e Appendix 2

```
* eAppendix  5.9.11: written by W. Dana Flanders;
* Program to simulate Power and Area Under ROC for detecting model misspecification due to    ;
*   inappropriate omission of a confounder.                                      ;
* ----- Make up a hypothetical data set, to use for illustration                 -----;
* ----- to make the simulations realistic, recommend using real observations instead         ;
* ----- that is, use "ap2.all" in data step about 20 lines below, rather than generated data - ;
options nonotes nomprint;  *to suppress note, for simulations;
*options notes mprint;
data all;
seed=99;
do area= 1 to 100;
 do subject = 1 to 100;
     call rannor(seed, CO);
         CO= exp(CO);                    *Exposure = CO (carbon monoxide), here log-normal;
         call ranbin(seed, 1, .5, Cov1);
         call rannor(seed, IndExposure);
         call rannor(seed, err0);
         IndExposure = 0.3*CO + 0.4*Cov1 + 0.5*err0; *Indicator is associated with both Exposure and
Covariate;
     call rannor(seed, err);
         weight = 2000 - 0.5*CO + Cov1 + 10*err;    *Weight(Dependent Variable) depends only on CO
(exposure) and Cov1 (covariates);
         output;
 end;
end;
* --- Read in a data set with Actual observations: these data are analyszed to get
      parameter estimates, so the simulations will be realistic (use in place of hypothetical data) ------;
*libname ap2 'C:\Documents and Settings\wflande\My Documents';      * libname, if read in real data;
data all;  set all; *ap2.anal;   run;

* ------- Input the names of the Variables to be used in the analysis as macro Variables ----;
* ------- Reserved names: simWt, pred, err, Exposure, Indicator, seed, seeed, xx, scenari  ;
%let nsim = 25;                *Enter the number of simulations;
%let sed=37009;                 *Enter a starting seed;
%let Y = weight;               *Enter the name of the Dependent Variable here;
%let stdDev = 10;               *Enter the residual standard Error of the Dependent Variable here;
%let Exposure = CO;              *Enter the name of the Exposure Variable here;
%let Indicator = IndExposure;          *Enter the name of the (future) Variable to be use as the Indicator
of confounding;
* -------- List all the releveant Categorical Covariates;
%let CatCovariates= Cov1;
%let ConCovariates= ;             *List all the releveant Continuous Covariates;
%let CovarDrop= cov1;              *List the name of the Covariate to be dropped from the
MissSpecified Model;

* proc contents data=all; run;
* -------- Descriptive statistics for Analysis Variables;
proc corr data=all; var &exposure &indicator; run;
proc freq data=all;
 tables  &CatCovariates; run;
proc means data=all min max mean  p25 median p75 std; var weight &Exposure &CatCovariates; run;
```

```
* -------- Start simulations ----------------- ;
data all; set all;  * rename key variables for use in Simulations;
 Exposure = &Exposure;  Indicator = &Indicator;
 if Exposure ne . and Indicator ne . and &Y ne . ;
 keep Exposure Indicator  &Y &CatCovariates &ConCovariates;
* -------- Analyze the Real Observations to get parameter estimates so simulations are realistic;
proc genmod data=all ;
 class  &CatCovariates; *;
 model &Y = Exposure &CatCovariates &ConCovariates;
 output out=PredWsv p=pred;          *data set with the parameter estimates: get "true" predicted
values for simulations;
 title ' Fit the observed data to get parameter estimates';
 title2 '       (to make simulations realistic)       ';
run;   title; title2;

ods listing;
run;
* ------- Macro to carry out the simulations ---- ;
%macro SimFut(seed, scenario);
data save;  xx=.; output;            *create a data set to save the outputs;
data seed; seeed=&seed; output; run;    *create a data set to save the seed, from one simulation to the
next;

%do i = 1 %to &nsim;
%if &i > 1 %then %do; proc delete data=PredW; proc delete data=est1; proc delete data=est2;
run; %end;
* ------- Generate data with the simulated Dependent Variable;
data PredW seed; set seed (in=aa keep=seeed) PredWsv (in=bb) nobs=nn;; *generate Outcome counts
for analysis, based on Exposure;
 retain seed 0;
 if aa then seed=seeed;
 if bb then do;
  if pred ne . and seed ne . then do;
      call rannor(seed, err);          *Simulate Gaussian Random Error, with Std = &stdDev;
      simWt = pred + err*&stdDev;
  end;
  scenari = &scenario;
  if scenari =2 then do;                *Misspecify the model: in effect, drop &CovarDrop;
      &CovarDrop = 0;
  end;
  if simWt ne . then output PredW;
 end; *end if bb;
 if _n_=nn then do;
   seeed=seed;
   output seed;                      *Save the updated seed for next simulated data set;
 end;
run;
ods listing close;

run;
* ------- Fit model without Indicator, correctly specified (scenario=1) or
misspecified(&Scenario=2,covariate ommitted);
```

```
ods output ParameterEstimates=est1;
proc genmod data=PredW;
 class &Catcovariates;
 model simWt =  Exposure &ConCovariates &Catcovariates
        /dist=normal link=id;
 title ' Possibly Misspecified (Biased) model , without Indicator';
run; title;
**;

data est1; set est1;  * select parameter estimate for exposure (Exposure). This part is descriptive;
 if  parameter in ('Exposure');
 reject=0;  if ProbChiSq < 0.05 then reject=1;
 drop LowerWaldCL UpperWaldCL level1 Parameter ChiSq ProbChiSq; run;

* ------- Fit model with Indicator, correctly specified (scenario=1) or
misspecified(&Scenario=2,covariate ommitted);
ods output ParameterEstimates=est2;
proc genmod data=PredW;
  class  &Catcovariates;
  model simWt =  Exposure &ConCovariates &Catcovariates
          Indicator
        /dist=normal link=id; *assess;
 title ' Possibly Misspecified (Biased) model, With Indicator variable included';
run; title;

* ------- Calculate the Indicator from the output of fitted model, possibly misspecified ---;
data est2; set est2;
  if parameter = 'Indicator' then do;
     est_fut = abs(estimate/StdErr);   * est_fut is the Indicator Variable for detecting model
misspecification;
          est_fut2= estimate/StdErr;      * estimated slope for indicator and without taking absolute
value;
          reject2=0; if ProbChiSq < 0.05 then reject2=1; *used to reject the null of no model
misspecification;
          std_fut = StdErr;            * std_fut - the standard error of beta(future exposure);
     output;
  end;
  drop LowerWaldCL UpperWaldCL level1 StdErr estimate ChiSq ProbChiSq; run;

data est2; merge est1 est2 ;
  drop DF Parameter ;
run;
data save; set save est2; scenario=&scenario; run;  *store the results of each simulation in dataset Save;
%end;  * end do i;
run;

%mend;


%simFut(seed=&sed, scenario=1);          *scenario 1: no misspecification- use for specificity and Size
under the Null;
```

```
 data save11; set save; run;
 data final; set save; run;

 %simFut(seed=&sed, scenario=2);    *scenario 2: misspecify model, drop variable=CovarDrop - use for
 sensitivity and Power (non-Null);
 data save22; set save; drop xx; run;

 data final; set final save;
  label estimate = 'Beta (Exposure)';
  label StdErr = 'Std Err of Beta (Exposure)';
  label reject =     '% reject Ho:Beta (Exposure)=0';
  label est_fut ='Abs[Beta (Indicator)]/Std Err';
  label est_fut2='Beta (Indicator)';
  label std_fut='Std Err of Beta(Indicator)';
  label reject2 = '% reject Ho: no model misspec';
 run;

 ods listing;
 proc format;
  value scen 1 = 'no misspecification'
         2 = 'model misspecified ';

 proc means data=final mean p50;
  class scenario;
  var estimate StdErr Reject;
  format scenario scen.;
  title ' Descriptive Results Only: model does not include the Indicator';
   run; title;
 data final final2; set final;
  if scenario=1 then output final;   *output results separately for mis-specified and correctly specifed
 models;
  if scenario=2 then output final2; run;
 proc means data=final mean p50;
  class scenario;
  var est_fut std_fut reject2 est_fut2;
  format scenario scen.;
  title '  Simulated size under Null for Test of Model Misspecification';
  title2'  (Model Includes Exposure and Indicator,  No Dropped Variables)';
  title3 '  size (null) is given by mean of: reject2 '; run; title; title2; title3;
 proc means data=final2 mean p50;
  class scenario;
  var est_fut std_fut reject2 est_fut2;
  format scenario scen.;
  title '   Simulated Power (non-Null) to Detect Model Misspecification' ;
  title2'   (Model Includes Exposure and Indicator,  Dropped &CovarDrop)';
  title3 '  Power is given by mean of: reject2 '; run; title; title2; title3;

*---- get AUC by all pairwise comparisons percentage where  test (misspecified) > test(true), ties
ignored ;
proc iml;
 use  save11 var { Est_Fut };
 read all into SpecA;     * SpecA has Indicator for each simulated data set, correct model;
 use  save22 var { Est_Fut };
 read all into SensA2;     * SensA has Indicator for each simulated data set, Incorrect model;
 *print SpecA SensA2;
 gt2=0;               * count the number of times Indicator for incorrect model exceed that for correct;
 n=0;
 do i = 1 to nrow(SpecA);   * loop over all combinations of simulated Indicators;
  do j = 1 to nrow(SensA2);
    if (SensA2[i,1] > SpecA[j,1] & SpecA[j,1] ^= . ) then gt2 = gt2 + 1;
          if (SensA2[i,1] ^= . & SpecA[j,1] ^= . ) then n = n+1;
  end; *j;
 end; *i;
 AUC_A2=gt2/n;                   * AUC estimated as % of comparisons with Indicator for incorrect >
Indicator for correct model;
 varnam2 = {'AUC_A2'};*, 'AUC_A3', 'AUC_A4', 'AUC_A5','AUC_A6', 'AUC_A7', 'AUC_A8'};
 create AUC from AUC_A2 [colname=varnam2];
 append from AUC_A2;
quit;
proc print data=AUC; title " AUC by pairwise comparison"; run; title;
```