M215_HW7

```
library(survival)
library(KMsurv)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#install.packages('zoo')
library(zoo)
```

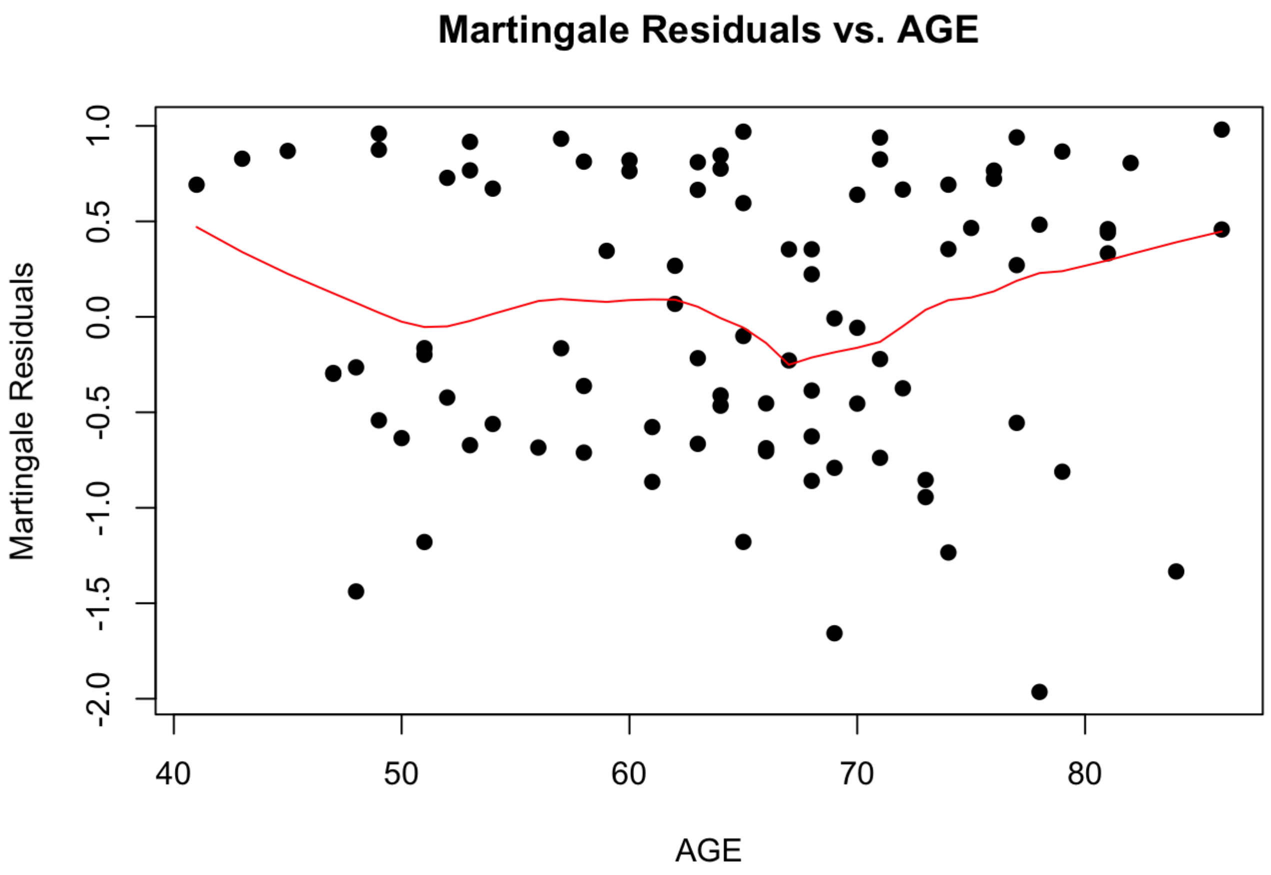
```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

11.1

a.

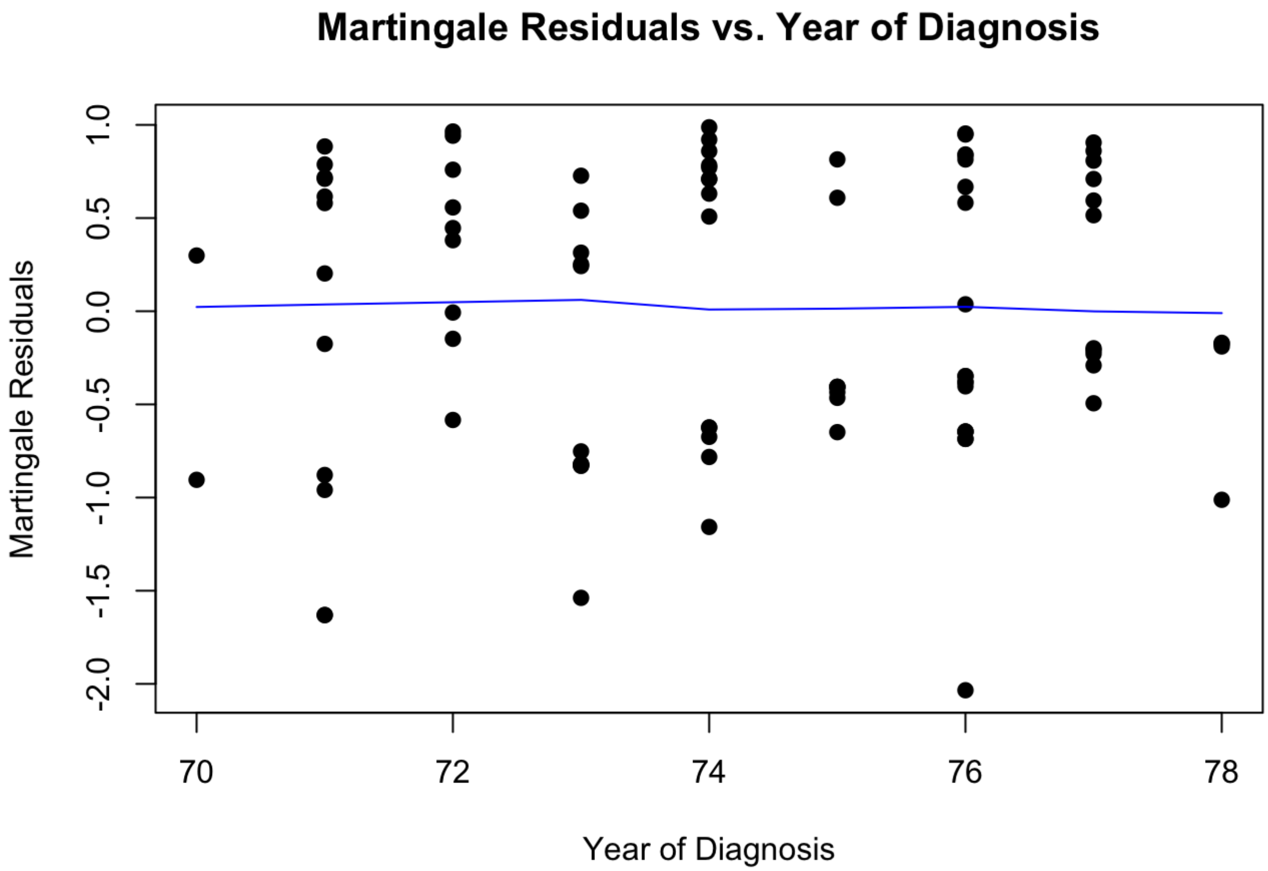
```
help("larynx")
data(larynx)

#create dummy variables
larynx$s2 <- ifelse(larynx$stage == 2, 1, 0)
larynx$s3 <- ifelse(larynx$stage == 3, 1, 0)
larynx$s4 <- ifelse(larynx$stage == 4, 1, 0)

#### a ####
# Fit the model
#cut.points_la <- unique(larynx$time[larynx$delta == 1])
#larynx1 <- survSplit(data = larynx, cut = cut.points_la, end = "time", start = "t0", event = "delta")
# fit.larynx <- coxph(Surv(t0, time, delta) ~ s2 + s3 + s4, data = larynx1, ties = 'breslow')

fit.larynx <- coxph(Surv(time,delta) ~ age + factor(stage), data = larynx, ties = 'breslow')

#Get Cox-Snell residual based on Martingale residuals
mg.residual <- resid(fit.larynx, type = "martingale")

plot(mg.residual ~ larynx$age, xlab = "AGE", ylab = "Martingale Residuals",
     main='Martingale Residuals vs. AGE', pch = 19)
lines(lowess(larynx$age, mg.residual, f = 0.35), col = 'red')
```



- The redisual plot showed that model might be ** threshod model or quadratic model **.

b.

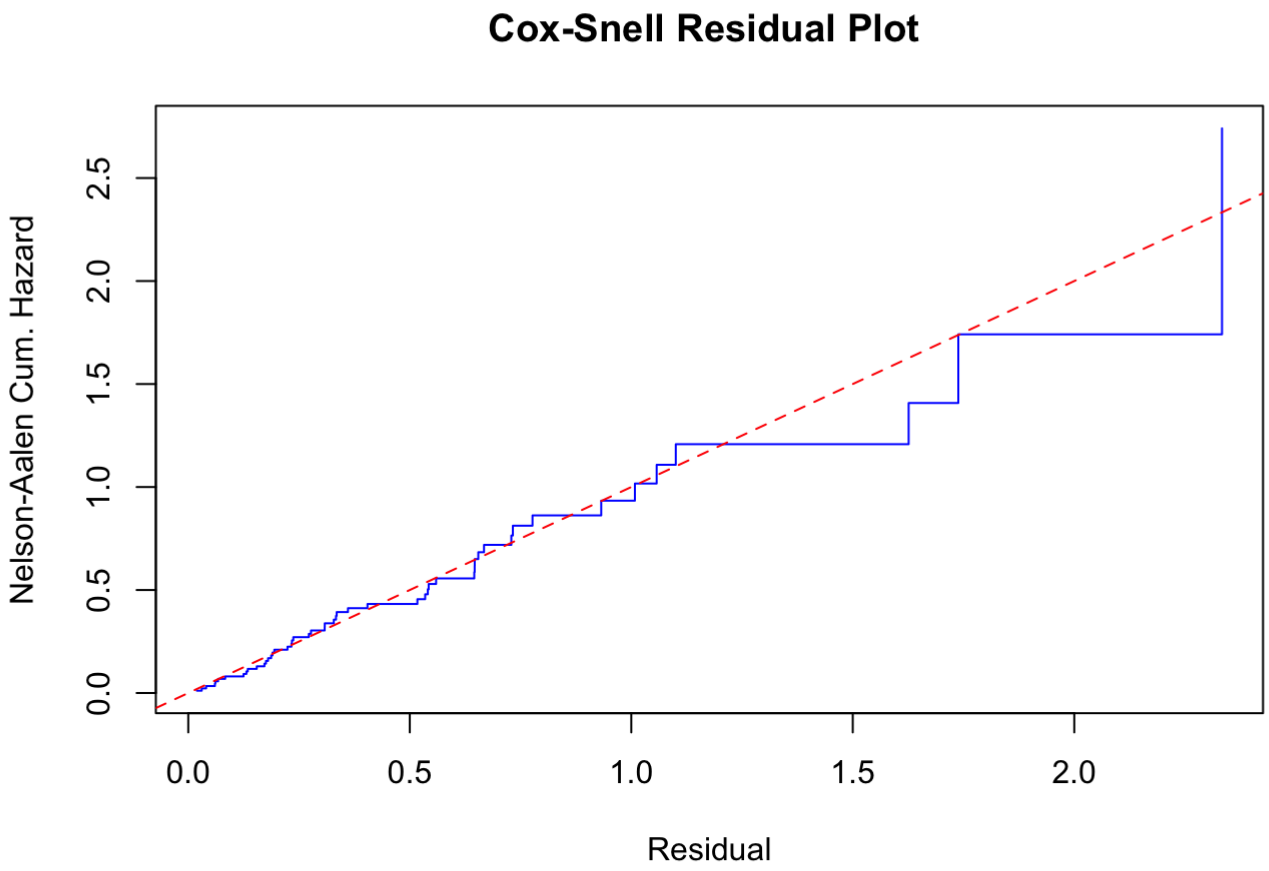
```
#### b ####
fit.larynx1 <- coxph(Surv(time,delta) ~ diagyr + factor(stage), data = larynx, ties = 'breslow')

#Get Cox-Snell residual based on Martingale residuals
mg.residual1 <- resid(fit.larynx1, type = "martingale")

plot(mg.residual1 ~ larynx$diagyr, xlab = "Year of Diagnosis",
     ylab = "Martingale Residuals",
     main='Martingale Residuals vs. Year of Diagnosis', pch = 19)
lines(lowess(larynx$diagyr, mg.residual1), col = 'blue')
```



- The plot showed that regression coeffient is ** not significantly different from 0 **.

c.

```
#### c ####
mg.residual2 <- resid(fit.larynx, type = "martingale")
cs.residual <- larynx$delta - mg.residual2

#Graphical Plot
fit.cs <- survfit(Surv(cs.residual, larynx$delta) ~ 1) #Get Kaplan-Meier estiamtes
H.cs <- cumsum(fit.cs$n.event/fit.cs$n.risk)
plot(fit.cs$time, H.cs, type='s', col='blue', main = 'Cox-Snell Residual Plot ', xlab = 'Residual', ylab = 'Nelson-Aalen Cum. Hazard') #Note here that 'time' is the value of the Cox-Snell residual
abline(0, 1, col='red', lty=2)
```



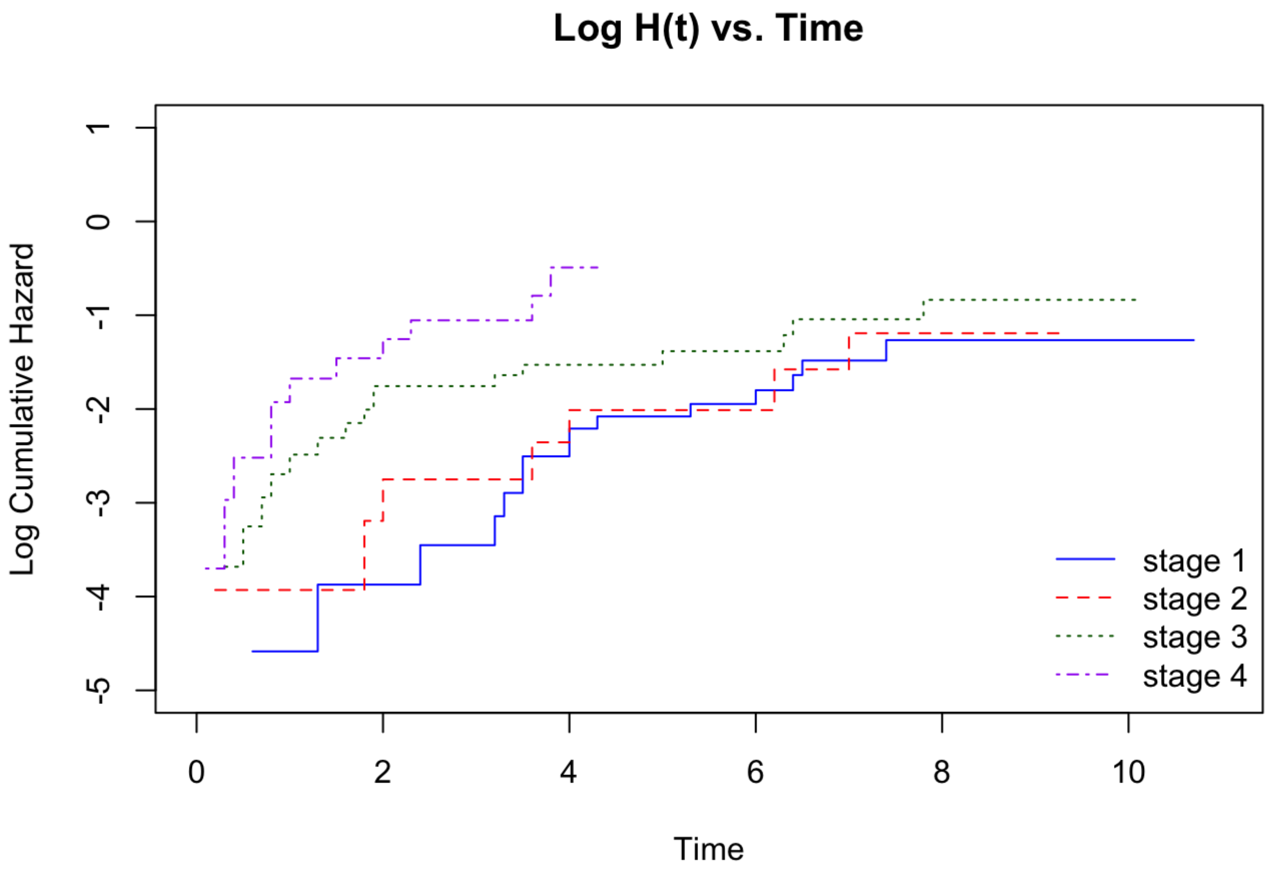
- The model seems fit well.

11.3

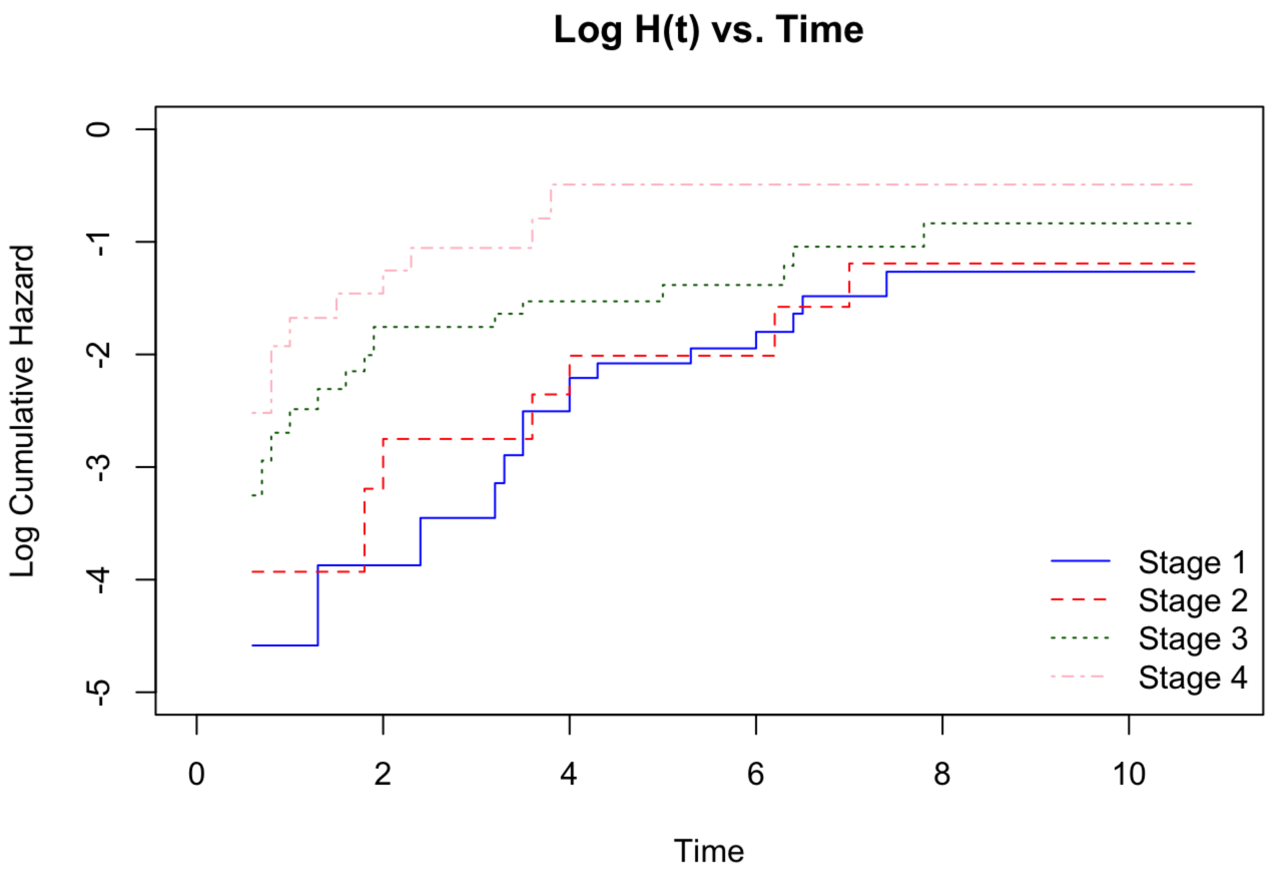
a.

- I did two version plot. Second one looks better.

```
### a ###
# stratify on stage and plot the log estimated baseline cumulative hazard rates for each
 strata against time

fit.larynx2 <- basehaz(coxph(Surv(time,delta) ~ age + strata(factor(stage)),
                             data = larynx, ties = 'breslow'), centered = F)

plot(log(fit.larynx2$hazard[fit.larynx2$strata == 1]) ~
       fit.larynx2$time[fit.larynx2$strata == 1],  type = 's',
     ylab = 'Log Cumulative Hazard', xlab = 'Time',  main = 'Log H(t) vs. Time',
     col = 'blue', lty = 1, xlim = c(0, 11), ylim = c(-5, 1))
lines(log(fit.larynx2$hazard[fit.larynx2$strata == 2]) ~
        fit.larynx2$time[fit.larynx2$strata == 2],  col = 'red',
      lty = 2, type = 's')
lines(log(fit.larynx2$hazard[fit.larynx2$strata == 3]) ~
        fit.larynx2$time[fit.larynx2$strata == 3],  col = 'darkgreen',
      lty = 3, type = 's')
lines(log(fit.larynx2$hazard[fit.larynx2$strata == 4]) ~
        fit.larynx2$time[fit.larynx2$strata == 4],  col = 'purple', lty = 4, type = 's')

legend('bottomright', c('stage 1', 'stage 2','stage 3', 'stage 4'),
       col = c('blue', 'red', 'darkgreen', 'purple'),  lty = c(1, 2, 3, 4), bty = 'n')
```



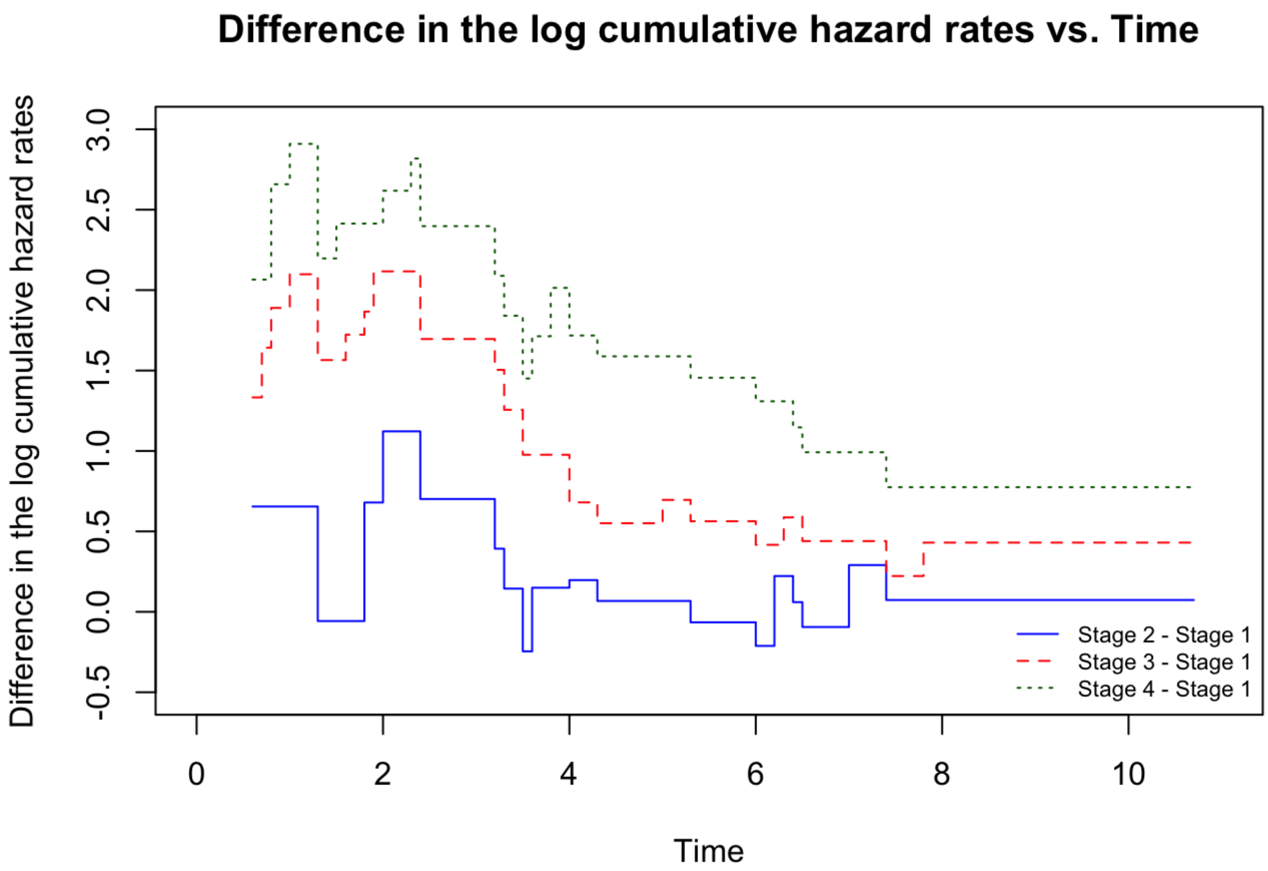
```
### a (new version according to updated lab) ###
s1 <- data.frame("H1" = fit.larynx2$hazard[fit.larynx2$strata == 1],
                 "time" = fit.larynx2$time[fit.larynx2$strata == 1])
s2 <- data.frame("H2" = fit.larynx2$hazard[fit.larynx2$strata == 2],
                 "time" = fit.larynx2$time[fit.larynx2$strata == 2])
s3 <- data.frame("H3" = fit.larynx2$hazard[fit.larynx2$strata == 3],
                 "time" = fit.larynx2$time[fit.larynx2$strata == 3])
s4 <- data.frame("H4" = fit.larynx2$hazard[fit.larynx2$strata == 4],
                 "time" = fit.larynx2$time[fit.larynx2$strata == 4])



#Merge data and impute using na.locf (Thanks Emilie!)
impute.dat <- full_join(s1, s2, by = "time") %>%
  full_join(., s3, by = "time") %>%
  full_join(., s4, by = "time") %>%
  arrange(time) %>%
  do(na.locf(.))

# Plot
plot(log(impute.dat$H1) ~ impute.dat$time, type = 's', ylab = 'Log Cumulative Hazard',
     xlab = 'Time', main = 'Log H(t) vs. Time', col = 'blue', lty = 1,
     xlim = c(0, 11), ylim = c(-5, 0))
lines(log(impute.dat$H2) ~ impute.dat$time, col = 'red', lty = 2, type = 's')
lines(log(impute.dat$H3) ~ impute.dat$time, col = 'darkgreen', lty = 3, type = 's')
lines(log(impute.dat$H4) ~ impute.dat$time, col = 'pink', lty = 4, type = 's')
legend('bottomright', c('Stage 1', 'Stage 2', 'Stage 3', 'Stage 4'),
       col = c('blue', 'red', 'darkgreen', 'pink'), lty = c(1, 2, 3, 4), bty = 'n')
```



b.

Using the data gotten from part a to take difference and got the plot below

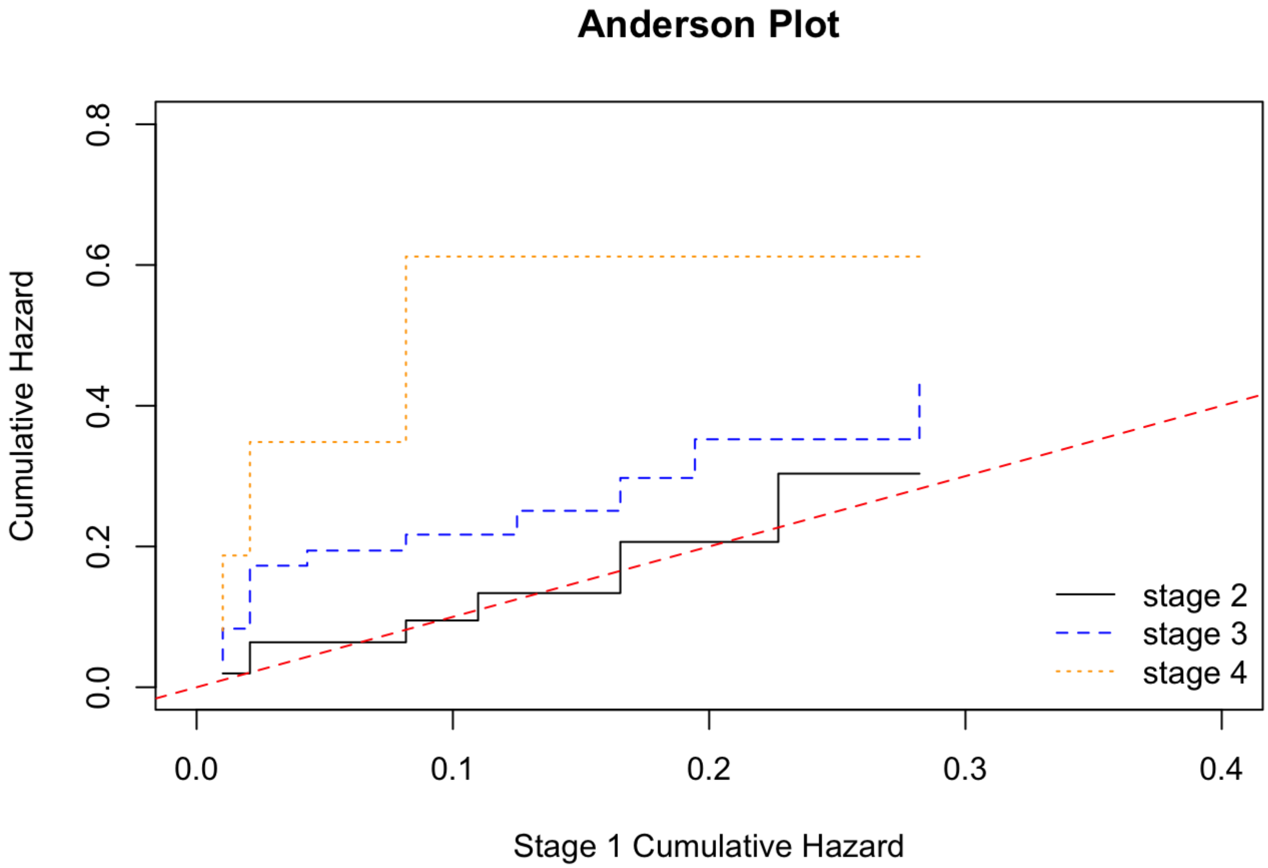
```
plot((log(impute.dat$H2) - log(impute.dat$H1)) ~ impute.dat$time, type = 's',
     ylab = 'Difference in the log cumulative hazard rates', xlab = 'Time',
     main = 'Difference in the log cumulative hazard rates vs. Time',
     col = 'blue', lty = 1, xlim = c(0, 11), ylim = c(-0.5, 3))
lines((log(impute.dat$H3) - log(impute.dat$H1)) ~ impute.dat$time,
      col = 'red', lty = 2, type = 's')
lines((log(impute.dat$H4) - log(impute.dat$H1)) ~ impute.dat$time,
      col = 'darkgreen', lty = 3, type = 's')

legend('bottomright', c('Stage 2 - Stage 1', 'Stage 3 - Stage 1', 'Stage 4 - Stage 1'),
       col = c('blue', 'red', 'darkgreen'), lty = c(1, 2, 3),
       bty = 'n', cex = .7)
```



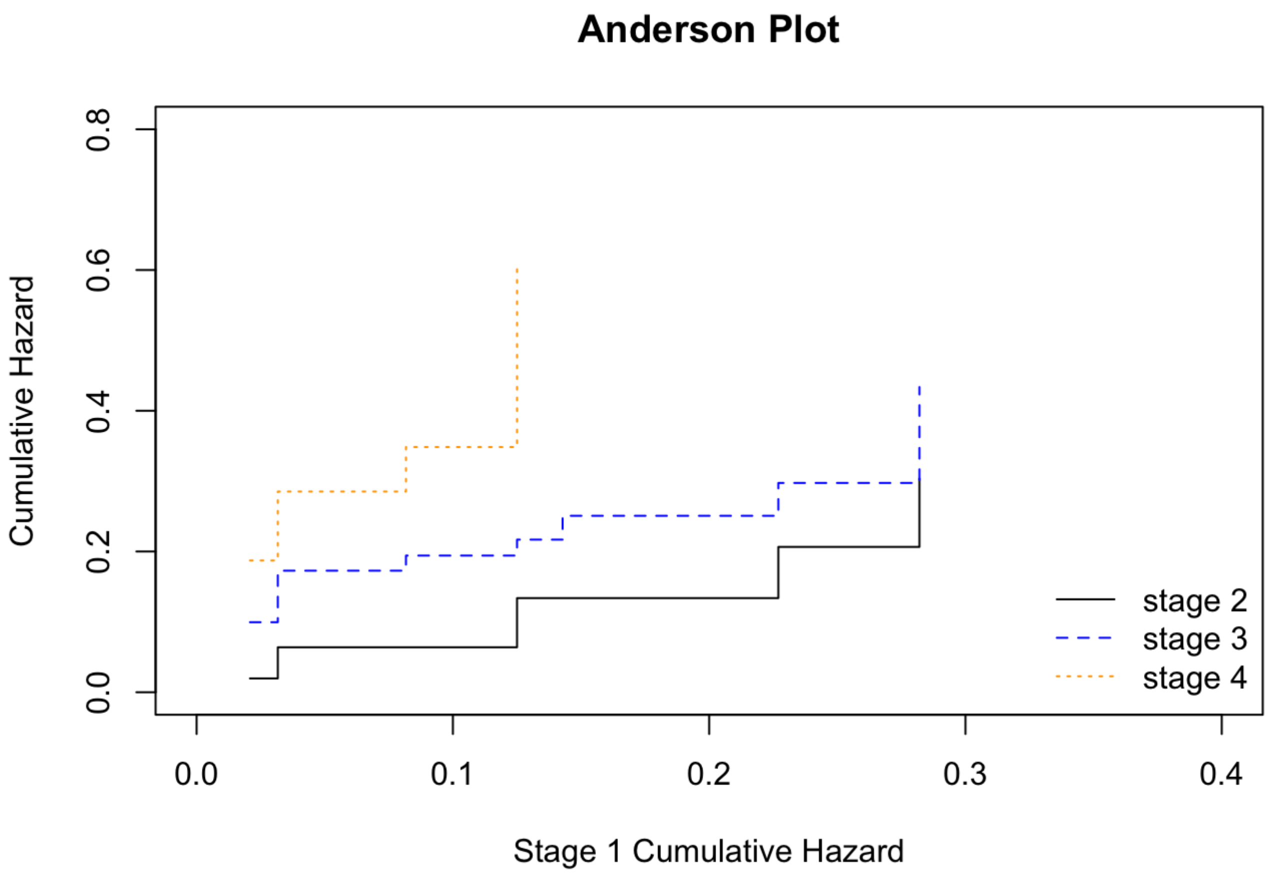
c.

- Also have two versions of Anderson plots.

```
plot(impute.dat$H2 ~ impute.dat$H1, main = 'Anderson Plot', ylab = 'Cumulative Hazard',
     xlab = 'Stage 1 Cumulative Hazard', type = 's', xlim = c(0, 0.4), ylim = c(0, 0.8))
lines(impute.dat$H3 ~ impute.dat$H1, col = 'blue', lty = 2, type = 's')
lines(impute.dat$H4 ~ impute.dat$H1, col = 'orange', lty = 3, type = 's')
abline(0, 1, col='red', lty=2)

legend('bottomright', c('stage 2','stage 3', 'stage 4'),
       col = c('black', 'blue', 'orange'),  lty = c(1, 2, 3), bty = 'n')
```


Anderson Plot
Cumulative Hazard
0.0
0.2
0.4
0.6
0.8
0.0
0.1
0.2
0.3
0.4
Stage 1 Cumulative Hazard
stage 2
stage 3
stage 4

```
H1 <- fit.larynx2$hazard[fit.larynx2$strata == 1]
H2 <- fit.larynx2$hazard[fit.larynx2$strata == 2]
H3 <- fit.larynx2$hazard[fit.larynx2$strata == 3]
H4 <- fit.larynx2$hazard[fit.larynx2$strata == 4]
t1 <- fit.larynx2$time[fit.larynx2$strata == 1]
t2 <- fit.larynx2$time[fit.larynx2$strata == 2]
t3 <- fit.larynx2$time[fit.larynx2$strata == 3]
t4 <- fit.larynx2$time[fit.larynx2$strata == 4]


reptime <- function(l, t){
  x <- numeric(max(t))
  for(i in min(t):max(t)){
    diff <- i - t
    diff <- diff[diff >= 0]
    x[i] <- l[which.min(diff)]
    }
  return(x)
}

H_1 <- reptime(H1, t1)
H_2 <- reptime(H2, t2)
H_3 <- reptime(H3, t3)
H_4 <- reptime(H4, t4)

plot(H_2[1:10] ~ H_1[1:10], main = 'Anderson Plot', ylab = 'Cumulative Hazard',
     xlab = 'Stage 1 Cumulative Hazard', type = 's', xlim = c(0, 0.4), ylim = c(0, 0.8))
lines(H_3[1:9] ~ H_1[1:9], col = 'blue', lty = 2, type = 's')
lines(H_4[1:10] ~ H_1[1:10], col = 'orange', lty = 3, type = 's')


legend('bottomright', c('stage 2','stage 3', 'stage 4'),
       col = c('black', 'blue', 'orange'),  lty = c(1, 2, 3), bty = 'n')
```



- If the proportionality assumption holds, then the line should be straight through the origin. Therefore, the assumption is not meet.

12.3

a.

```
### a ###
help(hodg)
data(hodg)

fit.12.3a <- survreg(Surv(time, delta) ~ factor(gtype) + factor(dtype) + factor(gtype)*f
actor(dtype),
                    data = hodg, dist = "weibull")
summary(fit.12.3a)
```

```
##
## Call:
## survreg(formula = Surv(time, delta) ~ factor(gtype) + factor(dtype) +
##     factor(gtype) * factor(dtype), data = hodg, dist = "weibull")
##                                 Value Std. Error     z        p
## (Intercept)                      7.831      0.753 10.40 2.36e-25
## factor(gtype)2                  -2.039      0.930 -2.19 2.83e-02
## factor(dtype)2                  -4.198      1.067 -3.94 8.31e-05
## factor(gtype)2:factor(dtype)2  5.358      1.377  3.89 9.98e-05
## Log(scale)                       0.503      0.167  3.01 2.63e-03
##
## Scale= 1.65
##
## Weibull distribution
## Loglik(model)= -176.5   Loglik(intercept only)= -183.3
##  Chisq= 13.54 on 3 degrees of freedom, p= 0.0036
## Number of Newton-Raphson Iterations: 5
## n= 43
```

```
mu <- fit.12.3a$coefficients[1]
gamma <- fit.12.3a$coefficients[2:4]
sigma <- fit.12.3a$scale
#Parameter Estimates (Weibull Assumption):
lambda <- exp(-mu / sigma) #Type in book (.02 not .002)
alpha <- 1 / sigma
beta.hat <- -gamma / sigma

#rm(fit.12.3a, mu, gamma, sigma, alpha, lambda, beta.hat)
```

```
# s.e. for the Weibull estimates, using multivariate delta method
# Thanks Eric
source('/Users/huiyuhu/Desktop/Study/UCLA_Biostat/M215/getWeibullEstimates.R')
getWeibullEstimates(fit.12.3a)
```

```
## $WeibullModel
##                                Value    se      z     p
## lambda                         0.009 0.007  1.195 0.232
## alpha                          0.605 0.101 -3.908 2.000
## factor(gtype)2                 1.233 0.574  2.148 0.032
## factor(dtype)2                 2.539 0.699  3.634 0.000
## factor(gtype)2:factor(dtype)2 -3.241 0.878 -3.690 2.000
## 
## $estimates
##                                logHR     HR
## factor(gtype)2                 1.233  3.433
## factor(dtype)2                 2.539 12.668
## factor(gtype)2:factor(dtype)2 -3.241  0.039
## 
## $var
##                                     lambda         alpha factor(gtype)2
## lambda                         5.384626e-05 -0.0006271302   -0.002595961
## alpha                         -6.271302e-04  0.0102262194    0.013725172
## factor(gtype)2                -2.595961e-03  0.0137251720    0.329532420
## factor(dtype)2                -3.594903e-03  0.0300143011    0.240283846
## factor(gtype)2:factor(dtype)2  3.879969e-03 -0.0346626938   -0.357633821
##                               factor(dtype)2 factor(gtype)2:factor(dtype)2
## lambda                          -0.003594903                   0.003879969
## alpha                            0.030014301                  -0.034662694
## factor(gtype)2                   0.240283846                  -0.357633821
## factor(dtype)2                   0.488092993                  -0.501736183
## factor(gtype)2:factor(dtype)2   -0.501736183                   0.771460584
## 
## $LogLinearModel
##                                   Value Std. Error         z            p
## (Intercept)                    7.8313074  0.7526664 10.404752 2.358705e-25
## factor(gtype)2                -2.0392680  0.9296009 -2.193703 2.825678e-02
## factor(dtype)2                -4.1982878  1.0668680 -3.935152 8.314410e-05
## factor(gtype)2:factor(dtype)2  5.3583409  1.3770623  3.891139 9.977475e-05
## Log(scale)                     0.5028789  0.1672072  3.007519 2.633899e-03
```

-

```
### b ###
fit.12.3b <- survreg(Surv(time, delta) ~ factor(gtype) + factor(dtype),
                     data = hodg, dist = "weibull")
summary(fit.12.3b)
```

```
##
## Call:
## survreg(formula = Surv(time, delta) ~ factor(gtype) + factor(dtype),
##     data = hodg, dist = "weibull")
##                    Value Std. Error        z        p
## (Intercept)      6.90728      0.651 10.61012 2.67e-26
## factor(gtype)2   0.00197      0.960  0.00205 9.98e-01
## factor(dtype)2  -0.61574      0.934 -0.65893 5.10e-01
## Log(scale)       0.70042      0.166  4.21296 2.52e-05
##
## Scale= 2.01
##
## Weibull distribution
## Loglik(model)= -183   Loglik(intercept only)= -183.3
##  Chisq= 0.59 on 2 degrees of freedom, p= 0.75
## Number of Newton-Raphson Iterations: 4
## n= 43
```

```
1-pchisq(13.54,1)
```

```
## [1] 0.0002335324
```

** The likelihood ratio test statistic is L = 2*(-176.5 - (-183)) = 13.54, degree of freedom is 1, so p-value will be about 0.0002 (<0.05). Therefore, the null hypothesis is rejected at alpha = 0.05. **

c.

- The baseline group is NHL Allogenic patients. Using the estimates and proportional hazards property of the Weibull regression model, the relative risk of death for an **NHL Auto** patient as compared to an **NHL Allo** patient is:

$$RR = exp(\beta 1) = exp(1.233) = 3.34151$$

- 95% C.I. for

$$\beta 1$$

:

$$\beta 1 \pm 1.96 * SE(\beta 1) = 1.233 \pm 1.96 * 0.574 = (0.108, 2.358)$$

- Then 95% C.I. for RR is:

$$(exp(0.108), exp(2.358)) = (1.114, 10.570)$$

d.

- H0: The death rates are same for HOD Allo and NHL Allo ($\beta 2 = 0$)
- According to result of (a), the p-value for $\beta 2$ is < 0.001, which means we have enough evidence to reject H0 at alpha = 0.05. Therefore, there is statistical significant difference in death rates for HOD Allo and NHL Allo patients.
- H0: The death rates are same for HOD Auto and NHL Auto ($\beta 2 + \beta 3 = 0$, where $C = [0, 0, 0, 1, 1]$)

```
V <- getWeibullEstimates(fit.12.3a)$var
beta <- getWeibullEstimates(fit.12.3a)$WeibullModel[,1]
C <- c(0,0,0,1,1)
C <- t(as.vector(C)) #transpose
chi_sq <- t(C %*% beta) %*% solve(C %*% V %*% t(C)) %*% (C %*% beta)
1-pchisq(chi_sq,1)
```

```
##           [,1]
## [1,] 0.1653719
```

$$chi - sq = [C\beta]^t[CVC^t]^{-1}[C\beta] = 0.17$$

- The p-value = 0.17 is larger than 0.05, so we do not have enough evidence to reject H0 at alpha = 0.05. Therefore, the death rates are NOT same for HOD Auto and NHL Auto.

e.

- H0 : h(t | NHL Allo) = h(t | NHL Auto) and h(t | HOD Allo) = h(t |HOD Auto)
- Using contrast to do chi-square test.

$$\mathbf{C2} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

```
V <- getWeibullEstimates(fit.12.3a)$var
beta <- getWeibullEstimates(fit.12.3a)$WeibullModel[,1]
C1 <- c(0,0,1,0,0)
C2 <- c(0,0,1,0,1)
C <- rbind(C1, C2)
chi_sq <- t(C %*% beta)%*%solve(C %*% V %*% t(C))%*%(C%*%beta)
1-pchisq(chi_sq,2)
```

```
##              [,1]
## [1,] 0.0008852483
```

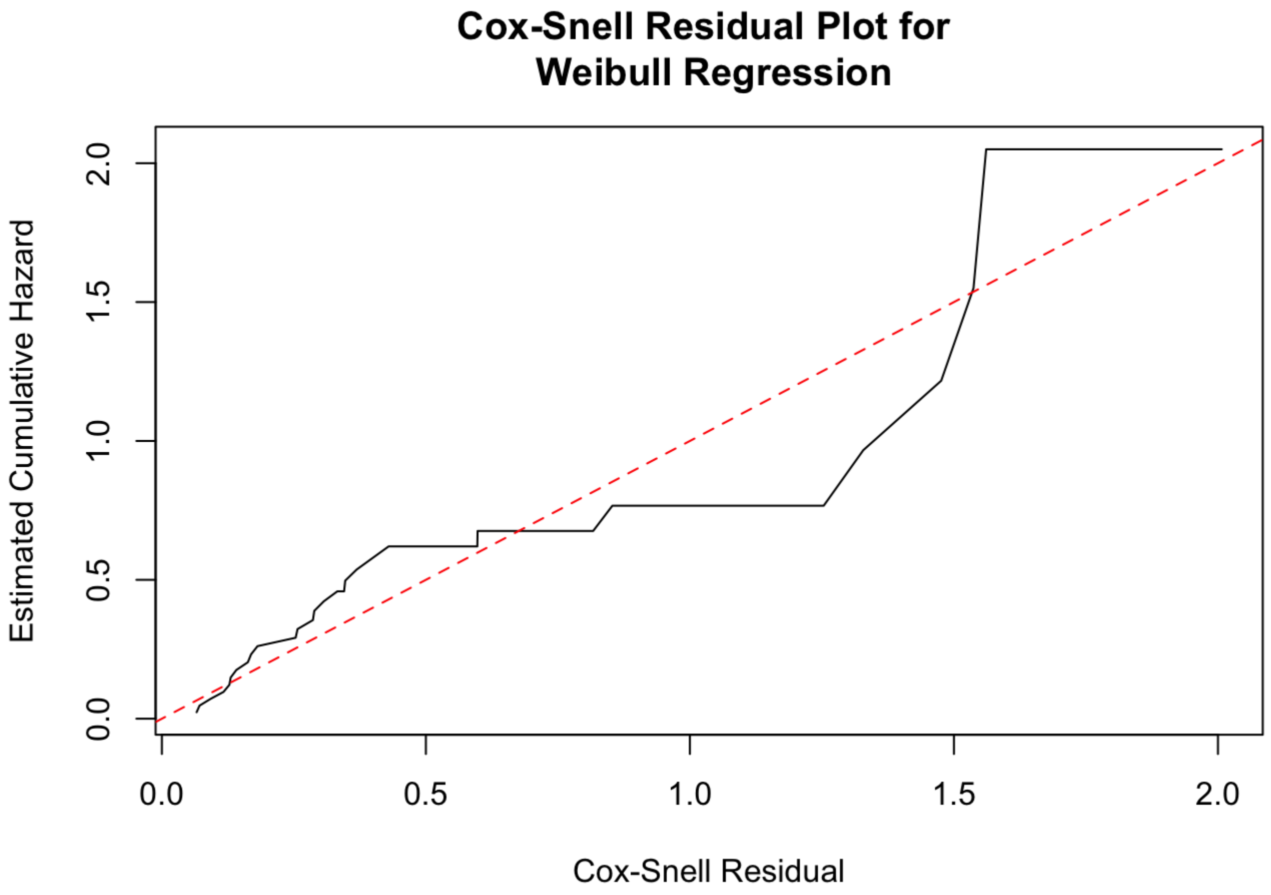
- The p-value = 0.0009 is smaller than 0.05, so we have enough evidence to reject H0 at alpha = 0.05. Therefore, death rates for Auto transplant and Allo transplant patients are different against the alternative they are different for at least one disease group.

f.

- The semiparametric proportional hazards model (weibull): The RR for NHL auto to NHL allo is larger. Both methods suggest that there is a significant difference between the death rates between the two types in allo groups and no significant difference in the auto group.

12.14

```
### a ###

# weibull
sigma <- fit.12.3a$scale
alpha <- 1 / sigma
eta <- -fit.12.3a$linear.predictors / sigma
r.wb <- hodg$time^alpha * exp(eta)
fit <- survfit(Surv(r.wb, hodg$delta) ~ 1)
H.wb <- cumsum(fit$n.event/fit$n.risk)

plot(H.wb ~ fit$time, type = 'l', main = 'Cox-Snell Residual Plot for \n Weibull Regress
ion',
ylab = 'Estimated Cumulative Hazard', xlab = 'Cox-Snell Residual')
abline(0, 1, col='red', lty=2)
```



- The curve is not perfectly linear, especially after 1.0. Therefore,the model under Weibull Regression is lack of fit.