Statistical Analysis of Decision-Making

Jun Hernandez
Stat 183
Dr. Xu

Introduction:

Today, an important skill is decision-making under pressure and proper resource management. We can see these skills in settings like business management, team production, and everyday life. The aim of this study is to answer the following:

What role does emotional response have on decision-making? Is there a way to overall quantify and model this decision-making process, and if so, how can we relate this to an individual based on a number of explanatory variables?

Researchers have published an open dataset, claiming that “this is the first dataset simultaneously covering [the] four facets of decision-making,” hopefully used to provide insight.

Study Design:

The dataset contains detailed data reflecting the decision situations, decision strategies, decision outcomes, and the emotional responses of 1,144 participants from diverse backgrounds. The data was collected through *Agile Manager*, a game simulating complex project management processes (http://agilemanager.algorithmic- crowdsourcing.com/). The game puts participants into various scenarios of managing a team of virtual worker agents (WAs) with diverse characteristics. It unobtrusively collects participants’ sequential decision-making behaviour trajectory data over time under various conditions of uncertainty and resource constraints.

The Agile Manager game platform was made available starting in December 2013 through a dedicated website hosted by the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University (NTU), Singapore for personal computers running the Windows XP operating system or higher. Enrollment is open to anyone who chooses to download and play the game.

In order to track participant performance, the researchers implemented a scoring system based on tasks successfully completed by deciding how tasks should be assigned to the WAs in each round of the game The participant’s score is increased by the value of the task if both the quality and timeliness requirements are fulfilled. Otherwise, his/her score remains unchanged. In order to provide a benchmark for participants to know how well their strategies perform, which hopefully motivates them to improve their strategies, an artificial intelligence (AI) competitor is included in the game. At the end of each game session, the overall outcome of a participant’s decisions made during the session (information about the score and whether the participant beat the AI competitor) is presented to him/her (Fig. 1e). The participant is then required to report the strategy he/she used during the game session. The participant is also required to report his/her emotions after knowing the outcome of the game session. The participant can specify the degrees of the six basic emotions and select an emoticon that best represents his/her facial expression at the moment.

Data:

A total of six data tables are included in this dataset, however the most useful towards this study were:

1. **Users.xlsx**: Shows the categories of participants' demographic information released in this dataset, including sex, age, education level, location, and personality survey data.

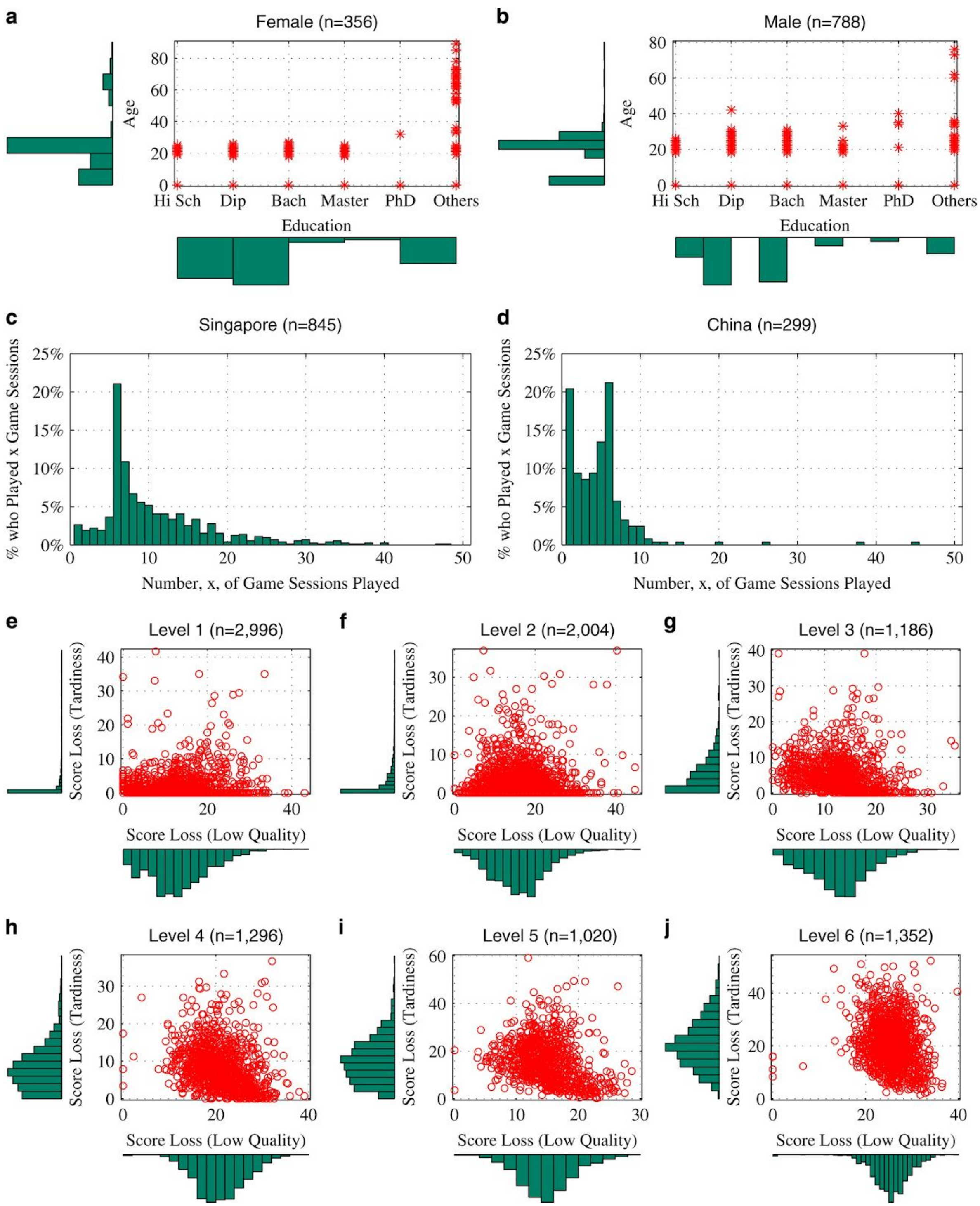
Variable Name	Range	Description
ID	NA	The participant's unique identification number
Gender	'Male', 'Female'	The participant's gender
Education	'High School', 'Diploma', 'Bachelor', 'Master', 'PhD', 'Others'	The participant's highest level of education
Country	'Singapore', 'China'	The country the participant is located in
Age	NA`	The participant's age at the time when he/she joined the study
Account Creation Time	NA	The exact date and time a participant joined the study
PQ1—PQ10	{1,2,3,4,5}	10 survey questions used for assessing the participant's personality
AQ1—AQ20	{1,2,3,4,5}	20 survey questions used for assessing the participant's affective-oriented disposition

2. **Game Sessions.xlsx**: Includes data regarding the collective outcome of the participant's decisions, such as a *User Strategy Index,* an encoded variable used to express what decision strategies were used. Also includes participants' emotional response based on facial expression and a survey of 6 basic emotions.

Variable Name	Range	Description
ID	NA	The unique identification number of a game session
User ID	NA	The unique identification number of the participant who played this game session
Game Level	1–6	The identification number of the game level played in this game session
Player Score	0–100%	The score obtained by the participant in this game session
Player Score Loss (Low Quality)	0–100%	The score lost by the participant as a result of tasks being completed with low quality in this game session
Player Score Loss (Tardiness)	0–100%	The score lost by the participant as a result of tasks not completed before their stipulated deadlines in this game session
AI Score	0–100%	The score obtained by the AI participant in this game session
AI Score Loss (Low Quality)	0–100%	The score lost by the AI participant as a result of tasks being completed with low quality in this game session
AI Score Loss (Tardiness)	0–100%	The score lost by the AI participant as a result of tasks not completed before their stipulated deadlines in this game session
User Strategy Index	'100,000'–'111,111'	The index value expressing the participant's self-reported task allocation strategy used in this game session
User Strategy Description	NA	The participant's explanation about his/her task allocation strategy used in this game session (optional)
Facial Expression ID	0–36	The unique identification of the emoticon selected by a participant to represent his/her emotion
Happiness	0–10	The participant's self-reported degree of happiness
Sadness	0–10	The participant's self-reported degree of sadness
Excitement	0–10	The participant's self-reported degree of excitement
Boredom	0–10	The participant's self-reported degree of boredom
Anger	0–10	The participant's self-reported degree of anger
Surprise	0–10	The participant's self-reported degree of surprise

Dependent variables: New variables were created as a response when cleaning the data, and are the following: Average Player Score per user, Proportion of the player beating the ai over games played, and Player Score.

Initial EDA:



Sub-figures (a,b) show the scatter-plots and the density distributions of the age and education levels for female and male participants, respectively. Sub-figures (c,d) show the density distributions of the number of game sessions by participants from Singapore and China, respectively. Sub-figures (e–j) illustrate the scatter-plots and the density distributions of the normalized scores (in the range of 0–100%) lost by the participants due to 1) low quality of work and 2) failure to meet deadlines in game levels 1–6, respectively. The higher the game level, the higher the overall workload placed on the virtual team of WAs (i.e., the more challenging for decision-making).
From: https://www.nature.com/articles/sdata2016127#rightslink

1.1 Question 1

For question 1, we want to see how emotions play a role in a users "Player Score", in particular how Happiness levels and Sadness levels compare.

1.2 Methodology - ANOVA

We are interested in testing the effects of emotions (in this case, happiness and sadness) on the player score for a game session.

- Dependent variables: Player Score(response)
- Factors:
  - Average happiness levels:
    - Low
    - Mid
    - High
  - Average Sadness levels:
    - Low
    - Mid
    - High
- Objective: Test both happiness and sadness effects on the Player Score.

Statistical Model

Two-Way ANOVA:

$y_{ijk} = \mu + \tau_i + \beta_j + \tau\beta_{ij} + \epsilon_{ijk}$

i - happiness average level(low, mid, high)

j - sadness average level(low, mid, high)

k - player(1,…,721)

$\epsilon \sim N(0, \sigma^2)$

Where:

Y_{ijk} : The kth measurement corresponding to the ith and jth factors

μ : the overall mean

τ_i : the average happiness effect

β_j : the average sadness effect

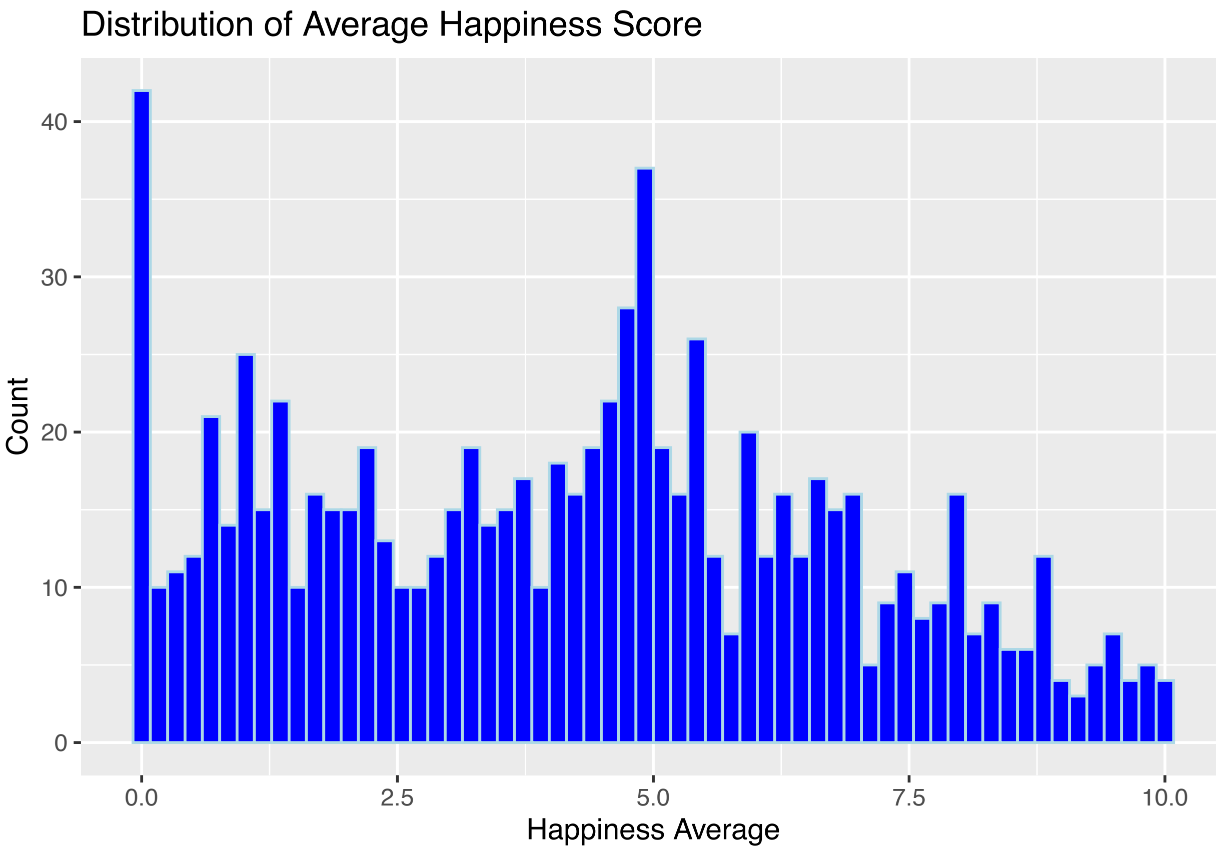
$\tau\beta_{ij}$: the interaction effect between happiness and sadness

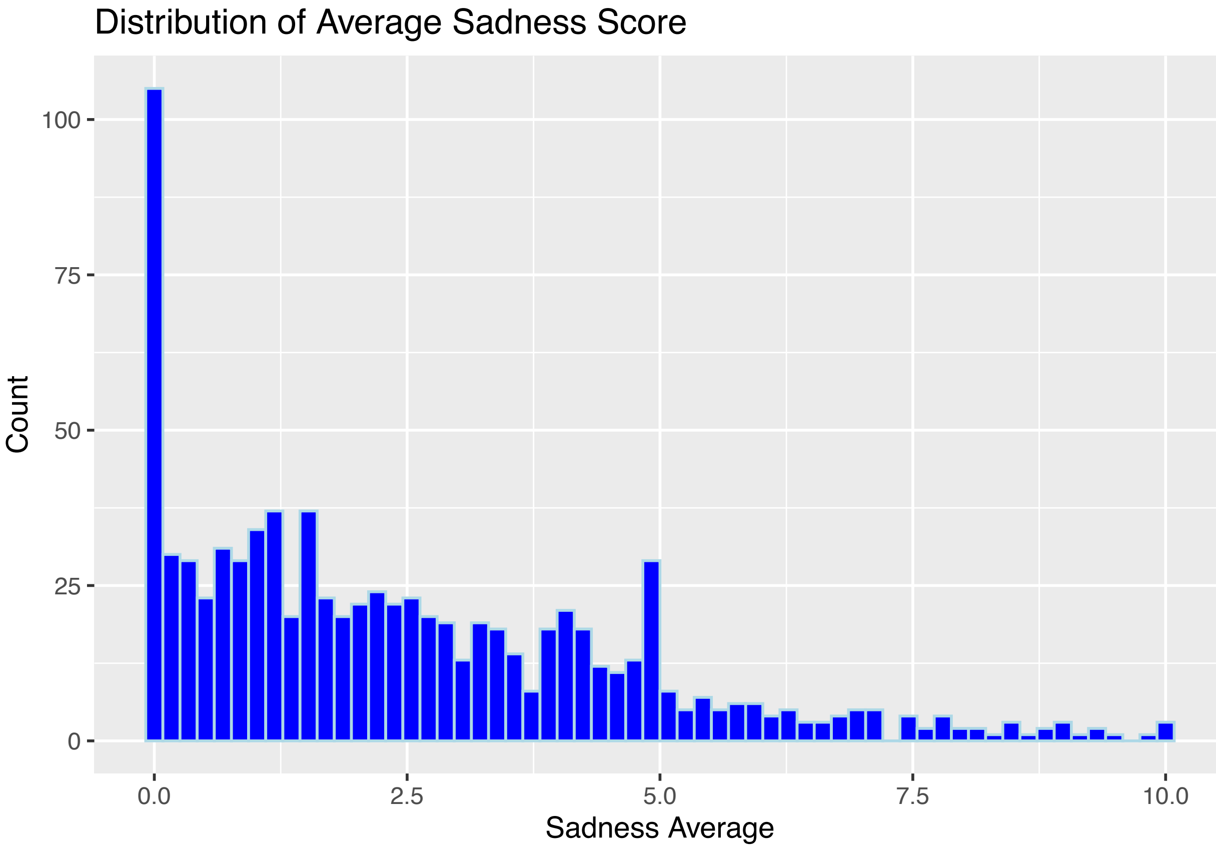
ϵ_{ijk} : the random error

1.3 Cleaning the data

The dataset we use for the ANOVA is the Game Sessions data, since it has information on the players self reported emotional levels after playing a game. After loading the data, we clean it by handling missing values, and selecting a small subset of the variables we are interested in. Based on analysis done during question 2, it makes sense to partition the data since there are a number of users with only 3 or less games played. In order to have the data be better representative of performance and with less bias, it is justifiable to only include players with 3 or more game session in the main modeling data set. Further analysis is provided in question 2. Overall, we keep data on 721 users after subsetting and cleaning the data. Finally, we average the players performance scores so the data is independent.

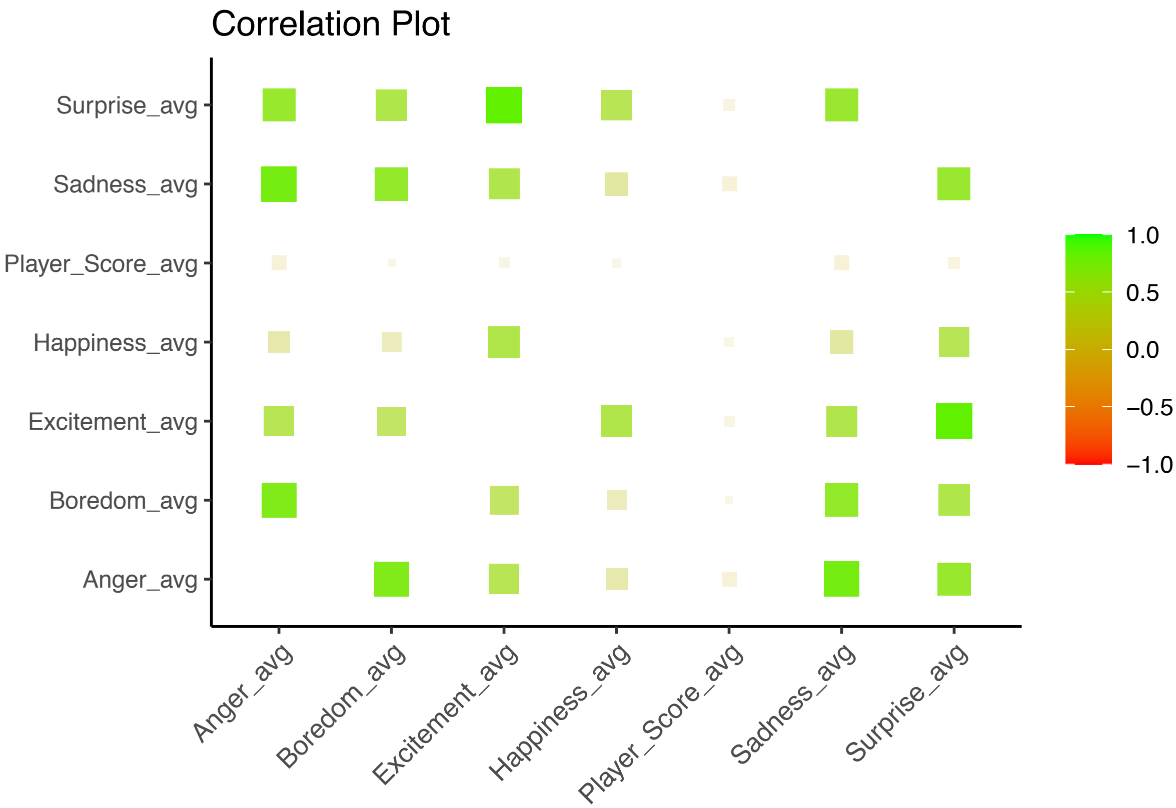
EDA





Since we are interested in comparing the average happiness score and the average sadness score for the response "Player Score", we plot the distributions for both variables. For the ANOVA, the data has been grouped into 3 levels based on subjective ratings 1-3 = low, 4-7 = mid, 8-10 = high. This is subjective choice, however makes sense in the context of emotional levels.

1.4 Correlation Check and Plots

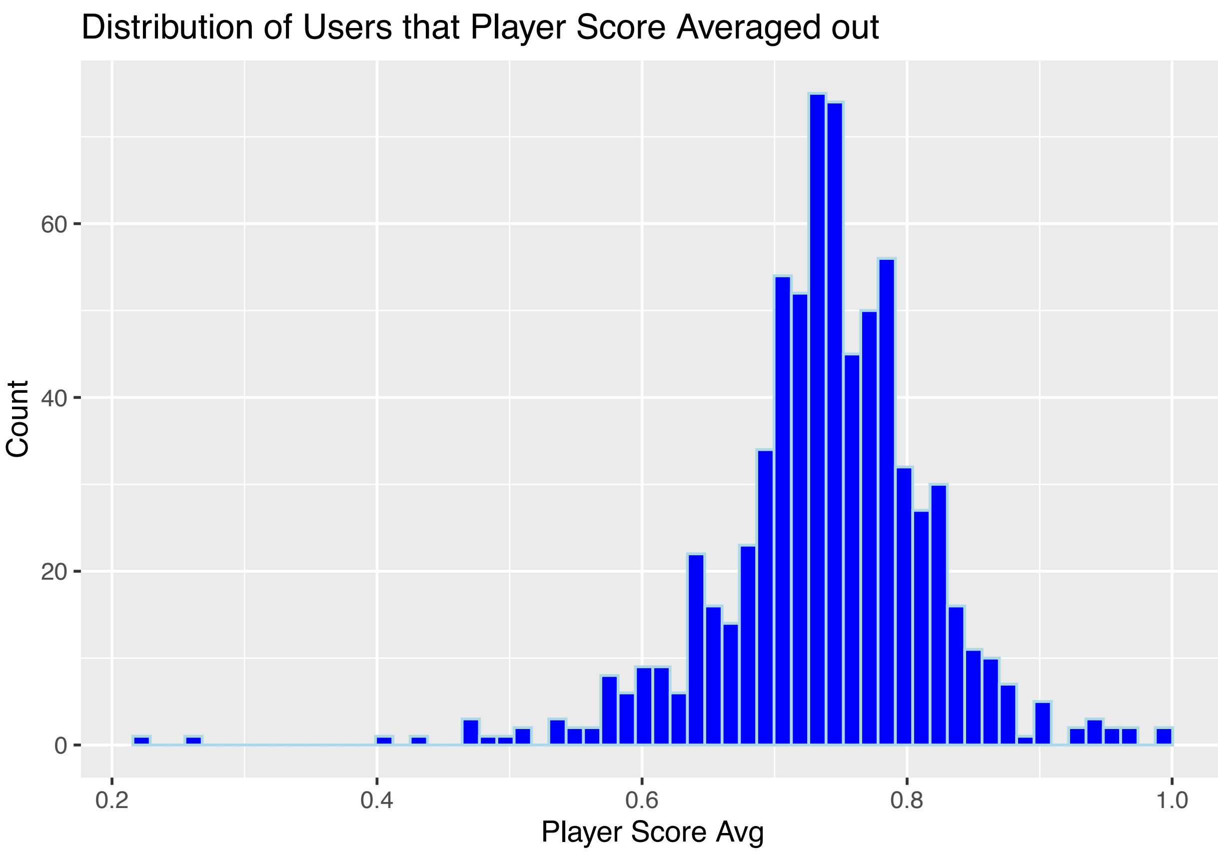



```
## [1] "Surprise_avg" "Sadness_avg"  "Anger_avg"
```

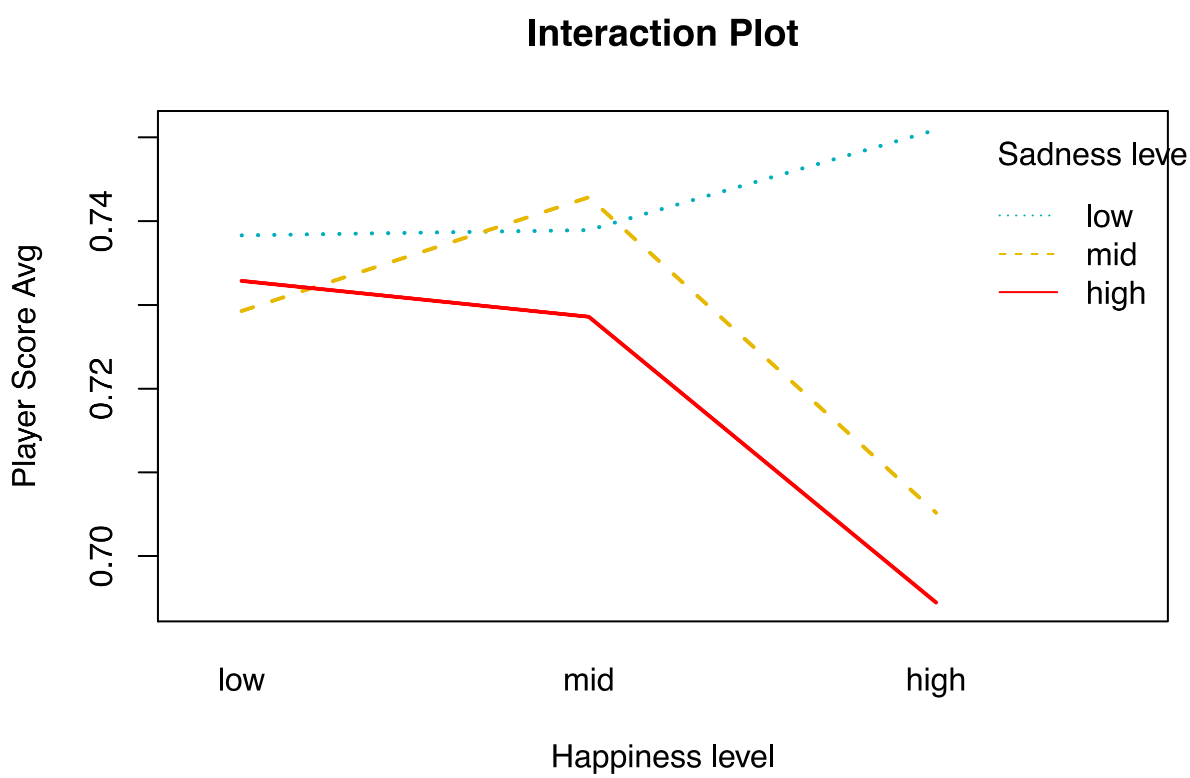
We chose "Happiness_avg" and "Sadness_avg" since the variables have a low correlation of 0.2597974. Based on the graph and correlation matrix, a number of variables like "Surprise_avg", "Anger_avg", including "Sadness_avg" have high correlation with other variables. We can now pick the final variables we are interested in:

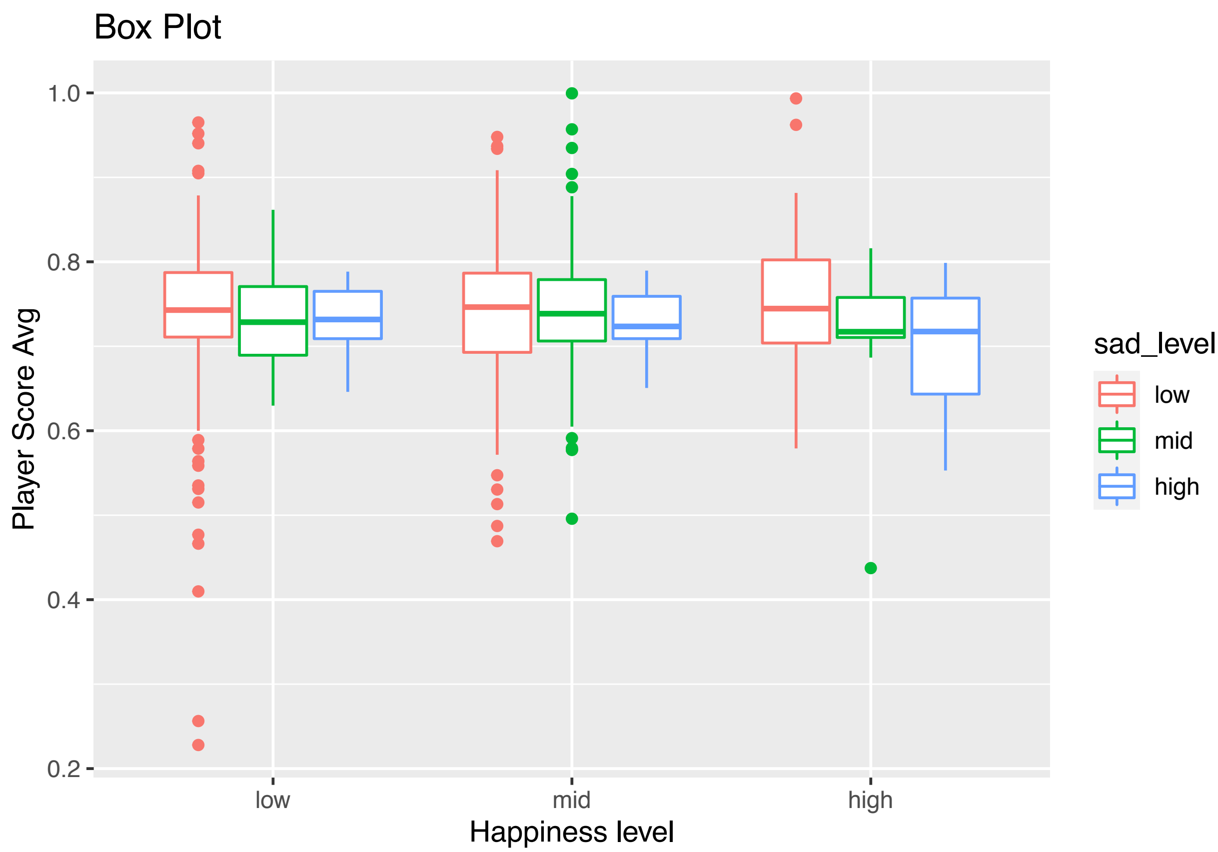
"Player_Score_avg", "Happiness_level", "Sadness_level", and "count_avg", which tells us the number of games a user played. Since the counts are different, this data set is unbalanced.

EDA



Box plot & Interaction plot



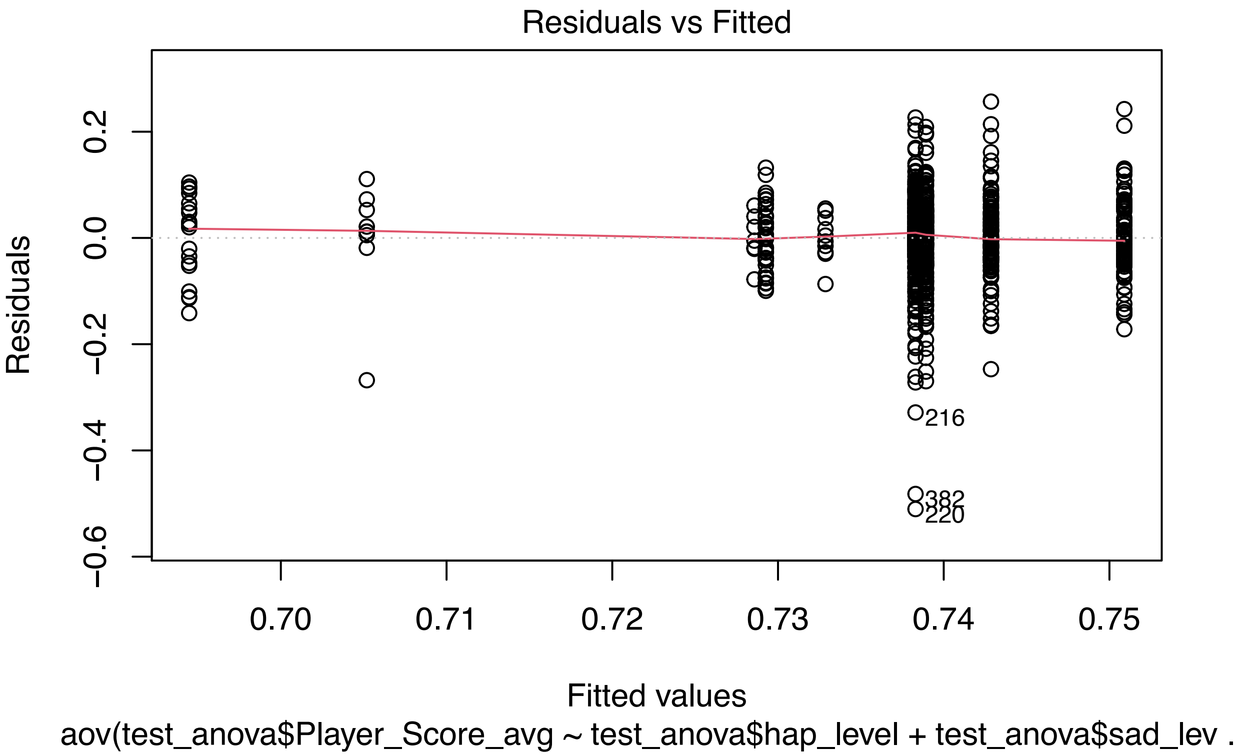


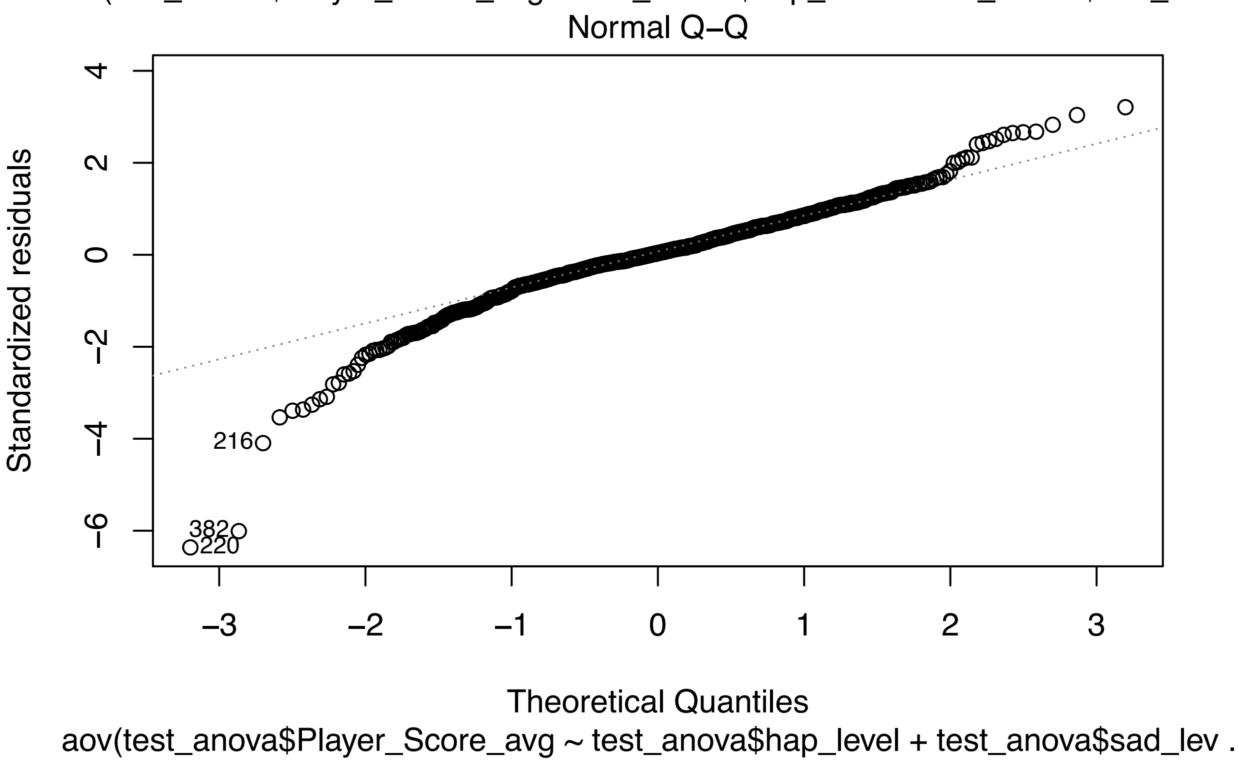
From the interaction plot, there may be an interaction between happiness and sadness levels

1.5 ANOVA

We first fit a two-way ANOVA model with interaction on the original scale of the dependent variable.

```
## Analysis of Variance Table
##
## Response: test_anova$Player_Score_avg
##                                         Df Sum Sq   Mean Sq F value Pr(>F)
## test_anova$hap_level                     2 0.0015 0.0007724  0.1196 0.8873
## test_anova$sad_level                     2 0.0269 0.0134406  2.0815 0.1255
## test_anova$hap_level:test_anova$sad_level   4 0.0342 0.0085576  1.3253 0.2589
## Residuals                              712 4.5975 0.0064572
```



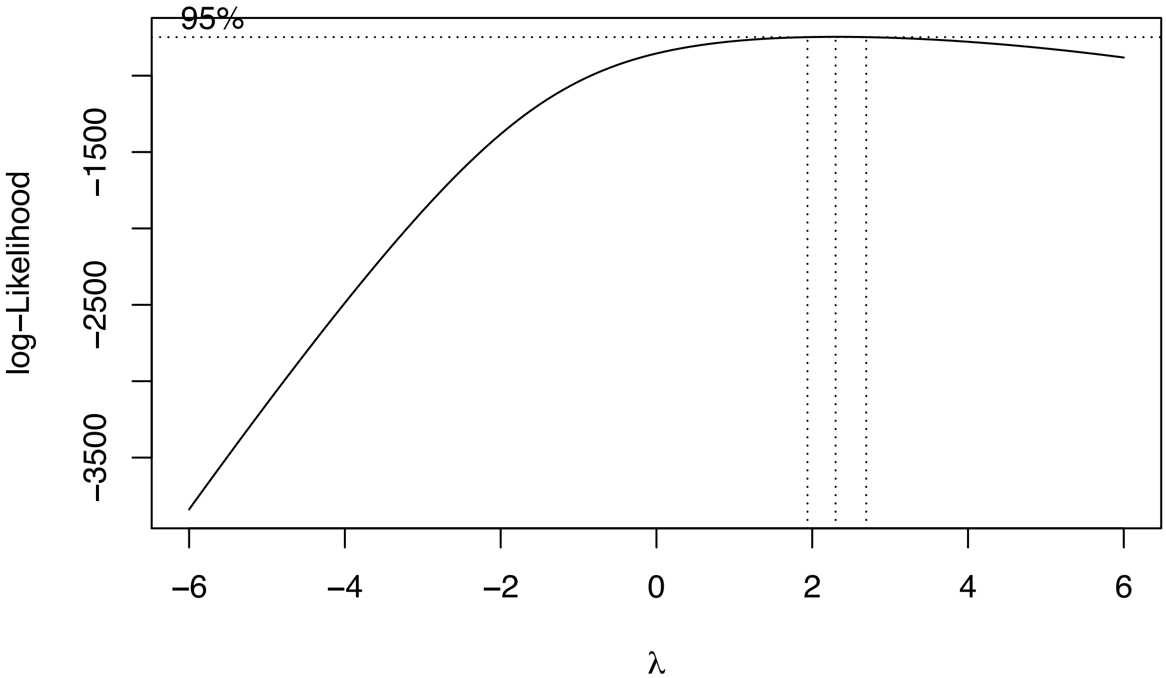



```
##
##  Shapiro-Wilk normality test
##
## data:  aov_residuals
## W = 0.9418, p-value = 3.389e-16
```

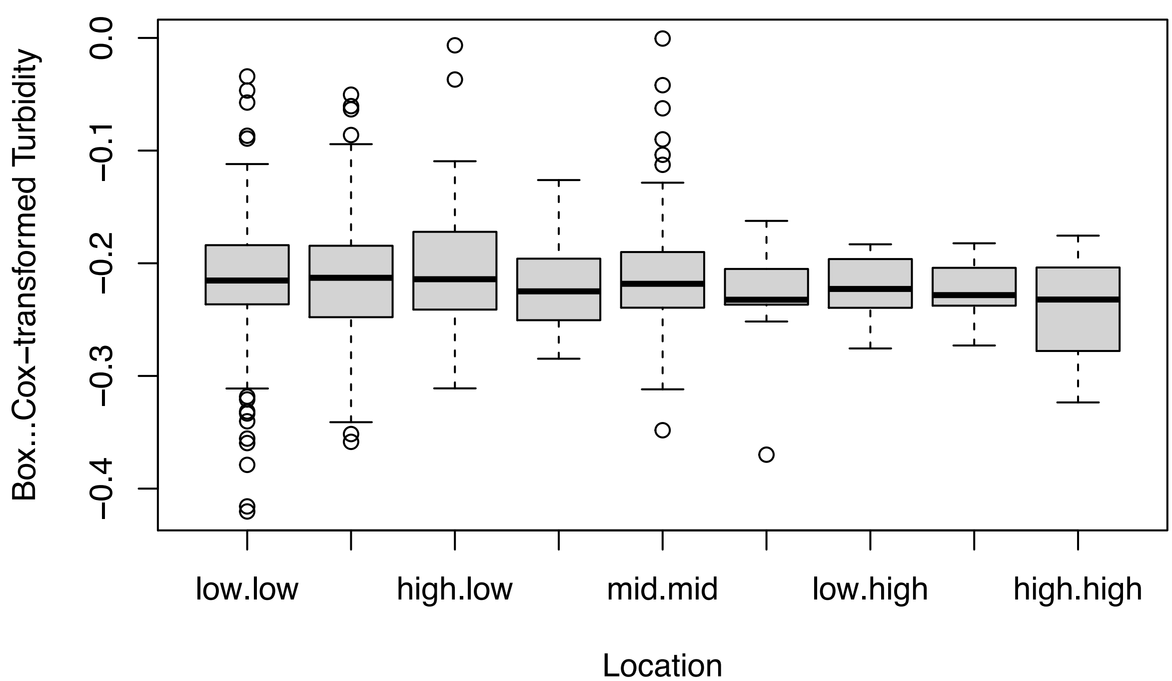
We can see that in the ANOVA, none of the factors are significant. Further analysis shows there is unequal variances from the residuals vs fitted plot, and the Shapiro-Wilk normality test has a p-value = 3.389e-16, which confirms that the data is not normally distributed. The Interaction plot suggests there is an interaction even though none of the factors are statistically significant.

Transformation

Box Cox transformation was used and λ was deterrmined to be 2.3. Let y* = $y^{2.3}$, we fit a Type III SS ANOVA model since this is an unbalanced design, on the transformed data.

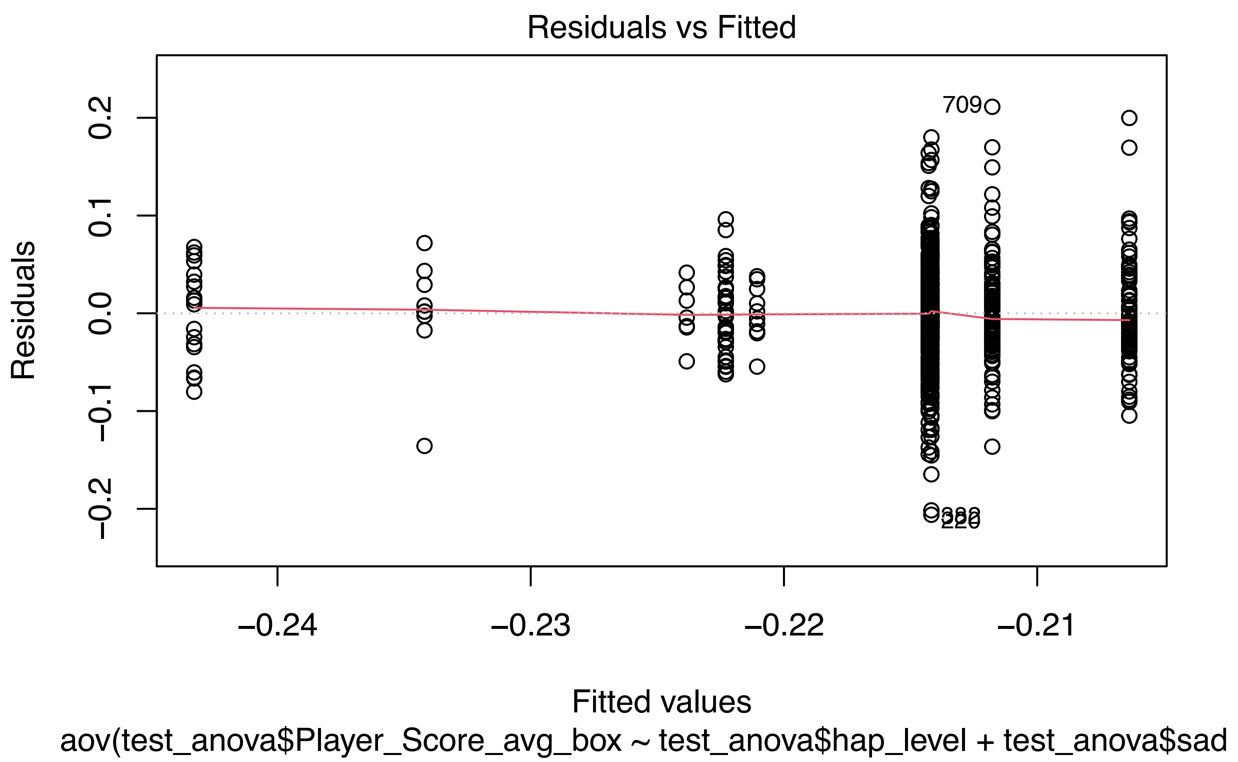


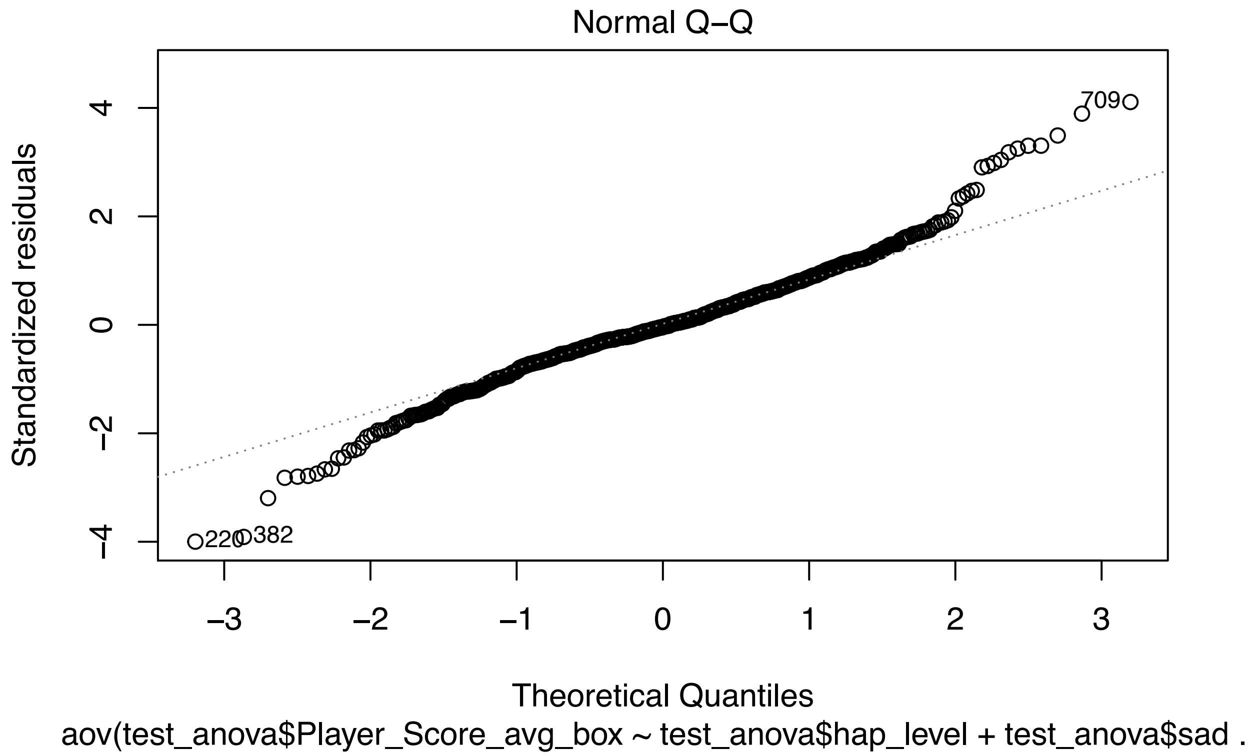

```
##    Box.x     Box.y
## 84   2.3 -745.6827
```




```
## Analysis of Variance Table
```

```
##
## Response: test_anova$Player_Score_avg_box
##                                          Df  Sum Sq   Mean Sq F value  Pr(>F)
## test_anova$hap_level                      2 0.00049 0.0002430  0.0911 0.91296
## test_anova$sad_level                      2 0.01384 0.0069184  2.5933 0.07548
## test_anova$hap_level:test_anova$sad_level   4 0.01316 0.0032898  1.2332 0.29533
## Residuals                               712 1.89945 0.0026678
##
## test_anova$hap_level
## test_anova$sad_level                        .
## test_anova$hap_level:test_anova$sad_level
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```






```
##
##  Shapiro-Wilk normality test
##
## data:  aov_residuals2
## W = 0.97532, p-value = 1.121e-09
```

```
## Single term deletions
##
## Model:
## test_anova$Player_Score_avg_box ~ test_anova$hap_level + test_anova$sad_level +
##     test_anova$hap_level * test_anova$sad_level
##                                           Df Sum of Sq    RSS     AIC F value
## <none>                                                 1.8995 -4264.1
## test_anova$hap_level                       2 0.0040296 1.9035 -4266.5  0.7552
## test_anova$sad_level                       2 0.0024395 1.9019 -4267.1  0.4572
## test_anova$hap_level:test_anova$sad_level  4 0.0131593 1.9126 -4267.1  1.2332
##                                           Pr(>F)
## <none>
## test_anova$hap_level                      0.4703
## test_anova$sad_level                      0.6332
## test_anova$hap_level:test_anova$sad_level 0.2953
```

1.6 ANOVA Results

Unfortunately, even after a transformation, the data is still non-normal, and there are unequal variances. Based on this, the final type III ANOVA is inconclusive, and the unequal variances in the data will not hold for non-parametric methods.

2.1 Question 2

For question 2, we want to see if we can model whether or not a player was able to beat the AI score.

2.2 Methodology - Logistic Regression Model

In order to use a logistic regression model, we need to set up a binary or grouped proportion variable as a response variable for the data. Since we are trying to model whether or not a player was able to beat the AI score, we can make a new binary variable from the data to show if the player during the game session beat the AI score. Here we are using both the Game Sessions and Users data sets, in order to have as many significant predictors as we can. Both data sets required significant cleaning, such as renaming variables, dealing with missing values, and overall preparing the data to work well with modeling.

2.3 Cleaning the data

After the inital summary analysis of the data sets and renaming the variables, we join both data sets by User ID. The combined data set has a total of 9,269 game sessions played, and a total of 994 unique users.

Setting up new datasets for modeling:

As stated earlier, we create a new response binary variable named beat_ai. This variable is a 1 if the Player Score is greater than the AI Score, and 0 otherwise. This is the response variable for our modeling.

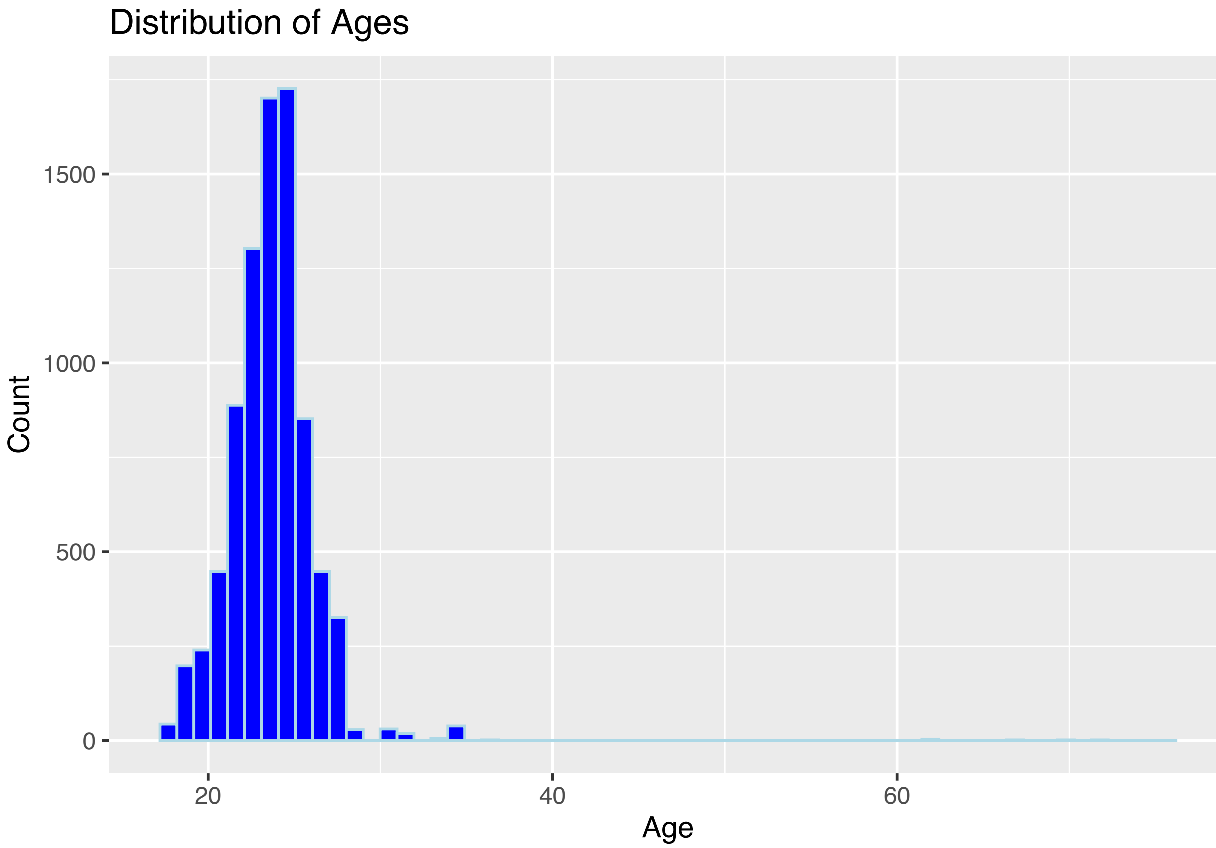
Next, we pick a subset of the total number of variable that would try to best explain the response. These variables were as follows:

"User_ID", "beat_ai", "Game_Level", "User_ScLoss_LowQuality", "User_ScLoss_Tardiness", "AI_ScLoss_LowQuality", "AI_ScLoss_Tardiness", "Happiness", "Sadness", "Excitement", "Boredom", "Anger", "Surprise", "Gender", "Education", "Country", "Age"

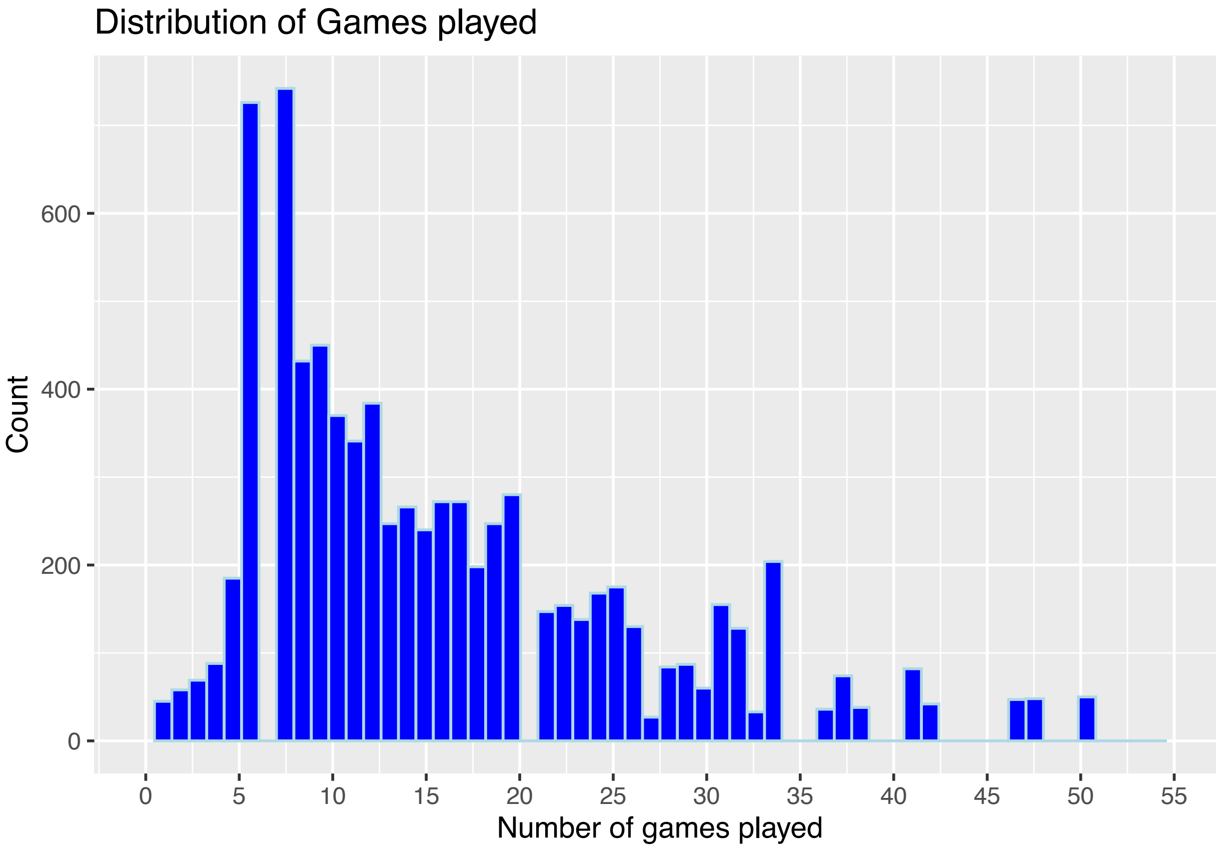
We had to drop the variables of "User_Strategy_Description" since only a few number of participants described their strategies. We also had to drop "Facial_Expression_ID" and "User_Strategy_Index". Since these variable had over 20 categorical levels, logistic regression would not be able to handle these well as predictors, although they can potentially explain the response. The personality question survey answers were also dropped, since a very few number of participants answered the survey.

More data cleaning is necessary, as there are a number of nonsensical values due to errors with the data collection. Some of the problems were scores outside the given range in "User_Score_Loss" and "Ages" inputted as 0. After cleaning, we are left with 789 unique users, down from 994, and a total of 8,309 rows of data.

Finally, we have to turn the character values into factors, or categorical levels. We turn the variables "Education", "Gender", "Country" into factors, as well as releveling the variables to have arbitrary undordered reference levels.



Simple EDA:
Some initial EDA is shown, to see the distribution of ages and make sure that there are no nonsensical values from before. The values range from 18 to 76, with the average age of 24.



Data Cleaning Cont.

Number of games played:

n > 3 : 692 Users - 692 / 789 = 87.7 % of user data kept

n > 5 : 633 Users

n > 10 : 265 Users

n > 15 : 148 Users

n > 20 : 77 Users

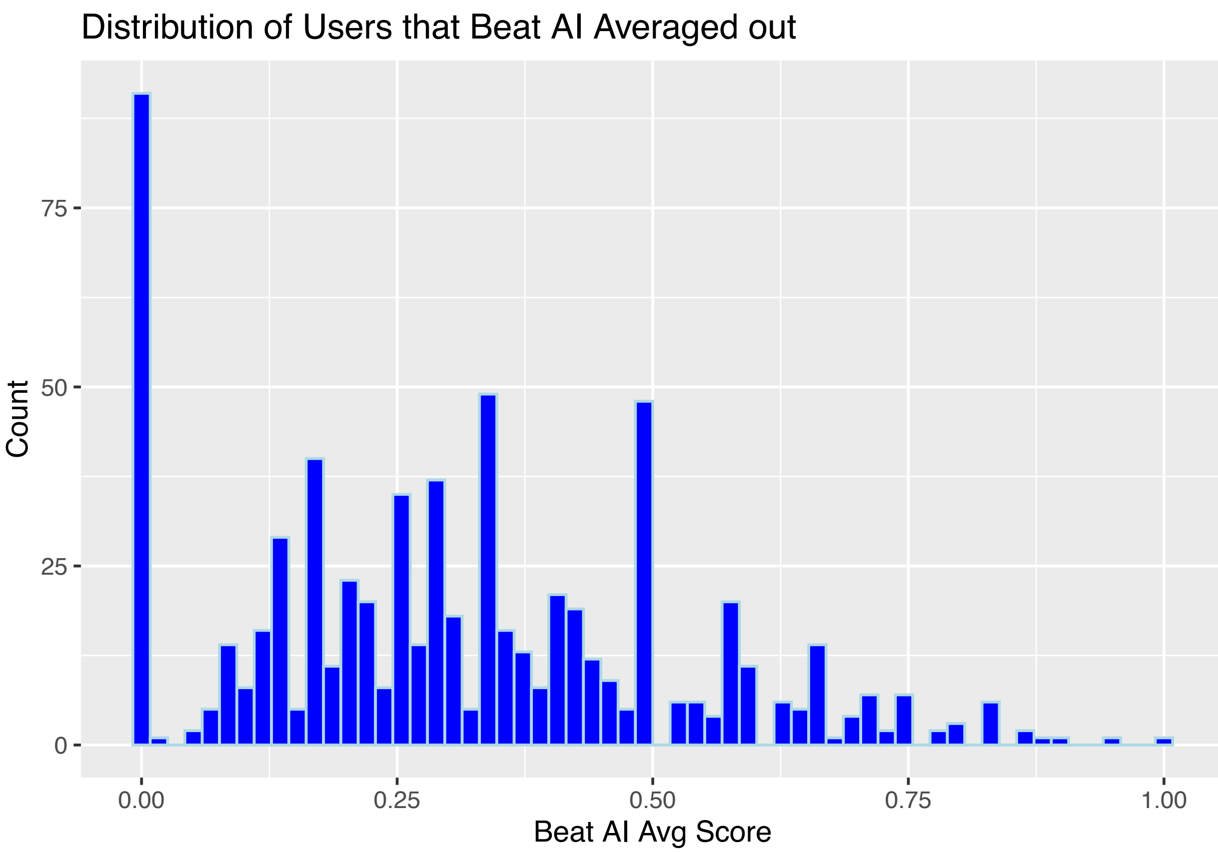
n > 25 : 42 Users

n > 30 : 29 Users

Since there are a number of users with only 3 or less games played, in order to have the data be better representative of performance and with less bias, it is justifiable to only include players with 3 or more game session in the main modeling data set. Overall, we keep data on 692 users, which is still 87.7% of the user data kept.

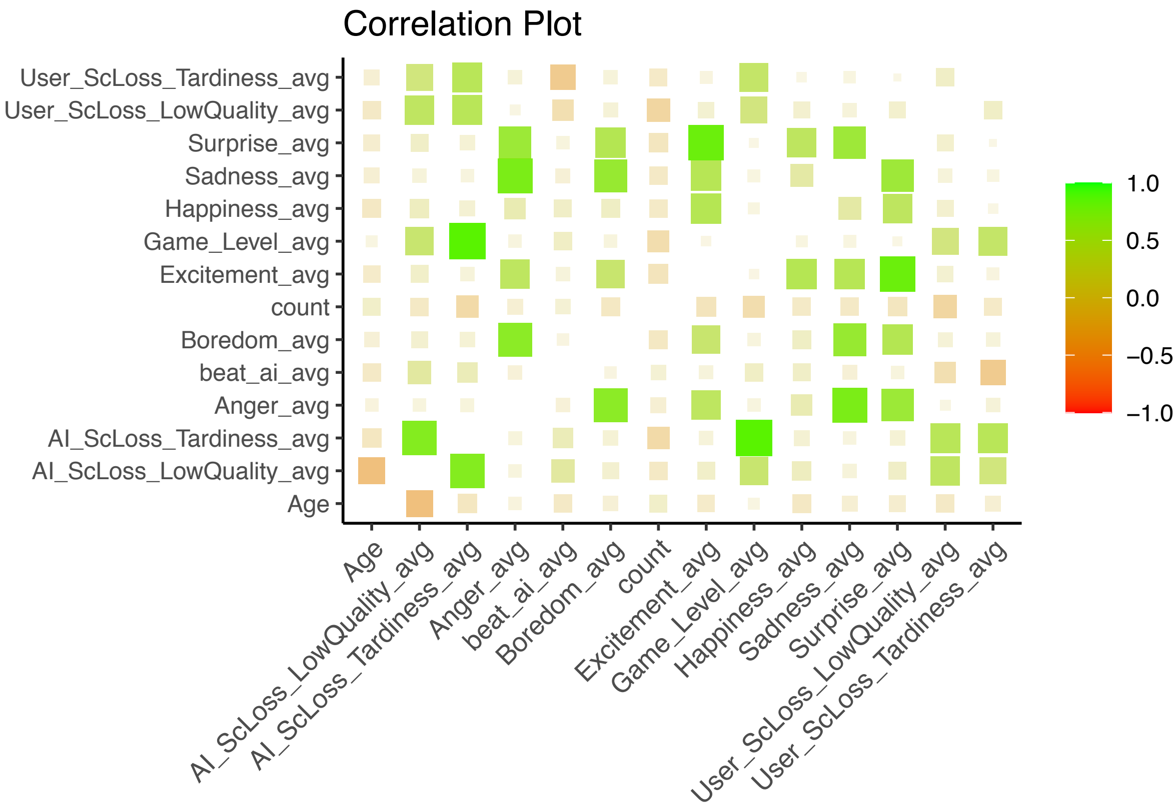
Group by Users

We are now ready to aggregate and summarize the data by user, so each row is an average of the players games sessions, and the data is independent. The continous variables were averaged based on number of games played, and the categorical variables were left as is. A new variable is also added, "Count", which is the number of games played by the user. Since the response variable "beat_ai" was originally a binary variable, it is now "beat_ai_avg", which is the proportion of games won over the AI. We will use this variable as the response since logistic regression also works with proportion response variables, with the "Count" variable as a weight for the model.



Distribution of Users that Beat AI Averaged out
Count
75
50
25
0
0.00
0.25
0.50
0.75
1.00
Beat AI Avg Score

2.4 Correlation Check




```
## [1] "Surprise_avg"              "AI_ScLoss_Tardiness_avg"
```

We can see from the graph and correlation matrix that only a few variables are correlated. Setting a cutoff point at .7, only the variables "AI_ScLoss_Tardiness_avg" and "Surprise" are highly correlated with other variables. From this, we can leave out the variables "AI_ScLoss_Tardiness_avg", but decide to keep "Surprise" since this may provide good information as a predictor in the model.

2.5 Initial Logistic Regression Model

First we split the data into training and validation sets to check for model accuracy, and run the model based on our cleaned data set.

```
##
## Call:
## glm(formula = beat_ai_avg ~ ., family = binomial(link = "logit"),
##     data = TrainSet_log, weights = count)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8621  -0.7805  -0.0270   0.5621   3.1712
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -1.423177   0.488346  -2.914  0.00357 **
## Game_Level_avg              0.863304   0.078897  10.942  < 2e-16 ***
## User_ScLoss_LowQuality_avg -17.992491   0.961347 -18.716  < 2e-16 ***
```

```
## User_ScLoss_Tardiness_avg  -20.786483   1.130722 -18.383  < 2e-16 ***
## AI_ScLoss_LowQuality_avg    24.042038   1.851638  12.984  < 2e-16 ***
## Happiness_avg                0.023941   0.015117   1.584  0.11326    
## Sadness_avg                 -0.023720   0.027163  -0.873  0.38252    
## Excitement_avg               0.002597   0.026497   0.098  0.92192    
## Boredom_avg                 -0.018296   0.021649  -0.845  0.39803    
## Anger_avg                   -0.004291   0.028008  -0.153  0.87824    
## Surprise_avg                 0.026740   0.029796   0.897  0.36948    
## GenderFemale                 0.004289   0.075597   0.057  0.95475    
## EducationDiploma            -0.049238   0.109153  -0.451  0.65193    
## EducationBachelor           -0.167575   0.105416  -1.590  0.11191    
## EducationMaster              0.162794   0.297086   0.548  0.58371    
## EducationPhD                 1.099036   0.603060   1.822  0.06839 .  
## EducationOthers             -0.002948   0.132084  -0.022  0.98220    
## CountrySingapore             0.024935   0.165700   0.150  0.88039    
## Age                         -0.006463   0.019806  -0.326  0.74418    
## count                       -0.004385   0.002059  -2.130  0.03320 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1225.08  on 482  degrees of freedom
## Residual deviance:  443.51  on 463  degrees of freedom
## AIC: 1562.5
##
## Number of Fisher Scoring iterations: 4
```

Initial Model

$logit(\pi_i) = log[\pi_i/(1-\pi_i)] = \beta_0 + \beta_1 x_1 + ... + \beta_{19} x_{19}$

where:

π_i = the proportion of games beating the AI by the player i

β_i = the regression coefficients for factor x_i

with 19 independent variables in the model.

Initial Model Perfomance

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

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: beat_ai_avg
##
## Terms added sequentially (first to last)
##
##
##                            Df Deviance Resid. Df Resid. Dev  Pr(>Chi)    
## NULL                                         482    1225.08              
## Game_Level_avg              1   29.248       481    1195.83 6.369e-08 ***
## User_ScLoss_LowQuality_avg  1  145.997       480    1049.83 < 2.2e-16 ***
## User_ScLoss_Tardiness_avg   1  295.594       479     754.24 < 2.2e-16 ***
```

```
## AI_ScLoss_LowQuality_avg    1  286.157       478     468.08 < 2.2e-16 ***
## Happiness_avg               1    6.616       477     461.47   0.01011 *
## Sadness_avg                 1    1.458       476     460.01   0.22728
## Excitement_avg              1    0.457       475     459.55   0.49884
## Boredom_avg                 1    0.343       474     459.21   0.55817
## Anger_avg                   1    0.511       473     458.70   0.47449
## Surprise_avg                1    1.648       472     457.05   0.19920
## Gender                      1    0.283       471     456.76   0.59452
## Education                   5    8.258       466     448.51   0.14260
## Country                     1    0.113       465     448.39   0.73727
## Age                         1    0.182       464     448.21   0.66978
## count                       1    4.707       463     443.51   0.03004 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For the training data, we conduct the Deviance test, which has a p-value = 0. We also conduct the Wald test, which confirms the model's significance, meaning it predicts the response variable in the training data at a quality that is unlikely to be pure chance. Finally, we find the psuedo $R^2 = 0.6387$, which means the model explains 63.87% of the deviance. For the validation data, we find the psuedo $R^2 = 10.99\%$. Unfortunately, this tells us that we haven't yet identified all the factors that actually predict the response variable.

2.6 Reduced Logistic Regression Model

Fitting a second model based on backwards selection, we find the reduced logistic regression model.

```
##
## Call:
## glm(formula = beat_ai_avg ~ Game_Level_avg + User_ScLoss_LowQuality_avg +
##     User_ScLoss_Tardiness_avg + AI_ScLoss_LowQuality_avg + Happiness_avg +
##     count, family = binomial(link = "logit"), data = TrainSet_log,
##     weights = count)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8758  -0.7646  -0.0480   0.5512   3.0734
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -1.599580   0.192745  -8.299   <2e-16 ***
## Game_Level_avg              0.808999   0.068774  11.763   <2e-16 ***
## User_ScLoss_LowQuality_avg -17.947270   0.923278 -19.439   <2e-16 ***
## User_ScLoss_Tardiness_avg  -20.373037   1.096436 -18.581   <2e-16 ***
## AI_ScLoss_LowQuality_avg    24.516873   1.484000  16.521   <2e-16 ***
## Happiness_avg               0.025012   0.012126   2.063   0.0391 *
## count                      -0.004333   0.001873  -2.313   0.0207 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1225.08  on 482  degrees of freedom
## Residual deviance:  455.88  on 476  degrees of freedom
## AIC: 1548.8
##
```

```
## Number of Fisher Scoring iterations: 4
```

Reduced Model:

$logit(\pi_i) = log[\pi_i/(1-\pi_i)] = \beta_0 + \beta_1 x_1 + ... + \beta_6 x_6$

where:

π_i = the proportion of games beating the AI by the player i

β_i = the regression coefficients for factor x_i

with 6 independent variables in the model.

The most significant predictors for the response were average Game Level difficulty, the average User Score Loss due to Low Quality performance, the averge User Score Loss due to Tardiness in the game, the average Happiness level, and the number of games played.

Reduced Model Performance

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

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: beat_ai_avg
##
## Terms added sequentially (first to last)
##
##
##                            Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                         482    1225.08
## Game_Level_avg              1   29.248       481    1195.83 6.369e-08 ***
## User_ScLoss_LowQuality_avg  1  145.997       480    1049.83 < 2.2e-16 ***
## User_ScLoss_Tardiness_avg   1  295.594       479     754.24 < 2.2e-16 ***
## AI_ScLoss_LowQuality_avg    1  286.157       478     468.08 < 2.2e-16 ***
## Happiness_avg               1    6.616       477     461.47   0.01011 *
## count                       1    5.587       476     455.88   0.01810 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## [1] 0.6278777
```

Again, for the training data, we conduct the Deviance test and the Wald test, which shows the model is significant. We find the psuedo $R^2 = 0.6278$, which means the model explains 62.78% of the deviance. For the validation data, we find the psuedo $R^2 = 10.95\%$. Unfortunately, this model still does not perform well with the validation data, meaning there is only so much that the included predictors can explain in predicting the response variable.

3.1 Question 3

For question 3, we want to see if we can model the overall Player Score with a number of predictors.

3.2 Methodology - Random Forest Model

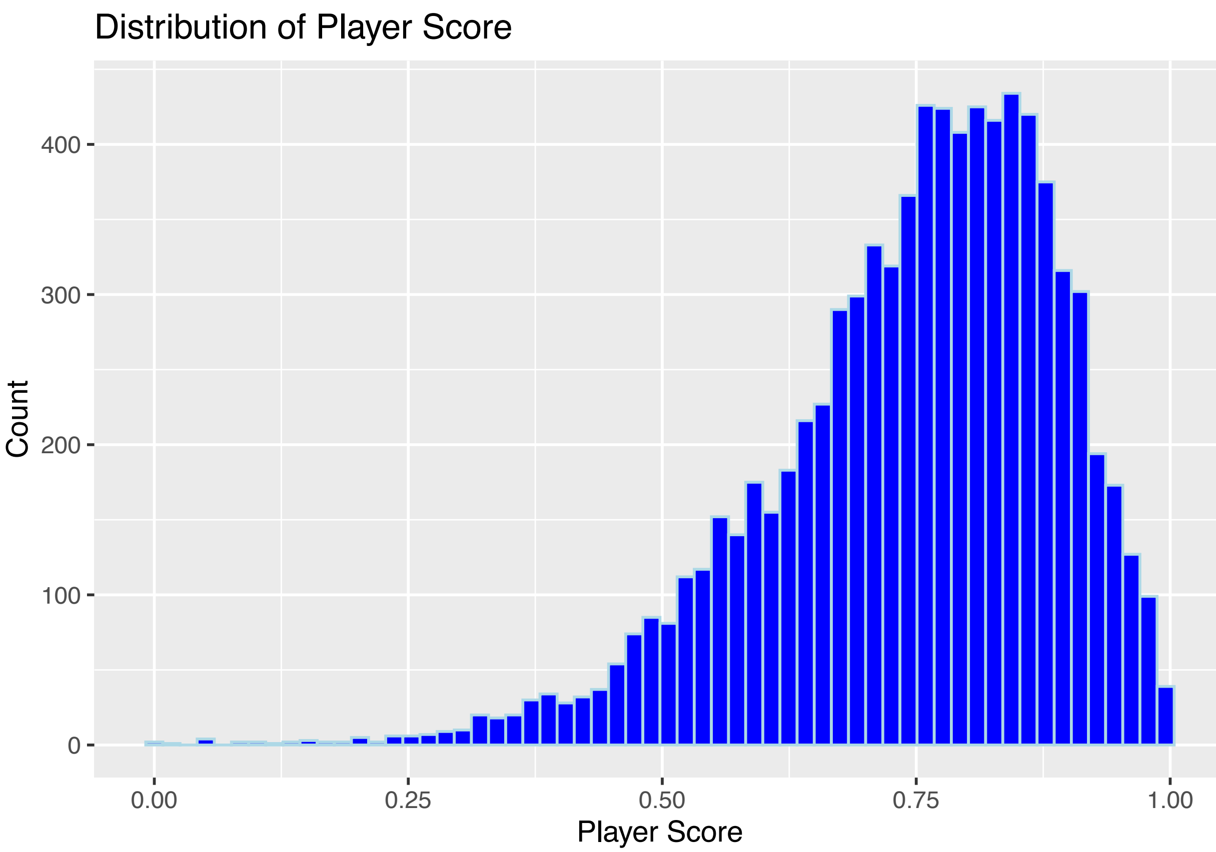
Random forests are a modification of bagged decision trees that build a large collection of de-correlated trees, that has powerful predictive performance. Using random forest for regression, we can treat the "User Score" as a response variable and include similar predictors as the logistic regression model. Again, we use both the Game Sessions and Users data sets. The "ranger" package was used to build the random forest model, which has a strong out-of-the-box performance. We can then tune the parameters to further reduce the RMSE, which is an estimate of how well the model was able to predict the validation set outcomes, with the added benefit of being measured in the same units as the response variable "Player Score".

3.3 Cleaning the data

In order to use random forest, we must remove any columns with many NAs. We use a very similar cleaned data set compared to the logistic regression model, but can now include categorical variables with many levels, such as "Facial Expression ID" with 26 levels and "User Strategy Index" with 37 levels. We also turn any character data in factors.

We overall have 780 unique users, and a total of 8,241 rows of data. We can use most of the data since random forest is non-parametric. The variables in the model were as follows:

"Player_Score", "beat_ai", "Game_Level", "User_ScLoss_LowQuality", "User_ScLoss_Tardiness", "AI_Score","AI_ScLoss_LowQuality", "AI_ScLoss_Tardiness","User_Strategy_Index", "Facial_Expression_ID","Happiness "Sadness", "Excitement", "Boredom", "Anger", "Surprise", "Gender", "Education", "Country", "Age"



3.4 Initial Random Forest Model

First we split the data into training and validation sets to check for model performance, and run the model based on our cleaned data set.

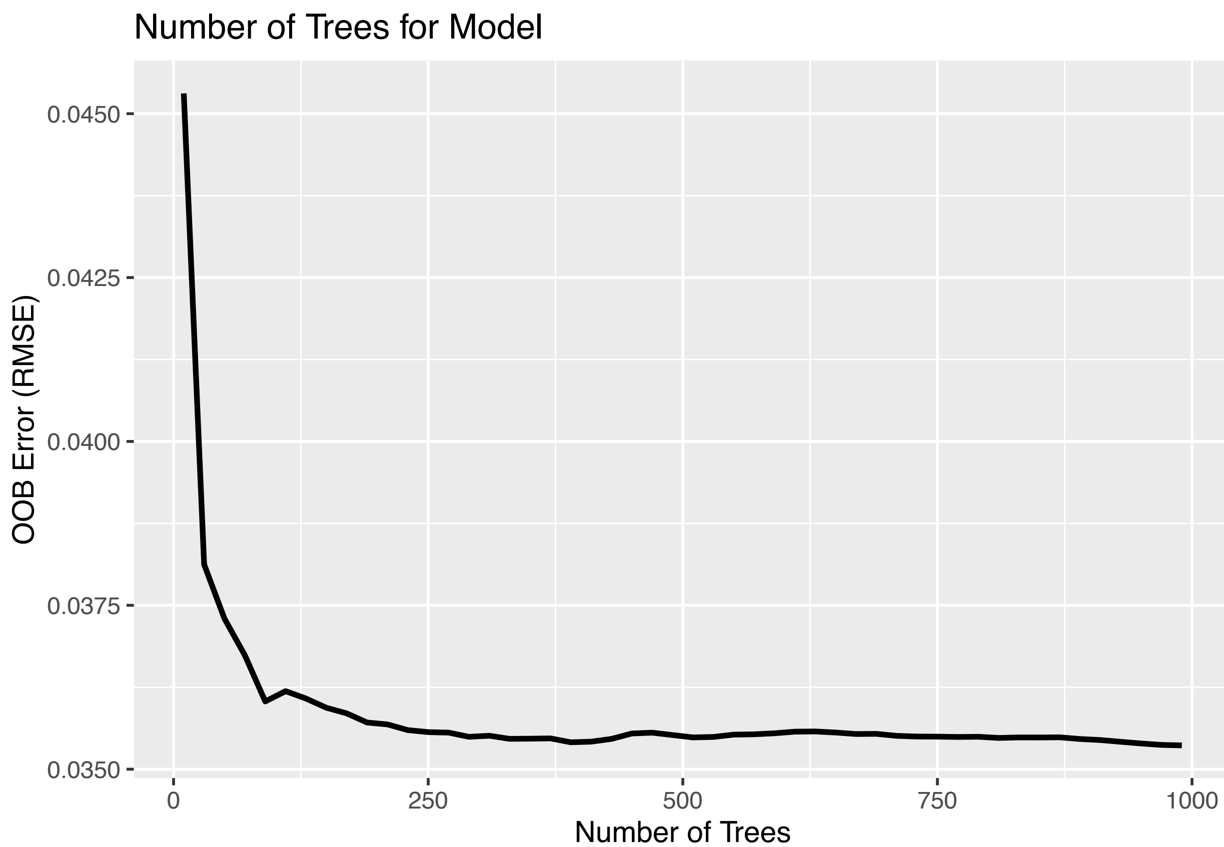
Initial Model Performance

An important parameter for random forest is the mtry parameter, which is split-variable randomization where each time a split is to be performed, the search for the split variable is limited to a random subset of mtry of the original number of features. Since we are doing regression based modeling, a good rule of thumb is to use mtry = the number of features divided by 3, which is mtry = 6. Based on the out-of-box-performance, we get an initial RMSE = 0.03551626.

3.5 Tuning the model parameters

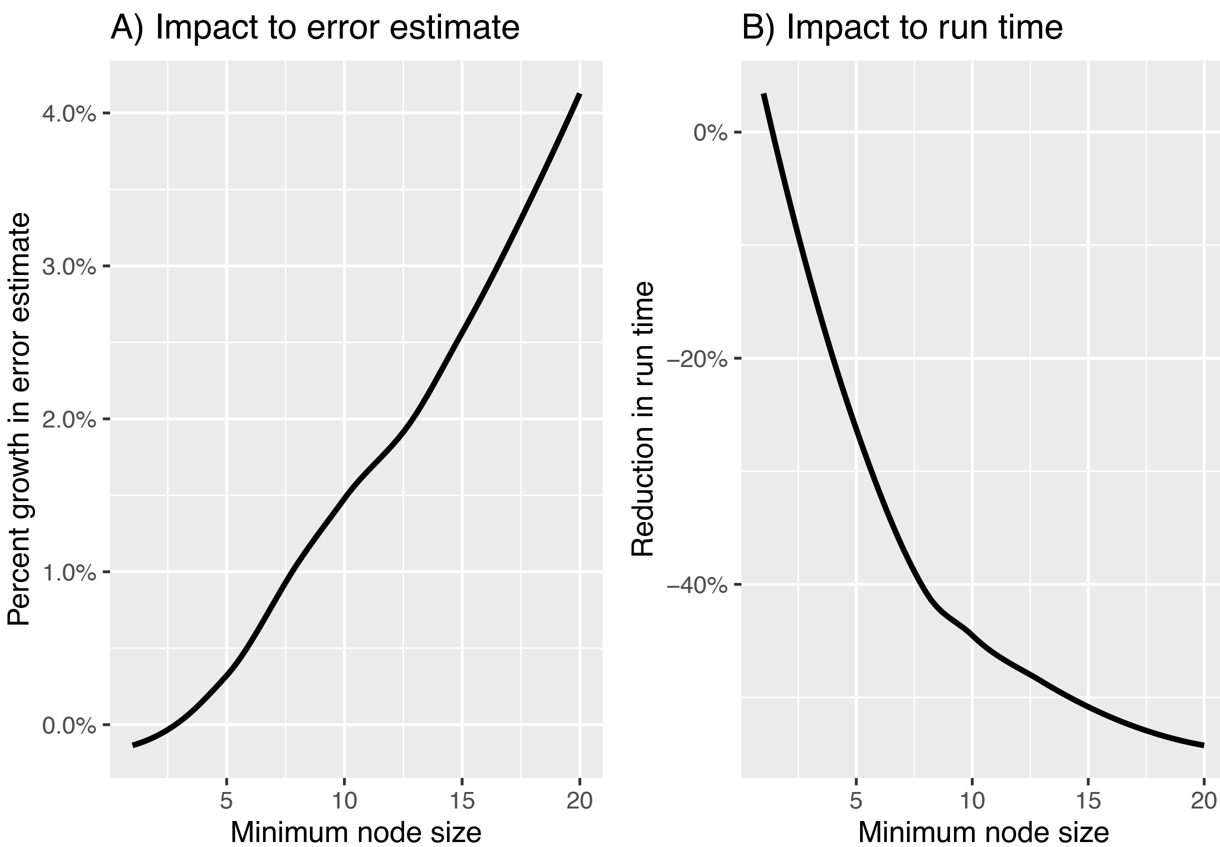
Now that we have our initial model, we can tune a number of important parameters besides mtry, such as:

1. The number of trees in the forest
2. The complexity of each tree
3. The sampling scheme used

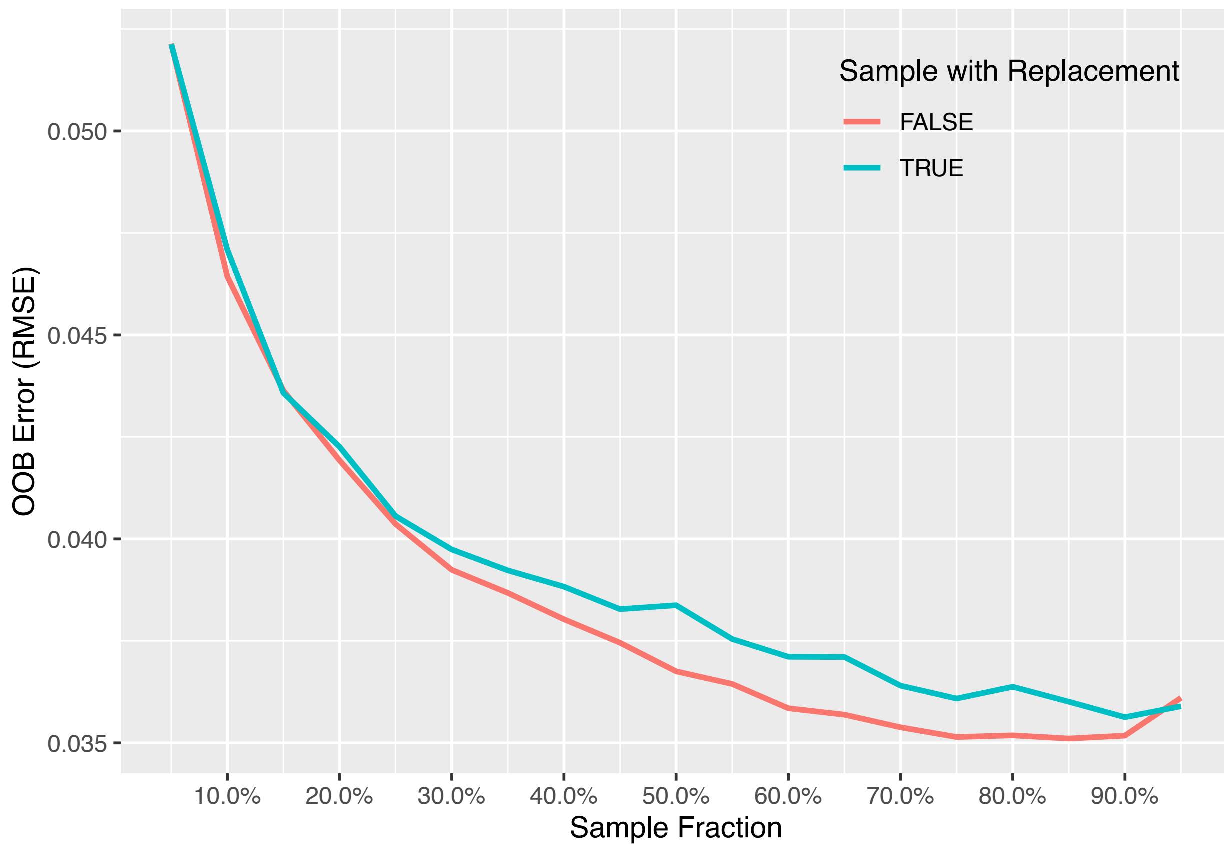


We can see from the graph that an ideal number of trees is around 375. More trees may reduce the RMSE, however this comes at a cost of computational complexity.

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The next parameter is tree complexity, and we can see the relationship with the minimum node size and percent growth in error estimate and reduction in computational run time.



Moving on to the parameter of sampling scheme, we see that we can actually lower the RMSE by sampling without replacement, at around 80% sample size. This parameter determines how many observations are drawn for the training of each tree. Decreasing the sample size leads to more diverse trees and lowers the between-tree correlation, which may have a positive effect on the prediction accuracy.

3.6 Hyper Grid Tuning Strategy and Final Model

We conduct a full Cartesian grid search to assess every combination of hyperparameters of interest.

```
##    mtry min.node.size replace sample.fraction       rmse perc_gain
## 1     7             3   FALSE            0.80 0.03453105 2.7739700
## 2     7             1   FALSE            0.63 0.03458200 2.6305222
## 3     7             5   FALSE            0.80 0.03466088 2.4084186
## 4     7             3   FALSE            0.63 0.03475535 2.1424378
## 5     7             1   FALSE            0.80 0.03478304 2.0644727
## 6     7            10   FALSE            0.80 0.03481969 1.9612950
## 7     7             5   FALSE            0.63 0.03502258 1.3900220
## 8     6             5   FALSE            0.80 0.03519132 0.9149141
## 9     7             1    TRUE            0.80 0.03528090 0.6626829
## 10    6             3   FALSE            0.80 0.03528940 0.6387617
```

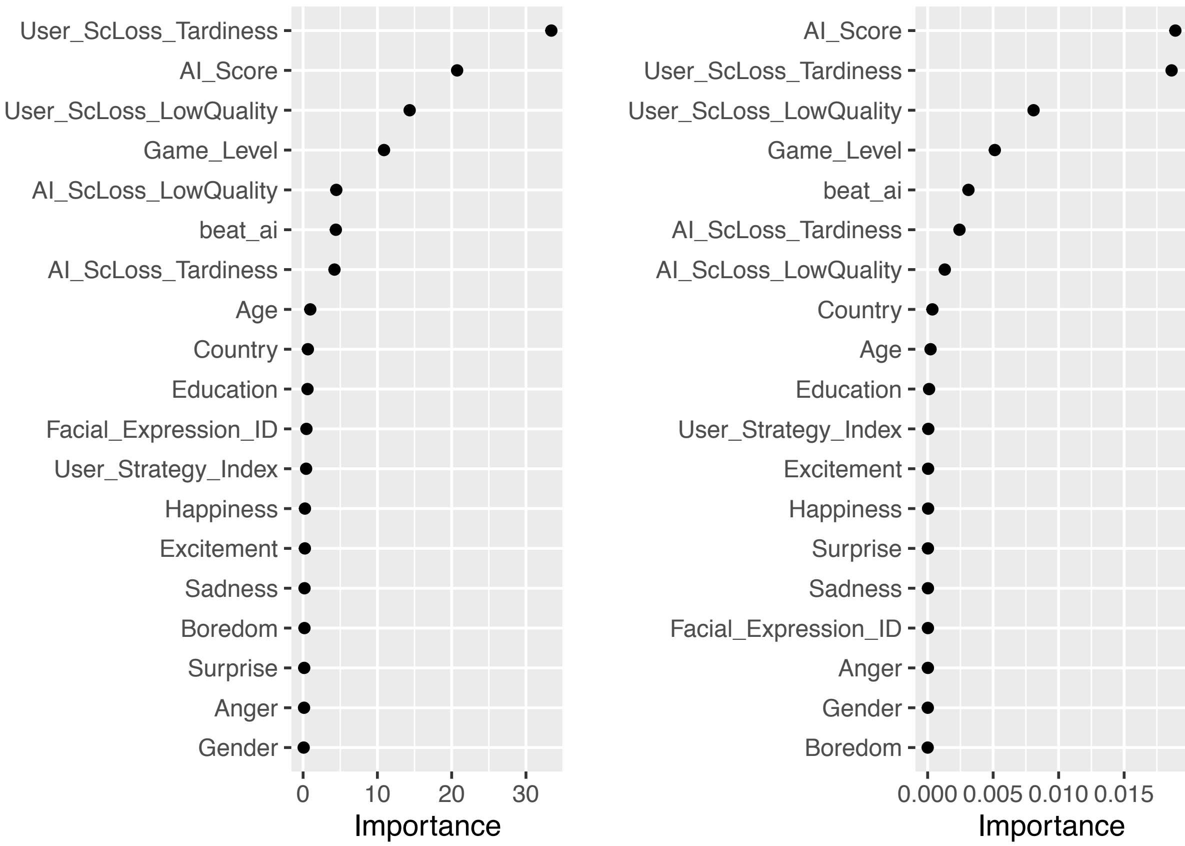
```
## [1] 0.04598446
```

We see that the top model has hyperparameters mtry = 7, min.node.size = 3, sample_with_replace = false, sample.fraction = 0.80, all with an RMSE = 0.03453105, which has a 2.77% performance gain over the out-of-box model. Overall, the final random forest model RMSE is 4.5984% as large as the mean of the validation set “Player Score” response variable.

3.7 Feature Importance

```
## Warning in vip.default(rf_impurity, num_features = 25, bar = FALSE): The `bar`
## argument has been deprecated in favor of the new `geom` argument. It will be
## removed in version 0.3.0.
```

```
## Warning in vip.default(rf_permutation, num_features = 25, bar = FALSE): The
## `bar` argument has been deprecated in favor of the new `geom` argument. It will
## be removed in version 0.3.0.
```



We can see from the graph the impurity-based measure of feature importance, where we base feature importance on the average total reduction of the loss function for a given feature across all trees, and the permutation-based importance measure, where for each tree, the out-of-box sample is passed down the tree and the prediction accuracy is recorded. These are the most important features for predicting "User Score".

4 Conclusion

In this study, we tried to find out the answers to these questions:

1. How do emotions play a role in a users "Player Score", in particular how Happiness levels and Sadness levels compare?

These results are unfortunately inconclusive since the data is non-normal and has unequal variances. Due to the unequal variances, we cannot use non-parametric methods.

2. Can we model whether or not a player was able to beat the AI score?

While the models did not perform as well on the validation set, we found that the most significant predictors for whether or not the player beat the AI, on average, were average Game Level difficulty, the average User Score Loss due to Low Quality performance, the averge User Score Loss due to Tardiness in the game, the average Happiness level, and the number of games played. There were other interesting descriptive statistics, such as the distribution of beating the AI on average.

3. Can we model the overall User Score with a number of predictors?

Our final model had a RMSE = 0.03453105 and the is about 4.60% as large as the mean of the validation set "Player Score" response variable. We can also see from the analysis the most important predictors in determing User Score. Based on this, we can variable screen and eliminate variables that are not of interest and identify important variables for future modeling, without affecting the quality of the final model.