M_calculation_&eda

November 12, 2020

EDA

[1]:
```
import matplotlib.pyplot as plt
```

[2]:
```
import os
import gc
import requests
from bs4 import BeautifulSoup
import pandas as pd
import zipfile
import re
import time
```

[3]:
```
from etl import *
```

[3]:
```
ld_to_csv('data/unzipped')
```

```
creat 5-million-row table 1
creat 5-million-row table 2
creat 5-million-row table 3
creat 5-million-row table 4
creat 5-million-row table 5
creat 5-million-row table 6
creat 5-million-row table 7
creat 5-million-row table 8
creat 5-million-row table 9
creat 5-million-row table 10
creat 5-million-row table 11
creat 5-million-row table 12
creat 5-million-row table 13
creat 5-million-row table 14
creat 5-million-row table 15
creat 5-million-row table 16
creat 5-million-row table 17
creat 5-million-row table 18
creat 5-million-row table 19
creat 5-million-row table 20
creat 5-million-row table 21
```

```
creat 5-million-row table 22
creat 5-million-row table 23
creat 5-million-row table 24
creat 5-million-row table 25
creat 5-million-row table 26
creat 5-million-row table 27
creat 5-million-row table 28
creat 5-million-row table 29
creat 5-million-row table 30
creat 5-million-row table 31
creat 5-million-row table 32
creat 5-million-row table 33
creat 5-million-row table 34
creat 5-million-row table 35
creat 5-million-row table 36
creat 5-million-row table 37
creat 5-million-row table 38
creat 5-million-row table 39
creat 5-million-row table 40
creat 5-million-row table 41
creat 5-million-row table 42
creat 5-million-row table 43
creat 5-million-row table 44
```

[4]:
```python
def add_dicts(dict1, dict2):
    result={}
    if len(dict1.keys())==0:
        return dict2
    elif len(dict2.keys())==0:
        return dict1
    else:
        for key in dict1.keys():
            if key in dict2.keys():
                result[key]=dict1[key]+dict2[key]
            else:
                result[key]=dict1[key]
        for key2 in dict2.keys():
            if key2 not in result.keys():
                result[key2]=dict2[key2]
        return result
```

[5]:
```python
def get_csvs_route(road):
    result=[]
    for filename in os.listdir(road):
        fileroute=os.path.join(road,filename)
        result.append(fileroute)
```

```
    return result
```

[6]:
```
def bots_raw_dict(road):
    total_freq={}
    revert_freq={}
    for csvname in os.listdir(road):
        csvroute=os.path.join(road, csvname)
        csvdict={}
        df=pd.read_csv(csvroute)
        df['user']=df['user'].str.lower()
        df['revert']=df['revert'].astype('int')
        dfrevert=df[df['revert']==1]
        bots=df[df['user'].str.contains('bot')]
        reverts=bots[bots['revert']==1]
        thef=bots['user'].value_counts().to_dict()
        ther=reverts['user'].value_counts().to_dict()
        total_freq=add_dicts(total_freq, thef)
        revert_freq=add_dicts(revert_freq, ther)
        print('finish dict csv', csvname, len(total_freq))
        del thef, ther
        del [[df, bots, reverts, dfrevert]]
        gc.collect()
        df=pd.DataFrame()
        bots=pd.DataFrame()
        reverts=pd.DataFrame()
        thef={}
        ther={}
        dfrevert=pd.DataFrame()
    return total_freq, revert_freq
```

[6]:
```
total_freq, revert_freq=bots_raw_dict('data/csvs/en_wiki')
```

```
finish dict csv en_wiki_13.csv 781
finish dict csv en_wiki_14.csv 1037
finish dict csv en_wiki_21.csv 1230
finish dict csv en_wiki_26.csv 1425
finish dict csv en_wiki_28.csv 1608
finish dict csv en_wiki_1.csv 1761
finish dict csv en_wiki_42.csv 1946
finish dict csv en_wiki_39.csv 2090
finish dict csv en_wiki_6.csv 2212
finish dict csv en_wiki_37.csv 2311
finish dict csv en_wiki_8.csv 2432
finish dict csv en_wiki_30.csv 2548
finish dict csv en_wiki_29.csv 2645
finish dict csv en_wiki_27.csv 2713
finish dict csv en_wiki_20.csv 2811
```

```
finish dict csv en_wiki_15.csv 2912
finish dict csv en_wiki_12.csv 2997
finish dict csv en_wiki_31.csv 3091
finish dict csv en_wiki_36.csv 3174
finish dict csv en_wiki_9.csv 3251
finish dict csv en_wiki_38.csv 3361
finish dict csv en_wiki_7.csv 3429
finish dict csv en_wiki_44.csv 3504
finish dict csv en_wiki_43.csv 3598
finish dict csv en_wiki_35.csv 3652
finish dict csv en_wiki_32.csv 3710
finish dict csv en_wiki_3.csv 3773
finish dict csv en_wiki_40.csv 3839
finish dict csv en_wiki_4.csv 3920
finish dict csv en_wiki_23.csv 3980
finish dict csv en_wiki_24.csv 4046
finish dict csv en_wiki_11.csv 4114
finish dict csv en_wiki_16.csv 4179
finish dict csv en_wiki_18.csv 4245
finish dict csv en_wiki_5.csv 4303
finish dict csv en_wiki_2.csv 4380
finish dict csv en_wiki_41.csv 4448
finish dict csv en_wiki_33.csv 4510
finish dict csv en_wiki_34.csv 4578
finish dict csv en_wiki_19.csv 4641
finish dict csv en_wiki_17.csv 4707
finish dict csv en_wiki_10.csv 4762
finish dict csv en_wiki_25.csv 4807
finish dict csv en_wiki_22.csv 4873
```

[9]:
```
total_f=pd.Series(total_freq)
```

[10]:
```
total_f.describe()
```

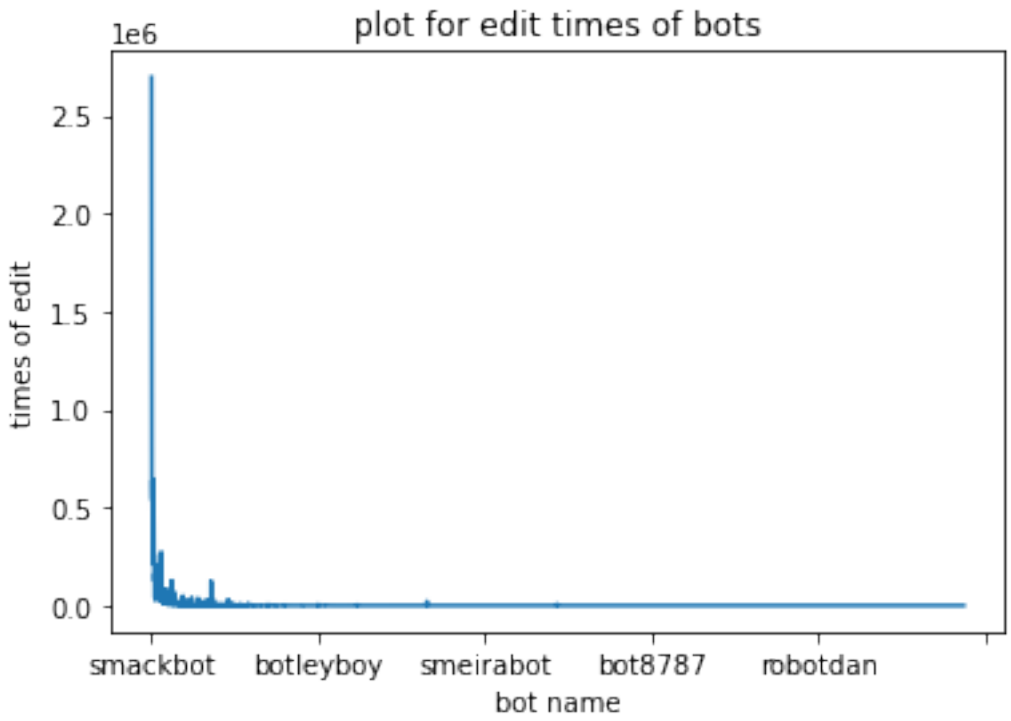
[10]:
```
count    4.873000e+03
mean     3.389500e+03
std      4.717530e+04
min      1.000000e+00
25%      1.000000e+00
50%      3.000000e+00
75%      1.100000e+01
max      2.695952e+06
dtype: float64
```

[11]:
```
total_f.sort_values(ascending=False).head(20)
```

[11]:
```
smackbot                  2695952
cydebot                    763366
cluebot                    644069
lightbot                   638751
russbot                    606590
the_anomebot2              536912
thijs!bot                  418576
alaibot                    379167
full-date_unlinking_bot    322808
bluebot                    320537
siebot                     291204
botijo                     289988
yurikbot                   272996
xqbot                      261936
cmdrobot                   234560
erik9bot                   233001
txikibot                   223028
flabot                     212084
dumzibot                   208023
volkovbot                  207434
dtype: int64
```

[42]:
```
#make plot of bots edit
total_f.plot()
plt.xlabel('bot name')
plt.ylabel('times of edit')
plt.title('plot for edit times of bots')
```

[42]:
```
Text(0.5, 1.0, 'plot for edit times of bots')
```



[12]:
```
total_r=pd.Series(revert_freq)
```

[14]:
```
total_r.describe()
```

[14]:
```
count      1237.000000
mean       1180.736459
std       19732.676887
min           1.000000
25%           1.000000
50%           2.000000
75%          14.000000
max      643853.000000
dtype: float64
```

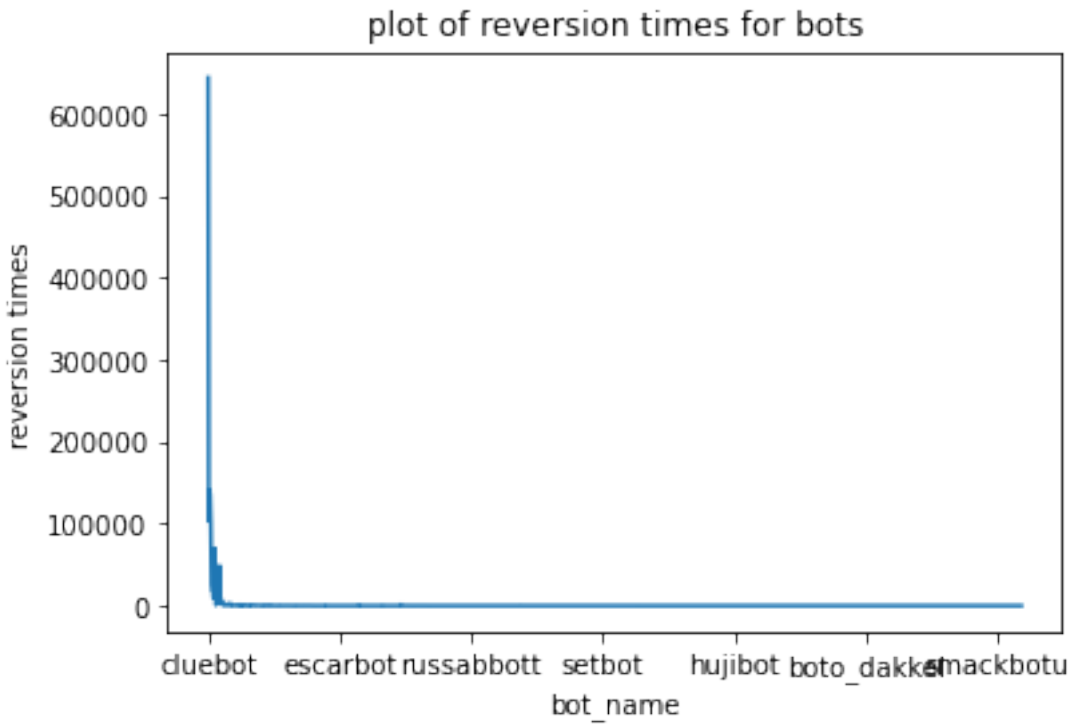
[13]:
```
total_r.sort_values(ascending=False).head(20)
```

[13]:
```
cluebot               643853
antivandalbot         141364
voabot_ii             134472
xlinkbot              103205
martinbot             100789
tawkerbot2             69941
pseudobot              48919
```

```
russbot                  25174
xqbot                    19643
antispambot              19338
darknessbot              13990
tawkerbot4               13533
soxbot_iii               12911
cydebot                   9175
dashbot                   8093
countervandalismbot       7379
  _robot                 5775
dumbbot                   4776
scepbot                   4301
robotman1974              2733
dtype: int64
```

[41]:
```
#make plot about revert time of bots
total_r.plot()
plt.xlabel('bot_name')
plt.ylabel('reversion times')
plt.title('plot of reversion times for bots')
```

[41]:
```
Text(0.5, 1.0, 'plot of reversion times for bots')
```



[23]:
```
combined_bots=pd.concat([total_f, total_r], axis=1)
combined_bots=combined_bots.fillna(0)
combined_bots.columns=['edit_frequency', 'revert_frequency']
combined_bots.head(10)
```

[23]:

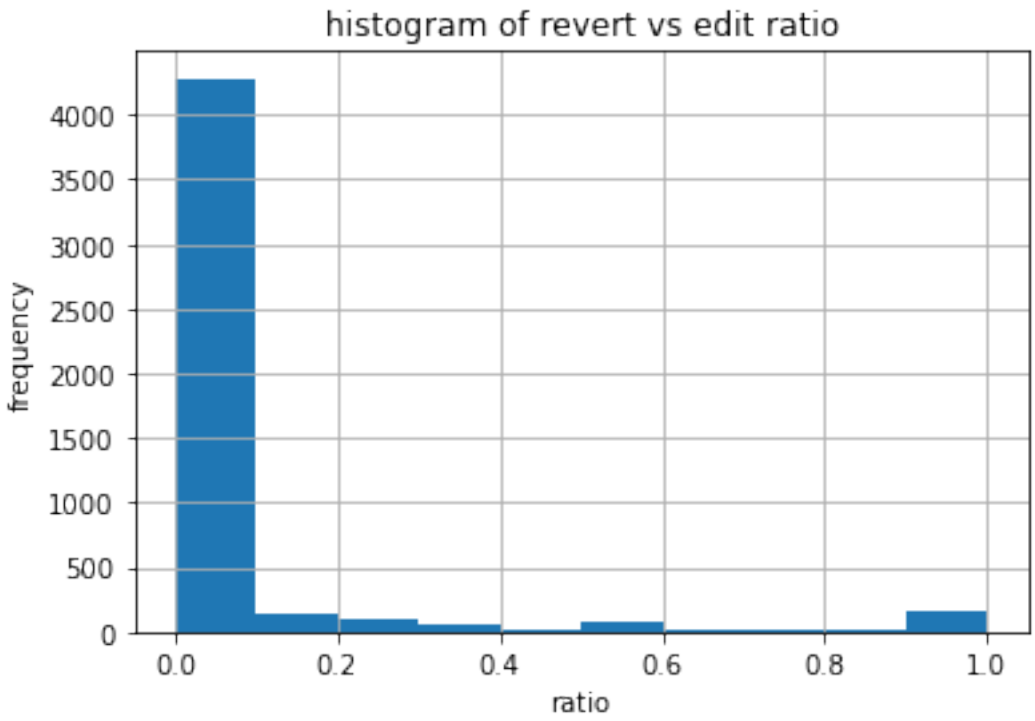
	edit_frequency	revert_frequency
smackbot	2695952	2727.0
cydebot	763366	9175.0
russbot	606590	25174.0
the_anomebot2	536912	437.0
lightbot	638751	398.0
thijs!bot	418576	1290.0
alaibot	379167	1098.0
full-date_unlinking_bot	322808	65.0
siebot	291204	1717.0
bluebot	320537	706.0

[25]:
```
#get revet/edit ratio
combined_bots['revert_vs_edit_ratio']=combined_bots['revert_frequency']/
 ↪combined_bots['edit_frequency']
combined_bots['revert_vs_edit_ratio'].describe()
```

[25]:
```
count    4873.000000
mean        0.063933
std         0.200478
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000332
max         1.000000
Name: revert_vs_edit_ratio, dtype: float64
```

[40]:
```
combined_bots['revert_vs_edit_ratio'].hist()
plt.xlabel('ratio')
plt.ylabel('frequency')
plt.title('histogram of revert vs edit ratio')
```

[40]:
```
Text(0.5, 1.0, 'histogram of revert vs edit ratio')
```



[7]:
```
def user_raw_dict(road):
    total_freq={}
    revert_freq={}
    for csvname in os.listdir(road):
        csvroute=os.path.join(road, csvname)
        csvdict={}
        df=pd.read_csv(csvroute)
        df['revert']=df['revert'].astype('int')
        reverts=df[df['revert']==1]
        thef=df['user'].value_counts().to_dict()
        ther=reverts['user'].value_counts().to_dict()
        total_freq=add_dicts(total_freq, thef)
        revert_freq=add_dicts(revert_freq, ther)
        print('finish dict csv', csvname, len(total_freq))
        del thef, ther
        del [[df, reverts]]
        gc.collect()
        df=pd.DataFrame()
        reverts=pd.DataFrame()
        thef={}
        ther={}
    return total_freq, revert_freq
```

[8]:
```
user_edit,user_revert=user_raw_dict('data/csvs/en_wiki')
```

```
finish dict csv en_wiki_13.csv 855530
finish dict csv en_wiki_14.csv 1577078
finish dict csv en_wiki_21.csv 2363298
finish dict csv en_wiki_26.csv 3124394
finish dict csv en_wiki_28.csv 3862676
finish dict csv en_wiki_1.csv 4255704
finish dict csv en_wiki_42.csv 5112855
finish dict csv en_wiki_39.csv 5830756
finish dict csv en_wiki_6.csv 6233712
finish dict csv en_wiki_37.csv 6933172
finish dict csv en_wiki_8.csv 7360302
finish dict csv en_wiki_30.csv 7957724
finish dict csv en_wiki_29.csv 8511945
finish dict csv en_wiki_27.csv 9042362
finish dict csv en_wiki_20.csv 9535546
finish dict csv en_wiki_15.csv 9988439
finish dict csv en_wiki_12.csv 10393546
finish dict csv en_wiki_31.csv 10920323
finish dict csv en_wiki_36.csv 11472583
finish dict csv en_wiki_9.csv 11849470
finish dict csv en_wiki_38.csv 12345611
finish dict csv en_wiki_7.csv 12657515
finish dict csv en_wiki_44.csv 13221377
finish dict csv en_wiki_43.csv 13790687
finish dict csv en_wiki_35.csv 14294245
finish dict csv en_wiki_32.csv 14747623
finish dict csv en_wiki_3.csv 15078318
finish dict csv en_wiki_40.csv 15599834
finish dict csv en_wiki_4.csv 15912687
finish dict csv en_wiki_23.csv 16318492
finish dict csv en_wiki_24.csv 16721855
finish dict csv en_wiki_11.csv 17051928
finish dict csv en_wiki_16.csv 17406399
finish dict csv en_wiki_18.csv 17756713
finish dict csv en_wiki_5.csv 17999204
finish dict csv en_wiki_2.csv 18256294
finish dict csv en_wiki_41.csv 18723635
finish dict csv en_wiki_33.csv 19130459
finish dict csv en_wiki_34.csv 19536597
finish dict csv en_wiki_19.csv 19884625
finish dict csv en_wiki_17.csv 20204136
finish dict csv en_wiki_10.csv 20494451
finish dict csv en_wiki_25.csv 20849974
finish dict csv en_wiki_22.csv 21191308
```

[9]:
```
user_e=pd.Series(user_edit)
user_r=pd.Series(user_revert)
```

[34]:
```
user_e.describe()
```

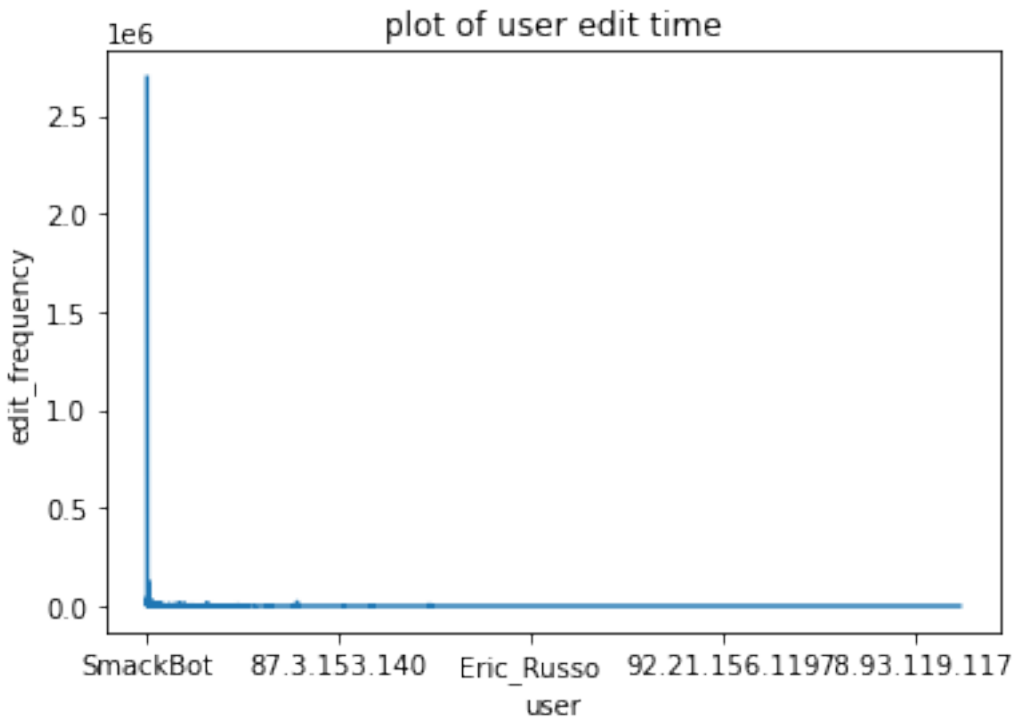
[34]:
```
count    2.119131e+07
mean     1.037460e+01
std      7.964880e+02
min      1.000000e+00
25%      1.000000e+00
50%      1.000000e+00
75%      3.000000e+00
max      2.695617e+06
dtype: float64
```

[35]:
```
user_r.describe()
```

[35]:
```
count    2.333106e+06
mean     9.548600e+00
std      5.206152e+02
min      1.000000e+00
25%      1.000000e+00
50%      1.000000e+00
75%      1.000000e+00
max      6.438530e+05
dtype: float64
```

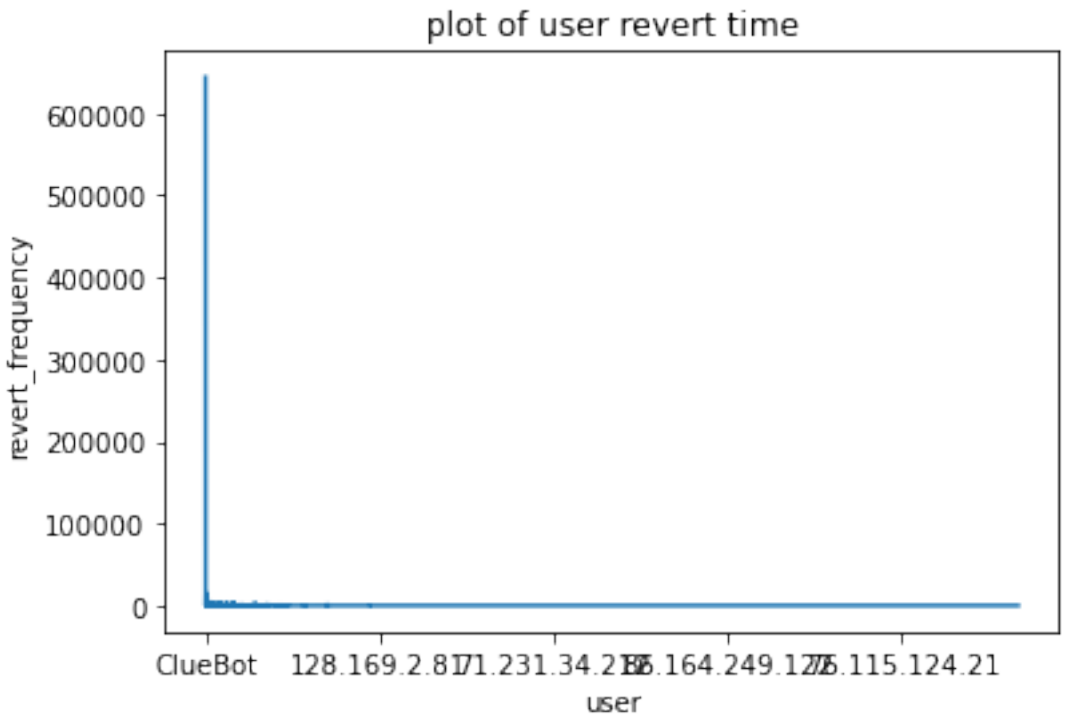
[39]:
```
user_e.plot()
plt.xlabel('user')
plt.ylabel("edit_frequency")
plt.title('plot of user edit time')
```

[39]:
```
Text(0.5, 1.0, 'plot of user edit time')
```



[43]:
```
user_r.plot()
plt.xlabel('user')
plt.ylabel('revert_frequency')
plt.title('plot of user revert time')
```

[43]: Text(0.5, 1.0, 'plot of user revert time')

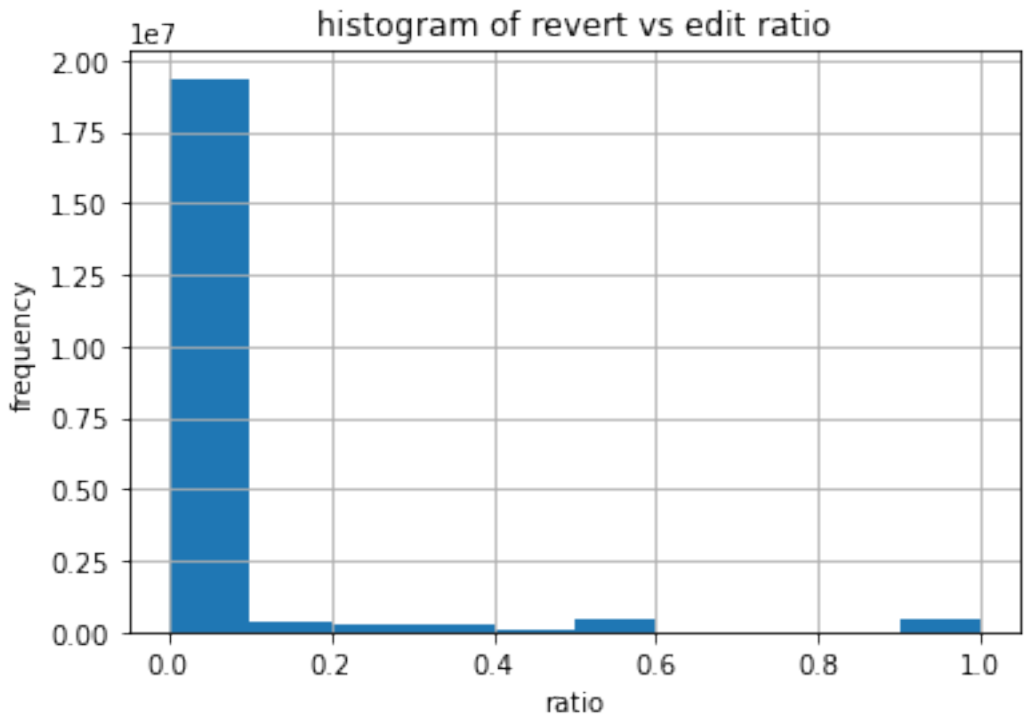


[10]:
```
combined_user=pd.concat([user_e,user_r], axis=1)
combined_user.columns=['user_edit', 'user_revert']
combined_user=combined_user.fillna(0)
combined_user['revert_vs_edit_ratio']=combined_user['user_revert']/
 ↪combined_user['user_edit']
combined_user['revert_vs_edit_ratio'].describe()
```

[10]:
```
count    2.119131e+07
mean     4.372522e-02
std      1.654792e-01
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      0.000000e+00
max      1.000000e+00
Name: revert_vs_edit_ratio, dtype: float64
```

[11]:
```
combined_user['revert_vs_edit_ratio'].hist()
plt.xlabel('ratio')
plt.ylabel('frequency')
plt.title('histogram of revert vs edit ratio')
```

[11]:
```
Text(0.5, 1.0, 'histogram of revert vs edit ratio')
```



[8]:
```
def article_raw_dict(road):
    total_freq={}
    revert_freq={}
    for csvname in os.listdir(road):
        csvroute=os.path.join(road, csvname)
        csvdict={}
        df=pd.read_csv(csvroute)
        df['revert']=df['revert'].astype('int')
        reverts=df[df['revert']==1]
        thef=df['article'].value_counts().to_dict()
        ther=reverts['article'].value_counts().to_dict()
        total_freq=add_dicts(total_freq, thef)
        revert_freq=add_dicts(revert_freq, ther)
        print('finish dict csv', csvname, len(total_freq))
        del thef, ther
        del [[df, reverts]]
        gc.collect()
        df=pd.DataFrame()
        reverts=pd.DataFrame()
        thef={}
        ther={}
    return total_freq, revert_freq
```

[8]:
```
article_edit, article_revert=article_raw_dict('data/csvs/en_wiki')
```

```
finish dict csv en_wiki_13.csv 143749
finish dict csv en_wiki_14.csv 280563
finish dict csv en_wiki_21.csv 369159
finish dict csv en_wiki_26.csv 436513
finish dict csv en_wiki_28.csv 489965
finish dict csv en_wiki_1.csv 853829
finish dict csv en_wiki_42.csv 860641
finish dict csv en_wiki_39.csv 880436
finish dict csv en_wiki_6.csv 1117734
finish dict csv en_wiki_37.csv 1136178
finish dict csv en_wiki_8.csv 1329572
finish dict csv en_wiki_30.csv 1376066
finish dict csv en_wiki_29.csv 1432868
finish dict csv en_wiki_27.csv 1493265
finish dict csv en_wiki_20.csv 1589220
finish dict csv en_wiki_15.csv 1708427
finish dict csv en_wiki_12.csv 1859087
finish dict csv en_wiki_31.csv 1906032
finish dict csv en_wiki_36.csv 1930126
finish dict csv en_wiki_9.csv 2104001
finish dict csv en_wiki_38.csv 2146542
finish dict csv en_wiki_7.csv 2380500
finish dict csv en_wiki_44.csv 2387132
finish dict csv en_wiki_43.csv 2393888
finish dict csv en_wiki_35.csv 2421556
finish dict csv en_wiki_32.csv 2460650
finish dict csv en_wiki_3.csv 2709865
finish dict csv en_wiki_40.csv 2722204
finish dict csv en_wiki_4.csv 2948020
finish dict csv en_wiki_23.csv 3024250
finish dict csv en_wiki_24.csv 3098564
finish dict csv en_wiki_11.csv 3252565
finish dict csv en_wiki_16.csv 3368597
finish dict csv en_wiki_18.csv 3471693
finish dict csv en_wiki_5.csv 3732896
finish dict csv en_wiki_2.csv 4032696
finish dict csv en_wiki_41.csv 4043017
finish dict csv en_wiki_33.csv 4077799
finish dict csv en_wiki_34.csv 4107741
finish dict csv en_wiki_19.csv 4209645
finish dict csv en_wiki_17.csv 4324303
finish dict csv en_wiki_10.csv 4489494
finish dict csv en_wiki_25.csv 4560631
finish dict csv en_wiki_22.csv 4644458
```

[9]:
```
article_e=pd.Series(article_edit)
article_r=pd.Series(article_revert)
combined_article=pd.concat([article_e,article_r], axis=1)
combined_article=combined_article.fillna(0)
```

[11]:
```
combined_article.columns=['article_edit', 'article_revert']
combined_article['revert_edit_ratio']=combined_article['article_revert']/
 ↪combined_article['article_edit']
```

[12]:
```
combined_article['article_edit'].describe()
```

[12]:
```
count    4.644458e+06
mean     4.733616e+01
std      2.175524e+02
min      2.000000e+00
25%      3.000000e+00
50%      1.100000e+01
75%      3.100000e+01
max      4.365000e+04
Name: article_edit, dtype: float64
```

[17]:
```
combined_article['article_revert'].describe()
```

[17]:
```
count    4.644458e+06
mean     4.796642e+00
std      4.395998e+01
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      1.000000e+00
max      1.508100e+04
Name: article_revert, dtype: float64
```

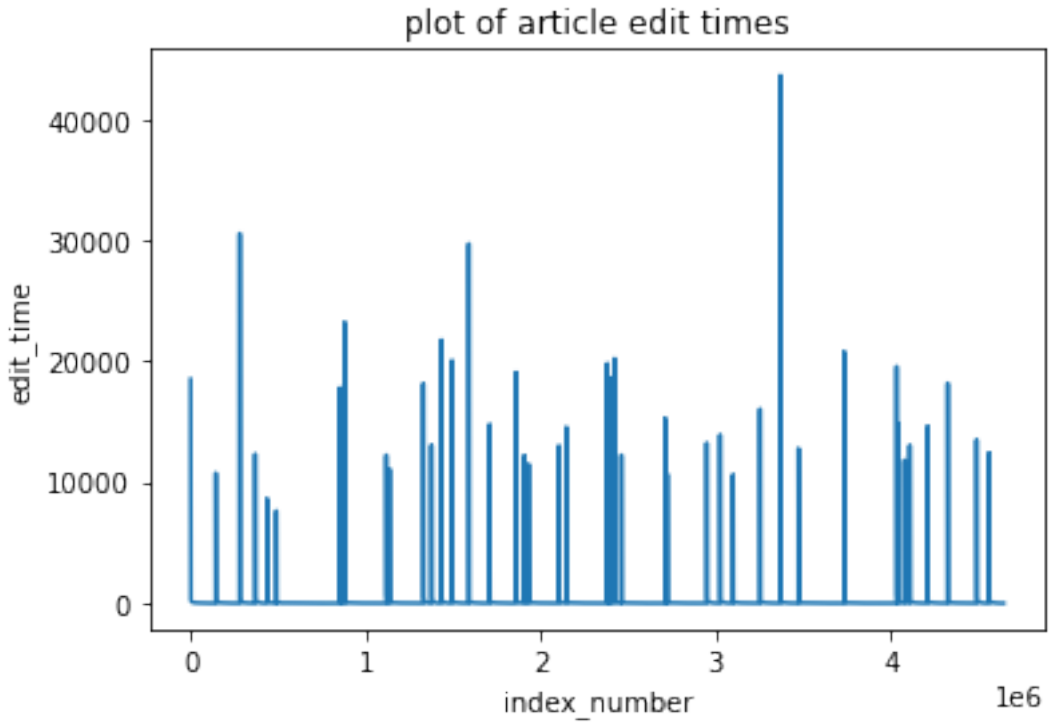
[18]:
```
combined_article['revert_edit_ratio'].describe()
```

[18]:
```
count    4.644458e+06
mean     4.328443e-02
std      7.863970e-02
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      6.091371e-02
max      9.411765e-01
Name: revert_edit_ratio, dtype: float64
```

[15]:
```
combined_article=combined_article.reset_index()
combined_article['article_edit'].plot()
```

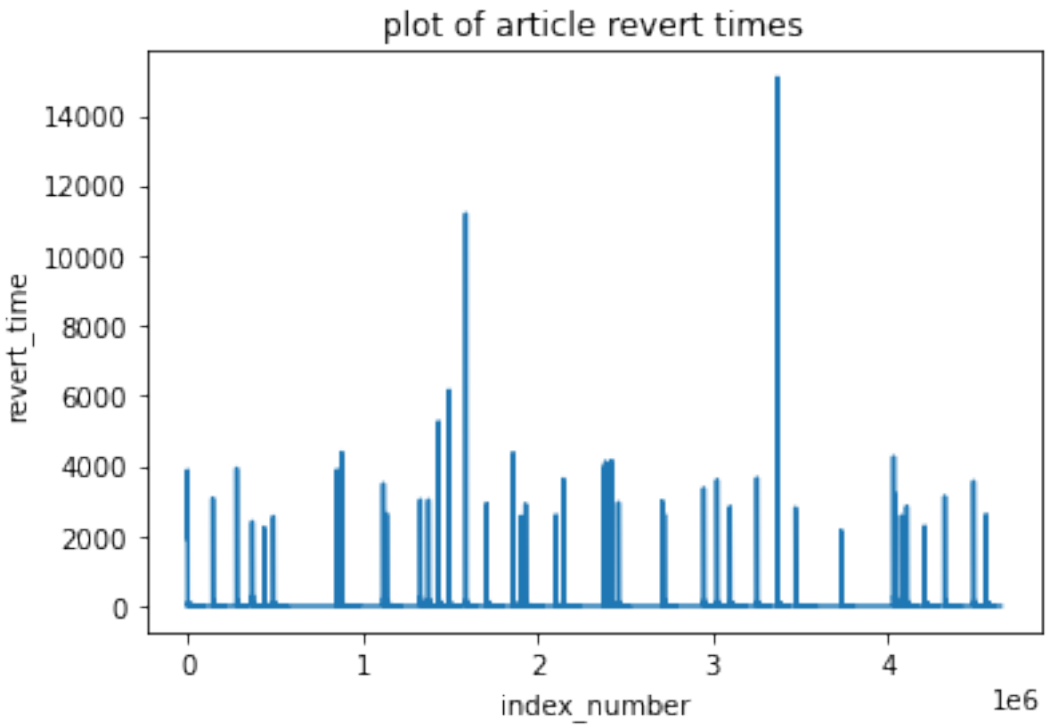
```
plt.xlabel('index_number')
plt.ylabel('edit_time')
plt.title('plot of article edit times')
```

[15]: Text(0.5, 1.0, 'plot of article edit times')



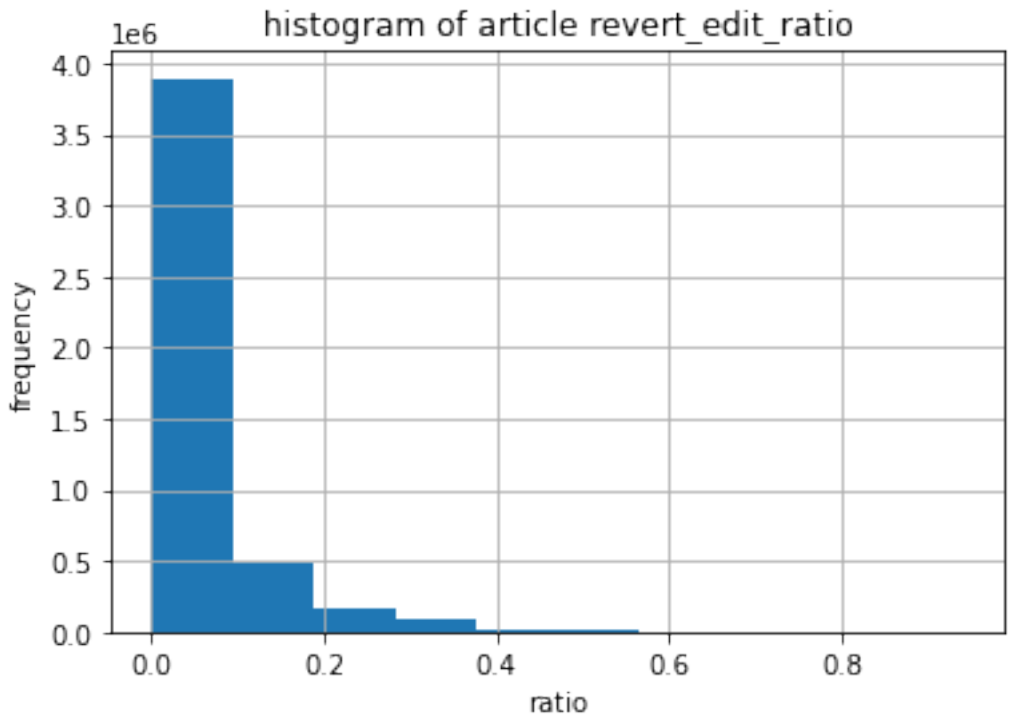
[16]:
```
combined_article['article_revert'].plot()
plt.xlabel('index_number')
plt.ylabel('revert_time')
plt.title('plot of article revert times')
```

[16]: Text(0.5, 1.0, 'plot of article revert times')



[20]:
```
combined_article['revert_edit_ratio'].hist()
plt.xlabel('ratio')
plt.ylabel('frequency')
plt.title('histogram of article revert_edit_ratio')
```

[20]: Text(0.5, 1.0, 'histogram of article revert_edit_ratio')



[9]:
```
def day_raw_dict(road):
    total_freq={}
    revert_freq={}
    for csvname in os.listdir(road):
        csvroute=os.path.join(road, csvname)
        csvdict={}
        df=pd.read_csv(csvroute)
        df['revert']=df['revert'].astype('int')
        df['time']=pd.to_datetime(df['time'])
        df['time']=df['time'].dt.floor('D')
        reverts=df[df['revert']==1]
        thef=df['time'].value_counts().to_dict()
        ther=reverts['time'].value_counts().to_dict()
        total_freq=add_dicts(total_freq, thef)
        revert_freq=add_dicts(revert_freq, ther)
        print('finish dict csv', csvname, len(total_freq))
        del thef, ther
        del [[df, reverts]]
        gc.collect()
        df=pd.DataFrame()
        reverts=pd.DataFrame()
        thef={}
        ther={}
```

```
    return total_freq, revert_freq
```

[10]:
```
day_edit, day_revert=day_raw_dict('data/csvs/en_wiki')
```

```
finish dict csv en_wiki_13.csv 2748
finish dict csv en_wiki_14.csv 2870
finish dict csv en_wiki_21.csv 2936
finish dict csv en_wiki_26.csv 2949
finish dict csv en_wiki_28.csv 2971
finish dict csv en_wiki_1.csv 3157
finish dict csv en_wiki_42.csv 3295
finish dict csv en_wiki_39.csv 3295
finish dict csv en_wiki_6.csv 3295
finish dict csv en_wiki_37.csv 3295
finish dict csv en_wiki_8.csv 3295
finish dict csv en_wiki_30.csv 3295
finish dict csv en_wiki_29.csv 3295
finish dict csv en_wiki_27.csv 3295
finish dict csv en_wiki_20.csv 3295
finish dict csv en_wiki_15.csv 3295
finish dict csv en_wiki_12.csv 3295
finish dict csv en_wiki_31.csv 3295
finish dict csv en_wiki_36.csv 3295
finish dict csv en_wiki_9.csv 3296
finish dict csv en_wiki_38.csv 3296
finish dict csv en_wiki_7.csv 3296
finish dict csv en_wiki_44.csv 3299
finish dict csv en_wiki_43.csv 3302
finish dict csv en_wiki_35.csv 3302
finish dict csv en_wiki_32.csv 3302
finish dict csv en_wiki_3.csv 3302
finish dict csv en_wiki_40.csv 3302
finish dict csv en_wiki_4.csv 3302
finish dict csv en_wiki_23.csv 3302
finish dict csv en_wiki_24.csv 3302
finish dict csv en_wiki_11.csv 3302
finish dict csv en_wiki_16.csv 3302
finish dict csv en_wiki_18.csv 3302
finish dict csv en_wiki_5.csv 3302
finish dict csv en_wiki_2.csv 3302
finish dict csv en_wiki_41.csv 3302
finish dict csv en_wiki_33.csv 3302
finish dict csv en_wiki_34.csv 3302
finish dict csv en_wiki_19.csv 3303
finish dict csv en_wiki_17.csv 3303
finish dict csv en_wiki_10.csv 3303
finish dict csv en_wiki_25.csv 3303
```

```
finish dict csv en_wiki_22.csv 3303
```

[12]:
```
day_e=pd.Series(day_edit)
day_r=pd.Series(day_revert)
combined_day=pd.concat([day_e, day_r], axis=1)
combined_day=combined_day.fillna(0)
combined_day.columns=['daily_edit', 'daily_revert']
combined_day['daily_revert_edit_ratio']=combined_day['daily_revert']/
 ↪combined_day['daily_edit']
combined_day=combined_day.sort_index()
combined_day.head(10)
```

[12]:

	daily_edit	daily_revert	daily_revert_edit_ratio
2001-01-16 00:00:00+00:00	1	0.0	0.0
2001-01-17 00:00:00+00:00	5	0.0	0.0
2001-01-18 00:00:00+00:00	2	0.0	0.0
2001-01-19 00:00:00+00:00	7	0.0	0.0
2001-01-20 00:00:00+00:00	6	0.0	0.0
2001-01-21 00:00:00+00:00	18	0.0	0.0
2001-01-22 00:00:00+00:00	5	0.0	0.0
2001-01-23 00:00:00+00:00	10	0.0	0.0
2001-01-24 00:00:00+00:00	4	0.0	0.0
2001-01-25 00:00:00+00:00	10	0.0	0.0

[15]:
```
combined_day['daily_edit'].describe()
```

[15]:
```
count      3303.000000
mean      66561.087496
std       62589.939336
min           1.000000
25%        2835.500000
50%       44372.000000
75%      132099.000000
max      218485.000000
Name: daily_edit, dtype: float64
```

[16]:
```
combined_day['daily_revert'].describe()
```

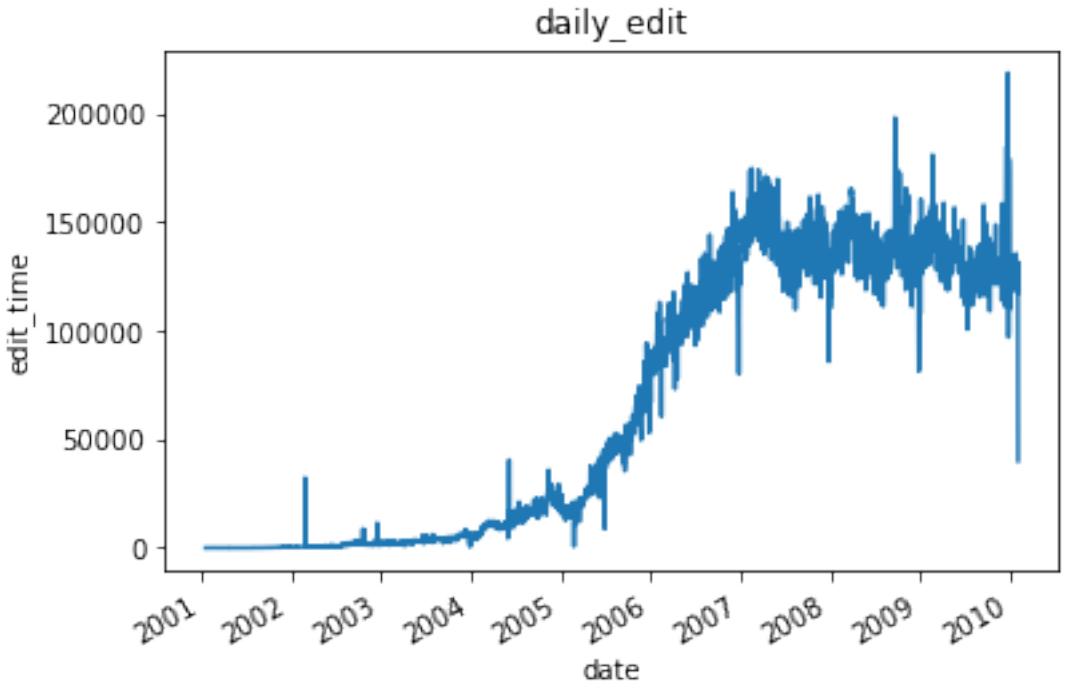
[16]:
```
count     3303.000000
mean      6744.745686
std       7193.217115
min          0.000000
25%         68.000000
50%       2925.000000
75%      13202.500000
max      22191.000000
Name: daily_revert, dtype: float64
```

[17]:
```
combined_day['daily_revert_edit_ratio'].describe()
```

[17]:
```
count    3303.000000
mean        0.065263
std         0.042087
min         0.000000
25%         0.027679
50%         0.066481
75%         0.101164
max         0.197917
Name: daily_revert_edit_ratio, dtype: float64
```

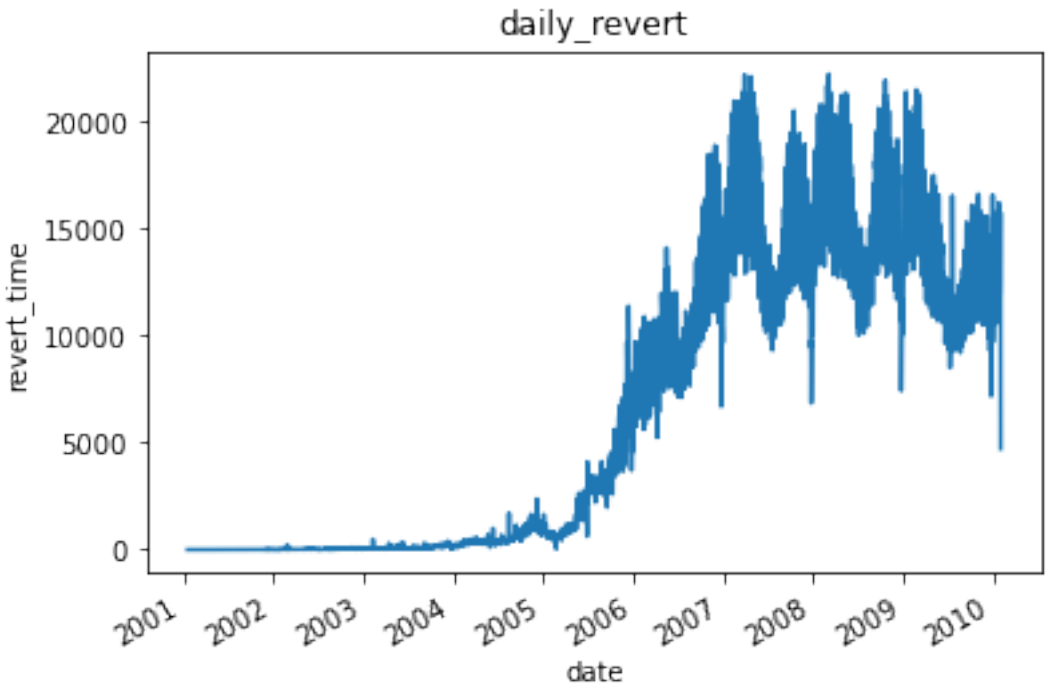
[13]:
```
combined_day['daily_edit'].plot()
plt.xlabel('date')
plt.ylabel('edit_time')
plt.title('daily_edit')
```

[13]:
```
Text(0.5, 1.0, 'daily_edit')
```



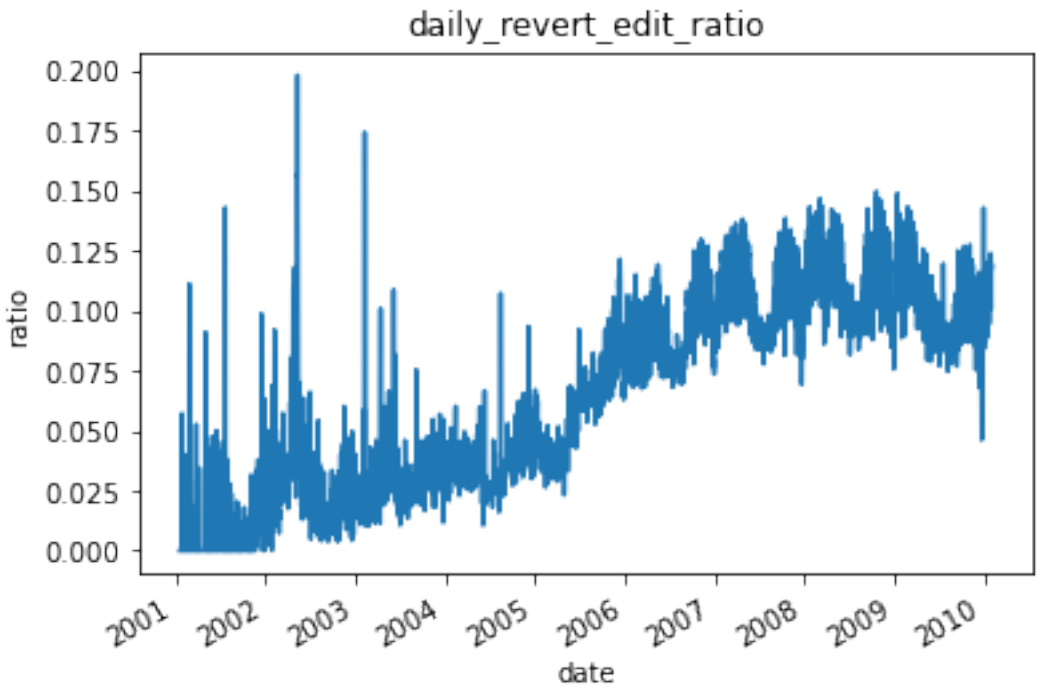
[18]:
```
combined_day['daily_revert'].plot()
plt.xlabel('date')
plt.ylabel('revert_time')
plt.title('daily_revert')
```

[18]: Text(0.5, 1.0, 'daily_revert')



[19]:
```
combined_day['daily_revert_edit_ratio'].plot()
plt.xlabel('date')
plt.ylabel('ratio')
plt.title('daily_revert_edit_ratio')
```

[19]: Text(0.5, 1.0, 'daily_revert_edit_ratio')



[ ]:

[ ]:

M-Stats

[6]:
```
def calculateM(df):
    # find revert pairs
    revert_pairs = []
    revert_users = []
    ones=df[df['revert']==1]
    twodf=df[df['revert']==0]
    for oness in ones['user'].unique():
        masker=ones[ones['user']==oness]
        for indi in masker['version'].values:
            one = oness
            the_version=indi+1
            alti=the_version-1
            try:
                twocolumn=twodf[twodf['version']==the_version]
                two=twocolumn['user'].values[0]
            except Exception as e:
                twocolumn=twodf[twodf['version']==alti]
                two=twocolumn['user'].values[0]
            revert_pairs.append((one, two))
```

```
            if one not in revert_users:
                revert_users.append(one)
            if two not in revert_users:
                revert_users.append(two)
    #find mutual reverts
    mutual_rev_users = []
    for pair in revert_pairs:
        one = pair[0]
        two = pair[1]

        #mutual revert found
        if (two, one) in revert_pairs:
            mutual_rev_users.append(two)
            mutual_rev_users.append(one)

    #remove duplicates, calculate num
    E = len(list(set(mutual_rev_users)))
    #calculate M
    M = 0
    num_edits = df['user'].value_counts()
    revert_pairs = list(set(revert_pairs))
    for pair in revert_pairs:
        one = pair[0]
        two = pair[1]
        if num_edits[one] < num_edits[two]:
            N = num_edits[one]
        else:
            N = num_edits[two]

        M += N

    M *= E
    return M
```

[7]:
```
def better_calculate_M(road):
    Ms={}
    for csvname in os.listdir(road):
        st=time.time()
        csvroute=os.path.join(road, csvname)
        csvdict={}
        df=pd.read_csv(csvroute)
        df['revert']=df['revert'].astype('int')
        df['version']=df['version'].astype('int')
        first_clear=df.groupby('article')['revert'].sum()
        fs=first_clear[first_clear<2].to_dict()
        for k in fs:
            fs[k]=0
```

```
        Ms.update(fs)
        need_calculate=first_clear[first_clear>1].index
        df=df[df['user'].isin(need_calculate)]
        count=0
        stt=time.time()
        for i in need_calculate:
            partdf=df.loc[df['article']==i, :]
            M=calculateM(partdf)
            Ms[i]=M
            count +=1
            if count%1000==0:
                thet=time.time()-stt
                print('the time needed for ', count, 'articles is', thet,␣
↪'seconds')
            del [[partdf]]
            gc.collect()
            partdf=pd.DataFrame()
        totalt=time.time()-st
        print('complete M calculation on', csvname, 'at the time used in␣
↪seconds', totalt)
        del [[df, first_clear, need_calculate]]
        gc.collect()
        df=pd.DataFrame()
        first_clear=pd.Series()
    return Ms
```

[3]:
```
import warnings
warnings.filterwarnings('error')
```

[16]:
```
Ms=better_calculate_M('data/csvs/en_wiki')
```

```
the time needed for  1000 articles is 42.061967611312866 seconds
```

```
        IndexErrorTraceback (most recent call last)

        <ipython-input-12-56cd1ace8728> in calculateM(df)
         14                 twocolumn=twodf[twodf['version']==the_version]
    ---> 15                 two=twocolumn['user'].values[0]
         16             except Exception as e:


        IndexError: index 0 is out of bounds for axis 0 with size 0
```

```
During handling of the above exception, another exception occurred:


    IndexErrorTraceback (most recent call last)

    <ipython-input-16-91c6631152ab> in <module>
----> 1 Ms=better_calculate_M('data/csvs/en_wiki')


    <ipython-input-14-b8eca8ab2a3c> in better_calculate_M(road)
     19         for i in need_calculate:
     20             partdf=df.loc[df['article']==i, :]
---> 21             M=calculateM(partdf)
     22             Ms[i]=M
     23             count +=1


    <ipython-input-12-56cd1ace8728> in calculateM(df)
     16             except Exception as e:
     17                 twocolumn=twodf[twodf['version']==alti]
---> 18                 two=twocolumn['user'].values[0]
     19             revert_pairs.append((one, two))
     20             if one not in revert_users:


    IndexError: index 0 is out of bounds for axis 0 with size 0
```

[2]:
```
from sqlalchemy import create_engine
import numpy as np
```

[3]:
```
df=pd.read_csv('data/csvs/en_wiki/en_wiki_1.csv')
df['revert']=df['revert'].astype('int')
df['version']=df['version'].astype('int')
first_clear=df.groupby('article')['revert'].sum()
need_calculate=first_clear[first_clear>1].index
df=df[df['article'].isin(need_calculate)]
```

[13]:
```
thefreq=df.groupby('article')['user'].value_counts().to_dict()
```

[14]:
```
def finder(x,y,dict1):
    return dict1[(x,y)]
```

[17]:
```
df['edit']=df.apply(lambda x: finder(x['article'], x['user'], thefreq), axis=1)
```

[20]:
```
engine = create_engine('sqlite://', echo=False)
df.to_sql('articles', con=engine)
```

[21]:
```
engine.execute("CREATE TABLE reverted AS SELECT edit,user,version, article FROM articles WHERE revert=1")
```

[21]:
```
<sqlalchemy.engine.result.ResultProxy at 0x7f9a57dcc910>
```

[22]:
```
engine.execute("CREATE TABLE nonreverted AS SELECT edit,user,version, article FROM articles WHERE revert=0 ")
```

[22]:
```
<sqlalchemy.engine.result.ResultProxy at 0x7f9a34cb8c10>
```

[23]:
```
engine.execute("CREATE TABLE merged AS SELECT r.edit AS revertedit, n.edit AS nedit, r.user AS revertor, n.user AS revertee, r.article \
                FROM reverted AS r, nonreverted AS n WHERE r.version=n.version+1 AND r.article=n.article")
```

[23]:
```
<sqlalchemy.engine.result.ResultProxy at 0x7f9a34c9c4d0>
```

[26]:
```
sigma=engine.execute("SELECT SUM(MIN(revertedit, nedit)), article FROM merged GROUP BY article").fetchall()
```

[ ]:
```
engine.execute("SELECT COUNT(m1.revertor), m1.article, m1.revertor, m1.revertee FROM merged AS m1 WHERE EXISTS \
                (SELECT m2.revertor, m2.revertee FROM merged AS m2 WHERE m1.revertor=m2.revertee AND m1.revertee=m2.revertor AND m2.article=m1.article)\
                GROUP BY article").fetchall()[:10]
```

[ ]:
```

```