User Total Active Time as Metric

In [629]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

In [630]:
```
# reading the data sets
t1_user_active_min     = pd.read_csv("t1_user_active_min.csv")
t2_user_variant        = pd.read_csv("t2_user_variant.csv")
t3_user_active_min_pre = pd.read_csv("t3_user_active_min_pre.csv")
t4_user_attributes     = pd.read_csv("t4_user_attributes.csv")
```

Use table 1 and 2

In [631]:
```
t1_user_active_min.head()
```

Out[631]:

	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

In [632]:
```
t2_user_variant.head()
```

Out[632]:

	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 [633]:
```
# Drop data with active_mins more than (24 hrs * 60 mins) on a single da
te.
t1_user_active_min = t1_user_active_min[t1_user_active_min.active_mins <
= 24*60]
```

In [634]:
```
# Total minutes per user
t1_user_active_min_total = t1_user_active_min.groupby(by=['uid'], as_ind
ex=False).sum()
```

In [635]:
```
# merge t1 and t2 by uid
t12 = pd.merge(t2_user_variant, t1_user_active_min_total, on='uid')
```

In [636]:
```
t12.head()
```

Out[636]:

	uid	variant_number	dt	signup_date	active_mins
0	0	0	2019-02-06	2018-09-24	43.0
1	1	0	2019-02-06	2016-11-07	15205.0
2	2	0	2019-02-06	2018-09-17	17.0
3	3	0	2019-02-06	2018-03-04	77.0
4	4	0	2019-02-06	2017-03-09	39.0

In [637]:
```
t12.variant_number.value_counts()
```

Out[637]:
```
0    37425
1     9208
Name: variant_number, dtype: int64
```

Compute confidence interval on difference of means.

In [638]:
```
stats12 = t12.groupby(['variant_number'])['active_mins'].agg(['mean', 'c
ount', 'std','var'])
stats12
```

Out[638]:

variant_number	mean	count	std	var
0	458.221162	37425	1653.447132	2.733887e+06
1	458.402476	9208	1680.571091	2.824319e+06

In [639]:
```
sigma_diff = np.sqrt(stats12.loc[0]['var'] / stats12.loc[0]['count'] +
                     stats12.loc[1]['var'] / stats12.loc[1]['count'])
upper = (stats12.loc[1]['mean'] - stats12.loc[0]['mean']) + (1.96 * sigm
a_diff)
lower = (stats12.loc[1]['mean'] - stats12.loc[0]['mean']) - (1.96 * sigm
a_diff)
print([lower, upper])
```

```
[-38.01476992103114, 38.37739748719736]
```

Perform a t-test:

In [640]:
```
stats.ttest_ind(t12.active_mins[t12.variant_number==1], t12.active_mins[
t12.variant_number==0], equal_var =False)
```

Out[640]:
```
Ttest_indResult(statistic=0.009303964709989936, pvalue=0.99257675062736
44)
```

The high p-value suggests INSUFFICIENT evidence for the new UI design to make a positive impact on total active time per user.

Add Table 3

In [641]:
```
t3_user_active_min_pre.head()
```

Out[641]:

	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 [642]:
```
# Drop data with active_mins more than (24 hrs * 60 mins) on a single da
te.
t3_user_active_min_pre = t3_user_active_min_pre[t3_user_active_min_pre.a
ctive_mins <= 24*60]
```

In [643]:
```
# Total minutes per user
t3_user_active_min_pre_total = t3_user_active_min_pre.groupby(by=['uid'
], as_index=False).sum()
```

In [644]:
```
t123 = pd.merge(t12, t3_user_active_min_pre_total, on='uid', suffixes=(
'_post', '_pre'))
```

In [645]:
```
t123.head()
```

Out[645]:

	uid	variant_number	dt	signup_date	active_mins_post	active_mins_pre
0	0	0	2019-02-06	2018-09-24	43.0	70.0
1	1	0	2019-02-06	2016-11-07	15205.0	19158.0
2	2	0	2019-02-06	2018-09-17	17.0	37.0
3	3	0	2019-02-06	2018-03-04	77.0	108.0
4	4	0	2019-02-06	2017-03-09	39.0	66.0

In [646]:
```
t123['active_mins_diff'] = t123['active_mins_post'] - t123['active_mins_pre']
```

In [648]:
```
stats123 = t123.groupby(['variant_number'])['active_mins_post', 'active_mins_pre', 'active_mins_diff'].agg(['mean', 'count', 'std','var'])
stats123
```

Out[648]:

	active_mins_post				active_mins_pre			
	mean	count	std	var	mean	count	std	v
variant_number								
0	459.544824	37313	1655.747990	2.741501e+06	506.84027	37313	1874.763969	3
1	460.465139	9165	1684.238222	2.836658e+06	295.81102	9165	1118.898491	1

Difference in difference

For each user, first obtain difference in post-test total time and pre-test total time: $\Delta x_1 \equiv x_{1,post} - x_{1,pre}$ and $\Delta x_2 \equiv x_{0,post} - x_{0,pre}$

Then obtain the difference in mean: $\overline{\Delta x_1} - \overline{\Delta x_2}$

Standard Error: $\sqrt{\frac{s^2_{\Delta x_1}}{n_1} + \frac{s^2_{\Delta x_2}}{n_1}}$

In [649]:
```
mean_did = stats123.loc[1,'active_mins_diff']['mean'] - stats123.loc[0,
'active_mins_diff']['mean']

sigma_did = np.sqrt(stats123.loc[1,'active_mins_diff']['var'] / stats123
.loc[1,'active_mins_diff']['count'] +
                    stats123.loc[0,'active_mins_diff']['var'] / stats123
.loc[0,'active_mins_diff']['count'])

upper = mean_did + (1.96 * sigma_did)
lower = mean_did - (1.96 * sigma_did)
print([lower, upper])
```

```
[188.86398661059266, 235.03514450519842]
```

Perform a t-test:

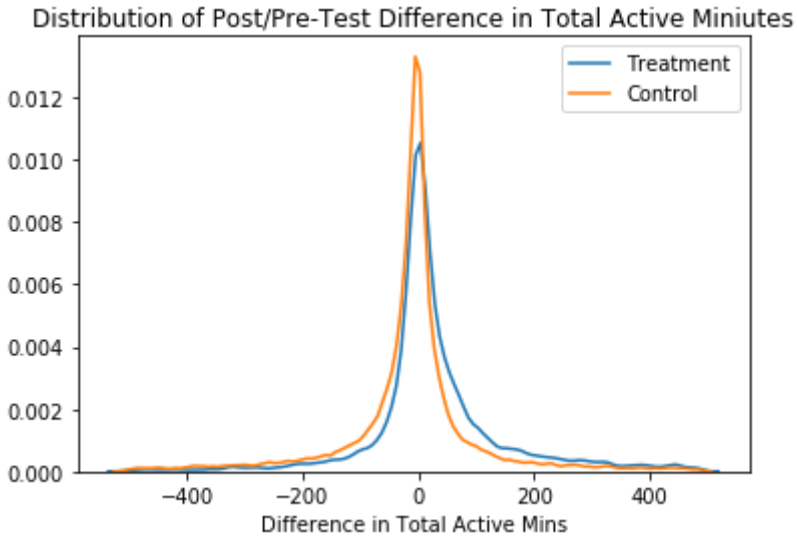
In [650]:
```
stats.ttest_ind(t123.active_mins_diff[t123.variant_number==1], t123.acti
ve_mins_diff[t123.variant_number==0], equal_var =False)
```

Out[650]:
```
Ttest_indResult(statistic=17.994833460393252, pvalue=1.462502346186859e
-71)
```

The low p-value suggests high evidence for the new UI design to make a positive impact on total active time per user.

In [651]:
```
def build_dist(df1, df2, col1, col2, xlabel, legend1, legend2, title):
    sns.distplot(df1[col1], hist=False, label = legend1, axlabel = xlabe
l)
    sns.distplot(df2[col2], hist=False, label = legend2, axlabel = xlabe
l)
    plt.title(title)
    #plt.xlim([-1000,1000])
    plt.show()
```

In [652]:
```
build_dist(t123[(t123.variant_number==1) &
                (t123.active_mins_diff < t123.active_mins_diff.quantile(
0.95)) &
                (t123.active_mins_diff > t123.active_mins_diff.quantile(
0.05))],
           t123[(t123.variant_number==0) &
                (t123.active_mins_diff < t123.active_mins_diff.quantile(
0.95)) &
                (t123.active_mins_diff > t123.active_mins_diff.quantile(
0.05))],
           "active_mins_diff","active_mins_diff","Difference in Total Ac
tive Mins", "Treatment","Control",
           title='Distribution of Post/Pre-Test Difference in Total Acti
ve Miniutes')
```



Add Table 4

In [654]:
```
t4_user_attributes.head()
```

Out[654]:

	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 [655]:
```
t1234 = pd.merge(t4_user_attributes, t123, on='uid')
```

In [656]:
```
t1234.head()
```

Out[656]:

	uid	gender	user_type	variant_number	dt	signup_date	active_mins_post	active_mins_pr
0	0	male	non_reader	0	2019-02-06	2018-09-24	43.0	70.
1	1	male	reader	0	2019-02-06	2016-11-07	15205.0	19158.
2	2	male	non_reader	0	2019-02-06	2018-09-17	17.0	37.
3	3	male	non_reader	0	2019-02-06	2018-03-04	77.0	108.
4	4	male	non_reader	0	2019-02-06	2017-03-09	39.0	66.

In [657]:
```
pd.crosstab(t1234['variant_number'],t1234['gender']).apply(lambda r: r/r.sum(), axis=1)
```

Out[657]:

gender	female	male	unknown
variant_number			
0	0.286737	0.561493	0.151770
1	0.282815	0.559083	0.158101

In [658]:
```
pd.crosstab(t1234['variant_number'],t1234['user_type']).apply(lambda r: r/r.sum(), axis=1)
```

Out[658]:

user_type	contributor	new_user	non_reader	reader
variant_number				
0	0.024415	0.061185	0.735240	0.17916
1	0.013857	0.084015	0.764648	0.13748

In [659]:
```
pd.crosstab(t1234['gender'],t1234['user_type']).apply(lambda r: r/r.sum(), axis=1)
```

Out[659]:

user_type	contributor	new_user	non_reader	reader
gender				
female	0.018734	0.074637	0.745166	0.161463
male	0.025849	0.055801	0.732579	0.185772
unknown	0.016170	0.085208	0.764342	0.134280

Gender

In [660]:
```
stats1234_gender = t1234.groupby(['variant_number','gender'])['active_mi
ns_post', 'active_mins_pre', 'active_mins_diff'].agg(['mean', 'count',
'var'])
stats1234_gender
```

Out[660]:

		active_mins_post			active_mins_pre			ac
		mean	count	var	mean	count	var	me
variant_number	gender							
0	female	347.295635	10699	1.775838e+06	388.001309	10699	2.291656e+06	-4
	male	555.593957	20951	3.547350e+06	602.397213	20951	4.430440e+06	-4
	unknown	316.268939	5663	1.506830e+06	377.835246	5663	2.361617e+06	-6
1	female	355.527778	2592	1.705491e+06	224.615741	2592	6.196977e+05	13
	male	534.548009	5124	3.581443e+06	341.800546	5124	1.663136e+06	19
	unknown	386.204969	1449	2.184961e+06	260.536922	1449	9.123354e+05	12

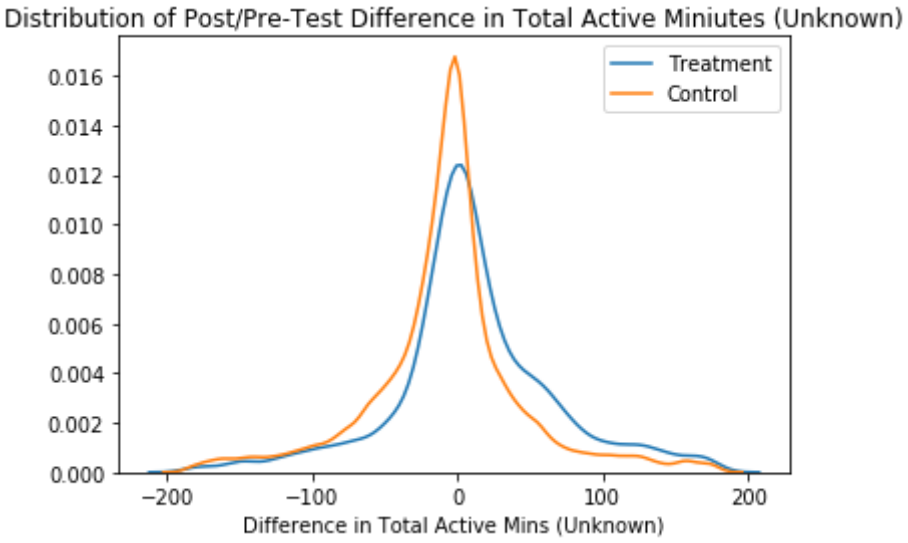
In [714]:
```
def ConfidenceInterval(stats, attribute):
    mean_did = stats.loc[1, attribute]['active_mins_diff', 'mean'] - sta
ts.loc[0,attribute]['active_mins_diff', 'mean']
    SE_did = np.sqrt(stats.loc[1, attribute]['active_mins_diff','var'] /
stats.loc[1, attribute]['active_mins_diff','count'] +
                     stats.loc[0, attribute]['active_mins_diff','var'] /
stats.loc[0, attribute]['active_mins_diff','count'])

    upper = mean_did + (1.96 * SE_did)
    lower = mean_did - (1.96 * SE_did)
    return [lower, upper]
```

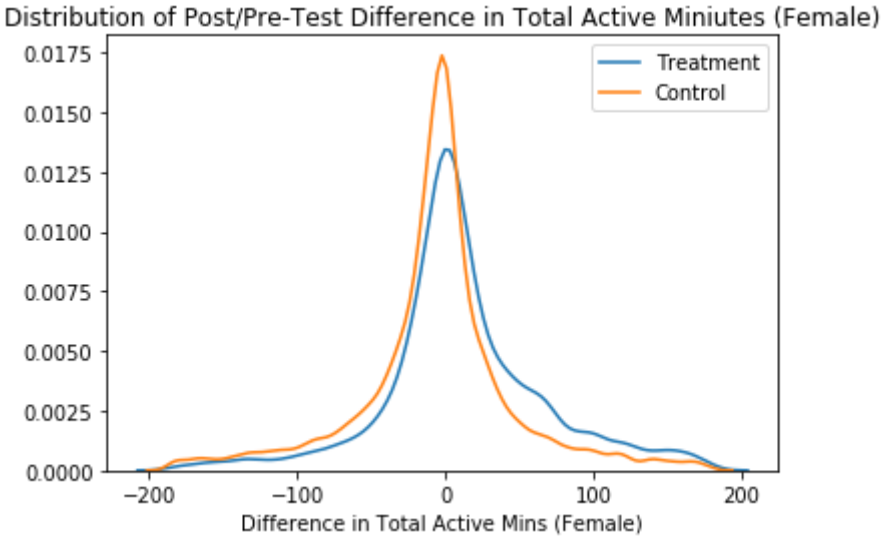
In [715]:
```
intervals = {}
for g in set(t4_user_attributes['gender']):
    intervals[g] = ConfidenceInterval(stats1234_gender, g)
    print(g, ':', ConfidenceInterval(stats1234_gender, g))
```

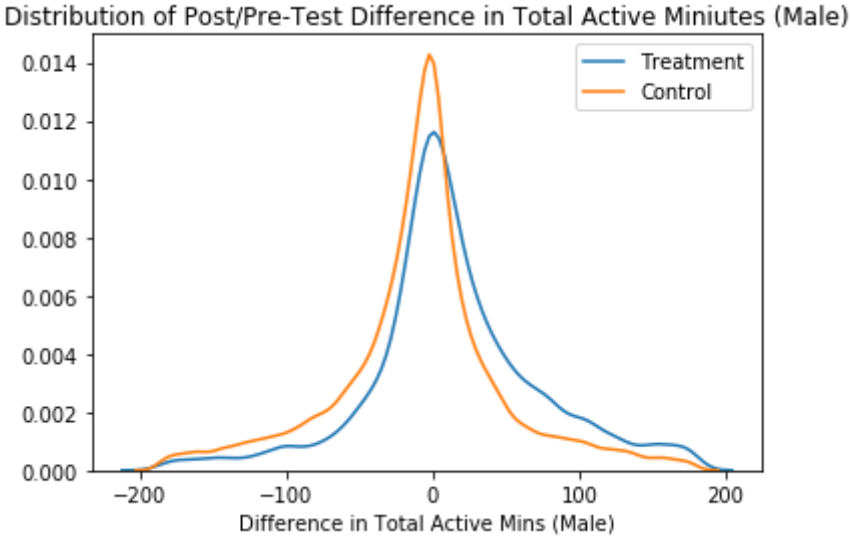
```
unknown : [136.26857431526759, 238.20013476417938]
female : [134.88598802514008, 208.34943290378027]
male : [205.61842607578922, 273.4830101924954]
```

In [767]:

```
for g in set(t4_user_attributes['gender']):
    build_dist(t1234[(t1234.variant_number==1) & (t1234.gender==g) &
                 (t1234.active_mins_diff < t1234.active_mins_diff.quantil
e(0.90)) &
                 (t1234.active_mins_diff > t1234.active_mins_diff.quantil
e(0.10))],
            t1234[(t1234.variant_number==0) & (t1234.gender==g) &
                 (t1234.active_mins_diff < t1234.active_mins_diff.quantil
e(0.9)) &
                 (t1234.active_mins_diff > t1234.active_mins_diff.quantil
e(0.1))],
            "active_mins_diff","active_mins_diff","Difference in Total Ac
tive Mins " + '(' + g.title() + ')', "Treatment","Control",
            title='Distribution of Post/Pre-Test Difference in Total Acti
ve Miniutes ' + '(' + g.title() + ')')
```


Distribution of Post/Pre-Test Difference in Total Active Miniutes (Unknown)
Treatment
Control
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
−200
−100
0
100
200
Difference in Total Active Mins (Unknown)


Distribution of Post/Pre-Test Difference in Total Active Miniutes (Female)
Treatment
Control
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
−200
−100
0
100
200
Difference in Total Active Mins (Female)


Distribution of Post/Pre-Test Difference in Total Active Miniutes (Male)
Treatment
Control
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
−200
−100
0
100
200
Difference in Total Active Mins (Male)

User Type

In [664]:
```
stats1234_user_type = t1234.groupby(['variant_number','user_type'])['active_mins_post', 'active_mins_pre', 'active_mins_diff'].agg(['mean', 'count', 'var'])
stats1234_user_type
```

Out[664]:

		active_mins_post			active_mins_pre		
		mean	count	var	mean	count	var
variant_number	user_type						
0	contributor	4309.835346	911	3.489298e+07	4967.092206	911	4.838156e+07
	new_user	29.132720	2283	2.346789e+04	6.105563	2283	8.846519e+01
	non_reader	104.923671	27434	1.121262e+05	108.010352	27434	9.982839e+04
	reader	1537.135378	6685	6.323741e+06	1706.748691	6685	7.733619e+06
1	contributor	4708.031496	127	5.280503e+07	3231.299213	127	2.824765e+07
	new_user	54.623377	770	4.513559e+04	6.487013	770	7.679892e+01
	non_reader	164.454623	7008	1.961370e+05	99.442066	7008	3.998564e+04
	reader	1926.734127	1260	9.683510e+06	1268.926984	1260	3.980188e+06

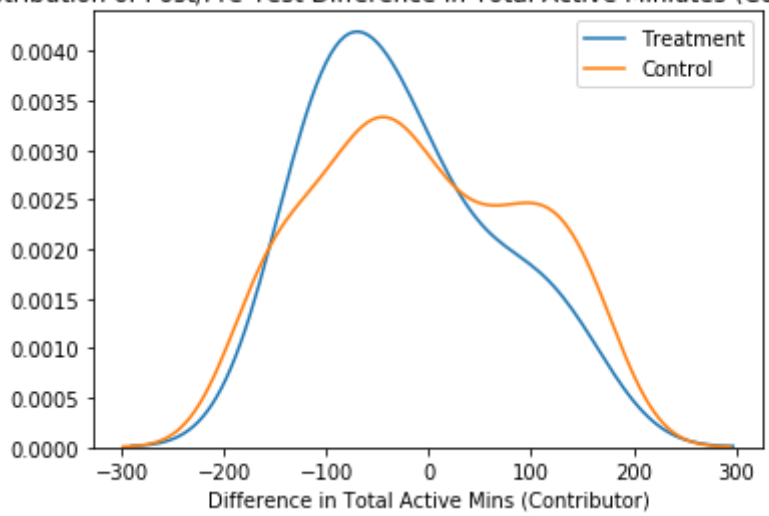
In [716]:
```
for g in set(t4_user_attributes['user_type']):
    intervals[g] = ConfidenceInterval(stats1234_user_type, g)
    print(g, ':', ConfidenceInterval(stats1234_user_type, g))
```

```
contributor : [1284.297301979685, 2983.680986134959]
non_reader : [58.15701884268577, 78.04145683476409]
reader : [710.25891739744, 944.5819950932107]
new_user : [8.967288343457433, 41.2511244308029]
```

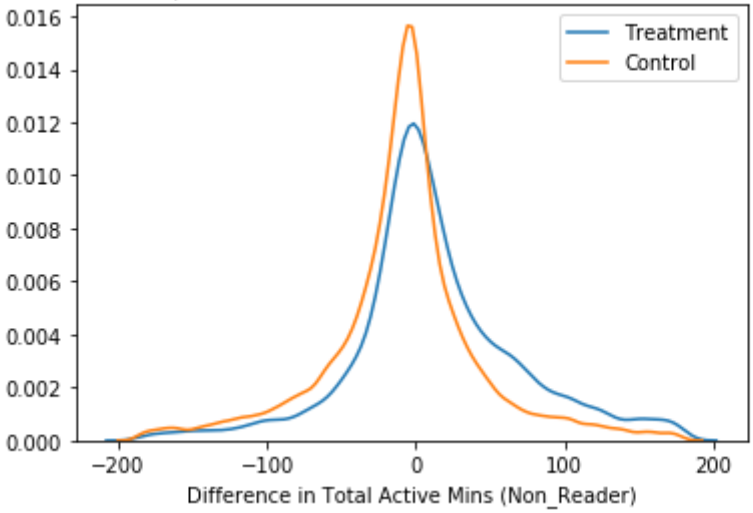
In [666]:

```
for g in set(t4_user_attributes['user_type']):
    build_dist(t1234[(t1234.variant_number==1) & (t1234.user_type==g) &
                (t1234.active_mins_diff < t1234.active_mins_diff.quantil
e(0.9)) &
                (t1234.active_mins_diff > t1234.active_mins_diff.quantil
e(0.10))],
            t1234[(t1234.variant_number==0) & (t1234.user_type==g) &
                (t1234.active_mins_diff < t1234.active_mins_diff.quantil
e(0.9)) &
                (t1234.active_mins_diff > t1234.active_mins_diff.quantil
e(0.1))],
            "active_mins_diff","active_mins_diff","Difference in Total Ac
tive Mins " + '(' + g.title() + ')', "Treatment","Control",
            title='Distribution of Post/Pre-Test Difference in Total Acti
ve Miniutes ' + '(' + g.title() + ')')
```

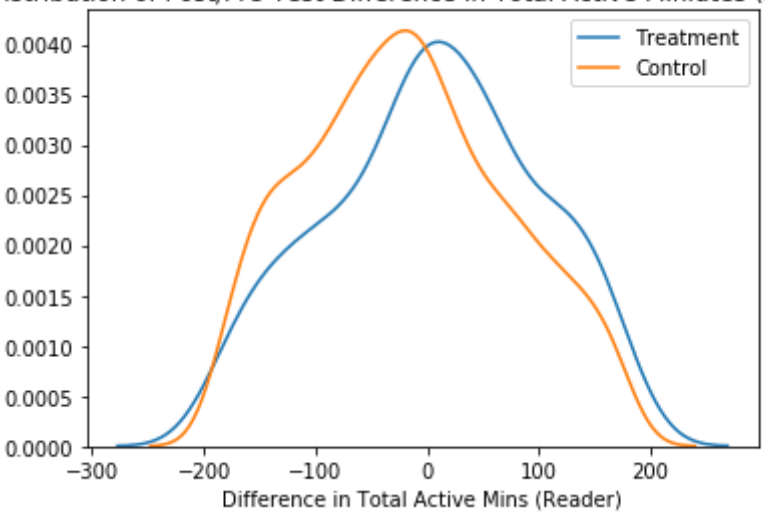
Distribution of Post/Pre-Test Difference in Total Active Miniutes (Contributor)

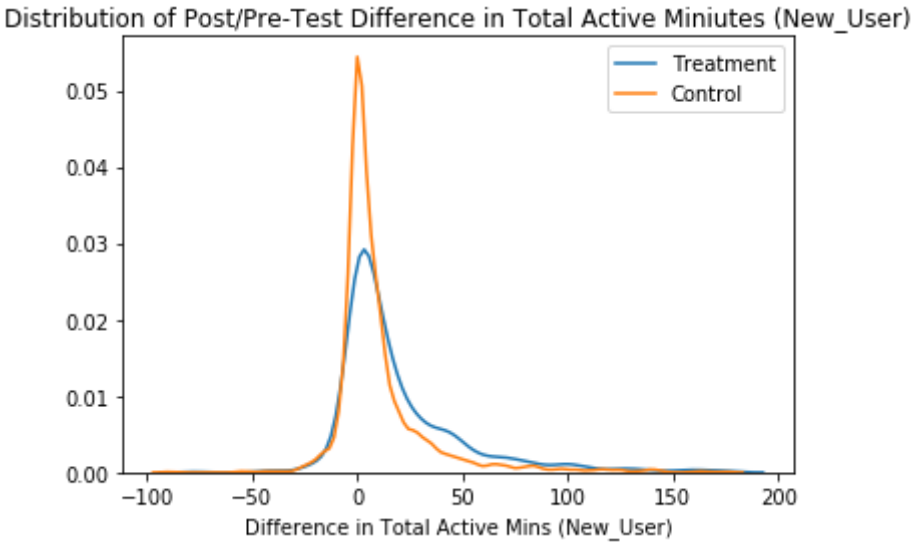


Distribution of Post/Pre-Test Difference in Total Active Miniutes (Non_Reader)



Distribution of Post/Pre-Test Difference in Total Active Miniutes (Reader)





Gender and User Type

In [706]:
```
stats1234 = t1234.groupby(['variant_number','user_type','gender'])['active_mins_post', 'active_mins_pre', 'active_mins_diff'].agg(['mean', 'count', 'var'])
stats1234
```

Out[706]:

			active_mins_post			active_mins_pre		
			mean	count	var	mean	count	var
variant_number	user_type	gender						
0	contributor	female	2891.901345	223	1.919445e+07	3680.219731	223	2.7
		male	4959.790541	592	4.056130e+07	5466.097973	592	5.5
		unknown	3595.520833	96	2.918201e+07	4879.187500	96	4.7
	new_user	female	28.274590	732	2.253973e+04	5.845628	732	5.3
		male	30.543119	1090	2.800025e+04	6.227523	1090	8.6
		unknown	27.160521	461	1.430522e+04	6.229935	461	1.4
	non_reader	female	91.746596	7932	1.876154e+05	97.312532	7932	2.0
		male	115.496277	15175	9.035496e+04	116.640461	15175	5.7
		unknown	92.000462	4327	4.926881e+04	97.354749	4327	5.4
	reader	female	1281.672737	1812	5.308603e+06	1409.699227	1812	6.4
		male	1689.815584	4094	7.074096e+06	1858.350513	4094	8.4
		unknown	1328.952503	779	4.437375e+06	1600.966624	779	6.6
1	contributor	female	3514.038462	26	2.606714e+07	2304.115385	26	5.3
		male	5354.695122	82	6.312565e+07	3830.280488	82	3.9
		unknown	3551.052632	19	4.398827e+07	1915.000000	19	7.1
	new_user	female	50.650000	260	1.585127e+04	6.334615	260	6.6
		male	58.923288	365	6.926217e+04	6.487671	365	8.0
		unknown	50.924138	145	3.735781e+04	6.758621	145	8.6
	non_reader	female	137.462475	1972	1.131500e+05	80.724645	1972	2.4
		male	184.613955	3927	2.676727e+05	108.973262	3927	4.0
		unknown	141.066727	1109	8.735659e+04	98.974752	1109	6.7
	reader	female	1634.485030	334	7.857262e+06	1082.218563	334	3.0
		male	2071.270667	750	1.057237e+07	1342.662667	750	4.2
		unknown	1865.420455	176	9.208636e+06	1309.034091	176	4.7

In [711]:
```
covariates = [[a,b] for a in set(t4_user_attributes['user_type']) for b in set(t4_user_attributes['gender'])]
len(covariates)
```

Out[711]: 12

In [712]:
```
def ConfidenceInterval_covariate(stats, covariate):
    mean_did = stats.loc[1, covariate[0], covariate[1]]['active_mins_dif
f', 'mean'] - stats.loc[0, covariate[0], covariate[1]]['active_mins_dif
f', 'mean']
    SE_did = np.sqrt(stats.loc[1, covariate[0], covariate[1]]['active_mi
ns_diff','var'] / stats.loc[1, covariate[0], covariate[1]]['active_mins_
diff','count'] +
                     stats.loc[0, covariate[0], covariate[1]]['active_mi
ns_diff','var'] / stats.loc[0, covariate[0], covariate[1]]['active_mins_
diff','count'])

    upper = mean_did + (1.96 * SE_did)
    lower = mean_did - (1.96 * SE_did)
    return [lower, upper]
```

In [717]:
```
for c in covariates:
    intervals['_'.join(c)] = ConfidenceInterval_covariate(stats1234, c)
    print(c, ':', ConfidenceInterval_covariate(stats1234, c))
```

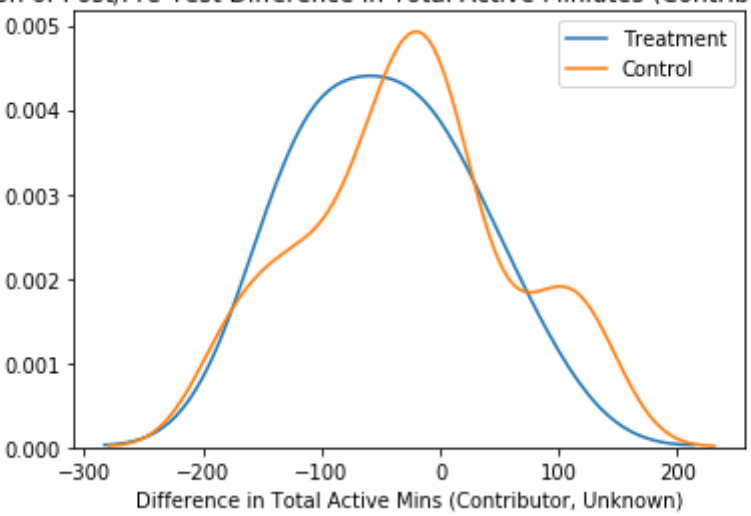
```
['contributor', 'unknown'] : [653.6048080357796, 5185.8337884554485]
['contributor', 'female'] : [343.4674981069095, 3653.0154270396924]
['contributor', 'male'] : [925.4251510097495, 3136.018982147799]
['non_reader', 'unknown'] : [28.39717566566377, 66.49534790818825]
['non_reader', 'female'] : [49.22753265482336, 75.3799974770583]
['non_reader', 'male'] : [61.20339354405016, 92.36636076534171]
['reader', 'unknown'] : [545.8458322158888, 1110.9551363918833]
['reader', 'female'] : [460.6913673558314, 899.8945469083554]
['reader', 'male'] : [739.678278996958, 1054.607579332304]
['new_user', 'unknown'] : [-9.935825860614052, 56.405688976778315]
['new_user', 'female'] : [3.262515103845068, 40.5103306296564]
['new_user', 'male'] : [-0.4068322301028964, 56.646872446264766]
```

In [718]:
```
df_intervals = pd.DataFrame(intervals)
```

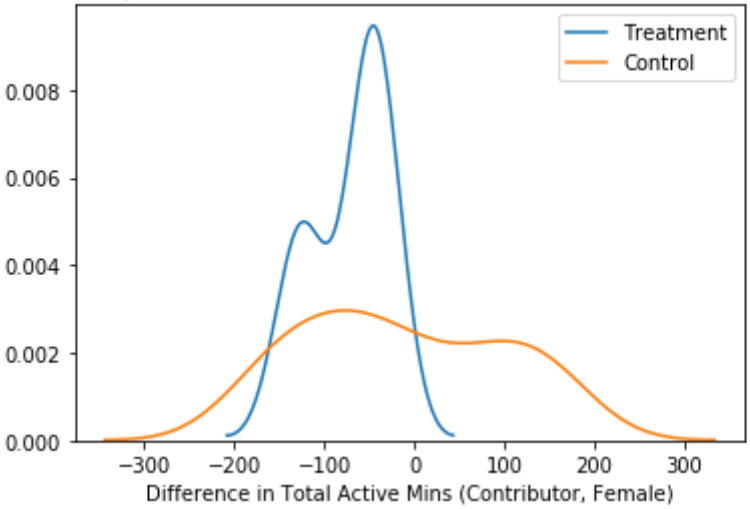
In [769]:

```
for c in covariates:
    build_dist(t1234[(t1234.variant_number==1) & (t1234.user_type==c[0]) & (t1234.gender==c[1]) &
                (t1234.active_mins_diff < t1234.active_mins_diff.quantile(0.90)) &
                (t1234.active_mins_diff > t1234.active_mins_diff.quantile(0.10))],
            t1234[(t1234.variant_number==0) & (t1234.user_type==c[0]) & (t1234.gender==c[1]) &
                (t1234.active_mins_diff < t1234.active_mins_diff.quantile(0.90)) &
                (t1234.active_mins_diff > t1234.active_mins_diff.quantile(0.1))],
            "active_mins_diff","active_mins_diff","Difference in Total Active Mins " + '(' + c[0].title() + ', ' + c[1].title() + ')', "Treatment","Control",
            title='Distribution of Post/Pre-Test Difference in Total Active Miniutes ' + '(' + c[0].title() + ', ' + c[1].title() + ')')
```

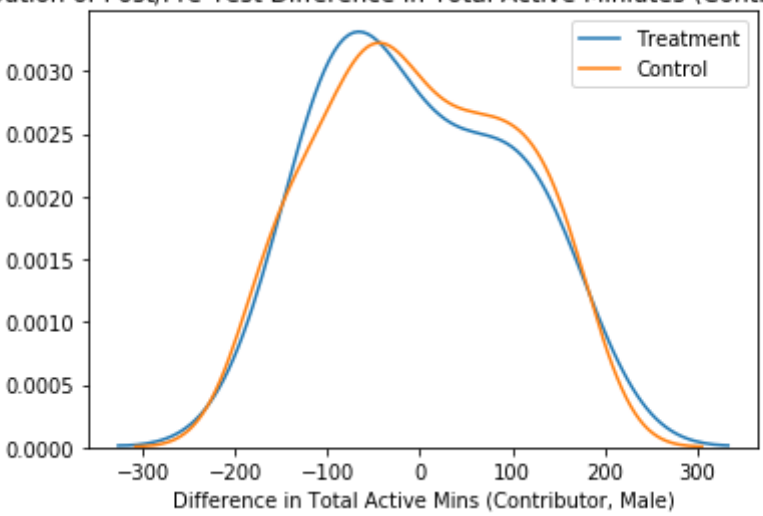
Distribution of Post/Pre-Test Difference in Total Active Miniutes (Contributor, Unknown)



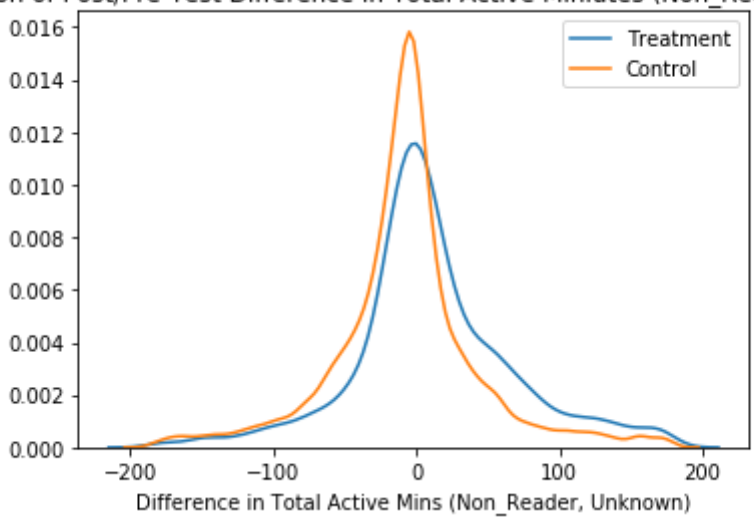
Distribution of Post/Pre-Test Difference in Total Active Miniutes (Contributor, Female)



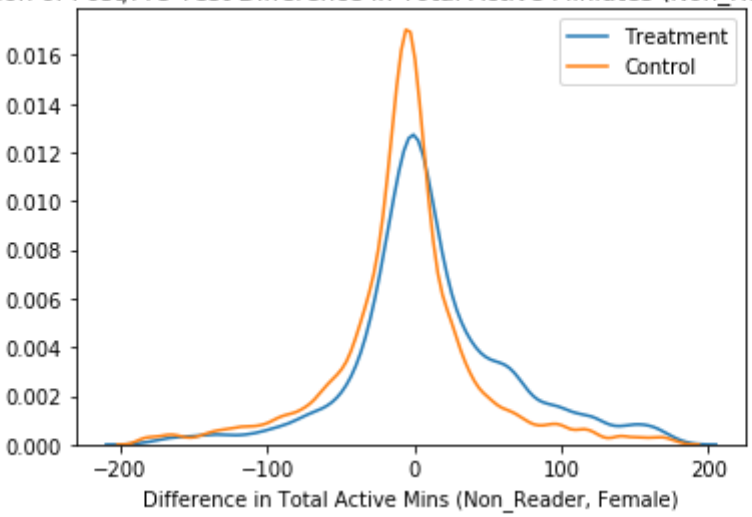
Distribution of Post/Pre-Test Difference in Total Active Miniutes (Contributor, Male)



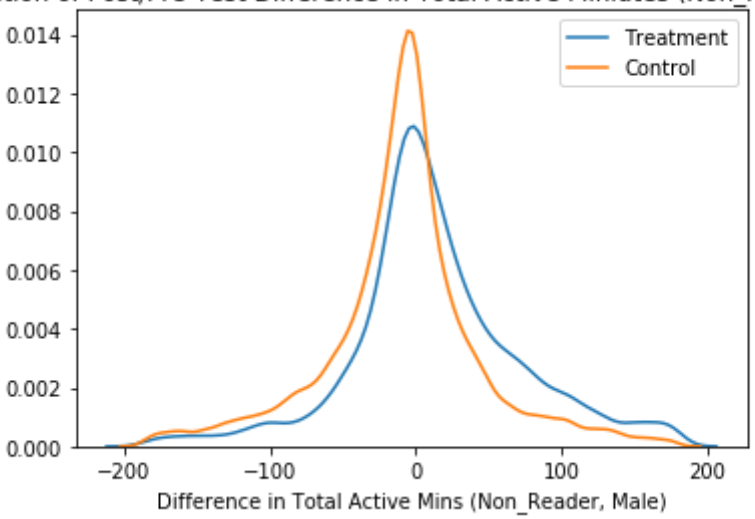
Distribution of Post/Pre-Test Difference in Total Active Miniutes (Non_Reader, Unknown)



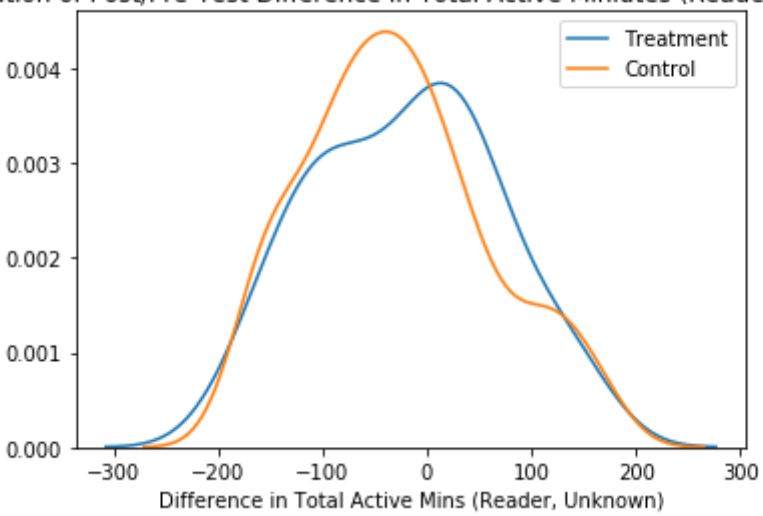
Distribution of Post/Pre-Test Difference in Total Active Miniutes (Non_Reader, Female)



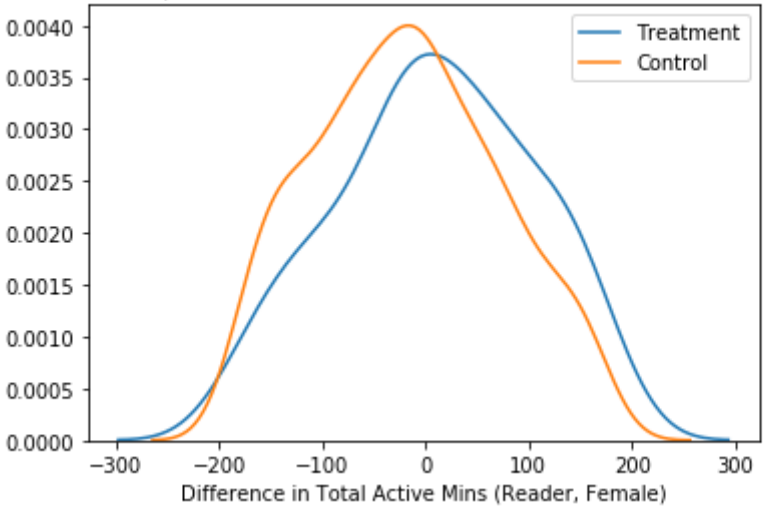
Distribution of Post/Pre-Test Difference in Total Active Miniutes (Non_Reader, Male)



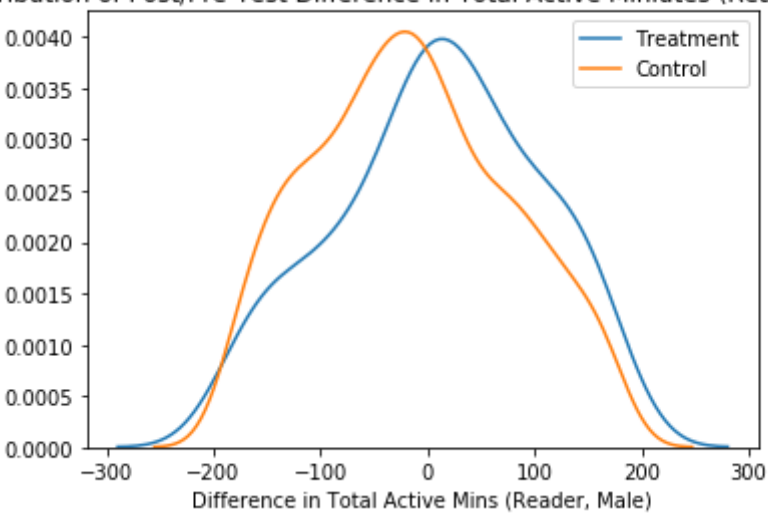
Distribution of Post/Pre-Test Difference in Total Active Miniutes (Reader, Unknown)



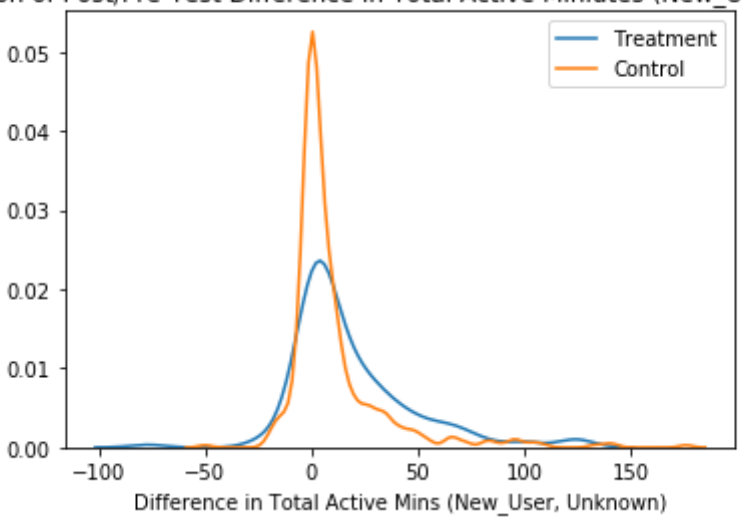
Distribution of Post/Pre-Test Difference in Total Active Miniutes (Reader, Female)



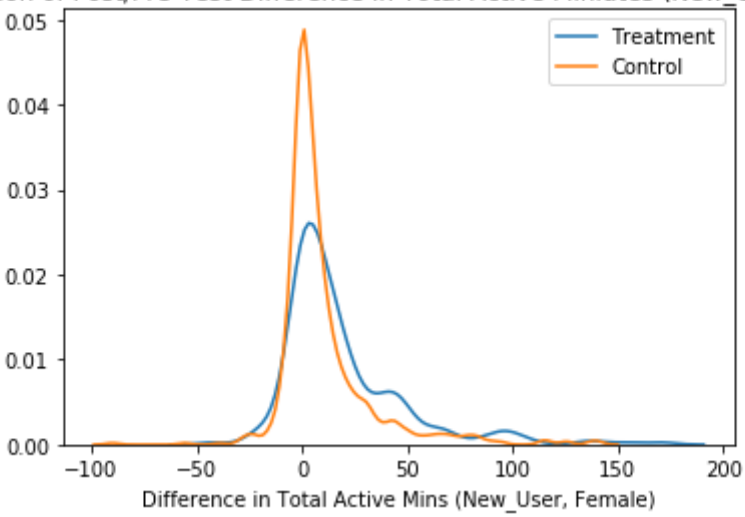
Distribution of Post/Pre-Test Difference in Total Active Miniutes (Reader, Male)



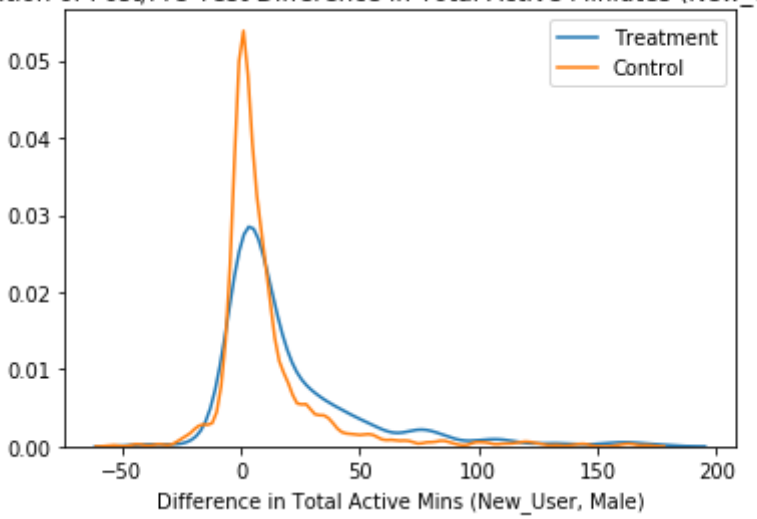
Distribution of Post/Pre-Test Difference in Total Active Miniutes (New_User, Unknown)



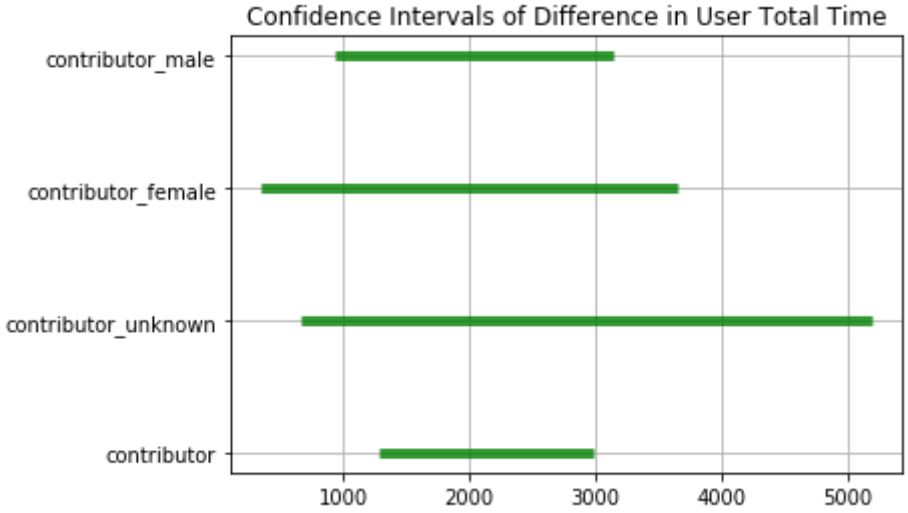
Distribution of Post/Pre-Test Difference in Total Active Miniutes (New_User, Female)

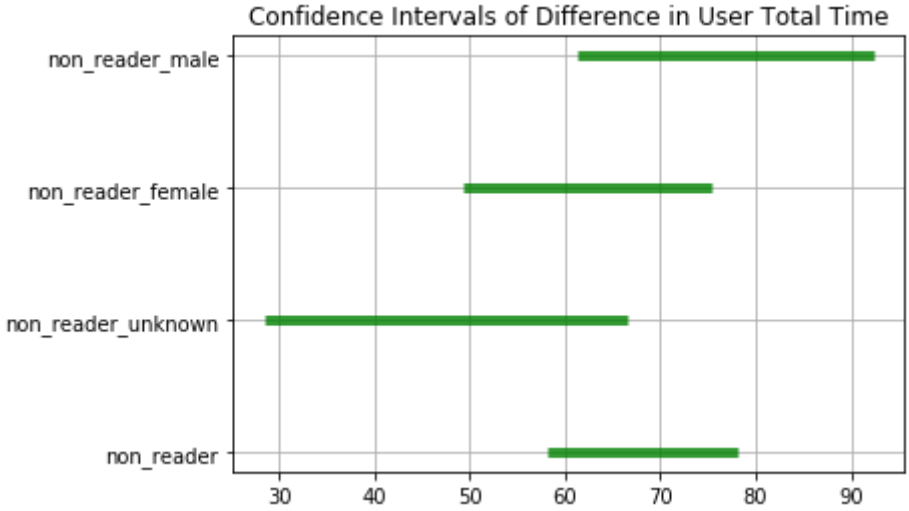


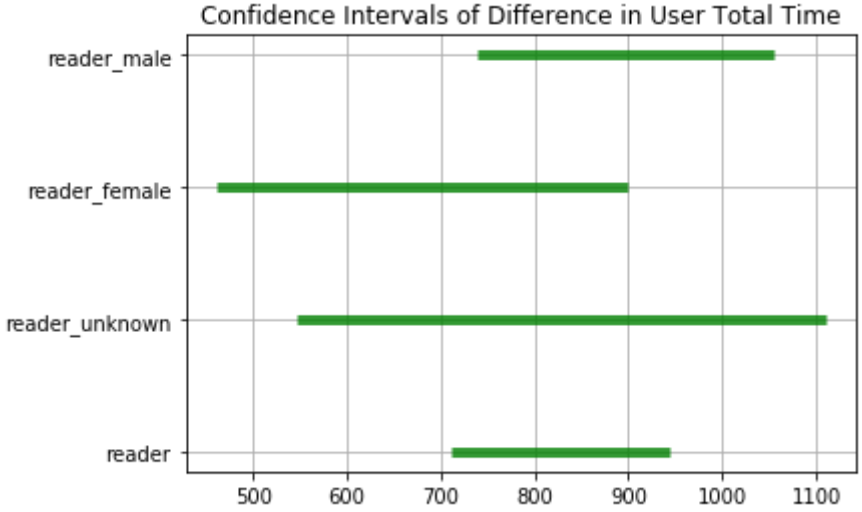
Distribution of Post/Pre-Test Difference in Total Active Miniutes (New_User, Male)

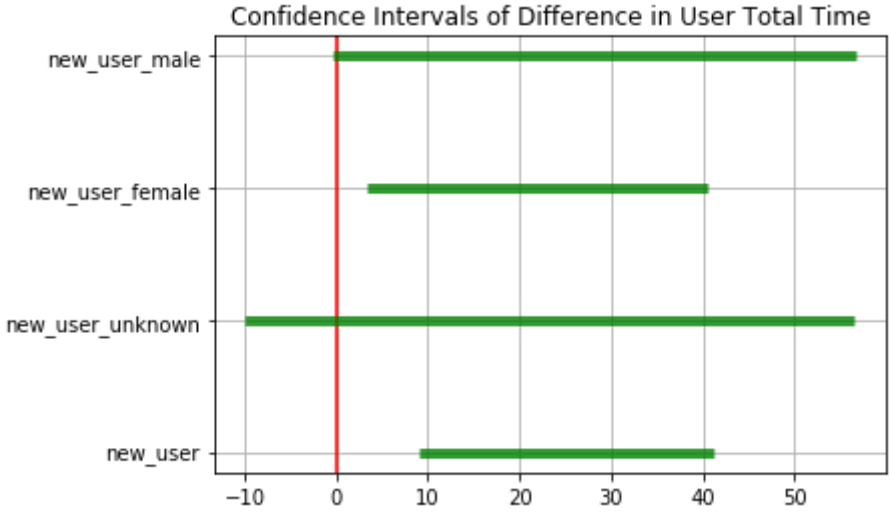


In [771]:
```
for u in set(t4_user_attributes.user_type):
    plt.figure()
    plt.title('Confidence Intervals of Difference in User Total Time')
    plt.grid()
    plt.vlines(0, -1, 4, colors='r')
    for i in df_intervals.columns[df_intervals.columns.str.startswith(u
)]:
        plt.hlines(i, df_intervals[i][0], df_intervals[i][1], colors='g'
, alpha=0.8, linestyles='solid', label=i, linewidth=5.0)
```


Confidence Intervals of Difference in User Total Time
contributor_male
contributor_female
contributor_unknown
contributor
1000
2000
3000
4000
5000


Confidence Intervals of Difference in User Total Time
non_reader_male
non_reader_female
non_reader_unknown
non_reader
30
40
50
60
70
80
90


Confidence Intervals of Difference in User Total Time
reader_male
reader_female
reader_unknown
reader
500
600
700
800
900
1000
1100


Confidence Intervals of Difference in User Total Time
new_user_male
new_user_female
new_user_unknown
new_user
−10
0
10
20
30
40
50

User Average Active Time as Metric

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

In [2]:
```
# reading the data sets
t1_user_active_min     = pd.read_csv("t1_user_active_min.csv")
t2_user_variant        = pd.read_csv("t2_user_variant.csv")
t3_user_active_min_pre = pd.read_csv("t3_user_active_min_pre.csv")
t4_user_attributes     = pd.read_csv("t4_user_attributes.csv")
```

Use table 1 and 2

In [3]:
```
t1_user_active_min.head()
```

Out[3]:

	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

In [4]:
```
t2_user_variant.head()
```

Out[4]:

	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 [5]:
```
# Drop data with active_mins more than (24 hrs * 60 mins) on a single da
te.
t1_user_active_min = t1_user_active_min[t1_user_active_min.active_mins <
= 24*60]
```

In [6]:
```
# Total minutes per user
t1_user_active_min_mean = t1_user_active_min.groupby(by=['uid'], as_inde
x=False).mean()
```

In [21]:
```
# merge t1 and t2 by uid
t12 = pd.merge(t2_user_variant, t1_user_active_min_mean, on='uid')
```

In [22]:
```
t12.head()
```

Out[22]:

	uid	variant_number	dt	signup_date	active_mins
0	0	0	2019-02-06	2018-09-24	3.307692
1	1	0	2019-02-06	2016-11-07	160.052632
2	2	0	2019-02-06	2018-09-17	2.428571
3	3	0	2019-02-06	2018-03-04	3.208333
4	4	0	2019-02-06	2017-03-09	1.950000

In [23]:
```
t12.variant_number.value_counts()
```

Out[23]:
```
0    37425
1     9208
Name: variant_number, dtype: int64
```

Compute confidence interval on difference of means.

In [24]:
```
stats12 = t12.groupby(['variant_number'])['active_mins'].agg(['mean', 'c
ount', 'std','var'])
stats12
```

Out[24]:

variant_number	mean	count	std	var
0	8.761316	37425	18.780503	352.707296
1	11.638743	9208	21.102090	445.298205

In [25]:
```
sigma_diff = np.sqrt(stats12.loc[0]['var'] / stats12.loc[0]['count'] +
                     stats12.loc[1]['var'] / stats12.loc[1]['count'])
upper = (stats12.loc[1]['mean'] - stats12.loc[0]['mean']) + (1.96 * sigm
a_diff)
lower = (stats12.loc[1]['mean'] - stats12.loc[0]['mean']) - (1.96 * sigm
a_diff)
print([lower, upper])
```

```
[2.406274815429466, 3.3485787569999346]
```

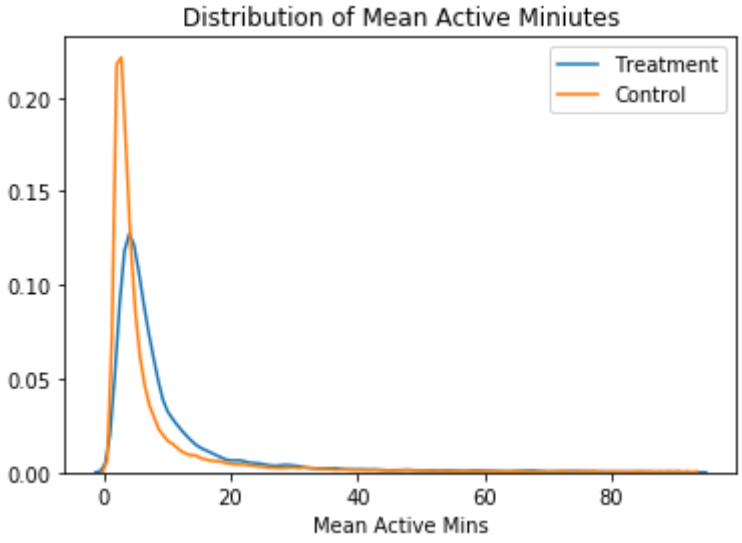
Perform a t-test:

In [26]:
```
stats.ttest_ind(t12.active_mins[t12.variant_number==1], t12.active_mins[
t12.variant_number==0], equal_var =False)
```

Out[26]:
```
Ttest_indResult(statistic=11.970143076301513, pvalue=7.574338206129901e
-33)
```

The low p-value suggests strong evidence for the new UI design to make a positive impact on total active time per user.

In [116]:
```
build_dist(t12[(t12.variant_number==1) &
                (t12.active_mins < t12.active_mins.quantile(0.99)) &
                (t12.active_mins > t12.active_mins.quantile(0.01))],
           t12[(t12.variant_number==0) &
                (t12.active_mins < t12.active_mins.quantile(0.99)) &
                (t12.active_mins > t12.active_mins.quantile(0.01))],
           "active_mins","active_mins","Mean Active Mins", "Treatment",
"Control",
           title='Distribution of Mean Active Miniutes')
```



Add Table 3

In [27]:
```
t3_user_active_min_pre.head()
```

Out[27]:

	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 [28]:
```
# Drop data with active_mins more than (24 hrs * 60 mins) on a single da
te.
t3_user_active_min_pre = t3_user_active_min_pre[t3_user_active_min_pre.a
ctive_mins <= 24*60]
```

In [29]:
```
# Total minutes per user
t3_user_active_min_pre_mean = t3_user_active_min_pre.groupby(by=['uid'],
as_index=False).mean()
```

In [31]:
```
t123 = pd.merge(t12, t3_user_active_min_pre_mean, on='uid', suffixes=('_
post', '_pre'))
```

In [32]:
```
t123.head()
```

Out[32]:

	uid	variant_number	dt	signup_date	active_mins_post	active_mins_pre
0	0	0	2019-02-06	2018-09-24	3.307692	3.333333
1	1	0	2019-02-06	2016-11-07	160.052632	158.330579
2	2	0	2019-02-06	2018-09-17	2.428571	3.700000
3	3	0	2019-02-06	2018-03-04	3.208333	5.684211
4	4	0	2019-02-06	2017-03-09	1.950000	2.357143

In [33]:
```
t123['active_mins_diff'] = t123['active_mins_post'] - t123['active_mins_
pre']
```

In [34]:
```
stats123 = t123.groupby(['variant_number'])['active_mins_post', 'active_
mins_pre', 'active_mins_diff'].agg(['mean', 'count', 'std','var'])
stats123
```

Out[34]:

	active_mins_post				active_mins_pre			
	mean	count	std	var	mean	count	std	var
variant_number								
0	8.778037	37313	18.805182	353.634855	9.114909	37313	18.827314	354.467740
1	11.668744	9165	21.145444	447.129817	7.221025	9165	12.555481	157.640098

Difference in difference

For each user, first obtain difference in post-test user average time and pre-test user average time:
$\Delta x_1 \equiv x_{1,post} - x_{1,pre}$ and $\Delta x_2 \equiv x_{0,post} - x_{0,pre}$

Then obtain the difference in mean: $\overline{\Delta x_1} - \overline{\Delta x_2}$

Standard Error: $\sqrt{\frac{s^2_{\Delta x_1}}{n_1} + \frac{s^2_{\Delta x_2}}{n_1}}$

In [35]:
```
mean_did = stats123.loc[1,'active_mins_diff']['mean'] - stats123.loc[0,
'active_mins_diff']['mean']

sigma_did = np.sqrt(stats123.loc[1,'active_mins_diff']['var'] / stats123
.loc[1,'active_mins_diff']['count'] +
                    stats123.loc[0,'active_mins_diff']['var'] / stats123
.loc[0,'active_mins_diff']['count'])

upper = mean_did + (1.96 * sigma_did)
lower = mean_did - (1.96 * sigma_did)
print([lower, upper])
```

```
[4.499019292447001, 5.070162945220937]
```

Perform a t-test:

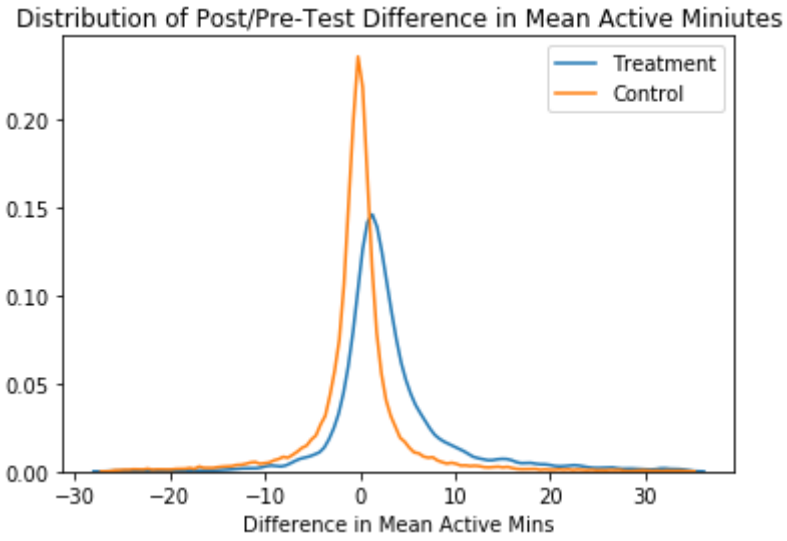
In [36]:
```
stats.ttest_ind(t123.active_mins_diff[t123.variant_number==1], t123.acti
ve_mins_diff[t123.variant_number==0], equal_var =False)
```

Out[36]:
```
Ttest_indResult(statistic=32.838668686479814, pvalue=2.6124834427323526
e-226)
```

The low p-value suggests high evidence for the new UI design to make a positive impact on total active time per user.

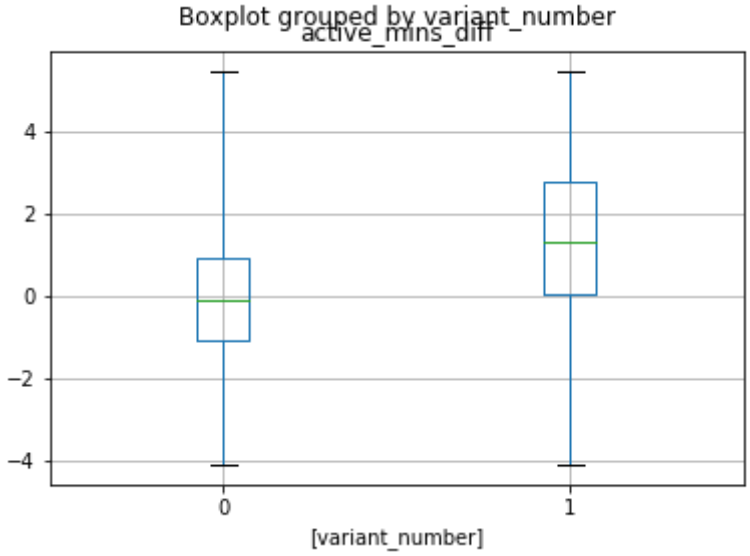
In [37]:
```
def build_dist(df1, df2, col1, col2, xlabel, legend1, legend2, title):
    sns.distplot(df1[col1], hist=False, label = legend1, axlabel = xlabe
l)
    sns.distplot(df2[col2], hist=False, label = legend2, axlabel = xlabe
l)
    plt.title(title)
    #plt.xlim([-1000,1000])
    plt.show()
```

In [111]:
```
build_dist(t123[(t123.variant_number==1) &
                (t123.active_mins_diff < t123.active_mins_diff.quantile(
0.99)) &
                (t123.active_mins_diff > t123.active_mins_diff.quantile(
0.01))],
           t123[(t123.variant_number==0) &
                (t123.active_mins_diff < t123.active_mins_diff.quantile(
0.99)) &
                (t123.active_mins_diff > t123.active_mins_diff.quantile(
0.01))],
           "active_mins_diff","active_mins_diff","Difference in Mean Act
ive Mins", "Treatment","Control",
           title='Distribution of Post/Pre-Test Difference in Mean Activ
e Miniutes')
```



In [49]:
```
t123[(t123.active_mins_diff < t123.active_mins_diff.quantile(0.9)) &
     (t123.active_mins_diff > t123.active_mins_diff.quantile(0.1))][['variant_number','active_mins_diff']].boxplot(by=['variant_number'],whis='range')
```

Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x134035d30>



Add Table 4

In [57]:
```
t4_user_attributes.head()
```

Out[57]:

	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 [58]:
```
t1234 = pd.merge(t4_user_attributes, t123, on='uid')
```

In [59]:
```
t1234.head()
```

Out[59]:

	uid	gender	user_type	variant_number	dt	signup_date	active_mins_post	active_mins_pr
0	0	male	non_reader	0	2019-02-06	2018-09-24	3.307692	3.33333
1	1	male	reader	0	2019-02-06	2016-11-07	160.052632	158.33057
2	2	male	non_reader	0	2019-02-06	2018-09-17	2.428571	3.70000
3	3	male	non_reader	0	2019-02-06	2018-03-04	3.208333	5.68421
4	4	male	non_reader	0	2019-02-06	2017-03-09	1.950000	2.35714

In [60]:
```
pd.crosstab(t1234['variant_number'],t1234['gender']).apply(lambda r: r/r.sum(), axis=1)
```

Out[60]:

gender	female	male	unknown
variant_number			
0	0.286737	0.561493	0.151770
1	0.282815	0.559083	0.158101

In [61]:
```
pd.crosstab(t1234['variant_number'],t1234['user_type']).apply(lambda r: r/r.sum(), axis=1)
```

Out[61]:

user_type	contributor	new_user	non_reader	reader
variant_number				
0	0.024415	0.061185	0.735240	0.17916
1	0.013857	0.084015	0.764648	0.13748

In [62]:
```
pd.crosstab(t1234['gender'],t1234['user_type']).apply(lambda r: r/r.sum(), axis=1)
```

Out[62]:

user_type	contributor	new_user	non_reader	reader
gender				
female	0.018734	0.074637	0.745166	0.161463
male	0.025849	0.055801	0.732579	0.185772
unknown	0.016170	0.085208	0.764342	0.134280

Gender

In [63]:
```
stats1234_gender = t1234.groupby(['variant_number','gender'])['active_mi
ns_post', 'active_mins_pre', 'active_mins_diff'].agg(['mean', 'count',
'var'])
stats1234_gender
```

Out[63]:

		active_mins_post			active_mins_pre			active_min
		mean	count	var	mean	count	var	mean
variant_number	gender							
0	female	7.642952	10699	234.200195	8.213522	10699	265.149126	-0.570570
	male	9.704124	20951	424.230905	9.872825	20951	414.573699	-0.168701
	unknown	7.496357	5663	310.960230	8.013876	5663	296.083171	-0.517519
1	female	10.590656	2592	325.528234	6.584398	2592	97.235650	4.006258
	male	12.397131	5124	516.836077	7.553167	5124	187.825623	4.843964
	unknown	11.021509	1449	414.340209	7.185305	1449	158.030374	3.836204

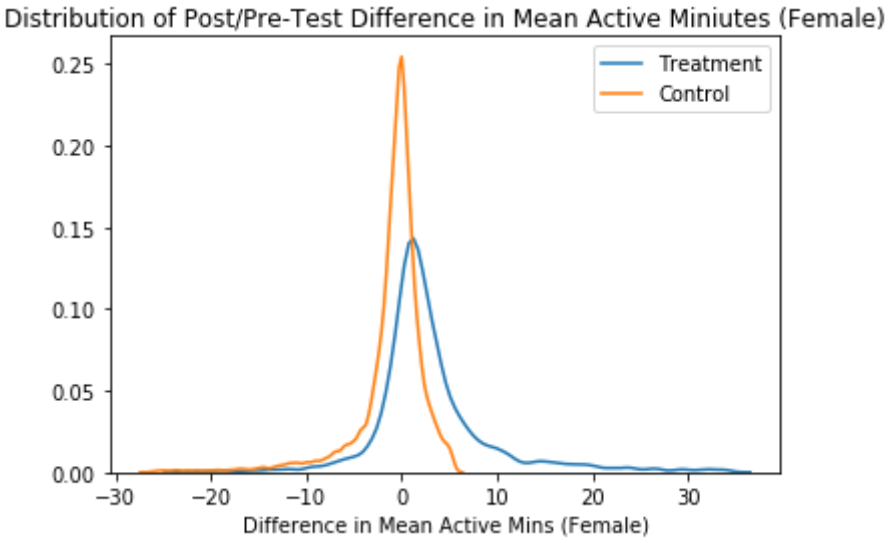
In [64]:
```
def ConfidenceInterval(stats, attribute):
    mean_did = stats.loc[1, attribute]['active_mins_diff', 'mean'] - sta
ts.loc[0,attribute]['active_mins_diff', 'mean']
    SE_did = np.sqrt(stats.loc[1, attribute]['active_mins_diff','var'] /
stats.loc[1, attribute]['active_mins_diff','count'] +
                     stats.loc[0, attribute]['active_mins_diff','var'] /
stats.loc[0, attribute]['active_mins_diff','count'])

    upper = mean_did + (1.96 * SE_did)
    lower = mean_did - (1.96 * SE_did)
    return [lower, upper]
```

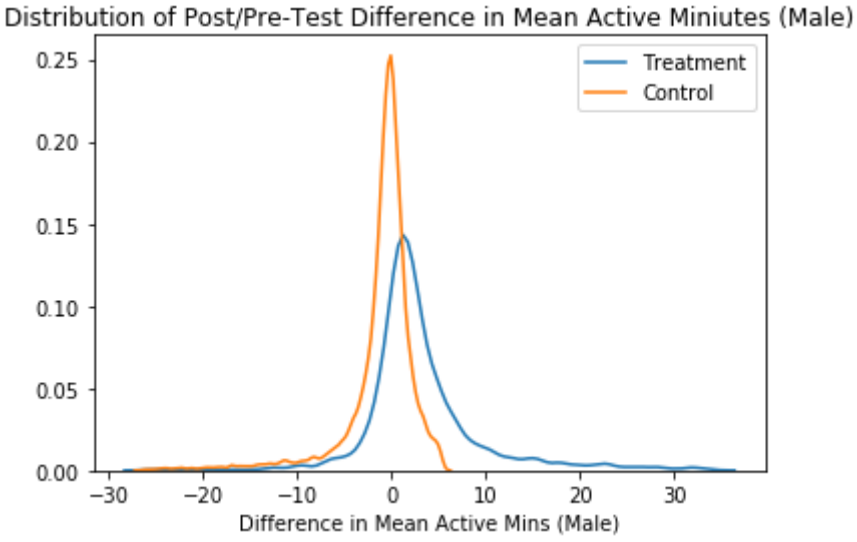
In [65]:
```
intervals = {}
for g in set(t4_user_attributes['gender']):
    intervals[g] = ConfidenceInterval(stats1234_gender, g)
    print(g, ':', ConfidenceInterval(stats1234_gender, g))
```

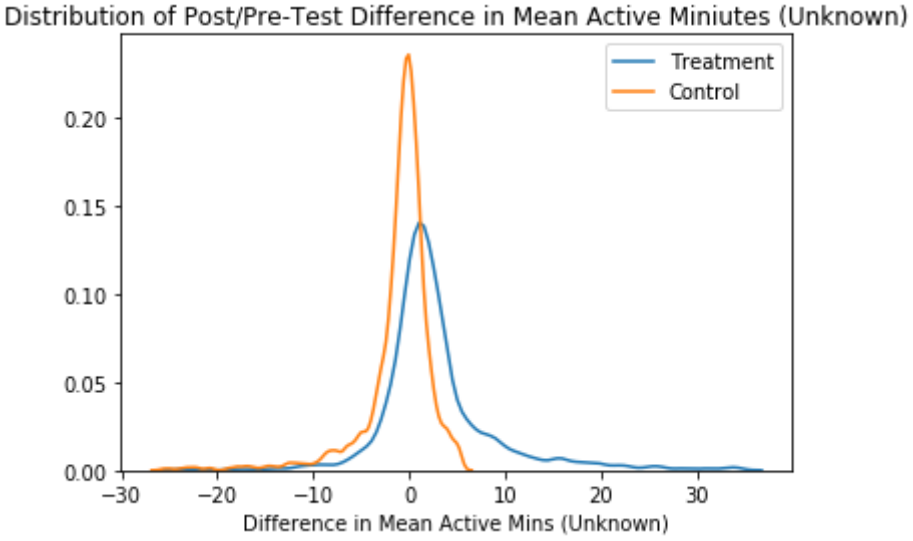
```
female : [4.082668281657273, 5.070986716879877]
male : [4.609718461419328, 5.415611021626094]
unknown : [3.6834453016973905, 5.023999837578068]
```

In [112]:

```
for g in set(t4_user_attributes['gender']):
    build_dist(t1234[(t1234.variant_number==1) & (t1234.gender==g) &
                  (t1234.active_mins_diff < t1234.active_mins_diff.quantil
e(0.99)) &
                  (t1234.active_mins_diff > t1234.active_mins_diff.quantil
e(0.01))],
               t1234[(t1234.variant_number==0) & (t1234.gender==g) &
                  (t1234.active_mins_diff < t1234.active_mins_diff.quantil
e(0.90)) &
                  (t1234.active_mins_diff > t1234.active_mins_diff.quantil
e(0.01))],
               "active_mins_diff","active_mins_diff","Difference in Mean Act
ive Mins " + '(' + g.title() + ')', "Treatment","Control",
               title='Distribution of Post/Pre-Test Difference in Mean Activ
e Miniutes ' + '(' + g.title() + ')')
```


Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Female)
Treatment
Control
0.25
0.20
0.15
0.10
0.05
0.00
−30
−20
−10
0
10
20
30
Difference in Mean Active Mins (Female)


Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Male)
Treatment
Control
0.25
0.20
0.15
0.10
0.05
0.00
−30
−20
−10
0
10
20
30
Difference in Mean Active Mins (Male)


Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Unknown)
Treatment
Control
0.20
0.15
0.10
0.05
0.00
−30
−20
−10
0
10
20
30
Difference in Mean Active Mins (Unknown)

User Type

In [68]:
```
stats1234_user_type = t1234.groupby(['variant_number','user_type'])['active_mins_post', 'active_mins_pre', 'active_mins_diff'].agg(['mean', 'count', 'var'])
stats1234_user_type
```

Out[68]:

		active_mins_post			active_mins_pre			active
		mean	count	var	mean	count	var	mean
variant_number	user_type							
0	contributor	53.160145	911	4151.228369	56.656796	911	4184.622372	-3.49
	new_user	3.441046	2283	22.749363	4.483866	2283	32.671914	-1.04
	non_reader	4.447460	27434	29.756430	4.555002	27434	25.243865	-0.10
	reader	22.324380	6685	740.317113	22.930689	6685	702.626548	-0.60
1	contributor	65.306837	127	6568.871801	44.492685	127	2990.703181	20.81
	new_user	6.242814	770	54.127892	4.698840	770	27.781261	1.54
	non_reader	7.508955	7008	68.836087	4.610943	7008	16.323160	2.89
	reader	32.714566	1260	1333.164449	19.522641	1260	406.913170	13.19

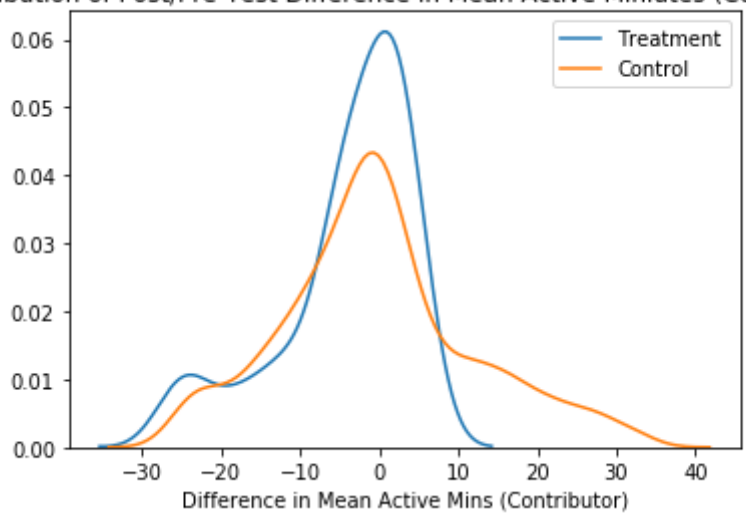
In [69]:
```
for g in set(t4_user_attributes['user_type']):
    intervals[g] = ConfidenceInterval(stats1234_user_type, g)
    print(g, ':', ConfidenceInterval(stats1234_user_type, g))
```

```
contributor : [16.995995749451346, 31.625610810993443]
non_reader : [2.8231488470903954, 3.1879584027166366]
reader : [12.376071976734565, 15.220396987874775]
new_user : [1.9489593821827849, 3.2246293485292896]
```

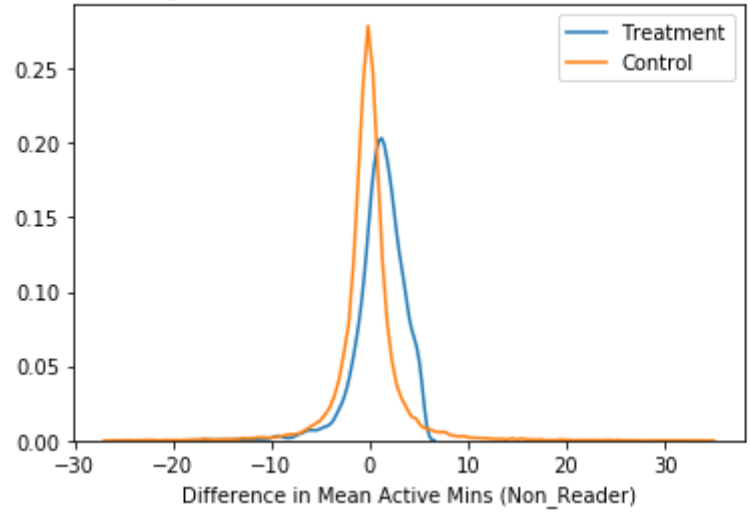
In [113]:

```
for g in set(t4_user_attributes['user_type']):
    build_dist(t1234[(t1234.variant_number==1) & (t1234.user_type==g) &
                  (t1234.active_mins_diff < t1234.active_mins_diff.quantil
e(0.90)) &
                  (t1234.active_mins_diff > t1234.active_mins_diff.quantil
e(0.01))],
               t1234[(t1234.variant_number==0) & (t1234.user_type==g) &
                  (t1234.active_mins_diff < t1234.active_mins_diff.quantil
e(0.99)) &
                  (t1234.active_mins_diff > t1234.active_mins_diff.quantil
e(0.01))],
               "active_mins_diff","active_mins_diff","Difference in Mean Act
ive Mins " + '(' + g.title() + ')', "Treatment","Control",
               title='Distribution of Post/Pre-Test Difference in Mean Activ
e Miniutes ' + '(' + g.title() + ')')
```

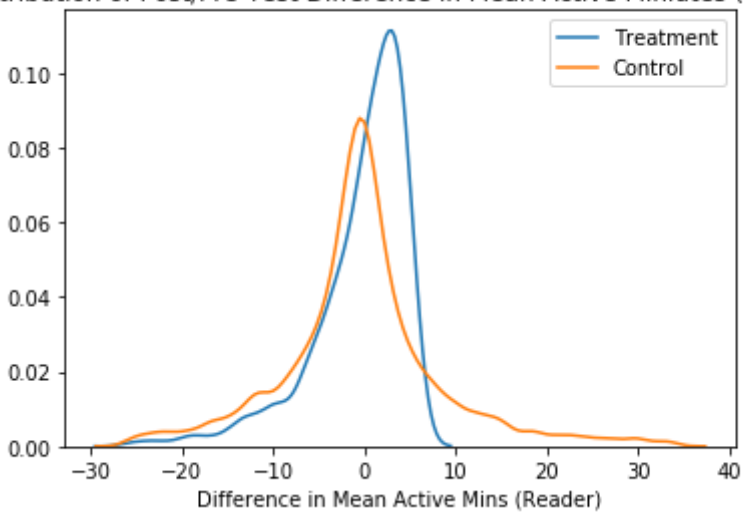
Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Contributor)

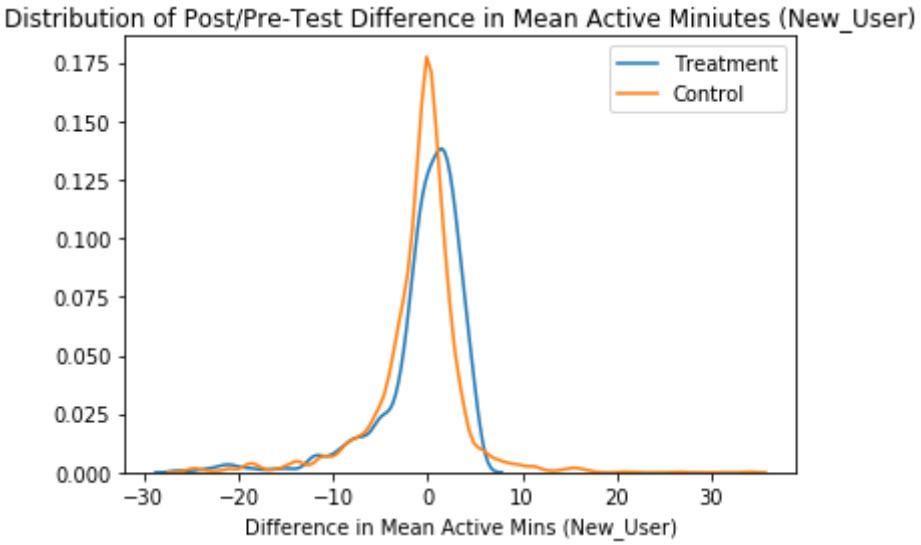


Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Non_Reader)



Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Reader)





Gender and User Type

In [72]:
```
stats1234 = t1234.groupby(['variant_number','user_type','gender'])['active_mins_post', 'active_mins_pre', 'active_mins_diff'].agg(['mean', 'count', 'var'])
stats1234
```

Out[72]:

			active_mins_post			active_mins_pre		
			mean	count	var	mean	count	var
variant_number	user_type	gender						
0	contributor	female	38.714459	223	2066.235264	46.924005	223	2498.233
		male	58.017646	592	4452.855001	59.510130	592	4642.289
		unknown	56.761676	96	6584.499784	61.669696	96	5067.883
	new_user	female	3.306261	732	12.007781	4.266963	732	23.383
		male	3.476038	1090	17.301803	4.591055	1090	31.122
		unknown	3.572328	461	52.765472	4.574837	461	51.131
	non_reader	female	4.353322	7932	41.899578	4.525700	7932	40.190
		male	4.545964	15175	27.172688	4.592553	15175	20.056
		unknown	4.274572	4327	16.490376	4.477025	4327	16.035
	reader	female	19.971238	1812	615.897040	21.187164	1812	655.815
		male	23.495545	4094	786.285890	23.673501	4094	711.371
		unknown	21.642918	779	769.423072	23.082419	779	757.384
1	contributor	female	53.337951	26	3460.047432	34.521595	26	892.456
		male	72.437659	82	7224.506929	49.235937	82	3690.044
		unknown	50.910186	19	8008.855289	37.666457	19	2794.911
	new_user	female	6.033502	260	33.294186	4.900733	260	30.799
		male	6.098764	365	38.696240	4.404795	365	22.607
		unknown	6.980743	145	130.679167	5.077011	145	35.380
	non_reader	female	7.242834	1972	60.681066	4.389482	1972	14.319
		male	7.719148	3927	76.425539	4.685607	3927	16.264
		unknown	7.237863	1109	56.219090	4.740352	1109	20.001
	reader	female	30.576685	334	1262.135957	18.679489	334	342.441
		male	33.391819	750	1364.267695	19.542601	750	407.295
		unknown	33.885655	176	1338.367584	21.037655	176	528.938

In [73]:
```
covariates = [[a,b] for a in set(t4_user_attributes['user_type']) for b in set(t4_user_attributes['gender'])]
len(covariates)
```

Out[73]: 12

In [74]:
```
def ConfidenceInterval_covariate(stats, covariate):
    mean_did = stats.loc[1, covariate[0], covariate[1]]['active_mins_dif
f', 'mean'] - stats.loc[0, covariate[0], covariate[1]]['active_mins_dif
f', 'mean']
    SE_did = np.sqrt(stats.loc[1, covariate[0], covariate[1]]['active_mi
ns_diff','var'] / stats.loc[1, covariate[0], covariate[1]]['active_mins_
diff','count'] +
                     stats.loc[0, covariate[0], covariate[1]]['active_mi
ns_diff','var'] / stats.loc[0, covariate[0], covariate[1]]['active_mins_
diff','count'])

    upper = mean_did + (1.96 * SE_did)
    lower = mean_did - (1.96 * SE_did)
    return [lower, upper]
```

In [75]:
```
for c in covariates:
    intervals['_'.join(c)] = ConfidenceInterval_covariate(stats1234, c)
    print(c, ':', ConfidenceInterval_covariate(stats1234, c))
```

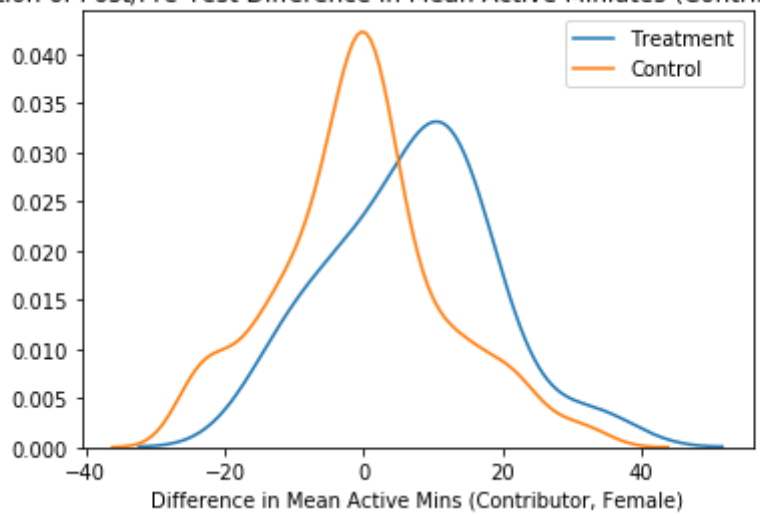
```
['contributor', 'female'] : [12.130744823642814, 41.92105855818548]
['contributor', 'male'] : [15.730777970918131, 33.65763474572131]
['contributor', 'unknown'] : [-5.206375673663516, 41.50987221834224]
['non_reader', 'female'] : [2.700530875954784, 3.350930907232195]
['non_reader', 'male'] : [2.8255329034443393, 3.334727150265173]
['non_reader', 'unknown'] : [2.27444359904215, 3.1254824622100603]
['reader', 'female'] : [10.39133884786024, 15.834906003385534]
['reader', 'male'] : [12.112992512053513, 15.941355559229901]
['reader', 'unknown'] : [11.037669873335581, 17.537331735804244]
['new_user', 'female'] : [1.0973757799120127, 3.0895669986213443]
['new_user', 'male'] : [2.033044325735124, 3.584928320328269]
['new_user', 'unknown'] : [0.7886594445558002, 5.023821684977663]
```

In [76]:
```
df_intervals = pd.DataFrame(intervals)
```

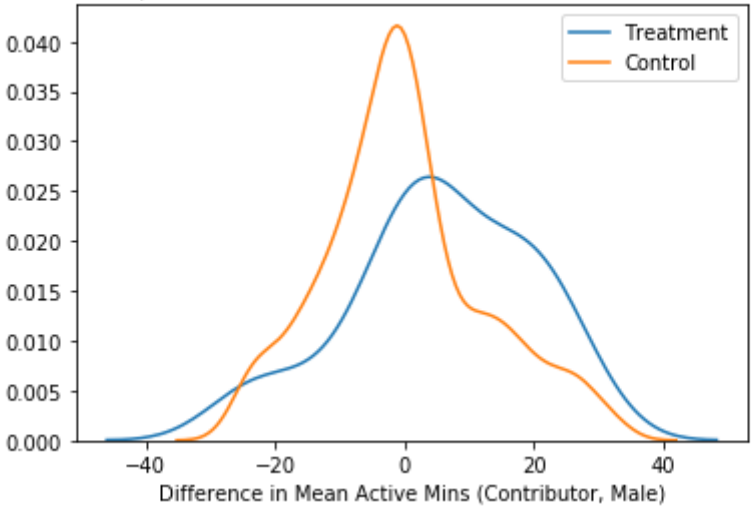
In [114]:

```
for c in covariates:
    build_dist(t1234[(t1234.variant_number==1) & (t1234.user_type==c[0])
& (t1234.gender==c[1]) &
                (t1234.active_mins_diff < t1234.active_mins_diff.quantil
e(0.99)) &
                (t1234.active_mins_diff > t1234.active_mins_diff.quantil
e(0.01))],
           t1234[(t1234.variant_number==0) & (t1234.user_type==c[0]) & (
t1234.gender==c[1]) &
                (t1234.active_mins_diff < t1234.active_mins_diff.quantil
e(0.99)) &
                (t1234.active_mins_diff > t1234.active_mins_diff.quantil
e(0.01))],
           "active_mins_diff","active_mins_diff","Difference in Mean Act
ive Mins " + '(' + c[0].title() + ', ' + c[1].title() + ')', "Treatment"
,"Control",
           title='Distribution of Post/Pre-Test Difference in Mean Activ
e Miniutes ' + '(' + c[0].title() + ', ' + c[1].title() + ')')
```

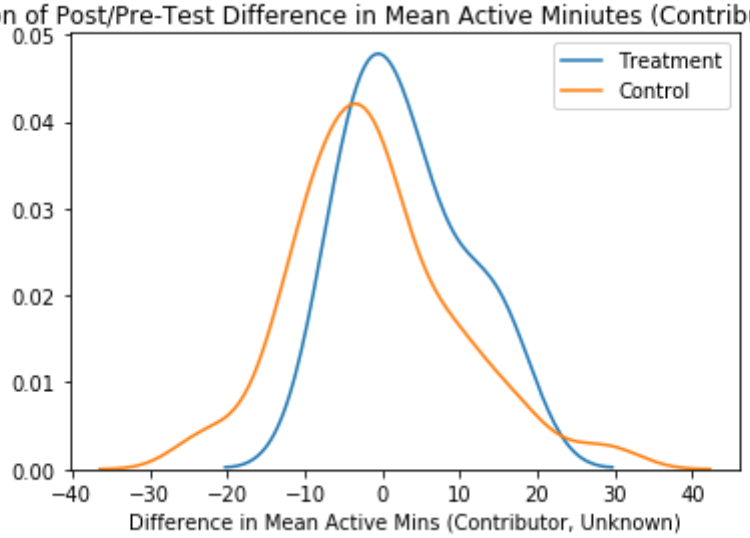
Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Contributor, Female)



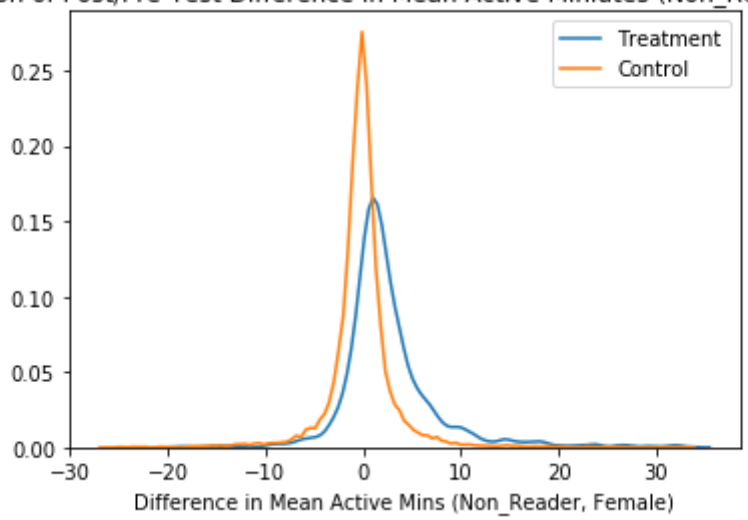
Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Contributor, Male)



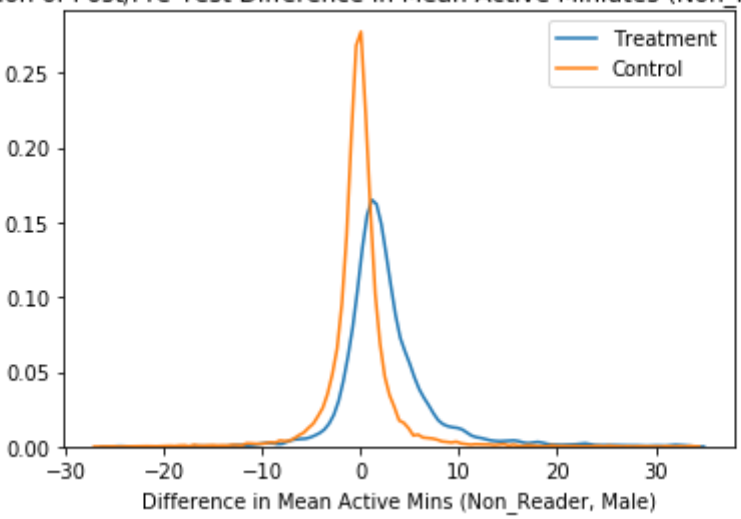
Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Contributor, Unknown)



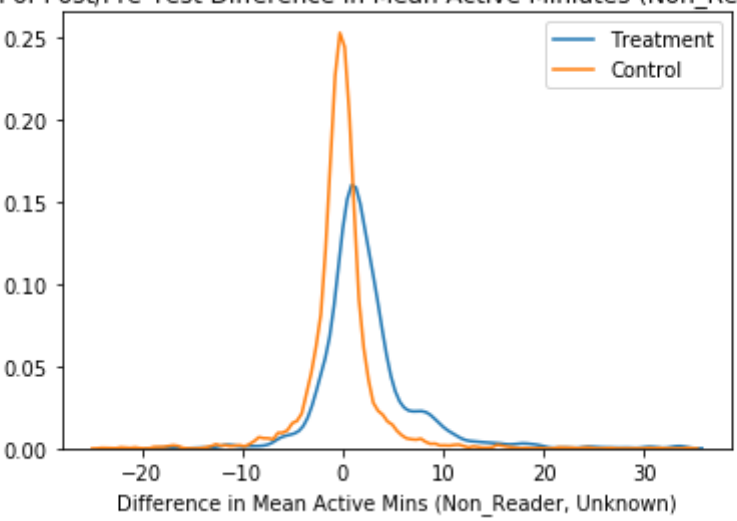
Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Non_Reader, Female)



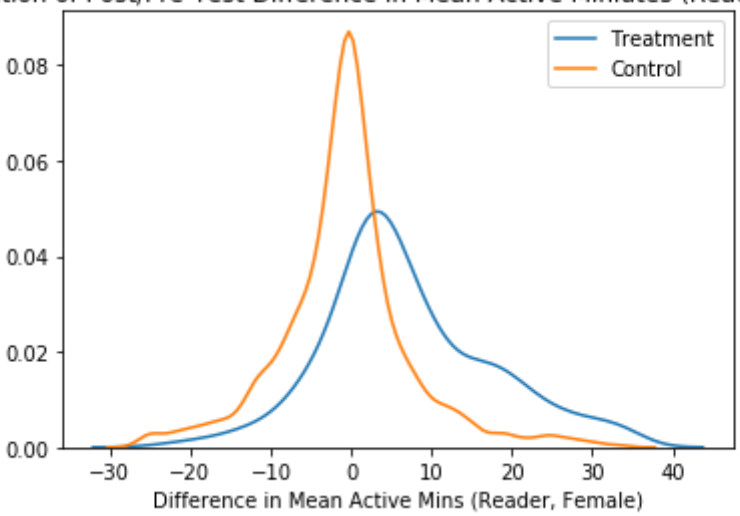
Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Non_Reader, Male)



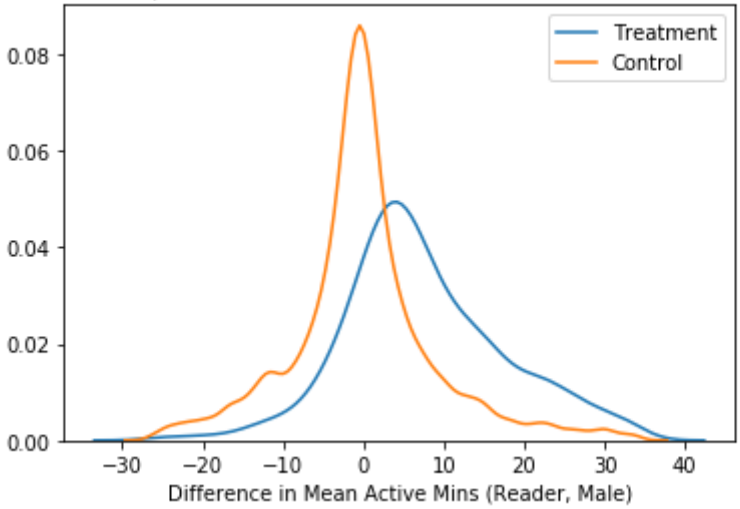
Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Non_Reader, Unknown)



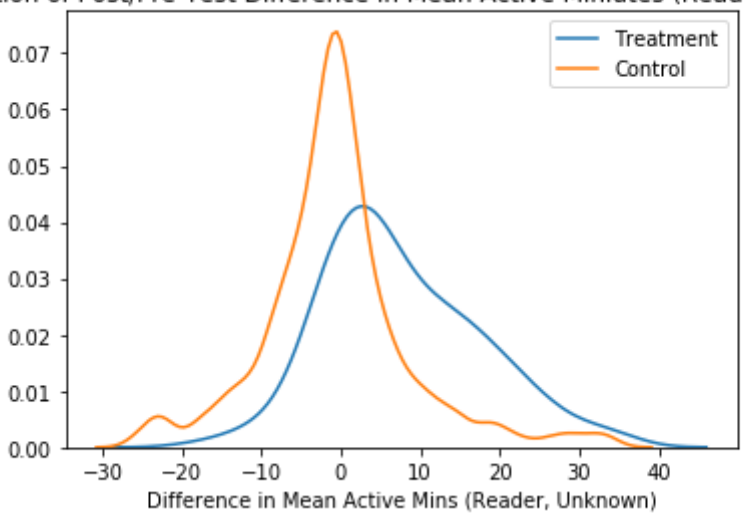
Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Reader, Female)



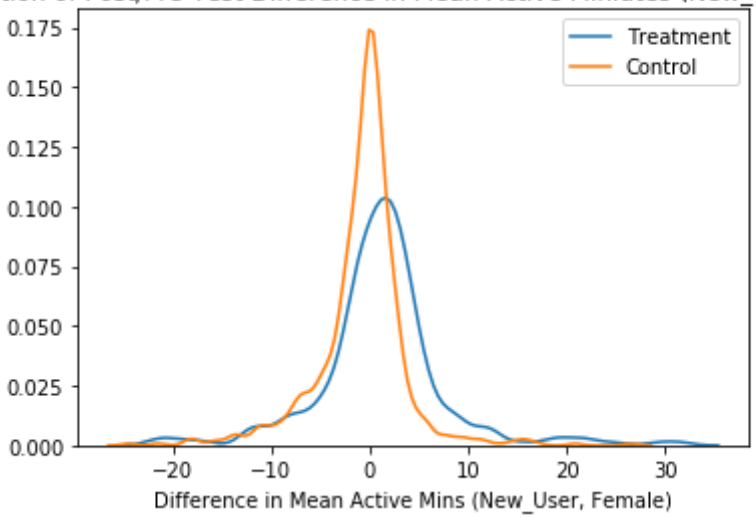
Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Reader, Male)



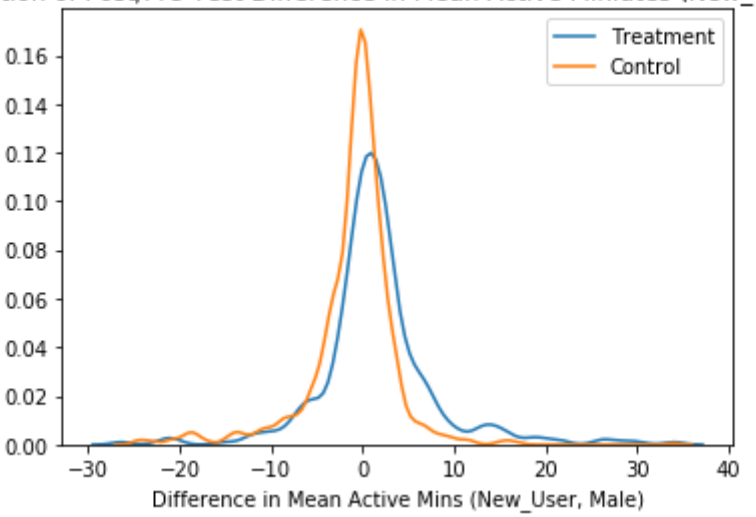
Distribution of Post/Pre-Test Difference in Mean Active Miniutes (Reader, Unknown)



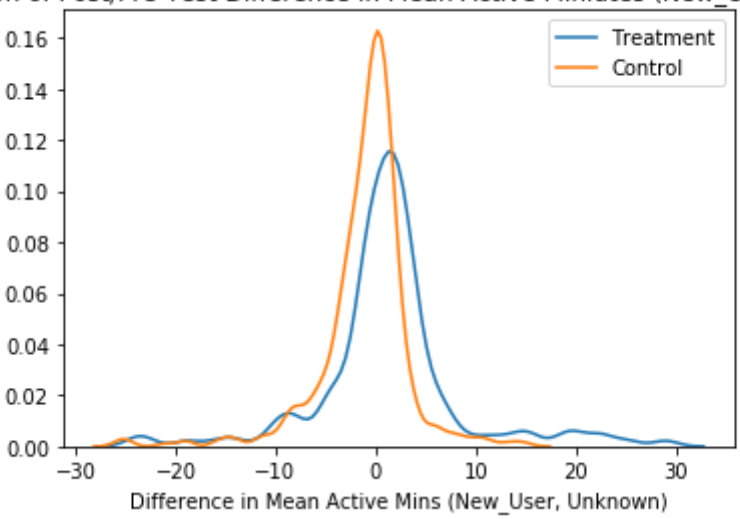
Distribution of Post/Pre-Test Difference in Mean Active Miniutes (New_User, Female)



Distribution of Post/Pre-Test Difference in Mean Active Miniutes (New_User, Male)

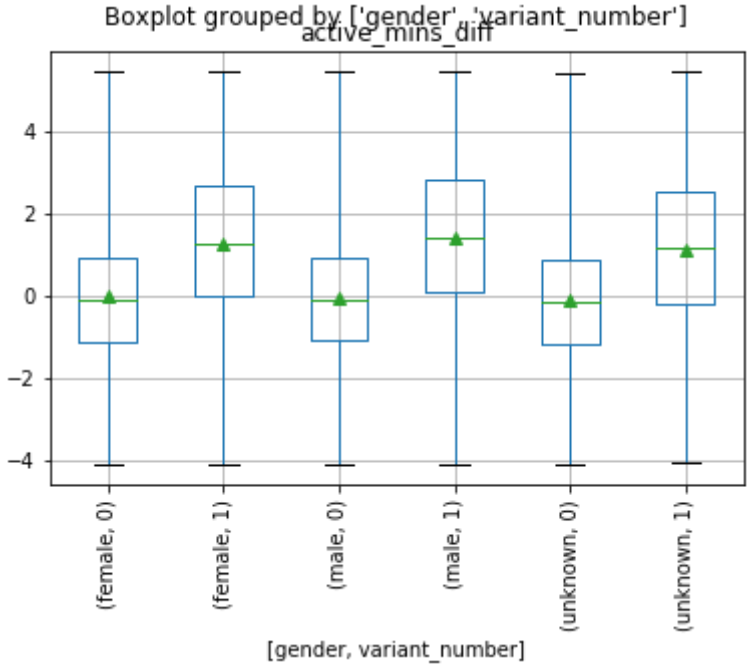


Distribution of Post/Pre-Test Difference in Mean Active Miniutes (New_User, Unknown)



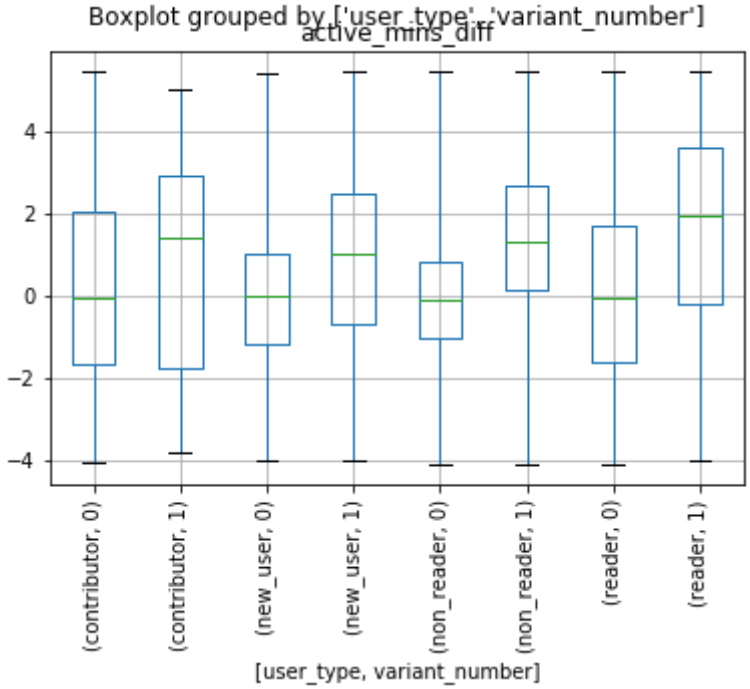
In [103]:
```
t1234[(t1234.active_mins_diff < t1234.active_mins_diff.quantile(0.9)) &
     (t1234.active_mins_diff > t1234.active_mins_diff.quantile(0.1))][[
'variant_number','gender', 'active_mins_diff']].boxplot(by=['gender', 'v
ariant_number'],whis='range',

showmeans=True)
plt.xticks(rotation=90)
```

Out[103]:
```
(array([1, 2, 3, 4, 5, 6]), <a list of 6 Text xticklabel objects>)
```



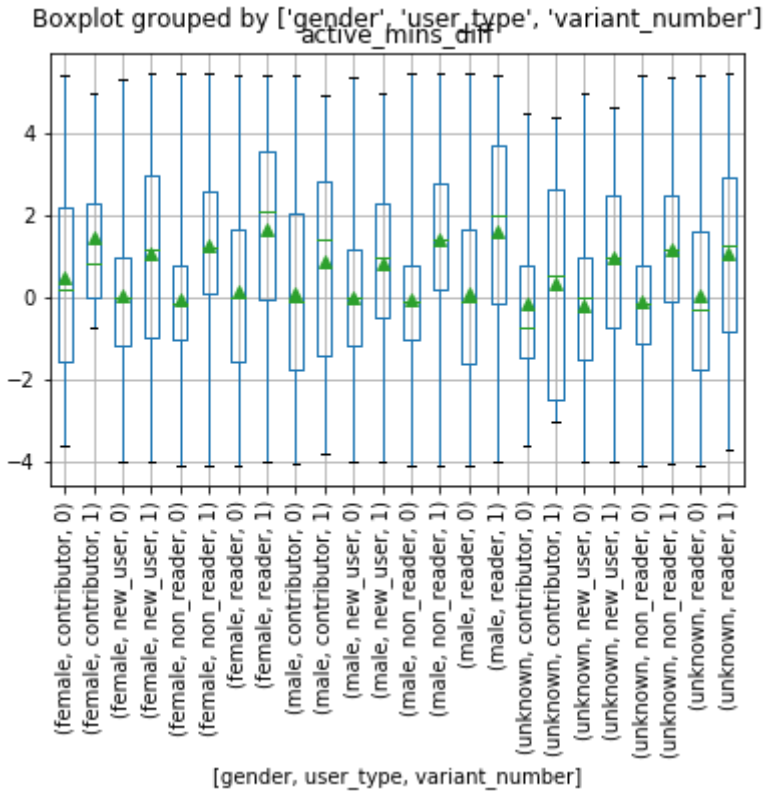
In [104]:
```
t1234[(t1234.active_mins_diff < t1234.active_mins_diff.quantile(0.9)) &
     (t1234.active_mins_diff > t1234.active_mins_diff.quantile(0.1))][[
'variant_number', 'user_type', 'active_mins_diff']].boxplot(by=['user_ty
pe', 'variant_number'], whis='range')
plt.xticks(rotation=90)
```

Out[104]:
```
(array([1, 2, 3, 4, 5, 6, 7, 8]), <a list of 8 Text xticklabel objects
>)
```

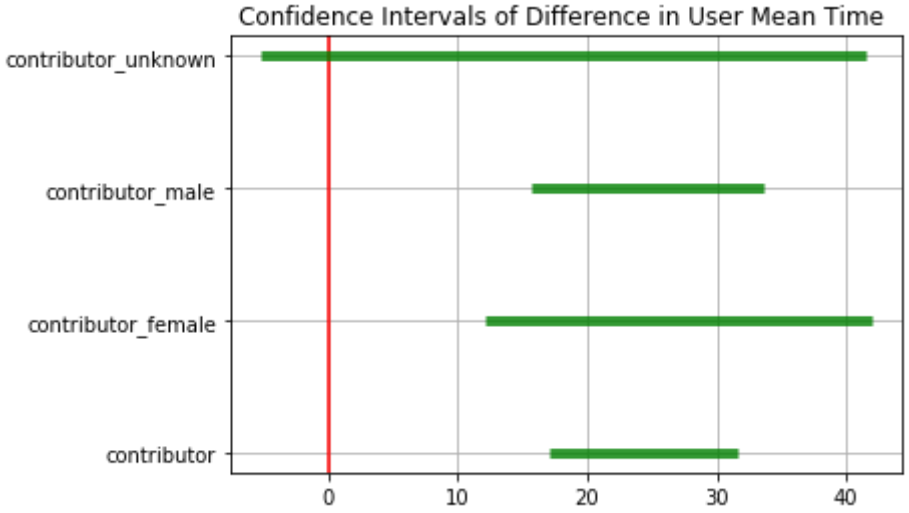


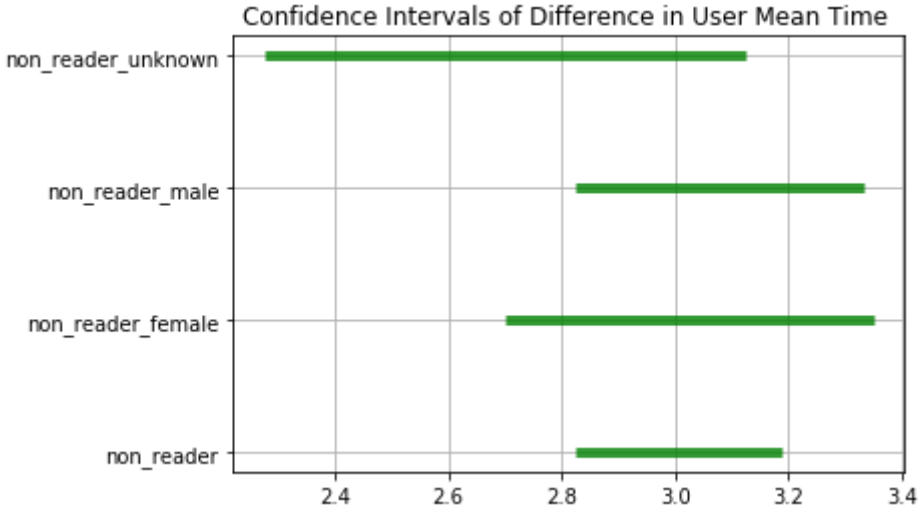
In [105]:
```
t1234[(t1234.active_mins_diff < t1234.active_mins_diff.quantile(0.9)) &
     (t1234.active_mins_diff > t1234.active_mins_diff.quantile(0.1))][[
'variant_number', 'gender', 'user_type', 'active_mins_diff']].boxplot(by
=['gender', 'user_type', 'variant_number'], showmeans=True, whis='range'
)
plt.xticks(rotation=90)
```

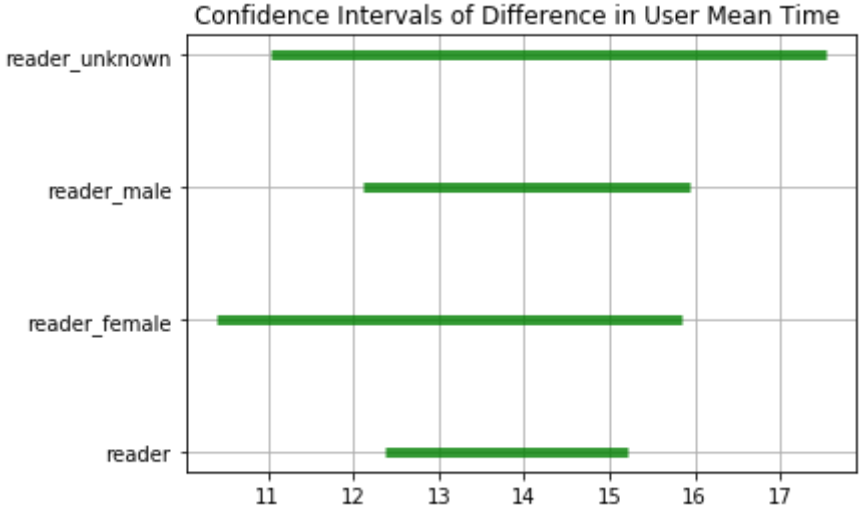
Out[105]:
```
(array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
17,
        18, 19, 20, 21, 22, 23, 24]), <a list of 24 Text xticklabel obj
ects>)
```

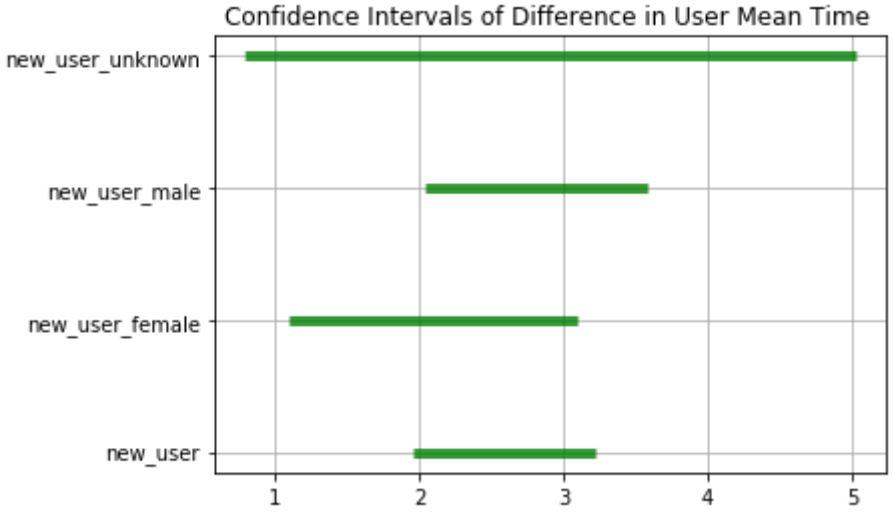


In [107]:
```
for u in set(t4_user_attributes.user_type):
    plt.figure()
    plt.title('Confidence Intervals of Difference in User Mean Time ')
    plt.grid()
    plt.vlines(0, -1, 4, colors='r')
    for i in df_intervals.columns[df_intervals.columns.str.startswith(u
)]:
        plt.hlines(i, df_intervals[i][0], df_intervals[i][1], colors='g'
, alpha=0.8, linestyles='solid', label=i, linewidth=5.0)
```


Confidence Intervals of Difference in User Mean Time
contributor_unknown
contributor_male
contributor_female
contributor
0
10
20
30
40


Confidence Intervals of Difference in User Mean Time
non_reader_unknown
non_reader_male
non_reader_female
non_reader
2.4
2.6
2.8
3.0
3.2
3.4


Confidence Intervals of Difference in User Mean Time
reader_unknown
reader_male
reader_female
reader
11
12
13
14
15
16
17


Confidence Intervals of Difference in User Mean Time
new_user_unknown
new_user_male
new_user_female
new_user
1
2
3
4
5