Statistical Modeling to Decide Whether the Evaluation for an Education Program is Fair Enough

Ziyan Xia
Department of Statistics and Data Science, Carnegie Mellon University
zxia2@andrew.cmu.edu
December 10, 2021

Abstract:

Colleges usually need to evaluate the quality of their education programs. While they are doing this, it's important for them to know whether the evaluation process is fair and appropriate. To learn this, they need to know what factors affect the ratings of the evaluation. We used the ratings of project papers sampled from two semesters that were rated by 3 rates across the college to do the analysis. We used methods including plots, Intraclass Correlation, backward elimination, ANOVA tests and likelihood ratio test. We find that ratings are affected by semester, rubric, rater and artifact. However, based on exploratory data analysis, there seems to be other factors that also affect ratings. We still need to do further work to find out whether these factors really have a large influence on ratings.

1. Introduction

It's always important for colleges to evaluate the quality of their education programs. Some colleges use the ratings of education-relevant statistics to decide whether an education program is successful. Dietrich College at Carnegie Mellon University is implementing a new "General Education" program for undergraduates. In order to determine whether this program is successful, the college hopes to rate student work performed in each of the "Gen Ed" courses each year. Recently the college has been experimenting with rating work in freshman statistics, using raters from across the college. For experiments like this, we always wonder whether it's truly fair to use the ratings from these raters based on these rubrics. To be more specific, there are four questions to answer and they are as follows:

1. **Do rater's ratings vary much by raters or rubrics?**

   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. **Do rater's ratings reach a consensus?**

   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. **How do various factors affect ratings?**

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. Other Interesting things about ratings

Is there anything else interesting to say about this data?

2. Data

In a recent experiment with rating work in freshman statistics, 91 project papers—referred to as "artifacts"—were randomly sampled from a Fall and Spring section of freshman statistics. The job to rate these 91 artifacts on seven rubrics was done by three raters from three different departments. Among these 91 artifacts, only 13 of them were rated by all three raters. Each of the other 78 artifacts was rated by only rater. (Junker (2021))

Rubrics for rating Freshman Statistics projects are shown in Table 1. The rating scale used for all rubrics is shown in Table 2. Variables in the file that we are using are shown in Table 3. Summary statistics for each rubric are shown in Table 4. Summary statistics for variables Semester and Gender are shown in Table 5.

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

Table 1: Rubrics for rating freshman statistics projects

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.

Table 2: Rating scale used for all rubrics

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

Table 3: Variables in the file that we are using

Rubric / Rating	RsrchQ	CritDes	InitEDA	SelMeth	InterpRes	VisOrg	TxtOrg
Rating 1	6	47	8	10	6	7	8
Rating 2	65	39	56	89	49	59	37
Rating 3	45	28	47	18	61	45	66
Rating 4	1	2	6	0	1	5	6
Missing Value	0	1	0	0	0	1	0
Total Count	117	117	117	117	117	117	117

Table 4: Summary statistics for each rubric

Gender / Semester	Female	Male	Missing Value	Total
Fall	38	44	1	83
Spring	26	8	0	34
Total	64	52	1	117

Table 5: Summary statistics for variables Semester and Gender

3. Methods

We first created a subset of the original dataset and this dataset contains the data of 13 artifacts seen by all 3 raters. We call this subset "reduced dataset" and call the original dataset "full dataset".

1. Do rater's ratings vary much?

To answer this question, we first made bar plots and tables for the counts of ratings for each rubric on both the reduced dataset and the full dataset. We also made bar plots and tables for the counts of ratings (with possibly missing values) for each rater on both the reduced dataset and the full dataset. (See Technical Appendix, Page 1-6)

2. Do rater's ratings reach a consensus?

To answer this question, we fitted seven random-intercept models with artifact as the grouping variable for each rubric on the reduced dataset. Then we calculated the seven intraclass correlations (ICC) of these seven rubric-specific models to measure of agreement among the raters. We did the same process also on the full dataset. Besides, we made a two-way table of counts for the ratings of each pair of raters, on each rubric to determine who is agreeing with whom on each rubric. From these tables, we were able to calculate % exact agreement between each pair of raters in ratings for each rubric. (See Technical Appendix, Page 6-7)

The random effect here says how much the scores vary across artifacts, from the prediction made by the fixed effects. The ICC is an estimate of the correlation between the raters. If the correlation is high, then when one rater's rating goes up from one artifact to the next, we expect the other raters' ratings to go up as well. Therefore, the ICC can be a measure of agreement between the raters, in the sense that high ICC means they agree that one artifact is better than another.

3. How do various factors affect ratings?

a) Adding fixed effects to the seven rubric-specific models using the reduced dataset

After fitting seven random-intercept models for each rubric on the reduced dataset for Question 2, we got seven intercept-only models grouped by artifact as our baseline models. We created seven "full-fixed-effects" models with all the possible fixed effects (Rater, Semester, Sex) and the random effect of the baseline models (the random-intercept with artifact as the grouping variable), then we did backward elimination to conduct fixed effect selection (See Technical Appendix, Page 8-9). Backward elimination will give us new models and we compared them with the baseline models using ANOVA tests to gain the best model fits for each rubric.

b) Adding fixed effects to the seven rubric-specific models using the full dataset

Before using the full dataset to do modeling, we deleted all the observations with missing data to ensure that we use the same dataset for every model fit and model comparison. We repeated the same process as a) on the full dataset and compared the seven rubric-specific

models gained this time with the seven models gained from a) (See Technical Appendix, Page 9-10).

c) Trying adding interactions and new random effects for the seven rubric specific models using the full dataset

We should only add random effects that are also present as fixed effects and we should only add interactions if there are already more than one fixed effect in the model. As one of seven rubric-specific models fitted on the full dataset in b) has more than one fixed effect, we tried adding interaction and new random effects grouped by artifact using ANOVA test and likelihood ratio test for this model. Also, as some of the seven rubric-specific models fitted on the full dataset from b) have one fixed effect, we tried adding new random effects with the same grouping variable artifact for these models using likelihood ratio tests. (See Technical Appendix, Page 10-21) After this process, we got seven final rubric-specific models which were the best fits.

d) Trying adding fixed effects, interactions, and new random effects to the "combined" model using the full dataset

For this step, we used a combined model. In this model, we didn't treat each rubric separately, instead, we treat Rubric as an explanatory variable and directly tested how it affected the response variable Rating. We started with an intercept-only model with Rubric as a random intercept grouped by artifact, then added all the possible fixed effects (Rater, Semester, Sex, Repeated, Rubric) to the model. We applied backward elimination method to this model to find out which fixed effect should be kept (See Technical Appendix, Page 29-31). The model with selected fixed effects was again added interactions among all left fixed effects and we also did backward elimination to select which interaction should be kept (See Technical Appendix, Page 32-34). Then, we used ANOVA test to select the current best fit using AIC, BIC, p values. After this, we began adding random effects to the previous best bit using likelihood ratio test. Finally, we got the best model fit (See Technical Appendix, Page 37).

4. Other Interesting things about ratings

From results of Question 2, we were able to find out some kind of interaction between rater and rubric. We were also interested in whether the not-selected fixed effects in the model interact with Rubric. Therefore, we made facet plots of counts of ratings partitioned by [Rubric, Rater], [Rubric, Sex] and [Rubric, Semester].

4. Results

1. Do rater's ratings vary much?

Figure 1 and Figure 2 are the bar plots for the counts of ratings for each rubric on the reduced dataset and the full dataset. Figure 3 and Figure 4 are the bar plots for the counts of ratings for each rater on the reduced dataset and the full dataset.

From these four figures and other summary statistics (See Technical Appendix, Page 1-6). We have some findings to answer Question 1 and they are as follows:

a) The distribution of ratings varies through rubrics. For example, Rubric CritDes tends to get lower ratings than other rubrics and Rubric TxtOrg tends to get higher ratings than other rubrics.
b) There aren't many differences in the distribution of ratings for each rubric between the reduced dataset and the full dataset.
c) The scoring patterns for all 3 raters aren't obviously different. Rater 3 tends to give slightly lower ratings compared to other raters.
d) There is a slight difference in the distribution of ratings for Rater 2 between the two datasets. Rater 2 tends to give higher ratings in the full dataset which contains all the artifacts ratings.

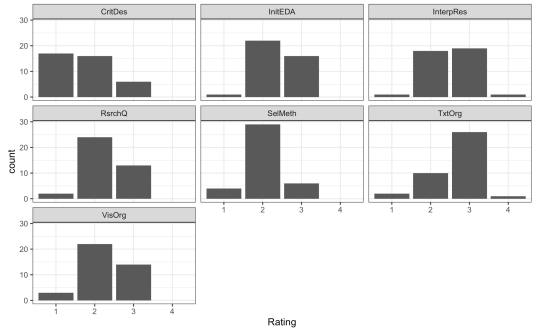


Figure 1: Bar plots of ratings count on each rubric (reduced dataset)

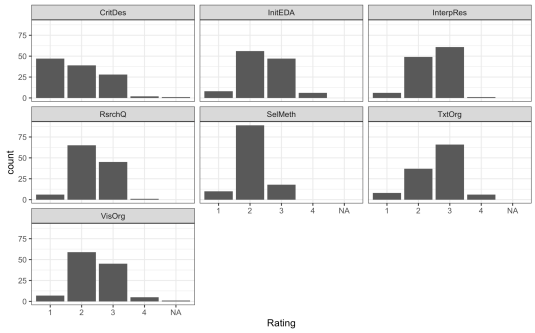


Figure 2: Bar plots of ratings count on each rubric (full dataset)

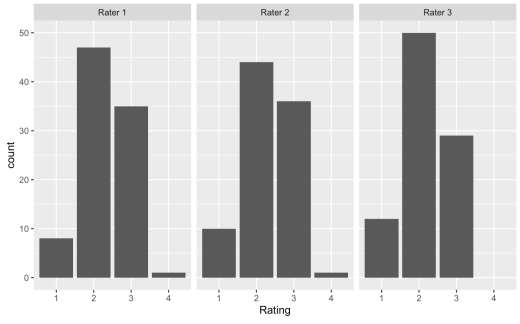


Figure 3: Bar plots of ratings count for each rater (reduced dataset)

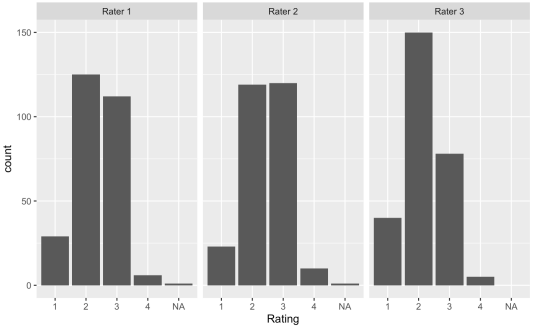


Figure 4: Bar plots ratings count for each rater (full dataset)

2. Do rater's ratings reach a consensus?

After calculating the Intraclass Correlation (ICC) on both the reduced dataset and the full dataset and calculating % exact agreement between each pair of raters in ratings for each rubric, we created Table 6 to compare them. As is shown in table 6, the column "Intraclass Correlation (ICC)" means the ICCs calculated from the 14 random-intercept models from Question 1. These ICCs are separated into two categories based on whether from the full dataset or the reduced dataset. From Table 6, we find that:

a) For rubric CritDes, InitEDA and VisOrg, the raters generally agree about which artifacts are better than which other ones because the ICC is quite high. All three raters give quite similar scores to the same artifacts as the % agreement is also quite high.

b) For rubric InterpRes, RsrchQ and TxtOrg, the raters generally disagree about which artifacts are better than which other ones because the ICC is quite low. However, all three raters give quite similar scores to the same artifacts because the % agreement is quite high.

c) For rubric SelMeth, % agreement between each pair of raters is quite high, especially for Rater 1 and Rater 2, which is more than 90%. However, the ICC is not that high compared to the % agreement.

Type / Rubric	Intraclass Correlation (ICC)		% agreement between each pair of raters		
	full data	reduced data	Rater 1 & 2	Rater 2 & 3	Rater 1 & 3
CritDes	0.67	0.57	0.54	0.69	0.62
InitEDA	0.69	0.49	0.69	0.85	0.54
InterpRes	0.22	0.23	0.62	0.62	0.54
RsrchQ	0.21	0.19	0.38	0.54	0.77
SelMeth	0.47	0.52	0.92	0.69	0.62
TxtOrg	0.19	0.14	0.69	0.54	0.62
VisOrg	0.66	0.59	0.54	0.77	0.77

Table 6: ICCs and raters agreement rate for each rubric

3. How do various factors affect ratings?

Coefficients of fixed effects of final models are shown in Table 7. Table 7 has eight columns, the first seven are "rubric-specific Models" and the last one is the combined model. It is worth mentioning that all these final models are fitted on the full dataset. (See Technical Appendix, Page 38-43)

Model / Coefficient Name	Rubric-Specific Models							Combined Model
	CritDes	InitEDA	InterpRes	RsrchQ	SelMeth	TxtOrg	VisOrg	--
Intercept	--	2.44	--	2.35	--	2.59	--	1.76
Rater1	1.69	--	2.70	--	2.25	--	2.38	--
Rater2	2.11	--	2.59	--	2.23	--	2.65	0.37
Rater3	1.89	--	2.14	--	2.03	--	2.28	0.20
SemesterS19	--	--	--	--	-0.36	--	--	-0.16
RubricInitEDA	--	--	--	--	--	--	--	0.74
RubricInterpRes	--	--	--	--	--	--	--	0.99
RubricRsrchQ	--	--	--	--	--	--	--	0.73
RubricSelMeth	--	--	--	--	--	--	--	0.41

RubricTxtOrg	--	--	--	--	--	--	--	1.02
RubricVisOrg	--	--	--	--	--	--	--	0.65
Rater2: RubricInitEDA	--	--	--	--	--	--	--	-0.30
Rater3: RubricInitEDA	--	--	--	--	--	--	--	-0.29
Rater2: RubricInterpRes	--	--	--	--	--	--	--	-0.51
Rater3: RubricInterpRes	--	--	--	--	--	--	--	-0.71
Rater2: RubricRsrchQ	--	--	--	--	--	--	--	-0.49
Rater3: RubricRsrchQ	--	--	--	--	--	--	--	-0.32
Rater2: RubricSelMeth	--	--	--	--	--	--	--	-0.39
Rater3: RubricSelMeth	--	--	--	--	--	--	--	-0.39
Rater2: RubricTxtOrg	--	--	--	--	--	--	--	-0.55
Rater3: RubricTxtOrg	--	--	--	--	--	--	--	-0.44
Rater2: RubricVisOrg	--	--	--	--	--	--	--	-0.10
Rater3: RubricVisOrg	--	--	--	--	--	--	--	-0.28

Table 7: Coefficients of fixed effects of final models

a) Interpreting rubric-specific models

All seven rubric-specific models have random effect grouped by artifact but some of them have different fixed effects. (See Technical Appendix, Page 10-21) To interpret the coefficients of these models, we separate them into three types:

i. The first type is the intercept-only model including Rubric InitEDA, RsrchQ and TxtOrg. The overall mean of ratings for each rubric here is the intercept coefficient shown in Table. 7. The mean of ratings vary across artifact. For example, the overall mean of ratings for [Rubric InitEDA] is 2.44 and the mean of ratings for [Rubric InitEDA, Artifact 94] is 2.44 + 1.07 = 3.51, which is relatively larger than most ratings of other artifacts.

ii. The second type is model with fixed effect Rater including Rubric CritDes, InterpRes and VisOrg. The mean of ratings now varies across rater and artifact. For example, the overall mean of ratings for [Rubric CritDes, Rater 1] is 1.69 and the mean of ratings for [Rubric CritDes, Rater 1, Artifact O9] is 1.69 - 0.75 = 0.94, which is relatively smaller than most ratings of other artifacts.

iii. The third type only has the model of Rubric SelMeth, which is a model with fixed effect. The mean of ratings now varies across rater, semester and artifact. For example, the overall mean of ratings for [Rubric SelMeth, Rater 1, Semester Fall 2019] is 2.25. The mean of ratings for [Rubric SelMeth, Rater 1, Semester Fall 2019, Artifact O4] is 2.25 + 0.59 = 2.84, which is relatively larger than most ratings of other artifacts.

b) Interpreting the combined models

We only give the coefficients of fixed effect (Rater, Semester and Rubric) and interaction between fixed effects (Rater x Rubric) in Table 7. There are also random effects in the final combined model. To better explain the final combined model's coefficients, we interpret the model into four parts:

Part A: fixed effect Rater + random effect Rater group by Artifact.

There is a kind of Rater x Artifact interaction: each Rater's rating on each Artifact differs from what we would expect (from the fixed effects alone) by a small random effect that depends on the Artifact.

For example, controlling for other effects in the model, the mean of ratings for [Rater 1, Artifact O7] is 1.76 + 0.15 = 1.91, which is relatively larger than most ratings of other artifacts.

Part B: fixed effect Rater + fixed effect Rubric +interaction between Rater and Rubric

There is a Rater x Rubric interaction: each Rater uses each rubric in a way that is not like, or even parallel to, other rater's rubric usage.

For example, controlling for other effects in the model, the mean of ratings for [Rater 2, Rubric InitEDA] will be 0.37 + 0.74 - 0.30 = 0.81 unit higher than that of [Rater 1, Rubric CritDes].

Part C: fixed effect Rubric + random effect Rubric group by Artifact.

There is a kind of Rubric x Artifact interaction: There are different average scores on each rubric, but the rubric averages also vary a bit from one artifact to the next, by a small random effect that depends on artifact.

For example, controlling for other effects in the model, the mean of ratings for [Rubric InitEDA, Artifact 16] is 0.74 + 1.15 = 1.89

Part D: fixed effect Semester

This is very easy to interpret. Controlling for other effects, which are the effects appearing in Part A, Part B and Part C, the mean of ratings for Semester Spring 19 is 0.16 units lower than that of Semester Fall 19.

In conclusion, if model each rubric separately, for the reduced dataset, ratings are affected only by Artifact. But the final rubric-specific models for the full dataset are different because, for these models, ratings are also affected by Rater or both Semester and Rater for some rubrics. For the combined model where we treat rubrics as a variable Rubric, ratings are affected by Semester, Rubric, Rater and Artifact based on the final formula. There also exist interactions between these variables.

4. Other Interesting things about ratings

Counts of ratings partitioned by Rubric and Rater are shown in Figure 5. Counts of ratings partitioned by Rubric and Sex are shown in Figure 6. Counts of ratings partitioned by Rubric and Semester are shown in Figure 7. From these figures, we find that:

a) From Figure 5, 3 raters seem to have different ways of scoring the 7 rubrics. It is not the case that one rater is simply harsher than another. The scoring patterns of Rater 3 are more stable across rubrics compared to the other two. The scoring patterns of Rater 1 and Rater 2 are alike for rubric InitEDA, InterpRes, RsrchQ, SelMeth and TxtOrg but quite different for the other two rubrics. The scoring patterns of Rater 1 and Rater 2 vary across rubrics.

b) From Figure 6, the distribution of ratings varies across both Rubric and Sex. The variation of ratings across rubric is more for females than males.

c) From Figure 7, the distribution of ratings varies across both Rubric and Semester. The observations for the fall semester are a lot more than the spring semester. The distribution of ratings is most different between semesters for rubric RsrchQ and SelMeth.

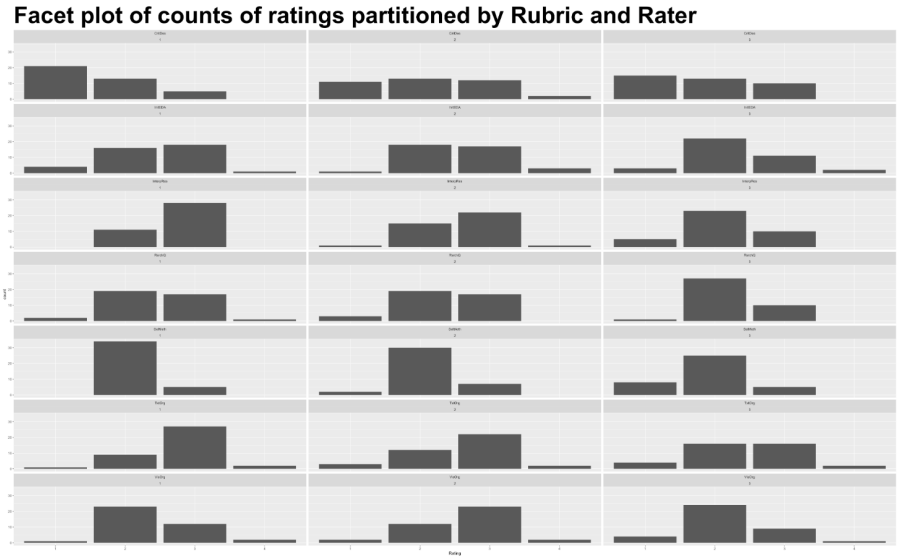


Figure 5: Facet plot of counts of ratings partitioned by Rubric and Rater

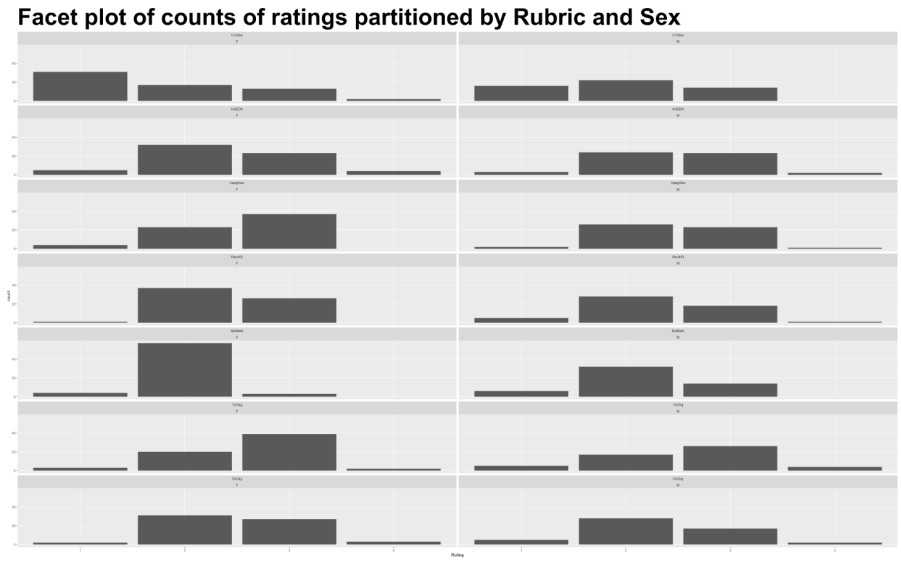


Figure 6: Facet plot of counts of ratings partitioned by Rubric and Sex

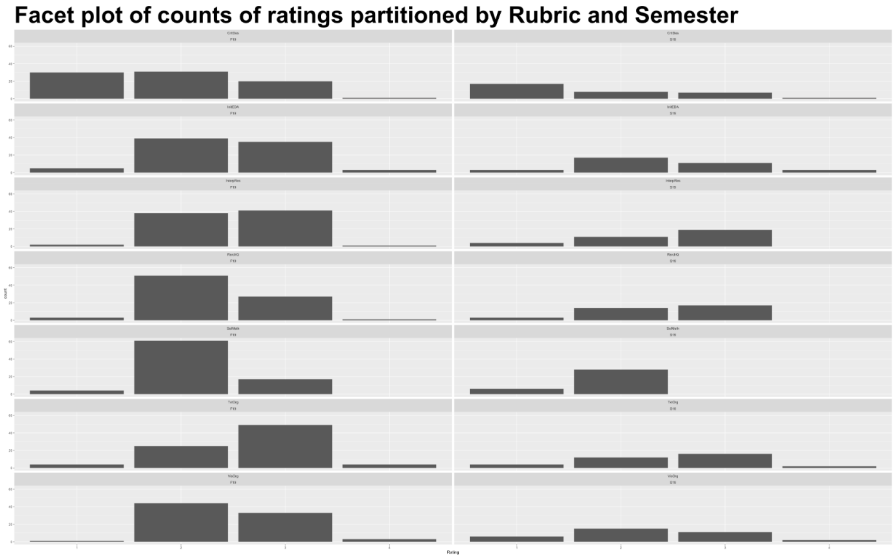


Figure 7: Facet plot of counts of ratings partitioned by Rubric and Semester

5. Discussion

The results show when evaluating how successful the education program is, we should also consider the effects that Semester, Rater, Rubric and Artifact as well as their interactions on the ratings. Besides, whether there are other factors (Sex, Repeated and interaction between Semester and Rubric) also affecting ratings still needs further work to find out. We will discuss these in the following.

1. **Do rater's ratings vary much by raters or rubrics?**

   For this question, we find whether using the reduced dataset or not doesn't create obvious differences for the distribution of ratings for each rubric or for each rater. Therefore, we can think of the reduced dataset as a good representative of the full dataset.

2. **Do rater's ratings reach a consensus?**

   Even though the raters may agree that one artifact is better than another, they may not rate in exactly the same way. For example, rater 1 could rate the first artifact 3 and the second artifact 4, and rater 2 could rate the first artifact 2 and the second artifact 3. The correlation would be exactly 1, but the % agreement would be zero. Therefore, we look at ICC to see if the raters generally agree about which artifacts are better than which other ones, but we look at % exact agreement to see if all three raters give exactly the same scores to the same artifacts.

   For some rubrics, the raters didn't seem to have a consensus about which artifact is better as ICCs for these rubrics are really small. For these rubrics which have small ICCs, there should be more work on deciding whether to use these rubrics because if raters have very different opinions about whether the artifact is a good one, it's probably not that appropriate to use these rubrics and thus not that fair to use these rubrics to evaluate the program. However, if the ICC is high while

the % exact agreement is low, it probably means that one rater is harsher than others on scoring these artifacts.

3. **How do various factors affect ratings?**

From Question 3's results, the fact that rubric scores depend on Artifact (that is, there is a kind of Rubric x Artifact interaction) is what we might expect: the artifacts aren't all of equal quality on each rubric, and so we should expect the average scores on each rubric to vary from one artifact to the next. More troubling are the Rater x Rubric interaction and the "kind of" Rater x Artifact interaction. The Rater x Rubric interaction suggests that the raters are not all interpreting the rubrics in the same way. The "kind of" Rater x Artifact interaction suggests that the raters are not interpreting the evidence in the artifacts in the same way. These interactions suggest that perhaps the raters should be trained more, to make the raters' ratings more similar to each other.

It's quite interesting that some rubric-specific models we got on the full dataset are different from the models on the reduced dataset after doing backward elimination and ANOVA tests. It potentially indicates that the interaction of Rubric and Repeated could affect ratings as the reduced dataset is the subset of the full dataset when Repeated = 1. This kind of contradicts what we find in Question 1 because we think of the reduced dataset as a good representative of the full dataset and the distribution of ratings for each rubric or each rater doesn't vary a lot across two datasets.

4. **Other Interesting things about ratings**

Figure 5 actually verifies the interaction we found for the combined model in Question 3 by showing 3 raters seem to have different ways of scoring the 7 rubrics. It also verifies the coefficients of this interaction shown in Table 7, for example, for Rubric VisOrg, the Rater 3 scores lower ratings than Rater 2, which by coefficients of interactions shown in Table 7, is the mean of ratings for Rater 3 is 0.28 – 0.10 = 0.17 unit lower than Rater 2.

Figure 6 shows that the distribution of ratings varies across both Rubric and Sex, which indicates the interaction between Rubric and Sex, and Figure 7 shows that the distribution of ratings varies across both Rubric and Semester, which indicates the interaction between Rubric and Semester. However, these interactions don't exist in the combined model due to the reasons that Sex wasn't selected as a fixed effect and interaction between Rubric and Semester wasn't considered statistically significant by likelihood ratio test. This is a weakness of our study because apparently, the final combined model doesn't capture the pattern shown in Figure 6 and Figure 7.

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 29, 2021 from https://canvas.cmu.edu/courses/25337/files/folder/Project02

Technical Appendix

Ziyan Xia

11/28/2021

```
tall <- read.csv("/Users/ceciliaxia/Desktop/tall.csv",header=T)
ratings <-  read.csv("/Users/ceciliaxia/Desktop/ratings.csv",header=T)
tall$Rating <- factor(tall$Rating,levels=1:4)
for (i in unique(tall$Rubric)) {
  ratings[,i] <- factor(ratings[,i],levels=1:4)
}

## Make the "tall" be consistent with the "ratings" coding.
tall$Sex[nchar(tall$Sex)==0] <- "--"

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

Data

Summary Statistics of Ratings of Rubrics and Summary Statistics of Sex and Semester

```
summary(ratings[,c(7:13)])%>%
  kbl() %>%
  kable_styling()
```

```
table(ratings[,c(5:6)])%>%
  kbl() %>%
  kable_styling()
```

1. Do rater's ratings vary much?

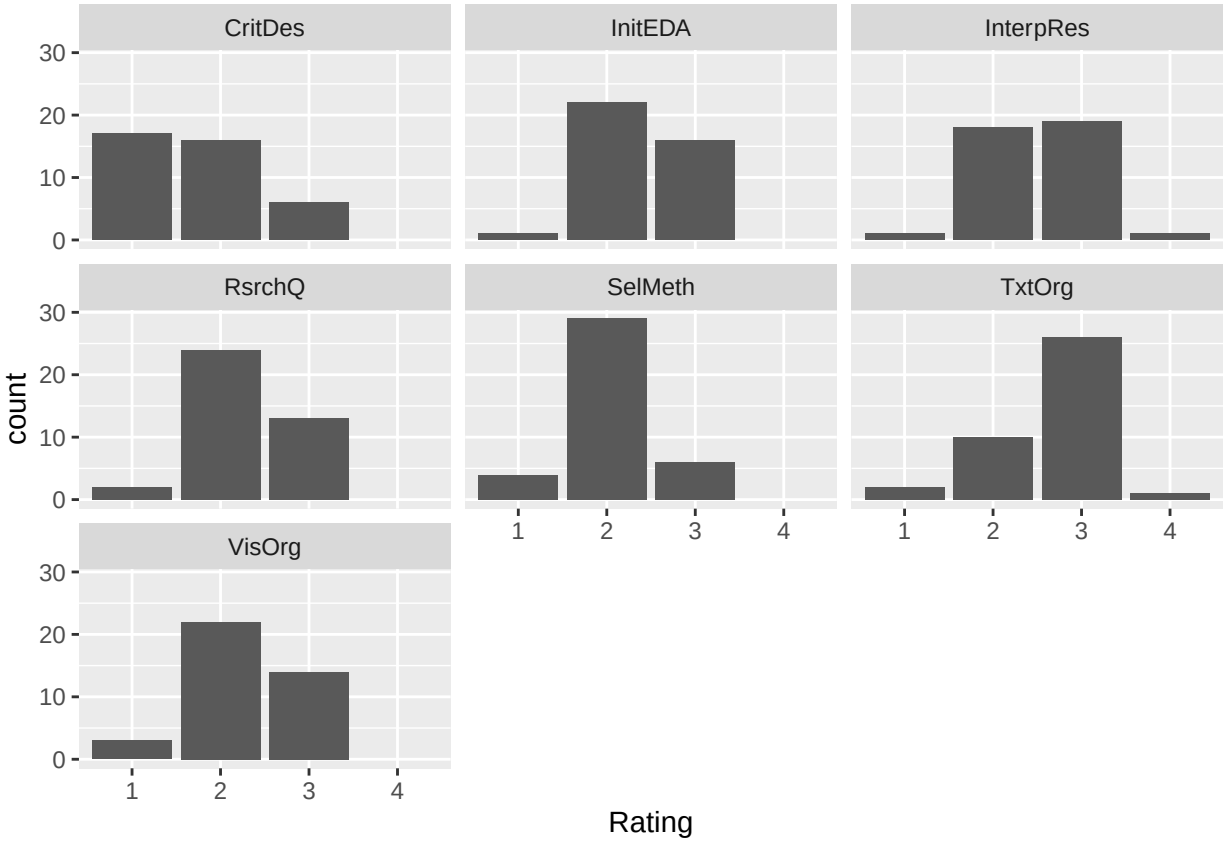
Figure 1: Bar plots of ratings count on each rubric (reduced dataset)

```
ggplot(tall.13,aes(x = Rating)) +
 facet_wrap( ~ Rubric) +
 geom_bar()
```

	RsrchQ	CritDes	InitEDA	SelMeth	InterpRes	VisOrg	TxtOrg
	1: 6	1 :47	1: 8	1:10	1: 6	1 : 7	1: 8
	2:65	2 :39	2:56	2:89	2:49	2 :59	2:37
	3:45	3 :28	3:47	3:18	3:61	3 :45	3:66
	4: 1	4 : 2	4: 6	4: 0	4: 1	4 : 5	4: 6
	NA	NA's: 1	NA	NA	NA	NA's: 1	NA

	–	F	M
Fall	1	38	44
Spring	0	26	8

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



Counts of rating for each rubric (reduced dataset)

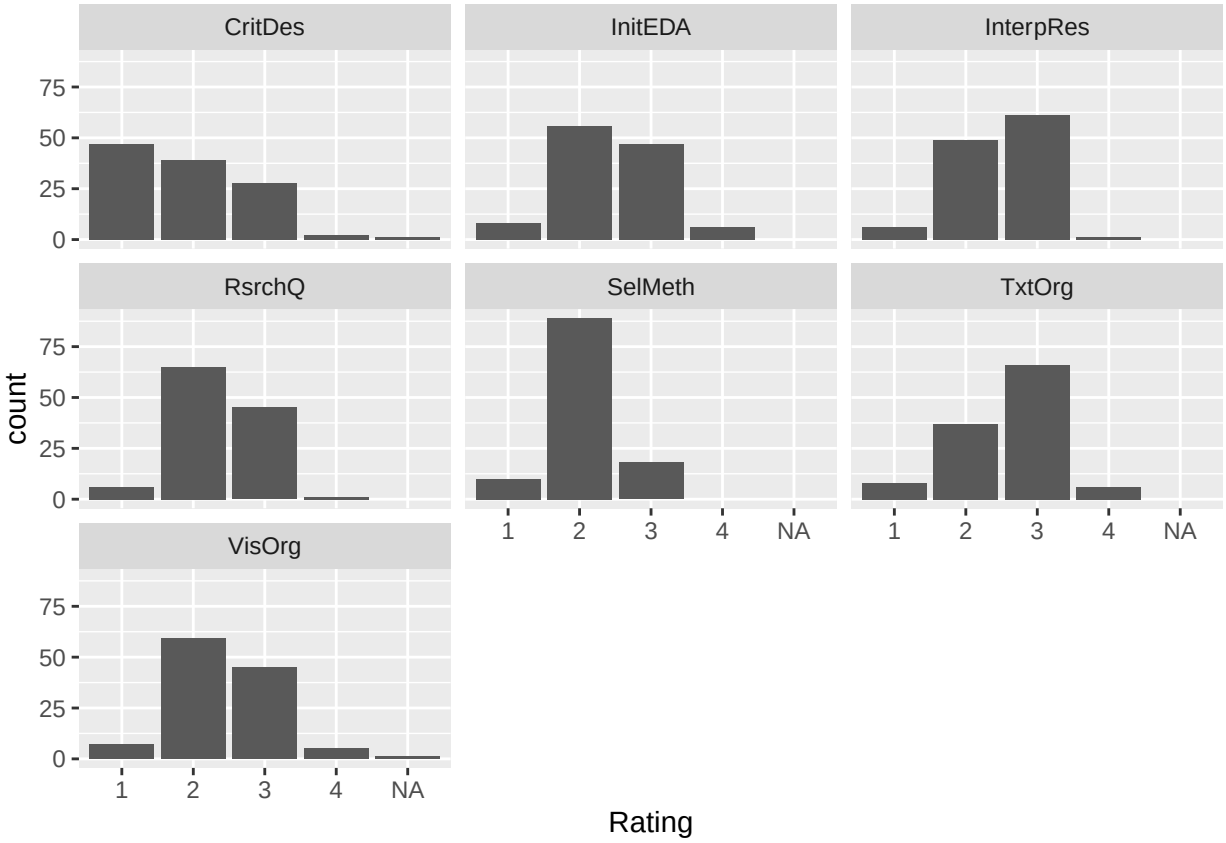
```
tmp <- data.frame(lapply(split(tall.13$Rating,tall.13$Rubric),summary))
row.names(tmp) <- paste("Rating",1:4)

tmp %>%
  kbl() %>%
  kable_styling()
```

Figure 2: Bar plots of ratings count on each rubric (full dataset)

```
ggplot(tall,aes(x = Rating)) +
  facet_wrap( ~ Rubric) +
  geom_bar()
```

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



Counts of rating for each rubric (full dataset)

```
tmp0 <- lapply(split(tall$Rating,tall$Rubric),summary)
tmp <- data.frame(matrix(0,nrow=5,ncol=7))
names(tmp) <- names(tmp0)
row.names(tmp) <- c(paste("Rating",1:4),"<NA>")
for (i in names(tmp0)) {
  tmp[,i] <- tmp[,i] + c(tmp0[[i]],0)[1:5]
}
tmp %>%
  kbl() %>%
  kable_styling()
```

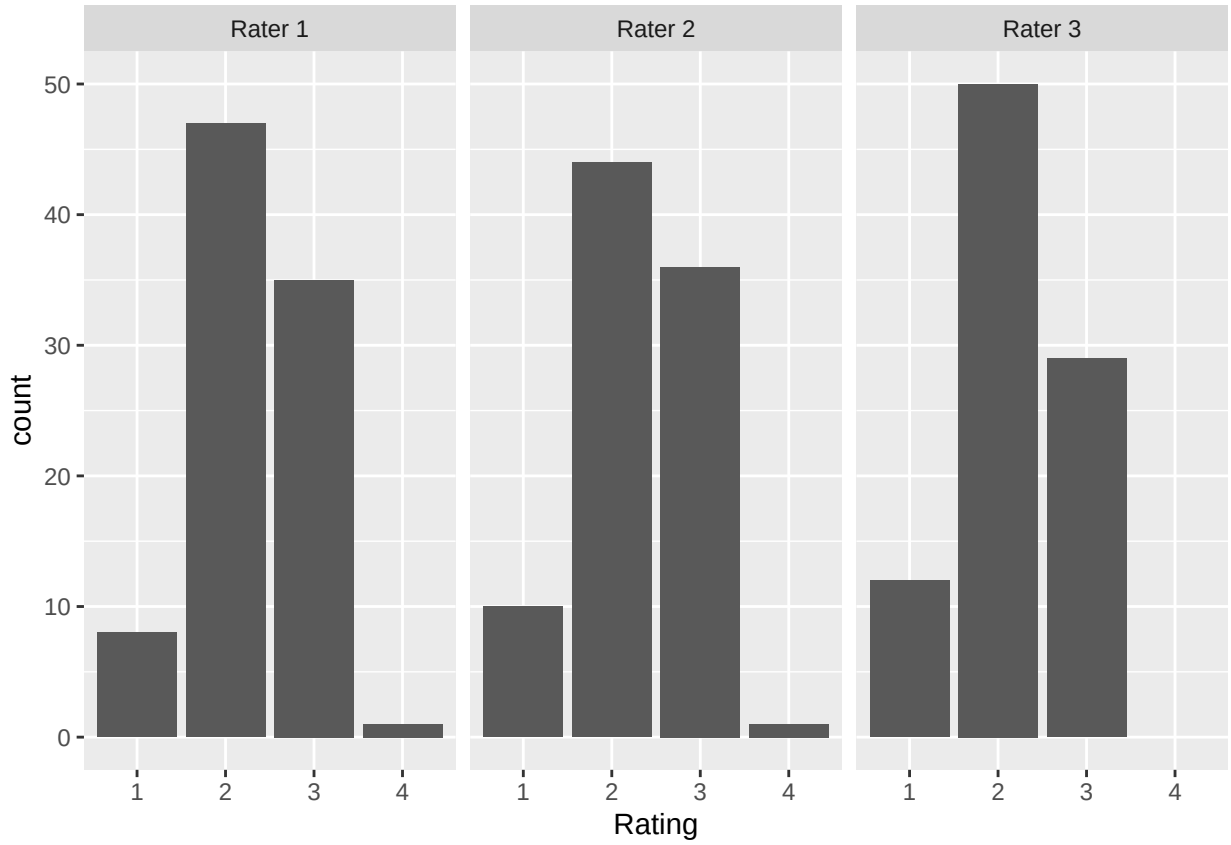
Figure 3: Bar plots of ratings count for each rater (reduced dataset)

```
rater.name <- function(x) { paste("Rater",x) }
g <- ggplot(tall.13,aes(x = Rating)) +
  facet_wrap( ~ Rater, labeller=labeller(Rater=rater.name)) +
```

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

```
  geom_bar()

g
```



Counts of rating for each rubric (reduced dataset)

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

tmp %>%
  kbl() %>%
  kable_styling()
```

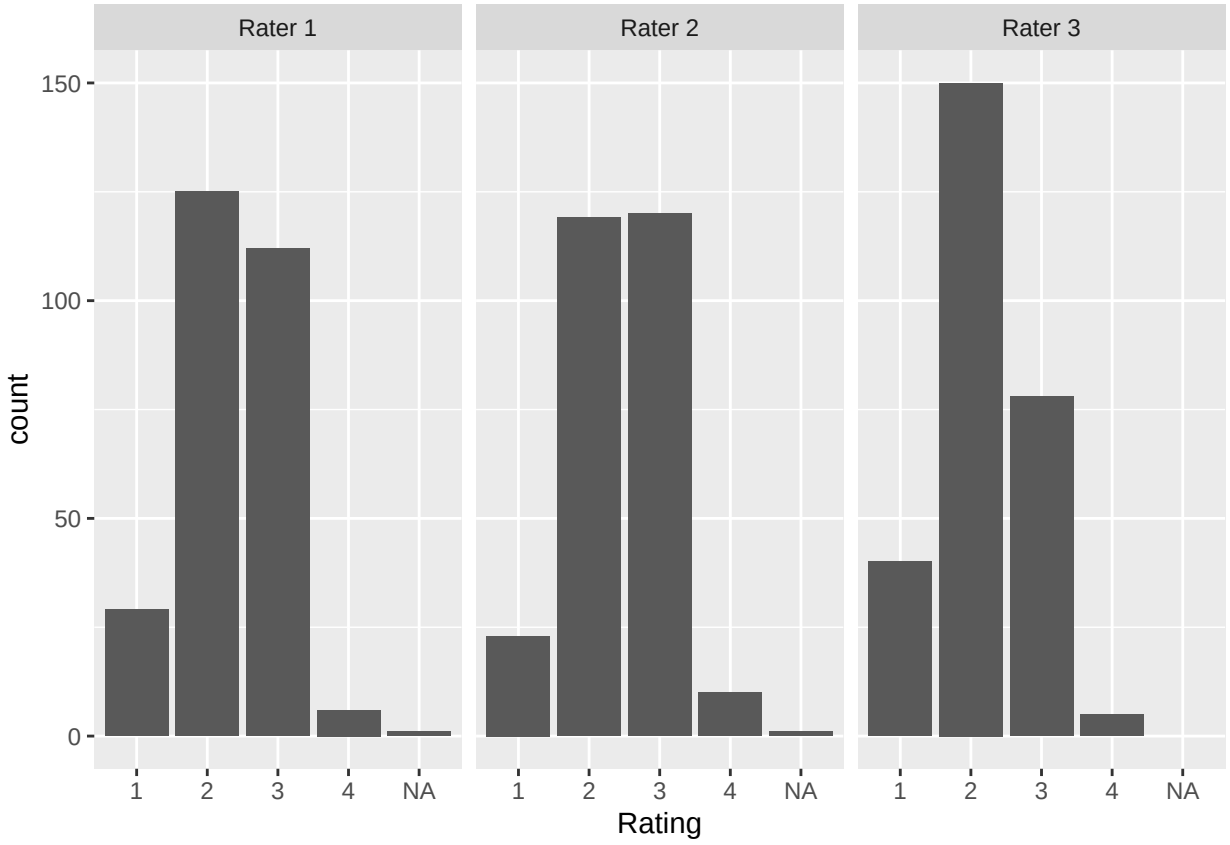
Figure 4: Bar plots of ratings count for each rater (full dataset)

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

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

```
  geom_bar()

g
```



Counts of rating for each rubric (full dataset)

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

	X	Rater	Sample	Overlap	Semester	Sex	RsrchQ	CritDes	InitEDA	SelMeth	InterpRes	VisOrg	T
5	5	3	5	NA	Fall	–	3	3	3	3	3	3	3

Missing Value in the dataset

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

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

```
ratings[ratings$Sex=="--",]%>%
  kbl() %>%
  kable_styling()
```

None of the missing values occur in the smaller 13-rubric data set so we don't have to worry about missing data at all in analyses that just involve this smaller data set.

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

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

2. Do rater's ratings reach a consensus?

```
Rubric.names <- sort(unique(tall$Rubric))
ICC.vec <- NULL
for (i in Rubric.names) {
  tmp <- lmer(as.numeric(Rating) ~ 1 + (1|Artifact), data=tall.13[tall.13$Rubric==i,])
  sig2 <- summary(tmp)$sigma^2
  tau2 <- attr(summary(tmp)$varcor[[1]],"stddev")^2
  ICC <- tau2 / (tau2 + sig2)
  ICC.vec <- c(ICC.vec,ICC)
}
names(ICC.vec) <- Rubric.names
agreement.results <- cbind(ICC.common=ICC.vec,"       a12"=0,a23=0,a13=0)
agreement.tables <- as.list(rep(NA,7))
names(agreement.tables) <- Rubric.names
for (i in Rubric.names) {
  r12 <- data.frame(r1=factor(ratings.13[ratings.13$Rater==1,i],levels=1:4),
                    r2=factor(ratings.13[ratings.13$Rater==2,i],levels=1:4),
                    a1=ratings.13[ratings.13$Rater==1,"Artifact"],
                    a2=ratings.13[ratings.13$Rater==2,"Artifact"])
  if(any(r12[,3]!=r12[,4])) { stop(paste("Rater 1-2 Artifact mismatch on rubric",i)) }
  a12 <- mean(r12[,1]==r12[,2])
  r12 <- table(r12[,1:2])  ## print this to see how much agreement there is among raters 1-2
```

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

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

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

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

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

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

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

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

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

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

round(agreement.results,2)%>%
  kbl() %>%
  kable_styling()
```

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

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

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

First, we try to add fixed effects to our seven rubric-specific models on reduced dataset

```
Rubric.names <- sort(unique(tall$Rubric))
model.formula.13 <- as.list(rep(NA,7))
names(model.formula.13) <- Rubric.names
for (i in Rubric.names) {

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

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

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

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

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

}
```

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

```
## $CritDes
## as.numeric(Rating) ~ (1 | Artifact)
##
## $InitEDA
```

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

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

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

Then, we try to add fixed effects to our seven rubric-specific models on the full dataset

```
Rubric.names <- sort(unique(tall$Rubric))
## eliminate missing data
tall[c(161,684),]
```

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

```
tall.nonmissing <- tall[-c(161,684),]
tall.nonmissing[tall.nonmissing$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
```

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

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

```
for (i in Rubric.names) {

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

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

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

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

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

}
```

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

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

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

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

1. final model for `SelMeth`

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

## now check to make sure we really need "Rater" as a factor...
tmp.single_intercept <- update(tmp, . ~ . + 1 - as.factor(Rater))
anova(tmp.single_intercept,tmp)
## now let's check for fixed-effect interactions... Since only Rater and Semester
## are involved, we only need to examine Rater*Semester
tmp.fixed_interactions <- update(tmp, . ~ . + as.factor(Rater)*Semester - Semester)
## I've specified the model so that I can see (a) a different intercept for each
## rater, and (b) a different semester effect for each rater.
anova(tmp,tmp.fixed_interactions)
## Looks like the fixed-effect interactions are not needed
```

```
## Testing (Semester|Artifact)...

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

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

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

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

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

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

for model mA is: there are more random effects than there are observations in the data set. Thus, the mode as.numeric(Rating) ~ -1 + as.factor(Rater) + Semester + (1 | Artifact) + (Semseter | Artifact) isn't even possible, so no testing is needed.

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

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

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

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

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

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

for model mA is: there are more random effects than there are observations in the data set. Thus, the mode as.numeric(Rating) ~ -1 + as.factor(Rater) + Semester + (1 | Artifact) + (as.factor(Rater)|Artifact)) isn't even possible, so no testing is needed.

```
summary(tmp)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ as.factor(Rater) + Semester + (1 | Artifact) -
##     1
##    Data: tall.nonmissing[tall.nonmissing$Rubric == "SelMeth", ]
##
```

```
## REML criterion at convergence: 143.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0480 -0.3923 -0.0551  0.2674  2.5827
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Artifact (Intercept) 0.08973  0.2996
##  Residual             0.10842  0.3293
## Number of obs: 116, groups:  Artifact, 90
##
## Fixed effects:
##                   Estimate Std. Error t value
## as.factor(Rater)1  2.25037    0.07503  29.992
## as.factor(Rater)2  2.22653    0.07424  29.991
## as.factor(Rater)3  2.03316    0.07521  27.033
## SemesterS19       -0.35860    0.09796  -3.661
##
## Correlation of Fixed Effects:
##             a.(R)1 a.(R)2 a.(R)3
## as.fctr(R)2  0.285
## as.fctr(R)3  0.287  0.280
## SemesterS19 -0.413 -0.391 -0.394
```

```
ranef(tmp)
```

```
## $Artifact
##     (Intercept)
## 100  0.33946601
## 101  0.04901108
## 102 -0.11338077
## 103  0.33946601
## 104 -0.11338077
## 105 -0.11338077
## 106 -0.11338077
## 107 -0.11338077
## 111  0.04901108
## 112 -0.11338077
## 113  0.04901108
## 114  0.04901108
## 115  0.04901108
## 116 -0.11338077
## 117 -0.11338077
## 118 -0.11338077
## 13  -0.46786343
## 15  -0.01501665
## 16  -0.01501665
## 17  -0.30547158
## 21   0.14737520
## 22  -0.01501665
## 23  -0.30547158
## 24  -0.01501665
## 25  -0.30547158
## 26   0.43783013
```

```
## 27  -0.01501665
## 28  -0.30547158
## 32  -0.01501665
## 33   0.43783013
## 34   0.43783013
## 35  -0.01501665
## 36  -0.01501665
## 37  -0.01501665
## 38  -0.01501665
## 39   0.14737520
## 40  -0.01501665
## 45   0.05980677
## 46   0.05980677
## 47  -0.39304001
## 48   0.35026170
## 49  -0.10258508
## 53   0.35026170
## 54  -0.10258508
## 55  -0.10258508
## 56  -0.10258508
## 57  -0.10258508
## 6   -0.01501665
## 61  -0.10258508
## 62   0.05980677
## 63   0.05980677
## 64  -0.10258508
## 65  -0.10258508
## 66   0.05980677
## 67  -0.10258508
## 68   0.05980677
## 7   -0.01501665
## 72   0.05980677
## 73  -0.10258508
## 74   0.35026170
## 75  -0.10258508
## 76  -0.10258508
## 77  -0.10258508
## 78   0.35026170
## 79   0.35026170
## 8    0.14737520
## 84   0.04901108
## 85  -0.11338077
## 86   0.04901108
## 87  -0.11338077
## 88   0.04901108
## 9    0.14737520
## 92  -0.11338077
## 93   0.04901108
## 94  -0.11338077
## 95   0.33946601
## 96  -0.11338077
## O1  -0.35883486
## O10 -0.12120653
## O11  0.13443559
```

```
## O12 -0.12120653
## O13 -0.12120653
## O2  -0.59646319
## O3  -0.12120653
## O4   0.59167847
## O5   0.35405014
## O6  -0.12120653
## O7   0.11642180
## O8  -0.10319274
## O9   0.13443559
##
## with conditional variances for "Artifact"
```

2.final model for `CritDes`

```
fla <- formula(model.formula.alldata[["CritDes"]])
tmp <- lmer(fla,data=tall.nonmissing[tall.nonmissing$Rubric=="CritDes",])
round(summary(tmp)$coef,2)  ## fixed effects and their t-values
```

```
##                   Estimate Std. Error t value
## as.factor(Rater)1     1.69       0.12   13.98
## as.factor(Rater)2     2.11       0.12   17.34
## as.factor(Rater)3     1.89       0.12   15.51
```

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

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

```
## Data: tall.nonmissing[tall.nonmissing$Rubric == "CritDes", ]
## Models:
## tmp.single_intercept: as.numeric(Rating) ~ (1 | Artifact)
## tmp: as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
##                      npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## tmp.single_intercept    3 277.68 285.91 -135.84   271.68
## tmp                     5 273.62 287.35 -131.81   263.62 8.0535  2    0.01783 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
## Error: number of observations (=115) <= number of random effects (=267) for term (as.factor(Rater) |
```

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

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

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

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

for model mA is: there are more random effects than there are observations in the data set. Thus, the mode as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1 + (as.factor(Rater)|Artifact)) isn't even possible, so no testing is needed.

```
summary(tmp)
```

```
## Linear mixed model fit by REML ['lmerMod']
```

```
## Formula: as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
##    Data: tall.nonmissing[tall.nonmissing$Rubric == "CritDes", ]
##
## REML criterion at convergence: 271
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.55495 -0.50027 -0.08228  0.64663  1.60935
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Artifact (Intercept) 0.4349   0.6595
##  Residual             0.2473   0.4972
## Number of obs: 115, groups:  Artifact, 89
##
## Fixed effects:
##                   Estimate Std. Error t value
## as.factor(Rater)1   1.6863     0.1207   13.98
## as.factor(Rater)2   2.1129     0.1219   17.34
## as.factor(Rater)3   1.8908     0.1219   15.51
##
## Correlation of Fixed Effects:
##             a.(R)1 a.(R)2
## as.fctr(R)2 0.244
## as.fctr(R)3 0.244  0.246
```

```
ranef(tmp)
```

```
## $Artifact
##     (Intercept)
## 100  0.83753019
## 101 -0.43756481
## 102 -0.43756481
## 103  0.19998269
## 104 -0.43756481
## 105 -0.43756481
## 106  0.19998269
## 107 -0.43756481
## 111 -0.43756481
## 112 -0.43756481
## 113 -0.43756481
## 114 -0.43756481
## 115 -0.43756481
## 116 -0.43756481
## 117 -0.43756481
## 118 -0.43756481
## 13   0.06962462
## 15   0.70717212
## 16   0.70717212
## 17   0.06962462
## 21   0.70717212
## 22   0.70717212
## 23  -0.56792288
## 24   0.06962462
## 25   0.70717212
```

```
## 26  -0.56792288
## 27   0.06962462
## 28  -0.56792288
## 32   0.70717212
## 33   0.06962462
## 34   0.70717212
## 35  -0.56792288
## 36   0.06962462
## 37   0.70717212
## 38   0.06962462
## 39  -0.56792288
## 40   0.06962462
## 46  -0.07196897
## 47   0.56557853
## 48   0.56557853
## 49  -0.70951647
## 53   1.20312602
## 54  -0.70951647
## 55  -0.07196897
## 56   0.56557853
## 57  -0.70951647
## 6   -0.56792288
## 61  -0.07196897
## 62   1.20312602
## 63   0.56557853
## 64   0.56557853
## 65   0.56557853
## 66   0.56557853
## 67  -0.70951647
## 68   0.56557853
## 7   -0.56792288
## 72  -0.07196897
## 73  -0.70951647
## 74  -0.70951647
## 75  -0.07196897
## 76  -0.07196897
## 77  -0.07196897
## 78   0.56557853
## 79  -0.07196897
## 8   -0.56792288
## 84   0.19998269
## 85   0.83753019
## 86   0.19998269
## 87   0.19998269
## 88   0.83753019
## 9   -0.56792288
## 92  -0.43756481
## 93  -0.43756481
## 94   0.83753019
## 95   0.19998269
## 96   0.19998269
## O1  -0.47358754
## O10 -0.47358754
## O11 -0.75381650
```

```
## O12 -0.19335859
## O13  0.08687037
## O2  -0.47358754
## O3   0.36709933
## O4   0.08687037
## O5   0.92755724
## O6  -0.75381650
## O7   0.36709933
## O8   0.08687037
## O9  -0.75381650
##
## with conditional variances for "Artifact"
```

3.final model for `InterpRes`

```
fla <- formula(model.formula.alldata[["InterpRes"]])
tmp <- lmer(fla,data=tall.nonmissing[tall.nonmissing$Rubric=="InterpRes",])
round(summary(tmp)$coef,2)  ## fixed effects and their t-values
```

```
##                   Estimate Std. Error t value
## as.factor(Rater)1     2.70       0.09   30.34
## as.factor(Rater)2     2.59       0.09   29.01
## as.factor(Rater)3     2.14       0.09   23.70
```

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

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

```
## Data: tall.nonmissing[tall.nonmissing$Rubric == "InterpRes", ]
## Models:
## tmp.single_intercept: as.numeric(Rating) ~ (1 | Artifact)
## tmp: as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
##                      npar    AIC    BIC   logLik deviance  Chisq Df Pr(>Chisq)
## tmp.single_intercept    3 218.53 226.79 -106.263   212.53
## tmp                     5 200.66 214.43  -95.331   190.66 21.864  2  1.787e-05
##
## tmp.single_intercept
## tmp                  ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

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

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

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

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

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

for model mA is: there are more random effects than there are observations in the data set. Thus, the mode as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1 + (as.factor(Rater)|Artifact)) isn't even possible, so no testing is needed.

```
summary(tmp)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
##    Data: tall.nonmissing[tall.nonmissing$Rubric == "InterpRes", ]
##
## REML criterion at convergence: 199.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5317 -0.7627  0.2635  0.6614  2.6535
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Artifact (Intercept) 0.06224  0.2495
##  Residual             0.25250  0.5025
## Number of obs: 116, groups:  Artifact, 90
##
## Fixed effects:
##                   Estimate Std. Error t value
## as.factor(Rater)1  2.70421    0.08912   30.34
## as.factor(Rater)2  2.58574    0.08912   29.01
## as.factor(Rater)3  2.13918    0.09027   23.70
##
## Correlation of Fixed Effects:
##             a.(R)1 a.(R)2
## as.fctr(R)2 0.061
## as.fctr(R)3 0.062  0.062
```

```
ranef(tmp)
```

```
## $Artifact
##     (Intercept)
## 100  0.05848973
## 101 -0.13925357
## 102  0.05848973
## 103  0.05848973
## 104  0.05848973
## 105 -0.13925357
## 106 -0.13925357
## 107  0.05848973
## 111 -0.13925357
## 112  0.05848973
## 113  0.05848973
## 114  0.05848973
## 115  0.05848973
## 116 -0.13925357
## 117  0.05848973
## 118  0.05848973
## 13  -0.22526567
## 15  -0.02752238
## 16   0.17022092
## 17  -0.02752238
## 21   0.17022092
```

```
## 22   0.17022092
## 23  -0.22526567
## 24  -0.02752238
## 25  -0.22526567
## 26  -0.02752238
## 27  -0.02752238
## 28  -0.22526567
## 32   0.17022092
## 33  -0.02752238
## 34   0.17022092
## 35  -0.02752238
## 36  -0.02752238
## 37  -0.02752238
## 38  -0.02752238
## 39  -0.02752238
## 40  -0.02752238
## 45  -0.11582665
## 46  -0.11582665
## 47  -0.11582665
## 48   0.08191665
## 49   0.08191665
## 53   0.08191665
## 54  -0.11582665
## 55   0.08191665
## 56  -0.11582665
## 57  -0.11582665
## 6   -0.02752238
## 61  -0.11582665
## 62   0.08191665
## 63   0.08191665
## 64   0.08191665
## 65  -0.31356994
## 66   0.08191665
## 67   0.08191665
## 68   0.08191665
## 7   -0.02752238
## 72   0.08191665
## 73  -0.11582665
## 74   0.08191665
## 75   0.08191665
## 76  -0.11582665
## 77   0.08191665
## 78   0.08191665
## 79   0.08191665
## 8   -0.02752238
## 84   0.05848973
## 85   0.05848973
## 86   0.05848973
## 87  -0.13925357
## 88   0.05848973
## 9   -0.02752238
## 92   0.05848973
## 93   0.05848973
## 94   0.05848973
```

```
## 95   0.05848973
## 96   0.05848973
## O1   0.08089221
## O10  0.08089221
## O11  0.22259425
## O12 -0.06080983
## O13 -0.20251187
## O2  -0.06080983
## O3   0.08089221
## O4   0.22259425
## O5   0.08089221
## O6  -0.20251187
## O7   0.08089221
## O8  -0.34421391
## O9   0.22259425
##
## with conditional variances for "Artifact"
```

4.final model for `VisOrg`

```
fla <- formula(model.formula.alldata[["VisOrg"]])
tmp <- lmer(fla,data=tall.nonmissing[tall.nonmissing$Rubric=="VisOrg",])
round(summary(tmp)$coef,2)  ## fixed effects and their t-values
```

```
##                   Estimate Std. Error t value
## as.factor(Rater)1     2.38        0.1   24.62
## as.factor(Rater)2     2.65        0.1   27.70
## as.factor(Rater)3     2.28        0.1   23.64
```

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

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

```
## Data: tall.nonmissing[tall.nonmissing$Rubric == "VisOrg", ]
## Models:
## tmp.single_intercept: as.numeric(Rating) ~ (1 | Artifact)
## tmp: as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
##                      npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## tmp.single_intercept    3 227.21 235.44 -110.60   221.21
## tmp                     5 220.82 234.54 -105.41   210.82 10.392  2   0.005539
##
## tmp.single_intercept
## tmp                  **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
## Error: number of observations (=115) <= number of random effects (=267) for term (as.factor(Rater) |
```

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

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

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

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

for model mA is: there are more random effects than there are observations in the data set. Thus, the mode as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1 + (as.factor(Rater)|Artifact)) isn't even possible, so no testing is needed.

```
summary(tmp)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
##    Data: tall.nonmissing[tall.nonmissing$Rubric == "VisOrg", ]
##
## REML criterion at convergence: 219.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.5004 -0.3365 -0.2483  0.3841  1.8552
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Artifact (Intercept) 0.2907   0.5392
##  Residual             0.1467   0.3830
## Number of obs: 115, groups:  Artifact, 89
##
## Fixed effects:
##                   Estimate Std. Error t value
## as.factor(Rater)1  2.37794    0.09658   24.62
## as.factor(Rater)2  2.64891    0.09564   27.70
## as.factor(Rater)3  2.28355    0.09658   23.64
##
## Correlation of Fixed Effects:
##             a.(R)1 a.(R)2
## as.fctr(R)2 0.263
## as.fctr(R)3 0.265  0.263
```

```
#random effect coefficient
ranef(tmp)
```

```
## $Artifact
##     (Intercept)
## 101  0.41341681
## 102 -0.25117695
## 103 -0.25117695
## 104 -0.25117695
## 105 -0.25117695
## 106 -0.25117695
## 107 -0.25117695
## 111 -0.25117695
## 112  0.41341681
## 113 -0.25117695
## 114 -0.25117695
## 115  0.41341681
## 116  0.41341681
## 117  1.07801056
```

```
## 118  0.41341681
## 13  -0.85303625
## 15   1.14074503
## 16   0.47615127
## 17  -0.18844249
## 21   0.47615127
## 22  -0.18844249
## 23  -0.85303625
## 24  -0.18844249
## 25  -0.18844249
## 26  -0.18844249
## 27  -0.18844249
## 28  -0.85303625
## 32   0.47615127
## 33  -0.18844249
## 34  -0.18844249
## 35   0.47615127
## 36  -0.18844249
## 37  -0.18844249
## 38   0.47615127
## 39  -0.18844249
## 40  -0.18844249
## 45  -0.43126354
## 46  -0.43126354
## 47  -1.09585730
## 48  -0.43126354
## 49   0.89792397
## 53   0.23333021
## 54  -0.43126354
## 55   0.23333021
## 56   0.23333021
## 57   0.23333021
## 6   -0.18844249
## 61   0.23333021
## 62   0.89792397
## 63   0.23333021
## 64   0.23333021
## 65   0.23333021
## 66   0.23333021
## 67   0.23333021
## 68   0.23333021
## 7   -0.18844249
## 72   0.23333021
## 73  -0.43126354
## 74   0.23333021
## 75  -0.43126354
## 76  -0.43126354
## 77   0.23333021
## 78   0.23333021
## 79   0.23333021
## 8   -0.18844249
## 84   0.41341681
## 85   0.41341681
## 86  -0.25117695
```

```
## 87  -0.25117695
## 88   1.07801056
## 9   -0.18844249
## 92  -0.25117695
## 93  -0.25117695
## 94   0.41341681
## 95  -0.25117695
## 96   0.41341681
## O1  -0.37389990
## O10 -0.08856703
## O11 -0.37389990
## O12 -0.37389990
## O13  0.19676584
## O2  -0.08856703
## O3   0.19676584
## O4  -0.08856703
## O5  -0.37389990
## O6  -0.08856703
## O7   0.48209871
## O8  -1.22989851
## O9   0.48209871
##
## with conditional variances for "Artifact"
```

5. final model for rurbic "InitEDA"

```
fla <- formula(model.formula.alldata[["InitEDA"]])
tmp <- lmer(fla,data=tall.nonmissing[tall.nonmissing$Rubric=="InitEDA",])
round(summary(tmp)$coef,2)
```

```
##             Estimate Std. Error t value
## (Intercept)     2.44       0.08    32.4
```

```
#random effect coefficient
ranef(tmp)
```

```
## $Artifact
##     (Intercept)
## 100 -0.30430226
## 101  0.38376223
## 102 -0.30430226
## 103  0.38376223
## 104  0.38376223
## 105 -0.30430226
## 106 -0.99236674
## 107 -0.30430226
## 111 -0.30430226
## 112  0.38376223
## 113 -0.99236674
## 114  0.38376223
## 115  0.38376223
## 116 -0.30430226
## 117 -0.30430226
## 118 -0.30430226
## 13  -0.99236674
## 15   0.38376223
```

```
## 16   1.07182672
## 17  -0.30430226
## 21   1.07182672
## 22   0.38376223
## 23  -0.99236674
## 24  -0.30430226
## 25  -0.30430226
## 26  -0.30430226
## 27   0.38376223
## 28  -0.99236674
## 32   0.38376223
## 33   0.38376223
## 34  -0.30430226
## 35  -0.30430226
## 36  -0.30430226
## 37  -0.30430226
## 38  -0.30430226
## 39   0.38376223
## 40   0.38376223
## 45  -0.30430226
## 46   0.38376223
## 47  -0.30430226
## 48   1.07182672
## 49   0.38376223
## 53   0.38376223
## 54   0.38376223
## 55  -0.30430226
## 56  -0.30430226
## 57  -0.30430226
## 6   -0.30430226
## 61   0.38376223
## 62   1.07182672
## 63   0.38376223
## 64  -0.30430226
## 65  -0.30430226
## 66   1.07182672
## 67   0.38376223
## 68  -0.30430226
## 7    0.38376223
## 72   0.38376223
## 73  -0.99236674
## 74   0.38376223
## 75   0.38376223
## 76  -0.30430226
## 77  -0.30430226
## 78   0.38376223
## 79   0.38376223
## 8   -0.30430226
## 84  -0.30430226
## 85   0.38376223
## 86  -0.30430226
## 87  -0.99236674
## 88   0.38376223
## 9   -0.30430226
```

```
## 92  -0.30430226
## 93  -0.30430226
## 94   1.07182672
## 95   0.38376223
## 96   0.38376223
## O1   0.48452197
## O10 -0.09462545
## O11 -0.38419916
## O12 -0.38419916
## O13  0.19494826
## O2  -0.09462545
## O3  -0.09462545
## O4   0.19494826
## O5  -0.38419916
## O6  -0.67377288
## O7   0.48452197
## O8  -0.38419916
## O9   0.48452197
##
## with conditional variances for "Artifact"
```

6. final model for rubric "RsrchQ"

```
fla <- formula(model.formula.alldata[["RsrchQ"]])
tmp <- lmer(fla,data=tall.nonmissing[tall.nonmissing$Rubric=="RsrchQ",])
round(summary(tmp)$coef,2)
```

```
##             Estimate Std. Error t value
## (Intercept)     2.35       0.06   40.59
```

```
#random effect coefficient
ranef(tmp)
```

```
## $Artifact
##      (Intercept)
## 100 -0.072903664
## 101 -0.280199221
## 102 -0.280199221
## 103 -0.072903664
## 104 -0.072903664
## 105 -0.072903664
## 106  0.134391893
## 107  0.134391893
## 111 -0.072903664
## 112  0.134391893
## 113 -0.072903664
## 114 -0.072903664
## 115  0.134391893
## 116 -0.072903664
## 117 -0.072903664
## 118 -0.072903664
## 13  -0.072903664
## 15  -0.072903664
## 16  -0.072903664
## 17   0.134391893
## 21   0.134391893
```

```
## 22   0.134391893
## 23  -0.072903664
## 24  -0.072903664
## 25  -0.072903664
## 26  -0.072903664
## 27  -0.072903664
## 28  -0.280199221
## 32   0.134391893
## 33  -0.072903664
## 34  -0.072903664
## 35  -0.072903664
## 36  -0.072903664
## 37  -0.072903664
## 38  -0.072903664
## 39   0.134391893
## 40  -0.072903664
## 45  -0.072903664
## 46  -0.072903664
## 47   0.134391893
## 48   0.134391893
## 49   0.134391893
## 53   0.134391893
## 54  -0.280199221
## 55   0.134391893
## 56  -0.072903664
## 57  -0.072903664
## 6   -0.072903664
## 61  -0.072903664
## 62   0.134391893
## 63   0.134391893
## 64  -0.072903664
## 65   0.134391893
## 66   0.134391893
## 67   0.134391893
## 68   0.134391893
## 7   -0.072903664
## 72  -0.072903664
## 73  -0.072903664
## 74  -0.072903664
## 75  -0.072903664
## 76  -0.072903664
## 77  -0.072903664
## 78   0.134391893
## 79   0.134391893
## 8   -0.072903664
## 84   0.134391893
## 85   0.341687450
## 86   0.134391893
## 87   0.134391893
## 88   0.134391893
## 9    0.134391893
## 92   0.134391893
## 93   0.134391893
## 94   0.134391893
```

```
## 95  -0.072903664
## 96   0.134391893
## O1  -0.008069777
## O10 -0.008069777
## O11  0.285012168
## O12 -0.154610749
## O13 -0.154610749
## O2  -0.154610749
## O3   0.138471195
## O4  -0.008069777
## O5   0.138471195
## O6  -0.154610749
## O7   0.138471195
## O8  -0.301151721
## O9  -0.154610749
##
## with conditional variances for "Artifact"
```

7.final model for rubric "TxtOrg"

```
fla <- formula(model.formula.alldata[["TxtOrg"]])
tmp <- lmer(fla,data=tall.nonmissing[tall.nonmissing$Rubric=="TxtOrg",])
round(summary(tmp)$coef,2)
```

```
##             Estimate Std. Error t value
## (Intercept)     2.59       0.07   37.93
```

```
#random effect coefficient
ranef(tmp)
```

```
## $Artifact
##     (Intercept)
## 100 -0.11247941
## 101  0.07899020
## 102 -0.30394902
## 103  0.07899020
## 104  0.07899020
## 105 -0.11247941
## 106  0.07899020
## 107 -0.11247941
## 111 -0.11247941
## 112  0.07899020
## 113 -0.11247941
## 114  0.07899020
## 115  0.07899020
## 116  0.07899020
## 117  0.07899020
## 118  0.07899020
## 13  -0.30394902
## 15   0.07899020
## 16   0.27045981
## 17  -0.11247941
## 21   0.27045981
## 22   0.07899020
## 23  -0.30394902
## 24   0.07899020
```

```
## 25  -0.11247941
## 26  -0.11247941
## 27   0.07899020
## 28  -0.30394902
## 32   0.07899020
## 33  -0.11247941
## 34  -0.11247941
## 35  -0.11247941
## 36   0.07899020
## 37   0.07899020
## 38  -0.11247941
## 39  -0.11247941
## 40  -0.11247941
## 45   0.07899020
## 46  -0.11247941
## 47  -0.30394902
## 48   0.27045981
## 49   0.07899020
## 53   0.07899020
## 54   0.07899020
## 55  -0.11247941
## 56  -0.11247941
## 57   0.07899020
## 6   -0.11247941
## 61   0.27045981
## 62   0.07899020
## 63   0.07899020
## 64   0.07899020
## 65   0.07899020
## 66  -0.11247941
## 67  -0.30394902
## 68   0.07899020
## 7   -0.11247941
## 72  -0.11247941
## 73   0.07899020
## 74  -0.11247941
## 75  -0.11247941
## 76  -0.11247941
## 77  -0.11247941
## 78   0.07899020
## 79   0.07899020
## 8   -0.11247941
## 84   0.07899020
## 85   0.07899020
## 86   0.07899020
## 87   0.07899020
## 88   0.07899020
## 9   -0.11247941
## 92   0.07899020
## 93   0.27045981
## 94   0.07899020
## 95   0.07899020
## 96   0.07899020
## 01  -0.10554956
```

```
## O10  0.03290165
## O11  0.17135286
## O12 -0.10554956
## O13  0.17135286
## O2  -0.10554956
## O3   0.17135286
## O4   0.17135286
## O5   0.03290165
## O6   0.03290165
## O7   0.17135286
## O8  -0.38245198
## O9   0.17135286
##
## with conditional variances for "Artifact"
```

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

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

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

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

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

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

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

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

summary(comb.full)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Sex + Repeated + Rubric
##    Data: tall.nonmissing
##
## REML criterion at convergence: 1429.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1091 -0.5065 -0.0178  0.5242  3.7932
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.55311  0.7437
##           RubricInitEDA   0.35239  0.5936   0.47
##           RubricInterpRes 0.17512  0.4185   0.23 0.75
##           RubricRsrchQ    0.16997  0.4123   0.58 0.44 0.71
##           RubricSelMeth   0.06816  0.2611   0.39 0.60 0.74 0.41
##           RubricTxtOrg    0.26339  0.5132   0.34 0.62 0.70 0.56 0.67
##           RubricVisOrg    0.25809  0.5080   0.35 0.73 0.68 0.52 0.41 0.76
##  Residual                 0.18916  0.4349
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
##                     Estimate Std. Error t value
## (Intercept)         2.013748   0.109103  18.457
## as.factor(Rater)2   0.001977   0.054887   0.036
## as.factor(Rater)3  -0.174867   0.055045  -3.177
## SemesterS19        -0.175017   0.087850  -1.992
## SexM                0.010506   0.081271   0.129
## Repeated           -0.073586   0.098522  -0.747
## RubricInitEDA       0.547054   0.095710   5.716
## RubricInterpRes     0.587091   0.100893   5.819
## RubricRsrchQ        0.460875   0.087516   5.266
## RubricSelMeth       0.164863   0.094265   1.749
## RubricTxtOrg        0.692880   0.099523   6.962
## RubricVisOrg        0.530182   0.099136   5.348
##
## Correlation of Fixed Effects:
##             (Intr) a.(R)2 a.(R)3 SmsS19 SexM   Repetd RbIEDA RbrcIR RbrcRQ
## as.fctr(R)2 -0.245
## as.fctr(R)3 -0.237  0.499
## SemesterS19 -0.361  0.008  0.000
## SexM        -0.398 -0.026 -0.035  0.302
## Repeated    -0.154  0.001 -0.003  0.079  0.009
## RubrcIntEDA -0.552 -0.001  0.000 -0.001  0.000  0.007
## RbrcIntrpRs -0.660 -0.001  0.000 -0.001  0.000 -0.009  0.734
```

```
## RubrcRsrchQ -0.626 -0.001  0.000 -0.001  0.000 -0.039  0.585  0.756
## RubricSlMth -0.689 -0.001  0.000 -0.001  0.000 -0.088  0.659  0.777  0.689
## RubrcTxtOrg -0.611 -0.001  0.000 -0.001  0.000  0.005  0.674  0.751  0.682
## RubricVsOrg -0.607 -0.001 -0.001 -0.002 -0.001 -0.021  0.715  0.745  0.668
##             RbrcSM RbrcTO
## as.fctr(R)2
## as.fctr(R)3
## SemesterS19
## SexM
## Repeated
## RubrcIntEDA
## RbrcIntrpRs
## RubrcRsrchQ
## RubricSlMth
## RubrcTxtOrg  0.725
## RubricVsOrg  0.680  0.750
```

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

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

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

```
## RubricVisOrg       0.5289027  0.0990973   5.337
##
## Correlation of Fixed Effects:
##             (Intr) a.(R)2 a.(R)3 SmsS19 RbIEDA RbrcIR RbrcRQ RbrcSM RbrcTO
## as.fctr(R)2 -0.281
## as.fctr(R)3 -0.277  0.499
## SemesterS19 -0.264  0.017  0.011
## RubrcIntEDA -0.610 -0.001  0.000 -0.002
## RbrcIntrpRs -0.735 -0.001  0.000  0.000  0.734
## RubrcRsrchQ -0.701 -0.001  0.000  0.002  0.586  0.756
## RubricSlMth -0.782  0.000  0.000  0.006  0.662  0.779  0.688
## RubrcTxtOrg -0.679 -0.001  0.000 -0.001  0.674  0.751  0.682  0.728
## RubricVsOrg -0.675 -0.001 -0.001  0.000  0.715  0.745  0.667  0.681  0.750
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
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| = 0.00431172 (tol = 0.002, component 1)
```

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

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

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

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Rubric + as.factor(Rater):Semester + as.factor(Rater):Rubric +
##     Semester:Rubric + as.factor(Rater):Semester:Rubric
##    Data: tall.nonmissing
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
## REML criterion at convergence: 1424.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9141 -0.5141 -0.0653  0.5023  3.6609
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.48550  0.6968
##           RubricInitEDA   0.35257  0.5938   0.42
##           RubricInterpRes 0.14619  0.3824   0.32 0.80
##           RubricRsrchQ    0.16444  0.4055   0.66 0.43 0.72
##           RubricSelMeth   0.06297  0.2509   0.45 0.64 0.78 0.49
##           RubricTxtOrg    0.25441  0.5044   0.44 0.65 0.67 0.60 0.62
##           RubricVisOrg    0.25527  0.5052   0.35 0.73 0.68 0.57 0.35 0.76
##  Residual                 0.18839  0.4340
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
```

```
##                                              Estimate Std. Error t value
## (Intercept)                                  1.739538   0.136568  12.738
## as.factor(Rater)2                            0.302995   0.155107   1.953
## as.factor(Rater)3                            0.237851   0.155863   1.526
## SemesterS19                                 -0.129077   0.250318  -0.516
## RubricInitEDA                                0.765215   0.165241   4.631
## RubricInterpRes                              0.979228   0.162160   6.039
## RubricRsrchQ                                 0.710427   0.147386   4.820
## RubricSelMeth                                0.462750   0.155274   2.980
## RubricTxtOrg                                 1.011251   0.160899   6.285
## RubricVisOrg                                 0.647869   0.166603   3.889
## as.factor(Rater)2:SemesterS19                0.268014   0.303883   0.882
## as.factor(Rater)3:SemesterS19               -0.072789   0.301026  -0.242
## as.factor(Rater)2:RubricInitEDA             -0.325018   0.204108  -1.592
## as.factor(Rater)3:RubricInitEDA             -0.374190   0.205354  -1.822
## as.factor(Rater)2:RubricInterpRes           -0.469281   0.201051  -2.334
## as.factor(Rater)3:RubricInterpRes           -0.711515   0.202316  -3.517
## as.factor(Rater)2:RubricRsrchQ              -0.447050   0.189326  -2.361
## as.factor(Rater)3:RubricRsrchQ              -0.474411   0.190681  -2.488
## as.factor(Rater)2:RubricSelMeth             -0.301450   0.193678  -1.556
## as.factor(Rater)3:RubricSelMeth             -0.365656   0.194970  -1.875
## as.factor(Rater)2:RubricTxtOrg              -0.449164   0.200927  -2.235
## as.factor(Rater)3:RubricTxtOrg              -0.407754   0.202209  -2.016
## as.factor(Rater)2:RubricVisOrg               0.009042   0.205059   0.044
## as.factor(Rater)3:RubricVisOrg              -0.287443   0.206299  -1.393
## SemesterS19:RubricInitEDA                   -0.050212   0.301475  -0.167
## SemesterS19:RubricInterpRes                  0.127813   0.295706   0.432
## SemesterS19:RubricRsrchQ                     0.133874   0.267750   0.500
## SemesterS19:RubricSelMeth                   -0.089616   0.282837  -0.317
## SemesterS19:RubricTxtOrg                     0.166097   0.293176   0.567
## SemesterS19:RubricVisOrg                     0.146845   0.302496   0.485
## as.factor(Rater)2:SemesterS19:RubricInitEDA  0.020326   0.392376   0.052
## as.factor(Rater)3:SemesterS19:RubricInitEDA  0.252422   0.389961   0.647
## as.factor(Rater)2:SemesterS19:RubricInterpRes -0.266618  0.385390  -0.692
## as.factor(Rater)3:SemesterS19:RubricInterpRes -0.152392  0.383354  -0.398
## as.factor(Rater)2:SemesterS19:RubricRsrchQ   -0.217348   0.360414  -0.603
## as.factor(Rater)3:SemesterS19:RubricRsrchQ    0.354319   0.357388   0.991
## as.factor(Rater)2:SemesterS19:RubricSelMeth  -0.401035   0.370200  -1.083
## as.factor(Rater)3:SemesterS19:RubricSelMeth  -0.192670   0.367887  -0.524
## as.factor(Rater)2:SemesterS19:RubricTxtOrg   -0.542267   0.385011  -1.408
## as.factor(Rater)3:SemesterS19:RubricTxtOrg   -0.316395   0.382614  -0.827
## as.factor(Rater)2:SemesterS19:RubricVisOrg   -0.603626   0.392909  -1.536
## as.factor(Rater)3:SemesterS19:RubricVisOrg   -0.186749   0.390759  -0.478
##
## Correlation matrix not shown by default, as p = 42 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
## optimizer (bobyqa) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

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

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

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

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

```
formula(comb.inter.u)
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues
```

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

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

```
## Data: tall.nonmissing
## Models:
```

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

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


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

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

```
## anova(m0,mA)     -- Not needed!
```

the "final" model appears to be

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

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

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

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

```
#all coefficients
summary(comb.final)$varcor
```

```
##  Groups     Name              Std.Dev. Corr
##  Artifact   RubricCritDes     0.70466
##             RubricInitEDA     0.56380   0.318
##             RubricInterpRes   0.31948   0.142  0.674
##             RubricRsrchQ      0.42313   0.500  0.194  0.538
##             RubricSelMeth     0.19554   0.145  0.227  0.376 -0.240
##             RubricTxtOrg      0.50030   0.269  0.437  0.364  0.305  0.213
##             RubricVisOrg      0.48204   0.175  0.504  0.445  0.276 -0.160
##  Artifact.1 as.factor(Rater)1 0.11317
##             as.factor(Rater)2 0.33429  -0.486
##             as.factor(Rater)3 0.30682   0.332  0.663
##  Residual                     0.36699
##
##
##
##
##
##
##
##   0.537
##
##
##
```

```
##
```

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

```
##                                    Estimate Std. Error    t value
## (Intercept)                       1.7575637 0.11404553 15.4110695
## as.factor(Rater)2                 0.3660357 0.13918435  2.6298628
## as.factor(Rater)3                 0.1959108 0.12966732  1.5108727
## SemesterS19                      -0.1591847 0.07647713 -2.0814681
## RubricInitEDA                     0.7394956 0.12995561  5.6903703
## RubricInterpRes                   0.9915251 0.12770499  7.7641838
## RubricRsrchQ                      0.7261970 0.11792700  6.1580215
## RubricSelMeth                     0.4106840 0.12470528  3.2932364
## RubricTxtOrg                      1.0157859 0.12999662  7.8139408
## RubricVisOrg                      0.6542619 0.13352531  4.8999093
## as.factor(Rater)2:RubricInitEDA  -0.2998031 0.15608617 -1.9207535
## as.factor(Rater)3:RubricInitEDA  -0.2947458 0.15634742 -1.8851978
## as.factor(Rater)2:RubricInterpRes -0.5132313 0.15348164 -3.3439265
## as.factor(Rater)3:RubricInterpRes -0.7148636 0.15363623 -4.6529623
## as.factor(Rater)2:RubricRsrchQ    -0.4874212 0.14721814 -3.3108771
## as.factor(Rater)3:RubricRsrchQ    -0.3224080 0.14726165 -2.1893547
## as.factor(Rater)2:RubricSelMeth   -0.3863728 0.15030727 -2.5705530
## as.factor(Rater)3:RubricSelMeth   -0.3871739 0.14961201 -2.5878532
## as.factor(Rater)2:RubricTxtOrg    -0.5510430 0.15646077 -3.5219242
## as.factor(Rater)3:RubricTxtOrg    -0.4449139 0.15673150 -2.8387015
## as.factor(Rater)2:RubricVisOrg    -0.1049002 0.15860588 -0.6613892
## as.factor(Rater)3:RubricVisOrg    -0.2752337 0.15884361 -1.7327340
```

```
ranef(comb.final)
```

```
## $Artifact
##     RubricCritDes RubricInitEDA RubricInterpRes RubricRsrchQ RubricSelMeth
## 100   0.799230984  -0.260928075    -0.121651475 -0.165004009   0.232286186
## 101  -0.496516992   0.434125016    -0.166202880 -0.741517647   0.032297023
## 102  -0.770758310  -0.332827494    -0.232327983 -0.744350171   0.048752382
## 103   0.139634991   0.330246495     0.098760531 -0.281560432   0.271528536
## 104  -0.576717711   0.308485599     0.091136083 -0.260861913   0.073419462
## 105  -0.590170519  -0.482765966    -0.340063870 -0.404241272  -0.097483786
## 106   0.204268619  -1.028871435    -0.399212804  0.233395507  -0.136338774
## 107  -0.559991953  -0.401625699     0.057176400  0.183109336  -0.102376401
## 111  -0.461664048  -0.351106160    -0.241480009 -0.311690005  -0.066960064
## 112  -0.499244181   0.322561462     0.271616561  0.197865302  -0.103842509
## 113  -0.586396232  -0.804699335    -0.145347161 -0.131243069   0.004115453
## 114  -0.448211240   0.440145405     0.189719944 -0.168310647   0.103943184
## 115  -0.370737710   0.454221268     0.370200422  0.290416568  -0.073318786
## 116  -0.588732781  -0.354356263    -0.287568101 -0.351805370  -0.123704956
## 117  -0.672282118  -0.137052102     0.011826585 -0.194894762  -0.104413152
## 118  -0.654154931  -0.212171980    -0.029490861 -0.209038369  -0.027874376
## 13    0.388153004  -0.746409972    -0.434651112  0.050673339  -0.158699671
## 15    0.688739627   0.476183778    -0.046817466 -0.110369427  -0.051999805
## 16    0.682661521   1.153733296     0.314175593 -0.008255305   0.186497414
## 17    0.335780462  -0.087897740     0.067097447  0.549575049  -0.199095708
## 21    0.776081195   1.090454720     0.451665435  0.434757500   0.085256164
## 22    0.730634647   0.381944249     0.248963623  0.431021584   0.090356721
## 23   -0.272110055  -0.723593827    -0.327021594 -0.010000648  -0.176402129
```

```
## 24    0.048984722  -0.202384494    -0.150347211 -0.130275726   0.068967797
## 25    1.160468165   0.007524296    -0.288482306  0.163053030  -0.130277171
## 26   -0.863455248  -0.484067019    -0.160119775 -0.518539461   0.112779994
## 27    0.101585126   0.386518690     0.006449881 -0.172787985   0.092129143
## 28   -0.298924176  -0.592075307    -0.413414923 -0.405043664  -0.059340280
## 32    0.660777592   0.404064894     0.250596452  0.407908975   0.001530719
## 33   -0.083986232   0.150259963    -0.059855689 -0.452407944   0.169464397
## 34    0.465648765  -0.311699143    -0.060532368 -0.201129132   0.261592479
## 35   -0.743838592  -0.254934345    -0.085356754 -0.283423105  -0.098124763
## 36    0.048984722  -0.202384494    -0.150347211 -0.130275726   0.068967797
## 37    0.775853335  -0.156960696    -0.206880216 -0.021631951   0.102490855
## 38   -0.016969980  -0.209510548    -0.141889759 -0.174779329  -0.064601706
## 39   -0.527961459   0.248569617     0.207297352  0.140190050  -0.087378665
## 40    0.105487479   0.357271991     0.013274504 -0.194178980   0.047385643
## 45    0.014461472  -0.241816467    -0.172353572 -0.089855423   0.049752153
## 46    0.149535808   0.324267199    -0.013510483 -0.141114723   0.032095536
## 47    1.130990266  -0.177526452    -0.048977060  0.677378540  -0.148529882
## 48    0.536946258   0.660736990     0.224637147  0.134903575   0.242019372
## 49   -0.903799572   0.215215414     0.273278530  0.159332994  -0.189828692
## 53    1.141878890   0.106667713     0.017170859  0.239121077   0.118008532
## 54   -0.662567949   0.383752151    -0.092019478 -0.662675251   0.147428187
## 55   -0.148638648  -0.372322971     0.070612548  0.317348100  -0.133690127
## 56    0.687866384  -0.264793628    -0.275901530 -0.060662995  -0.062064659
## 57   -0.769777000  -0.326398423    -0.169663060 -0.256562292  -0.084368178
## 6    -0.673981537  -0.277054990    -0.086989583 -0.260310496  -0.009298761
## 61    0.001532371   0.332813499    -0.079489581 -0.172105912   0.016048505
## 62    1.385468004   0.997826978     0.303484112  0.483127831  -0.052782108
## 63    0.682985544   0.348683798     0.207056600  0.445243639  -0.018947062
## 64    0.550928564  -0.162635081    -0.076518531  0.054799415   0.062449452
## 65    0.832053360  -0.452026132    -0.411562337  0.253229781  -0.216946994
## 66    0.741191311   0.910085082     0.371984780  0.382566905  -0.040124706
## 67   -0.816110956   0.144874267     0.292987057  0.146885024  -0.188108044
## 68    0.631067013  -0.239520783     0.050782606  0.488246987  -0.041946445
## 7    -0.621381133   0.311848194     0.069807510 -0.302822756   0.013862585
## 72   -0.056228922   0.416357005     0.192126550 -0.072224501   0.022357917
## 73   -0.746581386  -0.931337558    -0.323536901 -0.186660142  -0.017292859
## 74   -1.048691725   0.085932829     0.117133465 -0.493919747   0.092832965
## 75   -0.041429598   0.337827603     0.148256130 -0.088764859   0.098106237
## 76    0.037302455  -0.325732228    -0.216055049 -0.141550534  -0.005230230
## 77   -0.168462275  -0.233642423    -0.010418016 -0.072660313  -0.014967848
## 78    0.416200815   0.062463668     0.074617187  0.131334735   0.084768259
## 79   -0.309477259   0.018259622     0.132063515  0.023548394   0.051527986
## 8    -0.607375983  -0.208815047    -0.035893069 -0.212340708   0.006521839
## 84    0.308535786  -0.084130010     0.160710135  0.445177446  -0.061768185
## 85    1.073523595   0.376369806     0.315687800  0.876610792  -0.131954376
## 86    0.326662973  -0.159249888     0.119392688  0.431033838   0.014770591
## 87    0.204268619  -1.028871435    -0.399212804  0.233395507  -0.136338774
## 88    1.088302163   0.644193506     0.316426077  0.538722059  -0.077246235
## 9    -0.580561862  -0.340333567     0.050500260  0.182702309  -0.110540011
## 92   -0.540427028  -0.348335874     0.068354723  0.221401630  -0.052058795
## 93   -0.392355632  -0.163386243     0.178116906  0.352245190   0.028782533
## 94    1.037232913   1.033191279     0.338469161  0.394347249  -0.006476119
## 95    0.139634991   0.330246495     0.098760531 -0.281560432   0.271528536
## 96    0.239339349   0.379987642     0.224070665  0.314946243  -0.067536845
```

```
## O1   -0.501892690   0.422434324     0.120652157 -0.008571032  -0.092424623
## O10  -0.441626766  -0.001262869     0.155477124  0.049609057   0.036642815
## O11  -0.767188988  -0.330015401     0.328361447  0.512573093   0.013705453
## O12  -0.354477429  -0.488363858    -0.316209500 -0.338375175  -0.015431821
## O13   0.030644084   0.083176224    -0.316864835 -0.367378846  -0.071038038
## O2   -0.115914679   0.329833349     0.171406761  0.140224318  -0.072055784
## O3    0.283469723  -0.135261673    -0.013022636  0.231060493  -0.039721746
## O4   -0.111663933   0.044736443     0.203063448 -0.216904724   0.364965134
## O5    0.878776899  -0.426263751    -0.026178431  0.189700396   0.249925759
## O6   -0.897799288  -0.773107017    -0.419275496 -0.412772399  -0.112132493
## O7    0.137959776   0.206759645    -0.005685051 -0.013647771  -0.042656474
## O8    0.358903384  -0.209351926    -0.396345493 -0.310992705   0.029137961
## O9   -0.799401021   0.585026327     0.349338727 -0.190733068   0.074739959
##      RubricTxtOrg RubricVisOrg as.factor(Rater)1 as.factor(Rater)2
## 100 -0.4896794517  -0.44053275       0.034917710      -0.050162056
## 101  0.2892855198   0.46519565      -0.031016758       0.044558030
## 102 -1.1195238409  -0.43420405      -0.071689073       0.102987030
## 103  0.1091378602  -0.23982583       0.041745384      -0.059970550
## 104  0.0889452842  -0.11674446      -0.025609460       0.036790017
## 105 -0.5321954192  -0.35614616      -0.059252658       0.085121135
## 106  0.0172495838  -0.33471745      -0.022646958       0.032534148
## 107 -0.5132347471  -0.30992051      -0.011077825       0.015914173
## 111 -0.4110929451  -0.25284914      -0.035954838       0.051651971
## 112  0.2083790545   0.41584821       0.019749656      -0.028371945
## 113 -0.4729645571  -0.30641168      -0.016194729       0.023265010
## 114  0.2100477583  -0.01344744      -0.002311640       0.003320853
## 115  0.3294815285   0.51914522       0.043047476      -0.061841108
## 116  0.1298214349   0.27755828      -0.030950499       0.044462843
## 117  0.2257452041   0.84092875       0.010804646      -0.015521730
## 118  0.1275467704   0.31606001      -0.008665068       0.012448056
## 13  -0.7208211977  -0.62612179      -0.065721426      -0.387578433
## 15   0.4109514403   0.90099838       0.013933962       0.082172641
## 16   0.8984053089   0.58526073       0.020545356       0.121161961
## 17  -0.1152893314   0.03068501      -0.018002284      -0.106164727
## 21   0.9175852659   0.59123147       0.043489334       0.256469295
## 22   0.2103910968  -0.12223770       0.018241929       0.107577979
## 23  -0.6710014131  -0.56009900      -0.056910588      -0.335618351
## 24   0.2458312627  -0.13976280      -0.007464489      -0.044020270
## 25  -0.0006787891   0.07486396      -0.038417307      -0.226558075
## 26  -0.4702228800  -0.47170058       0.015720568       0.092708780
## 27   0.2991051601  -0.05309192      -0.006286522      -0.037073453
## 28  -0.6274133459  -0.51253038      -0.068987014      -0.406836561
## 32   0.2598401172   0.36104915       0.027323814       0.161136505
## 33  -0.4040007431  -0.39751311       0.018955249       0.111784639
## 34  -0.5024006366  -0.50576116       0.030229306       0.178271043
## 35  -0.2593711836   0.26603541      -0.004559319      -0.026887635
## 36   0.2458312627  -0.13976280      -0.007464489      -0.044020270
## 37   0.2587795022  -0.15224622      -0.005407775      -0.031891228
## 38  -0.2464229441   0.25355199      -0.002502605      -0.014758593
## 39  -0.2363663495  -0.12460982       0.010480741       0.061807990
## 40  -0.2425980671  -0.14306398      -0.010406523      -0.061370303
## 45   0.2993986757  -0.21577038       0.013962996      -0.084802723
## 46  -0.1861730460  -0.21912632       0.017900736      -0.108718148
## 47  -0.6836848125  -0.67088224       0.060699220      -0.368650037
```

```
## 48   0.6090159936  -0.35404619      -0.057901182       0.351656455
## 49   0.2596856040   0.72489009      -0.028110701       0.170727247
## 53   0.0732500452  -0.06260334      -0.066343698       0.402931150
## 54   0.3347924725  -0.11292581       0.048083364      -0.292029019
## 55  -0.3650699866   0.05821081      -0.010297844       0.062542823
## 56  -0.2393229797   0.11188885       0.019211671      -0.116679969
## 57   0.2764801825   0.22682469       0.019202169      -0.116622259
## 6   -0.3088202040  -0.21725144      -0.013641204      -0.080446161
## 61   0.8803935423   0.38756209       0.008571120      -0.052055753
## 62   0.4046355394   0.81917435      -0.054441080       0.330641908
## 63   0.2956831109   0.26743091      -0.036637725       0.222515194
## 64   0.2363878598   0.18601321      -0.001769927       0.010749459
## 65   0.3137018868   0.16068677       0.010826270      -0.065752159
## 66  -0.1911413293   0.26549866      -0.032385073       0.196687178
## 67  -0.8637207219   0.06971896      -0.003072252       0.018658987
## 68   0.2430517986   0.18130813      -0.034934997       0.212173860
## 7   -0.2555463066  -0.13058056      -0.012463237      -0.073499344
## 72  -0.2054231378   0.24600407      -0.010253168       0.062271488
## 73   0.1793540491  -0.33835915       0.034032584      -0.206693152
## 74  -0.4515087937  -0.05714937      -0.034013262       0.206575802
## 75  -0.3067576966  -0.28153970       0.018583351      -0.112863937
## 76  -0.2956440959  -0.35373181       0.035312297      -0.214465348
## 77  -0.3148941877   0.11139859       0.007158393      -0.043475712
## 78   0.0614237501  -0.04916294      -0.063370758       0.384875329
## 79   0.0495974550  -0.03572253      -0.060397819       0.366819509
## 8   -0.2460521798  -0.16371208      -0.002773652      -0.016357037
## 84   0.2938727745   0.42401430       0.044931330      -0.064547413
## 85   0.2775907778   0.41743883       0.054457410      -0.078232381
## 86   0.1956743408  -0.10085444       0.025461616      -0.036577627
## 87   0.0172495838  -0.33471745      -0.022646958       0.032534148
## 88   0.4756563489   1.03788067       0.071335543      -0.102479155
## 9   -0.2896402470  -0.21128070       0.009302774       0.054861173
## 92   0.0505836734  -0.20108481      -0.002245380       0.003225667
## 93   0.7355045680   0.01104792       0.029884885      -0.042932003
## 94   0.3159523886   0.50177918       0.031093330      -0.044668031
## 95   0.1091378602  -0.23982583       0.041745384      -0.059970550
## 96   0.2323672478   0.41278156       0.024158832      -0.034706075
## O1  -0.2366040843  -0.25851406      -0.212387375       0.271897172
## O10  0.1214338624   0.01134540       0.108499191      -0.367908961
## O11  0.3056882685  -0.26220621       0.052727694       0.052427066
## O12 -0.3628349814  -0.45357712       0.058313167      -0.016889682
## O13  0.2747051291   0.14796597      -0.008184147       0.048581903
## O2   0.1740844778   0.34229023       0.172115212      -1.028634929
## O3   0.2431033934   0.13200460      -0.038540523       0.159671322
## O4   0.1611929225  -0.27835154      -0.089483142       0.418600580
## O5  -0.0424788260  -0.48865981       0.030687678       0.059849944
## O6  -0.0545902931  -0.18323806      -0.050132631       0.116067534
## O7   0.1229928756   0.22420616       0.147788227       0.139018259
## O8  -0.5684692847  -0.90878758      -0.039480476      -0.248710070
## O9   0.3976521629   0.55918498       0.048176486       0.035307667
##     as.factor(Rater)3
## 100       0.031425105
## 101      -0.027914341
## 102      -0.064518452
```

```
## 103       0.037569848
## 104      -0.023047902
## 105      -0.053325976
## 106      -0.020381721
## 107      -0.009969778
## 111      -0.032358495
## 112       0.017774218
## 113      -0.014574869
## 114      -0.002080420
## 115       0.038741700
## 116      -0.027854709
## 117       0.009723923
## 118      -0.007798354
## 13       -0.536738118
## 15        0.113796809
## 16        0.167791181
## 17       -0.147022256
## 21        0.355171586
## 22        0.148979399
## 23       -0.464781182
## 24       -0.060961485
## 25       -0.313749024
## 26        0.128387784
## 27       -0.051341183
## 28       -0.563407744
## 32        0.223149941
## 33        0.154804993
## 34        0.246878712
## 35       -0.037235349
## 36       -0.060961485
## 37       -0.044164578
## 38       -0.020438442
## 39        0.085594815
## 40       -0.084988683
## 45       -0.051586096
## 46       -0.066134018
## 47       -0.224252423
## 48        0.213915107
## 49        0.103854591
## 53        0.245105867
## 54       -0.177643316
## 55        0.038045241
## 56       -0.070977250
## 57       -0.070942144
## 6        -0.111405892
## 61       -0.031665883
## 62        0.201131810
## 63        0.135357566
## 64        0.006538972
## 65       -0.039997503
## 66        0.119646201
## 67        0.011350394
## 68        0.129066859
## 7        -0.101785590
```

```
## 72        0.037880186
## 73       -0.125732906
## 74        0.125661521
## 75       -0.068655930
## 76       -0.130460787
## 77       -0.026446583
## 78        0.234122383
## 79        0.223138900
## 8        -0.022652049
## 84        0.040437123
## 85        0.049010367
## 86        0.022914846
## 87       -0.020381721
## 88        0.064200283
## 9         0.075974513
## 92       -0.002020789
## 93        0.026895682
## 94        0.027983254
## 95        0.037569848
## 96        0.021742371
## O1       -0.224054358
## O10      -0.112457996
## O11       0.174457620
## O12       0.118751458
## O13       0.029122880
## O2       -0.619341812
## O3        0.068666630
## O4        0.206869172
## O5        0.130604142
## O6       -0.001472421
## O7        0.481124968
## O8       -0.338168896
## O9        0.146920141
##
## with conditional variances for "Artifact"
```

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

```
ggplot(tall.nonmissing, aes(x=Rating)) +
  geom_bar() +
  facet_wrap( ~ Rubric + Rater, nrow=7)+
  ggtitle("Facet plot of counts of ratings partitioned by Rubric and Rater")+
  theme(plot.title = element_text(size = 60, face = "bold"))
```


```
ggplot(tall.nonmissing, aes(x=Rating)) +
  geom_bar() +
  facet_wrap( ~ Rubric + Sex, nrow=7)+
  ggtitle("Facet plot of counts of ratings partitioned by Rubric and Sex")+
  theme(plot.title = element_text(size = 60, face = "bold"))
```


```
ggplot(tall.nonmissing, aes(x=Rating)) +
  geom_bar() +
  facet_wrap( ~ Rubric + Semester, nrow=7)+
  ggtitle("Facet plot of counts of ratings partitioned by Rubric and Semester")+
  theme(plot.title = element_text(size = 60, face = "bold"))
```

Facet plot of co