R Notebook

Zoe Berling & Daniel Valverde

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
library(ltm)

## Warning: package 'ltm' was built under R version 4.0.5

## Loading required package: MASS

## Loading required package: msm

## Warning: package 'msm' was built under R version 4.0.5

## Loading required package: polycor

## Warning: package 'polycor' was built under R version 4.0.5

library(mokken)

## Warning: package 'mokken' was built under R version 4.0.5

## Loading required package: poLCA

## Warning: package 'poLCA' was built under R version 4.0.5

## Loading required package: scatterplot3d

library(car)

## Warning: package 'car' was built under R version 4.0.5

## Loading required package: carData

## Error : 'format_warning' is not an exported object from 'namespace:cli'

## Warning: replacing previous import 'ellipsis::check_dots_unnamed' by
## 'rlang::check_dots_unnamed' when loading 'hms'

## Warning: replacing previous import 'ellipsis::check_dots_used' by
## 'rlang::check_dots_used' when loading 'hms'

## Warning: replacing previous import 'ellipsis::check_dots_empty' by
## 'rlang::check_dots_empty' when loading 'hms'
```

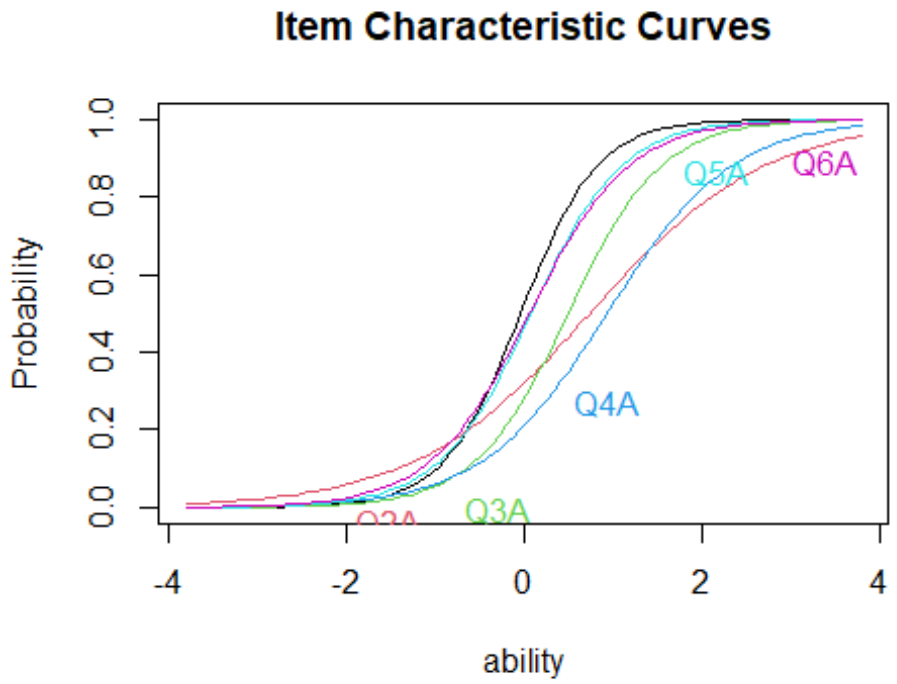
```
## Warning: replacing previous import 'ellipsis::check_dots_unnamed' by
## 'rlang::check_dots_unnamed' when loading 'tibble'

## Warning: replacing previous import 'ellipsis::check_dots_used' by
## 'rlang::check_dots_used' when loading 'tibble'

## Warning: replacing previous import 'ellipsis::check_dots_empty' by
## 'rlang::check_dots_empty' when loading 'tibble'

## Warning: replacing previous import 'ellipsis::check_dots_unnamed' by
## 'rlang::check_dots_unnamed' when loading 'pillar'

## Warning: replacing previous import 'ellipsis::check_dots_used' by
## 'rlang::check_dots_used' when loading 'pillar'

## Warning: replacing previous import 'ellipsis::check_dots_empty' by
## 'rlang::check_dots_empty' when loading 'pillar'

##
## Attaching package: 'car'

## The following object is masked from 'package:mokken':
##
##     recode

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.0 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.8
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::recode() masks car::recode(), mokken::recode()
## x dplyr::select() masks MASS::select()
## x purrr::some()   masks car::some()

library(latticeExtra)

## Warning: package 'latticeExtra' was built under R version 4.0.5

## Loading required package: lattice
```

```
##
## Attaching package: 'latticeExtra'

## The following object is masked from 'package:ggplot2':
##
##     layer

data <- read.csv("q1Throughq6andGender.csv", header=TRUE)
data<- data[c('Q1A', 'Q2A', 'Q3A', 'Q4A', 'Q5A', 'Q6A', 'gender')]
head(data)

##   Q1A Q2A Q3A Q4A Q5A Q6A gender
## 1   4   4   2   4   4   4      2
## 2   4   1   2   3   4   4      2
## 3   3   1   4   1   4   3      2
## 4   2   3   2   1   3   3      2
## 5   2   2   3   4   4   2      2
## 6   1   1   2   1   3   1      2

data$gender[data$gender==1]<-0
data$gender[data$gender==2]<-1
# converting the data to binary for dichotomous purposes
questions = c('Q1A', 'Q2A', 'Q3A', 'Q4A', 'Q5A', 'Q6A')

for (c in questions) {
data[[c]] <- car::recode(data[[c]],"c(1, 2)='0';c(3, 4)='1'")

}

head(data)

##   Q1A Q2A Q3A Q4A Q5A Q6A gender
## 1   1   1   0   1   1   1      1
## 2   1   0   0   1   1   1      1
## 3   1   0   1   0   1   1      1
## 4   0   1   0   0   1   1      1
## 5   0   0   1   1   1   0      1
## 6   0   0   0   0   1   0      1

dat_base <-data[c('Q1A', 'Q2A', 'Q3A', 'Q4A', 'Q5A', 'Q6A')]
head(dat_base)

##   Q1A Q2A Q3A Q4A Q5A Q6A
## 1   1   1   0   1   1   1
## 2   1   0   0   1   1   1
## 3   1   0   1   0   1   1
## 4   0   1   0   0   1   1
## 5   0   0   1   1   1   0
## 6   0   0   0   0   1   0
```

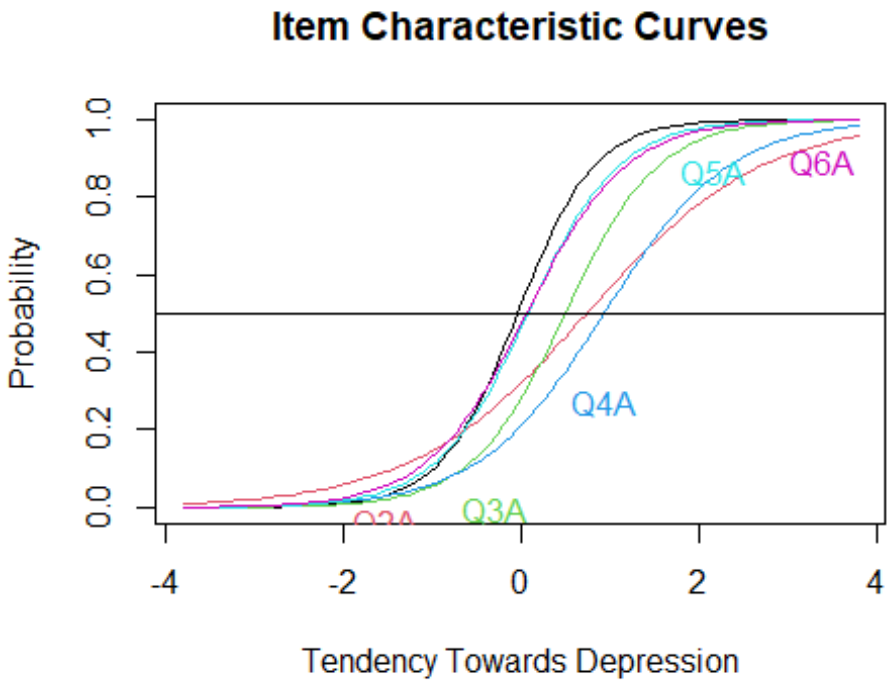
```r
datM <- data[data$gender == 0,]
datM <- datM[c('Q1A', 'Q2A', 'Q3A', 'Q4A', 'Q5A', 'Q6A')]
datF <- data[data$gender == 1,]
datF <- datF[c('Q1A', 'Q2A', 'Q3A', 'Q4A', 'Q5A', 'Q6A')]

#at first glance the mean of each question seems to be good because not
everyone is answering one way to these wquestions
summary(dat_base)
```

```
##       Q1A              Q2A              Q3A              Q4A
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.5117   Mean   :0.3491   Mean   :0.3558   Mean   :0.2738
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##       Q5A              Q6A
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000
##  Mean   :0.4769   Mean   :0.4817
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

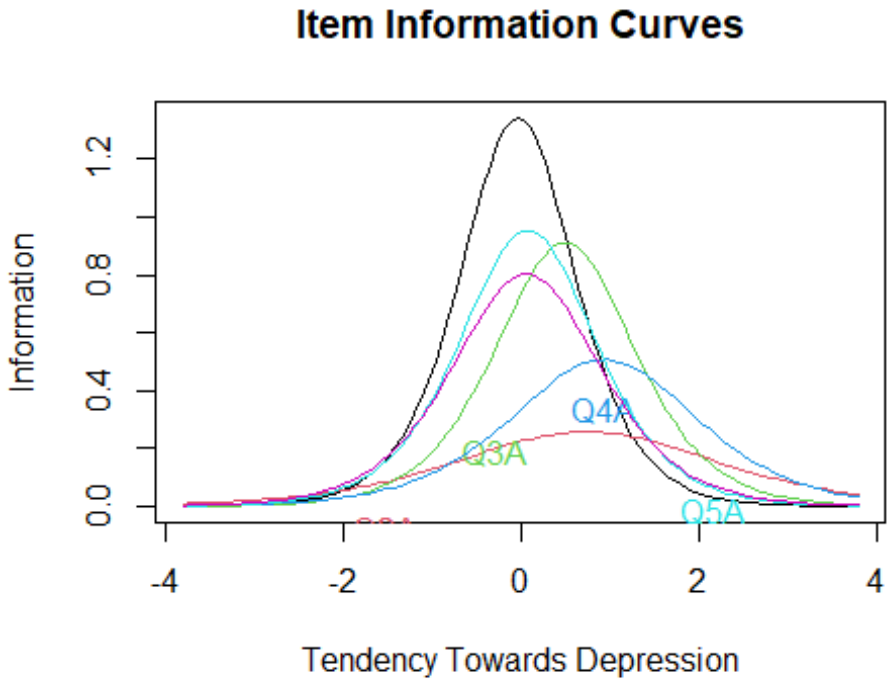
```r
#testing for monotonicity
#z1 implies that we have one latent variable that we are predicting
#IRT.param=TRUE is how we set this IRT Model to 2PL (difficulty and
discrimination)
model<-ltm(dat_base~z1, IRT.param=TRUE)

# the steeper the slope the more
## discriminable an item is
plot(model, type="ICC", xlab='ability')
```




```
#difficulty is a zcore of whatever the latent variable is (anxiety)
#discriminable is the slope or how good the question is at figuring a person
out
model<-ltm(dat_base~z1, IRT.param=TRUE)
print(coef(model))

##          Dffclt   Dscrmn
## Q1A -0.03346154 2.317070
## Q2A  0.74285394 1.016176
## Q3A  0.49802722 1.908842
## Q4A  0.93501859 1.423853
## Q5A  0.07939943 1.953463
## Q6A  0.06521985 1.790820

plot(model, type="ICC", items=c(1,2,3,4,5, 6), xlab='Tendency Towards
Depression')
abline(.5,0)
```




```r
plot(model, type="IIC", xlab='Tendency Towards Depression') #test information function
```



```
#gives you every combination of response patterns

factor.scores(model)

##
## Call:
## ltm(formula = dat_base ~ z1, IRT.param = TRUE)
##
## Scoring Method: Empirical Bayes
##
## Factor-Scores for observed response patterns:
##    Q1A Q2A Q3A Q4A Q5A Q6A  Obs      Exp     z1 se.z1
## 1    0   0   0   0   0   0 8336 8061.059 -1.001 0.617
## 2    0   0   0   0   0   1 1421 1595.205 -0.472 0.486
## 3    0   0   0   0   1   0 1407 1406.395 -0.435 0.479
## 4    0   0   0   0   1   1  514  733.086 -0.067 0.435
## 5    0   0   0   1   0   0  478  592.888 -0.562 0.504
## 6    0   0   0   1   0   1  240  245.247 -0.169 0.443
## 7    0   0   0   1   1   0  216  228.727 -0.137 0.440
## 8    0   0   0   1   1   1  135  208.194  0.196 0.428
## 9    0   0   1   0   0   0  539  649.741 -0.445 0.481
## 10   0   0   1   0   0   1  204  332.363 -0.076 0.435
## 11   0   0   1   0   1   0  729  315.288 -0.045 0.433
## 12   0   0   1   0   1   1  390  343.448  0.285 0.431
## 13   0   0   1   1   0   0   78  104.066 -0.146 0.441
## 14   0   0   1   1   0   1   51   93.161  0.188 0.428
## 15   0   0   1   1   1   0  164   92.694  0.218 0.429
## 16   0   0   1   1   1   1  144  172.037  0.559 0.450
## 17   0   1   0   0   0   0 1325 1419.015 -0.671 0.529
## 18   0   1   0   0   0   1  496  484.756 -0.250 0.452
## 19   0   1   0   0   1   0  416  445.380 -0.218 0.448
## 20   0   1   0   0   1   1  224  347.805  0.121 0.428
## 21   0   1   0   1   0   0  262  163.498 -0.327 0.462
## 22   0   1   0   1   0   1  190  103.961  0.024 0.430
## 23   0   1   0   1   1   0  128  100.407  0.054 0.429
## 24   0   1   0   1   1   1  160  133.124  0.385 0.436
## 25   0   1   1   0   0   0  142  203.466 -0.226 0.449
## 26   0   1   1   0   0   1   98  156.215  0.113 0.428
## 27   0   1   1   0   1   0  312  153.332  0.143 0.428
## 28   0   1   1   0   1   1  180  243.723  0.478 0.442
## 29   0   1   1   1   0   0   43   45.249  0.046 0.429
## 30   0   1   1   1   0   1   57   59.007  0.376 0.435
## 31   0   1   1   1   1   0  152   60.781  0.407 0.437
## 32   0   1   1   1   1   1  193  168.497  0.777 0.477
## 33   1   0   0   0   0   0 1329 1496.169 -0.353 0.466
## 34   1   0   0   0   0   1 1389  905.690  0.001 0.431
## 35   1   0   0   0   1   0  650  871.137  0.031 0.430
## 36   1   0   0   0   1   1 1190 1103.485  0.362 0.434
## 37   1   0   0   1   0   0  206  274.704 -0.068 0.435
## 38   1   0   0   1   0   1  408  286.120  0.263 0.430
```

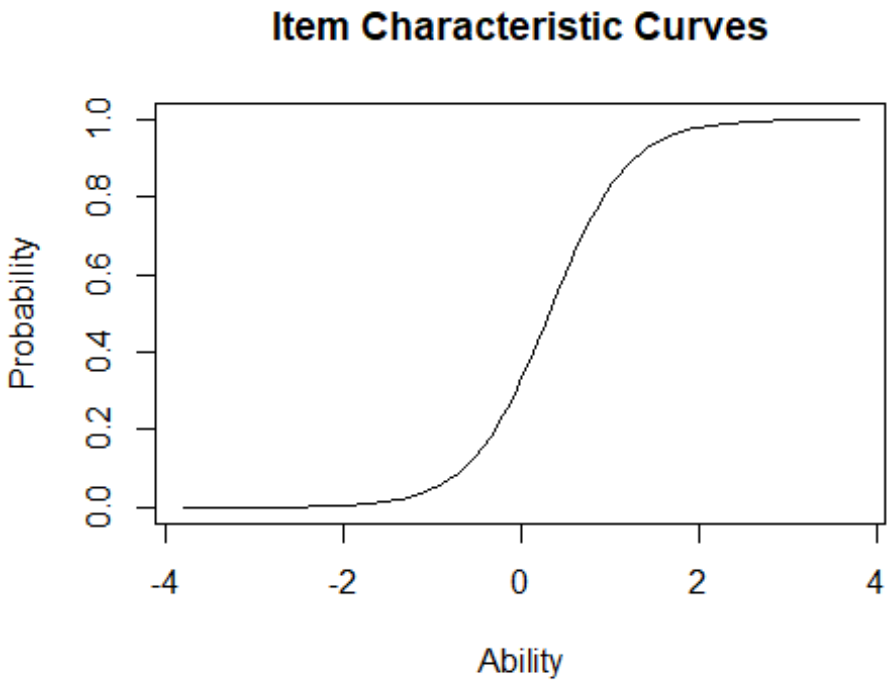
```
## 39   1   0   0   1   1   0  199  288.631  0.293 0.431
## 40   1   0   0   1   1   1  502  627.505  0.644 0.459
## 41   1   0   1   0   0   0  244  393.023  0.023 0.430
## 42   1   0   1   0   0   1  500  489.689  0.353 0.434
## 43   1   0   1   0   1   0  683  502.266  0.384 0.436
## 44   1   0   1   0   1   1 1800 1324.440  0.749 0.473
## 45   1   0   1   1   0   0   77  128.527  0.285 0.431
## 46   1   0   1   1   0   1  194  274.591  0.634 0.458
## 47   1   0   1   1   1   0  248  296.457  0.669 0.463
## 48   1   0   1   1   1   1 1224 1454.930  1.106 0.533
## 49   1   1   0   0   0   0  500  517.788 -0.146 0.441
## 50   1   1   0   0   0   1  663  463.626  0.188 0.428
## 51   1   1   0   0   1   0  337  461.311  0.218 0.429
## 52   1   1   0   0   1   1  716  856.360  0.560 0.450
## 53   1   1   0   1   0   0  202  130.238  0.121 0.428
## 54   1   1   0   1   0   1  410  197.776  0.455 0.441
## 55   1   1   0   1   1   0  234  206.667  0.486 0.443
## 56   1   1   0   1   1   1  742  679.121  0.873 0.492
## 57   1   1   1   0   0   0  139  206.199  0.210 0.428
## 58   1   1   1   0   0   1  271  376.292  0.551 0.449
## 59   1   1   1   0   1   0  381  400.139  0.584 0.453
## 60   1   1   1   0   1   1 1360 1625.403  0.995 0.513
## 61   1   1   1   1   0   0   78   91.141  0.478 0.442
## 62   1   1   1   1   0   1  238  293.882  0.862 0.490
## 63   1   1   1   1   1   0  380  330.550  0.901 0.496
## 64   1   1   1   1   1   1 2857 2759.328  1.431 0.601
```

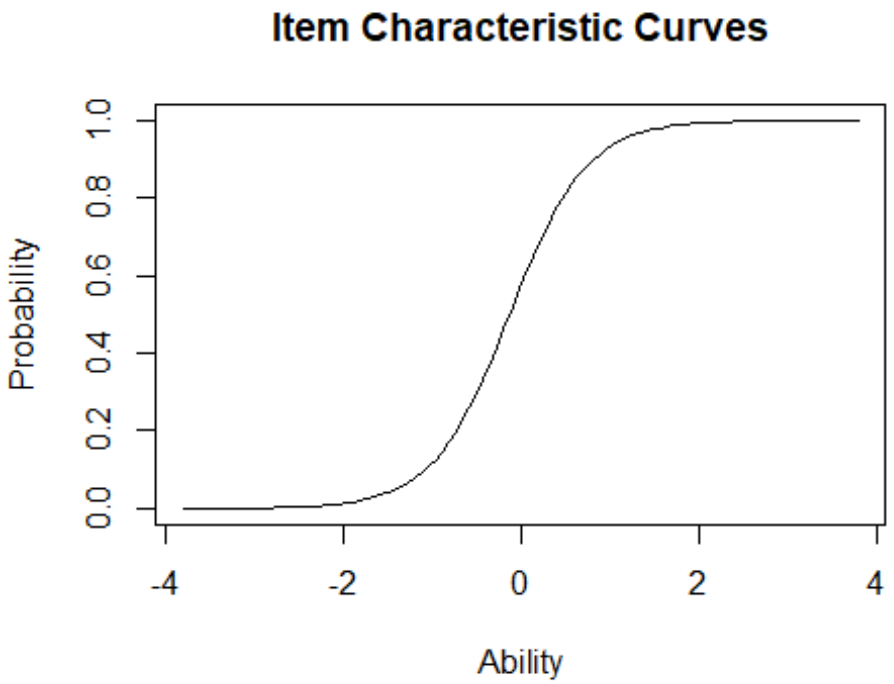
```
#measures person ability for each pearson who responded a specific way
person.fit(model)
```

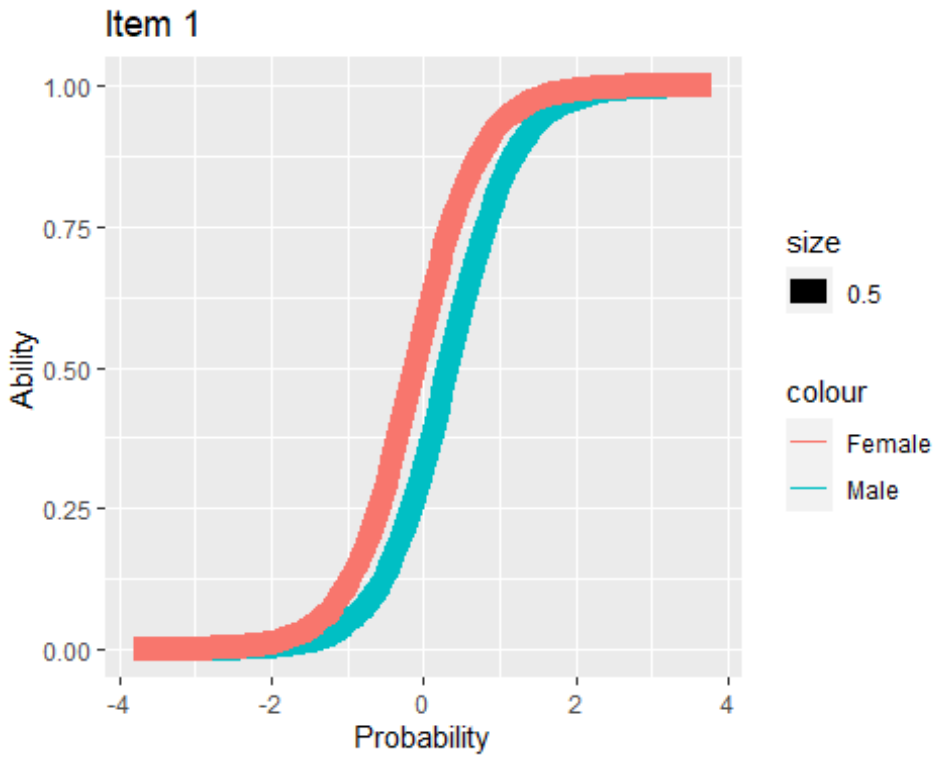
```
##
## Person-Fit Statistics and P-values
##
## Call:
## ltm(formula = dat_base ~ z1, IRT.param = TRUE)
##
## Alternative: Inconsistent response pattern under the estimated model
##
##    Q1A Q2A Q3A Q4A Q5A Q6A      L0      Lz Pr(<Lz)
## 1    0   0   0   0   0   0 -0.6278  0.7994   0.788
## 2    0   0   0   0   0   1 -2.4153  0.4719  0.6815
## 3    0   0   0   0   1   0 -2.5445  0.4472  0.6726
## 4    0   0   0   0   1   1 -3.1919  0.6408  0.7392
## 5    0   0   0   1   0   0 -3.3937 -0.4043   0.343
## 6    0   0   0   1   0   1 -4.2938 -0.7237  0.2346
## 7    0   0   0   1   1   0 -4.3618 -0.7762  0.2188
## 8    0   0   0   1   1   1 -4.4135 -0.7438  0.2285
## 9    0   0   1   0   0   0 -3.3159 -0.1785  0.4292
## 10   0   0   1   0   0   1 -3.9836 -0.3027   0.381
## 11   0   0   1   0   1   0 -4.0336 -0.3324  0.3698
```

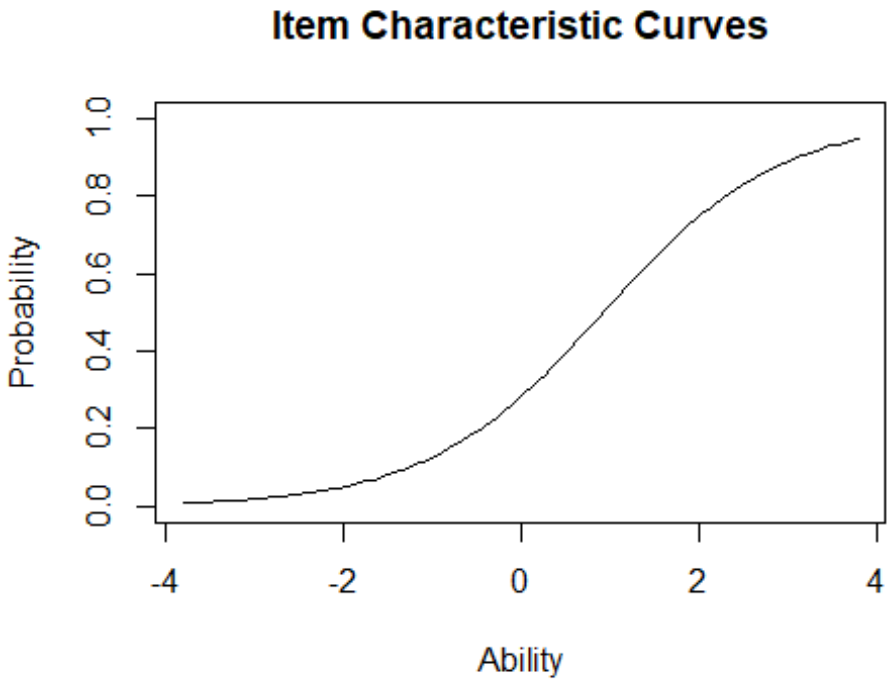
```
## 12   0   0   1   0   1   1 -3.8947  0.0272  0.5109
## 13   0   0   1   1   0   0 -5.1498 -1.6241  0.0522
## 14   0   0   1   1   0   1 -5.2192 -1.9399  0.0262
## 15   0   0   1   1   1   0 -5.2185 -1.9413  0.0261
## 16   0   0   1   1   1   1 -4.5118 -0.9074  0.1821
## 17   0   1   0   0   0   0 -2.4993  0.0646  0.5257
## 18   0   1   0   0   0   1 -3.6148 -0.1411  0.4439
## 19   0   1   0   0   1   0 -3.6989 -0.1744  0.4308
## 20   0   1   0   0   1   1 -3.9136 -0.0347  0.4862
## 21   0   1   0   1   0   0 -4.7015 -1.1988  0.1153
## 22   0   1   0   1   0   1 -5.1354 -1.7229  0.0425
## 23   0   1   0   1   1   0 -5.1662 -1.7881  0.0369
## 24   0   1   0   1   1   1 -4.8190 -1.3039  0.0961
## 25   0   1   1   0   0   0 -4.4825 -0.9480  0.1716
## 26   0   1   1   0   0   1 -4.7153 -1.1862  0.1178
## 27   0   1   1   0   1   0 -4.7290 -1.2099  0.1132
## 28   0   1   1   0   1   1 -4.1888 -0.4750  0.3174
## 29   0   1   1   1   0   0 -5.9644 -2.8564  0.0021
## 30   0   1   1   1   0   1 -5.6347 -2.4455  0.0072
## 31   0   1   1   1   1   0 -5.5972 -2.3556  0.0092
## 32   0   1   1   1   1   1 -4.4445 -0.9540    0.17
## 33   1   0   0   0   0   0 -2.4869  0.6690  0.7483
## 34   1   0   0   0   0   1 -2.9734  1.0677  0.8572
## 35   1   0   0   0   1   0 -3.0086  1.0923  0.8626
## 36   1   0   0   0   1   1 -2.7098  1.6659  0.9521
## 37   1   0   0   1   0   0 -4.1735 -0.5188   0.302
## 38   1   0   0   1   0   1 -4.0822 -0.2466  0.4026
## 39   1   0   0   1   1   0 -4.0669 -0.2279  0.4099
## 40   1   0   0   1   1   1 -3.1873  0.4953  0.6898
## 41   1   0   1   0   0   0 -3.8056  0.0258  0.5103
## 42   1   0   1   0   0   1 -3.5243  0.5242  0.6999
## 43   1   0   1   0   1   0 -3.4914  0.5382  0.7048
## 44   1   0   1   0   1   1 -2.3957  1.0960  0.8635
## 45   1   0   1   1   0   0 -4.8777 -1.4218  0.0775
## 46   1   0   1   1   0   1 -4.0174 -0.3965  0.3459
## 47   1   0   1   1   1   0 -3.9273 -0.3347  0.3689
## 48   1   0   1   1   1   1 -2.0851  0.6663  0.7474
## 49   1   1   0   0   0   0 -3.5453  0.0927  0.5369
## 50   1   1   0   0   0   1 -3.6145  0.4408  0.6703
## 51   1   1   0   0   1   0 -3.6137  0.4504  0.6738
## 52   1   1   0   0   1   1 -2.9067  0.9804  0.8366
## 53   1   1   0   1   0   0 -4.8960 -1.4469   0.074
## 54   1   1   0   1   0   1 -4.4045 -0.7430  0.2287
## 55   1   1   0   1   1   0 -4.3514 -0.6852  0.2466
## 56   1   1   0   1   1   1 -3.0008  0.2821  0.6111
## 57   1   1   1   0   0   0 -4.4206 -0.7529  0.2258
## 58   1   1   1   0   0   1 -3.7321  0.0208  0.5083
## 59   1   1   1   0   1   0 -3.6594  0.0632  0.5252
## 60   1   1   1   0   1   1 -2.0536  0.8908  0.8135
## 61   1   1   1   1   0   0 -5.1726 -1.7290  0.0419
```

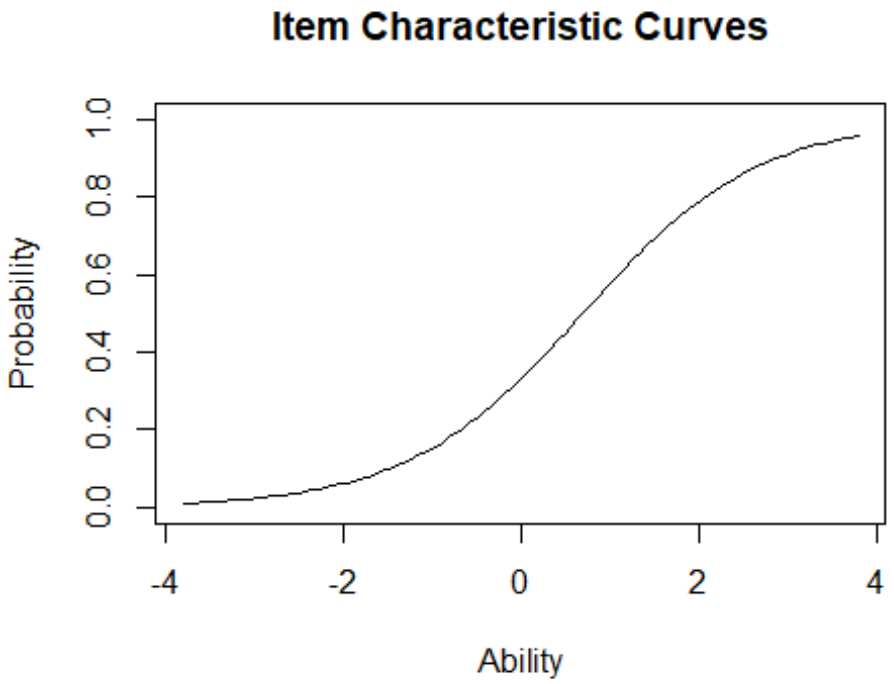
```
## 62   1   1   1   1   0   1 -3.8444 -0.4702  0.3191
## 63   1   1   1   1   1   0 -3.7044 -0.3910  0.3479
## 64   1   1   1   1   1   1 -1.1452  0.8861  0.8122

item.fit(model)

##
## Item-Fit Statistics and P-values
##
## Call:
## ltm(formula = dat_base ~ z1, IRT.param = TRUE)
##
## Alternative: Items do not fit the model
## Ability Categories: 10
##
##          X^2 Pr(>X^2)
## Q1A 3789.449  <0.0001
## Q2A 9243.511  <0.0001
## Q3A 1704.668  <0.0001
## Q4A 3804.557  <0.0001
## Q5A 2943.172  <0.0001
## Q6A 1302.295  <0.0001

library(glue)

##
## Attaching package: 'glue'

## The following object is masked from 'package:dplyr':
##
##     collapse

# graphing both genders ICC
modelM<-ltm(datM~z1, IRT.param=TRUE)
modelF<-ltm(datF~z1, IRT.param=TRUE)

for (i in 1:6) {
  g<- ggplot()
  p1 = as.data.frame(plot(modelM,type="ICC",item = {i}))
  p2= as.data.frame(plot(modelF,type="ICC", item = {i}))

  # p1<- as.data.frame(pm)
  # p2 <- as.data.frame(pf)

  g<-  g+ geom_line(aes(x=p1$z, y=p1$V2, color = 'Male', size = .5)) +
geom_line(aes(x=p2$z, y=p2$V2, color = 'Female', size=.5)) +
labs(title=glue("Item ",{i}) ,
          x ="Probability", y = "Ability")
print(g)
}
```

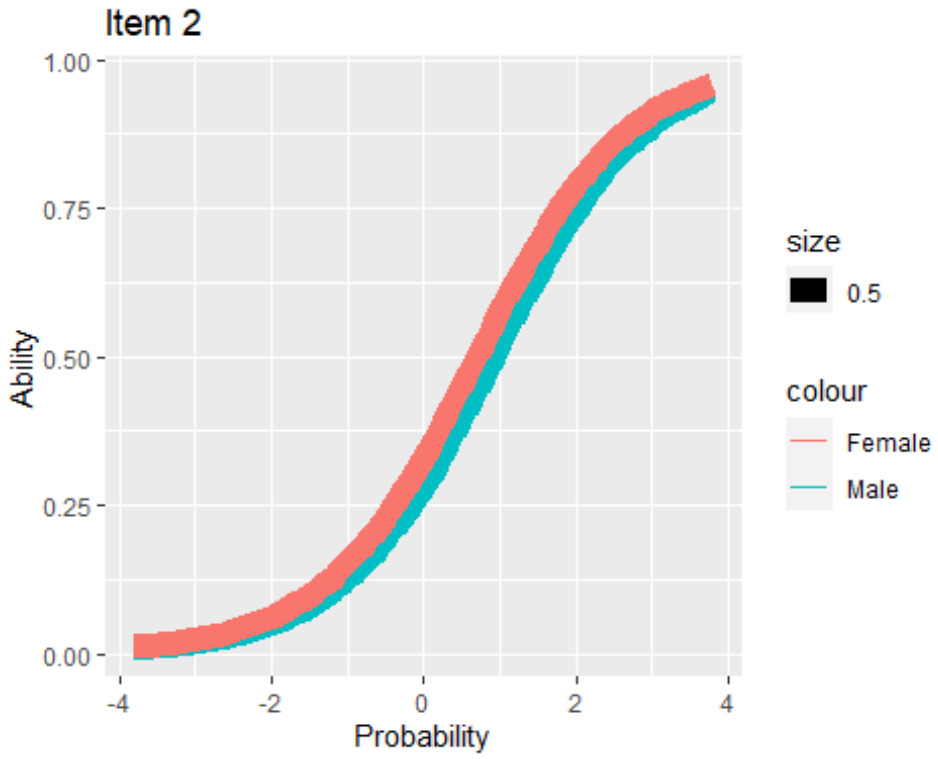

Item Characteristic Curves
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability

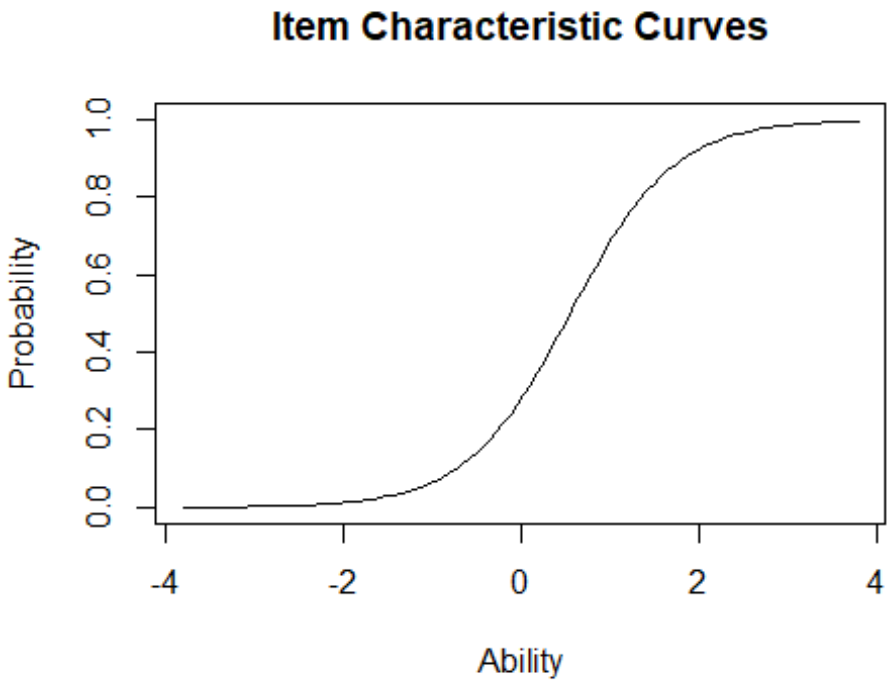

Item Characteristic Curves
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability

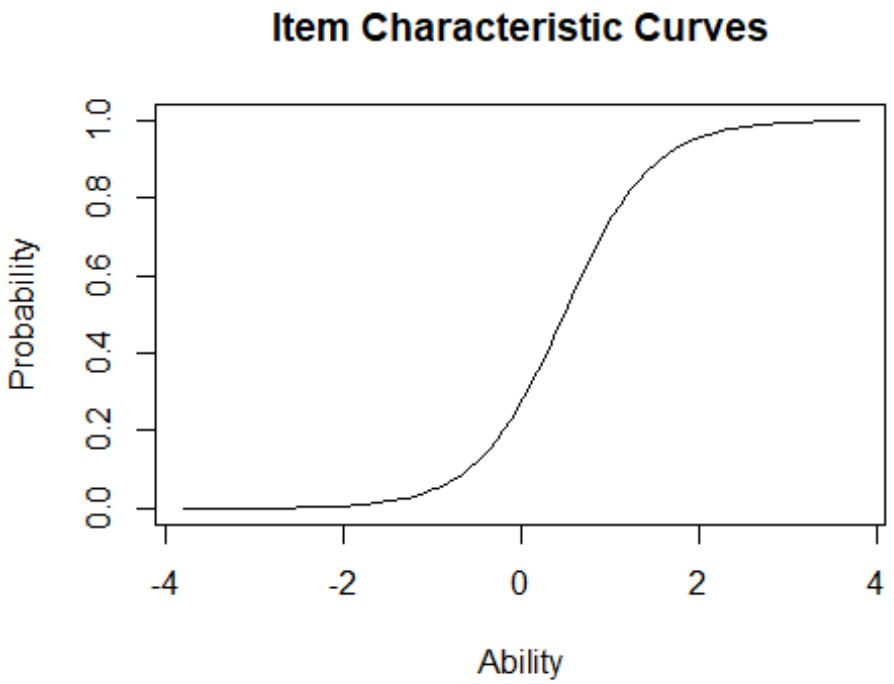

Item 1
1.00
0.75
0.50
0.25
0.00
Ability
-4
-2
0
2
4
Probability
size
0.5
colour
Female
Male

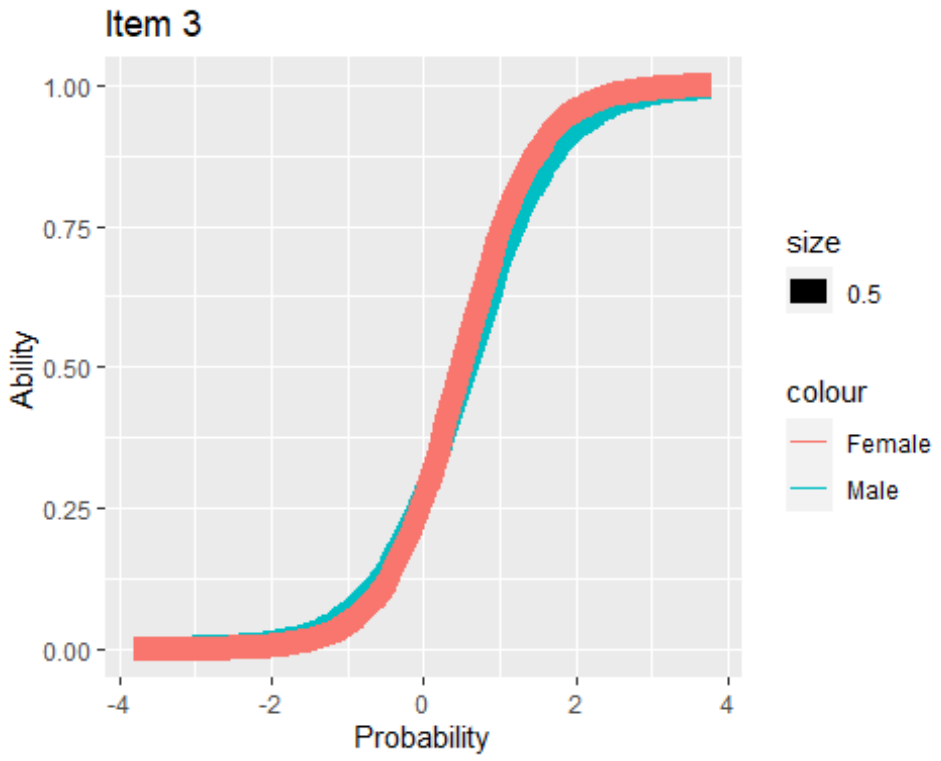

Item Characteristic Curves
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability

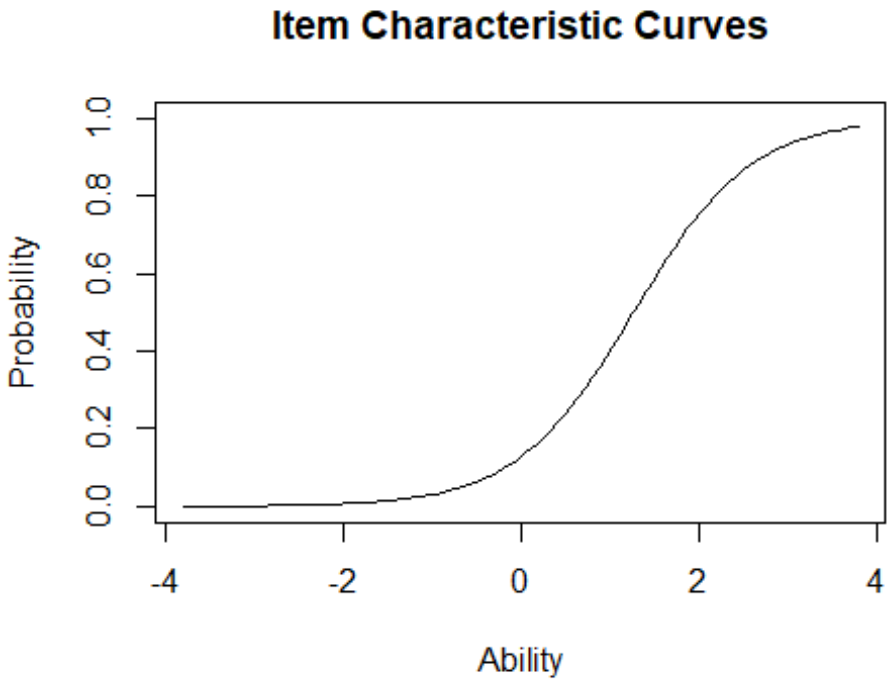

Item Characteristic Curves
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability

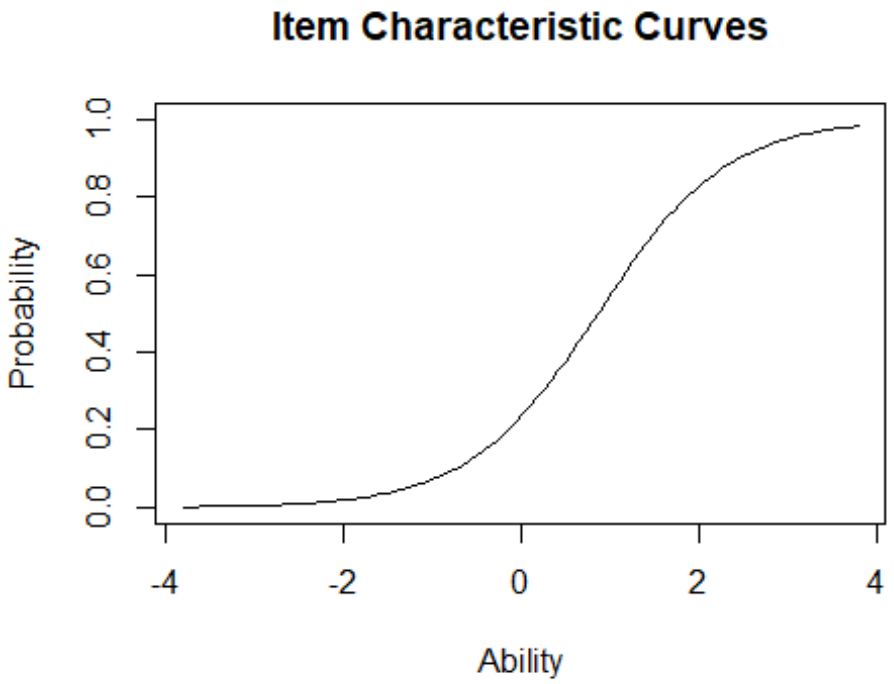

Item 2
1.00
0.75
0.50
0.25
0.00
Ability
-4
-2
0
2
4
Probability
size
0.5
colour
Female
Male

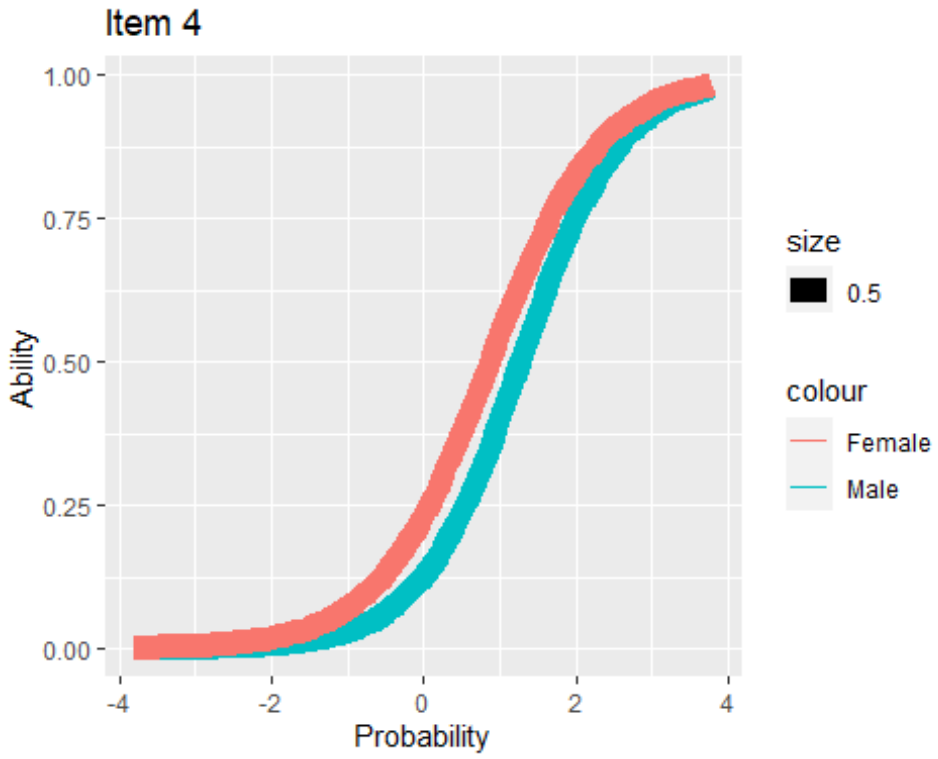

Item Characteristic Curves
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability

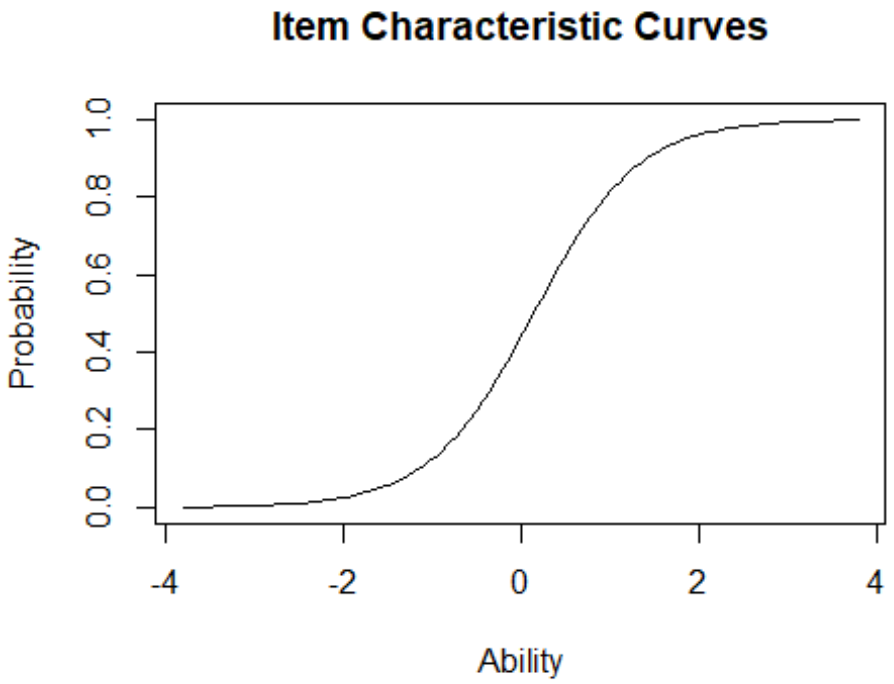

Item Characteristic Curves
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability

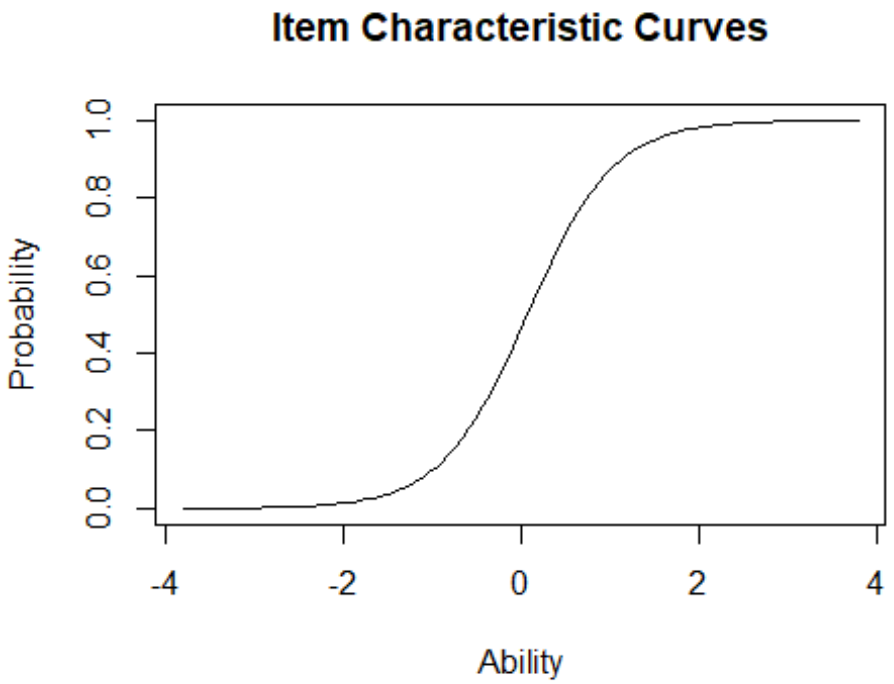

Item 3
1.00
0.75
0.50
0.25
0.00
Ability
-4
-2
0
2
4
Probability
size
0.5
colour
Female
Male

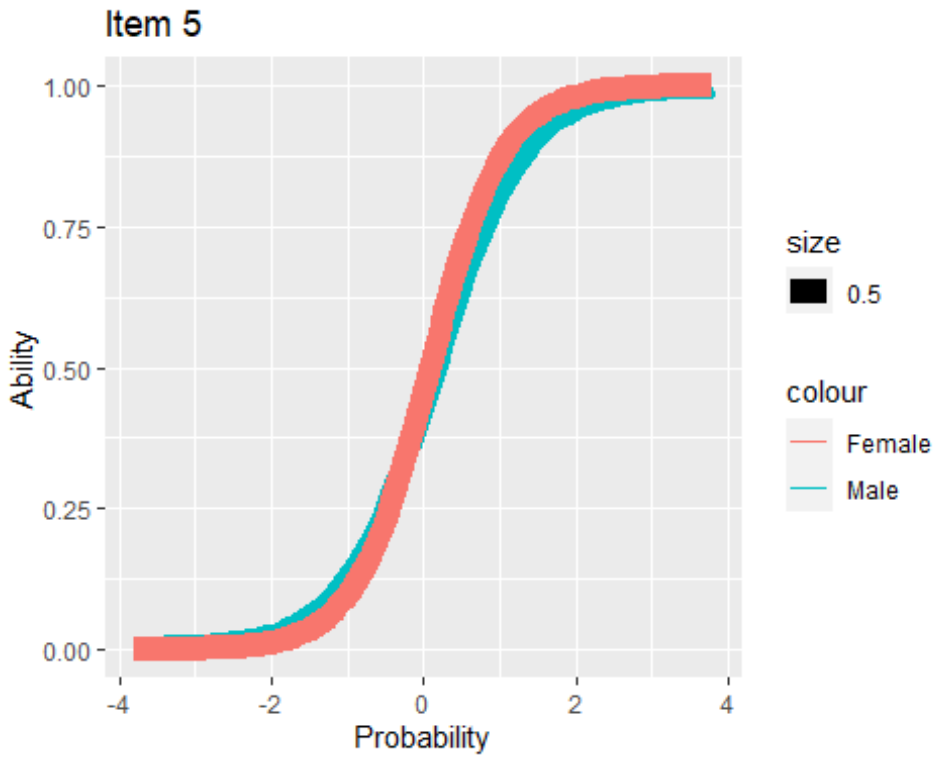

Item Characteristic Curves
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability

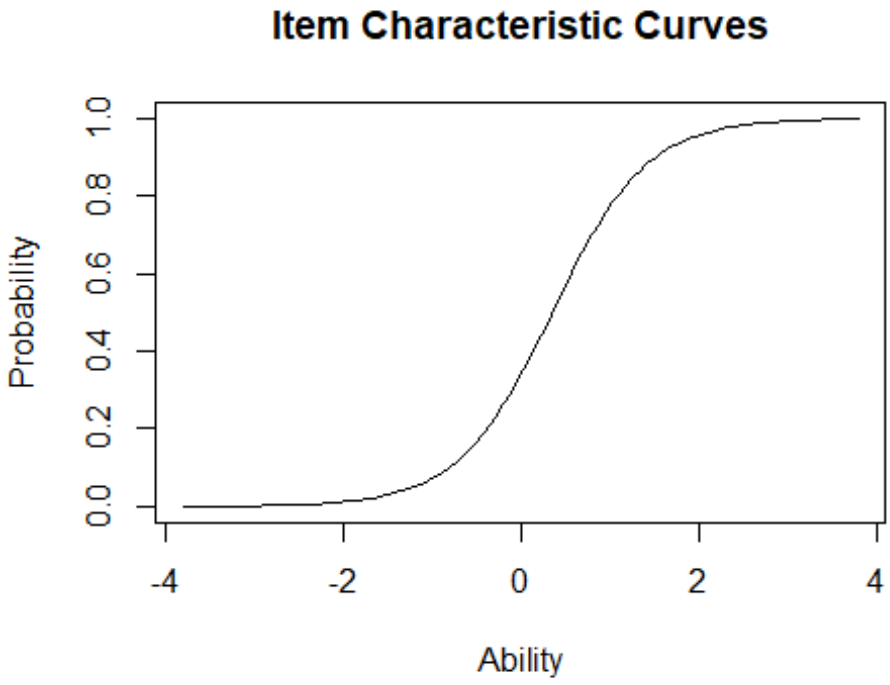

Item Characteristic Curves
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability

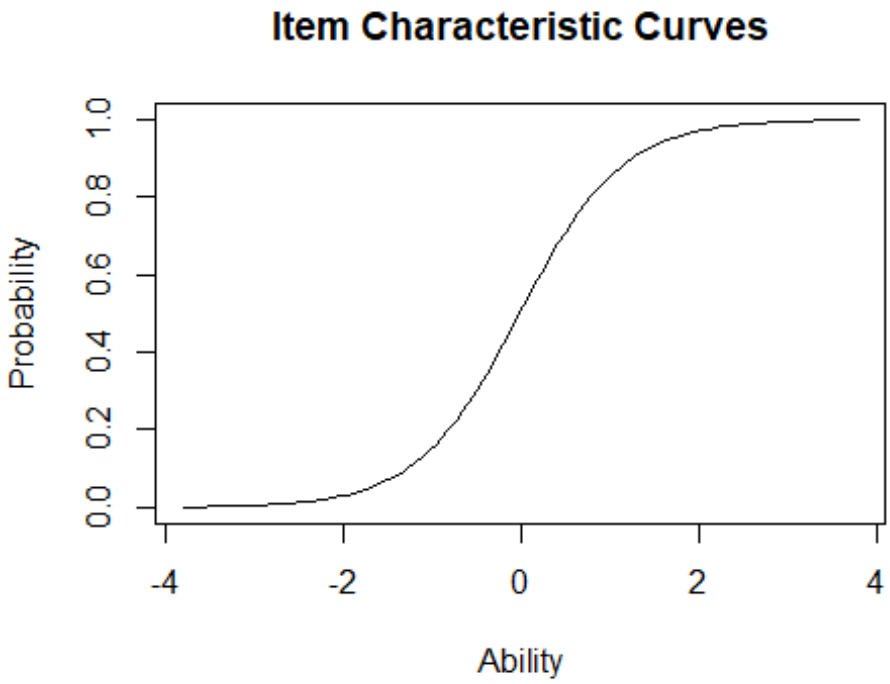

Item 4
1.00
0.75
0.50
0.25
0.00
Ability
-4
-2
0
2
4
Probability
size
0.5
colour
Female
Male

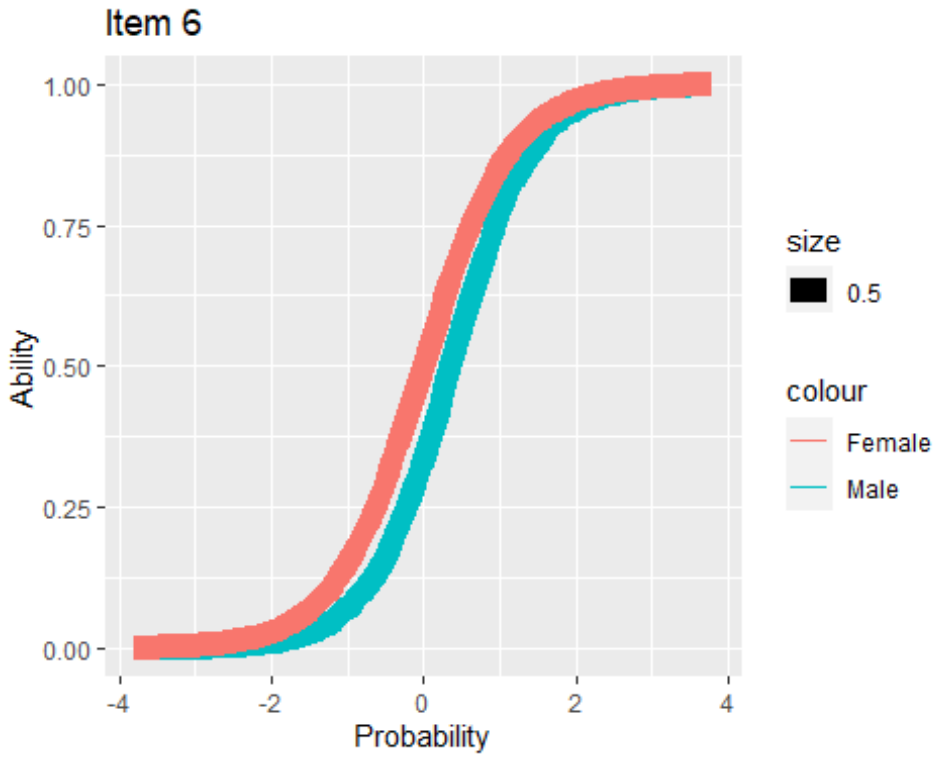

Item Characteristic Curves
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability


Item Characteristic Curves
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability


Item 5
1.00
0.75
0.50
0.25
0.00
Ability
-4
-2
0
2
4
Probability
size
0.5
colour
Female
Male


Item Characteristic Curves
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-4
-2
0
2
4
Ability






```
mod1pl<-rasch(dat_base)
mod2pl<-ltm(dat_base~z1, IRT.param=TRUE)
anova(mod1pl, mod2pl)
```

```
##
##  Likelihood Ratio Table
##             AIC      BIC   log.Lik     LRT df p.value
## mod1pl 279537.2 279597.4 -139761.6
## mod2pl 277908.5 278011.5 -138942.2 1638.79  5  <0.001

# anova(mod2pl, mod3pl)
# anova(mod2pl, mod1pl)
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.