Introduction to Basic Regression with R

Ashley I Naimi

Oct 2022

Contents

1 Regression in R

In this section, we will look at several ways to estimate an exposure-outcome association adjusting for several potential confounding variables. R offers a great degree of flexibility in fitting models to data. Again, there are many ways to do the same thing in R. This section will seek to provide a way forward, and demonstrate how to use outcome regression modeling, and propensity score regression modeling to estimate associations.

Suppose we wanted to use the NHEFS data to estimate the confounder adjusted effect of quitting smoking on weight change (continuous) and death (binary). We can do this using the analytic dataset we created in the previous section. First, we'll load the relevant libraries needed to conduct our analysis.

```
packages <- c("tidyverse", "here", "broom",
    "boot")

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

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

In the packages above, we've already been introduced to elements of the `tidyverse` and the `here` package. The `broom` package offers tools to extract information from generalized linear models, and create datsets with them. In effect, it allows us to extract regression information in a neat and tidy way. Finally, the `boot` package allows us to implement the bootstrap when needed.[1]

First, we'll load the relevant analytic dataset that we created in the previous section:

[1] Though there are many ways to implement the bootstrap, and we need not always use the `boot` package to do this.

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

```
## Rows: 1476 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (9): seqn, qsmk, sex, age, race, income, wt82_71, death, map
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dim(nhefs)
```

```
## [1] 1476    9
```

```
nhefs %>%
    print(n = 5)
```

```
## # A tibble: 1,476 x 9
##    seqn  qsmk   sex   age  race income wt82_71 death   map
##   <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>   <dbl> <dbl> <dbl>
## 1   233     0     0    42     1     19  -10.1      0 122.
## 2   235     0     0    36     0     18    2.60     0  94.3
## 3   244     0     1    56     1     15    9.41     0  88.3
## 4   245     0     0    68     1     15    4.99     1 101.
## 5   252     0     0    40     0     18    4.99     0  90.7
## # ... with 1,471 more rows
```

2 Conditionally Adjusted Reegression Model

We'll start with a regression model that allows us to estimate the association between quitting smoking and weight change:

```
#' Here, we start fitting relevant regression models to the data.
```

```r
#' This model can be used to quantify a conditionally adjusted
#' mean difference with correct standard error
model_MD <- glm(wt82_71 ~ qsmk + sex + age +
    race + income + map, data = nhefs, family = gaussian("identity"))

#' summary() is a base R function that reports the fit of a
#'  regression model neatly in the console
summary(model_MD)
```

```
##
## Call:
## glm(formula = wt82_71 ~ qsmk + sex + age + race + income + map,
##     family = gaussian("identity"), data = nhefs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -43.326   -3.864   -0.016    4.078   46.050
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.91357    2.34139   0.817    0.414
## qsmk         2.83202    0.45922   6.167 8.98e-10 ***
## sex         -0.06788    0.39735  -0.171    0.864
## age         -0.17504    0.01694 -10.331  < 2e-16 ***
## race        -0.26537    0.60902  -0.436    0.663
## income       0.05549    0.07769   0.714    0.475
## map          0.07065    0.01751   4.034 5.76e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 56.07977)
##
##     Null deviance: 90783  on 1475  degrees of freedom
## Residual deviance: 82381  on 1469  degrees of freedom
## AIC: 10141
```

```
##
## Number of Fisher Scoring iterations: 2
```

```
#' tidy() is a broom function that output the fit of a
#' regression model as a tidy dataset
tidy(model_MD)
```

```
## # A tibble: 7 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   1.91      2.34       0.817 4.14e- 1
## 2 qsmk          2.83      0.459      6.17  8.98e-10
## 3 sex          -0.0679    0.397     -0.171 8.64e- 1
## 4 age          -0.175     0.0169   -10.3   3.37e-24
## 5 race         -0.265     0.609     -0.436 6.63e- 1
## 6 income        0.0555    0.0777     0.714 4.75e- 1
## 7 map           0.0707    0.0175     4.03  5.76e- 5
```

Using the tidy function, we can save the estimates and standard errors that we want to an object in R.

```
mean_difference1 <- tidy(model_MD)[2, ]

mean_difference1
```

```
## # A tibble: 1 x 5
##   term  estimate std.error statistic  p.value
##   <chr>    <dbl>     <dbl>     <dbl>    <dbl>
## 1 qsmk      2.83     0.459      6.17 8.98e-10
```

Next, lets estimate the effect of quitting smoking on death using a conditionally adjusted logistic regression model

```
#' Here, we start fitting relevant regression models to the data.

#' This model can be used to quantify a conditionally adjusted
```

```r
#' mean difference with correct standard error
model_OR <- glm(death ~ qsmk + sex + age +
    race + income + map, data = nhefs, family = binomial("logit"))

#' summary() is a base R function that reports the fit of a
#'  regression model neatly in the console
summary(model_OR)
```

```
##
## Call:
## glm(formula = death ~ qsmk + sex + age + race + income + map,
##     family = binomial("logit"), data = nhefs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1414  -0.5626  -0.3180  -0.1710   3.0078
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.071678   0.973336  -6.238 4.43e-10 ***
## qsmk         0.002634   0.178800   0.015 0.988246
## sex         -0.600305   0.161616  -3.714 0.000204 ***
## age          0.108389   0.008022  13.511  < 2e-16 ***
## race        -0.014473   0.236661  -0.061 0.951235
## income      -0.137948   0.029553  -4.668 3.04e-06 ***
## map          0.021474   0.006789   3.163 0.001561 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1398.6  on 1475  degrees of freedom
## Residual deviance: 1032.3  on 1469  degrees of freedom
## AIC: 1046.3
##
```

```
## Number of Fisher Scoring iterations: 5
```

```
#' tidy() is a broom function that output the fit of a
#' regression model as a tidy dataset
tidy(model_OR)
```

```
## # A tibble: 7 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) -6.07      0.973     -6.24   4.43e-10
## 2 qsmk         0.00263   0.179      0.0147 9.88e- 1
## 3 sex         -0.600     0.162     -3.71   2.04e- 4
## 4 age          0.108     0.00802   13.5    1.35e-41
## 5 race        -0.0145    0.237     -0.0612 9.51e- 1
## 6 income      -0.138     0.0296    -4.67   3.04e- 6
## 7 map          0.0215    0.00679    3.16   1.56e- 3
```

To get the OR from the regression model, we have to exponentiate the coefficient from the model output.

```
qsmk_OR <- tidy(model_OR)[2, ]

qsmk_OR[1, 2] <- exp(qsmk_OR[1, 2])

qsmk_OR
```

```
## # A tibble: 1 x 5
##   term  estimate std.error statistic p.value
##   <chr>    <dbl>     <dbl>     <dbl>   <dbl>
## 1 qsmk      1.00     0.179    0.0147   0.988
```

If we were interested in estimating conditionally adjusted risk differences or risk ratios for the effect of quitting smoking on death, we could use a similar approach with the identity link function, ordinary least squares, or Poisson regression (Zou, 2004, Naimi and Whitcomb (2020)).

The procedures above constitute general procedures in which we can use the `glm` function to estimate assocaitions. However, before proceeding further,

it is useful to explore exactly what happens in R when we fit the glm function. We can explore this, in part, by looking at the contents of the fit from the models.

We can do this easily with the `str()` function. However, we won't look at the output here because it takes up several pages:

```
str(model_MD)


str(model_OR)
```

3 Marginally Adjusted Regression Model

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

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

```
#' Regress the outcome against the confounders with interaction
model_MD <- glm(wt82_71 ~ qsmk + sex + age +
    race + income + map, data = nhefs, family = gaussian("identity"))
##' Generate predictions for everyone in the sample to obtain
##' unexposed (mu0 predictions) and exposed (mu1 predictions) risks.
mu1 <- predict(model_MD, newdata = transform(nhefs,
    qsmk = 1), type = "response")
mu0 <- predict(model_MD, newdata = transform(nhefs,
    qsmk = 0), type = "response")

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

```
#' Using the bootstrap to obtain confidence intervals for the marginally adjusted
#' risk ratio and risk difference.
bootfunc <- function(data, index) {
    boot_dat <- data[index, ]
    model_MD_ <- glm(wt82_71 ~ qsmk + sex +
        age + race + income + map, data = boot_dat,
        family = gaussian("identity"))
    mu1_ <- predict(model_MD_, newdata = transform(boot_dat,
        qsmk = 1), type = "response")
    mu0_ <- predict(model_MD_, newdata = transform(boot_dat,
        qsmk = 0), type = "response")

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

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

boot_MD <- boot.ci(boot_res)

marg_stand_MD
```

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

```
boot_MD
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res)
```

```
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.835,  3.812 )   ( 1.866,  3.793 )
##
## Level     Percentile            BCa
## 95%   ( 1.871,  3.798 )   ( 1.862,  3.794 )
## Calculations and Intervals on Original Scale
```

We can do the same thing to estimate the association between quitting smoking and death:

```
#' Regress the outcome against the confounders with interaction
model_OR <- glm(death ~ qsmk + sex + age +
    race + income + map, data = nhefs, family = binomial("logit"))
##' Generate predictions for everyone in the sample to obtain
##' unexposed (mu0 predictions) and exposed (mu1 predictions) risks.
mu1 <- predict(model_OR, newdata = transform(nhefs,
    qsmk = 1), type = "response")
mu0 <- predict(model_OR, newdata = transform(nhefs,
    qsmk = 0), type = "response")

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

#' Using the bootstrap to obtain confidence intervals for the marginally adjusted
#' risk ratio and risk difference.
bootfunc <- function(data, index) {
    boot_dat <- data[index, ]
    model_OR_ <- glm(death ~ qsmk + sex +
        age + race + income + map, data = boot_dat,
```

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

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

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

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

marg_stand_OR
```

```
## [1] 1.001926
```

```
marg_stand_RR
```

```
## [1] 1.001576
```

```
marg_stand_RD
```

```
## [1] 0.0002860808
```

```
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%   (-0.0379,  0.0389 )   (-0.0400,  0.0378 )
##
## Level     Percentile            BCa
## 95%   (-0.0373,  0.0406 )   (-0.0354,  0.0424 )
## 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%   ( 0.786,  1.213 )   ( 0.765,  1.200 )
##
## Level     Percentile            BCa
## 95%   ( 0.803,  1.238 )   ( 0.814,  1.249 )
## 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%   ( 0.735,  1.259 )   ( 0.705,  1.238 )
##
## Level     Percentile            BCa
## 95%   ( 0.766,  1.299 )   ( 0.780,  1.314 )
## Calculations and Intervals on Original Scale
```

References

Ashley I Naimi and Brian W Whitcomb. Estimating risk ratios and risk differences using regression. *American Journal of Epidemiology*, 189(6):508–510, 2020.

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

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