P8108 Group 2 Survival Analysis Project

Yiming Zhao (yz3955) Wenshan Qu (wq2160) Tucker Morgan (tlm2152)
Junzhe Shao (js5959) Benjamin Goebel (bpg2118)

2022-10-19

```
library(survival)
library(tidyverse)
library(tidymodels)
library(glmnet)
library(ranger)
library(survminer)
library(arsenal)
knitr::opts_chunk$set(message = FALSE, warning = FALSE)
```

Train Test Split

```
set.seed(2022)

rotterdam_split <- initial_split(rotterdam, prop = 0.8, strata = death)
rotterdam_training <- training(rotterdam_split)
rotterdam_test <- testing(rotterdam_split)
```

Perform 10-fold Cross-Validation

The output contains 1 row for each fold/repeat. So, 10 folds * 5 repeats = 50 rows. The split_analysis column is a list column containing a data frame for each row with 9 folds combined, and the split_assessment column is a list column containing a data frame for each row with 1 fold.

```
set.seed(2022)

rotterdam_folds <- vfold_cv(rotterdam_training, v = 10, repeats = 5,
                            strata = death)

rotterdam_folds <- rotterdam_folds %>%
  mutate(split_analysis = map(splits, analysis),
         split_assessment = map(splits, assessment))
```

Introduction

Methods

The dataset of interest for this analysis comes from the Rotterdam tumor bank, including data from 2982 breast cancer patients. Follow up time for patients varied from just 1 month to as long as 231 months. Several prognostic variables are recorded including year of surgery, age at surgery, menopausal status (pre- or post-), tumor size (mm), differentiation grade, number of positive lymph nodes, progesterone receptors (fmol/l), estrogen receptors (fmol/l), and indicators for hormonal treatment and chemotherapy treatment. The outcome considered in this analysis was patient death.

(Placeholder for Cross-validation)

As part of this analysis, we consider the Cox Proportional Hazard (Cox PH) model, which allows us to model the hazard ratio based on covariates to understand their impact on the survival function. The Cox PH typically takes the form:

$$h(t|Z = z) = h_0(t)e^{\beta' z}.$$

In this application, we use the elastic net penalty, a mixture of the ℓ_1 and ℓ_2 norm regularization penalties. In the Cox PH framework, this penalty term takes the form of:

$$\lambda\Big(\alpha \sum |\beta_i| + \frac{1}{2}(1-\alpha)\sum \beta_i^2\Big)$$

where λ represents our penalty coefficient and α is the mixing parameter for the two regularization methods. This penalty helps to avoid over-fitting of our data. The algorithm used here in `glmnet` uses the Breslow approximation to handle ties. For more details on the derivation of this term and the algorithm used to fit the penalized Cox PH model, see Simon et al. (2011).

Exploratory Data Analysis

```
print(summary(tableby(hormon~age+meno+size+grade+nodes+pgr+er+chemo+dtime+death, rotterdam,numeric.simpl
```

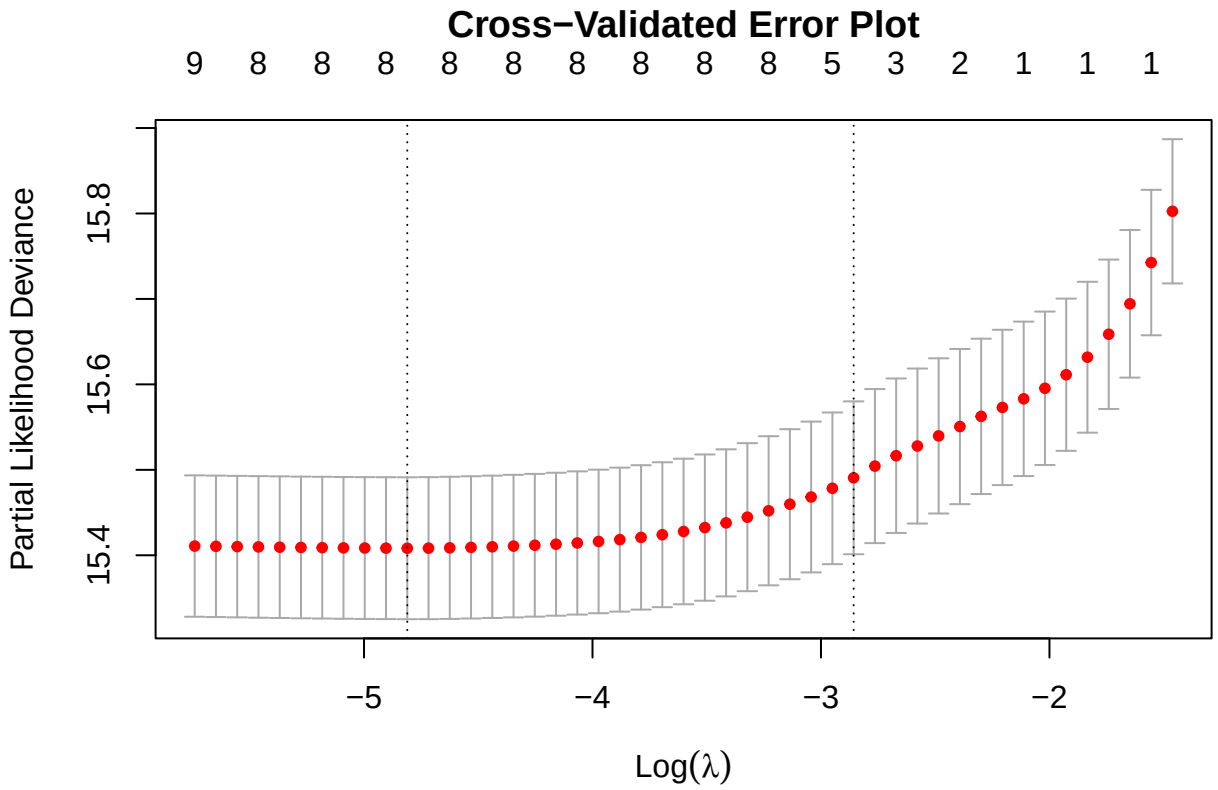
	0 (N=2643)	1 (N=339)	Total (N=2982)	p value
age				< 0.001
Mean (SD)	54.098 (12.984)	62.549 (9.921)	55.058 (12.953)	
Range	24.000 - 90.000	28.000 - 88.000	24.000 - 90.000	
meno				< 0.001
Mean (SD)	0.519 (0.500)	0.879 (0.327)	0.560 (0.496)	
Range	0.000 - 1.000	0.000 - 1.000	0.000 - 1.000	
size				< 0.001
<=20	1283 (48.5%)	104 (30.7%)	1387 (46.5%)	
20-50	1119 (42.3%)	172 (50.7%)	1291 (43.3%)	
>50	241 (9.1%)	63 (18.6%)	304 (10.2%)	
grade				< 0.001
Mean (SD)	2.722 (0.448)	2.826 (0.380)	2.734 (0.442)	
Range	2.000 - 3.000	2.000 - 3.000	2.000 - 3.000	
nodes				< 0.001
Mean (SD)	2.327 (4.207)	5.720 (4.576)	2.712 (4.384)	
Range	0.000 - 34.000	1.000 - 24.000	0.000 - 34.000	
pgr				< 0.001
Mean (SD)	168.706 (300.337)	108.233 (200.302)	161.831 (291.311)	

	0 (N=2643)	1 (N=339)	Total (N=2982)	p value
Range	0.000 - 5004.000	0.000 - 1497.000	0.000 - 5004.000	
er				0.069
Mean (SD)	164.792 (272.563)	180.608 (271.693)	166.590 (272.465)	
Range	0.000 - 3275.000	0.000 - 2444.000	0.000 - 3275.000	
chemo				< 0.001
Mean (SD)	0.209 (0.407)	0.083 (0.276)	0.195 (0.396)	
Range	0.000 - 1.000	0.000 - 1.000	0.000 - 1.000	
dtime				< 0.001
Mean (SD)	2679.067 (1309.178)	2030.534 (1043.971)	2605.340 (1298.078)	
Range	36.000 - 7043.000	45.000 - 6270.000	36.000 - 7043.000	
death				0.093
Mean (SD)	0.421 (0.494)	0.469 (0.500)	0.427 (0.495)	
Range	0.000 - 1.000	0.000 - 1.000	0.000 - 1.000	

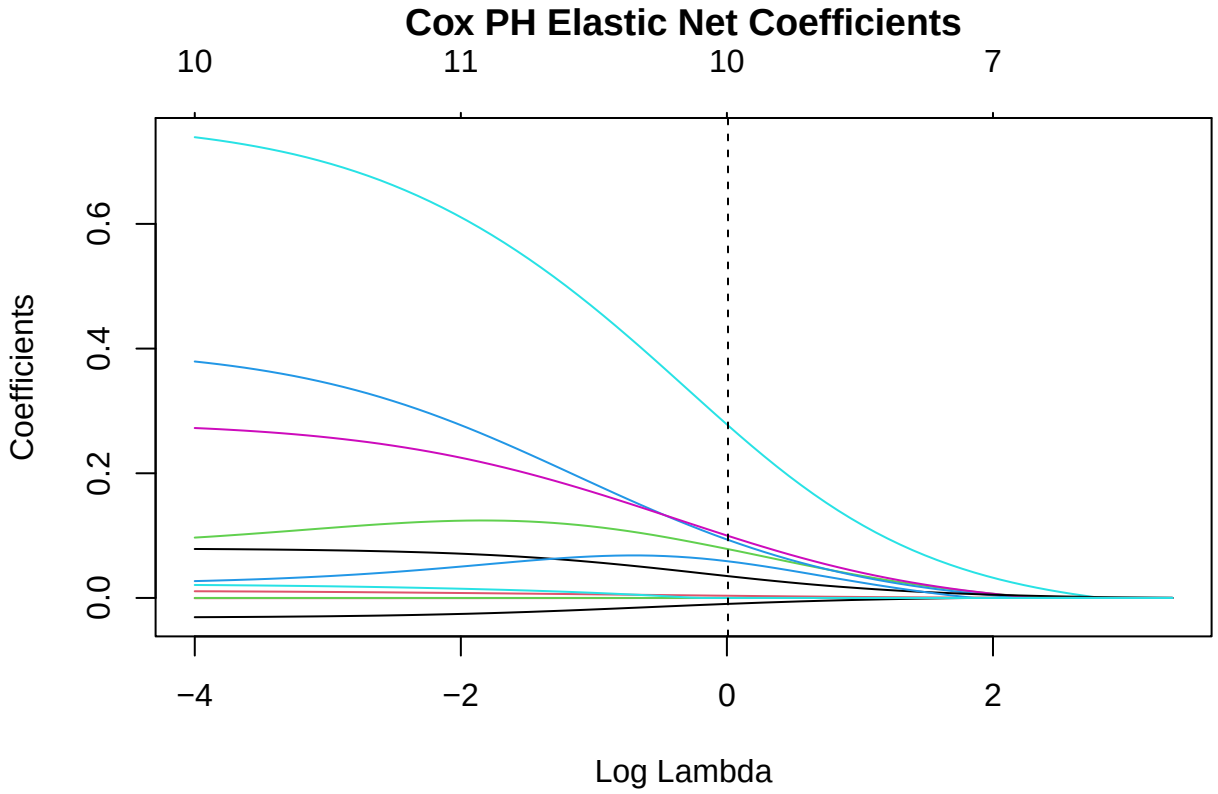
Cross-Validation

Cox/Cox with elastic net

```
set.seed(2022)
# removing relapse data, since death is the primary outcome
rotterdam_trn_d <-
  rotterdam_training %>%
  select(-rtime, -recur, -pid)

cox_trn_x <- model.matrix(Surv(dtime, death) ~ ., rotterdam_trn_d)[,-1]
cox_trn_y <- Surv(rotterdam_trn_d$dtime, rotterdam_trn_d$death)

cv_coxfit <- cv.glmnet(cox_trn_x, cox_trn_y, family = "cox", type.measure = "deviance")

par(mar = c(4,4,5,1))
plot(cv_coxfit, main = "Cross-Validated Error Plot")
```




```
coxnetfit <- glmnet(cox_trn_x, cox_trn_y, family = "cox", alpha = cv_coxfit$lambda.min)

par(mar = c(4,4,5,1))
plot(coxnetfit, xvar = "lambda",
     main = "Cox PH Elastic Net Coefficients")
abline(v = cv_coxfit$lambda.min, lty = 2)
```




```
coxnetfit_df <-
  data.frame(
    "coef" = as.vector(coef(coxnetfit, s = cv_coxfit$lambda.min)),
    "exp_coef" = as.vector(coef(coxnetfit, s = cv_coxfit$lambda.min)) %>% exp()
)

rownames(coxnetfit_df) <- labels(coef(coxnetfit, s = cv_coxfit$lambda.min))[[1]]

coxnetfit_df %>% round(digits = 4) %>%
  knitr::kable(caption = "Cox Proportion Hazard Elastic Net Coefficients")
```

Table 2: Cox Proportion Hazard Elastic Net Coefficients

	coef	exp_coef
year	-0.0309	0.9696
age	0.0108	1.0108
meno	0.0968	1.1016
size20-50	0.3793	1.4612
size>50	0.7391	2.0940
grade	0.2726	1.3134
nodes	0.0786	1.0817
pgr	-0.0004	0.9996
er	0.0000	1.0000
hormon	0.0270	1.0274

	coef	exp_coef
chemo	0.0208	1.0210

In the table above, we can see that the estrogen receptors covariate is selected out with a null value of 0 or $\exp(coef) = 1$. We can fit a cox proportional hazard model using only the selected covariates in the `coxph` function to find unbiased estimates of the coefficients along with standard errors and confidence intervals.

```
coxfit <- coxph(Surv(dtime, death) ~ year + age + meno + size + grade + nodes + pgr + hormon + chemo,
                data = rotterdam_training, ties = "breslow")
coxfit %>%
  broom::tidy() %>%
  mutate(estimate = exp(estimate))
```

```
## # A tibble: 10 x 5
##    term      estimate std.error statistic  p.value
##    <chr>        <dbl>     <dbl>     <dbl>    <dbl>
##  1 year         0.969  0.0114      -2.78  5.51e- 3
##  2 age          1.01   0.00425      2.76  5.73e- 3
##  3 meno         1.08   0.111        0.721 4.71e- 1
##  4 size20-50    1.50   0.0732       5.51  3.62e- 8
##  5 size>50      2.15   0.103        7.44  9.81e-14
##  6 grade        1.33   0.0802       3.52  4.27e- 4
##  7 nodes        1.08   0.00583     13.7   1.52e-42
##  8 pgr          1.00   0.000130    -2.83  4.59e- 3
##  9 hormon       1.02   0.103        0.210 8.33e- 1
## 10 chemo        1.02   0.0914       0.245 8.06e- 1
```

```
confint(coxfit) %>% exp() %>% knitr::kable()
```

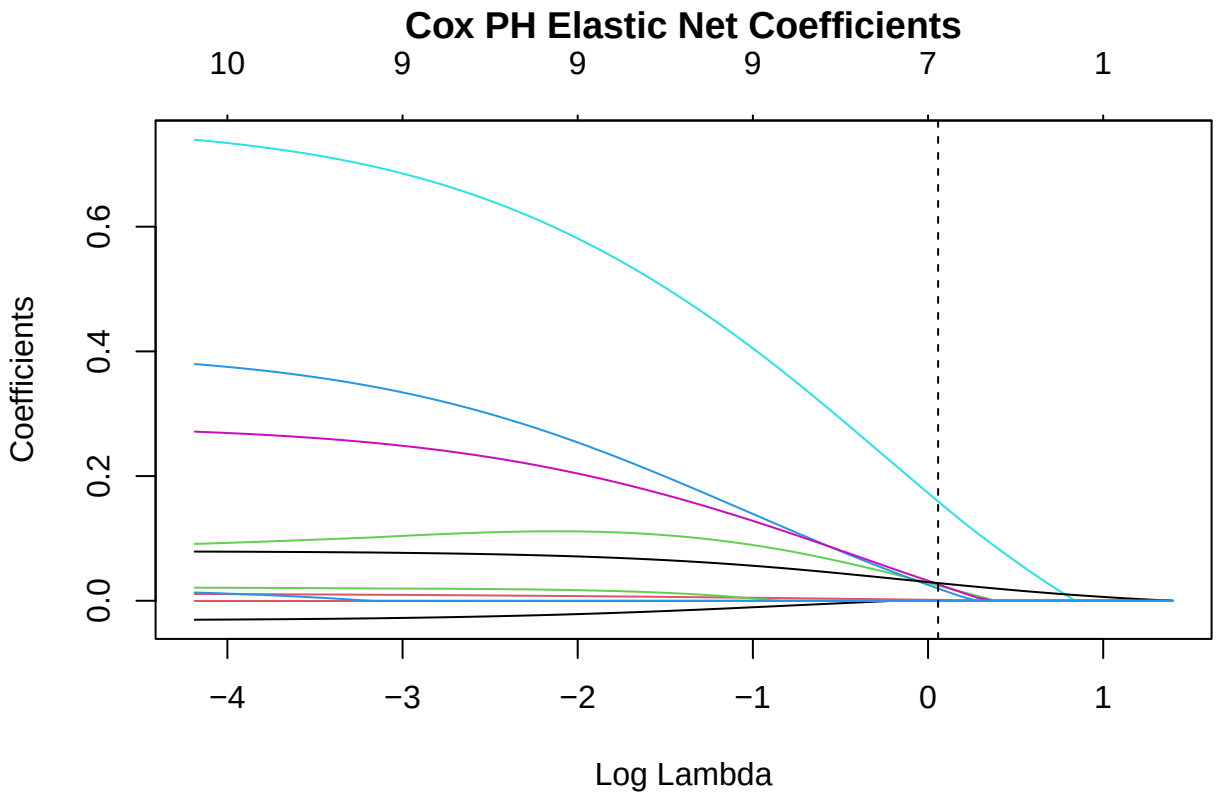
	2.5 %	97.5 %
year	0.9472377	0.9907047
age	1.0034196	1.0202786
meno	0.8713961	1.3469318
size20-50	1.2966461	1.7276336
size>50	1.7585648	2.6327710
grade	1.1335340	1.5523422
nodes	1.0706900	1.0954509
pgr	0.9993758	0.9998861
hormon	0.8346477	1.2513357
chemo	0.8549591	1.2232028

In our (minimum error) model, we find significant effects for year of surgery, age at surgery, size of tumor, differentiation grade, number of positive lymph nodes, and progesterone receptors. The largest magnitude effects come from increasing size of tumor.

Below, we see the analogous results for a "1se" rule model.

```
coxnetfit_1se <- glmnet(cox_trn_x, cox_trn_y, family = "cox", alpha = cv_coxfit$lambda.1se)

par(mar = c(4,4,5,1))
```

```
plot(coxnetfit_1se, xvar = "lambda",
     main = "Cox PH Elastic Net Coefficients")
abline(v = cv_coxfit$lambda.1se, lty = 2)
```




```
coxnetfit_1se_df <-
  data.frame(
    "coef" = as.vector(coef(coxnetfit_1se, s = cv_coxfit$lambda.1se)),
    "exp_coef" = as.vector(coef(coxnetfit_1se, s = cv_coxfit$lambda.1se)) %>% exp()
)

rownames(coxnetfit_1se_df) <- labels(coef(coxnetfit_1se, s = cv_coxfit$lambda.1se))[[1]]

coxnetfit_1se_df %>% round(digits = 4) %>%
  knitr::kable(caption = "Cox Proportion Hazard Elastic Net Coefficients (1se)")
```

Table 4: Cox Proportion Hazard Elastic Net Coefficients (1se)

	coef	exp_coef
year	-0.0269	0.9734
age	0.0092	1.0092
meno	0.1059	1.1117
size20-50	0.3254	1.3846
size>50	0.6744	1.9628

	coef	exp_coef
grade	0.2440	1.2763
nodes	0.0764	1.0793
pgr	-0.0003	0.9997
er	0.0000	1.0000
hormon	0.0194	1.0196
chemo	0.0000	1.0000

Here, we remove `er` and `chemo` and find the following results.

```
coxfit_1se <- coxph(Surv(dtime, death) ~ year + age + meno + size + grade + nodes + pgr + hormon,
                data = rotterdam_training, ties = "breslow")
coxfit_1se %>%
  broom::tidy() %>%
  mutate(estimate = exp(estimate))
```

```
## # A tibble: 9 x 5
##   term      estimate std.error statistic  p.value
##   <chr>        <dbl>     <dbl>     <dbl>    <dbl>
## 1 year         0.969  0.0114      -2.78  5.39e- 3
## 2 age          1.01   0.00419      2.76  5.76e- 3
## 3 meno         1.08   0.109        0.687 4.92e- 1
## 4 size20-50    1.50   0.0732       5.51  3.56e- 8
## 5 size>50      2.15   0.103        7.46  8.78e-14
## 6 grade        1.33   0.0802       3.53  4.22e- 4
## 7 nodes        1.08   0.00570     14.0   7.79e-45
## 8 pgr          1.00   0.000130    -2.83  4.67e- 3
## 9 hormon       1.02   0.103        0.198 8.43e- 1
```

```
confint(coxfit_1se) %>% exp() %>% knitr::kable()
```

	2.5 %	97.5 %
year	0.9471652	0.9906283
age	1.0033625	1.0199828
meno	0.8702403	1.3353102
size20-50	1.2969056	1.7279709
size>50	1.7605408	2.6350041
grade	1.1338007	1.5526634
nodes	1.0712983	1.0954912
pgr	0.9993772	0.9998870
hormon	0.8337266	1.2494257

Here we again find significant effects for year of surgery, age at surgery, size of tumor, differentiation grade, number of positive lymph nodes, and pgr.

Random survival forest

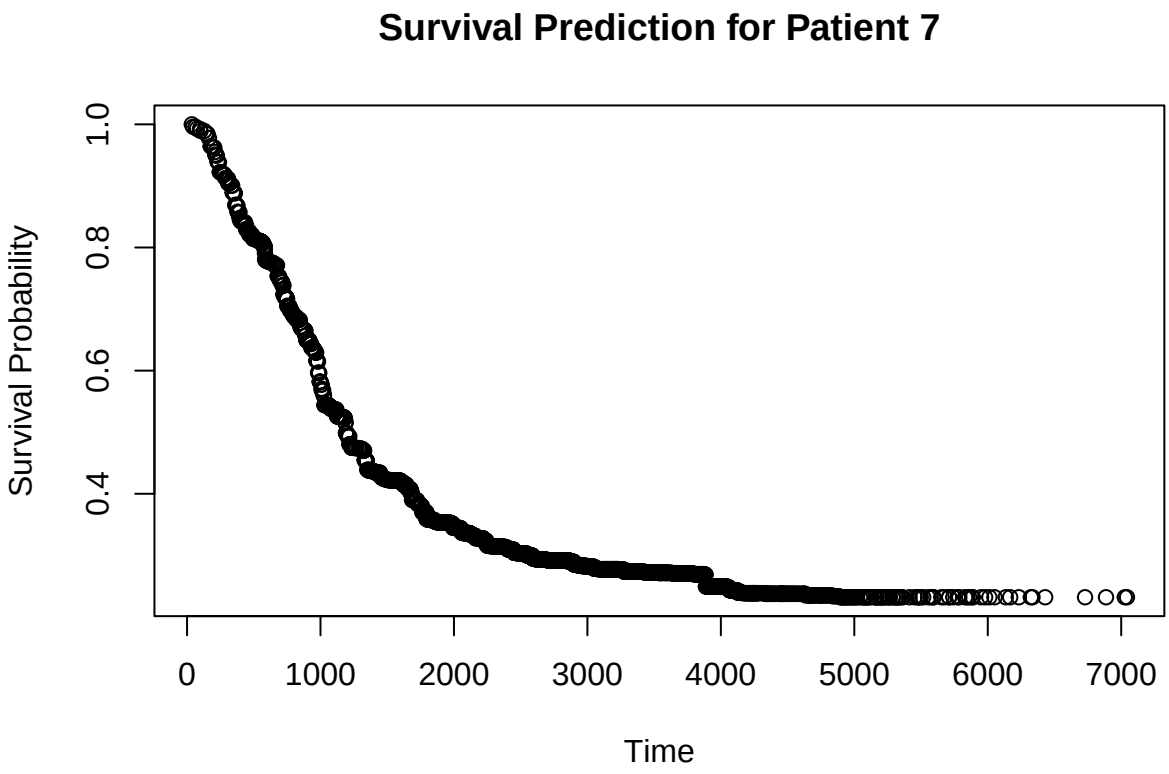
The survival tree and the corresponding random survival forest (RSF) are highly favorable non-parametric methods when studying survival data. Generally, for a single survival tree, it will assign subjects to groups

based on certain splitting rules regarding their covariates, and the subjects in each group will share a similar survival behavior.

```
## Random Survival Forest
rsf <- ranger(Surv(time = dtime, event = death) ~ .,
              data = rotterdam_trn_d,
              num.trees = 300,
              min.node.size = 15)

## Remove variables not for prediction, and the outcome
rotterdam_test_d <-
  rotterdam_test %>%
  select(-rtime, -recur, -pid, -death)

## Make prediction on all the test data points
pred_rsf <- predict(rsf, rotterdam_test_d, type = "response")
# Look at individual 7
pred_ref_7 <- data.frame(
  time = pred_rsf$unique.death.times,
  survival = pred_rsf$survival[7,])
head(pred_ref_7) %>% knitr::kable(align = "c")
```

time	survival
36	1.0000000
45	0.9960619
64	0.9960619
74	0.9928333
97	0.9925576
101	0.9892727

```
plot(pred_ref_7$time, pred_ref_7$survival,
     xlab = "Time", ylab = "Survival Probability",
     main = "Survival Prediction for Patient 7")
```




```
# Find estimated median survival time for individual 7
head(pred_ref_7[pred_ref_7$survival <= 0.5,]) %>% knitr::kable(align = "c") #1163
```

	time	survival
346	1191	0.4986885
347	1192	0.4984652
348	1193	0.4984652
349	1198	0.4984652
350	1200	0.4948371
351	1204	0.4931904

```
# See the truth of individual 7
rotterdam_test[7,] %>% knitr::kable(align = "c")
```

	pid	year	age	meno	size	grade	nodes	pgr	er	hormon	chemo	rtime	recur	dtime	death
2463	58	1992	69	1	20-50	2	8	5	6	1	0	1869	0	1869	0

With `ranger` package, we trained the random survival forest with training dataset used for survival prediction. As a non-parametric method, there is no parameters in RSF that could be interpreted. The ultimate goal of RSF is to predict the survival probability function of a given data point based on its covariate vector.

Compared to semi-parametric Cox-PH model which forces the outcome and the covariates to have a special connection, the RSF makes prediction based on the survival time of training data points that shares similar propensity with the given input data point.

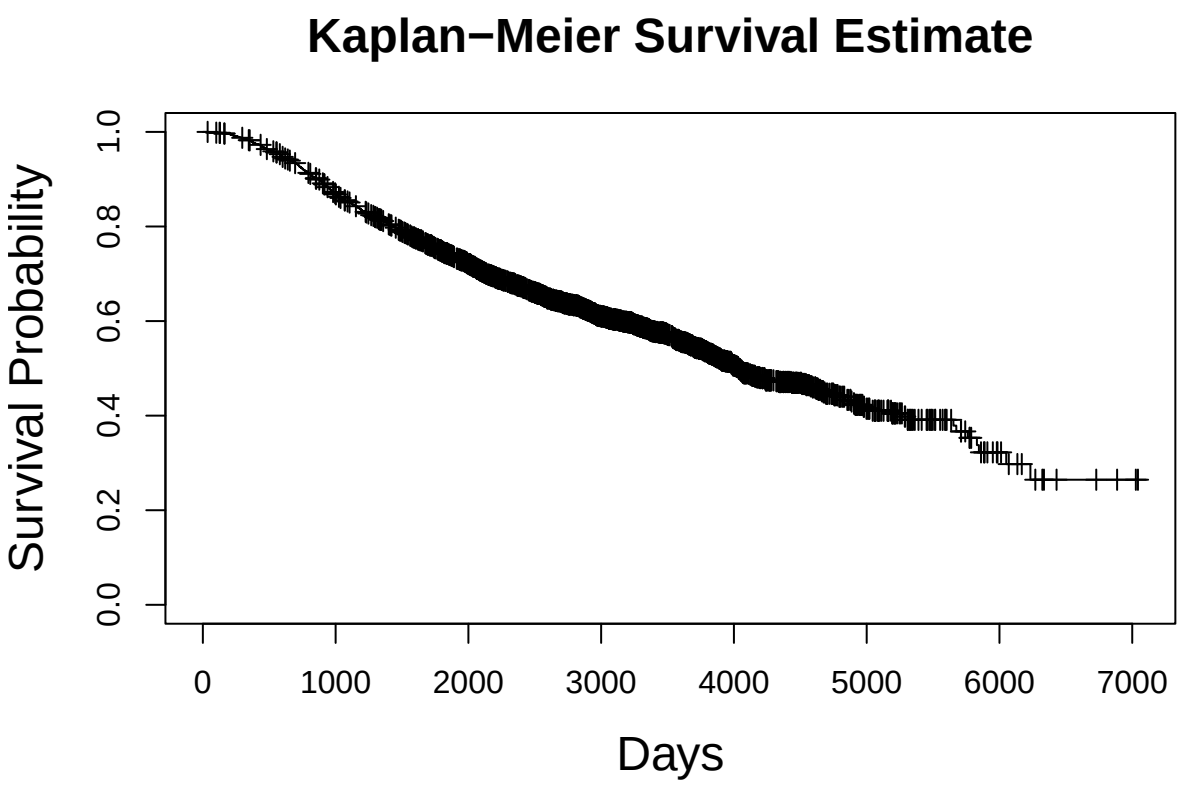
Since the "truth" of test data point (a single survival time) and the prediction we made here (a survival probability function) are not comparable, here we show the prediction result of the 7th test data point (pid = 58). The survival curve has been shown above, and the median survival time is 1163 days.

Conformalized survival analysis

Supplemental analyses

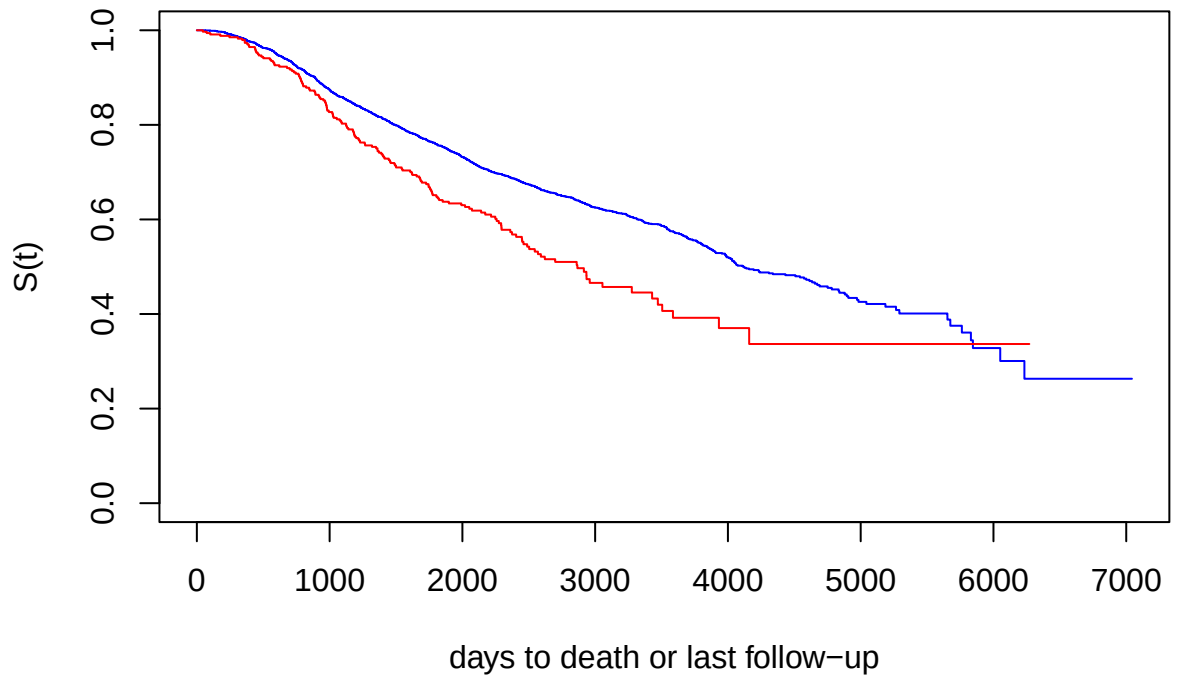
Kaplan-Meier Survival Estimate

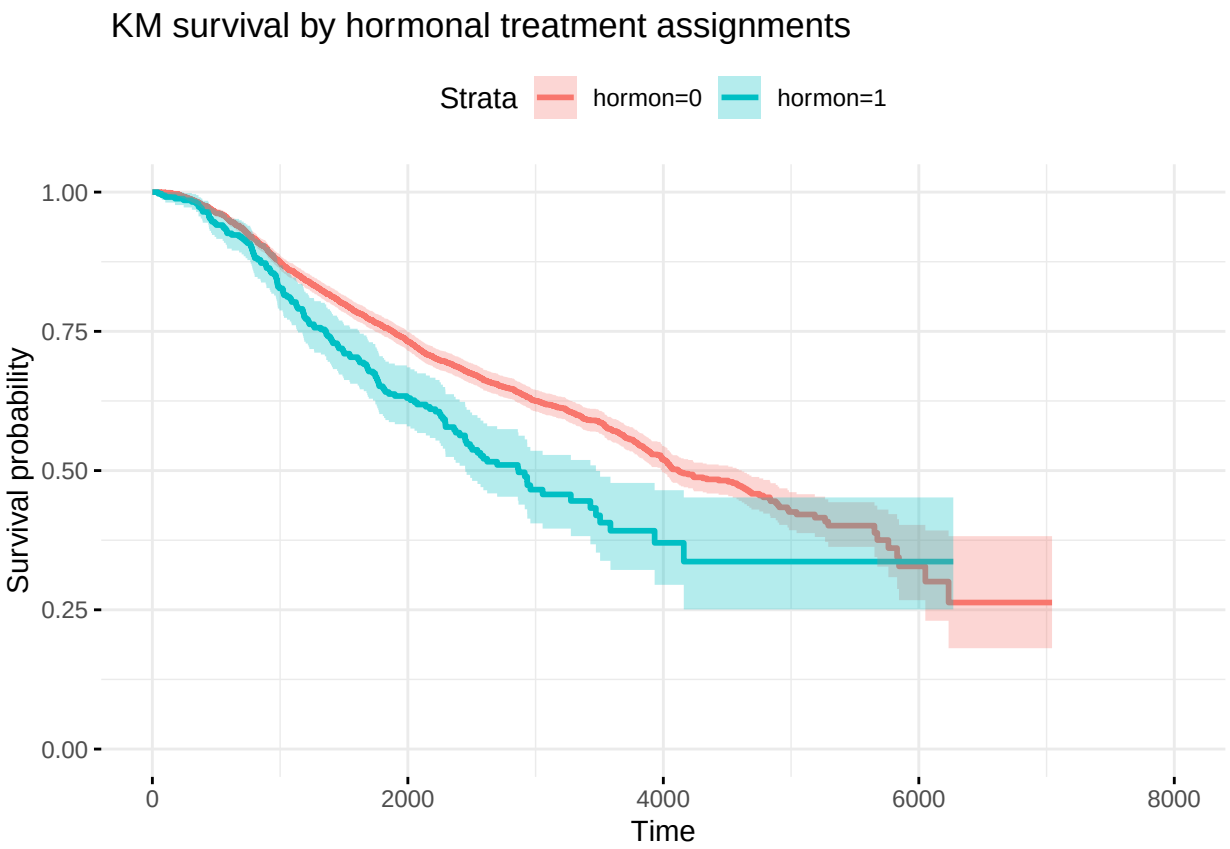
```
KM = survfit(Surv(dtime, death) ~ 1, data = rotterdam)
plot(KM, conf.int = FALSE, mark.time = TRUE,
     xlab = "Days", ylab = "Survival Probability",
     main = "Kaplan-Meier Survival Estimate", cex.lab = 1.5, cex.main = 1.5)
```




```
# make Kaplan-Meier estimates
kmfit <- survfit(Surv(dtime, death) ~ hormon, data = rotterdam,type=c("kaplan-meier"))
# print Kaplan-Meier table
#summary(kmfit)
```

```
plot(kmfit,
ylab="S(t)",
xlab="days to death or last follow-up",
main = "Kaplan Meier estimates of Breast cancer survival by hormonal treatment assignments for rotterdar
col = c("blue","red"))

ggsurvplot(kmfit, conf.int = 0.95, censor= F,title = " KM survival by hormonal treatment assignments",
           ggtheme = theme_minimal())
```





Log-rank Test

The null hypothesis of our log-rank test is: $H_0 : S_1(t) = S_0(t)$, where $S_1(t)$ is the survival function of hormon treatment group, $S_0(t)$ is the survival function of control group.

```
logrank <- survdiff(Surv(dtime, death) ~ hormon, data = rotterdam)
logrank
```

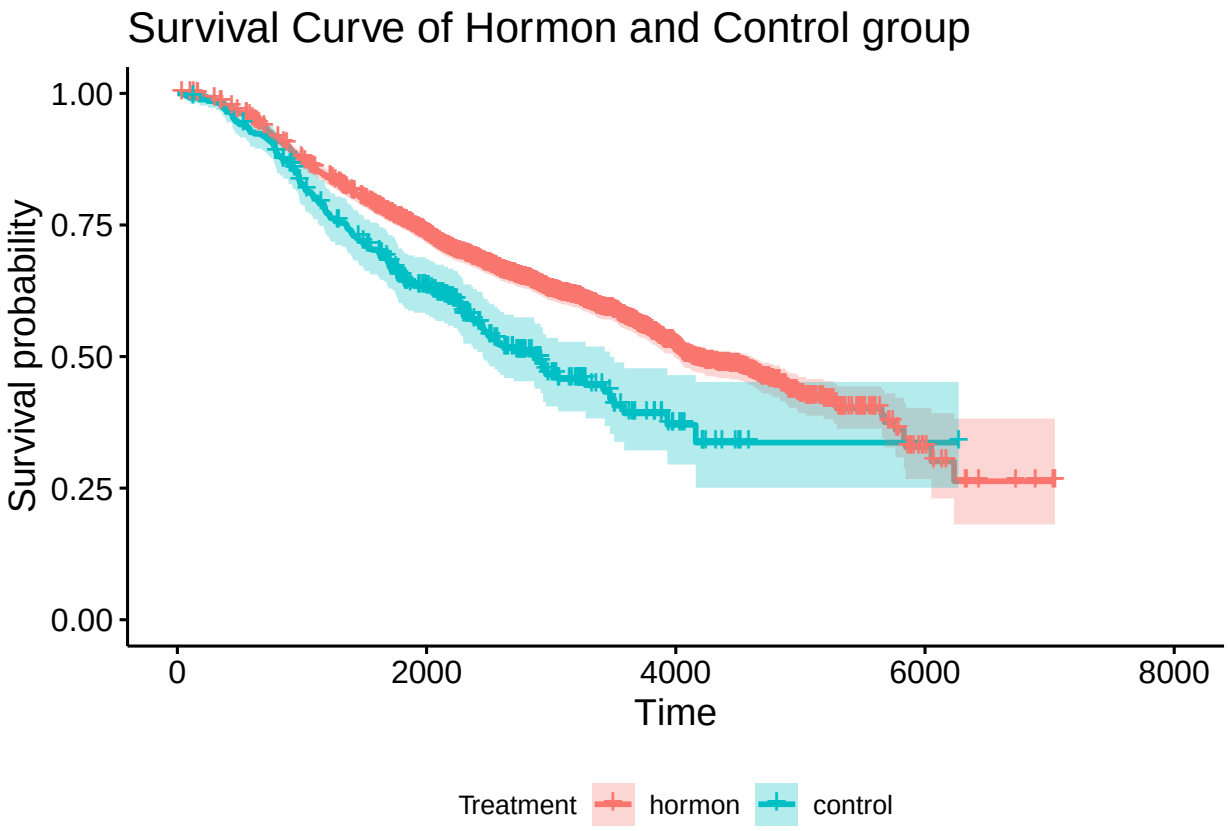
```
## Call:
## survdiff(formula = Surv(dtime, death) ~ hormon, data = rotterdam)
##
##             N Observed Expected (O-E)^2/E (O-E)^2/V
## hormon=0 2643     1113     1162      2.04      23.7
## hormon=1  339      159      110     21.43      23.7
##
##  Chisq= 23.7  on 1 degrees of freedom, p= 1e-06
```

```
logrank$pvalue
```

```
## [1] 1.133649e-06
```

The test statistic is 23.7, and the corresponding p-value is $1.133^{-6} \ll 0.05$, thus we reject the null and conclude that we are 95% confident that $S_1(t) \neq S_0(t$. And since the test statistic is positive, we can conclude that the hormon treatment is significantly effective to breast cancer.

```
ggsurvplot(survfit(Surv(dtime,death) ~ hormon, data = rotterdam),
           conf.int = TRUE,
           legend = c("bottom"),
           legend.title = c("Treatment"),
           legend.labs = c("hormon", "control")) +
  ggtitle("Survival Curve of Hormon and Control group")
```



Results

Discussion

How our results compare with past research

Conclusion

References

—Note this reference is in MLA format—

Simon, Noah et al. “Regularization Paths for Cox’s Proportional Hazards Model via Coordinate Descent.” Journal of statistical software vol. 39,5 (2011): 1-13. doi:10.18637/jss.v039.i05