

Rの導入と行動モデル

理論勉強会#3 2015/5/1

M2 笠原 和

2 Rの導入

Rとは

- 有名な統計言語「S言語」をオープンソースとして実装しなおした統計解析ソフト
- モデル推定, グラフィック表示や簡単な計算の際に使用する
- 様々なOSに対応しており自由にダウンロードできる

3

インストールしてみよう

RのDLは以下のリンクから

- R

<http://o-server.main.jp/r/install.html>



- R Studio (統合開発環境)

http://memorandum2015.sakura.ne.jp/index_rstudio.html



インストール方法

http://www.cwk.zaq.ne.jp/fkHUD708/files/R-prg-intro/R-prg-intro_01.pdf

or 行動モデル夏の学校の資料

4

基本事項

The image shows a screenshot of the RStudio software interface. The interface is divided into four main panels, each highlighted with a red dashed border and containing Japanese text annotations:

- Top Left Panel (Source Editor):** Contains R code. The annotation reads: **プログラムを書く場所** (Place to write programs) and **ソース画面** (Source screen).
- Top Right Panel (Environment/History):** Shows the Environment pane with variables 'a', 'b', and 'c' listed. The annotation reads: **作成された変数リスト** (List of created variables) and **実行プログラム履歴** (Execution program history).
- Bottom Left Panel (Console):** Shows the output of the R code from the source editor. The annotation reads: **実行結果が表示される場所** (Place where execution results are displayed) and **コンソール画面** (Console screen).
- Bottom Right Panel (Files/Plots/Packages/Help/Viewer):** Shows the Files pane. The annotation reads: **グラフの表示場所** (Place to display graphs) and **ファイルやパッケージ情報** (File or package information).

```
1  
2 #コメントの代入  
3 # "<->"の代入  
4  
5 a<-1 # 整数を代入  
6  
7 b<-c(1,2,3,4,5) #ベクトルを代入  
8  
9 c <- matrix(c(1,2,3,4,5,6), nrow=2, ncol=3) #行列を代入
```

```
> a  
[1] 1  
> b  
[1] 1 2 3 4 5  
> c  
      [,1] [,2] [,3]  
[1,] 1     3     5  
[2,] 2     4     6  
>  
> b^2  
[1] 1 4 9 16 25  
>  
> b+5  
[1] 6 7 8 9 10  
>  
> b^2 + 5*b + 3  
[1] 9 17 27 39 53  
>
```

5

基本事項～ベクトル処理～

The screenshot shows the RStudio interface with a script editor containing R code. The 'Run' button is highlighted with a red box. Red text annotations explain the execution process. The console shows the output of the code, with blue text annotations explaining the data types returned.

コンパイルはここ
Or
[Ctrl] + [Enter]

```
1  
2 #コメントアウト  
3 # "<" : 代入  
4  
5 a<-1 # 整数を代入  
6  
7 b<-c(1,2,3,4,5) #ベクトルを代入  
8  
9 c <- matrix(c(1,2,3,4,5,6), nrow=2, ncol=3) #行列を代入
```

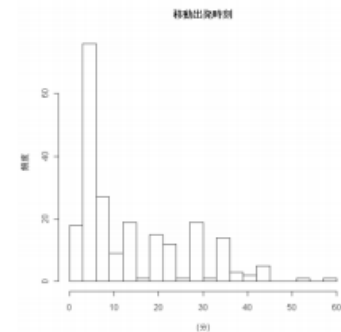
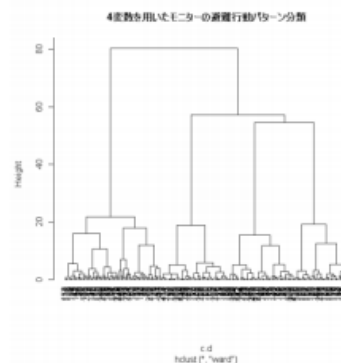
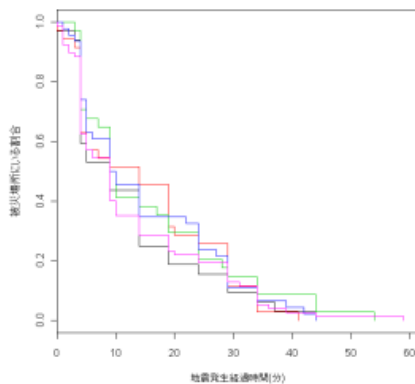
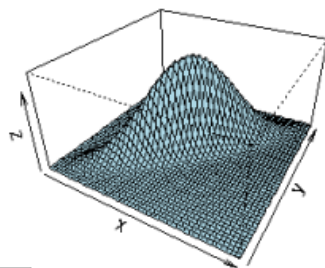
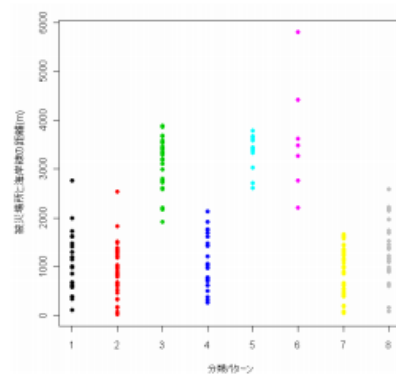
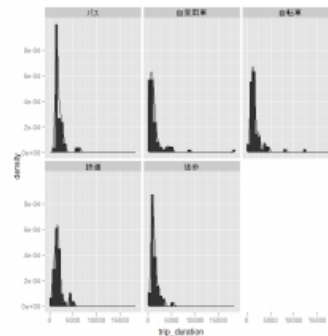
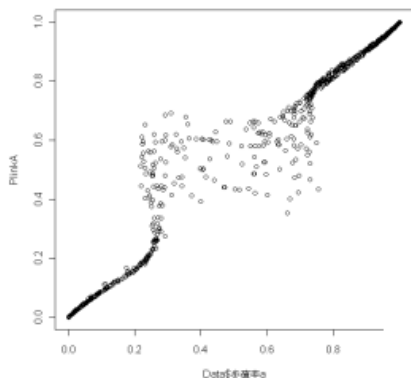
←a : 変数で返す
←b : ベクトルで返す
←c : 行列で返す
←ベクトルごとの計算も可

Environment History
Import Dataset
Clear Environment
Global Environment
Data
c num [1:2, 1:3...]
data 66 obs. of 6 ...
values
a 1
b num [1:5] 1 2 3...

Console ~/ ↻
> a
[1] 1
> b
[1] 1 2 3 4 5
> c
[,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 6
>
> b^2
[1] 1 4 9 16 25
>
> b+5
[1] 6 7 8 9 10
>
> b^2 + 5*b + 3
[1] 9 17 27 39 53
>

6

基本事項～グラフ作成～



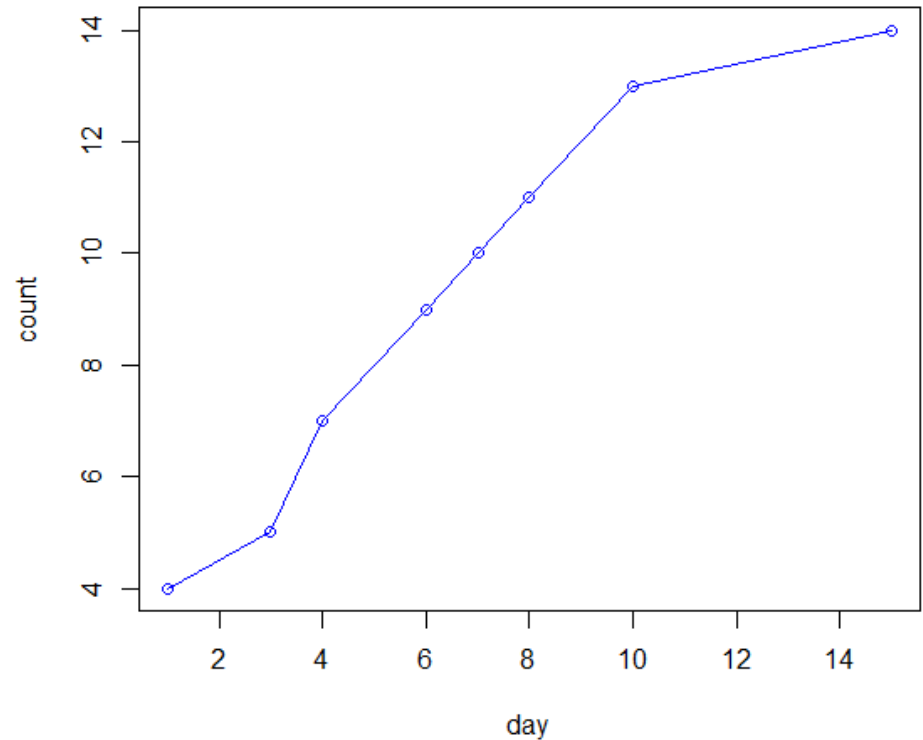
グラフの作成に役立ったりもします

7

基本事項～グラフ作成～

```
1 ##plot
2 x1<-c(1,3,4,6,7,8,10,15)
3 y1<-c(4,5,7,9,10,11,13,14)
4 y2<-c(1,3,4,6,7,8,10,15)
5
6 plot(x1,y2)
7 plot(x1,y1,type="p")
8 plot(x1,y1,type="l")
9 plot(x1,y1,type="o",col="blue",xlab="day", ylab="count")
10
```

- 大量のグラフを作りたいとき
- 同じ形式でグラフを作成したいとき



8 R参考サイト

- R tips

<http://cse.naro.affrc.go.jp/takezawa/r-tips/r.html>

-基本的な使い方はここで大体分かる

- R tipsのまとめ資料

<http://cse.naro.affrc.go.jp/takezawa/r-tips.pdf>

- R wiki

<http://www.okada.jp.org/RWiki/>

-日本最大のRコミュニティー

-新規パッケージはここでチェック

-情報量が多いのでサイト内検索で効率的に必要な情報にアクセスする

- Use R

<http://bin.t.u-tokyo.ac.jp/~hara/r.html>

-羽藤研OBによるRのメモ



ここからが本番です！
行動モデル

10

行動モデル編

行動モデルにも色々あります.

- 連続量のモデル
例)自動車の走行距離, 活動時間
- 離散量のモデル
例)交通手段, 目的地, 経路, トリップの頻度, 出発時間

今日の課題

- 今回のゼミで紹介したMNLモデル(多項ロジットモデル)を使って交通手段モデルを推定します.

11 推定に使うデータについて

- プロブパーソンデータ(PPデータ)
GPS搭載携帯電話と、インターネットを通じたWEBダイアリー によって、モニターの1日の行動履歴および移動中の位置情報データ

データクリーニング後の推定用データ

	A	B	D	E	F	I	J	L	Y	AA	AB	AD	AE	AF	AG
1	基礎データ					トリップデータ			個人属性など			選択結果		自家用車(100)	
2	トリップID	モニターID	目的	出発日時	到着日時	LatO	LonO	shikuname	年齢	性別	自宅位置	代表交通手段	選択結果時	代替手段	生距離car
3	255461	yd021	散歩・回遊	2009/10/31 06:19:00	2009/10/31 07:50:01	35.34554	139.4865	藤沢市	46	男	神奈川県茅ヶ崎市	自転車	108	1	26051
4	256674	yd027	買い物	2009/11/13 19:08:13	2009/11/13 20:13:43	35.44361	139.6377	中区	38	男	東京都世田谷区	鉄道	55.3	1	25494
5	255291	yd025	業務	2009/10/29 07:36:03	2009/10/29 08:21:20	35.36674	139.5715	栄区	39	女	神奈川県横浜市栄区	鉄道	38.6	1	13775
6	256382	yd021	出勤	2009/11/10 06:55:33	2009/11/10 07:58:02	35.32551	139.4091	茅ヶ崎市	46	男	神奈川県茅ヶ崎市	鉄道	49.8	1	30728
7	257459	yd028	娯楽	2009/11/21 10:40:54	2009/11/21 12:31:44	35.46426	139.5309	旭区	37	男	神奈川県横浜市旭区	鉄道	76.2	1	54540
8	257040	yd025	食事	2009/11/17 12:09:25	2009/11/17 12:52:15	35.44344	139.637	中区	39	女	神奈川県横浜市栄区	鉄道	11.2	1	1555
9	256978	yd018	帰宅	2009/11/16 20:41:50	2009/11/16 21:43:21	35.42393	139.3185	厚木市	54	男	神奈川県横浜市神奈	自動車	77.2	1	38923
10	256452	yd027	その他	2009/11/10 19:38:24	2009/11/10 19:46:20	35.63107	139.6799		38	男	東京都世田谷区	自転車	5.6	1	1864
11	256965	yd024	帰宅	2009/11/16 18:47:37	2009/11/16 19:45:39	35.44457	139.6368	中区	40	男	神奈川県川崎市多摩	鉄道	68.5	1	28208
12	256459	yd012	帰宅	2009/11/10 22:01:07	2009/11/10 23:03:18	35.4225	139.3166	厚木市	46	男	神奈川県横浜市西区	自動車	73.3	1	37380

	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD
	鉄道(200)											自転車(410)		徒歩(420)			バス(240)					
代替手	所要時間tr	費用train	出発駅	到着駅	乗換回	アクセス距	アクセス時	イグレス距	イグレス時	総所要時間	代替手段生距離bike	所要時間bil	代替手段生距離walk	所要時間wa	代替手段生距離bus	所要時間bu	費用bus	アクセス距				
0	50	570	藤沢本	中山	3	0	0	0	0	0	1	21607	108	1	21607	270.4	0	0	0	0	0	
1	40	440	関内	学芸大	1	450	5.4	817	9.9	55.3	1	23586	117.9	1	23586	293.1	0	0	0	0	0	
4	18	210	港南台	関内	0	1320	16.8	311	3.8	38.6	1	11604	58	1	11604	143.4	0	0	0	0	0	
6	37	570	茅ヶ崎	関内	1	717	9	313	3.8	49.8	1	27661	138.3	1	27661	342.3	0	0	0	0	0	
7	72	3660	二俣川	箱根板	4	258	3.1	87	1.1	76.2	1	48214	241.1	1	48214	595.9	0	0	0	0	0	
1	2	200	関内	桜木町	0	382	4.6	394	4.6	11.2	1	899	4.5	1	899	10.8	1	0.65	3	210	8	
1	61	680	愛甲石	白楽	2	3065	37.3	668	8.8	107.1	1	33859	169.3	1	33859	424.6	0	0	0	0	0	
0	0	0			0	0	0	0	0	0	1	1126	5.6	1	1126	13.9	0	0	0	0	0	
1	45	550	馬車道	登戸	1	855	10.2	1091	13.3	68.5	1	23435	117.2	1	23435	290.2	0	0	0	0	0	
1	57	450	愛甲石	平沼橋	2	3471	42.1	172	2.1	101.2	1	32014	160.1	1	32014	399.9	0	0	0	0	0	
1	61	660	千歳船	戸部	3	1630	20	1121	13.9	94.9	1	23881	119.4	1	23881	294.8	0	0	0	0	0	
0	0	0			0	0	0	0	0	0	1	1220	6.1	1	1220	15.2	0	0	0	0	0	
1	3	180	みなとみ	横浜	0	1225	14.7	314	3.8	21.5	1	2597	13	1	2597	31.1	0	0	0	0	0	

各交通手段のLOS(Level Of Service)や説明変数となりそうなデータは自分で作る

12

推定の手順

■ データファイルの読み込み

Rで推定する際はcsvファイルでデータを読み込みます。

```
1 ## 全変数消去
2 rm(list=ls())
3
4 ### Multinomial Logit model estimation
5 ### データファイルの読み込み
6 Data <- read.csv("C:/R/Data_Clean_Japanese.csv", header=TRUE)
7 ## データ数:Data の行数を数える
8 hh <- nrow(Data)
9
10 ## パラメータの初期値の設定
11 b0 <- numeric(6)
```

←前の計算の消去

データのディレクトリ指定

タイトル行あり/なし

←0の要素が6つ並んだベクトルを生成する
推定するパラメータの数の要素分だけ設定する

13

推定の手順

```
13. ##### Logit model の対数尤度関数の定義 #####
14. fr <- function(x) {
15.   ### パラメータの宣言 :
16.   ## 定数項
17.   b1 <- x[1]
18.   b2 <- x[2]           ←ひとつの定数項は0とする(選択肢数-1)
19.   b3 <- x[3]
20.   b4 <- x[4]
21.   ## 目的地までの所要時間
22.   d1 <- x[5]
23.   ## 料金               ←説明変数は目的地までの所要時間, 料金
24.   f1 <- x[6]
25.   ## 対数尤度のための変数を宣言
26.   LL = 0                ←初期値は0
```

説明変数はクロス集計などでデータ分析を行って決める。

例)女性はいり物目的が多い, 晴れの日には徒歩移動が多い...

14

推定の手順

多項ロジットモデル **ここを計算する**

$$P_n(i) = \frac{\exp(\mu V_{in})}{\sum_{j=1}^J \exp(\mu V_{jn})}, i = 1, \dots, J$$

```
28 ### 今回用いる目的地は以下の5つ。
29 ## 鉄道(train)
30 ## バス(bus)
31 ## 自動車(car)
32 ## 自転車(bike)
33 ## 徒歩(walk)
```

```
35 ## 効用の計算:説明変数にしたい列を入れる。
```

```
36 # 定数項 # 時間 # 料金
```

代替手段生成可否:0or1

```
38 train <- Data$代替手段生成可否train * exp( d1*Data$総所要時間train/100
39         + f1*Data$費用train/100 + b1*matrix(1,nrow =hh,ncol=1))
40 bus <- Data$代替手段生成可否bus * exp( d1*Data$総所要時間bus/100
41         + f1*Data$費用bus/100 + b2*matrix(1,nrow =hh,ncol=1))
42 car <- Data$代替手段生成可否car * exp( d1*Data$所要時間car/100
43         + b3*matrix(1,nrow =hh,ncol=1))
44 bike <- Data$代替手段生成可否bike * exp( d1*Data$所要時間bike/100
45         + b4*matrix(1,nrow =hh,ncol=1))
46 walk <- Data$代替手段生成可否walk * exp( d1*Data$所要時間walk/100)
```

およその桁を揃えて推定計算を安定させるため100で割る

サンプル数(hh), 列数1, 要素がすべて1である行列を作成する

15

推定の手順

多項ロジットモデル

$$P_n(i) = \frac{\exp(\mu V_{in})}{\sum_{j=1}^J \exp(\mu V_{jn})}, i = 1, \dots, J$$

```

48  ### 選択確率の計算
49  ## 分母となる, 各々のexp(V)の和をつくる
50  deno <- (car + train + bus + bike + walk)
51  ## それぞれ計算する
52  Ptrain <- Data$代替手段生成可否train*(train / deno)
53  Pbus <- Data$代替手段生成可否bus *(bus / deno)
54  Pcar <- Data$代替手段生成可否car *(car / deno)
55  Pbike <- Data$代替手段生成可否bike *(bike / deno)
56  Pwalk <- Data$代替手段生成可否walk *(walk / deno)
57
58  ## 選択確率が0 になってしまった場合に起こる問題の回避
59  Ptrain <- (Ptrain!=0) * Ptrain + (Ptrain ==0)
60  Pbus <- (Pbus!=0) * Pbus + (Pbus ==0)
61  Pcar <- (Pcar!=0) * Pcar + (Pcar ==0)
62  Pbike <- (Pbike!=0) * Pbike + (Pbike ==0)
63  Pwalk <- (Pwalk!=0) * Pwalk + (Pwalk ==0)
64

```

対数尤度を計算する際に, 選択確立が0だとエラーとなる. ($y = \ln(x), x > 0$)

→選択確立が0だった場合に1を返す処理

16

推定の手順

```
65  ## 選択結果
66  Ctrain <- Data$代表交通手段 == "鉄道"
67  Cbus <- Data$代表交通手段 == "バス"
68  Ccar <- Data$代表交通手段 == "自動車"
69  Cbike <- Data$代表交通手段 == "自転車"
70  Cwalk <- Data$代表交通手段 == "徒歩"
71
72  ## 対数尤度の計算
73  LL <- colSums(Ctrain*log(Ptrain) + Cbus*log(Pbus) +
74              Ccar *log(Pcar) + Cbike *log(Pbike) +Cwalk *log(Pwalk))
```

←“Data”の代表交通手段が“鉄道”
である行には1を, そうでない行には0
を出力し, Ctrainに代入

最尤推定で計算

→観測されたデータが得られる「もっともらしさ」(=尤度)が最大になるようにパラメータを定める

$$L = \prod_{n=1}^N \prod_{j=1}^J P_n(i)^{d_n} \quad \dots(1)$$

d ...個人nが選択肢iを選択したとき1,そうでないとき0

対数尤度の計算

$$\ln L = \sum_{n=1}^N \sum_{i=1}^J d_{in} \ln P_n(i) \quad \dots(2)$$

i ... 選択肢
n ... 個人

17

推定の手順

```
77. ##### 対数尤度関数fr の最大化#####
78 ##パラメータ値の最適化
79 res<-optim(b0,fr, method = "Nelder-Mead", hessian = TRUE, control=list(fnscale=-1))
80
81 ## パラメータ推定値, ヘッセ行列
82 b <- res$par
83 hhh <- res$hessian
84 ## t 値の計算
85 tval <- b/sqrt(-diag(solve(hhh)))
86 ## 初期尤度
87 L0 <- fr(b0)
88 ## 最終尤度
89 LL <- res$value
90
91 ##### 結果の出力 #####
92 print(res)
93 ## 初期尤度
94 print(L0)
95 ## 最終尤度
96 print(LL)
97 ## $p^2$  値
98 print((L0-LL)/L0)
99 ## 修正済 $p^2$  値
100 print((L0-(LL-length(b)))/L0)
101 ##パラメータ推定値
102 print(b)
103 ## t 値
104 print(tval)
```

optim関数: 詳細は行動モデルの資料参照

**推定結果について, t値, 尤度比を計算
→パラメータ推定した後, 得られた結果
が有意かどうかを検証する**

18

推定結果

コンソール画面

resはoptim関数の出力内容.

```
> ##### 結果の出力 #####
```

```
> print(res)
```

```
$par
```

```
[1] 0.53830901 -1.73206039 -1.53193165 -1.32466211 -11.22608054 0.02404709
```

パラメータ値の最適解の値

```
$value
```

```
[1] -1285.627
```

最適化関数の目的値の最適値(ここでは最大尤度)

```
$counts
```

```
function gradient
```

```
501 NA
```

function :最適化の過程での計算の繰り返し回数,
gradient :計算の中で1階偏微分計算が行われた回数

```
$convergence
```

```
[1] 1
```

収束判定

```
$message
```

```
NULL
```

エラーメッセージ

```
$hessian
```

```
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]  
[1,] -173.590913  7.4713557 119.150562  27.773508 -24.6242887 -623.53217  
[2,]  7.471356 -34.3662268  9.510892  7.722710  0.4814521 -52.26618  
[3,] 119.150562  9.5108924 -229.263071  56.706845  26.3706181  537.62951  
[4,]  27.773508  7.7227101  56.706845 -152.738277  6.8832340  79.17670  
[5,] -24.624289  0.4814521  26.370618  6.883234 -8.2720885 -110.87338  
[6,] -623.532174 -52.2661778  537.629510  79.176698 -110.8733781 -3570.91028
```

ヘッセ行列を用いてt値の計算を行っている.

19

推定結果

コンソール画面

```
> ## 初期尤度
> print(L0)
[1] -2135.675
> ## 最終尤度
> print(LL)
[1] -1285.627
> ## $\rho^2$  値
> print((L0-LL)/L0)
[1] 0.3980229
> ## 修正済み $\rho^2$  値
> print((L0-(LL-length(b)))/L0)
[1] 0.3952135
> ##パラメータ推定値
> print(b)
[1] 0.53830901 -1.73206039 -1.53193165 -1.32466211 -11.22608054 0.02404709
> ## t 値
> print(tval)
[1] 3.5953012 -8.7561680 -13.1144197 -12.2064934 -20.4028098 0.7733105
```

正常に回ったら・・・

- 尤度比チェック
- パラメータの符号は仮定に反していないか？
- パラメータは有意に効いているか？t値検定の結果は？

■ t 検定値

検定値は、パラメータ推定値をその標準偏差の推定値で除した値です。サンプル数が十分に多い場合、t 値の絶対値が1.96 以上であれば有意水準5%、2.26 以上であれば有意水準1%で有意といえます。

■ モデルの適合度

修正済み ρ^2 値は、パラメータ数の影響を差し引いたモデルの適合度を表します。

基本的な確認事項

- データセットは正しくできてるか
 - 空欄がないか
 - 誤字はないか(数字のはずが文字まざってるとか)
- ファイルの指定(場所・名前)はあってるか
- パラメータの指定(設定, 宣言)はあってるか
- 効用関数の式にミスはないか
- データをちゃんと読み込めているか

課題

PPデータを使って, 行動モデルを推定してみよう.