In [4]:
```python
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
from tqdm import tqdm
import random
from collections import Counter
from scipy import interpolate
```

In [2]:
```python
def seed_everything(seed):
    random.seed(seed)
    np.random.seed(seed)
    os.environ["PYTHONHASHSEED"] = str(seed)
seed_everything(0)
```

In [54]:
```python
user_spec = pd.read_csv('./data/user_spec.csv', encoding = 'utf8')
loan_result = pd.read_csv('./data/loan_result.csv', encoding = 'utf8')
log_data = pd.read_csv('./data/log_data.csv', encoding = 'utf8')
```

In [74]:
```python
data = pd.merge(loan_result, user_spec, on = 'application_id', how = 'left')
```

In [5]:
```python
data['날짜'] = data.loanapply_insert_time.str.extract('([0-9][0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9])')
```

In [6]:
```python
data.head()
```

Out[6]:

	application_id	loanapply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied	user_id	birth_year	gender	...	company_er
0	1748340	2022-06-07 13:05:41	7	191	42000000.0	13.6	NaN	430982.0	1996.0	1.0	...	
1	1748340	2022-06-07 13:05:41	25	169	24000000.0	17.9	NaN	430982.0	1996.0	1.0	...	
2	1748340	2022-06-07 13:05:41	2	7	24000000.0	18.5	NaN	430982.0	1996.0	1.0	...	
3	1748340	2022-06-07 13:05:41	4	268	29000000.0	10.8	NaN	430982.0	1996.0	1.0	...	
4	1748340	2022-06-07 13:05:41	11	118	5000000.0	16.4	NaN	430982.0	1996.0	1.0	...	

5 rows × 24 columns

In [6]:
```python
kospi = pd.read_csv('./data/코스피지수.csv', encoding = 'utf8', usecols = [0,1])
kospi.rename(columns={"종가":"kospi"}, inplace=True)
kospi = kospi.interpolate()
kospi
```

Out[6]:

	날짜	kospi
0	2022-02-28	2699.180000
1	2022-03-01	2701.350000
2	2022-03-02	2703.520000
3	2022-03-03	2747.080000
4	2022-03-04	2713.430000
...	...	...
118	2022-06-26	2390.146667
119	2022-06-27	2401.920000
120	2022-06-28	2422.090000
121	2022-06-29	2377.990000
122	2022-06-30	2332.640000

123 rows × 2 columns

In [7]:
```python
data = pd.merge(data, kospi, on = '날짜', how = 'left')
```

In [8]:
```python
data = data.drop(columns='날짜')
```

In [10]:
```python
data.head()
```

Out[10]:

	application_id	loanapply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied	user_id	birth_year	gender	...	company_er
0	1748340	2022-06-07 13:05:41	7	191	42000000.0	13.6	NaN	430982.0	1996.0	1.0	...	2
1	1748340	2022-06-07 13:05:41	25	169	24000000.0	17.9	NaN	430982.0	1996.0	1.0	...	2
2	1748340	2022-06-07 13:05:41	2	7	24000000.0	18.5	NaN	430982.0	1996.0	1.0	...	2
3	1748340	2022-06-07 13:05:41	4	268	29000000.0	10.8	NaN	430982.0	1996.0	1.0	...	2
4	1748340	2022-06-07 13:05:41	11	118	5000000.0	16.4	NaN	430982.0	1996.0	1.0	...	2

5 rows × 24 columns

In [ ]:

컬럼별 영어 한글 변환

In [12]:
```
data.columns
```
Out[12]:
```
Index(['application_id', 'loanapply_insert_time', 'bank_id', 'product_id',
       'loan_limit', 'loan_rate', 'is_applied', 'user_id', 'birth_year',
       'gender', 'insert_time', 'credit_score', 'yearly_income', 'income_type',
       'company_enter_month', 'employment_type', 'houseown_type',
       'desired_amount', 'purpose', 'personal_rehabilitation_yn',
       'personal_rehabilitation_complete_yn', 'existing_loan_cnt',
       'existing_loan_amt', 'kospi'],
      dtype='object')
```
In [13]:
```
set(data['income_type'])
```
Out[13]:
```
{'EARNEDINCOME',
 'EARNEDINCOME2',
 'FREELANCER',
 'OTHERINCOME',
 'PRACTITIONER',
 'PRIVATEBUSINESS',
 nan}
```
In [14]:
```
set(data['employment_type'])
```
Out[14]:
```
{nan, '계약직', '기타', '일용직', '정규직'}
```
In [15]:
```
set(data['houseown_type'])
```
Out[15]:
```
{nan, '기타가족소유', '배우자', '자가', '전월세'}
```
In [16]:
```
set(data['purpose'])
```
Out[16]:
```
{'BUSINESS',
 'BUYCAR',
 'BUYHOUSE',
 'ETC',
 'HOUSEDEPOSIT',
 'INVEST',
 'LIVING',
 'SWITCHLOAN',
 nan,
 '기타',
 '대환대출',
 '사업자금',
 '생활비',
 '자동차구입',
 '전월세보증금',
 '주택구입',
 '투자'}
```
In [9]:
```
data.loc[data['purpose']=="BUSINESS","purpose"]="사업자금"
data.loc[data['purpose']=="BUYCAR","purpose"]="자동차구입"
data.loc[data['purpose']=="BUYHOUSE","purpose"]="주택구입"
data.loc[data['purpose']=="ETC","purpose"]="기타"
data.loc[data['purpose']=="HOUSEDEPOSIT","purpose"]="전월세보증금"
data.loc[data['purpose']=="INVEST","purpose"]="투자"
data.loc[data['purpose']=="LIVING","purpose"]="생활비"
data.loc[data['purpose']=="SWITCHLOAN","purpose"]="대환대출"
```
In [18]:
```
set(data['purpose'])
```
Out[18]:
```
{nan, '기타', '대환대출', '사업자금', '생활비', '자동차구입', '전월세보증금', '주택구입', '투자'}
```

In [ ]:

파생컬럼 - log_length

In [23]:
```
log_data.head()
```

Out[23]:

	user_id	event	timestamp	mp_os	mp_app_version	date_cd
0	576409	StartLoanApply	2022-03-25 11:12:09	Android	3.8.2	2022-03-25
1	576409	ViewLoanApplyIntro	2022-03-25 11:12:09	Android	3.8.2	2022-03-25
2	72878	EndLoanApply	2022-03-25 11:14:44	Android	3.8.4	2022-03-25
3	645317	OpenApp	2022-03-25 11:15:09	iOS	3.6.1	2022-03-25
4	645317	UseLoanManage	2022-03-25 11:15:11	iOS	3.6.1	2022-03-25

In [7]:
```
log_data = log_data.sort_values(by=['user_id', 'timestamp'])
```

In [14]:
```
log_data.head(20)
```

Out[14]:

	user_id	event	timestamp	mp_os	mp_app_version	date_cd
11709372	1	GetCreditInfo	2022-05-03 14:52:28	android	464	2022-05-03
11709374	1	GetCreditInfo	2022-05-03 14:52:35	android	464	2022-05-03
2451691	1	UseLoanManage	2022-06-16 23:58:41	Android	3.12.1	2022-06-16
2451693	1	Login	2022-06-16 23:58:41	Android	3.12.1	2022-06-16
7071607	1	GetCreditInfo	2022-06-16 23:58:42	android	464	2022-06-16
10428909	7	GetCreditInfo	2022-05-22 16:39:49	android	465	2022-05-22
9627339	9	GetCreditInfo	2022-05-21 23:37:58	android	465	2022-05-21
9627368	9	GetCreditInfo	2022-05-21 23:43:33	android	465	2022-05-21
9627370	9	GetCreditInfo	2022-05-21 23:43:52	android	465	2022-05-21
9505105	11	OpenApp	2022-03-24 10:53:59	iOS	3.6.1	2022-03-24
13238918	11	GetCreditInfo	2022-03-24 10:54:07	iOS	3.6.1	2022-03-24
9505106	11	UseLoanManage	2022-03-24 10:54:08	iOS	3.6.1	2022-03-24
13238924	11	GetCreditInfo	2022-03-24 10:54:19	iOS	3.6.1	2022-03-24
9505107	11	UsePrepayCalc	2022-03-24 10:54:36	iOS	3.6.1	2022-03-24
9505108	11	StartLoanApply	2022-03-24 10:55:43	iOS	3.6.1	2022-03-24
9505109	11	ViewLoanApplyIntro	2022-03-24 10:55:43	iOS	3.6.1	2022-03-24
9505110	11	CompleteIDCertification	2022-03-24 10:56:26	iOS	3.6.1	2022-03-24
9505111	11	EndLoanApply	2022-03-24 10:59:46	iOS	3.6.1	2022-03-24
9505112	11	UseLoanManage	2022-03-24 11:05:31	iOS	3.6.1	2022-03-24
9505113	11	UseLoanManage	2022-03-24 11:05:31	iOS	3.6.1	2022-03-24

In [8]:
```
counter = Counter((log_data['user_id']))
```

In [9]:
```
log_set = list(set(log_data['user_id']))
```

In [10]:
```
log_user_sort = list(counter.most_common())
```

In [1]:
```
log_user_list = []
log_length_list = []
for i in range(len(log_user_sort)):
    #print(log_user_sort[i])
    log_user_list.append(log_user_sort[i][0])
    log_length_list.append(log_user_sort[i][1])
```

In [15]:
```
log_length_df = pd.concat([pd.DataFrame(log_user_list),pd.DataFrame(log_length_list)], axis=1)
```

In [16]:
```
log_length_df.columns = ['user_id', 'log_length']
```

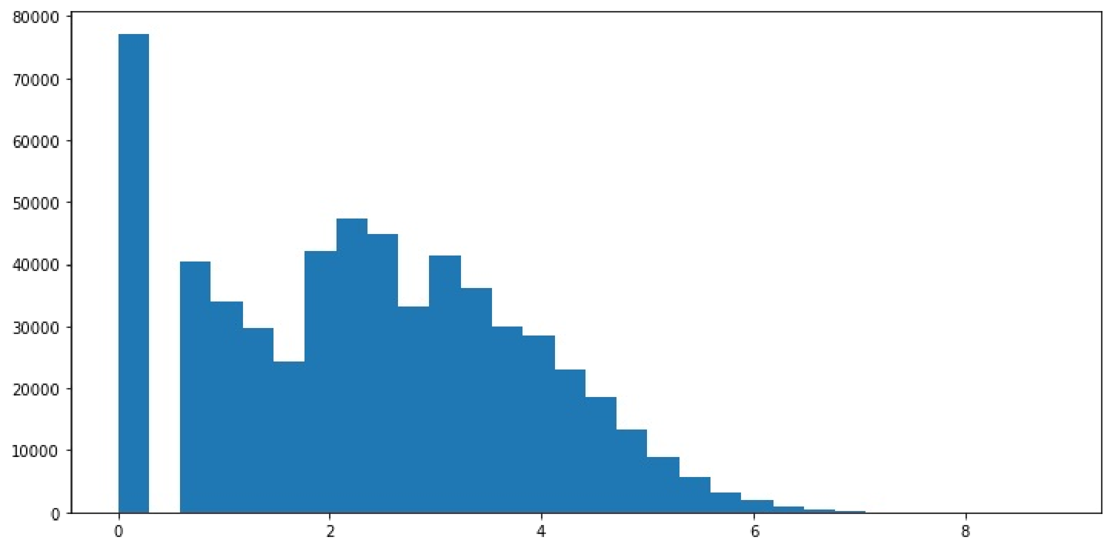
In [17]:
```
data = pd.merge(data, log_length_df, on = 'user_id', how = 'left')
data.head()
```

Out[17]:

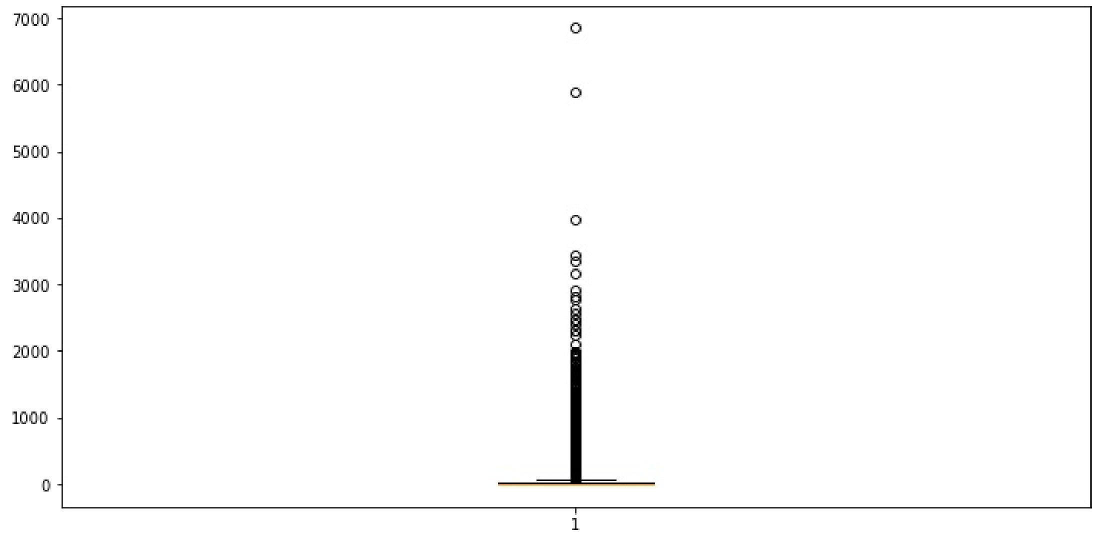
	application_id	loanapply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied	user_id	birth_year	gender	...	employment
0	1748340	2022-06-07 13:05:41	7	191	42000000.0	13.6	NaN	430982.0	1996.0	1.0	...	[illegible]
1	1748340	2022-06-07 13:05:41	25	169	24000000.0	17.9	NaN	430982.0	1996.0	1.0	...	[illegible]
2	1748340	2022-06-07 13:05:41	2	7	24000000.0	18.5	NaN	430982.0	1996.0	1.0	...	[illegible]
3	1748340	2022-06-07 13:05:41	4	268	29000000.0	10.8	NaN	430982.0	1996.0	1.0	...	[illegible]
4	1748340	2022-06-07 13:05:41	11	118	5000000.0	16.4	NaN	430982.0	1996.0	1.0	...	[illegible]

5 rows × 25 columns

In [26]:
```python
plt.figure(figsize=(12, 6))
plt.hist(np.log(log_length_list), bins=30)
plt.show()
```



In [27]:
```python
plt.figure(figsize=(12, 6))
plt.boxplot(log_length_list)
plt.show()
```



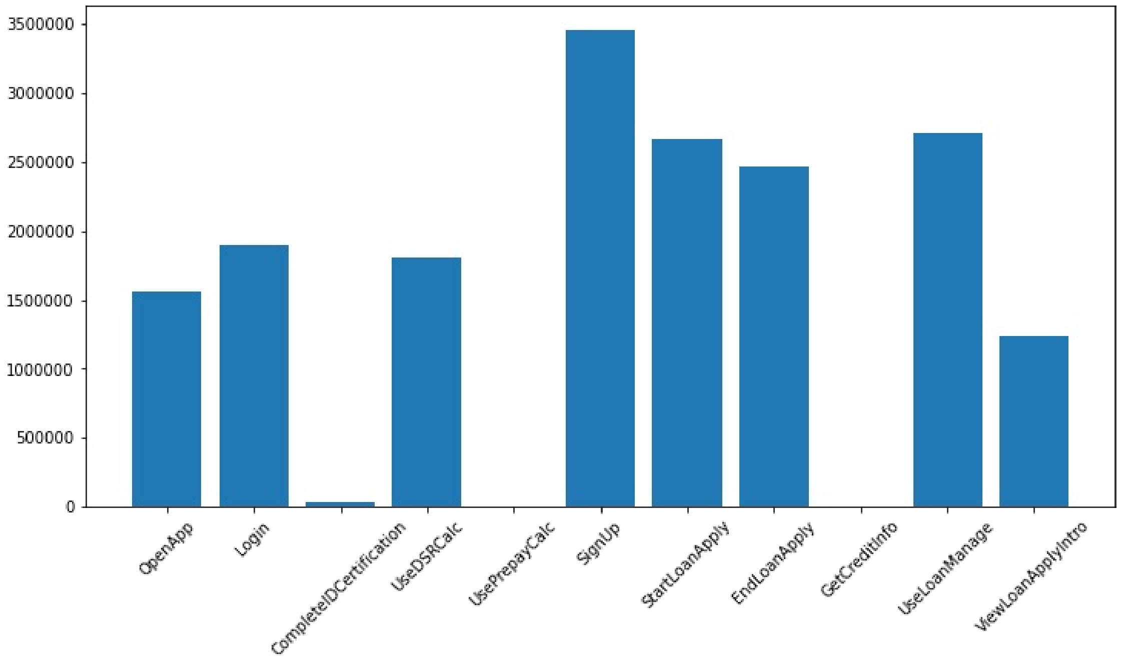
파생컬럼 - user별 각 log event의 횟수

In [18]:
```python
event_set = list(set(log_data['event']))
```

In [19]:
```python
event_num = []
for i in range(len(event_set)):
    print(event_set[i], len(log_data[log_data['event']==event_set[i]]))
    event_num.append(len(log_data[log_data['event']==event_set[i]]))
```

```
GetCreditInfo 2661997
Login 2463755
SignUp 34892
CompleteIDCertification 1237777
OpenApp 3460762
EndLoanApply 2715253
UseDSRCalc 4665
ViewLoanApplyIntro 1804712
StartLoanApply 1893914
UseLoanManage 1558906
UsePrepayCalc 7360
```

In [31]:
```
plt.figure(figsize=(12, 6))
plt.bar(np.arange(len(event_num)), event_num)
plt.ticklabel_format(style='plain')
plt.xticks([0, 1, 2,3,4,5,6,7,8,9,10], ['OpenApp', 'Login', 'CompleteIDCertification','UseDSRCalc','UsePrepayCa
                                        ,'SignUp','StartLoanApply','EndLoanApply','GetCreditInfo','UseLoanManag
plt.show()
```



In [20]:
```
c1 = data['user_id']
```

In [21]:
```
c1=c1.drop_duplicates()
c1 = c1.reset_index(drop=True)
c2 = np.zeros(len(c1))
```

In [40]:
```
log_df = pd.concat([pd.DataFrame(c1),pd.DataFrame(c2),pd.DataFrame(c2),pd.DataFrame(c2),pd.DataFrame(c2),pd.Dat
log_df.columns = ['user_id','OpenApp', 'Login', 'CompleteIDCertification','UseDSRCalc','UsePrepayCalc'
                                        ,'SignUp','StartLoanApply','EndLoanApply','GetCreditInfo','UseLoanManag
```

In [41]:
```
log_df
log_df.set_index('user_id',inplace=True)
log_df
```

Out[41]:

	OpenApp	Login	CompleteIDCertification	UseDSRCalc	UsePrepayCalc	SignUp	StartLoanApply	EndLoanApply	GetCreditInfo	Usel
user_id										
430982.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
345273.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3058.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
181137.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
197454.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	
564079.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
364214.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
77460.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
876482.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
31658.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

317468 rows × 11 columns

In [42]:
```
bab = log_df.index
```

```
In [42]: bab = log_df.index
```

```
In [43]: for i in tqdm(bab):
             vvv = log_data[log_data['user_id']==i]
             log_df.loc[i,Counter(vvv.iloc[:,1]).keys()]=Counter(vvv.iloc[:,1]).values()
```

```
100%|████████████████████████████████████████████████████████████████████████| 317468/317468 [2:51:23<00:00, 30
.87it/s]
```

```
In [46]: log_df = log_df.reset_index()
         log_df
```

Out[46]:

	user_id	OpenApp	Login	CompleteIDCertification	UseDSRCalc	UsePrepayCalc	SignUp	StartLoanApply	EndLoanApply	GetCreditI
0	430982.0	42.0	0.0	15.0	0.0	0.0	0.0	50.0	50.0	
1	345273.0	18.0	19.0	8.0	0.0	0.0	0.0	17.0	28.0	
2	3058.0	2.0	3.0	3.0	0.0	0.0	0.0	1.0	2.0	
3	181137.0	3.0	0.0	4.0	0.0	0.0	0.0	5.0	4.0	
4	197454.0	1.0	0.0	1.0	0.0	0.0	0.0	2.0	2.0	
...	...	...	...	...	...	...	...	...	...	
317463	564079.0	15.0	12.0	4.0	0.0	0.0	0.0	2.0	4.0	1
317464	364214.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	
317465	77460.0	38.0	32.0	5.0	0.0	0.0	0.0	7.0	12.0	1
317466	876482.0	2.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	
317467	31658.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	

317468 rows × 12 columns

```
In [47]: data = pd.merge(data, log_df, on = 'user_id', how = 'left')
         data.head()
```

Out[47]:

	application_id	loanapply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied	user_id	birth_year	gender	...	Login	Com
0	1748340	2022-06-07 13:05:41	7	191	42000000.0	13.6	NaN	430982.0	1996.0	1.0	...	0.0	
1	1748340	2022-06-07 13:05:41	25	169	24000000.0	17.9	NaN	430982.0	1996.0	1.0	...	0.0	
2	1748340	2022-06-07 13:05:41	2	7	24000000.0	18.5	NaN	430982.0	1996.0	1.0	...	0.0	
3	1748340	2022-06-07 13:05:41	4	268	29000000.0	10.8	NaN	430982.0	1996.0	1.0	...	0.0	
4	1748340	2022-06-07 13:05:41	11	118	5000000.0	16.4	NaN	430982.0	1996.0	1.0	...	0.0	

5 rows × 36 columns

```
In [48]: data.to_csv('./data/rawdata_tmp.csv', index=False, encoding = 'utf-8')
```

```
In [ ]:
```

파생컬럼 - application_id 별 금리순, 한도순 순위

```
In [52]: rawdata_tmp = data
```

```
In [58]: rawdata_tmp.head()
```

Out[58]:

	application_id	loanapply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied	user_id	birth_year	gender	...	Login	Com
0	1748340	2022-06-07 13:05:41	7	191	42000000.0	13.6	NaN	430982.0	1996.0	1.0	...	0.0	
1	1748340	2022-06-07 13:05:41	25	169	24000000.0	17.9	NaN	430982.0	1996.0	1.0	...	0.0	
2	1748340	2022-06-07 13:05:41	2	7	24000000.0	18.5	NaN	430982.0	1996.0	1.0	...	0.0	
3	1748340	2022-06-07 13:05:41	4	268	29000000.0	10.8	NaN	430982.0	1996.0	1.0	...	0.0	
4	1748340	2022-06-07 13:05:41	11	118	5000000.0	16.4	NaN	430982.0	1996.0	1.0	...	0.0	

5 rows × 36 columns

```
In [55]: tmp = rawdata_tmp[['application_id','bank_id','product_id','loan_limit','loan_rate']]
         tmp
```

Out[55]:

	application_id	bank_id	product_id	loan_limit	loan_rate
0	1748340	7	191	42000000.0	13.6
1	1748340	25	169	24000000.0	17.9
2	1748340	2	7	24000000.0	18.5
3	1748340	4	268	29000000.0	10.8
4	1748340	11	118	5000000.0	16.4
...	...	...	...	...	...
13527358	1428218	62	200	3000000.0	14.8
13527359	1428218	2	7	40000000.0	11.8
13527360	1428218	32	257	15000000.0	7.2
13527361	1428218	33	110	44000000.0	13.5
13527362	1428218	5	194	44000000.0	9.7

13527363 rows × 5 columns

In [56]:

```
app_list = list(dict.fromkeys(tmp['application_id']))
app_list
```

Out[56]:

```
[1748340,
 830336,
 728546,
 1641986,
 937515,
 629707,
 691052,
 2083853,
 1410836,
 1399034,
 560448,
 2155640,
 953094,
 1311433,
 68170,
 1797950,
 632967,
 1747456,
 1065758,
 1117883,
 385410,
 2021320,
 107628,
 2019890,
 2112580,
 2045163,
 2146983,
 2088127,
 1882657,
 226887,
 175852,
 1029177,
 1545691,
 1798392,
 1138885,
 97844,
 282459,
 1901749,
 1386770,
 681506,
 2097522,
 101440,
 2157489,
 2006317,
 1351600,
 1976826,
 1870283,
 1677450,
 2052716,
 2155379,
 1397132,
 900266,
 763283,
 2143794,
 1991412,
 1211448,
 424084,
 857133,
 1172882,
 1615033,
 337366,
 1966421,
 854546,
 1846802,
 1162353,
 43981,
 1505370,
 1808664,
 295053,
 990662,
 1963697,
 310991,
 1487870,
 1918113,
 1242350,
 1466123,
 658266,
 1464334,
 1429944,
 1996916,
 377700,
 10554,
 722245,
 1530135,
 1129704,
 31316,
 552022,
 641366,
 89073,
 1553085,
 1462160,
 1743530,
 2020704,
 1240130,
 1568216,
 2089615,
 1571170,
 273609,
 2088759,
 162637,
 1975382,
 1079867,
 895125,
 1238489,
 1050845,
 1834816,
 1639302,
 1661534,
 1363660,
 211455,
 4927,
 1843038,
 1041004,
 1665656,
 615238,
 691694,
 26224,
 1747446,
 609100,
 1579613,
 1149543,
 343761,
 1920272,
 401407,
 1744717,
 938588,
 825617,
 1387680,
 814728,
 760082,
 1702422,
 2029455,
 1072621,
 1793429,
 925589,
 2141799,
 879672,
 49341,
 751424,
 1238961,
 1171317,
 1592557,
 1979067,
 1537220,
 1736394,
 844938,
 1155932,
 1380436,
 1121556,
 1933505,
 1505352,
 818792,
 429231,
 1081791,
 111471,
 1079715,
 1297061,
 284397,
 160707,
 936696,
 2056056,
 1239123,
 624686,
 1041593,
 2166886,
 706302,
 610121,
 381811,
 1482593,
 1338110,
 1749216,
 1370837,
 421433,
 294432,
 251743,
 2148692,
 1493351,
 218391,
 1930872,
 1473809,
 1240559,
 1524572,
 1306703,
 1167318,
 71353,
 554546,
 1375565,
 936574,
 1850725,
 1340478,
 916677,
 1407872,
 1983880,
 88566,
 286488,
 420921,
 2059921,
 158814,
 1077671,
 1736166,
 1829520,
 550758,
 909801,
 1085479,
 1242777,
 474672,
 1631437,
 1530545,
 488257,
 1850503,
 1419203,
 1924848,
 1498476,
 239746,
 1105548,
 1125146,
 1009628,
 619077,
 1620228,
 73673,
 1825259,
 52661,
 323935,
 1835361,
 2017166,
 712675,
 246703,
 266775,
 116144,
 658665,
 1706307,
 1434248,
 1618885,
 1681966,
 1640236,
 118389,
 606218,
 1143706,
 54042,
 514203,
 459378,
 476548,
 1475594,
 670918,
 352759,
 2065157,
 1941765,
 247122,
 753548,
 1543219,
 40101,
 1947034,
 1628924,
 1493090,
 32242,
 1984511,
 437060,
 1532527,
 1167492,
 82201,
 848264,
 746178,
 725023,
 1136315,
 487049,
 2037203,
 668185,
 1447809,
 481462,
 804193,
 1025183,
 1390034,
 1903112,
 1920092,
 1188554,
 2138284,
 1478521,
 2116744,
 818581,
 1252673,
 1706047,
 2136975,
 2161564,
 1837804,
 75982,
 244602,
 262288,
 1540597,
 1624627,
 1470127,
 307465,
 1198909,
 1213779,
 591446,
 803334,
 1760944,
 886754,
 241165,
 1346121,
 1005429,
 1424308,
 1193176,
 1279389,
 324087,
 1951747,
 843802,
 368880,
 354357,
 1325375,
 1732592,
 1794312,
 1086409,
 1128213,
 1675848,
 1269027,
 881301,
 595300,
 929918,
 1795944,
 1587365,
 2141999,
 1487004,
 699925,
 2089709,
 508879,
 1007890,
 2119790,
 1695493,
 399254,
 405128,
 147433,
 1706185,
 1693877,
 1669169,
 653505,
 122182,
 1597384,
 1776381,
 65266,
 1053528,
 907586,
 1926556,
 977739,
 1467079,
 1092542,
 285225,
 2127205,
 1646617,
 676492,
 921634,
 789392,
 942508,
 819372,
 2105539,
 314963,
 1075068,
 1431435,
 774108,
 350977,
 1901567,
 1144359,
 473194,
 2089734,
 107792,
 226032,
 794241,
 1389334,
 1499341,
 895329,
 350605,
 245798,
 2089579,
 831354,
 1847075,
 1949521,
 1195513,
 1569500,
 1958238,
 120828,
 62235,
 1833187,
 648120,
 557937,
 894444,
 493052,
 296938,
 2129095,
 1095411,
 1250628,
 873074,
 377536,
 1976189,
 1624439,
 1360934,
 1470091,
 1542649,
 1451817,
 696374,
 148229,
 177733,
 936773,
 556261,
 1298956,
 870472,
 366887,
 1611527,
 731419,
 99084,
 1656412,
 569067,
 518590,
 2077738,
 1510164,
 1360067,
 1577161,
 700750,
 855619,
 1819995,
 2153206,
 2138863,
 1060863,
 1558052,
 1778785,
 1923395,
 1036400,
 1165140,
 1844678,
 1396375,
 1160035,
 1879689,
 1875052,
 1828998,
 1936187,
 1959912,
 1940454,
 337566,
 1843238,
 265301,
 1967760,
 1238450,
 571144,
 113481,
 1545962,
 1495142,
 1740560,
 1335040,
 634607,
 1633967,
 373087,
 192432,
 1419098,
 1031223,
 689194,
 41554,
 1669540,
 2119248,
 736381,
 103684,
 1058684,
 328000,
 872192,
 164687,
 1220504,
 82205,
 1326327,
 73756,
 2112935,
 130357,
 1604500,
 86898,
 2085833,
 264874,
 1879164,
 2122989,
 957539,
 1272741,
 1193826,
 1671167,
 2134455,
 1649029,
 1525024,
 1558484,
 465747,
 278107,
 129266,
 1359170,
 329223,
 2137639,
 1287717,
 1004969,
 1218390,
 2002456,
 1927006,
 1155709,
 1856263,
 1012309,
 1732700,
 1403572,
 1851394,
 1316830,
 1700594,
 873353,
 1020969,
 1594821,
 1164964,
 440451,
 1979240,
 308680,
 1615181,
 869524,
 175416,
 898778,
 927512,
 1178413,
 621216,
 1978415,
 890459,
 1737223,
 232848,
 1949948,
 520256,
 1120124,
 1499615,
 2009245,
 2155607,
 1828212,
 1848189,
 1989886,
 1203019,
 1299327,
 87662,
 1241490,
 1144623,
 1650002,
 1241995,
 690946,
 579940,
 1125394,
 595479,
 1121787,
 1913052,
 1551945,
 787591,
 1754728,
 534441,
 1245667,
 242840,
 1518012,
 2022232,
 123763,
 1681383,
 1697175,
 826501,
 1310355,
 218323,
 793031,
 538044,
 1705531,
 942839,
 479702,
 1338654,
 882346,
 1537243,
 2046433,
 316531,
 1242641,
 1458293,
 83214,
 1325112,
 490956,
 1579880,
 1507648,
 1461223,
 205522,
 2148879,
 194869,
 447492,
 1930253,
 1359277,
 486716,
 342817,
 1765657,
 1729955,
 533645,
 460612,
 470416,
 348326,
 969906,
 1408583,
 1999230,
 1773876,
 1295389,
 201603,
 341963,
 1436122,
 454366,
 303207,
 1896042,
 1383785,
 847475,
 209327,
 962186,
 253031,
 31081,
 1078928,
 311486,
 516636,
 1242727,
 1915776,
 938542,
 2083483,
 914470,
 975942,
 1857176,
 1483856,
 1131046,
 1175282,
 1683573,
 969387,
 1897879,
 122510,
 1501714,
 942388,
 510024,
 426820,
 1361069,
 1429734,
 260060,
 75489,
 1859434,
 1830373,
 206612,
 1711739,
 1410269,
 253178,
 1975542,
 1762904,
 426488,
 971685,
 978513,
 654592,
 649416,
 1062916,
 1337592,
 604073,
 2029755,
 423440,
 872747,
 277115,
 1403503,
 590701,
 1987655,
 1099590,
 631742,
 701360,
 1778567,
 735357,
 566141,
 2107772,
 1774293,
 867199,
 2078562,
 700874,
 1399030,
 1003893,
 585554,
 24929,
 1600285,
 1062439,
 30451,
 555199,
 1882563,
 1291816,
 2048659,
 830810,
 917270,
 1828164,
 1809140,
 2015939,
 2155867,
 854685,
 1233080,
 109653,
 623546,
 1150372,
 664508,
 1349328,
 1800993,
 33665,
 593544,
 164891,
 394544,
 312452,
 1195421,
 2116429,
 1136859,
 1956612,
 995748,
 121353,
 2071910,
 1557685,
 1187540,
 1342211,
 126111,
 45361,
 1148482,
 1046699,
 1009235,
 1718629,
 1694871,
 2038902,
 876014,
 1978312,
 1723741,
 14438,
 2089521,
 1457680,
 1081205,
 827987,
 1899992,
 1818490,
 1604299,
 821259,
 1893435,
 745900,
 670081,
 503621,
 575071,
 2003650,
 2060233,
 937702,
 1393528,
 865591,
 1727674,
 1043499,
 1494490,
 926622,
 453768,
 1940587,
 1018395,
 1025158,
 1467273,
 10284,
 925342,
 1596367,
 1153588,
 770999,
 1413329,
 196163,
 2157040,
 1485255,
 1143790,
 2165870,
 1824860,
 887347,
 650407,
 10512,
 911133,
 397712,
 326817,
 1833541,
 1876007,
 1230175,
 1389552,
 1510877,
 1590277,
 2113790,
 622158,
 1892527,
 971584,
 1698528,
 227352,
 864523,
 2104521,
 964027,
 535123,
 1753477,
 590832,
 1559400,
 1326056,
 1521787,
 645036,
 1390851,
 872965,
 244457,
 1494852,
 28489,
 596796,
 353841,
 41282,
 976831,
 1249141,
 1797191,
 519610,
 1903958,
 132801,
 595914,
 1847996,
 1718549,
 1737556,
 1012032,
 1388540,
 313511,
 609002,
 332755,
 889105,
 220920,
 921293,
 1504364,
 588720,
 1097553,
 493541,
 1040918,
 1059611,
 658801,
 1918765,
 1030047,
 1899799,
 185746,
 1615026,
 97848,
 1741168,
 1247398,
 1365134,
 2121294,
 123876,
 529851,
 2057292,
 1988708,
 139650,
 1323403,
 107002,
 808938,
 1556346,
 1045216,
 929049,
 1768559,
 270111,
 1643178,
 648004,
 430299,
 167373,
 486709,
 1304167,
 95011,
 179070,
 944843,
 786546,
 1514582,
 1086095,
 189676,
 902110,
 75974,
 1091360,
 1469959,
 102708,
 459477,
 922562,
 1995106,
 2053324,
 2059398,
 1417715,
 1777015,
 1954125,
 1689933,
 1962219,
 692930,
 1496856,
 1114999,
 972127,
 297402,
 768339,
 1506677,
 2052368,
 1191560,
 568453,
 459484,
 1661438,
 626140,
 943757,
 2028530,
 907704,
 937214,
 1992426,
 806503,
 1771789,
 1483696,
 2094217,
 2037067,
 242451,
 371131,
 1343795,
 1750924,
 2157865,
 576643,
 2136706,
 380021,
 949973,
 679142,
 1831090,
 1123617,
 178618,
 65346,
 508290,
 376364,
 1835670,
 1637434,
 1103808,
 2083697,
 618657,
 632202,
 147706,
 1913640,
 1756892,
 1694756,
 150384,
 284204,
 507504,
 548168,
 152054,
 2164015,
 1334350,
 114970,
 298967,
 1596273,
 1306495,
 478347,
 1268976,
 948866,
 1814257,
 1093287,
 395886,
 1141510,
 1001851,
 1382679,
 912932,
 2072355,
 750990,
 1875216,
 1575482,
 1664568,
 306421,
 454648,
 1144381,
 879220,
 436033,
 1192512,
 491923,
 2073269,
 1262644,
 857614,
 152866,
 380971,
 493989,
 286153,
 1753,
 655994,
 1281960,
 1747665,
 1057882,
 17730,
 1949132,
 1033717,
 2067149,
 1316949,
 349943,
 1954630,
 1986840,
 344253,
 1161882,
 138330,
 82717,
 330152,
 847844,
 1802142,
 1158508,
 598440,
 129742,
 924561,
 456144,
 40990,
 1815319,
 895920,
 1398986,
 998648,
 1496229,
 167285,
 28101,
 282266,
 2125830,
 1779190,
 1591772,
 1543366,
 1759253,
 1978734,
 615173,
 1864227,
 1229698,
 385835,
 1160489,
 ...]
```

In [57]:
```
rank_df = pd.DataFrame()

for i in tqdm(range(len(app_list))):
    ttmp = tmp[tmp['application_id']==app_list[i]][['loan_limit','loan_rate']]
    ttmp_rank = ttmp.rank(ascending=False)
    ttmp_df = pd.concat([ttmp,ttmp_rank],axis=1)
    ttmp_df.columns = ['loan_limit','loan_rate','loan_limit_rank','loan_rate_rank']
    rank_df = pd.concat([rank_df,ttmp_df],axis=0)
```

```
100%|████████████████████████████████████████████████████████████████████████| 968866/968866 [27:49:39<00:00,  9
.67it/s]
```

In [59]:
```
rank_df.head()
```

Out[59]:

	loan_limit	loan_rate	loan_limit_rank	loan_rate_rank
0	42000000.0	13.6	3.0	22.5
1	24000000.0	17.9	16.0	5.0
2	24000000.0	18.5	16.0	2.0
3	29000000.0	10.8	10.0	33.0
4	5000000.0	16.4	31.5	8.5

In [62]:
```
rank_df2 = rank_df.drop(['loan_limit','loan_rate'],axis=1, inplace=False)
rank_df2
```

Out[62]:

	loan_limit_rank	loan_rate_rank
0	3.0	22.5
1	16.0	5.0
2	16.0	2.0
3	10.0	33.0
4	31.5	8.5
...	...	...
13527358	37.0	11.0
13527359	10.5	23.0
13527360	24.5	35.0
13527361	5.0	16.0
13527362	5.0	26.0

13527363 rows × 2 columns

In [63]:

```python
data = pd.merge(rawdata_tmp, rank_df2, left_index=True, right_index=True, how='left')
data.head()
```

Out[63]:

	application_id	loanapply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied	user_id	birth_year	gender	...	UseDSRCalc
0	1748340	2022-06-07 13:05:41	7	191	42000000.0	13.6	NaN	430982.0	1996.0	1.0	...	0.0
1	1748340	2022-06-07 13:05:41	25	169	24000000.0	17.9	NaN	430982.0	1996.0	1.0	...	0.0
2	1748340	2022-06-07 13:05:41	2	7	24000000.0	18.5	NaN	430982.0	1996.0	1.0	...	0.0
3	1748340	2022-06-07 13:05:41	4	268	29000000.0	10.8	NaN	430982.0	1996.0	1.0	...	0.0
4	1748340	2022-06-07 13:05:41	11	118	5000000.0	16.4	NaN	430982.0	1996.0	1.0	...	0.0

5 rows × 38 columns

In [65]:

```python
data.isnull().sum()
```

Out[65]:

```
application_id                            0
loanapply_insert_time                     0
bank_id                                   0
product_id                                0
loan_limit                             7495
loan_rate                              7495
is_applied                          3257239
user_id                                 113
birth_year                           128209
gender                               128209
insert_time                             113
credit_score                        1509389
yearly_income                           119
income_type                             113
company_enter_month                  400450
employment_type                         113
houseown_type                           113
desired_amount                          113
purpose                                 113
personal_rehabilitation_yn          5888814
personal_rehabilitation_complete_yn 11794090
existing_loan_cnt                   2685822
existing_loan_amt                   3890276
kospi                                     0
log_length                           488164
OpenApp                                   0
Login                                     0
CompleteIDCertification                   0
UseDSRCalc                                0
UsePrepayCalc                             0
SignUp                                    0
StartLoanApply                            0
EndLoanApply                              0
GetCreditInfo                             0
UseLoanManage                             0
ViewLoanApplyIntro                        0
loan_limit_rank                        7495
loan_rate_rank                         7495
dtype: int64
```

In [ ]:

결측치 처리

In [34]:
```
df = data
```

In [29]:
```
df.drop(columns=['personal_rehabilitation_yn',
       'personal_rehabilitation_complete_yn'],axis=1, inplace =True)
```

In [30]:
```
df1 = df[df['is_applied'].isnull()==False]
```

In [31]:
```
df1.dropna(inplace=True)
```

```
C:\Users\sjkan\AppData\Local\Temp\ipykernel_6552\3614008390.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df1.dropna(inplace=True)
```

In [14]:
```
len(df1)
```

Out[14]:
```
6710290
```

In [67]:
```
df1
```

Out[67]:

	application_id	loanapply_insert_time	bank_id	product_id	loan_limit	loan_rate	is_applied	user_id	birth_year	gender	...	Usel
13284	2157865	2022-05-09 08:44:59	54	235	20000000.0	16.5	1.0	346970.0	1970.0	1.0	...	
13285	576643	2022-05-09 10:54:53	54	235	11000000.0	16.5	0.0	545882.0	1977.0	1.0	...	
13286	576643	2022-05-09 10:54:53	11	118	3000000.0	20.0	0.0	545882.0	1977.0	1.0	...	
13287	2136706	2022-05-09 10:41:06	42	216	10000000.0	13.5	0.0	558819.0	1983.0	1.0	...	
13288	2136706	2022-05-09 10:41:07	25	169	22000000.0	15.9	0.0	558819.0	1983.0	1.0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
13519634	1969227	2022-05-16 14:42:58	2	7	30000000.0	13.6	0.0	109899.0	1977.0	1.0	...	
13519635	1969227	2022-05-16 14:42:57	33	110	9000000.0	14.4	0.0	109899.0	1977.0	1.0	...	
13519636	1969227	2022-05-16 14:42:56	50	142	3000000.0	11.2	0.0	109899.0	1977.0	1.0	...	
13519637	1969227	2022-05-16 14:43:18	22	100	4000000.0	15.3	0.0	109899.0	1977.0	1.0	...	
13519638	1969227	2022-05-16 14:42:56	19	231	9000000.0	15.5	0.0	109899.0	1977.0	1.0	...	

6710290 rows × 36 columns

In [69]:
```
loan = df1

loan = loan[['bank_id','is_applied']]
loan.dropna(inplace=True)
a = loan.groupby('bank_id').count()
b = loan.groupby('bank_id').sum()
c = b/a
c = c.rename(columns={'is_applied':'bank_ratio'})
d = pd.merge(loan,c, on = 'bank_id', how = 'left')
```

```
C:\Users\sjkan\AppData\Local\Temp\ipykernel_6552\2531176081.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  loan.dropna(inplace=True)
```

In [72]:
```
df_INNER_JOIN = pd.merge(df1, d, left_on='bank_id', right_on='bank_id', how='inner')
```

In [73]:
```
df_INNER_JOIN.to_csv('./data/tmp777.csv', index=False, encoding = 'utf-8')
```

In [ ]:
```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js