Obtain variance for relative scale difference-in-difference

Jade Benjamin-Chung

10/25/2019

Objective

When using count data to obtain estimates of incidence rates, log-linear models are commonly since a population-offset can be conveniently included in the model, so individual-level data is not required. This document uses the Delta Method to obtain the standard error for a relative scale difference-in-difference parameter using the coefficients from a log-linear model, fit as follows:

$\ln\left(\frac{E[Y|X,T]}{PT}\right) = \beta_0 + \beta_1 X + \beta_2 T + \beta_3 X \cdot T$

- Y: count outcome
- PT: person-time used to calculate rate
- X: indicator for treatment
- T: indicator for time

The parameter of interest is:

$$\text{RDID} = 1 - \left(\frac{(E[Y|X=1,T=1] - E[Y|X=1,T=0])}{E[Y|X=0,T=1] - E[Y|X=0,T=0]}\right) \times 100$$

Using the coefficients from the model above, the transformation function is:

$$F(\beta) = \left(1 - \frac{e^{\beta_0+\beta_1+\beta_2+\beta_3} - e^{\beta_0+\beta_1}}{e^{\beta_0+\beta_2} - e^{\beta_0}}\right) \times 100$$

Calculate relative scale difference-in-difference by hand

```
# Pre-program numerator and denominator
data %>% group_by(dist) %>%
  filter(seas>=1112 & seas<1415) %>%
  group_by(seas, dist) %>%
  summarise(flucases = sum(flucases), pop = sum(N)) %>%
  group_by(dist) %>%
  summarise(flucases = mean(flucases), pop = mean(pop))
```

```
## # A tibble: 2 x 3
##   dist   flucases    pop
##   <fct>     <dbl>  <dbl>
## 1 OUSD       117. 447832
## 2 WCCUSD      56  255318
```

```
# 2014-15 numerator and denominator
data %>% filter(seas==1415) %>%
  group_by(dist) %>%
  summarise(flucases = sum(flucases), pop = sum(N))
```

```
## # A tibble: 2 x 3
##   dist    flucases    pop
##   <fct>      <dbl>  <dbl>
## 1 OUSD         158 447832
## 2 WCCUSD       113 255318
```

```r
# relative scale DID
(((158/447832) -(117/447832))  / ((113/255318) -(56/255318)))
```

```
## [1] 0.4100863
```

```r
# relative scale DID x 100%
(1-(((158/447832) -(117/447832))  / ((113/255318) -(56/255318)))) * 100
```

```
## [1] 58.99137
```

Obtain ratio of incidence ratios using Poisson model

```r
glm.fit=glm(flucases ~ tr*time,offset=logN,data=d1415,
             family=poisson(link=log))

glm.fit
```

```
##
## Call:  glm(formula = flucases ~ tr * time, family = poisson(link = log),
##     data = d1415, offset = logN)
##
## Coefficients:
## (Intercept)           tr         time      tr:time
##     -8.4249       0.1721       0.7020      -0.3988
##
## Degrees of Freedom: 1627 Total (i.e. Null);  1624 Residual
## Null Deviance:       2355
## Residual Deviance: 2314  AIC: 3548
```

Use Delta Method to get variance for relative-scale difference-in-difference

Step 1: Get partial derivatives of $F(\beta)$ with respect to each parameter

$$\frac{dF}{d\beta_0} = 0$$

$$\frac{dF}{d\beta_1} = \frac{e^{\beta_2}(e^{\beta_2+\beta_3} - 1)}{(e^{\beta_2} - 1)^2}$$

$$\frac{dF}{d\beta_2} = \frac{(e^{\beta_3} - 1)e^{\beta_1+\beta_2}}{(e^{\beta_2} - 1)^2}$$

$$\frac{dF}{d\beta_3} = \frac{e^{\beta_1+\beta_2+\beta_3}}{1 - e^{\beta_2}}$$

```r
# Gradient
b0 = glm.fit$coefficients[1]
b1 = glm.fit$coefficients[2]
b2 = glm.fit$coefficients[3]
b3 = glm.fit$coefficients[4]

dfdb0 = 0
dfdb1 = -(exp(b1) * (exp(b2 + b3) - 1)) / ((exp(b2) -1)^2)
dfdb2 = ((exp(b3)-1)*exp(b1+b2))/((exp(b2)-1)^2)
dfdb3 = (exp(b1+b2+b3))/(1-exp(b2))

grad <- c(dfdb0, dfdb1, dfdb2, dfdb3)
names(grad) <- c("dfdb0", "dfdb1", "dfdb2", "dfdb3")
grad
```

```
##      dfdb0      dfdb1      dfdb2      dfdb3
##  0.0000000 -0.4061674 -0.7607450 -1.5803328
```

Step 2: Get the covariance variance matrix from regression output

```r
vb <- vcov(glm.fit)
vb
```

```
##              (Intercept)           tr         time      tr:time
## (Intercept)  0.005952352 -0.005952352 -0.005952352  0.005952352
## tr          -0.005952352  0.008809490  0.005952352 -0.008809490
## time        -0.005952352  0.005952352  0.014801884 -0.014801884
## tr:time      0.005952352 -0.008809490 -0.014801884  0.023988043
```

Step 3: Calculate the variance

$Var(F(\beta)) = JVJ^T$, where J is the Jacobian or gradient (derivative of $F(\beta)$ with respect to each parameter) and V is the variance covariance matrix.

```r
vF <- t(grad) %*% vb %*% grad
vF
```

```
##            [,1]
## [1,] 0.02670726
```

```r
seF <- sqrt(vF)
seF
```

```
##           [,1]
## [1,] 0.1634235
```

Obtain relative scale difference-in-difference and 95% CI from log-linear model

```r
get_rdid = function(fit){

# Gradient
b0 = fit$coefficients[1]
```

```r
b1 = fit$coefficients[2]
b2 = fit$coefficients[3]
b3 = fit$coefficients[4]

dfdb0 = 0
dfdb1 = -(exp(b1) * (exp(b2 + b3) - 1)) / ((exp(b2) -1)^2)
dfdb2 = ((exp(b3)-1)*exp(b1+b2))/((exp(b2)-1)^2)
dfdb3 = (exp(b1+b2+b3))/(1-exp(b2))

grad <- c(dfdb0, dfdb1, dfdb2, dfdb3)
names(grad) <- c("dfdb0", "dfdb1", "dfdb2", "dfdb3")
grad

# Estimate relative scale DID
rr = (exp(b0+b1+b2+b3) - exp(b0+b1))/(exp(b0+b2) - exp(b0))
rdid = (1-rr) * 100
print(paste("RR adjusted for pre-intervention = ",
            sprintf("%0.02f",rr), sep = " "))
print(paste("Relative scale DID = ", sprintf("%0.0f",rdid), sep = " "))

# Variance-covariance matrix
vb <- vcov(fit)
vb

# Variance of DID
vF <- t(grad) %*% vb %*% grad
vF

se <- sqrt(vF)

# 95% CI
lb = log(rr) - (qnorm(0.975)*sqrt(vF))
ub = log(rr) + (qnorm(0.975)*sqrt(vF))

rdid = (1 - rr)*100
lb = (1 - exp(lb))*100
ub = (1 - exp(ub))*100

return(list = c(rdid = rdid, se = se, lb = ub, ub = lb))

}

get_rdid(glm.fit)
```

```
## [1] "RR adjusted for pre-intervention =  0.41"
## [1] "Relative scale DID =  59"
```

```
## rdid.(Intercept)               se               lb               ub
##       58.6579616        0.1634235       43.0494055       69.9886515
```

Use deltamethod package to obtain relative scale difference-in-difference and 95% CI from log-linear model

```
g = as.formula(~  (exp(x1+x2+x3+x4) - exp(x1+x2)) / (exp(x1+x3) - exp(x1)))

deltamethod(g = g, mean = coef(glm.fit), cov = vcov(glm.fit))
```

```
## [1] 0.1631667
```