TMA4275; Project 2

Ola Rasmussen

Contents

Problem 1:

Introduction:

In this problem we are interested in analyzing the effect of a cancer treatment, and if it results in a longer lifetime for the patients.

The patients who where analyzed had either a negative stained tumor of a positive stained tumor. They where divided into two groups, group 1 and group 2. The patients in group 1 had negatively stained tumors, and the patients in group 2 had negatively stained tumors. Their survival times are given in the table below,

```
## Group 1: Group 2:
##    23        5
##    47        8
##    69       10
##    70*      13
##   100*      18
##   101*      24
##   148       26
##   181       31
##   198*      35
##   208*      50
##   212*      59
##   244*      61
##             76*
##            109*
##            116*
##            118
##            143
##            154*
##            162*
##            225*
```

where the censored survival times are marked with an asterisk (*).

a)

Here we will make plots of $Y_1(t)$ and $Y_2(t)$, where $Y_i(t)$ is the number of patients at risk in group i. To do this we will use the following formula for $Y(t)$:

$$Y_i(T_j) = \sum_{k=1}^{n_i} I(T_j < \tau_k), \tag{1.1}$$

where T_j are the observed survival times and τ_k are the times we want to check how many are under risk.

```r
# Defining the survival times
G1 <- c(23, 47, 69, 70, 71, 100, 101, 148, 181, 198, 208, 212,
    224)
G2 <- c(5, 8, 10, 13, 18, 24, 26, 31, 35, 50, 59, 61, 76, 109,
    116, 118, 143, 154, 162, 225)
# Defining the censored survival times
d1 <- c(1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)
d2 <- c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0,
    0, 0)
# Creating Y1 and Y2 from Eq. (1.1)
Y1 <- seq(13, 1)
Y2 <- seq(20, 1)
# Making the graph look good
G1_Y <- append(append(0, G1), 225)
G2_Y <- append(append(0, G2), 225)
par(mfrow = c(2, 1))
plot(G1_Y, append(append(Y1[1], Y1 - 1), 0), lwd = 1.5, type = "s",
    col = "blue", main = "Group 1", xlab = latex2exp("$t$"),
    ylab = latex2exp("$Y_1(t)$"), xlim = c(0, 225), ylim = c(0,
        13))
plot(G2_Y, append(append(Y2[1], Y2 - 1), 0), lwd = 1.5, type = "s",
    col = "red", main = "Group 2", xlab = latex2exp("$t$"), ylab = latex2exp("$Y_2(t)$")
    xlim = c(0, 225), ylim = c(0, 20))
```

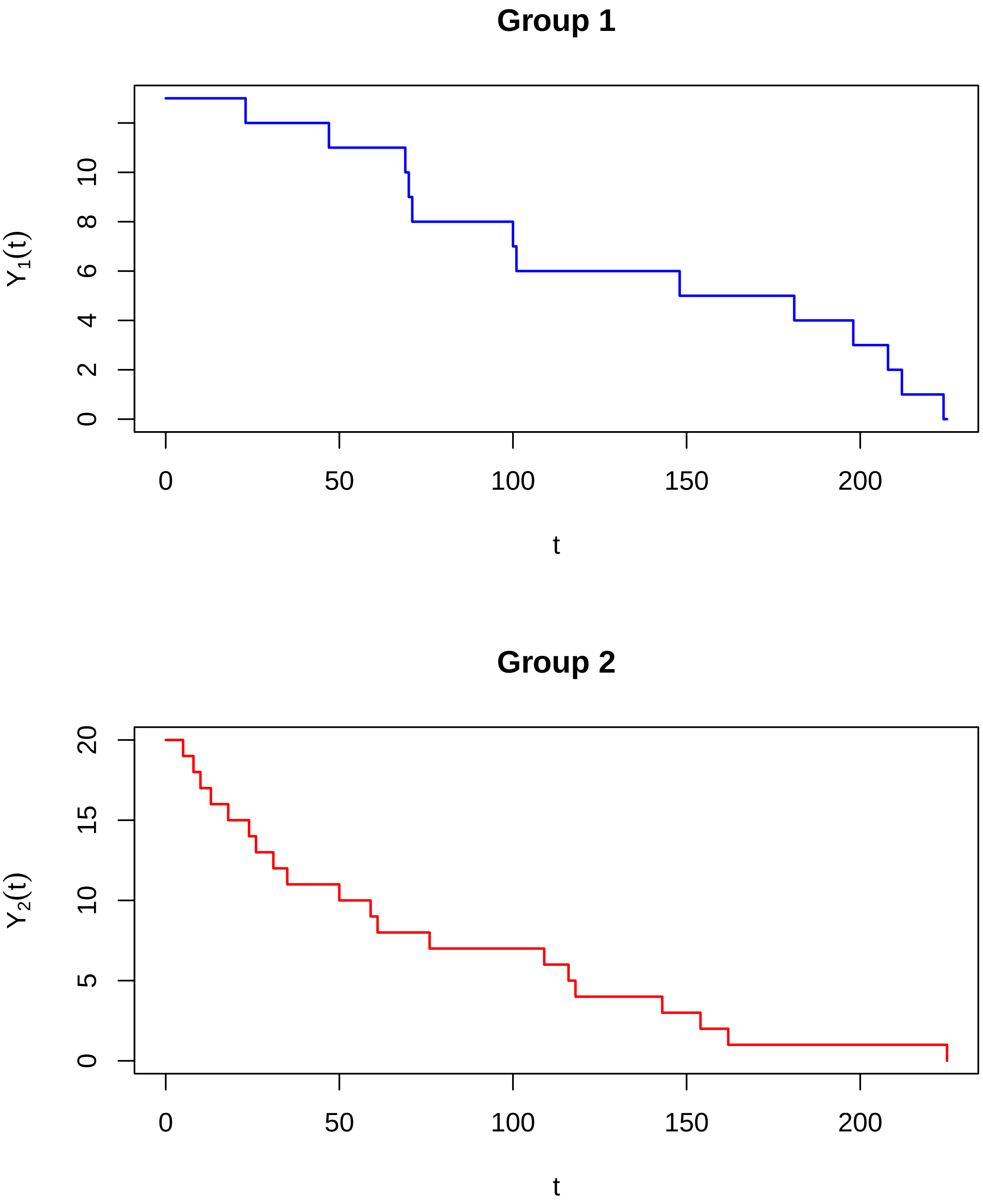


Figure 1: Number of individuals at risk in each group.

b)

Now we will make a function that returns the Nelson-Aalen estimator for the associated group, and other values so we can compute the associated confidence interval based on the log-transformation. The Nelson-Aalen estimator is given by,

$$\hat{A}_i(t) = \sum_{j|T_j \leq t} \frac{d_i(T_j)}{Y_i(T_j)}, \tag{1.2}$$

and the log-transformation confidence interval is given by,

$$\hat{A}_i(t) \cdot exp\left\{ \pm z_{1-\frac{\alpha}{2}} \cdot \frac{\hat{\sigma}_i(t)}{\hat{A}_i(t)} \right\}. \tag{1.3}$$

So our function needs to return $\hat{A}_i(t)$ and $\hat{\sigma}_i(t)$. $\hat{\sigma}_i^2(t)$, the variance of $\hat{A}_i(t)$, is given by,

$$\hat{\sigma}_i^2(t) = \sum_{T_j \leq t} \frac{d_i(T_j)}{Y_i^2(T_j)}. \tag{1.4}$$

The $d_i(T_j)$'s are there to indicate if the time is censored or not at time T_j. $d_i(T_j) = 1$ if the time is not censored, and $d_i(T_j) = 0$ if it's censored.

```r
fun1b <- function(Y1, Y2, d1, d2) {
    # Calc. Nelson-Aalen estimators from Eq. (1.2)
    hatA1 <- cumsum(d1/Y1)
    hatA2 <- cumsum(d2/Y2)
    # Calc. variance of the Nelson-Aalen estimators from
    # Eq. (1.4)
    hatSigma12 <- cumsum(d1/(Y1)^2)
    hatSigma22 <- cumsum(d2/(Y2)^2)
    return(list(hatA1, hatA2, hatSigma12, hatSigma22))
}
fun1 <- fun1b(Y1, Y2, d1, d2)
hatA1 <- fun1[[1]]
hatA2 <- fun1[[2]]
hatSigma12 <- fun1[[3]]
hatSigma22 <- fun1[[4]]
# Calc. the log-transformed 90% C.I. for the Nelson-Aalen
# estimators from Eq. (1.3)
z <- qnorm(0.95)
Nelson_upper_CI_1 <- hatA1 * exp(z * sqrt(hatSigma12)/hatA1)
Nelson_lower_CI_1 <- hatA1 * exp(-z * sqrt(hatSigma12)/hatA1)
Nelson_upper_CI_2 <- hatA2 * exp(z * sqrt(hatSigma22)/hatA2)
Nelson_lower_CI_2 <- hatA2 * exp(-z * sqrt(hatSigma22)/hatA2)
```

```
# Making the graph look good
G1_Nelson <- append(append(0, G1), 225)
G2_Nelson <- append(0, G2)
plot(G1_Nelson, append(append(0, hatA1), tail(hatA1, 1)), lwd = 1.5,
    type = "s", col = "blue", main = "", xlab = latex2exp("$t$"),
    ylab = "Value", xlim = c(0, 225), ylim = c(0, 3))
lines(G2_Nelson, append(0, hatA2), lwd = 1.5, type = "s", col = "red")
lines(G1_Nelson, append(append(0, Nelson_upper_CI_1), tail(Nelson_upper_CI_1,
    1)), lwd = 1.5, lty = 2, type = "s", col = "blue")
lines(G1_Nelson, append(append(0, Nelson_lower_CI_1), tail(Nelson_lower_CI_1,
    1)), lwd = 1.5, lty = 2, type = "s", col = "blue")
lines(G2_Nelson, append(0, Nelson_upper_CI_2), lwd = 1.5, lty = 2,
    type = "s", col = "red")
lines(G2_Nelson, append(0, Nelson_lower_CI_2), lwd = 1.5, lty = 2,
    type = "s", col = "red")
legend(0, 3, c("Nelson-Aalen estimator for Group 1", "Nelson-Aalen estimator for Group 2
    col = c("blue", "red"), lty = c(1, 1), lwd = c(2, 2))
```

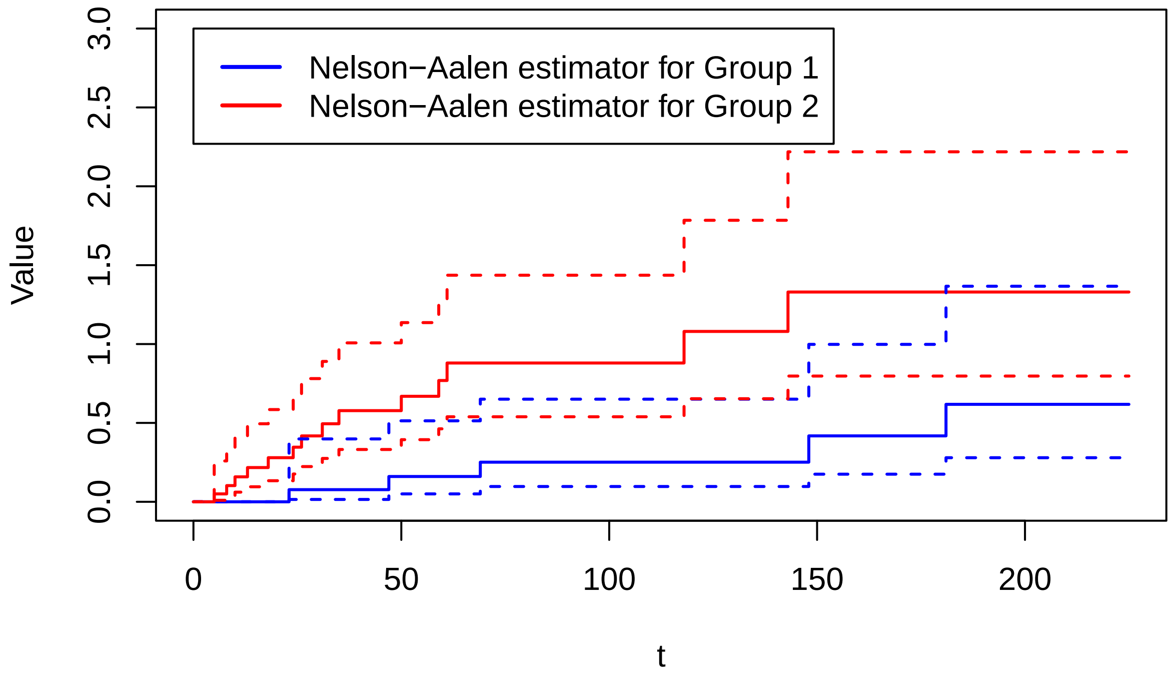


Figure 2: Nelson-Aalen estimators together with their 90 percent confidence interval.

c)

Here we will make a function that computes the Kaplan-Meier estimator and other values necessary to compute the log-minus-log-transformation confidence interval.

The Kaplan-Meier estimator is given by,

$$\hat{S}_i(t) = \prod_{j|T_j \leq t} \left(1 - \frac{d_i(T_j)}{Y_i(T_j)}\right). \tag{1.5}$$

The log-minus-log-transformation confidence interval is given by,

$$\hat{S}_i(t)^{exp\left\{\pm z_{1-\frac{\alpha}{2}} \frac{\hat{\tau}_i(t)}{\hat{S}_i(t) log(\hat{S}_i(t))}\right\}}, \tag{1.6}$$

where $\hat{\tau}_i^2(t)$, the variance of the Kaplan-Meier estimate, is given by,

$$\hat{\tau}_i^2(t) = \hat{S}_i^2(t) \sum_{T_j \leq t} \frac{d_i(T_j)}{Y_i^2(T_j)}. \tag{1.7}$$

```r
fun1c <- function(Y1, Y2, d1, d2) {
    # Calc. the Kaplan-Meier estimators from Eq. (1.5)
    hatS1 <- cumprod(1 - (d1/Y1))
    hatS2 <- cumprod(1 - (d2/Y2))
    # Calc. the variance of the Kaplan-Meier estimators rom
    # Eq. (1.7)
    hatTau12 <- hatS1^2 * cumsum(d1/(Y1)^2)
    hatTau22 <- hatS2^2 * cumsum(d2/(Y2)^2)
    return(list(hatS1, hatS2, hatTau12, hatTau22))
}
fun2 <- fun1c(Y1, Y2, d1, d2)
hatS1 <- fun2[[1]]
hatS2 <- fun2[[2]]
hatTau12 <- fun2[[3]]
hatTau22 <- fun2[[4]]
# Calc. the log-minus-log-transformed 90% C.I. for the
# Kaplan-Meier estimators from Eq. (1.6)
Kaplan_upper_CI_1 <- hatS1^exp(z * sqrt(hatTau12)/(hatS1 * log(hatS1)))
Kaplan_lower_CI_1 <- hatS1^exp(-z * sqrt(hatTau12)/(hatS1 * log(hatS1)))
Kaplan_upper_CI_2 <- hatS2^exp(z * sqrt(hatTau22)/(hatS2 * log(hatS2)))
Kaplan_lower_CI_2 <- hatS2^exp(-z * sqrt(hatTau22)/(hatS2 * log(hatS2)))
```

```r
# Making the graph look good
G1_Kaplan <- append(append(0, G1), 225)
G2_Kaplan <- append(0, G2)
plot(G1_Kaplan, append(append(1, hatS1), tail(hatS1, 1)), lwd = 1.5,
    type = "s", col = "blue", main = "", xlab = latex2exp("$t$"),
    ylab = "Value", xlim = c(0, 225), ylim = c(0, 1.4))
lines(G2_Kaplan, append(1, hatS2), lwd = 1.5, type = "s", col = "red")
lines(G1_Kaplan, append(append(1, Kaplan_upper_CI_1), tail(Kaplan_upper_CI_1,
    1)), lwd = 1.5, lty = 2, type = "s", col = "blue")
lines(G1_Kaplan, append(append(1, Kaplan_lower_CI_1), tail(Kaplan_lower_CI_1,
    1)), lwd = 1.5, lty = 2, type = "s", col = "blue")
lines(G2_Kaplan, append(1, Kaplan_upper_CI_2), lwd = 1.5, lty = 2,
    type = "s", col = "red")
lines(G2_Kaplan, append(1, Kaplan_lower_CI_2), lwd = 1.5, lty = 2,
    type = "s", col = "red")
legend(0, 1.4, c("Kaplan-Meier estimator for Group 1", "Kaplan-Meier estimator for Group
    col = c("blue", "red"), lty = c(1, 1), lwd = c(2, 2))
```

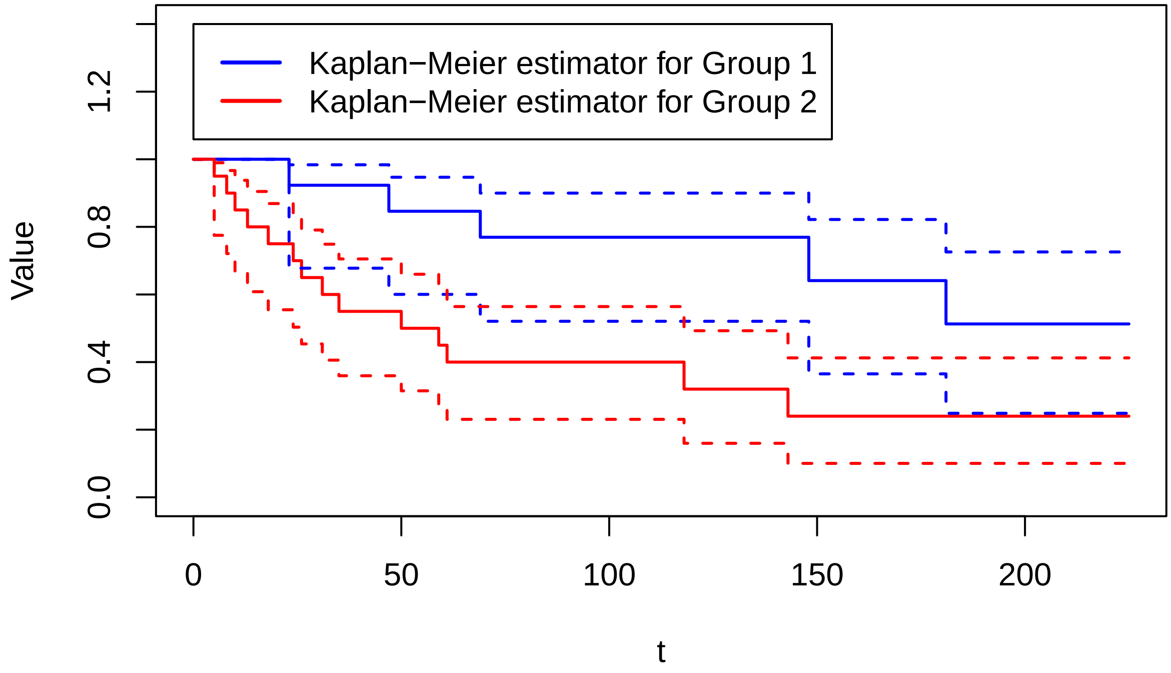


Figure 3: Kaplan-Meier estimators together with their 90 percent confidence interval.

Now we will check our function using the R function “survfit” to produce the Kaplan-Meier estimators for each group.

```r
S1 <- Surv(G1, d1)
S2 <- Surv(G2, d2)
survfit1 <- survfit(S1 ~ 1, conf.type = "log-log", conf.int = 0.9)
survfit2 <- survfit(S2 ~ 1, conf.type = "log-log", conf.int = 0.9)
plot(survfit1, col = "blue", lwd = 1.5, main = "", xlab = latex2exp("$t$"),
    ylab = "Value", xlim = c(0, 225), ylim = c(0, 1.4))
lines(survfit2, col = "red", lwd = 1.5)
legend(0, 1.4, c("Kaplan-Meier estimator for Group 1", "Kaplan-Meier estimator for Group
    col = c("blue", "red"), lty = c(1, 1), lwd = c(1.5, 1.5))
```

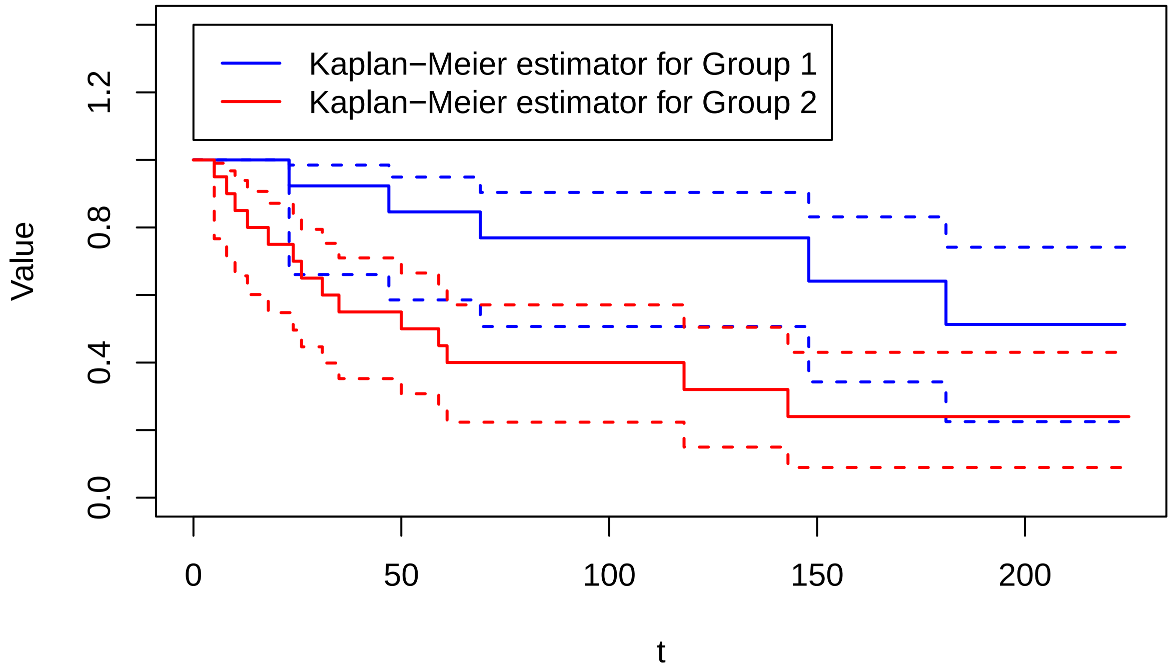


Figure 4: Kaplan-Meier estimators from survfit function, together with their 90 percent confidence interval.

We get the same result as the R function. We see in Figure 3 and Figure 4 that the blue curve (Group 1) is larger than the red curve (Group 2), so based on the results so far, we can conclude that the patients in Group 1 tends to live longer than the patients in Group 2.

d)

Now we want to estimate the 20 fractile, $\xi_{0.2}$, for each of the two groups and find the associated confidence intervals. The pth fractile $\xi_{i,p}$, for Group i, is given by,

$$F_i(\xi_{i,p}) = 1 - S_i(\xi_{i,p}) = p. \tag{1.8}$$

This can be estimated using the Kaplan-Meier estimator. Since the Kaplan-Meier estimator is a step function, it does not necessarily attain a value of $1-p$. We then rather define $\hat{\xi}_{i,p}$ to be the smallest value for $t$ for which,

$$\hat{S}_i(t) \leq 1 - p, \;\; t = \hat{\xi}_{i,p}. \tag{1.9}$$

We find below that $\xi_{1,0.2} = 69$ for Group 1 with confidence interval $(23, 181)$, and that $\xi_{2,0.2} = 13$ for Group 2 with confidence interval $(5, 26)$.

```r
# Finding Xi_0.2from Eq. (1.9) and C.I.
fr <- 1 - 0.2
xi1 <- G1[hatS1 <= fr][1]
xi_upper_CI_1 <- G1[Kaplan_upper_CI_1 <= fr][1]
xi_lower_CI_1 <- G1[Kaplan_lower_CI_1 <= fr][1]
xi2 <- G2[hatS2 <= fr][1]
xi_upper_CI_2 <- G2[Kaplan_upper_CI_2 <= fr][1]
xi_lower_CI_2 <- G2[Kaplan_lower_CI_2 <= fr][1]
cat("We find that the 20% fractile for Group 1 is", xi1, "with confidence interval",
    paste("(", xi_lower_CI_1, ",", xi_upper_CI_1, ")", sep = ""),
    "\nand", "\nwe find that the 20% fractile for Group 2 is",
    xi2, "with confidence interval", paste("(", xi_lower_CI_2,
        ",", xi_upper_CI_2, ")", sep = ""))
```

```
## We find that the 20% fractile for Group 1 is 69 with confidence interval (23,181)
## and
## we find that the 20% fractile for Group 2 is 13 with confidence interval (5,26)
```

```r
# Making the graph look good
plot(survfit1, col = "blue", lwd = 1, main = "", xlab = latex2exp("$t$"),
    ylab = "Value", xlim = c(0, 225), ylim = c(0, 1.5))
lines(survfit2, col = "red", lwd = 1)
```

```
abline(h = fr, lwd = 1.5, lty = 2, col = "black")
lines(c(xi1, xi1), c(0, fr), type = "l", lwd = 2.5, col = "blue")
lines(c(xi_upper_CI_1, xi_upper_CI_1), c(0, fr), type = "l",
    lwd = 2.5, lty = 2, col = "blue")
lines(c(xi_lower_CI_1, xi_lower_CI_1), c(0, fr), type = "l",
    lwd = 2.5, lty = 2, col = "blue")
lines(c(xi2, xi2), c(0, fr), type = "l", lwd = 2.5, lty = 1,
    col = "red")
lines(c(xi_upper_CI_2, xi_upper_CI_2), c(0, fr), type = "l",
    lwd = 2.5, lty = 2, col = "red")
lines(c(xi_lower_CI_2, xi_lower_CI_2), c(0, fr), type = "l",
    lwd = 2.5, lty = 2, col = "red")
axis(1, at = c(xi1, xi2), labels = c(xi1, xi2))
legend(0, 1.5, c("Kaplan-Meier estimator for Group 1", "Kaplan-Meier estimator for Group
    "20 percent fractile"), col = c("blue", "red", "black"),
    lty = c(1, 1, 2), lwd = c(1.5, 1.5, 1.5))
```

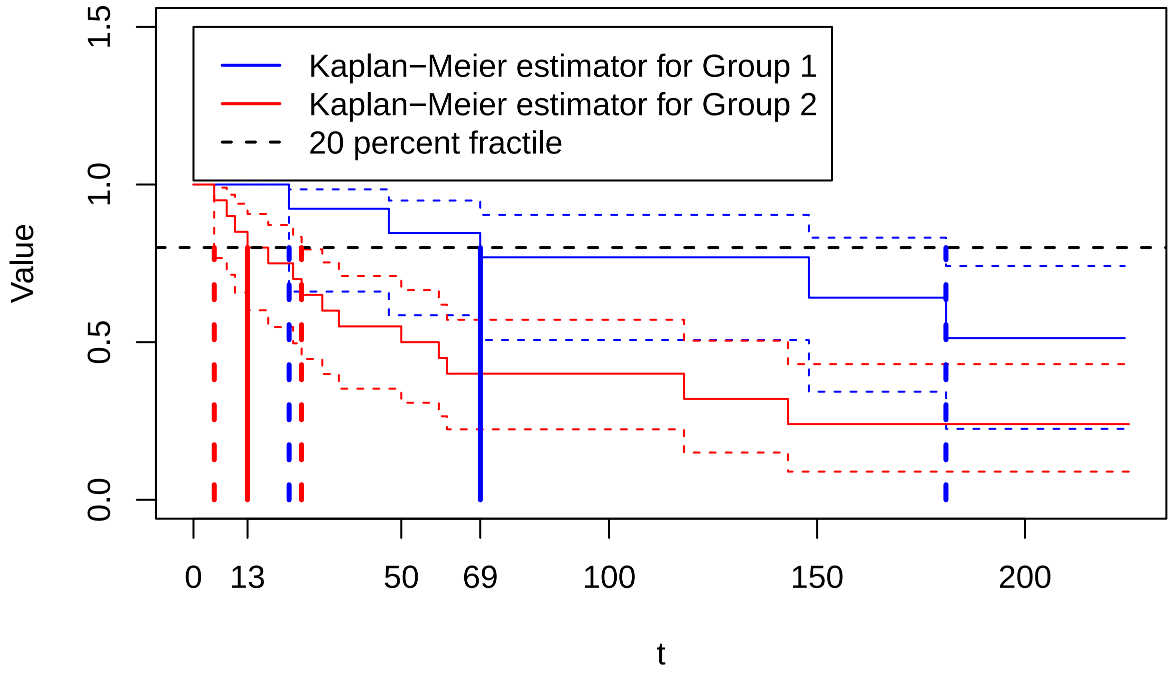


Figure 5: The dashed black line show where the estimate of the 20 percent fractile is located, the full vertical lines mark the 20 percent fractile for both groups, and the dashed lines show the confidence interval.

e)

To end this problem, we will use the log-rank test to test $H_0: \ \alpha_1(t) = \alpha_2(t)$, for $t \in [0, t_0]$ where $t_0 = 200$, and then find the p-value.

The test statistic for this test is,

$$U(t_0) = \frac{Z_1(t_0)}{\sqrt{V_{11}(t_0)}} \approx N(0,1), \tag{1.10}$$

where $Z_1(t_0)$ is given by,

$$Z_1(t_0) = \int_0^{t_0} L(t) \left(d\hat{A}_1(t) - d\hat{A}_2(t) \right), \tag{1.11}$$

and where $V_{11}(t_0)$ is given by,

$$V_{11}(t_0) = \int_0^{t_0} \frac{L^2(t)}{Y_1(t)Y_2(t)} dN_\bullet(t). \tag{1.12}$$

Alternatively, we can use the test statistic,

$$X^2(t_0) = \frac{Z_1^2(t_0)}{V_{11}(t_0)} \approx X_1^2, \tag{1.13}$$

but this will not be used here, because we will evaluate the normal approximation of the test statistic in Equation (1.10) in Problem 2.

The increments $d\hat{A}_i(t)$ can be found using,

$$\hat{A}_i(t) = \int_0^{t} \frac{J_i(s)}{Y_i(s)} dN_i(s), \tag{1.14}$$

where $J_i(s) = I(Y_i(s) > 0)$ for group i and $d\hat{A}_i(t) = \frac{J_i(s)}{Y_i(s)} dN_i(s)$.

Plugging this into $Z_1(t_0)$ we get,

$$\begin{aligned} Z_1(t_0) &= \int_0^{t_0} L(t) \left(\frac{J_1(t)}{Y_1(t)} dN_1(t) - \frac{J_2(t)}{Y_2(t)} dN_2(t) \right) \\ &= \int_0^{t_0} L(t) \left(\frac{J_1(t)}{Y_1(t)} dN_1(t) \right) - \int_0^{t_0} L(t) \left(\frac{J_2(t)}{Y_2(t)} dN_2(t) \right) \\ &= \int_0^{t_0} \left(\frac{L(t)}{Y_1(t)} dN_1(t) \right) - \int_0^{t_0} \left(\frac{L(t)}{Y_2(t)} dN_2(t) \right). \end{aligned}$$

When we are using the log-rank test, $L(t)$ becomes,

$$L(t) = \frac{Y_1(t)Y_2(t)}{Y_\bullet(t)}, \tag{1.15}$$

where $Y_\bullet(t) = Y_1(t) + Y_2(t)$. Plugging this into $Z_1(t_0)$ and $V_{11}(t_0)$ we then get,

$$Z_1(t_0) = \int_0^{t_0} \left(\frac{Y_2(t)}{Y_1(t) + Y_2(t)} dN_1(t) \right) - \int_0^{t_0} \left(\frac{Y_1(t)}{Y_1(t) + Y_2(t)} dN_2(t) \right),$$

$$V_{11}(t_0) = \int_0^{t_0} \frac{Y_1(t)Y_2(t)}{(Y_1(t) + Y_2(t))^2} dN_\bullet(t).$$

Since $N_i(s)$ is a counting process, $Z_1(t_0)$ becomes,

$$Z_1(t_0) = \sum_{j|T_{1,j} \leq t_0} \left(\frac{Y_2(T_{1,j})}{Y_1(T_{1,j}) + Y_2(T_{1,j})} \right) - \sum_{j|T_{2,j} \leq t_0} \left(\frac{Y_1(T_{2,j})}{Y_1(T_{2,j}) + Y_2(T_{2,j})} \right), \tag{1.16}$$

and $V_{11}(t_0)$ becomes,

$$V_{11}(t_0) = \sum_{j|T_j \leq t_0} \frac{Y_1(T_j)Y_2(T_j)}{(Y_1(T_j) + Y_2(T_j))^2}, \tag{1.17}$$

where $T_{1,j}$ is every uncensored survival time for Group 1, $T_{2,j}$ is every uncensored survival time for Group 2, and T_j is every uncensored survival time for both Group 1 and 2.

```r
fun1e <- function(Y1, Y2, G1, G2, d1, d2) {
    G1test <- G1[G1 <= 200]
    G2test <- G2[G2 <= 200]
    Y1test <- stepfun(append(G1test, 200), (append(head(Y1, 1),
        append(Y1[G1 <= 200] - 1, tail(Y1[G1 <= 200], 1) - 1))),
        right = T)
    Y2test <- stepfun(append(G2test, 200), (append(head(Y2, 1),
        append(Y2[G2 <= 200] - 1, tail(Y2[G2 <= 200], 1) - 1))),
        right = T)
    G1testtest <- (G1test * d1[G1 <= 200])[(G1test * d1[G1 <=
        200]) != 0]
    G2testtest <- (G2test * d2[G2 <= 200])[(G2test * d2[G2 <=
        200]) != 0]
    Gtesttest <- sort(append(G1testtest[G1testtest < 200], G2testtest))
    Z11 <- sum(Y2test(G1testtest)/(Y1test(G1testtest) + Y2test(G1testtest)))
    Z12 <- sum(Y1test(G2testtest)/(Y1test(G2testtest) + Y2test(G2testtest)))
    Z1 <- Z11 - Z12
    V11 <- sum((Y1test(Gtesttest) * Y2test(Gtesttest))/((Y1test(Gtesttest) +
        Y2test(Gtesttest))^2))
    U0 <- Z1/sqrt(V11)
    p_value <- 2 * min(pnorm(U0, lower.tail = F), pnorm(U0))
    return(list(U0, p_value))
}
U0 <- fun1e(Y1, Y2, G1, G2, d1, d2)[[1]]
p_value <- fun1e(Y1, Y2, G1, G2, d1, d2)[[2]]
cat("We get that the value of U_0 is", paste(round(U0, 3), ",",
    sep = ""), "\nand since this is outside of", paste("(", round(qnorm(0.95,
    lower.tail = F), 3), ",", round(qnorm(0.95), 3), ")", ",",
    sep = ""), "we get that H0 should be discarded. \nThe associated p-value is",
    paste(round(p_value, 3), ",", sep = ""), "which is sufficiently small.")
```

```
## We get that the value of U_0 is -2.237,
## and since this is outside of (-1.645,1.645), we get that H0 should be discarded.
## The associated p-value is 0.025, which is sufficiently small.
```

There is evidence that Group 1 and Group 2 have different hazard rates. From our findings, Group 2 seems to have a greater hazard rate than Group 1. So, people with negative HPA tends to die at a faster rate than people with positive HPA.

Problem 2:

Introduction:

In this problem, we will use stochastic simulation to evaluate the quality of the normal approximation of the log-rank test statistic when testing $\alpha_1(t) = \alpha_2$, $t \in [0, t_0]$. We have two groups with n individuals, with identical hazard rates $\alpha(t)$.

Their lifetime is independently Weibull distributed,

$$f_T(t; \alpha, \beta) = \alpha\beta t^{\beta-1} e^{-\alpha t^{\beta}},\ t \geq 0, \tag{2.1}$$

with $\alpha = 0.01$ and $\beta = 1.1$. The censoring times are independently exponentially distributed,

$$f_C(c; \lambda) = \lambda e^{-\lambda c},\ c \geq 0, \tag{2.2}$$

with $\lambda = 0.005$. We also assume that the study is terminated at time $t = 225$. So, for individual i, we observe for this individual the right censored survival time,

$$\tilde{T_i} = min\{T_i, C_i, 225\}, \tag{2.3}$$

and the censoring indicator,

$$D_i = \begin{cases} 1 & \text{, if } T_i \leq min\{C_i, 225\}, \\ 0 & \text{, otherwise.} \end{cases} \tag{2.4}$$

a)

Here we will make a function that simulates the situation described in the Introduction. We want the output to be suitable to use in the function we made in Problem 1:b). We will simulate the Weibull and exponential distributions using the probability integral transform method. This method is also called the inversion method.

We first find the Cumulative density function of Equations (2.1) and (2.2),

$$F_T(t;\alpha,\beta) = 1 - e^{-\alpha t^{\beta}},\ t \geq 0, \tag{2.5}$$

$$F_C(c;\lambda) = 1 - e^{-\lambda c},\ c \geq 0. \tag{2.6}$$

Then we find the inverse of Equation (2.5),

$$\begin{aligned} u &= 1 - e^{-\alpha t^{\beta}} \\ e^{-\alpha t^{\beta}} &= 1 - u \\ -\alpha t^{\beta} &= log(1-u) \\ t^{\beta} &= -\frac{1}{\alpha} log(1-u) \\ t &= \sqrt[\beta]{-\frac{1}{\alpha} log(1-u),} \end{aligned}$$

and Equation (2.6),

$$\begin{aligned} u &= 1 - e^{-\lambda c} \\ e^{-\lambda c} &= 1 - u \\ -\lambda c &= log(1-u) \\ c &= -\frac{1}{\lambda} log(1-u), \end{aligned}$$

where $u \sim U[0,1]$.

We then have,

$$F_T^{-1}(u;\alpha,\beta) = \sqrt[\beta]{-\frac{1}{\alpha} log(1-u),} \tag{2.7}$$

and

$$F_C^{-1}(u;\lambda) = -\frac{1}{\lambda} log(1-u). \tag{2.8}$$

```
sim2a <- function(n, alpha = 0.01, beta = 1.1, lambda = 0.005,
    t = 225) {
    logu <- log(1 - runif(n))
    # Calc. survival times from Eq. (2.7)
    TT <- (-(1/alpha) * logu)^(1/beta)
    # Calc. censoring times from Eq. (2.8)
    C <- -(1/lambda) * logu
    tildeTT <- c()
    D <- c()
    for (i in 1:n) {
        # Finding the right censored survival times from
        # Eq. (2.3)
        tildeTT[i] <- min(TT[i], C[i], t)
        # Finding the censoring indicators from Eq. (2.4)
        ifelse(TT[i] <= min(C[i], t), D[i] <- 1, D[i] <- 0)
    }
    # Sorting them in correct order
    tildeT <- sort(tildeTT)
    D <- D[order(tildeTT)]
    return(list(tildeT, D))
}
```

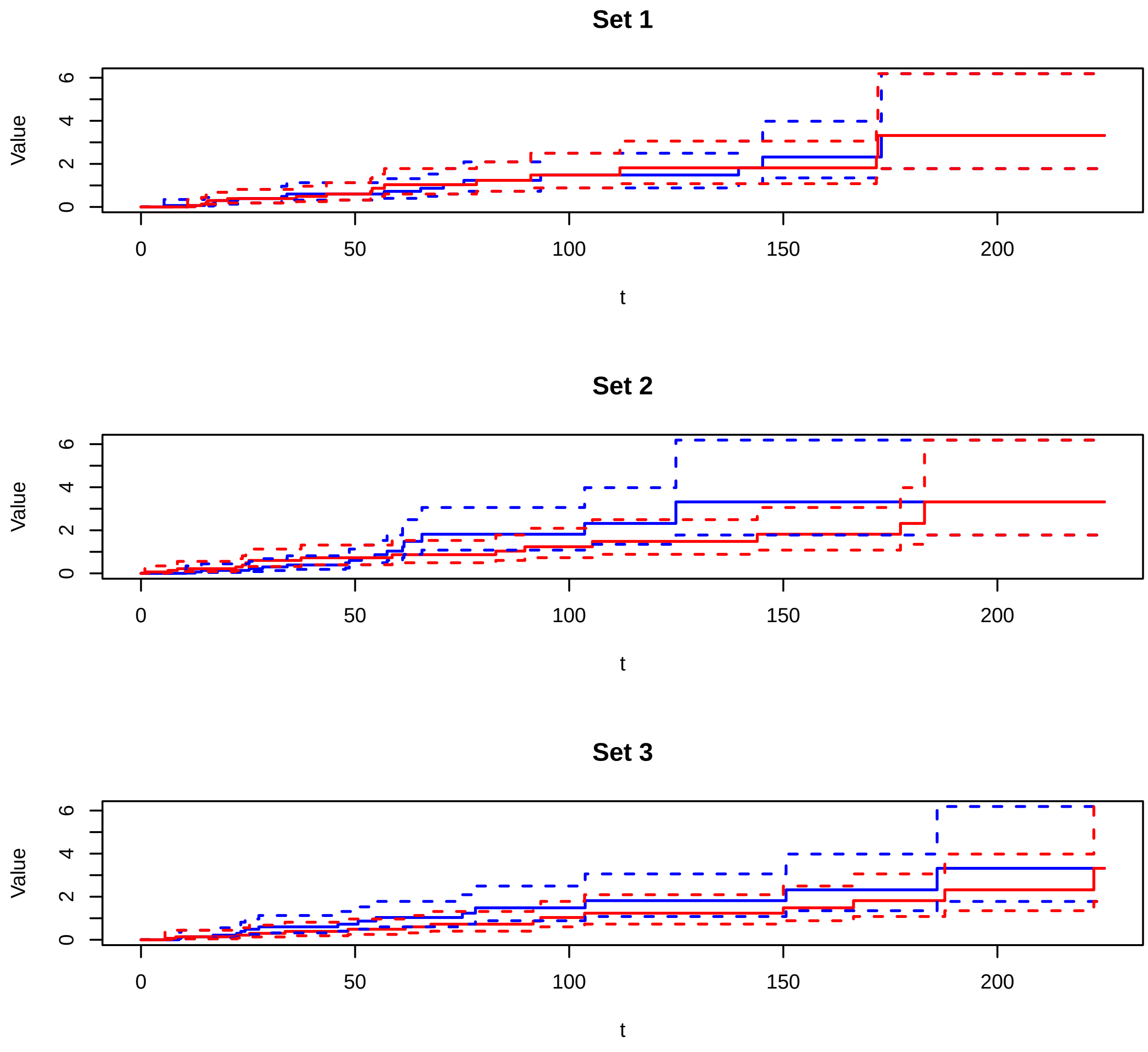
b)

Now we simulate three sets of two groups, and then plot their Nelson-Aalen estimators with confidence interval for all three groups, and after that we will do the same test we did in Problem 1:e), and find the associated p-values.

```
n <- 15
Y1123 <- seq(n, 1)
Y2123 <- seq(n, 1)
# Set 1
set.seed(1)
sim2a_11 <- sim2a(n)
G11 <- sim2a_11[[1]]
d11 <- sim2a_11[[2]]
set.seed(2)
sim2a_21 <- sim2a(n)
G21 <- sim2a_21[[1]]
d21 <- sim2a_21[[2]]
fun21 <- fun1b(Y1123, Y2123, d11, d21)
hatA11 <- fun21[[1]]
hatA21 <- fun21[[2]]
```

```
hatSigma112 <- fun21[[3]]
hatSigma212 <- fun21[[4]]
Nelson_upper_CI_11 <- hatA11 * exp(z * sqrt(hatSigma112)/hatA11)
Nelson_lower_CI_11 <- hatA11 * exp(-z * sqrt(hatSigma112)/hatA11)
Nelson_upper_CI_21 <- hatA21 * exp(z * sqrt(hatSigma212)/hatA21)
Nelson_lower_CI_21 <- hatA21 * exp(-z * sqrt(hatSigma212)/hatA21)
# Set 2
set.seed(3)
sim2a_12 <- sim2a(n)
G12 <- sim2a_12[[1]]
d12 <- sim2a_12[[2]]
set.seed(4)
sim2a_22 <- sim2a(n)
G22 <- sim2a_22[[1]]
d22 <- sim2a_22[[2]]
fun22 <- fun1b(Y1123, Y2123, d12, d22)
hatA12 <- fun22[[1]]
hatA22 <- fun22[[2]]
hatSigma122 <- fun22[[3]]
hatSigma222 <- fun22[[4]]
Nelson_upper_CI_12 <- hatA12 * exp(z * sqrt(hatSigma122)/hatA12)
Nelson_lower_CI_12 <- hatA12 * exp(-z * sqrt(hatSigma122)/hatA12)
Nelson_upper_CI_22 <- hatA22 * exp(z * sqrt(hatSigma222)/hatA22)
Nelson_lower_CI_22 <- hatA22 * exp(-z * sqrt(hatSigma222)/hatA22)
# Set 3
set.seed(5)
sim2a_13 <- sim2a(n)
G13 <- sim2a_13[[1]]
d13 <- sim2a_13[[2]]
set.seed(6)
sim2a_23 <- sim2a(n)
G23 <- sim2a_23[[1]]
d23 <- sim2a_23[[2]]
fun23 <- fun1b(Y1123, Y2123, d13, d23)
hatA13 <- fun23[[1]]
hatA23 <- fun23[[2]]
hatSigma132 <- fun23[[3]]
hatSigma232 <- fun23[[4]]
Nelson_upper_CI_13 <- hatA13 * exp(z * sqrt(hatSigma132)/hatA13)
Nelson_lower_CI_13 <- hatA13 * exp(-z * sqrt(hatSigma132)/hatA13)
Nelson_upper_CI_23 <- hatA23 * exp(z * sqrt(hatSigma232)/hatA23)
Nelson_lower_CI_23 <- hatA23 * exp(-z * sqrt(hatSigma232)/hatA23)
```

```
par(mfrow = c(3, 1))
G11_Nelson <- append(append(0, G11), 225)
G21_Nelson <- append(append(0, G21), 225)
plot(G11_Nelson, append(append(0, hatA11), tail(hatA11, 1)),
    lwd = 1.5, type = "s", col = "blue", main = "Set 1", xlab = latex2exp("$t$"),
    ylab = "Value", xlim = c(0, 225), ylim = c(0, max(Nelson_upper_CI_11,
        Nelson_upper_CI_21)))
lines(G21_Nelson, append(append(0, hatA21), tail(hatA21, 1)),
    lwd = 1.5, type = "s", col = "red")
lines(G11_Nelson, append(append(0, Nelson_upper_CI_11), tail(Nelson_upper_CI_11,
    1)), lwd = 1.5, lty = 2, type = "s", col = "blue")
lines(G11_Nelson, append(append(0, Nelson_lower_CI_11), tail(Nelson_lower_CI_11,
    1)), lwd = 1.5, lty = 2, type = "s", col = "blue")
lines(G21_Nelson, append(append(0, Nelson_upper_CI_21), tail(Nelson_upper_CI_21,
    1)), lwd = 1.5, lty = 2, type = "s", col = "red")
lines(G21_Nelson, append(append(0, Nelson_lower_CI_21), tail(Nelson_lower_CI_21,
    1)), lwd = 1.5, lty = 2, type = "s", col = "red")
G12_Nelson <- append(append(0, G12), 225)
G22_Nelson <- append(append(0, G22), 225)
plot(G12_Nelson, append(append(0, hatA12), tail(hatA12, 1)),
    lwd = 1.5, type = "s", col = "blue", main = "Set 2", xlab = latex2exp("$t$"),
    ylab = "Value", xlim = c(0, 225), ylim = c(0, max(Nelson_upper_CI_12,
        Nelson_upper_CI_22)))
lines(G22_Nelson, append(append(0, hatA22), tail(hatA22, 1)),
    lwd = 1.5, type = "s", col = "red")
lines(G12_Nelson, append(append(0, Nelson_upper_CI_12), tail(Nelson_upper_CI_12,
    1)), lwd = 1.5, lty = 2, type = "s", col = "blue")
lines(G12_Nelson, append(append(0, Nelson_lower_CI_12), tail(Nelson_lower_CI_12,
    1)), lwd = 1.5, lty = 2, type = "s", col = "blue")
lines(G22_Nelson, append(append(0, Nelson_upper_CI_22), tail(Nelson_upper_CI_22,
    1)), lwd = 1.5, lty = 2, type = "s", col = "red")
lines(G22_Nelson, append(append(0, Nelson_lower_CI_22), tail(Nelson_lower_CI_22,
    1)), lwd = 1.5, lty = 2, type = "s", col = "red")
G13_Nelson <- append(append(0, G13), 225)
G23_Nelson <- append(append(0, G23), 225)
plot(G13_Nelson, append(append(0, hatA13), tail(hatA13, 1)),
    lwd = 1.5, type = "s", col = "blue", main = "Set 3", xlab = latex2exp("$t$"),
    ylab = "Value", xlim = c(0, 225), ylim = c(0, max(Nelson_upper_CI_13,
        Nelson_upper_CI_23)))
lines(G23_Nelson, append(append(0, hatA23), tail(hatA23, 1)),
    lwd = 1.5, type = "s", col = "red")
lines(G13_Nelson, append(append(0, Nelson_upper_CI_13), tail(Nelson_upper_CI_13,
    1)), lwd = 1.5, lty = 2, type = "s", col = "blue")
lines(G13_Nelson, append(append(0, Nelson_lower_CI_13), tail(Nelson_lower_CI_13,
    1)), lwd = 1.5, lty = 2, type = "s", col = "blue")
lines(G23_Nelson, append(append(0, Nelson_upper_CI_23), tail(Nelson_upper_CI_23,
    1)), lwd = 1.5, lty = 2, type = "s", col = "red")
lines(G23_Nelson, append(append(0, Nelson_lower_CI_23), tail(Nelson_lower_CI_23,
    1)), lwd = 1.5, lty = 2, type = "s", col = "red")
```



Set 1

Set 2

Set 3

Figure 6: Nelson-Aalen estimators for all three sets, with both groups, together with the 90 percen confidence interval.

```
fun2b1 <- fun1e(Y1123, Y2123, G11, G21, d11, d21)
fun2b2 <- fun1e(Y1123, Y2123, G12, G22, d12, d22)
fun2b3 <- fun1e(Y1123, Y2123, G13, G23, d13, d23)
data.frame(U0 = c(fun2b1[[1]], fun2b2[[1]], fun2b3[[1]]), `p-value` = c(fun2b1[[2]],
    fun2b2[[2]], fun2b3[[2]]), row.names = c("Set 1", "Set 2",
    "Set 3"))
```

```
##                U0   p.value
## Set 1 -0.2681357 0.7885949
## Set 2  0.8782774 0.3797932
## Set 3  1.2724256 0.2032219
```

Since we simulated all the sets with the same hazard rate, these three tests also indicate that their hazard rate is similar. Looking at Problem 1:b) we also see that the three sets in Figure 6, are pretty similar to each other, while the set in Problem 1:b), Figure 2, are pretty different. Also, the p-value we find in this problem is much larger that what we found in Problem 1:e).

c)

Now we will, for $n = 15$, make $M = 1000$ data sets from the situation above, use a Q-Q plot to evaluate the normal distribution of our test statistic, and then later make a histogram of the associated p-values. We will then discuss how good the normal approximation of our test statistic is for this amount of data.

```
M <- 1000
U0c <- c()
p_val <- c()
for (i in 1:M) {
    Y1123 <- seq(n, 1)
    Y2123 <- seq(n, 1)
    set.seed(i + 1000)
    sim2a_1123 <- sim2a(n)
    G1123 <- sim2a_1123[[1]]
    d1123 <- sim2a_1123[[2]]
    set.seed(i + 1000 + M)
    sim2a_2123 <- sim2a(n)
    G2123 <- sim2a_2123[[1]]
    d2123 <- sim2a_2123[[2]]
    fun2c <- fun1e(Y1123, Y2123, G1123, G2123, d1123, d2123)
    U0c <- append(U0c, fun2c[[1]])
    p_val <- append(p_val, fun2c[[2]])
}
```

```
qqnorm(U0c)
abline(v = 0, lwd = 1.5, col = "blue")
abline(h = 0, lwd = 1.5, col = "blue")
abline(a = 0, b = 1, lwd = 3, col = "red")
```

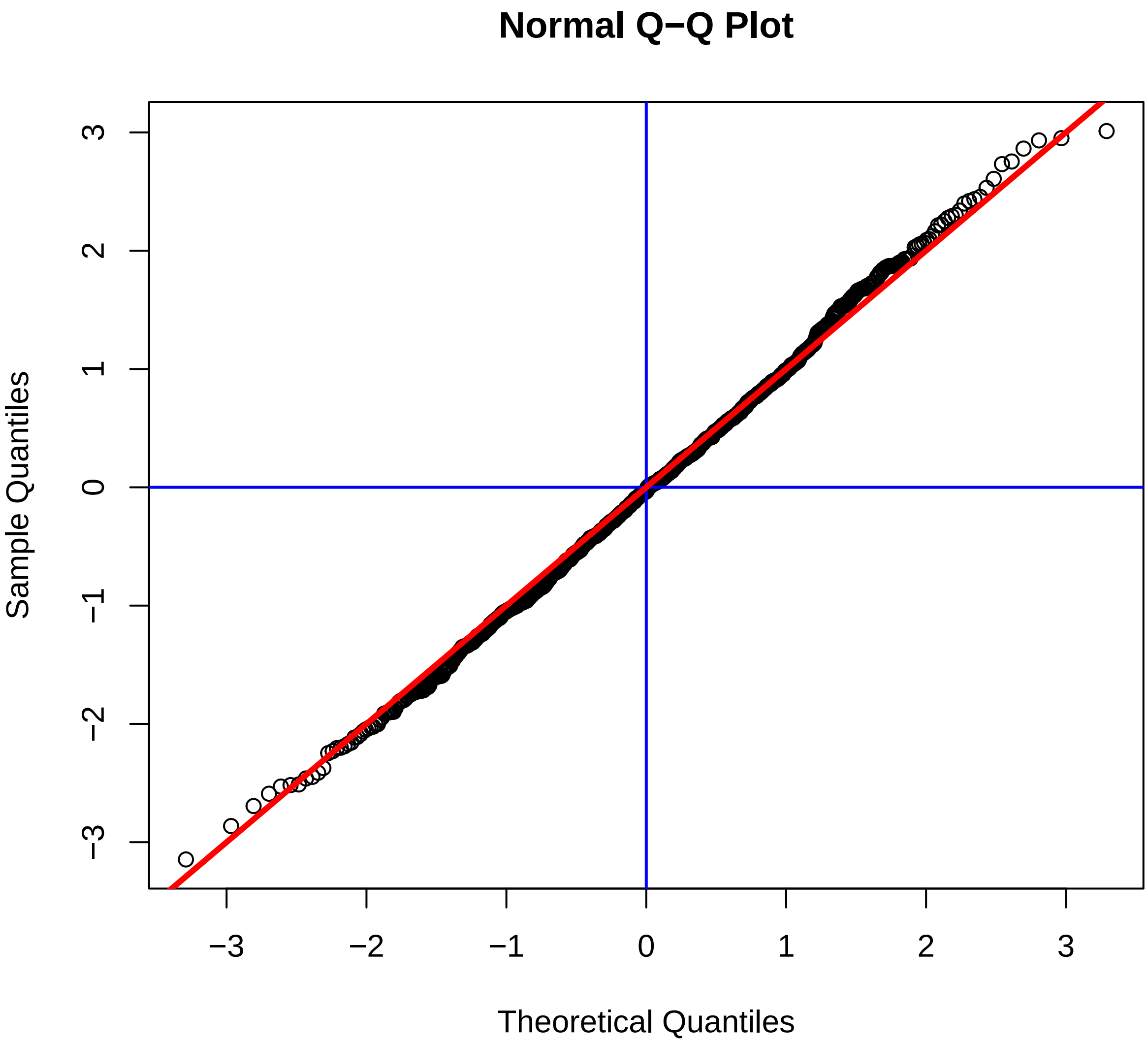


Figure 7: Q-Q plot of the test statistics we have calculated.

```
set.seed(98)
snd <- rnorm(M)
p_val_snd <- 0
for (i in 1:length(pnorm(rnorm(M)))) {
    p_val_snd[i] <- 2 * min(pnorm(snd, lower.tail = F)[i], pnorm(snd,
        lower.tail = T)[i])
}
par(mfrow = c(2, 1))
hist(p_val, freq = F, col = "darkgray", lwd = 4, breaks = 100,
    xlim = c(0, 1), main = "Histogram of our p-values", xlab = "p-values")
hist(p_val_snd, freq = F, col = "darkgray", lwd = 4, breaks = 100,
    xlim = c(0, 1), main = "Histogram of the p-values from known standard normal distr.n
    xlab = "p-values")
```

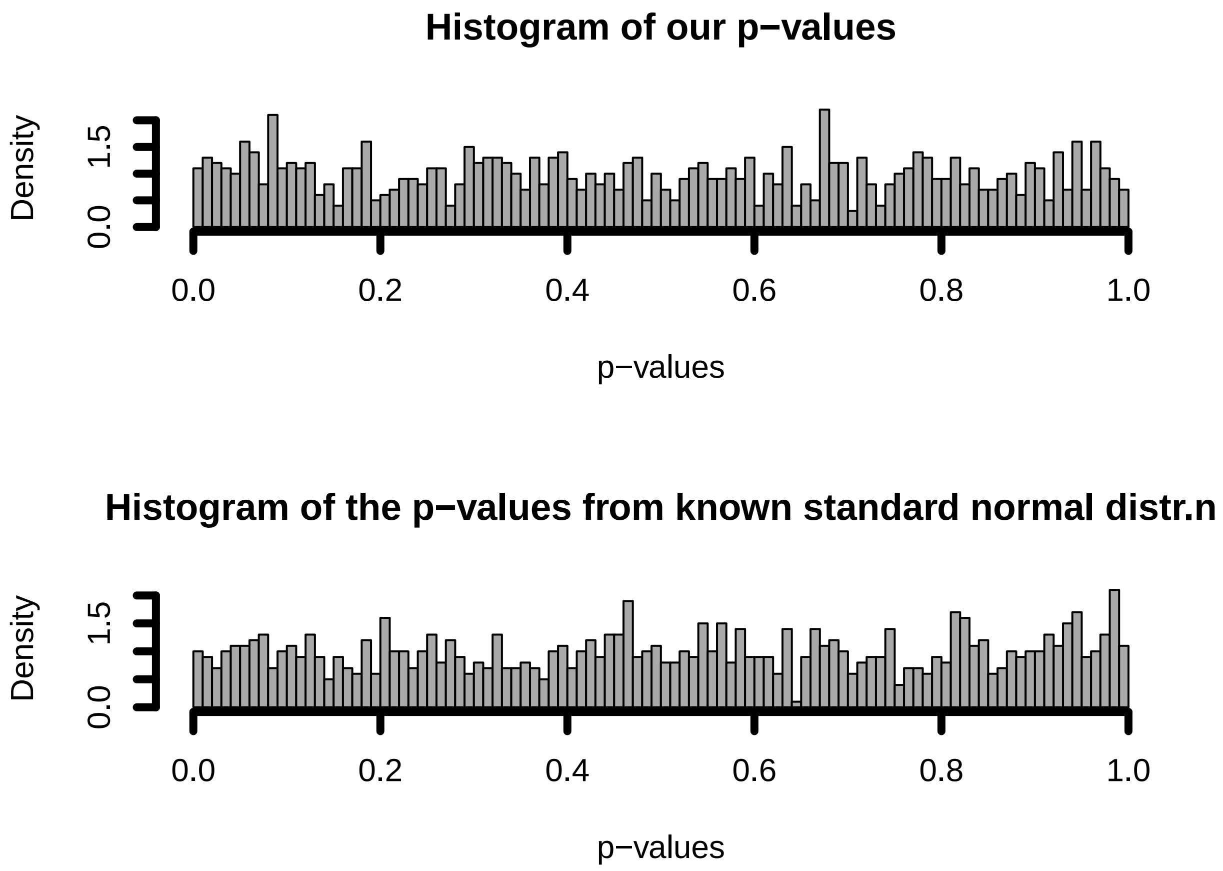


Figure 8: Histogram of our p-values.

The points in the Q-Q plot in Figure 7 seem fall in a straight line, so the normal approximation holds up here. Looking at the histograms of our p-values in Figure 8, we see that they look pretty similar, so the standard normal approximation also seem to hold up here.

d)

To end this problem, we will find out how low n can be until our standard normal distribution assumption break.

```
n <- 5
M <- 1000
U0c <- c()
p_val <- c()
for (i in 1:M) {
    Y1123 <- seq(n, 1)
    Y2123 <- seq(n, 1)
    set.seed(i + 1000)
    sim2a_1123 <- sim2a(n)
    G1123 <- sim2a_1123[[1]]
    d1123 <- sim2a_1123[[2]]
    set.seed(i + 1000 + M)
    sim2a_2123 <- sim2a(n)
    G2123 <- sim2a_2123[[1]]
    d2123 <- sim2a_2123[[2]]
    fun2c <- fun1e(Y1123, Y2123, G1123, G2123, d1123, d2123)
    U0c <- append(U0c, fun2c[[1]])
    p_val <- append(p_val, fun2c[[2]])
}
qqnorm(U0c, main = paste("Normal Q-Q Plot, w. n = ", n))
abline(v = 0, lwd = 1.5, col = "blue")
abline(h = 0, lwd = 1.5, col = "blue")
abline(a = 0, b = 1, lwd = 3, col = "red")
```

```
set.seed(98)
snd <- rnorm(M)
p_val_snd <- 0
for (i in 1:length(pnorm(rnorm(M)))) {
    p_val_snd[i] <- 2 * min(pnorm(snd, lower.tail = F)[i], pnorm(snd,
        lower.tail = T)[i])
}
par(mfrow = c(2, 1))
hist(p_val, freq = F, col = "darkgray", lwd = 4, breaks = 100,
    xlim = c(0, 1), main = paste("Histogram of our p-values, w. n = ",
```

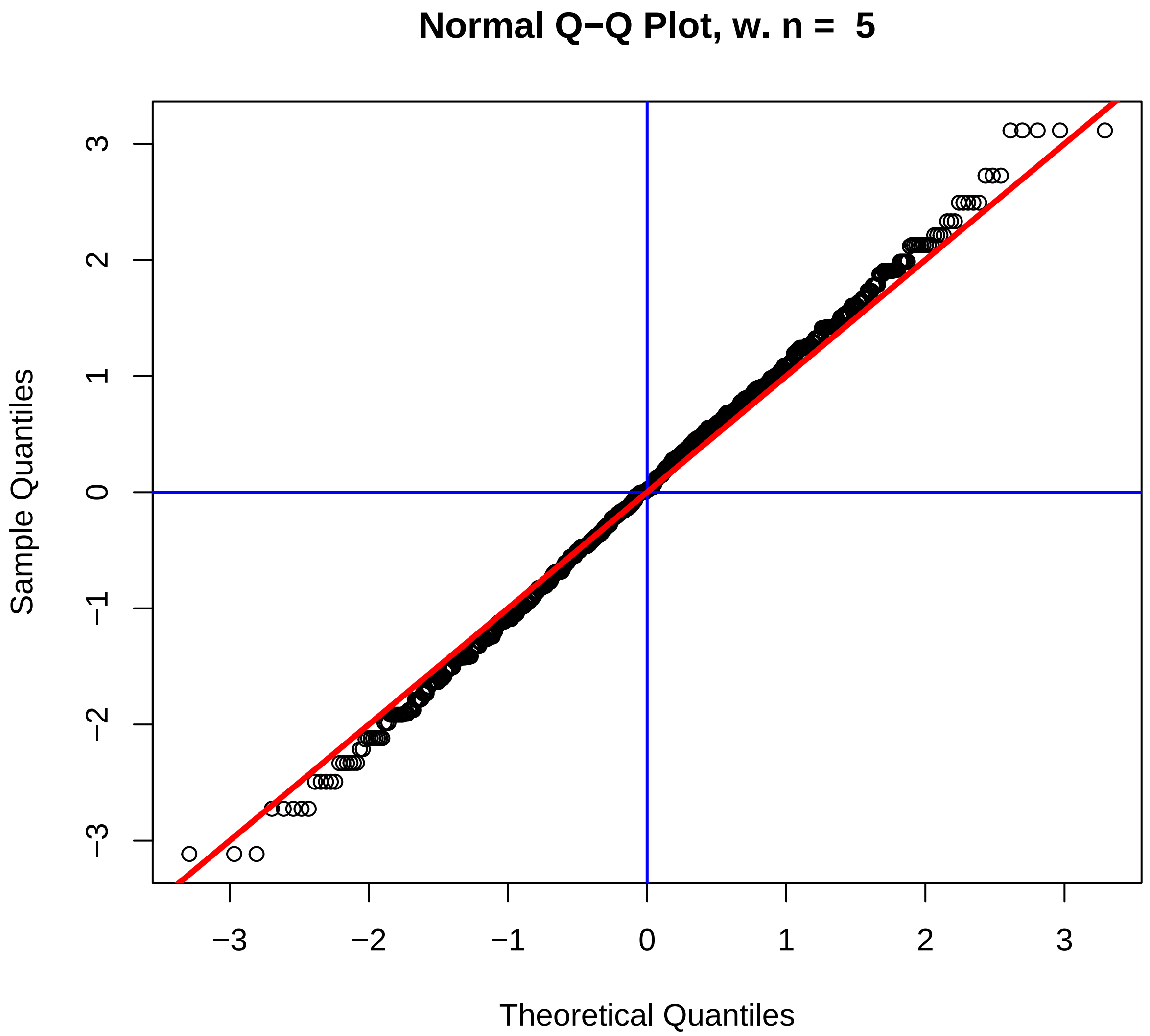


Figure 9: Q-Q plot of the test statistics we have calculated, w. n = 5.

```
         n), xlab = "p-values")
hist(p_val_snd, freq = F, col = "darkgray", lwd = 4, breaks = 100,
    xlim = c(0, 1), main = "Histogram of the p-values from known standard normal distr.n
    xlab = "p-values")
```

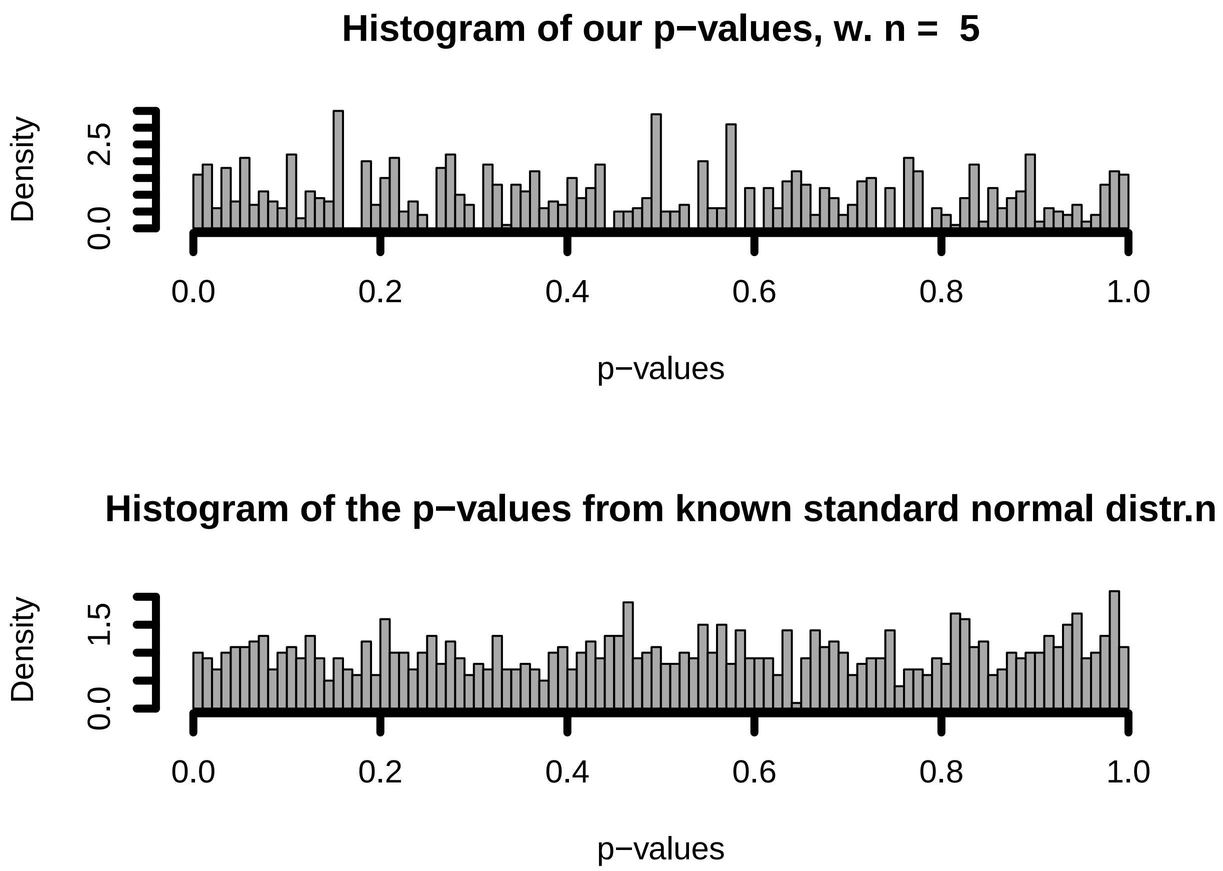


Figure 10: Histogram of our p-values, w. n = 5.

We see now with $n = 5$ that the standard normal distribution seem to break. In Figure 9 we see that the line sags off at the ends, so it isn't standard normal distributed anymore, and in Figure 10 we see that the p-values are no longer uniformly distributed. Testing with $n = 6$ we get:

```r
n <- 6
M <- 1000
U0c <- c()
p_val <- c()
for (i in 1:M) {
    Y1123 <- seq(n, 1)
    Y2123 <- seq(n, 1)
    set.seed(i + 1000)
    sim2a_1123 <- sim2a(n)
    G1123 <- sim2a_1123[[1]]
    d1123 <- sim2a_1123[[2]]
    set.seed(i + 1000 + M)
    sim2a_2123 <- sim2a(n)
    G2123 <- sim2a_2123[[1]]
    d2123 <- sim2a_2123[[2]]
    fun2c <- fun1e(Y1123, Y2123, G1123, G2123, d1123, d2123)
    U0c <- append(U0c, fun2c[[1]])
    p_val <- append(p_val, fun2c[[2]])
}
qqnorm(U0c, main = paste("Normal Q-Q Plot, w. n = ", n))
abline(v = 0, lwd = 1.5, col = "blue")
abline(h = 0, lwd = 1.5, col = "blue")
abline(a = 0, b = 1, lwd = 3, col = "red")
```

```r
set.seed(98)
snd <- rnorm(M)
p_val_snd <- 0
for (i in 1:length(pnorm(rnorm(M)))) {
    p_val_snd[i] <- 2 * min(pnorm(snd, lower.tail = F)[i], pnorm(snd,
        lower.tail = T)[i])
}
par(mfrow = c(2, 1))
hist(p_val, freq = F, col = "darkgray", lwd = 4, breaks = 100,
    xlim = c(0, 1), main = paste("Histogram of our p-values, w. n = ",
        n), xlab = "p-values")
hist(p_val_snd, freq = F, col = "darkgray", lwd = 4, breaks = 100,
    xlim = c(0, 1), main = "Histogram of the p-values from known standard normal distr.n
    xlab = "p-values")
```

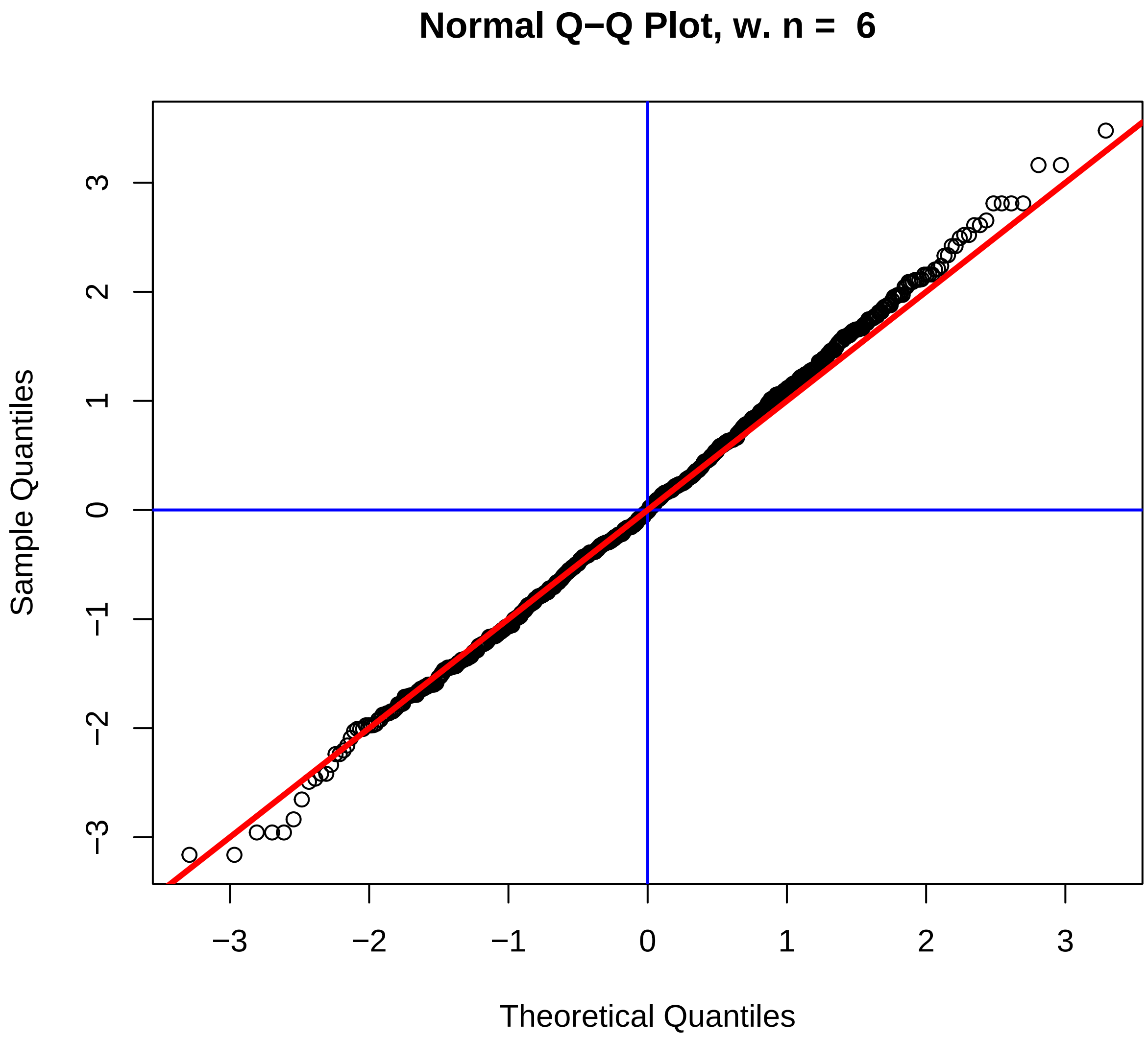


Figure 11: Q-Q plot of the test statistics we have calculated, w. n = 6.

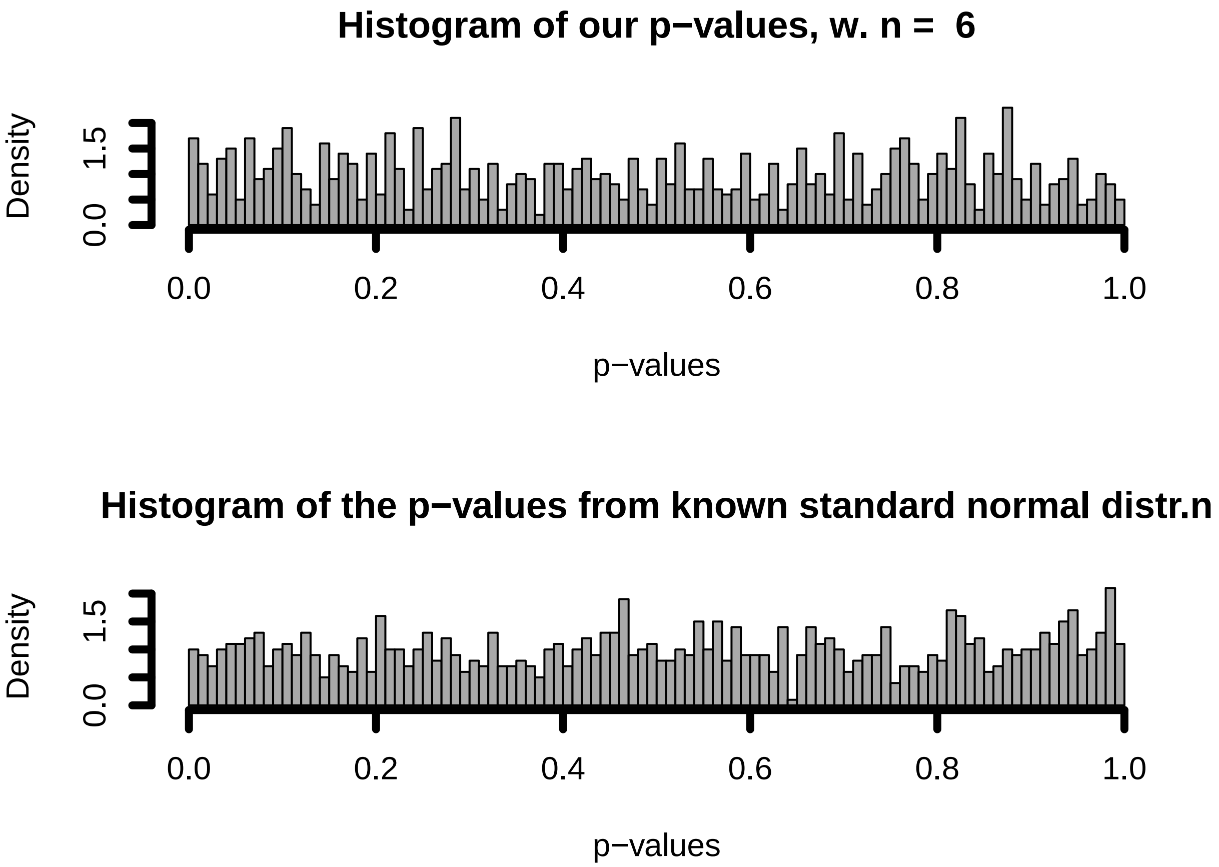


Figure 12: Histogram of our p-values, w. n = 6.

We see in Figure 11 and Figure 12 that the Q-Q plot follows the line better, and that the p-values are closer to a uniform distribution than in Figure 10. So $n = 6$ is the lowest number of survival times we should have.