

# 離散選択モデル Part 2: きのこの山 VS たけのこの里

講師: 遠山祐太

最終更新: 2025-01-20

# イントロダクション

# イントロダクション

- 内容：
  1. 事例：きのこの山 VS たけのこの里
  2. コード：`mlogit`パッケージを使った推定

**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います。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：
  1. (きのこの山、たけのこの里) = (200円、200円)
  2. (きのこの山、たけのこの里) = (180円、200円)
  3. (きのこの山、たけのこの里) = (200円、170円)
  4. (きのこの山、たけのこの里) = (220円、200円)
  5. (きのこの山、たけのこの里) = (190円、210円)
- 選択肢：
  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。

# 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。



# 結果を見る前に：選択型コンジョイント分析

- 今から見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。
- 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、顕示選好データ(Revealed preference) と呼ぶ。
  - POS データ、購買履歴データ、など。
- 表明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")

# Cleaning
data <- data %>%
  filter(experience != "4 : 食べたことがない" ) %>%
  drop_na() # NA(未回答?)の行を除外

# filter(experience != "4 : 食べたことがない" & gender != "3 : 回答したくない") %>%
```

# 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
```

*# Latex用出力*

```
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, "latex") %>%  
  kableExtra::save_kable(tab, file = "output/tab_choice_share.tex")
```



# 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	0.21
Q2: (180円, 200円)	0.57	0.27	0.16
Q3: (200円, 170円)	0.10	0.80	0.11
Q4: (220円, 200円)	0.07	0.63	0.30
Q5: (190円, 210円)	0.51	0.27	0.22

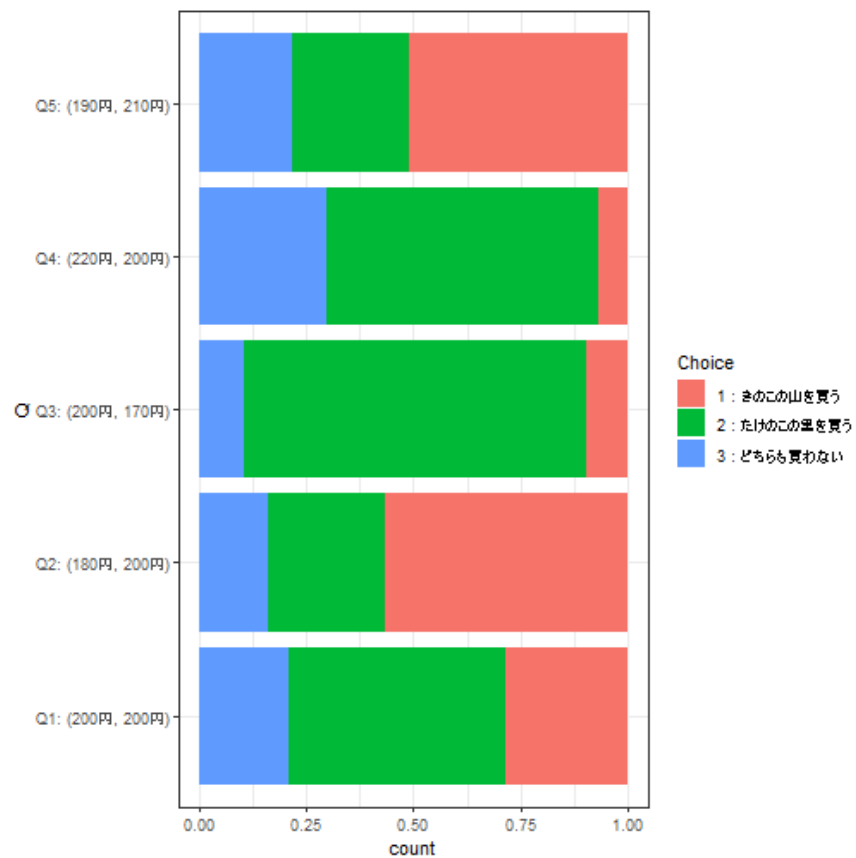
# 【R分析】 グラフで出力

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") + theme_bw()
```

```
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	0.21
Q2: (180円, 200円)	0.57	0.27	0.16
Q3: (200円, 170円)	0.10	0.80	0.11
Q4: (220円, 200円)	0.07	0.63	0.30
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

$$U_{i,k,Kinoko} = \beta_{Kinoko} - \alpha \cdot p_{k,Kinoko} + \epsilon_{i,k,Kinoko}$$

$$U_{i,k,Takenoko} = \beta_{Takenoko} - \alpha \cdot p_{k,Takenoko} + \epsilon_{i,k,Takenoko}$$

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

- $\beta_{Kinoko}, \beta_{Takenoko}$  : きのこ・たけのこ自体への好み
  - $\alpha$  : 価格への反応度
  - $p_{k,j}$  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  $\beta_{Kinoko}, \beta_{Takenoko}, \alpha$  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

$$P_k(j|\theta) = \frac{\exp(\beta_j - \alpha \cdot p_{j,k})}{1 + \exp(\beta_{Kinoko} - \alpha \cdot p_{Kinoko,k}) + \exp(\beta_{Takenoko} - \alpha \cdot p_{Takenoko,k})}$$

- 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におけるインデックス情報

```
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_atr
```

```
# 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	0.34
	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	0.05
	関東	0.86
	関西	0.09

# 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と・たけのこ係数が、消費者属性  $z_i$  に依存すると考える。

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

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

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

- 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_atr <- 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_atr <- mlogit(formula = with_atr,
                          data = datalogit,
                          relevel = '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_atr,  
                    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

```



# 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？

$$U_{i,k,Kinoko} = \beta_{i,Kinoko} - \alpha_i \cdot p_{k,Kinoko} + \epsilon_{i,k,Kinoko}$$

$$U_{i,k,Takenoko} = \beta_{i,Takenoko} - \alpha_i \cdot p_{k,Takenoko} + \epsilon_{i,k,Takenoko}$$

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

- ここで、
  - $\beta_{i,j} \sim N(\theta_j, \sigma_j^2)$  for  $j \in \{Kinoko, Takenoko\}$
  - $\alpha_i \sim N(\theta_\alpha, \sigma_\alpha^2)$ .
- 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)

# 【R分析】 ランダム係数ロジットモデルの推定

```
# 乱数固定
set.seed(100)

# 推定
rcdclogit <- mlogit(choice ~ price + Kinoko + Takenoko | 0,
  data = datalogit,
  panel = TRUE,
  rpar = c(price = "n", Kinoko = "n", Takenoko = "n") ,
  R = 100,
  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ー数。
  - `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)

# 推定時間

```
# 乱数固定
set.seed(100)

system.time(
  rcdclogit <- mlogit(choice ~ price + Kinoko + Takenoko | 0,
    data = datalogit,
    panel = TRUE,
    rpar = c(price = "n", Kinoko = "n", Takenoko = "n") ,
    R = 100,
    correlation = FALSE)
)
```

```
## ユーザ システム 経過
## 11.08 0.41 11.41
```

# 推定結果

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

## 【参考】乱数のドロー数

- 乱数のドロー数を便宜上100としているが、これは少なすぎる。
- ドロー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。
- 推定結果がドロー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)

# 推定結果(ロジットとランダム係数ロジット)

```
##
## =====
##                   choice
##                   (1)         (2)
## -----
## price              0.058***    0.204***
##                   (0.004)      (0.013)
##
## Kinoko             11.685***    41.378***
##                   (0.726)      (2.595)
##
## Takenoko           12.224***    42.660***
##                   (0.741)      (2.629)
##
## sd.price              -0.040***
##                       (0.004)
##
## sd.Kinoko            1.520***
##                       (0.399)
##
## sd.Takenoko          5.306***
##                       (0.444)
##
```

## (おまけ) 推定結果の外部ファイル保存

```
writeLines(capture.output(  
  stargazer::stargazer(multilogit, rcdclgit, type="latex", dep.var.caption = "")  
), "output/tab_estimated_parameter.tex")
```



# 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、

$$\begin{aligned} \mathbf{E}[\beta_{i,Kinoko} - \beta_{i,Takenoko}] &= \mathbf{E}[\beta_{i,Kinoko}] - \mathbf{E}[\beta_{i,Takenoko}] \\ \mathbf{Var}[\beta_{i,Kinoko} - \beta_{i,Takenoko}] &= \mathbf{Var}[\beta_{i,Kinoko}] + \mathbf{Var}[\beta_{i,Takenoko}] \end{aligned}$$

なお、 $\beta_{i,Kinoko}$  と  $\beta_{i,Takenoko}$  は独立という仮定を利用している。

# 【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')
```

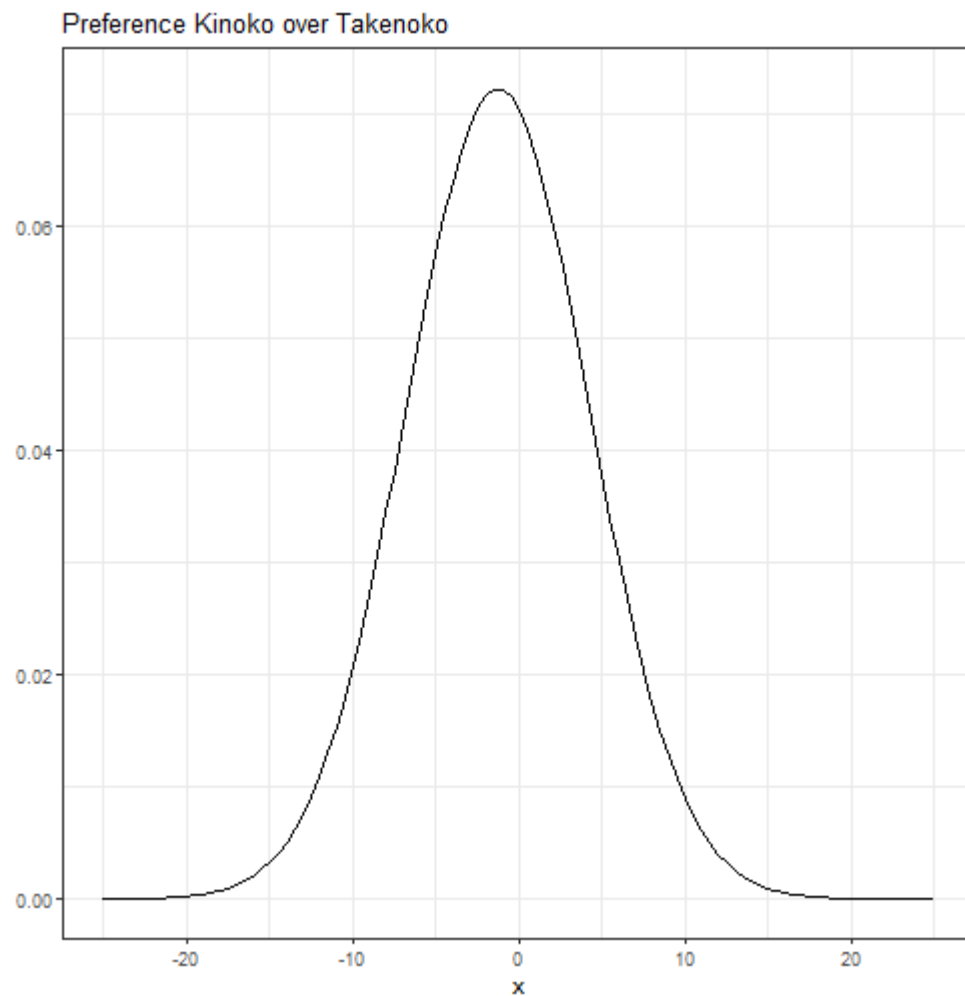
# 【R分析】 ランダム係数の分布

- 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")  
ggsave(file = "output/fig_dist_K_over_T.pdf", plot = p1)
```

```
plot(p1)
```

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



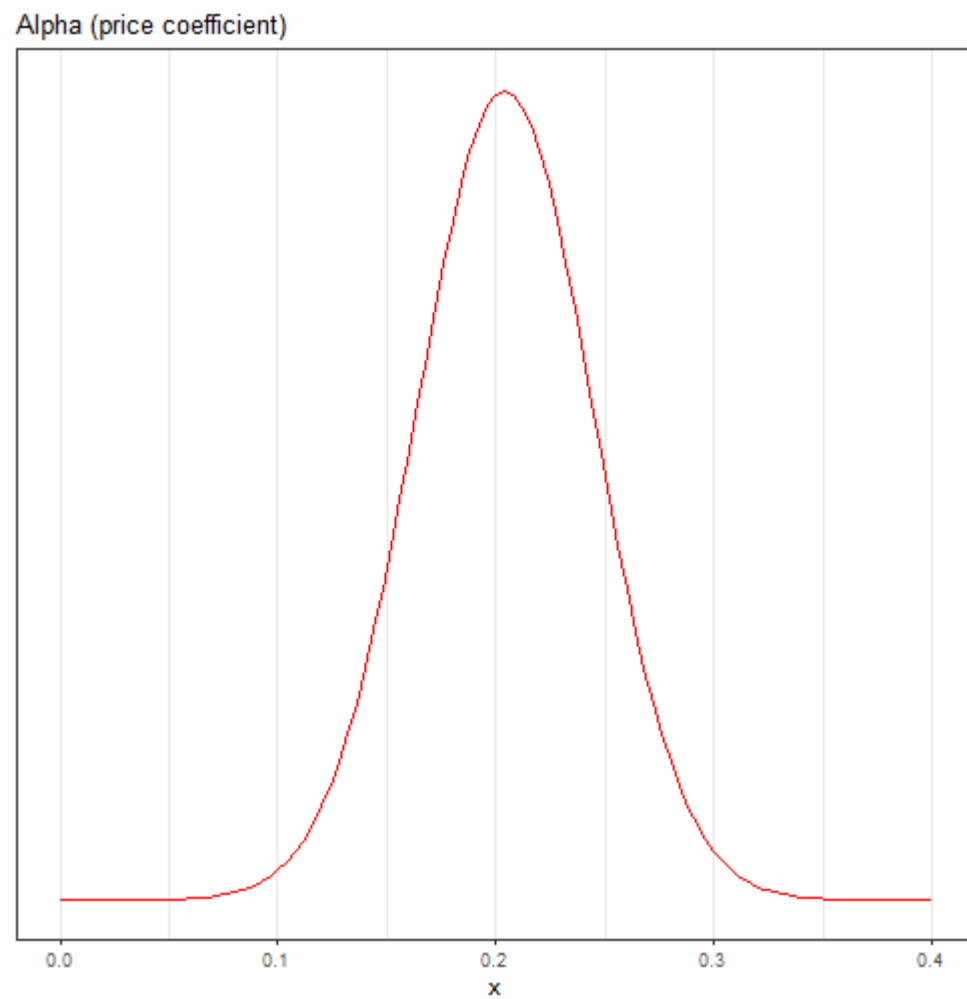
# 【R分析】 価格係数の分布

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

```
p2 <- ggplot(data = data.frame(x = c(0, 0.4) ), 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("Alpha (price coefficient)")  
ggsave(file = "output/fig_dist_beta.pdf", plot = p2)
```

```
plot(p2)
```

# 価格係数の分布



# 支払い意思額(Willingness-to-pay, WTP)

- 各消費者のきのこ・たけのこへの支払い意思額は  $\$ \beta_{i, \text{Kinoko}} / \alpha_i \$$  で表される。
- 支払い意思額の分布をプロットしよう。

# WTPの計算

```
# 乱数固定
set.seed(100)

# パラメタについての乱数を発生
dt_brand =
  tibble( beta_kinoko_draw = rnorm(30000, mean = dist_Kinoko$mean, sd = abs(dist_Kinoko$sigma) ),
          beta_takenoko_draw = rnorm(30000, mean = dist_Takenoko$mean, sd = abs(dist_Takenoko$sigma) ),
          alpha_draw = rnorm(30000, mean = dist_price$mean, sd = abs(dist_price$sigma) ) )

# 支払い意思額の計算
dt_brand %>%
  mutate( brand_Kinoko = beta_kinoko_draw / alpha_draw,
          brand_Takenoko = beta_takenoko_draw / alpha_draw,
          love_Kinoko = ifelse(beta_kinoko_draw > beta_takenoko_draw, 1, 0 ) ) -> dt_brand
```



## 続・WTPの計算

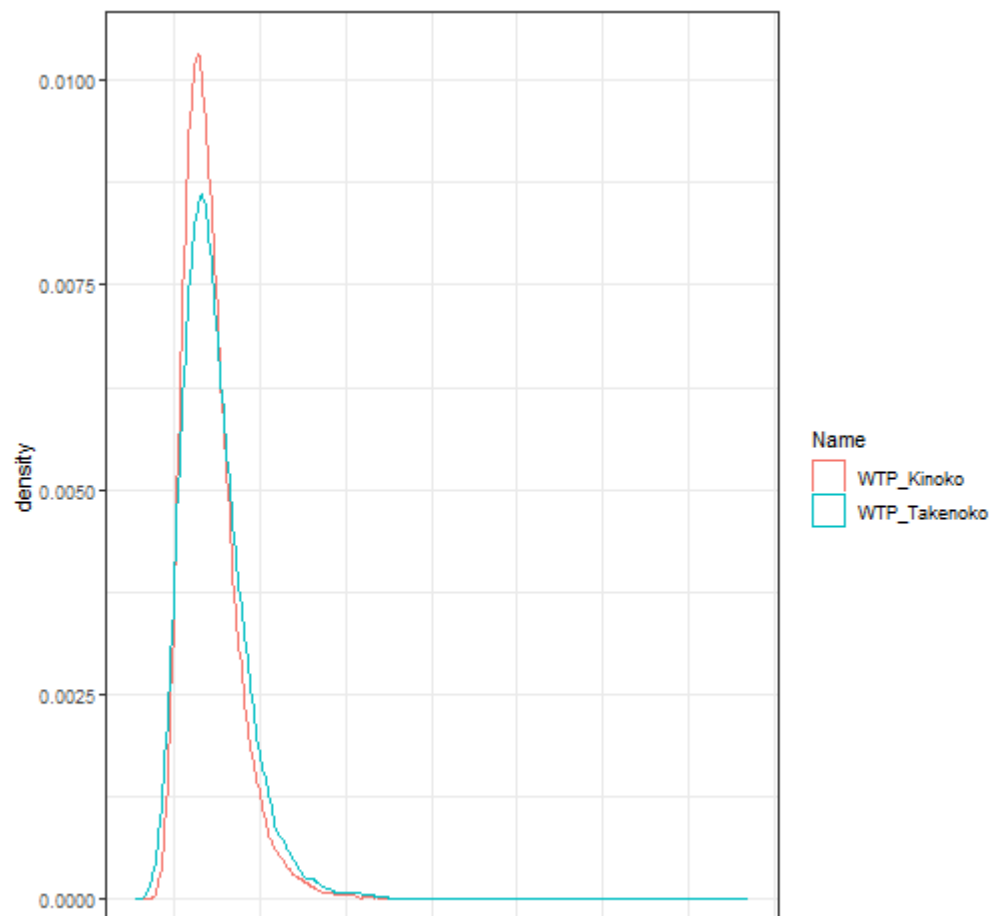
```
dt_brand %>%
  select( brand_Kinoko, brand_Takenoko) %>%
  rename( WTP_Kinoko = brand_Kinoko,
          WTP_Takenoko = brand_Takenoko) %>%
  pivot_longer( cols = everything(), names_to = "Name", values_to = "WTP") -> dt

dt %>%
  ggplot( aes(x = WTP, colour = Name)) + theme_bw() +
  geom_density( ) -> fig_brand

ggsave(filename = "output/brand_value.pdf", plot = fig_brand, width = 12, height = 6)
```

# WTPのプロット

```
plot(fig_brand)
```



# 需要曲線の構築

# 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、 $\text{選択確率} = \text{市場シェア}$ と解釈可能。
- $\text{選択確率(市場シェア)} \times \text{潜在的な消費者数}$ 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。
beta_Kinoko_vec = rnorm(n = R, mean = dist_Kinoko$mean, sd = abs(dist_Kinoko$sigma) )
beta_Takenoko_vec = rnorm(n = R, mean = dist_Takenoko$mean, sd = abs(dist_Takenoko$sigma) )
alpha_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  $\beta, \beta_{Kinoko}, \beta_{Takenoko}$  と 価格を与えることで、選択確率を計算する。

```
f_logit_prob <- function(beta_Kinoko, beta_Takenoko, beta, price_Kinoko, price_Takenoko){  
  util_Kinoko <- beta_Kinoko - beta*price_Kinoko  
  util_Takenoko <- beta_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( beta_Kinoko_vec, beta_Takenoko_vec, alpha_vec, price_Kinoko, price_Takenoko )  
  
  R = length(beta_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( beta_Kinoko_vec[r], beta_Takenoko_vec[r], alpha_vec[r], price_Kinoko, price_Takenoko )  
    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( beta_Kinoko_vec, beta_Takenoko_vec,
                    alpha_vec, price_vec[i], price_Takenoko = 200 )
  kinoko_vec[i] <- result[1]
}

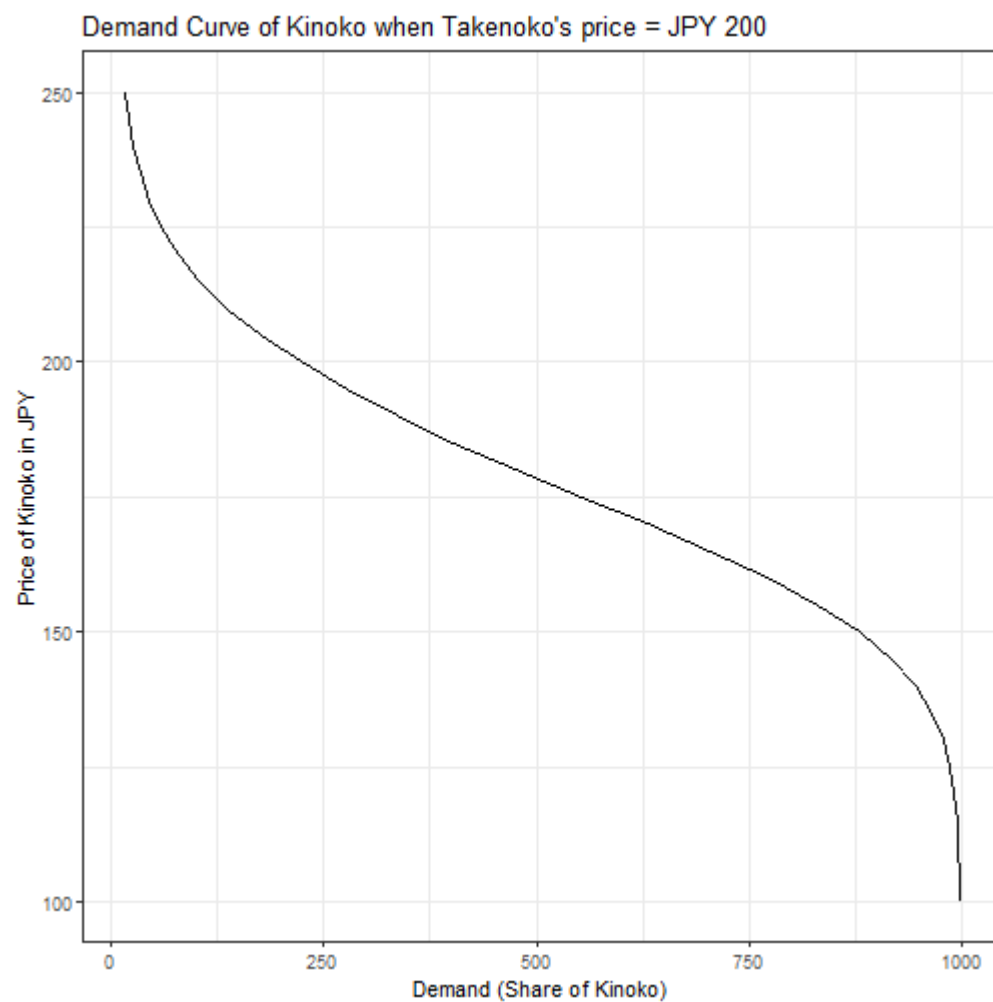
# データフレームに結果を保存
data_demand_kinoko = tibble( price = price_vec,
                             demand = kinoko_vec,
                             revenue = price_vec*kinoko_vec)
```

# 【R分析】プロット

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

```
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() + theme_bw() +  
  xlab("Price of Kinoko in JPY") +  
  ylab("Revenue of Kinoko in JPY") +  
  ggtitle("Revenue Curve of Kinoko when Takenoko's price = JPY 200")  
ggsave(filename = "output/revenue_kinoko.pdf", plot = fig_rev)
```

```
plot(fig_rev)
```

# 推定された収入関数

