Part 4: Fitting the Rasch Model in R

The **eRm** Package

Mair & Hatzinger, 2010

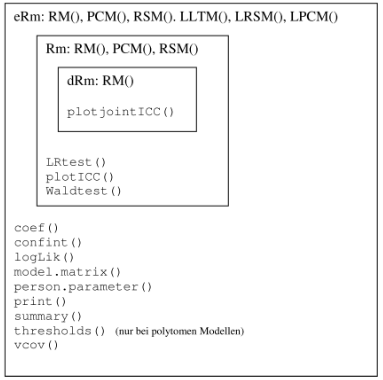
The R package eRm (extended Rasch modelling)

```
> library(eRm)
```

main functions concerning fit of the RM:

- `RM(data)` fits the RM and generates object of class `dRm`
- `person.parameter(drmobj)` generates object of class `ppar`
- plots from `dRm` object:
  - `plotPImap()`, `plotICC()`, `plotjointICC()`
- plots from `ppar` object:
  - `plot()`
- extract information from `dRm` object:
  - `coef()`, `vcov()`, `confint()`, `logLik()`, `model.matrix()`
- extract information from `ppar` object:
  - `confInt()`, `logLik()`

eRm Object Hierarchy



eRm: RM(), PCM(), RSM(). LLTM(), LRSM(), LPCM()

Rm: RM(), PCM(), RSM()

dRm: RM()

plotjointICC()

LRtest()
plotICC()
Waldtest()

coef()
confint()
logLik()
model.matrix()
person.parameter()
print()
summary()
thresholds() (nur bei polytomen Modellen)
vcov()

`plotjointICC()` only works for `dRm` (dichotomous RM) objects
`LRtest()` works for `dRm` and `Rm` objects, etc.

Fitting the RM

```
> rm.res <- RM(stress)
> rm.res
Results of RM estimation:

Call:  RM(X = stress)

Conditional log-likelihood: -202.1232
Number of iterations: 13
Number of parameters: 5

Item (Category) Difficulty Parameters (eta):
                 I2          I3        I4        I5          I6
Estimate -0.1101525 -0.06109055 0.2413530 0.6812941 -0.3995985
Std.Err   0.2027044  0.20317038 0.2077760 0.2203618  0.2014672
```

– default is: `RM(datamatrix, sum0 = TRUE, other options)`
– `sum0` defines constraints (for estimability):
`TRUE` . . . sum zero, `FALSE` . . . first item set to 0
– the output gives difficulty parameters

Constraints and the Design Matrix

```
> model.matrix(rm.res)
        eta 1 eta 2 eta 3 eta 4 eta 5
beta I1    -1    -1    -1    -1    -1
beta I2     1     0     0     0     0
beta I3     0     1     0     0     0
beta I4     0     0     1     0     0
beta I5     0     0     0     1     0
beta I6     0     0     0     0     1

> model.matrix(RM(stress, sum0 = FALSE))
        eta 1 eta 2 eta 3 eta 4 eta 5
beta I1     0     0     0     0     0
beta I2     1     0     0     0     0
beta I3     0     1     0     0     0
beta I4     0     0     1     0     0
beta I5     0     0     0     1     0
beta I6     0     0     0     0     1
```

```
> summary(rm.res)
Results of RM estimation:

Call:  RM(X = stress)

Conditional log-likelihood: -202.1232
Number of iterations: 13
Number of parameters: 5

Item (Category) Difficulty Parameters (eta) with 0.95 CI:
   Estimate Std. Error lower CI upper CI
I2   -0.110      0.203   -0.507    0.287
I3   -0.061      0.203   -0.459    0.337
I4    0.241      0.208   -0.166    0.649
I5    0.681      0.220    0.249    1.113
I6   -0.400      0.201   -0.794   -0.005

Item Easiness Parameters (beta) with 0.95 CI:
        Estimate Std. Error lower CI upper CI
beta I1    0.352      0.201   -0.043    0.747
beta I2    0.110      0.203   -0.287    0.507
beta I3    0.061      0.203   -0.337    0.459
beta I4   -0.241      0.208   -0.649    0.166
beta I5   -0.681      0.220   -1.113   -0.249
beta I6    0.400      0.201    0.005    0.794
```

Extracting Information

the item parameter estimates

```
> coef(rm.res)
    beta I1     beta I2     beta I3     beta I4     beta I5     beta I6
 0.35180555  0.11015250  0.06109055 -0.24135303 -0.68129407  0.39959849
```

the variance-covariance matrix of item parameter estimates

```
> vcov(rm.res)
             [,1]         [,2]         [,3]         [,4]         [,5]
[1,]  0.041089075 -0.007883430 -0.008375266 -0.009639669 -0.007580402
[2,] -0.007883430  0.041278204 -0.008399889 -0.009630899 -0.007669705
[3,] -0.008375266 -0.008399889  0.043170879 -0.009727786 -0.008336451
[4,] -0.009639669 -0.009630899 -0.009727786  0.048559311 -0.009799411
[5,] -0.007580402 -0.007669705 -0.008336451 -0.009799411  0.040589014
```

Extracting Information (cont'd)

confindence intervals for the item parameter estimates

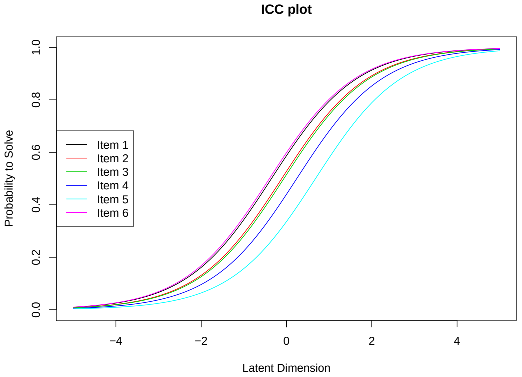
```
> confint(rm.res, "beta")
               2.5 %     97.5 %
beta I1 -0.043119485  0.7467306
beta I2 -0.287140826  0.5074458
beta I3 -0.337116079  0.4592972
beta I4 -0.648586564  0.1658805
beta I5 -1.113195206 -0.2493929
beta I6  0.004730127  0.7944669
```

the conditional log likelihood

```
> logLik(rm.res)
Conditional log Lik.: -202.1232 (df=5)
```

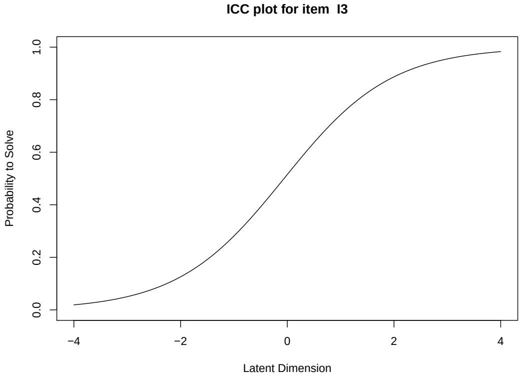
Plot ICCs

```
> plotjointICC(rm.res, xlim = c(-5, 5))
```



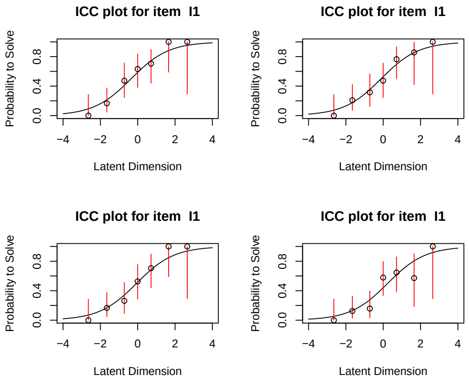
Plot single ICC

```
> plotICC(rm.res, item.subset = 3, ask = FALSE)
```



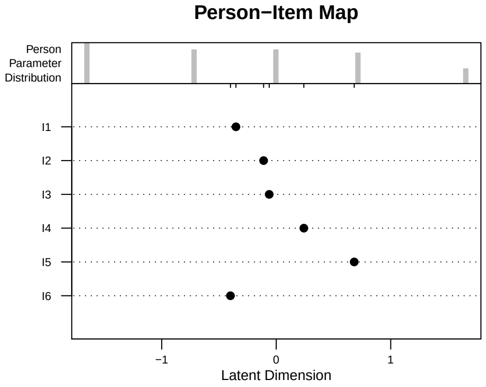
Plot ICCs

```
> plotICC(rm.res, item.subset = 1:4, ask = F, empICC = list("raw"),
+     empCI = list(lty = "solid"))
```



Plot Person-Item Map

```
> plotPImap(rm.res)
```



Person Parameter Estimation

```
> pp <- person.parameter(rm.res)
> pp
Person Parameters:

 Raw Score     Estimate Std.Error
         0 -2.644767050        NA
         1 -1.653469546 1.1039389
         2 -0.717574139 0.8777577
         3 -0.002996399 0.8301051
         4  0.713595517 0.8800761
         5  1.655491600 1.1076269
         6  2.653641042        NA
```

if NAs in the data, different person parameters are estimated for every NA-pattern group

Methods for Person Parameter Estimation Results

```
> logLik(pp)
Unconditional (joint) log Lik.: -17.37912 (df=5)
```

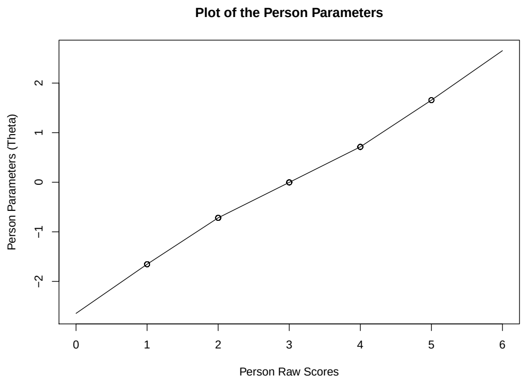
```
> confint(pp)

       2.5 %    97.5 %
P1 -1.629972 1.6239796
P2 -1.629972 1.6239796
P3 -3.817150 0.5102109
P4 -3.817150 0.5102109
P5 -1.011322 2.4385129
P6 -1.011322 2.4385129
...
```

attention: `confint(pp)` gives values for all subjects
if there are NAs in the data, confidence intervals are printed for each NA group

Plot of Person Parameter Estimates

```
> plot(pp)
```


Plot of the Person Parameters
Person Parameters (Theta)
Person Raw Scores