Assumptions and Properties of IRT Models

Derek Briggs

EDUC 8720

Contents

Overview

IRT models rest on several key **assumptions** and have important **properties** that distinguish them from classical test theory. Understanding these is essential for proper application and interpretation.

Assumptions	Properties
Local Independence	Parameter Invariance
Appropriate Dimensionality	Scale Indeterminacy
Functional Form (ICC shape)	
Continuous latent variable	

Assumption 1: Local Independence

Definition

Item responses are **statistically independent, conditional on the latent variable** θ.

Put differently: the only reason item responses are correlated is due to their common dependence on θ.

Mathematical Expression

This assumption gives us the important result:

$$P(X_{1p} = 1 \text{ and } X_{2p} = 1|\theta_p) = P(X_{1p} = 1|\theta_p) \times P(X_{2p} = 1|\theta_p)$$

More generally, for a full response pattern:

$$P(X_1, X_2, \ldots, X_I|\theta) = \prod_{i=1}^{I} P(X_i|\theta)$$

This is the same assumption made in factor analysis. For multidimensional IRT, we assume item responses are independent conditional on **all** dimensions of θ included in the model.

Demonstration

```
# 3PL probability function
calc_prob <- function(theta, a, b, c) {
  c + (1 - c) * exp(a * (theta - b)) / (1 + exp(a * (theta - b)))
}

# Two items
a <- c(1.5, 1.2)
b <- c(-0.5, 0.5)
c <- c(0.2, 0.15)

# For a person with theta = 0
theta <- 0
p1 <- calc_prob(theta, a[1], b[1], c[1])
p2 <- calc_prob(theta, a[2], b[2], c[2])

cat("P(X1 = 1 | theta = 0) =", round(p1, 3), "\n")
```

```
## P(X1 = 1 | theta = 0) = 0.743
```

```
cat("P(X2 = 1 | theta = 0) =", round(p2, 3), "\n")
```

```
## P(X2 = 1 | theta = 0) = 0.451
```

```
cat("\nUnder local independence:\n")
```

```
##
## Under local independence:
```

```
cat("P(X1 = 1 AND X2 = 1 | theta = 0) =", round(p1 * p2, 3), "\n")
```

```
## P(X1 = 1 AND X2 = 1 | theta = 0) = 0.335
```

What Causes Violations of Local Independence?

1. **Multidimensionality**: Test measures more than one latent trait
2. **Item chaining**: Answer to one item depends on previous item
3. **Testlet effects**: Items share common stimulus (e.g., reading passage)
4. **Speededness**: Time pressure creates dependence among later items
5. **Method effects**: Similar item formats create additional covariance

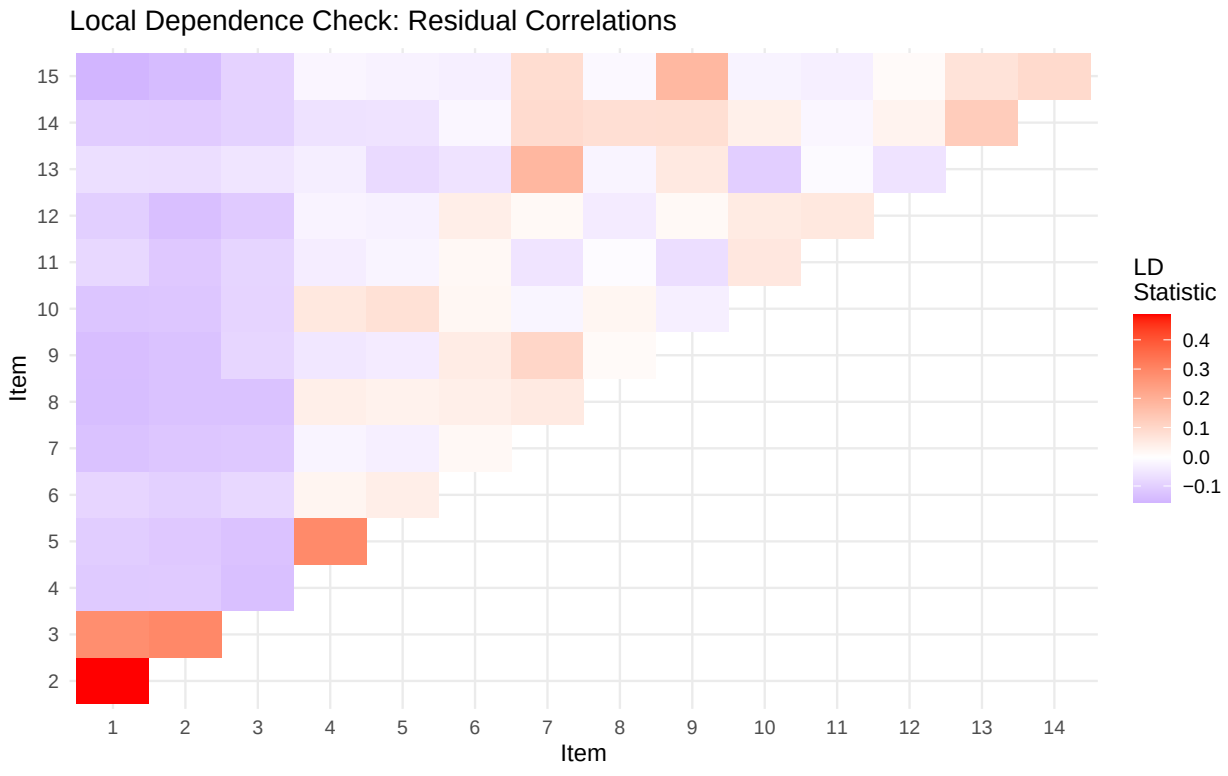
Checking Local Independence

```r
library(mirt)

# Load example data
forma <- read.csv("../Data/pset1_formA.csv")
forma <- forma[, 1:15]

# Fit model
mod <- mirt(forma, 1, itemtype = "2PL", verbose = FALSE)

# Residual correlations can indicate local dependence
residuals_ld <- residuals(mod, type = "LD")
```

```
## LD matrix (lower triangle) and standardized residual correlations (upper triangle)
##
## Upper triangle summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -0.154  -0.093  -0.032  -0.016   0.038   0.484
##
##          V1      V2     V3      V4     V5     V6     V7     V8     V9    V10
## V1            0.484  0.279  -0.110 -0.104 -0.086 -0.127 -0.136 -0.134 -0.121
## V2  459.027          0.293  -0.109 -0.113 -0.098 -0.119 -0.127 -0.127 -0.118
## V3  152.230 167.610         -0.130 -0.125 -0.080 -0.114 -0.126 -0.084 -0.089
## V4   23.655  23.454 32.860          0.289  0.025 -0.023  0.042 -0.050  0.059
## V5   21.146  24.949 30.685 164.081         0.042 -0.033  0.034 -0.041  0.076
## V6   14.506  18.701 12.507   1.234  3.447         0.019  0.041  0.048  0.021
## V7   31.387  27.572 25.366   1.068  2.155  0.687         0.054  0.104 -0.021
## V8   35.958  31.482 31.212   3.388  2.221  3.232  5.642         0.013  0.024
## V9   34.987  31.693 13.973   4.852  3.221  4.535 21.169  0.327         -0.032
## V10  28.624  27.450 15.452   6.754 11.401  0.896  0.850  1.115  2.063
## V11  12.485  24.975 14.941   2.563  0.934  0.661  5.875  0.125  9.037  7.042
## V12  19.548  33.934 23.200   1.096  1.678  3.597  0.500  3.328  0.495  5.216
## V13   8.582   9.172  5.771   2.357 11.101  6.358 64.484  0.959  6.220 19.683
## V14  22.078  22.152 17.009   6.990  6.615  0.664 15.912 12.914 13.127  2.843
## V15  46.480  38.273 16.661   0.784  1.408  2.084 14.515  0.378 62.048  1.261
##        V11    V12    V13    V14    V15
## V1  -0.080 -0.100 -0.066 -0.106 -0.154
## V2  -0.113 -0.132 -0.068 -0.106 -0.140
## V3  -0.087 -0.109 -0.054 -0.093 -0.092
## V4  -0.036 -0.024 -0.035 -0.060 -0.020
## V5  -0.022 -0.029 -0.075 -0.058 -0.027
## V6   0.018  0.043 -0.057 -0.018 -0.033
## V7  -0.055  0.016  0.181  0.090  0.086
```

```
## V8  -0.008 -0.041 -0.022  0.081 -0.014
## V9  -0.068  0.016  0.056  0.082  0.178
## V10  0.060  0.052 -0.100  0.038 -0.025
## V11         0.060 -0.011 -0.018 -0.034
## V12  6.969        -0.058  0.030  0.012
## V13  0.227  6.535         0.128  0.070
## V14  0.666  1.767 32.279         0.094
## V15  2.217  0.274  9.583 17.458
```

```r
# Visualize (upper triangle of matrix)
library(ggplot2)

# Convert to long format for plotting
ld_matrix <- as.matrix(residuals_ld)
ld_df <- expand.grid(Item1 = 1:15, Item2 = 1:15)
ld_df$LD <- as.vector(ld_matrix)
ld_df <- ld_df[ld_df$Item1 < ld_df$Item2, ]

ggplot(ld_df, aes(x = factor(Item1), y = factor(Item2), fill = LD)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0,
                       name = "LD\nStatistic") +
  labs(x = "Item", y = "Item", title = "Local Dependence Check: Residual Correlations") +
  theme_minimal()
```



Large positive values suggest items may be locally dependent.

Quick Aside: Useful Probability Rules

These rules are essential for understanding IRT:

1. **Complement Rule**: $P(A) = 1 - P(\neg A)$
   - Probability of A is 1 minus the probability of A not happening
2. **Multiplication Rule (Independence)**: $P(A \text{ and } B) = P(A) \times P(B)$
   - When A and B are independent
   - This is the **joint probability** of A and B occurring
   - Local independence allows us to use this for item responses
3. **Conditional Probability**: $P(A|B) = \frac{P(A \text{ and } B)}{P(B)}$
   - Probability of A given that B happened
4. **Bayes' Rule**: $P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$

Application to Response Patterns

```
# Calculate probability of response pattern (1, 0, 1) for theta = 0.5
theta <- 0.5
items <- data.frame(a = c(1, 1.5, 1.2), b = c(-1, 0, 1), c = c(0.2, 0.2, 0.15))

probs <- sapply(1:3, function(i) calc_prob(theta, items$a[i], items$b[i], items$c[i]))

cat("Item probabilities at theta = 0.5:\n")
```

```
## Item probabilities at theta = 0.5:
```

```
cat("P(X1 = 1) =", round(probs[1], 3), "\n")
```

```
## P(X1 = 1) = 0.854
```

```
cat("P(X2 = 1) =", round(probs[2], 3), "\n")
```

```
## P(X2 = 1) = 0.743
```

```
cat("P(X3 = 1) =", round(probs[3], 3), "\n")
```

```
## P(X3 = 1) = 0.451
```

```
# Pattern (1, 0, 1)
pattern_prob <- probs[1] * (1 - probs[2]) * probs[3]
cat("\nP(X1=1, X2=0, X3=1 | theta=0.5) =", round(pattern_prob, 4), "\n")
```

```
##
## P(X1=1, X2=0, X3=1 | theta=0.5) = 0.0989
```

Assumption 2: Appropriate Dimensionality

Definition

The model contains the **appropriate number of latent dimensions** to account for the covariance among items.

- For unidimensional IRT: One θ is sufficient
- For multidimensional IRT: Multiple θs are needed

Key Points

- If a test is **multidimensional** and we fit only a single latent variable, this can cause **local item dependence**
- Dimensionality refers to the number of latent variables needed to model the response data
- This may or may not correspond to the hypothesized dimensionality of the theoretical construct
- Most IRT models used in practice are unidimensional

Assessing Dimensionality

```
par(mfrow = c(1, 2))

# Eigenvalue plot (scree plot)
eigenvalues <- eigen(cor(forma))$values
plot(eigenvalues, type = "b", pch = 19,
     xlab = "Component Number", ylab = "Eigenvalue",
     main = "Scree Plot")
abline(h = 1, lty = 2, col = "red")

# Ratio of first to second eigenvalue
cat("First eigenvalue:", round(eigenvalues[1], 2), "\n")
```

```
## First eigenvalue: 5.29
```

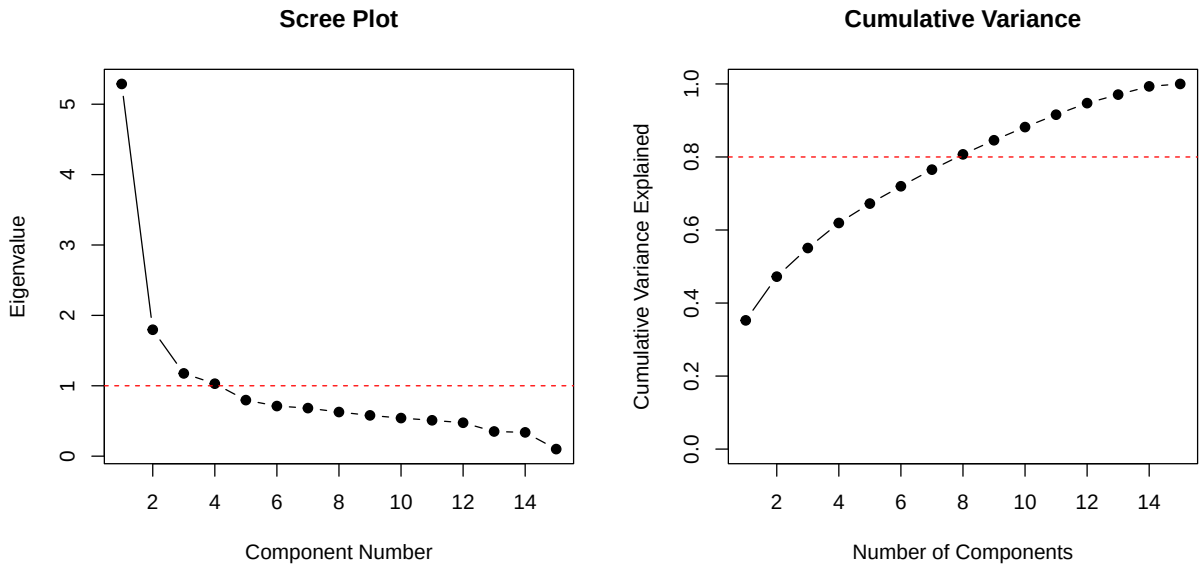
```
cat("Second eigenvalue:", round(eigenvalues[2], 2), "\n")
```

```
## Second eigenvalue: 1.8
```

```
cat("Ratio (1st/2nd):", round(eigenvalues[1]/eigenvalues[2], 2), "\n")
```

```
## Ratio (1st/2nd): 2.94
```

```
# Cumulative variance explained
cum_var <- cumsum(eigenvalues) / sum(eigenvalues)
plot(cum_var, type = "b", pch = 19,
     xlab = "Number of Components", ylab = "Cumulative Variance Explained",
     main = "Cumulative Variance", ylim = c(0, 1))
abline(h = 0.8, lty = 2, col = "red")
```




```r
par(mfrow = c(1, 1))
```

Guidelines:

- A dominant first eigenvalue suggests unidimensionality
- Ratio of first to second eigenvalue > 3 often indicates essential unidimensionality
- But these are rough guidelines, not strict rules

Assumption 3: Functional Form (ICC Shape)

Definition

We assume that the probability of correct responses follows the **logistic form** (or normal ogive):

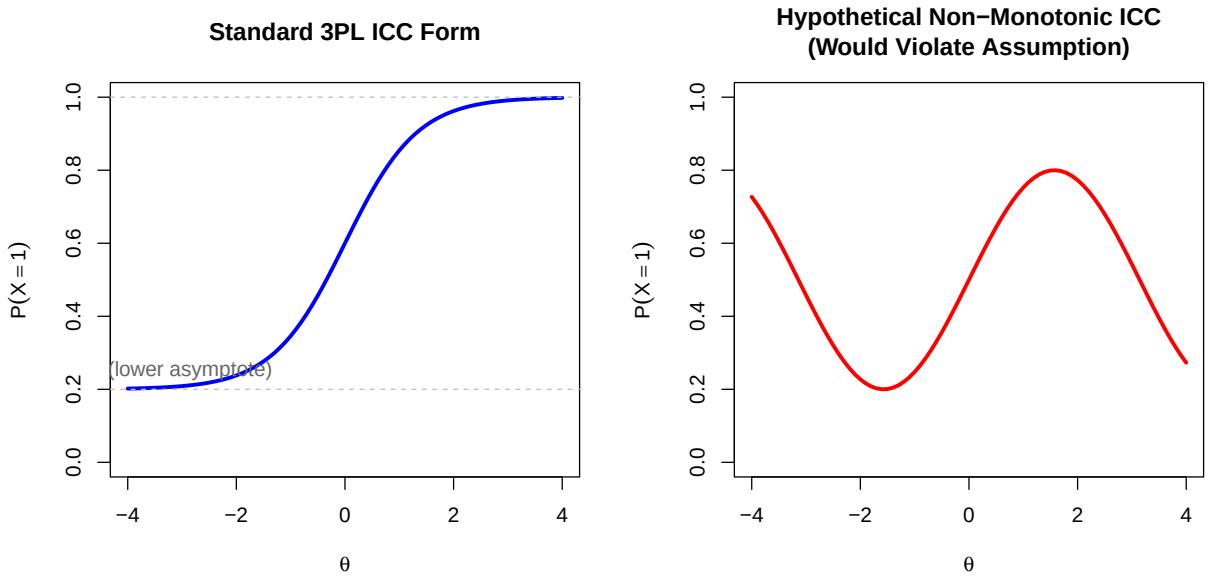
$$P(X_{ip} = 1|\theta_p) = c_i + (1 - c_i)\frac{\exp(a_i(\theta_p - b_i))}{1 + \exp(a_i(\theta_p - b_i))}$$

Key Characteristics

1. **Monotonically increasing** probability as a function of θ
2. **S-shaped** (sigmoidal) curve
3. Described by the a, b, and c parameters
4. Lower asymptote determined by c
5. Upper asymptote is 1 (or can be less with a 4PL model)

```r
theta <- seq(-4, 4, 0.1)

par(mfrow = c(1, 2))
```

```r
# Standard ICC
plot(theta, calc_prob(theta, 1.5, 0, 0.2), type = "l", lwd = 3, col = "blue",
     xlab = expression(theta), ylab = expression(P(X == 1)),
     main = "Standard 3PL ICC Form",
     ylim = c(0, 1))
abline(h = c(0.2, 1), lty = 2, col = "gray")
text(-3, 0.25, "c (lower asymptote)", col = "gray40")

# What if ICC were NOT monotonic? (This would violate the assumption)
theta_range <- seq(-4, 4, 0.1)
non_monotonic <- 0.5 + 0.3 * sin(theta_range)  # Hypothetical non-monotonic

plot(theta_range, non_monotonic, type = "l", lwd = 3, col = "red",
     xlab = expression(theta), ylab = expression(P(X == 1)),
     main = "Hypothetical Non-Monotonic ICC\n(Would Violate Assumption)",
     ylim = c(0, 1))
```




```r
par(mfrow = c(1, 1))
```

Checking ICC Form with Empirical Plots

```r
# Compare model-implied ICC to empirical ICC
library(mirt)

par(mfrow = c(1, 3))

# Check item fit for 3 items
for (item in 1:3) {
  itemfit(mod, group.bins = 10, empirical.plot = item)
}
```

```r
par(mfrow = c(1, 1))
```

If the empirical points deviate systematically from the model-implied curve, the functional form assumption may be violated.

Property 1: Parameter Invariance

Definition

If the model fits the data, **item parameter estimates should be the same regardless of the group used to estimate them**.

- We should get the same a, b, c parameters whether we use high-ability or low-ability examinees
- Similarly, person parameters should be the same regardless of which items are used

Why This Matters

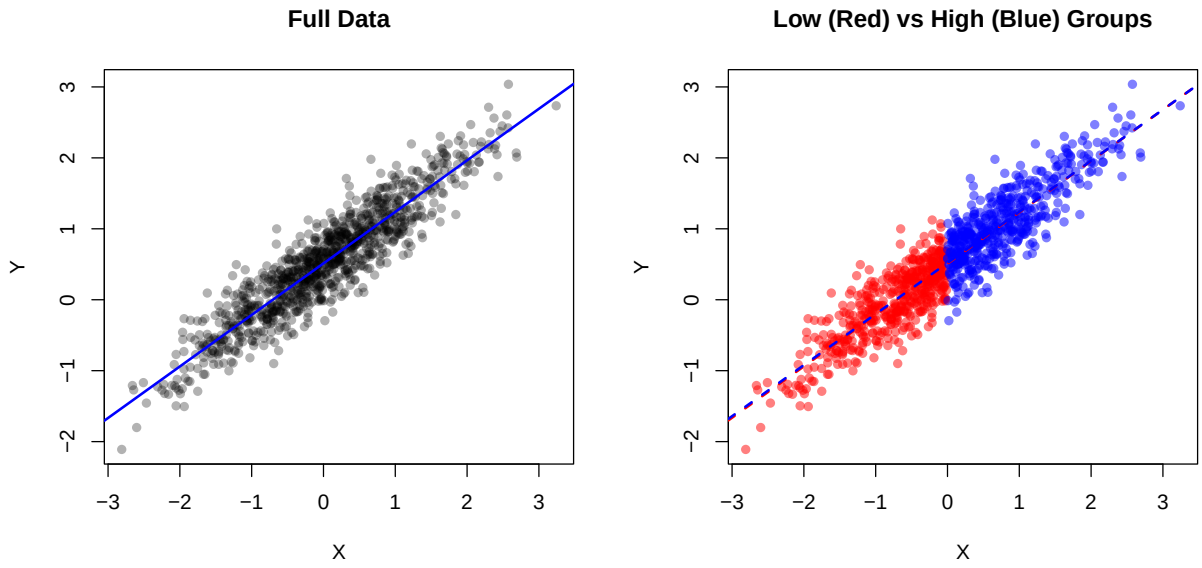
CTT	IRT
Item p-values depend on sample	Item parameters are invariant (if model fits)
Discrimination depends on sample variance	Discrimination is a stable property
Need matched samples for comparison	Can compare across different samples

Analogy: Linear Regression

In linear regression, we assume the slope and intercept are the same regardless of which subset of data we use to estimate them.

```r
set.seed(123)

# Generate population data
n <- 1000
x <- rnorm(n, 0, 1)
y <- 0.5 + 0.7 * x + rnorm(n, 0, 0.3)

# Split into "low" and "high" groups
low_group <- x < 0
high_group <- x >= 0

par(mfrow = c(1, 2))

# Full data
plot(x, y, pch = 16, col = rgb(0, 0, 0, 0.3),
     main = "Full Data", xlab = "X", ylab = "Y")
abline(lm(y ~ x), col = "blue", lwd = 2)

# Separate groups
```

```r
plot(x[low_group], y[low_group], pch = 16, col = rgb(1, 0, 0, 0.5),
     xlim = range(x), ylim = range(y),
     main = "Low (Red) vs High (Blue) Groups", xlab = "X", ylab = "Y")
points(x[high_group], y[high_group], pch = 16, col = rgb(0, 0, 1, 0.5))
abline(lm(y[low_group] ~ x[low_group]), col = "red", lwd = 2, lty = 2)
abline(lm(y[high_group] ~ x[high_group]), col = "blue", lwd = 2, lty = 2)
```




```r
par(mfrow = c(1, 1))

# Compare estimates
cat("Full data: slope =", round(coef(lm(y ~ x))[2], 3), "\n")
```

```
## Full data: slope = 0.726
```

```r
cat("Low group: slope =", round(coef(lm(y[low_group] ~ x[low_group]))[2], 3), "\n")
```

```
## Low group: slope = 0.724
```

```r
cat("High group: slope =", round(coef(lm(y[high_group] ~ x[high_group]))[2], 3), "\n")
```

```
## High group: slope = 0.72
```

Checking Parameter Invariance in IRT

```r
# Split sample by total score
total_scores <- rowSums(forma)
median_score <- median(total_scores)
```

```r
low_ability <- forma[total_scores <= median_score, ]
high_ability <- forma[total_scores > median_score, ]

cat("Low ability group: n =", nrow(low_ability), "\n")
```

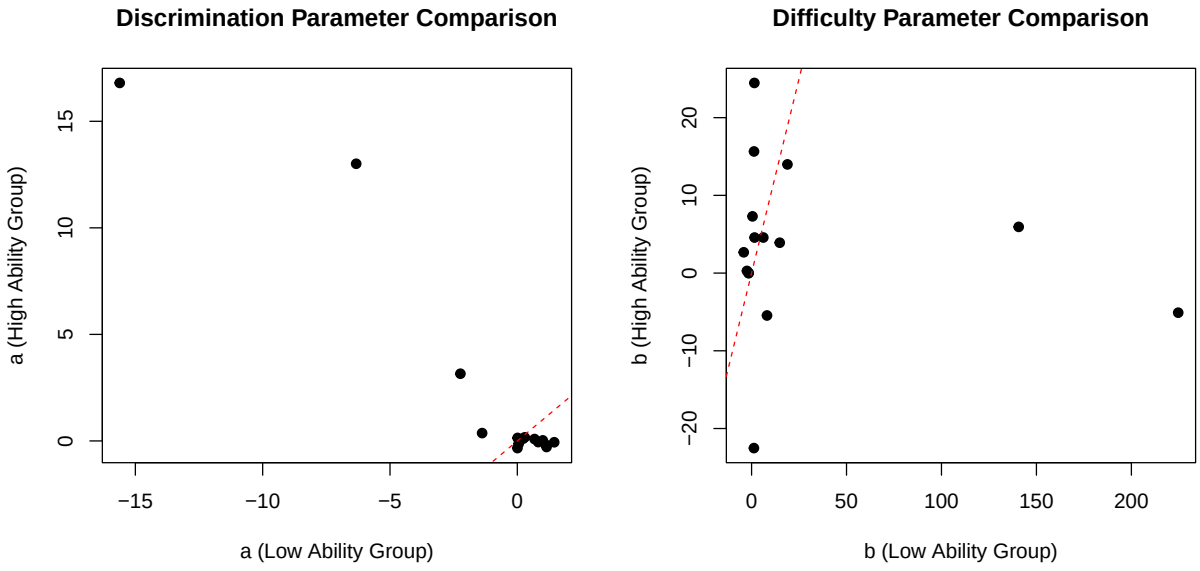
```
## Low ability group: n = 1013
```

```r
cat("High ability group: n =", nrow(high_ability), "\n")
```

```
## High ability group: n = 945
```

```r
# Fit models separately (note: this is for illustration - real invariance testing is more complex)
mod_low <- mirt(low_ability, 1, itemtype = "2PL", verbose = FALSE)
mod_high <- mirt(high_ability, 1, itemtype = "2PL", verbose = FALSE)
```

```r
# Extract parameters
coef_low <- coef(mod_low, simplify = TRUE, IRTpars = TRUE)$items
coef_high <- coef(mod_high, simplify = TRUE, IRTpars = TRUE)$items

par(mfrow = c(1, 2))

# Compare discrimination parameters
plot(coef_low[, 1], coef_high[, 1], pch = 19,
     xlab = "a (Low Ability Group)", ylab = "a (High Ability Group)",
     main = "Discrimination Parameter Comparison")
abline(0, 1, col = "red", lty = 2)

# Compare difficulty parameters
plot(coef_low[, 2], coef_high[, 2], pch = 19,
     xlab = "b (Low Ability Group)", ylab = "b (High Ability Group)",
     main = "Difficulty Parameter Comparison")
abline(0, 1, col = "red", lty = 2)
```



```r
par(mfrow = c(1, 1))
```

Note: Perfect invariance is rarely achieved in practice due to: - Sampling error - Scale differences between groups - Potential model misfit

---

Property 2: Scale Indeterminacy

The Problem

Because we cannot observe a, b, c, or θ directly, and they occur together in the model, there is **inherent indeterminacy** in the scale.

Consider the expression $a_i(\theta_p - b_i)$.

Now define new values:

- $b_i^* = (b_i + k_1)/k_2$
- $\theta_p^* = (\theta_p + k_1)/k_2$
- $a_i^* = k_2 \times a_i$
- $c_i^* = c_i$

It will be true that: $a_i(\theta_p - b_i) = a_i^*(\theta_p^* - b_i^*)$

So which set of parameters should we use?

Demonstration

```r
# Original parameters
a <- 1.5
b <- 0.5
theta <- 1.0

# Original value
original <- a * (theta - b)
cat("Original: a*(theta - b) =", a, "* (", theta, "-", b, ") =", original, "\n\n")
```

```
## Original: a*(theta - b) = 1.5 * ( 1 - 0.5 ) = 0.75
```

```r
# Transform with k1 = 1, k2 = 2
k1 <- 1
k2 <- 2

a_star <- k2 * a
b_star <- (b + k1) / k2
theta_star <- (theta + k1) / k2

transformed <- a_star * (theta_star - b_star)
cat("Transformed parameters:\n")
```

```
## Transformed parameters:
```

```
cat("a* =", a_star, "\n")
```

```
## a* = 3
```

```
cat("b* =", b_star, "\n")
```

```
## b* = 0.75
```

```
cat("theta* =", theta_star, "\n")
```

```
## theta* = 1
```

```
cat("a**(theta* - b*) =", a_star, "* (", theta_star, "-", b_star, ") =", transformed, "\n")
```

```
## a**(theta* - b*) = 3 * ( 1 - 0.75 ) = 0.75
```

Resolving Scale Indeterminacy

There are two main approaches:

1. Item-Side Anchoring

- Place constraints on item parameters (e.g., mean item difficulty = 0)
- Freely estimate the θ distribution
- More common in Rasch modeling and in Europe

2. Person-Side Anchoring

- Fix distribution of θ to have mean = 0 and SD = 1
- Freely estimate item parameters
- More common in US and in achievement testing

```
# Example: Different software may use different anchoring
# mirt uses person-side anchoring by default

# Get theta estimates
theta_estimates <- fscores(mod, method = "EAP")

cat("Theta distribution (person-side anchored):\n")
```

```
## Theta distribution (person-side anchored):
```

```
cat("Mean:", round(mean(theta_estimates), 3), "\n")
```

```
## Mean: 0.001
```

```r
cat("SD:", round(sd(theta_estimates), 3), "\n")
```

```
## SD: 0.92
```

```r
# Get item parameters
item_params <- coef(mod, simplify = TRUE, IRTpars = TRUE)$items

cat("\nItem difficulty (b) distribution:\n")
```

```
##
## Item difficulty (b) distribution:
```

```r
cat("Mean b:", round(mean(item_params[, 2]), 3), "\n")
```

```
## Mean b: 0.464
```

```r
cat("SD of b:", round(sd(item_params[, 2]), 3), "\n")
```

```
## SD of b: 0.573
```

Practical Implications

1. **Comparing across studies**: Must ensure same scale/anchoring
2. **Linking**: Need anchor items or anchor persons to equate scales
3. **Interpretation**: The absolute value of θ is arbitrary; only relative positions matter

---

Summary

Assumptions

Assumption	Description	Violation Consequences
Local Independence	Responses independent given θ	Biased parameter estimates
Dimensionality	Correct number of latent dimensions	Local dependence, poor fit
Functional Form	ICC follows logistic/normal ogive	Poor item fit
Continuous θ	Latent variable is continuous	(Rarely problematic)

Properties

Property	Description	Practical Implication
Parameter Invariance	Parameters same across groups	Enables fair comparison
Scale Indeterminacy	Scale defined up to linear transform	Need anchoring conventions

Key Takeaways

1. **Local independence** is crucial - violations can seriously bias estimates
2. **Dimensionality** should be checked before fitting unidimensional models
3. **Parameter invariance** is a property, not an assumption - it holds if model fits
4. **Scale indeterminacy** means we need conventions to interpret parameters
5. Always check model assumptions before interpreting results!