The parametric g-formula in SAS

JESSICA G. YOUNG

CIMPOD 2017

CASE STUDY 2

Structure of the workshop

Part I: Motivation

- ➢Why we might use the parametric g-formula to and how it works in general

Part II: GFORMULA SAS macro

- ➢Structure of the macro
- ➢Sample code

GFORMULA macro

Contributors: Miguel Hernán, Sarah Taubman, Roger Logan, Jessica Young, Sara Lodi, Sally Picciotto, Goodarz Danaei

Version on web: 2.0

Version for today: 3.0

Contact info

Updates to macro and documentation:

https://www.hsph.harvard.edu/causal/software/

My email:

jessica.gerald.young@gmail.com

PART I: MOTIVATION

Case study

Young et al. Comparative effectiveness of dynamic treatment regimes: an application of the parametric g-formula. *Statistics in Biosciences* (2011).

Interested in estimating the **causal effect of following different cART initiation strategies on 5-year mortality risk in an HIV-infected population**.

Causal effect

Population causal effects can be formally defined in terms of contrasts in counterfactual outcome distributions associated with different treatment strategies :

- What would happen to the population 5 year mortality risk if, *possibly contrary to fact*, we implemented one rule for initiating cART versus another rule in a given HIV infected population?

Young et al:

Causal 5 year risk ratio/difference comparing different dynamic strategies of the form:

Start cART within *m* months of CD4 cell count first dropping below *x* cells/mm^3 or diagnosis of an AIDS-defining illness, whichever happens first"

where *x* can take values 200 and 500 (increments of 50).

Can think of *m* as a grace period

Special case m=0 (no grace period)

“Start cART as soon as CD4 cell count first drops below x cells/mm^3 or there is a diagnosis of an AIDS-defining illness, whichever happens first"

Dynamic strategies

- These strategies indexed by cutoff *x* are examples of time-varying dynamic treatment strategies
- Dynamic: Strategies under which treatment assignment at time k during follow-up is determined by time-evolving patient characteristics
    - At baseline, treatment assignment at a later time is not yet known for all patients
- Static: Treatment assignment at all future times known at baseline (e.g. “never treat”)

Ideal RCT

If we could, we would estimate causal effects of time-varying treatment strategies in an ideal randomized controlled trial:

- Baseline randomization: subjects randomized to one of each of these strategies *x*
- Full compliance with protocol until death or 5 years later (whichever comes first)
- No “censoring” (e.g. no loss to follow-up)

Ideal RCT

- No confounding (by design)
- No selection bias (by design)
- Unbiased estimate obtained via simple contrast proportions

Challenge to causal inference

Ideal RCTs are often not feasible

- Too costly
- Untimely
- Unethical

Alternative: Observational studies

Observational data

Since publication Young et al. (2011), RCTs have actually been conducted to answer this question (at the time there were none!)

At that time, we used observational data from the HIV-CAUSAL collaboration to try and estimate the causal effect of interest

HIV-CAUSAL collaboration

- Includes several cohort studies from five European countries and the United States
- Cohorts assembled prospectively and based on data collected for clinical purposes within national health care systems with universal access to care

Study population for analysis

Eligibility criteria:

- In data set between 1996 and 2009
- no history of CD4 cell count less than 500 cells/mm3;
- 18 years or older;
- not pregnant
- CD4 cell count and viral load (HIV RNA) measurements within 6 months of each other at baseline.

Study population for analysis

- Defined “baseline” as first month after meeting all eligibility criteria that CD4 dropped into range 200-499 cells/mm^3
- Think of “baseline” as time we would randomize that patient to strategy *x* if we were running an RCT
- Follow up time broken up into months
- Censored subjects at month of pregnancy or at the 12th consecutive month without a viral load or CD4 cell count measurement.

Confounding and assumptions

- In HIV-CAUSAL there is confounding (no physical randomization at any time, no “forcing”)
  - People who start CART earlier may be healthier or sicker than those who start later
- There is also selection bias: some subjects are censored

Confounding and assumptions

If we are willing to assume “no **unmeasured** confounding or selection bias” (NUCS) we can get an unbiased estimate

- NUCS: Measured variables are sufficient to control confounding and selection by unmeasured risk factors
- NUCS is an untestable assumption – cannot test with study variables

No unmeasured confounding

Key features of NUCS:

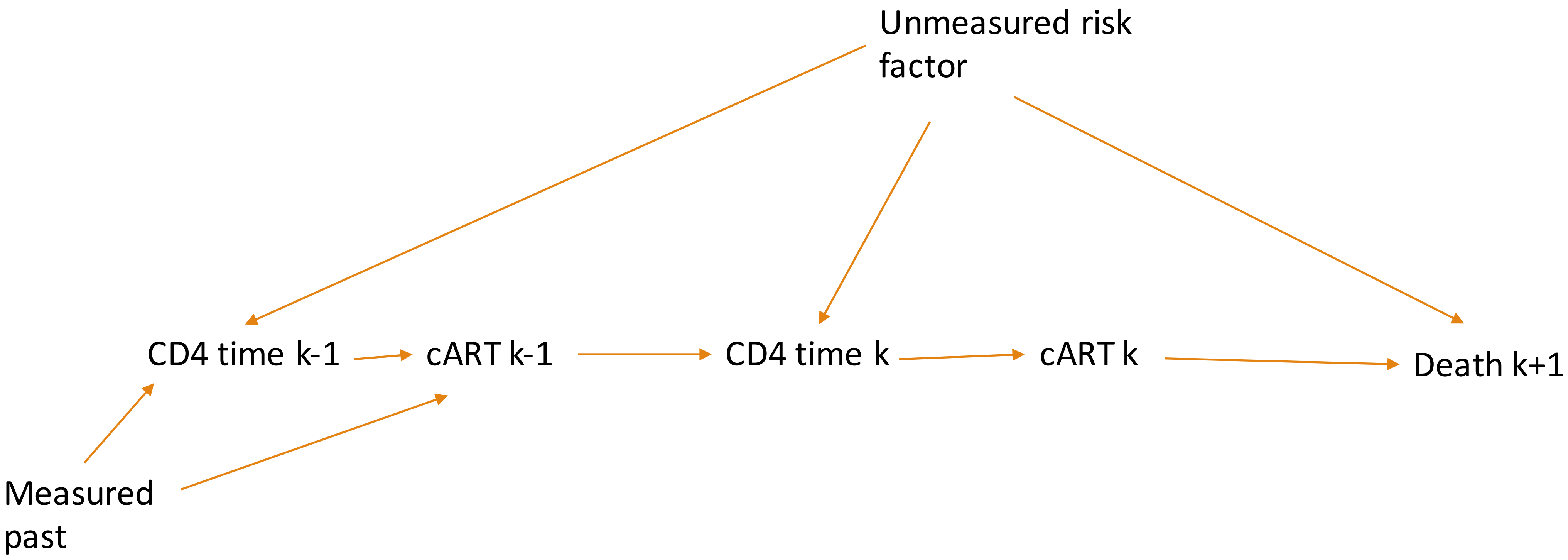
1. Allows presence of measured time-varying confounders (in addition to baseline confounding)

   - CD4 at k predicts future mortality and future treatment.

2. Also allows that measured time-varying confounders are themselves affected by past treatment

   - E.g. CD4 at k affected by past treatment

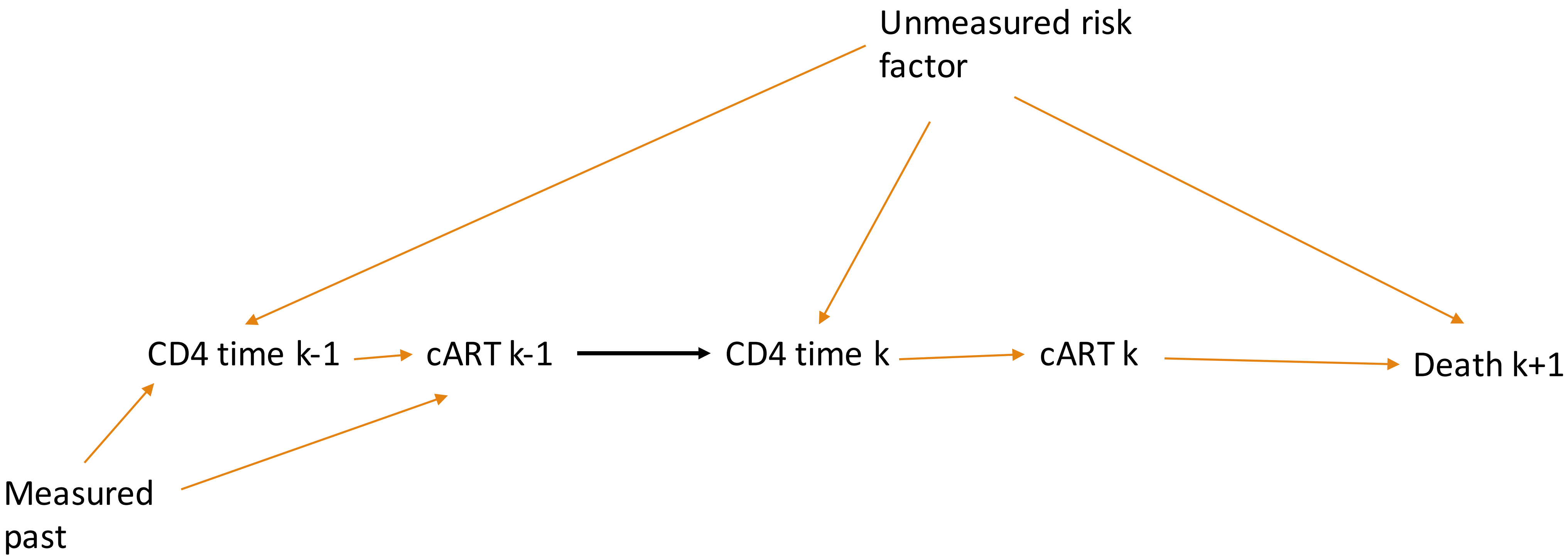
CAUSAL DAG REPRESENTING NO UNMEASURED CONFOUNDING



Absence of arrows from unmeasured risk factor into exposure guarantees no "unblocked backdoor paths" between exposure and outcome given measured past at any time.

Backdoor paths = confounded paths; Directed paths = causal paths

CD4 MEASURED TIME-VARYING CONFOUNDER AFFECTED BY TREATMENT



Unmeasured risk factor

CD4 time k-1

cART k-1

CD4 time k

cART k

Death k+1

Measured past

ALLOWS THAT MEASURED TIME-VARYING CONFOUNDER AFFECTED BY PAST TREATMENT
(“NO UNMEASURED CONFOUNDING” ALLOWS THIS STRUCTURE)

Time-varying confounding and standard regression

Turns out that under this type of data structure, even though we can get an unbiased estimate,

- We cannot get it via standard regression approaches
- We need other approaches
- Why?

Standard outcome regression

We might think to fit regression model for death hazard at a given time with independent variables:

- Function of time-varying treatment initiation indicator – maybe cumavg of these indicators
- Function of baseline and time-varying confounders -- cumulative average of CD4 through time k

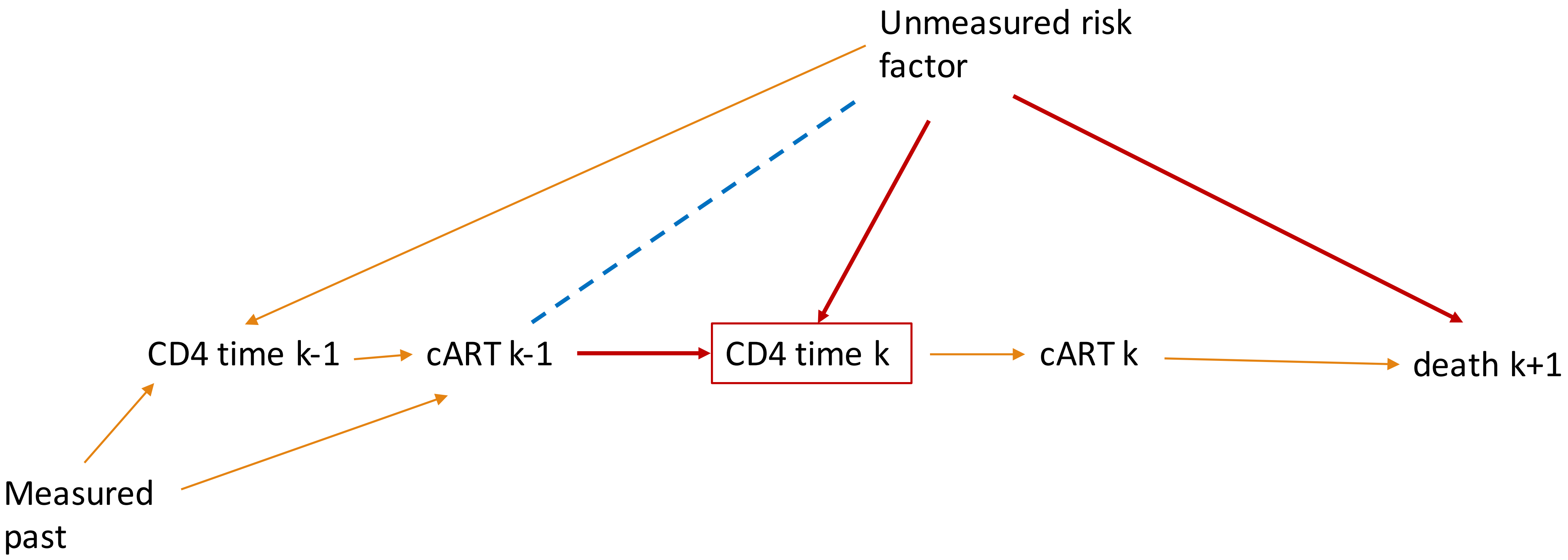
Use estimated coefficient on cumavg of treatment indicators as estimate of time-varying causal treatment effect

Standard outcome regression

Problem: even given “no unmeasured confounding” and outcome regression model correctly specified

- Coefficient estimate is a biased estimate under our causal DAG

CD4 MEASURED TIME-VARYING CONFOUNDER AFFECTED BY TREATMENT



INCLUDING FUNCTION OF CD4 AT k IN REGRESSION MODEL IS CONDITIONING ON IT

CONDITIONING ON COLLIDER OPENS UP NONCAUSAL PATH BETWEEN ITS CAUSES

Estimation in observational data with time-varying confounding

If not, standard regression, then how to proceed?

In this case, methods that derive from Robins' g-formula can remain valid:

- give unbiased estimates of time-varying causal treatment effects in the face measured time-varying confounding affected past treatment.

The g-formula

Robins (1986) showed that, given NUCS

- The counterfactual outcome mean/risk associated with a user-specified time-varying treatment strategy g can be written as the g-formula
- The g-formula is a particular function of the baseline and time-varying data
- Estimated contrasts in this function for different choices of g can give unbiased estimates of causal effects
- Also requires “positivity” assumption

G-formula for Risk by end of follow-up under intervention g

Can write as weighted average of conditional risks

- Each risk conditioned on a possible treatment and confounder history observable under g and no censoring
- Weights are function of joint distribution of measured confounders at each time k conditional on past history observable under g and no censoring
    - i.e. the “chance” of observing each confounder history (under g) and no censoring

G-formula for Risk by end of follow-up under intervention g

Weights can also be a function of the observed distribution of treatment at each time conditional on past history observable under g and no censoring

- This will be the case when g is defined in terms of intervention that depends on this distribution
- g will not depend on this distribution when m=0 but does when m>0 (see Young et al, 2011)

How to estimate this function

In typical high-dimensional settings, we require parametric models to estimate the g-formula.

Different methods rely on different types of model assumptions

- Parametric g-formula: imposes models directly on components of weighted average
- Other methods derive from alternative representations of this weighted average which suggest constraining different quantities (e.g. IPW of MSMs, DR methods like TMLE)
- Equivalent under saturated models

Parametric g-formula Algorithm (Step 1)

First fits parametric models for

- Discrete hazard at each time conditional on past measured treatment and confounders
- Joint distribution of treatment and confounders at each time given past
- Models are generally pooled over time

Modelling joint distribution of covariates at k

Model based on arbitrary factorization of covariates at k. E.g.

$f(cd4_k, rna_k | \text{past through } k\text{-}1)$ is the same as

1. $f(cd4_k | rna_k, \text{past through } k\text{-}1) * f(rna_k | \text{past through } k\text{-}1)$ or
2. $f(rna_k | cd4_k, \text{past through } k\text{-}1) * f(cd4_k | \text{past through } k\text{-}1)$

Depending on choice of factorization, you are modelling the components of product 1 or product 2

- In absence of model misspecification, equivalent.
- Deterministic relationships may favor one factorization (Young et al., 2011)

Algorithm: Step 2

N times (default sample size) do the following iteratively for each k:

- Simulate treatment and confounders at each time k using estimated model parameters from Step 1. Exception: at k=0 (baseline) assign values as observed values in data set.
- Reset treatment at k according to user-defined rule *g*
- Estimate hazard of event at time k given these generated covariate values using model in Step 1

Algorithm: Step 3

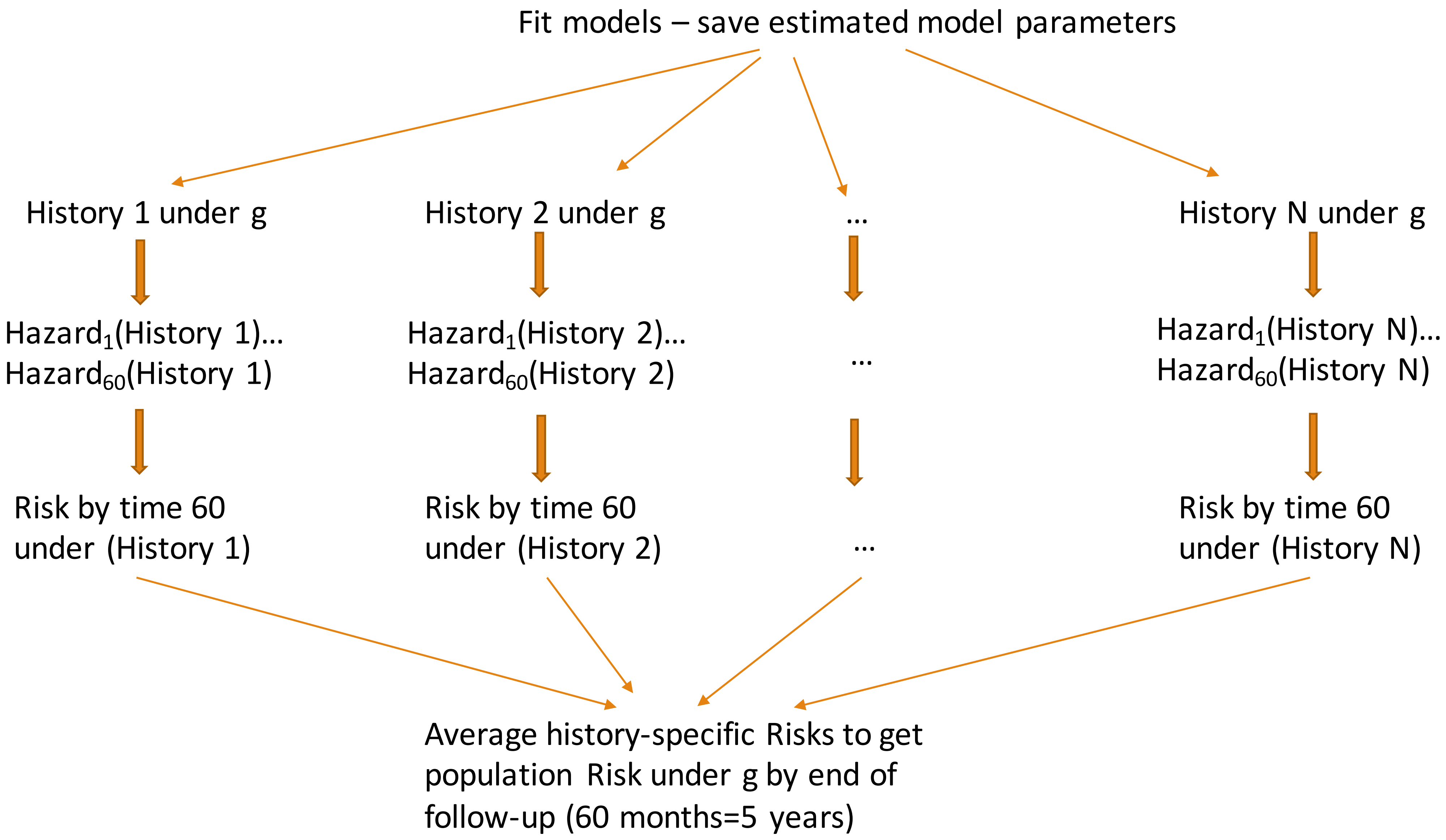
- Compute the Risk by end of follow-up for each of the N simulated histories from the N time-varying history-specific hazards.
- Average these Risks to get final estimate of Risk by end of follow-up under g

Final estimates and CIs

Repeat Steps 2 and 3 for each hypothetical intervention.

Obtain causal effect estimates from by risk differences/ratios for different g.

95% CIs obtained by repeating whole algorithm in B bootstrap samples.



Fit models – save estimated model parameters

History 1 under g

Hazard$_1$(History 1)…
Hazard$_{60}$(History 1)

Risk by time 60
under (History 1)

History 2 under g

Hazard$_1$(History 2)…
Hazard$_{60}$(History 2)

Risk by time 60
under (History 2)

…

…

…

History N under g

Hazard$_1$(History N)…
Hazard$_{60}$(History N)

Risk by time 60
under (History N)

Average history-specific Risks to get
population Risk under g by end of
follow-up (60 months=5 years)

Disadvantages of parametric g-formula

- Relies heavily on parametric models and subject to related bias
- Some model misspecification can be theoretically guaranteed when null of no treatment effect is true
  - “null paradox” (Robins and Wasserman, 1997)

Advantages of parametric g-formula

- More stable than other methods for continuous exposures and given "near positivity violations"
  - Occurs when a level of treatment under g is very unlikely for certain observed confounder histories
  - Parametric g-formula handles by heavier reliance on extrapolation
- Generally, the complexity of algorithm is the same for any choice of g
  - Very little change for complex *dynamic* rules

PART II: GFORMULA SAS macro

Different types of outcomes

Macro supports 3 types of outcomes (*outctype)*

1. Continuous outcome *at* end of follow-up time
   - Choose when interest is in t-v treatment effect on an *outcome mean at end of follow-up*
   - E.g. mean CD4 cell count at 5 years post-baseline
   - *outctype =conteofu*

Different types of outcomes

Macro supports 3 types of outcomes (*outctype)*

2. Binary outcome *at* fixed end of follow-up time
   - Choose when interest is in t-v treatment effect on probability that outcome occurs *at* end of follow-up
   - E.g. Probability of obesity *at* 5 years post-baseline
   - *outctype =bineofu*

Different types of outcomes

Macro supports 3 types of outcomes (*outctype)*

3. Time-varying indicator of failure event
   - Choose when interest is in t-v treatment effect on *risk by end of follow-up*
   - E.g. Mortality risk *by* 5 years post-baseline
   - ***outctype =binsurv***

Required structure for input data set

Requires a person-time data set with one record per subject and measurement time index

- Time index (*time*) must start at 0 (baseline) for each subject and increment by 1 for each subsequent time index.
- Time index represent a time interval in which covariates are measured
- Young et al.: each time index represents a month long interval

Required structure for input data set

Each person-time record will include

- Time-fixed baseline covariates (e.g. pre-baseline cd4, rna, race)
- Current covariate measurements for that time k (cd4, rna indicator of cART initiation in interval k)
- Time-varying indicator of censoring (e.g. indicator that subject has reached 12 consecutive months without lab measurement)

Required structure for input data set

For *outctype*=binsurv (Young et al.):

- Will also contain a time varying indicator of failure from event of interest for each time index k

Required structure for input data set

For *outctype*=binsurv (Young et al.):

- Time varying indicator of failure on line k can be coded 0, 1 or missing
  - Should be 0 if neither event nor censoring has occurred
  - Should be 1 if event has occurred
  - Should be missing if no event but censoring occurs
- First line k where outcome is 1 or missing is last line for that subject.

Required structure for input data set

Subjects who do not fail and are not censored by end of follow-up will be 0 at all times for censoring and event indicators (outctype=*binsurv*)

- ➢Macro parameter *timepoints* encodes end of follow-up in terms of intervals
- ➢Young et al.: timepoints=60 (60 months=5 years)
- ➢Because time index *time* starts at 0, can take maximum value of 59

SAMPLE DATA

FEATURES OF BASIC CALL

```
libname jess '/sasdata/CIMPOD_2017/Jessica_Young';
```

Define permanent libraries

```
%include '/sasdata/CIMPOD_2017/Jessica_Young/gformula.sas';
```

Include the file with the macro

```
options notes;
*options mprint;
```

Set options for printing in log file

```
data hivdata;
set jess.hivdata;
run;
```

Call permanent data set

Define interventions

```
%let interv1 =intno=1,        intlabel='always treat', nintvar=1,intvar1=art, inttype1=1, intvalue1=1, intpr1=1,
inttimes1=  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30     31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59        ;

%let interv2 =intno=2,        intlabel='never treat', nintvar=1,intvar1=art, inttype1=1, intvalue1=0, intpr1=1,
inttimes1=  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30     31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59      ;
```

Call to GFORMULA macro

%gformula(	
data=hivdata,	Input data set
id=id,	Subject identifier from input data set
time=month,	Time index from input data set
timeptype = conbin,	Specifies function of time for pooled over time models
timepoints=60,	End of follow-up (max value of time should be timepoints-1)
outctype=binsurv,	Specifies outcome is t-v binary failure indicator
outc=death,	Time-varying failure indicator for event of interest
censlost=censor,	Time-varying censoring indicator
numint= 2,	Number of interventions to be simulated
fixedcov= rna_0 cd4_0 age_0 yrshiv,	Time fixed baseline confounders
ncov=2,	Number of time-varying covariates (including treatment), up to 30
cov1=lncd4,	Time varying covariate 1
cov1otype=3, cov1ptype = lag1bin,	Model specifications for t-v covariate 1
cov2=art,	Time varying covariate 2
cov2otype=2, cov2ptype=lag1bin,	Model specifications for t-v covariate 2
nsamples= 0);	Number of bootstrap samples

Graphical comparisons of natural course versus observed

- Set macro parameter *rungraphs=1*
- Compares “observed risks” (nonparametric estimates in censored data) versus parametric g-formula estimates under no intervention (“natural course”) at each follow-up time
- Analogous comparison of “observed” versus “simulated” covariate means
- Used to get a sense of presence of gross model misspecification

REVIEW OF OUTPUT

covXotypes

For each covX, X=1,…,ncov:

- The macro parameter *covXotype* selects the SAS regression fitting procedure for the conditional distribution of *covX.*
- Also determines how covX is simulated at each time *k*.
- Options available for *covXotype* are summarized in Table 1 of the documentation.

Examples of covXotype

- covXotype=1, estimates the conditional density of covX via PROC LOGISTIC. Simulation based on estimated model parameters.
  - Might be appropriate for binary variables that can take value 1 or 0 at any time with no restriction.
- covXotype=2, estimates via PROC LOGISTIC amongst records with lagged value of covX=0. Simulates from model if last simulated value of covX was 0. If last value was 1, sets covX to 1.
  - Might be appropriate for binary variables that once they switch to 1, they stay 1 (e.g. initiating treatment *by* time k)

Examples of covXotype

- covXotype=3, estimate of conditional density of covX obtained via PROC REG. Simulation based on estimated model parameters under assumption of a normal distribution.
  - Might be appropriate for continuous variables.

covXptypes

Determines how the “history” of covX will appear in each model

- Hazard model
- Models for conditional covariate distributions
- Options for covXptype are in Table 2 of documentation

covXptypes (lag1- prefix)

Prefix *lag1-* (lag1bin, lag1qdc,lag1zqdc,lag1cat,lag1spl)

- Includes function of one lagged value of covX in covariate models
- Includes function of current value of covX only in hazard models
- Function depends on choice of suffix
- Need to include lagged value of covX in input data set
  - Must be named covX_l1 (e.g. lncd4_l1 if covX=lncd4)

covXptypes (suffix options)

Suffix options that determine function:

- lag1bin: identity function (linear assumption when covX is not binary)
- lag1qdc: quadratic function
- lag1cat: include indicators of categorization of covX (must also specify *covXknots* which give cutoffs for categories)
  - E.g. cov1=lncd4, cov1ptype=lag1cat,cov1knots= 4 6 8,…
- lag1spl: restricted cubic spline (must also specify covXknots)

covXptypes (lag2- prefix)

Prefix *lag2-* (lag2bin, lag2qdc,lag2zqdc,lag2cat,lag2spl)

- Includes function of two recent lagged values of covX in covariate models
- Includes function of current value of covX and covX_l1 only in hazard models
- Function depends on choice of suffix
- Need to include two lagged values of covX in input data set
  - Must be named covX_l1 and covX_l2

Other *covXptypes* (from Table 2 of documentation)

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.

Exercise 1

Edit hivcall1.sas

1. Change the covXptype for time-varying covariate *lncd4* so models include indicators for categories of first lagged value of *lncd4* (can use cutoffs 5.81, 6.21 and 6.58).
2. Add the baseline confounder *sex* to the macro call

Solution in hivexercise1.sas

Defining interventions

- Interventions are defined before the call to the main GFORMULA macro in global macro variables interv1, interv2…
- Table 3 in documentation describes different available types
- Do not need to define the natural course (by default this is run and is the default reference for causal contrasts)
  - Change reference by *refint* parameter
- In a given intervention definition, can include up to 8 treatment variables (variables to undergo intervention)

Code for static intervention– “never treat with cART”

```
%let interv1 = intno=1,   ←   Intervention number
intlabel=’never treat',          ←   Intervention label
nintvar=1,   ←   Number of variables to undergo intervention in this intervention
intvar1=art,   ←   First intervention variable
inttype1=1,   ←   Type of intervention on first intervention variable (Table 3)
intvalue1=0,←   When static intervention (inttype1=1), what is the value to assign
intpr1=1,   ←   Probability to perform this intervention on first intervention variable (default)
inttimes1=  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
23 24 25 26 27 28 29 30  31 32 33 34 35 36 37 38 39 40 41 42 43 44
45 46 47 48 49 50 51 52 53 54 55 56 57 58 59;        ←   Intervention times for first
                                                         intervention var
```

User-defined rules

- Dynamic rules hard to pre-code
- Macro allows you to code your own interventions
- Set *inttype1*=-1
- Must provide name of a user-defined macro containing the rule
- User-defined macro is called within a loop over values of *time*
- Can set *testing=1* to output simulated data set and check logic, name of permanent library assigned to *savelib* (e.g. *savelib=jess*)

DEFINING STRATEGIES X IN USER-DEFINED MACRO

(hivcall2.sas)

User-defined interventions

- Many ways to code the same intervention via a user-defined macro
- Documentation: give a different way to do it
  - More complicated
  - More easily accommodates the grace period (case where m>0)
  - Adds “trackers” of times intervened for certain parts of output to be useful (“percent intervened on”)

Exercise 2

Edit hivcall2.sas

1. Add a third strategy with cutoff x=350

Hint: remember to change *numint* in main GFORMULA macro call from 2 to 3

Solution: hivexercise2.sas

Random visit process

- Covariates lncd4 and lnrna represent last measured values
- Subjects do not come to the clinic every month ("clinical cohort")
- This visit process can itself be a time varying confounder (Hernán et al, 2008)
- Covariates visit_cd4 and visit_rna are indicators of whether lncd4 and lnrna are current measurements, respectively.
- Could add as separate "covX's" but…

Random visit process

This would blindly model visit_cd4 and lncd4.

Could in principle minimize model misspecification by incorporating deterministic knowledge that

1. if visit_cd4=0 then lncd4=lncd4_l1
2. Subjects are censored when they miss 12 consecutive lab measurements (so max sum of either visit indicator in the data is 12)

Automated options for incorporating this knowledge of the data

Random visit process

For a time-varying covariate (e.g. covX=lncd4) with a “visit process” (e.g. visit_cd4) can define additional macro parameters to incorporate

1. assumption that visit indicators are also time-varying confounders

2. deterministic relationships when modelling and simulating covX and its visit indicator.

Random visit process

Add following if covX=lncd4

- covXrandomvisitp=visit_cd4
- covXvisitpmaxgap=12
- covXvisitpcount=ts_last_cd4_l1

This is name of time-varying variable in input data set that has time since last measurement of covX

Analogous syntax for lnrna

Exercise 3:

Edit hivexercise1.sas

Update call so that you

1. Incorporate assumption that random visit processes for *lnrna* is a time-varying confounder
2. Incorporate deterministic knowledge in modelling and simulation that (i) max value of sum of rna visit indicator (visit_rna) can be 12 and (ii) if visit_rna=0 then lnrna=lnrna_l1

Note: data set has variable ts_last_rna_l1 that is time since last rna measurement at baseline.