Homework 2

BIOS6643 Fall 2021

Due Tues 10/5/2021 at midnight

Question 1. Principal Component Analysis

Consider the eNO data, and how we applied PCA to the data for graphical purposes (see Graphs slides). Determine the slope of the regression of Post (Y_2) on Pre (Y_1) values (i.e., a standard 'baseline as covariate' model), and compare this to the 'slope' of the $PC1$ axis. Compare the slopes numerically and superimpose the lines on a scatterplot of Post versus Pre values.

In order to do this, recall $PC1 = aY_1 + bY_2$, where a and b are chosen to maximize the variance of $PC1$ (recall $a = 0.51$, $b = 0.86$ for the data; see the slides).

Note: in terms of Y_2 versus Y_1, the 'slope' of the $PC1$ axis is simply b/a; to create a line to graph for $PC1$, you can have it go through the joint sample mean of Y_1 and Y_2. This exercise helps demonstrate the 'regression' principle in a regression line.

A few comments: First, in terms of the graph, $PC1$ is an axis rather than a line, just like Y_1 and Y_2. This is why we need to anchor it through something; it makes sense to have it go through the joint sample means of Y_1 and Y_2, just like the regression line does. This will allows us to determine an intercept for $PC1$ in addition to the slope, which we already know. **See the code below** that walks through the calculations. Note in the graph below I added the 95Note that the slope of the regression line is $(SD_{post}/SD_{pre}) \times r$ and the slope of the $PC1$ line is SD_{post}/SD_{pre}; since r is close to 1, we do not see much difference between the two.

```
eno <- here::here("data", "eno_data.txt") %>%
  read.table(header = T,  sep = " ", skip = 0)

# compute radius
N <- length(eno$eno_pre); n <- 2
f <- qf(0.95, n, N - n)
r <- sqrt((n * (N - 1) * f) / ((N - n) * N))

# covariance matrix
sigma <- mat.or.vec(2, 2)
sigma[1, 2] <- cov(eno$eno_pre, eno$eno_post); sigma[2, 1] <- sigma[1, 2]
sigma[1, 1] <- var(eno$eno_pre); sigma[2, 2] <- var(eno$eno_post)
```

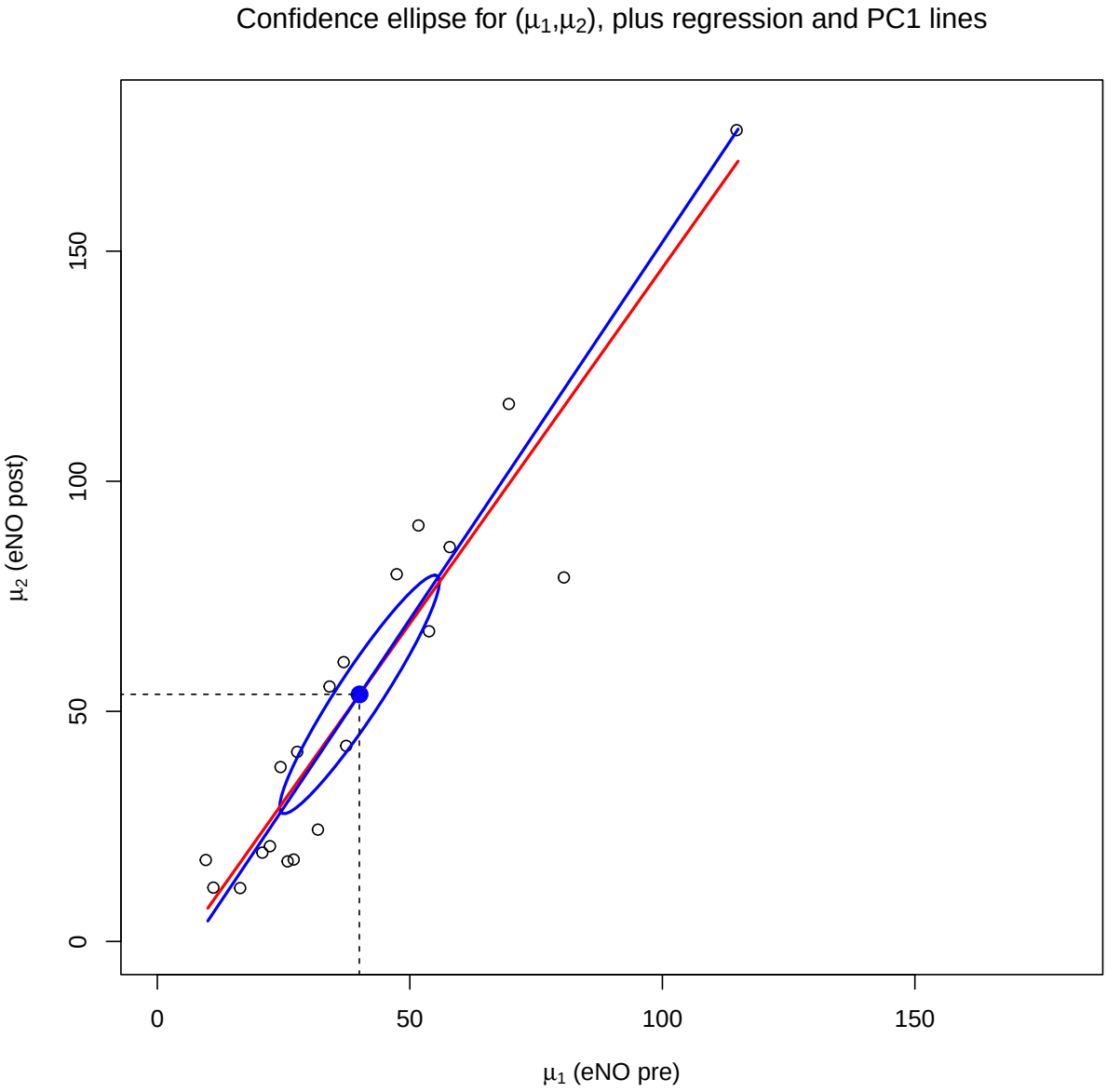
```r
# ellipse center (means)
mny1 <- mean(eno$eno_pre); mny2 <- mean(eno$eno_post)
# plot the data
matplot(eno$eno_pre, eno$eno_post,
  xlim = c(0, 180), ylim = c(0, 180),
  xlab = expression(mu[1] * " (eNO pre)"),
  ylab = expression(mu[2] * " (eNO post)"),
  main = expression("Confidence ellipse for (" * mu[1] * "," *  mu[2] *
                    "), plus regression and PC1 lines"), pch = 1)

# add the ellipse
ellipse(center = c(mny1, mny2), shape = sigma, radius = r)

# indicate marginal sample means
segments(40, -10, 40, 53.7, lty = 2); segments(-10, 53.7, 40, 53.7, lty = 2)

# Other Confidence ellipse info
eig <- eigen(sigma); corr <- cov2cor(sigma)

# Parts to answer the HW question
linreg <- lm(eno$eno_post ~ eno$eno_pre)
x <- c(10:115)
linregy <- -8.230 + 1.546 * x
lines(x, linregy, col = "red", lwd = 2)
slope <- sqrt(sigma[2, 2]) / sqrt(sigma[1, 1])
yint <- mean(eno$eno_post) - mean(eno$eno_pre) * (slope)
pcy <- yint + slope * x
lines(x, pcy, col = "blue", lwd = 2)
```


Confidence ellipse for (μ1,μ2), plus regression and PC1 lines
μ2 (eNO post)
μ1 (eNO pre)
0
50
100
150
0
50
100
150

Question 2. GLM, GzLM, LMM, and likelihood functions, and Variance in LMM

a. In a paragraph, explain the difference between a general linear model (GLM; not a generalized linear model, which I denote with GzLM and which will be discussed more later) and a linear mixed model (LMM).

Basically, a general linear model (GLM) is for independent (e.g., cross-sectional) data, and a linear mixed model (LMM) accounts for correlated data. The GLM is a special case of the LMM when there are no random effects and the error covariance matrix is simple ($\sigma^2\boldsymbol{I}$). Both modeling approaches are regression-type models, where we are trying to understand the relationship between an outcome and several.

b. In a short paragraph, explain the difference between a profiled likelihood and a restricted likelihood for a linear mixed model, and how and why they are used. Which one is a re-expression of the standard likelihood?

A profiled likelihood is a re-expression of the standard likelihood. The common profiled likelihood for a linear mixed model is expressed completely in terms of the covariance parameters. This is accomplished by maximizing the likelihood conditioned on the covariance parameters, and then solving for the fixed effects. This leads to an algebraic form for $\hat{\boldsymbol{\beta}}$, expressed as a function of the covariance parameters. This form can then be substituted back in for $\boldsymbol{\beta}$, so that the likelihood is completely expressed in terms of covariance parameters, but it is intrinsically the same likelihood. The restricted likelihood considers a linear form of the original $\boldsymbol{Y}$ that eliminates the fixed effects completely, so it is a different likelihood. The purpose is to get unbiased (or at least less biased) estimators of covariance parameters. The difficulty is there is no true mechanism to estimate the fixed effect parameters with the restricted likelihood, so what is typically done is that the ML algebraic form for $\hat{\boldsymbol{\beta}}$ is employed.

c. Derive $Var[\hat{\boldsymbol{\beta}}]$ in a full-rank linear mixed model, given the algebraic form of $\hat{\boldsymbol{\beta}}$ that is obtained via ML estimation.

NOTE: there are two types of variance, model-based and empirical (or sandwich estimator). The difference is whether the middle $\boldsymbol{V}$ is determined via the model or using squared residual quantities. To answer question c., work with the 'complete data' form of $\hat{\boldsymbol{\beta}}$.

The ML estimator has form $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^t\boldsymbol{V}^{-1}\boldsymbol{X})^-\boldsymbol{X}^t\boldsymbol{V}^{-1}\boldsymbol{Y}$, which is a linear form of $\boldsymbol{Y}$. Since we are dealing with a model with full rank $\boldsymbol{X}$, then $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^t\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{V}^{-1}\boldsymbol{Y}$. The linear form result says $Var[\boldsymbol{AY}] = \boldsymbol{A}Var[\boldsymbol{Y}]\boldsymbol{A}^t$; so let $\boldsymbol{A} = (\boldsymbol{X}^t\boldsymbol{V}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{V}^{-1}$ and

$$
\begin{aligned}
Var(\hat{\boldsymbol{\beta}}) &= Var((\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{V}^{-1}\mathbf{Y}) && \text{ML estimate for } \boldsymbol{\beta} \\
&= [(\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{V}^{-1}]Var(\mathbf{Y})[(\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{V}^{-1}]^t && Var(\mathbf{AX}) = \mathbf{A}Var(\mathbf{X})\mathbf{A}^t \\
&= [(\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{V}^{-1}]Var(\mathbf{Y})[(\mathbf{V}^{-1})^t(\mathbf{X}^t)^t((\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1})^t] && (\mathbf{AB})^t = \mathbf{B}^t\mathbf{A}^t \\
&= [(\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{V}^{-1}]Var(\mathbf{Y})[\mathbf{V}^{-1}\mathbf{X}(\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1}] && \mathbf{A}^t = \mathbf{A} \text{ symmetric} \\
&= (\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1}(\mathbf{X}^t\mathbf{V}^{-1}\cancel{\mathbf{V}\mathbf{V}^{-1}}\mathbf{X})(\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1} && Var(\mathbf{Y}) = \mathbf{V} \\
&= (\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1}\cancel{(\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})(\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1}} && \mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \\
&= (\mathbf{X}^t\mathbf{V}^{-1}\mathbf{X})^{-1}
\end{aligned}
$$

Question 3. Models for Beta Carotene data

For the Beta Carotene data (see the description of the data and the data itself in another link in the Data module). For parts **a** and **b**, model *time* and *group* as class variables, and include *group* × *time*. In order to account for repeated measures over *time*, specify the *UN* error covariance structure.

a. Conduct a test to compare the 30 and 60mg BASF trends over *time* to see if they differ, i.e., an interaction test, but only involving these 2 *groups*.

"I have no idea what he is talking about; he is answering something different from what he asked in the question."

```
bc <- here::here("data", "beta_carotene_univar.csv") %>%
  read.csv() %>%
  janitor::clean_names()

## set up the control for convergence
## we included so many parameters
ctrl <- lmeControl(niterEM = 2500,
                   # opt="optim",
                   msMaxIter = 2500)


mod0 <- lme(y ~ 0 + factor(time) * factor(prepar),
            ## random intercept
            random = ~1|id,
            ## UN for correlation! no covariance
            correlation = corSymm(form = ~1|id),
            ## for unequal variance over time
            weights = varIdent(form = ~1|time),
            ## convergence setting
            control = ctrl,
            data = bc)
```

```
mod1 <- lme(y ~ 1 + factor(time) * factor(prepar),
            ## random intercept
            random = ~1|id,
            ## UN for correlation! no covariance
            correlation = corSymm(form = ~1|id),
            ## for unequal variance over time
            weights = varIdent(form = ~1|time),
            ## convergence setting
            control = ctrl,
            data = bc)
```

no need for homework but useful

check whether the covariance matrix is correct.

```
## fixed effects
beta_hat <- fixed.effects(mod1)
## fixed effects variance-covariance
C <- vcov(mod1) %>% round(digits = 2)

##             [1]  [2] [3] [4]  [5]  [6] [7] [8] [9]   [10]  [11]   [12]   [13]  [14]
## coefs       Int  t6  t8  t10  t12  p2  p3  p4  t6:p2 t8:p2 t10:p2 t12:p2 t6:p3 t8:p
p3_p4_t0 <- c( 0,   0,  0,  0,   0,   0,  1,  -1,  0,    0,   0,     0,     0,    0,
## p3_t6 <- c( 1,   1,  0,  0,   0,   0,  1,  0,   0,    0,   0,     0,     1,    0,
## p4_t6 <- c( 1,   1,  0,  0,   0,   0,  0,  1,   0,    0,   0,     0,     0,    0,
## p3_p4_t6 <- p3_t6 - p4_t6, similarly for other time points
p3_p4_t6 <- c( 0,   0,  0,  0,   0,   0,  1,  -1,  0,    0,   0,     0,     1,    0,
p3_p4_t8 <- c( 0,   0,  0,  0,   0,   0,  1,  -1,  0,    0,   0,     0,     0,    1,
p3_p4_t10 <- c(0,   0,  0,  0,   0,   0,  1,  -1,  0,    0,   0,     0,     0,    0,
p3_p4_t12 <- c(0,   0,  0,  0,   0,   0,  1,  -1,  0,    0,   0,     0,     0,    0,

contr1 <- cbind(p3_p4_t0, p3_p4_t6, p3_p4_t8, p3_p4_t10, p3_p4_t12)
rownames(contr1) <- rownames(C)

p34_t6_0 <- p3_p4_t6 - p3_p4_t0
p34_t8_0 <- p3_p4_t8 - p3_p4_t0
p34_t10_0 <- p3_p4_t10 - p3_p4_t0
p34_t12_0 <- p3_p4_t12 - p3_p4_t0

contr0 <- cbind(p34_t6_0, p34_t8_0, p34_t10_0, p34_t12_0)

## contrast point estimates to be
(ce0 <- t(contr0) %*% beta_hat)
```

```
##                 [,1]
```

```
## p34_t6_0  -18.06667
## p34_t8_0  -51.86667
## p34_t10_0  22.20000
## p34_t12_0  48.80000
```

```
## contrast variance covariance matrix
cov0 <- t(contr0) %*% C %*% contr0

## with both point estimates and standard deviation
## an anova or pairwise comparison can be performed
W0 <- t(ce0) %*% solve(cov0) %*% ce0
pchisq(W0, df = 4, lower.tail = FALSE)
```

```
##             [,1]
## [1,] 0.009650473
```

```
## contrast point estimates to be
(ce1 <- t(contr1) %*% beta_hat)
```

```
##                 [,1]
## p3_p4_t0    7.533333
## p3_p4_t6  -10.533333
## p3_p4_t8  -44.333333
## p3_p4_t10  29.733333
## p3_p4_t12  56.333333
```

```
## contrast variance covariance matrix
cov1 <- t(contr1) %*% C %*% contr1
## contrast standard deviation
(se1 <- sqrt(diag(cov1)))
```

```
##  p3_p4_t0  p3_p4_t6  p3_p4_t8 p3_p4_t10 p3_p4_t12
##  32.18291  72.06088  72.95039  64.64070  67.84210
```

```
## with both point estimates and standard deviation
## an anova or pairwise comparison can be performed
W1 <- t(ce1) %*% solve(cov1) %*% ce1
pchisq(W1, df = 5, lower.tail = FALSE)
```

```
##             [,1]
## [1,] 0.002565421
```

```
comp1 <- multcomp::glht(mod1, t(contr1))
summary(comp1)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lme.formula(fixed = y ~ 1 + factor(time) * factor(prepar), data = bc,
```

```
##     random = ~1 | id, correlation = corSymm(form = ~1 | id),
##     weights = varIdent(form = ~1 | time), control = ctrl)
##
## Linear Hypotheses:
##                Estimate Std. Error z value Pr(>|z|)
## p3_p4_t0 == 0     7.533     32.183   0.234    0.992
## p3_p4_t6 == 0   -10.533     72.061  -0.146    0.999
## p3_p4_t8 == 0   -44.333     72.951  -0.608    0.830
## p3_p4_t10 == 0   29.733     64.641   0.460    0.918
## p3_p4_t12 == 0   56.333     67.842   0.830    0.675
## (Adjusted p values reported -- single-step method)
```

```
# comp0 <- multcomp::glht(mod1, test = "Chisqtest", t(contr1))
# summary(comp0)
```

```
## emmeans is a package cover
emm1 <- emmeans::emmeans(
  mod1, ## the first arg is the object
  ## specs arg is in formula status
  specs =  ~ time:prepar,
  ## only test given time and group
  at = list(time = c(0, 6, 8, 10, 12),
            prepar = c(3, 4)),
  digits = 2)

(con1 <- contrast(emm1, interaction = c(time = "consec")))
```

```
##  time_consec prepar_consec estimate   SE df t.ratio p.value
##  6 - 0       4 - 3             18.1 56.7 76   0.319  0.7507
##  8 - 6       4 - 3             33.8 28.7 76   1.178  0.2425
##  10 - 8      4 - 3            -74.1 30.4 76  -2.439  0.0171
##  12 - 10     4 - 3            -26.6 22.0 76  -1.208  0.2307
##
## Degrees-of-freedom method: containment
```

```
test(contrast(emm1), join = TRUE)
```

```
##  df1 df2 F.ratio p.value note
##    9  19   7.335  0.0001  d
##
## d: df1 reduced due to linear dependence
```

b. Conduct a test to compare to see if the 12 week - baseline value differs between the 4 *groups*.

```
##             [1]  [2] [3] [4]  [5]  [6] [7] [8] [9]   [10]  [11]   [12]  [13]  [14]
## coefs       Int  t6  t8  t10  t12  p2  p3  p4  t6:p2 t8:p2 t10:p2 t12:p2 t6:p3 t8:p
```

```
# p1_t12 <- c( 1,   0,  0,  0,   1,   0,  0,  0,  0,    0,    0,     0,     0,    0,
# p1_t0  <- c( 1,   0,  0,  0,   0,   0,  0,  0,  0,    0,    0,     0,     0,    0,
p1_t12_0 <- c( 0,   0,  0,  0,   1,   0,  0,  0,  0,    0,    0,     0,     0,    0,
p2_t12_0 <- c( 0,   0,  0,  0,   1,   0,  0,  0,  0,    0,    0,     1,     0,    0,
p3_t12_0 <- c( 0,   0,  0,  0,   1,   0,  0,  0,  0,    0,    0,     0,     0,    0,
p4_t12_0 <- c( 0,   0,  0,  0,   1,   0,  0,  0,  0,    0,    0,     0,     0,    0,

contr2 <- cbind(p1_t12_0, p2_t12_0, p3_t12_0, p4_t12_0)
rownames(contr2) <- rownames(C)

## contrast point estimates to be
(ce2 <- t(contr2) %*% beta_hat)
```

```
##               [,1]
## p1_t12_0 124.33333
## p2_t12_0  66.33333
## p3_t12_0 196.80000
## p4_t12_0 148.00000
```

```
## contrast variance covariance matrix
cov2 <- t(contr2) %*% C %*% contr2

## with both point estimates and standard deviation
## an anova or pairwise comparison can be performed
W2 <- t(ce2) %*% solve(cov2) %*% ce2
pchisq(W2, df = 4, lower.tail = FALSE)
```

```
##              [,1]
## [1,] 1.891552e-13
```

```
emm2 <- emmeans(## the first arg is the object
                mod1,
                ## specs arg is in formula status
                specs = ~ time:prepar,
                at = list(time = c(0, 12)))
(con2 <- contrast(emm2, interaction = c(time = "pairwise", prepar = "consec")))
```

```
##  time_pairwise prepar_consec estimate   SE df t.ratio p.value
##  0 - 12        2 - 1            58.0 47.5 76   1.221  0.2258
##  0 - 12        3 - 2          -130.5 49.8 76  -2.619  0.0106
##  0 - 12        4 - 3            48.8 49.8 76   0.980  0.3304
##
## Degrees-of-freedom method: containment
```

```
coef(con2)
```

```
##   time prepar c.1 c.2 c.3
```

```
## 1    0      1  -1   0   0
## 2   12      1   1   0   0
## 3    0      2   1  -1   0
## 4   12      2  -1   1   0
## 5    0      3   0   1  -1
## 6   12      3   0  -1   1
## 7    0      4   0   0   1
## 8   12      4   0   0  -1
```

```
## "type III" tests of interaction effects
## can be obtained via interaction contrast
test(con2, joint = TRUE)
```

```
##  df1 df2 F.ratio p.value
##    3  76   2.400  0.0743
```

```
joint_tests(mod1)
```

```
##  model term  df1 df2 F.ratio p.value
##  time         4  76  15.999  <.0001
##  prepar       3  19   1.524  0.2405
##  time:prepar 12  76   2.229  0.0181
```

```
joint_tests(mod1, by = "prepar")
```

```
## prepar = 1:
##  model term df1 df2 F.ratio p.value
##  time        4  76   5.136  0.0010
##
## prepar = 2:
##  model term df1 df2 F.ratio p.value
##  time        4  76   1.872  0.1240
##
## prepar = 3:
##  model term df1 df2 F.ratio p.value
##  time        4  76   8.835  <.0001
##
## prepar = 4:
##  model term df1 df2 F.ratio p.value
##  time        4  76   6.426  0.0002
```

I thought this should be a pairwised test but I just do not know what Matt was doing. There are four groups why only three contrasts

c. Consider the model that uses *time* as continuous, with up to cubic effects, plus interactions between group and time (up to cubic). How does this model compare with the one that uses *time* as class (plus interactions)? Discuss in a paragraph.

```
mod2 <- lme(y ~ time * factor(prepar) +
              I(time^2) * factor(prepar) +
              I(time^3) * factor(prepar),
            random = ~1|id,
            correlation = corSymm(form = ~1|id),
            weights = varIdent(form = ~1|time),
            control = ctrl,
            data = bc)

anova(mod2)
```

```
##                            numDF denDF   F-value p-value
## (Intercept)                    1    80 216.57796  <.0001
## time                           1    80  41.48904  <.0001
## factor(prepar)                 3    19   2.94104  0.0594
## I(time^2)                      1    80  13.36855  0.0005
## I(time^3)                      1    80   7.39342  0.0080
## time:factor(prepar)            3    80   3.65439  0.0159
## factor(prepar):I(time^2)       3    80   2.06054  0.1120
## factor(prepar):I(time^3)       3    80   1.42034  0.2430
```

```
AIC(mod1, mod2)
```

```
## Warning in AIC.default(mod1, mod2): models are not all fitted to the same number
## of observations
```

```
##      df      AIC
## mod1 36 1096.003
## mod2 32 1208.053
```

his answer does not make any sense, there is even no model fitting

It is actually the same model fit! This is because when we use up to cubic effects and add the interactions, we have saturated the fixed-effects part of the model. In other words, we have made the model as flexible as possible in terms of how we model group and time. However, with the time-as-continuous approach you obviously can do some things, like estimate effects at times in between those observed (i.e., interpolate). With the time as class approach, you can still estimate linear, quadratic and cubic trends by including polynomial contrasts.

d. Modeling the data using $Time0$ as a covariate value, with the remaining *times* as repeated measures on the outcome (6, 8, 10, 12 weeks). What are pros and cons of this approach, relative to using all measures as outcome values in a longitudinal model? In particular, focuses on the modeling of the repeated measures, how fixed effects need to be specified, and impact of modeling of *time* as class versus continuous.

```
bc2 <- bc %>%
  pivot_wider(names_from = time,
              values_from = y) %>%
```

```
  pivot_longer(cols = 4:7,
               names_to = "time",
               values_to = "y") %>%
  rename("baseline" = "0") %>%
  mutate(time = as.integer(time))


mod3 <- lme(y ~ baseline + time * factor(prepar),
            ## random intercept
            random = ~1|id,
            ## UN for correlation! no covariance
            correlation = corSymm(form = ~1|id),
            ## for unequal variance over time
            weights = varIdent(form = ~1|time),
            ## convergence setting
            control = ctrl,
            data = bc2)

mod4 <- lme(y ~ baseline + factor(time) * factor(prepar),
            ## random intercept
            random = ~1|id,
            ## UN for correlation! no covariance
            correlation = corSymm(form = ~1|id),
            ## for unequal variance over time
            weights = varIdent(form = ~1|time),
            ## convergence setting
            control = ctrl,
            data = bc2)
```

I just totally lost now. do we still need to care about the UN structure and random intercept? Where is the model fitting.

e. For the model in part **d**, estimate the linear, quadratic and cubic trends for the model that uses *time* as a class variable.

```
emm4_poly <- emmeans(mod4, ~factor(time))
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
contrast(emm4_poly, 'poly')
```

```
##  contrast  estimate   SE df t.ratio p.value
##  linear      32.667 32.8 57   0.995  0.3240
##  quadratic   17.933 12.3 57   1.462  0.1491
##  cubic        0.833 31.0 57   0.027  0.9786
##
```

```
## Results are averaged over the levels of: prepar
## Degrees-of-freedom method: containment
```

```
# emm4_poly_g1 <- emmeans(mod4, ~factor(time):factor(prepar), at = list(prepar = c(1))
# contrast(emm4_poly_g1, 'poly')
# emm4_poly_g2 <- emmeans(mod4, ~factor(time):factor(prepar), at = list(prepar = c(2))
# contrast(emm4_poly_g2, 'poly')

# emm1_poly <- emmeans(mod1, ~time)
# contrast(emm1_poly, 'poly')

# cpoly <- t(matrix(c(-3, 1, 1, 3,
#                     1, -1, -1, 1,
#                     -1, 3, -3, 1),
#                   nrow = 4,
#                   ncol = 3))
# rownames(cpoly) <- c("linear", "quadratic", "cubic")
# con4 <- multcomp::glht(mod4, cpoly)
```

I did not see any connection between his code and his output

Question 4. Constrasts

Consider a study where *subjects* in 3 *groups* (e.g., race or treatment) are observed over 3 equally spaced *times* and some health outcome, y, is measured. Unless otherwise mentioned, include a random intercept for subjects to account for the repeated measures. For simplicity, use 2 *subjects* per *group*.

a. Consider modeling *group* and *time* as class variables, plus interaction. Write statistical models and the $\boldsymbol{X}$ matrix for the following cases.

b. No restriction placed on the model. i.e., write the less-than-full-rank statistical model.

$$
\begin{aligned}
&Y_{grp=g,sub=i,time=t} = \mu_0 + \alpha_g + \tau_t + \gamma_{g\times t} + b_i + \epsilon_{g,i,t} \\
&b_i \overset{iid}{\sim} \mathcal{N}(0, \sigma_b^2) \\
&\epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)
\end{aligned}
$$

```
## setup dataset
group <- rep(c("A", "B", "C"), each = 6)
time <- rep(c("1", "2", "3"))
id <- rep(1:6, each = 3)
y <- "NA"
data_s <- cbind(id, group, time, y) %>% as.data.frame()

form1 <- formula(y ~ I(group == "A") + I(group == "B") + I(group == "C") +
```

```
                    I(time == "1") + I(time == "2") + I(time == "t3") +
                    group:time)
mod_f1 <- model.frame(form1, # Formula
                      # Data frame
                      data = data_s,
                      # Identifier of data records
                      SubjectId = id)
Xmtx1 <- model.matrix(form1, mod_f1)
colnames(Xmtx1) <- NULL; Xmtx1
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## 1     1    1    0    0    1    0    0    1    0     0     0     0     0     0
## 2     1    1    0    0    0    1    0    0    0     0     1     0     0     0
## 3     1    1    0    0    0    0    0    0    0     0     0     0     0     1
## 4     1    1    0    0    1    0    0    1    0     0     0     0     0     0
## 5     1    1    0    0    0    1    0    0    0     0     1     0     0     0
## 6     1    1    0    0    0    0    0    0    0     0     0     0     0     1
## 7     1    0    1    0    1    0    0    0    1     0     0     0     0     0
## 8     1    0    1    0    0    1    0    0    0     0     0     1     0     0
## 9     1    0    1    0    0    0    0    0    0     0     0     0     0     0
## 10    1    0    1    0    1    0    0    0    1     0     0     0     0     0
## 11    1    0    1    0    0    1    0    0    0     0     0     1     0     0
## 12    1    0    1    0    0    0    0    0    0     0     0     0     0     0
## 13    1    0    0    1    1    0    0    0    0     1     0     0     0     0
## 14    1    0    0    1    0    1    0    0    0     0     0     0     1     0
## 15    1    0    0    1    0    0    0    0    0     0     0     0     0     0
## 16    1    0    0    1    1    0    0    0    0     1     0     0     0     0
## 17    1    0    0    1    0    1    0    0    0     0     0     0     1     0
## 18    1    0    0    1    0    0    0    0    0     0     0     0     0     0
##    [,15] [,16]
## 1      0     0
## 2      0     0
## 3      0     0
## 4      0     0
## 5      0     0
## 6      0     0
## 7      0     0
## 8      0     0
## 9      1     0
## 10     0     0
## 11     0     0
## 12     1     0
## 13     0     0
## 14     0     0
```

```
## 15     0     1
## 16     0     0
## 17     0     0
## 18     0     1
## attr(,"assign")
##  [1] 0 1 2 3 4 5 6 7 7 7 7 7 7 7 7 7
## attr(,"contrasts")
## attr(,"contrasts")$`I(group == "A")`
## [1] "contr.treatment"
##
## attr(,"contrasts")$`I(group == "B")`
## [1] "contr.treatment"
##
## attr(,"contrasts")$`I(group == "C")`
## [1] "contr.treatment"
##
## attr(,"contrasts")$`I(time == "1")`
## [1] "contr.treatment"
##
## attr(,"contrasts")$`I(time == "2")`
## [1] "contr.treatment"
##
## attr(,"contrasts")$`I(time == "t3")`
## [1] "contr.treatment"
##
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
## attr(,"contrasts")$time
## [1] "contr.treatment"
```

there are multiple way for setting the less than full rank model, and his model equation is fatally wrong.

ii. A set-to-0 restriction is placed on the parameters associated with highest levels.

```
form2 <- formula(y ~ 1 + group + time + group:time)
mod_f2 <- model.frame(form2, # Formula
                      # Data frame
                      data = data_s,
                      # Identifier of data records
                      SubjectId = id)
Xmtx2 <- model.matrix(form2, mod_f2,
                      contrasts.arg = list(group = "contr.SAS",
                                           time = "contr.SAS"))
colnames(Xmtx2) <- NULL; Xmtx2
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## 1     1    1    0    1    0    1    0    0    0
## 2     1    1    0    0    1    0    0    1    0
## 3     1    1    0    0    0    0    0    0    0
## 4     1    1    0    1    0    1    0    0    0
## 5     1    1    0    0    1    0    0    1    0
## 6     1    1    0    0    0    0    0    0    0
## 7     1    0    1    1    0    0    1    0    0
## 8     1    0    1    0    1    0    0    0    1
## 9     1    0    1    0    0    0    0    0    0
## 10    1    0    1    1    0    0    1    0    0
## 11    1    0    1    0    1    0    0    0    1
## 12    1    0    1    0    0    0    0    0    0
## 13    1    0    0    1    0    0    0    0    0
## 14    1    0    0    0    1    0    0    0    0
## 15    1    0    0    0    0    0    0    0    0
## 16    1    0    0    1    0    0    0    0    0
## 17    1    0    0    0    1    0    0    0    0
## 18    1    0    0    0    0    0    0    0    0
## attr(,"assign")
## [1] 0 1 1 2 2 3 3 3 3
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.SAS"
##
## attr(,"contrasts")$time
## [1] "contr.SAS"
```

b. Show that the linear trend for one *group* compared to another (say *GroupA* versus *GroupB*) is estimable by showing that $\boldsymbol{L} = \boldsymbol{LH}$, where the Moore-Penrose inverse is used in calculating $\boldsymbol{H}$. First you need to construct $\boldsymbol{L}$. (As a check, you can repeat using SAS's g-inverse in calculating $\boldsymbol{H}$, but you don't need to turn that in.)

```
##     [1]   [2]  [3]   [4]   [5]  [6]  [7]   [8]  [9]  [10]  [11]  [12]  [13]  [14]
L1 <- c(0,    0,   0,    0,    0,   0,   0,   -1,   0,   1,    1,    0,   -1,    0,    0
XtX1 <- t(Xmtx1) %*% Xmtx1
H1 <- MASS::ginv(XtX1) %*% XtX1
round(L1 %*% H1)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## [1,]    0    0    0    0    0    0    0   -1    0     1     1     0    -1     0
##      [,15] [,16]
## [1,]     0     0
```

```
##     [1]   [2]  [3]   [4]   [5]  [6]  [7]   [8]  [9]
L2 <- c(1,    0,   -1,    0,    1,   0,   0,   -1,   0)
```

```
XtX2 <- t(Xmtx2) %*% Xmtx2
H2 <- solve(XtX2) %*% XtX2
round(L2 %*% H2)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    1    0   -1    0    1    0    0   -1    0
```

c. How would answers in a change in part **a** if an AR(1) structure for $\boldsymbol{R}$ is included? (You do not need to rewrite entire models, just mention what changes).

d. Say that *Time* is treated as continuous (i.e., not included in the CLASS statement in SAS or factor argument in R). Rewrite either the full-rank or less-than-full-rank model (clearly specify which one) and $\boldsymbol{X}$ matrices in **a**. Say the linear term for *Time* is sufficient.

```
form3 <- formula(y ~ 1 + group + as.integer(time) + group:as.integer(time))
mod_f3 <- model.frame(form3, # Formula
                      # Data frame
                      data = data_s,
                      # Identifier of data records
                      SubjectId = id)
Xmtx3 <- model.matrix(form3, mod_f3,
                      contrasts.arg = list(group = "contr.SAS"))
colnames(Xmtx3) <- NULL; Xmtx3
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6]
## 1     1    1    0    1    1    0
## 2     1    1    0    2    2    0
## 3     1    1    0    3    3    0
## 4     1    1    0    1    1    0
## 5     1    1    0    2    2    0
## 6     1    1    0    3    3    0
## 7     1    0    1    1    0    1
## 8     1    0    1    2    0    2
## 9     1    0    1    3    0    3
## 10    1    0    1    1    0    1
## 11    1    0    1    2    0    2
## 12    1    0    1    3    0    3
## 13    1    0    0    1    0    0
## 14    1    0    0    2    0    0
## 15    1    0    0    3    0    0
## 16    1    0    0    1    0    0
## 17    1    0    0    2    0    0
## 18    1    0    0    3    0    0
## attr(,"assign")
## [1] 0 1 1 2 3 3
```

```
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.SAS"
```

e. Say that the times of observation were at 0, 1 and 6 months rather than equally spaced.

f. Would it be appropriate to treat *Time* as a class variable in this case? Explain.

There is no problem in using equally spaced or unequally spaced times for a class variable, since you are estimating levels separately. The unequal spacing does not impose any constraints metrically.Note: for this question I was considering interpretation of the fixed effects. If you are thinking about implications for the covariance structure, just clearly state that in your argument. For example, if you use the standard AR(1) structure, it would not work well with unequally spaced time points.

ii. Suggest a structure for $\boldsymbol{R}_1$ and write it out.

lmm models in `R` programming is defined in a different way. in R we do not define a R covariance matrix, we define a R correlation matrix under equal-variance assumption; if there is a violation, we use `weights` arguement to adjust variances. Then a R covariance matrix will be build with correlation and variances.

So in `R`, the SAS UN structure in fact is a Symmetric correlation matrix (with 3 parameters) and a variance vector (with 3 parameter, more precisely one variance, and two correlation parameters)