REPEATED EVENTS

Dr. Aric LaBarr

Institute for Advanced Analytics

INTRODUCTION

Multiple Events

- Previously discussed how to analyze:
  - Time to **single** event
  - Time to **one** of many events

- What if we extended this again to the possibility of multiple occurrences of a single event?
- **Repeated events**, like competing risks, is a particular type of multi-state analysis that builds upon the previous things we have learned.

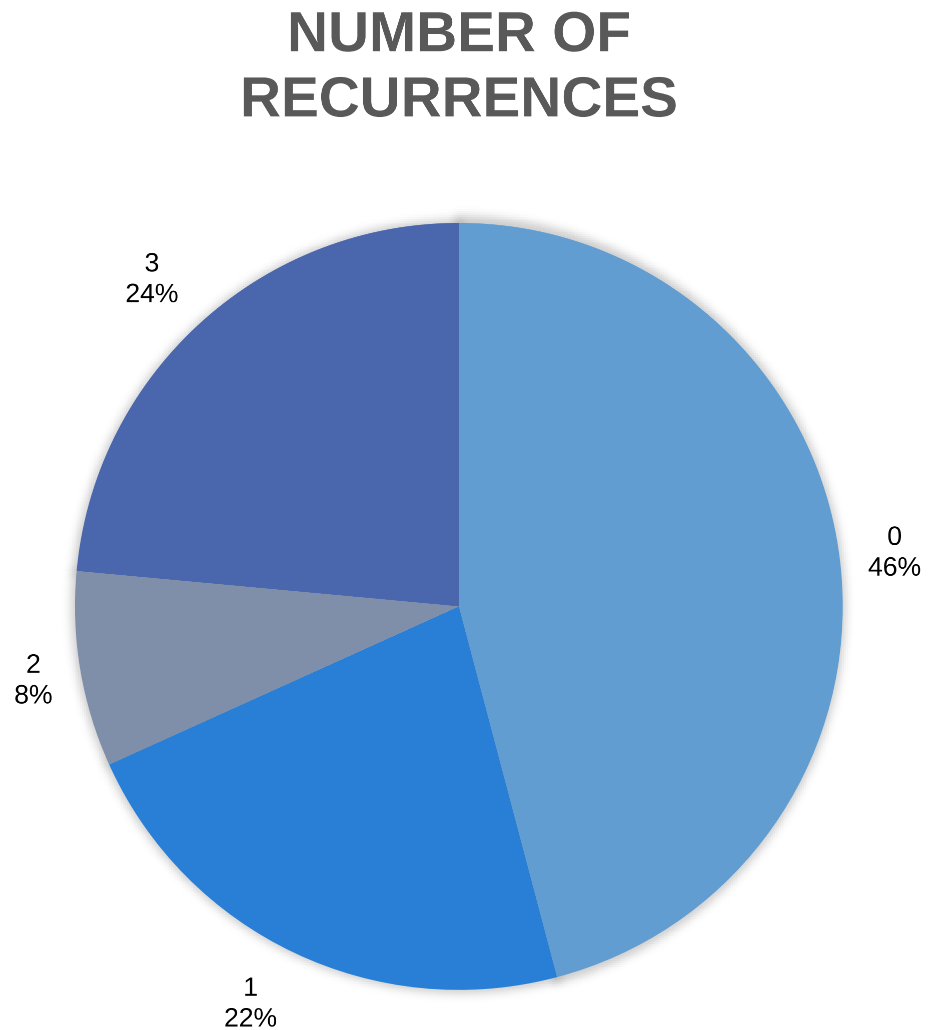
Independent Observations

- For some situations/data, it may be informative to consider the event of interest happening more than once:
  - Repairs for a car/machine/part
  - Hospital readmissions
  - Defaults on loans

- Most techniques are the same, however, the independence of observations no longer applies since an observation can potentially contribute multiple events.

Bladder Tumors Data Set

- Randomized trial of 85 patients.
- Count of recurrences of bladder tumors.
- Andrews DF, Hertzberg AM (1985)



Bladder Tumors Data Set

- **Start**: Either a 0 or time of previous recurrence (in months)
- **Stop**: Current recurrence time (or time of censoring)
- **Event**: Tumor recurrence during the observed **start**, **stop** time period (in months)
- **ID**: Patient ID
- **rx**: placebo (1) or treatment (2) group
- **number**: number of tumors initially present (truncated at 8)
- **size**: diameter (cm) of largest initial tumor
- **enum**: # of previous times with tumors (up to max of 4)

INTENSITY PROCESS

Approaches for Repeated Events

- New modeling choices with repeated events.
- Each modeling choice will end up answering a different question.

- How to treat the repeated events?
  - Order of events important?
  - Additional events indicative of a more serious issue?
- What is the time scale?
  - Time to the event?
  - Time between events?

Intensity Process

- In repeated events, the hazard function is called the **intensity process**:

$$h_i(t) = y_i(t)h_0(t)e^{\beta_1 x_{i,1}+\cdots+\beta_k x_{i,k}}$$

  where $y_i(t)$ is an indicator for being in the risk set at time *t*
- All previous models implied $y_i(t) = 1$ for an observation's entire tenure in the data set.

Intensity Process

- In repeated events, the hazard function is called the **intensity process**:

$$h_i(t) = y_i(t)h_0(t)e^{\beta_1 x_{i,1}+\cdots+\beta_k x_{i,k}}$$

  where $y_i(t)$ is an indicator for being in the risk set at time *t*
- How you answer the previous questions determines the value of $y_i(t)$.
  - Who is in the risk set?
  - When are they in the risk set?
  - Which risk set are they in?

Example Patients

ID	rx	number	size	start	stop	event	enum
5	1	4	1	0	6	1	1
5	1	4	1	6	10	0	2
13	1	3	1	0	3	1	1
13	1	3	1	3	9	1	2
13	1	3	1	9	21	1	3
13	1	3	1	21	23	0	4
16	1	1	2	0	26	0	1
41	1	3	1	0	35	1	1
41	1	3	1	35	51	0	2

MODELS FOR REPEATED EVENTS

Independence Model

Independence Model

- Easiest approach is modeling the recurrences as separate, independent events.
- Assumes that all recurrences are identical – the risk of the event is the same regardless of previous events.
- Only care about the overall effect, ignoring the order or type of recurrence.

Independence Model – Risk Set

- Each observation has time intervals of (start, 1st], (1st, 2nd], …, (kth,stop]

ID	start	stop	event	enum
5	0	6	1	1
5	6	10	0	2
13	0	3	1	1
13	3	9	1	2
13	9	21	1	3
13	21	23	0	4
16	0	26	0	1
41	0	35	1	1
41	35	51	0	2

Accounting for Dependence

- Easiest approach is modeling the recurrences as separate, **independent events**.
- But they aren't! Right?

- 2 Approaches:
  1. Time-Dependent Variables
  2. Robust Standard Errors

Accounting for Dependence

- Easiest approach is modeling the recurrences as separate, **independent events**.
- But they aren’t! Right?

- 2 Approaches:
  1. Time-Dependent Variables → Counting Process Data Structure
  2. Robust Standard Errors

Counting Process Example

- Create a “new” person starting after time = 5 who is the *exact same* as Person 1, but with new *x* value:

Person	Start	Stop	x	Event
1	0	5	3	0
1	5	9	7	1

- We observe this “new” person until either *x* changes again or their tenure ends (whichever comes first).

- This is **exactly** how we structured the independence model risk set!

Independence Model – Risk Set

- Each observation has time intervals of (start, 1^{st}], (1^{st}, 2^{nd}], …, (k^{th},stop]

ID	start	stop	event	enum
5	0	6	1	1
5	6	10	0	2
13	0	3	1	1
13	3	9	1	2
13	9	21	1	3
13	21	23	0	4
16	0	26	0	1
41	0	35	1	1
41	35	51	0	2

Time dependent variable!

Independence Model – Risk Set

- Each observation has time intervals of (start, 1st], (1st, 2nd], …, (kth,stop]

ID	start	stop	event	enum
5	0	6	1	1
5	6	10	0	2
13	0	3	1	1
13	3	9	1	2
13	9	21	1	3
13	21	23	0	4
16	0	26	0	1
41	0	35	1	1
41	35	51	0	2

These could be thought of as "different" people!

Independence! Maybe?

Independence Model – SAS

```
proc phreg data=Survival.Bladder;
    model (start, stop)*event(0) = rx number size enum /
                                   ties=efron;
run;
```

Independence Model – SAS

The PHREG Procedure

Model Information	
Data Set	SURVIVAL.BLADDER
Dependent Variable	start
Dependent Variable	stop
Censoring Variable	event
Censoring Value(s)	0
Ties Handling	EFRON

Number of Observations Read	178
Number of Observations Used	178

Summary of the Number of Event and Censored Values			
Total	**Event**	**Censored**	**Percent Censored**
178	112	66	37.08

Independence Model – SAS

Convergence Status
Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics		
Criterion	Without Covariates	With Covariates
-2 LOG L	917.479	873.585
AIC	917.479	881.585
SBC	917.479	892.459

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	43.8935	4	<.0001
Score	50.6853	4	<.0001
Wald	45.6483	4	<.0001

Independence Model – SAS

Analysis of Maximum Likelihood Estimates						
Parameter	**DF**	**Parameter Estimate**	**Standard Error**	**Chi-Square**	**Pr > ChiSq**	**Hazard Ratio**
rx	1	-0.30125	0.20440	2.1722	0.1405	0.740
number	1	0.14193	0.04949	8.2228	0.0041	1.152
size	1	-0.01586	0.06926	0.0524	0.8189	0.984
enum	1	0.53604	0.10192	27.6638	<.0001	1.709

Independence Model – R

```
bladder.td <- coxph(Surv(start, stop, event == 1) ~ rx + number +
                    size + enum, data = bladder)

summary(bladder.td)
```

Independence Model – R

```
## Call:
## coxph(formula = Surv(start, stop, event == 1) ~ rx + number +
##     size + enum, data = bladder)
##
##   n= 178, number of events= 112
##
##            coef exp(coef) se(coef)      z Pr(>|z|)
## rx     -0.30125   0.73989  0.20440 -1.474  0.14052
## number  0.14193   1.15249  0.04949  2.868  0.00414 **
## size   -0.01586   0.98427  0.06926 -0.229  0.81892
## enum    0.53604   1.70922  0.10192  5.260 1.44e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Independence Model – R

```
##          exp(coef) exp(-coef) lower .95 upper .95
## rx          0.7399     1.3516    0.4957     1.104
## number      1.1525     0.8677    1.0459     1.270
## size        0.9843     1.0160    0.8593     1.127
## enum        1.7092     0.5851    1.3997     2.087
##
## Concordance= 0.673  (se = 0.03 )
## Likelihood ratio test= 43.89  on 4 df,   p=7e-09
## Wald test            = 45.65  on 4 df,   p=3e-09
## Score (logrank) test = 50.69  on 4 df,   p=3e-10
```

Accounting for Dependence

- Easiest approach is modeling the recurrences as separate, **independent events**.
- But they aren't! Right?

- 2 Approaches:
  1. Time-Dependent Variables
  2. Robust Standard Errors → Still possible correlation between observations that can not be explained away with time-dependent variables.

Independence Model – SAS

```
proc phreg data=Survival.Bladder covs(aggregate) covm;
   model (start, stop)*event(0) = rx number size enum /
                                ties=efron;
   id id;
run;
```

Independence Model – SAS

Convergence Status
Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics		
Criterion	**Without Covariates**	**With Covariates**
-2 LOG L	917.479	873.585
AIC	917.479	881.585
SBC	917.479	892.459

Testing Global Null Hypothesis: BETA=0			
Test	**Chi-Square**	**DF**	**Pr > ChiSq**
Likelihood Ratio	43.8935	4	<.0001
Score (Model-Based)	50.6853	4	<.0001
Score (Sandwich)	21.8793	4	0.0002
Wald (Model-Based)	45.6483	4	<.0001
Wald (Sandwich)	41.4133	4	<.0001

Independence Model – SAS

Analysis of Maximum Likelihood Estimates with Model-Based Variance Estimate						
Parameter	**DF**	**Parameter Estimate**	**Standard Error**	**Chi-Square**	**Pr > ChiSq**	**Hazard Ratio**
rx	1	-0.30125	0.20440	2.1722	0.1405	0.740
number	1	0.14193	0.04949	8.2228	0.0041	1.152
size	1	-0.01586	0.06926	0.0524	0.8189	0.984
enum	1	0.53604	0.10192	27.6638	<.0001	1.709

Analysis of Maximum Likelihood Estimates with Sandwich Variance Estimate							
Parameter	**DF**	**Parameter Estimate**	**Standard Error**	**StdErr Ratio**	**Chi-Square**	**Pr > ChiSq**	**Hazard Ratio**
rx	1	-0.30125	0.21277	1.041	2.0046	0.1568	0.740
number	1	0.14193	0.05321	1.075	7.1154	0.0076	1.152
size	1	-0.01586	0.06175	0.892	0.0659	0.7973	0.984
enum	1	0.53604	0.10516	1.032	25.9816	<.0001	1.709

Independence Model – R

```
bladder.rse <- coxph(Surv(start, stop, event == 1) ~ rx + number +
                     size + enum + cluster(id), data = bladder)

summary(bladder.rse)
```

Independence Model – R

```
## Call:
## coxph(formula = Surv(start, stop, event == 1) ~ rx + number +
##     size + enum + cluster(id), data = bladder)
##
##   n= 178, number of events= 112
##
##            coef exp(coef) se(coef) robust se      z Pr(>|z|)
## rx     -0.30125   0.73989  0.20440   0.21277 -1.416  0.15682
## number  0.14193   1.15249  0.04949   0.05321  2.667  0.00764 **
## size   -0.01586   0.98427  0.06926   0.06175 -0.257  0.79734
## enum    0.53604   1.70922  0.10192   0.10516  5.097 3.45e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Independence Model – R

```
##        exp(coef) exp(-coef) lower .95 upper .95
## rx        0.7399     1.3516    0.4876     1.123
## number    1.1525     0.8677    1.0384     1.279
## size      0.9843     1.0160    0.8721     1.111
## enum      1.7092     0.5851    1.3909     2.100
##
## Concordance= 0.673  (se = 0.031 )
## Likelihood ratio test= 43.89  on 4 df,   p=7e-09
## Wald test            = 41.41  on 4 df,   p=2e-08
## Score (logrank) test = 50.69  on 4 df,   p=3e-10,   Robust = 21.88  p=2e-04
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

MODELS FOR REPEATED EVENTS

Conditional Model

Stratified Models

- Unlike the independence model, we can preserve the ordering of events if it's important.
- In the **conditional model**, we stratify on the number of events, so only those who have had a previous event are in the risk set for the next one.
  - Example: Not in the risk set for the 3rd event until you have had the 2nd event.

- Each recurrence is a separate stratum (imagine own model) with its **own baseline hazard** – no estimates/inferences on the number of recurrences.

Conditional Model – Risk Set

- Risk set for 1st event:

ID	start	stop	event	enum
5	0	6	1	1
13	0	3	1	1
16	0	26	0	1
41	0	35	1	1

- Risk set for 2nd event:

ID	start	stop	event	enum
5	6	10	0	2
13	3	9	1	2
41	35	51	0	2

Conditional Model – SAS

```
proc phreg data=Survival.Bladder;
   model (start, stop)*event(0) = rx number size /
                                  ties=efron;
   strata enum;
run;
```

Code above assumes variable effects (coefficients) same across each stratum.

Conditional Model – SAS

Summary of the Number of Event and Censored Values					
Stratum	**enum**	**Total**	**Event**	**Censored**	**Percent Censored**
1	1	85	47	38	44.71
2	2	46	29	17	36.96
3	3	27	22	5	18.52
4	4	20	14	6	30.00
Total		178	112	66	37.08

Convergence Status
Convergence criterion (GCONV=1E-8) satisfied.

Conditional Model – SAS

Analysis of Maximum Likelihood Estimates						
Parameter	**DF**	**Parameter Estimate**	**Standard Error**	**Chi-Square**	**Pr > ChiSq**	**Hazard Ratio**
rx	1	-0.33349	0.21617	2.3800	0.1229	0.716
number	1	0.11962	0.05334	5.0294	0.0249	1.127
size	1	-0.00849	0.07276	0.0136	0.9071	0.992

Effect of k^{th} event since *entry time*

Conditional Model – R

```
bladder.con <- coxph(Surv(start, stop, event == 1) ~ rx + number +
                     size + strata(enum), data = bladder)

summary(bladder.con)
```

Conditional Model – R

```
## coxph(formula = Surv(start, stop, event == 1) ~ rx + number +
##     size + strata(enum), data = bladder)
##
##   n= 178, number of events= 112
##
##              coef exp(coef)  se(coef)      z Pr(>|z|)
## rx      -0.333489  0.716420  0.216168 -1.543   0.1229
## number   0.119617  1.127065  0.053338  2.243   0.0249 *
## size    -0.008495  0.991541  0.072762 -0.117   0.9071
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##         exp(coef) exp(-coef) lower .95 upper .95
## rx         0.7164     1.3958    0.4690     1.094
## number     1.1271     0.8873    1.0152     1.251
## size       0.9915     1.0085    0.8598     1.144
##
## Concordance= 0.616  (se = 0.038 )
## Likelihood ratio test= 6.51  on 3 df,   p=0.09
## Wald test            = 6.85  on 3 df,   p=0.08
## Score (logrank) test = 6.91  on 3 df,   p=0.07
```

Conditional Model – SAS

```
proc phreg data=Survival.Bladder;
   model (start, stop)*event(0) = rx1-rx4
                                  number1-number4
                                  size1-size4 /
                                  ties=efron;
   rx1 = rx * (enum=1);
   rx2 = rx * (enum=2);
   rx3 = rx * (enum=3);
   rx4 = rx * (enum=4);
   number1 = number * (enum=1);
   number2 = number * (enum=2);
   number3 = number * (enum=3);
   number4 = number * (enum=4);
   size1 = size * (enum=1);
   size2 = size * (enum=2);
   size3 = size * (enum=3);
   size4 = size * (enum=4);
   strata enum;
run;
```

Code assumes variable effects (coefficients) differ across each stratum.

Conditional Model – SAS

Analysis of Maximum Likelihood Estimates						
Parameter	**DF**	**Parameter Estimate**	**Standard Error**	**Chi-Square**	**Pr > ChiSq**	**Hazard Ratio**
rx1	1	-0.52593	0.31583	2.7730	0.0959	0.591
rx2	1	-0.50384	0.40617	1.5388	0.2148	0.604
rx3	1	0.14066	0.67306	0.0437	0.8345	1.151
rx4	1	0.05033	0.79171	0.0040	0.9493	1.052
number1	1	0.23824	0.07588	9.8579	0.0017	1.269
number2	1	-0.02464	0.08987	0.0752	0.7840	0.976
number3	1	0.04966	0.18532	0.0718	0.7887	1.051
number4	1	0.20428	0.24204	0.7123	0.3987	1.227
size1	1	0.06963	0.10156	0.4700	0.4930	1.072
size2	1	-0.16072	0.12247	1.7222	0.1894	0.852
size3	1	0.16810	0.26904	0.3904	0.5321	1.183
size4	1	0.00910	0.33893	0.0007	0.9786	1.009

Conditional Model – R

```
bladder.con2 <- coxph(Surv(start, stop, event == 1) ~
                     rx:strata(enum) + number:strata(enum) +
                     size:strata(enum), data = bladder)

summary(bladder.con2)
```

Conditional Model – R

```
## coxph(formula = Surv(start, stop, event == 1) ~ rx:strata(enum) +
##     number:strata(enum) + size:strata(enum), data = bladder)
##
##   n= 178, number of events= 112
##
##                                coef exp(coef)  se(coef)      z Pr(>|z|)
## rx:strata(enum)enum=1     -0.525984  0.590973  0.315826 -1.665   0.0958 .
## rx:strata(enum)enum=2     -0.503837  0.604208  0.406167 -1.240   0.2148
## rx:strata(enum)enum=3      0.140657  1.151029  0.673063  0.209   0.8345
## rx:strata(enum)enum=4      0.050331  1.051619  0.791710  0.064   0.9493
## strata(enum)enum=1:number  0.238180  1.268937  0.075885  3.139   0.0017 **
## strata(enum)enum=2:number -0.024641  0.975660  0.089873 -0.274   0.7840
## strata(enum)enum=3:number  0.049661  1.050915  0.185323  0.268   0.7887
## strata(enum)enum=4:number  0.204277  1.226637  0.242040  0.844   0.3987
## strata(enum)enum=1:size    0.069613  1.072094  0.101559  0.685   0.4931
## strata(enum)enum=2:size   -0.160716  0.851534  0.122467 -1.312   0.1894
## strata(enum)enum=3:size    0.168099  1.183053  0.269040  0.625   0.5321
## strata(enum)enum=4:size    0.009095  1.009137  0.338928  0.027   0.9786
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Gap Time

- Notice that in the conditional model, each event's **start** time is determined by the previous event's **stop** time!
- An alternative time scale is the **gap time**, where we instead choose to model the time *since last event*.
- In gap-time models, time is reset to 0 after each event, so the time until the prior event has no bearing on the current event's risk set.

Gap Time – Risk Set

- Risk set for 1st event:

ID	start	stop	event	enum
5	0	6	1	1
13	0	3	1	1
16	0	26	0	1
41	0	35	1	1

- Risk set for 2nd event:

ID	start	stop	event	enum
5	0	4	0	2
13	0	6	1	2
41	0	16	0	2

Gap Time – SAS

```
data bladder_gap;
        set Survival.Bladder;
        gaptime = stop - start;
run;

proc phreg data=bladder_gap;
    model gaptime*event(0) = rx number size / ties=efron;
    strata enum;
run;
```

Code above assumes variable effects (coefficients) same across each stratum.
Could easily extend to different variable effects as before.

Gap Time – SAS

Analysis of Maximum Likelihood Estimates						
Parameter	**DF**	**Parameter Estimate**	**Standard Error**	**Chi-Square**	**Pr > ChiSq**	**Hazard Ratio**
rx	1	-0.27900	0.20735	1.8106	0.1784	0.757
number	1	0.15805	0.05194	9.2582	0.0023	1.171
size	1	0.00742	0.07002	0.0112	0.9157	1.007

Effect of k^{th} event since *time of previous event*

Gap Time – R

```
bladder.gap <- coxph(Surv(time = (stop - start), event == 1) ~ rx +
                     number + size + strata(enum), data = bladder)

summary(bladder.gap)
```

Gap Time – R

```
## coxph(formula = Surv(time = (stop - start), event == 1) ~ rx +
##     number + size + strata(enum), data = bladder)
##
##   n= 178, number of events= 112
##
##             coef exp(coef)  se(coef)      z Pr(>|z|)
## rx     -0.279005  0.756536  0.207348 -1.346  0.17844
## number  0.158046  1.171220  0.051942  3.043  0.00234 **
## size    0.007415  1.007443  0.070023  0.106  0.91567
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##        exp(coef) exp(-coef) lower .95 upper .95
## rx        0.7565     1.3218    0.5039     1.136
## number    1.1712     0.8538    1.0579     1.297
## size      1.0074     0.9926    0.8782     1.156
##
## Concordance= 0.596  (se = 0.035 )
## Likelihood ratio test= 9.33  on 3 df,   p=0.03
## Wald test            = 10.11  on 3 df,   p=0.02
## Score (logrank) test = 10.27  on 3 df,   p=0.02
```