# Instrumental Variable

**Professor Song Yao**
Olin Business School

**Customer Analytics**

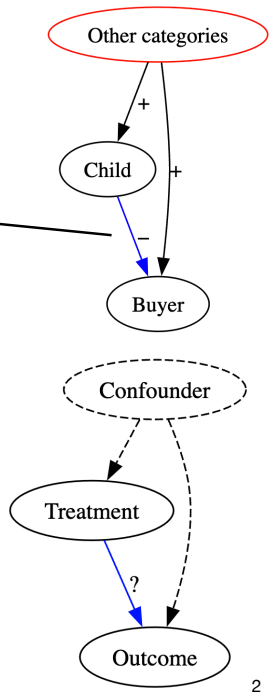
## Let's revisit an old example

### Original Model

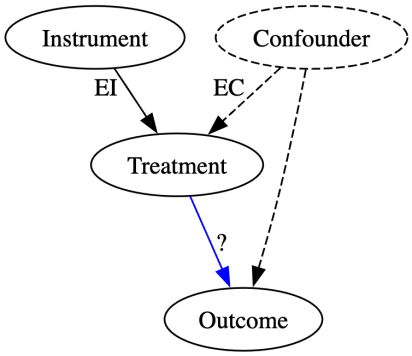
| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | −1.3676 | 0.043 | −31.457 | 0.000 | −1.453 | −1.282 |
| last | −0.0947 | 0.003 | −33.918 | 0.000 | −0.100 | −0.089 |
| total_scaled | 0.1131 | 0.020 | 5.630 | 0.000 | 0.074 | 0.152 |
| female | −0.7607 | 0.036 | −21.272 | 0.000 | −0.831 | −0.691 |
| child | −0.1862 | 0.017 | −10.775 | 0.000 | −0.220 | −0.152 |
| youth | −0.1130 | 0.026 | −4.327 | 0.000 | −0.164 | −0.062 |
| cook | −0.2703 | 0.017 | −15.782 | 0.000 | −0.304 | −0.237 |
| do_it | −0.5392 | 0.027 | −19.994 | 0.000 | −0.592 | −0.486 |
| refernce | 0.2347 | 0.027 | 8.837 | 0.000 | 0.183 | 0.287 |
| art | 1.1556 | 0.022 | 52.185 | 0.000 | 1.112 | 1.199 |
| geog | 0.5743 | 0.019 | 30.823 | 0.000 | 0.538 | 0.611 |

### Child only Model

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | −2.3744 | 0.020 | −119.412 | 0.000 | −2.413 | −2.335 |
| child | 0.0741 | 0.013 | 5.606 | 0.000 | 0.048 | 0.100 |



# When we cannot control the "Confounder/Omitted Variables," what to do? Instrumental Variable (IV)



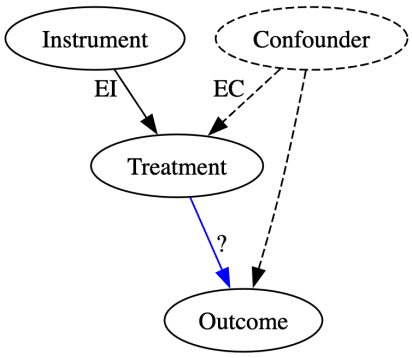

- Isolate the variation in treatment caused by the instrument (EI in the graph)
- Evaluate the *change in outcome* due to the IV's impact on treatment (EI).
- This "change in outcome" is the effect of treatment on the outcome (the blue edge we are interested)
  - Outcome ~ Treatment + Confounder (unobserved)
    - The two are correlated
  - Outcome ~ Treatment changes caused by IV + Confounder (unobserved)
    - The two are uncorrelated (by design or by assumption)



# Requirements of a Valid IV

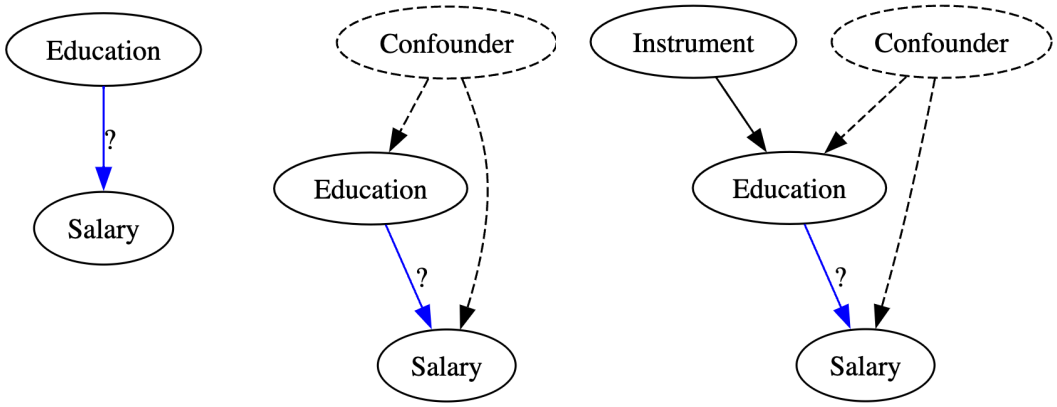
## Key assumptions

- 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
- "Exogeneity" and "Exclusion Restriction" cannot be directly tests (some indirect tests), especially "Exogeneity"
  - Implications: Good IVs are hard to come by



# Let's formalize this idea with one of the most classical examples of instrumental variable

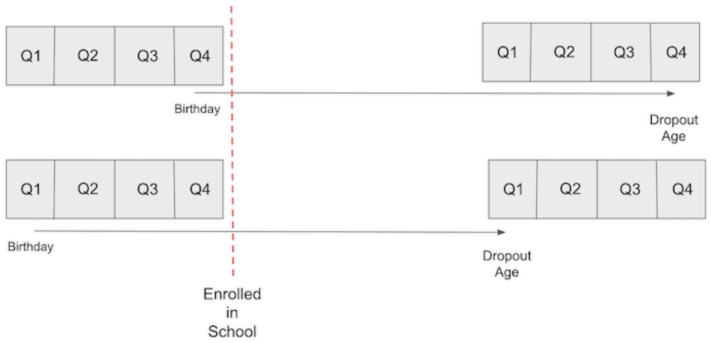
### Angrist and Krueger (1991)






# IV: Birth Quarter of the Person

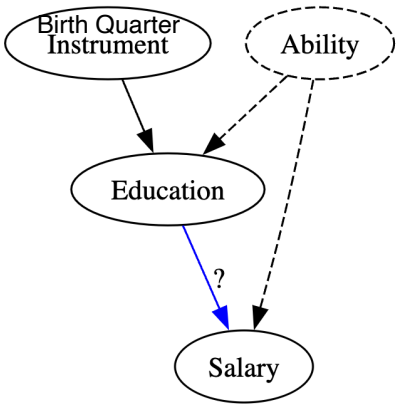
### US School Compulsory Attendance Law




- Must have turned 6 years by Jan 1 of the year they enter school.
- Cannot drop out unless turning 16
- Kids who were born later in a year have more schools than those who were born earlier in a year

# Validity of the IV

- Relevance
  - Birth quarter is strongly correlated duration of schooling
- Exogeneity (or Independence)
  - Birth quarter *seems to* be independent of confounders
- Exclusion Restriction
  - Birth quarter *seems to* not directly affect the outcome
  - Its effect is only through the treatment






# Let's first consider an OLS (biased)

## Baseline

```
### OLS without IV
formula='log_wage ~ years_of_schooling + C(year_of_birth) + C(state_of_birth)'
OLS_model = smf.ols(formula=formula, data=df).fit()
print(OLS_model.summary())
```

| | | | | | | |
|---|---|---|---|---|---|---|
| C(state_of_birth)[T.51.0] | 0.0320 | 0.010 | 3.169 | 0.002 | 0.012 | 0.052 |
| C(state_of_birth)[T.53.0] | 0.1882 | 0.013 | 14.938 | 0.000 | 0.164 | 0.213 |
| C(state_of_birth)[T.54.0] | 0.1275 | 0.010 | 12.172 | 0.000 | 0.107 | 0.148 |
| C(state_of_birth)[T.55.0] | 0.1277 | 0.010 | 13.176 | 0.000 | 0.109 | 0.147 |
| C(state_of_birth)[T.56.0] | 0.1439 | 0.025 | 5.799 | 0.000 | 0.095 | 0.193 |
| years_of_schooling | 0.0673 | 0.000 | 194.382 | 0.000 | 0.067 | 0.068 |

# IV by Hand (ad hoc 2-Stage Least Squared, 2SLS)

## First Stage

- Regress Treatment on IV + other covariates

```
### IV by hand

### First create dummy variables for quarter_of_birth
# Create dummy variables for quarter_of_birth using pd.get_dummies
quarter_dummies = pd.get_dummies(df["quarter_of_birth"].astype(int),
                                 prefix="q", prefix_sep="")
df = pd.concat([df, quarter_dummies], axis=1)
#### First stage
formula = 'years_of_schooling ~ q2 + q3 + q4 + C(year_of_birth) + C(state_of_birth)'
first_stage = smf.ols(formula=formula, data=df).fit()
print(first_stage.summary().tables[1])
```

```
C(state_of_birth)[T.55.0]     1.2584      0.049     25.854      0.000       1.163       1.354
C(state_of_birth)[T.56.0]     1.8870      0.125     15.122      0.000       1.642       2.132
q2                            0.0456      0.016      2.875      0.004       0.015       0.077
q3                            0.1060      0.016      6.824      0.000       0.076       0.136
q4                            0.1526      0.016      9.650      0.000       0.122       0.184
==============================================================================================
```



# IV by Hand (ad hoc 2-Stage Least Squared, 2SLS)

## Second Stage

- Regress Outcome on IV_Predicted_Treatment + other covariates
  - Informally, Treatment = IV_Predicted_Treatment + Treatment Related to Confounder
  - The 2-stage regression only consider the "IV_Predicted_Treatment"

```
### IV by hand
### Second Stage
df['years_of_schooling_fitted'] = first_stage.fittedvalues
## fittedvalues here is essentially the "predicted" treatment level, years_of_schooling,
## based on the first stage model. Alternatively, we can use the following command, the same result
# df['years_of_schooling_fitted'] = first_stage.predict(df)

formula = 'log_wage ~ C(year_of_birth) + C(state_of_birth) + years_of_schooling_fitted'
iv_by_hand = smf.ols(formula=formula, data=df).fit()
print(iv_by_hand.summary().tables[1])
```

```
C(state_of_birth)[T.54.0]     0.1233      0.011     10.966      0.000       0.101       0.145
C(state_of_birth)[T.55.0]     0.0769      0.027      2.812      0.005       0.023       0.131
C(state_of_birth)[T.56.0]     0.0678      0.046      1.466      0.143      -0.023       0.158
years_of_schooling_fitted     0.1077      0.020      5.333      0.000       0.068       0.147
==============================================================================================
```

# IV using the "linearmodels" library

## More convenient and compute SE properly

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

formula = 'log_wage ~ C(year_of_birth) + C(state_of_birth) + [years_of_schooling ~  q2 + q3 + q4]'
iv2sls = IV2SLS.from_formula(formula = formula, data = df).fit()
print(iv2sls.summary.tables[1])
```

```
C(state_of_birth)[T.53.0]     0.1099     0.0401     2.7426     0.0061      0.0314      0.1884
C(state_of_birth)[T.54.0]     0.1233     0.0112     10.964     0.0000      0.1013      0.1454
C(state_of_birth)[T.55.0]     0.0769     0.0267     2.8854     0.0039      0.0247      0.1292
C(state_of_birth)[T.56.0]     0.0678     0.0456     1.4869     0.1370     -0.0216      0.1571
years_of_schooling            0.1077     0.0196     5.5066     0.0000      0.0694      0.1460
==============================================================================================
```



# Check the Assumptions

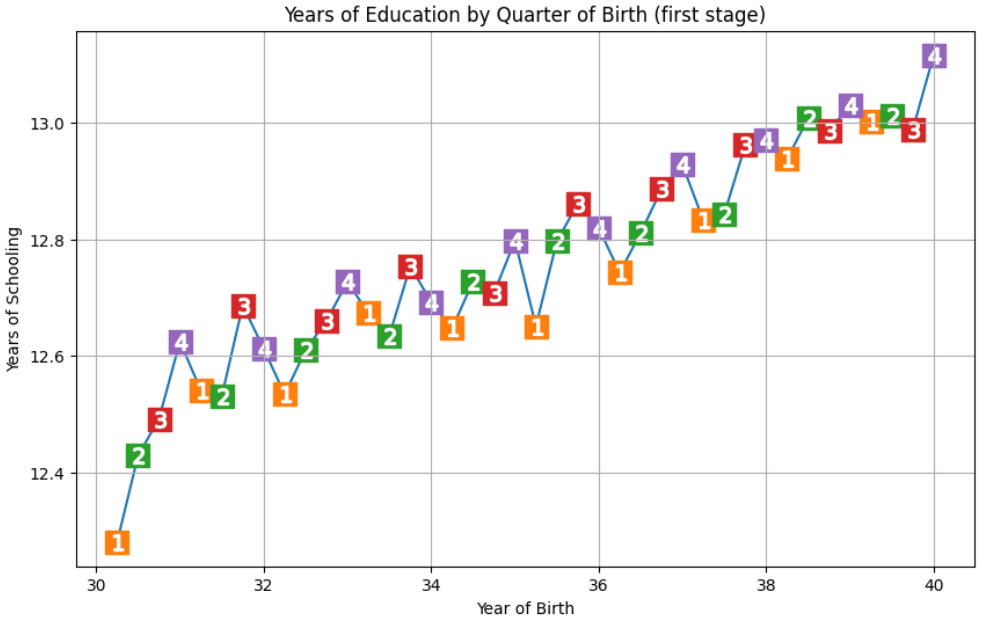
## Relevance: Does the IV strongly correlate with the treatment?

```
### IV by hand

### First create dummy variables for quarter_of_birth
# Create dummy variables for quarter_of_birth using pd.get_dummies
quarter_dummies = pd.get_dummies(df["quarter_of_birth"].astype(int),
                                 prefix="q", prefix_sep="")
df = pd.concat([df, quarter_dummies], axis=1)
df.head()

#### First stage
formula = 'years_of_schooling ~ q2 + q3 + q4 + C(year_of_birth) + C(state_of_birth)'
first_stage = smf.ols(formula=formula, data=df).fit()
print(first_stage.summary().tables[1])
```

```
C(state_of_birth)[T.55.0]     1.2584      0.049     25.854      0.000       1.163       1.354
C(state_of_birth)[T.56.0]     1.8870      0.125     15.122      0.000       1.642       2.132
q2                            0.0456      0.016      2.875      0.004       0.015       0.077
q3                            0.1060      0.016      6.824      0.000       0.076       0.136
q4                            0.1526      0.016      9.650      0.000       0.122       0.184
==============================================================================================
```

# IV Relevance

## Birth quarter is strongly correlated duration of schooling






# IV Relevance

## *First-stage F-stat* (the most formal test)

- Intuition: For the first stage regression, does including the instruments significantly improve the model's fit? Calculate an F-stat calculated using R2 with and without the instruments

```
## 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:
108.20
Instruments are not weak
```

# Testing for Exclusion Restriction and Exogeneity of IV

### Sargan overidentification test (also called J-stat)

- Only work if we have more IVs than endogeneous variable
  - In this example, we have three IVs (q2, q3, q4), and one endogeneous variable (years_of_schooling)
- Intuition:
  - If all the IVs are valid, we should have the same estimate of years_of_schooling after removing any one IV.
  - So, we iteratively drop each one of the IVs and check if each time the coefficient remains the same
  - H0: All instruments are valid
    - So a p-val>0.05, we accept H0—confirms the validity of IVs



# Sargan Overidentification Test

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

## Overidentification test (Sargan-Hansen J test)
### Intuition: If all the IVs are valid, they should give the same coefficient estimates
### If some of them are not valid, the coefficient estimates will be different
### The Sargan-Hansen J test is to test whether the coefficient estimates are the same.
### It iteratively drops one IV at a time and see if the coefficient estimates are the same
### The null hypothesis is that all instruments are valid
### Since we have 3 IVs and 1 endogenous variable, we can perform this overidentification test
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: 3.0652
P-value: 0.2160
Distributed: chi2(2)
```

# When there are the same number for IVs and Endogenous Variables—No available test for IV exogeneity

### An imperfect test for exclusion restriction

- Y ~ X_fitted + *First_stage_residuals* + other control covariates
  - If *First_stage_residuals* is significant, the IV may violate exclusion restriction
- Intuition: If X_fitted cannot fully account for Instrument's effect on Y, it implies
  - (1) Instruments may have some direct effect on Y, and/or
  - (2) Instruments are weak so it does not measure X_fitted well
- But if first-stage F-stat shows the IVs are not weak, it is consistent with the first explanation.