

STATISTICS: AN INTRODUCTION USING R

By M.J. Crawley

Exercises

12. SURVIVAL ANALYSIS

A great many studies in statistics deal with deaths or with failures of components: the numbers of deaths, the timing of death, and the risks of death to which different classes of individuals are exposed. The analysis of survival data is a major focus of the statistics business (see Kalbfleisch & Prentice 1980, Miller 1981, Fleming & Harrington 1991), and S-Plus supports a wide range of tools for the analysis of survival data. The main theme of this chapter is the analysis of data that take the form of measurements of the *time to death*, or the *time to failure* of a component. Up to now, we have dealt with mortality data by considering the proportion of individuals that were dead *at a given time*. In this chapter each individual is followed until it dies, then the time of death is recorded (this will be the response variable). Individuals that survive to the end of the experiment will die at an unknown time in the future; they are said to be *censored* (see below).

A Monte Carlo experiment

With data on time-to-death, the most important decision to be made concerns the error distribution. The key point to understand is that the variance in age at death is almost certain to increase with the mean, and hence standard models (assuming constant variance and normal errors) will be inappropriate. You can see this at once with a simple Monte Carlo experiment. Suppose that the per-week probability of failure of a component is 0.1 from one factory but 0.2 from another. We can simulate the fate of an individual component in a given week by generating a uniformly distributed random number between 0 and 1. If the value of the random number is less than or equal to 0.1 (or 0.2 for the second factory) the component fails during that week and its lifetime can be calculated. If the random number is larger than 0.1 the component survives to the next week. The lifetime of the component is simply the number of the week in which it finally failed. Thus, a component that failed in the first week has an age at failure of 1 (this convention means that there are no zeros in the data frame).

The simulation is very simple. We create a vector of random numbers, *rnos*, that is long enough to be almost certain to contain a value that is less than our failure probabilities of 0.1 and 0.2. Remember that the mean life expectancy is the reciprocal of the failure rate, so our mean lifetimes will be $1/0.1 = 10$ and $1/0.2 = 5$ weeks respectively. A length of 100 should be more than sufficient.

```
rnos<-runif(100)
```

The trick is to find the week number in which the component failed; this is the lowest subscript for which *rnos* ≤ 0.1 for factory 1. We can do this very efficiently using the **which** function: **which** returns *a vector of subscripts* for which the specified logical condition is true. So for factory 1 we would write

```
which(rnos<= 0.1)
```

```
[1] 5 8 9 19 29 33 48 51 54 63 68 74 80 83 94 95
```

This means that for my first set of 100 random numbers, 16 of them were less than or equal to 0.1. The important point is that the *first* such number occurred in week 5. So the simulated value of the age of death of this first component is 5 and is obtained from the vector of failure ages using the subscript [1]

```
which(rnos<= 0.1)[1]
```

```
[1] 5
```

All we need to do to simulate the life spans of a sample of 20 components, death1, is to repeat the above procedure 20 times

```
death1<-numeric(20)
```

```
for (i in 1:20) {  
  rnos<-runif(100)  
  death1[i]<- which(rnos<= 0.1)[1]  
}
```

```
death1
```

```
[1] 5 8 12 23 11 3 8 3 12 13 1 5 9 1 7 9 11 1 2 8
```

The 4th component survived for a massive 23 weeks but the 11th component failed during its first week. The simulation has roughly the right average weekly failure rate:

```
1/mean(death1)
```

```
[1] 0.1315789
```

which is as close to 0.1 as we could reasonably expect from a sample of only 20 components. Now we do the same for the second factory with its failure rate of 0.2:

```
death2<-numeric(20)
```

```
for (i in 1:20) {
```

```
rnos<-runif(100)
death2[i]<- which(rnos<= 0.2)[1]
}
```

The sample mean is again quite reasonable (if a little on the low side):

```
1/mean(death2)
```

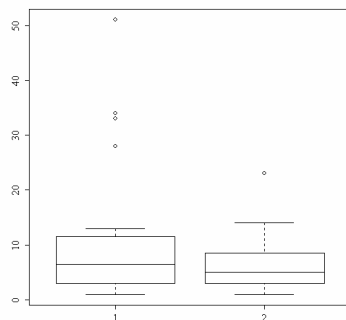
```
[1] 0.1538462
```

We now have the simulated raw data to carry out a comparison in age at death between factories 1 and 2. We combine the two vectors into one, and generate a vector to represent the factory identities:

```
death<-c(death1,death2)
factory<-factor(c(rep(1,20),rep(2,20)))
```

We get a visual assessment of the data using plot

```
plot(factory,death)
```



The median age at death for factory 1 is somewhat greater, but the variance in age a death is much higher than from factory 2. For data like this we expect the variance to be proportional to the square of the mean, so an appropriate error structure is the gamma (as explained below). We model the data very simply as a one-way analysis of deviance with a **glm** of family = Gamma (note the upper case G)

```
model1<-glm(death~factory,Gamma)
summary(model1)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.08474	0.01861	4.554	5.29e-005 ***
factory	0.06911	0.03884	1.779	0.0832 .

(Dispersion parameter for Gamma family taken to be 0.983125)

```
Null deviance: 37.175 on 39 degrees of freedom
Residual deviance: 33.670 on 38 degrees of freedom
```

We conclude that the factories are not significantly different in mean age at failure of these components ($p = 0.0832$). So, even with a 2-fold difference in the true failure rate, we are unable to detect a significant difference in mean age at death with samples of size $n = 20$. But the **glm** with gamma errors comes closer to detecting the difference than did a conventional analysis with normal errors and constant variance (see if you can work out how to demonstrate this: for my data, $p = 0.113$ with the inappropriate anova). The moral is that *for data like this on age at death you are going to need really large sample sizes in order to find significant differences.*

```
rm(death)
```

Background

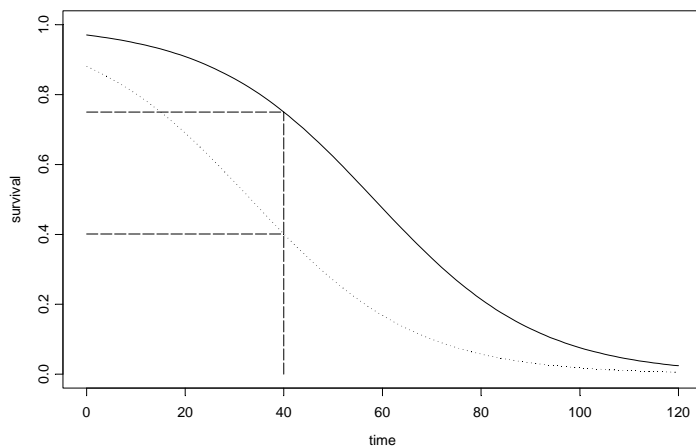
We are unlikely to know much about the error distribution in advance of the study, except that it will certainly not be normal! In S-Plus we are offered several choices for the analysis of survival data:

- gamma
- exponential
- piece-wise exponential
- extreme value
- log-logistic
- log normal
- Weibull

and, in practice, it is often difficult to choose between them. In general, the best solution is to try several distributions and to pick the error structure that produces the minimum error deviance. Parametric survival models are used in circumstances where prediction is the object of the exercise (e.g. in analyses where extreme conditions are used to generate accelerated failure times). Alternatively, we could use nonparametric methods, the most important of which are Kaplan-Meier and Cox proportional hazards, which are excellent for comparing the effects of different treatments on survival, but they do not predict beyond the last observation and hence can not be used for extrapolation.

Some theoretical demography

Since everything dies eventually, it is often not possible to analyse the results of survival experiments in terms of the proportion that were killed (as we did in Practical 10); in due course, they *all* die. Look at the following figure:

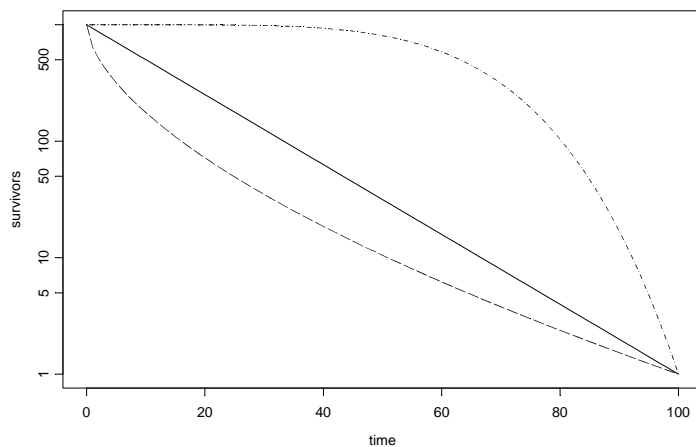


It is clear that the two treatments caused different patterns of mortality, but both start out with 100% survival and both end up with zero. We could pick some arbitrary point in the middle of the distribution at which to compare the percentage survival (say at time = 40), but this may be difficult in practice, because one or both of the treatments might have few observations at the same location. Also, the choice of when to measure the difference is entirely subjective and hence open to bias. It is much better to use S-Plus's powerful facilities for the analysis of survival data than it is to pick an *arbitrary* time at which to compare two proportions.

Demographers, actuaries and ecologists use three interchangeable concepts when dealing with data on the timing of death:

- survivorship
- age at death
- instantaneous risk of death

There are 3 broad patterns of survivorship. Type I: most of the mortality occurs late in life (e.g. humans); Type II: mortality occurs at a roughly constant rate throughout life; Type III: most of the mortality occurs early in life (e.g. salmonid fishes). On a log scale, the numbers surviving from an initial cohort of 1000, say, would look like this:



1) The survivor function

The survivorship curve plots the natural log of the proportion of a cohort of individuals that started out at time 0 that is still alive at time t . For the so-called Type II survivorship curve, there is a linear decline in log numbers with time. This means that a constant proportion of the individuals alive at the beginning of a time interval will die during that time interval (i.e. the proportion dying is density independent and constant for all ages). When the death rate is highest for the younger age classes we get a steeply descending, Type III survivorship curve. When it is the oldest animals that have the highest risk of death, we obtain the Type I curve (characteristic of human populations in affluent societies where there is low infant mortality).

2) The density function

The density function describes the fraction of all deaths from our initial cohort that are likely to occur in a given instant of time. For the Type II curve this is a negative exponential. Because the fraction of individuals dying is constant with age, the number dying declines exponentially as the number of survivors (the number of individuals at risk of death) declines exponentially with the passage of time. The density function declines more steeply than exponentially for Type III survivorship curves. In the case of Type I curves, however, the density function has a maximum at the time when the product of the risk of death and the number of survivors is greatest (see below).

3) The hazard function

The hazard is the instantaneous risk of death; i.e. the derivative of the survivorship curve. It is the instantaneous rate of change in the log of the number of survivors per unit time (it is the slope of the survivorship curves). Thus, for the Type II survivorship the hazard function is a horizontal line, because the risk of death is constant with age. Although this sounds highly unrealistic, it is a remarkably robust assumption in many applications. It also has the substantial advantage of parsimony. In some cases, however, it is clear that the risk of death changes substantially with the age of the individuals, and we need to be

able to take this into account in carrying out our statistical analysis. In the case of Type III survivorship, the risk of death declines with age, while for Type I survivorship (as in humans) the risk of death increases with age.

The Exponential Distribution

This is a 1-parameter distribution in which the hazard function is independent of age (i.e. it describes a Type II survivorship curve). The exponential is a special case of the gamma distribution in which the shape parameter α is equal to 1 .

Density function

The density function is the probability of dying in the small interval of time between t and $t+dt$; a plot of the number dying in the interval around time t as a function of t (i.e. the proportion of the original cohort dying at a given age) declines exponentially:

$$f(t) = \frac{e^{-\frac{t}{\mu}}}{\mu}$$

where both μ and $t > 0$. Note that the density function has an intercept of $1/\mu$ (remember that e^0 is 1). The probability of dying declines exponentially with time and a fraction $1/\mu$ dies during the first time interval (and, indeed, during every subsequent time interval).

Survivor function

This shows the proportion of individuals from the initial cohort still alive at time t :

$$S(t) = e^{-\frac{t}{\mu}}$$

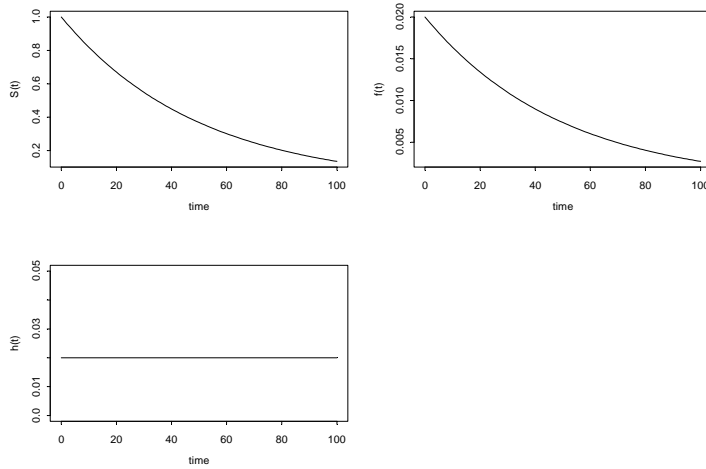
The survivor function has an intercept of 1 (i.e. all the cohort is alive at time 0), and shows the probability of surviving at least as long as t .

Hazard function

This is the statisticians equivalent of the ecologist's *instantaneous death rate*. It is defined as the ratio between the density function and the survivor function, and is the conditional density function at time t , given survival up to time t . In the case of Type II curves this has an extremely simple form:

$$h(t) = \frac{f(t)}{S(t)} = \frac{e^{-t/\mu}}{\mu e^{-t/\mu}} = \frac{1}{\mu}$$

because the exponential terms cancel out. Thus, with the exponential distribution the *hazard is the reciprocal of the mean time to death*, and vice versa. For example, if the mean time to death is 3.8 weeks, then the hazard is 0.2632; if the hazard were to increase to 0.32, then the mean time of death would decline to 3.125 weeks.

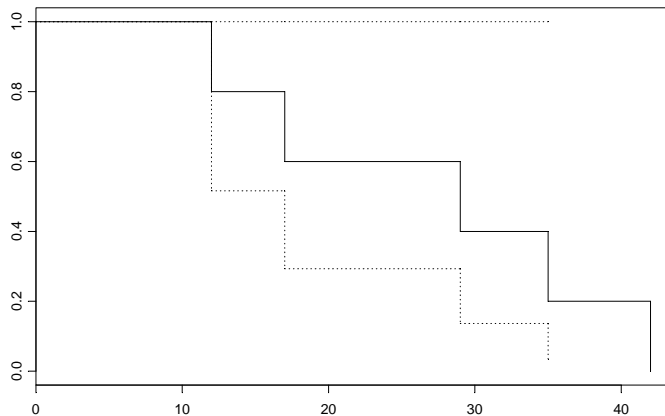


These are the **survivor**, **density** and **hazard** functions of the **exponential distribution**.

Of course, the death rate may not be a linear function of age. For example, the death rate may be high for very young as well as for very old animals, in which case the survivorship curve is like an S-shape on its side.

Kaplan-Meier survival distributions

This is a discrete stepped survivorship curve that adds information as each death occurs. Suppose the times at death were 12, 17, 29, 35 and 42 weeks after the beginning of a trial. Survivorship is 1 at the outset, and stays at 1 until time 12, when it steps down to $4/5 = 0.8$. It stays level until time 17 when it drops to $0.8 \times 3/4 = 0.6$. Then there is a long level period until time 29, when survivorship drops to $0.6 \times 2/3 = 0.4$, then drops at time 35 to $0.4 \times 1/2 = 0.2$ then to zero at time 42.



In general, therefore, we have two groups at any one time: the number of deaths $d(t_i)$ and the number at risk $r(t_i)$ (i.e. those that have not died yet: the survivors). The Kaplan-Meier survivor function is

$$\hat{S}_{KM} = \prod_{t_i < t} \frac{r(t_i) - d(t_i)}{r(t_i)}$$

which as we have seen, produces a step at every time at which one or more deaths occurs. The censored individuals that survive beyond the end of the study are shown by a + on the plot or after their age in the data frame (thus 65 means died at time 65, but 65+ means still alive when last seen at age 65).

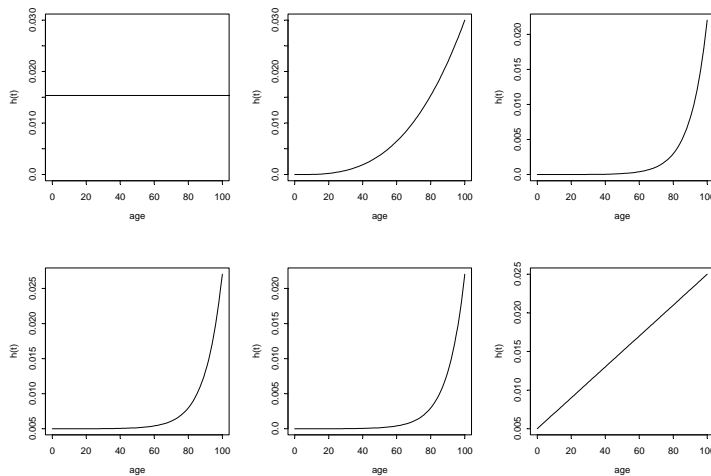
Age-specific hazard models

In many circumstances, the risk of death increases with age. There are many models to choose from:

Distribution	Hazard
Exponential	constant = $\frac{1}{\mu}$
Weibull	$\alpha\lambda(\lambda t)^{\alpha-1}$
Gompertz	be^{ct}
Makeham	$a + be^{ct}$
Extreme value	$\frac{1}{\sigma} e^{(t-\eta)/\sigma}$
Rayleigh	$a + bt$

The Rayleigh is obviously the simplest model in which hazard increases with time, but the Makeham is widely regarded as the best description of hazard for human subjects. Post infancy, there is a constant hazard (a) which is due to age independent accidents,

murder, suicides, etc., with an exponentially increasing hazard in later life. The Gompertz assumption was that “the average exhaustion of a man’s power to avoid death is such that at the end of equal infinitely small intervals of time he has lost equal portions of his remaining power to oppose destruction which he had at the commencement of these intervals”. Note that the Gompertz differs from the Makeham only by the omission of the extra background hazard (a), and this becomes negligible in old age.



These plots show how hazard changes with age for the following distributions: from top left to bottom right: **exponential, Weibull, Gompertz, Makeham, Extreme value and Rayleigh**.

Survival analysis in R

There are three cases that concern us here:

- constant hazard and no censoring
- constant hazard with censoring
- age-specific hazard, with or without censoring

The first case is dealt with in R by specifying a **glm** with exponential errors. This involves using gamma errors with the scale factor fixed at 1 .

The second case involves the use of a **glm** with Poisson errors and a log link, where *the censoring indicator is the response variable*, and $\log(\text{time of death})$ is an offset (see below).

The third case is the one that concerns us mainly in this Practical. We can choose to use **parametric** models, based round the **Weibull** distribution, or **non parametric** techniques, based round **Cox proportional hazard** model.

Cox proportional hazards model

This is the most widely used regression model for survival data. It assumes that the hazard is of this form

$$\lambda(t; Z_i) = \lambda_0(t) r_i(t)$$

where $Z_i(t)$ is the set of explanatory variables for individual i at time t . The *risk score* for subject i is

$$r_i(t) = e^{\beta Z_i(t)}$$

in which β is a vector of parameters from the linear predictor and $\lambda_0(t)$ is an *unspecified baseline hazard function* that will cancel out in due course. The antilog guarantees that λ is positive for any regression model $\beta Z_i(t)$. If a death occurs at time t^* , then conditional on this death occurring, the likelihood that it is individual i that dies rather than any other individual at risk, is

$$L_i(\beta) = \frac{\lambda_0(t^*) r_i(t^*)}{\sum_j Y_j(t^*) \lambda_0(t^*) r_j(t^*)} = \frac{r_i(t^*)}{\sum_j Y_j(t^*) r_j(t^*)}$$

The product of these terms over all times of death $L(\beta) = \prod L_i(\beta)$ was christened a partial likelihood by Cox (1972). This is clever, because maximising $\log(L(\beta))$ allows an estimate of β without knowing anything about the baseline hazard function ($\lambda_0(t)$ is a nuisance variable in this context). The proportional hazards model is nonparametric in the sense that it depends only on the **ranks** of the survival times.

An example of survival analysis without censoring

To see how the exponential distribution is used in modelling we take an example from plant ecology, in which individual seedlings were followed from germination until death. We have the times to death measured in weeks for two cohorts, each of 30 seedlings. The plants were germinated at two times (cohorts), in early September (treatment 1) and mid October (treatment 2). We also have data on the size of the gap into which each seed was sown (a covariate x). The questions are these:

- is an exponential distribution suitable to describe these data?
- was survivorship different between the 2 planting dates?
- did gap size affect the time to death of a given seedling?

```
seedlings<-read.table("c:\\temp\\seedlings.txt",header=T)
attach(seedlings)
names(seedlings)
```

```
[1] "cohort" "death" "gapsize"
```

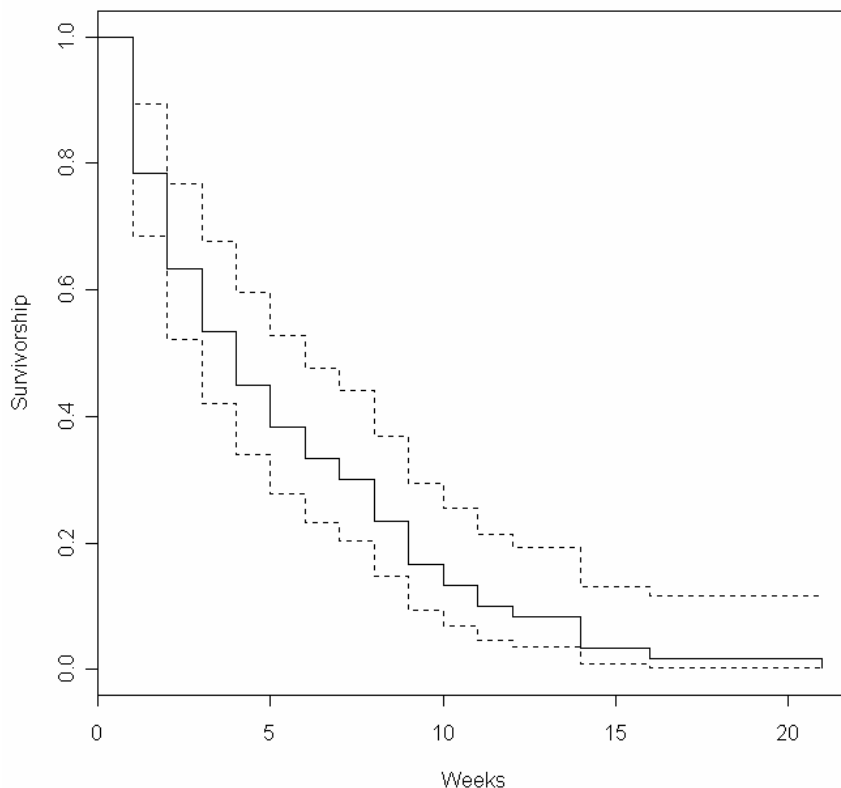
We need to open the library of survival analysis functions:

```
library(survival)
```

There are several important functions for plotting and analysing survival data. The function **Surv** (note the capital S) takes 2 vectors. The first contains the time (or age) at which the individual was last seen, and the second indicates the status of that individual (i.e. whether it was dead or alive when it was last seen: 1 = dead, 0 = alive). Individuals are said to be censored if they were alive at the time they were last seen. All the seedlings died in this example, so status = 1 for all the cases. The function **survfit** calculates the survivorship curve on the basis of the age at death and censoring information in **Surv**. Use of **plot** with **survfit** as its argument produces a graph of the survivorship curve.

```
status<-1*(death>0)
```

```
plot(survfit(Surv(death,status)),ylab="Survivorship",xlab="Weeks")
```



This shows the survivorship of seedlings over the 20 weeks of the study period until the last of the seedlings died in week 21. The dotted lines show the confidence limits and are the default when only 1 survivorship curve is plotted. No axis labels are plotted unless we provide them (as here).

Statistical modelling is extremely straightforward, but somewhat limited, because interaction effects and continuous explanatory variables are not allowed with **survfit** (but see below). We begin with a simple model for the effect of cohort on its own:

```
model1<-survfit(Surv(death,status)~cohort)
```

Just typing the name of the model object produces a useful summary of the two survivorship curves, their means, standard errors and 95% confidence intervals for the age at death:

```
model1
```

```
Call: survfit(formula = Surv(death, status) ~ cohort)
```

	n	events	mean	se(mean)	median	0.95LCL	0.95UCL
cohort=October	30	30	5.83	0.903	5	3	9
cohort=September	30	30	4.90	0.719	4	2	7

Survival in the two cohorts is clearly not significantly different (just look at the overlap in the confidence intervals for median age at death). Using the **summary** function produces fully documented survival schedules for each of the two cohorts:

```
summary(model1)
```

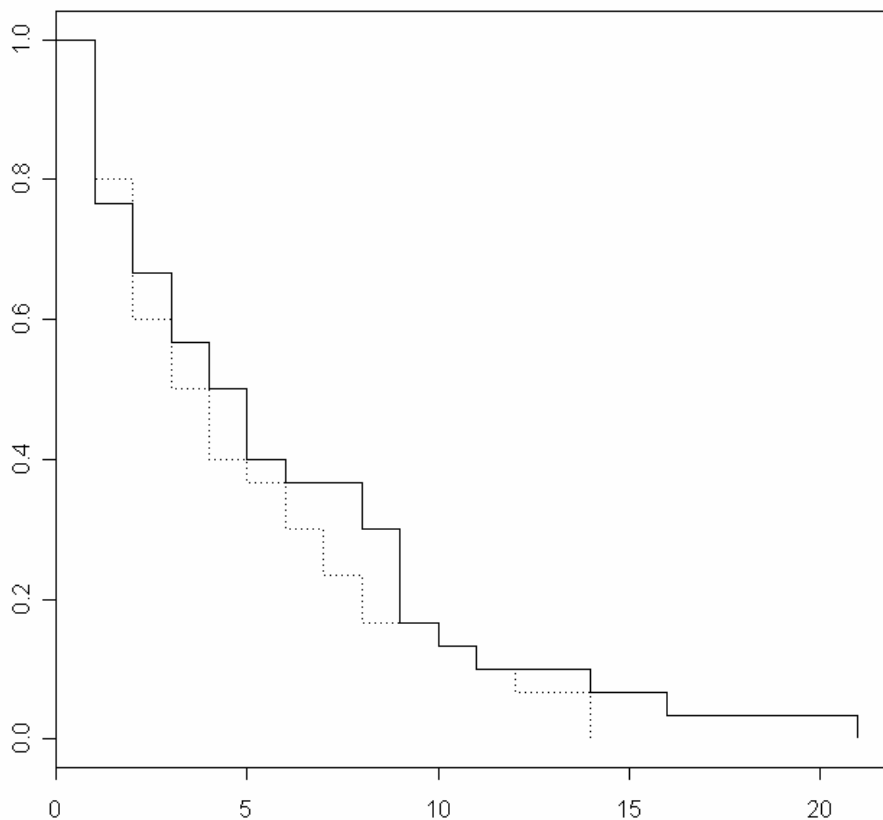
cohort=October								
time	n.risk	n.event	survival	std.err	lower	95% CI	upper	95% CI
1	30	7	0.7667	0.0772		0.62932		0.934
2	23	3	0.6667	0.0861		0.51763		0.859
3	20	3	0.5667	0.0905		0.41441		0.775
4	17	2	0.5000	0.0913		0.34959		0.715
5	15	3	0.4000	0.0894		0.25806		0.620
6	12	1	0.3667	0.0880		0.22910		0.587
8	11	2	0.3000	0.0837		0.17367		0.518
9	9	4	0.1667	0.0680		0.07488		0.371
10	5	1	0.1333	0.0621		0.05355		0.332
11	4	1	0.1000	0.0548		0.03418		0.293
14	3	1	0.0667	0.0455		0.01748		0.254
16	2	1	0.0333	0.0328		0.00485		0.229
21	1	1	0.0000	NA		NA		NA

cohort=September								
time	n.risk	n.event	survival	std.err	lower	95% CI	upper	95% CI
1	30	6	0.8000	0.0730		0.6689		0.957
2	24	6	0.6000	0.0894		0.4480		0.804
3	18	3	0.5000	0.0913		0.3496		0.715
4	15	3	0.4000	0.0894		0.2581		0.620

5	12	1	0.3667	0.0880	0.2291	0.587
6	11	2	0.3000	0.0837	0.1737	0.518
7	9	2	0.2333	0.0772	0.1220	0.446
8	7	2	0.1667	0.0680	0.0749	0.371
10	5	1	0.1333	0.0621	0.0535	0.332
11	4	1	0.1000	0.0548	0.0342	0.293
12	3	1	0.0667	0.0455	0.0175	0.254
14	2	2	0.0000	NA	NA	NA

Using the **plot** function produces a set of survivorship curves. Note the use of the vector of different line types `lty=c(1,3)` for plotting

```
plot(model1,lty=c(1,3))
```



This produces a plot of the two survivorship curves using line types 1 and 3 for the October and September cohorts respectively (the factor levels are in alphabetic order, as usual). Note that the axes are unlabelled unless you specify **xlab** and **ylab**.

To investigate the effects of a continuous explanatory variable like gap size we use Cox proportional hazards to give the survivorship curve at the average gap size:

```
model2<-survfit(coxph(Surv(death,status)~gapsize))
```

Typing the model name alone gives this:

`model2`

Call: `survfit.coxph(object = coxph(Surv(death, status) ~ gapsize))`

n	events	mean	se(mean)	median	0.95LCL	0.95UCL
60	60	5.49	0.619	4	3	6

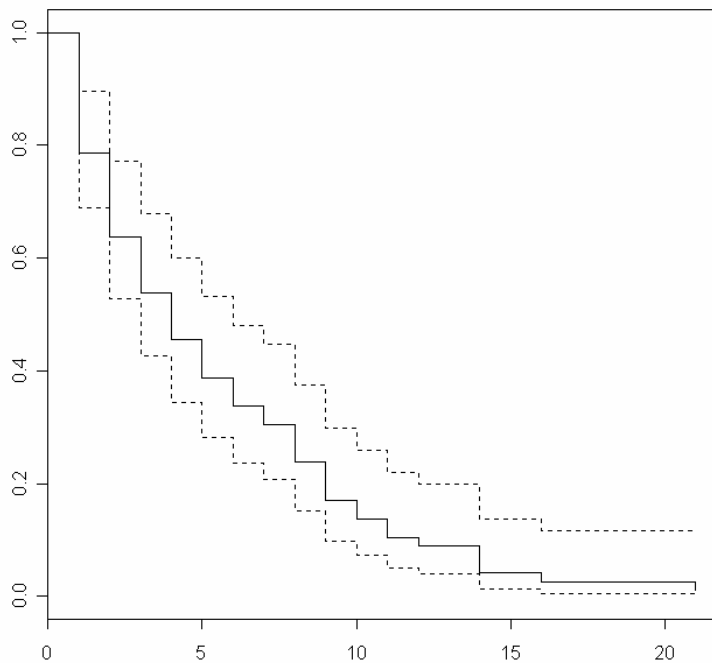
Using the summary function gives the full survival table and confidence intervals

`summary(model2)`

Call: `survfit.coxph(object = coxph(Surv(death, status) ~ gapsize))`

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
1	60	13	0.78600	0.0527	0.689271	0.896
2	47	9	0.63769	0.0618	0.527386	0.771
3	38	6	0.53812	0.0641	0.426011	0.680
4	32	5	0.45467	0.0641	0.344883	0.599
5	27	4	0.38793	0.0628	0.282503	0.533
6	23	3	0.33807	0.0609	0.237442	0.481
7	20	2	0.30474	0.0593	0.208086	0.446
8	18	4	0.23773	0.0549	0.151129	0.374
9	14	4	0.17098	0.0487	0.097862	0.299
10	10	2	0.13785	0.0446	0.073147	0.260
11	8	2	0.10501	0.0396	0.050169	0.220
12	6	1	0.08885	0.0366	0.039606	0.199
14	5	3	0.04132	0.0252	0.012487	0.137
16	2	1	0.02565	0.0199	0.005607	0.117
21	1	1	0.00935	0.0119	0.000771	0.113

`plot(model2)`



To test for differences in baseline survival for each cohort, we use **survdif** like this:

```
survdif(Surv(death,status)~cohort)
```

Call:

```
survdif(formula = Surv(death, status) ~ cohort)
```

	N	Observed	Expected	(O-E) ² /E	(O-E) ² /V
cohort=October	30	30	32.9	0.259	0.722
cohort=September	30	30	27.1	0.315	0.722

Chisq= 0.7 on 1 degrees of freedom, p= 0.395

The baseline survival is not significantly different for the two cohorts.

For a full analysis of covariance, fitting gap size separately for each cohort we use the **strata** option in the model formula with Cox proportional hazards:

```
coxph(Surv(death,status)~strata(cohort)*gapsize)
```

Call:

```
coxph(formula = Surv(death, status) ~ strata(cohort) * gapsize)
```

coef	exp(coef)	se(coef)	z	p
------	-----------	----------	---	---

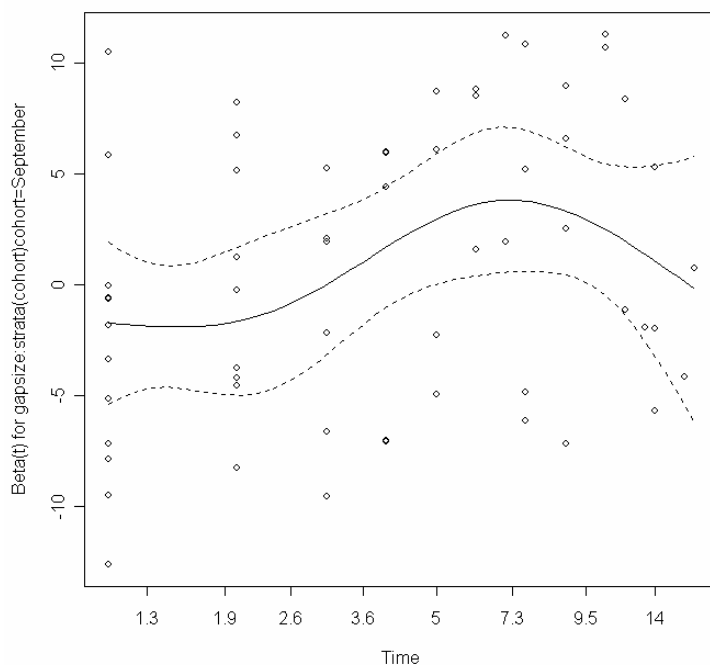
gapsize	-0.00189	0.998	0.593	-0.00319	1.0
gapsize:strata(cohort)Septembr	0.71741	2.049	0.861	0.83341	0.4

Likelihood ratio test=1.35 on 2 df, p=0.51 n= 60

Gapsize has no effect on survival in either cohort.

To test whether the coefficients are a function of time, we use the **cox.zph** function

```
model3<-cox.zph(coxph(Surv(death,status)~strata(cohort)*gapsize))
plot(model3)
```



There is no evidence of temporal changes in the parameters (you can easily fit a horizontal line between the confidence intervals). This is a truly dull data set. Nothing is happening at all.

detach(seedlings)

Censoring

Censoring occurs when we do not know the time to death for all of the individuals. This comes about principally because some individuals outlive the experiment. We can say they survived for the duration of the study but we have no way of knowing at what age they will die. These individuals contribute something to our knowledge of the survivor function, but nothing to our knowledge of the age at death. Another reason for censoring occurs when individuals are lost from the study; they may be killed in accidents, they may emigrate, or they may lose their identity tags.

In general, then, our survival data may be a mixture of times at death and times after which we have no more information on the individual. We deal with this by setting up an extra vector called the *censoring indicator* to distinguish between the two kinds of numbers. If a time really is a time to death, then the censoring indicator takes the value 1. If a time is just the last time we saw an individual alive, then the censoring indicator is set to 0. Thus, if we had the time data T and censoring indicator W:

T	4	7	8	8	12	15	22
W	1	1	0	1	1	0	1

this would mean that all the data were genuine times at death except for two cases, one at time 8 and another at time 15, when animals were seen alive but never seen again.

With repeated sampling in survivorship studies, it is usual for the degree of censoring to decline as the study progresses. Early on, many of the individuals are alive at the end of each sampling interval, whereas few if any survive to the end of the last study period.

An example with censoring and non-constant hazard

This study involved 4 cohorts of cancer patients each of 30 individuals. They were allocated to Drug A, Drug B, Drug C or a placebo at random. The year in which they died (recorded as time after treatment began) is the response variable. Some patients left the study before their age at death was known (these are the censored individuals with status = 0). Remember to **remove the variables status and death** between each analysis:

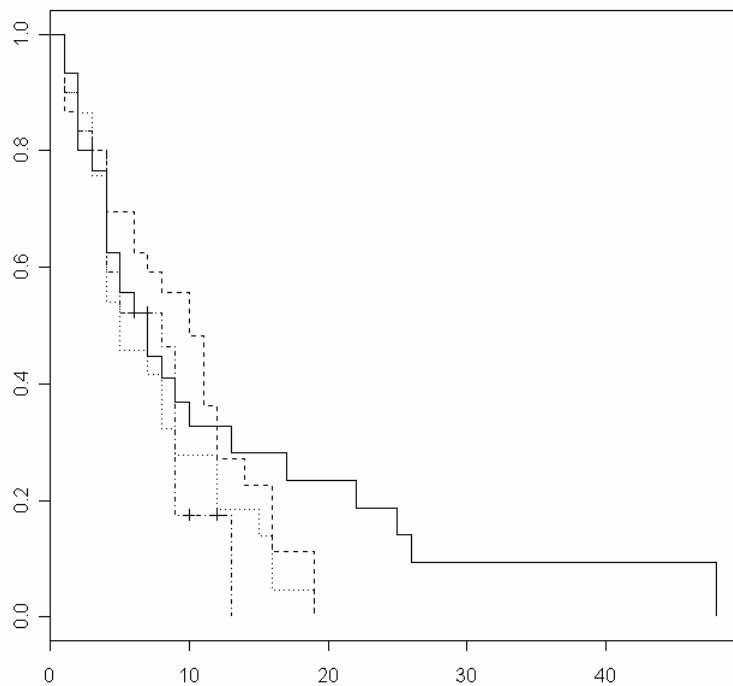
```
rm(status)
rm(death)
```

```
cancer<-read.table("c:\\temp\\cancer.txt",header=T)
attach(cancer)
names(cancer)
```

```
[1] "death"      "treatment" "status"
```

We start by plotting the survivorship curves for patients in the 4 different treatments:

```
plot(survfit(Surv(death,status)~treatment),lty=c(1,2,3,4))
```



The mean age at death for the 4 treatments (ignoring censoring) was

```
tapply(death[status==1],treatment[status==1],mean)
```

DrugA	DrugB	DrugC	placebo
9.480000	8.360000	6.800000	5.238095

The patients receiving Drug A lived an average of more than 4 years longer than those receiving the placebo. We need to test the significance of these differences, remembering that the variance in age-at-death is very high:

```
tapply(death[status==1],treatment[status==1],var)
```

DrugA	DrugB	DrugC	placebo
117.51000	32.65667	27.83333	11.39048

We start with the simplest model, assuming exponential errors and constancy in the risk of death:

```
model<-survreg(Surv(death,status)~treatment,dist="exponential")
summary(model)
```

	Value	Std. Error	z	p
(Intercept)	2.448	0.200	12.238	1.95e-34
treatmentDrugB	-0.125	0.283	-0.442	6.58e-01

treatmentDrugC	-0.430	0.283	-1.520	1.28e-01
treatmentplacebo	-0.333	0.296	-1.125	2.61e-01

Scale fixed at 1

Exponential distribution

Loglik(model)= -310.1 Loglik(intercept only)= -311.5
 Chisq= 2.8 on 3 degrees of freedom, p= 0.42

This analysis does not distinguish between the 4 treatments, despite the large differences in mean age at death. The placebo is not significantly different from Drug A ($p = 0.261$). Next we try a model with a different error assumption: the extreme value distribution:

```
model<-survreg(Surv(death,status)~treatment,dist="extreme")
summary(model)
```

	Value	Std. Error	z	p
(Intercept)	22.91	2.0686	11.07	1.69e-28
treatmentDrugB	-11.16	2.7548	-4.05	5.13e-05
treatmentDrugC	-13.38	2.7487	-4.87	1.12e-06
treatmentplacebo	-13.29	2.9357	-4.53	5.97e-06
Log(scale)	2.21	0.0717	30.76	8.32e-208

Scale= 9.08

Extreme value distribution

Loglik(model)= -371.7 Loglik(intercept only)= -383.7
 Chisq= 23.94 on 3 degrees of freedom, p= 2.6e-05

This model indicates significant effects of treatment on death rate: Drug A gave improved survival compared to Drugs B and C and the placebo, but the difference between Drug C and the placebo was not significant ($t = (13.38 - 13.29) / 2.9357$). The scale parameter = 9.08 indicates that mortality is significantly age dependent, so the earlier exponential distribution (scale = 1), assuming that mortality was not age dependent, was not appropriate. A full analysis of these data would fit more covariates (details about each patient) and test for time-dependency in the model parameters. The present point is that if, as here, the death risk is a function of age, then assuming the simpler exponential model does not allow us to detect the significant differences that exist between the treatments.

```
detach(cancer)
```

The likelihood function for censored data

A given individual contributes to the likelihood function depending upon whether it is alive or dead at time t . If it has died, then the censoring indicator is 1 and we learn more about $f(t)$; if it is still alive, then w_i is 0 and we learn only about $S(t)$:

$$L(\beta) = \prod_{i=1}^n [f(t_i)]^{w_i} [S(t_i)]^{1-w_i}$$

Now recall that the hazard function $h(t)$ is given by $f(t)/S(t)$. Thus we have $S(t)^w$ in the denominator and $S(t)^{1-w}$ in the numerator, so the likelihood function becomes:

$$L(\beta) = \prod_{i=1}^n [h(t_i)]^{w_i} S(t_i)$$

which involves data only from the hazard function of the uncensored individuals. If we replace $1/\mu_i$ by λ_i , then we can write the likelihood function for the exponential distribution as follows:

$$L(\beta) = \prod_{i=1}^n \lambda_i^{w_i} e^{-\lambda_i t_i}$$

For reasons that will become clear in a moment, it is convenient to multiply both the numerator and the denominator by $\prod_{i=1}^n t_i^{w_i}$. This gives

$$L(\beta) = \frac{\prod_{i=1}^n (\lambda_i t_i)^{w_i} e^{-\lambda_i t_i}}{\prod_{i=1}^n t_i^{w_i}}$$

Because the denominator is not a function of the estimated parameters β , it can be omitted from the likelihood formula, leaving only a term for the likelihood of a set of n observations w_i , having independent Poisson distributions, with means $\lambda_i t_i$, where w_i is either 0 or 1. The model can be fit to the hazard rate λ in one of two ways. Let θ_i represent the Poisson mean $\lambda_i t_i$. Using the *linear hazard model*

$$\lambda_i = \beta' x_i$$

and

$$\theta_i = \beta'(t_i x_i)$$

The modelling proceeds as follows:

- use the censoring indicator w_i as the response variable
- declare the error as Poisson
- declare the link function as the identity link
- multiply each of the explanatory variables, including the unit vector, by t_i
- fit the model as usual.

This is somewhat long-winded, and the fitting is easier if the *log linear hazard model is employed*, because

$$-\log \mu_i = \log \lambda = \beta'x_i$$

and so

$$\log \theta_i = \log \lambda_i + \log t_i = \beta'x_i + \log t_i$$

This is easier to fit, because we do not need to multiply through all the explanatory vectors by t_i . Instead, we use $\log t_i$ as an offset, and proceed as follows:

- use the censoring indicator w_i as the response variable;
- declare the error as Poisson;
- declare the link function as the log link;
- declare $\log t_i$ as an offset;
- fit the model as usual.

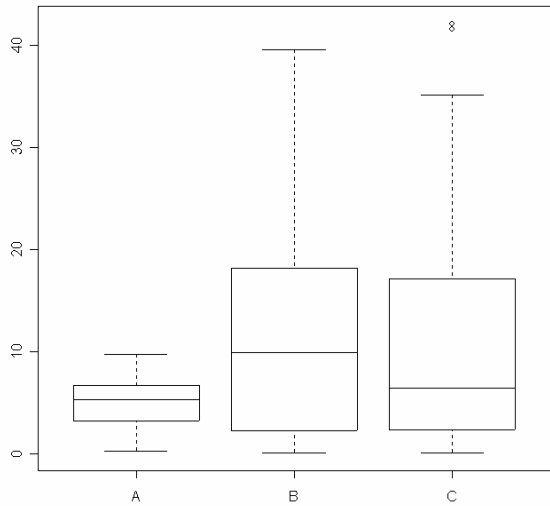
This rather curious procedure of using the censoring indicator full of 0's and 1's as the response variable should become clearer with an example.

An example with censoring

The next example comes from a study of mortality in 150 adult cockroaches. There were three experimental *groups*, and the animals were followed for 50 days. The groups were treated with three different insecticidal BT toxins added to their diet. The initial body mass of each insect (*weight*) was recorded as a covariate. The day on which each animal died (*death*) was recorded, and animals which survived up to the 50th day were recorded as being censored (for them, the censoring indicator *status* = 0).

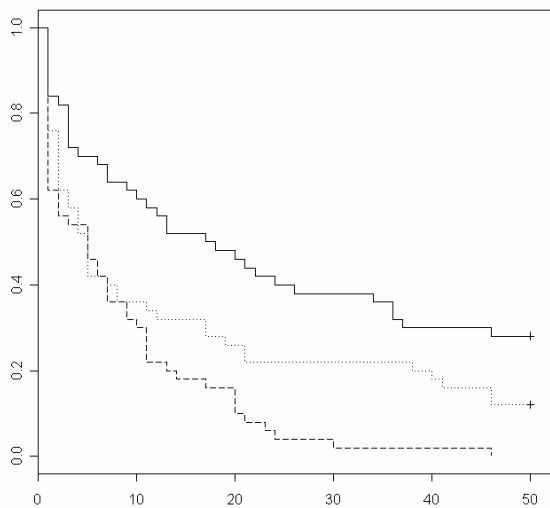
```
rm(status,death)
roaches<-read.table("c:\\temp\\roaches.txt",header=T)
attach(roaches)
names(roaches)
[1] "death"  "status" "weight" "group"
```

```
plot(group,weight)
```



The insects in batches B and C are much more variable than those in batch A. The overall survivorship curves for the 3 groups are obtained as before:

```
plot(survfit(Surv(death,status)~group),lty=c(1,3,5))
```



The crosses + at the end of the survivorship curves for groups A and B indicate that there was censoring in these groups (not all of the individuals were dead at the end of the experiment). Parametric regression in survival models uses the **survreg** function, for which you can specify a wide range of different error distributions. Here we use the exponential distribution for the purposes of demonstration (we can chose from `dist =`

"extreme", "logistic", "gaussian" or "exponential" and from link = "log" or "identity"), and fit the full analysis of covariance model to begin with:

```
model<-survreg(Surv(death,status)~weight*group,dist="exponential")
```

```
model
```

```
Call:
survreg(formula = Surv(death, status) ~ weight * group, dist =
"exponential")

Coefficients:
(Intercept)      weight      groupB      groupC weight:groupB
  3.87018131  -0.08030300  -0.88529869  -1.78043935   0.06425282
weight:groupC
  0.07957341

Scale fixed at 1

Loglik(model)= -480.6   Loglik(intercept only)= -502.1
    Chisq= 43.11 on 5 degrees of freedom, p= 3.5e-08
n= 150
```

To see the parameter estimates and their standard errors, use **summary**:

```
summary(model)
```

	Value	Std. Error	z	p
(Intercept)	3.8702	0.3854	10.041	1.00e-23
weight	-0.0803	0.0659	-1.219	2.23e-01
groupB	-0.8853	0.4508	-1.964	4.95e-02
groupC	-1.7804	0.4386	-4.059	4.92e-05
weight:groupB	0.0643	0.0674	0.954	3.40e-01
weight:groupC	0.0796	0.0674	1.180	2.38e-01

```
Scale fixed at 1
```

```
Exponential distribution
Loglik(model)= -480.6   Loglik(intercept only)= -502.1
    Chisq= 43.11 on 5 degrees of freedom, p= 3.5e-08
Number of Newton-Raphson Iterations: 4
n= 150
```

Model simplification proceeds in the normal way. You could use **update**, but here (for variety only) we re-fit progressively simpler models and test them using **anova**. First we take out the different slopes for each group:

```
model2<-survreg(Surv(death,status)~weight+group,dist="exponential")
```

```
anova(model,model2,test="Chi")
```


Terms	Resid. Df	-2*LL	Test Df	Deviance	P(> Chi)
1 weight * group	144	961.1800	NA	NA	NA
2 weight + group	146	962.9411	-weight:group -2	-1.761142	0.4145462

The interaction is not significant so we leave it out and try deleting weight:

```
model3<-survreg(Surv(death,status)~group,dist="exponential")
```

```
anova(model2,model3,test="Chi")
```

Terms	Resid. Df	-2*LL	Test Df	Deviance	P(> Chi)
1 weight + group	146	962.9411	NA	NA	NA
2 group	147	963.9393	-weight -1	-0.9981333	0.3177626

This is not significant, so we leave it out and try deleting group:

```
model4<-survreg(Surv(death,status)~1,dist="exponential")
```

```
anova(model3,model4,test="Chi")
```

Terms	Resid. Df	-2*LL	Test Df	Deviance	P(> Chi)
1 group	147	963.9393	NA	NA	NA
2 1	149	1004.2865	-2	-40.34721	1.732661e-09

This is highly significant, so we add it back . The minimal adequate model is model3 with the 3-level factor *group*, but there is no evidence that initial body *weight* had any influence on survival.

```
summary(model3)
```

	Value	Std. Error	z	p
(Intercept)	3.467	0.167	20.80	3.91e-96
groupB	-0.671	0.225	-2.99	2.83e-03
groupC	-1.386	0.219	-6.34	2.32e-10

Scale fixed at 1

Exponential distribution

```
Loglik(model)= -482    Loglik(intercept only)= -502.1
      Chisq= 40.35 on 2 degrees of freedom, p= 1.7e-09
```

You can immediately see the advantage of doing proper survival analysis when you compare the predicted mean ages at death for *model3* with the crude arithmetic averages of the raw data on age at death:

```
tapply(predict(model3,type="response"),group,mean)
```

A	B	C
32.05555	16.38635	8.02000

tapply(death,group,mean)

	A	B	C
	23.08	14.42	8.02

If there is no censoring (all the individuals died, as in Group C) then the estimated mean ages at death are identical. But when there is censoring, the arithmetic mean underestimates the age at death, and when the censoring is substantial (as in Group A) this underestimate is very large (23.08 vs. 32.06).

Weibull distribution

The origin of the Weibull distribution is in *weakest link analysis*. If there are r links in a chain, and the strengths of each link Z_i are independently distributed $(0, \infty)$, then the distribution of weakest links $V = \min(Z_j)$ approaches the Weibull distribution as the number of links increases.

The Weibull is a 2-parameter model that has the exponential distribution as a special case. Its value in demographic studies and survival analysis is that it allows for the death rate to increase or to decrease with age, so that all 3 kinds of survivorship curve can be analysed (see above). The density, hazard and survival functions with $\lambda = \mu^{-\alpha}$ are:

$$f(t) = \alpha \lambda t^{\alpha-1} e^{-\lambda t^\alpha}$$

$$h(t) = \alpha \lambda t^{\alpha-1}$$

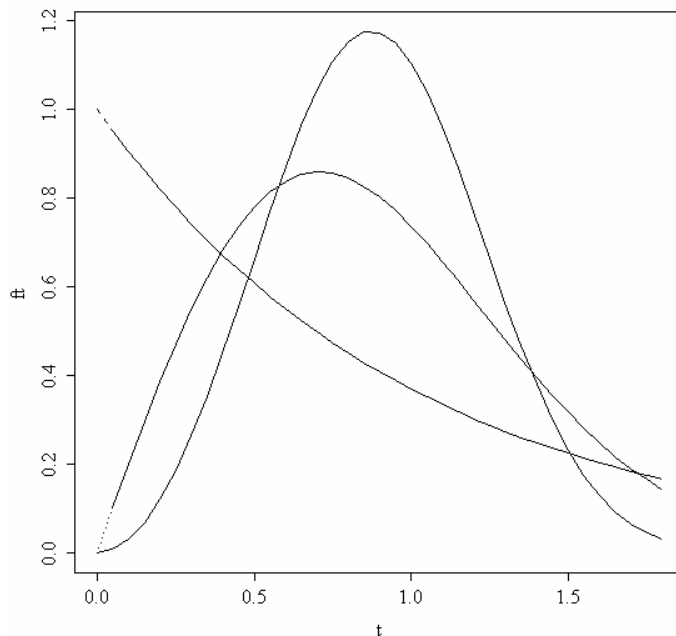
$$S(t) = e^{-\lambda t^\alpha}$$

The mean of the Weibull distribution is $\Gamma(1 + \alpha^{-1})\mu$ and the parameter α describes the shape of the hazard function (the background to determining the likelihood equations is given by Aitkin et al. (1989) pp 281-283). For $\alpha = 1$ (the exponential distribution) the hazard is constant, while for $\alpha > 1$ the hazard increases with age and for $\alpha < 1$ the hazard decreases with age.

Because the Weibull, lognormal and log-logistic all have positive skewness, it will be difficult to discriminate between them with small samples. This is an important problem, because each distribution has differently shaped hazard functions, and it will be hard, therefore, to discriminate between different ecological assumptions about the age-specificity of death rates. In survival studies, parsimony requires that we fit the exponential rather than the Weibull when the shape parameter α is not significantly different from 1.

Here is a family of 3 Weibull distributions with $\alpha = 1$, $\alpha = 2$ and $\alpha = 3$ (dotted, dashed and solid lines, respectively). Note that for large values of α the distribution becomes symmetrical, while for $\alpha \leq 1$ the distribution has its mode at $t = 0$. The variable name is lower-case L (for lambda)

```
a<-3
l<-1
t<-seq(0,1.8,.05)
ft<-a*l*t^(a-1)*exp(-l*t^a)
plot(t,ft,type="l")
a<-1
ft<-a*l*t^(a-1)*exp(-l*t^a)
lines(t,ft,type="l",lty=2)
a<-2
ft<-a*l*t^(a-1)*exp(-l*t^a)
lines(t,ft,type="l",lty=3)
```



Recall that that the shape parameter of the Weibull reflects the way that the risk of death changes as a function of age.

An example of censored survival data analysed using glm with Poisson errors

Suppose that in another GM trial we have two groups of caterpillars, each comprising 21 larvae of the same initial size and age. They are fed on leaves from two kinds of plants; the first group get leaves from a plant that has been genetically engineered to express BT toxin in its foliage, while the second group get leaves from an otherwise identical, but

non-transgenic strain. The data consist of the time at death in days (when $w = 1$) or the time when the animal was lost to the study ($w = 0$). The survival analysis is very simple. The response variable is the censoring indicator (*status*) with Poisson errors and $\log(\text{time})$ as an offset:

```
rm(status)
```

```
transgenic<-read.table("c:\\temp\\transgenic.txt",header=T)
attach(transgenic)
names(transgenic)
[1] "time"      "status"    "diet"
```

where *time* is age at death (days), *status* is the censoring indicator (1 = dead, 0 = alive at the end of the experiment), and *diet* is a 2-level factor. Preliminary data inspection involves calculating the mean age at death for insect fed on control and transgenic Bt expressing leaves using **tapply**:

```
tapply(time,diet,mean)
```

```
      control  transgenic
17.095238    8.666667
```

Evidently the control insects lived much longer on average. It is useful to know how the censoring is distributed across individuals in the different treatments. We use **table** to count the cases:

```
table(status,diet)
```

```
status control  transgenic
      0  12         0
      1   9        21
```

That is very revealing. All of the censoring (12 cases of *status* = 0) occurred in the control group of insects. None of the insects fed on Bt expressing leaves survived until the end of the experiment. Nine control insects died compared with all 21 of the insects fed a diet of transgenic leaves. The model has the **censoring indicator as the response variable**, and **the variation to be explained by the model is introduced by the offset** (log time of death):

```
model<-glm(status~diet+offset(log(time)),family=poisson)
```

Note how the **offset** appears as an additive part of the model formula.

```
summary(model)
```

Coefficients:

```

              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -3.6861     0.3333 -11.060 < 2e-016 ***
diet          1.5266     0.3984   3.832 0.000127 ***

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 54.503 on 41 degrees of freedom
Residual deviance: 38.017 on 40 degrees of freedom
AIC: 102.02

```

Diet looks highly significant, but we prefer to test by deletion, using chi square:

```

model2<-glm(status~1+offset(log(time)),family=poisson)
anova(model,model2,test="Chi")

```

Analysis of Deviance Table

```

Model 1: status ~ diet + offset(log(time))
Model 2: status ~ 1 + offset(log(time))
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1         40      38.017
2         41      54.503 -1   -16.485 4.903e-05

```

So, no doubt there then. Diet had an enormously significant effect on mean time at death. It is useful to know how to back transform these coefficient tables when there are offsets.

The antilogs of the estimates give the hazard. The mean age at death is the reciprocal of the hazard. So mean age at death is given by

```

1/exp(tapply(predict(model)-log(time),diet,mean))

```

```

      control transgenic
39.888889    8.666667

```

Note that where there was lots of censoring (as in the case of the control insects) the estimated mean age at death is substantially *greater* than the arithmetic mean age at death

```

tapply(time,diet,mean)

```

```

      control transgenic
17.095238    8.666667

```

(39.89 vs. 17.10) whereas the non-censored means are identical (8.67 vs. 8.67).