RPSFT models and the Concorde study

1 Rank-preserving structural failure-time models

We provide here an analysis of simulated data similar to the Concorde study (Lancet, 1993). These data have been generated by Bond and Allison, in a vignette accompanying their `R` package `RPSFTM`.

```
library(rpsftm)

head(immdef)

##   id def imm censyrs xo    xoyrs prog  progyrs entry
## 1  1   0   1       3  0 0.000000    0 3.000000     0
## 2  2   1   0       3  1 2.652797    0 3.000000     0
## 3  3   0   1       3  0 0.000000    1 1.737838     0
## 4  4   0   1       3  0 0.000000    1 2.166291     0
## 5  5   1   0       3  1 2.122100    1 2.884646     0
## 6  6   1   0       3  1 0.557392    0 3.000000     0
```

For example, subject 2 was randomised to the deferred arm, started treatment at 2.65 years and was censored at 3 years (the end of the study). Subject 3 was randomised to the immediate treatment arm and progressed (observed the event) at 1.74 years. Subject 5 was randomised to the deferred treatment arm, started treatment at 2.12 years and progressed at 2.88 years. The trial lasted 3 years with staggered entry over the first 1.5 years. The variable censyrs gives the time from entry to the end of the trial. The table below shows summary statistics for the immdef data:

```
library(tableone)
vars       <- c("def", "imm", "censyrs", "xo", "xoyrs",
                "prog", "progyrs", "entry")
factorVars <- c("def", "imm", "xo", "prog")
CreateTableOne(vars=vars, data=immdef, factorVars=factorVars,
               includeNA=FALSE, test=FALSE)

##
##                         Overall
##   n                     1000
##   def = 1 (%)            500 (50.0)
##   imm = 1 (%)            500 (50.0)
##   censyrs (mean (sd)) 2.25 (0.45)
##   xo = 1 (%)             189 (18.9)
##   xoyrs (mean (sd))   0.78 (0.93)
##   prog = 1 (%)           312 (31.2)
##   progyrs (mean (sd)) 1.93 (0.66)
##   entry (mean (sd))   0.75 (0.45)
```

The following is the observed survival in the two arms

```
# Plot of the unadjusted model
library(survival)
plot(survfit(Surv(progyrs, prog)~(imm==1), data = immdef),
             lty=c(1,2), ylab="AIDS-free survival probability",
     xlab="Years since ART start")
legend(1.25, 0.25, c("Immediate ZDV", "Deferred ZDV"), lty=c(1,2))
lrtest.immdef<-survdiff(Surv(progyrs, prog)~(imm==1), data = immdef)
text(2.25,.3, paste("Log-rank p=",
                    format(pchisq(1,lrtest.immdef$chisq),digits=3)))
```

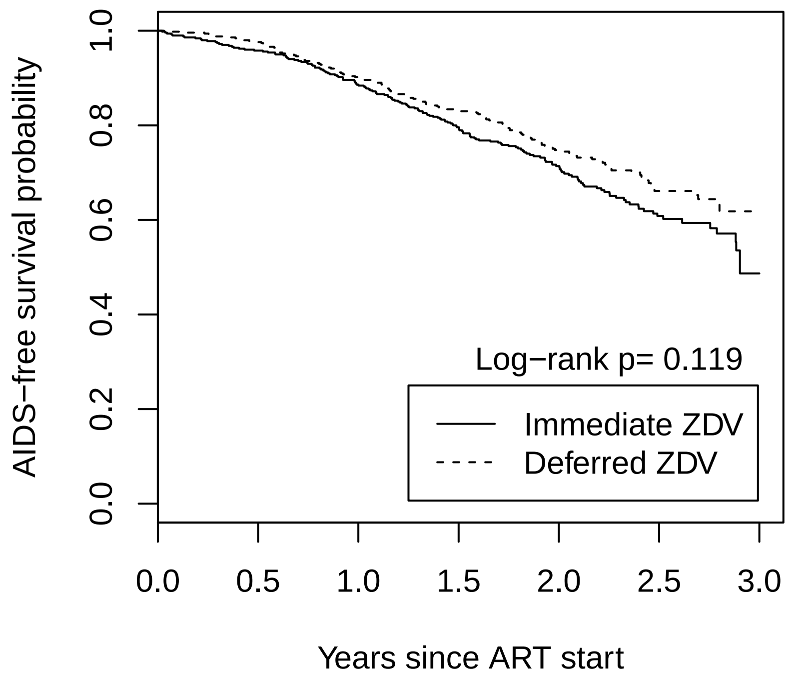

Figure 1: Observed progression-free survival in the immediate versus the deferred ZDV arm

1.1 Analysis through the RPSFT model

We now show how to use `rpsftm` with the `immdef` data. First, a variable `rx` for the proportion of time spent on treatment must be created:

```
rx <- with(immdef, 1 - xoyrs/progyrs)
```

This sets `rx` to 1 in the immediate treatment arm (since no patients could switch to the deferred arm), 0 in the deferred arm patients that did not receive treatment and `1 - xoyrs/progyrs` in the deferred arm patients that did receive treatment. Using the default options, the fitted model is

```
rpsftm_fit_lr <- rpsftm(formula=Surv(progyrs, prog) ~ rand(imm, rx),
                        data=immdef, censor_time=censyrs)

summary(rpsftm_fit_lr)

##   arm   rx.Min. rx.1st Qu. rx.Median   rx.Mean rx.3rd Qu.   rx.Max.
## 1   0 0.0000000  0.0000000 0.0000000 0.1574062  0.2547779 0.9770941
## 2   1 1.0000000  1.0000000 1.0000000 1.0000000  1.0000000 1.0000000
##         Length Class      Mode
## psi        1   -none-     numeric
## fit       14   survfit    list
## CI         2   -none-     numeric
## Sstar   2000   Surv       numeric
## rand    2000   rand       numeric
## ans        5   -none-     list
## eval_z     2   data.frame list
## n          2   table      numeric
## obs        2   -none-     numeric
## exp        2   -none-     numeric
## var        4   -none-     numeric
## chisq      1   -none-     numeric
## call       4   -none-     call
## formula    3   terms      call
## terms      3   terms      call
##
## psi: -0.1810871
## exp(psi): 0.8343627
## Confidence Interval, psi -0.3496948 0.002042503
## Confidence Interval, exp(psi)  0.7049032 1.002045
```

The above formula fits a RPSFTM where `progyrs` is the observed event time, `prog` is the indicator of disease progression, `imm` is the randomised treatment group indicator, `rx` is the proportion of time spent on treatment and censyrs is the censoring time. The log rank test is used in finding the point estimate of ψ, $\hat{\psi}$.

Recensoring is performed since the `censor_time` parameter is specified; if not specified then recensoring would not be performed. After finding $\hat{\psi}$, rpsftm refits the model at $\hat{\psi}$ and produces a `survdiff` object of the counter-factual event times to be used in plotting Kaplan-Meier curves. The function

1. `psi` the estimated parameter

2. `fit` a `survdiff` object to produce Kaplan-Meier curves of the estimated counter-factual event times in each treatment arm using `plot()`
3. `formula` a formula representing any adjustments, strata or clusters used
4. `regression` the survival regression object at `psi`
5. `Sstar` the (possibly) recensored `Surv()` data using psi
6. `ans` the object returned from `uniroot` used to solve the estimating equation
7. `CI`, a vector of the confidence interval around `psi`
8. `call` the R call object
9. `eval_z` a data frame giving values of the Z-statistics at different values of `psi`

The point estimate and 95% confidence interval can be returned using `rpsftm_fit_lr$psi` and `rpsftm_fit_lr$CI` which gives $\hat{\psi} = 0.181$, confidence interval $(-0.35, 0.00199)$. The function `plot()` produces Kaplan-Meier curves of the counter-factual event times in each group and can be used to check that the distributions are indeed the same at $\hat{\psi}$.
We now provide examples of using the Cox regression model and the Weibull model in place of the log rank test. To use the Wald test from a Cox regression model, we specify test=coxph in the function parameters. Covariates can also be included in the estimation procedure by adding them to the right hand side of the formula.

For example, baseline covariates that are included in the intention-to-treat analysis may also be incorporated into the estimation procedure of the RPSFTM. In the following example we add entry time as a covariate and use summary() to find the value of $\hat{\psi}$ and its 95% confidence interval:

```
rpsftm_fit_cph <- rpsftm(formula=Surv(progyrs, prog) ~ rand(imm, rx),
                         data=immdef,
                         censor_time=censyrs,
                         test=coxph)
summary(rpsftm_fit_cph)

##   arm   rx.Min. rx.1st Qu. rx.Median   rx.Mean rx.3rd Qu.   rx.Max.
## 1   0 0.0000000  0.0000000 0.0000000 0.1574062  0.2547779 0.9770941
## 2   1 1.0000000  1.0000000 1.0000000 1.0000000  1.0000000 1.0000000
##   n= 1000, number of events= 286
##
## Concordance= 0.496  (se = 0.016 )
## Rsquare= 0   (max possible= 0.976 )
##
## psi: -0.1810871
## exp(psi): 0.8343627
## Confidence Interval, psi -0.3497012 0.002369558
## Confidence Interval, exp(psi)  0.7048987 1.002372
```

```r
require(ggplot2)
ss1<-plot(rpsftm_fit_lr, main=" ")
ss2<-ss1+theme(legend.position = c(.8,.25))+
  theme(axis.text=element_text(size=10))+
  theme(axis.title=element_text(size=15,face="bold"))+
  ggtitle("")

ss2
```

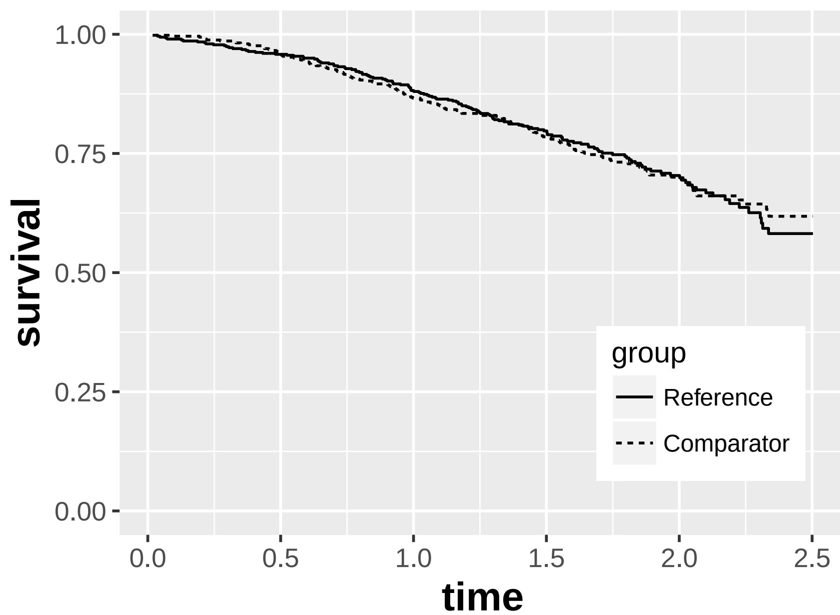

Figure 2: Plot of the the RPSFT model estimates

This means that the revised hazard ratio is $\theta = 1.004$ (essentially no difference between the arms). We can also create the same hazard ratio by hand, by taking into consideration the re-censored observations (these are stored in rpsftm$_{fit_c}ph$regression$y).Thisisdoneasfollows$:

```
summary(coxph(rpsftm_fit_cph$y~immdef$imm))

## Call:
## coxph(formula = rpsftm_fit_cph$y ~ immdef$imm)
##
##   n= 1000, number of events= 286
##
##                  coef exp(coef)  se(coef)     z Pr(>|z|)
## immdef$imm -0.003542  0.996465  0.118285 -0.03    0.976
##
##            exp(coef) exp(-coef) lower .95 upper .95
```

```
require(ggplot2)
ss1<-plot(rpsftm_fit_cph, main=" ")
ss2<-ss1+theme(legend.position = c(.8,.25))+
  theme(axis.text=element_text(size=10))+
  theme(axis.title=element_text(size=15,face="bold"))+
  ggtitle("")
ss2
```

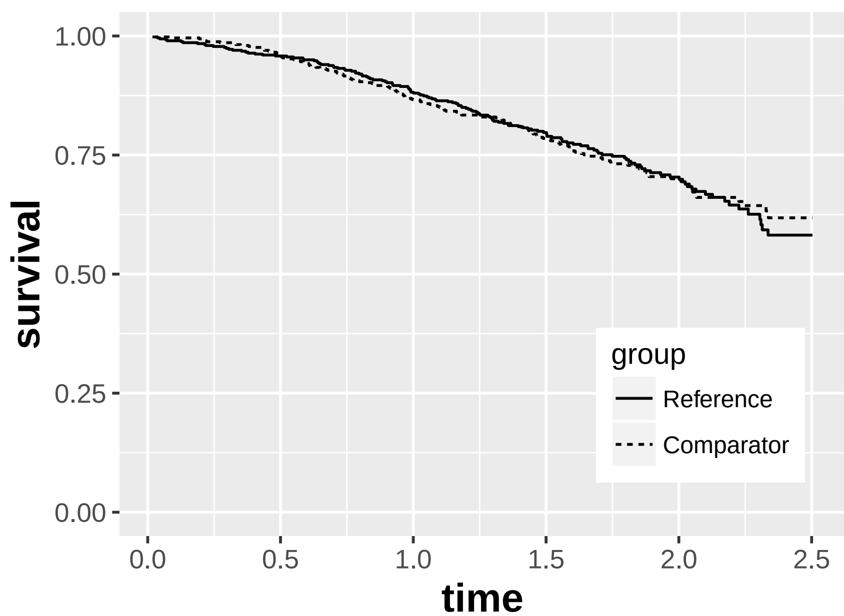

Figure 3: Plot of Cox proportional-hazard model fit

```
## immdef$imm    0.9965      1.004    0.7903     1.256
##
## Concordance= 0.496  (se = 0.016 )
## Rsquare= 0   (max possible= 0.976 )
## Likelihood ratio test= 0  on 1 df,   p=0.9761
## Wald test            = 0  on 1 df,   p=0.9761
## Score (logrank) test = 0  on 1 df,   p=0.9761
```

Similary, for the Weibull model we have:

```
rpsftm_fit_wb <- rpsftm(formula=Surv(progyrs, prog) ~ rand(imm, rx) + entry,
                        data=immdef,
                        censor_time=censyrs,
                        test=survreg)
summary(rpsftm_fit_wb)

##   arm   rx.Min. rx.1st Qu. rx.Median   rx.Mean rx.3rd Qu.   rx.Max.
## 1   0 0.0000000  0.0000000 0.0000000 0.1574062  0.2547779 0.9770941
## 2   1 1.0000000  1.0000000 1.0000000 1.0000000  1.0000000 1.0000000
##
## Call:
## rpsftm(formula = Surv(progyrs, prog) ~ rand(imm, rx) + entry,
##     data = immdef, censor_time = censyrs, test = survreg)
##               Value Std. Error      z        p
## (Intercept)  1.3881     0.0857 16.197 5.34e-59
## entry       -0.0582     0.0906 -0.642 5.21e-01
## Log(scale)  -0.4176     0.0568 -7.349 2.00e-13
##
## Scale= 0.659
##
## Weibull distribution
## Loglik(model)= -759.8   Loglik(intercept only)= -760
## Number of Newton-Raphson Iterations: 6
## n= 1000
##
##
## psi: -0.1811851
## exp(psi): 0.8342809
## Confidence Interval, psi -0.3501459 0.005170935
## Confidence Interval, exp(psi)  0.7045852 1.005184
```

Investigation of a plot of $Z(\psi)$ against ψ (example shown below) for a range of values of ψ could show why the functions fails to find a root. The fitted object, `rpsftm_fit`, a data frame (`rpsftm_fit$eval_z`) with values of the Z-statistic evaluated at 100 points between the limits of the search interval. It is therefore straightforward to plot $Z(\psi)$ against ψ.

```r
plot(rpsftm_fit_wb, main="")
```

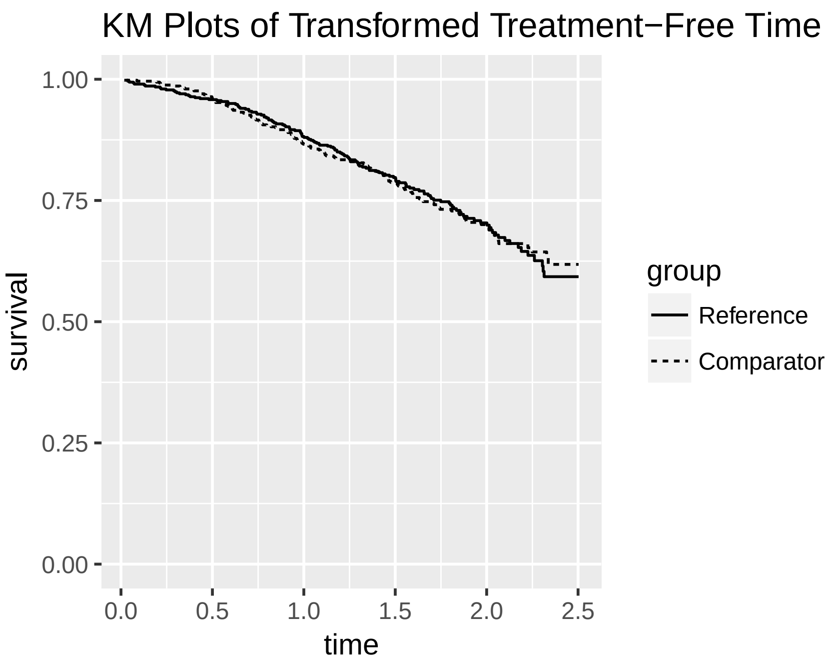

Figure 4: Plot of the RPSFT model from the Weibull regression model

```r
rpsftm_fit <- rpsftm(formula=Surv(progyrs, prog) ~ rand(imm, rx),
                     data=immdef, censor_time=censyrs,
                     low_psi=-0.5, hi_psi=.1)
summary(rpsftm_fit)
```

```
##   arm   rx.Min. rx.1st Qu. rx.Median   rx.Mean rx.3rd Qu.   rx.Max.
## 1   0 0.0000000  0.0000000 0.0000000 0.1574062  0.2547779 0.9770941
## 2   1 1.0000000  1.0000000 1.0000000 1.0000000  1.0000000 1.0000000
##         Length Class      Mode
## psi        1   -none-     numeric
## fit       14   survfit    list
## CI         2   -none-     numeric
## Sstar   2000   Surv       numeric
## rand    2000   rand       numeric
## ans        5   -none-     list
## eval_z     2   data.frame list
## n          2   table      numeric
## obs        2   -none-     numeric
## exp        2   -none-     numeric
## var        4   -none-     numeric
## chisq      1   -none-     numeric
## call       6   -none-     call
## formula    3   terms      call
```

```
## terms      3   terms      call
##
## psi: -0.1812513
## exp(psi): 0.8342257
## Confidence Interval, psi -0.3497069 0.002004934
## Confidence Interval, exp(psi)  0.7048946 1.002007
```

```
plot(rpsftm_fit$eval_z, type="s", ylim=c(-2, 6))
abline(h=qnorm(c(0.025, 0.5, 0.975)))
abline(v=rpsftm_fit$psi)
abline(v=rpsftm_fit$CI)
```

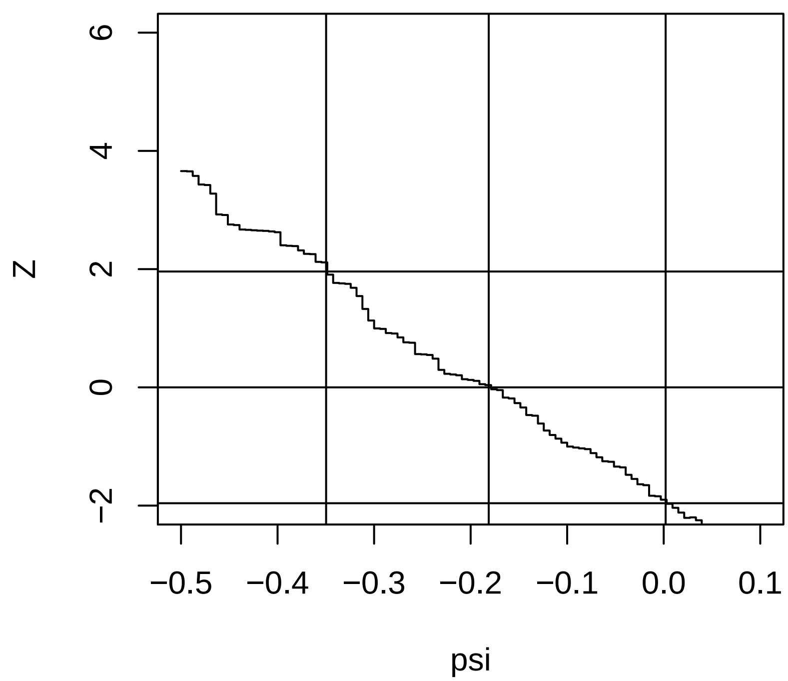

Figure 5: Plot of $Z(\psi)$ versus ψ