

# Fitting Cox Regression Models

James H. Steiger

Department of Psychology and Human Development  
Vanderbilt University

GCM, 2010

# Fitting Cox Regression Models

- 1 Introduction
- 2 Toward a Model for Continuous-Time Hazard
- 3 A Log Hazard Model
- 4 Fitting the Cox Regression Model to Data
  - Introduction
  - The Partial Likelihood Method
  - Implications and Consequences of the Cox Approach
- 5 Interpreting Results from a Cox Regression
  - Introduction
  - Interpreting Parameter Estimates
  - Evaluating Overall Goodness-of-Fit
  - Drawing Inference using Estimated Standard Errors
  - Summarizing Findings Using Risk Scores
- 6 Nonparametric Strategies for Displaying Results
  - Introduction
  - Recovered Baseline Functions
  - Graphing the Predicted Functions

# Introduction

In this module, we discuss the Cox proportional hazards model and how to fit it.

## A Model for Continuous-Time Hazard

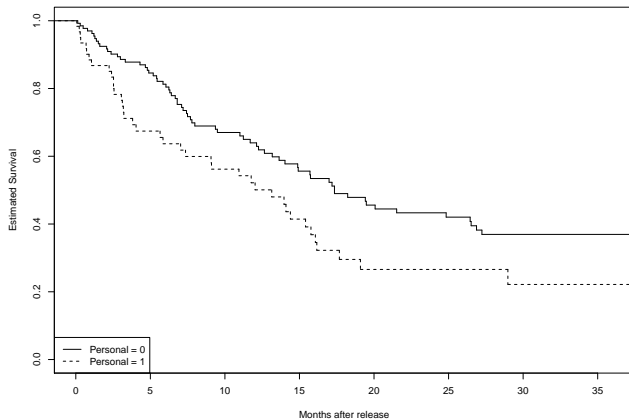
The basic form of the model is that log-hazard is predicted as a baseline function plus a linear combination of predictors. We illustrate the model with data from a recidivism study by Henning and Frueh (1996). We begin by examining the effect of a single 0-1 predictor, *PERSONAL*, which indicates whether the crime the person was incarcerated for was person-related.

# Survivor Function Plot

Here is code for an exploratory survivor function plot, using the Kaplan-Meier estimates.

```
> rearrest<-read.table("rearrest.csv", sep=",", header=T)
> rearrest0 <- subset(rearrest, personal == 0)
> rearrest1 <- subset(rearrest, personal == 1)
> f14.1.0 <- summary(survfit(Surv(rearrest0$months,
+ abs(rearrest0$censor - 1))^1))
> f14.1.1 <- summary(survfit(Surv(rearrest1$months,
+ abs(rearrest1$censor - 1))^1))
> s.hat.0 <- f14.1.0[[1]][[1]]
> time.0 <- f14.1.0[[2]][[1]]
> s.hat.1 <- f14.1.1[[1]][[1]]
> time.1 <- f14.1.1[[2]][[1]]
> s.hat.steps.0 <- stepfun(time.0, c(1, s.hat.0))
> s.hat.steps.1 <- stepfun(time.1, c(1, s.hat.1))
> plot(s.hat.steps.0, do.points = FALSE,
+ xlim = c(0, 36), ylim = c(0,1),
+ ylab = "Estimated Survival",
+ xlab = "Months after release", main = "")
> lines(s.hat.steps.1, do.points = FALSE,
+ xlim = c(0,36), lty = 2)
> legend("bottomleft", c("Personal = 0",
+ "Personal = 1"), lty = c(1, 2))
```

# Survivor Function Plot

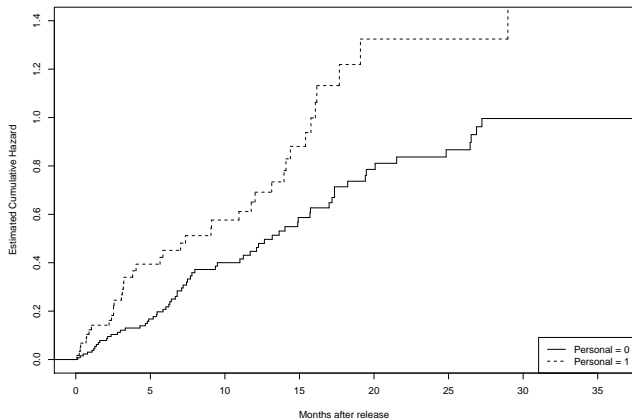


# Cumulative Hazard Function Plot

Here is code for a cumulative hazard function plot, using the estimates based on the negative-log survivor function.

```
> h.hat.0 <- -log(s.hat.0)
> h.hat.1 <- -log(s.hat.1)
> h.hat.steps.0 <- stepfun(time.0, c(0, h.hat.0))
> h.hat.steps.1 <- stepfun(time.1, c(0, h.hat.1))
> plot(h.hat.steps.0, do.points = FALSE,
+ xlim = c(0, 36), ylim = c(0, 1.4),
+ ylab = "Estimated Cumulative Hazard",
+ xlab = "Months after release", main = "")
> lines(h.hat.steps.1, do.points = FALSE,
+ xlim = c(0, 36), lty = 2)
> legend("bottomright", c("Personal = 0",
+ "Personal = 1"), lty = c(1, 2))
```

# Cumulative Hazard Function Plot



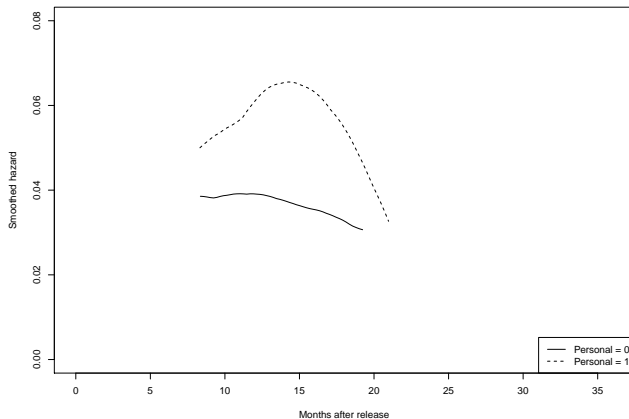


# Hazard Function Plot

Here is code for a hazard function plot, using the kernel smoothed estimates.

```
> smooth<- function(width, time, survive){
+   n=length(time)
+   lo=time[1] + width
+   hi=time[n] - width
+   npt=50
+   inc=(hi-lo)/npt
+   s=lo+t(c(1:npt))*inc
+   slag = c(1, survive[1:n-1])
+   h=1-survive/slag
+   x1 = as.vector(rep(1, npt))%*(t(time))
+   x2 = t(s)%*as.vector(rep(1,n))
+   x = (x1 - x2) / width
+   k=.75*(1-x*x)*(abs(x)<=1)
+   lambda=(k%*/h)/width
+   smoothed= list(x = s, y = lambda)
+   return(smoothed)
+ }
> bw1.0 <- smooth(8, time.0, s.hat.0)
> bw1.1 <- smooth(8, time.1, s.hat.1)
> plot(bw1.0$x, bw1.0$y, type = "l", xlim = c(0, 36), ylim = c(0, .08),
+   xlab = "Months after release",
+   ylab = "Smoothed hazard")
> points(bw1.1$x, bw1.1$y, type = "l", lty = 2)
> legend("bottomright", c("Personal = 0", "Personal = 1"), lty = c(1, 2))
```

# Hazard Function Plot



## A Model for Log Cumulative Hazard

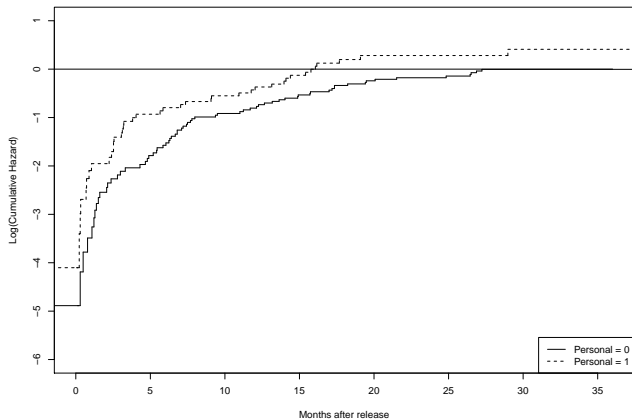
We will find it mathematically convenient to develop a model for the log cumulative hazard, which is also equal to  $\log(-\log S(t))$ . Let's look at some graphs.

# A Model for Log Cumulative Hazard

Here is some R code for generating a graph of the log cumulative hazard function.

```
> l.hat.0 <- log(h.hat.0)
> l.hat.1 <- log(h.hat.1)
> log.h.steps.0<- stepfun(time.0, c(l.hat.0[1], l.hat.0))
> log.h.steps.1<- stepfun(time.1, c(l.hat.1[1], l.hat.1))
> par(mfrow=c(1,1))
> plot(log.h.steps.0, do.points = FALSE,
+ xlim = c(0, 36), ylim = c(-6.0,1),
+ ylab = "Log(Cumulative Hazard)",
+ xlab = "Months after release", main = "")
> lines(log.h.steps.1, do.points = FALSE,
+ xlim = c(0,36), lty = 2)
> points(c(-5, 36), c(0,0), type = "l")
> legend("bottomright", c("Personal = 0",
+ "Personal = 1"), lty = c(1, 2))
```

# Log Cumulative Hazard Function Plot

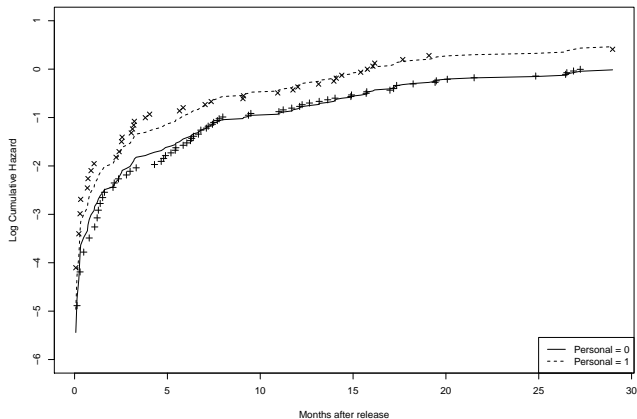


# Log Cumulative Hazard Plot with Data

Here is some R code for generating a graph of the fitted log cumulative hazard function, with the data points superimposed.

```
> attach(rearrest)
> f14.2 <- coxph(Surv(months, abs(censor - 1))~personal,
+ method = "efron")
> personal.0 <- data.frame(personal=0)
> personal.1 <- data.frame(personal=1)
> s.base <- survfit(f14.2, newdata = personal.0,
+ type = "kaplan-meier")
> s.personal <- survfit(f14.2, newdata = personal.1,
+ type = "kaplan-meier")
> h.base <- -log(s.base$surv)
> h.personal <- -log(s.personal$surv)
> l.h.base <- log(h.base)
> l.h.personal <- log(h.personal)
> plot(s.base$time, l.h.base, type = "l", lty = 1,
+ ylim = c(-6, 1),
+ xlab = "Months after release",
+ ylab = "Log Cumulative Hazard")
> points(s.personal$time, l.h.personal, type = "l",
+ lty = 2)
> points(time.0, l.hat.0, pch = 3)
> points(time.1, l.hat.1, pch = 4)
> legend("bottomright", c("Personal = 0", "Personal = 1"),
+ lty = c(1, 2))
```

# Log Cumulative Hazard Function Plot with Data



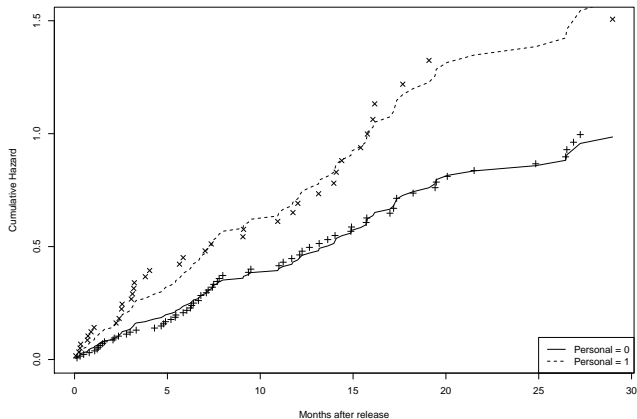
## Cumulative Hazard Plot with Data

Compare that to the graph of the raw cumulative hazard, produced by the following code.

```
> plot(s.base$time, h.base, type = "l",  
+ lty = 1, ylim = c(0,1.5),  
+ xlab = "Months after release",  
+ ylab = "Cumulative Hazard")  
> points(s.personal$time, h.personal,  
+ type = "l", lty = 2)  
> points(time.0, h.hat.0, pch = 3)  
> points(time.1, h.hat.1, pch = 4)  
> legend("bottomright", c("Personal = 0",  
+ "Personal = 1"), lty = c(1, 2))
```



# Cumulative Hazard Function Plot with Data



## Implications of the Graphs for Modeling

We can see that the log cumulative hazard plots are almost parallel. So a reasonable model is of the form

$$\log H(t_{ij}) = \log H_0(t_j) + \beta_1 PERSONAL_i \quad (1)$$

Exponentiating both sides gives us

$$H(t_{ij}) = H_0(t_j) \exp(\beta_1 PERSONAL_i) \quad (2)$$

An immediate implication is that, if we take the ratio of the two hazard functions  $H_1$  (when  $PERSONAL = 1$ ) and  $H_0$  (when  $PERSONAL = 0$ ), we get

$$H_1/H_0 = \exp(\beta_1) \quad (3)$$

## A Log Hazard Model

The way the mathematics works out, the preceding model remains essentially of the same form when log hazard is modeled instead of log cumulative hazard. That is, for  $P$  predictors, we have

$$\log h(t_{ij}) = \log h_0(t_j) + \exp \left( \sum_{k=1}^P \beta_k X_{kij} \right) \quad (4)$$

This implies that the difference in the  $\log h$  and  $\log h_0$  curves is the same as the difference between the  $\log H$  and  $\log H_0$  curves for any set of predictor values. This is a surprising result, but a useful one.

# Introduction

The Cox regression model approach to survival analysis is based on the use of a partial likelihood approach. This function is constructed by “conditioning” on the observed event times and computing a conditional probability that individual  $i$  experienced the event, *given that someone did*.

# The Partial Likelihood Method

Suppose that  $k$  of the survival times from  $n$  individuals are uncensored and distinct, and  $n - k$  are right-censored. Let  $t_{(1)} < t_{(2)} < \dots < t_{(k)}$  be the ordered  $k$  distinct failure times with corresponding covariates  $\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(k)}$ . Let  $\mathbf{R}(t_{(i)})$  be the risk set at time  $t_{(i)}$ .  $\mathbf{R}(t_{(i)})$  consists of all persons whose survival times are at least  $t_{(i)}$ . The total partial likelihood is the product of all the individual partial likelihoods, and the log partial likelihood can be shown (e.g., Lee & Wang, 2003, pp. 301–302) to be equal to

$$\log L(\mathbf{b}) = \sum_{i=1}^k \left\{ \mathbf{b}' \mathbf{x}_{(i)} - \log \left[ \sum_{l \in \mathbf{R}(t_{(i)})} \exp \mathbf{b}' \mathbf{x}_l \right] \right\} \quad (5)$$

# Implications and Consequences

The result of Equation 5 has several implications:

- 1 The shape of the baseline hazard is irrelevant—it is nowhere to be seen in the log likelihood calculation
- 2 The precise event times are irrelevant as well
- 3 Ties substantially complicate calculations, in which case an exact method is available, but often it is impractical, and one of several approximations can be tried (the textbook recommends Efron's 1977 method)

# Introduction

The focus of a Cox regression is the relative hazard over and above the baseline associated with the predictors.

By exponentiating a  $\beta$  attached to a particular predictor, we can evaluate the *hazard ratio* associated with a 1-point increase in the predictor.

# Interpreting Parameter Estimates

Models A, B, C, systematically add *PERSONAL*, *PROPERTY*, and (centered) *AGE* as individual predictors.

Model D adds all 3.

In the following slides, we show the results of fitting the 4 models.



# Model A

```
> tab14.1A <- coxph(Surv(months, abs(censor - 1))~personal)
> summary(tab14.1A)
```

Call:

```
coxph(formula = Surv(months, abs(censor - 1)) ~ personal)
```

n= 194

	coef	exp(coef)	se(coef)	z	Pr(> z )
personal	0.479	1.614	0.202	2.36	0.018 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
personal	1.61	0.62	1.09	2.4

Rsquare= 0.027 (max possible= 0.994 )

Likelihood ratio test= 5.32 on 1 df, p=0.0210

Wald test = 5.59 on 1 df, p=0.0181

Score (logrank) test = 5.69 on 1 df, p=0.017

# Model B

```
> tab14.1B <- coxph(Surv(months, abs(censor - 1))~property)
> summary(tab14.1B)
```

Call:

```
coxph(formula = Surv(months, abs(censor - 1)) ~ property)
```

n= 194

	coef	exp(coef)	se(coef)	z	Pr(> z )	
property	1.195	3.302	0.349	3.42	0.00063	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
property	3.3	0.303	1.67	6.55

Rsquare= 0.08 (max possible= 0.994 )

Likelihood ratio test= 16.2 on 1 df, p=5.68e-05

Wald test = 11.7 on 1 df, p=0.000626

Score (logrank) test = 13.1 on 1 df, p=0.000290

# Model C

```
> tab14.1C <- coxph(Surv(months, abs(censor - 1))~cage)
> summary(tab14.1C)
```

Call:

```
coxph(formula = Surv(months, abs(censor - 1)) ~ cage)
```

n= 194

	coef	exp(coef)	se(coef)	z	Pr(> z )	
cage	-0.0681	0.9341	0.0156	-4.36	1.3e-05	***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
cage	0.934	1.07	0.906	0.963

Rsquare= 0.112 (max possible= 0.994 )

Likelihood ratio test= 23.0 on 1 df, p=1.65e-06

Wald test = 19 on 1 df, p=1.31e-05

Score (logrank) test = 19.2 on 1 df, p=1.19e-05

# Model D

```
> tab14.1D <- coxph(Surv(months, abs(censor - 1))~personal + property + cage)
> summary(tab14.1D)
```

Call:

```
coxph(formula = Surv(months, abs(censor - 1)) ~ personal + property +
      cage)
```

n= 194

	coef	exp(coef)	se(coef)	z	Pr(> z )
personal	0.5691	1.7667	0.2052	2.77	0.0055 **
property	0.9358	2.5492	0.3509	2.67	0.0077 **
cage	-0.0667	0.9355	0.0168	-3.98	7e-05 ***

---  
 Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
personal	1.767	0.566	1.182	2.641
property	2.549	0.392	1.282	5.071
cage	0.935	1.069	0.905	0.967

Rsquare= 0.182 (max possible= 0.994 )

Likelihood ratio test= 39.0 on 3 df, p=1.77e-08

Wald test = 29 on 3 df, p=2.21e-06

Score (logrank) test = 30.3 on 3 df, p=1.20e-06

## Evaluating Goodness-of-Fit

Although Cox regression output does not, strictly speaking, produce a “deviance” statistic, it matters little to us in practice. Standard software output includes a likelihood ratio chi-square statistic that compares the fitted model to a model with no predictors, and a  $-2LL$  statistic.

This statistic allows one to test the overall significance of the set of predictors included in the fitted model. Moreover, by constructing and fitting a sequence of nested models, one may, by subtraction, obtain chi-square difference statistics in essentially the same way as before.

## Evaluating Goodness-of-Fit

### Example (Chi-Square Difference Test)

The output for Model D, with all 3 predictors included, shows  $-2LL = 950.44$  with 3 parameters, while Model C shows  $-2LL = 966.43$  with 1 parameter.

This shows that adding *PROPERTY* and *PERSONAL* to *AGE* yields a chi-square of 15.99 with 2 degrees of freedom, which is significant beyond the .01 level.

Introduction

Toward a Model for Continuous-Time Hazard

A Log Hazard Model

Fitting the Cox Regression Model to Data

**Interpreting Results from a Cox Regression**

Nonparametric Strategies for Displaying Results

Introduction

Interpreting Parameter Estimates

Evaluating Overall Goodness-of-Fit

**Drawing Inference using Estimated Standard Errors**

Summarizing Findings Using Risk Scores

# Wald Tests via Estimated Standard Errors

## Using Risk Scores

Risk scores for an individual utilize that individual's predictor values in conjunction with the  $\beta$  estimates to produce a hazard ratio estimate for that person relative to the baseline. Let's look at an example to see how risk scores work.

Suppose an individual in the recidivism study is 4 years below the mean age at the time of release, and consequently has an  $AGE = -4$ , and has  $PERSONAL = 1$  and  $PROPERTY = 1$ . Then the log hazard ratio is  $-4 \times -.0667 + 1 \times .5691 + 1 \times .9358 = 1.7717$ . So the hazard ratio, or "risk score" for the individual is  $\exp 1.7717$ , or 5.8805. In other words, this individual is almost 6 times as likely to recidivate as the person who is of average age, with scores of 0 on the  $PERSONAL$  and  $PROPERTY$  variables.

Keep in mind that this risk score is an estimate of a population quantity, and itself has a standard error.

(Question: What kind of information would we need to construct a standard error for the risk score? C.P.)



# Computing Risk Scores

Here is code, a modified version of the code on the UCLA site, for constructing functions that can create a table of risk scores for any list of IDs. Note that this function is highly customized, but you could, with some effort, create a much more general function that would require a list of the predictors to be used in creating the risk score.

```
> make.a.row <- function(ID.num)
+ {
+ ##grab data for the individual
+ temp <- subset(rearrest, id==ID.num)
+ ##compute the log score and exponentiate it
+ log.score <- tab14.ID$coef[1]*temp$personal +
+ tab14.ID$coef[2]*temp$property +
+ tab14.ID$coef[3]*temp$age
+ score <- exp(log.score)
+ ## convert to a day metric
+ day <- temp$months * (365/12)
+ ## construct a row in the table
+ table.row <- cbind(temp$id, temp$personal,
+ temp$property, temp$age, score, day,
+ temp$months, temp$censor)
+ colnames(table.row) <- c("ID", "personal",
+ "property", "centered.age", "risk.score", "day",
+ "months", "censor")
+ rownames(table.row) <- c()
+ return(table.row)
+ }
> make.table <- function(list.of.ids)
+ {
+ the.table <- c()
+ for(i in 1:length(list.of.ids))
+ {
+ next.row <- make.a.row(list.of.ids[i])
+ the.table <- rbind(the.table,next.row)
+ }
+ return(the.table)
+ }
```

# Computing Risk Scores

Here is the table.

```
> make.table(c(22,8,187,26,5,130,106,33))
```

	ID	personal	property	centered.age	risk.score	day	months	censor
[1,]	22	0	0	0.2577	0.9830	51.964	1.7084	1
[2,]	8	1	1	22.4507	1.0072	18.987	0.6242	1
[3,]	187	1	0	-7.2002	2.8562	1095.000	36.0000	1
[4,]	26	0	1	-7.3015	4.1491	71.951	2.3655	0
[5,]	5	1	1	-7.1646	7.2638	8.994	0.2957	0
[6,]	130	0	1	22.3905	0.5724	485.667	15.9671	1
[7,]	106	0	0	16.2030	0.3393	355.756	11.6961	0
[8,]	33	1	0	27.0613	0.2905	84.942	2.7926	1

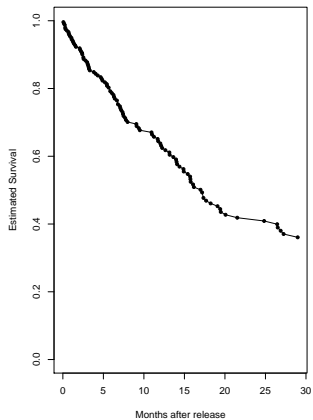
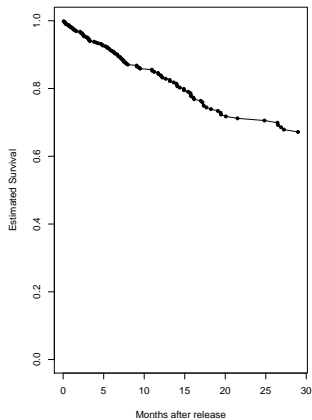
# Survival Function Plot

```

> tab14.1D <- coxph(Surv(months, abs(censor - 1))~personal +
+   property + cage)
> base <- data.frame(personal=0, property=0, cage=0)
> ss <- survfit(tab14.1D)
> ss.base <- survfit(tab14.1D, newdata = base)
> avg <- data.frame(personal=mean(personal),
+   property=mean(property), cage=0)
> ss.avg <- survfit(tab14.1D, newdata = avg)
> par(mfrow=c(1,2))
> plot(ss.base$time, ss.base$surv, type = "l",
+   ylim = c(0,1),
+   xlab = "Months after release",
+   ylab = "Estimated Survival")
> points(ss.base$time, ss.base$surv, pch = 20)
> plot(ss.avg$time, ss.avg$surv, type = "l",
+   ylim = c(0,1),
+   xlab = "Months after release",
+   ylab = "Estimated Survival")
> points(ss.avg$time, ss.avg$surv, pch = 20)

```

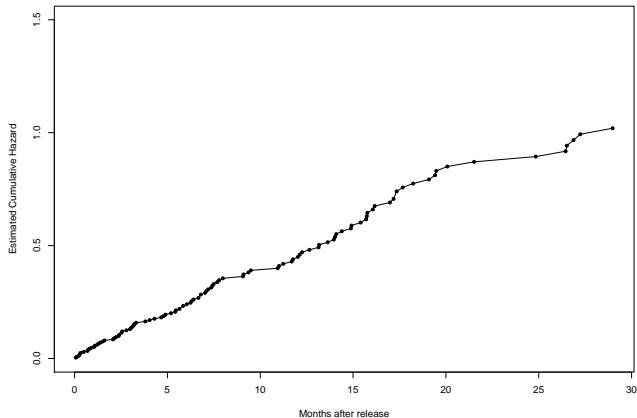
# Survival Function Plot



# Cumulative Hazard Function Plot

```
> plot(ss.base$time, -log(ss.base$surv),  
+      type = "l", ylim = c(0,1.5),  
+      xlab = "Months after release",  
+      ylab = "Estimated Cumulative Hazard")  
> points(ss.base$time, -log(ss.base$surv), pch = 20)  
> plot(ss.avg$time, -log(ss.avg$surv), type = "l",  
+      ylim = c(0,1.5), xlab = "Months after release",  
+      ylab = "Estimated Cumulative Hazard")  
> points(ss.avg$time, -log(ss.avg$surv), pch = 20)
```

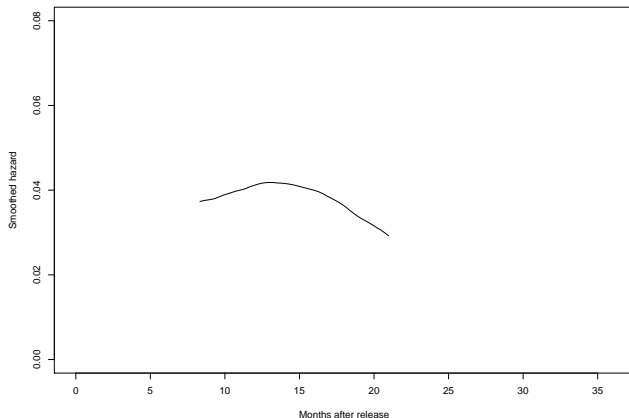
# Survival Function Plot



# Hazard Function Plot

```
> smooth.base <- smooth(8, ss.base$time, ss.base$surv)
> smooth.avg <- smooth(8, ss.avg$time, ss.avg$surv)
> plot(smooth.base$x, smooth.base$y, type = "l",
+      xlim = c(0, 36), ylim = c(0, .08),
+      xlab = "Months after release",
+      ylab = "Smoothed hazard")
> plot(smooth.avg$x, smooth.avg$y, type = "l",
+      xlim = c(0, 36), ylim = c(0, .08),
+      xlab = "Months after release",
+      ylab = "Smoothed hazard")
```

# Survival Function Plot

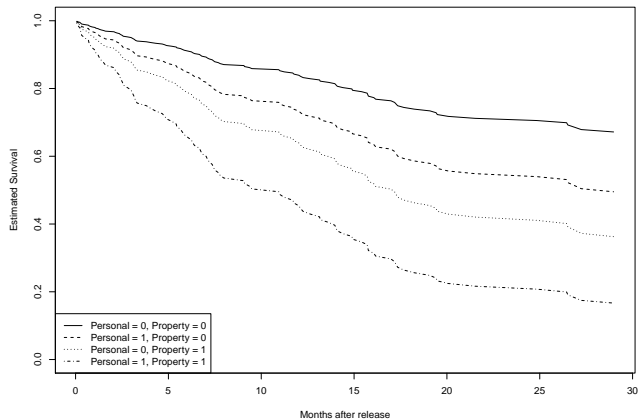




# Survivor Function Code

```
> tab14.1D <- coxph(Surv(months, abs(censor - 1))~personal + property + cage)
> base <- data.frame(personal=0, property=0, cage=0)
> personal.only <- data.frame(personal=1, property=0, cage=0)
> property.only <- data.frame(personal=0, property=1, cage=0)
> both <- data.frame(personal=1, property=1, cage=0)
> s.base <- survfit(tab14.1D, newdata = base)
> s.personal <- survfit(tab14.1D, newdata = personal.only)
> s.property <- survfit(tab14.1D, newdata = property.only)
> s.both <- survfit(tab14.1D, newdata = both)
> par(mfrow=c(1,1))
> plot(s.base$time, s.base$surv, type = "l", lty = 1, ylim = c(0,1),
+      xlab = "Months after release", ylab = "Estimated Survival")
> points(s.personal$time, s.personal$surv, type = "l", lty = 2)
> points(s.property$time, s.property$surv, type = "l", lty = 3)
> points(s.both$time, s.both$surv, type = "l", lty = 4)
> legend("bottomleft", c("Personal = 0, Property = 0",
+ "Personal = 1, Property = 0",
+ "Personal = 0, Property = 1", "Personal = 1, Property = 1"),
+ lty = c(1, 2, 3, 4))
```

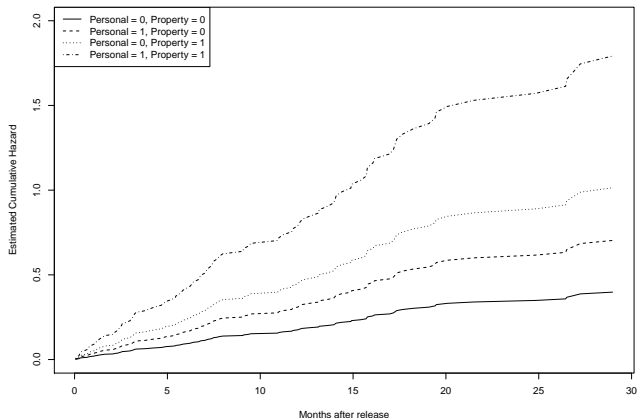
# Survivor Function



# Cumulative Hazard Function Code

```
> h.base <- -log(s.base$surv)
> h.personal <- -log(s.personal$surv)
> h.property <- -log(s.property$surv)
> h.both <- -log(s.both$surv)
> plot(s.base$time, h.base, type = "l", lty = 1, ylim = c(0,2),
+      xlab = "Months after release",
+      ylab = "Estimated Cumulative Hazard")
> points(s.personal$time, h.personal, type = "l", lty = 2)
> points(s.property$time, h.property, type = "l", lty = 3)
> points(s.both$time, h.both, type = "l", lty = 4)
> legend("topleft", c("Personal = 0, Property = 0",
+ "Personal = 1, Property = 0",
+ "Personal = 0, Property = 1", "Personal = 1, Property = 1"),
+      lty = c(1, 2, 3, 4))
```

# Cumulative Hazard Function



# Log Cumulative Hazard Function Code

```
> l.h.base <- log(h.base)
> l.h.personal <- log(h.personal)
> l.h.property <- log(h.property)
> l.h.both <- log(h.both)
> plot(s.base$time, l.h.base, type = "l", lty = 1,
+      ylim = c(-7,1),
+      xlab = "Months after release",
+      ylab = "Log(Cumulative Hazard)")
> points(s.personal$time, l.h.personal, type = "l", lty = 2)
> points(s.property$time, l.h.property, type = "l", lty = 3)
> points(s.both$time, l.h.both, type = "l", lty = 4)
> legend("bottomright", c("Personal = 0, Property = 0",
+ "Personal = 1, Property = 0",
+ "Personal = 0, Property = 1", "Personal = 1, Property = 1"),
+ lty = c(1, 2, 3, 4))
```

# Log Cumulative Hazard Function

