Practical 3

02/03/2022

1. We will proceed with analysis of the dataset of tutorial 1 and 2 on serum cholesterol from the National Cooperative Gallstone Study.

Lets start with model 4 of exercise 3 of tutorial 2. The code to fit this model was

```
dataall<-read.table("datachol.txt")
colnames(dataall)<-c("grp","id","t1","t2","t3","t4","t5")
data<-as.data.frame(dataall)
dataalllong<-reshape(dataall, idvar="id",
                     varying=c("t1","t2","t3","t4","t5"),
                     v.names="Y",timevar="time",time=c(0,6,12,20,24),
                     direction="long")
dataalllong$Y<-as.numeric(dataalllong$Y)
week.f <- factor(dataalllong$time, c(0,6,12,20,24))
tt<-as.integer(week.f)
model4 <- lme(Y ~ grp*week.f, random= ~1| id,data=dataalllong,method="REML",na.action=na.omit)
summary(model4)
```

```
## Linear mixed-effects model fit by REML
##   Data: dataalllong
##        AIC      BIC    logLik
##   4315.728 4364.688 -2145.864
##
## Random effects:
##  Formula: ~1 | id
##         (Intercept) Residual
## StdDev:    37.51785 23.71617
##
## Fixed effects:  Y ~ grp * week.f
##                    Value Std.Error  DF   t-value p-value
## (Intercept)    216.10543 13.234410 336 16.329056  0.0000
## grp              9.91070  8.934474 101  1.109265  0.2700
## week.f6         31.78836 10.000599 336  3.178645  0.0016
## week.f12        41.64067 10.385922 336  4.009338  0.0001
## week.f20        32.91190 11.079720 336  2.970463  0.0032
## week.f24        35.98146 11.627740 336  3.094450  0.0021
## grp:week.f6    -12.27223  6.751347 336 -1.817745  0.0700
## grp:week.f12   -16.42571  6.978203 336 -2.353860  0.0192
## grp:week.f20    -4.79793  7.324612 336 -0.655042  0.5129
## grp:week.f24    -7.59439  7.659611 336 -0.991485  0.3222
##  Correlation:
##              (Intr) grp    wek.f6 wk.f12 wk.f20 wk.f24 grp:.6 gr:.12 gr:.20
## grp          -0.944
## week.f6      -0.378  0.357
## week.f12     -0.364  0.343  0.481
## week.f20     -0.341  0.322  0.451  0.451
## week.f24     -0.325  0.307  0.430  0.427  0.428
```

```
## grp:week.f6   0.357 -0.378 -0.944 -0.454 -0.426 -0.406
## grp:week.f12  0.345 -0.366 -0.457 -0.944 -0.427 -0.405  0.484
## grp:week.f20  0.329 -0.348 -0.435 -0.434 -0.945 -0.409  0.461  0.462
## grp:week.f24  0.314 -0.333 -0.416 -0.413 -0.411 -0.945  0.441  0.440  0.440
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -2.75298512 -0.57321896 -0.04586449  0.50818935  2.82120637
##
## Number of Observations: 447
## Number of Groups: 103
```

Write down the estimates for the mean of the response Y for the five time points for grp=0 and for grp=1 based on this model.

```
#mean at time 0 for group1
mean_time0_grp1<-model4$coefficients$fixed[1]
#mean at time 0 for group2
mean_time0_grp2<-sum(model4$coefficients$fixed[1:2])
#mean at time 6 for group1
mean_time6_grp1<-sum(model4$coefficients$fixed[c(1,3)])
#mean at time 6 for group2
mean_time6_grp2<-sum(model4$coefficients$fixed[1:3])+model4$coefficients$fixed[7]
#mean at time 12 for group1
mean_time12_grp1<-sum(model4$coefficients$fixed[c(1,4)])
#mean at time 12 for group2
mean_time12_grp2<-sum(model4$coefficients$fixed[1:2])+sum(model4$coefficients$fixed[c(4,8)])
#mean at time 20 for group1
mean_time20_grp1<-sum(model4$coefficients$fixed[c(1,5)])
#mean at time 20 for group2
mean_time20_grp2<-sum(model4$coefficients$fixed[1:2])+sum(model4$coefficients$fixed[c(5,9)])
#mean at time 24 for group1
mean_time24_grp1<-sum(model4$coefficients$fixed[c(1,6)])
#mean at time 24 for group2
mean_time24_grp2<-sum(model4$coefficients$fixed[1:2])+sum(model4$coefficients$fixed[c(6,10)])

mean_grp1_mod4<-c(mean_time0_grp1,mean_time6_grp1,mean_time12_grp1,mean_time20_grp1,mean_time24_grp1)
mean_grp2_mod4<-c(mean_time0_grp2,mean_time6_grp2,mean_time12_grp2,mean_time20_grp2,mean_time24_grp2)
mean_mod4<-as.data.frame(cbind(as.numeric(mean_grp1_mod4),as.numeric(mean_grp2_mod4)))
colnames(mean_mod4)<-c("group 1","group 2")
```

Use a wald type of test statistic to test whether there is a difference in mean between the response at time=20 and time = 24 in grp=0.

```
#you can use the linearHypothesis function
#you need to specify your linear hypothesis in ""
#use the exact names of the coefficients
linearHypothesis(model4,"week.f24 - week.f20=0")
```

```
## Linear hypothesis test
##
## Hypothesis:
## - week.f20  + week.f24 = 0
##
## Model 1: restricted model
## Model 2: Y ~ grp * week.f
```

```
##
##   Df  Chisq Pr(>Chisq)
## 1
## 2  1 0.0638     0.8005
```

```
#p-value>0.05
#The two means are not significantly different
```

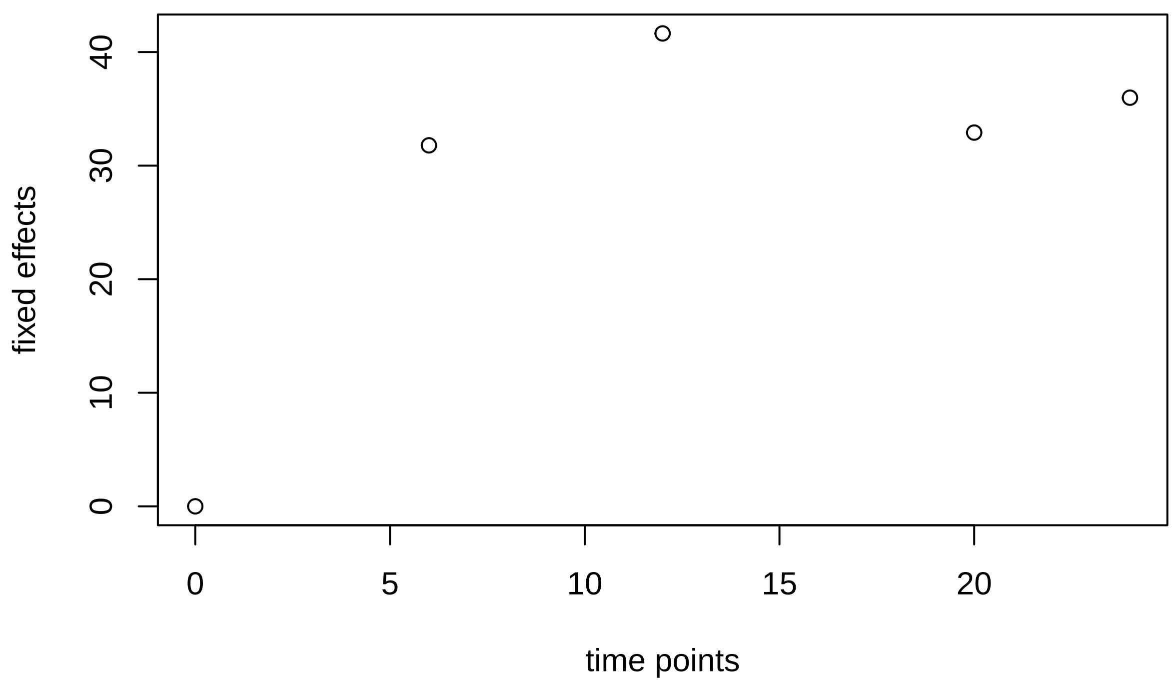
Question (b)

Now looking at the coefficents in (a) the relationship might not be linear in time. On the other hand the standard errors are huge.

```
model4$coef$fixed
```

```
##  (Intercept)          grp      week.f6      week.f12      week.f20      week.f24
##   216.105429     9.910700    31.788356     41.640671     32.911896     35.981462
##  grp:week.f6 grp:week.f12 grp:week.f20 grp:week.f24
##   -12.272227   -16.425710    -4.797929    -7.594387
```

```
#plot coefficients from model 4
plot(c(0,6,12,20,24),c(0,model4$coef$fixed[3:6]),xlab="time points",ylab="fixed effects")
```



So let's fit a linear model while keeping in mind that the mean structure might not fit well. Include a random intercept in the model.

```
#time is treated as a continuous covariate
model5<-lme(Y ~ grp*time, random= ~1| id,data=dataalllong, method="REML",na.action=na.omit)
summary(model5)
```

```
## Linear mixed-effects model fit by REML
##   Data: dataalllong
##        AIC      BIC   logLik
##   4360.359 4384.921 -2174.18
##
## Random effects:
```

```
##  Formula: ~1 | id
##         (Intercept) Residual
## StdDev:     37.4473   24.036
##
## Fixed effects:  Y ~ grp * time
##                 Value Std.Error  DF   t-value p-value
## (Intercept) 229.10674 12.509862 342 18.314089  0.0000
## grp           3.50820  8.439952 101  0.415666  0.6785
## time          1.32052  0.420309 342  3.141778  0.0018
## grp:time     -0.18082  0.277166 342 -0.652372  0.5146
##  Correlation:
##          (Intr) grp    time
## grp      -0.944
## time     -0.348  0.326
## grp:time  0.333 -0.350 -0.945
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -2.97385627 -0.59998880 -0.03179352  0.55140671  2.57800057
##
## Number of Observations: 447
## Number of Groups: 103
```

Write down the estimates for the mean of the response Y for the five time points for grp=0 and for grp=1 based on this new model.

```r
#mean at time 0 for group1
mean_time0_grp1_mod5<-model5$coefficients$fixed[1]
#mean at time 0 for group2
mean_time0_grp2_mod5<-sum(model5$coefficients$fixed[1:2])
#mean at time 6 for group1
mean_time6_grp1_mod5<-model5$coefficients$fixed[1]+6*model5$coefficients$fixed[3]
#mean at time 6 for group2
mean_time6_grp2_mod5<-sum(model5$coefficients$fixed[1:2])+6*sum(model5$coefficients$fixed[3:4])
#mean at time 12 for group1
mean_time12_grp1_mod5<-model5$coefficients$fixed[1]+12*model5$coefficients$fixed[3]
#mean at time 12 for group2
mean_time12_grp2_mod5<-sum(model5$coefficients$fixed[1:2])+12*sum(model5$coefficients$fixed[3:4])
#mean at time 20 for group1
mean_time20_grp1_mod5<-model5$coefficients$fixed[1]+20*model5$coefficients$fixed[3]
#mean at time 20 for group2
mean_time20_grp2_mod5<-sum(model5$coefficients$fixed[1:2])+20*sum(model5$coefficients$fixed[3:4])
#mean at time 24 for group1
mean_time24_grp1_mod5<-model5$coefficients$fixed[1]+24*model5$coefficients$fixed[3]
#mean at time 24 for group2
mean_time24_grp2_mod5<-sum(model5$coefficients$fixed[1:2])+24*sum(model5$coefficients$fixed[3:4])

mean_grp1_mod5<-c(mean_time0_grp1_mod5,mean_time6_grp1_mod5,
                  mean_time12_grp1_mod5,mean_time20_grp1_mod5,mean_time24_grp1_mod5)
mean_grp2_mod5<-c(mean_time0_grp2_mod5,mean_time6_grp2_mod5,
                  mean_time12_grp2_mod5,mean_time20_grp2_mod5,mean_time24_grp2_mod5)
mean_mod5<-as.data.frame(cbind(as.numeric(mean_grp1_mod5),as.numeric(mean_grp2_mod5)))
colnames(mean_mod5)<-c("group 1","group 2")
mean_mod5$times<-c(0,6,12,20,24)
```

Compare these estimates with the ones in (a).

```
print(cbind(mean_mod4,mean_mod5))
```

```
##    group 1  group 2  group 1  group 2 times
## 1 216.1054 226.0161 229.1067 232.6149     0
## 2 247.8938 245.5323 237.0298 239.4531     6
## 3 257.7461 251.2311 244.9529 246.2914    12
## 4 249.0173 254.1301 255.5171 255.4090    20
## 5 252.0869 254.4032 260.7991 259.9678    24
```

What is the estimate for the variance of the random intercept?

```
getVarCov(model5)
```

```
## Random effects variance covariance matrix
##             (Intercept)
## (Intercept)      1402.3
##   Standard Deviations: 37.447
```

```
#1402.3
```

Question c

Now fit a random slope model. Give the covariance matrix of the random effects.

```
#fit the model
model6<-lme(Y ~ grp*time, random= ~1+time| id,data=dataalllong, method="REML",na.action=na.omit)
summary(model6)
```

```
## Linear mixed-effects model fit by REML
##   Data: dataalllong
##        AIC      BIC    logLik
##   4359.447 4392.195 -2171.723
##
## Random effects:
##  Formula: ~1 + time | id
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev     Corr
## (Intercept) 39.9143997 (Intr)
## time         0.7971145 -0.372
## Residual    22.8506287
##
## Fixed effects:  Y ~ grp * time
##                 Value Std.Error  DF   t-value p-value
## (Intercept) 228.82217 13.060134 342 17.520660  0.0000
## grp           3.67168  8.811358 101  0.416698  0.6778
## time          1.37825  0.475188 342  2.900434  0.0040
## grp:time     -0.21388  0.314361 342 -0.680378  0.4967
##  Correlation:
##          (Intr) grp    time
## grp      -0.944
## time     -0.442  0.415
## grp:time  0.423 -0.445 -0.945
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
```

```
## -2.53100392 -0.58867998 -0.04753427  0.55035505  2.49571641
##
## Number of Observations: 447
## Number of Groups: 103
```

```r
#get the variance-covariance matrix
getVarCov(model6)
```

```
## Random effects variance covariance matrix
##             (Intercept)      time
## (Intercept)    1593.200 -11.82200
## time            -11.822   0.63539
##   Standard Deviations: 39.914 0.79711
```

Question d

Now compare the models from b (model5) and c (model6) using

```r
#compare the two models
anova(model5,model6)
```

```
##        Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## model5     1  6 4360.359 4384.921 -2174.180
## model6     2  8 4359.447 4392.195 -2171.723 1 vs 2 4.912696  0.0857
```

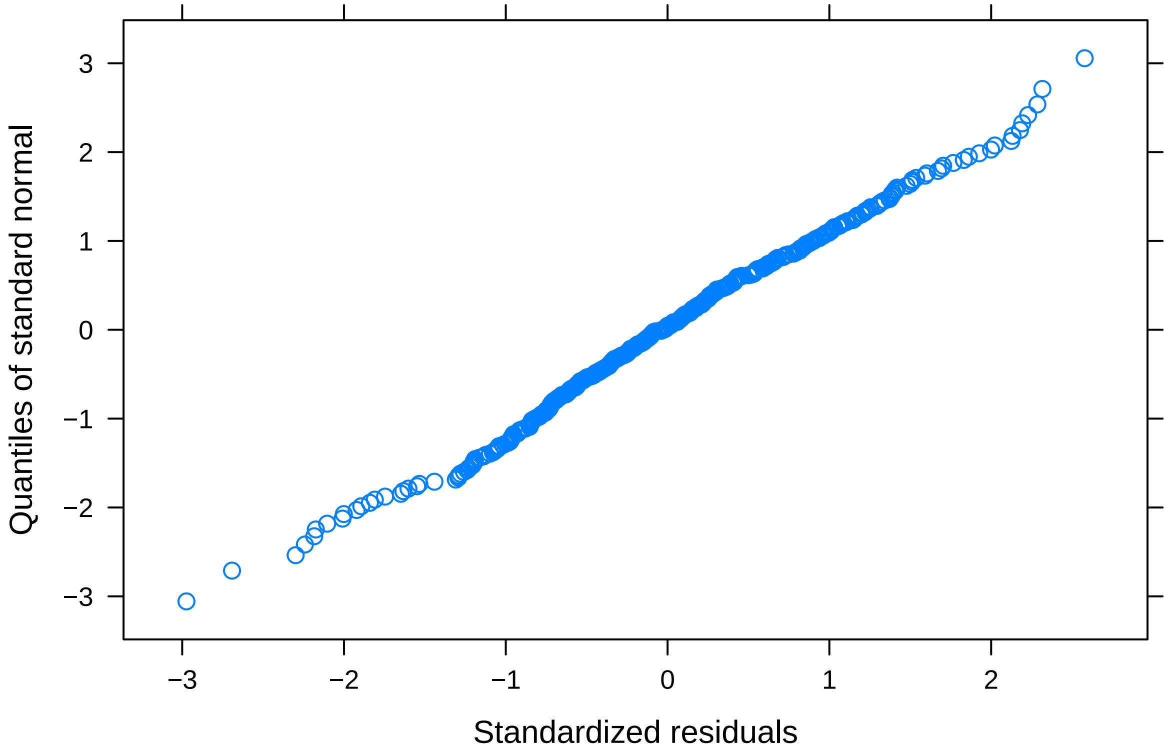
```r
#mixture of two chi-squares with 1 dof and 2 dof
critical_value<-(qchisq(0.95,1)+qchisq(0.95,2))/2
#value of the LRT
test_value<-4.912
#decision
test_value<critical_value
```

```
## [1] TRUE
```

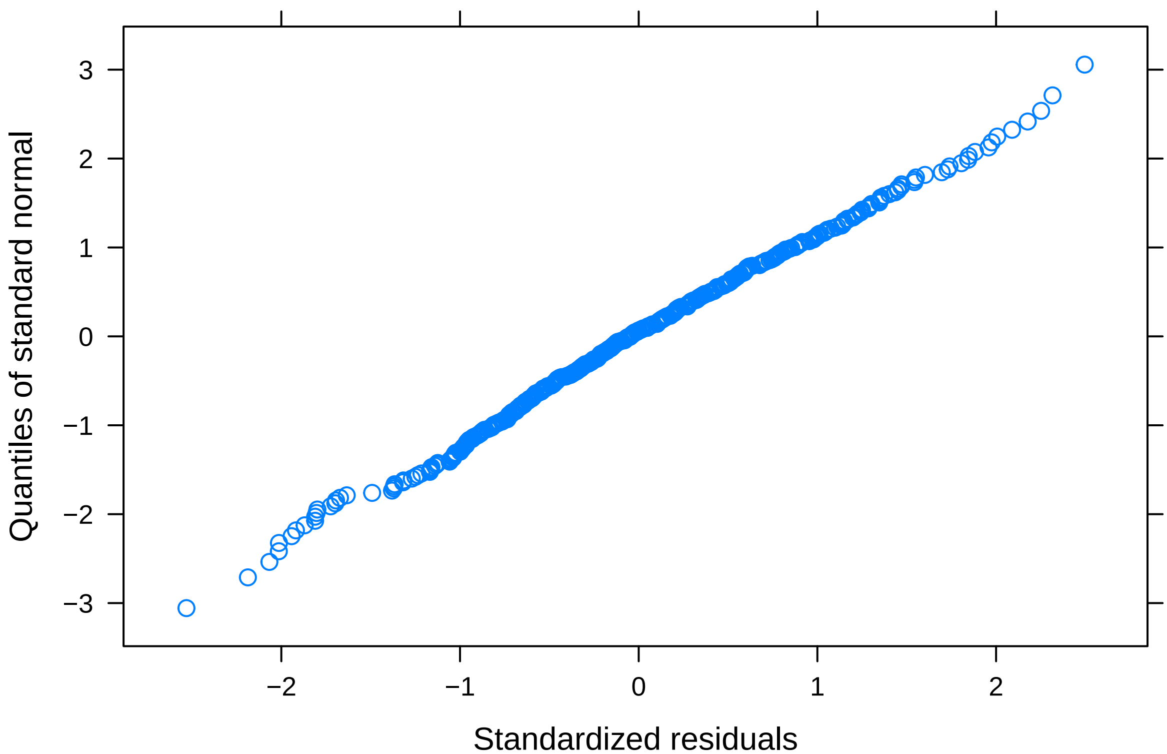
Question e

Check the fit of the two models:

```r
qqnorm(model5, ~ resid(., type = "n") )
```




```
qqnorm(model6, ~ resid(., type = "n") )
```




```
Variogram(model5)
```

```
##      variog dist n.pairs
## 1 1.0786336    0     125
```

```
## 2 1.1006838    1      272
## 3 0.8984578    2      212
## 4 0.8089512    3      126
## 5 1.3633947    4       63
```

```
Variogram(model6)
```

```
##      variog dist n.pairs
## 1 1.0273428    0     125
## 2 0.9830969    1     272
## 3 0.8428731    2     212
## 4 0.7741358    3     126
## 5 1.2733589    4      63
```

Formulate your conclusions:

Question f

Finally compare model 5 with model 4. Can we assume a linear model for time in the mean structure? For comparisson of the models, you can use the function anova, but you have to use update to change the method of estimation (Why?):

```
anova(update(model5,method="ML"),update(model4,method="ML"))
```

```
##                                Model df      AIC      BIC    logLik   Test
## update(model5, method = "ML")     1  6 4367.992 4392.607 -2177.996
## update(model4, method = "ML")     2 12 4364.788 4414.019 -2170.394 1 vs 2
##                                L.Ratio p-value
## update(model5, method = "ML")
## update(model4, method = "ML") 15.2035  0.0187
```

What is your conclusion? From the models 4 to 6 which one would you prefer? What would be a next model to fit?

Exercise 2 Fit a model with randomly varying intercepts and slopes, and allow the mean values of the intercept and slope to depend on treament group (i.e inlude main effect of treatment, a linear time trend, and a treatment by linear time interaction as fixed effects)

```
#read the data set
dat <- read.dta13("exercise.dta")
#read the data in long format
datalong <- reshape(dat, idvar="ID", varying=c("y0","y2","y4","y6","y8","y10","y12"),v.names="Y",
                    timevar="time",time=c(0,2,4,6,8,10,12), direction="long")
#fit the linear mixed model with random intercept and slope
model7<-lme(Y ~ group*time, random= ~1+time| ID,data=datalong, method="REML",na.action=na.omit)
#look at the summary of your model
summary(model7)
```

```
## Linear mixed-effects model fit by REML
##   Data: datalong
##        AIC      BIC    logLik
##   834.5395 862.2162 -409.2698
##
## Random effects:
##  Formula: ~1 + time | ID
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev    Corr
## (Intercept) 3.1548522 (Intr)
## time        0.1852790 -0.029
## Residual    0.8152861
##
## Fixed effects:  Y ~ group * time
##                Value Std.Error  DF  t-value p-value
## (Intercept) 79.00099 1.7486351 200 45.17866  0.0000
## group        1.13141 1.0637314  35  1.06363  0.2948
## time         0.06501 0.1104520 200  0.58857  0.5568
## group:time   0.05198 0.0674459 200  0.77066  0.4418
##  Correlation:
##            (Intr) group  time
## group      -0.954
## time       -0.084  0.081
## group:time  0.081 -0.085 -0.953
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -1.94321191 -0.61990241 -0.06124059  0.53517425  3.25115453
##
## Number of Observations: 239
## Number of Groups: 37
```

What is the estimated covariance matrix of the random effects?

```
getVarCov(model7)
```

```
## Random effects variance covariance matrix
##             (Intercept)      time
## (Intercept)    9.953100 -0.016846
## time          -0.016846  0.034328
##   Standard Deviations: 3.1549 0.18528
```

Based on the model provide 95% intervals for the values of the intercepts for subjects and for the values of the slopes for subjects.

```r
#95% confidence interval for the intercept
lower_bound_intercept<-model7$coefficients$fixed[1]-
  1.96*9.953
upper_bound_intercept<-model7$coefficients$fixed[1]+
  1.96*9.953
print(c(lower_bound_intercept,upper_bound_intercept))
```

```
## (Intercept) (Intercept)
##    59.49311    98.50887
```

```r
#95% confidence interval for the slope
lower_bound_slope<-model7$coefficients$fixed[3]-
  1.96*0.0343
upper_bound_slope<-model7$coefficients$fixed[3]+
  1.96*0.0343
```

Question (b)

Do we need the random slope in the model?

```r
#fit the model without the random slope
model8<-lme(Y ~ group*time, random= ~1| ID,data=datalong, method="REML",na.action=na.omit)
#compare the two models
anova(model8,model7)
```

```
##        Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## model8     1  6 893.2104 913.9679 -440.6052
## model7     2  8 834.5395 862.2162 -409.2698 1 vs 2 62.67088  <.0001
```

```r
#mixtures of two chi-squares with 1 dof and 2 dof
critical_value<-(qchisq(0.95,1)+qchisq(0.95,2))/2
#value of the LRT
observed_test<-62.671
#decision
observed_test<critical_value
```

```
## [1] FALSE
```

Question (c)

Give the mean intercept and slope for the two groups based on model from (a)

```r
#mean intercept group 1
mean(coef(model7)[1:16,1])
```

```
## [1] 79.00099
```

```r
#mean intercept group 2
mean(coef(model7)[17:37,1])
```

```
## [1] 79.00099
```

```r
#give the mean slope for the two groups
#mean slope group 1
mean(coef(model7)[1:16,3])
```

```
## [1] 0.0650088
```

```
#mean slope group 2
mean(coef(model7)[17:37,3])
```

```
## [1] 0.0650088
```

Question d

Based on the previous results, interpret the effect of treatment on changes in strength. Does your analyses suggest a difference between the two groups?

Question e

Give the estimate of $VAR(Y_{i1}|b)$ and $VAR(Y_{i1})$

```
#conditional variance at time 1 VAR(Yi1|bi)
cond_var<-(sigma(model7))^2
print(cond_var)
```

```
## [1] 0.6646913
```

```
#extract the variance of the random intercept and slope
getVarCov(model7)
```

```
## Random effects variance covariance matrix
##             (Intercept)      time
## (Intercept)    9.953100 -0.016846
## time          -0.016846  0.034328
##   Standard Deviations: 3.1549 0.18528
```

```
#unconditional variance at time 1 VAR(Yi1)
uncond_var<-(sigma(model7))^2+9.953+1*0.03-0.016846
print(uncond_var)
```

```
## [1] 10.63085
```

Question f

Obtain the predicted intercept and slope (BLUP) for each subject.

```
coef(model7)[c(1,3)]
```

```
##    (Intercept)        time
## 1     77.94893 -0.01220805
## 2     81.90997  0.29345942
## 3     80.64260  0.03072013
## 4     79.85749  0.04376234
## 5     79.00010  0.32934428
## 6     75.00374 -0.09183001
## 7     80.96411  0.21457446
## 8     76.39267  0.29321837
## 9     84.12637  0.12995432
## 10    74.29415  0.24158987
## 11    75.61046 -0.07488097
## 12    83.03027  0.12889824
## 13    77.80212  0.17520680
## 14    76.61512 -0.25501273
## 15    77.41757 -0.33869116
```

```
## 16    83.40024 -0.06796457
## 17    81.69061 -0.14792964
## 18    72.59772  0.02474448
## 19    80.40836 -0.12412992
## 20    84.16748 -0.06474716
## 21    80.23512  0.16815051
## 22    76.89080 -0.05468079
## 23    76.39151  0.23746848
## 24    85.70517  0.28128515
## 25    78.45417  0.08692145
## 26    79.71431  0.24453411
## 27    76.82971  0.11565101
## 28    77.21799  0.28144243
## 29    81.25438 -0.08049023
## 30    78.79331  0.14885736
## 31    75.87732 -0.02703365
## 32    80.06357  0.03045511
## 33    77.21408 -0.03153779
## 34    78.87855 -0.07959315
## 35    83.29318 -0.03670933
## 36    75.67447  0.29256266
## 37    77.66906  0.09996365
```

```
#predicted intercept per subject: first column
#predicted slope per subject: second column
```

Question g

Now select the data on subject 24 and estimate a linear model for this subject by using OLS and compare the obtained estimates with the ones obtained in (f). How and why are they different?

```
#select data from subject 24
data_subject24<-datalong[datalong$id=="24",]
#fit a linear model by using OLS
model_subject24<-lm(Y~time,data_subject24)

coef_subject24<-as.data.frame(model_subject24$coefficients)
#slope and intercept for subject 24 obtained in f
coef(model7)[24,c(1,3)]
```

```
##    (Intercept)      time
## 24    85.70517 0.2812851
```

```
#look at the estimates for the slope and the intercept
#slope and intercept for subject 24 using the linear model
model_subject24$coefficients
```

```
## (Intercept)        time
##       87.80        0.45
```

Exercise 3

Simulation study on missing data: ## Question a Check that you understand the program. How many individuals are in the data? How many time points? Why is the missing data mechanism MAR in datMAR and why MNAR in MNARdat?

```r
#set your seed to ensure the consistency of your results
set.seed(123)
#generate 50 values from a normal with mean 0 and variance 0.5
#repeat the generated value 4 times
a<-rep(rnorm(50,0,0.5),each=4)
#generate 50 values from a normal with mean 0 and variance 0.3
#repeat the generated value 4 times
b<-rep(rnorm(50,0,0.3),each=4)
#generate 200 values for the error term from a normal with mean 0
#and variance 0.2
e<-rnorm(200,0,0.2)
#assign to each observation the values 0,1,2,3
t<-rep(c(0,1,2,3),50)
#define the covariate x
#x=0 to half of the subjects
#x=1 to the other half
x<-c(rep(0,100),rep(1,100))
#calculate the responses
y<-0.3+a+(0.1+b)*t+0.3*t*x+0.1*x+e
#define an ID variable in our data set
id<-rep(1:50,each=4)
#store your simulated data set in a dataframe
dat<-as.data.frame(cbind(id,y,t,x))
colnames<-c("id", "y", "t","x")
#fit the model with random slope and intercept
model<-lme(y~ t*x, random= ~1+t| id,data=dat, method="REML",na.action=na.omit)

#time points: 4
#individuals: 50

#simulate MNAR mechanism
datMNAR<-numeric()
for (i in 1:50)
{tt<-4
datind<-dat[(id==i),]
if (sum(datind$y>2)>0)
  tt<-min(datind$t[(datind$y>2)])
if (tt==0) {tt<-tt+1}
datMNAR<-rbind(datMNAR,datind[(1:tt),])
}
#fit the model with random slope and intercept under MNAR
model2<-lme(y~ t*x, random= ~1+t| id,data=datMNAR, method="REML",na.action=na.omit)
#simulate MAR mechanism
datMAR<-numeric()
for (i in 1:50)
{tt<-4
datind<-dat[(id==i),]
if (sum(datind$y>1.7)>0)
  tt<-(min(datind$t[(datind$y>1.7)])+1)
if (tt==5) {tt<-tt-1}
datMAR<-rbind(datMAR,datind[(1:tt),])

}
```

```
#fit the model with random slope and intercept under MAR
model3<-lme(y~ t*x, random= ~1+t| id,data=datMAR, method="REML",na.action=na.omit)
```

Question b

If you fit a model with maximal mean structures to the MNAR data set which parameter would be most biased? Which one would have the largest standard error?

```
week.f <- factor(datMNAR$t, c(0,1,2,3))
tt<-as.integer(week.f)
model_mean<-gls(y ~ week.f*x, corr=corSymm(, form= ~ tt | id),weights = varIdent(form = ~ 1 | week.f), 
#parameter estimate
summary(model_mean)
```

```
## Generalized least squares fit by REML
##   Model: y ~ week.f * x
##   Data: datMNAR
##        AIC      BIC    logLik
##   218.5929 275.3521 -91.29644
##
## Correlation Structure: General
##  Formula: ~tt | id
##  Parameter estimate(s):
##  Correlation:
##   1     2     3
## 2 0.639
## 3 0.503 0.903
## 4 0.346 0.823 0.894
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | week.f
##  Parameter estimates:
##        0        1        2        3
## 1.000000 1.051740 1.277038 1.520731
##
## Coefficients:
##                 Value  Std.Error  t-value p-value
## (Intercept) 0.2787550 0.10326824 2.699329  0.0076
## week.f1     0.0764313 0.09010911 0.848208  0.3975
## week.f2     0.1563714 0.12033471 1.299470  0.1955
## week.f3     0.1970795 0.15738650 1.252200  0.2122
## x           0.2266801 0.14604334 1.552142  0.1225
## week.f1:x   0.3574063 0.12743353 2.804649  0.0056
## week.f2:x   0.7488790 0.17261366 4.338469  0.0000
## week.f3:x   1.0524532 0.22804098 4.615193  0.0000
##
##  Correlation:
##           (Intr) wek.f1 wek.f2 wek.f3 x      wk.f1: wk.f2:
## week.f1   -0.375
## week.f2   -0.307  0.883
## week.f3   -0.310  0.840  0.888
## x         -0.707  0.265  0.217  0.220
## week.f1:x  0.265 -0.707 -0.625 -0.594 -0.375
## week.f2:x  0.214 -0.616 -0.697 -0.619 -0.303  0.871
```

```
## week.f3:x  0.214 -0.580 -0.613 -0.690 -0.303  0.820  0.873
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.6058140 -0.7767618 -0.1417359  0.5324833  2.1792643
##
## Residual standard error: 0.5163412
## Degrees of freedom: 181 total; 173 residual
```

```
#interaction between time 4 and x
```

Question c

Based on your simulation model what should the value of this parameter from (b) be?

```
model_mean$coefficients[8]
```

```
## week.f3:x
##  1.052453
```

Question d

Fit a maximal mean structure model using unstructured, independence structure equal variance, LMM with random intercept and random slopes to complete dataset and MAR dataset (total 6 models), and fit a LMM with random intercept and random slopes to the MNAR dataset.

```
#fit the maximal mean structure model with unstructured variance
#datMAR
week.f <- factor(datMAR$t, c(0,1,2,3))
tt<-as.integer(week.f)
model3<-gls(y ~ week.f*x, corr=corSymm(, form= ~ tt | id),
            weights = varIdent(form = ~ 1 | week.f), data=datMAR, method="REML")
summary(model3)
```

```
## Generalized least squares fit by REML
##   Model: y ~ week.f * x
##   Data: datMAR
##        AIC      BIC    logLik
##   232.6182 290.0915 -98.30912
##
## Correlation Structure: General
##  Formula: ~tt | id
##  Parameter estimate(s):
##  Correlation:
##   1     2     3
## 2 0.639
## 3 0.498 0.907
## 4 0.418 0.833 0.923
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | week.f
##  Parameter estimates:
##        0        1        2        3
## 1.000000 1.051742 1.377114 1.823411
##
## Coefficients:
```

```
##                 Value  Std.Error  t-value p-value
## (Intercept) 0.2787550 0.10326827 2.699328  0.0076
## week.f1     0.0764313 0.09010894 0.848210  0.3974
## week.f2     0.1757511 0.12752047 1.378219  0.1698
## week.f3     0.2929507 0.17344442 1.689018  0.0929
## x           0.2266801 0.14604339 1.552142  0.1224
## week.f1:x   0.3574063 0.12743329 2.804654  0.0056
## week.f2:x   0.8095470 0.18114563 4.469039  0.0000
## week.f3:x   1.1784374 0.25153845 4.684920  0.0000
##
##  Correlation:
##           (Intr) wek.f1 wek.f2 wek.f3 x      wk.f1: wk.f2:
## week.f1   -0.375
## week.f2   -0.254  0.887
## week.f3   -0.142  0.794  0.902
## x         -0.707  0.265  0.180  0.100
## week.f1:x  0.265 -0.707 -0.627 -0.561 -0.375
## week.f2:x  0.179 -0.625 -0.704 -0.635 -0.253  0.883
## week.f3:x  0.098 -0.548 -0.622 -0.690 -0.138  0.774  0.883
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.2750848 -0.7533353 -0.1565574  0.5621373  2.4116044
##
## Residual standard error: 0.5163414
## Degrees of freedom: 188 total; 180 residual
```

```r
#fit the maximal mean structure model with equal variance
#datMAR
week.f <- factor(datMAR$t, c(0,1,2,3))

model4 <- gls(y ~ week.f*x,data=datMAR, method="REML")
summary(model4)
```

```
## Generalized least squares fit by REML
##   Model: y ~ week.f * x
##   Data: datMAR
##        AIC      BIC    logLik
##   397.2495 425.9862 -189.6248
##
## Coefficients:
##                 Value Std.Error   t-value p-value
## (Intercept) 0.2787550 0.1293994 2.1542210  0.0326
## week.f1     0.0764313 0.1829984 0.4176608  0.6767
## week.f2     0.1757511 0.1829984 0.9603969  0.3381
## week.f3     0.2071232 0.1848948 1.1202219  0.2641
## x           0.2266801 0.1829984 1.2386996  0.2171
## week.f1:x   0.3574063 0.2587988 1.3810196  0.1690
## week.f2:x   0.7167237 0.2615967 2.7398037  0.0068
## week.f3:x   0.8234124 0.2776565 2.9655795  0.0034
##
##  Correlation:
##           (Intr) wek.f1 wek.f2 wek.f3 x      wk.f1: wk.f2:
## week.f1   -0.707
## week.f2   -0.707  0.500
```

```
## week.f3   -0.700  0.495  0.495
## x         -0.707  0.500  0.500  0.495
## week.f1:x  0.500 -0.707 -0.354 -0.350 -0.707
## week.f2:x  0.495 -0.350 -0.700 -0.346 -0.700  0.495
## week.f3:x  0.466 -0.330 -0.330 -0.666 -0.659  0.466  0.461
##
## Standardized residuals:
##          Min           Q1          Med           Q3          Max
## -3.178020664 -0.631279343 -0.009855484  0.648733185  2.981407656
##
## Residual standard error: 0.6469971
## Degrees of freedom: 188 total; 180 residual
```

```
#fit a LMM with random intercept and slope
#datMAR
model5<-lme(y~ week.f*x, random= ~1+t| id,data=datMAR, method="REML",na.action=na.omit)
summary(model5)
```

```
## Linear mixed-effects model fit by REML
##   Data: datMAR
##        AIC      BIC    logLik
##   233.5911 271.9066 -104.7955
##
## Random effects:
##  Formula: ~1 + t | id
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev    Corr
## (Intercept) 0.4438015 (Intr)
## t           0.2657745 -0.021
## Residual    0.2079523
##
## Fixed effects:  y ~ week.f * x
##                 Value  Std.Error  DF  t-value p-value
## (Intercept) 0.2787550 0.09802121 132 2.843823  0.0052
## week.f1     0.0764313 0.07927784 132 0.964094  0.3368
## week.f2     0.1757511 0.12149613 132 1.446557  0.1504
## week.f3     0.2887065 0.17056593 132 1.692639  0.0929
## x           0.2266801 0.13862293  48 1.635228  0.1085
## week.f1:x   0.3574063 0.11211580 132 3.187832  0.0018
## week.f2:x   0.8012622 0.17335533 132 4.622080  0.0000
## week.f3:x   1.1470431 0.24773836 132 4.630059  0.0000
##  Correlation:
##           (Intr) wek.f1 wek.f2 wek.f3 x      wk.f1: wk.f2:
## week.f1   -0.236
## week.f2   -0.162  0.766
## week.f3   -0.122  0.755  0.902
## x         -0.707  0.167  0.115  0.086
## week.f1:x  0.167 -0.707 -0.542 -0.534 -0.236
## week.f2:x  0.114 -0.537 -0.701 -0.632 -0.161  0.759
## week.f3:x  0.084 -0.520 -0.621 -0.688 -0.118  0.735  0.884
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -1.80213821 -0.41075135 -0.03499759  0.45660225  1.91716652
##
```

```
## Number of Observations: 188
## Number of Groups: 50
```

```
#fit the maximal mean structure model with unstructured variance
#complete data
week.f <- factor(dat$t, c(0,1,2,3))
tt<-as.integer(week.f)
model6<-gls(y ~ week.f*x, corr=corSymm(, form= ~ tt | id),
            weights = varIdent(form = ~ 1 | week.f), data=dat, method="REML")
summary(model6)
```

```
## Generalized least squares fit by REML
##   Model: y ~ week.f * x
##   Data: dat
##        AIC      BIC    logLik
##   243.7606 302.3955 -103.8803
##
## Correlation Structure: General
##  Formula: ~tt | id
##  Parameter estimate(s):
##  Correlation:
##   1     2     3
## 2 0.639
## 3 0.505 0.905
## 4 0.361 0.822 0.911
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | week.f
##  Parameter estimates:
##        0        1        2        3
## 1.000000 1.051736 1.360446 1.797810
##
## Coefficients:
##                 Value Std.Error  t-value p-value
## (Intercept) 0.2787550 0.1032689 2.699312  0.0076
## week.f1     0.0764313 0.0901092 0.848207  0.3974
## week.f2     0.1757511 0.1254451 1.401021  0.1628
## week.f3     0.2985336 0.1768666 1.687903  0.0931
## x           0.2266801 0.1460443 1.552133  0.1223
## week.f1:x   0.3574063 0.1274337 2.804646  0.0056
## week.f2:x   0.7977259 0.1774061 4.496609  0.0000
## week.f3:x   1.0816945 0.2501271 4.324579  0.0000
##
##  Correlation:
##           (Intr) wek.f1 wek.f2 wek.f3 x      wk.f1: wk.f2:
## week.f1   -0.375
## week.f2   -0.257  0.882
## week.f3   -0.205  0.825  0.909
## x         -0.707  0.265  0.182  0.145
## week.f1:x  0.265 -0.707 -0.624 -0.583 -0.375
## week.f2:x  0.182 -0.624 -0.707 -0.643 -0.257  0.882
## week.f3:x  0.145 -0.583 -0.643 -0.707 -0.205  0.825  0.909
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
```

```
## -2.31348195 -0.71524422 -0.08049225  0.63173689  2.44113625
##
## Residual standard error: 0.5163445
## Degrees of freedom: 200 total; 192 residual
```

```
#fit the maximal mean structure model with equal variance
#complete data
week.f <- factor(dat$t, c(0,1,2,3))
model7 <- gls(y ~ week.f*x, data=dat, method="REML")
summary(model7)
```

```
## Generalized least squares fit by REML
##   Model: y ~ week.f * x
##   Data: dat
##        AIC     BIC    logLik
##   447.3695 476.687 -214.6848
##
## Coefficients:
##                 Value Std.Error  t-value p-value
## (Intercept) 0.2787550 0.1384444 2.013480  0.0455
## week.f1     0.0764313 0.1957899 0.390374  0.6967
## week.f2     0.1757511 0.1957899 0.897651  0.3705
## week.f3     0.2985336 0.1957899 1.524765  0.1290
## x           0.2266801 0.1957899 1.157772  0.2484
## week.f1:x   0.3574063 0.2768888 1.290794  0.1983
## week.f2:x   0.7977259 0.2768888 2.881034  0.0044
## week.f3:x   1.0816945 0.2768888 3.906603  0.0001
##
##  Correlation:
##           (Intr) wek.f1 wek.f2 wek.f3 x      wk.f1: wk.f2:
## week.f1   -0.707
## week.f2   -0.707  0.500
## week.f3   -0.707  0.500  0.500
## x         -0.707  0.500  0.500  0.500
## week.f1:x  0.500 -0.707 -0.354 -0.354 -0.707
## week.f2:x  0.500 -0.354 -0.707 -0.354 -0.707  0.500
## week.f3:x  0.500 -0.354 -0.354 -0.707 -0.707  0.500  0.500
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -3.10244522 -0.59003609 -0.08683032  0.62441990  3.16928574
##
## Residual standard error: 0.6922219
## Degrees of freedom: 200 total; 192 residual
```

```
#fit a LMM with random intercept and slope
#complete data
model8<-lme(y~ week.f*x, random= ~1+t| id,data=dat, method="REML",na.action=na.omit)
summary(model8)
```

```
## Linear mixed-effects model fit by REML
##   Data: dat
##        AIC      BIC    logLik
##   242.1128 281.2027 -109.0564
##
## Random effects:
```

```
##  Formula: ~1 + t | id
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev    Corr
## (Intercept) 0.4507461 (Intr)
## t           0.2712043 -0.063
## Residual    0.2041279
##
## Fixed effects:  y ~ week.f * x
##                 Value  Std.Error  DF  t-value p-value
## (Intercept) 0.2787550 0.09896267 144 2.816769  0.0055
## week.f1     0.0764313 0.07921822 144 0.964819  0.3363
## week.f2     0.1757511 0.12288913 144 1.430160  0.1548
## week.f3     0.2985336 0.17266178 144 1.729008  0.0860
## x           0.2266801 0.13995436  48 1.619671  0.1119
## week.f1:x   0.3574063 0.11203148 144 3.190231  0.0017
## week.f2:x   0.7977259 0.17379147 144 4.590133  0.0000
## week.f3:x   1.0816945 0.24418063 144 4.429895  0.0000
##  Correlation:
##           (Intr) wek.f1 wek.f2 wek.f3 x      wk.f1: wk.f2:
## week.f1   -0.252
## week.f2   -0.188  0.776
## week.f3   -0.152  0.767  0.910
## x         -0.707  0.178  0.133  0.107
## week.f1:x  0.178 -0.707 -0.548 -0.542 -0.252
## week.f2:x  0.133 -0.548 -0.707 -0.644 -0.188  0.776
## week.f3:x  0.107 -0.542 -0.644 -0.707 -0.152  0.767  0.910
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -1.82084079 -0.44060854 -0.02968237  0.40471853  1.95098066
##
## Number of Observations: 200
## Number of Groups: 50
```

```
#fit a LMM with random intercept and slope
#MNAR data
week.f <- factor(datMNAR$t, c(0,1,2,3))
model9<-lme(y~ week.f*x, random= ~1+t| id,data=datMNAR, method="REML",na.action=na.omit)
summary(model9)
```

```
## Linear mixed-effects model fit by REML
##   Data: datMNAR
##        AIC      BIC    logLik
##   218.7906 256.6301 -97.39532
##
## Random effects:
##  Formula: ~1 + t | id
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev    Corr
## (Intercept) 0.4615502 (Intr)
## t           0.2419617 -0.142
## Residual    0.2052976
##
## Fixed effects:  y ~ week.f * x
##                 Value  Std.Error  DF  t-value p-value
```

```
## (Intercept) 0.2787550 0.10102984 125 2.759135  0.0067
## week.f1     0.0764313 0.07558827 125 1.011152  0.3139
## week.f2     0.1525646 0.11390069 125 1.339453  0.1829
## week.f3     0.2085492 0.15924769 125 1.309590  0.1927
## x           0.2266801 0.14287777  48 1.586531  0.1192
## week.f1:x   0.3574063 0.10689796 125 3.343434  0.0011
## week.f2:x   0.7396319 0.16579609 125 4.461094  0.0000
## week.f3:x   1.0579469 0.23344626 125 4.531865  0.0000
##  Correlation:
##           (Intr) wek.f1 wek.f2 wek.f3 x      wk.f1: wk.f2:
## week.f1   -0.304
## week.f2   -0.256  0.740
## week.f3   -0.223  0.724  0.881
## x         -0.707  0.215  0.181  0.158
## week.f1:x  0.215 -0.707 -0.523 -0.512 -0.304
## week.f2:x  0.176 -0.508 -0.687 -0.606 -0.249  0.719
## week.f3:x  0.152 -0.494 -0.601 -0.682 -0.215  0.698  0.869
##
## Standardized Within-Group Residuals:
##           Min            Q1           Med            Q3           Max
## -1.8221183639 -0.4507489396 -0.0001470156  0.3973418973  1.8529293867
##
## Number of Observations: 181
## Number of Groups: 50
```

Compare the estimates of the parameter from question (b) in the seven models.

```
#unstructured variance (MAR)
model3$coefficients[8]
```

```
## week.f3:x
##  1.178437
```

```
#independence structure equal variance (MAR)
model4$coefficients[8]
```

```
## week.f3:x
## 0.8234124
```

```
#LMM with random intercept and slope (MAR)
model5$coefficients$fixed[8]
```

```
## week.f3:x
##  1.147043
```

```
#unstructured variance (complete)
model6$coefficients[8]
```

```
## week.f3:x
##  1.081695
```

```
#independence structure equal variance (complete)
model7$coefficients[8]
```

```
## week.f3:x
##  1.081695
```

```
#LMM with random intercept and slope (complete)
model8$coefficients$fixed[8]
```

```
## week.f3:x
##  1.081695
```

```
#LMM with random intercept and slope (MNAR)
model9$coefficients$fixed[8]
```

```
## week.f3:x
##  1.057947
```

```
#estimate from point (b)
model_mean$coefficients[8]
```

```
## week.f3:x
##  1.052453
```

Compare also the variance components of LMM in the three considered models

```
#MAR data
getVarCov(model5)
```

```
## Random effects variance covariance matrix
##             (Intercept)          t
## (Intercept)   0.1969600 -0.0025243
## t            -0.0025243  0.0706360
##   Standard Deviations: 0.4438 0.26577
```

```
#complete data
getVarCov(model8)
```

```
## Random effects variance covariance matrix
##             (Intercept)          t
## (Intercept)   0.2031700 -0.0076999
## t            -0.0076999  0.0735520
##   Standard Deviations: 0.45075 0.2712
```

```
#MNAR data
getVarCov(model9)
```

```
## Random effects variance covariance matrix
##             (Intercept)         t
## (Intercept)    0.213030 -0.015817
## t             -0.015817  0.058545
##   Standard Deviations: 0.46155 0.24196
```

Question (e)

Finally estimate the bias for the fixed effect parameter at time point 4 (interaction term with grp) using unstructured, independence structure equal variance, LMM with random intercept and random slopes to complete dataset and MAR dataset (in total 6 models) by simulation.

```
#true value of the parameter
true_parameter<-3*0.3
#unstructured variance (MAR)
bias_model3<-model3$coefficients[8]-true_parameter
#independence structure equal variance (MAR)
bias_model4<-model4$coefficients[8]-true_parameter
#LMM with random intercept and slope (MAR)
bias_model5<-model5$coefficients$fixed[8]-true_parameter
#unstructured variance (complete)
```

```r
bias_model6<-model6$coefficients[8]-true_parameter
#independence structure equal variance (complete)
bias_model7<-model4$coefficients[8]-true_parameter
#LMM with random intercept and slope (complete)
bias_model8<-model8$coefficients$fixed[8]-true_parameter
print(c(bias_model3,bias_model4,bias_model5,
      bias_model6,bias_model7,bias_model8))
```

```
##   week.f3:x   week.f3:x   week.f3:x   week.f3:x   week.f3:x   week.f3:x
##  0.27843738 -0.07658755  0.24704312  0.18169452 -0.07658755  0.18169452
```