# Effects of Various Factors on Ratings of Freshman Statistics Projects in New “General Education” Program

Olivia Wang
ziyanw2@andrew.cmu.edu

14 November 2021

**Abstract**

This paper will investigate the effects of various factors on ratings of Freshman Statistics projects in new “General Education” program. We examine data for this study come from a recent experiment of the new “General Education” program with rating work in Freshman Statistics using exploratory data analyses. We calculated the intraclass correlation (ICC), made a 2-way table of counts, and started with the “combined” intercept-only model by adding fixed effects, fixed-effect interactions, and new random effects, random-effect interactions to get the final "combined" model using all the data. It appears that the rating is highly influenced by three fixed effects Rater, Semester, Rubric; two random effects Rubric and Rater; and one fixed-effect interaction Rubric and Rater. We did model diagnostics by plotting four types of residuals for the final “combined” model to find something interesting.

## 1. Introduction

General education is very important in college, because it can provide students with the foundation need to become highly intelligent in chosen field of study, and in life after college. To emphasize the importance of general education for undergraduates, Dietrich College at Carnegie Mellon University is in the process of implementing a new “General Education” program for undergraduates. This program specifies a set of courses and experiences that all undergraduates must take, and in order to determine whether the new program is successful, the college hopes to rate student work performed in each of the “General Education” courses each year. In this report, we investigate the relationship between various factors (Rater, Semester, Sex, Repeated, Rubric) and ratings of Freshman Statistics projects in new “General Education” program.

In particular, this paper will address the following research questions:

- 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?
- 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?
- 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?
- Is there anything else interesting to say about this data?

## 2. Data

The data for this study come from a recent experiment of the new “General Education” program with rating work in Freshman Statistics. The data are given in the two files ratings.csv and tall.csv. While ratings.csv contains variables available for analysis are defined in Table 1. The file ratings.csv contains data organized exactly as in Table 1. Another file tall.csv contains the same data but organized so that each row contains just one rating, in the column labelled Rating, and the rubric for that rating is listed in the column labelled Rubric.

The additional background information of experiment with rating work in Freshman Statistics, rubrics for rating Freshman Statistics projects, and rating scale used for all rubrics are presented in Appendix 0, p. 4: Data Background.

The variables in the ratings.csv data set are shown in Table 1.

| 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 1: Variable Definitions for the ratings.csv data set.

| | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. | SD |
|---|---|---|---|---|---|---|---|
| RsrchQ | 1 | 2 | 2 | 2.36 | 3.0 | 4 | 0.60 |
| CritDes | 1 | 1 | 2 | 1.86 | 2.5 | 4 | 0.84 |
| InitEDA | 1 | 2 | 2 | 2.44 | 3.0 | 4 | 0.70 |
| SelMeth | 1 | 2 | 2 | 2.06 | 2.0 | 3 | 0.48 |
| InterpRes | 1 | 2 | 3 | 2.49 | 3.0 | 4 | 0.61 |
| VisOrg | 1 | 2 | 2 | 2.42 | 3.0 | 4 | 0.68 |
| TxtOrg | 1 | 2 | 3 | 2.60 | 3.0 | 4 | 0.70 |

Table 2: Summary Statistics for ratings of each rubric based on whole set of 91 artifacts.

| | **Min.** | **1st Qu.** | **Median** | **Mean** | **3rd Qu.** | **Max.** | **SD** |
|---|---|---|---|---|---|---|---|
| Rater 1 | 1 | 2 | 2 | 2.35 | 3 | 4 | 0.7 |
| Rater 2 | 1 | 2 | 2 | 2.43 | 3 | 4 | 0.7 |
| Rater 3 | 1 | 2 | 2 | 2.18 | 3 | 4 | 0.69 |

Table 3: Summary Statistics for ratings given by each rater based on whole set of 91 artifacts.

There are total of 117 observations of 15 variables collected in ratings.csv data set. Among all the variables, there are 5 continuous variables, 3 categorical variables and 7 factorial variables. Based on this data set, there are total of 91 project papers referred to as “artifacts”. Thirteen of the 91 artifacts were rated by all three raters; each of the remaining 78 artifacts were rated by only rater. Besides there are four variables (Overlap, Sex, CritDes, and VisOrg) appear to have some missing values in the data set. Details of analyses for missing values of ratings.csv in R can be found in Appendix 1, Part B, p. 4.

While there are total of 819 observations of 8 variables collected in tall.csv data set, which contains the same data but organized different from ratings.csv. Among all the variables, there are 3 continuous variables, 4 categorical variables and 1 factorial variable. In addition, there are two variables (Rating, Sex) appear to have some missing values in the data set. Details of analyses for missing values of tall.csv in R can be found in Appendix 1, Part C, p. 4.

One-dimensional summary statistics for ratings of each rubric based on whole set of 91 Freshman Statistics projects are given in Table 2. Table 3 shows the summary statistics for ratings given by each rater based on whole set of 91 artifacts.

## 3. Methods

### 3.1.1 Is the distribution of ratings for each rubric pretty much indistinguishable from the other rubrics, or are there rubrics that tend to get especially high or low ratings?

For this question, we mainly focused on tall.csv data set, except for Section 3.1.1, Part 2, where we used ratings.csv data set. Our analysis contains of four parts:

- First, we set up an assumption to define low/high rating.
- Second, we relied on one-dimensional summary statistics for the subset of the ratings.csv data set with only ratings for seven rubric variables based on all artifacts to compare the numeric summaries of ratings for each rubric.
- Then, we made visual comparison of bar plots of ratings for each rubric based on all artifacts to capture univariate distributions of ratings for each rubric.
- Last, we investigated table of counts of ratings for each rubric based on all artifacts to compare the total number of high/low ratings for each rubric. Details of these analyses in R can be found in Appendix 2, Part A, p. 6-9.

### 3.1.2 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?

For this question, we focused on tall.csv data set. Our analysis contains of four parts:

- First, we used the same assumption to define low/high rating as Section 3.1.1.
- Second, we relied on one-dimensional summary statistics for the subset of the tall.csv data set with only ratings given by three raters based on all artifacts to compare the numeric summaries of ratings given by each rater.
- Then, we made visual comparison of bar plots of ratings given by each rater based on all artifacts to capture univariate distributions of ratings given by each rater.
- Last, we investigated table of counts of ratings given by each rater based on all artifacts to compare the total number of high/low ratings for each rubric. Details of these analyses in R can be found in Appendix 2, Part B, p. 6-9.

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

For this question, mainly focused on the subset of the data for just the 13 artifacts seen by all three raters, except for Section 3.2, Part 3, where we used tall.csv data set. We considered two parts:

- **Measure of agreement among the raters:** we treated each artifact as a cluster of three ratings, and fitted seven random-intercept models, one for each rubric, and calculated the seven intraclass correlations (ICC's), which is the common correlation among the raters' ratings for each artifact. Details of these analyses in R can be found in Appendix 3, p. 6-9.
- **Find raters might be contributing to disagreement:** we made a 2-way table of counts for the ratings of each pair of raters, on each rubric (since there are three pairs of raters, each rubric will get three tables). For each table, the percentage of observations on the main diagonal is the percent exact agreement between the two raters. The tables, and the percent exact agreement from each table, can help to determine who is agreeing with whom on each rubric. Details of these analyses in R can be found in Appendix 3, p. 6-9.

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

To explore this question, we focused on tall.csv data set. We split it into four parts:

- **Added fixed effects to the seven rubric-specific models using just the data from the 13 common artifacts that all three raters saw:** 1) We added fixed effects for Rater, Semester, Sex and/or Repeated to the seven random intercept models, one for each rubric, for the 13 common artifacts that all three raters saw; 2) We used R package LMERConvenienceFuctions automated backwards selection of fixed effects; 3) Finally, used ANOVA tests based on AIC, BIC and likelihood ratio tests (LRT) to get the final seven rubric-specific models by adding fixed effects using just the data from the 13 common artifacts that all three raters saw. Details of these analyses in R can be found in Appendix 4, Part A, p. 6-9.
- **Added fixed effects to the seven rubric-specific models using all the data:** 1) We eliminated by hand the two observations with missing data from tall.csv data set and only do fitting and comparison on this "slightly" reduced data set; 2) We added fixed effects for Rater, Semester, Sex and/or Repeated to the seven random intercept models, one for each rubric, for the full data set; 3) We used R package LMERConvenienceFuctions automated backwards selection of fixed effects; 4) Finally, used ANOVA tests based on AIC, BIC and likelihood ratio tests (LRT) to get the final seven rubric-specific models by adding fixed effects using all the data. Details of these analyses in R can be found in Appendix 4, Part B, p. 6-9.
- **Tried interactions and new random effects for the seven rubric specific models using all the data:** We refitted the seven random intercept models, one for each rubric, for the full data set from Section 3.3, Part 2 by adding fixed-effect interactions and new random effects for the seven rubric specific models using all the data: 1) We checked on the t-statistics to make sure whether all the variables matter; 2) We checked for fixed-effect interactions for the seven rubric specific models; 3) We added random effects that are also present as fixed effects for the seven rubric specific models; 4) We used R package LMERConvenienceFuctions automated backwards selection of fixed-effect interactions and forward selection of random effects; 5) Finally, used ANOVA tests based on AIC, BIC and likelihood ratio tests (LRT) and used R package RLRsim to perform simulation-based exact LRT for random effects to get the final seven rubric-specific models by adding fixed-effect interactions and new random effects using all the data. Details of these analyses in R can be found in Appendix 4, Part C, p. 6-9.
- **Tried to add fixed effects, interactions, and new random effects to the "combined" model Rating ~ 1 + (0 + Rubric | Artifact), using all the data:** 1) We started with the "combined" intercept-only model; 2) Added fixed effects and fixed-effect interactions for all of the variables Rater, Semester, Sex, Repeated and/or Rubric; 3) We considered adding random effects and

random-effect interactions by adding each of these terms without a random intercept, to preserve the structure of the model (separate random intercepts for each rubric); 4) We used R package LMERConvenienceFuctions automated backwards selection of fixed effects, fixed-effect interactions and forward selection of random effects, random-effect interactions; 5) Finally, used ANOVA tests based on AIC, BIC and likelihood ratio tests (LRT) to get the final "combined" model by adding fixed effects, fixed-effect interactions, and new random effects, random-effect interactions. Details of these analyses in R can be found in Appendix 4, Part D, p. 6-9.

### 3.4 Is there anything else interesting to say about this data?

Lastly, we illustrated on other things we could say about our analysis, that will be of interest to the associate dean and something that is interesting and useful to the college. For the final "combined" model, we illustrated residual diagnostic plots might be interesting.

## 4. Results

### 4.1.1 Is the distribution of ratings for each rubric pretty much indistinguishable from the other rubrics, or are there rubrics that tend to get especially high or low ratings?

- First, we set up an assumption to define low/high rating. Here, among all rubrics, we defined low rating as artifact was rated less and equal to 2; high rating as artifact was rated above 2.
- As shown in Table 2, which is the one-dimensional summary statistics for the subset of the ratings.csv data set with only ratings for seven rubric variables, we found that rubric TxtOrg gets the highest values (Max, Min, Median, Mean, 1st Quartile, 3rd Quartile) of ratings except for standard deviation among all the rubrics. In addition, both of the max value and standard deviation of ratings on rubric SelMeth are the lowest ones among all rubrics for rating Freshman Statistics projects (Appendix 2, Part A, p. 6-9: Detail analyses of distribution of ratings on rubric SelMeth).
- Then, we made visual comparison of bar plots of ratings for each rubric based on all artifacts. Figure 1 gives the univariate distributions of ratings for each rubric, we clearly found that there are over 70 artifacts were rated as high ratings for rubric TxtOrg which is the highest number of artifacts were rated as high ratings among all the rubrics; there are nearly 100 artifacts were rated as low ratings which is the highest number of artifacts were rated as low ratings among all the rubrics for rating Freshman Statistics projects.
- Last, we investigated table of counts of ratings for each rubric based on all artifacts as shown in Table 4, the total number of high ratings for rubric TxtOrg is 72 which is the highest one among all the rubrics; and the total number of low ratings for rubric SelMeth is 99 which is the highest one among all the rubrics for rating Freshman Statistics projects.

Hence, we concluded that rubric Text Organization tends to get especially high ratings, and rubric Select Method(s) tends to get especially low ratings.

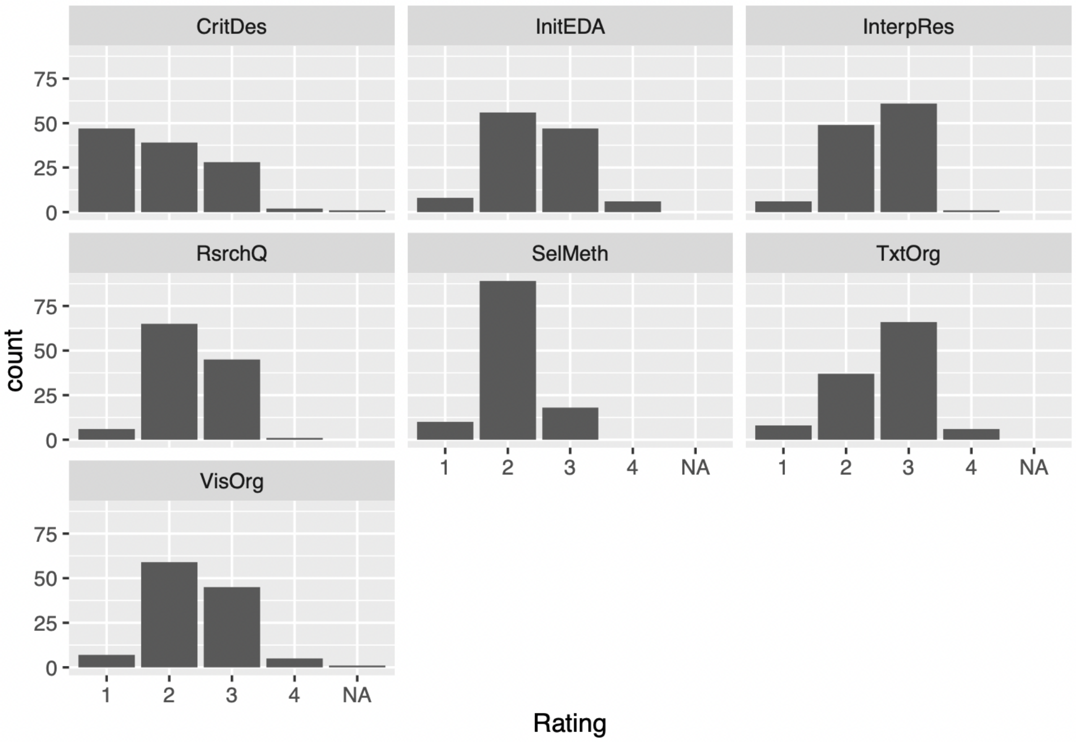


Figure 1: Bar plots of ratings for each rubric based on whole set of 91 artifacts.

```
##            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
```

Table 4: Counts of ratings for each rubric based on whole set of 91 artifacts.

### 4.1.2 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?

- First, we used the same assumption to define low/high rating as Section 4.1.1.
- As shown in Table 3, which is the one-dimensional summary statistics for the subset of the tall.csv data set with only ratings given by three raters based on all artifacts, we found that all of the values (Max, Min, Median, Mean, 1st Quartile, 3rd Quartile, Standard Deviation, etc.) of ratings given by three raters are very similar.
- Then, we made visual comparison of bar plots of ratings given by each rater based on all artifacts to capture univariate distributions of ratings given by each rater. Figure 2 gives the univariate distributions of ratings given by each rater, we clearly found that Rater 2 tends to give higher ratings (Appendix 2, Part B, p. 6-9). But there is no significant phenomenon from distributions of ratings given by Rater 1 and Rater 2 that we can conclude which rater tends to give especially high ratings. Because both of distributions of ratings given by Rater 1 and Rater 2 are very similar. Rater 3 tends to give especially low ratings (Appendix 2, Part B, p. 6-9).

- Last, we investigated table of counts of ratings given by each rater based on all artifacts as shown in Table 5, the total number of high ratings given by Rater 2 is 130 which is the highest one among all the ratings given by each rater, and the total number of low ratings given by Rater 3 is 190 which is the highest one among all the ratings given by each rater for rating Freshman Statistics projects.

Hence, we concluded that Rater 3 tends to give especially low ratings, and Rater 2 tends to give higher ratings, but there is no rater tends to give especially high ratings.

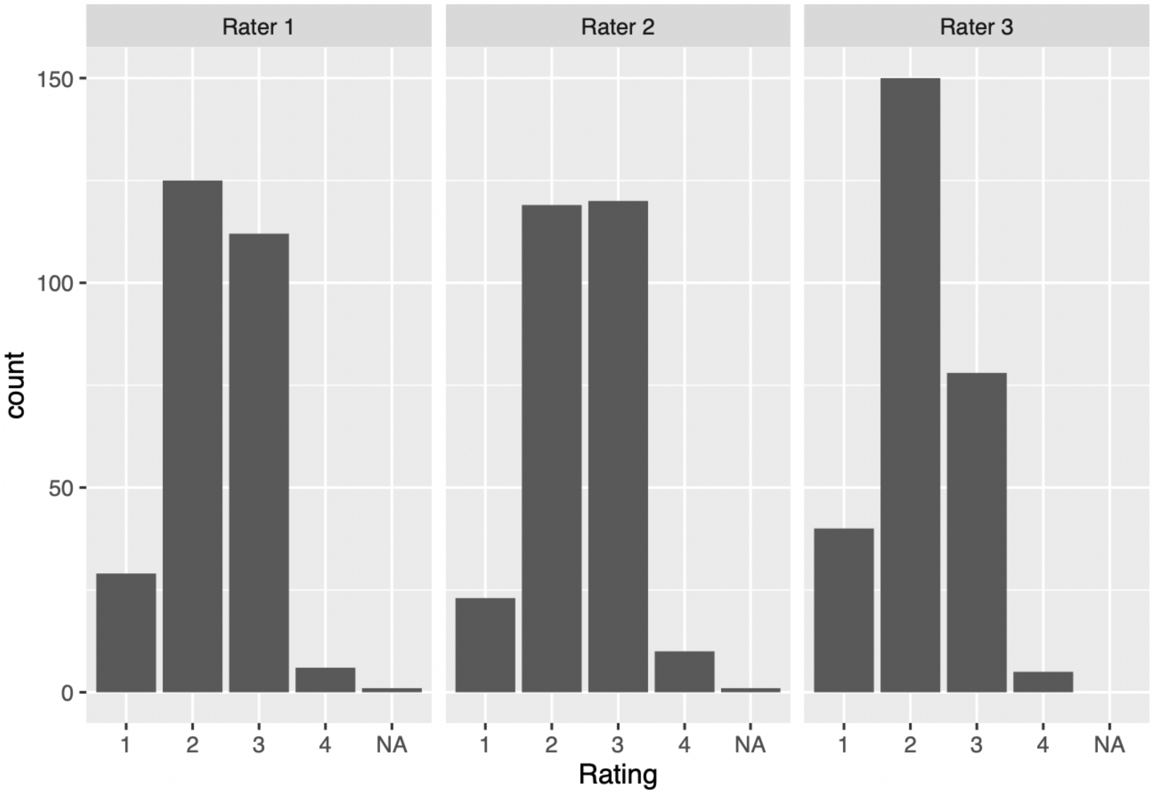
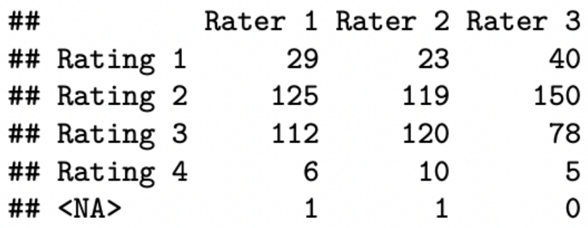


Figure 2: Bar plots of ratings given by each rater based on whole set of 91 artifacts.



```
##           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
```

Table 5: Counts of ratings for each rubric based on whole set of 91 artifacts.

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

### Measure of agreement among the raters

One measure of agreement among the raters is the intraclass correlation (ICC); it is the common correlation among the raters' ratings for each artifact. The ICC's can help us determine whether the raters are generally in agreement (high ICC = high correlation among the raters) or not (low ICC = low correlation among the raters) on each rubric. Based on the output of the seven intraclass correlations (ICC's) (Appendix 3, p. 6-9). In this case, we concluded that raters generally are consistent with one another in how they rate on rubrics CritDes, SelMeth and VisOrg. However, raters generally are not consistent with one another in how they rate on rubrics RsrchQ, InitEDA, InterpRes and TxtOrg. Details of these analyses in R can be found in Appendix 3, p. 6-9.

### Find raters might be contributing to disagreement

The ICC's can help us determine whether the raters are generally in agreement, but they cannot tell us which raters might be contributing to disagreement. So we made a 2-way table of counts for the ratings of each pair of raters, on rubrics RsrchQ, InitEDA, InterpRes and TxtOrg. Based on the percent exact agreement from each table (Appendix 3, p. 6-9), we concluded that Rater 1 and Rater 2 might be contributing to disagreement on their scores on rubric RsrchQ; Rater 1 and Rater 3 might be contributing to disagreement on their scores on rubrics InitEDA and InterpRes; Rater 2 and Rater 3 might be contributing to disagreement on their scores on rubric TxtOrg. Details of these analyses in R can be found in Appendix 3, p. 6-9.

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

### Added fixed effects to the seven rubric-specific models using just the data from the 13 common artifacts that all three raters saw

We added fixed effects for Rater, Semester, Sex and/or Repeated to the seven random intercept models, one for each rubric, for the 13 common artifacts that all three raters saw. The final seven rubric-specific models by adding fixed effects using just the data from the 13 common artifacts that all three raters saw (Appendix 4, Part C, p. 6-9) are only the seven random-intercept models, without any fixed effects or interactions to the models for each rubric. Details of these analyses in R can be found in Appendix 4, Part C, p. 6-9.

### Added fixed effects to the seven rubric-specific models using all the data

We added fixed effects for Rater, Semester, Sex and/or Repeated to the seven random intercept models, one for each rubric, for the full data set. Now we see there are some differences among the final seven rubric-specific models by adding fixed effects using all the data (Appendix 4, Part C, p. 6-9). For rubrics InitEDA, RsrchQ and TxtOrg, the models are just the simple random-intercept models. For the other four, the models are a little more complex. Details of these analyses in R can be found in Appendix 4, Part C, p. 6-9.

### Tried interactions and new random effects for the seven rubric specific models using all the data

We refitted the seven random intercept models, one for each rubric, for the full data set from Section 4.3, Part 2 by adding fixed-effect interactions and new random effects for the seven rubric specific models using all the data. We found that there are some differences among the final seven rubric-specific models using all the data (Appendix 4, Part C, p. 6-9).

| | SelMeth | InitEDA | RsrchQ | TxtOrg | CritDes | InterpRes | VisOrg |
|---|---|---|---|---|---|---|---|
| $\hat{\beta}_0$: (Intercept) | - | 2.442 | 2.352 | 2.587 | - | - | - |
| $\hat{\beta}_1$: SemesterS19 | -0.359 | - | - | - | - | - | - |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\hat{\beta}_2$: Rater1 | 2.250 | - | - | - | 1.686 | 2.704 | 2.378 |
| $\hat{\beta}_3$: Rater2 | 2.227 | - | - | - | 2.113 | 2.586 | 2.649 |
| $\hat{\beta}_4$: Rater3 | 2.033 | - | - | - | 1.891 | 2.139 | 2.284 |
| $\hat{\sigma}^2$ | 0.108 | 0.166 | 0.278 | 0.396 | 0.247 | 0.253 | 0.147 |
| $\hat{\tau}^2$ | 0.090 | 0.365 | 0.073 | 0.094 | 0.435 | 0.062 | 0.291 |

Table 6: Estimated coefficients for final seven rubric specific models using all the data.

### Select Method(s)

The final Rubric specific model for rubric Select Method(s), shown below as:

$$\text{Rating} = \beta_1 \times \text{SemesterS19} + \beta_2 \times \text{Rater1} + \beta_3 \times \text{Rater2} + \beta_4 \times \text{Rater3} + (1 \mid \text{Artifact}) - 1 + \varepsilon$$

Table 6 gives the estimated coefficients for the final Rubric specific model for rubric SelMeth and each coefficient is significantly different from zero. We can interpret this model as followings:

- Considering the same rater in the same semester (Semester Spring 19 or Semester Fall 19), different artifacts of the total 91 artifacts tend to get different ratings on rubric Select Method(s). The mean ratings of the total 91 artifacts on rubric Select Method(s) equals to the overall mean ratings on rubric Select Method(s) which is 0 plus the random effect deviations from the overall mean ratings on rubric Select Method(s).
- Considering the same artifact in the same semester, Rater 3 tends to give the lowest ratings on the rubric Select Method(s), followed by Rater 2 tends to give 0.194 higher ratings than Rater 3 on the rubric Select Method(s) and rater1 tends to give 0.023 higher ratings than Rater 2 on the rubric Select Method(s) for the total of 91 artifacts.
- Considering the same artifact rated by the same rater, ratings on the rubric SelMeth tend to be 0.359 lower on Semester Spring 19 than on Semester Fall 19.

### Initial EDA

The final Rubric specific model for rubric Initial EDA, shown below as:

$$\text{Rating} = \beta_0 + (1 \mid \text{Artifact}) + \varepsilon$$

Table 6 gives the estimated coefficients for the final Rubric specific model for rubric Initial EDA and each coefficient is significantly different from zero. We can interpret this model as followings:

- Different artifacts of the total 91 artifacts tend to get different ratings on rubric Initial EDA. The mean ratings of the total 91 artifacts on rubric Initial EDA equals to the overall mean ratings on rubric Initial EDA which is 2.442 plus the random effect deviations from the overall mean ratings on rubric Initial EDA.

### Research Question

The final Rubric specific model for rubric Research Question, shown below as:

$$\text{Rating} = \beta_0 + (1 \mid \text{Artifact}) + \varepsilon$$

Table 6 gives the estimated coefficients for the final Rubric specific model for rubric Research Question and each coefficient is significantly different from zero. We can interpret this model as followings:

- Different artifacts of the total 91 artifacts tend to get different ratings on rubric Research Question. The mean ratings of the total 91 artifacts on rubric Research Question equals to the overall mean ratings on rubric Research Question which is 2.352 plus the random effect deviations from the overall mean ratings on rubric Research Question.

### Text Organization

The final Rubric specific model for rubric Text Organization, shown below as:

$$\text{Rating} = \beta_0 + \ (1 \mid \text{Artifact}) \ + \ \varepsilon$$

Table 6 gives the estimated coefficients for the final Rubric specific model for rubric Text Organization and each coefficient is significantly different from zero. We can interpret this model as followings:

- Different artifacts of the total 91 artifacts tend to get different ratings on rubric Text Organization. The mean ratings of the total 91 artifacts on rubric Text Organization equals to the overall mean ratings on rubric Text Organization which is 2.587 plus the random effect deviations from the overall mean ratings on rubric Text Organization.

### Critique Design

The final Rubric specific model for rubric Critique Design, shown below as:

$$\text{Rating} = \beta_2 \times \text{Rater1} + \beta_3 \times \text{Rater2} + \beta_4 \times \text{Rater3} + (1 \mid \text{Artifact}) - 1 \ + \ \varepsilon$$

Table 6 gives the estimated coefficients for the final Rubric specific model for rubric Critique Design and each coefficient is significantly different from zero. We can interpret this model as followings:

- Considering the same rater, different artifacts of the total 91 artifacts tend to get different ratings on rubric Critique Design. The mean ratings of the total 91 artifacts on rubric Critique Design equals to the overall mean ratings on rubric Critique Design which is 0 plus the random effect deviations from the overall mean ratings on rubric Critique Design.
- Considering the same artifact in the same semester, Rater 1 tends to give the lowest ratings on the rubric Critique Design, followed by Rater 3 tends to give 0.205 higher ratings than Rater 1 on the rubric Critique Design and Rater 2 tends to give 0.222 higher ratings than Rater 3 on the rubric Critique Design for the total of 91 artifacts.

### Interpret Results

The final Rubric specific model for rubric Interpret Results, shown below as:

$$\text{Rating} = \beta_2 \times \text{Rater1} + \beta_3 \times \text{Rater2} + \beta_4 \times \text{Rater3} + (1 \mid \text{Artifact}) - 1 \ + \ \varepsilon$$

Table 6 gives the estimated coefficients for the final Rubric specific model for rubric Interpret Results and each coefficient is significantly different from zero. We can interpret this model as followings:

- Considering the same rater, different artifacts of the total 91 artifacts tend to get different ratings on rubric Interpret Results. The mean ratings of the total 91 artifacts on rubric Interpret Results equals to the overall mean ratings on rubric Interpret Results which is 0 plus the random effect deviations from the overall mean ratings on rubric Interpret Results.
- Considering the same artifact in the same semester, Rater 3 tends to give the lowest ratings on the rubric Interpret Results, followed by Rater 2 tends to give 0.447 higher ratings than Rater 3 on the rubric Interpret Results and Rater 1 tends to give 0.118 higher ratings than Rater 2 on the rubric Interpret Results for the total of 91 artifacts.

### Visual Organization

The final Rubric specific model for rubric Visual Organization, shown below as:

$$\text{Rating} = \beta_2 \times \text{Rater1} + \beta_3 \times \text{Rater2} + \beta_4 \times \text{Rater3} + (1 \mid \text{Artifact}) - 1 \ + \ \varepsilon$$

Table 6 gives the estimated coefficients for the final Rubric specific model for rubric Visual Organization and each coefficient is significantly different from zero. We can interpret this model as followings:

- Considering the same rater, different artifacts of the total 91 artifacts tend to get different ratings on rubric Visual Organization. The mean ratings of the total 91 artifacts on rubric Visual Organization equals to the overall mean ratings on rubric Visual Organization which is 0 plus the random effect deviations from the overall mean ratings on rubric Visual Organization.
- Considering the same artifact in the same semester, Rater 3 tends to give the lowest ratings on the rubric Visual Organization, followed by rater1 tends to give 0.094 higher ratings than Rater 3 on the rubric Visual Organization and Rater 2 tends to give 0.271 higher ratings than Rater 1 on the rubric Visual Organization for the total of 91 artifacts.

### Tried to add fixed effects, interactions, and new random effects to the "combined" model Rating ~ 1 + (0 + Rubric | Artifact), using all the data

These approaches mentioned in Section 4.3, Part 2 and Part 3, don't let us directly examine interactions with Rubric, since each model considers only one Rubric at a time (though we may find differences between the models, or in variable selection, that do suggest interactions with Rubric).

One way to explore interactions with Rubric directly would be to switch to tall.csv. We started with the "combined" intercept-only model by adding fixed effects, fixed-effect interactions, and new random effects, random-effect interactions for all of the variables Rater, Semester, Sex, Repeated and/or Rubric to get the final "combined" model using all the data. Details of these analyses in R can be found in Appendix 4, Part D, p. 6-9. The final "combined" model we got, as shown in Table 6 with estimated regression coefficients:

$$\begin{aligned}\text{Rating} = \ & \beta_0 + \beta_1 \times \text{SemesterS19} + \beta_2 \times \text{Rater1} + \beta_3 \times \text{Rater2} + \beta_4 \times \text{Rater3} \\ & + \beta_5 \times \text{Rater: Rubric} + (0\ +\ \text{Rubric} \mid \text{Artifact})\ +\ (0\ +\ \text{Rater} \mid \text{Artifact}) +\ \varepsilon\end{aligned}$$

| | Estimate | Std. Error | t value |
|---|---|---|---|
| $\hat{\beta}_0$: (Intercept) | 1.758 | 0.114 | 15.413 |
| $\hat{\beta}_1$: SemesterS19 | -0.159 | 0.076 | -2.081 |
| $\hat{\beta}_2$: Rater2 | 0.366 | 0.139 | 2.630 |
| $\hat{\beta}_3$: Rater3 | 0.196 | 0.130 | 1.511 |
| $\hat{\beta}_4$: RubricInitEDA | 0.740 | 0.130 | 5.690 |
| $\hat{\beta}_5$: RubricInterpRes | 0.992 | 0.128 | 7.764 |
| $\hat{\beta}_6$: RubricRsrchQ | 0.726 | 0.118 | 6.158 |
| $\hat{\beta}_7$: RubricSelMeth | 0.411 | 0.125 | 3.293 |
| $\hat{\beta}_8$: RubricTxtOrg | 1.015 | 0.130 | 7.814 |
| $\hat{\beta}_9$: RubricVisOrg | 0.654 | 0.134 | 4.900 |
| $\hat{\beta}_{10}$: Rater2:RubricInitEDA | -0.300 | 0.156 | -1.921 |
| $\hat{\beta}_{11}$: Rater2:RubricInterpRes | -0.513 | 0.153 | -3.344 |
| $\hat{\beta}_{12}$: Rater2:RubricRsrchQ | -0.487 | 0.147 | -3.311 |
| $\hat{\beta}_{13}$: Rater2:RubricSelMeth | -0.386 | 0.150 | -2.571 |
| $\hat{\beta}_{14}$: Rater2:RubricTxtOrg | -0.551 | 0.156 | -3.522 |
| $\hat{\beta}_{15}$: Rater2:RubricVisOrg | -0.105 | 0.159 | -0.661 |
| $\hat{\beta}_{16}$: Rater3:RubricInitEDA | -0.295 | 0.156 | -1.885 |
| $\hat{\beta}_{17}$: Rater3:RubricInterpRes | -0.715 | 0.154 | -4.653 |
| $\hat{\beta}_{18}$: Rater3:RubricRschQ | -0.322 | 0.147 | -2.189 |
| $\hat{\beta}_{19}$: Rater3:RubricSelMeth | -0.387 | 0.150 | -2.588 |
| $\hat{\beta}_{20}$: Rater3:RubricTxtOrg | -0.445 | 0.157 | -2.834 |
| $\hat{\beta}_{21}$: Rater3:RubricVisOrg | -0.275 | 0.159 | -1.732 |

Table 7: Estimated coefficients and standard errors for final “combined” model.

Table 7 gives the estimated coefficients for the final “combined” model along with standard errors and the usual t-test for testing whether each coefficient is significantly different from zero. We can interpret this model as followings:

From Semester, considering the same artifact rated by the same rater, ratings on the all the seven rubrics for rating Freshman Statistics projects tend to be 0.159 lower on Semester Spring 19 than on Semester Fall 19.

From Rubric x Artifact interaction: there are different average scores on each rubric, but the rubric averages also vary a bit from one Artifact to the next, by a small random effect that depends on Artifact. In all of this, the fact that Rubric scores depend on Artifact (that is, there is a kind of Rubric x Artifact interaction) is what we might expect: the artifacts aren't all of equal quality on each rubric, and so we should expect the average scores on each Rubric to vary from one Artifact to the next.

From 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.

From Rater x Rubric interaction: each Rater uses each Rubric in a way that is not like, or even parallel to, other rater's Rubric usage. We saw that in the facets plot also. Details analyses for facet plots showed different patterns of scoring among the 3 raters can be found in Appendix 4, Part D, p. 6-9.

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.

In addition, we can see that some of the random effects are highly correlated with one another (Appendix 4, Part D, p. 6-9: Analyses of “combined” intercept-only model):

- The random effects for rubrics Interpret Results and Initial EDA are highly correlated.
- The random effects for rubrics Select Method(s) and Critique Design are highly correlated.
- The random effects for rubrics Research question and Interpret Results are highly correlated.
- The random effects for rubrics Visual Organization and Text Organization seem highly correlated with each other and with everything except for the random effects for Critique Design and SelMeth, etc.

In some ways we should not be surprised: these rubrics all represent features of a good research report, and we would expect that if someone is good at one or two of these features, they are probably good at the others.

### Conclusion

In conclusion, after doing variable selection for all the fixed effects, random effects, fixed-effect interactions and random-effect interactions based on all of the variables Rater, Semester, Sex, Repeated and/or Rubric, we got the final “combined” model as interpreted from Section 4.3, Part 4. Hence, we concluded that there are three fixed effects have a significant effect in predicting rating, they are Rater, Semester and Rubric; there are two random effects that we can justify adding to the model, they are Rubric and Rater. However, there is a fixed-effect interaction has a significant effect in predicting ratings, which is the fixed interaction between Rater and Rubric; and there is no random interaction has a significant effect in predicting ratings.

## 4.4 Is there anything else interesting to say about this data?

Lastly, we considered two parts about other things we could say about our analysis, that will be of interest to the associate dean and something that is interesting and useful to the college. Based on the analyses from Section 4.3, Part 4, we got the final "combined" model. In this Section, we focused on residual diagnostic plots for the final "combined" model. We did model diagnostics by plotting four types of residuals for the final "combined" model to find something interesting. Specifically, they are marginal residuals, conditional residuals, random effect residuals and cholesky residuals.

### Marginal Residuals

The marginal residual for an observation equals the observed value of the response minus the marginal fitted value of the response. From the residual diagnostic plots of marginal residuals in Appendix 5, p. 6-9, we found that the marginal residuals for the final "combined" model are mean zero. Although, there are some plots (Appendix 5, p. 6-9) may show grouping structures, which indicates there are some correlations in the marginal residuals, but they may not be homoskedastic. Overall nice set of the marginal residuals. Fixed effects in the final "combined" model are good for predicting ratings. Details of marginal residuals analyses in R can be found in Appendix 4, Part D, p. 6-9.

### Conditional Residuals

The conditional residual for an observation equals the observed value of the response minus the conditional fitted value of the response. As shown from the residual diagnostic plots of conditional residuals in Appendix 5, p. 6-9, we can see that the conditional residuals for the final "combined" model are also mean zero but without grouping structures. Conditional residuals for the final "combined" model are homoskedastic. Besides, conditional residuals are not much spread like marginal residuals. Also, conditional residuals for the final "combined" model are shown as normally distributed. We can conclude nice set of the conditional residuals. Details of conditional residuals analyses in R can be found in Appendix 4, Part D, p. 6-9.

### Random Effect Residuals

From the residual diagnostic plots of random effect residuals in Appendix 5, p. 6-9, we can see that the random effect residuals for the final "combined" model are generally not be mean-zero and they may not be homoskedastic. Besides, the random effect residuals for the final "combined" model are also normally distributed, which is the same as the conditional residuals. We can conclude nice set of the random effect residuals. Details of random effect residuals analyses in R can be found in Appendix 4, Part D, p. 6-9.

### Cholesky Residuals

Cholesky residuals are marginal residuals, transformed to remove the correlation. As shown from the residual diagnostic plots of cholesky residuals in Appendix 5, p. 6-9, we found that the distributions of residuals in all artifacts are little more random compared with the marginal residuals. However, there is very little difference between the residual plots of these two residuals, which indicates that the correlations in the marginal residuals are very small for the final "combined" model.  Details of cholesky residuals analyses in R can be found in Appendix 4, Part D, p. 6-9.

In conclusion, our analyses on well-behaved residual diagnostic plots for the final "combined" model indicates that the final "combined" model we got is a good model for predicting ratings.

# 5. Discussion

## 5.1 Recap Findings

In this report, we investigate the relationship between various factors (Rater, Semester, Sex, Repeated, Rubric) and ratings of Freshman Statistics projects in new "General Education" program. Based on our exploratory analyses of ratings (Appendix 2, Part A&B, p. 6-9), we found that rubric Text Organization tends to get especially high ratings, and rubric Select Method(s) tends to get especially low ratings; in addition, Rater 3 tends to give especially low ratings, and Rater 2 tends to give higher ratings, but there is no rater tends to give especially high ratings. Raters generally are consistent with one another in how they rate on rubrics CritDes, SelMeth and VisOrg. However, raters generally are not consistent with one another in how they rate on rubrics RsrchQ, InitEDA, InterpRes and TxtOrg. We got the final "combined" model as interpreted from Section 4.3, Part 4, we concluded that there are three fixed effects Rater, Semester and Rubric; two random effects Rubric and Rater; and one fixed-effect interaction Rubric and Rater have a significant effect in predicting ratings. However, there is no random interaction has a significant effect in predicting ratings. We did model diagnostics by plotting four types of residuals for the final "combined" model to find something interesting.

### 5.2 Compare the seven ICC's for the full data set with the subset corresponding to the 13 artifacts that all three raters saw

One measure of agreement among the raters is the intraclass correlation (ICC); it is the common correlation among the raters' ratings for each artifact. The ICC's can help us determine whether the raters are generally in agreement. As what we have analyzed from Section 4.2, we re-did the ICC calculations on the full data set, but not the percent exact agreement calculations (Appendix 3, p. 6-9). Based on the comparison between the seven ICC's for the full data set (Appendix 3, p. 6-9) with the subset corresponding to the 13 artifacts that all three raters saw (Appendix 3, p. 6-9), we concluded that the seven ICC's for the full data set agree with the seven ICC's for the subset corresponding to the 13 artifacts that all three raters saw. Details of these analyses in R can be found in Appendix 3, p. 6-9.

## References

Kutner, M.H., Nachsheim, C.J., Neter, J. & Li, W. (2005) *Applied Linear Statistical Models*, Fifth Edition. NY: McGraw-Hill/Irwin

Nelson, G. D., & Rae, A. (2016). *An economic geography of the United States: From commutes to megaregions. PLoS ONE*, *11*(11), 1–23. https://doi.org/10.1371/journal.pone.0166083

# Technical Appendix

Olivia Wang

11/13/2021

## Appendix 0. Data Background

1) Dietrich College at Carnegie Mellon University is in the process of implementing a new "General Education" program for undergraduates. This program specifies a set of courses and experiences that all undergraduates must take, and in order to determine whether the new 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. In a recent experiment, 91 project papers referred to as "artifacts" which were randomly sampled from a Fall and Spring section of Freshman Statistics. Three raters from three different departments were asked to rate these artifacts on seven rubrics. The raters did not know which class or which students produced the artifacts that they rated. 13 of the 91 artifacts were rated by all three raters; each of the remaining 78 artifacts were rated by only rater.

2) Rubrics for rating Freshman Statistics projects. NOTE: These are not the rubrics used by instructors or TA's in Freshman Statistics. They are only approved to be used in this experiment.

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

3) Rating scale used for all rubrics. NOTE: This is not the rating scale used by instructors or TA's in Freshman Statistics. It is only approved to be used in this experiment.

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

# Appendix 1. Initial Data Import & Exploration

## Part A

### Initial Look at the Data

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

```
##        X           Rater       Sample          Overlap     Semester
##  Min.   :  1   Min.   :1   Min.   :  1.00   Min.   : 1   Length:117
##  1st Qu.: 30   1st Qu.:1   1st Qu.: 31.00   1st Qu.: 4   Class :character
##  Median : 59   Median :2   Median : 60.00   Median : 7   Mode  :character
##  Mean   : 59   Mean   :2   Mean   : 59.89   Mean   : 7
##  3rd Qu.: 88   3rd Qu.:3   3rd Qu.: 89.00   3rd Qu.:10
##  Max.   :117   Max.   :3   Max.   :118.00   Max.   :13
##                                             NA's   :78
##      Sex                RsrchQ        CritDes          InitEDA
##  Length:117         Min.   :1.00   Min.   :1.000   Min.   :1.000
##  Class :character   1st Qu.:2.00   1st Qu.:1.000   1st Qu.:2.000
##  Mode  :character   Median :2.00   Median :2.000   Median :2.000
##                     Mean   :2.35   Mean   :1.871   Mean   :2.436
##                     3rd Qu.:3.00   3rd Qu.:3.000   3rd Qu.:3.000
##                     Max.   :4.00   Max.   :4.000   Max.   :4.000
##                                    NA's   :1
##     SelMeth        InterpRes        VisOrg          TxtOrg
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :2.000   Median :3.000   Median :2.000   Median :3.000
##  Mean   :2.068   Mean   :2.487   Mean   :2.414   Mean   :2.598
##  3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:3.000
##  Max.   :3.000   Max.   :4.000   Max.   :4.000   Max.   :4.000
##                                  NA's   :1
##    Artifact            Repeated
##  Length:117         Min.   :0.0000
##  Class :character   1st Qu.:0.0000
##  Mode  :character   Median :0.0000
##                     Mean   :0.3333
##                     3rd Qu.:1.0000
##                     Max.   :1.0000
##
```

```r
summary(tall)
```

```
##        X             Rater     Artifact            Repeated
##  Min.   :  1.0   Min.   :1   Length:819         Min.   :0.0000
##  1st Qu.:205.5   1st Qu.:1   Class :character   1st Qu.:0.0000
##  Median :410.0   Median :2   Mode  :character   Median :0.0000
##  Mean   :410.0   Mean   :2                      Mean   :0.3333
```

Table 1:

| X | Rater | Sample | Overlap | Semester | Sex | RsrchQ | CritDes |
|---|---|---|---|---|---|---|---|
| 1 | 3 | 1 | 5 | Fall | M | 3 | 3 |
| 2 | 3 | 2 | 7 | Fall | F | 3 | 3 |
| 3 | 3 | 3 | 9 | Spring | F | 2 | 1 |
| 4 | 3 | 4 | 8 | Spring | M | 2 | 2 |
| 5 | 3 | 5 | NA | Fall | – | 3 | 3 |
| 6 | 3 | 6 | NA | Fall | M | 2 | 1 |

Table 2:

| X | InitEDA | SelMeth | InterpRes | VisOrg | TxtOrg | Artifact | Repeated |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 2 | 3 | O5 | 1 |
| 2 | 3 | 3 | 3 | 3 | 3 | O7 | 1 |
| 3 | 3 | 2 | 3 | 3 | 3 | O9 | 1 |
| 4 | 2 | 1 | 1 | 1 | 1 | O8 | 1 |
| 5 | 3 | 3 | 3 | 3 | 3 | 5 | 0 |
| 6 | 2 | 2 | 2 | 2 | 2 | 6 | 0 |

```
##  3rd Qu.:614.5   3rd Qu.:3                      3rd Qu.:1.0000
##  Max.   :819.0   Max.   :3                      Max.   :1.0000
##
##    Semester             Sex               Rubric              Rating
##  Length:819         Length:819         Length:819         Min.   :1.000
##  Class  :character  Class  :character  Class  :character  1st Qu.:2.000
##  Mode   :character  Mode   :character  Mode   :character  Median :2.000
##                                                           Mean   :2.318
##                                                           3rd Qu.:3.000
##                                                           Max.   :4.000
##                                                           NA's   :2
```

## Part B

### ratings.csv

First I'll just look at the "head" of all the variables; this is a bit like looking at `str()`, which is also worth looking at (I split the "head" into two parts, because there are too many variables to fit horizontally on the page).

```
head(ratings[,1:8]) %>% kbl(booktabs=T,caption=" ") %>% kable_classic()
```

```
head(ratings[,c(1,9:15)]) %>% kbl(booktabs=T,caption=" ") %>% kable_classic()
```

We can also check to see how many unique values each variable has (this is especially relevant for the categorical variables). From the table below, we found there is a strange thing that Sex variable (Sex or gender of student who created the artifact) has three unique values. But at this part, we haven't decide to consider Sex in our analysis yet, however we will discuss it more in Appendix 3, Part A.

Table 3:

| | unique values |
|---|---|
| X | 117 |
| Rater | 3 |
| Sample | 117 |
| Overlap | 14 |
| Semester | 2 |
| Sex | 3 |
| RsrchQ | 4 |
| CritDes | 5 |
| InitEDA | 4 |
| SelMeth | 3 |
| InterpRes | 4 |
| VisOrg | 5 |
| TxtOrg | 4 |
| Artifact | 91 |
| Repeated | 2 |

```
apply(ratings,2,function(x) {length(unique(x))}) %>%
  kbl(booktabs=T,col.names="unique values",caption=" ") %>%
  kable_classic(full_width=F)
```

We can check for NA's directly:

```
apply(ratings,2,function(x) any(is.na(x)))
```

```
##         X     Rater    Sample   Overlap  Semester       Sex    RsrchQ   CritDes
##     FALSE     FALSE     FALSE      TRUE     FALSE     FALSE     FALSE      TRUE
##   InitEDA   SelMeth InterpRes    VisOrg    TxtOrg  Artifact  Repeated
##     FALSE     FALSE     FALSE      TRUE     FALSE     FALSE     FALSE
```

```
## find the row number (X) of each missing value of three variables (CritDes, VisOrg and Overlap)
which(is.na(ratings$CritDes))
```

```
## [1] 44
```

```
which(is.na(ratings$VisOrg))
```

```
## [1] 99
```

```
which(is.na(ratings$Overlap))
```

```
##  [1]   5   6   7   8   9  13  14  15  16  20  21  22  23  24  25  26  27  31  32
## [20]  33  34  35  36  37  38  39  44  45  46  47  48  52  53  54  55  56  60  61
## [39]  62  63  64  65  66  67  71  72  73  74  75  76  77  78  83  84  85  86  87
## [58]  91  92  93  94  95  99 100 101 102 103 104 105 106 110 111 112 113 114 115
## [77] 116 117
```

```
length(which(is.na(ratings$Overlap)))
```

```
## [1] 78
```

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

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

- There do appear to be any missing values in the data! As we can see there are three variables (Overlap, CritDes, and VisOrg) have NA's (In general we might also check for old-fashioned missing value codes like "9", "99", "98", etc., but there's no evidence of that in Table 5 (look at the Min and Max values - no "9's", "99's", etc.))
- Specifically the row number (X) of one missing value for CritDes is 44, the row number (X) of one missing value for VisOrg is 99, there are total of 78 missing values for Overlap.
- 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, I'm just going to leave it as a third "Sex" category for now…
- Considering the Research Question #1 is only associated with the 7 rubrics for rating Freshman Statistics projects. So we decided to remove rows of missing value for CritDes (44) and VisOrg (99) of ratings.csv to get the ratings_rubrics.
- Note that none of the missing values occur in the smaller 13-rubric data set (how can we tell this from the output above?). So we don't have to worry about missing data at all in analyses that just involve this smaller data set.

## Part C

### tall.csv

We can check for NA's directly:

```
apply(tall,2,function(x) any(is.na(x)))
```

```
##         X     Rater Artifact Repeated Semester       Sex    Rubric    Rating
##     FALSE     FALSE    FALSE    FALSE    FALSE     FALSE     FALSE      TRUE
```

```
## find the row number (X) of each missing value of variables (CritDes, VisOrg and Overlap)
which(is.na(tall$Sex))
```

```
## integer(0)
```

```
which(is.na(tall$Rating))
```

```
## [1] 161 684
```

```
length(which(is.na(tall$Sex)))
```

```
## [1] 0
```

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

- There do appear to be any missing values in the data! As we can see there are two variables (Sex and Rating) have NA’s. (In general we might also check for old-fashioned missing value codes like “9”, “99”, “98”, etc., but there’s no evidence of that in Table 5 (look at the Min and Max values - no “9’s”, “99’s”, etc.))

- 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).

- Specifically the row number (X) of two missing values for Rating are 161 and 684, there are total of 7 missing values for Sex. 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, I’m just going to leave it as a third “Sex” category for now…

- Considering the Research Question #1 is only associated with the 7 rubrics for rating Freshman Statistics projects. So we decided to remove rows of missing values for Rating (161, 684) of tall.csv to get the tall_rubrics.

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

Table 4:

| | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. | SD |
|---|---|---|---|---|---|---|---|
| RsrchQ | 1 | 2 | 2 | 2.36 | 3.0 | 4 | 0.60 |
| CritDes | 1 | 1 | 2 | 1.86 | 2.5 | 4 | 0.84 |
| InitEDA | 1 | 2 | 2 | 2.44 | 3.0 | 4 | 0.70 |
| SelMeth | 1 | 2 | 2 | 2.06 | 2.0 | 3 | 0.48 |
| InterpRes | 1 | 2 | 3 | 2.49 | 3.0 | 4 | 0.61 |
| VisOrg | 1 | 2 | 2 | 2.42 | 3.0 | 4 | 0.68 |
| TxtOrg | 1 | 2 | 3 | 2.60 | 3.0 | 4 | 0.70 |

# Appendix 2. Research Question #1

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

### Part A

#### Is the distribution of ratings for each rubrics pretty much indistinguishable from the other rubrics, or are there rubrics that tend to get especially high or low ratings?

First of all, let us use ratings.csv to do the analysis. So let's make a table with the usual one-dimensional summary statistics based on the subset of the ratings.csv data set with only seven rubrics of all artifacts named as ratings_rubrics.

```
ratings_rubrics <- ratings[-c(44,99),c(7:13)]
apply(ratings_rubrics,2,function(x) c(summary(x),SD=sd(x))) %>% as.data.frame %>% t() %>%
  round(digits=2) %>% kbl(booktabs=T,caption=" ") %>% kable_classic()
```

Secondly, let's consider tall.csv So let's make a table based on the subset of the tall.csv data set with only Rater and Rating of all artifacts named as tall_rubrics.

```
tall_rubrics <- tall[-c(161,684),c(2,8)]
```

Then let's make 3 subsets of ratings given by three raters named as rater1, rater2 and rater3.

```
rater1 = tall_rubrics[which(tall_rubrics$Rater==1),]
rater2 = tall_rubrics[which(tall_rubrics$Rater==2),]
rater3 = tall_rubrics[which(tall_rubrics$Rater==3),]
```

Next, let's make a table with the usual one-dimensional summary statistics (Table 5) based on the subset of ratings given by three raters named as rater1, rater2 and rater3.

```
rater1$Rating <- as.numeric(rater1$Rating)
rater2$Rating <- as.numeric(rater2$Rating)
rater3$Rating <- as.numeric(rater3$Rating)
```

Table 5:

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

```
rater_rating <- cbind(c(summary(rater1$Rating),SD=sd(rater1$Rating)),
                      c(summary(rater2$Rating),SD=sd(rater2$Rating)),
                      c(summary(rater3$Rating),SD=sd(rater3$Rating))) %>%
                 as.data.frame
colnames(rater_rating) = c('Rater1', 'Rater2', 'Rater3')

rater_rating %>% t() %>%
  round(digits=2) %>% kbl(booktabs=T,caption=" ") %>% kable_classic()
```

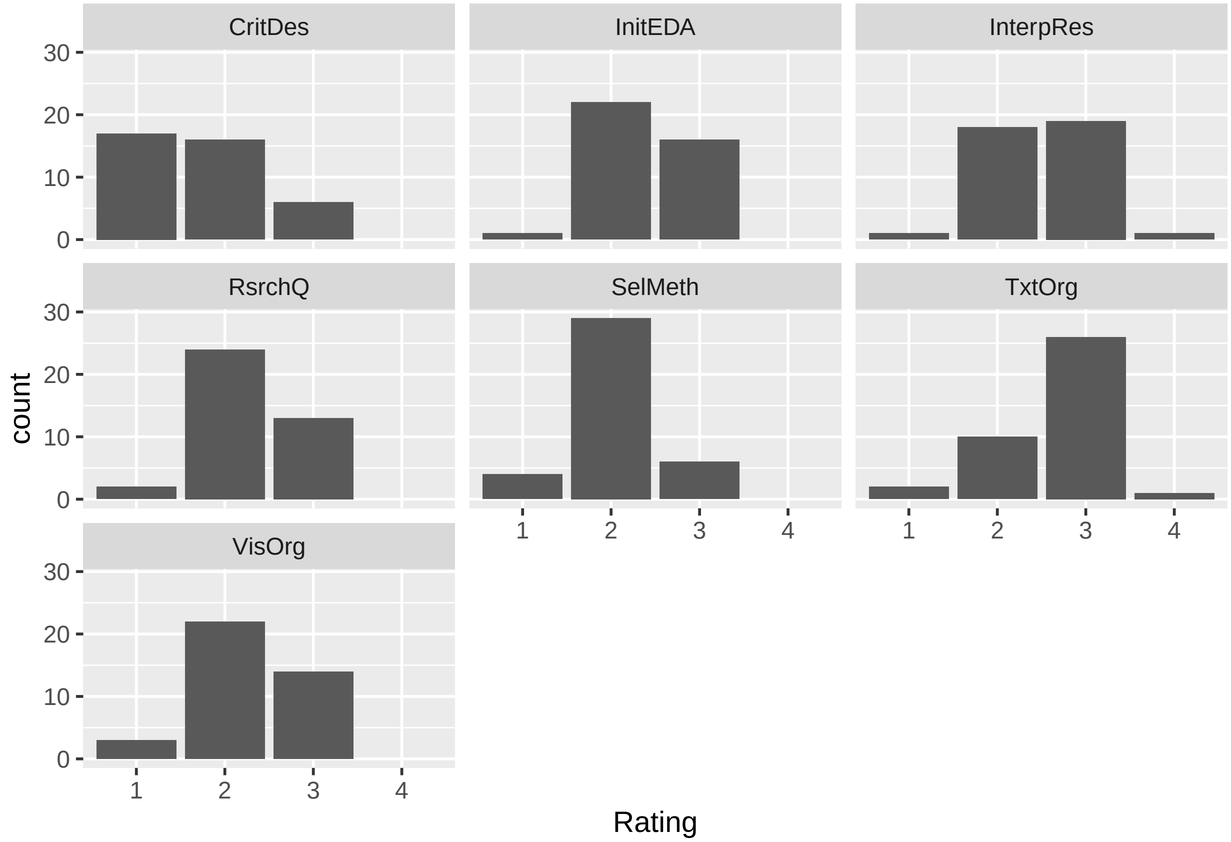
```
## take care that all ratings run from 1 to 4,
## whether or not rater used all categories...
tall$Rating <- factor(tall$Rating,levels=1:4)
for (i in unique(tall$Rubric)) {
  ratings[,i] <- factor(ratings[,i],levels=1:4)
}

## extract the reduced data set with the 13 artifacts that all 3 raters saw...
ratings.13 <- ratings[grep("0",ratings$Artifact),]
tall.13 <-  tall[grep("0",tall$Artifact),]
```

Here are some ideas to compare distributions across Rubrics.

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

g
```



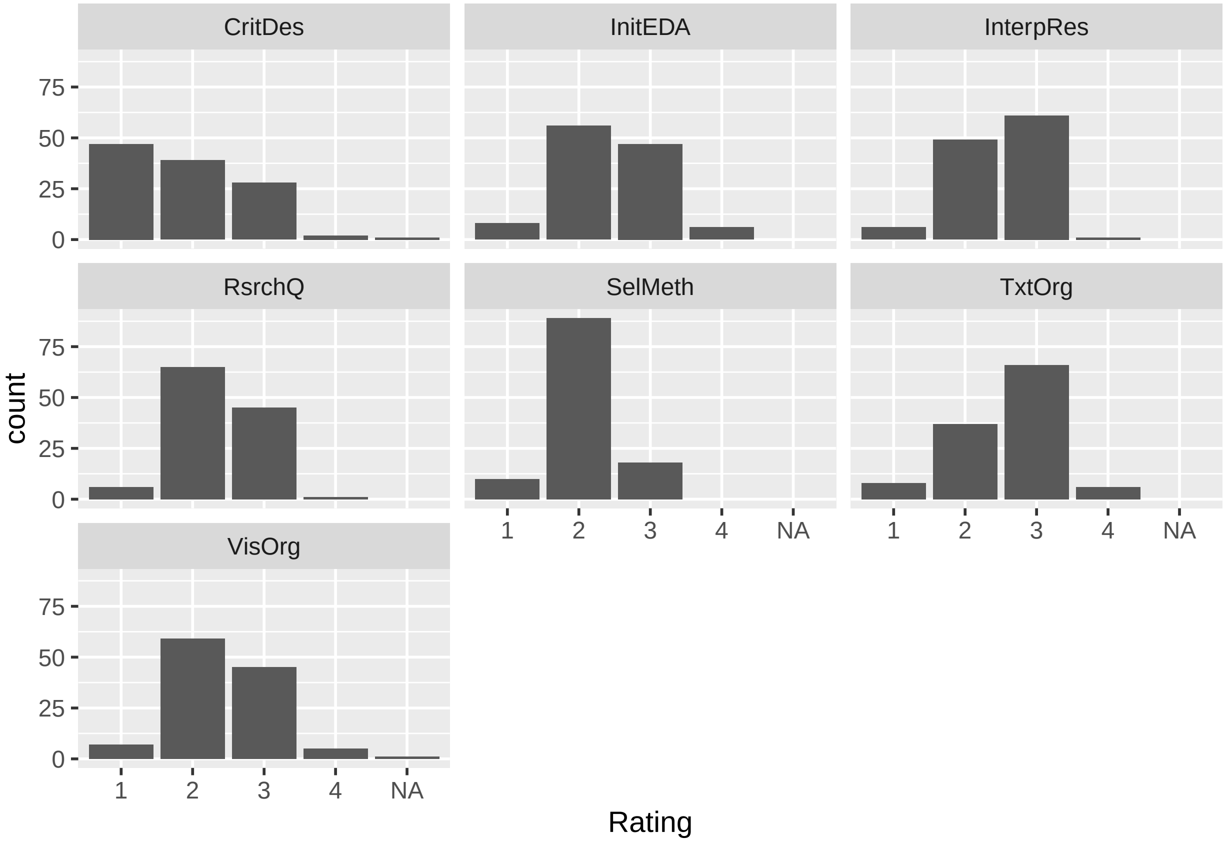
```
##
## Table of counts might make a nice supplement
tmp <- data.frame(lapply(split(tall.13$Rating,tall.13$Rubric),summary))
row.names(tmp) <- paste("Rating",1:4)

tmp
```

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

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

g
```



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

tmp
```

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

**Is the distribution of ratings for each rubrics pretty much indistinguishable from the other rubrics, or are there rubrics that tend to get especially high or low ratings?**

- In order to answer this question, we made an assumption first, among all rubrics, we defined low rating as artifact was rated less and equal to 2; high rating as artifact was rated above 2.

- Based on the table of counts, numerical summaries table and the barplots of ratings for seven rubrics for full data set, we can get the followings:

1) Rubric SelMeth (Rating on Select Method(s)) tends to get especially low ratings.

1.1) From the table of counts, the total number of low ratings for rubric SelMeth is 99 which is the highest one among all the rubrics for rating Freshman Statistics projects.

1.2) If we take a look at the distribution of ratings on rubric SelMeth, there are nearly 100 artifacts were rated as low ratings which is the highest number of artifacts were rated as low ratings among all the rubrics.

1.3) The Max Value of ratings on rubric SelMeth is 3 which is the lowest Max Value of all rubrics; besides, both of the Median Value and 3rd Quartile Value of ratings on rubric SelMeth are 2 which means about 75% of the artifacts were rated under than 2. Considering that the Mean Value of ratings on rubric SelMeth is 2.06 which is close to the Median Value and 3rd Quartile Value of ratings on rubric SelMeth, and the Standard Deviation of ratings on rubric SelMeth is 0.48 which is the lowest one among all rubrics, which means the dispersion of the ratings on SelMeth relatives to its mean is low, so overall most of the artifacts were rated between 2 and 3 on rubric SelMeth.

1.4) Although, from the numerical summaries table, the Mean Value of CritDes (Rating on Critique Design) is 1.86 which is the lowest Mean Value among all rubrics, and also from the the distribution of rubric CritDes, the number of the artifacts rated at grade 1 is the highest among all rubrics. But considering we assumed low ratings as artifacts were rated at grade less and equal to 2, the number of artifacts were rated as low ratings of rubric CritDes are nearly 86 which is less than rubric SelMeth, so we still continue with rubric SelMeth tends to get especially low ratings.

2) Rubric TxtOrg (Rating on Text Organization) tends to get especially high ratings.

2.1) From the table of counts, the total number of high ratings for rubric TxtOrg is 72 which is the highest one among all the rubrics for rating Freshman Statistics projects.

2.2) The Max Value of ratings on rubric TxtOrg is 4 which is the highest one of all rubrics; besides, the Mean Value of ratings on rubric TxtOrg is 2.60 which is the highest Mean Value among all the rubrics. Also both of the Median Value and 3rd Quartile Value of ratings on rubric TxtOrg are 3, all of both are also the highest ones among all the rubrics.

2.3) If we take a look at the distributions of ratings on rubric TxtOrg, the number of artifacts rated at grade 1 and 4 are very close. There are over 70 artifacts were rated as high ratings for rubric TxtOrg which is the highest number of artifacts were rated as high ratings among all the rubrics.

## Part B

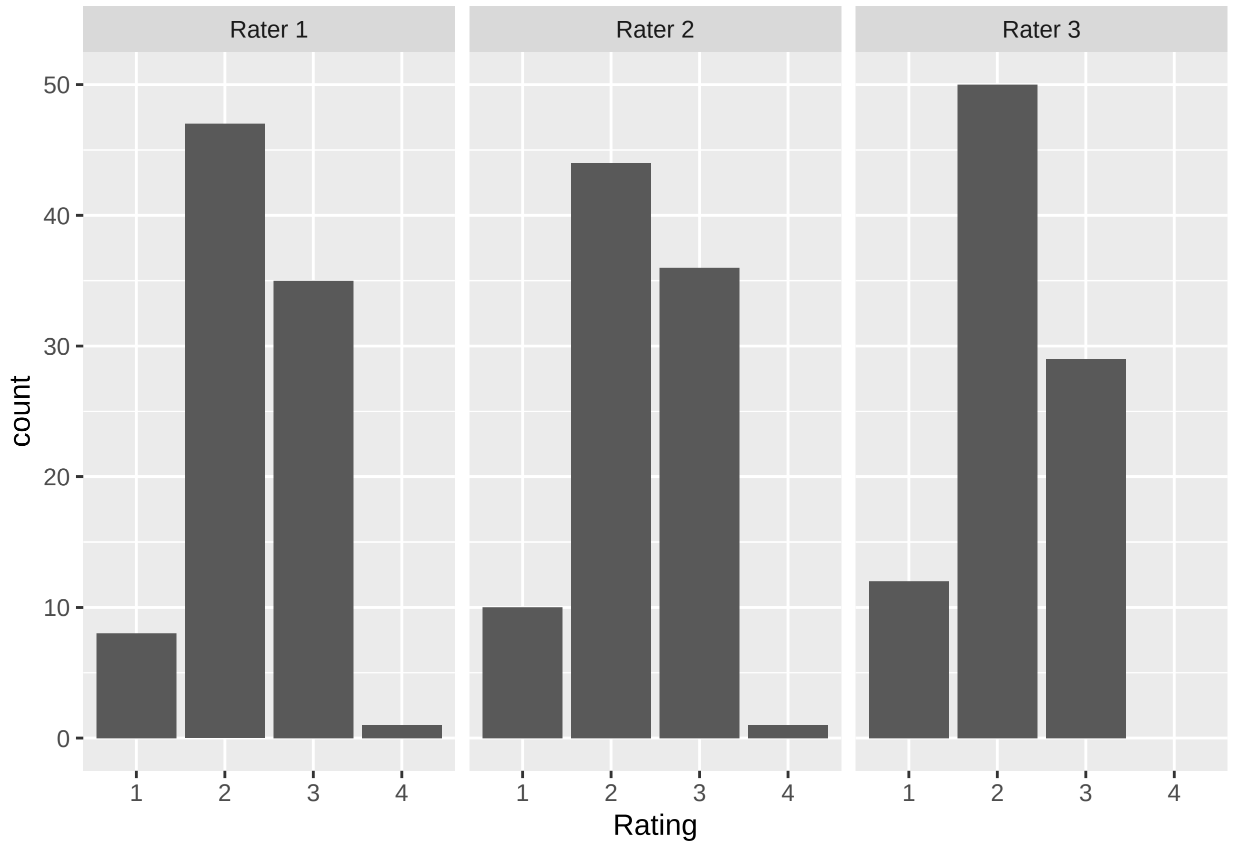
**Is the distribution of ratings given by each rater pretty much indistinguishable from the other raters, or are there raters that tend to give especially high or low ratings?**

And here are some idea to compare distributions across Raters.

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

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

g
```



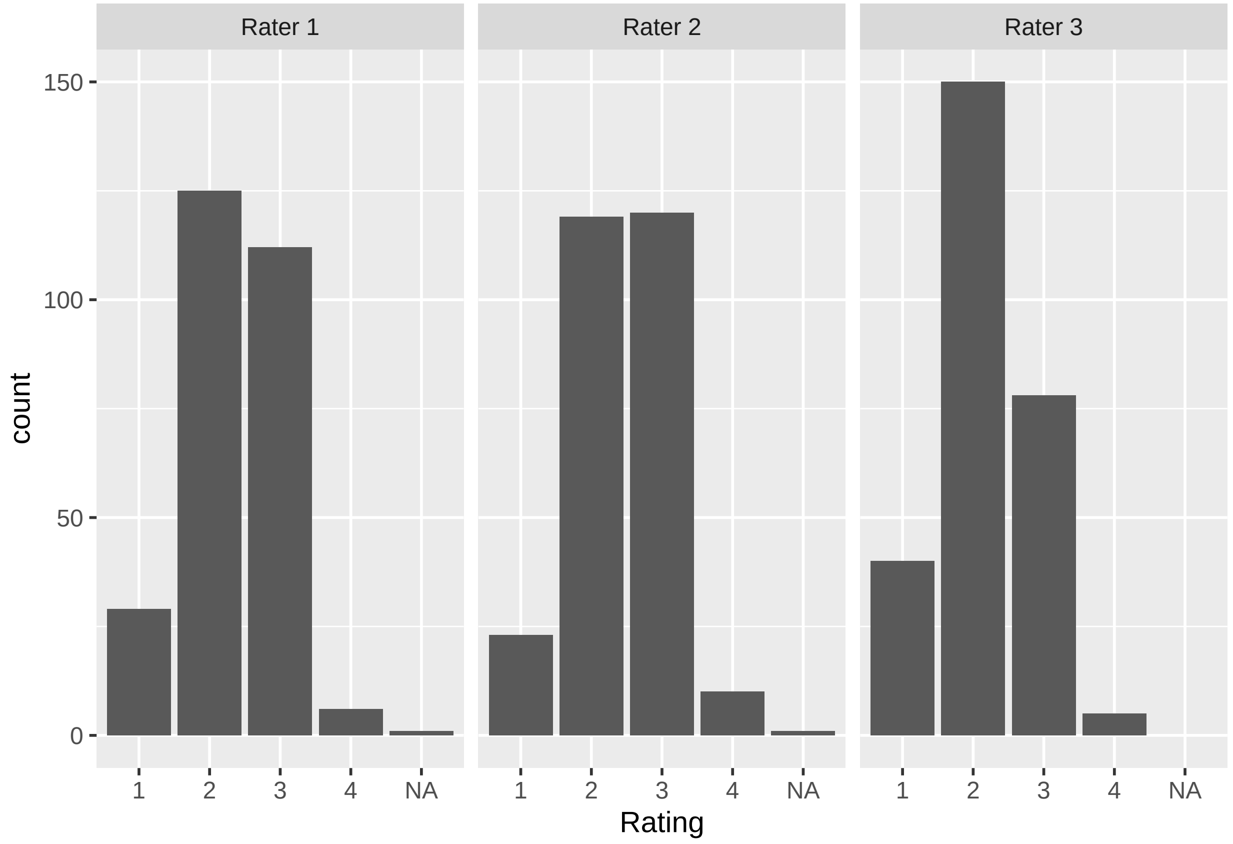
```
##
## Corresponding table of counts...
tmp <- data.frame(lapply(split(tall.13$Rating,tall.13$Rater),summary))
row.names(tmp) <- paste("Rating",1:4)
names(tmp) <- paste("Rater",1:3)

tmp
```

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

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

```
g
```



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

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

**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?**

- In order to answer this question, we made an assumption first, among all rubrics, we defined low rating as artifact was rated less and equal to 2; high rating as artifact was rated above 2.
- Based on the table of counts, numerical summaries table and the barplots of ratings given by each rater for full data set, we can get the followings:

0) From the numerical summaries table, all of the values (Max, Min, Median, Mean, 1st Quartile, 3rd Quartile, Standard Deviation, etc.) of ratings given by three raters are very similar.

1) Rater3 tends to give especially low ratings.

1.1) From the table of counts, the total number of low ratings given by rater3 is 190 which is the highest one among all the ratings given by each rater for rating Freshman Statistics projects.

1.2) If we take a look at the distributions of ratings given by rater3, we can still get rater3 tends to give especially low ratings, because from table rater1 (272 objects), rater2 (272 objects) and rater3 (273 objects), all of the raters have rated similar number of artifacts, however, there are nearly 150 artifacts rated at grade 2 from rater3, this number is significantly higher than number of artifacts rated at grade 2 by other two raters, also the number of artifacts rated at grade 1 by rater3 are also higher than number of artifacts rated at grade 1 by rater1 and rater2.

2) Rater2 tends to give higher ratings.

2.1) From the table of counts, the total number of high ratings given by rater2 is 130 which is the highest one among all the ratings given by each rater for rating Freshman Statistics projects.

2.2) If we take a look at the distributions of ratings given by rater2, we can still get rater2 tends to give higher ratings, because from table rater1 (272 objects), rater2 (272 objects) and rater3 (273 objects), all of the raters have rated similar number of artifacts, also we can see the distribution of ratings given by rater2 and rater1 are very similar, both of them are tend to give high ratings. However, we can clearly see the number of artifacts rated at both grade 3 and 4 by rater2 are both higher than by rater1.

2.3) However, there is no significant phenomenon from distributions of ratings given by rater1 and rater2 that we can conclude which rater tends to give especially high ratings. Because both of distributions of ratings given by rater1 and rater2 are very similar.

## Part C

### Compare and determine whether these thirteen artifacts are representative of the whole set of 91 artifacts?

We want to compare and determine whether these thirteen artifacts are representative of the whole set of 91 artifacts.

```
ratings.13$RsrchQ = as.numeric(ratings.13$RsrchQ)
ratings.13$CritDes = as.numeric(ratings.13$CritDes)
ratings.13$InitEDA = as.numeric(ratings.13$InitEDA)
ratings.13$SelMeth = as.numeric(ratings.13$SelMeth)
ratings.13$InterpRes = as.numeric(ratings.13$InterpRes)
ratings.13$VisOrg = as.numeric(ratings.13$VisOrg)
ratings.13$TxtOrg = as.numeric(ratings.13$TxtOrg)
```

I'll just look at the "head" of all the variables; this is a bit like looking at `str()`, which is also worth looking at (I split the "head" into two parts, because there are too many variables to fit horizontally on the page).

Table 6:

| | X | Rater | Sample | Overlap | Semester | Sex | RsrchQ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 3 | 1 | 5 | Fall | M | 3 |
| 2 | 2 | 3 | 2 | 7 | Fall | F | 3 |
| 3 | 3 | 3 | 3 | 9 | Spring | F | 2 |
| 4 | 4 | 3 | 4 | 8 | Spring | M | 2 |
| 10 | 10 | 3 | 10 | 10 | Fall | F | 2 |
| 11 | 11 | 3 | 11 | 13 | Fall | M | 2 |

Table 7:

| | X | CritDes | InitEDA | SelMeth | InterpRes | VisOrg | TxtOrg | Artifact |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 3 | 2 | 2 | 2 | 2 | 3 | O5 |
| 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | O7 |
| 3 | 3 | 1 | 3 | 2 | 3 | 3 | 3 | O9 |
| 4 | 4 | 2 | 2 | 1 | 1 | 1 | 1 | O8 |
| 10 | 10 | 1 | 2 | 2 | 3 | 2 | 3 | O10 |
| 11 | 11 | 2 | 2 | 2 | 2 | 3 | 3 | O13 |

```
## extract the reduced data set with the 13 artifacts that all 3 raters saw...
ratings.13 <- ratings[grep("O",ratings$Artifact),]
head(ratings.13[,1:7]) %>% kbl(booktabs=T,caption=" ") %>% kable_classic()
```

```
head(ratings.13[,c(1,8:14)]) %>% kbl(booktabs=T,caption=" ") %>% kable_classic()
```

Next let's make a table with the usual one-dimensional summary statistics (Table 5) based on the subset of the ratings.13 data with only seven rubrics of all artifacts named as ratings13_rubrics.

```
ratings13_rubrics <- ratings.13[,c(7:13)]
ratings13_rubrics$RsrchQ = as.numeric(ratings13_rubrics$RsrchQ)
ratings13_rubrics$CritDes = as.numeric(ratings13_rubrics$CritDes)
ratings13_rubrics$InitEDA = as.numeric(ratings13_rubrics$InitEDA)
ratings13_rubrics$SelMeth = as.numeric(ratings13_rubrics$SelMeth)
ratings13_rubrics$InterpRes = as.numeric(ratings13_rubrics$InterpRes)
ratings13_rubrics$VisOrg = as.numeric(ratings13_rubrics$VisOrg)
ratings13_rubrics$TxtOrg = as.numeric(ratings13_rubrics$TxtOrg)

apply(ratings13_rubrics,2,function(x) c(summary(x),SD=sd(x))) %>% as.data.frame %>% t() %>%
  round(digits=2) %>% kbl(booktabs=T,caption=" ") %>% kable_classic()
```

**Compare and determine whether these thirteen artifacts are representative of the whole set of 91 artifacts?** Yes. For the following reasons:

- Based on the table of counts, numerical summaries table and the barplots of ratings from Appendix 2, Part A & B for the subset of the data for just the 13 artifacts seen by all three raters and full data set, we can get the followings:

Table 8:

| | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. | SD |
|---|---|---|---|---|---|---|---|
| RsrchQ | 1 | 2 | 2 | 2.28 | 3 | 3 | 0.56 |
| CritDes | 1 | 1 | 2 | 1.72 | 2 | 3 | 0.72 |
| InitEDA | 1 | 2 | 2 | 2.38 | 3 | 3 | 0.54 |
| SelMeth | 1 | 2 | 2 | 2.05 | 2 | 3 | 0.51 |
| InterpRes | 1 | 2 | 3 | 2.51 | 3 | 4 | 0.60 |
| VisOrg | 1 | 2 | 2 | 2.28 | 3 | 3 | 0.60 |
| TxtOrg | 1 | 2 | 3 | 2.67 | 3 | 4 | 0.62 |

1) Compared one-dimensional summary statistics based on the subset of the data for just the 13 artifacts seen by all three raters with full data set from Appendix 2, Part A & B, we can see the all of Mean Values and Min Values for ratings of all rubrics are the same for both datasets; standard Deviations, 1st Quartiles, 3rd Quartiles for ratings of all rubrics are very similar. Except of most of the Max Values of all ratings is 4 in ratings data, however most of the Max Values of all ratings is 3 in ratings.13 data.

2) Compared univariate distributions of ratings based on the subset of the data for just the 13 artifacts seen by all three raters with full data set from Appendix 2, Part A & B, we can see the univariate distributions of ratings have very similar trends. Except some ratings of have different number of unique values for each of situation mentioned in Part A & B.

3) Compared the table of counts based on the subset of the data for just the 13 artifacts seen by all three raters with full data set from Appendix 2, Part A & B to calculate the total number of ratings based on different situations for rating Freshman Statistics projects mentioned in Part A & B, we will get exact the same conclusions as Part A & B.

4) Hence, we approved that these thirteen artifacts are representative of the whole set of 91 artifacts.

# Appendix 3. Research Question #2

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

For this question, we focus on the subset of the data for just the 13 artifacts seen by all three raters (tall.13).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
##

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

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

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

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

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

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

round(agreement.results,2)
```

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

**For each rubric, do the raters generally agree on their scores?** If we try to fit the model, now there will be 13 groups, one for each artifact, and three y's per group: one for each rater's rating on that artifact. Now the ICC is the correlation between any two rater's ratings on the same artifact. If the raters are consistent with one another in how they rate, we would expect this correlation to be higher. Moreover, this between-raters correlation does tell us something useful about rater agreement: raters agree more when their correlations are higher.

Based on the rules of thumb for interpreting ICC, we would conclude that an ICC of 0.782 indicates that the rubrics can be rated with "good" reliability by different raters.

The output of the values of seven ICC's, we usually define ICC between 0.5 and 0.75 as moderate reliability. In this case, we can conclude that raters generally are consistent with one another in how they rate on rubrics CritDes, SelMeth and VisOrg. However, we usually define ICC below 0.5 as poor reliability. In this case, we can see that raters generally are not consistent with one another in how they rate on rubrics RsrchQ, InitEDA, InterpRes and TxtOrg.

**Is there one rater who disagrees with the others? Or do they all disagree?** In this case, we need to find which raters might be contributing to disagreement. So we start to make a 2-way table of counts for the ratings of each pair of raters, on each rubric (since there are three pairs of raters, each rubric will get three tables).

To compute exact agreement rates between raters (and also to have an idea of how severe disagreements can be) it is useful to create tables of counts cross-classifying the rating that each pair of raters gives. It's easiest to start with the original, wide data, but again we want to subset to (a) just the 13 artifacts seen by everyone, and (b) just the data related to a single rubric at a time.

From all the results of 2-way table of counts for the ratings of each pair of raters, on each rubric (tables of counts cross-classifying the rating that each pair of raters gives):

- RsrchQ (Rating on Research Question): The 2-way table of counts for the ratings of Rater1 vs Rater2 has the accuracy of 0.3846 which is lowest among all the three pairs of raters. Hence rater1 and rater2 might be contributing to disagreement on their scores on rubric RsrchQ.
- InitEDA (Rating on Initial EDA): The 2-way table of counts for the ratings of Rater1 vs Rater3 has the accuracy of 0.5385 which is lowest among all the three pairs of raters. Hence rater1 and rater3 might be contributing to disagreement on their scores on rubric InitEDA.
- InterpRes (Rating on Interpret Results): The 2-way table of counts for the ratings of Rater1 vs Rater3 has the accuracy of 0.5385 which is lowest among all the three pairs of raters. Hence rater1 and rater3 might be contributing to disagreement on their scores on rubric InterpRes.
- TxtOrg (Rating on Text Organization): The 2-way table of counts for the ratings of Rater2 vs Rater3 has the accuracy of 0.5385 which is lowest among all the three pairs of raters. Hence rater2 and rater3 might be contributing to disagreement on their scores on rubric TxtOrg.

**You can re-do the ICC calculations on the full data set (but not the percent exact agreement calculations—why not?). Do the seven ICC's for the full data set agree with the seven ICC's for the subset corresponding to the 13 artifacts that all three raters saw?**

- We do not need to re-do the percent exact agreement calculations because thirteen of the 91 artifacts were rated by all three raters; each of the remaining 78 artifacts were rated by only rater. Whatever we used the full data set or focus on the subset of the data for just the 13 artifacts seen by all three raters. We all only based on the 13 artifacts seen by all three raters to make a 2-way table of counts for the ratings of each pair of raters on each rubric (the percent exact agreement calculations). So for both full data set and the subset of the data for just the 13 artifacts seen by all three raters, the percent exact agreement calculations will be the same.
- The seven ICC's for the full data set agree with the seven ICC's for the subset corresponding to the 13 artifacts that all three raters saw, because all of the seven ICC's for the full data set are close to the seven ICC's for the subset, the maximum difference is around 0.1.

# Appendix 4. Research Question #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?

### Part A

**Adding fixed effects to the seven rubric-specific models using just the data from the 13 common artifacts that all three raters saw.**

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

I will start with the reduced data - tall.13 (so of course I can't check "repeated" on this reduced data—since all the repeated = 1 on this reduced data).

```
library(LMERConvenienceFunctions)
library(LMERConvenienceFunctions, warn.conflicts=F, quietly=T)
library(RLRsim)

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

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

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

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

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

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

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

formula(tmp.back_elim)
```

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

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

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

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

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

```
## Data: tall.13[tall.13$Rubric == "RsrchQ", ]
## Models:
## tmp.int_only: as.numeric(Rating) ~ (1 | Artifact)
## tmp.back_elim: as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) - 1
##               npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## tmp.int_only     3 69.457 74.447 -31.728   63.457
## tmp.back_elim    5 72.018 80.335 -31.009   62.018 1.4391  2      0.487
## Again the models are nested so I really only need to look at the p-value
## from the likelihood ratio chi-squared test.  A little fooling around with
## names(anova(tmp.int_only,tmp.back_elim)), etc., shows that I can get this
## as

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

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

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

## Note: since no main effects were retained, there's really no reason to
## check for interactions.

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


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

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

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

for (i in Rubric.names) {

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

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

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

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

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

}
```

```
## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.2229 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.1826 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.8137 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.6429 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.8294 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.2947 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.7355 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.279 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.9383 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.4287 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## =========================================================
## ===            forwardfitting random effects       ===
## =========================================================
##  ===       random slopes       ===
## =========================================================
## ===            re-backfitting fixed effects        ===
## =========================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## =========================================================
## ===              backfitting fixed effects         ===
## =========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.5358 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.1319 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## =========================================================
## ===            forwardfitting random effects       ===
## =========================================================
##  ===       random slopes       ===
## =========================================================
## ===            re-backfitting fixed effects        ===
## =========================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## =========================================================
## ===              backfitting fixed effects         ===
## =========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.1922 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.1078 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
```

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

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

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

## Part B

**Adding fixed effects to the seven rubric-specific models using all the data.**

Now let's try with the full data…

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

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

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

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

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

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

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

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

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

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

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

for (i in Rubric.names) {

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

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

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

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

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

}
```

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

## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.7154 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.5297 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune

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

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

## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.8802 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.7402 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune

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

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

## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.608 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.5312 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
## refitting model(s) with ML (instead of REML)

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

## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.6166 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.3987 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune

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

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

## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.1935 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune

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

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

## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.5041 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.205 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune

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

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

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

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

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

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

## Part C

**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 (2) if there are any interactions or additional random effects to consider.

```
## 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
```

### SelMeth

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

```
## apparently they do.

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

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

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

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

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

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

anova(tmp,tmp.fixed_interactions)
```

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

```
## Data: tall.nonmissing[tall.nonmissing$Rubric == "SelMeth", ]
## Models:
```

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

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

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

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

## Testing (Semester|Artifact)...

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

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

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

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

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

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

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



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

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

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

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

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

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

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

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


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

summary(tmp)
```

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

```
display(tmp)
```

```
## lmer(formula = as.numeric(Rating) ~ as.factor(Rater) + Semester +
##     (1 | Artifact) - 1, data = tall.nonmissing[tall.nonmissing$Rubric ==
##     "SelMeth", ])
##                    coef.est coef.se
## as.factor(Rater)1  2.25     0.08   
## as.factor(Rater)2  2.23     0.07   
## as.factor(Rater)3  2.03     0.08   
## SemesterS19       -0.36     0.10   
## 
## Error terms:
##  Groups   Name        Std.Dev.
##  Artifact (Intercept) 0.30    
##  Residual             0.33    
## ---
## number of obs: 116, groups: Artifact, 90
## AIC = 155.6, DIC = 116.6
## deviance = 130.1
```

```
formula(tmp)
```

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

The final Rubric specific model for rubric SelMeth: $as.numeric(Rating)\ as.factor(Rater) + Semester + (1|Artifact) - 1$.

- Considering the same rater in the same semester (Semester Spring 19 or Semester Fall 19), different artifacts of the total 91 artifacts tend to get different ratings on rubric SelMeth. The mean ratings of the total 91 artifacts on rubric SelMeth equals to the overall mean ratings on rubric SelMeth which is 0 plus the random effect deviations from the overall mean ratings on rubric SelMeth.
- Considering the same artifact in the same semester, rater3 tends to give the lowest ratings on the rubric SelMeth, followed by rater2 tends to give 0.19337 higher ratings than rater3 on the rubric SelMeth and rater1 tends to give 0.02384 higher ratings than rater2 on the rubric SelMeth for the total of 91 artifacts.
- Considering the same artifact rated by the same rater, ratings on the rubric SelMeth tend to be 0.35860 lower on Semester Spring 19 than on Semester Fall 19.

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

### InitEDA

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

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

## now let's check for fixed-effect interactions... Since we do not have
## any fixed effects are involved, so the fixed-effect interactions are not needed.
## Finally we check for random effects.  We should only add random effects that
## are also present as fixed effects.  This means, for this model isn't even possible,
## so no testing is needed.


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

summary(tmp)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (1 | Artifact)
##    Data: tall.nonmissing[tall.nonmissing$Rubric == "InitEDA", ]
##
## REML criterion at convergence: 239
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8889 -0.3391 -0.1427  0.4276  1.6035
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Artifact (Intercept) 0.3651   0.6042
##  Residual             0.1655   0.4068
## Number of obs: 116, groups:  Artifact, 90
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  2.44226    0.07537    32.4
```

```
display(tmp)
```

```
## lmer(formula = as.numeric(Rating) ~ (1 | Artifact), data = tall.nonmissing[tall.nonmissing$Rubric ==
##     "InitEDA", ])
## coef.est  coef.se
##     2.44     0.08
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Artifact (Intercept) 0.60
##  Residual             0.41
## ---
## number of obs: 116, groups: Artifact, 90
## AIC = 245, DIC = 232.3
## deviance = 235.6
```

```
formula(tmp)
```

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

```
# ranef(tmp)

## Random effect: deviation from the overall mean (beta 0), ranef() will get the eita, eita zero on the
```

The final Rubric specific model for rubric InitEDA: $as.numeric(Rating)$ $(1|Artifact)$.

- Different artifacts of the total 91 artifacts tend to get different ratings on rubric InitEDA. The mean ratings of the total 91 artifacts on rubric TxtOrg equals to the overall mean ratings on rubric InitEDA which is 2.44226 plus the random effect deviations from the overall mean ratings on rubric InitEDA.

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

**RsrchQ**

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

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

## now let's check for fixed-effect interactions... Since we do not have
## any fixed effects are involved, so the fixed-effect interactions are not needed.

## Finally we check for random effects.  We should only add random effects that
## are also present as fixed effects.  This means, for this model isn't even possible,
## so no testing is needed.


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

summary(tmp)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (1 | Artifact)
##    Data: tall.nonmissing[tall.nonmissing$Rubric == "RsrchQ", ]
##
## REML criterion at convergence: 209.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2694 -0.5285 -0.3736  0.9743  2.4770
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Artifact (Intercept) 0.07276  0.2697
##  Residual             0.27825  0.5275
## Number of obs: 116, groups:  Artifact, 90
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  2.35169    0.05794   40.59
```

```
display(tmp)
```

```
## lmer(formula = as.numeric(Rating) ~ (1 | Artifact), data = tall.nonmissing[tall.nonmissing$Rubric ==
##     "RsrchQ", ])
## coef.est  coef.se
##     2.35     0.06
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Artifact (Intercept) 0.27
##  Residual             0.53
## ---
## number of obs: 116, groups: Artifact, 90
## AIC = 215.1, DIC = 201.3
## deviance = 205.2
```

```
formula(tmp)
```

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

```
# ranef(tmp)

## Random effect: deviation from the overall mean (beta 0), ranef() will get the eita, eita zero on the
```

The final Rubric specific model for rubric RsrchQ: $as.numeric(Rating)$ $(1|Artifact)$.

- Different artifacts of the total 91 artifacts tend to get different ratings on rubric RsrchQ. The mean ratings of the total 91 artifacts on rubric RsrchQ equals to the overall mean ratings on rubric RsrchQ which is 2.35169 plus the random effect deviations from the overall mean ratings on rubric RsrchQ.

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

**TxtOrg**

```
##             Estimate Std. Error t value
## (Intercept)     2.59       0.07   37.93
## looks like we do, so we keep "tmp" as our best model so far...

## now let's check for fixed-effect interactions... Since we do not have
## any fixed effects are involved, so the fixed-effect interactions are not needed.

## Finally we check for random effects.  We should only add random effects that
## are also present as fixed effects.  This means, for this model isn't even possible,
## so no testing is needed.


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

summary(tmp)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (1 | Artifact)
##    Data: tall.nonmissing[tall.nonmissing$Rubric == "TxtOrg", ]
##
## REML criterion at convergence: 247.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3557 -0.7550  0.3834  0.5302  2.4132
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Artifact (Intercept) 0.09371  0.3061
##  Residual             0.39573  0.6291
## Number of obs: 116, groups:  Artifact, 90
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  2.58745    0.06821   37.93
```

```
display(tmp)
```

```
## lmer(formula = as.numeric(Rating) ~ (1 | Artifact), data = tall.nonmissing[tall.nonmissing$Rubric ==
##     "TxtOrg", ])
## coef.est  coef.se
##     2.59     0.07
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Artifact (Intercept) 0.31
##  Residual             0.63
## ---
## number of obs: 116, groups: Artifact, 90
## AIC = 253.5, DIC = 240.5
## deviance = 244.0
```

```
formula(tmp)
```

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

```
# ranef(tmp)

## Random effect: deviation from the overall mean (beta 0), ranef() will get the eita, eita zero on the
```

The final Rubric specific model for rubric TxtOrg: $as.numeric(Rating)$ $(1|Artifact)$.

- Different artifacts of the total 91 artifacts tend to get different ratings on rubric TxtOrg. The mean ratings of the total 91 artifacts on rubric TxtOrg equals to the overall mean ratings on rubric TxtOrg which is 2.58745 plus the random effect deviations from the overall mean ratings on rubric TxtOrg. #### CritDes

```
## refit the model and check on the t-statistics -- do all the variables matter?
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
```

```
## apparently they do.

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

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

```
## Data: tall.nonmissing[tall.nonmissing$Rubric == "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
```

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

## now let's check for fixed-effect interactions... Since only Rater
## is involved, so the fixed-effect interactions are not needed.

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

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

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

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
```

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


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

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
```

```
display(tmp)
```

```
## lmer(formula = as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) -
##     1, data = tall.nonmissing[tall.nonmissing$Rubric == "CritDes",
##     ])
##                   coef.est coef.se
## as.factor(Rater)1 1.69     0.12
## as.factor(Rater)2 2.11     0.12
## as.factor(Rater)3 1.89     0.12
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Artifact (Intercept) 0.66
##  Residual             0.50
## ---
## number of obs: 115, groups: Artifact, 89
## AIC = 281, DIC = 256.3
## deviance = 263.6
```

```
formula(tmp)
```

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

The final Rubric specific model for rubric CritDes: $as.numeric(Rating)\ as.factor(Rater)+(1|Artifact)-1$.

- Considering the same rater, different artifacts of the total 91 artifacts tend to get different ratings on rubric CritDes. The mean ratings of the total 91 artifacts on rubric CritDes equals to the overall mean ratings on rubric CritDes which is 0 plus the random effect deviations from the overall mean ratings on rubric CritDes.
- Considering the same artifact in the same semester, rater1 tends to give the lowest ratings on the rubric CritDes, followed by rater3 tends to give 0.2045 higher ratings than rater1 on the rubric CritDes and rater2 tends to give 0.2221 higher ratings than rater3 on the rubric CritDes for the total of 91 artifacts.

```
## refit the model and check on the t-statistics -- do all the variables matter?
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
```

### InterpRes

```
##                     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
```

```
## apparently they do.

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

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

## Data: tall.nonmissing[tall.nonmissing$Rubric == "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
```

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

## now let's check for fixed-effect interactions... Since only Rater
## is involved, so the fixed-effect interactions are not needed.

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

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

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

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

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

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

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

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

```
## Error in exactRLRT(m0 = m0, mA = mA, m = m): object 'm' not found
## Same thing happened!  Again, the model
##
## as.numeric(Rating) ~ -1 + as.factor(Rater) +
##                            (1 | Artifact) + (as.factor(Rater) | Artifact)
##
## isn't even possible, so no testing is needed.


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

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
```

```
display(tmp)
```

```
## lmer(formula = as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) -
##     1, data = tall.nonmissing[tall.nonmissing$Rubric == "InterpRes",
##     ])
##                   coef.est coef.se
## as.factor(Rater)1 2.70     0.09
## as.factor(Rater)2 2.59     0.09
## as.factor(Rater)3 2.14     0.09
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Artifact (Intercept) 0.25
##  Residual             0.50
## ---
## number of obs: 116, groups: Artifact, 90
## AIC = 209.7, DIC = 181.6
## deviance = 190.7
```

```
formula(tmp)
```

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

The final Rubric specific model for rubric InterpRes: $as.numeric(Rating)\ as.factor(Rater)+(1|Artifact)-1$.

- Considering the same rater, different artifacts of the total 91 artifacts tend to get different ratings on rubric InterpRes. The mean ratings of the total 91 artifacts on rubric InterpRes equals to the overall mean ratings on rubric InterpRes which is 0 plus the random effect deviations from the overall mean ratings on rubric InterpRes.
- Considering the same artifact in the same semester, rater3 tends to give the lowest ratings on the rubric InterpRes, followed by rater2 tends to give 0.44656 higher ratings than rater3 on the rubric InterpRes and rater1 tends to give 0.11847 higher ratings than rater2 on the rubric InterpRes for the total of 91 artifacts.

```
## refit the model and check on the t-statistics -- do all the variables matter?
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
```

**VisOrg**

```
##                    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
```

```
## apparently they do.

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

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

```
## Data: tall.nonmissing[tall.nonmissing$Rubric == "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
```

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

## now let's check for fixed-effect interactions... Since only Rater
## is involved, so the fixed-effect interactions are not needed.

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

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

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

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 f
```

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

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

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


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

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
```

```
display(tmp)
```

```
## lmer(formula = as.numeric(Rating) ~ as.factor(Rater) + (1 | Artifact) -
##     1, data = tall.nonmissing[tall.nonmissing$Rubric == "VisOrg",
##     ])
##                     coef.est coef.se
## as.factor(Rater)1 2.38     0.10
## as.factor(Rater)2 2.65     0.10
## as.factor(Rater)3 2.28     0.10
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Artifact (Intercept) 0.54
##  Residual             0.38
## ---
## number of obs: 115, groups: Artifact, 89
## AIC = 229.6, DIC = 202
## deviance = 210.8
```

```
formula(tmp)
```

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

The final Rubric specific model for rubric VisOrg: $as.numeric(Rating)\ as.factor(Rater) + (1|Artifact) - 1$.

- Considering the same rater, different artifacts of the total 91 artifacts tend to get different ratings on rubric VisOrg. The mean ratings of the total 91 artifacts on rubric VisOrg equals to the overall mean ratings on rubric VisOrg which is 0 plus the random effect deviations from the overall mean ratings on rubric VisOrg.

- Considering the same artifact in the same semester, rater3 tends to give the lowest ratings on the rubric VisOrg, followed by rater1 tends to give 0.09439 higher ratings than rater3 on the rubric VisOrg and rater2 tends to give 0.27097 higher ratings than rater1 on the rubric VisOrg for the total of 91 artifacts.

## Part D

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

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

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

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

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

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

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ 1 + (0 + Rubric | Artifact)
##    Data: tall.nonmissing
##
## REML criterion at convergence: 1471.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0217 -0.4940 -0.0753  0.5271  3.7760
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.64068  0.8004
##           RubricInitEDA   0.38291  0.6188   0.26
##           RubricInterpRes 0.25656  0.5065   0.00 0.79
##           RubricRsrchQ    0.17398  0.4171   0.38 0.50 0.74
##           RubricSelMeth   0.09621  0.3102   0.56 0.37 0.41 0.26
##           RubricTxtOrg    0.40421  0.6358   0.03 0.69 0.80 0.64 0.24
##           RubricVisOrg    0.31877  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)  2.23211    0.04013   55.62
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
display(comb.0)
```

```
## lmer(formula = as.numeric(Rating) ~ 1 + (0 + Rubric | Artifact),
##     data = tall.nonmissing)
## coef.est  coef.se
##     2.23     0.04
##
## Error terms:
##  Groups   Name             Std.Dev. Corr
##  Artifact RubricCritDes    0.80
##           RubricInitEDA    0.62     0.26
##           RubricInterpRes  0.51     0.00 0.79
##           RubricRsrchQ     0.42     0.38 0.50 0.74
##           RubricSelMeth    0.31     0.56 0.37 0.41 0.26
##           RubricTxtOrg     0.64     0.03 0.69 0.80 0.64 0.24
##           RubricVisOrg     0.56     0.17 0.78 0.76 0.60 0.29 0.79
##  Residual                  0.44
## ---
## number of obs: 810, groups: Artifact, 90
## AIC = 1531.7, DIC = 1462.5
## deviance = 1467.1
```

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

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

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

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

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00201393 (tol = 0.002, component 1)
```

```
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 -0.5066 -0.0178  0.5244  3.7932
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.55313  0.7437
##           RubricInitEDA   0.35240  0.5936   0.47
##           RubricInterpRes 0.17510  0.4184   0.23 0.75
##           RubricRsrchQ    0.16996  0.4123   0.58 0.44 0.71
##           RubricSelMeth   0.06816  0.2611   0.39 0.60 0.74 0.41
##           RubricTxtOrg    0.26336  0.5132   0.34 0.62 0.70 0.56 0.67
##           RubricVisOrg    0.25810  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)        2.013760   0.109104  18.457
## as.factor(Rater)2  0.001969   0.054888   0.036
## as.factor(Rater)3 -0.174877   0.055045  -3.177
## SemesterS19       -0.175027   0.087852  -1.992
## SexM               0.010495   0.081272   0.129
## Repeated          -0.073577   0.098523  -0.747
## RubricInitEDA      0.547054   0.095707   5.716
## RubricInterpRes    0.587095   0.100892   5.819
## RubricRsrchQ       0.460875   0.087514   5.266
## RubricSelMeth      0.164861   0.094263   1.749
## RubricTxtOrg       0.692879   0.099519   6.962
## RubricVisOrg       0.530183   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
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00201393 (tol = 0.002, component 1)
```

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

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

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

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

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

```
## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.8871 >= 0.05
##     not part of higher-order interaction
```

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

```
##     removing term
##   iteration 2
##     p-value for term "Repeated" = 0.0919 >= 0.05
##     not part of higher-order interaction
##     removing term
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
## pruning random effects structure ...
##   nothing to prune
```

```
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  3.7753
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.55497  0.7450
##           RubricInitEDA   0.35067  0.5922   0.47
##           RubricInterpRes 0.16891  0.4110   0.23 0.75
##           RubricRsrchQ    0.16780  0.4096   0.59 0.44 0.70
##           RubricSelMeth   0.06498  0.2549   0.40 0.60 0.74 0.40
##           RubricTxtOrg    0.25616  0.5061   0.33 0.61 0.69 0.55 0.66
##           RubricVisOrg    0.25897  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)        2.0084214  0.0987622  20.336
## as.factor(Rater)2  0.0003157  0.0547446   0.006
## as.factor(Rater)3 -0.1771100  0.0548891  -3.227
## SemesterS19       -0.1730497  0.0826935  -2.093
## RubricInitEDA      0.5474745  0.0957124   5.720
## RubricInterpRes    0.5864561  0.1008582   5.815
## RubricRsrchQ       0.4584072  0.0874168   5.244
## RubricSelMeth      0.1590763  0.0937777   1.696
## RubricTxtOrg       0.6930032  0.0995436   6.962
## RubricVisOrg       0.5289037  0.0990922   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
```

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

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

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

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

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00206896 (tol = 0.002, component 1)
```

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

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

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

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

summary(comb.inter.u)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Rubric + as.factor(Rater):Semester + as.factor(Rater):Rubric +
##     Semester:Rubric + as.factor(Rater):Semester:Rubric
##    Data: tall.nonmissing
## Control: lmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
## REML criterion at convergence: 1424.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9141 -0.5141 -0.0653  0.5023  3.6609
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.48550  0.6968
##           RubricInitEDA   0.35257  0.5938   0.42
##           RubricInterpRes 0.14619  0.3824   0.32 0.80
##           RubricRsrchQ    0.16444  0.4055   0.66 0.43 0.72
##           RubricSelMeth   0.06297  0.2509   0.45 0.64 0.78 0.49
##           RubricTxtOrg    0.25441  0.5044   0.44 0.65 0.67 0.60 0.62
##           RubricVisOrg    0.25527  0.5052   0.35 0.73 0.68 0.57 0.35 0.76
##  Residual                 0.18839  0.4340
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
##                                                Estimate Std. Error t value
## (Intercept)                                    1.739538   0.136568  12.738
## as.factor(Rater)2                              0.302994   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    -0.542266   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
```

```
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
```

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

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

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

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular

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

## boundary (singular) fit: see ?isSingular
##     removing term
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## pruning random effects structure ...
##   nothing to prune
## ======================================================
## ===            forwardfitting random effects       ===
## ======================================================
##  ===       random slopes       ===
## ======================================================
## ===            re-backfitting fixed effects        ===
## ======================================================
## processing model terms of interaction level 2
##   all terms of interaction level 2 significant
## processing model terms of interaction level 1
##   all terms of interaction level 1 significant
## resetting REML to TRUE
```

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

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

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

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

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

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

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

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

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

formula(comb.inter.u)
```

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

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

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

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

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

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

```
##  Groups   Name            Std.Dev. Corr
##  Artifact RubricCritDes   0.69678
##           RubricInitEDA   0.59378  0.416
##           RubricInterpRes 0.38235  0.324 0.800
##           RubricRsrchQ    0.40551  0.655 0.430 0.723
##           RubricSelMeth   0.25094  0.446 0.639 0.784 0.488
##           RubricTxtOrg    0.50440  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   0.74496
##           RubricInitEDA   0.59217  0.467
##           RubricInterpRes 0.41099  0.230 0.749
##           RubricRsrchQ    0.40963  0.588 0.436 0.704
##           RubricSelMeth   0.25491  0.399 0.603 0.736 0.397
##           RubricTxtOrg    0.50612  0.335 0.614 0.691 0.551 0.656
##           RubricVisOrg    0.50889  0.350 0.731 0.679 0.516 0.414 0.752
##  Residual                 0.43513
```

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

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

```
## Data: tall.nonmissing
## Models:
## comb.back_elim: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Rubric
## comb.inter_elim: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Rubric
## comb.inter.u: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Rubric + a
##                 npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## comb.back_elim    39 1464.0 1647.2 -693.02   1386.0
## comb.inter_elim   51 1454.5 1694.1 -676.26   1352.5 33.526 12   0.000801 ***
## comb.inter.u      71 1471.4 1804.8 -664.68   1329.4 23.161 20   0.280962
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## the models are nested so we can use AIC, BIC or likelihod ratio (deviance)
## tests... AIC and the LRT agree on comb.inter_elim; BIC likes the simpler
## comb.back_elim.

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

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

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

```
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
```

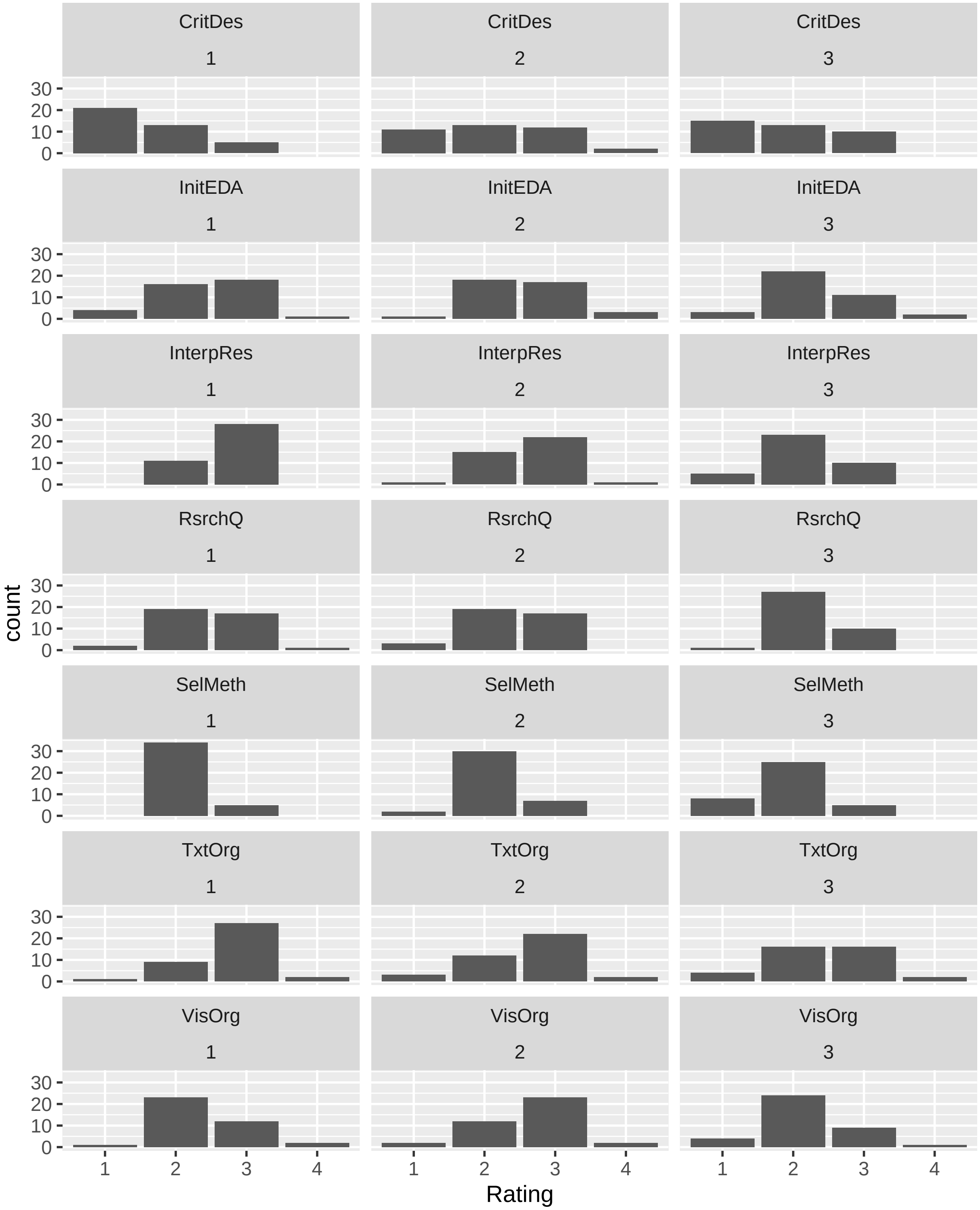
```
display(comb.inter_elim)
```

```
## lmer(formula = as.numeric(Rating) ~ (0 + Rubric | Artifact) +
##     as.factor(Rater) + Semester + Rubric + as.factor(Rater):Rubric,
##     data = tall.nonmissing, REML = TRUE, control = lmerControl(optimizer = "bobyqa",
##         optCtrl = list(maxfun = 2e+05)), start = ss)
##                                   coef.est coef.se
## (Intercept)                        1.76     0.12
## as.factor(Rater)2                  0.37     0.13
## as.factor(Rater)3                  0.21     0.13
## SemesterS19                       -0.18     0.08
## RubricInitEDA                      0.75     0.14
## RubricInterpRes                    1.01     0.13
## RubricRsrchQ                       0.75     0.12
## RubricSelMeth                      0.43     0.13
## RubricTxtOrg                       1.05     0.14
## RubricVisOrg                       0.68     0.14
## as.factor(Rater)2:RubricInitEDA   -0.31     0.17
## as.factor(Rater)3:RubricInitEDA   -0.30     0.17
## as.factor(Rater)2:RubricInterpRes -0.54     0.17
## as.factor(Rater)3:RubricInterpRes -0.75     0.17
## as.factor(Rater)2:RubricRsrchQ    -0.50     0.16
## as.factor(Rater)3:RubricRsrchQ    -0.37     0.16
## as.factor(Rater)2:RubricSelMeth   -0.40     0.16
## as.factor(Rater)3:RubricSelMeth   -0.41     0.17
## as.factor(Rater)2:RubricTxtOrg    -0.58     0.17
## as.factor(Rater)3:RubricTxtOrg    -0.49     0.17
## as.factor(Rater)2:RubricVisOrg    -0.14     0.17
## as.factor(Rater)3:RubricVisOrg    -0.33     0.17
##
## Error terms:
##  Groups   Name             Std.Dev. Corr
##  Artifact RubricCritDes   0.71
##            RubricInitEDA   0.60     0.44
##            RubricInterpRes 0.39     0.35 0.82
##            RubricRsrchQ    0.42     0.63 0.44 0.72
##            RubricSelMeth   0.26     0.42 0.60 0.74 0.36
##            RubricTxtOrg    0.51     0.42 0.64 0.67 0.55 0.64
##            RubricVisOrg    0.50     0.34 0.71 0.68 0.51 0.38 0.77
##  Residual                  0.43
## ---
## number of obs: 810, groups: Artifact, 90
## AIC = 1521.6, DIC = 1285.4
## deviance = 1352.5
```

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

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

g
```



```
## and it does look as if the 3 raters have different ways of scoring the 7 rubrics,
## so the interaction we found in comb.inter_elim makes sense.  (Clearly it is
## not the case that one rater is simply more harsh than another, or something
## like that.
```

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

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

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

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

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00493103 (tol = 0.002, component 1)
```

```r
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(Rate
##    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
```

```r
##
## 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)
```

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

```
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(Rate
```

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

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

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

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00493103 (tol = 0.002, component 1)
```

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

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

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

```
##  Groups     Name             Std.Dev. Corr
##  Artifact   RubricCritDes    0.70447
##             RubricInitEDA    0.56380   0.318
##              RubricInterpRes   0.31944   0.142  0.674
##              RubricRsrchQ      0.42306   0.500  0.194  0.538
##              RubricSelMeth     0.19554   0.145  0.227  0.376 -0.241
##              RubricTxtOrg      0.50027   0.268  0.437  0.364  0.305  0.213
##              RubricVisOrg      0.48205   0.175  0.504  0.445  0.276 -0.160
##  Artifact.1 as.factor(Rater)1 0.11322
##              as.factor(Rater)2 0.33430  -0.486
##              as.factor(Rater)3 0.30679   0.332  0.663
##  Residual                      0.36700
##
##
##
##
##
##
##
##   0.537
##
##
##
##
```

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

```
##                                      Estimate Std. Error    t value
## (Intercept)                         1.7575357 0.11402967 15.4129676
## as.factor(Rater)2                   0.3660743 0.13917859  2.6302488
## as.factor(Rater)3                   0.1959298 0.12965892  1.5111170
## SemesterS19                        -0.1591747 0.07647292 -2.0814524
## RubricInitEDA                       0.7395208 0.12995961  5.6903895
## RubricInterpRes                     0.9915188 0.12770181  7.7643286
## RubricRsrchQ                        0.7262014 0.11791907  6.1584732
## RubricSelMeth                       0.4107115 0.12469405  3.2937535
## RubricTxtOrg                        1.0157913 0.12999164  7.8142821
## RubricVisOrg                        0.6542375 0.13353097  4.8995188
## as.factor(Rater)2:RubricInitEDA    -0.2998406 0.15609130 -1.9209308
## as.factor(Rater)3:RubricInitEDA    -0.2947790 0.15635257 -1.8853480
## as.factor(Rater)2:RubricInterpRes -0.5132331 0.15348295 -3.3439094
## as.factor(Rater)3:RubricInterpRes -0.7148403 0.15363779 -4.6527632
## as.factor(Rater)2:RubricRsrchQ    -0.4874343 0.14721456 -3.3110472
## as.factor(Rater)3:RubricRsrchQ    -0.3224062 0.14725825 -2.1893929
## as.factor(Rater)2:RubricSelMeth   -0.3864167 0.15030393 -2.5709018
## as.factor(Rater)3:RubricSelMeth   -0.3871985 0.14960917 -2.5880668
## as.factor(Rater)2:RubricTxtOrg    -0.5510611 0.15645949 -3.5220690
## as.factor(Rater)3:RubricTxtOrg    -0.4449033 0.15673034 -2.8386545
## as.factor(Rater)2:RubricVisOrg    -0.1048823 0.15861238 -0.6612494
## as.factor(Rater)3:RubricVisOrg    -0.2751871 0.15885035 -1.7323667
```

```
summary(comb.final)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## as.numeric(Rating) ~ (0 + Rubric | Artifact) + (0 + as.factor(Rater) |
##     Artifact) + as.factor(Rater) + Semester + Rubric + as.factor(Rater):Rubric
##    Data: tall.nonmissing
##
## REML criterion at convergence: 1370.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.06443 -0.46911 -0.02987  0.45353  2.74012
##
## Random effects:
##  Groups     Name              Variance Std.Dev. Corr
##  Artifact   RubricCritDes     0.49628  0.7045
##             RubricInitEDA     0.31787  0.5638    0.32
##             RubricInterpRes   0.10204  0.3194    0.14  0.67
##             RubricRsrchQ      0.17898  0.4231    0.50  0.19  0.54
##             RubricSelMeth     0.03823  0.1955    0.14  0.23  0.38 -0.24
##             RubricTxtOrg      0.25027  0.5003    0.27  0.44  0.36  0.31  0.21
##             RubricVisOrg      0.23237  0.4821    0.17  0.50  0.45  0.28 -0.16
##  Artifact.1 as.factor(Rater)1 0.01282  0.1132
##             as.factor(Rater)2 0.11176  0.3343   -0.49
##             as.factor(Rater)3 0.09412  0.3068    0.33  0.66
##  Residual                     0.13469  0.3670
##
##
##
##
##
##
##
##   0.54
##
##
##
##
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
##                                   Estimate Std. Error t value
## (Intercept)                        1.75754    0.11403  15.413
## as.factor(Rater)2                  0.36607    0.13918   2.630
## as.factor(Rater)3                  0.19593    0.12966   1.511
## SemesterS19                       -0.15917    0.07647  -2.081
## RubricInitEDA                      0.73952    0.12996   5.690
## RubricInterpRes                    0.99152    0.12770   7.764
## RubricRsrchQ                       0.72620    0.11792   6.158
## RubricSelMeth                      0.41071    0.12469   3.294
## RubricTxtOrg                       1.01579    0.12999   7.814
## RubricVisOrg                       0.65424    0.13353   4.900
## as.factor(Rater)2:RubricInitEDA   -0.29984    0.15609  -1.921
## as.factor(Rater)3:RubricInitEDA   -0.29478    0.15635  -1.885
## as.factor(Rater)2:RubricInterpRes -0.51323    0.15348  -3.344
## as.factor(Rater)3:RubricInterpRes -0.71484    0.15364  -4.653
## as.factor(Rater)2:RubricRsrchQ    -0.48743    0.14721  -3.311
## as.factor(Rater)3:RubricRsrchQ    -0.32241    0.14726  -2.189
## as.factor(Rater)2:RubricSelMeth   -0.38642    0.15030  -2.571
## as.factor(Rater)3:RubricSelMeth   -0.38720    0.14961  -2.588
## as.factor(Rater)2:RubricTxtOrg    -0.55106    0.15646  -3.522
## as.factor(Rater)3:RubricTxtOrg    -0.44490    0.15673  -2.839
## as.factor(Rater)2:RubricVisOrg    -0.10488    0.15861  -0.661
## as.factor(Rater)3:RubricVisOrg    -0.27519    0.15885  -1.732
```

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

```
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00493103 (tol = 0.002, component 1)
```

```
display(comb.final)
```

```
## lmer(formula = as.numeric(Rating) ~ (0 + Rubric | Artifact) +
##     (0 + as.factor(Rater) | Artifact) + as.factor(Rater) + Semester +
##     Rubric + as.factor(Rater):Rubric, data = tall.nonmissing)
##                                   coef.est coef.se
## (Intercept)                        1.76     0.11
## as.factor(Rater)2                  0.37     0.14
## as.factor(Rater)3                  0.20     0.13
## SemesterS19                       -0.16     0.08
## RubricInitEDA                      0.74     0.13
## RubricInterpRes                    0.99     0.13
## RubricRsrchQ                       0.73     0.12
## RubricSelMeth                      0.41     0.12
## RubricTxtOrg                       1.02     0.13
## RubricVisOrg                       0.65     0.13
## as.factor(Rater)2:RubricInitEDA   -0.30     0.16
## as.factor(Rater)3:RubricInitEDA   -0.29     0.16
## as.factor(Rater)2:RubricInterpRes -0.51     0.15
## as.factor(Rater)3:RubricInterpRes -0.71     0.15
## as.factor(Rater)2:RubricRsrchQ    -0.49     0.15
## as.factor(Rater)3:RubricRsrchQ    -0.32     0.15
## as.factor(Rater)2:RubricSelMeth   -0.39     0.15
## as.factor(Rater)3:RubricSelMeth   -0.39     0.15
## as.factor(Rater)2:RubricTxtOrg    -0.55     0.16
## as.factor(Rater)3:RubricTxtOrg    -0.44     0.16
## as.factor(Rater)2:RubricVisOrg    -0.10     0.16
## as.factor(Rater)3:RubricVisOrg    -0.28     0.16
##
## Error terms:
##  Groups     Name             Std.Dev. Corr
##  Artifact   RubricCritDes    0.70
##             RubricInitEDA    0.56      0.32
##             RubricInterpRes  0.32      0.14  0.67
##             RubricRsrchQ     0.42      0.50  0.19  0.54
##             RubricSelMeth    0.20      0.14  0.23  0.38 -0.24
##             RubricTxtOrg     0.50      0.27  0.44  0.36  0.31  0.21
##             RubricVisOrg     0.48      0.17  0.50  0.45  0.28 -0.16  0.54
##  Artifact.1 as.factor(Rater)1 0.11
##             as.factor(Rater)2 0.33     -0.49
##             as.factor(Rater)3 0.31      0.33  0.66
##  Residual                     0.37
```

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

```
## ---
## number of obs: 810, groups: Artifact, 90
## AIC = 1484.6, DIC = 1233.2
## deviance = 1301.9
```

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

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

**Do you find that any of these fixed effects have a significant effect in predicting ratings? Are there any other random effects that you can justify adding to these models?**

- In conclusion, after doing variable selection for all the fixed effects, random effects, fixed-effect interactions and random-effect interactions based on all of the variables Rater, Semester, Sex, Repeated and/or Rubric, we got the final "combined" model as interpreted from Section 4.3, Part 4. Hence, we concluded that there are three fixed effects have a significant effect in predicting rating, they are

Rater, Semester and Rubric; there are two random effects that we can justify adding to the model, they are Rubric and Rater. However, there is a fixed-effect interaction has a significant effect in predicting ratings, which is the fixed interaction between Rater and Rubric; and there is no random interaction has a significant effect in predicting ratings.

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

```
##                                     Estimate Std. Error    t value
## (Intercept)                        1.7594507 0.11785119 14.9294272
## as.factor(Rater)2                  0.3653710 0.13295661  2.7480470
## as.factor(Rater)3                  0.2142107 0.13297242  1.6109410
## SemesterS19                       -0.1777990 0.08227893 -2.1609295
## RubricInitEDA                      0.7462463 0.13675930  5.4566400
## RubricInterpRes                    1.0145336 0.13478613  7.5269880
## RubricRsrchQ                       0.7492623 0.12419095  6.0331472
## RubricSelMeth                      0.4267187 0.13039725  3.2724513
## RubricTxtOrg                       1.0496707 0.13551082  7.7460289
## RubricVisOrg                       0.6835365 0.13947307  4.9008496
## as.factor(Rater)2:RubricInitEDA   -0.3084278 0.17249386 -1.7880511
## as.factor(Rater)3:RubricInitEDA   -0.2952153 0.17282355 -1.7081894
## as.factor(Rater)2:RubricInterpRes -0.5367419 0.17007960 -3.1558275
## as.factor(Rater)3:RubricInterpRes -0.7524695 0.17048676 -4.4136533
## as.factor(Rater)2:RubricRsrchQ    -0.5015691 0.16150818 -3.1055335
## as.factor(Rater)3:RubricRsrchQ    -0.3706753 0.16179322 -2.2910434
## as.factor(Rater)2:RubricSelMeth   -0.3960247 0.16466832 -2.4049845
## as.factor(Rater)3:RubricSelMeth   -0.4132361 0.16503757 -2.5038912
## as.factor(Rater)2:RubricTxtOrg    -0.5838002 0.17140825 -3.4059050
## as.factor(Rater)3:RubricTxtOrg    -0.4864856 0.17177161 -2.8321652
## as.factor(Rater)2:RubricVisOrg    -0.1444388 0.17442083 -0.8281055
## as.factor(Rater)3:RubricVisOrg    -0.3338015 0.17480764 -1.9095361
```

```
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
## -3.0217 -0.4940 -0.0753  0.5271  3.7760
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.64068  0.8004
##           RubricInitEDA   0.38291  0.6188   0.26
##           RubricInterpRes 0.25656  0.5065   0.00 0.79
##           RubricRsrchQ    0.17398  0.4171   0.38 0.50 0.74
##           RubricSelMeth   0.09621  0.3102   0.56 0.37 0.41 0.26
##           RubricTxtOrg    0.40421  0.6358   0.03 0.69 0.80 0.64 0.24
##           RubricVisOrg    0.31877  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)  2.23211    0.04013   55.62
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

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

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

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

```
##  Groups     Name              Std.Dev. Corr
##  Artifact   RubricCritDes     0.70447
##             RubricInitEDA     0.56380   0.318
##             RubricInterpRes   0.31944   0.142  0.674
##             RubricRsrchQ      0.42306   0.500  0.194  0.538
##             RubricSelMeth     0.19554   0.145  0.227  0.376 -0.241
##             RubricTxtOrg      0.50027   0.268  0.437  0.364  0.305  0.213
##             RubricVisOrg      0.48205   0.175  0.504  0.445  0.276 -0.160
##  Artifact.1 as.factor(Rater)1 0.11322
##             as.factor(Rater)2 0.33430  -0.486
##             as.factor(Rater)3 0.30679   0.332  0.663
##  Residual                     0.36700
##
##
##
##
##
##
##
##   0.537
##
##
##
##
```

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

```
##                                      Estimate Std. Error    t value
## (Intercept)                         1.7575357 0.11402967 15.4129676
## as.factor(Rater)2                   0.3660743 0.13917859  2.6302488
## as.factor(Rater)3                   0.1959298 0.12965892  1.5111170
## SemesterS19                        -0.1591747 0.07647292 -2.0814524
## RubricInitEDA                       0.7395208 0.12995961  5.6903895
## RubricInterpRes                     0.9915188 0.12770181  7.7643286
## RubricRsrchQ                        0.7262014 0.11791907  6.1584732
## RubricSelMeth                       0.4107115 0.12469405  3.2937535
## RubricTxtOrg                        1.0157913 0.12999164  7.8142821
## RubricVisOrg                       0.6542375 0.13353097  4.8995188
## as.factor(Rater)2:RubricInitEDA   -0.2998406 0.15609130 -1.9209308
## as.factor(Rater)3:RubricInitEDA   -0.2947790 0.15635257 -1.8853480
## as.factor(Rater)2:RubricInterpRes -0.5132331 0.15348295 -3.3439094
## as.factor(Rater)3:RubricInterpRes -0.7148403 0.15363779 -4.6527632
## as.factor(Rater)2:RubricRsrchQ    -0.4874343 0.14721456 -3.3110472
## as.factor(Rater)3:RubricRsrchQ    -0.3224062 0.14725825 -2.1893929
## as.factor(Rater)2:RubricSelMeth   -0.3864167 0.15030393 -2.5709018
## as.factor(Rater)3:RubricSelMeth   -0.3871985 0.14960917 -2.5880668
## as.factor(Rater)2:RubricTxtOrg    -0.5510611 0.15645949 -3.5220690
## as.factor(Rater)3:RubricTxtOrg    -0.4449033 0.15673034 -2.8386545
## as.factor(Rater)2:RubricVisOrg    -0.1048823 0.15861238 -0.6612494
## as.factor(Rater)3:RubricVisOrg    -0.2751871 0.15885035 -1.7323667
```

```
summary(comb.final)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: 
## as.numeric(Rating) ~ (0 + Rubric | Artifact) + (0 + as.factor(Rater) |  
##     Artifact) + as.factor(Rater) + Semester + Rubric + as.factor(Rater):Rubric
##    Data: tall.nonmissing
## 
## REML criterion at convergence: 1370.6
## 
## Scaled residuals: 
##      Min       1Q   Median       3Q      Max 
## -3.06443 -0.46911 -0.02987  0.45353  2.74012 
## 
## Random effects:
##  Groups     Name              Variance Std.Dev. Corr                   
##  Artifact   RubricCritDes     0.49628  0.7045                          
##             RubricInitEDA     0.31787  0.5638    0.32                  
##             RubricInterpRes   0.10204  0.3194    0.14  0.67            
##             RubricRsrchQ      0.17898  0.4231    0.50  0.19  0.54      
##             RubricSelMeth     0.03823  0.1955    0.14  0.23  0.38 -0.24
##             RubricTxtOrg      0.25027  0.5003    0.27  0.44  0.36  0.31  0.21
##             RubricVisOrg      0.23237  0.4821    0.17  0.50  0.45  0.28 -0.16
##  Artifact.1 as.factor(Rater)1 0.01282  0.1132                          
##             as.factor(Rater)2 0.11176  0.3343   -0.49                  
##             as.factor(Rater)3 0.09412  0.3068    0.33  0.66            
##  Residual                     0.13469  0.3670                          
##                                                                        
##                                                                        
##                                                                        
##                                                                        
##                                                                        
##                                                                        
##   0.54                                                                 
##                                                                        
##                                                                        
##                                                                        
##                                                                        
## Number of obs: 810, groups:  Artifact, 90
```

```
##
## Fixed effects:
##                                   Estimate Std. Error t value
## (Intercept)                        1.75754    0.11403  15.413
## as.factor(Rater)2                  0.36607    0.13918   2.630
## as.factor(Rater)3                  0.19593    0.12966   1.511
## SemesterS19                       -0.15917    0.07647  -2.081
## RubricInitEDA                      0.73952    0.12996   5.690
## RubricInterpRes                    0.99152    0.12770   7.764
## RubricRsrchQ                       0.72620    0.11792   6.158
## RubricSelMeth                      0.41071    0.12469   3.294
## RubricTxtOrg                       1.01579    0.12999   7.814
## RubricVisOrg                       0.65424    0.13353   4.900
## as.factor(Rater)2:RubricInitEDA   -0.29984    0.15609  -1.921
## as.factor(Rater)3:RubricInitEDA   -0.29478    0.15635  -1.885
## as.factor(Rater)2:RubricInterpRes -0.51323    0.15348  -3.344
## as.factor(Rater)3:RubricInterpRes -0.71484    0.15364  -4.653
## as.factor(Rater)2:RubricRsrchQ    -0.48743    0.14721  -3.311
## as.factor(Rater)3:RubricRsrchQ    -0.32241    0.14726  -2.189
## as.factor(Rater)2:RubricSelMeth   -0.38642    0.15030  -2.571
## as.factor(Rater)3:RubricSelMeth   -0.38720    0.14961  -2.588
## as.factor(Rater)2:RubricTxtOrg    -0.55106    0.15646  -3.522
## as.factor(Rater)3:RubricTxtOrg    -0.44490    0.15673  -2.839
## as.factor(Rater)2:RubricVisOrg    -0.10488    0.15861  -0.661
## as.factor(Rater)3:RubricVisOrg    -0.27519    0.15885  -1.732
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00493103 (tol = 0.002, component 1)
```

```
display(comb.final)
```

```
## lmer(formula = as.numeric(Rating) ~ (0 + Rubric | Artifact) + 
##     (0 + as.factor(Rater) | Artifact) + as.factor(Rater) + Semester + 
##     Rubric + as.factor(Rater):Rubric, data = tall.nonmissing)
##                                   coef.est coef.se
## (Intercept)                        1.76     0.11  
## as.factor(Rater)2                  0.37     0.14  
## as.factor(Rater)3                  0.20     0.13  
## SemesterS19                       -0.16     0.08  
## RubricInitEDA                      0.74     0.13  
## RubricInterpRes                    0.99     0.13  
## RubricRsrchQ                       0.73     0.12  
## RubricSelMeth                      0.41     0.12  
## RubricTxtOrg                       1.02     0.13  
## RubricVisOrg                       0.65     0.13  
## as.factor(Rater)2:RubricInitEDA   -0.30     0.16  
## as.factor(Rater)3:RubricInitEDA   -0.29     0.16  
## as.factor(Rater)2:RubricInterpRes -0.51     0.15  
## as.factor(Rater)3:RubricInterpRes -0.71     0.15  
## as.factor(Rater)2:RubricRsrchQ    -0.49     0.15  
## as.factor(Rater)3:RubricRsrchQ    -0.32     0.15  
## as.factor(Rater)2:RubricSelMeth   -0.39     0.15  
## as.factor(Rater)3:RubricSelMeth   -0.39     0.15  
## as.factor(Rater)2:RubricTxtOrg    -0.55     0.16  
## as.factor(Rater)3:RubricTxtOrg    -0.44     0.16  
## as.factor(Rater)2:RubricVisOrg    -0.10     0.16
## as.factor(Rater)3:RubricVisOrg    -0.28     0.16
##
## Error terms:
##  Groups     Name               Std.Dev. Corr
##  Artifact   RubricCritDes      0.70
##             RubricInitEDA      0.56      0.32
##             RubricInterpRes    0.32      0.14  0.67
##             RubricRsrchQ       0.42      0.50  0.19  0.54
##             RubricSelMeth      0.20      0.14  0.23  0.38 -0.24
##             RubricTxtOrg       0.50      0.27  0.44  0.36  0.31  0.21
##             RubricVisOrg       0.48      0.17  0.50  0.45  0.28 -0.16  0.54
##  Artifact.1 as.factor(Rater)1 0.11
##             as.factor(Rater)2 0.33     -0.49
##             as.factor(Rater)3 0.31      0.33  0.66
##  Residual                      0.37
## ---
## number of obs: 810, groups: Artifact, 90
## AIC = 1484.6, DIC = 1233.2
## deviance = 1301.9
```

**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?**

1) Based on the summary of the “combined” intercept-only model, we have found that: What this typically means is that some of the random effects are highly correlated with one another. We can see this in the “Random effects” block of summary(comb.0):

- The random effects for VisOrg and TxtOrg seem highly correlated with each other and with everything except for the random effects for CritDes and SelMeth.
- The random effects for InterpRes and InitEDA are highly correlated.
- The random effects for SelMeth and CritDes are highly correlated.
- The random effects for RsrchQ and InterpRes are highly correlated, etc.

In some ways we should not be surprised: these rubrics all represent features of a good research report, and we would expect that if someone is good at one or two of these features, they are probably good at the others.

2) In conclusion, we accept comb.final as our final model, we can interpret the pieces as follows:

**Semester** Considering the same artifact rated by the same rater, ratings on the all the seven rubrics for rating Freshman Statistics projects tend to be 0.1591747 lower on Semester Spring 19 than on Semester Fall 19.

$(0 + Rubric|Artifact) + Rubric$ There is a kind of Rubric x Artifact interaction: There are different average scores on each rubric, but the rubric averages also vary a bit from one Artifact to the next, by a small random effect that depends on Artifact.

In all of this, the fact that Rubric scores depend on Artifact (that is, there is a kind of Rubric x Artifact interaction) is what we might expect: the artifacts aren’t all of equal quality on each rubric, and so we should expect the average scores on each Rubric to vary from one Artifact to the next.

$(0 + as.factor(Rater)|Artifact) + as.factor(Rater)$ There is a kind of Rater x Artifact interaction: each Rater's rating on each Artifact differs from what we would expect (from the fixed effects alone) by a small random effect that depends on the Artifact.

$Rubric + as.factor(Rater) + as.factor(Rater) : Rubric$ There is a Rater x Rubric interaction: each Rater uses each Rubric in a way that is not like, or even parallel to, other rater's Rubric usage. (we saw that in the facets plot above also).

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

In addition to looking at the fixed effect coefficients in summary(comb.inter_elim)$coef, we could also see if there's a pattern in an appropriate facets plot. Based on the facets plot, we have found that it does look as if the 3 raters have different ways of scoring the 7 rubrics, so the interaction we found in comb.inter_elim makes sense. Clearly it is not the case that one rater is simply more harsh than another, or something like that.

Relate the different patterns of scoring among the 3 raters to coefficient estimates in summary(comb.final)$coef: They are arriving the same conclusions. As explained below:

In order to interpret, we made an assumption as before, among all rubrics, we defined low rating as artifact was rated less and equal to 2; high rating as artifact was rated above 2.

### Across Rubrics

#### CritDes

- Based on the patterns of scoring among the 3 raters on rubric CritDes, we can clearly see the number of artifacts rated at both grade 1 and 2 by rater1 are both higher than by rater2 and rater3. Besides, the number of artifacts rated at both grade 3 and 4 by rater2 are both higher than by rater1 and rater3. Hence rater1 tends to give the lowest ratings on the rubric CritDes, followed by rater3, and rater2 tends to give the highest ratings on the rubric CritDes.
- Based on the coefficient estimates in summary(comb.final)$coef, considering the same artifact in the same semester (Semester Spring 19 or Semester Fall 19), rater1 tends to give the lowest ratings on the rubric CritDes, followed by rater3 tends to give 0.1959298 higher ratings than rater1 on the rubric CritDes and rater2 tends to give 0.1701445 higher ratings than rater3 on the rubric CritDes for the total of 91 artifacts.

#### InitEDA

- Based on the patterns of scoring among the 3 raters on rubric InitEDA, we can clearly see the number of artifacts rated at both grade 1 and 2 by rater3 are both higher than by rater1 and rater2. Besides, the number of artifacts rated at both grade 3 and 4 by rater2 are both higher than by rater1 and rater3. Hence rater3 tends to give the lowest ratings on the rubric InitEDA, followed by rater1, and rater2 tends to give the highest ratings on the rubric InitEDA.
- Based on the coefficient estimates in summary(comb.final)$coef, considering the same artifact in the same semester (Semester Spring 19 or Semester Fall 19), rater3 tends to give the lowest ratings on the rubric InitEDA, followed by rater1 tends to give 0.0988492 higher ratings than rater3 on the rubric InitEDA and rater2 tends to give 0.0662337 higher ratings than rater1 on the rubric InitEDA for the total of 91 artifacts.

### InterpRes

- Based on the patterns of scoring among the 3 raters on rubric InterpRes, we can clearly see the number of artifacts rated at both grade 1 and 2 by rater3 are both higher than by rater1 and rater2. Besides, the number of artifacts rated at both grade 3 and 4 by rater1 are both higher than by rater2 and rater3. Hence rater3 tends to give the lowest ratings on the rubric InterpRes, followed by rater2, and rater1 tends to give the highest ratings on the rubric InterpRes.
- Based on the coefficient estimates in summary(comb.final)$coef, considering the same artifact in the same semester (Semester Spring 19 or Semester Fall 19), rater3 tends to give the lowest ratings on the rubric InterpRes, followed by rater2 tends to give 0.3717517 higher ratings than rater3 on the rubric InterpRes and rater1 tends to give 0.1471588 higher ratings than rater2 on the rubric InterpRes for the total of 91 artifacts.

### RsrchQ

- Based on the patterns of scoring among the 3 raters on rubric RsrchQ, we can clearly see the number of artifacts rated at both grade 1 and 2 by rater3 are both higher than by rater1 and rater2. Besides, the number of artifacts rated at both grade 3 and 4 by rater1 are both higher than by rater2 and rater3. Hence rater3 tends to give the lowest ratings on the rubric RsrchQ, followed by rater2, and rater1 tends to give the highest ratings on the rubric RsrchQ.
- Based on the coefficient estimates in summary(comb.final)$coef, considering the same artifact in the same semester (Semester Spring 19 or Semester Fall 19), rater3 tends to give the lowest ratings on the rubric RsrchQ, followed by rater2 tends to give 0.0051164 higher ratings than rater3 on the rubric RsrchQ and rater1 tends to give 0.12136 higher ratings than rater2 on the rubric RsrchQ for the total of 91 artifacts.

### SelMeth

- Based on the patterns of scoring among the 3 raters on rubric SelMeth, we can clearly see the number of artifacts rated at both grade 1 and 2 by rater3 are both higher than by rater1 and rater2. Besides, the number of artifacts rated at both grade 3 and 4 by rater1 are both higher than by rater2 and rater3. Hence rater3 tends to give the lowest ratings on the rubric SelMeth, followed by rater2, and rater1 tends to give the highest ratings on the rubric SelMeth.
- Based on the coefficient estimates in summary(comb.final)$coef, considering the same artifact in the same semester (Semester Spring 19 or Semester Fall 19), rater3 tends to give the lowest ratings on the rubric SelMeth, followed by rater2 tends to give 0.0211242 higher ratings than rater3 on the rubric SelMeth and rater1 tends to give 0.0203424 higher ratings than rater2 on the rubric SelMeth for the total of 91 artifacts.

### TxtOrg

- Based on the patterns of scoring among the 3 raters on rubric TxtOrg , we can clearly see the number of artifacts rated at both grade 1 and 2 by rater3 are both higher than by rater1 and rater2. Besides, the number of artifacts rated at both grade 3 and 4 by rater1 are both higher than by rater2 and rater3. Hence rater3 tends to give the lowest ratings on the rubric TxtOrg, followed by rater2, and rater1 tends to give the highest ratings on the rubric TxtOrg.
- Based on the coefficient estimates in summary(comb.final)$coef, considering the same artifact in the same semester (Semester Spring 19 or Semester Fall 19), rater3 tends to give the lowest ratings on the rubric TxtOrg, followed by rater2 tends to give 0.0639867 higher ratings than rater3 on the rubric TxtOrg and rater1 tends to give 0.1849868 higher ratings than rater2 on the rubric TxtOrg for the total of 91 artifacts.

### VisOrg

- Based on the patterns of scoring among the 3 raters on rubric VisOrg , we can clearly see the number of artifacts rated at both grade 1 and 2 by rater3 are both higher than by rater1 and rater2. Besides, the number of artifacts rated at both grade 3 and 4 by rater2 are both higher than by rater1 and rater3. Hence rater3 tends to give the lowest ratings on the rubric VisOrg, followed by rater2, and rater1 tends to give the highest ratings on the rubric VisOrg.
- Based on the coefficient estimates in summary(comb.final)$coef, considering the same artifact in the same semester (Semester Spring 19 or Semester Fall 19), rater3 tends to give the lowest ratings on the rubric VisOrg, followed by rater1 tends to give 0.0792573 higher ratings than rater3 on the rubric VisOrg and rater2 tends to give 0.261192 higher ratings than rater1 on the rubric VisOrg for the total of 91 artifacts.

**Across Raters** Based on the different patterns of scoring among the 3 raters and the coefficient estimates in summary(comb.final)$coef, considering the all of the 91 artifacts in the same semester (Semester Spring 19 or Semester Fall 19), rater3 tends to give especially low ratings, rater1 tends to give higher ratings, which is not match with our conclusions from Appendix 2, Part B.

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.

## Part E

### Do the ICC's from these models agree with your earlier ICC's?

```
summary(comb.final)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## as.numeric(Rating) ~ (0 + Rubric | Artifact) + (0 + as.factor(Rater) |
##     Artifact) + as.factor(Rater) + Semester + Rubric + as.factor(Rater):Rubric
##    Data: tall.nonmissing
##
## REML criterion at convergence: 1370.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.06443 -0.46911 -0.02987  0.45353  2.74012
##
## Random effects:
##  Groups     Name              Variance Std.Dev. Corr
##  Artifact   RubricCritDes     0.49628  0.7045
##             RubricInitEDA     0.31787  0.5638    0.32
##             RubricInterpRes   0.10204  0.3194    0.14  0.67
##             RubricRsrchQ      0.17898  0.4231    0.50  0.19  0.54
##             RubricSelMeth     0.03823  0.1955    0.14  0.23  0.38 -0.24
##             RubricTxtOrg      0.25027  0.5003    0.27  0.44  0.36  0.31  0.21
##             RubricVisOrg      0.23237  0.4821    0.17  0.50  0.45  0.28 -0.16
##  Artifact.1 as.factor(Rater)1 0.01282  0.1132
##             as.factor(Rater)2 0.11176  0.3343   -0.49
##             as.factor(Rater)3 0.09412  0.3068    0.33  0.66
##  Residual                     0.13469  0.3670
##
##
##
##
##
##
##
##   0.54
##
##
##
##
## Number of obs: 810, groups:  Artifact, 90
##
## Fixed effects:
##                                   Estimate Std. Error t value
## (Intercept)                        1.75754    0.11403  15.413
## as.factor(Rater)2                  0.36607    0.13918   2.630
## as.factor(Rater)3                  0.19593    0.12966   1.511
## SemesterS19                       -0.15917    0.07647  -2.081
## RubricInitEDA                      0.73952    0.12996   5.690
## RubricInterpRes                    0.99152    0.12770   7.764
## RubricRsrchQ                       0.72620    0.11792   6.158
## RubricSelMeth                      0.41071    0.12469   3.294
## RubricTxtOrg                       1.01579    0.12999   7.814
## RubricVisOrg                       0.65424    0.13353   4.900
## as.factor(Rater)2:RubricInitEDA   -0.29984    0.15609  -1.921
## as.factor(Rater)3:RubricInitEDA   -0.29478    0.15635  -1.885
## as.factor(Rater)2:RubricInterpRes -0.51323    0.15348  -3.344
## as.factor(Rater)3:RubricInterpRes -0.71484    0.15364  -4.653
## as.factor(Rater)2:RubricRsrchQ    -0.48743    0.14721  -3.311
## as.factor(Rater)3:RubricRsrchQ    -0.32241    0.14726  -2.189
## as.factor(Rater)2:RubricSelMeth   -0.38642    0.15030  -2.571
## as.factor(Rater)3:RubricSelMeth   -0.38720    0.14961  -2.588
## as.factor(Rater)2:RubricTxtOrg    -0.55106    0.15646  -3.522
## as.factor(Rater)3:RubricTxtOrg    -0.44490    0.15673  -2.839
## as.factor(Rater)2:RubricVisOrg    -0.10488    0.15861  -0.661
## as.factor(Rater)3:RubricVisOrg    -0.27519    0.15885  -1.732
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00493103 (tol = 0.002, component 1)
```

```
display(comb.final)
```

```
## lmer(formula = as.numeric(Rating) ~ (0 + Rubric | Artifact) + 
##     (0 + as.factor(Rater) | Artifact) + as.factor(Rater) + Semester + 
##     Rubric + as.factor(Rater):Rubric, data = tall.nonmissing)
##                                   coef.est coef.se
## (Intercept)                        1.76     0.11  
## as.factor(Rater)2                  0.37     0.14  
## as.factor(Rater)3                  0.20     0.13  
## SemesterS19                       -0.16     0.08
## RubricInitEDA                      0.74     0.13
## RubricInterpRes                    0.99     0.13
## RubricRsrchQ                       0.73     0.12
## RubricSelMeth                      0.41     0.12
## RubricTxtOrg                       1.02     0.13
## RubricVisOrg                       0.65     0.13
## as.factor(Rater)2:RubricInitEDA   -0.30     0.16
## as.factor(Rater)3:RubricInitEDA   -0.29     0.16
## as.factor(Rater)2:RubricInterpRes -0.51     0.15
## as.factor(Rater)3:RubricInterpRes -0.71     0.15
## as.factor(Rater)2:RubricRsrchQ    -0.49     0.15
## as.factor(Rater)3:RubricRsrchQ    -0.32     0.15
## as.factor(Rater)2:RubricSelMeth   -0.39     0.15
## as.factor(Rater)3:RubricSelMeth   -0.39     0.15
## as.factor(Rater)2:RubricTxtOrg    -0.55     0.16
## as.factor(Rater)3:RubricTxtOrg    -0.44     0.16
## as.factor(Rater)2:RubricVisOrg    -0.10     0.16
## as.factor(Rater)3:RubricVisOrg    -0.28     0.16
##
## Error terms:
##  Groups     Name              Std.Dev. Corr
##  Artifact   RubricCritDes     0.70
##             RubricInitEDA     0.56      0.32
##             RubricInterpRes   0.32      0.14  0.67
##             RubricRsrchQ      0.42      0.50  0.19  0.54
##             RubricSelMeth     0.20      0.14  0.23  0.38 -0.24
##             RubricTxtOrg      0.50      0.27  0.44  0.36  0.31  0.21
##             RubricVisOrg      0.48      0.17  0.50  0.45  0.28 -0.16  0.54
##  Artifact.1 as.factor(Rater)1 0.11
##             as.factor(Rater)2 0.33     -0.49
##             as.factor(Rater)3 0.31      0.33  0.66
##  Residual                     0.37
## ---
## number of obs: 810, groups: Artifact, 90
## AIC = 1484.6, DIC = 1233.2
## deviance = 1301.9
```

Then, we calculate the ICC's from model comb.final.

```
## intraclass correlation (ICC) for rubric RsrchQ
0.17898/(0.17898 +0.13469)
```

```
## [1] 0.5705997
```

```
## intraclass correlation (ICC) for rubric CritDes
0.49628/(0.49628+0.13469)
```

```
## [1] 0.786535
```

```
## intraclass correlation (ICC) for rubric InitEDA
0.31787/(0.31787+0.13469)
```

```
## [1] 0.702382
```

```
## intraclass correlation (ICC) for rubric SelMeth
0.03823/(0.03823+0.13469)
```

```
## [1] 0.2210849
```

```
## intraclass correlation (ICC) for rubric InterpRes
0.10204/(0.10204+0.13469)
```

```
## [1] 0.4310396
```

```
## intraclass correlation (ICC) for rubric VisOrg
0.23237/(0.23237+0.13469)
```

```
## [1] 0.6330573
```

```
## intraclass correlation (ICC) for rubric TxtOrg
0.25027/(0.25027+0.13469)
```

```
## [1] 0.6501195
```

From Appendix 3, Part C, we calculate the seven ICC's on the full data set.

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

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

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

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

round(agreement.results,2)
```

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

Next, we compare the ICC's from lcomb.final with our earlier ICC's from Appendix 3, Part C. Based on the comparison, I think the ICC's from model comb.final does not agree with my earlier ICC's. Especially intraclass correlation (ICC) for rubrics RsrchQ, SelMeth, InterpRes, CritDes and TxtOrg, which have a big changes (ICC changes for larger than 0.1).

- ICC for RsrchQ: increased from 0.21 to 0.57.
- ICC for SelMeth: decreased from 0.47 to 0.22.
- ICC for VisOrg: decreased from 0.66 to 0.63.
- ICC for TxtOrg: increased from 0.19 to 0.65.
- ICC for InterpRes: increased from 0.22 to 0.43.
- ICC for CritDes: increased from 0.67 to 0.79.
- ICC for InitEDA: decreased from 0.69 to 0.70.

# Appendix 5. Research Question #4

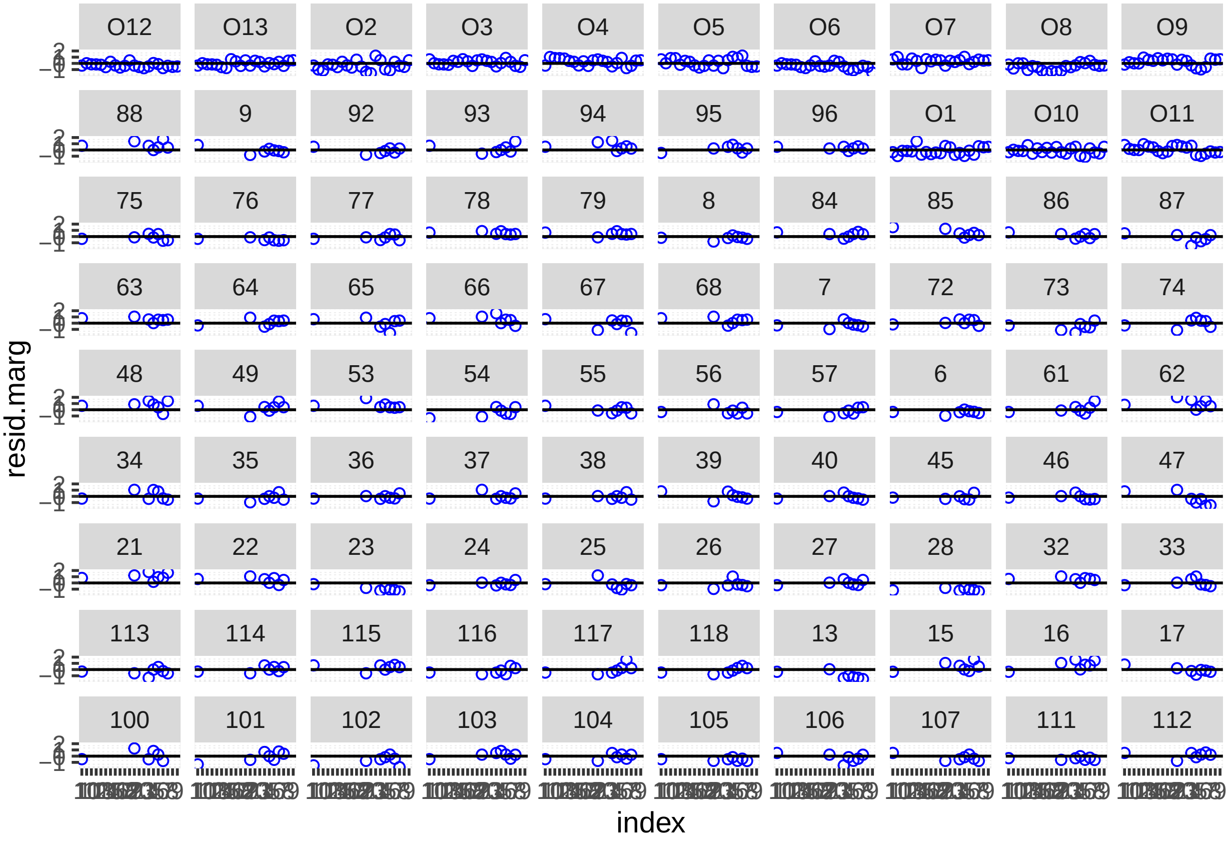
## Is there anything else interesting to say about this data?

### Residual Diagnostic Plots

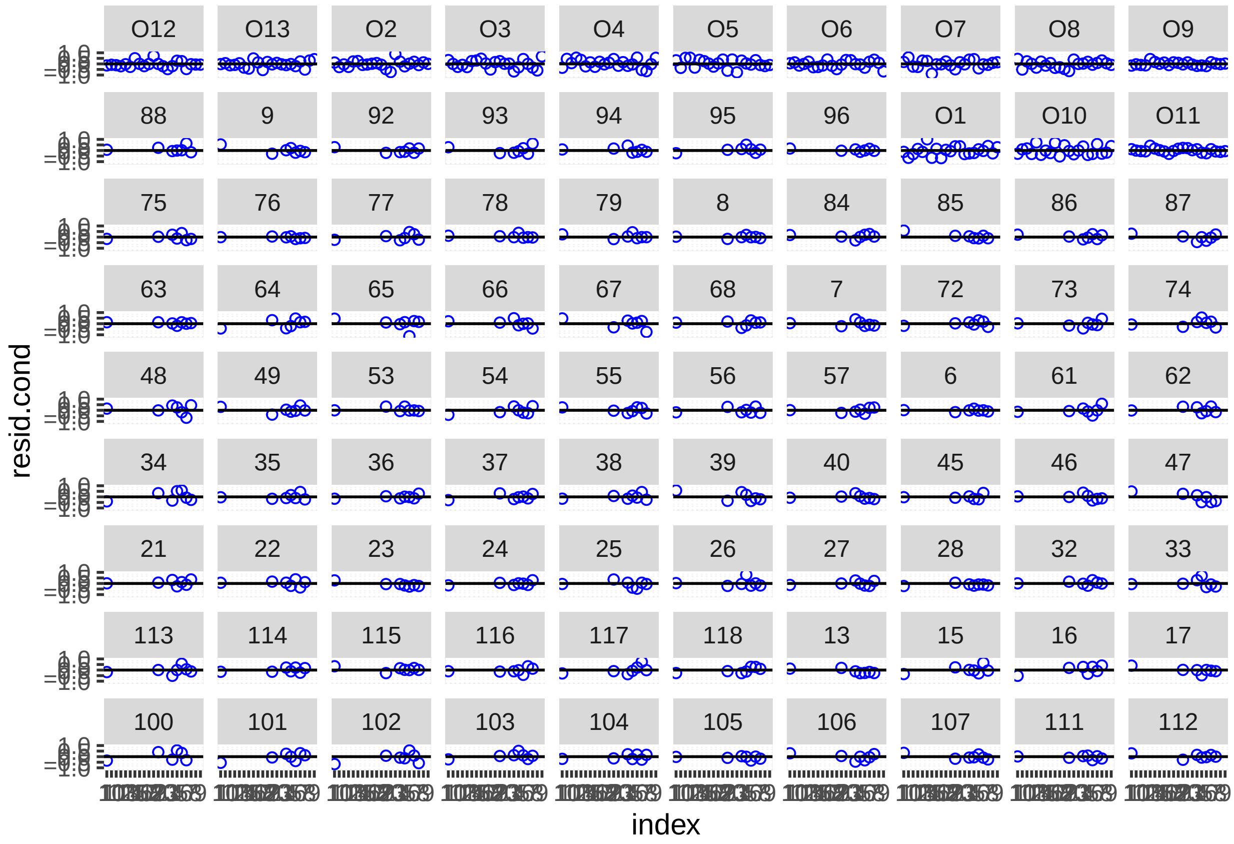
We do model diagnostics by plotting four types of residuals for comb.final to find something interesting.

```
source("residual-functions.r")
resid.chol <- r.chol(comb.final)
resid.marg <- r.marg(comb.final)
resid.cond <- r.cond(comb.final)
resid.reff <- r.reff(comb.final)
```

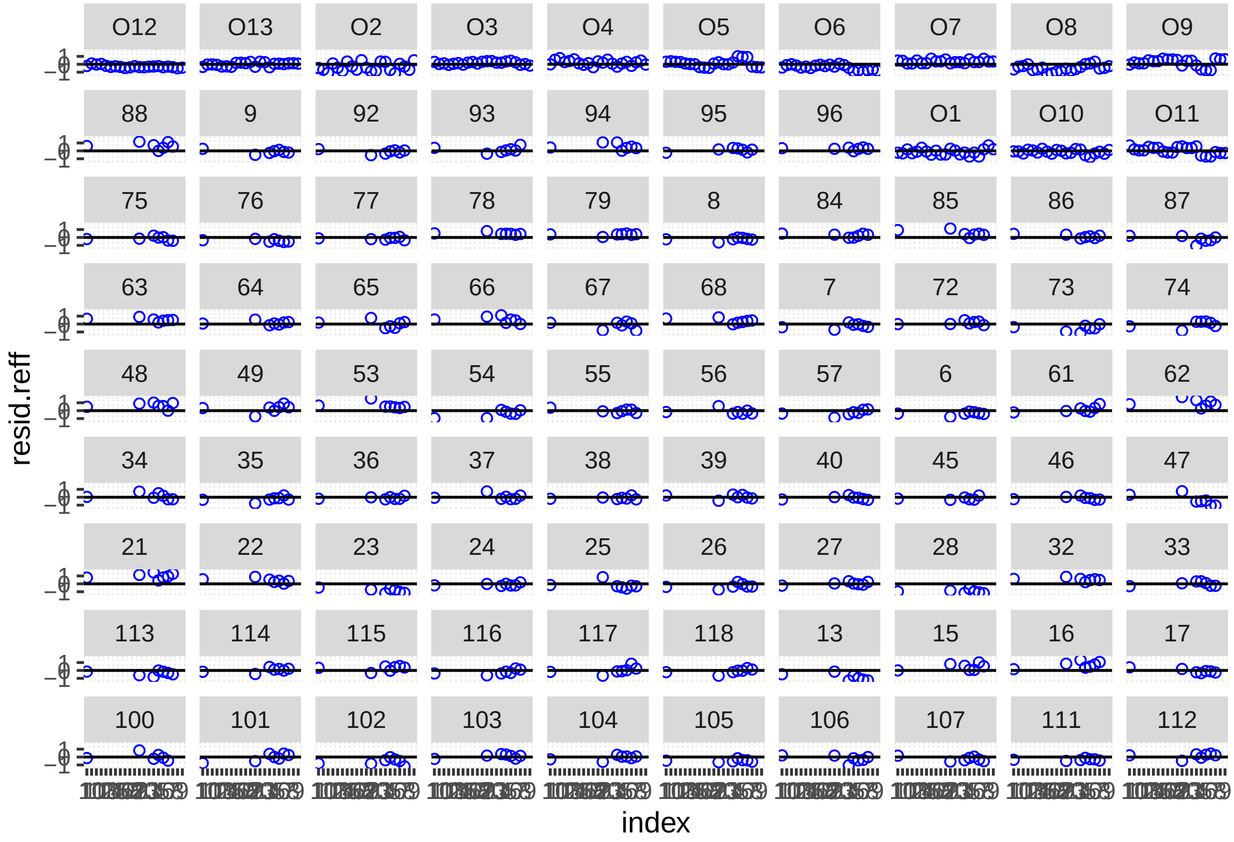
```
art <- tall.nonmissing$Artifact
index <- art
for (j in unique(art)) {
  len <- sum(art==j)
  index[art==j] <- 1:len
}
## Marginal residuals
new.data <- data.frame(index,resid.marg,tall.nonmissing$Artifact)
names(new.data) <- c("index","resid.marg","Artifact")
ggplot(new.data,aes(x=index,y=resid.marg)) +
  facet_wrap( ~ Artifact, as.table=F) +
  geom_point(pch=1,color="Blue") +
  geom_hline(yintercept=0)
```



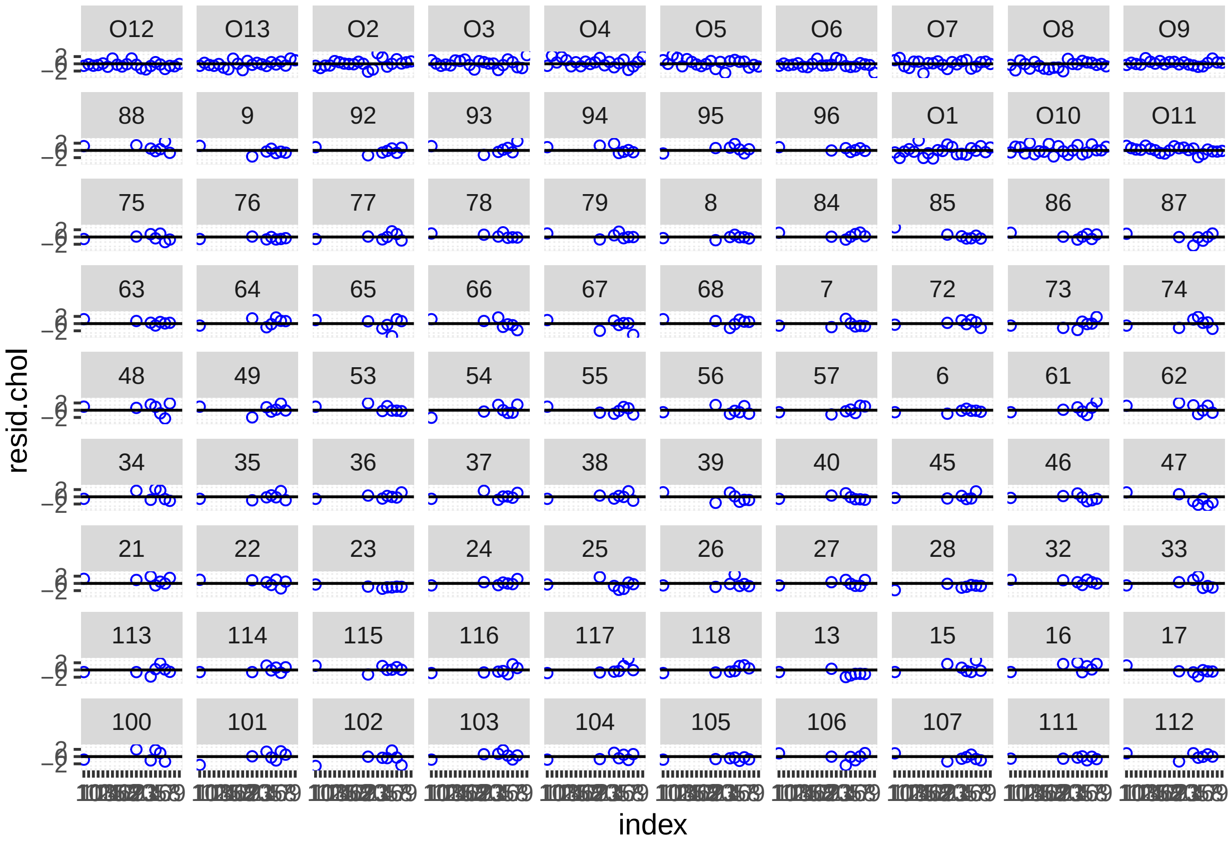
```
## Conditional residuals
new.data <- data.frame(index,resid.cond,tall.nonmissing$Artifact)
names(new.data) <- c("index","resid.cond","Artifact")
ggplot(new.data,aes(x=index,y=resid.cond)) +
  facet_wrap( ~ Artifact, as.table=F) +
  geom_point(pch=1,color="Blue") +
  geom_hline(yintercept=0)
```



```r
## Random effect residuals
new.data <- data.frame(index,resid.reff,tall.nonmissing$Artifact)
names(new.data) <- c("index","resid.reff","Artifact")
ggplot(new.data,aes(x=index,y=resid.reff)) +
  facet_wrap( ~ Artifact, as.table=F) +
  geom_point(pch=1,color="Blue") +
  geom_hline(yintercept=0)
```

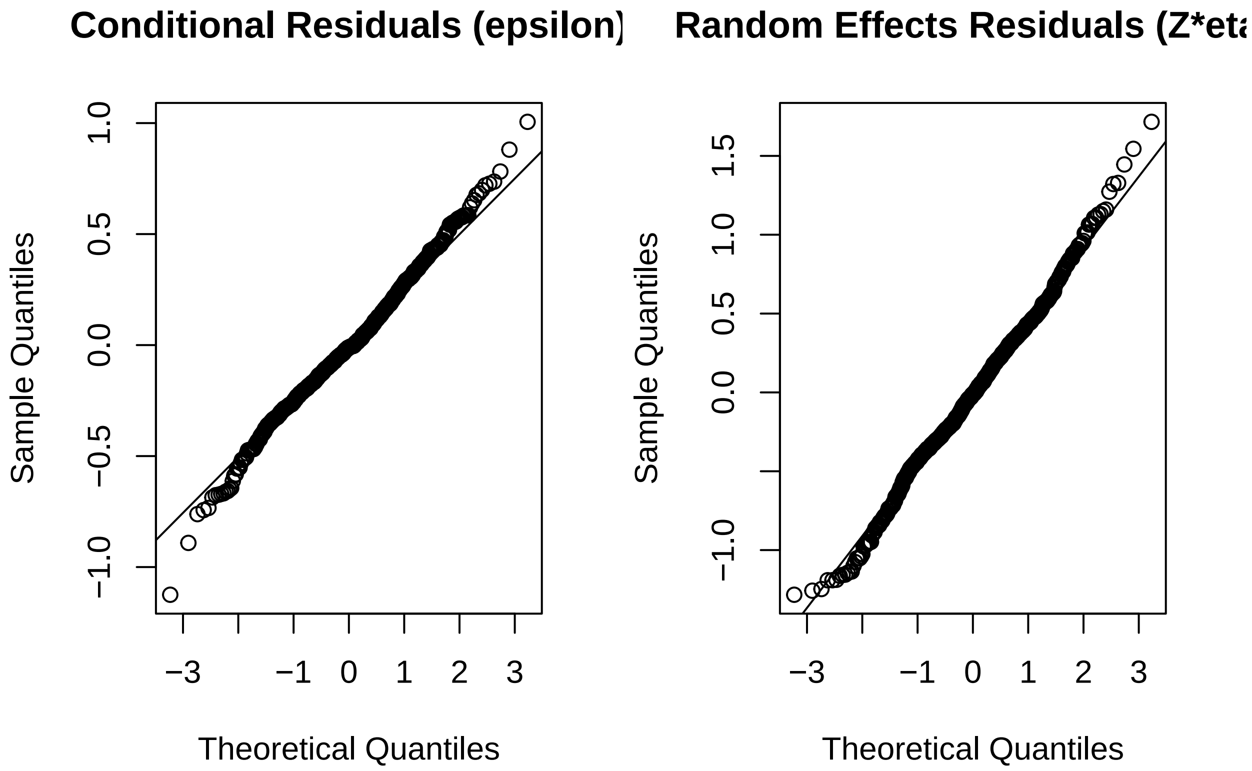


```
## Cholesky residuals
new.data <- data.frame(index,resid.chol,tall.nonmissing$Artifact)
names(new.data) <- c("index","resid.chol","Artifact")
ggplot(new.data,aes(x=index,y=resid.chol)) +
  facet_wrap( ~ Artifact, as.table=F) +
  geom_point(pch=1,color="Blue") +
  geom_hline(yintercept=0)
```



```
par(mfrow=c(1,2))
qqnorm(resid.cond,main="Conditional Residuals (epsilon)")
qqline(resid.cond)

qqnorm(resid.reff,main="Random Effects Residuals (Z*eta)")
qqline(resid.reff)
```



- Marginal residuals:

1) From the residual plots of marginal residuals, we find that there are no noticeable vertical patterns for most artifacts and the vertical lines in most plots are centered around 0, which indicates that marginal residuals for comb.final are mean 0. Although, there are some plots (78, 79, 8 etc.) may show grouping structures, which indicates there are some correlations in the marginal residuals, but they may not be homoskedastic. But overall nice set of the marginal residuals. Fixed effects in comb.final are good for predicting ratings.

- Conditional residuals:

1) From the residual plots of conditional residuals, we find that there are no noticeable vertical patterns for most artifacts and the vertical lines in most plots are centered around 0, which indicates that conditional residuals for comb.final are mean zero with no grouping structures. Conditional residuals for comb.final are homoskedastic. We also find that conditional residuals are not much spread like marginal residuals for comb.final.
2) Also from the Q-Q plot of conditional residuals, we find that most points are clustered around the straight line, except for some deviations on the top right and bottom left. Obviously there are two outliers on the top right and three outliers on the bottom left should be further investigated. But overall, we can see conditional residuals for comb.final are normally distributed and they are good to indicate normality of $\epsilon$. We can conclude nice set of the conditional residuals.

- Random effect residuals

1) From the residual plots of random effect residuals, we find that the scale of these residuals are smaller. There are no noticeable vertical patterns for most artifacts and the vertical lines. A few artifacts show noticeable deviation from 0, we do not expect mean-zero, but the BLUP estimates should cluster around a mean, which indicates that random effect residuals for comb.final are generally not be mean-zero and they may not be homoskedastic. But we do find there are some variance patterns in artifacts 68, 72, and 7 with all the points above/below the vertical lines, which should be further investigated.

2) Also from the Q-Q plot of random effect residuals, we find that most points are clustered around the straight line, except for some deviations on the top right. But overall random effect residuals are good. In indicates that random effect residuals for comb.final are normally distributed and they are good to indicate normality of $\eta$. We can conclude nice set of the random effect residuals.

- Cholesky residuals

1) Cholesky residuals are marginal residuals, transformed to remove the correlation. Hence we find that in the residual plots of cholesky residuals, the distributions of residuals in all artifacts are little more random compared with the marginal residuals. However, we find that there is very little difference between the residual plots of these two residuals since the correlations in the marginal residuals are small for comb.final.

Overall, we get $comb.final : as.numeric(Rating) = (0{+}Rubric|Artifact){+}(0{+}as.factor(Rater)|Artifact){+} as.factor(Rater) + Semester + Rubric + as.factor(Rater) : Rubric$ is a good model for predicting ratings in this case based on well-behaved residuals.