



Derive Value From Excellence ...

A case study of fitting time dependent covariates in Cox model using SAS

Pankaj Kumar, GCE Solutions, 07 Nov 2018

PhUSE EU Connect 2018
Paper#AS05



Agenda

- Overview Cox proportional model
- Fixed covariate and time dependent covariate
- Counting process approach
- Programming statement approach
- SAS Code and Output



Cox proportional hazard model

- Kaplan Meir Plot and Logrank test
- Number of Events, patients at risk
- Number of Event, patient at risk, covariate

Hazard is modelled as:

$$h(t, X) = h_0(t) \exp(\beta_1 X_1 + \beta_2 X_2 + \dots)$$



Fixed and time dependent covariate

- Usual fixed covariate
- Time dependent covariate
 - Categorical e.g. Smoking habit over time
 - Continuous e.g. Biomarker value at various visits

```
/*Proc PHREG statement*/
```

```
proc phreg data=data1;
```

```
class treatment ;
```

```
model time*censor(1) = treatment;
```

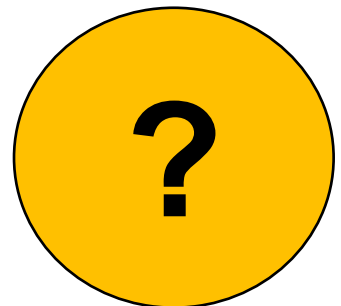
```
run;
```

```
proc phreg data=data1;
```

```
class treatment ;
```

```
model time*censor(1) = treatment change ;
```

```
run;
```





Case study:

- This is based on 9000 subjects, grouped into four treatment arms with 2225 subjects in each arm. hsCRP data is collected at baseline and at multiple visits during the study. We are interested in assessing effect of hsCRP on event of stroke.
- Cox model with treatment, baseline hsCRP and time varying hsCRP as explanatory variable is fitted
 - this can be done using Cox model with counting process approach
 - Alternatively same can be done using programming statement approach



Sample data

SUBJID Subject identifier

AVAL Time to event

TRT01PN Treatment

CNSR Censoring information,
0: Event and 1: censored

ADY Analysis relative day

LHSCR P log(hsCRP)

LBASE log(hsCRP) at baseline

TMLCRP time varying change from
Baseline in log(hsCRP)

Subject Identifier	Treatment	Censor	Time to event	Analysis relative day	Log(hsCR P)	Log(hsCR P) at baseline	Change from baseline
117008	1	1	190	0	2.821	2.821	0
117008	1	1	190	14	2.092	2.821	-0.73
117008	1	1	190	42	2.303	2.821	-0.519
117008	1	1	190	118	1.526	2.821	-1.295
141007	4	1	182	0	1.082	1.082	0
141007	4	1	182	15	1.03	1.082	-0.052
141007	4	1	182	46	1.946	1.082	0.864
141007	4	1	182	94	1.589	1.082	0.507
150017	2	0	73	0	2.447	2.447	0
150017	2	0	73	15	0.588	2.447	-1.859
150017	2	0	73	43	0.588	2.447	-1.859
150017	2	0	73	100	1.386	2.447	-1.06
150017	2	0	73	747	1.435	2.447	-1.012
150017	2	0	73	926	0.788	2.447	-1.658



Counting process approach

- multiple records are created for each subject
- time to event is divided in such intervals that one record for each pattern of the time-dependent measurements

Subject Identifier	Treatment	Event	Time to event	Log(hsC RP) at baseline	Start Value	End Value	time varying change from Baseline
117008	1	0	190	2.821	0	14	0
117008	1	0	190	2.821	14	42	-0.73
117008	1	0	190	2.821	42	118	-0.519
117008	1	0	190	2.821	118	190	-1.295
141007	4	0	182	1.082	0	15	0
141007	4	0	182	1.082	15	46	-0.052
141007	4	0	182	1.082	46	94	0.864
141007	4	0	182	1.082	94	182	0.507
150017	2	0	73	2.447	0	15	0
150017	2	0	73	2.447	15	43	-1.859
150017	2	1	73	2.447	43	73	-1.859



Counting process approach continued...

```
/*Proc PHREG statement*/  
ods output ParameterEstimates = est1;  
proc phreg data=all;  
    class trt01pn (ref='4');  
    model (stval,enal)*event(0) = trt01pn lbase tmlcrp /  
ties=exact risklimits;  
    output out= dfb1 dfbeta = dt1-dt5; /* DFBETA estimates*/  
run;
```

Parameter	ClassVal 0	DF	Parameter Estimate	Standard Error	Chi- Square	Pr > ChiSq	Hazard Ratio	95% Lower Confiden ce Limit	95% Upper Confiden ce Limit
TRT01PN	1	1	-0.14188	0.17991	0.6219	0.4303	0.868	0.61	1.235
TRT01PN	2	1	0.06002	0.16798	0.1277	0.7209	1.062	0.764	1.476
TRT01PN	3	1	0.07587	0.1711	0.1966	0.6574	1.079	0.771	1.509
Lbase		1	0.2635	0.08807	8.9512	0.0028	1.301	1.095	1.547
Tmlcrp		1	-0.13961	0.06598	4.4778	0.0343	0.87	0.764	0.99



Programming statement approach

- data has one record for each subject
- The time-varying covariates are defined in programming statements that are part of the PHREG step

Subject Identifier	chg1	chg2	chg3	chg4	chg5	day1	day2	day3	day4	day5	Time to event	Censor
117008	0	-0.73	-0.519	-1.295	.	0	14	42	118	.	190	1
141007	0	-0.052	0.864	0.507	.	0	15	46	94	.	182	1
150017	0	-1.859	-1.859	.	.	0	15	43	73	.	73	0



Programming statement approach continued...

```
proc phreg data=all;
    class trt01pn (ref='4');
    model aval*cnsr(1) = trt01pn lbase tmlcrp / ties=exact
risklimits;
    tmlcrp= 0 ;
    array day(*) day1 - day%left(&number);
    array chg(*) chg1 - chg%left(&number);
    do j=1 to &number;
    if aval > day[j] and day[j] ne . then tmlcrp=chg[j];
    end;

run;
```

Parameter	ClassVal 0	DF	Parameter Estimate	Standard Error	Chi- Square	Pr > ChiSq	Hazard Ratio	95% Lower Confiden ce Limit	95% Upper Confiden ce Limit
TRT01PN	1	1	-0.14188	0.17991	0.6219	0.4303	0.868	0.61	1.235
TRT01PN	2	1	0.06002	0.16798	0.1277	0.7209	1.062	0.764	1.476
TRT01PN	3	1	0.07587	0.1711	0.1966	0.6574	1.079	0.771	1.509
Lbase		1	0.2635	0.08807	8.9512	0.0028	1.301	1.095	1.547
Tmlcrp		1	-0.13961	0.06598	4.4778	0.0343	0.87	0.764	0.99



Conclusion

- Counting process and programming statement, both gives same result for handling time-varying covariate in Cox regression
- The programming statements are faster to code, but we should be careful while coding
- Two approaches can be used for QC of each other
- Counting process is preferable as it is certain benefits over programming statement methods
 - Dataset is created, and we can debug errors easily in counting process approach
 - Using counting process approach, we can create DFBETA estimates, and other residuals but using programming statement we can't get these.
- Same approach can be used for multiple time-dependent covariates, recurrent events etc.



SAS Code

```
/**      Input data for counting process approach**/
/**      *****/
data adef;
    set adep;
    by subjid ady;
    /*      STCHG:  log(hsCRP) Change from Baseline at Start Time*/
    /*      ENCHG:  log(hsCRP) Change from Baseline at End Time*/
    retain stval enval stchg enchg;
    /*Assign Zero to the Start Time/Change for the First Interval*/
    if first.subjid then
        do; stval=0; enval= 0; stchg = 0; enchg=0;          end;
    /*Carry Over Start Time/Change from the Previous End Time/Change*/
    /*Record the Current ADY and CHG as End Time/Change for Next Record */
    if ady ne . and ady < aval then
        do; stval=enval; enval=ady; stchg=enchg; enchg=lchg;
            tmlcrp=stchg; enent= 0 ; output; end;
    /*Final Event/Censored Record          */
    if last.subjid then do;
        stval=enval; enval=aval; stchg=enchg; enchg= . ;
        tmlcrp= stchg;
        event=(cnsr=0); output; end;

run;
```



SAS Code continued..

```
/*Proc PHREG statement*/  
ods output ParameterEstimates = est1;  
proc phreg data=all;  
    class trt01pn (ref='4');  
    model (stval,enval)*event(0) = trt01pn lbase tmlcrp /  
ties=exact risklimits;  
    output out= dfb1 dfbeta = dt1-dt5; /* DFBETA estimates*/  
run;
```



SAS Code continued..

```

/*****
**      Input data for programing statement approach      **
*****/
/* Maximum number of measurements per patient */
proc sql;
create table a1 as select (count(subjid)) as m1 from adep group by
subjid;
select max(m1) into: number from a1;
quit;
proc sort data=adep;
    by subjid trt01pn ady;

run;
/*create 3 datasets: CHG: change from baseline hsCRP, DAY:
analysis relative day, BASE: baseline hsCRP information */
proc transpose data=adep out=chg prefix=chg;
    by subjid trt01pn paramcd;
    var lchg;

run;
proc transpose data=adep out=day prefix=day;
    by subjid trt01pn paramcd;
    var ady;

run;
```



SAS Code continued..

```
proc sql;
create table base as select distinct subjid,trt01pn,aval,cnsr,paramcd,
armgr2,qmitgr4,lbase from adep order by subjid, trt01pn;
quit;
/*Merge CHG,DAY and BASE datasets*/
data all;
    merge chg day base;
    by subjid trt01pn ;

run;
/*Proc PHREG step*/
proc phreg data=all;
    class trt01pn (ref='4');
    model aval*cnsr(1) = trt01pn lbase tmlcrp / ties=exact
risklimits;
    tmlcrp= 0 ;
    array day(*) day1 - day%left(&number);
    array chg(*) chg1 - chg%left(&number);
    do j=1 to &number;
    if aval > day[j] and day[j] ne . then tmlcrp=chg[j];
    end;

run;
```



Thank You



Derive Value From Excellence ...

Questions ??