Quora Data Challenge

Increasing user engagement on Quora app

Suppose you are a Data Scientist on the Mobile team at Quora. The team has just introduced a new UI design to the Quora app. The goal of the new design is to increase user engagement (measured by minutes spent on site). The team ran an A/B test to evaluate the change. Using the data, help the team understand the impact of the UI change better.

Tables provided are as follows:

1. t1_user_active_min.csv
2. t2_user_variant.csv
3. t3_user_active_min_pre.csv
4. t4_user_attributes.csv

In [73]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as st
from scipy.stats import norm
from scipy.stats import zscore
from scipy import stats
from scipy.stats import ttest_ind
import random
import math

import warnings
%matplotlib inline
```

In [74]:

```
t1 = pd.read_csv("/Users/hojin/Desktop/quora-data-challenge/dataset/t1_user_active_min.csv")
t2 = pd.read_csv("/Users/hojin/Desktop/quora-data-challenge/dataset/t2_user_variant.csv")
t3 = pd.read_csv("/Users/hojin/Desktop/quora-data-challenge/dataset/t3_user_active_min_pre.csv")
t4 = pd.read_csv("/Users/hojin/Desktop/quora-data-challenge/dataset/t4_user_attributes.csv")
```

In [79]:

```
print("t1_user_active_min")
display(t1.head(3))
print("t2_user_variant")
display(t2.head(3))
print("t3_user_active_min")
display(t3.head(3))
print("t4_user_attributes")
display(t4.head(3))
```

t1_user_active_min

	uid	dt	active_mins
0	0	2019-02-22	5.0
1	0	2019-03-11	5.0
2	0	2019-03-18	3.0

t2_user_variant

	uid	variant_number	dt	signup_date
0	0	0	2019-02-06	2018-09-24
1	1	0	2019-02-06	2016-11-07
2	2	0	2019-02-06	2018-09-17

t3_user_active_min

	uid	dt	active_mins
0	0	2018-09-24	3.0
1	0	2018-11-08	4.0
2	0	2018-11-24	3.0

t4_user_attributes

	uid	gender	user_type
0	0	male	non_reader
1	1	male	reader
2	2	male	non_reader

In [4]:

```
display(t1.shape)
print("min:",min(t1.active_mins))
print("max:",max(t1.active_mins))
print("\n")
t1.info()
```

```
(1066402, 3)

min: 1.0
max: 99999.0


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1066402 entries, 0 to 1066401
Data columns (total 3 columns):
 #   Column       Non-Null Count    Dtype  
---  ------       --------------    -----  
 0   uid          1066402 non-null  int64  
 1   dt           1066402 non-null  object 
 2   active_mins  1066402 non-null  float64
dtypes: float64(1), int64(1), object(1)
memory usage: 24.4+ MB
```

In [5]:

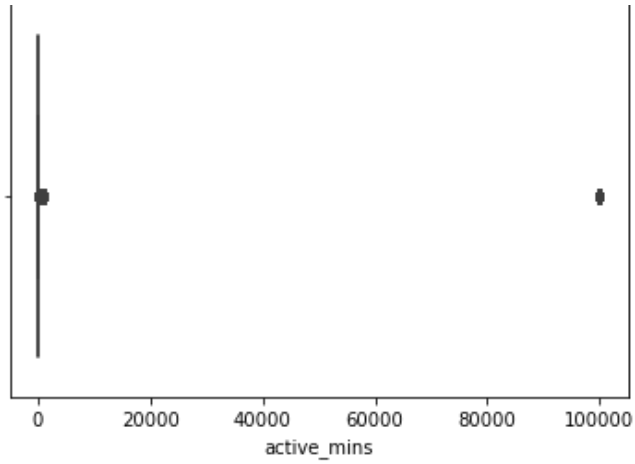
```
t1.isnull().sum()
```

Out[5]:

```
uid            0
dt             0
active_mins    0
dtype: int64
```

In [6]:

```
# checking outliers of t1
sns.boxplot(x=t1['active_mins'])
plt.show()
```



In [7]:

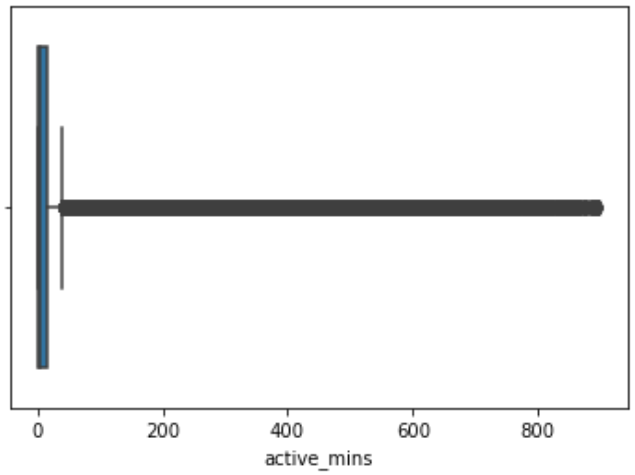
```
#removing outliers
z_scores = stats.zscore(t1['active_mins'])
abs_z_scores = np.abs(z_scores)
filtered_entries = (abs_z_scores < 3)
new_t1 = t1[filtered_entries]
```

In [8]:

```
sns.boxplot(x=new_t1['active_mins'])
display(new_t1.shape)
print("min:",min(new_t1.active_mins))
print("max:",max(new_t1.active_mins))
```

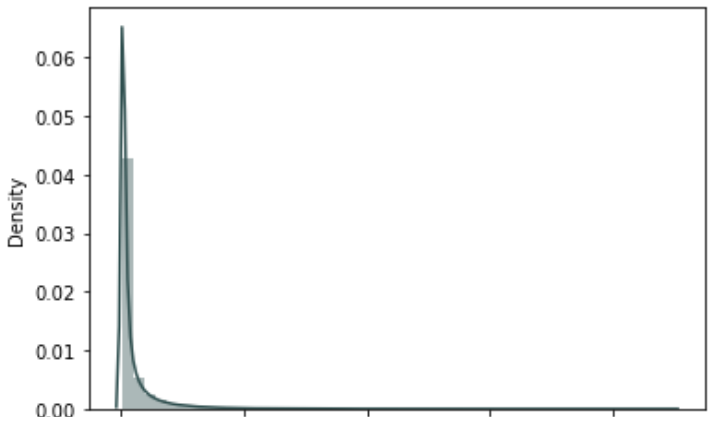
```
(1066230, 3)

min: 1.0
max: 897.0
```



In [9]:

```
warnings.filterwarnings('ignore')
sns.distplot(new_t1['active_mins'], color="darkslategrey")
plt.show()
```



In [10]:

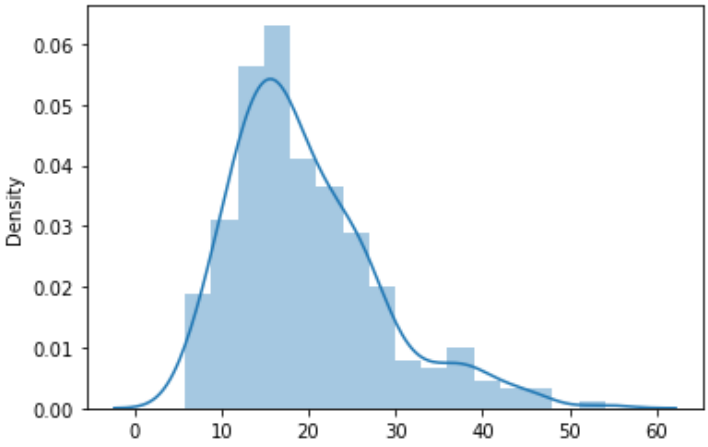
```
t1_mean = round(new_t1['active_mins'].mean(),3)
t1_std = round(new_t1['active_mins'].std(),3)
print("std:", t1_std)
print("mean:", t1_mean)
```

```
std: 46.538
mean: 20.042
```

In [11]:

```
sample_means = []
n = 30
for sample in range(0,300):
    sample_values = np.random.choice(a=new_t1['active_mins'], size=n)
    sample_mean = np.mean(sample_values)
    sample_means.append(sample_mean)

sns.distplot(sample_means)
plt.show()
```



In [12]:

```
min_date = min(new_t1['dt'])
max_date = max(new_t1['dt'])
print(min_date)
print(max_date)
```

```
2019-02-06
2019-07-05
```

In [13]:

```
print(t2.shape)
t2.head()
```

```
(50000, 4)
```

Out[13]:

	uid	variant_number	dt	signup_date
0	0	0	2019-02-06	2018-09-24
1	1	0	2019-02-06	2016-11-07
2	2	0	2019-02-06	2018-09-17
3	3	0	2019-02-06	2018-03-04
4	4	0	2019-02-06	2017-03-09

In [14]:

```
t2.isnull().sum()
```

Out[14]:

```
uid               0
variant_number    0
dt                0
signup_date       0
dtype: int64
```

In [15]:

```
# checking number of unique users
print("t1:",t1.uid.nunique())
print("t2:",t2.uid.nunique())
print("t3:",t3.uid.nunique())
```

```
t1: 46633
t2: 50000
t3: 49697
```

In [16]:

```
df = pd.merge(left=new_t1, right=t2, left_on ="uid", right_on="uid")
```

In [17]:

```
df.shape
```

Out[17]:

```
(1066230, 6)
```

In [18]:

```
df.head(15)
```

Out[18]:

	uid	dt_x	active_mins	variant_number	dt_y	signup_date
0	0	2019-02-22	5.0	0	2019-02-06	2018-09-24
1	0	2019-03-11	5.0	0	2019-02-06	2018-09-24
2	0	2019-03-18	3.0	0	2019-02-06	2018-09-24
3	0	2019-03-22	4.0	0	2019-02-06	2018-09-24
4	0	2019-04-03	9.0	0	2019-02-06	2018-09-24
5	0	2019-04-06	1.0	0	2019-02-06	2018-09-24
6	0	2019-04-17	1.0	0	2019-02-06	2018-09-24
7	0	2019-05-07	3.0	0	2019-02-06	2018-09-24
8	0	2019-05-14	1.0	0	2019-02-06	2018-09-24
9	0	2019-05-19	1.0	0	2019-02-06	2018-09-24
10	0	2019-05-22	3.0	0	2019-02-06	2018-09-24
11	0	2019-06-14	5.0	0	2019-02-06	2018-09-24
12	0	2019-06-16	2.0	0	2019-02-06	2018-09-24
13	1	2019-02-07	79.0	0	2019-02-06	2016-11-07
14	1	2019-02-09	211.0	0	2019-02-06	2016-11-07

In [95]:

```
from datetime import datetime
t1['dt'] = pd.to_datetime(t1['dt'])
t3['dt'] = pd.to_datetime(t3['dt'])
df['signup_date'] = pd.to_datetime(df['signup_date'])
```

In [96]:

```
df.dtypes
```

Out[96]:

```
uid                          int64
dt_x                datetime64[ns]
active_mins                float64
variant_number               int64
dt_y                datetime64[ns]
signup_date         datetime64[ns]
dtype: object
```

In [97]:

```
min_date_t1 = min(t1['dt'])
max_date_t1 = max(t1['dt'])

min_date_t3 = min(t3['dt'])
max_date_t3 = max(t3['dt'])

min_signup_date = min(df['signup_date'])
max_signup_date = max(df['signup_date'])


print('min date_t1:', min_date_t1)
print('max date_t1:', max_date_t1)
print('total date:',max_date_t1 - min_date_t1)

print('\nmin date_t3:', min_date_t3)
print('max date_t3:', max_date_t3)
print('total date:',max_date_t3 - min_date_t3)

print('\nmin signup date:', min_signup_date)
print('max signup date:', max_signup_date)
print('total date:',max_signup_date - min_signup_date)
```

```
min date_t1: 2019-02-06 00:00:00
max date_t1: 2019-07-05 00:00:00
total date: 149 days 00:00:00

min date_t3: 2018-08-10 00:00:00
max date_t3: 2019-02-05 00:00:00
total date: 179 days 00:00:00

min signup date: 1970-01-01 00:00:00
max signup date: 2019-02-04 00:00:00
total date: 17931 days 00:00:00
```

It's been about 150 days for users to use a new UI deisgn on Quora mobile app.

In [24]:

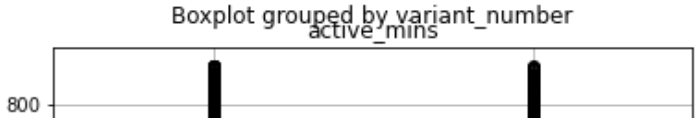
```
df.variant_number.value_counts()
```

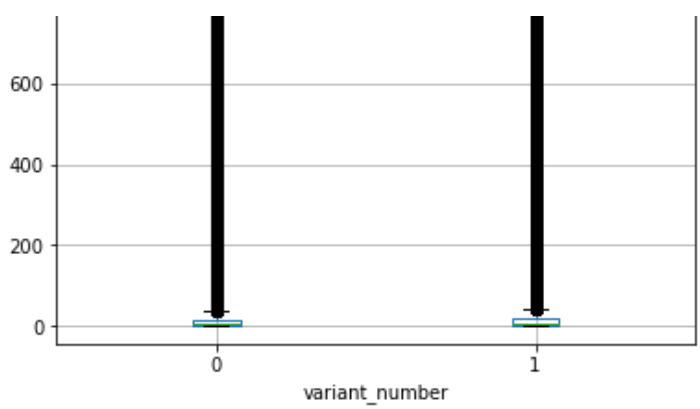
Out[24]:

```
0    886815
1    179415
Name: variant_number, dtype: int64
```

In [25]:

```
df.boxplot(column="active_mins", by="variant_number")
plt.show()
```





In [26]:

```
# var0 = df[df['variant_number']==0]
# var1 = df[df['variant_number']==1]
# ttest = df[['active_mins', 'variant_number']]
# pvalue = [(x, ttest_ind(var0[x].dropna(), var1[x].dropna()).pvalue) for x in ttest.colu
mns]
# p = [item for item in pvalue if item[1] < 0.05]
# sel_feature = [item[0] for item in p]
# sel_feature
```

In [27]:

```
stats = df.groupby('variant_number')['active_mins'].agg(['mean', 'std', 'var', 'count'])
.reset_index()
stats
```

Out[27]:

	variant_number	mean	std	var	count
0	0	19.337660	44.797631	2006.827734	886815
1	1	23.526294	54.191356	2936.703110	179415

In [28]:

```
print('mean of variant_num 0:', round(stats.loc[0]['mean'],3))
print('mean of variant_num 1:', round(stats.loc[1]['mean'],3))
```

```
mean of variant_num 0: 19.338
mean of variant_num 1: 23.526
```

In [29]:

```
dif = math.sqrt((stats.loc[0]['var']/stats.loc[0]['count'])+((stats.loc[1]['var']/stats.
loc[1]['count'])))
upper = (stats.loc[1]['mean']-stats.loc[0]['mean']) + (1.96 * dif)
lower = (stats.loc[1]['mean']-stats.loc[0]['mean']) - (1.96 * dif)
print([round(lower,2),round(upper,2)])
```

```
[3.92, 4.46]
```

In [30]:

```
df.head(1)
```

Out[30]:

	uid	dt_x	active_mins	variant_number	dt_y	signup_date
0	0	2019-02-22	5.0	0	2019-02-06	2018-09-24

In [31]:

```
# df.rename(columns={'active_mins':'experiment_active_mins'})
```

In [32]:

```
# gathering additional data with t3 table
t3.head()
```

Out[32]:

	uid	dt	active_mins
0	0	2018-09-24	3.0
1	0	2018-11-08	4.0
2	0	2018-11-24	3.0
3	0	2018-11-28	6.0
4	0	2018-12-02	6.0

In [33]:

```
t3['active_mins'].describe()
```

Out[33]:

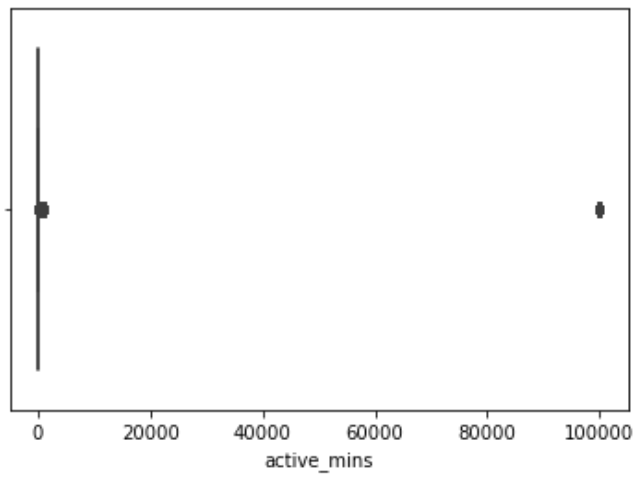
```
count    1.190093e+06
mean     3.220315e+01
std      1.181531e+03
min      1.000000e+00
25%      2.000000e+00
50%      4.000000e+00
75%      1.400000e+01
max      9.999900e+04
Name: active_mins, dtype: float64
```

In [34]:

```
#checking outliers on t3
sns.boxplot(x=t3['active_mins'])
```

Out[34]:

```
<AxesSubplot:xlabel='active_mins'>
```

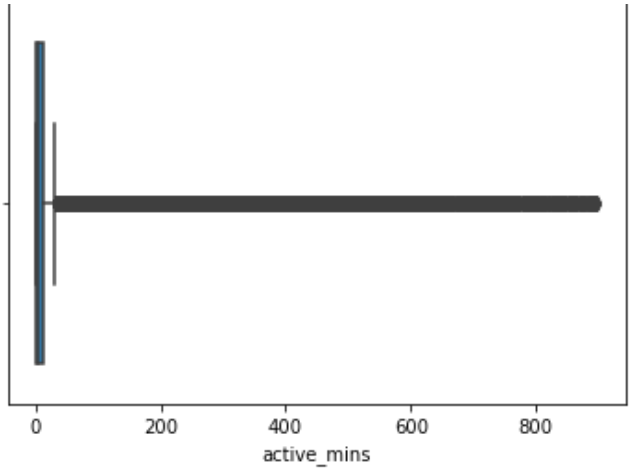


In [38]:

```
# removing outliers
z_scores = stats.zscore(t3['active_mins'])
abs_z_scores = np.abs(z_scores)
filtered_entries = (abs_z_scores < 3)
new_t3 = t3[filtered_entries]
```

In [39]:

```
sns.boxplot(x=new_t3['active_mins'])
plt.show()
```



In [40]:

```
df2 = pd.merge(left=new_t3, right=t2, left_on='uid', right_on='uid')
```

In [41]:

```
df2.head()
```

Out[41]:

	uid	dt_x	active_mins	variant_number	dt_y	signup_date
0	0	2018-09-24	3.0	0	2019-02-06	2018-09-24
1	0	2018-11-08	4.0	0	2019-02-06	2018-09-24
2	0	2018-11-24	3.0	0	2019-02-06	2018-09-24
3	0	2018-11-28	6.0	0	2019-02-06	2018-09-24
4	0	2018-12-02	6.0	0	2019-02-06	2018-09-24

In [42]:

```
stats2 = df2.groupby('variant_number')['active_mins'].agg(['mean', 'std', 'var', 'count'
]).reset_index()
stats2
```

Out[42]:

	variant_number	mean	std	var	count
0	0	19.204351	45.459884	2066.601053	989328
1	1	13.586847	32.087608	1029.614606	200599

In [43]:

```
dif2 = math.sqrt((stats2.loc[0]['var']/stats2.loc[0]['count'])+((stats2.loc[1]['var']/st
ats2.loc[1]['count'])))
upper2 = (stats2.loc[1]['mean']-stats2.loc[0]['mean']) + (1.96 * dif2)
lower2 = (stats2.loc[1]['mean']-stats2.loc[0]['mean']) - (1.96 * dif2)
print([round(lower2,2),round(upper2,2)])
```

```
[-5.78, -5.45]
```

In [44]:

```
# deepr dive with t4 table
t4.head()
```

Out[44]:

	uid	gender	user_type
0	0	male	non_reader

	uid	gender	user_type
2	2	male	non_reader
3	3	male	non_reader
4	4	male	non_reader

In [45]:

```
df3 = pd.merge(left=df2, right=t4, left_on='uid', right_on='uid')
df3.head()
```

Out[45]:

	uid	dt_x	active_mins	variant_number	dt_y	signup_date	gender	user_type
0	0	2018-09-24	3.0	0	2019-02-06	2018-09-24	male	non_reader
1	0	2018-11-08	4.0	0	2019-02-06	2018-09-24	male	non_reader
2	0	2018-11-24	3.0	0	2019-02-06	2018-09-24	male	non_reader
3	0	2018-11-28	6.0	0	2019-02-06	2018-09-24	male	non_reader
4	0	2018-12-02	6.0	0	2019-02-06	2018-09-24	male	non_reader

In [46]:

```
df3.isna().sum()
```

Out[46]:

```
uid                0
dt_x               0
active_mins        0
variant_number     0
dt_y               0
signup_date        0
gender             0
user_type          0
dtype: int64
```

In [47]:

```
df3['variant_number'].value_counts()
```

Out[47]:

```
0    989328
1    200599
Name: variant_number, dtype: int64
```

In [48]:

```
df3['gender'].value_counts()
```

Out[48]:

```
male       743499
female     290241
unknown    156187
Name: gender, dtype: int64
```

In [49]:

```
df3['user_type'].value_counts()
```

Out[49]:

```
non_reader     655429
reader         454749
contributor     73793
new_user         5956
Name: user_type, dtype: int64
```

In [50]:

```
stats3 = df3.groupby(['user_type'])['active_mins'].agg(['mean', 'count', 'std', 'var']).
reset_index()
stats3
```

Out[50]:

	user_type	mean	count	std	var
0	contributor	66.897727	73793	102.641666	10535.311594
1	new_user	4.621390	5956	6.712236	45.054112
2	non_reader	5.680745	655429	12.442689	154.820498
3	reader	28.669609	454749	48.017817	2305.710726

In [51]:

```
stats3_g = df3.groupby(['gender'])['active_mins'].agg(['mean', 'count', 'std', 'var']).r
eset_index()
stats3_g
```

Out[51]:

	gender	mean	count	std	var
0	female	16.371446	290241	39.679423	1574.456573
1	male	19.421729	743499	45.431600	2064.030249
2	unknown	16.219077	156187	40.963706	1678.025245

In [52]:

```
dif3 = math.sqrt((stats3.loc[0]['var']/stats3.loc[0]['count'])+((stats3.loc[1]['var']/st
ats3.loc[1]['count'])))
upper3 = (stats3.loc[1]['mean']-stats3.loc[0]['mean']) + (1.96 * dif3)
lower3 = (stats3.loc[1]['mean']-stats3.loc[0]['mean']) - (1.96 * dif3)
print([round(lower3,2),round(upper3,2)])
```

```
[-63.04, -61.52]
```

In [53]:

```
dif3 = math.sqrt((stats3_g.loc[0]['var']/stats3_g.loc[0]['count'])+((stats3_g.loc[1]['va
r']/stats3_g.loc[1]['count'])))
upper3 = (stats3_g.loc[1]['mean']-stats3_g.loc[0]['mean']) + (1.96 * dif3)
lower3 = (stats3_g.loc[1]['mean']-stats3_g.loc[0]['mean']) - (1.96 * dif3)
print([round(lower3,2),round(upper3,2)])
```

```
[2.87, 3.23]
```

In [77]:

```
# new_t1(active-mins)+t2
df.head()
```

Out[77]:

	uid	dt_x	active_mins	variant_number	dt_y	signup_date
0	0	2019-02-22	5.0	0	2019-02-06	2018-09-24
1	0	2019-03-11	5.0	0	2019-02-06	2018-09-24
2	0	2019-03-18	3.0	0	2019-02-06	2018-09-24
3	0	2019-03-22	4.0	0	2019-02-06	2018-09-24
4	0	2019-04-03	9.0	0	2019-02-06	2018-09-24

In [78]:

```
# new_t3(active_mins) + t2
df2.head()
```

Out[78]:

	uid	dt_x	active_mins	variant_number	dt_y	signup_date
0	0	2018-09-24	3.0	0	2019-02-06	2018-09-24
1	0	2018-11-08	4.0	0	2019-02-06	2018-09-24
2	0	2018-11-24	3.0	0	2019-02-06	2018-09-24
3	0	2018-11-28	6.0	0	2019-02-06	2018-09-24
4	0	2018-12-02	6.0	0	2019-02-06	2018-09-24

In [54]:

```
# new_t3+t2+t4
df3.head()
```

Out[54]:

	uid	dt_x	active_mins	variant_number	dt_y	signup_date	gender	user_type
0	0	2018-09-24	3.0	0	2019-02-06	2018-09-24	male	non_reader
1	0	2018-11-08	4.0	0	2019-02-06	2018-09-24	male	non_reader
2	0	2018-11-24	3.0	0	2019-02-06	2018-09-24	male	non_reader
3	0	2018-11-28	6.0	0	2019-02-06	2018-09-24	male	non_reader
4	0	2018-12-02	6.0	0	2019-02-06	2018-09-24	male	non_reader

In [72]:

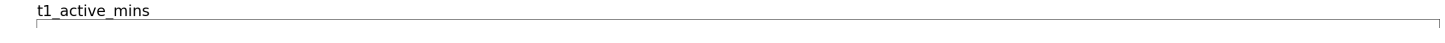
```
t4.head()
```

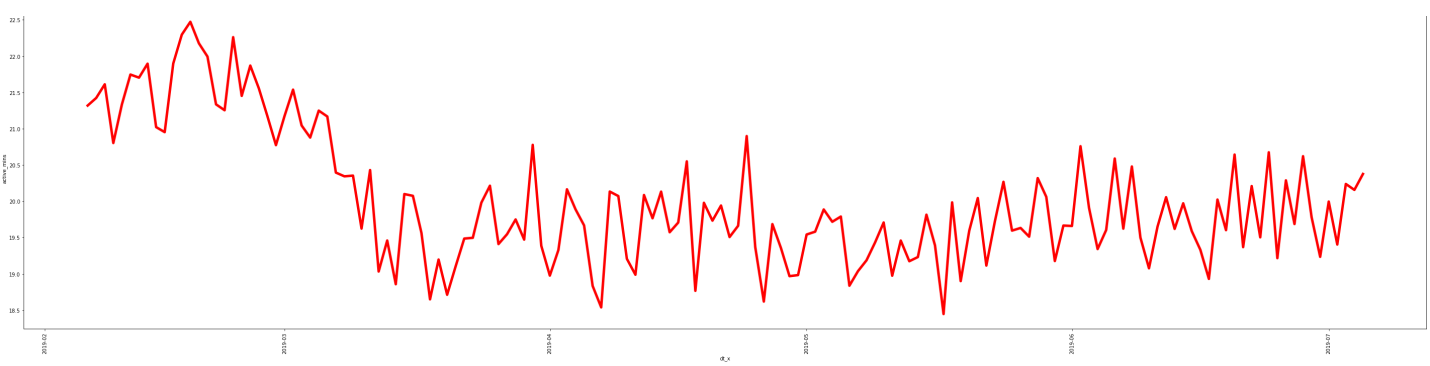
Out[72]:

	uid	gender	user_type
0	0	male	non_reader
1	1	male	reader
2	2	male	non_reader
3	3	male	non_reader
4	4	male	non_reader

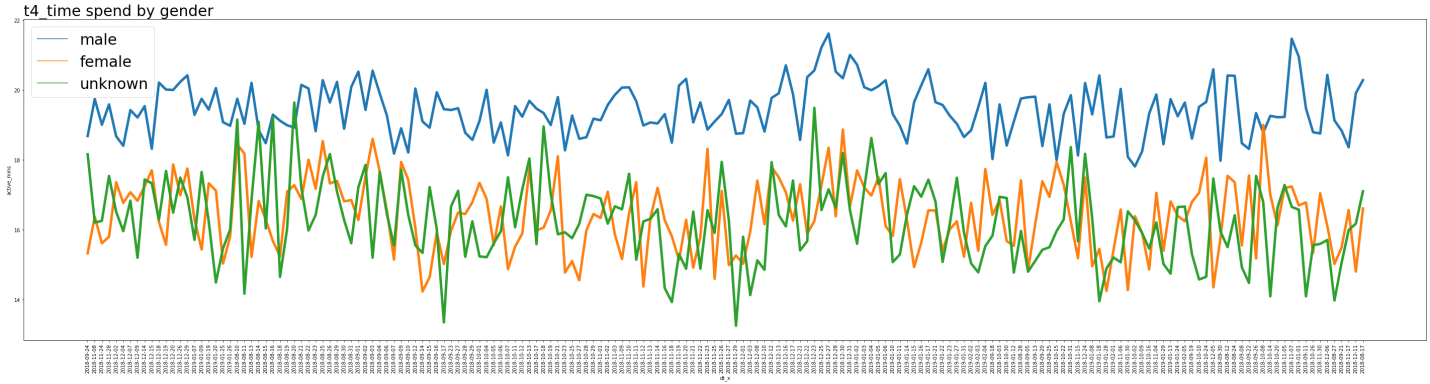
In [140]:

```
plt.figure(figsize=(50,40))
plt.subplot(3,1,1)
plt.title("t1_active_mins", loc='left', fontsize=30)
sns.lineplot(data=df, x="dt_x", y="active_mins", ci=None, color='red',linewidth=5)
plt.xticks(rotation=90)
plt.subplot(3,1,2)
plt.title("t4_time spend by gender", loc='left', fontsize=30)
sns.lineplot(data=df3, x="dt_x", y="active_mins", hue='gender', ci=None, linewidth=5)
plt.xticks(rotation=90)
plt.legend(prop={'size':30})
plt.subplot(3,1,3)
plt.title("t4_ime spend by user_type", loc='left', fontsize=30)
sns.lineplot(data=df3, x="dt_x", y="active_mins", hue="user_type", ci=None, linewidth=5)
plt.xticks(rotation=90)
plt.legend(prop={'size':30})
plt.show()
```

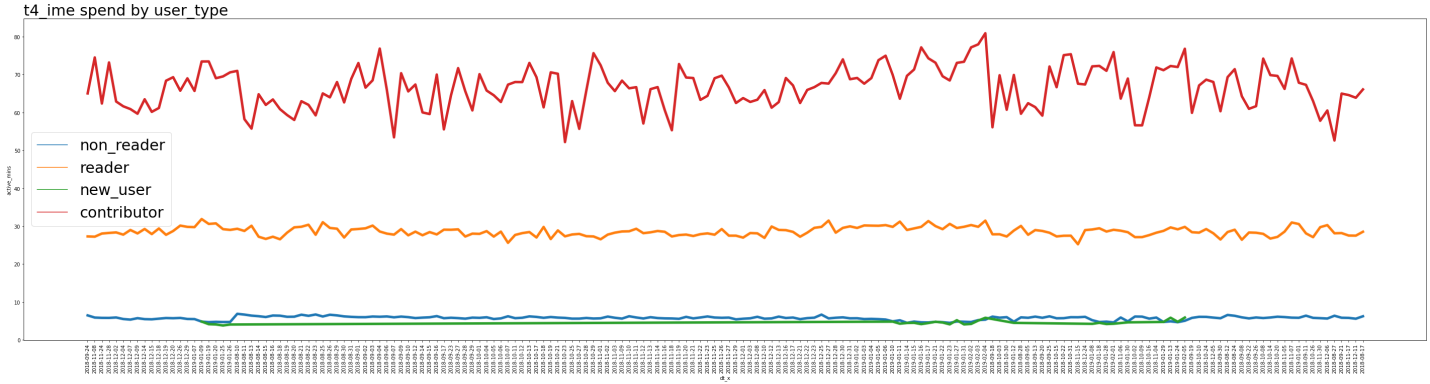




t4_time spend by gender



t4_ime spend by user_type



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: