딥러닝을 이용한 크라우드펀딩 성공 예측 ¶

문제

- NGO, NPO, CBO, CSO, 사회적 기업, 소셜 벤처 등 운영 또는 프로그램/사업 수행 자금 및 재정 문제
- 특히 비영리적 성격이 강하거나 완전한 비영리인 경우 이러한 문제는 고질적
- 하지만 정부 지원, 재단 후원, 기업의 투자, 협력 사업 등을 통한 자금 지원을 얻는 것엔 한계나 어려움이 항상 있음

해결

- **크라우드펀딩**을 통한 모금
- 크라우드펀딩 성공을 미리 예측하는 것이 펀딩 성공에 큰 도움이 될 것으로 판단

설명

- 세계 최대 크라우드펀딩 플랫폼 **Kistarter** 데이터 수집
  - 17만개 이상의 성공적인 펀드레이징, 1,700만명 이상의 후원자, 4조 8천억원의 총 모금액
  - 방대한 양의 데이터를 통해 다양한 인사이트를 얻을 수 있을 것으로 판단
- **펀딩 런칭 전(pre-launch)** 관련 요인/특성 데이터만 사용
  - 지금까지의 크라우드펀딩 예측 관련 연구나 논문들은 펀딩 런칭 후(post-launch)에 발생하는 요인, 결과 또는 역학(dynamics)을 분석한 펀딩 결과 예측이 대부분 (ex: 댓글, 답글, 소셜미디어/온라인 상의 전파, 파급력, 공유 횟수, 후원자 수 등)
  - 하지만 이러한 펀딩 런칭 후 발생하는 요인과 결과는 모금을 하는 대상이 그 결과를 만들어내거나 제어 할 수 없는 또는 한계가 큰 요인인 것이 대부분이며, 이러한 런칭 후 발생하는 요인을 이용한 결과 예측은 런칭 후 일정 시간이 지난 뒤에야 펀딩 결과를 예측을 할 수 있거나 시기적절하게 결과를 예측 할 수 없고 펀딩을 수정하거나 개선하는 것에도 어려움과 한계가 있음
  - 하지만 펀딩 런칭 전(pre-launch)과 관련된 요인들은 직접 설정과 제어를 할 수 있으며, 그리고 펀딩 런칭 전에 펀딩 성공/성공율을 미리 예측 할 수 있다면 무엇이 문제일지 고민하고 파악할 수 있고 성공율을 더 높이는 방향으로 펀딩 프로젝트를 수정/개선해 갈 수 있고, 이렇게 사전에 미리 잘 준비하고 퀄리티 있는 프로젝트를 만들 수 있다면 펀딩 성공율은 크게 증가할 것
  - 한 마디로, 펀딩을 런칭하기 전에 미리 펑딩에 성공 할 수 있는 펀딩 프로젝트를 만드는 것
- 탐색적 분석 및 **딥러닝**을 활용한 펀딩 성공 예측 모델링

기대 효과

- 성공 할 수 있는 또는 가능성이 큰 크라우드펀딩 프로젝트를 만들 수 있다면:
  1. 대중에게 기업/기관 또는 개인의 일과 분야에 대해 알릴 수 있고 대중의 관심을 이끌어내거나 대중의 인식을 높일 수 있으며
  2. 모금을 통해 필요한 자원/자금 또한 얻을 수 있음

데이터셋

- 기본 프로필 데이터/메타 데이터
  - 목표 모금액, 모금기간 등 펀딩에 대한 다양한 설정
- 이미지 데이터
  - 펀딩 프로젝트에 사용된 메인 사진
  - 웹사이트 메뉴나 펀딩 페이지 상에서 가장 먼저 눈에 보임
- 텍스트(text) 데이터
  - 펀딩 제목, 부제목(펀딩에 대한 간략한 설명글), 본문

분석 방향

- 탐색적 분석 및 복합적/다각적 딥러닝 모델 구현
- 메타 데이터, 이미지 데이터, 텍스트 데이터를 전처리 및 가공한 결과를 결합하여 딥러닝 모델에 적용

데이터 읽기

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime, timedelta
import os
import json
import time
import random
np.random.seed(1)
```

In [5]:
```
# 40여개 csv파일로 나뉘어져 있는 데이터셋
# 모든 데이터셋 파일 하나로 합쳐서 가져오기
import os
import glob

path = r'/Users/DaeyoungKim/Python/Final_Project/Kickstarter_2019-12-12'
all_files = glob.glob(os.path.join(path, "*.csv")) #glob.iglob --> returns iterator, instead of a list

df_from_each_file = (pd.read_csv(f) for f in all_files)
concatenated_df = pd.concat(df_from_each_file, ignore_index=True)

#df = pd.concat(map(pd.read_csv, glob.glob(os.path.join('', "my_files*.csv"))))
```

In [73]:
```
concatenated_df.head(1)
```

Out[73]:

	backers_count	blurb	category	converted_pledged_amount	country	country_displayable_name	created_at	creator	currency	currency
0	31	Un livre enfant pour l'apprentissage des émoti...	{"id":46,"name":"Children's Books","slug":"pub...	709	FR	France	1561554849	{"id":469036700,"name":"Camille de Germond","i...	EUR	

1 rows × 38 columns

In [74]:
```
concatenated_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215800 entries, 0 to 215799
Data columns (total 38 columns):
backers_count               215800 non-null int64
blurb                       215792 non-null object
category                    215800 non-null object
converted_pledged_amount    215800 non-null int64
country                     215800 non-null object
country_displayable_name    215800 non-null object
created_at                  215800 non-null int64
creator                     215800 non-null object
currency                    215800 non-null object
currency_symbol             215800 non-null object
currency_trailing_code      215800 non-null bool
current_currency            215800 non-null object
deadline                    215800 non-null int64
disable_communication       215800 non-null bool
friends                     252 non-null object
fx_rate                     215800 non-null float64
goal                        215800 non-null float64
id                          215800 non-null int64
is_backing                  252 non-null object
is_starrable                215800 non-null bool
is_starred                  252 non-null object
launched_at                 215800 non-null int64
location                    215583 non-null object
name                        215800 non-null object
permissions                 252 non-null object
photo                       215800 non-null object
pledged                     215800 non-null float64
profile                     215800 non-null object
slug                        215800 non-null object
source_url                  215800 non-null object
spotlight                   215800 non-null bool
staff_pick                  215800 non-null bool
state                       215800 non-null object
state_changed_at            215800 non-null int64
static_usd_rate             215800 non-null float64
urls                        215800 non-null object
usd_pledged                 215800 non-null float64
usd_type                    215656 non-null object
dtypes: bool(5), float64(5), int64(7), object(21)
memory usage: 55.4+ MB
```

데이터 크기:

- 현재 데이터 행 수: **215,799**
- 현재 데이터 열(컬럼) 수: **38**

각 컬럼 및 데이터 형식 탐색

backers_count

- 해당 펀딩 프로젝트의 후원자 수

In [54]:
```
column_to_look = 'backers_count'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  3284
data size:  215800
dtype:  <class 'numpy.int64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

31
```

blurb

- 펀딩의 부제목; 프로젝트 관련 짤막한 promotiional 설명글

In [53]:
```
column_to_look = 'blurb'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  186790
data size:  215800
dtype:  <class 'str'>
null/nan 수:  8
------------------------------
데이터 형태 예시

Un livre enfant pour l'apprentissage des émotions.
```

category

- 펀딩(제품/서비스) 유형

In [52]:
```
column_to_look = 'category'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  170
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

{'id': 46, 'name': "Children's Books", 'slug': "publishing/children's books", 'position': 5, 'parent_id': 18, 'color': 14867664, 'urls':
{'web': {'discover': "http://www.kickstarter.com/discover/categories/publishing/children's%20books"}}}
```

In [ ]:
```
# 'category'컬럼이 생긴 형태는 dictionary이지만, dtype은 string
# --> dictionary 기능을 하도록 실제 dictionary로 변경
```

In [75]:
```
category_dic = kickstarter_data #작업할 변수 생성
type(category_dic['category'][0]) #기존 category컬럼 = string
```

Out[75]:
```
str
```

In [67]:
```
import ast
category_dic1 = category_dic['category'].apply(lambda x: ast.literal_eval(x))
category_dic1.head() #category 컬럼의 모든 값들을 dictionary로 변경한 새로운 시리즈 생성 (category 컬럼을 이걸로 대체하자)
```

Out[67]:
```
0    {'id': 46, 'name': 'Children's Books', 'slug':...
1    {'id': 252, 'name': 'Graphic Novels', 'slug': ...
2    {'id': 332, 'name': 'Apps', 'slug': 'technolog...
3    {'id': 252, 'name': 'Graphic Novels', 'slug': ...
4    {'id': 49, 'name': 'Periodicals', 'slug': 'pub...
Name: category, dtype: object
```

In [ ]:
```
#기존 category컬럼 --> dictionary로 바꾼 category컬럼으로 대체
category_dic['category'] = category_dic1
```

In [77]:
```
type(category_dic['category'][0]) #dictionary로 바뀐 것 확인
```

Out[77]:
```
dict
```

In [72]:
```
# 이제 'category'컬럼의 모든 값은 dictionary
# category dictionary에서 'name' key만 추출
category_name = category_dic1.apply(lambda x: x['name'])
category_name.head()
```

Out[72]:
```
0    Children's Books
1      Graphic Novels
2                Apps
3      Graphic Novels
4         Periodicals
Name: category, dtype: object
```

In [73]:
```
# category dictionary에서 'slug' key만 추출
category_specific = category_dic1.apply(lambda x: x['slug'])
category_specific.head()
```

Out[73]:
```
0    publishing/children's books
1          comics/graphic novels
2                technology/apps
3          comics/graphic novels
4          publishing/periodicals
Name: category, dtype: object
```

In [ ]:
```
#기존 dataframe에 category_name, category_specific 새 컬럼으로 삽입
category_dic.insert(3, 'category_name', category_name)
category_dic.insert(4, 'category_specific', category_specific)
```

category_name

- 펀딩 유형 이름

In [51]:
```
column_to_look = 'category_name'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  160
unique value별 개수:
 Web                 4644
Product Design      4403
Tabletop Games      4047
Accessories         3651
Comic Books         3543
                    ...
Games                 99
Letterpress           75
Chiptune              51
Social Practice       49
Taxidermy             18
Name: category_name, Length: 160, dtype: int64
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

Children's Books
```

category_specific

- 해당 펀딩의 분류/유형에 대한 slug (접근하기 위한 root -- url 가장 뒤에 붙는다)
- 펀딩의 상위 유형(분류) 포함 (value 형태: 큰 유형/세부 유형)

In [55]:
```
column_to_look = 'category_specific'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  170
unique value별 개수:
 design/product design     4403
games/tabletop games      4047
fashion/accessories       3651
comics/comic books        3543
art/illustration          3117
                          ...
music/comedy                84
publishing/letterpress      75
music/chiptune              51
art/social practice         49
crafts/taxidermy            18
Name: category_specific, Length: 170, dtype: int64
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

publishing/children's books
```

In [6]:
```
#2019년 데이터만 뽑아서 category 관련 컬럼들 탐색 시도
df_2019['category_name'].nunique()
```

Out[6]: 159

In [7]:
```
df_2019['category_specific'].nunique()
```

Out[7]: 169

In [ ]:
```
#처음엔 이 두 컬럼의 유일값(unique)이 같을 것으로 기대했지만
#조회 결과 두 컬럼의 값이 다르다
```

In [115]:
```
category_df = df_2019['category_specific'].groupby(df_2019['category_name'])
c_df1 = category_df.describe()
c_df1
```

Out[115]:

category_name	count	unique	top	freq
3D Printing	108	1	technology/3d printing	108
Academic	152	1	publishing/academic	152
Accessories	1584	1	fashion/accessories	1584
Action	101	1	film & video/action	101
Animals	25	1	photography/animals	25
...	...	...	...	...
Woodworking	157	1	crafts/woodworking	157
Workshops	14	1	dance/workshops	14
World Music	232	1	music/world music	232
Young Adult	148	1	publishing/young adult	148
Zines	152	1	publishing/zines	152

159 rows × 4 columns

In [128]:
```
c_df1[c_df1['unique'] > 1]
```

Out[128]:

category_name	count	unique	top	freq
Anthologies	416	2	comics/anthologies	243
Comedy	581	4	film & video/comedy	362
Events	93	2	food/events	52
Experimental	148	2	film & video/experimental	85
Festivals	160	2	theater/festivals	115
Spaces	157	3	food/spaces	91
Web	480	2	technology/web	279

In [ ]:
```
# 조회 결과 아래의 Anthologies, Comdedy, Events 등의 category는 더 큰 분류에서 보면 1개 이상이다
# Comedy를 예로 들면, music/comedy, publishing/comedy, theater/comedy, film & video/comedy 이렇게 4개의 큰 카테고리들이 다 comedy라는 subcategory가
  있음
# Spaces도 예로 들면, dance/spaces, food/spaces, theater/spaces에 각 spaces라는 subcategory가 있다
```

In [124]:
```
test2 = df_2019['category_name'].groupby(df_2019['category_specific'])
c_df2 = test2.describe()
c_df2
```

Out[124]:

category_specific	count	unique	top	freq
art	1001	1	Art	1001
art/ceramics	83	1	Ceramics	83
art/conceptual art	131	1	Conceptual Art	131
art/digital art	480	1	Digital Art	480
art/illustration	1503	1	Illustration	1503
...	...	...	...	...
theater/festivals	115	1	Festivals	115
theater/immersive	64	1	Immersive	64
theater/musical	160	1	Musical	160
theater/plays	252	1	Plays	252
theater/spaces	36	1	Spaces	36

169 rows × 4 columns

In [129]:
```
#c_df2[c_df2['count'] != c_df2['freq']]
c_df2[c_df2['unique'] > 1]

#이건 category_specific이랑 category_name 개수가 맞다
```

Out[129]:

category_specific	count	unique	top	freq

In [72]:
```
np.unique(df_2019['category_specific'].values)
```

Out[72]:
```
array(['art', 'art/ceramics', 'art/conceptual art', 'art/digital art',
       'art/illustration', 'art/installations', 'art/mixed media',
       'art/painting', 'art/performance art', 'art/public art',
       'art/sculpture', 'art/social practice', 'art/textiles',
       'art/video art', 'comics', 'comics/anthologies',
       'comics/comic books', 'comics/events', 'comics/graphic novels',
       'comics/webcomics', 'crafts', 'crafts/candles', 'crafts/crochet',
       'crafts/diy', 'crafts/embroidery', 'crafts/glass',
       'crafts/knitting', 'crafts/pottery', 'crafts/printing',
       'crafts/quilts', 'crafts/stationery', 'crafts/weaving',
       'crafts/woodworking', 'dance', 'dance/performances',
       'dance/residencies', 'dance/spaces', 'dance/workshops', 'design',
       'design/architecture', 'design/civic design',
       'design/graphic design', 'design/interactive design',
       'design/product design', 'design/typography', 'fashion',
       'fashion/accessories', 'fashion/apparel', 'fashion/childrenswear',
       'fashion/couture', 'fashion/footwear', 'fashion/jewelry',
       'fashion/pet fashion', 'fashion/ready-to-wear', 'film & video',
       'film & video/action', 'film & video/animation',
       'film & video/comedy', 'film & video/documentary',
       'film & video/drama', 'film & video/experimental',
       'film & video/family', 'film & video/fantasy',
       'film & video/festivals', 'film & video/horror',
       'film & video/movie theaters', 'film & video/music videos',
       'film & video/narrative film', 'film & video/romance',
       'film & video/science fiction', 'film & video/shorts',
       'film & video/television', 'film & video/thrillers',
       'film & video/webseries', 'food', 'food/bacon',
       'food/community gardens', 'food/cookbooks', 'food/drinks',
       'food/events', "food/farmer's markets", 'food/farms',
       'food/food trucks', 'food/restaurants', 'food/small batch',
       'food/spaces', 'food/vegan', 'games', 'games/gaming hardware',
       'games/live games', 'games/mobile games', 'games/playing cards',
       'games/puzzles', 'games/tabletop games', 'games/video games',
       'journalism', 'journalism/audio', 'journalism/photo',
       'journalism/print', 'journalism/video', 'journalism/web', 'music',
       'music/blues', 'music/chiptune', 'music/classical music',
       'music/comedy', 'music/country & folk', 'music/electronic music',
       'music/faith', 'music/hip-hop', 'music/indie rock', 'music/jazz',
       'music/kids', 'music/latin', 'music/metal', 'music/pop',
       'music/punk', 'music/r&b', 'music/rock', 'music/world music',
       'photography', 'photography/animals', 'photography/fine art',
       'photography/nature', 'photography/people',
       'photography/photobooks', 'photography/places', 'publishing',
       'publishing/academic', 'publishing/anthologies',
       'publishing/art books', 'publishing/calendars',
       "publishing/children's books", 'publishing/comedy',
       'publishing/fiction', 'publishing/letterpress',
       'publishing/literary journals', 'publishing/literary spaces',
       'publishing/nonfiction', 'publishing/periodicals',
       'publishing/poetry', 'publishing/radio & podcasts',
       'publishing/translations', 'publishing/young adult',
       'publishing/zines', 'technology', 'technology/3d printing',
       'technology/apps', 'technology/camera equipment',
       'technology/diy electronics', 'technology/fabrication tools',
       'technology/flight', 'technology/gadgets', 'technology/hardware',
       'technology/makerspaces', 'technology/robots',
       'technology/software', 'technology/sound',
       'technology/space exploration', 'technology/wearables',
       'technology/web', 'theater', 'theater/comedy',
       'theater/experimental', 'theater/festivals', 'theater/immersive',
       'theater/musical', 'theater/plays', 'theater/spaces'], dtype=object)
```

In [79]:
```
test = df_2019.groupby('category_name')
test.groups.keys()
```

Out[79]:
```
dict_keys(['3D Printing', 'Academic', 'Accessories', 'Action', 'Animals', 'Animation', 'Anthologies', 'Apparel', 'Apps', 'Architecture',
'Art', 'Art Books', 'Audio', 'Bacon', 'Blues', 'Calendars', 'Camera Equipment', 'Candles', 'Ceramics', "Children's Books", 'Childrenswea
r', 'Chiptune', 'Civic Design', 'Classical Music', 'Comedy', 'Comic Books', 'Comics', 'Community Gardens', 'Conceptual Art', 'Cookbook
s', 'Country & Folk', 'Couture', 'Crafts', 'Crochet', 'DIY', 'DIY Electronics', 'Dance', 'Design', 'Digital Art', 'Documentary', 'Dram
a', 'Drinks', 'Electronic Music', 'Embroidery', 'Events', 'Experimental', 'Fabrication Tools', 'Faith', 'Family', 'Fantasy', "Farmer's M
arkets", 'Farms', 'Fashion', 'Festivals', 'Fiction', 'Film & Video', 'Fine Art', 'Flight', 'Food', 'Food Trucks', 'Footwear', 'Gadgets',
'Games', 'Gaming Hardware', 'Glass', 'Graphic Design', 'Graphic Novels', 'Hardware', 'Hip-Hop', 'Horror', 'Illustration', 'Immersive',
'Indie Rock', 'Installations', 'Interactive Design', 'Jazz', 'Jewelry', 'Journalism', 'Kids', 'Knitting', 'Latin', 'Letterpress', 'Liter
ary Journals', 'Literary Spaces', 'Live Games', 'Makerspaces', 'Metal', 'Mixed Media', 'Mobile Games', 'Movie Theaters', 'Music', 'Music
Videos', 'Musical', 'Narrative Film', 'Nature', 'Nonfiction', 'Painting', 'People', 'Performance Art', 'Performances', 'Periodicals', 'P
et Fashion', 'Photo', 'Photobooks', 'Photography', 'Places', 'Playing Cards', 'Plays', 'Poetry', 'Pop', 'Pottery', 'Print', 'Printing',
'Product Design', 'Public Art', 'Publishing', 'Punk', 'Puzzles', 'Quilts', 'R&B', 'Radio & Podcasts', 'Ready-to-wear', 'Residencies', 'R
estaurants', 'Robots', 'Rock', 'Romance', 'Science Fiction', 'Sculpture', 'Shorts', 'Small Batch', 'Social Practice', 'Software', 'Soun
d', 'Space Exploration', 'Spaces', 'Stationery', 'Tabletop Games', 'Technology', 'Television', 'Textiles', 'Theater', 'Thrillers', 'Tran
slations', 'Typography', 'Vegan', 'Video', 'Video Art', 'Video Games', 'Wearables', 'Weaving', 'Web', 'Webcomics', 'Webseries', 'Woodwor
king', 'Workshops', 'World Music', 'Young Adult', 'Zines'])
```

In [ ]:
```
# 어떤 프로젝트들은 category_name, category_specific 둘 다 같은 것이 있다; 예)art, comics, crafts 등
# 이런 프로젝트들은 main category는 설정 했지만 subcategory는 설정 안 했기 때문일 것으로 판단
```

converted_pledged_amount

- 단위 변환된 모금액수

In [56]:
```
column_to_look = 'converted_pledged_amount'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  32379
data size:  215800
dtype:  <class 'numpy.int64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

709
```

country

- 국가 코드

In [57]:
```
column_to_look = 'country'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  22
unique value별 개수:
 US    149987
GB     24540
CA     10162
AU      5228
DE      3822
FR      3007
MX      2880
IT      2666
ES      2276
NL      1881
SE      1550
HK      1376
NZ       978
DK       958
SG       801
CH       735
IE       683
BE       632
AT       546
NO       521
JP       504
LU        67
Name: country, dtype: int64
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

FR
```

country_displayable_name

- 국가명

In [58]:
```
column_to_look = 'country_displayable_name'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  22
unique value별 개수:
 the United States     149987
the United Kingdom     24540
Canada                 10162
Australia               5228
Germany                 3822
France                  3007
Mexico                  2880
Italy                   2666
Spain                   2276
the Netherlands         1881
Sweden                  1550
Hong Kong               1376
New Zealand              978
Denmark                  958
Singapore                801
Switzerland              735
Ireland                  683
Belgium                  632
Austria                  546
Norway                   521
Japan                    504
Luxembourg                67
Name: country_displayable_name, dtype: int64
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

France
```

created_at

- 펀딩 프로젝트가 만들어진 시간 (milliseconds)

In [59]:
```
column_to_look = 'created_at'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  188280
data size:  215800
dtype:  <class 'numpy.int64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

1561554849
```

In [ ]:
```
#'created_at' 컬럼 datetime형식으로 변경
date_created_s = concatenated_df['created_at']
date_created = pd.to_datetime(date_created_s, unit='s')
print(date_created.head(3))
print(date_created.tail(3))
print("first project: ", date_created.min())
print("last project: ", date_created.max())
```

```
0   2019-06-26 13:14:09
1   2015-08-05 02:11:53
2   2018-06-07 19:53:22
Name: created_at, dtype: datetime64[ns]
215797   2019-05-16 19:20:15
215798   2014-07-08 22:35:09
215799   2013-06-08 14:12:57
Name: created_at, dtype: datetime64[ns]
first project:  2009-04-22 02:11:10
last project:  2019-12-12 02:42:36
```

In [ ]:
```
#생성한 datetime 컬럼 dataframe에 삽입
concatenated_df.insert(7, 'date_created', date_created)
```

date_created

- 펀딩 프로젝트가 만들어진 날짜 (datetime 형태)

In [60]:
```
column_to_look = 'date_created'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  188280
data size:  215800
dtype:  <class 'pandas._libs.tslibs.timestamps.Timestamp'>
null/nan 수:  0
------------------------------
데이터 형태 예시

2019-06-26 13:14:09
```

creator

- 펀딩 프로젝트 만든이 정보 (dictionary 형태)

In [61]:
```
column_to_look = 'creator'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  215165
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

{"id":469036700,"name":"Camille de Germond","is_registered":null,"chosen_currency":null,"is_superbacker":null,"avatar":{"thumb":"http
s://ksr-ugc.imgix.net/assets/025/618/811/ca566ff21d110f9cbe451cc60e5f6c81_original.jpeg?ixlib=rb-2.1.0&w=40&h=40&fit=crop&v=1561558757&a
uto=format&frame=1&q=92&s=81691e924e37a2ffca7b5e901ff944d6","small":"https://ksr-ugc.imgix.net/assets/025/618/811/ca566ff21d110f9cbe451c
c60e5f6c81_original.jpeg?ixlib=rb-2.1.0&w=160&h=160&fit=crop&v=1561558757&auto=format&frame=1&q=92&s=4ff5ad957532e8b7ffa9268d15f3ffb
1","medium":"https://ksr-ugc.imgix.net/assets/025/618/811/ca566ff21d110f9cbe451cc60e5f6c81_original.jpeg?ixlib=rb-2.1.0&w=160&h=160&fit=
crop&v=1561558757&auto=format&frame=1&q=92&s=4ff5ad957532e8b7ffa9268d15f3ffb1"},"urls":{"web":{"user":"https://www.kickstarter.com/profi
le/469036700"},"api":{"user":"https://api.kickstarter.com/v1/users/469036700?signature=1576212594.3913eae0433e83a23dfa9662805147a43a5ced
f1"}}}
```

currency

- 펀딩 프로젝트에 설정된 돈 단위

In [63]:
```
column_to_look = 'currency'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  14
unique value별 개수:
 USD    149987
GBP     24540
EUR     15580
CAD     10162
AUD      5228
MXN      2880
SEK      1550
HKD      1376
NZD       978
DKK       958
SGD       801
CHF       735
NOK       521
JPY       504
Name: currency, dtype: int64
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

EUR
```

currency_symbol

- 펀딩 프로젝트에 설정된 돈 단위 상징

In [65]:
```
column_to_look = 'currency_symbol'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  6
unique value별 개수:
 $      171412
£       24540
€       15580
kr       3029
Fr        735
¥         504
Name: currency_symbol, dtype: int64
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

€
```

currency_trailing_code

In [66]:
```
column_to_look = 'currency_trailing_code'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  2
unique value별 개수:
 True     174441
False     41359
Name: currency_trailing_code, dtype: int64
data size:  215800
dtype:  <class 'numpy.bool_'>
null/nan 수:  0
------------------------------
데이터 형태 예시

False
```

current_currency

- 단위 변환 후의 현재 돈 단위

In [68]:
```
column_to_look = 'current_currency'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  5
unique value별 개수:
 USD    215656
GBP        48
CAD        48
AUD        36
EUR        12
Name: current_currency, dtype: int64
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

USD
```

In [ ]:
```
#다른 label이 5개나 있다
#만약 필요하다면 currency는 다 통일
```

deadline

- 펀딩 마감 시간 (milliseconds)

In [69]:
```
column_to_look = 'deadline'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  176815
data size:  215800
dtype:  <class 'numpy.int64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

1566744397
```

In [ ]:
```
#'deadline' 컬럼 datetime형식으로 변경
deadline_s = concatenated_df['deadline']
date_deadline = pd.to_datetime(deadline_s, unit='s')
print(date_deadline.head(3))
print(date_deadline.tail(3))
print("first project: ", date_deadline.min())
print("last project: ", date_deadline.max())
```

```
0   2019-08-25 14:46:37
1   2015-09-14 04:19:27
2   2018-08-18 15:43:54
Name: deadline, dtype: datetime64[ns]
215797   2019-07-16 06:59:00
215798   2014-08-31 22:35:00
215799   2015-04-08 17:20:06
Name: deadline, dtype: datetime64[ns]
first project:  2009-05-16 09:59:00
last project:  2020-02-10 04:37:15
```

In [ ]:
```
#생성한 datetime 컬럼 dataframe에 삽입
concatenated_df.insert(14, 'date_deadline', date_deadline)
```

date_deadline

- 펀딩 마감 날짜 (datetime 형태)

In [70]:
```
column_to_look = 'date_deadline'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  176815
data size:  215800
dtype:  <class 'pandas._libs.tslibs.timestamps.Timestamp'>
null/nan 수:  0
------------------------------
데이터 형태 예시

2019-08-25 14:46:37
```

disable_communication

- 펀딩 크리에이터와의 커뮤니케이션 가능 유무

In [109]:
```
column_to_look = 'disable_communication'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look].value_counts())
```

```
unique values:  2
unique value별 개수:
 False    215152
True        648
Name: disable_communication, dtype: int64
data size:  215800
dtype:  <class 'numpy.bool_'>
null/nan 수:  0
------------------------------
데이터 형태 예시

False    215152
True        648
Name: disable_communication, dtype: int64
```

friends

In [85]:
```
column_to_look = 'friends'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look].value_counts()[1:2])
```

```
unique values:  27
data size:  215800
dtype:  <class 'float'>
null/nan 수:  215548
------------------------------
데이터 형태 예시

[{"id":1528750001,"name":"Blacklist Games","slug":"blacklistgames","is_registered":null,"chosen_currency":null,"is_superbacker":null,"av
atar":{"thumb":"https://ksr-ugc.imgix.net/assets/012/512/828/b761b350a991b24b42622274f46bb375_original.png?ixlib=rb-2.1.0&w=40&h=40&fit=
crop&v=1464152704&auto=format&frame=1&q=92&s=0761c36aa7a5a934c927d9ca41f332c1","small":"https://ksr-ugc.imgix.net/assets/012/512/828/b76
1b350a991b24b42622274f46bb375_original.png?ixlib=rb-2.1.0&w=160&h=160&fit=crop&v=1464152704&auto=format&frame=1&q=92&s=ca895cc6d08253e9d
9eb56e2b5ab9f3a","medium":"https://ksr-ugc.imgix.net/assets/012/512/828/b761b350a991b24b42622274f46bb375_original.png?ixlib=rb-2.1.0&w=1
60&h=160&fit=crop&v=1464152704&auto=format&frame=1&q=92&s=ca895cc6d08253e9d9eb56e2b5ab9f3a"},"urls":{"web":{"user":"https://www.kickstar
ter.com/profile/blacklistgames"},"api":{"user":"https://api.kickstarter.com/v1/users/1528750001?signature=1576211015.2dbb000da9ab5d21d71
7071215b1b9b44463f1db"}}}]    1
Name: friends, dtype: int64
```

fx_rate

- 환율 (Foreign Exchange Rate)

In [90]:
```
column_to_look = 'fx_rate'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look].value_counts().head())
```

```
unique values:  56
data size:  215800
dtype:  <class 'numpy.float64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

1.000000    149922
1.313810     14535
1.321770      9993
1.108973      9648
0.755601      6216
Name: fx_rate, dtype: int64
```

goal

- 목표 모금액 (in local currency)

In [91]:
```
column_to_look = 'goal'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  5370
data size:  215800
dtype:  <class 'numpy.float64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

600.0
```

id

- 펀딩 프로젝트 ID

In [92]:
```
column_to_look = 'id'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  188385
data size:  215800
dtype:  <class 'numpy.int64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

1458932991
```

is_backing

In [108]:
```
column_to_look = 'is_backing'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look].value_counts())
```

```
unique values:  2
unique value별 개수:
 False    251
True       1
Name: is_backing, dtype: int64
data size:  215800
dtype:  <class 'float'>
null/nan 수:  215548
------------------------------
데이터 형태 예시

False    251
True       1
Name: is_backing, dtype: int64
```

is_starrable

- 관심있는 펀딩 프로젝트에 등록 가능 유무 (관심 프로젝트로 등록하면 해당 펀딩 프로젝트의 진행과정을 계속 업데이트 받거나 추적할 수 있음)

In [107]:
```
column_to_look = 'is_starrable'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look].value_counts())
```

```
unique values:  2
unique value별 개수:
 False    210245
True       5555
Name: is_starrable, dtype: int64
data size:  215800
dtype:  <class 'numpy.bool_'>
null/nan 수:  0
------------------------------
데이터 형태 예시

False    210245
True       5555
Name: is_starrable, dtype: int64
```

is_starred

- 관심 프로젝트로 등록이 되었는지 유무

In [106]:
```
column_to_look = 'is_starred'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look].value_counts())
```

```
unique values:  2
unique value별 개수:
 False    236
True      16
Name: is_starred, dtype: int64
data size:  215800
dtype:  <class 'float'>
null/nan 수:  215548
------------------------------
데이터 형태 예시

False    236
True      16
Name: is_starred, dtype: int64
```

launched_at

- 펀딩 프로젝트 런칭 시간 (milliseconds)

In [111]:
```
column_to_look = 'launched_at'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  188222
data size:  215800
dtype:  <class 'numpy.int64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

1561560397
```

In [ ]:
```
#'launched_at' 컬럼 datetime형식으로 변경
launch_date_s = concatenated_df['launched_at']
date_launched = pd.to_datetime(launch_date_s, unit='s')
print(date_launched.head(3))
print(date_launched.tail(3))
print("first project: ", date_launched.min())
print("last project: ", date_launched.max())
# profile이 created된 날짜랑 launched된 날짜가 다른 것들 존재
```

```
0   2019-06-26 14:46:37
1   2015-08-15 04:19:27
2   2018-06-19 15:43:54
Name: launched_at, dtype: datetime64[ns]
215797   2019-06-15 12:54:21
215798   2014-07-21 20:01:35
215799   2015-03-09 17:20:06
Name: launched_at, dtype: datetime64[ns]
first project:  2009-04-25 15:36:21
last project:  2019-12-12 05:10:43
```

In [ ]:
```
#생성한 datetime 컬럼 dataframe에 삽입
concatenated_df.insert(24, 'date_launched', date_launched)
```

date_launched

- 펀딩 런칭 날짜 (datetime 형태)

In [113]:
```
column_to_look = 'date_launched'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  188222
data size:  215800
dtype:  <class 'pandas._libs.tslibs.timestamps.Timestamp'>
null/nan 수:  0
------------------------------
데이터 형태 예시

2019-06-26 14:46:37
```

location

- 펀딩 위치/지역 정보

In [115]:
```
column_to_look = 'location'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("-----------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  15473
data size:  215800
dtype:  <class 'str'>
null/nan 수:  217
-----------------------------
데이터 형태 예시

{'id': 12665488, 'name': 'Ste.-Maxime', 'slug': 'ste-maxime-var-fr', 'short_name': 'Ste.-Maxime, France', 'displayable_name': 'Ste.-Maxi
me, France', 'localized_name': 'Ste.-Maxime', 'country': 'FR', 'state': "Provence-Alpes-Cote d'Azur", 'type': 'LocalAdmin', 'is_root': F
alse, 'expanded_country': 'France', 'urls': {'web': {'discover': 'https://www.kickstarter.com/discover/places/ste-maxime-var-fr', 'locat
ion': 'https://www.kickstarter.com/locations/ste-maxime-var-fr'}, 'api': {'nearby_projects': 'https://api.kickstarter.com/v1/discover?si
gnature=1576191242.5a0210773ec3677f3f235ec1f32cc75d563f4be5&woe_id=12665488'}}}
```

In [ ]:
```
#'location'컬럼 또한 형태만 dictionary이지만 타입이 string이다
```

In [ ]:
```
#정규표현식(regular expression)으로 location 'name: ... ' 부분만 추출 시도
import re

def search_location(x):
    s_l = re.search(r"(\"name\"):(\".+?\")", x)
    print(s_l.group(2))

location_re = kickstarter_dataset['location'].apply(lambda x: search_location(x))

# 결측치로 인하여 오류
```

In [ ]:
```
# 결측치는 나중에 전처리 과정에서 처리하고
# 우선 'location'컬럼 string --> dictionary로 전환
# 결측치의 존재로 json.loads()과 eval() 둘 다 안 되기에, 다른 방법 시도
```

In [61]:
```
#결측치 부분은 그냥 스킵하고 json.loads될 수 있도록 설계
import json

def json_loads(x):
    if pd.isna(x) == False:
        j = json.loads(x)
        return j

location_dic = kickstarter_dataset['location'].apply(lambda x: json_loads(x))
location_dic.head()
```

Out[61]:
```
0    {'id': 12665488, 'name': 'Ste.-Maxime', 'slug'...
1    {'id': 2452078, 'name': 'Minneapolis', 'slug':...
2    {'id': 44418, 'name': 'London', 'slug': 'londo...
3    {'id': 2459115, 'name': 'New York', 'slug': 'n...
4    {'id': 2441116, 'name': 'Logan', 'slug': 'loga...
Name: location, dtype: object
```

In [62]:
```
print(type(location_dic[0])) #데이터 타입이 dict로 바뀜
```

```
<class 'dict'>
217
```

In [63]:
```
location_dic[0]['name'] #key 값 조회 가능
```

Out[63]:
```
'Ste.-Maxime'
```

In [86]:
```
kickstarter_data['location'] = location_dic #컬럼 대체
```

In [148]:
```
#'location' dictionary에서 'name' key만 추출 (지역 이름)
def get_loc_name(x):
    if pd.isna(x) == False:
        loc_name = x['name']
        return loc_name

location_name = kickstarter_data['location'].apply(lambda x: get_loc_name(x))

print(location_name.size)
print(type(location_name[0]))
print(location_name.isna().sum())
location_name[0]
```

```
215800
<class 'str'>
217
```

Out[148]:
```
'Ste.-Maxime'
```

In [ ]:
```
#추출한 location_name --> 데이터셋에 새 컬럼으로 삽입
kickstarter_data.insert(28, 'location_name', location_name)
```

location_name

- 펀딩 위치/지역 이름

In [116]:
```
column_to_look = 'location_name'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  13060
unique value별 개수:
 Los Angeles           9771
London                8458
New York              8124
Chicago               3886
San Francisco         3378
                      ...
Chetek                   1
Kawaihae                 1
St.-Lievens-Houtem       1
East Bloomfield          1
30005                    1
Name: location_name, Length: 13060, dtype: int64
data size:  215800
dtype:  <class 'str'>
null/nan 수:  217
------------------------------
데이터 형태 예시

Ste.-Maxime
```

name

- 펀딩 프로젝트 이름/제목

In [117]:
```
column_to_look = 'name'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  187774
unique value별 개수:
 Debut Album                                          8
Home                                                  8
A Midsummer Night's Dream                             7
Reflections                                           6
The Other Side                                        6
                                                     ..
Milan Mode (Canceled)                                 1
Get Rat Ruckus Out of the Woods!                      1
"THE PRESENT" - a short brought to you by SkyCorp®    1
AMONG WOLVES • Doc film about wild horses & bikers    1
Tumbling Bones' First Full-Length Album               1
Name: name, Length: 187774, dtype: int64
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

Louli à l'école
```

permissions

In [119]:
```
column_to_look = 'permissions'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  1
unique value별 개수:
 []    252
Name: permissions, dtype: int64
data size:  215800
dtype:  <class 'float'>
null/nan 수:  215548
------------------------------
데이터 형태 예시

nan
```

In [ ]:
```
```

photo

- 메인 이미지 정보

In [120]:
```
column_to_look = 'photo'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  188375
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

{'key': 'assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png', 'full': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876
a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=560&h=315&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=a0e9ad6ed
015ce4a5413939a086210ac', 'ed': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.
1.0&crop=faces&w=352&h=198&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=409695b0864128d046d911045e9a46b1', 'med': 'https://ksr-ugc.i
mgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=272&h=153&fit=crop&v=1561555002&au
to=format&frame=1&q=92&s=efba74ea8eb8f861704d243622e14af4', 'little': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e4
4204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=208&h=117&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=f83a4a720aacf79a13576b9
153fec385', 'small': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=fac
es&w=160&h=90&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=266bc438753183005d088f26abad1093', 'thumb': 'https://ksr-ugc.imgix.net/as
sets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=48&h=27&fit=crop&v=1561555002&auto=format&fra
me=1&q=92&s=0e552a8c7467ff6b43e50c6752a5bf61', '1024x576': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747
f_original.png?ixlib=rb-2.1.0&crop=faces&w=1024&h=576&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=e58d01a541355ca1fe4b7916af90970
d', '1536x864': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=
1552&h=873&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=8a4db126a54868018443d312ffdc0952'}
```

In [104]:
```
#'photo'컬럼 string --> dictionary 전환
photo_dic = kickstarter_data['photo'].apply(lambda x: json.loads(x))

print(type(photo_dic[0]))
print(photo_dic.isna().sum())
photo_dic[0]
```

```
<class 'dict'>
0
```

Out[104]:
```
{'key': 'assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png',
 'full': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=560&h=3
15&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=a0e9ad6ed015ce4a5413939a086210ac',
 'ed': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=352&h=198
&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=409695b0864128d046d911045e9a46b1',
 'med': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=272&h=15
3&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=efba74ea8eb8f861704d243622e14af4',
 'little': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=208&h
=117&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=f83a4a720aacf79a13576b9153fec385',
 'small': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=160&h=
90&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=266bc438753183005d088f26abad1093',
 'thumb': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=48&h=2
7&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=0e552a8c7467ff6b43e50c6752a5bf61',
 '1024x576': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=102
4&h=576&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=e58d01a541355ca1fe4b7916af90970d',
 '1536x864': 'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=155
2&h=873&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=8a4db126a54868018443d312ffdc0952'}
```

In [ ]:
```
kickstarter_data['photo'] = photo_dic #전환한 컬럼으로 대체
```

In [232]:
```
# 'photo'컬럼 string --> dictionary로 변환
# 이미지 크롤링 때 사용할 url('full' key)만 뽑아서 새 컬럼 생성
photo_link = kickstarter_data['photo'].apply(lambda x: x['full'])
print(photo_link[0])
kickstarter_data.insert(32, 'photo_link', photo_link)
```

Out[232]:
```
'https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=560&h=315&fit=cr
op&v=1561555002&auto=format&frame=1&q=92&s=a0e9ad6ed015ce4a5413939a086210ac'
```

pledged

- 총 모금액 (in 현지 통화)

In [121]:
```
column_to_look = 'pledged'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  47030
data size:  215800
dtype:  <class 'numpy.float64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

631.0
```

profile

- 펀딩 프로젝트 프로필 정보

In [123]:
```
column_to_look = 'profile'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  188390
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

{"id":3761403,"project_id":3761403,"state":"inactive","state_changed_at":1561554849,"name":null,"blurb":null,"background_color":null,"text_color":null,"link_background_color":null,"link_text_color":null,"link_text":null,"link_url":null,"show_feature_image":false,"background_image_opacity":0.8,"should_show_feature_image_section":true,"feature_image_attributes":{"image_urls":{"default":"https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=1552&h=873&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=8a4db126a54868018443d312ffdc0952","baseball_card":"https://ksr-ugc.imgix.net/assets/025/618/188/1ee876a3b9923c72e2e44204021d747f_original.png?ixlib=rb-2.1.0&crop=faces&w=560&h=315&fit=crop&v=1561555002&auto=format&frame=1&q=92&s=a0e9ad6ed015ce4a5413939a086210ac"}}}
```

In [ ]:
```
#profile 컬럼 string --> dictionary 변환
profile_dic = kickstarter_data['profile'].apply(lambda x: json_loads(x))
```

slug

- 해당 펀딩 페이지에 대한 접근 url root (url 가장 뒤에 붙는다)
- 프로젝트의 상품/서비스에 대한 간략한 설명의 구문

In [127]:
```
column_to_look = 'slug'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look].value_counts().head())
```

```
unique values:  188385
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

simon-bisley-2018-art-book                          2
make-100-guinevere-and-the-divinity-factory-1-sket  2
the-female-samurai-a-dance-research-about-japanese  2
blues-k9-kitchen                                    2
reanimated-a-short-film-based-on-the-works-of-hp-l  2
Name: slug, dtype: int64
```

source_url

- Kickstarter 웹사이트에서의 펀딩 유형/분류 별 링크

In [129]:
```
column_to_look = 'source_url'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  170
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

https://www.kickstarter.com/discover/categories/publishing
```

spotlight

- spotlight(펀딩 성공 후 해당 크리에이터가 그 펀딩 페이지를 계속해서 관리하고 뉴스, 소식, 홍보 등을 지속적으로 업데이트 할 수 있는 서비스)을 시행하고 있는지 유무

In [130]:
```
column_to_look = 'spotlight'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  2
unique value별 개수:
 True     124272
False     91528
Name: spotlight, dtype: int64
data size:  215800
dtype:  <class 'numpy.bool_'>
null/nan 수:  0
------------------------------
데이터 형태 예시

True
```

staff_pick

- “Project we love”(Kickstarter 플랫폼 제공자/웹사이트가 선택한/추천하는 프로젝트)에 선택 되었는지 여부

In [132]:
```
column_to_look = 'staff_pick'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  2
unique value별 개수:
 False    187867
True      27933
Name: staff_pick, dtype: int64
data size:  215800
dtype:  <class 'numpy.bool_'>
null/nan 수:  0
------------------------------
데이터 형태 예시

False
```

state

- 펀딩 프로젝트 상태 (성공함, 실패함, 취소됨, 현재 진행중, 강제 중단됨)
- Target(종속변수)로 사용할 변수: 성광, 실패

In [134]:
```
column_to_look = 'state'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  5
unique value별 개수:
 successful    124272
failed         75960
canceled        8857
live            6063
suspended        648
Name: state, dtype: int64
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

successful
```

state_changed_at

- 프로젝트 상태 변경 시간 (milliseconds)

In [135]:
```
column_to_look = 'state_changed_at'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  177894
data size:  215800
dtype:  <class 'numpy.int64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

1566744397
```

In [ ]:
```
#'state_changed_at' 컬럼 datetime형식으로 변경
state_changed_at_s = concatenated_df['state_changed_at']
date_ended = pd.to_datetime(state_changed_at_s, unit='s')
print(date_ended.head(3))
print(date_ended.tail(3))
print("first project: ", date_ended.min())
print("last project: ", date_ended.max())
```

```
0   2019-08-25 14:46:37
1   2015-09-14 04:19:28
2   2018-08-18 15:43:54
Name: state_changed_at, dtype: datetime64[ns]
215797   2019-07-16 06:59:00
215798   2014-08-31 22:35:00
215799   2015-04-08 17:20:08
Name: state_changed_at, dtype: datetime64[ns]
first project:  2009-05-16 10:00:25
last project:  2019-12-12 05:30:24
```

In [ ]:
```
#생성한 datetime 컬럼 dataframe에 삽입
concatenated_df.insert(37, 'date_ended', date_ended)
```

date_ended

- 프로젝트 상태가 바뀐 날짜 (datetime 형태)

In [136]:
```
column_to_look = 'date_ended'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  177894
data size:  215800
dtype:  <class 'pandas._libs.tslibs.timestamps.Timestamp'>
null/nan 수:  0
------------------------------
데이터 형태 예시

2019-08-25 14:46:37
```

static_usd_rate

- 미국 달러 환율 (미국 달러로 변환할 수 있는 값 제공)

In [138]:
```
column_to_look = 'static_usd_rate'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look].value_counts().head())
```

```
unique values:  12742
data size:  215800
dtype:  <class 'numpy.float64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

1.000000    149988
1.133748        58
1.228667        57
1.215900        56
1.115888        54
Name: static_usd_rate, dtype: int64
```

urls

- 펀딩 페이지에 대한 url 주소 정보
- keys: 'web' 그리고 'api' 두 종류

In [142]:
```
column_to_look = 'urls'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  188634
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

{'web': {'project': 'https://www.kickstarter.com/projects/469036700/louli-a-lecole?ref=discovery_category_newest', 'rewards': 'https://w
ww.kickstarter.com/projects/469036700/louli-a-lecole/rewards'}}
```

In [107]:
```
#'urls' 컬럼 string --> dictionary변환
urls_dic = kickstarter_data['urls'].apply(lambda x: json.loads(x))

print(type(urls_dic[0]))
print(urls_dic.isna().sum())
urls_dic[0]
```

```
<class 'dict'>
0
```

Out[107]:
```
{'web': {'project': 'https://www.kickstarter.com/projects/469036700/louli-a-lecole?ref=discovery_category_newest',
  'rewards': 'https://www.kickstarter.com/projects/469036700/louli-a-lecole/rewards'}}
```

In [115]:
```
kickstarter_data['urls'] = urls_dic
kickstarter_data['urls'][0]['web']
```

Out[115]:
```
{'project': 'https://www.kickstarter.com/projects/469036700/louli-a-lecole?ref=discovery_category_newest',
 'rewards': 'https://www.kickstarter.com/projects/469036700/louli-a-lecole/rewards'}
```

In [120]:
```
#'urls_dic'컬럼의 값인 dictionary안에 또 2개의 dict로 구성
#'web'외에 다른 key도 있는 것 확인
urls_keys = kickstarter_data['urls'].apply(lambda x: x.keys())
urls_keys.value_counts() # 'api' key도 있다
```

Out[120]:
```
(web)          215547
(api, web)        251
(web)               1
(api, web)          1
Name: urls, dtype: int64
```

In [131]:
```
# 'web' key 값들만 가져와 하나의 컬럼으로 만들기
urls_web = kickstarter_data['urls'].apply(lambda x: x['web'])
print(urls_web.size) #데이터 수 변경 없음
print(type(urls_web[0]))
print(urls_web.isna().sum())
urls_web[0]
```

```
215800
<class 'dict'>
0
```

Out[131]:
```
{'project': 'https://www.kickstarter.com/projects/469036700/louli-a-lecole?ref=discovery_category_newest',
 'rewards': 'https://www.kickstarter.com/projects/469036700/louli-a-lecole/rewards'}
```

In [135]:
```
#'urls_web' 새 컬럼으로 데이터셋에 삽입
kickstarter_data.insert(42, 'urls_web', urls_web)
```

urls_web

- 프로젝트 링크 정보
- 'web' key 정보들만 가져와서 만듬 컬럼
- 'web' key안에도 'project'와 'reward' 두 종류가 있음
- 이 컬럼은 굳이 안 써도 됨

In [143]:
```
column_to_look = 'urls_web'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  188634
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

{'project': 'https://www.kickstarter.com/projects/469036700/louli-a-lecole?ref=discovery_category_newest', 'rewards': 'https://www.kicks
tarter.com/projects/469036700/louli-a-lecole/rewards'}
```

In [133]:
```
#이번엔 urls_web에서 'project' key 값들만 뽑아 컬럼 생성하기
urls_project = urls_web.apply(lambda x: x['project'])
print(urls_project.size)
print(type(urls_project[0]))
print(urls_project.isna().sum())
urls_project[0]
```

```
215800
<class 'str'>
0
```

Out[133]: 'https://www.kickstarter.com/projects/469036700/louli-a-lecole?ref=discovery_category_newest'

url_project

- 프로젝트 메인 페이지 링크

In [144]:
```
column_to_look = 'url_project'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  188385
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

https://www.kickstarter.com/projects/469036700/louli-a-lecole?ref=discovery_category_newest
```

In [138]:
```
#urls_web에서 'rewards' key만 추출하여 새 컬럼 생성하기
urls_rewards = urls_web.apply(lambda x: x['rewards'])
print(urls_rewards.size)
print(type(urls_rewards[0]))
print(urls_rewards.isna().sum())
urls_rewards[0]
```

```
215800
<class 'str'>
0
```

Out[138]: 'https://www.kickstarter.com/projects/469036700/louli-a-lecole/rewards'

In [142]:
```
# 'urls_rewards' 데이터셋에 컬럼으로 삽입
kickstarter_data.insert(44, 'url_reward', urls_rewards)
```

url_reward

- 프로젝트 reward 페이지 링크

In [145]:
```
column_to_look = 'url_reward'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  188385
data size:  215800
dtype:  <class 'str'>
null/nan 수:  0
------------------------------
데이터 형태 예시

https://www.kickstarter.com/projects/469036700/louli-a-lecole/rewards
```

usd_pledged

- 총 모금액 (in 미국 달러 USD)

In [146]:
```
column_to_look = 'usd_pledged'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
#print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  84043
data size:  215800
dtype:  <class 'numpy.float64'>
null/nan 수:  0
------------------------------
데이터 형태 예시

719.75681336
```

usd_type

In [148]:
```
column_to_look = 'usd_type'
print("unique values: ", kickstarter_dataset[column_to_look].nunique())
print("unique value별 개수:\n", kickstarter_dataset[column_to_look].value_counts()) #normalize=True -- 비율 집계
print("data size: ", kickstarter_dataset[column_to_look].size)
print("dtype: ", type(kickstarter_dataset[column_to_look][0]))
print("null/nan 수: ", kickstarter_dataset[column_to_look].isna().sum())
print("------------------------------")
print("데이터 형태 예시\n")
print(kickstarter_dataset[column_to_look][0])
```

```
unique values:  2
unique value별 개수:
 domestic         164253
international     51403
Name: usd_type, dtype: int64
data size:  215800
dtype:  <class 'str'>
null/nan 수:  144
------------------------------
데이터 형태 예시

domestic
```

1차 탐색 후 데이터 크기:

- 데이터 행 수: **215,800**
- 데이터 열(컬럼) 수: **49**

In [234]:
```
kickstarter_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215800 entries, 0 to 215799
Data columns (total 49 columns):
backers_count               215800 non-null int64
blurb                       215792 non-null object
category                    215800 non-null object
category_name               215800 non-null object
category_specific           215800 non-null object
converted_pledged_amount    215800 non-null int64
country                     215800 non-null object
country_displayable_name    215800 non-null object
created_at                  215800 non-null int64
date_created                215800 non-null datetime64[ns]
creator                     215800 non-null object
currency                    215800 non-null object
currency_symbol             215800 non-null object
currency_trailing_code      215800 non-null bool
current_currency            215800 non-null object
deadline                    215800 non-null int64
date_deadline               215800 non-null datetime64[ns]
disable_communication       215800 non-null bool
friends                     252 non-null object
fx_rate                     215800 non-null float64
goal                        215800 non-null float64
id                          215800 non-null int64
is_backing                  252 non-null object
is_starrable                215800 non-null bool
is_starred                  252 non-null object
launched_at                 215800 non-null int64
date_launched               215800 non-null datetime64[ns]
location                    215583 non-null object
location_name               215583 non-null object
name                        215800 non-null object
permissions                 252 non-null object
photo                       215800 non-null object
photo_link                  215800 non-null object
pledged                     215800 non-null float64
profile                     215800 non-null object
slug                        215800 non-null object
source_url                  215800 non-null object
spotlight                   215800 non-null bool
staff_pick                  215800 non-null bool
state                       215800 non-null object
state_changed_at            215800 non-null int64
date_ended                  215800 non-null datetime64[ns]
static_usd_rate             215800 non-null float64
urls                        215800 non-null object
urls_web                    215800 non-null object
url_project                 215800 non-null object
url_reward                  215800 non-null object
usd_pledged                 215800 non-null float64
usd_type                    215656 non-null object
dtypes: bool(5), datetime64[ns](4), float64(5), int64(7), object(28)
memory usage: 73.5+ MB
```

In [235]:
```
# csv파일 저장
kickstarter_dataset.to_csv('kickstarter_dataset_updated.csv', index=False, encoding='utf-8')
```

In [60]:
```
# '런칭 날짜'에서 연도만 조회 및 새 컬럼 생성
date_launched_year = kickstarter_dataset['date_launched'].dt.year
print(date_launched_year.value_counts())
date_launched_year.value_counts(normalize=True)
```

```
2019    44158
2015    37169
2016    29066
2018    28437
2017    28134
2014    25742
2013    10089
2012     7594
2011     3811
2010     1374
2009      226
Name: date_launched, dtype: int64
```

Out[60]:
```
2019    0.204625
2015    0.172238
2016    0.134690
2018    0.131775
2017    0.130371
2014    0.119286
2013    0.046752
2012    0.035190
2011    0.017660
2010    0.006367
2009    0.001047
Name: date_launched, dtype: float64
```

In [19]:
```
kickstarter_dataset.insert(27, 'date_launched_year', date_launched_year)
```

In [124]:
```
#연도별 펀딩결과('state') 수 조회
kickstarter_dataset.groupby('date_launched_year')['state'].value_counts()
```

Out[124]:
```
date_launched_year  state
2009                successful      177
                    failed           38
                    canceled         11
2010                successful     1019
                    failed          306
                    canceled         49
2011                successful     2942
                    failed          744
                    canceled        120
                    suspended         5
2012                successful     5868
                    failed         1606
                    canceled        119
                    suspended         1
2013                successful     7978
                    failed         1938
                    canceled        173
2014                successful    13176
                    failed        10864
                    canceled       1614
                    suspended        88
2015                failed        18491
                    successful    16205
                    canceled       2178
                    suspended       295
2016                successful    14250
                    failed        13163
                    canceled       1568
                    suspended        85
2017                successful    14595
                    failed        12271
                    canceled       1203
                    suspended        65
2018                successful    18060
                    failed         9338
                    canceled        977
                    suspended        62
2019                successful    30002
                    failed         7201
                    live           6063
                    canceled        845
                    suspended        47
Name: state, dtype: int64
```

In [84]:
```
#연도별 펀딩결과('state') 비율 조회
kickstarter_dataset.groupby('date_launched_year')['state'].value_counts(normalize=True)
```

Out[84]:
```
date_launched_year  state
2009                successful    0.783186
                    failed        0.168142
                    canceled      0.048673
2010                successful    0.741630
                    failed        0.222707
                    canceled      0.035662
2011                successful    0.771976
                    failed        0.195224
                    canceled      0.031488
                    suspended     0.001312
2012                successful    0.772715
                    failed        0.211483
                    canceled      0.015670
                    suspended     0.000132
2013                successful    0.790762
                    failed        0.192090
                    canceled      0.017147
2014                successful    0.511848
                    failed        0.422034
                    canceled      0.062699
                    suspended     0.003419
2015                failed        0.497484
                    successful    0.435982
                    canceled      0.058597
                    suspended     0.007937
2016                successful    0.490264
                    failed        0.452866
                    canceled      0.053946
                    suspended     0.002924
2017                successful    0.518767
                    failed        0.436163
                    canceled      0.042760
                    suspended     0.002310
2018                successful    0.635088
                    failed        0.328375
                    canceled      0.034357
                    suspended     0.002180
2019                successful    0.679424
                    failed        0.163074
                    live          0.137302
                    canceled      0.019136
                    suspended     0.001064
Name: state, dtype: float64
```

추가적인 파생변/컬럼 생성

- 크라우드펀딩 성공에 영향을 줄 수 있을 것 같은 파생변수 생성
- 일반적인 데이터 탐색에 쓰일 수 있는 파생변수 생성

In [22]:
```
# 설정된 펀딩 기간 (데드라인 - 런칭날짜)
set_fundraising_period = kickstarter_dataset['deadline']-kickstarter_dataset['launched_at']
set_fundraising_period.shape
```

Out[22]:
```
(200232,)
```

In [24]:
```
# 런칭 날짜 (월))
date_launched_month = kickstarter_dataset['date_launched'].dt.month
date_launched_month.shape
```

Out[24]:
```
(200232,)
```

In [28]:
```
# 런칭 지연 시간 (런칭날짜 - 펀딩생성날짜)
launching_delay_time = kickstarter_dataset['launched_at']-kickstarter_dataset['created_at'] # in seconds
launching_delay_time.shape
```

Out[28]:
```
(200232,)
```

In [30]:
```
# 펀딩 목표액 in local currency --> in USD
# currency는 하나로 통일하는게 낫다고 판단
usd_goal = kickstarter_dataset['goal']*kickstarter_dataset['static_usd_rate']
usd_goal.shape
```

Out[30]:
```
(200232,)
```

In [35]:
```
# 펀딩 목표액과 설정한 펀딩 기간 비율 (펀딩 목표액/설정된 런칭 기간)
# 펀딩기간 대비 펀딩 목표액을 적절히 설정 했는지가 펀딩 성공에 영향을 줄 수 있다고 판단
target_goal_period_rate = kickstarter_dataset['usd_goal']/set_fundraising_period
target_goal_period_rate.shape
```

Out[35]:
```
(200232,)
```

In [42]:
```
# 펀딩 종료까지 실제 걸린 시간 (펀딩상태변경날짜 - 펀딩런칭날짜)
actual_time_taken = kickstarter_dataset['state_changed_at']-kickstarter_dataset['launched_at']
actual_time_taken.head()
```

Out[42]:
```
0    5184000
1    2592001
2    5184000
3    2565470
4    2592000
dtype: int64
```

In [ ]:
```
# 새로운 특성/파생변수 --> 데이터프레임 삽입 완료
```