IRT 4: Anwendung von Item Response Modellen

Anwendungsbeispiele

- In diesem Kapitel soll eine Reihe von psychologischen Tests mithilfe von Item Response Modellen skaliert werden.

- Anwendungsbeispiele:
  - Testmodelle für dichotome Items:
    - Skala *Leistungsmotivation* aus dem „Freiburger Persönlichkeitsinventar“ (FPI-R)
    - Matrizentest aus dem „Intelligenz Struktur Test“ (I-S-T 2000 R)
  - Testmodelle für ordinale Items:
    - Skala *Ordentlichkeit* aus dem „NEO – Persönlichkeitsinventar“ (NEO-PI-R)
    - Skala *Überblick* aus dem „Fragebogen Räumliche Strategien“ (FRS)

- Für alle vorgestellten Analysen wird das R – Paket „ltm“ verwendet (ltm: latent trait models).

Ablauf

- Schätzung der Itemparameter der für den Itemtyp angemessenen Modelle (1PL und 2PL oder PCM und GPCM) mithilfe der MML – Methode.
- Absoluter Modelltest:
  - Pearson-χ^2-Test (mit und ohne Bootstrap)
- Modellvergleiche:
  - Likelihood – Quotienten – Test (LRT)
  - AIC und BIC
- Interpretation der Itemparameter für das am besten passende Modell:
  - Parameterschätzungen
  - Geschätzte ICCs und CCCs
- Schätzung der Werte der Personen auf der latenten Variable mithilfe der EAP – Methode:
  - Konfidenzintervalle für θ_p

1. Beispiel: Leistungsmotivation

- Skala Leistungsmotivation aus dem „Freiburger Persönlichkeitsinventar“ (FPI-R)
  - 12 Items (z.B.: „Ich bin leicht beim Ehrgeiz zu packen“)
  - Dichotomes Itemformat („stimmt nicht“, „stimmt“)
  - Stichprobe bestehend aus 459 Studierende der Psychologie an der LMU
    - Funktion „gpcm“ kann auch für dichotome Items verwendet werden.
    - Restriktion der Diskriminationsparameter um das 1PL Modell zu erhalten (constraint = “1PL“).
- Schätzung der Itemparameter:
  - 1PL Modell:

```
> one_pl <- gpcm(data = Leistung, constraint = "1PL")
```

  - 2PL Modell:

```
> two_pl <- gpcm(data = Leistung)
```

1. Leistungsmotivation: Absoluter Modelltest 1PL


```
> GoF.gpcm(one_pl, simulate.p.value = FALSE)

Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Leistung, constraint = "1PL")

Tobs: 3912.89
df: 4082
p-value: 0.971
```

Sowohl für die Variante mit der theoretischen Prüfverteilung (oben) als auch für die Variante mit Bootstrap (unten) wird die Nullhypothese, dass das 1PL Modell in der Population gilt, beibehalten.

```
> GoF.gpcm(one_pl, simulate.p.value = TRUE, B = 200)

Parametric Bootstrap Approximation to Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Leistung, constraint = "1PL")

Tobs: 3912.89
# data-sets: 201
p-value: 0.423
```

Bootstrap = Zufallsziehungen
→ Jedes Mal anderes Ergebnis, außer man setzt einen seed.

1. Leistungsmotivation: Absoluter Modelltest 2PL Modell


```
> GoF.gpcm(two_pl, simulate.p.value = FALSE)

Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Leistung)

Tobs: 3993.58
df: 4071
p-value: 0.804

> GoF.gpcm(two_pl, simulate.p.value = TRUE, B = 200)

Parametric Bootstrap Approximation to Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Leistung)

Tobs: 3993.58
# data-sets: 201
p-value: 0.453
```

Sowohl für die Variante mit der theoretischen Prüfverteilung (oben) als auch für die Variante mit Bootstrap (unten) wird die Nullhypothese, dass das 2PL Modell in der Population gilt, beibehalten.

1. Leistungsmotivation: Modellvergleiche


```
> anova(one_pl, two_pl)

 Likelihood Ratio Table

          AIC     BIC  log.Lik   LRT df p.value
one_pl 6209.59 6263.27 -3091.80        13
two_pl 6187.43 6286.53 -3069.72 44.16 24  <0.001
```

Hinweis:
Im Folgenden werden trotzdem die Parameterschätzungen des sparsameren 1PL Modells weiter betrachtet

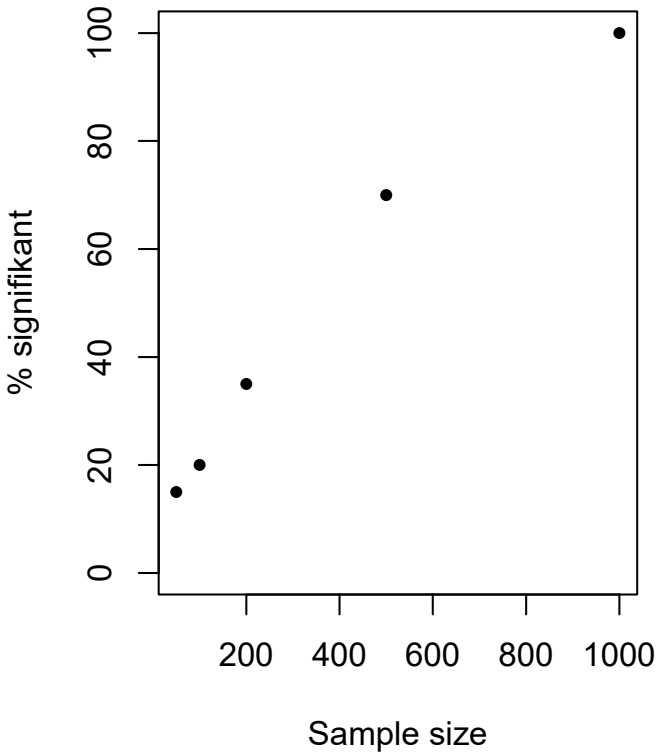
- Likelihood-Quotienten-Test:
  Die Nullhypothese, dass das einfachere (1PL) Modell in der Population genauso gut passt wie das komplexere (2PL) Modell, wird abgelehnt.

- Informationskriterien:
  - AIC: 2PL Modell „besser“ als 1PL Modell
  - BIC: 1PL Modell „besser“ als 2PL Modell

Überlegungen zu den Modelltests

- Power beim absoluten Modelltest:
  - H_0 = Das Modell gilt in der Population; H_1 = das Modell gilt nicht. Die Annahme der H_0 ist typischerweise das „Wunschergebnis“
  - Power = $P(H_0 \text{ ablehnen} \mid H_1) = P(p < \alpha \mid H_1)$
  - Geringe Stichprobengröße = niedrige Power; d.h., man behält öfter fälschlicherweise die H_0 bei
  - Kleine Stichprobe → man trifft oft (fälschlicherweise) die Entscheidung „Modell passt“
  - Sehr große Stichprobe: Selbst kleinste Abweichungen vom Idealmodell (≙ kleine Effektstärke) werden signifikant und führen zum Schluss „Modell passt nicht“
- LRT, AIC, BIC; Pearson-χ^2 mit oder ohne Bootstrap, …: Welchen nimmt man nun?
  - Researcher Degrees of Freedom und damit potentielles „*p*-hacking“: Man nimmt das Ergebnis, das einem am besten passt. Man findet dann schon eine passende Literaturstelle, mit der man post-hoc die Wahl des „besten“ Modelltests begründen kann.
  - Präregistrierung: Vorher begründet festlegen, ob man die Entscheidung auf LRT, AIC, oder BIC stützt, und ob man den absoluten Modelltest mit oder ohne Bootstrap rechnet (bzw. wie man diese Indizes gemeinsam in einer Entscheidung verrechnet)

Verletztes Raschmodell (Test sollte signifikant sein)



Hinweis: Die hier dargestellte Power bezieht sich nur auf die hier simulierte Modellverletzung. Diese Poweranalyse kann also nicht verallgemeinert werden.

1. Leistungsmotivation: Interpretation Itemparameter 1PL Modell

```
> summary(one_pl)

Call:
gpcm(data = Leistung, constraint = "1PL")

Model Summary:
   log.Lik      AIC      BIC
 -3091.797 6209.593 6263.271

Coefficients:
$FPI1_1
         value std.err z.value
Catgr.1 -0.585   0.123  -4.772
Dscrmn   0.955   0.055  17.406

$FPI2_10
         value std.err z.value
Catgr.1 -0.585   0.123  -4.772
Dscrmn   0.955   0.055  17.406

$FPI3_64
         value std.err z.value
Catgr.1 -1.619   0.157 -10.290
Dscrmn   0.955   0.055  17.406
```

...

Items 4 – 10 fehlen aus Platzgründen

...

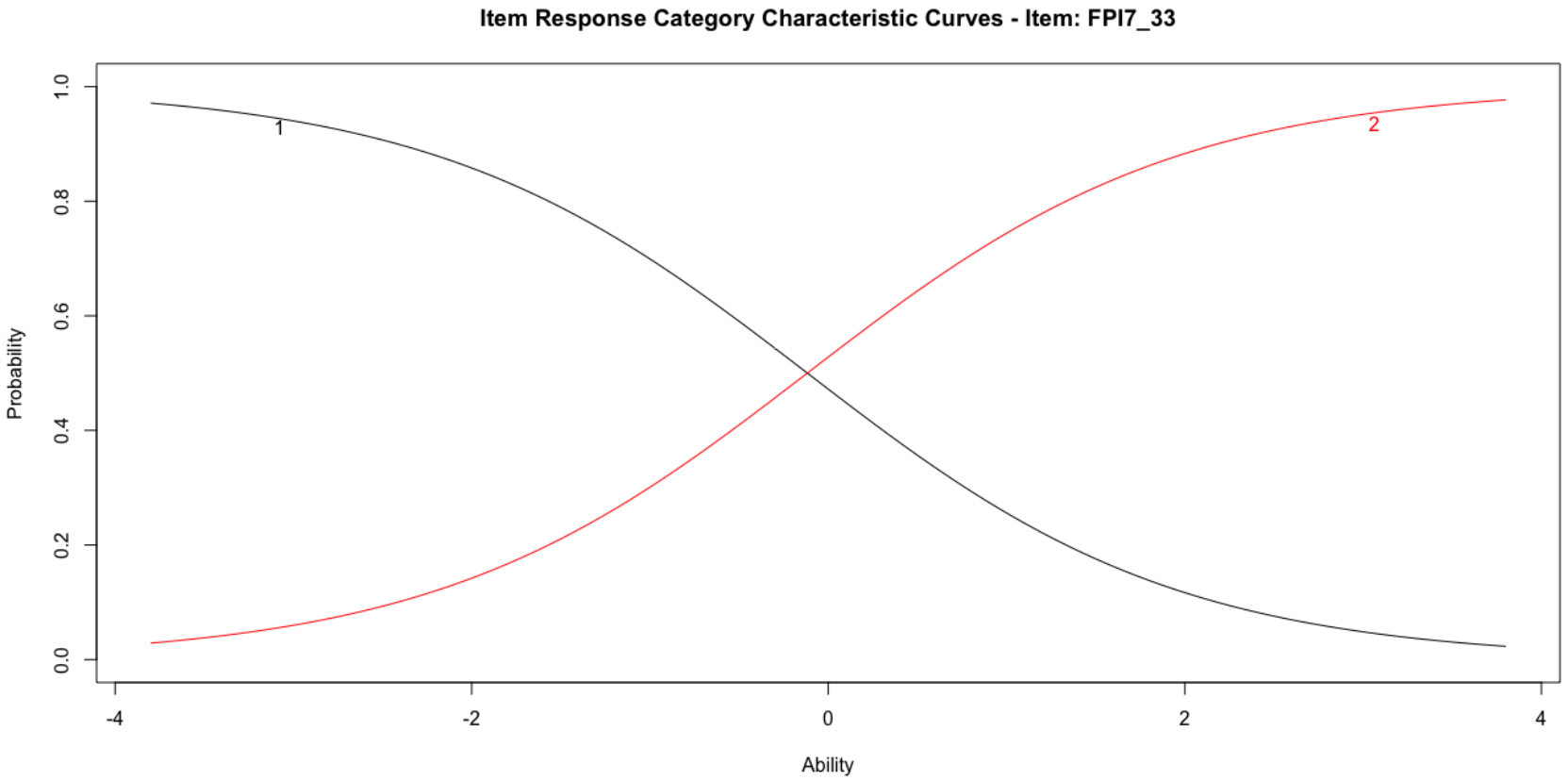
```
$FPI11_27
         value std.err z.value
Catgr.1 0.530   0.121   4.387
Dscrmn  0.955   0.055  17.406

$FPI12_63
          value std.err z.value
Catgr.1 -0.150   0.117  -1.281
Dscrmn   0.955   0.055  17.406


Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.016
optimizer: nlminb
```

1. Leistungsmotivation: Interpretation Itemparameter 1PL Modell

- Hinweis:
  - Das „ltm“ Paket verwendet eine spezielle Parametrisierung des 1-PL Modells, bei der statt der Varianz der latenten Variable ein für alle Items als konstant angenommener Diskriminationsparameter geschätzt wird (der ≠ 1 sein kann).
  - Die Varianz der latenten Variable muss in dieser Parametrisierung auf 1 gesetzt werden. Dies bedeutet, dass im Vergleich zu unserer Parametrisierung eine andere Einheit für die latente Variable gewählt wird.
  - Durch diesen Einheitswechsel ändern sich auch auch die Werte der Schwierigkeitsparameter und ihrer Schätzwerte. Die Interpretation der Schwierigkeitsparameter ändert sich jedoch nicht.
  - Man kann die Schätzwerte für die Parameter der ltm-Parametrisierung in Schätzwerte für die Parameter unserer Parametrisierung umrechnen. Dies werden wir jedoch im Rahmen dieser Vorlesung nicht besprechen.

1. Leistungsmotivation: Geschätzte ICC von Item 7


```
> plot(one_pl, items = 7)
```




```
> one_pl$coefficients[[7]][1]
   Catgr.1
-0.1173273
```

```
> table(Leistung[, 7])

  0   1
218 241
```

```
> thetas <- factor.scores(one_pl, method = "EAP")
```

- Exp: Erwartete Häufigkeit
- z1: EAP Schätzer
- se.z1: Standardfehler

```
> thetas

Call:
gpcm(data = Leistung, constraint = "1PL")

Scoring Method: Expected A Posteriori

Factor-Scores for observed response patterns:
```

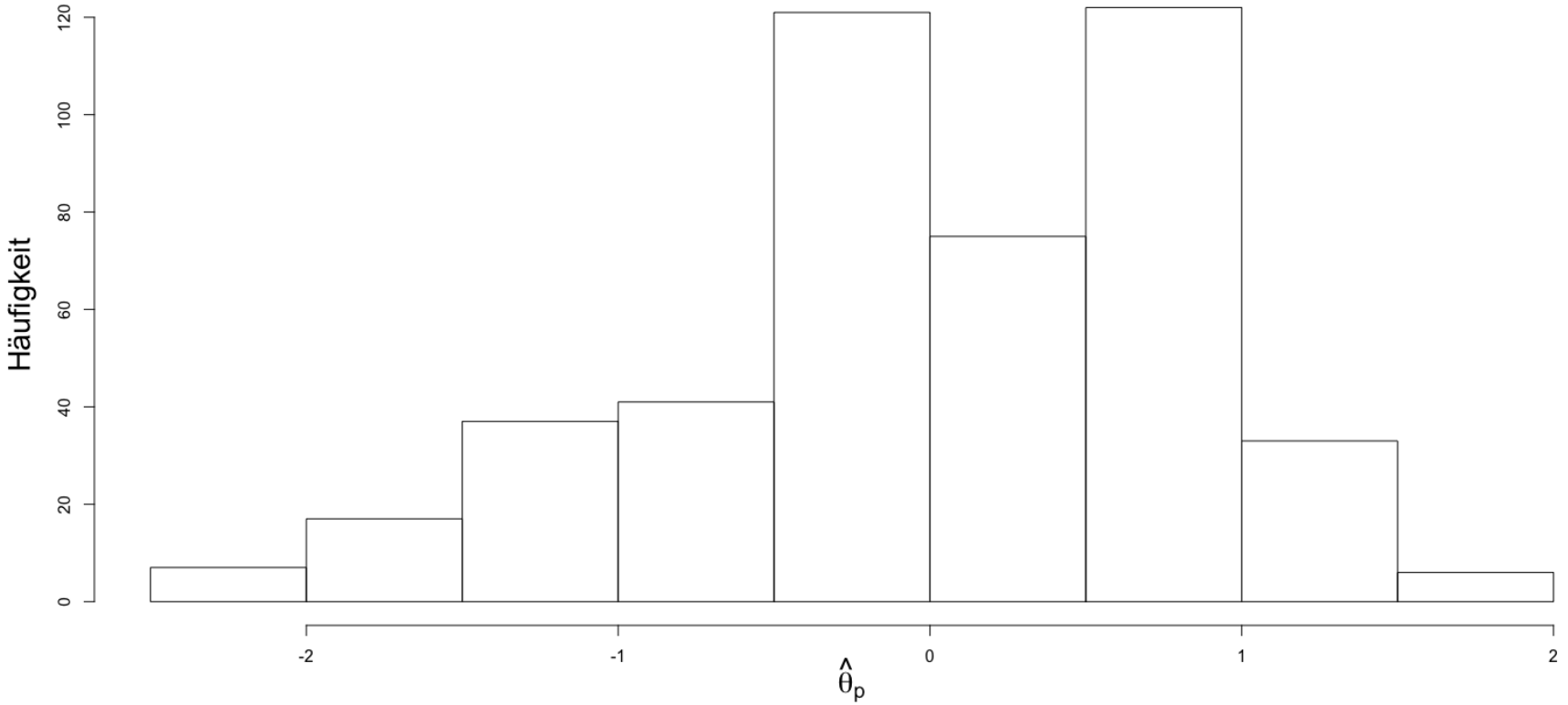
	FPI1_1	FPI2_10	FPI3_64	FPI4_74	FPI5_6	FPI6_43	FPI7_33	FPI8_77	FPI9_52	FPI10_49	FPI11_27	FPI12_63	Obs	Exp	z1	se.z1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.406	-2.469	0.658
2	1	1	1	1	1	1	1	1	1	1	2	1	1	0.028	-2.075	0.628
3	1	1	1	1	1	1	1	2	1	2	1	1	1	0.113	-1.713	0.604
4	1	1	1	1	1	2	1	1	1	1	1	1	2	0.082	-2.075	0.628
5	1	1	1	1	2	1	1	1	1	1	1	1	2	2.273	-2.075	0.628
6	1	1	1	1	2	1	1	1	1	1	2	1	1	0.225	-1.713	0.604
7	1	1	1	1	2	1	1	1	1	2	1	2	1	0.486	-1.374	0.587
8	1	1	1	1	2	1	1	2	1	1	1	1	1	1.129	-1.713	0.604
9	1	1	1	1	2	1	1	2	1	1	1	2	1	0.299	-1.374	0.587
10	1	1	1	1	2	1	1	2	1	2	1	1	1	1.273	-1.374	0.587
11	1	1	1	1	2	1	2	1	1	1	1	1	2	0.417	-1.713	0.604
12	1	1	1	1	2	1	2	2	1	1	1	1	1	0.290	-1.374	0.587
13	1	1	1	1	2	1	2	2	1	2	1	1	1	0.447	-1.053	0.574
14	1	1	1	1	2	1	2	2	1	2	1	2	1	0.219	-0.742	0.568

95% KI für eine Person mit Antwortmuster 2:

$$\left[\hat{\theta} \pm 1{,}96 \cdot \widehat{SE}\left(\hat{\theta}\right)\right] = [-2{,}08 \pm 1{,}96 \cdot 0{,}63] = [-3{,}31; -0{,}85]$$

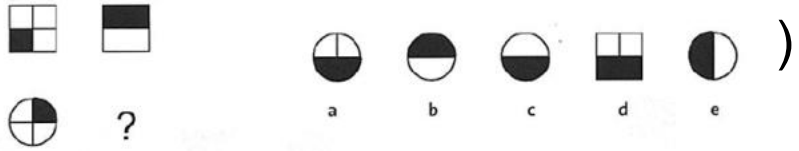
weitere beobachtete Antwortmuster fehlen aus Platzgründen

1. Leistungsmotivation: Verteilung der geschätzten Personenparameter



Würde es sich bei dem analysierten Datensatz um eine Normstichprobe handeln, so könnte die Verteilung der geschätzten Personenparameter zur Normierung verwendet werden (z.B.: Berechnung eines Prozentranges für eine neue beobachtete Person)

2. Beispiel: Matrizentest

- Matrizentest aus dem „Intelligenz Struktur Test“ (I-S-T 2000 R)
  - 20 Items (z.B. 

)
  - Dichotomes Itemformat („nicht gelöst“, „gelöst“)
  - Stichprobe bestehend aus 341 Personen

- Schätzung der Itemparameter:
  - 1PL Modell:

```
> one_pl <- gpcm(data = Matrizen, constraint = "1PL")
```

  - 2PL Modell:

```
> set.seed(2)
> two_pl <- gpcm(data = Matrizen, start.val = "random")
```

- Funktion „gpcm“ kann auch für dichotome Items verwendet werden.
- Restriktion der Diskriminationsparameter um das 1PL Modell zu erhalten (constraint = “1PL“).

Hinweis: Konvergenzprobleme des 2PL Modells für diesen Datensatz, daher Wahl eines „Seeds“ bei dem das Modell konvergiert.

2. Matrizentest: Absoluter Modelltest 1PL Modell


```
> GoF.gpcm(one_pl, simulate.p.value = FALSE)

Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Matrizen, constraint = "1PL")

Tobs: 1252894
df: 1048554
p-value: <0.001

> GoF.gpcm(one_pl, simulate.p.value = TRUE, B = 200)

Parametric Bootstrap Approximation to Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Matrizen, constraint = "1PL")

Tobs: 1252894
# data-sets: 201
p-value: 0.149
```

Für die Variante mit der theoretischen Prüfverteilung (oben) wird die Nullhypothese, dass das 1PL Modell in der Population gilt, abgelehnt.

Für die Variante mit Bootstrap (unten) wird die Nullhypothese nicht abgelehnt.

2. Matrizentest: Absoluter Modelltest 2PL Modell


```
> GoF.gpcm(two_pl, simulate.p.value = FALSE)

Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Matrizen, start.val = "random")

Tobs: 1348792
df: 1048535
p-value: <0.001
```

Für die Variante mit der theoretischen Prüfverteilung (oben) wird die Nullhypothese, dass das 2PL Modell in der Population gilt, abgelehnt.

Für die Variante mit Bootstrap (unten) wird die Nullhypothese nicht abgelehnt.

```
> GoF.gpcm(two_pl, simulate.p.value = TRUE, B = 200)

Parametric Bootstrap Approximation to Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Matrizen, start.val = "random")

Tobs: 1348792
# data-sets: 196
p-value: 0.102
```

2. Matrizentest: Modellvergleiche


```
> anova(one_pl, two_pl)

 Likelihood Ratio Table

           AIC     BIC  log.Lik   LRT df p.value
one_pl 7022.13 7102.60 -3490.07        21
two_pl 6984.48 7137.75 -3452.24 75.65 40  <0.001
```

- Likelihood-Quotienten-Test:
  Die Nullhypothese, dass das 1PL Modell in der Population genauso gut passt wie das 2PL Modell, wird abgelehnt.
- Informationskriterien:
  - AIC: 2PL Modell „besser“ als 1PL Modell
  - BIC: 1PL Modell „besser“ als 2PL Modell

Hinweis:
Im Folgenden werden die Parameterschätzungen des flexibleren 2PL Modells weiter betrachtet

2. Matrizentest: Interpretation Itemparameter 2PL Modell

Items 4 – 18 fehlen aus Platzgründen

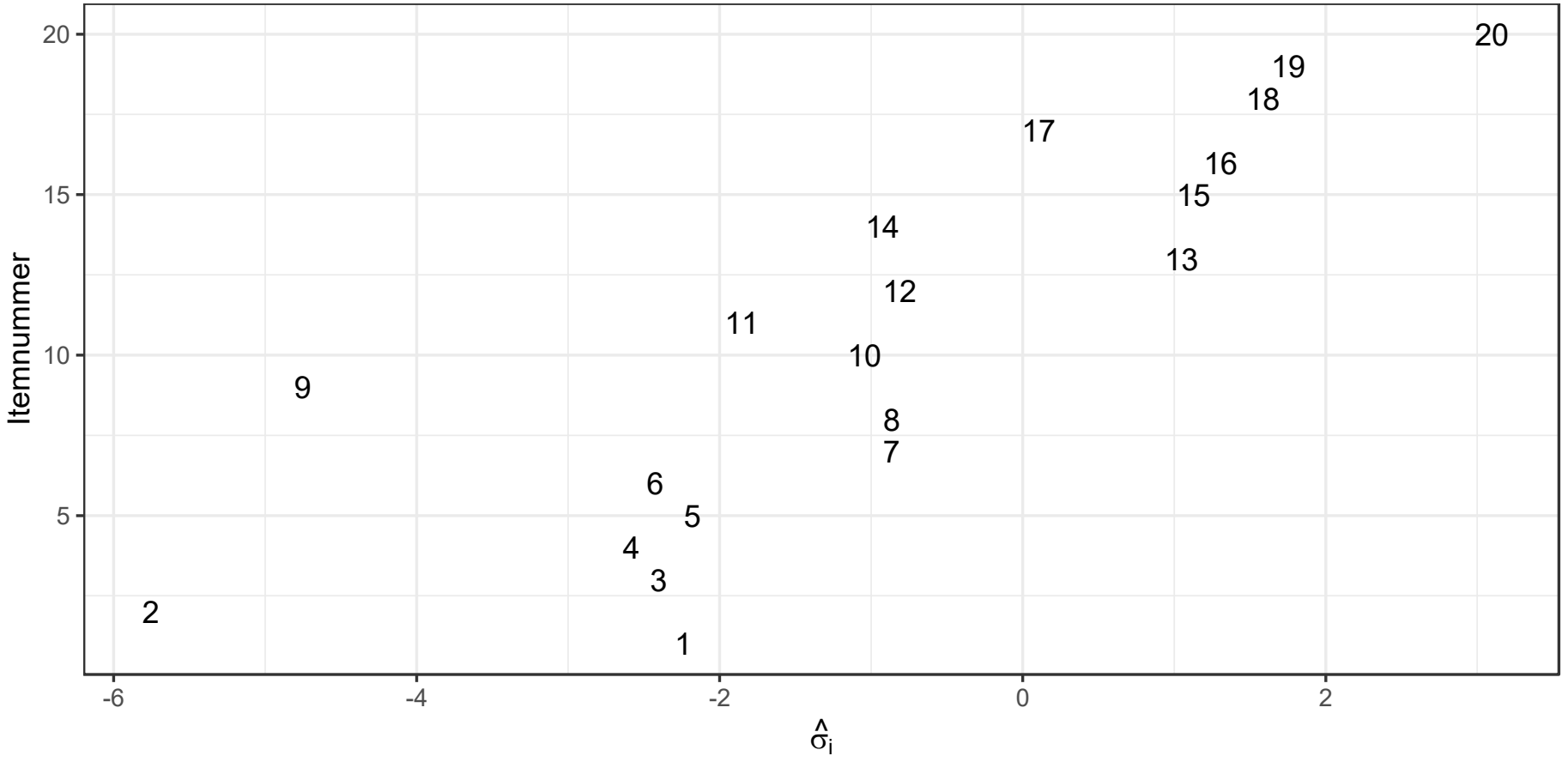
```
> summary(two_pl)

Call:
gpcm(data = Matrizen, start.val = "random")

Model Summary:
   log.Lik      AIC      BIC
 -3452.239 6984.479 7137.754

Coefficients:
$SISR1161
         value std.err z.value
Catgr.1 -2.241   0.291  -7.710
Dscrmn   3.234   1.399   2.311

$SISR1162
         value std.err z.value
Catgr.1 -5.754   3.193  -1.802
Dscrmn   0.330   0.189   1.748

$SISR1163
         value std.err z.value
Catgr.1 -2.404   0.513  -4.689
Dscrmn   0.868   0.218   3.987
```

...

...

```
$SISR1179
         value std.err z.value
Catgr.1 1.754   0.342   5.125
Dscrmn  1.098   0.276   3.973

$SISR1180
         value std.err z.value
Catgr.1 3.097   0.917   3.376
Dscrmn  0.857   0.304   2.822


Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.031
optimizer: nlminb
```

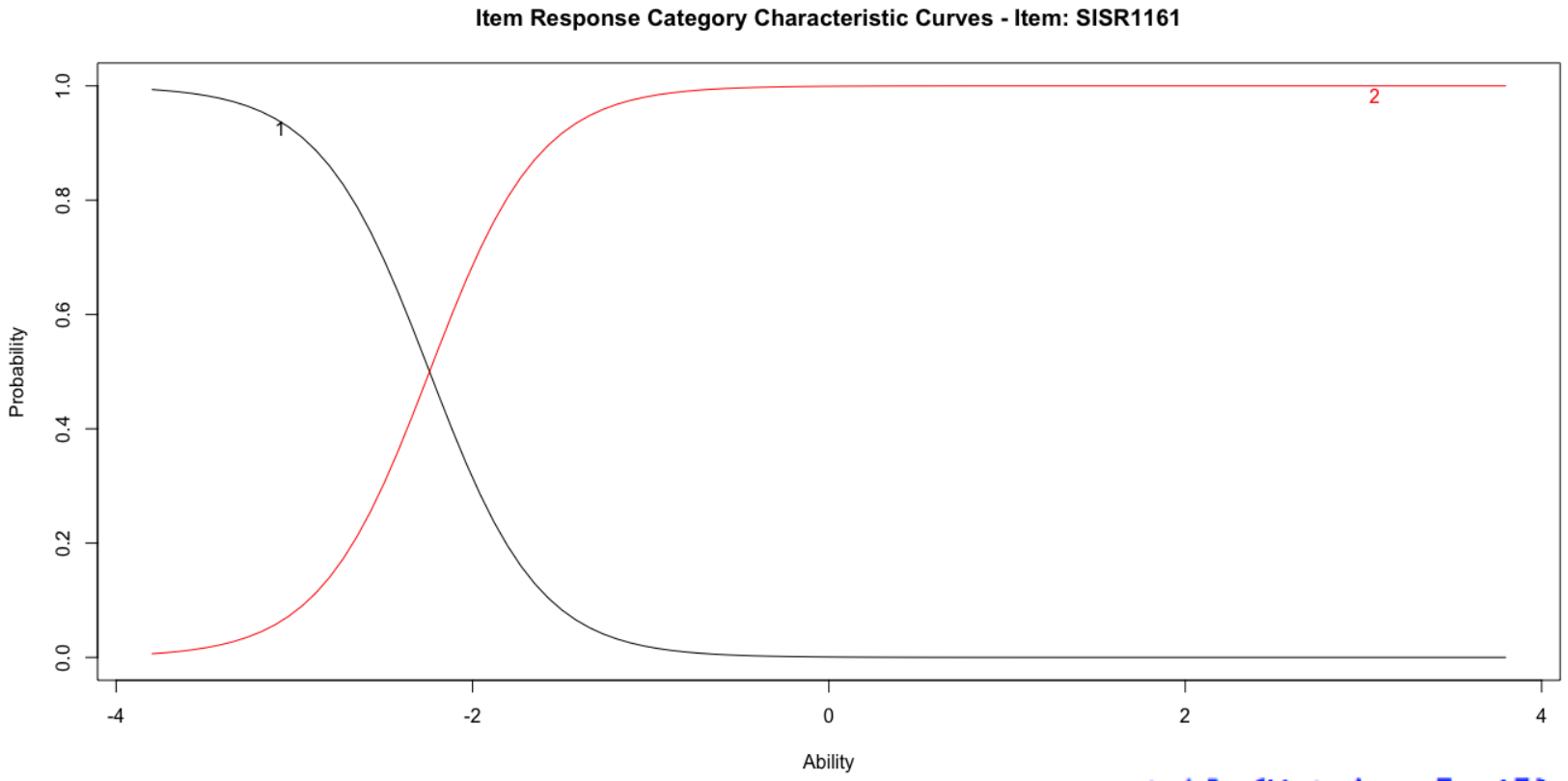
2. Matrizentest: Geschätzte Itemschwierigkeiten



Die Rangreihe der geschätzten Itemschwierigkeiten entspricht nicht perfekt der theoretischen Rangreihe (Matrizen sollten eigentlich in ihrer Schwierigkeit ansteigen)

2. Matrizentest: Geschätzte ICC von Item 1


```
> plot(two_pl, items = 1)
```

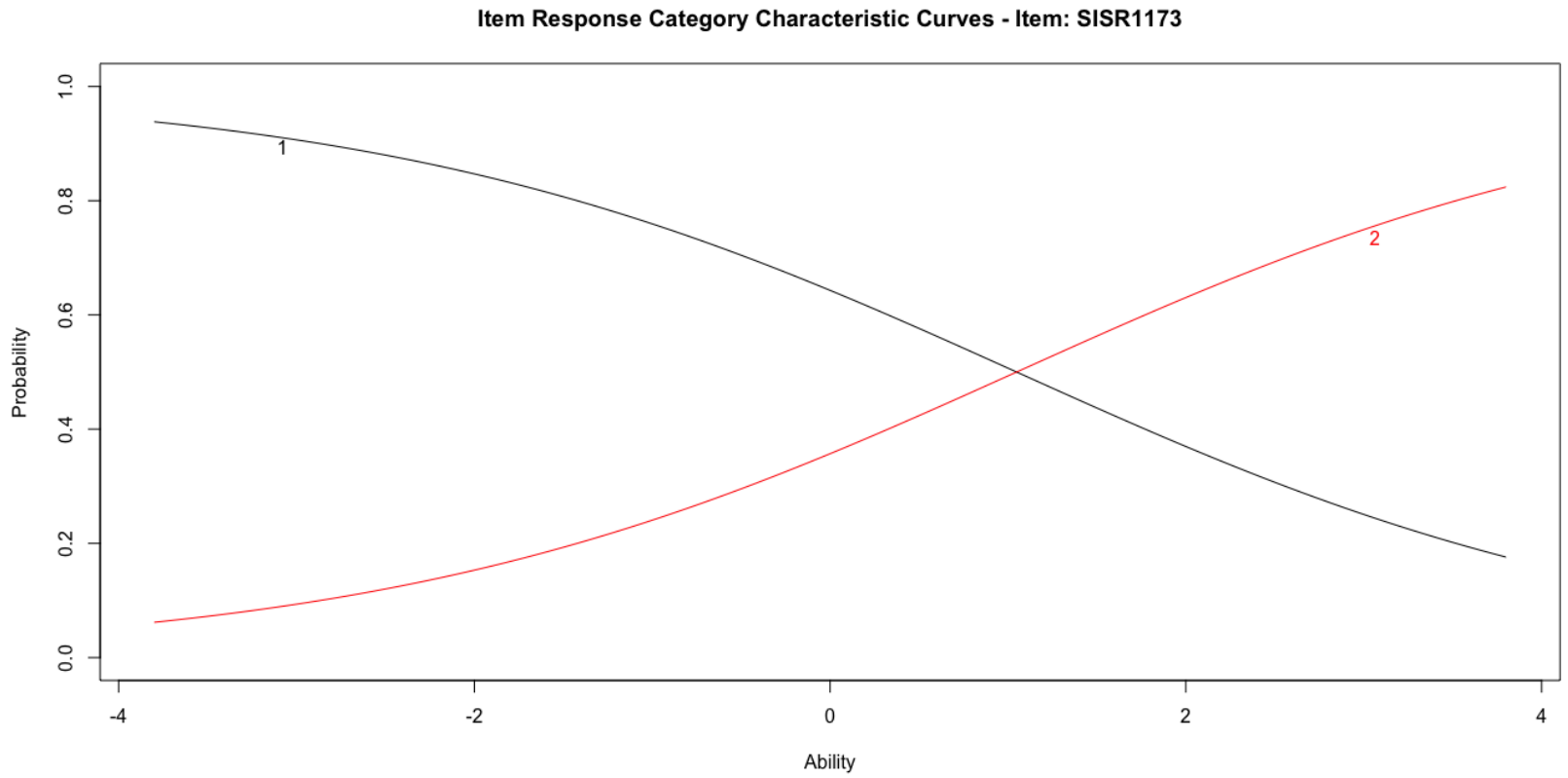



```
> two_pl$coefficients[[1]]
  Catgr.1    Dscrmn
-2.241109  3.233848
```

```
> table(Matrizen[, 1])

  0   1
 10 331
```

2. Matrizentest: Geschätzte ICC von Item 13


```
> plot(two_pl, items = 13)
```




```
> two_pl$coefficients[[13]]
  Catgr.1    Dscrmn
1.0484967 0.5611531
```

```
> table(Matrizen[, 13])

  0   1
216 125
```

2. Matrizentest: Schätzung der Personenparameter


```
> thetas <- factor.scores(two_pl, method = "EAP")
> thetas

Call:
gpcm(data = Matrizen, start.val = "random")

Scoring Method: Expected A Posteriori
```

- Exp: Erwartete Häufigkeit
- z1: EAP Schätzer
- se.z1: Standardfehler

95% KI für eine Person mit Antwortmuster 8:

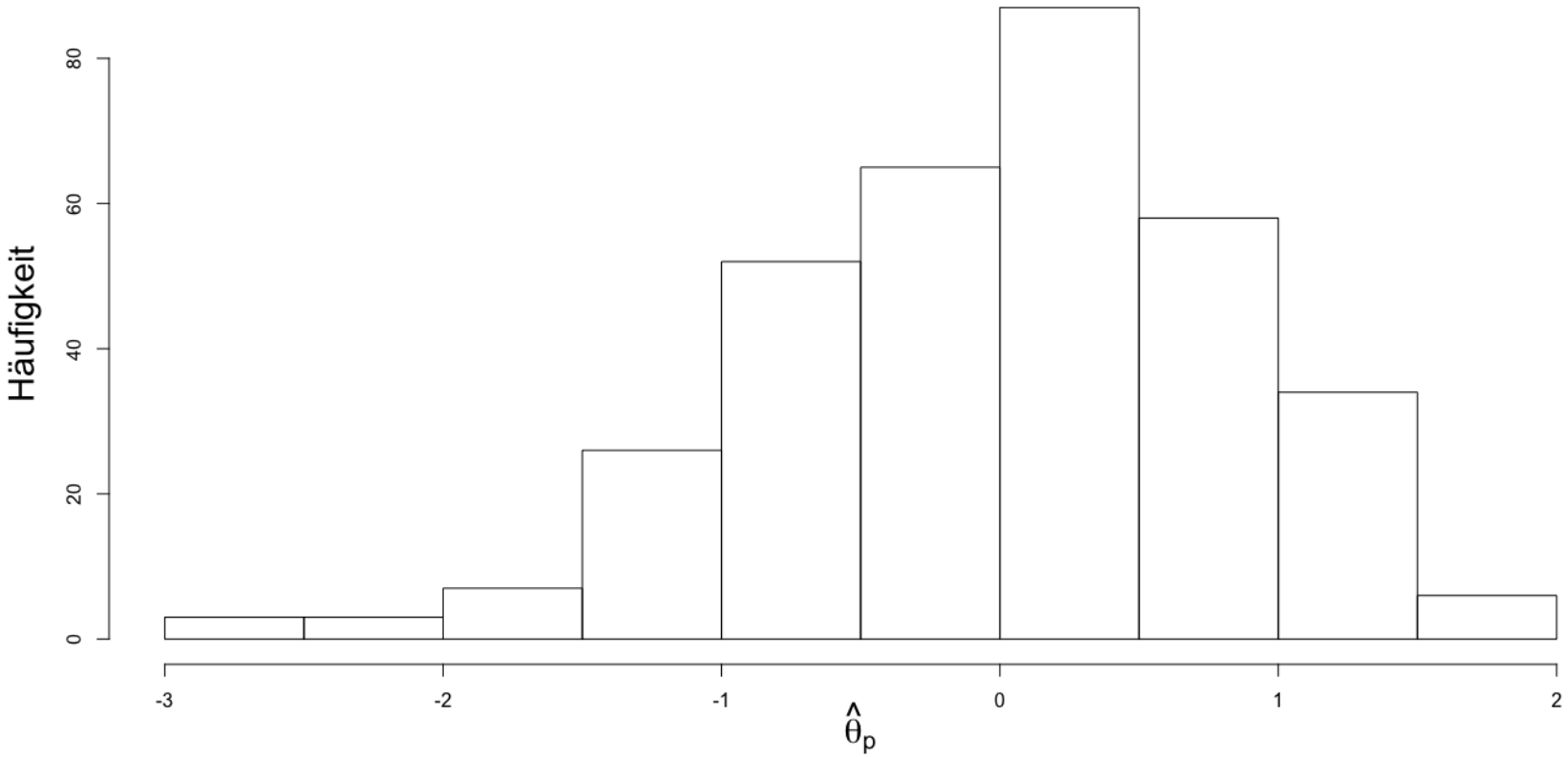
$$\left[\hat{\theta} \pm 1{,}96 \cdot \widehat{SE}\left(\hat{\theta}\right)\right] = [-1{,}64 \pm 1{,}96 \cdot 0{,}5] = [-2{,}62; -0{,}66]$$

Factor-Scores for observed response patterns:

	SISR1161	SISR1162	SISR1163	SISR1164	SISR1165	SISR1166		SISR1178	SISR1179	SISR1180	Obs	Exp	z1	se.z1
1	1	1	1	1	1	1		1	1	1	3	0.007	-2.983	0.572
2	1	1	2	1	2	1		1	1	1	1	0.003	-2.440	0.501
3	1	2	1	1	2	2		1	1	1	1	0.005	-2.213	0.484
4	1	2	1	2	2	2		1	1	1	1	0.006	-2.112	0.480
5	1	2	2	2	1	2		1	1	1	1	0.003	-1.766	0.491
6	1	2	2	2	1	2		1	1	1	1	0.001	-1.392	0.517
7	1	2	2	2	2	2		1	1	1	1	0.003	-1.643	0.500
8	1	2	2	2	2	2	…	1	1	1	1	0.005	-1.640	0.500
9	2	1	1	1	2	1		1	1	2	1	0.000	-1.100	0.530
10	2	1	1	1	2	1		1	1	1	1	0.000	-0.918	0.535
11	2	1	1	1	2	2		1	1	1	1	0.001	-0.569	0.540
12	2	1	1	2	1	1		1	1	1	1	0.000	-0.177	0.541
13	2	1	1	2	2	2		1	1	1	1	0.009	-1.133	0.529
14	2	1	1	2	2	2		2	1	1	1	0.000	0.066	0.542
15	2	1	1	2	2	2		1	1	1	1	0.000	0.333	0.546
16	2	1	2	1	1	2		1	1	1	1	0.000	-0.780	0.538

Items 7 – 17 sowie weitere beobachtete Antwortmuster fehlen aus Platzgründen

2. Matrizentest: Verteilung der geschätzten Personenparameter



Würde es sich bei dem analysierten Datensatz um eine Normstichprobe handeln, so könnte die Verteilung der geschätzten Personenparameter zur Normierung verwendet werden (z.B.: Berechnung eines Prozentranges für eine neue beobachtete Person)

3. Beispiel: Ordentlichkeit

- Skala Ordentlichkeit aus dem „NEO – Persönlichkeitsinventar“ (NEO-PI-R)
  - 8 Items
    (z.B.: „Ich lasse gerne alles an seinem Platz, damit ich weiß, wo es ist “)
  - 5 – stufiges Itemformat („starke Ablehnung“, „Ablehnung“, „neutral“, „Zustimmung“, „starke Zustimmung“)
  - Stichprobe bestehend aus 789 Personen

- Schätzung der Itemparameter:

Restriktion der Diskriminationsparameter um das PCM zu erhalten (constraint = “1PL“).

  - PCM:

```
> pcm <- gpcm(data = Ordentlichkeit, constraint = "1PL")
```

  - GPCM:

```
> gpcm <- gpcm(data = Ordentlichkeit)
```

3. Ordentlichkeit: Absoluter Modelltest PCM


```
> GoF.gpcm(pcm, simulate.p.value = FALSE)

Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Ordentlichkeit, constraint = "1PL")

Tobs: 1179454
df: 390591
p-value: <0.001

> GoF.gpcm(pcm, simulate.p.value = TRUE, B = 200)

Parametric Bootstrap Approximation to Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Ordentlichkeit, constraint = "1PL")

Tobs: 1179454
# data-sets: 201
p-value: 0.015
```

Sowohl für die Variante mit der theoretischen Prüfverteilung (oben) als auch für die Variante mit Bootstrap (unten) wird die Nullhypothese, dass das PCM in der Population gilt, abgelehnt.

3. Ordentlichkeit: Absoluter Modelltest GPCM


```
> GoF.gpcm(gpcm, simulate.p.value = FALSE)

Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Ordentlichkeit)

Tobs: 636470.1
df: 390584
p-value: <0.001
```

Für die Variante mit der theoretischen Prüfverteilung (oben) wird die Nullhypothese, dass das GPCM in der Population gilt, abgelehnt.

Für die Variante mit Bootstrap (unten) wird die Nullhypothese (gerade noch) beibehalten.

```
> GoF.gpcm(gpcm, simulate.p.value = TRUE, B = 200)

Parametric Bootstrap Approximation to Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Ordentlichkeit)

Tobs: 636470.1
# data-sets: 201
p-value: 0.05
```

3. Ordentlichkeit: Modellvergleiche

```
> anova(pcm, gpcm)

 Likelihood Ratio Table

          AIC      BIC  log.Lik    LRT df p.value
pcm  16386.64 16540.78 -8160.32         33
gpcm 16179.62 16366.45 -8049.81 221.02 40  <0.001
```

- Likelihood-Quotienten-Test:
  Die Nullhypothese, dass das PCM in der Population genauso gut passt wie das GPCM, wird abgelehnt.

- Informationskriterien:
  - AIC: GPCM „besser“ als PCM
  - BIC: GPCM „besser“ als PCM

Hinweis:
Wir werden im Folgenden aus pädagogischen Gründen die Parameterschätzungen des PCMs weiter betrachten, obwohl basierend auf den absoluten Modelltests und der Modellvergleiche eher das GPCM zu bevorzugen wäre!

3. Ordentlichkeit: Interpretation Itemparameter PCM


```
> summary(pcm)

Call:
gpcm(data = Ordentlichkeit, constraint = "1PL")

Model Summary:
   log.Lik      AIC      BIC
 -8160.322 16386.64 16540.78

Coefficients:
$010R_a
         value std.err z.value
Catgr.1 -2.994   0.244 -12.275
Catgr.2  0.623   0.156   3.992
Catgr.3  0.832   0.180   4.621
Catgr.4  4.719   0.465  10.138
Dscrmn   0.609   0.027  22.817

$040_a
         value std.err z.value
Catgr.1 -5.022   0.602  -8.346
Catgr.2 -1.517   0.210  -7.229
Catgr.3 -1.081   0.160  -6.756
Catgr.4  2.257   0.198  11.401
Dscrmn   0.609   0.027  22.817

$070R_a
         value std.err z.value
Catgr.1 -3.492   0.392  -8.903
Catgr.2 -0.888   0.215  -4.138
Catgr.3 -1.586   0.182  -8.696
Catgr.4  1.698   0.175   9.708
Dscrmn   0.609   0.027  22.817
```

Es fällt auf, dass bei 6 von 8 Items ungeordnete Schwellenparameter geschätzt werden

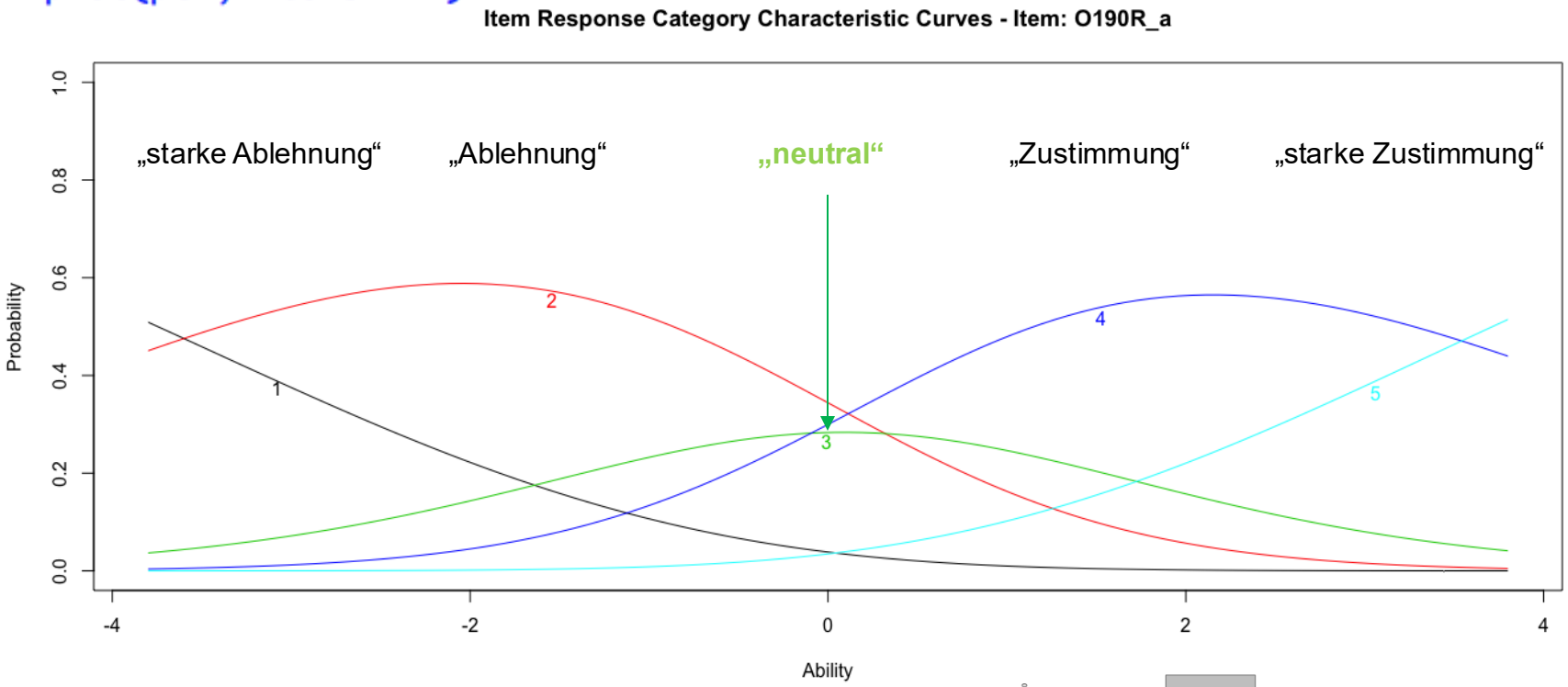
```
$0100_a
         value std.err z.value
Catgr.1 -4.723   0.480  -9.849
Catgr.2 -0.713   0.186  -3.830
Catgr.3 -1.037   0.163  -6.359
Catgr.4  2.986   0.237  12.606
Dscrmn   0.609   0.027  22.817

$0130R_a
         value std.err z.value
Catgr.1 -4.162   0.443  -9.398
Catgr.2 -0.308   0.222  -1.390
Catgr.3 -2.061   0.204 -10.098
Catgr.4  1.512   0.167   9.055
Dscrmn   0.609   0.027  22.817

$0160_a
         value std.err z.value
Catgr.1 -4.617   0.604  -7.639
Catgr.2 -1.296   0.248  -5.216
Catgr.3 -2.290   0.200 -11.436
Catgr.4  1.947   0.176  11.082
Dscrmn   0.609   0.027  22.817

$0190R_a
         value std.err z.value
Catgr.1 -3.600   0.299 -12.022
Catgr.2  0.319   0.164   1.946
Catgr.3 -0.093   0.167  -0.557
Catgr.4  3.541   0.304  11.644
Dscrmn   0.609   0.027  22.817

$0220R_a
         value std.err z.value
Catgr.1 -3.521   0.336 -10.486
Catgr.2 -0.188   0.188  -1.001
Catgr.3 -1.101   0.175  -6.277
Catgr.4  2.176   0.202  10.780
Dscrmn   0.609   0.027  22.817
```

...

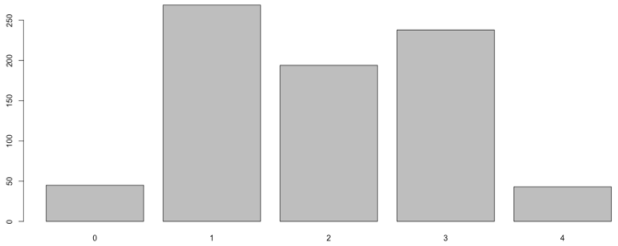
- Hinweis:
  - Das „ltm“ Paket verwendet eine spezielle Parametrisierung des PCM, bei der statt der Varianz der latenten Variable ein für alle Items als konstant angenommener Diskriminationsparameter geschätzt wird.
  - Die Varianz der latenten Variable muss in dieser Parametrisierung auf 1 gesetzt werden. Dies bedeutet, dass im Vergleich zu unserer Parametrisierung eine andere Einheit für die latente Variable gewählt wird.
  - Durch diesen Einheitswechsel ändern sich auch auch die Werte der Schwellenparameter und ihrer Schätzwerte. Die Interpretation der Schwellenparameter ändert sich jedoch nicht.
  - Man kann die Schätzwerte für die Parameter der ltm-Parametrisierung in Schätzwerte für die Parameter unserer Parametrisierung umrechnen. Dies werden wir jedoch im Rahmen dieser Vorlesung nicht besprechen.

3. Ordentlichkeit: Geschätzte CCCs von Item 7


```
> plot(pcm, items = 7)
```




```
> barplot(table(Ordentlichkeit[, 7]))
```



```
> pcm$coefficients[[7]]
    Catgr.1     Catgr.2     Catgr.3     Catgr.4      Dscrmn
-3.59987652  0.31923599 -0.09305706  3.54142565  0.60900092
```

3. Ordentlichkeit: Schätzung der Personenparameter


```
> thetas <- factor.scores(pcm, method = "EAP")
```

- Exp: Erwartete Häufigkeit
- z1: EAP Schätzer
- se.z1: Standardfehler

```
> thetas
```

95% KI für eine Person mit Antwortmuster 731:

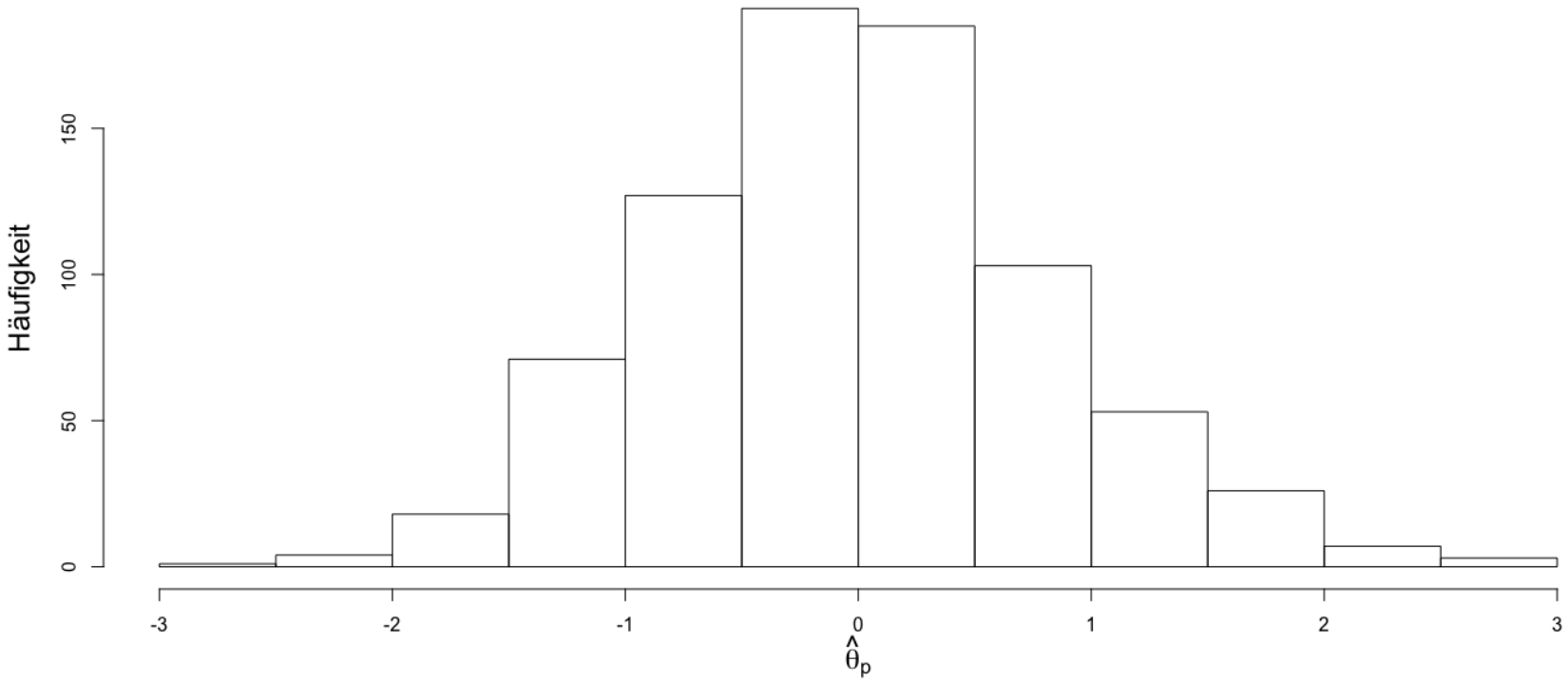
$$\left[\hat{\theta} \pm 1{,}96 \cdot \widehat{SE}\left(\hat{\theta}\right)\right] = [1{,}77 \pm 1{,}96 \cdot 0{,}63] = [0{,}54; 3{,}00]$$

```
Call:
gpcm(data = Ordentlichkeit, constraint = "1PL")

Scoring Method: Expected A Posteriori

Factor-Scores for observed response patterns:
    O10R_a O40_a O70R_a O100_a O130R_a O160_a O190R_a O220R_a Obs    Exp     z1 se.z1
1        1     1      2      3       1      2       1       1   1 0.001 -2.720 0.585
2        1     1      4      1       3      5       1       4   1 0.000 -1.253 0.525
```

...

731	5	5	4	5	4	4	4	5	1	1.96e-02	1.77	0.631
732	5	5	5	4	5	4	2	4	1	1.13e-02	1.31	0.605
733	5	5	5	5	5	3	3	4	1	8.92e-04	1.54	0.618
734	5	5	5	5	5	4	4	5	1	3.26e-02	2.28	0.660
735	5	5	5	5	5	5	4	3	1	5.18e-03	2.02	0.645
736	5	5	5	5	5	5	4	5	1	4.35e-02	2.55	0.675

3. Ordentlichkeit: Verteilung der geschätzten Personenparameter



Würde es sich bei dem analysierten Datensatz um eine Normstichprobe handeln, so könnte die Verteilung der geschätzten Personenparameter zur Normierung verwendet werden (z.B.: Berechnung eines Prozentranges für eine neue beobachtete Person)

4. Beispiel: Überblick

- Skala Überblick aus dem „Fragebogen Räumliche Strategien“ (FRS)
  - 7 Items
    (z.B.: „ Ich stelle mir die Umgebung stets wie auf einer „mentalen Karte“ (Überblicksansicht) vor.“)
  - 7 – stufiges Itemformat mit beschrifteten Endpolen
    („trifft überhaupt nicht zu“, „trifft vollkommen zu“)
  - Repräsentative Stichprobe bestehend aus 1041 Personen
    (Datensatz besteht aus dem Campus File des longitudinalen GESIS Panel)

Restriktion der Diskriminationsparameter um das PCM zu erhalten (constraint = “1PL“).

- Schätzung der Itemparameter:
  - PCM:

```
> pcm <- gpcm(data = Ueberblick, constraint = "1PL")
```

  - GPCM:

```
> gpcm <- gpcm(data = Ueberblick)
```

4. Überblick: Absoluter Modelltest PCM


```
> GoF.gpcm(pcm, simulate.p.value = FALSE)

Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Ueberblick, constraint = "1PL")

Tobs: 86496679
df: 823499
p-value: <0.001
```

Sowohl für die Variante mit der theoretischen Prüfverteilung (oben) als auch für die Variante mit Bootstrap (unten) wird die Nullhypothese, dass das PCM in der Population gilt, abgelehnt.

```
> GoF.gpcm(pcm, simulate.p.value = TRUE, B = 200)

Parametric Bootstrap Approximation to Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Ueberblick, constraint = "1PL")

Tobs: 86496679
# data-sets: 200
p-value: 0.005
```

4. Überblick: Absoluter Modelltest GPCM


```
> GoF.gpcm(gpcm, simulate.p.value = FALSE)

Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Ueberblick)

Tobs: 29201649
df: 823493
p-value: <0.001
```

Sowohl für die Variante mit der theoretischen Prüfverteilung (oben) als auch für die Variante mit Bootstrap (unten) wird die Nullhypothese, dass das GPCM in der Population gilt, abgelehnt.

```
> GoF.gpcm(gpcm, simulate.p.value = TRUE, B = 200)

Parametric Bootstrap Approximation to Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Ueberblick)

Tobs: 29201649
# data-sets: 201
p-value: 0.005
```

4. Überblick: Modellvergleiche


```
> anova(pcm, gpcm)

 Likelihood Ratio Table

          AIC      BIC   log.Lik    LRT df p.value
pcm  23282.81 23495.57 -11598.41         43
gpcm 23021.28 23263.73 -11461.64 273.53 49  <0.001
```

Hinweis:
Im Folgenden werden die Parameterschätzungen des flexibleren GPCM weiter betrachtet

- Likelihood-Quotienten-Test:
  Die Nullhypothese, dass das PCM in der Population genauso gut passt wie das GPCM, wird abgelehnt.

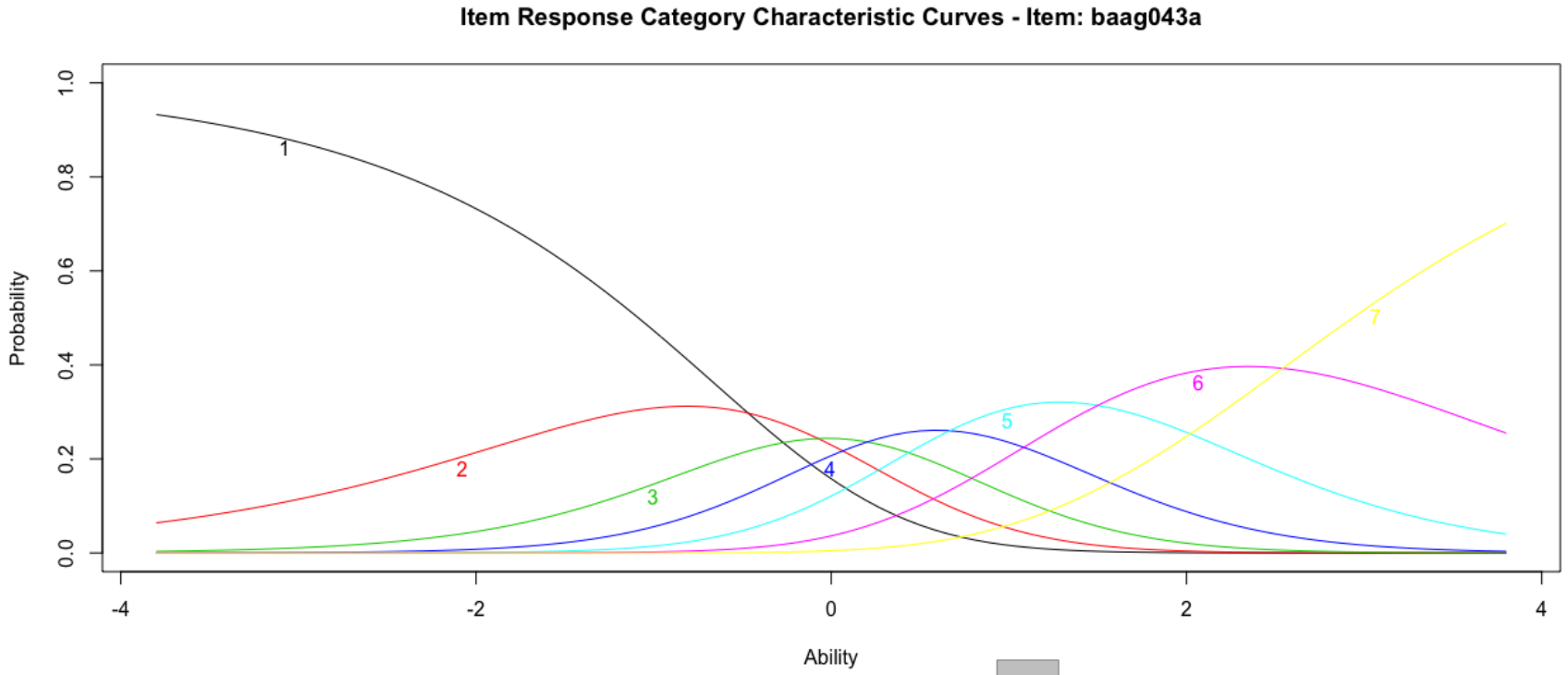
- Informationskriterien:
  - AIC: GPCM „besser“ als PCM
  - BIC: GPCM „besser“ als PCM

4. Überblick: Interpretation Itemparameter GPCM

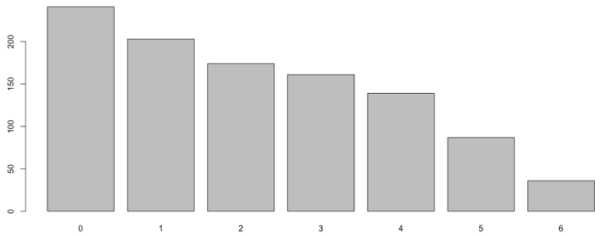

```
> summary(gpcm)

Call:
gpcm(data = Ueberblick)

Model Summary:
   log.Lik      AIC      BIC
 -11461.64 23021.28 23263.73

Coefficients:
$baag036a
         value std.err z.value
Catgr.1 -1.421   0.207  -6.877
Catgr.2 -0.815   0.193  -4.226
Catgr.3 -0.803   0.181  -4.441
Catgr.4 -0.451   0.161  -2.807
Catgr.5 -0.252   0.143  -1.756
Catgr.6  1.035   0.140   7.374
Dscrmn   0.741   0.049  15.151

$baag037a
         value std.err z.value
Catgr.1 -1.003   0.133  -7.561
Catgr.2 -0.423   0.132  -3.210
Catgr.3 -0.274   0.133  -2.069
Catgr.4  0.150   0.131   1.141
Catgr.5  0.305   0.128   2.376
Catgr.6  1.458   0.142  10.247
Dscrmn   0.929   0.061  15.190

$baag037a
         value std.err z.value
Catgr.1 -1.003   0.133  -7.561
Catgr.2 -0.423   0.132  -3.210
Catgr.3 -0.274   0.133  -2.069
Catgr.4  0.150   0.131   1.141
Catgr.5  0.305   0.128   2.376
Catgr.6  1.458   0.142  10.247
Dscrmn   0.929   0.061  15.190

$baag041a
         value std.err z.value
Catgr.1 -1.108   0.073 -15.128
Catgr.2 -0.504   0.068  -7.389
Catgr.3 -0.294   0.068  -4.342
Catgr.4  0.213   0.065   3.287
Catgr.5  0.744   0.071  10.480
Catgr.6  1.406   0.088  15.907
Dscrmn   2.049   0.144  14.269

$baag043a
         value std.err z.value
Catgr.1 -0.468   0.129  -3.631
Catgr.2 -0.074   0.135  -0.548
Catgr.3  0.203   0.142   1.428
Catgr.4  0.667   0.151   4.411
Catgr.5  1.501   0.182   8.256
Catgr.6  2.541   0.266   9.567
Dscrmn   0.804   0.053  15.242

$baag045a
         value std.err z.value
Catgr.1 -1.553   0.122 -12.739
Catgr.2 -0.931   0.101  -9.228
Catgr.3 -0.285   0.099  -2.895
Catgr.4 -0.306   0.095  -3.227
Catgr.5  0.267   0.086   3.111
Catgr.6  1.265   0.100  12.702
Dscrmn   1.323   0.082  16.048

$baag050a
         value std.err z.value
Catgr.1 -2.168   0.207 -10.462
Catgr.2 -1.387   0.152  -9.150
Catgr.3 -0.690   0.132  -5.244
Catgr.4 -0.571   0.121  -4.730
Catgr.5 -0.193   0.105  -1.839
Catgr.6  1.026   0.108   9.523
Dscrmn   0.997   0.065  15.337

$baag053a
         value std.err z.value
Catgr.1 -1.170   0.066 -17.719
Catgr.2 -0.450   0.062  -7.306
Catgr.3 -0.239   0.059  -4.021
Catgr.4  0.203   0.058   3.512
Catgr.5  0.838   0.064  13.027
Catgr.6  1.754   0.093  18.956
Dscrmn   2.442   0.176  13.862


Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.042
optimizer: nlminb
```

4. Überblick: Geschätzte CCCs von Item 4


```
> plot(gpcm, items = 4)
```



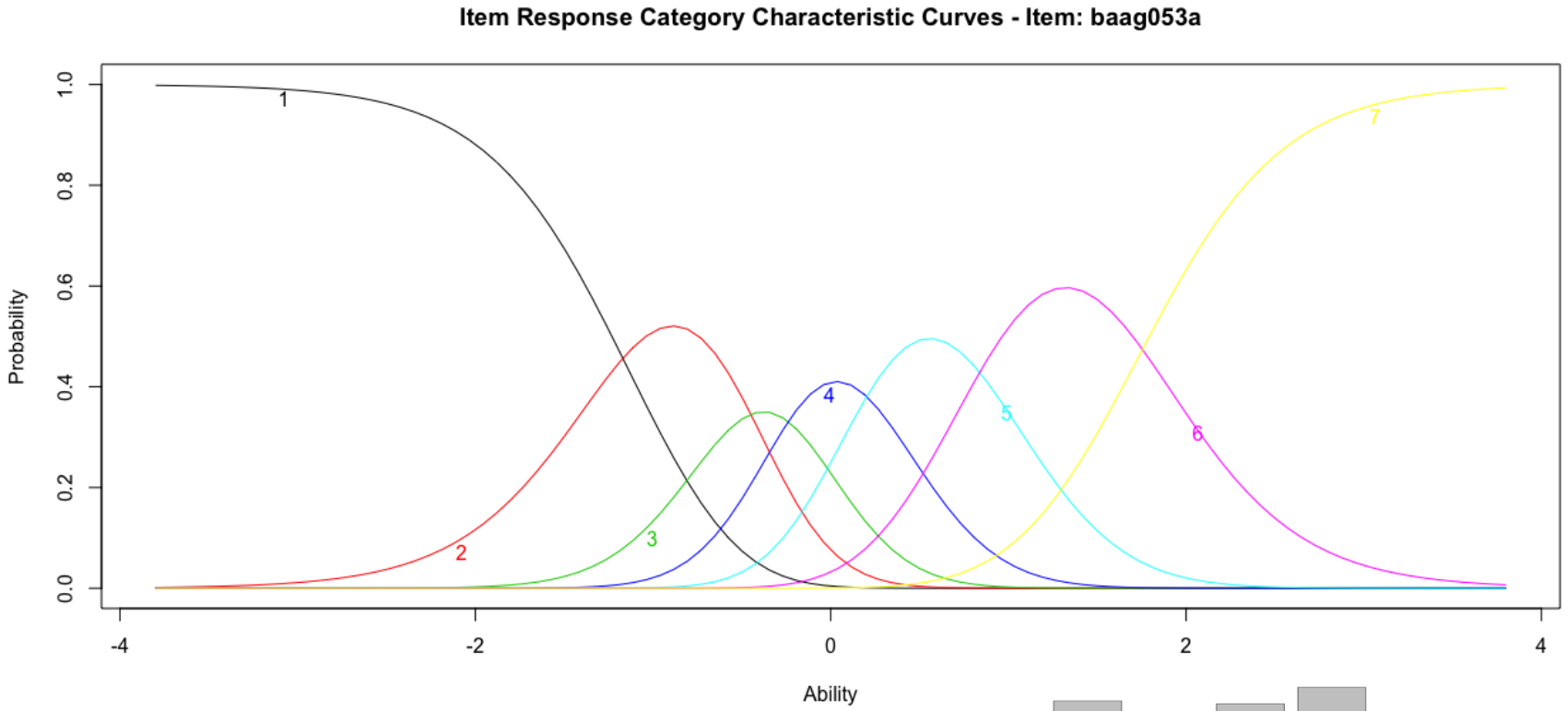

```
> barplot(table(Ueberblick[, 4]))
```



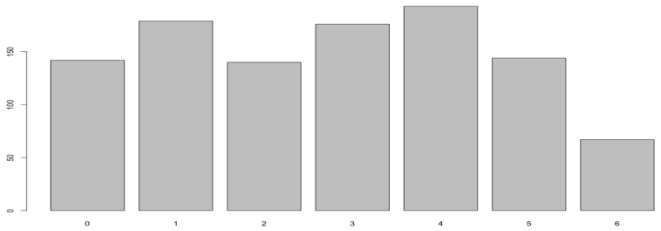
```
> gpcm$coefficients[[4]]
    Catgr.1     Catgr.2     Catgr.3     Catgr.4     Catgr.5     Catgr.6      Dscrmn
-0.46778964 -0.07404667  0.20281732  0.66679991  1.50081440  2.54054584  0.80415109
```

4. Überblick: Geschätzte CCCs von Item 7


```
> plot(gpcm, items = 7)
```




```
> barplot(table(Ueberblick[, 7]))
```



```
> gpcm$coefficients[[7]]
   Catgr.1    Catgr.2    Catgr.3    Catgr.4    Catgr.5    Catgr.6     Dscrmn
-1.1700664 -0.4499181 -0.2388708  0.2025510  0.8380184  1.7541790  2.4417811
```

4. Überblick: Schätzung der Personenparameter

```
> thetas <- factor.scores(gpcm, method = "EAP")

> thetas

Call:
gpcm(data = Ueberblick)

Scoring Method: Expected A Posteriori

Factor-Scores for observed response patterns:
    baag036a baag037a baag041a baag043a baag045a baag050a baag053a Obs   Exp     z1 se.z1
1          1        1        1        1        1        1        1   7 3.314 -2.486 0.534
2          1        1        1        1        1        2        1   7 2.754 -2.232 0.478
3          1        1        1        1        1        2        2   1 0.375 -1.765 0.405
4          1        1        1        1        1        3        1   4 1.320 -2.023 0.440
5          1        1        1        1        1        4        1   1 0.384 -1.840 0.417
6          1        1        1        1        1        6        1   2 0.029 -1.547 0.340
7          1        1        1        1        2        1        1   2 1.209 -2.160 0.463
8          1        1        1        1        2        3        2   1 0.284 -1.467 0.302
9          1        1        1        1        2        3        3   1 0.030 -1.281 0.282
10         1        1        1        1        2        4        1   1 0.303 -1.632 0.372
11         1        1        1        1        2        5        1   1 0.112 -1.511 0.323
12         1        1        1        1        2        6        1   1 0.032 -1.421 0.281
```

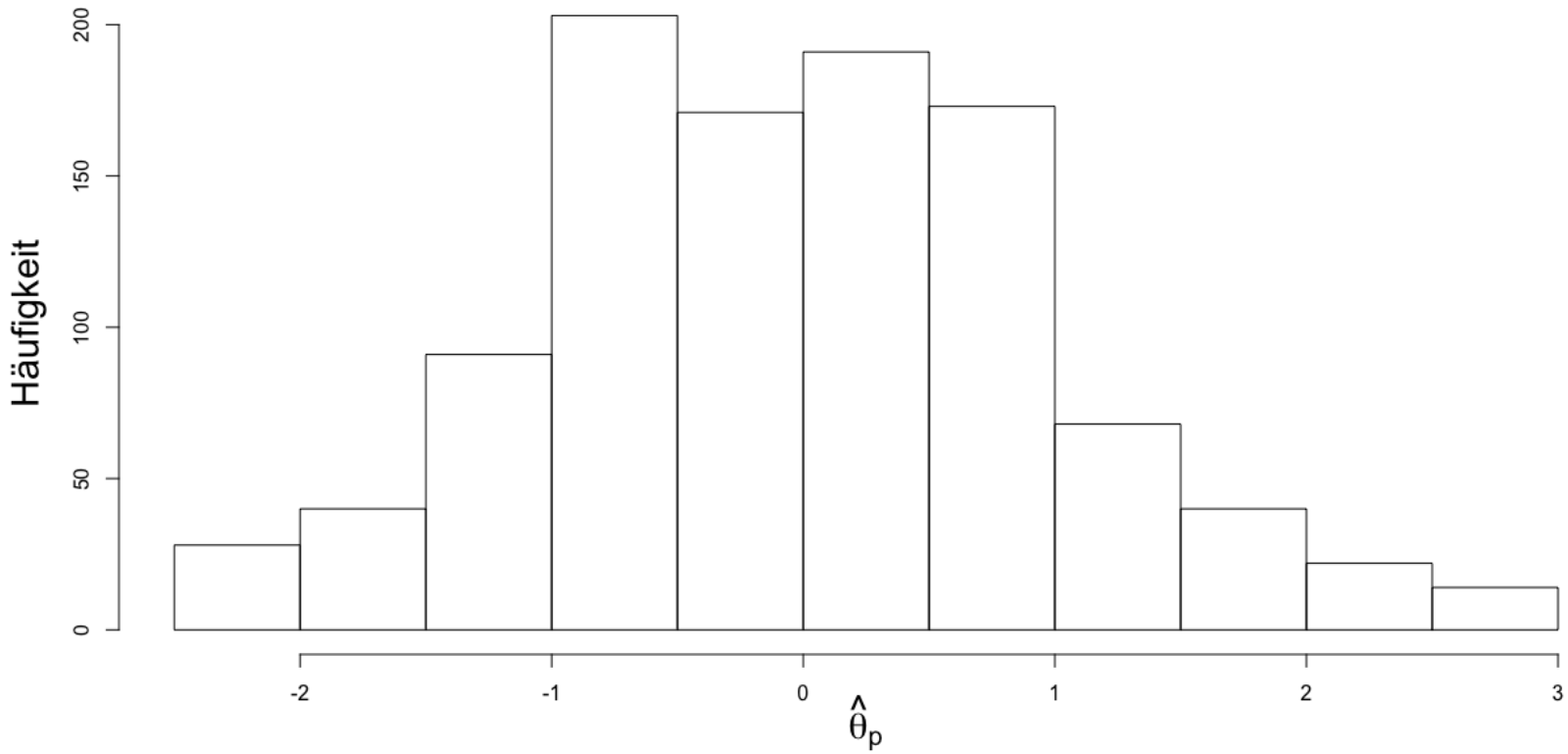
- Exp: Erwartete Häufigkeit
- z1: EAP Schätzer
- se.z1: Standardfehler

95% KI für eine Person mit Antwortmuster 11:

$$\left[\hat{\theta} \pm 1{,}96 \cdot \widehat{SE}\left(\hat{\theta}\right)\right] = [-1{,}51 \pm 1{,}96 \cdot 0{,}32] = [-2{,}14; -0{,}88]$$

Weitere beobachtete Antwortmuster fehlen aus Platzgründen

4. Überblick: Verteilung der geschätzten Personenparameter



Würde es sich bei dem analysierten Datensatz um eine Normstichprobe handeln, so könnte die Verteilung der geschätzten Personenparameter zur Normierung verwendet werden (z.B.: Berechnung eines Prozentranges für eine neue beobachtete Person)

Anmerkungen

- AIC, BIC und LRT kommen nicht immer zum gleichen Ergebnis
  → Entscheidung in der Praxis schwierig

- Varianten des Pearson-χ^2-Tests kommen nicht immer zum gleichen Ergebnis
  → Entscheidung in der Praxis schwierig, Bootstrap wird teilweise kritisiert!

- Oft muss davon ausgegangen werden, dass keines der betrachteten Testmodelle gut auf die vorliegenden Daten passt
  → Ist ein schlecht passendes Modell besser als gar keins?

- Die betrachteten Stichproben waren teilweise eher klein
  → Normstichproben sollten mehrere Tausend Personen umfassen