```r
getres = function(heartdat){
  if(ncol(heartdat) == 3) heartdat = cbind(heartdat, cen = 1)
  if(!is.matrix(heartdat)) heartdat = as.matrix(heartdat)
  heartdat_uncen = heartdat[heartdat[,4] == 1,]
  heartdat_cen = heartdat[heartdat[,4] == 0,]

  loglik_cen = function(pars, heartdat){
    theta = pars[1:2]
    sigma = pars[3]
    if(length(heartdat) == 0) return(0)
    x = heartdat[,1:2]
    y = heartdat[,3]
    h = exp(x %*% theta)
    n = nrow(heartdat)
    z = (y / h)^(1 / sigma)
    return(sum(- z))
  }

  loglik_uncen = function(pars, heartdat){
    theta = pars[1:2]
    sigma = pars[3]
    if(length(heartdat) == 0) return(0)
    x = heartdat[,1:2]
    y = heartdat[,3]
    h = exp(x %*% theta)
    n = nrow(heartdat)
    zL = ((y - 10^(-3))/ h)^(1 / sigma)
    zU = ((y + 10^(-3))/ h)^(1 / sigma)
    #z = (y / h)^(1 / sigma)
    #return(sum(-log(sigma) + (1 / sigma - 1) * log(z) - z / sigma))
    return(sum(log(exp(-zL) - exp(-zU))))
  }

  loglik = function(pars, heartdat){
    heartdat_uncen = heartdat[heartdat[,4] == 1,]
    heartdat_cen = heartdat[heartdat[,4] == 0,]
    return(loglik_uncen(pars, heartdat_uncen) +
             loglik_cen(pars, heartdat_cen))
  }

  loglik_neg = function(pars, heartdat){
    return(-loglik(pars, heartdat))
  }

  theta_ini = unname(coef(lm(log(heartdat[,"y"]) ~ 0 + heartdat[,c("trt", "ldl")])))
  sigma_vec = seq(from = 0.01, to = 10, length = 100)
  loglik_vec = sapply(sigma_vec, function(k) loglik(c(theta_ini, k), heartdat))
  #plot(sigma_vec, loglik_vec)
  sigma_ini = sigma_vec[which.max(loglik_vec)]
  pars_ini = c(theta_ini, sigma_ini)

  res = optim(par = pars_ini, fn = loglik_neg, heartdat = heartdat, hessian = TRUE)
  pars = res$par
```

```
  #res$hessian #Information matrix
  var_est = diag(solve(res$hessian)) #Estimated variance
  ME = sqrt(var_est) * qnorm(1 - 0.025) # Margin of Error
  res_table = cbind(pars - ME, pars, pars + ME)
  row.names(res_table) <- c("theta1", "theta2", "sigma")
  colnames(res_table) <- c("95%LL", "Estimates", "95%UL")

  return(list(res_table = res_table, loglik = loglik(res$par, heartdat)))
}

heartdat = read.csv("heartdat70_variable.txt", sep = " ")
getres(heartdat)

sprintf("heartdat%g_variable.txt", 10)
cens_vec = seq(from = 10, to = 70, by = 10)
sapply(cens_vec, function(n)
  sprintf("heartdat%g_variable.txt", n))

filename_vec = c("heartdat0.txt", sapply(cens_vec, function(n)
  c(sprintf("heartdat%g_variable.txt", n), sprintf("heartdat%g_fixed.txt", n))))

heartdat = read.csv(filename_vec[1], sep = " ")

list_res = lapply(filename_vec, function(name){
  heartdat = read.csv(name, sep = " ")
  return(getres(heartdat))
})

list_res[[1]]
table1 = t(sapply(2:8, function(k) list_res[[k]]$res_table[,2]))
row.names(table1) <- cens_vec

cens_vec[1]
for(i in 1:7){
print(sprintf("variable censoring; censoring rate = %d", cens_vec[i]))
print(list_res[[i+1]]$res_table)
}

for(i in 1:7){
  print(sprintf("fixed censoring; censoring rate = %d", cens_vec[i]))
  print(list_res[[i+8]]$res_table)
}

table2 = t(sapply(9:15, function(k) list_res[[k]]$res_table[,2]))
row.names(table2) <- cens_vec


png(file="theta1_variable.png")
tmpmat = sapply(2:8, function(k) list_res[[k]]$res_table[1,])
L = tmpmat[1,]
x = tmpmat[2,]
U = tmpmat[3,]
plot(U - L ~ cens_vec, xlab = "censoring rate", ylab = "width",
```

```
     main = "theta1; variable")
dev.off()

png(file="theta1_fixed.png")
tmpmat = sapply(9:15, function(k) list_res[[k]]$res_table[1,])
L = tmpmat[1,]
x = tmpmat[2,]
U = tmpmat[3,]
plot(U - L ~ cens_vec, xlab = "censoring rate", ylab = "width",
     main = "theta1; fixed")
dev.off()

png(file="theta2_variable.png")
tmpmat = sapply(2:8, function(k) list_res[[k]]$res_table[2,])
L = tmpmat[1,]
x = tmpmat[2,]
U = tmpmat[3,]
plot(U - L ~ cens_vec, xlab = "censoring rate", ylab = "width",
     main = "theta2; variable")
dev.off()

png(file="theta2_fixed.png")
tmpmat = sapply(9:15, function(k) list_res[[k]]$res_table[2,])
L = tmpmat[1,]
x = tmpmat[2,]
U = tmpmat[3,]
plot(U - L ~ cens_vec, xlab = "censoring rate", ylab = "width",
     main = "theta2; fixed")
dev.off()

png(file="sigma_variable.png")
tmpmat = sapply(2:8, function(k) list_res[[k]]$res_table[3,])
L = tmpmat[1,]
x = tmpmat[2,]
U = tmpmat[3,]
plot(U - L ~ cens_vec, xlab = "censoring rate", ylab = "width",
     main = "sigma; variable")
dev.off()

png(file="sigma_fixed.png")
tmpmat = sapply(9:15, function(k) list_res[[k]]$res_table[3,])
L = tmpmat[1,]
x = tmpmat[2,]
U = tmpmat[3,]
plot(U - L ~ cens_vec, xlab = "censoring rate", ylab = "width",
     main = "sigma; fixed")
dev.off()
```