Homework 2 - Solution

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.

ANSWER A few comments: First, in terms of the graph, $PC1$ is an direction/axis rather than a line. 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 95% confidence ellipse for the joint mean (like a confidence interval but generalizing to 2 dimensions). You only need to plot the 2 lines on the scatterplot for full credit (blue = PC1 'line', red = regression line). In this case there is not much 'regression' in the regression line.

Note 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)

fit1 <-lm(eno_post ~ eno_pre, data = eno)
coef(fit1)
```

```
## (Intercept)     eno_pre
##   -8.229517    1.546124
```

```
## 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)

## 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)"),
  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)
```

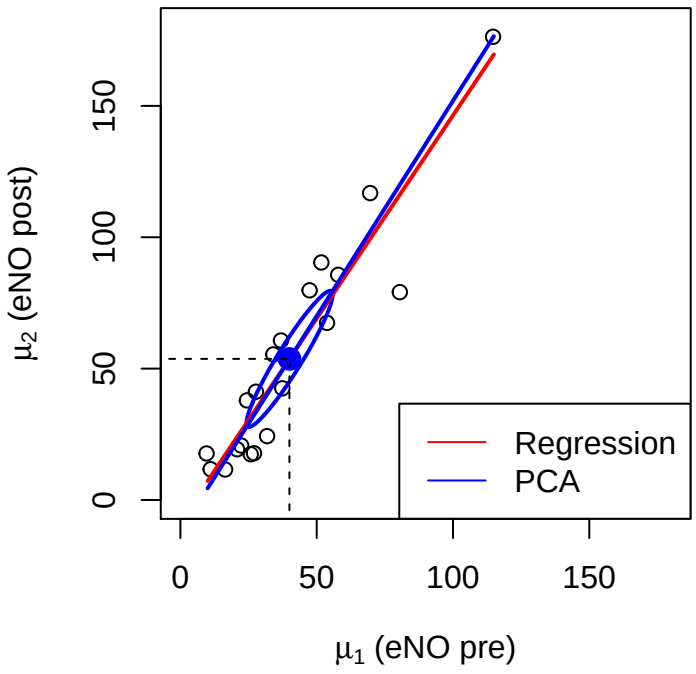
```r
## CODE to answer the HW question
linreg <- lm(eno$eno_post ~ eno$eno_pre)
x <- c(10:115)
linregy <- coef(fit1)[1] + coef(fit1)[2] * x
lines(x, linregy, col = "red", lwd = 2)

## slope is the b term in the question
(slope <- sqrt(sigma[2, 2]) / sqrt(sigma[1, 1]))
```

```
## [1] 1.638786
```

```r
## plug-in the b term to get intercept a
(yint <- mean(eno$eno_post) - mean(eno$eno_pre) * (slope))
```

```
## [1] -11.9402
```

```r
pcy <- yint + slope * x
lines(x, pcy, col = "blue", lwd = 2)
legend("bottomright", lty = 1,
       col = c("red", "blue"),
       legend =c("Regression", "PCA"))
```



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).

A general linear model (GLM) is a regression model for independent (e.g., cross-sectional or one-way ANOVA) data, and a linear mixed model (LMM) accounts for correlated data.

When there are violation on certain assumptions, such as independence or equal-variance assumption, GLM is not reasonable to be used directly. LMM is a powerful tool, allowing us to model correlation between responses through the use of sophisticated terms: random effect $\boldsymbol{b}$ and the covariance matrix of repeated residual $\boldsymbol{R}$ matrices. The GLM is a special case of the LMM when there are no random effects and the error covariance matrix is simple ($\sigma^2\boldsymbol{I}$).

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?

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. (We can profile out either fixed effect β or covariance $V(\alpha)$, in either way, the final MLE estimators are the same. In lecture we conditioned on β first, and express β with $V(\alpha)$. Then calculate the mle for $V(\alpha)$. Finally plug-in $\hat{V}(\alpha)$ to get $\hat{\beta}$).

The restricted likelihood considers a linear form of the original $\boldsymbol{Y}$ that eliminates the fixed effects completely (think about it as a contrast), so it is a different likelihood (a likelihood for residuals). 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.

A profiled likelihood is a re-expression of the standard likelihood.

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

\textcolor{blue}{We have provided the full answer to this question in the SAS complementary file BIOS6643_HW2Q3_sol_sas.pdf. However, here we outline a solution using R. Note that some results are consistent with the SAS outputs. However, the tests for contrasts are based on F test in SAS versus wald-type tests in R, so they will not necessarily provide exactly the same results. Note that if you used R to answer this question, you should explain your results accordingly.}

```
## import data beta
beta <- read.csv("/Users/juarezce/Documents/OneDrive - The University of Colorado Denver/BIOS6643/BIOS6643_Notes/data/beta_carotene_univar.csv",
                 header=TRUE)
head(beta,3)
```

```
##   Prepar Id   y time
## 1      1 71 116    0
## 2      1 71 174    6
## 3      1 71 178    8
```

```
names(beta) <- c("prepar", "id", "y", "time")
#  mutate(time = as.integer(time))

## set up the control for convergence
## we included so many parameters
ctrl <- lmeControl(niterEM = 1000,
                   # opt="optim",
                   msMaxIter = 1000)


mod1 <- gls(y ~ 1 + factor(time) * factor(prepar),
            ## UN for correlation! not covariance
            correlation = corSymm(form = ~1|id),
            ## for unequal variance over time
            weights = varIdent(form = ~1|time),
            ## convergence setting
            control = ctrl,
            method = "REML",
            data = beta)
```

Please see the Rmd file for extra information about model fitting.

We outline two methods to get the contrast test in Question 3a.

1. Manually setup

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

## check the Rmd file for a better version
##             [1]  [2] [3] [4]  [5]  [6] [7] [8] [9]   [10]  [11]   [12]   [13]  [14]  [15]   [16]   [17]  [18]  [19]   [20]
## coefs       Int  t6  t8  t10  t12  p2  p3  p4  t6:p2 t8:p2 t10:p2 t12:p2 t6:p3 t8:p3 t10:p3 t12:p3 t6:p4 t8:p4 t10:p4 t12:p4
p3_p4_t0 <- c( 0,   0,  0,  0,   0,   0,  1,  -1,  0,    0,    0,     0,     0,    0,    0,     0,     0,    0,    0,     0)
## expected means for prepar=3 - prepar=4 at time=6
## p3_t6 <- c( 1,   1,  0,  0,   0,   0,  1,  0,   0,    0,    0,     0,     1,    0,    0,     0,     0,    0,    0,     0)
## p4_t6 <- c( 1,   1,  0,  0,   0,   0,  0,  1,   0,    0,    0,     0,     0,    0,    0,     0,     1,    0,    0,     0)
## difference of means b/w prepar=3 - prepar=4 at time=6
## 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,    0,     0,     -1,   0,    0,     0)
p3_p4_t8 <- c( 0,   0,  0,  0,   0,   0,  1,  -1,  0,    0,    0,     0,     0,    1,    0,     0,     0,    -1,   0,     0)
p3_p4_t10 <- c(0,   0,  0,  0,   0,   0,  1,  -1,  0,    0,    0,     0,     0,    0,    1,     0,     0,    0,    -1,    0)
p3_p4_t12 <- c(0,   0,  0,  0,   0,   0,  1,  -1,  0,    0,    0,     0,     0,    0,    0,     1,     0,    0,    0,     -1)

## build the matrix for the contrast
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)

t(contr0)
```

```
##           [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## p34_t6_0     0    0    0    0    0    0    0    0    0     0     0     0     1
## p34_t8_0     0    0    0    0    0    0    0    0    0     0     0     0     0
## p34_t10_0    0    0    0    0    0    0    0    0    0     0     0     0     0
## p34_t12_0    0    0    0    0    0    0    0    0    0     0     0     0     0
##           [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## p34_t6_0      0     0     0    -1     0     0     0
## p34_t8_0      1     0     0     0    -1     0     0
## p34_t10_0     0     1     0     0     0    -1     0
## p34_t12_0     0     0     1     0     0     0    -1
```

```
## contrast point estimates to be
ce0 <- t(contr0) %*% beta_hat
## 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,] 0.00965162
```

```
### The matrix contrast may be set up direclty too
## coefs       Int       time         | prepar      |       prepar=2        |           prepar=3      |      prepar=4       |
## coefs       Int   t6  t8  t10  t12  p2  p3  p4   t6:p2 t8:p2 t10:p2 t12:p2 t6:p3 t8:p3 t10:p3 t12:p3 t6:p4 t8:p4 t10:p4 t12:p4
c.w <- matrix(c( 0,   0,  0,  0,   0,   0,  0,  0,  0,    0,   0,     0,     1,    0,    0,     0,     -1,   0,    0,    0,
                 0,   0,  0,  0,   0,   0,  0,  0,  0,    0,   0,     0,     0,    1,    0,     0,     0,   -1,    0,    0,
                 0,   0,  0,  0,   0,   0,  0,  0,  0,    0,   0,     0,     0,    0,    1,     0,     0,    0,   -1,    0,
                 0,   0,  0,  0,   0,   0,  0,  0,  0,    0,   0,     0,     0,    0,    0,     1,     0,    0,    0,   -1),
               byrow=T, 4, 20)
c.wa <- c.w%*%beta_hat
## chisquare test value
chisq.v <- t(c.wa)%*%solve(c.w%*% vcov(mod1)  %*%t(c.w)) %*% c.wa
pval.w <- pchisq(chisq.v, 4, ncp=0, lower.tail=FALSE, log.p=FALSE)
pval.w
```

```
##             [,1]
## [1,] 0.009650807
```

2. ‘multcomp::glht()‘

```
## emmeans is a package cover
test1 <- multcomp::glht(mod1, t(contr0))
summary(test1, test = Chisqtest())
```

```
##
##   General Linear Hypotheses
##
## Linear Hypotheses:
##                 Estimate
## p34_t6_0 == 0    -18.07
## p34_t8_0 == 0    -51.87
## p34_t10_0 == 0    22.20
## p34_t12_0 == 0    48.80
##
## Global Test:
##   Chisq DF Pr(>Chisq)
## 1 13.36  4   0.009651
```

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]  [15]   [16]   [17]  [18]  [19]   [20]
## coefs      Int  t6  t8  t10  t12  p2  p3  p4  t6:p2 t8:p2 t10:p2 t12:p2 t6:p3 t8:p3 t10:p3 t12:p3 t6:p4 t8:p4 t10:p4 t12:p4
# p1_t12 <- c( 1,   0,  0,  0,   1,   0,  0,  0,  0,    0,    0,     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,    0,     0,     0,    0,    0,     0)
## expected mean differences for each group between time=12 and time=0
p1_t12_0 <- c( 0,   0,  0,  0,   1,   0,  0,  0,  0,    0,    0,     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,    0,     0,     0,    0,    0,     0)
p3_t12_0 <- c( 0,   0,  0,  0,   1,   0,  0,  0,  0,    0,    0,     0,     0,    0,    0,     1,     0,    0,    0,     0)
p4_t12_0 <- c( 0,   0,  0,  0,   1,   0,  0,  0,  0,    0,    0,     0,     0,    0,    0,     0,     0,    0,    0,     1)

contr2 <- cbind(p2_t12_0 - p1_t12_0,
                p3_t12_0 - p1_t12_0,
                p4_t12_0 - p1_t12_0)
## contrast point estimates to be
(ce2 <- t(contr2) %*% beta_hat)
```

```
##           [,1]
## [1,] -58.00000
## [2,]  72.46667
## [3,]  23.66667
```

```
## 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 = 3, lower.tail = FALSE)
```

```
##            [,1]
## [1,] 0.06575128
```

```
## emmeans is a package cover
test2 <- multcomp::glht(mod1, t(contr2))
summary(test2, test = Chisqtest())
```

```
##
##   General Linear Hypotheses
##
## Linear Hypotheses:
##        Estimate
## 1 == 0   -58.00
## 2 == 0    72.47
## 3 == 0    23.67
##
## Global Test:
##   Chisq DF Pr(>Chisq)
## 1 7.201  3    0.06575
```

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.

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

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

## formula and model.frame are important object
## for lme and gls model fitting.
form1 <- formula(y ~ I(group == "A") + I(group == "B") + I(group == "C") +
                   I(time == "1") + I(time == "2") + I(time == "3") +
                   time:group)
mod_f1 <- model.frame(form1, # Formula
                      # Data frame
                      data = data_s,
                      # Identifier of data records
                      SubjectId = id)
## calling design matrix
```

```
Xmtx1 <- model.matrix(form1, mod_f1)
colnames(Xmtx1) <- NULL; kable(Xmtx1, "simple")
```

1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0
1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0
1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0
1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0
1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1

There are multiple ways to set up a LTFR model. You could also set up the matrix "by hand" writing each column out.

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

$$Y_{ij} = \beta_0 + \beta_1 G_1 + \beta_2 G_2 + \beta_3 t_1 + \beta_4 t_2 + \beta_5 G_1 t_1 + \beta_6 G_1 t_2 + \beta_7 G_2 t_1 + \beta_8 G_2 t_2 + b_i + \epsilon_{ij}$$
$$b_i \overset{iid}{\sim} N(0, \sigma_b^2)$$
$$\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$$
$$b_i \perp \epsilon_{ij}$$

where G_g is an indicator for gorup g, i.e. $G_g = I(group = g)$, $g = 1, 2$. Time indicators are similarly defined as $t_j = I(time = j)$, $j = 1, 2$.

```
form2 <- formula(y ~ 1 + group + time + time:group)
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",
                      ## SAS uses the highest as reference group
                                           time = "contr.SAS"))
colnames(Xmtx2) <- NULL; kable(Xmtx2, "simple")
```

1	1	0	1	0	1	0	0	0
1	1	0	0	1	0	0	1	0
1	1	0	0	0	0	0	0	0
1	1	0	1	0	1	0	0	0
1	1	0	0	1	0	0	1	0
1	1	0	0	0	0	0	0	0
1	0	1	1	0	0	1	0	0
1	0	1	0	1	0	0	0	1
1	0	1	0	0	0	0	0	0
1	0	1	1	0	0	1	0	0
1	0	1	0	1	0	0	0	1
1	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0
1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0
1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0

You could also set up the matrix "by hand" writing each column out.

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.)

You can use SAS PROC IML or R to construct H; 'ginv' is the function in both that uses the MP inverse. So, for example, you can use 'h=ginv(t(x)*x)*t(x)*x'; in SAS PROC IML. It is possible that there will be

some really small numbers that should be 0, but this is just rounding error (in SAS).

The hypothesis of interest is whether there is a difference in the time trends for groups A and B. However, **in office hours I (EJC) mentioned that you could test the simpler hypothesis of whether there was a difference between groups A and B at the first time point**, to simplify the problem. Note the hypothesis is not the same. Here are the two solutions.

i. Testing whether there is a difference between groups A and B at the first time point. Since this is a hypothesis that requires testing a single equation, i.e. $\gamma_{21} = \gamma_{11}$, then the L matrix is a vector.

- The expected mean for group A at time point 1 is $E(y_{ij}) = \mu_0 + \alpha_1 + \tau_1 + \gamma_{11}$
- The expected mean for group B at time point 1 is $E(y_{ij}) = \mu_0 + \alpha_2 + \tau_1 + \gamma_{21}$

Then the vector to test the hypothesis is $L1 = (0, 1, -1, 0, 0, 0, 0, 1, 0, 0, -1, 0, 0, 0, 0, 0)'$ - see R code below for calculation of L and LH.

ii. Testing whether the time trend for group A versus group B is the same requires 3 equations because we need to show the following: a) differences at time 1 are the same between the two groups, b) differences at time 2 are the same, and c) differences at time 3 are the same. Thus, this contrast requires a test that $\gamma_{11} = \gamma_{21}$, $\gamma_{12} = \gamma_{22}$ and $\gamma_{13} = \gamma_{23}$. This is translated into the $L2$ matrix in the code below.

```
##     [1]   [2]  [3]   [4]   [5]  [6]  [7]   [8]  [9]  [10]  [11]  [12]  [13]  [14]  [15]  [16]
##    mu0 alp1  alp2   alp3 tau1 tau2  tau3 ga11 ga12  ga13  ga21  ga22  ga23  ga31  ga32  ga33
L1 <- c(0,   1,   -1,    0,    0,   0,   0,    1,   0,   0,    -1,    0,    0,    0,    0,    0)
XtX1 <- t(Xmtx1) %*% Xmtx1

H1 <- MASS::ginv(XtX1) %*% XtX1
kable(round(L1 %*% H1), "simple")
```

0	1	-1	0	0	0	0	1	0	0	-1	0	0	0	0	0

```
## testing that the trend over time in group A is the same as the trend in group B
##            mu0 alp1  alp2   alp3 tau1 tau2  tau3 ga11 ga12  ga13  ga21  ga22  ga23   ga31  ga32  ga33
L2 <- matrix(c(0,   1,   -1,    0,    0,   0,   0,    1,   0,   0,    -1,    0,    0,    0,    0,    0,
               0,   1,   -1,    0,    0,   0,   0,    0,   1,   0,    0,    -1,   0,    0,    0,    0,
               0,   1,   -1,    0,    0,   0,   0,    0,   0,   1,    0,    0,    -1,   0,    0,    0), ncol=16, byrow=T)

kable(round(L2), "simple")
```

0	1	-1	0	0	0	0	1	0	0	-1	0	0	0	0	0
0	1	-1	0	0	0	0	0	1	0	0	-1	0	0	0	0
0	1	-1	0	0	0	0	0	0	1	0	0	-1	0	0	0

```
kable(round(L2 %*% H1), "simple")
```

0	1	-1	0	0	0	0	1	0	0	-1	0	0	0	0	0
0	1	-1	0	0	0	0	0	1	0	0	-1	0	0	0	0
0	1	-1	0	0	0	0	0	0	1	0	0	-1	0	0	0

We can see that $\boldsymbol{L}$ and $\boldsymbol{LH}$ are identical, which means these contrasts are estimable. You can verify other contrasts or matrices forms.

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).

You can write this as $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_i)$, where R_i has the AR(1) structure. Note this structure does not change the contrasts in b.

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.

Here we just give a full rank model as an example. Note you could write the design matrix "by hand."

$$Y_{ij} = \beta_0 + \beta_1 G_1 + \beta_2 G_2 + \beta_3 time + \beta_4 (G_1 \times time) + \beta_5 (G_2 \times time) + b_i + \epsilon_{ij}$$
$$b_i \overset{iid}{\sim} N(0, \sigma_b^2)$$
$$\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$$
$$b_i \perp \epsilon_{ij}$$

```
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; kable(Xmtx3, "simple")
```

1	1	0	1	1	0
1	1	0	2	2	0
1	1	0	3	3	0
1	1	0	1	1	0
1	1	0	2	2	0
1	1	0	3	3	0
1	0	1	1	0	1
1	0	1	2	0	2
1	0	1	3	0	3
1	0	1	1	0	1
1	0	1	2	0	2
1	0	1	3	0	3
1	0	0	1	0	0
1	0	0	2	0	0
1	0	0	3	0	0
1	0	0	1	0	0
1	0	0	2	0	0
1	0	0	3	0	0

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

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

A categorical/class variable for time allows estimation of mean levels separately at each time point, but there is no reason why unequally spaced times would be better handled with a categorical time variable, if the focus is on estimating fixed effects. *Note*: If we were concerned about implications for the covariance structure, the standard AR(1) structure would not work well. Discussing this point regarding covariance structure would be an acceptable answer here too.

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

Here are a couple of possibilities. Since there are only 3 times, it is not very expensive to use the UN structure, since it only adds 6 covariance parameters.

$$\boldsymbol{R}_i = \begin{pmatrix} \sigma_0^2 & \sigma_{01} & \sigma_{06} \\ \sigma_{01} & \sigma_1^2 & \sigma_{56} \\ \sigma_{06} & \sigma_{56} & \sigma_6^2 \end{pmatrix}$$

Another option would be the spatial autoregressive power structure. This handles the unequal spacing.

$$\boldsymbol{R}_i = \sigma_\epsilon^2 \begin{pmatrix} 1 & \phi & \phi^6 \\ \phi & 1 & \phi^5 \\ \phi^6 & \phi^5 & 1 \end{pmatrix}$$