trying_glm

Milla Juntunen

2022-07-13

*** TRYING TO USE GLM FOR CURVE FITTING ***

This code tries to use the glm() to model the mortality rate after giving prehospital blood transfusions. This code is a work in process, not ready due to some problems described below.

```
library("ggfortify")
```

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

```
## Loading required package: ggplot2
```

```
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
```

```
library("ggformula")
```

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

```
## Loading required package: ggstance
```

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

```
##
## Attaching package: 'ggstance'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

```
## Loading required package: scales
```

```
## Loading required package: ggridges
```

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

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

Reading data from excel. Trying first with data that uses PRBCs (packed red blood cells) for prehospital transfusion. Y axis has the mortality percentage, X-axis the time in minutes.

```
# Reading the data from excel (time in minutes AND PROBABILITIES IN DECIMAL NUMBER)
df <- read_excel("C:\\Projektit\\whole_blood_research\\excel\\Emergencyprocess_PHBT_splines.xlsx", shee
PRBCS_data <- filter(df, products == "O-negative PRBCs" )

print(PRBCS_data)
```

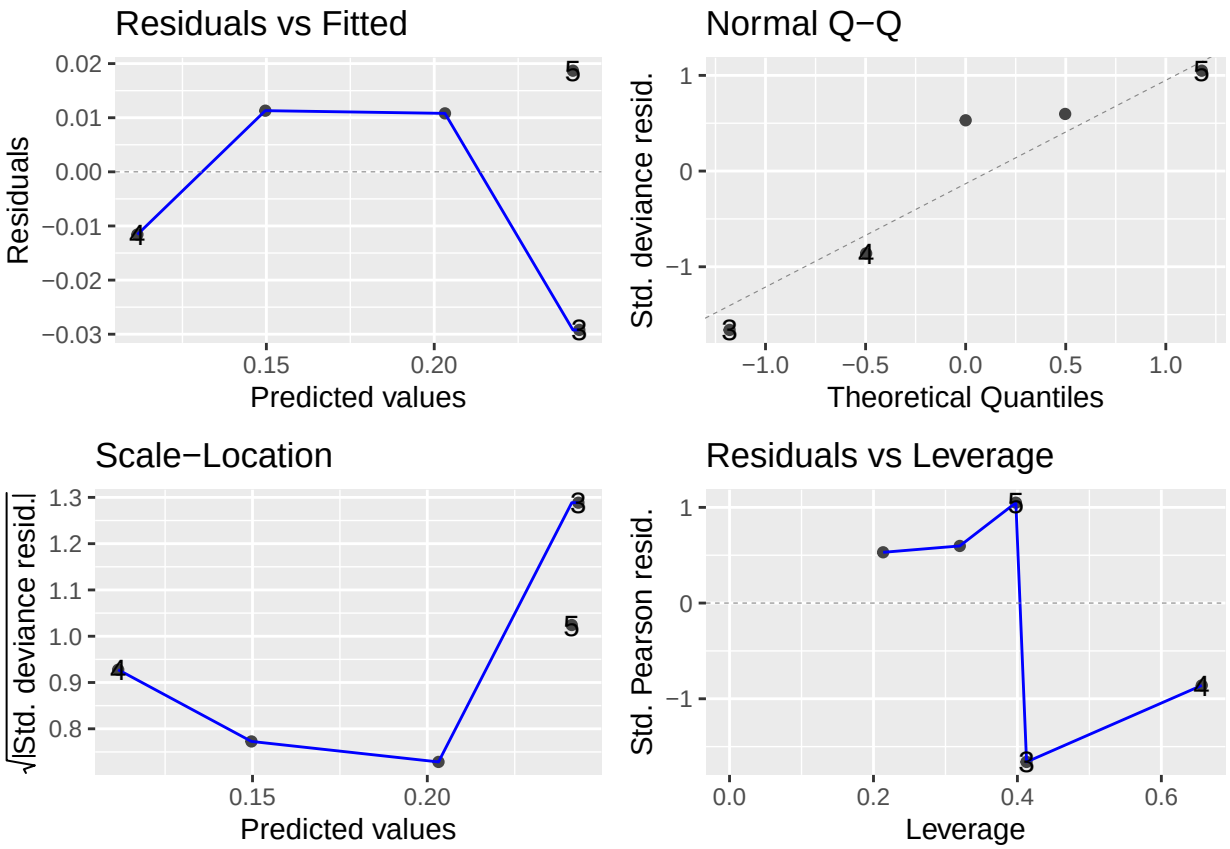
```
## # A tibble: 5 x 10
##   Article       `Link to source` products  time mortality_perce~ mortality n_tot
##   <chr>         <chr>            <chr>    <dbl>            <dbl>     <dbl> <dbl>
## 1 Civilian pre~ https://onlinel~ O-negat~  1440             16.1     0.161    56
## 2 <NA>          <NA>             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. First mortality ~ log(time), without link functions.

```
p <- glm(mortality ~ log(time), data = PRBCS_data)
p
```

```
##
## Call:  glm(formula = mortality ~ log(time), data = PRBCS_data)
##
## Coefficients:
## (Intercept)    log(time)
##    -0.05025      0.02749
##
## Degrees of Freedom: 4 Total (i.e. Null);  3 Residual
## Null Deviance:       0.01499
## Residual Deviance: 0.001581  AIC: -20.11
```

```
autoplot(p) # for mortality ~ log(time)
```

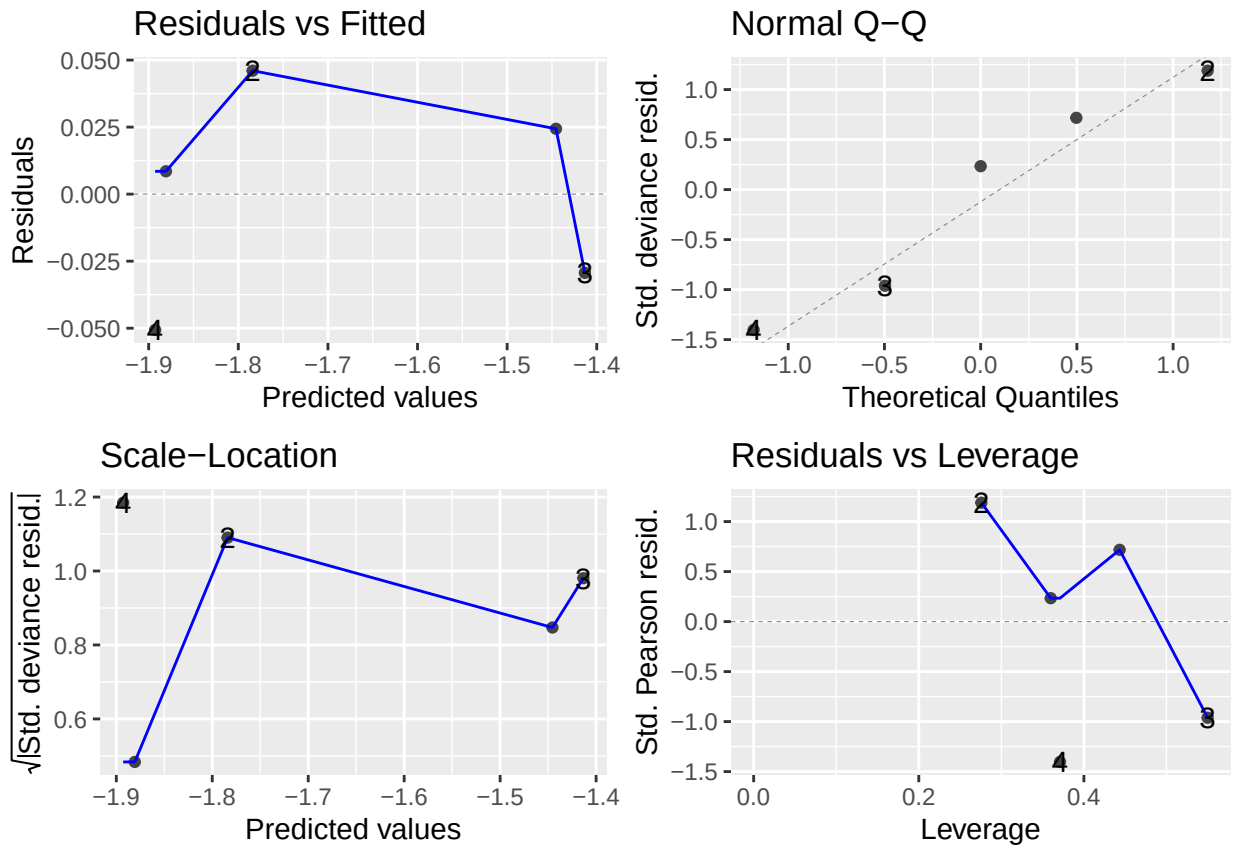


Trying then with mortality ~ time, using gaussian and log link

```
p <- glm(mortality ~ time, data = PRBCS_data, family = gaussian(link="log"))
p
```

```
##
## Call:  glm(formula = mortality ~ time, family = gaussian(link = "log"),
##     data = PRBCS_data)
##
## Coefficients:
## (Intercept)         time
##  -1.897e+00    1.119e-05
##
## Degrees of Freedom: 4 Total (i.e. Null);  3 Residual
## Null Deviance:       0.01499
## Residual Deviance: 0.006212  AIC: -13.26
```

```
autoplot(p) # for gaussian(link = "log")
```

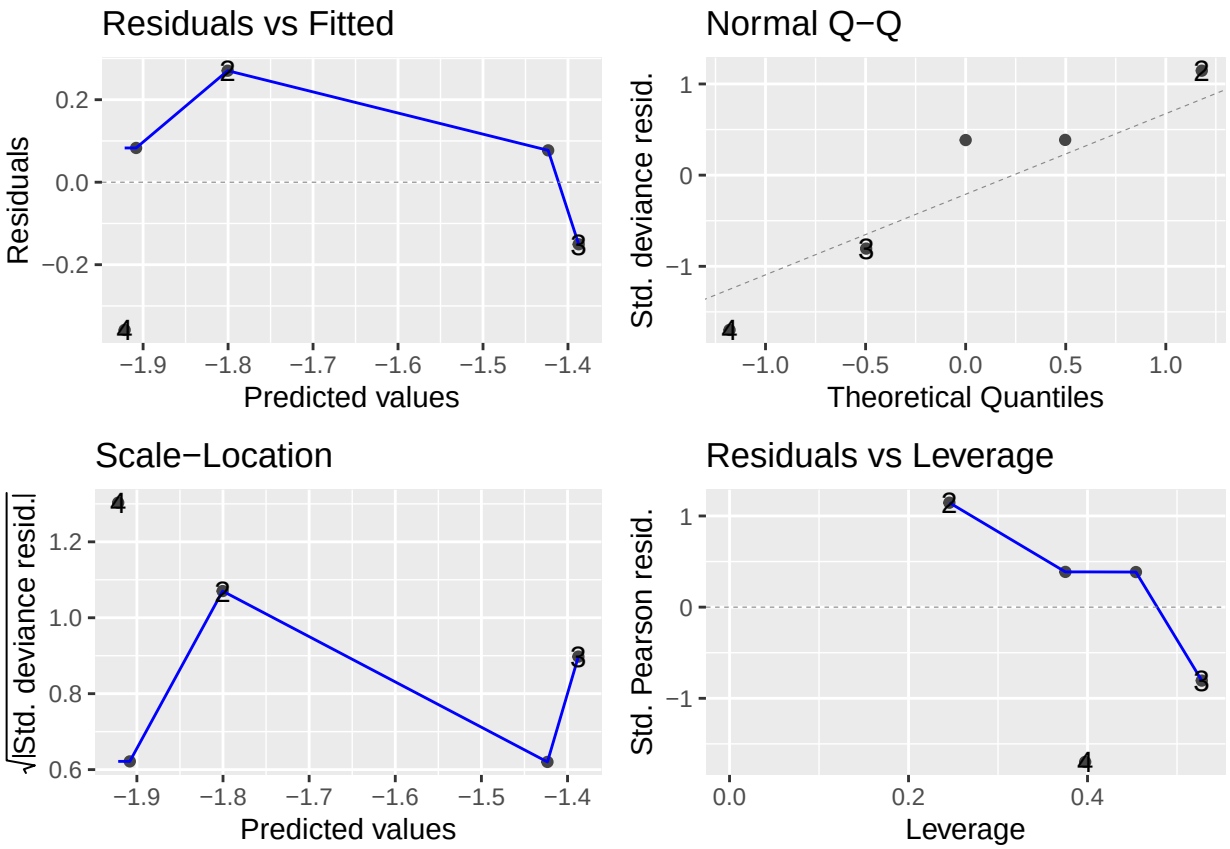


Trying mortality ~ time, usin Gamma with log link

```
p <- glm(mortality ~ time, data = PRBCS_data, family = Gamma(link="log"))
p
```

```
##
## Call:  glm(formula = mortality ~ time, family = Gamma(link = "log"),
##     data = PRBCS_data)
##
## Coefficients:
## (Intercept)         time
##  -1.926e+00    1.247e-05
##
## Degrees of Freedom: 4 Total (i.e. Null);  3 Residual
## Null Deviance:       0.5013
## Residual Deviance: 0.2369    AIC: -12.14
```

```
autoplot(p) # for gamma(link = "log")
```



Gamma has the best AIC so using it to predict new values..? This is very confusing. It is not the smallest AIC but somehow it was still thought to be the best one. Random, should probably discuss about this. This might be caused by the mistakes in excel that are now hopefully all corrected.

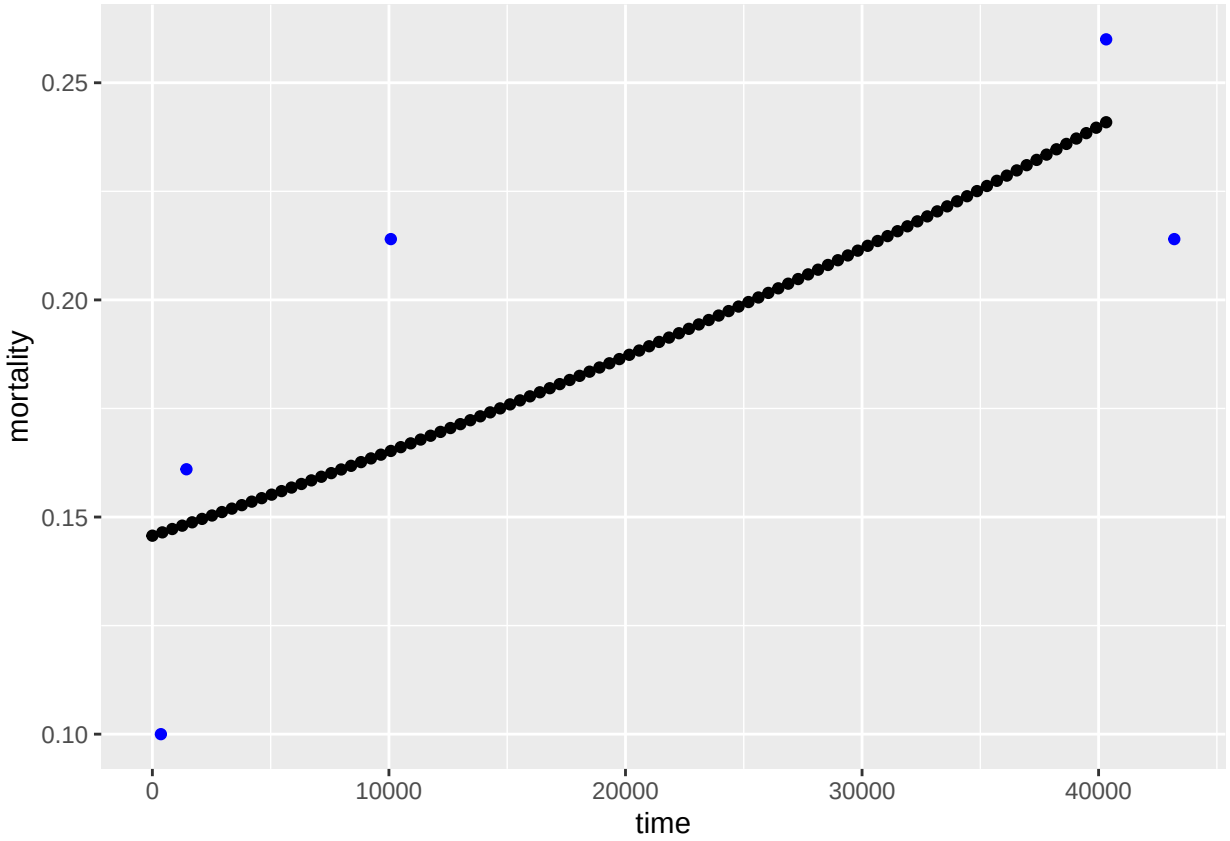
Anyway, trying to simulate new datapoints with the mortality ~ time with gamma and log link, trying to see the curve that this simulation makes (didn't add any bias for that reason)

```
# Plotting the combined data and the simulated and original points

# Trying to use this model
p <- glm(mortality ~ time, data = PRBCS_data, family = Gamma(link = "log"))

# Making timepoints in every 6 hours
variable_time <- data.frame(time=seq(0, 40500, by=420))
tmp <- predict(object = p, newdata = variable_time, type="response")

simulation.data <- data.frame(time = variable_time, mortality = tmp)
ggplot(simulation.data, aes(x = time, y = mortality))+
  geom_point() + geom_point(data = PRBCS_data, color = "blue")
```



That doesn't seem a good fit to original data. The curve should be other way round?

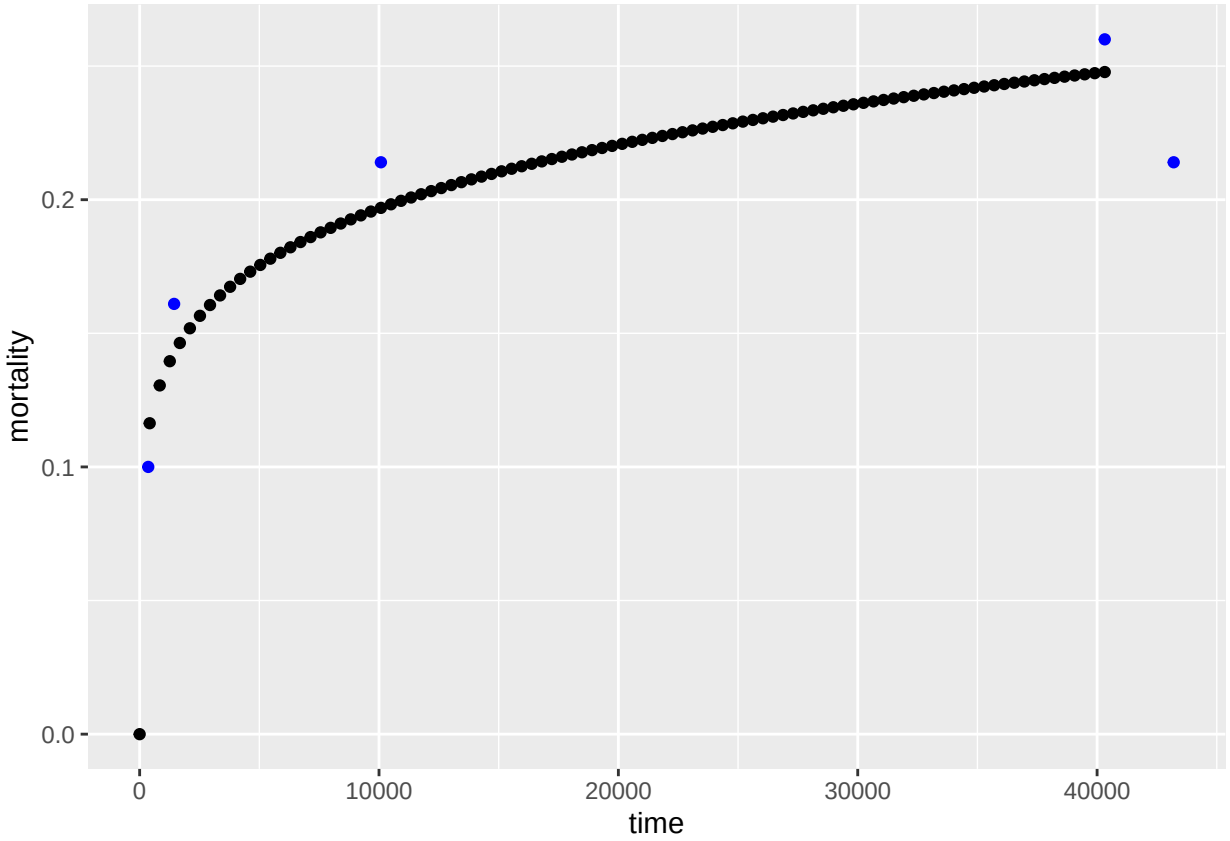
Trying for curiosity to use mortality ~log(time) with gamma and log link.

```
# Plotting the combined data and the simulated and original points again
# This makes a curve (why? or why the first one didn't work???)
p <- glm(mortality ~ log(time), data = PRBCS_data, family = Gamma(link = "log"))
p
```

```
##
## Call:  glm(formula = mortality ~ log(time), family = Gamma(link = "log"),
##     data = PRBCS_data)
##
## Coefficients:
## (Intercept)    log(time)
##     -3.1516       0.1656
##
## Degrees of Freedom: 4 Total (i.e. Null);  3 Residual
## Null Deviance:       0.5013
## Residual Deviance: 0.06353   AIC: -18.75
```

```
# Making timepoints
variable_time <- data.frame(time=seq(0, 40500, by=420))
tmp <- predict(object = p, newdata = variable_time, type="response")

simulation.data <- data.frame(time = variable_time, mortality = tmp)
ggplot(simulation.data, aes(x = time, y = mortality))+
  geom_point() + geom_point(data = PRBCS_data, color = "blue")
```



Seems better, but don't know if it is anywhere near correct to do it in that way. The big question is why the first curve doesn't work as wanted, and it remains unsolved for now.

Trying same with all data (trying to see if this is caused by lack of data).

```
#Plotting all useful data we have
# Reading the data from excel (time in minutes AND PROBABILITIES IN DECIMAL NUMBER)
df <- read_excel("C:\\Projektit\\whole_blood_research\\excel\\Emergencyprocess_PHBT_splines.xlsx", shee
all_data <- filter(df, products == "O-negative PRBCs" | products == "LTOWB" | products == "plasma and/o
print(all_data)
```

```
## # 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>
```

```
# ggplot doesn't understand the gamma link log so not drawing the code below
#ggplot(data = all_data, aes(x = time, y = mortality))+
  #geom_point()+
  #geom_smooth(method = "glm", formula = y~(x), se = FALSE, family = Gamma(link="log"))
```

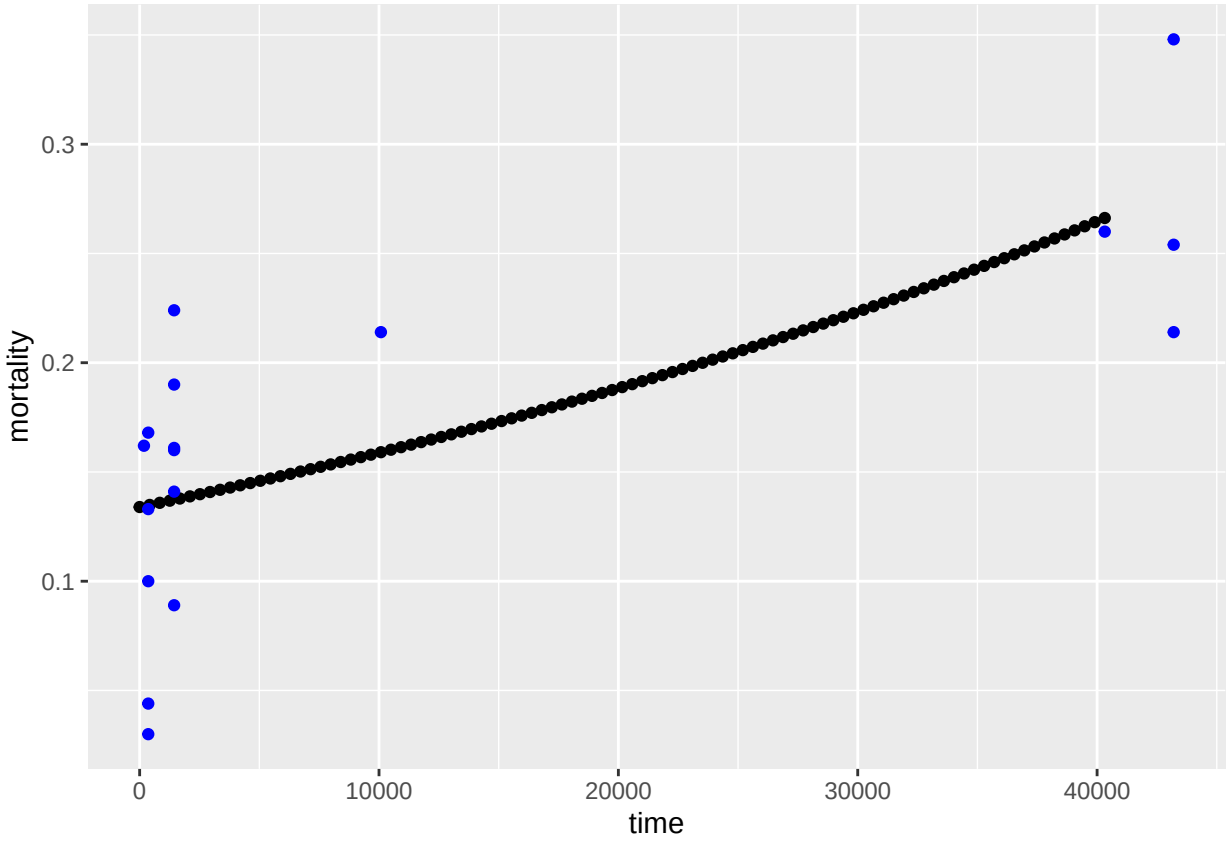
Then plotting with all data in excel:

```
p <- glm(mortality ~ time, data = all_data, family = Gamma(link = "log"))
p
```

```
##
## Call:  glm(formula = mortality ~ time, family = Gamma(link = "log"),
##     data = all_data)
##
## Coefficients:
## (Intercept)         time
##  -2.010e+00    1.704e-05
##
## Degrees of Freedom: 16 Total (i.e. Null);  15 Residual
## Null Deviance:       4.986
## Residual Deviance: 3.354     AIC: -38
```

```
# Making timepoints
variable_time <- data.frame(time=seq(0, 40500, by=420))
tmp <- predict(object = p, newdata = variable_time, type="response")

simulation.data <- data.frame(time = variable_time, mortality = tmp)
ggplot(simulation.data, aes(x = time, y = mortality))+
  geom_point() + geom_point(data = all_data, color = "blue")
```



Didn't solve the problem. Checking also if adding interaction with articles would change anything.

```
p <- glm(mortality ~ time*ArticleAbbr, data = all_data, family = Gamma(link = "log"))
p
```

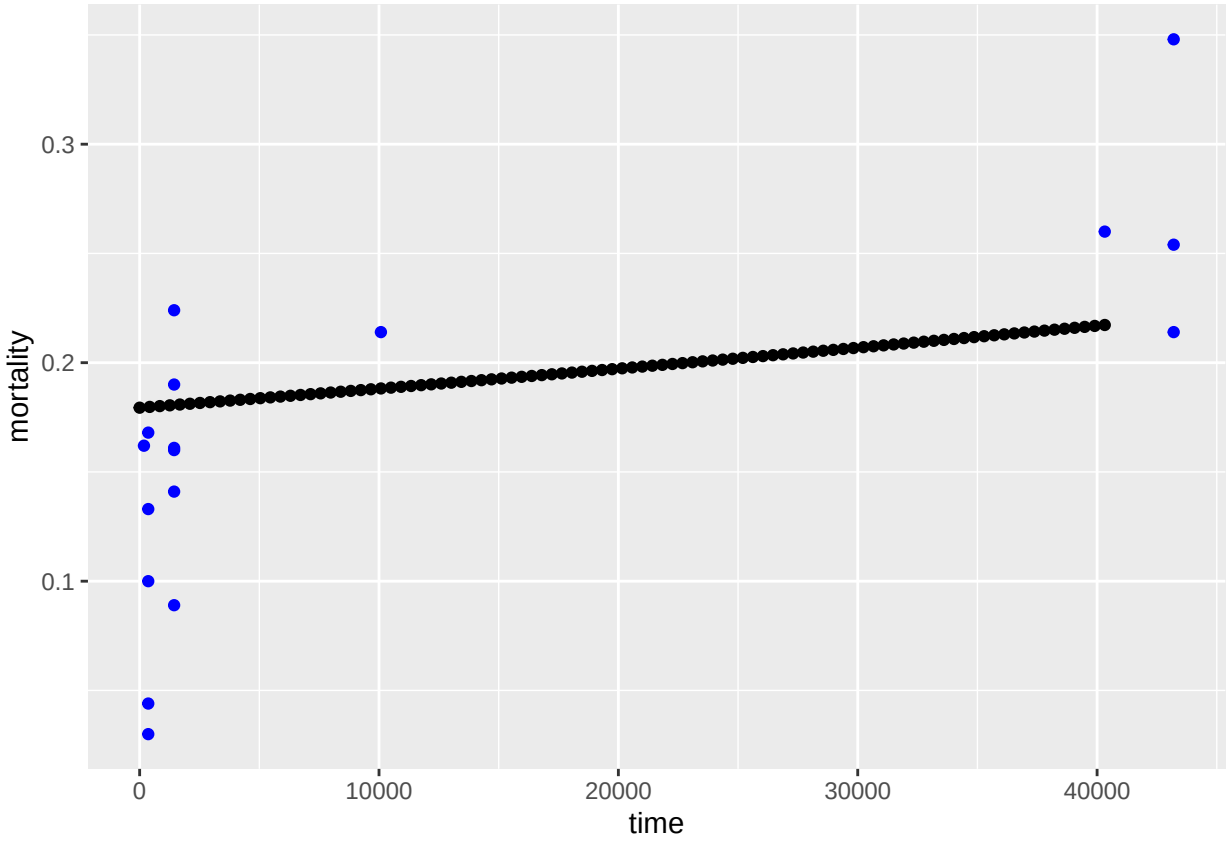
```
##
## Call:  glm(formula = mortality ~ time * ArticleAbbr, family = Gamma(link = "log"),
##     data = all_data)
##
## Coefficients:
##                     (Intercept)                             time
##                       -1.880e+00                        2.664e-04
##      ArticleAbbrCassignol 2020          ArticleAbbrGriggs 2018
##                        1.616e-01                       -4.315e-01
##      ArticleAbbrHenriksen 2016         ArticleAbbrHolcomb 2018
##                       -1.993e-01                        1.338e-01
##        ArticleAbbrSeheult 2018           ArticleAbbrYazer 2021
##                       -1.989e+00                       -5.452e-01
## time:ArticleAbbrCassignol 2020     time:ArticleAbbrGriggs 2018
##                       -2.616e-04                       -2.425e-04
## time:ArticleAbbrHenriksen 2016    time:ArticleAbbrHolcomb 2018
##                       -9.524e-05                       -2.576e-04
##   time:ArticleAbbrSeheult 2018      time:ArticleAbbrYazer 2021
##                        7.405e-04                       -2.344e-04
##
## Degrees of Freedom: 16 Total (i.e. Null);  3 Residual
## Null Deviance:       4.986
```

```
## Residual Deviance: 0.6492    AIC: -42.37
```

AIC is getting smaller, which is a nice thing. Trying to plot with this interaction:

```
# Making timepoints
variable_time <- data.frame(time=seq(0, 40500, by=420), ArticleAbbr=rep("Cassignol 2020",length(tmp)))
tmp <- predict(object = p, newdata = variable_time, type="response")

simulation.data <- data.frame(time = variable_time$time, mortality = tmp)
ggplot(simulation.data, aes(x = time, y = mortality))+
  geom_point() + geom_point(data = all_data, color = "blue")
```



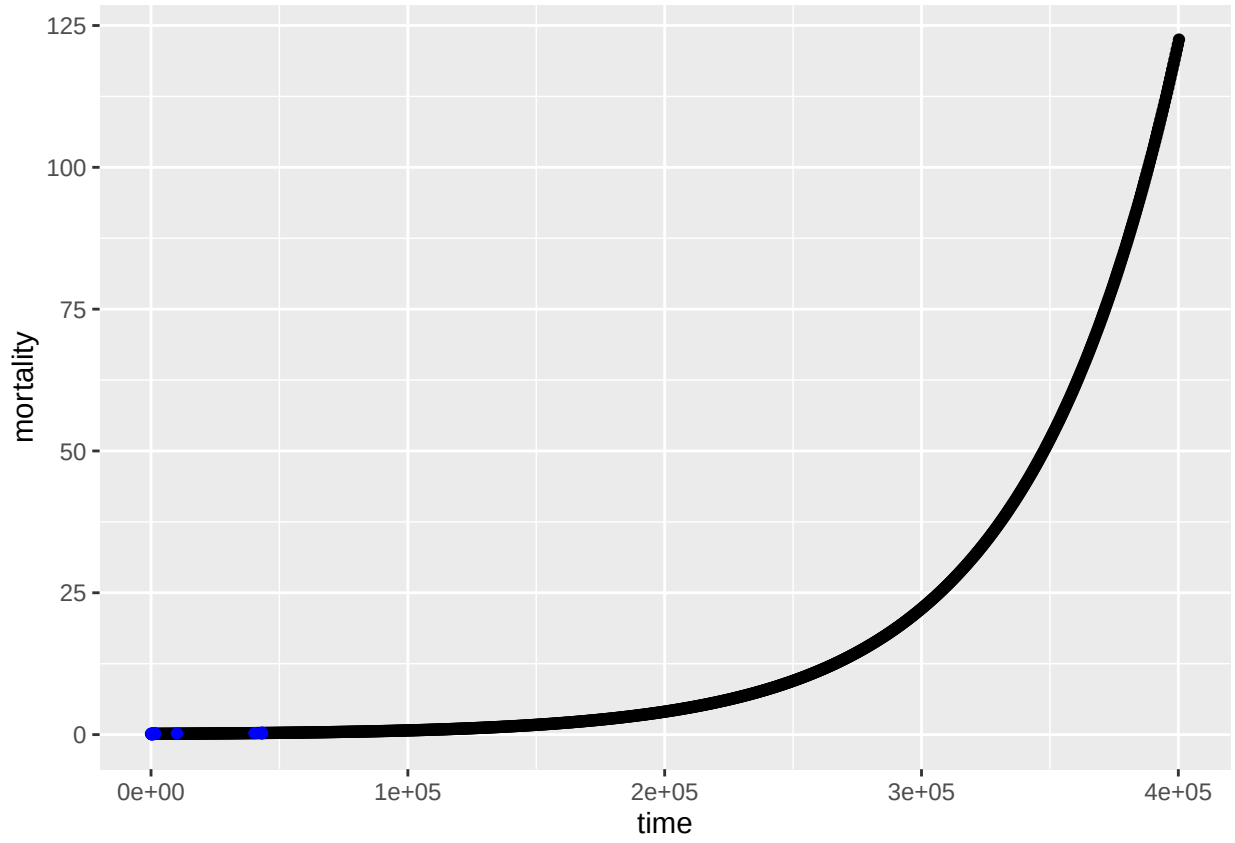
The curve just gets slighter. This is clearly not working. Simulating more points to see what happens to the curve when the time goes by.

```
p <- glm(mortality ~ time, data = all_data, family = Gamma(link = "log"))
p
```

```
##
## Call:  glm(formula = mortality ~ time, family = Gamma(link = "log"),
##     data = all_data)
##
## Coefficients:
## (Intercept)         time
##  -2.010e+00    1.704e-05
##
## Degrees of Freedom: 16 Total (i.e. Null);  15 Residual
## Null Deviance:       4.986
## Residual Deviance: 3.354     AIC: -38
```

```r
# Making timepoints
variable_time <- data.frame(time=seq(0, 400500, by=420))
tmp <- predict(object = p, newdata = variable_time, type="response")

simulation.data <- data.frame(time = variable_time, mortality = tmp)
ggplot(simulation.data, aes(x = time, y = mortality))+
  geom_point() + geom_point(data = all_data, color = "blue")
```



Well it is clearly wrong way round. This problem remains unsolved and has to be rethought in the fall. Also checked how the suspicious model y ~ log(x) works in the long run:

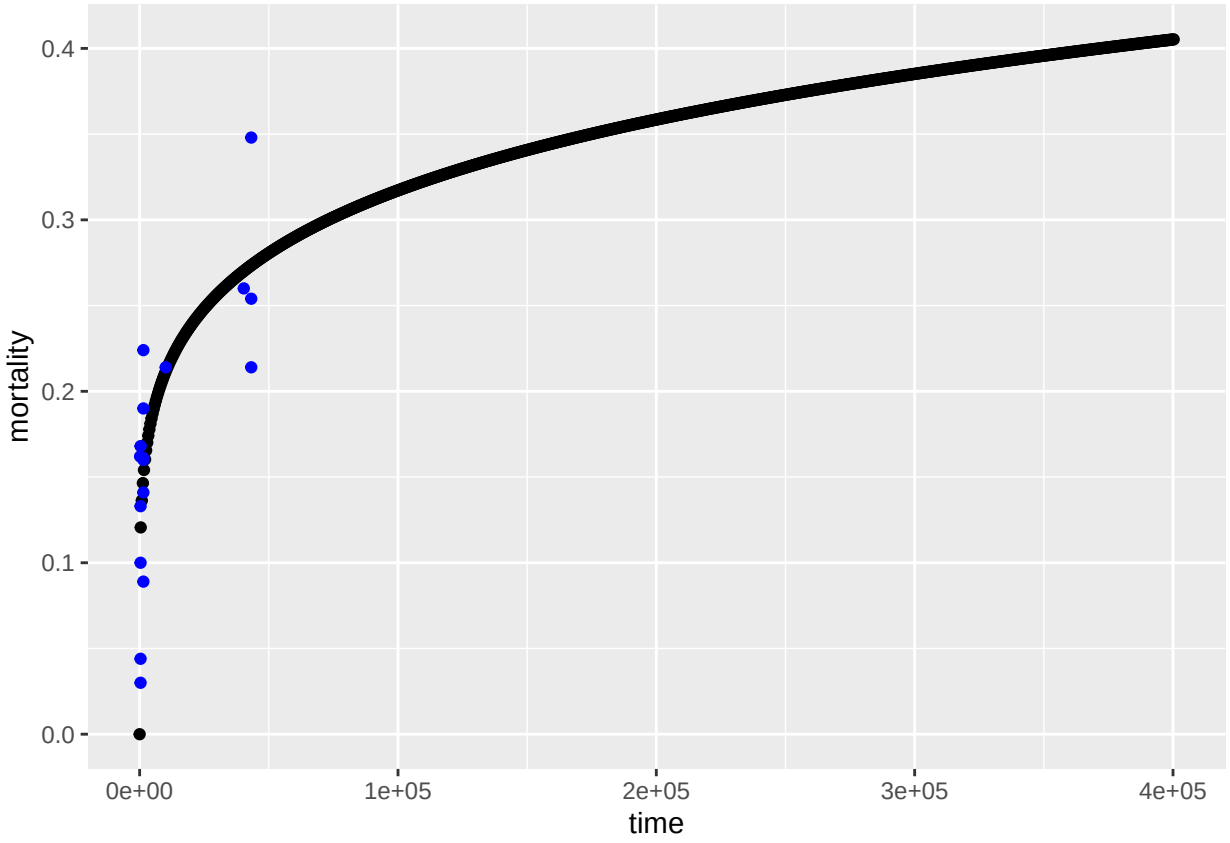
```r
p <- glm(mortality ~ log(time), data = all_data, family = Gamma(link = "log"))
p
```

```
##
## Call:  glm(formula = mortality ~ log(time), family = Gamma(link = "log"),
##     data = all_data)
##
## Coefficients:
## (Intercept)    log(time)
##     -3.1822       0.1767
##
## Degrees of Freedom: 16 Total (i.e. Null);  15 Residual
## Null Deviance:       4.986
## Residual Deviance: 2.983     AIC: -40.05
```

```r
# Making timepoints
variable_time <- data.frame(time=seq(0, 400500, by=420))
```

```
tmp <- predict(object = p, newdata = variable_time, type="response")

simulation.data <- data.frame(time = variable_time, mortality = tmp)
ggplot(simulation.data, aes(x = time, y = mortality))+
  geom_point() + geom_point(data = all_data, color = "blue")
```



Seems more reasonable than the other one but as said before, I don't know if this is okay to do like this or how this is interpreted...