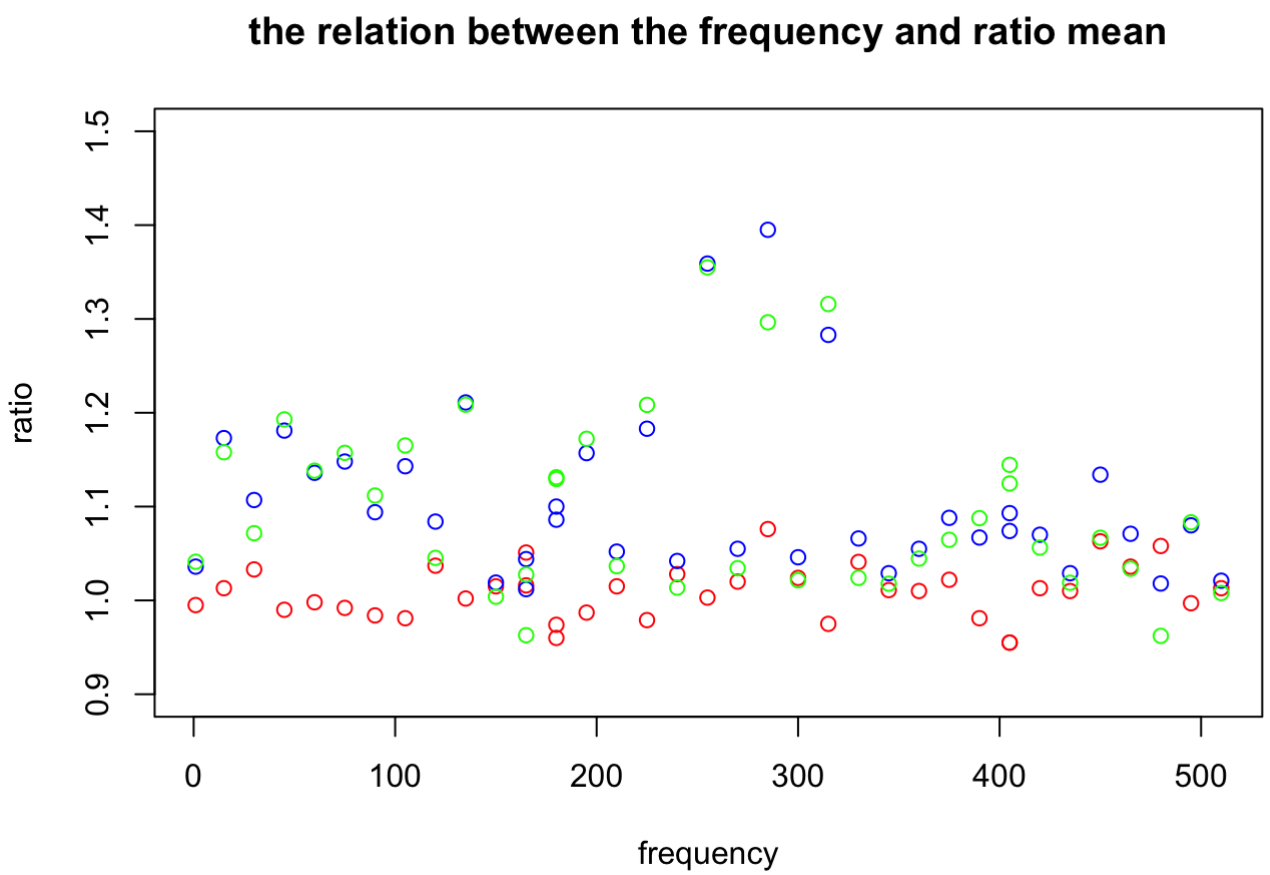
homework 7a

homework 7a

1.

A.

```
## read the data
data <- read.table('chickens.txt',header= FALSE)
colnames(data) <- c('Frequency','Treat S Ratio N','Treat S Ratio Mean','Treat S Ratio S
E','Treat E N','Treat E Mean','Treat E SE')
plot(data[,c('Frequency')],data[,c('Treat S Ratio Mean')],col='red',xlab='frequency',yla
b='ratio',main='the relation between the frequency and ratio mean',ylim=c(0.9,1.5))
points(data[,c('Frequency')],data[,c('Treat E Mean')],col='blue')
points(data[,c('Frequency')],data[,c('Treat E Mean')]/data[,c('Treat S Ratio Mean')],col
='green')
```



As we can see from the plot, the difference in ration mean between the exposure treatment vs. control (blue points) and sham vs. control (red points). Based on the plot, we have the idea that the the ratio of sham case is always around 1 and the variance of the ratios is very small. But for the exposure case, the variance is bigger and the ratio is always higher than 1.

And for the ratio of these two ratios (green points) the variance tends to be bigger.

It seems like, for the exposure treatment vs. control and ratio of these two ratios, the model indeed have some changing effect across the frequencies. The effect is going to be bigger and then becoming smaller.

```
# calculate the pearson correlatio
cor(data[,c('Frequency')],data[,c('Treat S Ratio Mean')]) / (sd(data[,c('Frequency')]) *
 sd(data[,c('Treat S Ratio Mean')]))
```

```
## [1] 0.03682923
```

```
cor(data[,c('Frequency')],data[,c('Treat E Mean')]) / (sd(data[,c('Frequency')]) * sd(da
ta[,c('Treat E Mean')]))
```

```
## [1] -0.01532392
```

As we can see that the pearson correlation are all very small, in particular the linear correlation between frequency and exposure ratio is very week.

And it's interesting that the correlation is one for positive and one for negative.

```
## rank correlations
cor.test(x= data[,c('Frequency')],y=data[,c('Treat S Ratio Mean')],method='spearman')
```

```
## Warning in cor.test.default(x = data[, c("Frequency")], y = data[, c("Treat
## S Ratio Mean")], : Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  data[, c("Frequency")] and data[, c("Treat S Ratio Mean")]
## S = 7749.2, p-value = 0.362
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1520775
```

```
cor.test(x= data[,c('Frequency')],y=data[,c('Treat E Mean')],method='spearman')
```

```
## Warning in cor.test.default(x = data[, c("Frequency")], y = data[, c("Treat
## E Mean")], : Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  data[, c("Frequency")] and data[, c("Treat E Mean")]
## S = 12037, p-value = 0.05242
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.3170798
```

Based on the two p value we can say that the correlation is very week. But the correlation effect for exposure case has a p value 0.05242 which means the rank correlation is relatively strong.

B.

The ‘sham’ treatment maybe used to measure the so called ‘placebo effect’. The target of this model is explore the effect of electromagnetic fields only. We want to exclude the possible effects from other factors which may also lead to a big or small ration mean.

For example, in control group the half brain in put in a water bath while for the treatment group half brain is put in the air. The difference in enviroment may also matters which we need to use the sham treatment to measure its effect.

C.

The ii. summary could be better. As I have answered in subquestion b. There are some ‘placebo effect’ may exist which could also matter. Thus, if we just measure the ration between exposure treatment vs. control we cannot exclude the effect that may come from the difference between putting the brain in the air or in the water. After all, we can always expect to have a better estimate if we add more information in the model.

First try: simple linear regression with weak prior

Here, I do two different bayesian regression model for the i and ii.

```
## sibutation 1
library(rstan)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: StanHeaders
```

```
## rstan (Version 2.17.4, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
print_file <- function(file) {
  cat(paste(readLines(file), "\n", sep=""), sep="")
}
print_file("fit1.stan")
```

```
## Warning in readLines(file): incomplete final line found on 'fit1.stan'
```

```
## data{
##     int<lower=0> N;
##     vector[N] y;
##     vector[N] x;
## }
## parameters{
##     real beta;
##     real alpha;
##     real<lower=0> sigma;
## }
## model{
##     beta ~ cauchy(1,1);
##     y ~ normal(alpha + beta * x,sigma);
## }
```

```
## Inference for Stan model: fit1.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##        mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## beta   0.00    0.00 0.00  0.00  0.00  0.00  0.00  0.00  1707    1
## alpha  1.14    0.00 0.03  1.08  1.12  1.14  1.15  1.19  1323    1
## sigma  0.09    0.00 0.01  0.07  0.08  0.09  0.10  0.12  1023    1
## lp__  69.75    0.04 1.23 66.57 69.20 70.06 70.66 71.18  1133    1
##
## Samples were drawn using NUTS(diag_e) at Tue Oct 16 11:31:17 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

As we can see, when we use the ratio of ratios

```
## sibutation 2
stan_data_2 <- list(x = data[,c('Frequency')],
             y = data[,c('Treat E Mean')]/data[,c('Treat S Ratio Mean')],
             N = length(data[,c('Frequency')]))
fit2 <- stan('fit1.stan',data=stan_data_2)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on '/
## Users/yi/Desktop/study/subjects/bayesian-data-analysis/homework/homework
## 10/fit1.stan'
```

```
fit2
```

```
## Inference for Stan model: fit1.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##        mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## beta   0.00    0.00 0.00  0.00  0.00  0.00  0.00  0.00  1555    1
## alpha  1.14    0.00 0.03  1.08  1.12  1.14  1.16  1.20  1358    1
## sigma  0.10    0.00 0.01  0.08  0.09  0.09  0.10  0.12  1068    1
## lp__  67.92    0.04 1.29 64.61 67.34 68.25 68.85 69.38  1146    1
##
## Samples were drawn using NUTS(diag_e) at Tue Oct 16 11:31:20 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Accorcding to the linear regression result, the estimation of the effects are almost the same. The parameters esitmated for both ratios are 0.

But as we discuss in a) the relationship between the ratio and frequencies may not be linear. Another problem is that, we do not incorperate the information about the group size and standard deviation at all.

Second try: hierarchical model

Next I will hierarchical model, first let's make the fake data

```
print_file('fit2.stan')
```

```
## Warning in readLines(file): incomplete final line found on 'fit2.stan'
```

```
## data{
##   int<lower=0> N;
##   real y[N];
##   real<lower=0> sigma[N];
## }
## parameters{
##   real mu;
##   real<lower=0> tau;
##   real eta[N];
## }
## transformed parameters{
##   real theta[N];
##   for (i in 1:N){
##     theta[i] = mu + tau * eta[i];
##   }
## }
## model{
##   target += normal_lpdf(eta |0,1);
##   target += normal_lpdf(y | theta,sigma);
## }
```

```
data_3 <- list(N=dim(data)[1],
             y = data$`Treat E Mean`,
             sigma = data$`Treat E SE`)
fit3 <- stan('fit2.stan',data=data_3)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on '/
## Users/yi/Desktop/study/subjects/bayesian-data-analysis/homework/homework
## 10/fit2.stan'
```

```
result1 <- extract(fit3)
```

```
print_file('fit3.stan')
```

```
## Warning in readLines(file): incomplete final line found on 'fit3.stan'
```

```
## data{
##   int<lower=0> N;
##   real y_s[N];
##   real<lower=0> sigma_s[N];
##   real y_e[N];
##   real<lower=0> sigma_e[N];
##
## }
## parameters{
##   real mu_s;
##   real<lower=0> tau_s;
##   real eta_s[N];
##   real mu_e;
##   real<lower=0> tau_e;
##   real eta_e[N];
##
## }
## transformed parameters{
##   vector[N] theta_s;
##   vector[N] theta_e;
##   vector[N] ratios;
##   for (i in 1:N){
##     theta_s[i] = mu_s + tau_s * eta_s[i];
##     theta_e[i] = mu_e + tau_e * eta_e[i];
##     ratios[i] = theta_e[i] / theta_s[i];
##   }
## }
## model{
##   target += normal_lpdf(eta_s |0,1);
##   target += normal_lpdf(y_s | theta_s,sigma_s);
##   target += normal_lpdf(eta_e |0,1);
##   target += normal_lpdf(y_e | theta_e,sigma_e);
## }
## generated quantities{
##   real overall_ratio;
##   overall_ratio = mu_e / mu_s;
## }
```

```
data_4 <- list(N=dim(data)[1],
             y_e = data$`Treat E Mean`,
             sigma_e = data$`Treat E SE`,
             y_s = data$`Treat S Ratio Mean`,
             sigma_s = data$`Treat S Ratio SE`)
fit4 <- stan('fit3.stan',data=data_4)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on '/
## Users/yi/Desktop/study/subjects/bayesian-data-analysis/homework/homework
## 10/fit3.stan'
```

```
result2 <- extract(fit4)
```

Summary

As we can see from the the last two stan. The second model is better for the following reasons: 1. For the overall treatment effect

```
# model 1
mean(result1$mu); sd(result1$mu)
```

```
## [1] 1.101121
```

```
## [1] 0.01306786
```

```
# model 2
mean(result2$mu_s);sd(result2$mu_s)
```

```
## [1] 1.004292
```

```
## [1] 0.006417483
```

```
mean(result2$mu_e);sd(result2$mu_e)
```

```
## [1] 1.101432
```

```
## [1] 0.01323707
```

```
mean(result2$overall_ratio);sd(result2$overall_ratio)
```

```
## [1] 1.096771
```

```
## [1] 0.01499692
```

We can see that for the first model the overall treatment effect is about 1.1 and the standard deviation is about 0.013. Compared with the second model, this esimator actually overestimate the treatment effect and have a relative small standard estimation. But no matter which model to pick, it is clear that there exist some treatment effect since the ratio is significantly different from 1.

2.

```
# model 1
mean(result1$tau); sd(result1$tau)
```

```
## [1] 0.07018835
```

```
## [1] 0.01224972
```

```
# model 2
mean(result2$tau_s);sd(result2$tau_s)
```

```
## [1] 0.007959768
```

```
## [1] 0.005849171
```

```
mean(result2$tau_e);sd(result2$tau_e)
```

```
## [1] 0.07042556
```

```
## [1] 0.01204269
```

As we can see, for the ratio of ratios, the standard deviation of treatment effect is relatively large which means that the treatment effects are not a constant over the frequency (it is not stable) If we just use the first method, we would msitakly think the treatment effects are more statable than it should be, this is because the placebo effect is more stable (we can see from the model 2 tau_s) and it is included in the first model.

Thus, it's clear that choose the second model would give us a more precise result and distinguish the different effect from placebo and treatment.

2.

I personally would prefer the second analysis. Assuming that there is no dependence is relatively a strong assumption. Even if we know that the cell culturres are far enought, it probability would have some other dependence among them Trough the hiercatical modeling we can somehow exclude the effect from dish effect and get a better estimation. And even the effect from the dishes is very small as the assumption, we can still tell from the data and model results. But is would be better to have more data within in dish not just 5 which would not be very reliable.