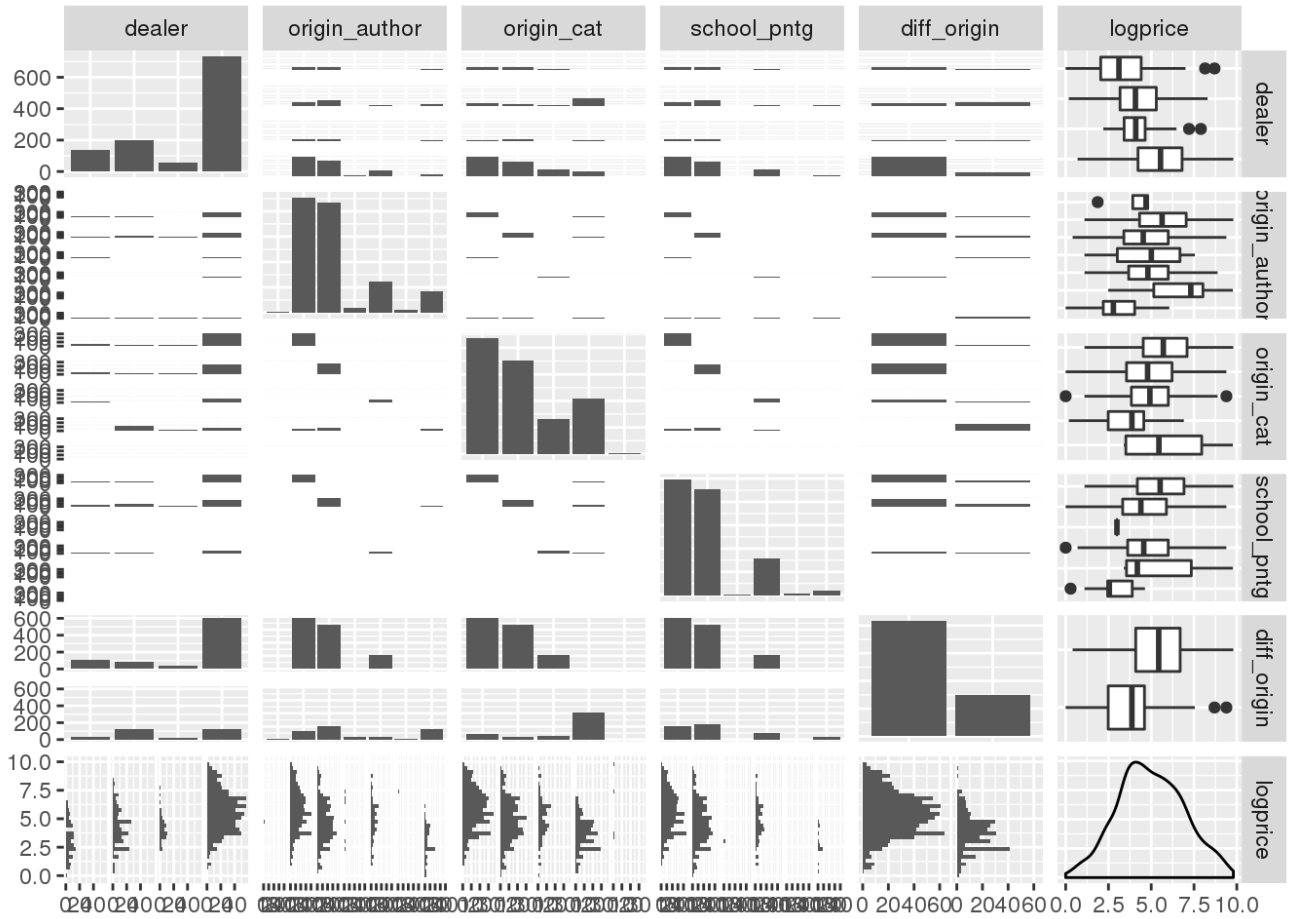
Part-II-Writeup

Part II: Write Up

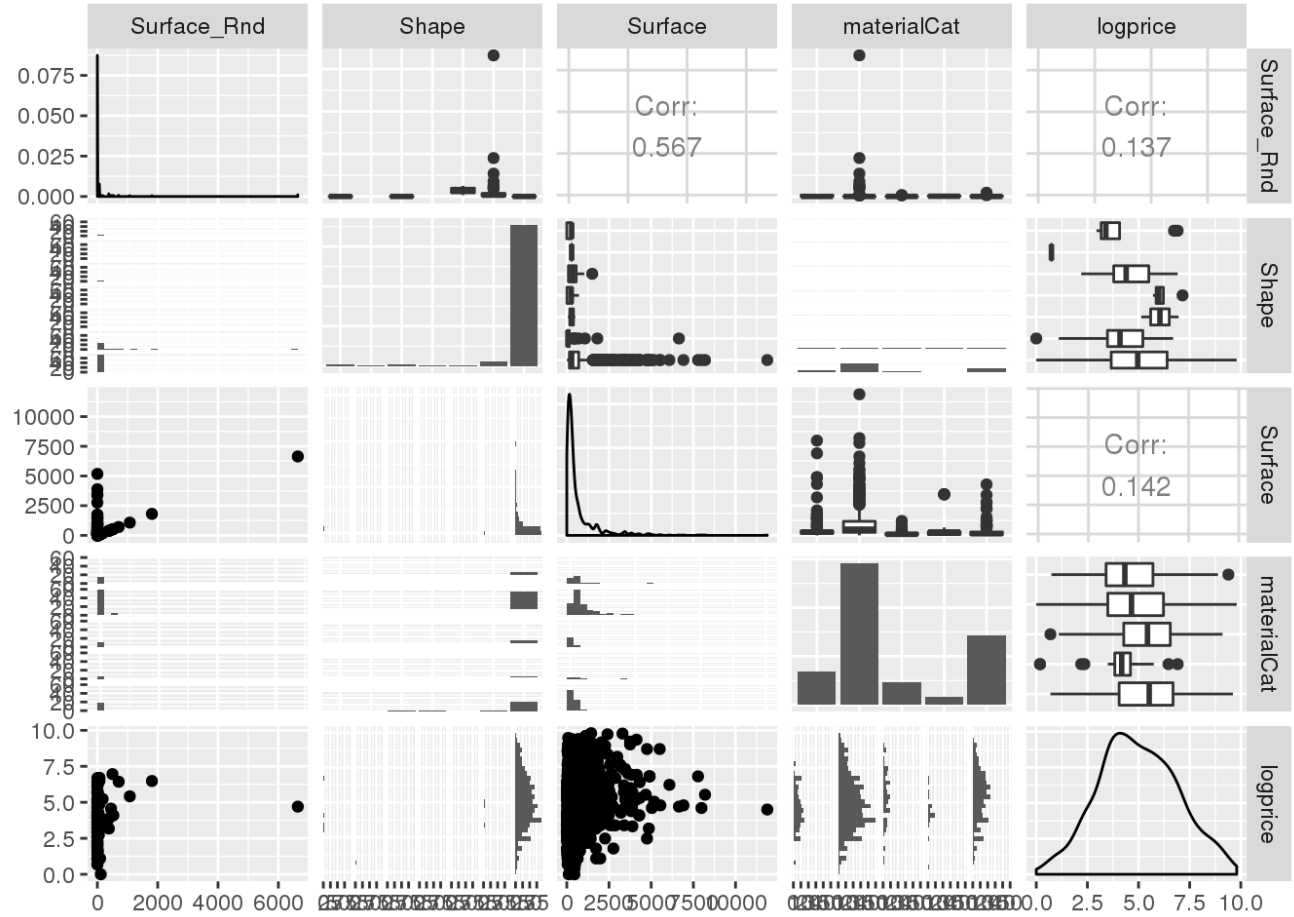
1. Introduction
   It's hard to judge artwork via standardized objective criteria, but everybody has different aesthetic standards – these two are the main reasons why it is very challenging to determine the price of a piece of artwork based on outside factors. Nowadays, historians are very interested in how paintings are priced historically. Understand how people price artwork and different factors will enable us to understand how aesthetics change over time. Valuing artwork based on outside objective factors is the only thing that we and statistics can do in current stage.
   While intrinsic aesthetic values influence the price of artwork, we also believe that the price might well depend on a lot of objective factors, such as the artists, styles of painting, dealers, buyers, etc.
   In this project, we will help art historian understand what factors drive prices of painting in the past and decide whether paintings might be overvalued or undervalued. Here we have the original data with 59 possible variables and 1131 observations. We are devoted to finding the best model that can predict the actual sale price of paintings, and in order to accomplish that we explore a variety of models including OLS,boosting, Lasso, Random Forest, BMA, BART. We look at the prediction performance and accuracy of each model, and select the best one among them to become our final model.
2. Exploratory data analysis :

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



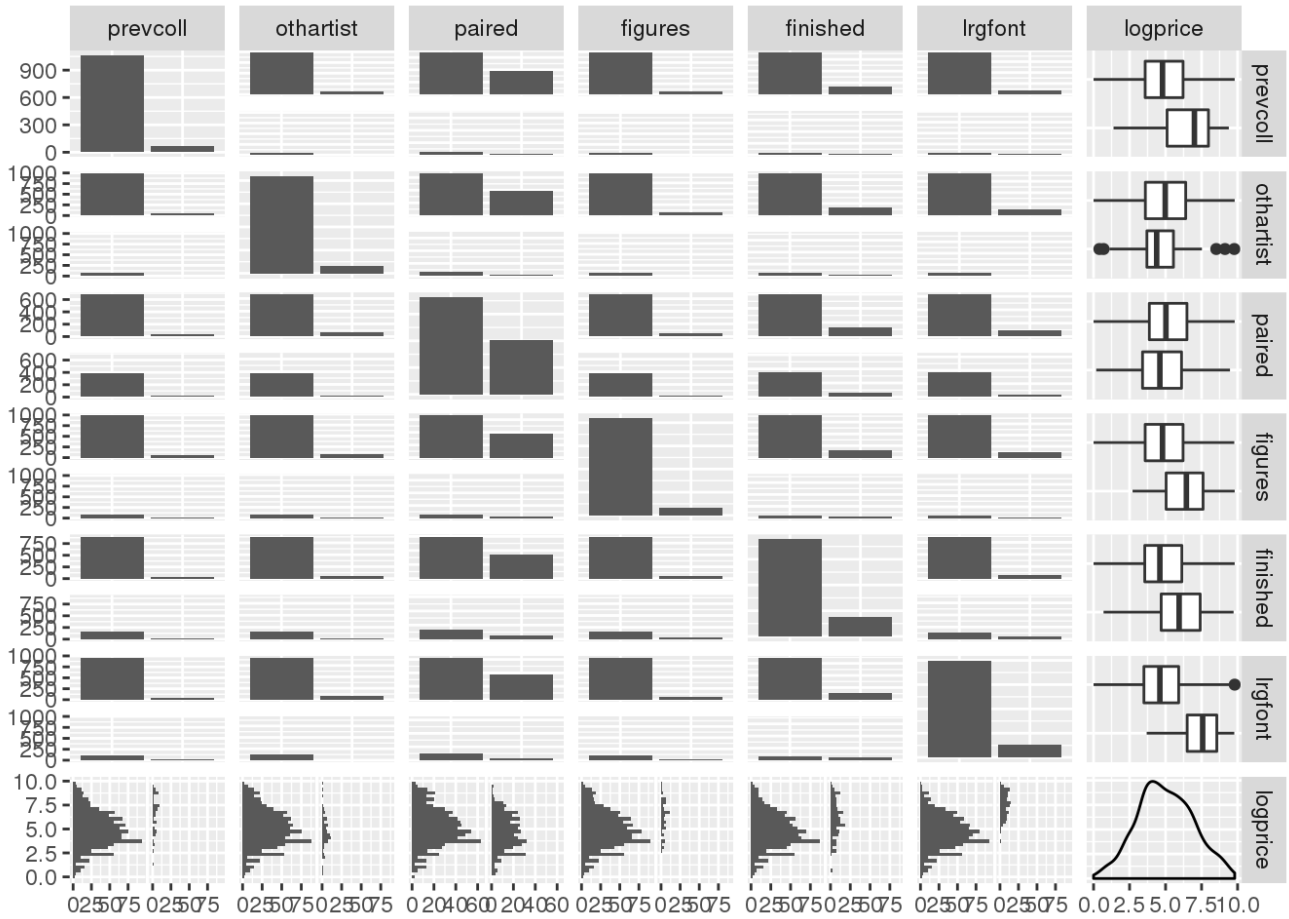
We first use ggpairs to see the relationship between different variables and log price. The above graph shows us the distribution of categorical variables and their relationship with logprice. The graph in last row is a feature that we focus on, as the last row represents the relationship between log price and other variables. We want to find the variable that have distinct different mean of logprice in each level. For example, it is very obvious that different type of dealer(four levels) has data gathered into different center. The same with Origin_author. As a result, we include those into our model selection process

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

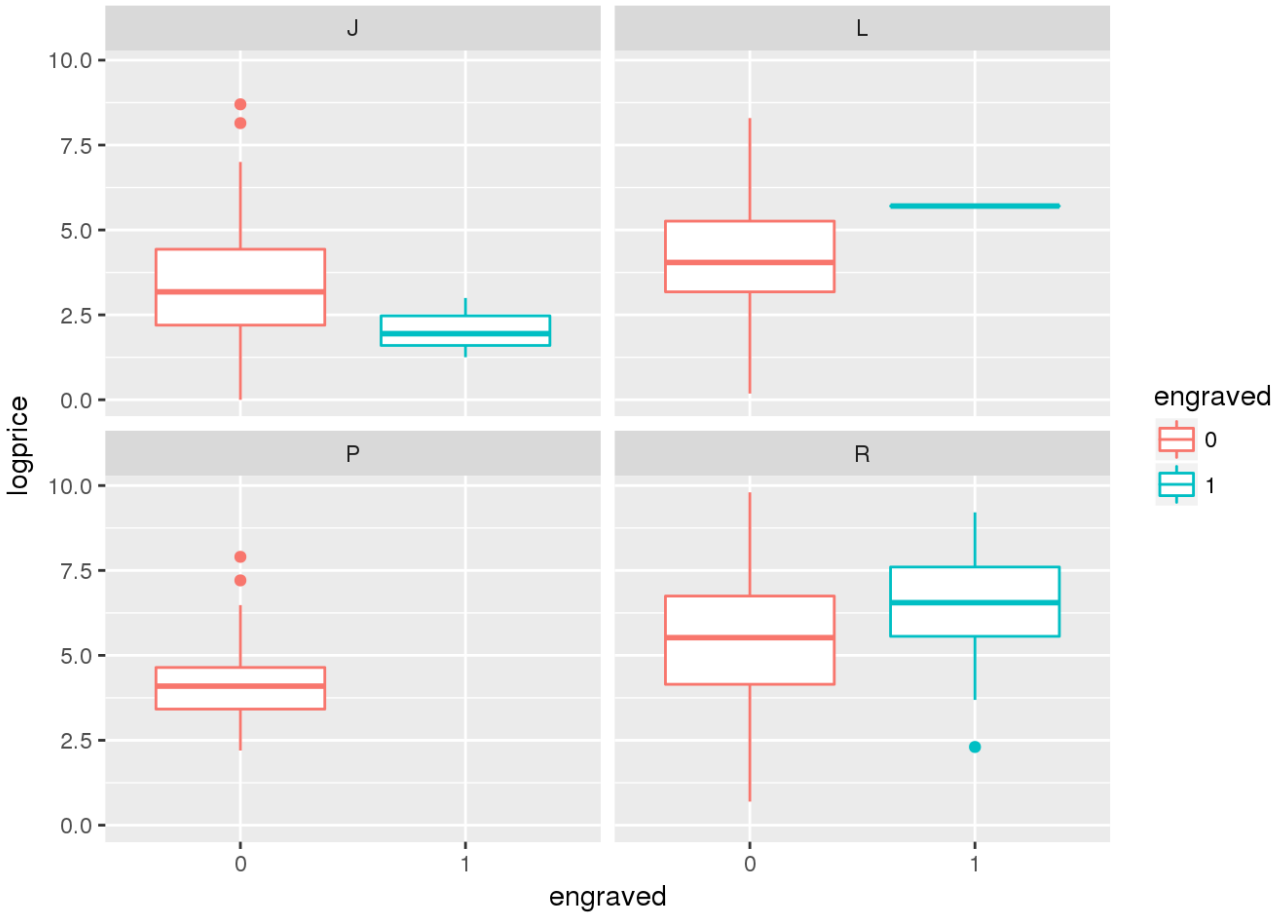


This ggpair graph includes a continuous variable, Surface. Correlation index is the feature that we look at. For example, the correlation of log price and surface here is relative large, so we propose that surface may is a good predictor in predicting price, so we include it into our model selection process. We can also see a correlated trend between Surface and log price in the last row. The log price is significant different for different levels of materialCat, so we also include that variable into our initial model selection.

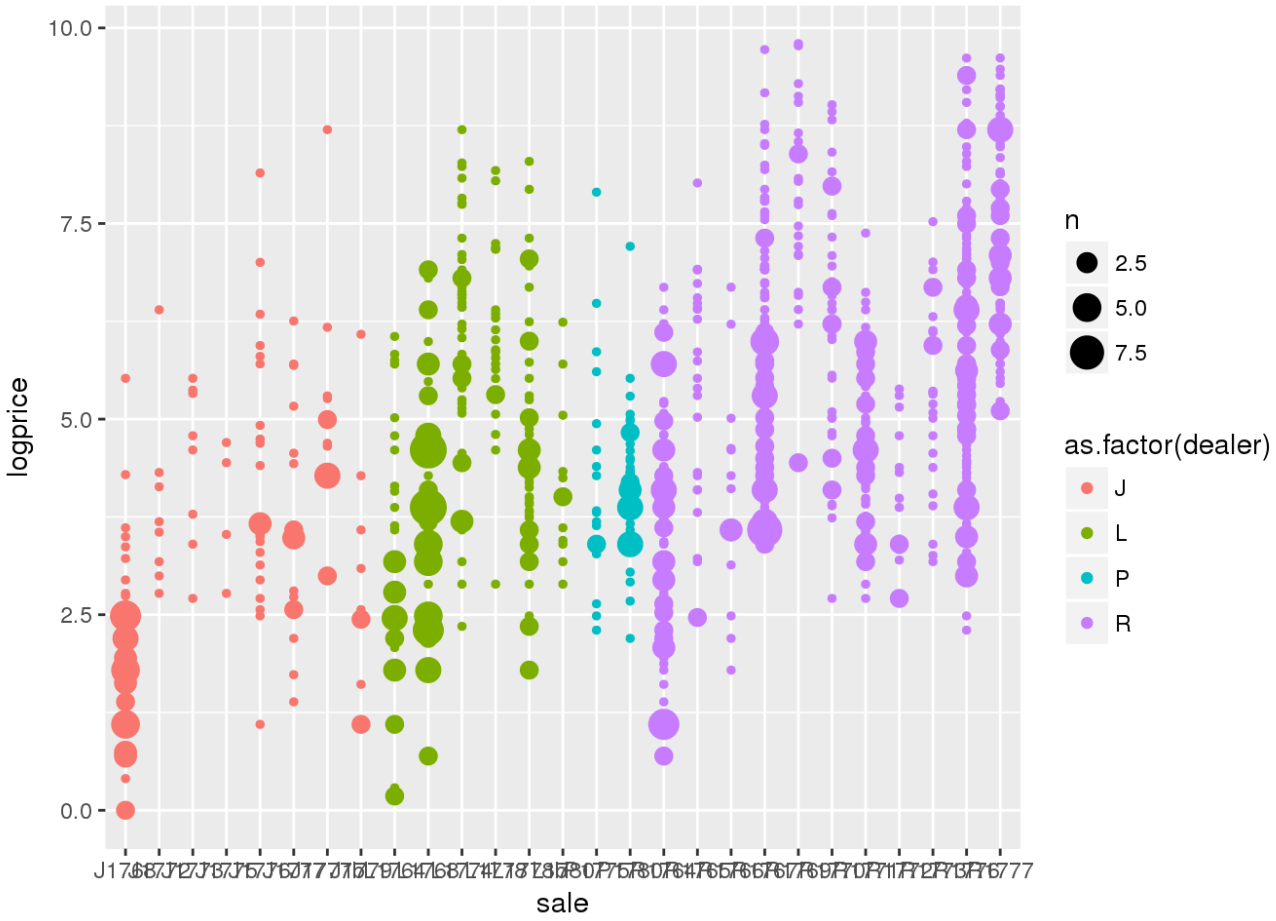
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



This ggpairs graph helps us to explore dummy variables. The last row also represents important features. For example, we can identify that the log price is centered at a place very different for having figures versus not having figure. lrgfont also shows the significant difference between two groups. As a result, we also include figures and lrgfont in the model.



We use ggplot with different facets to indentify interaction. The boxplots here show that the mean and quantile of log price for painting that is engraved or not engraved is different for different dealers. As a result, we speculate that there is a interaction between this two varaibles. So we add it into our model selection process.



New findings: We use ggplot to plot the relationship between price and sale in this part and find that different sale number tendst to correspond to different price of painting. As a result, we speculate that sale is a very important predictor.

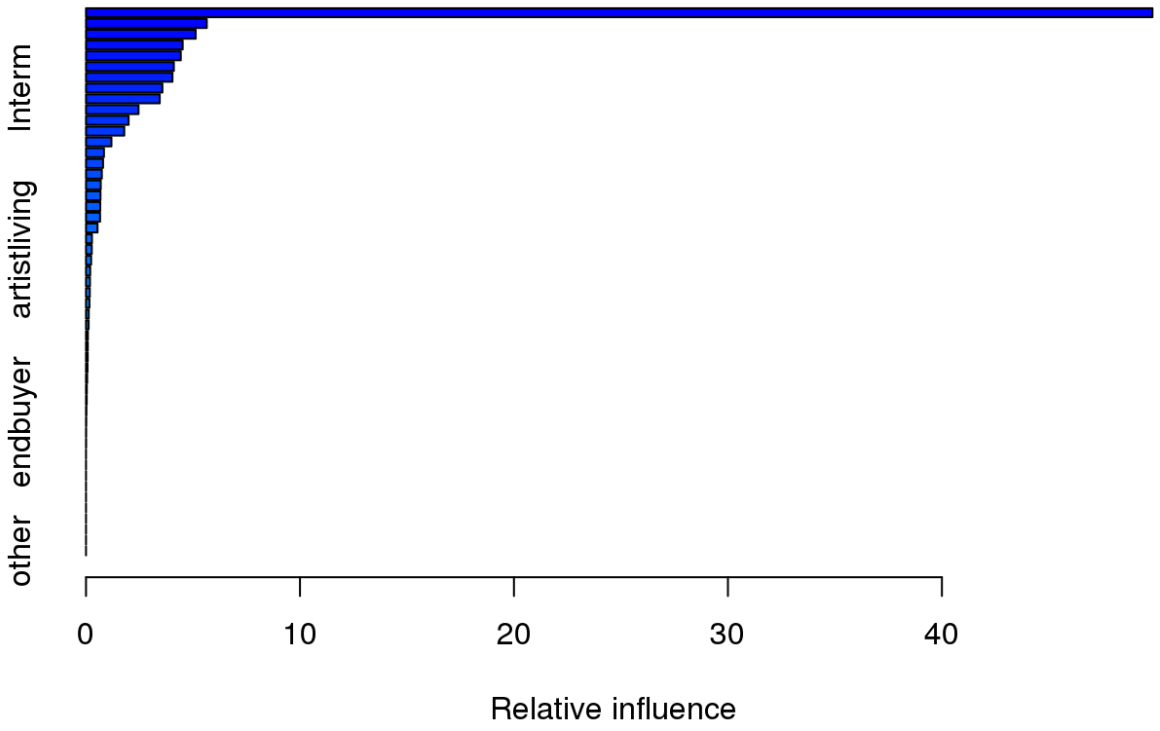
3. Discuss performance based on leader board results and suggested refinements.
   The model we used is the following: model1=lm(formula = logprice ~ year + dealer + origin_cat + lrgfont +Surface + diff_origin + engraved + endbuyer + lands_sc +finished + paired + discauth + type_intermed + origin_cat:arch +dealer:engraved + dealer:paired + shape_recode, data = paintings_train)

The bias of this model is 184 and the test RMSE is 1388. The RMSE on training set for this model is around 1500, which, together with the testing results, is a possible indication that the model does not fit the train dataset very well, but overfit the testdata to some extent. Comparing to other results on leaderboard, our bias is comparitively lower, but the RMSE is a bit higher than the average. Our coverage does a relatively good job though.
The relatively-high test RMSE may be due to that some of the variables in our model are highly-variable and not very significant, such origin_cat:arch; or it may be due to that we still haven't figured out all true influencers and add them to our model – which is also one of the reasons why out bias is not low enough. It would have been better if we got the chance to explore more possible interactions. In addition, we believe that some variables such as origin_cat, origin_author and school_pntg are similar to each other, and they share some collinearity. Thus, it might be helpful taking a further look at these variables and adjust the model accordingly. I would also suggest try out some transformations on predictors – such as using powerTransform(), which might help to explain more variability of the dataset.

4. Development of the final model

- Final model:

```
bs.nes=gbm(as.numeric(logprice) ~ .-subject-author-lot-authorstandard-winningbidder, dis
tribution = "gaussian",data=paintings_train.e,n.trees = 300,interaction.depth =7,shrinka
ge = 0.03)
summary(bs.nes)
```



	var <fctr>	rel.inf <dbl>
sale	sale	49.839404980
Surface_Rect	Surface_Rect	5.643953277
lrgfont	lrgfont	5.126301623
material	material	4.518188883
position	position	4.429170422
winningbiddertype	winningbiddertype	4.105437506
Width_in	Width_in	4.040557778
Interm	Interm	3.570828585
origin_author	origin_author	3.446092376
origin_cat	origin_cat	2.454705852

1-10 of 51 rows Previous **1** 2 3 4 5 6 Next

* Variables: must include an explanation
* Variable selection/shrinkage:

Note： we discuss this two questions together

There are 59 variables in our data originally.

We first filter out some predictors on our own: subject, author, winnerbiddertype, authorstandard, and lot. We filter them out because all these variables have an extremely amount of levels, most of which are subject-unique: for each painting there is very likely to be a new level specifically generated for that painting. Given that given that each painting is unique, these subject-specific variables might not appear in other dataset very often. Also because of such specificity, it is very likely that these variables will dominate and be the main influence that drives price, which is not what we'd like to see.

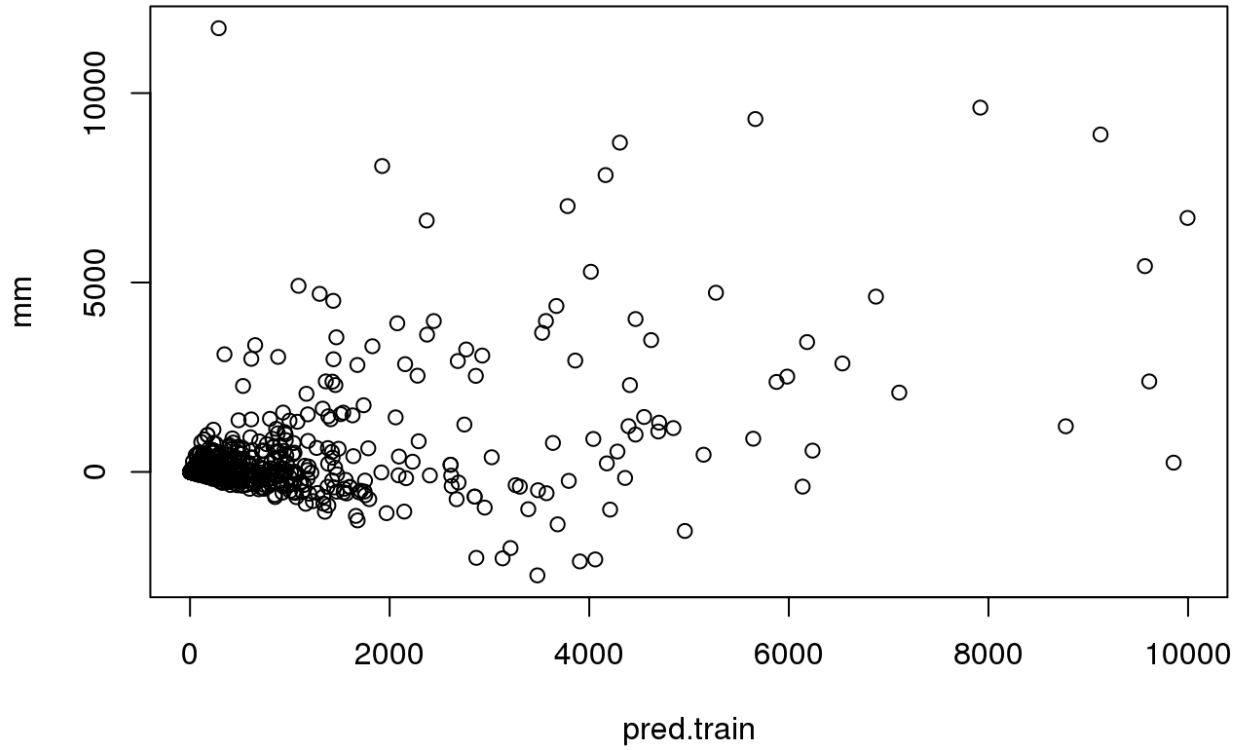
Then we recode some data, performe data cleaning to prepare the dataset used for training. For models such as blasso, xgboosting and bart which requires numeric input, we recode some categorical variables by create new binary variables using sparseMatrix(). We also recode some categorical variables to binary variables (0: the main level v.s. 1: the rest levels). For other models, however, they do not have strict restrict on the input variable types, so we just recode some NA values into median, and turn some character columns into factors.

We then feed different kinds of models with our training dataset, and tune the parameter in order to get some insights about how to choose and adjust parameters. We find that variables such as sale, Surface_Rect, material, position, lrgfont, winningbiddertype, Width_in are important variables with high relative importance in the boosting model, which is the one we finally choose to use. The variable sale is actually a combination: dealer's initial + year of sale, which is actually the interaction between dealer and year of sale. From a plot on this variable (which is the graph in question 2), we can see across years different dealers sold paintings at difference prices, which is pretty interesting. Different type of dealer, as well as the origin of paintings also influence a painting's price. We think this might be because different dealers tend to value paintings in different standards, and buyers may have a preference to paintings of some origins over paintings of other origins. Surface_rect is also revealed as a variable with high importantance, and we believe that different sizes can lead to different prices, given that rectangular paintings is the most common type of paintings. It is pretty interesting for us to find out that positions is also one of the most important variables because we consider this variable as not related to the painting itself, and thus exclude this variable from the begninning of exploring OLS model.

Some variables are excluded by the automatic selection done by our boosting model, including dealer, diam_in,

material_Cat, original, lands_ment, history, and pastorale.

* Residual:




```
## NULL
```

Looking at our residual plot – it looks like the residuals are kind of randomly distributed – although there is still some pattern, which means our boosting method isn't perfect – it still has not captured all the patterns hidden inside the dataset. the closer the predicted value to zero, the smaller the difference between predicted value and actual value is. This suggests that our model does a good job in predicting paintings with low price, but it does not very well in predicting expensive paintings – maybe because expensive paintings are evaluated more on their instrinsic values/their authors, which is something the model does not take into account.

* discussion of how prediction intervals obtained

Inspired by professor Merlise as well as the package explanation of gbm, we first set the distribution argument in gbm boosting model to quantile, and changing alpha to get the models that make predictions on the 2.5% value and the 97.5% value. Then we use these two model together with our testing dataset, to predict the corresponding quantile value, which builds up the upper and lower bounds of our prediction interval.

5. Assessment of the final model

* Model evaluation:

```
##                Bias RMSE
## OLS             184 1388
## Xgboost         657 1867
## Lasso           255 1557
## BMA             161 1694
## Random Forest   248 1494
## BART            193 1509
## GBM             172 1338
```

We explored 7 models in total: OLS, lasso, xgboost, BMA, Random Forest, BART, and gbm boosting. All of their bias and RMSE on the test dataset are shown in the table below.
As mentioned above, our OLS model has a bias of 184 and a test RMSE of 1388. The training data RMSE is about 1500, and we thus conclude that our OLS model might exclude some important variables as the model fit the test data but fails to capture training data.
As discussed in part 4, we recode some variables and transfer all valuable categorical variables to numeric binary variables in order to fit the Lasso and xgboost. The Lasso gives us a RMSE of over 1500 and xgboost about 1800, which are considerably large values. They also have super high bias, suggesting not fitting the testdata. Besides these reasons, it is also possible that the recoding procedure has some influence over the accuracy. We didn't further explore or refine these two models.
Random Forsest and BART is give us better results than xgboost and Lasso. However, we still get the RMSE that are larger than 1400 and bias are larger than 190 for both models. BMA has a good bias though, but it takes a long time to run and we couldn't reduce its RMSE remarkably via tuning parameters, so we choose not to focus on them.
We also tried to use blasso, but the result is a bit crazy, not shown here (there may be some issues inside the normalizing process, as such case happened in the blasso homework).
Overall, boosting with gbm gives us the best test RMSE and bias.
* Model testing :
Our final model has a RMSE of 1338.9 and a bias of 172.3. We further calculate the RMSE of our training dataet to test our model, and get a RMSE of about 1000. Since the RMSE of training dataset is smaller than the RMSE of test data, but the different is not very big, it generally suggests that boosting model fits both dataset (it's not overfitting training set too much), it effectively catches the major relationships of variables, and its performance is kind of consistent across both datasets. The decreased bias also shows that the model has improved from the original OLS model. It seems that our boosting models captures some underlying interactions that OLS and other methods not able to capture. The parameter, number of trees, are set to 300, which avoids the possibility of using too many trees and causing overfitting in consequence. Overall we are pretty satisfied with this boosting model.
* Model result:

```
##      Subject                                          Author
## 79   "Int\x8erieur, servante, volaille"          "Gerard Dow"
## 789  "March\x8e aux herbes d'Amsterdam"          "Gabriel Metzu"
## 1080 "Adoration des Rois"                          "G\x8erard Lairesse"
## 741  "Une Femme assise sur un mulet"               "Nicolas Berghem"
## 609  "Chasse et nombreux personnages"              "Nicolas Berghem"
## 712  "Abraham recevant les 3 anges"                "G\x8erard Lairesse"
## 874  "Vertumne et Pomone"                          "Rembrandt"
## 822  "March\x8e aux chevaux"                       "Philippe Wouwermans"
## 135  "Interieur, femme, servante, aiguiere, homme" "Gabriel Metzu"
## 294  "Nombreuses figures, sc\x8fne de village"   "JB V\x8eeninx"
##      Predicted Price    Surface    Sale    Lrgfont
## 79   "7948.54923417631" "126.75"   "R1777" "1"
## 789  "10377.8987413826" "1067.5"   "R1776" "1"
## 1080 "8751.06478463186" "3690"     "R1777" "1"
## 741  "6531.4047570185"  "3168.375" "R1767" "1"
## 609  "6388.89998947202" "936"      "R1776" "1"
## 712  "7681.00585876351" "2762.5"   "R1777" "1"
## 874  "5346.20999780733" "1530"     "R1776" "1"
## 822  "6658.92328509094" "792"      "R1769" "1"
## 135  "4716.21249348232" "775"      "R1769" "1"
## 294  "5165.41288105528" "771.75"   "R1777" "1"
```

According to the table, we can see that all of the top 10 valued paintings are sold by dealer R. Also, all the top valued paintings actually include an additional paragraph to the painting. Artist Gabriel Metzu, Nicolas Berghem, and G8erard Lairesse all have 2 paintings that rank among the top 10 highest priced paintings. This suggests that the variable author may also influence on our response a lot, however we failed to put it inside our model because it has too many levels. From the top 10 paintings, we couldn't see any obvious strong trend between Surface and price. 7 of the 10 paintings are sold in year 1777 and 1776, this might suggest that these two years are bull markets for paintings and many high priced paintings successfully sold in these two years.

6. Conclusion

Of all the models we used, boosting gives us the best model with the smallest RMSE and a relatively small bias. The variables selected by boosting as significant include sale, Surface_Rect, lrgfont, winningbiddertype, position, material, width_in and etc, although we are not very sure in which way are they influencing the result prediction (boosting is like a black box to some extent). They might be simple independent predictors, or might play a role in a multi-way interaction.

We indeed learned a lot from the process of exploring models. We learned how to recode categorical data into binary to satistfy the requirements of certain models (lasso, for instance), how to recode missing values in different ways. Besides different approaches of recoding dataset, we also get a better understanding of how to use different models, and how to tune model parameters via training and testing. For instance, Lasso and xgboost both models require numeric input, while gbm and random forest accept factor variables. It is interesting that none of them accept NA values, though.

We also learned how to get the predictive interval for different models. For blasso we need to use draws and calculate quantile percentage, for boosting we need to make use of quantile regression… It is hard but also interesting exploring these on our own.

We started this project with exploring OLS, where we simply add variables into the model based on our understanding of the meanings of the variables, as well as the correlation between the variable and the response. During this process we realize that this is very limited, as OLS cannot itself actively explore interactions efficiently. It is also very hard for us to judge whether an high-dimensional interaction should be add into model or not. With

that experience, we learned how useful tree-based methods are: they captures interaction really well (which can be seen from the decrease in the test RMSE and Bias). Although we are not able to clearly extract those interactions out and interpret them (which is the limit of multi-tree models), the predictions are actually very competitive and accurate.