[STAT 4400] Exam-2

Michael Ghattas

4/25/2022

# Problem 1

library (foreign)  
library(arm)  
library(cdlTools)  
require(ggplot2)  
require(GGally)  
require(reshape2)  
require(lme4)  
require(compiler)  
require(parallel)  
require(boot)  
require(lattice)  
library(gridExtra)  
library(grid)  
library(dplyr)  
  
  
frisk = read.table(file = "/Users/Home/Documents/Michael\_Ghattas/School/CU\_Boulder/BA-BS/2022/Spring 2022/STAT - 4400/Data/frisk\_with\_noise.dat.txt", header=TRUE, skip = 6)  
head(frisk)

## stops pop past.arrests precinct eth crime  
## 1 75 1720 191 1 1 1  
## 2 36 1720 57 1 1 2  
## 3 74 1720 599 1 1 3  
## 4 17 1720 133 1 1 4  
## 5 37 1368 62 1 2 1  
## 6 39 1368 27 1 2 2

dim(frisk) # 900 x 6

## [1] 900 6

names(frisk)[3] <- "arrests"  
attach(frisk)

### (a)

n.precinct <- max (precinct)  
n.eth <- max (eth)  
n.crime <- max(crime)  
dcjs <- log(arrests\*15/12)  
  
## first let's aggregate  
friskagg <- aggregate(cbind(stops, arrests) ~ precinct + eth, data=frisk, sum)  
  
## These give me the percentages I want  
fr2 <- friskagg %>%  
 group\_by(precinct, eth) %>%  
 summarise(n = sum(stops)) %>%  
 mutate(percentage = n / sum(n))  
  
table2 = aggregate(fr2$percentage, list(fr2$eth) ,FUN = mean)  
colnames(table2) <- c("eth","percbyprec")  
  
head(fr2)

## # A tibble: 6 × 4  
## # Groups: precinct [2]  
## precinct eth n percentage  
## <int> <int> <int> <dbl>  
## 1 1 1 202 0.525  
## 2 1 2 102 0.265  
## 3 1 3 81 0.210  
## 4 2 1 132 0.380  
## 5 2 2 144 0.415  
## 6 2 3 71 0.205

dim(fr2) #225 x 4

## [1] 225 4

## Now I want to classify the precincts  
  
  
precinct.category.vec = ifelse(fr2$eth==1 & fr2$percentage <.1, 1,  
 ifelse(fr2$eth==1 & fr2$percentage < .4, 2,  
 ifelse(fr2$eth==1 & fr2$percentage <= 1, 3,NA)))   
fr3 = as.data.frame(na.omit(cbind(fr2$precinct,precinct.category.vec)))  
  
# length 900, or 12 of each precinct  
fr12 = cbind(frisk, dcjs, rep(as.vector(fr3[,2]), each=12) )  
colnames(fr12)[8] = "precinct.category"  
head(fr12)

## stops pop arrests precinct eth crime dcjs precinct.category  
## 1 75 1720 191 1 1 1 5.475417 3  
## 2 36 1720 57 1 1 2 4.266195 3  
## 3 74 1720 599 1 1 3 6.618405 3  
## 4 17 1720 133 1 1 4 5.113493 3  
## 5 37 1368 62 1 2 1 4.350278 3  
## 6 39 1368 27 1 2 2 3.518980 3

## USE THIS as model 15.1log(arrests) is an offset  
M1 <- as.list (rep (NA, 12))  
index <- 0  
for (j in 1:3){  
 for (k in 1:4){  
 index <- index + 1  
 ok <- fr12$precinct.category==j & fr12$crime==k & fr12$arrests > 0  
 M1[[index]] <- glmer (stops ~ 1 + (1 | eth) + (1|precinct) ,   
 offset = log(arrests),  
 family=poisson(link=log), data=fr12, subset=ok)  
 }}  
  
  
  
allbeta = rep(0,12)  
alltheta = matrix(rep(0,24), nrow= 12, ncol = 2)  
alleth = matrix(rep(0,36), nrow= 12, ncol = 3)  
for(i in 1:12){  
 allbeta[i] = M1[[i]]@beta  
 alltheta[i,] = M1[[i]]@theta  
 alleth[i,] = as.data.frame(coef(M1[[i]])$eth)[,1]  
}  
  
## USE THIS as model 15.5 # log(arrests) is a predictor not the dispersion factor...  
M2 <- as.list (rep (NA, 12))  
index <- 0  
for (j in 1:3){  
 for (k in 1:4){  
 index <- index + 1  
 ok <- fr12$precinct.category==j & fr12$crime==k & fr12$arrests > 0  
 M2[[index]] <- glmer (stops ~ 1 + log(arrests) + (1 | eth) + (1|precinct) ,   
 family=poisson(link=log), data=fr12, subset=ok)  
 }}  
  
allbeta2 = matrix(rep(0,24), nrow= 12, ncol = 2)  
alltheta2 = matrix(rep(0,24), nrow= 12, ncol = 2)  
alleth2 = matrix(rep(0,36), nrow= 12, ncol = 3)  
#allu = matrix(rep(0,84), nrow= 12, ncol = 7)  
for(i in 1:12){  
 allbeta2[i,] = M2[[i]]@beta  
 alltheta2[i,] = M2[[i]]@theta  
 alleth2[i,] = as.data.frame(coef(M2[[i]])$eth)[,1]  
}  
  
theta <- cbind(alltheta,alltheta2)  
beta <- cbind(allbeta,allbeta2)  
eths <- cbind(alleth,alleth2)  
  
theta

## [,1] [,2] [,3] [,4]  
## [1,] 0.3892011 0.37654090 0.4209755 0.40424925  
## [2,] 0.2664679 0.34623141 0.2948581 0.40571387  
## [3,] 0.2712174 0.25946858 0.4564906 0.04537502  
## [4,] 0.5822364 0.13231499 0.7313298 0.33265600  
## [5,] 0.4289294 0.29870804 0.4131146 0.29297015  
## [6,] 0.3977210 0.18106453 0.4038553 0.19267373  
## [7,] 1.0844688 0.29421901 1.0593928 0.30389517  
## [8,] 0.9335737 0.28506005 0.8137612 0.30315303  
## [9,] 0.5902343 0.46840143 0.5604335 0.49240802  
## [10,] 0.5009774 0.32484906 0.4998344 0.34206010  
## [11,] 0.9813271 0.13573197 0.9422604 0.09272197  
## [12,] 0.7582749 0.02935469 0.7616550 0.03032263

beta

## allbeta   
## [1,] -0.3981067 -0.59031096 1.0397312  
## [2,] 0.4062756 -0.09508297 1.1380006  
## [3,] 0.1723844 -1.98836933 1.5071079  
## [4,] -0.9176366 -1.39692357 1.1285541  
## [5,] -0.5877452 0.18942226 0.8570694  
## [6,] 0.6419945 0.90380730 0.9402201  
## [7,] -0.1473768 0.24470419 0.9220117  
## [8,] -1.4625660 -0.59750342 0.8419582  
## [9,] -1.0464958 -0.65380304 0.9312586  
## [10,] 0.5103797 0.63081152 0.9740098  
## [11,] -0.8384846 -0.42146479 0.9186169  
## [12,] -1.8688607 -1.89669751 1.0046404

eths

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] -0.2551779 -0.06464272 -0.86857358 -0.4015183 -0.2601472 -1.1033444  
## [2,] 0.4890077 0.75652331 -0.01628958 0.1813071 0.1735764 -0.6286888  
## [3,] -0.1520131 0.31058394 0.36388634 -1.9908851 -1.9598823 -2.0142507  
## [4,] -0.8109567 -0.92381955 -1.01626489 -1.0138804 -1.5226347 -1.6458958  
## [5,] -0.2750959 -0.51532798 -0.96886518 0.4635086 0.3069800 -0.1982845  
## [6,] 0.7759285 0.74768765 0.40520727 1.0289981 1.0353146 0.6500497  
## [7,] -0.5046673 0.11047986 -0.04557899 -0.1244835 0.5179323 0.3431861  
## [8,] -1.6256891 -1.10042829 -1.65643764 -0.7903844 -0.2036841 -0.7921550  
## [9,] -0.5640511 -0.90522627 -1.66596023 -0.1343643 -0.5210262 -1.3017503  
## [10,] 0.7697108 0.69974341 0.06469146 0.9109909 0.8232484 0.1612331  
## [11,] -1.0010210 -0.74173643 -0.77052895 -0.5248653 -0.3418552 -0.3964415  
## [12,] -1.8860651 -1.84191328 -1.87833755 -1.9162100 -1.8689557 -1.9046459

M <- cbind(M1, M2)  
M

## M1   
## [1,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [2,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [3,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [4,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [5,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [6,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [7,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [8,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [9,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [10,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [11,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [12,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## M2   
## [1,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [2,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [3,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [4,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [5,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [6,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [7,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [8,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [9,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [10,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [11,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>  
## [12,] <S4 class 'glmerMod' [package "lme4"] with 13 slots>

library(sjPlot) #for plotting lmer and glmer mods

## Registered S3 method overwritten by 'parameters':  
## method from   
## format.parameters\_distribution datawizard

help(sjPlot)  
tab\_model(M1, show.re.var= TRUE, dv.labels= "OVERDISPERSED POISSON REGRESSION OF POLICE STOPS")

OVERDISPERSED POISSON REGRESSION OF POLICE STOPS

Predictors

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

(Intercept)

0.67

0.38 – 1.20

0.177

1.50

0.93 – 2.43

0.098

1.19

0.79 – 1.78

0.403

0.40

0.21 – 0.74

0.004

0.56

0.38 – 0.81

0.002

1.90

1.46 – 2.47

<0.001

0.86

0.50 – 1.49

0.598

0.23

0.14 – 0.38

<0.001

0.35

0.20 – 0.61

<0.001

1.67

1.12 – 2.47

0.011

0.43

0.31 – 0.60

<0.001

0.15

0.12 – 0.19

<0.001

Random Effects

σ2

2.01

2.01

2.01

2.01

2.01

2.01

2.01

2.01

2.01

2.01

2.01

2.01

τ00

0.15 precinct

0.07 precinct

0.07 precinct

0.34 precinct

0.18 precinct

0.16 precinct

1.18 precinct

0.87 precinct

0.35 precinct

0.25 precinct

0.96 precinct

0.57 precinct

0.14 eth

0.12 eth

0.07 eth

0.02 eth

0.09 eth

0.03 eth

0.09 eth

0.08 eth

0.22 eth

0.11 eth

0.02 eth

0.00 eth

ICC

0.13

0.09

0.07

0.15

0.12

0.09

0.39

0.32

0.22

0.15

0.33

0.22

N

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

4 precinct

4 precinct

4 precinct

4 precinct

24 precinct

24 precinct

24 precinct

24 precinct

47 precinct

47 precinct

47 precinct

47 precinct

Observations

12

12

12

12

72

72

72

72

141

141

140

141

Marginal R2 / Conditional R2

0.000 / 0.127

0.000 / 0.087

0.000 / 0.065

0.000 / 0.150

0.000 / 0.120

0.000 / 0.087

0.000 / 0.386

0.000 / 0.321

0.000 / 0.220

0.000 / 0.150

0.000 / 0.328

0.000 / 0.222

tab\_model(M2, show.re.var= TRUE, dv.labels= "OVERDISPERSED POISSON REGRESSION OF POLICE STOPS")

OVERDISPERSED POISSON REGRESSION OF POLICE STOPS

Predictors

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

Incidence Rate Ratios

CI

p

(Intercept)

0.55

0.24 – 1.30

0.176

0.91

0.45 – 1.82

0.789

0.14

0.06 – 0.31

<0.001

0.25

0.07 – 0.91

0.036

1.21

0.77 – 1.89

0.409

2.47

1.77 – 3.44

<0.001

1.28

0.70 – 2.32

0.423

0.55

0.30 – 1.02

0.060

0.52

0.28 – 0.96

0.037

1.88

1.23 – 2.88

0.004

0.66

0.46 – 0.94

0.020

0.15

0.11 – 0.21

<0.001

arrests [log]

2.83

2.51 – 3.19

<0.001

3.12

2.78 – 3.51

<0.001

4.51

3.87 – 5.26

<0.001

3.09

2.38 – 4.02

<0.001

2.36

2.25 – 2.47

<0.001

2.56

2.45 – 2.67

<0.001

2.51

2.40 – 2.64

<0.001

2.32

2.16 – 2.50

<0.001

2.54

2.45 – 2.63

<0.001

2.65

2.59 – 2.71

<0.001

2.51

2.41 – 2.61

<0.001

2.73

2.62 – 2.84

<0.001

Random Effects

σ2

0.02

0.02

0.02

0.02

0.02

0.02

0.02

0.02

0.02

0.02

0.02

0.02

τ00

0.18 precinct

0.09 precinct

0.21 precinct

0.53 precinct

0.17 precinct

0.16 precinct

1.12 precinct

0.66 precinct

0.31 precinct

0.25 precinct

0.89 precinct

0.58 precinct

0.16 eth

0.16 eth

0.00 eth

0.11 eth

0.09 eth

0.04 eth

0.09 eth

0.09 eth

0.24 eth

0.12 eth

0.01 eth

0.00 eth

ICC

0.94

0.92

0.91

0.97

0.92

0.90

0.98

0.97

0.96

0.94

0.98

0.96

N

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

3 eth

4 precinct

4 precinct

4 precinct

4 precinct

24 precinct

24 precinct

24 precinct

24 precinct

47 precinct

47 precinct

47 precinct

47 precinct

Observations

12

12

12

12

72

72

72

72

141

141

140

141

Marginal R2 / Conditional R2

0.849 / 0.991

0.912 / 0.993

0.914 / 0.992

0.862 / 0.995

0.573 / 0.966

0.729 / 0.973

0.278 / 0.987

0.550 / 0.987

0.689 / 0.988

0.789 / 0.988

0.610 / 0.991

0.764 / 0.991

### (b)

The advantage of using the level of past arrests as an offset rather than a linear predictor is the reduction of bias in terms of our model and arrests. Since past arrests are taken into consideration as an offset for the model instead of a predictor of outcome.

# Problem 2

library(arm)  
library(ggplot2)  
library(RColorBrewer)  
library(reshape)  
library(wesanderson)  
library(gridExtra)  
library(grid)  
hiv.dataf <- read.csv ("/Users/Home/Documents/Michael\_Ghattas/School/CU\_Boulder/BA-BS/2022/Spring 2022/STAT - 4400/Data/allvar.csv")  
head(hiv.dataf)

## VISIT newpid VDATE CD4PCT arv visage treatmnt CD4CNT baseage  
## 1 1 1 6/29/1988 18 0 3.910000 1 323 3.91  
## 2 4 1 1/19/1989 37 0 4.468333 1 610 3.91  
## 3 7 1 4/13/1989 13 0 4.698333 1 324 3.91  
## 4 10 1 NA 0 5.005000 1 NA 3.91  
## 5 13 1 11/30/1989 13 0 5.330833 1 626 3.91  
## 6 16 1 NA NA NA 1 220 3.91

dim(hiv.dataf) # 1254 x 9

## [1] 1254 9

table(hiv.dataf$treatmnt)

##   
## 1 2   
## 675 579

summary(hiv.dataf$treatmnt)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 1.000 1.462 2.000 2.000

### (a)

attach(hiv.dataf)  
ok <- treatmnt==1 & !is.na(CD4PCT) & (baseage>1 & baseage<5)& !is.na(baseage)   
table(ok) # 369 meet the criteria

## ok  
## FALSE TRUE   
## 885 369

hiv.data = (hiv.dataf[ok,])  
head(hiv.data)

## VISIT newpid VDATE CD4PCT arv visage treatmnt CD4CNT baseage  
## 1 1 1 6/29/1988 18 0 3.910000 1 323 3.9100  
## 2 4 1 1/19/1989 37 0 4.468333 1 610 3.9100  
## 3 7 1 4/13/1989 13 0 4.698333 1 324 3.9100  
## 5 13 1 11/30/1989 13 0 5.330833 1 626 3.9100  
## 7 19 1 6/7/1990 12 1 5.848333 1 220 3.9100  
## 17 1 4 6/23/1988 30 0 2.302500 1 1021 2.3025

dim(hiv.data) # 369 x 9

## [1] 369 9

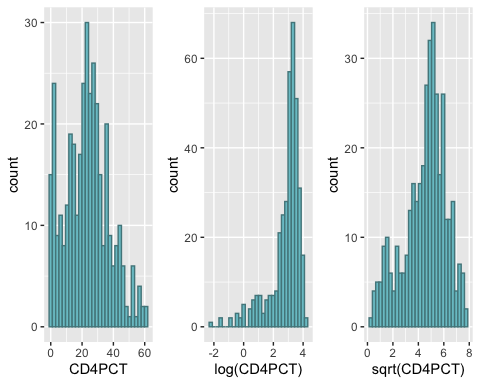
attach(hiv.data)

## The following objects are masked from hiv.dataf:  
##   
## arv, baseage, CD4CNT, CD4PCT, newpid, treatmnt, VDATE, visage,  
## VISIT

p1 = ggplot(hiv.data, aes(x=CD4PCT))+  
 geom\_histogram(color="cadetblue4", fill="cadetblue3")  
  
p2 = ggplot(hiv.data, aes(x=log(CD4PCT)))+  
 geom\_histogram(color="cadetblue4", fill="cadetblue3")  
  
p3 = ggplot(hiv.data, aes(x=sqrt(CD4PCT)))+  
 geom\_histogram(color="cadetblue4", fill="cadetblue3")  
  
grid.arrange(p1,p2,p3, ncol=3)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

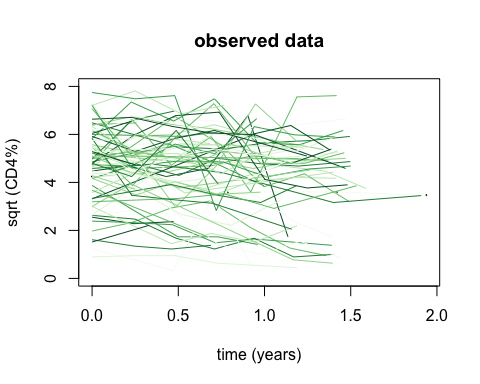
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Redefining variables  
y <- sqrt (CD4PCT) # we are using the square root of the percentage  
age.baseline <- baseage # kid's age (yrs) at the beginning of the study  
age.measurement <- visage # kids age (yrs) at the time of measurement  
treatment <- treatmnt  
time <- visage - baseage  
  
length(unique (hiv.data$newpid)) # there are 83 patients in the dataset of length 369

## [1] 83

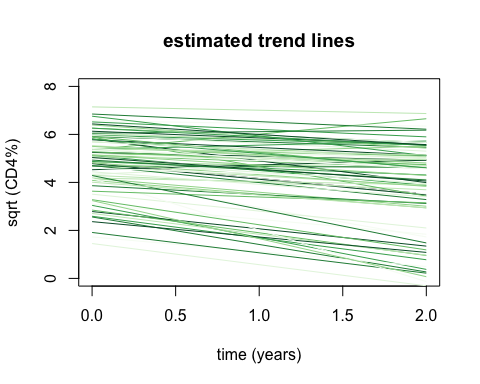
## Set up new patient id numbers from 1 to J  
unique.pid <- unique (newpid)  
n <- length (y)  
J <- length (unique.pid)  
person <- rep (NA, n)  
for (j in 1:J){  
person[newpid==unique.pid[j]] <- j  
}  
  
cols <- rep(brewer.pal(8,'Greens'),20)  
for (j in 1:J){  
if(j==1){  
plot(time[newpid==unique.pid[j]], y[newpid==unique.pid[j]], xlab="time (years)", ylab="sqrt (CD4%)",   
 main="observed data", cex = .1, ylim=c(0,8))  
}  
points(time[newpid==unique.pid[j]], y[newpid==unique.pid[j]], col = cols[j], type="l", ylim=c(0,8))  
}



M1 <- lmer (y ~ time + (1 + time | person))  
display (M1)

## lmer(formula = y ~ time + (1 + time | person))  
## coef.est coef.se  
## (Intercept) 4.85 0.16   
## time -0.47 0.13   
##   
## Error terms:  
## Groups Name Std.Dev. Corr   
## person (Intercept) 1.33   
## time 0.68 0.15   
## Residual 0.75   
## ---  
## number of obs: 369, groups: person, 83  
## AIC = 1108.1, DIC = 1087.8  
## deviance = 1091.9

coef.1 <- matrix(0, J, 1)  
coef.2 <- matrix(0, J, 1)  
coef.1 <- coef(M1)$person[1]  
coef.2 <- coef(M1)$person[2]  
t = time[newpid==unique.pid[1]]  
for (j in 1:J){  
 if(j==1){  
 plot(t , y=coef.1[j,1] + coef.2[j,1]\*t, type="l", xlab="time (years)", ylab="sqrt (CD4%)",   
 main="estimated trend lines", xlim=c(0,2), ylim=c(0,8))  
 }  
 curve(coef.1[j,1] + coef.2[j,1]\*x,col=cols[j], add=T)  
}



CD4.fake <- function(J, K){  
 time <- rep (seq(0,1,length=K), J) # K measurements during the year  
 person <- rep (1:J, each=K) # person ID's  
 treatment <- sample (rep(0:1, J/2))  
 treatment1 <- treatment[person]   
# # hyperparameters  
 mu.a.true <- 4.8 # more generally, these could  
 g.0.true <- -.5 # be specified as additional  
 g.1.true <- .5 # arguments to the function  
 sigma.y.true <- .7  
 sigma.a.true <- 1.3  
 sigma.b.true <- .7  
# # personal-level parameters  
 a.true <- rnorm (J, mu.a.true, sigma.a.true)  
 b.true <- rnorm (J, g.0.true + g.1.true\*treatment, sigma.b.true)  
# # data  
 y <- rnorm (J\*K, a.true[person] + b.true[person]\*time, sigma.y.true)  
 return (data.frame (y, time, person, treatment1))  
}  
   
fake.83.7 = CD4.fake (83,7)  
head(fake.83.7)

## y time person treatment1  
## 1 7.567169 0.0000000 1 0  
## 2 6.485642 0.1666667 1 0  
## 3 6.203211 0.3333333 1 0  
## 4 7.384666 0.5000000 1 0  
## 5 6.281348 0.6666667 1 0  
## 6 6.220887 0.8333333 1 0

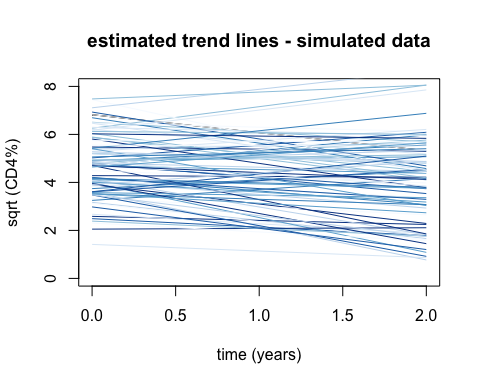
dim(fake.83.7) # 581 x 4 83\*7 = 581

## [1] 581 4

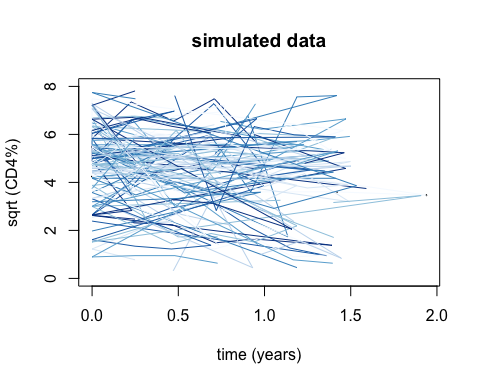
unique.pidf <- unique (fake.83.7$person)  
nf <- length (y)  
Jf <- length (unique.pidf)  
personf <- rep (NA, n)  
for (j in 1:Jf){  
 personf[fake.83.7$person==unique.pidf[j]] <- j  
}  
  
## Fit the model  
M1f <- lmer (y ~ time + (1 + time | person), data=fake.83.7)  
display (M1f)

## lmer(formula = y ~ time + (1 + time | person), data = fake.83.7)  
## coef.est coef.se  
## (Intercept) 4.68 0.16   
## time -0.27 0.13   
##   
## Error terms:  
## Groups Name Std.Dev. Corr   
## person (Intercept) 1.36   
## time 0.82 -0.12   
## Residual 0.70   
## ---  
## number of obs: 581, groups: person, 83  
## AIC = 1584.1, DIC = 1563.6  
## deviance = 1567.9

## Figure 20.5 (c) (using fake data)  
cols <- rep(brewer.pal(8,'Blues'),20)  
coef.1 <- matrix(0, J, 1)  
coef.2 <- matrix(0, J, 1)  
coef.1 <- coef(M1f)$person[1]  
coef.2 <- coef(M1f)$person[2]  
t = time[fake.83.7$person==unique.pidf[1]]  
for (j in 1:J){  
 if(j==1){  
 plot(t , y=coef.1[j,1] + coef.2[j,1]\*t, type="l", xlab="time (years)", ylab="sqrt (CD4%)",   
 main="estimated trend lines - simulated data", xlim=c(0,2), ylim=c(0,8))  
 }  
 curve(coef.1[j,1] + coef.2[j,1]\*x, col = cols[j], add=T)  
}



for (j in 1:J){  
 if(j==1){  
 plot(time[fake.83.7$person==unique.pidf[j]], y[fake.83.7$person==unique.pidf[j]], xlab="time (years)", ylab="sqrt (CD4%)",   
 main="simulated data", cex = .1, ylim=c(0,8))  
 }  
 points(time[fake.83.7$person==unique.pidf[j]], y[fake.83.7$person==unique.pidf[j]], type="l", col=cols[j], ylim=c(0,8))  
}



CD4.power <- function (J, K, n.sims=1000){  
 signif <- rep (NA, n.sims)  
 for (s in 1:n.sims){  
 fake <- CD4.fake (J,K)  
 lme.power <- lmer (y ~ time + time:treatment1 + (1 + time | person),  
 data=fake)  
 theta.hat <- fixef(lme.power)["time:treatment1"]  
 theta.se <- se.fixef(lme.power)["time:treatment1"]  
 signif[s] <- (theta.hat - 2\*theta.se) > 0 # return TRUE or FALSE  
 }  
 power <- mean (signif) # proportion of TRUE  
 return (power)  
}  
  
## these really vary wildly from run to run if nsims is only 100  
CD4.power (J=150, K=7, n.sims=100)

## [1] 0.8

CD4.power (J=110, K=7, n.sims=100)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  
## Model failed to converge with max|grad| = 0.0169073 (tol = 0.002, component 1)

## [1] 0.71

CD4.power (J=80, K=7, n.sims=100)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  
## Model failed to converge with max|grad| = 0.00349175 (tol = 0.002, component 1)

## [1] 0.56

CD4.power (J=50, K=7, n.sims=100)

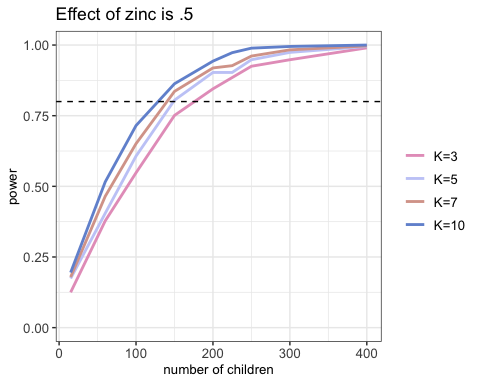
## boundary (singular) fit: see help('isSingular')

## [1] 0.44

J.values <- c(15, 60, 100, 150, 200, 225, 250, 300, 400)  
n.sims.values <- rep(1000,9)  
K.values <- c(3,5,7,10)  
#power.values <- array (NA, c(length(J.values),length(K.values)))  
# for (i1 in 1:length(J.values)){  
# for (i2 in 1:length(K.values)){  
# #cat ("computing power calculation for J =", J.values[i1], ", K =", K.values[i2], "\n")  
# power.values[i1,i2] <- CD4.power (J=J.values[i1], K=K.values[i2], n.sims=n.sims.values[i1])  
# #cat ("power =", power.values[i1,i2], "\n")  
# }  
#}  
  
#save(power.values, J.values, n.sims.values, K.values, file = 'powervalues3.RData')  
load('powervalues3.RData')  
  
dfp = as.data.frame(cbind(seq(1:length(J.values)), J.values, power.values))  
colnames(dfp) = c("ID", "J.values", "K=3", "K=5","K=7", "K=10")  
dfpmelt = melt(dfp,id = c("ID", "J.values"))

### (b)

p <- ggplot(dfpmelt, aes(x = J.values, y = value, color = variable)) +  
 geom\_line(size=1) + ylim(0, 1) +  
 scale\_color\_manual(values = wes\_palette("GrandBudapest2", n = 4)) +  
 theme\_bw() +   
 theme(axis.text=element\_text(size=10),   
 axis.title=element\_text(size=10),   
 legend.text=element\_text(size=10)) +  
 geom\_hline(yintercept = 0.80, linetype = 2) +  
 xlab("number of children") + ylab("power") +   
 ggtitle("Effect of zinc is .5") +   
 theme(legend.title = element\_blank())  
p



# Problem 3

load("/Users/Home/Documents/Michael\_Ghattas/School/CU\_Boulder/BA-BS/2022/Spring 2022/STAT - 4400/Data/schooldata.Rdata")  
head(schooldata)

## id extro open agree social class school  
## 1 1 63.69356 43.43306 38.02668 75.05811 d IV  
## 2 2 69.48244 46.86979 31.48957 98.12560 a VI  
## 3 3 79.74006 32.27013 40.20866 116.33897 d VI  
## 4 4 62.96674 44.40790 30.50866 90.46888 c IV  
## 5 5 64.24582 36.86337 37.43949 98.51873 d IV  
## 6 6 50.97107 46.25627 38.83196 75.21992 d I

### (a)

mod1 <- lm(extro ~ open + agree + social, data = schooldata)  
summary(mod1)

##   
## Call:  
## lm(formula = extro ~ open + agree + social, data = schooldata)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -30.3151 -6.0743 -0.1586 6.2851 30.0167   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 57.839518 3.148056 18.373 <2e-16 \*\*\*  
## open 0.024749 0.046471 0.533 0.594   
## agree 0.026538 0.053347 0.497 0.619   
## social 0.005082 0.017303 0.294 0.769   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9.342 on 1196 degrees of freedom  
## Multiple R-squared: 0.0005222, Adjusted R-squared: -0.001985   
## F-statistic: 0.2083 on 3 and 1196 DF, p-value: 0.8907

display(mod1)

## lm(formula = extro ~ open + agree + social, data = schooldata)  
## coef.est coef.se  
## (Intercept) 57.84 3.15   
## open 0.02 0.05   
## agree 0.03 0.05   
## social 0.01 0.02   
## ---  
## n = 1200, k = 4  
## residual sd = 9.34, R-Squared = 0.00

The model is unhelpful, we are unable to make any inference with certinty, and the response looks independent from the predictors for the most part.

### (b)

require(lme4)  
  
mod2 <- lmer (extro ~ open + agree + social + (open | school) + (agree | school) + (social | school), data = schooldata)

## boundary (singular) fit: see help('isSingular')

summary(mod2)

## Linear mixed model fit by REML ['lmerMod']  
## Formula: extro ~ open + agree + social + (open | school) + (agree | school) +   
## (social | school)  
## Data: schooldata  
##   
## REML criterion at convergence: 5821  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -5.8769 -0.5140 0.0006 0.5174 6.0296   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## school (Intercept) 2.469e+01 4.96860   
## open 2.504e-04 0.01582 1.00   
## school.1 (Intercept) 2.474e+01 4.97437   
## agree 1.414e-05 0.00376 -1.00  
## school.2 (Intercept) 2.455e+01 4.95520   
## social 1.123e-04 0.01059 1.00   
## Residual 7.102e+00 2.66487   
## Number of obs: 1200, groups: school, 6  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 59.121156 3.625935 16.305  
## open 0.009540 0.014814 0.644  
## agree 0.027026 0.015359 1.760  
## social -0.001843 0.006576 -0.280  
##   
## Correlation of Fixed Effects:  
## (Intr) open agree   
## open 0.112   
## agree -0.201 -0.008   
## social 0.265 0.002 -0.003  
## optimizer (nloptwrap) convergence code: 0 (OK)  
## boundary (singular) fit: see help('isSingular')

display(mod2)

## lmer(formula = extro ~ open + agree + social + (open | school) +   
## (agree | school) + (social | school), data = schooldata)  
## coef.est coef.se  
## (Intercept) 59.12 3.63   
## open 0.01 0.01   
## agree 0.03 0.02   
## social 0.00 0.01   
##   
## Error terms:  
## Groups Name Std.Dev. Corr   
## school (Intercept) 4.97   
## open 0.02 1.00   
## school.1 (Intercept) 4.97   
## agree 0.00 -1.00   
## school.2 (Intercept) 4.96   
## social 0.01 1.00   
## Residual 2.66

## Warning in optwrap(optimizer, devfun, x@theta, lower = x@lower, calc.derivs =  
## TRUE, : convergence code 1 from bobyqa: bobyqa -- maximum number of function  
## evaluations exceeded

## ---  
## number of obs: 1200, groups: school, 6  
## AIC = 5849, DIC = 5785.7  
## deviance = 5803.4

With this model, the parameters are estimated as follows:

Unexplained within-school variation = 2.66 \

School-Open intercepts variation = 4.97 \ School-agree intercepts variation = 4.97 \ School-Social intercepts variation = 4.96 \

School-Open slopes variation = 0.02 \ School-agree slopes variation = 0.00 \ School-Social slopes variation = 0.01 \

Correlation between intercepts and slopes (School-Open) $ $ = 1 \ Correlation between intercepts and slopes (School-agree) $ $ = -1 \ Correlation between intercepts and slopes (School-Social) $ $ = 1 \

Fixed effect, school mean intercept = 59.12 \ Fixed effect, School-Open mean slope & = 0.01 \ Fixed effect, School-Open mean slope & = 0.03 \ Fixed effect, School-Open mean slope & = 0.00 \

### (c)

require(lme4)  
  
mod3 <- lmer (extro ~ open + agree + social + school:class + (1 + open | school) + (1 + open | school:class) + (1 + agree | school:class) + (1 + social | school:class), data = schooldata)

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient

## boundary (singular) fit: see help('isSingular')

summary(mod3)

## Linear mixed model fit by REML ['lmerMod']  
## Formula: extro ~ open + agree + social + school:class + (1 + open | school) +   
## (1 + open | school:class) + (1 + agree | school:class) +   
## (1 + social | school:class)  
## Data: schooldata  
##   
## REML criterion at convergence: 3418.4  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -9.9923 -0.3343 0.0042 0.3386 10.6753   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## school.class (Intercept) 9.690e-01 0.9843771   
## social 2.401e-11 0.0000049 -1.00  
## school.class.1 (Intercept) 9.652e-01 0.9824428   
## agree 4.023e-05 0.0063426 -0.95  
## school.class.2 (Intercept) 9.693e-01 0.9845173   
## open 7.326e-06 0.0027066 -1.00  
## school (Intercept) 9.638e-01 0.9817460   
## open 5.878e-06 0.0024245 0.13   
## Residual 9.670e-01 0.9833553   
## Number of obs: 1200, groups: school:class, 24; school, 6  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 7.999e+01 1.854e+00 43.137  
## open 6.150e-03 5.093e-03 1.208  
## agree -7.650e-03 5.848e-03 -1.308  
## social 5.361e-04 1.852e-03 0.289  
## schoolI:classa -3.973e+01 2.574e+00 -15.434  
## schoolII:classa -2.775e+01 2.569e+00 -10.799  
## schoolIII:classa -2.309e+01 2.573e+00 -8.972  
## schoolIV:classa -1.946e+01 2.575e+00 -7.558  
## schoolV:classa -1.517e+01 2.572e+00 -5.900  
## schoolVI:classa -1.015e+01 2.151e+00 -4.719  
## schoolI:classb -3.432e+01 2.573e+00 -13.339  
## schoolII:classb -2.639e+01 2.571e+00 -10.265  
## schoolIII:classb -2.224e+01 2.572e+00 -8.646  
## schoolIV:classb -1.829e+01 2.570e+00 -7.115  
## schoolV:classb -1.390e+01 2.574e+00 -5.401  
## schoolVI:classb -8.153e+00 2.144e+00 -3.803  
## schoolI:classc -3.161e+01 2.572e+00 -12.289  
## schoolII:classc -2.523e+01 2.572e+00 -9.807  
## schoolIII:classc -2.124e+01 2.575e+00 -8.246  
## schoolIV:classc -1.726e+01 2.571e+00 -6.713  
## schoolV:classc -1.287e+01 2.573e+00 -5.003  
## schoolVI:classc -5.309e+00 2.146e+00 -2.473  
## schoolI:classd -2.974e+01 2.574e+00 -11.554  
## schoolII:classd -2.423e+01 2.570e+00 -9.429  
## schoolIII:classd -2.011e+01 2.571e+00 -7.825  
## schoolIV:classd -1.623e+01 2.572e+00 -6.312  
## schoolV:classd -1.161e+01 2.570e+00 -4.518

##   
## Correlation matrix not shown by default, as p = 27 > 12.  
## Use print(x, correlation=TRUE) or  
## vcov(x) if you need it

## fit warnings:  
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient  
## optimizer (nloptwrap) convergence code: 0 (OK)  
## boundary (singular) fit: see help('isSingular')

display(mod3)

## lmer(formula = extro ~ open + agree + social + school:class +   
## (1 + open | school) + (1 + open | school:class) + (1 + agree |   
## school:class) + (1 + social | school:class), data = schooldata)  
## coef.est coef.se  
## (Intercept) 79.99 1.85   
## open 0.01 0.01   
## agree -0.01 0.01   
## social 0.00 0.00   
## schoolI:classa -39.73 2.57   
## schoolII:classa -27.75 2.57   
## schoolIII:classa -23.09 2.57   
## schoolIV:classa -19.46 2.58   
## schoolV:classa -15.17 2.57   
## schoolVI:classa -10.15 2.15   
## schoolI:classb -34.32 2.57   
## schoolII:classb -26.39 2.57   
## schoolIII:classb -22.24 2.57   
## schoolIV:classb -18.29 2.57   
## schoolV:classb -13.90 2.57   
## schoolVI:classb -8.15 2.14   
## schoolI:classc -31.61 2.57   
## schoolII:classc -25.23 2.57   
## schoolIII:classc -21.24 2.58   
## schoolIV:classc -17.26 2.57   
## schoolV:classc -12.87 2.57   
## schoolVI:classc -5.31 2.15   
## schoolI:classd -29.74 2.57   
## schoolII:classd -24.23 2.57   
## schoolIII:classd -20.11 2.57   
## schoolIV:classd -16.23 2.57   
## schoolV:classd -11.61 2.57   
##   
## Error terms:  
## Groups Name Std.Dev. Corr   
## school.class (Intercept) 0.98   
## social 0.00 -1.00   
## school.class.1 (Intercept) 0.98   
## agree 0.01 -0.95   
## school.class.2 (Intercept) 0.98   
## open 0.00 -1.00   
## school (Intercept) 0.98   
## open 0.00 0.13   
## Residual 0.98

## Warning in optwrap(optimizer, devfun, x@theta, lower = x@lower, calc.derivs =  
## TRUE, : convergence code 1 from bobyqa: bobyqa -- maximum number of function  
## evaluations exceeded

## ---  
## number of obs: 1200, groups: school:class, 24; school, 6  
## AIC = 3498.4, DIC = 3281  
## deviance = 3349.7

This model would be helpful if we are trying to predict extroversion based on openness, agreeableness, and social ability within a school and per class. It should be helpful as it takes into account multiple predictors and effects, allowing for a more accurate model and improved certainty. ### (d)

library(sjPlot) #for plotting lmer and glmer mods  
library(gridExtra)  
  
plot1 = plot\_model(mod2, show.values=FALSE, show.p=TRUE, title="Varying Intercept")  
plot2 = plot\_model(mod3, show.values=FALSE, show.p=TRUE, title="varying slope & Intercept")  
  
grid.arrange(plot1, plot2, ncol = 2)

