Survival analysis

Patrick Breheny

December 10, 2014

Today's lab is about survival analysis. Obviously, survival analysis is a big subject and we're just scratching the surface, but all biostatisticians should be familiar with what survival data looks like and how it is organized, and be able to construct a Kaplan-Meier curve and carry out some basic inference concerning it.

1 The `survival` package

1.1 Kaplan-Meier curves

Survival analysis functions are not loaded by default in `R`; to access them you have to load the `survival` package. The package is very rich with features and can do many things, but we're just going to use a few of its most common functions. First, let's take a look at the data and how it is organized (using the same anemia data we looked at in class):

```
> require(survival)
> anemia <- read.delim("http://myweb.uiowa.edu/pbreheny/data/anemia.txt")
> head(anemia)

  Trt Time_gvhd Status_gvhd Time Status
1 MTX         9           1   30      1
2 MTX        11           1   44      1
3 MTX        12           1  104      1
4 MTX        20           1  106      1
5 MTX        20           1  181      1
6 MTX        25           1  329      0
```

As we discussed in class, survival outcomes consist of two components: the time on study and the event/censoring indicator. In the data, these appear as two distinct columns (there are actually four "outcome" columns here, two for GVHD and two for mortality). To analyze the data, we need to "bundle" these columns together into what the `survival` package calls a `Surv` object:

```
> S.gvhd <- with(anemia, Surv(Time_gvhd, Status_gvhd))
> S.mort <- with(anemia, Surv(Time, Status))
> head(S.mort)

[1]  30   44  104  106  181  329+
```

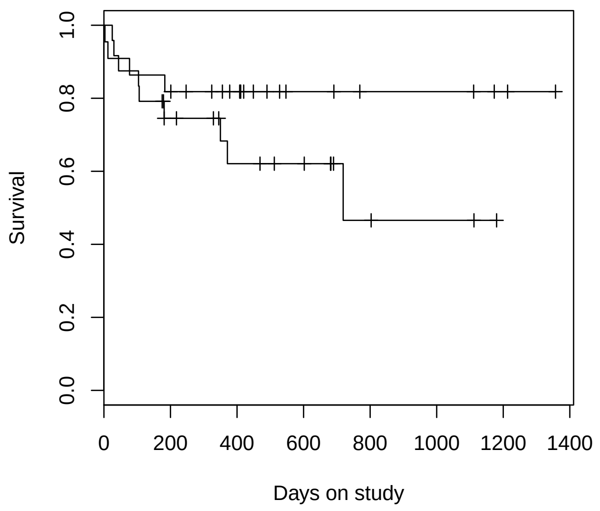
Note that the censored observations appear with a little `+` after them to indicate that the true time-to-event is over 329 days. Now that we've created the survival outcomes, we can analyze them using `survfit`, which works like the other `R` functions we've seen in class:

```
> Trt <- anemia$Trt
> fit <- survfit(S.mort ~ Trt)
> fit

Call: survfit(formula = S.mort ~ Trt)

            records n.max n.start events median 0.95LCL 0.95UCL
Trt=MTX          24    24      24      9    719     371      NA
Trt=MTX+CSP      22    22      22      4     NA      NA      NA
```

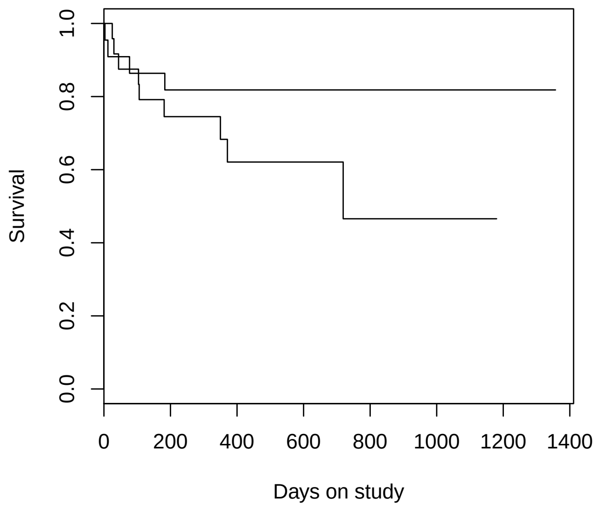
This tells us the sample size, the number of events (here, deaths) in each group, and tries to estimate the median survival time and give us a confidence interval for it.

What's going on with the `NA`'s? The reason for them will become clear once we plot the Kaplan-Meier curves. To do so, we can simply call `plot(fit)`; here is that call, with a few options turned on and off so you can see what they do:

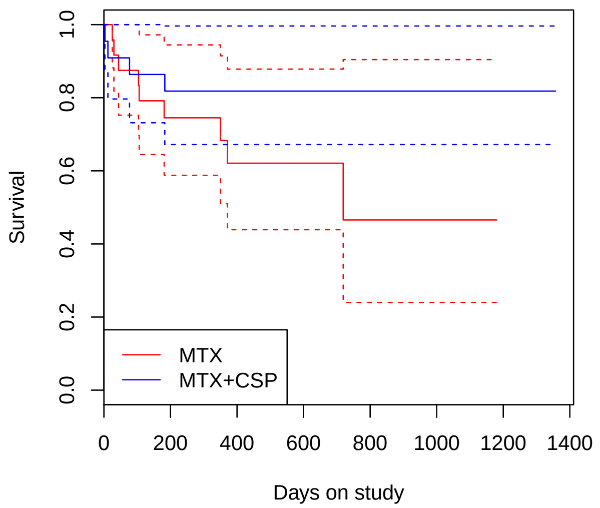
```
> plot(fit, xlab="Days on study", ylab="Survival")
```



```
> plot(fit, xlab="Days on study", ylab="Survival", mark.time=FALSE)
```



```
> plot(fit, xlab="Days on study", ylab="Survival", mark.time=FALSE, conf.int=TRUE,
+      col=c("red", "blue"))
> legend("bottomleft", legend=unique(Trt), col=c("red", "blue"), lty=1)
```



So as we can see, the median survival time for the MTX+CSP group never drops below 50%, so we can't estimate a median for it (nor do either of its confidence limits). Likewise, we can't estimate an upper confidence limit for the median survival in the MTX group, because its upper confidence limit (the dashed red line) never reaches 50%.

So there we are: our first Kaplan-Meier curve. I happen to think that this figure is kind of ugly, so I use a different package for plotting KM curves (we'll see it in the next section). But before we do that, let's discuss the question of how to test for differences between survival curves.

1.2 Log-rank tests

As we (hopefully) touched upon in class, the most common approach is something called a log-rank test, which constructs contingency tables for each observed event time and then tests for differences by aggregating these tables.

```
> survdiff(S.mort ~ Trt)

Call:
survdiff(formula = S.mort ~ Trt)

             N Observed Expected (O-E)^2/E (O-E)^2/V
Trt=MTX     24        9     6.45     1.007      2.01
Trt=MTX+CSP 22        4     6.55     0.992      2.01

 Chisq= 2  on 1 degrees of freedom, p= 0.156

> survdiff(S.gvhd ~ Trt)

Call:
survdiff(formula = S.gvhd ~ Trt)

             N Observed Expected (O-E)^2/E (O-E)^2/V
Trt=MTX     24       13     7.95      3.21      6.51
Trt=MTX+CSP 22        3     8.05      3.17      6.51

 Chisq= 6.5  on 1 degrees of freedom, p= 0.0107
```

These tests indicate that in terms of mortality, we see a few more deaths than we would expect in the MTX group (9 vs. 6.5), but this could happen fairly easily by chance alone ($p = 0.16$). However, in terms of GVHD, we see quite a few more cases in the MTX group than we would expect (13 vs. 8), and this is not so easily explained by chance ($p = 0.01$), suggesting that CSP is important for warding off GVHD. Whether it also improves survival or not is inconclusive.

2 The rms package

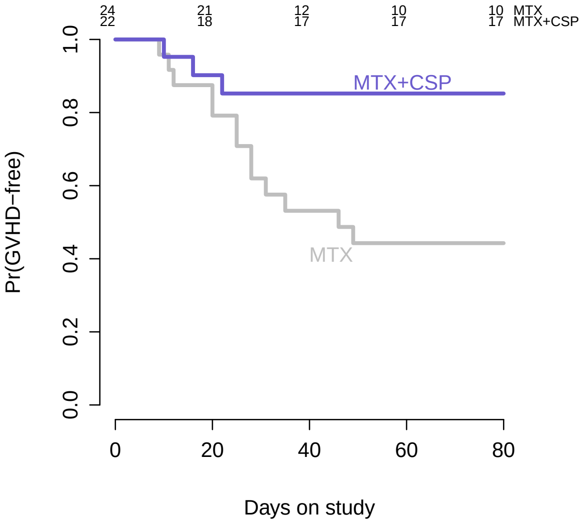
There are a number of packages out there that provide different options for Kaplan-Meier curves. I can't claim to have make an exhaustive trial of all the different packages out there – I just know that I've tried the **rms** package and thought it had some nice options and made some attractive plots, so I wanted to provide some examples of how to use it.

After installing the **rms** package, we can fit Kaplan-Meier curves with **npsurv**, which is **rms**'s equivalent to **survfit** (notice that **rms** uses **survival**'s **Surv** objects):

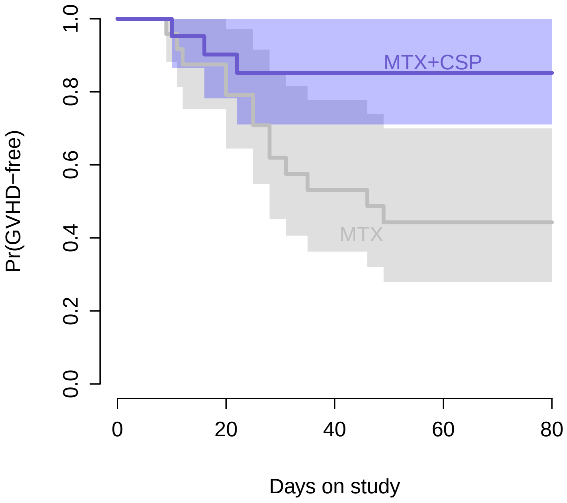
```
> require(rms)
> fit.gvhd <- npsurv(S.gvhd ~ Trt)
> fit.mort <- npsurv(S.mort ~ Trt)
```

You'll notice that **rms** comes with a bit of overhead, loading a number of supporting packages and redefining some basic R functions. And it's not necessarily the easiest package to use either, but it does make very nice plots:

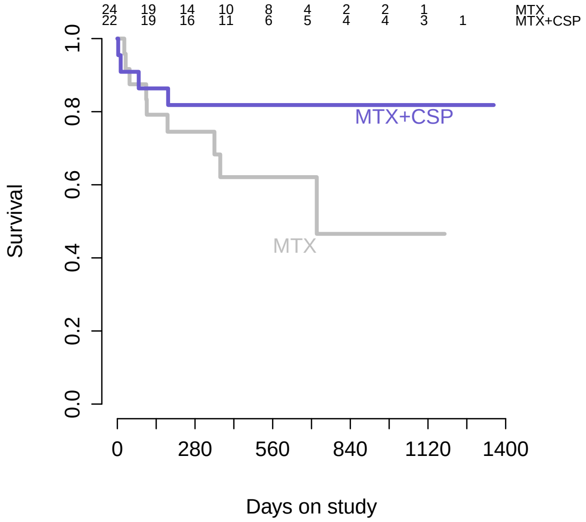
```
> col <- c("gray","slateblue")
> survplot(fit.gvhd, mark.time=FALSE, col=col, xlab="Days on study", ylab="Pr(GVHD-free)",
+          xlim=c(0,80), lty=1, conf="none", lwd=3, n.risk=TRUE, y.n.risk=1.05, cex.n.risk=0.7,
+          sep.n.risk=0.03, levels.only=TRUE, time.inc=20)
```



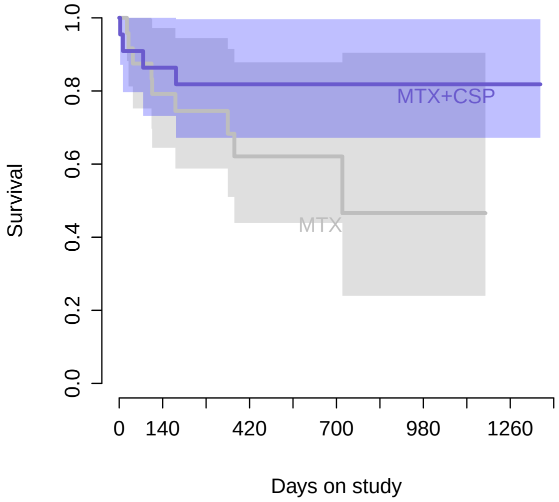
```
> col.fill <- c(rgb(.5,.5,.5,alpha=0.25), rgb(0,0,1,alpha=0.25))
> survplot(fit.gvhd, mark.time=FALSE, col=col, col.fill=col.fill, xlab="Days on study",
+          ylab="Pr(GVHD-free)", xlim=c(0,80), lty=1, lwd=3, levels.only=TRUE, time.inc=20)
```



```
> survplot(fit.mort, mark.time=FALSE, col=col, xlab="Days on study", ylab="Survival", lty=1,
+          conf="none", lwd=3, n.risk=TRUE, y.n.risk=1.05, cex.n.risk=0.7, sep.n.risk=0.03,
+          levels.only=TRUE)
```

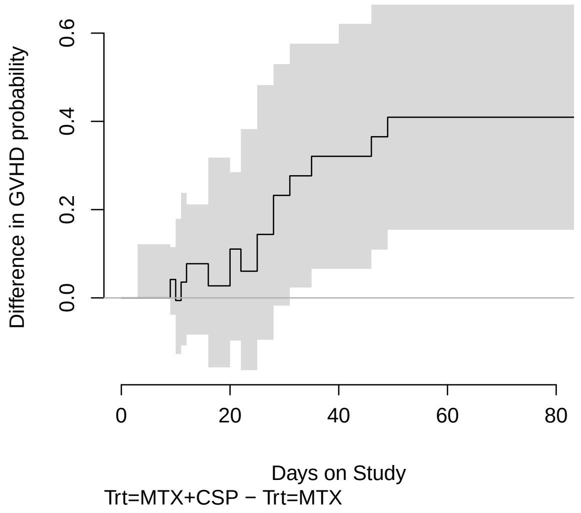


```
> survplot(fit.mort, mark.time=FALSE, col=col, col.fill=col.fill, xlab="Days on study",
+          ylab="Survival", lty=1, lwd=3, levels.only=TRUE)
```

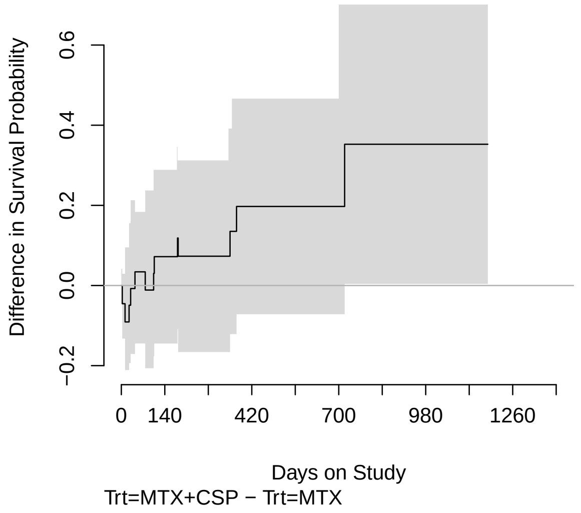


I find these plots much more appealing than the ones you get from `survival`. Another really nice option provided by `rms` is the ability to plot differences in survival. As we've said many times, just because confidence intervals overlap doesn't mean there isn't a significant difference between the groups. Because of this, `rms` provides a `survdiffplot` function:

```
> survdiffplot(fit.gvhd, xlim=c(0,80), time.inc=20, xlab="Days on Study",
+                 ylab="Difference in GVHD probability", order=2:1)
```



```
> survdiffplot(fit.mort, xlab="Days on Study", order=2:1)
```



As we said in class, it is particularly important to look at confidence intervals for Kaplan-Meier curves because the width can vary quite a bit over time due to subjects dropping out and other censoring events.

This concludes our brief introduction to survival analysis; for more information, consider taking Survival Data Analysis (BIOS 7210) or Applied Survival Analysis (BIOS 6210).