

GSERM - St. Gallen 2019

Survival Model Extensions

June 21, 2019 (morning session)

Standard models (e.g.):

$$h(T_i|\mathbf{X}_i, \beta) = \frac{f(T_i|\mathbf{X}_i, \beta)}{S(T_i|\mathbf{X}_i, \beta)}$$

assume:

$$\int_0^{\infty} f(t) dt = 1 \quad \forall i.$$

All observations will (eventually) experience the event of interest.

Assume (unobserved):

$$Y_i = \begin{cases} 1 & \text{for observations that will eventually fail,} \\ 0 & \text{for those that will not.} \end{cases}$$

For observations with $Y = 1$:

$$\begin{aligned} f(T_i | \mathbf{X}_i, \beta, Y_i = 1) &= g(T | \mathbf{X}_i, \beta) \\ F(T_i | \mathbf{X}_i, \beta, Y_i = 1) &= G(T | \mathbf{X}_i, \beta) \end{aligned}$$

For observations with $Y = 0$, $f(T)$ and $F(T)$ are undefined.

Mixture Cure Model (continued)

Define:

$$\Pr(Y_i = 1) = \delta_i.$$

Overall survival is then just:

$$S_i(T) = (1 - \delta_i) + \delta_i[1 - G_i(t)]$$

Mixture Cure Model: Likelihood

Then for $C_i = 1$:

$$\begin{aligned} L_i | C_i = 1 &= \Pr(Y_i = 1) \Pr(T_i = t | Y_i = 1, \mathbf{X}_i, \beta) \\ &= \delta_i g(T_i | \mathbf{X}_i, \beta) \end{aligned}$$

For $C_i = 0$:

$$\begin{aligned} L_i | C_i = 0 &= \Pr(Y_i = 0) + \Pr(Y_i = 1) \Pr(T_i > t_i | Y_i = 1, \mathbf{X}_i, \beta) \\ &= (1 - \delta_i) + \delta_i [1 - G(T_i | \mathbf{X}_i, \beta)] \end{aligned}$$

Mixture Cure Model: Likelihood

Implies:

$$\mathbf{L} = \prod_{i=1}^N [\delta_i g(T_i | \mathbf{X}_i, \beta)]^{C_i} \{(1 - \delta_i) + \delta_i [1 - G(T_i | \mathbf{X}_i, \beta)]\}^{(1 - C_i)}$$

and:

$$\begin{aligned} \ln \mathbf{L} &= \sum_{i=1}^N C_i \{ \ln(\delta_i) + \ln [g(T_i | \mathbf{X}_i, \beta)] \} \\ &\quad + (1 - C_i) \ln \{ (1 - \delta_i) + \delta_i [1 - G(T_i | \mathbf{X}_i, \beta)] \} \end{aligned}$$

Mixture Cure Model: Specification

Typically:

$$\delta_i = \frac{\exp(\mathbf{Z}_i\gamma)}{1 + \exp(\mathbf{Z}_i\gamma)}$$

or:

$$\delta_i = \Phi(\mathbf{Z}_i\gamma).$$

Identified even if $\mathbf{Z} \equiv \mathbf{X}$.

Non-Mixture Cure Model (e.g. Sposto 2002)

N_i = number of pre-cancerous cell clusters, with:

$$N_i \sim \text{Poisson}(\lambda).$$

$\Pr(\text{Cure})$ is:

$$\pi_i = \Pr(N_i = 0).$$

Time to cancer onset for cluster j of observation i is:

$$Z_{ij} \sim F(t), \quad j = \{1, 2, \dots, N_i\}.$$

Non-Mixture Cure Model (continued)

Survival to first onset:

$$S(t) = \pi^{F(t)}$$

with hazard function:

$$h(t) = -\ln(\pi)f(t)$$

which reflects the fact that $\int_0^\infty h(t)dt = -\ln(\pi)$.

Non-Mixture Cure Model (continued)

Rewritten $S(t)$:

$$S(t) = \exp[\ln(\pi)F(t)].$$

Assuming:

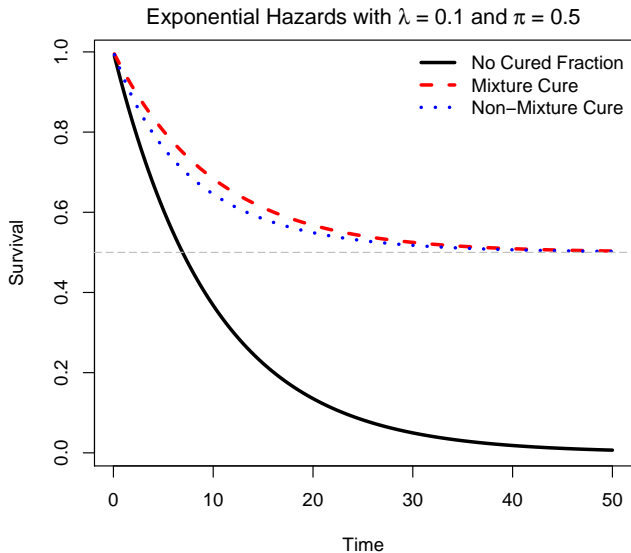
$$\pi_i = \exp[-\exp(\mathbf{X}_i\beta)]$$

we get:

$$S(t) = \exp\{[-\exp(\mathbf{X}_i\beta)]F(t)\}.$$

which is the Cox.

Mixture vs. Non-Mixture Models



Discrete-Time Cure Models

- Parametric / Cox \longrightarrow Poisson
- Mixture Cure Model \longrightarrow Zero-Inflated Poisson
- Non-Mixture Cure Model \longrightarrow “Hurdle” Poisson

R

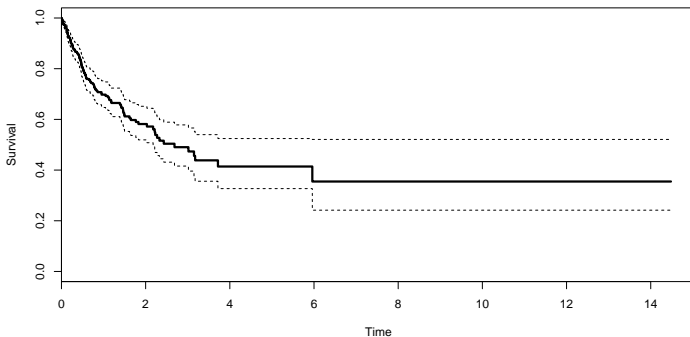
- `smcure` (semiparametric mixture models via EM)
- `semicure` (same; old)
- `nltm` (various; see Tsodikov 2003)
- CR, NPHMC (power analysis for cure models)

Stata

- `strsmix` and `strsnmix` (general parametric mixture & non-mixture cure models)
- `cureregr` (an old version)
- `lncure` (log-normal cure model)
- `spsurv` (discrete-time cure model)
- `zip` / `zinb` (discrete-time kludge)

A Simulated Example

```
> set.seed=7222009  
> X<-rnorm(500)  
> Z<-rbinom(500,1,0.5)  
> T<-rweibull(500,shape=1.2,scale=1/(exp(0.5+1*X)))  
> C<-rbinom(500,1,(0.4-0.3*Z))  
> S<-Surv(T,C)
```



```
> coxph(S~X)
```

```
Call:
```

```
coxph(formula = S ~ X)
```

	coef	exp(coef)	se(coef)	z	p
X	1.05	2.85	0.124	8.44	0

```
Likelihood ratio test=77.7 on 1 df, p=0 n= 500, number of events= 130
```

```
> coxph(S~X+Z)
```

```
Call:
```

```
coxph(formula = S ~ X + Z)
```

	coef	exp(coef)	se(coef)	z	p
X	1.08	2.956	0.122	8.9	0.0e+00
Z	-1.59	0.204	0.230	-6.9	5.4e-12

```
Likelihood ratio test=140 on 2 df, p=0 n= 500, number of events= 130
```

```
> cure.fit<-smcure(S~X,cureform=~Z,data=data.cure,model="ph")
```

Program is running..be patient... done.

Call:

```
smcure(formula = S ~ X, cureform = ~Z, data = data.cure, model = "ph")
```

Cure probability model:

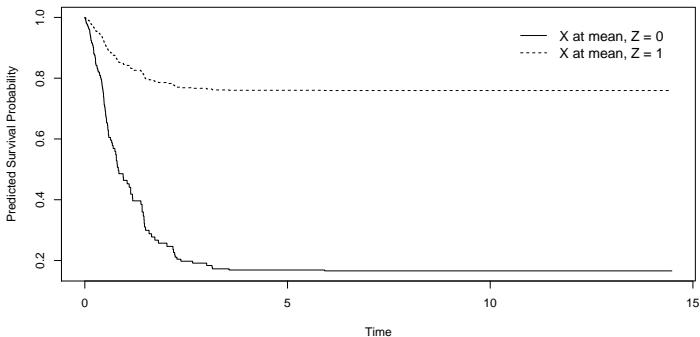
	Estimate	Std.Error	Z value	Pr(> Z)
(Intercept)	1.6	0.39	4.1	3.4e-05
Z	-2.8	0.41	-6.7	2.5e-11

Failure time distribution model:

	Estimate	Std.Error	Z value	Pr(> Z)
X	1.1	0.14	8.1	6.7e-16

An Interesting Plot

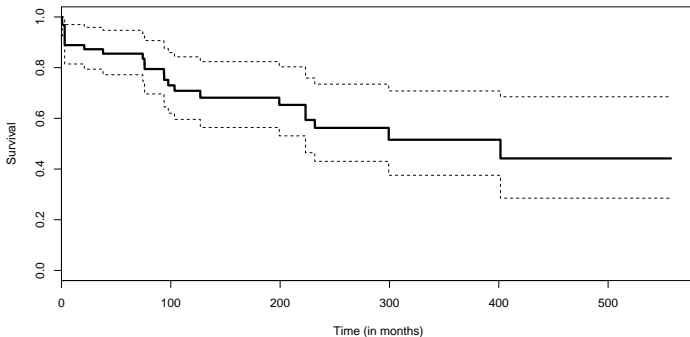
```
> cure.hat<-predictsmcure(cure.fit,c(rep(mean(X),times=2)),  
                           c(0,1),model="ph")  
  
> cure.pic<-plotpredictsmcure(cure.hat,type="S",model="ph")
```



An Example: Ceasefire Durability

Data are a subset used in Fortna (2004) (full data are [here](#)).

- $N = 63$
- Non-time-varying



Ceasefires: Cox Model

```
> CF.cox<-coxph(CF.S~tie+imposed+lndeaths+contig+onedem+twodem,  
                data=CF,method="efron")
```

```
> CF.cox
```

Call:

```
coxph(formula = CF.S ~ tie + imposed + lndeaths + contig + onedem +  
      twodem, data = CF, method = "efron")
```

	coef	exp(coef)	se(coef)	z	p
tie	1.845	6.327	0.557	3.314	0.00092
imposed	0.210	1.233	0.594	0.353	0.72000
lndeaths	-0.135	0.874	0.193	-0.699	0.48000
contigyes	2.898	18.143	0.948	3.058	0.00220
onedem	3.423	30.648	1.144	2.991	0.00280
twodem	-0.723	0.485	1.209	-0.598	0.55000

```
Likelihood ratio test=36.8 on 6 df, p=0.00000197 n= 63, number of events= 23
```

(hours of fiddling...)

A Typical Result

```
> CF.cure1.fit<-smcure(CF.S~tie+lndeaths+imposed,  
  cureform=~contig,data=CF,model="ph",  
  link="logit",emmax=500)
```

Program is running..be patient... done.

Call:

```
smcure(formula = CF.S ~ tie + lndeaths + imposed, cureform = ~contig,  
  data = CF, model = "ph", link = "logit", emmax = 500)
```

Cure probability model:

	Estimate	Std.Error	Z value	Pr(> Z)
(Intercept)	-3.4	12.4	-0.27	0.79
contig	2.1	7.4	0.28	0.78

Failure time distribution model:

	Estimate	Std.Error	Z value	Pr(> Z)
tie	2.05	4.06	0.50	0.61
lndeaths	-0.37	0.34	-1.10	0.27
imposed	0.97	2.40	0.41	0.68

There were 50 or more warnings (use warnings() to see the first 50)

From Svulik (2008)

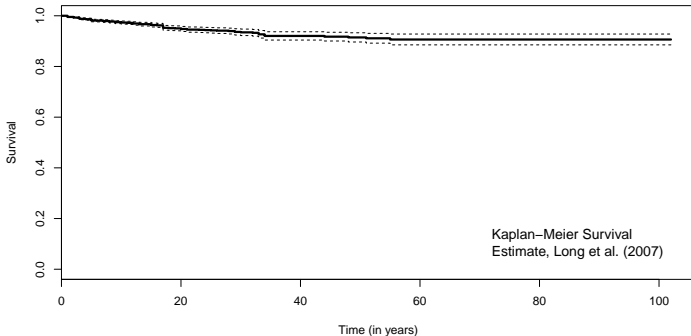
Consolidation status model^b

<i>GDP per capita</i>	2.121*** (0.586)	—	2.045*** (0.555)	2.121*** (0.586)
<i>GDP growth</i>	-0.014 (0.227)	—	-0.048 (0.246)	-0.014 (0.227)
<i>Military (vs. Not independent)</i>	-4.061** (1.895)	—	-3.985** (1.857)	-4.061** (1.895)
<i>Civilian (vs. Not independent)</i>	-0.421 (1.097)	—	-0.549 (1.067)	-0.421 (1.097)
<i>Monarchy (vs. Not independent)</i>	-20.158 (2888.609)	—	-15.844 (680.185)	-13.965 (891.870)
<i>Parliamentary (vs. Mixed)</i>	2.231 (2.230)	—	2.290 (2.326)	2.231 (2.230)
<i>Presidential (vs. Mixed)</i>	-8.310** (3.958)	—	-8.186** (4.035)	-8.310** (3.958)
<i>Intercept</i>	-6.144** (2.646)	—	-5.920** (2.644)	-6.145** (2.647)

Another Example: Peace Duration

Long, Nordstrom and Baek (2007 *JOP*)

- Peace duration among allies
- Time-varying dyadic data, 1816-2001 ($NT = 57,819$)



Cox Model (replicating LNB)

```
> LNB.cox<-coxph(LNB.S~relcap+major+jdem+border+wartime+s_wt_glo+  
  medarb+noagg+arbcom+organ+milinst+cluster(dyad),  
  data=LNB,method="breslow")
```

```
> LNB.cox
```

```
Call:
```

```
coxph(formula = LNB.S ~ relcap + major + jdem + border + wartime +  
  s_wt_glo + medarb + noagg + arbcom + organ + milinst + cluster(dyad),  
  data = LNB, method = "breslow")
```

	coef	exp(coef)	se(coef)	robust se	z	p
relcap	-1.431	0.239	0.614	0.683	-2.096	0.036000
major	1.137	3.118	0.241	0.280	4.064	0.000048
jdem	-0.987	0.373	0.367	0.380	-2.600	0.009300
border	1.931	6.897	0.190	0.206	9.378	0.000000
wartime	-0.359	0.699	0.367	0.467	-0.768	0.440000
s_wt_glo	-0.284	0.752	0.332	0.355	-0.802	0.420000
medarb	-0.367	0.693	0.285	0.306	-1.202	0.230000
noagg	-0.463	0.630	0.126	0.152	-3.051	0.002300
arbcom	1.306	3.690	0.325	0.316	4.133	0.000036
organ	0.353	1.423	0.280	0.285	1.236	0.220000
milinst	-0.373	0.689	0.187	0.177	-2.101	0.036000

(hours of fiddling...)

```
> LNB.cure<-smcure(LNB.altS~relcap+major+jdem+border+wartime+s_wt_glo+  
  medarb+noagg+arbcom+organ+milinst,  
  cureform=~border,model="ph",data=LNB)
```

Program is running..be patient...

Cure Models (Stata Remix)

```
. stset count1, id(episode) f(buofmzmid==1)
. gen h0=0
. strsmix major jdem border wartime, bhazard(h0) distribution(weibull) link(logistic) k1
> (relcap major jdem border wartime s_wt_glo medarb noagg arbcom organ milinst)
```

```

                                Number of obs   =      57819
                                Wald chi2(4)      =       36.82
                                Prob > chi2       =      0.0000
Log likelihood = -793.21263
```

	_t	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
-----+-----							
pi							
major		-7.921296	3.764002	-2.10	0.035	-15.2986	-.5439877
jdem		-.6177566	.7656096	-0.81	0.420	-2.118324	.8828107
border		-1.943181	.3786093	-5.13	0.000	-2.685241	-1.20112
wartime		2.583909	1.051959	2.46	0.014	.5221065	4.645711
_cons		2.659179	.3980719	6.68	0.000	1.878972	3.439385
-----+-----							
ln_lambda							
relcap		-1.408332	.7129111	-1.98	0.048	-2.805613	-.0110523
major		-1.232928	.395653	-3.12	0.002	-2.008394	-.4574626
jdem		-1.69796	.4596442	-3.69	0.000	-2.598846	-.7970736
border		1.224114	.2622007	4.67	0.000	.7102103	1.738018
wartime		.42086	.4072876	1.03	0.301	-.377409	1.219129
s_wt_glo		-.274703	.3579769	-0.77	0.443	-.9763249	.4269188
medarb		-.8221547	.3503126	-2.35	0.019	-1.508755	-.1355545
noagg		-.68365	.1465971	-4.66	0.000	-.970975	-.3963251
arbcom		1.667284	.4562532	3.65	0.000	.7730438	2.561524
organ		.9298395	.3595899	2.59	0.010	.2250563	1.634623
milinst		-.4428979	.2251323	-1.97	0.049	-.8841491	-.0016468
_cons		-2.060399	.7260061	-2.84	0.005	-3.483344	-.6374528
-----+-----							
ln_gamma							
_cons		.0969349	.0733007	1.32	0.186	-.0467319	.2406018
-----+-----							

Cure models...

- ...Powerful
- ...Intuitive
- ...Temperamental
- ...Ask a lot of your data

[Break]

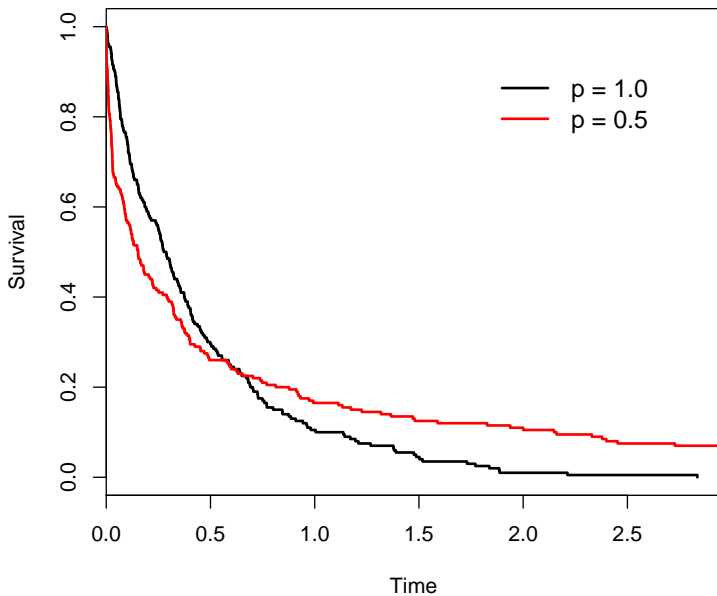
Stratification

- Allow different groups to have different baseline hazards
- Akin to different intercepts, but more flexible.
- *Assumes covariate effects are otherwise identical*
- Uses:
 - Unit/group heterogeneity
 - Nonproportional hazards
 - Simple models for duration dependence

Stratification, Simulated

```
> set.seed=7222009
> Z<-rnorm(200)
> X0<-rep(0,times=200)
> X1<-rep(1,times=200)
> T0<-rweibull(200,shape=1,scale=1/exp(2+0.5*Z))
> T1<-rweibull(200,shape=0.5,scale=1/exp(2+0.5*Z))
> C<-rep(1,times=400)
> X<-append(X0,X1)
> T<-append(T0,T1)
> data<-as.data.frame(cbind(T,C,X,rep(Z,times=2)))
> colnames(data)<-c("T","C","X","Z")
```


Stratified Weibull Hazards



Stratification, Simulated

```
> cox<-coxph(S~Z+X,data=data)
> summary(cox)
Call:
coxph(formula = S ~ Z + X, data = data)

n= 400, number of events= 400

      coef exp(coef) se(coef)      z Pr(>|z|)
Z  0.28286   1.32692  0.05133  5.510 3.58e-08 ***
X -0.22866   0.79560  0.10639 -2.149  0.0316 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      exp(coef) exp(-coef) lower .95 upper .95
Z    1.3269    0.7536    1.1999    1.4674
X    0.7956    1.2569    0.6459    0.9801

Concordance= 0.571  (se = 0.017 )
Rsquare= 0.08  (max possible= 1 )
Likelihood ratio test= 33.25  on 2 df,   p=6.022e-08
Wald test               = 33.02  on 2 df,   p=6.749e-08
Score (logrank) test = 33.07  on 2 df,   p=6.601e-08
```

Stratification, Simulated

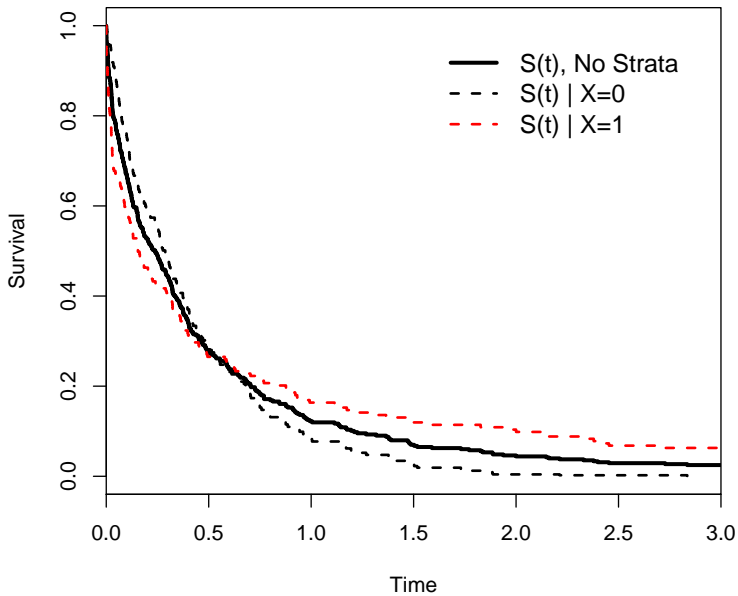
```
> cox.strata<-coxph(S~Z+strata(X),data=data)
> summary(cox.strata)
Call:
coxph(formula = S ~ Z + strata(X), data = data)

      n= 400, number of events= 400

      coef exp(coef) se(coef)      z Pr(>|z|)
Z 0.32140   1.37906  0.05176  6.21  5.3e-10 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

      exp(coef) exp(-coef) lower .95 upper .95
Z      1.379      0.7251      1.246      1.526

Concordance= 0.597 (se = 0.024 )
Rsquare= 0.092 (max possible= 1 )
Likelihood ratio test= 38.69 on 1 df,  p=4.955e-10
Wald test              = 38.56 on 1 df,  p=5.303e-10
Score (logrank) test = 38.62 on 1 df,  p=5.151e-10
```



Stratified Weibull Model

```
> summary(survreg(S~Z+strata(X),data=data,dist="weibull"))
```

Call:

```
survreg(formula = S ~ Z + strata(X), data = data, dist = "weibull")
```

	Value	Std. Error	z	p
(Intercept)	-0.9976	0.0675	-14.781	1.93e-49
Z	-0.4140	0.0577	-7.178	7.06e-13
X=0	0.0152	0.0555	0.274	7.84e-01
X=1	0.6864	0.0543	12.650	1.11e-36

Scale:

```
  X=0  X=1  # Recall: scale = 1 / p  
1.02 1.99
```

Weibull distribution

```
Loglik(model)= -7.8   Loglik(intercept only)= -31.4
```

```
Chisq= 47.36 on 1 degrees of freedom, p= 5.9e-12
```

```
Number of Newton-Raphson Iterations: 6
```

```
n= 400
```

$$h_i(t) = \lambda_i(t)\nu_i$$

- $\nu_i = 1 \approx$ “baseline,”
- $\nu_i > 1 \rightarrow i$ has a greater-than-average hazard,
- $\nu_i < 1 \rightarrow$ the opposite.

Implies:

$$\begin{aligned} S(t|\nu_i) &= \exp \left[- \int_0^t h(t|\nu_i) dt \right] \\ &= \exp \left[- \int_0^t \nu_i h(t) dt \right] \\ &= \exp \left[- \int_0^t h(t) dt \right]^{\nu_i} \\ &= S(t)^{\nu_i} \end{aligned}$$

Typically:

- Assume $\nu_i \sim g(\nu)$, with
- $E(\nu) = 1$ and
- $\text{Var}(\nu) = \theta$

Example: Cox with Frailty

$$\begin{aligned}h_i(t) &= h_0(t)\nu_i\exp(\mathbf{X}_i\beta) \\ &= h_0(t)\exp(\mathbf{X}_i\beta + \alpha_i)\end{aligned}$$

where $\alpha_i = \ln(\nu_i)$.

(Also weibull, log-normal, etc.)

Frailty Distributions: Gamma

$$\begin{aligned}g(\nu) &= \mathcal{G}(\theta, 1/\theta) \\ &= \frac{\nu^{1/\theta-1} \exp\left(\frac{-\nu}{\theta}\right)}{\theta^{(1/\theta)} \Gamma(1/\theta)}\end{aligned}$$

with

$$S_{\theta}(t) = \{1 - \theta \ln[S(t)]\}^{-1/\theta}$$

Frailty Distributions: Inverse-Gaussian

$$\begin{aligned}g(\nu) &= \mathcal{IG}(\theta, 1/\theta) \\&= (2\pi\theta\nu^3)^{-1/2} \exp\left[-\frac{1}{2\theta}\left(\alpha - 2 + \frac{1}{\nu}\right)\right]\end{aligned}$$

with

$$S_{\theta}(t) = \exp\left\{\frac{1}{\theta}\left[1 - (1 - 2\theta \ln\{S(t)\})^{1/2}\right]\right\}$$

An Important Distinction

Individual- (or Unit-) Specific Survival Function:

$$S(t|\nu_i) = S(t)^{\nu_i}$$

Population Average Survival Function:

$$\overline{S(t)} = \int_0^{\infty} S(t|\nu_i)g(\nu)d\nu$$

- Originally: E-M algorithm (e.g. Klein 1992)
- Later: Penalized Likelihood
 - Two-level iterative procedure
 - Intuition: Iterate between fitting $\hat{\beta}|\theta$ for a range of θ s, and searching over the (univariate) marginal likelihood for θ to obtain $\hat{\theta}$
 - Details: Therneau and Grambsch (2000, §9.6)

- Computation...

"...if there are 300 families, each with their own frailty, and four other variables, then the full information matrix has $304^2 = 92,416$ elements. The Cholesky decomposition must be applied to this matrix with each Newton-Raphson iteration."

– Therneau and Grambsch (2000, p. 258)

- Fitting choices (fix θ vs. estimation, etc.)
- Predictions / interpretation (typically assume $\hat{\nu}_i = 1$).

R

- `survival`: Fits a single [frailty](#) term via `frailty.gamma`, `frailty.gaussian`, or `frailty.t` to either Cox or parametric models.
- [coxme](#) (Cox w/Gaussian random effects; see below)
- `frailtypack` (parallel to `frailty` and `coxme`)
- Others (see the [task view](#))

Stata

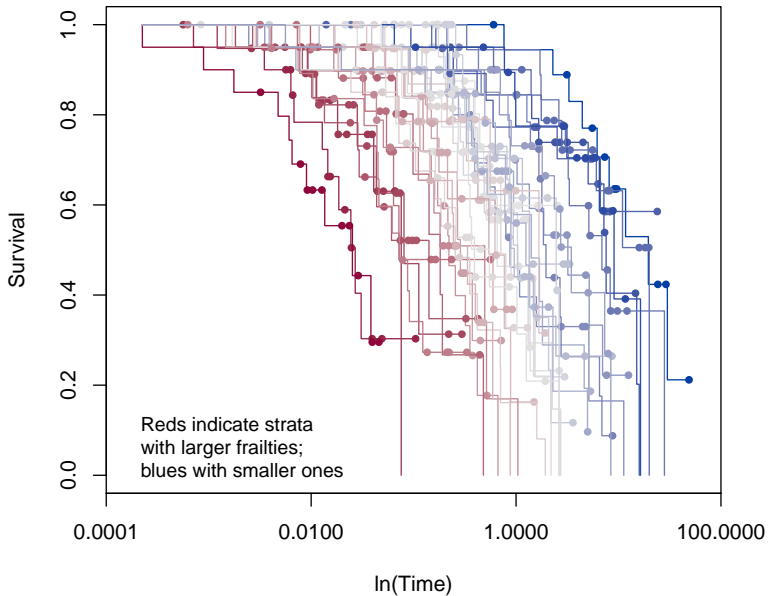
- The option `shared()` introduces one-level gamma-distributed frailties into `stcox`
- `streg` allows unshared or shared frailties (via `frailty()` and `shared()`, respectively) in both gamma and inverse-gaussian flavors in its parametric survival models; see [Guiterrez \(2002\)](#) for a good starting point.

Simulated Example

```
> set.seed(7222009)
> G<-1:40      # "groups"
> F<-rnorm(40)  # frailties
> data<-data.frame(cbind(G,F))
> data<-data[rep(1:nrow(data),each=20),]
> data$X<-rbinom(nrow(data),1,0.5)
> data$T<-rexp(nrow(data),rate=exp(0+1*data$X+(2*data$F)))
> data$C<-rbinom(nrow(data),1,0.5)
> data<-data[order(data$F),]

> S<-Surv(data$T,data$C)
```

K-M Plots By Strata



Cox Fit (No Frailty)

```
> cox.noF<-coxph(S~X,data=data)
> summary(cox.noF)
Call:
coxph(formula = S ~ X, data = data)

      n= 800, number of events= 381

      coef exp(coef) se(coef)      z    Pr(>|z|)
X 0.522      1.685      0.104 5.02 0.00000051 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

      exp(coef) exp(-coef) lower .95 upper .95
X           1.69           0.593       1.37       2.07

Concordance= 0.577  (se = 0.015 )
Rsquare= 0.031  (max possible= 0.996 )
Likelihood ratio test= 25.2  on 1 df,  p=0.000000521
Wald test              = 25.2  on 1 df,  p=0.000000508
Score (logrank) test = 25.8  on 1 df,  p=0.000000382
```

Weibull Fit (No Frailty)

```
> weib.noF<-survreg(S~X,data=data,dist="weib")  
> summary(weib.noF)
```

Call:

```
survreg(formula = S ~ X, data = data, dist = "weib")
```

	Value	Std. Error	z	p
(Intercept)	1.595	0.1450	11.00	3.92e-28
X	-1.031	0.1974	-5.22	1.76e-07
Log(scale)	0.653	0.0383	17.04	3.98e-65

Scale= 1.92

Weibull distribution

Loglik(model)= -581 Loglik(intercept only)= -594

Chisq= 27 on 1 degrees of freedom, p= 0.00000023

Number of Newton-Raphson Iterations: 5

n= 800

Cox Fit With Frailty

```
> cox.F<-coxph(S~X+frailty.gaussian(F),data=data)
> summary(cox.F)
Call:
coxph(formula = S ~ X + frailty.gaussian(F), data = data)
```

```
    n= 800, number of events= 381
```

	coef	se(coef)	se2	Chisq	DF	p
X	1.01	0.112	0.112	81.9	1.0	0
frailty.gaussian(F)				609.0	37.6	0

	exp(coef)	exp(-coef)	lower .95	upper .95
X	2.76	0.363	2.21	3.43

```
Iterations: 7 outer, 47 Newton-Raphson
```

```
Variance of random effect= 1.8
```

```
Degrees of freedom for terms= 1.0 37.6
```

```
Concordance= 0.791 (se = 0.017 )
```

```
Likelihood ratio test= 414 on 38.5 df, p=0
```

Weibull Fit With Frailty

```
> weib.F<-survreg(S~X+frailty.gaussian(F),data=data,dist="weib")
```

```
> summary(weib.F)
```

Call:

```
survreg(formula = S ~ X + frailty.gaussian(F), data = data, dist = "weib")
```

	Value	Std. Error	z	p
(Intercept)	0.6188	0.2622	2.36	1.83e-02
X	-1.1386	0.1121	-10.16	3.12e-24
Log(scale)	0.0546	0.0417	1.31	1.91e-01

Scale= 1.06

Weibull distribution

Loglik(model)= -372 Loglik(intercept only)= -594

Chisq= 443 on 37 degrees of freedom, p= 0

Number of Newton-Raphson Iterations: 5 18

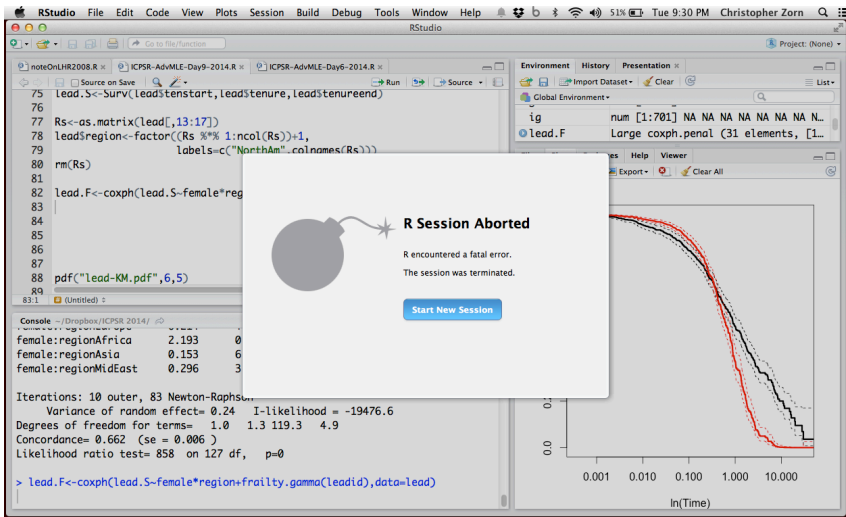
n= 800

Example: Leader Tenure

```
> lead.S<-Surv(lead$tenstart,lead$tenure,lead$tenureend)

> Rs<-as.matrix(lead[,13:17])
> lead$region<-factor((Rs %*% 1:ncol(Rs))+1,
                      labels=c("NorthAm",colnames(Rs)))
> rm(Rs)

> lead.F<-coxph(lead.S~female*region+frailty.gamma(leadid),data=lead)
```



Let's Try That Again

```
> lead.F<-coxph(lead.S~female*region+frailty.gamma(ccode),data=lead)
Warning message:
In coxpenal.fit(X, Y, strats, offset, init = init, control, weights = weights, :
  Inner loop failed to coverage for iterations 2 3

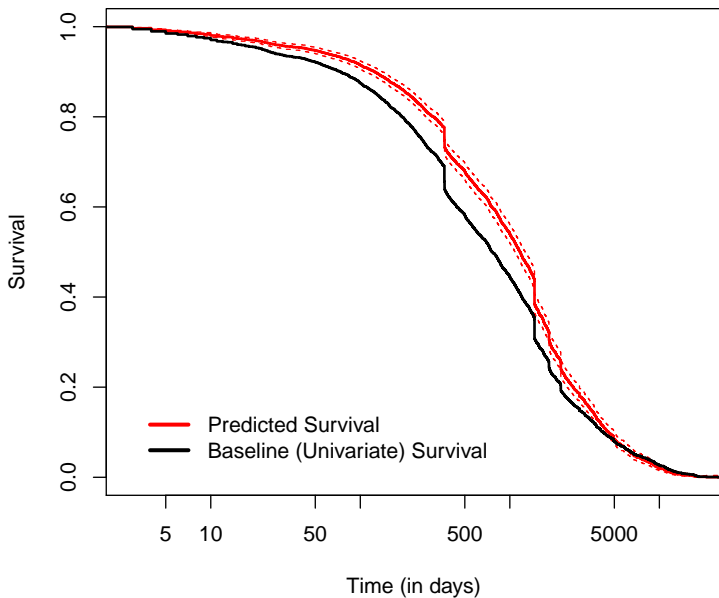
> summary(lead.F)
Call:
coxph(formula = lead.S ~ female * region + frailty.gamma(ccode),
      data = lead)

n= 15222, number of events= 2806
(22 observations deleted due to missingness)

              coef      se(coef) se2    Chisq  DF  p
female              1.2427  0.462   0.4594   7.24   1 0.007100
regionLatinAm      -0.1259  0.208   0.0333   0.37   1 0.540000
regionEurope        0.0414  0.160   0.0545   0.07   1 0.800000
regionAfrica       -0.7047  0.160   0.0840  19.45   1 0.000010
regionAsia         -0.3896  0.164   0.0742   5.65   1 0.017000
regionMidEast      -0.7478  0.186   0.0986  16.13   1 0.000059
frailty.gamma(ccode)              523.81 119 0.000000
female:regionLatinAm -1.8826  0.851   0.8495   4.89   1 0.027000
female:regionEurope -1.5424  0.624   0.6212   6.11   1 0.013000
female:regionAfrica  0.7854  0.861   0.8556   0.83   1 0.360000
female:regionAsia   -1.8765  0.572   0.5666  10.76   1 0.001000
female:regionMidEast -1.2175  0.861   0.8551   2.00   1 0.160000

Iterations: 10 outer, 83 Newton-Raphson
Variance of random effect= 0.24    I-likelihood = -19476.6
Degrees of freedom for terms= 1.0  1.3 119.3  4.9
Concordance= 0.662 (se = 0.006 )
Likelihood ratio test= 858 on 127 df,  p=0
```

Predicted vs. Actual



Extensions: Mixed-Effects Survival Models

- HLMs for survival data / outcomes
- Combined fixed, random, and mixed effects (random-coefficient) models
- R: Implemented in [coxme](#)
- Stata: [stmixed](#) (parametric models)
- Terry Therneau has a nice [vignette](#)

Mixed Effects Example

```
> lead.coxME<-coxme(lead.S~female + (1 | ccode/female),data=lead)
> lead.coxME
Cox mixed-effects model fit by maximum likelihood
Data: lead
events, n = 2806, 15222 (22 observations deleted due to missingness)
Iterations= 38 160
              NULL Integrated Fitted
Log-likelihood -19738      -19505 -19314

              Chisq  df p AIC  BIC
Integrated loglik   465   3 0 459  441
Penalized loglik    849 129 0 590 -177

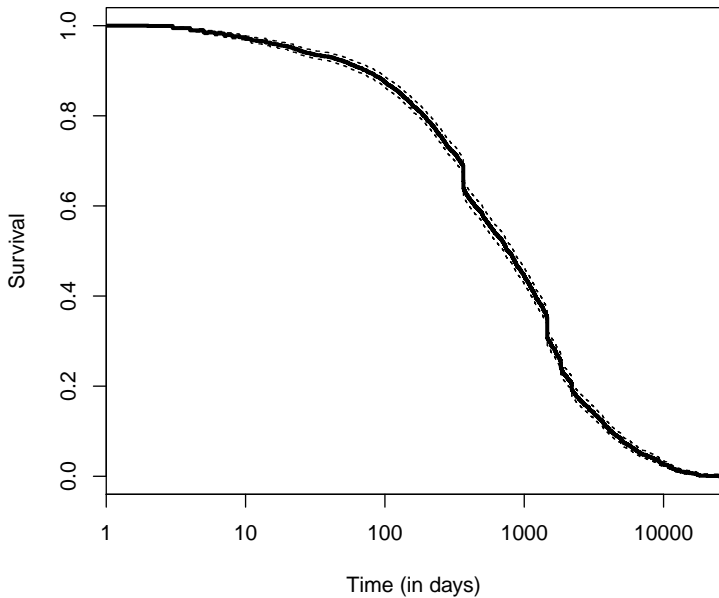
Model: lead.S ~ female + (1 | ccode/female)
Fixed coefficients
      coef exp(coef) se(coef)      z      p
female -0.07      0.93      0.22 -0.31 0.75

Random effects
Group      Variable      Std Dev Variance
ccode/female (Intercept) 0.279    0.078
ccode      (Intercept) 0.487    0.237
```

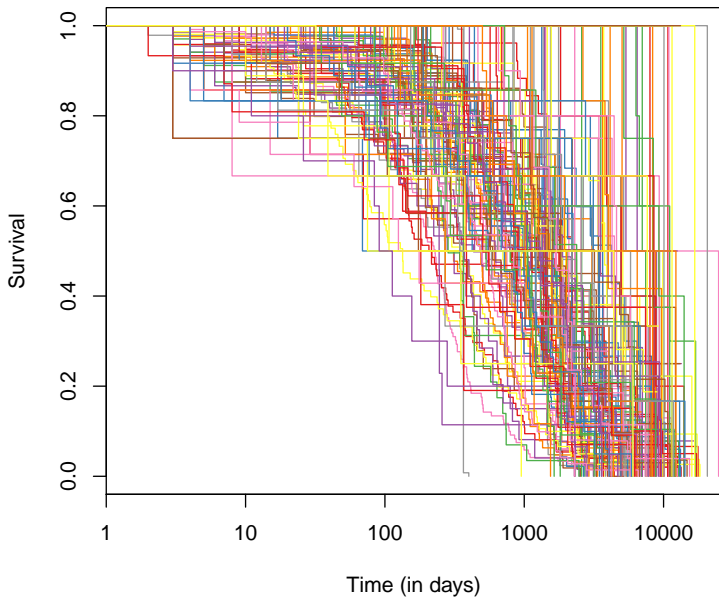
Stratify? Frailties? Clustering?

- Stratification \approx “fixed effects”
- Frailties \approx “random effects”
- “Robust” / cluster \approx GEE / PCSEs, etc.
- Not all combinations are possible, or make sense

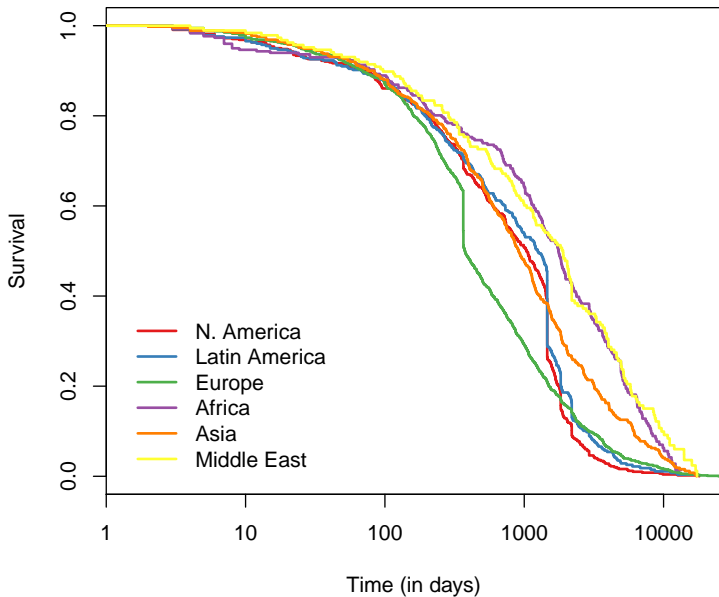
K-M Plot: Leaders



K-M Plot: Leaders (by country)



K-M Plot: Leaders (by region)



```
> lead.Fstrat<-coxph(lead.S~female*strata(region)+
  frailty.gamma(ccode),data=lead)
```

Warning message:

```
In coxpenal.fit(X, Y, strats, offset, init = init, control, weights = weights, :
  Inner loop failed to coverge for iterations 2 3 4
```

```
> summary(lead.Fstrat)
```

Call:

```
coxph(formula = lead.S ~ female * strata(region) + frailty.gamma(ccode),
  data = lead)
```

```
n= 15222, number of events= 2806
(22 observations deleted due to missingness)
```

	coef	se(coef)	se2	Chisq	DF	p
female	1.46	0.463	0.461	9.88	1	0.00170
frailty.gamma(ccode)				594.82	121	0.00000
female:strata(region)regi	-2.20	0.853	0.851	6.63	1	0.01000
female:strata(region)regi	-1.75	0.625	0.623	7.81	1	0.00520
female:strata(region)regi	0.13	0.869	0.864	0.02	1	0.88000
female:strata(region)regi	-2.07	0.573	0.568	13.04	1	0.00031
female:strata(region)regi	-1.31	0.862	0.857	2.32	1	0.13000

Strata + Clustering

```
> lead.stratCl<-coxph(lead.S~female*strata(region)+
  cluster(ccode),data=lead)

> summary(lead.stratCl)
Call:
coxph(formula = lead.S ~ female * strata(region) + cluster(ccode),
      data = lead)

n= 15222, number of events= 2806
(22 observations deleted due to missingness)

              coef exp(coef) se(coef) robust se      z
female              1.234    3.436   0.453    0.288  4.28
female:strata(region)region=LatinAm -1.881    0.152   0.842    0.627 -3.00
female:strata(region)region=Europe  -1.618    0.198   0.610    0.415 -3.90
female:strata(region)region=Africa   0.473    1.605   0.849    0.382  1.24
female:strata(region)region=Asia    -1.711    0.181   0.555    0.342 -5.00
female:strata(region)region=MidEast -0.709    0.492   0.846    0.349 -2.03

Concordance= 0.503 (se = 0.002 )
Rsquare= 0.001 (max possible= 0.864 )
Likelihood ratio test= 13.8 on 6 df,  p=0.0323
Wald test               = 81.6 on 6 df,  p=1.67e-15
Score (logrank) test = 20.1 on 6 df,  p=0.00263,  Robust = 14.4 p=0.0255

(Note: the likelihood ratio and score tests assume independence of
      observations within a cluster, the Wald and robust score tests do not).
```


From the frailty documentation:

“Note that use of a frailty term implies a mixed effects model and use of a cluster term implies a GEE approach; these cannot be mixed.”

```
> lead.FstratCl<-coxph(lead.S~female*strata(region)+frailty.gamma(ccode)+
                        cluster(ccode),data=lead)
Error in residuals.coxph(fit2, type = "dfbeta", collapse = cluster,
weighted = TRUE) :
  length of 'dimnames' [2] not equal to array extent
In addition: Warning message:
In coxpenal.fit(X, Y, strats, offset, init = init, control, weights = weights,
  Inner loop failed to coverge for iterations 2 3 4
```



693 posts

In reply to [this post](#) by Ehsan Karim

Addition of a cluster() term fits a Generalized Estimating Equations (GEE) type of model, addition of frailty() fits a random effects model (Mixed Effect or ME). In glm analysis (linear regression, logistic regression, etc) the arguments about the advantages/disadvantages of GEE vs ME would easily fill a volume. Most of this argument carries over to the coxph case; I find both approaches useful.

Caveats:

1. Coxph with cluster() only allows the "working independence" variance structure. The details for other variance structures were worked out by Alicia Z in her Iowa State PhD thesis, but I've never gotten around to implementing it.
2. For random effects, the coxme function is preferred.
3. In comparing GEE and ME one part of the argument is that the former model is "marginal" and the second "conditional", and thus the coefficients from the models mean different things. I take this with a grain of salt. Remember that ALL models are wrong.

Terry Therneau

[\[hidden email\]](#) mailing list

<https://stat.ethz.ch/mailman/listinfo/r-help>

PLEASE do read the posting guide <http://www.R-project.org/posting-guide.html> and provide commented, minimal, self-contained, reproducible code.

Topics We Didn't Cover

- ★ Joint Models for Survival and Longitudinal Outcomes
 - e.g., survival + binary / multinomial / continuous variables
 - *inter alia* R package [JM](#) (Rizopolous 2010)
 - Recent reference is [Viviani et al. \(2014\)](#)
- ★ Causal Inference ([IVs](#), RDDs, matching, etc.)
- ★ Variable Selection: regularization, bagging, boosting, stacking, lasso, etc.
- ★ Bayesian approaches (esp. for high-dimensional competing risks & hierarchical models); see [Ibrahim et al. \(2005\)](#)
- ★ New / better tools for interpretation and graphics (e.g. [simPH](#))

Journals:

- *Biometrics* / *Biometrika*
- *Statistics in Medicine*
- *Statistical Methods in Medical Research*
- *Lifetime Data Analysis*

Places:

- Biostatistics / Epidemiology / Public Health
- Statistics departments
- *Not* economics, psychology, etc.