P8108 Group 2 Survival Analysis Project

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

2022-12-1

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

Train Validation Test Split

```
set.seed(2022)
rotterdam = rotterdam %>%
  mutate(hormon = as.factor(hormon),
         chemo = as.factor(chemo))

rotterdam_split <- initial_split(select(rotterdam, -rtime, -recur, -pid),
                                 prop = 0.8, strata = death)
rotterdam_training <- training(rotterdam_split)
rotterdam_test <- testing(rotterdam_split)

rotterdam_train_val_split <- initial_split(rotterdam_training,
                                           prop = 0.8, strata = death)
rotterdam_training <- training(rotterdam_train_val_split)
rotterdam_validation <- testing(rotterdam_train_val_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

For our target population is hormone treatment an effective therapy in breast cancer survival? For our target population, is chemotherapy an effective therapy in breast cancer survival? How do predictions from non-parametric models like the random forest compare to semi-parametric in the Cox proportional hazard model?

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.simplify = TRUE, numeric.test = "kwt")))
```

	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

	0 (N=2643)	1 (N=339)	Total (N=2982)	p value
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)	
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
0	2091 (79.1%)	311 (91.7%)	2402 (80.5%)	
1	552 (20.9%)	28 (8.3%)	580 (19.5%)	
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	

```r
print(summary(tableby(chemo~age+meno+size+grade+nodes+pgr+er+hormon+dtime+death,
                      rotterdam,numeric.simplify = TRUE, numeric.test = "kwt")))
```

	0 (N=2402)	1 (N=580)	Total (N=2982)	p value
age				< 0.001
Mean (SD)	57.560 (12.775)	44.698 (7.322)	55.058 (12.953)	
Range	25.000 - 90.000	24.000 - 73.000	24.000 - 90.000	
meno				< 0.001
Mean (SD)	0.658 (0.474)	0.153 (0.361)	0.560 (0.496)	
Range	0.000 - 1.000	0.000 - 1.000	0.000 - 1.000	
size				0.002
<=20	1148 (47.8%)	239 (41.2%)	1387 (46.5%)	
20-50	1028 (42.8%)	263 (45.3%)	1291 (43.3%)	
>50	226 (9.4%)	78 (13.4%)	304 (10.2%)	
grade				0.964
Mean (SD)	2.734 (0.442)	2.734 (0.442)	2.734 (0.442)	
Range	2.000 - 3.000	2.000 - 3.000	2.000 - 3.000	
nodes				< 0.001
Mean (SD)	2.353 (4.240)	4.198 (4.651)	2.712 (4.384)	
Range	0.000 - 34.000	1.000 - 34.000	0.000 - 34.000	
pgr				0.002
Mean (SD)	157.556 (283.077)	179.536 (322.848)	161.831 (291.311)	

	0 (N=2402)	1 (N=580)	Total (N=2982)	p value
Range	0.000 - 3000.000	0.000 - 5004.000	0.000 - 5004.000	
er				< 0.001
Mean (SD)	183.599 (286.924)	96.148 (186.163)	166.590 (272.465)	
Range	0.000 - 3275.000	0.000 - 1929.000	0.000 - 3275.000	
hormon				< 0.001
0	2091 (87.1%)	552 (95.2%)	2643 (88.6%)	
1	311 (12.9%)	28 (4.8%)	339 (11.4%)	
dtime				0.805
Mean (SD)	2605.028 (1286.877)	2606.633 (1344.609)	2605.340 (1298.078)	
Range	36.000 - 7043.000	164.000 - 6270.000	36.000 - 7043.000	
death				0.322
Mean (SD)	0.422 (0.494)	0.445 (0.497)	0.427 (0.495)	
Range	0.000 - 1.000	0.000 - 1.000	0.000 - 1.000	

```
print(summary(tableby(~age+meno+size+grade+nodes+pgr+er+chemo+hormon+dtime+death,
                     rotterdam,numeric.simplify = TRUE, numeric.test = "kwt")))
```

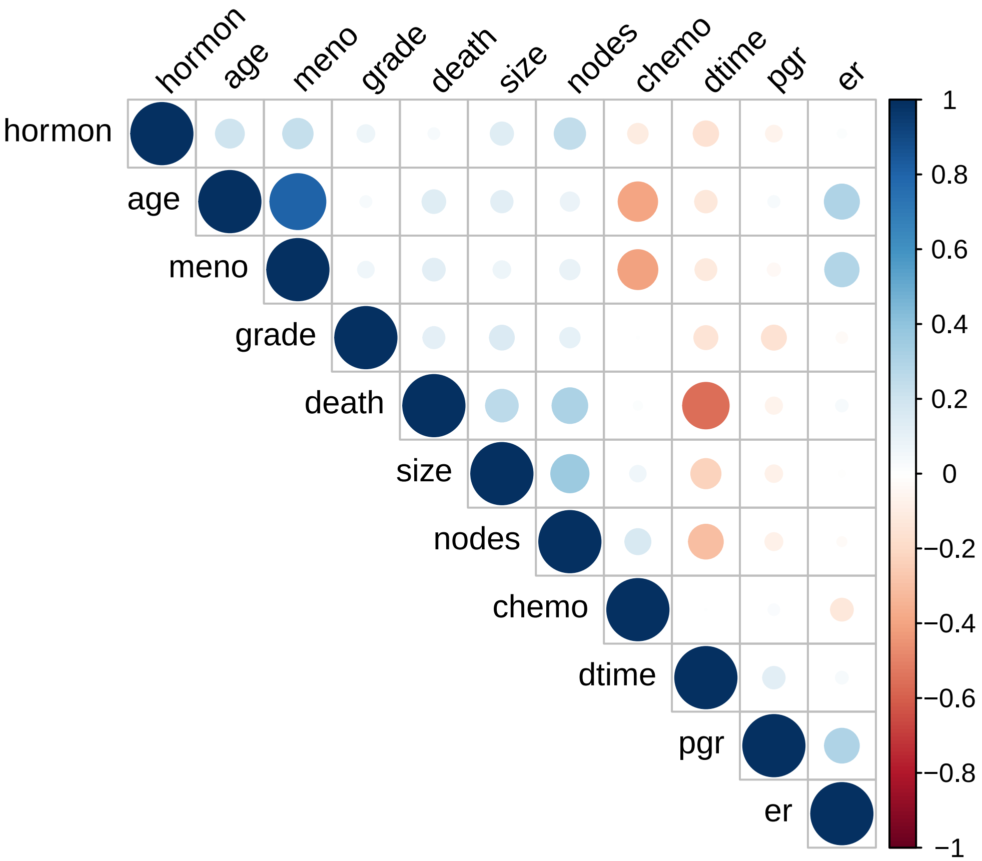
	Overall (N=2982)
age	
Mean (SD)	55.058 (12.953)
Range	24.000 - 90.000
meno	
Mean (SD)	0.560 (0.496)
Range	0.000 - 1.000
size	
<=20	1387 (46.5%)
20-50	1291 (43.3%)
>50	304 (10.2%)
grade	
Mean (SD)	2.734 (0.442)
Range	2.000 - 3.000
nodes	
Mean (SD)	2.712 (4.384)
Range	0.000 - 34.000
pgr	
Mean (SD)	161.831 (291.311)
Range	0.000 - 5004.000
er	
Mean (SD)	166.590 (272.465)
Range	0.000 - 3275.000
chemo	
0	2402 (80.5%)
1	580 (19.5%)
hormon	
0	2643 (88.6%)
1	339 (11.4%)
dtime	
Mean (SD)	2605.340 (1298.078)
Range	36.000 - 7043.000
death	

	Overall (N=2982)
Mean (SD)	0.427 (0.495)
Range	0.000 - 1.000

```r
my_data <- rotterdam[, c(3,4,5,6,7,8,9,10,11,14,15)] %>%
  mutate(size = as.numeric(size),
         chemo = as.numeric(chemo),
         hormon = as.numeric(hormon))

res <- cor(my_data)
round(res, 2)
```

```
##          age  meno  size grade nodes   pgr    er hormon chemo dtime death
## age     1.00  0.80  0.12  0.03  0.09  0.03  0.31   0.21 -0.39 -0.12  0.14
## meno    0.80  1.00  0.07  0.07  0.10 -0.04  0.30   0.23 -0.40 -0.12  0.13
## size    0.12  0.07  1.00  0.15  0.37 -0.07 -0.01   0.13  0.06 -0.23  0.27
## grade   0.03  0.07  0.15  1.00  0.10 -0.15 -0.03   0.07  0.00 -0.14  0.12
## nodes   0.09  0.10  0.37  0.10  1.00 -0.08 -0.02   0.25  0.17 -0.30  0.32
## pgr     0.03 -0.04 -0.07 -0.15 -0.08  1.00  0.30  -0.07  0.03  0.12 -0.07
## er      0.31  0.30 -0.01 -0.03 -0.02  0.30  1.00   0.02 -0.13  0.04  0.04
## hormon  0.21  0.23  0.13  0.07  0.25 -0.07  0.02   1.00 -0.10 -0.16  0.03
## chemo  -0.39 -0.40  0.06  0.00  0.17  0.03 -0.13  -0.10  1.00  0.00  0.02
## dtime  -0.12 -0.12 -0.23 -0.14 -0.30  0.12  0.04  -0.16  0.00  1.00 -0.55
## death   0.14  0.13  0.27  0.12  0.32 -0.07  0.04   0.03  0.02 -0.55  1.00
```

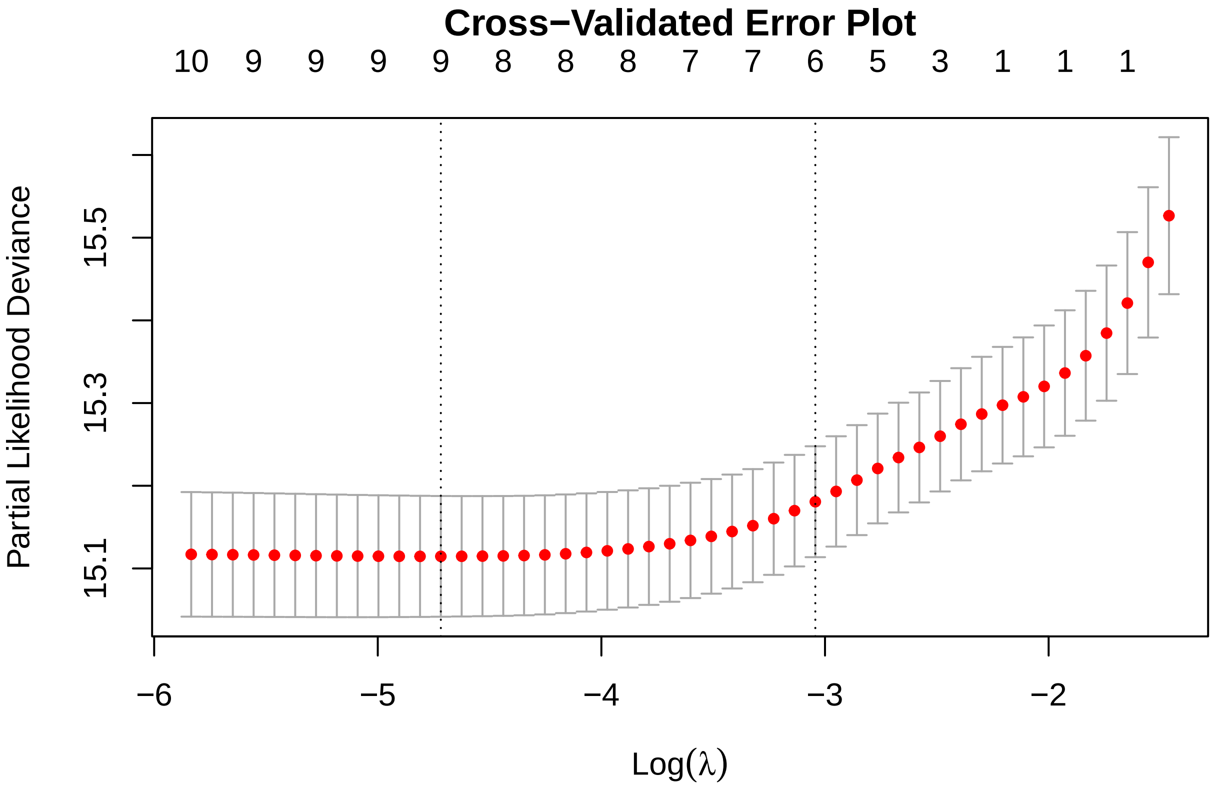
```r
corrplot(res, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```



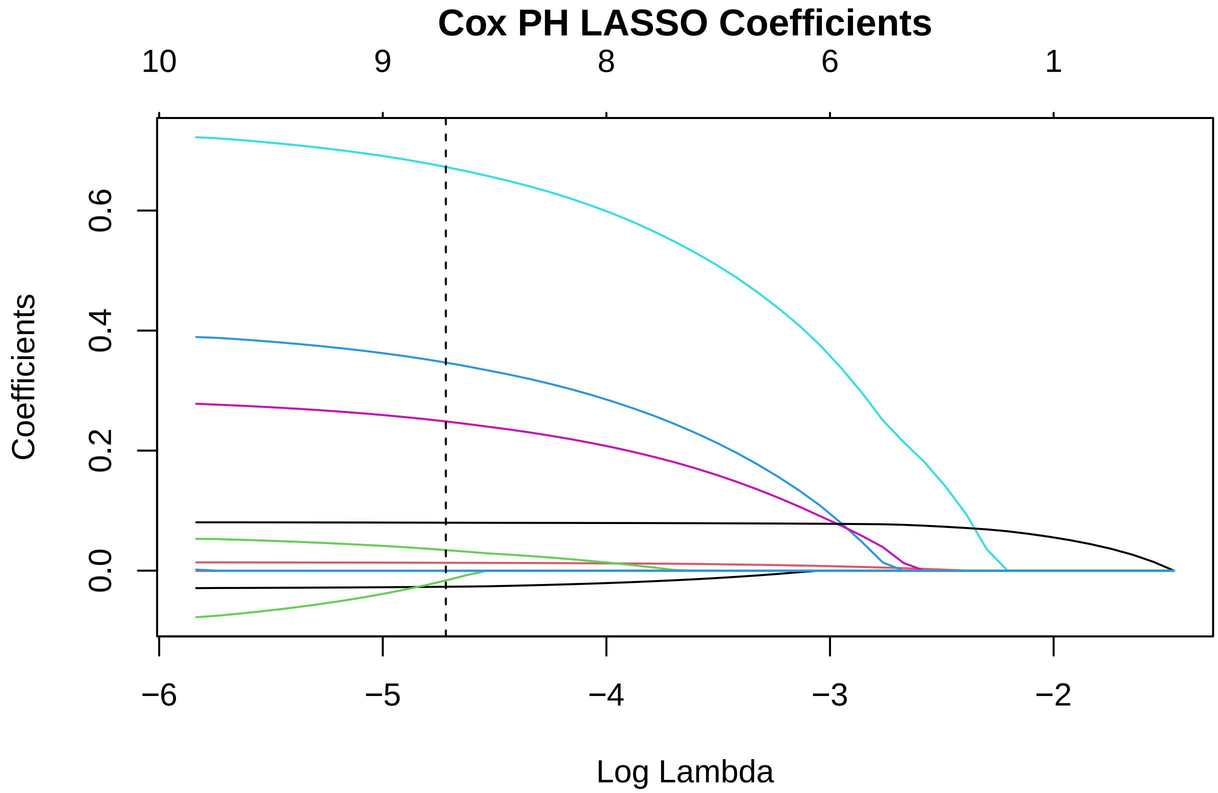
Cross-Validation

Cox with LASSO

```
set.seed(2022)

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

par(mar = c(4,4,5,1))
plot(coxnetfit, xvar = "lambda",
     main = "Cox PH LASSO Coefficients")
abline(v = log(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 LASSO Coefficients")
```

Table 4: Cox Proportion Hazard LASSO Coefficients

	coef	exp_coef
year	-0.0268	0.9735
age	0.0132	1.0132
meno	0.0343	1.0349
size20-50	0.3467	1.4144
size>50	0.6727	1.9596
grade	0.2487	1.2823
nodes	0.0799	1.0832
pgr	-0.0004	0.9996
er	0.0000	1.0000
hormon1	-0.0163	0.9838

	coef	exp_coef
chemo1	0.0000	1.0000

In the table above, we can see that estrogen receptors and chemotherapy are 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,
                data = rotterdam_training, ties = "breslow")
coxfit %>%
  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         0.970  0.0129      -2.34  1.92e- 2
## 2 age          1.01   0.00464      3.05  2.27e- 3
## 3 meno         1.07   0.122        0.517 6.05e- 1
## 4 size20-50    1.51   0.0825       4.99  6.16e- 7
## 5 size>50      2.11   0.115        6.51  7.56e-11
## 6 grade        1.34   0.0903       3.24  1.19e- 3
## 7 nodes        1.08   0.00633     12.8   2.11e-37
## 8 pgr          1.00   0.000147    -3.35  8.01e- 4
## 9 hormon1      0.897  0.117       -0.928 3.53e- 1
```

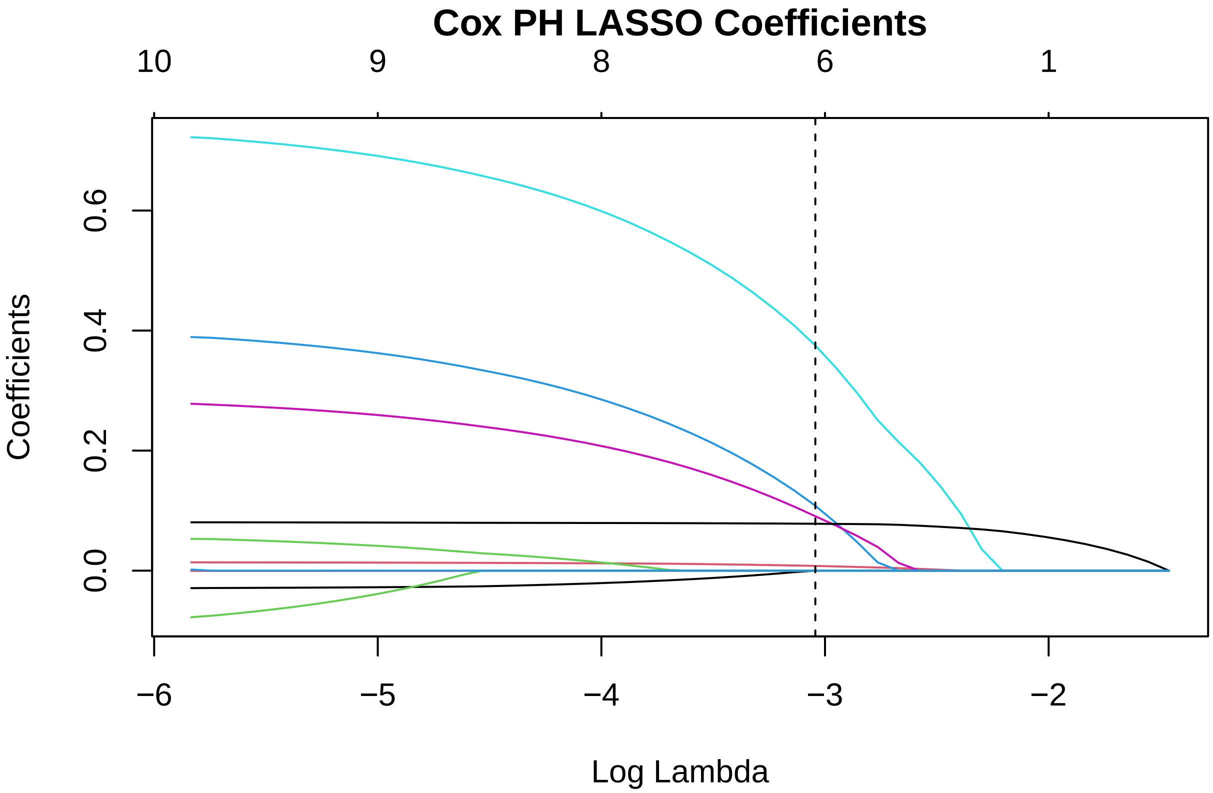
```
confint(coxfit) %>% exp() %>% knitr::kable()
```

	2.5 %	97.5 %
year	0.9462185	0.9951111
age	1.0050778	1.0235067
meno	0.8384713	1.3529343
size20-50	1.2837050	1.7740258
size>50	1.6860598	2.6446630
grade	1.1227352	1.5993840
nodes	1.0709185	1.0978328
pgr	0.9992211	0.9997958
hormon1	0.7133152	1.1282404

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.

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



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

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

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

Table 6: Cox Proportion Hazard LASSO Coefficients (1se)

	coef	exp_coef
year	0.0000	1.0000
age	0.0079	1.0079
meno	0.0000	1.0000
size20-50	0.1081	1.1142
size>50	0.3752	1.4553
grade	0.0907	1.0949
nodes	0.0781	1.0813
pgr	-0.0001	0.9999
er	0.0000	1.0000
hormon1	0.0000	1.0000

	coef	exp_coef
chemo1	0.0000	1.0000

Here, the regularization procedure removes `meno`, `er`, `hormon` and `chemo`. However, we are still interested in assessing the treatment effects of hormone therapy and chemotherapy, so we will add these back to the model.

```
# creating our final model
coxfit_1se <- coxph(Surv(dtime, death) ~ year + age + size + grade + nodes + pgr +
                      # adding our treatment variables
                      hormon + chemo,
                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         0.970  0.0128      -2.35  1.86e- 2
## 2 age          1.02   0.00324      5.06  4.17e- 7
## 3 size20-50    1.50   0.0824       4.94  7.76e- 7
## 4 size>50      2.09   0.114        6.47  1.01e-10
## 5 grade        1.34   0.0901       3.29  1.01e- 3
## 6 nodes        1.08   0.00642     12.6   2.07e-36
## 7 pgr          1.00   0.000147    -3.40  6.62e- 4
## 8 hormon1      0.904  0.117       -0.863 3.88e- 1
## 9 chemo1       1.02   0.102        0.225 8.22e- 1
```

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

	2.5 %	97.5 %
year	0.9461194	0.9949487
age	1.0100862	1.0229814
size20-50	1.2785538	1.7661763
size>50	1.6742487	2.6213387
grade	1.1269399	1.6040688
nodes	1.0707234	1.0980177
pgr	0.9992125	0.9997879
hormon1	0.7195108	1.1364530
chemo1	0.8375808	1.2499811

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. We find non-significant effects for each of the two treatments of interest.

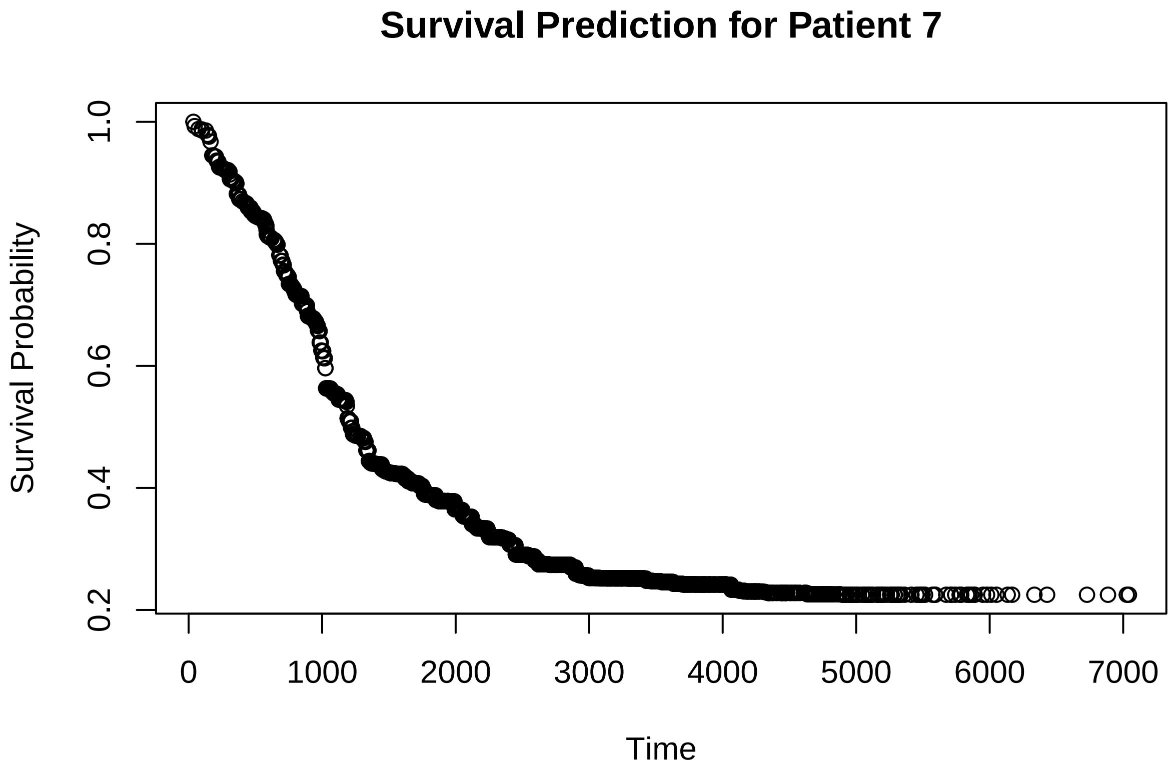
Random survival forest

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.

```
set.seed(2023)
## Random Survival Forest
rsf <- ranger(Surv(time = dtime, event = death) ~ .,
              data = rotterdam_training,
              num.trees = 300,
              min.node.size = 15,
              importance = "permutation",
              scale.permutation.importance = TRUE)

## Remove variables not for prediction, and the outcome
rotterdam_test_d <-
  rotterdam_test %>%
  select(-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	0.9929877
74	0.9881553
97	0.9877437
101	0.9857489
129	0.9857489

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



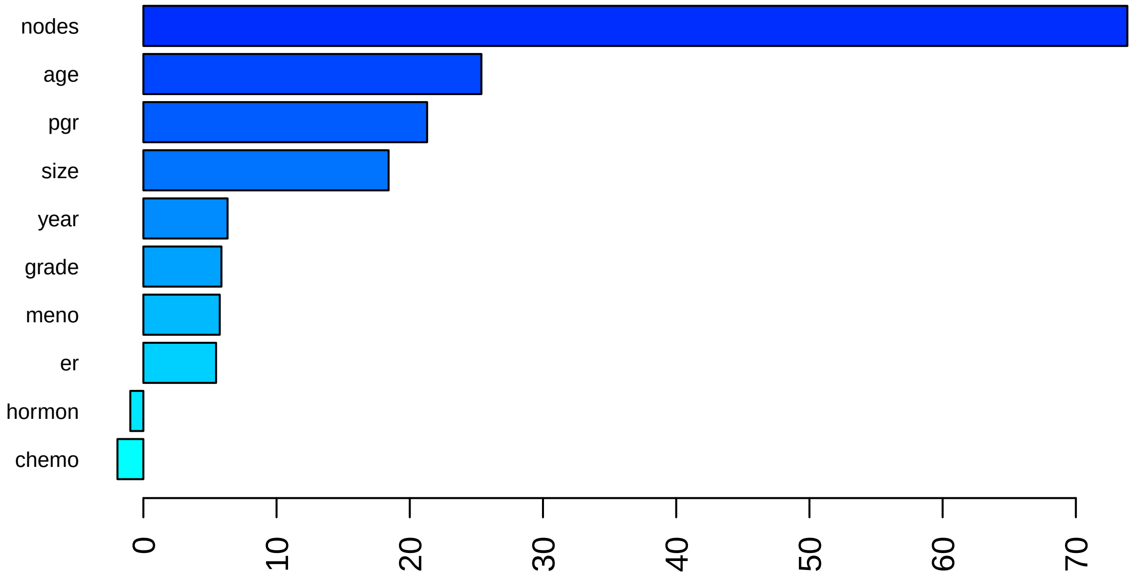
```
# Find estimated median survival time for individual 7
head(pred_ref_7[pred_ref_7$survival <= 0.5,]) %>% knitr::kable(align = "c") #1217
```

	time	survival
306	1217	0.4984613
307	1218	0.4984613
308	1222	0.4984613
309	1226	0.4984613
310	1229	0.4927672
311	1231	0.4927672

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

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

```
# Variable Importance
barplot(sort(ranger::importance(rsf), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan", "blue"))(12))
```



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 1217 days.

Comparison of Cox Proportional-Hazards Elastic Net with Random Survival Forest

We compared the Cox proportional-hazards elastic net model with the random survival forest by calculating the Brier score for each model on the validation set. The formula for the Brier score is as follows.

$$BS = \frac{1}{n}\sum_{i=1}^{n}(p_i - o_i)^2$$

The Brier score is used to evaluate the accuracy of probabilistic predictions from a model; its value ranges from 0 to 1 with 0 being perfect and 1 being the opposite. We calculated the Brier score using the validation set. For each observation in the validation set, predictions were made at the observed time of censoring or event. Our analysis proceeds as follows.

First, we calculated the Brier score for the Cox proportional-hazards elastic net model.

```r
# Purpose: Calculates the Brier score for the Cox proportional-hazards elastic
#          net model.
# Arguments: fit: The Cox proportional-hazards elastic net model.
#            train: A dataframe, the training data used to fit the model.
#            test: A dataframe, the data to use to calculate the Brier score.
# Returns: A double, the Brier score.
brier_coxnet <- function(fit, train, test) {
  train_x <- model.matrix(Surv(dtime, death) ~ ., train)[,-1]
  train_y <- Surv(pull(train, dtime), pull(train, death))
  test_x <- model.matrix(Surv(dtime, death) ~ ., test)[,-1]
  test_y <- pull(test, death)
  num_obs <- nrow(test_x)
  p <- vector(mode = "double", length = num_obs)
  for(i in 1:num_obs) {
    surv_fit <- survival::survfit(fit, s = cv_coxfit$lambda.min,
                                  x = train_x,
                                  y = train_y,
                                  newx = test_x[i, ])
    time_index <- tail(which(surv_fit$time <= test[i, "dtime"]), n = 1)
    p[i] <- 1 - surv_fit$surv[time_index]
  }
  return(DescTools::BrierScore(resp = test_y, pred = p))
}
(brier_coxnet <- round(brier_coxnet(coxnetfit, rotterdam_training, rotterdam_validation), 3))
```

```
## [1] 0.329
```

The Brier score for the Cox elastic net model is 0.329.

Second, let's calculated the Brier score for the random survival forest model.

```r
# Purpose: Calculates the Brier score for the random survival forest model.
# Arguments: fit: The random survival forest model.
#            df: A dataframe, the data to use to calculate the Brier score.
# Returns: A double, the Brier score.
brier_ranger <- function(fit, df) {
  x <- df
  pred <-  predict(fit, data = x)
  num_obs <- nrow(df)
  p <- vector(mode = "double", length = num_obs)
  for(i in 1:num_obs) {
    time_index <- tail(which(pred$unique.death.times <= x[i, "dtime"]), n = 1)
    p[i] <- 1 - pred$survival[i, time_index]
  }
  return(DescTools::BrierScore(resp = df$death, pred = p))
}
(brier_ranger <- round(brier_ranger(rsf, rotterdam_validation), 3))
```

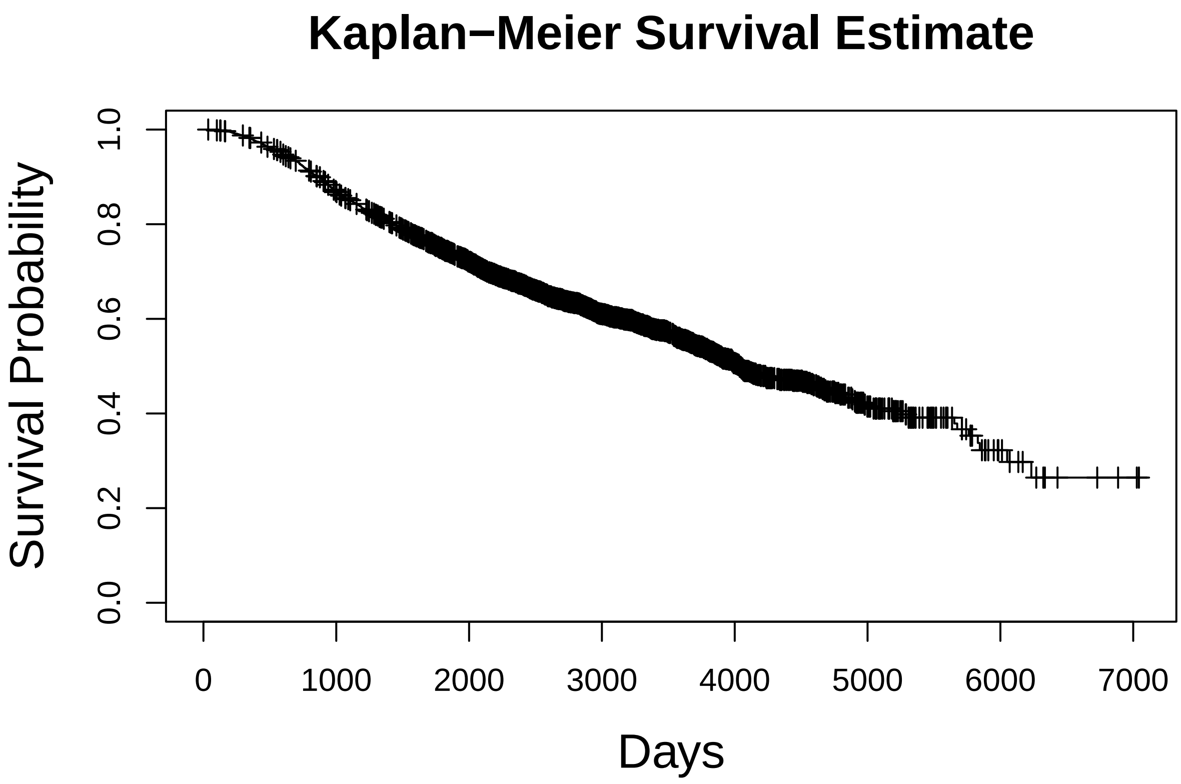
```
## [1] 0.322
```

The Brier score for the random survival forest model is 0.322. The two models have very similar Brier scores.

Supplemental analyses

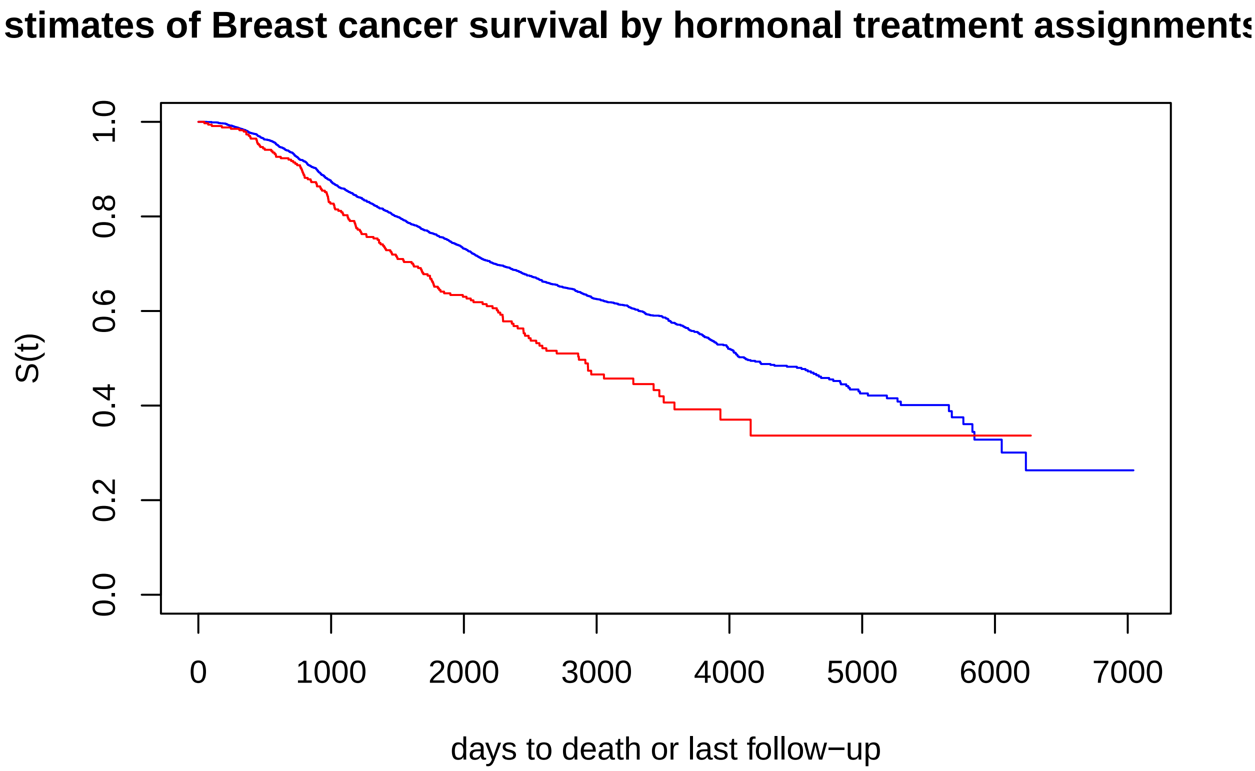
Kaplan-Meier Survival Estimate

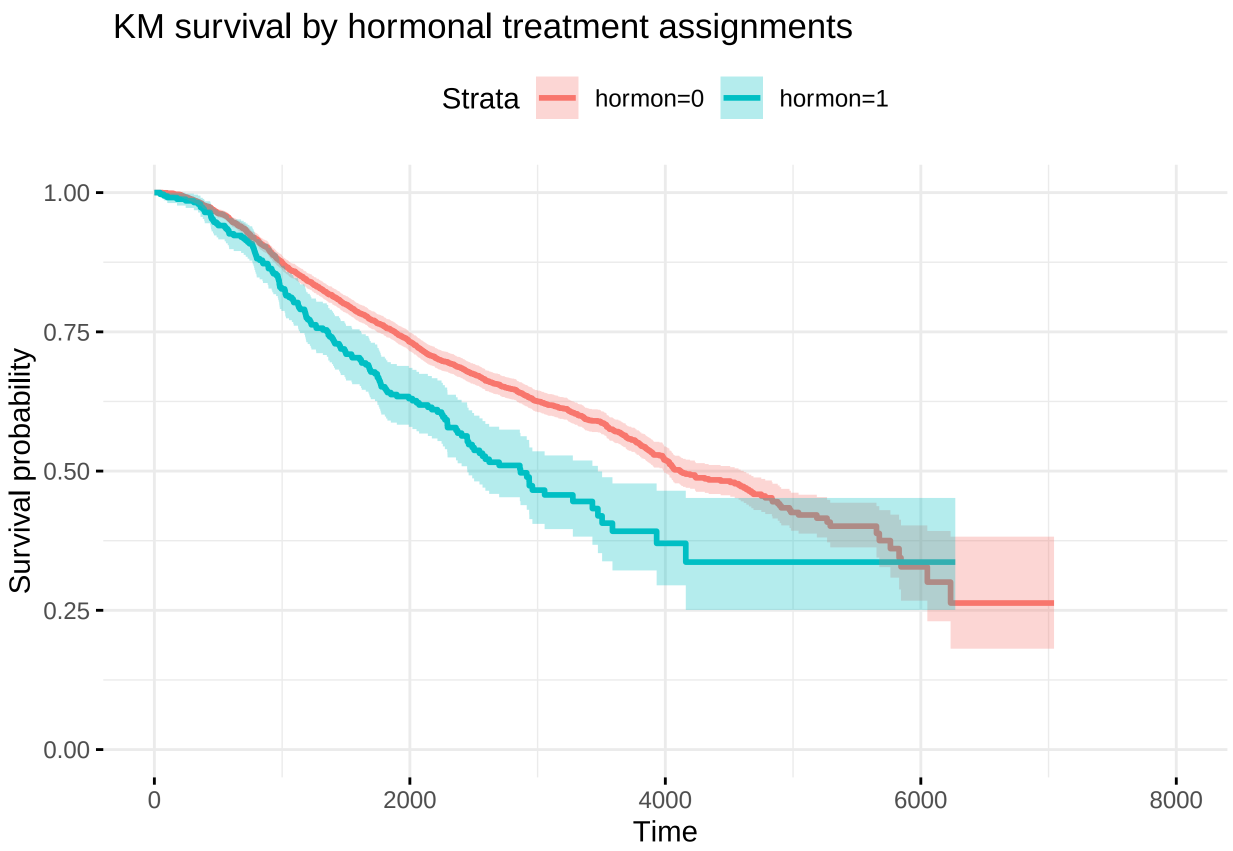
```
KM = survfit(Surv(dtime, death) ~ 1, data = rotterdam)
plot(KM, conf.int = FALSE, mark.time = TRUE,
     xlab = "Days", ylab = "Survival Probability",
     main = "Kaplan-Meier Survival Estimate", cex.lab = 1.5, cex.main = 1.5)
```



```
# 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 rotterda
col = c("blue","red"))

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


stimates of Breast cancer survival by hormonal treatment assignments
S(t)
0.0
0.2
0.4
0.6
0.8
1.0
0
1000
2000
3000
4000
5000
6000
7000
days to death or last follow−up



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.

Combined

```
# Add 1
rotterdam1 <-
  rotterdam %>%
  mutate(
    trt_label = case_when(
      hormon == 1 & chemo == 1 ~ "hormon+chemo",
      hormon == 1 & chemo == 0 ~ "hormon",
      hormon == 0 & chemo == 1 ~ "chemo",
      hormon == 0 & chemo == 0 ~ "none"
    )
  )

table(rotterdam1$trt_label)
```

```
##
##        chemo       hormon hormon+chemo         none
##          552          311           28         2091
```

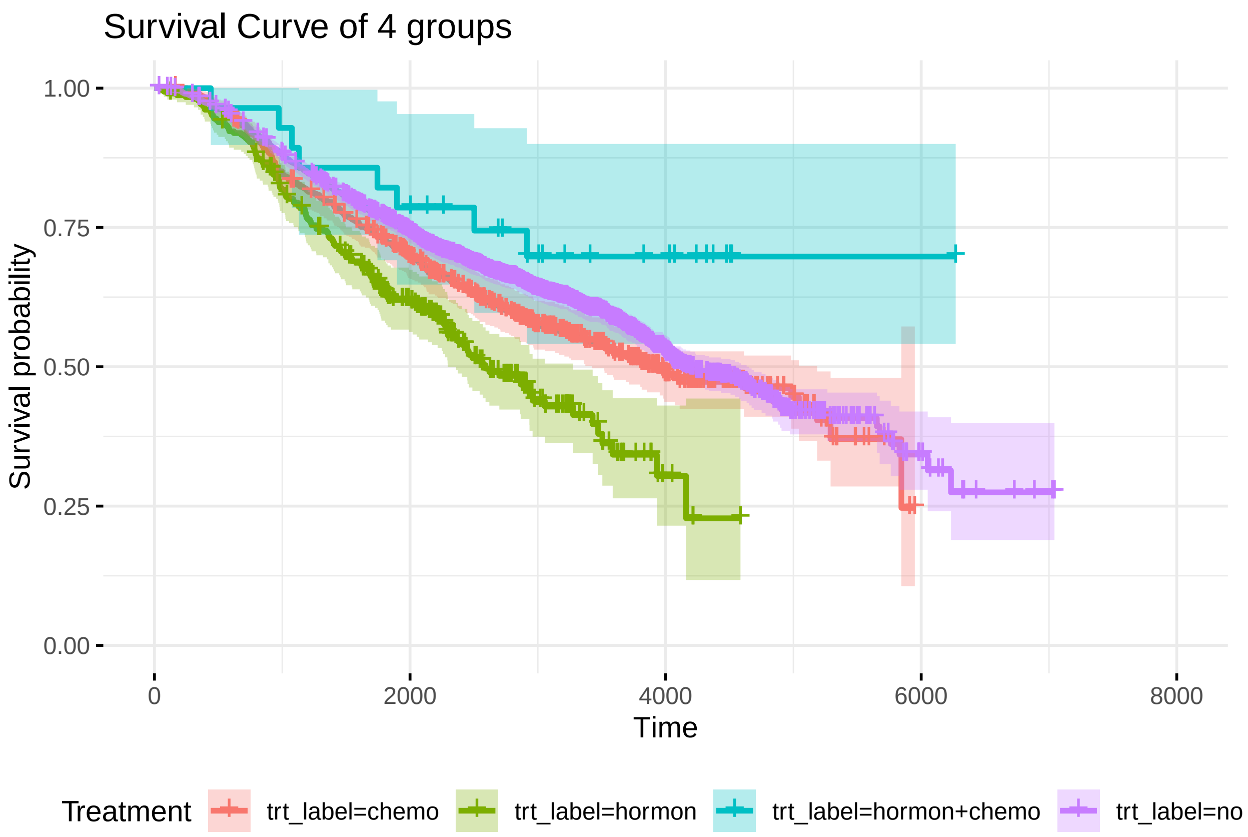
```
# Add 2
logrank1 <- survdiff(Surv(dtime, death) ~ trt_label, data = rotterdam1)
logrank1
```

```
## Call:
## survdiff(formula = Surv(dtime, death) ~ trt_label, data = rotterdam1)
##
##                            N Observed Expected (O-E)^2/E (O-E)^2/V
## trt_label=chemo          552      250    233.7      1.13      1.39
## trt_label=hormon         311      151     96.0     31.45     34.43
## trt_label=hormon+chemo    28        8     14.3      2.79      2.83
## trt_label=none          2091      863    927.9      4.54     16.84
##
##  Chisq= 40.4  on 3 degrees of freedom, p= 9e-09
```

```
logrank1$pvalue
```

```
## [1] 8.838693e-09
```

```
# Add 3
ggsurvplot(survfit(Surv(dtime,death) ~ trt_label, data = rotterdam1),
           conf.int = TRUE,
           legend = c("bottom"),
           legend.title = c("Treatment"),
           ggtheme = theme_minimal()) +
  ggtitle("Survival Curve of 4 groups")
```



Hormon

```
logrank2 <- survdiff(Surv(dtime, death) ~ hormon, data = rotterdam)
logrank2
```

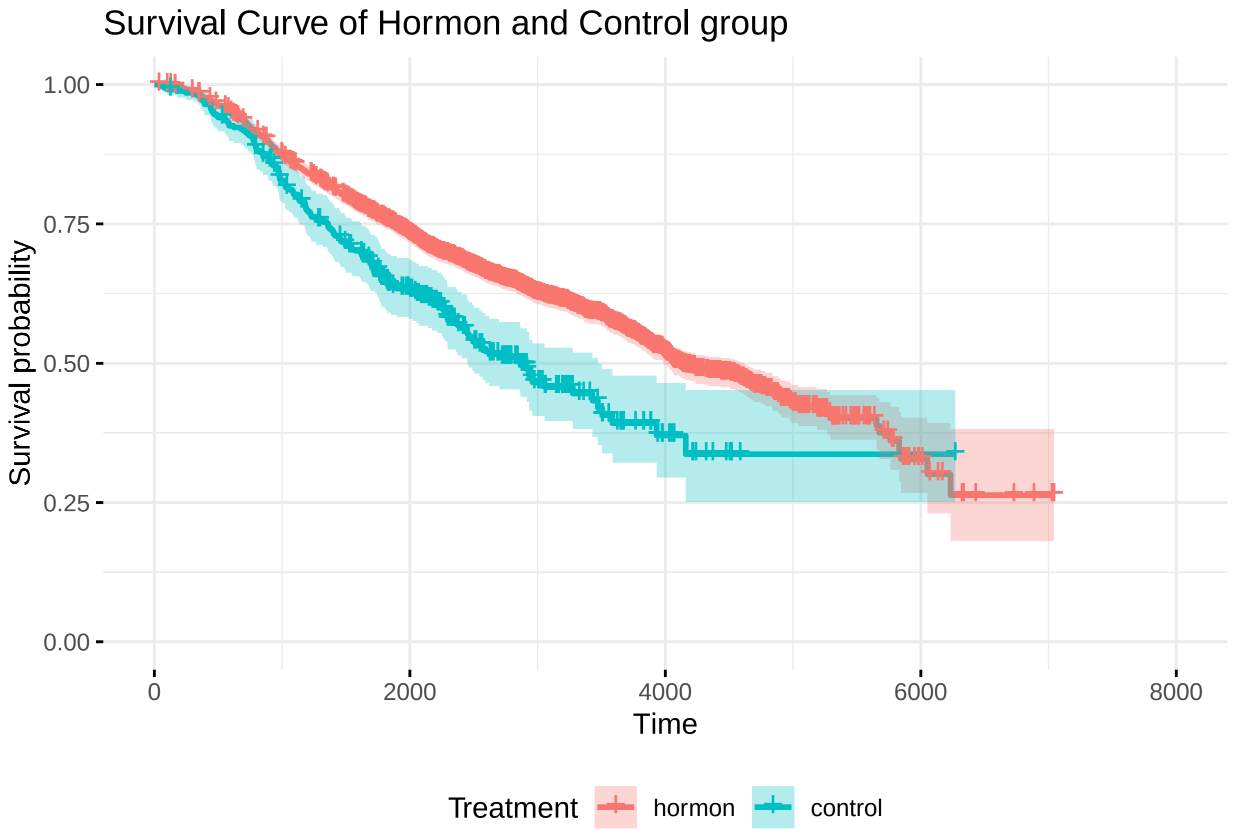
```
## Call:
## survdiff(formula = Surv(dtime, death) ~ hormon, data = rotterdam)
##
##             N Observed Expected (O-E)^2/E (O-E)^2/V
## hormon=0 2643     1113     1162      2.04      23.7
## hormon=1  339      159      110     21.43      23.7
##
##  Chisq= 23.7  on 1 degrees of freedom, p= 1e-06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

```
logrank2$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"),
           ggtheme = theme_minimal()) +
  ggtitle("Survival Curve of Hormon and Control group")
```



Chemo

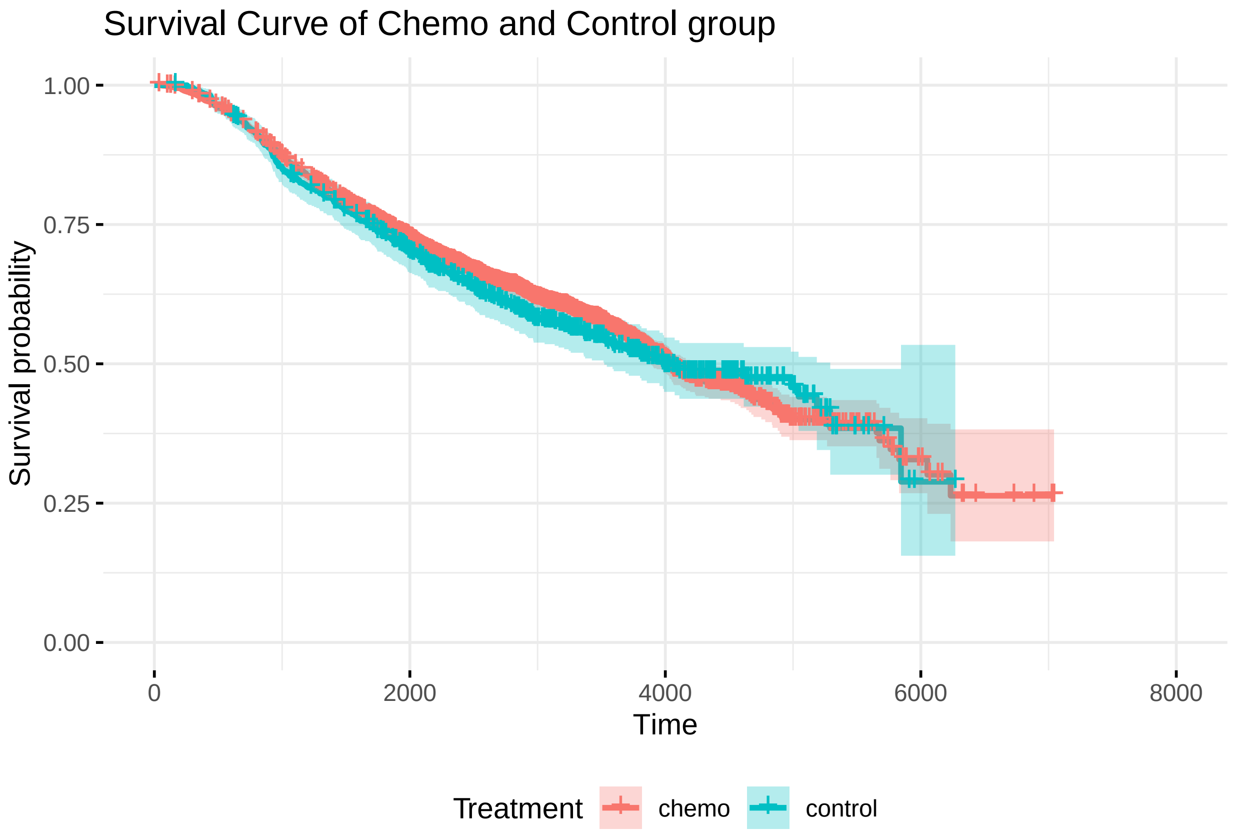
```
logrank3 <- survdiff(Surv(dtime, death) ~ chemo, data = rotterdam)
logrank3
```

```
## Call:
## survdiff(formula = Surv(dtime, death) ~ chemo, data = rotterdam)
##
##            N Observed Expected (O-E)^2/E (O-E)^2/V
## chemo=0 2402     1014     1024    0.0963     0.495
## chemo=1  580      258      248    0.3977     0.495
##
##  Chisq= 0.5  on 1 degrees of freedom, p= 0.5
```

```
logrank3$pvalue
```

```
## [1] 0.4818191
```

```
ggsurvplot(survfit(Surv(dtime,death) ~ chemo, data = rotterdam),
           conf.int = TRUE,
           legend = c("bottom"),
           legend.title = c("Treatment"),
           legend.labs = c("chemo", "control"),
           ggtheme = theme_minimal()) +
  ggtitle("Survival Curve of Chemo 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