GFORMULA_RCT SAS MACRO version 1.1

Abstract
The GFORMULA_RCT macro implements the parametric g-formula in SAS to estimate the per-protocol effect in randomized trials with protocol deviations.

Authors
Roger W. Logan, Sara Lodi, Jessica G. Young, Miguel A. Hernán

Version
Version 1.1 June 2019. For questions and comments, email rwlogan@hsph.harvard.edu or slodi@bu.edu

Table of Contents

Overview 3
Outline of the algorithm 3
Natural course .... **Error! Bookmark not defined.**
Required structure of the input data set .... **Error! Bookmark not defined.**
Example of the g-formula macro call .... **Error! Bookmark not defined.**
Specifying parametric model assumptions 9
Specifying the functional form for the time index 10
Specifying the intervention .... **Error! Bookmark not defined.**
Frequently asked questions .... **Error! Bookmark not defined.**
Appendix 20
Structure of the GFORMULA macro .... **Error! Bookmark not defined.**
Examples of user-defined interventions .... **Error! Bookmark not defined.**
References 26

Overview

The GFORMULA_RCT macro implements the parametric g-formula (Robins 1986, Robins et al. 2004) to estimate the per-protocol effect in individually-randomized controlled trials (RCT) with two parallel arms.

Most RCT are subject to some degree of protocol deviation. Examples are treatment crossover, when patients switch to another trial arm treatment, and partial adherence to study medications. The intention to treat effect (ITT) is the effect of randomization assignment. The per-protocol effect is the effect of an intervention that would have been found if all participants had perfectly complied with the study protocol and nobody had been lost to follow-up. When protocol deviations are common, the ITT analyses may provide biased estimates of the effect of the intervention and should be complemented with per-protocol analyses [1, 4].

The parametric g-formula was developed in the context of observational studies to estimate counterfactual risk or mean of an outcome under hypothetical treatment strategies to adjust for time-varying confounders (Taubman et al. (2009), Young et al. (2011), Danaei et al. (2013), Lajous et al. (2013), Garcia-Aymerich et al (2014), Lodi et al (2015,2016,2018)). Like other g-methods, the parametric g-formula can also used to estimate the per-protocol effect in clinical trials (Lodi et al 2017, Lodi et al 2019) to adjust for baseline and time-varying predictors of protocol deviation.

When the focus is on estimating the per-protocol effect, each arm of the RCT should be analyzed and interpreted as a separate observational study because trials participants who choose not to comply with their assigned treatment may differ systematically from those who comply. The statistical analyses then should be aimed at adjusting for all pre and post-randomization determinants of protocol deviation and loss to follow-up and these determinants might be different in each trial arm. Therefore, in addition to the parameters of the GFORMULA macro, the GFORMULA_RCT macro has a set of parameters to allow great flexibility in specifying different parametric models in each arm of the trial.

Like the GFORMULA macro, the GFORMULA_RCT supports time to event, continuous and binary outcomes, binary, categorical and continuous treatments, and time-varying covariates while allowing for great flexibility on the model specification. The GFORMULA_RCT macro is an extension of the GFORMULA macro and basic understanding of the GFORMULA macro is required to correctly use the GFORMULA_RCT macro.

Outline of the algorithm

The parametric g-formula is a generalization of standardization for time-varying treatments and confounders28. When measured covariates are sufficient to adjust for confounding and selection bias (Robins 1986, Hernán and Robins 2016), the g-formula estimates the risk of the outcome that would have been observed if all participants in the trial had adhered to the intervention assigned at randomization and none had been lost to follow-up, under the assumptions of no residual confounding, no measurement error, and no model misspecification. The parametric g-formula is an approach to estimating the g-formula under parametric model assumptions. Technical details on the estimation procedure can be found in the GFORMULA macro documentation. Briefly, the procedure in the context of the per-protocol analysis of randomized trial has three steps:

- *Step 1: parametric estimation.* Separately for each trial arm, we specify regression models for each post-randomization time-varying covariates (including adherence is adherence adjustment is performed) and for the outcome variable at time t as a function of treatment and covariate history up to t and of baseline covariates.

- *Step 2: Monte Carlo simulations.* The above parametric models are used to simulate the post-randomization variables and outcome separately for each randomization arm under a user-specified definition of protocol compliance and no loss to follow-up.
- *Step 3: calculation of risk/mean under each intervention (standardization).* In the simulated datasets the average is mean/risk outcome at the end of the study is computed and compared across trial arms.

Definition of adherence to protocol

For each randomization arm the user will specify an intervention corresponding to the definition of protocol compliance. Examples of interventions are 'Adherence to medication >80%' or 'Compliance with the arm intervention' (see example). The definition of protocol compliance might differ by treatment arm (see example) or be the same for both randomization arms. Interventions are currently defined by setting certain macro parameter values prior to calling the GFORMULA_RTC macro using the following syntax:

```
 %let interv1 = intno=1, intlabel = , intcond = , nintvar= , … ;
```

See GFORMULA macro for details on types of interventions and how to define an intervention.

Outcome and censoring events

Suppose we have a clinical trial in which patients are randomized to treatment A or treatment B. The GFORMULA_RCT supports time to event, continuous, and binary outcomes. The type of outcome is specified in the macro parameter *outctype* in the macro call.

Time to event outcomes are specified as *outctype=binsurv*. For a time to event outcome the macro calculates the cumulative incidence for each randomization arm under the user-specified definition(s) of protocol compliance, and risk difference, risk ratio, number needed to treat, and (optionally) hazard ratios for treatment arm A versus B. Please note that calculation of hazard ratios increases substantially computational times.

Continuous outcomes are specified as outctype=*conteofu*. For continuous outcomes, the macro calculates the mean outcome at the end of the study period for each randomization arms and the mean difference under the user-specified definition(s) of protocol compliance for treatment A versus B.

Binary outcomes are specified as outctype= *bineofu*. For binary outcomes, the macro calculates the odds ratios of the outcome at the end of the study period for treatment A versus B.

Like the GFORMULA macro, the GFORMULA_RCT macro supports data with censoring events and, for time to event data, competing risks. When competing risks are present and the outcome is time-to-event, two choices are available: 1) the observed risk is computed by applying the complement of Kaplan-Meier to the observed data with competing risks treated as censoring events or 2) the observed risk is computed via the subdistribution cumulative incidence function. For more details on calculation and interpretation of competing risks with the g-formula see the GFORMULA documentation.

For all types of outcome, the macro provides estimates under the natural course, i.e. under no intervention on protocol compliance at any time but by eliminating of censoring due to loss to follow-up.

Required structure of the input data set

Consider a hypothetical clinical trial to evaluate the effect of daily exercise on the risk of diabetes. Diabetes-free participants were randomized to exercise at least 30 minutes daily or at least 45 minute daily. Primary outcome was time to diagnosis of diabetes. Patients attended up to 5 clinical visits during which they compiled a questionnaire about exercise and they were assessed for diabetes.

The GFORMULA_RCT macro requires a data structure in which each row contains the values for an individual during a time interval. These could be, days, months, years or study visit since baseline, usually randomization date. The example dataset *sample.sas* has the following data structure. The variable treat indicates the randomization group/arm. The variable *visit* indicates time since randomization. For each participant, uniquely identified by the variable *id*, the record *visit*=0 corresponds to baseline/pre-randomization assessment. The time variable is increments by one unit at each subsequent time intervals or visits. The time variable *visit* will than take values 0,1,2,etc until outcome, loss to follow-up or a competing risk occur. Participants will typically contribute different lengths of follow-up.

The variable *arm* encodes the randomization group and *agebase* is age at randomization. The dataset includes the following time varying covariates updated at each visit: *act* the number of minutes of exercise per day and *hbp* an indicator for hypertension.

The variables *diab*, *cens* and *dead* are indicators corresponding to outcome (diabetes diagnosis), loss to follow-up and competing risk, respectively between visit *k* and *k+1*. These indicators are recorded on the same line as interval k covariate measurements. For example, patient 3 was diagnosed with diabetes between visits 2 and 3 and their outcome indicator takes value 1 at visit 2. We assume that they occurred after interval k covariate measurement and prior to those that would be measured at time k+1. No records should be included for a subject following any of these indicators taking the value 1. When time intervals are of long duration, as it often happens in clinical trials, coding decisions to ensure covariates “precede” the outcome on a given line k may require assumptions.

Id	***visit***	***treat***	***agebase***	***diab***	***cens***	***dead***	***act***	***hbp***
1	0	1	35	0	0	0	32	1
1	1	1	35	0	0	0	22	1
1	2	1	35	0	0	0	0	1
1	3	1	35	0	0	0	30	0
1	4	1	35	.	1	.	0	1
2	0	2	79	0	0	0	10	0
2	1	2	79	.	0	1	0	0
3	0	2	50	0	0	0	10	0
3	1	2	50	1	.	.	0	0

Example of gformula_rct call

In both arms of our hypothetical trial, on average patients exercised less then expected under their assigned intervention. In addition, a number of patients were lost to follow-up before the end the study. We now illustrate how to use the GFORMULA_RTC macro to estimate the per-protocol effect of at least 30 versus at least 45 minutes of exercise under compliance to the assigned intervention and no loss to follow-up.

First, we use the *interva1* and *intervb1* macro to define what constitutes protocol compliance. These macros must appear before the *gformula_rtc* call. In our example, protocol compliance in arm 1 is defined as at least 30 minutes of exercise per day (*interva1*) and at least 45 minutes per day in arm 2 (*intervb1*). Specifying *inttype1=2* and *intmin1=3*0 allows us to estimate the risk of diabetes under a 'threshold intervention' on exercise such that all subjects in arm A "exercise at least 30 minutes per day". Many different types of interventions can be specified (more details in Appendix Table 1).

Then, we call the GFORMULA_RTC macro with input data set "sample", *id* the name of a variable in the data set "sample" containing the subject identifier "id" (here it happens to have the same name as the GFORMULA_RCT macro parameter but does not have to), *time* the name of a variable in the data set "sample" containing the follow-up time index "visit" and *timepoints=6* specifies the number of time intervals between time 0 and end of follow-up, including 0. The outcome *outc* is the time-varying failure binary indicator for diabetes "diab", thus we set *outctype=binsurv*. To guarantee that rerunning the code yields the same results we set the seed=9458. The call sets *comprisk=dead* and *censlost=cens* to identify variables in the data set *sample* containing time varying indicators of death (a competing risk in this case) and loss to follow-up, respectively. The call sets *crcensor=1* in order to estimate effects on the scale of the subdistribution cumulative incidence function (see GFORMULA macro documentation for more information on competing risks). We use non parametric bootstrap with 100 repetitions to compute confidence intervals (*nsample=100*). The randomization group is specified in *arm=treat*. Treatments arms are internally renamed as A and B, where A corresponds to the treatment arm with lower value of the variable *treat*, in this case at least 30 minutes of daily exercise. Hazard ratios are not estimated by default. If we wish to compute the hazard ratio of treatment A versus B with bootstrap confidence interval we need to specify the following additional parameters: *hazardratio=1,bootstrap_hazard=1,forrct=1*. We save the cumulative survival estimates at each 'visit' in the dataset *mysurv* in the work library.

Finally, we specify the set of time fixed and time-varying covariates for each arm. With the GFORMULA_RTC macro the set of time-varying covariates and model specification can vary by treatment arm. In this example, the time fixed and time-varying covariates are the same for the two randomization arms, but it does not have to. We adjust for potential baseline confounding by age (*fixedcova=baseage; fixedcovb=baseage*) and time-varying confounding by hypertension (*cova1=hbp; covb1=hbp*), a prognostic factor that potentially affects daily exercise. The treatment is time-varying physical activity (*cova2=act; covb2=act*). The parameters *ncova=2* and *ncovb=2 indicate that* in each treatment arm there are two time varying covariates, *hbp (a confounder)* and *act* (treatment variable).. The macro requires specifying the functional form of each time-varying covariate when included as an independent variable/predictor in the outcome model and the other time-varying covariates models (see below for more information on *ptype*). In this case, we set *cov1aptype=lag1bin* (function of value at previous time) and *cova2ptype = cumavg* (cumulative average of from time 0). The macro also requires specifying the functional form of each time-varying covariate when included as dependent variables/response (see below and Appendix Table 2 for more information *otype*). In this example, we set *cova1otype =2* (binary outcome analyzed with logistic regression) and *cova2otype = 4* (continuous outcome analyzed with linear regression). In addition to the per-protocol effect we also estimate the effect under the natural course (*runnca=1* and *runncb=1*).

```
** Read the gformula and gformula_rct macros;
 %include ‘mypath/gformula.sas’;
 %include ‘mypath/gformula_rct.sas’;
```

```
options mprint notes nomlogic nosymbolgen ;
/* Define intervention in arm A as indicated in the trial protocol*/
%let interva1 = intno=1 ,  nintvar=1,
    intlabel='All subjects exercise at least 30 minutes per day in all intervals',
    intvar1 = act, inttype1 = 2, intmin1=30, intpr1=1, inttimes1 = 0 1 2 3 4 5 ;

/* Define intervention in arm B as indicated in the trial protocol*/
%let intervb1 = intno=1 ,  nintvar=1,
    intlabel='All subjects exercise at least 45 minutes per day in all intervals',
    intvar1 = act, inttype1 = 2, intmin1=45, intpr1=1, inttimes1 = 0 1 2 3 4 5 ;

ods graphics off ;
options notes mprint mprintnest   ;
options nonotes nomprint ;


%let numint = ;
%gformula_rtc(

data= sample, /* declare input dataset */
id=id,            /* unique patient identifier */
time=visit,       /* time units or time intervals */
timepoints = 6, /* time interval of end of follow-up (risk/mean calculated at visit
5 */
outc=diab,              /* outcome variable */
outctype=binsurv, /* type of outcome */
comprisk =  dead  , /* death is a competing risk */
arm = treat ,     /* randomization group/arm */
timeptype= concat, /* time is modeled as a categorical variable */
timeknots = 1 2 3 4 5, /* knots/cutoff when timeptype=concat or conspl */

seed= 9458,

hazardratio = 1,
bootstrap_hazard = 1,
forrct = 1 ,
intcomp = 0 1 ,
nsimul = , /* use same number in each simulated data set for each arm */
nsamples = 100,
survdata = mysurv,/* save simulated survival probability in a dataset called mysurv
in work library */
intervname = myinterv,
resultsdata = myresults, /* save summary estimates of per protocol effects in a
dataset called myresults */

/* A refers to first arm  (the one with lower value of arm variable)*/
/* Model specification for the first arm */
ncova=2,
numinta=1 , /*number of interventions in arm A */
runnca = 1 ,/* output estimates under the natural course in arm A */
refinta = 1, /* reference intervention is 1 (in case versus 0,ie natural course*/
fixedcova = baseage,
cova1  = hbp,    cova1otype  = 2, cova1ptype = lag1bin,
cova2  = act,    cova2otype  = 4, cova2ptype = cumavg,

/* Model specification for the second arm */
ncovb = 2 ,
numintb = 1,
runncb = 1,
refintb = 1,
fixedcovb= baseage ,
covb1   = hbp,     covb1otype   = 2, covb1ptype = lag1bin,
covb2   = act,     covb2otype   = 4, covb2ptype = cumavg

);

ods graphics on ;
```

Selected output from the above call is provided here:

PREDICTED RISK UNDER SEVERAL INTERVENTIONS UNDER TWO ARMS : A (treat = 0) and

Interv. Interventions under A

0 Natural course

1 All subjects exercise at least 30 minutes per day in all intervals

Interventions under B

Natural course

```
  All subjects exercise at least 45 minutes per day in all intervals
 PREDICTED RISK UNDER SEVERAL INTERVENTIONS UNDER TWO ARMS :  A (treat = 0)   and

               Observed risk= 5.22 for arm A and 5.36 for arm B.
                                  Data= sample
                        Number of bootstrap samples= 100
                            Reference is using arm A

                                                       Lower     Upper
                     Risk (%)    Risk (%)     Risk     limit     limit
          Interv.     under A     under B     ratio    95% CI    95% CI

             0         5.51        5.40        0.98     0.31      1.32

             1         5.84        5.52        0.95     0.29      1.35


                                          Lower  Upper            Lower  Upper
            Risk (%) Risk (%)     Risk     limit  limit # Needed  limit  limit
    Interv.  under A  under B difference 95% CI 95% CI to Treat 95% CI 95% CI

       0      5.51     5.40      -0.11   -11.67  1.47    -873    -246    771

       1      5.84     5.52      -0.32   -12.53  1.55    -315    -215    219

  Lower     Upper

                                Hazard     limit     limit

                       int        Ratio     95% CI    95% CI

                        0         0.980     0.31      1.37
```

```
                    1         1.000      0.28       1.52
```

The output shows that the observed cumulative risk of diabetes is 5.22% in arm A (at least 30 minutes of exercise per day) and 5.36% in arm 2 (at least 45 minutes of exercise per day). The difference and/or ratio of these risks are estimates of the intention to treat effect. Because we selected *crcensor*=1, is calculated non parametrically by computing the subdistribution cumulative incidence function in the observed data.

The estimated cumulative risk, risk difference, risk ratios and number needed to treat assuming compliance to the exercise schedule assigned at randomization and no loss to follow-up can be found in the rows where interv = 1. For instance, the risk of diabetes at the end of the study is 5.84% and 5.52% in arm A and B, respectively and the risk difference is 0.32% (95% CI -12.53,1.55) using arm A as reference group. The hazard ratio of arm B versus A is 0.98 (0.31,1.32). These measures of absolute and relative risks estimate the per protocol effect in the trial under the untestable assumptions of no residual confounding and correct model specification.

The estimated risks, risk ratios, risk difference and number needed to treat under the natural course be found in the rows where interv = 0. These can be interpreted as the effect of randomization arm (regardless of compliance to the exercise schedule) and no loss to follow-up. The estimated risk ratio under the natural course, using arm A as reference groups, is 0.98 with a 95% confidence interval based on 100 bootstrap samples of (0.31, 1.32).

Specifying parametric model assumptions for the time-varying covariates

For each time-varying covariate *covX* (*cov1,…covp*) the user should specify a covXotype and covXptype:

COVXOTYPE

The macro parameter *covXotype* is used to select the SAS regression fitting procedure for the density of each time-varying covariate *covX (cov1,…covp)*. The selection of *covXotype* also determines how the value of the time-varying covariate is simulated at each time *k*. Options available for *covXotype* are summarized in Appendix Table 2.

COVXPTYPE

The macro parameters *covXptype* (*cov1ptype,…,covpptype*) are jointly used to create and select the functional form of the "past covariate history" included as independent variables in each regression model fit. The program contains options to incorporate different function of this history through appropriate selection of these parameters. Lagged values of *covX* must be included in the input data set data with names covX_l1, covX_l2, covX_l3, for certain choices of *covXptype*. Options available for *covXptype* are summarized in Appendix Table 3 2).For any given choices of *covXptype*, X=1,…,p, independent variables included in any given regression model will depend on the "order" of *covX* relative to the dependent variable of that model. For this reason, Appendix Table 3, in describing how different choices of *covXptype* impact how the "history" of *covX* appears in a given regression model, we will distinguish between:

- a model for the density of covY, where Y<X ; heretofore "preceding covariate model"
- a model for the density of covY, where Y>X ; heretofore "succeeding covariate model"

- any "outcome regression model"; either a pooled over time regression model for the discrete hazard of the event or competing risk (*outctype=binsurv*) or a regression model for the mean outcome using only records with time=timepoints-1 (*outcype=conteofu or bineofu*).

Note: The log file will automatically display both externally and internally created variables included as independent variables in each regression fit so the user can see the implications of selecting different values of *covXptype*.

More details and examples on the choice and specification of covXptype can be found in the GFORMULA documentation.

Specifying the functional form for the time index

All regression models are pooled over time (except for the regression models for the outcome when *outctype=conteofu* or *bineofu*). The macro parameter *timeptype* determines the function of time included as independent variables in each model. Choices include *conbin, concat, conqdc, concub, and conspl*. The function is determined by the choice of suffix. These are as described as in Appendix Table 2 for *covXptype*. For suffix -cat and -*spl*, the user must also define *timeknots* (the chosen category cutoffs/knots, separated by spaces).

Description of the GFORMULA_RCT macro parameters

Parameters to specify the interventions defining protocol compliance to be set before the GFORMULA_RCT call:

```
%let interv1 = intno=1, intlabel = , intcond = , nintvar= , … ;
```

Parameters are as follows:

intno (required)

Intervention number. This should be indexed 1 to numint.

intlabel (required)

Description of the intervention to be output with the results.

intcond (optional)

Condition under with the intervention is implemented (if not always implemented).

nintvar (required)

Number of intervened on variables. A maximum value of 8 is allowed.

For #=1,…nintvar

intvar# (required)

Variable undergoing intervention.

inttype# (required for all intervention variables)

Type of intervention for intvar#. There are 6 possible types, summarized in Table 1.

inttimes# (optional, default = -1)

Times at which intvar# will be intervened on. The default is -1, which will lead to no intervention.

intpr# (optional, default=1)

Probability that the intervention occurs. The default is 1, which will lead to always intervening when other conditions are met.

intmin# (required for inttype 2)
intmax# (required for inttype 2)

For inttype#= 2, the user assigns the threshold value to intmin# if the goal is to maintain treatment values under intervention above the threshold. The intmax# is parallel to intmin# but sets a maximum rather than a minimum. Both intmin# and intmax# may be specified to maintain treatment values under intervention within some range (e.g. eat at least 1 serving of fish but no more than 3 servings of fish per week)

intchg# (required for inttype 3)

Fixed amount that is added to intvar#.

intvalue# (required for inttype 1 interventions)

Fixed target for the values of intvar#. There are several pre-defined interventions supported by the algorithm which are chosen using the inttype# macro parameter for # possibly ranging from 1,…4. Here we describe some choices of inttype# that have been applied in the literature with references. Other choices are more briefly summarized in Table 1.

Parameters of the GFORMULA_RCT macro:

Parameters common to the GFORMULA macro:

`data` (required)

Input dataset to be used for the analysis

`id` (required)

Unique identifier for subjects in the dataset *data*

`time` (required)

Time index in the dataset *data*. It must begin at 0 (the interval that subject enters the study or “baseline”) for each subject (indexed by *id*, see above) and increment by 1 for each subsequent interval. The largest possible value of *time* for any subject must be *timepoints*-1.

`timeptype` (required)

Function of time/interval to be included in pooled over time models. Choices for *timeptype* are described in the GFORMULA documentation

`timeknots` (optional)

Knots or category cutoffs when *timeptype* is selected as *conspl* or *concat*.

`timepoints` (required)

The parameter specifies the desired end of follow-up interval. For example, if *time* indexes month of follow-up and the 60- month risk/mean is of interest then the user should set *timepoints*=60. The largest possible value of *time* allowable for any subject must be *timepoints*-1 as the *time* index must begin at 0.

`timefuncgen` (optional)

User-defined macro that provides general functions of time (other than splines and categories) which can be used in the covariate and outcome models. It should be used together with *covXaddvars* and *covXsaddvrs* when the time function is included in the model for covX, or with *eventaddvars* and *eventsaddvars* when the new time function is included in the model for the outcome. See GFORMULA macro documentation for examples.

`interval` (optional, default = 1)

Length of time between one unit increments in *time* (assumed constant). The value of this parameter is only used if any *covXptype* is chosen as a skp- type. See GFORMULA macro documentation for examples on skp ptypes.

`outc` (required)

Outcome variable in the dataset *data*.

`outctype` (optional, default=binsurv)

Type of outcome: *binsurv* (time-varying failure indicator; default), *bineofu* (binary outcome at end of follow-up) and *conteofu* (continuous outcome at end of follow-up).

`outcinteract` (optional)

Product ("interaction") terms to be included as independent variables in the outcome regression model. These can include interac7tions between any baseline variable included in fixedcov, any time-varying covariate covX or time (note, interactions with time for the outcome model are only relevant for outctype=binsurv). The structure of the interaction is a product term bN*bM or bN*M or N*M, where the b indicates that interaction is with a baseline variable and the N or M alone indicates that the interaction is with a covX or time. The N(M) in bN(bM) is the numeric location of the baseline covariate included in *fixedcov*. The N(M) in N(M) alone is the index X for the time-varying covariate covX or 0 for time. See FAQ 2 in the GFORMULA documentation for more details.

`censlost` (optional)

Variable indicating censoring due to any event other than a competing risk event in the dataset *data*. For input data sets with no loss to follow-up the parameters *censlost* may be left blank.

`comprisk` (optional)

Variable indicating a competing risk event in the dataset *data*. For input data sets with no competing risks, the parameter *comprisk* may be left blank. Note that an indicator of loss to follow-up and/or competing risk may be assigned to the parameters *censlost* and *comprisk*, respectively, for outcome types *conteofu* or *bineofu*. However, both causes of failure to be present in the study by end of follow-up will be treated identically in these cases (i.e. as forms of censoring) and any value assigned by the user to crcensor is ignored; in this case the default *crcensor*=0 is always used.

`crcensor` (optional, default = 0)

When *outctype*=binsurv, *crcensor*=0 simulates an intervention that eliminates competing risk events. If *crcensor*=0 is used, risks are computed on the scale of the subdistribution cumulative incidence function and risk estimates will be therefore a function of the discrete hazard for competing risk conditional on past treatment and covariates.

`compriskinteract` (optional)

Product ("interaction') terms to be included in the above hazard model if *crcensor* is set to 1. See *outcinteract* for the required syntax.

`eventaddvars` (optional)
List of variables (should be baseline variables not included in all of the modeled covariate models) to add to the outcome model. For more information, see *covXaddvars*.

`compriskaddvars` (optional)
Additional variables and user defined macros for adding extra variables to the *comprisk* models that were not included in other lists. For more information see *covXaddvars*.

`usesplines` (optional, default = 1)
Indicator of whether or not to use splines in the predictors of a variable when the ptype is one of tsswtich1, lag1cumavg, or lag2cumavg. For more details see Appendix Table 2 and examples in the GORMULA macro documentation.

`keepsimuldata` (optional)
List of variables not created by any specified *covXptype* or *covXotype* that will be needed in the generation of the simulated data set. For more details see examples in the GORMULA macro documentation.

`equalitiessimuldata` (optional)
User defined macro that equates pre baseline simulated variables to observed variables.

`intervname` (optional)
Specifies the name of a data set to hold the results from the INTERV macro for each intervention when running the bootstraps in parts. This must be set to a permanent data set so that the results can be put back together for each part. If *sample_start* is set to 0 and *sample_end* is set to a value other than *nsamples* then *intervname* must be defined.

`resultsdata` (optional)
Specifies a dataset to store the results.

`simuldata` (optional)
Specifies a dataset to store the simulated dataset for each arm under the natural course.

`survdata` (optional, default = work._survdata_all)
Specifies the dataset to hold the estimated cumulative survival probabilities at each time point under each interventions for each arm. When not running the bootstraps in parts, this should include the libname for where to keep the data. The default will be the work space. When running the bootstraps in parts with *outctype* = binsurv this variable needs to be defined to calculate the restricted mean survival time. See FAQ 5 in the GOFRMULA documentation for more details of how to run the bootstraps in parts.

`check_cov_models` (optional, default = 0)
Specifies if the program will save the means of the observed and generated/simulated covariates under the natural course.

`covmeandata` (optional, default=work._covmean_all)

Specifies the name of the dataset to hold the differences between the observed and simulated means of each covariate if *check_cov_models=1*.

`print_cov_means` (optional, default = 0)
Specifies if the program should print out the comparison of the observed and generated variables if *check_cov_models=1*.

`save_raw_covmean` (optional, default = 0, default = _covmean_all_raw)
Specifies if the program should save the results of the comparisons for each sample and each time point if *check_cov_models=1*.

`observed_surv` (optional)
Dataset to hold the observed survival probabilities at each time point under the natural course. When not running the bootstraps in parts, this should include the libname for where to keep the data. The default will be the work space. See FAQ 5 in the GFORMULA documentation for more details of how to run the bootstraps in parts.

`print_stats` (optional, default = 1)
Indicates whether to print basic proc means statistics for the generated variables for each *timepoint* in the base sample.

`outputs` (optional, default = yes)
Option to suppress the model output for the base sample. Model results for all bootstrap samples are suppressed by default.

`seed` (optional, default = 7834)
Random numbers seed.

`nsamples` (optional, default = 50)
Specifies the number of bootstrap samples for construction of confidence intervals and estimated standard errors.

`sample_start` (optional, default = 0)
Specifies the starting sample sample when running bootstraps in parts. See FAQ 5 in the GOFRMULA macro documentation for more details of how to run the bootstraps in parts.

`sample_end` (optional, default = -1)
Specifies the starting sample and the ending sample when running bootstraps. When *sample_end* is set to be -1, it will be replaced with the value of nsamples. See FAQ 5 in the GOFRMULA macro documentation for more details of how to run the bootstraps in parts.

`nparam` (optional, default = sample size)
Specifies the sample size drawn for estimating model parameters in each bootstrap sample (less than or equal to the sample size of the dataset *data*).

`nsimul` (optional, default = sample size)

Specifies the sample size of the Monte-Carlo simulation.

`hazardratio` (optional, default = 0)

When *outctype* = binsurv, fits a Cox model and estimates the hazard ratio of the outcome for two interventions listed in *intcomp*.

`intcomp` (optional)

When *hazardratio*=1, *intcomp* is an ordered list of two interventions, int1 int2, where the reference level will be taken to be int1. The natural course can be used by setting int1 = 0.

`bootstrap_hazard` (optional, default = 0)

When running bootstraps and *hazardratio* = 1, the program will also calculate the bootstrap confidence interval for the hazard ratio. (Warning: with large data this will increase the time required to run the program.)

`hazardname` (optional)

When running the program in parts and *bootstrap_hazard* = 1, *hazardname* is the name of data set to hold the temporary hazard ratio for each iteration of the program. This needs to be the same name in each call to the macro. The data sets will be stored in the directory indicated in *savelib*.

`savelib` (optional, default = work)

Library to hold the intermediate results and data sets that are to be saved by the macro. Currently the *savelib* variable is only used when running bootstraps in parts (chunks), or when running the macro with the testing option.

`rungraphs` (optional, default = 0)

Indicates whether the macro should also generate graphs for comparing the simulated (generated) data under the natural course and the observed data. Running the macro with *rungraphs* = 1 will also cause the GFORMULA macro to save more intermediate results. If the user does not set the variables *check_cov_models, covmeandata, observed_surv,* and *survdata* then the macro will set them as follows : check_cov_models = 1, covmeandata=_covmean_ , observed_surv=_obssurv_, and survdata=_survdata_. To redo the graphs later the parameters s*urvdata, observed_surv and covmeandata* need to be set to permanent data sets that include the libname in the name. Note, when setting *rungraphs* = 1 the macro will save more data than during a typical run.

`title1,title2, title3` (optional, default=)

Used in generating the three-line title on the first page of the graphs obtained when setting *rungraphs* = 1.

`weight` (optional, default =)

A weight variable to be included in all of the covariate models. The value of the variable should be non-negative and can have non-integer values. See the SAS documentation on the use of this variable. This variable will be used only in the estimation of the regression models but will not be used in the generation of the simulated data sets.

`extcovlimit` (optional, default = 0)

For continuous variables with otype 3,4, 6 or 7 allows for the simulated value to be outside the observed range by a specified percent. The possible values are = 1 for 10% and =2 for 20% above or below the observed limits.

`printlogstats` (optional, default = 1)
Indicator of whether or not to include diagnostic output in the log file, including seeds , number intervened on for each intervention and bootstrap, number of times variables generated outside bounds, etc. This may be useful when calling the gformula macro from another macro.

`minimalistic` (optional = no)
Indicator of whether or not to reduce the number of variables that are saved when creating the data sets under each intervention. When set to yes, this will set the following macro variables: rungraphs = 0 , print_cov_means = 0, and check_cov_models = 0. In addition, this will reduce the number of variables that are saved when running the Monte-Carlo simulation part of the macro. Only those variables that are needed for the final table will be saved. Generally, these are variables based on the outcome.

`testing` (optional, default= no)
Indicates whether or not to run the macro in a testing mode. When equal to yes, only one sample will be run. In this case the macro will set *nsamples* = 0 and *mimimalistic* = no. When more than the natural course is run each simulated data set based on the intervention will be saved with all the desired variables for each observation. The results for each intervention will be saved in a data set named simuldata&intcount, were intcount ranges from 0 to &numint. These data sets will be saved in the library listed in the savelib variable. This will essentially double the time needed to run the macro with a single sample. *Simuldata* must be set to a name for this option to work.

Parameters specific to the GFORMULA_RCT macro:
`fixedcova ; fixedcovb` (optional)
List of time-fixed covariates at baseline (i.e. any variables defined in the input data set *data* which have constant values over time within levels of *id*) to control confounding/selection bias. Variables should be separated by spaces. For example, to define *fixedcova* to include sex, race and birthplace, define *fixedcova=sex race birthplace*. The user may also use *fixedcova (fixedcovb)* as a means of including the baseline values of any time-varying covariates *covX* in all functions of time-varying covariate history; thus allowing for a slightly more flexible function of covariate history than that created using certain choices of *covXptype*. For example, say we define *covX*=CD4 for some X=1,…,p. Defining *fixedcov*=sex race CD4 would allow the outcome model and all time k covariate models to depend on the baseline value of CD4 as well as the pre-baseline subject characteristics sex and race; this would be in addition to dependence on the post-baseline function of CD4 determined by the choice of *covXptype* in this example.

`ncova; ncovb` (required)
Number of time-varying covariates in the analysis of each treatment arm.The maximum allowable value is 30.

`numinta; numintb` (optional, default = 0)
Number of interventions. The default is 0 which produces results only for the natural course.

`refinta; refintb` (optional, default = 0)
Intervention to be used as the reference level when calculating risk ratios and risk differences. The default is the natural course (*refinta (refintb)*=0).

`runnca; runncb` (optional, default = 1)
Indicates whether the macro should run the natural course intervention. When *runnca* = 0 and *numinta* = 0, the

GFORMULA_RCT macro will only calculate the model parameter estimates. This might be useful for running basic diagnostics of the covariate models.

usebetadataa; usebetadatab (optional default = 0)
Specifies whether regression procedures should be fit (*usebetadata*=0) or a saved data set with the estimated coefficients from a previous run stored in the data set *betadata* should be used (*usebetadata*=1).

betadata (optional)
Specifies
a dataset to store the coefficient estimates. If *usebetadata*=1 then *betadata* must be defined as a permanent data set created on a previous run.

titledataa; titledatab (optional, default=)
Data set that contains the label for each covariate comparison that will be generated when setting *rungraphs* = 1. The data set should have one observation with a character string for each modeled variable (other than the *time* variable) which has the value of what the desired label should be. The macro will generate a default label of "Mean covX" (where *covX* is the variable listed in the *covX* macro variable.)

tsizea; tsizeb (optional, default = 1)
Relative size of the titles used in generating the graphs when setting rungraphs = 1. Depending on the length of the titles that are being used, this value can be modified to alter the size of the titles so that they will fit on the page. This should typically not be larger than around 1.5.

graphfilea; graphfileb (optional, default=gfilea.pdf and graphfileb)
Files which will hold the generated graphs when running the macro with *rungraphs*=1.

For all time varying covariates X for each treatment arm analysis:

covXa; covXb (required for X=1)
Specifies a variable corresponding to a time varying covariate value for each person-interval in the dataset *data*.
covaXotype; covbXotype (required for X=1 and, when covX not empty, X>1)
Specifies the regression procedure where *covaX* (or *covbX*) is the dependent variable and related specifications for the simulation in step 2 of the algorithm. Possible types are described in Appendix Table 2.

covaXptype; covbXptype (required for X=1 and, when covX not empty, X>1)
Specifies how the history of *covaX (covbX)* will be included in all regression models. Possible types are described in Appendix Table 3.

covaXmtype; covbXmtype (optional, default = all, can be all, some , or nocheck)
checkaddvars (optional, default = 1)
CovXmtype along with *checkaddvars* specifies how the variable covX appears in the models for covY. See GFORMULA macro documentation for more details and examples.

covaXcumint; covbXcumint (optional for covaXpytpe= *lag1cumavg* and *lag2cumavg*)
Specifies a variable to indicate which observations to include in the average terms when the ptype is lag1cumavg or lag2cumavg. See Appendix Table 3 and GFORMULA macro documentation for more details.

`covaXknots; covbXknots;` (required for –cat and -spl type variables

Knots or category cutoffs when *covaXptype (covbXptype)* is a -cat or -spl type. See Appendix Table 3 for more details.

`covaXskip; covbXskip` (required for skp- type variables)

Values of *time* for which *covaX* w (*covbX*)as not measured for any subject. For example if *covX* was not measured for any subject at *time*=0, *time*=2 and *time*=5, *covaX (covbX)* was carried forward from actual measurement times at those times and we selected *covaXptype (covbXptype)* as a skp- type then we would define *covaXskip*= 0 2 5. See Appendix Table 3 for more details and GFORMULA macro documentation for an example.

`covaXinteract; covbXinteract` (optional)

Product terms (interactions) to include in the covariate model. Syntax is as in *outcinteract*.

`covaXclass; covbXclass` (optional for covaXotype (covbXotype)=4)

Specifies the variable in the data that contains the lagged value of *covaX (covbX)* that will be used to split the model for *z_var* into two parts based on covaXclass = 0 and covaXclass ne 0.

`covaXwherem; covbXwherem` (optional)

Specifies a condition under which to fit any regression procedures specified by the choice of *covaXotype (covbXotype)*. See FAQ 3 in the GFORMULA macro documentation for examples.

`covaXwherenosim; covbXwherenosim` (optional default = (1=0))

Indicates a condition under which not to simulate a value under an intervention but to assign a fixed value. See FAQ 3 in the GFORMULA macro documentation for examples.

`covaXnosimelse; covbXnosimelse` (optional)

User defined macro used to assign a value to *covaX (covbX)* when the *covaXwherenosim (covbXwherenosim)* condition holds. See FAQ 3 in the GFORMULA macro documentation for examples.

`covaXrandomvisitp; covbXrandomvisitp` (optional)

Time-varying variable corresponding to an indicator that covaX (covbX) is measured in interval k. On lines where this variable is set to 0, *covaX (covbX)* should be carried forward from the last time in the input data set *data*. When this option is used, this time-varying indicator of “visit process” for *covaX (covbX)* is assumed an additional time-varying confounder and covaX (covbX) corresponds to a “last measured value” (Hernan et al., 2008). Defining this option will also change the way covX is modeled and simulated to minimize reliance on parametric model assumptions where deterministic knowledge of the data structure can be used (Young et al. , 2011). For more details see FAQ 7 in the GFORMULA macro documentation.

`covaXvisitpmaxgap; covbXvisitpmaxgap` (optional, default = 9e10)

Fixed value for the maximum number of periods allowed between visits before a subject will be censored (censlost=1). This parameter is used as a condition under which to model *covaXrandomvisitp (covbXrandomvisitp)* process and is also used in the simulation of the random visit process by imposing the condition that if the time-since-last-visit lagged by one period is equal to *covXvisitpmaxgap*, then the simulation will force the simulated value of *covXrandomvisitp* to be 1. If there is no limit (i.e. subjects will not be censored for too many missed measurements), then this must be set to a large number (larger than *timepoints*). Additional details are given in FAQ 7 in the GFORMULA macro documentation.

`covaXvisitpwherem; covbXvisitpwherem` (optional, default = (1=1))
Condition under which to model *covaXrandomvisitp (covbXrandomvisitp)*, and is specified using standard SAS syntax, referring to the variables in the original dataset by their parameter names (e.g. &cova1). See FAQ 7 in the GFORMULA macro documentation for examples.

`covaXvisitpcount; covbXvisitpcount` (optional, required when using a visit process)
Fixed value to assign to the lagged time-since-last-visit variable at baseline. Used to initialize the time-since-last-covaX counter at baseline. See FAQ 7 in the GFORMULA macro documentation for examples.

`covaXaddvars; covbXaddvars` (optional)
List of any additional variables to include as predictors of *covaX* (*covbX*) which are not included in *fixedcova* (*fixedcovb*) or as a time-varying covariate. The variables should appear in the dataset and could be functions of time (see timefuncgen), a time fixed or time-varying variable. This option should be used in combination with covaXsaddvars (covbXsaddvars). See FAQ 4 in the GFORMULA macro documentation for an example.

`covaXgenmacro; covbXgenmacro;` (optional)
User defined macro used for creating additional variables used the covariate and outcome models listed in covXaddvars or eventaddvars, etc. The code should be the same as how the variables were created prior to calling the GFORMULA_RCT macro. See FAQ 4 in the GFORMULA macro documentation for an example.

Appendix

Table 1: Summary of intervention types

inttype	**Description**	**Action**	**Required additional parameters**
1	Static deterministic intervention	Sets to fixed value intvalue#	intvalue#
2	Threshold intervention as considered by Taubman et al. (2009), Young et al. (2014).	Assigns intmin#/intmax# if natural value is below intmin#/above intmax#, otherwise sets to the natural value	intmin#/intmax#
3	Fixed change	Adds a fixed amount intchg# to the natural value	intchg#
4	Previous value	Carries forward value from previous time	
-1	User defined intervention. The user must provide a user-defined macro with code to implement the intervention.	Examples of user-defined interventions are provided in the supporting documentation and in the appendix of the GFORMULA macro documentation	User-defined macro intusermacro#. The macro should be placed above the call to %gformula. Can also use

Table 2: Summary of SAS regression procedures and simulation rules by choice of covXotype

covXotype	covX variable type	SAS procedure(s)	Notes on model fit	Rules for simulation
1	Binary variable	PROC LOGISTIC	Fits model to all records	CovX is generated based on estimated model parameters
2	Binary variable	PROC LOGISTIC	Fits model only to records where the first lagged value of covX=0. Should be used for binary covariates that, once they switch from 0 to 1, they stay 1 (e.g. indicator of first diabetes diagnosis *by* time k)	CovX is generated based on the estimated model parameters until the first 1 is generated. After that the value of covX is always set to 1.
3	Continuous variable	PROC REG	Fits model to all records	CovX is generated according to a normal density with mean estimated by the estimated regression model coefficients and variance estimated by the sample variance of covX . Generated values greater than the maximum (minimum) observed value of covX in the dataset are subsequently set to this maximum (minimum) value.
4	Continuous variable	PROC LOGISTIC & PROC REG	Fits a logistic model for indicator that covX is > 0 and a linear regression model for the natural log of covX restricting to records where covX>0. The procedure internally creates two variables: z_covX (indicator for covX>0) and l_covX (natural log of covX). covXotpye = 4 might be appropriate when covX has many zero values (e.g., number of cigarettes per day).	z_*covX* is simulated using coefficients estimated by PROC LOGISTIC. If the generated value of z_*covX* is 0 then *covX* is generated as 0. Otherwise if the generated value of z_*var* is 1, l_*covX* is generated using the second set of coefficients estimated by PROC REG and covX is set to the exponentiated , l_*covX*. Generated values greater than the maximum (minimum) observed value in the *dataset* are subsequently set to this maximum (minimum) value.
5	Categorical/ordinal variable	PROC LOGISTIC Calls j-1 times where j is the number of levels of covX	Dependent variable in first call is indicator that covX=1; dependent variable in second call is indicator that covX=2 among records where covX ne 1 and so on. Levels of *covX* must be coded as integers beginning at 1.	Levels of *covX* are simulated as integers beginning at 1 from the estimated model parameters.

6	Continuous variable	PROC QLIM (TRUNCATED option)	Fits a truncated normal regression model TRUNCATED option with lower bound set to the minimum value of *covX* in the observed dataset plus an offset term (-.01%) and upper bound set to the maximum value of *covX* plus an offset term (+.01%).	CovX is generated according to truncated normal using estimated regression parameters and sample variance of covX
7	Continuous variable	PROC QLIM (CENSORED option)	Fits Tobit regression model CENSORED option with lower bound set to the minimum value of covX in observed dataset and upper bound set to the maximum value of *covX*	CovX is generated according to a normal density with mean defined by the estimated regression model coefficients and variance estimated as sample variance of covX. Generated values greater than the maximum (minimum) observed value of covX in the dataset are subsequently set to this maximum (minimum) value.

Table 3: Summary of SAS regression procedures and simulation rules by choice of covXptype

covXptype	**Description**	**Details**	**Required additional parameters**
lag1- types: *lag1bin* *lag1cat* *lag1qdc* *lag1zqdc* *lag1cub* *lag1spl*	Function of most recent lag	The algorithm uses and creates a function of *covX* and *covX_l1*. The function is determined by the suffix (bin, cat, qdc,zqdc,cub,spl). For ‘‘preceding covariate models’’, the algorithm includes the suffix-determined function of *covX*_l1 as an independent variable; for ‘‘succeeding covariate models’’, the suffix-determined function of *covX* and *covX*_l1; for ‘‘outcome regression models’’, the function of *covX*. The user must include a variable named covX_l1 that corresponds to the first lagged value of *covX* in the input data set (e.g. if *covX*=cig then must include a variable named cig_l1).	

lag2- types: *lag2bin* *lag2cat* *lag2qdc* *lag2zqdc* *lag2cub* *lag2spl*	Function of most recent two lags	The algorithm uses and creates a function of *covX*, *covX_l1* and *covX_l2*.. The function is determined by the suffix (bin, cat, qdc,zqdc,cub,spl). For ‘’preceding covariate models’’, the algorithm includes the suffix-determined function of *covX* _l1 and *covX* _l2 as independent variables; for ‘’succeeding covariate models'', the suffix-determined function of *covX*, *covX* _l1 and *covX* _l2; for ‘’outcome regression models’’, the function of *covX* and *covX* _l1. The user must include variables named covX_l1 and covX_l2 that correspond to the first and second lagged values of covX, respectively, in the input data set (e.g. if *covX*=cig then must include variables named cig_l1 and cig_l2).	
lag3- types: *lag3bin* *lag3cat* *lag3qdc* *lag3zqdc* *lag3cub* *lag3spl*	Function of most recent three lags	The algorithm uses and creates a function of the lag variables *covX* _l1, *covX* _l2 and *covX* _l3 are created internally The function is determined by the suffix (bin, cat, qdc,zqdc,cub,spl). For ‘'preceding covariate models’', the algorithm includes the suffix-determined function of *covX* _l1, *covX* _l2 and *covX* _l3 as independent variables; for '‘succeeding covariate models’', the suffix-determined function of *covX*, *covX* _l1, *covX* _l2, and *covX* _l3; for ‘'outcome regression models'', the function of *covX*, *covX* _l1 and *covX* _l2. The user must include variables named covX_l1, covX_l2 and covX_l3 that correspond to the first, second and third lagged values of covX, respectively, in the input data set (e.g. if *covX*=cig then must include variables named cig_l1, cig_l2 and cig_l3).	
skp- types: skpbin skpcat skpqdc skpzqdc skpcub skpspl	Function of last measured value	The algorithm uses and creates (i) a function of *covX* and *covX* _l1 and (ii) interaction terms between each variable comprising these functions and the time between the current interval k and the last time *covX* was actually measured. The function is determined by the suffix (bin, cat, qdc,zqdc,cub,spl). This lapse of time is determined by the length of each interval — as defined by the macro parameter *interval* — and the intervals in which *covX* is not measured for any subject — as defined by the macro parameter *covXskip*. The skp type should only be used in data sets arising from cohort studies where time=0 (baseline) corresponds to the same calendar time for all subjects and for variables *covX* that are not measured during certain follow-up times for any subject. See FAQ 5. The user must include a variable named covX_l1 that corresponds to the first lagged value of covX in the input data set (e.g. if covX=cig then must include a variable named cig_l1).	covXskip
-bin types: lag1bin lag2bin lag3bin skpbin	No transformation	identity function (no tranformation)	
-cat types: lag1cat	Function of category indicators	transformed to category indicators, cutoffs for categories defined by *covXknots*,. Reference level is highest level	covXknots

lag2cat lag3cat skpcat			
-qdc types: lag1qdc lag2qdc lag3qdc skpqdc	Quadractic function		
-zqdc types: lag1zqdc lag2zqdc lag3zqdc skpzqdc	Quadractic function	Includes an interaction that variable is greater than zero	
-cub types: lag1cub lag2cub lag3cub skpcub	Cubic function	cubic function (linear, squared and cubed terms)	
-spl types: lag1spl lag2spl lag3spl skpspl	Splines	restricted cubic spline transformation; knots defined by covXknots	covXknots
tsswitch1	Function of time since covX switched from 0 to 1	Intended for any variable *covX* with covXotype=2 (e.g. an indicator of disease diagnosis by interval k) to allow for modelling the history of covX as a function of the time since covX last switched from 0 to 1 at each time k. Specifically, for “preceding covariate models” (excepting Y=X), the algorithm includes *covX* _l1 and an internally created variable *tscovX _l1_inter* (the product of *covX* _l1 and the cumulative sum of *covX* from time=0 through time=k-1) as independent variables; for “succeeding covariate” and “outcome regression models”, *covX* and internally created *tscovX _inter* (the product of *covX* and the cumulative sum through k) are included as independent variables. The user must include a variable named covX_l1 that corresponds to the first lagged	

		value of covX in the input data set (e.g. if covX=cig then must include a variable named cig_l1).
cumavg	Cumulative average	Creates and includes the cumulative average of entire history of *covX* relative to interval k beginning from time=0.
lag1cumavg	Cumulative average where the last term is pulled off the average	A variation of the cumavg ptype where the last term is pulled off of the average. In this case there are two generated predictors. At time = k these will be *covX* _l1 and the average of *covX* from time = 0 to time = k-2.
lag2cumavg	Cumulative average where the last two terms are pulled off the average	A variation of the cumavg ptype where the last two terms are pulled off of the average. In this case there are two generated predictors. At time = k these will be covX_l1, covX_l2, and the average of *covX* from time = 0 to time = k-3.
rcumavg	Recent cumulative average	Creates and includes the cumulative average of restricted history of *covX* relative to interval k based on two most recent values only.

References

Danaei G, Pan A, Hu FB, Hernán MA. Hypothetical lifestyle interventions in middle-aged women and risk of type 2 diabetes: a 24-year prospective study. *Epidemiology* 2013; 24(1):122-128.

Garcia-Aymerich J, Varraso R, Danaei G, Camargo CA, Hernán MA. Incidence of adult-onset asthma after hypothetical interventions on body mass index and physical activity. An application of the parametric g-formula. *American Journal of Epidemiology* 2014; 179(1):20-26.

Hernán MA, McAdams M, McGrath N, Lanoy E, Costagliola D. Observation plans in longitudinal studies with time-varying treatments. Statistical Methods in Medical Research 2009;18(1):27-52.

Lajous M, Willett WC, Robins JM, Young JG, Rimm E, Mozaffarian D, Hernán MA. Changes in fish consumption in midlife and the risk of coronary heart disease in men and women. American Journal of *Epidemiology* 2013; 178(3):382-91.

Lodi S, Phillips A, Logan R, Olson A, Costagliola D, Abgrall S, van Sighem A, Reiss P, Miró JM, Ferrer E, Justice A, Gandhi N, Bucher HC, Furrer H, Moreno S, Monge S, Touloumi G, Pantazis N, Sterne J, Young JG, Meyer L, Seng R, Dabis F, Vandehende MA, Pérez-Hoyos S, Jarrín I, Jose S, Sabin C, Hernán MA; HIV-CAUSAL Collaboration. Comparative effectiveness of strategies for antiretroviral treatment initiation in HIV-positive individuals in high-income countries: immediate universal treatment versus CD4-based initiation. *Lancet HIV* 2015; 2(8):e335-343.

Lodi S, Sharma S, Lundgren JD, Phillips AN, Cole SR, Logan RW, Agan BK, Babiker A, Klinker H, Chu H, Law M, Neaton JD, Hernán MA.The per-protocol effect of immediate versus deferred ART initiation. *AIDS* 2016; 30(17):2659-63.

Lodi S, Phillips A, Lundgren J, Logan R, Sharma S, Cole SR, Babiker A, Law M, Chu H, Byrne D, Horban A, Sterne JAC, Porter K, Sabin C, Costagliola D, Abgrall S, Gill J, Touloumi G, Pacheco AG, van Sighem A, Reiss P, Bucher HC, Montoliu Giménez A, Jarrin I, Wittkop L, Meyer L, Perez-Hoyos S, Justice A, Neaton JD, Hernán MA. Effect estimates in randomized trials and observational studies: comparing apples with apples. Am J Epidemiol. 2019 May 07. PMID: 31063192.

Robins JM. A new approach to causal inference in mortality studies with a sustained exposure period: application to the healthy worker survivor effect. Mathematical Modelling, 7:1393–1512, 1986. [Errata (1987) in Computers and Mathematics with Applications 14, 917-921. Addendum (1987) in Computers and Mathematics with Applications 14, 923-945. Errata (1987) to addendum in *Computers and Mathematics with Applications* 18, 477.

Richardson TS and Robins JM. Single World Intervention Graphs (SWIGs): a unification of the counterfactual and graphical approaches to causality. Center for Statistics and the Social Sciences, University of Washington Series, 2013. URL http://www.csss.washington.edu/Papers/. Working Paper Number 128.

Robins JM. Causal inference from complex longitudinal data. In M. Berkane, editor, Latent Variable Modeling and Applications to Causality. Lecture notes in statistics 120, pages 69–117. Springer-Verlag, 1997.

Taubman SL, Robins JM, Mittleman MA, Hernán MA. Intervening on risk factors for coronary heart disease: an application of the parametric g-formula. *Int J Epidemiol* 2009; 38(6):1599-611.

Young JG, Cain LE, Robins JM, O'Reilly E, Hernán MA. Comparative effectiveness of dynamic treatment regimes: an application of the parametric g-formula. *Statistics in Biosciences* 2011;3:119-143.

Young JG, Hernán MA, Robins JM. Identification, estimation and approximation of risk under interventions that depend on the natural value of treatment using observational data. *Epidemiologic Method*s 2014; 3(1):1-19.

Gooley TA, Lessening W, Crowley J, Storer BE. Estimation of failure probabilities in the presence of competing risks: new representations of old estimators. *Statistics in Medicine* 1999. 18(6):695-706.

Kalbfleisch, J. D. and Prentice, R. L. The Statistical Analysis of Failure Time Data, John Wiley, New York, 1980.