集計データによる離散選択需要モデル: Berry (1994)

講師：遠山祐太

更新: 2025-01-20

イントロダクション

集計データによる離散選択モデルの推定

- 前回：ロジットモデル
  - 個票データを用いて最尤法で推定
- データの制約：**消費者の個票データ**が常に使えるとは限らない。
- 今回：**集計データのみ(マーケットシェア)**が利用可能な状況における離散選択モデルの推定
  - 産業組織論の実証研究においてメジャーなツール
  - 同時に、**価格の内生性問題**についても取り組む。

今日のプランと参考文献

- プラン
  1. Berry (1994)によるロジットモデルの推定
  2. 価格の内生性問題
  3. ロジットモデルからの拡張：ネスト型ロジットモデル
  4. 応用例：自動車需要の推定

- 参考文献
  - 上武・遠山・若森・渡辺 「実証ビジネス・エコノミクス」第２回及び第３回
  - Berry (1994) "Estimating Discrete-Choice Models of Product Differentiation" *Rand Journal of Economics*

多項ロジットモデル

離散選択モデルによる需要モデリング

- 消費者需要モデルは応用ミクロ経済学(特に産業組織)における基本ツール
  - プライシング：価格差別、動学プライシング
  - 企業合併分析・費用転嫁分析 (Pass-through)

- 特に**差別化財**の需要分析に着目
  - 同じカテゴリーにおいて数多くのバラエティが展開されている
  - (参考)同質財：企業間で製品に差がないケース

- 差別化財の需要モデルとして、**離散選択フレームワーク**が用いられる。
  - 消費者は、複数の選択肢(製品)から一つを選ぶ。
  - 製品は、水平的・垂直的に差別化されている。
  - 例：持ち運びやすいが遅いPC・デスクトップの高性能PC

多項ロジットモデル

- マーケット t において、合計で J_t 個の製品が販売されている
- 消費者 i が製品 j を購入した場合の間接効用

$$U_{ijt} = \begin{cases} \beta^0 + \sum_{k=1}^{K} \beta^k x_{jt}^k - \alpha p_{jt} + \xi_{jt} + \epsilon_{ijt} & j = 1, \cdots, J_t \\ \epsilon_{i0t} & j = 0 \end{cases}$$

- p_{jt}: 価格, x_{jt}^k: 製品属性(サイズ、燃費、馬力などなど)
- ξ_{jt}: 観察されない製品属性(誤差項)
- ϵ_{ijt}: 個人・製品レベルの選好へのランダムなショック
- アウトサイド・グッズ $(j = 0)$: -「何も購入しない」という選択.
  - 効用水準をゼロと基準化 (location normalization)

平均効用の定義

- **平均効用**を定義

$$\delta_{jt} = \beta^0 + \sum_{k=1}^{K} \beta^k x_{jt}^k - \alpha p_{jt} + \xi_{jt}$$

- すなわち、

$$U_{ijt} = \delta_{jt} + \epsilon_{ijt}$$

- δ_{jt}は全員に共通な効用部分(平均効用)
- ϵ_{ijt} は個々人で異なる効用の部分
  - 同じ製品であっても、人によって買ったり買わなかったりする。

消費者の購入確率

- 消費者は自分の効用が最も高くなるような選択肢を選ぶ

$$\max_{j \in \{0,1,\cdots,J_t\}} \delta_{jt} + \epsilon_{ijt}$$

- ϵ_{ijt}は個人特有の選好ショック。i.i.d.な第一種極値分布に従うとする。
- ロジット型の購入確率が得られる。

$$\Pr(j \text{ を購入 }) = \frac{\exp\left(\beta^0 + \sum_{k=1}^{K} \beta^k x_{jt}^k - \alpha p_{jt} + \xi_{jt}\right)}{1 + \sum_{l=1}^{J_t} \exp\left(\beta^0 + \sum_{k=1}^{K} \beta^k x_{lt}^k - \alpha p_{lt} + \xi_{lt}\right)}$$

- 注： $\delta_{0t} = 0, exp(0) = 1$

市場シェア

- 市場における潜在的な消費者数(市場サイズとも呼ぶ)を M_t とする。
- 製品 j の需要関数は(サイズ)×(選択確率)となる。

$$q_{jt}\left(p_{1t},\cdots,p_{J_tt}\right)=M_t\frac{\exp\left(\beta^0+\sum_{k=1}^K\beta^k x_{jt}^k-\alpha p_{jt}+\xi_{jt}\right)}{1+\sum_{l=1}^{J_t}\exp\left(\beta^0+\sum_{k=1}^K\beta^k x_{lt}^k-\alpha p_{lt}+\xi_{lt}\right)}$$

- 製品 j の需要は、他の全ての製品の価格・性能に依存している！

価格弾力性

- 差別化財では二種類の価格弾力性
  - **自己価格弾力性**：価格が1%上昇したときに、自身の需要が何%変化するか？
  - **交差価格弾力性**：価格が1%上昇したときに、他の需要が何%変化するか？

- 自己価格弾力性と交差価格弾力性は以下となる。

$$\eta_{jl} = \frac{\partial q_j}{\partial p_l}\frac{p_l}{q_j} = \begin{cases} -\alpha p_j \left(1 - s_j\right) & l = j \\ \alpha p_l s_l & l \neq j \end{cases}$$

Berryのインバージョン

集計データを用いたモデルの推定

- 前回：個票データを用いた推定。
- 仮に、**集計データ**(市場シェア)しかない場合にどうするか？
- このようなケースにおいて、離散選択需要モデルを推定する方法を提示したのが、Steven Berry, Ariel Pakes, そして James Levinsohn たちの1990年後半ー2000年代中盤の一連の研究である。
- 重要論文：
  - Berry (1994, *Rand Journal of Economics*)
  - Berry, Levinsohn, and Pakes (1995, *Econometrica*)
  - サーベイ：*Handbook of IO Volume 4*におけるBerry and Haile (2021)とGandhi and Nevo (2021)

市場シェアの定義

- 以下のように市場シェアをデータから定義する。

$$s_{jt} = \begin{cases} \frac{q_{jt}}{M_t} & j = 1, \cdots, J_t \\ \frac{M_t - \sum_{j=1}^{J_t} q_{jt}}{M_t} & j = 0 \end{cases}$$

- モデルの予測

$$s_{jt} = \begin{cases} \frac{\exp(\delta_{jt})}{1 + \sum_{l=1}^{J_t} \exp(\delta_{lt})} & j = 1, \cdots, J_t \\ \frac{1}{1 + \sum_{l=1}^{J_t} \exp(\delta_{lt})} & j = 0 \end{cases}$$

- 平均効用 $\delta_{jt} = \beta^0 + \sum_{k=1}^{K} \beta^k x_{jt}^k - \alpha p_{jt} + \xi_{jt}$

Berryのインバージョン(逆変換)

- 式変形すると、以下のような線形の式が得られる。

$$\log\left(\frac{s_{jt}}{s_{0t}}\right) = \beta^0 + \sum_{k=1}^{K} \beta^k x_{jt}^k - \alpha p_{jt} + \xi_{jt}$$

  - 右辺：製品から得られる(平均)効用
  - 左辺：市場シェアの対数値

- 直観：マーケットシェアの大小は、その製品からの効用の大小を反映している。

Berryインバージョンの直観的アイデア

- 100世帯の村を考える。
- 3車種が販売：
  - 自動車1: 45世帯
  - 自動車2: 40世帯
  - 自動車3: 10世帯
  - 車を持っていない人：5世帯
- 平均的には「自動車１」から得られる効用が最も高いと考えられる。

続・Berryインバージョン

- 自動車１の平均効用は

$$\delta_1 = \log(0.45/0.05) \approx 2.2$$

- 自動車２と３も同様。
- これらの値を価格や製品属性に回帰することで、これらが効用に与える効果(要するに係数)を推定することができる。

価格の内生性問題と操作変数

内生性が引き起こす問題

- OLS推定すると・・・価格の係数が正になったり・ほぼゼロになったりすることも

$$\log\left(\frac{s_{jt}}{s_{0t}}\right) = \beta^0 + \sum_{k=1}^{K} \beta^k x_{jt}^k - \alpha p_{jt} + \xi_{jt}$$

- 典型的な**内生性**問題！！
  - 価格 p_{jt} と(分析者にとって)観察されない ξ_{jt}の相関
  - 消費者＆企業は、分析者よりも需要について詳しい情報を持っているであろう。
  - 企業は、そのような情報(つまり ξ_{jt})を踏まえて価格を決める！

- OLSで推定すると、価格の係数 $-\alpha$ には上方バイアスがかかる(つまりゼロに近づく)
  - 欠落変数バイアスの公式を思い出そう。

内生性問題へのアプローチ

- アプローチ１：固定効果
  - パネルデータの構造を活用する。
  - 市場 t のダミー変数や、製品 j のダミー変数

- アプローチ２：操作変数
  - 価格 p_{jt} と相関するが、誤差項 ξ_{jt}と相関しない変数
  - 代表的な操作変数が３種類(次のスライド)

1：生産の費用情報

- 製品の生産コストに関する情報
  - 関連性：OKであろう。
  - 独立性：生産コストと需要ショックの間に相関がなければOK

- 実務上は難しい！！
  - 差別化財なので、マーケット t かつ 製品 jレベルの変数。
  - 例えば「鉄鉱石の価格」はマーケット(時間)レベルの変動しか持たない。弱操作変数の懸念。

- うまい例たち
  - Doi and Ohashi (2019, IJIO): あるルートにおける航空券価格への操作変数として、そのルートで運行している航空機の重量に原油価格をかけたもの
  - Hollenbeck and Uetake (2021, RAND): 娯楽用マリファナの小売価格への操作変数として、卸売価格を利用

2：BLP操作変数・差別化操作変数

- 「市場における製品間の競争状況」を捉える変数
- BLP操作変数 (Berry et al 1995 EMA)
  - 1: 同じ企業が生産している他の製品の品質の和 $z_{jt}^{BLP,other} = \sum_{k \in J_{ft}, k \neq j} x_{kt}$
  - 2: 他の企業が生産している製品の品質の和 $z_{jt}^{BLP,rival} \sum_{k \notin J_{ft}} x_{kt}$
- 関連性：市場における製品間の競争度合い。
  - 競争相手企業が多くかつ製品数も多い場合、市場競争はより盛んとなり、価格低下
- 独立性：重要な仮定として「製品品質 x_{jt} は外生的に決まっており ξ_{jt} と相関しない」
  - 製品品質がすぐには変化しないであろう、短期的な状況では妥当。
  - 企業が製品品質の意思決定を内生化するような状況では成り立ちづらいであろう。
- 同様のアイデアを改良したものとして差別化操作変数(Gandhi and Houde 2020)

3：他のマーケットにおける価格

- **Hausman–Nevo操作変数**と呼ばれる(Hausman 1996; Nevo 2001）
- 同一時点での複数の地理的マーケットの情報が必要。
- **他の地理マーケットにおける同じ製品の平均価格**を、自身の価格 p_{jt} への操作変数とする。
- 関連性：共通の費用要因
- 独立性：観察されない需要要因 ξ_{jt} がマーケットをまたいで独立であること。これが成り立つかはケース・バイ・ケース。

ロジットモデルの拡張

ロジットモデルの限界

- ロジットモデルの価格弾力性は

$$\eta_{jl} = \frac{\partial q_j}{\partial p_l}\frac{p_l}{q_j} = \begin{cases} -\alpha p_j \left(1 - s_j\right) & l = j \\ \alpha p_l s_l & l \neq j \end{cases}$$

- 交差弾力性を見ると、製品 l からの交差弾力性は、代替先の財に関わらず一定。
- 言い換えると、プリウスの価格が1%上昇したときに、アクアとカローラへの代替が同じ。
- ロジットモデルには「無関係な選択肢からの独立性」(independence from irrelevant alternatives, IIA)という性質がある。
  - 「青バス・赤バス問題」としても知られている。

ロジットモデルからの拡張

- **ネスト型ロジットモデル** (nested logit model)
  - 製品のグルーピングを考える。
  - Berry ロジットと同様に線形回帰で推定可能
  - 以下で簡単に解説。詳細は「実証ビジエコ」第3回を参照。

- **ランダム係数ロジットモデル** (random coefficient logit model)
  - モデルは前回と同様。
  - Berryロジットのように線形式では推定できず、非線形なモデルとなる。
  - 集計データのみで推定する方法をBerry, Levinsohn and Pakes (1995)で提案。
  - 今回は割愛。「実証ビジエコ」第3回を参照。

ネスト型ロジットモデル (Nested Logit Model)

- 製品に関するグルーピングを考える。
  - 例：ガソリン車・ハイブリッド車

- 需要関数(市場シェア関数)は以下

$$
\begin{aligned}
s_j &= Pr(\text{ group } g) \times \Pr(\text{ choose } j \mid \text{ group } g) \\
&= \frac{D_g^{(1-\sigma)}}{\sum_{g=0}^{G} D_g^{(1-\sigma)}} \times \frac{\exp\left(\frac{\delta_j}{1-\sigma}\right)}{D_g}
\end{aligned}
$$

  ここで $D_g = \sum_{k \in \mathcal{G}_g} \exp\left(\frac{\delta_k}{1-\sigma}\right)$

- グループ内での代替性の強さをパラメタ σ で捉える。

ネスト型ロジットの推定

- ロジットと同様に線形回帰モデルに帰着

$$\ln\left(s_{jt}\right) - \ln(s_{0t}) = \beta X_{jt} + \alpha p_{jt} + \sigma \ln\left(s_{jt/g}\right) + \xi_{jt}$$

- ここで $\ln\left(s_{jt/g}\right)$ はグループ内での製品 j のシェア。内生変数。
- 内生変数 $p_{jt}, s_{jt/g}$ を踏まえて操作変数法で推定を行う。

ネスト型ロジットにおける価格弾力性

- 価格弾力性

$$\frac{\partial s_j}{\partial p_k}\frac{p_k}{s_j} = \begin{cases} -\alpha p_j \left(1 - \sigma s_{j/g} - (1-\sigma)s_j\right)/(1-\sigma), & \text{if } k = j \\ \alpha p_k \left(\sigma s_{k/g} + (1-\sigma)s_k\right)/(1-\sigma), & \text{if } j,k \in g \\ \alpha p_k s_k, & \text{otherwise} \end{cases}$$

- 製品が同じグループに属す($j \in g, k \in g$)とき、交差弾力性が大きくなる(より代替的)
- $\sigma = 0$ のとき、多項ロジットと同様になる。

実証例：自動車市場の需要推定

留意点

- この話に出てくる会社名・製品名などはすべて仮想のものです。

ケース：ある自動車会社の悩み

- あなた：経済学を活用したコンサルティングサービスを提供する「実証ビジネス・エコノミクス社」の新入社員
- お客様：日評自動車株式会社のマーケティング部門の担当者
- マイナーモデルチェンジに合わせて自動車の値下げを検討中

ケース：社内での分析

$$\log q_j = \alpha_0 - \alpha_1 logp_j + \alpha_2 size_j + \alpha_3 fuelefficiency_j + \alpha_4 hppw_j + u_j$$

- $size$：自動車の大きさ
- $fuelefficiency$：燃費
- $hppw$：燃費／重量、
- $price$：自動車価格
- 推定値: $\hat{\alpha}_1 = 1.25.$

(エコノメ) OLS推定

```
rm(list =ls())
library("tidyverse")
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library("fixest")
data_NIPPYO <- readRDS("intermediate/data_auto_NIPPYO.RDS")
data <- readRDS("intermediate/data_auto_all.RDS")

ols_intro <-
  fixest::feols(log_sales ~ log_price + hppw + FuelEfficiency + size, data = data_NIPPYO)
```

推定結果

```
print(result)
```

```
##                          ols_intro
## 定数項                 5.2*** (1.3)
## log(価格)               -0.21 (0.46)
## 馬力／重量             -8.0 (5.6)
## 燃費(キロメートル／ 1 リットル) 0.16*** (0.04)
## サイズ               0.17*** (0.05)
## ------------------ --------------
## S.E. type           Heterosk.-rob.
## R2                            0.24
## Observations                   208
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ケース：自動車市場データの収集

- 自動車のカタログ情報(価格・製品属性)：ウェブサイト「CarView!」
- 販売台数 q_{jt}: 各年かつ車種(例：プリウス)レベルの「自動車の新車登録台数」
- 潜在的な消費者数 M_t ：日本全体の家計数。

ケース：データの記述統計

	平均	標準偏差	25% タイル	メディアン	75% タイル
販売台数	24586.39	40195.27	2958	8544	27404
価格（100 万円）	2.53	1.83	1.42	2.05	2.87
燃費（km / ℓ）	16.16	5.55	12	15.4	19.4
サイズ（立方メートル）	11.51	2.48	9.58	11.47	13.04
馬力 / 重量（ps / kg）	0.1	0.04	0.08	0.09	0.11

注）燃費は、ガソリン 1 ℓ 当たり何km走行するかを示す。サイズは、車高×車長×車幅として定義している。

(エコノメ)Rのコード

```
data %>%
  dplyr::select( Sales, price, FuelEfficiency, size, hppw ) %>%
  summarytools::descr( transpose = TRUE, stats = c("mean", "sd", "q1", "med", "q3"), order = "prese
```

```
## Descriptive Statistics
##
##                          Mean     Std.Dev         Q1     Median          Q3
## -------------------- ---------- ---------- --------- ---------- ----------
##                Sales   24586.39   40195.27   2958.00   8544.00   27404.00
##                price       2.53       1.83      1.42      2.05       2.87
##       FuelEfficiency      16.16       5.55     12.00     15.40      19.40
##                 size      11.51       2.48      9.58     11.47      13.04
##                 hppw       0.10       0.04      0.08      0.09       0.11
```

ロジットモデルの推定結果

	OLS	BLP 操作変数	差別化 操作変数
定数項	−12.3 (0.36)	−12.3 (0.38)	−13.0 (0.39)
自動車価格	−0.26 (0.03)	−0.28 (0.07)	−0.55 (0.08)
馬力 / 重量 (ps / kg)	−0.65 (1.3)	0.21 (2.3)	8.4 (2.6)
燃費（km / ℓ）	0.13 (0.010)	0.13 (0.010)	0.13 (0.010)
サイズ	0.18 (0.02)	0.19 (0.02)	0.24 (0.02)
第1段階における操作変数の F 値		45.8	53
決定係数	0.22	0.22	0.18
観測数	1823	1823	1823

注）カッコ内の数字は、不均一分散について頑健な標準誤差を示している。なお自動車価格の係数は、$-\alpha$ を示している。

(エコノメ) Rでの推定

```
# まず被説明変数を定義する。
data <-
  data %>%
  dplyr::mutate(logit_share = log(share) - log(share0))

# OLSの結果
ols <-
  fixest::feols(logit_share ~ price + hppw + FuelEfficiency + size, data = data)

# BLP操作変数を用いた結果
iv_BLP <-
  fixest::feols(
    logit_share ~ hppw + FuelEfficiency + size | 0 |
      price ~ iv_BLP_own_hppw + iv_BLP_own_FuelEfficiency + iv_BLP_own_size +
      iv_BLP_other_hppw + iv_BLP_other_FuelEfficiency + iv_BLP_other_size,
    data = data
  )
```

```r
# Differentiation IVを用いた結果
iv_GH <-
  fixest::feols(
    logit_share ~ hppw + FuelEfficiency + size | 0 |
      price ~ iv_GH_own_hppw + iv_GH_own_FuelEfficiency + iv_GH_own_size +
      iv_GH_other_hppw + iv_GH_other_FuelEfficiency + iv_GH_other_size,
    data = data
  )

# 推定結果をレポートする。
fixest::etable( list(ols, iv_BLP, iv_GH),
                se = "hetero",
                fitstat = c("r2", "n", "ivf" ) ,
                signifCode = NA,
                dict = c(price = "自動車価格",
                         hppw = "馬力／重量",
                         FuelEfficiency = "燃費(キロメートル／ 1 リットル)",
                         size = "サイズ",
                         `(Intercept)` = "定数項"),
                digits = 2,
                digits.stats = 2,
                depvar = FALSE) -> result
```

```
kableExtra::kable(result)
```

	model 1	model 2	model 3
定数項	-12.3* (0.36)	-12.3* (0.38)	-13.0* (0.39)
自動車価格	-0.26* (0.03)	-0.28* (0.07)	-0.55* (0.08)
馬力／重量	-0.65 (1.3)	0.21 (2.3)	8.4** (2.6)
燃費(キロメートル／ 1 リットル)	0.13* (0.010)	0.13* (0.010)	0.13* (0.010)
サイズ	0.18* (0.02)	0.19* (0.02)	0.24* (0.02)
–	–	–	–
S.E. type	Heteroske.-rob.	Heteroske.-rob.	Heteroske.-rob.
R2	0.22	0.22	0.18
Observations	1,823	1,823	1,823
F-test (1st stage), 自動車価格	--	45.8	53.0

(エコノメ) 1st Stage Regression

```
# BLP操作変数を用いた結果
iv1st_BLP <-
  fixest::feols(
    price ~ hppw + FuelEfficiency + size +
      iv_BLP_own_hppw + iv_BLP_own_FuelEfficiency + iv_BLP_own_size +
      iv_BLP_other_hppw + iv_BLP_other_FuelEfficiency + iv_BLP_other_size,
    data = data
  )

# Differentiation IVを用いた結果
iv1st_GH <-
  fixest::feols(
    price ~ hppw + FuelEfficiency + size +
      iv_GH_own_hppw + iv_GH_own_FuelEfficiency + iv_GH_own_size +
      iv_GH_other_hppw + iv_GH_other_FuelEfficiency + iv_GH_other_size,
    data = data
  )
```

```r
# 推定結果をレポートする。
fixest::etable( list( iv1st_BLP, iv1st_GH),
                se = "hetero",
                fitstat = c("r2", "n") ,
                signifCode = NA,
                dict = c(price = "自動車価格",
                         hppw = "馬力／重量",
                         FuelEfficiency = "燃費(キロメートル／ 1 リットル)",
                         size = "サイズ",
                         `(Intercept)` = "定数項"),
                digits = 2,
                digits.stats = 2,
                depvar = FALSE) -> result_1st
```

```
print(result_1st)
```

```
##                                        model 1             model 2
## 定数項                              -3.2. (1.7)       -1.3** (0.42)
## 馬力／重量                       28.7*** (0.99)       23.5*** (1.5)
## 燃費(キロメートル／ 1 リットル)       -0.01 (0.008)     -0.07*** (0.01)
## サイズ                           0.20*** (0.01)      0.19*** (0.02)
## iv_BLP_own_hppw                    -0.73* (0.34)
## iv_BLP_own_FuelEfficiency    -0.007*** (0.0007)
## iv_BLP_own_size                 0.01*** (0.004)
## iv_BLP_other_hppw                  0.17 (0.26)
## iv_BLP_other_FuelEfficiency    0.001** (0.0004)
## iv_BLP_other_size                -0.002 (0.003)
## iv_GH_own_hppw                                      -1.9*** (0.44)
## iv_GH_own_FuelEfficiency                          5.3e-5. (3.2e-5)
## iv_GH_own_size                                  -0.001*** (0.0002)
## iv_GH_other_hppw                                    0.42*** (0.10)
## iv_GH_other_FuelEfficiency                      4.9e-5*** (8.7e-6)
## iv_GH_other_size                                0.0001*** (3.9e-5)
## ------------------------- ------------------ ------------------
## S.E. type                 Heteroskedas.-rob. Heteroskedas.-rob.
## R2                                      0.62                0.62
## Observations                           1,823               1,823
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

自己価格弾力性の推定値

	平均	標準偏差	メディアン	最小値	最大値
OLS	−0.65	0.47	−0.52	−3.22	−0.18
BLP 操作変数	−0.72	0.52	−0.58	−3.58	−0.2
差別化 操作変数	−1.4	1.01	−1.13	−6.97	−0.39

(エコノメ)自己価格弾力性の計算

```
data %>%
  dplyr::mutate( own_elas_ols = ols$coefficients["price"]*price*(1-share),
          own_elas_ivblp = iv_BLP$coefficients["fit_price"]*price*(1-share),
          own_elas_ivgh  = iv_GH$coefficients["fit_price"]*price*(1-share) ) -> data_elas

data_elas %>%
  dplyr::select( starts_with("own_elas") ) %>%
  summarytools::descr( transpose = TRUE,
                       stats = c("mean", "sd", "med", "min", "max") )
```

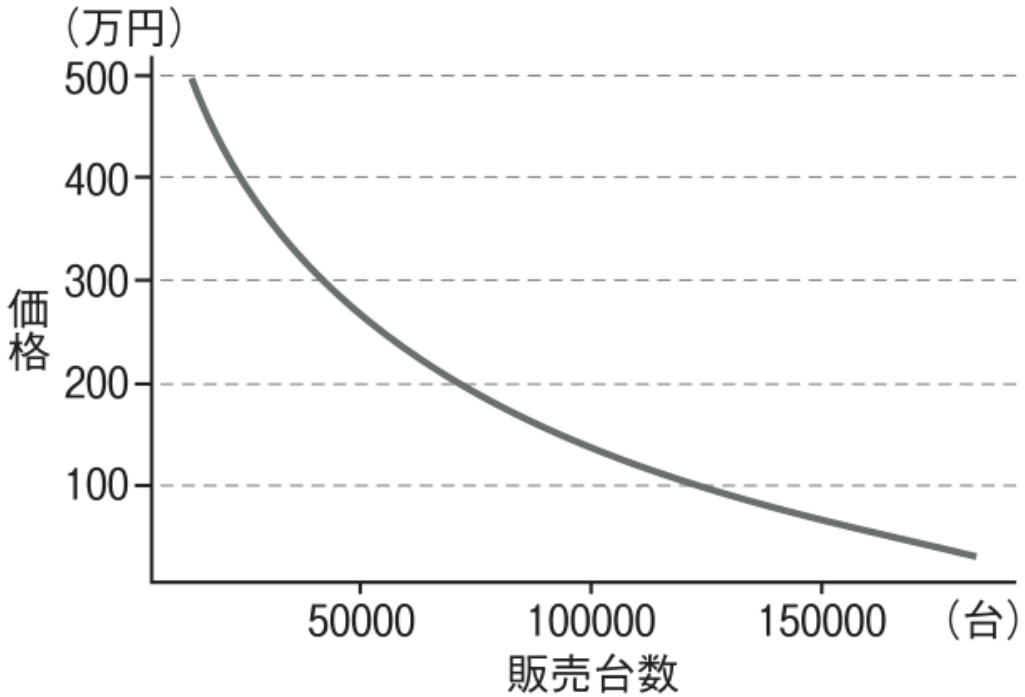
```
## Descriptive Statistics
## data_elas
## N: 1823
##
##                        Mean    Std.Dev    Median       Min       Max
## -------------------- ------- --------- -------- ------- -------
##       own_elas_ivblp   -0.72      0.52    -0.58   -3.58   -0.20
##        own_elas_ivgh   -1.40      1.01    -1.13   -6.97   -0.39
##         own_elas_ols   -0.65      0.47    -0.52   -3.22   -0.18
```

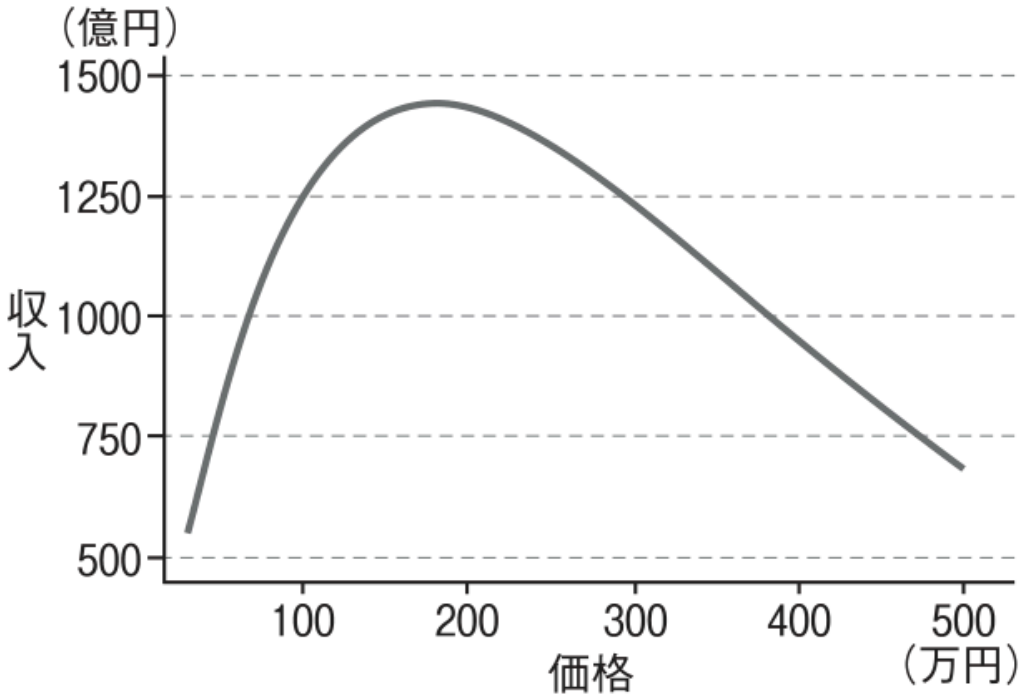
ケース：ベータードの価格付け

- 日評自動車の値下げ検討対象車「ベータード」
  - ミニバン
  - 2016年における価格は約320万円であり、年間の販売台数は約3万7000台
- ベータードの需要曲線と収入曲線(=価格×販売台数)を推定値から作ろう
  - 注：他の車種の価格はデータ上のものに固定している。

ベータードの需要曲線(左)と収入曲線(右)

- 収入を最大化する価格は約181万円。





(エコノメ) 需要・収入曲線の構築手順

1. まず、推定結果から ξ_{jt} をゲットする。
2. 関数を作成する。
3. どこか市場を固定する。
4. 製品を一つ固定する。
5. その製品の価格を変えたときに、どの製品のSalesがどう変わるかをPredictする。その際、他の製品価格はデータのものに固定しておく。

(エコノメ)

```
data %>%
  select( NameID, year, Sales, price, FuelEfficiency, size, hppw, HH, share ) %>%
  mutate( xi_fit = resid(iv_GH)) -> dt_application
```

- 日評自動車が販売する、NameID 87(ベータード)について考える。

```
NameID_target <- 87

dt_application %>%
  filter( NameID == NameID_target & year == 2016) %>%
  head()
```

```
## # A tibble: 1 × 10
##   NameID  year Sales price FuelEfficiency  size   hppw       HH    share xi_fit
##    <int> <dbl> <dbl> <dbl>          <dbl> <dbl>  <dbl>    <dbl>    <dbl>  <dbl>
## 1     87  2016 37069  3.20           11.6  17.1 0.0948 56950757 0.000651   1.16
```

(エコノメ) 販売台数を計算する関数の作成

- 価格をインプットとして、販売台数を返す関数を考える。

```
f_share <- function(price_cand, year, modelID_target, dt, estparam){

  dt %>%
    filter( year == 2016) %>%
    mutate( temp_price = ifelse( NameID == modelID_target, price_cand, price) ) %>%
    mutate( delta = estparam[1] + estparam[2]*temp_price + estparam[3]*hppw +
              estparam[4]*FuelEfficiency + estparam[5]*size + xi_fit ) %>%
    mutate( denom = 1 + sum(exp(delta))) %>%
    mutate( pred_sales = exp(delta)/denom*HH) %>%
    filter( NameID == modelID_target ) -> dt_result

  return(dt_result$pred_sales)

}
```

(エコノメ) 価格ごとに販売台数・収入を計算

```
estparam <- iv_GH$coefficients
NameID_target <- 87

# 価格の範囲として、0.5(50万円)から5(500万円)を考える。なお、もとの価格は3.198 (319.8万円)
pricevec <- seq(from = 0.3, to = 5,  by = 0.05)
quantvec <- numeric(length(pricevec))
revenuevec <- numeric(length(pricevec))

for ( i in 1:length(pricevec)){

  quantvec[i] <- f_share(price_cand = pricevec[i], year = 2016,
                         modelID_target = NameID_target,
                         dt = dt_application, estparam = estparam )
  revenuevec[i] <- pricevec[i]*quantvec[i]

}
```

(エコノメ) 需要曲線と収入曲線のプロット

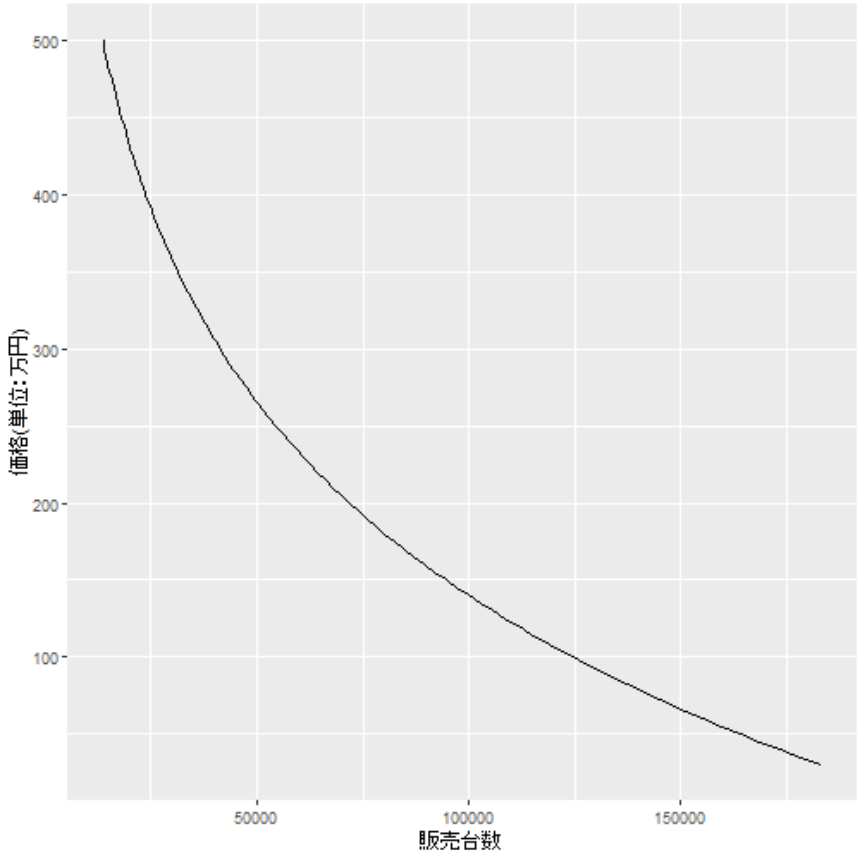
```
fig_demand <- qplot(quantvec, pricevec*100, xlab = "販売台数",
                    ylab = "価格(単位：万円)", geom = "line" )
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
fig_rev <- qplot(pricevec*100, revenuevec*100/10000,
                 ylab = "収入(単位：億円)", xlab = "価格(単位：万円)",
                 geom = c("line"), ylim = c(500,1500) )
```

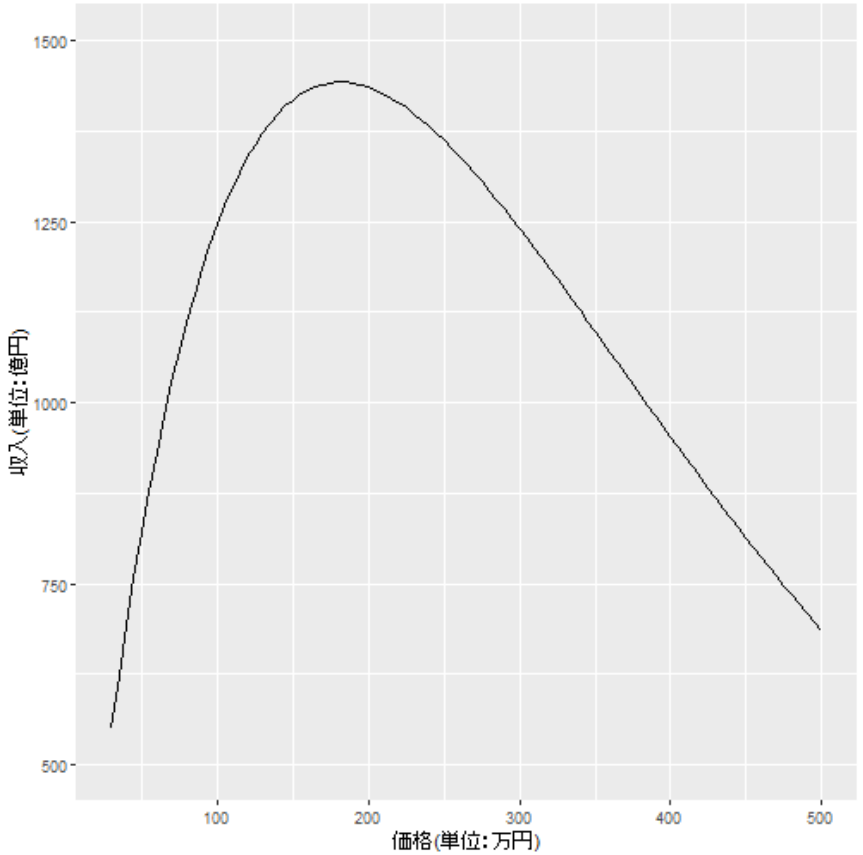
(エコノメ) 需要曲線

```
plot(fig_demand)
```



(エコノメ) 収入曲線

```
plot(fig_rev)
```



(エコノメ) 収入を最大化する価格

```
# 最適化で利用する関数を定義する。
f_revenue <- function(price_cand, year , modelID_target, dt , estparam){
  quantity <- f_share(price_cand, year , modelID_target, dt , estparam  )
  revenue <- price_cand*quantity
  return(revenue)
}

# 最適化アルゴリズムを使って、収入を最大にする価格を求める。
result <- optimise( f_revenue, interval = c(0.3, 3), maximum = TRUE,
          year = 2016,
          modelID_target = NameID_target, dt = dt_application, estparam = estparam )

print(result)
```

```
## $maximum
## [1] 1.813709
##
## $objective
## [1] 144273.8
```