Analyzing Ratings of Freshmen Statistics Projects

Zach Ohl
Department of Statistics and Data Science, Carnegie Mellon University
zohl@andrew.cmu.edu

ABSTRACT

The goal of this study is to find out what factors relate to freshman statistics project ratings in Dietrich College's General Education program. The data consist of 91 projects and a total of 817 ratings on a variety of rubrics from a group of 3 raters. To address this issue, I used a mix of exploratory data analysis and multilevel regression models including fixed effects and random effects clustered by individual student projects. Through EDA and modeling, factors other than which student wrote the paper were found to relate to the ratings a paper received, including which rubric was being graded, which rater graded it, the semester the student took the course, and combinations of these were discovered. Dietrich college will be interested in ideas for eliminating these factors so that ratings are as fair and consistent as possible.

INTRODUCTION

Dietrich College at Carnegie Mellon University is implementing a new General Education undergraduate program. One way they're evaluating the new program is by rating the work done by its students. Data on student work from a freshman statistics course in the program has recently become available. The data was obtained from an experiment where 91 projects were given a score of 1-4 on 7 different rubrics. Three different raters rated the projects using rubrics and rating scales which were only used for this experiment. The dataset includes scores for each project in each of the 7 categories, which rater(s) rated the paper, and information on the student who wrote it, including their sex and in which semester they took the class.

I've been asked to use this data to answer the following questions:

1. *Is the distribution of ratings for each rubric 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?*

2. *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?*

3. *More generally, how are the various factors in this experiment (Rater, Semester, Sex, Repeated, Rubric) related to the ratings? Do the factors interact in any interesting ways?*

4. *Is there anything else interesting to say about this data?*

Each section of the paper after Data is broken down according to these four questions. The answers to these questions will be useful in determining whether students are being graded fairly and consistently, and whether factors outside of a student's own effort, ability, and grasp of the material are influencing

their grades. I'm not aware of any previous studies on student project ratings at Carnegie Mellon University.

DATA

The data for this study were obtained from Junker (2021) and originate from Dietrich College at Carnegie Mellon University. It contains the ratings (scores) of 91 statistics projects submitted in either the fall or spring semester of a freshmen "General Education" course. The 91 projects were randomly selected from pool of projects with unknown size. Each project was assigned a unique artifact and rated on 7 different rubrics, where each rubric's rating is an integer from 1 to 4. Three different raters from different departments rated all the projects.

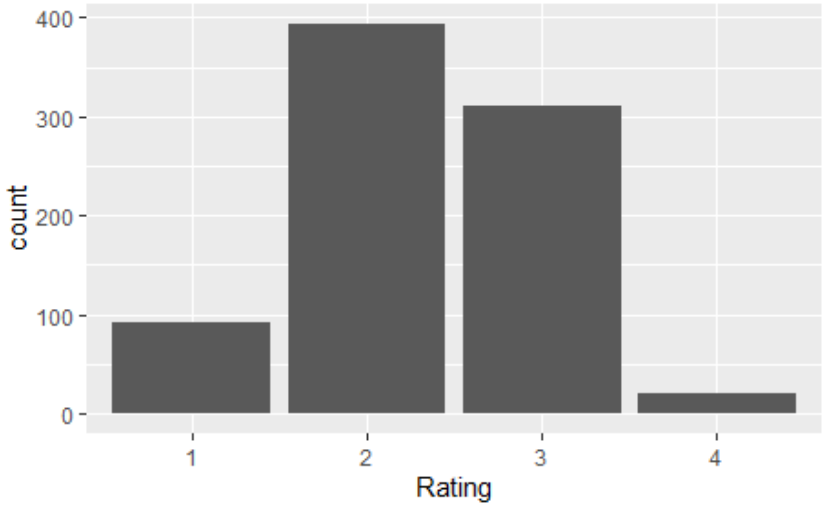
The identities of the undergraduate students that submitted the projects were unknown to the raters. Thirteen of the projects were graded by all 3 raters, while the other 78 were only graded by 1 rater. The rubric scoring guide used by the raters is experimental and not typical of freshman statistics classes at CMU. Guides for assigning ratings are shown in the table below.

Table 1: Rating Guides

Rating	Meaning
1	Student does not generate any relevant evidence.
2	Student generates evidence with significant flaws.
3	Student generates competent evidence; no flaws, or only minor ones.
4	Student generates outstanding evidence; comprehensive and sophisticated.

Because there were 7 rubric categories, 78 projects graded once, and 13 projects graded 3 times, the total number of ratings should have been $7 \times (78 + 13 \times 3) = 819$. However, two ratings were missing, so the total number of ratings was 817. The figure below shows the overall distribution of all available ratings.

Figure 1: Distribution of Ratings



A list of the variables contained in the dataset is shown below.

Table 2: Variable definitions

	Variable	Values	Description
1	X	1, 2, 3, . . .	Row number in the data set
2	Rater	1, 2 or 3	Which of the three raters gave a rating
3	Sample	1, 2, 3, . . .	Sample number
4	Overlap	1, 2, . . . , 13	Unique identifier for artifact seen by all 3 raters
5	Semester	Fall or Spring	Which semester the artifact came from
6	Sex	M or F	Sex or gender of student who created the artifact
7	RsrchQ	1, 2, 3 or 4	Rating on Research Question
8	CritDes	1, 2, 3 or 4	Rating on Critique Design
9	InitEDA	1, 2, 3 or 4	Rating on Initial EDA
10	SelMeth	1, 2, 3 or 4	Rating on Select Method(s)
11	InterpRes	1, 2, 3 or 4	Rating on Interpret Results
12	VisOrg	1, 2, 3 or 4	Rating on Visual Organization
13	TxtOrg	1, 2, 3 or 4	Rating on Text Organization
14	Artifact	(text labels)	Unique identifier for each artifact
15	Repeated	0 or 1	1 = this is one of the 13 artifacts seen by all 3 raters

The following table describes the meaning of each rubric.

Table 3: Rubric descriptions

	Abbreviation	Rubric name	Description
1	RsrchQ	Research Question	Given a scenario, the student generates, critiques or evaluates a relevant empirical research question.
2	CritDes	Critique Design	Given an empirical research question, the student critiques or evaluates to what extent a study design convincingly answer that question.
3	InitEDA	Initial EDA	Given a data set, the student appropriately describes the data and provides initial Exploratory Data Analysis.
4	SelMeth	Select Method(s)	Given a data set and a research question, the student selects appropriate method(s) to analyze the data.
5	InterpRes	Interpret Results	The student appropriately interprets the results of the selected method(s).
6	VisOrg	Visual Organization	The student communicates in an organized, coherent and effective fashion with visual elements (charts, graphs, tables, etc.).
7	TxtOrg	Text Organization	The student communicates in an organized, coherent and effective fashion with text elements (words, sentences, paragraphs, section and subsection titles, etc.).

The tables below summarize the numeric variables (ratings by rubric and by rater) and categorical variables (Rater, Semester, Sex, and Repeated).

Table 4: Summaries of Ratings by Rubric

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
RsrchQ	1	2	2	2.35	3.0	4	0.59
InitEDA	1	2	2	2.44	3.0	4	0.70
SelMeth	1	2	2	2.07	2.0	3	0.49
InterpRes	1	2	3	2.49	3.0	4	0.61
TxtOrg	1	2	3	2.60	3.0	4	0.70
CritDes	1	1	2	1.86	2.5	4	0.84
VisOrg	1	2	2	2.42	3.0	4	0.68

Table 5: Summaries of Ratings by Rater

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
Rater1	1	2	2	2.35	3	4	0.70
Rater2	1	2	2	2.43	3	4	0.70
Rater3	1	2	2	2.18	3	4	0.69

Table 6: Summaries of Ratings by Semester

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
Fall	1	2	2	2.36	3	4	0.67
Spring	1	2	2	2.23	3	4	0.78

Table 7: Summaries of Ratings by Sex

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
Male	1	2	2	2.31	3	4	0.71
Female	1	2	2	2.31	3	4	0.70

Table 8: Summaries of Ratings by Repeated or Not

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
Repeated	1	2	2	2.27	3	4	0.66
NotRepeated	1	2	2	2.34	3	4	0.72

Tables 9-12: Summaries of Categorical Variables

Rater	**Count**
1	39
2	39
3	39

Semester	**Count**
Fall	83
Spring	34

Sex	**Count**
F	64
M	53
Other/NA	1

Repeated	**Count**
No	78
Yes	39

METHODS

1. *Is the distribution of ratings for each rubric 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?*

To answer the first research question, I mainly used exploratory data analysis. I looked at numerical summaries and plots of the distributions of ratings grouped by rubric and grouped by rater. I also subsetted the data to only the 13 papers graded by all 3 graders and repeated the analyses on the reduced data set, to see if it was representative of the dataset as a whole.

2. *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?*

To explore this question, I first looked at the reduced dataset of 13 papers rated by each grader. I fit a random intercept model to predict rating for each of the 7 rubrics where the intercept randomly varied based on the artifact (unique paper) for that observation. So, each model had coefficients defining 13 different intercepts, where each intercept represented a cluster of 3 ratings.

For each of these 7 models, I used the intraclass correlation (ICC) as a metric for agreement between raters in a given model. The ICC for any of these 7 models represents the correlation between any 2 of the 3 different raters' ratings on the same artifact. For a random intercept model, ICC is calculated by dividing the variance of mean ratings for artifact groups (called τ^2) by the sum of τ^2 and the variance of individual ratings given an artifact mean (called σ^2).

High ICC for a given rubric means the raters tend to agree on the rating of that rubric. I also found the ICCs for the same models fitted on the whole dataset and compared.

Then, using the reduced data set again, I made two-way tables for each pair of raters and for each rubric (21 tables total) that counted up the 13 ratings each rater assigned for that rubric. This allowed us to count the proportion of times out of 13 that two raters gave the same rating for a certain rubric. Using these seven measures of agreement for each pair of ratings, I could get an idea of how often any two raters were in agreement.

3. *More generally, how are the various factors in this experiment (Rater, Semester, Sex, Repeated, Rubric) related to the ratings? Do the factors interact in any interesting ways?*

Adding fixed effects to the 7 reduced dataset rubric-specific models

I tried adding fixed effects (Rater, Semester, and Sex) to the random intercept models based on the reduced dataset of only 13 commonly rated artifacts. Note that the Repeated variable was not added because observations in this dataset were all Repeated. Then I did variable selection on these models.

If the variable selection process resulted in rubric models different than the original model, I would compare the updated models to the originals and examine any new fixed effects.

Adding fixed effects to the 7 full dataset rubric-specific models

Missing Data

There were two missing Rating values and one missing or unspecified Sex value in the dataset. These issues were not important in the previous models on the reduced dataset, since none of the missing values occurred in the commonly rated projects. But going forward with the full dataset, I wanted to make sure I used the same set of observations for all models, so I removed the observations with missing Ratings. To make interpretation easier, I also removed the observation with the missing/unspecified Sex value, rather than consider it an additional category of the Sex variable. (See Tech. Appx. p. 14)

With the dataset cleaned up, I tried adding fixed effects (Rater, Semester, Sex, and Repeated) to the random intercept models based on the full remaining dataset. Then I did variable selection on these models where variables were considered significant based on their T-statistic.

If the variable selection process resulted in rubric models different than the original model, I compared the updated models to the originals and examined the meanings of any new fixed effects.

Adding interactions and additional random effects to the 7 full dataset rubric-specific models

For the subset of rubric models that were improved by adding fixed effects, I tried using these fixed effects to add additional interactions and random effects, if possible. If these tests resulted in rubric models different than the original model, I would compare the updated models to the originals and examine any new interactions and random effects.

Adding fixed effects, interactions, and additional random effects to the combined full dataset model

I then sought to fit a single multilevel model that predicts Rating using the full dataset. Instead of seven random intercept models where that intercept depends on artifact, I started with a single model with Rubric modeled as a random effect, grouped by Artifact. So, for each of the 91 artifacts, there are 7 coefficients, one for each rubric.

I also fit a model with the same random effect as above, but with the addition of fixed effects for each of the 5 categorical variables Rater, Semester, Sex, Repeated, and Rubric. I performed variable selection on this model and only kept the predictors that were deemed significant using T-tests.

Then using the fixed effects remaining after variable selection, I tried including their interactions, and ran the tests again. After trying the interactions, I added additional random effects based on the fixed effects that were still present in the model and tested whether they improved the model using ANOVA.

Finally, I chose the best overall model using ANOVA and by comparing information criterion AIC and BIC.

4. *Is there anything else interesting to say about this data?*

Because the Rating variable is an ordered categorical variable, not a numeric variable, I wanted to try some classification models. To reduce the scope, I chose to subset the data down to a set of observations with only two different values of Ratings, and fit logistic regression models. I used formulas from the combined models in Research Question 3 to decide on sets of predictors to include in the logistic models. Then I examined the coefficients and variances and used ANOVA, AIC, and BIC to compare the models.

RESULTS

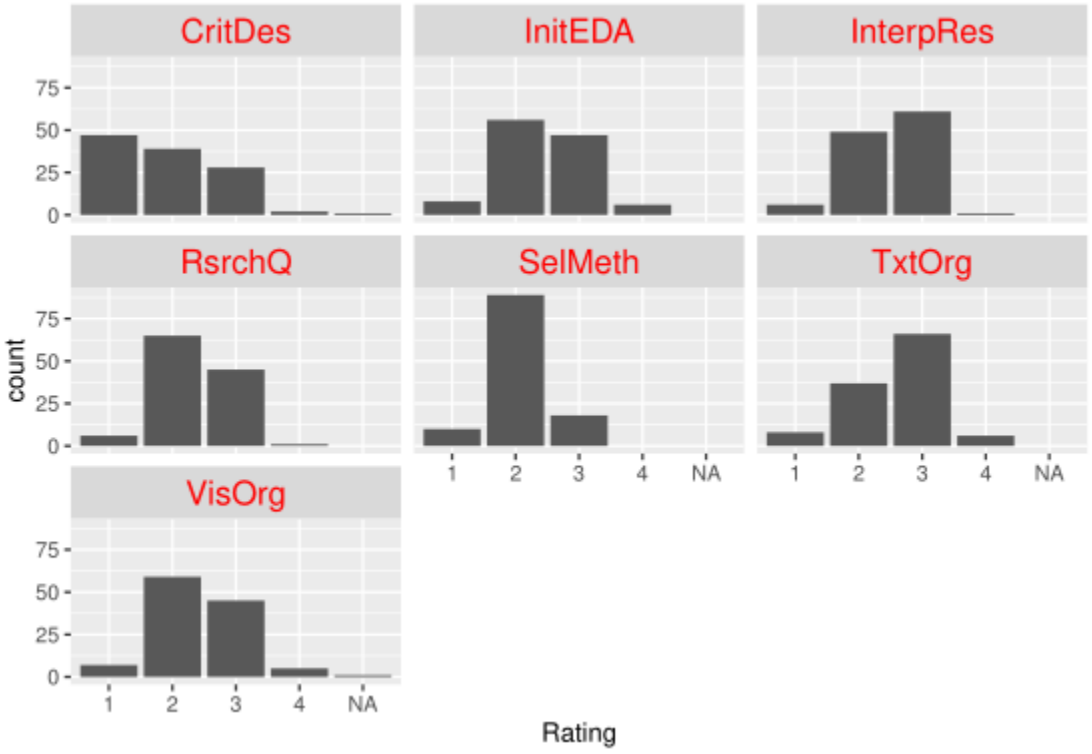
1. *Is the distribution of ratings for each rubric pretty much indistinguishable from the other rubrics, or are there rubrics that tend to get especially high or low ratings? Is the distribution of ratings given by each rater pretty much indistinguishable from the other raters, or are there raters that tend to give especially high or low ratings?*

Rubric distributions

Center and spread: The summary table (Table 4 in Data, p. 4) showed that rubric score distributions have mostly similar centers except that CritDes (critical design) and to a lesser extent, SelMeth (select methods), are lower than the rest. The averages for CritDes and SelMeth are 1.86 and 2.07, respectively, while the other rubrics all have averages close to 2.4 or 2.5. The standard deviations of ratings by rubric are all comparable and range from about 0.5 to 0.85.

Shape: From the bar plots, InitEDA (initial EDA), InterpRes (interpret results), RsrchQ (research question), TxtOrg (text organization), and VisOrg (visual organization) are all similar. The distributions are all relatively symmetric with mostly 2 and 3 ratings. CritDes has many more 1 ratings than the rest and almost no 4s. It has a strictly decreasing shape with lower numbers of each subsequent score. rating SelMeth has a much higher percent of 2s than the others and a lower average rating than all the others except CritDes. It also is the only rubric with no papers scoring 4.

Figure 2: Ratings by Rubric (full dataset)



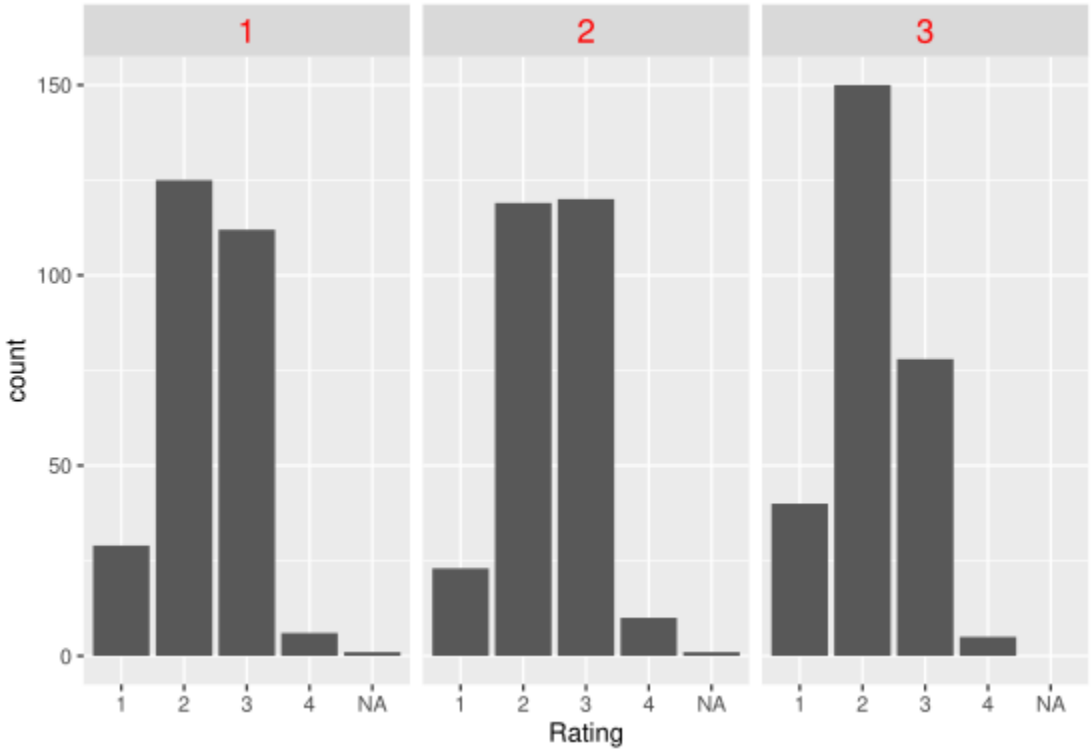
The distributions of ratings by rubric for the reduced set of 13 papers graded by all 3 raters are similar to those above (See Tech. Appx. p. 4)

Rater distributions

Center and spread: The summary table (Table 5 in Data, p. 4) showed that score distributions can vary by rater. Rater 3 sticks out the most as giving the lowest ratings (an average of 2.18), while Raters 1 and 2 give average ratings of 2.35 and 2.43, respectively. Each of the 3 raters' set of ratings has a standard deviation of about 0.7.

Shape: From the bar plots, Raters 1 and 2 distribute their ratings somewhat normally, while Rater 3 gives more irregular ratings. Raters 1 and 2 give mostly 2s and 3s by far, plus a few 1s and hardly any 4s. Rater 3 gives mostly 2s, with about half as many 3s, and about half as many 1s as that. Like the others, they give very few 4s. (See Tech. Appx. p. 4)

Figure 3: Ratings by Rater (full dataset)



The distributions of ratings by rater for the reduced set of 13 are similar to those above (See Tech. Appx. p. 7)

2. *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?*

Intraclass correlations

The intraclass correlations (ICC) for models based on the 13 papers graded by all graders show that for some rubrics (CritDes, InitEDA, SelMeth, and VisOrg), the correlation between is raters is average—in the 0.5 to 0.6 range. For the RsrchQ, InterpRes, and TxtOrg rubrics, the correlation is very low at around 0.2 or below (see Tech. Appx p. 10).

Based on papers in the full data set, the ICCs were all close to the ICCs based on the reduced data set. For five rubrics, the difference was less than 0.1, while for CritDes, the full data set ICC was 0.1 higher, and for InitEDA, the full data set ICC was almost 0.2 higher. The table below shows all 14 ICCs.

Table 13: Intraclass Correlations

Rubric	Reduced dataset ICC	Full dataset ICC
RsrchQ	0.189	0.210
CritDes	0.573	0.673
InitEDA	0.493	0.687
SelMeth	0.521	0.472
InterpRes	0.230	0.220
VisOrg	0.592	0.661
TxtOrg	0.143	0.188

Rater agreement

Based on the same set of papers, any pair of two raters gives the exact same score on the rubric for a certain paper around 65% of the time. Specific agreement rates by rubric category for any two pair of raters can be found in the Technical Appendix on page 11. These rates range from 38.5% at the lowest to 92.3% at the highest. Surprisingly, these minimum and maximum agreement rates both occur between Raters 1 and 2. Raters 1 and 3 have a narrower range of agreement rates across rubrics (53.8% to 76.9%), as do Raters 2 and 3 (53.8% to 84.6%). The average rate of agreement for any two raters is show below.

Table 14: Average Agreement Between Each Pair of Raters

Raters.1.and.2.agreement	Raters.1.and.3.agreement	Raters.2.and.3.agreement
0.626	0.637	0.67

The highest agreement rate, 67%, is between Raters 2 and 3, while the other two agreement rates are about 63%. These percentages are based on exact agreement.

It is also worth considering how often raters are not even close to agreeing. The total number of comparisons between two raters for individual student rubric ratings in the reduced set is $3 \times 13 \times 7 = 273$. The number of times that the raters disagree by more than a point is only 6 times out of 273 ratings in the whole dataset (about 2%). Interestingly, there is exactly 1 time where two raters disagree by 3 points. For a certain artifact (O2), Rater 1 gave the TxtOrg category a 4 and Rater 2 gave that same category a 1. Rater 3 gave this item a 2, making Rater 1 look like the odd one out. (See Tech. Appx p. 12).

3. *More generally, how are the various factors in this experiment (Rater, Semester, Sex, Repeated, Rubric) related to the ratings? Do the factors interact in any interesting ways?*

Adding fixed effects to the 7 reduced dataset rubric-specific models

The results of trying to add fixed effects (rater, semester, and sex) to the random intercept models based on reduced data were the same for each of the 7 rubrics—the new fixed effects were not significant based on T-statistics. Based on this subset of data, it seems that these additional factors are not related to the ratings. However, more analysis is needed on the full dataset to be sure.

Adding fixed effects to the 7 full dataset rubric-specific models

The results of trying to add fixed effects (rater, semester, and sex) to the random intercept models based on full dataset were mixed. Based on the F-test, three of the rubrics' models (InitEDA, RsrchQ, TxtOrg) included none of the potential fixed effects, just like the models based on the reduced data. However, three other rubric models (CritDes, InterpRes, VisOrg) included Rater as a fixed effect when using the full dataset. Furthermore, a single rubric model (SelMeth) included both Rater and Semester as fixed effects. The table below summarizes these results.

Table 15: Variables Included in Single Rubric Models

Rubric	Random Intercept (grouped by Artifact)	Rater (Fixed)	Semester (Fixed)	Sex (Fixed)
RsrchQ	yes	no	no	no
CritDes	yes	yes	no	no
InitEDA	yes	no	no	no
SelMeth	yes	yes	yes	no
InterpRes	yes	yes	no	no
VisOrg	yes	yes	no	no
TxtOrg	yes	no	no	no

These results suggest that for the four rubrics CritDes, InterpRes, VisOrg, and SelMeth, the rating is related to which rater is grading the paper (and for SelMeth, the rating is also related to which semester the student took the class). Note that this subset of rubrics does not line up with the subset of rubrics that had lower intraclass correlations. I thought that perhaps the rubrics with the lowest measures of rater agreement would also be the rubrics whose rating models depended on rater. But that is not the case. The three rubrics with low ICCs were RsrchQ, InterpRes, and TxtOrg; only one of these (InterpRes) has a model that includes Rater as a predictor.

Adding interactions and additional random effects to the 7 full dataset rubric-specific models

For a variable to make sense as a random effect in a model, it should also be present as a fixed effect, so I only intended to add additional random effects in the models for the four rubrics CritDes, InterpRes, VisOrg, and SelMeth. But the dataset does not have enough observations to handle any more random effects. Note that each individual rubric model using the full dataset only has 116 or 117 observations, depending on missing values.

The only rubric model that included two fixed effects was the model for SelMeth. It included the predictors Rater and Semester. I tested the addition of these two variables' interaction to the SelMeth model, but the interaction did not improve the model.

So, for all seven rubric-specific models, there were no additional interactions or random effects that both improved a model and could possibly be added to the model.

Adding fixed effects, interactions, and additional random effects to the combined full dataset model

After testing different combinations of fixed effects, random effects, and interactions and comparing the various models, the best model for predicting rating included the following variables:

- Random effect for Rubric, grouped by Artifact
- Random effect for Rater, grouped by Artifact
- Rater as a fixed effect
- Semester as a fixed effect
- Rubric as a fixed effect
- Interaction between fixed effects Rater and Rubric

The model selection process can be read on pages 18-27 of the Technical Appendix.

So, the factors rubric, rater, and semester are all related to the ratings. Rubric and rater are related as fixed effects, as random effects, and as an interaction with each other.

The table below summarizes the coefficients for the fixed effects, as well as the values of τ^2 for the random effects. Recall that τ^2 represents the variance of the mean Artifact ratings for a certain rubric. There are far too many individual random effect coefficients to list here. More of the coefficients can be found on page 27 of the Technical Appendix.

Tables 16-17: Fixed Effect Coefficients and Random Effect Variances in Final Model

Fixed Effect	**Coefficient**
Intercept	1.76
Rater2	0.37
Rater3	0.20
SpringSemester	-0.16
InitEDA	0.74
InterpRes	0.99
RsrchQ	0.73
SelMeth	0.41
TxtOrg	1.02
VisOrg	0.65
InitEDA: Rater2	-0.30
InitEDA: Rater3	-0.29
InterpRes: Rater2	-0.51
InterpRes: Rater3	-0.71
RsrchQ: Rater2	-0.49
RsrchQ: Rater3	-0.32
SelMeth: Rater2	-0.39
SelMeth: Rater3	-0.39
TxtOrg: Rater2	-0.55
TxtOrg: Rater3	-0.44
VisOrg: Rater2	-0.10
VisOrg: Rater3	-0.28

Random Effect	τ^2
CritDes	0.50
InitEDA	0.32
InterpRes	0.10
RsrchQ	0.18
SelMeth	0.04
TxtOrg	0.25
VisOrg	0.23
Rater1	0.01
Rater2	0.11
Rater3	0.09

4. Is there anything else interesting to say about this data?

The subset of data I used was all observations with Ratings of 2 or 3. Besides the fact that I wanted to do logistic regression, and not multinomial or some other method with more than two categories of the response variable, there were two main reasons for choosing this subset. The first was that this would give me the largest possible subset of data. Ratings of 2 and 3 are by far the most common in the dataset. So, this subset only reduces the dataset from 810 to 697 total ratings. The other reason was that scores of 2 and 3 can intuitively be divided into generally "good" and "bad" ratings. According to Table 1 on page 2, ratings of 3 are given when "Student generates competent evidence; no flaws, or only minor ones" and ratings of 2 are given when "Student generates evidence with significant flaws." The key words are "competent" vs "flaws." On the other hand, Ratings of 3 and 4 could both be considered good and Ratings of 1 and 2 could both be considered bad, so I would expect it to be harder to distinguish between ratings in either of those pairings.

The model formulas I chose were all the results of variable selection procedures in Research Question 3, so insignificant predictors should already be filtered out. I also selected the most basic combined model that had only a random intercept for each artifact group and no other predictors, so I could test whether additional variables were necessary in the model at all. The four formulas I chose to examine are summarized below:

1. Response: Rating with categories 2 and 3
   Predictors: Random effect for Rubric grouped by Artifact
2. Response: Rating with categories 2 and 3
   Predictors: Random effect for Rubric grouped by Artifact, fixed effects for Rater, Semester, and Rubric
3. Response: Rating with categories 2 and 3
   Predictors: Random effect for Rubric grouped by Artifact, fixed effects for Rater, Semester, and Rubric, interaction between fixed effects Rater and Rubric
4. Response: Rating with categories 2 and 3
   Predictors: Random effect for Rubric grouped by Artifact, random effect for Rater grouped by Artifact fixed effects for Rater, Semester, and Rubric, interaction between fixed effects Rater and Rubric

The model summaries can be found in the Technical Appendix on pages 37-41.

A comparison of all four models using ANOVA, AIC, and BIC showed that the 2nd model list above was the best. Therefore, the original random effect for Rubric grouped by Artifact and fixed effects for Rater, Semester, and Rubric are optimal set of predictors when modeling the ratings this way.

The coefficients of the final model mostly had high standard errors, resulting in the coefficient for Rater 3 being the only fixed effect coefficient that was significant based on its T-statistic. The table below summarizes the coefficients for the fixed effects and the values of τ^2 for the random effects in this model.

Tables 18-19: Fixed Effect Coefficients and Random Effect Variances in Logistic Model

Fixed Effect	**Coefficient**
Intercept	-1.39
Rater2	0.07
Rater3	-1.19
SpringSemester	-0.05
InitEDA	1.43
InterpRes	2.04
RsrchQ	1.21
SelMeth	-5.58
TxtOrg	2.71
VisOrg	1.32

Random Effect	τ^2
CritDes	27.23
InitEDA	3.39
InterpRes	1.92
RsrchQ	2.05
SelMeth	64.87
TxtOrg	2.18
VisOrg	2.55

DISCUSSION

1. *Is the distribution of ratings for each rubric 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?*

Rubric distributions

The results of the EDA on ratings grouped by rubric showed that rubrics CritDes and SelMeth had the lowest average ratings (1.86 and 2.07, respectively) by a noticeable margin. CritDes had the lowest of all. The other 5 rubric averages were all close to one another at around 2.4-2.6. The bar graphs Figure 7, page 2 shed light on these two low averages. CritDes was the only rubric with a mode rating of 1 and SelMeth had the highest percent of 2 ratings of any rubric. This implies that CritDes and SelMeth were the most difficult aspects of the project for students.

There are always going to be certain aspects of a project that are more difficult than others, but the fact the group of students had difficulty with these two suggests that an adjustment to the project could be made. Perhaps instructors could explain these two rubrics better to students to clear up confusion. Or if these two areas are just more difficult by nature, maybe graders should be asked to go easier on students when rating these sections. These two rubrics are defined in Table 3, page 3, as follows:

> *CritDes* (Critique Design): Given an empirical research question, the student critiques or evaluates to what extent a study design convincingly answer that question.
>
> *SelMeth* (Select Methods): Given a data set and a research question, the student selects appropriate method(s) to analyze the data.

I can imagine how critiquing someone else's work could be an especially difficult ask of freshman students, compared with producing their own work. Maybe instructors could give more examples of critiques of study designs so that students are more likely to score higher in this area.

I can also see how selecting an appropriate method would result in a lot of low scores. It's hard to partially select a correct method—usually you select a correct method or an incorrect one. This could be why so many students score 2s ("evidence with significant flaws") on this rubric. One idea would be to ask raters to give more 3s if a student selects the wrong method but justifies it in a reasonable way. Another possibility is leaving it alone and acknowledging that some parts of a project are harder than others.

Rater distributions

The results of the EDA on ratings grouped by rater showed that Rater 3 gave the lowest average ratings. The bar graphs in Figure 3, page 8 illustrate why. Rater 1 gives more 1s, more 2s, and fewer 3s than Raters 1 and 2.

There are many potential reasons for this discrepancy. The raters all come from different departments, so they all have different training and backgrounds. Rater 3 could be in the statistics department or another quantitative department where they took a few statistics courses. This might lead them to have higher standards when grading statistics projects. Rater 3 could also have different qualities as an individual compared to the others. Maybe they're just stricter or maybe they didn't understand the rating scale as well as the other two. Conversely, Rater 3 could be the one scoring papers the way they're supposed to and Raters 1 and 2 could be grading too generously.

If every student was being graded by each rater and given the averages of their scores, this would not be a big issue. However, because most students are only graded by one rater, the fact that one rater gives lower ratings is a problem. The Dietrich College should make an effort to train raters better or choose raters more carefully, so they give more comparable ratings.

Something to note regarding this and other discussions is that the rubrics and rating scale used for these ratings were experimental, and not typical of the way freshman statistics projects are graded. Despite this, much of the discussion makes recommendations based on the assumption that these ratings are representative of the regular way projects are rated at the Dietrich School, in terms of quality and consistency.

2. *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?*

Intraclass correlations

For rubrics CritDes, InitEDA, SelMeth, and VisOrg, the ICC was average to somewhat high, with values around 0.5-0.7. These can be considered moderately reliable ratings according to Koo (2016). The RsrchQ, InterpRes, and TxtOrg rubrics had lower correlation of around 0.2. These correlations indicate low reliability of ratings.

If all the rubric ICCs were in that 0.5-0.7 range, I would still have suggested aiming to increase the correlations and trying to reach ICCs of 0.75 or even 0.9. But given the three low ICCs around 0.2, I suggest trying to at least increase these to the level of the others. Clearly, raters need more hours of training or higher quality training in order to give fairer ratings to students.

Rater agreement

The agreement percentages paint a better picture of rates of agreement between raters than the ICCs. The three possible pairs of raters agree with each other 63%, 64%, and 67% of the time, which doesn't sound that bad. Rater 1 is part of the pairing in the two lowest of those percentages, which means Rater 1 is the one who disagrees with the others the most (by a very slight amount).

For individual rubrics, it doesn't look as good. Pairwise agreement rates range from 38.5% to 92.3% (See Tech. Appx p. 13), which is quite a wide range. This again points to inconsistency in ratings. Percentages like 84.6% (Raters 1 and 2 on InitEDA) and 76.9% (Raters 2 and 3 on RsrchQ) are numbers to aim for, but

the percentages like 38.5% (Raters 1 and 2 on RsrchQ) and 53.8% (many examples) show that more work needs to be done to get raters on the same page.

Most of the disagreement discussed above was disagreement by 1 point. There were also cases where raters disagreed by more than 1 point. When the ratings only go from 1 to 4, you would hope that disagreement by 2 or 3 points is rare. Thankfully it is, as raters only disagreed by 2 or more points 6 times out the 273 times their ratings were compared (about 2%). There was also one case (TxtOrg on artifact O2) where Rater 1 scored it 4, Rater 2 scored it 1, and Rater 3 scored it 2. Something seriously wrong must happened for a Rater to give a 4 where the others gave 1 and 2. Hopefully it was some sort of recording error rather than a sign of a rater's complete misunderstanding of how to grade TxtOrg. Because this 'more than 1' disagreement happens so rarely, but represents a very serious discrepancy in rater's perceptions, I would recommend flagging any project where this occurs and examining it more closely.

3. *More generally, how are the various factors in this experiment (Rater, Semester, Sex, Repeated, Rubric) related to the ratings? Do the factors interact in any interesting ways?*

To recap, the factors that I found to be related to the ratings based on their inclusion in the multilevel model are listed below:

- Random effect for Rubric, grouped by Artifact
- Random effect for Rater, grouped by Artifact
- Rater as a fixed effect
- Semester as a fixed effect
- Rubric as a fixed effect
- Interaction between fixed effects Rater and Rubric

The random effect for Rubric grouped by Artifact means that for any given rubric, for instance InitEDA, the change in mean ratings for different Artifacts have a variance of τ^2, or in this case $\tau^2 = 0.32$ (See Tables 13-14, page 12). Because the Rubric InitEDA also has a fixed effect coefficient of 0.74 and the model has an overall intercept of 1.76, this means scores in the InitEDA category are normally distributed across Artifacts with a mean of 1.76 + 1.74 = 2.5 and a variance of 0.32.

Once a specific Artifact is drawn from this distribution, we can see the actual predicted mean rating for a given rubric for an Artifact. For instance, the random effect coefficient for rubric InitEDA in Artifact 100 is −0.26 (See Tech. Appx. p. 29), so the expected rating in the InitEDA category for Artifact 100 is 2.5 – 0.26 = 2.24.

The random effect for Rater grouped by Artifact can be interpreted in a similar way. For example, ratings by Rater 2 are centered at 1.76 + 0.37 = 2.13, with a variance of 0.11 across Artifacts. (See Tables 13-14, page 12).

The fixed effect coefficients for Rater and Rubric are difficult to interpret because of the interaction between them also included in the model. For instance, for Raters, the intercept 1.76 represents Rater 1, and the coefficient for Rater 3, 0.20, suggests that Rater 3 ratings are predicted to be 0.20 higher than Rater 1 ratings, all other variables held constant. But we know that Rater 3 actually gave the lowest average ratings overall (See Table 5, p. 4). This conflicting information is due to the interaction term between Rater and Rubric. You can't switch from Rater 1 to Rater 3 and hold all other variables constant because Rater 1 would also switch to Rater 3 in that interaction terms. The fact that Rater 3 gives the lowest ratings is likely accounted for somewhere in the coefficients of the interaction terms, but in a way that is not obvious. Rubric coefficients are similarly hard to interpret.

Semester is the only variable that's easy to interpret. Ratings for papers written in the spring semester are predicted to be 0.16 less than ratings in the fall semester. This makes sense considering Table 6 on page 4, which shows that spring ratings are lower than fall ratings. It doesn't seem desirable for spring papers to score lower. One possible explanation is that Raters became more perceptive towards flaws after a semester of grading. This problem would most likely go away over time if the same raters kept their jobs. It also could keep repeating the pattern if new raters are hired every year and those raters grade papers in the fall and again in the spring. If it is possible to keep raters on the job for longer terms, I would recommend that, so that they gain more experience and improve their consistency over time.

A worse possibility is that students received lower quality teaching in one semester compared to the other. It seems unlikely, but I would suggest checking professor ratings for each General Education freshman statistics class and checking whether lower rating on projects correspond to lower ratings of instructors.

The presence of Rater as a fixed and random effect in the model gives additional evidence to that found in Research Questions 1 and 2 (See Discussion, p. 14) that the three raters grade differently. The fixed effect suggests that they grade differently in general, and the random effect suggests that the differences in their grading changes from one artifact to the other. Again, this suggests that raters need better training, more training, or some other strategy to get them grade more consistently. Perhaps choosing raters that are more alike in measurable ways would help. For instance, you could choose raters with the same major, from the same department, in the same year of school, or raters that have taken a specific class.

The presence of Rubric as a fixed and random effect in the model gives additional evidence to that found in Research Question 1 (See Discussion, p. 14) that the seven rubrics have different difficulty levels. I don't think this is necessarily a terrible problem, but if the college wants rubrics to all end up with similar average scores, steps could be taken to attempt to raise scores in the two rubrics with lower scores, as mentioned in the Question 1 Discussion. Note that the two rubrics discussed in that section with the lowest average scores, CritDes and SelMeth, also have the lowest coefficients in the model. CritDes is represented by the intercept, while all other rubric coefficients are positive, and SelMeth is the least positive coefficient. This gives further evidence that they are the most difficult.

The interaction of Rubric and Rater in the model suggests that each rater grades rubrics in different ways that does not exactly correspond to the ways that raters already grade differently in general, or the way that rubrics are already graded differently in general. Because these differences are very

difficult to pinpoint, I would recommend focusing on the other factors that relate to ratings discussed above.

4. Is there anything else interesting to say about this data?

The coefficient/variance table of the best logistic model shows some extreme values for fixed effect coefficients, standard errors of those coefficient, and τ^2 values for random effects. In fact, all four models ended up having very large values of τ^2 for their random effects, as well as highly varying fixed effect coefficients with values as high as 7.3 and as low as −6.1 (See Tech. Appx. p. 41).

Although coefficients β in logistic models are interpreted as multiplicative change by e^{β} instead of additive change as in linear models, the values still seem extreme. However, it is interesting to note that all but one fixed effect coefficient has the same sign as they did in the equivalent model with numeric Ratings. In that model, all coefficients were positive except that of Rater 3 and the Spring Semester. In this logistic model, those same two coefficients are negative. One of them, Rater 3, is the only significant coefficient, so this model seems to at least pick up some of the same signal as the other model (recall that Rater 3 gave the lowest average ratings of the raters). But it also has one additional negative coefficient—the coefficient for the SelMeth rubric is the most extreme in the model at −5.58, whereas it was slightly positive in the equivalent linear model. This coefficient is not significant so it may not be worth interpreting, but either way, it doesn't reflect highly on this model.

The fact that variances were so high and hardly any coefficients were significant suggests that using logistic instead of linear regression was not a great way to model the data. It is also possible that subsetting the data down to only ratings of 2 and 3 is throwing off the model. It's possible that rubrics rated 2 and 3 do not contrast enough for the predictors to distinguish them. It may be worthwhile to try this analysis again with some kind of ordered multinomial regression that includes all 4 ratings.

Then again, the original modeling framework that treated Ratings as numeric may have been the best method after all. Since our main goal is looking for the factors in the experiment that relate to ratings, not predicting ratings, using a numeric version of the response may not have been problematic at all.

REFERENCES

Junker, B. W. (2021). *Project 02 assignment sheet and data for 36-617: Applied Regression Analysis.* Department of Statistics and Data Science, Carnegie Mellon University, Pittsburgh PA. Accessed Nov 15, 2021 from https://canvas.cmu.edu/courses/25337/files/folder/Project02

Koo, Terry K, and Mae Y Li. "A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research." *Journal of chiropractic medicine vol. 15,2* (2016): 155-63. doi:10.1016/j.jcm.2016.02.012

TECHNICAL APPENDIX

Zach Ohl

12/10/2021

```
library(lme4, quietly = T) #for lmer()
library(ggplot2)
library(tidyverse)
library(kableExtra)
library(performance)  #for icc
library(LMERConvenienceFunctions, quietly = T ) #for MLM var selection
library(GGally)
library(ggpubr)
library(RLRsim) # for exactRLRT test
library(gridExtra) # for grid arrange
```

```
#read  data
ratings <- read.csv(file = paste0("C:/Users/Zachary Ohl/Desktop/CMU courses/",
                                  "Applied Linear Models/project 2/ratings.csv"))
ratings_tall <- read.csv(file = paste0("C:/Users/Zachary Ohl/Desktop/CMU courses/",
                                       "Applied Linear Models/project 2/tall.csv"))
#Make non M/F sex values consinsent:
ratings_tall$Sex[ratings_tall$Sex==""] <- "--"

rubric_ratings <- ratings[, 7:13]

# Make sure all ratings run from 1 to 4,
ratings_tall$Rating <- factor(ratings_tall$Rating,levels=1:4)
for (i in unique(ratings_tall$Rubric)) {
  ratings[,i] <- factor(ratings[,i],levels=1:4)
}

#attach(ratings)
rubric_all3 <- ratings[ !is.na(ratings$Overlap), c(2, 7:13, 14)]
#includes rater(col 2) and artifact (col 14)
ratings_all3 <- ratings[ !is.na(ratings$Overlap), ] #includes all columns
ratings_tall_all3 <- ratings_tall[ ratings_tall$Repeated==1, ] #includes all columns
```

EDA Ratings overall:

```
ratings_tall_noNA <- ratings_tall[-c(161,684),]   #remove NAs

ggplot(ratings_tall_noNA, aes(x=Rating)) + geom_bar()
```

Ratings by rubric:

```
temp_summary <- apply(rubric_ratings[, c(1, 3,4,5, 7)],2,
                      function(x) c(summary(x),
                     SD=sd(x))) %>%
```

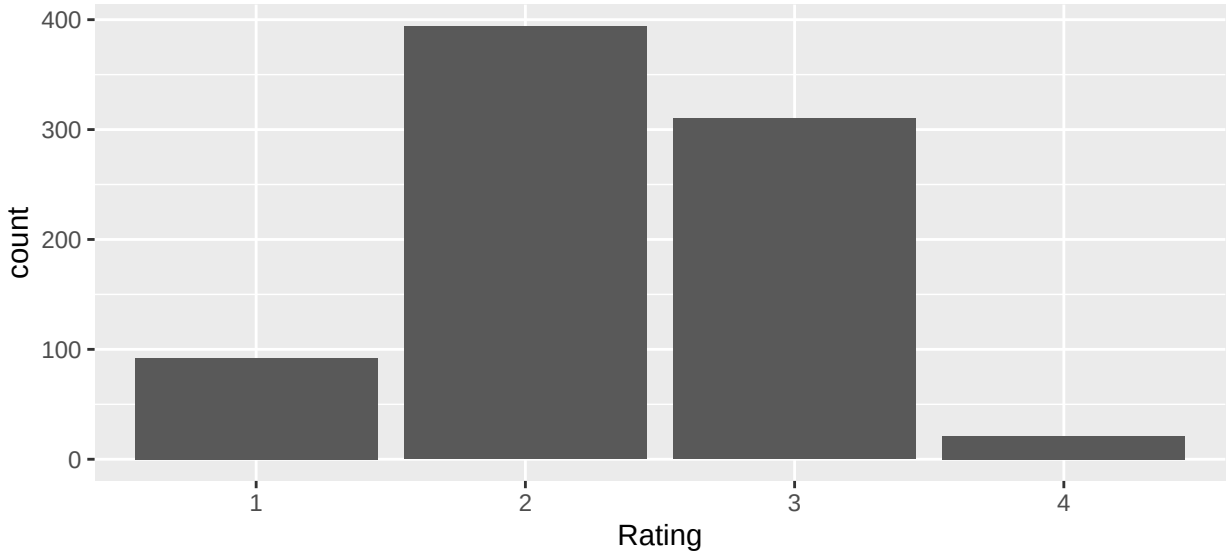


Figure 1: Overall distribution of ratings

```
  as.data.frame %>% t() %>% round(digits=2)
temp_summ_na <- apply(na.omit(rubric_ratings[, c(2, 6)]),2,
                      function(x) c(summary(x),
                                    SD=sd(x))) %>%
  as.data.frame %>% t() %>% round(digits=2)

sum_table <- rbind(temp_summary, temp_summ_na)
sum_table %>%
  kable(caption = "Summary tables of the ratings by rubric") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 1: Summary tables of the ratings by rubric

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
RsrchQ	1	2	2	2.35	3.0	4	0.59
InitEDA	1	2	2	2.44	3.0	4	0.70
SelMeth	1	2	2	2.07	2.0	3	0.49
InterpRes	1	2	3	2.49	3.0	4	0.61
TxtOrg	1	2	3	2.60	3.0	4	0.70
CritDes	1	1	2	1.86	2.5	4	0.84
VisOrg	1	2	2	2.42	3.0	4	0.68

Ratings by rater:

```
rater_list <- list("Rater1" = as.numeric(ratings_tall_noNA$Rating[ratings_tall_noNA$Rater==1]),
                   "Rater2" = as.numeric(ratings_tall_noNA$Rating[ratings_tall_noNA$Rater==2]),
                   "Rater3" = as.numeric(ratings_tall_noNA$Rating[ratings_tall_noNA$Rater==3]) )

temp_summary2 <- lapply(rater_list, function(x) c(summary(x),SD=sd(x))) %>%
  as.data.frame %>% t() %>% round(digits=2)
```

```
temp_summary2  %>%
  kable(caption = "Summary tables of the ratings by rater") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 2: Summary tables of the ratings by rater

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
Rater1	1	2	2	2.35	3	4	0.70
Rater2	1	2	2	2.43	3	4	0.70
Rater3	1	2	2	2.18	3	4	0.69

Ratings by semester:

```
sem_list <-
  list("Fall" = as.numeric(ratings_tall_noNA$Rating[ratings_tall_noNA$Semester=='F19']),
       "Spring" = as.numeric(ratings_tall_noNA$Rating[ratings_tall_noNA$Semester=='S19']) )

temp_summary3 <- lapply(sem_list, function(x) c(summary(x),SD=sd(x))) %>%
  as.data.frame %>% t() %>% round(digits=2)

temp_summary3  %>%
  kable(caption = "Summary tables of the ratings by semester") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 3: Summary tables of the ratings by semester

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
Fall	1	2	2	2.36	3	4	0.67
Spring	1	2	2	2.23	3	4	0.78

Ratings by sex:

```
sex_list <- list("Male" = as.numeric(ratings_tall_noNA$Rating[ratings_tall_noNA$Sex=='M']),
                 "Female" = as.numeric(ratings_tall_noNA$Rating[ratings_tall_noNA$Sex=='F']) )

temp_summary4 <- lapply(sex_list, function(x) c(summary(x),SD=sd(x))) %>%
  as.data.frame %>% t() %>% round(digits=2)

temp_summary4  %>%
  kable(caption = "Summary tables of the ratings by sex") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 4: Summary tables of the ratings by sex

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
Male	1	2	2	2.31	3	4	0.71
Female	1	2	2	2.31	3	4	0.70

Ratings by Repeated or not Repeated:

```
rep_list <- list("Repeated" = as.numeric(ratings_tall_noNA$Rating[ratings_tall_noNA$Repeated==1]),
                 "NotRepeated" = as.numeric(ratings_tall_noNA$Rating[ratings_tall_noNA$Repeated==0]) )

temp_summary5 <- lapply(rep_list, function(x) c(summary(x),SD=sd(x))) %>%
  as.data.frame %>% t() %>% round(digits=2)

temp_summary5  %>%
  kable(caption = "Summary tables of the ratings by Repeated") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 5: Summary tables of the ratings by Repeated

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
Repeated	1	2	2	2.27	3	4	0.66
NotRepeated	1	2	2	2.34	3	4	0.72

Categorical variable counts:

```
#tmplist %>% kable()
#  kable(caption = "Summary tables of the categorical variables") %>%
#  kable_styling(latex_options = "HOLD_position")

knitr::kable(
  tmplist,
  caption = 'Summary tables of the categorical variables',
  booktabs = TRUE, valign = 't'
)
#this chunk prevents knitting for some reason. cant evaluate
```

Question 1: 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?

Look at distributions of ratings by rubric

Look at mean and 5-number summaries of ratings by rubric:

```
sum_table %>%
  kable(caption = "Summary table of the numeric variables") %>%
  kable_styling(latex_options = "HOLD_position")
```

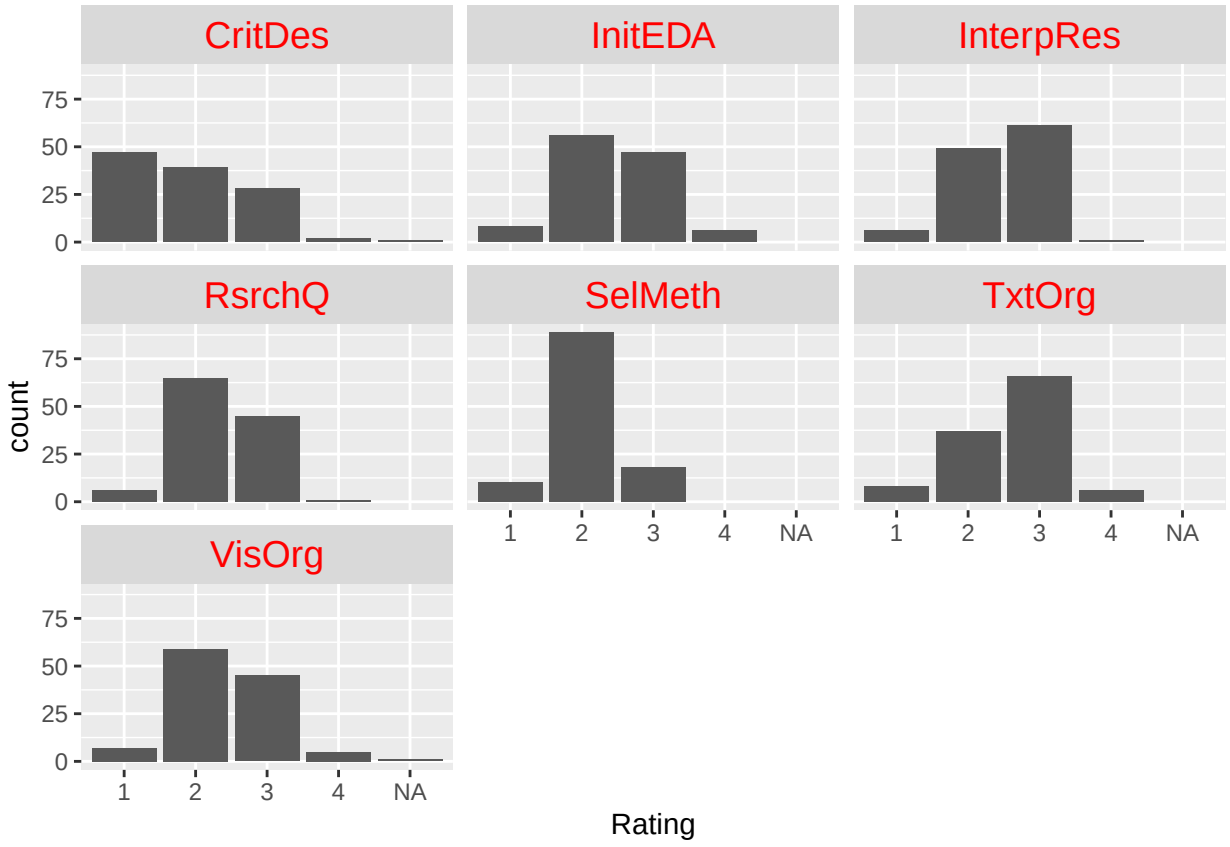
Table 6: Summary table of the numeric variables

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
RsrchQ	1	2	2	2.35	3.0	4	0.59
InitEDA	1	2	2	2.44	3.0	4	0.70
SelMeth	1	2	2	2.07	2.0	3	0.49
InterpRes	1	2	3	2.49	3.0	4	0.61
TxtOrg	1	2	3	2.60	3.0	4	0.70
CritDes	1	1	2	1.86	2.5	4	0.84
VisOrg	1	2	2	2.42	3.0	4	0.68

The rubric score distributions are mostly similar except CritDes (critical design) and to a lesser extent, SelMeth (Method selection), are lower than the rest.

Look at the shapes of distributions of ratings by rubric:

```
## Bar plots for the whole data set. NAs dont show up?
ggplot(ratings_tall, aes(x = Rating)) +  facet_wrap( ~ Rubric) +  geom_bar() +
  theme(strip.text = element_text(size = 14, color = "red"))
```

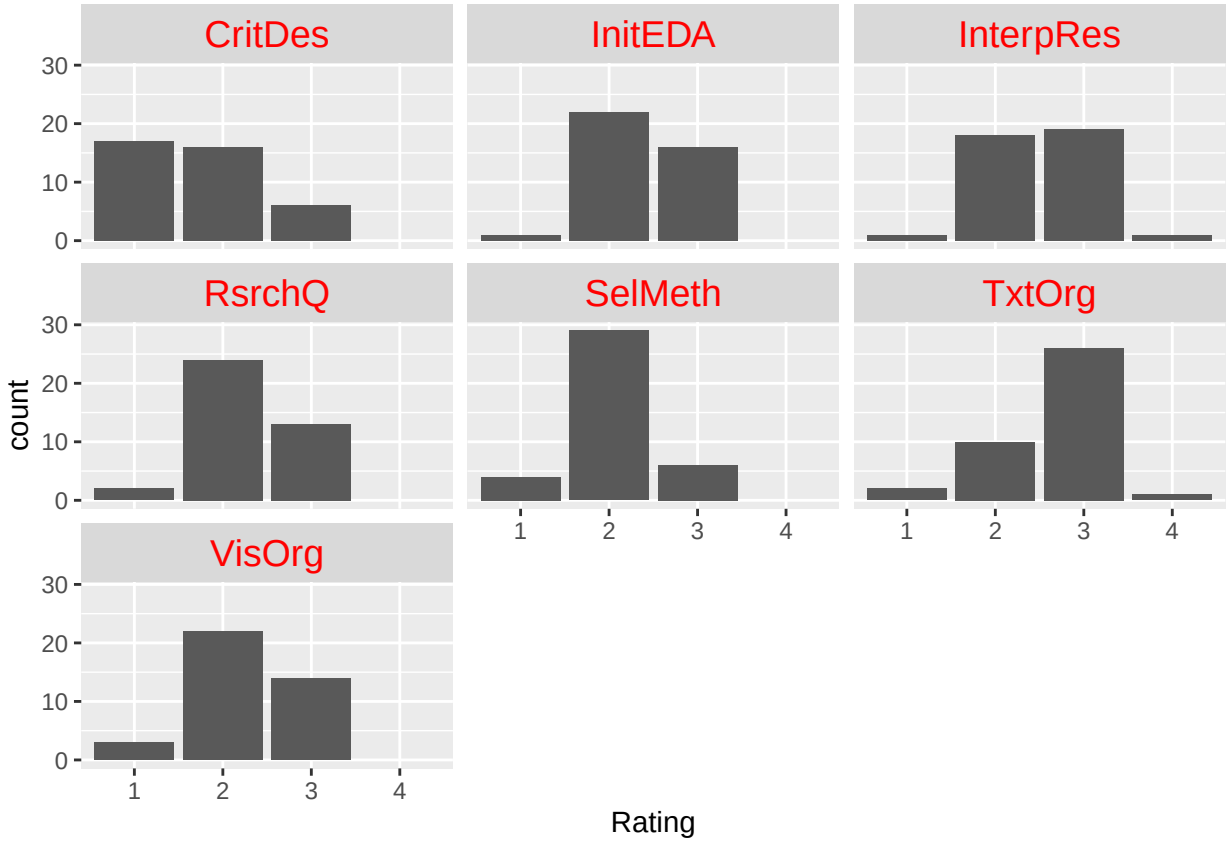


InitEDA, InterpRes, RserchQ, TxtOrg, and VisOrg are all similar. CritDes has much more 1 ratings than the rest and almost no 4s. SelMeth has a much higher percent of 2s than the others and a lower average rating than all the others except CritDes.

Same plots but for only papers graded by all 3 raters:

```
## Bar plots for the reduced data set
ggplot(ratings_tall_all3, aes(x = Rating)) +  facet_wrap( ~ Rubric) +
```

```
  geom_bar() + theme(strip.text = element_text(size = 14, color = "red"))
```



The distributions look similar to the overall ratings.

Now look at distributions of ratings by rater:

Look at mean and 5-number summaries of ratings by rater:

```
temp_summary2  %>%
  kable(caption = "Summary tables of the ratings by rater") %>%
  kable_styling(latex_options = "HOLD_position")
```

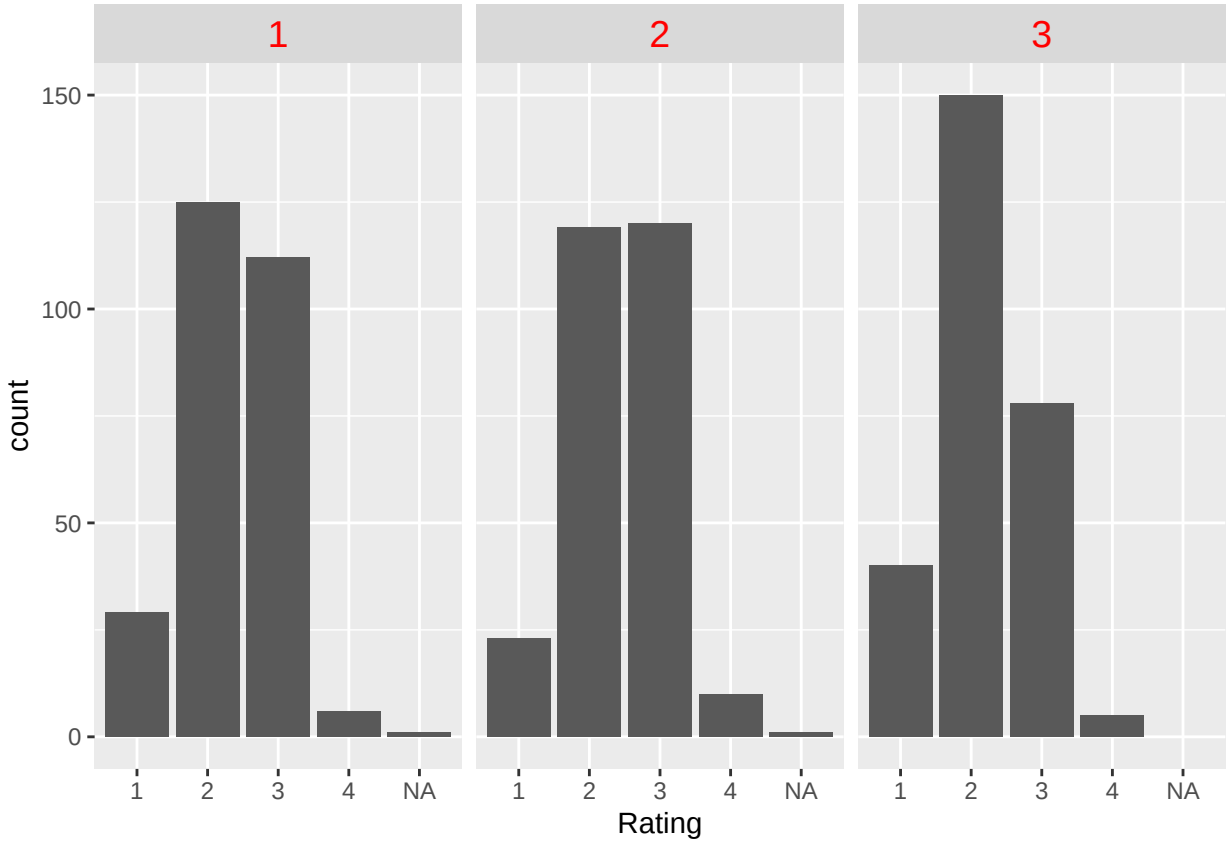
Table 7: Summary tables of the ratings by rater

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
Rater1	1	2	2	2.35	3	4	0.70
Rater2	1	2	2	2.43	3	4	0.70
Rater3	1	2	2	2.18	3	4	0.69

The rater score distributions are not too dissimilar, except Rater 3. They all have the same spread, but Rater 3 gives a noticeably lower average rating than the other two.

Look at the shapes of distributions of ratings by rater:

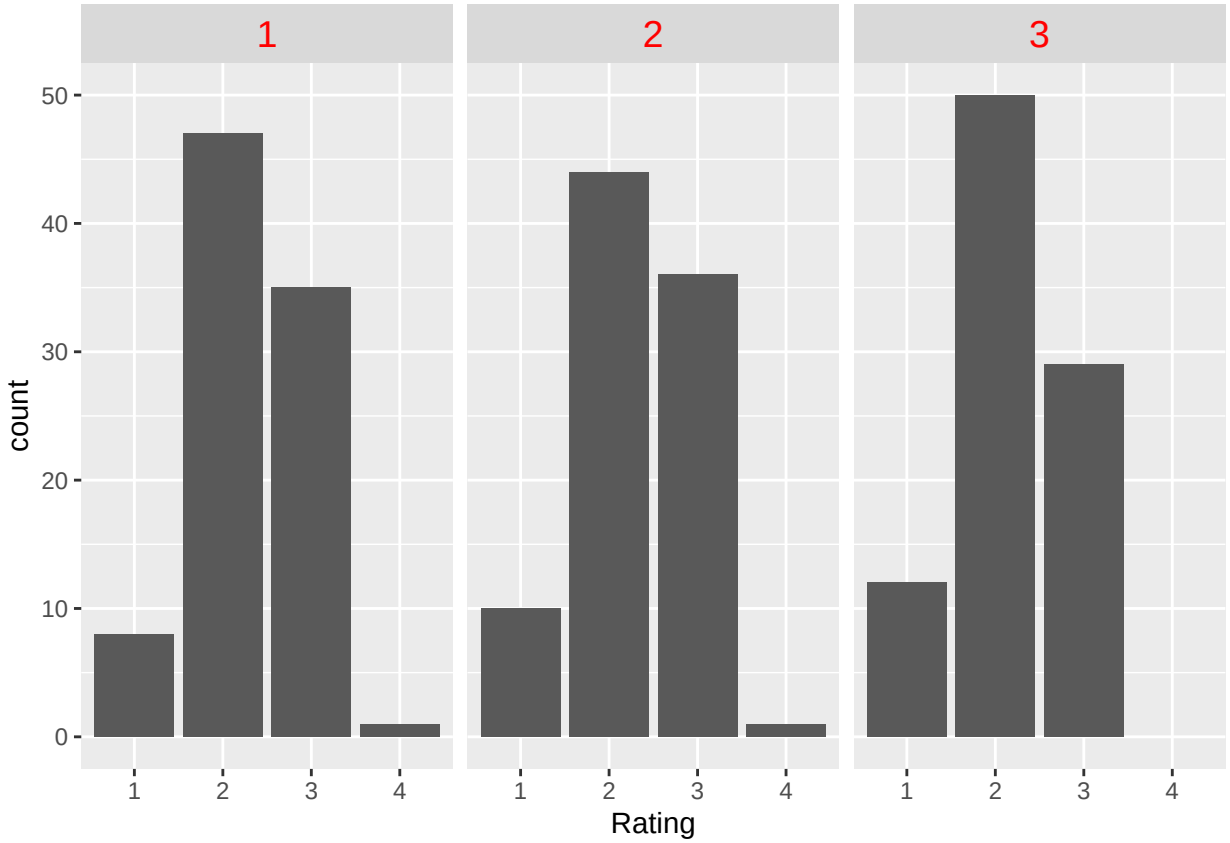
```
## Bar plots for the whole data set. NAs dont show up?
ggplot(ratings_tall, aes(x = Rating)) +  facet_wrap( ~ Rater) +  geom_bar() +
  theme(strip.text = element_text(size = 14, color = "red"))
```



From the distributions, it seems Raters 1 and 2 distribute their ratings somewhat normally, while Rater 3 gives more irregular ratings.

Same plots but for only papers graded by all 3 raters:

```
## Bar plots for the reduced data set
ggplot(ratings_tall_all3, aes(x = Rating)) +  facet_wrap( ~ Rater) +
  geom_bar() + theme(strip.text = element_text(size = 14, color = "red"))
```



The distributions look more similar between raters based on this smaller subset. Each rater gives the most 2s, followed by 3s, and then 1s.

Check for NAs:

```
#any(is.na(ratings_tall$Rating))  #True
ratings_tall[apply(ratings_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>
```

One NA score for CritDes and one for VisOrg. None are the in the data set of the 13 papers graded by all raters. Will need to drop these two observations of replace the NAs with values for models on the full data set, so R doesn't fit models to slightly different data sets depending on the rubric used.

Also, note the one missing or nonbinary sex value:

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

```
##       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
```

This artifact is also not in the set of 13 commonly graded papers.

Question 2: 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?

Use the 13 papers that were each rated by all 3 raters to fit seven random intercept models - one for each rubric: These models have 13 groups each - one for each artifact.

```
randint_models = list()
#, REML = F
randint_models[[1]] <- lmer(as.numeric(RsrchQ) ~ 1 + (1 | Artifact), data = rubric_all3)
randint_models[[2]] <- lmer(as.numeric(CritDes) ~ 1 + (1 | Artifact), data = rubric_all3)
randint_models[[3]] <- lmer(as.numeric(InitEDA) ~ 1 + (1 | Artifact), data = rubric_all3)
randint_models[[4]] <- lmer(as.numeric(SelMeth) ~ 1 + (1 | Artifact), data = rubric_all3)
randint_models[[5]] <- lmer(as.numeric(InterpRes) ~ 1 + (1 | Artifact), data = rubric_all3)
randint_models[[6]] <- lmer(as.numeric(VisOrg) ~ 1 + (1 | Artifact), data = rubric_all3)
randint_models[[7]] <- lmer(as.numeric(TxtOrg) ~ 1 + (1 | Artifact), data = rubric_all3)
names(randint_models) <- c("RsrchQ", "CritDes", "InitEDA", "SelMeth", "InterpRes","VisOrg","TxtOrg")
```

Note that each model has 13 coefficients, one for each artifact. For example, coefficients of the first rubric model:

```
coef(randint_models[[1]])
```

```
## $Artifact
##     (Intercept)
## O1     2.303167
## O10    2.303167
## O11    2.577677
## O12    2.165913
## O13    2.165913
## O2     2.165913
## O3     2.440422
## O4     2.303167
## O5     2.440422
## O6     2.165913
## O7     2.440422
## O8     2.028658
## O9     2.165913
##
## attr(,"class")
## [1] "coef.mer"
```

Find the intraclass correlation (ICC) between raters for each rubric. This can be used as a measure of agreement between raters.

ICCs: Make sure *icc* function from *performance* library works:

```
#check:
artifact_RsrchQ_randint <- lmer(as.numeric(RsrchQ) ~ 1 + (1 | Artifact),
                                data = rubric_all3)  #fit one model
#summary(artifact_RsrchQ_randint)
performance::icc(artifact_RsrchQ_randint)  #find icc using function
```

```
## # Intraclass Correlation Coefficient
##
##      Adjusted ICC: 0.189
##   Conditional ICC: 0.189
```

```
0.05983/(0.05983 + 0.25641)    #find icc using printed values and formula
```

```
## [1] 0.1891918
```

Both outputs from the *icc* function match the hand-calculated value.

Find all ICCs:

```
unlist(lapply(randint_models, FUN = performance::icc))[seq(from=2, to=14, by=2)]
```

```
##    RsrchQ.ICC_conditional   CritDes.ICC_conditional   InitEDA.ICC_conditional
##                 0.1891892                 0.5725594                 0.4929577
##   SelMeth.ICC_conditional InterpRes.ICC_conditional    VisOrg.ICC_conditional
##                 0.5212766                 0.2295720                 0.5924529
##    TxtOrg.ICC_conditional
##                 0.1428571
```

For the CritDes, InitEDA, SelMeth, and VisOrg rubrics, the correlation between is raters is average, at around 0.5 for all. For the RsrchQ, InterpRes, and TxtOrg rubrics, the correlation is very low at around 0.2.

For each of the 7 rubrics, make 3 two-way tables for cross-classification of ratings for each pair of raters:

```
#RsrchQ
RsrchQ_t12 <- table( "R1"=factor(rubric_all3$RsrchQ[rubric_all3$Rater==1], levels=1:4),
                     "R2"=factor(rubric_all3$RsrchQ[rubric_all3$Rater==2], levels=1:4) )
RsrchQ_t13 <- table( "R1"=factor(rubric_all3$RsrchQ[rubric_all3$Rater==1], levels=1:4),
                     "R2"=factor(rubric_all3$RsrchQ[rubric_all3$Rater==3], levels=1:4) )
RsrchQ_t23 <- table( "R2"=factor(rubric_all3$RsrchQ[rubric_all3$Rater==2], levels=1:4),
                     "R3"=factor(rubric_all3$RsrchQ[rubric_all3$Rater==3], levels=1:4) )

#CritDes
CritDes_t12 <- table( "R1"=factor(rubric_all3$CritDes[rubric_all3$Rater==1], levels=1:4),
                      "R2"=factor(rubric_all3$CritDes[rubric_all3$Rater==2], levels=1:4) )
CritDes_t13 <- table( "R1"=factor(rubric_all3$CritDes[rubric_all3$Rater==1], levels=1:4),
                      "R3"=factor(rubric_all3$CritDes[rubric_all3$Rater==3], levels=1:4) )
CritDes_t23 <- table( "R2"=factor(rubric_all3$CritDes[rubric_all3$Rater==2], levels=1:4),
                      "R3"=factor(rubric_all3$CritDes[rubric_all3$Rater==3], levels=1:4) )

#InitEDA
InitEDA_t12 <- table( "R1"=factor(rubric_all3$InitEDA[rubric_all3$Rater==1], levels=1:4),
                      "R2"=factor(rubric_all3$InitEDA[rubric_all3$Rater==2], levels=1:4) )
InitEDA_t13 <- table( "R1"=factor(rubric_all3$InitEDA[rubric_all3$Rater==1], levels=1:4),
                      "R3"=factor(rubric_all3$InitEDA[rubric_all3$Rater==3], levels=1:4) )
InitEDA_t23 <- table( "R2"=factor(rubric_all3$InitEDA[rubric_all3$Rater==2], levels=1:4),
                      "R3"=factor(rubric_all3$InitEDA[rubric_all3$Rater==3], levels=1:4) )
#SelMeth
SelMeth_t12 <- table( "R1"=factor(rubric_all3$SelMeth[rubric_all3$Rater==1], levels=1:4),
                      "R2"=factor(rubric_all3$SelMeth[rubric_all3$Rater==2], levels=1:4) )
SelMeth_t13 <- table( "R1"=factor(rubric_all3$SelMeth[rubric_all3$Rater==1], levels=1:4),
                      "R3"=factor(rubric_all3$SelMeth[rubric_all3$Rater==3], levels=1:4) )
SelMeth_t23 <- table( "R2"=factor(rubric_all3$SelMeth[rubric_all3$Rater==2], levels=1:4),
                      "R3"=factor(rubric_all3$SelMeth[rubric_all3$Rater==3], levels=1:4) )

#InterpRes
InterpRes_t12 <- table( "R1"=factor(rubric_all3$InterpRes[rubric_all3$Rater==1], levels=1:4),
                        "R2"=factor(rubric_all3$InterpRes[rubric_all3$Rater==2], levels=1:4) )
InterpRes_t13 <- table( "R1"=factor(rubric_all3$InterpRes[rubric_all3$Rater==1], levels=1:4),
```

```
                        "R3"=factor(rubric_all3$InterpRes[rubric_all3$Rater==3], levels=1:4) )
InterpRes_t23 <- table( "R2"=factor(rubric_all3$InterpRes[rubric_all3$Rater==2], levels=1:4),
                        "R3"=factor(rubric_all3$InterpRes[rubric_all3$Rater==3], levels=1:4) )

#VisOrg
VisOrg_t12 <- table( "R1"=factor(rubric_all3$VisOrg[rubric_all3$Rater==1], levels=1:4),
                     "R2"=factor(rubric_all3$VisOrg[rubric_all3$Rater==2], levels=1:4) )
VisOrg_t13 <- table( "R1"=factor(rubric_all3$VisOrg[rubric_all3$Rater==1], levels=1:4),
                     "R3"=factor(rubric_all3$VisOrg[rubric_all3$Rater==3], levels=1:4) )
VisOrg_t23 <- table( "R2"=factor(rubric_all3$VisOrg[rubric_all3$Rater==2], levels=1:4),
                     "R3"=factor(rubric_all3$VisOrg[rubric_all3$Rater==3], levels=1:4) )

#VisOrg
TxtOrg_t12 <- table( "R1"=factor(rubric_all3$TxtOrg[rubric_all3$Rater==1], levels=1:4),
                     "R2"=factor(rubric_all3$TxtOrg[rubric_all3$Rater==2], levels=1:4) )
TxtOrg_t13 <- table( "R1"=factor(rubric_all3$TxtOrg[rubric_all3$Rater==1], levels=1:4),
                     "R3"=factor(rubric_all3$TxtOrg[rubric_all3$Rater==3], levels=1:4) )
TxtOrg_t23 <- table( "R2"=factor(rubric_all3$TxtOrg[rubric_all3$Rater==2], levels=1:4),
                     "R3"=factor(rubric_all3$TxtOrg[rubric_all3$Rater==3], levels=1:4) )
```

21 agreement tables:

```
grid.arrange(
   tableGrob(RsrchQ_t12), tableGrob(RsrchQ_t13), tableGrob(RsrchQ_t23),
   tableGrob(CritDes_t12), tableGrob(CritDes_t13), tableGrob(CritDes_t23),
   tableGrob(InitEDA_t12), tableGrob(InitEDA_t13), tableGrob(InitEDA_t23),
   tableGrob(SelMeth_t12), tableGrob(SelMeth_t13), tableGrob(SelMeth_t23),
   tableGrob(InterpRes_t12), tableGrob(InterpRes_t13), tableGrob(InterpRes_t23),
   tableGrob(VisOrg_t12), tableGrob(VisOrg_t13), tableGrob(VisOrg_t23),
   tableGrob(TxtOrg_t12), tableGrob(TxtOrg_t13), tableGrob(TxtOrg_t23),
    nrow=7, ncol=3
)
```

	1	2	3	4
1	0	0	0	0
2	1	4	3	0
3	1	3	1	0
4	0	0	0	0

	1	2	3	4
1	0	0	0	0
2	0	7	1	0
3	0	2	3	0
4	0	0	0	0

	1	2	3	4
1	0	2	0	0
2	0	5	2	0
3	0	2	2	0
4	0	0	0	0

	1	2	3	4
1	3	2	1	0
2	2	3	1	0
3	0	0	1	0
4	0	0	0	0

	1	2	3	4
1	4	2	0	0
2	2	3	1	0
3	0	0	1	0
4	0	0	0	0

	1	2	3	4
1	5	0	0	0
2	1	3	1	0
3	0	2	1	0
4	0	0	0	0

	1	2	3	4
1	0	1	0	0
2	0	4	0	0
3	0	3	5	0
4	0	0	0	0

	1	2	3	4
1	0	1	0	0
2	0	4	0	0
3	0	5	3	0
4	0	0	0	0

	1	2	3	4
1	0	0	0	0
2	0	8	0	0
3	0	2	3	0
4	0	0	0	0

	1	2	3	4
1	0	0	0	0
2	1	10	0	0
3	0	0	2	0
4	0	0	0	0

	1	2	3	4
1	0	0	0	0
2	3	7	1	0
3	0	1	1	0
4	0	0	0	0

	1	2	3	4
1	1	0	0	0
2	2	7	1	0
3	0	1	1	0
4	0	0	0	0

	1	2	3	4
1	0	0	0	0
2	0	3	1	1
3	0	3	5	0
4	0	0	0	0

	1	2	3	4
1	0	0	0	0
2	1	3	1	0
3	0	4	4	0
4	0	0	0	0

	1	2	3	4
1	0	0	0	0
2	1	4	1	0
3	0	2	4	0
4	0	1	0	0

	1	2	3	4
1	1	0	0	0
2	0	4	5	0
3	0	1	2	0
4	0	0	0	0

	1	2	3	4
1	1	0	0	0
2	0	7	2	0
3	0	1	2	0
4	0	0	0	0

	1	2	3	4
1	1	0	0	0
2	0	5	0	0
3	0	3	4	0
4	0	0	0	0

	1	2	3	4
1	0	0	0	0
2	0	2	2	0
3	0	1	7	0
4	1	0	0	0

	1	2	3	4
1	0	0	0	0
2	1	1	2	0
3	0	1	7	0
4	0	1	0	0

	1	2	3	4
1	0	1	0	0
2	1	0	2	0
3	0	2	7	0
4	0	0	0	0

```
  #
```

Number of times raters disagree by 2 points: 5 Number of times raters disagree by 3 points: 1

Disagreement by 3 points:

```
rubric_all3[rubric_all3$Artifact=='O2', c(1,8,9)] %>% kable()
```

	Rater	TxtOrg	Artifact
29	3	2	O2
69	2	1	O2
108	1	4	O2

Find percent of times pairs of raters have exact agreement:

```
r1r2_percent_agree <- round(c(RsrchQ_t12 %>% diag%>%sum / RsrchQ_t12 %>% sum,
                          CritDes_t12 %>% diag%>%sum / CritDes_t12 %>% sum,
                          InitEDA_t12 %>% diag%>%sum / InitEDA_t12 %>% sum,
                          SelMeth_t12 %>% diag%>%sum / SelMeth_t12 %>% sum,
                          InterpRes_t12 %>% diag%>%sum / InterpRes_t12 %>% sum,
                          VisOrg_t12 %>% diag%>%sum / VisOrg_t12 %>% sum,
                          TxtOrg_t12 %>% diag%>%sum / TxtOrg_t12 %>% sum ), 3)

r1r3_percent_agree <- round(c(RsrchQ_t13 %>% diag%>%sum / RsrchQ_t13 %>% sum,
                          CritDes_t13 %>% diag%>%sum / CritDes_t13 %>% sum,
                          InitEDA_t13 %>% diag%>%sum / InitEDA_t13 %>% sum,
                          SelMeth_t13 %>% diag%>%sum / SelMeth_t13 %>% sum,
                          InterpRes_t13 %>% diag%>%sum / InterpRes_t13 %>% sum,
                          VisOrg_t13 %>% diag%>%sum / VisOrg_t13 %>% sum,
                          TxtOrg_t13 %>% diag%>%sum / TxtOrg_t13 %>% sum ), 3)

r2r3_percent_agree <- round(c(RsrchQ_t23 %>% diag%>%sum / RsrchQ_t23 %>% sum,
                          CritDes_t23 %>% diag%>%sum / CritDes_t23 %>% sum,
                          InitEDA_t23 %>% diag%>%sum / InitEDA_t23 %>% sum,
                          SelMeth_t23 %>% diag%>%sum / SelMeth_t23 %>% sum,
                          InterpRes_t23 %>% diag%>%sum / InterpRes_t23 %>% sum,
                          VisOrg_t23 %>% diag%>%sum / VisOrg_t23 %>% sum,
                          TxtOrg_t23 %>% diag%>%sum / TxtOrg_t23 %>% sum ), 3)

rater_percent_agree = data.frame("Rubric" = names(randint_models),
                                 "Raters 1 and 2 agreement" = r1r2_percent_agree,
                                 "Raters 1 and 3 agreement" = r1r3_percent_agree,
                                 "Raters 2 and 3 agreement" = r2r3_percent_agree)
```

Rater agreement for each rubric:

```
rater_percent_agree  %>%
  kable(caption = "Agreement between each pair of raters for each rubric") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 8: Agreement between each pair of raters for each rubric

Rubric	Raters.1.and.2.agreement	Raters.1.and.3.agreement	Raters.2.and.3.agreement
RsrchQ	0.385	0.769	0.538
CritDes	0.538	0.615	0.692
InitEDA	0.692	0.538	0.846
SelMeth	0.923	0.615	0.692
InterpRes	0.615	0.538	0.615
VisOrg	0.538	0.769	0.769
TxtOrg	0.692	0.615	0.538

Average rater agreement:

```
round(summarize_all(rater_percent_agree[,2:4], mean), 3) %>%
  kable(caption = "Average agreement between each pair of raters") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 9: Average agreement between each pair of raters

Raters.1.and.2.agreement	Raters.1.and.3.agreement	Raters.2.and.3.agreement
0.626	0.637	0.67

Each pair of raters all agree with each around 2/3 of the time. Raters 2 and 3 agree the most by a small margin.

Find random intercept models with all ratings, not just papers commonly rated by all 3 raters:

```
randint_models_all = list()
randint_models_all[[1]] <- lmer(as.numeric(RsrchQ) ~ 1 + (1 | Artifact), data = ratings)
randint_models_all[[2]] <- lmer(as.numeric(CritDes) ~ 1 + (1 | Artifact), data = ratings)
randint_models_all[[3]] <- lmer(as.numeric(InitEDA) ~ 1 + (1 | Artifact), data = ratings)
randint_models_all[[4]] <- lmer(as.numeric(SelMeth) ~ 1 + (1 | Artifact), data = ratings)
randint_models_all[[5]] <- lmer(as.numeric(InterpRes) ~ 1 + (1 | Artifact), data = ratings)
randint_models_all[[6]] <- lmer(as.numeric(VisOrg) ~ 1 + (1 | Artifact), data = ratings)
randint_models_all[[7]] <- lmer(as.numeric(TxtOrg) ~ 1 + (1 | Artifact), data = ratings)
names(randint_models_all) <- names(randint_models)
```

Find ICCs of above models:

```
unlist(lapply(randint_models_all, FUN = performance::icc))[seq(from=2, to=14, by=2)]
```

```
##    RsrchQ.ICC_conditional   CritDes.ICC_conditional   InitEDA.ICC_conditional
##                 0.2096214                 0.6730647                 0.6867210
##   SelMeth.ICC_conditional InterpRes.ICC_conditional    VisOrg.ICC_conditional
##                 0.4719014                 0.2200285                 0.6607372
##    TxtOrg.ICC_conditional
##                 0.1879927
```

Look at the ICCs of the two sets of models:

```
common_rated_ICCs <- round(unlist(lapply(randint_models,
                                         FUN = performance::icc))[seq(from=2, to=14, by=2)], 3)
all_rating_ICCs <- round(unlist(lapply(randint_models_all,
                                       FUN = performance::icc))[seq(from=2, to=14, by=2)], 3)
```

```
data.frame(common_rated_ICCs, all_rating_ICCs) %>%
  kable(caption = "Common correlation between raters for the
        commonly rated papers and for all papers,
        shown for each rubric") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 10: Common correlation between raters for the commonly rated papers and for all papers, shown for each rubric

	common_rated_ICCs	all_rating_ICCs
RsrchQ.ICC_conditional	0.189	0.210
CritDes.ICC_conditional	0.573	0.673
InitEDA.ICC_conditional	0.493	0.687
SelMeth.ICC_conditional	0.521	0.472
InterpRes.ICC_conditional	0.230	0.220
VisOrg.ICC_conditional	0.592	0.661
TxtOrg.ICC_conditional	0.143	0.188

The ICCs for all ratings are pretty close to the ICCs for only papers rated by all 3 raters. Most of the all-rating models have ICCs that are <= 0.1 bigger than the others. Only the SelMeth rubric has a smaller ICC and the InitEDA rubric an IDD almost 0.2 bigger. No single rater is disagreeing with the others more.

Question 3: More generally, how are the various factors in this experiment (Rater, Semester, Sex, Repeated, Rubric) related to the ratings? Do the factors interact in any interesting ways?

Try adding fixed effects to 7 intercept models based on only the 13 commonly rated papers. Add three fixed effects (rater, semester, and sex) to each of the 7 intercept models. Eliminate variables using fitLMER.fnc() function (Rater will always be left in this version of model). Then test whether Rater belongs in each rubric's model by removing it from the previous model and comparing using ANOVA. Either way, save each preferred model in a list of length 7 called *model.formula.13*.

```
rubric.names <- sort(unique(ratings_tall$Rubric))

model.formula.13 <- list()
#

for (i in rubric.names) {

  ## fit each base model
  rubric.data <- ratings_tall_all3[ratings_tall_all3$Rubric==i,]
  tmp <- lmer(as.numeric(Rating) ~ -1 + as.factor(Rater) +
              Semester + Sex + (1|Artifact),
            data=rubric.data,REML=FALSE)

  ## do backwards elimination. Rater will always be kept in the model due to
   #the intercept being removed in lmer above.
  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 FORMULA to list:
  model.formula.13[[i]] <- formula(tmp_final)

}
```

Look at 7 chosen models:

```
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)
```

For all 7 rubrics, no fixed effects are deemed important, not even Rater.

Now try adding fixed effects to 7 intercept models based on all data. As before, add three fixed effects (rater, semester, and sex) to each of the 7 intercept models. Eliminate variables using fitLMER.fnc() function (Rater will always be left in this version of model). Then test whether Rater belongs in each rubric's model by removing it from the previous model and comparing using ANOVA. Either way, save each preferred model in a list called *model.formula.alldata*.

```
rubric.names <- sort(unique(ratings_tall$Rubric))

# Remove 2 observations with missing ratings so that we use the same
 #data set for every model fit and model comparison:
ratings_tall[c(161,684),] ## Confirm from ealier code that these are
 #the rows with missing ratings.
ratings_tall_noNA <- ratings_tall[-c(161,684),]


#Remove observation with sex non M/F sex, to ease interpretation of model
 #if Sex variable is included:
```

```
ratings_tall_noNA[ratings_tall_noNA$Sex=="--",]  ## check which rows will be eliminated
ratings_tall_noNA <- ratings_tall_noNA[ratings_tall_noNA$Sex!="--",] ## remove Sex = '--' rows


model.formula.alldata <- list()
model.alldata <- list()


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

for (i in rubric.names) {

  ## fit each base model
  rubric.data <- ratings_tall_noNA[ratings_tall_noNA$Rubric==i,]
  tmp <- lmer(as.numeric(Rating) ~ -1 + as.factor(Rater) +
              Semester + Sex + Repeated + (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 FORMULA to the list:
  model.formula.alldata[[i]] <- formula(tmp_final)
  #Plus add model to a list:
  model.alldata[[i]] <- tmp_final
}
```

Look at 7 chosen models:

```
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
```

Three of the rubric's models (InitEDA, RsrchQ, TxtOrg) include none of the potential fixed effects, as with the models based on the reduced data. However, 3 rubric models (CritDes, InterpRes, VisOrg) include Rater this time, and 1 rubric model (SelMeth) includes Rater AND Semester as FEs.

Try adding interactions and new random effects for the 7 rubric-models fit using all the data.

The InitEDA, RsrchQ and SelMeth models only include the random-intercept for artifact. Since we only add random effects that are also present as fixed effects, there is nothing to try for these three.
The CritDes, InterpRes, and VisOrg models include Rater as a lone fixed effect, so we'll try adding it as a random effect as well.

```
#null hypotheses (no new RE):
CritDes_rater_tmp0 <- model.alldata[[1]]
InterpRes_rater_tmp0 <- model.alldata[[3]]
VisOrg_rater_tmp0 <- model.alldata[[7]]

#alternate hypothese (1 new RE: Rater|Artifact)
CritDes_rater_tmpA <- update(model.alldata[[1]], .~. + (as.factor(Rater) | Artifact))
InterpRes_rater_tmpA <- update(model.alldata[[3]], .~. + (as.factor(Rater) | Artifact))
VisOrg_rater_tmpA <- update(model.alldata[[7]], .~. + (as.factor(Rater) | Artifact))

#models with just new RE (Rater|Artifact):
CritDes_rater_tmpN <- update(CritDes_rater_tmpA, .~. - (1 | Artifact))
InterpRes_rater_tmpN <- update(InterpRes_rater_tmpA, .~. - (1 | Artifact))
VisOrg_rater_tmpN <- update(VisOrg_rater_tmpA, .~. - (1 | Artifact))
```

Attempting to fit any of these models with the new RE (Rater|Artifact) results in a 'number of observations <= number of random effects' error, so testing for the new RE is not possible.

Now let's try to test new interactions and REs in the final rubric model for SelMeth. There are only two FE in the model, Rater and Semester, so we'll try adding their interaction:

```
#SelMeth
#original:
SelMeth_rater <- lmer(as.numeric(Rating) ~ as.factor(Rater) + Semester + (1 | Artifact) -
    1, data = ratings_tall_noNA[ratings_tall_noNA$Rubric=='SelMeth',])
#new interaction
SelMeth_rater_int <- lmer(as.numeric(Rating) ~ as.factor(Rater) + Semester + (1 | Artifact) -
    1 + as.factor(Rater)*Semester - Semester, data =
      ratings_tall_noNA[ratings_tall_noNA$Rubric=='SelMeth',])
#Specify the model to show a different intercept for each
 #rater as before, as well as a different semester effect for each rater.
anova(SelMeth_rater, SelMeth_rater_int)
```

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

```
## Data: ratings_tall_noNA[ratings_tall_noNA$Rubric == "SelMeth", ]
## Models:
## SelMeth_rater: as.numeric(Rating) ~ as.factor(Rater) + Semester + (1 | Artifact) - 1
```

```
## SelMeth_rater_int: as.numeric(Rating) ~ as.factor(Rater) + Semester + (1 | Artifact) - 1 + as.factor
##                    npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## SelMeth_rater         6 142.05 158.58 -65.027   130.05
## SelMeth_rater_int     8 143.46 165.49 -63.731   127.46 2.592  2     0.2736
```

Based on the ANOVA test, the new interaction, Rater:Semester, does not improve the model.

Now try adding new random effects to the SelMeth model, if possible. Start by trying Semester as a new RE

```
#null hypotheses (no new RE):
SelMeth_rater_tmp0 <- model.alldata[[5]]

#alternate hypothese (1 new RE: Rater|Artifact)
SelMeth_rater_tmpA <- update(model.alldata[[5]], .~. + (Semester | Artifact))

#models with just new RE (Rater|Artifact):
SelMeth_rater_tmpN <- update(SelMeth_rater_tmpA, .~. - (1 | Artifact))
```

Once again, the attempt to add new REs results in a 'number of observations <= number of random effects' error, so the test is not possible. We saw with the attempts to add a new Rater RE to the models for CritDes, InterpRes, and VisOrg, that such an attempt would also cause the same error, since the Rater variable has even more levels than Semester.

Overall, no new random effects or new fixed effect interactions could be reasonably added to the seven rubric-specific models on the whole data set.

Try adding fixed effects, interactions, and new random effects to the combined model with only a random intercept for each Rubric depending on Artifact.

Intercept-only model on all data:

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

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

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

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ 1 + (0 + Rubric | Artifact)
##    Data: ratings_tall_noNA
##
## 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
```

```
#coef(comb.0)
```

Try adding all possible FEs to the intercept-only model and then running variable selection with fitLMER. fitLMER just does by backward elimination on fixed effects, since no aditional random effects are tested:

```
comb.full <- update(comb.0, . ~ . + as.factor(Rater) + Semester +
                      Sex + Repeated + Rubric)
#summary(comb.full)
comb.back_elim <- fitLMER.fnc(comb.full, log.file.name = FALSE)
```

Check resulting model:

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

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Rubric
##    Data: ratings_tall_noNA
##
## 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
```

Based on the T-tests performed by the fitLMER.fnc function, variables Sex and Repeated are determined to be unnecessary.

Try adding FE interactions, including 3-way and 2-way interactions between the variables Rater, Semester, and Rubric:

```
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)
```

```
#fit produces warning. Try different optimizer/more iterations:

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

#summary(comb.inter.u)
```

Now run variable selection on the model with FE interactions:

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

View model:

```
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: ratings_tall_noNA
## Control: lmerControl(optimizer = "Nelder_Mead", optCtrl = list(maxfun = 2e+05))
##
## REML criterion at convergence: 1419.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9187 -0.5122 -0.0439  0.4820  3.5875
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.50273  0.7090
```

```
##           RubricInitEDA   0.35392  0.5949   0.45
##           RubricInterpRes 0.15244  0.3904   0.35 0.81
##           RubricRsrchQ    0.17964  0.4238   0.63 0.44 0.72
##           RubricSelMeth   0.06729  0.2594   0.42 0.60 0.74 0.36
##           RubricTxtOrg    0.26145  0.5113   0.42 0.64 0.67 0.55 0.63
##           RubricVisOrg    0.25549  0.5055   0.34 0.71 0.67 0.51 0.38 0.77
##  Residual                 0.18501  0.4301
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
##                                  Estimate Std. Error t value
## (Intercept)                       1.75956    0.11779  14.939
## as.factor(Rater)2                 0.36533    0.13290   2.749
## as.factor(Rater)3                 0.21397    0.13291   1.610
## SemesterS19                      -0.17781    0.08226  -2.162
## RubricInitEDA                     0.74601    0.13663   5.460
## RubricInterpRes                   1.01436    0.13483   7.523
## RubricRsrchQ                      0.74884    0.12424   6.028
## RubricSelMeth                     0.42655    0.13038   3.272
## RubricTxtOrg                      1.04956    0.13551   7.745
## RubricVisOrg                      0.68355    0.13943   4.902
## as.factor(Rater)2:RubricInitEDA  -0.30822    0.17235  -1.788
## as.factor(Rater)3:RubricInitEDA  -0.29485    0.17268  -1.707
## as.factor(Rater)2:RubricInterpRes -0.53661    0.17010  -3.155
## as.factor(Rater)3:RubricInterpRes -0.75212    0.17051  -4.411
## as.factor(Rater)2:RubricRsrchQ   -0.50122    0.16153  -3.103
## as.factor(Rater)3:RubricRsrchQ   -0.36993    0.16181  -2.286
## as.factor(Rater)2:RubricSelMeth  -0.39586    0.16464  -2.404
## as.factor(Rater)3:RubricSelMeth  -0.41292    0.16500  -2.502
## as.factor(Rater)2:RubricTxtOrg   -0.58390    0.17140  -3.407
## as.factor(Rater)3:RubricTxtOrg   -0.48627    0.17176  -2.831
## as.factor(Rater)2:RubricVisOrg   -0.14452    0.17437  -0.829
## as.factor(Rater)3:RubricVisOrg   -0.33347    0.17476  -1.908
##
## Correlation matrix not shown by default, as p = 22 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.039115 (tol = 0.002, component 1)
```

The only interaction kept in the model is that between Rater and Rubric.

Compare the three combined models fitted so far by their formulas:

```
cat("All possible FEs:\n")
```

```
## All possible FEs:
```

```
formula(comb.full)
```

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

```
cat("\nAbove model after variable selection:\n")
```

```
##
```

```
## Above model after variable selection:
```

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

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

```
cat("\nAll possible interactions between above model FEs added in:\n")
```

```
##
## All possible interactions between above model FEs added in:
```

```
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
```

```
cat("\nAbove model after variable selection:\n")
```

```
##
## Above model after variable selection:
```

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

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

Compare the three combined models fitted so far by the correlation between predictors:

```
cat("All possible FEs:\n")
```

```
## All possible FEs:
```

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

```
##  Groups   Name            Std.Dev. Corr
##  Artifact RubricCritDes   0.74372
##           RubricInitEDA   0.59362  0.466
##           RubricInterpRes 0.41847  0.232 0.750
##           RubricRsrchQ    0.41227  0.585 0.440 0.710
##           RubricSelMeth   0.26108  0.389 0.602 0.744 0.406
##           RubricTxtOrg    0.51321  0.338 0.618 0.702 0.563 0.671
##           RubricVisOrg    0.50803  0.347 0.732 0.678 0.516 0.411 0.756
##  Residual                 0.43492
```

```
cat("\nAbove model after variable selection:\n")
```

```
##
## Above model after variable selection:
```

```
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
```

```
cat("\nAll possible interactions between above model FEs added in:\n")
```

```
##
## All possible interactions between above model FEs added in:
```

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

```
##  Groups   Name            Std.Dev. Corr
##  Artifact RubricCritDes   0.69675
##           RubricInitEDA   0.59376  0.416
##           RubricInterpRes 0.38236  0.324 0.800
##           RubricRsrchQ    0.40550  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    0.50523  0.349 0.727 0.675 0.567 0.346 0.757
##  Residual                 0.43405
```

```
cat("\nAbove model after variable selection:\n")
```

```
##
## Above model after variable selection:
```

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

```
##  Groups   Name            Std.Dev. Corr
##  Artifact RubricCritDes   0.70903
##           RubricInitEDA   0.59491  0.445
##           RubricInterpRes 0.39044  0.352 0.814
##           RubricRsrchQ    0.42384  0.629 0.440 0.715
##           RubricSelMeth   0.25941  0.422 0.601 0.736 0.361
##           RubricTxtOrg    0.51132  0.416 0.636 0.669 0.547 0.634
##           RubricVisOrg    0.50546  0.339 0.715 0.674 0.513 0.376 0.770
##  Residual                 0.43013
```

Now compare the models using ANOVA and info criteria (not the original *comb.full*, because it is not nested within the others):

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

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

```
## Data: ratings_tall_noNA
## 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 model *comb.inter_elim* with one interaction, Rater:Rubric, is preferred by the F-test and by AIC.

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

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

```
##     Semester + Rubric + as.factor(Rater):Rubric
```

This model suggests Rating is affected by the Rater who graded the project, the Semester it was assigned, and the Rubric being graded, but that the Rubric affects the grade differently depending on Rater.

Look at coefficients for model FEs to see this varying effect :

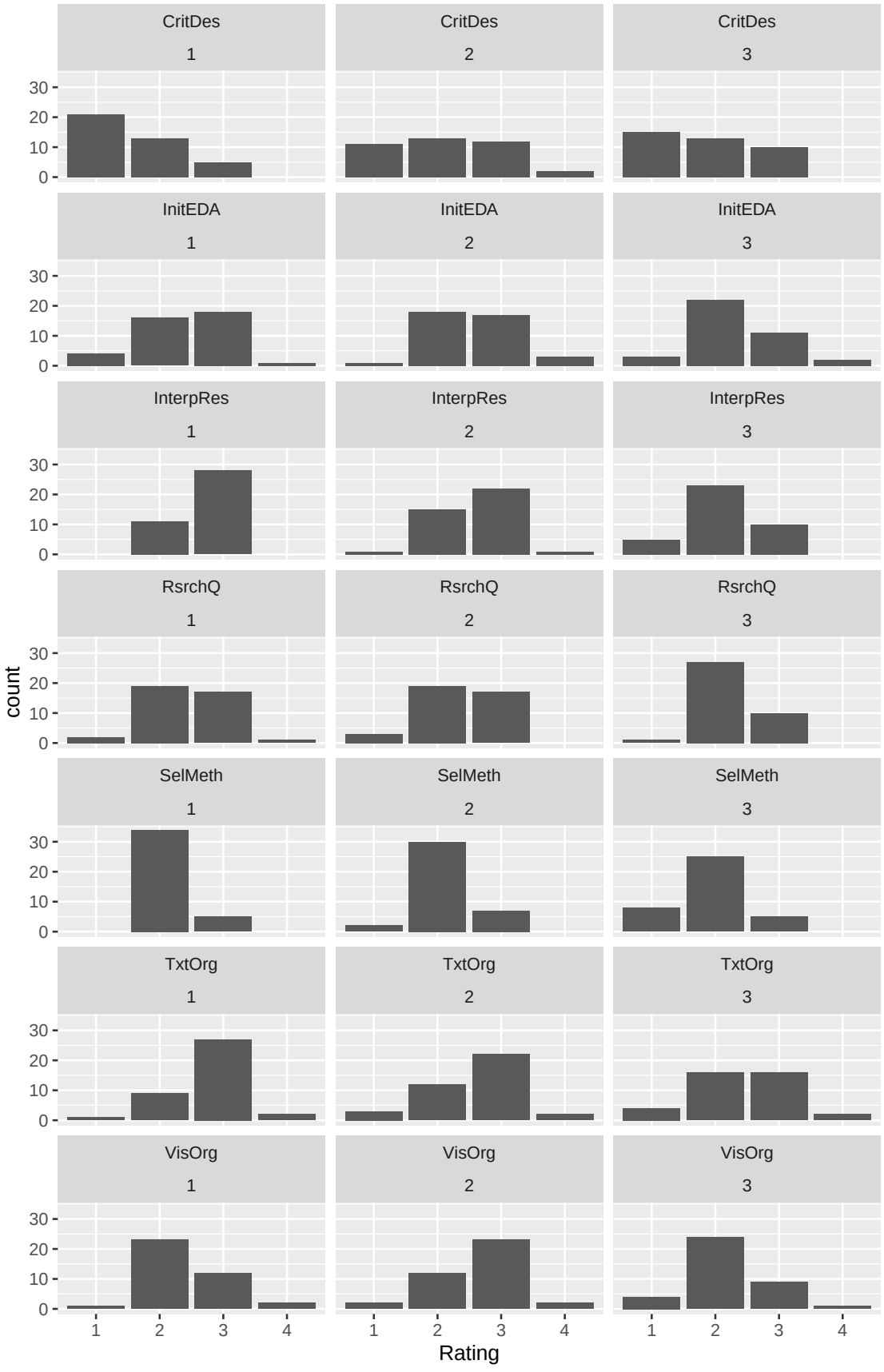
```
summary(comb.inter_elim)$coef
```

```
##                                    Estimate Std. Error    t value
## (Intercept)                       1.7595626 0.11778520 14.9387405
## as.factor(Rater)2                 0.3653298 0.13289753  2.7489585
## as.factor(Rater)3                 0.2139686 0.13291323  1.6098368
## SemesterS19                      -0.1778096 0.08225811 -2.1616056
## RubricInitEDA                     0.7460134 0.13662956  5.4601174
## RubricInterpRes                   1.0143629 0.13482598  7.5234971
## RubricRsrchQ                      0.7488442 0.12423680  6.0275554
## RubricSelMeth                     0.4265498 0.13038072  3.2715714
## RubricTxtOrg                      1.0495614 0.13551294  7.7451008
## RubricVisOrg                      0.6835512 0.13943106  4.9024310
## as.factor(Rater)2:RubricInitEDA  -0.3082206 0.17235495 -1.7882900
## as.factor(Rater)3:RubricInitEDA  -0.2948486 0.17268392 -1.7074467
## as.factor(Rater)2:RubricInterpRes -0.5366147 0.17009971 -3.1547068
## as.factor(Rater)3:RubricInterpRes -0.7521200 0.17050700 -4.4110799
## as.factor(Rater)2:RubricRsrchQ   -0.5012240 0.16152526 -3.1030688
## as.factor(Rater)3:RubricRsrchQ   -0.3699310 0.16181075 -2.2861953
## as.factor(Rater)2:RubricSelMeth  -0.3958571 0.16463537 -2.4044472
## as.factor(Rater)3:RubricSelMeth  -0.4129206 0.16500464 -2.5024787
## as.factor(Rater)2:RubricTxtOrg   -0.5838997 0.17139667 -3.4067157
## as.factor(Rater)3:RubricTxtOrg   -0.4862692 0.17175987 -2.8310989
## as.factor(Rater)2:RubricVisOrg   -0.1445162 0.17436925 -0.8287944
## as.factor(Rater)3:RubricVisOrg   -0.3334744 0.17475568 -1.9082321
```

There are a range of interaction coefficients that show the different in Rater's use of Rubrics. For example, Rater 2 tends to rate higher based on their coefficient alone, but Rater 2 rates the lowest for TxtOrg.

Also check for patterns using the plots :

```
ggplot(ratings_tall_noNA, aes(x=Rating)) +
geom_bar() + facet_wrap( ~ Rubric + Rater, nrow=7)
```


CritDes
1
CritDes
2
CritDes
3
InitEDA
1
InitEDA
2
InitEDA
3
InterpRes
1
InterpRes
2
InterpRes
3
RsrchQ
1
RsrchQ
2
RsrchQ
3
SelMeth
1
SelMeth
2
SelMeth
3
TxtOrg
1
TxtOrg
2
TxtOrg
3
VisOrg
1
VisOrg
2
VisOrg
3
count
0
10
20
30
1
2
3
4
Rating

These plots show how Raters' ratings for certain rubrics differ from each other.

Now try adding additional random effects to the model. THere are 3 fixed effects that can be tried as random effects: Rater, Semester, and the Rater:Rubric interaction.
First try adding Rater as a RE:

```
comb.inter_elim_RE1 <- lmer(as.numeric(Rating) ~ (0 + Rubric | Artifact) +
(0 + as.factor(Rater) | Artifact) + as.factor(Rater) +
Semester + Rubric + as.factor(Rater):Rubric, data = ratings_tall_noNA)
```

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

```
anova(comb.inter_elim, comb.inter_elim_RE1)
```

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

```
## Data: ratings_tall_noNA
## Models:
## comb.inter_elim: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Rubric
## comb.inter_elim_RE1: as.numeric(Rating) ~ (0 + Rubric | Artifact) + (0 + as.factor(Rater) | Artifact)
##                      npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## comb.inter_elim        51 1454.5 1694.1 -676.26   1352.5
## comb.inter_elim_RE1    57 1415.9 1683.6 -650.94   1301.9 50.647  6  3.487e-09
##
## comb.inter_elim
## comb.inter_elim_RE1 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA test, as well as AIC/BIC both suggest including this new random effect for Rater in the model.

Now try adding Semester as a RE:

```
comb.inter_elim_RE2 <- lmer(as.numeric(Rating) ~ (0 + Rubric | Artifact) +
(0 + as.factor(Rater) | Artifact) +
(0 + Semester | Artifact) + as.factor(Rater) +
Semester + Rubric + as.factor(Rater):Rubric, data = ratings_tall_noNA)
```

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

```
anova(comb.inter_elim_RE1, comb.inter_elim_RE2)
```

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

```
## Data: ratings_tall_noNA
## Models:
## comb.inter_elim_RE1: as.numeric(Rating) ~ (0 + Rubric | Artifact) + (0 + as.factor(Rater) | Artifact)
## comb.inter_elim_RE2: as.numeric(Rating) ~ (0 + Rubric | Artifact) + (0 + as.factor(Rater) | Artifact)
##                      npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## comb.inter_elim_RE1    57 1415.9 1683.6 -650.94   1301.9
## comb.inter_elim_RE2    60 1421.6 1703.4 -650.81   1301.6 0.252  3     0.9688
```

Neither the test or AIC/BIC want the new random effect for Semester in the model.

Now try adding the Rater:Rubric interaction as a RE:

```
comb.inter_elim_RE3 <- 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 = ratings_tall_noNA)
anova(comb.inter_elim_RE1, comb.inter_elim_RE3)
```

This causes an error as there are not enough observations in the data for the number of REs we are trying to add to the model.

So, the final model will include one additional random effect (Rater), as well as the fixed effect interaction betweeen Rater and Rubric. The model is:

```
formula(comb.inter_elim_RE1)
```

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

Summary of the model:

```
summary(comb.inter_elim_RE1)  #sigma^2 = 0.13468, rubric: , tau^2 = ; rater: tau^2 =
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## as.numeric(Rating) ~ (0 + Rubric | Artifact) + (0 + as.factor(Rater) |
##     Artifact) + as.factor(Rater) + Semester + Rubric + as.factor(Rater):Rubric
##    Data: ratings_tall_noNA
##
## REML criterion at convergence: 1370.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.06428 -0.46900 -0.02983  0.45341  2.74000
##
## Random effects:
##  Groups     Name              Variance Std.Dev. Corr
##  Artifact   RubricCritDes     0.49642  0.7046
##             RubricInitEDA     0.31786  0.5638    0.32
##             RubricInterpRes   0.10206  0.3195    0.14  0.67
##             RubricRsrchQ      0.17899  0.4231    0.50  0.19  0.54
##             RubricSelMeth     0.03824  0.1956    0.14  0.23  0.38 -0.24
##             RubricTxtOrg      0.25028  0.5003    0.27  0.44  0.36  0.31  0.21
##             RubricVisOrg      0.23234  0.4820    0.18  0.50  0.45  0.28 -0.16
##  Artifact.1 as.factor(Rater)1 0.01281  0.1132
##             as.factor(Rater)2 0.11175  0.3343   -0.49
##             as.factor(Rater)3 0.09414  0.3068    0.33  0.66
##  Residual                     0.13468  0.3670
##
##
##
##
##
##
##
##
##   0.54
##
##
##
##
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
##                                   Estimate Std. Error t value
```

```
## (Intercept)                          1.75755    0.11404  15.412
## as.factor(Rater)2                    0.36606    0.13918   2.630
## as.factor(Rater)3                    0.19591    0.12967   1.511
## SemesterS19                         -0.15917    0.07647  -2.081
## RubricInitEDA                        0.73950    0.12996   5.690
## RubricInterpRes                      0.99152    0.12771   7.764
## RubricRsrchQ                         0.72619    0.11793   6.158
## RubricSelMeth                        0.41068    0.12470   3.293
## RubricTxtOrg                         1.01578    0.13000   7.814
## RubricVisOrg                         0.65425    0.13353   4.900
## as.factor(Rater)2:RubricInitEDA     -0.29981    0.15609  -1.921
## as.factor(Rater)3:RubricInitEDA     -0.29473    0.15635  -1.885
## as.factor(Rater)2:RubricInterpRes   -0.51324    0.15348  -3.344
## as.factor(Rater)3:RubricInterpRes   -0.71484    0.15364  -4.653
## as.factor(Rater)2:RubricRsrchQ      -0.48741    0.14722  -3.311
## as.factor(Rater)3:RubricRsrchQ      -0.32238    0.14727  -2.189
## as.factor(Rater)2:RubricSelMeth     -0.38638    0.15031  -2.571
## as.factor(Rater)3:RubricSelMeth     -0.38716    0.14961  -2.588
## as.factor(Rater)2:RubricTxtOrg      -0.55105    0.15646  -3.522
## as.factor(Rater)3:RubricTxtOrg      -0.44488    0.15673  -2.839
## as.factor(Rater)2:RubricVisOrg      -0.10490    0.15861  -0.661
## as.factor(Rater)3:RubricVisOrg      -0.27521    0.15885  -1.733
##
## Correlation matrix not shown by default, as p = 22 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

The model FE coefficients are:

```
summary(comb.inter_elim_RE1)$coef  %>%
  kable(caption = "Coefficients in final model", digits = 2) %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 11: Coefficients in final model

	Estimate	Std. Error	t value
(Intercept)	1.76	0.11	15.41
as.factor(Rater)2	0.37	0.14	2.63
as.factor(Rater)3	0.20	0.13	1.51
SemesterS19	-0.16	0.08	-2.08
RubricInitEDA	0.74	0.13	5.69
RubricInterpRes	0.99	0.13	7.76
RubricRsrchQ	0.73	0.12	6.16
RubricSelMeth	0.41	0.12	3.29
RubricTxtOrg	1.02	0.13	7.81
RubricVisOrg	0.65	0.13	4.90
as.factor(Rater)2:RubricInitEDA	-0.30	0.16	-1.92
as.factor(Rater)3:RubricInitEDA	-0.29	0.16	-1.89
as.factor(Rater)2:RubricInterpRes	-0.51	0.15	-3.34
as.factor(Rater)3:RubricInterpRes	-0.71	0.15	-4.65
as.factor(Rater)2:RubricRsrchQ	-0.49	0.15	-3.31
as.factor(Rater)3:RubricRsrchQ	-0.32	0.15	-2.19
as.factor(Rater)2:RubricSelMeth	-0.39	0.15	-2.57
as.factor(Rater)3:RubricSelMeth	-0.39	0.15	-2.59
as.factor(Rater)2:RubricTxtOrg	-0.55	0.16	-3.52
as.factor(Rater)3:RubricTxtOrg	-0.44	0.16	-2.84
as.factor(Rater)2:RubricVisOrg	-0.10	0.16	-0.66
as.factor(Rater)3:RubricVisOrg	-0.28	0.16	-1.73

Some of the RE coefficient values are shown (too many to list):

```
ranef(comb.inter_elim_RE1)[1:10]
```

```
## $Artifact
##     RubricCritDes RubricInitEDA RubricInterpRes RubricRsrchQ RubricSelMeth
## 100   0.799254427  -0.261024838    -0.121652629 -0.165076057   0.232376229
## 101  -0.496487528   0.434224814    -0.166227792 -0.741399207   0.032277197
## 102  -0.770637636  -0.332737858    -0.232329893 -0.744249801   0.048851503
## 103   0.139669409   0.330198946     0.098810078 -0.281592140   0.271543098
## 104  -0.576639438   0.308527575     0.091189508 -0.260811468   0.073418658
## 105  -0.590070680  -0.482686330    -0.340032335 -0.404172183  -0.097453016
## 106   0.204327384  -1.028883953    -0.399125707  0.233398883  -0.136307599
## 107  -0.559957851  -0.401608649     0.057206795  0.183134156  -0.102336429
## 111  -0.461603073  -0.351055688    -0.241463797 -0.311642336  -0.066937777
## 112  -0.499291170   0.322602196     0.271596738  0.197920492  -0.103867697
## 113  -0.586333414  -0.804706394    -0.145286443 -0.131179242   0.004220762
## 114  -0.448171831   0.440158218     0.189758047 -0.168281621   0.103933897
## 115  -0.370823564   0.454232839     0.370165276  0.290450339  -0.073352458
## 116  -0.588696034  -0.354273530    -0.287550231 -0.351685885  -0.123688575
## 117  -0.672337208  -0.136981915     0.011788252 -0.194733999  -0.104365858
## 118  -0.654134340  -0.212126192    -0.029465039 -0.208922753  -0.027825745
## 13    0.388204535  -0.746415810    -0.434670181  0.050610449  -0.158694230
## 15    0.688599423   0.476194361    -0.046966797 -0.110359209  -0.052055836
## 16    0.682592162   1.153695841     0.314120896 -0.008298752   0.186414900
## 17    0.335722581  -0.087916027     0.067039563  0.549517881  -0.199138172
## 21    0.775959601   1.090388721     0.451599362  0.434671629   0.085149958
```

```
## 22    0.730597615   0.381859393    0.248952158  0.430908625   0.090316590
## 23   -0.272029381  -0.723561310   -0.327004020 -0.010003856  -0.176380289
## 24    0.049051941  -0.202392191   -0.150283279 -0.130267442   0.068983982
## 25    1.160386392   0.007484406   -0.288600146  0.162946839  -0.130358471
## 26   -0.863325838  -0.484028339   -0.160023193 -0.518470516   0.112870050
## 27    0.101621683   0.386518050    0.006466788 -0.172809189   0.092091510
## 28   -0.298810862  -0.592028804   -0.413393496 -0.405015156  -0.059299014
## 32    0.660676549   0.404012104    0.250523950  0.407846922   0.001491722
## 33   -0.083936224   0.150252790   -0.059850296 -0.452438878   0.169479969
## 34    0.465688484  -0.311783603   -0.060504469 -0.201190616   0.261678796
## 35   -0.743816350  -0.254851355   -0.085370671 -0.283309904  -0.098077654
## 36    0.049051941  -0.202392191   -0.150283279 -0.130267442   0.068983982
## 37    0.775871813  -0.157021304   -0.206860449 -0.021694056   0.102486373
## 38   -0.016996478  -0.209480468   -0.141947841 -0.174736518  -0.064575263
## 39   -0.527958103   0.248599055    0.207286071  0.140180432  -0.087410199
## 40    0.105494330   0.357277062    0.013230435 -0.194216563   0.047357133
## 45    0.014660685  -0.241819859   -0.172285657 -0.089837830   0.049763027
## 46    0.149525211   0.324259400   -0.013546205 -0.141173524   0.032052349
## 47    1.130913925  -0.177627220   -0.049065979  0.677169620  -0.148596800
## 48    0.536956446   0.660643346    0.224703742  0.134774160   0.241950475
## 49   -0.903859261   0.215306545    0.273251512  0.159455304  -0.189834377
## 53    1.141774702   0.106548956    0.017115942  0.238983257   0.117982413
## 54   -0.662444940   0.383824093   -0.091935077 -0.662602135   0.147420999
## 55   -0.148685190  -0.372320725    0.070586894  0.317360372  -0.133655360
## 56    0.687791192  -0.264816882   -0.275989459 -0.060702569  -0.062069789
## 57   -0.769717846  -0.326314233   -0.169596192 -0.256439553  -0.084339398
## 6    -0.673895284  -0.277004067   -0.086942463 -0.260248201  -0.009252785
## 61    0.001538814   0.332842683   -0.079455720 -0.172069062   0.015992936
## 62    1.385202498   0.997741439    0.303261700  0.482991511  -0.052910156
## 63    0.682867532   0.348616818    0.206990349  0.445159596  -0.019000749
## 64    0.550893544  -0.162679319   -0.076524733  0.054788067   0.062473378
## 65    0.831967874  -0.452051603   -0.411612839  0.253164656  -0.217018183
## 66    0.741010208   0.910028221    0.371808631  0.382435120  -0.040223724
## 67   -0.816168926   0.144942635    0.292898045  0.146922110  -0.188097241
## 68    0.630981070  -0.239593320    0.050764537  0.488193481  -0.041945976
## 7    -0.621325541   0.311906175    0.069807605 -0.302789949   0.013854742
## 72   -0.056294130   0.416364668    0.192049401 -0.072221463   0.022353775
## 73   -0.746426401  -0.931290811   -0.323360413 -0.186557681  -0.017211085
## 74   -1.048639783   0.086000721    0.117160731 -0.493825375   0.092899126
## 75   -0.041412284   0.337817601    0.148248009 -0.088802210   0.098105036
## 76    0.037342687  -0.325731365   -0.216034999 -0.141568370  -0.005215155
## 77   -0.168476653  -0.233626097   -0.010439393 -0.072616310  -0.014913729
## 78    0.416148290   0.062400913    0.074608810  0.131269470   0.084763507
## 79   -0.309478122   0.018252870    0.132101679  0.023555683   0.051544602
## 8    -0.607309327  -0.208778680   -0.035853472 -0.212289120   0.006563547
## 84    0.308445851  -0.084163650    0.160689154  0.445180654  -0.061765352
## 85    1.073382862   0.376335653    0.315614108  0.876524405  -0.132031144
## 86    0.326648719  -0.159307927    0.119435862  0.430991900   0.014774761
## 87    0.204327384  -1.028883953   -0.399125707  0.233398883  -0.136307599
## 88    1.088096465   0.644180229    0.316282000  0.538699798  -0.077309776
## 9    -0.580527846  -0.340311186    0.050536003  0.182722180  -0.110517727
## 92   -0.540380336  -0.348340126    0.068435608  0.221431701  -0.052031876
## 93   -0.392335215  -0.163440961    0.178232959  0.352259092   0.028787916
## 94    1.037123761   1.033203354    0.338368009  0.394281236  -0.006580611
```

```
## 95    0.139669409   0.330198946     0.098810078 -0.281592140   0.271543098
## 96    0.239270278   0.380003753     0.224028454  0.314950845  -0.067576299
## 01   -0.501833135   0.422492400     0.120678820 -0.008547298  -0.092444312
## 010  -0.441620718  -0.001273012     0.155500759  0.049620157   0.036660616
## 011  -0.767149032  -0.330048966     0.328469563  0.512570342   0.013735953
## 012  -0.354418918  -0.488343083    -0.316167902 -0.338352508  -0.015400491
## 013   0.030664989   0.083217052    -0.316869316 -0.367351944  -0.071062643
## 02   -0.115962217   0.329821712     0.171346258  0.140222882  -0.072079827
## 03    0.283484440  -0.135250509    -0.012995229  0.231072663  -0.039719106
## 04   -0.111608085   0.044715167     0.203176133 -0.216903187   0.365019800
## 05    0.878801403  -0.426327177    -0.026091150  0.189637291   0.249949381
## 06   -0.897730628  -0.773057517    -0.419215540 -0.412698862  -0.112079073
## 07    0.137946203   0.206775492    -0.005709495 -0.013640063  -0.042674422
## 08    0.358956229  -0.209366221    -0.396311268 -0.311054388   0.029106551
## 09   -0.799406126   0.585057999     0.349324920 -0.190664552   0.074764753
##      RubricTxtOrg RubricVisOrg as.factor(Rater)1 as.factor(Rater)2
## 100 -0.4896793592  -0.44086319       0.034936568      -0.050188783
## 101  0.2892691284   0.46522893      -0.031032144       0.044579810
## 102 -1.1194088528  -0.43426972      -0.071722885       0.103034859
## 103  0.1091043260  -0.23982951       0.041764049      -0.059996930
## 104  0.0889463482  -0.11665635      -0.025624712       0.036811663
## 105 -0.5321703335  -0.35606582      -0.059281355       0.085161745
## 106  0.0172554445  -0.33465937      -0.022656943       0.032548258
## 107 -0.5131733071  -0.30985920      -0.011087353       0.015927745
## 111 -0.4110812031  -0.25279054      -0.035974074       0.051679232
## 112  0.2084053061   0.41592672       0.019756175      -0.028381105
## 113 -0.4728814134  -0.30644203      -0.016213573       0.023291914
## 114  0.2100354786  -0.01338107      -0.002317431       0.003329149
## 115  0.3294944365   0.51920199       0.043063456      -0.061863618
## 116  0.1298393248   0.27762843      -0.030969730       0.044490148
## 117  0.2258387455   0.84088523       0.010801963      -0.015517763
## 118  0.1276080694   0.31606861      -0.008677326       0.012465576
## 13  -0.7208355472  -0.62615682      -0.065719175      -0.387555964
## 15   0.4109255876   0.90084408       0.013941355       0.082214292
## 16   0.8983321370   0.58517698       0.020551785       0.121197001
## 17  -0.1152932168   0.03066764      -0.017999182      -0.106143914
## 21   0.9175021411   0.59118795       0.043496254       0.256504023
## 22   0.2103492050  -0.12229450       0.018247527       0.107608441
## 23  -0.6709941883  -0.56004326      -0.056913772      -0.335629164
## 24   0.2458067915  -0.13973909      -0.007466290      -0.044029849
## 25  -0.0007623467   0.07474060      -0.038408438      -0.226500397
## 26  -0.4701947997  -0.47160185       0.015712610       0.092659652
## 27   0.2990395462  -0.05305148      -0.006285748      -0.037068011
## 28  -0.6274025691  -0.51252569      -0.068990828      -0.406849402
## 32   0.2598320935   0.36094579       0.027330074       0.161169600
## 33  -0.4040417806  -0.39749927       0.018955163       0.111781475
## 34  -0.5023732572  -0.50591233       0.030230838       0.178275849
## 35  -0.2593062540   0.26606022      -0.004563978      -0.026914474
## 36   0.2458067915  -0.13973909      -0.007466290      -0.044029849
## 37   0.2587270560  -0.15232412      -0.005404281      -0.031869865
## 38  -0.2463859896   0.25347519      -0.002501969      -0.014754490
## 39  -0.2363863837  -0.12448150       0.010478486       0.061793227
## 40  -0.2426361233  -0.14307750      -0.010403973      -0.061353811
## 45   0.2993599871  -0.21570738       0.013972781      -0.084828503
```

```
## 46  -0.1862331518  -0.21911100       0.017905584      -0.108704483
## 47  -0.6837413431  -0.67097263       0.060711050      -0.368575702
## 48   0.6088839750  -0.35396522      -0.057928838       0.351684939
## 49   0.2597538164   0.72498630      -0.028115605       0.170689334
## 53   0.0731895763  -0.06274354      -0.066382216       0.403005249
## 54   0.3347476834  -0.11279192       0.048111314      -0.292082929
## 55  -0.3649965249   0.05817936      -0.010301114       0.062537877
## 56  -0.2393315344   0.11174846       0.019211956      -0.116635439
## 57   0.2765067671   0.22694723       0.019220334      -0.116686303
## 6   -0.3087891425  -0.21718008      -0.013646525      -0.080475634
## 61   0.8803198375   0.38765095       0.008576186      -0.052065873
## 62   0.4045615711   0.81896334      -0.054483093       0.330765888
## 63   0.2956495312   0.26735516      -0.036660224       0.222563567
## 64   0.2364079963   0.18588162      -0.001773938       0.010769546
## 65   0.3136067693   0.16069817       0.010823239      -0.065707688
## 66  -0.1911907522   0.26538362      -0.032412150       0.196773583
## 67  -0.8636321586   0.06975226      -0.003071244       0.018645465
## 68   0.2430608129   0.18121685      -0.034953085       0.212199557
## 7   -0.2555563878  -0.13049247      -0.012465983      -0.073513795
## 72  -0.2053946515   0.24592661      -0.010259031       0.062282394
## 73   0.1793900774  -0.33820540       0.034061407      -0.206786196
## 74  -0.4514453068  -0.05708319      -0.034017432       0.206519220
## 75  -0.3067556086  -0.28155979       0.018589867      -0.112858749
## 76  -0.2956548559  -0.35372140       0.035327686      -0.214473751
## 77  -0.3148163557   0.11131621       0.007163071      -0.043486875
## 78   0.0613942261  -0.04919909      -0.063400420       0.384902821
## 79   0.0495988760  -0.03565463      -0.060418625       0.366800392
## 8   -0.2460275192  -0.16365154      -0.002779113      -0.016388850
## 84   0.2939129506   0.42396437       0.044953934      -0.064579419
## 85   0.2776085260   0.41740124       0.054502535      -0.078296642
## 86   0.1956822745  -0.10085225       0.025474644      -0.036596080
## 87   0.0172554445  -0.33465937      -0.022656943       0.032548258
## 88   0.4757000508   1.03772669       0.071387507      -0.102553066
## 9   -0.2896191384  -0.21116911       0.009297944       0.054831388
## 92   0.0506056750  -0.20098157      -0.002255017       0.003239488
## 93   0.7354737876   0.01117134       0.029884599      -0.042931283
## 94   0.3159491992   0.50172646       0.031132840      -0.044724467
## 95   0.1091043260  -0.23982951       0.041764049      -0.059996930
## 96   0.2323927752   0.41278076       0.024178556      -0.034734159
## O1  -0.2365794802  -0.25844395      -0.212422212       0.271867968
## O10  0.1214373205   0.01136612       0.108526252      -0.367920428
## O11  0.3057043639  -0.26212604       0.052729231       0.052389292
## O12 -0.3628191000  -0.45355879       0.058315334      -0.016909001
## O13  0.2746815820   0.14800387      -0.008185723       0.048571895
## O2   0.1740665556   0.34225375       0.172176876      -1.028615066
## O3   0.2431308629   0.13200407      -0.038547653       0.159655530
## O4   0.1612199844  -0.27831215      -0.089503452       0.418557635
## O5  -0.0424660924  -0.48868914       0.030697279       0.059839532
## O6  -0.0545743949  -0.18315278      -0.050149497       0.116024369
## O7   0.1229848636   0.22421465       0.147803610       0.139018362
## O8  -0.5685054544  -0.90877683      -0.039475305      -0.248713156
## O9   0.3976758757   0.55919622       0.048179918       0.035280649
##     as.factor(Rater)3
## 100       0.031428653
```

```
## 101      -0.027916265
## 102      -0.064521325
## 103       0.037570599
## 104      -0.023051783
## 105      -0.053329026
## 106      -0.020382002
## 107      -0.009974093
## 111      -0.032361985
## 112       0.017772495
## 113      -0.014585599
## 114      -0.002084742
## 115       0.038739536
## 116      -0.027860118
## 117       0.009717358
## 118      -0.007806052
## 13       -0.536722039
## 15        0.113857679
## 16        0.167844409
## 17       -0.146997552
## 21        0.355229632
## 22        0.149025759
## 23       -0.464809179
## 24       -0.060976459
## 25       -0.313677937
## 26        0.128323345
## 27       -0.051335085
## 28       -0.563441313
## 32        0.223202027
## 33        0.154804949
## 34        0.246892285
## 35       -0.037273562
## 36       -0.060976459
## 37       -0.044136230
## 38       -0.020433333
## 39        0.085576768
## 40       -0.084968226
## 45       -0.051599354
## 46       -0.066122599
## 47       -0.224196674
## 48        0.213922386
## 49        0.103826651
## 53        0.245139427
## 54       -0.177667765
## 55        0.038040446
## 56       -0.070946830
## 57       -0.070977769
## 6        -0.111449830
## 61       -0.031670551
## 62        0.201197777
## 63        0.135380632
## 64        0.006550884
## 65       -0.039968574
## 66        0.119693139
## 67        0.011341636
```

```
## 68        0.129076428
## 7        -0.101808455
## 72        0.037885041
## 73       -0.125783597
## 74        0.125621200
## 75       -0.068649550
## 76       -0.130459771
## 77       -0.026452131
## 78        0.234128109
## 79        0.223116791
## 8        -0.022696740
## 84        0.040440194
## 85        0.049030038
## 86        0.022916784
## 87       -0.020382002
## 88        0.064219622
## 9         0.075935393
## 92       -0.002028595
## 93        0.026883943
## 94        0.028006850
## 95        0.037570599
## 96        0.021750833
## O1       -0.224083493
## O10      -0.112465388
## O11       0.174412982
## O12       0.118720019
## O13       0.029115529
## O2       -0.619306645
## O3        0.068657607
## O4        0.206836530
## O5        0.130611929
## O6       -0.001531173
## O7        0.481130258
## O8       -0.338167869
## O9        0.146890570
##
## $<NA>
## NULL
##
## $<NA>
## NULL
##
## $<NA>
## NULL
##
## $<NA>
## NULL
##
## $<NA>
## NULL
##
## $<NA>
## NULL
##
```

```
## $<NA>
## NULL
##
## $<NA>
## NULL
##
## $<NA>
## NULL
```

And the RE standard deviations and correlations are:

```
summary(comb.inter_elim_RE1)$varcor
```

```
##  Groups     Name              Std.Dev. Corr
##  Artifact   RubricCritDes     0.70457
##             RubricInitEDA     0.56379   0.318
##             RubricInterpRes   0.31947   0.142  0.674
##             RubricRsrchQ      0.42308   0.500  0.194  0.538
##             RubricSelMeth     0.19556   0.145  0.226  0.376 -0.241
##             RubricTxtOrg      0.50028   0.268  0.437  0.364  0.305  0.213
##             RubricVisOrg      0.48202   0.175  0.504  0.445  0.276 -0.161
##  Artifact.1 as.factor(Rater)1 0.11320
##             as.factor(Rater)2 0.33429  -0.486
##             as.factor(Rater)3 0.30682   0.332  0.663
##  Residual                     0.36699
##
##
##
##
##
##
##
##   0.537
##
##
##
##
```

Question 4: Is there anything else interesting to say about this data?

Although i've been modeling Ratings as numeric, it is actually an ordered categorical variable. It might be worthwhile repeating some of the procedures in this project using multinomial logistic regresssion or ordered logistic regression. That would be a whole other project, but I will try some logistic models on a reduced version of the dataset.
I subset the tall dataset into only observations with ratings of 2 and 3. These are the most common ratings. According to the definitions, the main difference between 2- and 3-scoring rubrics is 'flawed evidence' (2) vs. 'competent evidence' (3), so this subset should still give us a good idea of how raters assign scores based on perceived quality of student work.

I chose a group of nested models that we examined in Research Question 3, including the basic random intercept model, the model with backward variable selection performed with all FEs in the model, the model with additional FE interactions, and the final model-with a FE interaction and an additional RE.

Options for combined models from last part of Question 3:

```
#model with only random intercept
form0 <- formula(comb.0)
```

```
form0
```

```
## as.numeric(Rating) ~ 1 + (0 + Rubric | Artifact)
```

```
#model with all FEs, no interactions after variable selection:
form1 <- formula(comb.back_elim)
form1
```

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

```
#model with all FEs, and their interactions after variable selection:
form2 <- formula(comb.inter_elim)
form2
```

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

```
#final model with 1 FE interaction and 2 REs
form3 <- formula(comb.inter_elim_RE1)
form3
```

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

Subset data:

```
ratings_tall_noNA_2_3 <-
  ratings_tall_noNA[ratings_tall_noNA$Rating==2 | ratings_tall_noNA$Rating==3, ]
length(ratings_tall_noNA$Rating)
```

```
## [1] 810
```

```
length(ratings_tall_noNA_2_3$Rating)
```

```
## [1] 697
```

```
ratings_tall_noNA_2_3$Rating <- factor(ratings_tall_noNA_2_3$Rating, levels=c(2,3))
```

I fit the models using glmer.

```
#log version of model with only random intercept:
log0 <- glmer(Rating ~ 1 + (0 + Rubric | Artifact),
              data = ratings_tall_noNA_2_3, family=binomial)
#(takes awhile to run)

#log version of model with all FEs, no interactions after variable selection:
log1 <- glmer((Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
    Semester + Rubric, data = ratings_tall_noNA_2_3, family=binomial)
#(takes awhile to run)

#log version of model with all FEs, and their interactions after variable selection:
log2 <- glmer((Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
    Semester + Rubric + as.factor(Rater):Rubric, data = ratings_tall_noNA_2_3,
    family=binomial)
#(takes awhile to run)

#log version of final model with 1 FE interaction and 2 REs
log3 <- glmer((Rating) ~ (0 + Rubric | Artifact) +
                (0 + as.factor(Rater) | Artifact) + as.factor(Rater) +
```

```
                Semester + Rubric + as.factor(Rater):Rubric,
              data = ratings_tall_noNA_2_3, family=binomial)
#(takes awhile to run)
```

Model summaries:

```
cat("log0 summary\n")
```

```
## log0 summary
```

```
summary(log0)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Rating ~ 1 + (0 + Rubric | Artifact)
##    Data: ratings_tall_noNA_2_3
##
##      AIC      BIC   logLik deviance df.resid
##    908.9   1040.7   -425.4    850.9      668
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3269 -0.5367 -0.2861  0.5915  2.7265
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes    3.4723  1.8634
##           RubricInitEDA    4.2513  2.0619    0.07
##           RubricInterpRes 13.3643  3.6557    0.90  0.51
##           RubricRsrchQ     2.4896  1.5779    0.94  0.33  0.96
##           RubricSelMeth    0.8829  0.9397    0.18 -0.04  0.13 -0.02
##           RubricTxtOrg    47.5472  6.8954    0.49  0.59  0.69  0.75 -0.48
##           RubricVisOrg     2.6257  1.6204    0.52  0.79  0.80  0.73 -0.29  0.79
## Number of obs: 697, groups:  Artifact, 89
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.063      0.204  -5.211 1.88e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## optimizer (Nelder_Mead) convergence code: 4 (failure to converge in 10000 evaluations)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 2 negative eigenvalues
## failure to converge in 10000 evaluations
```

```
cat("log1 summary\n")
```

```
## log1 summary
```

```
summary(log1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: (Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester +
##     Rubric
```

```
##    Data: ratings_tall_noNA_2_3
##
##      AIC      BIC   logLik deviance df.resid
##    858.3   1031.1   -391.2    782.3      659
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.48891 -0.53758 -0.02177  0.53077  2.37937
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   27.227   5.218
##           RubricInitEDA    3.388   1.841    -0.13
##           RubricInterpRes  1.915   1.384     0.87  0.34
##           RubricRsrchQ     2.045   1.430     0.83  0.23  0.95
##           RubricSelMeth   64.867   8.054     0.15  0.27  0.27  0.07
##           RubricTxtOrg     2.183   1.477     0.53  0.65  0.81  0.75  0.06
##           RubricVisOrg     2.546   1.596     0.49  0.58  0.65  0.48  0.00  0.80
## Number of obs: 697, groups:  Artifact, 89
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.38555    4.37787  -0.316    0.752
## as.factor(Rater)2  0.07489    0.45106   0.166    0.868
## as.factor(Rater)3 -1.18569    0.56833  -2.086    0.037 *
## SemesterS19       -0.05150    0.43945  -0.117    0.907
## RubricInitEDA      1.43338    4.63787   0.309    0.757
## RubricInterpRes    2.03822    4.49484   0.453    0.650
## RubricRsrchQ       1.21102    4.50630   0.269    0.788
## RubricSelMeth     -5.58474    4.71346  -1.185    0.236
## RubricTxtOrg       2.70765    4.53590   0.597    0.551
## RubricVisOrg       1.31700    4.57559   0.288    0.773
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) a.(R)2 a.(R)3 SmsS19 RbIEDA RbrcIR RbrcRQ RbrcSM RbrcTO
## as.fctr(R)2  0.624
## as.fctr(R)3  0.754  0.746
## SemesterS19  0.083  0.102  0.111
## RubrcIntEDA -0.996 -0.653 -0.774 -0.114
## RbrcIntrpRs -0.997 -0.655 -0.780 -0.114  0.996
## RubrcRsrchQ -0.996 -0.657 -0.779 -0.109  0.995  0.997
## RubricSlMth -0.931 -0.608 -0.719 -0.160  0.933  0.933  0.931
## RubrcTxtOrg -0.996 -0.649 -0.779 -0.104  0.995  0.996  0.996  0.930
## RubricVsOrg -0.996 -0.654 -0.778 -0.112  0.996  0.996  0.996  0.930  0.996
## optimizer (Nelder_Mead) convergence code: 4 (failure to converge in 10000 evaluations)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 2 negative eigenvalues
## failure to converge in 10000 evaluations
```

```
cat("log2 summary\n")
```

```
## log2 summary
```

```
summary(log2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: (Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester +
##     Rubric + as.factor(Rater):Rubric
##    Data: ratings_tall_noNA_2_3
##
##      AIC      BIC   logLik deviance df.resid
##    864.1   1091.4   -382.0    764.1      647
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.5765 -0.4617 -0.1073  0.5177  2.6690
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes    6.056   2.461
##           RubricInitEDA    4.440   2.107    -0.07
##           RubricInterpRes  5.209   2.282     0.86  0.44
##           RubricRsrchQ     2.242   1.497     0.96  0.11  0.91
##           RubricSelMeth   13.183   3.631     0.23  0.31  0.41  0.08
##           RubricTxtOrg     1.949   1.396     0.67  0.50  0.83  0.83 -0.05
##           RubricVisOrg     2.687   1.639     0.41  0.64  0.66  0.63 -0.20  0.95
## Number of obs: 697, groups:  Artifact, 89
##
## Fixed effects:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -2.4074     1.2972  -1.856  0.06348 .
## as.factor(Rater)2                  2.4846     1.4977   1.659  0.09712 .
## as.factor(Rater)3                  2.0932     1.3523   1.548  0.12165
## SemesterS19                       -0.2982     0.5038  -0.592  0.55388
## RubricInitEDA                      2.7751     1.5146   1.832  0.06691 .
## RubricInterpRes                    4.2109     1.3869   3.036  0.00240 **
## RubricRsrchQ                       2.3106     1.3329   1.734  0.08300 .
## RubricSelMeth                     -1.8254     3.0379  -0.601  0.54791
## RubricTxtOrg                       3.9427     1.3762   2.865  0.00417 **
## RubricVisOrg                       1.3614     1.4085   0.967  0.33377
## as.factor(Rater)2:RubricInitEDA   -3.0678     1.8860  -1.627  0.10382
## as.factor(Rater)3:RubricInitEDA   -3.4805     2.1892  -1.590  0.11186
## as.factor(Rater)2:RubricInterpRes -3.5624     1.6073  -2.216  0.02666 *
## as.factor(Rater)3:RubricInterpRes -5.1227     1.5937  -3.214  0.00131 **
## as.factor(Rater)2:RubricRsrchQ    -2.5855     1.5547  -1.663  0.09632 .
## as.factor(Rater)3:RubricRsrchQ    -3.0551     1.4484  -2.109  0.03492 *
## as.factor(Rater)2:RubricSelMeth   -2.1850     1.8276  -1.196  0.23187
## as.factor(Rater)3:RubricSelMeth   -2.1022     1.7896  -1.175  0.24013
## as.factor(Rater)2:RubricTxtOrg    -3.1423     1.6000  -1.964  0.04953 *
## as.factor(Rater)3:RubricTxtOrg    -3.2211     1.4777  -2.180  0.02927 *
## as.factor(Rater)2:RubricVisOrg    -0.4565     1.6959  -0.269  0.78779
## as.factor(Rater)3:RubricVisOrg    -2.2130     1.5125  -1.463  0.14343
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation matrix not shown by default, as p = 22 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
## optimizer (Nelder_Mead) convergence code: 4 (failure to converge in 10000 evaluations)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 2 negative eigenvalues
## failure to converge in 10000 evaluations
```

```
cat("log3 summary\n")
```

```
## log3 summary
```

```
summary(log3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: (Rating) ~ (0 + Rubric | Artifact) + (0 + as.factor(Rater) |
##     Artifact) + as.factor(Rater) + Semester + Rubric + as.factor(Rater):Rubric
##    Data: ratings_tall_noNA_2_3
##
##      AIC      BIC   logLik deviance df.resid
##    871.9   1126.5   -380.0    759.9      641
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0728 -0.4403 -0.0898  0.4919  2.0982
##
## Random effects:
##  Groups     Name              Variance Std.Dev. Corr
##  Artifact   RubricCritDes     128.6163 11.3409
##             RubricInitEDA       2.4130  1.5534  -0.12
##             RubricInterpRes     3.2132  1.7925   0.85  0.41
##             RubricRsrchQ        2.0170  1.4202   0.95 -0.03  0.85
##             RubricSelMeth       8.5629  2.9262   0.26  0.29  0.39 -0.01
##             RubricTxtOrg        1.9088  1.3816   0.56  0.57  0.81  0.75 -0.12
##             RubricVisOrg        3.8846  1.9709   0.31  0.59  0.59  0.57 -0.35
##  Artifact.1 as.factor(Rater)1   0.0724  0.2691
##             as.factor(Rater)2   1.5868  1.2597  -1.00
##             as.factor(Rater)3   0.3028  0.5503  -0.37  0.37
##
##
##
##
##
##
##
##   0.95
##
##
##
## Number of obs: 697, groups:  Artifact, 89
##
## Fixed effects:
```

```
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -5.7053     3.1725  -1.798   0.0721 .
## as.factor(Rater)2                  5.5526     3.2948   1.685   0.0919 .
## as.factor(Rater)3                  3.6664     3.0733   1.193   0.2329
## SemesterS19                       -0.3232     0.4894  -0.660   0.5090
## RubricInitEDA                      5.9129     3.1971   1.849   0.0644 .
## RubricInterpRes                    7.2951     3.0805   2.368   0.0179 *
## RubricRsrchQ                       5.6530     3.0975   1.825   0.0680 .
## RubricSelMeth                      2.0083     3.8069   0.528   0.5978
## RubricTxtOrg                       7.3179     3.1558   2.319   0.0204 *
## RubricVisOrg                       4.5249     3.1642   1.430   0.1527
## as.factor(Rater)2:RubricInitEDA   -5.9562     3.3655  -1.770   0.0768 .
## as.factor(Rater)3:RubricInitEDA   -4.7478     3.1759  -1.495   0.1349
## as.factor(Rater)2:RubricInterpRes -6.3041     3.2357  -1.948   0.0514 .
## as.factor(Rater)3:RubricInterpRes -6.4009     3.0240  -2.117   0.0343 *
## as.factor(Rater)2:RubricRsrchQ    -5.7318     3.2265  -1.776   0.0757 .
## as.factor(Rater)3:RubricRsrchQ    -4.6573     3.0109  -1.547   0.1219
## as.factor(Rater)2:RubricSelMeth   -5.3687     3.4140  -1.573   0.1158
## as.factor(Rater)3:RubricSelMeth   -3.6575     3.1824  -1.149   0.2504
## as.factor(Rater)2:RubricTxtOrg    -6.1156     3.2975  -1.855   0.0636 .
## as.factor(Rater)3:RubricTxtOrg    -4.8963     3.0844  -1.587   0.1124
## as.factor(Rater)2:RubricVisOrg    -3.1644     3.3479  -0.945   0.3446
## as.factor(Rater)3:RubricVisOrg    -3.8555     3.1010  -1.243   0.2138
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation matrix not shown by default, as p = 22 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
## optimizer (Nelder_Mead) convergence code: 4 (failure to converge in 10000 evaluations)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 4 negative eigenvalues
## failure to converge in 10000 evaluations
```

The things that stick out the most from the model summaries are the irregular and large Tau^2 values. These indicate that the model fit is not great. The variances we saw in the Question 3 models that ranged from 0-1 made sense for amounts that a mean score would vary, but Tau^2 values like 13.183 and 128.6163 do not. Some of the FE coefficients are also larger in size than you would expect, with values like 7.3179 and -6.1156. These represent log(odds) instead of slopes like before, so maybe these values are okay, but they stand out.

The glmer function resulted in the same models each time i ran it, so i think it is fitting them correctly (was generating warnings at first). So it seems that the problem is that this is just not as good of a way to model the data as treating Ratings as numeric. Or perhaps the 2 and 3 rated rubrics are not dissimilar enough from each other, and excluding all the other ratings is causing the bad fits.

Nevertheless, i will test which one of these models is the best using F-tests, AIC, and BIC.

```
anova(log0, log1, log2, log3)
```

```
## Data: ratings_tall_noNA_2_3
## Models:
## log0: Rating ~ 1 + (0 + Rubric | Artifact)
## log1: (Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Rubric
## log2: (Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Rubric + as.factor(Rater):Ru
## log3: (Rating) ~ (0 + Rubric | Artifact) + (0 + as.factor(Rater) | Artifact) + as.factor(Rater) + Sem
```

```
##      npar    AIC    BIC  logLik deviance   Chisq Df Pr(>Chisq)
## log0   29 908.87 1040.7 -425.43   850.87
## log1   38 858.32 1031.1 -391.16   782.32 68.5426  9  2.938e-11 ***
## log2   50 864.07 1091.4 -382.03   764.07 18.2546 12     0.1082
## log3   56 871.92 1126.5 -379.96   759.92  4.1523  6     0.6561
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All test results point to the 'log1' model. This was the model that included the random intercept and 3 fixed effects for Rater, Semester, and Rubric.

```
formula(log1)
```

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