STAT850 HW11

Stewart Kerr

May 1, 2019

Problem 1

See attached page.

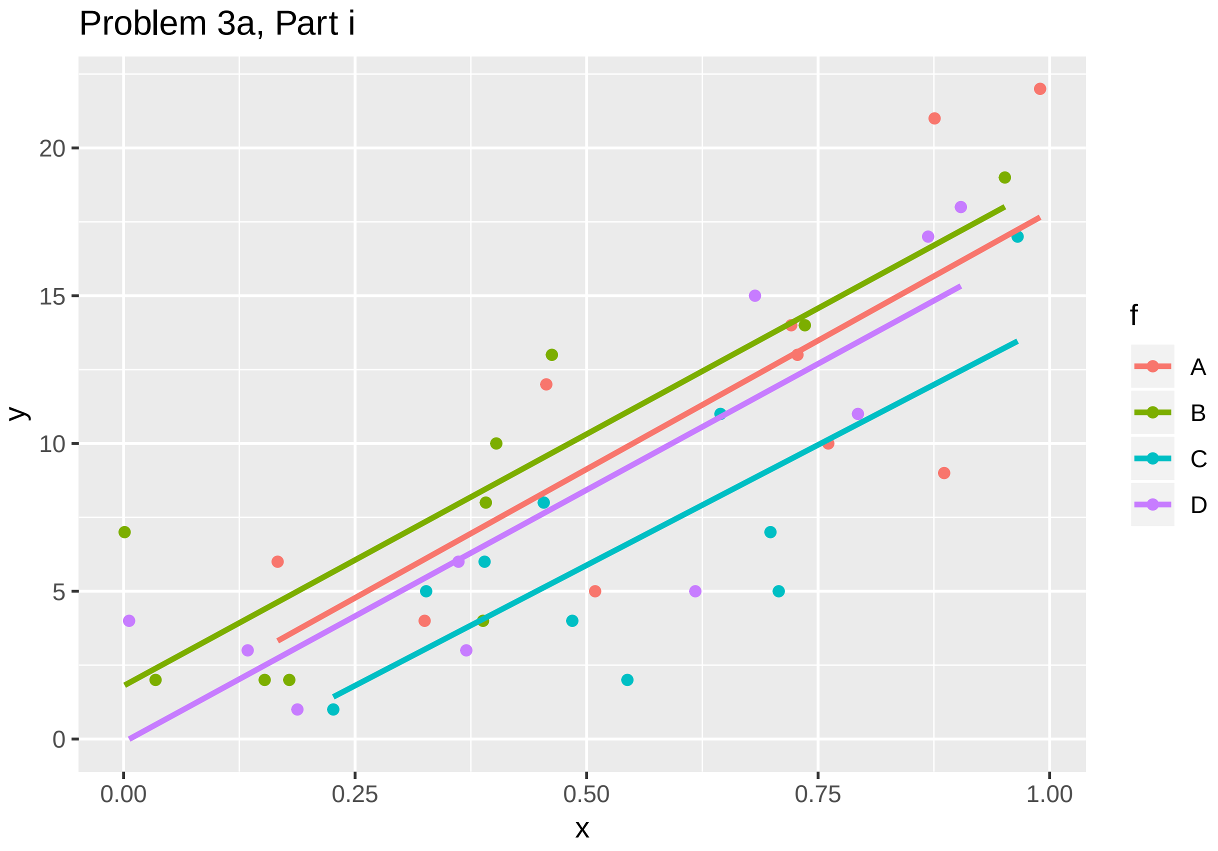
Problem 2

See attached page.

Problem 3

a. i.

```r
## Problem 3a, part i
set.seed(12345)
#simulate data set
p3_1 = simfun(ng=4, nr=10, fsd=0, indsd=0, b=c(1,2))
#plot dataset
p3_plot(p3_1, "Problem 3a, Part i")
```



```
#Fit the data sets
p3_1f1 <- glm(y ~ x, family="quasipoisson", data = p3_1)
p3_1f2 <- glm(y ~ x + f, family="quasipoisson", data = p3_1)
#Look at the fits
summary(p3_1f1)
```

```
##
## Call:
## glm(formula = y ~ x, family = "quasipoisson", data = p3_1)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.50818  -0.94431   0.02315   0.69997   2.21616
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9708     0.1788   5.430 3.44e-06 ***
## x             2.0123     0.2516   7.997 1.15e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.445401)
##
##     Null deviance: 157.194  on 39  degrees of freedom
## Residual deviance:  55.847  on 38  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

```
summary(p3_1f2)
```

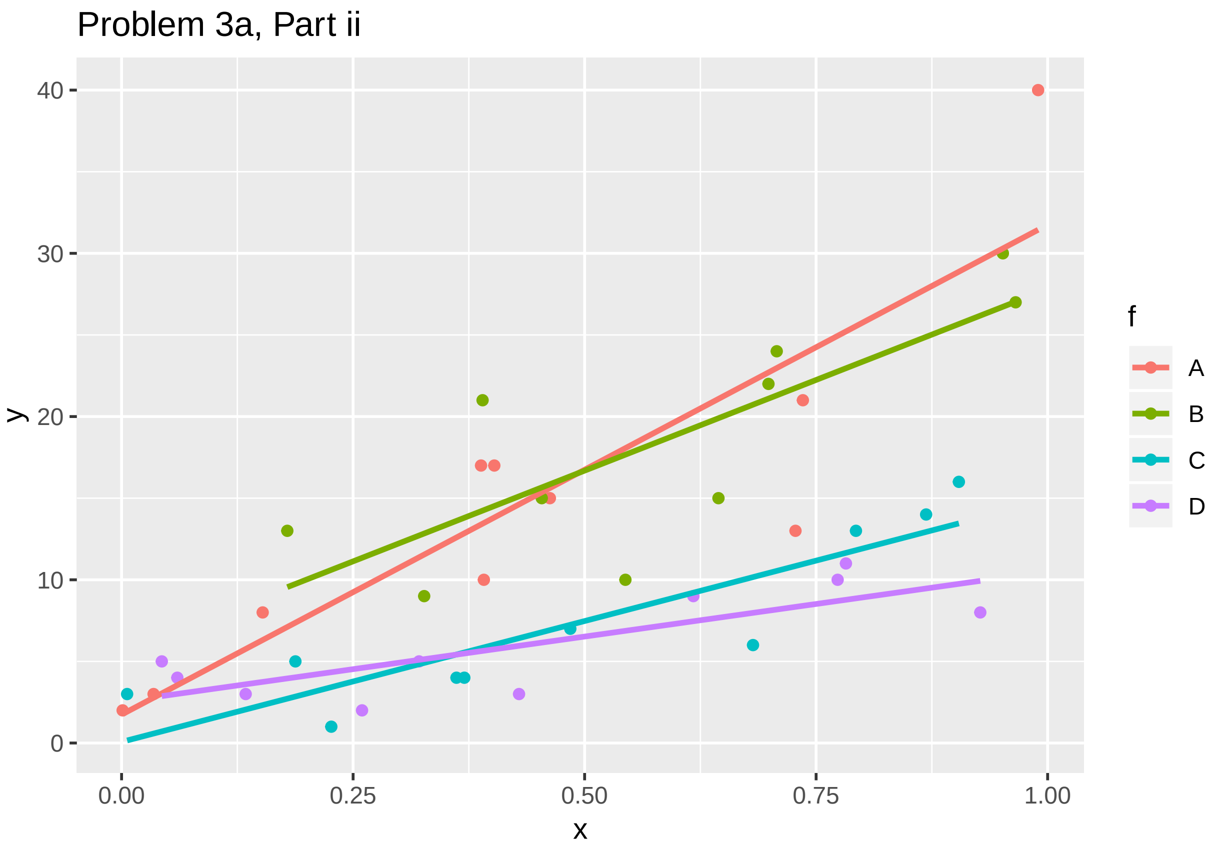
```
##
## Call:
## glm(formula = y ~ x + f, family = "quasipoisson", data = p3_1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1089  -1.1059   0.0854   0.7548   1.8350
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.03177    0.21242   4.857 2.47e-05 ***
## x            2.02552    0.24464   8.280 9.26e-10 ***
## fB           0.12796    0.17199   0.744   0.4618
## fC          -0.33779    0.17811  -1.897   0.0662 .
## fD          -0.09394    0.16542  -0.568   0.5737
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.299973)
##
##     Null deviance: 157.194  on 39  degrees of freedom
## Residual deviance:  47.087  on 35  degrees of freedom
## AIC: NA
```

```
##
## Number of Fisher Scoring iterations: 4
```

If we look at the model summaries, both models have a low dispersion parameter (less than 2) indicating that the data do not exhibit overdispersion. For this simulation, there were no random effects for groups or individuals. This is reflected in the 2nd model in which we modeled group as a fixed factor but the factor is not significant.

ii.

```
## Problem 3a, part ii
set.seed(12345)
#simulate data set
p3_2 = simfun(ng=4, nr=10, fsd=1, indsd=0, b=c(1,2))
#plot dataset
p3_plot(p3_2, "Problem 3a, Part ii")
```



```
#Fit the data sets
p3_2f1 <- glm(y ~ x, family="quasipoisson", data = p3_2)
p3_2f2 <- glm(y ~ x + f, family="quasipoisson", data = p3_2)
#Look at the fits
summary(p3_2f1)
```

```
##
## Call:
## glm(formula = y ~ x, family = "quasipoisson", data = p3_2)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max  
## -3.4924  -1.2914  -0.5503   1.2531   3.5694  
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   1.4575     0.2049   7.113 1.73e-08 ***
## x             1.7762     0.2964   5.993 5.83e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for quasipoisson family taken to be 3.134825)
## 
##     Null deviance: 237.84  on 39  degrees of freedom
## Residual deviance: 117.37  on 38  degrees of freedom
## AIC: NA
## 
## Number of Fisher Scoring iterations: 5
```

```r
summary(p3_2f2)
```

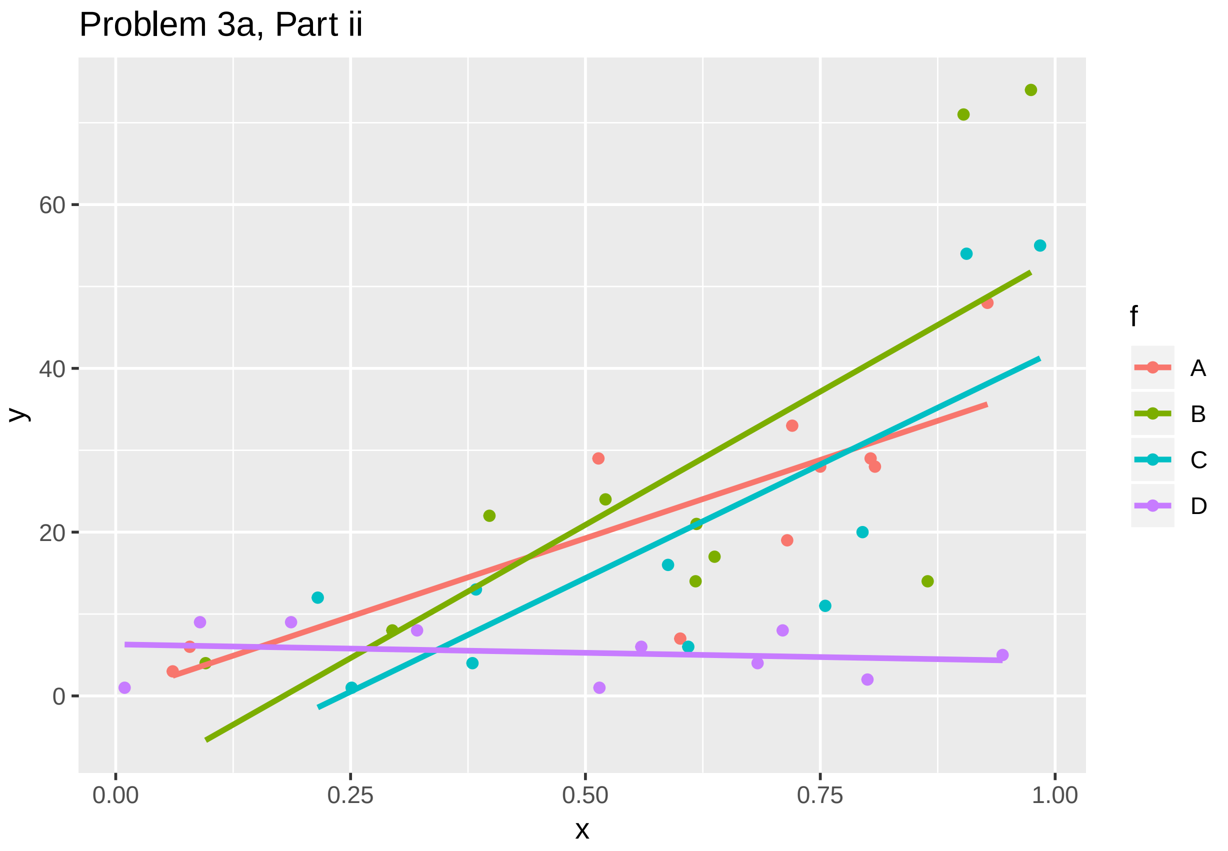
```
## 
## Call:
## glm(formula = y ~ x + f, family = "quasipoisson", data = p3_2)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.9741  -0.9344   0.1488   0.6449   2.2590  
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  1.82533    0.13812  13.216 3.62e-15 ***
## x            1.68545    0.18113   9.305 5.39e-11 ***
## fB           0.02515    0.11916   0.211    0.834    
## fC          -0.78105    0.15290  -5.108 1.16e-05 ***
## fD          -0.90013    0.16339  -5.509 3.43e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for quasipoisson family taken to be 1.135179)
## 
##     Null deviance: 237.839  on 39  degrees of freedom
## Residual deviance:  41.475  on 35  degrees of freedom
## AIC: NA
## 
## Number of Fisher Scoring iterations: 4
```

For this simulation, the first model, without a factor for group, does exhibit overdispersion as it has a dispersion parameter greater than 2. The second model, however, does not exhibit overdispersion. This is expected because we added in group random effects. Those random effects are not accounted for in the first model, but are handled by the fixed effect of f in the second model.

iii.

```r
## Problem 3a, part iii
set.seed(12345)
#simulate data set
```

```
p3_3 = simfun(ng=4, nr=10, fsd=1, indsd=0.5, b=c(1,2))
#plot dataset
p3_plot(p3_3, "Problem 3a, Part ii")
```



```
#Fit the data sets
p3_3f1 <- glm(y ~ x, family="quasipoisson", data = p3_3)
p3_3f2 <- glm(y ~ x + f, family="quasipoisson", data = p3_3)
#Look at the fits
summary(p3_3f1)
```

```
##
## Call:
## glm(formula = y ~ x, family = "quasipoisson", data = p3_3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.0488  -2.4941   0.0797   1.4708   5.1410
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.1518     0.3513   3.279  0.00224 **
## x             2.6946     0.4589   5.872 8.54e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 7.233927)
```

```
##
##     Null deviance: 619.92  on 39  degrees of freedom
## Residual deviance: 309.87  on 38  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
summary(p3_3f2)
```

```
##
## Call:
## glm(formula = y ~ x + f, family = "quasipoisson", data = p3_3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.2188  -1.7206  -0.3565   1.4977   4.1109
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.4451     0.3462   4.174 0.000188 ***
## x             2.5004     0.4181   5.980  8.2e-07 ***
## fB            0.1662     0.2186   0.760 0.452066
## fC           -0.1572     0.2379  -0.661 0.513051
## fD           -1.2404     0.3716  -3.338 0.002011 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 5.921775)
##
##     Null deviance: 619.92  on 39  degrees of freedom
## Residual deviance: 188.17  on 35  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

For this simulation, both models exhibit significant overdispersion. Neither model is accounting for the individual random effects and thus the expected variance from the model is much smaller than the observed variance from the simulation.

b.

```
#Problem 3b
m0 <- glmer(y ~ x + (1|f), family="poisson", data=p3_3)
m1 <- glmer(y ~ x + (1|f) + (1|obs), family="poisson", data=p3_3)

#Perform LRT
drop1(m0, test = "Chisq")
```

```
## Single term deletions
##
## Model:
## y ~ x + (1 | f)
##        Df    AIC    LRT   Pr(Chi)
## <none>    384.70
## x       1 640.44 257.73 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
drop1(m1, test = "Chisq")
```

```
## Single term deletions
##
## Model:
## y ~ x + (1 | f) + (1 | obs)
##         Df    AIC    LRT   Pr(Chi)
## <none>     297.40
## x       1 315.84 20.435 6.169e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

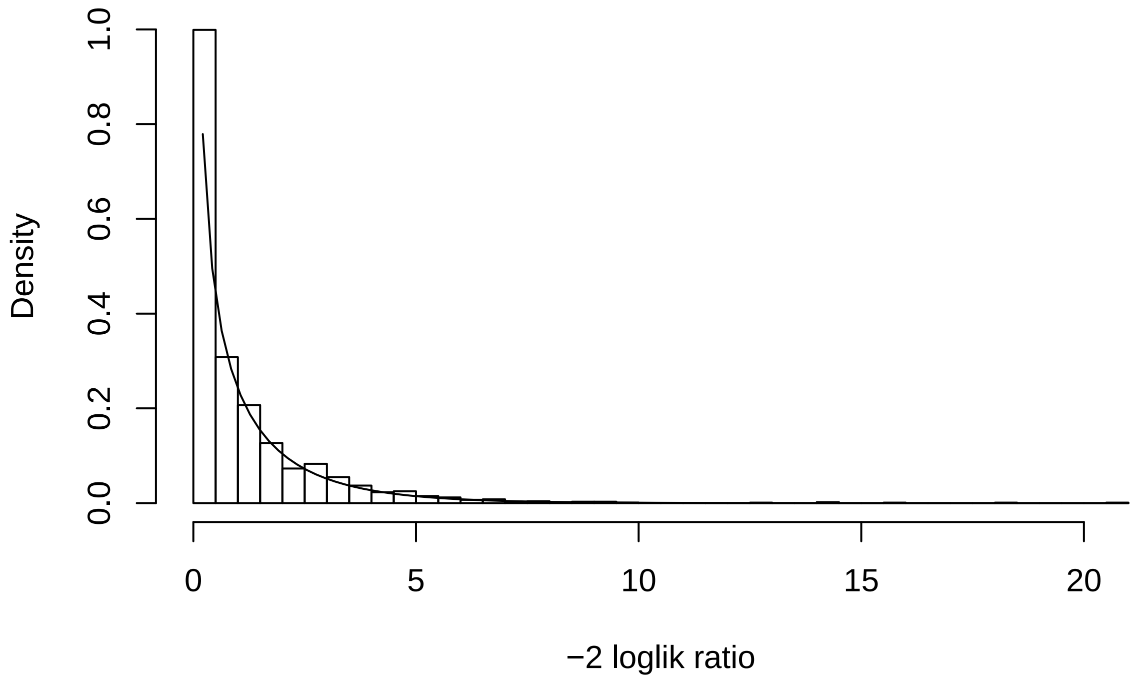
Both tests have very low p-values, suggesting that x is a significant predictor of y in both models. For the model with random effects, m1, the p-value is slightly larger. Intuitively, this makes sense as the “relative” effect of x on y is diminished when we add in an additional random effect.

c.

```
#Problem 3c
oneX2 = function(x){
  #Define null model to generate data
  null_model = glmer(y ~ 1 + (1|f) + (1|obs), family="poisson", data = p3_3)
  simy = simulate(null_model)$sim_1

  #Make alternative model from data simulated by null
  alt_model = glmer(simy ~ x + (1|f) + (1|obs), family="poisson", data = p3_3)

  #Return the LRT statistic
  x2 = drop1(alt_model, test="Chisq")$LRT[2]

  return(x2)
}

#Now, perform bootstrap
set.seed(4)
if (!exists('bag_3c')) {
  #Perform the bootstrap
  bag_3c = unlist(parallel::mclapply(1:2000,oneX2))
}
bag_df <- as.data.frame(bag_3c)

#Now find the p-value
bag_df$bag_3c[bag_df$bag_3c<0] = 0.0
pval = mean(bag_df$bag_3c >= 257.73)
paste("The p-value for the parametric bootstrap test is: ",pval)
```

```
## [1] "The p-value for the parametric bootstrap test is:  0"
```

```
#Plot the null distribution
hist(bag_df$bag_3c, freq = FALSE, breaks = 30, xlab = "-2 loglik ratio", main = "")
curve(dchisq(x,df=1), add = T, n = 100)
```



In part b, we observed a p-value of 6.2e-6. Thus, we would expect our p-value from the bootstrap to be very low. In fact, the bootstrap gives us a p-value of 0 for 2000 simulations. I also attempted 4000 simulations and still observed a p-value of 0. If we increased the number of simulations large enough, we would eventually observe a p-value, but the runtime for 2000 simulations is already too large. Regardless, it's clear from both part b and the bootstrap that x is a significant predictor of y (as expected).

d.

```
# Problem 3d
## repeat many times
rr = replicate(100, cfun(simfun()))
#Find means
obs_eff_mean <- mean(rr[1,])
group_eff_mean <- mean(rr[2,])
int_eff_mean <- mean(rr[3,])
x_eff_mean <- mean(rr[4,])
eff_mean <- rbind(obs_eff_mean, group_eff_mean, int_eff_mean, x_eff_mean)

#Find variance
obs_eff_var <- var(rr[1,])
group_eff_var <- var(rr[2,])
int_eff_var <- var(rr[3,])
x_eff_var <- var(rr[4,])
eff_var <- rbind(obs_eff_var, group_eff_var, int_eff_var, x_eff_var)


#Actual effects
obs_eff_real <- 0.5
```

```
group_eff_real <- 1.0
int_eff_real <- 1.0
x_eff_real <- 2.0
eff_real <- rbind(obs_eff_real, group_eff_real, int_eff_real, x_eff_real)


#cbind for table
p3d_table <- cbind(eff_real,eff_mean,eff_var)
rownames(p3d_table) <- c("Observation RE","Group RE","Intercept FE","Slope FE")
kable(p3d_table,
      row.names = TRUE,
      col.names = c("Actual","Mean","Variance"),
      digits = 3)
```

	Actual	Mean	Variance
Observation RE	0.5	0.491	0.008
Group RE	1.0	0.777	0.168
Intercept FE	1.0	0.899	0.247
Slope FE	2.0	1.990	0.167

From the table, we see that estimated fixed effect parameters are fairly unbiased across the 100 simulations but have a larger variance than the estimated random effect parameters. The random effects are a little biased (both biased towards smaller variance) and the models perform worse in estimating the group random effects than the individual random effects. While both the estimated random effects have smaller variance than the fixed effects, the variance of the estimated observation random effects is particularly small.

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)
#library(lsmeans)

set.seed(1104)             # make random results reproducible

this_file <- "kerr_stat850_hw11.Rmd"  # used to automatically generate code appendix

## ---- echo = FALSE, message = FALSE---------------------------------------
## define functions for this problem
## function to simulate data:
## ng groups, nr replicates (or individuals) per group, 1 observation / individual
## fsd = SD of group random effects, indsd = SD of individual random effects
## b = fixed effect coefficients, of size 2: intercept and slope
simfun = function(ng=4, nr=10, fsd=1, indsd=0.5, b=c(1,2)) {
  ntot = nr*ng
  b.reff = rnorm(ng, sd=fsd)
  b.rind = rnorm(ntot, sd=indsd)
  x = runif(ntot) # predictor
  dd = data.frame(x, f=factor(rep(LETTERS[1:ng], each=nr)), obs=factor(1:ntot))
  dd$eta0 = as.vector(model.matrix(~x,data=dd) %*% b)
  dd$bi = b.rind
  dd$eta = with(dd, eta0 + b.reff[f] + b.rind) # log E(Yi) = Xbeta + group RE + indiv RE bi
  dd$mu = exp(dd$eta)
  dd$y = with(dd, rpois(ntot, lambda=mu))
  dd
}

## function to extract the variance components and fixed effects
cfun = function(d) {
  m = glmer(y ~ x + (1|f) + (1|obs), family="poisson", data=d)
  c(sqrt(unlist(VarCorr(m))), fixef(m))
}

p3_plot <- function(df, title){
  ggplot(data = df, aes(x = x,y = y, color = f)) +
    geom_point() +
    geom_smooth(method = "glm", se = F) +
    ggtitle(title)
}

## ---- message=FALSE-------------------------------------------------------
## Problem 3a, part i
set.seed(12345)
#simulate data set
p3_1 = simfun(ng=4, nr=10, fsd=0, indsd=0, b=c(1,2))
#plot dataset
p3_plot(p3_1, "Problem 3a, Part i")
#Fit the data sets
p3_1f1 <- glm(y ~ x, family="quasipoisson", data = p3_1)
p3_1f2 <- glm(y ~ x + f, family="quasipoisson", data = p3_1)
#Look at the fits
summary(p3_1f1)
summary(p3_1f2)

## ---- message=FALSE-------------------------------------------------------
## Problem 3a, part ii
set.seed(12345)
#simulate data set
p3_2 = simfun(ng=4, nr=10, fsd=1, indsd=0, b=c(1,2))
#plot dataset
p3_plot(p3_2, "Problem 3a, Part ii")
#Fit the data sets
p3_2f1 <- glm(y ~ x, family="quasipoisson", data = p3_2)
p3_2f2 <- glm(y ~ x + f, family="quasipoisson", data = p3_2)
#Look at the fits
```

```r
summary(p3_2f1)
summary(p3_2f2)

## ---- message=FALSE-------------------------------------------------------
## Problem 3a, part iii
set.seed(12345)
#simulate data set
p3_3 = simfun(ng=4, nr=10, fsd=1, indsd=0.5, b=c(1,2))
#plot dataset
p3_plot(p3_3, "Problem 3a, Part ii")
#Fit the data sets
p3_3f1 <- glm(y ~ x, family="quasipoisson", data = p3_3)
p3_3f2 <- glm(y ~ x + f, family="quasipoisson", data = p3_3)
#Look at the fits
summary(p3_3f1)
summary(p3_3f2)

## -------------------------------------------------------------------------
#Problem 3b
m0 <- glmer(y ~ x + (1|f), family="poisson", data=p3_3)
m1 <- glmer(y ~ x + (1|f) + (1|obs), family="poisson", data=p3_3)

#Perform LRT
drop1(m0, test = "Chisq")
drop1(m1, test = "Chisq")

## ---- message=FALSE, warning=FALSE----------------------------------------
#Problem 3c
oneX2 = function(x){
  #Define null model to generate data
  null_model = glmer(y ~ 1 + (1|f) + (1|obs), family="poisson", data = p3_3)
  simy = simulate(null_model)$sim_1

  #Make alternative model from data simulated by null
  alt_model = glmer(simy ~ x + (1|f) + (1|obs), family="poisson", data = p3_3)

  #Return the LRT statistic
  x2 = drop1(alt_model, test="Chisq")$LRT[2]

  return(x2)
}

#Now, perform bootstrap
set.seed(4)
if (!exists('bag_3c')) {
  #Perform the bootstrap
  bag_3c = unlist(parallel::mclapply(1:2000,oneX2))
}
bag_df <- as.data.frame(bag_3c)

#Now find the p-value
bag_df$bag_3c[bag_df$bag_3c<0] = 0.0
pval = mean(bag_df$bag_3c >= 257.73)
```

```
paste("The p-value for the parametric bootstrap test is: ",pval)

#Plot the null distribution
hist(bag_df$bag_3c, freq = FALSE, breaks = 30, xlab = "-2 loglik ratio", main = "")
curve(dchisq(x,df=1), add = T, n = 100)

## ---- message=FALSE, warning=FALSE---------------------------------------
# Problem 3d
## repeat many times
rr = replicate(100, cfun(simfun()))
#Find means
obs_eff_mean <- mean(rr[1,])
group_eff_mean <- mean(rr[2,])
int_eff_mean <- mean(rr[3,])
x_eff_mean <- mean(rr[4,])
eff_mean <- rbind(obs_eff_mean, group_eff_mean, int_eff_mean, x_eff_mean)

#Find variance
obs_eff_var <- var(rr[1,])
group_eff_var <- var(rr[2,])
int_eff_var <- var(rr[3,])
x_eff_var <- var(rr[4,])
eff_var <- rbind(obs_eff_var, group_eff_var, int_eff_var, x_eff_var)


#Actual effects
obs_eff_real <- 0.5
group_eff_real <- 1.0
int_eff_real <- 1.0
x_eff_real <- 2.0
eff_real <- rbind(obs_eff_real, group_eff_real, int_eff_real, x_eff_real)


#cbind for table
p3d_table <- cbind(eff_real,eff_mean,eff_var)
rownames(p3d_table) <- c("Observation RE","Group RE","Intercept FE","Slope FE")
kable(p3d_table,
      row.names = TRUE,
      col.names = c("Actual","Mean","Variance"),
      digits = 3)


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