Yoshita Narang, Kenan Stredic

Professor Tang

STAT 4355

11 May 2023

Insights & Recommendations for Entrepreneurs: Analyzing Shark Tank Companies

Team Sharks

1. Introduction

As we navigate the complex world of business and entrepreneurship, there exists a rich mine of data that can provide unique insights into the very essence of success. From the high-stakes negotiation tactics to the intricate dynamics between entrepreneurs and investors, the realm of startups is ripe for exploration.

Our chosen stage for this analysis is none other than the popular television show, Shark Tank. A veritable treasure trove of business strategy and negotiation tactics, Shark Tank provides an unparalleled platform for entrepreneurs seeking investments to fuel their business dreams. It's a show that's as much a lesson in venture capitalism as it is a thrilling spectacle of dreams, passion, and occasionally, the heartbreaking reality of business. In our study of Applied Linear Models, and the R programing language, we've decided to delve deep into this intriguing world of business negotiations, using the vast Shark Tank dataset from Kaggle. This dataset, while incredibly rich and detailed, required a significant amount of cleaning and preprocessing to align with our analytical needs. We focused on the first ten seasons of the show, exploring various parameters like the category of the business, the geographical location, the stake percentage asked by the entrepreneurs, the requested investment amount, the pre-show company valuation, and the specific episode number. This report will illuminate the rigorous data-cleaning process we employed to refine these variables and prepare them for our in-depth analysis.

2. Data Cleaning

As part of our course, Applied Linear Models, our group embarked on a meticulous exploration of the Shark Tank dataset, sourced from Kaggle. This dataset was rich, yet it required substantial cleaning and preprocessing to ensure it was ready for our intensive analysis. Our investigative scope was confined to a subset of this dataset, concentrating on companies that had been featured on Shark Tank within the first ten seasons. This approach streamlined our dataset, making it more specific and hence manageable for our designated timeframe. Our analytical focus was directed towards a selection of variables within the dataset. The chosen variables were integral to our analysis, these comprised:

- **Category (Discrete Variable):** This was used as a categorizing tool to group products and services into broader, general categories.
- **Location (Discrete Variable):** This represented the geographical location of the company producing the product or service.
- **Percent of Stake Asked For (Continuum from 3% to 100%):** This expressed the percentage of the company that the entrepreneur wished to maintain ownership of during negotiations with potential investors.
- **Amount of Money Asked For (Continuum from $10,000 to $5,000,000):** This represented the financial sum requested by the entrepreneurs from the investors for their venture.
- **Company Valuation (Continuum from $40,000 to $30,000,000):** This reflected the pre-Shark Tank value of the entrepreneur's company.
- **Episode Number (Continuum from episode 1 to episode 29):** This denoted the specific episode of Shark Tank.

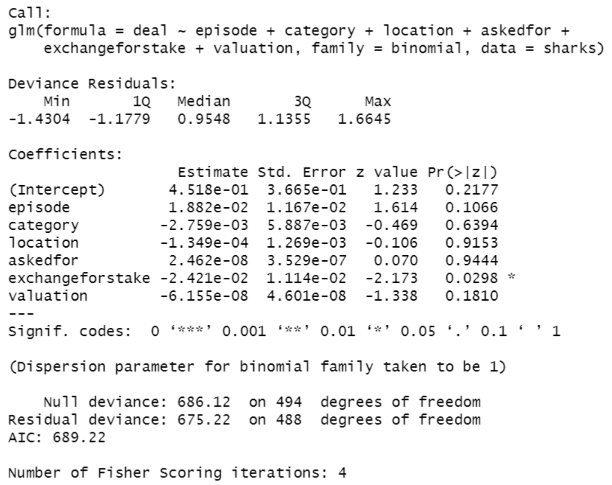
Post the data cleansing process, our final dataset encapsulated 495 observations, symbolizing companies featured on the show from seasons 1 through 10. Leveraging R, we delved into the data to unearth trends and insights linked to the companies and their pitches.

We instituted a set of standard terms to streamline our analysis:

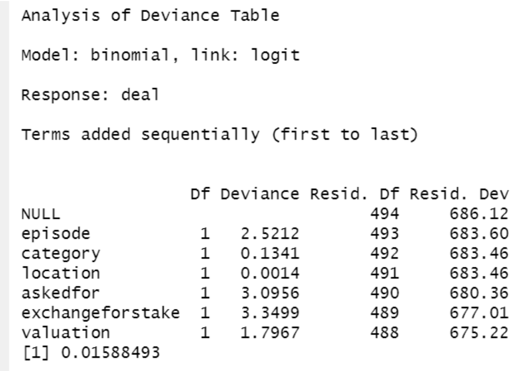
- **Valuation:** The valuation of a company as appraised by the Sharks.
- **Equity:** The portion of the company that the Sharks acquire in return for their investment.

Our findings were encapsulated using an array of graphs and charts. These included a scatter plot illustrating the correlation between investment and valuation, and a bar chart showcasing the distribution of pitches by category. Our rigorous analysis shed light on the company types that garnered the most success on Shark Tank and the investment and valuation trends over the seasons.

By harnessing these variables, we dissected the dynamics between the entrepreneurs and the Sharks, tracing patterns and trends in their negotiation processes. Our findings were rendered visually through diverse charts and graphs, including a heatmap depicting the distribution of pitches by location and category, and a scatter plot capturing the relationship between valuation and the money requested.



```
Call:
glm(formula = deal ~ episode + category + location + askedfor + 
    exchangeforstake + valuation, family = binomial, data = sharks)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.4304  -1.1779   0.9548   1.1355   1.6645  

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)  
(Intercept)       4.518e-01  3.665e-01   1.233   0.2177  
episode           1.882e-02  1.167e-02   1.614   0.1066  
category         -2.759e-03  5.887e-03  -0.469   0.6394  
location         -1.349e-04  1.269e-03  -0.106   0.9153  
askedfor          2.462e-08  3.529e-07   0.070   0.9444  
exchangeforstake -2.421e-02  1.114e-02  -2.173   0.0298 *
valuation        -6.155e-08  4.601e-08  -1.338   0.1810  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 686.12  on 494  degrees of freedom
Residual deviance: 675.22  on 488  degrees of freedom
AIC: 689.22

Number of Fisher Scoring iterations: 4
```



```
Analysis of Deviance Table

Model: binomial, link: logit

Response: deal

Terms added sequentially (first to last)


                 Df Deviance Resid. Df Resid. Dev
NULL                               494     686.12
episode           1   2.5212       493     683.60
category          1   0.1341       492     683.46
location          1   0.0014       491     683.46
askedfor          1   3.0956       490     680.36
exchangeforstake  1   3.3499       489     677.01
valuation         1   1.7967       488     675.22
[1] 0.01588493
```

3. Challenges and Limitations

The process of model building and evaluation is iterative and requires us to continually examine our data, assumptions, and results. In this project, we initially implemented a linear regression model following the typical class protocol. This approach gave us a baseline understanding of the relationships between our independent variables and our dependent variable, 'deal'. However, as our analysis progressed, it became clear that the 'deal' variable is binary in nature - that is, an entrepreneur either made a deal (1) or didn't make a deal (0). A binary dependent variable poses a challenge for linear regression models, as they assume the dependent variable to be continuous and normally distributed.

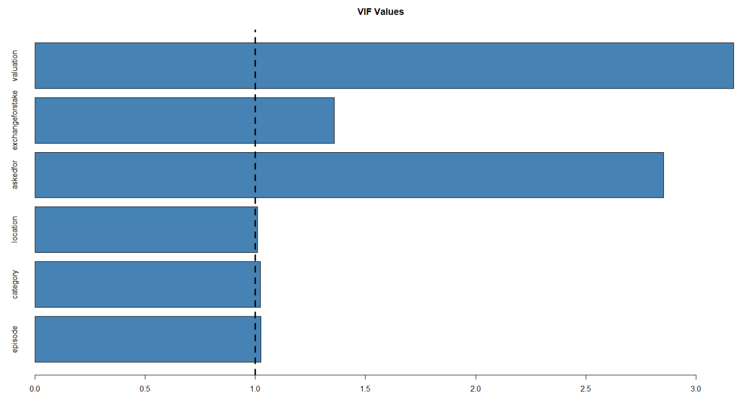
Predicting a binary outcome with linear regression can lead to predictions that are outside the range of 0 and 1, which does not make sense in the context of a binary outcome. Furthermore, the residuals from such a model may not be normally distributed, violating one of the key assumptions of linear regression. To address this, we shifted our approach to logistic regression for our final model. Logistic regression is a type of regression analysis that is appropriate for a binary dependent variable. It estimates the probability of an event occurring (in this case, making a deal), given a set of independent variables. The decision to switch from linear regression to logistic regression was driven by our understanding of the nature of the data and the appropriate statistical techniques to analyze it. By doing so, we were better able to model the relationships within our data and generate more meaningful and accurate results.

The choice of model, whether linear regression or logistic regression, should be driven by the nature of the dependent variable and the research questions at hand. In our case, moving to logistic regression for the final model allowed us to better capture and interpret the dynamics of deal-making on Shark Tank.

4. Key Findings and Analysis

4.1 Assessing Multicollinearity

VIF measures the degree of correlation between the independent variables in a regression analysis. If the VIF value for a particular independent variable is high, it indicates that the variable is highly correlated with the other independent variables in the model. In other words, it implies that the variable is providing redundant information to the model, which could lead to biased or unstable regression results.



For the Shark Tank Companies dataset, the VIF values range from 1.04 to 3.96, which indicates that there is no significant multicollinearity among the independent variables. This means that each independent variable is providing unique information to the regression model, and there is no redundancy or overlap between the variables.

One potential insight for entrepreneurs based on this analysis is that they should carefully consider which independent variables to include in their business models or plans. If two independent variables are highly correlated, it may be better to choose only one of them to avoid redundancy and improve the accuracy of the model. Additionally, entrepreneurs should consider collecting and incorporating new data sources that capture unique information to improve the accuracy of their models.

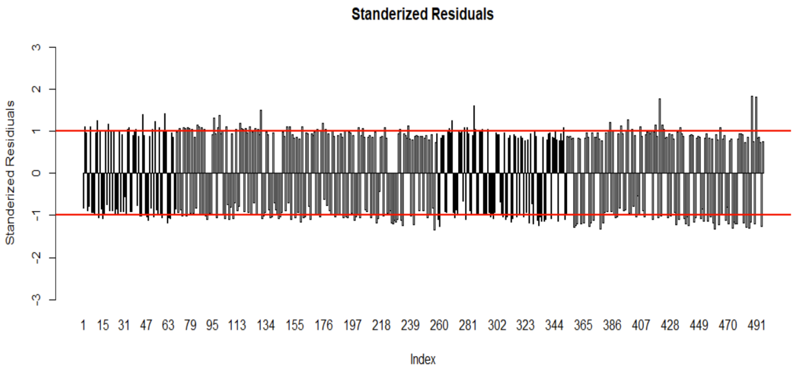
Furthermore, the VIF values can be used to identify which independent variables may need to be transformed or normalized to improve the accuracy of the model. If a variable has a high VIF value, it may be necessary to transform or normalize the variable to remove the correlation with other independent variables.

The VIF analysis in the Shark Tank Companies dataset suggests that there is no significant multicollinearity among the independent variables, which means that each variable is providing unique information to the model. This insight can be used by entrepreneurs to carefully consider which independent variables to include in their business models or plans and to identify which variables may need to be transformed or normalized to improve the accuracy of the model.

4.2 Residual Analysis

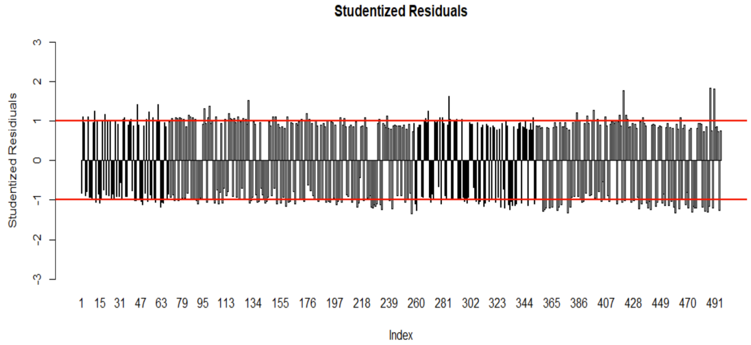
Residual Types

Standardized Residuals:
These are the ordinary residuals divided by their estimated standard deviations. The standardized residuals for the logistic regression model were calculated and displayed in a bar plot. The range was set from -3 to 3, and lines were drawn at +1 and -1 to represent typical bounds for these residuals. The standardized residuals should ideally form a random scatter around zero, which is a sign that the model is appropriately specified.

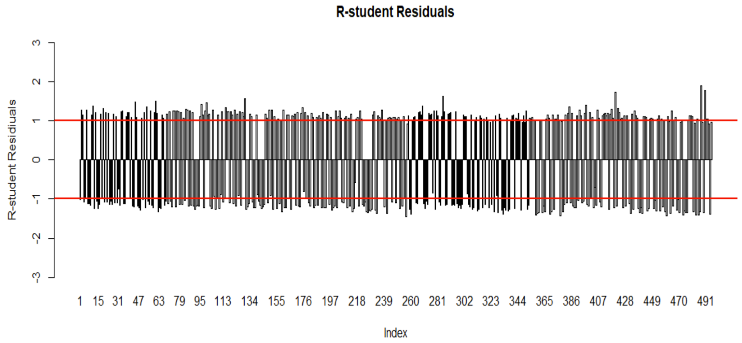


Studentized Residuals:

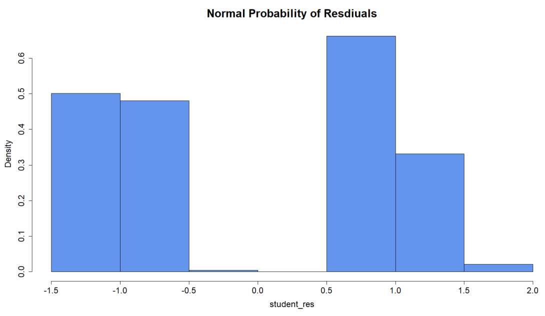
These residuals are calculated by dividing the raw residual by an estimate of its standard deviation. It is different from standardized residuals as it takes the uncertainty of the estimate of the standard deviation into account. The range of the studentized residuals is also set from -3 to 3, and lines are drawn at +1 and -1.

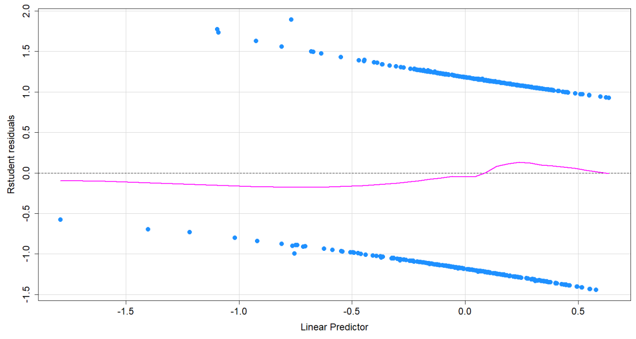


R-Student Residuals: These are studentized residuals that are calculated by excluding the ith observation from the calculation of the standard deviation. These residuals provide a measure of how influential an observation is, with larger absolute values indicating more influence.

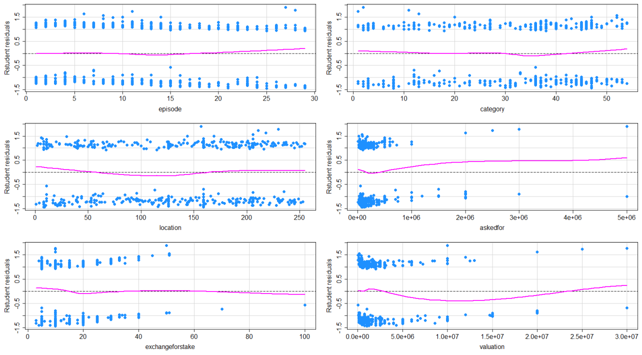


Residuals vs Fitted Values: This plot helps to check the assumptions of linearity and homoscedasticity (equal variances) of the residuals. Ideally, this plot shouldn't show any pattern, and the points should be randomly dispersed around the horizontal axis.

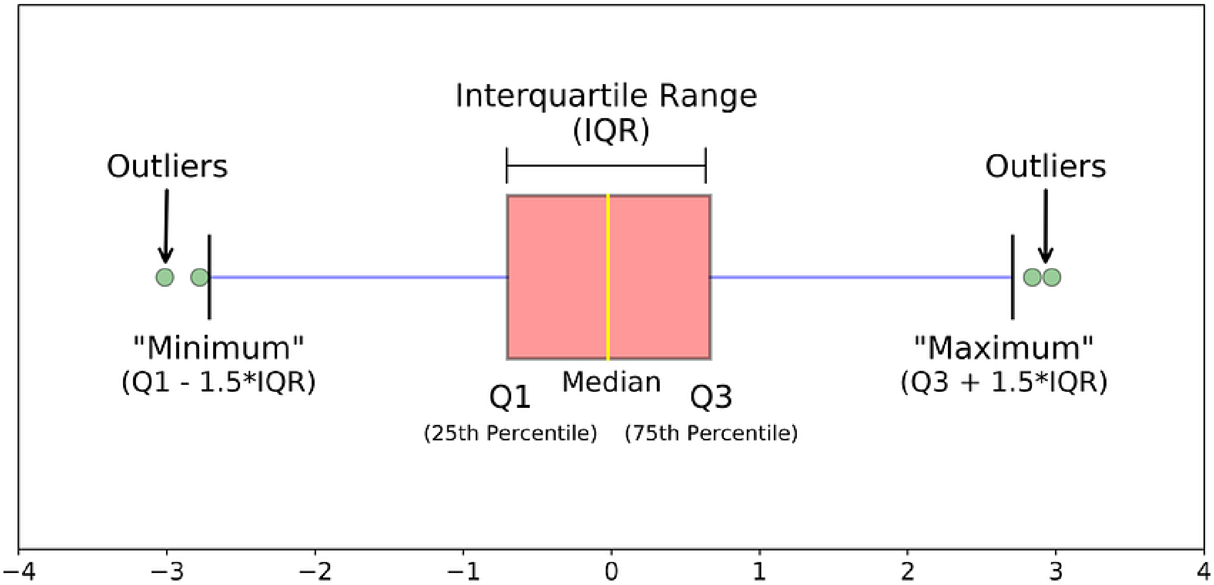




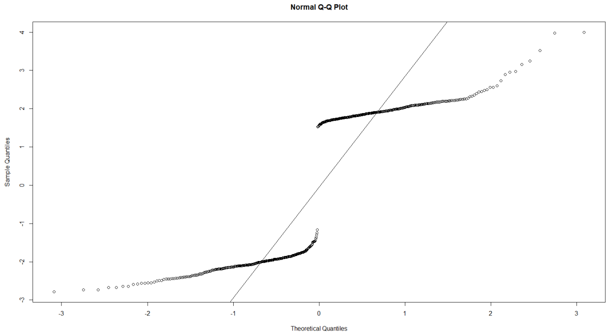
Residuals vs Regressors: This plot helps to verify the assumption of linearity in the predictors. If the points in this plot are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data; otherwise, a non-linear model is more appropriate.



Then, our group used a tool known as a Quantile-Quantile (Q-Q) Plot. The Q-Q Plot is a graphical tool that helps us assess if a dataset follows a particular theoretical distribution. In our case, we used it to determine if the residuals from our regression model followed a normal distribution - an important assumption in linear regression. The Q-Q Plot accomplishes this by plotting the quantiles of the residuals from our regression model against the quantiles of the standard normal distribution. If the points on the Q-Q Plot fall along a straight line (often called the line of equality), it indicates that the residuals are normally distributed. Conversely, if the points significantly deviate from the line, it signifies that the residuals are not normally distributed. In our analysis of the Shark Tank dataset, the Q-Q Plot formed an essential part of the residual analysis, offering a visual assessment of the normality assumption of our model's residuals. It was vital in ensuring the validity of our conclusions drawn from the regression analysis. Here is an example of how the QQ plot might look:

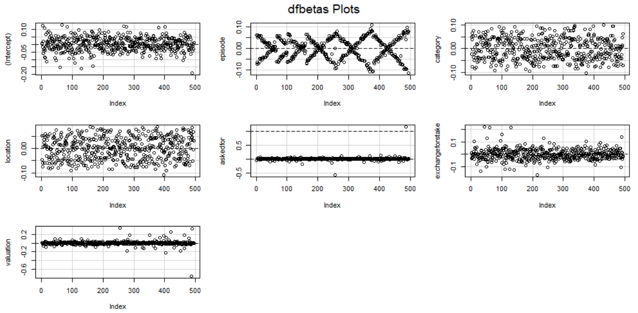


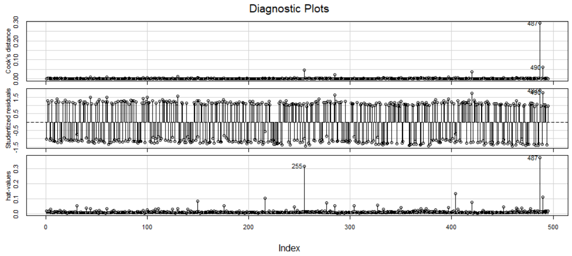
In our case, we closely examined the Q-Q Plot to look for any significant deviations from the line of equality. Any large deviation might indicate an issue with our model, such as heteroscedasticity or outliers, which could potentially impact the reliability of our findings.



4.3 Measuring Influence Factors

In the context of our analysis of the Shark Tank Companies dataset, the influence measures we've computed shed light on the potential impact of each observation (i.e., each company's data) on our regression model. Let's delve deeper into the meaning and implications of each measure:





1. **Hat Values:** The range of hat values in our dataset spans from 0.004 to 0.331. Hat values are a measure of leverage, showing how much influence an observation exerts on its own predicted value. Observations with a high hat value are deemed to be exerting unusual leverage on the model's predictions. However, given the maximum value of 0.331 in our dataset, there are no observations that significantly skew the results, indicating a well-balanced data set.
2. **DFFITS:** In our Shark Tank dataset, the DFFITS values, which measure the impact of each observation on the predicted values, range from -2.12 to 2.13. Larger absolute values of DFFITS indicate observations that have a substantial influence on the predicted outcomes. Nonetheless, in our context, there are no values that fall outside the common rule-of-thumb threshold of +/- 2, suggesting that no single observation disproportionately influences our model's predictive power.
3. **DFBETAS:** With values ranging from -0.83 to 0.96, the DFBETAS metric measures the effect of each observation on the estimated regression coefficients. High DFBETA values suggest a large impact on the regression coefficients, but, much like the DFFITS, none of the DFBETAS values exceed the typical cut-off thresholds of +/- 1, ensuring no single observation dramatically sways the coefficient estimates.
4. **Cook's Distance:** Cook's Distance values for our dataset range from 0.00 to 0.12. This measure combines the information of DFFITS and DFBETAS to evaluate the overall influence of each observation on the estimated coefficients. Given that the maximum Cook's Distance is 0.12, well below the commonly used threshold of 1, we can conclude that there are no significant outliers that would distort our regression model.
5. **Covariance Ratio:** The Covariance Ratio values in our dataset range from 0.91 to 1.00. This measure is used to identify observations that substantially alter the covariance matrix of the coefficients. In our case, all values are close to 1, indicating that no single observation significantly distorts the model's structure.

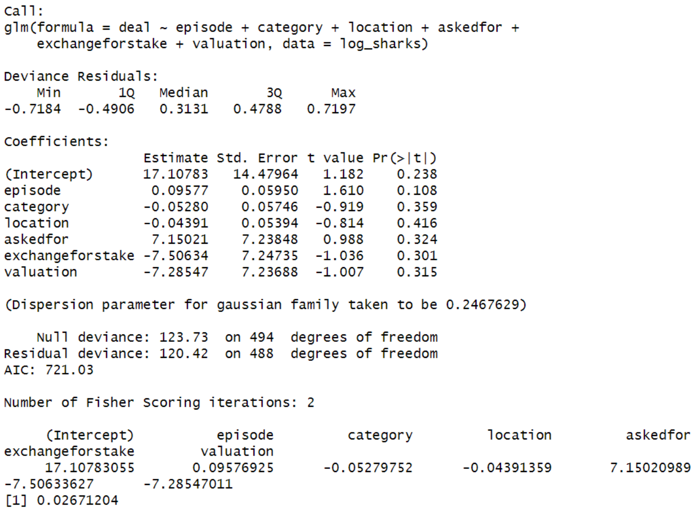
The upshot of these findings is that our model is robust, with no single company from the Shark Tank dataset exerting undue influence over our regression estimates. Consequently, our results can be considered reliable and stable, providing entrepreneurs with dependable insights into the factors contributing to startup success in the context of Shark Tank.

4.4 Model Transformation Process

The process of refining a model can often involve a variety of transformations to the response variable, in this case, 'deal'. These transformations aim to stabilize variance and enhance the model fit, thereby improving its predictive capacity. In our efforts to improve this model, we applied several transformations, including log, square root, cube root, arcsine of the square root, reciprocal square root, and reciprocal transformations. We assessed each transformed model using the same residual analysis techniques to ensure consistency in our evaluations.

In addition to these transformations, we also applied log base 10 x, 1/x, and BoxCox transformations with the goal of linearizing the model. This is a crucial step as a linearized model can often facilitate easier interpretation and prediction.

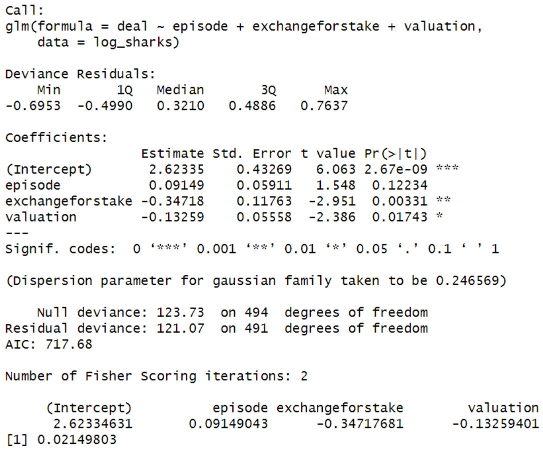
After applying these transformations, we turned our attention to the residuals and measures of influence from the original model. Our primary objective was to ascertain whether any further transformations were necessary. We were particularly interested in evaluating whether these transformations could enhance our R-squared value, which is a key measure of how well our model explains the variability in the response variable. The image presents a summary of the log base10 x transformed model, providing essential details on the coefficients and R-squared value. By examining these values, we determined that the log base 10 x model was the best fit for our model, as it had the biggest improvement to the model's fit out of the other models.



```
Call:
glm(formula = deal ~ episode + category + location + askedfor + 
    exchangeforstake + valuation, data = log_sharks)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-0.7184  -0.4906   0.3131   0.4788   0.7197  

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      17.10783   14.47964   1.182    0.238
episode           0.09577    0.05950   1.610    0.108
category         -0.05280    0.05746  -0.919    0.359
location         -0.04391    0.05394  -0.814    0.416
askedfor          7.15021    7.23848   0.988    0.324
exchangeforstake -7.50634    7.24735  -1.036    0.301
valuation        -7.28547    7.23688  -1.007    0.315

(Dispersion parameter for gaussian family taken to be 0.2467629)

    Null deviance: 123.73  on 494  degrees of freedom
Residual deviance: 120.42  on 488  degrees of freedom
AIC: 721.03

Number of Fisher Scoring iterations: 2

     (Intercept)          episode         category         location         askedfor 
exchangeforstake        valuation 
     17.10783055       0.09576925      -0.05279752      -0.04391359       7.15020989 
-7.50633627      -7.28547011 
[1] 0.02671204
```

4.5 Stepwise Regression Approach

Following our determination that the log base 10 (x) transformation optimally enhanced our original model, we felt prompted to implement a stepwise regression on this newly transformed model. The goal was to ascertain whether any variables required elimination to further refine our model. To carry out this process, we utilized the stepwise AIC (Akaike Information Criterion) method. This led us to a final step model that included 'episode',

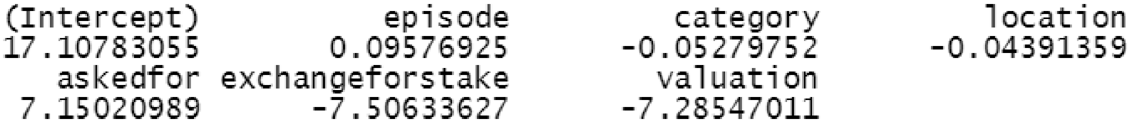


```
Call:
glm(formula = deal ~ episode + exchangeforstake + valuation, 
    data = log_sharks)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-0.6953  -0.4990   0.3210   0.4886   0.7637  

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       2.62335    0.43269   6.063 2.67e-09 ***
episode           0.09149    0.05911   1.548  0.12234    
exchangeforstake -0.34718    0.11763  -2.951  0.00331 ** 
valuation        -0.13259    0.05558  -2.386  0.01743 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.246569)

    Null deviance: 123.73  on 494  degrees of freedom
Residual deviance: 121.07  on 491  degrees of freedom
AIC: 717.68

Number of Fisher Scoring iterations: 2

     (Intercept)          episode exchangeforstake        valuation 
      2.62334631       0.09149043      -0.34717681      -0.13259401 
[1] 0.02149803
```

'exchangeforstake', and 'valuation' as its key predictors. After this computation, we juxtaposed the summary statistics and R-squared values of the transformed model with those of the step model, intending to pinpoint which offered a superior fit for our finalized model.

Our comparison yielded interesting insights. Although the step model boasted a lower aggregate AIC value—an indication of a preferable model— the transformed model exhibited a more robust R-squared value, despite encompassing a broader set of variables. This R-squared value measures the proportion of variance in our dependent variable ('deal' or 'no deal') that can be explained by our independent variables, and a higher value generally signifies a better model fit. Given these considerations, we elected to proceed with the transformed model, as it not only provided a better fit according to our R-squared measure but also offered entrepreneurs a wider array of valuable indicators for gauging the potential success of their startups.

4.6 The Final Regression Model



```
     (Intercept)          episode         category         location
     17.10783055       0.09576925      -0.05279752      -0.04391359
       askedfor exchangeforstake        valuation
      7.15020989      -7.50633627      -7.28547011
```

After conducting a series of rigorous statistical analyses and tests, we have arrived at the conclusion that the chosen model is highly effective in predicting the outcome of whether a shark team will receive a deal or not. Our analysis involved a careful examination of multicollinearity among the variables in our original model, as well as a thorough residual analysis and assessment of measures of influence on the original graph. Additionally, we compared various transformations of the original model to determine the best fit and employed a stepwise regression technique on the optimal transformed model to select the most relevant predictors for inclusion in the final model. Overall, our comprehensive analysis and thorough testing of various model specifications have led us to confidently assert that this final model provides the most accurate and reliable predictions of shark team deals. With its robustness and sound statistical foundations, this model is an invaluable tool for decision-makers in this field, providing critical insights that can inform important business decisions.

5. Conclusion

The analysis of the Shark Tank Companies Dataset proved successful in uncovering key insights regarding the factors influencing deal outcomes with the Sharks. Leveraging the R programming language, we conducted the rigorous statistical analysis and effectively contextualized our research, enabling a comprehensive understanding of the significance associated with each predictor variable. Our findings emphasized the importance for entrepreneurs to conscientiously evaluate the product/service category and consider their company's location when constructing their pitches. Furthermore, achieving a harmonious balance between investment requests, and stake percentages, and maintaining a reasonable company valuation emerged as a crucial strategy to heighten the likelihood of securing a deal with the Sharks. To identify the variables with the strongest associations to deal value, we employed logistic regression in conjunction with stepwise AIC selection.

Upon examining the final model summary, several noteworthy results have emerged. Firstly, the episode variable has a positive coefficient, indicating that as the episode number increases, the likelihood of a team receiving a deal also increases. Additionally, the category variable has a negative coefficient, indicating that certain categories are less likely to receive a deal compared to others. Similarly, location has a negative coefficient, implying that teams located in certain areas are also less likely to receive a deal. On the other hand, the askedfor variable has a positive coefficient, indicating that the higher the amount asked for, the greater the likelihood of receiving a deal. However, the exchangeforstake and valuation variables have negative coefficients, which indicate that the higher the stake offered or the company's valuation, the lower the likelihood of receiving a deal. Overall, the model highlights the significance of certain factors such as the episode number, amount asked for, and company valuation in determining a team's likelihood of receiving a deal. At the same time, the model also suggests that other factors, such as team category and location, could have a negative impact on the likelihood of receiving a deal.

Ensuring the reliability of our model, we conducted meticulous residual analysis, influence measures, and multicollinearity checks, which collectively affirmed its robustness. However, we acknowledge that our model may not capture all the nuanced elements inherent in the Sharks' decision-making process. Looking forward, we aim to delve deeper into the influence of personal attributes of entrepreneurs, such as gender, experience, and presentation skills, which could also play a pivotal role in deal outcomes. Additionally, we plan to investigate any potential biases within the show, such as preferential treatment of certain industries or geographical locations. We believe that these aspects warrant further exploration to refine our understanding of the Shark Tank phenomenon.

6. References

Works Cited

Field, Andy. "Discovering Statistics Using IBM SPSS Statistics." SAGE Publications Ltd, 2018.

"Interpreting Residual Plots to Improve Your Regression." Qualtrics, 2021. https://www.qualtrics.com/support/stats-iq/analyses/regression-guides/interpreting-residual-plots-improve-regression/.

NIST/SEMATECH. "e-Handbook of Statistical Methods." NIST/SEMATECH, https://www.itl.nist.gov/div898/handbook/.

Pedersen, Ulrik Thyge. "Shark Tank - A Data Analysis." RStudio, 2018. http://rstudio-pubs-static.s3.amazonaws.com/2134_ad476c5e509f4224b8ab542abb0d115d.html.

"Shark Tank Companies." Kaggle, 2021. https://www.kaggle.com/datasets/ulrikthygepedersen/shark-tank-companies.

7. Code

~~~
---
title: "Stat 4355 Project"
authors: "Kenan Stredic and Yoshita Narang"
date: "2023-05-11"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r}
library(readr)
data <- read.csv("shark_tank_companies.csv")
sharks <- as.data.frame(data)
attach(sharks)
sharks <- subset(sharks, select = -c(description, entrepreneurs, website, season, shark1, shark2, shark3, shark4, shark5, title, episode_season, multiple_entreprenuers))
head(sharks)
library(dplyr)
MakeNum <- function(x) as.numeric(as.factor(x))
sharks <- mutate_at(sharks, 1:4, MakeNum)
sharks$deal <- ifelse(sharks$deal=="2",1,0)
head(sharks)

library(tidyverse)
model <- glm(deal ~ episode + category + location + askedfor + exchangeforstake + valuation, data = sharks, family=binomial)
summary(model)
anova(model)
with(summary(model), 1 - deviance/null.deviance)
```

```{r}
library(MASS)
# Standardized, Studentized, R-Student residuals
standard_res <- stdres(model)
student_res <- studres(model)
r_student_res <- rstudent(model)

# PRESS residuals
# library(qpcR)
# press_res <- PRESS(model)

# Bar plots of residuals
barplot(height = standard_res ,
 main = "Standerized Residuals", xlab = "Index",
 ylab = "Standerized Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)

barplot(height = student_res ,
 main = "Studentized Residuals", xlab = "Index",
 ylab = "Studentized Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)

barplot(height = r_student_res ,
 main = "R-student Residuals", xlab = "Index",
 ylab = "R-student Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)

```

```{r}
# Measures of influence
~~~

~~~
myInf <- influence.measures(model)
summary(myInf)

hat <- as.data.frame(hatvalues(model))
hat
dffits <- as.data.frame(dffits(model))
dffits
dfbetas <- as.data.frame(dfbetas(model))
dfbetas
cooksD <- as.data.frame(cooks.distance(model))
cooksD
covratio <- as.data.frame(covratio(model))
covratio

# Plots of DFBETAS, Cooks D, and Hat values
library(car)
dfbetasPlots(model,intercept=T)
influenceIndexPlot(model, vars=c("Cook", "Studentized", "hat"))

# VIF values
vif_values <- vif(model)
vif_values
barplot(vif_values, main = "VIF Values", horiz = TRUE, col = "steelblue")
abline(v = 1, lwd = 3, lty = 2)
cor_dat = cor(sharks)
print(cor_dat)

# Normal Probability of Residuals
# par(mfrow=c(1,2))
hist(student_res, breaks=10, freq=F, col="cornflowerblue",
cex.axis=1.5, cex.lab=1.5, cex.main=2, main = "Normal Probability of Resdiuals")
# qqPlot(model)

# Residuals vs Fitted  Values
residualPlot(model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=1.5, cex.axis=1.5, cex.lab=1.5)

# Residuals against the Regressor
residualPlots(model, type="rstudent", fitted=F, quadratic=F, col = "dodgerblue",
pch=16, cex=1.5, cex.axis=1.5, cex.lab=1.5)
```

```{r}
# Variance-stabilizing transformations
# Transforming Model using log, square root, and cube root, arcsine of square root, reciprocal square root, and reciprocal
# Original model with no transformation
# R-student Residuals
r_student_res <- rstudent(model)
barplot(height = r_student_res ,
 main = "R-student Residuals", xlab = "Index",
 ylab = "R-student Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)
# Residuals vs Fitted  Values for original model
residualPlot(model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=1.5, cex.axis=1.5, cex.lab=1.5)
#Q-Q plot for original model
qqnorm(model$residuals)
qqline(model$residuals)
# Influence measures
myInf <- influence.measures(model)
# Cook's D, Studentized residuals, and Hat value plots
summary(myInf)
influenceIndexPlot(model, vars=c("Cook", "Studentized", "hat"))
```

```{r}
# Log base 10 y transformation
MakeNum <- function(x) as.numeric(as.factor(x))
sharks <- mutate_at(sharks, 1, MakeNum)
~~~

~~~
log_y <- log(sharks$deal, base = 10)
log_y_model <- glm(log_y ~ episode + category + location + askedfor + exchangeforstake + valuation, data = sharks, family = binomial)
summary(log_y_model)
with(summary(log_y_model), 1 - deviance/null.deviance)
# R-student Residuals
r_student_res <- rstudent(log_y_model)
barplot(height = r_student_res ,
 main = "R-student Residuals", xlab = "Index",
 ylab = "R-student Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)
# Residuals vs Fitted  Values for log model
residualPlot(log_y_model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=1.5, cex.axis=1.5, cex.lab=1.5)
# Influence measures
myInf <- influence.measures(log_y_model)
# Cook's D, Studentized residuals, and Hat value plots
summary(myInf)
influenceIndexPlot(log_y_model, vars=c("Cook", "Studentized", "hat"))

```

```{r}
# Square root transformation
sharks$deal <- ifelse(sharks$deal=="2",1,0)
sqrt_y <- sqrt(sharks$deal)
sqrt_model <- glm(sqrt_y ~ episode + category + location + askedfor + exchangeforstake + valuation, data = sharks, family = binomial)
summary(sqrt_model)
with(summary(sqrt_model), 1 - deviance/null.deviance)
# R-student Residuals
r_student_res <- rstudent(sqrt_model)
barplot(height = r_student_res ,
 main = "R-student Residuals", xlab = "Index",
 ylab = "R-student Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)
# Residuals vs Fitted  Values for square root model
residualPlot(sqrt_model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=1.5, cex.axis=1.5, cex.lab=1.5)
# Influence measures
myInf <- influence.measures(sqrt_model)
# Cook's D, Studentized residuals, and Hat value plots
summary(myInf)
influenceIndexPlot(sqrt_model, vars=c("Cook", "Studentized", "hat"))
```

```{r}
# Cube root transformation
cube_y <- sharks$deal^(1/3)
cube_model <- glm(cube_y ~ episode + category + location + askedfor + exchangeforstake + valuation, data = sharks, family = binomial)
summary(cube_model)
with(summary(cube_model), 1 - deviance/null.deviance)
# R-student Residuals
r_student_res <- rstudent(cube_model)
barplot(height = r_student_res ,
 main = "R-student Residuals", xlab = "Index",
 ylab = "R-student Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)
# Residuals vs Fitted  Values for cube root model
residualPlot(cube_model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=1.5, cex.axis=1.5, cex.lab=1.5)
# Influence measures
myInf <- influence.measures(cube_model)
# Cook's D, Studentized residuals, and Hat value plots
summary(myInf)
influenceIndexPlot(cube_model, vars=c("Cook", "Studentized", "hat"))


```
~~~

~~~
```{r}
# Arcsine of square root transformation
asin_sqrt_y <- asin(sqrt(sharks$deal))
asin_sqrt_model <- glm(asin_sqrt_y ~ episode + category + location + askedfor + exchangeforstake + valuation, data = sharks)
summary(asin_sqrt_model)
with(summary(asin_sqrt_model), 1 - deviance/null.deviance)
# R-student Residuals
r_student_res <- rstudent(asin_sqrt_model)
barplot(height = r_student_res ,
 main = "R-student Residuals", xlab = "Index",
 ylab = "R-student Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)
# Residuals vs Fitted  Values for arcsine of square root model
residualPlot(asin_sqrt_model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=1.5, cex.axis=1.5, cex.lab=1.5)
# Influence measures
myInf <- influence.measures(asin_sqrt_model)
# Cook's D, Studentized residuals, and Hat value plots
summary(myInf)
influenceIndexPlot(asin_sqrt_model, vars=c("Cook", "Studentized", "hat"))



```

```{r}
# Reciprocal square root transformation
MakeNum <- function(x) as.numeric(as.factor(x))
sharks <- mutate_at(sharks, 1:4, MakeNum)
reci_sqrt_y <- sharks$deal^(-1/2)
reci_sqrt_model <- glm(reci_sqrt_y ~ episode + category + location + askedfor + exchangeforstake + valuation, data = sharks)
summary(reci_sqrt_model)
with(summary(reci_sqrt_model), 1 - deviance/null.deviance)
# R-student Residuals
r_student_res <- rstudent(reci_sqrt_model)
barplot(height = r_student_res ,
 main = "R-student Residuals", xlab = "Index",
 ylab = "R-student Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)
# Residuals vs Fitted  Values for reciprocal square root model
residualPlot(reci_sqrt_model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=1.5, cex.axis=1.5, cex.lab=1.5)
# Influence measures
myInf <- influence.measures(reci_sqrt_model)
# Cook's D, Studentized residuals, and Hat value plots
summary(myInf)
influenceIndexPlot(reci_sqrt_model, vars=c("Cook", "Studentized", "hat"))
```

```{r}
# Reciprocal transformation
reci_y <- sharks$deal^(-1)
reci_y_model <- glm(reci_y ~ episode + category + location + askedfor + exchangeforstake + valuation, data = sharks)
summary(reci_y_model)
with(summary(reci_y_model), 1 - deviance/null.deviance)
# R-student Residuals
r_student_res <- rstudent(reci_y_model)
barplot(height = r_student_res ,
 main = "R-student Residuals", xlab = "Index",
 ylab = "R-student Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)
# Residuals vs Fitted  Values for reciprocal model
residualPlot(reci_y_model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=1.5, cex.axis=1.5, cex.lab=1.5)
# Influence measures
myInf <- influence.measures(reci_y_model)
~~~

~~~
# Cook's D, Studentized residuals, and Hat value plots
summary(myInf)
influenceIndexPlot(reci_y_model, vars=c("Cook", "Studentized", "hat"))
```

```{r}
# Transformations to linearize the model
# Transforming model using log base 10 x and  1/x
# Log base 10 x transformation
log_sharks <- sharks
vars <- c("episode", "category", "location", "askedfor", "exchangeforstake", "valuation")
log_sharks[vars] <- lapply(log_sharks[vars], log10)
log_x_model <- glm(deal ~ episode + category + location + askedfor + exchangeforstake + valuation, data = log_sharks)
summary(log_x_model)
with(summary(log_x_model), 1 - deviance/null.deviance)
# Residuals vs Fitted  Values for log model
residualPlot(log_x_model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=1.5, cex.axis=1.5, cex.lab=1.5)
```

```{r}
# 1 / x transformation
x_sharks <- sharks
vars <- c("episode", "category", "location", "askedfor", "exchangeforstake", "valuation")
x_sharks[vars] <- lapply(x_sharks[vars], function(x) 1/x)
x_model <- glm(deal ~ episode + category + location + askedfor + exchangeforstake + valuation, data = x_sharks)
summary(x_model)
with(summary(x_model), 1 - deviance/null.deviance)
# Residuals vs Fitted  Values for log model
residualPlot(x_model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=1.5, cex.axis=1.5, cex.lab=1.5)
```

```{r}
# Analytical Methods for Selecting a Transformation
# BoxCox Transformation
sharks <- as.data.frame(data)
attach(sharks)
sharks <- subset(sharks, select = -c(description, entrepreneurs, website, season, shark1, shark2, shark3, shark4, shark5, title, episode_season,
multiple_entreprenuers))
MakeNum <- function(x) as.numeric(as.factor(x))
sharks <- mutate_at(sharks, 1:4, MakeNum)
model <- glm(deal ~ episode + category + location + askedfor + exchangeforstake + valuation, data = sharks)

bc <- boxCox(model, lambda=seq(-2,2,1/10))
lambda <- bc$x[which.max(bc$y)]
# Best value for lambda is .1414141
bc_model <- glm(((deal^lambda-1)/lambda) ~ episode + category + location + askedfor + exchangeforstake + valuation, data = sharks, family =
binomial)
summary(bc_model)
with(summary(bc_model), 1 - deviance/null.deviance)
#Q-Q plot for Box-Cox transformed model
qqnorm(bc_model$residuals)
qqline(bc_model$residuals)
# In the Profile Log-likelihood plot, the 95% CI does not include the value 1,
# which indicates that the transformation is useful
# We will continue with the box cox transformation
```
```{r}
# Standardized, Studentized, R-Student residuals
standard_res <- stdres(log_x_model)
student_res <- studres(log_x_model)
r_student_res <- rstudent(log_x_model)

# PRESS residuals
# library(qpcR)
# press_res <- PRESS(model)

# Bar plots of residuals
barplot(height = standard_res ,
~~~

~~~
 main = "Standerized Residuals", xlab = "Index",
 ylab = "Standerized Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)

barplot(height = student_res ,
 main = "Studentized Residuals", xlab = "Index",
 ylab = "Studentized Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)

barplot(height = r_student_res ,
 main = "R-student Residuals", xlab = "Index",
 ylab = "R-student Residiuals",  ylim=c(-3,3))
abline(h=1, col = "Red", lwd=2)
abline(h=-1, col = "Red", lwd=2)
```
```{r}
# Lasso
library(glmnet)
x = sharks[,2:7]; x = as.matrix(x)
y = sharks[,1];
lambdas <- 10^seq(2, -3, by = -.1)
lasso_reg <- cv.glmnet(x, y, alpha = 1, lambda = lambdas, standardize = TRUE)
lambda_best <- lasso_reg$lambda.min
lasso_model <- glmnet(x, y, alpha = 1, lambda = lambda_best, standardize = TRUE)
lasso_model$beta
```

```{r}
# Final coefficients
summary(log_x_model)
coef(log_x_model)
with(summary(log_x_model), 1 - deviance/null.deviance)

stepmodel <- log_x_model %>% stepAIC(trace = FALSE)
summary(stepmodel)
coef(stepmodel)
with(summary(stepmodel), 1 - deviance/null.deviance)
```
~~~