**Analysing a test with packages in R**

**Classical Test Theory**

**Item Difficulty**

#no new library has to be invoked these are part of base package

#use na.rm=true only if missing data present, otherwise drop

item\_diff<-colMeans(MTcsv\_noID*, na.rm=true*)

round(item\_diff, 3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V1 | V2 | V3 | V4 | V5 |
| 0.787 | 0.392 | 0.379 | 0.552 | 0.571 |

**Item Discrimination**

##first create a total score for each person

total\_score<-rowSums(MTcsv\_noID)

##correlate each item with the total score

item\_discr<-cor(MTcsv\_noID, total\_score)

##display item discrimination values

item\_discr

[,1]

V1 0.012824108

V2 0.535991439

V3 0.322218523

V4 0.120527496

V5 0.094385108….

**Test Statistics**

## invoke psych package in R

library(psych)

#this command calculates cronbach’s alpha for the whole data set as one factor

> alpha(MTcsv\_noID, keys=NULL,cumulative=FALSE, title=NULL, max=10,na.rm = TRUE, check.keys=FALSE,n.iter=1,delete=TRUE,use="pairwise",warnings=TRUE,n.obs=NULL)

Some items ( V1 V4 V5 V6 V7 V9 V11 V12 V13 V14 V15 V17 V20 V21 V23 V25 V26 V28 V29 V33 V34 V39 V40 V42 V46 V48 V50 ) were negatively correlated with the total scale and probably should be reversed. To do this, run the function again with the 'check.keys=TRUE' option

Reliability analysis

raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r

0.71 0.68 0.83 0.042 2.2 0.02 0.48 0.12 -0.012

lower alpha upper 95% confidence boundaries

0.67 0.71 0.75

Reliability if an item is dropped:

raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r

V1 0.72 0.69 0.84 0.044 2.3 0.020 0.086 -0.012

V2 0.69 0.66 0.82 0.039 2.0 0.022 0.080 -0.012

V3 0.70 0.68 0.83 0.041 2.1 0.021 0.085 -0.013

V4 0.72 0.69 0.83 0.043 2.2 0.020 0.082 -0.012

V5 0.72 0.69 0.83 0.043 2.2 0.020 0.082 -0.012

Item statistics

n raw.r std.r r.cor r.drop mean sd

V1 375 0.0128 0.023 -0.0496 -0.0560 0.787 0.410

V2 375 0.5360 0.508 0.5329 0.4736 0.392 0.489

V3 375 0.3222 0.312 0.2790 0.2464 0.379 0.486

V4 375 0.1205 0.131 0.1043 0.0372 0.552 0.498

V5 375 0.0944 0.111 0.0825 0.0112 0.571 0.496….

#note the warnings on items that are negatively correlated; in CTT those items should be deleted and run again.

# rerun with inverse discrimination items reversed; don’t do this unless you KNOW that the item was negatively loaded.

Call: alpha(x = MTcsv\_noID, keys = NULL, cumulative = FALSE, title = NULL,

max = 10, na.rm = TRUE, check.keys = TRUE, n.iter = 1, delete = TRUE,

use = "pairwise", warnings = TRUE, n.obs = NULL)

raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r

0.94 0.93 0.96 0.21 14 0.0043 0.49 0.23 0.16

lower alpha upper 95% confidence boundaries

0.93 0.94 0.95

Reliability if an item is dropped:

raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r

V1- 0.94 0.93 0.96 0.22 14 0.0043 0.039 0.17

V2 0.93 0.93 0.95 0.21 13 0.0045 0.037 0.15

V3 0.94 0.93 0.96 0.22 14 0.0043 0.039 0.16

V4- 0.94 0.93 0.95 0.21 13 0.0044 0.039 0.16

V5- 0.94 0.93 0.95 0.21 13 0.0044 0.039 0.16

Item statistics

n raw.r std.r r.cor r.drop mean sd

V1- 375 0.111 0.114 0.081 0.075 0.21 0.410

V2 375 0.735 0.735 0.740 0.715 0.39 0.489

V3 375 0.332 0.330 0.307 0.293 0.38 0.486

V4- 375 0.531 0.532 0.525 0.499 0.45 0.498

V5- 375 0.563 0.566 0.560 0.532 0.43 0.496

Non missing response frequency for each item

0 1 miss

V1 0.21 0.79 0

V2 0.61 0.39 0

V3 0.62 0.38 0

V4 0.45 0.55 0

V5 0.43 0.57 0

Warning message:

In alpha(MTcsv\_noID, keys = NULL, cumulative = FALSE, title = NULL, :

Some items were negatively correlated with total scale and were automatically reversed.

This is indicated by a negative sign for the variable name.

#test info

describe(total\_score)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | M | SD | median | min | max | skew | kurtosis | se |
| 375 | 23.78 | 5.96 | 23 | 7 | 41 | 0.45 | -0.29 | 0.31 |

SEM=SD\*√(1-reliability); =5.96\*√(1-.712)=5.96\*.537=3.20

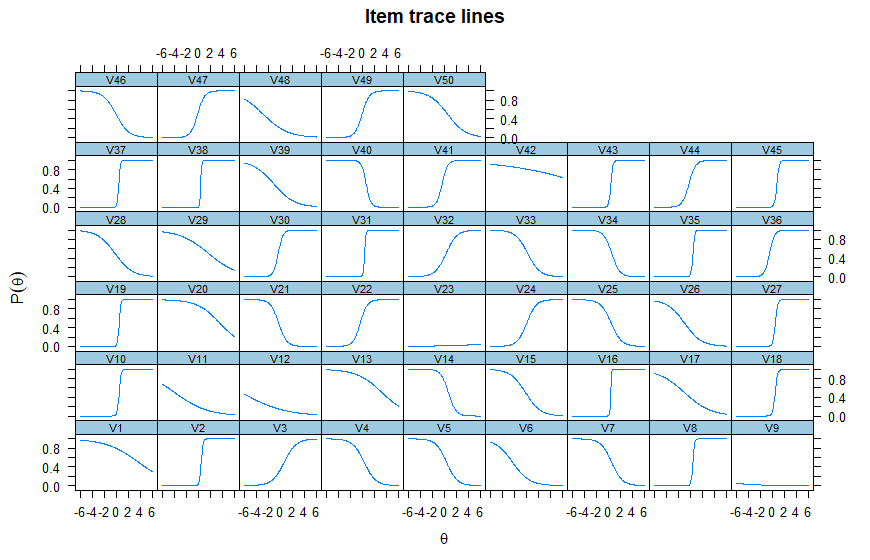
## item characteristic curves

x <- mirt(MTcsv\_noID, 1, SE=TRUE)

Iteration: 301, Log-Lik: -8700.584, Max-Change: 0.00010

plot(x)

plot(x, type = 'trace')



**QUESTION**

Do trace lines match the items negatively correlated with the total scale?

**1 parameter Rasch**

Library(ltm)

response\_matrix = (XXXX)

# 1 parameter Rasch analysis of dichotomous data file; Note na.action=null means ignore any item responses coded na for missing values

item\_weights <- rasch(response\_matrix, na.action = NULL)

##to create Rasch parameters and test-taker location scores calculate the values for each; note: "resp.patterns = response\_matrix" binds the ordering of the students and ensures they're not sorted.

locations <- factor.scores(item\_weights, resp.patterns = response\_matrix)

##to display the discrimination coefficients for each item; these should all be the same in RASCH; (nb negative bad):

item\_weights[1]$coefficients[,2]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V1 | V2 | V3 | V4 | V5 |
| 0.4943342 | 0.4943342 | 0.4943342 | 0.4943342 | 0.4943342 |

## to display the Item weights or locations (b parameter):

item\_weights[1]$coefficients[,1] / item\_weights[1]$coefficients[,2] \* -1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V1 | V2 | V3 | V4 | V5 |
| -2.775158399 | 0.934183524 | 1.054175386 | -0.452468592 | -0.613780719 |

#calculate person scores based on Rasch item analysis; to get student locations on theta (negative means weak, positive means strong); these values now match the order of cases as input.

locations$score.dat["z1"]

z1

1 -1.67902284

2 1.28893716

3 1.58109032

4 2.03496864

5 1.00293723

6 1.73017668

7 0.72156051

8 0.86176266

9 1.28893716

> #obtain Rasch fit statistics and difficulty locations for items

> summary(item\_weights)

Call:

rasch(data = response\_matrix, na.action = NULL)

Model Summary:

log.Lik AIC BIC

-11240.15 22582.3 22782.57

Coefficients:

value std.err z.vals

Dffclt.V1 -2.7752 0.2994 -9.2694

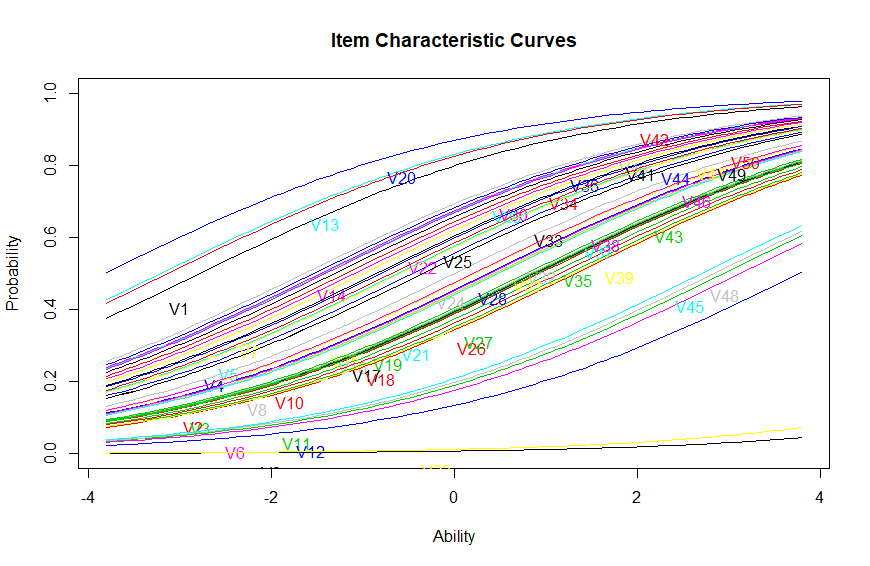
Dffclt.V2 0.9342 0.2309 4.0464

Dffclt.V3 1.0542 0.2335 4.5152

NOTE: if z>1.96, then item does NOT fit the Rasch model and should be rejected

> #plot ICCs

> plot(fitRasch)



#generate item responses and logit scores per person; z1 is the person score, se.z1 is the standard error for the score; note items deleted from display

factor.scores(item\_weights)

Call:

rasch(data = response\_matrix, na.action = NULL)

Scoring Method: Empirical Bayes

Factor-Scores for observed response patterns:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| It | V1 | V2 | V3 | V4 | V5 | V6 | … | z1 | se.z1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |  | -0.805 | 0.534 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 |  | -1.090 | 0.539 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 |  | -0.805 | 0.534 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 |  | -1.090 | 0.539 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |  | -1.090 | 0.539 |

**2 parameter IRT**

(data file, items only scored 0,1)

> MTcsv\_noID <- read.csv("H:/00 All PUBS/Assessment/work/2017 Brown & Abdulnabi MCQ in HE/current data/MTcsv\_noID.txt", header=FALSE)

> View(MTcsv\_noID)

## invoke ltm package in R

library(ltm)

## Descriptive information for XXXX data where XXXX=data file name

dsc <- descript(XXXX, 3)

dsc

Descriptive statistics for the 'MTcsv\_noID' data-set

Sample:

50 items and 375 sample units; 0 missing values

Proportions for each level of response:

0 1 logit

V1 0.2133 0.7867 1.3049

V2 0.6080 0.3920 -0.4389

V3 0.6213 0.3787 -0.4952

…

Point Biserial correlation with Total Score:

Included Excluded

V1 0.0128 -0.0560

V2 0.5360 0.4736

V3 0.3222 0.2464

….

Cronbach's alpha:

value

All Items 0.7123

Excluding V1 0.7183

Excluding V2 0.6928

Excluding V3 0.7049

….

##to create a matrix that won’t change order from the data set

response\_matrix = (XXXX)

## run the 2PL analysis on the data set; Note na.action=null means ignore any item responses coded na for missing values

item\_weights <- ltm(response\_matrix ~ z1, na.action = NULL)

##to create 2PL parameters and test-taker location scores calculate the values for each; note: "resp.patterns = response\_matrix" binds the ordering of the students and ensures they're not sorted.

locations <- factor.scores(item\_weights, resp.patterns = response\_matrix)

##to display the discrimination coefficients (a parameter) for each item (nb negative bad):

item\_weights[1]$coefficients[,2]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V1 | V2 | V3 | V4 | V5 |
| 0.3618220 | -6.6244443 | -0.9466713 | 1.1649277 | 1.1900982 |

## to display the Item weights or locations (b parameter):

item\_weights[1]$coefficients[,1] / item\_weights[1]$coefficients[,2] \* -1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V1 | V2 | V3 | V4 | V5 |
| -3.158484689 | 0.124698553 | -0.096567721 | 0.275685030 | 0.199588221 |

## to get student locations on theta (negative means weak, positive means strong); these values now match the order of cases as input.

locations$score.dat["z1"]

z1

1 0.2104774892

2 0.0252190294

3 -0.0443625504

4 -0.0948245513

5 0.1039349259

6 0.0352090683

7 0.0457723256

8 0.0720680887

9 0.0149202265

## To display all parameters at once and get fit indices

> summary(item\_weights)

Call:

ltm(formula = response\_matrix ~ z1, na.action = NULL)

Model Summary:

log.Lik AIC BIC

-8726.661 17653.32 18046.01

Coefficients:

value std.err z.vals

Dffclt.V1 -3.1585 1.2184 -2.5923

Dffclt.V2 0.1247 0.0391 3.1927

Dffclt.V3 -0.0966 0.1272 -0.7594

…

Dscrmn.V1 0.3618 0.1510 2.3957

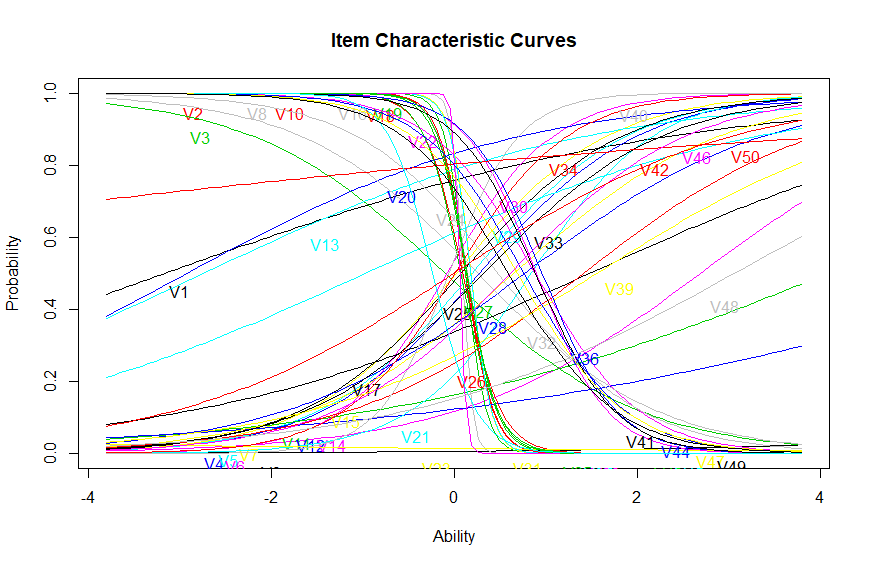
Dscrmn.V2 -6.6244 1.0467 -6.3286

Dscrmn.V3 -0.9467 0.1571 -6.0241

….

##Draw ICC for XXXX

plot(item\_weights)



**Model selection**

Compare AIC values. Smaller is better

To calculate relative merit of models using AIC:

(1) find difference to AIC Min;

(2) calculate exponent of (-0.5\*difference);

(3) Sum all weights;

(4) determine proportion of sum for each model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Models** | **AIC** | **Δi (diff AIC minus Min)** | **Weight AIC (exponent)** | **w*i* (proportion of weight)** | **Sum of w*i*** |
| Minimum AIC (2PL) | 17653.3 | 0.00 | 1.00 | 1.0000 | 1.00 |
| Comparison AIC (Rasch) | 22582.3 | 4928.98 | 0.00 | 0.0000 | 1.00 |
| sum weights |  |  | 1.00 |  |  |

**Interpretation**

1. As a rule of thumb, a ∆ i < 2 suggests substantial evidence for the **comparison** model, values between 3 and 7 indicate that the **comparison** model has considerably less support, whereas a ∆ i > 10 indicates that the **comparison** model is very unlikely (Burnham and Anderson 2002).

2. Based on the Akaike weights (wi). To obtain a 95% conﬁdence set on the actual K-L best model, summing the Akaike weights from largest to smallest until that sum is just ≥ models is the conﬁdence set on the K-L best model.0.95, and the corresponding subset of

Procedure taken from: <http://avesbiodiv.mncn.csic.es/estadistica/ejemploaic.pdf>

Interpretation taken from: <http://www4.ncsu.edu/~shu3/Presentation/AIC.pdf>