Simple Parametric and Piecewise Methods

Philip Cooney

16 September 2019

Contents

Check Likeilhood plot switch labels

Following this, I will provide background to a publically available dataset. This dataset will be used to illustrate some of the concepts discussed in the introduction. Distributional Assumpions (maybe see what censors need to be there to identify the appropriate distribution), Piecewise hazards

0.1 Introduction to a publically available Cancer Dataset

In order to consider the various issues in survival analysis it is useful (if not essential) to have real-world data. There a number of datasets which have survival outcomes, however, an ideal dataset has a large number of observations, information on covariates and multiple outcomes (i.e. PFS and OS).

The E1690 dataset available online has many of these attributes. This dataset is a combination of two randomized control trials which evaluated the efficacy of high-dose Interferon alpha2b (HDI) for 1 year and low-dose interferon alpha2b (LDI) for 2 years versus Obs in high-risk (stage IIB and III) melanoma with replase free survival (RFS) and overall survival (OS) end points. The eariler trial E1684 observed a larger than expected treatment effect and as a result a second trial (E1690) was begun in 1991 to attempt to confirm the results of E1684. Further details are available in Kirkwood et al 2000.

Covariates in the dataset include treatment (x1: IFN, OBS), age (x2), sex (xa), logarithm of Breslow depth (x4), logarithm of size of primary (xs), and type of primary tumor(x6). The dataset includes only those observations assigned to high dose interferon and observation arms. Time to event was available for both OS and RPS, however, for consitency I will refer to any relapse event as a progression event and therefore refer to this as PFS (progression free survival).

0.2 Descriptive analysis of the data

Plotting the survial for both events of both arms provides an illustration that the inteferon group has a higher expected survival in the E1684 trial Figure 1.

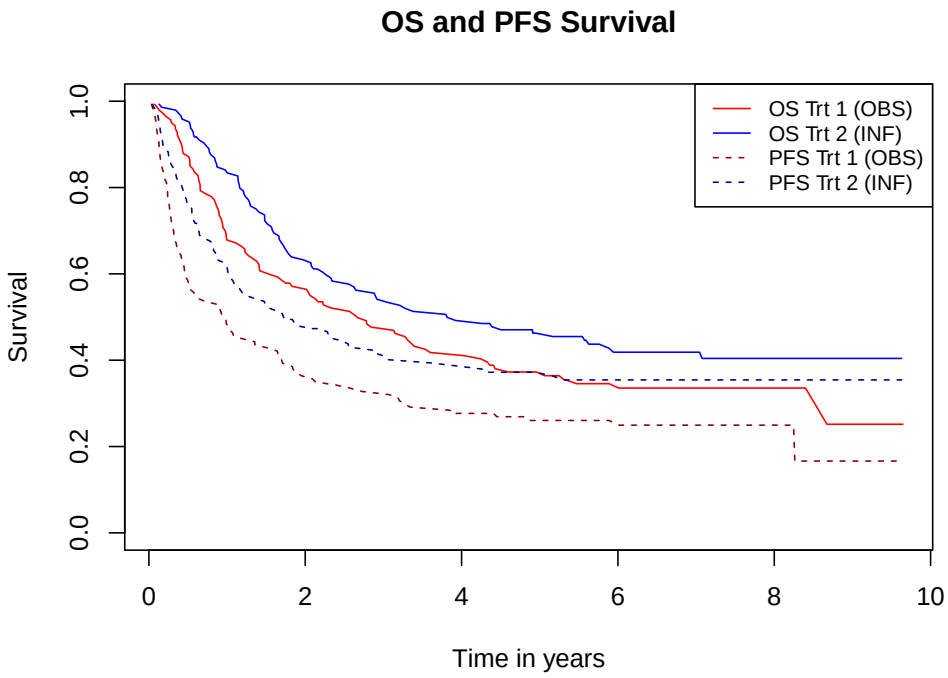

Figure 1: PFS and OS outcomes in E1684 trial

Focusing in on the OS and PFS outcomes seperately (Figures 2 & 3 respectively) we can see censoring the presence of censoring denoted by the vertical tick marks and compute the log rank test. This test evaluates whether or not KM curves for two or more groups are statistically equivalent.

The pvalue is 0.064 suggesting that we do not have do not have evidence to indicate that the true (population) OS survival curves are different at a 5% significance level. However, the PFS survival curves are different at a 5% significane level.

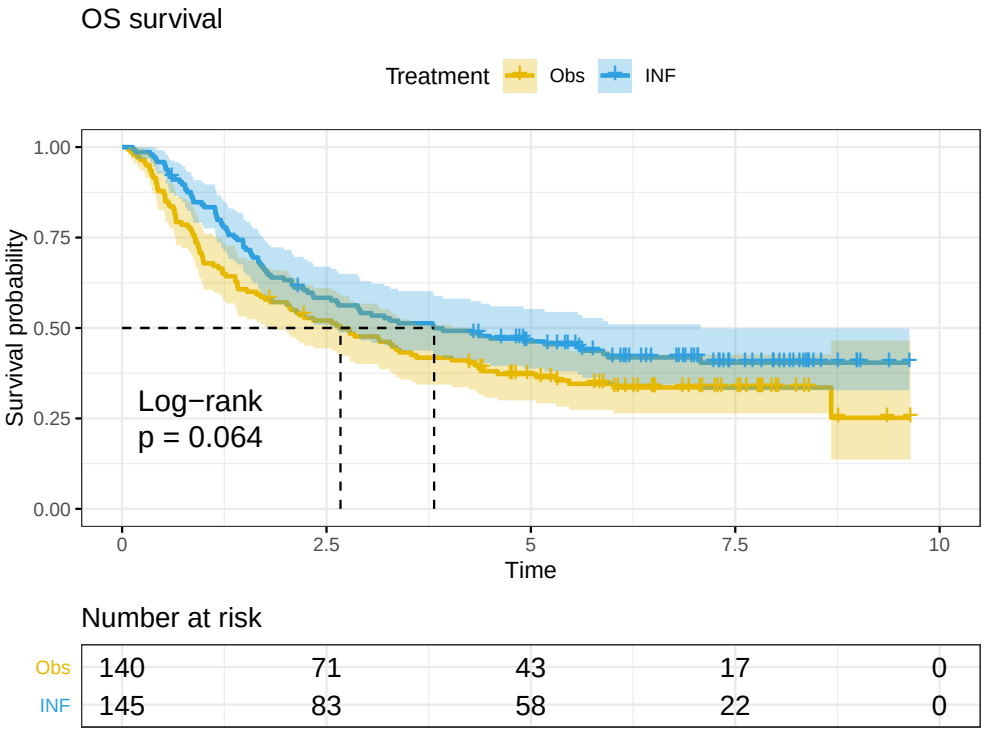

Figure 2: OS outcomes in E1684 trial

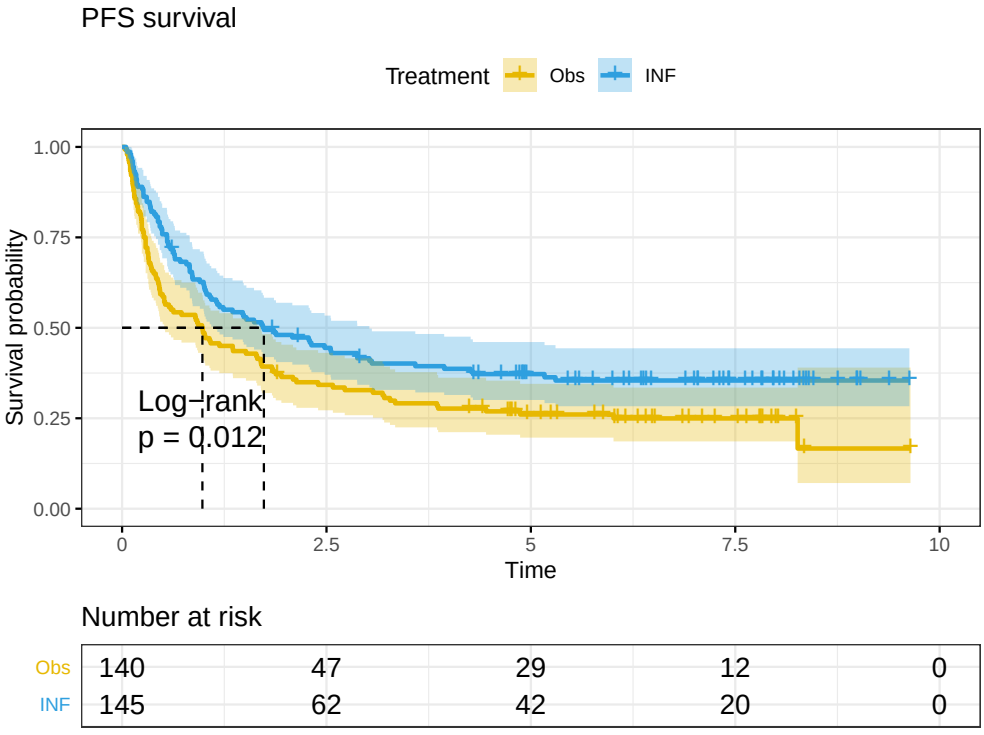

Figure 3: PFS outcomes in E1684 trial

0.3 Exploration of observed hazards

The next feature to consider of this data is how the hazards behave. From the Kaplan Meier plots presented in the previous section it appears that the hazards decrease over time with the survival plot "levelling off" after around 4 to 5 years. One way to assess this is to plot the hazards over time.

In Figures 4 & 5(created using the pehaz function from the muhaz package in R), the time is divided into bins of equal width, and then estimates the hazard in each bin as the number of events d_i in that bin divided by the number of patients at risk in each interval, n_i; the hazard for that interval is $h_i = \frac{d_i}{n_i}$. Check page 32 of collett for different approach

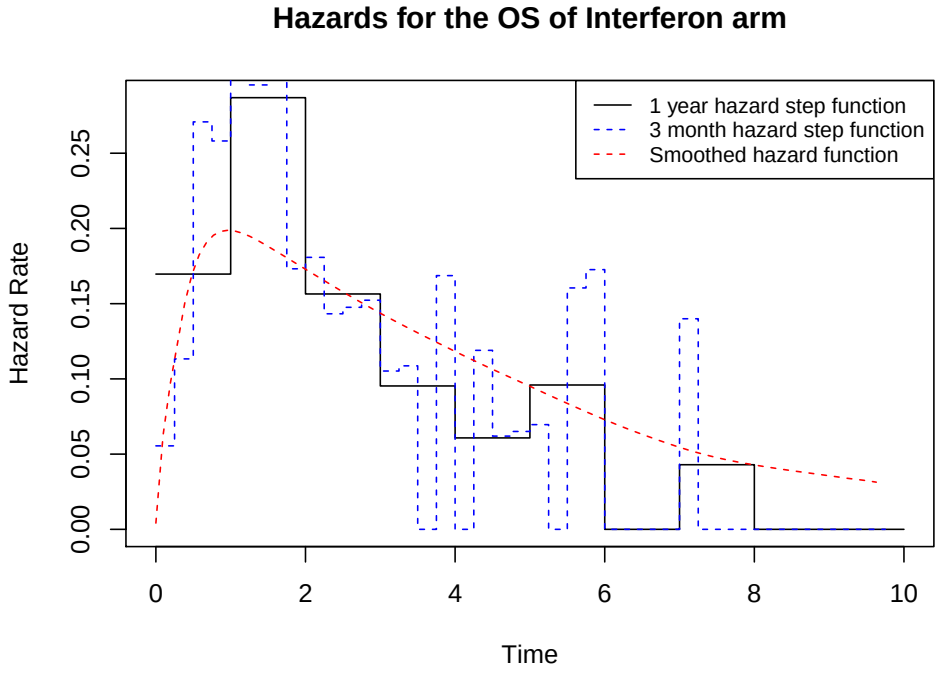

Figure 4: PFS outcomes in E1684 trial

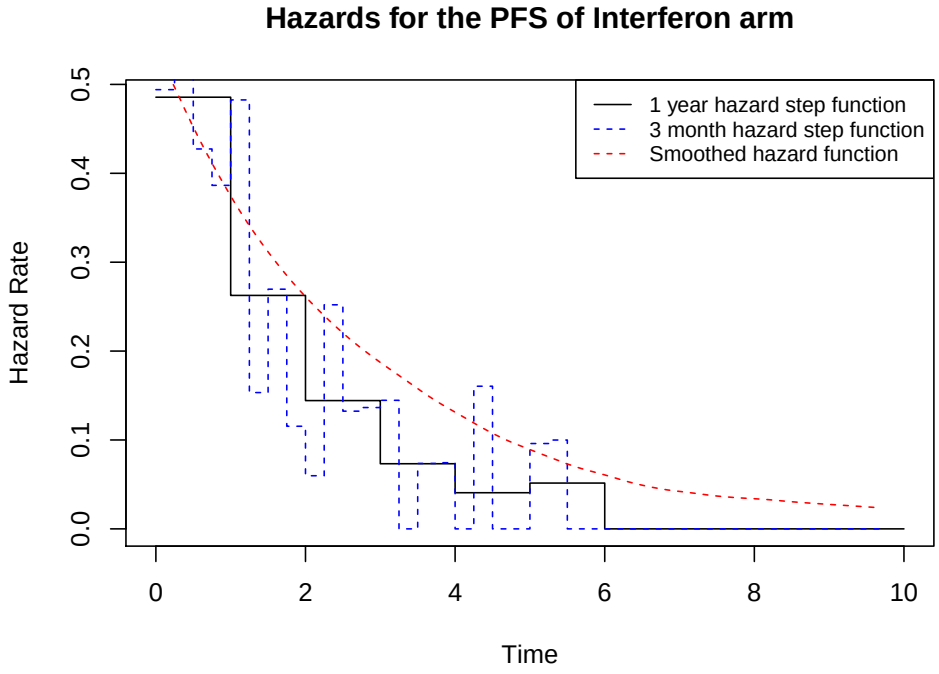

Figure 5: PFS outcomes in E1684 trial

It is evident from the Hazard plots above, that the hazard functions (particularly the one month harzard) jumps around quite a bit from one interval to the next, which limits its utility in visualizing the hazard function. This behaviour is observed because within the shorter timeframe there may be periods where several events may occur at a similar time due to random chance followed by another period in which few events occur. Additionally for investigator assessed outcomes such as (some types of) PFS, patients may be assessed at particular intervals and therefore events may occur in clusters.

To aid visizualization of the hazard, we may compute a smooth hazard estimate. This smooth hazard is computed using a "kernel smoother". A kernel is a function $K(u)$, which we center at each failure time. Typically we choose a smooth-shaped kernel, with the amount of smoothing controlled by a parameter b. The estimate of the hazard function is given by:

$$\hat{h}(t) = \frac{1}{b}\sum_{i=1}^{D} K(\frac{t-t_i}{b})\frac{d_i}{t_i}$$

where $t_1 < t_2 < \ldots . < t_D$ are distinct ordered failure times, the subscript "$(\ i)$" in t_i indicates that this is the i'th ordered failure time, d_i is the number of deaths at time t_i and n_i is the number at risk at that time.

One method to define the Kernel is known as the "Epanechnikov" kernel where $K(u) = \frac{3}{4}(1-u^2)$ defined for $-1 \leq u \leq 1$, and zero elsewhere. In the above formula for the hazard, there is one kernel function placed at each failure time, scaled by the smoothing parameter b. Larger values of b result in wider kernel functions, and hence more smoothing.

Fitting of this smoothed hazard can be accomplished using the muhaz function (again from the muhaz package in R). Selection of the appropriate amount of smoothing is one of the most difficult problems in non-parametric hazard estimation. If the bandwidth parameter is too small, the estimate may gyrate widely. If too wide a parameter is chosen the hazard function may be too smooth to observe real variations in the hazard function over time. The "muhaz" function includes an automatic method for selecting a variable

width bandwidth, so that for time regions with few events, a wider smoothing parameter is used than for time regions densely populated with events.

In the muhaz package a number of grids are defined (default of 101) and the optimal bandwidth at a grid point is obtained by minimizing the local MSE (Mean square error) Muller and Wang 1994. Another option is "knn" - k nearest neighbors distance bandwidth based on Gefeller and Dette (1992). See also

In summary the plots for both outcomes indicate that the hazard of an event decreases over time and for PFS outcome falls to zero at 6 years. Additionally we can use the smoothed hazard function is to obtain a smooth estimate of the survival function, using the relationship $\widetilde{S(t)} = e^{-\int_{u=0}^{t} \hat{h}(u)\,du}$, in which the hazard is evaluated for each grid. These survival functions are presented in Figures 6 & 7. Neither smoothed survival estimate provides a good fit to the data suggesting a loss of precision when smoothing the hazard.

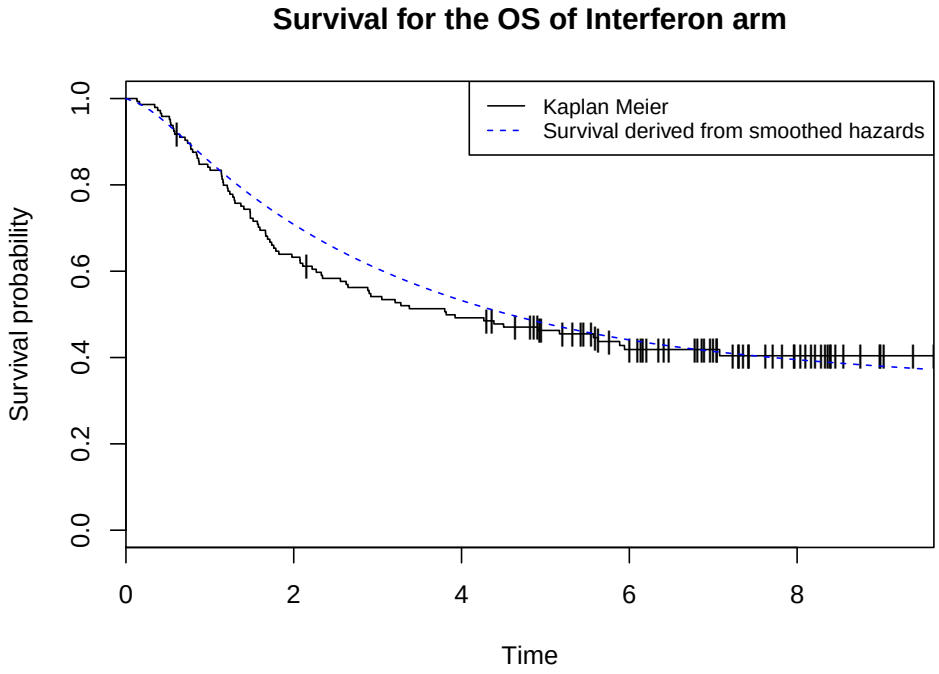

Figure 6: PFS outcomes in E1684 trial

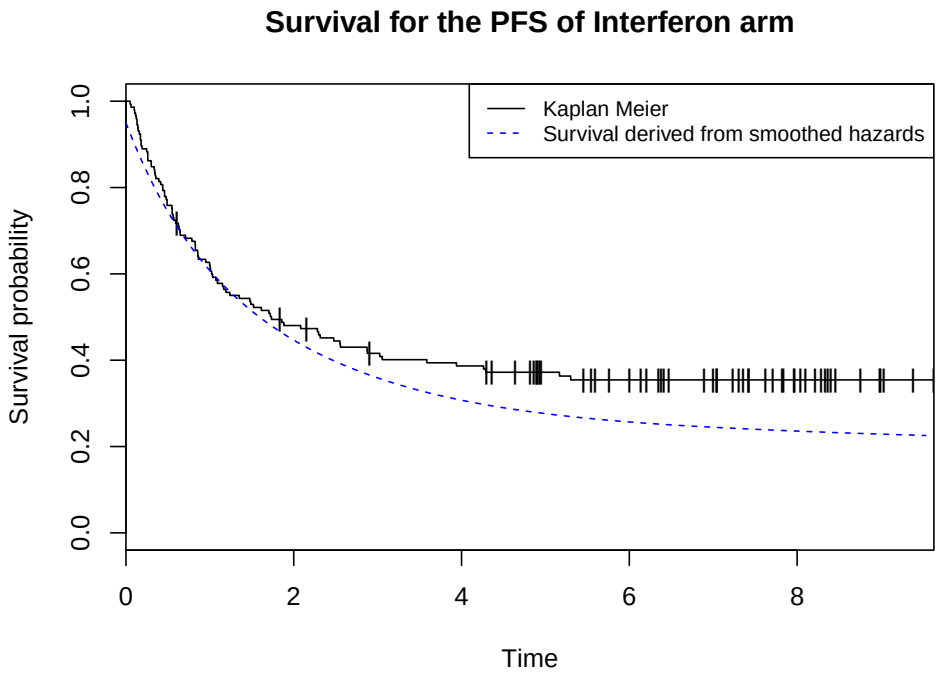

Figure 7: PFS outcomes in E1684 trial

1 Checking distributional assumptions

1.1 Piecewise hazard models

The methods used to up to this point to identify the expected survival and hazard are non-parametric. As these methods do not specify any functional form of the data, their assumptions are robust irrespective of the observed data, however, they cannot be used to make out of sample predictions.

One model which could be used to make predictions (or extrapolations) is the piecewise constant model.

Let $X_1,\ldots..,X_n$ dentote independent identically distributed survival times and $C_1,\ldots\ldots,C_n$ be the censoring times which are assumed to be independent of X. We observed only the pairs $(T_i, \delta_i), i = 1, 2, ..., n$ where $T_i = min(X_i, C_i)$ and $\delta_i = 1$ if $X_i \leq C_i$ and zero otherwise.

We could assume a number of intervals with τ_i indicating the time at which a new interval begins. Within each of these intervals we assume a constant hazard α_i between the time points τ_{j-1} and τ_j.

$$\lambda(t) = \begin{cases} \alpha_1 & 0 \leq t < \tau_1 \\ \alpha_2 & \tau_1 \leq t < \tau_2 \\ . & \\ . & \\ \alpha_{K+1} & t \geq \tau_k \end{cases}$$

Let $X(t)$ denote the number of deaths observed up to time t:

$$X(t) = \sum_{i=1}^{n} I(T_i < t)\delta_i$$

For a given set of $\tau_i's$ the maximum likelihood estimates (MLE's) of the parameters $\alpha_1,\ldots,\alpha_k$ are given by:

$$\hat{\alpha_1} = \frac{X_{(\tau_1)}}{\sum_{i=1}^{n} T_i \wedge \tau_1}$$

$$\hat{\alpha_2} = \frac{X_{(\tau_2)} - X_{(\tau_1)}}{\sum_{i=1}^{n} T_i \wedge (\tau_2 - \tau_1) I(T_i > \tau_1)}$$

$$\hat{\alpha_{k-1}} = \frac{X_{(\tau_{k-1})} - X_{(\tau_{k-2})}}{\sum_{i=1}^{n} T_i \wedge (\tau_{k-1} - \tau_{k-2}) I(T_i > \tau_{k-2})}$$

and

$$\hat{\alpha_{k-1}} = \frac{n_u - X_{(\tau_{k-1})}}{\sum_{i=1}^{n} (T_i - \tau_{k-1}) I(T_i > \tau_{k-1})}$$

where where n_u is the total number of non-censored events.

The log-Likelihood of this model is:

$$logL(\alpha_1, ..., \alpha_k, \tau_1, ..., \tau_{k-1}) = X(\tau_1) log\alpha_1 + [X(\tau_2) - X(\tau_1)] log\alpha_2$$

$$+[n_u - X(\tau_{k-1})] log\alpha_k - .. - \alpha_1 \sum_{i=1}^{n} (T_i \wedge \tau_1) - \alpha_2 \sum_{i=1}^{n} (T_i \wedge \tau_2 - \tau_1) I(T_i > \tau_1) - .. - \alpha_k \sum_{i=1}^{n} (T_i - \tau_{k-1}) I(T_i > \tau_{k-1})$$

Although strictly speaking this is a parameteric model, the model can accomodate any shaped hazard through the introduction of more frequent breakpoints. A piecewise exponential model fit to the Interferon arm of the E1984 dataset with 10 changepoints (coloured in green) is plotted in 8 below. The model is fit using the pchreg package in R and then predictions were made by extrapolating the last hazard from the last interval (red line).

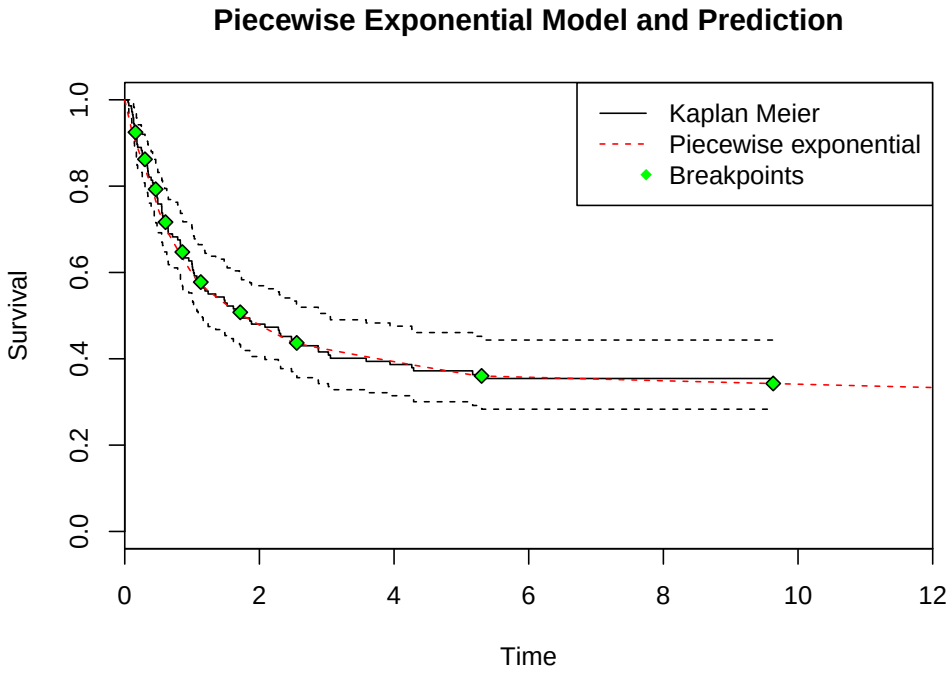

Figure 8: PFS outcomes in E1684 trial

One of the key issues with piecewise models is indentifying the location and number of changepoints; if too few intervals are chosen, the model may fail to accurately capture the variation of the observed hazard while if too many are chosen the extrapolation may be unduly influenced by the final few observations. In the pchreg package the default behaviour (when the location of the changepoints are not supplied is to use the empirical quantiles of event times (i.e. time points are based on quantiles of events). In the following section I review several approaches that have been suggested to find the appropriate number of breakpoints and their locations.

1.1.1 Goodman et al

Goodman et al noted that the log-Likelihood for a given set of change points is:

$$logL(\tau_1, ..., \tau_{k-1}) = X(\tau_1) log\left[\frac{X(\tau_1)}{\sum_{i=1}^{n} T_i \wedge \tau_1}\right] +$$
$$+[X(\tau_2) - X(\tau_1)] log\left[\frac{X(\tau_2) - X(\tau_1)}{\sum_{i=1}^{n}(T_i \wedge \tau_2 - \tau_1) I(Ti > \tau_1)}\right] + ...$$
$$+[n_u - X(\tau_{k-1})] log\left[\frac{n_u - X(\tau_{k-1})}{\sum_{i=1}^{n}(T_i - \tau_{k-1}) I(Ti > \tau_{k-1})}\right] - n_u$$

Goodman et al used the Nelder-Mead optimization function to maximize the likelihood of this model. However, when I used this algorithm to maximize the log-likelihood I believe that the algorithm would get stuck at one of the many local maxima of the likelihood surface and therefore would be very sensitive to the choice of intial values. Figure 9 appears the support this assumption.

```
optim(par=c(0.7, 2 ), fn=nelder.mead.piecewise.pchreg, method= "Nelder-Mead",
      control=list(fnscale = -1),
      time=trt.df$FAILTIME, status=trt.df$FAILCENS)
```

```
## $par
## [1] 1.19178 1.88219
##
## $value
## [1] -201.3611
##
## $counts
## function gradient
##      251       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

```
optim(par=c(1, 3 ), fn=nelder.mead.piecewise.pchreg, method= "Nelder-Mead",
      control=list(fnscale = -1),
      time=trt.df$FAILTIME, status=trt.df$FAILCENS)
```

```
## $par
## [1] 1.191780 3.054792
##
```

```
## $value
## [1] -196.3845
##
## $counts
## function gradient
##      175       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

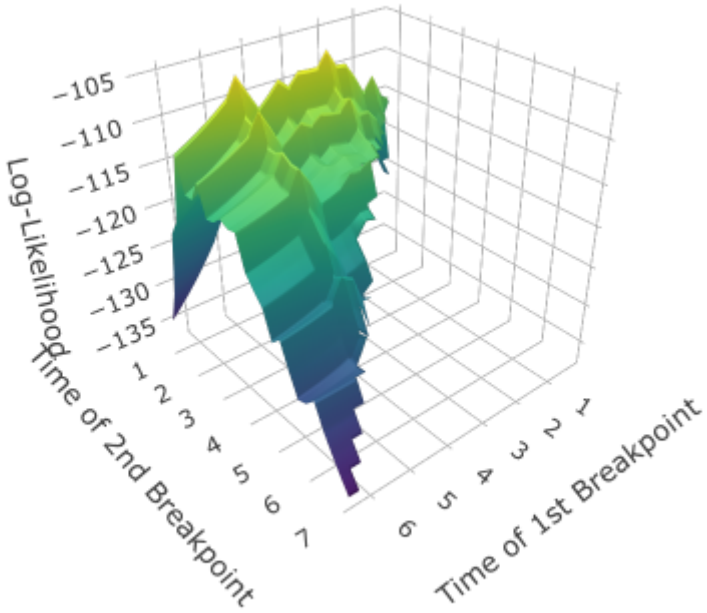

Figure 9: Log-Likelihood surface

In order to evaluate the log-likelihood, I have used a grid search approach, whereby I evaluate the likelihood across a grid of timepoints and select the change point which maximizes the likelihood.

```
result <- result[[2]]
result[which(result$Likelihood == max(result$Likelihood, na.rm = T)),]
```

```
##     X1  X2  X3      X4 Likelihood
## 110  0 1.2 3.2 9.63014  -199.2592
```

1.1.2 Zhang (Least squares approach)

Another approach was suggested by Zhang 2014 in which he suggested that the function on page 56 be minimized.

I understand $Y_n^*(x_j)$ to be the average hazard as obtained from the Nelson Aalen cumulative hazard (based on Equation 5.4 on page 27) . My understanding is that t_{in} is a potential change point and tj is a time point at which the Nelson Aalen cumulative hazard is computed (i.e. it is computed at every event time). This is based on the information presented in the Equation for $Q(\Theta, x_i)$ for a single changepoint (pg 28). However in the second line and third line I do not know why the notation is changed to x_{im}?

I tried to implement this function in R, however because of the grid search and the fact that it needs to use the Nelder Mead optimizer at every point it is quite slow. I assume that there are bugs in this as with grid values the optimal hazards are very low, but I haven't spend much more time trying to fix them as I don't know if this method is very useful compared to the Goodman approach.

1.1.3 Zhang (Counting proccess approach)

Zhang also discusses a counting process approach which was first proposed by Chang et al 1994. They define $Y(x)$ (different to the average Nelson Aalen hazard) on pg 11, however, it also seems to be defined differently in Chang et al 1994. In any event I'm pretty sure that I implemented it incorrectly as the $Y(x)$ function should increase on the interval $(0, \tau]$ and decrease thereafter when $\alpha_2 > \alpha_1$, however it is clear from previous analysis that the hazards decrease and $\alpha_2 < \alpha_1$. Figure 10 appears to indicate that $\alpha_2 > \alpha_1$.As per my understanding the first changepoint is the first point at which the function begins to decrease. In later chapters Zhang proposes extensions to this method to account for selection of multiple change points.

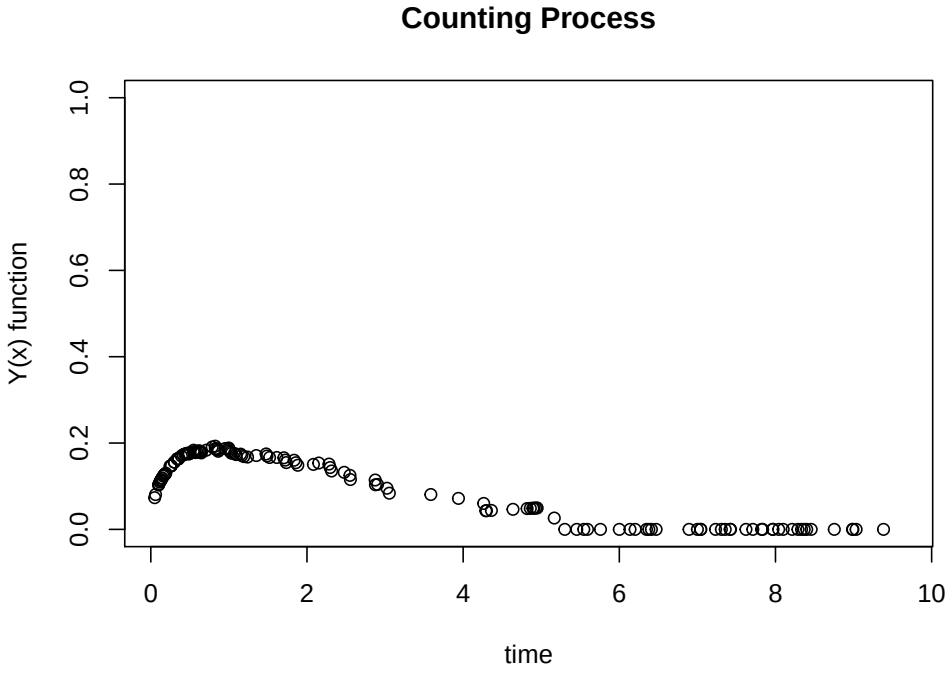

Figure 10: Counting process method

1.1.4 Summary

Based on my initial research the most robust (albeit quite a computationally intense) approach is to evaluate the log-likelihood for a large number of change points. The combination of changepoints that maximizes the log-liklihood is the best estimate of the change point.

Based on Goodman et al I wrote a piecewise.exponential function for which code is provided below:

```
piecewise.exponential
```

```
## function (breakpoints = NULL, time, status)
## {
##     time <- time
##     status <- status
##     df <- data.frame(time = time, status = status)
##     df <- df[order(time), ]
##     log.likelihood <- -sum(df[, "status"])
##     if (length(breakpoints) > 0) {
##         breaks <- c(0, breakpoints, max(time) + 0.001)
##         df$timepoints <- cut(df$time, breaks = breaks, right = FALSE)
##         df <- cbind(df, model.matrix(~timepoints + 0, data = df))
##         df <- df[, -which(colnames(df) == "timepoints")]
##         if (any(apply(df[, 3:(ncol(df) - 1)], 2, function(x) all(x ==
##             0)))) {
##             log.likelihood = NA
##             return(log.likelihood)
##         }
##         else {
##             ncol.df <- ncol(df)
##             piecewise.haz <- rep(NA, length(breakpoints) + 1)
##             nrows <- nrow(df)
##             j <- 0
##             t.prev <- 0
##             for (i in 3:ncol.df) {
##                 j <- j + 1
##                 temp.df <- df[min(which(df[, i] == 1)):nrow(df),
##                   ]
##                 nrows <- nrow(temp.df)
##                 end.point <- max(which(temp.df[, i] == 1))
##                 if (end.point < nrows) {
##                   temp.df[end.point:nrows, "status"] <- 0
##                   temp.df[end.point:nrows, "time"] <- breakpoints[j]
##                 }
##                 temp.df$time <- temp.df$time - t.prev
##                 log.likelihood.new <- sum(temp.df[, "status"]) *
##                   log(sum(temp.df[, "status"])/sum(temp.df$time))
##                 if (is.nan(log.likelihood.new)) {
##                   log.likelihood.new <- 0
##                 }
##                 log.likelihood <- log.likelihood.new + log.likelihood
##                 t.prev <- breakpoints[j]
##             }
##             return(log.likelihood)
##         }
##     }
##     else {
##         log.likelihood <- sum(df[, "status"]) * log(sum(df[,
##             "status"])/sum(df$time)) + log.likelihood
##         return(log.likelihood)
##     }
```

```
## }
```

The function produces the same log-likelihood as the one parameter exponential model, however the log-likelihood is different when comparing against the piecewise model from the pchreg package.

Until I can validate my function I have used the pchreg function. Using the hesim package I have simulated 100 observations arising from a piecewise exponential distribution. The gridwise search was able to identify the correct change points.

```
#https://cran.r-project.org/web/packages/hesim/hesim.pdf
rate <- c(.6, 1, 0.2,2)
n <- 500
ratemat <- matrix(rep(rate, n/2), nrow = n,
                  ncol = 4, byrow = TRUE)
t <- c(0, 1, 2,4)
samp <- rpwexp(n, ratemat, t)
summary(samp)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.002513 0.481349 1.141075 1.503153 1.814374 5.392287
```

```
event <-  rep(1,n)

result <- grid.search.piecewise.pchreg(min.break = 0.2,
                      max.break = max(samp+0.1),
                      grid.width = 0.1,
                      num.breaks = 3,
                      min.break.width = 0.5,
                      time = as.vector(samp),
                      status = as.vector(event))
```

```
## [1] "Number of evaluations = 7110"
```

```
result[which(result$Likelihood == max(result$Likelihood, na.rm = T)),]
```

```
##       X1  X2 X3 X4       X5 Likelihood
## 12191  0 1.3  2  4 5.392287  -625.4883
```

```
result_quantile <-  mutate(result, quantile_rank = ntile(result$Likelihood,50)) %>% arrange(desc(Likeli
```

Next Steps:

Implement the spending function hypothesized by Goodman et al. Perform some tests/ (maybe a simulation study) to assess the preformance of this approach in the presence of censoring. *Check what hazards are typically observed in a clincial trial and estimate the survival extrapolations (It will be interesting to see how misspecifications affect the extrapolation... which is what we are interested in)* Identify why the pchreg function produces a different log-likelihood than my function. *Implement the piecwise linear model which appeared to have better performance?

Discuss some ideas for other research topics.