操作変数法２：同時性と需要供給分析への応用

講師：遠山祐太

最終更新：2024-12-01

はじめに

同時性

- 同時性 (simultaneity)：被説明変数と説明変数が同時に決定される。
- 授業の流れ：
  1. 同時性と操作変数による解決方法：需要・供給推定の例
  2. Rでの実践例：需要関数の推定
  3. 不完全競争への拡張

同時性

同時性（逆因果）

- **説明変数と被説明変数が同時に決定される。**
- 需要・供給曲線を例に考えていこう。

需要・供給曲線

- 市場 i における需要曲線

$$q_i^d = \beta_0^d + \beta_1^d p_i + \beta_2^d I_i + u_i^d$$

  - q_i^d : 需要量, p_i: 価格、
  - I_i :平均所得(需要要因), u_i^d :観察されない需要ショック(誤差項)

- 供給曲線

$$q_i^s = \beta_0^s + \beta_1^s p_i + \beta_2^s z_i + u_i^s$$

  - z_i : 原材料価格 (費用要因), u_i^s :供給ショック(誤差項)
  - 完全競争のFOC $p_i = mc_i$ を考え、 $mc_i = \pi_0 + \pi_1 q_i^s + \pi_2 z_i + u_i$ を変形すると上のような供給曲線が得られる。

市場均衡

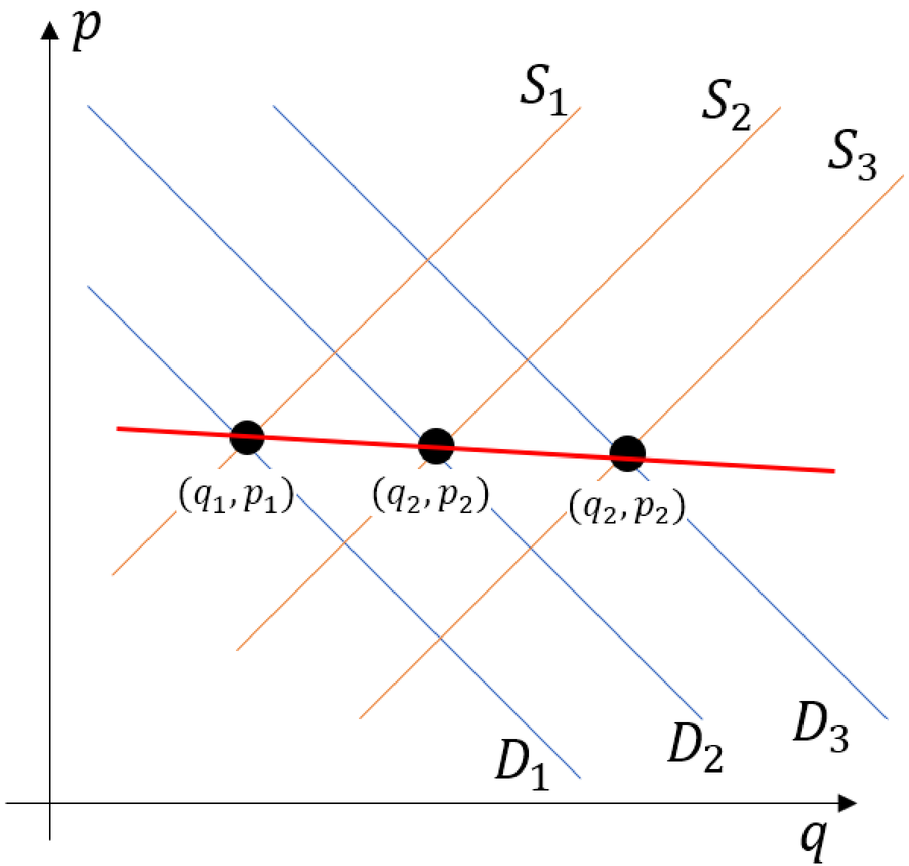
- 市場均衡における価格と数量は、 $q_i^d = q_i^s$ (需給の一致)により決定される。
- データ： $\{q_i, p_i, I_i, z_i\}_{i=1}^N$
- この例では均衡価格が

$$p_i = \frac{(\beta_2^s z_i - \beta_2^d I_i) + (\beta_0^s - \beta_0^d) + (u_i^s - u_i^d)}{\beta_1^d - \beta_1^s}$$

となり、価格と誤差項は必ず相関する。

図示：需要と供給は分離できるだろうか

- ポイント：観測されるデータ点は需要曲線と供給曲線の交点である。



操作変数による解決

- 需要関数から考えよう

$$q_i^d = \beta_0^d + \beta_1^d p_i + \beta_2^d I_i + u_i^d$$

- 需要関数の推定のためには、 z_i を操作変数として用いる。
- 関連性：z_i が費用要因である限りOK。
- 独立性： z_i が需要要因と独立であればOK。

識別の直観：図示

- 費用要因 z_i によって引き起こされる 価格 p_i の変化に着目すれば、背後にある需要曲線を浮き彫り出すことができる。

需要曲線の推定

はじめに：パッケージの読み込み

```
library(summarytools)
library(tidyverse)
library(fixest)
library(AER)
```

例：煙草の需要推定

- 煙草の需要推定
  - 喫煙は多くの病気や負の外部性へとつながる。
  - たばこ税をサポートする一つの考え方

- 問い：煙草の消費を一定量削減するために、どれだけ課税すればよいか？
  - **煙草需要の価格弾力性**を知る必要がある。

データ：米国の州レベルパネルデータ(1985・1995年)

- `AER` パッケージの `CigarrettesSW` データセットを用いる。

```r
# データセットの読み込みと概観
data("CigarettesSW")
desc_stat <- descr(CigarettesSW,
                   stats = c("mean", "sd", "min", "med", "max"),
                   transpose = T)
```

記述統計

```r
print(desc_stat)
```

```
## Non-numerical variable(s) ignored: state, year

## Descriptive Statistics
## CigarettesSW
## N: 96
##
##                          Mean        Std.Dev         Min        Median           Max
## ---------------- ------------- --------------- ------------ ------------- --------------
##              cpi          1.30            0.23         1.08          1.30           1.52
##           income   99878735.74    120541138.18   6887097.00   61661644.00   771470144.00
##            packs        109.18           25.87        49.27        110.16         197.99
##       population    5168866.32      5442344.66    478447.00    3697471.50    31493524.00
##            price        143.45           43.89        84.97        137.72         240.85
##              tax         42.68           16.14        18.00         37.00          99.00
##             taxs         48.33           19.33        21.27         41.05         112.63
```

需要モデル

- 次のモデルを考える。

$$\log(Q_{it}) = \beta_0 + \beta_1 \log(P_{it}) + \beta_2 \log(income_{it}) + u_i + e_{it}$$

  - Q_{it} ：州 i の t 期における、一人当たりの箱の量
  - P_{it} ：一箱あたりの税込価格
  - $income_{it}$ ：一人当たりの所得（需要を変化させる）

- 価格の操作変数として、次を用いる。
  - $SalesTax_{it}$ ：煙草にかかる税のうち消費税の割合
  - $CigTax_{it}$ ：煙草に特別にかかる税
- 関連性：税込価格に含まれる
- 外生性：需要に直接影響しない（価格を通じて間接的に関わる）

データの抽出

```
Cigdata <- CigarettesSW %>%
    mutate( rincome = (income / population) / cpi) %>%
    mutate( rprice  = price / cpi ) %>%
    mutate( salestax = (taxs - tax) / cpi ) %>%
    mutate( cigtax = tax/cpi )
```

回帰の実行 (固定効果なし)

```
# 最小二乗法
cig_OLS <- feols(log(packs) ~ log(rprice) + log(rincome) | 0,
                 se = "hetero",
                 data = Cigdata)

# 操作変数法
cig_IV <- feols(log(packs) ~ log(rincome) | 0 | log(rprice) ~ salestax + cigtax,
                se = "hetero",
                data = Cigdata)

# 変数名を揃える
rownames(cig_IV$coeftable)[2] <- "log(rprice)"

# 結果の表示
cigres <- list()
cigres[["OLS"]] <- cig_OLS
cigres[["IV"]] <-cig_IV

result_cigres <- etable(cigres,
                        signifCode = NA,
                        fitstat = c("r2", "n", "ivwald" ),
                        digits = 2,
                        digits.stats = 2)
```

推定結果

```
print(result_cigres)
```

```
##                                              OLS              IV
## Dependent Var.:                       log(packs)      log(packs)
##
## Constant                       10.1*** (0.50)  9.7*** (0.51)
## log(rprice)                    -1.3*** (0.15) -1.2*** (0.15)
## log(rincome)                     0.32* (0.15)   0.26. (0.15)
## ______________________________ ______________ ______________
## S.E. type                      Heterosk.-rob. Heterosk.-rob.
## R2                                       0.55           0.55
## Observations                               96             96
## Wald (1st stage), log(rprice)              --          139.4
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1st stage

```r
# F 検定
fitstat(cig_IV, "ivwald")
```

```
## Wald (1st stage), log(rprice): stat = 139.4, p < 2.2e-16, on 2 and 92 DoF, VCOV: Heteroskedasticity-rob
```

固定効果ありバージョン

```r
# 最小二乗法
result_1 <- feols(log(packs) ~ log(rprice) + log(rincome) | 0,
                  cluster = "state",
                  data = Cigdata)

# 州の固定効果
result_2 <- feols(log(packs) ~ log(rprice) + log(rincome) | state,
                  cluster = "state",
                  data = Cigdata)

# 固定効果なし操作変数法
result_3 <- feols(log(packs) ~ log(rincome) | 0 | log(rprice) ~ salestax + cigtax,
                  cluster = "state",
                  data = Cigdata)

# 固定効果あり操作変数法
result_4 <- feols(log(packs) ~ log(rincome) | state | log(rprice) ~ salestax + cigtax,
                  cluster = "state",
                  data = Cigdata)
```

```r
# 名前を揃える
models <- list()
models[["OLS"]] <- result_1
models[["state FE"]] <- result_2
models[["IV & no FE"]] <- result_3
models[["IV & FE"]] <- result_4

result_cig_panel <- etable(models,
                           signifCode = NA,
                           fitstat = c("r2", "n", "ivwald" ),
                           digits = 2,
                           digits.stats = 2)
```

推定結果

```
print(result_cig_panel)
```

```
##                                         OLS       state FE       IV & no FE
## Dependent Var.:                 log(packs)     log(packs)       log(packs)
##
## Constant                    10.1*** (0.46)                 9.7*** (0.56)
## log(rprice)                 -1.3*** (0.17) -1.2*** (0.14) -1.2*** (0.18)
## log(rincome)                   0.32 (0.21)    0.12 (0.22)    0.26 (0.20)
## Fixed-Effects:              -------------- -------------- --------------
## state                                   No            Yes             No
## _____________________________ ______________ ______________ ______________
## S.E.: Clustered                  by: state      by: state      by: state
## R2                                    0.55           0.97           0.55
## Observations                            96             96             96
## Wald (1st stage), log(rprice)           --             --          237.1
##
##                                     IV & FE
## Dependent Var.:                 log(packs)
##
## Constant
## log(rprice)                 -1.3*** (0.16)
## log(rincome)                   0.20 (0.24)
```

需要・供給分析の拡張

需要・供給分析の拡張

- 先に導入した需要・供給分析は以下の2つを仮定
  - 同質財 -> 差別化財は今後の授業で(離散選択モデル)
  - 完全競争

- 不完全競争を考慮したケースへの拡張を考えよう。

- 特に、クールノーモデルを用いた実証分析を考える。
  - モデルの導入
  - 需要関数の推定
  - 限界費用の推定

クールノーモデル(の復習)

- N 企業存在する市場を考える。 企業のインデックスを i とする。
- 企業 i の利潤

$$\pi_i = P(Q)q_i - C_i(q_i)$$

  - $P(Q)$: 逆需要関数、 $Q = \sum_{i=1}^{N} q_i$
  - $C_i(q_i)$: 総費用, q_i : 生産量
- ナッシュ均衡を考えよう。

一階条件とナッシュ均衡

- 各企業の最適生産量に関する一階条件は

$$P(Q) - MC_i(q_i) + q_i \frac{\partial P}{\partial Q} = 0$$

  - この式を q_i について解くことで、各企業の最適反応が求められる。

- 一階条件を N 企業について連立させて $\{q_i\}_{i=1}^{N}$ を解くことでナッシュ均衡の生産量が求まる。
- 解き方については後述（数値計算）

推定

- 推定したいもの：
  - 需要関数 $Q(P)$ (もしくは逆需要関数 $P(Q)$)
  - 費用関数 $C_i(q_i)$ ないし、 限界費用関数 $MC_i(q_i)$

- 需要関数の推定は前と同様。
  - 操作変数が必要。 費用関数に入るであろう費用要因をIVとして用いる。

- 費用関数の推定は 一階条件を用いる。

限界費用関数の推定

- 一階条件を書き換えると

$$MC_i = P(Q) + q_i \frac{\partial P}{\partial Q}$$

  - 右辺については、需要関数を推定すれば、全て計算可能。

- 需要関数の情報に基づいて、「実現した限界費用の値」が推定できる。

- もし限界費用が一定(収穫一定の技術)を仮定すればこれで終わり。

- 限界費用が生産量に応じて変化するのであれば、限界費用関数を推定する。

$$MC_i(q_i) = \gamma_0 + \gamma_1 q_i + \gamma_2 z_i + u_i^s$$

  - 注：要操作変数

推定したモデルを用いたシミュレーション分析

- クールノーモデルの要素(需要・供給)が揃うと、様々なシミュレーション分析が可能！
  - 税金・補助金を導入したらどうなるか？
  - 企業が技術革新をして費用が下がったらどうなるか？
  - 企業が合併したらどうなるか？
- シミュレーション分析のためには、ナッシュ均衡を計算する必要がある。
  - 次のスライド

ナッシュ均衡の数値計算

- 各企業の最適生産量に関する一階条件は

$$P(Q) - MC_i(q_i) + q_i \frac{\partial P(Q)}{\partial Q} = 0$$

であり、これを $i = 1, \ldots, N$ まで並べた連立方程式を $(q_i, \ldots, q_N)$ について解く。

- 解き方の例：
  1. 初期値として解の候補をセットする: $\{q_1^{(0)}, \cdots, q_N^{(1)}\}$
  2. 現在の解の候補から $Q^{(0)} = \sum_{i=1}^N q_i^{(0)}$ を計算し、これを一階条件の Q に代入する。すると、 q_i が求まる。これを $q_i^{(1)}$ と呼ぶ。
  3. 新しく $Q^{(1)}$ を計算し、同じ手順を繰り返す。
  4. 最終的に $\{q_1^{(r)}, \cdots, q_N^{(r)}\}$ が変化しなくなった時点で計算を止める。その点がナッシュ均衡となる。