

Using `rjags` for survival data with right censoring

Malcolm Farrow
Newcastle University

October 4, 2016

1 Method using data augmentation

Here is a simple example of a model specification for a survival problem with right censoring.

```
model{
  for(i in 1:119)
  {
    is.censored[i]~dinterval(t[i],t.cen[i])
    t[i]~dexp(lambda[i])
    lambda[i]<-exp(beta0+beta.trt*trt[i])
  }

  beta0~dnorm(0.0,0.001)
  beta.trt~dnorm(0.0,0.001)
}
```

Suppose that this model specification is in a file called `MFbug.txt`.

Suppose that the data are initially in the form of a data frame called `data`.

This contains elements

- `data$t` : the event or censoring times
- `data$status` : 1 if the event is observed or 0 if the time is censored
- `data$trt` : a covariate

The data set used by `rjags` contains four vectors.

- `t`: Element i of `t` is the event time if the event was observed for case i . If case i was censored then element i of `t` is `NA`. We can construct this as follows.

```
t<-data$t
is.na(t)<-status==0
```

- `is.censored`: This is an indicator taking the value 1 if the time is censored and 0 if the event is observed. We can construct this as follows.

```
is.censored<-1-data$status
```

- `t.cen`: Element i of `t.cen` is the censoring time for case i if this case was censored. If the event was observed for case i then the “censoring time” must be greater than the observed event time. We can construct a suitable vector as follows.

```
t.cen<-data$t+data$status
```

- `trt`: A covariate.

```
trt<-data$trt
```

Notice that, in the model specification, `is.censored` is given a special distribution

```
is.censored[i]~dinterval(t[i],t.cen[i])
```

Experience has shown that it is usually necessary to initialise the missing values of `t`. This is because the specification if `is.censored` declares these cases to be censored. Therefore the unobserved event times must be greater than the censoring times. Allowing JAGS to choose initial values at random is likely to lead to some values for unobserved event times which are less than the censoring times and therefore inconsistent with this. So we should initialise the missing values of `t` to values greater than the corresponding values of `t.cen`. For example, to construct two sets of initial values, one for each of two parallel chains:

```
tinits1<-data$t+5
is.na(tinits1)<-data$status==1
tinits2<-tinits1+5
```

We can now form the data and initial values lists and build the model as follows.

```
kidneydata<-list(t=t,t.cen=t.cen,trt=trt,is.censored=is.censored)
kidneyinits<-list(list(t=tinits1),list(t=tinits2))
kidneyjags<-jags.model("MFbug.txt",data=kidneydata,inits=kidneyinits,n.chains=2)
```

Now away we go!

```
update(kidneyjags,5000)
kidneysamples<-coda.samples(kidneyjags,c('beta0','beta.trt'),100000)
```

2 Examples

2.1 Renal transplant data

The data in Table 1 are taken from Henderson and Milner (1991). They show the graft survival times (t) in months of 148 renal transplant patients. There is one covariate, x , the total number of HLA-B or DR antigen mismatches between donor and recipient. The data can be obtained in the file `renal.txt` from

<http://www.mas.ncl.ac.uk/~nmf16/teaching/mas8391/>

You can read the data and put them into a form ready for us with commands such as the following.

```
renal<-read.table("renal.txt",header=TRUE)
n<-length(renal[,1])
t<-renal$t
is.na(t)<-renal$status==0
is.censored<-1-renal$status
t.cen<-renal$t+renal$status
renaldata<-list(n=n,t=t,is.censored=is.censored,t.cen=t.cen,x=renal$x)
```

A suitable model specification is shown in Table 2. Type this into a file called `renalbug.txt`.

Create initial values as follows.

```
tinits1<-renal$t+5
is.na(tinits1)<-renal$status==1
tinits2<-tinits1+5
renalinits<-list(list(t=tinits1),list(t=tinits2))
```

Build the `rjags` model as follows.

```
renaljags<-jags.model("renalbug.txt",data=renaldata,init=renalinits,n.chains=2)
```

Use a “burn-in”.

```
update(renaljags,5000)
```

Collect samples. As the mixing is rather poor, we will use a large number of iterations.

```
renalsamples<-coda.samples(renaljags,c('beta0','beta.x'),50000)
```

Look at the results.

```
par(ask=TRUE)
traceplot(renalsamples)
summary(renalsamples)
densplot(renalsamples)
```

t	Status	x	t	Status	x	t	Status	x	t	Status	x
0.035	1	3	6.600	0	0	16.200	0	1	24.705	0	0
0.068	1	0	7.180	1	3	16.300	0	0	25.705	0	2
0.100	1	0	7.667	0	1	16.344	0	1	25.213	0	1
0.101	1	1	7.733	1	1	16.600	0	0	29.705	0	3
0.167	0	4	7.800	0	2	16.700	0	1	30.443	0	1
0.168	1	2	7.933	0	1	16.933	0	3	31.667	0	0
0.197	0	1	7.967	0	1	17.033	0	3	31.934	0	2
0.213	1	1	8.016	1	2	17.067	0	0	32.180	0	1
0.233	0	1	8.300	1	1	17.475	0	1	32.367	0	0
0.234	1	2	8.410	0	0	17.667	0	1	32.672	0	1
0.508	1	0	8.607	0	1	17.700	0	1	32.705	0	2
0.508	0	2	8.667	1	1	17.967	0	1	33.148	0	1
0.533	1	3	8.800	0	1	18.115	0	2	33.567	0	1
0.633	0	0	9.100	0	0	18.115	0	1	33.770	0	1
0.767	1	3	9.233	1	1	18.933	0	0	33.869	0	2
0.768	1	4	10.541	0	2	18.934	0	1	34.836	0	0
0.770	0	0	10.607	0	3	19.508	0	2	34.869	0	1
1.066	1	4	10.633	0	1	19.574	0	3	34.934	0	2
1.267	0	2	10.667	1	2	19.733	0	0	35.738	0	0
1.300	1	3	10.869	0	3	20.148	0	2	36.180	0	1
1.600	1	1	11.067	1	2	20.180	0	0	36.213	0	1
1.639	0	2	11.180	0	0	20.900	1	2	39.410	0	1
1.803	0	2	11.443	0	0	21.167	0	0	39.433	0	0
1.867	1	4	12.213	1	1	21.233	0	0	39.672	0	0
2.180	1	3	12.508	1	3	21.600	0	3	40.001	0	0
2.667	1	4	12.533	0	2	22.100	0	1	41.733	0	2
2.967	0	1	13.467	0	0	22.148	0	2	41.734	0	0
3.328	0	2	13.800	0	2	22.180	0	0	42.311	0	2
3.393	1	3	14.267	0	0	22.180	0	0	42.869	0	0
3.700	1	4	14.475	0	4	22.267	0	0	43.180	0	0
3.803	0	3	14.500	0	1	22.300	0	2	43.279	0	1
4.311	0	1	15.213	0	1	22.500	0	1	43.902	0	2
4.867	0	0	15.333	0	0	22.533	0	1	44.267	0	2
5.180	1	1	15.525	0	1	22.867	0	1	44.475	0	1
6.233	0	2	15.533	0	2	23.738	0	1	44.900	0	1
6.367	0	2	15.541	0	1	24.082	0	1	45.148	0	1
6.600	0	1	15.934	0	0	24.180	0	0	46.451	0	0

Table 1: Graft survival times in months for 148 renal transplant patients, with the number of antigen mismatches, x . Status is 1 for an observed graft failure and 0 for a censored observation.

```

model{

for (i in 1:n)
  {
    is.censored[i]~dinterval(t[i],t.cen[i])
    t[i]~dexp(lambda[i])
    lambda[i]<-exp(beta0+beta.x*x[i])
  }

beta0~dnorm(0.0,0.001)
beta.x~dnorm(0.0,0.001)
}

```

Table 2: Model specification for renal transplant data.

You might like to try fitting a Weibull model instead of the exponential model. In the model specification, change the definition of $t[i]$ to

```
t[i]~dweib(alpha,lambda[i])
```

and add the prior for α at the end.

```
alpha~dgamma(1.1,1.1)
```

2.2 Gastric cancer data

Table 3 shows survival times for two groups of patients with gastric cancer. Group 1 received chemotherapy and radiation. Group 2 just received chemotherapy. The data are taken from Gamerman (1991).

The data are available in the file `gastric.txt` from

<http://www.mas.ncl.ac.uk/~nmf16/teaching/mas8391/>

You can read the data and put them into a form ready for us with commands such as the following.

```

gastric<-read.table("gastric.txt",header=TRUE)
n<-length(gastric[,1])
t<-gastric$t
is.na(t)<-gastric$status==0
is.censored<-1-gastric$status
t.cen<-gastric$t+gastric$status
gastricdata<-list(n=n,t=t,is.censored=is.censored,t.cen=t.cen,
                 group=gastric$group)

```

Group 1						Group 2					
<i>t</i>	Status	<i>t</i>	Status	<i>t</i>	Status	<i>t</i>	Status	<i>t</i>	Status	<i>t</i>	Status
17	1	185	1	542	1	1	1	383	1	778	1
42	1	193	1	567	1	63	1	383	1	786	1
44	1	195	1	577	1	105	1	388	1	797	1
48	1	197	1	580	1	125	1	394	1	955	1
60	1	208	1	795	1	182	1	408	1	968	1
72	1	234	1	855	1	216	1	460	1	977	1
74	1	235	1	1174	0	250	1	489	1	1245	1
95	1	254	1	1214	1	262	1	523	1	1271	1
103	1	307	1	1232	0	301	1	524	1	1420	1
108	1	315	1	1366	1	301	1	535	1	1460	0
122	1	401	1	1455	0	342	1	562	1	1516	0
144	1	445	1	1585	0	354	1	569	1	1551	1
167	1	464	1	1622	0	356	1	675	1	1690	0
170	1	484	1	1626	0	358	1	676	1	1694	1
183	1	528	1	1736	0	380	1	748	1		

Table 3: Survival times of patients with gastric cancer. Status is 1 for an observed death and 0 for a censored observation.

A suitable model specification, with an exponential survival distribution, is shown in Table 4. Type this into a file called `gastricbug.txt`.

Create initial values as follows.

```
tinits1<-gastric$t+5
is.na(tinits1)<-gastric$status==1
tinits2<-tinits1+5
gastricinits<-list(list(t=tinits1),list(t=tinits2))
```

Build the `rjags` model as follows.

```
gastricjags<-jags.model("gastricbug.txt",data=gastricdata,inits=gastricinits,
                        n.chains=2)
```

Try computing and examining the posterior distribution as in the case of the renal transplant data. You might also try using a Weibull distribution.

3 Alternative method by splitting the data

As an alternative to using data augmentation, we can divide the data into observed and censored subsets. This method may be more efficient. As an example, we will use the gastric cancer data as above.

We read the data as before but then partition the data into two subsets.

```
gastric<-read.table("gastric.txt",header=TRUE)
```

```

model{

for (i in 1:n)
  {
    is.censored[i]~dinterval(t[i],t.cen[i])
    t[i]~dexp(lambda[i])
    lambda[i]<-exp(beta0+beta.g[group[i]])
  }

beta0~dnorm(0.0,0.001)
beta1~dnorm(0.0,0.001)
beta.g[1]<-beta1
beta.g[2]<- -beta1
}

```

Table 4: Model specification for gastric cancer data.

```

model{

for (i in 1:n.obs)
  {t.obs[i]~dexp(lambda.obs[i])
   lambda.obs[i]<-exp(beta0+beta.g[group.obs[i]])
  }

for (i in 1:n.cen)
  {status.cen[i]~dbern(S[i])
   S[i]<-pexp(t.cen[i],lambda.cen[i])
   lambda.cen[i]<-exp(beta0+beta.g[group.cen[i]])
  }

beta0~dnorm(0.0,0.001)
beta1~dnorm(0.0,0.001)
beta.g[1]<-beta1
beta.g[2]<- -beta1
}

```

Table 5: Alternative model specification for the gastric cancer example, without data augmentation.

```

gastric.obs<-subset(gastric,status==1)
gastric.cen<-subset(gastric,status==0)
n.obs<-length(gastric.obs$t)
n.cen<-length(gastric.cen$t)
gastricdata<-list(n.obs=n.obs,n.cen=n.cen,t.obs=gastric.obs$t,t.cen=gastric.cen$t,
                 status.cen=gastric.cen$status,
                 group.obs=gastric.obs$group,group.cen=gastric.cen$group)

```

It is advisable to use initial values for the parameters so that the survival probabilities do not become too close to 0 or 1 at the first iteration.

```
gastricinits<-list(list(beta0=-100,beta1=0),list(beta0=-100,beta1=0))
```

The model specification is shown in Table 5.

The two parts of the likelihood, corresponding to the observed events (at times `t.obs`) and the censored observations (at times `t.cen`), are defined separately. The likelihood for the censored observations uses the distribution function `pexp()`.

Try this. You might also like to try using a Weibull distribution, using the following.

```
S[i]<-pexp(t.cen[i],alpha,lambda.cen[i])
```

References

- Gamerman, D. (1991). Dynamic Bayesian models for survival data. *Applied Statistics*, **40**, 63-79.
- Henderson, R. and Milner, A. (1991). Aalen plots under proportional hazards. *Applied Statistics*, **40**, 401-409.