

RF Predictions

2025-11-13

Fitting Random Forest Models

On Data simulated from Cox

Survival

Simulating data, fitting model, predicting, 10 times.

```
set.seed(375)
beta = matrix(c(0, 0, 0.5, -0.5), ncol = 2, nrow = 2)
newdata <- data.frame(L0 = c(0.5, 0.5), A0 = c(1, 0))
y.pred <- list()

for (i in 1:10){
  data <- simSurvData(N = 2000, beta = beta)
  obj <- rfsrc(Surv(Time, Delta == 1) ~ L0 + A0, data = data, nodesize = 10)

  y.pred[[i]] <- predict(obj, newdata = newdata)
}
```

True survival function:

```
true_surv_death_func1 <- function(t) exp(-0.1 * t^(1.1) * exp(sum(newdata[1,] * c(0.5, -0.5))))
true_surv_death_func2 <- function(t) exp(-0.1 * t^(1.1) * exp(sum(newdata[2,] * c(0.5, -0.5))))
# Evaluate the true survival functions at these time points
time_points <- round(y.pred[[1]]$time.interest, 2)
true_surv1 <- true_surv_death_func1(time_points)
true_surv2 <- true_surv_death_func2(time_points)
```

Visualizing fit

```
par(mfrow = c(1,2), cex.axis = 2.0, cex.lab = 2.0, cex.main = 2.0, mar = c(6,6,1,1), mgp = c(4,1,0))

# Initialize plot for process 1
plot(NULL, xlim = range(y.pred[[1]]$time.interest), ylim = c(0,1),
      xlab = "Time (Year)", ylab = "Survival")

# Loop over all elements in y.pred
for (i in seq_along(y.pred)) {
  lines(y.pred[[i]]$time.interest, y.pred[[i]]$survival[,],
        col = rgb(0,0,1,alpha=0.3)) # blue semi-transparent
}
```

```

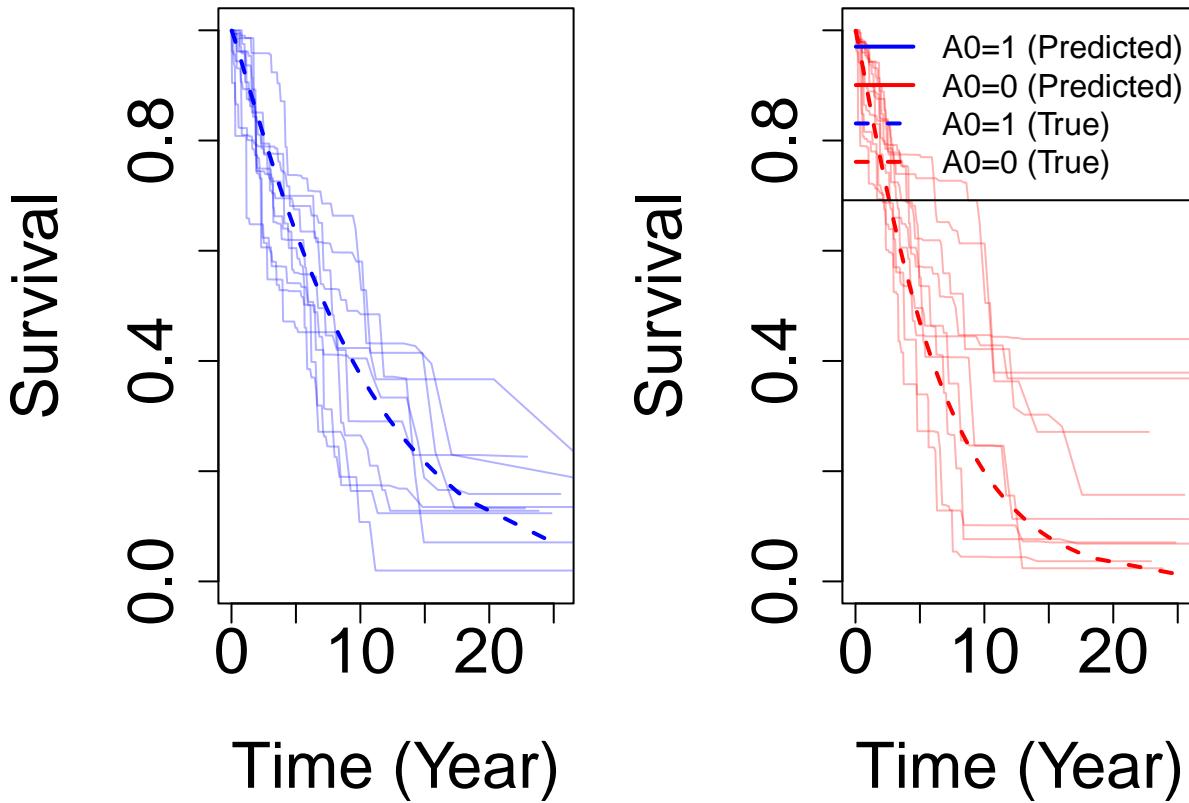
lines(time_points, true_surv1, col="blue", lty=2, lwd=2)

# Initialize plot for process 2
plot(NULL, xlim = range(y.pred[[1]]$time.interest), ylim = c(0,1),
     xlab = "Time (Year)", ylab = "Survival")

# Loop over all elements in y.pred
for (i in seq_along(y.pred)) {
  lines(y.pred[[i]]$time.interest, y.pred[[i]]$survival[2,],
        col = rgb(1,0,0,alpha=0.3)) # red semi-transparent
}
lines(time_points, true_surv2, col="red", lty=2, lwd=2)

legend("topright",
       legend=c("A0=1 (Predicted)", "A0=0 (Predicted)", "A0=1 (True)", "A0=0 (True)"),
       col=c("blue","red","blue","red"),
       lty=c(1,1,2,2), lwd=2, cex=1)

```



Competing Risk

Simulate data, fit RSF model, create predictions

```

set.seed(373)
beta = matrix(c(0.5,-1,-0.5,0.5,0,0.5), ncol = 3, nrow = 2)

```

```

newdata <- data.frame(L0 = c(0.5,0.5), A0 = c(1,0))
y.pred <- list()
for(i in 1:10){
  data <- simCRdata(N = 3000, beta = beta)

  #tune.nodesize(Surv(Time,Delta) ~ L0 + A0, data = data)
  RF <- rfsrc(Surv(Time, Delta) ~ L0 + A0, data = data, nodesize = 150)

  y.pred[[i]] <- predict(RF, newdata = newdata)
}

```

True CHF

```

true_chf_func1 <- function(t, cov) 0.1 * t^(1.1) * exp(sum(cov * c(-0.5,0.5)))
true_chf_func2 <- function(t, cov) 0.1 * t^(1.1) * exp(sum(cov * c(0,0.5)))

# Evaluate the true survival functions at time points
time_points <- y.pred[[length(y.pred)]]$time.interest
true_chf11 <- true_chf_func1(time_points, newdata[1,])
true_chf12 <- true_chf_func1(time_points, newdata[2,])
true_chf21 <- true_chf_func2(time_points, newdata[1,])
true_chf22 <- true_chf_func2(time_points, newdata[2,])

```

Plot the true CHF together with the 10 estimated CHF's

```

par(mfrow = c(2, 2), mar = c(2, 2, 2, 1), cex.axis = 1,
    cex.lab = 1, cex.main = 1, mgp = c(2.5, 0.8, 0)) #oma = c(5, 5, 2, 1)

# ---- Process 1, A0 = 1 ----
plot(time_points, true_chf11, type="l",
      xlab="Time", ylab="CHF",
      col=1, lty=1, lwd=2, xlim=c(0,10), ylim=c(0, 2.5))
for (i in seq_along(y.pred)) {
  lines(y.pred[[i]]$time.interest, y.pred[[i]]$chf[,1],
        col=adjustcolor(1, alpha.f=0.3), lty=2, lwd=2)
}
title(main = "Process 1, A0 = 1")

# ---- Process 1, A0 = 0 ----
plot(time_points, true_chf12, type="l",
      xlab="Time", ylab="CHF",
      col=2, lty=1, lwd=2, xlim=c(0,10), ylim=c(0, 1))
for (i in seq_along(y.pred)) {
  lines(y.pred[[i]]$time.interest, y.pred[[i]]$chf[,1],
        col=adjustcolor(2, alpha.f=0.3), lty=2, lwd=2)
}
title(main = "Process 1, A0 = 0")

# ---- Process 2, A0 = 1 ----
plot(time_points, true_chf21, type="l",
      xlab="Time", ylab="CHF",
      col=3, lty=1, lwd=2, xlim=c(0,10), ylim=c(0, 2.5))
for (i in seq_along(y.pred)) {

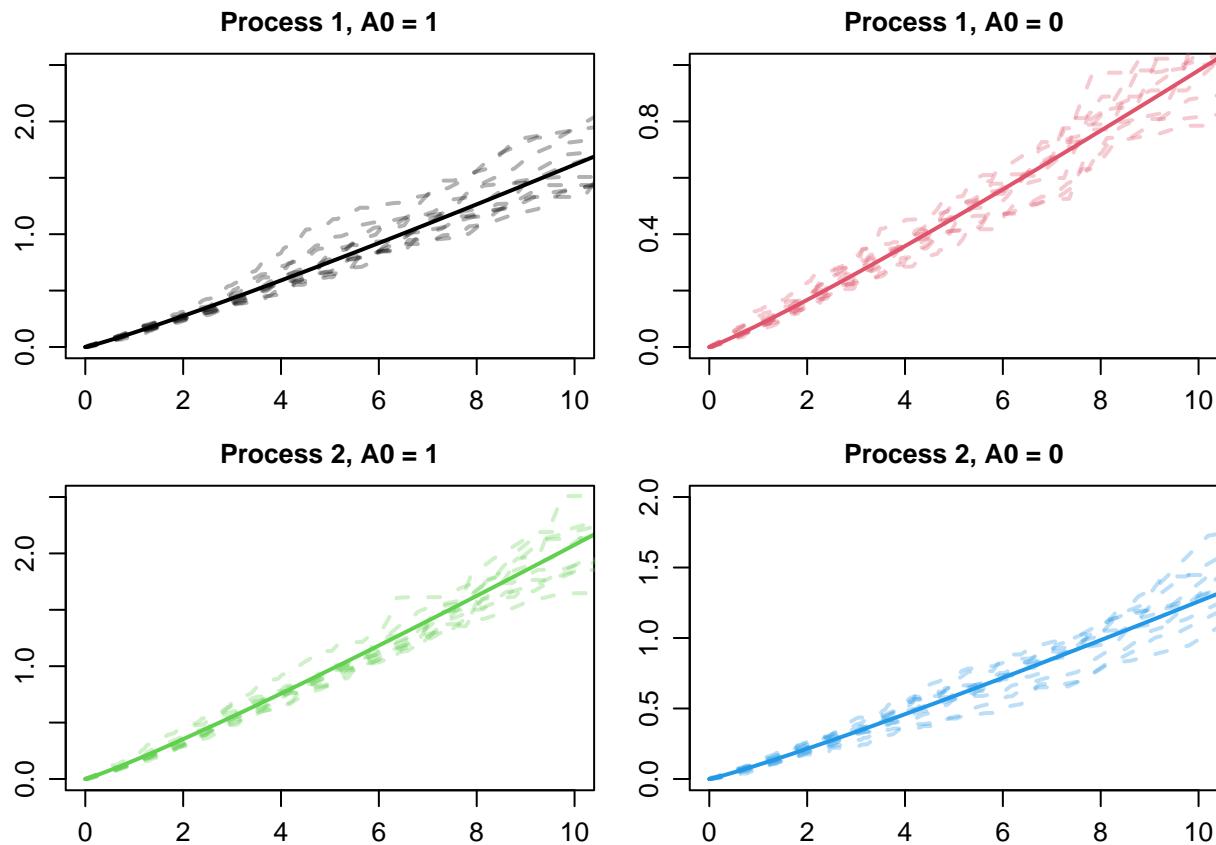
```

```

    lines(y.pred[[i]]$time.interest, y.pred[[i]]$chf[1,,2],
          col=adjustcolor(3, alpha.f=0.3), lty=2, lwd=2)
}
title(main = "Process 2, A0 = 1")

# ----- Process 2, A0 = 0 -----
plot(time_points, true_chf22, type="l",
      xlab="Time", ylab="CHF",
      col=4, lty=1, lwd=2, xlim=c(0,10), ylim=c(0, 2))
for (i in seq_along(y.pred)) {
  lines(y.pred[[i]]$time.interest, y.pred[[i]]$chf[2,,2],
        col=adjustcolor(4, alpha.f=0.3), lty=2, lwd=2)
}
title(main = "Process 2, A0 = 0")

```



Fit Cox models

```

cox1 <- coxph(Surv(Time, Delta == 1) ~ L0 + A0, data = data)
cox2 <- coxph(Surv(Time, Delta == 2) ~ L0 + A0, data = data)
basehazz1 <- basehaz(cox1, centered = TRUE)
basehazz2 <- basehaz(cox2, centered = TRUE)
cox_term1 <- exp(predict(cox1, newdata=newdata, type="lp"))
cox_term2 <- exp(predict(cox2, newdata=newdata, type="lp"))
cum_int1 <- outer(basehazz1[['hazard']], cox_term1, "*")
cum_int2 <- outer(basehazz2[['hazard']], cox_term2, "*")

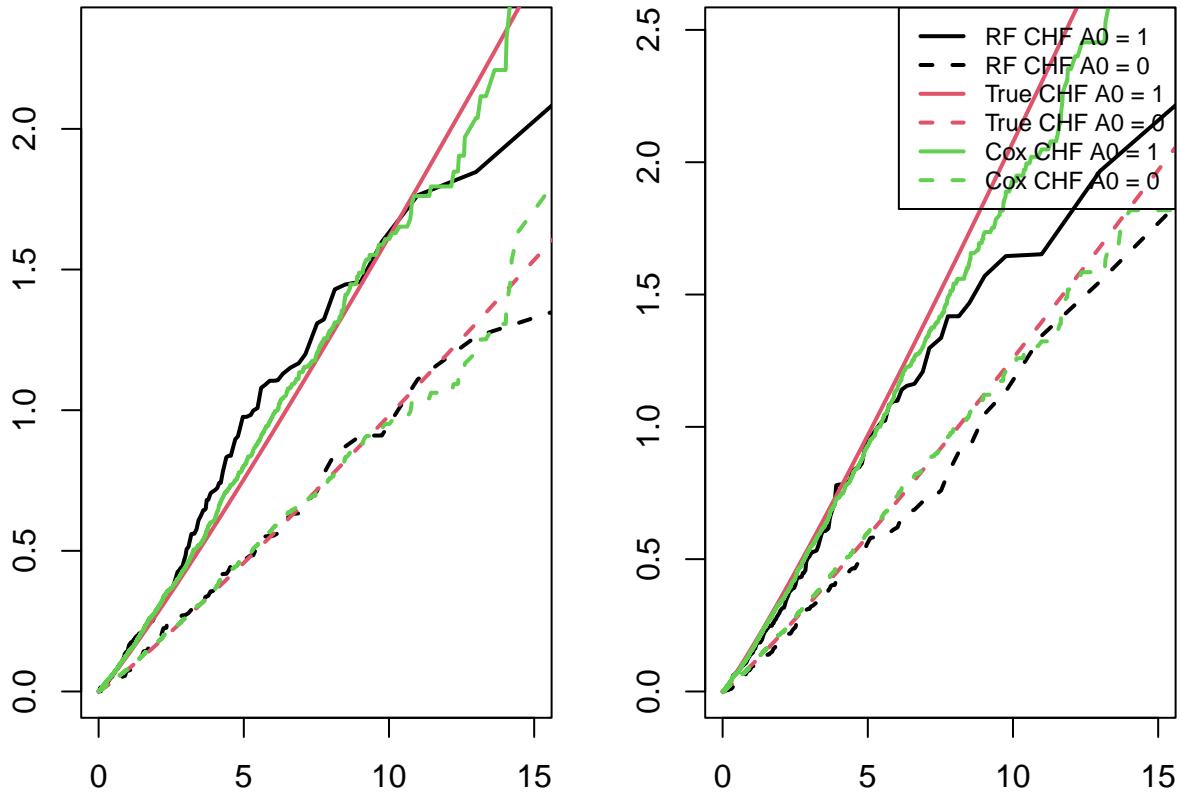
```

Plot a single fit with Cox models:

```
par(mfrow = c(1,2), cex.axis = 1.0, cex.lab = 1.0, cex.main = 2.0, mar = c(3.0,3,1,1), mgp = c(4, 1, 0))
# Process 1
plot(time_points,y.pred[[length(y.pred)]]$chf[1,,1], type="l", xlab="Time",
      ylab="Process 1 CHF", col=1, lty=1, lwd=2, xlim = c(0,15))
lines(time_points, y.pred[[length(y.pred)]]$chf[2,,1], col=1, lty=2, lwd=2)
# Add lines for the true survival functions
lines(time_points, true_chf11, col=2, lty=1, lwd=2)
lines(time_points, true_chf12, col=2, lty=2, lwd=2)
lines(basehazz1$time, cum_int1[,1], col=3, lty=1, lwd=2)
lines(basehazz2$time, cum_int1[,2], col=3, lty=2, lwd=2)

# Process 2
plot(time_points,y.pred[[length(y.pred)]]$chf[1,,2], type="l", xlab="Time",
      ylab="Process 2 CHF", col=1, lty=1, lwd=2, xlim = c(0,15))
lines(time_points, y.pred[[length(y.pred)]]$chf[2,,2], col=1, lty=2, lwd=2)
# Add lines for the true survival functions
lines(time_points, true_chf21, col=2, lty=1, lwd=2)
lines(time_points, true_chf22, col=2, lty=2, lwd=2)
lines(basehazz2$time, cum_int2[,1], col=3, lty=1, lwd=2)
lines(basehazz2$time, cum_int2[,2], col=3, lty=2, lwd=2)

legend("topright",
       legend = c("RF CHF A0 = 1", "RF CHF A0 = 0", "True CHF A0 = 1", "True CHF A0 = 0",
                 "Cox CHF A0 = 1", "Cox CHF A0 = 0"),
       col = c(1, 1, 2, 2, 3, 3),
       lty = c(1, 2, 1, 2, 1, 2),
       lwd = 2,
       cex = 0.75)
```



On Data not from Cox

Survival

```

beta <- matrix(c(0, 0, 0, 0, -0.5, 1, 0, 0), ncol = 2)
beta_TV <- matrix(c(0, 0, 0, 0, 0.5, -2, 0, 0), ncol = 2)
t_prime <- 2
data <- simEventTV(10^4, beta = beta, term_deltas = c(0,1), tv_eff = beta_TV, t_prime = t_prime)
#Fit model
obj <- rfsrc(Surv(Time, Delta) ~ L0 + A0, data = data, nodesize = 10)
# Predictions
newdata <- data.frame(L0 = c(0.5,0.5), A0 = c(1,0))
y.pred <- predict(obj, newdata = newdata)
# True survival function:
true_surv_death_func1 <- function(t) {
  if(t <= t_prime) return(exp(- 0.1 * t^(1.1) * exp(sum(newdata[1,] * c(-0.5,1)))) +
    exp(- (0.1 * t_prime^(1.1) * exp(sum(newdata[1,] * c(-0.5,1)))) +
      0.1 * t^(1.1) * exp(sum(newdata[1,] * c(0,-1))) -
      0.1 * t_prime^(1.1) * exp(sum(newdata[1,] * c(0,-1)))))

}
true_surv_death_func2 <- function(t) {
  if(t <= t_prime) return(exp(- 0.1 * t^(1.1) * exp(sum(newdata[2,] * c(-0.5,1)))) +
    exp(- (0.1 * t_prime^(1.1) * exp(sum(newdata[2,] * c(-0.5,1)))) +
      0.1 * t^(1.1) * exp(sum(newdata[2,] * c(0,-1))) -
      0.1 * t_prime^(1.1) * exp(sum(newdata[2,] * c(0,-1)))))

}

```

```

        0.1 * t_prime^(1.1) * exp(sum(newdata[2,] * c(0,-1))))
}
# Evaluate the true survival functions at these time points
time_points <- y.pred$time.interest
true_surv1 <- sapply(time_points, FUN = true_surv_death_func1)
true_surv2 <- sapply(time_points, FUN = true_surv_death_func2)

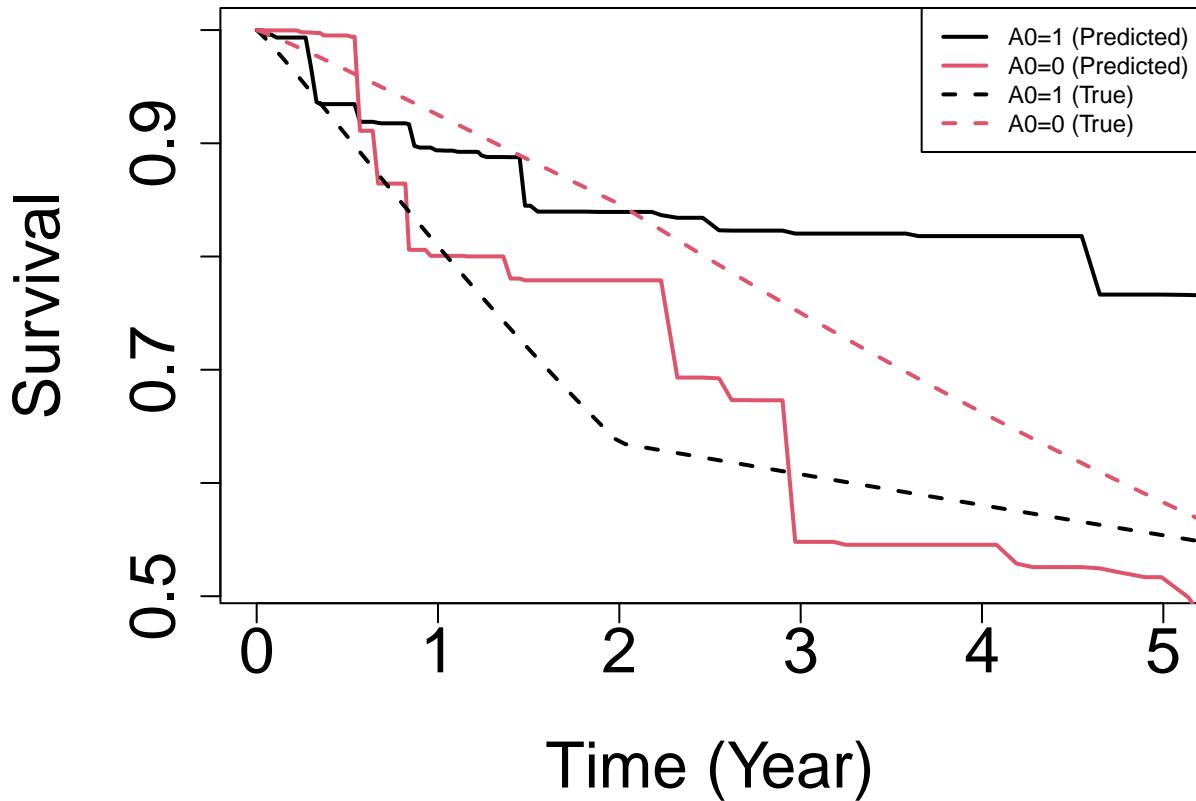
```

Visualizing fit

```

par(mfrow = c(1,1), cex.axis = 2.0, cex.lab = 2.0, cex.main = 2.0, mar = c(6.0,6,1,1), mgp = c(4, 1, 0))
plot(round(y.pred$time.interest,2), y.pred$survival[,], type="l", xlab="Time (Year)",
      ylab="Survival", col=1, lty=1, lwd=2, xlim = c(0,5))
lines(round(y.pred$time.interest,2), y.pred$survival[,], col=2, lty=1, lwd=2)
# Add lines for the true survival functions
lines(time_points, true_surv1, col=1, lty=2, lwd=2)
lines(time_points, true_surv2, col=2, lty=2, lwd=2)
legend("topright",
       legend=c("A0=1 (Predicted)", "A0=0 (Predicted)", "A0=1 (True)", "A0=0 (True)"),
       col=c(1, 2, 1, 2),
       lty=c(1, 1, 2, 2),
       cex=0.75,
       lwd=2)

```



Competing Risk

```
beta <- matrix(c(0, 0, 0, 0, 0, -0.5, 0.5, 0, 0, 0, 0.5, -0.5, 0, 0, 0), ncol = 3)
beta_TV <- matrix(c(0, 0, 0, 0, 0, 1, -1.5, 0, 0, 0, -1, 1, 0, 0, 0), ncol = 3)
t_prime <- 2
data <- simEventTV(3000, beta = beta, term_deltas = c(0,1,2), tv_eff = beta_TV, t_prime = t_prime)
```

Fit RSF model

```
#tune.nodesize(Surv(Time,Delta) ~ L0 + A0, data = data)
RF <- rfsrc(Surv(Time, Delta) ~ L0 + A0, data = data, nodesize = 150)
```

New Data

```
newdata <- data.frame(L0 = c(0.5,0.5), A0 = c(1,0))
y.pred <- predict(RF, newdata = newdata)
```

True CHF

```
true_chf_func1 <- function(t, cov) {
  if(t <= t_prime) return(0.1 * t^(1.1) * exp(sum(cov * c(-0.5,0.5))))
  0.1 * t_prime^(1.1) * exp(sum(cov * c(-0.5,0.5))) +
  0.1 * t^(1.1) * exp(sum(cov * c(0.5,-1))) -
  0.1 * t_prime^(1.1) * exp(sum(cov * c(0.5,-1)))
}

true_chf_func2 <- function(t, cov) {
  if(t <= t_prime) return(0.1 * t^(1.1) * exp(sum(cov * c(0.5,-0.5))))
  0.1 * t_prime^(1.1) * exp(sum(cov * c(0.5,-0.5))) +
  0.1 * t^(1.1) * exp(sum(cov * c(-0.5,0.5))) -
  0.1 * t_prime^(1.1) * exp(sum(cov * c(-0.5,0.5)))
}

# Evaluate the true CIF functions at time points
time_points <- y.pred$time.interest
true_chf11 <- sapply(time_points, FUN = function(t) true_chf_func1(t, newdata[1,]))
true_chf12 <- sapply(time_points, FUN = function(t) true_chf_func1(t, newdata[2,]))
true_chf21 <- sapply(time_points, FUN = function(t) true_chf_func2(t, newdata[1,]))
true_chf22 <- sapply(time_points, FUN = function(t) true_chf_func2(t, newdata[2,]))
```

Fit Cox models

```
cox1 <- coxph(Surv(Time, Delta == 1) ~ L0 + A0, data = data)
cox2 <- coxph(Surv(Time, Delta == 2) ~ L0 + A0, data = data)
basehazz1 <- basehaz(cox1, centered = TRUE)
basehazz2 <- basehaz(cox2, centered = TRUE)
cox_term1 <- exp(predict(cox1, newdata=newdata, type="lp"))
cox_term2 <- exp(predict(cox2, newdata=newdata, type="lp"))
cum_int1 <- outer(basehazz1[['hazard']], cox_term1, "*")
cum_int2 <- outer(basehazz2[['hazard']], cox_term2, "*")
```

Plot:

```

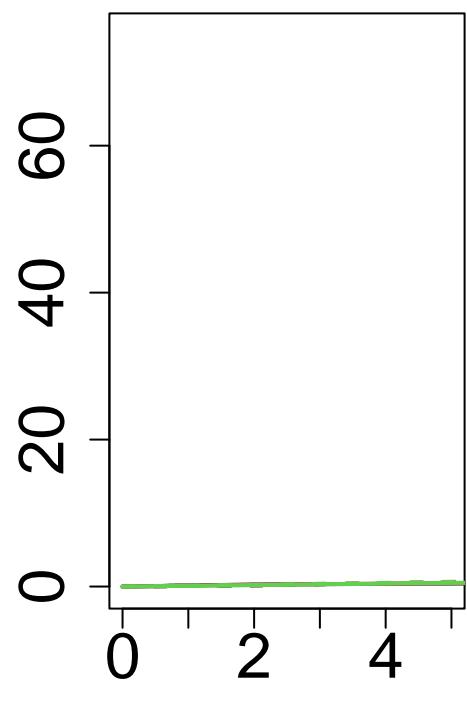
par(mfrow = c(1,2), cex.axis = 2.0, cex.lab = 2.0, cex.main = 2.0, mar = c(6.0,6,1,1), mgp = c(4, 1, 0)
# Process 1
plot(time_points,y.pred$chf[1,,1], type="l", xlab="Time",
      ylab="Process 1 CHF", col=1, lty=1, lwd=2, xlim = c(0,5), ylim = c(0,75))
lines(time_points, y.pred$chf[2,,1], col=1, lty=2, lwd=2)
# Add lines for the true survival functions
lines(time_points, true_chf11, col=2, lty=1, lwd=2)
lines(time_points, true_chf12, col=2, lty=2, lwd=2)
lines(basehazz1$time, cum_int1[,1], col=3, lty=1, lwd=2)
lines(basehazz2$time, cum_int1[,2], col=3, lty=2, lwd=2)

# Process 2
plot(time_points,y.pred$chf[1,,2], type="l", xlab="Time",
      ylab="Process 2 CHF", col=1, lty=1, lwd=2, xlim = c(0,5), ylim = c(0,75))
lines(time_points, y.pred$chf[2,,2], col=1, lty=2, lwd=2)
# Add lines for the true survival functions
lines(time_points, true_chf21, col=2, lty=1, lwd=2)
lines(time_points, true_chf22, col=2, lty=2, lwd=2)
lines(basehazz2$time, cum_int2[,1], col=3, lty=1, lwd=2)
lines(basehazz2$time, cum_int2[,2], col=3, lty=2, lwd=2)

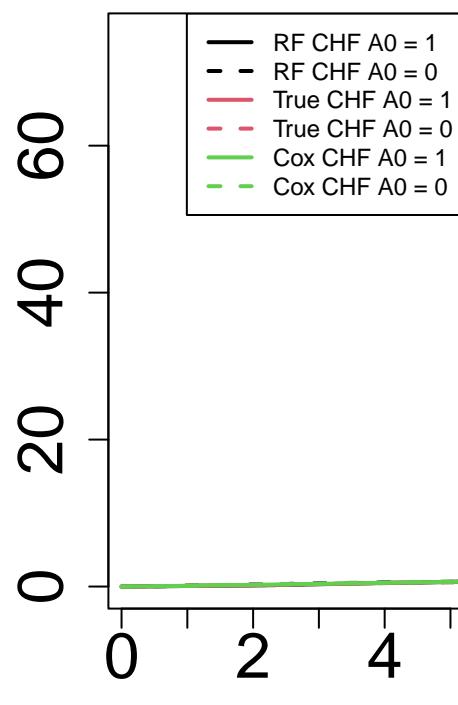
legend("topright",
       legend = c("RF CHF A0 = 1", "RF CHF A0 = 0", "True CHF A0 = 1", "True CHF A0 = 0",
                  "Cox CHF A0 = 1", "Cox CHF A0 = 0"),
       col = c(1, 1, 2, 2, 3, 3),
       lty = c(1, 2, 1, 2, 1, 2),
       lwd = 2,
       cex = 0.75)

```

Process 1 CHF



Process 2 CHF



General Event History

Yet to be implemented...