

項目パラメタの推定 — 1PL, 2PL, 3PLモデル

四分相関係数の計算

```
library(polycor)
library(psych)
オブジェクト名 <- polychoric(データフレーム名)
相関行列名 <- オブジェクト名$rho
```

あらかじめpolycorパッケージとpsychパッケージをインストールしておく必要がある。
polycorの出力には四分相関係数以外のものも含まれるので、\$rhoで四分相関係数だけを切り取る。

一次元性の確認

```
VSS.scree(相関行列名)
```

スクリープロットを表示する。

項目パラメタの推定

```
library(ltm)
library(irtoys)
オブジェクト名 <- est(resp=データフレーム名, model="モデル名", engine="ltm")
```

あらかじめltmパッケージとirtoysパッケージをインストールしておく必要がある。
model には "1PL", "2PL", "3PL" を指定できる。
engine には "ic1", "bilog", "ltm" を指定できる。"ltm" にしておくとRだけで完結するが、"bilog"などだとするとBILOGソフトが別途必要になる。

```
> setwd("i:\Rdocuments\scripts")
> d1 <- read.table("irt_データ.csv", header=TRUE, sep=",")
> head(d1)
  s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
1    1    1    1    1    1    0    1    1    1    1    1
2    1    1    1    0    1    0    0    1    0    0    0
3    1    1    1    1    1    1    1    1    1    1    1
4    1    1    1    0    1    0    0    0    0    0    0
5    1    1    1    0    1    0    0    1    1    1    0
6    1    1    0    1    0    0    0    0    0    1    0
>
> # 人数
> nrow(d1)
[1] 778
>
```

```
> # 項目正答率
> pv <- colMeans(d1)
> pv
      s03      s04      s05      s06      s07
0.71208226 0.82519280 0.73778920 0.73007712 0.88688946 0.06298201 0.56683805
      s12      s14      s15      s16
0.53727506 0.55398458 0.58097686 0.44344473
>
>
```

> # 四分相関係数の計算

```
> library(polycor)
> library(psych)
> t.cor <- polychoric(d1)
> t.cor <- t.cor$rho
> round(t.cor, 2)
```

```
      s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
s03 1.00 0.06 0.33 0.34 0.21 0.10 0.25 0.30 0.12 0.15 0.27
s04 0.06 1.00 0.31 0.33 0.24 0.16 0.10 0.15 0.17 0.16 0.21
s05 0.33 0.31 1.00 0.52 0.29 0.35 0.32 0.40 0.26 0.24 0.29
s06 0.34 0.33 0.52 1.00 0.48 0.14 0.38 0.38 0.30 0.25 0.40
s07 0.21 0.24 0.29 0.48 1.00 0.36 0.48 0.42 0.48 0.37 0.41
```

	A	B	C	D	E	F	G	H	I	J	K
1	s03	s04	s05	s06	s07	s08	s09	s12	s14	s15	s16
2	1	1	1	1	1	0	1	1	1	1	1
3	1	1	1	0	1	0	0	1	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	0	1	0	0	0	0	0	0
6	1	1	1	0	1	0	0	1	1	1	0
7	1	1	0	1	0	0	0	0	0	1	0
8	1	0	1	1	1	0	1	0	1	0	0
9	0	1	1	1	1	0	0	1	1	0	1
10	0	1	0	0	0	0	0	0	0	0	0
11	1	1	1	1	1	0	1	1	1	1	1
12	1	1	1	1	1	0	0	1	0	0	0
13	1	1	1	1	1	0	1	1	1	0	0
14	1	1	1	1	1	0	1	1	1	1	1
15	0	1	1	1	0	0	0	1	0	1	1
16	1	0	1	1	1	0	1	1	0	0	0
17	1	1	1	0	0	0	0	1	1	0	1
18	1	1	1	1	1	0	0	0	1	1	1
19	1	1	1	1	1	0	0	1	0	1	1
20	1	1	1	0	1	0	0	0	0	1	0
21	0	1	1	1	1	1	1	0	1	1	1

```

s08 0.10 0.16 0.35 0.14 0.36 1.00 0.38 0.29 0.13 0.27 0.24
s09 0.25 0.10 0.32 0.38 0.48 0.38 1.00 0.27 0.27 0.33 0.35
s12 0.30 0.15 0.40 0.38 0.42 0.29 0.27 1.00 0.40 0.28 0.53
s14 0.12 0.17 0.26 0.30 0.48 0.13 0.27 0.40 1.00 0.31 0.35
s15 0.15 0.16 0.24 0.25 0.37 0.27 0.33 0.28 0.31 1.00 0.35
s16 0.27 0.21 0.29 0.40 0.41 0.24 0.35 0.53 0.35 0.35 1.00
>

```

```

> # 一次元性の確認
> VSS.scree(t.cor)
>
>
> # 項目母数の推定
> library(ltm)
> library(irtoys)
> p.all <- est(resp=d1, model="2PL", engine="ltm")
> colnames(p.all$est) <- c("aj", "bj", "cj")
> colnames(p.all$se) <- c("aj", "bj", "cj")
> rownames(p.all$se) <- rownames(p.all$est)
> p.all

```

```

$est
      aj      bj  cj
s03 0.7519081 -1.3479054  0
s04 0.6336668 -2.6447011  0
s05 1.2141256 -1.0870115  0
s06 1.4940753 -0.9263137  0
s07 1.7233574 -1.7290162  0
s08 0.9815908  3.1497698  0
s09 1.0866067 -0.3036178  0
s12 1.4498429 -0.1380151  0
s14 1.0159834 -0.2558882  0
s15 0.9046749 -0.4216555  0
s16 1.4752955  0.2216158  0

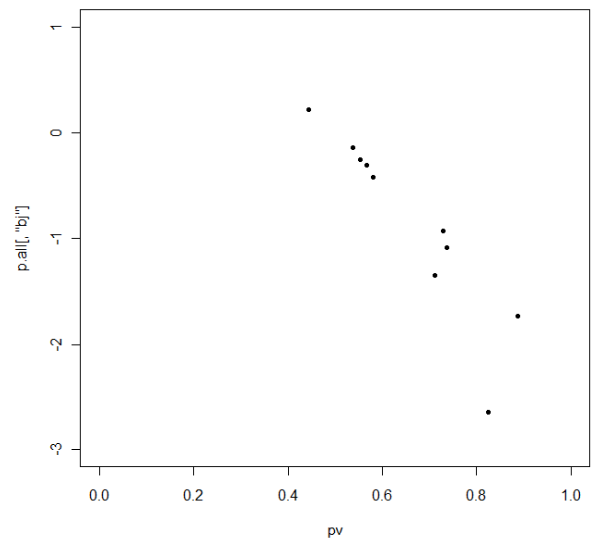
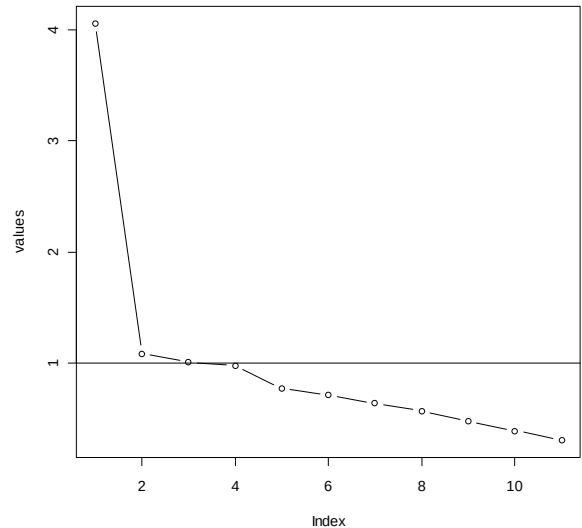
```

```

$se
      aj      bj  cj
s03 0.1155375 0.20441485  0
s04 0.1250574 0.47668423  0
s05 0.1526624 0.12175749  0
s06 0.1800035 0.09653309  0
s07 0.2457579 0.15253089  0
s08 0.2276118 0.58227949  0
s09 0.1333925 0.08662509  0
s12 0.1680297 0.06968662  0
s14 0.1275844 0.08936450  0
s15 0.1194251 0.10329955  0
s16 0.1764927 0.06974481  0

```

scree plot



```

>
> # 項目正答率と困難度の散布図
> plot(pv, p.all$est[, "bj"], xlim=c(0, 1), ylim=c(-3, 1), pch=20)

```

特性関数，情報関数の図示

```
library(irtoys)
```

あらかじめltmパッケージとirtoysパッケージをインストールしておく必要がある．

項目特性曲線

```
plot(irf(項目パラメタデータ行列))
```

項目情報曲線

```
plot(iif(項目パラメタデータ行列))
```

テスト特性曲線

```
plot(trf(項目パラメタデータ行列))
```

テスト情報曲線

```
plot(tif(項目パラメタデータ行列))
```

項目パラメタデータ行列を1項目の項目パラメタだけにすれば，その項目だけの特性曲線を描く．
項目パラメタデータ行列に複数の項目パラメタを入れれば，それらすべての特性曲線を描く．

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> dl <- read.table("irt_データ.csv", header=TRUE, sep=",")
>
>
>
> # 項目母数の推定
> library(ltm)
> library(irtoys)
> p.all <- est(resp=dl, model="2PL", engine="ltm")
> colnames(p.all$est) <- c("aj", "bj", "cj")
> colnames(p.all$sse) <- c("aj", "bj", "cj")
> rownames(p.all$sse) <- rownames(p.all$est)
> p.all
```

```
$est
      aj      bj  cj
s03 0.7519081 -1.3479054 0
s04 0.6336668 -2.6447011 0
s05 1.2141256 -1.0870115 0
s06 1.4940753 -0.9263137 0
s07 1.7233574 -1.7290162 0
s08 0.9815908  3.1497698 0
s09 1.0866067 -0.3036178 0
s12 1.4498429 -0.1380151 0
s14 1.0159834 -0.2558882 0
s15 0.9046749 -0.4216555 0
s16 1.4752955  0.2216158 0
```

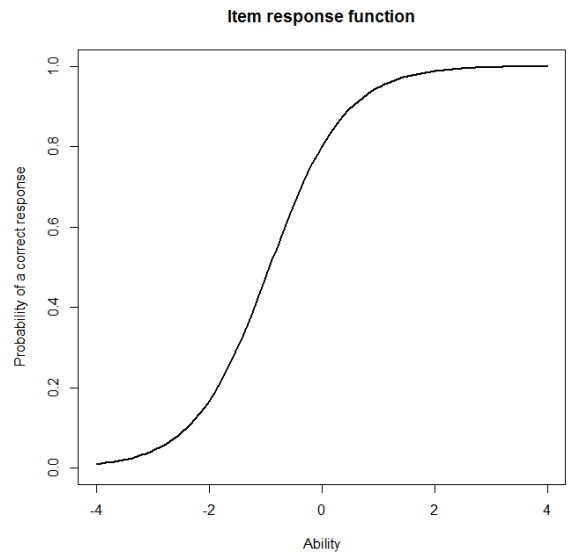
```
$sse
      aj      bj  cj
s03 0.1155375 0.20441485 0
s04 0.1250574 0.47668423 0
s05 0.1526624 0.12175749 0
s06 0.1800035 0.09653309 0
s07 0.2457579 0.15253089 0
s08 0.2276118 0.58227949 0
s09 0.1333925 0.08662509 0
s12 0.1680297 0.06968662 0
s14 0.1275844 0.08936450 0
s15 0.1194251 0.10329955 0
s16 0.1764927 0.06974481 0
```

	A	B	C	D	E	F	G	H	I	J	K
1	s03	s04	s05	s06	s07	s08	s09	s12	s14	s15	s16
2	1	1	1	1	1	0	1	1	1	1	1
3	1	1	1	0	1	0	0	1	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	0	1	0	0	0	0	0	0
6	1	1	1	0	1	0	0	1	1	1	0
7	1	1	0	1	0	0	0	0	0	1	0
8	1	0	1	1	1	0	1	0	1	0	0
9	0	1	1	1	1	0	0	1	1	0	1
10	0	1	0	0	0	0	0	0	0	0	0
11	1	1	1	1	1	0	1	1	1	1	1
12	1	1	1	1	1	0	0	1	0	0	0
13	1	1	1	1	1	0	1	1	1	0	0
14	1	1	1	1	1	0	1	1	1	1	1
15	0	1	1	1	0	0	0	1	0	1	1
16	1	0	1	1	1	0	1	1	0	0	0
17	1	1	1	0	0	0	0	1	1	0	1
18	1	1	1	1	1	0	0	0	1	1	1
19	1	1	1	1	1	0	0	1	0	1	1
20	1	1	1	0	1	0	0	0	0	1	0
21	0	1	1	1	1	1	1	0	1	1	1

```

>
> # 特性関数, 情報関数の図示
>
> # 項目特性曲線
> plot(irf(p.all$est[~s06",]))
>
>

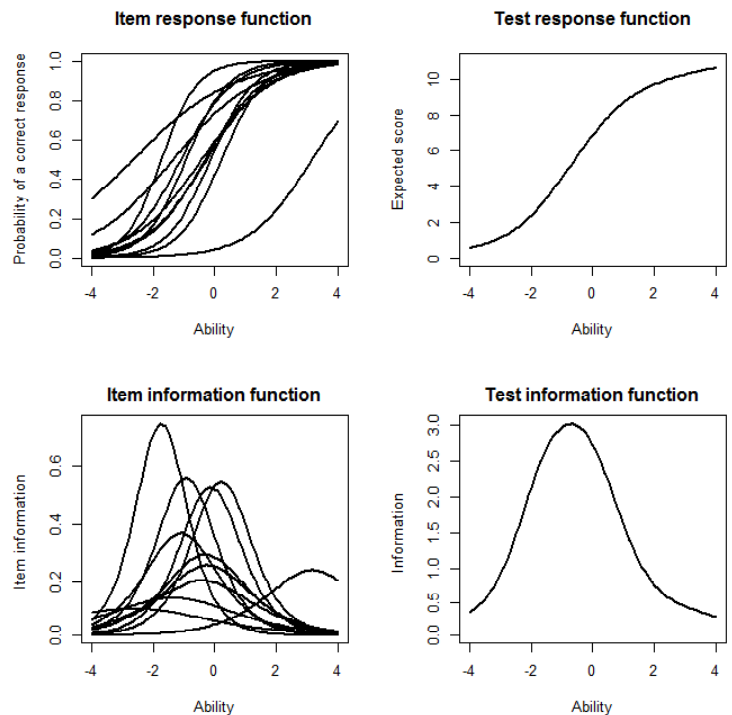
```



```

> # 図出力画面を2*2に分割
> par(mfrow=c(2,2))
>
> # 項目特性曲線
> plot(irf(p.all$est))
>
> # テスト特性曲線
> plot(trf(p.all$est))
>
> # 項目情報曲線
> plot(iif(p.all$est))
>
> # テスト情報曲線
> plot(tif(p.all$est))
>
>
> # 図出力画面を1*1に戻す
> par(mfrow=c(1,1))
>

```



特性値の推定

```
library(irtos)
```

あらかじめirtosパッケージをインストールしておく必要がある。

最尤推定値・ベイズモーダル推定値

```
mlebm(resp=データ行列, ip=項目パラメタ行列, method="推定方法")
```

method を"ML"または省略すると最尤推定値, "BM"とするとベイズモーダル推定値を計算する.
"BM"とした場合は mu= と sigma= で事前分布 (正規分布) を指定する. デフォルトは標準正規分布.

期待事後推定値

```
eap(resp=データ行列, ip=項目パラメタ行列, qu=normal.qu())
```

qu は求積点と重みの指定をする. 標準正規分布の場合 qu=normal.qu() とする.

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> dl <- read.table("irt_データ.csv", header=TRUE, sep=",")
> head(dl)
  s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
1    1    1    1    1    1    0    1    1    1    1    1
2    1    1    1    0    1    0    0    1    0    0    0
3    1    1    1    1    1    1    1    1    1    1    1
4    1    1    1    0    1    0    0    0    0    0    0
5    1    1    1    0    1    0    0    1    1    1    0
6    1    1    0    1    0    0    0    0    0    1    0
>
>
> # 項目母数の推定
> library(ltm)
> library(irtos)
> p.all <- est(resp=dl, model="2PL", engine="ltm")
> colnames(p.all$est) <- c("aj", "bj", "cj")
> colnames(p.all$se) <- c("aj", "bj", "cj")
> rownames(p.all$se) <- rownames(p.all$est)
> p.all
$est
      aj      bj  cj
s03 0.7519081 -1.3479054 0
s04 0.6336668 -2.6447011 0
s05 1.2141256 -1.0870115 0
s06 1.4940753 -0.9263137 0
s07 1.7233574 -1.7290162 0
s08 0.9815908  3.1497698 0
s09 1.0866067 -0.3036178 0
s12 1.4498429 -0.1380151 0
s14 1.0159834 -0.2558882 0
s15 0.9046749 -0.4216555 0
s16 1.4752955  0.2216158 0

$se
      aj      bj  cj
s03 0.1155375 0.20441485 0
s04 0.1250574 0.47668423 0
s05 0.1526624 0.12175749 0
s06 0.1800035 0.09653309 0
s07 0.2457579 0.15253089 0
s08 0.2276118 0.58227949 0
s09 0.1333925 0.08662509 0
s12 0.1680297 0.06968662 0
s14 0.1275844 0.08936450 0
s15 0.1194251 0.10329955 0
s16 0.1764927 0.06974481 0
>
```

	A	B	C	D	E	F	G	H	I	J	K
1	s03	s04	s05	s06	s07	s08	s09	s12	s14	s15	s16
2	1	1	1	1	1	0	1	1	1	1	1
3	1	1	1	0	1	0	0	1	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	0	1	0	0	0	0	0	0
6	1	1	1	0	1	0	0	1	1	1	0
7	1	1	0	1	0	0	0	0	0	1	0
8	1	0	1	1	1	0	1	0	1	0	0
9	0	1	1	1	1	0	0	1	1	0	1
10	0	1	0	0	0	0	0	0	0	0	0
11	1	1	1	1	1	0	1	1	1	1	1
12	1	1	1	1	1	0	0	1	0	0	0
13	1	1	1	1	1	0	1	1	1	0	0
14	1	1	1	1	1	0	1	1	1	1	1
15	0	1	1	1	0	0	0	1	0	1	1
16	1	0	1	1	1	0	1	1	0	0	0
17	1	1	1	0	0	0	0	1	1	0	1
18	1	1	1	1	1	0	0	0	1	1	1
19	1	1	1	1	1	0	0	1	0	1	1
20	1	1	1	0	1	0	0	0	0	1	0
21	0	1	1	1	1	1	1	0	1	1	1

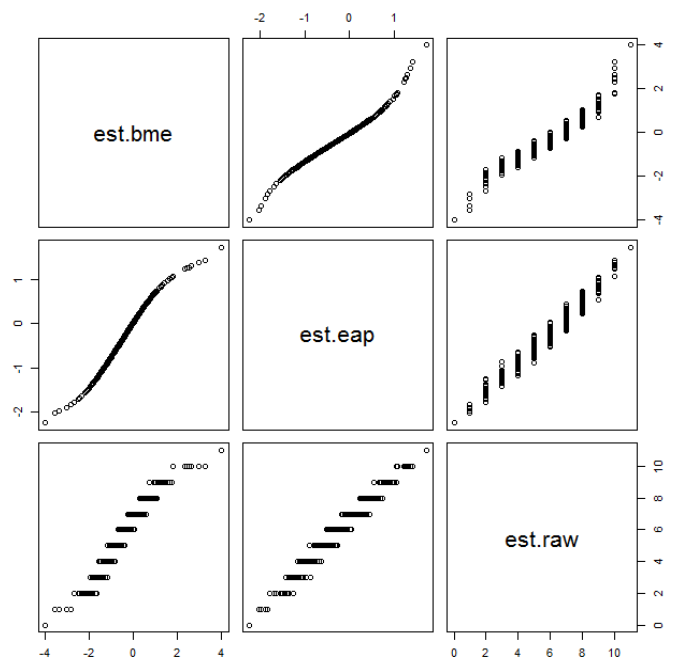
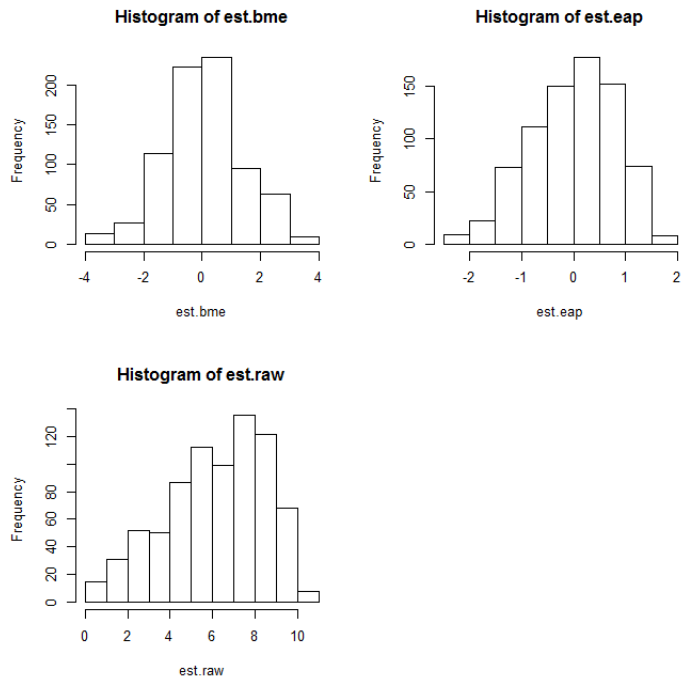
```
> # 特性値の推定
> th.bme <- mlebme(resp=d1, ip=p.all$est)
> head(th.bme)
```

```
      est      sem  n
[1,]  2.4947583 1.3271304 11
[2,] -0.7873886 0.5763237 11
[3,]  3.9999379 1.9051382 11
[4,] -1.2792655 0.5933037 11
[5,] -0.1386948 0.5942225 11
[6,] -1.4734953 0.6085549 11
```

```
> th.eap <- eap(resp=d1, ip=p.all$est, qu=normal.qu())
> head(th.eap)
```

```
      est      sem  n
[1,]  1.26809690 0.6508009 11
[2,] -0.58653249 0.5089365 11
[3,]  1.71697773 0.7015662 11
[4,] -0.96204634 0.5107556 11
[5,] -0.07748696 0.5242392 11
[6,] -1.10349864 0.5143460 11
```

```
> # 特性値のヒストグラム
> est.bme <- th.bme[,c("est")]
> est.eap <- th.eap[,c("est")]
> est.raw <- rowSums(d1)
>
> par(mfrow=c(2,2))
> hist(est.bme)
> hist(est.eap)
> hist(est.raw)
> par(mfrow=c(1,1))
>
> # 特性値の散布図
> ests <- cbind(est.bme, est.eap, est.raw)
> pairs(ests)
>
```



等化 — 共通項目法

```
library(irtoys)
```

あらかじめirtoysパッケージをインストールしておく必要がある。

```
sca(old.ip = 基準とする冊子の項目パラメタ行列,
    old.items = 基準冊子における共通項目の名前または列番号,
    new.ip = 基準に等化したい冊子の項目パラメタ行列,
    new.items = 基準に等化したい冊子における共通項目の名前または列番号,
    method = "等化法")
```

method には "MM" (Mean/Mean), "MS" (Mean/Sigma), "SL" (Stocking-Lord), "HB" (Haebara) を指定できる。デフォルトは "MS" である。

method に "SL" または "HB" を用いた場合は, old.qu=norma.qu(), new.qu=norma.qu(), bec=FALSE などの指定も必要になる。

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
>
> # 冊子Sのデータファイルの読み込み
> ds <- read.table("irt_データ1.csv", header=TRUE, sep=",")
> head(ds)
```

```
  s03 s04 s05 s06 s07 s08 s09 s12
1    1    1    1    1    1    0    1    1
2    1    1    1    0    1    0    0    1
3    1    1    1    1    1    1    1    1
4    1    1    1    0    1    0    0    0
5    1    1    1    0    1    0    0    1
6    1    1    0    1    0    0    0    0
```

```
> # 人数
> nrow(ds)
[1] 380
>
```

```
> # 冊子tのデータファイルの読み込み
> dt <- read.table("irt_データ2.csv", header=TRUE, sep=",")
> head(dt)
```

```
  t03 t04 t05 t06 t12 t14 t15 t16
1    1    1    0    0    0    1    1    0
2    0    1    1    1    1    1    0    0
3    0    1    1    1    0    1    0    0
4    1    1    1    1    1    1    1    1
5    1    0    1    1    1    1    1    0
6    0    0    0    1    0    0    0    0
```

```
> # 人数
> nrow(dt)
[1] 398
>
```

```
> # 四分相関係数の計算
> library(polycor)
> library(psych)
> t.cor.s <- polychoric(ds)
> t.cor.s <- t.cor.s$rho
> round(t.cor.s, 2)
```

```
      s03 s04 s05 s06 s07 s08 s09 s12
s03  1.00 -0.02 0.36 0.28 0.24 0.06 0.15 0.26
s04 -0.02  1.00 0.35 0.42 0.19 0.03 -0.03 0.21
s05  0.36  0.35  1.00 0.46 0.35 0.31 0.26 0.45
s06  0.28  0.42  0.46  1.00 0.41 0.09 0.32 0.33
s07  0.24  0.19  0.35  0.41  1.00 0.92 0.40 0.41
s08  0.06  0.03  0.31  0.09  0.92  1.00 0.36 0.22
s09  0.15 -0.03  0.26  0.32  0.40  0.36  1.00 0.23
s12  0.26  0.21  0.45  0.33  0.41  0.22  0.23  1.00
```

冊子s データ

	A	B	C	D	E	F	G	H
1	s03	s04	s05	s06	s07	s08	s09	s12
2	1	1	1	1	1	0	1	1
3	1	1	1	0	1	0	0	1
4	1	1	1	1	1	1	1	1
5	1	1	1	0	1	0	0	0
6	1	1	1	0	1	0	0	1
7	1	1	0	1	0	0	0	0
8	1	0	1	1	1	0	1	0
9	0	1	1	1	1	0	0	1
10	0	1	0	0	0	0	0	0
11	1	1	1	1	1	0	1	1
12	1	1	1	1	1	0	0	1
13	1	1	1	1	1	0	1	1
14	1	1	1	1	1	0	1	1
15	0	1	1	1	0	0	0	1
16	1	0	1	1	1	0	1	1
17	1	1	1	0	0	0	0	1
18	1	1	1	1	1	0	0	0
19	1	1	1	1	1	0	0	1
20	1	1	1	0	1	0	0	0
21	0	1	1	1	1	1	1	0

冊子t データ

	A	B	C	D	E	F	G	H
1	t03	t04	t05	t06	t12	t14	t15	t16
2	1	1	0	0	0	1	1	0
3	0	1	1	1	1	1	0	0
4	0	1	1	1	0	1	0	0
5	1	1	1	1	1	1	1	1
6	1	0	1	1	1	1	1	0
7	0	0	0	1	0	0	0	0
8	1	0	1	1	0	0	1	0
9	0	0	0	0	1	0	0	1
10	1	1	0	1	1	1	1	1
11	1	1	1	0	0	1	1	0
12	0	1	1	1	1	1	1	1
13	0	1	1	0	0	1	1	1
14	1	1	1	1	0	1	1	1
15	0	0	0	0	0	1	0	0
16	1	1	0	0	0	0	1	0
17	1	1	1	1	1	1	0	0
18	0	1	0	0	0	0	0	0
19	0	1	1	0	0	0	0	0
20	1	0	1	1	0	0	0	0
21	0	1	1	1	0	1	1	0

```

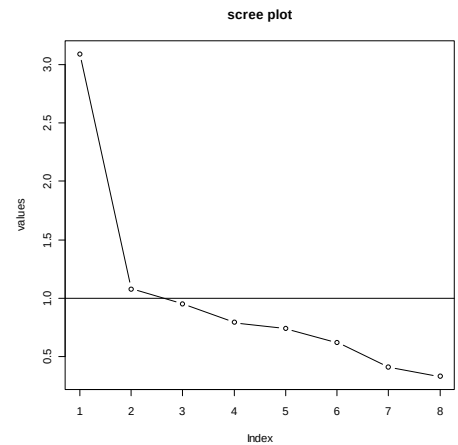
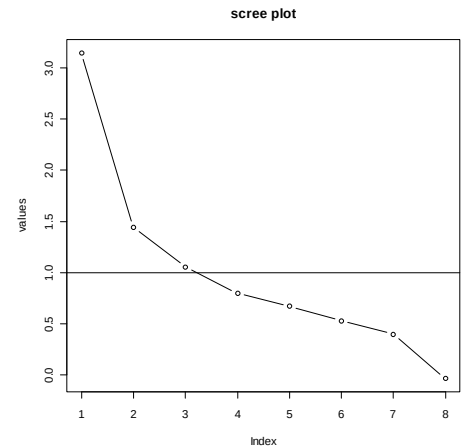
>
> t.cor.t <- polychoric(dt)
> t.cor.t <- t.cor.t$rho
> round(t.cor.t, 2)
      t03  t04  t05  t06  t12  t14  t15  t16
t03 1.00 0.14 0.29 0.40 0.34 0.09 0.23 0.28
t04 0.14 1.00 0.26 0.22 0.08 0.07 0.12 0.35
t05 0.29 0.26 1.00 0.58 0.35 0.24 0.24 0.30
t06 0.40 0.22 0.58 1.00 0.43 0.18 0.24 0.44
t12 0.34 0.08 0.35 0.43 1.00 0.34 0.27 0.55
t14 0.09 0.07 0.24 0.18 0.34 1.00 0.31 0.29
t15 0.23 0.12 0.24 0.24 0.27 0.31 1.00 0.39
t16 0.28 0.35 0.30 0.44 0.55 0.29 0.39 1.00
>
>
> # 一次元性の確認
> VSS.scree(t.cor.s)
> VSS.scree(t.cor.t)
>
>
> # 項目母数の推定
> library(ltm)
> library(irtos)
>
> # 冊子s
> (p.s <- est(resp=ds, model="2PL", engine="ltm"))
$est
      [, 1]      [, 2] [, 3]
s03 0.7705687 -1.35005808 0
s04 0.7313849 -2.17329785 0
s05 1.7760631 -0.81834536 0
s06 1.5016486 -0.77167362 0
s07 1.4794851 -1.74290303 0
s08 0.7880403 3.53510139 0
s09 0.7771965 -0.41847844 0
s12 1.2080520 -0.06943932 0

$se
      [, 1]      [, 2] [, 3]
[1,] 0.1835226 0.3046659 0
[2,] 0.2008428 0.5276427 0
[3,] 0.3662551 0.1233839 0
[4,] 0.2969631 0.1309896 0
[5,] 0.3409735 0.2637599 0
[6,] 0.3097197 1.1783263 0
[7,] 0.1764709 0.1684350 0
[8,] 0.2323375 0.1099113 0

>
> # 冊子t
> (p.t <- est(resp=dt, model="2PL", engine="ltm"))
$est
      [, 1]      [, 2] [, 3]
t03 0.9003816 -1.1510996 0
t04 0.6039492 -2.9640459 0
t05 1.2714928 -1.1421325 0
t06 1.6534735 -1.0351572 0
t12 1.5431110 -0.2142970 0
t14 0.7095130 -0.3159867 0
t15 0.8331013 -0.4619254 0
t16 1.7019090 0.1444933 0

$se
      [, 1]      [, 2] [, 3]
[1,] 0.1830130 0.22550849 0
[2,] 0.1880158 0.84133720 0
[3,] 0.2415768 0.17788000 0
[4,] 0.3139183 0.14187396 0
[5,] 0.2693993 0.09572488 0

```




```

[6,] 0.1569843 0.16815224    0
[7,] 0.1688423 0.15787306    0
[8,] 0.3138695 0.08975633    0
>
>
> # 冊子 t の項目母数を冊子 s の尺度へ等化
> # 共通項目の番号 03, 04, 05, 06, 12

> # irtoys パッケージの読み込み
> library(irtoys)
> (p.tsa <- sca(old.ip=p.s$est, old.items=c(1:4, 8),
+             new.ip=p.t$est, new.items=c(1:5),
+             method="MS"))
$slope
[1] 0.7753178

$intercept
[1] -0.02760581

$scaled.ip
      [,1]      [,2] [,3]
t03 1.1613065 -0.92007377 0
t04 0.7789699 -2.32568328 0
t05 1.6399634 -0.91312149 0
t06 2.1326397 -0.83018161 0
t12 1.9902949 -0.19375408 0
t14 0.9151254 -0.27259594 0
t15 1.0745288 -0.38574476 0
t16 2.1951115 0.08442242 0

> # 項目パラメタだけの取り出し
> (p.ts <- p.tsa$scaled.ip)
      [,1]      [,2] [,3]
t03 1.1613065 -0.92007377 0
t04 0.7789699 -2.32568328 0
t05 1.6399634 -0.91312149 0
t06 2.1326397 -0.83018161 0
t12 1.9902949 -0.19375408 0
t14 0.9151254 -0.27259594 0
t15 1.0745288 -0.38574476 0
t16 2.1951115 0.08442242 0
>

> # 共通項目のパラメタ推定値の統合
> p.sc <- p.s$est[c(1:4, 8),]
> p.tsc <- p.ts[c(1:5),]
> p.c <- p.sc
> p.sc
      [,1]      [,2] [,3]
s03 0.7705687 -1.35005808 0
s04 0.7313849 -2.17329785 0
s05 1.7760631 -0.81834536 0
s06 1.5016486 -0.77167362 0
s12 1.2080520 -0.06943932 0

> p.tsc
      [,1]      [,2] [,3]
t03 1.1613065 -0.9200738 0
t04 0.7789699 -2.3256833 0
t05 1.6399634 -0.9131215 0
t06 2.1326397 -0.8301816 0
t12 1.9902949 -0.1937541 0
>

> # 困難度パラメタ－幾何平均
> p.c[,1] <- sqrt(p.sc[,1] * p.tsc[,1])
>
> # 識別力パラメタ－算術平均
> p.c[,2] <- (p.sc[,2] + p.tsc[,2])/2
>

```

```
> p.c
      [,1]      [,2] [,3]
s03 0.9459738 -1.1350659  0
s04 0.7548025 -2.2494906  0
s05 1.7066571 -0.8657334  0
s06 1.7895462 -0.8009276  0
s12 1.5506062 -0.1315967  0
>

> # 共通でない項目のパラメタ推定値
> p.sr <- p.s$est[c(-1:-4, -8),]
> p.tsr <- p.ts[c(-1:-5),]
```

```
> p.sr
      [,1]      [,2] [,3]
s07 1.4794851 -1.7429030  0
s08 0.7880403  3.5351014  0
s09 0.7771965 -0.4184784  0
>

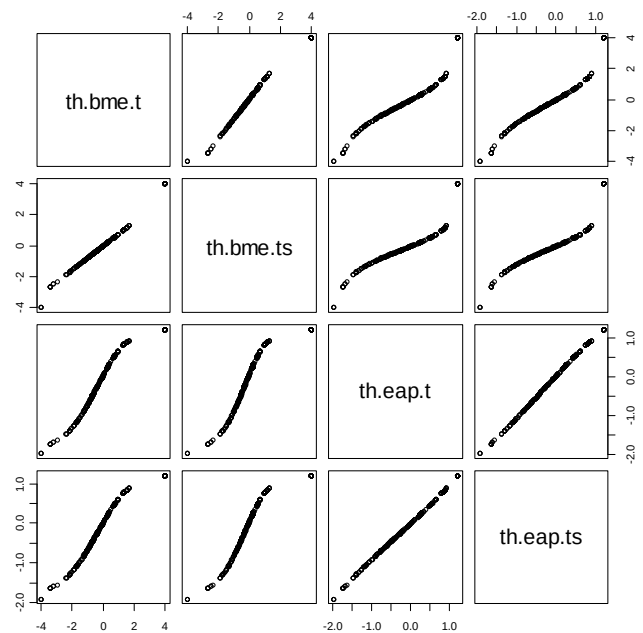
> p.tsr
      [,1]      [,2] [,3]
t14 0.9151254 -0.27259594  0
t15 1.0745288 -0.38574476  0
t16 2.1951115  0.08442242  0
>
```

```
> # 等化による全項目のパラメタ推定値
> p.all2 <- rbind(p.c[c(1:4),], p.sr, p.c[c(5),], p.tsr)
> colnames(p.all2) <- c("aj", "bj", "cj")
> p.all2
      aj      bj  cj
s03 0.9459738 -1.13506592  0
s04 0.7548025 -2.24949057  0
s05 1.7066571 -0.86573343  0
s06 1.7895462 -0.80092762  0
s07 1.4794851 -1.74290303  0
s08 0.7880403  3.53510139  0
s09 0.7771965 -0.41847844  0
      1.5506062 -0.13159670  0      # s12
t14 0.9151254 -0.27259594  0
t15 1.0745288 -0.38574476  0
t16 2.1951115  0.08442242  0
>
```

```
> # 冊子tの特性値の推定
> th.bme.t <- mlebme(resp=dt, ip=p.t$est)
> th.bme.ts <- mlebme(resp=dt, ip=p.ts)
> th.eap.t <- eap(resp=dt, ip=p.t$est, qu=normal.qu())
> th.eap.ts <- eap(resp=dt, ip=p.ts, qu=normal.qu())
> th.bme.t <- th.bme.t[,c("est")]
> th.bme.ts <- th.bme.ts[,c("est")]
> th.eap.t <- th.eap.t[,c("est")]
> th.eap.ts <- th.eap.ts[,c("est")]
>

> # 特性値の散布図
> ests <- cbind(th.bme.t, th.bme.ts, th.eap.t, th.eap.ts)
> pairs(ests)
> round(cor(ests), 3)
      th.bme.t th.bme.ts th.eap.t th.eap.ts
th.bme.t    1.000    0.993    0.943    0.953
th.bme.ts    0.993    1.000    0.901    0.914
th.eap.t     0.943    0.901    1.000    0.999
th.eap.ts    0.953    0.914    0.999    1.000
```

```
# 同じ推定法であれば、等化の
# 前後の特性値はほぼ完全な相関
```



DIF分析

パッケージの読み込み

```
library(difR)
```

あらかじめdifRパッケージを読み込んでおく必要がある

一般化Lordカイ2乗法

```
difGenLord(データ名, group="グループ変数名", focal.name=c("焦点群1", "焦点群2", ...),
            nrFocal=焦点群の数, model="IRTモデル名")
```

一般化マンテル-ヘンツェル法

```
difGMH(データ名, group="グループ変数名", focal.name=c("焦点群1", "焦点群2", ...))
```

一般化ロジスティック回帰法

```
difGenLogistic(データ名, group="グループ変数名", focal.name=c("焦点群1", "焦点群2", ...))
```

3つの分析結果の比較

```
genDichoDif(データ名, group="グループ変数名", focal.names=c("焦点群1", "焦点群2", ...),
            method=c("GMH", "genLogistic", "genLord"), nrFocal=焦点群の数, model="IRTモデル名")
```

データは、分析に用いる項目と、グループ変数だけからなる。

IRTモデル名： 1PL, 2PL, 3PL

カイ2乗値のプロット

```
plot(DIF分析オブジェクト名)
```

スケーリングした項目母数

```
群名 <- c("参照群名", "焦点群1", "焦点群2", ...)
```

```
項目母数オブジェクト名 <- DIF分析オブジェクト名$itemParInit
```

```
rownames(項目母数オブジェクト名) <- NULL
```

```
items <- rep(DIF分析オブジェクト名$names, length(群名))
```

```
group <- rep(群名, each=length(DIF分析オブジェクト名$names))
```

```
(項目母数オブジェクト名 <- data.frame(items, 項目母数オブジェクト名, group))
```

項目特性曲線の描画

```
plot(DIF分析オブジェクト名, plot="itemCurve", item=変数番号, ref.name="参照群名")
```

itemは、"変数名"で指定してもよい

```
> setwd("i:\¥¥Rdocuments¥¥scripts¥¥")
```

```
> dA11 <- read.table("DIF10.csv", header=TRUE, sep=",")
```

```
> head(dA11)
```

```
Student.ID Age Group Gender Item1 Item2 Item3 Item4 Item5 Item6 Item7 Item8 Item9 Item10
1          1  14 Minhla female      1      1      1      0      0      1      0      0      0      0
2          1  14 Minhla female      1      1      1      1      1      1      1      1      1      1
3          2  14 Minhla  male      1      1      1      1      1      1      1      0      0      1
4          2  14 Minhla female      1      1      1      1      1      1      0      0      0      1
5          2  14 Minhla  male      1      1      1      1      1      1      1      1      1      1
6          3  14 Minhla  male      1      1      1      1      1      1      1      1      1      1>
```

```
> # dataset
```

```
> dEth <- dA11[, c("Item1", "Item2", "Item3", "Item4", "Item5",
+                 "Item6", "Item7", "Item8", "Item9", "Item10", "Group")]
```

```
>
```

```

> # DIF analysis
>
> library(difR)

> # 2PL model, "ltm" engine
>
> #Lord chi-squares, p values, and DIF items
> (rEth <- difGenLord(dEth, group="Group", focal.name=c("Sittway", "Minhla"),
+                   nrFocal=2, model="2PL"))

```

Detection of Differential Item Functioning using generalized Lord's method (2 focal groups), with 2PL model and without item purification

Engine 'ltm' for item parameter estimation

Generalized Lord's chi-square statistic:

	Stat.	P-value
Item1	1.8699	0.7597
Item2	17.6189	0.0015 **
Item3	0.2782	0.9912
Item4	4.0210	0.4032
Item5	5.0350	0.2837
Item6	3.1650	0.5306
Item7	7.4321	0.1147
Item8	8.2183	0.0839 .
Item9	10.6927	0.0302 *
Item10	25.1297	0.0000 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Detection threshold: 9.4877 (significance level: 0.05)

Items detected as DIF items:

```

Item2
Item9
Item10

```

Output was not captured!

```

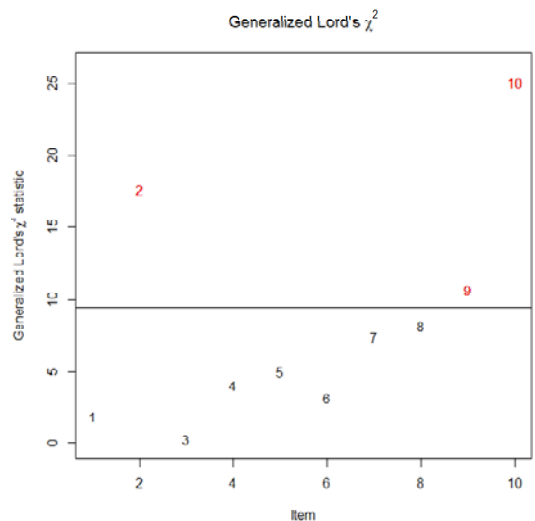
>
>

```

```

> # plot of chi square statistics
> plot(rEth)
The plot was not captured!
>

```



```

> #M-H , p values, and DIF items
> (rMHEth <- difGMH(dEth, group="Group", focal.name=c("Sittway", "Minhla")))

```

Detection of Differential Item Functioning using Generalized Mantel-Haenszel method, without item purification and with 2 focal groups

Focal groups:

```

Sittway
Minhla

```

Generalized Mantel-Haenszel chi-square statistic:

	Stat.	P-value
Item1	3.3104	0.1911
Item2	3.5828	0.1667
Item3	0.0710	0.9651
Item4	1.3160	0.5179
Item5	1.7659	0.4136
Item6	7.8493	0.0197 *
Item7	4.1496	0.1256

```
Item8 3.3832 0.1842
Item9 3.3504 0.1873
Item10 0.2418 0.8861
```

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Detection threshold: 5.9915 (significance level: 0.05)
```

```
Items detected as DIF items:
```

```
Item6
```

```
Output was not captured!
```

```
>
>
```

```
> #Logistic, p values, and DIF items
```

```
> (rLogsEth <- difGenLogistic(dEth, group="Group", focal.name=c("Sittway", "Minhla")))
```

```
Detection of both types of Differential Item Functioning
using Generalized logistic regression method,
without item purification and with 2 focal groups
```

```
Focal groups:
```

```
Sittway
Minhla
```

```
DIF flagging criterion: Likelihood ratio test
```

```
Generalized Logistic regression statistic:
```

	Stat.	P-value
Item1	4.3921	0.3555
Item2	6.3406	0.1751
Item3	0.0854	0.9991
Item4	6.3593	0.1739
Item5	14.9842	0.0047 **
Item6	9.3672	0.0525 .
Item7	12.6210	0.0133 *
Item8	10.5531	0.0321 *
Item9	11.9417	0.0178 *
Item10	3.1398	0.5347

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Detection threshold: 9.4877 (significance level: 0.05)
```

```
Items detected as DIF items:
```

```
Item5
Item7
Item8
Item9
```

```
Effect size (Nagelkerke's R^2):
```

```
Effect size code:
```

```
'A': negligible effect
'B': moderate effect
'C': large effect
```

	R^2	ZT	JG
Item1	0.0247	A	A
Item2	0.0357	A	B
Item3	0.0006	A	A
Item4	0.0357	A	B
Item5	0.0492	A	B

```
Item6 0.0542 A B
Item7 0.0459 A B
Item8 0.0389 A B
Item9 0.0505 A B
Item10 0.0176 A A
```

Effect size codes:

```
Zumbo & Thomas (ZT): 0 'A' 0.13 'B' 0.26 'C' 1
Jodoign & Gierl (JG): 0 'A' 0.035 'B' 0.07 'C' 1
```

Output was not captured!

```
>
>
>
```

```
> # compare three results
> (gEth <- genDichoDif(dEth, group="Group", focal.names=c("Sittway", "Minhla"),
+ method=c("GMH", "genLogistic", "genLord"),
+ nrFocal=2, model="2PL"))
```

Comparison of DIF detection among multiple groups, using 3 methods

Generalized methods used: Mantel-Haenszel, Logistic regression,
Lord's chi-square test

Parameters:

```
Significance level: 0.05
DIF effects tested by generalized logistic regression: both effects
DIF flagging criterion: Likelihood ratio test
Item response model: 2PL
Engine 'ltm' for item parameter estimation
Item purification: No
```

Comparison of DIF detection results:

	M.-H.	Logistic	Lord	#DIF
Item1	NoDIF	NoDIF	NoDIF	0/3
Item2	NoDIF	NoDIF	DIF	1/3
Item3	NoDIF	NoDIF	NoDIF	0/3
Item4	NoDIF	NoDIF	NoDIF	0/3
Item5	NoDIF	DIF	NoDIF	1/3
Item6	DIF	NoDIF	NoDIF	1/3
Item7	NoDIF	DIF	NoDIF	1/3
Item8	NoDIF	DIF	NoDIF	1/3
Item9	NoDIF	DIF	DIF	2/3
Item10	NoDIF	NoDIF	DIF	1/3

Output was not captured!

```
>
>
>
> #all group values
> agnEth<-c("Myitkyinar", "Sittway", "Minhla")
>
>
>
```

> # discriminant and difficulty parameters after rescaling

```
> sIParEth <- rEth$itemParInit
> rownames(sIParEth) <-NULL
> items <- rep(rEth$names, length(agnEth))
> group <- rep(agnEth, each=length(rEth$names))
> (sIParEth <- data.frame(items, sIParEth, group))
```

	items	a	b	se. a.	se. b.	cov. a. b.	group
1	Item1	1.5380969	-1.83828193	0.4002920	0.4027601	0.1258846110	Myitkyinar
2	Item2	2.0400659	-2.27464886	0.6365087	0.8644785	0.4885943013	Myitkyinar
3	Item3	1.8825491	-2.24371338	0.5673737	0.7308262	0.3608187242	Myitkyinar
4	Item4	1.0259302	-2.07074036	0.2852980	0.2599655	0.0459145403	Myitkyinar
5	Item5	2.0394004	-1.03381906	0.4998997	0.3801215	0.1481786587	Myitkyinar

```

6 Item6 1.6522546 -2.14141890 0.4639729 0.5379622 0.2064284148 Myitkyinar
7 Item7 1.6141457 -0.90759031 0.3687412 0.2561227 0.0588601541 Myitkyinar
8 Item8 1.4403277 -0.45802015 0.3197056 0.1894547 0.0202640546 Myitkyinar
9 Item9 0.9098867 -0.10301649 0.2249735 0.1485777 0.0015230895 Myitkyinar
10 Item10 0.3859422 -2.49122292 0.1874362 0.1472450 0.0054810075 Myitkyinar
11 Item1 1.7485795 -2.07722535 0.6299268 0.8291413 0.3796993850 Sittway
12 Item2 1.7896214 -2.00405041 0.6358558 0.8220252 0.3747947839 Sittway
13 Item3 1.7366152 -2.36205236 0.6609115 0.9835012 0.5179701918 Sittway
14 Item4 0.5252554 -2.32556132 0.2589588 0.3479698 0.0214411582 Sittway
15 Item5 0.8503619 -1.66917263 0.3078722 0.3880381 0.0328646430 Sittway
16 Item6 1.5452392 -2.50287354 0.5916726 0.9006141 0.4159219367 Sittway
17 Item7 0.6821572 -0.86888040 0.2746271 0.3425511 0.0008528652 Sittway
18 Item8 0.7549751 -0.53868432 0.2879522 0.3563929 -0.0096446392 Sittway
19 Item9 1.4346514 -0.04610927 0.5215074 0.5863117 -0.1663181721 Sittway
20 Item10 1.0869772 -1.16786276 0.3611431 0.4084324 0.0219354347 Sittway
21 Item1 1.4715802 -2.48778654 0.4395723 0.6226570 0.2215434634 Minhla
22 Item2 4.0055900 -2.22321350 2.8993142 5.1271227 14.6986631053 Minhla
23 Item3 1.4974351 -2.69569898 0.4888017 0.7262490 0.2971297476 Minhla
24 Item4 1.2258118 -1.68406363 0.3235524 0.3334834 0.0641006553 Minhla
25 Item5 1.5558869 -1.16685236 0.4192775 0.3415130 0.0792228246 Minhla
26 Item6 1.0678651 -2.22144101 0.3130277 0.3579627 0.0715949813 Minhla
27 Item7 0.7816893 -0.74675965 0.2257378 0.2090731 0.0051960524 Minhla
28 Item8 0.6073449 -1.06547986 0.2036376 0.1998782 0.0057353174 Minhla
29 Item9 0.2157375 -0.02383844 0.1708070 0.1787988 -0.0002313110 Minhla
30 Item10 1.0188381 -1.24733840 0.2809263 0.2524992 0.0272445964 Minhla
>
>

```

```

> #draw ICCs
> par(mfrow=c(2,2))
> plot(rEth, plot="itemCurve", item=1, ref.name="Myitkyinar")
> plot(rEth, plot="itemCurve", item=2, ref.name="Myitkyinar")
> plot(rEth, plot="itemCurve", item=9, ref.name="Myitkyinar")
> plot(rEth, plot="itemCurve", item="Item10", ref.name="Myitkyinar")

```

The plot was not captured!

```

> par(mfrow=c(1,1))
>
>
>

```

