Survival Analysis

Liam Murphy

2024-11-23

```r
library(survival)
library(ranger)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggfortify)
library(broom)
library(gtsummary)
data(cancer, package="survival")
myeloid <- na.omit(myeloid)
```

```r
# Create kaplan meier curve variable
km <- with(myeloid, Surv(futime, death ))
```

```r
# Produce kaplan meier survival estimates of the probability of survival over time
km_fit <- survfit(Surv(futime, death) ~ 1, data = myeloid )

summary(km_fit, death = c(1,30,60,90*(1:10)))
```

```
## Call: survfit(formula = Surv(futime, death) ~ 1, data = myeloid)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   140    136       1    0.993 0.00733        0.978        1.000
##   181    135       1    0.985 0.01032        0.965        1.000
##   191    134       2    0.971 0.01449        0.943        0.999
##   237    132       1    0.963 0.01614        0.932        0.995
##   245    131       1    0.956 0.01761        0.922        0.991
##   255    130       1    0.949 0.01895        0.912        0.986
```

```
##   258    129       2    0.934 0.02132         0.893         0.977
##   261    127       1    0.926 0.02238         0.884         0.971
##   275    126       2    0.912 0.02432         0.865         0.961
##   278    124       1    0.904 0.02521         0.856         0.955
##   284    123       1    0.897 0.02606         0.847         0.950
##   286    122       1    0.890 0.02686         0.839         0.944
##   293    121       1    0.882 0.02763         0.830         0.938
##   295    120       1    0.875 0.02836         0.821         0.932
##   307    119       1    0.868 0.02906         0.813         0.927
##   318    118       1    0.860 0.02973         0.804         0.921
##   322    117       1    0.853 0.03037         0.795         0.915
##   326    116       2    0.838 0.03158         0.779         0.902
##   332    114       1    0.831 0.03214         0.770         0.896
##   337    113       2    0.816 0.03321         0.754         0.884
##   344    111       1    0.809 0.03372         0.745         0.878
##   355    110       1    0.801 0.03420         0.737         0.871
##   365    109       1    0.794 0.03467         0.729         0.865
##   372    108       1    0.787 0.03512         0.721         0.859
##   388    107       1    0.779 0.03556         0.713         0.852
##   396    106       1    0.772 0.03597         0.705         0.846
##   400    105       1    0.765 0.03637         0.697         0.839
##   403    104       1    0.757 0.03676         0.689         0.833
##   408    103       1    0.750 0.03713         0.681         0.826
##   409    102       1    0.743 0.03749         0.673         0.820
##   411    101       1    0.735 0.03783         0.665         0.813
##   432    100       1    0.728 0.03816         0.657         0.807
##   446     98       1    0.721 0.03849         0.649         0.800
##   458     97       1    0.713 0.03880         0.641         0.793
##   465     96       1    0.706 0.03910         0.633         0.787
##   476     95       1    0.698 0.03939         0.625         0.780
##   486     94       1    0.691 0.03966         0.617         0.773
##   499     93       1    0.683 0.03993         0.609         0.766
##   517     91       1    0.676 0.04019         0.602         0.759
##   518     90       1    0.668 0.04044         0.594         0.752
##   531     89       1    0.661 0.04067         0.586         0.746
##   539     88       1    0.653 0.04090         0.578         0.739
##   575     87       1    0.646 0.04111         0.570         0.732
##   603     85       1    0.638 0.04132         0.562         0.725
##   609     84       1    0.631 0.04153         0.554         0.718
##   614     83       1    0.623 0.04171         0.546         0.710
##   651     82       1    0.615 0.04189         0.539         0.703
##   658     81       1    0.608 0.04206         0.531         0.696
##   664     80       1    0.600 0.04221         0.523         0.689
##   670     79       1    0.593 0.04236         0.515         0.682
##   671     78       1    0.585 0.04249         0.507         0.675
##   692     77       1    0.577 0.04261         0.500         0.667
##   696     76       1    0.570 0.04272         0.492         0.660
##   697     75       1    0.562 0.04282         0.484         0.653
##   707     74       2    0.547 0.04299         0.469         0.638
##   727     72       1    0.539 0.04306         0.461         0.631
##   736     71       1    0.532 0.04312         0.454         0.623
##   743     70       1    0.524 0.04317         0.446         0.616
##   759     69       1    0.517 0.04321         0.439         0.609
##   762     68       1    0.509 0.04323         0.431         0.601
```

```
##    768     67       1    0.501 0.04325        0.423        0.594
##    774     66       1    0.494 0.04326        0.416        0.586
##    805     65       1    0.486 0.04326        0.408        0.579
##    829     64       1    0.479 0.04324        0.401        0.571
##    834     63       1    0.471 0.04322        0.394        0.564
##    874     62       2    0.456 0.04314        0.379        0.549
##    892     60       1    0.448 0.04308        0.371        0.541
##    907     59       1    0.441 0.04302        0.364        0.534
##    963     57       1    0.433 0.04295        0.356        0.526
##    964     56       1    0.425 0.04288        0.349        0.518
##   1007     55       1    0.417 0.04279        0.342        0.510
##   1020     54       1    0.410 0.04269        0.334        0.503
##   1021     53       1    0.402 0.04258        0.327        0.495
##   1034     52       1    0.394 0.04245        0.319        0.487
##   1102     51       1    0.387 0.04232        0.312        0.479
##   1164     50       1    0.379 0.04217        0.305        0.471
##   1241     49       1    0.371 0.04202        0.297        0.463
##   1305     48       1    0.363 0.04185        0.290        0.455
##   1370     43       1    0.355 0.04172        0.282        0.447
##   1409     39       1    0.346 0.04163        0.273        0.438
##   1621     33       1    0.335 0.04166        0.263        0.428
##   2283      4       1    0.252 0.07904        0.136        0.466
```

```
summary(survfit(Surv(futime,death) ~ 1, data = myeloid), times = 365.25)
```

```
## Call: survfit(formula = Surv(futime, death) ~ 1, data = myeloid)
##
##   time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    365    108      28    0.794  0.0347        0.729        0.865
```

```
myeloid %>%
  filter(death==1) %>%
  summarize(median_surv =median(futime))
```

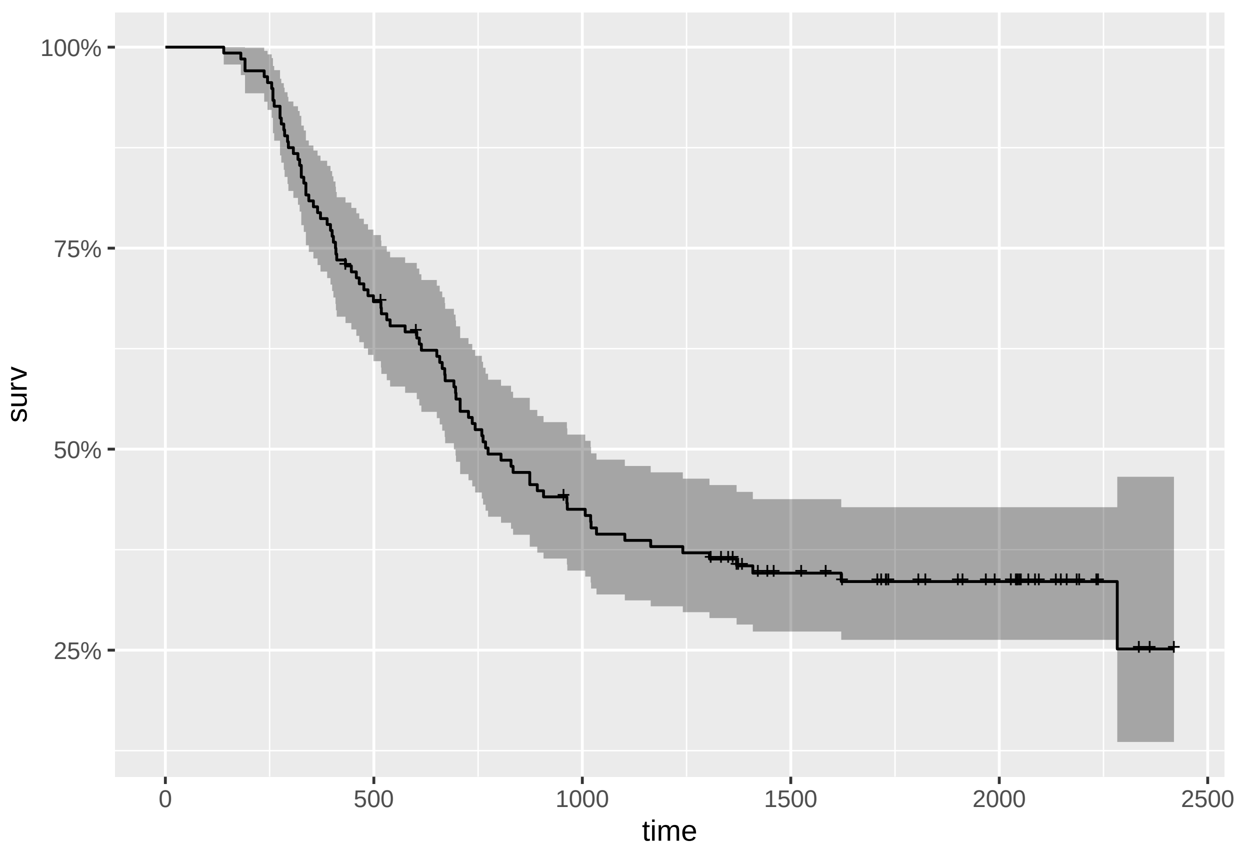
```
##   median_surv
## 1         518
```

```
# Create for loop to conduct between group significance test using a log rank test. The log-rank test w

for (i in colnames(myeloid)) {
  if (i != "futime" && i != "death" && i != "id" && i != "txtime" && i!="crtime" && i!="rltime") { # Ex
    formula <- as.formula(paste("Surv(futime, death) ~", i))
    result <- survdiff(formula, data = myeloid)
    print(result) # Display the results
  }
}
```

```
## Call:
## survdiff(formula = formula, data = myeloid)
##
##        N Observed Expected (O-E)^2/E (O-E)^2/V
## trt=A 60       43     34.2      2.27      3.71
```

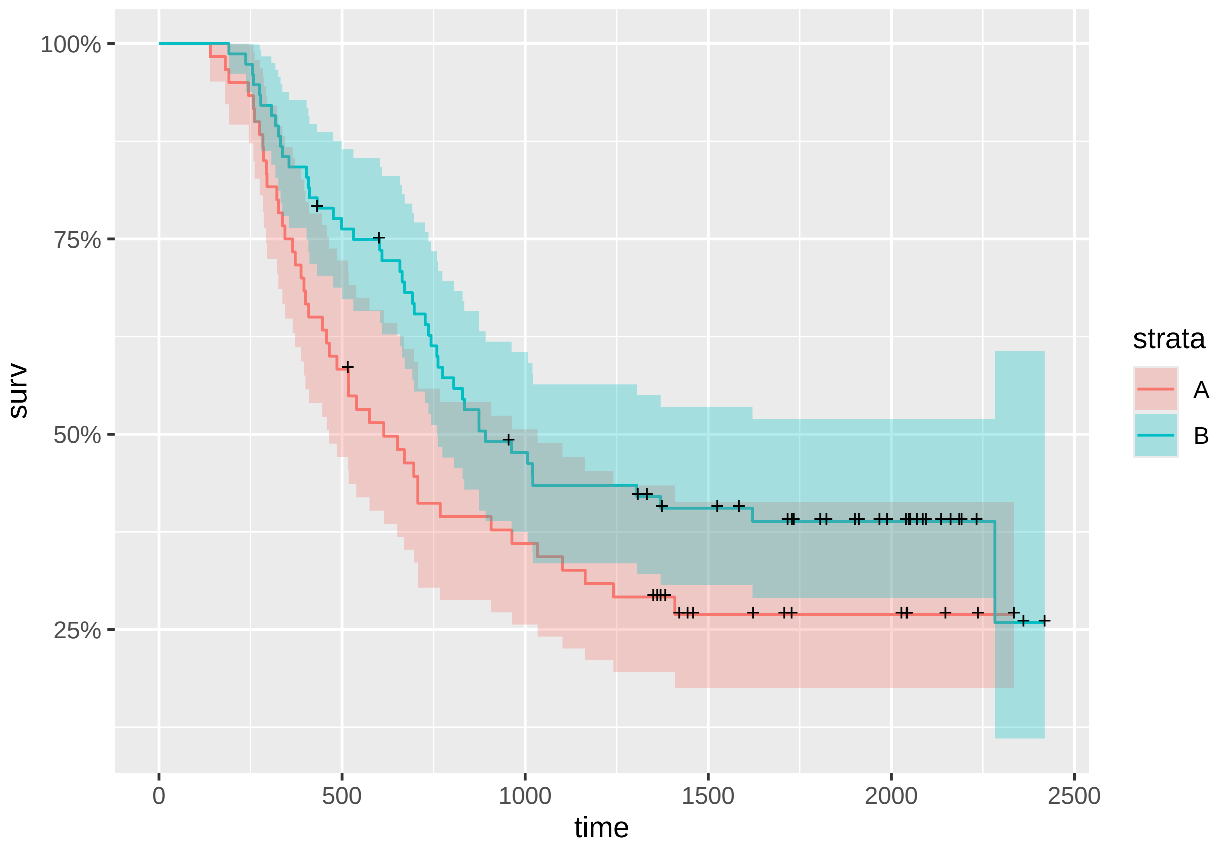
```
## trt=B 76       46     54.8      1.42      3.71
##
##  Chisq= 3.7  on 1 degrees of freedom, p= 0.05
## Call:
## survdiff(formula = formula, data = myeloid)
##
##        N Observed Expected (O-E)^2/E (O-E)^2/V
## sex=f 79       50     52.4     0.111      0.27
## sex=m 57       39     36.6     0.158      0.27
##
##  Chisq= 0.3  on 1 degrees of freedom, p= 0.6
## Call:
## survdiff(formula = formula, data = myeloid)
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## flt3=A 21       14     14.9    0.0518    0.0625
## flt3=B 69       41     49.8    1.5497    3.5451
## flt3=C 46       34     24.3    3.8353    5.3245
##
##  Chisq= 5.5  on 2 degrees of freedom, p= 0.06
```

```r
autoplot(km_fit)
```




```r
# Looking at survival times stratified by treatment
km_trt_fit <- survfit(Surv(futime, death) ~ trt, data=myeloid)
```

```r
autoplot(km_trt_fit)
```




```r
# Fitting a cox proportional hazards model (semi-parametric multiplicative model) making use of covaira
cox <- coxph(Surv(futime, death) ~ trt + sex + flt3 + txtime + crtime +rltime ,  data = myeloid)
summary(cox)
```

```
## Call:
## coxph(formula = Surv(futime, death) ~ trt + sex + flt3 + txtime +
##     crtime + rltime, data = myeloid)
##
##   n= 136, number of events= 89
##
##              coef  exp(coef)   se(coef)      z Pr(>|z|)
## trtB   -0.2922871  0.7465541  0.2314302 -1.263  0.20660
## sexm    0.2554729  1.2910720  0.2295895  1.113  0.26582
## flt3B  -0.1508864  0.8599454  0.3163588 -0.477  0.63340
## flt3C   0.1486204  1.1602325  0.3319984  0.448  0.65440
## txtime -0.0001530  0.9998470  0.0006649 -0.230  0.81799
## crtime -0.0021755  0.9978268  0.0036357 -0.598  0.54959
## rltime -0.0020069  0.9979952  0.0006936 -2.893  0.00381 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##        exp(coef) exp(-coef) lower .95 upper .95
```

```
## trtB       0.7466     1.3395     0.4743     1.1750
## sexm       1.2911     0.7746     0.8232     2.0248
## flt3B      0.8599     1.1629     0.4626     1.5987
## flt3C      1.1602     0.8619     0.6053     2.2240
## txtime     0.9998     1.0002     0.9985     1.0012
## crtime     0.9978     1.0022     0.9907     1.0050
## rltime     0.9980     1.0020     0.9966     0.9994
##
## Concordance= 0.708  (se = 0.028 )
## Likelihood ratio test= 27.53  on 7 df,   p=3e-04
## Wald test            = 23.05  on 7 df,   p=0.002
## Score (logrank) test = 23.23  on 7 df,   p=0.002
```

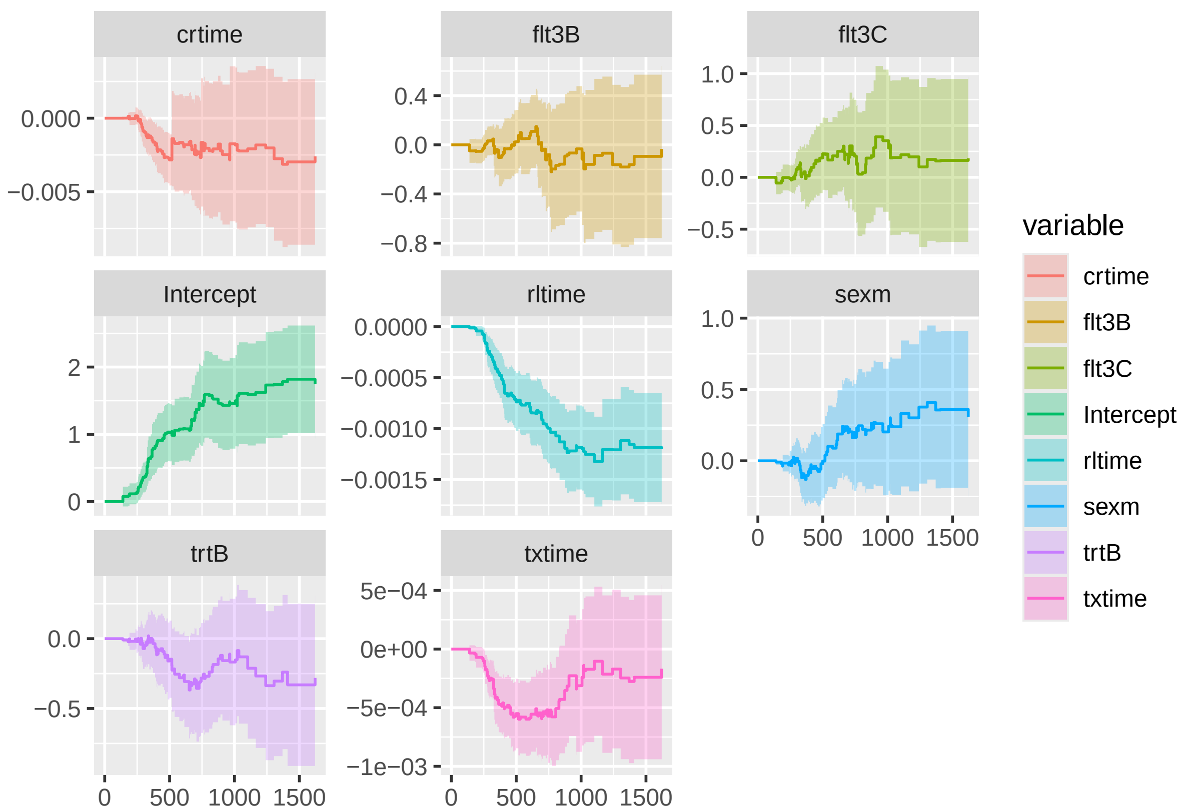
```r
cox_fit <- survfit(cox)

#plot(cox_fit, main = "cph model", xlab="Days")
autoplot(cox_fit)
```

The only significant covariate is rltime (time to relapse of disease) with a p-value > .01.

```r
# The Aalen model(non-parametric additive model) assumes that the cumulative hazard H(t) for a subject

aa_fit <-aareg(Surv(futime, death) ~ trt + sex + flt3 + txtime + crtime +rltime ,  data = myeloid)

aa_fit
```

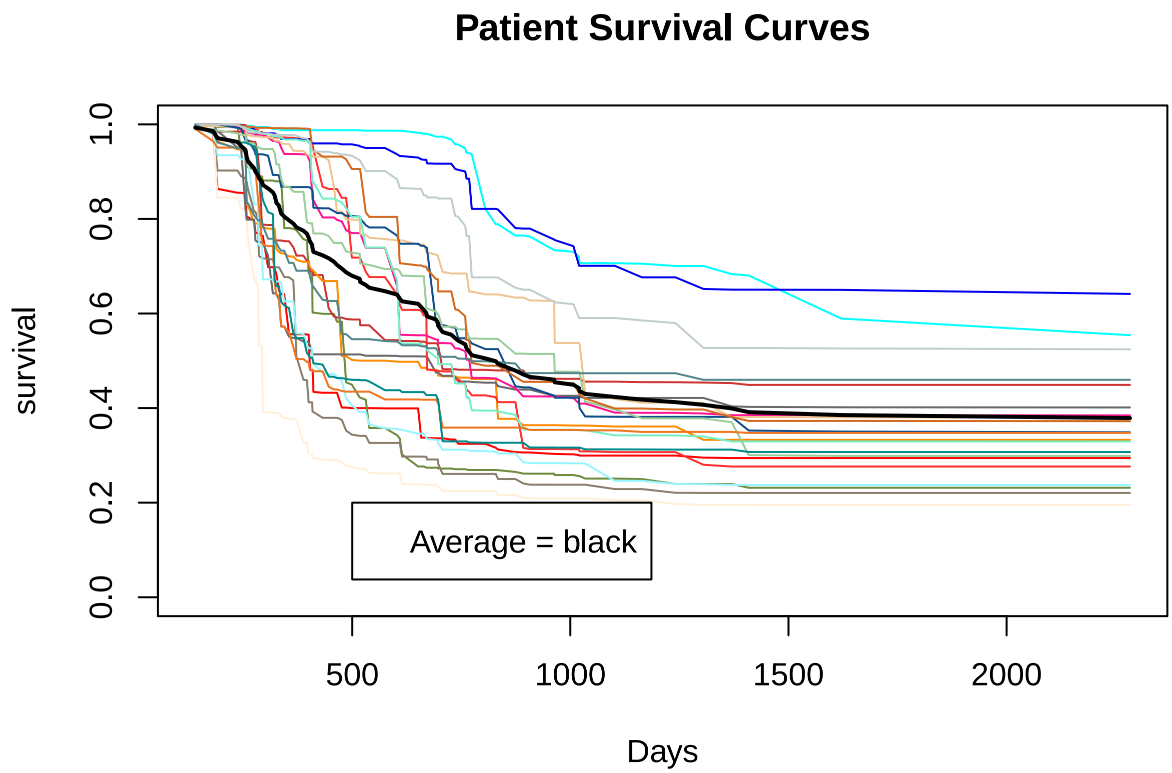
```
## Call:
## aareg(formula = Surv(futime, death) ~ trt + sex + flt3 + txtime +
##     crtime + rltime, data = myeloid)
##
##   n= 136
##     81 out of 82 unique event times used
##
##                 slope      coef se(coef)      z        p
## Intercept  2.55e-03  2.13e-02 4.35e-03  4.890 9.94e-07
## trtB      -4.39e-04 -3.32e-03 2.79e-03 -1.190 2.34e-01
## sexm       2.24e-04  2.69e-03 2.69e-03  0.998 3.18e-01
## flt3B     -1.09e-04 -1.03e-03 3.22e-03 -0.319 7.50e-01
## flt3C      3.67e-04  2.50e-03 3.88e-03  0.645 5.19e-01
## txtime    -7.59e-07 -3.70e-06 3.62e-06 -1.020 3.06e-01
## crtime    -4.02e-06 -3.19e-05 2.99e-05 -1.070 2.85e-01
## rltime    -1.41e-06 -1.41e-05 2.90e-06 -4.870 1.11e-06
##
## Chisq=40.71 on 7 df, p=9.21e-07; test weights=aalen
```

```
autoplot(aa_fit)
```




```
# Ranger builds a model for each obs in the dataset


# Model using the same variables as the cox model
```

```r
r_fit <- ranger(Surv(futime, death) ~ trt + sex + flt3 + txtime + crtime +rltime ,  data = myeloid, mtry
                     importance = "permutation",
                     splitrule = "extratrees",
                     verbose = TRUE)

death_times <- r_fit$unique.death.times
surv_prob <- data.frame(r_fit$survival)
avg_prob <- sapply(surv_prob, mean)


# Plot the survival models for each patient
plot(r_fit$unique.death.times,r_fit$survival[1,],
     type = "l",
     ylim = c(0,1),
     col = "red",
     xlab = "Days",
     ylab = "survival",
     main = "Patient Survival Curves")

cols <- colors()

for (n in sample(c(2:dim(myeloid)[1]), 20)) {
  lines(r_fit$unique.death.times, r_fit$survival[n,], type = "l", col = cols[n])
}

lines(death_times, avg_prob, lwd = 2)
legend(500, 0.2, legend = c('Average = black'))
```




```
# Ranking importance of each variable from the RF model
vi <- data.frame(sort(round(r_fit$variable.importance, 4), decreasing = TRUE))
names(vi) <- "importance"
head(vi)
```

```
##          importance
## rltime       0.0464
## txtime       0.0230
## sex          0.0222
## trt          0.0175
## crtime      -0.0014
## flt3        -0.0022
```

```
# The Harrell's c-index is similar to the concordance statistic
# This is a generalization of the ROC curve which reduces the Wilcoxon-Mann-Whitney statistic for binar

cat("Prediction Error = 1 - Harrell's c-index = ", r_fit$prediction.error)
```

```
## Prediction Error = 1 - Harrell's c-index =  0.3207212
```

```
# Set up for ggplot
kmi <- rep("KM",length(km_fit$time))
km_df <- data.frame(km_fit$time,km_fit$surv,kmi)
names(km_df) <- c("Time","Surv","Model")
```

```r
coxi <- rep("Cox",length(cox_fit$time))
cox_df <- data.frame(cox_fit$time,cox_fit$surv,coxi)
names(cox_df) <- c("Time","Surv","Model")

rfi <- rep("RF",length(r_fit$unique.death.times))
rf_df <- data.frame(r_fit$unique.death.times,avg_prob,rfi)
names(rf_df) <- c("Time","Surv","Model")

plot_df <- rbind(km_df,cox_df,rf_df)

p <- ggplot(plot_df, aes(x = Time, y = Surv, color = Model))
p + geom_line()
```