

# 離散選択モデル: 講義編

応用社会科学RAブートキャンプ

講師: 遠山祐太 (早稲田大学)

最終更新: 2024-08-27 23:16:12

# イントロダクション

# 自己紹介

- 遠山祐太（とおやま ゆうた）
  - ホームページ: <https://yutatoyama.github.io/>
- 経歴：
  - 仙台市生まれ、仙台一高卒
  - 京大経済→東大院修士（公共政策→経済学）→Ph.D留学
  - 米国ノースウェスタン大学で Ph.D. in Economics 取得
  - 帰国後、2018年9月より早稲田大学政治経済学部准教授
  - 大学外のお仕事：UTEconアドバイザー、など
- 研究・教育：実証産業組織論、エネルギー・環境経済学、応用計量経済学

# テーマ：離散選択モデルの推定

- 离散選択モデル：**離散変数**の決定に関するモデル
  - Yes or No (二項選択)
  - 複数の選択肢からどれを選ぶか(多項選択)
  - 順序を伴う選択肢：悪い<普通<良い
- 特徴 1：応用範囲が非常に広い
  - マーケティング・消費者行動：どの財を購入するか？
  - 交通：通勤経路の選択（電車・バス・自家用車）
  - 労働・教育：どの学校に出願するか？いつ引退するか？
  - 政治経済学：誰に投票するか？
- 特徴 2：経済学モデルと密接な繋がりを持つ

# 講義の目標

- 目標 1：離散選択モデルによって、どのような分析ができるかを学ぶ。
- 目標 2：パッケージを利用した離散選択モデルの推定に親しむ。

# 今日のプラン

- 1コマ目（13時半から15時）：講義パート
  - イントロダクション（5分）
  - 離散選択モデルと最尤法の導入（30分）
  - 事例：きのこの山 VS たけのこの里（10分）
  - コーディング：[mlogit](#)パッケージによる推定（45分）
- 2コマ目（15時半から17時）：演習パート
  - 講義パートの残り&質疑応答
  - データを用いたグループ実習

# 本授業の進め方とお願い

- 全体的に密度が高く、カバーする量も多めです。
- **講義中の質問を強く推奨します！！**
- スライドタイトルについて：
  - 「【R分析】」：講義パートではサラッとカバー。後ほど説明。
  - 「【参考】」：計量経済学の知識が必要な点。ひとまず割愛してOK

# 参考文献

- 上武・遠山・若森・渡辺 「実証ビジネスエコノミクス」 の第2章
  - 経済セミナー連載記事に基づき、現在書籍化進行中！
- Croissant "Estimation of Random Utility Models in R: The mlogit Package" *Journal of Statistical Software*
- Train "Discrete Choice Methods with Simulation"
  - 特にChapter 2,3, and 6
- ダハナ・勝又 「Rによるマーケティング・データ分析」 第5章

# 離散選択モデル

# 例：あなたはどのiPhoneを買いますか？

iPhone 14      iPhone SE(第3世代)      iPhone 13

ブルー パープル ブラック ホワイト レッド  
ミッドナイト  
グリーン

119,800円(税込)から  
購入  
さらに詳しく >

62,800円(税込)から  
購入  
さらに詳しく >

107,800円(税込)から\*  
購入  
さらに詳しく >

iPhone 14      iPhone SE(第3世代)      iPhone 13

**6.1インチ**  
Super Retina XDRディスプレイ<sup>1</sup>

**4.7インチ**  
Retina HDディスプレイ

**6.1インチ**  
Super Retina XDRディスプレイ<sup>1</sup>

SOS  
緊急SOS  
衝突事故検出

SOS  
緊急SOS

SOS  
緊急SOS

先進的なデュアルカメラシステム  
12MPメイン | 超広角  
オートフォーカスに対応したTrueDepth  
フロントカメラ  
圧倒的なディテールと色彩のための  
Photonic Engine

先進的なカメラシステム  
12MPメイン  
フロントカメラ

デュアルカメラシステム  
12MPメイン | 超広角  
TrueDepthフロントカメラ

**2倍**  
の光学ズームレンジ

-

**2倍**  
の光学ズームレンジ

最大20時間のビデオ再生<sup>2</sup>  
最大15時間のビデオ再生<sup>3</sup>  
最大19時間のビデオ再生<sup>3</sup>

# 離散選択モデル (Discrete Choice Model)

- 離散選択モデル：「**数多くの選択肢の中からどれを選ぶか**」を表現する数理モデル
- 意思決定者は**複数(有限個)の選択肢**に直面している。
  - iPhoneの例：iPhone 13, 14, SE(3世代), 13 Pro, などなど
- それぞれの選択肢(例：製品)には、付随する**特徴**がある
  - iPhoneの例：価格、ディスプレイサイズ、カメラの画素数、バッテリー持続時間、などなど
- 意思決定者は「どの特徴を重視するか」という**好み**をもっている。
  - 例：値段は気にしない（親が払ってくれるから）、大きい画面が良い、などなど
- 選択肢の特徴を踏まえて、**自分の満足度(効用)が最も高くなる選択**をする

# セットアップ

- 消費者  $i = 1, \dots, N$
- 製品・選択肢  $j \in \mathbf{J} = \{1, \dots, J\}$ 
  - $\mathbf{J}$  を選択肢集合(choice set)と呼ぶ。
- 消費者  $i$  が製品  $j$  から得られる効用

$$u_{ij} = \alpha p_j + \beta X_j + \epsilon_{ij}$$

- $p_j$ : 価格,  $X_j$ : 製品の属性・特徴のベクトル。
- $(\alpha, \beta)$ : 選好パラメタ
- $\epsilon_{ij}$ : 個人かつ製品レベルのランダムな選好ショック。価格や製品属性で捉えられない要素、言わば誤差項として考える。

# 簡単な例 1 : iPhoneの購入

- 以下の効用関数を考えよう (選好ショックは捨象)

$$u_{ij} = -2p_{jt} + 3 \times (\text{画面サイズ}) + 0.5 \times (\text{バッテリー時間})$$

- 以下の三機種を比較しよう

モデル	価格(万円)	画面サイズ(インチ)	バッテリー(時間)	効用
iPhone 14	11.98	6.1	20	4.34
iPhone SE	6.28	4.7	15	<b>9.04</b>
iPhone 13	10.78	6.1	19	6.24

- iPhone SEを購入する！！**

# 簡単な例2：もし価格をあまり気にしないと？

- 価格の係数が小さい (=価格をあまり気にしない)

$$u_{ij} = -1p_{jt} + 3 \times (\text{画面サイズ}) + 0.5 \times (\text{バッテリー時間})$$

- 以下の三機種を比較しよう

モデル	価格(万円)	画面サイズ(インチ)	バッテリー(時間)	効用
iPhone 14	11.98	6.1	20	16.32
iPhone SE	6.28	4.7	15	15.32
iPhone 13	10.78	6.1	19	<b>17.02</b>

- iPhone 13を購入する！！**
- ポイント：人々の「好み」によって、選択は異なってくる！！

# 離散選択問題のより一般的な定式化

- 以下のように効用を定義

$$u_{ij} = V_{ij} + \epsilon_{ij}$$

- $V_{ij}$ は製品属性や消費者属性に依存する。
  - 例1:  $V_{ij} = \alpha p_j + \beta X_j$
  - 例2:  $V_{ij} = (\alpha_0 + \alpha z_i)p_j + \beta X_j$ , ここで  $z_i$ は消費者属性(例えば所得)
- $\epsilon_{ij}$ は選好ショック
- 消費者  $i$  は最も高い効用が得られる製品・選択肢  $j$  を一つ選ぶ。

$$d_i = \arg \max_{j \in \{1, \dots, J\}} u_{ij}$$

# 離散選択問題から選択確率の導出

- ・消費者は効用の全て要素を把握した上で選択 -> 決定論的選択
- ・しかしながら、分析者には選好ショック  $\{\epsilon_{ij}\}_{j=1}^J$  は観察できない。(いわゆる誤差項)
- ・モデルの予測として、以下の**選択確率**を考える。

$$P(d_i = j) = \Pr \left( \{\epsilon_{ij}\}_{j=1}^J : V_j + \epsilon_{ij} \geq V_k + \epsilon_{ik} \forall k \neq j \right)$$

- 解釈：選択肢  $j$  から得られる効用が最も大きくなる確率
- ・この選択確率の形は  $\{\epsilon_{ij}\}_{j=1}^J$  に関する分布の仮定による。
  - 例：ロジットモデル、プロビットモデル

# 離散選択モデルのバラエティについて

- 二項ロジット・プロビットモデル：選択肢が2個 ( $J = 2$ ) の場合
- 多項ロジットモデル：選択肢の個数が2以上の場合
- ランダム係数ロジットモデル：効用パラメタの係数が人によってランダムな場合
- その他(本日は割愛)：
  - ネスト型ロジットモデル
  - 順序ロジット・プロビットモデル
  - 多項プロビットモデル：多項ロジットよりも計算が非常に複雑
  - BLPモデル

# 多項ロジットモデル

- $\epsilon_{i,j}$  が i.i.d. の 第一種極値分布(type I extreme value distribution) に従うと仮定。

$$F(x) = \exp(-\exp(-x))$$

- 分散は  $\pi^2/6$ .
- この分布の元で、選択確率は

$$\Pr(d_i = j) = \frac{\exp(V_{ij})}{\sum_{k=1}^J \exp(V_{ik})}$$

- **ロジット確率**とも呼ばれる。
- 導出はTrainのChapter 3を参照。

# 単純化としての二項ロジット

- 選択肢が2個 ( $J = 2$ ) の場合を考える。
- 二項ロジットモデル：

$$\Pr(d = 1) = \frac{\exp(V_{i1})}{\exp(V_{i1}) + \exp(V_{i2})}$$

- 二項プロビットモデル： $\epsilon_{i,j}$  が i.i.d. の標準正規分布に従うと考える。
- 二項ロジット・プロビット(と最尤法の理論)については、  
[Lecture\\_9\\_MLE\\_binary\\_choice.pdf](#) を参照。

# 離散選択モデルの応用

- 以下のモデルを考えよう。

$$\Pr(d_i = j | \{p_j, X_j\}_{j=1}^J) = \frac{\exp(\alpha p_j + \beta X_j)}{\sum_{k=1}^J \exp(\alpha p_k + \beta X_k)}$$

- モデルの使い方
  - 限界効果
  - パラメタの解釈と支払い意思額
  - 予測

# 利用方法 1：限界効果

- 線形回帰モデルと異なり、係数そのものは左辺への限界効果ではない。
- ロジットモデルにおける限界効果は

$$\frac{\partial \Pr(d = j)}{\partial p_j} = \Pr(d = j) (1 - \Pr(d = j)) \times \alpha$$

- 応用例：自己価格弾力性

$$\frac{\partial \Pr(d = j)}{\partial p_j} \frac{p_j}{\Pr(d = j)} = (1 - \Pr(d = j)) \times \alpha p_j$$

- 価格が1%変化したときに、需要が何%変化するか？

## 利用方法2：パラメタの解釈と支払い意思額

- 効用の中に価格が入っている場合、製品属性への支払い意思額を計算することが可能。
- 例： $V_j = \alpha p_j + \beta size_j$  とする。 $p_j$  は価格、 $size_j$  はモニターサイズ(単位：インチ)。
- モニターサイズ1インチへの支払い意思額は

$$\frac{\beta}{|\alpha|}$$

- 考え方： $size_j$ を増加させたとき、効用を一定水準に保つには価格  $p_j$  がどの程度動く必要があるか、を捉えている。

# 利用方法3：モデルを用いた予測

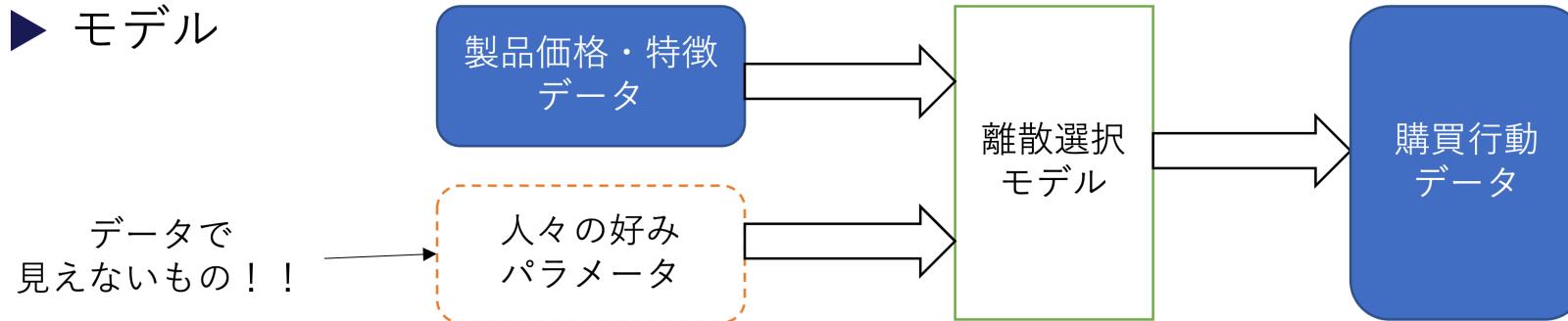
- ロジットモデル再掲

$$\Pr(d_i = j | \{p_j, X_j\}_{j=1}^J) = \frac{\exp(\alpha p_j + \beta X_j)}{\sum_{k=1}^J \exp(\alpha p_k + \beta X_k)}$$

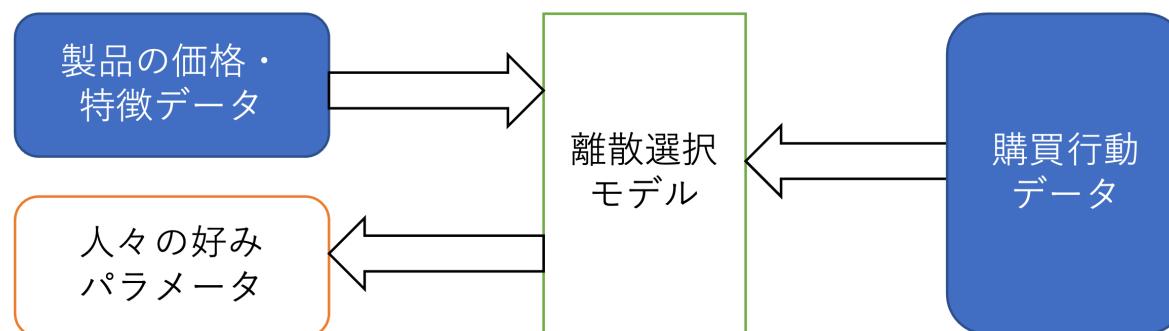
- 例1：製品の価格  $p_j$  を下げたら需要がどう変化するか？
  - iPhone SEの値下げ -> SEの需要増加
  - 同時に、iPhone 13の需要は減るかも（製品の代替）
- 例2：新製品を導入したらどうなるか？
  - 例えば、iPhone Max という  $J+1$  番目の新製品  $\{p_{J+1}, X_{J+1}\}$
  - 新しい製品の需要予測、そして既存の製品への影響を予測できる。

# 推定：モデルからデータへ

## ▶ モデル



## ▶ データ分析：



▶ 「モデルの予測」と「実際の購買行動」が近くなるような「パラメタ」を推定する！！

# 最尤法 (maximum likelihood) による推定

- データ: 各個人  $i$  について  $\{X_j, p_j, d_{ij}\}_{j=1}^J$ 
  - $d_{ij} = 1$  消費者  $i$  が 製品  $j$  を選んだとき 1, それ以外は0
- 尤度関数(likelihood):** ある実現値が発生するような確率をパラメタの関数と表したもの。
- 多項ロジットモデルの尤度関数 (パラメタ  $\theta = (\alpha, \beta)$ )

$$L(\theta) = \prod_{i=1}^N \left[ \prod_{j=1}^J (Pr(d_i = j|\theta))^{d_{ij}} \right]$$

- 尤度関数は、「手元にあるデータが、モデルによって生成される確率」と解釈される。
- この確率を最大にするようなパラメタを「良い推定量」とする。
  - 一定の仮定のもとで一致性・漸近正規性・効率性などの理論的性質が担保されている。

# 対数尤度関数の最大化

- 最尤法では、以下の対数尤度を用いる。

$$\log L(\theta) = \sum_{i=1}^N \sum_{j=1}^J d_{ij} \log Pr(d_i = j | \theta)$$

- 対数尤度関数を最大化するようなパラメタ  $\theta$  が最尤推定量となる。
- ポイント：最大化するパラメタは、基本的に数値計算・数値最適化で求める。
  - 非常にシンプルなモデル（例：線形回帰モデル）では解析的に得られる。

事例：きのこの山 VS たけのこの里

# きのこの山 VS たけのこの里



# きのこの山 VS たけのこの里

- アンケートデータ&離散選択モデルを用いて、「きのこの山」と「たけのこの里」の需要関数を推定しよう。
- 何が知りたいか?
  - 「きのこの山」と「たけのこの里」はどちらが人気か？
  - 「ブランド価値」と「価格」のトレードオフ -> 需要関数・価格弾力性
  - 価格を変えたときに、需要量・収入はどのように変化するか？

# アンケート(メインクエスチョン)

- あなたはスーパー・マーケットのお菓子コーナーに来てします。目の前に「きのこの山」と「たけのこの里」が並んでいます。「きのこの山」と「たけのこの里」の一箱あたり税込価格が、以下の組み合わせで与えられるとき、あなたにとって最も望ましい選択肢を一つ選んでください。
- 価格の組み合わせ：
  - (きのこの山、たけのこの里) = (200円、 200円)
  - (きのこの山、たけのこの里) = (180円、 200円)
  - (きのこの山、たけのこの里) = (200円、 170円)
  - (きのこの山、たけのこの里) = (220円、 200円)
  - (きのこの山、たけのこの里) = (190円、 210円)
- 選択肢：
  - きのこの山を買う。
  - たけのこの里を買う。
  - どちらも買わない。

# レクチャーで使うデータ

- 2024年春学期「産業組織論」@早稲田大学政治経済学部で行ったアンケート結果を利用。
- アンケートでは上述の選択肢問題に加えて、消費者属性についての設問も。
- 演習においてRAブートキャンプ参加者に行ったアンケートデータを用いた分析を行う。

# 結果を見る前に：選択型コンジョイント分析

- 今から見していく分析を一般に、**選択型コンジョイント分析**と呼ぶ。
- 仮想状況における選択に関する質問から、消費者の需要・選好を推定することが可能。
- マーケティング、環境経済学などにおいて幅広く使われている手法。
- 限界点：あくまで仮想の状況であるので「実際の行動」を反映するか否かは要注意。(次ページ)
- マーケティングにおけるコンジョイント分析の解説として、Allenby et al "Economic foundations of conjoint analysis" *Handbook of the Economics of Marketing*

# データの種類と限界点

- 今回用いたサーベイデータを表明選好データ(Stated preference)と呼ぶ。
  - 仮想のアンケートを用いて、人々の選好に関する情報を収集している。
- 一方、現実の購買行動データを、顯示選好データ(Revealed preference)と呼ぶ。
  - POS データ、購買履歴データ、など。
- 表示選好データにおいては、常に「その行動が真の行動を反映しているのか？」を考えなければならない。
- 同時に、サーベイ一般の問題として、サーベイ対象者が興味ある母集団全体を反映しているのかについても要検討。

# 分析の下準備

# 【R分析】下準備

```
rm(list = ls())
library("tidyverse")
library("knitr")
library("mlogit") # ロジットモデル推定のためのパッケージ
library("stargazer") # 推定結果の表作成
```

# 【R分析】データの読み込み

```
data <- readr::read_csv("data/KinokoTakenokoSurvey_IOSpring2024.csv")
```

# 変数一覧

変数	説明
ID	回答者のID
experience	きのこの山・たけのこの里を最後にいつ食べたか？
Q1	(きのこの山、たけのこの里) = (200円、200円)
Q2	(きのこの山、たけのこの里) = (180円、200円)
Q3	(きのこの山、たけのこの里) = (200円、170円)
Q4	(きのこの山、たけのこの里) = (220円、200円)
Q5	(きのこの山、たけのこの里) = (190円、210円)
age	年齢
gender	性別
region	出身地
familyhouse	実家暮らしか 一人暮らしか

# 【R分析】アンケートのメインの結果

各問における選択肢のシェアを計算する。

```
# 回答者数
N <- length(data$ID)

# 必要なデータを抽出
data %>%
  select(ID, Q1, Q2, Q3, Q4, Q5) %>%
  gather(key = Q, value = choice, Q1, Q2, Q3, Q4, Q5) -> datafig

# データ整形
datafig %>%
  mutate( Q = ifelse(Q == "Q1", "Q1: (200円, 200円)", Q),
         Q = ifelse(Q == "Q2", "Q2: (180円, 200円)", Q),
         Q = ifelse(Q == "Q3", "Q3: (200円, 170円)", Q),
         Q = ifelse(Q == "Q4", "Q4: (220円, 200円)", Q),
         Q = ifelse(Q == "Q5", "Q5: (190円, 210円)", Q) ) -> datafig
```

# 【R分析】シェアの計算

```
datafig %>%
  group_by(Q, choice) %>%
  tally() %>%
  mutate( n = n/N) %>%
  pivot_wider( id_cols = "Q", names_from = "choice", values_from = "n") %>%
  knitr::kable( digits = 2) -> tab
```

# 選択肢のシェア

tab

Q	1: きのこの山を買う	2: たけのこの里を買う	3: どちらも買わない
Q1: (200円, 200円)	0.29	0.50	0.22
Q2: (180円, 200円)	0.57	0.27	0.17
Q3: (200円, 170円)	0.10	0.79	0.11
Q4: (220円, 200円)	0.07	0.63	0.31
Q5: (190円, 210円)	0.51	0.27	0.22

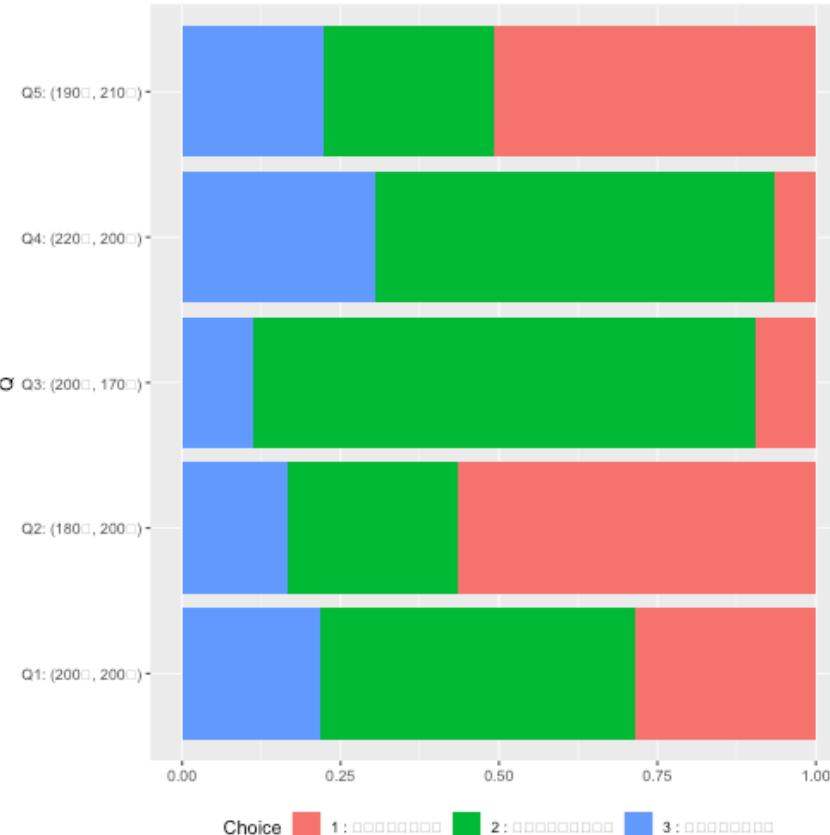
# 【R分析】 グラフで出力

帯グラフで結果を出力する。

```
p <- ggplot() +  
  geom_bar(data = datafig, aes(x = Q, fill = as.factor(choice) ), position = "fill" ) +  
  coord_flip() +  
  theme(legend.position="bottom", axis.title.x=element_blank())+  
  scale_fill_discrete(name = "Choice")
```

```
plot(p)
```

# アンケートのメインの結果



# アンケート結果からのポイント

Q	1: きのこの山を買う	2: たけのこの里を買う	3: どちらも買わない
Q1: (200円, 200円)	0.29	0.50	0.22
Q2: (180円, 200円)	0.57	0.27	0.17
Q3: (200円, 170円)	0.10	0.79	0.11
Q4: (220円, 200円)	0.07	0.63	0.31
Q5: (190円, 210円)	0.51	0.27	0.22

- ポイント 1 : 価格が上がるとシェアが下がる → 価格に反応
- ポイント 2 : 価格差が大きくても選ぶ人もいる → きのこ／たけのこのブランド価値
- 離散選択モデルを使って、より詳しく分析してみよう！

# 多項ロジットモデル

# 離散選択モデル：多項ロジットモデル

- 消費者  $i$  が設問  $k$  において、選択肢  $j \in \{Kinoko, Takenoko, outside\}$  から得る効用

$$U_{i,k,Kinoko} = \alpha_{Kinoko} - \beta \cdot p_{k,Kinoko} + \epsilon_{i,k,Kinoko}$$

$$U_{i,k,Takenoko} = \alpha_{Takenoko} - \beta \cdot p_{k,Takenoko} + \epsilon_{i,k,Takenoko}$$

$$U_{i,k,outside} = \epsilon_{i,k,outside}$$

- $\alpha_{Kinoko}, \alpha_{Takenoko}$ : きのこ・たけのこ自体への好み
  - $p_{k,j}$  設問  $k$  における選択肢  $j$  の価格
  - $\epsilon_{i,k,j}$ : ランダムな選好ショック
- 補足：「何も買わない」オプションを「アウトサイド・グッズ」と呼ぶ。
  - 推定するパラメタは  $\alpha_{Kinoko}, \alpha_{Takenoko}, \beta$  の3つ。
    - 全消費者で共通のパラメタとする。後ほど拡張（ランダム係数ロジット）

# 多項ロジットモデルにおける選択確率

- 各選択肢の選択確率は

$$P_k(j|\theta) = \frac{\exp(\alpha_j - \beta \cdot p_{j,k})}{1 + \exp(\alpha_{Kinoko} - \beta \cdot p_{Kinoko,k}) + \exp(\alpha_{Takenoko} - \beta \cdot p_{Takenoko,k})}$$

- 尤度関数は

$$L(\theta) = \prod_{i=1}^N \prod_{k=1}^5 P_k(j = y_{i,k}|\theta)$$

- 対数尤度関数を、パラメタ  $\theta$  に関して最大化する。
  - `mlogit` パッケージを用いる。
  - 自分でプログラムを書く。

# 【R分析】 推定：mlogitパッケージ

- mlogitパッケージ：Rにおける離散選択モデル推定のためのパッケージ
- ホームページは[こちら](#)
- 様々な手法をカバー
  - 多項ロジット:[[モデル](#)] [[例](#)]
  - ランダム係数ロジット: [[モデル](#)] [[例](#)]
- (個人的な感想として) パッケージで用いるためのデータ加工が若干独特。
  - データ加工の詳細は[こちら](#)

# 【R分析】 mlogitパッケージのためのデータ加工

まずは推定用にデータを加工する。

```
# きのこ・たけのこを食べたことない人をDropする。
data %>%
  filter( experience != "4 : 食べたことがない") -> data_for_estimation

# データをLong形式に加工する。
# 各行は「回答者ID-設問」という単位になる。
data_for_estimation %>%
  gather(key = "occasion", value = choice, starts_with("Q")) -> data_for_estimation
```

# 【R分析】価格データの作成

- データにマージする、各設問・各選択肢の価格情報を準備する。

```
pricedata <-  
  data.frame( occasion = c("Q1", "Q2", "Q3", "Q4", "Q5"),  
              price_0 = numeric(5),  
              price_1 = c(200, 180, 200, 220, 190),  
              price_2 = c(200, 200, 170, 200, 210) )  
pricedata <- as_tibble(pricedata)
```

- 注意点
  - `price_0` という形で、`変数名_選択肢番号` という定義になっている。
  - 0はアウトサイドグッズ、1はきのこ、2はたけのこ
  - 後ほど`mlogit`で利用するため（後述）

# 【R分析】データをマージ

価格データをアンケート結果データに結合する。

```
data_for_estimation %>%
  left_join(pricedata) %>%
  arrange(ID, occasion) -> data_for_estimation
```

# 【R分析】きのこの山・たけのこの里ダミー変数

- ダミー変数を作成する。

```
data_for_estimation %>%
  mutate( Kinoko_0 = 0,
         Kinoko_1 = 1,
         Kinoko_2 = 0,
         Takenoko_0 = 0,
         Takenoko_1 = 0,
         Takenoko_2 = 1    ) %>%
  arrange(ID, occasion) -> data_for_estimation
```

- 1は「きのこ」、2は「たけのこ」、0は「どちらも買わない」なので、kinoko\_1はきのこを買うことから1、takenoko\_2はたけのこを買うことから1となる。

# 【R分析】選択に関する変数

- 各設問での選択に関する変数choiceを再定義する。

```
data_for_estimation %>%  
  mutate( choice = case_when( choice == "1 : きのこの山を買う" ~ 1,  
                             choice == "2 : たけのこの里を買う" ~ 2,  
                             choice == "3 : どちらも買わない" ~ 0 )) -> data_for_estimation
```

# 【R分析】選択の状況に関する変数の定義

- 現在のデータの各行は、各個人が各設問においてどう行動したかを示している。
- 変数choiceidは、このような「選択の状況」を示すIDとなっている。
  - 構築としては、単純に上から下まで連番を振ればよい。

```
data_for_estimation$choiceid <- 1:nrow(data_for_estimation)

# slide_4_code_scratch.RMDのために保存
write_csv(file = "intermediate/data_for_estimation.csv", x = data_for_estimation)
```

# 【R分析】 mlogit用のデータセット

- mlogit専用のデータ形式に変更する。

```
datalogit <- dfidx(data = as.data.frame(data_for_estimation),  
                    choice = "choice",  
                    varying = 9:17,  
                    sep = "_",  
                    idx = list(c("choiceid", "ID")),  
                    idnames = c("chid", "alt"),  
                    opposite = c("price"))
```

# 【R分析】各引数の説明

- `data`: データセット。
- `choice`: 各選択状況（各行）における選択を示す変数
- `varying`: データセットにおいて、選択肢ごとに異なる値をとる変数。今回は9-17列目。
- `sep`: 上のvaryingな変数は、`変数名_選択肢`となっており、この変数名と選択肢を分離する記号を指定する。
- `idx`: 各選択状況を示す変数(`choiceid`)と、各個人を示す変数(`ID`)を指定する。
- `idnames`: 新しいデータにおいて、各選択状況を示す変数と選択肢をしめす変数の名前
- `opposite`: 効用関数において符号がマイナスにはいるような変数の指定。今回は`price`.

# 【R分析】新しいデータセットのインデックス情報

```
head(datalogit$idx)
```

```
## ~~~ indexes ~~~~  
##   chid ID alt  
## 1     1  1   0  
## 2     1  1   1  
## 3     1  1   2  
## 4     2  1   0  
## 5     2  1   1  
## 6     2  1   2  
## indexes:  1, 1, 2
```

# 【R分析】多項ロジットモデルの推定

- 多項ロジットモデルの推定を行う。

```
multilogit <- mlogit( formula = choice ~ price + Kinoko + Takenoko | 0,  
                      data = datalogit)
```

- 引数：
  - formula**: 効用の定式化を指定。今回は、価格、きのこダミー、たけのこダミー。なお、| の後の0はひとまず無視してOK.
  - data**: 上で作成したデータセット
- stargazer**で推定結果をレポート

```
stargazer::stargazer(multilogit, type = "text")
```

# 多項ロジットモデルの推定結果

```
##  
## =====  
## Dependent variable:  
## -----  
## choice  
## -----  
## price          0.058***  
##                  (0.004)  
##  
## Kinoko        11.685***  
##                  (0.726)  
##  
## Takenoko     12.224***  
##                  (0.741)  
##  
## -----  
## Observations      1,190  
## Log Likelihood   -1,077.499  
## =====  
## Note:           *p<0.1; **p<0.05; ***p<0.01
```

# 拡張 1：消費者属性の追加

# 消費者属性

- サーベイでは消費者属性についても質問している。
  - 年齢、性別、出身地、などなど
- きのこの山・たけのこの里への選好や価格の反応度はこれらにも依存するかもしれない。
- これらの要素を取り込んだ多項ロジットモデルの推定を行おう。

# 【R分析】消費者属性の記述統計準備

```
# 回答者の属性の比率を表にするためのデータ整形
data %>%
  mutate(gender = case_when(gender == "1 : 男性" ~ "男性",
                            gender == "2 : 女性" ~ "女性",
                            gender == "3 : 回答したくない" ~ "未回答"),
         exp     = case_when(experience == "1 : 過去半年以内" ~ "過去半年以内",
                            experience == "2 : 過去半年から1年以内" ~ "過去半年から1年以内",
                            experience == "3 : 1年以上前" ~ "1年以上前",
                            experience == "4 : 食べたことがない" ~ "食べたことがない"),
         adult = if_else(age>=20, "成人", "未成年") ) -> temp
```

```
temp %>%  
  mutate(region = case_when(region == "1" : 北海道地方" ~ "関東",  
                            region == "2" : 東北地方" ~ "関東",  
                            region == "3" : 関東地方" ~ "関東" ,  
                            region == "4" : 中部地方" ~ "関東",  
                            region == "5" : 近畿地方" ~ "関西",  
                            region == "6" : 中国地方" ~ "関西",  
                            region == "7" : 四国地方" ~ "関西",  
                            region == "8" : 九州地方(沖縄含む)" ~ "関西",  
                            region == "9" : 海外" ~ "海外"),  
  familyhouse = if_else(familyhouse==1,"実家暮らし","実家暮らしでない") ) -> data_attr
```

```
# 地域
data_atr %>%
  group_by(region) %>%
  tally() %>%
  mutate(割合 = n/N) %>%
  select (-n) %>%
  mutate(変数 = c("出身地方", "", "")) %>%
  rename(属性 = region) %>%
  select(変数, everything()) ->region_fig

# 性別
data_atr %>%
  group_by(gender) %>%
  tally() %>%
  mutate( 割合 = n/N) %>%
  select(-n) %>%
  mutate(変数 = c("性別", "", "")) %>%
  rename(属性 = gender) %>%
  select(変数, everything()) ->gender_fig
```

```
# 実家  
data_atr %>%  
  group_by(familyhouse) %>%  
  tally() %>%  
  mutate( "割合" = n/N) %>%  
  select(-n) %>%  
  mutate("変数" = c("実家暮らしかどうか", "")) %>%  
  rename(属性 = familyhouse) %>%  
  select(変数, everything(),) -> fam_fig
```

```
# 成人  
data_atr %>%  
  group_by(adult) %>%  
  tally() %>%  
  mutate( 割合 = n/N) %>%  
  select(-n) %>%  
  mutate(変数 = c("成人かどうか", "")) %>%  
  rename(属性 = adult) %>%  
  select(変数, everything()) ->adult_fig
```

```

rbind(gender_fig,adult_fig, fam_fig, region_fig) %>%
  knitr::kable( digits = 2) -> tab_atr

tab_atr

```

変数	属性	割合
性別	女性	0.33
	未回答	0.01
	男性	0.66
成人かどうか	成人	0.53
	未成年	0.47
実家暮らしかどうか	実家暮らし	0.74
	実家暮らしでない	0.26
出身地方	海外	0.06
	関東	0.85
	関西	0.10

# 推定のためのデータセット

```
data_for_estimation %>%  
  mutate( gender = case_when(gender == "1 : 男性" ~ "0",  
                             gender == "2 : 女性" ~ "female",  
                             gender == "3 : 回答したくない" ~ "0"),  
        exp     = case_when(experience == "1 : 過去半年以内" ~ "halfyear",  
                           experience == "2 : 過去半年から1年以内" ~ "half_to_1year",  
                           experience == "3 : 1年以上前" ~ "0" ),  
        adult = if_else(age>=20, 1, 0),  
        region = case_when(region == "1 : 北海道地方" ~ "_Kanto",  
                            region == "2 : 東北地方" ~ "_Kanto",  
                            region == "3 : 関東地方" ~ "_Kanto" ,  
                            region == "4 : 中部地方" ~ "_Kanto",  
                            region == "5 : 近畿地方" ~ "Kansai",  
                            region == "6 : 中国地方" ~ "Kansai",  
                            region == "7 : 四国地方" ~ "Kansai",  
                            region == "8 : 九州地方(沖縄含む)" ~ "Kansai",  
                            region == "9 : 海外" ~ "Oversea",  
                          )  
  ) -> data_for_estimation
```

```
datalogit <- dfidx(data = as.data.frame(data_for_estimation),  
                    choice = "choice",  
                    varying = 9:17,  
                    sep = "_",  
                    idx = list(c("choiceid", "ID")),  
                    idnames = c("chid", "alt"),  
                    opposite = c("price"))
```

# 消費者属性入りの定式化

- 価格の係数及びきのこ・たけのこ係数が、消費者属性  $z_i$  に依存すると考える。

$$U_{i,k,Kinoko} = \alpha_{Kinoko}(z_i) - \beta_{Kinoko}(z_i)p_{k,Kinoko} + \epsilon_{i,k,Kinoko}$$

$$U_{i,k,Takenoko} = \alpha_{Takenoko}(z_i) - \beta_{Takenoko}(z_i)p_{k,Takenoko} + \epsilon_{i,k,Takenoko}$$

$$U_{i,k,outside} = \epsilon_{i,k,other}$$

- ここで、係数は以下のように与える

$$\alpha(z_i) = \alpha_0 + \sum_{d=1}^D \alpha_d z_{di}$$

$$\beta_j(z_i) = \beta_{0,j} + \sum_{d=1}^D \beta_{d,j} z_{di}, j \in \{Kinoko, Takenoko\}$$

- $z_{di}$  は消費者  $i$  の属性  $d$  (成人ダミーなど)

# 【R分析】消費者属性入りの多項ロジットモデル推定

```
with_attr <- formula(choice ~ price+Kinoko+Takenoko+
                      price:gender+Takenoko:gender+Kinoko:gender+
                      price:familyhouse+Takenoko:familyhouse+Kinoko:familyhouse+
                      price:adult+Takenoko:adult+Kinoko:adult+
                      price:region+Takenoko:region+Kinoko:region | 0 | 0) # 式が長いので事前に格納

ml_consumer_attr <- mlogit(formula = with_attr,
                           data = datalogit,
                           reflevel = '0')
```

- **gender**: 女性ダミー、 **familyhouse**: 実家ダミー、 **adult**: 成人ダミー、 **region**: 地域（関西、 海外）

# 推定結果の出力

```
stargazer::stargazer(multilogit,ml_consumer_attr,  
                      single.row=TRUE,type="text",align = TRUE)
```

```

## 
## =====
##          Dependent variable:
## 
##          choice
##          (1)      (2)
## -----
## price           0.058*** (0.004)  0.063*** (0.011)
## Kinoko        11.685*** (0.726) 12.648*** (2.058)
## Takenoko     12.224*** (0.741) 12.794*** (2.088)
## price:genderfemale      0.013 (0.008)
## Takenoko:genderfemale   3.320** (1.658)
## Kinoko:genderfemale    3.464** (1.624)
## price:familyhouse      -0.002 (0.010)
## Takenoko:familyhouse   -0.035 (1.940)
## Kinoko:familyhouse    -0.399 (1.915)
## price:adult            -0.018** (0.008)
## Takenoko:adult         -3.494** (1.536)
## Kinoko:adult           -3.808** (1.506)
## price:regionKansai     0.016 (0.016)
## price:regionOversea    0.013 (0.018)
## Takenoko:regionKansai  3.812 (3.282)
## Takenoko:regionOversea 2.798 (3.579)
## Kinoko:regionKansai    3.353 (3.188)
## Kinoko:regionOversea   3.385 (3.574)
## -----
## Observations       1,190      1,190
## Log Likelihood   -1,077.499  -1,048.912

```

## 拡張2：ランダム係数ロジットモデル

# 離散選択モデル：ランダム係数ロジットモデル

- もし消費者間で異なった選好パラメタを持っていたらどうなるか？

$$U_{i,k,Kinoko} = \alpha_{i,Kinoko} - \beta_i p_{k,Kinoko} + \epsilon_{i,k,Kinoko}$$

$$U_{i,k,Takenoko} = \alpha_{i,Takenoko} - \beta_i p_{k,Takenoko} + \epsilon_{i,k,Takenoko}$$

$$U_{i,k,outside} = \epsilon_{i,k,other}$$

- ここで、
  - $\alpha_{i,j} \sim N(\theta_j, \sigma_j^2)$  for  $j \in \{ \text{Kinoko, Takenoko} \}$
  - $\beta_i \sim N(\theta_\beta, \sigma_\beta^2)$ .
- これらの分布のパラメタを推定する。
  - 技術的な詳細は補足資料を参照(少し難しめ)
- 注：消費者属性も加味可能だが講義では省略。(実証ビジエコ第2章を参照)

# 【R分析】 ランダム係数ロジットモデルの推定

```
rcdclogit <- mlogit(choice ~ price + Kinoko + Takenoko | 0,  
                      data = datalogit,  
                      panel = TRUE,  
                      rpar = c(price = "ln", Kinoko = "n", Takenoko = "n") ,  
                      R = 50,  
                      correlation = FALSE)
```

- 引数：
  - **panel**: 今回は各消費者について複数回の選択を観察しているのでパネルデータ構造となっている。
  - **rpar**: ランダム係数の定式化。**ln**は対数正規分布、**n**は正規分布
  - **R**: モンテカルロ積分における乱数のドロ一数。多いほうが望ましいが、計算時間の関係上ひとまず50としておく。(要追加説明)
  - **correlation**: ランダム係数の間の相関構造。多くの場合は独立(つまり **FALSE**)

```
stargazer::stargazer(multilogit, rcdclogit, type="text")
```

## 【参考】乱数のドロ一数

- 亂数のドロ一数を便宜上50としているが、これは少なすぎる。
- ドロ一数が多いとより正確。ただし、計算時間(推定時間)が長くなる。
- 推定結果がドロ一数によってあまり変化しなくなるくらい大きい値にするのが重要。
- なお、ガウス求積など、正確な値を得つつ、計算負荷を下げる方法もある。(興味ある人は個人的に)

```
##  
## =====  
## choice  
## (1) (2)  
## -----  
## price 0.058*** 0.154***  
## (0.004) (0.011)  
##  
## Kinoko 11.685*** 32.007***  
## (0.726) (2.275)  
##  
## Takenoko 12.224*** 32.904***  
## (0.741) (2.302)  
##  
## sd.price -0.021***  
## (0.002)  
##  
## sd.Kinoko 1.844***  
## (0.337)  
##  
## sd.Takenoko 3.883***  
## (0.350)  
##  
## -----  
## Observations 1,190 1,190  
## Log Likelihood -1,077.499 -750.039  
## =====  
## Note: *p<0.1; **p<0.05; ***p<0.01
```

# 推定したランダム係数の分布

- まず、 $\alpha_{i,Kinoko} - \alpha_{i,Takenoko}$  という選好パラメタの差をプロットする。
- 平均及び分散は、

$$E[\alpha_{i,Kinoko} - \alpha_{i,Takenoko}] = E[\alpha_{i,Kinoko}] - E[\alpha_{i,Takenoko}]$$

$$\text{Var}[\alpha_{i,Kinoko} - \alpha_{i,Takenoko}] = \text{Var}[\alpha_{i,Kinoko}] + \text{Var}[\alpha_{i,Takenoko}]$$

ここで、 $\alpha_{i,Kinoko}$  と  $\alpha_{i,Takenoko}$  は独立と仮定している。

# 【R分析】 ランダム係数の分布

- `mlogit::rpar`関数を使うと、ランダム係数の分布パラメタを取得できる。

```
dist_Kinoko = rpar(rcdclogit, 'Kinoko')
dist_Takenoko = rpar(rcdclogit, 'Takenoko')
dist_price = rpar(rcdclogit, 'price')
```

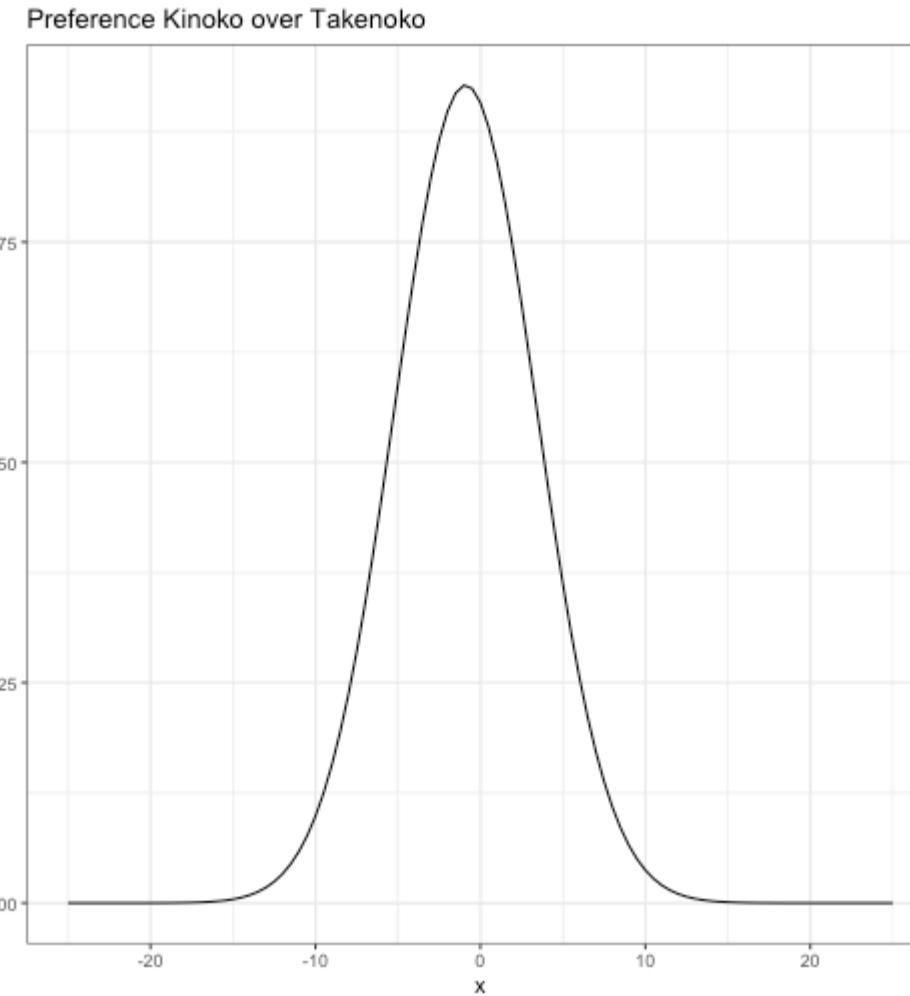
# 【R分析】 ランダム係数の分布

- 推定値から、 $\beta_{i,Kinoko} - \beta_{i,Takenoko}$  の分布プロットを作成する。

```
p1 <- ggplot(data = data.frame(x = c(-25, 25)), aes(x)) +
  stat_function(fun = dnorm, n = 101,
                args = list(mean = dist_Kinoko$mean -dist_Takenoko$mean ,
                            sd = sqrt(dist_Kinoko$sigma^2 + dist_Takenoko$sigma^2) ) ) +
  ylab("") + theme_bw() +
  ggtitle("Preference Kinoko over Takenoko")
```

```
plot(p1)
```

# ランダム係数の分布： $\beta_{i,Kinoko} - \beta_{i,Takenoko}$



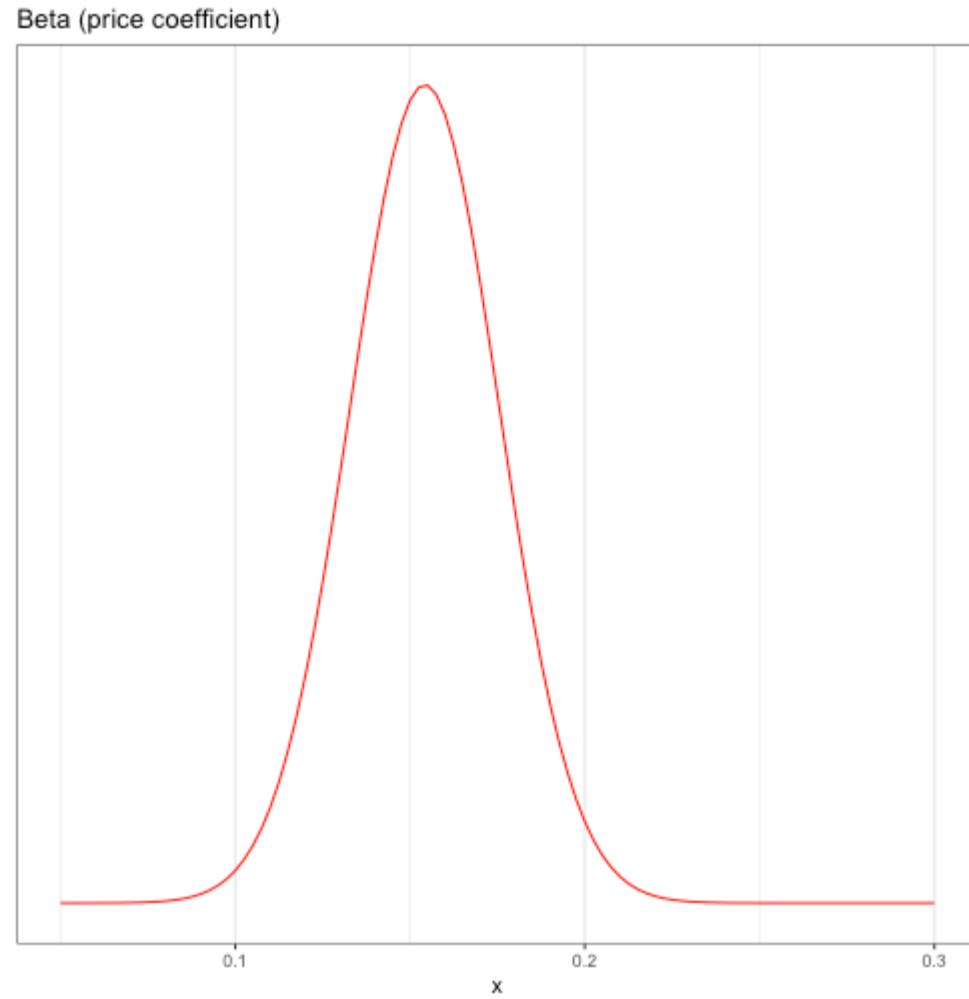
# 【R分析】価格係数の分布

- 価格係数  $\beta_i \sim \log N(\theta_\beta, \sigma_\beta^2)$  についてプロットする。

```
p2 <- ggplot(data = data.frame(x = c(0.05, 0.3) ), aes(x)) +
  stat_function(fun = dnorm, n = 101,
                args = list(mean = dist_price$mean, sd = abs(dist_price$sigma)), colour = "red" ) +
  ylab("") + theme_bw() +
  scale_y_continuous(breaks = NULL) + ggtitle("Beta (price coefficient)")
```

```
plot(p2)
```

# 価格係数の分布



## 応用例：需要曲線のプロット

# 需要曲線のプロット

- ランダム係数ロジットモデル(消費者属性なし)の推定結果にもとづいて、需要曲線をプロットする。
- 解釈：推定しているものは消費者の選択確率のモデル。しかしながら、選択確率=市場シェアと解釈可能。
- 選択確率(市場シェア)に潜在的な消費者数をかけ合わせると、市場全体の需要が得られる。

# 設定

- 1000人の消費者がいると考える。
- 各消費者の選好パラメタは推定された分布に従っており、ランダムに決まっている。
- 各消費者の購買確率を計算し、それを足し合わせることで、全体の需要を予測する。

# 【R分析】下準備

- 推定した分布パラメタにもとづいて、個々の消費者のパラメタをドローする。

```
# 以下では乱数を用いるので乱数のシードを固定
set.seed(101)

# 消費者数
R = 1000

# 消費者のパラメタを、乱数によって発生させる。
alpha_Kinoko_vec = rnorm(n = R, mean = dist_Kinoko$mean, sd = abs(dist_Kinoko$sigma) )
alpha_Takenoko_vec = rnorm(n = R, mean = dist_Takenoko$mean, sd = abs(dist_Takenoko$sigma) )
beta_vec = rnorm(n = R, mean = dist_price$mean, sd = abs(dist_price$sigma) )
```

# 乱数について

- 今、1000人の消費者が異なったパラメタをもっており、そのパラメタは推定した分布に従う。
- 1000人のパラメタについて、その分布に従う**乱数**として発生させる。
  - `rnorm` と `rlnorm` が正規乱数・対数正規乱数
- 【重要】乱数を発生させる際には、乱数のシードを固定する。
  - `set.seed(101)` に相当する。
  - これをしないと、分析を回す度に異なった乱数が生成される。(再現性の観点から重要)

# 【R分析】ロジット確率計算のための関数

- パラメタ  $\beta, \alpha_{Kinoko}, \alpha_{Takenoko}$  と価格を与えることで、選択確率を計算する。

```
f_logit_prob <- function(alpha_Kinoko, alpha_Takenoko, beta, price_Kinoko, price_Takenoko){  
  util_Kinoko <- alpha_Kinoko - beta*price_Kinoko  
  util_Takenoko <- alpha_Takenoko - beta*price_Takenoko  
  
  prob_Kinoko <- exp( util_Kinoko ) / ( 1 + exp( util_Kinoko ) + exp( util_Takenoko ) )  
  prob_Takenoko <- exp( util_Takenoko ) / ( 1 + exp( util_Kinoko ) + exp( util_Takenoko ) )  
  prob_Other <- 1 - (prob_Kinoko + prob_Takenoko)  
  
  return( cbind(prob_Kinoko, prob_Takenoko, prob_Other))  
}
```

# 【R分析】需要関数

```
f_demand <- function( alpha_Kinoko_vec, alpha_Takenoko_vec, beta_vec, price_Kinoko, price_Takenoko )  
  
R = length(alpha_Kinoko_vec) # Number of consumers  
  
# 結果を保存するベクトルを事前に準備  
prob_Kinoko = numeric(R)  
prob_Takenoko = numeric(R)  
prob_Other = numeric(R)  
  
for (r in 1:R){  
    result = f_logit_prob( alpha_Kinoko_vec[r], alpha_Takenoko_vec[r], beta_vec[r], price_Kinoko,  
    prob_Kinoko[r] = result[1]  
    prob_Takenoko[r] = result[2]  
    prob_Other[r] = result[3]  
}  
  
# 選択確率をすべての消費者について足し合わせて、各オプションの需要を得る。  
return( c(sum(prob_Kinoko), sum(prob_Takenoko), sum(prob_Other)))  
}
```

# 【R分析】きのこの需要関数

- たけのこの価格を200円に固定したときの、きのこの需要関数を求めてみる。

```
price_Takenoko = 200

# きのこの価格を100円から250円まで動かす。
price_vec = seq(from = 100, to = 250, by = 5)
kinoko_vec = numeric(length(price_vec))

# ループで需要を計算
for ( i in 1:length(price_vec)){
  result <- f_demand( alpha_Kinoko_vec, alpha_Takenoko_vec,
                      beta_vec, price_vec[i], price_Takenoko = 200 )
  kinoko_vec[i] <- result[1]
}

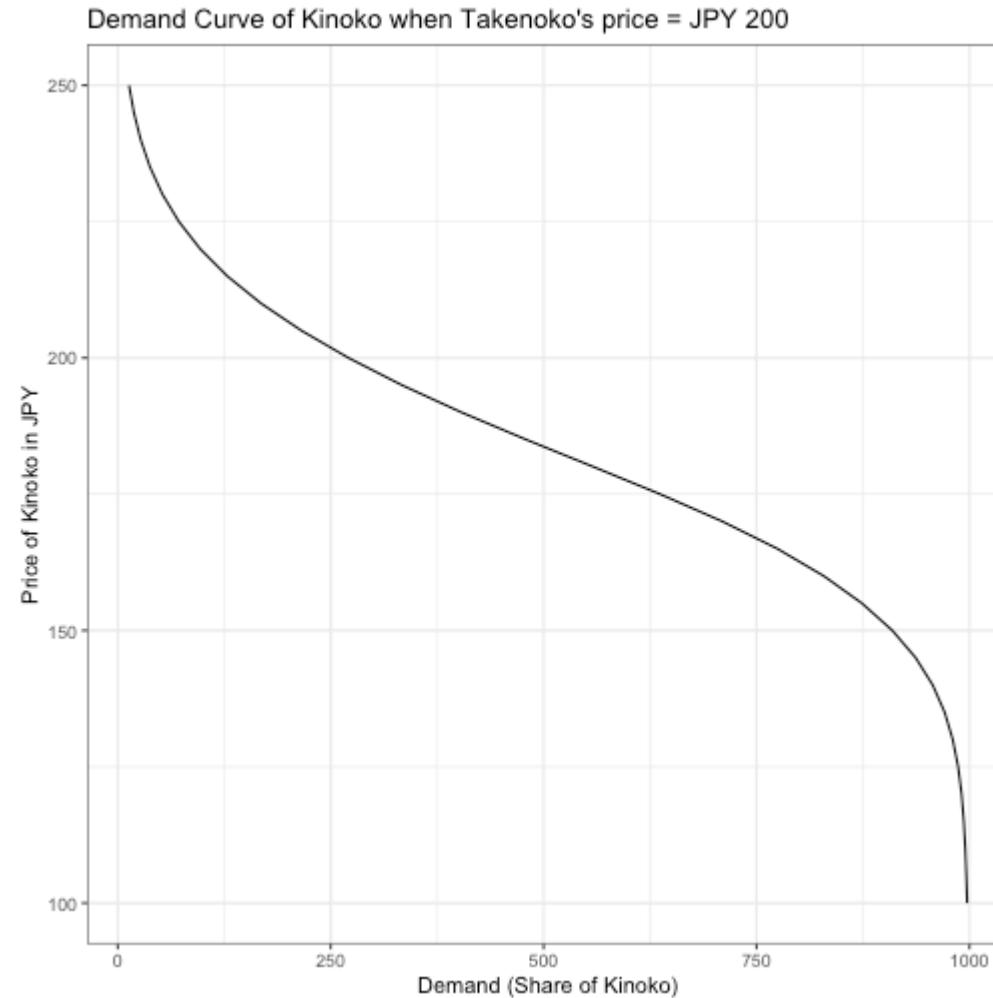
# データフレームに結果を保存
data_demand_kinoko = tibble( price = price_vec,
                             demand = kinoko_vec,
                             revenue = price_vec*kinoko_vec)
```

# 【R分析】プロット

```
fig_demand <- ggplot(data = data_demand_kinoko, aes(x = demand, y = price) ) +  
  geom_line() +  
  xlab("Demand (Share of Kinoko)") +  
  ylab("Price of Kinoko in JPY") + theme_bw() +  
  ggtitle("Demand Curve of Kinoko when Takenoko's price = JPY 200")
```

```
plot(fig_demand)
```

# 推定された需要関数



# 【R分析】 収入関数のプロット

```
fig_rev <- ggplot(data = data_demand_kinoko, aes(x = price, y = revenue) ) +  
  geom_line() +  
  xlab("Price of Kinoko in JPY") +  
  ylab("Revenue of Kinoko in JPY") + theme_bw() +  
  ggtitle("Revenue Curve of Kinoko when Takenoko's price = JPY 200")
```

```
plot(fig_rev)
```

# 推定された収入関数

Revenue Curve of Kinoko when Takenoko's price = JPY 200

