**Supplement “Risk Ratio regression - simple concept yet complex computation” by Murthy N Mittinty and John Lynch**

**S1**

Let’s take a look at the error of approximating the OR to RR in terms of probabilities in a slightly different way. The RR is usually computed as , where Y=1 is having outcome and X=1 is referred to as having the exposure. P(Y=1|X=1) is the conditional probability of having an outcome given that the respondent/individual has been exposed. RR is the ratio of the outcome among the exposed to risk of the outcome among the unexposed. Similarly, the odds ratio (OR) is expressed as . Now, the relation between OR and RR can be rederive it as

Thus, the relative approximation error is then given as

**Case where**

When RR < 1 and for a fixed baseline prevalence, say , the relative approximation error is , with the largest discrepancy (corresponding to the smallest value of OR/RR as ).

**Case where**

The case when , for fixed the relative error increases without limit as the

Suppose and then using we get the value as .

Say if we have and the prevalence is 0.0001 then we . However, if the prevalence is 0.1 and is 5 then we have Relative error is 80% here. That is in the worst-case scenario the OR is 1.8 times as large as the RR. Here the question can be what is and how does one get this value? Suppose that we know the value of the OR, for example from case-control data, but we are interested in the RR. Further suppose that we know the baseline prevalence (), we believe that if the alternative is true then the , and that we have an idea about the maximum possible size of the RR, which we will call ; This could be based on experience with this disease or others with similar aetiology. In other words, is a worst-case scenario in terms of the approximation error. If in fact the RR is much closer to 1 than then the approximation error in using the OR in place of the RR will be small.

**S2**

Let *Y* be the outcome and *X* be the exposure, assume that the exposure is a binary variable and the outcome is also a binary variable. The simple cross tabulation of these variables yields a 2x2 table as

|  |  |  |  |
| --- | --- | --- | --- |
| Outcome | Exposure | | Total |
|  |  |
|  |  |  | 2 = |
|  |  |  | 16 = |
| Total |  |  |  |

, from the above table one may note that these numbers are nothing but the frequencies of being exposed and unexposed. Now if these counts (*a, b, c,* and *d*) are divided by total sample then the above RR is expressed in terms of probability as . This shows that whenever the counts *a* and *c* are equal or also rare RR is estimating the ratio of the exposure distribution.

Let’s look at the formula in terms of the probability. Count *a* refers to the joint distribution of having exposure and also the outcome, denote this as When we divide *a* by the total number we can call the fraction as the joint probability denoted as Similarly, *b/N* is ; and *d/N* is . We now are ready to estimate OR in terms of probabilities from the above 2x2 table as.

The final estimate, in this example, OR is equating to the ratio of the joint distribution of the absence of outcome when exposed and unexposed. These two estimates, in this example, as one can see are estimating two different things and hence must not be approximated. This is because they are answering two different questions 1) What is the ratio of having an exposure to not having an exposure (RR case) and 2) What is the joint probability of not having an outcome when not exposed to the joint probability of not having an outcome when exposed (OR case). Compared to 1) What is the ratio of the probability of having the disease if exposed to the probability of having the disease if not exposed and 2) What is the ratio of the odds of having the disease when exposed to the odds of having the disease when not exposed. Hence one need to be careful when interpreting or approximating these values.

**S 3**

**Derivation of the log likelihood**

Maximum likelihood estimation of the log-binomial model is

where

**S 4**

**Risk Ratio (RR) estimation in Stata and R**

The following section will provide details of estimating RR using the listed approaches, which have been classified into three groups: 1) Mantel-Haenszel stratified method for a small set of covariates (in this same group, we also have listed the log-binomial models and variants of log-binomial); 2) logbin regression with various computational methods; and 3) the Non-GLM, doubly robust method.

**#Reading data into R and making necessary manipulation to attain results we presented in Table 1 of the manuscript**

*Nd<- url("https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1268/1268/20/nhefs.csv")*

*nd<-read.csv(Nd)*

#data manipulations implemented to make income, marital status and outcome as binary variables

*nd$incomeb<-ifelse(nd$income>15,1,0) #income binary*

*nd$maritalb<-ifelse(nd$marital>2,1,0) #marital status binary*

*nd$wtb<-ifelse(nd$wt82\_71>median(nd$wt82\_71,na.rm=TRUE),1,0) #weight binary*

*nd<subset(nd,select=c(qsmk,wtb,exercise,sex,age,race,incomeb,maritalb,school,asthma,bronch))*

*factor\_names <- c("exercise","incomeb","maritalb","sex","race","asthma","bronch")*

*nd[,factor\_names] <- lapply(nd[,factor\_names] , factor)*

*formulaVars <- paste(names(nd)[-c(2)],collapse = "+")*

*modelForm <- as.formula(paste0("wtb ~", formulaVars))*

*modelForm*

**Copy the data into Stata to do the Mantel Haenszel and GLM models**

**Mantel Haenszel, log-binomial and variants**

*Method: Stratified Mantel Haenszel method*

Useful when there is one confounder apart from the exposure. Stata code is

*Clear*

*Use nd*

*cs wtb qsmk, by(sex) pool*

**Going back to R program for conducting the GLM, logbin and BRM methods**

**Log-Binomial**

*Method: Log binomial model*

Log-binomial is a natural choice for developing log linear model and directly estimate RR. When GLM methods have convergence issues then. RR can be estimated using the following method

**Standard form if there is no convergence issue**

*bin\_id <- glm(modelForm,data=nd,family = binomial("log"))*

*bin\_id*

If the above command has convergence issues then one can force convergence using the following code

*bin\_id <- glm(modelForm,data=nd,family = binomial("log"),start=c(log(846/1629),rep(0,11)))*

*bin\_id*

**S 4**

**Alternate methods**

*Method: Logbin regression in R.*

*#installing packages*

*install.packages("logbin")*

*install.packages("sandwich")*

*library(sandwich)*

*library(logbin)*

#logbin regression with adaptive barrier (constrained optimisation) computational method

*start.p<-c(log(846/1629),cf)*

*fit.logbin <- logbin(formula(bin\_id), data = nd,*

*start = start.p, trace = 1,method="ab")*

#Extracting starting values from a Poisson model (we used these in the model)

*modelRR <- glm(modelForm,data=nd,family = poisson("log"))*

*cf<-modelRR$coefficients*

*cf<-cf[-1]*

#logbin regression with the Expectation maximization algorithm

*start.p<-c(log(846/1629),cf)*

*fit.logbin <- logbin(formula(bin\_id), data = nd,*

*start = start.p, trace = 1,method="ab")*

*fit.logbin.em <- update(fit.logbin, method = "em")*

# Speed up convergence by using acceleration methods

*fit.logbin.em.acc <- update(fit.logbin.em, accelerate = "squarem")*

*fit.logbin.em.acc*

**S 4**

*Method: binary regression model in R*

*install.packages("brm")*

*library(brm)*

*y<-nd$wtb*

*x<-nd$qsmk*

*v<-nd[,-c(1,2)]*

*int<-rep(1,nrow(v))*

*v<-cbind(int=int,v)*

*v<-as.matrix(v)*

*fit.mle=brm(y,x,v,v,'RR','MLE',v,TRUE)*

*fit.mle*

*fit.drw = brm(y,x,v,v,'RR','DR',v,TRUE)*

*fit.dru = brm(y,x,v,v,'RR','DR',v,FALSE)*

***The parameter RR its mean and standard deviation can be computed using the following code***

*mean(fit.drw$param.est)*

*mean(fit.dru$param.est)*

*sd(fit.drw$param.est)*

*sd(fit.dru$param.est)*