Assignment 3: Did deplatforming reduce misinformation on Twitter?

wdtmacss@berkeley.edu
Computational Social Science 1A
Human Psychology and Social Technologies Fall 2024
UC Berkeley Masters in Computational Social Science

Your goal is to write a brief computational essay providing data-driven answers to the question posed in the title and reflections on the strengths and limitations of the dataset and the Difference in Differences method. **Detailed guidelines for completing this assignment are avilable here.**

Practical Instructions:

- Take a copy of this notebook and complete Sections 2 - 5. Add as many code and markdown cells as you need within those sections.
- Answer the External Resources question in Section 6.
- Submit your completed notebook through gradescope.

Due date: 10/14/2024 (before midnight Pacific time)

Grading guidelines are included in the assignment description here.

The AI model usage policy is available here.

Class materials You are welcome to make use of the materials we have developed during this class. The original notebooks can be accessed below, and you are welcome to also consult your own copies of the notebooks you worked on during lab sessions.

- Notebook 1: Data Acquisition
- Notebook 2: Data Exploration

- Notebook 3: Data Simulation
- Notebook 4: Data Analysis
- Notebook 5: Class Project

Section 1: Twitter Dataset

Here is the research paper
Post-January 6th deplatforming reduced the reach of misinformation on Twitter

The dataset that accompanies this paper has been compiled and included below as a Pandas dataframe (assigned to the variable `mccabe_data`). Please base your main analyses on this shared dataset.

In [1]:
```
import pandas as pd
```

In [2]:
```
mccabe_data = pd.read_csv('mccabe_mine.csv')
```

You are welcome to rename the dataset or work with different subsets of this data or with additional datasets if neccesary, but this shared dataset should be the primary source for your analyses, so that we are all working with the same underlying source of information.

Section 2 Exploring the structure of the dataset

Describe the key variables you are interested in. Feel free to include data summaries and/or vizualizations that illustrate how the dataset is structured, such as the different groups of users you are interested in and the different measures of whether posts are classified as misinformation, etc.

In [6]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

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

Out[4]:

	Unnamed: 0	date	fake_merged	fake_merged_initiation	fake_merged_rt	fake_grinl
0	0	2019-11-30	875.0	199.0	676.0	
1	1	2019-12-01	3382.0	825.0	2557.0	
2	2	2019-12-02	3644.0	992.0	2652.0	
3	3	2019-12-03	4184.0	1110.0	3074.0	
4	4	2019-12-04	4436.0	1100.0	3336.0	

5 rows × 30 columns

In [13]:

```
print("\nBasic information about the dataset:")
print(mccabe_data.info())

print("\nSummary statistics of the dataset:")
print(mccabe_data.describe())

print("\nMissing values in each column:")
print(mccabe_data.isnull().sum())

print("\nColumn names in the dataset:")
print(mccabe_data.columns)

print("\nUnique values in 'group' column:")
print(mccabe_data['group'].unique())

print("\nUser group statistics:")
print(mccabe_data.groupby('group')['nusers'].describe())

print("\nTime range in the dataset:")
print(mccabe_data['date'].min(), "to", mccabe_data['date'].max())
```

```
Basic information about the dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32968 entries, 0 to 32967
Data columns (total 32 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Unnamed: 0                         32968 non-null  int64
 1   date                               32968 non-null  datetime64[ns]
 2   fake_merged                        32968 non-null  float64
 3   fake_merged_initiation             32968 non-null  float64
 4   fake_merged_rt                     32968 non-null  float64
 5   fake_grinberg_initiation           32968 non-null  float64
 6   fake_grinberg_rt                   32968 non-null  float64
 7   fake_grinberg_rb_initiation        32968 non-null  float64
 8   fake_grinberg_rb_rt                32968 non-null  float64
 9   fake_newsguard_initiation          32968 non-null  float64
 10  fake_newsguard_rt                  32968 non-null  float64
 11  not_fake                           32968 non-null  float64
 12  not_fake_initiation                32968 non-null  float64
 13  not_fake_rt                        32968 non-null  float64
 14  not_fake_conservative              32968 non-null  float64
 15  not_fake_conservative_initiation   32968 non-null  float64
 16  not_fake_conservative_rt           32968 non-null  float64
 17  not_fake_liberal                   32968 non-null  float64
 18  not_fake_liberal_initiation        32968 non-null  float64
 19  not_fake_liberal_rt                32968 non-null  float64
 20  not_fake_shopping                  32968 non-null  float64
 21  not_fake_shopping_initiation       32968 non-null  float64
 22  not_fake_shopping_rt               32968 non-null  float64
 23  not_fake_sports                    32968 non-null  float64
 24  not_fake_sports_initiation         32968 non-null  float64
 25  not_fake_sports_rt                 32968 non-null  float64
 26  n                                  32968 non-null  float64
 27  stat                               32968 non-null  object
 28  nusers                             32968 non-null  int64
 29  group                              32968 non-null  object
 30  post_intervention                  32968 non-null  int64
 31  treatment                          32968 non-null  int64
dtypes: datetime64[ns](1), float64(25), int64(4), object(2)
memory usage: 8.0+ MB
None

Summary statistics of the dataset:
         Unnamed: 0                           date   fake_merged  \
count  32968.000000                          32968  32968.000000
mean   16483.500000  2020-08-29 19:27:21.446251008    908.742582
min        0.000000            2019-11-30 00:00:00      0.000000
25%     8241.750000            2020-04-15 00:00:00      0.241366
50%    16483.500000            2020-08-29 00:00:00      5.023829
75%    24725.250000            2021-01-14 00:00:00    636.000000
max    32967.000000            2021-05-31 00:00:00  19143.000000
std     9517.186174                            NaN   1929.087503

       fake_merged_initiation  fake_merged_rt  fake_grinberg_initiation  \
count            32968.000000    32968.000000              32968.000000
mean               189.598042      719.144540                 58.298524
```

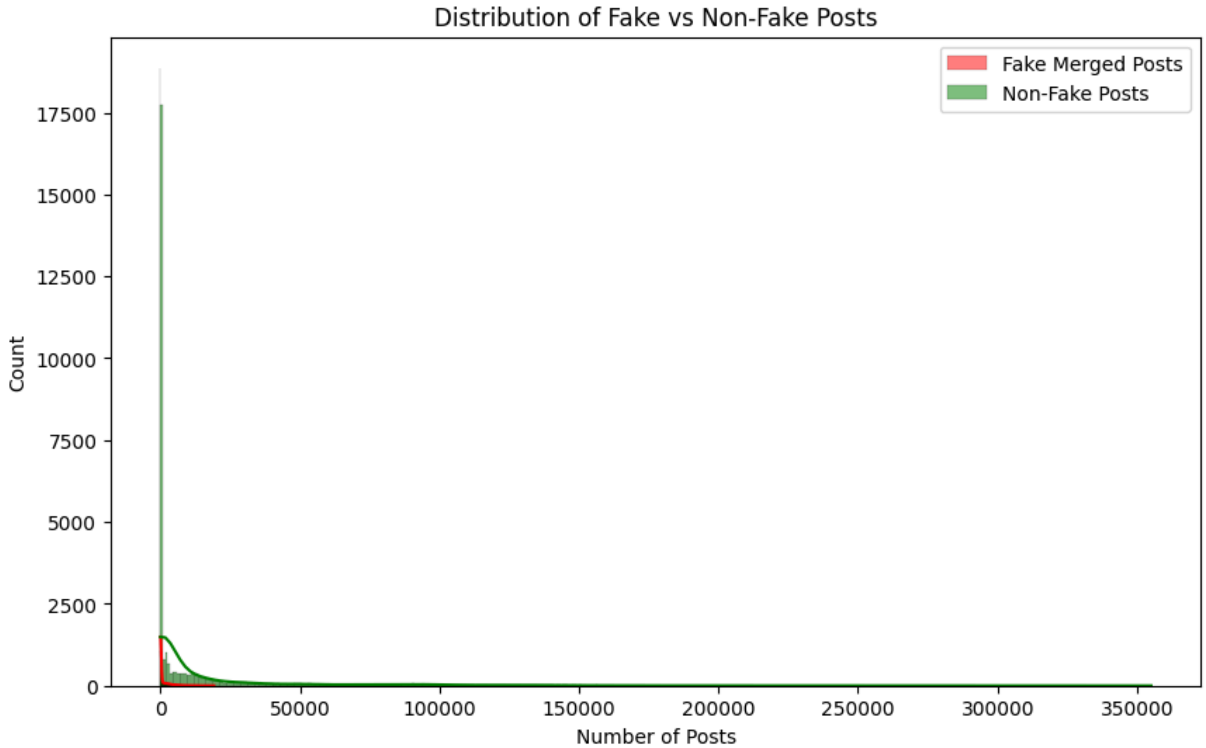
```
min                0.000000        0.000000                     0.000000
25%                0.056885        0.166247                     0.010742
50%                0.752520        4.115099                     0.174269
75%              146.250000      458.000000                    50.000000
max             3124.000000    16145.000000                  1034.000000
std              387.458512     1556.228804                   121.980244

       fake_grinberg_rt  fake_grinberg_rb_initiation  fake_grinberg_rb_rt  \
count      32968.000000                 32968.000000         32968.000000
mean         242.494367                    33.991539           151.367057
min            0.000000                     0.000000             0.000000
25%            0.038256                     0.001820             0.012547
50%            1.272802                     0.060986             0.680721
75%          134.000000                    32.000000            68.000000
max         5142.000000                   706.000000          4186.000000
std          543.356019                    73.742659           359.589577

       fake_newsguard_initiation  ...  not_fake_shopping  \
count               32968.000000  ...       32968.000000
mean                  180.477371  ...         355.111497
min                     0.000000  ...           0.000000
25%                     0.051722  ...           0.036345
50%                     0.713376  ...           0.714898
75%                   142.000000  ...         127.000000
max                  3033.000000  ...        6368.000000
std                   369.000258  ...         859.811119

       not_fake_shopping_initiation  not_fake_shopping_rt  not_fake_sports
\
count                  32968.000000          32968.000000     32968.000000
mean                     166.665689            188.445808        26.972055
min                        0.000000              0.000000         0.000000
25%                        0.014676              0.016265         0.004292
50%                        0.127660              0.385952         0.019805
75%                       59.000000             63.000000        16.000000
max                     3544.000000           3134.000000      1026.000000
std                      455.144240            439.546871        70.467300

       not_fake_sports_initiation  not_fake_sports_rt              n  \
count                32968.000000        32968.000000   32968.000000
mean                     9.513590           17.458465   18132.137786
min                      0.000000            0.000000       1.000000
25%                      0.000520            0.002443       5.078971
50%                      0.003257            0.012180      39.000000
75%                      4.000000           11.000000   14768.250000
max                    372.000000          718.000000  363619.000000
std                     25.509827           46.400901   39442.690214

             nusers  post_intervention     treatment
count  32968.000000       32968.000000  32968.000000
mean    9600.565821           0.265227      0.066246
min        1.000000           0.000000      0.000000
25%      519.750000           0.000000      0.000000
50%     1882.500000           0.000000      0.000000
75%     6164.250000           1.000000      0.000000
max    97893.000000           1.000000      1.000000
```

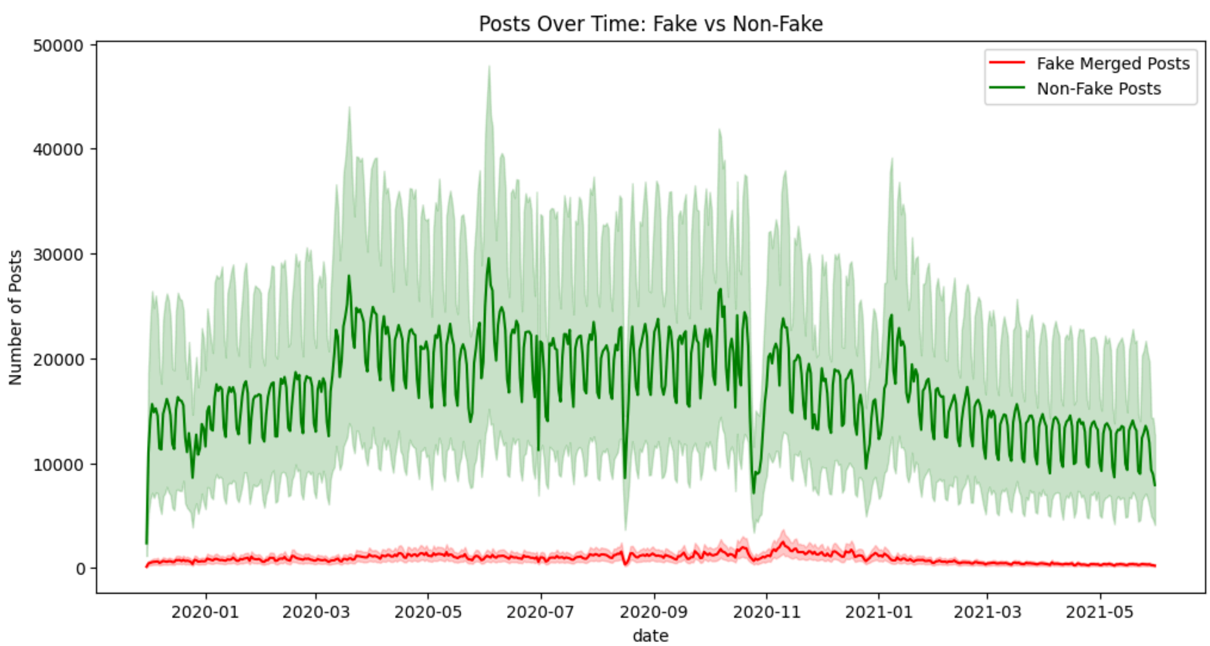
```
std    17992.132723           0.441461      0.248715

[8 rows x 30 columns]

Missing values in each column:
Unnamed: 0                          0
date                                0
fake_merged                         0
fake_merged_initiation              0
fake_merged_rt                      0
fake_grinberg_initiation            0
fake_grinberg_rt                    0
fake_grinberg_rb_initiation         0
fake_grinberg_rb_rt                 0
fake_newsguard_initiation           0
fake_newsguard_rt                   0
not_fake                            0
not_fake_initiation                 0
not_fake_rt                         0
not_fake_conservative               0
not_fake_conservative_initiation    0
not_fake_conservative_rt            0
not_fake_liberal                    0
not_fake_liberal_initiation         0
not_fake_liberal_rt                 0
not_fake_shopping                   0
not_fake_shopping_initiation        0
not_fake_shopping_rt                0
not_fake_sports                     0
not_fake_sports_initiation          0
not_fake_sports_rt                  0
n                                   0
stat                                0
nusers                              0
group                               0
post_intervention                   0
treatment                           0
dtype: int64

Column names in the dataset:
Index(['Unnamed: 0', 'date', 'fake_merged', 'fake_merged_initiation',
       'fake_merged_rt', 'fake_grinberg_initiation', 'fake_grinberg_rt',
       'fake_grinberg_rb_initiation', 'fake_grinberg_rb_rt',
       'fake_newsguard_initiation', 'fake_newsguard_rt', 'not_fake',
       'not_fake_initiation', 'not_fake_rt', 'not_fake_conservative',
       'not_fake_conservative_initiation', 'not_fake_conservative_rt',
       'not_fake_liberal', 'not_fake_liberal_initiation',
       'not_fake_liberal_rt', 'not_fake_shopping',
       'not_fake_shopping_initiation', 'not_fake_shopping_rt',
       'not_fake_sports', 'not_fake_sports_initiation', 'not_fake_sports_r
t',
       'n', 'stat', 'nusers', 'group', 'post_intervention', 'treatment'],
      dtype='object')

Unique values in 'group' column:
['fns' 'suspended' 'ha' 'ma' 'la' 'qanon' 'av' 'ss1' 'ss5' 'A' 'B' 'D' 'F'
```

```
 'all' 'nfns' 'nfns_ha' 'nfns_ma' 'nfns_la' 'A_ha' 'B_ha' 'D_ha' 'F_ha'
 'A_ma' 'B_ma' 'D_ma' 'F_ma' 'A_la' 'B_la' 'D_la' 'F_la']

User group statistics:
            count          mean           std     min      25%      50%  \
group
A          1100.0   1344.092727    263.914696   157.0   1188.0   1342.5
A_ha       1098.0    156.413479     17.038491     1.0    148.0    160.0
A_la       1086.0     62.257827     31.941995     1.0     38.0     56.0
A_ma       1100.0   1125.227273    238.531863     2.0    977.0   1131.5
B          1100.0   9118.687273   1641.219939  2696.0   7912.0   9288.0
B_ha       1100.0   2523.229091    331.738869     4.0   2319.0   2618.5
B_la       1100.0    380.816364    212.741773     3.0    211.0    330.0
B_ma       1100.0   6171.274545   1305.431216    17.0   5252.0   6288.5
D          1100.0   4662.180000    920.595718  1493.0   3983.0   4845.0
D_ha       1100.0   1569.501818    237.005782     2.0   1390.0   1651.5
D_la       1100.0    216.974545    131.206577     1.0    117.0    177.5
D_ma       1100.0   2846.629091    673.048789     6.0   2423.0   2929.5
F          1100.0   5655.423636   1022.558973  1363.0   5006.0   5662.0
F_ha       1100.0   1132.076364    133.799116    20.0   1053.0   1158.0
F_la       1100.0    535.489091    167.198625    84.0    412.0    528.0
F_ma       1100.0   3970.265455    829.019098    93.0   3443.0   3989.5
all        1100.0  65083.576364  13286.484104  6467.0  56975.0  66514.0
av         1100.0    235.558182     83.627142    71.0    144.0    278.5
fns        1100.0  15087.601818   2769.992545  4265.0  12956.0  15298.5
ha         1100.0   5432.318182    758.314274    54.0   4960.0   5616.0
la         1100.0   5628.063636   1856.949353   504.0   4387.0   5774.5
ma         1100.0  53085.232727  11734.563941   518.0  45931.0  53825.0
nfns       1100.0  49995.974545  10895.708734  2202.0  43424.0  51207.0
nfns_ha    1100.0   1634.800000    255.252876    28.0   1471.0   1647.0
nfns_la    1100.0   4685.347273   1506.069928   387.0   3726.0   4823.5
nfns_ma    1100.0  42799.060000   9794.658579   377.0  37059.0  43347.0
qanon      1100.0    460.090909    183.261154   135.0    284.0    534.0
ss1        1100.0    376.389091    112.038427   105.0    246.0    439.0
ss5        1100.0   1413.605455    366.336985   507.0   1058.0   1547.0
suspended  1084.0    355.780443    213.781263     1.0     97.0    449.0

                75%      max
group
A           1542.00   1879.0
A_ha         168.00    187.0
A_la          85.75    168.0
A_ma        1310.00   1575.0
B          10396.00  12288.0
B_ha        2781.00   2940.0
B_la         565.00   1003.0
B_ma        7161.00   8690.0
D           5384.00   6469.0
D_ha        1751.00   1866.0
D_la         326.00    592.0
D_ma        3362.00   4312.0
F           6500.00   7862.0
F_ha        1228.00   1339.0
F_la         651.00    948.0
F_ma        4650.00   5731.0
all        74467.00  97893.0
```

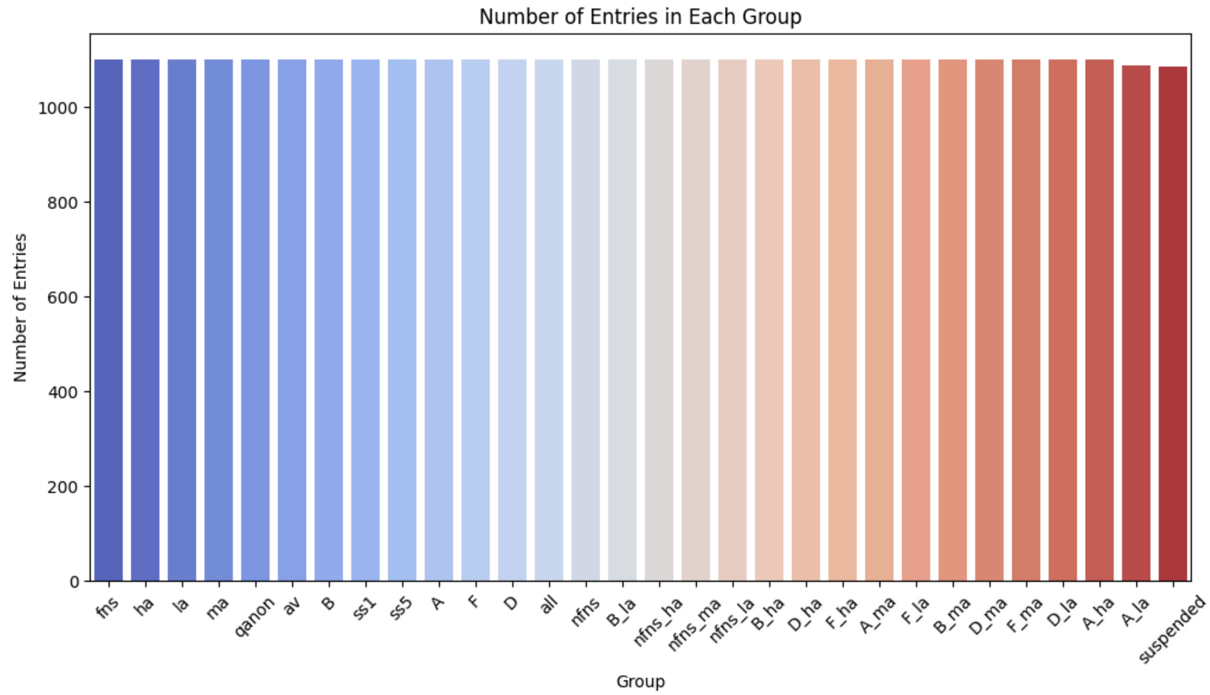
```
av            310.00    340.0
fns         17335.00  20434.0
ha           5997.00   6657.0
la           6808.00  10745.0
ma          61700.00  80466.0
nfns        57124.00  78097.0
nfns_ha      1824.00   2185.0
nfns_la      5655.00   8744.0
nfns_ma     49710.00  67295.0
qanon         621.00    709.0
ss1           472.00    512.0
ss5          1719.00   1976.0
suspended     519.00    656.0

Time range in the dataset:
2019-11-30 00:00:00 to 2021-05-31 00:00:00
```

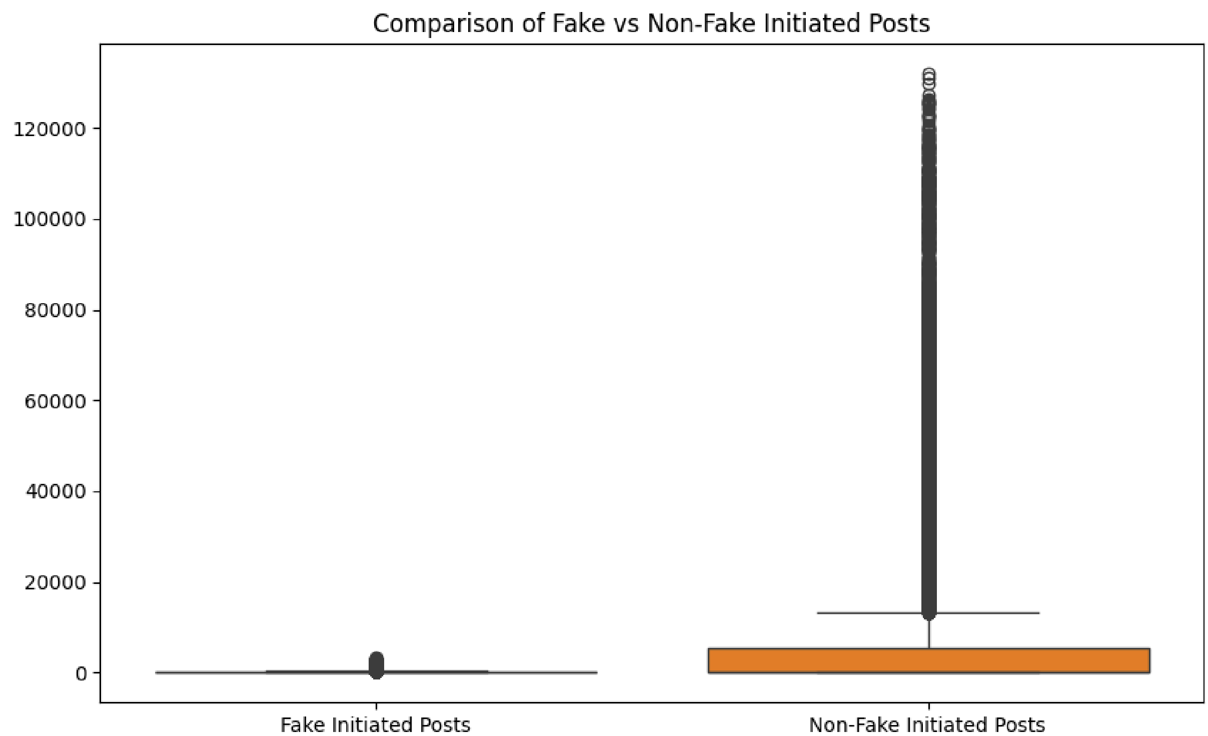
In [14]:
```
mccabe_data['date'] = pd.to_datetime(mccabe_data['date'])

plt.figure(figsize=(10, 6))
sns.histplot(data=mccabe_data, x='fake_merged', kde=True, color='red', label
sns.histplot(data=mccabe_data, x='not_fake', kde=True, color='green', label=
plt.title('Distribution of Fake vs Non-Fake Posts')
plt.xlabel('Number of Posts')
plt.legend()
plt.show()

plt.figure(figsize=(12, 6))
sns.lineplot(data=mccabe_data, x='date', y='fake_merged', label='Fake Merged
sns.lineplot(data=mccabe_data, x='date', y='not_fake', label='Non-Fake Posts
plt.title('Posts Over Time: Fake vs Non-Fake')
plt.ylabel('Number of Posts')
plt.legend()
plt.show()

plt.figure(figsize=(12, 6))
group_counts = mccabe_data['group'].value_counts()
sns.barplot(x=group_counts.index, y=group_counts.values, palette='coolwarm')
plt.title('Number of Entries in Each Group')
plt.xlabel('Group')
plt.ylabel('Number of Entries')
plt.xticks(rotation=45)
plt.show()

plt.figure(figsize=(10, 6))
sns.boxplot(data=mccabe_data[['fake_merged_initiation', 'not_fake_initiatior
plt.title('Comparison of Fake vs Non-Fake Initiated Posts')
plt.xticks([0, 1], ['Fake Initiated Posts', 'Non-Fake Initiated Posts'])
plt.show()
```





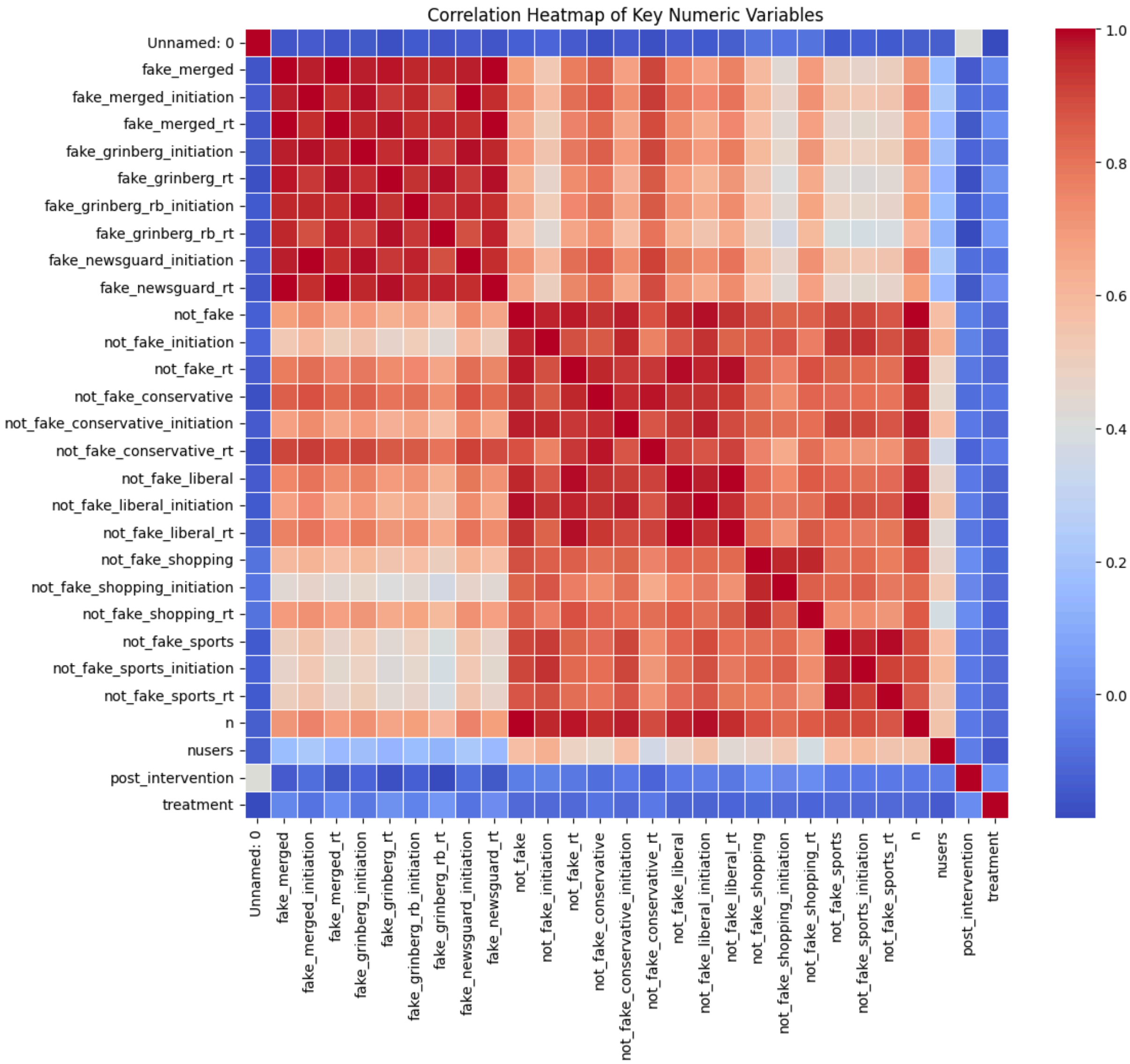

```
/var/folders/ms/m8xr2gmj6q10hz6k53q42l1c0000gn/T/ipykernel_13856/257293559.p
y:21: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(x=group_counts.index, y=group_counts.values, palette='coolwar
m')
```

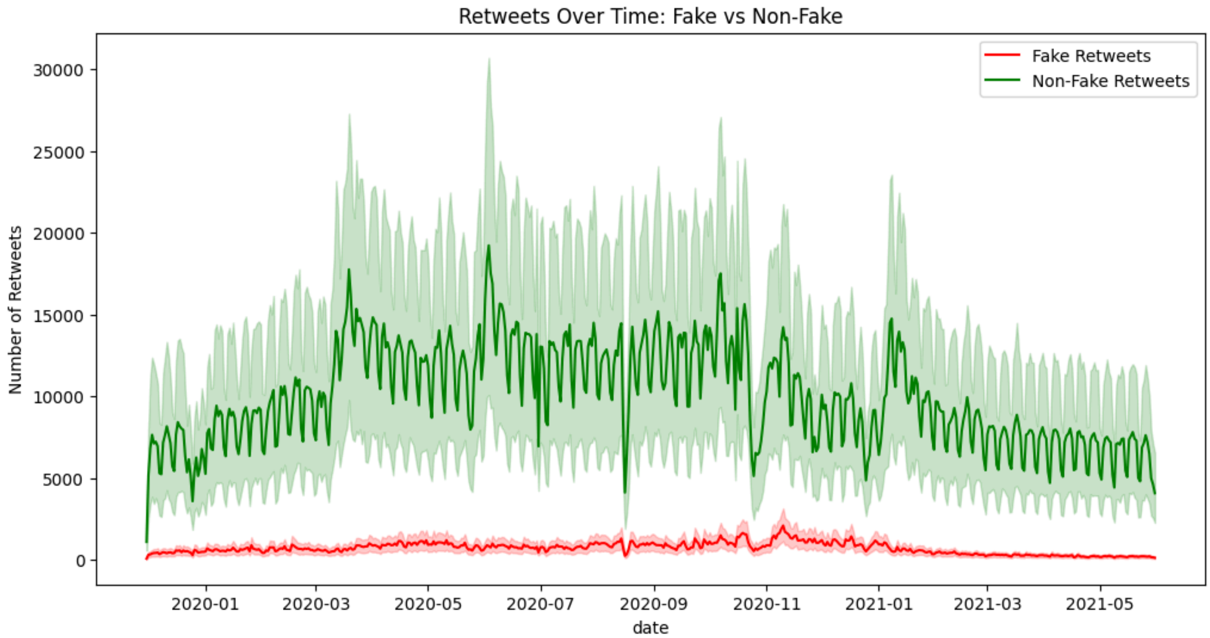


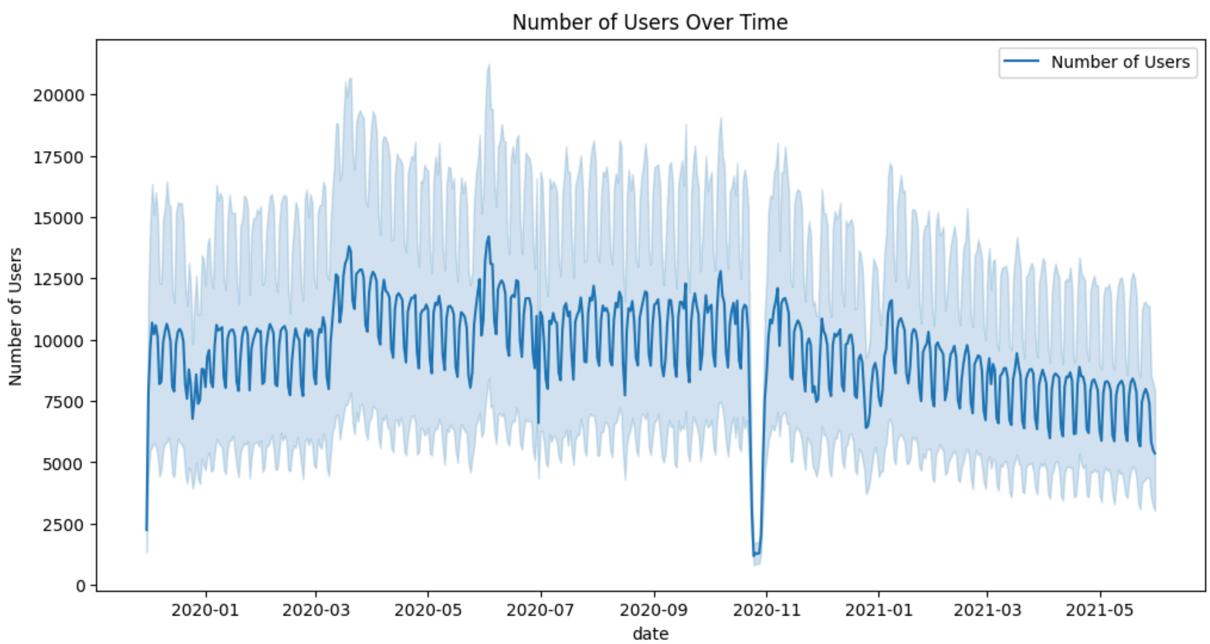


In [15]:
```
numeric_columns = mccabe_data.select_dtypes(include=[np.number]).columns
plt.figure(figsize=(12, 10))
sns.heatmap(mccabe_data[numeric_columns].corr(), annot=False, cmap='coolwarm
plt.title('Correlation Heatmap of Key Numeric Variables')
plt.show()

plt.figure(figsize=(12, 6))
sns.lineplot(data=mccabe_data, x='date', y='fake_merged_rt', label='Fake Ret
sns.lineplot(data=mccabe_data, x='date', y='not_fake_rt', label='Non-Fake Re
plt.title('Retweets Over Time: Fake vs Non-Fake')
plt.ylabel('Number of Retweets')
plt.legend()
```

```
plt.show()

plt.figure(figsize=(12, 6))
sns.lineplot(data=mccabe_data, x='date', y='nusers', label='Number of Users'
plt.title('Number of Users Over Time')
plt.ylabel('Number of Users')
plt.show()
```







Section 3 Replication of Main DiD Results

In this section, you will perform at least one Difference in Differences analysis with the goal of conceptually replicating the key DiD analysis that McCabe et al performed to support their primary conclusion.

In [16]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf

mccabe_data['date'] = pd.to_datetime(mccabe_data['date'])
```

```
intervention_date = pd.to_datetime('2021-01-06')
mccabe_data['post_intervention'] = (mccabe_data['date'] >= intervention_date

mccabe_data['treatment'] = mccabe_data['group'].apply(lambda x: 1 if x in ['

outcome_var = 'fake_merged'
outcome_var = 'fake_merged'

formula = f"{outcome_var} ~ treatment + post_intervention + treatment:post_i

model = smf.ols(formula, data=mccabe_data)
results = model.fit()

print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
==
Dep. Variable:            fake_merged   R-squared:                       0.0
19
Model:                            OLS   Adj. R-squared:                  0.0
19
Method:                 Least Squares   F-statistic:                     21
5.5
Date:                Fri, 18 Oct 2024   Prob (F-statistic):          2.01e-1
38
Time:                        19:54:01   Log-Likelihood:          -2.9586e+
05
No. Observations:               32968   AIC:                       5.917e+
05
Df Residuals:                   32964   BIC:                       5.918e+
05
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
===================
                                 coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
-------------------
Intercept                   1067.4086     12.706     84.005      0.000    1
042.504    1092.314
treatment                     -2.5842     49.196     -0.053      0.958
-99.009     93.841
post_intervention           -575.0139     24.656    -23.322      0.000    -
623.340    -526.688
treatment:post_intervention  -347.4205    96.400     -3.604      0.000    -
536.368    -158.473
==============================================================================
==
Omnibus:                    19460.559   Durbin-Watson:                   0.0
35
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           166678.8
45
Skew:                           2.804   Prob(JB):                         0.
00
Kurtosis:                      12.481   Cond. No.                         9.
98
==============================================================================
==

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corre
ctly specified.
```

In my DiD analysis, the results indicate that the interaction between treatment and post-intervention is significant, showing a negative coefficient (-347.42). This suggests that the deplatforming intervention significantly reduced the sharing of misinformation among treated users compared to the control group after January 6th. I used a simpler grouping of "qanon" and "suspended" users as the treatment group, while McCabe et al.

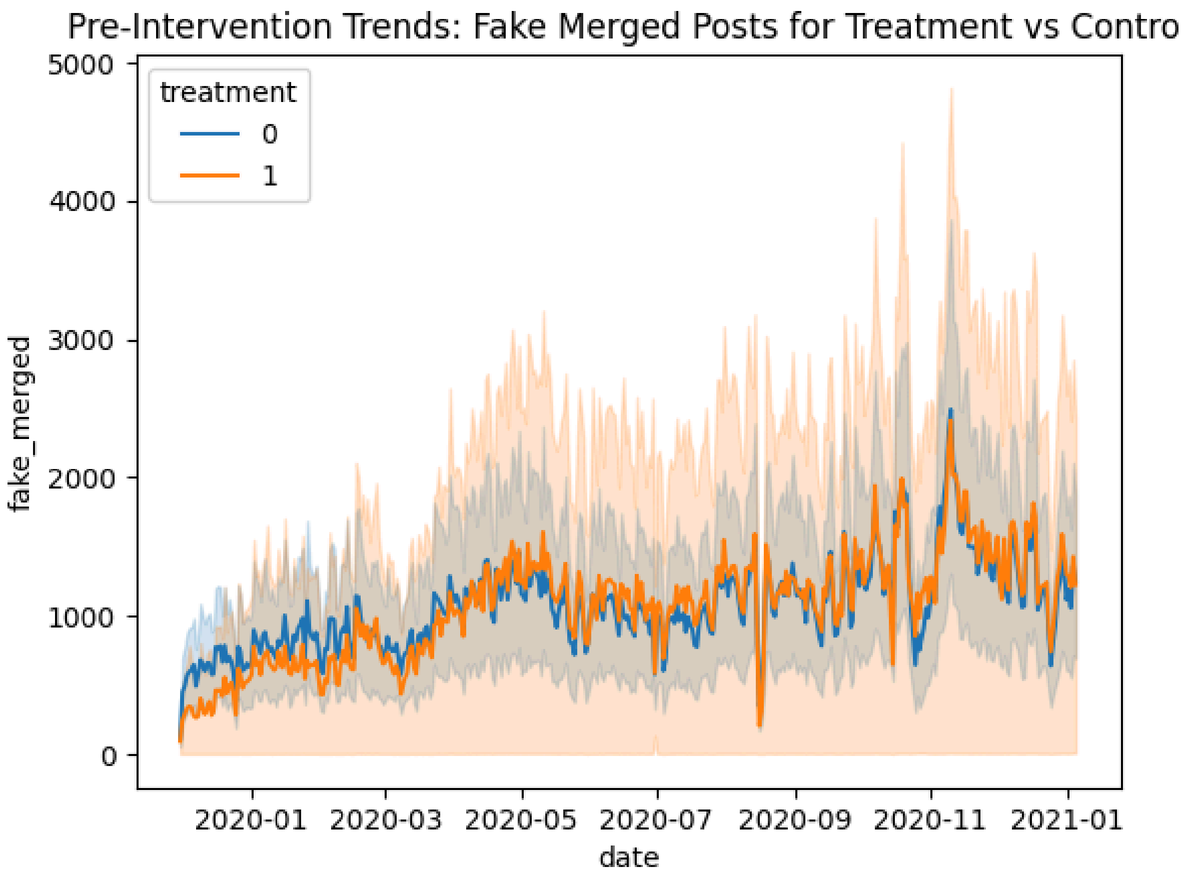
used a more nuanced approach, dividing users into three groups: deplatformed users, their followers, and unaffected users. This allowed McCabe to explore spillover effects on followers, while my analysis focuses on the direct impact on misinformation sharing. Despite these differences, both analyses point to a similar conclusion: deplatforming significantly reduced misinformation sharing, particularly in the treatment group, though McCabe's model provides a more comprehensive understanding of user behavior and event complexity.

Section 4 Extensions and follow up analyses

In this section, you will perform follow-up analyses, summaries, or visualizations that you feel help shed light on the robustness of the conclusion reached by McCabe et al. You are welcome to draw on insights you gained through data simulation, and to draw on the questions we discussed in class surrounding the **key assumptions and study decisions** in Notebook 1: Data Acquisition.

In [17]:

```
pre_intervention_data = mccabe_data[mccabe_data['date'] < intervention_date]

sns.lineplot(data=pre_intervention_data, x='date', y='fake_merged', hue='tre
plt.title('Pre-Intervention Trends: Fake Merged Posts for Treatment vs Contr
plt.show()
```



In [18]:

```
outcome_var_grinberg = 'fake_grinberg_rt'

formula_grinberg = f"{outcome_var_grinberg} ~ treatment + post_intervention
model_grinberg = smf.ols(formula_grinberg, data=mccabe_data)
results_grinberg = model_grinberg.fit()
print(results_grinberg.summary())
```

```
                            OLS Regression Results
==============================================================================
==
Dep. Variable:       fake_grinberg_rt   R-squared:                       0.0
30
Model:                            OLS   Adj. R-squared:                  0.0
30
Method:                 Least Squares   F-statistic:                     34
1.5
Date:                Fri, 18 Oct 2024   Prob (F-statistic):           2.07e-2
18
Time:                        19:54:47   Log-Likelihood:            -2.5390e+
05
No. Observations:               32968   AIC:                         5.078e+
05
Df Residuals:                   32964   BIC:                         5.078e+
05
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
===================
                                  coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
-------------------
Intercept                     293.5001      3.559     82.467      0.000
286.524     300.476
treatment                      71.2429     13.779      5.170      0.000
44.235      98.251
post_intervention            -201.7201      6.906    -29.210      0.000     -
215.256    -188.184
treatment:post_intervention  -129.0677     27.001     -4.780      0.000     -
181.991     -76.145
==============================================================================
==
Omnibus:                    20263.158   Durbin-Watson:                   0.0
56
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            192381.8
34
Skew:                           2.914   Prob(JB):                          0.
00
Kurtosis:                      13.300   Cond. No.                          9.
98
==============================================================================
==

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corre
ctly specified.
```

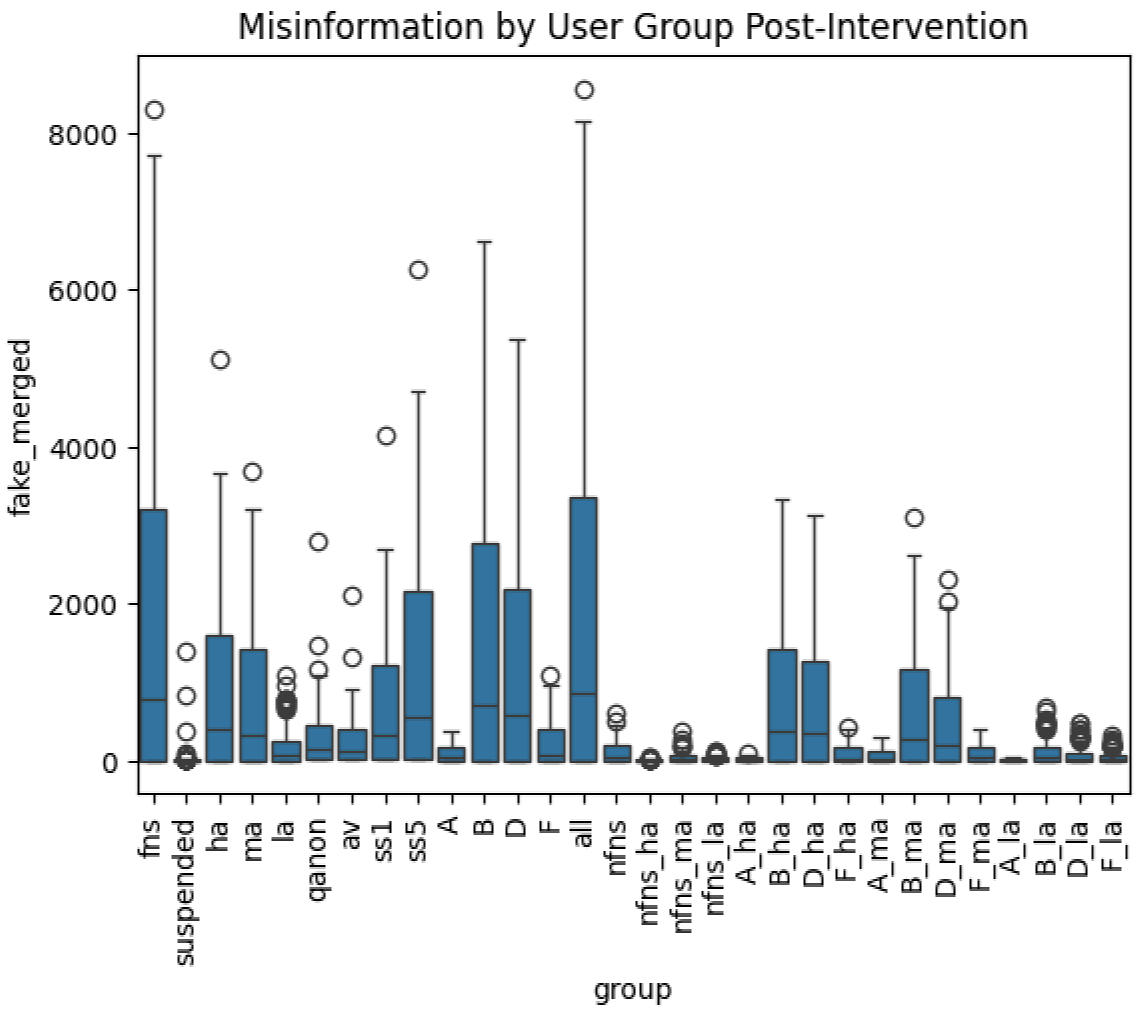
In [19]:

```
outcome_var_initiation = 'fake_merged_initiation'

formula_initiation = f"{outcome_var_initiation} ~ treatment + post_intervent
model_initiation = smf.ols(formula_initiation, data=mccabe_data)
results_initiation = model_initiation.fit()
print(results_initiation.summary())
```

```
                            OLS Regression Results
==============================================================================
======
Dep. Variable:     fake_merged_initiation   R-squared:
0.012
Model:                                OLS   Adj. R-squared:
0.012
Method:                     Least Squares   F-statistic:
138.8
Date:                    Fri, 18 Oct 2024   Prob (F-statistic):           2.
28e-89
Time:                            19:54:47   Log-Likelihood:            -2.43
05e+05
No. Observations:                   32968   AIC:                         4.8
61e+05
Df Residuals:                       32964   BIC:                         4.8
61e+05
Df Model:                               3
Covariance Type:                nonrobust
==============================================================================
===================
                                  coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
-------------------
Intercept                     216.9133      2.561     84.703      0.000
211.894     221.933
treatment                    -105.5282      9.915    -10.643      0.000    -
124.962     -86.095
post_intervention             -75.8205      4.969    -15.258      0.000
-85.560     -66.081
treatment:post_intervention   -12.4667     19.429     -0.642      0.521
-50.547      25.614
==============================================================================
==
Omnibus:                    17550.113   Durbin-Watson:                   0.0
50
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           109573.2
04
Skew:                           2.587   Prob(JB):                         0.
00
Kurtosis:                      10.279   Cond. No.                         9.
98
==============================================================================
==

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corre
ctly specified.
```

In [12]:

```
sns.boxplot(data=mccabe_data[mccabe_data['post_intervention'] == 1], x='grou
plt.xticks(rotation=90)
plt.title('Misinformation by User Group Post-Intervention')
plt.show()
```



Section 5 Conclusions and Reflections

Here is where you draw together insights you have gained by analyzing this dataset and reflections on the methods we have applied. You should provide a clear answer to the question:

> What are your conclusions about the question posed in this assignment:
> **Did deplatforming reduce misinformation on Twitter?**

You are welcome to use the bullet points below to guide your reflections if they are helpful, and also to include any additional insights.

- Is the current dataset sufficient to offer insight into this question? What are some key limitations of the dataset, and key merits?
- Is the DiD method sufficient to support strong conclusions related to this question?
- Overall, do you think the conclusions of McCabe et al. (2024) are justified?

- More generally, do you feel that misinformation on social media is a substantial threat to discourse and society that data science can address, and how has this project influenced your view?

Conclusions on Deplatforming and Misinformation:

Based on the Difference in Differences analysis, it is clear that deplatforming had a significant impact on reducing misinformation sharing on Twitter, especially among treated groups like "qanon" and "suspended." Both the retweeting of misinformation and the initiation of misinformation-related posts were significantly reduced post-intervention, although the reduction in retweets was more pronounced. These results align with the conclusion that deplatforming helped curb the spread of misinformation.

However, the current dataset offers both strengths and limitations. A key merit is how specific the dataset is of user behaviors, which allows for detailed analysis of misinformation at different levels. A limitation, however, is the presence of confounding real-world events that may influence user behavior, such as political influences, users that follow other users that are deplatformed but dont know about it, which were not fully isolated in the analysis.

The DiD method is generally appropriate for this type of intervention analysis, as it helps capture the effect of deplatforming by comparing trends before and after the event. However, the assumptions of trends and the presence of confounding factors can limit the strength of the conclusions. Despite these limitations, the method still provides strong evidence for the effectiveness of deplatforming in reducing misinformation.

Overall, McCabe et al.'s conclusions appear justified, particularly given the significant reduction in misinformation tweets in treated groups. Their use of complementary methods, like other types of regressions such as the Discontinuity Design (SRD), further strengthens their conclusions.

Misinformation on social media has grown into a substantial threat to the public, spreading false things, and creating distrust in institutions, and polarizing societies. The rapid spread of misinformation can fuel political unrest and with todays sociual media apps it is much easier to spread and at really fast rates. The findings i got from this project show how deplatforming, removing or limiting the posts that spread false information—can be an effective intervention in reducing the reach and impact of misinformation. By analyzing user behavior before and after deplatforming, we’ve seen a significant reduction in the spread of misinformation, particularly among the most engaged users, demonstrating that this approach can disrupt misinformation networks.

I really believe data analysis plays a critical role in combating misinformation, by understanding how it spreads and evaluating the effectiveness of interventions like deplatforming.

I also think this project helped me reinforce the idea that while social media platforms enable rapid information sharing, they also have the responsibility—and the tools— to manage the spread of harmful content. However i still believe that there freedom of speech and the way companies treat disinformation should be regulated by law, and not by the companies themselves, this is because they can create bias and also help different political interests without the public knowing.

Section 6 Use of External Resources

Please indicate here your use of external resources such as coding assitants or other AI systems to aid in completing this assignment. Please select one of the options below by placing an `[x]` next to the relevant option. You may also include any additional notes that may help gradeers assess your reliance on external resources.

No Usage

`[]` I attest that I did not use a coding assistant such as ChatGPT or other large language models to complete this assignment.

Declared Usage

`[X]` I made use of a coding assistant such as ChatGPT or other large language models to complete this assignment.

If you select this option, you are required to include a record of your interaction with the coding assistant here. Please include in the cell below either a link to the transcript or the transcript itself. If you provide a link, it is your responsibility to ensure that the link works and can be accessed by the graders.

Transcript

(https://chatgpt.com/share/67132258-c728-8009-84e2-aafdcc169a98)

Additional Notes

Any additional notes on your use of a codign assistant go here.