

プログラミングの基礎とRの導入

-MNLとその推定-

2018/4/9 (月)

スタートアップゼミ#1

M2 山野 壱成

目次

- はじめに
 - ープログラミングとは
 - ーなぜプログラミングを使うのか
- Rの導入
 - ーインストール方法
 - ー基礎知識
- パラメータ推定
 - ーモデル説明
 - ープログラム構築の流れ



目標

- 「プログラミングという名前にビビらず中身を知ること」
- 「Rを使ってモデル推定の流れを理解すること」



はじめに

プログラミングとは？

- ▶ プログラムを設計すること
- ▶ プログラムとは、コンピューターに処理の内容や順番を正しく指示し、動かすためのものです。
- ▶ コンピューターは機械語しかわからず、人間にはその機械語は難しいので、**両者が理解できるように作られた言葉 (=プログラミング言語)** でプログラムを作成します。
- ▶ プログラミング言語と単に言ってもたくさんの種類があります。
Ex.) R, Java, Python, C言語, HTML, ...
- ▶ これらはそれぞれ強み弱みを持っているので、**必要な処理に合わせて言語を選ぶ**必要があります。(→後ほど紹介)

プログラミング言語



なぜプログラミングを使うの？

A. コンピューターに任せないと出来ない複雑な処理をしたいから

Ex.) 膨大なデータの処理

高度な計算

可視化/シミュレーション

正確な繰り返し計算

羽藤研ではなんとこれらの処理すべてが求められます！

僕らが分析するデータは主に行動データです。

歴史研究では史料の分析をすることもあります、基本これです。

特に行動データは膨大なデータである場合がほとんどなので、

プログラミングをしないと行いたい処理ができません。。

行動データ

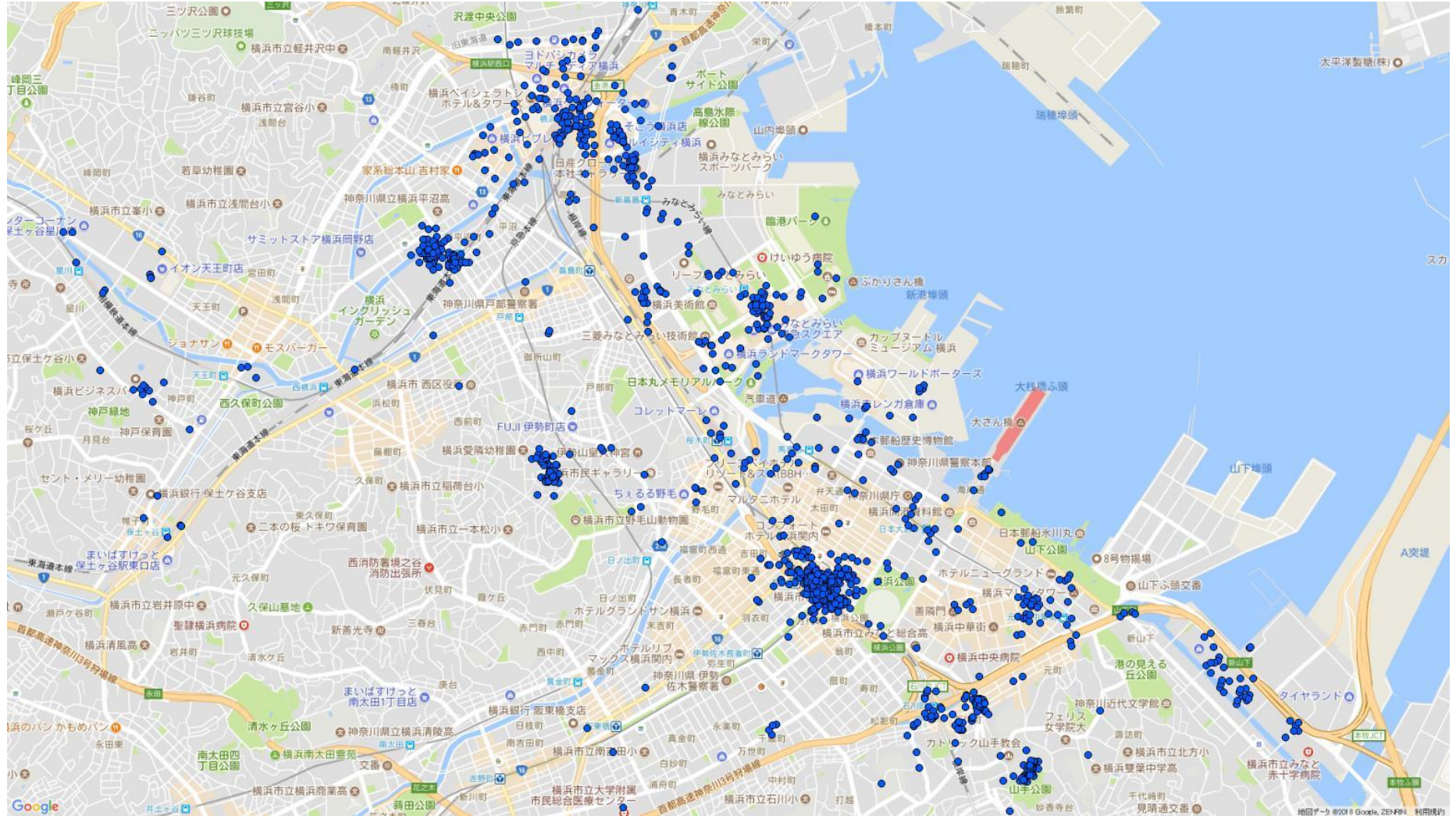
プローブパーソン(PP)データ：

移動体(被験者)に検出器を取り付けることで、移動状態を観測して得られるデータ。長期間・高精度・マイクロレベルのデータを取得でき、個人属性情報も同時に得られるため、様々な分析が可能に。

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V			
トリップID	モーターID	目的コード	目的	出発日時	到着日時	実トリップ時	OD直線距	LatO	LonO	shikucode	shikuname	zone4code	zone4name	zone5code	zone5name	LatD	LonD	shikucode	shikuname	zone4code	zone4name	zone5code	zone5name	
255461	yd021	800	散歩・回遊	#####	#####	5461	19341.21	35.34554	139.4865	205	藤沢市	2050201	本町	2.05E+08	本町一丁目	35.51435	139.5381	113	緑区	1008112	中山			
256674	yd027	500	買い物	#####	#####	3930	20793.14	35.44361	139.6377	104	中区	1003002	関内	1E+08	真砂町	35.62783	139.6774							
255291	yd025	400	業務	#####	#####	2717	10421.32	35.36674	139.5715	115	栄区	1009248	上郷	1.01E+08	上郷町A	35.444	139.6366	104	中区	1003002	関内			
256382	yd021	100	出勤	#####	#####	3749	24457	35.32551	139.4091	207	茅ヶ崎市	2070202	東海岸北	2.07E+08	東海岸北二丁目	35.44406	139.6364	104	中区	1003002	関内			
257459	yd028	700	娯楽	#####	#####	6650	42660.9	35.46426	139.5309	112	旭区	1005118	二俣川	1.01E+08	二俣川一丁目	35.24526	139.1446	206	小田原市	2060401	桜橋			
257040	yd025	600	食事	#####	#####	2570	740.7383	35.44344	139.637	104	中区	1003002	関内	1E+08	港町	35.44966	139.6341	104	中区	1003002	関内			
256978	yd018	200	帰宅	#####	#####	3691	29264.96	35.42393	139.3185	212	厚木市	2120802	岡津古久	2.12E+08	日産テクニ	35.4873	139.6322	102	神奈川区	1001015	白幡			
256452	yd027	999	その他	#####	#####	476	898.8058	35.63107	139.6799							35.62319	139.6822							
256965	yd024	200	帰宅	#####	#####	3482	19802.6	35.44457	139.6368	104	中区	1003002	関内	1E+08	真砂町	35.6163	139.5787	135	多摩区	1300502	宿河原			
256459	yd012	200	帰宅	#####	#####	3731	27332.68	35.4225	139.3166	212	厚木市	2120802	岡津古久	2.12E+08	日産テクニ	35.45974	139.6149	103	西区	1002008	岡野			
256769	yd002	200	帰宅	#####	#####	2338	20828.19	35.63568	139.6282							35.44844	139.6206	103	西区	1002004	西戸部			
256843	yd027	999	その他	#####	#####	392	924.5841	35.63111	139.68							35.62293	139.6818							
256812	yd022	600	食事	#####	#####	1827	2170.15	35.45211	139.6405	104	中区	1003002	関内	1E+08	新港一丁目	35.46591	139.6235	103	西区	1002001	高島			
256061	yd008	200	帰宅	#####	#####	999	1338.583	35.43014	139.6653	104	中区	1003010	本牧	1E+08	小港町	35.42545	139.6517	104	中区	1003012	西之谷			
256231	yd006	500	買い物	#####	#####	2274	3028.804	35.4524	139.5813	106	保土ヶ谷区	1005010	桜ヶ丘	1.01E+08	花見台	35.42985	139.5626	110	戸塚区	1009032	東戸塚			
256524	yd004	200	帰宅	#####	#####	8344	18830.48	35.42418	139.3186	212	厚木市	2120802	岡津古久	2.12E+08	日産テクニ	35.44254	139.5253	116	泉区	1009326	岡津北			
255843	yd023	100	出勤	#####	#####	3009	10939.6	35.45983	139.5173	112	旭区	1005103	中希望ヶ丘	1.01E+08	中希望ヶ丘	35.44449	139.6366	104	中区	1003002	関内			
256632	yd022	100	出勤	#####	#####	1921	10110.63	35.40567	139.5352	110	戸塚区	1009029	矢部	1.01E+08	矢部町A	35.4448	139.6359	104	中区	1003002	関内			
255777	yd005	700	娯楽	#####	#####	4196	14499.68	36.03217	139.5338							35.92111	139.6183							
256559	yd011	100	出勤	#####	#####	979	4273.904	35.4352	139.3605	212	厚木市	2120102	駅南	2.12E+08	旭町5	35.44417	139.3146	212	厚木市	2120601	森の里東			
255985	yd018	200	帰宅	#####	#####	4377	29602.16	35.42447	139.3187	212	厚木市	2120802	岡津古久	2.12E+08	日産テクニ	35.48868	139.6359	102	神奈川区	1001014	浦島丘			
256514	yd010	200	帰宅	#####	#####	7010	37144.88	35.42602	139.3203	212	厚木市	2120802	岡津古久	2.12E+08	日産テクニ	35.25568	139.6726	201	横須賀市	2010407	公郷			
256996	yd030	200	帰宅	#####	#####	9229	14183.93	35.44152	139.3143	212	厚木市	2120601	森の里東	2.12E+08	若宮A	35.53517	139.4207	209	相模原市	2090407	若松			
258021	yd008	999	その他	#####	#####	982	1817.245	35.42953	139.662	104	中区	1003010	本牧	1E+08	本牧町A	35.4388	139.6455	104	中区	1003017	山手			
257573	yd018	500	買い物	#####	#####	687	1721.696	35.50499	139.6257	109	港北区	1008040	富士塚	1.01E+08	富士塚二丁目	35.50059	139.644	102	神奈川区	1001001	西寺尾			
256738	yd023	800	散歩・回遊	#####	#####	4200	125.4635	35.45938	139.5169	112	旭区	1005103	中希望ヶ丘	1.01E+08	中希望ヶ丘	35.46048	139.5172	112	旭区	1005121	東希望ヶ丘			
256373	yd018	200	帰宅	#####	#####	3654	29638.74	35.42408	139.3184	212	厚木市	2120802	岡津古久	2.12E+08	日産テクニ	35.4886	139.636	102	神奈川区	1001014	浦島丘			
257313	yd002	100	出勤	#####	#####	3352	27612.94	35.44912	139.6214	103	西区	1002004	西戸部	1E+08	西戸部町	35.42463	139.3181	212	厚木市	2120802	岡津古久			
257169	yd017	200	帰宅	#####	#####	1119	3968.274	35.46508	139.6237	103	西区	1002001	高島	1E+08	高島二丁目	35.43554	139.6483	104	中区	1003017	山手			
255492	yd029	200	帰宅	#####	#####	1182	759.3317	35.42335	139.6646	104	中区	1003011	新本牧	1E+08	本牧原B	35.43003	139.6629	104	中区	1003010	本牧			
257194	yd017	999	その他	#####	#####	1298	4524.924	35.43564	139.6485	104	中区	1003017	山手	1E+08	山手町	35.46935	139.6205	102	神奈川区	1001021	鶴屋			
256199	yd027	500	買い物	#####	#####	995	1906.888	35.52152	139.7325	131	川崎区	1300102	塩浜	1.3E+08	池上新町3丁目	35.53335	139.7172	131	川崎区	1300109	港町			
257189	yd027	200	帰宅	#####	#####	628	506.1193	35.62883	139.685							35.63085	139.68							
255413	yd028	500	買い物	#####	#####	1987	1795.658	35.45005	139.6326	104	中区	1003001	野毛	1E+08	桜木町A	35.46493	139.6249	103	西区	1002001	高島			
257761	yd002	999	その他	#####	#####	424	1354.56	35.44965	139.6214	103	西区	1002004	西戸部	1E+08	西戸部町	35.45726	139.6331	103	西区	1002002	みなとみらい			
256112	yd025	500	買い物	#####	#####	1003	1783.713	35.36675	139.5713	115	栄区	1009248	上郷	1.01E+08	上郷町A	35.36841	139.5517	115	栄区	1009242	小菅ヶ谷			
257964	yd024	100	出勤	#####	#####	5813	19625.3	35.61496	139.5805	135	多摩区	1300502	宿河原	1.3E+08	宿河原4丁目	35.44432	139.636	104	中区	1003002	関内			
256672	yd011	200	帰宅	#####	#####	1125	4289.298	35.44415	139.3146	212	厚木市	2120601	森の里東	2.12E+08	青山1丁目	35.43643	139.361	212	厚木市	2120101	駅北			
255814	yd006	200	帰宅	#####	#####	2334	445.7958	35.43095	139.5556	110	戸塚区	1009032	東戸塚	1.01E+08	川上町C	35.42694	139.5554	110	戸塚区	1009034	柏尾			
255986	yd011	200	帰宅	#####	#####	1188	4303.119	35.44439	139.3146	212	厚木市	2120601	森の里東	2.12E+08	青山1丁目	35.43582	139.3609	212	厚木市	2120102	駅南			
257300	yd002	200	帰宅	#####	#####	2420	24363.91	35.66589	139.6608							35.44909	139.6215	103	西区	1002004	西戸部			
257197	yd005	200	帰宅	#####	#####	1857	7690.463	35.46485	139.623	103	西区	1002001	高島	1E+08	高島二丁目	35.42485	139.5537	110	戸塚区	1009034	柏尾			

データを取りあえず可視化してみる…

※ちなみにこの可視化はGISを用いています



→よくわからない

どのプログラミング言語/ツールを使う？

□ この人がどういう経路を通ってるのか見てみたい！

Javaで位置情報(ドット)を繋げて経路に (→#3 Javaの回で詳しく)

GISで経路をマップ上に可視化 (→#3 GISの回で詳しく)

□ どういう道路を好んで歩いてるか知りたい！

Rで離散連続モデルの推定

今日の内容

□ この分析やりたいけどちょっと複雑だし計算量もすごそうだなあ。

Pythonの豊富なオープンソース^{*}を使って複雑な計算(最適化など)も余裕

□ 論文ってどうやったら綺麗に書けるの？

Texで書いて処理すればかっこいい論文が完成

* 中身の処理が見えないので
エラーが出た時大変
(ブラックボックス)



Rの導入

Rとは

- 統計処理を目的とするプログラミング言語
- 無償＋オープンソースのソフトウェア
→誰もが便利なパッケージを無償で使える！
- R関連のおすすめWebサイト

R-Tips

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

Rの基本操作はここ！参考書的に使えて便利。

RjpWiki

(<http://www.okada.jp.org/RWiki/?RjpWiki>)

Rに関する情報を日本語でまとめたあるwiki。誰でも編集可。

特徴

- ベクトル・行列演算のプログラムを書きやすい
- 計算も比較的速くできる
- データ可視化も簡単にできる
- プログラミング初心者でも簡単に使える

■ R用の統合開発環境（エディタ＋実行＋表示機能；IDE）

The image shows the RStudio IDE interface. The top-left pane is the source editor with R code for a logit model. The top-right pane shows the Environment window with a list of variables. The bottom-left pane is the Console window showing the execution of the code. The bottom-right pane shows a plot of the data, including a histogram and a Q-Q plot.

エディタ

```
1 ### Logit model estimation
2
3 ### データファイルの読み込み
4 Data <- read.csv("C:/演習課題/サンプル/sample.csv",header=T)
5
6 ## データ数:Dataの行数を数える
7 hh <- nrow(Data)
8
9
10 ## 今回用いる選択肢の数
11 ch <- 5
12
13 ## パラメータの初期値の設定
14 b0<-c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
15
16
17 ### Logit model の対数尤度関数の定義
18
19 fr <- function(x) {
20
```

データ

Variable	Value
vc	-1.00201023921001
vhome	2.29784433609433
vlist	Large list (397768 elements, 105.3 Mb)
vnum	397768
vs	7384
vself	2.14626309356719
vvec	num [1:8] 0 1 1 0 0 0 0 0
w	4L
x	num [1:100] 1.6 3.33 3.07 2.57 6.78 ...

コンソール

```
hist> hist(x, freq = FALSE, ylim = c(0, 0.2))
hist> curve(dchisq(x, df = 4), col = 2, lty = 2, lwd = 2, add = TRUE)
hist> ## Don't show:
hist> par(op)
hist> ## End(Don't show)
hist>
hist>
hist>
Warning message:
In plot.histogram(r, freq = freq1, col = col, border = border, angle = angle, :
the AREAS in the plot are wrong -- rather use 'freq = FALSE'
```

プロット

Histogram of x

※基本的にエディタはプログラムを書く場所で、
それ以外の部分は実行結果というイメージ

Rのインストール

■ Rのインストール (windows)

<http://cse.naro.affrc.go.jp/takezawa/r-tips/r/01.html>

<https://qiita.com/FukuharaYohei/items/8e0ddd0af11132031355>

※他のOSのインストール方法もRjpWikiやR-tipsで調べられます

■ R studioのインストール

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

<https://qiita.com/FukuharaYohei/items/3468bd2a6b2f07b8963e>

Rを使ってみる

R Studioを起動 → 新規ファイル作成 → スクリプトを書いて実行！

R Scriptをクリック

The screenshot shows the RStudio interface with the 'File' menu open. The 'R Script' option is highlighted, and a blue box highlights the 'R Script' option in the 'File' menu. The 'Console' window shows the output of a script, including a message and a matrix of values.

```
32 2 * x[1] + 2 * x[2] + 2 * x[3] + x[4] + x[5] + 2 * x[2]
33  }
34  constroptim(c(2,2), fQP, NULL, ui=c(2,1), ci=5)
35
36 #配列の入力
33:2 (Top Level)
```

Environment History

Object	Class	Attributes
Global Environment	Environment	
D2	matrix	4 obs. of 2 variables
D4	matrix	5 obs. of 3 variables
Data	matrix	287 obs. of 23 variables
hhh	matrix	num [1:11, 1:11] -57.8 15.7 19.1 13 -105.4 ...
submatrix	matrix	1 obs. of 2 variables
x	matrix	int [1:14] 1 1 1 1 1 1 1 1 1 1 1 1 1 1

File Edit Code View Plots Session Build Debug Profile Tools Help

- New File
 - R Script** Ctrl+Shift+N
 - R Notebook
 - R Markdown...
 - Shiny Web App...
 - Text File
 - C++ File
 - R Sweave
 - R HTML
 - R Presentation
 - R Documentation
- New Project...
- Open File... Ctrl+O
- Reopen with Encoding...
- Recent Files
- Open Project...
- Open Project in New Session...
- Recent Projects
- Import Dataset
- Save Ctrl+S
- Save As...
- Save with Encoding...
- Save All Ctrl+Alt+S
- Knit Document Ctrl+Shift+K
- Compile Report...
- Print...
- Close Ctrl+W
- Close All Ctrl+Shift+W
- Close All Except Current Ctrl+Alt+Shift+W
- Close Project
- Quit Session... Ctrl+Q

Console

```
$message
NULL

$hessian
      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,] -57.756449 15.668185 19.128462 12.994144 -105.43810 15.400761 28.063424
[2,] 15.668185 -29.306707 5.577379 4.623000 16.85812 -34.162049 9.192850
[3,] 19.128462 5.577379 -31.381205 3.718730 31.49855 6.471114 -52.432860
[4,] 12.994144 4.623000 3.718730 -30.315144 32.75231 7.391842 8.604330
[5,] -105.438097 16.858118 31.498549 32.752310 -374.52130 29.372664 65.223792
[6,] 15.400761 -34.162049 6.471114 7.391842 29.37266 -70.259490 12.793144
[7,] 28.063424 9.192850 -52.432860 8.604330 65.22379 12.793144 -142.452037
[8,] 30.503258 9.129793 10.050229 -68.598618 113.45324 15.782912 24.442999
[9,] 24.763788 7.180775 8.401955 19.432528 95.52929 11.048345 20.202293
[10,] 9.008229 3.186487 2.729763 7.738181 21.70399 4.325596 5.961028
[11,] 13.371202 -24.663538 4.159586 4.083222 14.09876 -28.558237 7.613557

> print(tva1)
[1] 2.5405527 0.1028183 2.0132793 1.7558485 -7.8802568 -6.1700941 -6.5794795 -6.00634
>
```

Version

Package	Version
Angelo Canty for S)	1.3-20
its on the Checkpoint Server for	0.4.3
	7.3-14
iter Analysis Extended Rousseeuw et al.	2.0.6
	0.2-15
	3.4.3
ien for R	3.1
	3.4.3
	9.0.0
ie "parallel" Package	1.0.12
struct for R	1.4.5
"SAS" "SPSS" "Stata" "Svutab" "Mkex"	0.8-69

Rを使ってみる

R Studioを起動 → 新規ファイル作成 → スクリプトを書いて実行！
実行したいスクリプトを選択して [Ctrl] + [Enter]

The screenshot displays the RStudio interface. The main editor window contains the following R script:

```
1 x <- (20 %% 8) / 5
2 y <- 2^3 + 2*3
3 x
4 y
5 (x > 0) && (y > 0)
6 (x >= 1) && (y >= 1)
7 1:10
```

The text "実行したいスクリプト" (Script to be executed) is overlaid on the script. The console window at the bottom shows the output of the script:

```
>>>
> x <- (20 %% 8) / 5
> y <- 2^3 + 2*3
> x
[1] 0.4
> y
[1] 14
>>>
```

The text "結果" (Result) is overlaid on the console output. The Environment pane on the right shows the global environment with variables like 'Data', 'hh', 'b', 'b0', 'ch', 'hh', 'LO', 'LL', 'res', 'tval', 'x', 'y', and 'fr'.

基本的な知識

基本概念

- 演算子
- 変数とオブジェクト
- 関数
- 繰り返し文

→ここを見ながら自分で書いて理解しましょう！

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

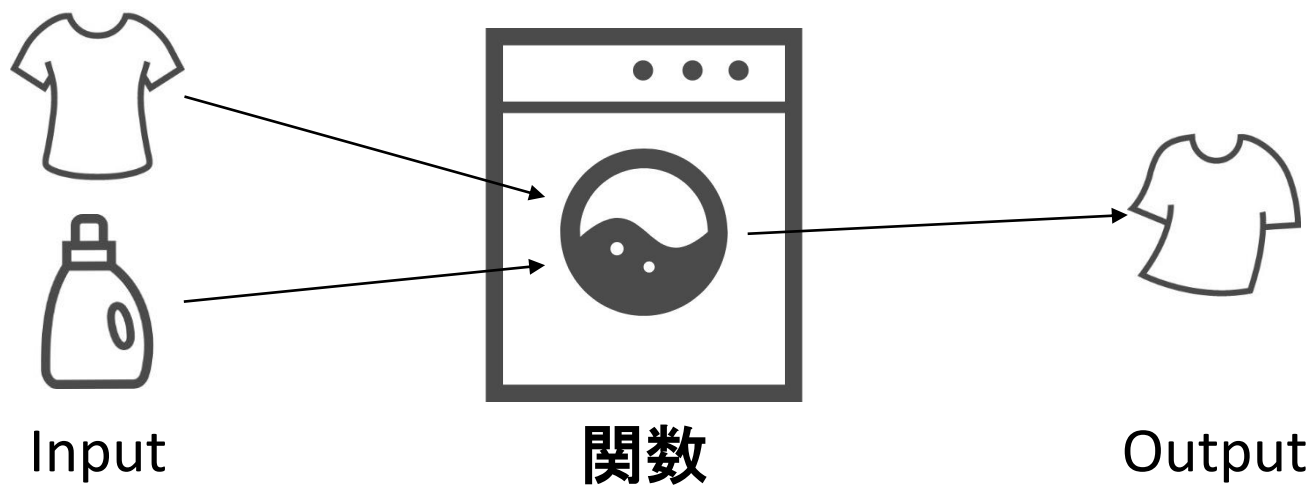
※上記の基本概念は**全プログラミングを通して重要**な概念です。

逆に言えばこれさえ理解すれば言語問わずプログラムが書けます！

注意

- ・ 全角と半角、大文字と小文字は区別されます
- ・ 誤字はもちろん、意味のない空白もNGです

関数の定義と使い方



定義

```
(関数名) <- function (引数) {  
    処理  
    return (戻り値)  
}
```

例：絶対値を求める

```
10 ▾ abs <- function(a,b){  
11 ▾   if(a>=b){  
12     return(a-b)  
13 ▾   }else{  
14     return(b-a)  
15   }  
16 }  
17
```

←aとbを入れたら
←そのa, bに
この処理をします

```
>  
> x <- abs(3,6)  
> print(x)  
[1] 3  
> |
```

←引数に代入
←戻り値GET



行動モデルと推定

行動モデルの基本

行動モデルの理論

合理的な個人は「**効用**」が最大になる選択肢を選択すると仮定することで、それぞれの選択肢の効用の大小によって選択が行われるという定式ができます。

効用 U_{nk} ：個人 n が選択肢 k に感じている魅力の度合い
(=「嬉しさの程度」)

パラメータ推定

ある母集団からランダムにサンプリングされたデータを用いて、母集団の特性値である未知パラメータを求めること

つまり、行動モデルでは、
各選択肢が実際に選択されたとき、それぞれの説明変数がどれくらい選択に影響を与えているか(パラメータの値)を調べることです。

パラメータ推定とは

推定の前に…

- ・ 政策変数(何を評価したいのか)を決める
ex.歩道拡幅したら歩行者行動は？
- ・ 仮説(何を確かめたいのか)を立てる
ex.歩道幅員の広い経路が選ばれやすい？

→そのために基礎分析が必要！

基礎分析を通してデータの分布や特性を把握すること

[基礎分析のやり方例]

- ・ Excelでクロス集計
- ・ Javaで集計プログラムを組む
- ・ Rでヒストグラムを書いてみる など

離散選択モデル

効用関数の設定

確定項 誤差項

$$U_{in} = V_{in} + \varepsilon_{in}$$

$$V_{in} = \beta_1 x_{1in} + \beta_2 x_{2in} + \cdots + \beta_J x_{Jin}$$

$\boldsymbol{\beta} = \{\beta_1, \beta_2, \dots, \beta_J\}$: 未知パラメータベクトル

$\mathbf{x}_{in} = \{x_{1in}, x_{2in}, \dots, x_{Jin}\}$: 個人 n の選択肢 i における説明変数ベクトル

個人 n が選択肢 i を選ぶ選択確率

$$P_{in} = \frac{\exp(\mu V_{in})}{\sum_{j \in I_n} \exp(\mu V_{jn})}$$

μ : 誤差項のばらつきを表すスケールパラメータ

I_n : 個人 n の選択肢集合

ロジットモデルでは
誤差項のガンベル分布を仮定

離散選択モデル

尤度関数

選択確率最大であるときに、実際の選択行動が観測されるという事象が生じることが尤もらしいです。

つまり、尤度関数の最大化問題を解けばよいわけです。

対数尤度関数

計算しやすいように対数をとります。

$$\ln L(\theta) = \sum_{n=1}^N \sum_{i \in I_n} \delta_{in} \ln(P_{in})$$
$$P_{in} = \frac{\exp(V_{in})}{\sum_{j \in I_n} \exp(V_{jn})}$$

δ_{in} : 個人 n が選択肢 i を実際に選んだ場合1, そうでなければ0を取る変数

I_n : 個人 n の選択肢集合

V_{in} : 個人 n にとっての選択肢 i の効用の確定項

尤度関数の定義までの流れ

1. パラメータ宣言
2. 効用確定項の計算
3. 選択確率の計算
4. 選択結果の判別
5. 対数尤度の計算

では実際に
MNLの推定コードを
確認してみましょう
(別紙参照)

データの参照方法

3 ### データファイルの読み込み

```
4 Data <- read.csv("C:/Users/S-YAMAMOTO2016/Desktop/Data_Clean_English.csv", header=TRUE)
```

読み込むファイルの場所

列名付き=T
列名なし=F

Dataの"PurposeCode"の列データが欲しい！

方法① : Data\$PurposeCode

← 「データ \$ 列名」

方法② : Data[, 2]

← 「データ[行番号, 列番号]」

※空欄は指定しない(=全て)ということ

	TripID	MonitorID	PurposeCode	PurposeJP	PurposeENG	DepartureTime	ArrivalTime	TripDurationSec	ODDistanceM	LatitudeO	LongitudeO	DistrictCodeO	DistrictNameO	Zone
1	255461	yd021	800	散歩・回遊	Leisure Walk	40117.26	40117.33	5461	19341.2131	35.34554	139.4865	205	藤沢市	
2	256674	yd027	500	買い物	Shopping	40130.80	40130.84	3930	20793.1352	35.44361	139.6377	104	中区	
3	255291	yd025	400	業務	Work	40115.32	40115.35	2717	10421.3243	35.36674	139.5715	115	栄区	
4	256382	yd021	100	出勤	Commute	40127.29	40127.33	3749	24456.9960	35.32551	139.4091	207	茅ヶ崎市	
5	257459	yd028	700	娯楽	Leisure	40138.45	40138.52	6650	42660.9018	35.46426	139.5309	112	旭区	
6	257040	yd025	600	食事	Meal	40134.51	40134.54	2570	740.7383	35.44344	139.6370	104	中区	
7	256978	yd018	200	帰宅	Return Home	40133.86	40133.91	3691	29264.9558	35.42393	139.3185	212	厚木市	
8	256452	yd027	999	その他	Other	40127.82	40127.82	476	898.8058	35.63107	139.6799	NA		
9	256965	yd024	200	帰宅	Return Home	40133.78	40133.82	3482	19802.6032	35.44457	139.6368	104	中区	
10	256459	yd012	200	帰宅	Return Home	40127.92	40127.96	3731	27332.6810	35.42250	139.3166	212	厚木市	
11	256769	yd002	200	帰宅	Return Home	40131.81	40131.84	2338	20828.1911	35.63568	139.6282	NA		
12	256843	yd027	999	その他	Other	40132.67	40132.68	392	924.5841	35.63111	139.6800	NA		
13	256812	yd022	600	食事	Meal	40132.52	40132.54	1827	2170.1501	35.45211	139.6405	104	中区	
14	256061	yd008	200	帰宅	Return Home	40123.61	40123.63	999	1338.5827	35.43014	139.6653	104	中区	
15	256231	yd006	500	買い物	Shopping	40125.67	40125.70	2274	3028.8040	35.45240	139.5813	106	保土ヶ谷区	
16	256524	yd004	200	帰宅	Return Home	40128.85	40128.95	8344	18830.4837	35.42418	139.3186	212	厚木市	
17	255843	yd023	100	出勤	Commute	40121.31	40121.34	3009	10939.5981	35.45983	139.5173	112	旭区	

Showing 1 to 19 of 1,522 entries

最適化関数

最適化関数optim() :

```
optim(par, fn, method, control = list(), hessian = FALSE)
```

par : パラメータの初期値

fn : 最適化する目的関数

method : 最適化手法 (Nelder-Mead, BFGS, CG, L-BFGS-B, SANN)

↑デフォルト

control : optim関数は最小化を行う関数なので、
最大化の場合はcontrol=list(fnscale=-1) と指定

hessian : ヘッセ行列を計算する場合にTRUEを指定

パラメータ推定ではt値の計算に必要なので基本TRUE
↑得られたパラメータ値を評価

最適化関数

resには何が入ってる？

print(res)してみた↓

```
> ##### 結果の出力 #####
> print(res)
$par
[1] 0.6094431 -1.7053668 -1.5367356 -1.3186491 -11.1276633

$value
[1] -1283.725

$counts
function gradient
      396          NA

$convergence
[1] 0

$message
NULL

$hessian
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] -174.135203  7.4149731 119.001294  28.149800 -24.5266351
[2,]  7.414973 -33.6138868  9.242299  7.542625  0.4983249
[3,] 119.001294  9.2422988 -228.143895  56.309560  26.1743460
[4,]  28.149800  7.5426252  56.309560 -152.676188  6.9525599
[5,] -24.526635  0.4983249  26.174346  6.952560  -8.2540995

< ## 初期値
```

←推定パラメータ値

←目的関数の最大値

←収束判定：0ならば収束
(他の値なら収束していない)

←ヘッセ行列

これらを使ってt値などの統計値の計算を行います

推定結果の書き方

説明変数	パラメータ	t値
所要時間[分/10]	-0.762	-7.41 **
費用[円/100]	-0.044	-0.98
乗車外時間[分/10]	-1.049	-8.07 **
女性ダミー	1.811	5.53 **
定数項 (鉄道)	3.168	7.46 **
定数項 (自動車)	0.868	2.08 *
定数項 (自転車)	0.046	0.13
定数項 (徒歩)	2.037	4.81 **
サンプル数	400	
初期尤度	-564.18	
最終尤度	-344.08	
尤度比	0.390	
修正済み尤度比	0.376	

*5%有意 **1%有意

注意点

- ・ 単位を書く
- ・ 桁数を揃える
- ・ 有意なパラメータに*印をつける

上手く回らないときの確認事項

- ✓ データセットは正しくできているか
 - ・ 空欄はない？
 - ・ 誤字はない？(数字と文字が混在してない？)
- ✓ パラメータの設定はあっているか(個数・宣言)
- ✓ ファイルの指定場所は正しいか
- ✓ 効用関数の式は正しいか
 - ・ 括弧の閉じ忘れ
 - ・ 説明変数の数
 - ・ 列名の指定

まとめ

- Rにはたくさんの便利な関数とパッケージがあります
 - 調べればたいていのことはできる気がする
- 中でどんな処理をしてるかに関心を払おう
- データセットを作るのは大変
- 推定は実際にやってみないとわからないと思います。

ということ

目的地選択モデルの推定

サンプルデータをWikiからダウンロードして
実際にコードを書いて実行してみてください。
交通手段選択のMNLのコードをもとにやればできるはず！

もし分からないことがあれば気軽に聞いて下さい
→yamano@bin.t.u-tokyo.ac.jp