Regression in Time-Fixed Settings

Ashley I Naimi

Spring 2023

Contents

1 Introduction to Regression

Regression is a cornerstone tool of any empirical analysis. It is arguably the most widely used tool in science. Regression models are often deployed in an attempt to understand cause-effect relations between exposures and outcomes of interest, or to obtain predictions for an outcome of interest. Consider a scenario in which we are interested in regressing an outcome Y against a set of covariates X. These covariates can be an exposure combined with a set of confounders needed for identification, or a set of predictors used to create a prediction algorithm via regression. In its most basic (?) formulation, a regression model can be written as[1]:

$$E(Y \mid X) = f(X)$$

In principle, this model is most flexible in that it states that the conditional mean of Y is simply a *arbitrary function* of the covariates X. We are not stating (or assuming) precisely **how** the conditional mean is related to these covariates. Using this model, we might get predictions from to facilitate a decision making process, or obtain a contrast of expected means between two groups.

Because of the flexibility of this model, we may be interested in fitting it to a given dataset. But we can't. There is simply not enough information in this equation for us to quantify $f(X)$, even if we had all the data we could use. In addition to data, we need some "traction" or "leverage" to be able to quantify the function of interest.

[1] Note that, in this formulation, the function of interest on the left hand side of the equation is the conditional mean function, $E(Y \mid X)$. However, there are other options, including hazards, failure times, distribution quantiles, and many more.

2 Ordinary Least Squares

The earliest attempt to find some "traction" to quantify $f(X)$ was proposed in the 1800s. The approach starts by accepting a few tenets[2]. First, we want the difference between the observed Y for any given individual and the fitted values $f(X)$ for that person to be "small." If we didn't care about the direction of these errors (we usually don't), we can square the error and take it's average[3]:

$$E[(Y - f(X))^2]$$

Thus, we can define the "optimal" $f(X)$ as the function of X that minimizes the mean squared error.

[2] Much of this section is based on the (excellent) forthcoming book by Cosma Shalizi (2019).

[3] Recall that *mean squared error* can be be re-written as the sum of the squared bias and the variance: $E[(Y - f(X))^2] = [E(Y) - f(X)]^2 + Var(Y)$. This formulation might help clarify why we like to minimize it.

You may recall that finding the $f(X)$ that minimizes mean squared error can be achieved by taking the derivative of the mean squared error with respect to $f(X)$, setting it to zero, then solving for $f(X)$.

We've made some progress, but without a better sense of what $f(X)$ looks like, we still can't move forward. For example, there are several functions where either the derivative simply does not exist (e.g., if $f(X)$ is discontinuous), or where the derivative is still complex enough that we can't make progress with finding a unique solution for $f(X)$ that minimizes mean squared error (see technical note on nonlinear models).

Early on, it was recognized that if we select $f(X)$ to be *linear* (more technically, *affine*) the problem of finding the optimal $f(X)$ becomes much easier. That is, if we can simplify $f(X) = b_0 + b_1 X$, then we can use calculus and simple algebra to find an optimal *linear* solution set $b_0 = \beta_0, b_1 = \beta_1$ that minimizes MSE.

In the case of this ordinary least squares regression estimator, taking the derivative

$$E[(Y - [\beta_0 + \beta_1 X])^2]$$

with respect to β_0 and β_1 and rearranging with (matrix) algebra, gives us the least squares "normal" equations, which ultimately leads to the ordinary least squares estimator for $\hat{\beta}$ (Rencher, 2000, Shalizi (2019)):

$$\hat{\beta} = (X^\intercal X^{-1}) X^\intercal y$$

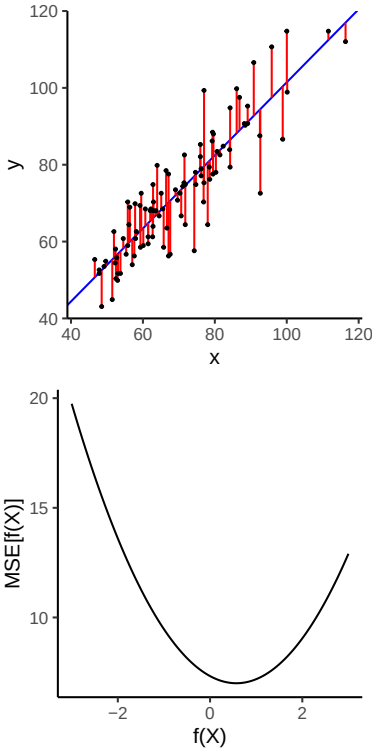

Figure 1: Line of 'best fit' (blue line) defined on the basis of minimizing the sum of squared residuals (red lines) displayed in the top panel; Partial representation of the mean squared error as a function of f(X) in the bottom panel.

Technical Note:

Technically (almost to the point of pedantry), a nonlinear model is a model where the first derivative of the expectation taken with respect to the parameters is itself a function of other parameters (Seber and Wild, 1989). For example,

$$E(Y \mid X) = \beta_0 + \frac{X}{\beta_1}$$

is a nonlinear model, because it's first derivative taken with respect to β_1 is still a function of β_1:

$$\frac{dE(Y \mid X)}{d\beta_1} = \frac{d\left(\beta_0 + \frac{X}{\beta_1}\right)}{d\beta_1} = -\frac{X}{\beta_1^2}$$

Why is this important? Solutions to these regression equations (which serve as our estimates), are obtained by finding where the slope of the tangent line of the parameters is zero. To do this, we need to set the first derivative of these regression equations to zero. But if there are still parameters in these first derivative equations, then there will not be a unique solution to the equation, and finding an estimate will require more complex approaches. This is the complication introduced by nonlinear models.

On the other hand, curvilinear models are easy to find solutions for, since their first derivatives are not functions of parameters. For instance, for a quadratic model such as:

$$E(Y \mid X) = \beta_0 + \beta_1 X + \beta_2 X^2$$

The first derivatives taken with respect to each parameters turn out to be:

$$\frac{dE(Y \mid X, C)}{d\beta_0} = 1$$

$$\frac{dE(Y \mid X, C)}{d\beta_1} = X$$

$$\frac{dE(Y \mid X, C)}{d\beta_2} = X^2$$

Thus, even though the regression function will not be a "straight line" on a plot, this model is still linear.

There are some important points to note in how we formulated the problem of estimating $E(Y \mid X) = f(X)$ with a linear model:

- What we needed to invoke to make this work is that a linear approximation to $f(X)$ is "good enough" for our interest. We need the linear approximation so we can take derivatives of the MSE function without running into problems. These assumptions are usually referred to as "regularity" condi-

tions (Longford, 2008).

- We didn't explicitly state it, but on the basis of Figure 1, this approach makes most sense if Y is continuous. If Y is binary, categorical, or time-to-event, the rationale motivating this approach starts to break down to a degree. That is, while it is possible to fit an OLS estimator to a binary outcome data, not everyone agrees that this is a good idea.
- We *did not* need to invoke any any assumptions about homoscedasticity, or independent and identically distributed (iid) observations. If we were able to make these assumptions, then we obtain an estimator that is the "best linear unbiased estimator" (Rencher, 2000).
- We *did not* need to invoke any distributional assumptions about Y (or, more specifically, the conditional mean of Y). If we can assume Gaussian with constant variance, then we can equate the OLS estimator with the maximum likelihood estimator with a Gaussian distribution and identity link function.

Unfortunately, in the way we formulated it above, the set of linear models we can use to find a solution $f(X)$ that minimizes $MSE(f(X))$ is limited. For instance, in the case where Y is binary, and $E(Y) = P(Y = 1)$, using a linear model such as $b_0 + b_1X$ can easily lead to problems, most notably that predicted probabilities lie outside of the bounds $[0, 1]$.

An additional problem is how we can judge whether the linear combination of parameters in the model is good enough (bullet point 1 above)? As we'll see, this is rarely an easy task. This issue, which we've referred to previously as "correct model specification" is one reason why machine learning methods are becoming so popular.

3 Maximum Likelihood Estimation

Instead of minimizing mean squared error, we could take another approach where we start with a distribution. To demonstrate the motivation here, consider that we, in fact, have a binary outcome ($Y \in [0, 1]$), and we're interested in regressing this outcome against some variable X.

We'll start with a model for the $P(Y = 1)$, which can be modeled using the binomial distribution, defined as:

$$P(Y = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

To understand this equation, say we're interested in understanding the probability of flipping a 50:50 coin "heads" exactly 5 times out of 10 flips.

Here, the total sample size n is 10, and the event number k is 5. With this, we can compute:

$$P(Y = 5) = \binom{10}{5} .5^5 (1-.5)^{10-5} = 0.246$$

In the context of an epidemiologic study, we're usually interested in the probability of a single event (such as a death). Let's say we're interested in understanding how the probability of death ($Y = 1$) is associated with smoking status ($X = 1$). With a single ($k = 1$) binary outcome, this probability function reduces to the Bernoulli distribution:

$$P(Y = 1) = p^y (1-p)^{1-y}$$

where p captures the probability of death. If we believe that p is a function of smoking status X, we can write $p = f(X)$ as we did above, which will give us something like:

$$P(Y = 1 \mid X) = [f(X)]^y (1 - [f(X)])^{1-y}$$

At this point, let's imagine that we have a dataset of three observations:

ID	Y	X
1	1	1
2	0	1
3	1	0

Let's also imagine that the true probability of death is $f(X = 1) = .3$ for smokers, and $f(X = 0) = .15$ for nonsmokers. With the data in the table above, we could compute (among many things) the probability of the observed Y using the equation above. For each row, we get:

$$P(Y = 1 \mid X) = [0.3]^1 (1 - [0.3])^{1-1} = 0.3$$

$$P(Y = 1 \mid X) = [0.3]^0(1 - [0.3])^{1-0} = 0.7$$

$$P(Y = 1 \mid X) = [0.15]^1(1 - [0.15])^{1-1} = 0.15$$

giving us:

ID	Y	X	$p(Y)$
1	1	1	0.3
2	0	1	0.7
3	1	0	0.15

More generally, for a given value of $f(X)$, we can use the data to determine the observed event probability (i.e., if we change the value of $f(X)$, how does $p(Y)$ change?). We can also calculate the overall observed probability of the outcome in the above dataset. If the events are independent, we just need to multiply the probabilities together: $0.3 \times 0.7 \times 0.15 = 0.0315$. Thus, the probability of the observed data given the value of $f(X)$ is 0.0315.

Unfortunately, this ability to compute the observed event probability is rarely useful, since we usually don't know the value of $f(X)$. Maximum likelihood estimation takes the logic of the above scenario and turns it around. Instead of asking: "for a given value of $f(X)$ and a given dataset, what's the observed event probability?" maximum likelihood estimation asks: "for a given dataset and observed event probability, what's the most likely value of $f(X)$?".

To distinguish the Bernoulli function above as a tool to quantify the most likely value of $f(X)$ (as opposed to using it to quantify the probability of Y), we often re-write it as:

$$\mathcal{L}(f(X) \mid Y, X) = \prod_{i=1}^{3} [f(X_i)]^{y_i}(1 - [f(X_i)])^{1-y_i}$$

Notice this is the same equation as above. The only difference is that we're using it to compute the value of $f(X)$ instead of the value of $P(Y = 1 \mid X)$. To understand exactly how we use this equation to compute the optimal value of $f(X)$, we can rewrite it to be more specific to our data:

$$\mathcal{L}(f(X) \mid Y, X) = f(X = 1) \times [1 - f(X = 1)] \times f(X = 0)$$

Written this way, we can now start to interpret this likelihood function as a measure of compatibility between the data and the estimated value of $f(X)$. The higher the likelihood, the more compatible the estimated value of $f(X)$ is with the observed data. This leads us to the idea of the likelihood function, which we can maximize. Similar to the situation with the least squares estimator above (minimizing MSE), we can take the derivative of this likelihood function, set this derivative to zero, and solve for the optimal value of $f(X)$.

The problem is that there are several details that make taking this derivative considerably more complex. When the outcome is binary, we may elect to constrain $f(X)$ so that the predictions are forced to lie within $[0, 1]$. For instance, we could use the inverse of the logistic function and define $f(X) = \frac{1}{1+\exp(-b_0-b_1X)}$.

However, in this case the derivatives would no longer work out as simply as we'd need them too (see Technical Note) because this is a nonlinear function, which means the derivatives of the model $f(X)$ cannot simply be set to zero, even if it is a linear (affine) function. As it turns out, in the early 1970's, Nelder and Wedderburn (Nelder and Wedderburn, 1972) made a seminal contribution that enabled fitting nonlinear models when the outcome belongs to the exponential family of distributions,[4] and the conditional mean of the outcome can be linked to the covariates through some smooth and invertible linearizing function (which includes the $\log$ and logit functions).

[4] The exponential family of distributions is not to be confused with the exponential distribution. It refers to a family of distributions that can be re-written such that they can be represented in a common form. These distributions include (but are not limited to) the Gaussian, exponential, bernoulli (binomial), Poisson, and negative binomial.

4 Generalized Linear Models

Generalized linear models consist of a family of regression models that are fully characterized by a selected distribution and a link function. That is, to fully specify a GLM, one must select a distribution (which determines the form of the conditional mean and variance of the outcome) and a link function (which determines how the conditional mean of the outcome relates to the covariates).

There are a wide variety of distributions and link functions available in standard statistical software programs that fit GLMs. Here, we'll consider a binary outcome Y with probability $P(Y = 1)$, and focus attention on three link functions:

1. Logit, or the log-odds: $\log P(Y = 1)/[1 - P(Y = 1)]$

2. Log: $\log[P(Y = 1)]$
3. Identity: $P(Y = 1)$.

A common misconception is that to use GLMs correctly, one must choose the distribution that best characterizes the data, as well as the canonical link function corresponding to this distribution. For example, if the outcome is binary, one "must" choose the binomial distribution with the logit link. While the binomial distribution and logit link work well together for binary outcomes, they do not easily provide contrasts like the risk difference or risk ratio, because of the selected link function. Alternative specification of the distribution and link function for GLMs can address this limitation.

4.1 Link Functions and Effect Measures

There is an important relation between the chosen link function, and the interpretation of the coefficients from a GLM. For models of a binary outcome and the logit or log link, this relation stems from the properties and rules governing the natural logarithm. The quotient rule states that $\log(X/Y) = \log(X) - \log(Y)$.

Because of this relation, the natural exponent of the coefficient in a logistic regression model yields an estimate of the odds ratio. However, by the same reasoning, exponentiating the coefficient from a GLM with a log link function and a binomial distribution (i.e., log-binomial regression) yields an estimate of the risk ratio. Alternately, for GLM models with a binomial distribution and identity link function, because logarithms are not used, the unexponentiated coefficient yields an estimate of the risk difference.

Unfortunately, using a binomial distribution can lead to convergence problems with the $\log()$ or identity link functions for reasons that have been explored (Zou, 2004). This will occur when, for example, the combined numerical value of all the independent variables in the model is very large. This can result in estimated probabilities that exceed 1, which violates the very definition of a probability (binomial) model (probabilities can only lie between zero and one) and hence, convergence problems. Let's see how these problems can be overcome.

4.2 A Data Example

We use data from the National Health and Nutrition Examination Survey (NHEFS), available as a companion dataset to the Hernán and Robins (Forthcoming) book. We are interested primarily in the covariate adjusted association (on the risk difference and risk ratio scales) between quitting smoking and a greater than median weight change between 1971 and 1982.

In our analyses, we regress an indicator of greater than median weight change against an indicator of whether the person quit smoking. We adjust for exercise status, sex, age, race, income, marital status, education, and indicators of whether the person was asthmatic or had bronchitis. We start by loading the data:

```
#' Load relevant packages
packages <- c("broom", "here", "tidyverse",
    "skimr", "rlang", "sandwich", "boot",
    "kableExtra")

for (package in packages) {
    if (!require(package, character.only = T,
        quietly = T)) {
        install.packages(package, repos = "http://lib.stat.cmu.edu/R/CRAN")
    }
}

for (package in packages) {
    library(package, character.only = T)
}

#' Define where the data are
file_loc <- url("https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1268/1268/20/nhefs.csv")

#' This begins the process of cleaning and formatting the data
nhefs <- read_csv(file_loc) %>%
    select(qsmk, wt82_71, wt82, wt71, exercise,
```

```
        sex, age, race, income, marital,
        school, asthma, bronch, starts_with("alcohol"),
        -alcoholpy, starts_with("price"),
        starts_with("tax"), starts_with("smoke"),
        smkintensity82_71) %>%
    mutate(income = as.numeric(income > 15),
        marital = as.numeric(marital > 2),
        alcoholfreq = as.numeric(alcoholfreq >
            1)) %>%
    na.omit(.)

factor_names <- c("exercise", "income", "marital",
    "sex", "race", "asthma", "bronch")
nhefs[, factor_names] <- lapply(nhefs[, factor_names],
    factor)

#' Define outcome
nhefs <- nhefs %>%
    mutate(id = row_number(), wt_delta = as.numeric(wt82_71 >
        median(wt82_71)), .before = qsmk)

#' Quick summary of data
nhefs %>%
    print(n = 5)
```

```
## # A tibble: 1,055 x 27
##      id wt_delta  qsmk wt82_71  wt82  wt71 exercise sex     age race  income
##   <int>    <dbl> <dbl>   <dbl> <dbl> <dbl> <fct>    <fct> <dbl> <fct> <fct>
## 1     1        0     0  -10.1   68.9  79.0 2        0        42 1     1
## 2     2        0     0    2.60  61.2  58.6 0        0        36 0     1
## 3     3        1     0    4.99  64.4  59.4 2        0        68 1     0
## 4     4        1     0    4.99  92.1  87.1 1        0        40 0     1
## 5     5        1     0    4.42 103.   99   1        1        43 1     0
## # ... with 1,050 more rows, and 16 more variables: marital <fct>, school <dbl>,
## #   asthma <fct>, bronch <fct>, alcoholfreq <dbl>, alcoholtype <dbl>,
```

```
## #   alcoholhowmuch <dbl>, price71 <dbl>, price82 <dbl>, price71_82 <dbl>,
## #   tax71 <dbl>, tax82 <dbl>, tax71_82 <dbl>, smokeintensity <dbl>,
## #   smokeyrs <dbl>, smkintensity82_71 <dbl>
```

4.3 GLMs for risk differences and ratios

For our analyses of the data described above using GLM with a binomial distributed outcome with a log link function to estimate the risk ratio and identity link function to estimate risk difference, an error is returned:

```
#' Here, we start fitting relevant regression models to the data.
#' modelForm is a regression argument that one can use to regress the
#' outcome (wt_delta) against the exposure (qsmk) and selected confounders.

formulaVars <- paste(names(nhefs)[c(3, 7:16)],
    collapse = "+")
modelForm <- as.formula(paste0("wt_delta ~",
    formulaVars))
modelForm
```

```
## wt_delta ~ qsmk + exercise + sex + age + race + income + marital +
##     school + asthma + bronch + alcoholfreq
```

```
#' This model can be used to quantify a conditionally adjusted
#' odds ratio with correct standard error
modelOR <- glm(modelForm, data = nhefs, family = binomial("logit"))
tidy(modelOR)[2, ]
```

```
## # A tibble: 1 x 5
##   term  estimate std.error statistic   p.value
##   <chr>    <dbl>     <dbl>     <dbl>     <dbl>
## 1 qsmk     0.623     0.153      4.07 0.0000471
```

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

```
#' However, error it returns an error and thus does not provide any results.
modelRR_binom <- glm(modelForm, data = nhefs,
    family = binomial("log"))
```

```
## Error: no valid set of coefficients has been found: please supply starting values
```

Why is this error returned? The most likely explanation in this context is as follows: We are modeling $P(Y = 1 \mid X) = \exp\{\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p\}$. In this context, there may be *no set of values* for the parameters in the model that yield $P(Y = 1 \mid X) < 1$ for every observation in the sample. Because R's glm function (under a binomial distribution) correctly recognizes this as a problem, it returns an error.

Instead, one may resort to using different distributions that are more compatible with the link functions that return the association measures of interest. For the risk ratio, one may use a GLM with a Poisson distribution and log link function. Doing so will return an exposure coefficient whose natural exponent can be interpreted as a risk ratio.

```
#' This model can be used to quantify a conditionally risk ratio
#' using the Poisson distribuiton and log link function.
#' However, because the Poisson distribution is used, the model
#' provides incorrect standard error estimates.
modelRR <- glm(modelForm, data = nhefs, family = poisson("log"))
tidy(modelRR)[2, ]
```

```
## # A tibble: 1 x 5
##   term  estimate std.error statistic p.value
##   <chr>    <dbl>     <dbl>     <dbl>   <dbl>
## 1 qsmk     0.277    0.0982      2.82 0.00482
```

It's important to recognize what we're doing here. We are using this model as a tool to quantify the log mean ratio contrasting $P(Y = 1 \mid X_{qsmk} = 1)$ to $P(Y = 1 \mid X_{qsmk} = 0)$ (all other things being equal). However, we should not generally assume that ever aspect of this model is correct. In particular, note that the max predicted probability from this model is 1.087:

```
summary(modelRR$fitted.values)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2214  0.3961  0.4865  0.4995  0.5857  1.0873
```

We can use the `augment` function in the `broom` package to evaluate the distribution of these probabilities (among other things):

```
fitted_dat <- augment(modelRR, type.predict = "response")

fitted_dat
```

```
## # A tibble: 1,055 x 18
##    wt_delta  qsmk exercise sex     age race  income marital school asthma bronch
##       <dbl> <dbl> <fct>    <fct> <dbl> <fct> <fct>  <fct>    <dbl> <fct>  <fct>
##  1        0     0 2        0        42 1     1      0            7 0      0
##  2        0     0 0        0        36 0     1      0            9 0      0
##  3        1     0 2        0        68 1     0      1            5 0      0
##  4        1     0 1        0        40 0     1      0           11 0      0
##  5        1     0 1        1        43 1     0      1            9 0      0
##  6        0     0 2        0        51 0     1      0           10 0      0
##  7        1     0 2        0        43 0     1      0           11 0      0
##  8        1     1 1        0        43 0     1      0           12 0      0
##  9        0     0 2        0        34 0     1      0           12 0      0
## 10        1     0 0        1        47 0     1      0           12 0      0
## # ... with 1,045 more rows, and 7 more variables: alcoholfreq <dbl>,
## #   .fitted <dbl>, .resid <dbl>, .std.resid <dbl>, .hat <dbl>, .sigma <dbl>,
## #   .cooksd <dbl>
```

```
plot_hist <- ggplot(fitted_dat) + geom_histogram(aes(.fitted)) +
    scale_y_continuous(expand = c(0, 0)) +
    scale_x_continuous(expand = c(0, 0))

ggsave(here("figures", "2022_02_21-rr_hist_plot.pdf"),
    plot = plot_hist)
```

This distribution is shown in margin Figure 2. We can also see that there are only two observations in the sample with predicted risks greater than 1.

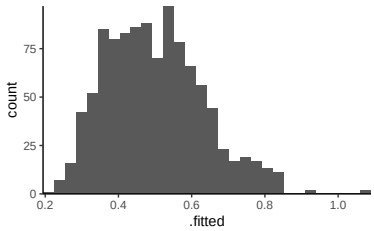

Figure 2: Distribution of fitted values from the Poisson GLM with log link function to obtain an estimate of the adjusted risk ratio for the association between quitting smoking and greater than median weight gain in the NHEFS.

```
fitted_dat %>%
    filter(.fitted >= 1) %>%
    select(wt_delta, qsmk, age, .fitted)
```

```
## # A tibble: 2 x 4
##   wt_delta  qsmk   age .fitted
##      <dbl> <dbl> <dbl>   <dbl>
## 1        1     1    32    1.06
## 2        1     1    25    1.09
```

For these reasons, we are not particularly concerned about the fact that the model predicts risks that are slightly large than 1. However, the model-based standard errors (i.e., the SEs that one typically obtains directly from the GLM output) are no longer valid. Instead, one should use the robust (or sandwich) variance estimator to obtain valid SEs (the bootstrap can also be used) (Zou, 2004).

```
#' To obtain the correct variance, we use the 'sandwich'
#' function to obtain correct sandwich (robust) standard
#' error estimates.
sqrt(sandwich(modelRR)[2, 2])
```

```
## [1] 0.06424196
```

For the risk difference, one may use a GLM with a Gaussian (i.e., normal) distribution and identity link function, or, equivalently, an ordinary least squares estimator. Doing so will return an exposure coefficient that can be interpreted as a risk difference. However, once again the robust variance estimator (or bootstrap) should be used to obtain valid SEs.

```
#' This model can be used to obtain a risk difference
#' with the gaussian distribiton or using ordinary least
#' squares (OLS, via the lm function). Again, the model
```

```
#' based standard error estimates are incorrect.
modelRD <- glm(modelForm, data = nhefs, family = gaussian("identity"))
modelRD <- lm(modelForm, data = nhefs)
tidy(modelRD)[2, ]
```

```
## # A tibble: 1 x 5
##   term  estimate std.error statistic   p.value
##   <chr>    <dbl>     <dbl>     <dbl>     <dbl>
## 1 qsmk     0.145    0.0353      4.10 0.0000438
```

```
#' To obtain the correct variance, we use the 'sandwich' function
#' to obtain correct sandwich (robust) standard error estimates.
sqrt(sandwich(modelRD)[2, 2])
```

```
## [1] 0.03474946
```

The risk ratio and difference, as well as the 95% sandwich variance confidence intervals, obtained for the relation between quitting smoking and greater than median weight change are provided Table 1.

```
knitr::kable(table1_data)
```

Method	Risk Difference	Risk Ratio
GLM	0.14 (0.09, 0.20)	1.32 (1.19, 1.46)
Marginal Standardization	0.14 (0.09, 0.21)	1.31 (1.18, 1.46)

Results in this table obtained using a conditionally adjusted regression model without interactions. Gaussian distribution and identity link was used to obtain the risk difference. A Poisson distribution and log link was used to obtain the risk ratio. 95% CIs obtained via the sandwich variance estimator. 95% CIs obtained using the bias-corrected and accelerated bootstrap CI estimator.

Unfortunately, use of a Poisson or Gaussian distribution for GLMs for a binomial outcome can introduce different problems. For one, while not entirely worrysome in our setting, a model that predicts probabilities greater than one should not instill confidence in the user. Second, performance of the robust variance estimator is notoriously poor with small sample sizes. Finally, the

interpretation of the risk differences and ratios becomes more complex when the exposure interacts with other variables in the model.

Table 3: Methods to use for quantifying conditionally adjusted odds ratios, risk ratios, and risk differences.

Odds Ratio	Risk Ratio	Risk Difference
GLM Family = Binomial	GLM Family = Binomial	GLM Family = Binomial
GLM Link = Logistic	GLM Link = Log	GLM Link = Identity
Standard Errors = Model Based	Standard Errors = Model Based	Standard Errors = Model Based
	GLM Family = Poisson	GLM Family = Gaussian
	GLM Link = Log	GLM Link = Identity
	Standard Errors = Sandwich	Standard Errors = Sandwich
		Least Squares Regression
		Standard Errors = Sandwich

For instance, let's assume that in the NHEFS data, the association between quitting smoking and weight gain interacts with baseline exercise status:

```
#' Potential evidence for interaction
#' between smoking and exercise on the risk difference scale?

table(nhefs$exercise)
```

```
##
##   0   1   2
## 222 465 368
```

```
names(nhefs)[c(3, 7:16)]
```

```
##  [1] "qsmk"        "exercise"    "sex"         "age"         "race"
##  [6] "income"      "marital"     "school"      "asthma"      "bronch"
## [11] "alcoholfreq"
```

```
formulaVars <- paste(names(nhefs)[c(3, 7:16)],
    collapse = "+")

modelForm <- as.formula(paste0("wt_delta ~",
    formulaVars))
modelForm
```

```
## wt_delta ~ qsmk + exercise + sex + age + race + income + marital +
##     school + asthma + bronch + alcoholfreq
```

```
modelForm_int <- as.formula(paste0("wt_delta ~",
    formulaVars, "+ qsmk*exercise"))
modelForm_int
```

```
## wt_delta ~ qsmk + exercise + sex + age + race + income + marital +
##     school + asthma + bronch + alcoholfreq + qsmk * exercise
```

```
summary(glm(modelForm, data = nhefs, family = binomial(link = "identity")))
```

```
##
## Call:
## glm(formula = modelForm, family = binomial(link = "identity"),
##     data = nhefs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8774  -1.1109  -0.4881   1.1171   1.7639
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.933190   0.106882   8.731  < 2e-16 ***
## qsmk         0.142133   0.034391   4.133 3.58e-05 ***
## exercise1   -0.051513   0.039628  -1.300   0.1936
## exercise2   -0.086949   0.042210  -2.060   0.0394 *
## sex1         0.013125   0.030968   0.424   0.6717
```

```
## age         -0.009192   0.001266  -7.258 3.92e-13 ***
## race1       -0.054870   0.044979  -1.220   0.2225
## income1     -0.035566   0.045700  -0.778   0.4364
## marital1     0.022557   0.038809   0.581   0.5611
## school      -0.002138   0.005681  -0.376   0.7067
## asthma1      0.116282   0.066745   1.742   0.0815 .
## bronch1     -0.033874   0.057587  -0.588   0.5564
## alcoholfreq  0.032848   0.030762   1.068   0.2856
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1462.5  on 1054  degrees of freedom
## Residual deviance: 1386.6  on 1042  degrees of freedom
## AIC: 1412.6
##
## Number of Fisher Scoring iterations: 5
```

```
summary(glm(modelForm_int, data = nhefs,
    family = binomial(link = "identity")))
```

```
##
## Call:
## glm(formula = modelForm_int, family = binomial(link = "identity"),
##     data = nhefs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9778  -1.1015  -0.4924   1.1180   1.7670
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.951771   0.107108   8.886  < 2e-16 ***
## qsmk           0.035309   0.082402   0.428   0.6683
```

```
## exercise1      -0.088077   0.045304  -1.944   0.0519 .
## exercise2      -0.104491   0.047673  -2.192   0.0284 *
## sex1            0.015022   0.030954   0.485   0.6275
## age            -0.009107   0.001269  -7.177 7.11e-13 ***
## race1          -0.056297   0.045041  -1.250   0.2113
## income1        -0.034171   0.045592  -0.750   0.4535
## marital1        0.018684   0.038871   0.481   0.6308
## school         -0.002105   0.005676  -0.371   0.7108
## asthma1         0.123167   0.068232   1.805   0.0711 .
## bronch1        -0.042762   0.057493  -0.744   0.4570
## alcoholfreq     0.030062   0.030759   0.977   0.3284
## qsmk:exercise1  0.157140   0.095850   1.639   0.1011
## qsmk:exercise2  0.090399   0.100681   0.898   0.3693
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1462.5  on 1054  degrees of freedom
## Residual deviance: 1383.8  on 1040  degrees of freedom
## AIC: 1413.8
##
## Number of Fisher Scoring iterations: 5
```

If this were the case, to properly interpret the association, the interaction between exercise status and qsmk should be considered. But how could we do this? One approach would be to include the interaction term and interpret the association between quitting smoking and weight change separately for each level of exercise.

For example, in the model that includes the interaction term with exercise, we can no longer simply interpret the coefficient for `qsmk` as the treatment effet of interest. Instead, (under causal identifiability) we have three treatment effects: The effect of `qsmk` for those with `exercise = 0, 1,` and `2`. If we were, in fact, interested in the average treatment effect in the sample (and not a unique treatment effect for each level of exercise), we would have to take a weighted average of the coefficients for these effects, where the weights are

defined as a function of the proportion of individuals in each exercise level.

Clearly, this approach can quickly become too burdensome when there are several relevant interactions in the model, and is not worth the effort when we are interested in the marginal association. As an alternative, we can use marginal or model-based standardization, which can greatly simplify the process.

4.4 Marginal or Model-Based Standardization

Another approach to obtaining risk differences and ratios from GLMs that are not subject to the limitations noted above is to use marginal standardization, which is equivalent to g computation when the exposure is measured at a single time point (Naimi et al., 2017). This process can be implemented by fitting a single logistic model, regressing the binary outcome against all confounder variables, including all relevant interactions. But instead of reading the coefficients the model, one can obtain odds ratios, risk ratios, or risk differences by using this model to generate predicted risks for each individual under "exposed" and "unexposed" scenarios in the dataset. To obtain standard errors, the entire procedure must be bootstrapped (see supplemental material for code). These marginal risk differences and ratios, as well as their bootstrapped CIs are presented in the table above.

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

```
#' Marginal Standardization: version 1
formulaVars <- paste(names(nhefs)[c(3, 7:16)],
    collapse = "+")
modelForm <- as.formula(paste0("wt_delta ~",
    formulaVars, "+ qsmk*exercise"))  ## include the interaction!
modelForm
```

```
## wt_delta ~ qsmk + exercise + sex + age + race + income + marital +
##     school + asthma + bronch + alcoholfreq + qsmk * exercise
```

```r
#' Regress the outcome against the confounders with interaction
ms_model <- glm(modelForm, data = nhefs,
    family = binomial("logit"))
##' Generate predictions for everyone in the sample to obtain
##' unexposed (mu0 predictions) and exposed (mu1 predictions) 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)

#' Using the bootstrap to obtain confidence intervals for the marginally adjusted
#' risk ratio and risk difference.
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)
    res <- c(marg_stand_RD_, marg_stand_RR_,
```

```
        marg_stand_OR_)
    return(res)
}

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

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

```
## Warning in boot.ci(boot_res, index = 1): bootstrap variances needed for
## studentized intervals
```

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

```
## Warning in boot.ci(boot_res, index = 2): bootstrap variances needed for
## studentized intervals
```

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

```
## Warning in boot.ci(boot_res, index = 3): bootstrap variances needed for
## studentized intervals
```

```
marg_stand_OR
```

```
## [1] 1.754113
```

```
marg_stand_RR
```

```
## [1] 1.299942
```

```
marg_stand_RD
```

```
## [1] 0.1389621
```

```
boot_RD
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 1)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 0.0704,  0.2113 )   ( 0.0688,  0.2088 )
##
## Level     Percentile            BCa
## 95%   ( 0.0691,  0.2091 )   ( 0.0720,  0.2129 )
## Calculations and Intervals on Original Scale
```

```
boot_RR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 2)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.137,  1.467 )   ( 1.125,  1.456 )
##
## Level     Percentile            BCa
## 95%   ( 1.144,  1.475 )   ( 1.151,  1.488 )
## Calculations and Intervals on Original Scale
```

```
boot_OR
```

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

```
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 3)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.226,  2.265 )   ( 1.149,  2.188 )
##
## Level     Percentile            BCa
## 95%   ( 1.320,  2.360 )   ( 1.333,  2.399 )
## Calculations and Intervals on Original Scale
```

While this marginal standardization approach is more flexible in that it accounts for the interaction between quitting smoking and exercise, and still yields an estimate of the average treatment effect (again, under identifiability), it still assumes a constant effect of qsmk on weight change across levels of all of the other variables in the model. This constant effect assumption might be true, but if one wanted to account for potential interactions between the exposure and all of the confounders in the model, there is an easy way. We call this the "stratified modeling approach."

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

```
#' Marginal Standardization
##' To avoid assuming no interaction between
##' smoking and any of the other variables
##' in the model, we subset modeling among
##' exposed/unexposed. This code removes smoking from the model,
```

```
##' which will allow us to regress the outcome
##' against the confounders among the exposed and
##' the unexposed searately. Doing so will allow us
##' to account for any potential exposure-covariate interactions
##' that may be present.
formulaVars <- paste(names(nhefs)[c(7:16)],
    collapse = "+")
modelForm <- as.formula(paste0("wt_delta ~",
    formulaVars))
modelForm
```

```
## wt_delta ~ exercise + sex + age + race + income + marital + school +
##     asthma + bronch + alcoholfreq
```

```
#' Regress the outcome against the confounders
#' among the unexposed (model0) and then among the exposed (model1)
model0 <- glm(modelForm, data = subset(nhefs,
    qsmk == 0), family = binomial("logit"))
model1 <- glm(modelForm, data = subset(nhefs,
    qsmk == 1), family = binomial("logit"))
##' Generate predictions for everyone in the sample using the model fit to only the
##' unexposed (mu0 predictions) and only the exposed (mu1 predictions).
mu1 <- predict(model1, newdata = nhefs, type = "response")
mu0 <- predict(model0, newdata = nhefs, type = "response")

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

#' Using the bootstrap to obtain confidence intervals for the marginally adjusted
#' risk ratio and risk difference.
```

```
bootfunc <- function(data, index) {
    boot_dat <- data[index, ]
    model0 <- glm(modelForm, data = subset(boot_dat,
        qsmk == 0), family = binomial("logit"))
    model1 <- glm(modelForm, data = subset(boot_dat,
        qsmk == 1), family = binomial("logit"))
    mu1 <- predict(model1, newdata = boot_dat,
        type = "response")
    mu0 <- predict(model0, newdata = boot_dat,
        type = "response")

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

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

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

```
## Warning in boot.ci(boot_res, index = 1): bootstrap variances needed for
## studentized intervals
```

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

```
## Warning in boot.ci(boot_res, index = 2): bootstrap variances needed for
## studentized intervals
```

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

```
## Warning in boot.ci(boot_res, index = 2): bootstrap variances needed for
## studentized intervals
```

```
marg_stand_OR
```

```
## [1] 1.820838
```

```
marg_stand_RR
```

```
## [1] 1.318331
```

```
marg_stand_RD
```

```
## [1] 0.1478221
```

```
boot_RD
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 1)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 0.0750,  0.2236 )   ( 0.0737,  0.2205 )
##
## Level     Percentile            BCa
## 95%   ( 0.0752,  0.2220 )   ( 0.0770,  0.2245 )
## Calculations and Intervals on Original Scale
```

```
boot_RR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 2)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.147,  1.493 )   ( 1.136,  1.482 )
##
## Level     Percentile            BCa
## 95%   ( 1.154,  1.501 )   ( 1.164,  1.507 )
## Calculations and Intervals on Original Scale
```

```
boot_OR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 2)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.147,  1.493 )   ( 1.136,  1.482 )
##
## Level     Percentile            BCa
## 95%   ( 1.154,  1.501 )   ( 1.164,  1.507 )
## Calculations and Intervals on Original Scale
```

When predicted risks are estimated using a logistic model, relying on marginal standardization will not result in probability estimates outside the bounds [0, 1]. And because the robust variance estimator is not required,

model-based standardization will not be as affected by small sample sizes. However, the bootstrap is more computationally demanding than alternative variance estimators, which may pose problems in larger datasets.

References

M. A. Hernán and JM Robins. *Causal Inference*. Chapman/Hall, Boca Raton, FL, Forthcoming.

NT Longford. *Studying Human Populations: An Advanced Course in Statistics*. Springer, New York, 2008.

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

J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *JRSS-A*, 135 (3):370–384, 1972.

Alvin C. Rencher. *Linear Models in Statistics*. Wiley, New York, 2000.

G. A. F. Seber and C. J. Wild. *Nonlinear regression*. Wiley, New York, 1989.

Cosma Rohilla Shalizi. *The Truth About Linear Regression*. https://www.stat.cmu.edu/ cshalizi/TALR/TALR.pdf, 2019.

Guangyong Zou. A modified poisson regression approach to prospective studies with binary data. *Am J Epidemiol*, 159(7):702–706, Apr 2004.