5-2017

Comparision of Survival Curves Between Cox Proportional Hazards, Random Forests, and Conditional Inference Forests in Survival Analysis

Brandon Weathers

Richard Cutler Dr.
Utah State University

COMPARISON OF SURVIVAL CURVES BETWEEN COX PROPORTIONAL HAZARDS, RANDOM SURVIVAL FORESTS, AND CONDITIONAL INFERENCE FORESTS IN SURVIVAL ANALYSIS

A Project Presented to the
Department of Mathematics and Statistics

Utah State University

In Partial Fulfillment
of the Requirements for the Degree

Master of Statistics

Utah State University

by

Brandon Weathers and D. Richard Cutler

April 2017

Supervisory Committee

Dr. Chris Corcoran

Dr. Richard Cutler

Dr. Luis Gordillo

Abstract

Survival analysis methods are a mainstay of the biomedical fields but are finding increasing use in other disciplines including finance and engineering. A widely used tool in survival analysis is the Cox proportional hazards regression model. For this model, all the predicted survivor curves have the same basic shape, which may not be a good approximation to reality. In contrast, Random Survival Forests does not make the proportional hazards assumption and has the flexibility to model survivor curves that are of dissimilar shapes for contrasting groups of subjects. We applied both techniques to a number of publicly available datasets and compared their fits using prediction error curves and the concordance index. In this process we identified 'types of data' in which Random Survival Forests may be expected to outperform the Cox model.

Important Concepts

Survival analysis is a collection of statistical techniques originally developed for analyzing lifetime data for biological organisms. This includes people as well as failure time data for electronic/mechanical systems and components. Survival analysis methods can be used to analyze the time until an event has occurred and have been applied to many other areas of inquiry including economics ("duration analysis") and sociology.

One way that survival (reliability) data differs from data in other areas is through the notion of censoring. Allison (2010, p. 416) defines three main types of censoring. By far the most common type of censoring is right censoring, which arises when either the study ends before the subject experiences the event, or the subject drops out of the study prematurely and before the event occurs. In this case, the exact time an event takes place is not known, but the event is known to have taken place at some point following the censoring time. A second type of censoring is interval censoring in which the exact time the event took place is not known, but it is known to have taken place in a known time interval. The third, and least common type of censoring, is left censoring, and occurs when one is unaware of the exact time the event occurred, but it is known that the event occurred before a certain time. Right censoring will be the only censoring type discussed in the remainder of the paper.

A key concept in survival analysis is the survivor, or reliability, function, which represents the probability that a subject lives beyond a particular time point t. That is, $S(t) = P(T > t) = 1 - F(t)$, where $F(t)$ is the cumulative distribution function of the lifetime, T. Estimation of the survival function may be carried out parametrically or non-parametrically depending on the validity of model assumptions. Lifetimes rarely are approximately normal in distribution; thus, distributional models for lifetime data such as the gamma distribution, Weibull distribution, and exponential distribution / extreme value distribution are more commonly used. These distributions may be fit to censored or uncensored data.

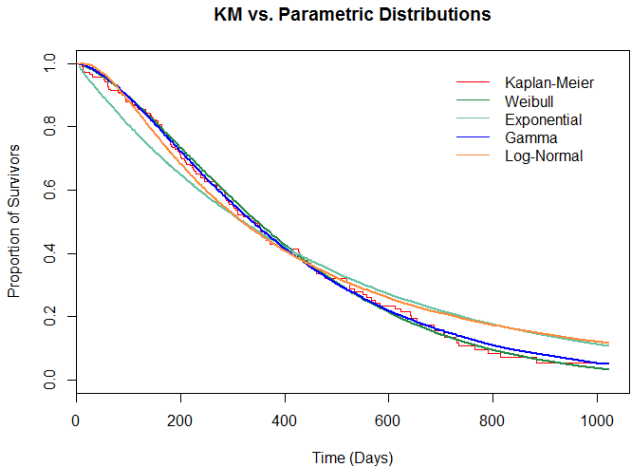

Figure 1: Comparing the Kaplan-Meier Curve to Multiple Parametric Models (LC data).

Kaplan-Meier Estimator

For a single subpopulation of individuals, a simple and widely used non-parametric estimator of the survival function is the Kaplan-Meier estimator, which is a non-parametric way to assess both the number of events that have occurred, such a death, for each unit of time, and the duration of time until an event occurs or recurs. In some cases, Kaplan-Meier estimates may aid in the prediction of specific outcomes, such as the amount of time until a couple will successfully conceive a child.

The survival curves, that are made up of the Kaplan-Meier estimates, are of a stepwise form, such that the values of the survival probability estimate, y, only change for times at which we actually observe the occurrence of an event, or the censoring of a survival time. The value of the survival curve is held constant for the time between two observed events.

Let t_i represent the time in which the event of interest occurred for the i^{th} observation where $i \in \{1, ..., n\}$, n being the total number of observations. Let d_i represent the number of subjects with an observed event time t_i and r_i be the number of subjects at risk at a time before t_i. Then,

$$\hat{S}(t) = \begin{cases} 1 & \text{if } t_j < t_1 \\ \prod_{j=1}^{i} [1 - \frac{d_j}{r_j}] & \text{if } t_1 \leq \ t_j \leq t_n \end{cases} .$$

The estimated variance of this function is given by:

$$\hat{V}[\hat{S(t)}] = [\hat{S}(t)]^2 \hat{\sigma}_S^2(t) = [\hat{S}(t)]^2 \sum_{j=1}^{i} \frac{d_j}{r_j(r_j - d_j)}.$$

If one is interested in computing Kaplan-Meier survival curves for multiple groups (e.g., men and women), one would estimate a different survival function $\hat{S}(t)$ for each group. The survival functions for the two groups may be formally compared using the log-rank test

(Kishore, Goel, & Khanna, 2010), for which the null hypothesis tests equality of the underlying (population) survival curve for the two subpopulations.

One shortcoming of the Kaplan-Meier estimator is that extrapolations are typically not justifiable as some of the censored subjects make it to the end of the study without the event occurring. In other words, although it might be known that the event must occur at some future time point, we only know that the subject lasted the duration of the study without the event occurring. Thus, we cannot make predictions about a right censored subject's event time without additional assumptions.

Cox Proportional Hazards Model

In many applications it is desirable to relate survival times and estimated survival curves to predictor (explanatory) variables, including possible treatments, age, gender, as well as many other characteristics. The Cox Proportional Hazards model is widely used in such circumstances. The Cox model is usually described as being semi-parametric because the component related to the predictor variables is parametric while the estimate of the survival function component is fully non-parametric and makes no assumption about the underlying distribution of the survival function. The covariates of the model can be continuous or categorical in nature and if they are categorical with more than 2 levels, can be converted into a series of binary classes in order to perform the regression.

Two assumptions that are critical for ensuring appropriateness of the Cox model are:

- The censoring must be non-informative or statistically independent of the failure times
- The proportional hazards assumption must be met, meaning that the survival curves for two strata must have hazard functions that are proportional over time.

As mentioned previously, the most common type of censoring is right censoring, where the time-to-failure is not obtained for a subject due to the subject either making it to the end of the study period without the event occurring, or being prematurely removed from the study for reasons unrelated to the study itself. Such a possible reason is a physician's recommendation of a therapy or treatment that is not consistent with the group that the subject is in. If subjects discontinue participation in the study (and, hence, are censored) due to the study itself, such as an illness related to why they are in the study, then the censoring is informative and does not meet the assumptions of the Cox model.

In survival analysis, examining the relationship between the covariates and the survival distribution is of particular interest and can be done by specifying a model for the log hazard (Crumer, 2008). As an example, the hazard function for a parametric model based on an exponential distribution, as previously mentioned, can be written as,

$$log h_i(t) = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$$

which can be rewritten as,

$$h_i(t) = exp(\alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik})$$

where i denotes the observation, the x's denote the covariates, and α is a constant which represents the log-baseline hazard for the model (Fox, 2011). Here β is a vector parameter that is estimated using the partial likelihood,

$$L(\beta) = \prod_{i=1}^{D} \frac{exp[\beta x_i]}{\sum_{j \epsilon R(t_i)} exp[\beta x_j]}$$

where D is the total number of observed event times (Diez, 2013). For the Cox model, the unspecified baseline hazard function $\alpha(t) = logh_0(t)$, when all covariates are set to zero, and thus the log-hazard function is,

$$logh_i(t) = \alpha(t) + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$$

which yields a hazard function of the form

$$h_i(t) = h_0(t) exp(\beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}),$$

which leads to the survival function,

$$S_i^{cox}(t) = exp(-h_0(t) exp(\beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik})),$$

as given by (Mogensen, pg. 4). Since the form of the baseline hazard function is unrestricted and the covariates in the model enter linearly, the Cox model is said to be semi-parametric.

To show that the Cox model is a proportional hazards model, consider any pair of observations, i and j, for which the values of the covariates differ. Taking the linear predictors, for these two observations, $\eta_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$ and $\eta_j = \beta_1 x_{j1} + \beta_2 x_{j2} + \cdots + \beta_k x_{jk}$, the ratio of the two hazard functions is given by

$$\frac{h_i(t)}{h_j(t)} = \frac{h_0(t) exp[\eta_i]}{h_0(t) exp[\eta_j]} = \frac{exp[\eta_i]}{exp[\eta_j]} = \exp[\eta_i - \eta_j],$$

and is independent of time. Thus, the ratio of the hazard functions is strictly proportional to the (exponential of) differences in values of the predictor variables.

The Cox proportional hazards regression model has become one of the most frequently used method in survival analysis. It is simple to use, no harder than carrying out an ordinary regression, requires very little computational time even with large datasets, and has been extremely successful in identifying relationships between predictor variables and failure time in many and varied applications.

Random Survival Forests

Random Forests has mainly been used for classification and regression problems, but even in the original paper, Breiman (2001) noted that the methodology could be extended to censored lifetime data. An advantage of this approach is that it is fully non-parametric, including the effects of the treatments and predictor variables, whereas traditional methods either assume a distribution for the lifetimes or, in the case of the Cox Proportional hazards model, a linear-exponential form for the treatment effects. Interaction effects must be specified

individually in traditional methods and situations in which there are very large numbers of predictor variables, sometimes many more variables than observations, are problematic for traditional methods. In contrast, the survival form of random forests handles all of these aspects and problems automatically, just as the random forest algorithm does in regression and classification. This is largely due to random survival forests's use of decision trees as its *base learners* (a building block for an ensemble process). Ishwaran 2008 describes the basic random forests algorithm for survival analysis:

1. Draw B bootstrap samples from the original data. That is, draw B random samples of the same size as the original dataset from the original dataset, *with replacement*. On average, about 63% of the original observations will occur one or more times in the bootstrap sample and 37% of the original data will not occur in the bootstrap sample. These observations are said to be out-of-bag (OOB) with respect to the bootstrap sample.

2. Grow a survival tree on each of the $b = 1, \ldots, B$ bootstrap samples. At each node, p predictor variables will be randomly selected as candidates to be split on, and the predictor and splitting value which maximizes the difference in the objective function. By default, p is equal to the square root of the total number of predictor variables.

3. Continue to grow each tree to full size until no less than one unique event occurs. These final nodes are called the *terminal nodes* of the tree.

4. Calculate a cumulative hazard function (CHF) for each tree and average over all CHFs for the B trees to obtain the ensemble CHF.

5. Calculate the prediction error for the ensemble CHF using only the OOB data.

In RSF, survival trees are grown just as they would for classification and regression trees in random forests. The process begins at the root node, the top of the tree comprising of all data in the bootstrap sample. From the bootstrap sample, p predictor variables are chosen at random and used to split the root node, according to a predetermined survival criterion, into two daughter nodes. For each of the two daughter nodes, another set of p predictor variables are chosen at random and used to split each of the two nodes into two additional daughter nodes. This process is repeated recursively until each node has, at minimum, one unique event. This process groups together (in nodes) observations that are similar in the values of their predictor variables, and 'pushes apart' observations that are different.

Using the notation of Ishwaran and Kogalur (2008):

- h: The h^{th} node of a tree.
- n: The number of individuals within node h.
- T_l: The survival time for the l^{th} individual where $l \in \{1, \ldots, n\}$.
- δ_l: The censoring information for the l^{th} individual. $\delta_l = 0$ if the l^{th} individual is right censored and $\delta_l = 1$ if the l^{th} individual died.
- x: A candidate predictor for node splitting.

- c: A split value for predictor x.
- x^*: The predictor which maximizes survival differences between daughter nodes.
- c^*: The split value which maximizes survival differences between daughter nodes for predictor x.
- j: Daughter node, $j \in \{1, 2\}$
- t_N: The distinct event times in node h.
- $Y_{i,j}$: The individuals who are at risk (alive) or who had an event (death) at time t_i in daughter node j.
- $d_{i,j}$: The number of events at time t_i in daughter node j.

Suppose we want to find a proposed split for node h. Then for a given predictor x, we want to find a value c such that survival differences between the two inequalities, $x \leq c$ (the criterion for an individual to be placed in daughter node 1) and $x > c$ (the criterion for an individual to be placed in daughter node 2), are maximized. This is done by first choosing an x from your set of p randomly selected predictor variables, determining a split value, c, and assigning individuals to either the right or left daughter node based on the above inequalities, calculating the survival difference between the two groups using a predetermined splitting method, repeating the process with another c until you find the value for c which maximizes survival differences for x, and repeating the entire process for the other p-1 predictor variables until you find both x^* and c^* which maximize the survival difference between the two daughter nodes. In other words, the best split for node h is the one in which predictor x^* and split value c^* maximize the difference in survival between the two daughter nodes for all x and c.

Currently, random survival forests (RSF) has four distinct methods that can be used to maximize a splitting value c for a variable x. The first method, and the one that is used in the survival analysis below, is the log-rank splitting method which, as the name suggests, uses a multitude of log-rank tests to measure the severity of node separation at a value c for a predictor x. The value of the log rank test is given by the following formula:

$$L(x, c) = \frac{\sum_{i=1}^{N} (d_{i,1} - Y_{i,1} \frac{d_i}{Y_i})}{\sqrt{\sum_{i=1}^{N} \frac{Y_{i,1}}{Y_i}(1 - \frac{Y_{i,1}}{Y_i})(\frac{Y_i - d_i}{Y_i - 1})d_i}}.$$

The goal is to find the x and c which gives the largest magnitude of the log rank test. Briefly put, we want to find a predictor x^* and split value c^* such that $|L(x^*, c^*)| \geq |L(x, c)|$ for every x and c. This process is repeated at every node until the terminal node is reach.

The second method is an approximation to the previous one and is therefore named approximate logrank splitting. In order to approximate the numerator of $L(x, c)$, a revision is done using the Nelson-Aalen cumulative hazard estimator for the parent node.

The Nelson-Aalen Estimator:

$$\hat{H}(t) = \sum_{t_i \leq t} \frac{d_i}{Y_i}.$$

We rewrite the current numerator of $L(x, c)$ as:

$$\sum_{i=1}^{N} (d_{i,1} - Y_{i,1} \frac{d_i}{Y_i}) = D_j - \sum_{l=1}^{n} I\{x_l \leq c\} \hat{H}(T_l),$$

where $D_j = \sum_{i=1}^{N} d_{i,j}$ for $j = 1, 2$.

As suggested by LeBlanc and Crowley (1993), we can further simplify the denominator by approximating the variance of $L(x, c)$'s numerator. This is done by letting $D = \sum_{i=1}^{N} d_i$ which leads to the following approximation to the log-rank test:

$$\frac{D^{1/2}(D_j - \sum_{l=1}^{n} I\{x_l \leq c\} \hat{H}(T_l)}{\sqrt{\{\sum_{l=1}^{n} I\{x_l \leq c\} \hat{H}(T_l)\}\{D - \sum_{l=1}^{n} I\{x_l \leq c\} \hat{H}(T_l)\}}}.$$

The next method is that of log-rank score splitting. This computes the "ranks" for each survival time T_l given an ordered predictor x, namely $x_1 \geq x_2 \geq \ \ldots \ \geq x_n$, using the following formula:

$$a_l = \delta_l - \sum_{k=1}^{\Gamma_l} \frac{\delta_k}{n - \Gamma_k + 1},$$

where $\Gamma_k = \#\{t : T_t \leq T_k\}$. If we let $\bar{a}$ and s_a^2 be the sample mean and sample variance of a_l for $l \ \epsilon \ \{1, \ \ldots \ n\}$, the formula for the log-rank score test is given by:

$$S(x, c) = \frac{\sum_{x_l \leq c} a_l - n_1 \bar{a}}{\sqrt{n_1 (1 - \frac{n_1}{n}) s_a^2}},$$

where the "best" split is given by the x and c which maximize the survival difference between the two daughter nodes.

One proposed drawback of the log rank splitting family pertains to its favoring of uneven splits (end-cut preferences) when dealing with continuous predictor variables, and although we, as well as many others, have not found this to be an inadequacy, it is worth the mention of an alternative approach. This brings us to our fourth and final method, conservation of event splitting.

The main condition that the conservation of event method follows is that the estimated cumulative hazard function must equal the total number of events when summed over all observed event and censored times. This is done by using an altered version of the Nelson-Aalen estimator which is now computed for each daughter node rather than the parent node. This estimator is given by the following formula:

$$\hat{H}_j(t) = \sum_{t_{i,j} \leq t} \frac{d_{i,j}}{Y_{i,j}},$$

where the ordered event times for daughter j are given by $t_{i,j}$. The case where we have 0/0 is defined to be 0.

We can retain the total number of events for each daughter j by using:

$$\sum_{l=1}^{n_j} \hat{H}_j(T_{l,j}) = \sum_{l=1}^{n_j} \delta_{l,j}.$$

Now, for each daughter j, we want to order the event times such that $T_{(1),j} \leq \ \ldots \ \leq T_{(n_j),j}$.

In order to get a measure of the accuracy of the conservation of events, we define

$$R_{k,j} = \sum_{l=1}^{k} \hat{H}_j(T_{(l),j}) - \sum_{l=1}^{k} \delta_{(l),j} \text{ for } k = 1, \ \ldots, \ n_J.$$

This leads us to the following measure of conservation of events for the split on x at the value c:

$$Conserve(x, c) = \frac{1}{Y_{1,1}+Y_{1,2}} \sum_{j=1}^{2} Y_{1,j} \sum_{k=1}^{n_j-1} |R_{k,j}|$$

In other words, for each daughter j, the magnitudes of the $R_{k,j}$s are summed and weighted by the number of individuals at risk within each daughter node. Since the level of separation between the two daughter nodes increases as the test statistics decreases, in order for us to obtain the "best" split, we have to minimize this value or maximize the transformed value, $1/(1 + Conserve(x, c))$. This statistic can be very time-consuming to compute because it sums over all the survival times within each daughter node. Fortunately, the computation time can be severely decreased by using only the event times which can be expressed by the following formula:

$$Conserve(x, c) = \frac{1}{Y_{1,1}+Y_{1,2}} \sum_{j=1}^{2} Y_{1,j} \sum_{k=1}^{N-1} \{N_{k,j} T_{k+1,j} \sum_{l=1}^{k} \frac{d_{l,j}}{Y_{l,j}}\},$$

where $N_{i,j} = Y_{i,j} - Y_{i+1,j}$ is the amount of observations within daughter j with observed time falling within the interval $[t_i, t_{i+1})$ for $i = 1, \ldots, N$ where $t_{N+1} = \infty$ as stated by Ishwaran and Kogalur (2008). Note that the two formulas for Conserve(x, c) can be shown to be equivalent. Each of the four methods are repeated for the h^{th} node as long as each daughter node contains at least one unique events. When this criterion ceases to be met, the terminal node has been reached.

Once each of the B survival forests have been fully grown, the cumulative hazard function, an estimate that is to measure error rate assessment and, as a result, grants us the ability to compare different survival analysis methods, can be found. For each of the $b \in \{1, \ldots, B\}$ trees, a cumulative hazard function is determined by grouping cumulative hazard estimates across each of the M terminal nodes. We note that a cumulative hazard estimate can be found for all h nodes of a tree and not just the terminal nodes. The cumulative hazard estimate for node h is represented by the following formula:

$$\hat{H}_h(t) = \sum_{t_{l,h} \leq t} \frac{d_{l,h}}{Y_{l,h}}.$$

In other words, for a tree b, the hazard estimate for a node h is the proportion of events ($d_{l,h}$) to individuals at risk ($Y_{l,h}$) summed across all distinct event times, $t_{l,h} \leq t$. If the cumulative

hazard estimate for a specific individual i with predictor x_i is desired, the estimate is found by simply dropping x_i down the b^{th} tree until the individual reaches a terminal node. Once the terminal node has been reached, the cumulative hazard estimate for that terminal node becomes that individual's hazard estimate. Using the same individual, this process is repeated over the remainder of the $B-1$ trees and then averaged over all B trees. This provides the ensemble cumulative hazard estimator for the i^{th} individual. This is summarized using the following equation.

$$\hat{H}_e(t|x_i) = \tfrac{1}{B} \sum_{b=1}^{B} \hat{H}_b(t|x_i).$$

The corresponding ensemble survival function is thus,

$$\hat{S}^{rsf}(t|x_i) = exp(-\tfrac{1}{B} \sum_{b=1}^{B} \hat{H}_b(t|x_i)),$$

as shown by (Mogensen, pg. 5).

As high model performance and predictive accuracy are some, if not the, most important aspects to any data analysis, it is crucial that we test our method using data outside of our current sample. Recall that each of the B survival trees are grown using a bootstrap sample consisting of only 63% of the original data on average; therefore, there is approximately 37% of the original data that is left out, the OOB data. This makes sense as the predictive accuracies for the "in-bag" data would be high, if not perfect, since it was the data that was used to create the survival trees in the first place.

In order to construct the OOB ensemble cumulative hazard estimator for an individual i, only one additional piece is needed; an indicator variable stating whether the individual is OOB or not. To do this, let $I_{i,b} = 1$ if the individual is OOB and $I_{i,b} = 0$ if it is not. Then the OOB ensemble cumulative hazard estimator for an individual i is given by:

$$\hat{H}^*_e(t|x_i) = \frac{\sum_{b=1}^{B} I_{i,b} \ \hat{H}_b(t|x_i)}{\sum_{b=1}^{B} I_{i,b}}.$$

Conditional Inference Forests

Conditional inference forests, or *cforests*, is another fully non-parametric, tree based, method used in survival analysis and like random survival forests, it is based off of Breiman's random forests. However, cforests is unique in its choice of base learners and the aggregation scheme it applies. For each of it's B base learners, a conditional inference tree is constructed as follows (Mogensen, 2012):

1. For each predictor variable, test the null hypothesis that there is independence between the response variable(s) and the predictor variable. If we fail to reject the null, stop, otherwise choose the predictor variable which has the strongest association with the response. The association strength is assessed using the p-values from all partial null hypotheses of a single predictor and the response(s). A split only occurs when the p-value is smaller than a specified value.

2. Divide the observations of the selected predictor variable using a binary split. This splitting criterion is based on multiplicity adjusted p-values (Bonferroni or Monte Carlo), univariate p-values (Univariate), or on values of the test statistic. When the criterion, specified by the option *mincriterion*, is exceeded, a split is made. This method allows a tree to be grown to the correct size without the need for pruning or cross-validation.

3. Repeat the previous steps until a terminal node is reached (no additional predictor variables can be split).

The ensemble survival function is:

$$\hat{S}^{cforest}(t|x_i) = \prod_{t_{l,h} \leq t}(1 - \frac{\sum_{b=1}^{B} d_{l,h}}{\sum_{b=1}^{B} Y_{l,h}})$$

Prediction Error Curves

Methods such as Mallow's Cp and AIC/BIC are standard in the process of model selection. However, these methods may not be suitable in the survival analysis setting as they fail to account for censoring. Instead, the three models have been compared using two newer methods, namely prediction error curves and the *concordance index (c-index).*

Prediction error is accessed by an expected time dependent *Brier score.* For right censored data, the squared residual, (observed status$-$predicted status)^2, of a subject at each particular time point, t, is weighted using inverse probability of censoring weights. This censoring weight is given by

$$\hat{W}_i(t) = \frac{(1-\tilde{Y}_i(t))\Delta_i}{\hat{G}(\tilde{T}_i - |X_i)} + \frac{\tilde{Y}_i(t)}{\hat{G}(t|X_i)},$$

where $\tilde{Y}_i(t) = I(\tilde{T}_i > t)$ is the observed status of an individual i at time t and $\hat{G}(t|x) \approx P(C_i > t|X_i = x)$ is the estimate of the conditional survival function of the censoring times.

For a new observations, or if a test dataset, $\tilde{D}_M$, is available, the expected Brier score is estimate by

$$\hat{BS}(t, \hat{S}) = \frac{1}{M} \sum_{i \epsilon \tilde{D}_M} \hat{W}_i(t)(\tilde{Y}_i(t) - \hat{S}(t|X_i))^2,$$

where M is the number of subjects in $\tilde{D}_M$ and $\hat{S}$ is the predicted survival probability for a subject i at time t based on a training dataset. In order to protect against overfitting, ten-fold cross-validation iterated five times was used for each of the three survival methods on each of the three datasets.

Concordance Index

Unlike prediction error curves and other measures of survival performance, the c-index, is not dependent on a fixed time point for the evaluation of a model and takes into account the censor status of an individual. Two observations are said to be *concordant* if the observation that fails first is predicted to have a worse outcome. The process of obtaining the c-index is as follows:

1. Over the entire data, form all possible pairs of observations.

2. If, within a pair, the observation with the shorter survival time is censored or when both observations have the same survival time but at least one is not an event, omit the pair. The total number of remaining pairs is denoted as *permissible*.

3. Scoring:

    a. A permissible pair receives a value of 1 if:

        - Their survival times are not equal and the shorter survival time is predicted to have a worse outcome.
        - Their survival times are equal and their predicted outcomes are also equal.
        - Their survival times are equal, not both are events, and the predicted outcome is worse for the observation with the observed event.

    b. A permissible pair receives a value of 0.5 if:

        - Their survival times are not equal but their predicted outcomes are.
        - Their survival times are equal but their predicted outcomes are not.
        - Their survival times are equal, not both are events, and the predicted outcome is not worse for the observation with the observed event.

4. The c-index, C, is given by $C = \text{Concordance/Permissible}$.

5. The error rate is given by $Error = 1 - C$ where $0 \leq \text{Error} \leq 1$. $C = 1$ or $\text{Error} = 0$ indicates perfect prediction where as $C = \text{Error} = 0.5$ indicates doing no better than guessing.

Analysis

Initially we started with 5 datasets: AIDS Clinical Trials Group Study 320 Data (AIDS1), Australian AIDS Survival Data (AIDS2), French Three Cities Study Data (FTCS), North Central Cancer Treatment Group (NCCTG) Lung Cancer Data (LC), and More Stanford Heart Transplant Data (HT). All datasets contained both censored and uncensored observations where the censor variable was given a value of 1 if the event of interest occurred and 0 if it did not occur (during the study period). After plotting Kaplan-Meier curves to each of these datasets using a variety of grouping methods, we found that only 3 of the datasets had survival curves that were deemed interesting meaning not all curves follow the same shape. The datasets that were kept for further analysis were AIDS1, AIDS2, and LC.

Kaplan-Meier Curves

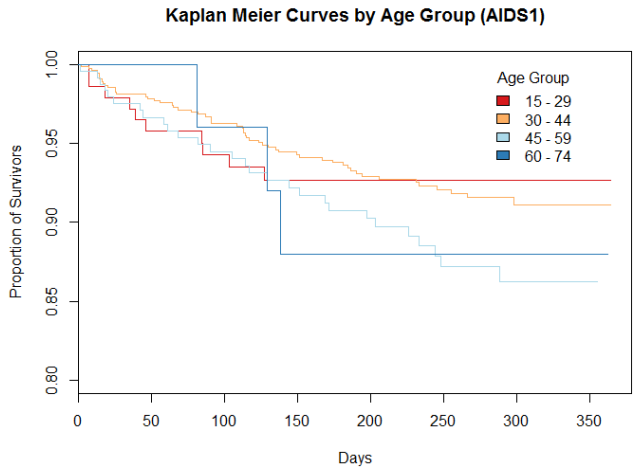

Figure 2: KM survivor curves for the AIDS1 data.

In this first set of Kaplan-Meier (KM) curves, we see that most are at least slightly concave up, but there is one that is clearly concave down. The same thing can be seen in our next KM plot.

Clearly the curves are different between male and female as the KM curve for males is concave up and the KM curve for females is concave down.

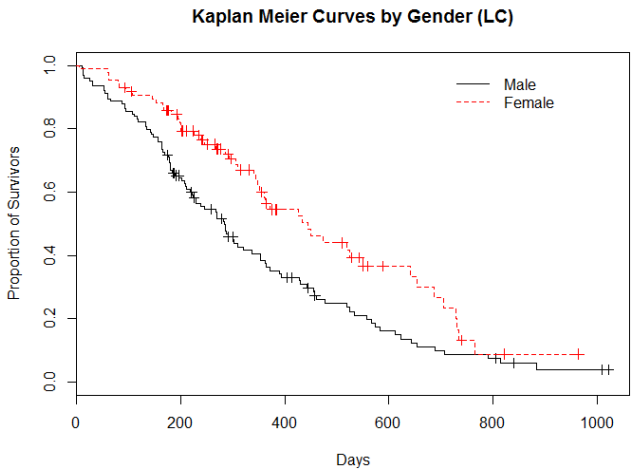

Figure 3: KM survivor curves for the LC data.

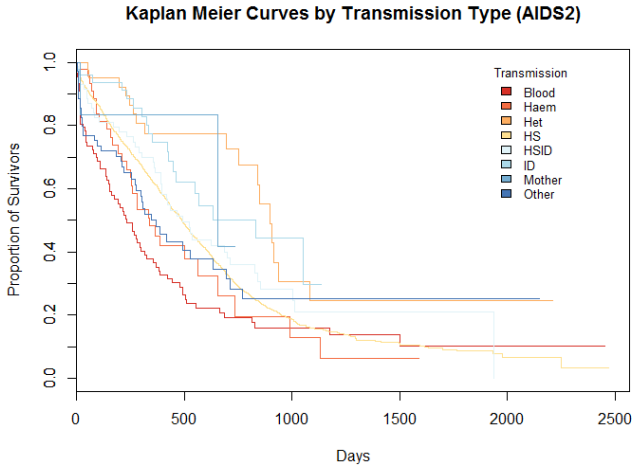

Figure 4: KM survivor curves for the AIDS2 data.

Again, as we saw in *Figure 1*, the majority of the curves are concave up, but there are one or two curves that are concave down. Thus we are interested in how accurately different models, such as Cox proportional hazard regression (CPH), random survival forests (RSF), and cforest (CF), fit curves to and predict survival probabilities for these types of data. In order to do this, we will use a couple different methods for comparing the three models and see which model is the better one to use.

Prediction Error Curve Comparisons

The first approach for comparing models is to see how each model's prediction error fluctuates as time goes on. For the three datasets of interest, I have plotted the prediction error curves for KM, CPH, RSF, and CF.

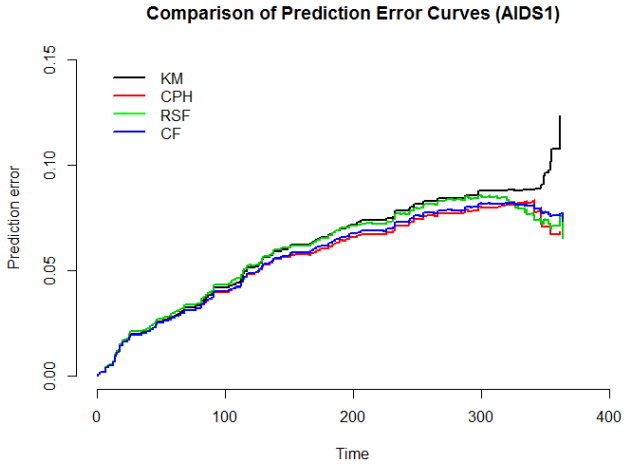

Figure 5: Comparison of prediction error curves for the AIDS1 data.

Notice that all four curves are practically the same for the AIDS1 data. The only real difference is that the KM error rate increases dramatically after about the 350 day mark, but the other three methods stay relatively the same.

In the prediction error plot for the LC data (*Figure 5*), we see that all methods approximately follow the same curve.

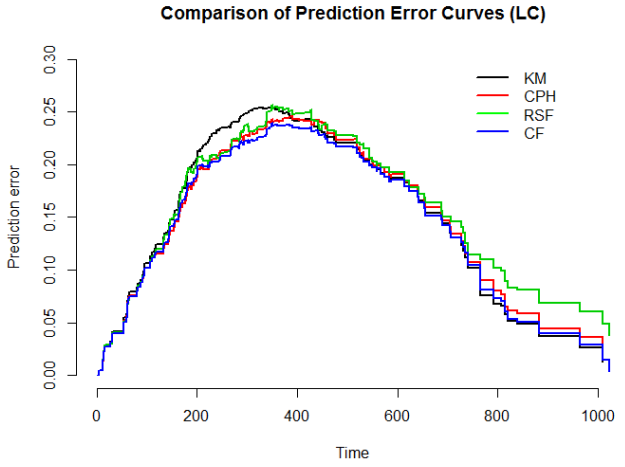

Figure 6: Comparison of prediction error curves for the LC data.

Figure 6 seems to be similar to *Figure 5* in that the prediction error curves for the different models are approximately the same over time. Random survival forests seems to consistently have a slightly higher prediction error than the other models.

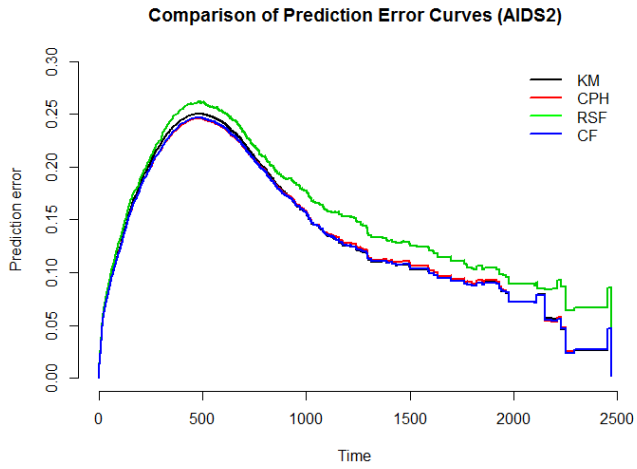

Figure 7: Comparison of prediction error curves for the AIDS2 data.

Comparison of Concordance

Our second method for comparing survival analysis models makes use of each model's concordance index (c-index)over time. This method has been deemed as one of the most commonly used methods for comparing survival models. The c-index gives the probability of concordance between the predicted and the observed survival. A c-index of 1 refers to the model making a prefect prediction and a c-index of 0.5 means the model did no better than guessing.

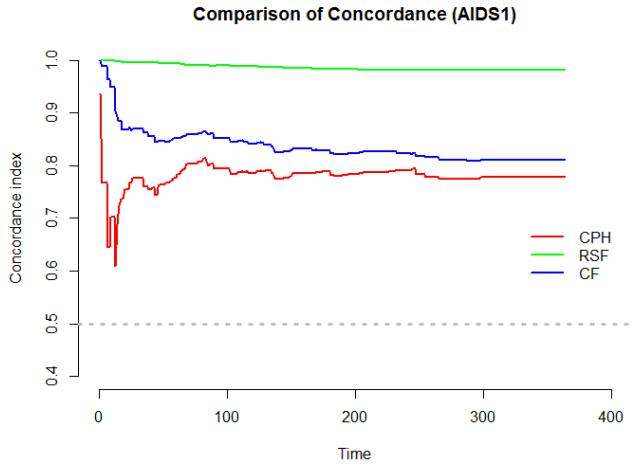

Figure 8: Comparison of c-indexes for the AIDS1 data.

In *figure 7* we see that random survival forests clearly out performs the other models in prediction performance of survival. We also we that conditional inference forests tend to be better than cox proportional hazards, but still far inferior to RSF. If also very interesting that CPH has a drastic dip at the beginning, but just as quickly levels off.

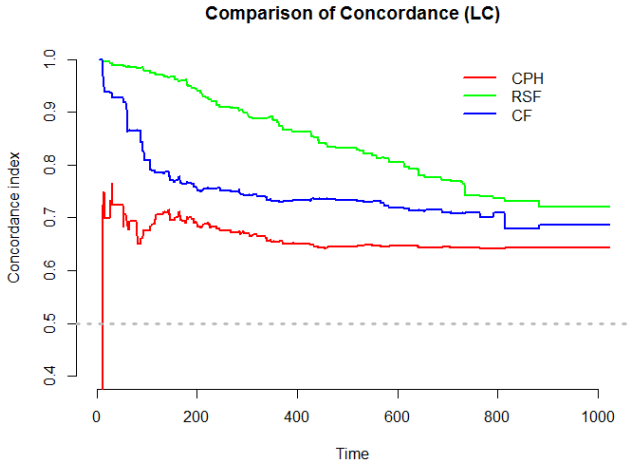

Figure 9: Comparison of c-indexes for the LC data.

As seen in *Figure 7*, *Figure 8* also shows RSF outperforming the other survival models. What's really interesting here is that RSF seems to be slightly concave down when CPH is concave up. I would say that this may be due to the model assumptions made by CPH that RSF does not, but CF's c-index follows more closely to the c-index of CPH over time and CF and RSF make the same model assumptions.

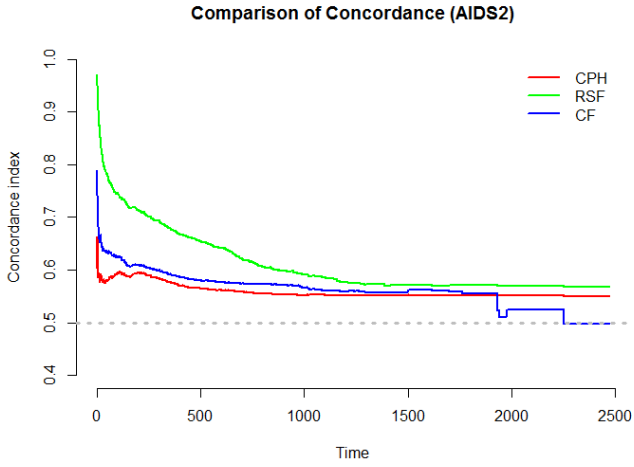

Figure 10: Comparison of c-indexes for the AIDS2 data.

Lastly *Figure 9* shows RSF doing better for approximately the first 1000 days, but then levels off to about the same c-index as CPH for the remainder of the time. CF follows the c-index of CPH very closely until about day 1900 then starts to diminish for the remainder of the time. It actually gets so low that the model is basically predicting at random.

Overall, random survival forest is clearly the most accurate method for predicting survival over time. Conditional inference forests generally does better than cox proportional hazards,

but still nowhere as good as random survival forests.

	Dataset	Process	Method	Time
1	AIDS1	CPH	predErr	0.032390634
2	AIDS1	RSF	predErr	1.700699886
3	AIDS1	CF	predErr	0.368538300
4	AIDS1	CPH	c-index	0.002769252
5	AIDS1	RSF	c-index	0.007998268
6	AIDS1	CF	c-index	0.033632048
7	LC	CPH	predErr	0.042314514
8	LC	RSF	predErr	3.924985035
9	LC	CF	predErr	3.668048267
10	LC	CPH	c-index	0.006904936
11	LC	RSF	c-index	0.016653951
12	LC	CF	c-index	0.424662968
13	AIDS2	CPH	predErr	0.168633167
14	AIDS2	RSF	predErr	192.149433966
15	AIDS2	CF	predErr	26.173246984
16	AIDS2	CPH	c-index	0.639157915
17	AIDS2	RSF	c-index	1.482246864
18	AIDS2	CF	c-index	3.390277600
19			Total	234.232594554

Figure 11: A table of run times (in minutes) for each comparison method.

Notice that random survival forests consistently runs longer than the other methods, but only when calculating its prediction error curve for the AIDS2 data is it very recognizable. In the future, we may want to figure out why this is. I thought it may be due to the number of observations since the AIDS2 data has 2,814 observations, which is a little over twice that of the AIDS1 data set (1,151 observations), and way more than the LC dataset (209 observations), but the time difference isn't roughly 2 times greater, it's 2^6 or 64 times greater. The AIDS2 data is also contains less variables than the other two datasets, so I'm really not sure where the difference in computational time is coming from. Therefore, if the results were to stand for all datasets 2000 or more, I would suggest the use of CF as it provides a more accurate prediction for survival times than CPH, but doesn't take that long to run.

References

Allison, Paul D. "31 Survival Analysis." *The Reviewer's Guide to Quantitative Methods in the Social Sciences*. Ed. Gregory R. Hancock and Ralph O. Mueller. New York: Routledge, 2010. 413+. Print.

Crumer, A. (2008). *Comparison Between Weibull and Cox Proportional Hazards Models*

Despa, S. (n.d.). What is Survival Analysis? *StatNews*, 78, 1.

Diez, D. M. (2013). *Survival Analysis in R*. OpenIntro.

Fan, Jianqing (2009). *New Developments in Biostatistics and Bioinformatics*. Eds. Xihong Lin and Jun S. Liu. Volume 1 ed. World Scientific Publishing Co Pte Ltd: Frontiers of Statistics. Print.

Fox, Weisberg (2011). Cox Proportional-Hazards Regression for Survival Data in R. *An Appendix to An R Companion to Applied Regression*, Second Edition.

Gerds TA (2017). **pec:** *Prediction Error Curves for Risk Prediction Models in Survival Analysis*. R package version 2.5.3, URL https://CRAN.R-project.org/package=pec.

Gerds TA, Kattan MW, Schumacher M, and Yu C (2013). Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. *Statistics in Medicine*, Ahead of print:to appear. DOI = 10.1002/sim.5681

Hothorn Torsten, Hornik Kurt, Strobl Carolin, & Zeileis Achim (2017). **party:** *A Laboratory for Recursive Partytioning*. R package version 1.2-2, URL https://cran.r-project.org/package=party.

Hothorn Torsten, Peter Buehlmann, Sandrine Dudoit, Annette Molinaro and Mark Van Der Laan (2006). Survival Ensembles. Biostatistics, *7(3)*, 355–373.

Ishwaran H., Kogalur U. B. (2017). **randomForestSRC:** *Random Forests for Survival, Regression and Classification (RF-SRC)*. R package version 2.4.2, URL https://cran.r-project.org/package=randomForestSRC.

Ishwaran H., Kogalur U. B., Blackstone E. H., & Lauer M. S. (2008). Random Survival Forests. *The Annal of Applied Statistics*, *2(3)*, doi: 10.1214/08-AOAS169

Kishore J., Goel M., & Khanna P. (2010). Understanding survival analysis: Kaplan-Meier estimate. *International Journal of Ayurveda Research*, *1(4)*, 212-216. doi: 10.4103/0974-7788.76794

Mogensen U., Ishwaran H., & Gerds T. (2012). Evaluating Random Forests for Survival Analysis Using Prediction Error Curves. *Journal of Statistical Software*, *50(11)*, 1-20. Retrieved from https://www.jstatsoft.org/article/view/v050i11.

LeBlanc, M. and Crowley, J. (1993). Survival trees by goodness of split. *J. Amer. Statist. Assoc.* 88 457-467. MR1224370

Neuwirth Erich (2014). **RColorBrewer:** *ColorBrewer Palettes*. R package version 1.1-2, URL https://CRAN.R-project.org/package=RColorBrewer.

Therneau Terry, Lumley Thomas (2016). **survival:** *Survival Analysis*. R package version 2.40-1, URL https://cran.r-project.org/package=survival.

VanDerWal Jeremy, Falconi Lorena, Januchowski Stephanie, Shoo Luke, and Storlie Collin (2014). **SDMTools:** *Species Distribution Modelling Tools: Tools for Processing data associated with species distribution modelling exercises*. R package version 1.1-221, URL https://cran.r-project.org/package=SDMTools

R-code for Analysis

NCCTG Lung Cancer Data

```
library(randomForestSRC)
library(survival)
library(pec)

setwd("~/Graduate Project/Data/Lung Cancer")

#Load and store the data
lcOrig <- read.csv("cancer.csv")

#Replace all the 1's with 0's (censored)
lcOrig$status <- gsub(pattern = "1", replacement = "0", x =
                        lcOrig$status, fixed = TRUE)

#Replace all the 2's with 1's (death)
lcOrig$status <- gsub (pattern = "2", replacement = "1", x =
                         lcOrig$status, fixed = TRUE)

#Do the same thing for sex (0 = Males, 1 = Females)
lcOrig$sex <- gsub(pattern = "1", replacement = "0", x =
                     lcOrig$sex, fixed = TRUE)

lcOrig$sex <- gsub(pattern = "2", replacement = "1", x =
                     lcOrig$sex, fixed = TRUE)

#Change the class of these variables to factor.
lcOrig$status <- as.integer(lcOrig$status)
lcOrig$sex <- as.integer(lcOrig$sex)
lcOrig$ph.ecog <- as.integer(lcOrig$ph.ecog)

#Remove missing values and column with over 20% missing data.
lcOrig <- lcOrig[, c(1:9, 11)]
lc <- lcOrig[complete.cases(lcOrig), ]

#Matrix of run times.
runTime <- matrix(data = NA, nrow = 19, ncol = 4)
colnames(runTime) <- c("Dataset", "Process", "Method", "Time")
runTime[, 1] <- c(rep(c("AIDS1", "LC", "AIDS2"), c(6, 6, 6)), " ")
runTime[, 2] <- c(rep(c("CPH", "RSF", "CF"), 6), " ")
runTime[, 3] <- c(rep(c("predErr", "c-index"), c(3, 3)),
                  rep(c("predErr", "c-index"), c(3, 3)),
```

```
                    rep(c("predErr", "c-index"), c(3, 3)), "Total")
```

```
############################
###KAPLAN-MEIER BY GROUP###
############################

library(RColorBrewer)
library(SDMTools)

#Kaplan-Meier Survival Curves for Patient Survival based on Gender.
lc$survObj <- with(lc, Surv(time, event = status))
km.by.sex <- survfit(survObj ~ sex, data = lc, conf.type = "log-log")
plot(km.by.sex, lty = 1:2, col = c("Black", "Red"), xlab = "Days", ylab =
       "Proportion of Survivors", main =
       "Kaplan Meier Curves by Gender", mark.time = TRUE)
legend(x = 500, y = 1, legend = c("Male", "Female"), lty = 1:2, col =
         c("Black", "Red"), title = "Gender", cex = .95, bty = "n")

#Determine age groups for individuals.
for(i in 1:nrow(lc))
{
  if(lc$age[i] < 50)
    lc$ageGroup[i] <- 1
  else if(lc$age[i] < 60)
    lc$ageGroup[i] <- 2
  else if(lc$age[i] < 70)
    lc$ageGroup[i] <- 3
  else if(lc$age[i] >= 70)
    lc$ageGroup[i] <- 4
}

#Kaplan Meier Curves based on age groups.
lc$survObj <- with(lc, Surv(time, censor == 1))
km.by.age <- survfit(survObj ~ ageGroup, data = lc, conf.type = "log-log")
plot(km.by.age, lty = 1, col = brewer.pal(4, "RdYlBu"), xlab =
       "Days", ylab = "Proportion of Survivors", main =
       "Kaplan Meier Curves by Age Group", ylim = c(0, 1.0))
legend(x = 500, y = 1, legend =
         c("39 - 49", "50 - 59", "60 - 69", "70+"), fill =
         brewer.pal(4, "RdYlBu"), title = "Age Group", bty = "n")
```

```
#############
###Weibull###
#############
fitform <- Surv(time, status) ~ inst + age + sex + ph.ecog + ph.karno +
  pat.karno + wt.loss
wie_lc <- flexsurvreg(formula = fitform, data = lc, dist = "weibull")
wie_lc
plot(wie_lc, xlab = "Days", ylab = "Proportion of Survivors",
     main = "Kaplan-Meier vs. Weibull for the LC Data")
legend(x = 700, y = 1, legend = c("KM", "Weibull"), lty = 1:2,
       col = c("Black", "Red"), cex = 1, bty = "n")

##############################
###Exponential Distribution###
##############################

exp_lc <- flexsurvreg(formula = fitform, data = lc, dist = "exp")
exp_lc
plot(exp_lc, xlab = "Days", ylab = "Proportion of Survivors")

###########
###GAMMA###
###########

gam_lc <- flexsurvreg(formula = fitform, data = lc, dist = "gamma")
gam_lc
plot(gam_lc, xlab = "Days", ylab = "Proportion of Survivors")

################
###LOG NORMAL###
################

lnorm_lc <- flexsurvreg(formula = fitform, data = lc, dist = "lnorm")

##############################################
###PLOT OF PARAMETRIC MODELS VS KAPLAN-MEIER###
##############################################

km.lc <- survfit(survObj ~ 1, data = lc, conf.type = "log-log")
plot(km.lc, conf.int = F, col = "black",
     main = "KM vs. Parametric Distributions", xlab = "Time (Days)",
     ylab = "Proportion of Survivors")
lines(wie_lc, col = "#238b45", ci = F)
lines(exp_lc, col = "#66c2a4", ci = F)
```

```r
lines(gam_lc, col = "blue", ci = F)
lines(lnorm_lc, col = "#fd8d3c", ci = F)
legend(x = 700, y = 1, legend = c("Kaplan-Meier", "Weibull", "Exponential",
                                  "Gamma", "Log-Normal"), lty = 1,
       col = c("Red", "#238b45", "#66c2a4", "blue", "#fd8d3c"), cex = 1,
       bty = "n")

AICs <- matrix(data = NA, nrow = 4, ncol = 1)
AICs[1, 1] <- wie_lc$AIC
AICs[2, 1] <- exp_lc$AIC
AICs[3, 1] <- gam_lc$AIC
AICs[4, 1] <- lnorm_lc$AIC
rownames(AICs) <- c("Weibull", "Exponential", "Gamma", "Log-Normal")
colnames(AICs) <- "AIC"
AICs
#Since weibull is the smallest, it fits the model best.
```

```r
############################################################
###PLOT OF PARAMETRIC MODELS VS KAPLAN-MEIER AND COX###
############################################################


fitform1 <- Surv(time, status) ~ inst + age + sex + ph.ecog + ph.karno +
  pat.karno + wt.loss

cox1 <- coxph(fitform1, data = lc)

plot(km.lc, conf.int = F, col = "black", main = "Comparison of Model Fits",
     xlab = "Time (Days)", ylab = "Proportion of Survivors")
lines(wie_lc, col = "#238b45", ci = F)
lines(exp_lc, col = "#66c2a4", ci = F)
lines(gam_lc, col = "blue", ci = F)
lines(lnorm_lc, col = "#fd8d3c", ci = F)
lines(survfit(cox1, conf.int = F), col = "red")
legend(x = 700, y = 1, legend = c("Kaplan-Meier", "Weibull", "Exponential",
                                  "Gamma", "Log-Normal", "Cox-PH"), lty = 1,
       col = c("black", "#238b45", "#66c2a4", "blue", "#fd8d3c", "red"),
       cex = 1, bty = "n")
```

```
##############################
###RANDOM SURVIVAL FORESTS###
##############################

set.seed(0692)
rsf1 <- rfsrc(fitform1, data = lc, forest = TRUE, ntree = 1000,
              splitrule = "logrank", importance = TRUE)
plot.survival.rfsrc(rsf1, plots.one.page = FALSE,
                    cens.model = "rfsrc")

##############
###C-FOREST###
##############

pecCforest <- function(formula, data, ...) {
  require("party")
  out <- list(forest = cforest(formula, data, ...))
  class(out) <- "pecCforest"
  out$call <- match.call()
  out
}

set.seed(0692)
cf1 <- pecCforest(fitform1, data = lc,
                  controls = cforest_classical(ntree = 1000))

##############################
###PREDICTION ERROR CURVES###
##############################

extends <- function(...) TRUE
library("doMC")
registerDoMC()

set.seed(0692)
fitpec1 <- pec(list("CPH" = cox1, "RSF" = rsf1, "CF" = cf1), data = lc,
               formula = fitform1, splitMethod = "cv10", B = 5,
               keep.index = TRUE, keep.matrix = TRUE)

#Reference is the marginal KM model (Only fitting y-intercept).
plot(fitpec1, what = "crossvalErr", xlim = c(0, 1022), legend = F)
legend(x = 725, y = 0.30, legend = c("KM", "CPH", "RSF", "CF"), lty = 1,
       col = c("black", "red", "green", "blue"), bty = "n", cex = 1, lwd = 2)
title("Comparison of Prediction Error Curves (LC)", line = 1.5, cex = 5)
```

```r
set.seed(0692)
startTime <- Sys.time()
pec(list("CPH" = cox1), data = lc, formula = fitform1, splitMethod = "cv10",
    B = 5, keep.index = TRUE, keep.matrix = TRUE, reference = F)
endTime <- Sys.time()
(runTime[1, 4] <- (endTime - startTime)/60)

set.seed(0692)
startTime <- Sys.time()
pec(list("RSF" = rsf1), data = lc, formula = fitform1, splitMethod = "cv10",
    B = 5, keep.index = TRUE, keep.matrix = TRUE, reference = F)
endTime <- Sys.time()
(runTime[2, 4] <- endTime - startTime)

set.seed(0692)
startTime <- Sys.time()
pec(list("CF" = cf1), data = lc, formula = fitform1, splitMethod = "cv10",
    B = 5, keep.index = TRUE, keep.matrix = TRUE, reference = F)
endTime <- Sys.time()
(runTime[3, 4] <- (endTime - startTime)/60)

##############
###C-INDEX###
##############
startTime <- Sys.time()
set.seed(0692)
ApparrentCindex1 <- cindex(list("Cox" = cox1, "RSF" = rsf1, "CF" = cf1),
                           formula = fitform1, data = lc,
                           eval.times = seq(1, 1022, 1))
endTime <- Sys.time()
(totalRunTime <- endTime - startTime)

plot(ApparrentCindex1, legend = F, col = c("red", "green", "blue"))
legend(x = 700, y = 1, legend = c("CPH", "RSF", "CF"), lty = 1,
       col = c("red", "green", "blue"), bty = "n", cex = 1, lwd = 2)
title("Comparison of Concordance (LC)", line = 1.5, cex = 5)

set.seed(0692)
startTime <- Sys.time()
cindex(list("Cox" = cox1), formula = fitform1, data = lc,
       eval.times = seq(1, 1022, 1))
endTime <- Sys.time()
(runTime[4, 4] <- (endTime - startTime)/60)
```

```r
set.seed(0692)
startTime <- Sys.time()
cindex(list("RSF" = rsf1), formula = fitform1, data = lc,
       eval.times = seq(1, 1022, 1))
endTime <- Sys.time()
(runTime[5, 4] <- (endTime - startTime)/60)

set.seed(0692)
startTime <- Sys.time()
cindex(list("CF" = cf1), formula = fitform1, data = lc,
       eval.times = seq(1, 1022, 1))
endTime <- Sys.time()
(runTime[6, 4] <- (endTime - startTime)/60)
```

AIDS Clinical Trails Group Study 320 Data

```r
##########################################
###AIDS CLINICAL TRAILS GROUP STUDY 320###
##########################################

setwd("~/Graduate_Project/Data/Aids")
aids <- read.csv("aids.csv")
head(aids)

##################
###Kaplan-Meier###
##################

for(i in 1:nrow(aids))
{
  if(aids$age[i] < 30)
    aids$ageGroup[i] <- 1
  else if(aids$age[i] < 45)
    aids$ageGroup[i] <- 2
  else if(aids$age[i] < 60)
    aids$ageGroup[i] <- 3
  else if(aids$age[i] < 75)
    aids$ageGroup[i] <- 4
}

#Kaplan Meier Curves based on age groups.
aids$survObj <- with(aids, Surv(time, censor == 1))
km.by.age.aids <- survfit(survObj ~ ageGroup, data = aids,
                          conf.type = "log-log")
plot(km.by.age.aids, lty = 1, col = brewer.pal(4, "RdYlBu"), xlab = "Days",
```

```
     ylab = "Proportion of Survivors",
     main = "Kaplan Meier Curves by Age Group (AIDS1)", ylim = c(0.8, 1.0))
legend(x = 275, y = 1,
       legend = c("15 - 29", "30 - 44", "45 - 59", "60 - 74"),
       fill = brewer.pal(4, "RdYlBu"), title = "Age Group", bty = "n")

#################################################################
###SELECTED COX REGRESSION USING STEPWISE VARIABLE SELECTION###
#################################################################

fitform2 <- Surv(time, censor) ~ tx + txgrp + strat2 + sex + raceth
+ ivdrug + hemophil + karnof + cd4 + priorzdv + age

cox2 <- coxph(fitform2, data = aids)

fitcox2 <- selectCox(fitform2, data = aids, rule = "aic")

###################################
###RANDOM SURVIVAL FOREST MODEL###
###################################

set.seed(0692)
rsf2 <- rfsrc(fitform2, data = aids, forest = TRUE, ntree = 1000)

#################################################
###CONDITIONAL INFERENCE FOREST MODEL (cforest)###
#################################################

pecCforest <- function(formula, data, ...) {
  require("party")
  out <- list(forest = cforest(formula, data, ...))
  class(out) <- "pecCforest"
  out$call <- match.call()
  out
}

set.seed(0692)
cf2 <- pecCforest(fitform2, data = aids,
                  controls = cforest_classical(ntree = 1000))
```

```
##############################
###PREDICTION ERROR CURVES###
##############################

#Uses 10-Fold Cross Validation repeated 10 times and averaged over
#10 error rates.
set.seed(0692)
fitpec2 <- pec(list("CPH" = cox2, "RSF" = rsf2, "CF" = cf2), data = aids,
               formula = fitform2, splitMethod = "cv10", B = 5,
               keep.index = TRUE, keep.matrix = TRUE)

plot(fitpec2, what = "crossvalErr", xlim = c(0, 400), legend = F,
     ylim = c(0, 0.15))
legend(x = 0, y = 0.15, legend = c("KM", "CPH", "RSF", "CF"), lty = 1,
       col = c("black", "red", "green", "blue"), bty = "n", cex = 1, lwd = 2)
title("Comparison of Prediction Error Curves (AIDS1)", line = 1.5, cex = 5)

set.seed(0692)
startTime <- Sys.time()
fitpec2 <- pec(list("CPH" = cox2), data = aids, formula = fitform2,
               splitMethod = "cv10", B = 5, keep.index = TRUE,
               keep.matrix = TRUE, reference = F)
endTime <- Sys.time()
(runTime[7, 4] <- (endTime - startTime)/60)

set.seed(0692)
startTime <- Sys.time()
fitpec2 <- pec(list("RSF" = rsf2), data = aids, formula = fitform2,
               splitMethod = "cv10", B = 5, keep.index = TRUE,
               keep.matrix = TRUE, reference = F)
endTime <- Sys.time()
(runTime[8, 4] <- endTime - startTime)

set.seed(0692)
startTime <- Sys.time()
fitpec2 <- pec(list("CF" = cf2), data = aids, formula = fitform2,
               splitMethod = "cv10", B = 5, keep.index = TRUE,
               keep.matrix = TRUE, reference = F)
endTime <- Sys.time()
(runTime[9, 4] <- endTime - startTime)
```

```
#############
###C-INDEX###
#############

set.seed(0692)
ApparrentCindex2 <- cindex(list("Cox" = cox2, "RSF" = rsf2, "CF" = cf2),
                           formula = fitform2, data = aids,
                           eval.times = seq(1, 364, 1))

plot(ApparrentCindex2, legend = F, col = c("red", "green", "blue"),
     xlim = c(0, 400))
legend(x = 325, y = .7, legend = c("CPH", "RSF", "CF"), lty = 1,
       col = c("red", "green", "blue"), bty = "n", cex = 1, lwd = 2)
title("Comparison of Concordance (AIDS1)", line = 1.5, cex = 5)

set.seed(0692)
startTime <- Sys.time()
cindex(list("Cox" = cox2), formula = fitform2, data = aids,
       eval.times = seq(1, 364, 1))
endTime <- Sys.time()
(runTime[10, 4] <- (endTime - startTime)/60)

set.seed(0692)
startTime <- Sys.time()
cindex(list("RSF" = rsf2), formula = fitform2, data = aids,
       eval.times = seq(1, 364, 1))
endTime <- Sys.time()
(runTime[11, 4] <- (endTime - startTime)/60)

set.seed(0692)
startTime <- Sys.time()
cindex(list("CF" = cf2), formula = fitform2, data = aids,
       eval.times = seq(1, 364, 1))
endTime <- Sys.time()
(runTime[12, 4] <- (endTime - startTime)/60)
```

Australian AIDS Survival Data

```
###################################
###Australian AIDS Survival Data###
###################################

#Set work directory.
setwd("~/Graduate_Project/Data/Aids2")

#Read in the data and look at the first 6 observations.
aids2 <- read.csv("aids2.csv")
head(aids2)


#Create a new "time" column that shows how long they either lived after
#diagnosis or survived to the end of the observation period.
aids2[, 9] <- aids2$death - aids2$diag
colnames(aids2)[9] <- "time"

#Rewrite A as 0 and D as 1 in status column.
#0 corresponding to alive, 1 corresponding to dead.
aids2$status <- gsub("A", 0, aids2$status)
aids2$status <- gsub("D", 1, aids2$status)

#Convert entries from class character to class numeric.
aids2$status <- as.numeric(aids2$status)

#Change the name of the column T.categ
colnames(aids2)[7] <- "trans"

#Keep observations where time > 0.
aids2 <- aids2[which(aids2$time > 0), ]

##################
###Kaplan-Meier###
##################

aids2$survObj <- with(aids2, Surv(time, status == 1))
km.by.sex.aids2 <- survfit(survObj ~ trans, data = aids2,
                           conf.type = "log-log")
plot(km.by.sex.aids2, lty = 1, col = brewer.pal(8, "RdYlBu"), xlab = "Days",
     ylab = "Proportion of Survivors",
     main = "Kaplan Meier Curves by Transmission Type (AIDS2)")
axis(2, at = seq(0, 1, 0.1))
legend(x = 1900, y = 1,
```

```
       legend = c("Blood", "Haem", "Het", "HS", "HSID", "ID", "Mother",
                  "Other"), fill = brewer.pal(8, "RdYlBu"),
       title = "Transmission", cex = .8, bty = "n")
```

```
################################################################
###SELECTED COX REGRESSION USING STEPWISE VARIABLE SELECTION###
################################################################

fitform3 <- Surv(time = time, event = status) ~ state + sex + trans + age

cox3 <- coxph(fitform3, data = aids2)

sum.surv3 <- summary(coxph(fitform3, data = aids2))

c_index3 <- sum.surv3$concordance

fitcox3 <- selectCox(fitform3, data = aids2, rule = "aic")

###################################
###RANDOM SURVIVAL FOREST MODEL###
###################################

set.seed(0692)
rsf3 <- rfsrc(fitform3, data = aids2, forest = TRUE, ntree = 1000)

#####################################################
###CONDITIONAL INFERENCE FOREST MODEL (cforest)###
#####################################################

set.seed(0692)
cf3 <- pecCforest(fitform3, data = aids2,
                  controls = cforest_classical(ntree = 1000))

#############################
###PREDICTION ERROR CURVES###
#############################
startTime <- Sys.time()
set.seed(0692)
fitpec3 <- pec(list("CPH" = fitcox3, "RSF" = rsf3, "CF" = cf3), data = aids2,
               formula = fitform3, splitMethod = "cv10", B = 1,
               keep.index = TRUE, keep.matrix = TRUE)
endTime <- Sys.time()
(totalRunTime <- endTime - startTime)
#Runtime 49.27563 minutes
```

```
plot(fitpec3, what = "crossvalErr", xlim = c(0, 2470), legend = F)
legend(x = 2000, y = 0.30, legend = c("KM", "CPH", "RSF", "CF"), lty = 1,
       col = c("black", "red", "green", "blue"), bty = "n", cex = 1, lwd = 2)
title("Comparison of Prediction Error Curves (AIDS2)", line = 1.5, cex = 5)

set.seed(0692)
startTime <- Sys.time()
pec(list("CPH" = fitcox3), data = aids2, formula = fitform3,
    splitMethod = "cv10", B = 5, keep.index = TRUE, keep.matrix = TRUE,
    reference = F)
endTime <- Sys.time()
(runTime[13, 4] <- (endTime - startTime)/60)

set.seed(0692)
startTime <- Sys.time()
pec(list("RSF" = rsf3), data = aids2, formula = fitform3,
    splitMethod = "cv10", B = 5, keep.index = TRUE, keep.matrix = TRUE,
    reference = F)
endTime <- Sys.time()
(runTime$Time[14] <- (endTime - startTime)*60)

set.seed(0692)
startTime <- Sys.time()
pec(list("CF" = cf3), data = aids2, formula = fitform3,
    splitMethod = "cv10", B = 1, keep.index = TRUE, keep.matrix = TRUE,
    reference = F)
endTime <- Sys.time()
(runTime[15, 4] <- endTime - startTime)

##############
###C-INDEX###
##############
startTime <- Sys.time()
ApparrentCindex3 <- cindex(list("Cox" = cox3, "RSF" = rsf3, "CF" = cf3),
                           formula = fitform3, data = aids2,
                           eval.times = seq(1, 2470, 1))
(totalRunTime <- endTime - startTime)

plot(ApparrentCindex3, legend = F, col = c("red", "green", "blue"))
legend(x = 2000, y = 1, legend = c("CPH", "RSF", "CF"), lty = 1,
       col = c("red", "green", "blue"), bty = "n", cex = 1, lwd = 2)
title("Comparison of Concordance (AIDS2)", line = 1.5, cex = 5)

set.seed(0692)
```

```
startTime <- Sys.time()
cindex(list("Cox" = cox3), formula = fitform3, data = aids2,
       eval.times = seq(1, 2470, 1))
endTime <- Sys.time()
(runTime[16, 4] <- (endTime - startTime)/60)

set.seed(0692)
startTime <- Sys.time()
cindex(list("RSF" = rsf3), formula = fitform3, data = aids2,
       eval.times = seq(1, 2470, 1))
endTime <- Sys.time()
(runTime[17, 4] <- endTime - startTime)

set.seed(0692)
startTime <- Sys.time()
cindex(list("CF" = cf3), formula = fitform3, data = aids2,
       eval.times = seq(1, 2470, 1))
endTime <- Sys.time()
(runTime[18, 4] <- endTime - startTime)

########################
###Table of Run Times###
########################

runTime <- temp
runTime <- as.data.frame(runTime)
options(scipen = 999)
runTime$Time <- as.numeric(as.character(runTime$Time))

runTime$Time[19] <- sum(runTime$Time[1:18])

runTime
```