143504   2.314   0.0207 *
## marital         0.128344   0.147510   0.870   0.3843
## school         -0.186770   0.156406  -1.194   0.2324
## active         -0.247713   0.122371  -2.024   0.0429 *
## hf             -0.606815   0.744444  -0.815   0.4150
## hbpmed         -0.481038   0.377352  -1.275   0.2024
## asthma          0.238478   0.290449   0.821   0.4116
## bronch          0.076413   0.209243   0.365   0.7150
## smokeintensity -0.004877   0.133743  -0.036   0.9709
## exercise        0.005742   0.154674   0.037   0.9704
## diabetes        0.629890   0.307314   2.050   0.0404 *
## hbp_71         -0.236019   0.267772  -0.881   0.3781
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1867.7  on 1475  degrees of freedom
## Residual deviance: 1783.3  on 1459  degrees of freedom
## AIC: 1817.3
##
## Number of Fisher Scoring iterations: 4
```

Implementation and Validation

We're now ready for **implementation**, but things are a little different with so many confounders and realistic sample sizes. First, to empirically model the joint distribution of all confounders, we simply need to resample them with replacement. The size of the resample will depend on the complexity of the confounder space. There are two competeing issues here: 1) Becuase we only have binary confounders, we can reduce the Monte Carlo[3] sample size. However, because we have many of them, we should choose a sufficiently large Monte Carlo sample size.

```
# resample data
index <- sample(1:nrow(a), size = 10000, replace = T)
length(index)
```

```
## [1] 10000
```

```
MC <- a[index, ]
nrow(MC)
```

```
## [1] 10000
```

```
MC$qsmk <- NULL
```

[3] The Monte Carlo method is an approach to solving things using simulation. In this case, we are solving the g formula by randomly resampling the baseline data, and simulating from the models we fit above.

```
# predict exposure
pA <- predict(model_A, newdata = MC, type = "response")
```

The variable pA is the predicted exposure. Let's **validate** our model by comparing this predicted exposure matches the actual exposure: 0.2454356 versus 0.2452575.

The new predicted variable for quitting smoking, pA is not a binary indicator, but rather takes on values *between* 0 and 1. To convert it to a binary exposure variable, we can compare each value to a uniform random value:

```
u <- runif(10000)
qA <- as.numeric(pA > u)
head(qA)
```

```
## [1] 0 0 0 0 0 0
```

```
mean(qA)
```

```
## [1] 0.2466
```

```
mean(a$qsmk)
```

```
## [1] 0.2452575
```

Now that we have our new simulated exposure qA, we can simulate and **validate** the outcome:

```
pY <- predict(model_Y, newdata = data.frame(MC, qsmk = qA), type = "response")

mean(pY)
```

```
## [1] 0.6736064
```

```
mean(a$delta)
```

```
## [1] 0.6720867
```

We can now estimate the effect of smoking on weight gain:

```
pY_1 <- predict(model_Y, newdata = data.frame(MC, qsmk = 1), type = "response")
mY_1 <- mean(pY_1)
```

```
pY_0 <- predict(model_Y, newdata = data.frame(MC, qsmk = 0), type = "response")
mY_0 <- mean(pY_0)

RD <- round((mY_1 - mY_0) * 100, 2)
RD
```

```
## [1] 11.37
```

```
RR <- round(mY_1/mY_0, 2)
RR
```

```
## [1] 1.18
```

This analysis yielded a risk difference of 11.37 per 100 participants and a risk ratio of 1.18 for the relation between quitting smoking and gaining weight. We'll now **interpret** this effect.

Side Note: In this empirical analysis, we chose a binary indicator of whether any weight was gained between the two study visits. We could just as easily have modeled weight change on the continuous scale, as in the other two examples.

Interpretation

Can we interpret this difference as the causal effect of quitting smoking on weight? Let's refer back to the set of identifiability assumptions. For counterfactual consistency 1) how many different ways are there to get someone to quit smoking?; and 2) will these different assignment mechanisms lead to different outcomes? There is a relatively narrow set of ways to get someone to quit smoking, and (to my knowledge) they are unlikely to lead to drastically different weight changes.[4]

For interference, we must ask if a given person quit smoking, will it affect the outcome of another person? This is absolutely possible, given the effects of second hand smoke and the motivational component of quitting smoking with someone. In this case, there are two

[4] Though I am not a subject matter expert here, so feel free to point out if you disagree.

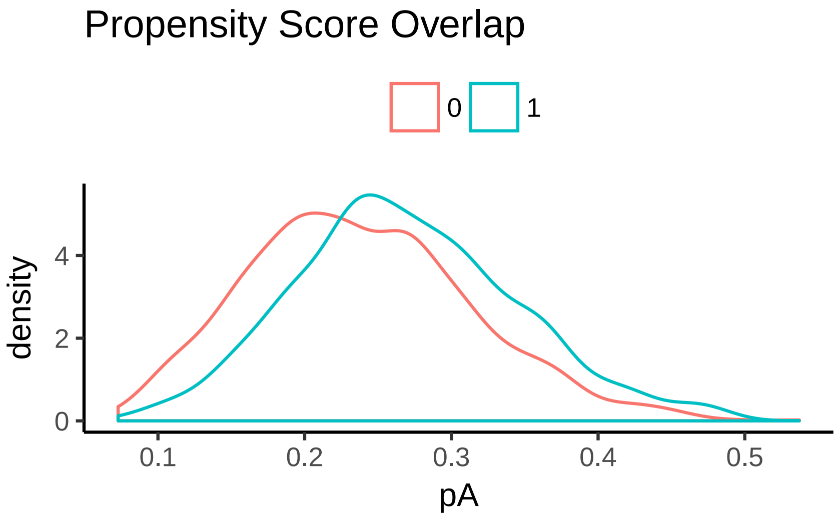
things we can possibly do: 1) we can assume that in these data, no two participants are "close enough" in space or a network of social connections that this will matter; 2) we can change our estimand from the average treatment effect to one that accounts for interference (Hudgens and Halloran 2008). This latter route is much more complicated to implement, and not possible with these data because we'd have to know how closely each participant in the study is connected with others.

Correct model specification is another issue to consider. To simplify the illustration, we dichotomized all of the confounding variables. This has a direct bearing on correct model specification. We also ignored any interactions between quitting smoking and any confounders in the outcome model. This is often a key challenge when using real data from an observational study.

Finally, for positivity, we must ask whether there are exposed and unexposed individuals in each confounder level. Becuase of the number of confounding variables, we cannot simply use a 2×2 table. Instead we should examine propensity score overlap:

```
prop <- model_A$fitted.values
propD <- data.frame(A = as.factor(a$qsmk), pA = prop)

ggplot(propD, aes(x = pA, color = A)) + geom_density() + ggtitle("Propensity Score Overlap")
```



This plot looks good, since there is reasonable overlap between the two groups.

Confidence Intervals

To get confidence intervals for our risk difference and risk ratio, the only option is to use the bootstrap. To do this, we have to resample (with replacement) the original data, re-fit the model for the outcome, and obtain a contrast from this resample. If we do this 100 times, we can use the standard deviation of these 100 point estimates as the standard error of the estimator, and obtain the usual Wald confidence limits:

```
res <- NULL
for (i in 1:100) {
    index <- sample(1:nrow(a), nrow(a), replace = T)
    boot_dat <- a[index, ]
    model_Y <- glm(delta ~ qsmk + sex + age + race + income + marital + school + active + hf +
        hbpmed + asthma + bronch + smokeintensity + exercise + diabetes + hbp_71, data = boot_dat,
        family = binomial("logit"))

    index <- sample(1:nrow(a), size = 10000, replace = T)
    MC <- a[index, ]
    MC$qsmk <- NULL

    mY_1 <- mean(predict(model_Y, newdata = data.frame(MC, qsmk = 1), type = "response"))
    mY_0 <- mean(predict(model_Y, newdata = data.frame(MC, qsmk = 0), type = "response"))

    RD <- (mY_1 - mY_0) * 100
    logRR <- log(mY_1/mY_0)

    res <- rbind(res, cbind(RD, logRR))
}

head(res)
```

```
##              RD     logRR
## [1,] 13.248493 0.1900819
## [2,] 11.922114 0.1704617
## [3,]  9.844808 0.1404582
## [4,] 14.836709 0.2226961
## [5,] 12.219702 0.1727288
## [6,] 13.000885 0.1840391
```

```
res_sd <- apply(res, 2, sd)

lclRD <- RD - 1.96 * res_sd[1]
uclRD <- RD + 1.96 * res_sd[1]

lclRR <- exp(log(RR) - 1.96 * res_sd[2])
uclRR <- exp(log(RR) + 1.96 * res_sd[2])
```

This bootstrap estimator yields 95% CIs of 4.47, 14.53 for the risk difference, and 1.1, 1.27 for the risk ratio.

Example 3: ART effect on CD4 Count (Simulated)

In the previous examples, we dealt with data that was neither longitudinal nor complex. We did not need to analyze these data using the g formula. In fact, a simple standard regression would have given us the same result. Here, we extend our previous example by adding an additional exposure, and converting our time-fixed confounder C to a time-dependent confounder Z. Our research question again deals with the effect of treatment for HIV on CD4 count.[5] The causal diagram representing this scenario is depicted in Figure 4.

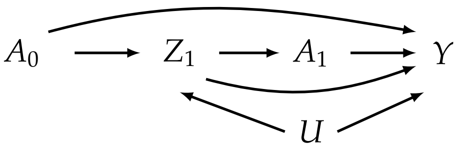


Figure 2: Causal diagram representing the relation between anti-retroviral treatment at time 0 (A_0), HIV viral load just prior to the second round of treatment (Z_1), anti-retroviral treatment status at time 1 (A_1), the CD4 count measured at the end of follow-up (Y), and an unmeasured common cause (U) of HIV viral load and CD4.

Study Question 5: Does the fact that U is unmeasured in Figure 4 create problems for our analysis? Why or why not?

[5] This example was taken from Naimi et al. (2016)

Table 1 presents data from a hypothetical observational cohort study ($A = 1$ for treated, $A = 0$ otherwise). Treatment is measured at baseline (A_0) and once during follow up (A_1). The sole covariate is elevated HIV viral load ($Z = 1$ for those with > 200 copies/ml, $Z = 0$ otherwise), which is constant by design at baseline ($Z_0 = 1$) and measured once during follow up just prior to the second treatment (Z_1). The outcome is CD4 count measured at the end of follow up in units of cells/mm^3. Again, the CD4 outcome in Table 1 is summarized (averaged) over the participants at each level of the treatments and covariate.

A_0	Z_1	A_1	Y	N
0	0	0	87.29	209,271
0	0	1	112.11	93,779
0	1	0	119.65	60,654
0	1	1	144.84	136,293
1	0	0	105.28	134,781
1	0	1	130.18	60,789
1	1	0	137.72	93,903
1	1	1	162.83	210,527

Table 2: Prospective study data illustrating the number of subjects (N) within each possible combination of treatment at time 0 (A_0), HIV viral load just prior to the second round of treatment (Z_1), and treatment status for the 2nd round of treatment (A_1). The outcome column (Y) corresponds to the mean of Y within levels of A_0, Z_1, A_1. Note that HIV viral load at baseline is high ($Z_0 = 1$) for everyone by design.

Setup

The number of participants is provided in the rightmost column of Table 1. In this hypothetical study of one million participants we ignore random error (i.e., we will not focus on confidence interval estimation). Let's again start with the problem **setup**, where we define our estimand, order our variables causally, write down our models, and "tie" them together into the g formula. Here, we focus on the average causal effect of always taking treatment, $(a_0 = 1, a_1 = 1) \equiv \bar{a}_1 = 1$, compared to never taking treatment, $(a_0 = 0, a_1 = 0) \equiv \bar{a}_1 = 0$:

$$\psi = E(Y^{\bar{a}_1=1}) - E(Y^{\bar{a}_1=0}).$$

This average causal effect consists of the joint effect of A_0 and A_1 on Y (Daniel et al. 2013). Here, $Y^{\bar{a}_1}$ represents a potential outcome value that would have been observed had the exposures been set to specific levels a_0 and a_1.

The causal order of our observed variables is: A_0, Z_1, A_1, and Y.[6]

[6] Note that we ignore U in this step becuase it is not measured.

For each of these variables, we can write down the following models:

Variable	Model
Y	$E(Y \mid A_1, Z_1, A_0) = \alpha_0 + \alpha_1 A_1 + \alpha_2 Z_1 + \alpha_3 A_0$
A_1	$P(A_1 \mid Z_1) = \text{expit}(\beta_0 + \beta_1 Z_1)$
Z_1	$P(Z_1 \mid A_0) = \text{expit}(\gamma_0 + \gamma_1 A_0)$
A_0	$P(A_0) = \text{expit}(\theta_0)$

Again, these models are obtained by regressing each variable against everything that comes before. Next, we tie each of these equations together to give us a precursor to the g formula. As in the previous example, we use the law of total probability to do this, which yields:

$$E(Y) = \sum_{A_1}\sum_{Z_1}\sum_{A_0} E(Y \mid A_1, Z_1, A_0) P(A_1 \mid Z_1) P(Z_1 \mid A_0) P(A_0).$$

We get the g formula when we replace all instances of A_0 and A_1 with a_0 and a_1, respectively, and remove the models for A_0 and A_1:

$$E(Y^{a_0,a_1}) = \sum_{Z_1} E(Y \mid A_1 = a_1, Z_1, A_0 = a_0) P(Z_1 \mid A_0 = a_0).$$

which holds under our identifiability assumptions.

Implementation

Let's now **implement** the g formula in our software programs. We will again start by estimating the unconditional (i.e., marginal) mean outcome in the sample, by first taking the sample average:

```
## CODE SET 5
# arrange into wide data
a0<-c(0,0,0,0,1,1,1,1);z1<-c(0,0,1,1,0,0,1,1);a1<-c(0,1,0,1,0,1,0,1)
y<-c(87.29,112.11,119.65,144.84,105.28,130.18,137.72,162.83)
N<-c(209271,93779,60654,136293,134781,60789,93903,210527)
D<-NULL
for(i in 1:8){
  d<-data.frame(cbind(rep(a0[i],N[i]),rep(z1[i],N[i]),rep(a1[i],N[i]),rep(y[i],N[i])))
  D<-rbind(D,d)
}
```

```
names(D)<-c("a0","z1","a1","y")
# take the mean of Y
mean(D$y)
```

```
## [1] 125.0948
```

```
## END CODE SET 5
```

Next, we compute the marginal mean using the law of total probability by estimating our models using the data, and then predicting from each in sequence:

```
## CODE SET 6
# fit models
mA0<-glm(a0~1,data=D,family=binomial("logit"))
mZ1<-glm(z1~a0,data=D,family=binomial("logit"))
mA1<-glm(a1~z1,data=D,family=binomial("logit"))
mY<-glm(y~a1+z1+a0,data=D,family=gaussian("identity"))

## obtain predictions
# obtain A0 predictions
pA0<-predict(mA0,type="response")
# use predicted A0 to obtain predicted Z1
pZ1<-predict(mZ1,newdata=data.frame(a0=pA0),type="response")
# use predicted Z1 to obtain predicted A1
pA1<-predict(mA1,newdata=data.frame(z1=pZ1),type="response")
# use predicted A0, Z1 and A1 to obtain predicted Y
pY<-predict(mY,newdata=data.frame(a0=pA0,z1=pZ1,a1=pA1),type="response")

# compute marginal mean of predicted Y
mean(pY)
```

```
## [1] 125.102
```

```
## END CODE SET 6
```

Validation

Once again, we have two versions of our outcome: the actual data (Y) and the predictions based on our models (pY). These latter predictions are obtained under a very specific scenario: by consistency and no interference, it is the outcome distribution that would be observed if the exposure distribution was what actually occurred in our data. This scenario, called the **natural course**, is in contrast to what might have been observed if everyone were exposed/unexposed at both time-points. Estimating the natural course is an important **validation step** when using the parametric g formula. If the empirical results align closely with the natural course, this offers some assurance that our models are not grossly mis-specified. On the other hand, if our empirical and natural course results differ substantially, this suggests that something may be wrong.[16]

In our example, the empirical and natural course means are again the same: 125.1.

Continuing with our **implementation**, we can also use this code to predict Y if $A = 1$ for everyone or if $A = 0$ for everyone:

```
## CODE SET 7
# for A=1
pZ_1<-predict(mZ1,newdata=data.frame(a0=1),type="response")
pY_1<-predict(mY,newdata=data.frame(a0=1,z1=pZ_1,a1=1),type="response")
mY_1<-mean(pY_1)

#for A=0
pZ_0<-predict(mZ1,newdata=data.frame(a0=0),type="response")
pY_0<-predict(mY,newdata=data.frame(a0=0,z1=pZ_0,a1=0),type="response")
mY_0<-mean(pY_0)
## END CODE SET 7
```

Interpretation

The difference between these two means is 50 cells/mL (a 25 cell/mL difference for each time-point, which corresponds to the true effect in our simulated scenario). If we make the same assumptions as in the previous example (counterfactual consistency, no interference, exchangeability, no model mis-specification, positivity), we can interpret this as our causal effect of interest.

Side Note: The parametric g formula is subject to what is known as

[16] Note the evasive language ("some assurance", "suggests", etc). This is because unbiased causal effect estimation is still possible if the natural course and empirical results are very different. It is also possible that a parameter estimate is biased if the natural course and empirical results are identical. Thus, this validation step provides evidence that is neither necessary nor sufficient for valid estimation. However, becuase these scenarios are unlikely to occurr in practice, the evidence provided by this validation step is informative.

the "g null paradox," which arises when the true exposure effect is null. In this setting, it is possible that the parametric g formula will estimate a non-null effect. Not much is known about the g null paradox, but it is currently the topic of active research by several groups.

Before wrapping up, let's take another look at our second simulated example. According to the causal diagram in Figure 4, we should be able to obtain an unbiased estimate of the A_0 and A_1 effects using simple regression models. For example, if we adjust for Z_1, there is no open back-door path from A_1 to Y. If we run the code to do this, we find this is actually the case:

```
# CODE SET 8
round(coef(glm(y~a1+z1,data=D,family=gaussian("identity"))),1)

## (Intercept)          a1          z1
##        94.3        25.0        36.4

# END CODE SET 8
```

Similarly, because there are no confounders of the relation between A_0 and Y, the causal diagram seems to suggest that simply regressing Y against A_0 will give us an unbiased effect estimate (the true effect is 25.0 cells/mL):

```
# CODE SET 9
round(coef(glm(y~a0,data=D,family=gaussian("identity"))),1)

## (Intercept)          a0
##       111.6        27.1

# END CODE SET 9
```

However, doing this overestimates the true effect by 2.1 cells/mL. Why? This is a consequence of feedback between A_0 and A_1. Becuase A_0 affects A_1 indirectly through Z_1, this regression model is estimating the overall effect of A_0 on Y. Thus, the estimate of 27.1 is not wrong *per se*. It is simply quantifying the direct effect of A_0 on Y, **plus** the indirect effect of A_0 on Y via A_1.

Note that while this estimate is not incorrect by itself, if we were intersted in estimating $E(Y^{\bar{a}_1=1} - Y^{\bar{a}_1=0})$, and we added the two estimates from these simple regression models to do this, we would be wrong because we'd be counting a portion of the A_1 effect twice.

References

Daniel, R.M., S.N. Cousens, B.L. De Stavola, M. G. Kenward, and J. A. C. Sterne. 2013. "Methods for Dealing with Time-Dependent Confounding." *Stat Med* 32 (9): 1584–1618.

Hudgens, M. G., and M. E. Halloran. 2008. "Toward Causal Inference with Interference." *J Am Stat Assoc* 103 (482): 832–42.

Naimi, Ashley I., Mireille E. Schnitzer, Erica E. M. Moodie, and Lisa M. Bodnar. 2016. "Mediation Analysis for Health Disparities Research." *American Journal of Epidemiology* 184 (4): 315–24. doi:10.1093/aje/kwv329.