# STAT850 HW8

*Stewart Kerr*

*March 25, 2019*

## Problem 1

**a.** See attached page.

**b.**

```
#Problem 1b
#Load in the data
cholesterol <- read.csv("~/2019spring/STAT850/hw8/cholesterol.csv")
cholesterol$patient = factor(cholesterol$patient)
cholesterol$run = factor(cholesterol$run)

#Fit the data (fixed for ANOVA table)
cholesterol_fixed <- lm(cholesterol ~ patient/run, data = cholesterol)
cholesterol_lmer <- lmer(cholesterol ~ 1 + (1|patient/run), data = cholesterol)
anova(cholesterol_fixed)
```

```
## Analysis of Variance Table
##
## Response: cholesterol
##             Df Sum Sq Mean Sq  F value    Pr(>F)
## patient      4  52608 13152.0 1990.168 < 2.2e-16 ***
## patient:run 15   1203    80.2   12.134 6.189e-07 ***
## Residuals   20    132     6.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(cholesterol_lmer)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: cholesterol ~ 1 + (1 | patient/run)
##    Data: cholesterol
##
## REML criterion at convergence: 255.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.70946 -0.49211 -0.07791  0.38972  1.91175
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  run:patient (Intercept)   36.795  6.066
##  patient     (Intercept) 1633.627 40.418
##  Residual                   6.608  2.571
## Number of obs: 40, groups:  run:patient, 20; patient, 5
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
```

```
## (Intercept)  167.435     18.131   4.001   9.235 0.000763 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For the design specified in part a, I have everything modeled as a random effect. However, to get the ANOVA table, I model each factor as fixed. I then use `lmer()` to model the data as specified in part a and call summary to look at the estimated variance components. From both the sum of squares in the ANOVA table and the estimated variance from the summary of the random effects, it appears that the patient effect explains most of the variance in the data.

**c.** The way I have specified the model, run is nested within patient. Therefore, there can be no interaction between patient and run. This excludes a multiplicative effect. Thus, I can only check if patient and run effects are additive in that both effects are significant and contribute to the model given that run is nested within patient. If we had assumed fixed effects rather than random, then we could look at the ANOVA table and see that with run nested within patient, we have an F-value of 12.134 and a p-value of 6.2e-07. This makes me believe that the effects of patient and run do appear to be additive.

**d.** As I've written on the attached sheet, the variances between measurements of the same patient and the same run and the variances between measurements of the same patient in different runs will be the same if $\sigma_R^2 = 0$, that is the random effect of run (nested within patient) does not differ. Then, our null hypothesis is $H_0 : \sigma_R^2 = 0$ vs. $H_1 : \sigma_R^2 \neq 0$. We can test this be performing an F-test with the test statistic $f = \frac{MSR}{MSE}$ where MSR is the mean squared error for run and MSE is the mean squared error for residuals. This F-test was performed in the ANOVA table above, with a test statistic of $F = 12.134$ and a p-value of 6.2e-7. This provides very strong evidence against the null hypothesis. In other words, it's likely that measurements between runs differs significantly compared to measures within runs.

# Problem 2

The formulation for this problem is on the attached sheet. As mentioned in the problem description, the data is log-transformed before analysis. We are interested in detecting a 10% increase in Y from the intervention. This is equivalent to a observing a difference in $log(\frac{10\mu_1}{\mu_1}) = 2.3026$ difference in the log-transformed response. That is, we want $\alpha_1 - \alpha_2 \geq 2.3026$ with $\alpha_2 = -\alpha_1$. Thus, we are constrained by $\alpha_1 \geq log(10)/2$.
Thus, the calculation of the non-centrality parameter becomes $\lambda = \frac{2vn\alpha_1^2}{0.77+0.02n}$. We want to detect a difference that is *at least* $\frac{log(10)}{2}$, thus lambda must be $\lambda \geq \frac{vnlog(10)^2}{2\times(0.77+0.02n)}$. With this formulation of lambda, I will plug in different numbers of n and v to calculate power and determine what an optimal choice for n and v might be. The question states that implementing the intervention is a village is costly - thus I will seek primarily to minimize v.

```
#Problem 2
Fpower <- function(n,v){
    #Calculate noncentral parameter
    noncentralp = v*n*(log(10)^2)/(2*(0.77+0.02*n))
    #Find F that would cause us to reject at 0.95 percent
    rejectionf = qf(0.95,1,v-1)
    #calculate power
    power = pf(rejectionf, 1, v-1, ncp = noncentralp, lower.tail = FALSE)
    return(power)
}

#This calculates the power for a bunch of different n,v combinations
n = seq(from = 1, to = 500)
v = seq(from = 2, to = 4)
df = expand.grid(v,n) %>%
  rename(v = Var1, n = Var2) %>%
  mutate(power = Fpower(n,v))
```

```
#Remove values of n & v that have too low of power and high power, return head
kable(filter(df, power >= 0.7 & power <= 0.85) %>% arrange(-power) %>% head())
```

| v | n | power |
|---|---|---|
| 3 | 4 | 0.8467631 |
| 2 | 500 | 0.7816539 |
| 2 | 499 | 0.7816209 |
| 2 | 498 | 0.7815878 |
| 2 | 497 | 0.7815546 |
| 2 | 496 | 0.7815212 |

If you look at the R results, we see that increasing v gives us a significant increase in power while increasing n only gives us a marginal increase in power. This effect is so extreme that we only need 4 participants with 3 villages to obtain a power greater than 0.80 but with 500 participants and 2 villages we do not even obtain a power greater than 0.80. With this result, I would likely recommend at least 3 villages for the study. If the intervention is *extremely* costly to implement on the village level, I would recommend 2 villages and at least 500 study participants in each village.

# Problem 3

**a.** The configuration of means that provides maximal power is one with $|\mu_i| = 1 \forall i$ and the number of group means equal to 1 is $\lfloor a/2 \rfloor$ or $\lceil a/2 \rceil$. For the proof, see attached page. Next, I will calculate this power for a = 4 and a = 5.

```
#Problem 3a
Fmaxpower <- function(a){
    #Calculate noncentral parameter
    n = floor(a/2)
    noncentralp = 12*n*(1-(n/a))
    #Find F that would cause us to reject at 0.95 percent
    rejectionf = qf(0.95,a-1,24)
    #calculate power
    power = pf(rejectionf, a-1, 24, ncp = noncentralp, lower.tail = FALSE)
    paste("The max power for a = ", a, " is ", round(power,4))

}
Fmaxpower(4)
```

```
## [1] "The max power for a =  4  is  0.7697"
```

```
Fmaxpower(5)
```

```
## [1] "The max power for a =  5  is  0.7963"
```

**b.** The configuration of means that provides minimal power in this scenario is one with $\mu_1 = 1, \mu_2 = -1$ and $\mu_i = 0 \forall i > 2$. For the proof, check the attached page. Next, I will calculate minimum power for a = 4 and a = 5.

```
#Problem 3b
Fminpower <- function(a){
  noncentralp = 2
  rejectionf = qf(0.95,a-1,24)
  power = pf(rejectionf, a-1, 24, ncp = noncentralp, lower.tail = FALSE)
```

```
  paste("The min power for a  = ",a, " is ", round(power,4))
}
Fminpower(4)
```

```
## [1] "The min power for a  =  4  is  0.169"
```

```
Fminpower(5)
```

```
## [1] "The min power for a  =  5  is  0.1479"
```

Compared to the maximal power in part a, we get a much lower power. It's interesting to note that as a increases, our maximum power increases and our minimum power increases. That is, the gap because minimum and maximum power increases.

# Problem 4

I was not able to prove this problem analytically because the derivative is gnarly. However, I wrote an R simulation to demonstrate that the maximum power is achieved when $n_0 \approx \sqrt{p}n$.

```
#Problem 4
p4power <- function(n0,n){
  sp = ((n0-1)+(n-1))/(n0+n-2)
  #Power is proportional to the following
  power = 1/(sqrt(sp)*sqrt((1/n0) + (1/n)))
  return(power)
}

#Let's assume N is set to 100 and p = 4. Then I will build a line of possible sample sizes
n = seq(1,20)
n0 = 100-4*n
df <- data.frame(n0,n) %>% mutate(prop_power = p4power(n0,n))
head(arrange(df, -prop_power))
```

```
##   n0  n prop_power
## 1 32 17   3.331973
## 2 36 16   3.328201
## 3 28 18   3.310064
## 4 40 15   3.302891
## 5 44 14   3.258940
## 6 24 19   3.256478
```

In the code above, we have p = 4 and N = 100. I coded up a function that returns a quantity that is proportional to power. I submitted input that matches the constraints defined in the problem. We see that the configuration the achieves maximum power is one with $n_0 = 32, n = 17$ which matches with our expectation that maximum power is achieved with $n_0 \approx \sqrt{p}n$. To further test my results, I will set p = 9 and N = 1000.

```
#problem 4
n = seq(1,100)
n0 = 1000-9*n
df <- data.frame(n0,n) %>% mutate(prop_power = p4power(n0,n))
head(arrange(df, -prop_power))
```

```
##    n0  n prop_power
## 1 253 83   7.905506
## 2 244 84   7.904923
```

```
## 3 262 82   7.902752
## 4 235 85   7.900752
## 5 271 81   7.896885
## 6 226 86   7.892710
```

Again, we have the optimal $n_0$ is about $\sqrt{p}n$. In conclusion, if we have a control group and we care only about comparing different groups to the control group, it is optimal to design our experiment such that it is unbalanced with the control group having approximately $\sqrt{p}n$ observations.

# Code

```
## ----include = FALSE-----------------------------------------------------
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(knitr)
library(car)
library(ggplot2)
library(MASS)
library(lme4)
library(lmerTest)
library(tidyr)
library(cowplot)
library(multcomp)

set.seed(1104)             # make random results reproducible

this_file <- "kerr_stat850_hw08.Rmd"  # used to automatically generate code appendix

## ------------------------------------------------------------------------
#Problem 1b
#Load in the data
cholesterol <- read.csv("~/2019spring/STAT850/hw8/cholesterol.csv")
cholesterol$patient = factor(cholesterol$patient)
cholesterol$run = factor(cholesterol$run)

#Fit the data (fixed for ANOVA table)
cholesterol_fixed <- lm(cholesterol ~ patient/run, data = cholesterol)
cholesterol_lmer <- lmer(cholesterol ~ 1 + (1|patient/run), data = cholesterol)
anova(cholesterol_fixed)
summary(cholesterol_lmer)

## ---- warning=FALSE-----------------------------------------------------
#Problem 2
Fpower <- function(n,v){
    #Calculate noncentral parameter
    noncentralp = v*n*(log(10)^2)/(2*(0.77+0.02*n))
    #Find F that would cause us to reject at 0.95 percent
    rejectionf = qf(0.95,1,v-1)
    #calculate power
    power = pf(rejectionf, 1, v-1, ncp = noncentralp, lower.tail = FALSE)
    return(power)
}
```

```r
#This calculates the power for a bunch of different n,v combinations
n = seq(from = 1, to = 500)
v = seq(from = 2, to = 4)
df = expand.grid(v,n) %>%
  rename(v = Var1, n = Var2) %>%
  mutate(power = Fpower(n,v))

#Remove values of n & v that have too low of power and high power, return head
kable(filter(df, power >= 0.7 & power <= 0.85) %>% arrange(-power) %>% head())

## ------------------------------------------------------------------------
#Problem 3a
Fmaxpower <- function(a){
    #Calculate noncentral parameter
    n = floor(a/2)
    noncentralp = 12*n*(1-(n/a))
    #Find F that would cause us to reject at 0.95 percent
    rejectionf = qf(0.95,a-1,24)
    #calculate power
    power = pf(rejectionf, a-1, 24, ncp = noncentralp, lower.tail = FALSE)
    paste("The max power for a = ", a, " is ", round(power,4))

}
Fmaxpower(4)
Fmaxpower(5)

## ------------------------------------------------------------------------
#Problem 3b
Fminpower <- function(a){
  noncentralp = 2
  rejectionf = qf(0.95,a-1,24)
  power = pf(rejectionf, a-1, 24, ncp = noncentralp, lower.tail = FALSE)
  paste("The min power for a  = ",a, " is ", round(power,4))
}
Fminpower(4)
Fminpower(5)

## ------------------------------------------------------------------------
#Problem 4
p4power <- function(n0,n){
  sp = ((n0-1)+(n-1))/(n0+n-2)
  #Power is proportional to the following
  power = 1/(sqrt(sp)*sqrt((1/n0) + (1/n)))
  return(power)
}

#Let's assume N is set to 100 and p = 4. Then I will build a line of possible sample sizes
n = seq(1,20)
n0 = 100-4*n
df <- data.frame(n0,n) %>% mutate(prop_power = p4power(n0,n))
head(arrange(df, -prop_power))

## ------------------------------------------------------------------------
```

```r
#problem 4
n = seq(1,100)
n0 = 1000-9*n
df <- data.frame(n0,n) %>% mutate(prop_power = p4power(n0,n))
head(arrange(df, -prop_power))

## ----code = readLines(purl(this_file, documentation = 1)), echo = T, eval = F----
## # this R markdown chunk generates a code appendix
```