

R 使い方入門：操作変数法

操作変数法 (Instrumental Variables Method)

回帰分析において、説明変数と誤差項が相關している場合（内生性がある場合）、通常の最小二乗法（OLS）による推定量はバイアスを持ち、一致性を失います。教育年数が賃金に与える影響を推定するミンサー方程式の例では、個人の「能力」などの観察できない要因が、教育年数と賃金の両方に影響を与える可能性があります。この場合、教育年数の係数は過大評価（上方バイアス）される傾向があります。

このような内生性の問題に対処するために用いられるのが操作変数法（IV 法）です。

データの読み込み

まず、データを読み込みます。ここでは `8_income.csv` を使用します。主な変数は以下の通りです。

- `lincome`: 賃金の対数値
- `yeduc`: 教育年数
- `exper`: 経験年数
- `payeduc`: 親の教育費支払い能力（操作変数）
- `sibs`: 兄弟姉妹数（操作変数）

```
income8 <- read.csv("8_income.csv")
head(income8)

#>   exper exper2 mbirth moyeduc payeduc sibs yeduc lincome
#> 1    19     361      6      9      9     1    9 5.703783
#> 2    19     361      4     12     12     1    9 5.991465
#> 3    19     361      1      9      9     1    9 6.214608
#> 4    19     361      9      9      9     1    9 6.214608
#> 5    23     529      8      9      9     3    9 6.309918
#> 6    23     529     11     12     13     2    9 6.309918
```

OLS による推定（ベースライン）

比較のために、まずは通常の最小二乗法（OLS）でミンサー方程式を推定します。

```

reg1 <- lm(lincome ~ yeduc + exper + exper2, data = income8)
summary(reg1)

#>
#> Call:
#> lm(formula = lincome ~ yeduc + exper + exper2, data = income8)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -1.36671 -0.15058 -0.00815  0.15856  1.25601
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 4.3197739  0.1387946 31.124 < 2e-16 ***
#> yeduc        0.0842350  0.0062404 13.498 < 2e-16 ***
#> exper         0.0576637  0.0152888  3.772 0.000175 ***
#> exper2       -0.0008313  0.0005259 -1.581 0.114371
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.3092 on 730 degrees of freedom
#> Multiple R-squared:  0.2487, Adjusted R-squared:  0.2456
#> F-statistic: 80.53 on 3 and 730 DF,  p-value: < 2.2e-16

```

`summary(reg1)` の結果を確認すると、教育年数 `yeduc` の係数は正で有意であることがわかります。しかし、前述の通り、この推定値には内生性によるバイアスが含まれている可能性があります。

操作変数法による推定

次に、操作変数法を用いて推定を行います。ここでは、本人の教育年数 `yeduc` の操作変数として、親の教育年数（あるいは学費の支払い能力などに関連する変数と想定される）`payeduc` を使用します。

操作変数が満たすべき条件は以下の 2 つです：

1. **関連性 (Relevance)**：操作変数 (`payeduc`) は内生変数 (`yeduc`) と相関していること。
2. **外生性 (Exogeneity)**：操作変数 (`payeduc`) は誤差項とは相関していないこと（賃金に対して、教育年数を通じた経路以外で直接的な影響を持たないこと）。

R で操作変数法を行うには、AER パッケージの `ivreg()` 関数が便利です。

```

library(AER)
# 操作変数法による推定
# 式の書き方: 従属変数 ~ 内生変数 + 外生変数 | 操作変数 + 外生変数

```

```

ivreg1 <- ivreg(lincome ~ yeduc + exper + exper2 +
  exper + exper2 + payeduc, data = income8)
summary(ivreg1)

#>
#> Call:
#> ivreg(formula = lincome ~ yeduc + exper + exper2 | exper + exper2 +
#>     payeduc, data = income8)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -1.350328 -0.146688 -0.009026  0.162186  1.236270
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 4.4485971  0.3489064 12.750 < 2e-16 ***
#> yeduc        0.0752049  0.0232869  3.230 0.001296 **
#> exper         0.0597410  0.0161570  3.698 0.000234 ***
#> exper2       -0.0009639  0.0006213 -1.552 0.121191
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.3097 on 730 degrees of freedom
#> Multiple R-Squared: 0.2465, Adjusted R-squared: 0.2434
#> Wald test: 23.22 on 3 and 730 DF, p-value: 2.31e-14

```

`ivreg()` の式の | の右側には、**モデルに含まれるすべての外生変数**（ここでは `exper`, `exper2`）と、**操作変数**（`payeduc`）を記述します。内生変数である `yeduc` は右側には含めません。

結果の解釈

OLS の結果（`reg1`）と IV 法の結果（`ivreg1`）を比較してみましょう。

- **OLS 推定値（`yeduc`）**：約 0.084
- **IV 推定値（`yeduc`）**：約 0.075

OLS による教育年数の収益率（約 8.4%）と比較して、操作変数法による推定値（約 7.5%）は小さくなりました。これは、OLS 推定値が「能力」などの観察されない要因によって上方バイアスを持っていた（過大評価されていた）という仮説と整合的です。能力が高い人は教育年数が長く、かつ賃金も高い傾向があるため、能力をコントロールできない OLS では教育の効果が実際よりも大きく推定されてしまう傾向があります。操作変数法を用いることで、このバイアスを取り除き、より純粋な教育の効果に近づくことができました。

ただし、IV 推定値も依然として統計的に有意であり、教育が賃金に正の効果を与えるという結論自体は変わりません。

複数の操作変数（過剰識別）

次に、操作変数を追加して分析を行います。ここでは、`payeduc`（親の教育費支払い能力）に加えて、`sibs`（兄弟姉妹数）も操作変数として使用します。兄弟姉妹数が多いと、一人当たりの教育資源が減少し、教育年数に影響を与える可能性がありますが、本人の賃金には直接影響しない（外生性がある）と仮定します。

ivreg による推定

`ivreg()` 関数の `|` の右側に `sibs` を追加します。

```
# 兄弟姉妹数 (sibs) も操作変数として追加
ivreg2 <- ivreg(lincome ~ yeduc + exper + exper2 |
  exper + exper2 + payeduc + sibs, data = income8)
summary(ivreg2)

#>
#> Call:
#> ivreg(formula = lincome ~ yeduc + exper + exper2 | exper + exper2 +
#>     payeduc + sibs, data = income8)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -1.340720 -0.153748 -0.009586  0.163970  1.224695
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 4.5241448  0.3286803 13.765 < 2e-16 ***
#> yeduc        0.0699093  0.0217875  3.209 0.001392 **
#> exper        0.0609592  0.0160773  3.792 0.000162 ***
#> exper2       -0.0010417  0.0006104 -1.707 0.088290 .
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.3103 on 730 degrees of freedom
#> Multiple R-Squared: 0.2432, Adjusted R-squared: 0.2401
#> Wald test: 23.09 on 3 and 730 DF, p-value: 2.753e-14
```

2段階最小二乗法（2SLS）の手動計算

操作変数法は「2段階最小二乗法（2SLS）」とも呼ばれます。この仕組みを理解するために、`lm()` 関数を使って手動で計算してみましょう。

1. 第1段階: 内生変数 (`yeduc`) を、すべての外生変数 (`exper`, `exper2`, `payeduc`, `sibs`) に回帰させ、予測値 (`fitted values`) を求めます。
2. 第2段階: 本来の回帰式の内生変数 (`yeduc`) を、第1段階で求めた予測値に置き換えて回帰を行います。

```
# 第1段階: 内生変数 (yeduc) をすべての外生変数に回帰
ols1 <- lm(yeduc ~ exper + exper2 + payeduc + sibs, data = income8)

# 第2段階: yeduc の予測値 (fitted(ols1)) を使って本回帰
ols2 <- lm(lincome ~ fitted(ols1) + exper + exper2, data = income8)
summary(ols2)

#>
#> Call:
#> lm(formula = lincome ~ fitted(ols1) + exper + exper2, data = income8)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -1.1985 -0.2114  0.0001  0.1888  1.1450
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 4.524145  0.364011 12.429 < 2e-16 ***
#> fitted(ols1) 0.069909  0.024130  2.897 0.003877 **
#> exper        0.060959  0.017805  3.424 0.000652 ***
#> exper2       -0.001042  0.000676 -1.541 0.123722
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.3437 on 730 degrees of freedom
#> Multiple R-squared:  0.0718, Adjusted R-squared:  0.06799
#> F-statistic: 18.82 on 3 and 730 DF,  p-value: 9.08e-12
```

手動計算による `fitted(ols1)` の係数は、`ivreg2` の `yeduc` の係数と一致することを確認してください。これにより、操作変数法が内生変数のうち、外生変数によって説明できる部分（予測値）のみを使って回帰を行っていることがわかります。ただし、標準誤差は異なっているために検定に使えません。

結果の解釈

複数の操作変数を用いた結果（ivreg2）を確認しましょう。

- **IV 推定値 (yeduc)**：約 0.070

単一の操作変数を用いた場合（約 0.075）と比較して、推定値はさらに少し小さくなりましたが、大きな違いはありません。また、統計的有意性も維持されています（p 値 < 0.01）。複数の操作変数を用いることで、推定の効率性が向上する可能性がありますが、同時に過剰識別検定（Sargan 検定など）を行うことで操作変数の妥当性をチェックすることも可能になります（後述の「診断統計量」の節で詳しく扱います）。兄弟姉妹数を操作変数に追加しても、教育の収益率が約 7% 程度であるという結論は頑健であることが示唆されます。

ivreg の拡張機能

ivreg 関数には、多くの拡張機能があります。

診断統計量

summary() 関数に diagnostics = TRUE オプションを指定することで、操作変数の妥当性を評価するための診断統計量を確認できます。これには、弱操作変数検定（Weak Instrument Test）、Wu-Hausman（ハウスマン検定）、過剰識別検定（Sargan Test）などが含まれます。弱操作変数検定は、操作変数が内生変数と十分に相関しているか（関連性条件）を評価し、過剰識別検定は、複数の操作変数がある場合に、それらがすべて外生性条件を満たしているか（誤差項と相関がないか）を評価します。

```
summary(ivreg2, diagnostics = TRUE)
#>
#> Call:
#> ivreg(formula = lincome ~ yeduc + exper + exper2 | exper + exper2 +
#>       payeduc + sibs, data = income8)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -1.340720 -0.153748 -0.009586  0.163970  1.224695
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 4.5241448  0.3286803 13.765 < 2e-16 ***
#> yeduc        0.0699093  0.0217875  3.209 0.001392 **
#> exper         0.0609592  0.0160773  3.792 0.000162 ***
#> exper2       -0.0010417  0.0006104 -1.707 0.088290 .
#>
```

```

#> Diagnostic tests:
#>
#>             df1 df2 statistic p-value
#> Weak instruments   2 729     32.831 2.23e-14 ***
#> Wu-Hausman        1 729      0.474    0.491
#> Sargan            1  NA      0.403    0.525
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.3103 on 730 degrees of freedom
#> Multiple R-Squared: 0.2432, Adjusted R-squared: 0.2401
#> Wald test: 23.09 on 3 and 730 DF, p-value: 2.753e-14

```

診断結果の解釈は以下の通りです。

- **Weak instruments (弱操作変数検定)** : p 値が非常に小さく (< 0.01)、帰無仮説（操作変数が弱い）は棄却されます。したがって、操作変数は内生変数と十分に強い相関を持っていると言えます。
- **Wu-Hausman (ハウスマン検定)** : p 値は約 0.491 であり、帰無仮説 (OLS と IV の推定値に差がない=内生性がない) を棄却できません。これは、統計的には OLS 推定値と IV 推定値に有意な差が見られないことを示唆しています（ただし、理論的に内生性が疑われる場合は IV 法を用いることが推奨されることもあります）。
- **Sargan (過剰識別検定)** : p 値は約 0.525 であり、帰無仮説（すべての操作変数が外生である）を棄却できません。これは、使用した操作変数 (payeduc と sibs) が外生性条件を満たしているという仮定と矛盾しないことを示しています。

なお過剰識別検定は操作変数の数が内生変数の数より厳密に多くなければ計算されません。

ロバスト標準誤差

回帰分析において、誤差項の分散が均一でない **不均一分散** (heteroskedasticity) が存在する場合、通常の標準誤差は過小評価される傾向があり、統計的推論が誤る可能性があります。このような場合、`summary()` 関数に `vcov` オプションを指定し、`vcovHC()` 関数 (`sandwich` パッケージ由来) を用いて**ロバスト標準誤差** (robust standard errors) を計算することで、不均一分散に頑健な標準誤差を得ることができます。`type = "HC1"` は、ロバスト標準誤差の計算方法の一つを指定しています。

```

summary(ivreg2, vcov = vcovHC(ivreg2, type = "HC1"))

#>
#> Call:
#> ivreg(formula = lincome ~ yeduc + exper + exper2 | exper + exper2 +
#>       payeduc + sibs, data = income8)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max

```

```

#> -1.340720 -0.153748 -0.009586  0.163970  1.224695
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 4.5241448  0.3252807 13.908 < 2e-16 ***
#> yeduc        0.0699093  0.0218722  3.196 0.001452 **
#> exper         0.0609592  0.0156393  3.898 0.000106 ***
#> exper2       -0.0010417  0.0005904 -1.764 0.078095 .
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.3103 on 730 degrees of freedom
#> Multiple R-Squared: 0.2432, Adjusted R-squared: 0.2401
#> Wald test: 25.21 on 3 and 730 DF, p-value: 1.577e-15

```

ロバスト標準誤差を用いた場合でも、yeduc の係数は有意 ($p < 0.01$) であり、教育の効果が統計的に確からしいことが支持されます。通常の標準誤差と比較して、標準誤差の値がどのように変化しているかを確認することも重要です（ここでは大きな変化は見られません）。

なお、`diagnostics = TRUE` を同時に指定することで、診断統計量を含めて確認できますが、その場合、通常の標準誤差の結果となることに注意されたい。

estimatr パッケージ

`estimatr` パッケージは、`ivreg` 関数を用いて操作変数法を実行する際に、より簡単に診断統計量を確認することができます。

```

library(estimatr)
ivreg3 <- iv_robust(lincome ~ yeduc + exper + exper2 | exper + exper2 + payeduc + sibs, data = income8,
summary(ivreg3)

#>
#> Call:
#> iv_robust(formula = lincome ~ yeduc + exper + exper2 | exper +
#>     exper2 + payeduc + sibs, data = income8, se_type = "HC1",
#>     diagnostics = TRUE)
#>
#> Standard error type: HC1
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)    CI Lower    CI Upper   DF

```

```

#> (Intercept) 4.524145 0.3252807 13.908 3.510e-39 3.885548 5.1627420 730
#> yeduc 0.069909 0.0218722 3.196 1.452e-03 0.026969 0.1128492 730
#> exper 0.060959 0.0156393 3.898 1.060e-04 0.030256 0.0916626 730
#> exper2 -0.001042 0.0005904 -1.764 7.809e-02 -0.002201 0.0001174 730
#>
#> Multiple R-squared: 0.2432 , Adjusted R-squared: 0.2401
#> F-statistic: 25.21 on 3 and 730 DF, p-value: 1.577e-15
#>
#> Diagnostics:
#>          numdf dendf value p.value
#> Weak instruments 2 729 34.692 4.05e-15 ***
#> Wu-Hausman 1 729 0.487 0.485
#> Overidentifying 1 NA 0.484 0.487
#> ---
#> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

`summary(ivreg3)` の結果を確認すると、Standard error type: HC1 と表示されており、ロバスト標準誤差が使用されていることがわかります。また、`yeduc` の係数 (Estimate) は約 0.0699 であり、標準誤差 (Std. Error) は約 0.0219 となっています。これは、先ほど `ivreg` と `vcovHC` を組み合わせて計算した結果と一致しています。

さらに、`Diagnostics` セクションには、弱操作変数検定 (Weak instruments)、Wu-Hausman 検定、過剰識別検定 (Overidentifying) の結果も表示されています。これらも `ivreg` で `diagnostics = TRUE` を指定した際の結果と整合的ですが、ここではロバスト標準誤差を考慮した上で計算されている点に注意が必要です（ただし、この例では診断統計量の値も大きくは変わっていません）。

`estimatr` パッケージを使うと、このように簡潔なコードでロバスト標準誤差と診断統計量を同時に得ることができるため、実務的にも非常に便利です。

なお、`se_type = "classical"` を指定することで、通常の標準誤差を計算することができます。

```

library(estimatr)
ivreg4 <- iv_robust(lincome ~ yeduc + exper + exper2 | exper + exper2 + payeduc + sibs, data = incom
summary(ivreg4)
#>
#> Call:
#> iv_robust(formula = lincome ~ yeduc + exper + exper2 | exper +
#>     exper2 + payeduc + sibs, data = income8, se_type = "classical",
#>     diagnostics = TRUE)
#>
#> Standard error type: classical

```

```

#>
#> Coefficients:
#>
#>             Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
#> (Intercept) 4.524145  0.3286803 13.765 1.712e-38 3.87887 5.1694162 730
#> yeduc        0.069909  0.0217875  3.209 1.392e-03  0.02714 0.1126830 730
#> exper         0.060959  0.0160773  3.792 1.620e-04  0.02940 0.0925226 730
#> exper2       -0.001042  0.0006104 -1.707 8.829e-02 -0.00224 0.0001565 730
#>
#> Multiple R-squared:  0.2432 ,   Adjusted R-squared:  0.2401
#> F-statistic: 23.09 on 3 and 730 DF,  p-value: 2.753e-14
#>
#> Diagnostics:
#>
#>             numdf dendf  value p.value
#> Weak instruments      2    729 32.831 2.23e-14 ***
#> Wu-Hausman            1    729  0.474   0.491
#> Overidentifying        1     NA  0.403   0.525
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

この結果は、`ivreg` 関数で `diagnostics = TRUE` を指定した場合（ロバスト標準誤差を使わない場合）の結果と一致します。