Monday Group Challenge

Group: aPPLE

Covering MickeyMouseClubHouse's Bad Science

Computational Social Science, University of Amsterdam

Building Blocks

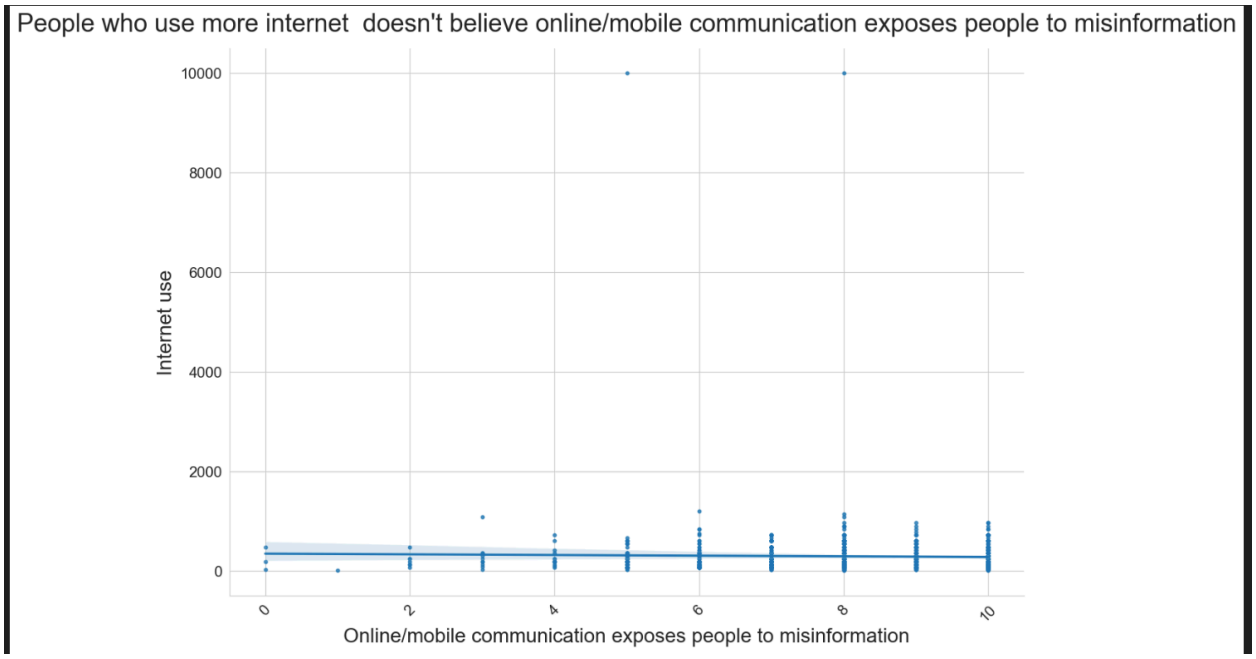
CL: Wahideh Achbari

February 3, 2025

Bad Science

MickeyMouseClubHouse group claims that people who use more internet do not believe communication exposes people to misinformation.

The group has two variables picked without consideration of confounding variables



Debunking

The data that they have used for their claim was not properly clean. It contains data like: 9999, which stands for no answer/missing value. We have cleaned their data to zoom in on the important information. *Online/mobile communication exposes people to misinformation*, should be presented as Dependent Variable in the graph, while *Internet Use* is supposed to be the independent variable.

Internet use does not imply use of communication, the latter is, theoretically the subset of the former, however in this study being specific to *Online/mobile communication exposes people to misinformation*, the variable *Internet Use* should not be arbitrarily taken.

Clean data:

```
replace_with_na = [6666, 7777, 8888, 9999]
replace_with_n = [7, 8, 9]
df['netustm'].replace(replace_with_na, np.nan, inplace=True)
```

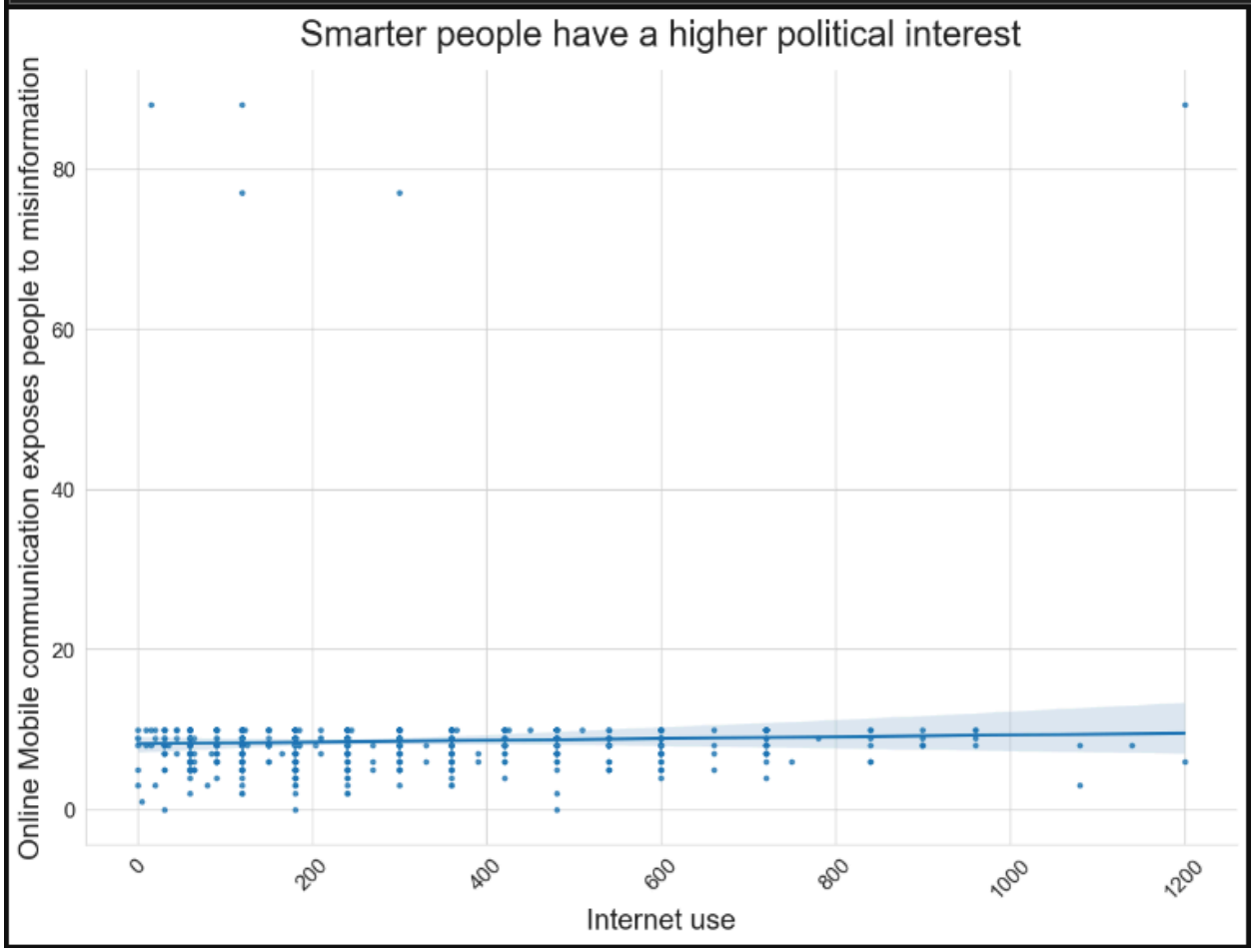
```
replace_with_n = [77, 88, 99]
df['mcmsinf'].replace(replace_with_na, np.nan, inplace=True)

df_red = df[["netustm", "mcmsinf"]].dropna()

from scipy.stats import pearsonr
pearsonr(df_red['netustm'], df_red['mcmsinf'])
```

PearsonRResult(statistic=0.040508623283611184, pvalue=0.21345094913977988)

Even with the data cleaned the correlation is nowhere near significant. r=0.04

Clean data graph:

```
sns.set_style("whitegrid")
sns.lmplot(data=df_red, x="netustm", y="mcmsinf",
           height=8.27, aspect=12/8.27,
          scatter_kws={"s": 10})
plt.title ("Smarter people have a higher political interest", fontsize=25, y=1.02)
plt.xlabel("Internet use", fontsize=20)
plt.ylabel("Online Mobile communication exposes people to misinformation", fontsize=20)
plt.xticks(fontsize=15, rotation=45)
plt.yticks(fontsize=15)
plt.show()
```



The correlation matrix reflected LOW relevance of the two selected variables.

```python
from scipy.stats import pearsonr
import numpy as np
def corrmat(name_df):
    rho = name_df.corr()
    pval = name_df.corr(method=lambda x, y: pearsonr(x, y)[1]) - np.eye(*rho.shape)
    p = pval.applymap(lambda x: ''.join(['*' for t in [.05, .01, .001] if x<=t]))
    return rho.round(2).astype(str) + p
```

[24] ✓ 0.0s Python

```python
correlation_matrix = df[["netustm", "mcmsinf", "ppltrst"]]
```

[25] ✓ 0.0s Python

```python
corrmat(correlation_matrix)
```

[26] ✓ 0.0s Python

```
C:\Users\19241\AppData\Local\Temp\ipykernel_10420\3200642117.py:6: FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map instead.
  p = pval.applymap(lambda x: ''.join(['*' for t in [.05, .01, .001] if x<=t]))
```

	netustm	mcmsinf	ppltrst
netustm	1.0***	0.03	-0.02
mcmsinf	0.03	1.0***	-0.02
ppltrst	-0.02	-0.02	1.0***

```python
# create a contingency table:
# by adding normalize="index" and multiplying by 100, the table contains percentages instead of absolute numbers.
contingency_table = pd.crosstab(df['netustm'], df['mcmsinf'], normalize='columns')*100 # Also try: normalize='index'
contingency_table.round()
```

[28] ✓ 0.0s Python

mcmsinf / netustm	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
5.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20.0	0.0	0.0	0.0	10.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
30.0	33.0	0.0	0.0	0.0	0.0	3.0	0.0	4.0	2.0	2.0	2.0
35.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
45.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0
60.0	0.0	0.0	14.0	0.0	8.0	17.0	20.0	12.0	11.0	12.0	13.0
64.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0
65.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0	1.0	0.0	1.0	0.0
80.0	0.0	0.0	0.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
85.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
90.0	0.0	0.0	0.0	0.0	8.0	0.0	8.0	3.0	4.0	1.0	5.0
120.0	0.0	0.0	29.0	10.0	8.0	12.0	11.0	19.0	13.0	20.0	19.0
123.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
125.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
150.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	3.0	1.0	2.0
165.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
180.0	33.0	0.0	14.0	20.0	31.0	22.0	14.0	11.0	12.0	10.0	12.0
185.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
204.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
210.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0
240.0	0.0	0.0	29.0	10.0	15.0	10.0	5.0	12.0	9.0	10.0	6.0