Complex methods for complex data: key considerations for interpretable and actionable results in exposome research: accompanying material

Illustration of basic analytical steps for incorporating social constructs in exposome research

Marta Ponzano, Ran Rotem, Andrea Bellavia

We present an illustrative example of the analytical steps for handling non-modifiable social constructs in exposome research. Specifically, this example will consider a variable *X*, representing a social construct for which a disparity in outcome *Y* is observed, and an environmental exposure *E* that partially contributes to that disparity. An additional variable *P* is used to describe proxies of the social constructs that could be used, when available, to replace *X*. Stata code presented below can be used to simulate the data (Part 1), evaluate regression models that adjust or not for the social construct and the proxy (part 2), and assess disparities through stratification (part 3).

PART 1: Data simulation

```
clear all
scalar n_iter=1000
scalar  Y_E_unadjusted_1000     =0
scalar  Y_E_products_P_1000 =0
scalar  Y_E_X_1000           =0
scalar  Y_E_products_1000       =0
scalar  Y_E_products_X_1000  =0
matrix define V_Y_E_unadjusted_1000       =J(1000, 1, 0)
matrix define V_Y_E_products_P_1000   =J(1000, 1, 0)
matrix define V_Y_E_X_1000             =J(1000, 1, 0)
matrix define V_Y_E_products_1000         =J(1000, 1, 0)
matrix define V_Y_E_products_X_1000    =J(1000, 1, 0)

scalar  Y_E_X1_1000           =0
scalar  Y_E_X0_1000           =0
matrix define V_Y_E_X1_1000   =J(1000, 1, 0)
matrix define V_Y_E_X0_1000   =J(1000, 1, 0)

scalar  _E_median_overall_1000             =0
scalar  _E_quartile1_overall_1000          =0
scalar  _E_quartile3_overall_1000          =0
scalar  _Y_median_overall_1000              =0
scalar  _Y_quartile1_overall_1000           =0
scalar  _Y_quartile3_overall_1000           =0
scalar  _E_median_X1_1000              =0
scalar  _E_quartile1_X1_1000           =0
scalar  _E_quartile3_X1_1000           =0
scalar  _Y_median_X1_1000               =0
scalar  _Y_quartile1_X1_1000            =0
scalar  _Y_quartile3_X1_1000            =0

scalar  _E_median_X0_1000            =0
scalar  _E_quartile1_X0_1000         =0
scalar  _E_quartile3_X0_1000         =0
scalar  _Y_median_X0_1000             =0
scalar  _Y_quartile1_X0_1000          =0
scalar  _Y_quartile3_X0_1000          =0

forvalues iter = 1/1000 {
```

```
set more off
set obs 10000
set seed   `iter'

gen X =.
replace X=rbinomial(1,0.19)

 gen products=.
replace products=rbinomial(1,0.494) if X==1
replace products=rbinomial(1,0.077) if X==0

gen P=.
replace P=rnormal(4, 3) if  X==1
replace P=rnormal(0, 3) if  X==0
 gen int_products_P=products*P

scalar  beta_E                =61
scalar  beta_products_E       =20
scalar  beta_P_E          =0.1
scalar  beta_P_products_E =0.5

gen E=rnormal(beta_E+ ///
                beta_products_E*products+   ///
                           beta_P_E*P+ ///
                           beta_P_products_E*int_products_P, 12)

 gen int_P_E   =P*E
 gen int_products_E=products*E

xtile E_QUART = E, nq(4)
tab E_QUART, generate(E_QUART)

 scalar beta_Y                =3490
scalar beta_E_QUART2_Y     =-34.6
scalar beta_E_QUART3_Y     =-200.2
scalar beta_E_QUART4_Y     =-72.1
scalar beta_products_Y       =-54
 scalar beta_products_E_Y   =-2
scalar beta_P_Y          =-2
scalar beta_P_products_Y =-0.5
scalar beta_P_E_Y      =-0.005

gen Y=rnormal(beta_Y+ ///
                      beta_E_QUART2_Y*E_QUART2+ ///
                      beta_E_QUART3_Y*E_QUART3+ ///
                       beta_E_QUART4_Y*E_QUART4+ ///
                        beta_products_Y*products+ ///
                        beta_products_E_Y*int_products_E + ///
                        beta_P_Y*P + ///
                        beta_P_products_Y*int_products_P + ///
                        beta_P_E_Y*int_P_E   ///
                        , 450)
```

PART 2: Regression models for outcome Y as a function of exposure E with and without the inclusion of the social construct X

```
regress Y E
 matrix b=e(b)
matrix V_Y_E_unadjusted_1000[`iter',1]=(b[1,1])
scalar  Y_E_unadjusted_1000=Y_E_unadjusted_1000+b[1,1]

 regress Y E products  P
 matrix b=e(b)
 matrix V_Y_E_products_P_1000[`iter',1]=(b[1,1])
scalar  Y_E_products_P_1000=Y_E_products_P_1000+b[1,1]
regress Y E X
```

```
 matrix b=e(b)
 matrix V_Y_E_X_1000[`iter',1]=(b[1,1])
scalar  Y_E_X_1000=Y_E_X_1000+b[1,1]

regress Y E products
 matrix b=e(b)
 matrix V_Y_E_products_1000[`iter',1]=(b[1,1])
scalar  Y_E_products_1000=Y_E_products_1000+b[1,1]

 regress Y E products X
 matrix b=e(b)
 matrix V_Y_E_products_X_1000[`iter',1]=(b[1,1])
scalar  Y_E_products_X_1000=Y_E_products_X_1000+b[1,1]
```

PART 3: Assessment of the disparity by stratification of regression model as well as exposure and outcome distributions over levels of X

```
 regress Y E if X==0
 matrix b=e(b)
 matrix V_Y_E_X0_1000[`iter',1]=(b[1,1])
 scalar  Y_E_X0_1000=Y_E_X0_1000+b[1,1]

 regress Y E if X==1
 matrix b=e(b)
 matrix V_Y_E_X1_1000[`iter',1]=(b[1,1])
 scalar  Y_E_X1_1000=Y_E_X1_1000+b[1,1]

 sum   E     , det
 scalar  _E_median_overall_1000=_E_median_overall_1000+r(p50)
 scalar  _E_quartile1_overall_1000=_E_quartile1_overall_1000+r(p25)
 scalar  _E_quartile3_overall_1000=_E_quartile3_overall_1000+r(p75)

 sum   Y     , det
 scalar  _Y_median_overall_1000=_Y_median_overall_1000+r(p50)
 scalar  _Y_quartile1_overall_1000=_Y_quartile1_overall_1000+r(p25)
 scalar  _Y_quartile3_overall_1000=_Y_quartile3_overall_1000+r(p75)

 sum   E if X==0 , det
 scalar  _E_median_X0_1000=_E_median_X0_1000+r(p50)
 scalar  _E_quartile1_X0_1000=_E_quartile1_X0_1000+r(p25)
 scalar  _E_quartile3_X0_1000=_E_quartile3_X0_1000+r(p75)

 sum   Y if X==0  , det
 scalar  _Y_median_X0_1000=_Y_median_X0_1000+r(p50)
 scalar  _Y_quartile1_X0_1000=_Y_quartile1_X0_1000+r(p25)
 scalar  _Y_quartile3_X0_1000=_Y_quartile3_X0_1000+r(p75)

 sum   E if X==1 , det
 scalar  _E_median_X1_1000=_E_median_X1_1000+r(p50)
 scalar  _E_quartile1_X1_1000=_E_quartile1_X1_1000+r(p25)
 scalar  _E_quartile3_X1_1000=_E_quartile3_X1_1000+r(p75)

 sum   Y if X==1 , det
 scalar  _Y_median_X1_1000=_Y_median_X1_1000+r(p50)
 scalar  _Y_quartile1_X1_1000=_Y_quartile1_X1_1000+r(p25)
 scalar  _Y_quartile3_X1_1000=_Y_quartile3_X1_1000+r(p75)

drop X products P int* E* Y
}

di Y_E_unadjusted_1000/n_iter
set obs 10000
matvsort V_Y_E_unadjusted_1000 V_Y_E_unadjusted_1000
di V_Y_E_unadjusted_1000[50, 1]
di V_Y_E_unadjusted_1000[950, 1]

di Y_E_products_P_1000/n_iter
set obs 10000
```

```
matvsort V_Y_E_products_P_1000 V_Y_E_products_P_1000
di V_Y_E_products_P_1000[50, 1]
di V_Y_E_products_P_1000[950, 1]

di Y_E_X_1000/n_iter
set obs 10000
matvsort V_Y_E_X_1000 V_Y_E_X_1000
di V_Y_E_X_1000[50, 1]
di V_Y_E_X_1000[950, 1]

di Y_E_products_1000/n_iter
set obs 10000
matvsort V_Y_E_products_1000 V_Y_E_products_1000
di V_Y_E_products_1000[50, 1]
di V_Y_E_products_1000[950, 1]

di Y_E_products_X_1000/n_iter
set obs 10000
matvsort V_Y_E_products_X_1000 V_Y_E_products_X_1000
di V_Y_E_products_X_1000[50,1]
di V_Y_E_products_X_1000[950,1]

di Y_E_X0_1000/n_iter
set obs 10000
matvsort V_Y_E_X0_1000 V_Y_E_X0_1000
di V_Y_E_X0_1000[50,1]
di V_Y_E_X0_1000[950,1]

di Y_E_X1_1000/n_iter
set obs 10000
matvsort V_Y_E_X1_1000 V_Y_E_X1_1000
di V_Y_E_X1_1000[50,1]
di V_Y_E_X1_1000[950,1]

di _E_median_overall_1000/n_iter
di _E_quartile1_overall_1000/n_iter
di _E_quartile3_overall_1000/n_iter

di _Y_median_overall_1000/n_iter
di _Y_quartile1_overall_1000/n_iter
di _Y_quartile3_overall_1000/n_iter

di _E_median_X0_1000/n_iter
di _E_quartile1_X0_1000/n_iter
di _E_quartile3_X0_1000/n_iter

di _Y_median_X0_1000/n_iter
di _Y_quartile1_X0_1000/n_iter
di _Y_quartile3_X0_1000/n_iter

di _E_median_X1_1000/n_iter
di _E_quartile1_X1_1000/n_iter
di _E_quartile3_X1_1000/n_iter

di _Y_median_X1_1000/n_iter
di _Y_quartile1_X1_1000/n_iter
di _Y_quartile3_X1_1000/n_iter
```