Homework 10 Solutions

11/17/2021

Problem 1

```
library(tidyverse)
library(arm)
library(lme4)
library(latex2exp) # for nice plot labeling (using latex notation)

cdi <- read.table(file = "../cdi.dat")

cdi$pci <- rescale(2*cdi$per.cap.income)
cdi$phg <- rescale(2*cdi$pct.hs.grad)
```

(a)

See Figure 1.

```
# pooled regression
lm_all <- lm(pci ~ phg, data = cdi)

lm_all_rep <- data.frame(intercept = lm_all[["coefficients"]][1],
                         slope = lm_all[["coefficients"]][2],
                         state = unique(cdi$state))

# unpooled regression
state_lm_func <- function(df){
    mod <- lm(pci ~ phg, data = df)
    return(data.frame(intercept = mod[["coefficients"]][1],
                      slope = mod[["coefficients"]][2]))
}

state_lm_df <- cdi %>%
  group_by(state) %>% do(., state_lm_func(.))

## unpooled regression version II
lm_unpooled <- lm(pci ~ state + phg:state - 1, data = cdi) # single model

coef_unpooled <- coef(lm_unpooled)
state_loc <- substring(names(coef_unpooled), 6,7)
slope_loc <- substring(names(coef_unpooled), 9,11)

intercepts <- coef_unpooled[slope_loc == ""]
intercepts_df <- data.frame(state = substring(names(intercepts), 6,7),
                            intercept = intercepts)
slopes <- coef_unpooled[slope_loc == "phg"]
```

```r
slopes_df <- data.frame(state = substring(names(slopes), 6,7),
                        slope = slopes)

state_lm_df2 <- intercepts_df %>% left_join(slopes_df, by = 'state')

testthat::expect_equivalent(state_lm_df, state_lm_df2) # no error - so true

## unpooled regression version III
lm_unpooled2 <- lm(pci ~ state*phg - 1 - phg, data = cdi) # single model

coef_unpooled2 <- coef(lm_unpooled2)
state_loc2 <- substring(names(coef_unpooled2), 6,7)
slope_loc2 <- substring(names(coef_unpooled2), 9,11)

intercepts2 <- coef_unpooled2[slope_loc2 == ""]
intercepts_df2 <- data.frame(state = substring(names(intercepts2), 6,7),
                             intercept = intercepts2)
slopes2 <- coef_unpooled2[slope_loc2 == "phg"]
slopes_df2 <- data.frame(state = substring(names(slopes2), 6,7),
                         slope = slopes2)

state_lm_df3 <- intercepts_df2 %>% left_join(slopes_df2, by = 'state')

testthat::expect_equivalent(state_lm_df, state_lm_df3) # no error - so true


# multilevel model
multilevel_lmer <- lmer(formula = pci ~ 1 + phg + (1 | state) + (0 + phg | state),
                        data = cdi)

multilevel_lmer_df <- coef(multilevel_lmer)$state %>%
  tibble::rownames_to_column() %>%
  rename(state = "rowname",
         intercept = "(Intercept)",
         slope = "phg")

# multilevel model version II
betas <- fixef(multilevel_lmer)
etas <- ranef(multilevel_lmer)$state

combo_multilevel_lmer_df <- etas +
  matrix(rep(betas, each = nrow(etas)), nrow = nrow(etas))

testthat::expect_equivalent(combo_multilevel_lmer_df,
                            coef(multilevel_lmer)$state) # doesn't error if true

# combined
combined <- rbind(mutate(lm_all_rep[, c("state", "intercept", "slope")],
                         id = "pooled regression"),
                  mutate(state_lm_df[, c("state", "intercept", "slope")],
                         id = "unpooled regression"),
                  mutate(multilevel_lmer_df[, c("state", "intercept", "slope")],
                         id = "multilevel model")) %>%
```

```
  mutate(id = factor(id, levels = c("pooled regression",
                                    "unpooled regression",
                                    "multilevel model")))
```

```
ggplot(cdi) +
  geom_point(aes(x = phg, y= pci)) +
  geom_abline(data = combined,
              aes(slope = slope, intercept= intercept, group = state,
                  color = id))+
  facet_wrap(~ state) +
  theme_bw() +
  labs(color = "model type")
```

(b)

See figure 2.

```
# multilevel model
multilevel_res_fit <- data.frame(fitted = fitted(multilevel_lmer),
                                 residual = resid(multilevel_lmer),
                                 state = cdi$state)

# multilevel model, part II
source("../residual-functions.r")
multilevel_res_fit2 <- data.frame(fitted = yhat.cond(multilevel_lmer),
                                  residual = r.cond(multilevel_lmer),
                                  state = cdi$state)

testthat::expect_equivalent(multilevel_res_fit, multilevel_res_fit2) # no error if true
```

```
multilevel_res_fit %>%
ggplot() +
  geom_point(aes(x = fitted, y = residual)) +
  facet_wrap(~ state) +
  geom_hline(yintercept = 0) +
  theme_bw() +
  labs(y = "conditional residual",
       x = TeX("conditional $\\hat{y}$"))
```

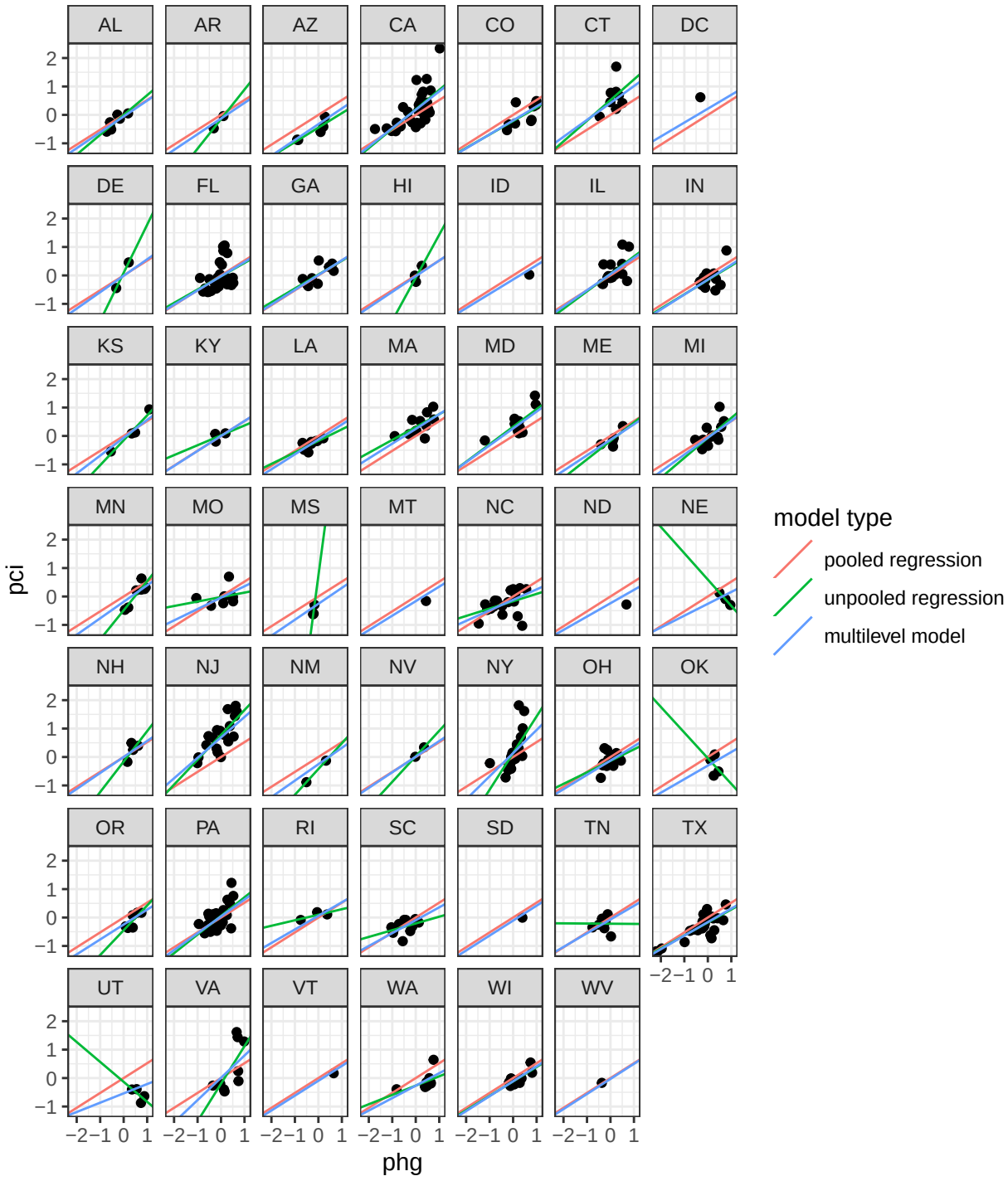


Figure 1: Problem 1a: Different linear models to compare per capital income to percent high school graduates
.

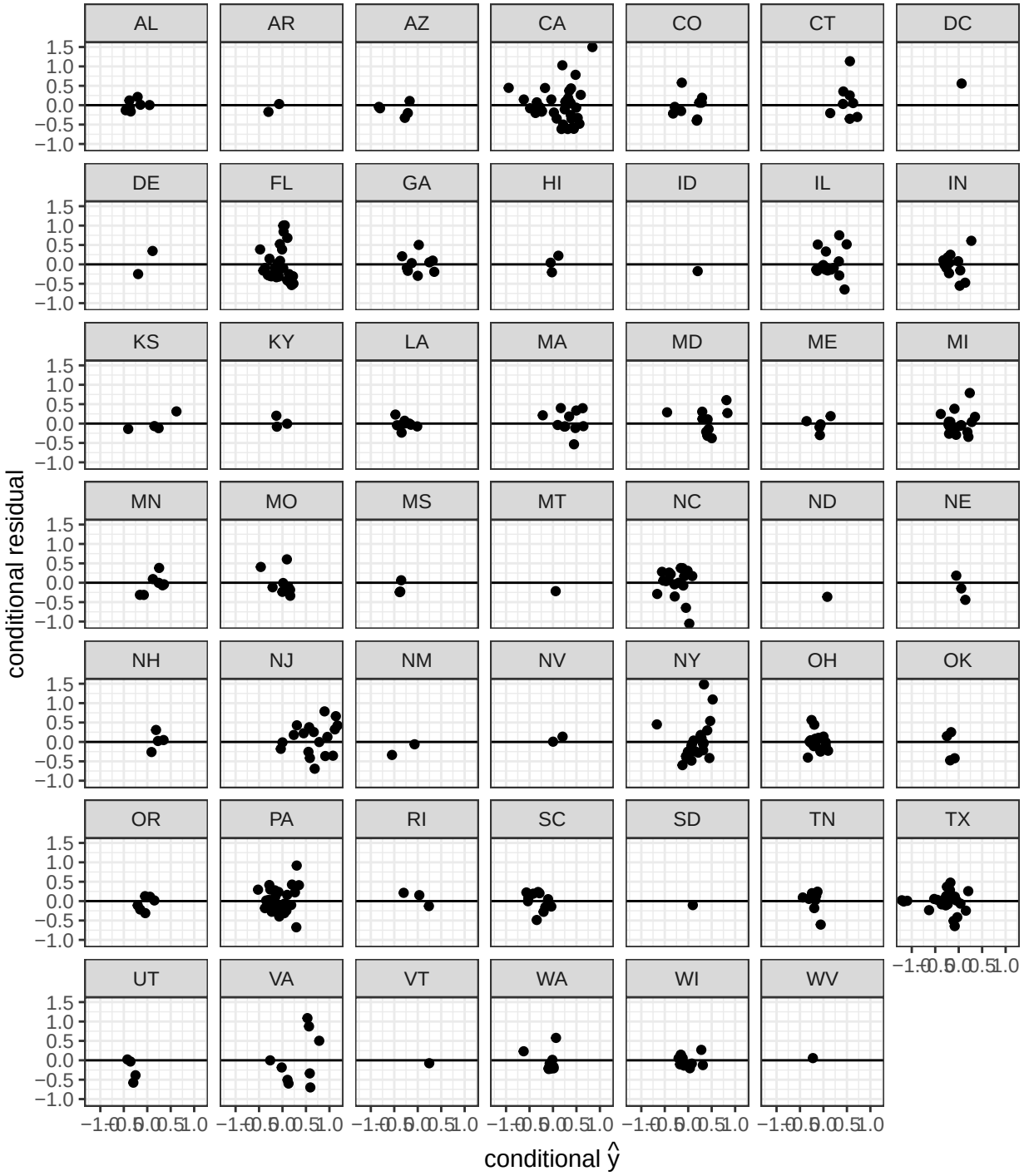


Figure 2: Problem 1b: Examinig the conditional residual vs conditional $\hat{y}$ for multilevel model .

Problem 2

NOTES:

(1) Instead of giving "full" solutions for #2, I'm just going to provide some suggestive code... Mostly I am not providing interpretations– you will need to do that in your IDMRAD paper for the college, after you are satisfied with your technical appendix!

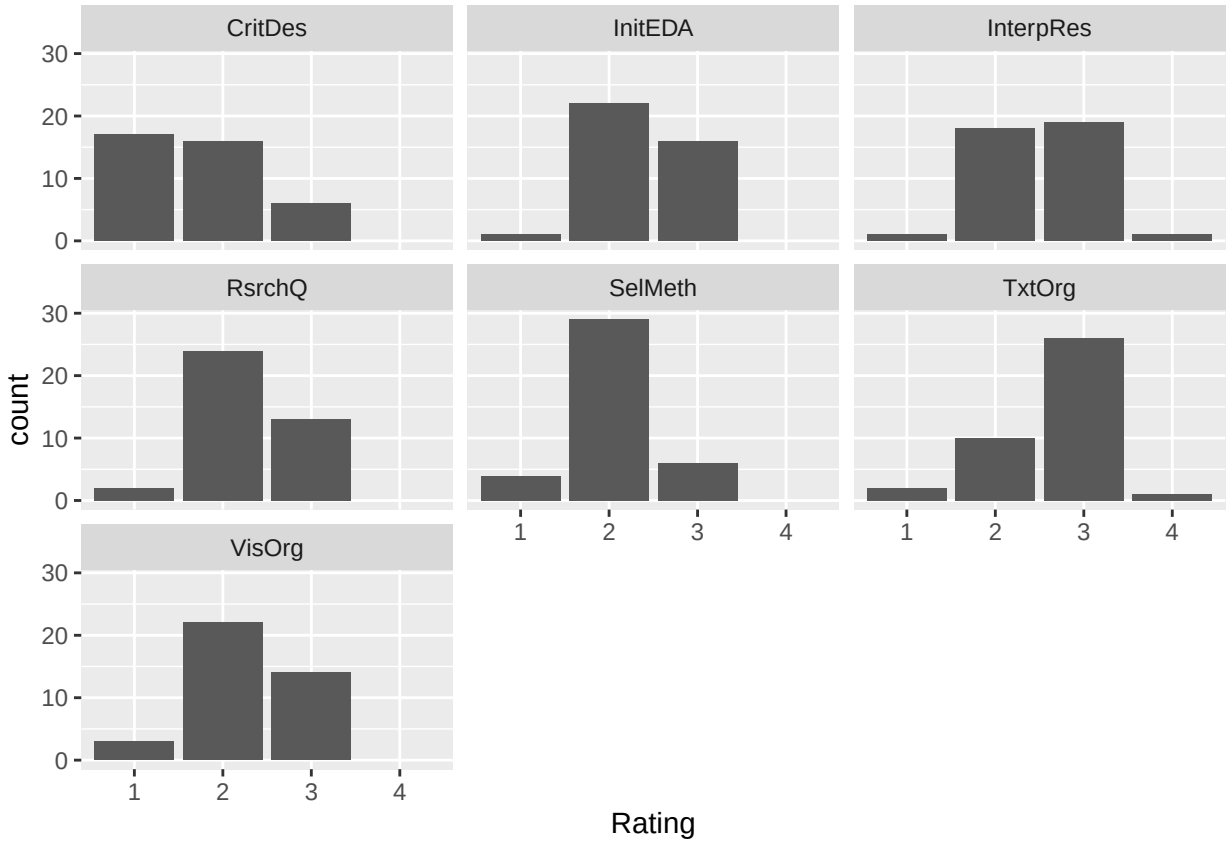
(2) There are many pages here because I couldn't suppress some of the output from fitLMER.fnc() from library(LMERconveniencefunctions).

```
library(lme4)
library(arm)
library(ggplot2)

##
## Read the data in wide and tall formats...
ratings <- read.csv("ratings.csv",header=T)
tall <-  read.csv("tall.csv",header=T)

##
## Some preliminary exploration of the data with View(), summary(), table(), etc.,
## omitted here -- BUT it is important that you do it, so you can catch things
## like missing data (however it is coded), not all raters using all categories, etc.

##
## Take care that all ratings run from 1 to 4,
## whether or not rater used all categories...
tall$Rating <- factor(tall$Rating,levels=1:4)
for (i in unique(tall$Rubric)) {
  ratings[,i] <- factor(ratings[,i],levels=1:4)
}

##
## Note that in the "ratings" data frame, the missing "Sex"
## value is "--" while in the "tall" data frame it is ""
## (a string of length 0).
##
## Make the "tall" be consistent with the "ratings" coding.
tall$Sex[nchar(tall$Sex)==0] <- "--"

##
## Extract the reduced data set with the 13 artifacts that all 3 raters saw...
ratings.13 <- ratings[grep("O",ratings$Artifact),]
tall.13 <-  tall[grep("O",tall$Artifact),]
```

2(a) Is the distribution of ratings for each rubrics pretty much indistinguishable from the other rubrics, or are there rubrics that tend to get especially high or low ratings? Is the distribution of ratings given by each rater pretty much indistinguishable from the other raters, or are there raters that tend to give especially high or low ratings?

Here are some ideas to compare distributions across Rubrics. I leave other possibilities, pretty formatting (wtih kable, etc.), and smart interpretations up to you...

```
##
## Bar plots for the reduced data set
g <- ggplot(tall.13,aes(x = Rating)) +
  facet_wrap( ~ Rubric) +
  geom_bar()

g
```

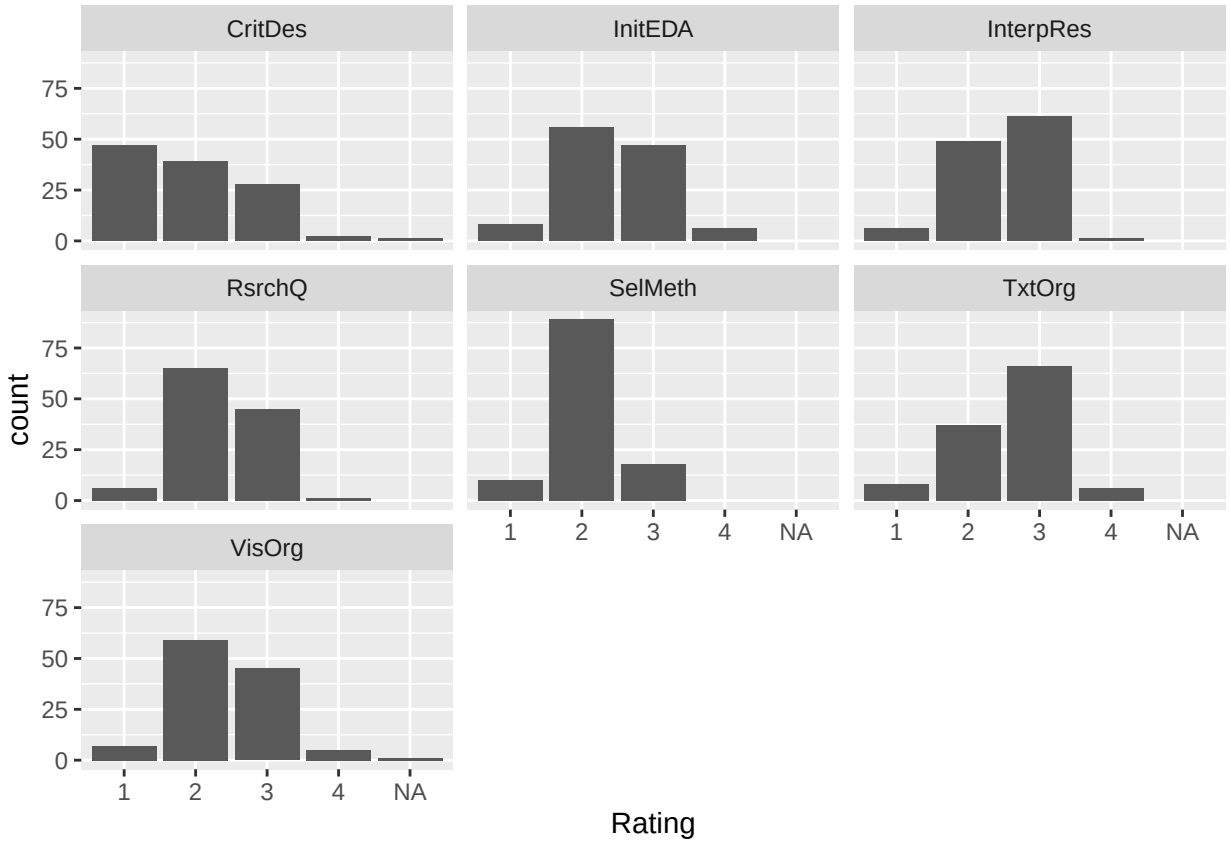



```
##
## Table of counts might make a nice supplement
tmp <- data.frame(lapply(split(tall.13$Rating,tall.13$Rubric),summary))
row.names(tmp) <- paste("Rating",1:4)

tmp
```

```
##          CritDes InitEDA InterpRes RsrchQ SelMeth TxtOrg VisOrg
## Rating 1      17       1         1      2       4       2       3
## Rating 2      16      22        18     24      29      10      22
## Rating 3       6      16        19     13       6      26      14
## Rating 4       0       0         1      0       0       1       0
```

```
##
## Barplots for full data set
g <- ggplot(tall,aes(x = Rating)) +
  facet_wrap( ~ Rubric) +
  geom_bar()
```

```
g
```



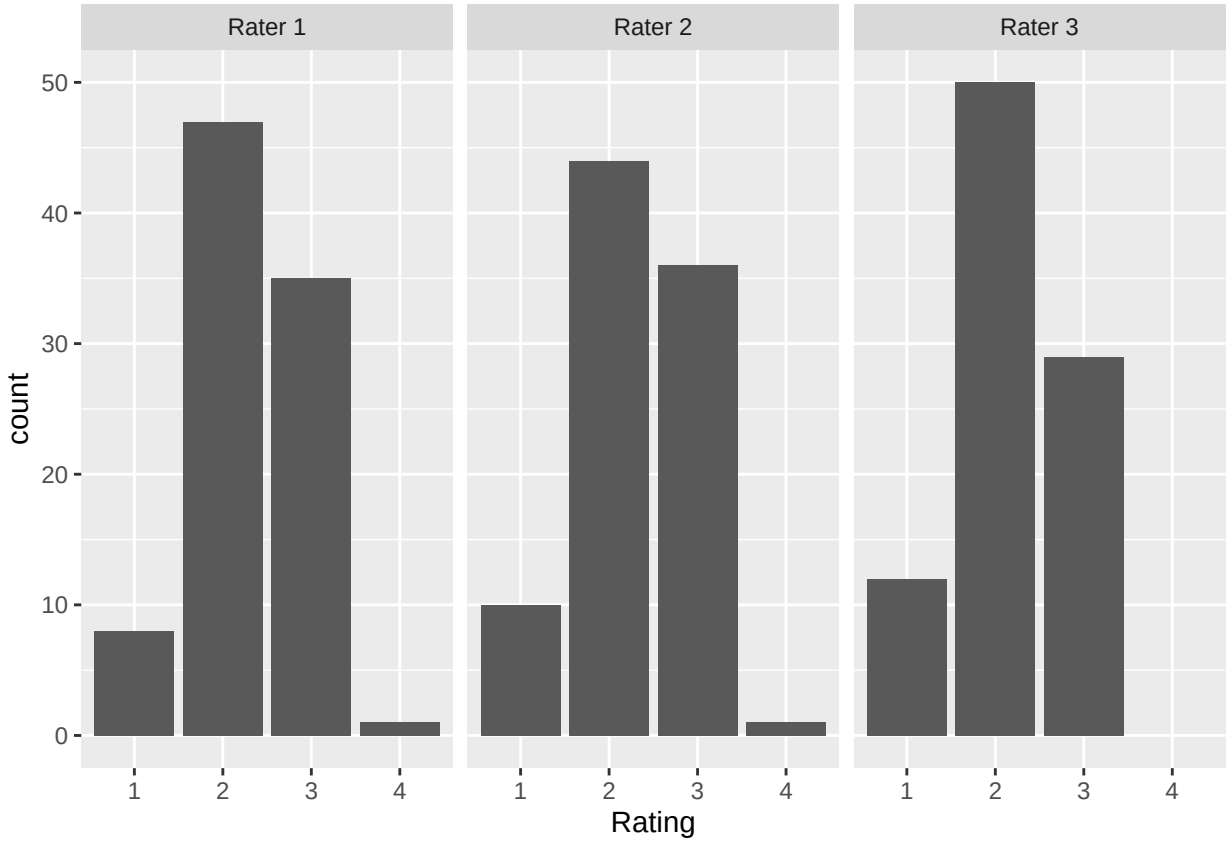

```
##
## Table of counts again.  A bit pesky since there are NA's...
tmp0 <- lapply(split(tall$Rating,tall$Rubric),summary)
tmp <- data.frame(matrix(0,nrow=5,ncol=7))  ## seven rubrics...
names(tmp) <- names(tmp0)
row.names(tmp) <- c(paste("Rating",1:4),"<NA>")
for (i in names(tmp0)) {
  tmp[,i] <- tmp[,i] + c(tmp0[[i]],0)[1:5]
}

tmp
```

```
##          CritDes InitEDA InterpRes RsrchQ SelMeth TxtOrg VisOrg
## Rating 1      47       8         6      6      10      8      7
## Rating 2      39      56        49     65      89     37     59
## Rating 3      28      47        61     45      18     66     45
## Rating 4       2       6         1      1       0      6      5
## <NA>           1       0         0      0       0      0      1
```

And here are some idea to compare distributions across Raters. Again, I leave other possibilities, attractive formatting (with kable, changing height of graphs, etc.), and smart intepretations up to you...

```
##
## Needed to make the title of each facet more human-readable...
rater.name <- function(x) { paste("Rater",x) }
```

```
##
## Barplots for reduced data...
g <- ggplot(tall.13,aes(x = Rating)) +
  facet_wrap( ~ Rater, labeller=labeller(Rater=rater.name)) +
  geom_bar()

g
```

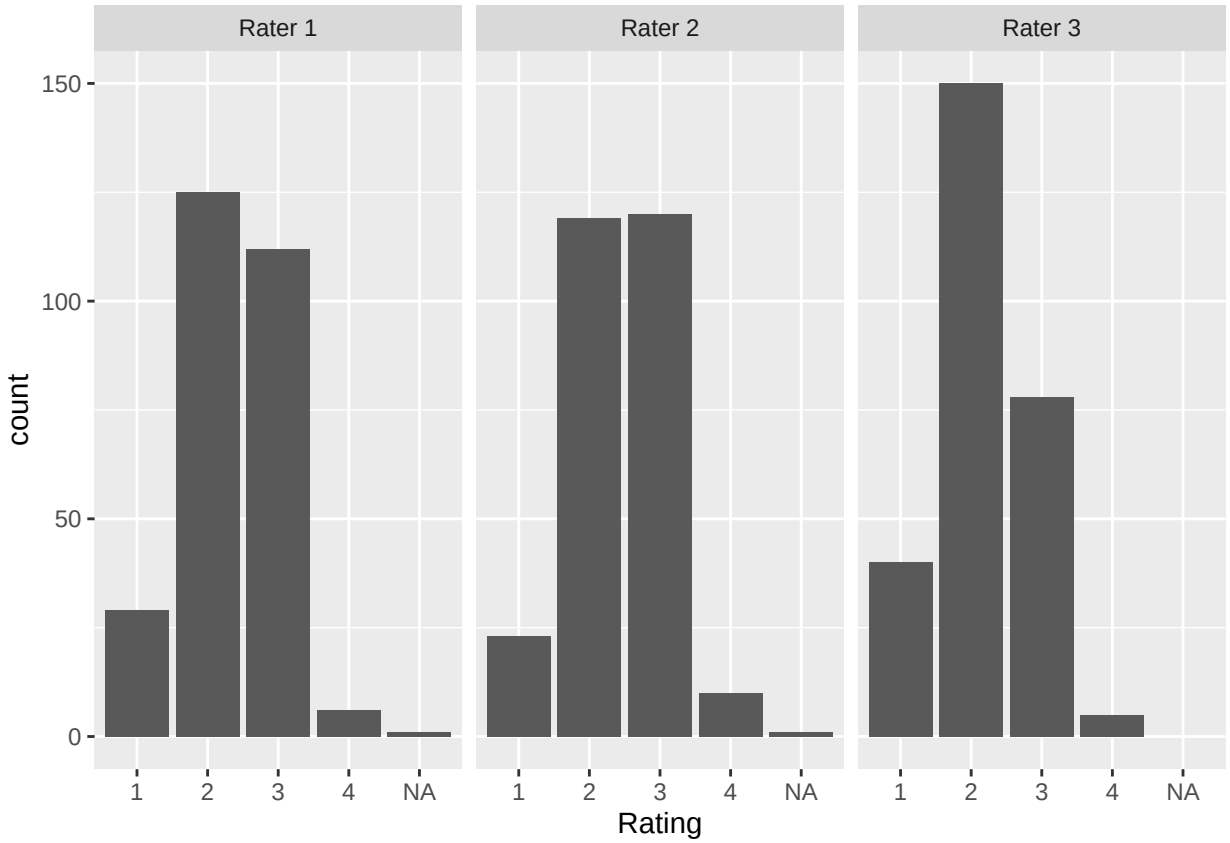



```
##
## Corresponding table of counts...
tmp <- data.frame(lapply(split(tall.13$Rating,tall.13$Rater),summary))
row.names(tmp) <- paste("Rating",1:4)
names(tmp) <- paste("Rater",1:3)

tmp
```

```
##          Rater 1 Rater 2 Rater 3
## Rating 1       8      10      12
## Rating 2      47      44      50
## Rating 3      35      36      29
## Rating 4       1       1       0
```

```
##
## Barplots for full data...
g <- ggplot(tall,aes(x = Rating)) +
  facet_wrap( ~ Rater, labeller=labeller(Rater=rater.name)) +
```

```
  geom_bar()

g
```




```
##
## Corresponding table of counts...
tmp0 <- lapply(split(tall$Rating,tall$Rater),summary)
tmp <- data.frame(matrix(0,nrow=5,ncol=3))  ## three raters...
names(tmp) <- names(tmp0)
row.names(tmp) <- c(paste("Rating",1:4),"<NA>")
for (i in names(tmp0)) {
  tmp[,i] <- tmp[,i] + c(tmp0[[i]],0)[1:5]
}
names(tmp) <- paste("Rater",1:3)
tmp
```

```
##          Rater 1 Rater 2 Rater 3
## Rating 1      29      23      40
## Rating 2     125     119     150
## Rating 3     112     120      78
## Rating 4       6      10       5
## <NA>           1       1       0
```

Now, those NA's have me curious...

```
tall[apply(tall,1,function(x){any(is.na(x))}),]
```

```
##       X Rater Artifact Repeated Semester Sex  Rubric Rating
## 161 161     2       45        0      S19   F CritDes   <NA>
## 684 684     1      100        0      F19   F  VisOrg   <NA>
```

```
ratings[ratings$Sex=="--",]
```

```
##   X Rater Sample Overlap Semester Sex RsrchQ CritDes InitEDA SelMeth InterpRes
## 5 5     3      5      NA     Fall  --      3       3       3       3         3
##   VisOrg TxtOrg Artifact Repeated
## 5      3      3        5        0
```

First, note that none of the missing values occur in the smaller 13-rubric data set (how can we tell this from the output above?). So we don't have to worry about missing data at all in analyses that just involve this smaller data set.

Second, in any modeling that we do, the "Rating" is the outcome variable, so R will just drop the two observations with missing Rating values. This will mean that the "full" data sets may be different for models that involve different rubrics: For models involving five of the rubrics we will get all the data from all the raters, but for models involving CritDes we would be missing a rating from Rater 2, and for models involving VisOrg we would be missing a rating from Rater 1. We need to be vigilant about when these differences actually occur, since they could undermine some model comparisons (different data sets).

Third, we will also have to be careful of the missing "Sex" value (currently coded as "–". If we coded it as NA, then R would drop it from models that have Sex as a predictor, which would make comparing models with and without Sex as a predictor more difficult (different data sets!). We could just drop this student from all analyses, but it seems like a waste to lose that data. We could code it as "F" or "M" if we had a convincing justification for doing so, but since I don't have convincing justification (do you??), I'm just going to leave it as a third "Sex" category for now...

2(b) For each rubric, do the raters generally agree on their scores? If not, is there one rater who disagrees with the others? Or do they all disagree?

Here are some coding ideas. NOTE: Once again, I leave other ideas, attractive formatting, and interpretations (actually answering the question!) to you...

```
##
## useful preliminaries
Rubric.names <- sort(unique(tall$Rubric))

##
## First we examine the 13 "common" artifacts that all 3 raters saw...

##
## Note: extracting sig^2 and tau^2 from the fitted lmer() object took a little
## spelunking...  First I fitted a "test model"
##
## tmp <- lmer(as.numeric(Rating) ~ 1 + (1|Artifact),data=tall.13[tall.13$Rubric=="RsrchQ",])
##
## then I looked at names(summary(tmp)) and tried to see what was under each name
## by looking at summary(tmp)$methTitle, summary(tmp)$objClass, etc. for all the
## names I found.  I quickly found summary(tmp)$sigma, which can be squared to get
## sig^2.  It took more exploring with summary(tmp)$varcor to get tau^2...
##

ICC.vec <- NULL
for (i in Rubric.names) {
```

```
  tmp <- lmer(as.numeric(Rating) ~ 1 + (1|Artifact), data=tall.13[tall.13$Rubric==i,])
  sig2 <- summary(tmp)$sigma^2
  tau2 <- attr(summary(tmp)$varcor[[1]],"stddev")^2
  ICC <- tau2 / (tau2 + sig2)
  ICC.vec <- c(ICC.vec,ICC)
}
names(ICC.vec) <- Rubric.names

agreement.results <- cbind(ICC.common=ICC.vec,"       a12"=0,a23=0,a13=0)

agreement.tables <- as.list(rep(NA,7))
names(agreement.tables) <- Rubric.names

for (i in Rubric.names) {
  r12 <- data.frame(r1=factor(ratings.13[ratings.13$Rater==1,i],levels=1:4),
                    r2=factor(ratings.13[ratings.13$Rater==2,i],levels=1:4),
                    a1=ratings.13[ratings.13$Rater==1,"Artifact"],
                    a2=ratings.13[ratings.13$Rater==2,"Artifact"])
  if(any(r12[,3]!=r12[,4])) { stop(paste("Rater 1-2 Artifact mismatch on rubric",i)) }
  a12 <- mean(r12[,1]==r12[,2])
  r12 <- table(r12[,1:2])  ## print this to see how much agreement there is among raters 1-2

  r23 <- data.frame(r2=factor(ratings.13[ratings.13$Rater==2,i],levels=1:4),
                    r3=factor(ratings.13[ratings.13$Rater==3,i],levels=1:4),
                    a2=ratings.13[ratings.13$Rater==2,"Artifact"],
                    a3=ratings.13[ratings.13$Rater==3,"Artifact"])
  if(any(r23[,3]!=r23[,4])) { stop(paste("Rater 2-3 Artifact mismatch on rubric",i)) }
  a23 <- mean(r23[,1]==r23[,2])
  r23 <- table(r23[,1:2])  ## print this to see how much agreement there is among raters 2-3

  r13 <- data.frame(r1=factor(ratings.13[ratings.13$Rater==1,i],levels=1:4),
                    r3=factor(ratings.13[ratings.13$Rater==3,i],levels=1:4),
                    a1=ratings.13[ratings.13$Rater==1,"Artifact"],
                    a3=ratings.13[ratings.13$Rater==3,"Artifact"])
  if(any(r13[,3]!=r13[,4])) { stop(paste("Rater 1-3 Artifact mismatch on rubric",i)) }
  a13 <- mean(r13[,1]==r13[,2])
  r13 <- table(r13[,1:2])  ## print this to see how much agreement there is among raters 1-3

  agreement.results[i,2:4] <- c(a12,a23,a13)

  agreement.tables[[i]] <- list(r12,r23,r13)

}
round(agreement.results,2)
```

```
##           ICC.common        a12  a23  a13
## CritDes         0.57       0.54 0.69 0.62
## InitEDA         0.49       0.69 0.85 0.54
## InterpRes       0.23       0.62 0.62 0.54
## RsrchQ          0.19       0.38 0.54 0.77
## SelMeth         0.52       0.92 0.69 0.62
## TxtOrg          0.14       0.69 0.54 0.62
## VisOrg          0.59       0.54 0.77 0.77
```

```
##

if (F) { print(agreement.tables) }

## change to "if (T)" to get a crude printout of all tables...

##
## Now add in ICC's calculated from all the data...

ICC.vec <- NULL
for (i in Rubric.names) {

  tmp <- lmer(as.numeric(Rating) ~ 1 + (1|Artifact), data=tall[tall$Rubric==i,])
  sig2 <- summary(tmp)$sigma^2
  tau2 <- attr(summary(tmp)$varcor[[1]],"stddev")^2
  ICC <- tau2 / (tau2 + sig2)
  ICC.vec <- c(ICC.vec,ICC)
}
names(ICC.vec) <- Rubric.names

agreement.results <- cbind(ICC.alldata=ICC.vec,agreement.results)

round(agreement.results,2)
```

```
##           ICC.alldata ICC.common        a12  a23  a13
## CritDes          0.67       0.57       0.54 0.69 0.62
## InitEDA          0.69       0.49       0.69 0.85 0.54
## InterpRes        0.22       0.23       0.62 0.62 0.54
## RsrchQ           0.21       0.19       0.38 0.54 0.77
## SelMeth          0.47       0.52       0.92 0.69 0.62
## TxtOrg           0.19       0.14       0.69 0.54 0.62
## VisOrg           0.66       0.59       0.54 0.77 0.77
```

2(c) More generally, how are the various factors in this experiement (Rater, Semester, Sex, Repeated, Rubric) related to the ratings? Do the factors interact in any interesting ways?

There is a lot here, so I will break it up into "subsections":

- 2(c)(i): Adding fixed effects to the seven rubric-specific models using just the data from the 13 common artifacts that al three raters saw
- 2(c)(ii): Adding fixed effects to the seven rubric-specific models using all the data
- 2(c)(iii): Trying interactions and new random effects for the seven rubric specific models using all the data
- 2(c)(iv): Trying to add fixed effects, interactions, and new random effects to the "combined" model Rating ~ 1 + (0 + Rubric|Artifact), using all the data.

I want to note that some of the output below is pretty verbose (that's why there are so many pages!), and some of it runs off the edge of the page. I dislike both of these (verbose output, and output you can't read because it runs off the page) but I haven't been able to fix it. Some R functions just do not respect page layout attributes, and some functions don't have a 'turn off verbose output' option.

Also, some of the models take lmer considerable time to fit. You may think your computer has hung. It probably has not hung; some models take a minute or two to fit. If anything is taking more than 5 minutes,

that's the time to consider whether you really need that model.

2(c)(i): Adding fixed effects to the seven rubric-specific models using just the data from the 13 common artifacts that al three raters saw

First, we try to add fixed effects to our seven rubric-specific models... In principle it will matter whether we use only the data reduced to the 13 common artifacts, or the full data set.

I will start with the reduced data (so of course I can't check "repeated" on this reduced data—why not?).

```
library(LMERConvenienceFunctions)
library(RLRsim)

## Experiments before trying "production" code:
##
## I started by fitting a single model and trying fitLMER.fnc() on it.
##
## As stated on Piazza, fitLMER.fnc() doesn't seem to like intecept-only
## as the final model, and so for models including rater, I removed
## the intercept, so that (effectively) rater would always be in the
## model, and fitLMER.fnc() wouldn't complain.
##
## So my starting model for experimenting was

tmp <- lmer(as.numeric(Rating) ~ -1 + as.factor(Rater) +
              Semester + Sex + (1|Artifact),
            data=tall.13[tall.13$Rubric=="RsrchQ",],REML=FALSE)

##
## Since backwards-elimination always involves nested models, I could
## use t-tests, F-tests or likelihood ratio tests to eliminate fixed
## effects.  The default for fitLMER.fnc() is to use t-tests with a
## threshold of 2 (cutoff for the t-statistic, rather than a cutoff
## like 0.05 for the p-value). This is good enough for me.  I will also
## force fitLMER.fnc() to fit using ML rather than REML, so the
## t-tests are as close to correct as I can get.
##
## So a typical function call would be

tmp.back_elim <- fitLMER.fnc(tmp,set.REML.FALSE = TRUE,log.file.name = FALSE)
```

```
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.7355 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.279 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
```

```
## ========================================================
## ===            forwardfitting random effects       ===
## ========================================================
##  ===       random slopes       ===
## ========================================================
## ===            re-backfitting fixed effects        ===
## ========================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
```

```
## (It would be nice to eliminate the verbose narrative here, but I don't see
## a way to do that.  Just stuck with it I guess...)

## Anyway, backwards elimination with fitLMER.fnc() yields a model
## with raters only:

formula(tmp.back_elim)
```

```
## as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
```

```
## (Note that raters will always remain in the model, since raters have to rate
## in a category >= 1, and the t-tests compares each rater's average rating to 0.
## Unless the sample size is really small, this should always yield a significant
## coefficient for each rater dummy variable in the model.)

## The estimates for raters don't look that different from each other,
## so we can test to see if they are different by comparing with the
## intercept-only model

tmp.int_only <- update(tmp.back_elim, . ~ . + 1 - as.factor(Rater))

anova(tmp.int_only,tmp.back_elim)
```

```
## Data: tall.13[tall.13$Rubric == "RsrchQ", ]
## Models:
## tmp.int_only: as.numeric(Rating) ~ (1 | Artifact)
## tmp.back_elim: as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
##               npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## tmp.int_only     3 69.457 74.447 -31.728   63.457
## tmp.back_elim    5 72.018 80.335 -31.009   62.018 1.4391  2      0.487
```

```
## Again the models are nested so I really only need to look at the p-value
## from the likelihood ratio chi-squared test.  A little fooling around with
## names(anova(tmp.int_only,tmp.back_elim)), etc., shows that I can get this
## as

anova(tmp.int_only,tmp.back_elim)$"Pr(>Chisq)"[2]
```

```
## [1] 0.4869707
```

```
## it looks like the intercept-only model is adequate here (the p-value
## is much greater than 0.05 or any other common significance level).

## Note: since no main effects were retained, there's really no reason to
```

```r
## check for interactions.

## Now, I need to code this into a loop so I don't have to do everything by hand...


Rubric.names <- sort(unique(tall$Rubric))

model.formula.13 <- as.list(rep(NA,7))
names(model.formula.13) <- Rubric.names

## There will be a lot of output from fitLMER.fnc() here... Sorry!

for (i in Rubric.names) {

  ## fit each base model
  rubric.data <- tall.13[tall.13$Rubric==i,]
  tmp <- lmer(as.numeric(Rating) ~ -1 + as.factor(Rater) +
              Semester + Sex + (1|Artifact),
            data=rubric.data,REML=FALSE)

  ## do backwards elimination
  tmp.back_elim <- fitLMER.fnc(tmp,set.REML.FALSE = TRUE,log.file.name = FALSE)

  ## check to see if the raters are significantly different from one another
  tmp.single_intercept <- update(tmp.back_elim, . ~ . + 1 - as.factor(Rater))
  pval <- anova(tmp.single_intercept,tmp.back_elim)$"Pr(>Chisq)"[2]

  ## choose the best model
  if (pval<=0.05) {
    tmp_final <- tmp.back_elim
  } else {
    tmp_final <- tmp.single_intercept
  }

  ## and add to list...
  model.formula.13[[i]] <- formula(tmp_final)

}
```

```
## ========================================================
## ===              backfitting fixed effects           ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.2229 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.1826 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ========================================================
```

```
## ===            forwardfitting random effects       ===
## ============================================================
##  ===       random slopes       ===
## ============================================================
## ===            re-backfitting fixed effects        ===
## ============================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## ============================================================
## ===              backfitting fixed effects         ===
## ============================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.8137 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.6429 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ============================================================
## ===            forwardfitting random effects       ===
## ============================================================
##  ===       random slopes       ===
## ============================================================
## ===            re-backfitting fixed effects        ===
## ============================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## ============================================================
## ===              backfitting fixed effects         ===
## ============================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.8294 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.2947 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ============================================================
## ===            forwardfitting random effects       ===
## ============================================================
```

```
##  ===       random slopes       ===
## ========================================================
## ===            re-backfitting fixed effects        ===
## ========================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.7355 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.279 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===            forwardfitting random effects       ===
## ========================================================
##  ===       random slopes       ===
## ========================================================
## ===            re-backfitting fixed effects        ===
## ========================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.9383 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.4287 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===            forwardfitting random effects       ===
## ========================================================
##  ===       random slopes       ===
## ========================================================
```

```
## ===            re-backfitting fixed effects        ===
## ========================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.5358 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.1319 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===            forwardfitting random effects       ===
## ========================================================
##  ===       random slopes       ===
## ========================================================
## ===            re-backfitting fixed effects        ===
## ========================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.1922 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.1078 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===            forwardfitting random effects       ===
## ========================================================
##  ===       random slopes       ===
## ========================================================
## ===            re-backfitting fixed effects        ===
## ========================================================
```

```
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
```

```
## see what "final models" we got...
model.formula.13
```

```
## $CritDes
## as.numeric(Rating) ~ (1 | Artifact)
##
## $InitEDA
## as.numeric(Rating) ~ (1 | Artifact)
##
## $InterpRes
## as.numeric(Rating) ~ (1 | Artifact)
##
## $RsrchQ
## as.numeric(Rating) ~ (1 | Artifact)
##
## $SelMeth
## as.numeric(Rating) ~ (1 | Artifact)
##
## $TxtOrg
## as.numeric(Rating) ~ (1 | Artifact)
##
## $VisOrg
## as.numeric(Rating) ~ (1 | Artifact)
```

So, it looks like we don't need to add any fixed effects or interactions to the models for each rubric, using only the data reduced to the 13 common rubrics.

2(c)(ii): Adding fixed effects to the seven rubric-specific models using all the data

Now let's try with the full data...

```
Rubric.names <- sort(unique(tall$Rubric))

## Note: Now the missing ratings become important.  We want to use the same data
## set for every model fit and model comparison.  I am going to eliminate by
## hand the two observations with missing data, and only do fitting and comparison
## on this "slightly" reduced data set.

tall[c(161,684),] ## just to check that these are the rows with missing ratings...
```

```
##       X Rater Artifact Repeated Semester Sex  Rubric Rating
## 161 161     2       45        0      S19   F CritDes   <NA>
## 684 684     1      100        0      F19   F  VisOrg   <NA>
```

```
tall.nonmissing <- tall[-c(161,684),]  ## now delete them...

## I can't think of a good justification for imputing the "Sex" of the student who
## didn't report this to either M or F, and leaving it as "--" makes the models
## harder to interpret.  So I will eliminate that person from the data set also...

tall.nonmissing[tall.nonmissing$Sex=="--",]  ## check which rows will be eliminated
```

```
##       X Rater Artifact Repeated Semester Sex    Rubric Rating
## 5     5     3        5        0      F19  --    RsrchQ      3
## 122 122     3        5        0      F19  --   CritDes      3
## 239 239     3        5        0      F19  --   InitEDA      3
## 356 356     3        5        0      F19  --   SelMeth      3
## 473 473     3        5        0      F19  -- InterpRes      3
## 590 590     3        5        0      F19  --    VisOrg      3
## 707 707     3        5        0      F19  --    TxtOrg      3
```

```
tall.nonmissing <- tall.nonmissing[tall.nonmissing$Sex!="--",] ## eliminate them

model.formula.alldata <- as.list(rep(NA,7))
names(model.formula.alldata) <- Rubric.names

## There will be a lot of output from fitLMER.fnc() here... Sorry!

for (i in Rubric.names) {

  ## fit each base model
  rubric.data <- tall.nonmissing[tall.nonmissing$Rubric==i,]
  tmp <- lmer(as.numeric(Rating) ~ -1 + as.factor(Rater) +
              Semester + Sex + (1|Artifact),
            data=rubric.data,REML=FALSE)

  ## do backwards elimination
  tmp.back_elim <- fitLMER.fnc(tmp,set.REML.FALSE = TRUE,log.file.name = FALSE)

  ## check to see if the raters are significantly different from one another
  tmp.single_intercept <- update(tmp.back_elim, . ~ . + 1 - as.factor(Rater))
  pval <- anova(tmp.single_intercept,tmp.back_elim)$"Pr(>Chisq)"[2]

  ## choose the best model
  if (pval<=0.05) {
    tmp_final <- tmp.back_elim
  } else {
    tmp_final <- tmp.single_intercept
  }

  ## and add to list...
  model.formula.alldata[[i]] <- formula(tmp_final)

}
```

```
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effects" is
## TRUE
```

```
## ============================================================
## ===              backfitting fixed effects               ===
## ============================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.7154 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
```

```
##     p-value for term "Sex" = 0.5297 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## =======================================================
## ===            forwardfitting random effects       ===
## =======================================================
##  ===       random slopes       ===
## =======================================================
## ===            re-backfitting fixed effects        ===
## =======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## refitting model(s) with ML (instead of REML)
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effects" is
## TRUE
## =======================================================
## ===              backfitting fixed effects         ===
## =======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.8802 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.7402 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## =======================================================
## ===            forwardfitting random effects       ===
## =======================================================
##  ===       random slopes       ===
## =======================================================
## ===            re-backfitting fixed effects        ===
## =======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## refitting model(s) with ML (instead of REML)
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effects" is
## TRUE
## =======================================================
## ===              backfitting fixed effects         ===
```

```
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.608 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.5312 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===            forwardfitting random effects       ===
## ========================================================
##  ===       random slopes       ===
## ========================================================
## ===            re-backfitting fixed effects        ===
## ========================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## refitting model(s) with ML (instead of REML)
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effects" is
## TRUE
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.6166 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.3987 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===            forwardfitting random effects       ===
## ========================================================
##  ===       random slopes       ===
## ========================================================
## ===            re-backfitting fixed effects        ===
## ========================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
```

```
##   nothing to prune
## refitting model(s) with ML (instead of REML)
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effects" is
## TRUE
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.1935 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===            forwardfitting random effects       ===
## ========================================================
##  ===       random slopes       ===
## ========================================================
## ===            re-backfitting fixed effects        ===
## ========================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## refitting model(s) with ML (instead of REML)
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effects" is
## TRUE
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.5041 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.205 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===            forwardfitting random effects       ===
## ========================================================
##  ===       random slopes       ===
## ========================================================
## ===            re-backfitting fixed effects        ===
## ========================================================
## processing model terms of interaction level 1
```

```
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effects" is
## TRUE
```

```
## ============================================================
## ===              backfitting fixed effects             ===
## ============================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.2158 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.3523 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ============================================================
## ===            forwardfitting random effects           ===
## ============================================================
##  ===       random slopes       ===
## ============================================================
## ===            re-backfitting fixed effects            ===
## ============================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
```

```
## refitting model(s) with ML (instead of REML)
```

```
## see what "final models" we got...
model.formula.alldata
```

```
## $CritDes
## as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
##
## $InitEDA
## as.numeric(Rating) ~ (1 | Artifact)
##
## $InterpRes
## as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
##
## $RsrchQ
## as.numeric(Rating) ~ (1 | Artifact)
##
## $SelMeth
## as.numeric(Rating) ~ as.factor(Rater) + Semester + (1 | Artifact) -
##     1
```

```
##
## $TxtOrg
## as.numeric(Rating) ~ (1 | Artifact)
##
## $VisOrg
## as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
```

2(c)(iii): Trying interactions and new random effects for the seven rubric specific models using all the data

Now we see there are some differences among the models: For `InitEDA`, `RsrchQ` and `SelMeth`, the models are just the simple random-intercept models. For the other four, the models are a little more complex. We should examine each of these 4 models to see (a) if the fixed effects make sense to us; and (2) if there are any interactions or additional random effects to consider.

I will illustrate a bit of this with the model for `SelMeth`, and leave the rest to you.

```
## refit the model and check on the t-statistics -- do all the variables matter?
fla <- formula(model.formula.alldata[["SelMeth"]])
tmp <- lmer(fla,data=tall.nonmissing[tall.nonmissing$Rubric=="SelMeth",])
round(summary(tmp)$coef,2)  ## fixed effects and their t-values
```

```
##                   Estimate Std. Error t value
## as.factor(Rater)1     2.25       0.08   29.99
## as.factor(Rater)2     2.23       0.07   29.99
## as.factor(Rater)3     2.03       0.08   27.03
## SemesterS19          -0.36       0.10   -3.66
```

```
## apparently they do.

## now check to make sure we really need "Rater" as a factor...
tmp.single_intercept <- update(tmp, . ~ . + 1 - as.factor(Rater))
anova(tmp.single_intercept,tmp)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: tall.nonmissing[tall.nonmissing$Rubric == "SelMeth", ]
## Models:
## tmp.single_intercept: as.numeric(Rating) ~ Semester + (1 | Artifact)
## tmp: as.numeric(Rating) ~ as.factor(Rater) + Semester + (1 | Artifact) - 1
##                      npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## tmp.single_intercept    4 145.07 156.08 -68.534   137.07
## tmp                     6 142.05 158.58 -65.027   130.05 7.0146  2    0.02998 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## looks like we do, so we keep "tmp" as our best model so far...

## now let's check for fixed-effect interactions... Since only Rater and Semester
## are involved, we only need to examine Rater*Semester

tmp.fixed_interactions <- update(tmp, . ~ . + as.factor(Rater)*Semester - Semester)
## I've specified the model so that I can see (a) a different intercept for each
## rater, and (b) a different semester effect for each rater.

anova(tmp,tmp.fixed_interactions)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: tall.nonmissing[tall.nonmissing$Rubric == "SelMeth", ]
## Models:
## tmp: as.numeric(Rating) ~ as.factor(Rater) + Semester + (1 | Artifact) - 1
## tmp.fixed_interactions: as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) + as.factor(Rater):Se
##                        npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## tmp                       6 142.05 158.58 -65.027   130.05
## tmp.fixed_interactions    8 143.46 165.49 -63.731   127.46 2.592  2     0.2736
```

```
## Looks like the fixed-effect interactions are not needed; again we keep
## "tmp" as our best model so far...

## Finally we check for random effects.  We should only add random effects that
## are also present as fixed effects.  This means, for this model, we should try
## (Rater|Artifact) and (Semester|Artifact).

## I will show how to test these with exactRLRT()...

## Testing (Semester|Artifact)...

m0 <- tmp                                       ## Null hypothesis
mA <- update(m0, . ~ . + (Semester|Artifact))   ## Alternative hypotheses
```

```
## Error: number of observations (=116) <= number of random effects (=180) for term (Semester | Artifac
```

```
m  <- update(mA, . ~ . - (1|Artifact))          ## Model with only the new R.E.
```

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'object' in selecting a method
```

```
exactRLRT(m0=m0,mA=mA,m=m)
```

```
## Error in exactRLRT(m0 = m0, mA = mA, m = m): object 'm' not found
```

```
## Many error messages!  But note what the first one, for model mA is: there are
## more random effects than there are observations in the data set!  As explained
## on Piazza, this means lmer() cannot fit a model.  Thus, the model
##
## as.numeric(Rating) ~ -1 + as.factor(Rater) + Semester +
##                          (1 | Artifact) + (Semseter | Artifact)
##
## isn't even possible, so no testing is needed.



## Testng (as.factor(Rater)|Artifact)

m0 <- tmp                                       ## Null hypothesis
mA <- update(m0, . ~ . + (as.factor(Rater)|Artifact))   ## Alternative hypotheses
```

```
## Error: number of observations (=116) <= number of random effects (=270) for term (as.factor(Rater) |
```

```
m  <- update(mA, . ~ . - (1|Artifact))          ## Model with only the new R.E.
```

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'object' in selecting a method
```

```
exactRLRT(m0=m0,mA=mA,m=m)
```

```
## Error in exactRLRT(m0 = m0, mA = mA, m = m): object 'm' not found
```

```
## Same thing happened!  Again, the model
##
## as.numeric(Rating) ~ -1 + as.factor(Rater) + Semester +
##                           (1 | Artifact) + (as.factor(Rater) | Artifact)
##
## isn't even possible, so no testing is needed.


## Thus, we weren't able to add or take away anything from the model "tmp",
## so this is our final model for SelMeth:

summary(tmp)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ as.factor(Rater) + Semester + (1 | Artifact) -
##     1
##    Data: tall.nonmissing[tall.nonmissing$Rubric == "SelMeth", ]
##
## REML criterion at convergence: 143.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0480 -0.3923 -0.0551  0.2674  2.5827
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Artifact (Intercept) 0.08973  0.2996
##  Residual             0.10842  0.3293
## Number of obs: 116, groups:  Artifact, 90
##
## Fixed effects:
##                   Estimate Std. Error t value
## as.factor(Rater)1  2.25037    0.07503  29.992
## as.factor(Rater)2  2.22653    0.07424  29.991
## as.factor(Rater)3  2.03316    0.07521  27.033
## SemesterS19       -0.35860    0.09796  -3.661
##
## Correlation of Fixed Effects:
##             a.(R)1 a.(R)2 a.(R)3
## as.fctr(R)2  0.285
## as.fctr(R)3  0.287  0.280
## SemesterS19 -0.413 -0.391 -0.394
```

```
## You would need to do something similar with the other models that are not
## just intercept-only models (i.e the models for CritDes and InterpRes)

## ALSO, please don't forget:  I am not giving interpretations to the model fits
## or coefficient estimates here. That is something I'm leaving for you, as you
## complete your analyses and write an IDMRAD paper for the college.
```

2(c)(iv): Trying to add fixed effects, interactions, and new random effects to the "combined" model Rating ~ 1 + (0 + Rubric|Artifact), using all the data.

Now we try something similar with the "combined" model suggested on p. 4 of the project assignment sheet.

```
## Start with the "combined" intercept-only model...

comb.0 <- lmer(as.numeric(Rating) ~ 1 + (0 + Rubric | Artifact),
               data=tall.nonmissing)
```

```
## boundary (singular) fit: see ?isSingular
```

```
summary(comb.0)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ 1 + (0 + Rubric | Artifact)
##    Data: tall.nonmissing
##
## REML criterion at convergence: 1471.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0218 -0.4940 -0.0753  0.5271  3.7759
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.64070  0.8004
##           RubricInitEDA   0.38288  0.6188   0.26
##           RubricInterpRes 0.25658  0.5065   0.00 0.79
##           RubricRsrchQ    0.17398  0.4171   0.38 0.50 0.74
##           RubricSelMeth   0.09619  0.3102   0.56 0.37 0.41 0.26
##           RubricTxtOrg    0.40425  0.6358   0.03 0.69 0.80 0.64 0.24
##           RubricVisOrg    0.31878  0.5646   0.17 0.78 0.76 0.60 0.29 0.79
##  Residual                 0.19477  0.4413
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  2.23210    0.04013   55.63
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
## R complains that we have a "boundary (singular) fit", i.e. the
## variance-covariance matrix for the random effects is singular
## (not of full rank), or nearly singular.
##
## What this typically means is that some of the random effects are highly
## correlated with one another.  We can see this in the "Random effects"
## block of summary(comb.0):
##
## * The random effects for VisOrg and TxtOrg seem highly correlated with
##   each other and with everything except for the rand. effect for SelMeth
##
## * The random effects for InterpRes and InitEDA are highly correlated
##
## * The random effects for RsrchQ and InterpRes are highly correlated
##
## etc.
##
## In some ways we should not be surprised: these rubrics all represent
```

```
## features of a good research report, and we would expect that if someone
## is good at one or two of these features, they are probably good at the
## others.

## Although the random effects are highly correlated, we can still proceed with
## our variable selection...

## Try adding fixed effects with no interactions...

comb.full <- update(comb.0, . ~ . + as.factor(Rater) + Semester +
                      Sex + Repeated + Rubric)

summary(comb.full)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Sex + Repeated + Rubric
##    Data: tall.nonmissing
##
## REML criterion at convergence: 1429.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1091 -0.5065 -0.0178  0.5242  3.7932
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.55311  0.7437
##           RubricInitEDA   0.35239  0.5936   0.47
##           RubricInterpRes 0.17512  0.4185   0.23 0.75
##           RubricRsrchQ    0.16997  0.4123   0.58 0.44 0.71
##           RubricSelMeth   0.06816  0.2611   0.39 0.60 0.74 0.41
##           RubricTxtOrg    0.26339  0.5132   0.34 0.62 0.70 0.56 0.67
##           RubricVisOrg    0.25809  0.5080   0.35 0.73 0.68 0.52 0.41 0.76
##  Residual                 0.18916  0.4349
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
##                    Estimate Std. Error t value
## (Intercept)        2.013748   0.109103  18.457
## as.factor(Rater)2  0.001977   0.054887   0.036
## as.factor(Rater)3 -0.174867   0.055045  -3.177
## SemesterS19       -0.175017   0.087850  -1.992
## SexM               0.010506   0.081271   0.129
## Repeated          -0.073586   0.098522  -0.747
## RubricInitEDA      0.547054   0.095710   5.716
## RubricInterpRes    0.587091   0.100893   5.819
## RubricRsrchQ       0.460875   0.087516   5.266
## RubricSelMeth      0.164863   0.094265   1.749
## RubricTxtOrg       0.692880   0.099523   6.962
## RubricVisOrg       0.530182   0.099136   5.348
##
## Correlation of Fixed Effects:
##             (Intr) a.(R)2 a.(R)3 SmsS19 SexM   Repetd RbIEDA RbrcIR RbrcRQ
```

```
## as.fctr(R)2 -0.245
## as.fctr(R)3 -0.237  0.499
## SemesterS19 -0.361  0.008  0.000
## SexM        -0.398 -0.026 -0.035  0.302
## Repeated    -0.154  0.001 -0.003  0.079  0.009
## RubrcIntEDA -0.552 -0.001  0.000 -0.001  0.000  0.007
## RbrcIntrpRs -0.660 -0.001  0.000 -0.001  0.000 -0.009  0.734
## RubrcRsrchQ -0.626 -0.001  0.000 -0.001  0.000 -0.039  0.585  0.756
## RubricSlMth -0.689 -0.001  0.000 -0.001  0.000 -0.088  0.659  0.777  0.689
## RubrcTxtOrg -0.611 -0.001  0.000 -0.001  0.000  0.005  0.674  0.751  0.682
## RubricVsOrg -0.607 -0.001 -0.001 -0.002 -0.001 -0.021  0.715  0.745  0.668
##             RbrcSM RbrcTO
## as.fctr(R)2
## as.fctr(R)3
## SemesterS19
## SexM
## Repeated
## RubrcIntEDA
## RbrcIntrpRs
## RubrcRsrchQ
## RubricSlMth
## RubrcTxtOrg  0.725
## RubricVsOrg  0.680  0.750
```

```
##
## It's interesting to note that comb.full is no longer a boundary (singular)
## fit.  Adding the fixed effects changed the residuals enough that the
## variance-covariance matrix for the random effects is no longer (nearly)
## singular.

comb.back_elim <- fitLMER.fnc(comb.full, log.file.name = FALSE)
```

```
## Warning in fitLMER.fnc(comb.full, log.file.name = FALSE): Argument "ran.effects" is empty, which mea
## TRUE
```

```
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.887 >= 0.05
##     not part of higher-order interaction
```

```
## boundary (singular) fit: see ?isSingular
```

```
##     removing term
##   iteration 2
##     p-value for term "Repeated" = 0.0919 >= 0.05
##     not part of higher-order interaction
```

```
## boundary (singular) fit: see ?isSingular
```

```
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===            forwardfitting random effects       ===
```

```
## ========================================================
##  ===       random slopes       ===
## ========================================================
## ===             re-backfitting fixed effects        ===
## ========================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE

## boundary (singular) fit: see ?isSingular

## pruning random effects structure ...
##   nothing to prune
```

```
summary(comb.back_elim)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Rubric
##    Data: tall.nonmissing
##
## REML criterion at convergence: 1424.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1200 -0.5125 -0.0173  0.5302  3.7752
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.55495  0.7449
##           RubricInitEDA   0.35064  0.5921   0.47
##           RubricInterpRes 0.16892  0.4110   0.23 0.75
##           RubricRsrchQ    0.16777  0.4096   0.59 0.44 0.70
##           RubricSelMeth   0.06499  0.2549   0.40 0.60 0.74 0.40
##           RubricTxtOrg    0.25615  0.5061   0.33 0.61 0.69 0.55 0.66
##           RubricVisOrg    0.25894  0.5089   0.35 0.73 0.68 0.52 0.41 0.75
##  Residual                 0.18934  0.4351
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
##                      Estimate Std. Error t value
## (Intercept)         2.0084130  0.0987610  20.336
## as.factor(Rater)2   0.0003231  0.0547446   0.006
## as.factor(Rater)3  -0.1771062  0.0548892  -3.227
## SemesterS19        -0.1730357  0.0826927  -2.093
## RubricInitEDA       0.5474747  0.0957148   5.720
## RubricInterpRes     0.5864544  0.1008618   5.814
## RubricRsrchQ        0.4584082  0.0874179   5.244
## RubricSelMeth       0.1590770  0.0937771   1.696
## RubricTxtOrg        0.6930033  0.0995479   6.962
## RubricVisOrg        0.5289027  0.0990973   5.337
##
## Correlation of Fixed Effects:
##             (Intr) a.(R)2 a.(R)3 SmsS19 RbIEDA RbrcIR RbrcRQ RbrcSM RbrcTO
## as.fctr(R)2 -0.281
## as.fctr(R)3 -0.277  0.499
```

```
## SemesterS19 -0.264  0.017  0.011
## RubrcIntEDA -0.610 -0.001  0.000 -0.002
## RbrcIntrpRs -0.735 -0.001  0.000  0.000  0.734
## RubrcRsrchQ -0.701 -0.001  0.000  0.002  0.586  0.756
## RubricSlMth -0.782  0.000  0.000  0.006  0.662  0.779  0.688
## RubrcTxtOrg -0.679 -0.001  0.000 -0.001  0.674  0.751  0.682  0.728
## RubricVsOrg -0.675 -0.001 -0.001  0.000  0.715  0.745  0.667  0.681  0.750
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
## The final model fit is a boundary fit again, but we will proceed to try
## interactions

comb.inter <- update(comb.back_elim, . ~ . + as.factor(Rater)*Semester*Rubric)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00371227 (tol = 0.002, component 1)
```

```
## This didn't quite converge, so we will try switching optimizers and increasing
## the number of iterations allowed...

ss <- getME(comb.inter,c("theta","fixef"))
comb.inter.u<- update(comb.inter,start=ss,
            control=lmerControl(optimizer="bobyqa",
                                optCtrl=list(maxfun=2e5)))
```

```
## boundary (singular) fit: see ?isSingular
```

```
## it takes a few seconds to fit, but at least we got a converged fit.
## again, boundary fit (near-singular random effects variance-covariance mtx)

summary(comb.inter.u)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Rubric + as.factor(Rater):Semester + as.factor(Rater):Rubric +
##     Semester:Rubric + as.factor(Rater):Semester:Rubric
##    Data: tall.nonmissing
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
## REML criterion at convergence: 1424.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9141 -0.5141 -0.0653  0.5023  3.6609
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.48550  0.6968
##           RubricInitEDA   0.35257  0.5938   0.42
##           RubricInterpRes 0.14619  0.3824   0.32 0.80
##           RubricRsrchQ    0.16444  0.4055   0.66 0.43 0.72
##           RubricSelMeth   0.06297  0.2509   0.45 0.64 0.78 0.49
##           RubricTxtOrg    0.25441  0.5044   0.44 0.65 0.67 0.60 0.62
##           RubricVisOrg    0.25527  0.5052   0.35 0.73 0.68 0.57 0.35 0.76
##  Residual                 0.18839  0.4340
```

```
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
##                                              Estimate Std. Error t value
## (Intercept)                                  1.739538   0.136568  12.738
## as.factor(Rater)2                            0.302995   0.155107   1.953
## as.factor(Rater)3                            0.237851   0.155863   1.526
## SemesterS19                                 -0.129077   0.250318  -0.516
## RubricInitEDA                                0.765215   0.165241   4.631
## RubricInterpRes                              0.979228   0.162160   6.039
## RubricRsrchQ                                 0.710427   0.147386   4.820
## RubricSelMeth                                0.462750   0.155274   2.980
## RubricTxtOrg                                 1.011251   0.160899   6.285
## RubricVisOrg                                 0.647869   0.166603   3.889
## as.factor(Rater)2:SemesterS19                0.268014   0.303883   0.882
## as.factor(Rater)3:SemesterS19               -0.072789   0.301026  -0.242
## as.factor(Rater)2:RubricInitEDA             -0.325018   0.204108  -1.592
## as.factor(Rater)3:RubricInitEDA             -0.374190   0.205354  -1.822
## as.factor(Rater)2:RubricInterpRes           -0.469281   0.201051  -2.334
## as.factor(Rater)3:RubricInterpRes           -0.711515   0.202316  -3.517
## as.factor(Rater)2:RubricRsrchQ              -0.447050   0.189326  -2.361
## as.factor(Rater)3:RubricRsrchQ              -0.474411   0.190681  -2.488
## as.factor(Rater)2:RubricSelMeth             -0.301450   0.193678  -1.556
## as.factor(Rater)3:RubricSelMeth             -0.365656   0.194970  -1.875
## as.factor(Rater)2:RubricTxtOrg              -0.449164   0.200927  -2.235
## as.factor(Rater)3:RubricTxtOrg              -0.407754   0.202209  -2.016
## as.factor(Rater)2:RubricVisOrg               0.009042   0.205059   0.044
## as.factor(Rater)3:RubricVisOrg              -0.287443   0.206299  -1.393
## SemesterS19:RubricInitEDA                   -0.050212   0.301475  -0.167
## SemesterS19:RubricInterpRes                  0.127813   0.295706   0.432
## SemesterS19:RubricRsrchQ                     0.133874   0.267750   0.500
## SemesterS19:RubricSelMeth                   -0.089616   0.282837  -0.317
## SemesterS19:RubricTxtOrg                     0.166097   0.293176   0.567
## SemesterS19:RubricVisOrg                     0.146845   0.302496   0.485
## as.factor(Rater)2:SemesterS19:RubricInitEDA  0.020326   0.392376   0.052
## as.factor(Rater)3:SemesterS19:RubricInitEDA  0.252422   0.389961   0.647
## as.factor(Rater)2:SemesterS19:RubricInterpRes -0.266618  0.385390  -0.692
## as.factor(Rater)3:SemesterS19:RubricInterpRes -0.152392  0.383354  -0.398
## as.factor(Rater)2:SemesterS19:RubricRsrchQ   -0.217348   0.360414  -0.603
## as.factor(Rater)3:SemesterS19:RubricRsrchQ    0.354319   0.357388   0.991
## as.factor(Rater)2:SemesterS19:RubricSelMeth  -0.401036   0.370200  -1.083
## as.factor(Rater)3:SemesterS19:RubricSelMeth  -0.192670   0.367887  -0.524
## as.factor(Rater)2:SemesterS19:RubricTxtOrg   -0.542267   0.385011  -1.408
## as.factor(Rater)3:SemesterS19:RubricTxtOrg   -0.316395   0.382614  -0.827
## as.factor(Rater)2:SemesterS19:RubricVisOrg   -0.603626   0.392909  -1.536
## as.factor(Rater)3:SemesterS19:RubricVisOrg   -0.186749   0.390759  -0.478
##
## Correlation matrix not shown by default, as p = 42 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
## optimizer (bobyqa) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```r
## If you compare with summary(comb.inter) you will see that
## there wasn't much difference in the fitted values; we could
## probably have just proceeded wth the model comb.inter.  But
## since we have the converged model we will use it for fixed
## effects selection

comb.inter_elim <- fitLMER.fnc(comb.inter.u, log.file.name = FALSE)
```

```
## Warning in fitLMER.fnc(comb.inter.u, log.file.name = FALSE): Argument "ran.effects" is empty, which r
## TRUE
```

```
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 3
##   iteration 1
##     p-value for term "as.factor(Rater):Semester:Rubric" = 0.5526 >= 0.05
##     not part of higher-order interaction
```

```
## boundary (singular) fit: see ?isSingular
```

```
##     removing term
## processing model terms of interaction level 2
##   iteration 2
##     p-value for term "as.factor(Rater):Semester" = 0.598 >= 0.05
##     not part of higher-order interaction
```

```
## boundary (singular) fit: see ?isSingular
```

```
##     removing term
##   iteration 3
##     p-value for term "Semester:Rubric" = 0.0761 >= 0.05
##     not part of higher-order interaction
```

```
## boundary (singular) fit: see ?isSingular
```

```
##     removing term
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## pruning random effects structure ...
##   nothing to prune
## ========================================================
## ===            forwardfitting random effects       ===
## ========================================================
##  ===       random slopes       ===
## ========================================================
## ===            re-backfitting fixed effects        ===
## ========================================================
## processing model terms of interaction level 2
##   all terms of interaction level 2 significant
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
```

```
## boundary (singular) fit: see ?isSingular
```

```
## pruning random effects structure ...
##   nothing to prune
```

```
summary(comb.inter_elim)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Rubric + as.factor(Rater):Rubric
##    Data: tall.nonmissing
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
## REML criterion at convergence: 1419.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9280 -0.5122 -0.0447  0.4827  3.5854
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.50348  0.7096
##           RubricInitEDA   0.35480  0.5956   0.44
##           RubricInterpRes 0.15192  0.3898   0.35 0.82
##           RubricRsrchQ    0.17953  0.4237   0.63 0.44 0.72
##           RubricSelMeth   0.06727  0.2594   0.42 0.60 0.74 0.36
##           RubricTxtOrg    0.26069  0.5106   0.42 0.64 0.67 0.55 0.64
##           RubricVisOrg    0.25491  0.5049   0.34 0.71 0.68 0.51 0.38 0.77
##  Residual                 0.18519  0.4303
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
##                                  Estimate Std. Error t value
## (Intercept)                       1.75945    0.11785  14.929
## as.factor(Rater)2                 0.36537    0.13296   2.748
## as.factor(Rater)3                 0.21421    0.13297   1.611
## SemesterS19                      -0.17780    0.08228  -2.161
## RubricInitEDA                     0.74625    0.13676   5.457
## RubricInterpRes                   1.01453    0.13479   7.527
## RubricRsrchQ                      0.74926    0.12419   6.033
## RubricSelMeth                     0.42672    0.13040   3.272
## RubricTxtOrg                      1.04967    0.13551   7.746
## RubricVisOrg                      0.68354    0.13947   4.901
## as.factor(Rater)2:RubricInitEDA  -0.30843    0.17249  -1.788
## as.factor(Rater)3:RubricInitEDA  -0.29522    0.17282  -1.708
## as.factor(Rater)2:RubricInterpRes -0.53674    0.17008  -3.156
## as.factor(Rater)3:RubricInterpRes -0.75247    0.17049  -4.414
## as.factor(Rater)2:RubricRsrchQ   -0.50157    0.16151  -3.106
## as.factor(Rater)3:RubricRsrchQ   -0.37068    0.16179  -2.291
## as.factor(Rater)2:RubricSelMeth  -0.39602    0.16467  -2.405
## as.factor(Rater)3:RubricSelMeth  -0.41324    0.16504  -2.504
## as.factor(Rater)2:RubricTxtOrg   -0.58380    0.17141  -3.406
## as.factor(Rater)3:RubricTxtOrg   -0.48649    0.17177  -2.832
## as.factor(Rater)2:RubricVisOrg   -0.14444    0.17442  -0.828
## as.factor(Rater)3:RubricVisOrg   -0.33380    0.17481  -1.910
##
## Correlation matrix not shown by default, as p = 22 > 12.
## Use print(x, correlation=TRUE)  or
```

```
##     vcov(x)        if you need it
```

```
## optimizer (bobyqa) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
## it's a little hard to compare summaries for such big models, so let's look
## at the highlights:

formula(comb.inter.u)
```

```
## as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Rubric + as.factor(Rater):Semester + as.factor(Rater):Rubric +
##     Semester:Rubric + as.factor(Rater):Semester:Rubric
```

```
formula(comb.inter_elim)
```

```
## as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Rubric + as.factor(Rater):Rubric
```

```
formula(comb.back_elim)
```

```
## as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Rubric
```

```
summary(comb.inter.u)$varcor
```

```
##  Groups   Name            Std.Dev. Corr
##  Artifact RubricCritDes   0.69678
##           RubricInitEDA   0.59378  0.416
##           RubricInterpRes 0.38235  0.324 0.800
##           RubricRsrchQ    0.40551  0.655 0.430 0.723
##           RubricSelMeth   0.25094  0.446 0.639 0.784 0.488
##           RubricTxtOrg    0.50439  0.436 0.649 0.667 0.604 0.622
##           RubricVisOrg    0.50524  0.349 0.727 0.675 0.567 0.346 0.757
##  Residual                 0.43404
```

```
summary(comb.inter_elim)$varcor
```

```
##  Groups   Name            Std.Dev. Corr
##  Artifact RubricCritDes   0.70956
##           RubricInitEDA   0.59565  0.445
##           RubricInterpRes 0.38977  0.354 0.815
##           RubricRsrchQ    0.42371  0.631 0.440 0.716
##           RubricSelMeth   0.25937  0.424 0.601 0.737 0.364
##           RubricTxtOrg    0.51058  0.417 0.637 0.675 0.547 0.636
##           RubricVisOrg    0.50489  0.339 0.715 0.677 0.512 0.376 0.772
##  Residual                 0.43034
```

```
summary(comb.back_elim)$varcor
```

```
##  Groups   Name            Std.Dev. Corr
##  Artifact RubricCritDes   0.74495
##           RubricInitEDA   0.59215  0.467
##           RubricInterpRes 0.41100  0.230 0.749
##           RubricRsrchQ    0.40960  0.588 0.436 0.704
##           RubricSelMeth   0.25493  0.399 0.603 0.736 0.397
##           RubricTxtOrg    0.50612  0.335 0.614 0.691 0.551 0.656
##           RubricVisOrg    0.50886  0.350 0.731 0.679 0.516 0.414 0.752
##  Residual                 0.43513
```

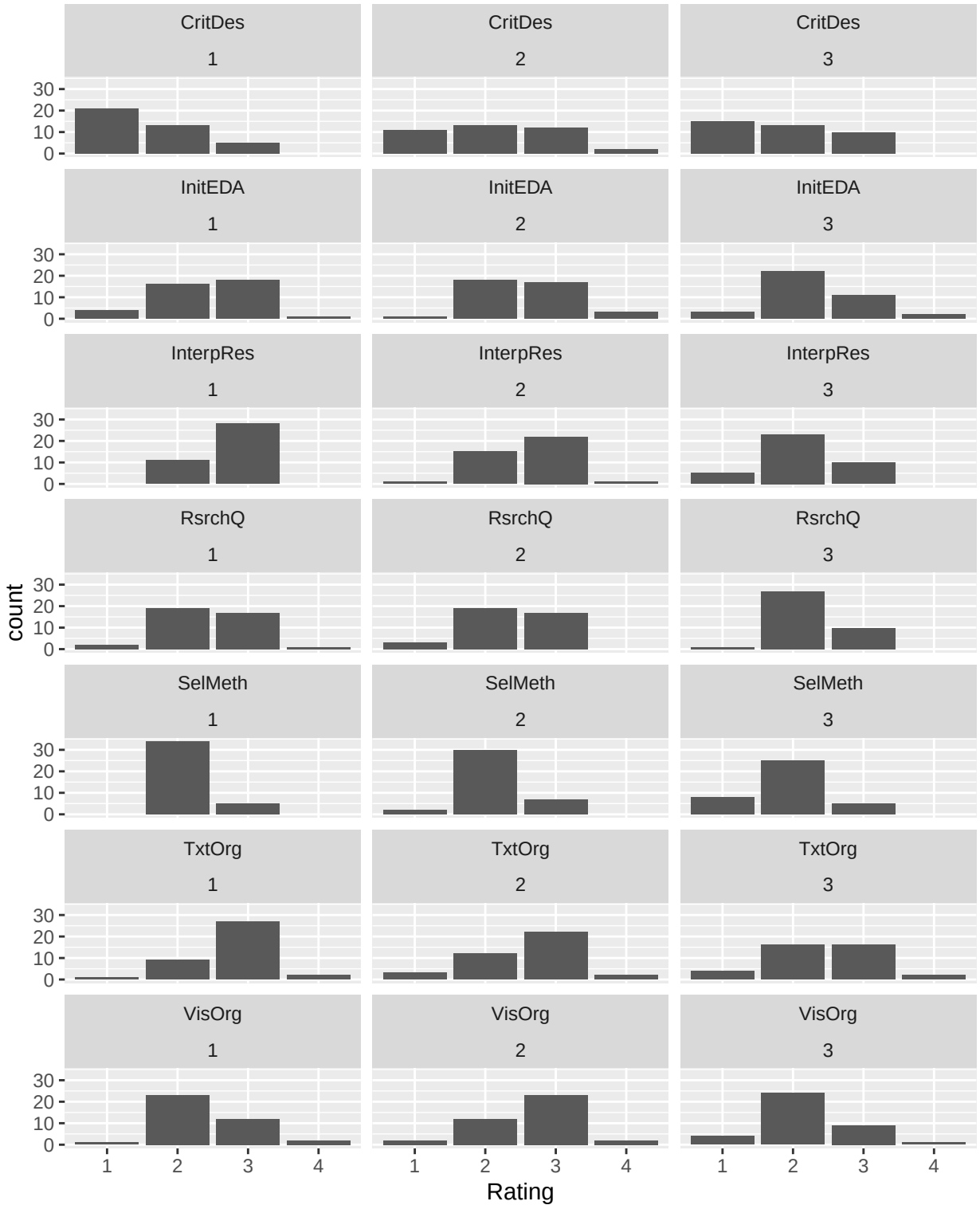
```
anova(comb.back_elim,comb.inter_elim,comb.inter.u)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: tall.nonmissing
## Models:
## comb.back_elim: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Rubric
## comb.inter_elim: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Rubric
## comb.inter.u: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Rubric + a
##                 npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## comb.back_elim    39 1464.0 1647.2 -693.02   1386.0
## comb.inter_elim   51 1454.5 1694.1 -676.26   1352.5 33.526 12   0.000801 ***
## comb.inter.u      71 1471.4 1804.8 -664.68   1329.4 23.161 20   0.280962
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## the models are nested so we can use AIC, BIC or likelihod ratio (deviance)
## tests... AIC and the LRT agree on comb.inter_elim; BIC likes the simpler
## comb.back_elim.

## Interestingly, comb.inter_elim adds a rater x rubric interaction to
## the main-effects model comb.back_elim.  This suggests that the raters
## do not all use the rubrics in the same way.
```

```
## In addition to looking at the fixed effect coefficients in
## summary(comb.inter_elim)$coef, we could also see if there's
## a pattern in an appropriate facets plot

g <- ggplot(tall.nonmissing, aes(x=Rating)) +
  geom_bar() +
  facet_wrap( ~ Rubric + Rater, nrow=7)

g
```




```
## and it does look as if the 3 raters have different ways of scoring the 7 rubrics,
## so the interaction we found in comb.inter_elim makes sense.  (Clearly it is
## not the case that one rater is simply more harsh than another, or something
## like that.  Can you describe the different patterns of scoring among the 3
## raters?  Can you relate it to coefficient estimates in summary()$coef?)
```

```
## Finally, we consider adding random effects to what seems like the
## best model so far, comb.inter_elim...

## The fixed-effects terms we have to work with are:
##
## as.factor(Rater)
## Semester
## as.factor(Rater):Rubric
##
## We want to add each of these *without* a random intercept, to preserve the
## structure of the model (separate random interepts for each rubric)
##
## In all cases, there is more than one random effect to test (3 for raters,
## 2 for semesters, 7 for rubrics, and 21 for the interaction).  Since exactRLRT()
## can only test single random effects, we can't use it.  Instead we inspect AIC
## andBIC from anova() tables for these...

## Fitting some of these models produces various errors and warnings; I am not
## going to worry about them too much, in order to get an idea of what random
## effects I may want...

m0 <- comb.inter_elim
mA <- lmer(as.numeric(Rating) ~ (0 + Rubric | Artifact) +
             (0 + as.factor(Rater) | Artifact) + as.factor(Rater) +
             Semester + Rubric + as.factor(Rater):Rubric, data=tall.nonmissing)
```

```
## boundary (singular) fit: see ?isSingular
```

```
anova(m0,mA)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Warning in commonArgs(par, fn, control, environment()): maxfun < 10 *
## length(par)^2 is not recommended.
```

```
## Data: tall.nonmissing
## Models:
## m0: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Rubric + as.factor(
## mA: as.numeric(Rating) ~ (0 + Rubric | Artifact) + (0 + as.factor(Rater) | Artifact) + as.factor(Rat
##    npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m0   51 1454.5 1694.1 -676.26   1352.5
## mA   57 1415.9 1683.6 -650.94   1301.9 50.647  6  3.487e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## AIC and BIC both like including (0 + as.factor(Rater) | Artifact) in the model

m0 <- comb.inter_elim
mA <- lmer(as.numeric(Rating) ~ (0 + Rubric | Artifact) +
             (0 + Semester | Artifact) + as.factor(Rater) +
             Semester + Rubric + as.factor(Rater):Rubric, data=tall.nonmissing)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
```

```
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues
```

```
anova(m0,mA)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: tall.nonmissing
## Models:
## m0: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Rubric + as.factor(
## mA: as.numeric(Rating) ~ (0 + Rubric | Artifact) + (0 + Semester | Artifact) + as.factor(Rater) + Se
##    npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## m0   51 1454.5 1694.1 -676.26   1352.5
## mA   54 1458.4 1712.0 -675.18   1350.4 2.1534  3     0.5412
```

```
##
## AIC and BIC do not like (0 + Semester | Artifact) in the model...


m0 <- comb.inter_elim
mA <- lmer(as.numeric(Rating) ~ (0 + Rubric | Artifact) +
             (0 + as.factor(Rater) | Artifact) +
             (0 + as.factor(Rater):Rubric | Artifact) + as.factor(Rater) +
             Semester + Rubric + as.factor(Rater):Rubric, data=tall.nonmissing)
```

```
## Error: number of observations (=810) <= number of random effects (=1890) for term (0 + as.factor(Rat
```

```
## anova(m0,mA)     -- Not needed!
##
## There are not enough observations to fit mA here, so we need not do any
## formal model comparison...

## So, to summarize, the "final" model appears to be

comb.final <- lmer(as.numeric(Rating) ~ (0 + Rubric | Artifact) +
             (0 + as.factor(Rater) | Artifact) + as.factor(Rater) +
             Semester + Rubric + as.factor(Rater):Rubric, data=tall.nonmissing)
```

```
## boundary (singular) fit: see ?isSingular
```

```
formula(comb.final)
```

```
## as.numeric(Rating) ~ (0 + Rubric | Artifact) + (0 + as.factor(Rater) |
##     Artifact) + as.factor(Rater) + Semester + Rubric + as.factor(Rater):Rubric
```

```
summary(comb.final)$varcor
```

```
##  Groups     Name               Std.Dev. Corr
##  Artifact   RubricCritDes      0.70461
##             RubricInitEDA      0.56379   0.318
##             RubricInterpRes    0.31946   0.142  0.674
##             RubricRsrchQ       0.42310   0.500  0.194  0.538
##             RubricSelMeth      0.19557   0.145  0.226  0.376 -0.240
##             RubricTxtOrg       0.50026   0.268  0.437  0.364  0.305  0.213
##             RubricVisOrg       0.48201   0.175  0.504  0.445  0.276 -0.161
##  Artifact.1 as.factor(Rater)1 0.11320
##             as.factor(Rater)2 0.33427  -0.486
##             as.factor(Rater)3 0.30681   0.332  0.663
##  Residual                     0.36699
```

```
##
##
##
##
##
##
##
##   0.537
##
##
##
##
```

```
summary(comb.final)$coef
```

```
##                                    Estimate Std. Error    t value
## (Intercept)                       1.7575545 0.11404151 15.4115336
## as.factor(Rater)2                 0.3660542 0.13918252  2.6300297
## as.factor(Rater)3                 0.1959088 0.12966636  1.5108686
## SemesterS19                      -0.1591805 0.07647529 -2.0814634
## RubricInitEDA                     0.7394940 0.12996076  5.6901329
## RubricInterpRes                   0.9915148 0.12770767  7.7639406
## RubricRsrchQ                      0.7261869 0.11793023  6.1577676
## RubricSelMeth                     0.4106797 0.12470498  3.2932102
## RubricTxtOrg                      1.0157815 0.12999540  7.8139797
## RubricVisOrg                      0.6542506 0.13353098  4.8996162
## as.factor(Rater)2:RubricInitEDA  -0.2998076 0.15609075 -1.9207264
## as.factor(Rater)3:RubricInitEDA  -0.2947319 0.15635201 -1.8850532
## as.factor(Rater)2:RubricInterpRes -0.5132297 0.15348482 -3.3438467
## as.factor(Rater)3:RubricInterpRes -0.7148433 0.15363960 -4.6527283
## as.factor(Rater)2:RubricRsrchQ   -0.4874137 0.14722146 -3.3107521
## as.factor(Rater)3:RubricRsrchQ   -0.3223799 0.14726517 -2.1891116
## as.factor(Rater)2:RubricSelMeth  -0.3863739 0.15030941 -2.5705236
## as.factor(Rater)3:RubricSelMeth  -0.3871581 0.14961457 -2.5877033
## as.factor(Rater)2:RubricTxtOrg   -0.5510439 0.15646043 -3.5219379
## as.factor(Rater)3:RubricTxtOrg   -0.4448937 0.15673122 -2.8385772
## as.factor(Rater)2:RubricVisOrg   -0.1048994 0.15861081 -0.6613632
## as.factor(Rater)3:RubricVisOrg   -0.2752130 0.15884865 -1.7325485
```

```
## if we accept comb.final as our final model, we can interpret the pieces as
## follows:
##
## (0 + as.factor(Rater) | Artifact) + as.factor(Rater)
##   * There is a kind of Rater x Artifact interaction: each Rater's
##     rating on each Artifact differs from what we would expect (from the
##     fixed effects alone) by a small random effect that depends on the Artifact
##
## Rubric + as.factor(Rater) + as.factor(Rater):Rubric
##   * There is a Rater x Rubric interaction: each Rater uses each
##     Rubric in a way that is not like, or even parallel to, other rater's
##     Rubric usage.  (we saw that in the facets plot above also).
##
## (0 + Rubric | Artifact) + Rubric
##   * There is a kind of Rubric x Artifact interaction: There are
##     different average scores on each rubric, but the rubric averages also
```

```
##     vary a bit from one Artifact to the next, by a small random effect that
##     depends on Artifact

## In all of this, the fact that Rubric scores depend on Artifact (that is,
## there is a kind of Rubric x Artifact interaction) is what we might expect:
## the artifacts aren't all of equal quality on each rubric, and so we should
## expect the average scores on each Rubric to vary from one Artifact to the next.
##
## More troubling are the Rater x Rubric interaction and the "kind of"
## Rater x Artifact interaction.  The Rater x Rubric interaction suggests
## that the Raters are not all interpreting the Rubrics in the same way.  The
## "kind of" Rater x Artifact interaction suggests that the Raters are not
## interpreting the evidence in the artifacts in the same way.  These
## interactions suggest that perhaps the raters should be trained more, to
## make the raters' ratings more similar to each other.
```

2(d) Is there anything else interesting to say about this data?

I leave this entirely up to you. By now you are pretty familiar with the data (probably sick of it actually!), so you may already have some ideas here.

Here are some things you could try (you don't need to try all of them, and there may be other, better things to try instead. Just try to come up with something that is interesting and useful to the college):

- Do some additional EDA. We really haven't tried to do a very complete EDA on this data yet.
- Are there better ways to illustrate or interpret the models we have fitted?
- Are there ways (through EDA, plots of model fits, etc.) to understand the differences in the models fitted to the data from the 13 common items, vs fitting to all the data?
- Is there any way to justify, based on the data and on where the data come from, imputing the missing Sex value as "M" or "F"?
- For some of the models, residual diagnostic plots might be interesting.
- What other models could you fit? More multi-level models? Some ordinary regression models? glm's? etc. . .
- etc.

I look forward to anything interesting that you can come up with!