第2回B4スタートアップゼミ
～Rでモデル推定～

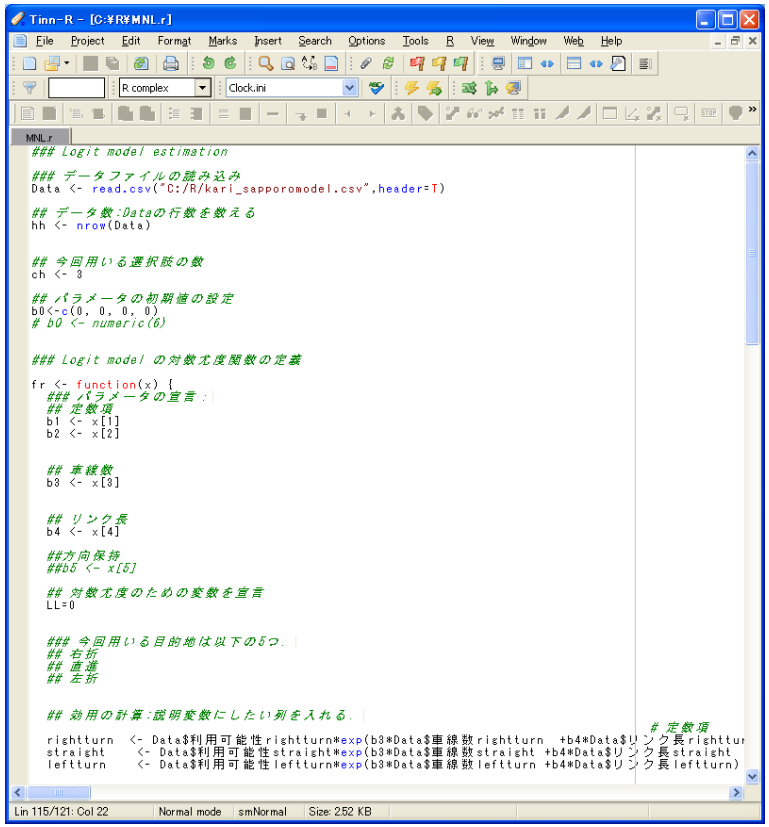
羽藤研　修士2年
植村 恵里

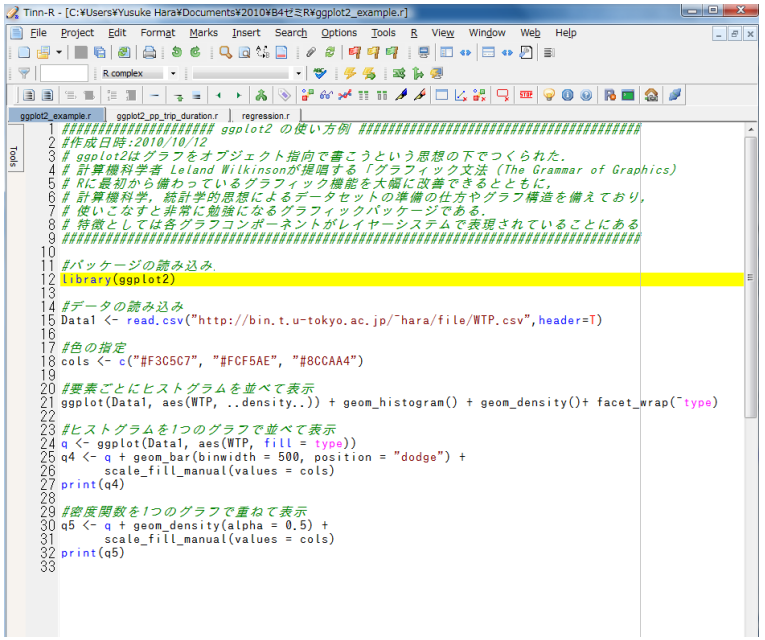
今日やること

- Rの紹介
- 推定手順の紹介
- グラフも作れます

1. Rとは

- S言語をオープンソース化したインタープリタ型言語（比較：コンパイラ）
- 基本的に統計処理に特化した言語と考えて良い
- モデル推定やグラフ作成，計算などに使用





1. 基礎的な紹介～ベクトル処理～

“代入”を表す

```
> a <- 1
> a
[1] 1
> a <- c(1,2,3,4,5)
> a
[1] 1 2 3 4 5
> a <- matrix(c(1,2,3,4,5,6), nrow=2, ncol=3)
> a
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

←変数を代入したら変数を返す

←ベクトルを代入したらベクトルを返す

←行列を代入したら行列を返す

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

←普通の数式を書くように，
各ベクトルごとの計算ができる

2. 推定手順の紹介

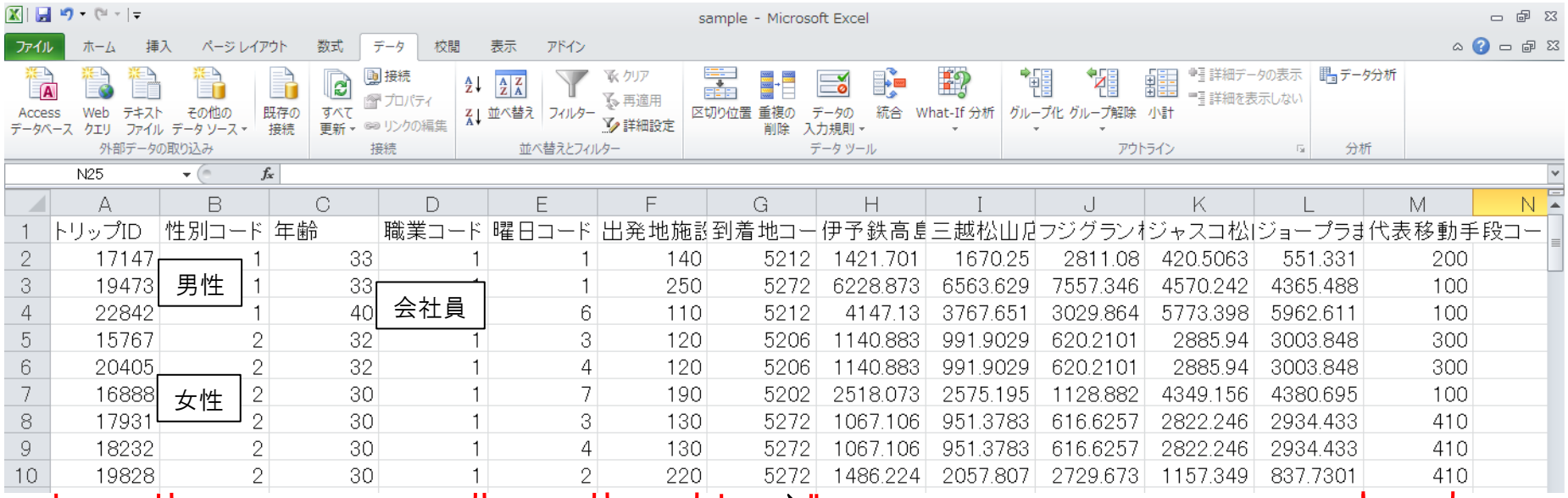
まずは準備

- 今回は前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，まわしてみます．
- 今回の例で推定しているのは
  - どの商業地を選択するかを決める意思決定モデル
  - 選択肢集合は5肢(伊予鉄高島屋，三越松山店，フジグラン松山店，ジャスコ松山，ジョー・プラ)
  - 説明変数は，出発地からの各店舗までの距離，男性ダミー，40代以上ダミーを用いる

- Sample.csvと，mnl.rを開いてください．

2.1. 推定のファイルを確認します

- 推定をするには，まず，推定の為のcsvファイルを作る必要があります．
- 今回使用するのがこちら．



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	トリップID	性別コード	年齢	職業コード	曜日コード	出発地施設	到着地コー	伊予鉄高島	三越松山店	フジグラン松	ジャスコ松山	ジョープラま	代表移動手段コー	
2	17147	1	33	1	1	140	5212	1421.701	1670.25	2811.08	420.5063	551.331	200	
3	19473	1	33		1	250	5272	6228.873	6563.629	7557.346	4570.242	4365.488	100	
4	22842	1	40		6	110	5212	4147.13	3767.651	3029.864	5773.398	5962.611	100	
5	15767	2	32	1	3	120	5206	1140.883	991.9029	620.2101	2885.94	3003.848	300	
6	20405	2	32	1	4	120	5206	1140.883	991.9029	620.2101	2885.94	3003.848	300	
7	16888	2	30	1	7	190	5202	2518.073	2575.195	1128.882	4349.156	4380.695	100	
8	17931	2	30	1	3	130	5272	1067.106	951.3783	616.6257	2822.246	2934.433	410	
9	18232	2	30	1	4	130	5272	1067.106	951.3783	616.6257	2822.246	2934.433	410	
10	19828	2	30	1	2	220	5272	1486.224	2057.807	2729.673	1157.349	837.7301	410	

男性

女性

会社員

トリップ番号

個人属性

曜日

出発地の施設属性

到着地の選択結果

5212:伊予鉄高島屋
5213:三越松山
5202:フジグラン
5272:ジャスコ
5303:ジョープラ

各選択肢の出発地からの距離

移動手段

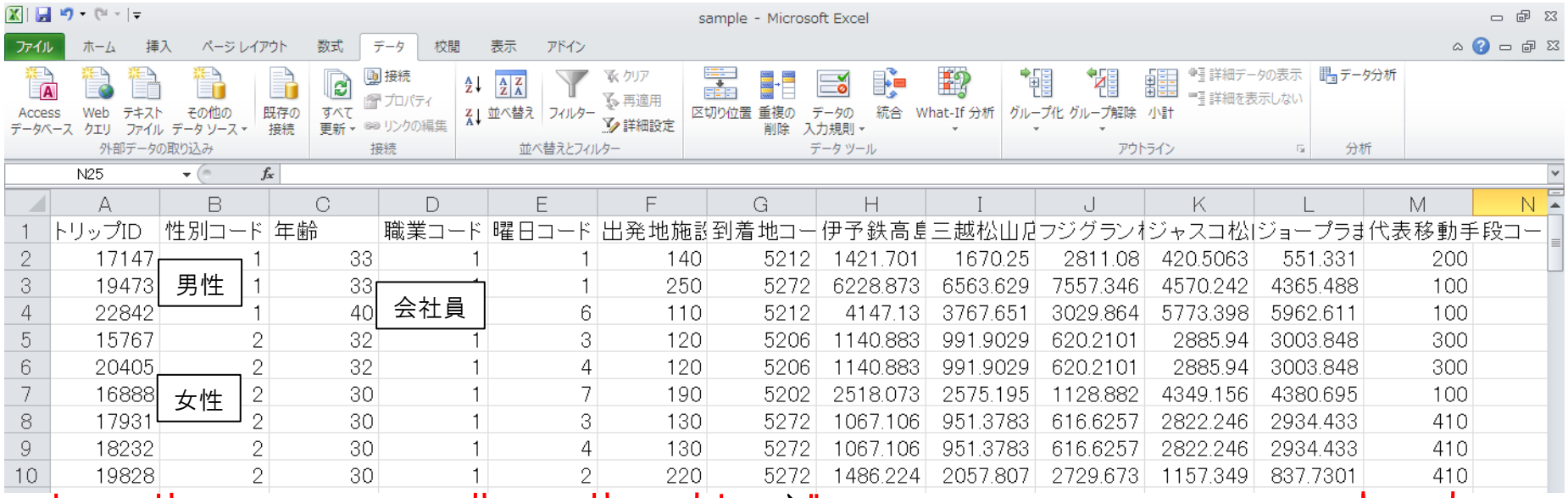
2.1. 推定のファイルを確認します

個人属性，曜日，出発地の施設属性，出発地と到着地の距離，移動手段

→説明変数として利用可能

Ex. 男性はジャスコによく行く / 月曜はセールをしているから伊予鉄に行く / 仕事帰りには三越に行く人が多い / 近いところに行く / 車利用者はフジグランによく行く　などなど

クロス集計からこういった傾向を分析し，説明変数に入れていく．



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	トリップID	性別コード	年齢	職業コード	曜日コード	出発地施設	到着地コー	伊予鉄高島	三越松山店	フジグラン松	ジャスコ松山	ジョープラ松	代表移動手段コー	
2	17147	1	33	1	1	140	5212	1421.701	1670.25	2811.08	420.5063	551.331	200	
3	19473	1	33		1	250	5272	6228.873	6563.629	7557.346	4570.242	4365.488	100	
4	22842	1	40		6	110	5212	4147.13	3767.651	3029.864	5773.398	5962.611	100	
5	15767	2	32	1	3	120	5206	1140.883	991.9029	620.2101	2885.94	3003.848	300	
6	20405	2	32	1	4	120	5206	1140.883	991.9029	620.2101	2885.94	3003.848	300	
7	16888	2	30	1	7	190	5202	2518.073	2575.195	1128.882	4349.156	4380.695	100	
8	17931	2	30	1	3	130	5272	1067.106	951.3783	616.6257	2822.246	2934.433	410	
9	18232	2	30	1	4	130	5272	1067.106	951.3783	616.6257	2822.246	2934.433	410	
10	19828	2	30	1	2	220	5272	1486.224	2057.807	2729.673	1157.349	837.7301	410	

男性

女性

会社員

トリップ番号

個人属性

曜日

出発地の施設属性

到着地の選択結果

5212:伊予鉄高島屋
5213:三越松山
5202:フジグラン
5272:ジャスコ
5303:ジョープラ

各選択肢の出発地からの距離

移動手段

2.2. 推定の流れを追っていきます

```
### Logit model estimation

### データファイルの読み込み
Data <- read.csv("C:/R/sample.csv",header=T)

## データ数:Dataの行数を数える
hh <- nrow(Data)


## 今回用いる選択肢の数
ch <- 5

## パラメータの初期値の設定
b0<-c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
```

←#:コメント入力(読み込まれない)

←データファイルの読み込み
Read.csv("c/…)で，ファイルの場所を指定しています

←行数を数える
(データがちゃんと読み込めていないと，0とか違う行数が出てくる．)

←パラメータの初期値を設定

```
### Logit model の対数尤度関数の定義

fr <- function(x) {
  ### パラメータの宣言：
  ## 定数項
  b1 <- x[1]
  b2 <- x[2]
  b3 <- x[3]
  b4 <- x[4]
  ## 買い物地までの距離
  d1 <- x[5]
  d2 <- x[6]
  d3 <- x[7]
  d4 <- x[8]
  d5 <- x[9]
  ## 個人属性
  w5 <- x[10] ## 女性ダミー
  a2 <- x[11] ## 年齢ダミー
```

パラメータの宣言
今回は11個設定．↑の初期値設定も同数に対応．

2.2. 推定の流れを追っていきます

```
### 今回用いる目的地は以下の5つ．()内は各施設の目的地コード
## takashi  (= 5212)
## jusco    (= 5272)
## mitsuko  (= 5213)
## matsu    (= 5202)
## joepla   (= 5303)

## 効用の計算：説明変数にしたい列を入れる
                  # 距離                                  # 個人属性ダミー          # 定数項
takashi <- exp(d1*Data$伊予鉄高島屋までの距離/1000                                  + b1*matrix(1,nrow=hh,ncol=1))
mitsuko <- exp(d2*Data$三越松山店までの距離/1000 + a2*(Data$年齢>=40)               + b2*matrix(1,nrow=hh,ncol=1))
matsu   <- exp(d3*Data$フジグラン松山店までの距離/1000                              + b3*matrix(1,nrow=hh,ncol=1))
jusco   <- exp(d4*Data$ジャスコ松山店までの距離/1000                                + b4*matrix(1,nrow=hh,ncol=1))
joepla  <- exp(d5*Data$ジョーブラまでの距離/1000 + w5*(Data$性別==2))
```

確定項

$U = V + \varepsilon$

復習：効用関数は“確定項＋誤差項”で表される

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	トリップID	性別コード	年齢	職業コード	曜日コード	出発地施設	到着地コード	伊予鉄高島	三越松山店	フジグラン松	ジャスコ松山	ジョーブラま	代表移動手段コード	
2	17147	1	33	1	1	140	5212	1421.701	1670.25	2811.08	420.5063	551.331	200	
3	19473	1	33	1	1	250	5272	6228.873	6563.629	7557.346	4570.242	4365.488	100	
4	22842	1	40	1	6	110	5212	4147.13	3767.651	3029.864	5773.398	5962.611	100	
5	15767	2	32	1	3	120	5206	1140.883	991.9029	620.2101	2885.94	3003.848	300	
6	20405	2	32	1	4	120	5206	1140.883	991.9029	620.2101	2885.94	3003.848	300	
7	16888	2	30	1	7	190	5202	2518.073	2575.195	1128.882	4349.156	4380.695	100	
8	17931	2	30	1	3	130	5272	1067.106	951.3783	616.6257	2822.246	2934.433	410	
9	18232	2	30	1	4	130	5272	1067.106	951.3783	616.6257	2822.246	2934.433	410	
10	19828	2	30	1	2	220	5272	1486.224	2057.807	2729.673	1157.349	837.7301	410	

式で表すと…

$$
\begin{aligned}
V_{takashi} &= d_1 * x_{dis-ta} & & & + b_1 * I \\
V_{mitsuko} &= d_2 * x_{dis-mi} & + a_2 * x_{40over} & & + b_2 * I \\
V_{matsu} &= d_3 * x_{dis-ma} & & & + b_3 * I \\
V_{jusco} &= d_4 * x_{dis-ju} & & & + b_4 * I \\
V_{joepla} &= d_5 * x_{dis-jo} & + w_5 * x_{woman} & &
\end{aligned}
$$

確定項　出発地と選択肢の距離　年齢ダミー　女性ダミー　定数項

復習：定数項は選択肢数より少なくなるように

2.2. 推定の流れを追っていきます

$$P_{takashi} = \frac{e^{V_{takashi}}}{e^{V_{takashi}} + e^{V_{mitsuko}} + e^{V_{matsu}} + e^{V_{jusco}} + e^{V_{joepla}}}$$

復習:logit modelの選択確率の式

```
### 選択確率の計算:
## 分母となる，各々のexp(V)の和をつくる
deno <- (takashi + mitsuko + matsu + jusco +joepla)

## それぞれ計算する
Ptakashi <- takashi / deno
Pmitsuko <- mitsuko / deno
Pmatsu   <- matsu   / deno
Pjusco   <- jusco   / deno
Pjoepla  <- joepla  / deno

## 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
Ptakashi <- (Ptakashi!=0)*Ptakashi + (Ptakashi==0)
Pmitsuko <- (Pmitsuko!=0)*Pmitsuko + (Pmitsuko==0)
Pmatsu   <- (Pmatsu  !=0)*Pmatsu   + (Pmatsu  ==0)
Pjusco   <- (Pjusco  !=0)*Pjusco   + (Pjusco  ==0)
Pjoepla  <- (Pjoepla !=0)*Pjoepla  + (Pjoepla ==0)

## 選択結果
Ctakashi <- Data[,7]==5212
Cmitsuko <- Data[,7]==5213
Cmatsu   <- Data[,7]==5202
Cjusco   <- Data[,7]==5272
Cjoepla  <- Data[,7]==5303

## 対数尤度の計算
LL <- colSums(Ctakashi*log(Ptakashi) + Cmitsuko*log(Pmitsuko) +
              Cmatsu  *log(Pmatsu)   + Cjusco  *log(Pjusco)   + Cjoepla*log(Pjoepla))

}


## 対数尤度関数frの最大化
res <- optim(b0,fr, method = "BFGS", hessian = TRUE, control=list(fnscale=-1))
```

各選択肢の選択確率

計算上の処理

選択結果の読み込み

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こと(詳細は省略)

数値計算方法の指定
(ほかにもいろいろある)

2.2. 推定の流れを追っていきます

```
## estimated parameter
b    <- res$par
hhh <- res$hessian

## t値の計算
tval <- b/sqrt(-diag(solve(hhh)))

## 初期尤度
L0 <- hh*log(1/ch)
## 最終尤度
LL <- res$value
## 適合度の計算



## 結果の出力
##ρ^2値
print((L0-LL)/L0)
## 修正済ρ^2値
print((L0-(LL-length(b)))/L0)


print(res)
print(tval)
```

推定結果について，t値，尤度比を計算
（詳細は省略）

「パラメータ推定した後，得られた結果が有意かどうか？」

2.3. 推定結果

```
> ## 結果の出力
> ##ρ^2値
> print((L0-LL)/L0)
[1] 0.2809693
> ## 修正済ρ^2値
> print((L0-(LL-length(b)))/L0)
[1] 0.2571551
>
>
> print(res)
$par
 [1]  1.73583335  0.08750992  1.48217027  1.18417522 -0.79413283
 [6] -1.18444108 -0.95048348 -0.84977429 -0.88747670  1.34897282
[11]  1.73061252

$value
[1] -332.1265
```

尤度比

推定されたパラメータ値（設定順）

```
> print(tval)
 [1]  2.5405526  0.1028183  2.0132793  1.7558484 -7.8802568 -6.1700941
 [7] -6.5794795 -6.0064356 -6.0008643  2.0808330  3.4099947
```

各パラメータのt値（設定順）

正常にまわったら・・・

- 尤度比はどうか？
- パラメータの符号は仮定に反してないか？
- パラメータは有意に効いているか？T値の値はどうか？

適宜パラメータの入れ替え，関数形の変更などなどを通して，どうすればより良い推定結果になるのか試行錯誤を行い，推定を繰り返してみる．

というのが推定の基本的な手順です！

2.4. エラー例

```
### データファイルの読み込み
Data <- read.csv("C:/演習課題/サンプル/sampl.csv",header=T)

## データ数:Dataの行数を数える
hh <- nrow(Data)
```

```
$データファイルの読み込み
$ <- read.csv("C:/R/sampl.csv",header=T)
 $エラー file(file, "rt") :  コネクションを開くことができません
 $報:    警告メッセージ:
$e(file, "rt") :
 $イル 'C:/R/sampl.csv' を開くことができません: No such file or directory
```

```
## パラメータの初期値の設定
b0<-c(0, 0, 0, 0, 0, 0, 0, 0, 0)
```

```
+ }
>
>
>
> ## 対数尤度関数frの最大化
> res <- optim(b0,fr, method = "BFGS", hessian = TRUE, control=list(fnscal$
 以下にエラー optim(b0, fr, method = "BFGS", hessian = TRUE, control = lis$
   vmmin の初期値が有限ではありません
>
>
```

最後の推定結果に至るまでに，途中でこのように青い文字が表示されていたら，何かエラーが起きているということ．修正する必要がある

2.4. 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

等々…

気を付けたつもりでも意外と細かいところでミスっててエラーが出てくる，といったことが多かったりするのでしっかり注意．

さっきの例以外にも「線型方程式は正確に特異です」などをはじめ，意味不明なエラーを示すこともしばしば．

特にはじめのうちは，エラーが出たら，ひとまず近くにいる先輩に聞いてみたほうがはやいかもしれません．

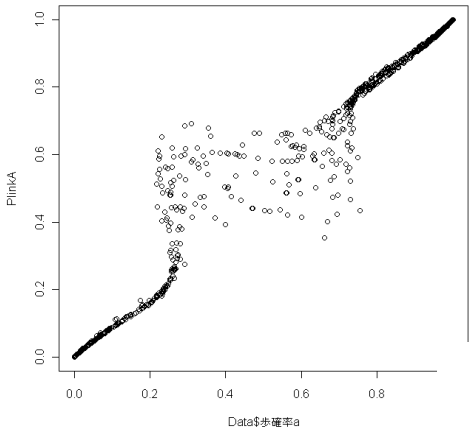
2.5. その他

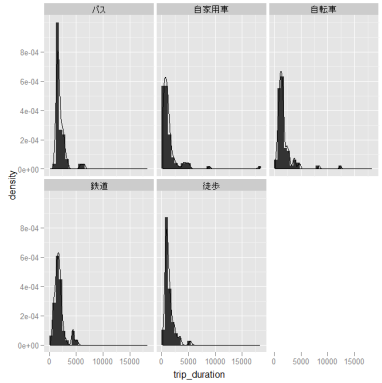
- 今回はスタートアップゼミということで，「どんな感じで推定するのか？」という流れを簡単に説明しました．

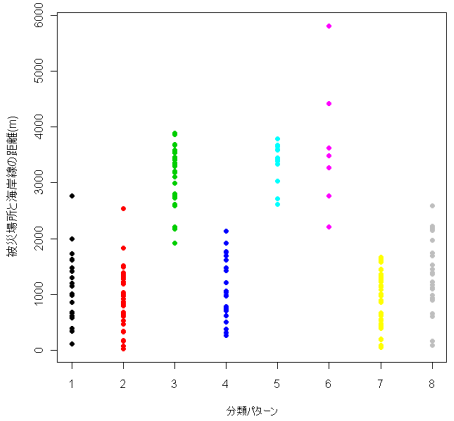
- 最尤推定の話とか，統計的検定の話とか，数学的な部分を省略しているところがあるので，その辺りの仕組みについては各自青本で勉強してください．

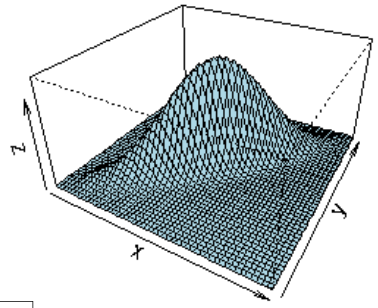
- 今後実際に卒論で使うのは11月とか12月頃だと思うので，その頃また確認してもらえたらと思います．今の段階ではイメージとして覚えていてもらえれば十分です．

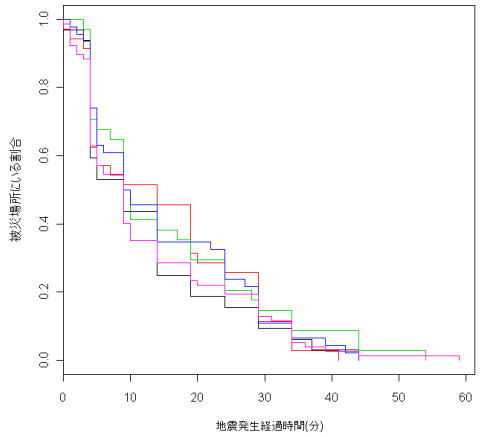
おまけ：推定以外にも

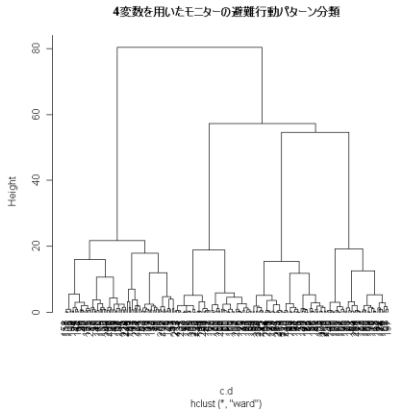


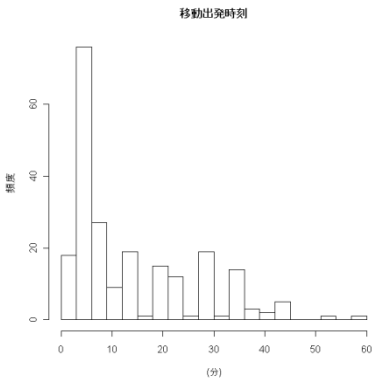












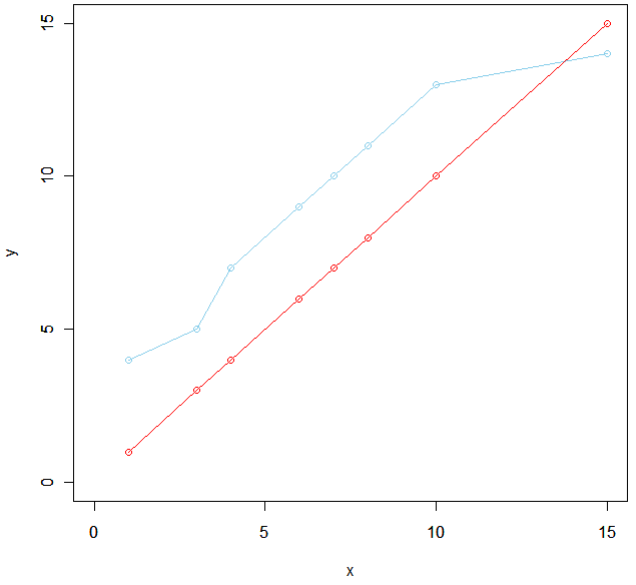
グラフの作成に役立ったりもします

例 プロット図

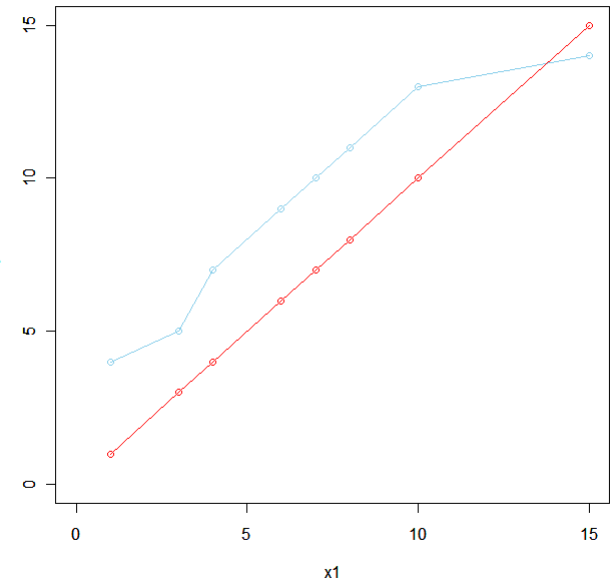
```
### plot
x1 <- c(1, 3, 4, 6, 7, 8, 10, 15)
y1 <- c(4, 5, 7, 9, 10, 11, 13, 14)
y2 <- c(1, 3, 4, 6, 7, 8, 10, 15)

plot(x1, y1)
plot(x1, y1, type="p")
plot(x1, y1, type="l")
plot(x1, y1, type="o")

plot(x1, y2)
```



引数	機能
type="p"	点プロット(デフォルト)
type="l"	線プロット(折れ線グラ
type="b"	点と線のプロット
type="c"	"b" において点を描かな
type="o"	点プロットと線プロットの
type="h"	各点から x 軸までの垂
type="s"	左側の値にもとづいて階
type="S"	右側の値にもとづいて階
type="n"	軸だけ描いてプロットし



おわり

- R-tips
  - http://cse.naro.affrc.go.jp/takezawa/r-tips/r.html
  - 基本的な使い方はここでマスターすると良い
- RjpWiki
  - http://www.okada.jp.org/RWiki/index.php?RjpWiki
  - 日本最大のRコミュニティ
  - 情報量が半端ないのでサイト内検索を利用
  - 新規パッケージがどんどん更新される
- seekR
  - http://seekr.jp/
  - Rは文字1つなので検索しづらい
  - Rに特化した検索エンジンなので便利
- Rのメモ
  - http://bin.t.u-tokyo.ac.jp/~hara/r.html

- 2010年のB4ゼミで，原さんがより高度な事もまとめてくださっているので，気になる方はそちらを参照してみてください．