

# Stat 571 HW2

Dongyang Wang

2023-01-28

```
rm(list=ls())
set.seed(42)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr  1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

## Question 1

## Question 2

d

The results from Newton Raphson can be very similar to the results obtained by LMM.

```
library(lme4)

## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following objects are masked from 'package:tidyr':
##
##      expand, pack, unpack

library(mvtnorm)

m = 200
n=3
theta = 1
sigma = 1
beta = c(1, 0.5)

b = rnorm(m, mean = 0, sd = sqrt(theta))
e = rnorm(m*n, mean = 0, sd = sqrt(sigma))

X = cbind(1, rep(1:n, m))
Y = X %*% beta + rep(b, each = n) + e

df = data.frame(X = X[,2], Y = Y, id = rep(1:m, each = n))

reml_model = lmer(Y ~ X + (1|id), data = df, REML = TRUE)
ml_model = lmer(Y ~ X + (1|id), data = df, REML = FALSE)

summary(reml_model)

## Linear mixed model fit by REML ['lmerMod']
## Formula: Y ~ X + (1 | id)
## Data: df
##
## REML criterion at convergence: 1959.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.20342 -0.61742  0.00392  0.59559  2.57093
##
## Random effects:
## Groups Name Variance Std.Dev.
## id (Intercept) 0.9564 0.9780
## Residual 0.9591 0.9793
## Number of obs: 600, groups: id, 200
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 1.00362 0.12638 7.941
## X 0.45537 0.04897 9.300
##
```

```
## Correlation of Fixed Effects:
##   (Intr)
## X -0.775

summary(ml_model)

## Linear mixed model fit by maximum likelihood ['lmerMod']
## Formula: Y ~ X + (1 | id)
##   Data: df
##
##      AIC      BIC   logLik deviance df.resid
## 1960.5   1978.1   -976.3   1952.5     596
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.20668 -0.61787  0.00409  0.59664  2.57496
##
## Random effects:
##   Groups   Name                Variance Std.Dev.
##   id       (Intercept)  0.9509   0.9751
##   Residual                  0.9567   0.9781
## Number of obs: 600, groups: id, 200
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  1.00362    0.12616   7.955
## X            0.45537    0.04891   9.311
##
## Correlation of Fixed Effects:
##   (Intr)
## X -0.775
```

### Question 3

a

The following is a chat with chatGPT.

Question: Should I use REML or maximum likelihood?

Answer: It depends on the specifics of your research question and the data you are working with.

In general, maximum likelihood (ML) is a commonly used method for estimating the parameters of a statistical model. It involves finding the values of the parameters that maximize the likelihood of observing the data.

However, in some cases, such as when you have a multi-level or mixed effects model, using restricted maximum likelihood (REML) instead of ML can produce more efficient and less biased estimates. REML adjusts the likelihood function to account for the dependence structure in the data, and it is particularly useful in situations where there is non-independence in the errors or residuals.

Ultimately, the choice between ML and REML will depend on the nature of the data and the goals of the analysis. Consulting with a statistician or expert in the field can help you determine the best approach for your study.

b

I will conduct some simulations for the response chatGPT gave me. The key point is that using restricted maximum likelihood (REML) instead of ML can produce more efficient and less biased estimates ... where

there is non-independence in the errors or residuals. I will basically conduct a simulation to compare the performance of ML and REML with LMM, in terms of efficiency (as indicated by beta variances) and bias (how much it deviates from the true betas).

The data generating process is similar to HW1. I recycled some of the code. To reiterate the logic: Under the random intercept model, since  $\text{var}(Y) = 1 = \theta + \sigma^2$  and  $\text{corr}(Y_{ij}, Y_{ik}) = \rho = \frac{\theta}{\theta + \sigma^2}$ , we solve the equations and get  $\theta = \rho$  and  $\sigma^2 = 1 - \rho$ . In this way, we can generate the x, e, b separately and use a linear relationship we choose to generate the y values, without the need to sample y directly but achieving the same results.

```
library(lme4)

# Set beta to 0.5 and 1
beta1 = 0.5
beta0 = 1
p = 0.5

params <- expand.grid(
  m = c(5,10,20, 50,100), # individuals
  n = c(5,10,20) # observations per individual
)

# For testing
#m = 10
#n = 5

gen.one <- function(m,n){

  total = m*n

  # Generate the variables
  x = rnorm(total, 0, 1)
  b = rep(rnorm(m, mean = 0, sd = sqrt(p)),n)
  e = rnorm(total, mean = 0, sd = sqrt(1-p))
  y = beta0 + beta1*x + b + e

  # LMM
  lmm1 = lmer(y ~ x + (1|b), REML = F)
  lmm2 = lmer(y ~ x + (1|b), REML = T)

  # Estimate variance for efficiency
  lmm1_var0 = vcov(lmm1)[1,1]
  lmm1_var1 = vcov(lmm1)[2,2]
  lmm2_var0 = vcov(lmm2)[1,1]
  lmm2_var1 = vcov(lmm2)[2,2]

  # Estimate coefficients
  lmm1_coef0 = fixef(lmm1)[1]
  lmm1_coef1 = fixef(lmm1)[2]
  lmm2_coef0 = fixef(lmm2)[1]
  lmm2_coef1 = fixef(lmm2)[2]

  # Estimate bias
  lmm1_bias0 = lmm1_coef0 - beta0
  lmm1_bias1 = lmm1_coef1 - beta1
  lmm2_bias0 = lmm2_coef0 - beta0
```

```

lmm2_bias1 = lmm2_coef1 - beta1

return(data.frame(m = m, n = n,
                  ML_var0 = lmm1_var0, ML_var1 = lmm1_var1,
                  REML_var0 = lmm2_var0, REML_var1 = lmm2_var1,
                  ML_bias0 = lmm1_bias0, ML_bias1 = lmm1_bias1,
                  REML_bias0 = lmm2_bias0, REML_bias1 = lmm2_bias1
                  ) )
}

```

Now we do a simulation for 1000 observations each.

```

nrep = 1000

simulation <- do.call(rbind, lapply(c(1:nrow(params)), function(i){
  m <- params$m[i]
  n <- params$n[i]
  res <- do.call(rbind, lapply(c(1:nrep), function(nrep){
    gen.one(m,n)
  })))
  mean_res <- colMeans(res)
}))

simulation_cor = as.data.frame(simulation)

```

Now, we want to visualize the results.

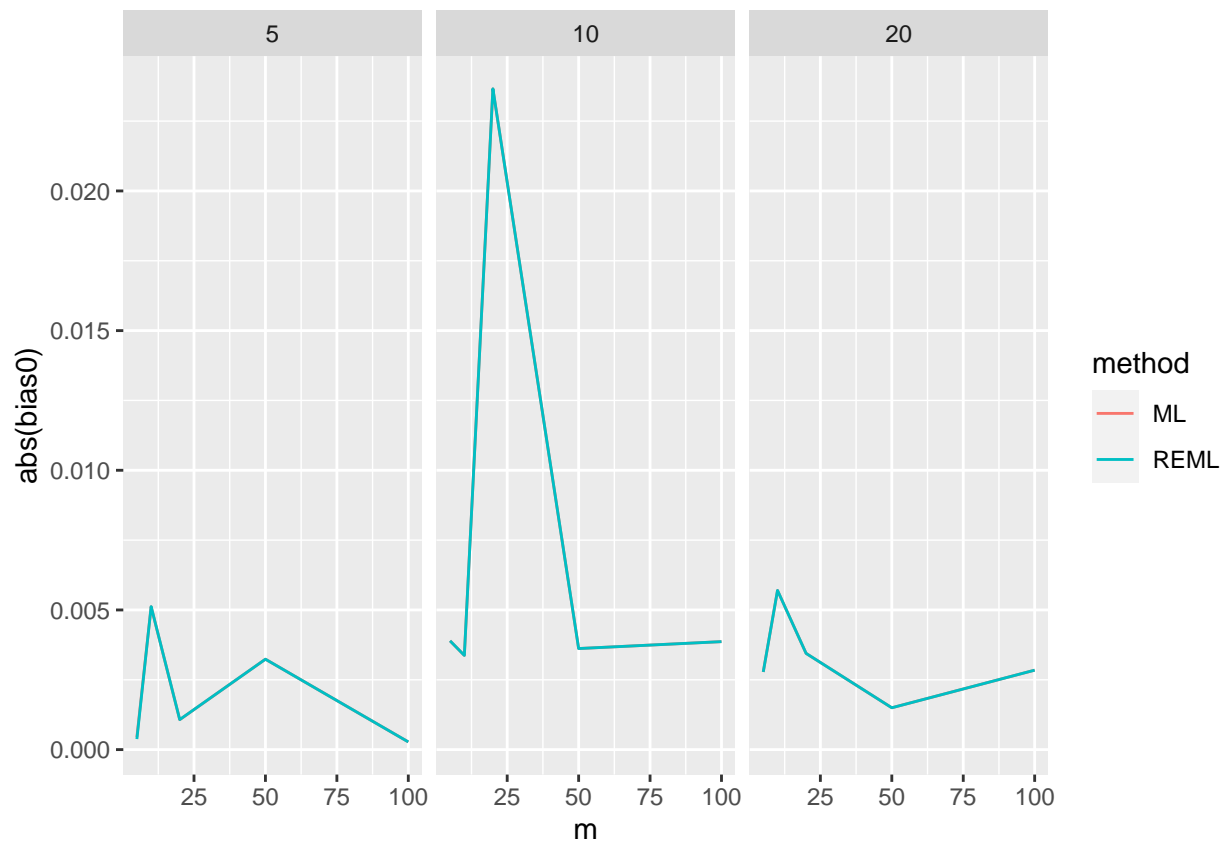
```

simulation_cor_reshaped = reshape(simulation_cor, direction="long",
  varying=list(c("ML_var0", "REML_var0"), c("ML_var1", "REML_var1"),
    c("ML_bias0", "REML_bias0"), c("ML_bias1", "REML_bias1")),
  v.names=c("var0", "var1", "bias0", "bias1"))

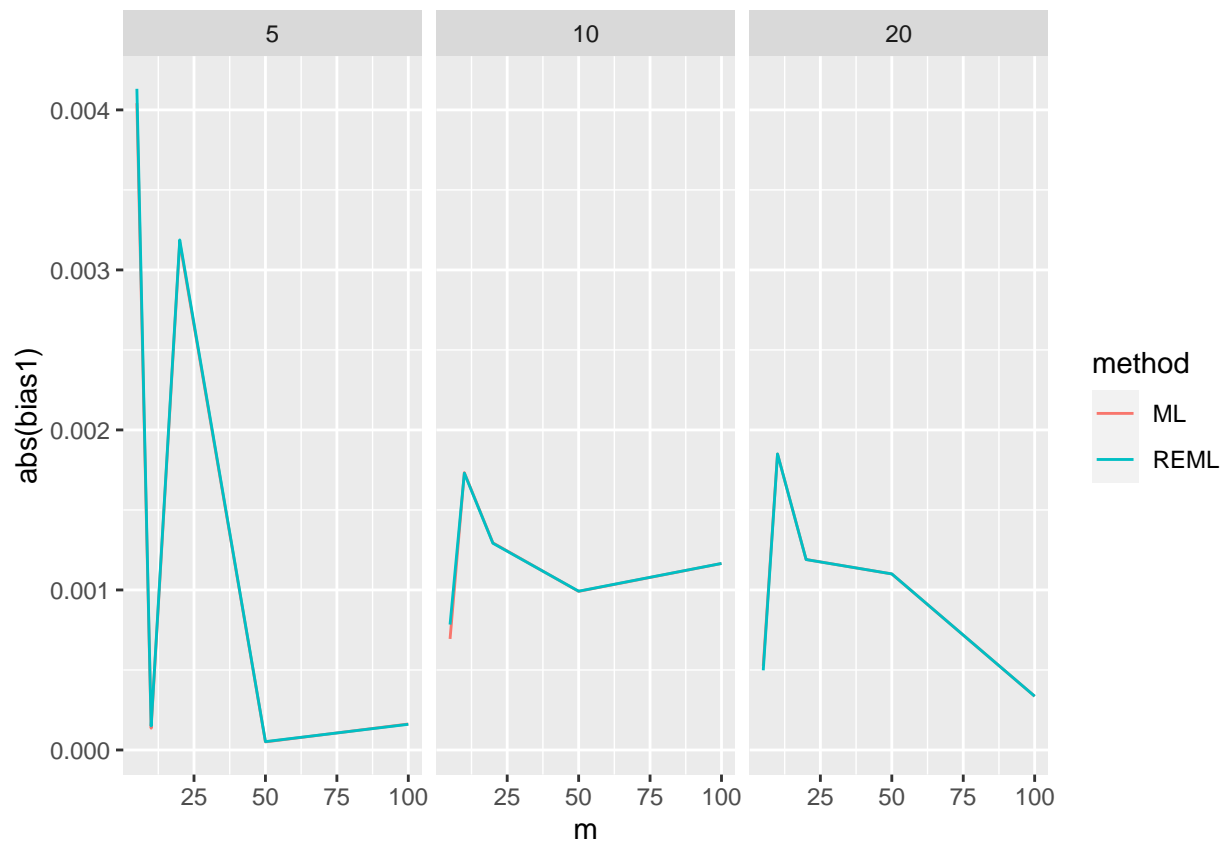
simulation_cor_reshaped$method = case_when(
  simulation_cor_reshaped$time == 1 ~ "ML",
  simulation_cor_reshaped$time == 2 ~ "REML")

ggplot(data=simulation_cor_reshaped, aes(x=m, y=abs(bias0), color = method))+geom_line()+
  facet_grid(cols=vars(n))

```

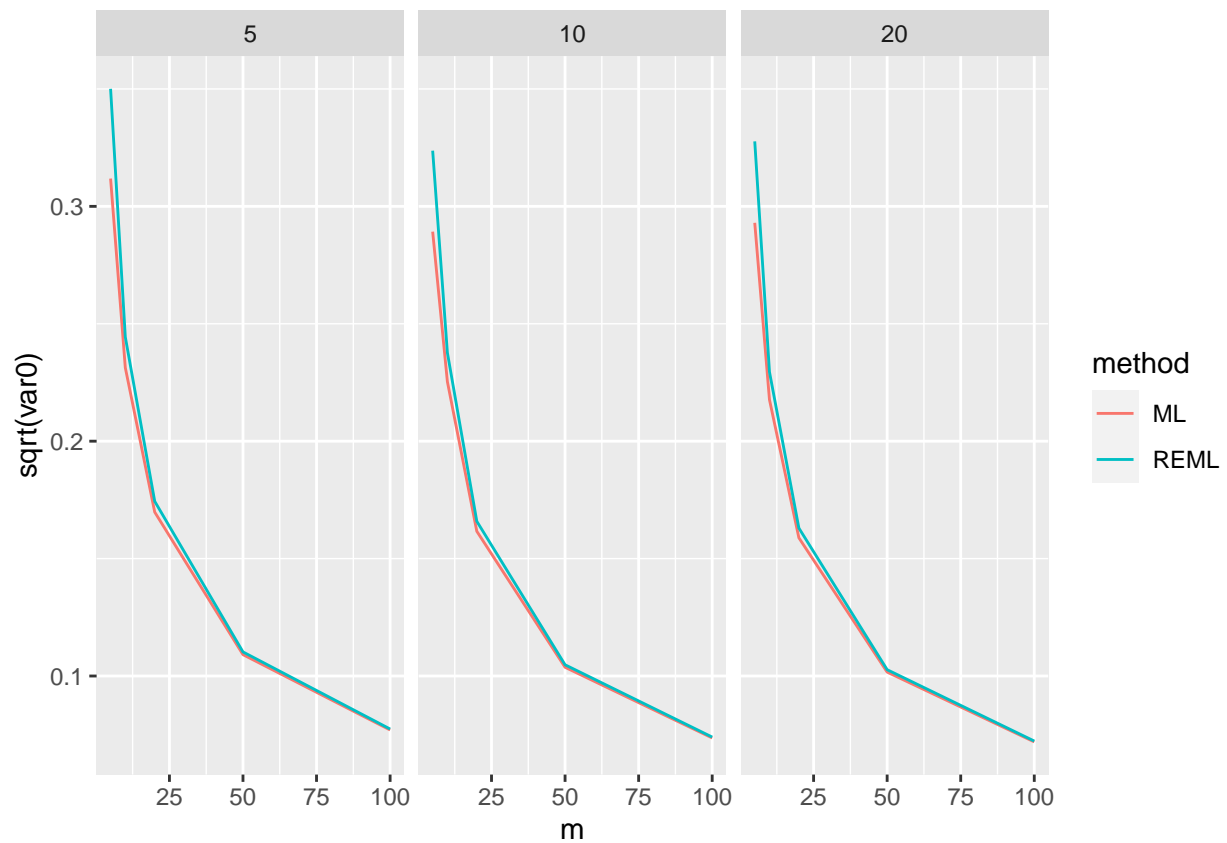


```
ggplot(data=simulation_cor_resaped, aes(x=m, y=abs(bias1), color = method))+geom_line()+
  facet_grid(cols=vars(n))
```

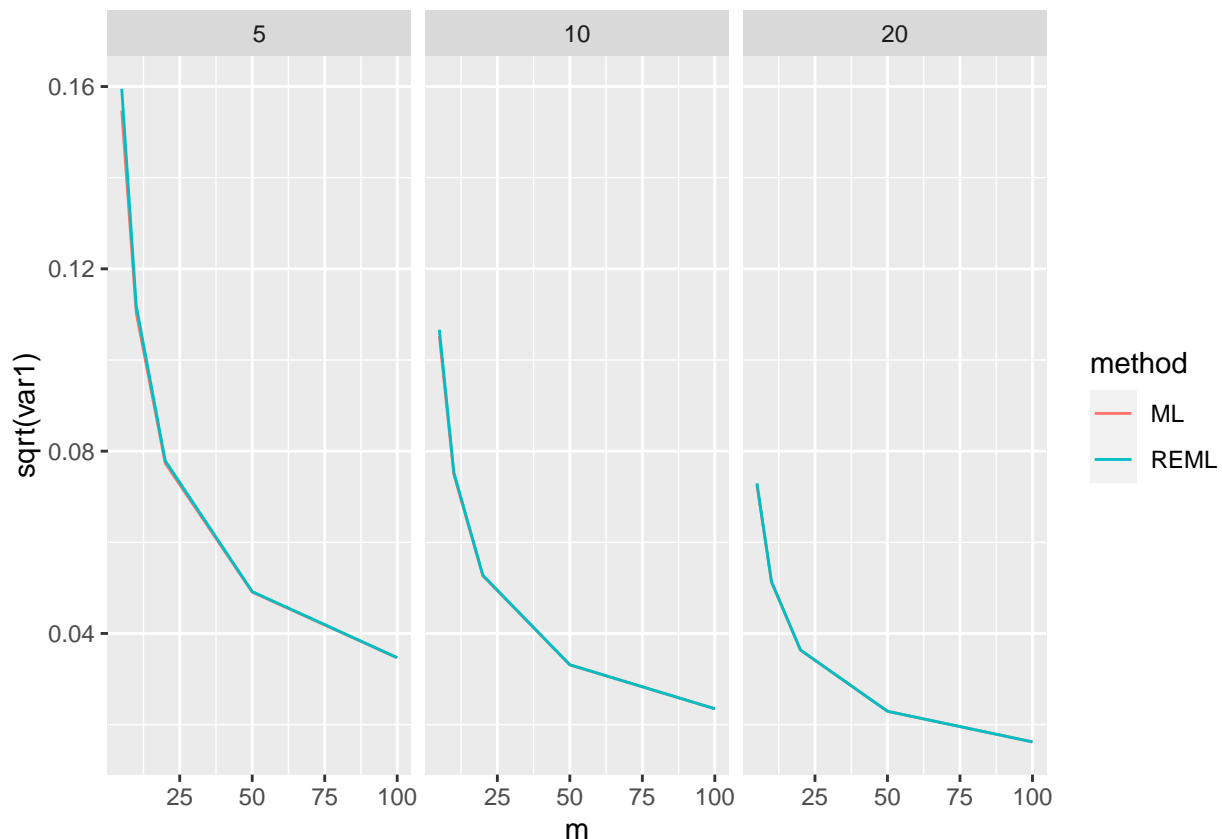


```
ggplot(data=simulation_cor_resaped, aes(x=m, y=sqrt(var0), color = method))+geom_line()+
  facet_grid(cols=vars(n))
```





```
ggplot(data=simulation_cor_resaped, aes(x=m, y=sqrt(var1), color = method))+geom_line()+
  facet_grid(cols=vars(n))
```



From the visualizations, we know that the ML and REML provide very similar results, with a few differences. For the variance on beta 0, the ML provides a better(lower) result. The bias for beta1 under the REML method is smaller when the sample size is small.

The differences are really small, compared with the results from HW1, it appears that as long as we fit the correct model (correlated model) for correlated data, the choice of REML is not going to make a huge difference. Also, it seems that increasing the sample size or observations in each cohort would also provide better efficiency and unbiasedness. However, it's true that REML can still provide some advantages such as less bias, especially when the sample size is small.

Based on the visualizations, we can observe empirically that chatGPT not necessarily provides the correct result. Since it says that REML will give more efficient and less biased estimates. It's either the case that we might not obtain more efficient results, or the efficiency is on something else other than betas. Either way, chatGPT gives some good information but not entirely accurate/correct.

The results are, however, consistent with the lecture notes. As in slides 2.1, on page 6-7 the REML estimator has a smaller bias and a larger variance compared to its MLE counterpart. This makes sense because to account for unbiasedness, by bias and variance tradeoff, we might increase the variance.

## Question 4

a

```
df = read.table("framingham.dat", header=F)
colnames(df) = c("age", "gender", "BMI_base", "BMI_10yrs", "cigarette_base", "cholst_base",
                 "cholst_2", "cholst_4", "cholst_6", "cholst_8", "cholst_10", "dead")
df$id <- seq.int(nrow(df))
df[df == -9] = NA
summary(df)
```

```
##      age      gender      BMI_base      BMI_10yrs
## Min.   :29.00   Min.   :1.000   Min.   :15.00   Min.   :15.00
## 1st Qu.:36.00   1st Qu.:1.000   1st Qu.:22.00   1st Qu.:23.00
## Median :42.00   Median :2.000   Median :25.00   Median :25.00
## Mean   :42.96   Mean   :1.552   Mean   :25.04   Mean   :25.45
## 3rd Qu.:50.00   3rd Qu.:2.000   3rd Qu.:27.00   3rd Qu.:28.00
## Max.   :62.00   Max.   :2.000   Max.   :56.00   Max.   :56.00
##                                     NA's   :4      NA's   :1
## cigarette_base cholst_base cholst_2 cholst_4
## Min.   : 0.000   Min.   :117.0   Min.   :115.0   Min.   :113.0
## 1st Qu.: 0.000   1st Qu.:188.0   1st Qu.:195.0   1st Qu.:200.0
## Median : 5.000   Median :217.0   Median :220.0   Median :225.0
## Mean   : 9.664   Mean   :219.3   Mean   :224.5   Mean   :228.8
## 3rd Qu.:20.000   3rd Qu.:246.0   3rd Qu.:248.0   3rd Qu.:254.0
## Max.   :60.000   Max.   :503.0   Max.   :479.0   Max.   :500.0
## NA's   :4      NA's   :357   NA's   :387
## cholst_6 cholst_8 cholst_10 dead
## Min.   :126.0   Min.   :135.0   Min.   :115.0   Min.   :0.0000
## 1st Qu.:208.2   1st Qu.:210.0   1st Qu.:218.0   1st Qu.:0.0000
## Median :236.0   Median :237.0   Median :246.0   Median :0.0000
## Mean   :238.4   Mean   :240.8   Mean   :249.3   Mean   :0.2088
## 3rd Qu.:265.0   3rd Qu.:266.0   3rd Qu.:276.0   3rd Qu.:0.0000
## Max.   :545.0   Max.   :696.0   Max.   :525.0   Max.   :1.0000
## NA's   :412   NA's   :456   NA's   :509
##      id
## Min.   : 1.0
## 1st Qu.: 659.2
## Median :1317.5
## Mean   :1317.5
## 3rd Qu.:1975.8
## Max.   :2634.0
##
```

Some brief summary statistics of the data show the number of missing values.

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##      between, first, last

## The following object is masked from 'package:purrr':
##
##      transpose

library(dplyr)
long_df <- melt(setDT(df), id.vars = c("age", "gender", "BMI_base", "BMI_10yrs",
                                       "cigarette_base", "cholst_base", "dead",
                                       "id"), variable.name = "year")

long_df$year = case_when(
  long_df$year == "cholst_2" ~ 2,
  long_df$year == "cholst_4" ~ 4,
```

```

long_df$year == "cholst_6" ~ 6,
long_df$year == "cholst_8" ~ 8,
long_df$year == "cholst_10" ~ 10)

long_df$age_current = long_df$year + long_df$age

```

Now we have transformed the variables to the desirable shape, so we can use the cholesterol level as the response variable across the different years. A new variable age has also been included to show the current age of the individual.

```

library(lme4)
lmm = lmer(value~ age+age_current+gender+gender*age_current + BMI_base
           +(1|id), data = long_df, REML = T)
summary(lmm)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## value ~ age + age_current + gender + gender * age_current + BMI_base +
##       (1 | id)
## Data: long_df
##
## REML criterion at convergence: 105303.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -5.1217 -0.5414 -0.0241  0.5151 12.0112
##
## Random effects:
## Groups Name Variance Std.Dev.
## id      (Intercept) 1278.6  35.76
## Residual          460.6  21.46
## Number of obs: 11030, groups: id, 2501
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    277.0456   10.7725  25.718
## age             -1.9197    0.1184 -16.218
## age_current      0.4381    0.1920   2.282
## gender          -80.1689   5.8286 -13.754
## BMI_base         0.5942    0.1802   3.298
## age_current:gender 1.7051    0.1154  14.779
##
## Correlation of Fixed Effects:
##              (Intr) age    ag_crr gender BMI_bs
## age           -0.178
## age_current  -0.789 -0.245
## gender        -0.866 -0.017  0.892
## BMI_base      -0.427 -0.169  0.087  0.106
## ag_crrnt:gn   0.838  0.005 -0.923 -0.966 -0.094

```

The results above have been obtained by modelling a LMM with random intercept. A random intercept/slope model has been attempted but the result does not converge. Therefore, I simply kept the simple way. It turns out that with the individual id as the random effect, there are indeed variations among the different individuals. As we can see from the results, the current age shows that as people grow older, their cholesterol level goes up. The other effects can also be observed from the result, such as females have lower levels and

BMI at the baseline seems to have a positive relationship with the cholesterol level.

**b**

In this case, we would not need to transform the original dataset to the long format, since each individual will simply have one row of concern. Therefore, modeling a logistic regression will help us study the effect of the cholesterol level change rate and baseline cholesterol level on the living status of the individual.

```
df$cholst_change = df$cholst_10/df$cholst_base - 1
glm = glm(dead ~ cholst_base + cholst_change + age +
          gender + gender*age + BMI_base, data = df, family = "binomial")
summary(glm)
```

```
##
## Call:
## glm(formula = dead ~ cholst_base + cholst_change + age + gender +
##      gender * age + BMI_base, family = "binomial", data = df)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.6235  -0.6644  -0.4619  -0.2980   2.5202
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.127226   1.242869  -4.930 8.23e-07 ***
## cholst_base    0.001872   0.001535   1.220   0.223
## cholst_change  0.134133   0.403636   0.332   0.740
## age           0.117841   0.022710   5.189 2.12e-07 ***
## gender       -0.484939   0.707745  -0.685   0.493
## BMI_base      0.010855   0.014396   0.754   0.451
## age:gender    -0.007109   0.014983  -0.474   0.635
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2173.8  on 2120  degrees of freedom
## Residual deviance: 1870.9  on 2114  degrees of freedom
## (513 observations deleted due to missingness)
## AIC: 1884.9
##
## Number of Fisher Scoring iterations: 5
```

```
glm1 = glm(dead ~ cholst_base + cholst_change + gender +
            BMI_base, data = df, family = "binomial")
summary(glm1)
```

```
##
## Call:
## glm(formula = dead ~ cholst_base + cholst_change + gender + BMI_base,
##      family = "binomial", data = df)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.2680  -0.7321  -0.6037  -0.4765   2.3820
##
```

```
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.508150   0.469886  -5.338 9.41e-08 ***
## cholst_base  0.004756   0.001385   3.433 0.000596 ***
## cholst_change -0.358345   0.373019  -0.961 0.336723
## gender      -0.592988   0.110889  -5.348 8.91e-08 ***
## BMI_base     0.041618   0.013075   3.183 0.001457 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2173.8  on 2120  degrees of freedom
## Residual deviance: 2104.6  on 2116  degrees of freedom
## (513 observations deleted due to missingness)
## AIC: 2114.6
##
## Number of Fisher Scoring iterations: 4
```

It's apparent from the result that age is the most important determining factor for a person's live and death. It makes sense since as people grow older they are more likely to die. Controlling for age, the effect of cholesterol level seems insignificant. But if we remove age and the interaction term, cholesterol baseline level would be significant, as shown in the summary for the second model. Here, the age seems like a confounding variable, since it simultaneously affects (increases) the cholesterol level and a person's living status (more likely to die). The direction(signs) of the estimates though are mostly consistent with the results from problem 1.

## c

A possible approach is to use the GLMM, which combines the logistic regression while taking into account the random effect of each individual. Moreover, it's possible that by adjusting the variables included, as well as higher order and interaction terms, we might obtain models with better fit, in terms of AIC/BIC, etc. It might also be possible that with the GLMM, we can get more detailed fit by creating a few new variables, such the increase rate of the cholesterol level every 2 years, so we may obtain a more precise and accurate prediction and better modeling results. Also, we may want to model the people who die in the process as well. To do this, we can do something more like survival analysis. In addition, complex causal models can be used to determine the causality in the model. In summary, we want to make the best use of the data available and capture the trend as much as possible.