How students' performance on different rubrics affects students' work in Freshman Statistics and whether the new program is successful

Author: *Hongsheng Xie*
Department of Statistics and Data Science, Carnegie Mellon University
hongshex@andrew.cmu.edu

Abstract

In this paper, we will use two datasets containing ratings of students' works in Freshman Statistics to explore the correlation between each pair of variables, factors that affect students' performance significantly, and how are the various factors in this experiment (Rater, Semester, Sex, Repeated, Rubric) related to the ratings. Data from the process of a new "General Education" program in Dietrich College at Carnegie Mellon University is used and it contains freshmen's personal information and information about the rating of their works. By doing exploratory data analysis, multiple linear regression, linear mixed-effects model, cross-classifying tables, and interaction variables analysis, several factors that affect students' performance significantly are summed up. We find that CirtDes(Given an empirical research question, the student critiques or evaluates to what extent a study design convincingly answer that question) is a special rubrics in the performance testing, and factors interaction terms between the variables Rater, Semester, Rubric, and Repeated have some positive effects on the mixed-effects model, and the reason why factor interactions work is explained. T

he limitations of the models and data, for example one of the seven rubrics has quite distinct distribution of rating, are discussed as well.

Introduction

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 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. With the dataset containing variables including rater, ratings, rubric, sex, etc., a statistical analysis can be developed to generate a professional conclusion that for each rubric, if the raters generally agree on their scores or not, and besides rubric, what factors play a great role for students' performance. Using dataset containing data of artifact (# of project papers), sex, repeated (whether rated by all three raters or not), etc. from the experiment of Freshmen Statistics, this paper attempts to provide a mixed-effects model analysis to study how rubrics works with raters and the relationship between variables except rubrics, and build a variable set which could develop the most appropriate prediction model.

We will answer these following questions:

1. Is the distribution of ratings for each rubric pretty much indistinguishable from the other rubrics, or are there rubrics that tend to get especially high or low ratings? Is the distribution of ratings given by each rater pretty much indistinguishable from the other raters, or are there raters that tend to give especially high or low ratings?
2. For each rubric, do the raters generally agree on their scores? If not, is there one rater who disagrees with the others? Or do they all disagree?

3. More generally, how are the various factors in this experiment (Rater, Semester, Sex, Repeated, Rubric) related to the ratings? Do the factors interact in any interesting ways?
4. Is there anything else interesting to say about this data?

Data

The data used to study how different rubrics work for different raters and the correlation between other variables taken from the experiment on Freshmen Statistics on Dietrich College. In a recent experiment, 91 project papers—referred to as "artifacts"—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, as shown in Table1. The rating scale for all rubrics is shown in Table 2.

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

Table 1: Rubrics for rating Freshman Statistics projects. *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.*

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

Table 2: Rating scale used for all rubrics. *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.*

We examine the numeric variables and develop a description table for min, max, mean, etc. of the numeric variables (Table 3). After doing exploratory data analysis, the distribution bar plots provide information on the distributions for the quantitative variables. We find out that most of the distributions of ratings for each rubric are similar: a small portion of rating 1, most of the works are rated 2 or 3. However, in rubric CritDes, most of the works are rated 1. This scenario is shown in Figure 1.

	Min.	1sr Qu.	Median	Mean	3rd Qu.	Max.
ratings.X	1	30	59	59	88	117
ratings.Sample	1	31	60	59.89	89	118
ratings.Overlap	1	4	7	7	10	13
tall.X	1	205.5	410	410	614.5	819

Table 3: Summary table for numeric variables in *ratings* and *tall* dataset.

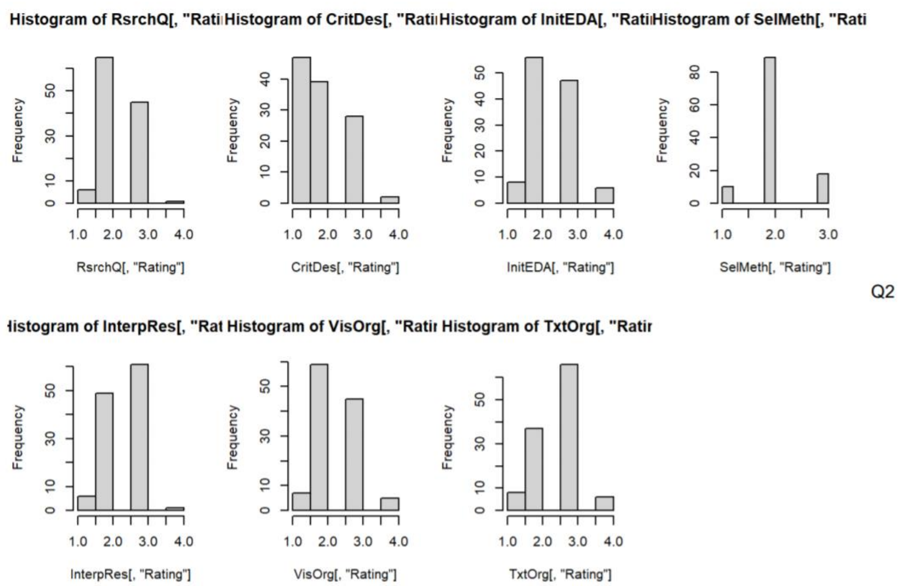


Figure 1: Initial EDA for *Ratings* dataset.

Methods

Question 1: Identify if a certain rater or rubric is quite distinguished from other raters or rubrics, or are there rubrics or raters tend to have especially high or low ratings?

we firstly checked the missing values of two datasets, and factorized Rating in *tall* dataset. Then we do summary tables on different rubrics to find out the distribution of different rubrics in numeric form. Finally, we make bar plots on each rubric to show the distributions visually, compare the difference, and decide whether the distribution of ratings for each rubric and each rater is distinguish with each other.

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?

We used cross-classifying model to check the intraclass correlation to quantify the degree of association between ratings within each rubric group and decide if the raters generally agree on their scores or not. We compare the percentage of rater's agreement to each other in pairs (rater 1 with rater 2, rater 1 with rater 3, and rater 2 with rater 3), and create tables of counts cross classifying the rating that each pair of raters gives.

Then we computed the ICC values using the full dataset to check whether the 7 ICC's for the full dataset agreed with the 7 ICC's for the subset corresponding to the 13 artifacts. This comparison showed the evidence that the subset we chose is reasonable and the method adopted on the subset could also be used on the full dataset.

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

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

We looked at the fixed effects on each of the 7 rubrics using the reduced dataset composed of the 13 common artifacts examined by all the raters. We started from big models considering all fixed-effect variables Rater, Semester, Sex, and Repeated and then doing backward elimination from the variables. Finally, we use ANOVA method to compare the performance of Chisq to choose the best performed variable combination for each rubric.

Adding fixed, random effects and interaction terms on the 7 rubric-specific models using the full dataset

We then fitted models using the full dataset. In the previous EDA part, we find that there are two students who has "NA" value in Rating column, and several students' gender are not shown. Because randomly imputing students' gender is quite unreasonable, and the model will be harder to interpret with missing gender value, we decided to eliminate the two rows with NA value in Rating and the rows lack of sex information.

Like the methods we used for the 13-artifact subset data, complete models with all the variables for all 7 rubrics were firstly tested. Then, we did backward elimination for each model to select the optimal subset of fixed effect. We mainly used t-values and ANOVA (Analysis of Variance) to check the performance of models with different variables combination, while using AIC, BIC, and Chisq values to compare various random effects.

Trying to add fixed effects, interactions, and new random effects to the "combined" model using all the data

Unlike the two previous steps in this question which tested models for each rubric, we used a single "combined" model that could similarly explain the 7 rubrics using random effects and interaction. We still firstly build a complete model with all fix-effect variables and did backward elimination. Then we would investigate interactions. Because the number of combinations of interactions is very large (several variables are categorical variables), part of the interaction combinations was chosen. Then AIC, BIC, and likelihood ratio tests were used to compare models with interactions terms since the models were nested. Finally, summary for the final selected model was shown to interpret the coefficients in the model.

Question 4: Some more EDA plots were made to see if there was anything else that could be deemed interesting for this study. Also, variable *gender* was studied specifically to look for a way to impute missing values as M or F.

Results

Question 1:

We firstly look at the distribution of ratings for individual rubrics and the distribution of ratings for each rater using data of the 13 common artifacts. Figure 2 shows the bar plots which present the rating distribution for each rubrics using data of the 13 artifacts, and Figure 3 shows the single bar plot which shows the rating distribution for each rater.

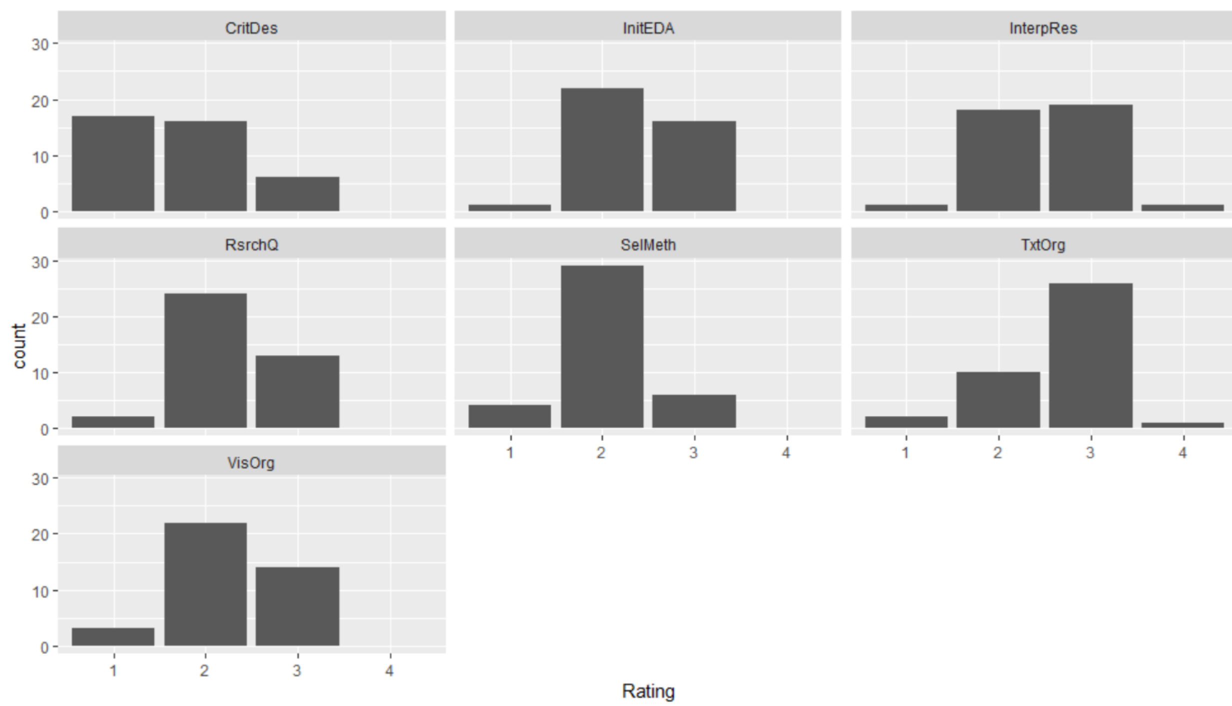


Figure 2: Rating distribution for each rubric with the 13 common artifacts

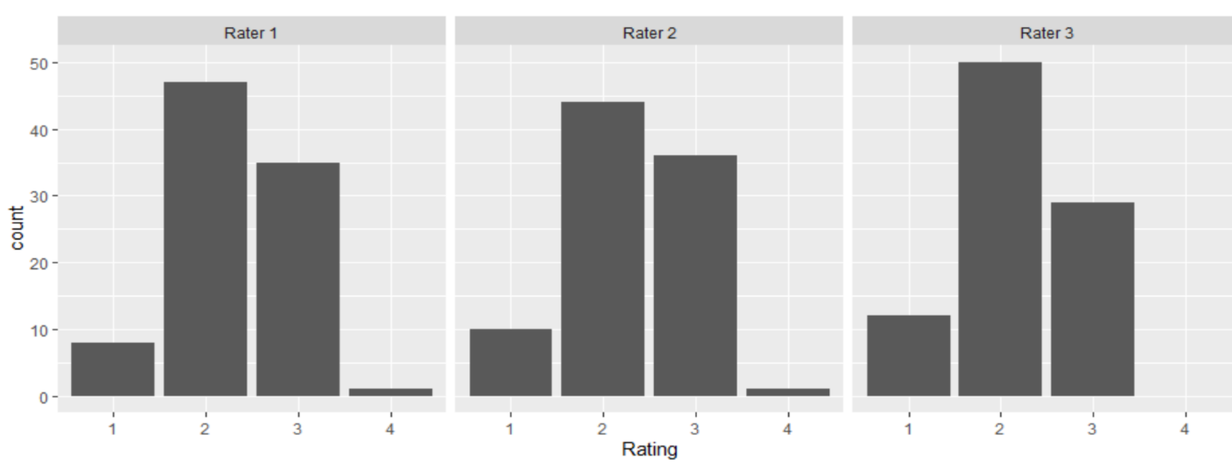


Figure 3: Rating distribution for each rater with the 13 common artifacts

From the bar plots for seven rubrics in the dataset with 13 artifacts, we found that the distributions of InitEDA, InterpRes, RsrchQ, and VisOrg are quite similar to each other: high percentage in score 2 and 3, very low percentage in score 1 and 4. The other three rubrics have their own pattern. CritDes has high percentage of score 1 and 2, SelMeth only has high percentage of score 2, and TxtOrg has very high percentage of score 3 and relatively less percentage of score 2. From the bar plots for three raters, we found that all three raters preferred to giving score 2 and then score 3.

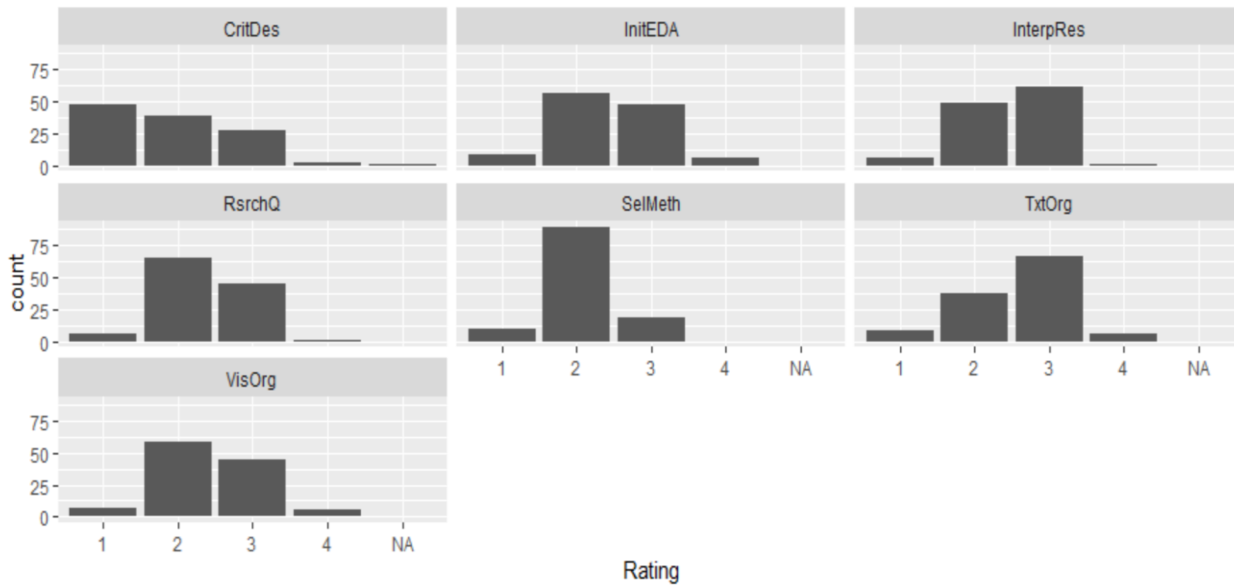


Figure 4: Rating distribution for each rubric with the full dataset

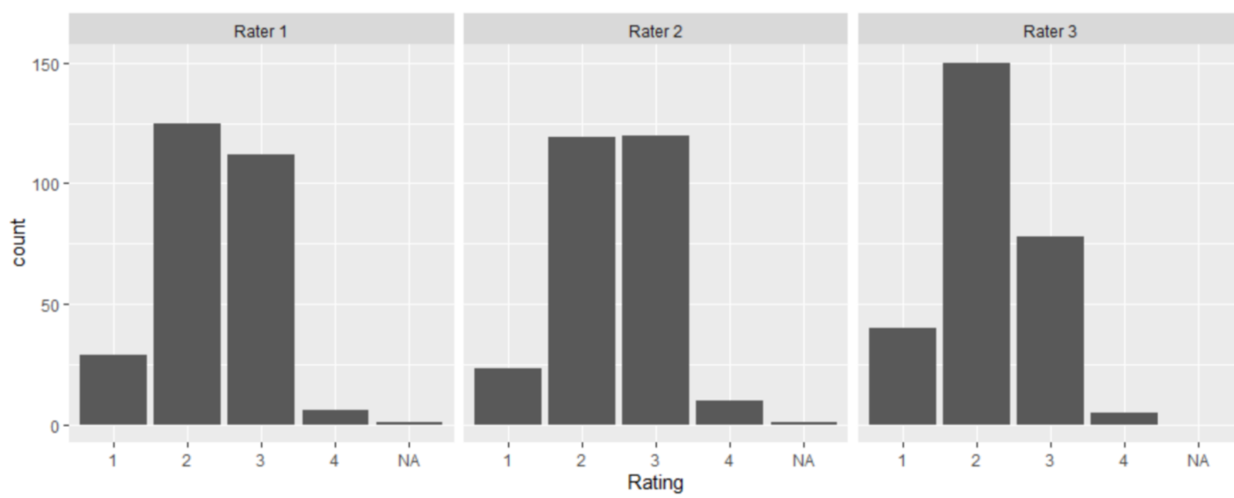


Figure 5: Rating distribution for each rater with the full dataset

From the bar plots for seven rubrics in the full dataset, we found that the pattern for each rubric is quite similar with the patterns appeared in the partial 13 common artifacts dataset. However, the distributions for each rater with the full dataset did not present the same pattern as those in the partial dataset. It showed that for Rater 1 and Rater 2, they did not have an apparent tendency to grade the tests score 2 or 3. But for Rater 3, he/she graded most of students' works as score 2.

Question 2:

Table 4 below shows the ICCs for each of the 7 rubrics in the full dataset and the partial artifacts dataset, and the percent exact agreement between each 2 raters for each rubric. The table shows that CritDes, InitEDA, SelMeth, and VisOrg have relatively large ICC for both the full and partial datasets, which means that in these rubrics, each pair of raters have a relatively high percentage of same rating for a specific artifact. On the other hand, for other three rubrics, raters might have some different consideration and gave different scores. Considering the percentage of same grading for each pair of raters for each rubric, most of the percentage of agreeing each other's score is larger than 50%, which means that most of time raters gave the same score.

	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	0.62	0.62
TxtOrg	0.19	0.14	0.69	0.54	0.62
VisOrg	0.66	0.59	0.46	0.69	0.69

Table 4: Rate of agreement for full data, 13 common artifacts, and between each two raters

Question 3:

For the first step in question 3, which just looked at the fixed effects to the seven rubric-specific models depending on the 13 common artifacts, we only built models with Rater, Semester, and Sex. We did not consider Repeated because in the 13 common artifacts subset, all raters graded, and no repeat scenario occurred. From the backward elimination method selection, we found that the final best performed models did not contain any fixed effect variables, and there was no need to check for any interaction terms between any two of them.

For the second step, we used the similar method: build a model containing all fixed effect variables and then do backward elimination by comparing AVONA. By using full dataset, we found that best performed models for rubrics InitEDA, RsrchQ, and TxtOrg did not include any fixed effect variables. Rater was retained in models for rubrics CritDes and InterpRes, and Rater and Semester were retained in models for rubric SelMeth.

For the third step, we wanted to check if interactions could improve the performance of a combined model. We did a full model with all interactions combined with the fixed effect variables, a backward eliminated model with some of the interactions, and a backward eliminated model with some of the fixed effect variables. Because the models are nested, we used ANOVA to look at all of AIC, BIC, log-likelihood, and deviance. It seemed that the best performed model was different under different standard. This suggested that the raters did not all use the rubrics in the same way.

Question 4:

From Table 5, we concluded that the mean scores for different genders cross the seven rubrics are quite similar. Average rating might not be an significant variable to impute missing Sex values.

Sex	total_score	mean_score	count
–	21	21.00000	1
F	1004	16.19355	62
M	841	16.17308	52

Table 5: Total and average ratings by gender

Then we looked at the total score and mean score for different genders by different rubrics. The summary tables showed that females performed 0.115 better than males in RsrchQ (Table 6), males performed 0.203 better than females in CritDes, males performed 0.042 better than females in InitEDA, males performed 0.1861 better than females in SelMeth, females performed 0.0738 better than males in InterpRes, females performed 0.193 better than males in VisOrg, and females performed 0.072 better than males in TxtOrg. For the apparently different performance in the seven rubrics for male and female, we believed that we could impute Sex appropriately depending on the performance on different rubrics.

Sex	total_score	mean_score	count
–	3	3.000000	1
F	149	2.403226	62
M	119	2.288461	52

Table 6: Total and average ratings for RsrchQ by gender

Sex	total_score	mean_score	count
–	3	3.000000	1
F	109	1.758064	62
M	102	1.961539	52

Table 7: Total and average ratings for CritDes by gender

Sex	total_score	mean_score	count
–	3	3.000000	1
F	150	2.419355	62
M	128	2.461539	52

Table 8: Total and average ratings for InitEDA by gender

Sex	total_score	mean_score	count
–	3	3.000000	1
F	122	1.967742	62
M	112	2.153846	52

Table 9: Total and average ratings for SelMeth by gender

Sex	total_score	mean_score	count
–	3	3.000000	1
F	156	2.516129	62
M	127	2.442308	52

Table 10: Total and average ratings for InterpRes by gender

Sex	total_score	mean_score	count
–	3	3.000000	1
F	155	2.500000	62
M	120	2.307692	52

Table 11: Total and average ratings for VisOrg by gender

Sex	total_score	mean_score	count
–	3	3.000000	1
F	163	2.629032	62
M	133	2.557692	52

Table 12: Total and average ratings for TxtOrg by gender

Discussion

From the exploratory data analysis, we classified the performance of artifacts of seven rubrics into five types: CritDes had about 40% score 1, 30% score 2, 20% score 3, and 10% others, InitEDA, RsrchQ, and VisOrg had about 50% score 2, 40% score 3, and 10% others, InterpRes had about 45% score 2, 45% score 3, and 10% others, SelfMeth had about 70% score 2, 20% score 3, and 10% score 1, and TxtOrg had about 60% score 3, 30% score 2, and 10% others.

When comparing the ICC value for full/partial datasets and the pairwise exact agreement, we found that the trend was not perfectly matched. This was because that only the observations on the main diagonal were considered for the exact agreement percentages, even though raters may have graded a rubric similarly (one rating off difference). If we consider the one rating off difference situation as agreement between pairwise raters, we believed that raters generally agreed on their scoring.

In the third question, we fitted fixed effect models for individual rubrics and a more general combined model. For the individual models using the 13 common artifacts, it turned out that none of the fixed effect variables and the interactions between these variables could improve the performance of the models. This is reasonable because we were in fact fitting individual models for each rubric for a very small dataset. However, when we enlarged the dataset for the model, the final best performed models for each rubric were no longer as simple as the previous ones. Some of the best performed models contained fixed effect variable Rater, and the model for SelMeth even contained fixed effect variable Semester.

When we analyzed the general model with backward elimination, the situation became more complicated. We fitted several models considering different factors. Model with only the fixed effect variables, and model with interactions between the variables were both considered. Because the models were all nested, we compared the performance of all the models depending on AIC, BIC, log likelihood, and deviance, and we found that no matter whether we considered the interaction or whether we did a backward elimination, each model performed well in some of the standard and performed relatively bad in some others. It was hard to say if interactions could improve the performance of the defined general model or not. More analysis was needed to make decision.

Additionally, before we fitted the models, we need to deal with the missing values. There were two missing values for Rating and several missing values for Sex in the full dataset. For missing values in Rating, we imputed them with the mode value across that specific Rubric. It was reasonable because the type of artifacts was categorical, and each rating was corresponded to the performance of an artifact for a certain rubric. However, imputing the Sex of the student who didn't report this to either M or F was a more difficult issue, because it was unreasonable to guess a student's gender. Thus, we decided to eliminate these observations from the dataset.

Because of the hard problem of imputing missing values in Sex column, we tried to figure out the pattern of Sex in the full dataset. When we looked at the whole dataset, we found that the difference of average ratings between male and female was only 0.020, which was too small to say females performed better than male. However, when we focused on the difference for each rubric, we found that males and females were good at difference rubrics. It was a big discovery, and we could impute Sex depending on the ratings on different rubrics for a specific case.

Appendix

[1] Dietrich College, Carnegie Mellon University. " Experiment on General Education program for undergraduates" . Project 2 instruction sheet

Project 2

Hongsheng Xie

11/18/2021

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
library(arm)
```

```
## Loading required package: MASS
```

```
##
## arm (Version 1.12-2, built: 2021-10-15)
```

```
## Working directory is C:/Users/danie/Desktop/CMU/36-617 applied regression analysis/Project 2
```

```
library(ggplot2)
```

```
ratings <- read.csv("C:/Users/danie/Desktop/CMU/36-617 applied regression analysis/Project 2/rat
ings.csv")
tall <- read.csv("C:/Users/danie/Desktop/CMU/36-617 applied regression analysis/Project 2/tall.c
sv")
```

```
summary(ratings$X)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       1      30      59      59      88     117
```

```
summary(ratings$Sample)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    1.00   31.00   60.00   59.89   89.00  118.00
```

```
summary(ratings$Overlap)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##       1       4       7       7      10      13      78
```

```
summary(tall$X)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0   205.5   410.0   410.0   614.5   819.0
```

```
tall$Rating <- factor(tall$Rating,levels=1:4)
for (i in unique(tall$Rubric)) {
ratings[,i] <- factor(ratings[,i],levels=1:4)
}
sum(is.na(tall))
```

```
## [1] 2
```

```
sum(is.na(ratings))
```

```
## [1] 80
```

```
tall$Sex[nchar(tall$Sex)==0] <- "--"

ratings.13 <- ratings[grep("O",ratings$Artifact),]
tall.13 <- tall[grep("O",tall$Artifact),]
```

```
sum(is.na(tall))
```

```
## [1] 2
```

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

```
## [1] 2
```

```
sum(is.na(ratings))
```

```
## [1] 80
```

```
tall[c(161,684),]
```

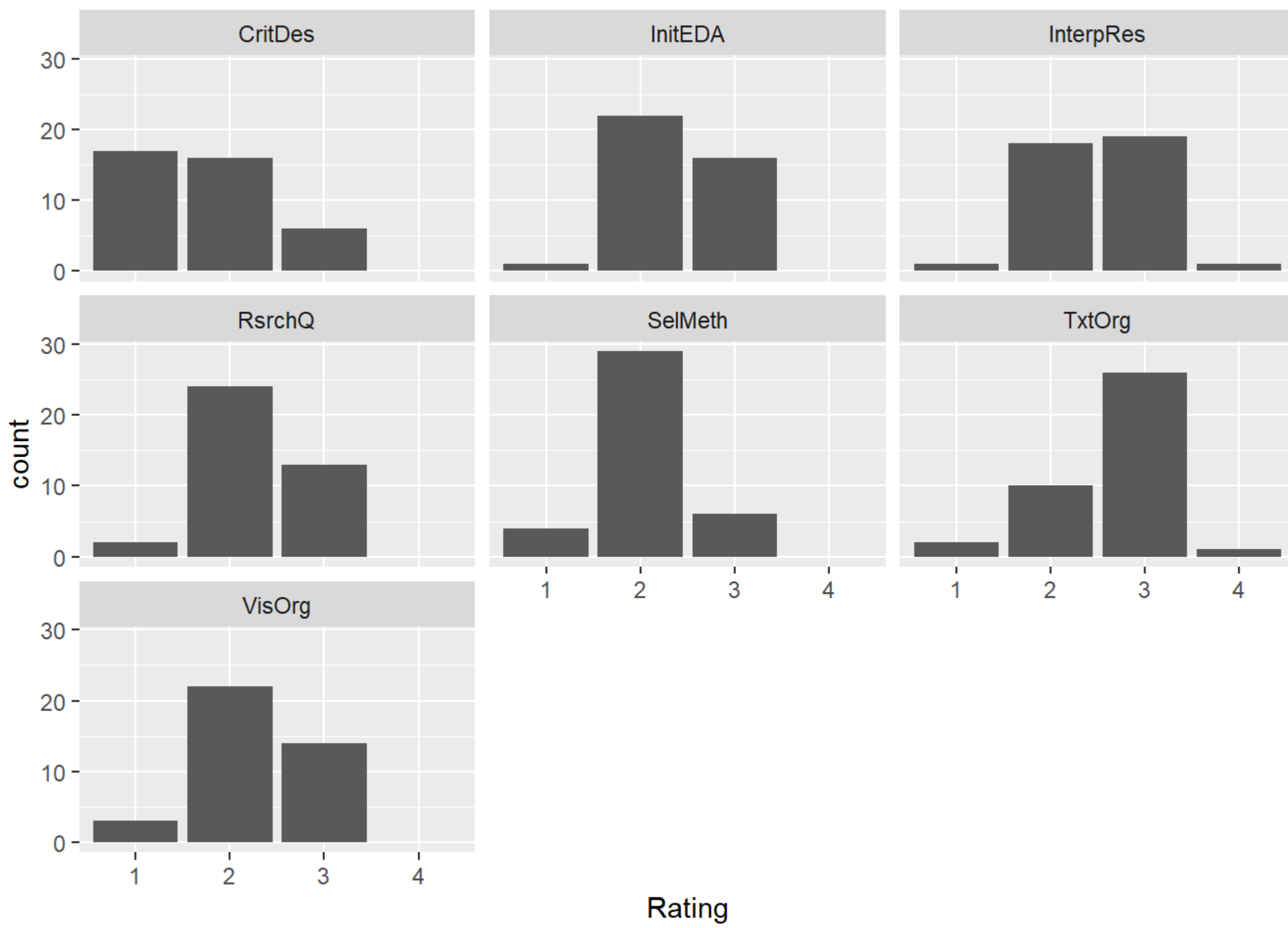
```
##       X Rater Artifact Repeated Semester Sex  Rubric Rating
## 161 161     2       45        0      S19   F CritDes   <NA>
## 684 684     1      100        0      F19   F  VisOrg   <NA>
```

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

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

Q1

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

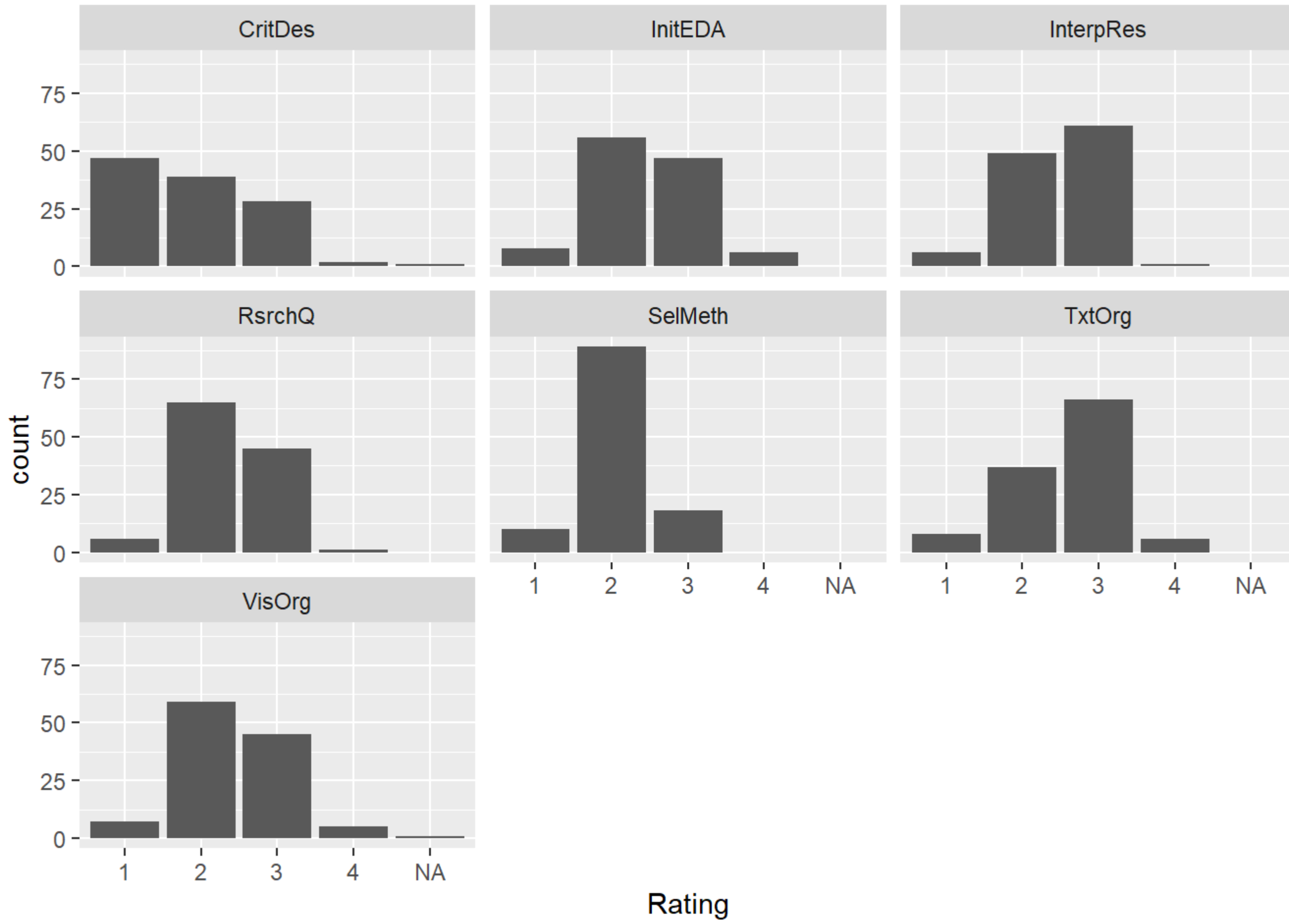



```
tmp <- data.frame(lapply(split(tall.13$Rating,tall.13$Rubric),summary))
row.names(tmp) <- paste("Rating",1:4)
tmp
```

```
##          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
```

#barplot for full dataset

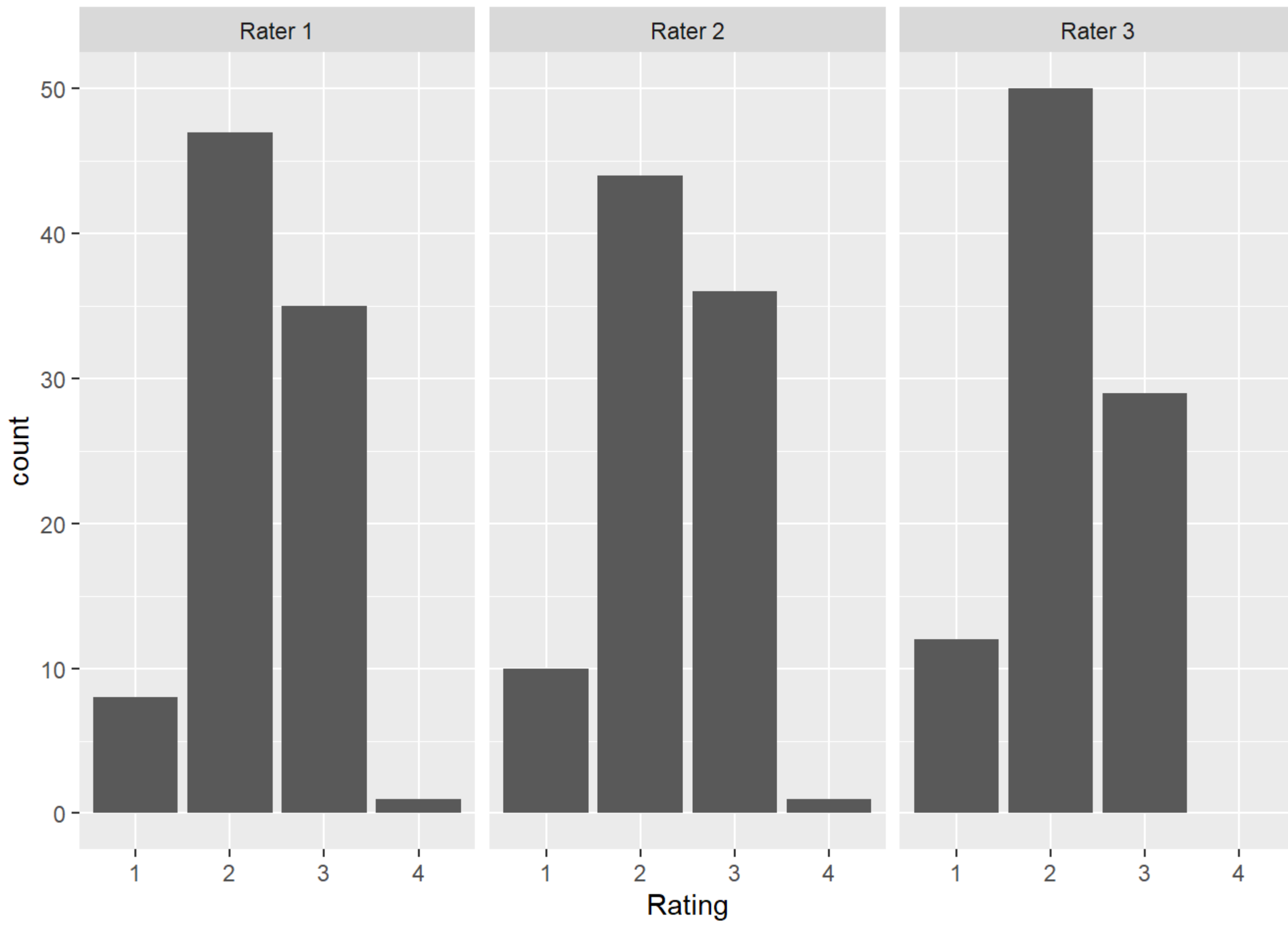
```
ggplot(tall,aes(x = Rating)) +
facet_wrap( ~ Rubric) +
geom_bar()
```




```
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
```

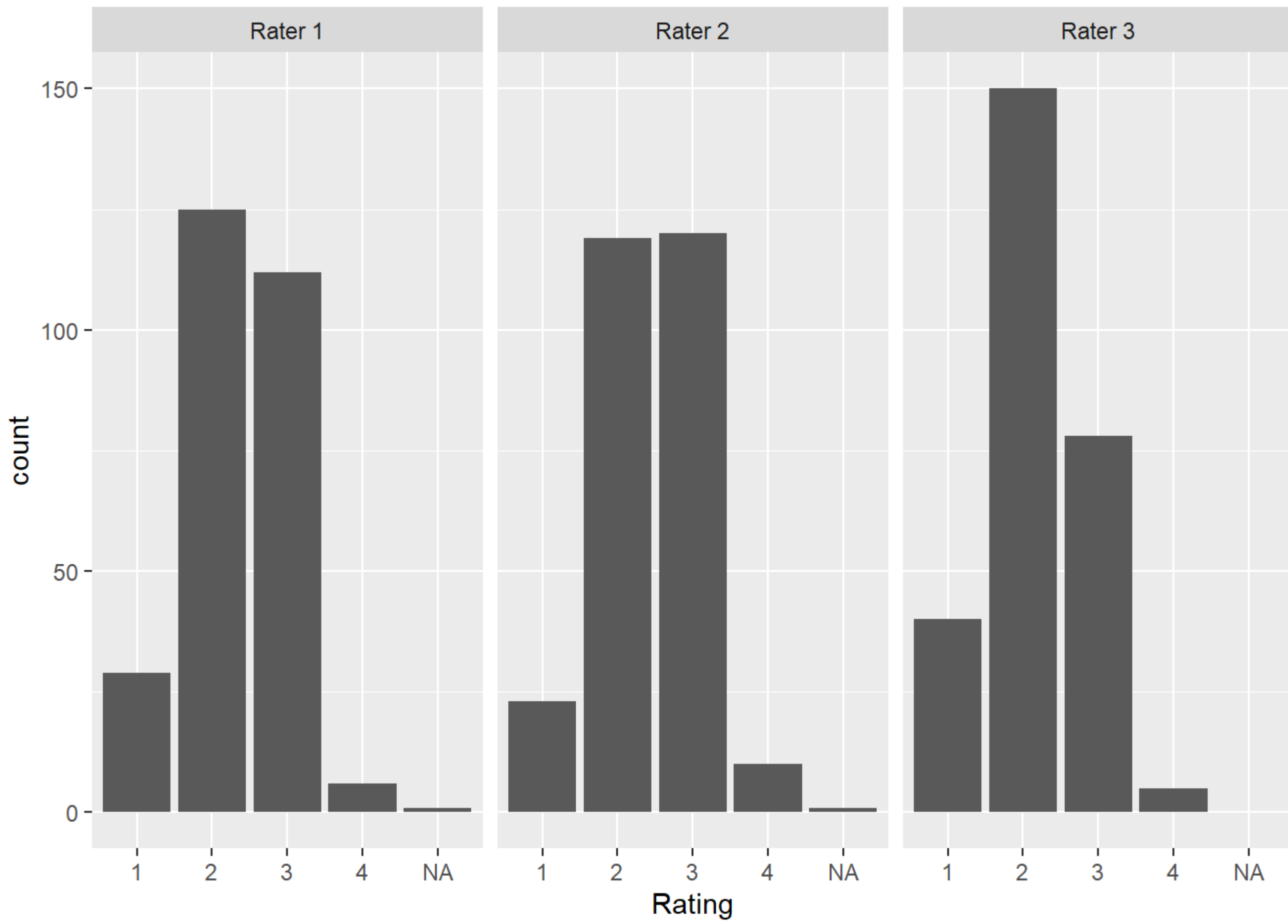
```
##
## Needed to make the title of each facet more human-readable...
rater.name <- function(x) { paste("Rater",x) }

##
## 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
```

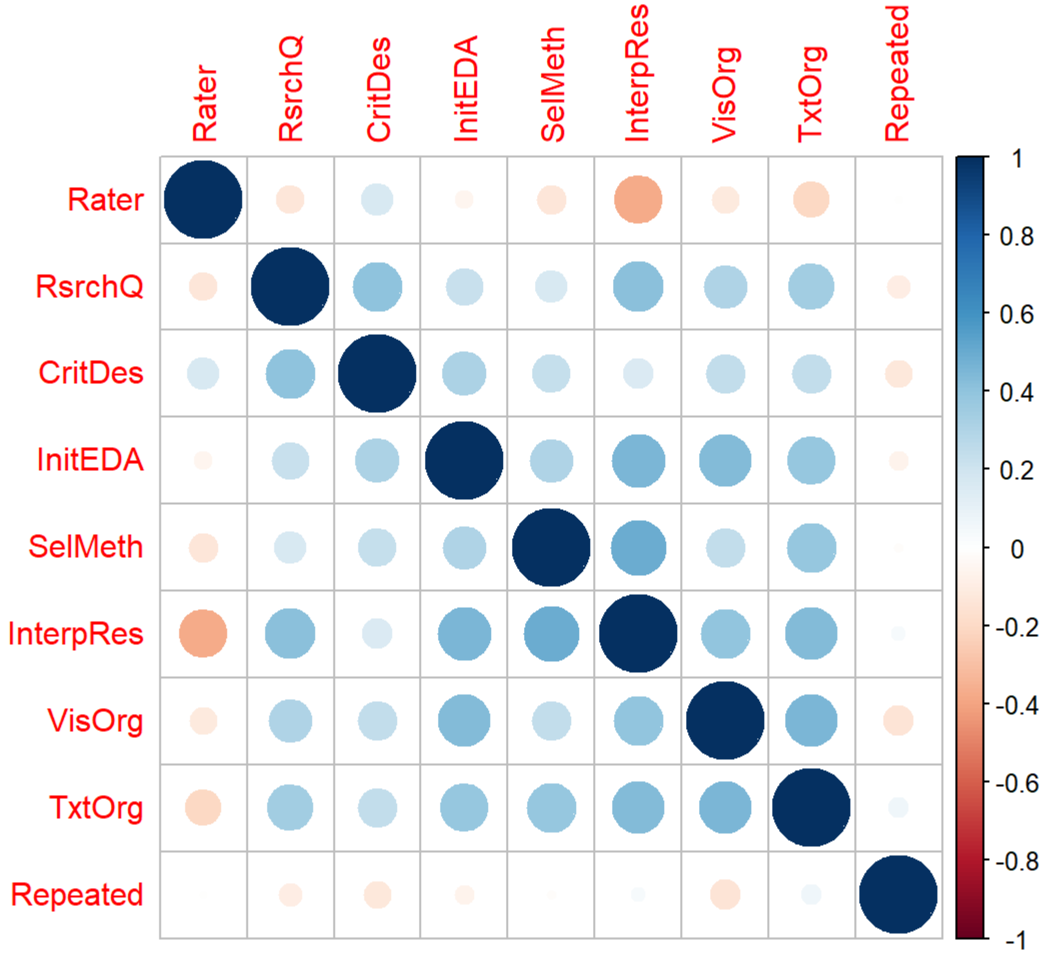
```
tall[apply(tall,1,function(x){any(is.na(x))}),]
```

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

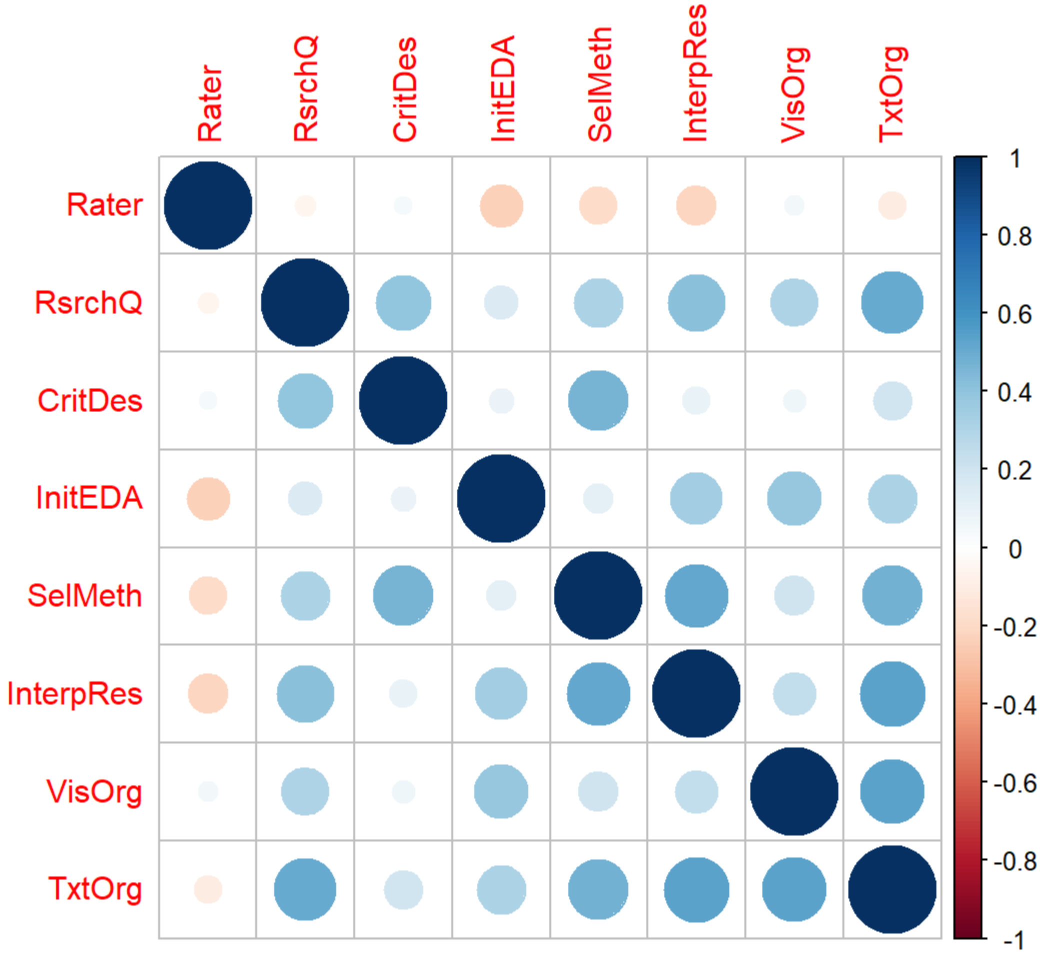
```
ratings[ratings$Sex=="--",]
```

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

```
ratings <- read.csv("C:/Users/danie/Desktop/CMU/36-617 applied regression analysis/Project 2/rat
ings.csv")

rate <- ratings[-c(44,99),-c(1,3,4)]
corrplot::corrplot(cor(rate[,-c(2,3,11)]))
```




```
data = rate[rate$Repeated == 1,]
corrplot::corrplot(cor(data[,-c(2,3,11,12)]))
```




```
# From the correlation tables, the correlation between each two factors is not quite strong. May
be considering interactions will help us.
```

```r
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
## The following objects are masked from 'package:arm':
##
##     logit, rescale, sim
```

```r
tall <- read.csv("C:/Users/danie/Desktop/CMU/36-617 applied regression analysis/Project 2/tall.c
sv")
describe(ratings)
```

```
##            vars   n  mean    sd median trimmed   mad min max range  skew
## X             1 117 59.00 33.92     59   59.00 43.00   1 117   116  0.00
## Rater         2 117  2.00  0.82      2    2.00  1.48   1   3     2  0.00
## Sample        3 117 59.89 34.09     60   59.98 43.00   1 118   117 -0.01
## Overlap       4  39  7.00  3.79      7    7.00  4.45   1  13    12  0.00
## Semester*     5 117  1.29  0.46      1    1.24  0.00   1   2     1  0.91
## Sex*          6 117  2.44  0.52      2    2.43  0.00   1   3     2  0.07
## RsrchQ        7 117  2.35  0.59      2    2.37  0.00   1   4     3 -0.03
## CritDes       8 116  1.87  0.84      2    1.82  1.48   1   4     3  0.42
## InitEDA       9 117  2.44  0.70      2    2.44  1.48   1   4     3  0.08
## SelMeth      10 117  2.07  0.49      2    2.07  0.00   1   3     2  0.17
## InterpRes    11 117  2.49  0.61      3    2.54  0.00   1   4     3 -0.52
## VisOrg       12 116  2.41  0.67      2    2.41  0.00   1   4     3  0.14
## TxtOrg       13 117  2.60  0.70      3    2.64  0.00   1   4     3 -0.50
## Artifact*    14 117 54.67 28.45     59   56.25 35.58   1  91    90 -0.35
## Repeated     15 117  0.33  0.47      0    0.29  0.00   0   1     1  0.70
##            kurtosis   se
## X             -1.23 3.14
## Rater         -1.53 0.08
## Sample        -1.22 3.15
## Overlap       -1.30 0.61
## Semester*     -1.18 0.04
## Sex*          -1.56 0.05
## RsrchQ        -0.47 0.05
## CritDes       -1.03 0.08
## InitEDA       -0.26 0.06
## SelMeth        1.06 0.04
## InterpRes     -0.44 0.06
## VisOrg        -0.20 0.06
## TxtOrg        -0.01 0.06
## Artifact*     -1.29 2.63
## Repeated      -1.53 0.04
```

Q2

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

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

names(ICC.vec) <- Rubric.names

agreement.results <- cbind(ICC.common=ICC.vec," a12"=0,a23=0,a13=0)
agreement.tables <- as.list(rep(NA,7))
names(agreement.tables) <- Rubric.names
##
## 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)

agreement.results
```

```
##           ICC.alldata ICC.common  a12 a23 a13
## CritDes     0.6730647  0.5725594    0   0   0
## InitEDA     0.6867210  0.4929577    0   0   0
## InterpRes   0.2200285  0.2295720    0   0   0
## RsrchQ      0.2096214  0.1891892    0   0   0
## SelMeth     0.4719014  0.5212766    0   0   0
## TxtOrg      0.1879927  0.1428571    0   0   0
## VisOrg      0.6607372  0.5924529    0   0   0
```

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

```
##           ICC.alldata ICC.common  a12 a23 a13
## CritDes          0.67       0.57    0   0   0
## InitEDA          0.69       0.49    0   0   0
## InterpRes        0.22       0.23    0   0   0
## RsrchQ           0.21       0.19    0   0   0
## SelMeth          0.47       0.52    0   0   0
## TxtOrg           0.19       0.14    0   0   0
## VisOrg           0.66       0.59    0   0   0
```

```
library(arm)
library(lme4)
#tall <- read.csv("C:/Users/danie/Desktop/CMU/36-617 applied regression analysis/Project 2/tall.
csv")
common <- tall[grep("0",tall$Artifact),]
```

```
RsrchQ.ratings <- common[common$Rubric=="RsrchQ",]
RsrchQ_Rater <- lmer(as.numeric(Rating) ~ 1 + (1|Rater), data=RsrchQ.ratings)
```

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

```
RsrchQ_Artifact <- lmer(as.numeric(Rating) ~ 1 + (1|Artifact), data=RsrchQ.ratings)
```

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

```
CritDes.ratings <- common[common$Rubric=="CritDes",]
CritDes_Rater <- lmer(as.numeric(Rating) ~ 1 + (1|Rater), data=CritDes.ratings)
```

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

```
CritDes_Artifact <- lmer(as.numeric(Rating) ~ 1 + (1|Artifact), data=CritDes.ratings)
```

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

```
InitEDA.ratings <- common[common$Rubric=="InitEDA",]
InitEDA_Rater <- lmer(as.numeric(Rating) ~ 1 + (1|Rater), data=InitEDA.ratings)
```

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

```
InitEDA_Artifact <- lmer(as.numeric(Rating) ~ 1 + (1|Artifact), data=InitEDA.ratings)
```

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

```
SelMeth.ratings <- common[common$Rubric=="SelMeth",]
SelMeth_Rater <- lmer(as.numeric(Rating) ~ 1 + (1|Rater), data=SelMeth.ratings)
```

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

```
SelMeth_Artifact <- lmer(as.numeric(Rating) ~ 1 + (1|Artifact), data=SelMeth.ratings)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearl
y unidentifiable: very large eigenvalue
##  - Rescale variables?
```

```
InterpRes.ratings <- common[common$Rubric=="InterpRes",]
InterpRes_Rater <- lmer(as.numeric(Rating) ~ 1 + (1|Rater), data=InterpRes.ratings)
```

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

```
InterpRes_Artifact <- lmer(as.numeric(Rating) ~ 1 + (1|Artifact), data=InterpRes.ratings)
```

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

```
VisOrg.ratings <- common[common$Rubric=="VisOrg",]
VisOrg_Rater <- lmer(as.numeric(Rating) ~ 1 + (1|Rater), data=VisOrg.ratings)
VisOrg_Artifact <- lmer(as.numeric(Rating) ~ 1 + (1|Artifact), data=VisOrg.ratings)
```

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

```
TxtOrg.ratings <- common[common$Rubric=="TxtOrg",]
TxtOrg_Rater <- lmer(as.numeric(Rating) ~ 1 + (1|Rater), data=TxtOrg.ratings)
```

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

```
TxtOrg_Artifact <- lmer(as.numeric(Rating) ~ 1 + (1|Artifact), data=TxtOrg.ratings)
```

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

```
#summary(RsrchQ_Rater)
#summary(CritDes_Rater)
#summary(InitEDA_Rater)
#summary(SelMeth_Rater)
#summary(InterpRes_Rater)
#summary(VisOrg_Rater)
#summary(TxtOrg_Rater)

#summary(RsrchQ_Artifact)
#summary(CritDes_Artifact)
#summary(InitEDA_Artifact)
#summary(SelMeth_Artifact)
#summary(InterpRes_Artifact)
#summary(VisOrg_Artifact)
#summary(TxtOrg_Artifact)
```

```
repeated <- ratings[ratings$Repeated==1,]
```

RsrchQ rubric

```
raters_1_and_2_on_RsrchQ <- data.frame(r1=repeated$RsrchQ[repeated$Rater==1],
                                       r2=repeated$RsrchQ[repeated$Rater==2],
                                       a1=repeated$Artifact[repeated$Rater==1],
                                       a2=repeated$Artifact[repeated$Rater==2]
)

r1 <- factor(raters_1_and_2_on_RsrchQ$r1,levels=1:4)
r2 <- factor(raters_1_and_2_on_RsrchQ$r2,levels=1:4)
(t12 <- table(r1,r2))
```

```
##    r2
## r1  1 2 3 4
##   1 0 0 0 0
##   2 1 4 3 0
##   3 1 3 1 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 2 for
the rubric RsrchQ have the same rate in 5/13 of the cases. From the rubric RsrchQ, rater 1 and 2
do not always disagree with each other.

agreement.results[4,3] = round(5/13,2)
```

```
raters_1_and_3_on_RsrchQ <- data.frame(r1=repeated$RsrchQ[repeated$Rater==1],
                                       r3=repeated$RsrchQ[repeated$Rater==3],
                                       a1=repeated$Artifact[repeated$Rater==1],
                                       a3=repeated$Artifact[repeated$Rater==3]
)

r1 <- factor(raters_1_and_3_on_RsrchQ$r1,levels=1:4)
r3 <- factor(raters_1_and_3_on_RsrchQ$r3,levels=1:4)
(t13 <- table(r1,r3))
```

```
##    r3
## r1  1 2 3 4
##   1 0 0 0 0
##   2 0 7 1 0
##   3 0 2 3 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 3 for
the rubric RsrchQ have the same rate in 10/13 of the cases. From the rubric RsrchQ, rater 1 and
 3 do not always disagree with each other.

agreement.results[4,5] = round(10/13,2)
```

```
raters_2_and_3_on_RsrchQ <- data.frame(r2=repeated$RsrchQ[repeated$Rater==2],
                                       r3=repeated$RsrchQ[repeated$Rater==3],
                                       a2=repeated$Artifact[repeated$Rater==2],
                                       a3=repeated$Artifact[repeated$Rater==3]
)

r2 <- factor(raters_2_and_3_on_RsrchQ$r2,levels=1:4)
r3 <- factor(raters_2_and_3_on_RsrchQ$r3,levels=1:4)
(t23 <- table(r2,r3))
```

```
##    r3
## r2  1 2 3 4
##   1 0 2 0 0
##   2 0 5 2 0
##   3 0 2 2 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 2 for
the rubric RsrchQ have the same rate in 7/13 of the cases. From the rubric RsrchQ, rater 2 and 3
do not always disagree with each other.
agreement.results[4,4] = round(7/13,2)
```

CritDes rubric

```
raters_1_and_2_on_CritDes <- data.frame(r1=repeated$CritDes[repeated$Rater==1],
                                        r2=repeated$CritDes[repeated$Rater==2],
                                        a1=repeated$Artifact[repeated$Rater==1],
                                        a2=repeated$Artifact[repeated$Rater==2]
)

r1 <- factor(raters_1_and_2_on_CritDes$r1,levels=1:4)
r2 <- factor(raters_1_and_2_on_CritDes$r2,levels=1:4)
(t12 <- table(r1,r2))
```

```
##    r2
## r1  1 2 3 4
##   1 3 2 1 0
##   2 2 3 1 0
##   3 0 0 1 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 2 for the rubric CritDes have the same rate in 7/13 of the cases. From the rubric CritDes, rater 1 and 2 do not always disagree with each other.
agreement.results[1,3] = round(7/13,2)
```

```
raters_1_and_3_on_CritDes <- data.frame(r1=repeated$CritDes[repeated$Rater==1],
                                        r3=repeated$CritDes[repeated$Rater==3],
                                        a1=repeated$Artifact[repeated$Rater==1],
                                        a3=repeated$Artifact[repeated$Rater==3]
)

r1 <- factor(raters_1_and_3_on_CritDes$r1,levels=1:4)
r3 <- factor(raters_1_and_3_on_CritDes$r3,levels=1:4)
(t13 <- table(r1,r3))
```

```
##    r3
## r1  1 2 3 4
##   1 4 2 0 0
##   2 2 3 1 0
##   3 0 0 1 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 3 for the rubric CritDes have the same rate in 8/13 of the cases. From the rubric CritDes, rater 1 and 3 do not always disagree with each other.
agreement.results[1,5] = round(8/13,2)
```

```
raters_2_and_3_on_CritDes <- data.frame(r2=repeated$CritDes[repeated$Rater==2],
                                        r3=repeated$CritDes[repeated$Rater==3],
                                        a2=repeated$Artifact[repeated$Rater==2],
                                        a3=repeated$Artifact[repeated$Rater==3]
)

r2 <- factor(raters_2_and_3_on_CritDes$r2,levels=1:4)
r3 <- factor(raters_2_and_3_on_CritDes$r3,levels=1:4)
(t23 <- table(r2,r3))
```

```
##    r3
## r2  1 2 3 4
##   1 5 0 0 0
##   2 1 3 1 0
##   3 0 2 1 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 3 and rater 2 for
the rubric CritDes have the same rate in 9/13 of the cases. From the rubric CritDes, rater 3 and
2 do not always disagree with each other.
agreement.results[1,4] = round(9/13,2)
```

InitEDA rubric

```
raters_1_and_2_on_InitEDA <- data.frame(r1=repeated$InitEDA[repeated$Rater==1],
                                        r2=repeated$InitEDA[repeated$Rater==2],
                                        a1=repeated$Artifact[repeated$Rater==1],
                                        a2=repeated$Artifact[repeated$Rater==2]
)

r1 <- factor(raters_1_and_2_on_InitEDA$r1,levels=1:4)
r2 <- factor(raters_1_and_2_on_InitEDA$r2,levels=1:4)
(t12 <- table(r1,r2))
```

```
##    r2
## r1  1 2 3 4
##   1 0 1 0 0
##   2 0 4 0 0
##   3 0 3 5 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 2 for
the rubric InitEDA have the same rate in 9/13 of the cases. From the rubric InitEDA, rater 1 and
2 do not always disagree with each other.
agreement.results[2,3] = round(9/13,2)
```

```
raters_1_and_3_on_InitEDA <- data.frame(r1=repeated$InitEDA[repeated$Rater==1],
                                       r3=repeated$InitEDA[repeated$Rater==3],
                                       a1=repeated$Artifact[repeated$Rater==1],
                                       a3=repeated$Artifact[repeated$Rater==3]
)

r1 <- factor(raters_1_and_3_on_InitEDA$r1,levels=1:4)
r3 <- factor(raters_1_and_3_on_InitEDA$r3,levels=1:4)
(t13 <- table(r1,r3))
```

```
##     r3
## r1  1 2 3 4
##   1 0 1 0 0
##   2 0 4 0 0
##   3 0 5 3 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 3 for
the rubric InitEDA have the same rate in 7/13 of the cases. From the rubric InitEDA, rater 1 and
3 do not always disagree with each other.
agreement.results[2,5] = round(7/13,2)
```

```
raters_2_and_3_on_InitEDA <- data.frame(r2=repeated$InitEDA[repeated$Rater==2],
                                       r3=repeated$InitEDA[repeated$Rater==3],
                                       a2=repeated$Artifact[repeated$Rater==2],
                                       a3=repeated$Artifact[repeated$Rater==3]
)

r2 <- factor(raters_2_and_3_on_InitEDA$r2,levels=1:4)
r3 <- factor(raters_2_and_3_on_InitEDA$r3,levels=1:4)
(t23 <- table(r2,r3))
```

```
##     r3
## r2  1 2 3 4
##   1 0 0 0 0
##   2 0 8 0 0
##   3 0 2 3 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 2 and rater 3 for
the rubric InitEDA have the same rate in 11/13 of the cases. From the rubric InitEDA, rater 2 an
d 3 do not always disagree with each other.
agreement.results[2,4] = round(11/13,2)
```

SelMeth rubric

```
raters_1_and_2_on_SelMeth <- data.frame(r1=repeated$SelMeth[repeated$Rater==1],
                                        r2=repeated$SelMeth[repeated$Rater==2],
                                        a1=repeated$Artifact[repeated$Rater==1],
                                        a2=repeated$Artifact[repeated$Rater==2]
)

r1 <- factor(raters_1_and_2_on_SelMeth$r1,levels=1:4)
r2 <- factor(raters_1_and_2_on_SelMeth$r2,levels=1:4)
(t12 <- table(r1,r2))
```

```
##    r2
## r1   1  2  3  4
##   1  0  0  0  0
##   2  1 10  0  0
##   3  0  0  2  0
##   4  0  0  0  0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 2 for
the rubric SelMeth have the same rate in 12/13 of the cases. From the rubric SelMeth, rater 1 an
d 2 do not always disagree with each other.
agreement.results[5,3] = round(12/13,2)
```

```
raters_1_and_3_on_SelMeth <- data.frame(r1=repeated$SelMeth[repeated$Rater==1],
                                        r3=repeated$SelMeth[repeated$Rater==3],
                                        a1=repeated$Artifact[repeated$Rater==1],
                                        a3=repeated$Artifact[repeated$Rater==3]
)

r1 <- factor(raters_1_and_3_on_SelMeth$r1,levels=1:4)
r3 <- factor(raters_1_and_3_on_SelMeth$r3,levels=1:4)
(t13 <- table(r1,r3))
```

```
##    r3
## r1  1 2 3 4
##   1 0 0 0 0
##   2 3 7 1 0
##   3 0 1 1 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 3 for
the rubric SelMeth have the same rate in 8/13 of the cases. From the rubric SelMeth, rater 1 and
3 do not always disagree with each other.
agreement.results[5,5] = round(8/13,2)
```

```
raters_2_and_3_on_SelMeth <- data.frame(r2=repeated$SelMeth[repeated$Rater==2],
                                       r3=repeated$SelMeth[repeated$Rater==3],
                                       a2=repeated$Artifact[repeated$Rater==2],
                                       a3=repeated$Artifact[repeated$Rater==3]
)

r2 <- factor(raters_2_and_3_on_SelMeth$r2,levels=1:4)
r3 <- factor(raters_2_and_3_on_SelMeth$r3,levels=1:4)
(t23 <- table(r2,r3))
```

```
##    r3
## r2  1 2 3 4
##   1 1 0 0 0
##   2 2 7 1 0
##   3 0 1 1 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 2 and rater 3 for the rubric SelMeth have the same rate in 8/13 of the cases. From the rubric SelMeth, rater 2 and 3 do not always disagree with each other.
agreement.results[5,4] = round(8/13,2)
```

InterpRes rubric

```
raters_1_and_2_on_InterpRes <- data.frame(r1=repeated$InterpRes[repeated$Rater==1],
                                       r2=repeated$InterpRes[repeated$Rater==2],
                                       a1=repeated$Artifact[repeated$Rater==1],
                                       a2=repeated$Artifact[repeated$Rater==2]
)

r1 <- factor(raters_1_and_2_on_InterpRes$r1,levels=1:4)
r2 <- factor(raters_1_and_2_on_InterpRes$r2,levels=1:4)
(t12 <- table(r1,r2))
```

```
##    r2
## r1  1 2 3 4
##   1 0 0 0 0
##   2 0 3 1 1
##   3 0 3 5 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 2 for the rubric InterpRes have the same rate in 8/13 of the cases. From the rubric InterpRes, rater 1 and 2 do not always disagree with each other.
agreement.results[3,3] = round(8/13,2)
```

```
raters_1_and_3_on_InterpRes <- data.frame(r1=repeated$InterpRes[repeated$Rater==1],
                                          r3=repeated$InterpRes[repeated$Rater==3],
                                          a1=repeated$Artifact[repeated$Rater==1],
                                          a3=repeated$Artifact[repeated$Rater==3]
)

r1 <- factor(raters_1_and_3_on_InterpRes$r1,levels=1:4)
r3 <- factor(raters_1_and_3_on_InterpRes$r3,levels=1:4)
(t13 <- table(r1,r3))
```

```
##    r3
## r1  1 2 3 4
##   1 0 0 0 0
##   2 1 3 1 0
##   3 0 4 4 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 3 for the rubric InterpRes have the same rate in 7/13 of the cases. From the rubric InterpRes, rater 1 and 3 do not always disagree with each other.
agreement.results[3,5] = round(7/13,2)
```

```
raters_2_and_3_on_InterpRes <- data.frame(r2=repeated$InterpRes[repeated$Rater==2],
                                          r3=repeated$InterpRes[repeated$Rater==3],
                                          a2=repeated$Artifact[repeated$Rater==2],
                                          a3=repeated$Artifact[repeated$Rater==3]
)

r2 <- factor(raters_2_and_3_on_InterpRes$r2,levels=1:4)
r3 <- factor(raters_2_and_3_on_InterpRes$r3,levels=1:4)
(t23 <- table(r2,r3))
```

```
##    r3
## r2  1 2 3 4
##   1 0 0 0 0
##   2 1 4 1 0
##   3 0 2 4 0
##   4 0 1 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 2 and rater 3 for the rubric InterpRes have the same rate in 8/13 of the cases. From the rubric InterpRes, rater 2 and 3 do not always disagree with each other.
agreement.results[3,4] = round(8/13,2)
```

VisOrg rubric

```
raters_1_and_2_on_VisOrg <- data.frame(r1=repeated$VisOrg[repeated$Rater==1],
                                       r2=repeated$VisOrg[repeated$Rater==2],
                                       a1=repeated$Artifact[repeated$Rater==1],
                                       a2=repeated$Artifact[repeated$Rater==2]
)

r1 <- factor(raters_1_and_2_on_VisOrg$r1,levels=1:4)
r2 <- factor(raters_1_and_2_on_VisOrg$r2,levels=1:4)
(t12 <- table(r1,r2))
```

```
##    r2
## r1  1 2 3 4
##   1 1 0 0 0
##   2 0 4 5 0
##   3 0 1 2 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 2 for the rubric VisOrg have the same rate in 6/13 of the cases. From the rubric VisOrg, rater 1 and 2 do not always disagree with each other.
agreement.results[7,3] = round(6/13,2)
```

```
raters_1_and_3_on_VisOrg <- data.frame(r1=repeated$VisOrg[repeated$Rater==1],
                                       r3=repeated$VisOrg[repeated$Rater==3],
                                       a1=repeated$Artifact[repeated$Rater==1],
                                       a3=repeated$Artifact[repeated$Rater==3]
)

r1 <- factor(raters_1_and_3_on_VisOrg$r1,levels=1:4)
r3 <- factor(raters_1_and_3_on_VisOrg$r3,levels=1:4)
(t13 <- table(r1,r3))
```

```
##    r3
## r1  1 2 3 4
##   1 1 0 0 0
##   2 0 7 2 0
##   3 0 1 2 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 3 for the rubric VisOrg have the same rate in 9/13 of the cases. From the rubric VisOrg, rater 1 and 3 do not always disagree with each other.
agreement.results[7,5] = round(9/13,2)
```

```
raters_2_and_3_on_VisOrg <- data.frame(r2=repeated$VisOrg[repeated$Rater==2],
                                       r3=repeated$VisOrg[repeated$Rater==3],
                                       a2=repeated$Artifact[repeated$Rater==2],
                                       a3=repeated$Artifact[repeated$Rater==3]
)

r2 <- factor(raters_2_and_3_on_VisOrg$r2,levels=1:4)
r3 <- factor(raters_2_and_3_on_VisOrg$r3,levels=1:4)
(t23 <- table(r2,r3))
```

```
##    r3
## r2  1 2 3 4
##   1 1 0 0 0
##   2 0 5 0 0
##   3 0 3 4 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 2 and rater 3 for the rubric VisOrg have the same rate in 9/13 of the cases. From the rubric VisOrg, rater 2 and 3 do not always disagree with each other.
agreement.results[7,4] = round(9/13,2)
```

TxtOrg rubric

```
raters_1_and_2_on_TxtOrg <- data.frame(r1=repeated$TxtOrg[repeated$Rater==1],
                                       r2=repeated$TxtOrg[repeated$Rater==2],
                                       a1=repeated$Artifact[repeated$Rater==1],
                                       a2=repeated$Artifact[repeated$Rater==2]
)

r1 <- factor(raters_1_and_2_on_TxtOrg$r1,levels=1:4)
r2 <- factor(raters_1_and_2_on_TxtOrg$r2,levels=1:4)
(t12 <- table(r1,r2))
```

```
##    r2
## r1  1 2 3 4
##   1 0 0 0 0
##   2 0 2 2 0
##   3 0 1 7 0
##   4 1 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 2 for the rubric TxtOrg have the same rate in 9/13 of the cases. From the rubric TxtOrg, rater 1 and 2 do not always disagree with each other.
agreement.results[6,3] = round(9/13,2)
```

```
raters_1_and_3_on_TxtOrg <- data.frame(r1=repeated$TxtOrg[repeated$Rater==1],
                                       r3=repeated$TxtOrg[repeated$Rater==3],
                                       a1=repeated$Artifact[repeated$Rater==1],
                                       a3=repeated$Artifact[repeated$Rater==3]
)

r1 <- factor(raters_1_and_3_on_TxtOrg$r1,levels=1:4)
r3 <- factor(raters_1_and_3_on_TxtOrg$r3,levels=1:4)
(t13 <- table(r1,r3))
```

```
##    r3
## r1  1 2 3 4
##   1 0 0 0 0
##   2 1 1 2 0
##   3 0 1 7 0
##   4 0 1 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 1 and rater 3 for
the rubric TxtOrg have the same rate in 8/13 of the cases. From the rubric TxtOrg, rater 1 and 3
do not always disagree with each other.
agreement.results[6,5] = round(8/13,2)
```

```
raters_2_and_3_on_TxtOrg <- data.frame(r2=repeated$TxtOrg[repeated$Rater==2],
                                       r3=repeated$TxtOrg[repeated$Rater==3],
                                       a2=repeated$Artifact[repeated$Rater==2],
                                       a3=repeated$Artifact[repeated$Rater==3]
)

r2 <- factor(raters_2_and_3_on_TxtOrg$r2,levels=1:4)
r3 <- factor(raters_2_and_3_on_TxtOrg$r3,levels=1:4)
(t23 <- table(r2,r3))
```

```
##    r3
## r2  1 2 3 4
##   1 0 1 0 0
##   2 1 0 2 0
##   3 0 2 7 0
##   4 0 0 0 0
```

```
#For the artifacts which are rated by all three raters, we find that the rater 2 and rater 3 for
the rubric TxtOrg have the same rate in 7/13 of the cases. From the rubric TxtOrg, rater 2 and 3
do not always disagree with each other.
agreement.results[6,4] = round(7/13,2)
```

```
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 0.62 0.62
## TxtOrg           0.19       0.14 0.69 0.54 0.62
## VisOrg           0.66       0.59 0.46 0.69 0.69
```

q3

```
tall$Rater = as.factor(tall$Rater)
```

fixed effect models on 13 common artifacts on rubric “RsrchQ”

```
library(LMERConvenienceFunctions)
```

```
## Warning: package 'LMERConvenienceFunctions' was built under R version 4.1.2
```

```
library(RLRsim)
```

```
## Warning: package 'RLRsim' was built under R version 4.1.2
```

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

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

```
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ======================================================
## ===              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
```

```
formula(tmp.back_elim)
```

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

fixed effect models on 13 common artifacts for each rubric

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

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

  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
  }

  model.formula.13[[i]] <- formula(tmp_final)
}
```

```
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ========================================================
## ===              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
```

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

```
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ========================================================
## ===              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
```

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

```
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## =====================================================
## ===              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
```

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

```
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ========================================================
## ===              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
```

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

```
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ======================================================
## ===              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
```

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

```
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ========================================================
## ===              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
```

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

```
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ======================================================
## ===              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
```

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

```
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)
```

```
# eliminate the rows with missing value
tall.nonmissing <- tall[-c(161,684),]

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

```
## [1] X        Rater    Artifact Repeated Semester Sex      Rubric   Rating
## <0 rows> (or 0-length row.names)
```

```
tall.nonmissing <- tall.nonmissing[tall.nonmissing$Sex!="--",] ## eliminate them
model.formula.alldata <- as.list(rep(NA,7))
names(model.formula.alldata) <- Rubric.names

for (i in Rubric.names) {
  ## fit each base model
  rubric.data <- tall.nonmissing[tall.nonmissing$Rubric==i,]
  tmp <- lmer(as.numeric(Rating) ~ -1 + as.factor(Rater) +
  Semester + Sex + (1|Artifact),
  data=rubric.data,REML=FALSE)
  ## do backwards elimination
  tmp.back_elim <- fitLMER.fnc(tmp,set.REML.FALSE = TRUE,log.file.name = FALSE)
  ## check to see if the raters are significantly different from one another
  tmp.single_intercept <- update(tmp.back_elim, . ~ . + 1 - as.factor(Rater))
  pval <- anova(tmp.single_intercept,tmp.back_elim)$"Pr(>Chisq)"[2]
  ## choose the best model
  if (pval<=0.05) {
    tmp_final <- tmp.back_elim
  } else {
    tmp_final <- tmp.single_intercept
  }

  model.formula.alldata[[i]] <- formula(tmp_final)
}
```

```
## Warning in fitLMER.fnc(tmp, set.REML.FALSE = TRUE, log.file.name = FALSE): Argument "ran.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ========================================================
## ===              backfitting fixed effects          ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.6474 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.3309 >= 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.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.8292 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.6014 >= 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.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.4701 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.2935 >= 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.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## =====================================================
## ===              backfitting fixed effects         ===
## =====================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Semester" = 0.4446 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Sex" = 0.3417 >= 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.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## =======================================================
## ===              backfitting fixed effects          ===
## =======================================================
## 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 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.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ========================================================
## ===              backfitting fixed effects         ===
## ========================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.5925 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.1874 >= 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.effe
cts" is empty, which means you will not be forward-fitting the random effect structure of your m
odel. You could just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## =====================================================
## ===              backfitting fixed effects         ===
## =====================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.2186 >= 0.05
##     not part of higher-order interaction
##     removing term
##   iteration 2
##     p-value for term "Semester" = 0.1977 >= 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)
```

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

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

```
# Start with the "combined" intercept-only model
comb.0 <- lmer(as.numeric(Rating) ~ 1 + (0 + Rubric | Artifact),
data=tall.nonmissing)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00236116 (tol = 0.002, component 1)
```

```
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: 1481.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0251 -0.4969 -0.0753  0.5165  3.7820
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.6484   0.8052
##           RubricInitEDA   0.3779   0.6147   0.27
##           RubricInterpRes 0.2524   0.5024   0.02 0.79
##           RubricRsrchQ    0.1734   0.4164   0.40 0.51 0.74
##           RubricSelMeth   0.1034   0.3216   0.58 0.39 0.42 0.29
##           RubricTxtOrg    0.3946   0.6281   0.04 0.69 0.80 0.64 0.25
##           RubricVisOrg    0.3152   0.5615   0.19 0.78 0.77 0.60 0.31 0.79
##  Residual                 0.1942   0.4407
## Number of obs: 817, groups:  Artifact, 91
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  2.24700    0.04048   55.51
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00236116 (tol = 0.002, component 1)
```

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

summary(comb.full)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) +
##     Semester + Sex + Repeated + Rubric
##    Data: tall.nonmissing
##
## REML criterion at convergence: 1436.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1156 -0.5058 -0.0211  0.5203  3.8014
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.54872  0.7408
##           RubricInitEDA   0.34949  0.5912   0.47
##           RubricInterpRes 0.17485  0.4181   0.23 0.75
##           RubricRsrchQ    0.16889  0.4110   0.59 0.45 0.71
##           RubricSelMeth   0.06806  0.2609   0.40 0.61 0.75 0.42
##           RubricTxtOrg    0.26217  0.5120   0.34 0.62 0.71 0.57 0.67
##           RubricVisOrg    0.25621  0.5062   0.35 0.74 0.68 0.52 0.42 0.76
##  Residual                 0.18836  0.4340
## Number of obs: 817, groups:  Artifact, 91
##
## Fixed effects:
##                    Estimate Std. Error t value
## (Intercept)        2.820352   0.388481   7.260
## as.factor(Rater)2  0.002014   0.054802   0.037
## as.factor(Rater)3 -0.174685   0.054959  -3.178
## SemesterS19       -0.175032   0.087853  -1.992
## SexF              -0.802713   0.383749  -2.092
## SexM              -0.792258   0.382756  -2.070
## Repeated          -0.074405   0.098553  -0.755
## RubricInitEDA      0.541262   0.094891   5.704
## RubricInterpRes    0.580887   0.100024   5.807
## RubricRsrchQ       0.455982   0.086769   5.255
## RubricSelMeth      0.162876   0.093285   1.746
## RubricTxtOrg       0.685736   0.098736   6.945
## RubricVisOrg       0.524294   0.098256   5.336
```

```
##
## Correlation matrix not shown by default, as p = 13 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
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, wh
ich means you will not be forward-fitting the random effect structure of your model. You could j
ust as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 1
##   iteration 1
##     p-value for term "Sex" = 0.091 >= 0.05
##     not part of higher-order interaction
```

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

```
##     removing term
##   iteration 2
##     p-value for term "Repeated" = 0.086 >= 0.05
##     not part of higher-order interaction
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00201387 (tol = 0.002, component 1)
```

```
##     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
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00201387 (tol = 0.002, component 1)
```

```
## 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: 1435.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1189 -0.5102 -0.0149  0.5188  3.7768
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.56192  0.7496
##           RubricInitEDA   0.35041  0.5920   0.48
##           RubricInterpRes 0.17215  0.4149   0.25 0.75
##           RubricRsrchQ    0.17153  0.4142   0.60 0.45 0.72
##           RubricSelMeth   0.06984  0.2643   0.44 0.62 0.75 0.44
##           RubricTxtOrg    0.25478  0.5048   0.35 0.62 0.70 0.56 0.67
##           RubricVisOrg    0.26008  0.5100   0.37 0.74 0.69 0.53 0.44 0.76
##  Residual                 0.18910  0.4349
## Number of obs: 817, groups:  Artifact, 91
##
## Fixed effects:
##                    Estimate Std. Error t value
## (Intercept)        2.022481   0.098863  20.457
## as.factor(Rater)2 -0.001366   0.054995  -0.025
## as.factor(Rater)3 -0.168823   0.054994  -3.070
## SemesterS19       -0.184473   0.084144  -2.192
## RubricInitEDA      0.541715   0.094889   5.709
## RubricInterpRes    0.580226   0.100003   5.802
## RubricRsrchQ       0.453253   0.086692   5.228
## RubricSelMeth      0.156549   0.092719   1.688
## RubricTxtOrg       0.685878   0.098769   6.944
## RubricVisOrg       0.522884   0.098208   5.324
##
## Correlation of Fixed Effects:
##             (Intr) a.(R)2 a.(R)3 SmsS19 RbIEDA RbrcIR RbrcRQ RbrcSM RbrcTO
## as.fctr(R)2 -0.282
## as.fctr(R)3 -0.282  0.500
## SemesterS19 -0.267  0.016  0.016
## RubrcIntEDA -0.610 -0.001  0.000 -0.001
## RbrcIntrpRs -0.732 -0.001  0.000  0.000  0.734
## RubrcRsrchQ -0.698 -0.001  0.000  0.002  0.588  0.756
## RubricSlMth -0.777  0.000  0.000  0.006  0.663  0.779  0.689
## RubrcTxtOrg -0.679 -0.001  0.000 -0.001  0.676  0.751  0.684  0.728
## RubricVsOrg -0.672 -0.001 -0.001  0.000  0.716  0.745  0.668  0.682  0.751
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00201387 (tol = 0.002, component 1)
```

```
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| = 0.00404493 (tol = 0.002, component 1)
```

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

```
## 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
##
## REML criterion at convergence: 1435.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9270 -0.5054 -0.0615  0.4975  3.6574
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.48972  0.6998
##           RubricInitEDA   0.35174  0.5931   0.43
##           RubricInterpRes 0.15058  0.3881   0.35 0.80
##           RubricRsrchQ    0.16850  0.4105   0.67 0.44 0.74
##           RubricSelMeth   0.06776  0.2603   0.47 0.65 0.80 0.53
##           RubricTxtOrg    0.25248  0.5025   0.45 0.65 0.67 0.61 0.63
##           RubricVisOrg    0.25615  0.5061   0.36 0.73 0.69 0.58 0.38 0.76
##  Residual                 0.18788  0.4335
## Number of obs: 817, groups:  Artifact, 91
##
## Fixed effects:
##                                              Estimate Std. Error t value
## (Intercept)                                  1.751457   0.136455  12.835
## as.factor(Rater)2                            0.302532   0.154870   1.953
## as.factor(Rater)3                            0.250652   0.154870   1.618
## SemesterS19                                 -0.142032   0.250415  -0.567
## RubricInitEDA                                0.762356   0.164513   4.634
## RubricInterpRes                              0.975228   0.161437   6.041
## RubricRsrchQ                                 0.707103   0.146783   4.817
## RubricSelMeth                                0.456378   0.154441   2.955
## RubricTxtOrg                                 1.007111   0.160333   6.281
## RubricVisOrg                                 0.644122   0.165805   3.885
## as.factor(Rater)2:SemesterS19                0.267585   0.303454   0.882
## as.factor(Rater)3:SemesterS19               -0.082424   0.300222  -0.275
## as.factor(Rater)2:RubricInitEDA             -0.324259   0.203594  -1.593
## as.factor(Rater)3:RubricInitEDA             -0.383669   0.203594  -1.884
## as.factor(Rater)2:RubricInterpRes           -0.469369   0.200519  -2.341
## as.factor(Rater)3:RubricInterpRes           -0.712153   0.200519  -3.552
## as.factor(Rater)2:RubricRsrchQ              -0.447406   0.188811  -2.370
## as.factor(Rater)3:RubricRsrchQ              -0.475446   0.188811  -2.518
## as.factor(Rater)2:RubricSelMeth             -0.301919   0.193077  -1.564
## as.factor(Rater)3:RubricSelMeth             -0.354607   0.193077  -1.837
## as.factor(Rater)2:RubricTxtOrg              -0.448519   0.200473  -2.237
## as.factor(Rater)3:RubricTxtOrg              -0.422438   0.200473  -2.107
## as.factor(Rater)2:RubricVisOrg               0.008512   0.204451   0.042
## as.factor(Rater)3:RubricVisOrg              -0.293205   0.204451  -1.434
## SemesterS19:RubricInitEDA                   -0.046337   0.300298  -0.154
## SemesterS19:RubricInterpRes                  0.133411   0.294527   0.453
## SemesterS19:RubricRsrchQ                     0.138282   0.266735   0.518
```

```
## SemesterS19:RubricSelMeth                        -0.081123   0.281446  -0.288
## SemesterS19:RubricTxtOrg                          0.171902   0.292278   0.588
## SemesterS19:RubricVisOrg                          0.152072   0.301186   0.505
## as.factor(Rater)2:SemesterS19:RubricInitEDA       0.020167   0.391289   0.052
## as.factor(Rater)3:SemesterS19:RubricInitEDA       0.258867   0.388223   0.667
## as.factor(Rater)2:SemesterS19:RubricInterpRes -0.268671   0.384281  -0.699
## as.factor(Rater)3:SemesterS19:RubricInterpRes -0.152972   0.381565  -0.401
## as.factor(Rater)2:SemesterS19:RubricRsrchQ    -0.218357   0.359363  -0.608
## as.factor(Rater)3:SemesterS19:RubricRsrchQ     0.354431   0.355607   0.997
## as.factor(Rater)2:SemesterS19:RubricSelMeth   -0.404640   0.368934  -1.097
## as.factor(Rater)3:SemesterS19:RubricSelMeth   -0.203139   0.365912  -0.555
## as.factor(Rater)2:SemesterS19:RubricTxtOrg    -0.542950   0.384072  -1.414
## as.factor(Rater)3:SemesterS19:RubricTxtOrg    -0.305750   0.381009  -0.802
## as.factor(Rater)2:SemesterS19:RubricVisOrg    -0.604388   0.391639  -1.543
## as.factor(Rater)3:SemesterS19:RubricVisOrg    -0.183663   0.388828  -0.472
```

```
##
## Correlation matrix not shown by default, as p = 42 > 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| = 0.00404493 (tol = 0.002, component 1)
```

```
# The result for interaction model is not quite converged, so switching optimizers and increasin
g 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
```

```
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: 1435.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9270 -0.5054 -0.0614  0.4975  3.6575
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.48972  0.6998
##           RubricInitEDA   0.35173  0.5931   0.43
##           RubricInterpRes 0.15058  0.3880   0.34 0.80
##           RubricRsrchQ    0.16853  0.4105   0.67 0.44 0.74
##           RubricSelMeth   0.06777  0.2603   0.47 0.65 0.80 0.53
##           RubricTxtOrg    0.25248  0.5025   0.45 0.65 0.67 0.61 0.63
##           RubricVisOrg    0.25614  0.5061   0.36 0.73 0.69 0.58 0.38 0.76
##  Residual                 0.18788  0.4334
## Number of obs: 817, groups:  Artifact, 91
##
## Fixed effects:
##                                            Estimate Std. Error t value
## (Intercept)                                1.751456   0.136454  12.835
## as.factor(Rater)2                          0.302533   0.154869   1.953
## as.factor(Rater)3                          0.250656   0.154869   1.619
## SemesterS19                               -0.142029   0.250413  -0.567
## RubricInitEDA                              0.762359   0.164514   4.634
## RubricInterpRes                            0.975227   0.161435   6.041
## RubricRsrchQ                               0.707102   0.146783   4.817
## RubricSelMeth                              0.456381   0.154440   2.955
## RubricTxtOrg                               1.007110   0.160332   6.281
## RubricVisOrg                               0.644127   0.165802   3.885
## as.factor(Rater)2:SemesterS19              0.267582   0.303451   0.882
## as.factor(Rater)3:SemesterS19             -0.082426   0.300220  -0.275
## as.factor(Rater)2:RubricInitEDA           -0.324263   0.203593  -1.593
## as.factor(Rater)3:RubricInitEDA           -0.383677   0.203593  -1.885
## as.factor(Rater)2:RubricInterpRes         -0.469370   0.200517  -2.341
## as.factor(Rater)3:RubricInterpRes         -0.712152   0.200517  -3.552
## as.factor(Rater)2:RubricRsrchQ            -0.447408   0.188810  -2.370
## as.factor(Rater)3:RubricRsrchQ            -0.475448   0.188810  -2.518
## as.factor(Rater)2:RubricSelMeth           -0.301920   0.193075  -1.564
## as.factor(Rater)3:RubricSelMeth           -0.354612   0.193075  -1.837
## as.factor(Rater)2:RubricTxtOrg            -0.448517   0.200471  -2.237
## as.factor(Rater)3:RubricTxtOrg            -0.422441   0.200471  -2.107
## as.factor(Rater)2:RubricVisOrg             0.008513   0.204448   0.042
## as.factor(Rater)3:RubricVisOrg            -0.293212   0.204448  -1.434
## SemesterS19:RubricInitEDA                 -0.046340   0.300300  -0.154
## SemesterS19:RubricInterpRes                0.133408   0.294525   0.453
```

```
## SemesterS19:RubricRsrchQ                        0.138291   0.266735   0.518
## SemesterS19:RubricSelMeth                      -0.081130   0.281445  -0.288
## SemesterS19:RubricTxtOrg                        0.171896   0.292275   0.588
## SemesterS19:RubricVisOrg                        0.152067   0.301182   0.505
## as.factor(Rater)2:SemesterS19:RubricInitEDA     0.020173   0.391289   0.052
## as.factor(Rater)3:SemesterS19:RubricInitEDA     0.258867   0.388223   0.667
## as.factor(Rater)2:SemesterS19:RubricInterpRes -0.268664   0.384277  -0.699
## as.factor(Rater)3:SemesterS19:RubricInterpRes -0.152973   0.381562  -0.401
## as.factor(Rater)2:SemesterS19:RubricRsrchQ    -0.218376   0.359363  -0.608
## as.factor(Rater)3:SemesterS19:RubricRsrchQ     0.354425   0.355606   0.997
## as.factor(Rater)2:SemesterS19:RubricSelMeth   -0.404633   0.368931  -1.097
## as.factor(Rater)3:SemesterS19:RubricSelMeth   -0.203130   0.365910  -0.555
## as.factor(Rater)2:SemesterS19:RubricTxtOrg    -0.542944   0.384068  -1.414
## as.factor(Rater)3:SemesterS19:RubricTxtOrg    -0.305748   0.381005  -0.802
## as.factor(Rater)2:SemesterS19:RubricVisOrg    -0.604387   0.391633  -1.543
## as.factor(Rater)3:SemesterS19:RubricVisOrg    -0.183661   0.388822  -0.472
```

```
##
## Correlation matrix not shown by default, as p = 42 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

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

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

```
## Warning in fitLMER.fnc(comb.inter.u, log.file.name = FALSE): Argument "ran.effects" is empty,
which means you will not be forward-fitting the random effect structure of your model. You could
just as well run function "bfFixefLMER_F.fnc" or "bfFixefLMER_t.fnc".
## TRUE
```

```
## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## processing model terms of interaction level 3
##   iteration 1
##     p-value for term "as.factor(Rater):Semester:Rubric" = 0.5402 >= 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" = 0.5569 >= 0.05
##     not part of higher-order interaction
```

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

```
##     removing term
##   iteration 3
##     p-value for term "Semester:Rubric" = 0.0696 >= 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: 1430.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9356 -0.5143 -0.0409  0.4895  3.5799
##
## Random effects:
##  Groups   Name            Variance Std.Dev. Corr
##  Artifact RubricCritDes   0.50832  0.7130
##           RubricInitEDA   0.35400  0.5950   0.46
##           RubricInterpRes 0.15704  0.3963   0.38 0.82
##           RubricRsrchQ    0.18267  0.4274   0.64 0.45 0.73
##           RubricSelMeth   0.07263  0.2695   0.45 0.62 0.76 0.41
##           RubricTxtOrg    0.25909  0.5090   0.43 0.64 0.68 0.56 0.65
##           RubricVisOrg    0.25612  0.5061   0.36 0.72 0.69 0.53 0.41 0.78
##  Residual                 0.18484  0.4299
## Number of obs: 817, groups:  Artifact, 91
##
## Fixed effects:
##                                   Estimate Std. Error t value
## (Intercept)                        1.77051    0.11793  15.013
## as.factor(Rater)2                  0.36517    0.13281   2.750
## as.factor(Rater)3                  0.22498    0.13235   1.700
## SemesterS19                       -0.18904    0.08384  -2.255
## RubricInitEDA                      0.74459    0.13625   5.465
## RubricInterpRes                    1.01239    0.13425   7.541
## RubricRsrchQ                       0.74689    0.12376   6.035
## RubricSelMeth                      0.42339    0.12972   3.264
## RubricTxtOrg                       1.04783    0.13506   7.758
## RubricVisOrg                       0.68170    0.13885   4.910
## as.factor(Rater)2:RubricInitEDA   -0.30781    0.17212  -1.788
## as.factor(Rater)3:RubricInitEDA   -0.30463    0.17167  -1.775
## as.factor(Rater)2:RubricInterpRes -0.53771    0.16967  -3.169
## as.factor(Rater)3:RubricInterpRes -0.75345    0.16929  -4.451
## as.factor(Rater)2:RubricRsrchQ    -0.50228    0.16112  -3.117
## as.factor(Rater)3:RubricRsrchQ    -0.37510    0.16057  -2.336
## as.factor(Rater)2:RubricSelMeth   -0.39806    0.16417  -2.425
## as.factor(Rater)3:RubricSelMeth   -0.40351    0.16374  -2.464
## as.factor(Rater)2:RubricTxtOrg    -0.58389    0.17104  -3.414
## as.factor(Rater)3:RubricTxtOrg    -0.49781    0.17061  -2.918
## as.factor(Rater)2:RubricVisOrg    -0.14568    0.17393  -0.838
## as.factor(Rater)3:RubricVisOrg    -0.33879    0.17354  -1.952
```

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

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

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

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

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

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

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

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

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

```
##  Groups   Name            Std.Dev. Corr
##  Artifact RubricCritDes   0.69980
##           RubricInitEDA   0.59307  0.427
##           RubricInterpRes 0.38804  0.345 0.802
##           RubricRsrchQ    0.41052  0.667 0.444 0.736
##           RubricSelMeth   0.26032  0.474 0.651 0.802 0.527
##           RubricTxtOrg    0.50247  0.445 0.655 0.672 0.611 0.630
##           RubricVisOrg    0.50610  0.364 0.733 0.685 0.580 0.378 0.762
##  Residual                 0.43345
```

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

```
##  Groups   Name            Std.Dev. Corr
##  Artifact RubricCritDes   0.71296
##           RubricInitEDA   0.59498  0.455
##           RubricInterpRes 0.39628  0.376 0.817
##           RubricRsrchQ    0.42740  0.642 0.454 0.727
##           RubricSelMeth   0.26949  0.455 0.616 0.758 0.407
##           RubricTxtOrg    0.50901  0.427 0.643 0.681 0.558 0.645
##           RubricVisOrg    0.50608  0.355 0.722 0.688 0.527 0.409 0.778
##  Residual                 0.42993
```

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

```
##  Groups   Name            Std.Dev. Corr
##  Artifact RubricCritDes   0.74961
##           RubricInitEDA   0.59195  0.477
##           RubricInterpRes 0.41491  0.252 0.755
##           RubricRsrchQ    0.41416  0.602 0.452 0.715
##           RubricSelMeth   0.26428  0.436 0.620 0.751 0.440
##           RubricTxtOrg    0.50475  0.346 0.622 0.699 0.564 0.665
##           RubricVisOrg    0.50998  0.365 0.739 0.689 0.531 0.442 0.760
##  Residual                 0.43485
```

```
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 + as.factor(Rater):Rubric
## comb.inter.u: as.numeric(Rating) ~ (0 + Rubric | Artifact) + as.factor(Rater) + Semester + Ru
bric + as.factor(Rater):Semester + as.factor(Rater):Rubric + Semester:Rubric + as.factor(Rater):
Semester:Rubric
##                 npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## comb.back_elim    39 1475.2 1658.7 -698.58   1397.2
## comb.inter_elim   51 1465.5 1705.5 -681.76   1363.5 33.653 12   0.000765 ***
## comb.inter.u      71 1481.8 1815.9 -669.91   1339.8 23.694 20   0.256027
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 4

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Firstly get rid of the rows with missing value in rubric ratings
rating <- ratings[is.na(ratings$CritDes) == FALSE, ]
rating <- rating[is.na(rating$VisOrg) == FALSE, ]
knitr::kable(rating %>%
  group_by(Sex) %>%
  mutate(ratingsum = as.numeric(RsrchQ) +
    as.numeric(CritDes) + as.numeric(InitEDA) +
    as.numeric(SelMeth) + as.numeric(InterpRes) +
    as.numeric(VisOrg) + as.numeric(TxtOrg)) %>%
  summarize(total_score = sum(ratingsum),
    mean_score = mean(ratingsum), count = total_score/mean_score))
```

Sex	total_score	mean_score	count
–	21	21.00000	1
F	1004	16.19355	62
M	841	16.17308	52

```
knitr::kable(rating %>%
  group_by(Sex) %>%
  mutate(ratingsum = as.numeric(TxtOrg))%>%
  summarize(total_score = sum(ratingsum),
    mean_score = mean(ratingsum), count = total_score/mean_score))
```

Sex	total_score	mean_score	count
–	3	3.000000	1
F	163	2.629032	62
M	133	2.557692	52

```
knitr::kable(rating %>%
  group_by(Sex) %>%
  mutate(ratingsum = as.numeric(RsrchQ))%>%
  summarize(total_score = sum(ratingsum),
    mean_score = mean(ratingsum), count = total_score/mean_score))
```

Sex	total_score	mean_score	count
–	3	3.000000	1
F	149	2.403226	62
M	119	2.288461	52

```
knitr::kable(rating %>%
  group_by(Sex) %>%
  mutate(ratingsum = as.numeric(CritDes))%>%
  summarize(total_score = sum(ratingsum),
    mean_score = mean(ratingsum), count = total_score/mean_score))
```

Sex	total_score	mean_score	count
–	3	3.000000	1
F	109	1.758064	62
M	102	1.961539	52

```
knitr::kable(rating %>%
  group_by(Sex) %>%
  mutate(ratingsum = as.numeric(InitEDA))%>%
  summarize(total_score = sum(ratingsum),
    mean_score = mean(ratingsum), count = total_score/mean_score))
```

Sex	total_score	mean_score	count
–	3	3.000000	1
F	150	2.419355	62
M	128	2.461539	52

```
knitr::kable(rating %>%
  group_by(Sex) %>%
  mutate(ratingsum = as.numeric(SelMeth))%>%
  summarize(total_score = sum(ratingsum),
    mean_score = mean(ratingsum), count = total_score/mean_score))
```

Sex	total_score	mean_score	count
–	3	3.000000	1

Sex	total_score	mean_score	count
F	122	1.967742	62
M	112	2.153846	52

```
knitr::kable(rating %>%
  group_by(Sex) %>%
  mutate(ratingsum = as.numeric(InterpRes))%>%
  summarize(total_score = sum(ratingsum),
    mean_score = mean(ratingsum), count = total_score/mean_score))
```

Sex	total_score	mean_score	count
–	3	3.000000	1
F	156	2.516129	62
M	127	2.442308	52

```
knitr::kable(rating %>%
  group_by(Sex) %>%
  mutate(ratingsum = as.numeric(VisOrg))%>%
  summarize(total_score = sum(ratingsum),
    mean_score = mean(ratingsum), count = total_score/mean_score))
```

Sex	total_score	mean_score	count
–	3	3.000000	1
F	155	2.500000	62
M	120	2.307692	52