STAT 635 F2022 Assignment 3 - Due on Dec. 9, Friday, 2022, 11:59pm

Instructor: Xuewen Lu (This PDF is made from an R Markdown File)

2022-12-09

Contents

.

General Policies [Total 60 marks]

- Your assignment solutions should be created in `"RStudio"` with R Markdown to include the R codes and should be saved as an R Markdown file and a PDF file generated from the R Markdown file. You should name your files as `"Lastname-Firstname-Stat635-A3.Rmd"` and `"Lastname-Firstname-Stat635-A3.pdf"`.
- Test your R codes before submission to make sure it can be executed successfully in `"RStudio"`.
- For each assignment, submit only one PDF file and the associated Rmd file to D2L, and the same PDF to gradescope.ca. Only the PDF file in gradescope is graded. The Rmd file may be used to test your R programs when needed.
- If Monte-Carlo methods are used, you must fix the random seed in your R code by using **set.seed(2022)**.
- For all numerical problems, summarize the computer generated results in tables or figures and interpret them using your own words, then draw conclusions from them. **Show all your work and spell out the details**.
- Late submission is not acceptable.
- For guidelines on how to write a good assignment, go to D2L to read two sample assignments, **Bad-assignment-example.pdf** and **Good-assignment-example.pdf**, one is bad and the other is good, you are expected to do a good one.

Note: Materials are based on Dobson and Barnett (D&B or DB), 3rd Edition, 2008.

Problem 1. [12 marks; (a) 3, (b) 3, (c), 3, (d), 3]

The data in Table 1 are numbers of insurance policies, n, and numbers of claims, y, for cars in various insurance categories, CAR, tabulated by age of policy holder, AGE, and district where the policy holder lived (DIST = 1, for London and other major cities, and DIST = 0, otherwise). The table is derived from the CLAIMS data set in Aitkin et al. (2005) obtained from a paper by Baxter et al. (1980).

Table 1: Car insurance claims: based on the CLAIMS data set reported by Aitkin et al. (2005).

		DIST=0		DIST=1	
CAR	AGE	y	n	y	n
1	1	65	317	2	20
1	2	65	476	5	33
1	3	52	486	4	40
1	4	310	3259	36	316
2	1	98	486	7	31
2	2	159	1004	10	81
2	3	175	1355	22	122
2	4	877	7660	102	724
3	1	41	223	5	18
3	2	117	539	7	39
3	3	137	697	16	68
3	4	477	3442	63	344
4	1	11	40	0	3
4	2	35	148	6	16
4	3	39	214	8	25
4	4	167	1019	33	114

```
Car = rep(1:4, each = 4)
Age = rep(1:4, 4)
D0_y = c(65,65,52,310,98,159,175,877,41,117,137,477,11,35,39,167)
D0_n = c(317,476,486,3259,486,1004,1355,7660,223,539,697,3442,40,148,214,1019)

D1_y = c(2,5,4,36,7,10,22,102,5,7,16,63,0,6,8,33)
D1_n = c(20,33,40,316,31,81,122,724,18,39,68,344,3,16,25,114)

CarInsurance = cbind(Car,Age,D0_y,D0_n,D1_y,D1_n)
head(CarInsurance)
```

```
##      Car Age D0_y D0_n D1_y D1_n
## [1,]   1   1   65  317    2   20
## [2,]   1   2   65  476    5   33
## [3,]   1   3   52  486    4   40
## [4,]   1   4  310 3259   36  316
## [5,]   2   1   98  486    7   31
## [6,]   2   2  159 1004   10   81
```

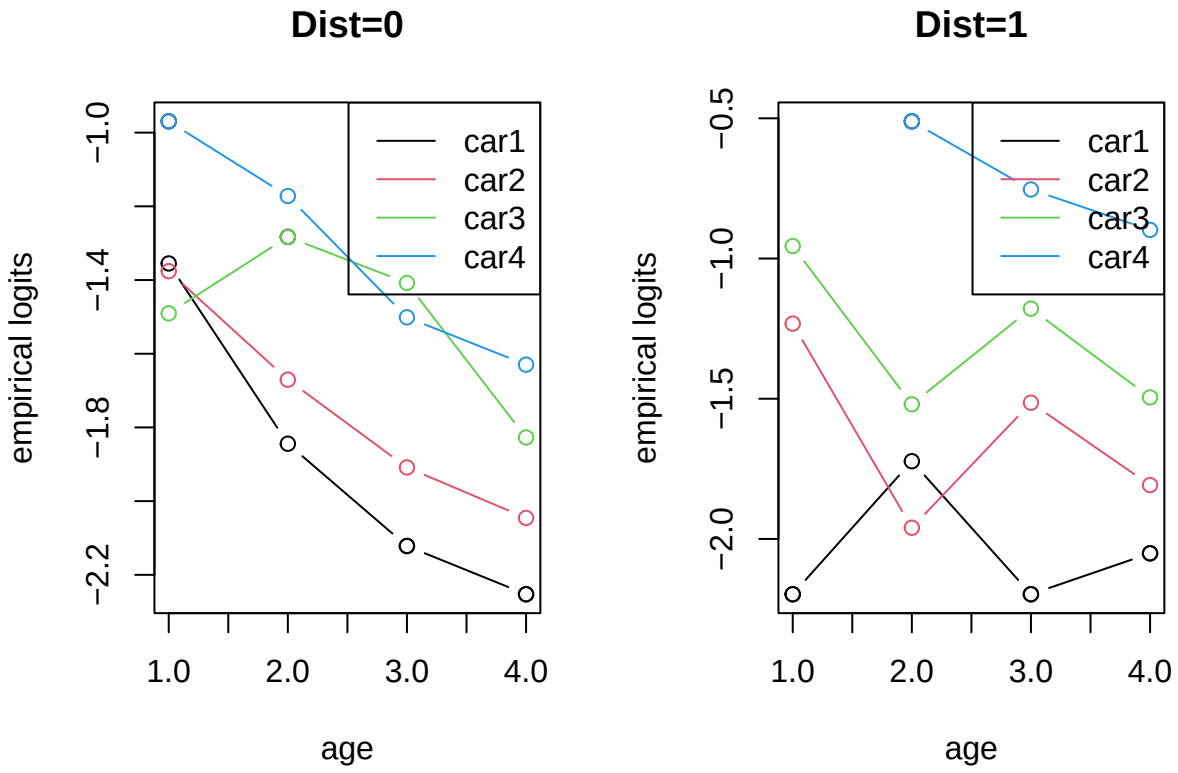
(a) Calculate the rate of claims $rate = y/n$ and empirical logit $emplogit = \log(rate/(1 - rate))$ for each category and plot the empirical logits by AGE,CAR and DIST to get an idea of the main effects of these factors.

```
D0_rate = D0_y/D0_n
D0_emp = log(D0_rate/(1-D0_rate))
D1_rate = D1_y/D1_n
D1_emp = log(D1_rate/(1-D1_rate))
CarInsurance = cbind(Car,Age,D0_y,D0_n,D0_rate,D0_emp,D1_y,D1_n,D1_rate,D1_emp)
head(CarInsurance)
```

```
##      Car Age D0_y D0_n   D0_rate    D0_emp D1_y D1_n   D1_rate    D1_emp
## [1,]   1   1   65  317 0.2050473 -1.355042    2   20 0.1000000 -2.197225
## [2,]   1   2   65  476 0.1365546 -1.844206    5   33 0.1515152 -1.722767
## [3,]   1   3   52  486 0.1069959 -2.121801    4   40 0.1000000 -2.197225
## [4,]   1   4  310 3259 0.0951212 -2.252649   36  316 0.1139241 -2.051271
## [5,]   2   1   98  486 0.2016461 -1.376038    7   31 0.2258065 -1.232144
## [6,]   2   2  159 1004 0.1583665 -1.670432   10   81 0.1234568 -1.960095
```

```
library(gplots)
par(mfrow = c(1,2))
## empirical logits for Dist 0
age = c(1:4)
car1 = D0_emp[1:4]
car2 = D0_emp[5:8]
car3 = D0_emp[9:12]
car4 = D0_emp[13:16]
yaxis = c(max(car4),max(car3),car1[3],min(car1)) # to make the graph look better
plot(yaxis~age, xlab = "age", ylab = "empirical logits", main = "Dist=0")
lines(car1~age, type = "b", lty =1, col =1, data = CarInsurance[1:4,])
lines(car2~age, type = "b", lty =1, col =2, data = CarInsurance[5:8,])
lines(car3~age, type = "b", lty =1, col =3, data = CarInsurance[9:12,])
lines(car4~age, type = "b", lty =1, col =4, data = CarInsurance[13:16,])
legend("topright",legend = c("car1","car2","car3","car4"), lty = 1, col = c(1:4))

## empirical logits for Dist 1
age = c(1:4)
car1 = D1_emp[1:4]
car2 = D1_emp[5:8]
car3 = D1_emp[9:12]
car4 = D1_emp[13:16]
yaxis = c(min(car1),max(car4),min(car1),car1[4]) # to make the graph look better
plot(yaxis~age, xlab = "age", ylab = "empirical logits", main = "Dist=1")
lines(car1~age, type = "b", lty =1, col =1, data = CarInsurance[1:4,])
lines(car2~age, type = "b", lty =1, col =2, data = CarInsurance[5:8,])
lines(car3~age, type = "b", lty =1, col =3, data = CarInsurance[9:12,])
lines(car4~age, type = "b", lty =1, col =4, data = CarInsurance[13:16,])
legend("topright",legend = c("car1","car2","car3","car4"), lty = 1, col = c(1:4))
```



(b) Use Poisson regression to fit a main effects model (a model without interactions) and state the fitted model in an equation (each covariate is treated as categorical and modeled using indicator variables), and then use deviance test to check if any interaction term (just test one of the three two way interactions each time, they are AGE×CAR, AGE×DIST, CAR×DIST) can improve the fit. **Note:** when a variable is categorical, let's treat the lowest level as the reference level when you create dummy variables or nominal variables for that variable. For example, when the variable `CAR` is treated as a categorical variable, it has 4 levels, you can use `CAR=1` as the reference level and define three dummy variable for it. Similarly, use `AGE=1` and `DIST=0` as the reference levels.

```
## create a data frame in long form
Car = rep(1:4, each = 4,2)
Age = rep(1:4, 8)
Dist = rep(0:1, each = 16)
y = c(65,65,52,310,98,159,175,877,41,117,137,477,11,35,39,167,
      2,5,4,36,7,10,22,102,5,7,16,63,0,6,8,33)
n = c(317,476,486,3259,486,1004,1355,7660,223,539,697,3442,40,148,214,1019,
      20,33,40,316,31,81,122,724,18,39,68,344,3,16,25,114)

CarData = data.frame(cbind(Car, Age, Dist, y, n))
CarData$Car = c(factor(CarData$Car))
CarData$Age = c(factor(CarData$Age))
CarData$Dist = c(factor(CarData$Dist))

model_1 = glm(y~Car+Age+Dist+offset(log(n)), family = poisson(link=log), data = CarData)
summary(model_1)
```

```
##
## Call:
## glm(formula = y ~ Car + Age + Dist + offset(log(n)), family = poisson(link = log),
##     data = CarData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8588  -0.6371  -0.1555   0.3749   1.7536
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.81021    0.07532 -24.034  < 2e-16 ***
## Car2         0.16229    0.05052   3.213 0.001315 **
## Car3         0.39352    0.05498   7.157 8.25e-13 ***
## Car4         0.56540    0.07228   7.823 5.18e-15 ***
## Age2        -0.18902    0.08282  -2.282 0.022477 *
## Age3        -0.34211    0.08130  -4.208 2.58e-05 ***
## Age4        -0.53275    0.06979  -7.634 2.28e-14 ***
## Dist1        0.21850    0.05853   3.733 0.000189 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 207.833  on 31  degrees of freedom
## Residual deviance:  23.709  on 24  degrees of freedom
## AIC: 208.07
##
## Number of Fisher Scoring iterations: 4
```

Model: $log(\mu) = log(n) + \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7$

where $x_1 = I[Car2], x_2 = I[Car3], x_3 = I[Car4]$, $x_4 = I[Age2], x_5 = I[Age3], x_6 = I[Age4]$, $x_7 = I[Dist1]$

Fitted Model: $log(\mu) = log(n) - 1.81 + 0.16x_1 + 0.39x_2 + 0.57x_3 - 0.19x_4 - 0.34x_5 - 0.53x_6 + 0.22x_7$

```
model.AC = glm(y~Car+Age+Dist+Car*Age+offset(log(n)), family = poisson(link=log), data = CarData)
1-pchisq(deviance(model_1)-deviance(model.AC),df.residual(model_1)-df.residual(model.AC))
```

```
## [1] 0.31025
```

```
model.AD = glm(y~Car+Age+Dist+Age*Dist+offset(log(n)), family = poisson(link=log), data = CarData)
1-pchisq(deviance(model_1)-deviance(model.AD),df.residual(model_1)-df.residual(model.AD))
```

```
## [1] 0.2851265
```

```
model.CD = glm(y~Car+Age+Dist+Car*Dist+offset(log(n)), family = poisson(link=log), data = CarData)
1-pchisq(deviance(model_1)-deviance(model.CD),df.residual(model_1)-df.residual(model.CD))
```

```
## [1] 0.2179738
```

The results show that adding two-term interaction is NOT going to improve the model.

(c) Based on the modelling in (b), Aitkin et al. (2005) determined that all the interactions were unimportant and decided that AGE and CAR could be treated as though they were continuous variables. Fit a main effects model incorporating these features and compare it with the best model obtained in (b) using, for example, AIC. What conclusions do you reach?

```
## create a data frame in long form
Car = rep(1:4, each = 4,2)
Age = rep(1:4, 8)
Dist = rep(0:1, each = 16)
y = c(65,65,52,310,98,159,175,877,41,117,137,477,11,35,39,167,
      2,5,4,36,7,10,22,102,5,7,16,63,0,6,8,33)
n = c(317,476,486,3259,486,1004,1355,7660,223,539,697,3442,40,148,214,1019,
      20,33,40,316,31,81,122,724,18,39,68,344,3,16,25,114)

CarData = data.frame(cbind(Car, Age, Dist, y, n))
CarData$Dist = c(factor(CarData$Dist))
model_2 = glm(y~Car+Age+Dist+offset(log(n)), family = poisson(link=log), data = CarData)
summary(model_2)
```

```
##
## Call:
## glm(formula = y ~ Car + Age + Dist + offset(log(n)), family = poisson(link = log),
##     data = CarData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7248  -0.5681  -0.1679   0.3384   1.9126
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.85253    0.07990 -23.185  < 2e-16 ***
## Car          0.19777    0.02080   9.507  < 2e-16 ***
## Age         -0.17674    0.01849  -9.559  < 2e-16 ***
## Dist1        0.21865    0.05853   3.736 0.000187 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 207.833  on 31  degrees of freedom
## Residual deviance:  24.685  on 28  degrees of freedom
## AIC: 201.05
##
## Number of Fisher Scoring iterations: 4
```

Since Model_1 and Model_2 are not in a nested relation, we use AIC instead of deviance. Specifically, AIC of Model_1 is 208.07 and AIC of Model_2 is 201.05. Smaller AIC means a better model. So Model_2 is more preferred.

(d) Read the paper *Logistic and Poisson Regression* by William Chiu (2017) to fit a Poisson regression main effects model with an offset and a logistic regression to the data, respectively, and show the obtained estimation results are equivalent or similar. Paper is found in D2L or at

https://rstudio-pubs-static.s3.amazonaws.com/238765_5a165fc24624448293f99a6a20434778.html

This paper defines $p = \frac{A}{A+B}$ and it proves $log(\frac{p}{1-p}) = log(\frac{A}{B})$. This teaches us the way to build a logistic regression model and a poisson regression model (note: this time, the offset term is different than what we got on part b). The code below indicates that the obtained estimation results are similar.

```
library(pander)
```

```
## Warning: package 'pander' was built under R version 4.2.2
```

```
logit.model = glm(cbind(y,n-y)~Car+Age+Dist, data = CarData, family = binomial(link = "logit"))
pander(logit.model, caption="logisitic model")
```

Table 2: logisitic model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.667	0.08748	-19.06	5.366e-81
Car	0.2317	0.02266	10.23	1.525e-24
Age	-0.2097	0.0204	-10.28	8.891e-25
Dist1	0.2589	0.0642	4.033	5.504e-05

```
poisson.model = glm(y~Car+Age+Dist+offset(log(n-y)), data = CarData, family = poisson(link = "log"))
pander(poisson.model, caption="poisson model")
```

Table 3: poisson model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.673	0.07998	-20.92	3.846e-97
Car	0.232	0.02086	11.12	9.487e-29
Age	-0.2083	0.01849	-11.27	1.869e-29
Dist1	0.26	0.05851	4.444	8.822e-06

Problem 2. [9 marks; (a) 3, (b) 3, (c), 3]

Consider the flu vaccine trial data in Table 4.

Table 4: Flu vaccine trial.

	Response			
	Small	Moderate	Large	Total
Placebo	25	8	5	38
Vaccine	6	18	11	35

(a) Using a conventional chi-squared test model, test the hypothesis that the distribution of responses is the same for the placebo and vaccine groups.

We can test $H_0 : Trmt * Response = 0$ and $H_a : Trmt * Response \neq 0$

full model is: $log(\mu_{ij}) = \mu + Trmt + Response + Trmt * Response$

reduced model is: $log(\mu_{ij}) = \mu + Trmt + Response$, and we compare these two models.

```
## read the data
Trmt = rep(0:1,each=3)
Response = rep(0:2,2)
Y = c(25,8,5,6,18,11)
TrialData = data.frame(cbind(Trmt,Response,Y))
TrialData
```

```
##   Trmt Response  Y
## 1    0        0 25
## 2    0        1  8
## 3    0        2  5
## 4    1        0  6
## 5    1        1 18
## 6    1        2 11
```

```
# fit the full model and reduced model
model_full = glm(Y~Trmt+Response+Trmt*Response, family = poisson, data = TrialData)
model_reduced = glm(Y~Trmt+Response, family = poisson, data = TrialData)
# Chi-squared test
1-pchisq(deviance(model_reduced)-deviance(model_full),
         df.residual(model_reduced)-df.residual(model_full))
```

```
## [1] 0.000289178
```

Since the p-value is smaller than $\alpha = 0.05$, we reject the null hypothesis, meaning that the interaction term is significant, which implies that the distribution of response is NOT the same for the placebo and vaccine groups

(b) For the model corresponding to the hypothesis of homogeneity of response distributions in part (a), fit a log-linear model, then calculate the fitted values (cell counts), the Pearson and deviance residuals, and the goodness of fit statistics X^2 and D. Which of the cells of the table contribute most to X^2 and D? Do X^2 and D indicate the model is a good fit to the data? Explain and interpret these results.

```
table_1 = data.frame(TrialData$Y,predict(model_reduced, type = "response"),
                     resid(model_reduced, "pearson"), resid(model_reduced, "deviance"))
names(table_1) = c("Original", "Fitted Value", "Pearson Res", "Deviance Res")
table_1
```

```
##   Original Fitted Value Pearson Res Deviance Res
## 1       25    16.772943   2.0088146    1.8711826
## 2        8    12.262333  -1.2171968   -1.3004673
## 3        5     8.964724  -1.3241723   -1.4459785
## 4        6    15.448763  -2.4039649   -2.7474157
## 5       18    11.294254   1.9953465    1.8350215
## 6       11     8.256983   0.9545922    0.9079418
```

```
# GOF Statistics Chi-squared
sum(resid(model_reduced, "pearson")^2)
```

```
## [1] 17.94204
```

```
# GOF Statistics Deviance
sum(resid(model_reduced, "deviance")^2)
```

```
## [1] 19.02335
```

We can see that Cell 4 (Vaccine with Small Response) contributes the most to deviance and Cell 4 (Vaccine with Small Response) contributes the most to the GOF X^2.

```
1-pchisq(sum(resid(model_reduced, "pearson")^2)-sum(resid(model_full, "pearson")^2),1)
```

```
## [1] 0.0006093786
```

```
1-pchisq(sum(resid(model_reduced, "deviance")^2)-sum(resid(model_full, "deviance")^2),1)
```

```
## [1] 0.000289178
```

Also, both X^2 and D indicate the model (the one without interaction term) is not a good fit to the data

(c) Re-analyze these data using ordinal logistic regression using `Small` as the first level of the response, write down the estimated model equations, and compute the fitted probability in each cell and the fitted count in each cell, compare the result with that in part (b).

```
# Asked to use SMALL as the first level of the response = treat Small as the reference
library(VGAM)
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Make a wide data form for using VGLM
Trmt = rep(c("Placebo","Vaccine"),each=3)
Response = rep(c("Small","Moderate","Large"),2)
Y = c(25,8,5,6,18,11)
TrialData = data.frame(cbind(Trmt,Response,Y))
TrialData$Trmt = factor(TrialData$Trmt, levels = c("Placebo", "Vaccine"))
TrialData$Response = factor(TrialData$Response, levels = c("Small", "Moderate", "Large"))
library(reshape2)
WideData = dcast(TrialData, Trmt~Response, value.var="Y")
WideData$Small = as.numeric(WideData$Small)
WideData$Moderate = as.numeric(WideData$Moderate)
WideData$Large = as.numeric(WideData$Large)
```

```r
prop.odds.model = vglm(cbind(Small, Moderate, Large)~Trmt,
                       family = cumulative(parallel = TRUE), data = WideData)
summary(prop.odds.model)
```

```
##
## Call:
## vglm(formula = cbind(Small, Moderate, Large) ~ Trmt, family = cumulative(parallel = TRUE),
##     data = WideData)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1   0.5650     0.3338   1.693 0.090472 .
## (Intercept):2   2.4408     0.4470   5.460 4.77e-08 ***
## TrmtVaccine    -1.8373     0.4878  -3.766 0.000166 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2])
##
## Residual deviance: 2.9601 on 1 degrees of freedom
##
## Log-likelihood: -8.6051 on 1 degrees of freedom
##
## Number of Fisher scoring iterations: 4
##
```

```
## No Hauck-Donner effect found in any of the estimates
##
##
## Exponentiated coefficients:
## TrmtVaccine
##   0.1592487
```

Proportional Odds Model Equations:

$L_1 = log\left(\frac{\pi_1}{\pi_2+\pi_3}\right); L2 = log\left(\frac{\pi_1+\pi_2}{\pi_3}\right)$

Estimated Model:

$L_1 = 0.565 - 1.8373 * Treatment; L2 = 2.4408 - 1.8373 * Treatment$

Fitted Probability:

```
# we can obtain fitted probability in each cell using this R-code
fit1 = attributes(prop.odds.model)
print(fit1$fitted.values)
```

```
##       Small  Moderate      Large
## 1 0.6376202 0.2822627 0.08011713
## 2 0.2188743 0.4275751 0.35355055
```

Fitted Count:

```
# we can obtain fitted count in each cell using this R-code
prob1 = c(0.6376202, 0.2822627, 0.08011713) # for Placebo
prob2 = c(0.2188743, 0.4275751, 0.35355055) # for Vaccine
Cellcount1 = prob1*38
Cellcount2 = prob2*35
setNames(Cellcount2,Cellcount1)
```

```
## 24.2295676 10.7259826 3.04445094
##   7.660601  14.965128  12.374269
```

By observation, the model can give us a quite close fitted count in each cell. I want to use McFadden's R^2 to check it. $R^2 = 1 - \frac{Dev(b)}{Dev(min)}$

```
dev.b = deviance(prop.odds.model)
prop.odds.model.null = vglm(cbind(Small, Moderate, Large)~1,
                      family = cumulative(parallel = TRUE), data = WideData)
dev.min = deviance(prop.odds.model.null)
McF.Rsq = 1-dev.b/dev.min
cat("McFadden's pseudo-R-squared of the model=", McF.Rsq, "%", "\n")
```

```
## McFadden's pseudo-R-squared of the model= 0.8412176 %
```

Problem 3. [27 marks; (a)-(f), each 2 marks, (g)-(k), each 3 mark]

Jensen, Birch, and Woodall (2008) considered profile monitoring of a calibration data set in which the data consists of 22 calibration samples. One of the purposes of the experiment was to determine the relationship between an absorbance measure (absorbance) of a chemical solution to the volume at which the solution was prepared (volume). The raw data are provided in calibration.xlsx.

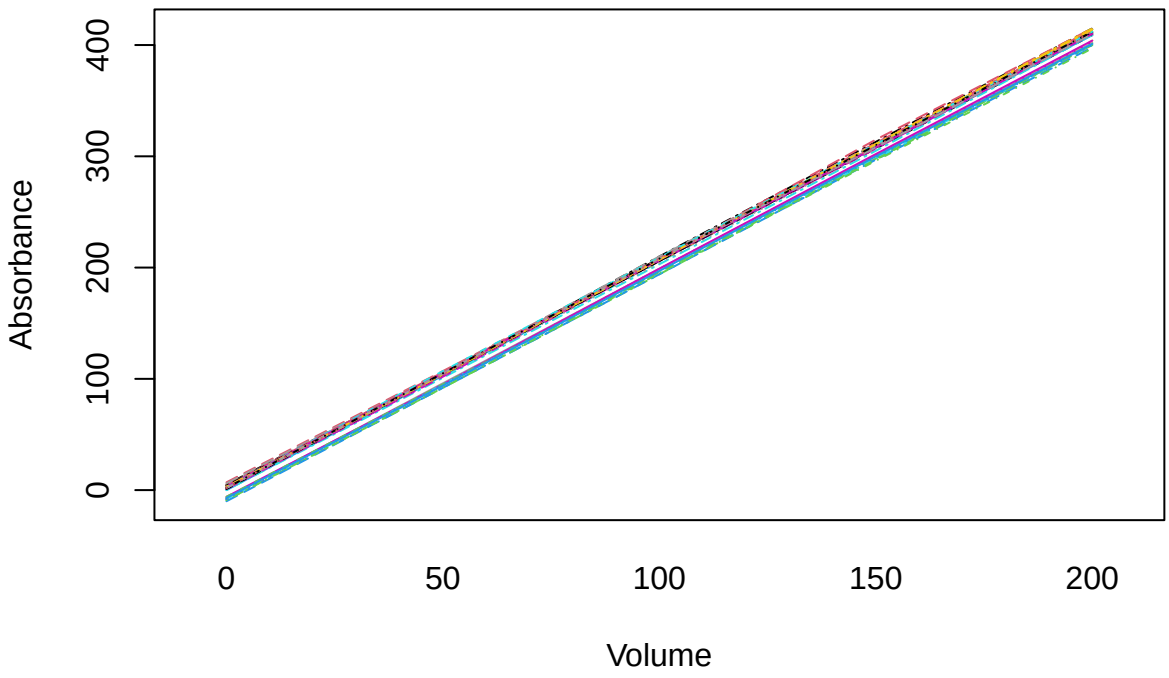
```
# Installing and loading readxl package
# install.packages("readxl")
# Loading
library("readxl")
# xlsx files
calib.dat<- read_excel("calibration.xlsx")
head(calib.dat)
```

```
## # A tibble: 6 x 4
##   Sample Volume Absorbance Repeat
##    <dbl>  <dbl>      <dbl>  <dbl>
## 1      1      0          1      1
## 2      2      0          4      1
## 3      3      0          3      1
## 4      4      0          4      1
## 5      5      0         -9      1
## 6      6      0          3      1
```

(a) Make a spaghetti plot to graphically investigate the relationship between absorbance and volume in a sample specific manner.

Hint: for each sample, you will get two spaghetti curves.

```
calib.dat$sample_repeat_concatenated = paste(calib.dat$Sample, calib.dat$Repeat)
interaction.plot(x.factor = calib.dat$Volume, trace.factor = calib.dat$sample_repeat_concatenated,
                 response = calib.dat$Absorbance,
                 xlab = "Volume", ylab = "Absorbance", col = c(1:44), legend = F)
```




```
# Method 2
#library(ggplot2)
#p = ggplot(data = calib.dat, aes(x = Volume, y=Absorbance, group=sample_repeat_concatenated))
#p+geom_line()
```

(b) Make a mean plot to show the average absorbance for 22 samples over every different volume values as that in Figure 11.3 in the textbook, describe what you can observe from the plot.

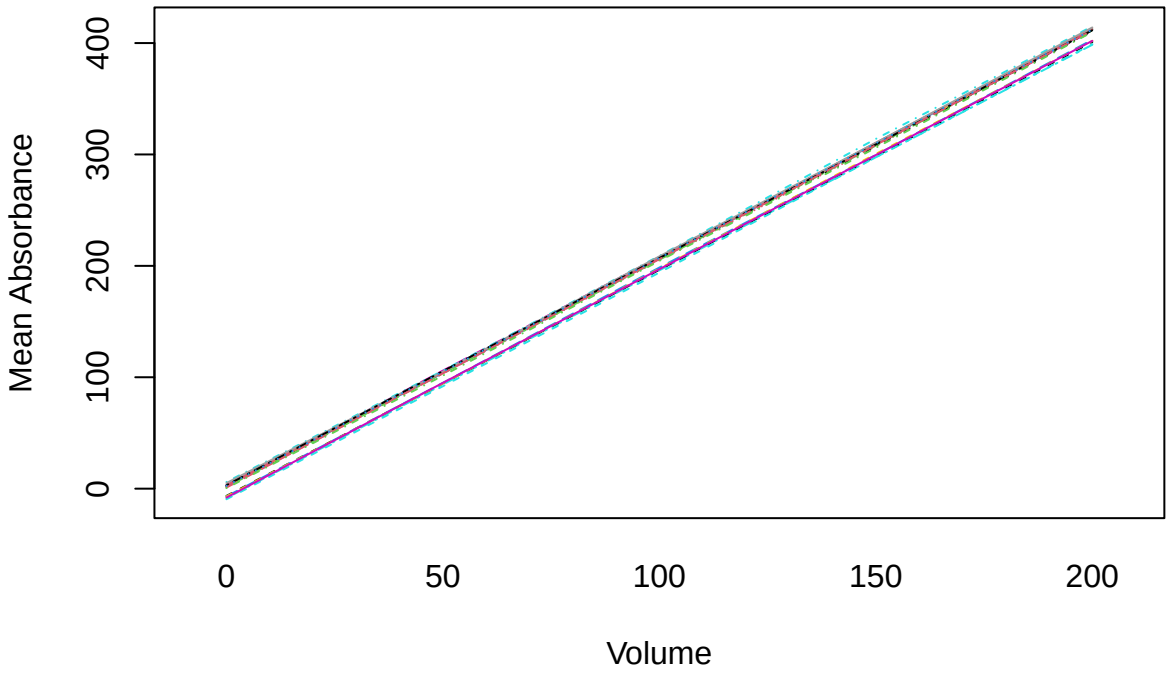
Hint: You can treat 22 samples as 22 different methods, and each sample has two subjects (repeats).

```
agg = aggregate(calib.dat$Absorbance, list(calib.dat$Sample, calib.dat$Volume), FUN=mean)
head(agg)
```

```
##   Group.1 Group.2    x
## 1       1       0  2.0
## 2       2       0  3.0
## 3       3       0  2.5
## 4       4       0  3.0
## 5       5       0 -8.5
## 6       6       0  3.0
```

```
average = agg$x

interaction.plot(x.factor = agg$Group.2, trace.factor = agg$Group.1,
                 response = agg$x,
                 xlab = "Volume", ylab = "Mean Absorbance", col = c(1:22), legend = F)
```



(c) Assuming all observations are independent. Using the normal linear model with different intercepts and different slopes for the 22 sample groups (each sample has two replications), perform a naive or pooled data analysis as that in Table 11.3 in the textbook.

We have the model: $E(Y_{ijk}) = \alpha_i + \beta_i g_k + \epsilon_{ijk}$

```
###Model: different slopes;
fit<-lm(Absorbance~Sample+Sample*Volume, data=calib.dat)
summary(fit)
```

```
##
## Call:
## lm(formula = Absorbance ~ Sample + Sample * Volume, data = calib.dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.873  -4.362   1.684   3.233   8.732
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.8194805  1.1822626   1.539   0.1253
## Sample        -0.1977414  0.0900159  -2.197   0.0291 *
## Volume         2.0442208  0.0096531 211.768   <2e-16 ***
## Sample:Volume  0.0001982  0.0007350   0.270   0.7877
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.89 on 216 degrees of freedom
## Multiple R-squared:  0.9989, Adjusted R-squared:  0.9989
## F-statistic: 6.421e+04 on 3 and 216 DF,  p-value: < 2.2e-16
```

(d) Perform a two-stage or data reduction analysis as that in Tables 11.4-6 in the textbook. That is, fit a linear model for each replication in a sample to obtain the intercept and slope, then summarize both in ANOVA to test if the 22 sample groups have the same intercept and slope.

```
### Two stage analysis in Tables 11.4-6;
options(digits=4)
# for Repeat=1
beta.est1<-matrix(0, 22, 2)
for (i in 1:22){
  x = c(0,50,100,150,200)
  y = subset(calib.dat$Absorbance, (calib.dat$Sample==i)&(calib.dat$Repeat==1))
  fit.lm<-lm(y~x)
  summary(fit.lm)
  beta.est1[i, ]<-fit.lm$coef
}

# for Repeat=2
beta.est2<-matrix(0, 22, 2)
for (i in 1:22){
  x = c(0,50,100,150,200)
  y = subset(calib.dat$Absorbance, (calib.dat$Sample==i)&(calib.dat$Repeat==2))
  fit.lm<-lm(y~x)
  summary(fit.lm)
  beta.est2[i, ]<-fit.lm$coef
}

## two stage for Repeat=1
twostage.data.1<-data.frame(Sample=seq(1:22),
                            group=c(1:22),beta0=beta.est1[,1], beta1=beta.est1[,2])
## two stage for Repeat=2
twostage.data.2<-data.frame(Sample=seq(1:22),
                            group=c(1:22),beta0=beta.est2[,1], beta1=beta.est2[,2])
#####################################Repeat=1#########################################
###Table 11.4 -> intercept and slope estimate
beta.est1
```

```
##       [,1]  [,2]
##  [1,]  1.6 2.038
##  [2,]  2.8 2.040
##  [3,]  2.4 2.056
##  [4,]  3.0 2.036
##  [5,] -9.0 2.032
##  [6,]  4.0 2.044
##  [7,]  2.0 2.058
##  [8,]  2.4 2.050
##  [9,] -6.6 2.032
## [10,]  1.4 2.058
## [11,]  0.4 2.056
```

```
## [12,] -7.8 2.052
## [13,]  4.0 2.056
## [14,]  3.6 2.042
## [15,] -8.0 2.042
## [16,]  2.4 2.058
## [17,]  2.4 2.038
## [18,]  1.2 2.058
## [19,] -0.8 2.044
## [20,]  1.4 2.050
## [21,] -9.8 2.048
## [22,] -8.0 2.042
```

```
##Analysis of variance of intercept estimates in Table 11.5;

interc.fit<-lm(beta0~group, data=twostage.data.1)
print(anova(interc.fit))
```

```
## Analysis of Variance Table
##
## Response: beta0
##           Df Sum Sq Mean Sq F value Pr(>F)
## group      1     49    49.5    2.21   0.15
## Residuals 20    447    22.4
```

```
print(summary(interc.fit))
```

```
##
## Call:
## lm(formula = beta0 ~ group, data = twostage.data.1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9.86  -3.68   1.64   3.41   5.04
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.036      2.087    0.98     0.34
## group         -0.236      0.159   -1.49     0.15
##
## Residual standard error: 4.73 on 20 degrees of freedom
## Multiple R-squared:  0.0996, Adjusted R-squared:  0.0546
## F-statistic: 2.21 on 1 and 20 DF,  p-value: 0.153
```

```
##Analysis of variance of slope estimates in Table 11.6;

slope.fit<-lm(beta1~group, data=twostage.data.1)
print(anova(slope.fit))
```

```
## Analysis of Variance Table
##
## Response: beta1
##           Df   Sum Sq  Mean Sq F value Pr(>F)
```

```
## group      1 0.000073 7.34e-05    0.93   0.35
## Residuals 20 0.001572 7.86e-05
```

```
print(summary(slope.fit))
```

```
##
## Call:
## lm(formula = beta1 ~ group, data = twostage.data.1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.01410 -0.00582 -0.00139  0.00917  0.01248
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.043506   0.003913  522.26   <2e-16 ***
## group       0.000288   0.000298    0.97     0.35
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.00887 on 20 degrees of freedom
## Multiple R-squared:  0.0446, Adjusted R-squared:  -0.00314
## F-statistic: 0.934 on 1 and 20 DF,  p-value: 0.345
```

```
## Summary statistics from stage 1, see lecture notes 12, p.21
## MARGIN = 1 (for row);MARGIN = 2 (for column)
mean.beta<-apply(beta.est1, 2, mean)
var.beta<-apply(beta.est1, 2, var)
sd.beta<-apply(beta.est1, 2, sd)

####################################Repeat=2#########################################
###Table 11.4 -> intercept and slope estimate
beta.est2
```

```
##        [,1]  [,2]
##  [1,]   2.2 2.044
##  [2,]   0.6 2.052
##  [3,]   2.0 2.046
##  [4,]   1.6 2.060
##  [5,]  -7.4 2.040
##  [6,]   3.2 2.034
##  [7,]   2.8 2.038
##  [8,]   2.0 2.050
##  [9,]  -7.4 2.044
## [10,]   3.4 2.042
## [11,]   1.8 2.042
## [12,]  -6.4 2.046
## [13,]   5.6 2.048
## [14,]   1.4 2.056
## [15,]  -6.6 2.054
## [16,]   5.4 2.036
## [17,]   3.8 2.044
## [18,]   0.6 2.046
```

```
## [19,]   0.4 2.050
## [20,]   3.6 2.046
## [21,] -10.0 2.042
## [22,]  -7.6 2.056
```

```
##Analysis of variance of intercept estimates in Table 11.5;

interc.fit<-lm(beta0~group, data=twostage.data.2)
print(anova(interc.fit))
```

```
## Analysis of Variance Table
##
## Response: beta0
##           Df Sum Sq Mean Sq F value Pr(>F)
## group      1     22    22.4    0.96   0.34
## Residuals 20    467    23.4
```

```
print(summary(interc.fit))
```

```
##
## Call:
## lm(formula = beta0 ~ group, data = twostage.data.2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8.26  -4.45   1.75   2.49   6.34
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.603      2.133    0.75     0.46
## group         -0.159      0.162   -0.98     0.34
##
## Residual standard error: 4.83 on 20 degrees of freedom
## Multiple R-squared:  0.0458, Adjusted R-squared:  -0.00193
## F-statistic: 0.96 on 1 and 20 DF,  p-value: 0.339
```

```
##Analysis of variance of slope estimates in Table 11.6;

slope.fit<-lm(beta1~group, data=twostage.data.2)
print(anova(slope.fit))
```

```
## Analysis of Variance Table
##
## Response: beta1
##           Df   Sum Sq  Mean Sq F value Pr(>F)
## group      1 0.000010 1.04e-05    0.23   0.64
## Residuals 20 0.000925 4.62e-05
```

```
print(summary(slope.fit))
```

```
##
```

```
## Call:
## lm(formula = beta1 ~ group, data = twostage.data.2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.011586 -0.004101 -0.000965  0.003899  0.014631
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.044935   0.003001  681.33   <2e-16 ***
## group       0.000108   0.000229    0.47     0.64
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0068 on 20 degrees of freedom
## Multiple R-squared:  0.0111, Adjusted R-squared:  -0.0383
## F-statistic: 0.225 on 1 and 20 DF,  p-value: 0.64
```

(e) Use the results from the two-stage analysis, comment on if the results are consistent with what you have observed from part (a), then state your conclusion. You can use conduct some Welch Two Sample t-test to compare different pairs of samples, for example, sample 1 and sample 2, sample 1 and sample 3, respectively (The Welch Two Sample t-test uses the Satterthwaite-Welch approximation to the degrees of freedom).

As we observed from part (a), the relationship between absorbance and volume among 22 samples are quite close / they are the same. From the ANOVA table, we determine that the null hypothesis is NOT rejected for equality of intercepts and also we do not reject the null hypothesis for equality of slops among 22 samples. This highlights the results are consistent with what we observed from part (a). Let us confirm this finding by using Welch Two Sample t-test to compare Sample 1 and Sample 2 and make another comparison between Sample 11 and Sample 19. We can see p-value is approximately equal to 1, which means fail to reject. The code-generated result is demonstrated below:

```
t.test(subset(calib.dat$Absorbance, (calib.dat$Sample==1)&(calib.dat$Repeat==1)),
       subset(calib.dat$Absorbance, (calib.dat$Sample==2)&(calib.dat$Repeat==1)),paired=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  subset(calib.dat$Absorbance, (calib.dat$Sample == 1) & (calib.dat$Repeat == 1)) and subset(cal
## t = -0.014, df = 8, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -236.5  233.7
## sample estimates:
## mean of x mean of y
##     205.4     206.8
```

```
t.test(subset(calib.dat$Absorbance, (calib.dat$Sample==11)&(calib.dat$Repeat==1)),
       subset(calib.dat$Absorbance, (calib.dat$Sample==19)&(calib.dat$Repeat==1)),paired=FALSE)
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  subset(calib.dat$Absorbance, (calib.dat$Sample == 11) & (calib.dat$Repeat == 1)) and subset(ca
## t = 0.023, df = 8, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -234.0  238.8
## sample estimates:
## mean of x mean of y
##     206.0     203.6
```

(f) Assuming a unstructured covariance matrix, use R package **geepack** to analyze the data, provide the inference about the fixed effects with model-based standard errors (since this package cannot produce sandwich robust standard errors).

```
############## For Repeat = 1 ############
library(geepack)
# id: a vector which identifies the clusters. The length of `id' should be the same
# as the number of observations. Data are assumed to be sorted so that observations
# on each cluster appear as contiguous rows in data. If data is not sorted this way,
# the function will not identify the clusters correctly. If data is not sorted this
# way, a warning will be issued.

calib.dat.repeat1 = subset(calib.dat, Repeat == 1)
calib.dat.repeat1$ID <- as.numeric(gsub(" ", "", calib.dat.repeat1$sample_repeat_concatenated))
calib.dat.repeat1$Sample = as.factor(calib.dat.repeat1$Sample)

calib.dat.repeat1 = calib.dat.repeat1[order(calib.dat.repeat1$ID),]

geeuns1 = geeglm(Absorbance~Volume*Sample, data = calib.dat.repeat1,
                 family = gaussian, id = ID, corstr = "unstructured")

summary(geeuns1)
```

```
##
## Call:
## geeglm(formula = Absorbance ~ Volume * Sample, family = gaussian,
##     data = calib.dat.repeat1, id = ID, corstr = "unstructured")
##
##  Coefficients:
##                  Estimate   Std.err     Wald Pr(>|W|)
## (Intercept)      1.58e+00  3.60e-05 1.92e+09   <2e-16 ***
## Volume           2.04e+00  2.64e-07 5.96e+13   <2e-16 ***
## Sample2          1.17e+00  6.93e-05 2.84e+08   <2e-16 ***
## Sample3          7.23e-01  1.18e-04 3.77e+07   <2e-16 ***
## Sample4          1.44e+00  4.21e-05 1.17e+09   <2e-16 ***
## Sample5         -1.03e+01  2.98e-04 1.20e+09   <2e-16 ***
## Sample6          2.41e+00  4.21e-05 3.29e+09   <2e-16 ***
## Sample7          4.90e-01  9.45e-05 2.69e+07   <2e-16 ***
## Sample8          9.10e-01  1.03e-04 7.80e+07   <2e-16 ***
## Sample9         -8.38e+00  2.46e-04 1.16e+09   <2e-16 ***
## Sample10        -2.25e-01  5.87e-05 1.47e+07   <2e-16 ***
## Sample11        -1.28e+00  1.18e-04 1.18e+08   <2e-16 ***
## Sample12        -9.32e+00  6.49e-05 2.06e+10   <2e-16 ***
## Sample13         2.87e+00  5.33e-04 2.90e+07   <2e-16 ***
```

```
## Sample14         2.10e+00  1.02e-04 4.27e+08   <2e-16 ***
## Sample15        -9.52e+00  7.48e-05 1.62e+10   <2e-16 ***
## Sample16         8.62e-01  6.81e-05 1.61e+08   <2e-16 ***
## Sample17         6.57e-01  2.03e-04 1.05e+07   <2e-16 ***
## Sample18        -5.98e-01  2.61e-04 5.24e+06   <2e-16 ***
## Sample19        -2.47e+00  1.16e-04 4.51e+08   <2e-16 ***
## Sample20        -5.99e-02  1.50e-04 1.60e+05   <2e-16 ***
## Sample21        -1.14e+01  4.14e-05 7.54e+10   <2e-16 ***
## Sample22        -9.95e+00  4.45e-04 4.99e+08   <2e-16 ***
## Volume:Sample2   3.11e-03  8.60e-07 1.31e+07   <2e-16 ***
## Volume:Sample3   1.97e-02  1.27e-06 2.41e+08   <2e-16 ***
## Volume:Sample4  -1.23e-03  4.05e-07 9.19e+06   <2e-16 ***
## Volume:Sample5  -7.18e-03  1.98e-06 1.32e+07   <2e-16 ***
## Volume:Sample6   6.15e-03  4.05e-07 2.31e+08   <2e-16 ***
## Volume:Sample7   2.09e-02  3.10e-07 4.56e+09   <2e-16 ***
## Volume:Sample8   1.17e-02  9.17e-07 1.62e+08   <2e-16 ***
## Volume:Sample9  -4.53e-03  1.56e-06 8.49e+06   <2e-16 ***
## Volume:Sample10  2.19e-02  1.13e-06 3.75e+08   <2e-16 ***
## Volume:Sample11  1.97e-02  1.27e-06 2.41e+08   <2e-16 ***
## Volume:Sample12  1.32e-02  7.94e-07 2.76e+08   <2e-16 ***
## Volume:Sample13  1.65e-02  2.79e-06 3.48e+07   <2e-16 ***
## Volume:Sample14  4.27e-03  6.14e-07 4.85e+07   <2e-16 ***
## Volume:Sample15  4.60e-03  3.01e-07 2.33e+08   <2e-16 ***
## Volume:Sample16  2.12e-02  6.29e-07 1.14e+09   <2e-16 ***
## Volume:Sample17  1.29e-03  1.11e-06 1.34e+06   <2e-16 ***
## Volume:Sample18  2.19e-02  1.92e-06 1.31e+08   <2e-16 ***
## Volume:Sample19  7.12e-03  9.58e-07 5.53e+07   <2e-16 ***
## Volume:Sample20  1.29e-02  4.25e-07 9.19e+08   <2e-16 ***
## Volume:Sample21  1.14e-02  6.83e-07 2.78e+08   <2e-16 ***
## Volume:Sample22  6.92e-03  2.95e-06 5.49e+06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = unstructured
## Estimated Scale Parameters:
##
##             Estimate Std.err
## (Intercept)    0.627  0.0843
##   Link = identity
##
## Estimated Correlation Parameters:
##           Estimate Std.err
## alpha.1:2  -0.6217   0.168
## alpha.1:3  -0.7705   0.205
## alpha.1:4  -0.4887   0.143
## alpha.1:5   0.7690   0.125
## alpha.2:3   0.2679   0.198
## alpha.2:4   0.0728   0.149
## alpha.2:5  -0.2618   0.143
## alpha.3:4  -0.0179   0.178
## alpha.3:5  -0.3341   0.187
## alpha.4:5  -0.9313   0.226
## Number of clusters:   22  Maximum cluster size: 5
```

```r
############# For Repeat = 2 ############
calib.dat.repeat2 = subset(calib.dat, Repeat == 2)
calib.dat.repeat2$ID <- as.numeric(gsub(" ", "", calib.dat.repeat2$sample_repeat_concatenated))
calib.dat.repeat2$Sample = as.factor(calib.dat.repeat2$Sample)

calib.dat.repeat2 = calib.dat.repeat2[order(calib.dat.repeat2$ID),]

geeuns2 = geeglm(Absorbance~Volume*Sample, data = calib.dat.repeat2,
                 family = gaussian, id = ID, corstr = "unstructured")

summary(geeuns2)
```

```
##
## Call:
## geeglm(formula = Absorbance ~ Volume * Sample, family = gaussian,
##     data = calib.dat.repeat2, id = ID, corstr = "unstructured")
##
##  Coefficients:
##                   Estimate   Std.err     Wald Pr(>|W|)
## (Intercept)      2.10e+00  1.39e-03 2.28e+06   <2e-16 ***
## Volume           2.04e+00  8.92e-06 5.25e+10   <2e-16 ***
## Sample2         -2.33e+00  1.78e-03 1.72e+06   <2e-16 ***
## Sample3          2.86e-01  1.58e-03 3.30e+04   <2e-16 ***
## Sample4         -2.10e-01  1.65e-03 1.63e+04   <2e-16 ***
## Sample5         -9.53e+00  1.84e-03 2.68e+07   <2e-16 ***
## Sample6          1.18e+00  1.39e-03 7.19e+05   <2e-16 ***
## Sample7          1.07e+00  1.48e-03 5.28e+05   <2e-16 ***
## Sample8          6.10e-01  2.09e-03 8.51e+04   <2e-16 ***
## Sample9         -8.89e+00  2.20e-03 1.63e+07   <2e-16 ***
## Sample10         1.39e+00  1.80e-03 6.04e+05   <2e-16 ***
## Sample11        -4.91e-01  1.41e-03 1.21e+05   <2e-16 ***
## Sample12        -8.15e+00  1.46e-03 3.09e+07   <2e-16 ***
## Sample13         3.34e+00  2.04e-03 2.69e+06   <2e-16 ***
## Sample14        -7.81e-01  1.94e-03 1.62e+05   <2e-16 ***
## Sample15        -7.84e+00  2.97e-03 6.96e+06   <2e-16 ***
## Sample16         4.42e+00  3.78e-03 1.37e+06   <2e-16 ***
## Sample17         1.57e+00  1.40e-03 1.25e+06   <2e-16 ***
## Sample18        -1.15e+00  1.46e-03 6.12e+05   <2e-16 ***
## Sample19        -9.58e-01  3.10e-03 9.54e+04   <2e-16 ***
## Sample20         1.17e+00  1.45e-03 6.51e+05   <2e-16 ***
## Sample21        -1.20e+01  1.39e-03 7.49e+07   <2e-16 ***
## Sample22        -9.10e+00  1.46e-03 3.87e+07   <2e-16 ***
## Volume:Sample2   1.21e-02  1.23e-05 9.58e+05   <2e-16 ***
## Volume:Sample3  -5.16e-04  9.80e-06 2.77e+03   <2e-16 ***
## Volume:Sample4   1.55e-02  1.11e-05 1.93e+06   <2e-16 ***
## Volume:Sample5  -4.23e-03  1.18e-05 1.29e+05   <2e-16 ***
## Volume:Sample6  -1.12e-02  9.01e-06 1.56e+06   <2e-16 ***
## Volume:Sample7  -9.51e-03  1.02e-05 8.62e+05   <2e-16 ***
## Volume:Sample8   1.06e-03  1.20e-05 7.81e+03   <2e-16 ***
## Volume:Sample9  -2.93e-03  1.39e-05 4.45e+04   <2e-16 ***
## Volume:Sample10 -2.25e-03  1.18e-05 3.62e+04   <2e-16 ***
## Volume:Sample11 -2.14e-03  8.99e-06 5.66e+04   <2e-16 ***
## Volume:Sample12 -3.55e-04  9.64e-06 1.35e+03   <2e-16 ***
```

```
## Volume:Sample13  7.46e-03  1.58e-05 2.24e+05   <2e-16 ***
## Volume:Sample14  1.08e-02  1.17e-05 8.60e+05   <2e-16 ***
## Volume:Sample15  5.50e-03  1.82e-05 9.09e+04   <2e-16 ***
## Volume:Sample16 -1.52e-02  2.17e-05 4.91e+05   <2e-16 ***
## Volume:Sample17  1.61e-04  9.00e-06 3.20e+02   <2e-16 ***
## Volume:Sample18 -3.55e-04  9.64e-06 1.35e+03   <2e-16 ***
## Volume:Sample19  9.00e-04  1.81e-05 2.47e+03   <2e-16 ***
## Volume:Sample20  2.42e-03  9.21e-06 6.87e+04   <2e-16 ***
## Volume:Sample21 -2.09e-03  8.92e-06 5.51e+04   <2e-16 ***
## Volume:Sample22  8.08e-03  9.06e-06 7.95e+05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = unstructured
## Estimated Scale Parameters:
##
##             Estimate Std.err
## (Intercept)    0.697   0.147
##   Link = identity
##
## Estimated Correlation Parameters:
##           Estimate Std.err
## alpha.1:2   0.1570   0.168
## alpha.1:3  -0.8297   0.196
## alpha.1:4  -0.1895   0.289
## alpha.1:5   0.5647   0.293
## alpha.2:3   0.1090   0.209
## alpha.2:4  -0.1121   0.208
## alpha.2:5   0.1074   0.154
## alpha.3:4  -0.3603   0.306
## alpha.3:5  -0.0424   0.244
## alpha.4:5  -0.9159   0.167
## Number of clusters:   22  Maximum cluster size: 5
```

(g) Determine which model-(1) random intercept model,(2) random slope but common intercept, or (3) random intercept plus random slope model-is most appropriate for these data. Use hypothesis testing with a significance level of 0.05 to make your decision. Model (3) is a linear mixed model given by

$$y = \beta_0 + \beta x + \sum_{j=1}^{m}\sum_{k=1}^{2} \delta_{0,jk} z_{jk} + \sum_{j=1}^{m}\sum_{k=1}^{2} \delta_{1,jk} z_{jk} x + \varepsilon,$$

where z_{jk} denotes the indicator variable for the kth replication in the jth cluster (sample), $\delta_{0,jk}$ is the *random intercept* term for the kth replication in the jth sample with $\delta_{0,jk} \sim N(0, \sigma^2_{\delta_0})$, $\delta_{1,jk}$ is the *random slope* term for the kth replication in the jth sample with $\delta_{1,jk} \sim N(0, \sigma^2_{\delta_1})$, $\varepsilon \sim N(0, \sigma^2_{\varepsilon})$.

Hint: When comparing two models, for example, (3) vs. (1), treat model (3) as a full model, model (1) as a reduced model, let $\ell_{\text{REML}^{\text{Full}}}$ denote the restricted log-likelihood for the full model, $\ell_{\text{REML}^{\text{Red}}}$ for the reduced model. The REML likelihood ratio test statistic is

$$\hat{\lambda} = -2(\ell^{\text{Red}}_{\text{REML}} - \ell^{\text{Full}}_{\text{REML}}).$$

In the full model, we assume unstructured variance structure for random effects and errors are iid, $\hat{\lambda}$ follows a χ^2 distribution with df=the difference of numbers of parameters in the two models.

Compare model 1 (random intercept only) with model 3 and then compare model 2 (random slope only) with model 3, using REMLRT

```r
library(nlme)
```

```
##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse
```

```r
calib.dat$ID <- as.numeric(gsub(" ", "", calib.dat$sample_repeat_concatenated))
calib.dat = calib.dat[order(calib.dat$ID),]
## linear mixed model with random intercept only,
## Results are the same as that of SAS with REML and model based se.
rndinter.reml = lme(Absorbance~Sample+Repeat+Volume, data = calib.dat, method="REML",
                    random=~1|ID)
summary(rndinter.reml)
```

```
## Linear mixed-effects model fit by REML
##   Data: calib.dat
##   AIC BIC logLik
##   876 896   -432
##
## Random effects:
##  Formula: ~1 | ID
##         (Intercept) Residual
## StdDev:         4.9     1.06
##
## Fixed effects:  Absorbance ~ Sample + Repeat + Volume
##              Value Std.Error  DF t-value p-value
## (Intercept)  1.005     2.705 175       0   0.711
## Sample      -0.178     0.117  41      -2   0.136
## Repeat       0.391     1.483  41       0   0.793
## Volume       2.046     0.001 175    2018   0.000
##  Correlation:
##        (Intr) Sample Repeat
## Sample -0.497
## Repeat -0.822  0.000
## Volume -0.037  0.000  0.000
##
## Standardized Within-Group Residuals:
##       Min        Q1       Med        Q3       Max
## -2.20e+00 -6.72e-01  1.52e-05  5.86e-01  2.40e+00
##
## Number of Observations: 220
## Number of Groups: 44
```

```r
## linear mixed model with random slope only
rndslope.reml = lme(Absorbance~Sample+Repeat+Volume, data = calib.dat, method="REML",
                    random=~0+Repeat|ID)
summary(rndslope.reml)
```

```
## Linear mixed-effects model fit by REML
```

```
##   Data: calib.dat
##   AIC BIC logLik
##   887 907   -437
##
## Random effects:
##  Formula: ~0 + Repeat | ID
##         Repeat Residual
## StdDev:   3.95     1.06
##
## Fixed effects:  Absorbance ~ Sample + Repeat + Volume
##              Value Std.Error  DF t-value p-value
## (Intercept)  1.209     2.759 175       0   0.662
## Sample      -0.196     0.119  41      -2   0.109
## Repeat       0.391     1.887  41       0   0.837
## Volume       2.046     0.001 175    2017   0.000
##  Correlation:
##        (Intr) Sample Repeat
## Sample -0.498
## Repeat -0.822  0.000
## Volume -0.037  0.000  0.000
##
## Standardized Within-Group Residuals:
##      Min       Q1      Med       Q3      Max
## -2.23434 -0.65272 -0.00265  0.58822  2.39525
##
## Number of Observations: 220
## Number of Groups: 44
```

```
## A comparable model should be a linear mixed model with both random intercept and slope,
## the results are the same as those by Table11_7.SAS with "reml" and "empirical" stderr  method;
rndeff.reml = lme(Absorbance~Sample+Repeat+Volume, data = calib.dat, method="REML",
                  random=~1+Repeat|ID)
summary(rndeff.reml)
```

```
## Linear mixed-effects model fit by REML
##   Data: calib.dat
##   AIC BIC logLik
##   880 907   -432
##
## Random effects:
##  Formula: ~1 + Repeat | ID
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev   Corr
## (Intercept) 4.896548 (Intr)
## Repeat      0.000125 0.001
## Residual    1.063484
##
## Fixed effects:  Absorbance ~ Sample + Repeat + Volume
##              Value Std.Error  DF t-value p-value
## (Intercept)  1.005     2.705 175       0   0.711
## Sample      -0.178     0.117  41      -2   0.136
## Repeat       0.391     1.483  41       0   0.793
## Volume       2.047     0.001 175    2018   0.000
##  Correlation:
```

```
##        (Intr) Sample Repeat
## Sample -0.497
## Repeat -0.822  0.000
## Volume -0.037  0.000  0.000
##
## Standardized Within-Group Residuals:
##       Min        Q1       Med        Q3       Max
## -2.20e+00 -6.72e-01  1.52e-05  5.86e-01  2.40e+00
##
## Number of Observations: 220
## Number of Groups: 44
```

```
a = logLik(rndeff.reml,REML = TRUE)
b = logLik(rndinter.reml,REML = TRUE)
c = logLik(rndslope.reml,REML = TRUE)
1-pchisq(-2*(b-a), df=2)
```

```
## 'log Lik.' 1 (df=6)
```

```
1-pchisq(-2*(c-a), df=2)
```

```
## 'log Lik.' 0.00391 (df=6)
```

(h) Expression model (3) in matrix notation, we get

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\delta} + \boldsymbol{\epsilon}$$

satisfying

$$\text{Var}(\boldsymbol{y}) = \boldsymbol{V} = \text{Var}(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\delta} + \boldsymbol{\varepsilon}) = \boldsymbol{Z}\text{Var}(\boldsymbol{\delta})\boldsymbol{Z}^\top + \text{Var}(\boldsymbol{\varepsilon}) = \boldsymbol{Z}\boldsymbol{D}\boldsymbol{Z}^\top + \boldsymbol{S},$$

where

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{L}_{n11} & \boldsymbol{x}_{n11} \\ \boldsymbol{L}_{n12} & \boldsymbol{x}_{n12} \\ \boldsymbol{L}_{n21} & \boldsymbol{x}_{n11} \\ \boldsymbol{L}_{n22} & \boldsymbol{x}_{n12} \\ \vdots & \vdots \\ \boldsymbol{L}_{n211} & \boldsymbol{x}_{n211} \\ \boldsymbol{L}_{n212} & \boldsymbol{x}_{n212} \\ \boldsymbol{L}_{n221} & \boldsymbol{x}_{n221} \\ \boldsymbol{L}_{n222} & \boldsymbol{x}_{n222} \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_{n11} \\ \boldsymbol{y}_{n12} \\ \boldsymbol{y}_{n21} \\ \boldsymbol{y}_{n22} \\ \vdots \\ \boldsymbol{y}_{n211} \\ \boldsymbol{y}_{n212} \\ \boldsymbol{y}_{n221} \\ \boldsymbol{y}_{n222} \end{bmatrix}, \quad \boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{\delta}_{0,11} \\ \boldsymbol{\delta}_{0,12} \\ \vdots \\ \boldsymbol{\delta}_{0,211} \\ \boldsymbol{\delta}_{0,212} \\ \boldsymbol{\delta}_{1,11} \\ \boldsymbol{\delta}_{1,12} \\ \vdots \\ \boldsymbol{\delta}_{1,211} \\ \boldsymbol{\delta}_{1,212} \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \boldsymbol{\varepsilon}_{n11} \\ \boldsymbol{\varepsilon}_{n12} \\ \boldsymbol{\varepsilon}_{n21} \\ \boldsymbol{\varepsilon}_{n22} \\ \vdots \\ \boldsymbol{\varepsilon}_{n211} \\ \boldsymbol{\varepsilon}_{n212} \\ \boldsymbol{\varepsilon}_{n221} \\ \boldsymbol{\varepsilon}_{n222} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix},$$

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{L}_{n11} & \boldsymbol{0}_{n11} & \boldsymbol{0}_{n11} & \boldsymbol{0}_{n11} & \cdots & \boldsymbol{x}_{n11} & \boldsymbol{0}_{n11} & \boldsymbol{0}_{n11} & \boldsymbol{0}_{n11} & \cdots \\ \boldsymbol{0}_{n12} & \boldsymbol{L}_{n12} & \boldsymbol{0}_{n11} & \boldsymbol{0}_{n12} & \cdots & \boldsymbol{0}_{n12} & \boldsymbol{x}_{n12} & \boldsymbol{0}_{n11} & \boldsymbol{0}_{n11} & \cdots \\ \boldsymbol{0}_{n21} & \boldsymbol{0}_{n21} & \boldsymbol{L}_{n21} & \boldsymbol{0}_{n21} & \cdots & \boldsymbol{0}_{n21} & \boldsymbol{0}_{n11} & \boldsymbol{x}_{n21} & \boldsymbol{0}_{n11} & \cdots \\ \boldsymbol{0}_{n22} & \boldsymbol{0}_{n22} & \boldsymbol{0}_{n22} & \boldsymbol{L}_{n22} & \cdots & \boldsymbol{0}_{n22} & \boldsymbol{0}_{n21} & \boldsymbol{0}_{n21} & \boldsymbol{x}_{n22} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}.$$

Based on the model (3) in part (g), estimate the common variance-covariance matrix for each sample in a repeat, for example, sample 1 in repeat 1, which is given by

$$\boldsymbol{V}_{n11} = \boldsymbol{Z}_{n11}\text{Var}(\boldsymbol{\delta}_{n11})\boldsymbol{Z}_{n11}^\top + \text{Var}(\boldsymbol{\varepsilon}_{n11}).$$

Note: The model form for the sample 1 in repeat 1 is

$$\boldsymbol{y}_{n11} = \boldsymbol{X}_{n11}\boldsymbol{\beta} + \boldsymbol{Z}_{n11}\boldsymbol{\delta}_{n11} + \boldsymbol{\epsilon}_{n11},$$

where

$$\boldsymbol{X}_{n11} = [\boldsymbol{L}_{n11}\ \boldsymbol{x}_{n11}], \quad \boldsymbol{Z}_{n11} = [\boldsymbol{L}_{n11}\ \boldsymbol{x}_{n11}],$$

$$\boldsymbol{\beta} = \begin{bmatrix}\beta_0 \\ \beta_1\end{bmatrix}, \quad \boldsymbol{\delta}_{n11} = \begin{bmatrix}\delta_{0,11} \\ \delta_{1,11}\end{bmatrix},$$

$$\boldsymbol{L}_{n11} = \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}, \quad \boldsymbol{x}_{n11} = \begin{bmatrix}0\\50\\100\\150\\200\end{bmatrix}, \quad \boldsymbol{y}_{n11} = \begin{bmatrix}1\\104\\206\\307\\409\end{bmatrix}, \quad \boldsymbol{\varepsilon}_{n11} = \begin{bmatrix}\varepsilon_{n11,1}\\\varepsilon_{n11,2}\\\varepsilon_{n11,3}\\\varepsilon_{n11,4}\\\varepsilon_{n11,5}\end{bmatrix}.$$

Hence, in the expression of

$$\boldsymbol{V}_{n11} = \boldsymbol{Z}_{n11}\mathrm{Var}(\boldsymbol{\delta}_{n11})\boldsymbol{Z}_{n11}^{\top} + \mathrm{Var}(\boldsymbol{\varepsilon}_{n11}),$$

$$\mathrm{Var}(\boldsymbol{\delta}_{n11}) = D_{n11} = \mathrm{Var}\left(\begin{bmatrix}\delta_{0,11}\\\delta_{1,11}\end{bmatrix}\right) = \begin{bmatrix}\sigma^2_{\delta_{0,11}} & \mathrm{Cov}(\delta_{0,11},\delta_{1,11})\\ \mathrm{Cov}(\delta_{0,11},\delta_{1,11}) & \sigma^2_{\delta_{1,11}}\end{bmatrix},$$

$$\mathrm{Var}(\boldsymbol{\varepsilon}_{n11}) = \mathrm{Var}\left(\begin{bmatrix}\varepsilon_{n11,1}\\\varepsilon_{n11,2}\\\varepsilon_{n11,3}\\\varepsilon_{n11,4}\\\varepsilon_{n11,5}\end{bmatrix}\right) = \mathrm{diag}(\sigma^2_\varepsilon,\sigma^2_\varepsilon,\sigma^2_\varepsilon,\sigma^2_\varepsilon,\sigma^2_\varepsilon).$$

Once these variance components are estimated, the variance-covariance is obtained easily. Provide the values of the estimated variance components and the variance-covariance matrix.

riance components are estimated, the variance-covariance is obtained easily. Provide the values of the estimated variance components and the variance-covariance matrix. \end{itemize}

```
# estimate of variance components
rndeff.reml = lme(Absorbance~Sample+Repeat+Volume, data = calib.dat, method="REML",
                  random=~1+Repeat|ID)
summary(rndeff.reml)
```

```
## Linear mixed-effects model fit by REML
##   Data: calib.dat
##   AIC BIC logLik
##   880 907   -432
##
## Random effects:
##  Formula: ~1 + Repeat | ID
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev   Corr
## (Intercept) 4.896548 (Intr)
## Repeat      0.000125 0.001
## Residual    1.063484
##
## Fixed effects:  Absorbance ~ Sample + Repeat + Volume
##               Value Std.Error  DF t-value p-value
## (Intercept)  1.005     2.705 175       0   0.711
```

```
## Sample      -0.178     0.117  41      -2   0.136
## Repeat       0.391     1.483  41       0   0.793
## Volume       2.047     0.001 175    2018   0.000
##  Correlation:
##        (Intr) Sample Repeat
## Sample -0.497
## Repeat -0.822  0.000
## Volume -0.037  0.000  0.000
##
## Standardized Within-Group Residuals:
##       Min        Q1       Med        Q3       Max
## -2.20e+00 -6.72e-01  1.52e-05  5.86e-01  2.40e+00
##
## Number of Observations: 220
## Number of Groups: 44
```

```
# estimate of variance covariance
# V = Z VAR Z^T + VAR(eps)
```

(i) Assuming a random intercept and a random slope model (3) in part (g), use R package `nlme` to fit a linear mixed model, provide the inference about the fixed effects with the REML method and model-based standard errors (since this package cannot produce sandwich robust standard errors). Compare the results with those obtained in parts (c), (d) and (f) with the naive, two-stage and GEE methods respectively.

```
library(nlme)
fm1 <- lme(Absorbance~Sample+Repeat+Volume, data = calib.dat, method="REML",
                  random=~1+Repeat|ID)
summary(fm1)
```

```
## Linear mixed-effects model fit by REML
##   Data: calib.dat
##   AIC BIC logLik
##   880 907   -432
##
## Random effects:
##  Formula: ~1 + Repeat | ID
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev   Corr
## (Intercept) 4.896548 (Intr)
## Repeat      0.000125 0.001
## Residual    1.063484
##
## Fixed effects:  Absorbance ~ Sample + Repeat + Volume
##              Value Std.Error  DF t-value p-value
## (Intercept)  1.005     2.705 175       0   0.711
## Sample      -0.178     0.117  41      -2   0.136
## Repeat       0.391     1.483  41       0   0.793
## Volume       2.047     0.001 175    2018   0.000
##  Correlation:
##        (Intr) Sample Repeat
## Sample -0.497
## Repeat -0.822  0.000
```

```
## Volume -0.037  0.000  0.000
##
## Standardized Within-Group Residuals:
##       Min        Q1       Med        Q3       Max
## -2.20e+00 -6.72e-01  1.52e-05  5.86e-01  2.40e+00
##
## Number of Observations: 220
## Number of Groups: 44
```

I could say that the estimations and standard errors obtained from these three models are quite different. We should know that unlike a random intercept model, a random slope model allows each group line to have a different slope and that means that the random slope model allows the explanatory variable to have a different effect for each group. It allows the relationship between the explanatory variable and the response to be different for each group. I believe this new model allow us to estimate the within variation and also the between variation.

Note: For the following parts (j) and (k), if you can find an R function or R package to calculate the vector of estimated marginal means and conditional means, you don't need report each component in their expressions, just add two columns in the original data file to report the final results of $\hat{E}(\boldsymbol{y})$ and $\hat{E}(\boldsymbol{y}|\boldsymbol{\delta})$. If you cannot find such an R function, then follow the expression details to program your own functions to do the calculations, and report the results in the same way.

(j) Based on the expression for the marginal mean relationship between absorbance and volume given below, using R to compute the vector of estimated marginal means, which is given by

$$\hat{E}(\boldsymbol{y}) = \hat{\boldsymbol{\mu}} = \boldsymbol{X}\boldsymbol{b},$$

where $\boldsymbol{b} = \hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^\top \hat{\boldsymbol{V}}^{-1}\boldsymbol{y}$.

(k) Based on the expression for the conditional mean relationship between absorbance and volume given below, using R to compute the vector of estimated conditional means, which is given by

$$\hat{E}(\boldsymbol{y}|\boldsymbol{\delta}) = \hat{\boldsymbol{\mu}}|\boldsymbol{\delta} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{Z}\hat{\boldsymbol{\delta}},$$

where $\boldsymbol{b} = \hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \hat{\boldsymbol{V}}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^\top \hat{\boldsymbol{V}}^{-1}\boldsymbol{y}$, $\hat{\boldsymbol{\delta}}$ are the predicted values for the random effects, commonly referred to as the BLUP (*best linear unbiased predictors*), given by

$$\hat{\boldsymbol{\delta}} = \hat{\boldsymbol{D}}\boldsymbol{Z}^\top \hat{\boldsymbol{V}}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}),$$

where $\hat{\boldsymbol{V}} = \boldsymbol{Z}\hat{\boldsymbol{D}}\boldsymbol{Z}^\top + \hat{\boldsymbol{S}}$, $\hat{\boldsymbol{D}}$ and $\hat{\boldsymbol{S}}$ are the REML estimates of the variance-covariance matrices, which could be obtained from the output of a statistical software.

Problem 4. [12 marks; (a) 3, (b) 3, (c), 3, (d), 3]

In a teratology experiment, female rats on iron-deficient diets were assigned to four groups. Group 1 received only placebo injections. The other groups received injections of an iron supplement according to various schedules. The rats were made pregnant and then sacrificed after 3 weeks. For each fetus in each rat's litter, the response was whether the fetus was dead. Data also include HB = mother's hemoglobin level. We treat the fetuses in a given litter as a cluster.

The data have two formats, one is the group data set `teratologyGr.txt`, and the other is the ungrouped data set, `teratologyUnGr.txt`, which are available from the D2L under the assignment folder. You can upload the data by

```
Grdata<-read.table(file="teratologyGr.txt", header=TRUE)
head(Grdata)
```

```
##    N  R  HB GRP Litter
## 1 10  1 4.1   1      1
## 2 11  4 3.2   1      2
## 3 12  9 4.7   1      3
## 4  4  4 3.5   1      4
## 5 10 10 3.2   1      5
## 6 11  9 5.9   1      6
```

```
UnGrdata<-read.table(file="teratologyUnGr.txt", header=TRUE)
head(UnGrdata)
```

```
##    HB GRP Litter Response
## 1 4.1   1      1     Dead
## 2 4.1   1      1    Alive
## 3 4.1   1      1    Alive
## 4 4.1   1      1    Alive
## 5 4.1   1      1    Alive
## 6 4.1   1      1    Alive
```

Specifically, the definition of the groups and the names of all the variables are given by

Group 1: placebo; Group 2: iron injections on days 7 and 10; Group 3: iron injections on days 0 and 7; Group 4: iron injections weekly;

In `teratologyGr.txt`, (N, R, HB, GRP, Litter) represents (Number of fetuses in a litter, Number dead in litter, Mother's hemoglobin level, Group number, Litter number).

In `teratologyUnGr.txt`, (HB, GRP, Litter, Response) represents (Mother's hemoglobin level, Group number, Litter number, Response Alive or Dead).

Let y_i denote the number of dead fetuses for the T_i fetuses in litter i. Let π_{it} denote the probability of death for fetus t in litter i. Use the first group as the reference level, let $z_{ig} = 1$ if litter i is in group g and 0 otherwise, $g = 2, 3, 4$.

(a) First, we use the grouped data and ignore the clustering and suppose that y_i is a Binomial(T_i, π_{it}) variate. Use z_{ig} as three covariates only to fit a logistic regression model:

$$\text{logit}(\pi_{it}) = \alpha + \beta_1 z_{i2} + \beta_2 z_{i3} + \beta_3 z_{i4}.$$

Test the goodness-of-fit using statistics X^2 and deviance D, respectively, and provide your conclusion for the model fit, does it fit data well or not?

```
Grdata$GRP = factor(Grdata$GRP)
model = glm(cbind(R,N-R)~GRP, family = binomial(link = "logit"), data = Grdata)
summary(model)
```

```
##
## Call:
## glm(formula = cbind(R, N - R) ~ GRP, family = binomial(link = "logit"),
##     data = Grdata)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -4.430  -0.975  -0.028   1.402   2.783
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.144      0.129    8.85  < 2e-16 ***
## GRP2          -3.323      0.331  -10.04  < 2e-16 ***
## GRP3          -4.476      0.731   -6.12  9.2e-10 ***
## GRP4          -4.130      0.476   -8.67  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 509.43  on 57  degrees of freedom
## Residual deviance: 173.45  on 54  degrees of freedom
## AIC: 252.9
##
## Number of Fisher Scoring iterations: 5
```

```
# GOF
s.p.r.s = sum(resid(model_reduced, "pearson")^2)  #SUM OF PEARSON RESIDUAL SQUARED
s.d.r.s = sum(resid(model_reduced, "deviance")^2) #SUM OF DEVIANCE RESIDUAL SQUARED
1-pchisq(s.p.r.s, df.residual(model))
```

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

```
1-pchisq(s.d.r.s, df.residual(model))
```

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

We do not reject the null hypothesis, which means the model fits the data well.

(b) There maybe inter-litter variability that cannot be accounted for in a binomial model by treatment group alone, because fetuses are more alike within litters than across litters, even within the same treatment group. This can result in overdispersion and make standard errors invalid (too small). Two possible solutions are: GEE and GLIMM to model the dependence within litters. To use GEE [by gee() in library(gee)], need data in ungrouped (binary) format. So use the data `teratologyUnGr.txt` and exchangeable structure as a working correlation matrix, obtain the estimated effects of each treatment group related to the Placebo (Group 1) and the correlation coefficient in the working correlation matrix.

```r
library(gee)
UnGrdata$GRP = as.factor(UnGrdata$GRP)
UnGrdata$Response = as.factor(UnGrdata$Response)
UnGrdata$ID =  seq.int(nrow(UnGrdata))
gee.model = gee((Response=="Dead")~GRP, id = Litter, data = UnGrdata, corstr = "exchangeable",
                family = binomial(link = "logit"))
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27

## running glm to get initial regression estimate

## (Intercept)        GRP2        GRP3        GRP4
##        1.14       -3.32       -4.48       -4.13
```

```r
summary(gee.model)
```

```
##
##  GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
##  gee S-function, version 4.13 modified 98/01/27 (1998)
##
## Model:
##  Link:                      Logit
##  Variance to Mean Relation: Binomial
##  Correlation Structure:     Exchangeable
##
## Call:
## gee(formula = (Response == "Dead") ~ GRP, id = Litter, data = UnGrdata,
##     family = binomial(link = "logit"), corstr = "exchangeable")
##
## Summary of Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7706 -0.1036 -0.0332  0.2294  0.9668
##
##
## Coefficients:
##             Estimate Naive S.E. Naive z Robust S.E. Robust z
## (Intercept)     1.21      0.225    5.40       0.270     4.49
## GRP2           -3.37      0.566   -5.95       0.430    -7.83
## GRP3           -4.58      1.309   -3.50       0.624    -7.35
## GRP4           -4.25      0.853   -4.98       0.605    -7.02
##
## Estimated Scale Parameter:  1.03
## Number of Iterations:  3
##
## Working Correlation
##        [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
##  [1,] 1.000 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185
##  [2,] 0.185 1.000 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185
##  [3,] 0.185 0.185 1.000 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185
##  [4,] 0.185 0.185 0.185 1.000 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185
##  [5,] 0.185 0.185 0.185 0.185 1.000 0.185 0.185 0.185 0.185 0.185 0.185 0.185
##  [6,] 0.185 0.185 0.185 0.185 0.185 1.000 0.185 0.185 0.185 0.185 0.185 0.185
```

```
##  [7,] 0.185 0.185 0.185 0.185 0.185 0.185 1.000 0.185 0.185 0.185 0.185 0.185
##  [8,] 0.185 0.185 0.185 0.185 0.185 0.185 0.185 1.000 0.185 0.185 0.185 0.185
##  [9,] 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 1.000 0.185 0.185 0.185
## [10,] 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 1.000 0.185 0.185
## [11,] 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 1.000 0.185
## [12,] 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 1.000
## [13,] 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185
## [14,] 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185
## [15,] 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185
## [16,] 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185
## [17,] 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185 0.185
##       [,13] [,14] [,15] [,16] [,17]
##  [1,] 0.185 0.185 0.185 0.185 0.185
##  [2,] 0.185 0.185 0.185 0.185 0.185
##  [3,] 0.185 0.185 0.185 0.185 0.185
##  [4,] 0.185 0.185 0.185 0.185 0.185
##  [5,] 0.185 0.185 0.185 0.185 0.185
##  [6,] 0.185 0.185 0.185 0.185 0.185
##  [7,] 0.185 0.185 0.185 0.185 0.185
##  [8,] 0.185 0.185 0.185 0.185 0.185
##  [9,] 0.185 0.185 0.185 0.185 0.185
## [10,] 0.185 0.185 0.185 0.185 0.185
## [11,] 0.185 0.185 0.185 0.185 0.185
## [12,] 0.185 0.185 0.185 0.185 0.185
## [13,] 1.000 0.185 0.185 0.185 0.185
## [14,] 0.185 1.000 0.185 0.185 0.185
## [15,] 0.185 0.185 1.000 0.185 0.185
## [16,] 0.185 0.185 0.185 1.000 0.185
## [17,] 0.185 0.185 0.185 0.185 1.000
```

(c) Use GLIMM to model the dependence within litters,

$$\text{logit}(\pi_{it}) = \alpha + u_i + \beta_1 z_{i2} + \beta_2 z_{i3} + \beta_3 z_{i4},$$

where $\pi_{it} = P(\text{fetus } t \text{ in litter } i \text{ dead})$. Assume $u_i \sim N(0, \sigma^2)$, σ^2 is unknown. To use GLIMM [by glmer() in library(lme4)], data can be in either grouped format or ungrouped (binary) format. So use the data `teratologyGr.txt` to obtain the estimated effects of treatment group, including σ^2. (Use mode-based standard errors of fixed effects for inference).

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'lme4'
```

```
## The following object is masked from 'package:nlme':
##
##     lmList
```

```
glimm = glmer((Response=="Dead")~GRP+(1|Litter), family = binomial, data = UnGrdata)
summary(glimm)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: (Response == "Dead") ~ GRP + (1 | Litter)
##    Data: UnGrdata
##
##      AIC      BIC   logLik deviance df.resid
##      446      468     -218      436      602
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.821 -0.267 -0.116  0.243  4.782
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Litter (Intercept) 2.28     1.51
## Number of obs: 607, groups:  Litter, 58
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.809      0.362    5.00  5.6e-07 ***
## GRP2          -4.540      0.735   -6.18  6.4e-10 ***
## GRP3          -5.883      1.175   -5.01  5.6e-07 ***
## GRP4          -5.606      0.908   -6.18  6.5e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr) GRP2   GRP3
## GRP2 -0.562
## GRP3 -0.373  0.235
## GRP4 -0.496  0.316  0.221
```

(d) Put your estimates of the regression coefficients and their standard errors in the following table, and compare the three approaches shown in parts (a), (b) and (c) and comment on the validity of these different methods. [For GEE, use sandwich robust standard errors; for ML and GLIMM, use model-based standard errors].

	Binomial ML		GEE		GLMM	
	Estimate	(M-based) SE	Estimate	(Robust) SE	Estimate	(M-based) SE
(Intercept)	1.144	0.129	1.21	0.270	1.809	0.362
GRP2	-3.323	0.331	-3.37	0.430	-4.540	0.735
GRP3	-4.476	0.731	-4.58	0.624	-5.883	1.175
GRP4	-4.130	0.476	-4.25	0.605	-5.606	0.908

I could say that the estimations and standard errors obtained from these three models are quite similar. It is important to take into account the correlation between repeated measures and the robustness of the results regarless of how the correlation is modelled. We should know that Binomial ML model ignores some randomness, which can lead to underestimation of effect size and underestimation of the overall variation. The GEE analysis, assuming equal correlation (exchangeable correlation), but it is not very plausible. GLIMM model is a GLM with both fixed and random effects. So, I would say GLIMM is the best model that captures random effect and also it does not require the assumption as GEE.