

# Fitting Basic Discrete-Time Hazard Models

James H. Steiger

Department of Psychology and Human Development  
Vanderbilt University

GCM, 2010

# Fitting Basic Discrete-Time Hazard Models

- 1 Introduction
- 2 An Example
- 3 Preliminary Analyses
  - Within-Group Plots
- 4 Logit-Based Models for the Hazard Function
  - Plotting on the Odds Scale
  - Plotting on the Logit Scale
- 5 A Discrete-Time Hazard Model
  - The Model
  - Interpreting the Model
- 6 Fitting the Discrete-Time Survival Model
  - Model A – Baseline
  - Model B – Baseline + PT
  - Model C – Baseline + PAS
  - Model D – Baseline + PT + PAS
- 7 Deviance-Based Hypothesis Tests
  - Tests for Individual Parameters
  - Simultaneous Deviance Tests for Groups of Parameters
- 8 Wald  $Z$  and  $\chi^2$  Tests
- 9 Asymptotic Confidence Intervals
  - Asymptotic Confidence Intervals for Parameters
  - Asymptotic Confidence Intervals for Odds Ratios
- 10 Computing and Plotting a Fitted Model
  - Computing and Plotting the Hazard Function

# Introduction

In this module, we examine the characteristics of some basic discrete-time hazard models, and explore how they are fit to data.

We address questions about the covariates of hazard and survival. Some examples:

- 1 What factors are connected with early relapse after treatment for alcoholism?
- 2 What coping strategies enable some sex offenders from re-offending?
- 3 Is one preventive care strategy better than another for preventing infection during dialysis?
- 4 Does choice of diet affect the likelihood of developing cancer?

# Introduction

We attempt to answer questions like these by fitting survival models to data. Our efforts will have much in common with regression analysis.

- 1 We'll fit a model, and then
- 2 Estimate its parameters and goodness of fit and
- 3 Decide whether perhaps another model would be better for our data
- 4 If the current model seems reasonable, we'll
- 5 Interpret the results in terms of our research questions and
- 6 Communicate our results in standard statistical terms

## An Example

Chapter 11 of Singer and Willett is built around the study Capaldi, et al. (1996) on the grade of first heterosexual intercourse for a sample of “at-risk” boys.

The key question we shall address is whether the survival time is systematically related to the whether the boys lived with both biological parents during their formative years.

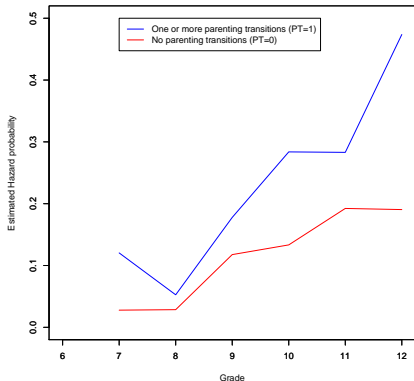
The covariate, PT, is scored 1 for boys who experienced at least one “parenting transition,” and 0 otherwise.

# Within-Group Estimated Hazard Function

A first step in exploratory analysis is to examine the within-sample estimated hazard function plots.

```
> library(survival)
> firstsex<-read.table("firstsex.csv", sep=",", header=T)
> ts0 <- survfit( Surv(time, 1-censor)~ 1, conf.type="none",
+   subset=(pt==0), data=firstsex)
> ts1 <- survfit( Surv(time, 1-censor)~ 1, conf.type="none",
+   subset=(pt==1), data=firstsex)
> h0<-ts0$n.event/ts0$n.risk
> h1<-ts1$n.event/ts1$n.risk
> plot(ts0$time, h0, type="l", ylab="Estimated Hazard probability",
+   xlab="Grade", ylim=c(0.0, 0.5), xlim=c(6, 12), col="red")
> par(new=T)
> plot(ts1$time, h1, type="l", ylab=" ", ylim=c(0.0, 0.5),
+   xlim=c(6, 12), xlab="", col="blue")
> legend(7,0.5, c("One or more parenting transitions (PT=1)",
+ "No parenting transitions (PT=0)"),lty=1,col=c("blue","red"))
```

# Within-Group Estimated Hazard Function



## Within-Group Estimated Hazard Function

As we can clearly see from the hazard function plots,

- 1 The hazard function for both groups starts low and begins to rise sharply at 8th grade
- 2 The hazard function for the PT=1 group is at a consistently higher level than that of the PT=0 group



# Within-Group Estimated Hazard Functions

We can examine the hazard and survival probabilities in life tables separately by groups. Here is the table for PT=0:

```
> tab11.1.0<-cbind(time=ts0$time, nleft=ts0$n.risk, failed=ts0$n.event, hazard=h0, survival=ts0$surv)
> tab11.1.0
```

	time	nleft	failed	hazard	survival
[1,]	7	72	2	0.02778	0.9722
[2,]	8	70	2	0.02857	0.9444
[3,]	9	68	8	0.11765	0.8333
[4,]	10	60	8	0.13333	0.7222
[5,]	11	52	10	0.19231	0.5833
[6,]	12	42	8	0.19048	0.4722

And here is the corresponding table for PT=1:

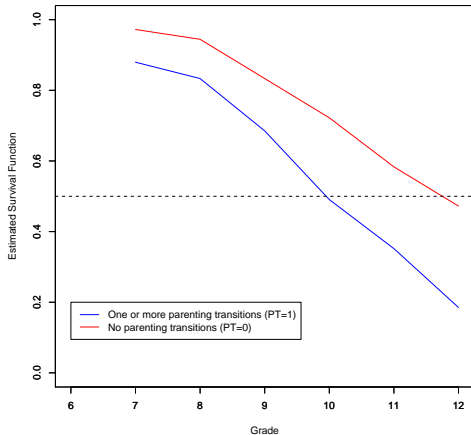
```
> tab11.1.1<-cbind(time=ts1$time, nleft=ts1$n.risk, failed=ts1$n.event, hazard=h1, survival=ts1$surv)
> tab11.1.1
```

	time	nleft	failed	hazard	survival
[1,]	7	108	13	0.12037	0.8796
[2,]	8	95	5	0.05263	0.8333
[3,]	9	90	16	0.17778	0.6852
[4,]	10	74	21	0.28378	0.4907
[5,]	11	53	15	0.28302	0.3519
[6,]	12	38	18	0.47368	0.1852

# Within-Group Estimated Survival Functions

The following code plots the estimated within-group survival functions.

```
> plot(ts0$time, ts0$surv, type="l",  
+      ylab="Estimated Survival Function",  
+      xlab="Grade", ylim=c(0.0, 1.0), xlim=c(6, 12), col="red")  
> par(new=T)  
> plot(ts1$time, ts1$surv, type="l", ylab=" ", ylim=c(0.0, 1.0),  
+      xlim=c(6, 12), xlab="", col="blue")  
> abline(h=c(.5), lty=2)  
> legend(6,0.2, c("One or more parenting transitions (PT=1)",  
+ "No parenting transitions (PT=0)"),lty=1,col=c("blue","red"))
```



# Combined Life Table

Here is the life table for the combined groups:

```
> tsall <- survfit( Surv(time, 1-censor)~ 1, conf.type="none",
+   data=firstsex)
> h<-tsall$n.event/tsall$n.risk
> tab11.1.all<-cbind(time=tsall$time, nleft=tsall$n.risk,
+ failed=tsall$n.event, hazard=h, survival=tsall$surv)
> tab11.1.all
```

	time	nleft	failed	hazard	survival
[1,]	7	180	15	0.08333	0.9167
[2,]	8	165	7	0.04242	0.8778
[3,]	9	158	24	0.15190	0.7444
[4,]	10	134	29	0.21642	0.5833
[5,]	11	105	25	0.23810	0.4444
[6,]	12	80	26	0.32500	0.3000

## Logit-Based Models for the Hazard Function

The hazard data we just examined suggest a regression model. However, probability is bounded between 0 and 1, a fact that, in practice, generates lots of problems (which is why we have logistic regression).

The odds of an event  $X$  are defined as

$$\text{Odds}(X) = \frac{\Pr(X)}{1 - \Pr(X)} \quad (1)$$

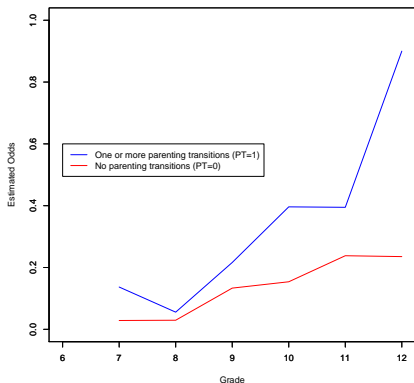
When we convert probabilities to odds, we convert monotonically to a scale that ranges from 0 to infinity, with odds of 1 corresponding to a probability of 0.50.

# Plotting on the Odds Scale

Here is code for calculating and displaying the hazard function on an odds scale:

```
> odds0<-h0/(1-h0)
> odds1<-h1/(1-h1)
> logith0<-log(odds0)
> logith1<-log(odds1)
> plot(ts0$time, odds0, type="l", ylab="Estimated Odds",
+      xlab="Grade",ylim=c(0,1), xlim=c(6, 12), col="red")
> par(new=T)
> plot(ts1$time, odds1, type="l", ylab=" ", ylim=c(0, 1),
+      xlim=c(6, 12), xlab="", col="blue")
> legend(6,0.6, c("One or more parenting transitions (PT=1)",
+ "No parenting transitions (PT=0)"),lty=1,col=c("blue","red"))
```

## Plotting on the Odds Scale



# Plotting on the Logit Scale

The odds function has some properties that are considered liabilities by some authors (see, e.g., Singer and Willett, p. 365). For example,

- 1 The odds function is bounded below by 0
- 2 The odds scale is asymmetric, in that similar differences in odds can amount to very dissimilar differences in corresponding hazard probabilities.

The log odds, or *logit* transformation, defined as

$$\text{logit } p = \log \frac{p}{1-p} \quad (2)$$

tends to eliminate or reduce these problems. Note that the logit function is invertible, i.e.,

$$p = \frac{1}{1 + \exp(-\text{logit } p)} \quad (3)$$

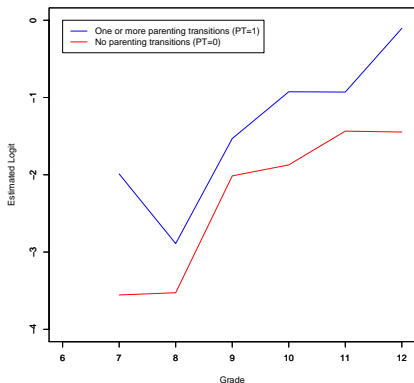


## Plotting on the Logit Scale

The code below creates the hazard plots on the logit scale, using the logits that have already been calculated in the code on a previous slide.

```
> plot(ts0$time, logith0, type="l", ylab="Estimated Logit",
+      xlab="Grade", ylim=c(-4, 0), xlim=c(6, 12), col="red")
> par(new=T)
> plot(ts1$time, logith1, type="l", ylab=" ", ylim=c(-4, 0),
+      xlim=c(6, 12), xlab="", col="blue")
> legend(6,0, c("One or more parenting transitions (PT=1)",
+ "No parenting transitions (PT=0)"),lty=1,col=c("blue","red"))
```

## Plotting on the Logit Scale



## A Discrete-Time Hazard Model

Let  $\mathbf{D}$  contain the unit-coded time variables for the time periods assessed in the study. For an observation at time  $j$ ,  $D_{ij} = 1$  and  $D_{ik} = 0$  for a time  $k$  where  $k \neq j$ . Let  $\mathbf{X}$  contain the values of the covariates that might predict hazard function differences, and let  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$  contain regression coefficients. The model for person  $i$  is

$$\text{logit } h_i = \mathbf{D}_i \boldsymbol{\alpha} + \mathbf{X}_i \boldsymbol{\beta} \quad (4)$$

## Interpreting the Model

In the preceding model, suppose that there is only one covariate  $X$  and that it is dichotomous, scored 0 or 1. If  $X = 0$ , then the vector  $\boldsymbol{\alpha}$  contains the values of logit  $\mathbf{h}$ , which may easily be transformed back to hazard probabilities using Equation 3. Singer and Willett refer to this as the “baseline” model.

What happens in the case where  $X = 1$  and is time-invariant? In that case, then logit  $\mathbf{h} = \boldsymbol{\alpha} + \boldsymbol{\beta}$ . That is, at each point in time, the logit (i.e, log-odds) of the baseline model have *the same constant added to them*. What does this imply about the ratio of the hazard odds when  $X = 1$  relative to the hazard odds when  $X = 0$ ? (C.P.)

## Proportional Hazard Odds

Let's work through step-by-step. Since when  $X = 1$ , we have, at time  $j$ ,  $\log \text{odds}_1 = \alpha_j + \beta$ , and when  $X = 0$ , we have

$\log \text{odds}_0 = \alpha_j$ , we have  $\log \left( \frac{\text{odds}_1}{\text{odds}_0} \right) = \log \text{odds}_1 - \log \text{odds}_0 = \beta$ .

Hence

$$\frac{\text{odds}_1}{\text{odds}_0} = \exp \beta \quad (5)$$

In other words, the hazard odds when  $X = 1$  are proportional at every time period to those when  $X = 0$ , and the constant of proportionality is  $\exp \beta$ . Note that, when  $\beta$  is close to 0,  $\exp \beta$  is close in value to  $1 + \beta$ , and so  $\beta$  is close to the proportional increase in the odds.

For example, if  $\beta = .05$ ,  $e^\beta = 1.0513$ , so the actual percentage increase is 5.1%, but 5% is a reasonably close approximation.

## Fitting the Model with Maximum Likelihood Estimation

Singer and Willett outline the procedure for maximum likelihood estimation on pages 381–384. We can use the R `glm` function to fit the model, using the person- period version of the data set.

```
> firstsex.pp<-read.table("firstsex_pp.csv",  
+ sep="," , header=T)
```

## Fitting the Model with Maximum Likelihood Estimation

The first “baseline” model includes only the time period. Note that we use logistic regression with no intercept.

```
> modelA<-glm(event~factor(period) - 1,  
+ family="binomial", data=firstsex.pp)
```

# Fitting the Model with Maximum Likelihood Estimation

```
> summary(modelA)
```

Call:

```
glm(formula = event ~ factor(period) - 1, family = "binomial",
    data = firstsex.pp)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.887	-0.698	-0.417	-0.294	2.514

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
factor(period)7	-2.398	0.270	-8.89	< 2e-16 ***
factor(period)8	-3.117	0.386	-8.07	7.0e-16 ***
factor(period)9	-1.720	0.222	-7.76	8.6e-15 ***
factor(period)10	-1.287	0.210	-6.13	8.6e-10 ***
factor(period)11	-1.163	0.229	-5.08	3.8e-07 ***
factor(period)12	-0.731	0.239	-3.06	0.0022 **

---

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1139.53 on 822 degrees of freedom  
 Residual deviance: 651.96 on 816 degrees of freedom  
 AIC: 664

Number of Fisher Scoring iterations: 5



## Adding PT as a Predictor

Model B adds PT as a predictor.

```
> modelB<-glm(event~factor(period) + pt - 1,  
+ family="binomial", data=firstsex.pp)
```

## Adding PT as a Predictor

```
> summary(modelB)

Call:
glm(formula = event ~ factor(period) + pt - 1, family = "binomial",
    data = firstsex.pp)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.051  -0.662  -0.441  -0.313   2.729

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
factor(period)7  -2.994     0.318   -9.43 < 2e-16 ***
factor(period)8  -3.700     0.420   -8.80 < 2e-16 ***
factor(period)9  -2.281     0.272   -8.37 < 2e-16 ***
factor(period)10 -1.823     0.258   -7.05 1.8e-12 ***
factor(period)11 -1.654     0.269   -6.15 7.9e-10 ***
factor(period)12 -1.179     0.272   -4.34 1.4e-05 ***
pt                0.874     0.217    4.02 5.9e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1139.53  on 822  degrees of freedom
Residual deviance: 634.66  on 815  degrees of freedom
AIC: 648.7

Number of Fisher Scoring iterations: 5
```

Introduction

An Example

Preliminary Analyses

Logit-Based Models for the Hazard Function

A Discrete-Time Hazard Model

**Fitting the Discrete-Time Survival Model**

Deviance-Based Hypothesis Tests

Wald  $Z$  and  $\chi^2$  Tests

Asymptotic Confidence Intervals

Computing and Plotting a Fitted Model

Model A – Baseline

Model B – Baseline + PT

Model C – Baseline + PAS

Model D – Baseline + PT + PAS

## Adding PAS to the Baseline Model

Model C adds PAS to the baseline model.

```
> modelC<-glm(event~factor(period) + pas - 1,  
+ family="binomial", data=firstsex.pp)
```

## Adding PAS to the Baseline Model

```
> summary(modelC)

Call:
glm(formula = event ~ factor(period) + pas - 1, family = "binomial",
    data = firstsex.pp)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.155  -0.636  -0.447  -0.272   2.683

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
factor(period)7  -2.465     0.274   -8.99 < 2e-16 ***
factor(period)8  -3.159     0.389   -8.12 4.6e-16 ***
factor(period)9  -1.730     0.224   -7.70 1.3e-14 ***
factor(period)10 -1.285     0.213   -6.04 1.5e-09 ***
factor(period)11 -1.136     0.232   -4.89 1.0e-06 ***
factor(period)12 -0.642     0.243   -2.64 0.00819 **
pas              0.443     0.114    3.89 0.00010 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1139.53  on 822  degrees of freedom
Residual deviance: 637.17  on 815  degrees of freedom
AIC: 651.2

Number of Fisher Scoring iterations: 5
```

## Model D – The Full Model

This model adds both PT and PAS as predictors.

```
> modelD<-glm(event~factor(period) + pt +  
+ pas - 1, family="binomial", data=firstsex.pp)
```

## Model D – The Full Model

```
> summary(modelD)

Call:
glm(formula = event ~ factor(period) + pt + pas - 1, family = "binomial",
    data = firstsex.pp)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.179  -0.618  -0.434  -0.284   2.786

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
factor(period)7  -2.893    0.321  -9.02 < 2e-16 ***
factor(period)8  -3.585    0.423  -8.47 < 2e-16 ***
factor(period)9  -2.150    0.277  -7.75 9.2e-15 ***
factor(period)10 -1.693    0.265  -6.40 1.6e-10 ***
factor(period)11 -1.518    0.276  -5.50 3.7e-08 ***
factor(period)12 -1.010    0.281  -3.59 0.00033 ***
pt                0.661    0.237   2.79 0.00527 **
pas               0.296    0.125   2.36 0.01809 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1139.53  on 822  degrees of freedom
Residual deviance: 629.15  on 814  degrees of freedom
AIC: 645.1

Number of Fisher Scoring iterations: 5
```

## Deviance-Based Hypothesis Tests

Since Models A,B,C,D are nested (in the sense that A is nested in B and C, and B and C are nested in D), we can test the significance of the coefficients for PT, and PAS, and then test whether PT adds in addition to PAS, and whether PAS adds in addition to PT, with a series of Deviance tests. Each deviance tests compares the deviance for the more restricted model (the one with fewer parameters) with the deviance for the less restricted model it is nested within. The test for significance of the parameter that differs between the two models is a chi-square with one degree of freedom.

## Deviance-Based Hypothesis Tests

For example, to test whether PT adds to the baseline model, we can use the `anova` command as follows:

```
> anova(modelA,modelB)
```

Analysis of Deviance Table

```
Model 1: event ~ factor(period) - 1
```

```
Model 2: event ~ factor(period) + pt - 1
```

	Resid. Df	Resid. Dev	Df	Deviance
1	816	652		
2	815	635	1	17.3



## Deviance-Based Hypothesis Tests

To test whether PAS adds to the baseline model, we have

```
> anova(modelA,modelC)
```

Analysis of Deviance Table

```
Model 1: event ~ factor(period) - 1
```

```
Model 2: event ~ factor(period) + pas - 1
```

	Resid. Df	Resid. Dev	Df	Deviance
1	816	652		
2	815	637	1	14.8

## Deviance-Based Hypothesis Tests

To test whether PAS adds to the baseline model once PT has been included, we have

```
> anova(modelB,modelD)
```

Analysis of Deviance Table

```
Model 1: event ~ factor(period) + pt - 1
```

```
Model 2: event ~ factor(period) + pt + pas - 1
```

	Resid. Df	Resid. Dev	Df	Deviance
1	815	635		
2	814	629	1	5.51

## Deviance-Based Hypothesis Tests

To test whether PT adds to the baseline model once PAS has been included, we have

```
> anova(modelC,modelD)
```

Analysis of Deviance Table

```
Model 1: event ~ factor(period) + pas - 1
```

```
Model 2: event ~ factor(period) + pt + pas - 1
```

	Resid. Df	Resid. Dev	Df	Deviance
1	815	637		
2	814	629	1	8.02

# Simultaneous Deviance Tests for Groups of Parameters

We can test whether two parameters together produce an improvement by comparing the model with both parameters against the baseline with neither parameter. The resulting  $\chi^2$  statistic has two degrees of freedom.

```
> anova(modelA,modelD)
```

Analysis of Deviance Table

```
Model 1: event ~ factor(period) - 1
```

```
Model 2: event ~ factor(period) + pt + pas - 1
```

	Resid. Df	Resid. Dev	Df	Deviance
1	816	652		
2	814	629	2	22.8

## Wald Tests

Wald tests in their simplest form compare a parameter estimate with an estimated standard error of the estimate, thereby yielding an asymptotic  $Z$ -statistic for testing the hypothesis that the parameter is zero.

So, for example, from the output for model B, we see an estimate of 0.8736 for the PT parameter, and an estimated standard error of 0.2174. The asymptotic  $Z$  statistic is thus 4.018, and the square of this statistic, 16.15, is a  $\chi^2$  with 1 degree of freedom, and can thus be compared directly with the corresponding deviance statistic.

The deviance and Wald statistics are reported in Table 11.3.

## Asymptotic Confidence Intervals for Parameters

In keeping with more modern views of statistical interpretation, a confidence interval for a parameter may be considerably more useful than its  $p$ -value. As usual, we construct these asymptotically normal intervals as

$$\hat{\beta} \pm Z_{1-\alpha/2}^* \hat{\sigma}(\hat{\beta}) \quad (6)$$

where  $Z^*$  is an appropriate critical value (e.g., 1.96 for a 95% confidence interval) from the standard normal distribution, and  $\hat{\sigma}(\hat{\beta})$  is the estimated standard error of the estimate  $\hat{\beta}$ .

## Asymptotic Confidence Intervals for Parameters

For example, in Table 11.3, we see that, in Model B, the parameter estimate for PT is 0.8736 and the estimated standard error is 0.2174. So the 95% confidence interval is

$$0.8736 \pm 1.96 \times 0.2174$$
$$0.8736 \pm 0.4261$$

So the confidence interval ranges from 0.4475 to 1.2997.

## Asymptotic Confidence Intervals for Odds Ratios

As we saw earlier, in the Discrete-Time survival model, a parameter value of  $\beta$  corresponds to an odds ratio of  $\exp(\beta)$ . Since the parameter and odds ratio are monotonically related, a confidence interval on one may be transformed directly into a confidence interval on the other.

Hence, in the Model B example, we might construct a 95% confidence interval on the odds-ratio for PT as ranging from  $\exp(0.4475) = 1.5644$  to  $\exp(1.2997) = 3.6682$ .

Note that an odds ratio of 1 corresponds to no effect, and the fact that the confidence interval excludes 1 indicates that the two-sided test for no effect is rejected at the .05 level.



## Computing and Plotting a Fitted Model

Often, rather than plotting the hazard or survival function directly from the life table, we plot the fitted model instead. This involves some straightforward computations from the estimated model coefficients.

## Computing and Plotting a Fitted Model

Recall that the basic model as shown in Equation 4 is  $\text{logit } h_i = \mathbf{D}_i \boldsymbol{\alpha} + \mathbf{X}_i \boldsymbol{\beta}$ . The logit function is invertible, and so

$$h_i = \text{logit}^{-1}(\mathbf{D}_i \boldsymbol{\alpha} + \mathbf{X}_i \boldsymbol{\beta}) \quad (7)$$

At time  $j$ , the fitted model therefore has hazard  $\text{logit}^{-1}(\alpha_j + \sum_k X_{ik} \beta_k)$ , which is equal to

$$\frac{1}{1 + \exp -(\alpha_j + \sum_k X_{ik} \beta_k)} \quad (8)$$

The fitted values of the hazard for a given model may then be converted into fitted values for the survival function by use of the product-limit formula.

## Computing the Hazard Function – Model A

Here is an example computing the fitted odds and fitted hazard, using the output from Model A. Notice that we define a function, `inverse.logit`, that will be useful in subsequent calculations.

```
> inverse.logit <- function(x){  
+   return(1/(1+exp(-x)))  
+ }  
> modelA<-glm(event~factor(period) - 1, family="binomial", data=firstsex.pp)  
> col0<-c(7:12)  
> col1<-c("D7", "D8", "D9", "D10", "D11", "D12")  
> col2<-exp(modelA$coefficients)  
> col3<- inverse.logit(modelA$coefficients)  
> tab11.4<-data.frame(time=col0, Predictor=col1, parameter=modelA$coefficients,  
+   fitted.odds=col2, fitted.hazard=col3, row.names=NULL)
```

# Computing the Hazard Function – Model A

```
> tab11.4
```

	time	Predictor	parameter	fitted.odds	fitted.hazard
1	7	D7	-2.3979	0.09091	0.08333
2	8	D8	-3.1167	0.04430	0.04242
3	9	D9	-1.7198	0.17910	0.15190
4	10	D10	-1.2867	0.27619	0.21642
5	11	D11	-1.1632	0.31250	0.23810
6	12	D12	-0.7309	0.48148	0.32500

## Computing the Fitted Hazard and Survival Functions – Model B

The next example is somewhat more ambitious. We calculate the fitted values for logit hazard, hazard, and survival, for the case where  $PT = 0$  and  $PT = 1$ .

Note that, in this special case of a single dichotomous 0-1 variable, we have, as explained before, fitted values as  $\text{logit } h_j = \alpha_j + \beta PT$ ,  $h_j = \text{logit}^{-1}(\alpha_j + \beta PT)$ , and  $S_j = S_{j-1}(1 - h_j)$ , with  $S_6 = 1$ .

# Computing the Fitted Hazard and Survival Functions – Model B

Here is the code:

```
> modelB<-glm(event~factor(period) + pt - 1, family="binomial", data=firstsex.pp)
> t<-data.frame(hazard=modelB$fitted.values, time=firstsex.pp$period, pt=firstsex.pp$pt)
> t$logit<-log(t$hazard/(1-t$hazard))
> ta<-aggregate(t, list(pt=t$pt, time=t$time),mean)
> ta.0<-ta[ta$pt==0, ]
> ta.1<-ta[ta$pt==1, ]
> c1<-c(7:12)
> c2<-ta.0$logit
> c3<-ta.1$logit-ta.0$logit
> c4<-ta.0$hazard
> c5<-ta.1$hazard
> c6<-ta.0$hazard
> c7<-ta.1$hazard
> tab11.5<-data.frame(time=c1, alpha=c2, beta=c3, logit_0 = c4, logit_1= c5,
+ hazard_0 = c6, hazard_1 = c7)
> tab11.5$surv_0<-0
> tab11.5$surv_1<-0
> tab11.5$surv_0[i]<-1-tab11.5$hazard_0[i]
> tab11.5$surv_1[i]<-1-tab11.5$hazard_1[i]
> for(i in 2:6) {
+ tab11.5$surv_0[i] = tab11.5$surv_0[i-1]*(1-tab11.5$hazard_0[i])
+ tab11.5$surv_1[i] = tab11.5$surv_1[i-1]*(1-tab11.5$hazard_1[i])
+ }
```

# Computing the Fitted Hazard and Survival Functions – Model B

```
> tab11.5
```

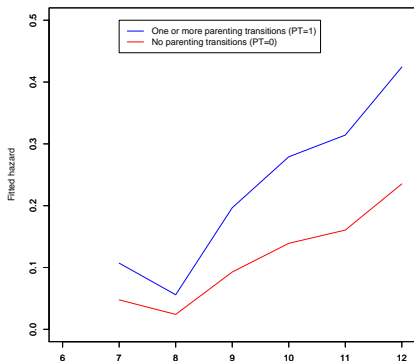
	time	alpha	beta	logit_0	logit_1	hazard_0	hazard_1	surv_0	surv_1
1	7	-2.994	0.8736	-2.994	-2.1207	0.04768	0.10710	0.9523	0.8929
2	8	-3.700	0.8736	-3.700	-2.8265	0.02412	0.05591	0.9293	0.8430
3	9	-2.281	0.8736	-2.281	-1.4075	0.09270	0.19663	0.8432	0.6772
4	10	-1.823	0.8736	-1.823	-0.9490	0.13912	0.27909	0.7259	0.4882
5	11	-1.654	0.8736	-1.654	-0.7806	0.16054	0.31419	0.6094	0.3348
6	12	-1.179	0.8736	-1.179	-0.3054	0.23522	0.42423	0.4660	0.1928

## Plotting the Fitted Hazard Function

```
> plot(tab11.5$time, tab11.5$hazard_0, type="l",  
+       ylab="Fitted hazard", ylim=c(0, 0.5),  
+       xlim=c(6, 12), xlab="", col="red")  
> par(new=T)  
> plot(tab11.5$time, tab11.5$hazard_1, type="l", ylab="",  
+       ylim=c(0, 0.5), xlim=c(6, 12), xlab="", col="blue")  
> legend(7,0.5, c("One or more parenting transitions (PT=1)",  
+ "No parenting transitions (PT=0)"),lty=1,col=c("blue","red"))
```



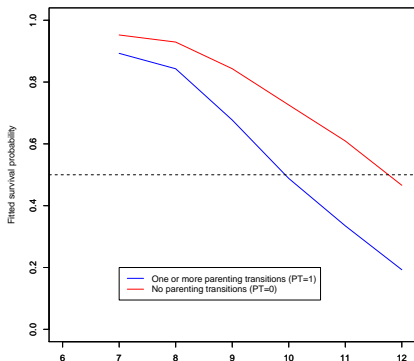
## Plotting the Fitted Hazard Function



## Plotting the Fitted Survival Function

```
> plot(tab11.5$time, tab11.5$urv_0, type="l",  
+       ylab="Fitted survival probability",  
+       ylim=c(0, 1), xlim=c(6, 12), xlab="", col="red")  
> par(new=T)  
> plot(tab11.5$time, tab11.5$urv_1, type="l", ylab="", ylim=c(0,1),  
+       xlim=c(6, 12), xlab="", col="blue")  
> abline(h=c(.5), lty=2)  
> legend(7,0.2, c("One or more parenting transitions (PT=1)",  
+ "No parenting transitions (PT=0)"),lty=1,col=c("blue","red"))
```

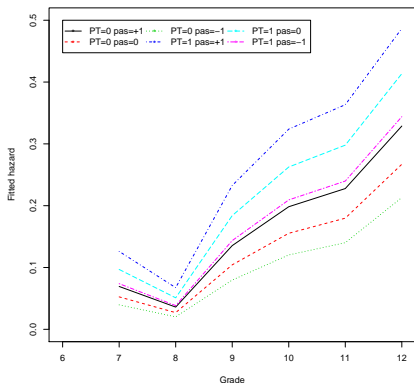
## Plotting the Fitted Survival Function



## Plotting the Fitted Hazard – Model D

```
> modelD<-glm(event~factor(period) + pt + pas - 1, family="binomial", data=firstsex.pp)
> coeff<-data.frame(modelD$coefficients)
> myt<-c(1:6)
> h0_pas1<-c(1:6)
> h0_pas0<-c(1:6)
> h0_pasn1<-c(1:6)
> h1_pas1<-c(1:6)
> h1_pas0<-c(1:6)
> h1_pasn1<-c(1:6)
> for(i in 1:6) {
+ myt[i]<-i+6
+ h0_pas1[i]<-1/(1+ exp(-(coeff[i,] + coeff[8,])))
+ h0_pas0[i]<-1/(1+ exp(-coeff[i,]))
+ h0_pasn1[i]<-1/(1+ exp(-(coeff[i,] - coeff[8,])))
+ h1_pas1[i]<-1/(1+ exp(-(coeff[i,] + coeff[8,] + coeff[7,])))
+ h1_pas0[i]<-1/(1+ exp(-(coeff[i,] + coeff[7,])))
+ h1_pasn1[i]<-1/(1+ exp(-(coeff[i,] - coeff[8,] + coeff[7,])))
+ }
> f<-cbind(h0_pas1,h0_pas0,h0_pasn1, h1_pas1,h1_pas0,h1_pasn1)
> matplot(myt, f, type="l", ylab="Fitted hazard", ylim=c(0, 0.5), xlim=c(6, 12),
+ xlab="Grade", col=1:6, lty=1:6)
> legend(6, .5, c("PT=0 pas=+1", "PT=0 pas=0", "PT=0 pas=-1",
+ "PT=1 pas=+1", "PT=1 pas=0", "PT=1 pas=-1"),
+ col=1:6, lty=1:6, pch = "*",
+ ncol =3, cex = 1)
```

## Plotting the Fitted Hazard – Model D



## Plotting the Fitted Survivor Function – Model D

```
> surv0_pas1<-c(1:6)
> surv0_pas0<-c(1:6)
> surv0_pasn1<-c(1:6)
> surv1_pas1<-c(1:6)
> surv1_pas0<-c(1:6)
> surv1_pasn1<-c(1:6)
> surv0_pas1[1]<-1-h0_pas1[1]
> surv0_pas0<-1-h0_pas0[1]
> surv0_pasn1<-1-h0_pasn1[1]
> surv1_pas1<-1-h1_pas1[1]
> surv1_pas0<-1-h1_pas0[1]
> surv1_pasn1<-1-h1_pas1[1]
> for(i in 2:6) {
+   surv0_pas1[i] = surv0_pas1[i-1]*(1-h0_pas1[i])
+   surv0_pas0[i] = surv0_pas0[i-1]*(1-h0_pas0[i])
+   surv0_pasn1[i] = surv0_pasn1[i-1]*(1-h0_pasn1[i])
+   surv1_pas1[i] = surv1_pas1[i-1]*(1-h1_pas1[i])
+   surv1_pas0[i] = surv1_pas0[i-1]*(1-h1_pas0[i])
+   surv1_pasn1[i] = surv1_pasn1[i-1]*(1-h1_pasn1[i])
+ }
> s<-cbind(surv0_pas1,surv0_pas0,surv0_pasn1,surv1_pas1,surv1_pas0,surv1_pasn1)
> matplot(myt, s, type="l", ylab="Fitted survival probability", ylim=c(0, 1), xlim=c(6, 12),
+   xlab="Grade", col=1:6, lty=1:6)
> abline(h=c(.5), lty=2)
> legend(6, .2, c("PT=0 pas=+1", "PT=0 pas=0", "PT=0 pas=-1",
+   "PT=1 pas=+1", "PT=1 pas=0", "PT=1 pas=-1"),
+   col=1:6, lty=1:6, pch = "*",
+   ncol = 2, cex = 1)
```

# Plotting the Fitted Survivor Function – Model D

