KickStater Data

Intro

In Late April of 2009 KickStarter was founded, it is a website where people can come together to fund projects, monitaraly supporting a project makes you a "backer" and can be given a certain level of rewards from the creators on the site. KickStarter funds everything from video games, to books, to art, and music. If its a project that can be created by a small team it can find a home on KickStarter.

Issues

It should be stressed that creators on KickStarter are not necessarily professionals (although they can be) and thus projects can be over-ambishish or fall through. After a project is fully funded there are no promises that the creators will go through with their plan

While I wish that all people were good and that any failed projects are not in that state for valid reasons I am smarter than that I understand that some people do use the site to scam people from their money.

I hope to find ...

one or more identifiers of if a project will fail, for instance does it's category or sub category affect the projects chances to succeeded, does the location of the project have an impact and can it be negatively affected by amount of funds requested?

KickStater Data

Data

There is a data set provided on Kaggle I used the second half of it for this analysis and saved it as ks.csv. The data is separated into the following columns.

1. ID

a unique ID used to store within the database system of KickStarter

2. name

The name of the project, this has some issues that will be discussed in Data Cleaning

3. category

Creators are allowed to separate their projects into categories to help backers to find them they are listed in this column

4. main_category

Generated by the system this is overarching through the entre site. While a category can be cookbook, the main_category would be set to cooking.

5. currency

How payment is received such as USD, GBP, or EUR this is also a way to check the region to which the project belongs.

6. deadline

A date in which the fundraising phase of the project ends

7. goal

How much money they wish to earn (note this casted into the currency)

8. launched

A datetime of when the projected launched

9. pledged

amount pledged by "crowd"

10. state

Current condition the project is in

1. success: the project has launched
2. failed: failed to reach goal
3. cancelled: reached goal but never released
4. live: in the process of being complete (post funding)
5. undefined: data is missing

11. backers

The number of people supporting this project

12. country

Country of Origin

13. usd pledged

amount pledged (forced USD for comparison)

14. usd_pledged_real

How much the creator reported seeing

15. usd_goal_real

The goal in USD

KickStater Data

Data Cleaning

The issue here is `,`, everyday people use ',' to separate information. In the english language it is used to pause in a sentence. People use it as a way to put emphasis in titles which people on KickStater people do a lot.

Because the data is stored as a Comma Separated Values (.csv) file, having commas in the data means the data gets confused about what goes in what column.

I used `python` to repair this issue

Explaining the script

first we open up the original file and read the data in

count is included so I can count how many changes were necessary

```
with open ('ks.csv', 'r') as in_file:
    records = in_file.readlines() # loads all records into a list split by
new lines (currently strings)
    count = 0
```

We then opening up the file to write too and begin to edit the data

we then split on the ','

and prepare to remove anything that shouldn't be there like empty spaces or newlines

for the first 15 spaces we look to replace it with `N/A`

after that that is excess data and is removed from the data set entirely

```
with open('clean_ks.csv', 'w+') as out_file: # opening an out file
    for i in range(len(records)): # loads through each record
        records[i] = records[i].split(',') # seperates the list by ,
        for illegal_char in ['', '\n']: # checks if there are any
characters their shouldn't be (either a new line or an empty)
            while illegal_char in records[i][:15]: # while its in the valid
range
                records[i][records[i].index(illegal_char)] = "N/A" #
replace with a /n
            while illegal_char in records[i][15:]: #while its past the
maximum
                records[i].remove(illegal_char) # removes the characters
past the limit
```

now we get rid of those pesky extra ,

we know that the title is in the second col (list location 1), hence we can just check to see if the list is slightly too long and if it is we just take those 2 concatenate them and then delete whatever is n 2 and let it sync back.

```
        while len(records[i]) != len(records[0]): # checks if it is the
proper size
            records[i][1] += " " + records[i][2]  # if its not appends the
split tittle
            del records[i][2] # deletes the other (shifts everything back)
            count += 1
```

then in the final step we just write out what we have to the file

```
        for h in range(len(records[i])): # goes through the entire record
            if h != 0: # skips the first so it doesn't add a first ','
                out_file.write(',')
            out_file.write(records[i][h]) # writes out the actual record
        out_file.write("\n") # adds a \n to denote a new record
```