The Effect of Weibo Activity on TV Viewership—Instrumental Variable

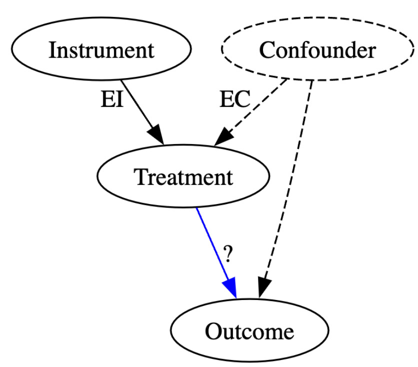
Professor Song Yao
Olin Business School

Customer Analytics

Requirements of a Valid IV

Key assumptions

- Relevance
  - The IV is strongly correlated with Treatment
- Exogeneity (or Independence)
  - The IV is independent of any confounders
- Exclusion Restriction
  - The IV does not directly affect the outcome
  - Its effect is only through the treatment



Why are censor_dummy and weibo_active_level reasonable IVs?

censor_dummy

- Relevance
  - Comments were reduced during the censor event
- Independence
  - The censor is targeted at the platform, not at TV shows or TV watching
- Exclusion restriction
  - The censor is targeted at the platform, not at TV shows or TV watching

Why are censor_dummy and weibo_active_level reasonable IVs?

weibo_active_level

- Relevance
  - Comments on TV shows are part of the audience's Weibo activity level, hence the two correlate with one another
- Independence
  - Weibo activity level focues on the overall activity at the platform, not about TV watching
- Exclusion restriction
  - Weibo activity level focues on the overall activity at the platform, not about TV watching

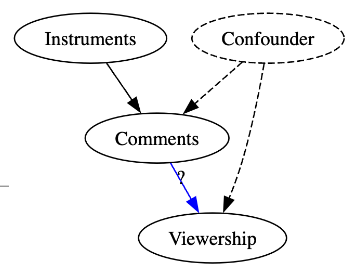
OLS Estimation

Biased Estimate of Effect

```python
### OLS without IV
formula='log_rating ~ log_comment + C(weekday)'
OLS_model = smf.ols(formula=formula, data=df).fit()
print(OLS_model.summary())
```

```
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
Intercept           -0.0159      0.166     -0.096      0.924      -0.342       0.310
C(weekday)[T.2]      0.2180      0.155      1.406      0.160      -0.086       0.522
C(weekday)[T.3]      0.0606      0.151      0.401      0.688      -0.236       0.357
C(weekday)[T.4]      0.1246      0.147      0.849      0.396      -0.163       0.412
C(weekday)[T.5]     -0.0877      0.151     -0.579      0.562      -0.385       0.209
C(weekday)[T.6]      0.0257      0.149      0.172      0.863      -0.267       0.318
C(weekday)[T.7]      0.1633      0.154      1.062      0.288      -0.138       0.465
log_comment          0.3588      0.012     30.236      0.000       0.336       0.382
==============================================================================
Omnibus:                        0.073   Durbin-Watson:                   1.952
Prob(Omnibus):                  0.964   Jarque-Bera (JB):                0.027
Skew:                           0.006   Prob(JB):                        0.987
Kurtosis:                       3.023   Cond. No.                         86.6
==============================================================================
```


IV by Hand



2nd Stage

```python
### IV by hand
### Second Stage
df['log_comment_fitted'] = first_stage.fittedvalues

formula = 'log_rating ~ C(weekday) + log_comment_fitted'
second_stage = smf.ols(formula=formula, data=df).fit()
print(second_stage.summary().tables[1])
```

```
======================================================================================
                         coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------
Intercept              2.6075      0.261      9.993      0.000       2.095       3.120
C(weekday)[T.2]        0.3135      0.211      1.483      0.138      -0.101       0.728
C(weekday)[T.3]        0.1924      0.206      0.933      0.351      -0.212       0.597
C(weekday)[T.4]        0.3393      0.200      1.694      0.091      -0.054       0.732
C(weekday)[T.5]        0.0088      0.206      0.043      0.966      -0.396       0.414
C(weekday)[T.6]        0.0377      0.203      0.185      0.853      -0.361       0.437
C(weekday)[T.7]        0.3776      0.210      1.800      0.072      -0.034       0.789
log_comment_fitted     0.1167      0.020      5.801      0.000       0.077       0.156
======================================================================================
```

Using the 2SLS Library

```python
## Import IV2SLS from linearmodels
## This is a more efficient way to run IV regression
from linearmodels.iv import IV2SLS

formula = 'log_rating ~ C(weekday) + [log_comment ~ censor_dummy + weibo_active_level]'
iv2sls = IV2SLS.from_formula(formula = formula, data = df).fit()
print(iv2sls.summary.tables[1])
```

```
                              Parameter Estimates
==================================================================================
                 Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
----------------------------------------------------------------------------------
C(weekday)[1]       2.6075     0.2155     12.101     0.0000      2.1852      3.0298
C(weekday)[2]       2.9210     0.2461     11.869     0.0000      2.4387      3.4034
C(weekday)[3]       2.8000     0.2356     11.883     0.0000      2.3381      3.2618
C(weekday)[4]       2.9468     0.2464     11.961     0.0000      2.4640      3.4297
C(weekday)[5]       2.6163     0.2339     11.186     0.0000      2.1579      3.0748
C(weekday)[6]       2.6452     0.2203     12.009     0.0000      2.2135      3.0770
C(weekday)[7]       2.9851     0.2511     11.888     0.0000      2.4930      3.4773
log_comment         0.1167     0.0176     6.6484     0.0000      0.0823      0.1511
==================================================================================
```


Diagnoses (Check the Three Assumptions)

```python
## Check the strength of the instruments (Relevance)
# Get first stage F-stat from the IV2SLS model
first_stage_f = iv2sls.first_stage.diagnostics['f.stat'].iloc[0]
print("\nFirst Stage F-statistic:")
print(f"{first_stage_f:.2f}")

# Rule of thumb: F-stat > 10 indicates strong instruments
print(f"Instruments are {'not weak' if first_stage_f > 10 else 'weak'}")
```

```
First Stage F-statistic:
1879.41
Instruments are not weak
```

$$F = \frac{(R^2_{\text{WithIV}} - R^2_{\text{NoIV}})/\#IV}{(1 - R^2_{\text{WithIV}})/(\#Obs - \#IV - \#X - 1)}$$

```python
## Check the IV's exclusion restriction and exogeneity
### The lucky case: If we have more IVs than endogenous variables:

j_stat = iv2sls.sargan
print(j_stat)
## A p-value > 0.05 indicates that we cannot reject the null hypothesis, that is, all IVs are valid!
```

```
Sargan's test of overidentification
H0: The model is not overidentified.
Statistic: 0.2206
P-value: 0.6386
Distributed: chi2(1)
```