# simple_curve_fitting

Milla Juntunen

2022-07-15

*** CURVE FITTING & SIMULATING NEW DATA ***

This code tries to fit different curves in the existing data about prehospital blood transfusion's effect on mortality rate, pick up the model that is considered the best one and then simulate new data points.

This is done separately to all blood products. At the end all products are combined to see if it makes any difference.

All results are plotted below.

```
source("C:/Projektit/whole_blood_research/time_point_simulation/simulating_timepoints.R")
```

```
##
## 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("readxl")
library("ggplot2")
library("dplyr")
library("npreg")
```

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

```
# Reading the data from excel (time in minutes)
df <- read_excel("C:\\Projektit\\whole_blood_research\\excel\\Emergencyprocess_PHBT_splines.xlsx", shee
```

Simulating new data points. For more information, see simulating_timepoints.Rmd

```
# Making simulated data
simulated_times <- make_timepoints.function(20)
```

```
## Chart A is the chart to follow
## Patient transported to Shock Room!
## 68.22347 is the total time from the 'Risk Analysis'
## 24.45856 is the total time from the 'Infusion Starts'
##
## Chart A is the chart to follow
## Patient transported to Shock Room!
## 88.52801 is the total time from the 'Risk Analysis'
## 43.84568 is the total time from the 'Infusion Starts'
##
```

```
## Chart A is the chart to follow
## Patient transported to Shock Room!
## 81.13964 is the total time from the 'Risk Analysis'
## 39.78225 is the total time from the 'Infusion Starts'
##
## Chart A is the chart to follow
## Patient transported to Shock Room!
## 87.84342 is the total time from the 'Risk Analysis'
## 47.03993 is the total time from the 'Infusion Starts'
##
## Chart B is the chart to follow
## Patient transported to Shock Room!
## 89.89174 is the total time from the 'Risk Analysis'
## 30.52388 is the total time from the 'Infusion Starts'
##
## Chart B is the chart to follow
## Patient transported to Shock Room!
## 79.81837 is the total time from the 'Risk Analysis'
## 22.88065 is the total time from the 'Infusion Starts'
##
## Chart B is the chart to follow
## Patient transported to Shock Room!
## 84.53452 is the total time from the 'Risk Analysis'
## 27.12424 is the total time from the 'Infusion Starts'
##
## Chart A is the chart to follow
## Patient transported to Shock Room!
## 94.37576 is the total time from the 'Risk Analysis'
## 52.73971 is the total time from the 'Infusion Starts'
##
## Chart B is the chart to follow
## Patient transported to Shock Room!
## 82.74505 is the total time from the 'Risk Analysis'
## 25.17064 is the total time from the 'Infusion Starts'
##
## Chart A is the chart to follow
## Patient transported to Shock Room!
## 98.36693 is the total time from the 'Risk Analysis'
## 43.23054 is the total time from the 'Infusion Starts'
##
## Chart B is the chart to follow
## Patient transported to Shock Room!
## 73.41506 is the total time from the 'Risk Analysis'
## 21.23338 is the total time from the 'Infusion Starts'
##
## Chart B is the chart to follow
## Patient transported to Shock Room!
## 89.0922 is the total time from the 'Risk Analysis'
## 23.22403 is the total time from the 'Infusion Starts'
##
## Chart A is the chart to follow
## Patient transported to Shock Room!
## 100.0893 is the total time from the 'Risk Analysis'
## 43.90479 is the total time from the 'Infusion Starts'
```

```
##
## Chart B is the chart to follow
## Patient transported to Shock Room!
## 85.01898 is the total time from the 'Risk Analysis'
## 28.77417 is the total time from the 'Infusion Starts'
##
## Chart B is the chart to follow
## Patient transported to Shock Room!
## 77.28692 is the total time from the 'Risk Analysis'
## 17.82524 is the total time from the 'Infusion Starts'
##
## Chart A is the chart to follow
## Patient transported to Shock Room!
## 92.31487 is the total time from the 'Risk Analysis'
## 35.17095 is the total time from the 'Infusion Starts'
##
## Chart B is the chart to follow
## Patient transported to Shock Room!
## 85.25268 is the total time from the 'Risk Analysis'
## 24.59611 is the total time from the 'Infusion Starts'
##
## Chart A is the chart to follow
## Patient transported to Shock Room!
## 68.66099 is the total time from the 'Risk Analysis'
## 25.95892 is the total time from the 'Infusion Starts'
##
## Chart B is the chart to follow
## Patient transported to Shock Room!
## 87.40262 is the total time from the 'Risk Analysis'
## 19.83401 is the total time from the 'Infusion Starts'
##
## Chart A is the chart to follow
## Patient transported to Shock Room!
## 71.45992 is the total time from the 'Risk Analysis'
## 41.15676 is the total time from the 'Infusion Starts'
```

*** PRCBs ***

Filtering different blood products from the data. Trying first with PRBCs (packed red blood cells).

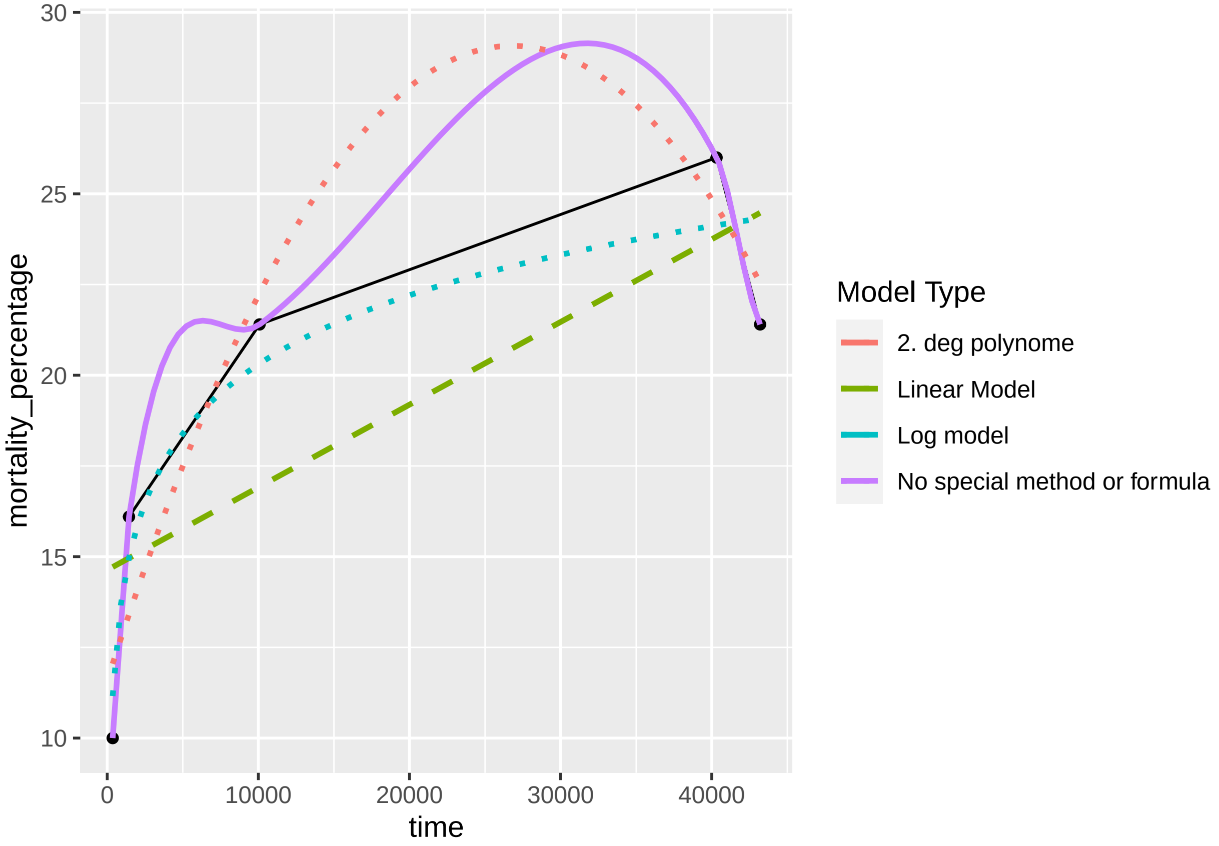
```r
# Filtering rows by products (only some of them are useful at a time)
PRBCS_data <- filter(df, products == "O-negative PRBCs" )
print(PRBCS_data)
```

```
## # A tibble: 5 x 10
##   Article       `Link to source` products  time mortality_perce~ mortality n_tot
##   <chr>         <chr>            <chr>    <dbl>            <dbl>     <dbl> <dbl>
## 1 Civilian pre~ https://onlinel~ O-negat~  1440             16.1     0.161    56
## 2 <NA>          <NA>             O-negat~ 10080             21.4     0.214    56
## 3 <NA>          <NA>             O-negat~ 43200             21.4     0.214    56
## 4 Mortality of~ https://www.ncb~ O-negat~   360             10       0.1      92
## 5 <NA>          <NA>             O-negat~ 40320             26       0.26     78
## # ... with 3 more variables: n_dead <dbl>, n_survived <dbl>, ArticleAbbr <chr>
```

Testing different models to see which one is the best fit.

```
ggplot(PRBCS_data, aes(x = time, y = mortality_percentage) ) +
    geom_point() +
    geom_line() +
    geom_smooth(aes(color="No special method or formula"), se = FALSE,  linetype = 1) +
    geom_smooth(method="lm", aes(color="Linear Model"), formula= (y ~ x), se = FALSE, linetype = 2) +
    geom_smooth(method="lm", aes(color="2. deg polynome"), formula= (y ~ poly(x,2)), se = FALSE, linetyp
    geom_smooth(method = "lm", aes(color="Log model"), formula = y ~ log(x), se = FALSE, linetype = 3)
    guides(color = guide_legend("Model Type"))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 145.8

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 9934.2

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.1112e+09
```



It seems that the logaritmic curve could be the best one, so using it to simulate new data.

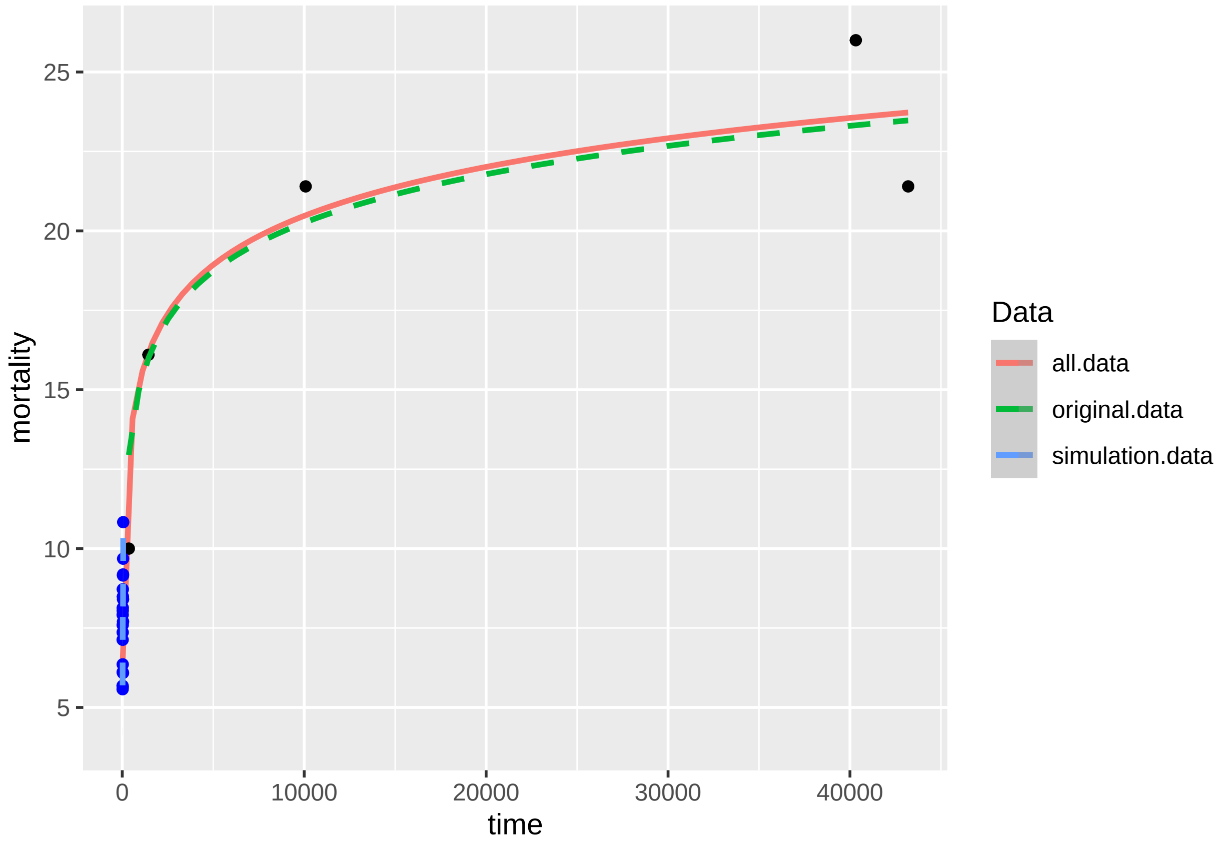
Simulating new data points. Forcing the curve to start from 0 (at that time it is assumed that everyone is still alive.). Adding some noise to the simulated data at the same time.

```
# Predicting new points
my_model <- lm(mortality_percentage~0+log(time), data = PRBCS_data)
variable_time <- data.frame(time=simulated_times)
predicted_y <- predict(object = my_model, newdata = variable_time)

# Adding some noise to predicted_y
for(i in 1:length(predicted_y)){
  predicted_y[i] <- rnorm(1, predicted_y[i],1)
}

# Making combined dataframe from original and simulated data
original.data <- data.frame(time = PRBCS_data$time, mortality = PRBCS_data$mortality_percentage)
simulation.data <- data.frame(time = simulated_times, mortality = predicted_y)
all.data <- rbind(original.data, simulation.data)
```

Plotting the results about PRCBs, curves with raw data, simulated data and with both points combined.

```
# Plotting the combined data and the simulated and original points (O-neg PRBCs)
ggplot(all.data, aes(x = time, y = mortality))+
  geom_smooth(method = "lm", aes(color="all.data"), formula = y~0+log(x), se = FALSE, linetype = 1)+
  geom_point(data = original.data) +
  geom_smooth(data = original.data, method = "lm", aes(color="original.data"), formula = y~0+log(x), se
  geom_point(data = simulation.data, color = "blue")+
  geom_smooth(data = simulation.data, aes(x = simulated_times, y = predicted_y, color = "simulation.data
  guides(color = guide_legend("Data"))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



*** PRBCS AND/OR PLASMA ***

Done as above, changing product.

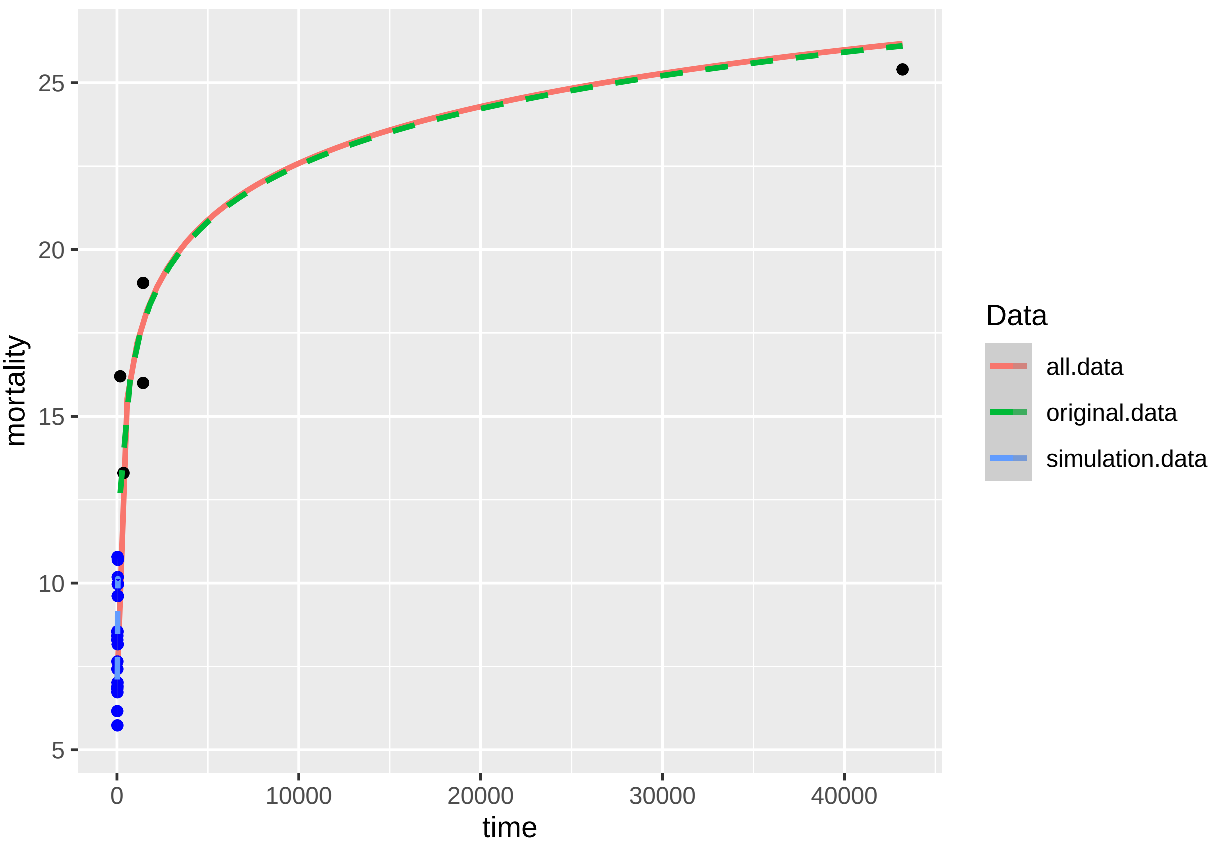
```
# Filtering PRBCS_and_plasma_data:
PRBCS_and_plasma_data <- filter(df, products == "plasma and/or RBCs" )
print(PRBCS_and_plasma_data)
```

```
## # A tibble: 5 x 10
##   Article        `Link to source` products  time mortality_perce~ mortality n_tot
##   <chr>          <chr>            <chr>    <dbl>            <dbl>     <dbl> <dbl>
## 1 Pre-hospital~ https://www.ncb~ plasma ~   360             13.3     0.133    75
## 2 <NA>           <NA>             plasma ~  1440             16       0.16     75
## 3 Multicenter ~ https://www.ncb~ plasma ~   180             16.2     0.162   142
## 4 <NA>           <NA>             plasma ~  1440             19       0.19    142
## 5 <NA>           <NA>             plasma ~ 43200             25.4     0.254   142
## # ... with 3 more variables: n_dead <dbl>, n_survived <dbl>, ArticleAbbr <chr>
```

```
# Making simulated data (using same simulated time points as earlier)
my_model <- lm(mortality_percentage~0+log(time), data = PRBCS_and_plasma_data)
variable_time <- data.frame(time=simulated_times)
predicted_y <- predict(object = my_model, newdata = variable_time)

# Adding some noise to predicted_y
for(i in 1:length(predicted_y)){
  predicted_y[i] <- rnorm(1, predicted_y[i],1)
}
# Making combined data frame from original and simulated data
original.data <- data.frame(time = PRBCS_and_plasma_data$time, mortality = PRBCS_and_plasma_data$mortali
simulation.data <- data.frame(time = simulated_times, mortality = predicted_y)
all.data <- rbind(original.data, simulation.data)
```

```
# Plotting the combined data and the simulated and original points (plasma and/or RCBs)
ggplot(all.data, aes(x = time, y = mortality))+
  geom_smooth(method = "lm", aes(color="all.data"), formula = y~0+log(x), se = FALSE, linetype = 1)+
  geom_point(data = original.data) +
  geom_smooth(data = original.data, method = "lm", aes(color="original.data"), formula = y~0+log(x), se
  geom_point(data = simulation.data, color = "blue")+
  geom_smooth(data = simulation.data, aes(x = simulated_times, y = predicted_y, color = "simulation.data
  guides(color = guide_legend("Data"))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Combination of red blood cells and plasma is the only product that has higher than 25% mortality according to these plots, the simulated time points have also the greatest distribution compared to other products…

*** WHOLE BLOOD *** Done as above, changing product.

Had an issue with negative mortality rates near zero at first, but it was solved by forcing the intercept to 0. However, this made the curve a lot lower, which might be problematic.

```r
# Filtering whole blood:
WB_data <- filter(df, products == "LTOWB" )
print(WB_data)
```

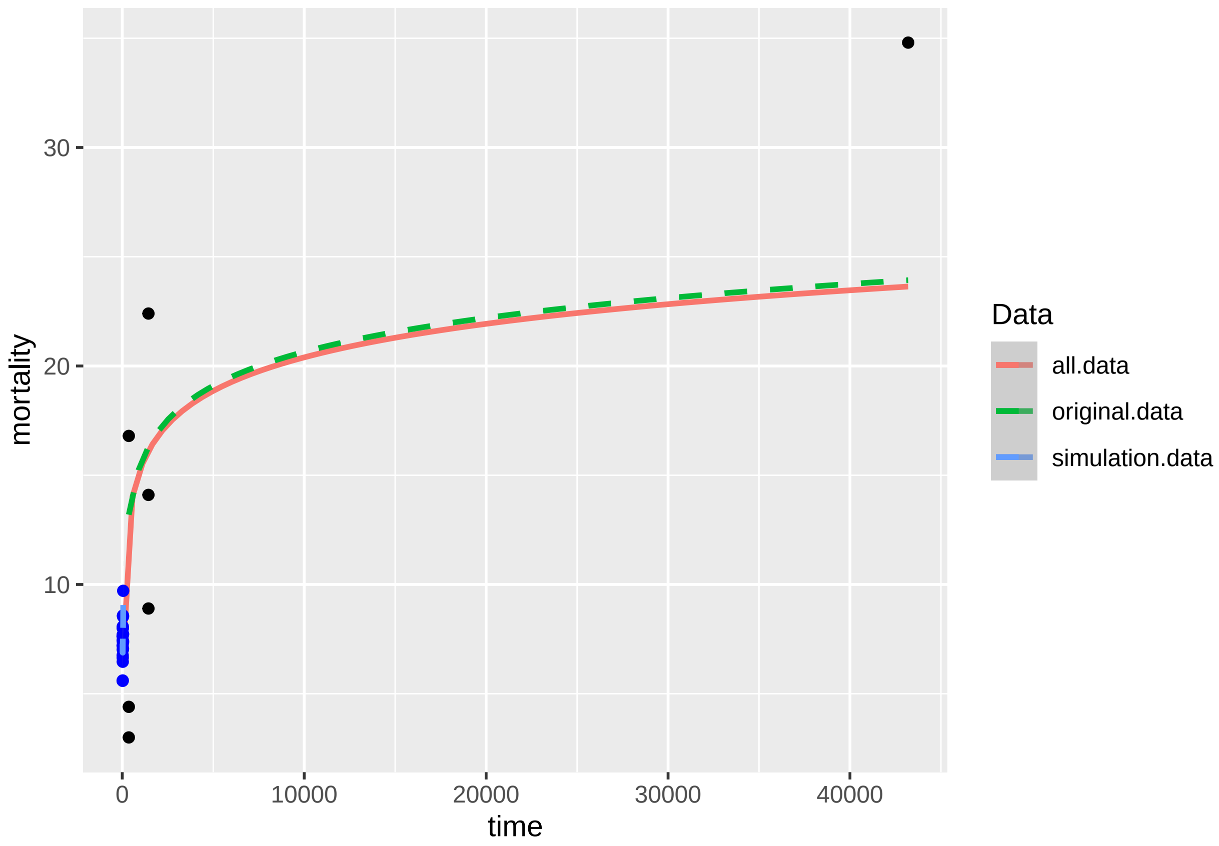
```
## # A tibble: 7 x 10
##   Article        `Link to source` products  time mortality_perce~ mortality n_tot
##   <chr>          <chr>            <chr>    <dbl>            <dbl>     <dbl> <dbl>
## 1 Prehospital ~ https://pubmed.~ LTOWB      360             16.8     0.168   107
## 2 <NA>          <NA>             LTOWB     1440             22.4     0.224   107
## 3 Clinical out~ https://pubmed.~ LTOWB      360              3       0.03    135
## 4 <NA>          <NA>             LTOWB     1440              8.9     0.089   135
## 5 Injured reci~ https://pubmed.~ LTOWB      360              4.4     0.044    92
## 6 <NA>          <NA>             LTOWB     1440             14.1     0.141    92
## 7 <NA>          <NA>             LTOWB    43200             34.8     0.348    92
## # ... with 3 more variables: n_dead <dbl>, n_survived <dbl>, ArticleAbbr <chr>
```

```r
# Making simulated data (using same simulated time points as earlier)
my_model <- lm(mortality_percentage~0+log(time), data = WB_data)
variable_time <- data.frame(time=simulated_times)
predicted_y <- predict(object = my_model, newdata = variable_time)
```

```
# Adding some noise to predicted_y
for(i in 1:length(predicted_y)){
  predicted_y[i] <- rnorm(1, predicted_y[i],1)
}
# Making combined data frame from original and simulated data
original.data <- data.frame(time = WB_data$time, mortality = WB_data$mortality_percentage)
simulation.data <- data.frame(time = simulated_times, mortality = predicted_y)
all.data <- rbind(original.data, simulation.data)
```

```
# Plotting the combined data and the simulated and original points (WB)
ggplot(all.data, aes(x = time, y = mortality))+
  geom_smooth(method = "lm", aes(color="all.data"), formula = y~0+log(x), se = FALSE, linetype = 1)+
  geom_smooth(data = original.data, method = "lm", aes(color="original.data"), formula = y~0+log(x), se
  geom_point(data = original.data) +
  geom_point(data = simulation.data, color = "blue")+
  geom_smooth(data = simulation.data, aes(x = simulated_times, y = predicted_y, color = "simulation.data
  guides(color = guide_legend("Data"))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



This curve doesn't seem as good fit as the others, the mortality is quite low compared to the time points in original data from the excel. Not sure if this could be prevented by deleting time points from the start (other products have less or no time points for example at 3 hours (360 min)).

*** ALL PRODUCTS COMBINED ***

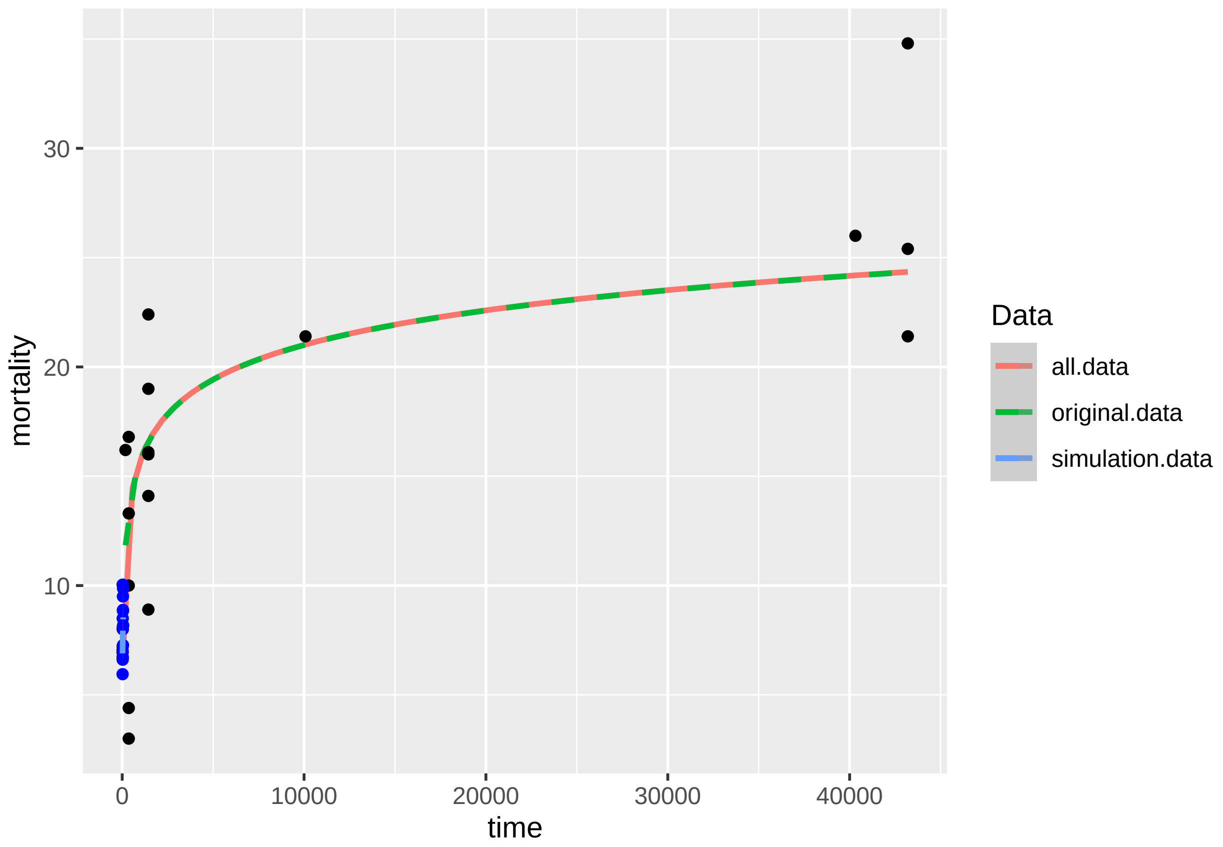
```
all_products <- filter(df, products == "O-negative PRBCs" | products == "LTOWB" | products == "plasma an
print(all_products)
```

```
## # A tibble: 17 x 10
##    Article      `Link to source` products  time mortality_perce~ mortality n_tot
##    <chr>        <chr>            <chr>    <dbl>            <dbl>     <dbl> <dbl>
##  1 Civilian pr~ https://onlinel~ O-negat~  1440             16.1     0.161    56
##  2 <NA>         <NA>             O-negat~ 10080             21.4     0.214    56
##  3 <NA>         <NA>             O-negat~ 43200             21.4     0.214    56
##  4 Pre-hospita~ https://www.ncb~ plasma ~   360             13.3     0.133    75
##  5 <NA>         <NA>             plasma ~  1440             16       0.16     75
##  6 Mortality o~ https://www.ncb~ O-negat~   360             10       0.1      92
##  7 <NA>         <NA>             O-negat~ 40320             26       0.26     78
##  8 Multicenter~ https://www.ncb~ plasma ~   180             16.2     0.162   142
##  9 <NA>         <NA>             plasma ~  1440             19       0.19    142
## 10 <NA>         <NA>             plasma ~ 43200             25.4     0.254   142
## 11 Prehospital~ https://pubmed.~ LTOWB      360             16.8     0.168   107
## 12 <NA>         <NA>             LTOWB     1440             22.4     0.224   107
## 13 Clinical ou~ https://pubmed.~ LTOWB      360              3       0.03    135
## 14 <NA>         <NA>             LTOWB     1440              8.9     0.089   135
## 15 Injured rec~ https://pubmed.~ LTOWB      360              4.4     0.044    92
## 16 <NA>         <NA>             LTOWB     1440             14.1     0.141    92
## 17 <NA>         <NA>             LTOWB    43200             34.8     0.348    92
## # ... with 3 more variables: n_dead <dbl>, n_survived <dbl>, ArticleAbbr <chr>
```

```
# Making simulated data (using same simulated time points as earlier)
my_model <- lm(mortality_percentage~0+log(time), data = all_products)
variable_time <- data.frame(time=simulated_times)
predicted_y <- predict(object = my_model, newdata = variable_time)

# Adding some noise to predicted_y
for(i in 1:length(predicted_y)){
  predicted_y[i] <- rnorm(1, predicted_y[i],1)
}
# Making combined data frame from original and simulated data
original.data <- data.frame(time = all_products$time, mortality = all_products$mortality_percentage)
simulation.data <- data.frame(time = simulated_times, mortality = predicted_y)
all.data <- rbind(original.data, simulation.data)
```

```
# Plotting the combined data and the simulated and original points
ggplot(all.data, aes(x = time, y = mortality))+
  geom_smooth(method = "lm", aes(color="all.data"), formula = y~0+log(x), se = FALSE, linetype = 1)+
  geom_smooth(data = original.data, method = "lm", aes(color="original.data"), formula = y~0+log(x), se
  geom_point(data = original.data) +
  geom_point(data = simulation.data, color = "blue")+
  geom_smooth(data = simulation.data, aes(x = simulated_times, y = predicted_y, color = "simulation.data
  guides(color = guide_legend("Data"))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Well, this seems similar to the others. Since the number of data points is so small and the results between different products are quite close to each other, decided to concentrate on getting the best fit to all data points available using other modelling methods (see for example trying_glm.Rmd).