A2 Project 2

Johnsen & Johnsen

2022-10-26

Projekt 2: Survival Data

Analysis of the Binary Data

Read the data Logistic.txt into R.

```
# if (Sys.getenv("LOGNAME") == "mortenjohnsen"){
#   setwd("/Users/mortenjohnsen/OneDrive - Danmarks Tekniske Universitet/DTU/9. Semester/02418 - Statis
# } else {
#   setwd("~/Documents/02418 Statistical Modelling/Assignments/Assignment 1/Project-1")
# }

log.data <- read.table("Logistic.txt", header=TRUE, sep="",
                       as.is=TRUE)

source("testDistribution.R")
```

Fit a logistic regression model for the binary outcome AIDS=”yes” versus AIDS=”no” with the explanatory variable treatment with AZT (Yes, NO). Present the odds ratio for the effect of AZT on AIDS with 95% confidence interval and interpret the result in words

The logistic regression model is given by the likelihood function:

$$L(\theta) = \prod_i \left(\frac{\theta_i}{1-\theta_i} \right)^{y_i} (1-\theta_i)$$

Here θ is calculated as:

$$\theta_i = \frac{e^{\beta_0 + \beta_1 t_{AZT}}}{1 + e^{\beta_0 + \beta_1 t_{AZT}}}$$

```
nll.log <- function(theta, event, n, treatment){

  beta0 <- theta[1]
  beta1 <- theta[2]

  theta <- exp(beta0 + beta1 * treatment)/(1 + exp(beta0 + beta1 * treatment))

  nll <- -sum(log(theta)*event  + log(1-theta)*(n - event))
  return(nll)
```

```
}

# For AZT treatment:
log.reg <- nlminb(start = c(1,1), objective = nll.log, event = log.data$AIDS_yes, n = log.data$n, treatr

beta0 <- log.reg$par[1]
beta1 <- log.reg$par[2]
```

```
logistic <- data.frame("AZT" = c(rep(1,170), rep(0,168))
                       ,"AIDS_yes" = c(rep(c(1,0),c(25,170-25)), rep(c(1,0), c(44, 168-44))))

fit.glm <- glm(AIDS_yes ~ AZT, data = logistic, family = binomial)
print(cat(paste0("with glm model: ", coef(fit.glm)
  ,"\nBy hand (according to slide 19 lect 4): "
  ,"\nbeta_0 = ", log.reg$par[1], ", beta_1 = ", log.reg$par[2])))
```

```
## with glm model: -1.03609193168383
## By hand (according to slide 19 lect 4):
## beta_0 = -1.03609192753506, beta_1 = -0.721765976832429 with glm model: -0.721765985868547
## By hand (according to slide 19 lect 4):
## beta_0 = -1.03609192753506, beta_1 = -0.721765976832429NULL
```

Calculating 95% wald confidence interval for β_1 and subsequently for the odds ratio.

```
sd <- sqrt(diag(solve(hessian(func = nll.log, x = c(beta0, beta1), event = log.data$AIDS_yes, n = log.da
sd.beta0 <- sd[1]
sd.beta1 <- sd[2]

beta1.wald.CI <- beta1 + c(-1,1)*dnorm(0.975)*sd.beta1
```

Odds ratio = $exp(\beta_1) = 0.4858934$, 95% CI [0.4534386, 0.5206711]. Thus for the individuals receiving the AZT treatment, the odds of developing AIDS or dying is reduced by a factor 0.486 95% CI [0.45, 0.52].

Test the hypothesis of no effect of AZT on AIDS using:

- The likelihood ratio test
- The Wald test
- The score test

Likelihood ratio test Calculate the objective value based on no treatment effect:

```
nll.log.no.effect <- function(theta, event, n, treatment){

  beta0 <- theta[1]

  theta <- exp(beta0)/(1 + exp(beta0))

  nll <- -sum(log(theta)*event  + log(1-theta)*(n - event))
  return(nll)
}
```

```
log.reg.no.effect <- nlminb(start = 1, objective = nll.log.no.effect
                          , event = sum(log.data$AIDS_yes)
                          , n = sum(log.data$n), treatment = 0)
```

Run the LRT (assuming regularity of the likelihoods (*page 36*)):

```
(chi.squared <- - 2 * (log.reg$objective - log.reg.no.effect$objective))
```

```
## [1] 6.929332
```

```
(p.value <- 1 - pchisq(chi.squared, df = 1))
```

```
## [1] 0.008479333
```

Wald test statistic Calculating the wald test statistic (*p. 156*, or *p. 42 with* $\theta_0 = 0$).

```
(waldTestStatistic <- beta1/sd.beta1)
```

```
## [1] -2.589515
```

```
(wald.p.value <- 2*pnorm(waldTestStatistic))
```

```
## [1] 0.009611115
```

The score test Calculating the score function for the logistic regression model.

$$S(\theta) = \frac{1}{\theta}\sum_i y_i + \frac{1}{1+\theta}\sum_i 1 - y_i$$

```
#ikke færdig
scoreFunction <- function(theta){
  beta0 <- theta[1]
  beta1 <- theta[2]
  event <- log.data$AIDS_yes
  n <- log.data$n
  treatment <- c(1,0)

  theta <- exp(beta0 + beta1 * treatment)/(1 + exp(beta0 + beta1 * treatment))

  score <- -sum(1/theta * event + 1/(1 + theta) * (n - event))
  return(score)
}

scoreFunction(log.reg$par) * hessian(func = scoreFunction, x = log.reg$par)
```

```
##           [,1]     [,2]
## [1,] 146131.87 77798.54
## [2,]  77798.54 77798.54
```

```
scoreFunction(log.reg$par)
```

```
## [1] -562.6744
```

Analysis of the survival time data

Descriptive statistics

Read the data actg320.txt into R Read in the data:

```
#tx: Treatment indicator. 1 = New treatment, 0 = Control treatment
#event: Indicator for AIDS or death. 1 = AIDS diagnosis or death, 0 = Otherwise
#time: Time to AIDS diagnosis or death. Days
#så tiden for event = 0 må angive at personen har været med i studiet time[X] dage uden at være enten d
actg320 <- read.table("actg320.txt", header=TRUE, sep="",
                      as.is=TRUE)

#select time, event and tx as they are the only relevant variables in this project
actg <- actg320 %>%
  dplyr::select(time, event, tx, cd4)
```

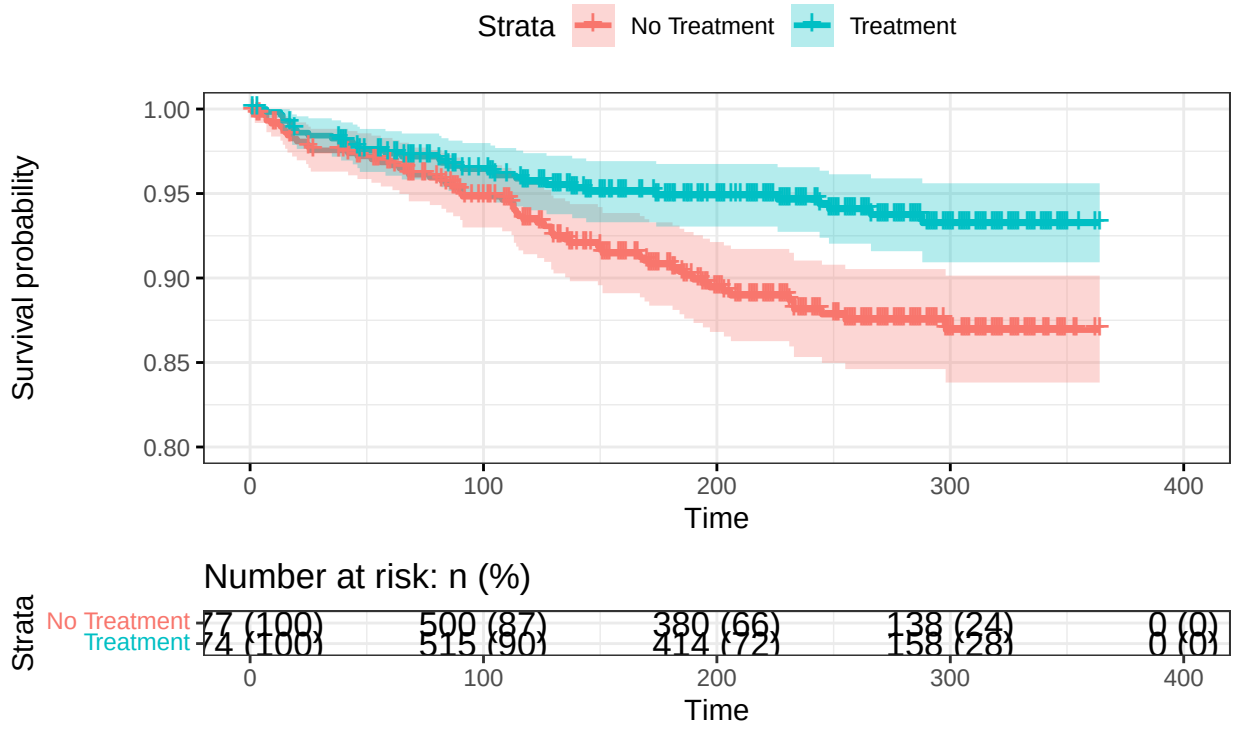
```
actg %>%
  group_by(tx) %>%
  summarise("Got AIDS or DIED" = sum(event),
            "Proportion" = sum(event)/n(),
            "Participants Given the Treatment" = n(),
            "Total follow up time" = sum(time))
```

How many patients got AIDS or died in the two treatment groups? And how long was the total follow-up time in the two groups?

```
## # A tibble: 2 x 5
##      tx `Got AIDS or DIED` Proportion `Participants Given the Treatment` Total~1
##   <int>              <int>      <dbl>                                <int>   <int>
## 1     0                 63     0.109                                  577  128991
## 2     1                 33     0.0575                                 574  135950
## # ... with abbreviated variable name 1: `Total follow up time`
```

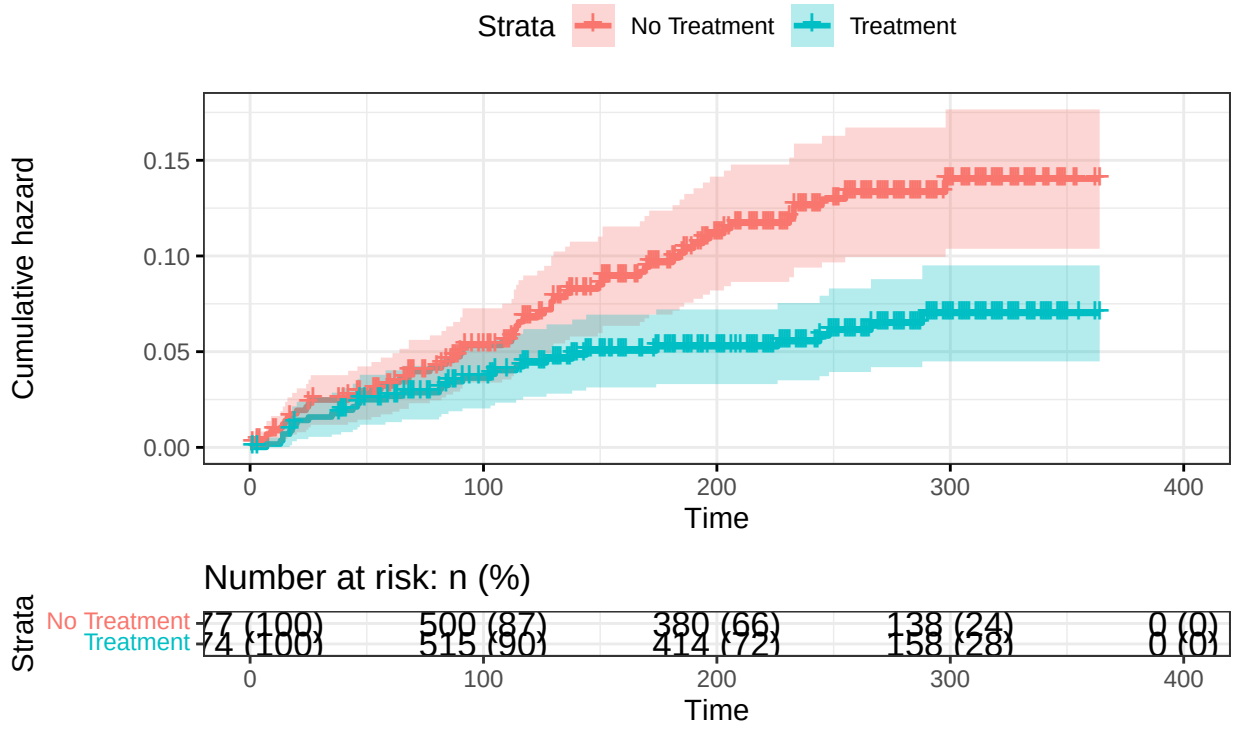
Plot the survival functions in the two treatment groups, which group seems to be doing best?
The treatment group seems to be doing the best (consider making this plot by hand instead of using survfit).

```
kaplan.meier <- survfit(Surv(time, event) ~ tx, data = actg)
ggsurvplot(kaplan.meier
                   , data = actg
                   , conf.int = T
                   , risk.table = "abs_pct"
                   , ylim = c(0.8,1)
                   ,ggtheme = theme_bw()
                   ,legend.labs = c("No Treatment", "Treatment"))
```




```
ggsurvplot(kaplan.meier
                    , data = actg
                    , conf.int = T
                    , risk.table = "abs_pct"
                    ,ggtheme = theme_bw()
                    ,fun = "cumhaz"
                    ,legend.labs = c("No Treatment", "Treatment"))
```

Plot the cumulative incidence functions for the two groups, which plot would you prefer?



Compare the survival in the two treatment groups using a log-rank test. See slides 31-33 from week 6 for calculations by hand.

```
survdiff(Surv(time, event) ~ tx, data = actg)
```

```
## Call:
## survdiff(formula = Surv(time, event) ~ tx, data = actg)
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## tx=0 577       63     47.1      5.37      10.5
## tx=1 574       33     48.9      5.17      10.5
##
##  Chisq= 10.5  on 1 degrees of freedom, p= 0.001
```

p-value of 0.001, thus there is a significant difference between the two treatment groups based on a significance level of 0.05.

Parametric survival models

Fit parametric survival models containing treatment (tx) and CD4 count (cd4) as explanatory variables

- Try using the exponential, Weibull and log-logistic models, which one gave the best fit (and why)?

```r
exp.model <- survreg(Surv(time, event) ~ tx + cd4, data = actg, dist = "exponential")
weibull.model <- survreg(Surv(time, event) ~ tx + cd4, data = actg, dist = "weibull")
loglogistic.model <- survreg(Surv(time, event) ~ tx + cd4, data = actg, dist = "loglogistic")
```

```r
summary(exp.model)
```

```
##
## Call:
## survreg(formula = Surv(time, event) ~ tx + cd4, data = actg,
##     dist = "exponential")
##               Value Std. Error    z       p
## (Intercept) 6.71473    0.15647 42.9 < 2e-16
## tx          0.66680    0.21489  3.1  0.0019
## cd4         0.01609    0.00251  6.4 1.5e-10
##
## Scale fixed at 1
##
## Exponential distribution
## Loglik(model)= -819.9   Loglik(intercept only)= -856.6
##  Chisq= 73.36 on 2 degrees of freedom, p= 1.2e-16
## Number of Newton-Raphson Iterations: 7
## n= 1151
```

```r
nll.exp <- function(theta
                    , time = actg$time
                    , event = actg$event
                    , treatment = actg$tx
                    , cd4 = actg$cd4){
  beta0 <- theta[1]
  beta1 <- theta[2]
  beta2 <- theta[3]

  h <- exp(- beta0 - beta1 * treatment - beta2 * cd4)
  H <- time/exp(beta0 + beta1 * treatment + beta2 * cd4)
  nll <- -sum(event*log(h) - H)
  return(nll)
}

exp.model.manual <- nlminb(start = c(1,1,1), objective = nll.exp)
sd.exp.manual <- sqrt(diag(solve(hessian(func = nll.exp, x = exp.model.manual$par))))
exp.model.manual.AIC <- - 2 * exp.model.manual$objective - 2 * 3
```

Weibull model Use: slide 10 for f(t), S(t), h(t) and H(t) relationships, slide 19 for general log-likelihood expression for censored data, slide 55 for S(t) and z, slide 60 for h(t).

```r
summary(weibull.model)
```

```
##
## Call:
## survreg(formula = Surv(time, event) ~ tx + cd4, data = actg,
##     dist = "weibull")
```

```
##               Value Std. Error     z       p
## (Intercept) 7.05743    0.25235 27.97 < 2e-16
## tx          0.84013    0.28582  2.94  0.0033
## cd4         0.02063    0.00379  5.44 5.3e-08
## Log(scale)  0.24834    0.09715  2.56  0.0106
##
## Scale= 1.28
##
## Weibull distribution
## Loglik(model)= -816.3   Loglik(intercept only)= -852.9
##  Chisq= 73.2 on 2 degrees of freedom, p= 1.3e-16
## Number of Newton-Raphson Iterations: 9
## n= 1151
```

```
nll.wei <- function(theta
                    , time = actg$time
                    , event = actg$event
                    , treatment = actg$tx
                    , cd4 = actg$cd4){
  sigma <- exp(theta[1])  #estimate scale parameter on log-scale
  beta0 <- theta[2]
  beta1 <- theta[3]
  beta2 <- theta[4]

  h <-1/sigma * time^(1/sigma - 1) * exp(-1/sigma * (beta0 + beta1 * treatment + beta2 * cd4))
  H <- time^(1/sigma) * exp(-1/sigma * (beta0 + beta1 * treatment + beta2 * cd4))
  nll <- -sum(event*log(h) - H)
  return(nll)
}

weibull.model.manual <- nlminb(start = c(1,1,1,1), objective = nll.wei)
sd.weibull.manual <- sqrt(diag(solve(hessian(func = nll.wei, x = weibull.model.manual$par))))
weibull.model.manual.AIC <- - 2 * weibull.model.manual$objective + 2 * length(weibull.model.manual$par)
```

```
summary(loglogistic.model)
```

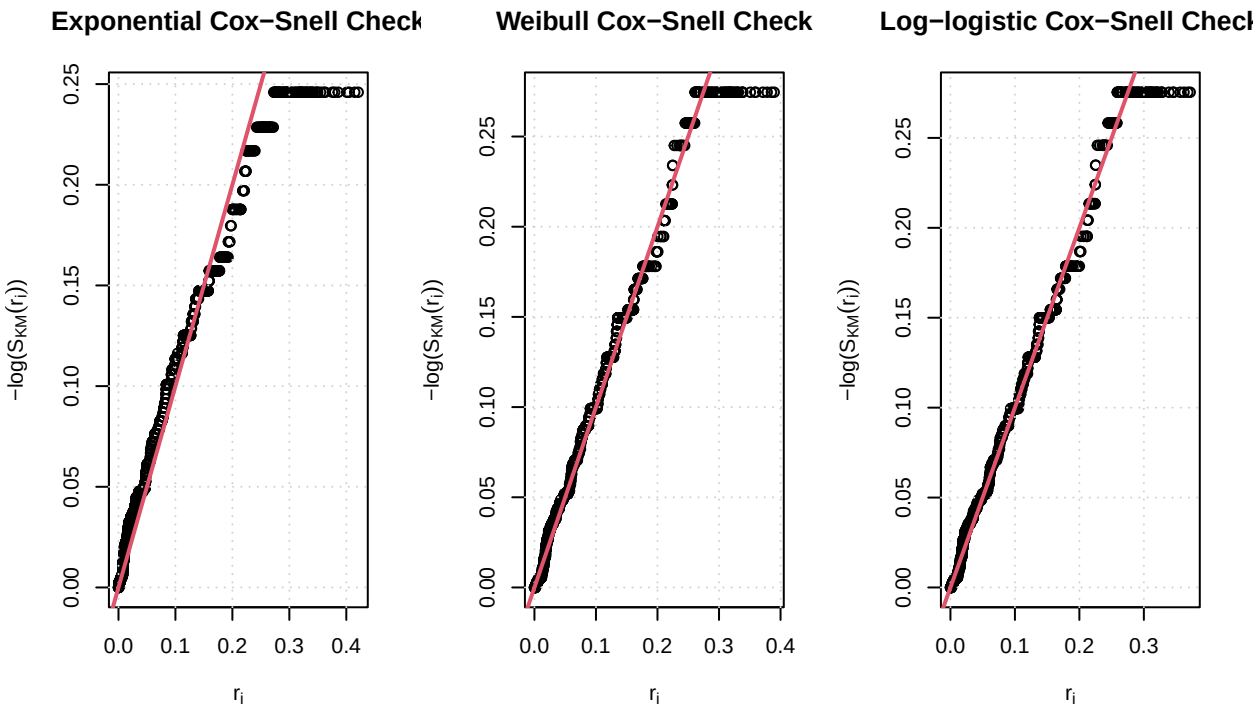
```
##
## Call:
## survreg(formula = Surv(time, event) ~ tx + cd4, data = actg,
##     dist = "loglogistic")
##               Value Std. Error     z       p
## (Intercept) 6.82584    0.25453 26.82 < 2e-16
## tx          0.84295    0.28980  2.91  0.0036
## cd4         0.02080    0.00375  5.55 2.9e-08
## Log(scale)  0.20259    0.09558  2.12  0.0340
##
## Scale= 1.22
##
## Log logistic distribution
## Loglik(model)= -815.8   Loglik(intercept only)= -852.7
##  Chisq= 73.73 on 2 degrees of freedom, p= 9.8e-17
## Number of Newton-Raphson Iterations: 6
## n= 1151
```

```r
nll.loglog <- function(theta
                       , time = actg$time
                       , event = actg$event
                       , treatment = actg$tx
                       , cd4 = actg$cd4){
  sigma <- theta[1]
  beta0 <- theta[2]
  beta1 <- theta[3]
  beta2 <- theta[4]

  z <- (log(time) - (beta0 + beta1 * treatment + beta2 * cd4))/sigma

  h <- exp(z)/(1 + exp(z)) * 1/sigma * 1/time
  S <- 1 / (1 + exp(z))
  H <- -log(S)
  #H <- time^(1/sigma) * exp(-1/sigma * (beta0 + beta1 * treatment + beta2 * cd4))
  nll <- -sum(event*log(h) - H)
  return(nll)
}

loglogistic.model.manual <- nlminb(start = c(1,1,1,1), objective = nll.loglog)
sd.loglogistic.manual <- sqrt(diag(solve(hessian(func = nll.loglog, x = loglogistic.model.manual$par))))
loglogistic.model.manual.AIC <- - 2 * loglogistic.model.manual$objective + 2 * length(loglogistic.model
```

Examining the fits of the models through the cox-snell residuals

```r
par(mfrow = c(1,3))
actg$cox.snell.exp <- actg$time * exp(- exp.model$linear.predictors)
fit.exp <- survfit(Surv(cox.snell.exp, event == 1) ~ 1, data = actg)
plot(fit.exp$time, -log(fit.exp$surv), main = "Exponential Cox-Snell Check"
     ,xlab = TeX("$r_i$")
     ,ylab = TeX("-log($S_{KM}(r_i)$)"))
grid()
abline(a=0, b=1, col = 2, lwd = 2)

#slide 51 week 7 for weibull cox-snell
beta <- t(t(weibull.model.manual$par[2:4]))
x <- as.matrix(cbind(1, actg$tx, actg$cd4))
y.pred <- x %*% beta
y <- log(actg$time)

sigma <- exp(weibull.model.manual$par[1])

actg$r <- exp((y - y.pred)/sigma)
#slide 35 week 7 for the plot. [r_i, log(S_KM(r_i))], should yield a linear relationship with slope = 1
fit.wei <- survfit(Surv(r, event == 1) ~ 1, data = actg)
#the output from the survfit function: time = r_i (as we input r_i as time).
#the 'surv' output is the kaplan meier survival function of r_i:
plot(fit.wei$time, -log(fit.wei$surv), main = "Weibull Cox-Snell Check"
     ,xlab = TeX("$r_i$")
     ,ylab = TeX("-log($S_{KM}(r_i)$)"))
grid()
abline(a=0, b=1, col = 2, lwd = 2)
```

```
#slide 55 week 7 + slide 33 week 7 + slide 65 week 7
actg$cox.snell.loglog <- log( 1 + exp((log(actg$time) - loglogistic.model$linear.predictors) / loglogis
fit.loglog <- survfit(Surv(cox.snell.loglog, event == 1) ~ 1, data = actg)
plot(fit.loglog$time, -log(fit.loglog$surv), main = "Log-logistic Cox-Snell Check"
     ,xlab = TeX("$r_i$")
     ,ylab = TeX("-log($S_{KM}(r_i)$)"))
grid()
abline(a=0, b=1, col = 2, lwd = 2)
```



Weibull and log-logistic seem quite similar. For both we still see a heavy tail, which is not accounted for in the model.

Use AIC to compare the models (exp excluded due to poor cox-snell fit):

```
cat("Weibull Regression Model AIC: ", weibull.model.manual.AIC,
    "\nLog-logistic Regression Model AIC: ", loglogistic.model.manual.AIC)
```

```
## Weibull Regression Model AIC:  -1624.671
## Log-logistic Regression Model AIC:  -1623.655
```

```
weibull.upper <- as.numeric(weibull.model.manual$par + qnorm(0.975)*sd.weibull.manual)
weibull.lower <- as.numeric(weibull.model.manual$par - qnorm(0.975)*sd.weibull.manual)

cat("beta_0 = ", weibull.model.manual$par[2], "95% CI [",weibull.lower[2],", ",weibull.upper[2],"]"
    ,"\nbeta_1 = ", weibull.model.manual$par[3], "95% CI [",weibull.lower[3],", ",weibull.upper[3],"]"
    ,"\nbeta_2 = ", weibull.model.manual$par[4], "95% CI [",weibull.lower[4],", ",weibull.upper[4],"]"
    ,"\nscale = ", exp(weibull.model.manual$par[1]), "95% CI [",exp(weibull.lower[1]),", ",exp(weibull.
```

Using the survival model you chose, make a table of estimates and their 95% confidence intervals

```
## beta_0 =  7.05743 95% CI [ 6.562829 ,  7.55203 ]
## beta_1 =  0.8401271 95% CI [ 0.2799285 ,  1.400326 ]
## beta_2 =  0.02063062 95% CI [ 0.01319748 ,  0.02806376 ]
## scale =  1.281897 95% CI [ 1.059638 ,  1.550775 ]
```

Table format:

```
tibble("parameter" = c("beta_0", "beta_1", "beta_2", "sigma")
       , "theta hat" = c(weibull.model.manual$par[2:4], exp(weibull.model.manual$par[1]))
       , "lower CI (2.5%)" = c(weibull.lower[2:4], exp(weibull.lower[1]))
       , "upper CI (97.5%)" = c(weibull.upper[2:4], exp(weibull.upper[1])))
```

```
## # A tibble: 4 x 4
##   parameter `theta hat` `lower CI (2.5%)` `upper CI (97.5%)`
##   <chr>           <dbl>             <dbl>              <dbl>
## 1 beta_0         7.06              6.56               7.55
## 2 beta_1         0.840             0.280              1.40
## 3 beta_2         0.0206            0.0132             0.0281
## 4 sigma          1.28              1.06               1.55
```

θ does not span 1, and thus the estimated weibull distribution is significantly different from simply using an exponential distribution.

Using your model compute the time ratio for the treatment effect. Similarly, compute the time ratio for the effect of increasing the CD4 count with 50. In both cases uncertainty evaluation (e.g. confidence intervals) should be included. Interpret the results in words For the treatment effect:

```
beta1 <- weibull.model.manual$par[3]

cat("Time Ratio for the treatment effect: TR(tx = 1, tx = 0) = ", exp(beta1), " 95% CI [",exp(c(weibull
```

```
## Time Ratio for the treatment effect: TR(tx = 1, tx = 0) =  2.316661  95% CI [ 1.323035 , 4.056521 ]
```

Based on this calculation of the time ratio of the treatment effect, it is evident that the treatment increase the median survival time by a factor 2.32 95% CI [1.32; 4.06].

For increasing the cd4 count by 50:

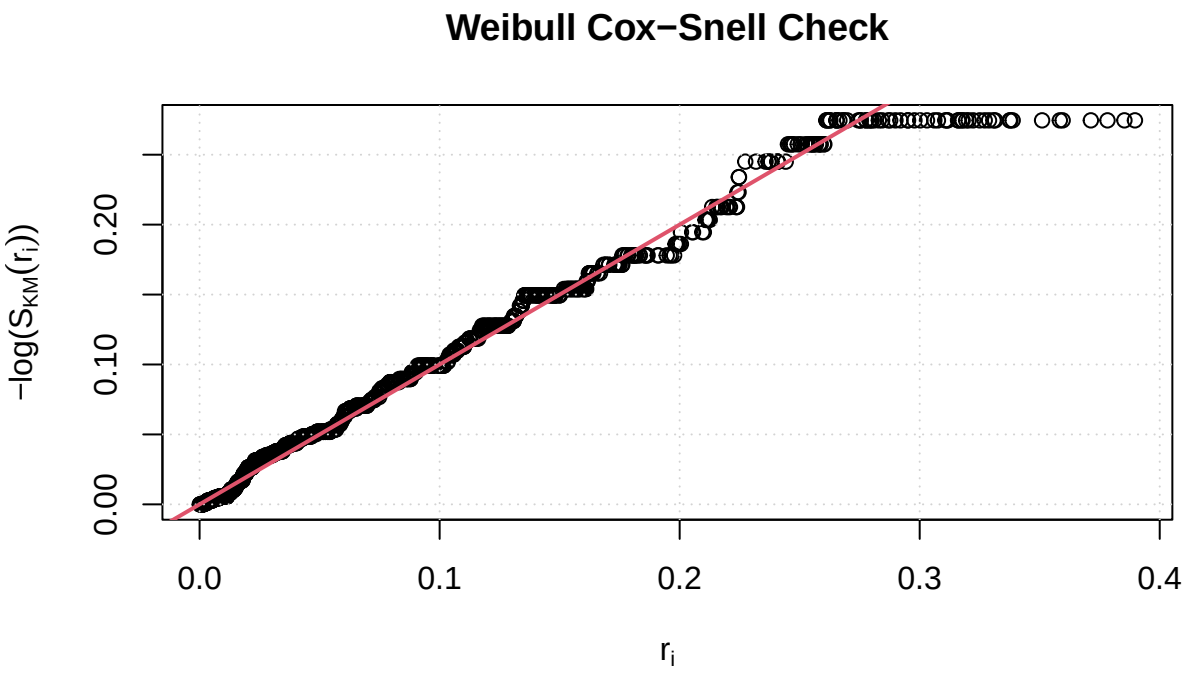
```
beta2 <- weibull.model.manual$par[4]

cat("Time Ratio for the treatment effect: TR(tx = 1, tx = 0) = ", exp(beta2*50), " 95% CI [",exp(c(50*w
```

```
## Time Ratio for the treatment effect: TR(tx = 1, tx = 0) =  2.805358  95% CI [ 1.934549 , 4.068148 ]
```

Here we see that by increasing the cd4 count by 50, the median survival time is increased by a factor 2.8 95% CI [1.93,4.07].

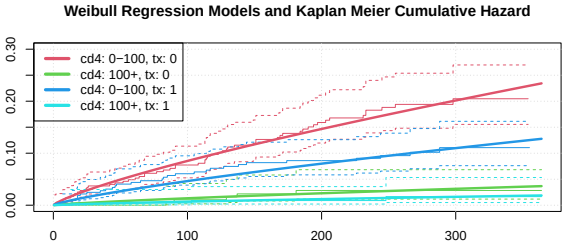
Assess the goodness of fit of this model using a plot based on the Cox Snell residuals This plot was already made earlier, but here it is again.

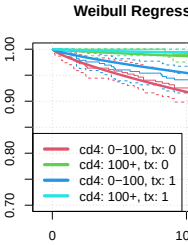
```
par(mfrow = c(1,1))

plot(fit.wei$time, -log(fit.wei$surv), main = "Weibull Cox-Snell Check"
     ,xlab = TeX("$r_i$")
     ,ylab = TeX("-log($S_{KM}(r_i)$)"))
grid()
abline(a=0, b=1, col = 2, lwd = 2)
```




```
actg$cd4.cats <- cut(actg$cd4, breaks = c(0,100,max(actg$cd4)))
survival.functions <- survfit(Surv(time, event) ~ tx + cd4.cats, data = actg)

actg$cd4.category.num <- 0
actg$cd4.category.num[actg$cd4 < 100] <- 1
actg$cd4.category.num[actg$cd4 >= 100] <- 2

wei.survival <- function(theta, time, tx, cd4){
  beta0 <- theta[1]
  beta1 <- theta[2]
  beta2 <- theta[3]
  scale <- theta[4]

  out <- exp(-(time/exp(beta0 + beta1*tx + beta2*cd4))^(1/scale))
  return(out)
```

```
}

#for fixed values
step.size <- 0.5
theta <- c(weibull.model$coefficients,weibull.model$scale)
time.steps <- seq(0.5,max(actg$time), step.size) #lader den lige starte i 0.5 for at undgå bøvl med h.c
first.sim <- wei.survival(theta, time.steps, 0, median(actg$cd4[actg$cd4 < 100 & actg$tx == 0]))
second.sim <- wei.survival(theta, time.steps, 0, median(actg$cd4[actg$cd4 > 100 & actg$tx == 0]))
third.sim <- wei.survival(theta, time.steps, 1, median(actg$cd4[actg$cd4 < 100 & actg$tx == 1]))
fourth.sim <- wei.survival(theta, time.steps, 1, median(actg$cd4[actg$cd4 > 100 & actg$tx == 1]))

#wei.survival(c(weibull.model$coefficients,weibull.model$scale), actg$time, actg$tx, actg$cd4)

par(mfrow = c(1,2))
plot(survival.functions, cumhaz = T, conf.int = T, col = 2:5, ylim = c(0,0.3), lwd = 1, main = "Weibull
grid()
f.weibull.t <- function(t, theta, x, scale){
  shape <- as.vector(theta %*% x)
  out <- shape/scale * (t/scale)^(shape-1) * exp(-(t/scale)^shape)
  return(out)
}

h.weibull.t <- function(t, theta, treatment, cd4, step.size){
  sigma <- exp(theta[1])
  beta0 <- theta[2]
  beta1 <- theta[3]
  beta2 <- theta[4]

  out <- cumsum(1/sigma * t^(1/sigma - 1) * exp(-1/sigma * (beta0 + beta1 * treatment + beta2 * cd4))*s
  return(out)
}

lines(time.steps, h.weibull.t(time.steps, weibull.model.manual$par, 0, median(actg$cd4[actg$cd4 < 100 &
lines(time.steps, h.weibull.t(time.steps, weibull.model.manual$par, 0, median(actg$cd4[actg$cd4 > 100 &
lines(time.steps, h.weibull.t(time.steps, weibull.model.manual$par, 1, median(actg$cd4[actg$cd4 < 100 &
lines(time.steps, h.weibull.t(time.steps, weibull.model.manual$par, 1, median(actg$cd4[actg$cd4 > 100 &
legend(legend = c("cd4: 0-100, tx: 0", "cd4: 100+, tx: 0", "cd4: 0-100, tx: 1", "cd4: 100+, tx: 1"), lty

plot(survival.functions, conf.int = T, col = 2:5, ylim = c(1,0.7), lwd = 1, main = "Weibull Regression
grid()
lines(time.steps, first.sim, col = 2 , lwd = 3)
lines(time.steps, second.sim, col = 3, lwd = 3)
lines(time.steps, third.sim, col = 4 , lwd = 3)
lines(time.steps, fourth.sim, col = 5, lwd = 3)
legend(legend = c("cd4: 0-100, tx: 0", "cd4: 100+, tx: 0", "cd4: 0-100, tx: 1", "cd4: 100+, tx: 1"), lty
```





Give a graphical presentation of your model