Survival Analysis

A great many studies in statistics deal with deaths or with failures of components: they involve the numbers of deaths, the timing of death, or 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 and Prentice, 1980; Miller, 1981; Fleming and Harrington 1991), for which R supports a wide range of tools. 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 (as explained 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), then the component fails during that week and its lifetime can be calculated. If the random number is larger than 0.1, then 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 dataframe).

The simulation is very simple. We create a vector of random numbers, mos, that is long enough to be 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:

```r
mos <- runif(100)
```
The trick is to find the week number in which the component failed; this is the lowest subscript for which `mos` ≤ 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

```r
which(mos <= 0.1)
```

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

This means that 16 of my first set of 100 random numbers 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]`:

```r
which(mos <= 0.1)[1]
```

```
[1] 5
```

All we need to do to simulate the life spans of a sample of 30 components, `death1`, is to repeat the above procedure 30 times:

```r
deadth1 <- numeric(30)
for (i in 1:30) {
  mos <- runif(100)
  death1[i] <- which(mos <= 0.1)[1]
}
deadth1
```

```
[1] 5 8 7 23 5 4 18 2 6 4 10 12 7 3 5 17 1 3 2 1 1 2
[22] 8 2 12 6 3 13 16 3 4
```

The fourth component survived for a massive 23 weeks but the 17th component failed during its first week. The simulation has roughly the right average weekly failure rate:

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

```
[1] 0.1351351
```

which is as close to 0.1 as we could reasonably expect from a sample of only 30 components.

Now we do the same for the second factory with its failure rate of 0.2:

```r
deadth2 <- numeric(30)
for (i in 1:30) {
  mos <- runif(100)
  death2[i] <- which(mos <= 0.2)[1]
}
deadth2
```

The sample mean is again quite reasonable:

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

```
[1] 0.2205882
```

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(1,30),rep(2,30)))
```

We get a visual assessment of the data as follows:

```r
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 using `glm` with family = Gamma (note the upper-case G):

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

Call:

```r
glm(formula = death ~ factory, family = Gamma)
```

Deviance Residuals:

```
  Min       1Q   Median       3Q      Max
-1.5077 -0.7356 -0.3772  0.2998  2.1323
```

Coefficients:

```
             Estimate Std. Error  t value Pr(>|t|)
(Intercept)  0.135140   0.022187    6.092  9.60e-08 ***
factory2     0.085455   0.042460    2.013   0.0498 *
```

The plot of simulated data shows the skewness of the Gamma distribution.
(Dispersion parameter for Gamma family taken to be 0.8082631)

Null deviance: 44.067 on 59 degrees of freedom
Residual deviance: 40.501 on 58 degrees of freedom
AIC: 329.62

Number of Fisher Scoring iterations: 6

We conclude that the factories are just marginally significantly different in mean age at failure of these components ($p = 0.0488$). So, even with a twofold difference in the true failure rate, it is hard to detect a significant difference in mean age at death with samples of size $n = 30$. 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.

It is good practice to remove variable names (like death) that you intend to use later in the same session (see rm on p. 804).

### Background

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 Chapter 16); in due course, they all die. Look at the following figure:

![Survivorship Curve](image)

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 R's powerful facilities

### SURVIVAL ANALYSIS

for the analysis of survival data than it is to pick an arbitrary time at which to compare proportions.

Demographers, actuaries and ecologists use three interchangeable concepts when dealing with data on the timing of death: survivorship, age at death and instantaneous risk of death. There are three broad patterns of survivorship: Type I, where most of the mortality occurs late in life (e.g. humans); Type II, where mortality occurs at a roughly constant rate throughout life; and Type III, where most of the mortality occurs early in life (e.g. salmon fishes). On a log scale, the numbers surviving from an initial cohort of 1000, say, look like this:

![Survivorship Curve](image)

### The survivor function

The survivorship curve plots the natural log of the proportion of a cohort of individuals starting 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 (see above). This means that a constant proportion of the individuals alive at the beginning of a time interval will die during that interval (i.e. the proportion dying is independent of density and constant for all age groups). When the death rate is highest for the youngest age classes we get Type III survivorship curve, which descends steeply at first, with the rate of descent easing later on. When it is the oldest animals that have the highest risk of death (as in the case of human population in affluent societies where the infant mortality rate is low) we obtain the Type I curve, which has a shallow descent to start, becoming steeper later.

### The density function

The density function describes the fraction of all deaths from our initial cohort that are likely to occur in a given brief instant of time. For the Type II curve this is a negati-
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).

The hazard function

The hazard is the instantaneous risk of death — that is, the derivative of the survivorship curve. It is the instantaneous rate of change in the log of the number of survivors per unit time. 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 one-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 \( \alpha \) is equal to 1.

Density function

The density function is the probability of dying in the small interval of time between \( t \) and \( t + \Delta t \), and 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:

\[
f(t) = \frac{e^{-t/\mu}}{\mu},
\]

where both \( \mu \) and \( t > 0 \). Note that the density function has an intercept of \( 1/\mu \) (remember that \( e^0 = 1 \)). The number from the initial cohort dying per unit time 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 that are still alive at time \( t \):

\[
S(t) = e^{-t/\mu}.
\]

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 \( t \).

SURVIVAL ANALYSIS

Hazard function

This is the statistician’s 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 is 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 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. The survivor, density and hazard functions of the exponential distribution are as follows (note the changes of scale on the y axis):

Of course, the death rate may not be a linear function of age. For example, the death rate may be high for young animals as well as for young animals, in which case the survivorship curves are like an L shape on its side.

Kaplan–Meier Survival Distributions

This is a discrete stepped survivorship curve that adds information as each death occurs. Suppose we had \( n = 5 \) individuals and that the times at death were 12, 17, 29, 35 and 4 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, and finally to zero at time 42.
The solid line shows the survival distribution and the dotted lines show the confidence intervals (see below). In general, therefore, we have two groups at any one time: the number of deaths \( d(t) \) and the number at risk \( r(t) \) (i.e. those that have not yet died: the survivors). The Kaplan–Meier survivor function is

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

which, as we have seen, produces a step at every time at which one or more deaths occurs. Censored individuals that survive beyond the end of the study are shown by a + on the plot or after their age in a dataframe (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:

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Hazard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential</td>
<td>constant = ( \frac{1}{\mu} )</td>
</tr>
<tr>
<td>Weibull</td>
<td>( \alpha \exp(-\lambda t) )</td>
</tr>
<tr>
<td>Gompertz</td>
<td>( be^t )</td>
</tr>
<tr>
<td>Makeham</td>
<td>( a + 6t )</td>
</tr>
<tr>
<td>Extreme value</td>
<td>( \frac{1}{\sigma} \exp(-\sigma</td>
</tr>
<tr>
<td>Rayleigh</td>
<td>( a + 0.5t )</td>
</tr>
</tbody>
</table>

**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 very simply in R by specifying a generalized linear model with gamma errors. The second case involves the use of exponential survival models with a censoring indicator (1 indicates that the response is a time at death, 0 indicates that the individual was alive when last seen; see below, and p. 801). The third case involves a
choice between parametric models, based on the Weibull distribution, and non-parametric techniques, based on the Cox proportional hazard model.

**Parametric models**

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

- gamma;
- exponential;
- piecewise exponential;
- extreme value;
- log-logistic;
- lognormal;
- Weibull.

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.

**Cox proportional hazards model**

This is the most widely used regression model for survival data. It assumes that the hazard is of the 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 $\beta$ 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 $\lambda$ 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

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

The product of these terms over all times of death, $L(\theta) = \prod L_i(\theta)$, was christened a partial likelihood by Cox (1972). This is clever, because maximizing $\log(L(\theta))$ allows an estimate of $\theta$ 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.

**Cox's proportional hazard or a parametric model?**

In cases where you have censoring, or where you want to use a more complex error structure, you will need to choose between a parametric model, fitted using survreg, and a non-parametric model, fitted using coxph. If you want to use the model for prediction, then you have no choice: you must use the parametric survreg because coxph does not extrapolate beyond the last observation. Traditionally, medical studies use coxph while engineering studies use survreg (so-called accelerated failure-time models), but both disciples could fruitfully use either technique, depending on the nature of the data and the precise question being asked. Here is a typical question addressed with coxph: 'How much does the risk of dying decrease if a new drug treatment is given to a patient? In contrast, parametric techniques are typically used for questions like this: 'What proportion of patients will die in 2 years based on data from an experiment that ran for just 4 months?'

**Parametric analysis**

The following example concerns survivorship of two cohorts of seedlings. All the seedling died eventually, so there is no censoring in this case. There are two questions:

- Was survivorship different in the two cohorts?
- Was survivorship affected by the size of the canopy gap in which germination occurred?

Here are the data:

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

(1) "cohort" "death" "gapsize"

We need to load the survival library:

```r
call: survfit(formula = Surv(death, status) - cohort)
call: survfit(formula = Surv(death, status) - cohort)
```

### Call: survfit(formula = Surv(death, status) - cohort)

```
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```

```r
call: survfit(formula = Surv(death, status) - cohort)
```
To plot these figures use `plot(model)` like this:

```r
plot(model, lty=c(1,3), ylab="Survivorship", xlab="Week")
```

The solid line is for the October cohort and the dashed line is for September. To see if median times at death for the two cohorts, just type:

```r
model
```

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

```
 n events median  0.95LCL  0.95UCL
cohort=October 30  30  4.5  3.9  9
cohort=September 30  30  3.5  2.2  7
```

### Cox's Proportional Hazards

The median age at death was one week later in the October cohort, but look at the width of the confidence intervals: 3 to 9 versus 2 to 7. Clearly there is no significant effect of cohort on time of death. What about gap size? We start with a full analysis of covariance using `coxph` rather than `survfit`.

```r
model1 <- coxph(Surv(death, status) ~ strata(cohort) * gapsize)
summary(model1)
```

Call:
```
coxph(formula = Surv(death, status) ~ strata(cohort) * gapsize)
n = 60
```
Models with Censoring

Censoring occurs when we do not know the time of death for all of the individuals. Censoring occurs principally because some individuals outlive the experiment, while others die during the experiment before they die. We know when we last saw them alive, but we have no information about their age at death. These individuals contribute something to our knowledge of the survival function, but nothing to our knowledge of the age at death. Another reason censoring occurs is 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 at which we have no more information on the individual. We deal with this by setting an extra vector called the censoring indicator to distinguish between the two kinds of times. If a time really is a time to death, then the censoring indicator takes the value one. If a time is just the last time we saw an individual alive, then the censoring indicator is zero. Thus, if we had the time data \( T \) and censoring indicator \( W \) on seven individuals,

\[ T = 4 7 8 8 12 15 22 \]
\[ W = 1 1 0 1 1 0 1 \]

this would mean that five of the times were times at death while in two cases, one at time 8 and another at time 15, individuals 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. The following example comes from a study of cancer patients undergoing one of four different treatment programmes (drugs A, B and C and a placebo):

```
summary(model2)
```

```
Call:
coxph(formula = Surv(death, status) ~ strata(cohort) + gapsize)
n= 60

 coef exp(coef) se(coef)  z    p
 gapsize -0.855  0.425  0.405 -2.11 0.035
 exp(coef) exp(-coef) lower .95 upper .95
 gapsize 0.425   2.350   0.192   0.942

Rsquare= 0.068 (max possible= 0.993 )
Likelihood ratio test = 4.24 on 1 df,  p=0.0395
Wald test = 4.44 on 1 df,  p=0.0350
Score (logrank) test = 4.54 on 1 df,  p=0.0331
```

We conclude that risk of seedling death is lower in bigger gaps (coef = -0.855) but this effect is similar in the September and October-germinating cohorts.

You see that the modelling methodology is exactly the same as usual: fit a complicated model and simplify it to find a minimal adequate model. The only difference is the use of \( \text{Surv}(\text{death}, \text{status}) \) if that the response is a Kaplan-Meier object.

```
detach(seedlings)
rm(status)
```
cancer=read.table("c:\temp\cancer.txt",header=T)
names(cancer)

[1] "death" "treatment" "status"
plot(survfit(Surv(death,status)~treatment),lty=c(1:4),ylab="Survivorship",xlab="Time")
tapply(death[status==1],treatment[status==1],mean)

DrugA DrugB DrugC placebo
9.480000 8.360000 6.800000 5.238095

The long tail is for drug A. The latest deaths in the other treatments were at times 14 and 19. The variances in age at death are dramatically different under the various treatments:
tapply(death[status==1],treatment[status==1],var)

DrugA DrugB DrugC placebo
117.51000 32.65667 27.83333 11.39048

Parametric models
The simplest model assumes a constant hazard: dist="exponential".
model1<-survreg(Surv(death,status)~treatment,dist="exponential")
summary(model1)

Call:
survreg(formula = Surv(death, status) ~ treatment, dist = "exponential")

Value Std. Error z  p
(Intercept) 2.448 0.200 12.238 1.95e-34

treatmentDrugB -0.125 0.283 -0.443 6.58e-01

treatmentDrugC -0.430 0.283 -1.520 1.28e-01

treatmentplacebo -0.333 0.296 -1.125 2.61e-01

Exponential distribution
Loglik(model)=-310.1 Loglik(intercept only)=-311.5
Chisq= 2.8 on 3 degrees of freedom, p= 0.42
Number of Newton-Raphson Iterations: 4
n= 120

Under the assumption of exponential errors there are no significant effects of drug treatment on survivorship (all p > 0.1). How about modelling non-constant hazard using Weibull errors instead (these are the default for survreg)?
model2<-survreg(Surv(death,status)~treatment)
summary(model2)

Call:
survreg(formula = Surv(death, status) ~ treatment)

Value Std. Error z  p
(Intercept) 2.531 0.1572 16.102 2.47e-58

treatmentDrugB -0.191 0.2193 -0.872 3.83e-01

treatmentDrugC -0.475 0.2186 -2.174 2.97e-02

treatmentplacebo -0.454 0.2313 -1.963 4.96e-02

Log(scale) -0.260 0.0797 -3.264 1.10e-03

SURVIVAL ANALYSIS

Scale= 0.771

Weibull distribution
Loglik(model)=-305.4 Loglik(intercept only)=-308.3
Chisq= 5.8 on 3 degrees of freedom, p= 0.12

The number of Newton-Raphson Iterations is 5
n= 120

The scale parameter 0.771, being less than 1, indicates that hazard decreases with age this study. Drug B is not significantly different from drug A (p = 0.38), but drug C is the placebo are significantly poorer (p < 0.05). We can use anova to compare model1 to model2:

anova(model1,model2)

Terms Resid. Df -2*LL Test Df Deviance P>|Chi|
1 treatment 116 620.1856 NA NA NA
2 treatment 115 610.7742 = 1 9.4114 0.002156405

model2 with Weibull errors is significant improvement over model1 with exponential err (p = 0.002).

We can try amalgamating the factor levels - the analysis suggests that we begin grouping A and B together:
treat2<-treatment
levels(treat2)

[1] "DrugA" "DrugB" "DrugC" "placebo"

levels(treat2)[1:2]<-"DrugsAB"
levels(treat2)

[1] "DrugsAB" "DrugC" "placebo"

model3<-survreg(Surv(death,status)~treat2)
anova(model2,model3)

Terms Resid. Df -2*LL Test Df Deviance P>|Chi|
1 treat2 116 611.5190 1 vs. 2 =1 0.744833 0.3861171
2 treatment 115 610.7742 NA NA NA

That model simplification was justified. What about drug C? Can we lump it together with the placebo?

levels(treat2)[2:3]<-"placebo"

levels(model3,placebo)
named="placebo"

model4<-survreg(Surv(death,status)~treat2)
anova(model3,model4)

Terms Resid. Df -2*LL Test Df Deviance P>|Chi|
1 treat2 116 611.5190 NA NA NA
2 treatment 117 611.5301 = 1 0.01101390 0.9164208

Yes we can. That simplification was clearly justified (p = 0.916):

summary(model4)

Call:
survreg(formula = Surv(death, status) ~ treat2)
We can summarize the results in terms of the mean age at death, taking account of the censoring:

tapply(predict(model4,type="response"),treat2,mean)

Weibull distribution

Loglik(model) = -305.8 Loglik(Intercept only) = -308.3

Chisq= 5.05 on 1 degrees of freedom, p= 0.025

Number of Newton-Raphson Iterations: 5

n= 120

Comparing coxph and survreg survival analysis

Finally, we shall compare the methods, parametric and non-parametric, by analysing the same data set both ways. It is an analysis of covariance with one continuous explanatory variable (initial weight) and one categorical explanatory variable (group):

insects<-read.table("c:\temp\roaches.txt",header=T)
attach(insects)

[1] "death" "status" "weight" "group"

First, we plot the survivorship curves of the three groups:

plot(survfit(Surv(death,status)~group),lty=c(1,3,5),ylab="Survivorship",xlab="Time")

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).

We begin the modelling with parametric methods (survreg). We shall compare the default error distribution (Weibull, which allows for non-constant hazard with age) with the simpler exponential (assuming constant hazard):

model1<-survreg(Surv(death,status)~weight*group,dist="exponential")
summary(model1)

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

value std.error z     p
(Intercept) 3.8702 0.3054 10.041 1.00e-23
weight -0.0803 0.0659 -1.219 2.23e-01
groupB -0.8853 0.4508 -1.964 4.95e-01
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: 5

n= 150

model2 employs the default Weibull distribution allowing non-constant hazard:

model2<-survreg(Surv(death,status)~weight*group)
summary(model2)

Call:
survreg(formula = Surv(death, status) ~ weight * group)

Value Std. Error z     p
(Intercept) 3.9506 0.5308 7.443 9.84e-16
weight -0.0973 0.0909 -1.071 2.84e-01
groupB -1.1337 0.6207 -1.826 6.78e-02
groupC -1.9841 0.6040 -3.285 1.02e-03
**Survival Analysis**

<table>
<thead>
<tr>
<th>Value</th>
<th>Std. Error</th>
<th>z</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>3.459</td>
<td>0.2283</td>
<td>15.15</td>
</tr>
<tr>
<td>groupB</td>
<td>-0.822</td>
<td>0.3097</td>
<td>-2.65</td>
</tr>
<tr>
<td>groupC</td>
<td>-1.540</td>
<td>0.3016</td>
<td>-5.11</td>
</tr>
<tr>
<td>Log(scale)</td>
<td>0.314</td>
<td>0.0705</td>
<td>4.46</td>
</tr>
</tbody>
</table>

Scale= 1.17

Weibull distribution

Loglik(model) = 470.5 Loglik(intercept only) = -483.3

Chisq = 25.63 on 2 degrees of freedom, p= 2.70e-06

Number of Newton-Raphson Iterations: 5

n= 150

It is clear that all three groups are required (B and C differ by 0.72, with standard error 0.31), so this is the minimal adequate model. Here are the predicted mean ages at death:

`tapply(predict(model3),group,mean)`

A      
B      
C      

31.796137  13.972647  6.814384

You can compare these with the mean ages of those insects that died

`tapply(status==1,group,mean)`

A      
B      
C      

12.611111  9.568182  8.020000

and the ages when insects were last seen (dead or alive)

`tapply(status==1,group,mean)`

A      
B      
C      

23.08  14.42  8.02

The predicted ages at death are substantially greater than the observed ages at last sighting when there is lots of censoring (e.g. 31.8 vs. 23.08 for group A).

Here are the same data analysed with the Cox proportional hazards model:

`model10<coxph(Surv(death,status)~weight*group)`

Summary:

<table>
<thead>
<tr>
<th>Value</th>
<th>Std. Error</th>
<th>z</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>weight</td>
<td>0.0633</td>
<td>1.05</td>
<td>0.0674</td>
</tr>
<tr>
<td>groupB</td>
<td>0.7910</td>
<td>2.206</td>
<td>0.4564</td>
</tr>
<tr>
<td>groupC</td>
<td>1.2863</td>
<td>3.620</td>
<td>0.4524</td>
</tr>
<tr>
<td>weight*groupB</td>
<td>-0.0557</td>
<td>0.946</td>
<td>0.0688</td>
</tr>
<tr>
<td>weight*groupC</td>
<td>-0.0587</td>
<td>0.943</td>
<td>0.0690</td>
</tr>
</tbody>
</table>

Coef exp(coef) se(coef)      z    p

weight 0.0633 1.05 0.0674 0.940 0.3500

Weight

<table>
<thead>
<tr>
<th>Value</th>
<th>Std. Error</th>
<th>z</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>weight</td>
<td>1.065</td>
<td>0.939</td>
<td>0.934</td>
</tr>
<tr>
<td>groupB</td>
<td>2.206</td>
<td>0.453</td>
<td>0.902</td>
</tr>
<tr>
<td>groupC</td>
<td>3.620</td>
<td>0.276</td>
<td>1.491</td>
</tr>
<tr>
<td>weight*groupB</td>
<td>0.946</td>
<td>1.057</td>
<td>0.827</td>
</tr>
<tr>
<td>weight*groupC</td>
<td>0.943</td>
<td>1.060</td>
<td>0.824</td>
</tr>
</tbody>
</table>
Rsquare = 0.135 (max possible = 0.999)
Likelihood ratio test = 21.8 on 5 df, p = 0.000564
Wald test
   = 20.8 on 5 df, p = 0.000903
Score (logrank) test = 22.1 on 5 df, p = 0.000513

As you see, the interaction terms are not significant (p > 0.39) so we simplify using step as before:

model11 <- step(model10)

Start: AIC = 1133.54
Surv(death, status) ~ weight * group
     Df   AIC
weight:group    2  1110.3
<none>          1113.5

Step: AIC = 1110.27
Surv(death, status) ~ weight + group
     Df   AIC
weight     1  1108.8
<none>    1110.3
     Df   AIC
group      2  1123.7

Step: AIC = 1108.82
Surv(death, status) ~ group
     Df   AIC
<none> 1108.8
     Df   AIC
group  2 1125.4

Note that the AIC values are different than they were with the parametric model. The interaction term is dropped because this simplification reduces AIC to 1110.3. Then the covariate (weight) is dropped because this simplification also reduces AIC to 1108.8. But removing group would increase AIC to 1125.4, so this is not done. The minimal model contains a main effect for group but no effect of initial weight.

summary(model11)

Call:
coxph(formula = Surv(death, status) ~ group)
 n = 150
 coef se(coef) z     p
  groupB  0.561  0.226  2.48 1.3e-02
  groupC  1.008  0.226  4.46 8.3e-06

exp(coef) exp(-coef) lower .95 upper .95
  groupB 1.75   0.571   1.13   2.73
  groupC 2.74   0.365   1.76   4.27

Rsquare = 0.128 (max possible = 0.999)
Likelihood ratio test = 20.6 on 2 df, p = 3.45e-05
Wald test
   = 19.9 on 2 df, p = 4.87e-05
Score (logrank) test = 21 on 2 df, p = 2.77e-05

tapply(death, group, mean)

A  B  C
23.08 14.42 8.02

Evidently, individuals in group A lived a lot longer than those in group C. The ratio of their mean ages at death is 23.08/8.02 which evaluates to:
23.08/8.02

[1] 2.877805

Likewise, individuals in group A lived longer than individuals in group B by a ratio
23.08/14.42

[1] 1.600555

These figures are the approximate hazards for an individual in group C or group B relative to an individual in group A. In the coxph output of model11 they are labelled exp(coef). The model values are slightly different from the raw means because of the way that the model has dealt with censoring (14 censored individuals in group A, 6 in group B and none in group C): 1.6 vs. 1.75 and 2.8778 vs. 2.74

You should compare the outputs from the two functions coxph and survreg to make sure you understand their similarities and their differences. One fundamental difference is that the parametric Kaplan–Meier survival curves refer to the population, whereas Cox proportional hazards refer to an individual in a particular group.

plot(survfit(model11))

legend(35, 0.8, c("Group A", "Group B", "Group C"), lty = c(2, 1, 2))

This plot shows the survivorship curve for the average individual in each group with the covariates held at their average values (but not in this example, since we have eliminated them).