

A1 Project 2

Johnsen & Johnsen

2022-10-06

Projekt 2: Survival Data

Analysis of the Binary Data

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

str(log.data)
```

Read the data Logistic.txt into R.

```
## 'data.frame':  2 obs. of  3 variables:
## $ AZT      : chr  "Yes" "No"
## $ AIDS_yes: int   25  44
## $ n        : int  170 168
```

```
#all data from one population:
bin.par <- nlminb(start = 0.1, objective = testDistribution
                 , x = c(sum(log.data$AIDS_yes), sum(log.data$n))
                 , distribution = "binomial")
```

Fit the Binomial distribution to the data (i.e. consider all data as coming from the same population)

```
#separately for the groups
x.AZT <- log.data %>%
  filter(AZT == "Yes") %>%
  dplyr::select(AIDS_yes, n) %>%
  as.numeric()

AZT.par <- nlminb(start = 0.1, objective = testDistribution
                 , x = c(x.AZT[1], x.AZT[2])
                 , distribution = "binomial")
```

```
x.no.AZT <- log.data %>%
  filter(AZT == "No") %>%
  dplyr::select(AIDS_yes, n) %>%
  as.numeric()

no.AZT.par <- nlminb(start = 0.1, objective = testDistribution
  , x = c(x.no.AZT[1], x.no.AZT[2])
  , distribution = "binomial")
```

Fit the Binomial separately to the two distributions and test if there is a difference between the groups Testing if there's a difference between the two groups:

```
p.hat <- sum(log.data$AIDS_yes)/sum(log.data$n)#bin.par$par

#Calculate expected values for this group based on each group size:
e.A.AZT <- log.data$n[log.data$AZT == "Yes"]*p.hat
e.A.no_AZT <- log.data$n[log.data$AZT == "No"]*p.hat

e.nA.AZT <- log.data$n[log.data$AZT == "Yes"]*(1-p.hat)
e.nA.no_AZT <- log.data$n[log.data$AZT == "No"]*(1-p.hat)

e <- c(e.A.AZT, e.A.no_AZT, e.nA.AZT, e.nA.no_AZT)

#### Without Continuity Correction: ####
#chi_squared <- sum((c(log.data$AIDS_yes,log.data$n-log.data$AIDS_yes)-e)^2/e)
#(chi_squared)
#probability of observing this chi-squared test statistic given that the null-hypothesis is true
#rows <- dim(log.data)[1]
#columns <- dim(log.data)[2]-1 #-1 because of the AZT column
#pchisq(chi_squared,df=(rows-1)*(columns-1),lower.tail=FALSE)

#### WITH CONTINUITY CORRECTION (correct): ####
#https://en.wikipedia.org/wiki/Yates%27s_correction_for_continuity
chi_squared_yates <- sum((abs(c(log.data$AIDS_yes,log.data$n-log.data$AIDS_yes)-0.5)^2/e)
#probability of observing this chi-squared test statistic given that the null-hypothesis is true
rows <- dim(log.data)[1]
columns <- dim(log.data)[2]-1 #-1 because of the AZT column
cat(paste("chi_squared test statistic: ", chi_squared_yates, "\nP-value: ", pchisq(chi_squared_yates,df=
  "\nThus there is a significant difference between the two groups. "))

## chi_squared test statistic: 6.17102276663049
## P-value: 0.0129859512880794
## Thus there is a significant difference between the two groups.
```

```
### Result: There's a difference between the two groups.
```

```
#### direct test of proportions using R: ####
#log.data.for.chi <- log.data; log.data.for.chi$f <- log.data.for.chi$n - log.data.for.chi$AIDS_yes
#prop.test(log.data$AIDS_yes, log.data$n)
#or
#chisq.test(as.matrix(log.data.for.chi[,c(2,4)]))
```

Result: There is a statistically significant effect of the treatment.

Calculating MLE and uncertainty for p_0 and p_1 :

```
#Using the Wald statistic:
pf.p <- function(p, y){
  nll <- testDistribution(p, y, distribution = "binomial")
  return(nll)
}
#For p0
hessian.p0 <- hessian(func = pf.p, x = no.AZT.par$par, y = c(x.no.AZT[1], x.no.AZT[2]))
sd.p0 <- sqrt(diag(solve(hessian.p0)))
CI.p0 <- no.AZT.par$par + c(-1,1)*qnorm(0.975)*sd.p0

#For p1
hessian.p1 <- hessian(func = pf.p, x = AZT.par$par, y = c(x.AZT[1], x.AZT[2]))
sd.p1 <- sqrt(diag(solve(hessian.p1)))
CI.p1 <- AZT.par$par + c(-1,1)*qnorm(0.975)*sd.p1

print(cat("MLE of p and 95% CI for group with AZT treatment: ", round(AZT.par$par,3), " [",round(CI.p1,3),
"\nMLE of p and 95% CI for group with no AZT treatment: ", round(no.AZT.par$par,3), " [",round(CI.p0,3),
"\n"))

## MLE of p and 95% CI for group with AZT treatment:  0.147  [ 0.094 ;  0.2 ]
## MLE of p and 95% CI for group with no AZT treatment:  0.262  [ 0.195 ;  0.328 ]NULL
```

#Profile likelihoods to ensure that the Wald CI is an appropriate approximation.

Estimate parameters in the model (p_0 probability of AIDS in control group, p_1 probability of AIDS in treatment group) and report a confidence interval for the parameter describing the difference, compare with the result above. Here p_0 indicate the risk of developing AIDS in the control group and p_1 indicate the risk of developing AIDS in the AZT treatment group.

$$p_0 = \frac{e^{\beta_0}}{1 + e^{\beta_0}}$$

$$p_1 = \frac{e^{\beta_0 + \beta_1}}{1 + e^{\beta_0 + \beta_1}}$$

```
#Estimate parameters in the model and report a confidence interval for the parameter
#describing the difference, compare with the result above.
#p_0: Probability of aids in control group
#p_1: Probability of aids in treatment group

#calculate likelihood
nll.p_0 <- function(beta, x = log.data$AIDS_yes[2], n = log.data$n[2]){
  p <- exp(beta)/(1+exp(beta))
  nll <- -sum(dbinom(x, size = n, prob = p, log = T))
  return(nll)
}
opt.p_0 <- nlminb(start = 1, objective = nll.p_0, x = log.data$AIDS_yes[2], n = log.data$n[2])
beta_0 <- opt.p_0$par

nll.p_1 <- function(beta_1, beta_0, x = log.data$AIDS_yes[1], n = log.data$n[1]){
```

```

p <- exp(beta_0+beta_1)/(1+exp(beta_0+beta_1))
nll <- -sum(dbinom(x, size = n, prob = p, log = T))
}
opt.p_1 <- nlminb(start = 1
                  , objective = nll.p_1
                  , beta_0 = beta_0
                  , x = log.data$AIDS_yes[1]
                  , n = log.data$n[1])
beta_1 <- opt.p_1$par

(p_0 <- exp(beta_0)/(1 + exp(beta_0)))

```

```
## [1] 0.2619047
```

```
(p_1 <- exp(beta_0 + beta_1) / (1 + exp(beta_0 + beta_1)))
```

```
## [1] 0.1470588
```

Comparing the results to the glm model in R:

```

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 = ", beta_0, ", beta_1 = ", beta_1)))

```

```

## with glm model: -1.03609193168383
## By hand (according to slide 19 lect 4):
## beta_0 = -1.03609206621491, beta_1 = -0.721765851664904 with glm model: -0.721765985868547
## By hand (according to slide 19 lect 4):
## beta_0 = -1.03609206621491, beta_1 = -0.721765851664904NULL

```

```
#summary(fit.glm)
```

```

#Profile likelihoods
prof.b0 <- function(beta0, x = log.data$AIDS_yes[2], n = log.data$n[2]){
  p <- exp(beta0)/(1+exp(beta0))
  return(-sum(dbinom(x, size = n, prob = p, log = T)))
}

prof.b1 <- function(beta1, beta0, x = log.data$AIDS_yes[1], n = log.data$n[1]){
  p <- exp(beta0+beta1)/(1+exp(beta0+beta1))
  return(-sum(dbinom(x, size = n, prob = p, log = T)))
}

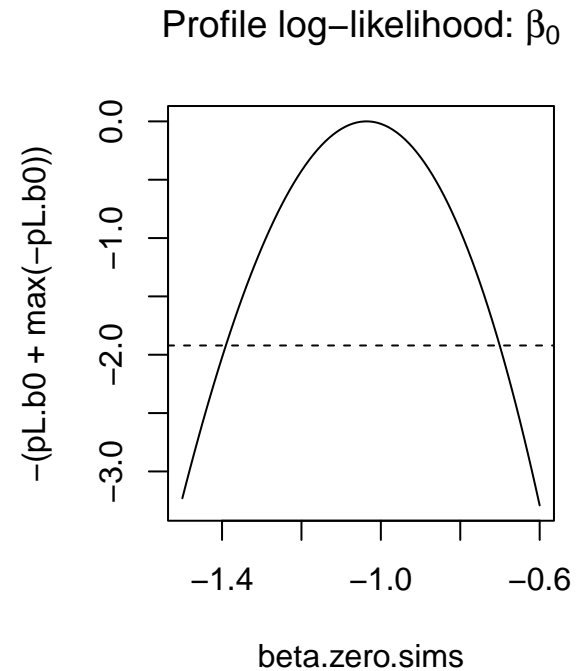
#beta intervals for examination
beta.zero.sims <- seq(-1.5,-0.6,0.01)

```

```

beta.one.sims <- seq(-1.3,-0.2,0.01)
#calculate profile likelihoods
pL.b0 <- sapply(beta.zero.sims, FUN = prof.b0)
pL.b1 <- sapply(beta.one.sims, FUN = prof.b1, beta0 = beta_0)
#plots
par(mfrow=c(1,2))
plot(beta.zero.sims
      , -(pL.b0+max(-pL.b0))
      , "l"
      , main = TeX("Profile log-likelihood: $\beta_0$"))
abline(h = -qchisq(0.95, df = 1)/2, lty = "dashed")
plot(beta.one.sims
      , -(pL.b1+max(-pL.b1))
      , "l"
      , main = TeX("Profile log-likelihood: $\beta_1$"))
abline(h = -qchisq(0.95, df = 1)/2, lty = "dashed")

```



Calculating confidence intervals for the two beta parameters

From these figures it can be concluded that the quadratic approximation of the CI through use of Fischers information matrix, is a sufficiently good approximation.

The Wald and likelihood-based confidence intervals for the two $\beta_i, i \in (0, 1)$ parameters can be seen printed below.

```

sd_0 <- as.numeric(sqrt(solve(hessian(beta_0, func = nll.p_0))))
sd_1 <- as.numeric(sqrt(solve(hessian(beta_1, func = nll.p_1, beta_0 = beta_0))))

#Wald 95% CIs and profile-likelihoods with approx 95% CI
W.CI.0 <- round(beta_0 + c(-1,1)*qnorm(0.975)*sd_0,4)
W.CI.1 <- round(beta_1 + c(-1,1)*qnorm(0.975)*sd_1,4)

```

```

#Direkte numerisk approksimation:
CI.0 <- round(c(min(beta.zero.sims[-(pL.b0+max(-pL.b0)) > -qchisq(0.95, df = 1)/2])
               ,max(beta.zero.sims[-(pL.b0+max(-pL.b0)) > -qchisq(0.95, df = 1)/2])), 4)
CI.1 <- round(c(min(beta.one.sims[-(pL.b1+max(-pL.b1)) > -qchisq(0.95, df = 1)/2])
               ,max(beta.one.sims[-(pL.b1+max(-pL.b1)) > -qchisq(0.95, df = 1)/2])), 4)

cat(paste("Wald Confidence intervals:"
          ,paste("\nbeta_0 = ", round(beta_0, 4), " [95% CI: ", W.CI.0[1],",", " ", W.CI.0[2],"]")
          ,paste("\nbeta_0 = ", round(beta_1, 4), " [95% CI: ", W.CI.1[1],",", " ", W.CI.1[2],"]")
          ,"\nLikelihood-based Confidence intervals:"
          ,paste0("\nbeta_0 = ", round(beta_0, 4), " [95% CI: ", CI.0[1],",", " ", CI.0[2],"]")
          ,paste0("\nbeta_0 = ", round(beta_1, 4), " [95% CI: ", CI.1[1],",", " ", CI.1[2],"]"))))

```

```

## Wald Confidence intervals:
## beta_0 = -1.0361 [95% CI: -1.38 , -0.6922 ]
## beta_0 = -0.7218 [95% CI: -1.1462 , -0.2973 ]
##
## Likelihood-based Confidence intervals:
## beta_0 = -1.0361 [95% CI: -1.39, -0.71]
## beta_0 = -0.7218 [95% CI: -1.16, -0.32]

```

Comparing these to eachother, we see that the Wald CI is a very good approximation of the actual CIs. However, when comparing the estimates from our glm model we see that the 95% CI for AZT is wider for the model estimate...

```
confint(fit.glm)
```

```

##                2.5 %      97.5 %
## (Intercept) -1.390358 -0.7006773
## AZT         -1.279159 -0.1827049

```

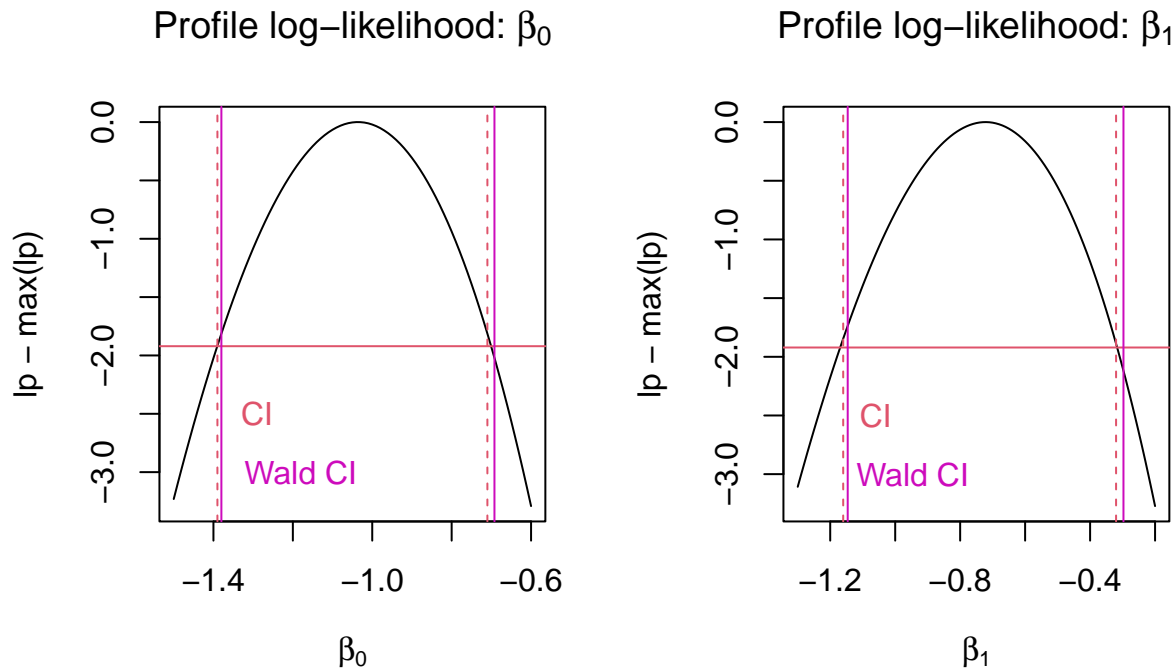
Below can be seen the profile log-likelihoods for the two parameters, alongside their CIs.

```

par(mfrow = c(1,2))
plot(beta.zero.sims
     , -(pL.b0+max(-pL.b0))
     , "1"
     ,main = TeX("Profile log-likelihood: $\\beta_0$")
     ,xlab = expression(beta[0])
     ,ylab = "lp - max(lp)")
abline(h = -qchisq(0.95, df = 1)/2, col = 2)
abline(v = c(W.CI.0), col = 6)
text(x = W.CI.0[1]+0.2, y = -3, "Wald CI", col = 6)
text(x = CI.0[1]+0.1, y = -2.5, "CI", col = 2)
abline(v = c(CI.0), lty = "dashed", col = 2)
plot(beta.one.sims
     , -(pL.b1+max(-pL.b1))
     , "1"
     ,main = TeX("Profile log-likelihood: $\\beta_1$")
     ,xlab = expression(beta[1])
     ,ylab = "lp - max(lp)")

```

```
abline(h = -qchisq(0.95, df = 1)/2, col = 2)
abline(v = c(W.CI.1), col = 6)
text(x = W.CI.1[1]+0.2, y = -3, "Wald CI", col = 6)
text(x = CI.1[1]+0.1, y = -2.5, "CI", col = 2)
abline(v = c(CI.1), lty = "dashed", col = 2)
```



Analysis of the Survival Time 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)
```

Read the data actg320.txt into R. If you are using RStudio you can use the “Import Dataset” button.

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

How many patients got AIDS or died in the two treatment groups? What is the proportion of patients that got AIDS or died in the two group? Other relevant number that could be calculated?

```
## # A tibble: 2 x 4
##   tx `Got AIDS or DIED` Proportion `Participants Given the Treatment`
##   <int>          <int>      <dbl>                <int>
## 1     0             63    0.109                 577
## 2     1             33    0.0575                574
```

Fit an exponential distribution, using numerical methods, to the time of event (time) in the data set, remember to take into account that some of the data is censored (i.e. we only know that the time to the event is longer than the reported time). 1: Using all data (i.e. ignore the treatment effect) 2: Separately for the two treatments Fitting an exponential model, where we account for censoring:

```
#page 310 in the course textbook
theta.hat.both <- sum(actg$time)/sum(actg$event)
```

Theta for each group individually:

```
#With treatment:
actg_tx <- filter(actg, tx == 1)
theta.hat.tx <- sum(actg_tx$time)/sum(actg_tx$event)
sigma.hat.tx <- theta.hat.tx/sum(actg$event)^(1/2)

#without treatment:
actg_no_tx = filter(actg, tx == 0)
theta.hat.no.tx <- sum(actg_no_tx$time)/sum(actg_no_tx$event)
sigma.hat.no.tx <- theta.hat.tx/sum(actg$event)^(1/2)

#Wald statistic for comparison
(theta.hat.tx - theta.hat.no.tx)/sqrt(sigma.hat.tx^2 + sigma.hat.no.tx^2)
```

```
## [1] 3.484908
```

```
#this is larger than 2 and thus statistically significant.
```

Fitting an exponential model to time for both and for each treatment where we ignore censoring

```
#only use times for event = 1, to filter out all the time of event indices with are longer than the rep
#given the fact that the participants in the event = 0 group, has not 'experienced' the event yet.
#Ved sgu ikke om ovenstående er en passende antagelse....
```



```

actg_event <- actg %>%
  filter(event == 1)

both <- nlminb(start = 2
              , objective = testDistribution
              , x = actg_event$time
              , distribution = "exponential")

#separate exponential models
t1 <- nlminb(start = 2
            , objective = testDistribution
            , x = filter(actg_event, tx == 1)$time
            , distribution = "exponential")

t0 <- nlminb(start = 2
            , objective = testDistribution
            , x = filter(actg_event, tx == 0)$time
            , distribution = "exponential")

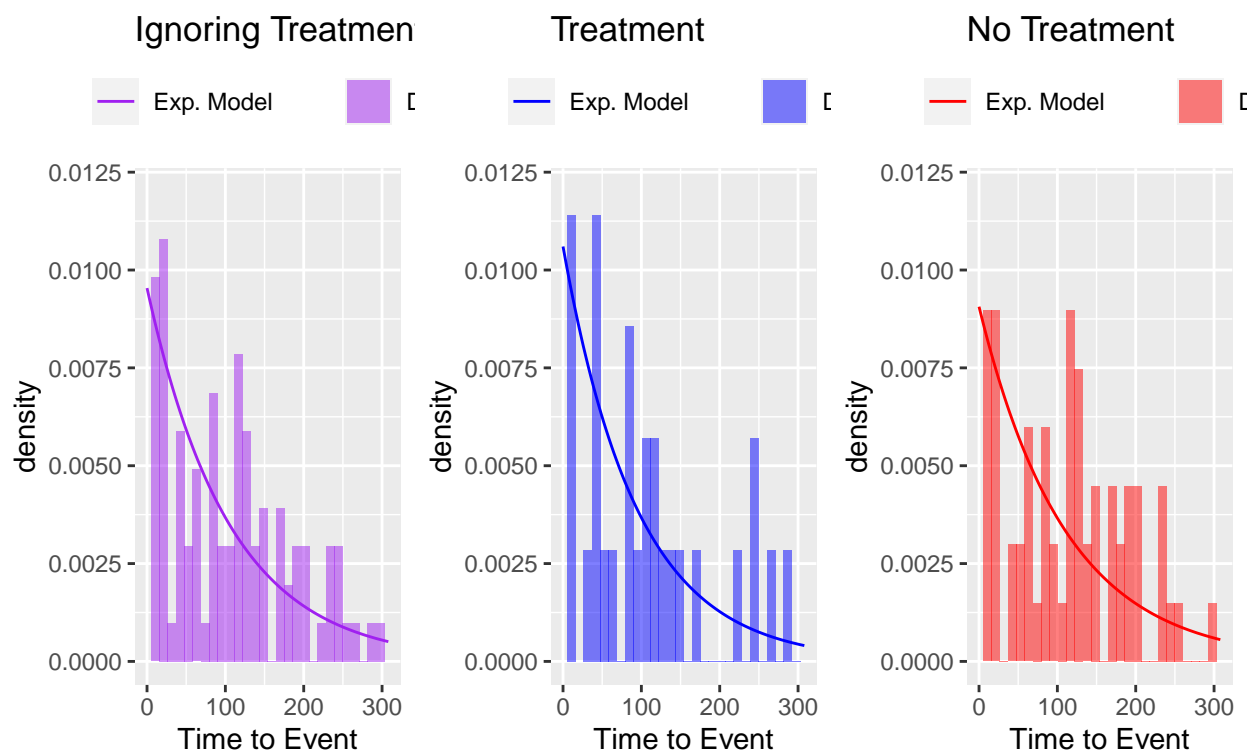
#Potato plots:
p.both <- ggplot(actg_event)+
  geom_histogram(aes(x = time, y = ..density.., fill = "Data"), alpha = 0.5)+
  stat_function(aes(colour = "Exp. Model"), fun = dexp, n = dim(actg_event)[1], args = list(rate = both))
  ggtitle("Ignoring Treatment Effect")+
  theme(legend.position = "top")+
  lims(x = c(0,max(actg_event$time)+10), y = c(0,0.012))+
  labs(fill = "", colour = "", x = "Time to Event")+
  scale_colour_manual(values = "purple")+
  scale_fill_manual(values = "purple")

p.t1 <- ggplot(actg_event[actg_event$tx == 1,])+
  geom_histogram(aes(x = time, y = ..density.., fill = "Data"), alpha = 0.5)+
  stat_function(aes(colour = "Exp. Model"), fun = dexp, n = dim(actg_event)[1], args = list(rate = t1$par))
  ggtitle("Treatment")+
  theme(legend.position = "top")+
  lims(x = c(0,max(actg_event$time)+10), y = c(0,0.012))+
  labs(fill = "", colour = "", x = "Time to Event")+
  scale_colour_manual(values = "blue")+
  scale_fill_manual(values = "blue")

p.t2 <- ggplot(actg_event[actg_event$tx == 0,])+
  geom_histogram(aes(x = time, y = ..density.., fill = "Data"), alpha = 0.5)+
  stat_function(aes(colour = "Exp. Model"), fun = dexp, n = dim(actg_event)[1], args = list(rate = t0$par))
  ggtitle("No Treatment")+
  theme(legend.position = "top")+
  lims(x = c(0,max(actg_event$time)+10), y = c(0,0.012))+
  scale_colour_manual(values = "red")+
  labs(fill = "", colour = "", x = "Time to Event")+
  scale_fill_manual(values = "red")

grid.arrange(p.both, p.t1, p.t2, nrow = 1)

```



```
#Likelihood Ratio Test (LRT) comparison
#one model:
chi_squared <- - 2 * ((t1$objective + t0$objective) - both$objective)
(p_value <- 1 - pchisq(chi_squared, df = 1))
```

Compared the likelihood for the above models and conclude

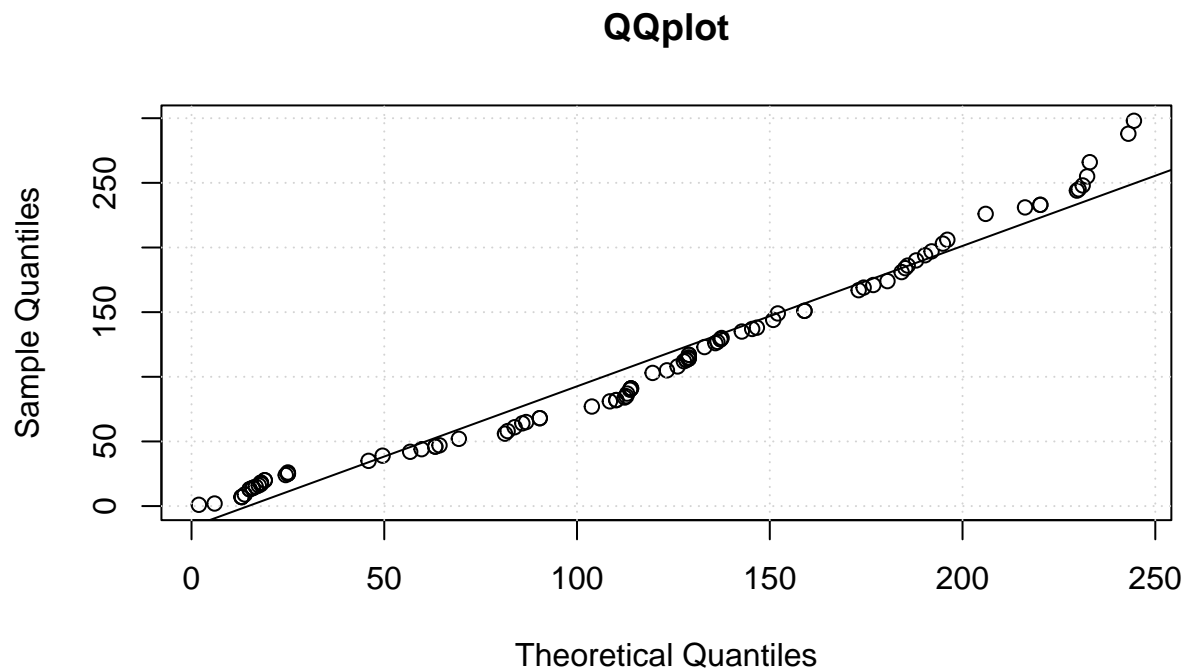
```
## [1] 0.4697636
```

```
#no difference as p_value: 0.46 > 0.05.
```

From the above calculation of the p-value = {r} p_value, we see that there's - according to this model - no significant difference is achieved by using the treatment. However, this model is flawed, as it does not account for all the censored datapoints. This will be addressed in the coming analysis, where we use the kaplan meier survival analysis.

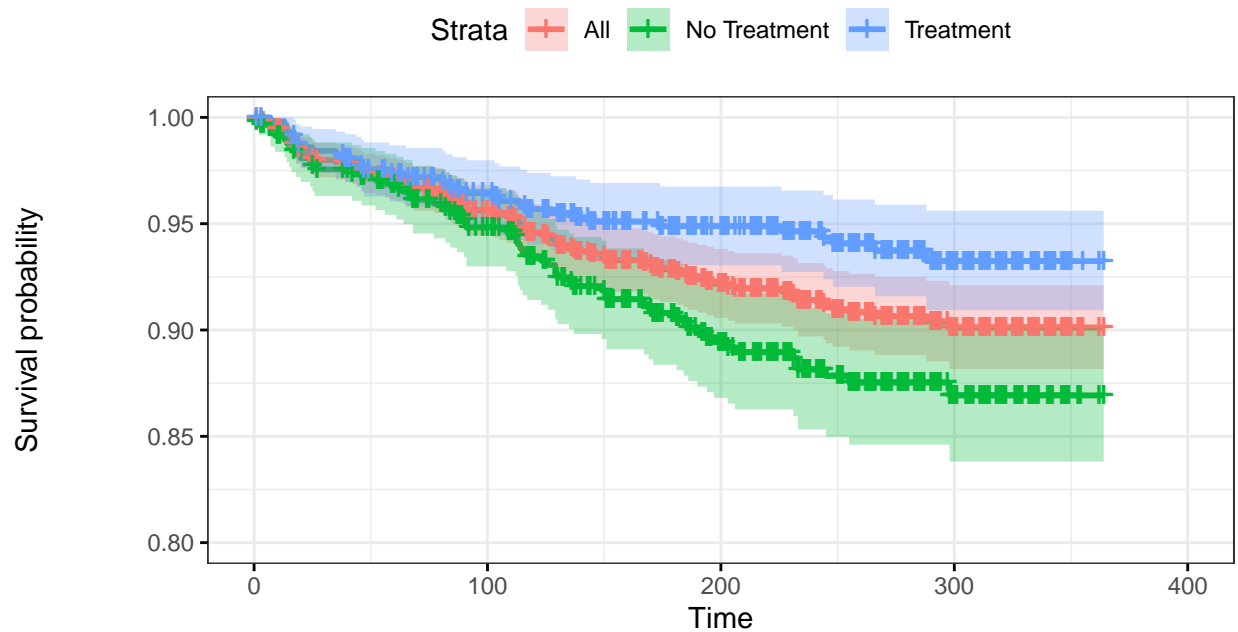
Below can be seen a QQ-plot of the model, which also clearly shows that the model is unable to capture the underlying distribution of the data.

```
theoretical <- quantile(x = actg_event$time, probs = pexp(q = actg_event$time, rate = both$par))
plot(theoretical, actg_event$time, main = "QQplot", xlab = "Theoretical Quantiles"
     ,ylab = "Sample Quantiles")
grid()
abline(lm(actg_event$time ~ theoretical))
```



Formulate a model where one parameter indicate the treatment effect, find the MLE and compare with the result above. (e.g. $E[T] = e_0^\beta$ if control group and $E[T] = e^{\beta_0 + \beta_1}$ if treatment group)

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

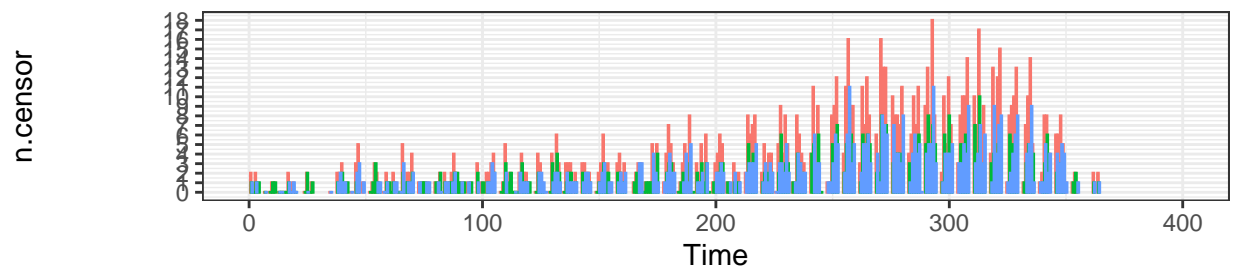


Number at risk: n (%)

All	51 (100)	1015 (88)	794 (69)	296 (26)	0 (0)
No Treatment	77 (100)	500 (87)	380 (66)	138 (24)	0 (0)
Treatment	74 (100)	515 (90)	414 (72)	158 (28)	0 (0)

Time

Number of censoring



```
fit <- survreg(Surv(time, event) ~ tx, data = actg,
               dist = "exponential")
summary(fit)
```

```
##
## Call:
## survreg(formula = Surv(time, event) ~ tx, data = actg, dist = "exponential")
##           Value Std. Error      z      p
## (Intercept)  7.624      0.126 60.52 <2e-16
## tx           0.699      0.215  3.25 0.0011
```

```
##
## Scale fixed at 1
##
## Exponential distribution
## Loglik(model)= -851   Loglik(intercept only)= -856.6
##  Chisq= 11.18 on 1 degrees of freedom, p= 0.00083
## Number of Newton-Raphson Iterations: 6
## n= 1151
```

```
confint(fit)
```

```
##                2.5 %    97.5 %
## (Intercept) 7.3774309 7.871295
## tx          0.2780026 1.120341
```

```
#Overvej residual plot
```

```
#Ifølge ovenstående:
```

```
#beta0 = 7.62 95% CI [7.38; 7.87]
```

```
#beta1 = 0.699 85% CI [0.28; 1.12]
```

```
# => Significant difference.
```

```
#ifølge ovenstående er der statistisk signifikant forskel. Herunder regnes i hånden i stedet, så vi ved hvad der foregår.
```

Are the effect of the treatment statistically significant?

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

```
## 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
```

Yes it is.

Same calculations but now performed by hand:

```
#I hånden (jvf. slides fra uge 7):
```

```
#model:  $T = \exp(B_0 + B_1 \cdot tx) \cdot \epsilon$ ,  $\epsilon \sim \exp(1)$ 
```

```
#Der kan opstilles to forskellige modeller afhængigt af  $tx = 0$  eller  $tx = 1$ .
```

```
#tx = 0:  $E[T] = \exp(b_0) \cdot \epsilon$ 
```

```
#tx = 1:  $E[T] = \exp(b_0 + b_1) \cdot \epsilon$ 
```

```
#Likelihood
```

```
nll.exp <- function(beta, time = actg$time, event = actg$event, treatment = actg$tx){
  beta0 <- beta[1]
```

```

#dont want to make two functions so let beta1 = 0 if no treatment is not considered/used:
if (max(treatment) == 0){
  beta1 <- 0
} else {
  beta1 <- beta[2]
}

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

beta_hat <- nlminb(start = c(1,1)
                  , objective = nll.exp
                  , time = actg$time
                  , event = actg$event
                  , treatment = actg$tx)
beta_hat$par

```

```
## [1] 7.6243647 0.6991732
```

```

#Comparing likelihoods with the result from bullet-point 4
beta_hat$objective #Ved ikke lige om der skal sammenlignes med de to modeller eller den ene? ahh

```

```
## [1] 851.0115
```

```

#måske skal man undersøge om begge værdier er statistisk signifikante således at vi kan argumentere for
#at der er tale om at behandlingen virker og sammenligne dette resultat med bullet-point 4.

```

```

#Calculate LRT
#optimise model without beta1 (no treatment):
beta_no_treatment_effect <- nlminb(start = 1
                                   , objective = nll.exp
                                   , time = actg$time
                                   , event = actg$event
                                   , treatment = rep(0, length(actg$tx)))

beta_no_treatment_effect$par

```

```
## [1] 7.922914
```

```

#LRT:
chi_squared <- - 2 * (beta_hat$objective - beta_no_treatment_effect$objective)
(p_value <- 1 - pchisq(chi_squared, df = 1))

```

```
## [1] 0.0008284118
```

```

#Here, we see that the treatment effect is statistically significant.

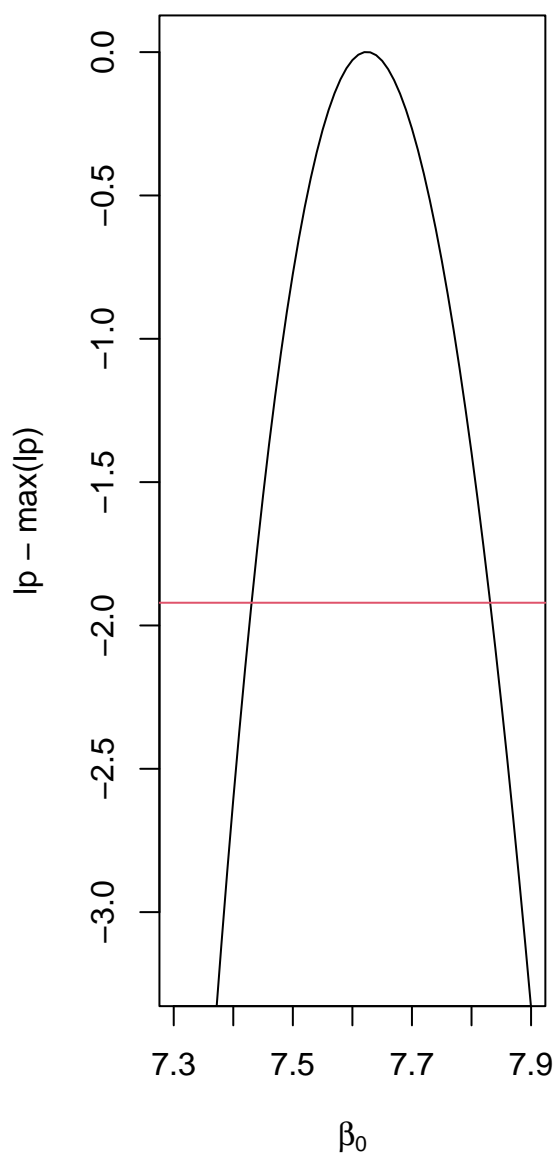
```

```
#### Bullet point 6 - Wald CI for the treatment parameters beta0 and beta1 ####
#Calculate profile likelihoods to ensure that the quadratic approximation by using Fischers Information
#is acceptable.
```

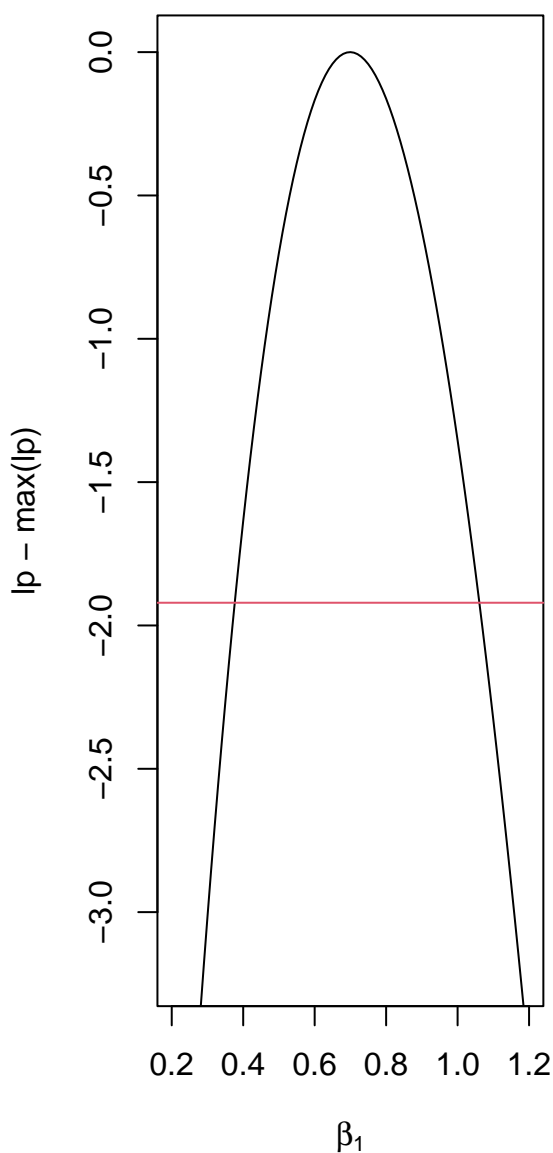
```
beta.zero.sims <- seq(7.3,7.9,0.01)
beta.one.sims <- seq(0.2,1.2,0.01)
pL.beta0 <- apply(X = data.frame(beta.zero.sims,beta_hat$par[2]), MARGIN = 1 , FUN = nll.exp, time = act)
pL.beta1 <- apply(X = data.frame(beta_hat$par[1],beta.one.sims), MARGIN = 1 , FUN = nll.exp, time = act)

par(mfrow=c(1,2))
plot(beta.zero.sims
      , -(pL.beta0+max(-pL.beta0))
      , "1"
      ,main = "Profile likelihood for Beta 0"
      ,xlab = expression(beta[0])
      ,ylab = "lp - max(lp)"
      ,ylim = c(-3.2,0))
abline(h = -qchisq(0.95, df = 1)/2, col = 2)
plot(beta.one.sims
      , -(pL.beta1+max(-pL.beta1))
      , "1"
      ,main = "Profile likelihood for beta 1"
      ,xlab = expression(beta[1])
      ,ylab = "lp - max(lp)"
      ,ylim = c(-3.2,0))
abline(h = -qchisq(0.95, df = 1)/2, col = 2)
```

Profile likelihood for Beta 0



Profile likelihood for beta 1



```
#CI:
sd <- as.numeric(sqrt(diag(solve(hessian(beta_hat$par, func = nll.exp))))))

#Wald 95 procent CIs and profile-likelihoods with approx 95 procent CI
#Måske er der et eller andet i vejen med de her WALD CIs
Wald.CI <- round(beta_hat$par + matrix(c(-1,1), 2,2, byrow = T) * matrix(qnorm(0.975)*sd, 2,2, byrow = T), 2)

#Direkte numerisk approksimation:
```



```

CI.0 <- c(min(beta.zero.sims[-(pL.beta0+max(-pL.beta0)) > -qchisq(0.95, df = 1)/2])
          ,max(beta.zero.sims[-(pL.beta0+max(-pL.beta0)) > -qchisq(0.95, df = 1)/2]))
CI.1 <- c(min(beta.one.sims[-(pL.beta1+max(-pL.beta1)) > -qchisq(0.95, df = 1)/2])
          ,max(beta.one.sims[-(pL.beta1+max(-pL.beta1)) > -qchisq(0.95, df = 1)/2]))

cat(paste("Wald Confidence intervals:"
          ,paste("\nbeta_0 = ", round(beta_hat$par[1], 4), " [95% CI: ", Wald.CI[1,1],", ", Wald.CI[1,2],"]"
          ,paste("\nbeta_1 = ", round(beta_hat$par[2], 4), " [95% CI: ", Wald.CI[2,1],", ", Wald.CI[2,2],"]"
          ,"\n\nLikelihood-based Confidence intervals:"
          ,paste0("\nbeta_0 = ", round(beta_hat$par[1], 4), " [95% CI: ", CI.0[1],", ", CI.0[2],"]")
          ,paste0("\nbeta_1 = ", round(beta_hat$par[2], 4), " [95% CI: ", CI.1[1],", ", CI.1[2],"]"))))

```

Find the Wald confidence interval for the treatment parameter in the model above.

```

## Wald Confidence intervals:
## beta_0 = 7.6244 [95% CI: 7.3774 , 7.8713 ]
## beta_1 = 0.6992 [95% CI: 0.278 , 1.1203 ]
##
## Likelihood-based Confidence intervals:
## beta_0 = 7.6244 [95% CI: 7.44, 7.83]
## beta_1 = 0.6992 [95% CI: 0.38, 1.06]

```

Comparing with the results from the survfit function:

```
confint(fit)
```

```

##                2.5 %    97.5 %
## (Intercept) 7.3774309 7.871295
## tx          0.2780026 1.120341

```

Same as the wald CI. The likelihood based CI is a bit more narrow.

```

par(mfrow = c(1,2))
plot(beta.zero.sims
     , -(pL.beta0+max(-pL.beta0))
     , "l"
     ,main = "Profile likelihood for Beta 0"
     ,xlab = expression(beta[0])
     ,ylab = "lp - max(lp)"
     ,ylim = c(-3.2,0))
abline(h = -qchisq(0.95, df = 1)/2, col = 2)
abline(v = Wald.CI[1,], col = 6)
text(x = Wald.CI[1,1]+.15, y = -2.4, "Wald CI", col = 6)
text(x = CI.0[1]+.05, y = -2.5, "CI", col = 2)
abline(v = c(CI.0), lty = "dashed", col = 2)
plot(beta.one.sims
     , -(pL.beta1+max(-pL.beta1))
     , "l"
     ,main = "Profile likelihood for beta 1"

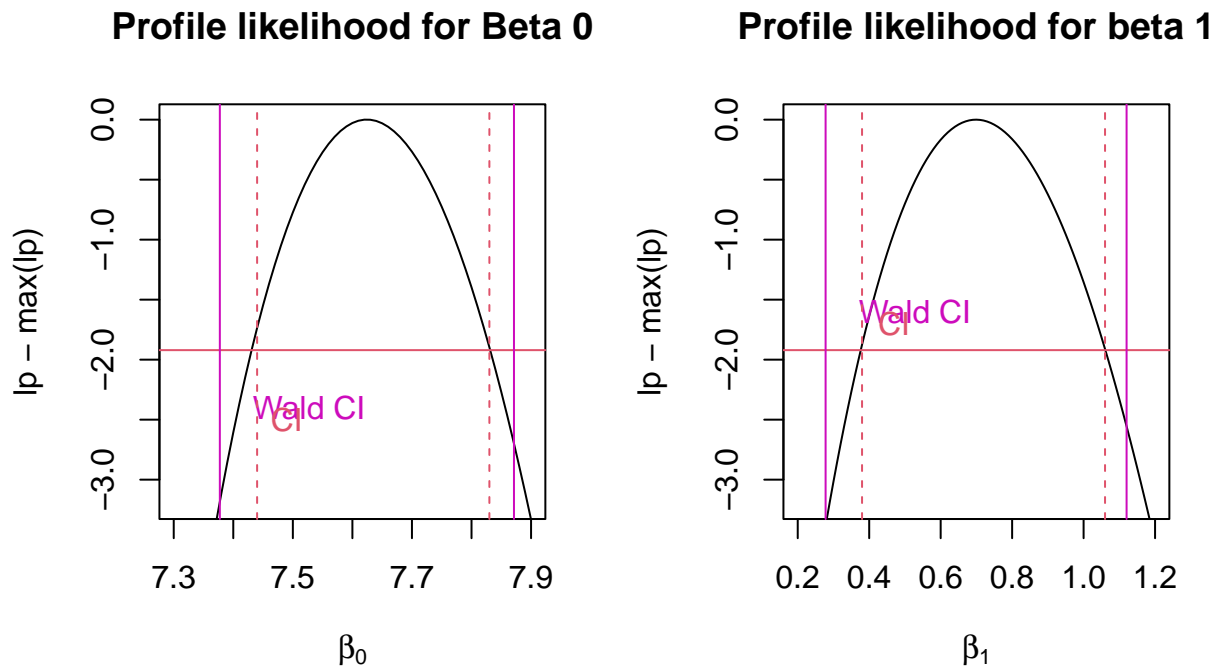
```

```

, xlab = expression(beta[1])
, ylab = "lp - max(lp)"
, ylim = c(-3.2, 0))
abline(h = -qchisq(0.95, df = 1)/2, col = 2)
abline(v = Wald.CI[2,], col = 6)
text(x = Wald.CI[2,1]+0.25, y = -1.6, "Wald CI", col = 6)
text(x = CI.1[1]+0.09, y = -1.7, "CI", col = 2)
abline(v = c(CI.1), lty = "dashed", col = 2)

```

Derive the theoretical results for the models above, including the standard error estimates, use this to formulate and implement the profile likelihood function for the treatment parameter.



(Have not included our analysis based on the weibull distribution)