Teaching Assistant Evaluation Data Analysis

STAT 479: Special Topics in Statistics (003)

Zhiyi Chen (zchen556@wisc.edu) Ho Gun Kang (hkang79@wisc.edu)

18 December, 2020

Contents

Introduction

Teaching evaluation is the process for students to review and rate the teaching performance, effectiveness, and satisfactions in the classroom. The evaluations will be submitted to the teachers for their future professional development, and for improving their teaching methods. Teaching assistants are also teachers that can enhance students' understanding and academic performances in the class and as such, evaluations of them are also required. Evaluations of teaching assistants might be related to several features they possess, which begs the question of what those features might be.

This study will be mainly focused on the the factors that influence the teaching assistant evaluation from University of Wisconsin-Madison, and the dataset used in this study is provided by Wei-Yin Loh and Tjen-Sien Lim (Department of Statistics, University of Wisconsin-Madison) on the UCI Machine Learning Repository. The dataset consists of evaluations of teaching performance over three regular semesters and two summer semesters of 151 teaching assistants at the Statistics Department of the University of Wisconsin-Madison.

The paper is categorized into following parts: Exploratory Data Analysis will provide the visualizations of the data provided by the dataset. Then, Data-Processing section will demonstrate how the data was processed and diagnostics were run, to indicate if there were any transformations needed. Implementation section will be about two algorithms fit on the data: Generalized Linear Mixed Model and Lasso Regression Model. Then, we will explain the outcomes based on the models and discuss the possible future plans for the work.

Dataset Description

In the dataset, the scores of evaluations as the response variable are being examined, and the scores are divided into three roughly equal-sized levels, "low", "medium", and "high" to form the class attributes. Five explanatory variables are present along with the scores of the evaluations. The course instructor will be divided into 25 categories and the course will be divided into 26 categories. Binary categorical variables like whether the teaching assistant is a native English speaker and whether the course is taught in the summer session are also included. Class sizes are measured numerically. There are two categorical attributes with large numbers of categories. This study will examine whether these factors can significantly influence the evaluation scores, provide analysis, and possible further study. Since the dataset is obtained from the repository website, a degree of data manipulations was required. The exact categories of course instructor and course were not present in the dataset and as such, will not be deemed significant. Regression analysis was performed on the data to determine the statistical significance.

Exploratory Data Analysis

The response variable is the "Attribute" collected in the dataset, and in order to analyze the data numerically, all the levels from "low", "medium", to "high" are stored as "1", "2", and "3". As for the explanatory variables, "Language" indicates whether the teaching assistant is a native English speaker or not; it is stored as a binary variable with "1" being native English speaker and "2" being non-native speaker. The factor "Semester" is also a binary variable with level "1" being summer semester and "2" being regular semester. The variable "Size" is the number of students who attend that class, which is stored as numerical data. Lastly, both "Type" and "Course" factors in the dataset are categorical variables that represent the types of course instructors and of courses themselves.

In this section, we will create bar plots, box plots, histograms, and density plots to help visualize the data, and we will store the original dataset to a new data frame with "Attribute" replaced by "Levels" representing the teaching assistant evaluation scores. The new data frame will be used for the rest of the study.

Data Table

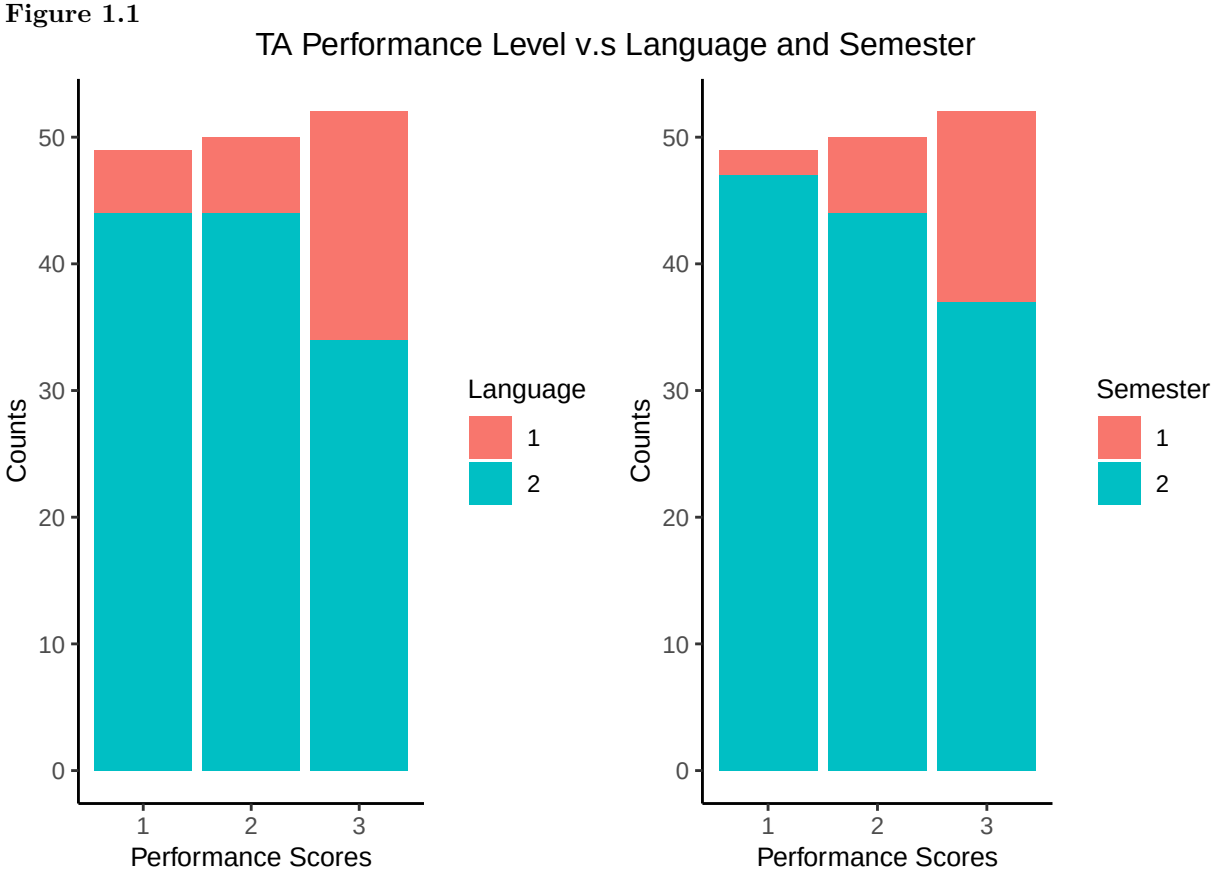
The first 6 rows of the dataset are shown in the table below, and, in total, there are 6 columns and 151 rows in this dataset.

	Language	Type	Course	Semester	Size	Attribute
1	1	23	3	1	19	3
2	2	15	3	1	17	3
3	1	23	3	2	49	3
4	1	5	2	2	33	3
5	2	7	11	2	55	3
6	2	23	3	1	20	3

Visualizations

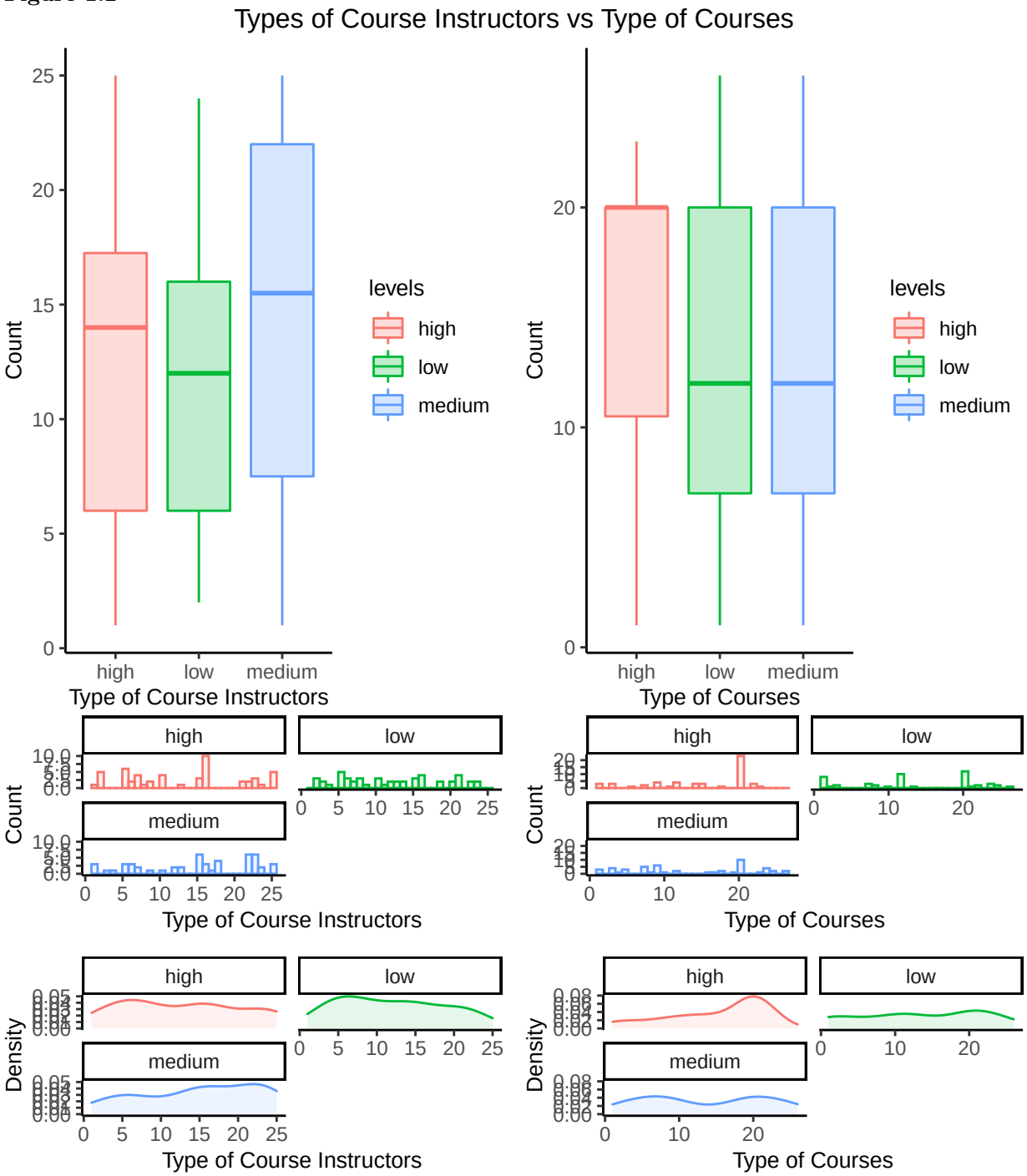
In Figure 1.1 below, the bar plot on the left is the performance scores of teacher assistants with three levels on the x-axis filled by the binary factor "Language", where the red bars represent teacher assistants who are English speakers, and the blue bars are teacher assistants who are non-English speakers. The bar plot on the right is the performance scores of teacher assistants with three levels on the x-axis filled by another binary factor "Semester", where the red bars are teacher assistants who teach in summer semesters, and blue bars are teacher assistants teach in regular semesters. It is clear teacher assistants who are non-English speakers and teach in regular semesters receive more attributes in all three levels, and teacher assistants who are English speakers and teach in summer sessions receive more "high" attributes than the other two levels. The significance of the factors "Language" and "Semester" will be performed using regression models in this study.

Figure 1.1



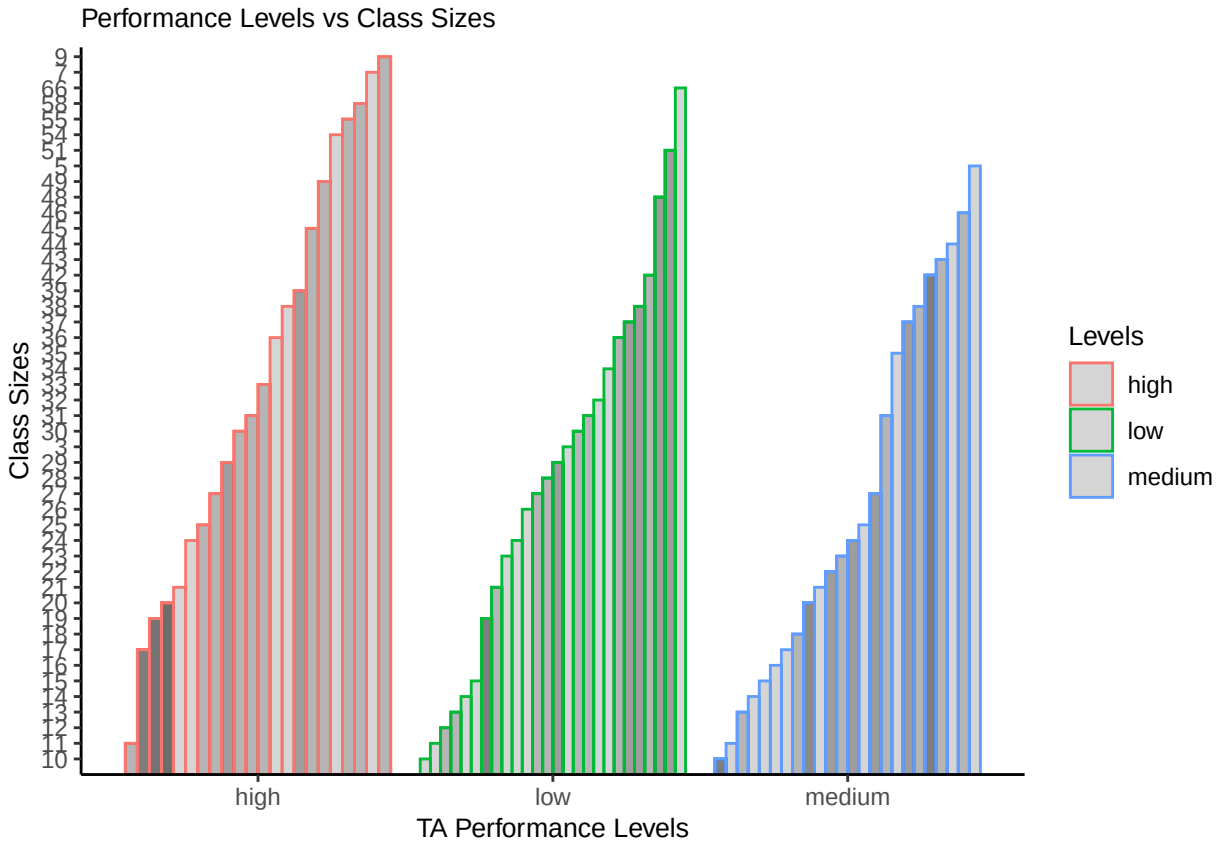
In Figure 1.2, Box plots, histograms, and density plots are used to visualize the differences of factor "Types" and "Courses" among three evaluation score levels. The plots with red colors are the teacher assistants with "high" attributes, the plots with blue colors are "medium" attributes, and the plots with green colors are "low" attributes. There is no clear trend shown in the plots below, further manipulations and transformations will be performed in later this study.

Figure 1.2



In Figure 1.3,bar plots are being used to visualize the class size distribution for each of the three levels of the evaluation scores and determine whether there are clear trends for class sizes and evaluation score levels. Red bars are the class sizes for teacher assistants who receive "high" attribute, green bars are teacher assistants who receive "low" attribute, and blue bars are teacher assistants who receive "medium" attribute. It seems like class sizes for teacher assistant who receive "high" attribute are generally larger than the class sizes for the other two levels. Further analysis will be applied to check the significance of the factors.

Figure 1.3



Data-Processing

Generalized linear mixed model, which is an extension of the original general linear model, is used in the data-processing section to test the significance, because the response variable is a multilevel variable, and the categorical variables courses and type of instructors are random effects as we assume that the levels chosen in the dataset are viewed as a sample from larger population. The other factors, semesters, languages, and class sizes, are fixed effects, as the levels are possible levels of interest. Then, after calling the summary() function for the model, the result can suggest that the binary factor "semester" is significant and predictive of teacher assistants evaluation scores with p-value 0.035 which is under the 5 percent significance level.

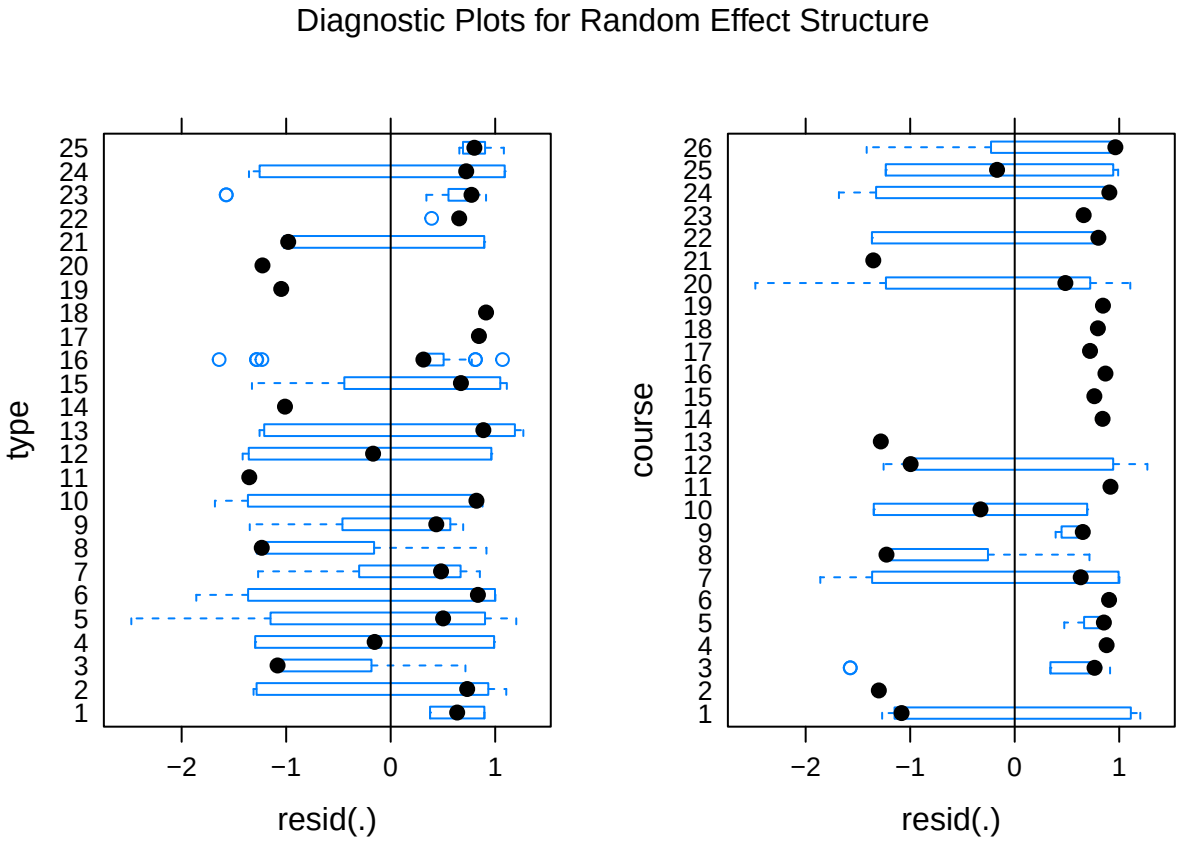
Both courses and type of instructors are random effect, and semesters, languages, and class sizes are fixed effects. Firstly, to ensure the application of data for further study, random effects will be checked using box plots and plots for standardized residuals versus fitted values to see if there are outliers. Then Levene's test is used to check the homoscedasticity of the random effects. Lastly, in order to further analyze the data, one-hot encode is implemented to create dummy variables by adding new columns for each category of course and type of instructor.

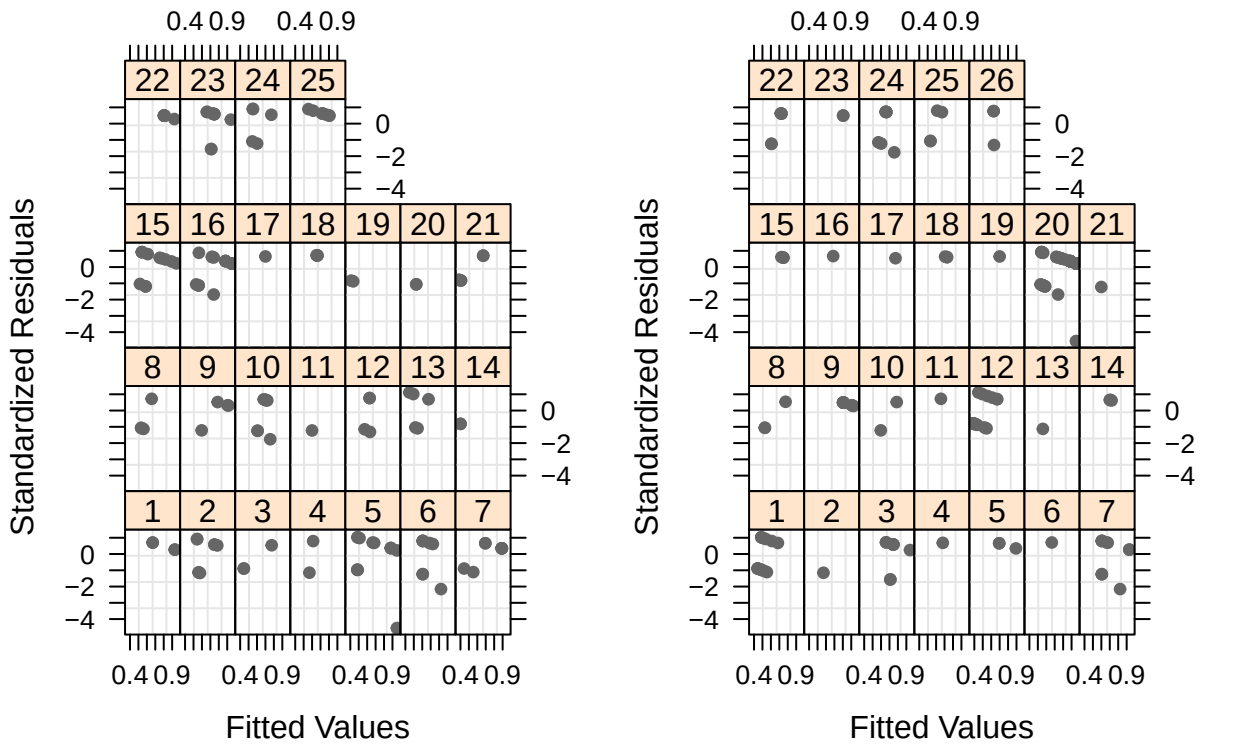
```
  FixedEffects  Estimate Std.Error      z       p
1  (intercept)  3.735616  1.152327  3.242 0.00119
2     Semester -1.823516  0.866426 -2.105 0.03532
3     Language -1.049599  0.621411 -1.689 0.09121
4         Size -0.009354  0.016493 -0.567 0.57059
```

Diagnostics

To check if there are outliers in the random effects, we want to evaluate the goodness of fit of the model and check if mathematical requirements and assumptions have been violated by focusing on the random effect structure.

By looking at the diagnostic plots and plots for standardized residuals versus fitted values, the outliers do not have unwarranted influence on the model. So there is no transformation needed for the random effects.





To check the homoscedasticity of the model, Levene's test is performed to check whether the assumption of variance homogeneity is violated. By looking at the ANOVA table presented below, since the p-values for type of instructors and type of courses are greater than 0.05, it can be concluded that the variance of the residuals is equal and the assumption of homoscedasticity is met.

```
Analysis of Variance Table

Response: fit.res2
           Df Sum Sq Mean Sq F value Pr(>F)
type       24 19.595 0.81644  1.0823 0.3764
course     21 11.551 0.55005  0.7291 0.7944
Residuals 105 79.211 0.75439
```

Encoding

For the categorical variables in random effects, courses and type of instructors, because there are many numbers of categories, and in order to visualize and analyze the data easier, we have implemented One-Hot encoding method to create dummy variables by creating new columns for each unique value of the categorical variable. Each of the new columns are binary with values 1 or 0, which mean yes and no for the category, depending on whether the value of the variable is equal to the unique value being encoded by the column.

```
  levels sizes semester language type.1
1      1    42        2        2      0
2      1    28        2        2      0
3      1    51        2        2      0
```

```
4       1     19          2          2       0
5       1     31          2          2       0
6       1     13          1          1       0
```

Implementation

This paper is aimed at exploring the factors that might affect the teacher assistants evaluation scores in general, and in order to further test if the factors measured in the given dataset can contribute to the the evaluation scores and can be employed to predict the results accurately, two algorithms were implemented to predict the evaluation score levels: Generalized Linear Mixed Model and Lasso Regression.

Generalized Linear Mixed Model

According to the theoretical formula for the General Linear Model with matrix notation, $Y = X\beta + \epsilon$, where Y, X, β, ϵ are matrices, and in the equation, X contains the fixed effects for the model, which are class sizes, semester, and language provided by the dataset. For random effects, courses and types, the model is expanded to include a matrix of random effect variables Z analogous to the X for the fixed effects and a vector of variance estimates α. Then the theoretical formula for the Generalized Linear Mixed Model becomes $Y = X\beta + Z\alpha + \epsilon$.

To better predict the model, we have used Leave-One-Out cross validation approach to separate the data to training data and test data. For $1 \leq k \leq n$, with n=151 in the dataset, we chose our k to be 51 folds because we have a relatively small data size and the prediction be biased if we use validation set method by splitting the data. We trained k-1 of the folds using for loop and used the generalized linear mixed model to get a fitted training response variable. Then, we used the model from the training data and the explanatory variables in the test data to predict the response for the test data.

The model used in training data is applied to the test data set, and the response variables, teacher assistant evaluation scores, are predicted. The R-squared, MSE, RMSE, and MAE matrix is listed below to evaluate the prediction error rate and model performance. R-squared (Coefficient of determination) is the coefficient of how well the values fit compared to the original values, and the higher the value is, the better the model is. MSE (Mean Squared Error) is the difference between the original and predicted values. RMSE (Root Mean Squared Error) is the error rate by the square root of mean squared error. MAE (Mean Absolute Error) is the difference between the original and predicted values extracted by averaged the absolute difference over the dataset.

```
        R2       MSE      RMSE       MAE
1 0.986501 0.2146231 0.4632743 0.3906879
```

Lasso Regularizations

Lasso regression is utilized to predict the teaching assistant evaluation scores, just as the generalized linear mixed model did above. In order to ensure that optimal lambda value was used in computation, cross validation on the dataset was carried out. Subsequently, with the optimal lambda value, Lasso regression was performed, producing the result in the format of R-squared value and RMSE, which are values explained above.

Regularization was introduced in order to deal with issues that maximum likelihood estimators had. They tend to overfit when the number of parameters is large, be unstable when variables are highly correlated, and ignore prior information about the dataset. Specifically, a regularization term is added to minimizing negative log-likelihood part of the estimator computation. In this study, the Lasso model was selected specifically for the added benefit of reduction in number of features.

$argmin(\frac{1}{N}||y - X\beta||_2^2 + \lambda||\beta||_1)$

Lambda is the regularization parameter which controls how much regularization is applied to the computation, and $||\beta||_1$ indicated that Lasso Regression is used. It is used when there is a large number of parameters but only a certain number of them are actually relevant; less important variables have their coefficients reduced to zero, causing them to be removed from the computation. With the number of features increased considerably as all factor variables were accounted for, it was appropriate that an algorithm capable of carrying out a type of feature selection is selected.

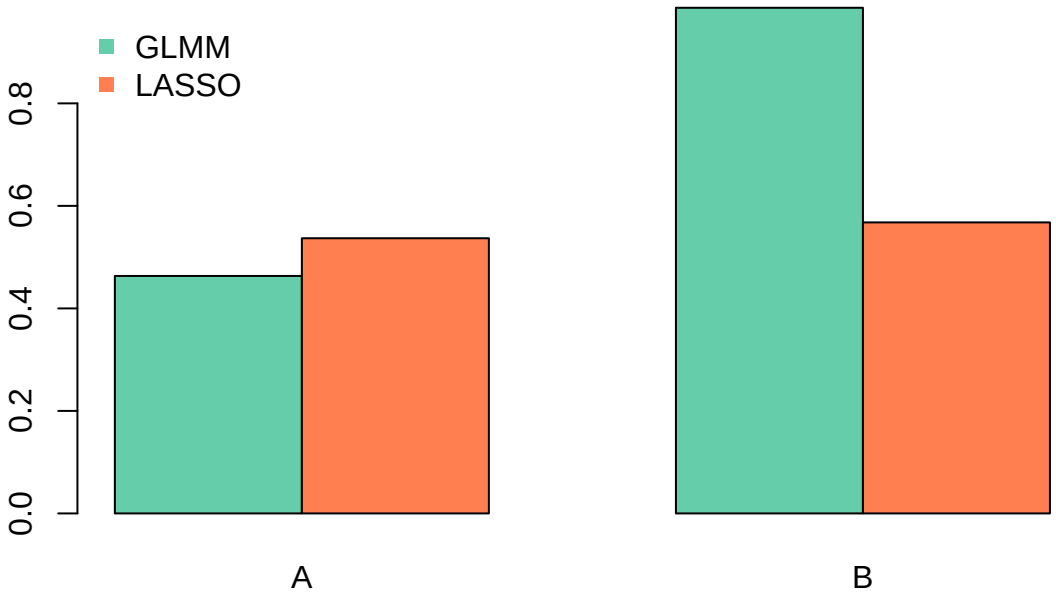
Features are selected in this manner, but in order to ensure that the best model is employed, lambda has to be determined as well. Model selection is finally complete when the lambda value that produces the best result is determined, and this is done through cross validation. In this project, 10-fold cross validation was performed to obtain the best lambda value. As a rule of thumb, the number of fold was chosen to be 10 as 10-fold tends to produce the best result. As a result, the Lasso regression model with the best lambda value was generated, yielding R-squared value and RMSE which allowed comparison with the other algorithm.

```
## Warning in model.frame.default(Terms, newdata, na.action = na.action, xlev =
## object$lvls): variable 'levels' is not a factor

## Warning in model.frame.default(Terms, newdata, na.action = na.action, xlev =
## object$lvls): variable 'levels' is not a factor

##    Data      RMSE  RSquared
## 1 Train 0.6669695 0.3345656
## 2  Test 0.5368255 0.5677276
```

Conclusion

In this study, we were trying to determine whether the factors in the dataset can effectively provide information directly related to assistant evaluation scores. We implemented two algorithms, namely generalized linear mixed model and the lasso regression, and separated the data into training and test sets to test if the given information can be used to accurately predict the scores. As a form of measurement, we calculated the R-squared and RMSE and found out that the predicted test results for the data using generalized linear mixed model is closer to the true value with a R-squared value 0.9865 than that for lasso regularization with the R-squared value of 0.5677. The RMSE, which is the root mean squared error between the fitted teaching assistant evaluation levels from test data and the true teaching assistant evaluation levels, is smaller for generalized linear mixed model than for lasso regression, which are 0.4633 and 0.5368 respectively. The figure below is the visualization of the results, where "A" represents the RMSE and "B" represents the R-Squared, and the value in blue color is from the generalized linear mixed model whereas the value in orange is from the lasso regression model. Therefore, the results of the models show that generalized linear mixed model predicts a better score for the teaching assistant evaluations given the factors provided.

Moreover, after running the regression and performing the ANOVA table, the result shows that only the binary factor "semester" is significant in determining the evaluation scores. Therefore, further study should be conducted to lead to better predictions of the evaluation scores, in order for the data to provide useful and effective feedback to the teachers facing evaluations.



Discussion

In this study, the data is pre-processed by features like selection and ranking. Then, We have used two different algorithms, which are Generalized Linear Mixed Model and Ridge Regularization, to predict the evaluation scores based on the explanatory variables. There are limitations in this study and dataset as well, the information of the categories for courses and type of instructors are not provided by the dataset, and the dataset is relatively small in size for this study to implement cross validation and prediction, as bias might be included the results. Moreover, as only the semester is found to be significant after fitting the model, we would consider add some other factors to the test in the future so the results will be more representative. In order to make improvements on this topic of study, a larger dataset with more informative variables is required. Some variables like gender, salary, fields of study, and professors they work with are considered for teaching assistant evaluations. With the enhancement of data, it would be possible to develop and explore better models and pattern of the variables that determine the evaluations of teaching assistants. This would help them improve their performance more effectively.

Reference

[1] “Uci machine learning repository: Teaching assistant evaluation data set,” “https://arch-ove.ice.uci.edu/ml/datasets/teaching+assistant+evaluation”, [Online: accessed 20 November, 2020]

Contribution

This study is conducted by Zhiyi Chen and Ho Gun Kang to fulfill the course project in Stat 479. The dataset is chosen by Zhiyi Chen and Ho Gun Kang, and the topic is decided by Zhiyi Chen and Ho Gun Kang. Zhiyi mainly contributed to both the writing and coding part in introduction, EDA, and GLMM. Ho Gun Kang mainly contributed to the writing and coding part for the Lasso, Conclusion and Discussion. The writing part is edited by both Zhiyi Chen and Ho Gun Kang.

Appendix

Packages

```
library(tidyr)
library(tidyverse)
library(ggplot2)
library(grid)
library(gridExtra)
library(MASS)
library(sjstats)
library(lme4)
library(lattice)
library(caret)
library(glmnet)
```

Setups

```
setwd("/Users/chenzhiyi/Documents/STAT 479")
ta<-read.table("ta")
ta<-ta %>%
  separate(V1, c("Language", "Type", "Course", "Semester", "Size", "Attribute"), ",")
ta$Language<-as.factor(ta$Language)
ta$Type<-as.factor(ta$Type)
ta$Course<-as.factor(ta$Course)
ta$Semester<-as.factor(ta$Semester)
ta$Attribute<-as.factor(ta$Attribute)
```

Exploratory Data Analysis

Visualization

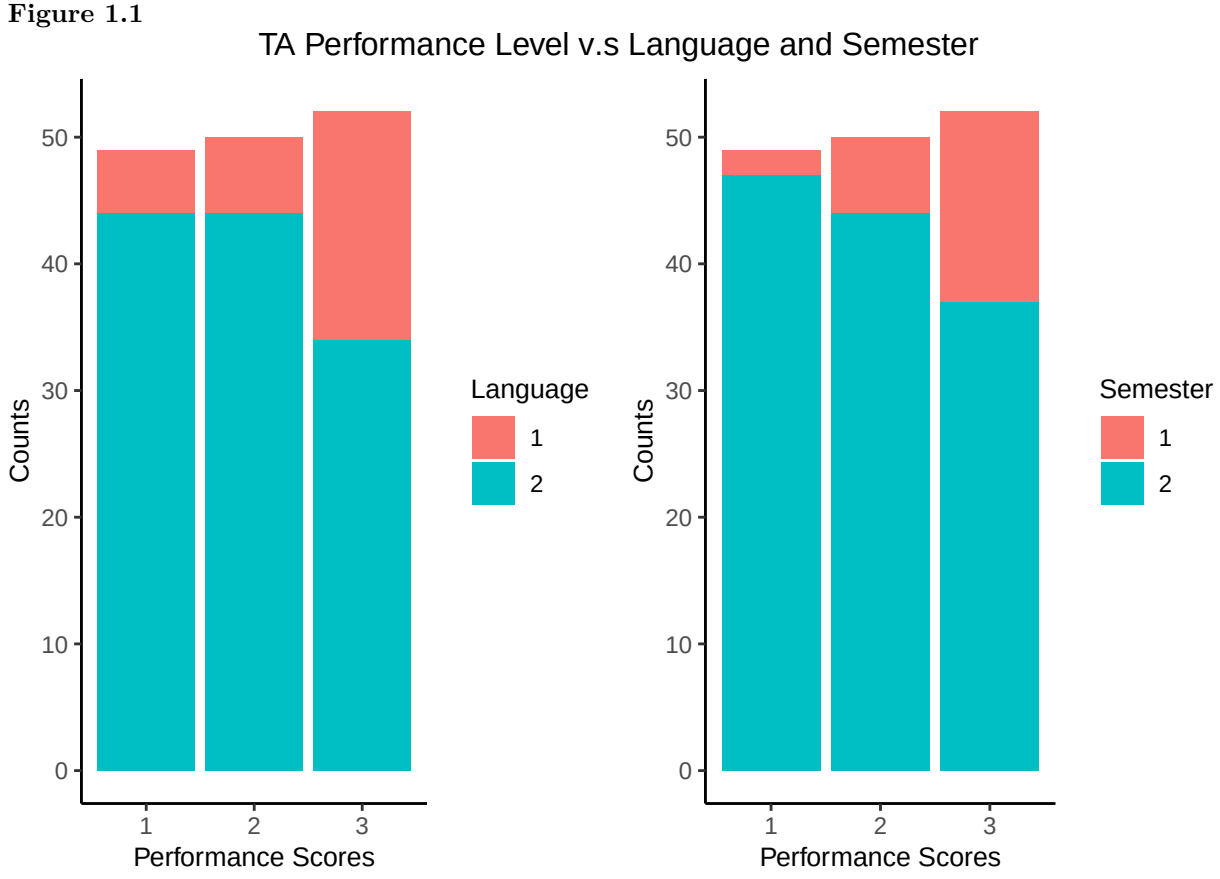
```

head(ta)
```

Data Table

```
##   Language Type Course Semester Size Attribute
## 1        1   23      3        1   19         3
## 2        2   15      3        1   17         3
## 3        1   23      3        2   49         3
## 4        1    5      2        2   33         3
## 5        2    7     11        2   55         3
## 6        2   23      3        1   20         3
```

```
p1<- ggplot(data=ta, aes(x=Attribute, fill=Language))+
  geom_bar() +
  labs(x="Performance Scores", y="Counts")+
  theme_classic()+
  theme(plot.title = element_text(size = 10),
        axis.title = element_text(size = 10),
        legend.title = element_text(size=10))

p2<- ggplot(data=ta, aes(x=Attribute, fill=Semester))+
  geom_bar() +
  labs(x="Performance Scores", y="Counts")+
  theme_classic()+
  theme(plot.title = element_text(size = 10),
        axis.title = element_text(size = 10),
        legend.title = element_text(size=10))
grid.arrange(p1, p2, ncol=2, nrow=1, top=textGrob("TA Performance Level v.s Language and Semester"))
```

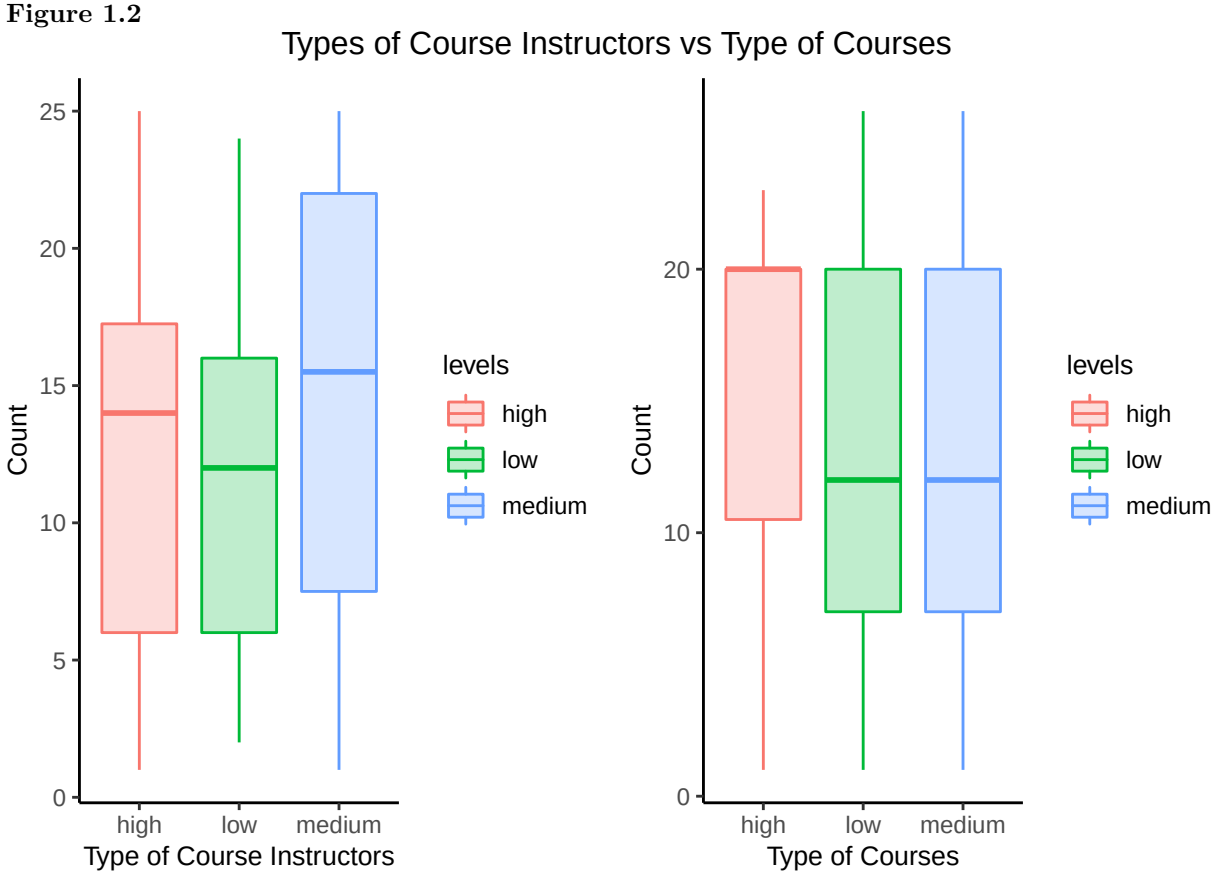
Figure 1.1




```
type1<-c(ta$Type[which(ta$Attribute=="1")])
type2<-c(ta$Type[which(ta$Attribute=="2")])
type3<-c(ta$Type[which(ta$Attribute=="3")])
```

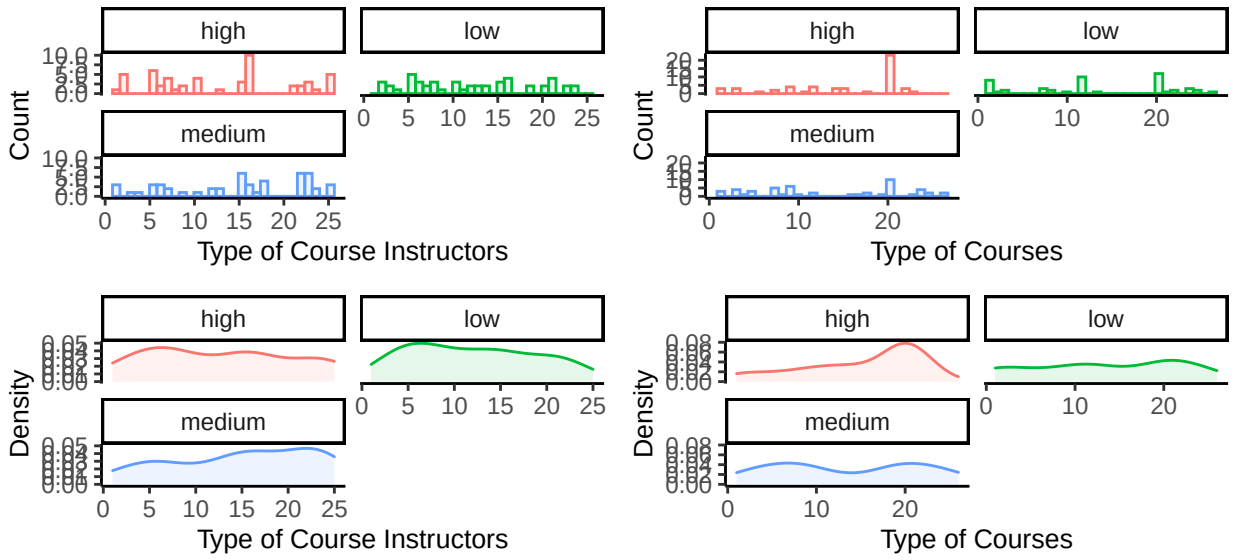
```r
low<-length(which(ta$Attribute=="1"))
medium<-length(which(ta$Attribute=="2"))
high<-length(which(ta$Attribute=="3"))
course1<-c(ta$Course[which(ta$Attribute=="1")])
course2<-c(ta$Course[which(ta$Attribute=="2")])
course3<-c(ta$Course[which(ta$Attribute=="3")])
size1<-c(ta$Size[which(ta$Attribute=="1")])
size2<-c(ta$Size[which(ta$Attribute=="2")])
size3<-c(ta$Size[which(ta$Attribute=="3")])
df1 <-data.frame(
  levels = c(rep("low", low), rep("medium", medium), rep("high", high)),
  sizes = c(size1, size2, size3),
  type = c(type1, type2, type3),
  course = c(course1, course2, course3),
  count = c(rep(low, low), rep(medium,medium), rep(high,high))
)

p1<- ggplot(df1, aes(x=levels, y=type))+geom_boxplot(aes(color=levels, fill=levels), alpha=0.25, outlie
  labs(x="Type of Course Instructors",
       y="Count")+
  theme_classic()+
  theme(plot.title = element_text(size=10),
        axis.title = element_text(size=10),
        legend.title = element_text(size=10))

p2<- ggplot(df1, aes(x=levels, y=course))+geom_boxplot(aes(color=levels, fill=levels), alpha=0.25, outli
  labs(x="Type of Courses",
       y="Count")+
  theme_classic()+
  theme(plot.title = element_text(size=10),
        axis.title = element_text(size=10),
        legend.title = element_text(size=10))

grid.arrange(p1, p2, ncol=2, nrow=1, top=textGrob("Types of Course Instructors vs Type of Courses"))
```

Figure 1.2



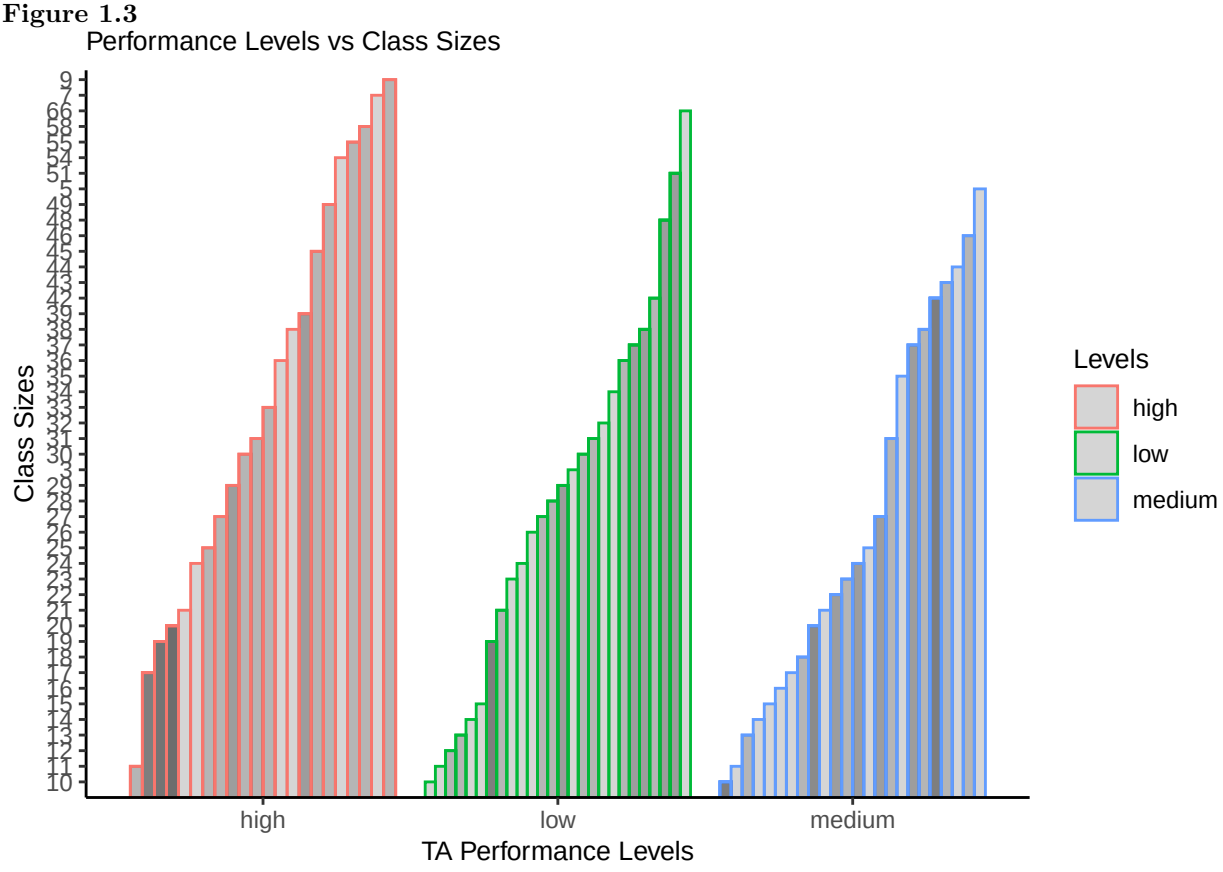

```
p3<-ggplot(df1, aes(x=type))+
  geom_histogram(aes(color=levels, fill=levels), bins=30, boundary=0, alpha=0.1)+
  facet_wrap(levels~., nrow=2)+
  labs(x="Type of Course Instructors",
       y="Count")+
  theme_classic()+
  theme(plot.title = element_text(size=10),
        axis.title = element_text(size=10),
        legend.title = element_text(size=10))+
  theme(legend.position="none")

p4<-ggplot(df1, aes(x=course))+
  geom_histogram(aes(color=levels, fill=levels), bins=30, boundary=0, alpha=0.1)+
  facet_wrap(levels~., nrow=2)+
  labs(x="Type of Courses",
       y="Count")+
  theme_classic()+
  theme(plot.title = element_text(size=10),
        axis.title = element_text(size=10),
        legend.title = element_text(size=10))+
  theme(legend.position="none")

p5<-ggplot(df1, aes(x=type))+
  geom_density(aes(color=levels, fill=levels), alpha=0.1)+
  facet_wrap(levels~., nrow=2)+
```

```
  labs(x="Type of Course Instructors",
       y="Density")+
  theme_classic()+
  theme(plot.title = element_text(size=10),
        axis.title = element_text(size=10),
        legend.title = element_text(size=10))+
  theme(legend.position="none")

p6<-ggplot(df1, aes(x=course))+
  geom_density(aes(color=levels, fill=levels), alpha=0.1)+
  facet_wrap(levels~., nrow=2)+
  labs(x="Type of Courses",
       y="Density")+
  theme_classic()+
  theme(plot.title = element_text(size=10),
        axis.title = element_text(size=10),
        legend.title = element_text(size=10))+
  theme(legend.position="none")

grid.arrange(p3, p4, p5, p6, ncol=2, nrow=3)
```




```
ggplot(df1, aes(x=levels, y=sizes)) +
  geom_bar(aes(color=levels), position="dodge", stat="identity",alpha=0.25)+
  labs(x="TA Performance Levels",
       y="Class Sizes",
       title="Performance Levels vs Class Sizes",
       fill="Levels",
       color="Levels")+
  theme_classic()+
  theme(plot.title = element_text(size=10),
        axis.title = element_text(size=10),
        legend.title = element_text(size=10))
```

Figure 1.3



Data-Processing

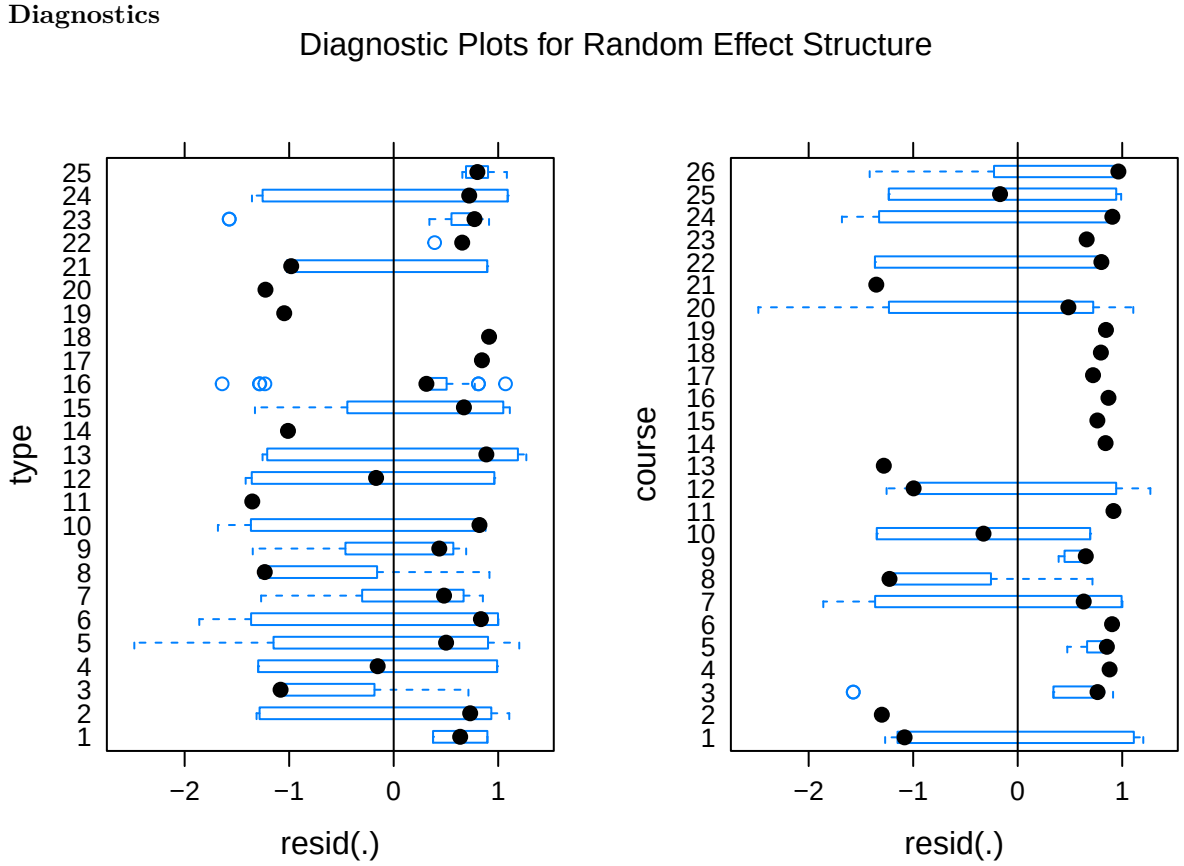
```
semester1<-c(ta$Semester[which(ta$Attribute=="1")])
semester2<-c(ta$Semester[which(ta$Attribute=="2")])
semester3<-c(ta$Semester[which(ta$Attribute=="3")])
language1<-c(ta$Language[which(ta$Attribute=="1")])
language2<-c(ta$Language[which(ta$Attribute=="2")])
language3<-c(ta$Language[which(ta$Attribute=="3")])
df2 <-data.frame(
  levels = as.factor(c(rep(1, low), rep(2, medium), rep(3, high))),
  sizes = as.numeric(c(size1, size2, size3)),
  type = as.factor(c(type1, type2, type3)),
  course = as.factor(c(course1, course2, course3)),
  semester=as.factor(c(semester1, semester2, semester3)),
  language=as.factor(c(language1, language2, language3)),
  count = c(rep(low, low), rep(medium,medium), rep(high,high))
)

fit<- glmer(levels~semester+language+sizes+(1|type)+(1|course), family = binomial("logit"), data=df2)
summary(fit)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
```

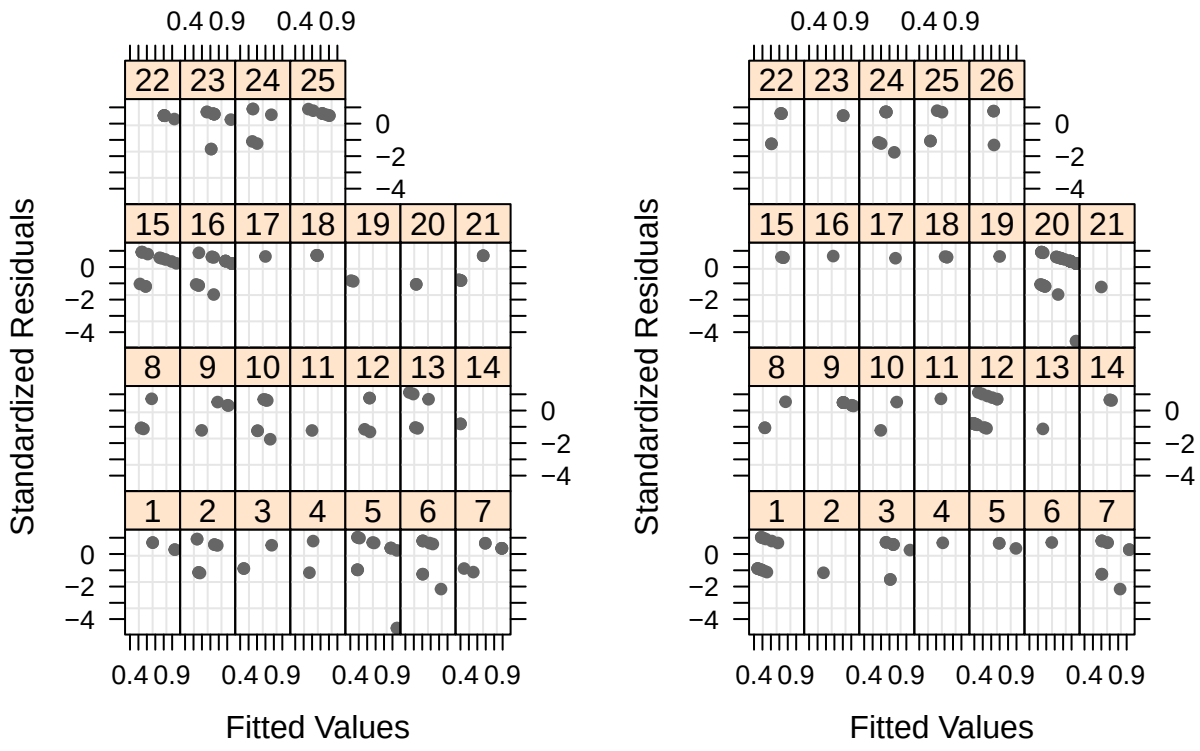
```
##  Family: binomial  ( logit )
## Formula: levels ~ semester + language + sizes + (1 | type) + (1 | course)
##    Data: df2
##
##      AIC      BIC   logLik deviance df.resid
##    188.5    206.6    -88.2    176.5      145
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5546 -0.9301  0.4815  0.6697  1.1124
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  course (Intercept) 0.4135   0.6431
##  type   (Intercept) 0.1953   0.4419
## Number of obs: 151, groups:  course, 26; type, 25
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.735616   1.152327   3.242  0.00119 **
## semester2   -1.823516   0.866426  -2.105  0.03532 *
## language2   -1.049599   0.621411  -1.689  0.09121 .
## sizes       -0.009354   0.016493  -0.567  0.57059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) smstr2 langg2
## semester2 -0.659
## language2 -0.585 -0.003
## sizes     -0.371 -0.189  0.241
```

```
pr1<-plot(fit, type~resid(.), abline=0)
pr2<-plot(fit, course~resid(.), abline=0)
grid.arrange(pr1, pr2, ncol=2, nrow=1, top=textGrob("Diagnostic Plots for Random Effect Structure"))
```

Diagnostics




```
pr3<-plot(fit, resid(., type="pearson")~fitted(.)|type, id=0.05, adj=-0.3, pch=20, col="gray40", ylab =
pr4<-plot(fit, resid(., type="pearson")~fitted(.)|course, id=0.05, adj=-0.3, pch=20, col="gray40", ylab
grid.arrange(pr3, pr4, ncol=2, nrow=1, top=textGrob("Diagnostic Plots for Random Effect Structure"))
```




```
df2$fit.res<-residuals(fit)
df2$abs.fit.res<-abs(df2$fit.res)
df2$fit.res2<-df2$abs.fit.res^2
Levene.fit<-lm(fit.res2~type+course, data=df2)
anova(Levene.fit)
```

```
## Analysis of Variance Table
##
## Response: fit.res2
##             Df Sum Sq Mean Sq F value Pr(>F)
## type        24 19.595 0.81644  1.0823 0.3764
## course      21 11.551 0.55005  0.7291 0.7944
## Residuals 105 79.211 0.75439
```

```
df3<-data.frame(
  levels = as.factor(c(rep(1, low), rep(2, medium), rep(3, high))),
  sizes = as.numeric(c(size1, size2, size3)),
  type = as.factor(c(type1, type2, type3)),
  course = as.factor(c(course1, course2, course3)),
  semester=as.factor(c(semester1, semester2, semester3)),
  language=as.factor(c(language1, language2, language3)),
  count = c(rep(low, low), rep(medium,medium), rep(high,high))
)
```

```
df4<-df3
dv<-dummyVars("~type + course",data=df4)
df4<-data.frame(predict(dv, newdata=df3))
df5<-cbind(df3$levels,df3$sizes,df3$semester,df3$language, df4)
df5<-df5%>%
  rename(levels=`df3$levels`,
         sizes=`df3$sizes`,
         semester=`df3$semester`,
         language=`df3$language`)
head(df5[, 1:5])
```

Encoding

```
##   levels sizes semester language type.1
## 1      1    42        2        2      0
## 2      1    28        2        2      0
## 3      1    51        2        2      0
## 4      1    19        2        2      0
## 5      1    31        2        2      0
## 6      1    13        1        1      0
```

Implementation

```
set.seed(479)
df3<- df3[sample(nrow(df3)),]
folds <- cut(seq(1, nrow(df3)), breaks=51, labels=FALSE)
for (i in 1:50){
  testIndexes <- which(folds==i, arr.ind = TRUE)
  testData <- df3[testIndexes, ]
  trainData <- df3[-testIndexes, ]
}
model_train<-glmer(levels~semester+language+sizes+(1|type)+(1|course), family = binomial("logit"), data=
prediction <- predict(model_train, testData)
data.frame(R2=R2(prediction, as.numeric(testData$levels)),
           MSE = (RMSE(prediction, as.numeric(testData$levels)))^2,
           RMSE = RMSE(prediction, as.numeric(testData$levels)),
           MAE = MAE(prediction, as.numeric(testData$levels)))
```

Generalized Linear Mixed Model

```
##         R2       MSE      RMSE       MAE
## 1 0.986501 0.2146231 0.4632743 0.3906879
```

```
#create dataset full of dummy variables to be used in Lasso regression
cols_reg <- c("levels", "sizes", "type", "course", "semester", "language")
dummies <- dummyVars(levels ~., data=df3[,cols_reg])
train_dummies=predict(dummies, newdata=trainData)
```

Lasso Regularization

```
## Warning in model.frame.default(Terms, newdata, na.action = na.action, xlev =
## object$lvls): variable 'levels' is not a factor
```

```
test_dummies=predict(dummies, newdata=testData)
```

```
## Warning in model.frame.default(Terms, newdata, na.action = na.action, xlev =
## object$lvls): variable 'levels' is not a factor
```

```
#convert the format of data
x_train = as.matrix(train_dummies)
y_train = as.numeric(trainData$levels)

x_test = as.matrix(test_dummies)
y_test = as.numeric(testData$levels)
#a function that calculates and presents R square value and Root Mean Square Error
eval_results <- function(true, predicted, df) {
  SSE <- sum((predicted - true)^2)
  SST <- sum((true - mean(true))^2)
  R_square <- 1 - SSE / SST
  RMSE = sqrt(SSE/nrow(df))

data.frame(
  RMSE = RMSE,
  Rsquare = R_square
)
}

#use cross 10-fold cross validation for hyperparameter tuning and get the best lambda value
set.seed(479)
lambdas=10^seq(2,-3,by=-0.1)
x=as.matrix(train_dummies)
lasso_reg <- cv.glmnet(x, y_train, alpha = 1, lambda = lambdas, standardize = TRUE, nfolds = 10)

lambda_best <- lasso_reg$lambda.min

#implement Lasso Regression with the best lambda value

lasso_model <- glmnet(x_train, y_train, alpha = 1, lambda = lambda_best, standardize = TRUE)

predictions_train <- predict(lasso_model, s = lambda_best, newx = x_train)
tr<-eval_results(y_train, predictions_train, trainData)

predictions_test <- predict(lasso_model, s = lambda_best, newx = x_test)
te<-eval_results(y_test, predictions_test, testData)

data.frame(
  Data=c("Train", "Test"),
  RMSE=c(0.6669695, 0.5368255),
  RSquared=c(0.3345656, 0.5677276)
)
```

```
##    Data      RMSE  RSquared
## 1 Train 0.6669695 0.3345656
## 2  Test 0.5368255 0.5677276
```

Conclusion

```
barplot(matrix(c(0.4632743, 0.5368255, 0.986501, 0.5677276), nr=2), beside=TRUE,
        col=c("aquamarine3", "coral"),
        names.arg = LETTERS[1:2])
legend("topleft", c("GLMM", "LASSO"), pch=15, col=c("aquamarine3", "coral"), bty="n")
```