PH C240B: Assignment 1

Rachael Phillips

9/14/2017

Problem 1

Consider the mouse carcinogenesis data of Appendix A (data set V). Compute the product limit (Kaplan-Meier) estimates of the survivor function for the endpoint, reticulum cell sarcoma, for the control and germ-free groups by:

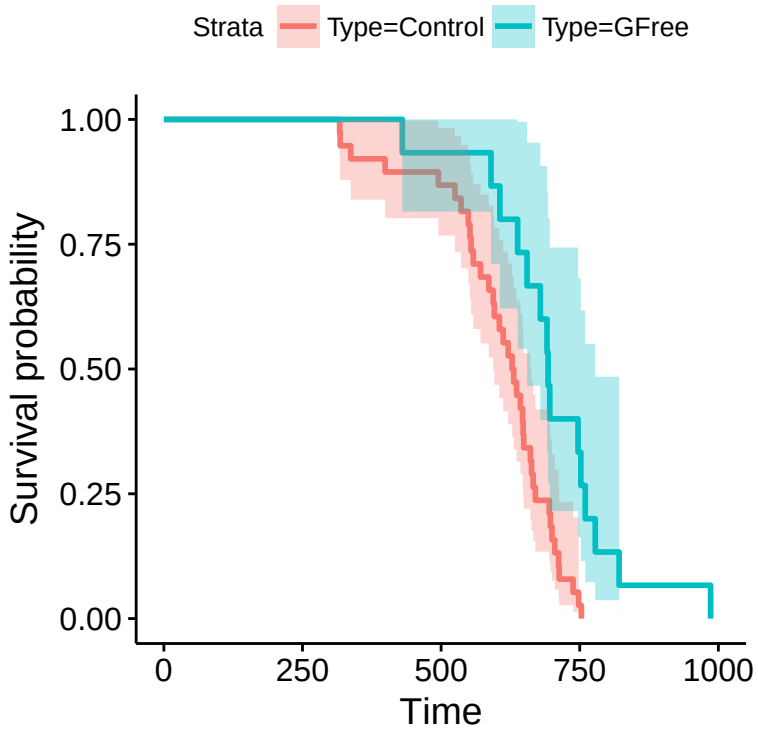
(a) Ignoring failures from thymic lymphoma and other causes (i.e., eliminate mice dying by these causes before carrying out calculations).

(b) Regarding failure times from lymphoma or other causes as right censored.

Comment on the relative merits of parts (a) and (b). (Hint: Try to understand what is being estimated in both cases.) On the basis of the survivor function plots, does the germ-free environment appear to reduce the risk of reticulum cell sarcoma?

```
# (a)

# eliminate mice dying by lymphoma or other causes
mice <- filter(micedf, Cause == 'RetCell.Sarc')
km_strata <- survfit(Surv(time = DTime) ~ Type, data = mice, type = "kaplan-meier")
ggsurvplot(km_strata, data = mice, conf.int = TRUE)
```



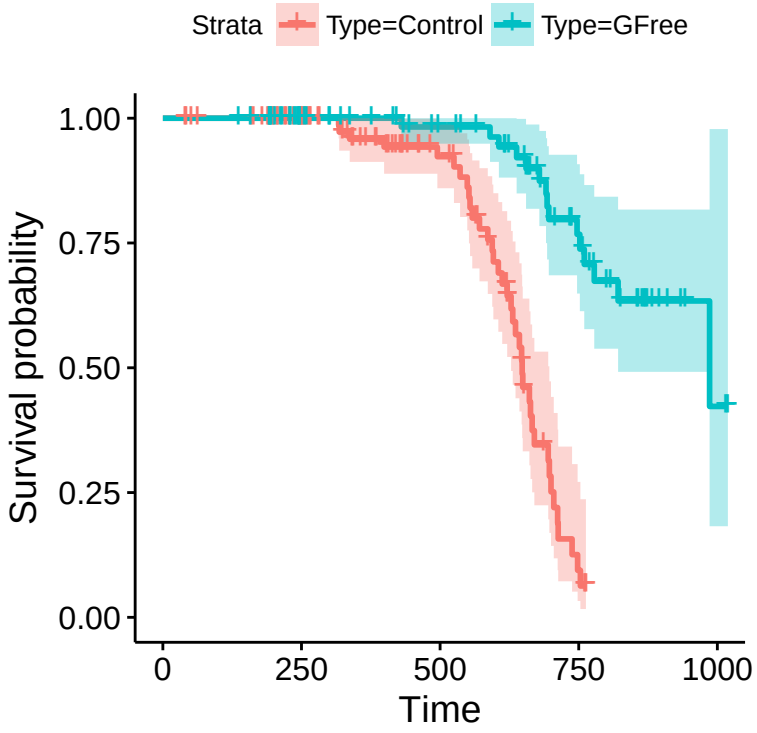

```
# (b)

# right censor failure times from lymphoma or other causes
mice_censored <- micedf %>%
```

```
  mutate(Delta = ifelse(Cause != 'RetCell.Sarc', 0, 1))

km_censored <- survfit(Surv(time = DTime, event = Delta, type = "right") ~ Type,
                       data = mice_censored, type = "kaplan-meier")

ggsurvplot(km_censored, data = mice_censored, conf.int = TRUE)
```



To examine the product limit (Kaplan-Meier) estimates of the survivor function for the endpoint, reticulum cell sarcoma, for the control and germ-free groups we had two differing approaches each with their own disadvantages and advantages:

In part (b) we regarded the failure times from lymphoma or other causes as right censored. So, we made the censoring dependent on the death time through the 'Cause' variable. The Kaplan-Meier estimator relies on independence between C and T so, by creating this dependence by basing the censoring on treatment status, we have a completely invalid estimator in (b). However, we do have a decent sample size when we consider all mice (181) so, if we used a different estimator that didn't require independence on C and T, we would feel better about the statistical inference coming from this larger sample size.

In part (a) we eliminated mice dying by thymic lymphoma and other causes before carrying out calculations. So, we omitted about 70% of our data. The sample size is reduced to 53 so we lose reliability in the inference stemming from the Kaplan-Meier estimator of the survival curve in part (a). This is because the statistical inference (here, confidence intervals) depend on the Central Limit Theorem which fails for small sample sizes or samples from a population distribution that is skewed. We don't know anything about the population distribution so we are being slighly presumptuous by assuming the sample in (a) converges to a normal distribution. However, here, T and C are indeed independent so we are not violating the independence assumption needed for the Kaplan-Meier estimator.

Even though we are not certain of the Kaplan-Meier estimates of the surivor function derived from parts (a) and (b) we can still visualize our results. On the basis of these survivor function plots, the germ-free environment does appear to reduce the risk of reticulum cell sarcoma. For both plots, the germ free environment survival rate is better than the survival rate for the control environment. However, we understand the limitations regarding our estimators so we are not certain of the validity of this statement or the reproducibility of our

results.

Problem 2

Regard the simulated distribution in Lab1sol Problem 3 in the Rlabs folder.

Perform a simulation that repeats the following experiment 1000 times: Draw a sample of 1000 iid copies of the observed data distribution and check the coverage of the truth for a 95% confidence interval for the survival probability at time t = 50. Also check simultaneous coverage for times t = 40, 50 and 60. Note, simultaneous coverage means every CI covers the corresponding truth. Use the survfit function to obtain the CI's. Briefly comment on the results.

```
# we repeat the experiment 1000 times
N = 1000

# allocating memory to store each experiment
df_40 <- data.frame(S0 = numeric(N), Sn= numeric(N),
           CI_lower = numeric(N), CI_upper = numeric(N),
           CI_indicator_40 = numeric(N))

df_50 <- data.frame(S0 = numeric(N), Sn= numeric(N),
           CI_lower = numeric(N), CI_upper = numeric(N),
           CI_indicator_50 = numeric(N))

df_60 <- data.frame(S0 = numeric(N), Sn= numeric(N),
           CI_lower = numeric(N), CI_upper = numeric(N),
           CI_indicator_60 = numeric(N))

# for each experiment:
for(i in 1:N){
# draw a sample of 1000 iid copies of the observed data distribution:
  n <- 1000
  T <- rexp(n,1/60)
  C <- rweibull(n,2,80)
  Ttilde <- pmin(T,C) #observed data
  Delta <- T < C & T <= 100
  S <- Surv(time = Ttilde, event = Delta, type = "right")
  # estimated survival function based on observed data distribution
  survival <- survfit(S~1, conf.int = .95, type = "kaplan-meier")

# check the coverage of the truth for a 95% confidence interval for the survival
# probability at time t=50 (we also need this coverage for
# t=40 and t=60 to calculate simultaneous coverage later):
  times <- c(40,50,60)

  Sn <- summary(survival,time=times)$surv #estimates
  df_40$Sn[i] <- Sn[1]
  df_50$Sn[i] <- Sn[2]
  df_60$Sn[i] <- Sn[3]

  S0 <- function(t) 1-pexp(t,1/60) #true survival function
  S0 <- S0(times) #true survival times
  df_40$S0[i] <- S0[1]
  df_50$S0[i] <- S0[2]
```

```
  df_60$S0[i] <- S0[3]

  CI_lower <- summary(survival,time=times)$lower #lower bound of CIs
  df_40$CI_lower[i] <- CI_lower[1]
  df_50$CI_lower[i] <- CI_lower[2]
  df_60$CI_lower[i] <- CI_lower[3]

  CI_upper <- summary(survival,time=times)$upper #upper bound of CIs
  df_40$CI_upper[i] <- CI_upper[1]
  df_50$CI_upper[i] <- CI_upper[2]
  df_60$CI_upper[i] <- CI_upper[3]

  # did the confidence interval capture the truth?
  indicator_40 <- as.numeric(CI_lower[1] <  S0[1] && CI_upper[1] >  S0[1])
  df_40$CI_indicator_40[i] <- indicator_40

  indicator_50 <- as.numeric(CI_lower[2] <  S0[2] && CI_upper[2] >  S0[2])
  df_50$CI_indicator_50[i] <- indicator_50

  indicator_60 <- as.numeric(CI_lower[3] <  S0[3] && CI_upper[3] >  S0[3])
  df_60$CI_indicator_60[i] <- indicator_60
}

# proportion of times the CI captured the truth across all iterations
print(paste('Our coverage for t=40 is', mean(df_40$CI_indicator_40)))
print(paste('Our coverage for t=50 is', mean(df_50$CI_indicator_50)))
print(paste('Our coverage for t=60 is', mean(df_60$CI_indicator_60)))

# simultaneous coverage:
df <- data.frame(df_40$CI_indicator_40,
                 df_50$CI_indicator_50,
                 df_60$CI_indicator_60)

# proportion of iterations where all CIs captured their truth
simultaneous_coverage <- sum(rowMeans(df) == 1)/N

print(paste('Our simultaneous for t=(40,50,60) is', simultaneous_coverage))
```

```
## [1] "Our coverage for t=40 is 0.952"
## [1] "Our coverage for t=50 is 0.942"
## [1] "Our coverage for t=60 is 0.943"
## [1] "Our simultaneous for t=(40,50,60) is 0.89"
```

The confidence intervals for each time point were able to capture their truth for around 95% of the 1000 experiments. We see that each confidence interval covers it's correspond truth simultaneously 90% of the time across all experiements. We do expect the simultaneous coverage probability to be lower than the individual coverage probability because of the way we calculated the simultaneous coverage. We calculated the simultaneous coverage based on the combination of individually based confidence intervals (i.e. a confidence for one time point) not based on the simultaneous confidence interval (i.e. a confidence for a set of times). If we had constructed a 95% simultaneous confidence interval (stemming from a multivariate normal distribution with mean equal to 0 and variance equal to the correlation martix of the covariance matrix) then we would expect all intervals to contain their corresponding truth with 95% confidence.

Problem 3

Repeat 2 but for the distribution in Problem 5 of Lab1sol. Briefly comment on the results.

```
# we repeat the experiment 1000 times
N = 1000

# allocating memory to store each experiment
df_40 <- data.frame(S0 = numeric(N), Sn= numeric(N),
           CI_lower = numeric(N), CI_upper = numeric(N),
           CI_indicator_40 = numeric(N))

df_50 <- data.frame(S0 = numeric(N), Sn = numeric(N),
           CI_lower = numeric(N), CI_upper = numeric(N),
           CI_indicator_50 = numeric(N))

df_60 <- data.frame(S0 = numeric(N), Sn= numeric(N),
           CI_lower = numeric(N), CI_upper = numeric(N),
           CI_indicator_60 = numeric(N))

# for each experiment:
for(i in 1:N){

  # generating the true survival curve
  A0 <- rbinom(1e6, 1, .5)
  T0 <- rexp(1e6,(A0*(1/90) + (1/180)))
  F0 <- ecdf(T0)
  S0 <- function(t) 1 - F0(t)

# draw a sample of 1000 iid copies of the observed data distribution:
  n = 1e3
  A = rbinom(n,1,.5)
  T = rexp(n,(A*(1/90) + (1/180)))
  C = rweibull(n,2,(-A*80 + 120))
  Ttilde = pmin(C,T) #observed data
  Delta = T <= C | T <= 100
  S <- Surv(time = Ttilde, event = Delta, type = "right")
  # estimated survival function based on observed data distribution
  survival <- survfit(S~1, conf.int = .95, type = "kaplan-meier")

# check the coverage of the truth for a 95% confidence interval for the survival
# probability at time t=50 (we also need this coverage for
# t=40 and t=60 to calculate simultaneous coverage later):
  times <- c(40,50,60)

  Sn <- summary(survival,time=times)$surv #estimates
  df_40$Sn[i] <- Sn[1]
  df_50$Sn[i] <- Sn[2]
  df_60$Sn[i] <- Sn[3]

  S0 <- S0(times) #true survival times
  df_40$S0[i] <- S0[1]
  df_50$S0[i] <- S0[2]
  df_60$S0[i] <- S0[3]
```

```
  CI_lower <- summary(survival,time=times)$lower #lower bound of CIs
  df_40$CI_lower[i] <- CI_lower[1]
  df_50$CI_lower[i] <- CI_lower[2]
  df_60$CI_lower[i] <- CI_lower[3]

  CI_upper <- summary(survival,time=times)$upper #upper bound of CIs
  df_40$CI_upper[i] <- CI_upper[1]
  df_50$CI_upper[i] <- CI_upper[2]
  df_60$CI_upper[i] <- CI_upper[3]

  # did the confidence interval capture the truth?
  indicator_40 <- as.numeric(CI_lower[1] <  S0[1] && CI_upper[1] >  S0[1])
  df_40$CI_indicator_40[i] <- indicator_40

  indicator_50 <- as.numeric(CI_lower[2] <  S0[2] && CI_upper[2] >  S0[2])
  df_50$CI_indicator_50[i] <- indicator_50

  indicator_60 <- as.numeric(CI_lower[3] <  S0[3] && CI_upper[3] >  S0[3])
  df_60$CI_indicator_60[i] <- indicator_60
}
# proportion of times the CI captured the truth across all iterations
print(paste('Our coverage for t=40 is', mean(df_40$CI_indicator_40)))
print(paste('Our coverage for t=50 is', mean(df_50$CI_indicator_50)))
print(paste('Our coverage for t=60 is', mean(df_60$CI_indicator_60)))

# simultaneous coverage:
df <- data.frame(df_40$CI_indicator_40,
                 df_50$CI_indicator_50,
                 df_60$CI_indicator_60)

# proportion of iterations where all CIs captured their truth
simultaneous_coverage <- sum(rowMeans(df) == 1)/N

print(paste('Our simultaneous for t=(40,50,60) is', simultaneous_coverage))
```

```
## [1] "Our coverage for t=40 is 0"
## [1] "Our coverage for t=50 is 0"
## [1] "Our coverage for t=60 is 0"
## [1] "Our simultaneous for t=(40,50,60) is 0"
```

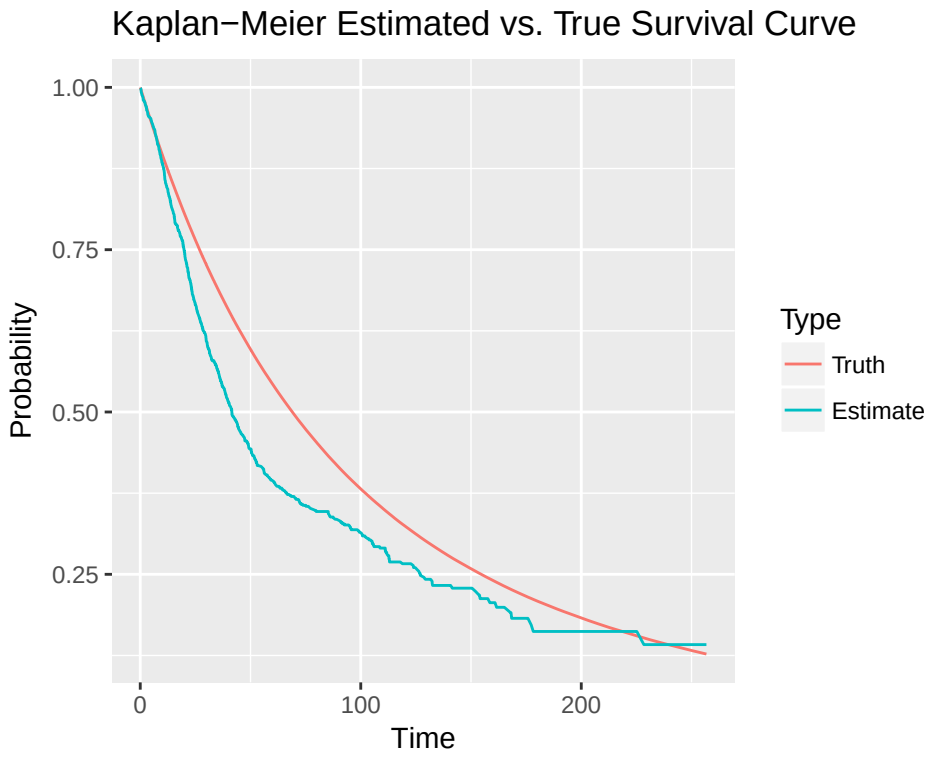
Again, we see that the simultaneous coverage is smaller than each of the individual coverages for the same reason explained in the previous question. However, in this problem we see the individual coverages all fail to reach 95%. Additionally, we notice that as the time increases the coverage dramatically decreases. Notice in the code that the censoring time and time to event are dependent through the treatment so the independence assumption of C and T needed by Kaplan-Meier estimator is not upheld. Thus, the Kaplan-Meier estimator of the survival funciton based on the observed data is a poor representation of the true data distribution and, according to our coverages, this gap in capturing the truth most likely increases as the time increases. We can visualize this discrepancy with a plot of these survival curves.

```
n = 1e3
A = rbinom(n,1,.5)
T = rexp(n,(A*(1/90) + (1/180)))
C = rweibull(n,2,(-A*80 + 120))
Ttilde = pmin(C,T)
```

```
Delta = T <= C | T <= 100
S <- Surv(time = Ttilde, event = Delta, type = "right")
est_survival <- survfit(S~1, conf.int = .95, type = "kaplan-meier")

A0 <- rbinom(1e6, 1, .5)
T0 <- rexp(1e6,(A0*(1/90) + (1/180)))
F0 <- ecdf(T0)
S0 <- function(t) 1 - F0(t)
true_survival <- S0(est_survival$time)

df <- melt(data.frame(Truth = true_survival, Estimate = est_survival$surv,
                      Time = est_survival$time), id="Time")
df <- data.frame(Time=df$Time, Type=df$variable, Probability = df$value)
ggplot(df, aes(x = Time, y = Probability, color = Type)) + geom_line() +
  labs(title="Kaplan-Meier Estimated vs. True Survival Curve")
```



From this plot we can visualize how the Kaplan-Meier estimator fails to represent the true data distribution when the independence assumption of C and T is violated.

Collaborators

Thomas Carpenito