

Trying to model mortality rate after prehospital whole blood transfusion by estimating the maximum likelihood.

This phenomenon is assumed to follow a binomial distribution so that the probability for single event p_i in time point t_i is defined as $p_i = a \cdot \exp(-b \cdot t_i)$.

```
library("readxl")
library("ggplot2")
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

library("npreg")

## Warning: package 'npreg' was built under R version 4.2.1

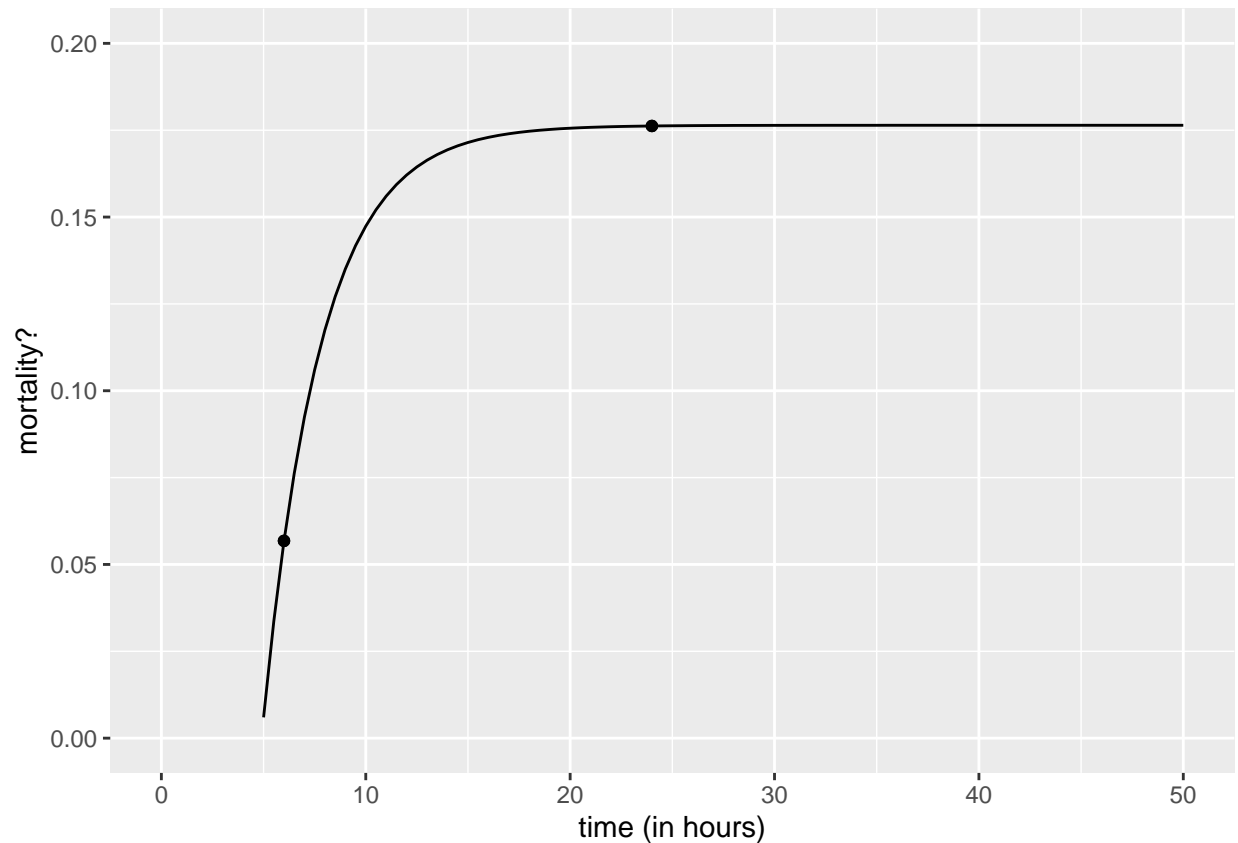
# Defining function to calculate the binomial loglikelihood
# Input: generic parameter vector theta and data including covariates (explanatory variables) and corre
bin.llhood<-function(theta,data){
  p_vec<-theta[1]-exp(-theta[2]*data[,1])
  logl<-sum(data[,2]*log(p_vec)+data[,3]*log(1-p_vec))
  return(-logl)
}

# Reading the data from excel (time in hours!)
df <- read_excel("C:\\\\Projektit\\whole_blood_research\\excel\\Emergencyprocess_PHBT_splines.xlsx", sheet = "Sheet1")

# Filtering to get just the data from whole blood
# Selecting only columns that are useful for the bin.llhood:
# Time in the first column (covariate)
# Number of patients that died in the second column ("successful events")
# Number of patients that survived in the third column ("failed events")
WB_data <- select(filter(df, products == "LTOWB" ), time, n_dead, n_survived)
print(WB_data)

## # A tibble: 7 x 3
##   time n_dead n_survived
##   <dbl> <dbl>     <dbl>
## 1     6     11         96
## 2    24     18         89
## 3     6      4        131
## 4    24     12        123
## 5     6      4         88
## 6    24     13         79
## 7   720     32         60

# Estimating the maximum likelihood using optim()
# Defining starting values for the optimization
a_b <- optim(c(0.01, 1), bin.llhood, data = WB_data, method="BFGS")
```

The first curve seems somewhat reasonable, but the mortality rate is probably not as high as in the real life? (The third point at 720 hours is somewhere in the future, it didn't disappear!)

The curve changes if the values in `optim()` are changed. Testing some other values to see the difference and to compare the outcomes.

```
a_b <- optim(c(0.1, 10), bin.llhood, data = WB_data, method="BFGS")
```

```
## Warning in FUN(X[[i]], ...): NaNs produced
```

```
## Warning in FUN(X[[i]], ...): NaNs produced
```

```
## Warning in FUN(X[[i]], ...): NaNs produced
```

```
## Warning in FUN(X[[i]], ...): NaNs produced
```

```
a <- a_b$par[1]
```

```
b <- a_b$par[2]
```

```
ownFun <- function(x){
  probs <- a-exp(-b*x)
  return(probs)
}
```

```
x <- WB_data$time
```

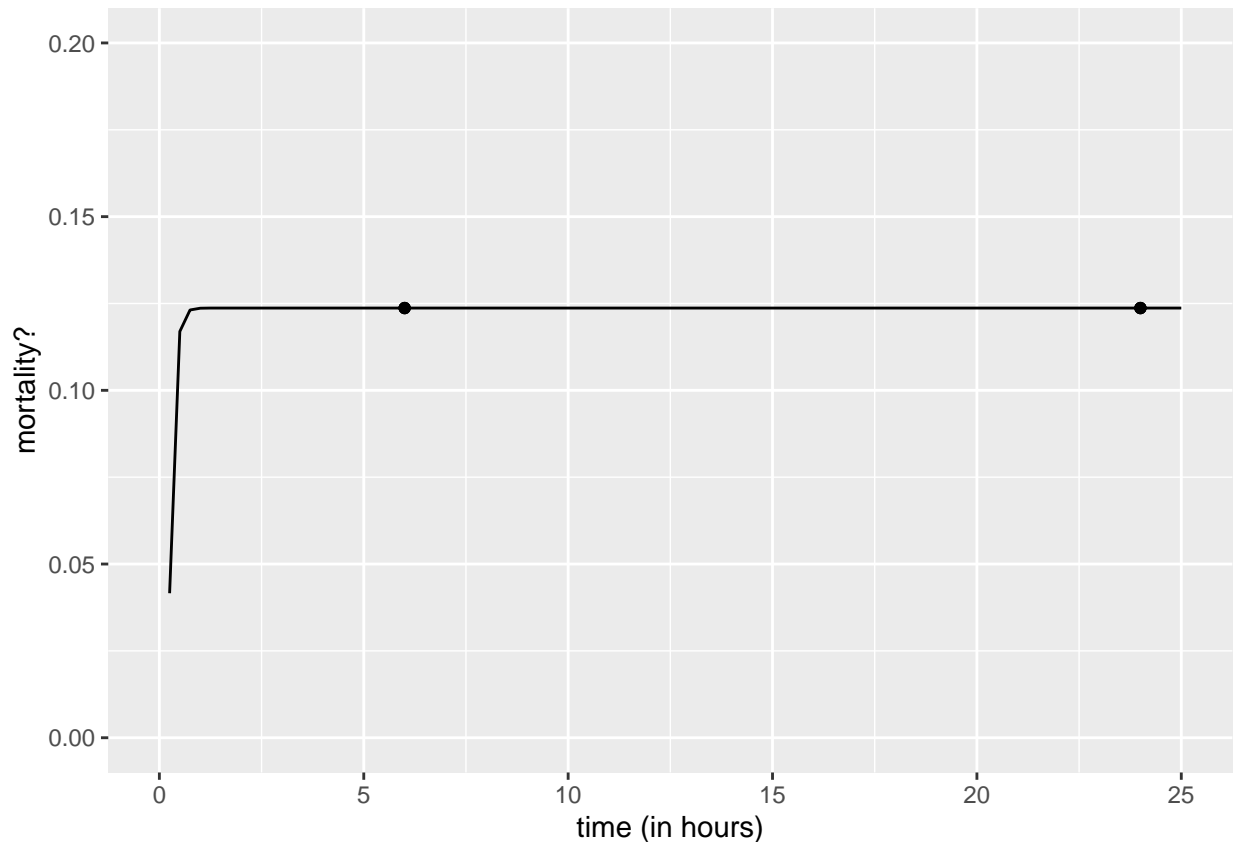
```
y <- ownFun(x)
```

```
df2 <- data.frame(x,y)
```

```
ggplot(data.frame(x), aes(x=x)) +
  geom_point(data = df2, aes(x=x, y=y)) +
  geom_function(fun = ownFun) +
  scale_x_continuous(name = "time (in hours)", limits = c(0, 25)) + # Had to change limits to see anything
  scale_y_continuous(name = "mortality?", limits = c(0, 0.2))
```

Warning: Removed 1 rows containing missing values (geom_point).

Warning: Removed 1 row(s) containing missing values (geom_path).



It seems that the value for “a” is either close to 0.175 or 0.125 when changing the values to start the optimization.

The first plot seems to be better (according to every article the mortality rate doesn’t stop growing just after few hours like in this second plot) so let’s keep those values for `optim()` and reject the second one.