Data Analysis on Crowdfunding

Key to Raise Fund and Success

Tingrui Huang

Abstract

Kickstarter is a well-known online crowdfunding platform that allows everyone to create their own projects and raise funds. To date, Kickstarted has received billions of pledges from over 15 million backers to fund over 400,000 projects. However, not all of the projects on Kickstarter could successfully raise funds. In the 300,000 projects, nearly half of all projects have failed to reach their goals and the winners are only take 36.58% of all projects. For fundraisers, the two most important thing they want to know is first whether they could successfully raise funds on Kickstarter, second, how many funds they are able to raise. For investors and backers, they care more about whether the project will succeed. My interest is in helping both fundraisers and investors to identify the key factors of successfully raising funds on Kickstarter.

In the data analysis, I build multiple models to analyze the relationships between two dependent variables – pledged amount and state of projects and various independent variables such as category, country, goal amount, year of launch, duration of the project and number of backers. The models that are used for analyzing these relationships include classic linear regression, logistic regression, multilevel models and random forest model. The results show that the number of backers, goal amount and duration of the project have effects on the success of the projects as well as the pledged amount.

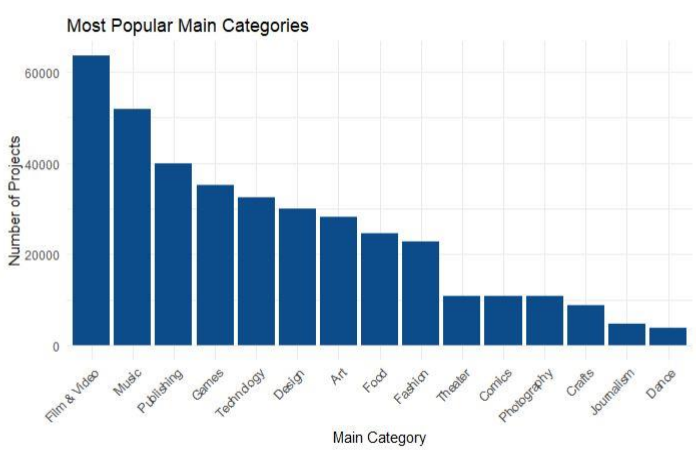
Introduction

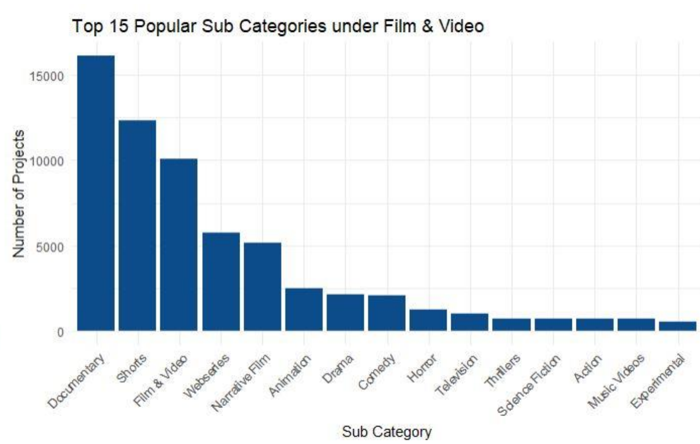
I. How it works

As it has been mentioned previously, Kickstarter is a crowdfunding platform, anyone from any country with any ideas can create their own projects on Kickstarter. However, Kickstarter has the “all-or-nothing” rule, which means if by the end of a project, the raised amount does not reach the goal amount, the project owner will get NOTHING, and all the backers will not be charged for failed projects.

II. More about Kickstarter

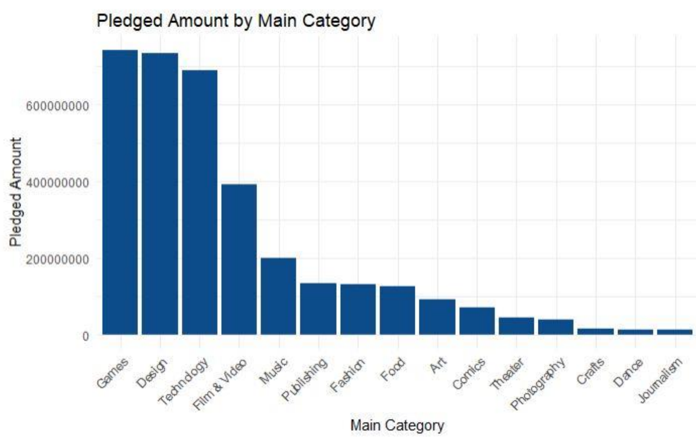
As one of the largest crowdfunding platform in the world, Kickstarter has attracted an enormous amount of people to create projects in various categories, from fashion to technology and from food to arts. Among these categories, the film and video is the most popular category overall time.

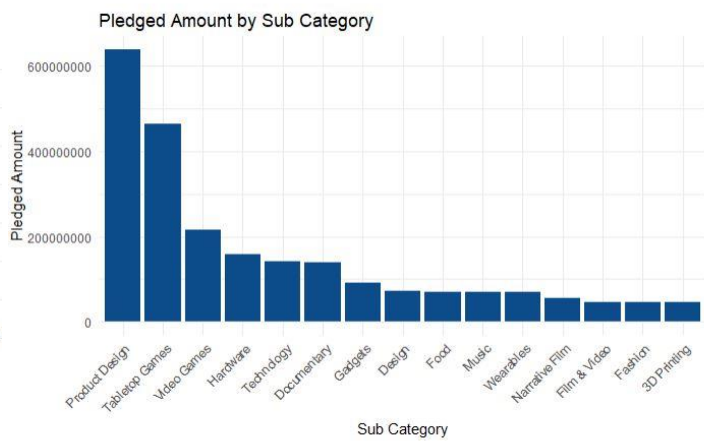




Within the film and video category, there are over 30 subcategories and the top 15 most popular sub-categories include documentary, shorts, webseries and so on.

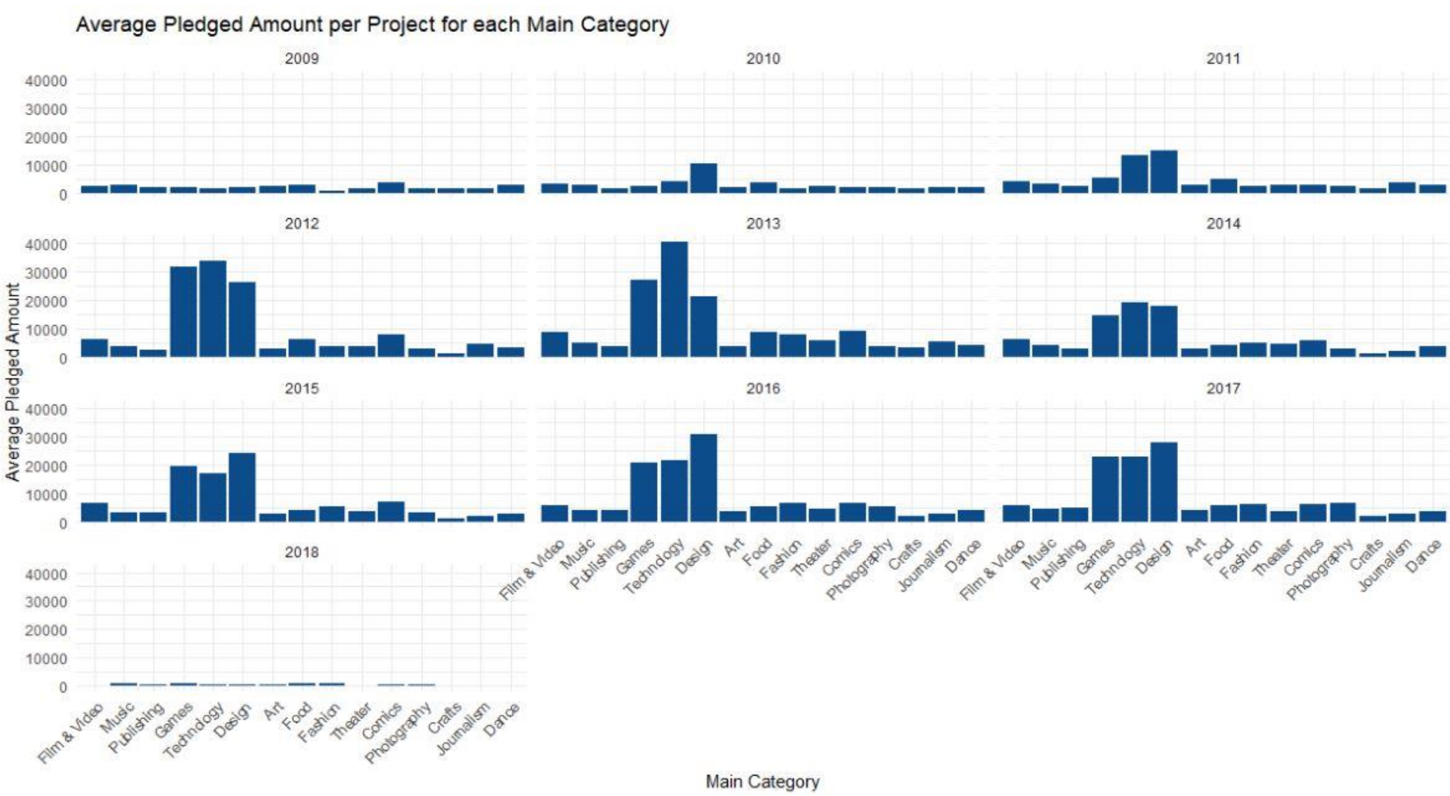
If we look at the categories by total pledged amount, we will find that games, design and technology surpass film and video become the winners in money collection.



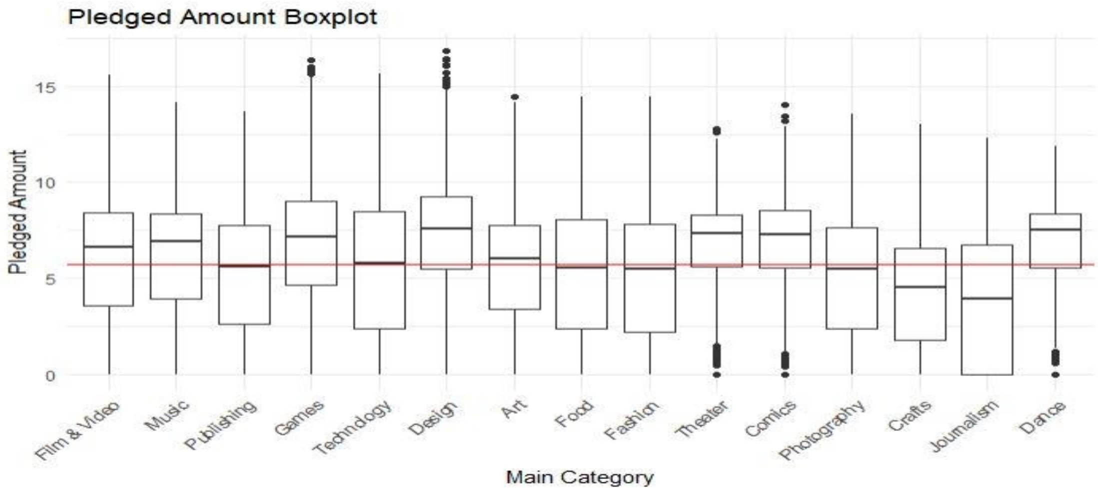


And surprisingly, product design and tabletop games have the highest and second highest pledged amount and they doubled amount of the third place holder which is video game. Technology, one of the hottest topic in today's world is only ranked fifth with significant less pledged amount comparing to the top 2.

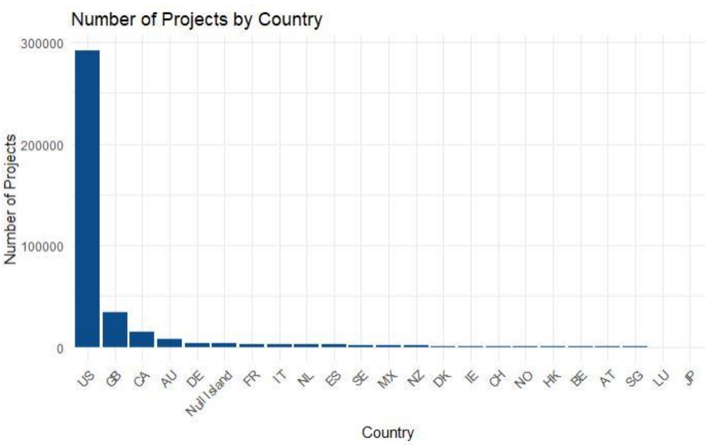
However, the total amount may not tell the whole story, let us look at the average pledged amount per project in each category.

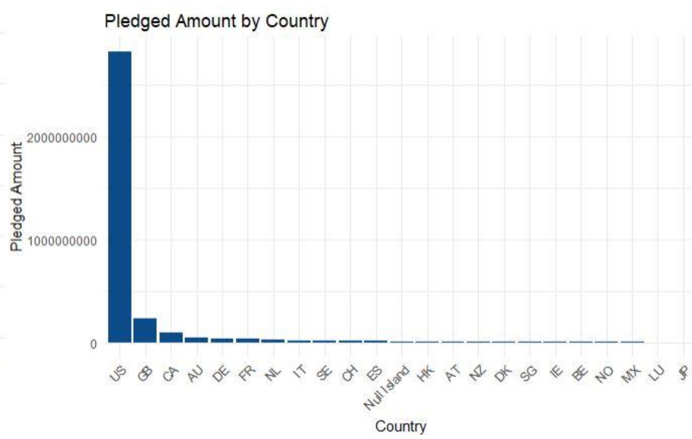


As we can see from the graph, design, games and technology seem to have higher average pledged amount from 2009 to 2017. Design and games are still the winners from this perspective. The boxplot below tells us more about the story. Film, game, technology and design have projects that pledged super high amount(over 3 million). The pledged amount of most crafts and journalism projects fell below the overall average pledged amount, so entrepreneurs and investors should be careful when investing in these categories.

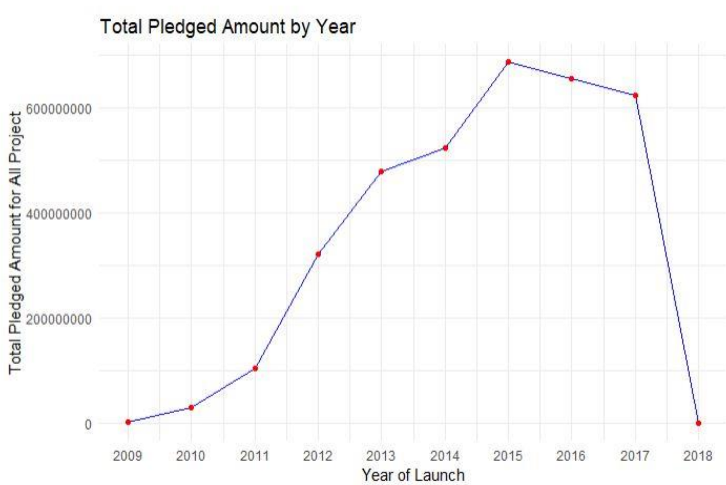


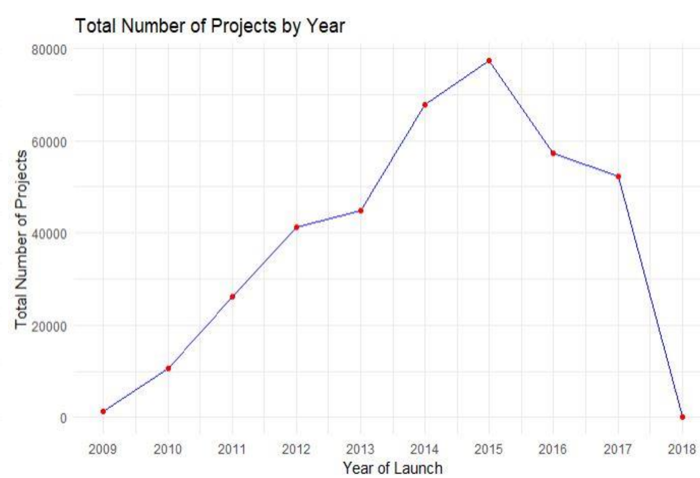
Kickstarter not only attracts talents in all fields, but also gather talents around all over the world. Projects on Kickstarter are from more than 20 countries, including the US, the UK, Canada, Australia and so on. Most projects are still based in the US.



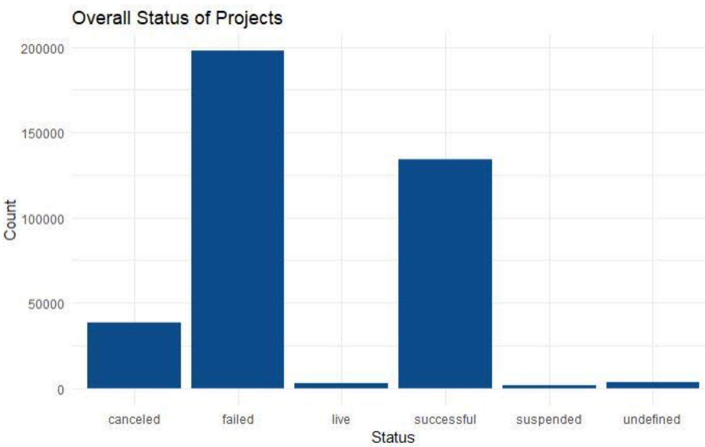


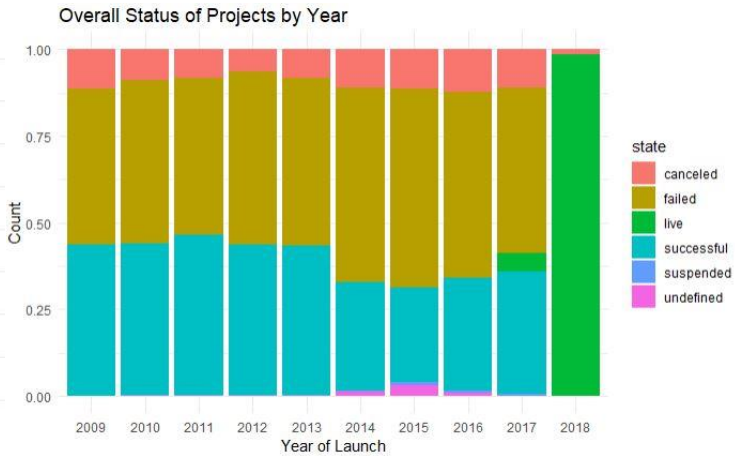
Having been found for more than 9 years, the pledged amount and number of projects on Kickstarter has been keeping growing until 2015. In 2016 and 2017, there seems to be some declining. This could due to more competitors in the market, such as Indiegogo, another major crowdfunding platform in the market. This could also be caused by more restricted requirement of publishing projects on Kickstarter or people’s interests in crowdfunding is decreasing.



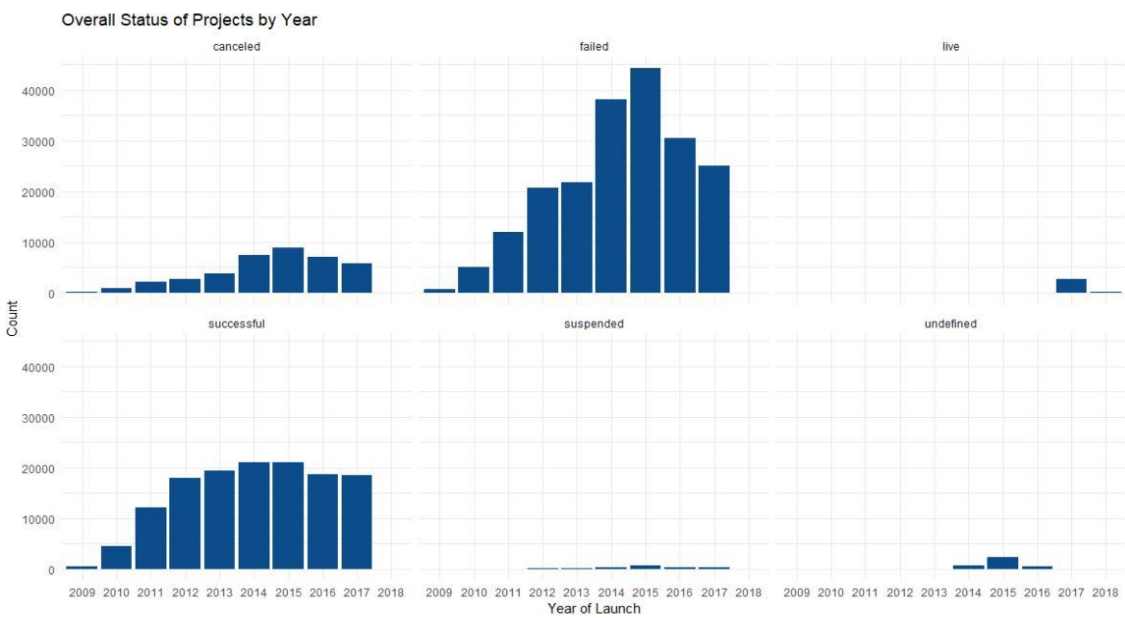


As I have mentioned previously, more than half of the projects on Kickstarter are failed to raise their goal amount, and according to their latest official announcement, only 36.58% projects have successfully raised their target amount.

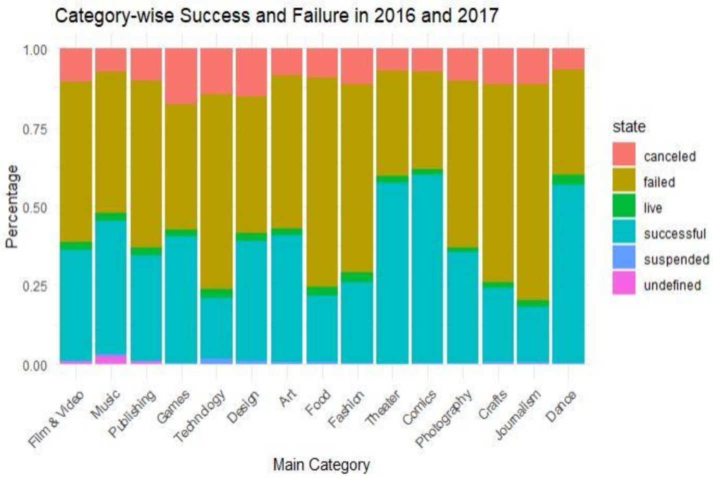


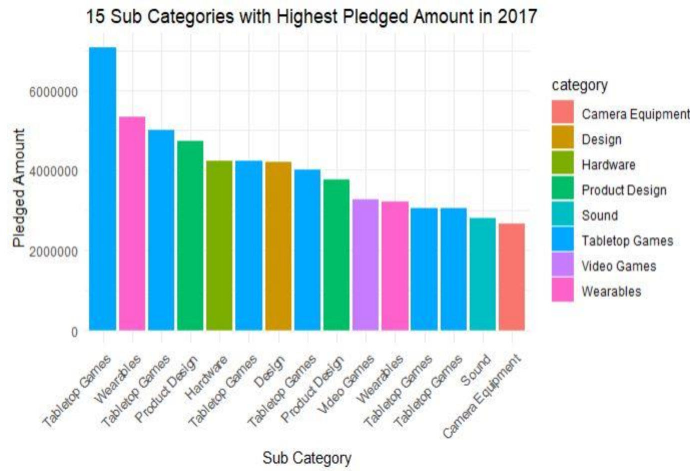


But we are happy to see that the number of failed projects are decreasing for each year from the following graph. As we have seen previously, in 2016, the number of projects was decreased compared to 2015, that trend is also showed in this graph by having less failed, success, canceled and projects. However, we can see the number of failed projects has decreased much more than the number of successful projects, I would say that is a potential indication on the higher quality of projects in 2016 comparing to previous years.



Finally, let us look at the data in the recent two years. In 2016 and 2017, theatre, comics and dance have the highest success rate. Meanwhile, surprisingly, technology is among one of the highest fail rate categories, nearly more than 75% projects under technology were failed to raise funds during 2016 to 2017, the success rate is only a bit higher than Journalism. And tabletop games is the hottest and the winner of money harvester in 2017.





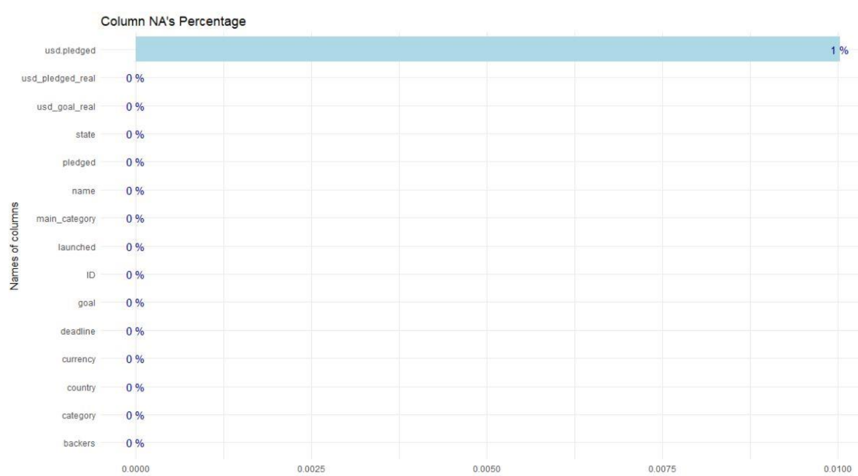
Materials and Methods

I. Dataset

The dataset is sourced from Kaggle. The dataset contains 378,661 observations with 15 variables. The dataset was updated in Jan 2018 and it is the latest version. The 15 variables include: ID, name, category, main_category, currency, deadline, goal, launched, pledged, state, backers, country, USD pledged, USD_pledged_real, USD_goal_real.

II. Data Preparation

By doing a routine check on missing values, we can see the only variable contains missing value is usd_pleddged, and it only has 1% missing value. Meanwhile, since I am not going to use this variable, instead I will use the usd_pledged_real, therefore I don't think the missing value is a big deal in this dataset.



Other data cleaning steps include:

1) Reformat data variables and split year and month from combinations.

2) Remove observations that Year of Launch is 1970. Kickstarter was found in 2009, any projects that is launched before that could be filed with wrong information.

3) Some of the projects that has reached their goal amount, however in the dataset they were classified as “failed”, therefore, change the state of these projects to “successful”.

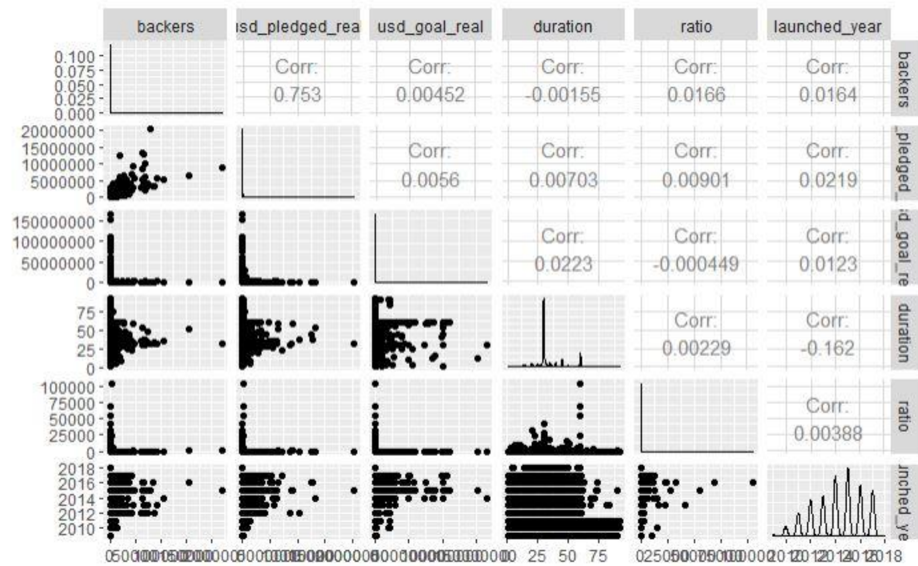
4) Create a duration variable to store the duration of each project

5) Create a indicator to indicate whether a project is successful or failed.

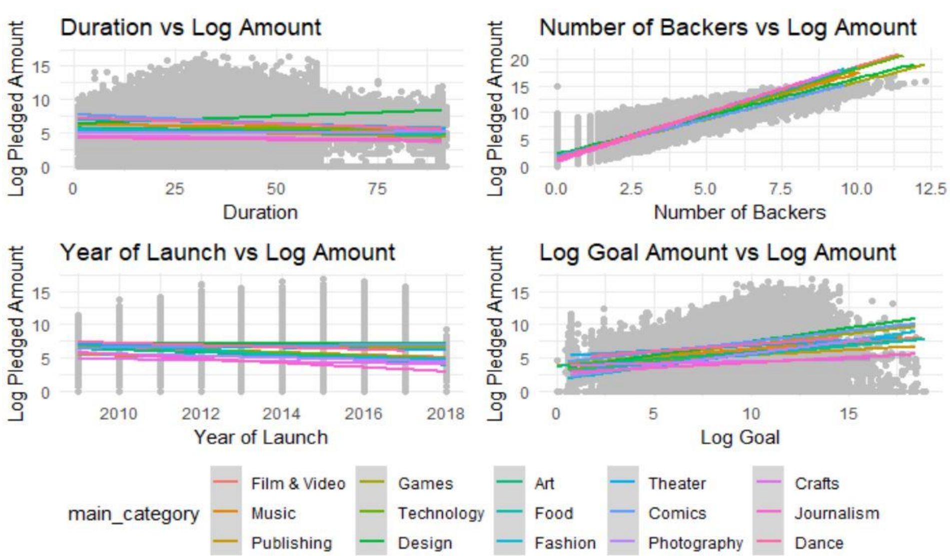
6) Scale continuous variables such as duration, number of backers and goal amount.

III. Variable Selection

After performing the data preparation, the dataset now contains 19 variables. My assumption is that, the pledged amount will be affected by continuous variables such as number of backers, duration, goal amount, year of launch; and it will also be affected be categorical variables such as category, main category and country.



As we can see in the correlation test, number of backers have relatively higher correlation with pledged amount, while other variables have relatively small correlation with pledged amount. Let us take a closer look at the relationships between pledged amount and each continuous variable.



From the graphs above, I would say except for number of backers, the goal amount seems to have correlation with pledged amount as well. The relationship between duration, year of launch and pledged amount seem to be weak.

Based on the previous visualization, I select the following variables for analysis:

Outcome Variable	Fixed Effects	Random Effects
usd_pledged_amount	backers	category
state	duration	main_category
	usd_goal_real	country
	launched_year	

IV. Statistical Modeling

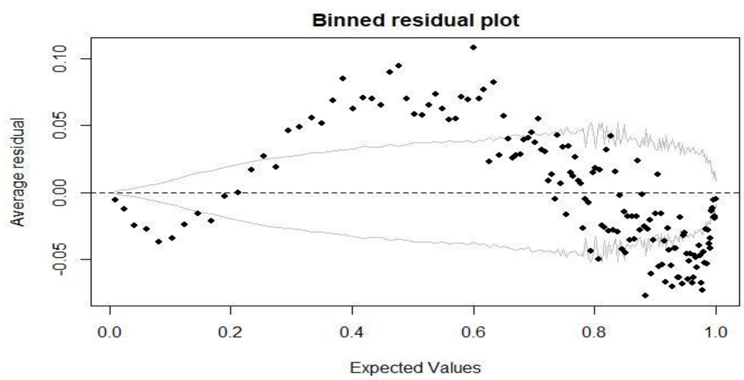
(i) Logistic Regression

In order to find out the factors that have effects on the state of a project, I choose to fit a logistic regression with state as the dependent variables and backers, duration, goal amount, launched year, main category and country as independent variables.

- Model 1: Model with only one variable - backers

$$y_{state} = -4.706 + 1.415X_{log(backers)}$$

```
Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)     -4.706455   0.027603  -170.5   <2e-16 ***
log(backers + 1) 1.415545   0.008106   174.6   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 156896  on 116354  degrees of freedom
Residual deviance:  82263  on 116353  degrees of freedom
AIC: 82267
```

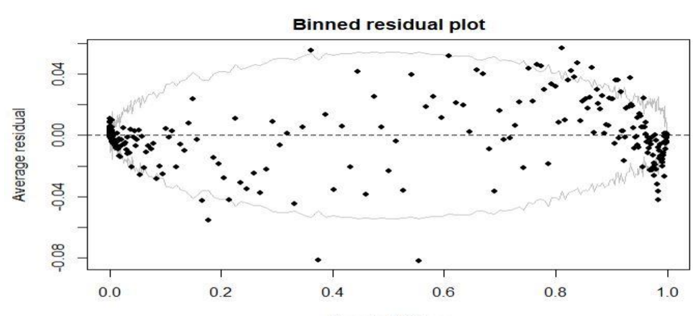


As we can see from the result table, the deviance is decreased by a huge amount. I would say the number of backers could have a large influence on the state of projects. On the right side, from the residual plot, we can see some obvious trend in the plot and almost half of the residuals fall outside the bin. I would add more predictors to see if the issue of the trend and outliers would be fixed.

- Model 2: Add duration and logarithm goal amount into the previous model

$$y_{state} = 5.305 + 2.931X_{log(backers)} - 0.031X_{duration.c} - 1.848X_{log(Goal Amount)}$$

```
Coefficients:
                      Estimate Std. Error  z value Pr(>|z|)
(Intercept)            5.30566    0.07512   70.631   <2e-16 ***
log(backers + 1)       2.93190    0.02024  144.832   <2e-16 ***
duration.c            -0.03159    0.01257   -2.514   0.0119 *
log(usd_goal_real + 1) -1.84873   0.01488 -124.283   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 156896  on 116354  degrees of freedom
Residual deviance:  44586  on 116351  degrees of freedom
AIC: 44594
```



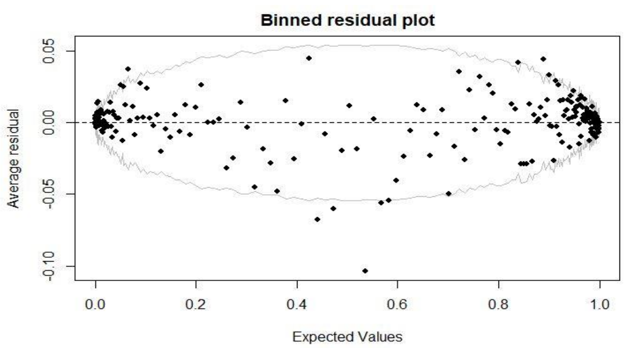
After adding two more predictors we now have a complete pooling model, and the residual deviance is decreased again from 82263 to 44586. Based on this result, I think it is reasonable to include these two predictors in the model. In the residual plot, I see the trend in the previous residual plot has been mitigated, however, there is still a slight trend in the plot. Another thing that is worth mentioning is that the coefficient of intercept is a bit larger than 5, and that may indicate some potential issues under logarithm format.

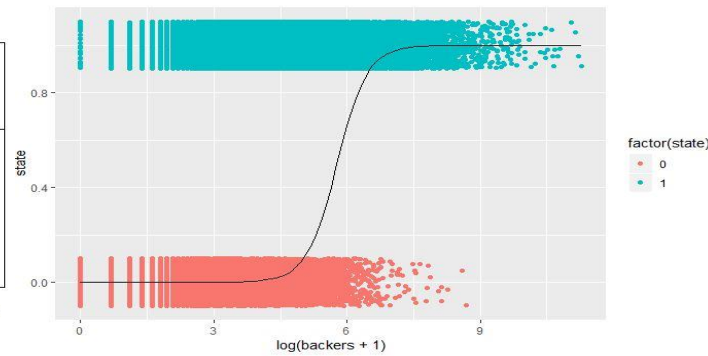
- *Model 3: Add group variables and build a no pooling model*

$$y_{state} = -42.948 + 3.204X_{log(backers)} - 0.004X_{duration.c} - 1.982X_{log(GoalAmount)} + 0.024X_{LaunchedYear} + \beta_{MainCategory}X_{MainCategory} + \beta_{country}X_{country}$$

```
Coefficients:
                     Estimate Std. Error  z value Pr(>|z|)
(Intercept)        -42.948700  14.864616   -2.889  0.00386 **
log(backers + 1)     3.204333   0.023001  139.313  < 2e-16 ***
duration            -0.004335   0.001082   -4.006 6.18e-05 ***
log(usd_goal_real + 1) -1.982097 0.016683 -118.813 < 2e-16 ***
main_categoryComics -1.415255   0.085281  -16.595  < 2e-16 ***
main_categoryCrafts -0.906400   0.099312   -9.127  < 2e-16 ***
```

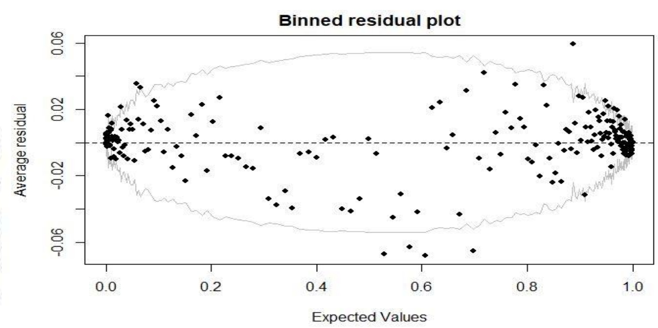
```
countryNull Island        10.461107   0.489102   21.388  < 2e-16 ***
countryNZ                 -0.311762   0.466574   -0.668  0.50401
countrySE                 -0.422492   0.454270   -0.930  0.35235
countrySG                 -0.742764   0.536909   -1.383  0.16654
countryUS                 -0.033022   0.410752   -0.080  0.93592
as.numeric(launched_year)  0.024228   0.007369    3.288  0.00101 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 156896  on 116354  degrees of freedom
Residual deviance:  39643  on 116314  degrees of freedom
AIC: 39725
```





In this no pooling model I found some uncommon values for coefficients, such as the coefficient of intercept and coefficient of one country level. The residual plot indicates there could be overpredicting in the model. Based on these findings, I would remove the "Null Island" from the country list. Meanwhile, the residual deviance is decreased by nearly 5000, and it possibly indicates that the model has been improved by adding those variables.

```
Coefficients:
                       Estimate Std. Error  z value Pr(>|z|)
(Intercept)          -44.208105  14.882657   -2.970 0.002974 **
log(backers + 1)       3.210348   0.023069  139.163  < 2e-16 ***
duration              -0.004211   0.001085   -3.883 0.000103 ***
log(usd_goal_real + 1) -1.987204  0.016736 -118.739  < 2e-16 ***
main_categoryComics   -1.418573   0.085365  -16.618  < 2e-16 ***
main_categoryCrafts   -0.908249   0.099406   -9.137  < 2e-16 ***
```



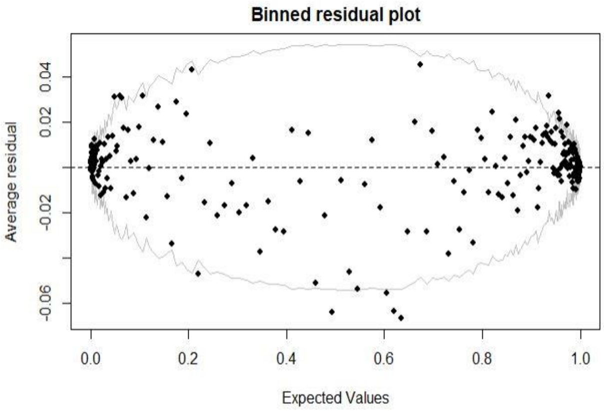
After removing the "Null Island", the coefficient of intercept is still very larger, and it's even larger than the previous -42.948. The residual plot looks very similar and the trend has become more clear. Therefore, I made some transformations to see if that could help fix the issue.

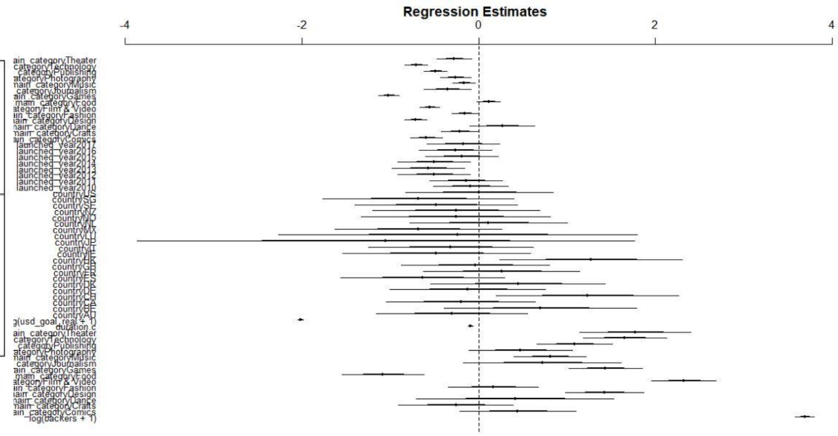
- *Model 4: Add interactions to see if that helps reduce residual deviance*

$$y_{state} = 4.773 + 3.691X_{log(backers)} - 0.09X_{duration.c} - 2.013X_{log(GoalAmount)} + \beta_{LaunchedYear}X_{LaunchedYear} + \beta_{MainCategory}X_{MainCategory} + \beta_{country}X_{country} + \beta_{Backer:Category}X_{Backer:Category}$$

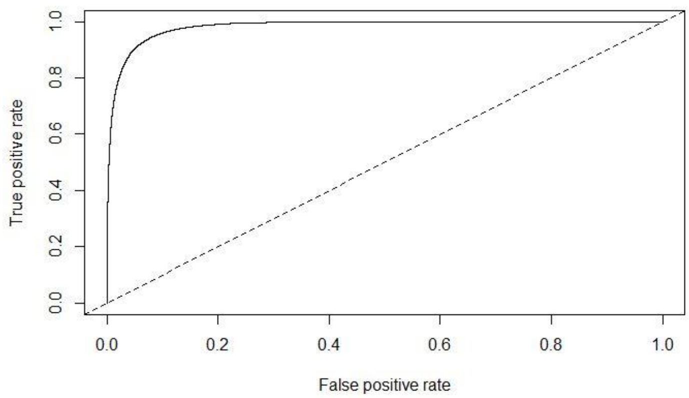
```
Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)          4.773306   0.491107   9.719  < 2e-16 ***
log(backers + 1)     3.691485   0.053867  68.530  < 2e-16 ***
main_categoryComics  0.442162   0.327752   1.349  0.17731
main_categoryCrafts -0.259185   0.323284  -0.802  0.42271
main_categoryDance   0.411383   0.560127   0.734  0.46268
main_categoryDesign  1.423831   0.222417   6.402 1.54e-10 ***
```

```
log(backers + 1):main_categoryTechnology  -0.706496  0.064313  -10.985  < 2e-16 ***
log(backers + 1):main_categoryTheater     -0.277135  0.098002   -2.828  0.00469 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 156774  on 116269  degrees of freedom
Residual deviance:  38567  on 116209  degrees of freedom
AIC: 38689
```





By accident, I didn't convert "launched_year" to numeric, and that unintentionally reduces the coefficient of intercept from -44 to positive 4.77. By adding the interaction, the residual deviance is reduced by 1000. From the coefficient plot we see most of the country levels are across 0, and that could be caused by small amount of observations from countries other than the US and the UK. I hope by using a multilevel model could help fix this issue.



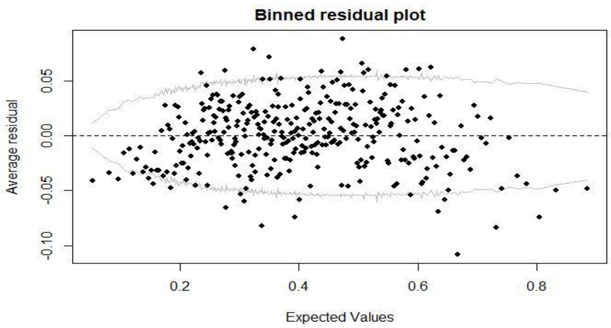

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 27752  1435
         1  2009 18744

               Accuracy : 0.931
                 95% CI : (0.9288, 0.9332)
    No Information Rate : 0.5959
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8575
 Mcnemar's Test P-Value : < 2.2e-16
```

By looking at the ROC curve and the confusion matrix I would say the model fit the data pretty good. However, when making the prediction, I also realize that in practice, fundraisers wouldn't know how many backers they would have at the very beginning of the fundraising. I come up with two possible solutions. First, when using the current model to make a prediction, we could use the average number of backers in each sub-category. Second, building another model without number of backers as a predictor.

By using the first solution, which is using average number of backers in each sub-category to make prediction, I get the following prediction accuracy. The accuracy is decreased from 93% to 55%. The model is making more Type II errors.

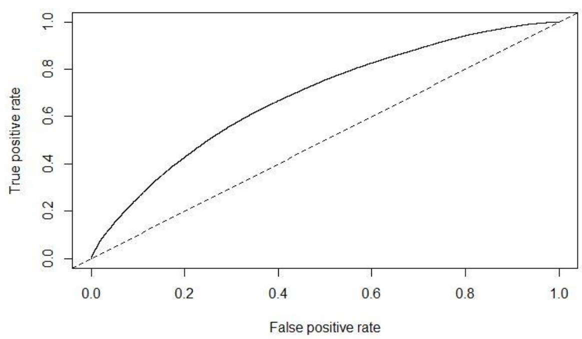
```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0  9920  2454
         1 19841 17725

               Accuracy : 0.5536
                 95% CI : (0.5492, 0.5579)
    No Information Rate : 0.5959
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1859
 Mcnemar's Test P-Value : <2e-16
```

By using the second solution, which is refitting another model similar to model 4 but exclude number of backers, I get following results.




```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 156774  on 116269  degrees of freedom
Residual deviance: 144620  on 116217  degrees of freedom
AIC: 144726
```

The residual deviance is increased by huge amount, recall from previous model, the residual deviance is 38,567, while in this model the residual deviance is 144,726. The residual plot is also getting worse by removing number of backers from the model.




```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 24342 12068
         1  5419  8111

               Accuracy : 0.6498
                 95% CI : (0.6456, 0.654)
    No Information Rate : 0.5959
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2322
 Mcnemar's Test P-Value : < 2.2e-16
```

The accuracy is higher than the accuracy by using the first solution, however, the ROC curve is much worse than previous one. For prediction purpose, I would remove the number of backers and refit the model. While for finding the relationships between dependent and independent variables, I would keep the number of backers in the model.

```
Analysis of Deviance Table

Model 1: state ~ log(backers + 1)
Model 2: state ~ log(backers + 1) + duration.c + log(usd_goal_real + 1)
Model 3: state ~ log(backers + 1) + duration + log(usd_goal_real + 1) +
    main_category + country + as.numeric(launched_year)
Model 4: state ~ log(backers + 1) + main_category + log(backers + 1):main_category +
    duration.c + log(usd_goal_real + 1) + country + launched_year
Model 5: state ~ main_category + duration + log(usd_goal_real + 1) + main_category:log(usd_goal_real +
    1) + country + as.numeric(launched_year)
  Resid. Df Resid. Dev Df Deviance
1    116268      81846
2    116266      43681  2    38165
3    116230      39481 36     4201
4    116209      38567 21      913
5    116217     144620 -8  -106053
```

From the ANOVA test, I would say model 4 is the winner of all logistic models.

(ii) Multilevel Logistic Regression

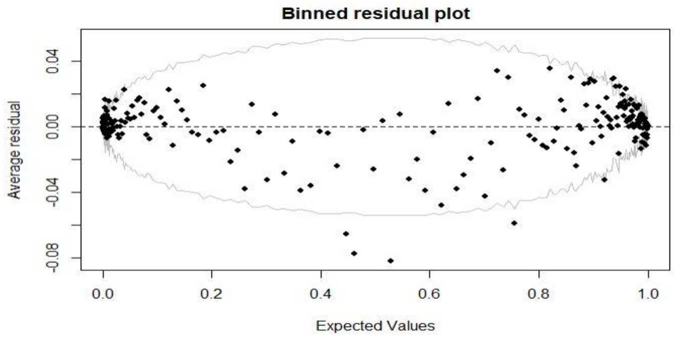
From previous analysis, I believe a partial pooling model could be a better choice for analyzing the dataset, since the outcomes of projects in same category and country may have correlations. Based on these thoughts, I add two groups, country and category.

- *Model 6: Add nested groups main_category/category as a random effect*

$$y_{state} \sim N(5.48 + 3.263X_{log(Backers)} - 0.044X_{duration.c} - 2.019X_{GoalAmount} + 0.034X_{LaunchedYear} + u_{j[i]} + w_{j[k]}, \sigma_y^2)$$

$$u_{j[i]} \sim N(0, 0.76^2), w_{j[k]} \sim N(0, 0.52^2)$$

```
     AIC       BIC    logLik deviance df.resid
 39027.1  39094.8 -19506.6  39013.1   116263

Scaled residuals:
    Min      1Q  Median      3Q     Max
-36.267  -0.057  -0.002   0.145  90.663

Random effects:
 Groups                 Name        Variance Std.Dev.
 category:main_category (Intercept) 0.2797   0.5289
 main_category          (Intercept) 0.5865   0.7658
Number of obs: 116270, groups:  category:main_category, 170; main_category, 15

Fixed effects:
                                 Estimate Std. Error  z value Pr(>|z|)
(Intercept)                       5.48080    0.22135   24.761  < 2e-16 ***
log(backers + 1)                  3.26361    0.02372  137.586  < 2e-16 ***
duration.c                       -0.04445    0.01399   -3.179  0.00148 **
log(usd_goal_real + 1)           -2.01950    0.01724 -117.141  < 2e-16 ***
scale(as.numeric(launched_year))  0.03453    0.01487    2.322  0.02021 *
```



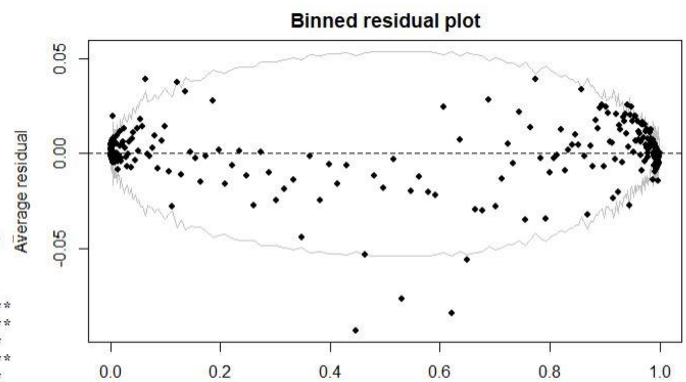
As we can see in the result table, the AIC of this multilevel model is similar to the model 4 which is the no pooling model, and the residual plot is fairly closed to previous residual plot. Most residuals between 0.2 and 0.8 are below the horizontal axis. This could indicate some problem in the model or the dataset.

- *Model 7: Add another un-nested group country as a random effect*

$$y_{state} \sim N(5.458+3.258X_{log(Backers)}-0.045X_{duration.c}-2.012X_{GoalAmount}+ 0.041X_{LaunchedYear}+u_{j[i]}+w_{j[k]}+o_{n[i]},\sigma_y^2)$$

$$u_{j[i]} \sim N(0,0.77^2), w_{j[k]} \sim N(0,0.52^2), o_{n[i]}(0,0.41^2)$$

```
     AIC       BIC    logLik deviance df.resid
 39106.6  39183.9 -19545.3  39090.6   116262

Scaled residuals:
    Min      1Q  Median      3Q     Max
-36.408  -0.057  -0.002   0.146  88.633

Random effects:
 Groups                 Name        Variance Std.Dev.
 category:main_category (Intercept) 0.2763   0.5256
 country                (Intercept) 0.1754   0.4189
 main_category          (Intercept) 0.5965   0.7724
Number of obs: 116270, groups:  category:main_category, 170; country, 22;

Fixed effects:
                                 Estimate Std. Error  z value Pr(>|z|)
(Intercept)                       5.45802    0.24675   22.120  < 2e-16 ***
log(backers + 1)                  3.25808    0.02369  137.524  < 2e-16 ***
duration.c                       -0.04512    0.01401   -3.222  0.00127 **
log(usd_goal_real)               -2.01285    0.01721 -116.941  < 2e-16 ***
scale(as.numeric(launched_year))  0.04194    0.01554    2.698  0.00698 **
```

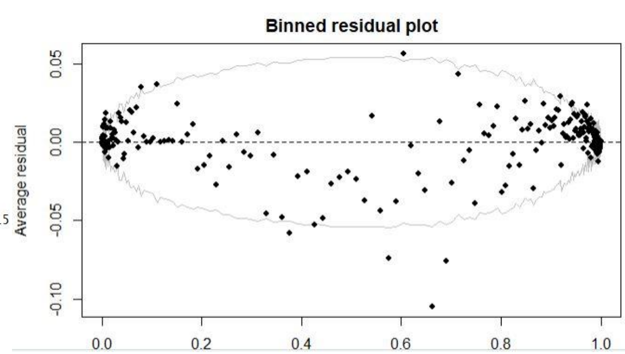


Surprisingly, by adding another random effect, the AIC has been increased, and the variance of the previous random effects only decreased a little. All of these evidence has proved that country does not help to reduce deviance and the explain uncertainty in the previous model.

- *Model 8: Remove country and vary slope by adding number of backers*

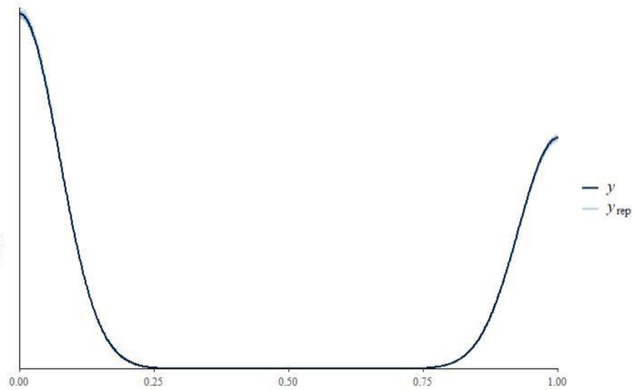
$$y_{state} \sim N(5.23+\beta_{j[i]}X_{log(Backers)}-0.05X_{duration.c}-2.044X_{GoalAmount}+ 0.037X_{LaunchedYear}+u_{j[i]}+w_{j[k]},\sigma_y^2)$$

```
     AIC       BIC    logLik deviance df.resid
 38440.4  38546.7 -19209.2  38418.4   116259

Scaled residuals:
    Min      1Q  Median      3Q     Max
-47.969  -0.055  -0.002   0.139  89.194

Random effects:
 Groups                 Name             Variance Std.Dev. Corr
 category:main_category (Intercept)      0.53622  0.7323
                        log(backers + 1) 0.05537  0.2353   -0.80
 main_category          (Intercept)      0.57050  0.7553
                        log(backers + 1) 0.06699  0.2588   -0.57
Number of obs: 116270, groups:  category:main_category, 170; main_category, 15

Fixed effects:
                                 Estimate Std. Error  z value Pr(>|z|)
(Intercept)                       5.23088    0.23137   22.609  < 2e-16 ***
log(backers + 1)                  3.42841    0.07696   44.548  < 2e-16 ***
duration.c                       -0.05001    0.01420   -3.521  0.00043 ***
log(usd_goal_real)               -2.04467    0.01780 -114.878  < 2e-16 ***
scale(as.numeric(launched_year))  0.03792    0.01501    2.527  0.01149 *
```



As we can see from the result table, AIC and deviance has been deceased for more than 500, and there is strong correlation between number of backers and category, therefore, I would say this model is better than the model 6 which does not vary slope.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 27816  1417
         1  1945 18762

               Accuracy : 0.9327
                 95% CI : (0.9304, 0.9349)
    No Information Rate : 0.5959
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8608
 Mcnemar's Test P-Value : < 2.2e-16
```



By doing the cross-validation, the model 8 has an accuracy at 93%. Again, I am using the actual number of backers to get this high accuracy, if we use the average number of backers, the accuracy will be decreased to 55%. By looking at the posterior check, I would say the model fit the data pretty good.

```
mllo3: state ~ log(backers + 1) + duration.c + log(usd_goal_real + 1) +
mllo3:     scale(as.numeric(launched_year)) + (1 | main_category/category)
mllo5: state ~ log(backers + 1) + duration.c + log(usd_goal_real) +
mllo5:     (1 | main_category/category) + (1 | country) + scale(as.numeric(launched_year))
mllo6: state ~ log(backers + 1) + duration.c + log(usd_goal_real) +
mllo6:     (1 + log(backers + 1) | main_category/category) + scale(as.numeric(launched_year))
lor8: state ~ main_category + duration + log(usd_goal_real + 1) + main_category:log(usd_goal_real +
lor8:     1) + country + as.numeric(launched_year)
lor7: state ~ log(backers + 1) + main_category + log(backers + 1):main_category +
lor7:     duration.c + log(usd_goal_real + 1) + country + launched_year
      Df    AIC    BIC logLik deviance     Chisq Chi Df Pr(>Chisq)
mllo3  7  39027  39095 -19507    39013
mllo5  8  39107  39184 -19545    39091      0.00      1          1
mllo6 11  38440  38547 -19209    38418    672.14      3     <2e-16 ***
lor8  53 144726 145238 -72310   144620      0.00     42          1
lor7  61  38689  39279 -19284    38567 106052.59      8     <2e-16 ***
```

By doing the ANOVA test, we can see model 8(mllo6) has overall better performance than other models.

(iii) Classic Linear Regression

In order to find out the key fund raising drivers, I decided to build multiple linear regression models to see the relationships between pledged amount and other independent variables.

Similar to the procedure in building logistic models, I build model from scratch. I will skip the first couple linear models and talk more about the final model.

- Model 1: no pooling model

$$y_{log(PledgedAmount)} = 277.1 + 0.00065X_{backers} - 0.019X_{duration} + \beta_{MainCategory}X_{MainCategory} + \beta_{country}X_{country} - 0.135X_{LaunchedYear} + 0.2578X_{log(GoalAmount))}$$

- Model 2: Transform continuous variables, logarithm and scale

$$y_{log(PledgedAmount)} = 66.32 + 1.632X_{log(backers)} + 0.007X_{duration.c} + \beta_{MainCategory}X_{MainCategory} + \beta_{country}X_{country} - 0.032X_{LaunchedYear} + 0.053X_{log(GoalAmount))}$$

- Model 3: Add interactions

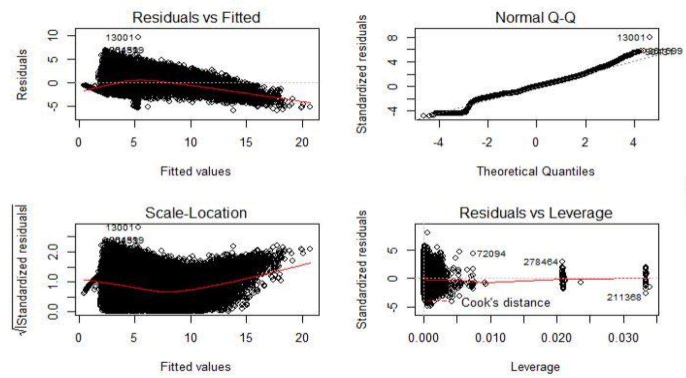
$$y_{log(PledgedAmount)} = 46.64 + 1.719X_{log(backers)} + 0.026X_{duration.c} + \beta_{MainCategory}X_{MainCategory} + \beta_{country}X_{country} - 0.022X_{LaunchedYear} - 0.059X_{log(GoalAmount))} + \beta_{MainCategory:log(backers)}X_{MainCategory:log(backers)}$$

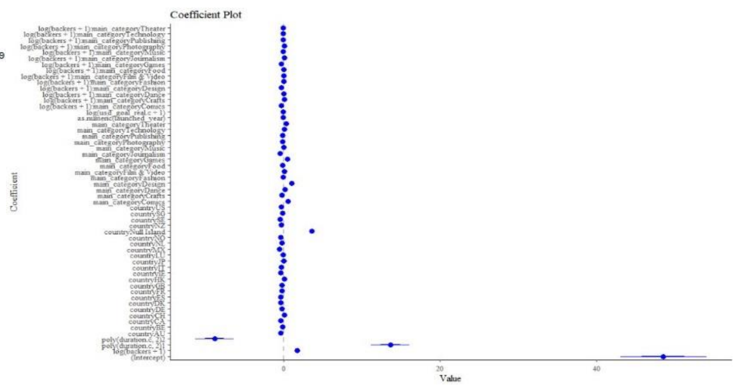
- Model 4: Add quadratic forms

$$y_{log(PledgedAmount)} = 48.515 + 1.718X_{log(backers)} + 13.64X_{duration.c} - 8.811X^2_{duration.c} + \beta_{MainCategory}X_{MainCategory} + \beta_{country}X_{country} - 0.023X_{LaunchedYear} - 0.062X_{log(GoalAmount))} + \beta_{MainCategory:log(backers)}X_{MainCategory:log(backers)}$$

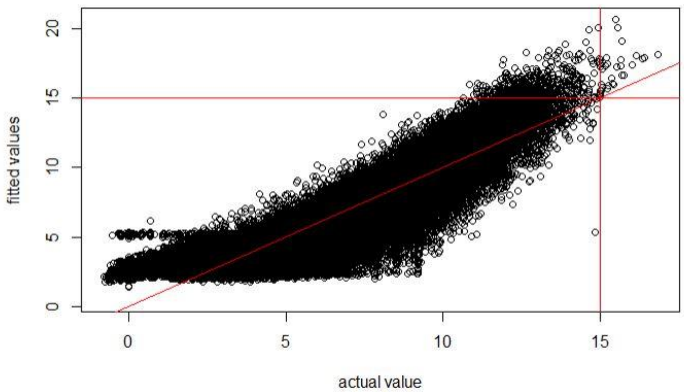
```
Residual standard error: 1.201 on 264914 degrees of freedom
Multiple R-squared:  0.8691,    Adjusted R-squared:  0.8691
F-statistic: 3.198e+04 on 55 and 264914 DF,  p-value: < 2.2e-16
```

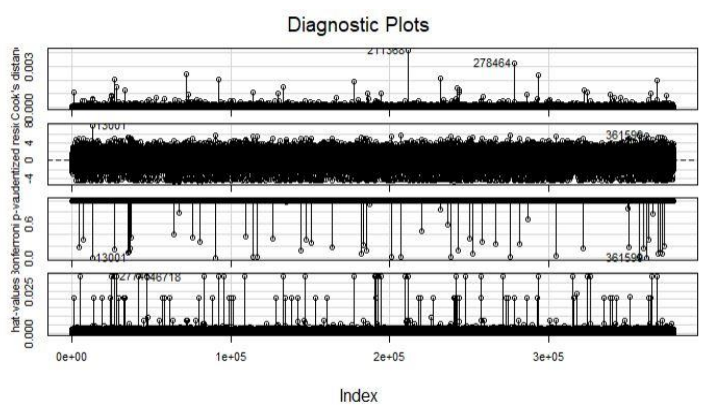
Comparing to the model 1 which has 0.10 adjusted R-squared, the model 4 has huge improvement.





However, the residual plot of this model is very bad, there is clear trend and residuals are not evenly split above and below the horizontal axis. In terms of the confidence interval of coefficients, we can see most of the coefficients are very close to zero if not across zero.





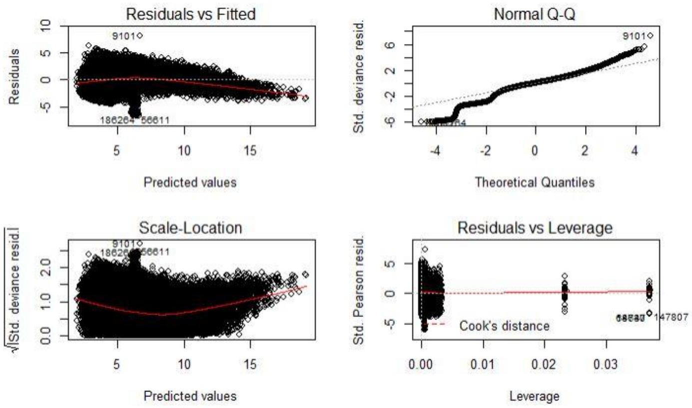
The actual vs fitted plot also shows the poor accuracy of the model. The influence index plot shows some outliers that have relative large influence on the regression coefficients.

Based on these findings, one possible solution could be to remove the zero values from the pledged amount and refit the model. Hurdle model is known for dealing with zero inflated data, however hurdle model usually is used for dealing with count data, while in my research the outcome variable is continuous. Therefore, I will borrow the concept from hurdle model to fit a two step model. First, fit a logistic regression on whether the outcome variable is zero or not, then fit a linear model conditionally on previous result.

Hurdle Model / Two Step Model

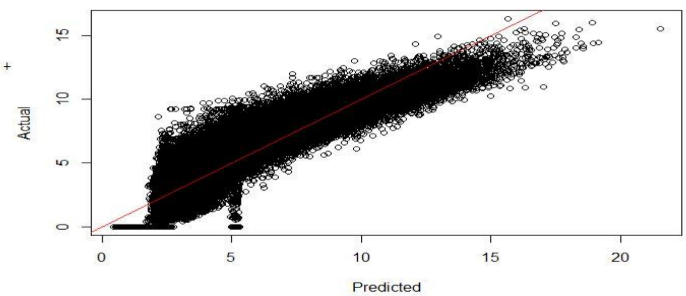
Step 1. $$y_{nonzero} = 421.4 + 69.45X_{log(backers)} - 0.06X_{duration.c} + \beta_{category}X_{category} + \beta_{country}X_{country} - 0.223X_{LaunchedYear} - 0.066X_{log(GoalAmount)}$$

Step 2. $$y_{PledgedAmount} = 78.99 + 1.4X_{log(backers)} - 0.014X_{duration.c} + \beta_{category}X_{category} + \beta_{country}X_{country} - 0.038X_{LaunchedYear} - 0.109X_{log(GoalAmount)}$$



The same issue is still in the residual pot. I will look for other possible solutions in the future.

```
Model 1: log(usd_pledged_real + 1) ~ backers + duration + main_category +
    country + as.numeric(launched_year) + log(usd_goal_real +
    1)
Model 2: log(usd_pledged_real + 1) ~ log(backers + 1) + duration.c + main_category +
    country + as.numeric(launched_year) + log(usd_goal_real +
    1)
Model 3: log(usd_pledged_real + 1) ~ log(backers + 1) + duration.c + country +
    main_category + as.numeric(launched_year) + log(usd_goal_real.c +
    1) + log(backers + 1):main_category
Model 4: log(usd_pledged_real + 1) ~ log(backers + 1) + poly(duration.c,
    2) + country + main_category + as.numeric(launched_year) +
    log(usd_goal_real.c + 1) + log(backers + 1):main_category
  Res.Df     RSS Df Sum of Sq       F  Pr(>F)
1 264929 2608395
2 264929  397825  0   2210570
3 264915  381912 14     15913 788.593 < 2e-16 ***
4 264914  381836  1        76  52.417 4.5e-13 ***
```



From the ANOVA test we can tell the model 4 is way better than the first model and slightly better than the second and third model. By looking at the predicted vs actual plot, the model 4 does not do well on cross-validation.

Based on these findings, I think a multilevel regression could handle the correlation between each observation more confidently, therefore, I will fit a couple multilevel linear models. I will skip the first couple models and get to the final model, since the procedure is similar to fit multilevel logistic regression models.

(iv) Multilevel Linear Model

- Model 5: Vary Intercept

$$y_{log(PledgedAmount)} = 38.1 + 1.64 + 1.61X_{log(backers)} + 0.019X_{duration.c} - 0.018X_{LaunchedYear} - 0.152X_{log(GoalAmount)} + \alpha_{j[i]}$$

$$\alpha_{j[i]} \sim N(0, 0.23^2)$$

- Model 6: Add vary intercept variable

$$y_{log(PledgedAmount)} = 66.227 + 1.5 + 1.63X_{log(backers)} + 0.007X_{duration.c} - 0.032X_{LaunchedYear} + 0.053X_{log(GoalAmount)} + \alpha_{j[i]} + u_{k[i]}$$

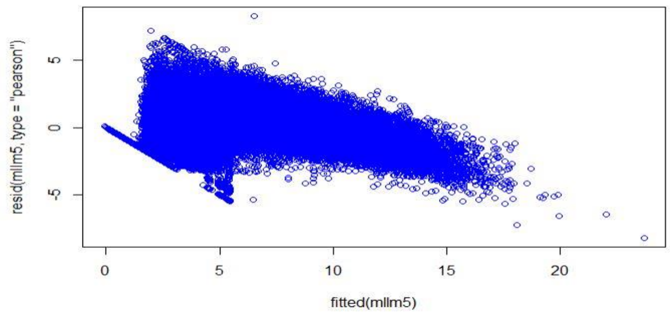
$$\alpha_{j[i]} \sim N(0, 0.21^2), u_{k[i]} \sim N(0, 0.78^2)$$

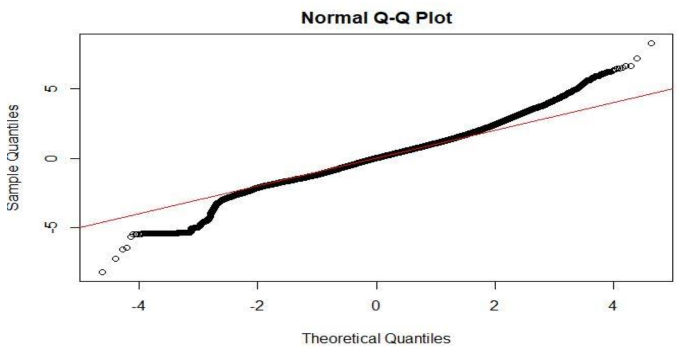
- Model 7: Vary slope and add nested random effects

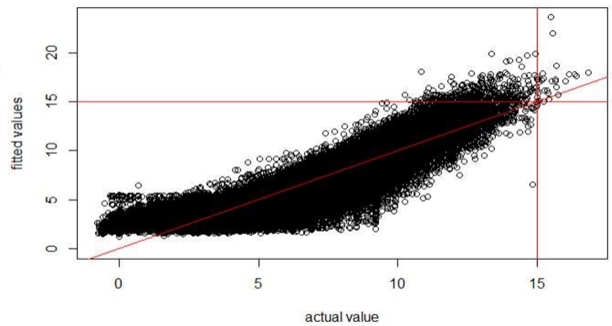
$$y \sim N(1.028 + \beta_{j[i]}X_{log(Backers)} + 0.01X_{duration} + 0.06X_{log(GoalAmount)} - 0.038X_{scaledLaunchedYear} + u_{j[i]} + w_{j[k]} + o_{n[i]}, \sigma_y^2)$$

$$u_{j[i]} \sim N(0, 0.11^2), w_{j[k]} \sim N(0, 0.41^2), o_{n[i]} \sim N(0, 0.8^2)$$

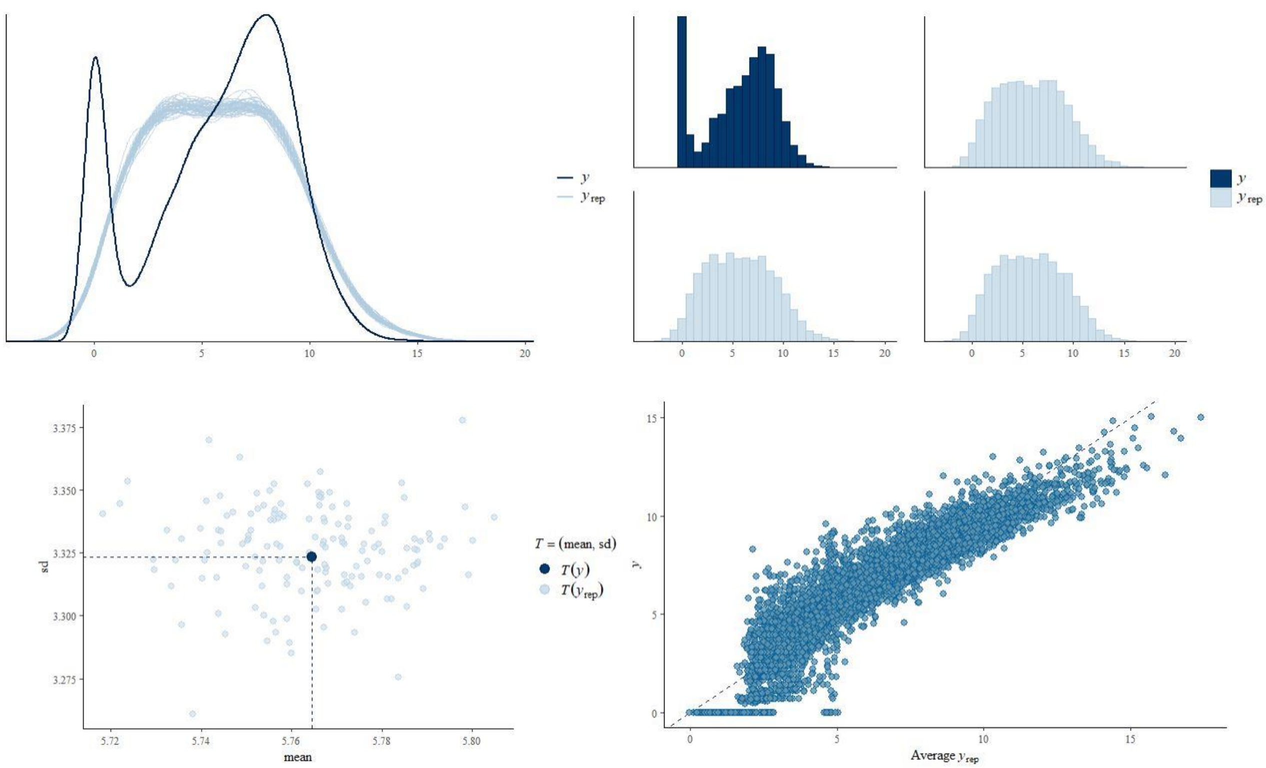
```
Random effects:
 Groups               Name             Variance Std.Dev. Corr
 category:main_category (Intercept)    0.17005  0.4124
                      log(backers + 1) 0.02461  0.1569   -0.91
 country              (Intercept)      0.64810  0.8050
 main_category        (Intercept)      0.06578  0.2565
                      log(backers + 1) 0.01247  0.1117   -0.89
 Residual                              1.36524  1.1684
Number of obs: 264970, groups:  category:main_category, 170; country, 23;

Fixed effects:
                                   Estimate Std. Error t value
(Intercept)                        1.028888   0.184868   5.566
log(backers + 1)                   1.704604   0.031851  53.517
duration.c                         0.010714   0.002373   4.516
log(usd_goal_real + 1)             0.060671   0.001487  40.792
scale(as.numeric(launched_year)) -0.038042   0.002746 -13.853
```



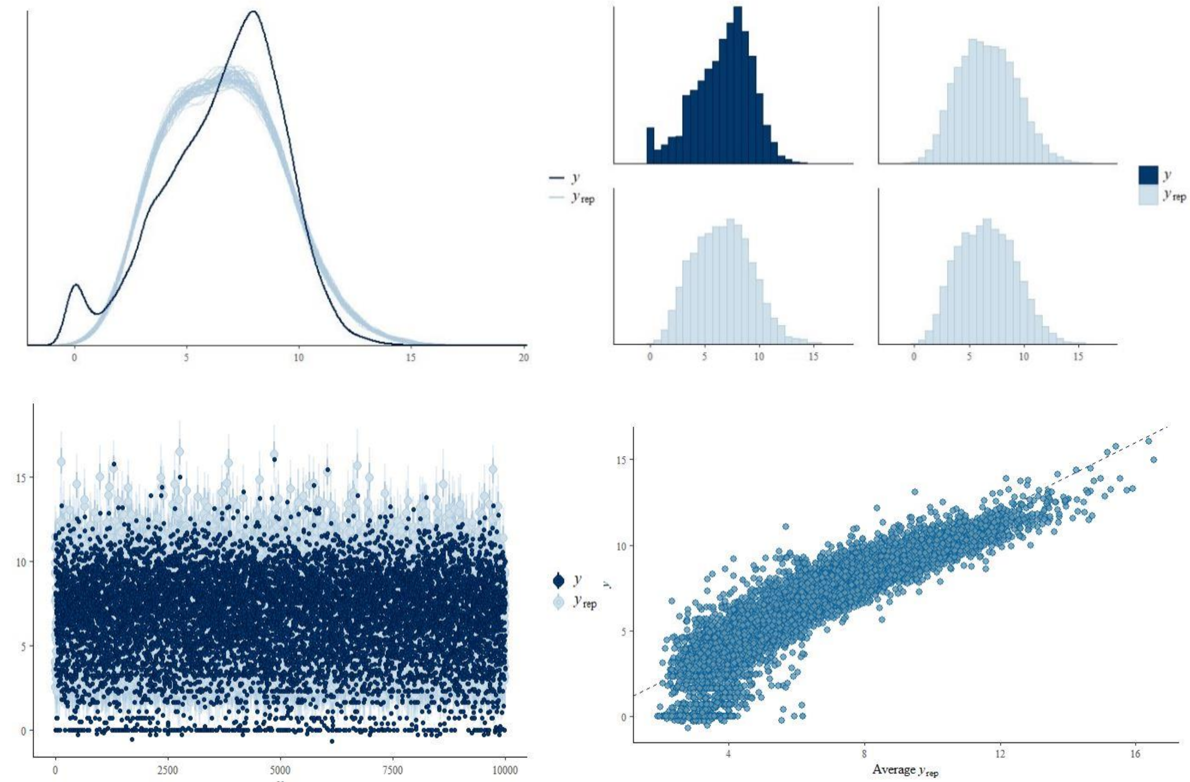




From the residual plot and the actual vs fitted plot we can see that the multilevel linear model does not correct the issue with residuals. I wonder if the problem is in the experiment design or the quality of dataset. For now, I do not have extra resource to complement this dataset, because of this, this could be the best model I could fit based on what I have right now.



From the posterior check we can see that, one major problem with the dataset is that there are too many zero values in the outcome variable. Removing all zero values in the outcome variables to see the results again.



As I removed all zero values, the plot looks much better. However, the bottom left plot still shows some concerns with either the dataset or the model.

Results and Findings

When analyzing the key factors that affect the success or failure of a project, I will choose the following model:

$$y_{state} \sim N(5.23 + \beta_{j[i]} X_{log(Backers)} - 0.05 X_{duration.c} - 2.044 X_{GoalAmount} + 0.037 X_{LaunchedYear} + u_{j[i]} + w_{j[k]}, \sigma_y^2)$$

The reason I choose this multilevel logistic model over classic logistic model is because the multilevel model has a lower deviance and slightly higher prediction accuracy. Moreover, I think it makes sense to put category and main category as random effects, since projects in same category may have correlated outcomes.

In the expression, the “beta” represents the varying slope, “u” represents the random intercept of main category, “w” represents the random intercept of “main category/category” which is a nested random effects structure. In this model, all of the fix effects are statistically significant, and there is strong correlation between number of backers and main category and sub category. This result confirmed my assumption and by adding number of backers as a vary slope variable, the AIC is decreased by 700. The interpretation of a multilevel model could be tricky, but I will do my best to interpret it.

On average, projects that has zero backer (log(backer+1)), 34 days launched period, zero goal amount(log(goal amount+1)), launched in year 2014(year is numeric variable and is scaled), will have 5.23+0.536(category:maincategory) +0.57(category)=6.336 log odds successfully raise funds on Kickstarter.

	(Intercept) <dbl>	log(backers + 1) <dbl>
Art	-0.45709260	0.25550710
Comics	-0.25127253	-0.28498311
Crafts	-0.81997241	0.10889369
Dance	0.13653989	0.34623979
Design	0.21718926	-0.24461935
Fashion	-0.39227125	0.16745459
Film & Video	• 1.31102560	• -0.18988104

	(Intercept) <dbl>	log(backers + 1) <dbl>
Musical:Theater	6.646630e-01	-1.077485e-01
Narrative Film:Film & Video	• 7.708677e-01	• -1.732083e-01

For a particular project, ID1000003930, it's under film & video category and narrative film sub-category. The log odds of successfully raise funds on Kickstarter of this project is 5.23 + (-0.189-1.732+3.428)*log(backers+1) – 0.05*(60-average duration)/sd – 2.044*log(30000+1) + 0.037*Year(2017-average year)/sd + 1.311 + 0.77 = -8.22.

Surprisingly, I didn't expect to see that duration and the state of a project has negative relationship, because in common sense, if the goal amount remain the same, the longer a project runs the more fund it would raise, and therefore, the probability of success should be increase.

When analyzing the key factors that affects the pledged amount, I will choose the following model:

$$y \sim N(1.028 + \beta_{j[i]} X_{log(Backers)} + 0.01 X_{duration} + 0.06 X_{log(GoalAmount)} - 0.038 X_{scaledLaunchedYear} + u_{j[i]} + w_{j[k]} + o_{n[i]}, \sigma_y^2)$$

$$u_{j[i]} \sim N(0, 0.11^2), w_{j[k]} \sim N(0, 0.41^2), o_{n[i]} \sim N(0, 0.8^2)$$

Although the multilevel model does not solve the issue with residuals, comparing to a classic linear model it could make more sense by adding country and category as random effects. And generally speaking, partial pooling model is at least no worse than complete pooling or no pooling models.

Interpretation will be similar to the previous logistic model, on average, a project with zero backer, last 34 days, zero goal amount, launched in 2014 and in the Art category from country Austria will raise exp(1.028+1.365)=10.94 dollars. All of the fix effects are statistically significant. Number of backers, duration and goal amount have positive relationship with pledged amount, while as year goes by, projects tend to raise less funds year by year.

For a particular project, ID 1000002330, it's from UK and under Publishing category and Poetry sub-category. The expected pledged amount of this project is exp(1.028+(3.428+0.02+0.01)*log(0+1)+0.01*1.93+0.06*log(1533.95+1)+ 0.038*(2015-2014)/1.92-0.014-0.415-0.179)=2.45 dollars.

Discussion

I. Implication

Based on the limited resource I have and all the regression analysis I have done, I would like to come to a conclusion that the success and failure of a project is largely affected by the number of backers and the goal amount. Meanwhile, the pledged amount of a project is affected largely by the number of backers.

II. Limitation

As it has been stated above, both state and pledged amount are affected largely by number of backers, however, it is impossible for fundraisers to know how many backers they will have in the future. Therefore, when making predictions, I would highly suggest using the average number of backers in the particular sub-category, or look for other resources to make assumptions on the number of backers.

Another important predictor in the model is the goal amount. However, I don't think simply increasing goal amount would help fund raisers to get more funds. I think, to some degree, goal amount could be a potential indicator of the quality of a project. For high quality and more complicated projects, fund raisers may set higher goal amount. While for simpler projects, fund raisers would set the goal amount relatively lower.

The reason for the poor model fit of most linear models could be due to limited information. It would be really difficult if not impossible to make predictions simply by using the duration, goal amount and category. Since one of the most important thing, the quality of the project, is missing. And currently, there is no way for us to find information like that. Therefore, I would say the model is limited for making predictions.

III. Future Direction

First of all, if we want to make a more precise model, the most important thing is to acquire more information about the projects. Such as people's review or rating of the project, and the market trend, the quality of the perks.

Second, since Kickstarter has the "all-or-nothing" rule, I would say it would be necessary to combine both state and pledged amount together, to do a joint data analysis.

Third, another interesting topic would be to analyze if there is fraud on Kickstarter. Due to the "all-or-nothing" rule, is it possible that fund raisers fund themselves in order to achieve the goal amount?

Last but not least, since number of backers is the most important thing and we know nothing about it beforehand, I am very interested in looking for methods to make predictions on the number of backers.

Reference

[1] https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html

[2] http://mc-stan.org/rstanarm/reference/pp_check.stanreg.html

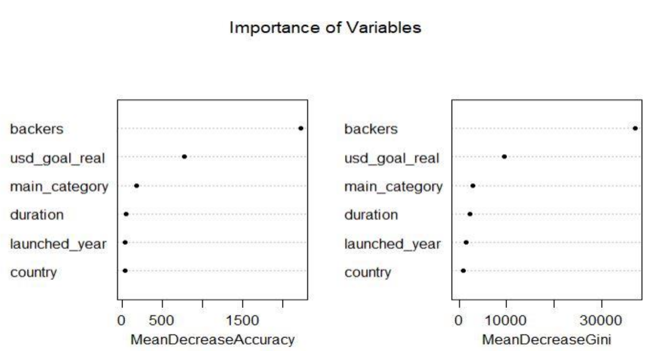
[3] https://datascienceplus.com/random-forests-in-r/

[4] https://data.library.virginia.edu/getting-started-with-hurdle-models/

Appendix

I also tried to fit a random forest model, the result is shown below:

```
rf4 <- randomForest(state~duration+usd_goal_real+main_category+country+launched_year+backers,
                    data = df_train_sf, ntree=500, mtry=2, importance=TRUE)
```




```
Confusion Matrix and Statistics

            Reference
Prediction   failed successful
  failed      27771       1520
  successful   2005      18675

               Accuracy : 0.9295
                 95% CI : (0.9272, 0.9317)
    No Information Rate : 0.5959
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8541
 Mcnemar's Test P-Value : 3.579e-16
```

The accuracy is lower than the multilevel logistic model by 1%.