In [1]:
```
import pandas as pd
import math
from scipy import stats as stat
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
df_active = pd.read_csv(r"D:\Temp\Quora\t1_user_active_min.csv")
df_active
```

Out[2]:

	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
3	0	2019-03-22	4.0
4	0	2019-04-03	9.0
...	...	...	...
1066397	49999	2019-04-14	24.0
1066398	49999	2019-04-26	1.0
1066399	49999	2019-05-31	6.0
1066400	49999	2019-06-02	2.0
1066401	49999	2019-06-24	5.0

1066402 rows × 3 columns

In [3]:
```
len(df_active.uid.unique())
```

Out[3]:
```
46633
```

In [4]:
```
#There can be only 24*60 minutes in a day
#So filtering out such records where active_mins > 24*60. Since its mentioned that ther

df_active = df_active[(df_active.active_mins < (24*60))]
print(len(df_active.uid.unique()))
print(df_active.shape)
```

```
46633
(1066230, 3)
```

In [5]:
```
# IQ Range for Filtering Outliers
q1 = df_active.active_mins.quantile(0.25)
q3 = df_active.active_mins.quantile(0.75)
print(q1,q3)
iqr = q3 - q1
c = 1.5*iqr
df_active = df_active[((df_active.active_mins)> q1 - c)]
df_active = df_active[((df_active.active_mins)< q3 + c)]

print(len(df_active.uid.unique()))
print(df_active.shape)
```

```
2.0 17.0
46605
(931953, 3)
```

In [6]:
```
df_variant = pd.read_csv(r"D:\Temp\Quora\t2_user_variant.csv")
print(len(df_variant.uid.unique()))
print(df_variant.shape)
df_variant
```

```
50000
(50000, 4)
```

Out[6]:

	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
...	...	...	...	...
49995	49995	1	2019-02-06	2018-10-11
49996	49996	1	2019-02-06	2014-12-06
49997	49997	1	2019-02-06	2018-11-15
49998	49998	1	2019-02-06	2016-04-05
49999	49999	1	2019-02-06	2015-12-29

50000 rows × 4 columns

In [7]:
```
#Merge
merged_df = pd.merge(left = df_active,right = df_variant, left_on='uid', right_on='uid'
print(merged_df.shape)
```

```
(931953, 6)
```

In [8]:
```
merged_df
```

Out[8]:

	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
...	...	...	...	...	...	...
931948	49999	2019-04-14	24.0	1	2019-02-06	2015-12-29
931949	49999	2019-04-26	1.0	1	2019-02-06	2015-12-29
931950	49999	2019-05-31	6.0	1	2019-02-06	2015-12-29

	uid	dt_x	active_mins	variant_number	dt_y	signup_date
931951	49999	2019-06-02	2.0	1	2019-02-06	2015-12-29
931952	49999	2019-06-24	5.0	1	2019-02-06	2015-12-29

931953 rows × 6 columns

In [9]:
```
# Compute Mean time across all user
temp_df = merged_df.groupby(['uid','variant_number'], as_index=False)['active_mins'].me
temp_df
```

Out[9]:

	uid	variant_number	active_mins
0	0	0	3.307692
1	1	0	19.800000
2	2	0	2.428571
3	3	0	3.208333
4	4	0	1.950000
...	...	...	...
46600	49995	1	5.277778
46601	49996	1	12.000000
46602	49997	1	7.150943
46603	49998	1	10.931034
46604	49999	1	6.500000

46605 rows × 3 columns

In [10]:
```
#Compute Mean time across Variants
temp_df1 = merged_df.groupby(['variant_number'], as_index=False)['active_mins'].mean()
temp_df1
```

Out[10]:

	variant_number	active_mins
0	0	7.480607
1	1	8.676429

In [11]:
```
# Calculating Confidence Interval
score = merged_df.groupby(['variant_number'])['active_mins'].agg(['mean', 'count', 'std
score
```

Out[11]:

	mean	count	std	var
variant_number				
0	7.480607	777891	8.491041	72.097780
1	8.676429	154062	8.688977	75.498315

In [12]:
```
# calculate Lower and Upper Bound
diff = math.sqrt((score.loc[0]['var']/score.loc[0]['count'])+((score.loc[1]['var']/scor
u = (score.loc[1]['mean']-score.loc[0]['mean']) + (1.96 * diff)
l = (score.loc[1]['mean']-score.loc[0]['mean']) - (1.96 * diff)
print([round(l,2),round(u,2)])
```

```
[1.15, 1.24]
```

In [13]:
```
a = merged_df.loc[merged_df.uid == 0 , 'active_mins']
b = merged_df.loc[merged_df.uid == 1 , 'active_mins']

stat.ttest_ind(a,b,equal_var=False)
```

Out[13]:
```
Ttest_indResult(statistic=-3.059634775607129, pvalue=0.036272908585250244)
```

Since we can see that p-value is less than 0.05, I conclude that new UI is better than old UI

In [14]:
```
df_active_pre = pd.read_csv(r"D:\Temp\Quora\t3_user_active_min_pre.csv")
df_active_pre
```

Out[14]:

	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
...	...	...	...
1190088	49999	2018-09-15	5.0
1190089	49999	2018-09-26	8.0
1190090	49999	2018-10-20	29.0
1190091	49999	2018-12-14	3.0
1190092	49999	2019-01-28	32.0

1190093 rows × 3 columns

In [15]:
```
df_active_pre = df_active_pre[(df_active_pre.active_mins < (24*60))]
print(len(df_active_pre.uid.unique()))
print(df_active_pre.shape)
```

```
49697
(1189927, 3)
```

In [16]:
```
# IQ Range for Filtering Outliers
q1 = df_active_pre.active_mins.quantile(0.25)
q3 = df_active_pre.active_mins.quantile(0.75)
print(q1,q3)
iqr = q3 - q1
c = 1.5*iqr
df_active_pre = df_active_pre[((df_active_pre.active_mins)> q1 - c)]
df_active_pre = df_active_pre[((df_active_pre.active_mins)< q3 + c)]
```

```
print(len(df_active_pre.uid.unique()))
print(df_active_pre.shape)
```

```
2.0 14.0
49643
(1024286, 3)
```

In [17]:
```
#Merge
merged_df1 = pd.merge(left = df_active_pre, right = df_variant, left_on='uid', right_on
print(merged_df1.shape)
merged_df1
```

```
(1024286, 6)
```

Out[17]:

	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
...	...	...	...	...	...	...
1024281	49998	2019-02-05	12.0	1	2019-02-06	2016-04-05
1024282	49999	2018-09-15	5.0	1	2019-02-06	2015-12-29
1024283	49999	2018-09-26	8.0	1	2019-02-06	2015-12-29
1024284	49999	2018-10-20	29.0	1	2019-02-06	2015-12-29
1024285	49999	2018-12-14	3.0	1	2019-02-06	2015-12-29

1024286 rows × 6 columns

In [18]:
```
# Compute Mean time across all user
temp_df2 = merged_df1.groupby(['uid','variant_number'], as_index=False)['active_mins'].
temp_df2
```

Out[18]:

	uid	variant_number	active_mins
0	0	0	3.333333
1	1	0	22.272727
2	2	0	3.700000
3	3	0	3.833333
4	4	0	2.357143
...	...	...	...
49638	49995	1	2.615385
49639	49996	1	5.714286
49640	49997	1	3.608696
49641	49998	1	5.166667

	uid	variant_number	active_mins
49642	49999	1	11.250000

49643 rows × 3 columns

In [19]:
```
#Compute Mean time across Variants
temp_df3 = merged_df1.groupby(['variant_number'], as_index=False)['active_mins'].mean()
temp_df3
```

Out[19]:

	variant_number	active_mins
0	0	6.459824
1	1	5.993754

In [20]:
```
before_mean = temp_df2.active_mins.mean()
before_var = temp_df2.active_mins.var()
before_n = temp_df2.shape[0]

after_mean = temp_df[temp_df.variant_number==1]['active_mins'].mean()
after_var = temp_df[temp_df.variant_number==1]['active_mins'].var()
after_n = temp_df[temp_df.variant_number==1].shape[0]
```

In [21]:
```
diff1 = math.sqrt((before_var/before_n)+(after_var/after_n))
u2 = (after_mean-before_mean) + (1.96 * diff1)
l2 = (after_mean-before_mean) - (1.96 * diff1)
print([round(l2,2),round(u2,2)])
```

```
[1.81, 2.01]
```

In [22]:
```
df_user = pd.read_csv(r"D:\Temp\Quora\t4_user_attributes.csv")
df_user
```

Out[22]:

	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
...	...	...	...
49995	49995	unknown	non_reader
49996	49996	male	non_reader
49997	49997	female	reader
49998	49998	male	non_reader
49999	49999	female	non_reader

50000 rows × 3 columns

In [23]:
```
#Merge
new_df = pd.merge(left = df_user,right = df_variant, left_on='uid', right_on='uid')
print(new_df.shape)
new_df
```

```
(50000, 6)
```

Out[23]:

	uid	gender	user_type	variant_number	dt	signup_date
0	0	male	non_reader	0	2019-02-06	2018-09-24
1	1	male	reader	0	2019-02-06	2016-11-07
2	2	male	non_reader	0	2019-02-06	2018-09-17
3	3	male	non_reader	0	2019-02-06	2018-03-04
4	4	male	non_reader	0	2019-02-06	2017-03-09
...	...	...	...	...	...	...
49995	49995	unknown	non_reader	1	2019-02-06	2018-10-11
49996	49996	male	non_reader	1	2019-02-06	2014-12-06
49997	49997	female	reader	1	2019-02-06	2018-11-15
49998	49998	male	non_reader	1	2019-02-06	2016-04-05
49999	49999	female	non_reader	1	2019-02-06	2015-12-29

50000 rows × 6 columns

Representing Variants per User Type

In [24]:
```
pd.crosstab(new_df.variant_number,new_df.user_type).apply(lambda r: round(r/r.sum(),2),
```

Out[24]:

user_type	contributor	new_user	non_reader	reader
variant_number				
0	0.02	0.09	0.72	0.17
1	0.01	0.12	0.74	0.13

Representing Variants per Gender

In [25]:
```
pd.crosstab(new_df.variant_number,new_df.gender).apply(lambda r: round(r/r.sum(),2), ax
```

Out[25]:

gender	female	male	unknown
variant_number			
0	0.29	0.56	0.15
1	0.29	0.55	0.16

Representing Gender per User Type

In [26]:
```
pd.crosstab(new_df.gender,new_df.user_type).apply(lambda r: round(r/r.sum(),2), axis=1)
```

Out[26]:

user_type	contributor	new_user	non_reader	reader
gender				
female	0.02	0.11	0.72	0.15
male	0.02	0.08	0.72	0.18
unknown	0.01	0.13	0.74	0.12

Merging User's Active after the Test v/s General

In [27]:
```
new_df2 = pd.merge(left=df_active,right=new_df, left_on='uid', right_on='uid')
new_df2.head()
```

Out[27]:

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

In [28]:
```
temps = new_df2.groupby(['uid','variant_number','gender','user_type'], as_index=False)[
temps.shape
```

Out[28]:
```
(46605, 5)
```

In [29]:
```
temps
```

Out[29]:

	uid	variant_number	gender	user_type	active_mins
0	0	0	male	non_reader	3.307692
1	1	0	male	reader	19.800000
2	2	0	male	non_reader	2.428571
3	3	0	male	non_reader	3.208333
4	4	0	male	non_reader	1.950000
...	...	...	...	...	...
46600	49995	1	unknown	non_reader	5.277778
46601	49996	1	male	non_reader	12.000000
46602	49997	1	female	reader	7.150943
46603	49998	1	male	non_reader	10.931034
46604	49999	1	female	non_reader	6.500000

46605 rows × 5 columns

Representing Active Mins after the A/B test per User-Type per Varaint

In [30]:
```
show = temps.groupby(['variant_number','user_type'])['active_mins'].agg(['mean', 'count

show
```

Out[30]:

	variant_number	user_type	mean	count	std	var
0	0	contributor	14.329884	903	6.658750	44.338948
1	0	new_user	3.134401	2372	2.445971	5.982776
2	0	non_reader	3.941514	27454	2.541566	6.459556
3	0	reader	10.810775	6679	5.742781	32.979539
4	1	contributor	13.545419	126	5.701383	32.505770
5	1	new_user	5.425513	807	4.388225	19.256515
6	1	non_reader	5.976151	7008	3.323227	11.043837
7	1	reader	12.458016	1256	5.177517	26.806686

In [31]:
```
new_df3 = pd.merge(left=df_active_pre,right=new_df, left_on='uid', right_on='uid')
new_df3.head()
```

Out[31]:

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

In [32]:
```
temps2 = new_df3.groupby(['uid','variant_number','gender','user_type'], as_index=False)
temps2.shape
```

Out[32]:
```
(49643, 5)
```

In [33]:
```
temps2
```

Out[33]:

	uid	variant_number	gender	user_type	active_mins
0	0	0	male	non_reader	3.333333
1	1	0	male	reader	22.272727
2	2	0	male	non_reader	3.700000
3	3	0	male	non_reader	3.833333
4	4	0	male	non_reader	2.357143
...	...	...	...	...	...

	uid	variant_number	gender	user_type	active_mins
49638	49995	1	unknown	non_reader	2.615385
49639	49996	1	male	non_reader	5.714286
49640	49997	1	female	reader	3.608696
49641	49998	1	male	non_reader	5.166667
49642	49999	1	female	non_reader	11.250000

49643 rows × 5 columns

Showing Active Mins Before the A/B Test per User-Type per Varaint

In [34]:
```
show2 = temps2.groupby(['variant_number','user_type'])['active_mins'].agg(['mean', 'cou

show2
```

Out[34]:

	variant_number	user_type	mean	count	std	var
0	0	contributor	11.689043	907	4.403890	19.394249
1	0	new_user	4.152310	3477	4.052415	16.422068
2	0	non_reader	3.928970	28623	2.140313	4.580940
3	0	reader	9.474669	6726	4.143193	17.166047
4	1	contributor	10.841399	128	4.223605	17.838840
5	1	new_user	4.332352	1165	4.159181	17.298788
6	1	non_reader	4.001534	7350	2.161581	4.672431
7	1	reader	9.059224	1267	3.751360	14.072705

In [35]:
```
stats = temps.groupby(['variant_number','gender','user_type'])['active_mins'].agg(['mea
stats
```

Out[35]:

	variant_number	gender	user_type	mean	count	std	var
0	0	female	contributor	12.369391	222	6.230767	38.822464
1	0	female	new_user	3.069377	762	2.345331	5.500576
2	0	female	non_reader	3.820953	7937	2.503481	6.267417
3	0	female	reader	10.343886	1812	5.537983	30.669252
4	0	male	contributor	15.245286	586	6.612557	43.725916
5	0	male	new_user	3.224292	1133	2.589829	6.707212
6	0	male	non_reader	4.023706	15186	2.554948	6.527761
7	0	male	reader	11.110354	4089	5.849210	34.213263
8	0	unknown	contributor	13.264665	95	6.827822	46.619155
9	0	unknown	new_user	3.024762	477	2.240080	5.017957

	variant_number	gender	user_type	mean	count	std	var
10	0	unknown	non_reader	3.874265	4331	2.553555	6.520642
11	0	unknown	reader	10.323660	778	5.544363	30.739966
12	1	female	contributor	13.005459	26	4.730706	22.379583
13	1	female	new_user	5.242303	275	3.680762	13.548008
14	1	female	non_reader	5.796804	1974	3.379302	11.419679
15	1	female	reader	11.866491	333	4.977957	24.780061
16	1	male	contributor	14.211443	81	5.820732	33.880926
17	1	male	new_user	5.499319	376	4.648713	21.610528
18	1	male	non_reader	6.106726	3927	3.307700	10.940879
19	1	male	reader	12.723727	747	5.168788	26.716372
20	1	unknown	contributor	11.444946	19	6.082392	36.995490
21	1	unknown	new_user	5.570586	156	4.879243	23.807011
22	1	unknown	non_reader	5.832759	1107	3.258879	10.620291
23	1	unknown	reader	12.449441	176	5.511405	30.375586

Covariate Tests

In [36]:
```
def ci_covariates(stats,cov):
    control = stats[(stats.variant_number==0) & (stats.gender==cov[0]) & (stats.user_ty
    treat = stats[(stats.variant_number==1) & (stats.gender==cov[0]) & (stats.user_type

    sigma_dif = math.sqrt((control.iloc[0]['var']/control.iloc[0]['count'])+((treat.ilo
    upper = (treat.iloc[0]['mean']-control.iloc[0]['mean']) + (1.96 * sigma_dif)
    lower = (treat.iloc[0]['mean']-control.iloc[0]['mean']) - (1.96 * sigma_dif)
    return (lower,upper)
```

In [37]:
```
utypes=stats['user_type'].unique()
utypes=utypes.tolist()

ugender=stats['gender'].unique()
ugender=ugender.tolist()

covariates = [(a,b) for a in ugender for b in utypes]
covariates

for c in covariates:
    inter = ci_covariates(stats,c)
    print(c, '---------------', inter)
```

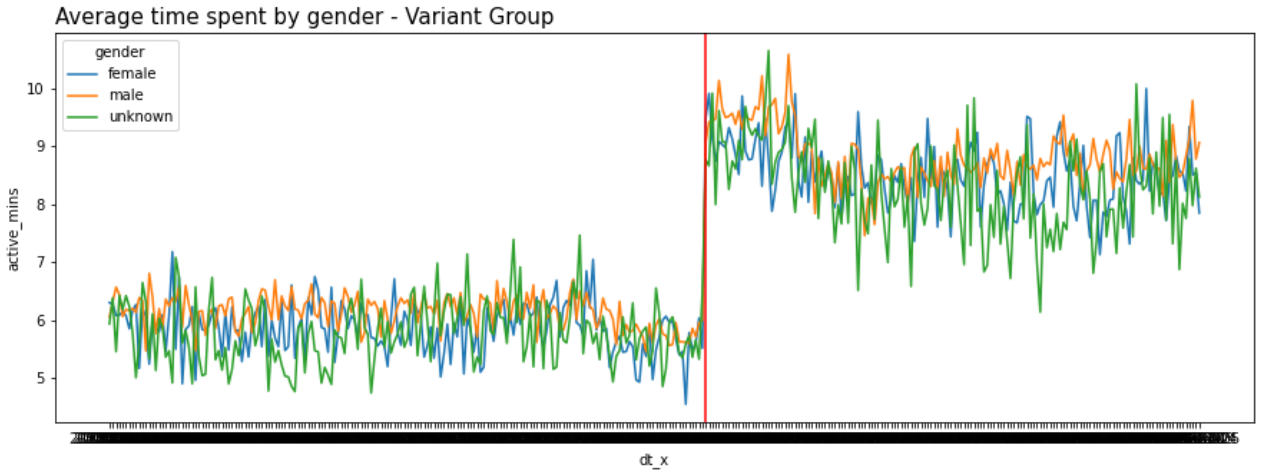
```
('female', 'contributor') --------------- (-1.3585433720015292, 2.630678602938575)
('female', 'new_user') --------------- (1.7071052376824718, 2.6387473875400373)
('female', 'non_reader') --------------- (1.8169263369532478, 2.1347774484366333)
('female', 'reader') --------------- (0.9302437544480275, 2.1149675844797367)
('male', 'contributor') --------------- (-2.409898121850264, 0.34221169879544777)
('male', 'new_user') --------------- (1.7815322157666627, 2.7685215708035926)
('male', 'non_reader') --------------- (1.9718703932871553, 2.1941699687824854)
('male', 'reader') --------------- (1.201623976993931, 2.0251229165302376)
('unknown', 'contributor') --------------- (-4.879993082704576, 1.2405556327609548)
```

```
('unknown', 'new_user') --------------- (1.7541954476909873, 3.3374527000848557)
('unknown', 'non_reader') --------------- (1.752001007901167, 2.164986220795146)
('unknown', 'reader') --------------- (1.2231168087345048, 3.0284463405269846)
```

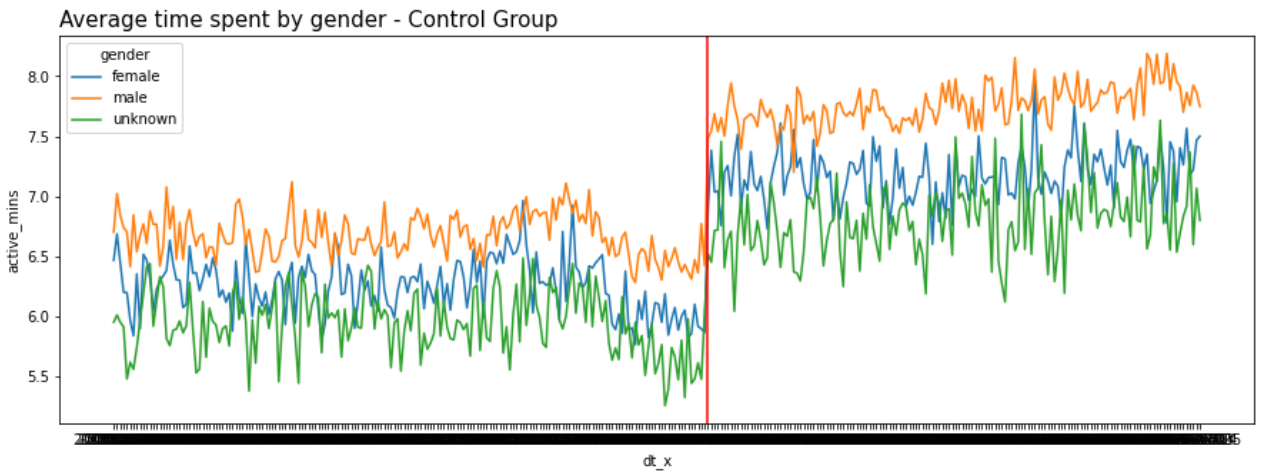
In [38]:
```
t1 = new_df3[new_df3.variant_number == 1]
t2 = new_df2[new_df2.variant_number == 1]
daily_usr_1 = pd.concat([t1,t2],ignore_index=True)

t3 = new_df3[new_df3.variant_number == 0]
t4 = new_df2[new_df2.variant_number == 0]
daily_usr_0 = pd.concat([t3,t4],ignore_index=True)
```

In [39]:
```
plt.figure(figsize=(15,5))
plt.title('Average time spent by gender - Variant Group',loc='left', fontsize=15)
data=daily_usr_1.groupby(['gender','dt_x'],as_index=False).active_mins.mean()
sns.lineplot(data=data,x='dt_x',y='active_mins',hue='gender')
plt.axvline(x = "2019-02-06", color = 'red')
plt.savefig("Time Series Analysis by gender - Variant Group")
```



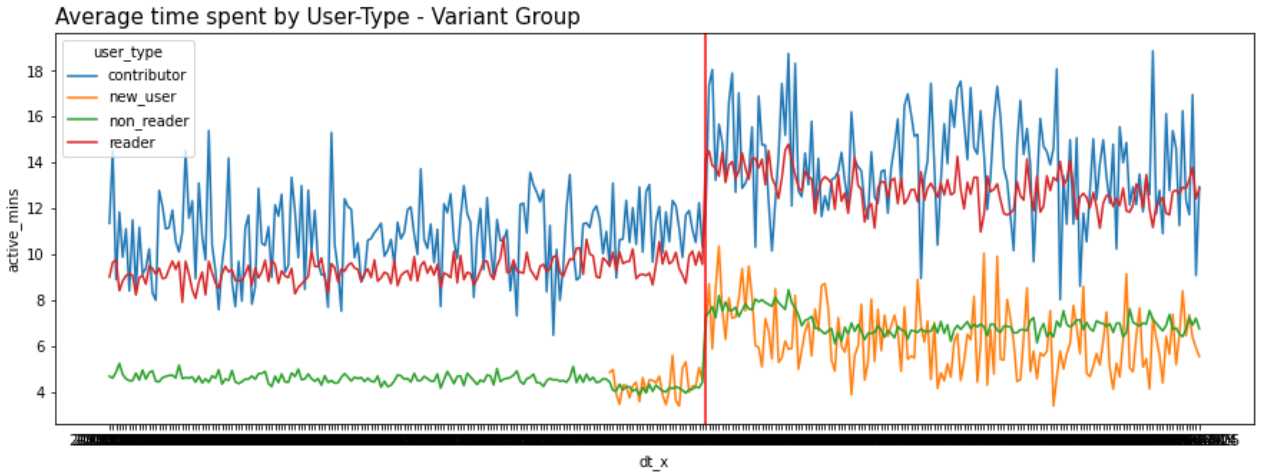
In [40]:
```
plt.figure(figsize=(15,5))
plt.title('Average time spent by gender - Control Group',loc='left',fontsize=15)
data1=daily_usr_0.groupby(['gender','dt_x'],as_index=False).active_mins.mean()
sns.lineplot(data=data1,x='dt_x',y='active_mins',hue='gender')
plt.axvline(x = "2019-02-06", color = 'red')
plt.savefig("Time Series Analysis by gender - Control Group")
```



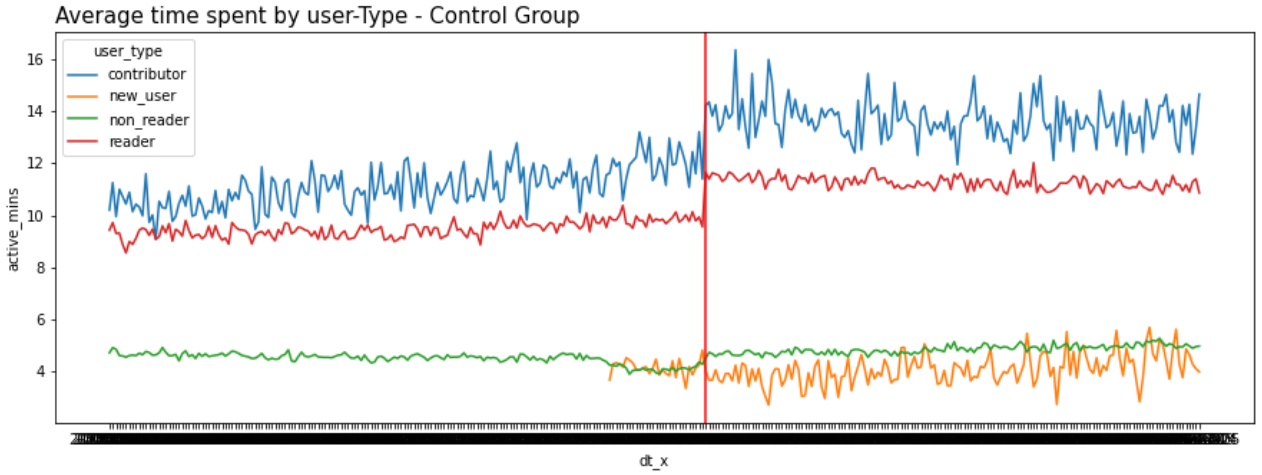
Thus we see from above graph that for Treatment group there is significant improvement in average

acitve minutes. Also, comparing within the genders, we see male members have a higher activity in comparison their female counterparts.

In [41]:
```
plt.figure(figsize=(15,5))
plt.title('Average time spent by User-Type - Variant Group',loc='left', fontsize=15)
data=daily_usr_1.groupby(['user_type','dt_x'],as_index=False).active_mins.mean()
sns.lineplot(data=data,x='dt_x',y='active_mins',hue='user_type')
plt.axvline(x = "2019-02-06", color = 'red')
plt.savefig("Time Series Analysis by User-Type - Variant Group")
```



In [42]:
```
plt.figure(figsize=(15,5))
plt.title('Average time spent by user-Type - Control Group',loc='left', fontsize=15)
data=daily_usr_0.groupby(['user_type','dt_x'],as_index=False).active_mins.mean()
sns.lineplot(data=data,x='dt_x',y='active_mins',hue='user_type')
plt.axvline(x = "2019-02-06", color = 'red')
plt.savefig("Time Series Analysis by User-Type - Control Group")
```



In [43]:
```
new_df.groupby(['variant_number','user_type']).agg({'uid':'count'})
```

Out[43]:

		uid
variant_number	**user_type**	
0	**contributor**	915
	new_user	3653
	non_reader	28699

		uid
variant_number	**user_type**	
	reader	6733
1	**contributor**	129
	new_user	1235
	non_reader	7367
	reader	1269

In [ ]: