

# eRm: ein Open Source Paket für IRT Modelle

Reinhold Hatzinger

Department für Statistik und Mathematik  
Wirtschaftsuniversität Wien

## Was ist eRm?

- **eRm** steht für *extended Rasch modelling*
- ist ein **R** package
- ist open source: keine Lizenzgebühren, Quellcode verfügbar, Änderungen und Weitergabe erlaubt (GPL Lizenz)
- aktuell implementierte Modelle:  
LPCM, PCM, LRSM, RSM, LLTM, RM, (LLRA)
- verwendet CML Schätzmethode

## Was ist R?

eine general-purpose Umgebung zur Datenanalyse und Erzeugung von Grafiken.

- R ist open source Implementierung der Sprache **S**
- ist *lingua franca* im Bereich der computational statistics
- zunehmend Verbreitung nicht nur in der akademischen Welt: von großen Konzernen bis zur Verwendung in Schulen
- objektorientiert ('everything in S is an object')
- Struktur: `neues_objekt <- funktion(objekt, optionen)`
- User sind 'ProgrammiererInnen'

## Rasch Modell für (dichotome Items)

$$P(X_{vi} = 1 | \theta_v, \beta_i) = \frac{\exp(\theta_v - \beta_i)}{1 + \exp(\theta_v - \beta_i)}$$

$X_{vi}$  ... Person  $v$  löst Aufgabe  $i$

$\theta_v$  ... 'Fähigkeit' der Person  $v$

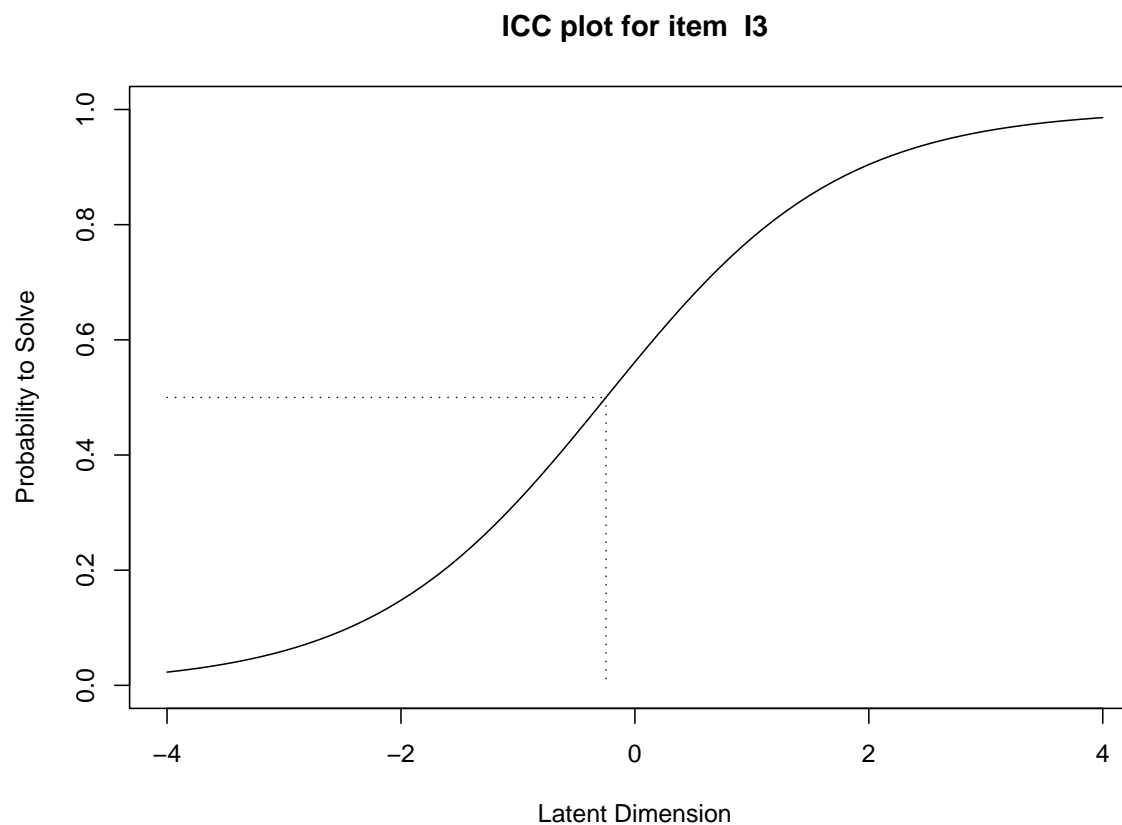
$\beta_i$  ... 'Schwierigkeit' des Items  $i$

	$I_1$	$I_2$	$I_3$	$I_4$	$r_v$
$P_1$	1	0	0	0	1
$P_2$	1	0	1	0	2
$P_3$	1	1	0	0	2
$P_4$	0	1	1	1	3
$s_i$	3	2	2	1	—

$$\sum_i x_{vi} = r_v$$

$$\sum_v x_{vi} = s_i$$

# Itemcharakteristik (ICC)



## Rasch Modell Eigenschaften

**Eindimensionalität**  $P(X_{vi} = 1 | \theta_v, \beta_i, \varphi) = P(X_{vi} = 1 | \theta_v, \beta_i)$

Lösungswahrscheinlichkeit hängt nicht von anderen Variablen ab

**Suffizienz**  $\sum_i x_{vi} = r_v$

$$f(x_{vi}, \dots, x_{vk} | \theta_v) = g(r_v | \theta_v) h(x_{vi}, \dots, x_{vk})$$

in Summe der Antworten ist gesamte Information über Fähigkeit, unabhängig welche Items gelöst wurden

**bedingte Unabhängigkeit**  $X_{vi} \perp X_{vj} | \theta_v$

bei fixem  $\theta$  keine Korrelation zwischen zwei Items

**Monotonizität**

mit zunehmender Fähigkeit steigt Lösungswahrscheinlichkeit

## Parameterschätzung im Rasch Modell

### Joint Maximum Likelihood (JML)

$$L_u = \frac{\exp(\sum_v \theta_v r_v) \exp(-\sum_i \beta_i s_i)}{\prod_v \prod_i (1 + \exp(\theta_v - \beta_i))}$$

### Conditional Maximum Likelihood (CML) (bedingen auf $r_v$ )

$$L_c = \exp(-\sum_i \beta_i s_i) / \prod_r \sum_{x|r} \exp(-\sum_i x_i \beta_i)^{n_r}$$

### Marginal Maximum Likelihood (MML) (Verteilung für $\theta$ )

$$L_m = \prod_r \left[ \exp(-\sum_i \beta_i s_i) \int \frac{\exp(\theta r)}{\prod_{i=1}^k (1 + \exp(\theta - \beta_i))} dG(\theta) \right]^{n_r}$$

## polytome Erweiterungen

### Partial Credit Modell (PCM)

$$P(X_{vi} = h) = \frac{\exp[h(\theta_v + \beta_i) + \omega_{hi}]}{\sum_{l=0}^{m_i} \exp[l(\theta_v + \beta_i) + \omega_{li}]}$$

$h$ ... Responsekategorien ( $h = 0, \dots, m_i$ )

$\omega_{hi}$ ... Kategorienparameter

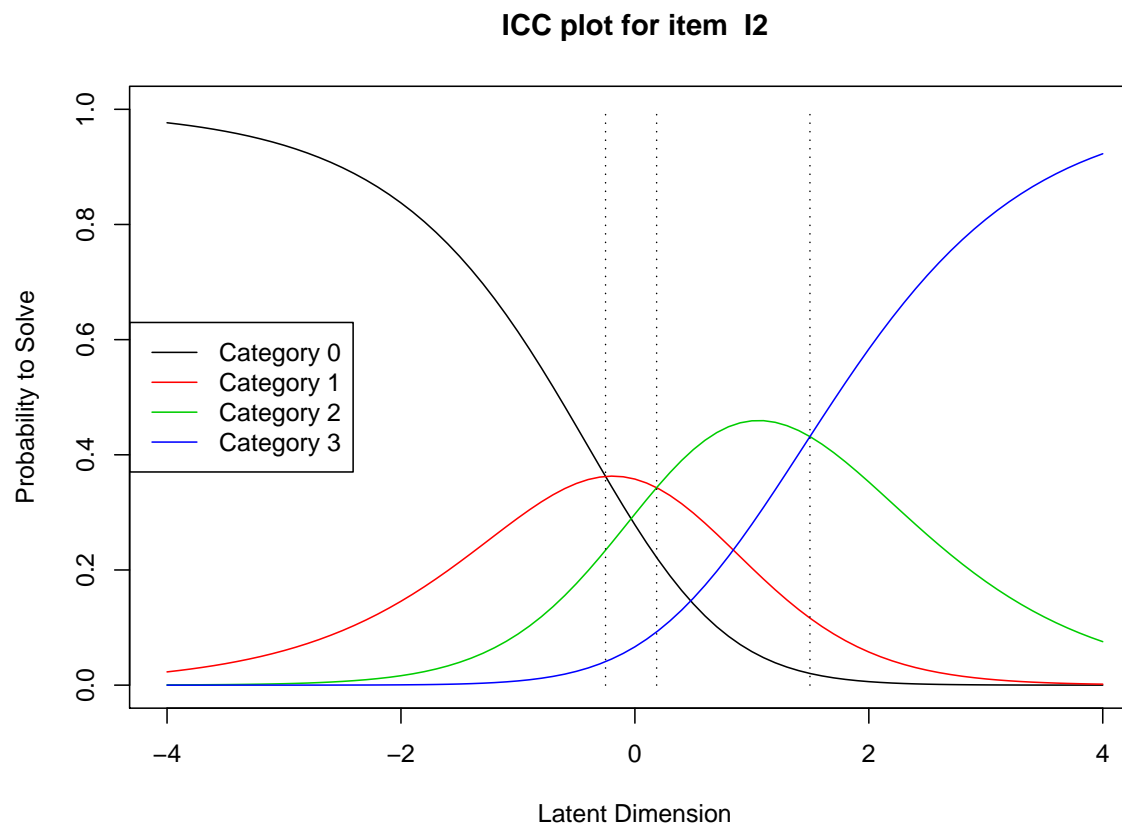
### Rating Scale Modell (RSM)

$m_i = m$ ... alle Items haben gleich viele Kategorien

$\omega_{hi} = \omega_h$ ... 'equistant scoring'



# ICC für ein polytomes Item



## eRm Basics

Kern ist das lineare partial credit model:

$$P(X_{vi} = h) = \frac{\exp(h\theta_v + \beta_{ih})}{\sum_{l=0}^{m_i} \exp(l\theta_v + \beta_{ih})}$$

wobei

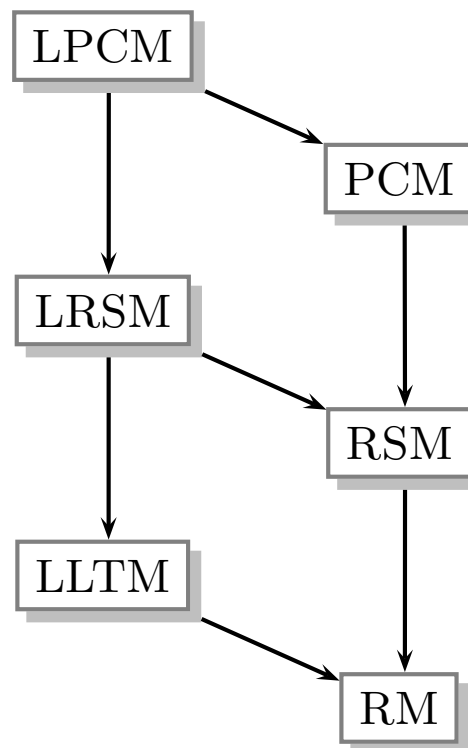
$$\beta_{ih} = \sum_p^m w_{ih,p} \eta_p$$

je nach Spezifikation der Designmatrix  $\mathbf{W} = (w_{ih,p})$  lassen sich verschiedene Modelle schätzen

aktuell implementierte Modelle:

LPCM, PCM, LRSM, RSM, LLTM, RM, (LLRA)

## Modellhierarchie



## Was kann eRm?

### Modelle:

- RM, RSM, PCM, LLTM, LRSM, LPCM, (LLRA)
- automatisierte Behandlung von missing values(MCAR)
- verschiedene Constraints
- Design Matrix (automatisiert/benutzerdefiniert)

### Parameterschätzung:

- Itemparameter, Basis- bzw. Effektparameter, Thresholdparameter mittels CML
- Personenparameter (JML)
- für alle: Varianz-Kovarianzmatrizen (Konfidenzintervalle)

## Was kann eRm?

### Diagnostik, Modelltests und Fitstatistiken:

- Andersen LR-test, Wald Test auf Item Basis
- Itemfit, Personfit (über standardisierte Residuen)
- Informationskriterien (AIC, BIC, cAIC)
- Prüfung auf Schätzbarkeit ('well-conditioned datamatrix') für RM

### Plots:

- Goodness-of-fit Plots
- ICC-Plots für einzelne Items (optional mit empirischen ICC's)
- gemeinsame Plots für RM (alle Items oder Subsets von Items in einem Plot)

## Rasch Modell

```
> rm.res <- RM(rdat)
```

```
> rm.res
```

```
Results of RM estimation:
```

```
Call: RM(X = rdat)
```

```
Conditional log-likelihood: -95.48076
```

```
Number of iterations: 8
```

```
Number of parameters: 3
```

```
Basic Parameters eta:
```

	eta 1	eta 2	eta 3
Estimate	1.0921364	-0.3970642	-1.0152016
Std.Err	0.2147963	0.2027998	0.2204647

## eRm: ein Open Source Paket für IRT Modelle

---

```
> summary(rm.res)
```

```
Results of RM estimation:
```

```
Call: RM(X = rdat)
```

```
Conditional log-likelihood: -95.48076
```

```
Number of iterations: 8
```

```
Number of parameters: 3
```

```
Basic Parameters (eta) with 0.95 CI:
```

	Estimate	Std. Error	lower CI	upper CI
eta 1	1.092	0.215	0.671	1.513
eta 2	-0.397	0.203	-0.795	0.000
eta 3	-1.015	0.220	-1.447	-0.583

```
Item Parameters (beta) with 0.95 CI:
```

	Estimate	Std. Error	lower CI	upper CI
beta I1	0.320	0.199	-0.070	0.710
beta I2	1.092	0.215	0.671	1.513
beta I3	-0.397	0.203	-0.795	0.000
beta I4	-1.015	0.220	-1.447	-0.583

## Modelltests:

```
> rm.lr <- LRtest(rm.res, splitcr = "mean")
```

```
> rm.lr
```

```
Andersen LR-test:
```

```
LR-value: 2.216
```

```
Chi-square df: 3
```

```
p-value: 0.529
```

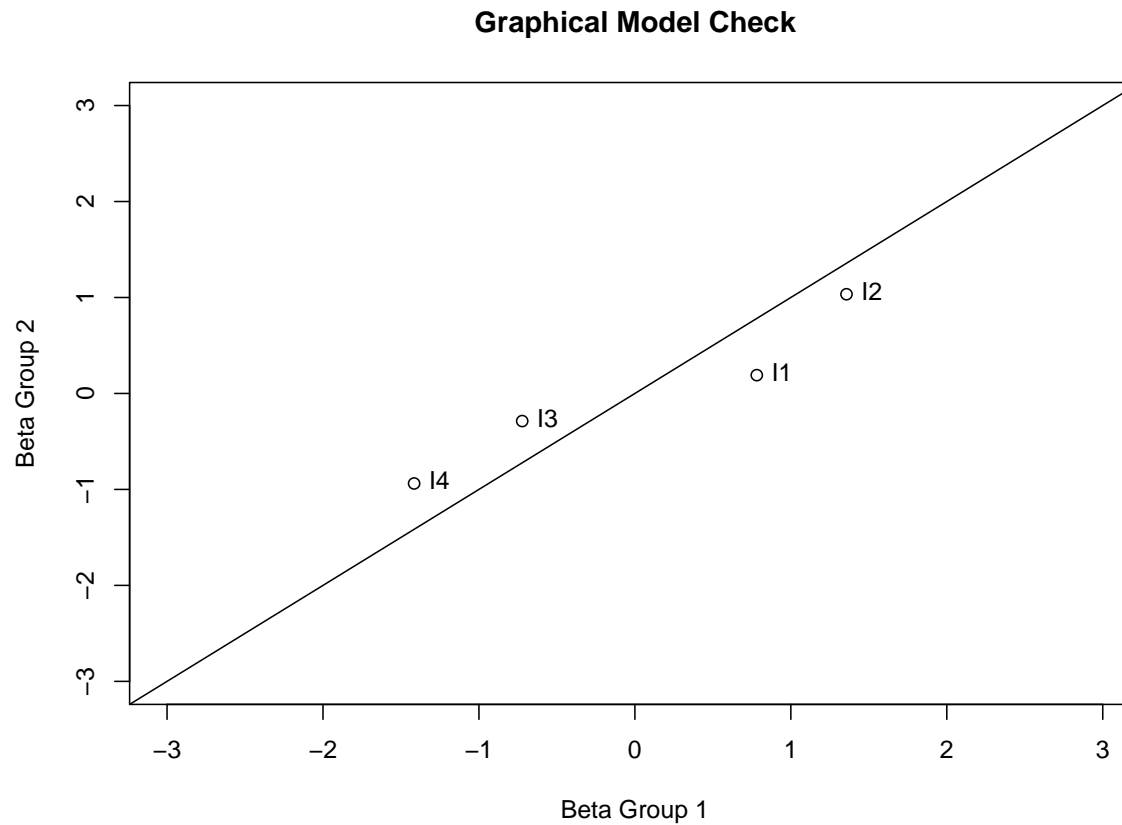
```
> Waldtest(rm.res)
```

```
Wald test on item level (z-values):
```

	z-statistic	p-value
beta I1	-0.533	0.594
beta I2	-0.280	0.780
beta I3	0.652	0.515
beta I4	0.511	0.610



```
> plotGOF(rm.lr)
```



## Rating Scale Modell

```
> rsm.res <- RSM(rsmdat)
```

```
> rsm.res
```

```
Results of RSM estimation:
```

```
Call: RSM(X = rsmdat)
```

```
Conditional log-likelihood: -107.5618
```

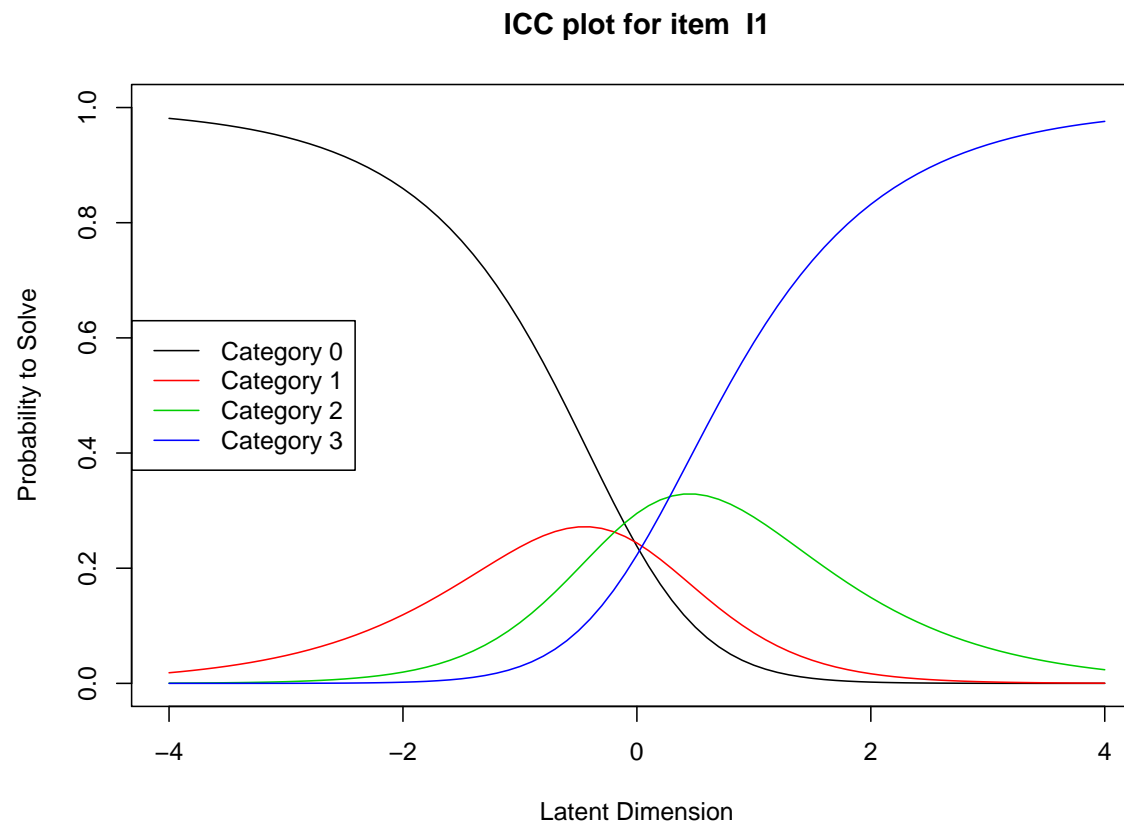
```
Number of iterations: 11
```

```
Number of parameters: 7
```

```
Basic Parameters eta:
```

	eta 1	eta 2	eta 3	eta 4	eta 5	eta 6
Estimate	-0.3051523	-0.2558638	-0.06730786	0.1601342	0.4442440	0.1673936
Std.Err	0.2053696	0.2028161	0.19650733	0.1965178	0.2084872	0.4596570
	eta 7					
Estimate	-0.1379066					
Std.Err	0.7324954					

```
> plotICC(rsm.res, item.subset = 1)
```



## Thresholds:

```
> thresholds(rsm.res)
```

```
Design Matrix Block 1:
```

	Location	Threshold 1	Threshold 2	Threshold 3
I1	0.02202	-0.02395	-0.19134	0.28135
I2	0.35112	0.30515	0.13776	0.61045
I3	0.30183	0.25586	0.08847	0.56116
I4	0.11328	0.06731	-0.10009	0.37261
I5	-0.11417	-0.16013	-0.32753	0.14517
I6	-0.39828	-0.44424	-0.61164	-0.13894

## Partial Credit Modell: Veränderungsmessung

- Setting über Designmatrix zu spezifizieren *oder*
- automatische Generierung von Standard-Designmatrizen
  - Zeitpunkte (`mpoints`)
  - Treatmentgruppen (`groupvec` - Spalte in Datenfile)
- Beispiel:  
3 Items (0-3) - 2 Zeitpunkte - Treatment-/Kontrollgruppe

Pseudodesign:

		Items	Trend	Treatment
KG	$t_1$	1	–	–
	$t_2$	1	1	–
TG	$t_1$	1	–	–
	$t_2$	1	1	1

```
> lpcm.res <- LPCM(lpcmdat, mpoints = 2, groupvec = G)
```

```
> lpcm.res
```

Results of LPCM estimation:

Call: LPCM(X = lpcmdat, mpoints = 2, groupvec = G)

Conditional log-likelihood: -103.7225

Number of iterations: 28

Number of parameters: 10

Basic Parameters eta:

	eta 1	eta 2	eta 3	eta 4	eta 5	eta 6
Estimate	0.4121568	-0.2989829	-0.1345081	0.03489371	-0.4288746	-0.2864050
Std.Err	0.3965853	0.4942288	0.4091779	0.39432809	0.4784496	0.4488165
	eta 7	eta 8	eta 9	eta 10		
Estimate	0.1641184	0.1007373	-0.2014872	1.0940382		
Std.Err	0.3992023	0.4351064	0.2608237	0.3870406		

## Weiterverarbeitung der Ergebnisse in R

### Varianzanalyse für Personenparameter nach 3 Altersgruppen:

```
> data(raschdat1)
> res <- RM(raschdat1)
> pres <- person.parameter(res)
> pp <- pres$thetapar$NAgroup1
> summary(aov(pp ~ age))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
age	2	94.236	47.118	163.74	< 2.2e-16
Residuals	97	27.914	0.288		

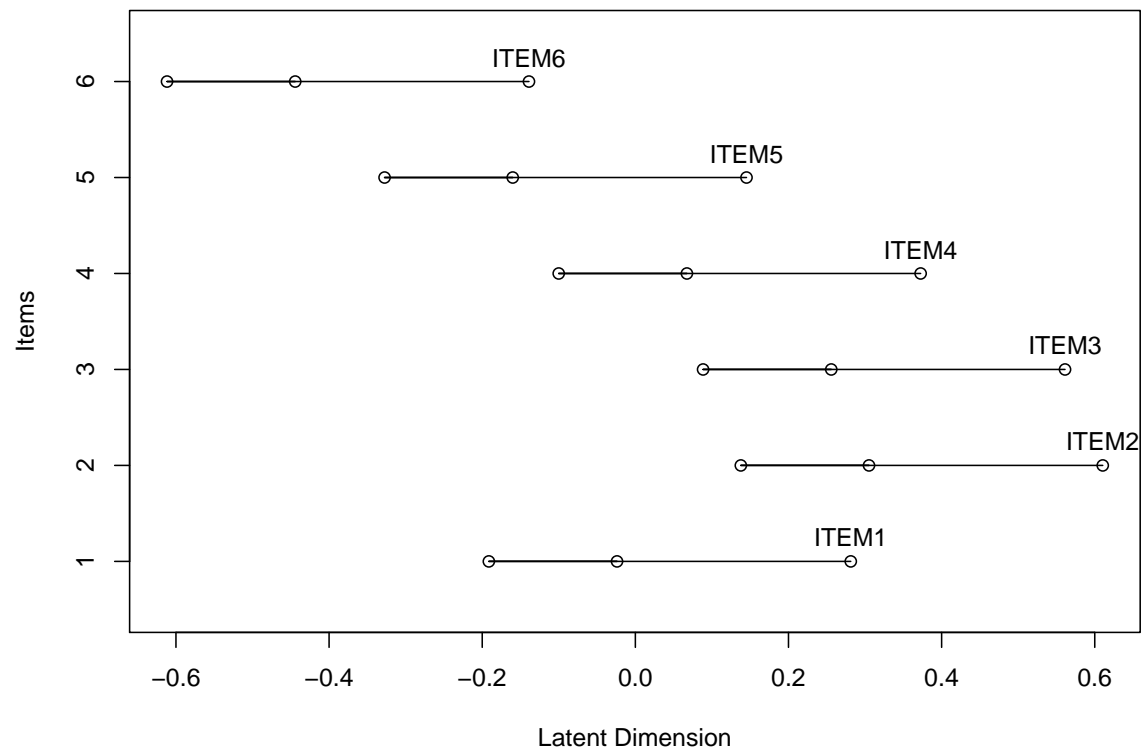
## Weiterverarbeitung der Ergebnisse in R

### Alternative Grafik für Thresholds:

```
> tp <- thresholds(rsm.res)$threshpar
> tpm <- matrix(tp, nc = 3, byrow = TRUE)
> plot(c(min(tpm), max(tpm)), c(0.5, 6.5), type = "n", xlab = "Latent Dimension",
+      ylab = "Items")
> for (i in 1:6) lines(tpm[i, ], rep(i, 3))
> for (i in 1:6) points(tpm[i, ], rep(i, 3))
> label <- paste("ITEM", 1:6, sep = "")
> text(tpm[, 3], 1:6, label, pos = 3)
```



# eRm: ein Open Source Paket für IRT Modelle



## Wiederverwenden der Ergebnisse

```
> rm.res <- RM(rdat)
> save(rm.res, "rm.res.Rdata")
```

später:

```
> load("rm.res")
```

Inhalt:

```
> names(rm.res)
 [1] "X"           "X01"         "model"       "loglik"      "npar"
 [6] "iter"        "convergence" "etapar"      "se.eta"     "hessian"
[11] "betapar"    "se.beta"    "W"           "call"
```

dann z.B.

```
> colSums(rm.res$X/nrow(rm.res$X))
  I1  I2  I3  I4
0.47 0.62 0.33 0.22
```

## Wie wird eRm-0.9-xx aussehen?

aktuelle Version: eRm-0.9-5

- 'exakte' Modelltests, Martin-Löf, Glas-Verhelst (für das RM)
- LLRA 'wrapper' (benutzerfreundliche Funktion)
- Mixed Rasch Modell
- OPLM (Verhelst & Glas)
- . . .

## Weitere Informationen zu R:

### R Projekt Homepage:

<http://www.r-project.org/>

### Comprehensive R Archive Network:

<http://cran.r-project.org/>

- R und Packages download
- Links zu Dokumentationen
- Diskussions- und Hilfe Foren
- Task Views – z.B. Psychometric Models and Methods

### R Wiki

<http://wiki.r-project.org/>

Dokumentationen, Foren, Tipps, Beispiele  
sehr hilfreich für Einsteiger!

## Weitere Informationen zu eRm:

### R Forge:

<http://r-forge.r-project.org/>

Entwicklungsplattform

download der jeweils neuesten Versionen

Diskussions- und Hilfe-Foren

Projekthomepage <http://erm.r-forge.r-project.org/>

### Manuals: im Entstehen bzw. in Planung

H. Poinstingl (Psychologische Diagnostik, Uni Wien)

R. Alexandrowics (Inst.f.Psychologie, Uni Klagenfurt)

### Publikationen:

Mair, P., and Hatzinger, R. (2007). J. Statistical Software

Mair, P., and Hatzinger, R. (2007). Psychology Science