WEB MATERIAL

eAppendix 1: SAS code for simulation

```
/* Create datasets with variable # of groups & variable # of individuals in a
group */
%MACRO create_simulated_dataset(ngroups=, groupsize=);
data simulation_parms;
      retain condition;
      condition=0;
      do ngroups=&ngroups;
            do groupsize=&groupsize;
                  nobs=ngroups*groupsize;
                  condition=condition+1;
                  output;
            end;
      end;
run;
data simulated_chs;
      set simulation_parms;
      retain ngroups groupsize;
      keep condition serial group i  poverty  no_healthy_food_available
income rand male black age_yrs;
      /* First, generate values of individual-level poverty */
      do group=1 to ngroups;
            if ranuni(-1) < .25 then no_healthy_food_available = 1; else
no_healthy_food_available = 0;
            income_group_base = rand('normal',10.5581293,0.4020605);
            do i=1 to groupsize;
                  male = ranuni(-1) < .442;
                  black = ranuni(-1) < .512;
                  age_yrs = round(rand('uniform') * (65-18) + 18);
                  income_individual = rand('normal',0,.5);
                  log_income = income_group_base + income_individual;
                  income = exp(log_income);
                  /* 2000 Federal Poverty Level for 4-person family */
                  poverty = income < 17050;
                  output;
            end;
      end;
run;
proc freq data=simulated_chs noprint;
      table poverty*group/outpct out=simulated_group_pov;
run;
data group_pov;
      set simulated_group_pov;
      if poverty eq 1;
run;
proc univariate data=simulated_chs noprint;
      var income;
      output out=group_income mean=mean median=median;
      by group;
run;
data simulated_chs;
```

```
      merge simulated_chs group_pov(rename= (PCT_COL=group_pov))
group_income(rename= (mean=group_mean_income));
      by group;
      if group_pov = . then group_pov = 0;
      keep condition serial group group_pov i poverty bmi obese
no_healthy_food_available income group_mean_income rand male black age_yrs;
      label group_pov = 'group-level percent poverty'
                  group_mean_income = 'group-level mean income';
run;
data simulated_chs;
      set simulated_chs;
      /* Generate BMI similar to real data. */
      e=rand('normal',0,.70711);
      bmi = 16 + .92*male + 1.26* black + 0.069*age_yrs + .00005*(100000-
group_mean_income) + 10*e;
      /* Cut off the implausible low BMI tail -- this should not be many
individuals */
      if bmi < 15 then delete;
      if bmi < 30 then obese = 0; else obese = 1;
      rand = ranuni(-1);
      /* For this simulation, keep only 5% as the 'sampled' individuals */
      if rand > 0.05 then delete;
run;
%MEND;

/* Add noise to a measure */
%macro create_measurement_error(input=, output=, input_var=, output_var=,
percent_noise=);
data &output.;
      set &input.;
      &output_var. = &input_var. * (100+(&percent_noise.*(.5-ranuni(-
1))))/100;
run;
%mend;
/* Generate a result at a given level of noise */
%macro generate_result(in=, cur_noise=, input_var=);
      %create_measurement_error(input=&in., output=internal,
input_var=&input_var., output_var=as_measured, percent_noise=&cur_noise.);
      proc univariate data=internal noprint;
            var as_measured;
            output out=internal_with_group_error mean=mean_with_error
median=median;
            by group;
      run;

      proc sort data=internal_with_group_error; by group;
      proc sort data=internal; by group;
      data internal;
            merge internal internal_with_group_error(rename=
(mean_with_error=group_mean_with_error));
            by group;
      run;
      proc sql;
            select count(*) into :nobs from internal;
            select count(unique(group)) into :ngroups from internal;
      quit;
      proc mixed data=internal cl covtest noclprint;
```

```
            class group;
            model bmi= age_yrs male black income
group_mean_with_error/solution ddfm=bw;
            random intercept/subject=group;
            ods output SolutionF=solution;
      run;
      data solution1;
            set solution;
            noise = &cur_noise.;
            keep Estimate StdErr noise;
            where Effect eq 'group_mean_with_erro';
      run;

      data results;
            set results solution1;
      run;

      data results;
            set results;
            where noise ne .;
      run;
%mend;
/* Generate results for lots of levels of noise */
%macro generate_error_results(in=, input_var=, result=, start=, end=, by=,
hits_per=1);
data results;run;
%local cur_noise;
%let cur_noise = &start.;
%do %while(&cur_noise. <= &end.);
      %put Percent Noise is &cur_noise.;
      %do i=1 %to &hits_per;
            %generate_result(in=&in., cur_noise=&cur_noise.,
input_var=&input_var.);
      %end;
      %let cur_noise = %eval(&cur_noise. + &by.);
      DM output 'clear';
      DM log 'clear';
%end;
%mend;
/* Compare the results with noise to results without */
%MACRO process_results();
data no_error_results;
      set results;
      where noise eq 0;
      true_estimate = estimate;
      true_std_error = stderr;
      keep true_estimate true_std_error;
run;
proc sql;
      create table results_with_bias as
      select results.*, no_error_results.true_estimate as true_beta,
no_error_results.true_std_error as true_err
      from results, no_error_results;
run; quit;
data results_with_bias;
      set results_with_bias;
      percent_overestimation_estimate = 100*((estimate*1.0/true_beta) - 1);
```

```
	percent_overestimation_stderr = 100*((stderr*1.0/true_err) - 1);
run;
%MEND;

/* Make datasets of variable sizes and analyze them */
%macro vary_dataset_size(start_ngroups=, end_ngroups=, by_ngroups=,
start_groupsize=, end_groupsize=, by_groupsize=);
%local groupsize;
%let groupsize = &start_groupsize.;
%do %while(&groupsize <= &end_groupsize);
	%put groupsize is &groupsize.;
	%local ngroups;
	%let ngroups = &start_ngroups.;
	%do %while(&ngroups <= &end_ngroups);
		%put ngroups is &ngroups.;
		%create_simulated_dataset(ngroups=&ngroups.,
groupsize=&groupsize.);
		%generate_error_results(in=simulated_chs, input_var=income,
result='Estimate', start=0, end=20, by=20, hits_per=20);
		%process_results();
		data run_results;
			set results_with_bias;
			ngroups = &ngroups;
			groupsize = &groupsize;
		run;
		data varied_datasetsize_results;
			set varied_datasetsize_results run_results;
		run;
		%let ngroups = %eval(&ngroups. + &by_ngroups.);
	%end;
	%let groupsize = %eval(&groupsize. + &by_groupsize.);
%end;
%mend;

/* Okay, now do the simulation */
data varied_datasetsize_results;run;
%vary_dataset_size(start_ngroups=3, end_ngroups=100, by_ngroups=1,
start_groupsize=4000, end_groupsize=4000, by_groupsize=1);
title1;
axis1  label=(a=90 font="helvetica" height=24pt "Effect Percent
Overestimation" ) value=(height=2) order=(-3 to 3 by 1);
axis2  label=(font="helvetica" height=24pt "Number of Neighborhoods" )
value=(height=2);
symbol1 interpol=none value=plus;
proc gplot data=varied_datasetsize_results;
	plot percent_overestimation_estimate*ngroups/vaxis=axis1 haxis=axis2
overlay legend=legend1;
run;quit;
```

eAppendix 2: Mathematical basis for observed effect of misclassification in individual level data aggregated to create neighborhood level variables.

A simple fixed effect model with no cross-level interaction has the general form:

$$h\left(E\left[Y_{ij}|X_{ij},\bar{P}_i\right]\right) = \alpha + \beta X_{ij} + \gamma \bar{P}_i$$

Where the individual-level variable ($X_{ij}$) and group-level proportion ($\bar{P}_i$) both contribute to the dependent variable. Following Brenner, et al.[1], if $\bar{P}_i$ was aggregated from an individual-level measure with sensitivity $Se$ and a specificity $Sp$, then the observed value $\hat{P}_i$ relates to the true value as follows:

$$\hat{P}_i = \bar{P}_i * Se + (1 - \bar{P}_i) * (1 - Sp)$$

$$= \bar{P}_i * Se + 1 - Sp - \bar{P}_i + \bar{P}_i * Sp$$

$$= \bar{P}_i(Se + Sp - 1) + 1 - Sp$$

And therefore:

$$\bar{P}_i = \frac{\hat{P}_i + Sp - 1}{Se + Sp - 1}$$

Which can be substituted into the model as follows:

$$h\left(E\left[Y_{ij}|X_{ij},\hat{P}_i, Se, Sp\right]\right) = \alpha + \beta X_{ij} + \gamma\left(\frac{\hat{P}_i + Sp - 1}{Se + Sp - 1}\right)$$

$$= \alpha + \beta X_{ij} + \gamma\left(\frac{\hat{P}_i}{Se + Sp - 1}\right) + \gamma\left(\frac{Sp - 1}{Se + Sp - 1}\right)$$

$$= \alpha_{error} + \beta X_{ij} + \gamma_{error}\hat{P}_i$$

where $\alpha_{error} = \alpha + \gamma\left(\frac{Sp-1}{Se+Sp-1}\right)$ and $\gamma_{error} = \gamma\frac{1}{Se+Sp-1}$

And thus, similar to what has been shown for ecologic analysis[1], the slope of a fixed effect model regression line for a group level variable constructed by aggregating non-differentially misclassified individual-level values will be elevated by $\frac{1}{Se+Sp-1}$, the intercept will be biased only in the event of imperfect specificity, and estimates for other parameters in the model are unaffected. Furthermore, the extent of bias is independent of the base prevalence of poverty and distribution of the poverty data.

## References


1. Brenner H, Greenland S, Savitz DA. The effects of nondifferential confounder misclassification in ecologic studies. *Epidemiology* 1992;**3**(5):456-9.

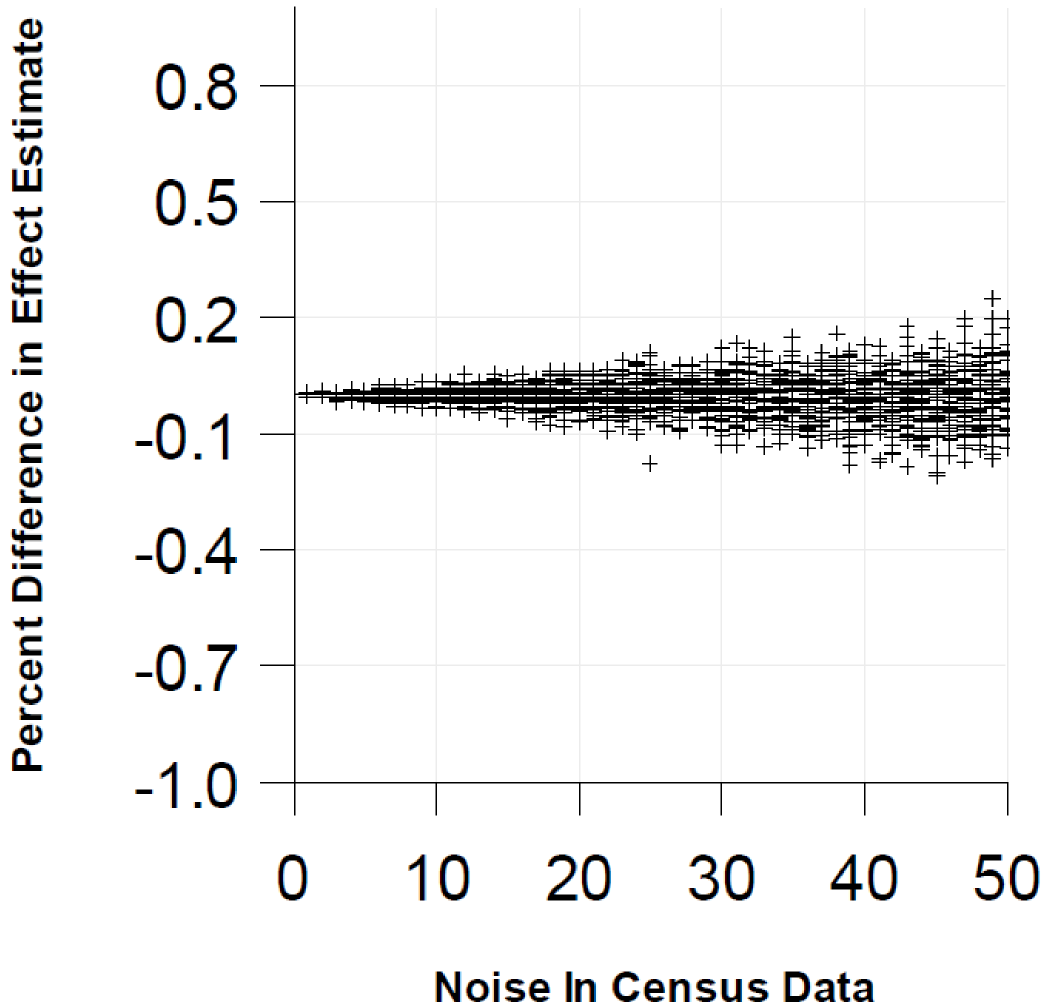
eAppendix 3: Mathematical basis for quantile-based comparisons' immunity from this bias

We return to the definition of misclassification:

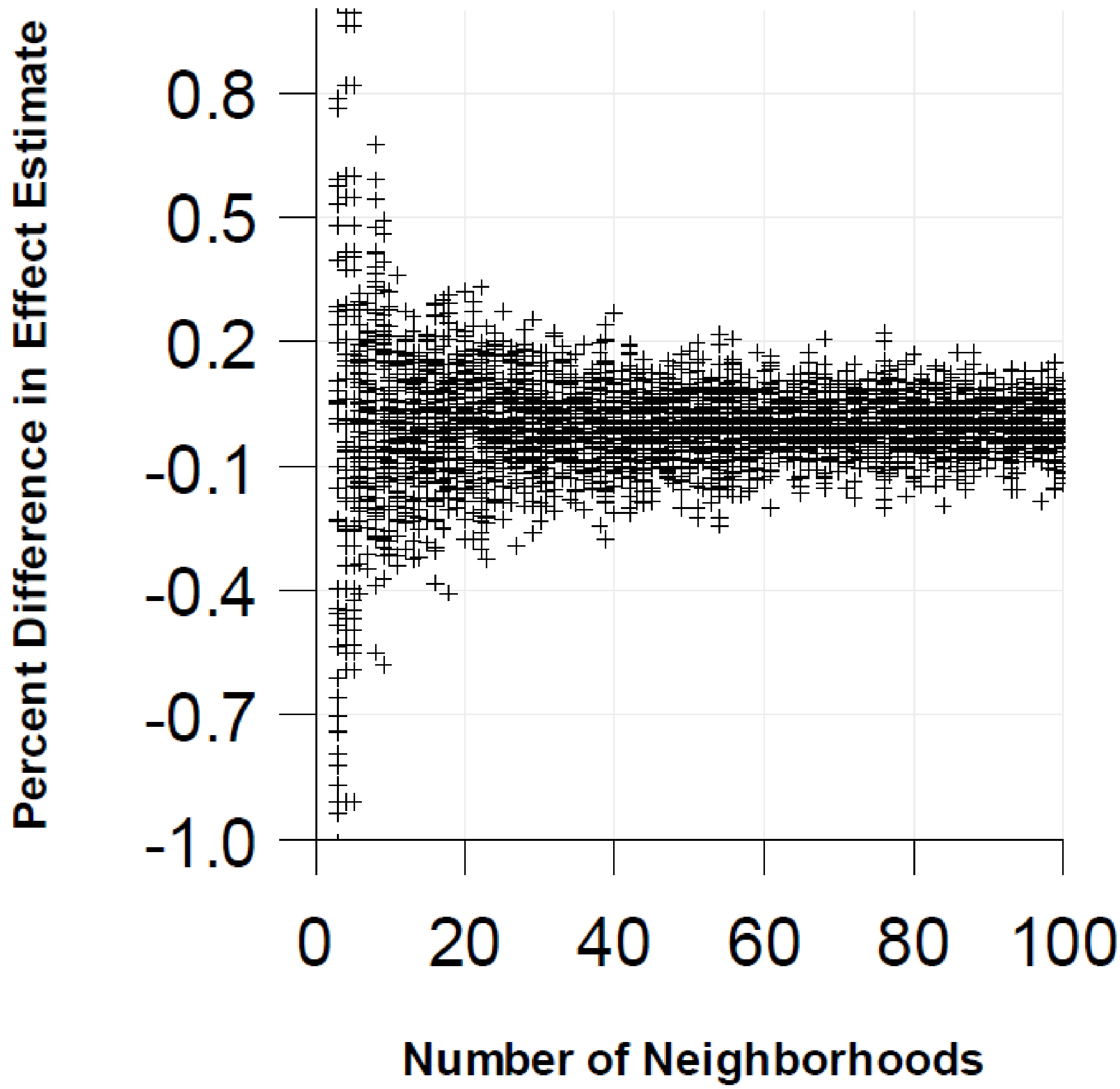
$$\hat{P}_i = \overline{P}_i * Se + (1 - \overline{P}_i) * (1 - Sp)$$

From this equation, we observe that non-differential misclassification does not affect the rank-order of $\hat{P}$ values; that is, if $\overline{P}_i > \overline{P}_j$, then $\hat{P}_i > \hat{P}_j$. As a result, quantiling data by $\hat{P}$ values results in the same categorization as quantiling by $\overline{P}$ would; thus, any non-differential misclassification in the underlying individual level data has no effect.

eFigure 1: Bias in effect estimates for the association between neighborhood mean income and Body Mass Index across repeated simulations, for increasing levels of measurement error in individual-level income data in population Census data.



eFigure 2: Bias in effect estimates for the association between neighborhood mean income and Body Mass Index across repeated simulations, by number of neighborhoods included in a study at a fixed level of measurement error in the individual level income data.



eFigure 3: Bias in effect estimates for the association between neighborhood mean income and Body Mass Index across repeated simulations, by the population size of the neighborhoods.