Full tutorial

Danny

1/5/2020

Contents

Introduction

This is meant to be a very general introduction for using the Rasch model to help construct measures and surveys in discipline based education research. This is meant to get you started but is by no means where you should stop. Please see the references section for where to go next.

Installing R and R-Studio

Instructions for installing R:

1. Go to this web page: http://cran.stat.ucla.edu/
2. Under the "Download and Install R" heading, select your operating system (Windows, Mac, Linux).
3. The directions diverge at this stage, depending on your OS.

For Mac, do the following:

1. Under the "Latest Release" heading, select the top ".pkg" link. Save the file to your computer.
2. This is the basic installer file.

For Windows do the following:

1. Under the "Subdirectories" heading, select the top "base" link. Save the file to your computer.
2. This is the basic installer file.

For Linux, you already know what you're doing. I'll stay out of your way.

1. Download and open the installer file. Now, just follow the instructions to set up R. The default settings are fine. No need to open the program yet.
2. Now, we're going to download R-Studio, which is the user interface that makes R faster and easier to use. It's an integrated development environment (IDE)
3. Once you have R-Studio, you won't need to open the "base R" GUI anymore, since R-Studio does this for you.

Instructions for installing R-Studio:

1. Go to this web page: https://www.rstudio.com/products/rstudio/download/#download
2. Under the `Installers for Supported Platforms` header, select your operating system (Windows, Mac, Linux).
3. Download and open the installer file and follow the instructions. The default settings are fine.
4. Once R-Studio is installed, go ahead and open the program from your applications list (Start Menu/Launchpad/Desktop).

Note: if R-Studio does not open, one common reason is that you have installed multiple versions of base R and R-Studio does not know which one to access. This can happen if you have updated R over time or uninstalled it and reinstalled it in the past. To remedy this situation, follow the instructions in the Appendix of this document. If that doesn't work, contact Anthony at `clairmont@ucsb.edu`.

- You may be asked whether you want to install "Command Line Developer Tools." R-Studio needs these to function, so say "Install" and "Agree" to the terms. When R-Studio is done installing its necessary tools, it is read to use! That's all you need to do for now.

Appendix: Troubleshooting R-Studio installation

- On Mac, go to MacintoshHD/Library/Frameworks/R.framework/Versions and delete files associated with the older version of R. When you try to launch R-Studio again, it should automatically find the remaining, current version of R.
- On Windows or Linux, this guide will show you how to tell R-Studio which version of base R to run: https://support.rstudio.com/hc/en-us/articles/200486138-Changing-R-versions-for- RStudio-desktop

The Rasch Model

set your working directory to the folder where you downloaded the CSV file.

1.Go to Session -> Set Working Directory -> Choose Directory

2. install TAM and the WrightMap package

```
install.packages("TAM")
install.packages("WrightMap")
```

3.1 Import data

Take a CSV from outside of R and read it in. This means that it is something you can now work with in R. The .csv file will be read in as something called a data frame or (dataframe). This is a type of object in R, that's essentially a spreadsheet that your're used to working with.

See the first few rows and columns

```r
head(hls)
```

```
##   Hls1 Hls2 Hls3 Hls4 Hls5 Hls6 Hls7 Hls8 Hls9 Hls10 Hls11 Hls12 Hls13
## 1    0    0    0    0    0    0    0    1    0     0     1     0     0
## 2    1    0    0    0    1    0    0    1    0     0     1     1     1
## 3    0    0    0    0    0    0    0    1    0     0     0     1     0
## 4    0    0    0    0    1    0    0    0    0     0     1     0     0
## 5    0    0    0    0    0    0    0    1    0     0     1     1     1
## 6    0    0    0    0    1    0    0    0    0     0     1     1     1
##   Hls14 Hls15 Hls16
## 1     0     0     0
## 2     0     0     1
## 3     0     0     0
## 4     0     1     0
## 5     0     0     1
## 6     0     1     0
```

If you want to see view the data frame:

```
View(hls)
```

Now we call the TAM library you installed in a prior step. This tells R to use the set of functions in available in `TAM` and `WrightMap`

```r
library(TAM)
```

```
## Warning: package 'TAM' was built under R version 3.5.3
```

```
## Loading required package: CDM
```

```
## Loading required package: mvtnorm
```

```
## **********************************
## ** CDM 7.2-30 (2019-02-08 11:13:48)
## ** Cognitive Diagnostic Models  **
## **********************************
```

```
## * TAM 3.3-10 (2019-08-23 13:42:07)
```

3.2 Run the Rasch Model

This command runs a Rasch model on the selected data frame. `mod1` is an object in R that holds the data from our Rasch model (along with a lot of other information). This is the main computation step, now we just ask TAM questions about this model.

Note that the object `hls` has to contain only items and no other information.

```r
mod1 <- tam(hls)
```

```r
summary(mod1)
```

```
## ------------------------------------------------------------
## TAM 3.3-10 (2019-08-23 13:42:07)
## R version 3.5.2 (2018-12-20) x86_64, mingw32 | nodename=LAPTOP-K7402PLE | login=katzd
##
## Date of Analysis: 2020-01-13 09:47:32
## Time difference of 0.07299709 secs
## Computation time: 0.07299709
##
## Multidimensional Item Response Model in TAM
##
## IRT Model: 1PL
## Call:
## tam.mml(resp = resp)
##
## ------------------------------------------------------------
## Number of iterations = 46
## Numeric integration with 21 integration points
##
## Deviance = 3761.76
## Log likelihood = -1880.88
## Number of persons = 317
## Number of persons used = 317
## Number of items = 16
## Number of estimated parameters = 17
##     Item threshold parameters = 16
##     Item slope parameters = 0
##     Regression parameters = 0
##     Variance/covariance parameters = 1
##
## AIC = 3796  | penalty = 34    | AIC=-2*LL + 2*p
## AIC3 = 3813  | penalty = 51    | AIC3=-2*LL + 3*p
## BIC = 3860  | penalty = 97.9    | BIC=-2*LL + log(n)*p
## aBIC = 3806  | penalty = 43.77    | aBIC=-2*LL + log((n-2)/24)*p  (adjusted BIC)
## CAIC = 3877  | penalty = 114.9    | CAIC=-2*LL + [log(n)+1]*p  (consistent AIC)
## AICc = 3798  | penalty = 36.05    | AICc=-2*LL + 2*p + 2*p*(p+1)/(n-p-1)  (bias corrected AIC)
##
## ------------------------------------------------------------
## EAP Reliability
## [1] 0.772
## ------------------------------------------------------------
## Covariances and Variances
##       [,1]
## [1,] 2.932
## ------------------------------------------------------------
## Correlations and Standard Deviations (in the diagonal)
##       [,1]
## [1,] 1.712
## ------------------------------------------------------------
## Regression Coefficients
##      [,1]
```

```
## [1,]    0
## ------------------------------------------------------------
## Item Parameters -A*Xsi
##     item   N     M xsi.item AXsi_.Cat1 B.Cat1.Dim1
## 1   Hls1 317 0.158    2.429      2.429           1
## 2   Hls2 317 0.107    3.004      3.004           1
## 3   Hls3 317 0.038    4.362      4.362           1
## 4   Hls4 317 0.028    4.707      4.707           1
## 5   Hls5 317 0.271    1.491      1.491           1
## 6   Hls6 317 0.098    3.134      3.134           1
## 7   Hls7 317 0.047    4.087      4.087           1
## 8   Hls8 317 0.356    0.919      0.919           1
## 9   Hls9 317 0.085    3.324      3.324           1
## 10 Hls10 317 0.047    4.087      4.087           1
## 11 Hls11 317 0.486    0.122      0.122           1
## 12 Hls12 317 0.426    0.487      0.487           1
## 13 Hls13 317 0.281    1.424      1.424           1
## 14 Hls14 317 0.227    1.821      1.821           1
## 15 Hls15 317 0.369    0.839      0.839           1
## 16 Hls16 317 0.199    2.053      2.053           1
##
## Item Parameters in IRT parameterization
##     item alpha  beta
## 1   Hls1     1 2.429
## 2   Hls2     1 3.004
## 3   Hls3     1 4.362
## 4   Hls4     1 4.707
## 5   Hls5     1 1.491
## 6   Hls6     1 3.134
## 7   Hls7     1 4.087
## 8   Hls8     1 0.919
## 9   Hls9     1 3.324
## 10 Hls10     1 4.087
## 11 Hls11     1 0.122
## 12 Hls12     1 0.487
## 13 Hls13     1 1.424
## 14 Hls14     1 1.821
## 15 Hls15     1 0.839
## 16 Hls16     1 2.053
```

3.3 Item Difficulties

So how difficult were those items? let's ask TAM. We'll extract difficulties (`xsi`) from the `mod1` object (think of `mod1` like a list). We'll access this via `indexing`. The `$` sign means, access `mod1` and extract the object `xsi` which exists in `mod1`. The command `mod1$xsi$xsi` accesses just the column `xsi`, though, you may want other information at times.

Assign those values to a column in the environment called `ItemDiff` using `<-`

```
mod1$xsi
```

```
##             xsi    se.xsi
## Hls1  2.4293464 0.1770696
## Hls2  3.0036806 0.2044455
## Hls3  4.3619882 0.3185594
```

```
## Hls4  4.7075062 0.3628821
## Hls5  1.4909467 0.1501134
## Hls6  3.1336976 0.2120718
## Hls7  4.0869798 0.2884162
## Hls8  0.9192688 0.1419071
## Hls9  3.3237826 0.2242965
## Hls10 4.0869798 0.2884162
## Hls11 0.1220405 0.1384023
## Hls12 0.4867469 0.1389679
## Hls13 1.4239100 0.1488687
## Hls14 1.8211298 0.1574645
## Hls15 0.8391479 0.1411647
## Hls16 2.0532023 0.1639480
```

```
ItemDiff <- mod1$xsi$xsi
ItemDiff
```

```
##  [1] 2.4293464 3.0036806 4.3619882 4.7075062 1.4909467 3.1336976 4.0869798
##  [8] 0.9192688 3.3237826 4.0869798 0.1220405 0.4867469 1.4239100 1.8211298
## [15] 0.8391479 2.0532023
```

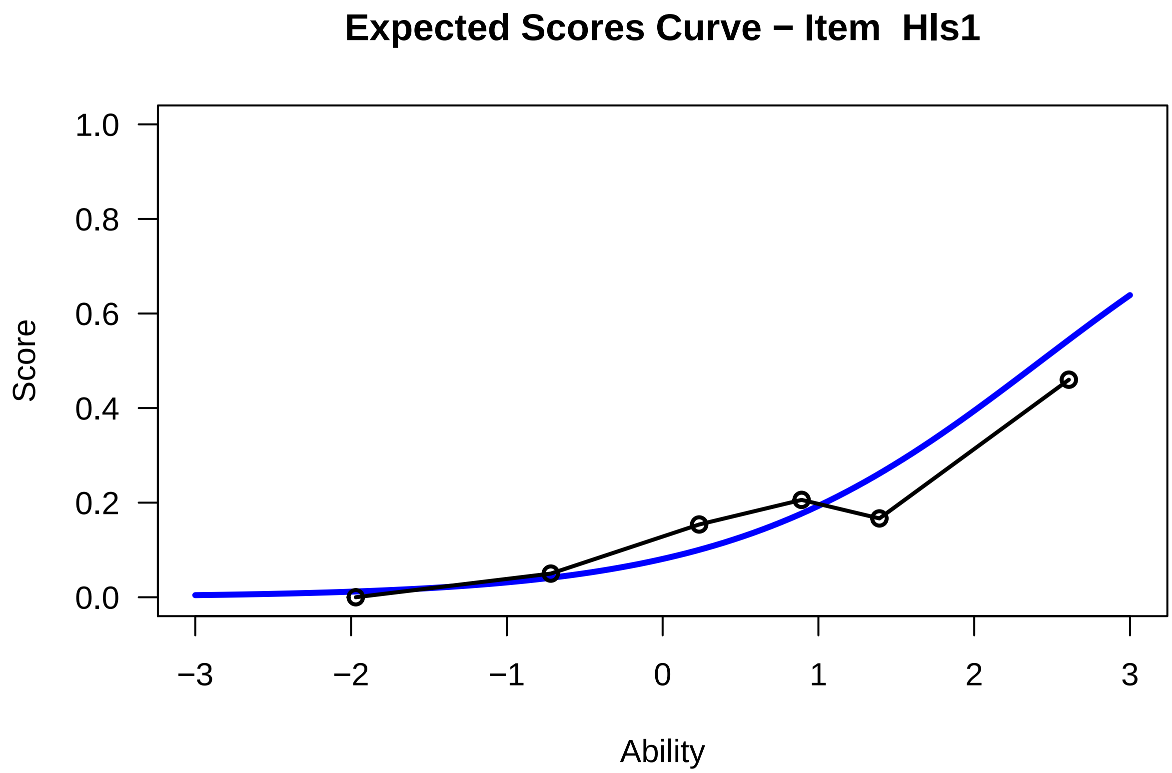
3.4 Visualize

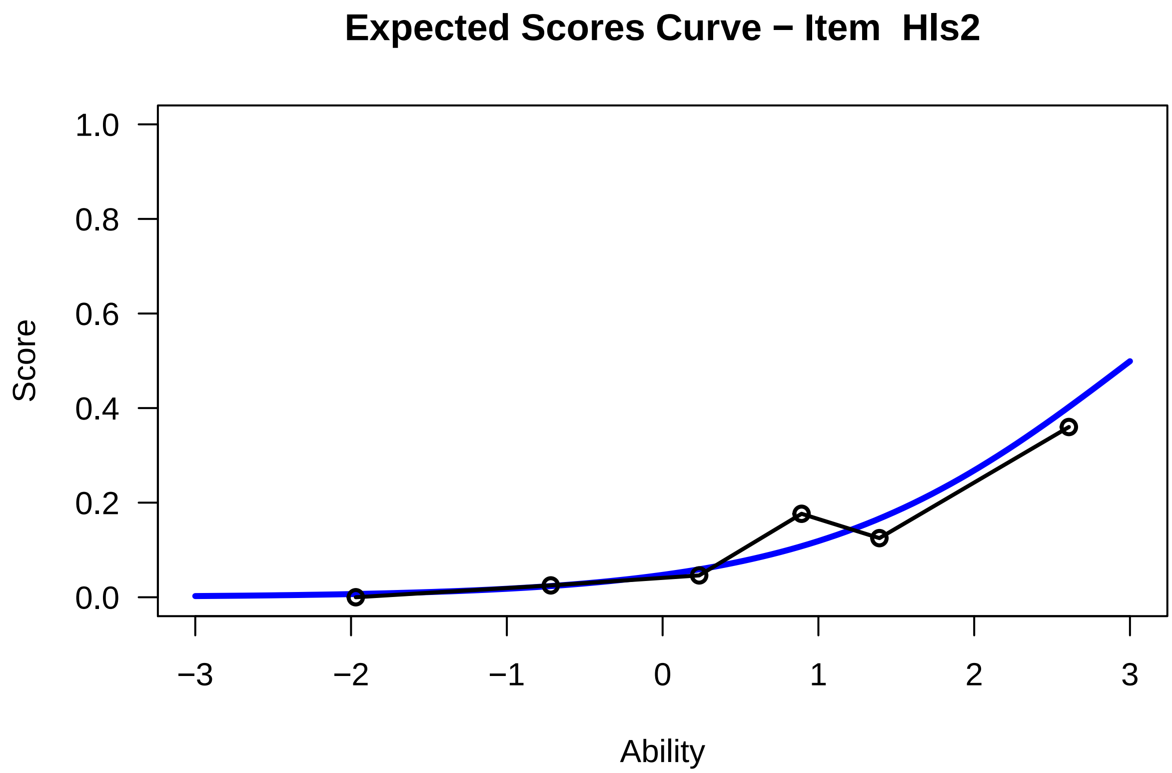
We may want to visualize or describe the distribution of item difficulties (if you want to play with binwidth, you can).

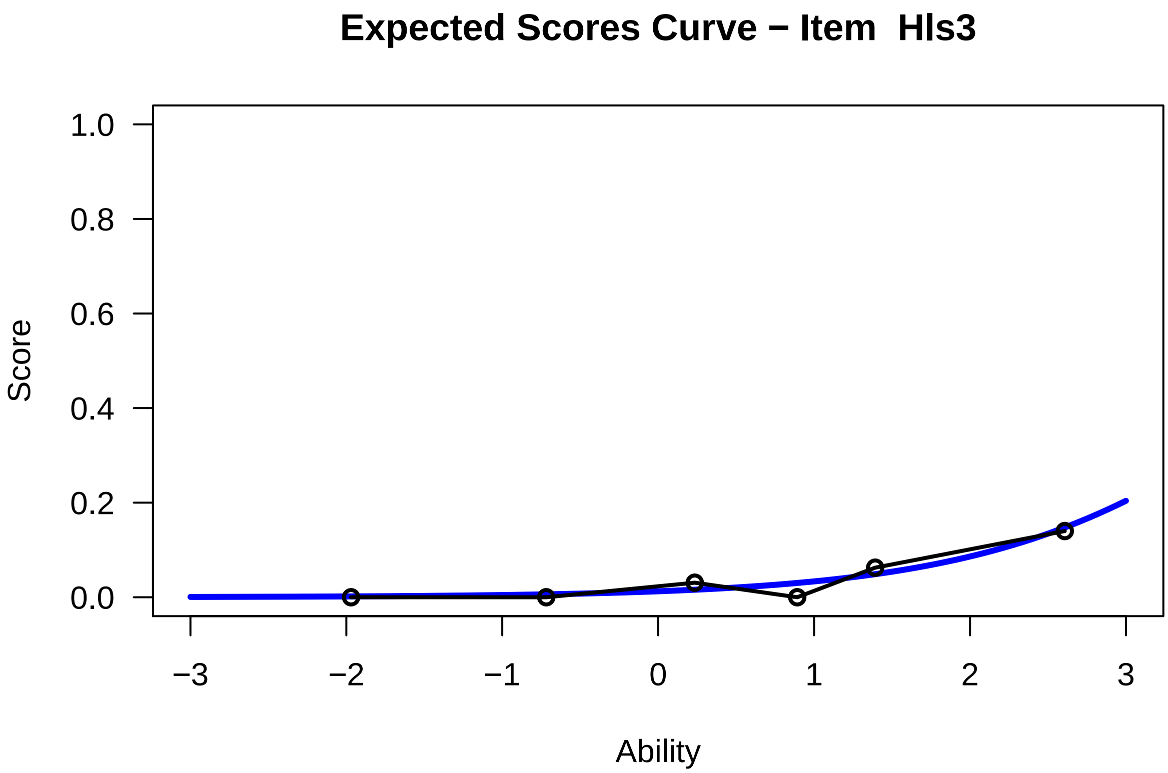
Get Item Characteristic Curves

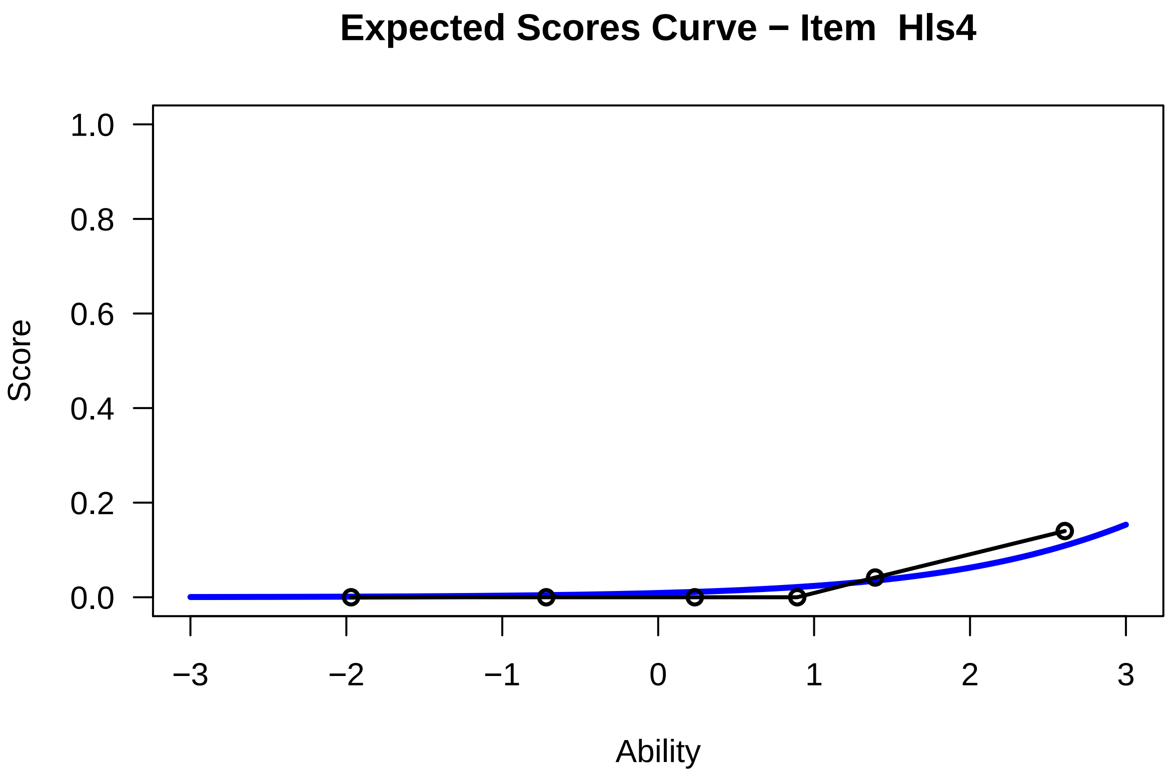
```
plot(mod1, export = F)
```

```
## Iteration in WLE/MLE estimation  1   | Maximal change  2.5494
## Iteration in WLE/MLE estimation  2   | Maximal change  2.2203
## Iteration in WLE/MLE estimation  3   | Maximal change  0.4122
## Iteration in WLE/MLE estimation  4   | Maximal change  0.0234
## Iteration in WLE/MLE estimation  5   | Maximal change  0.003
## Iteration in WLE/MLE estimation  6   | Maximal change  4e-04
## Iteration in WLE/MLE estimation  7   | Maximal change  1e-04
## ----
##  WLE Reliability= 0.615
```

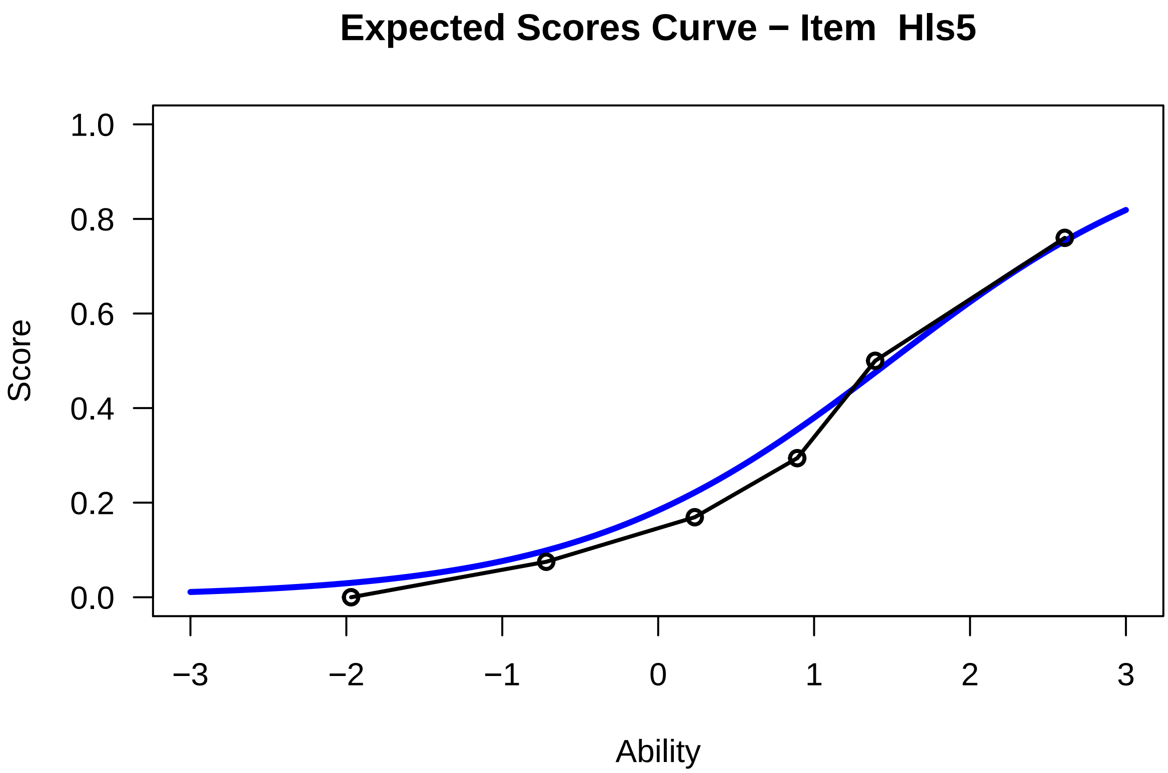

Expected Scores Curve − Item Hls1
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability

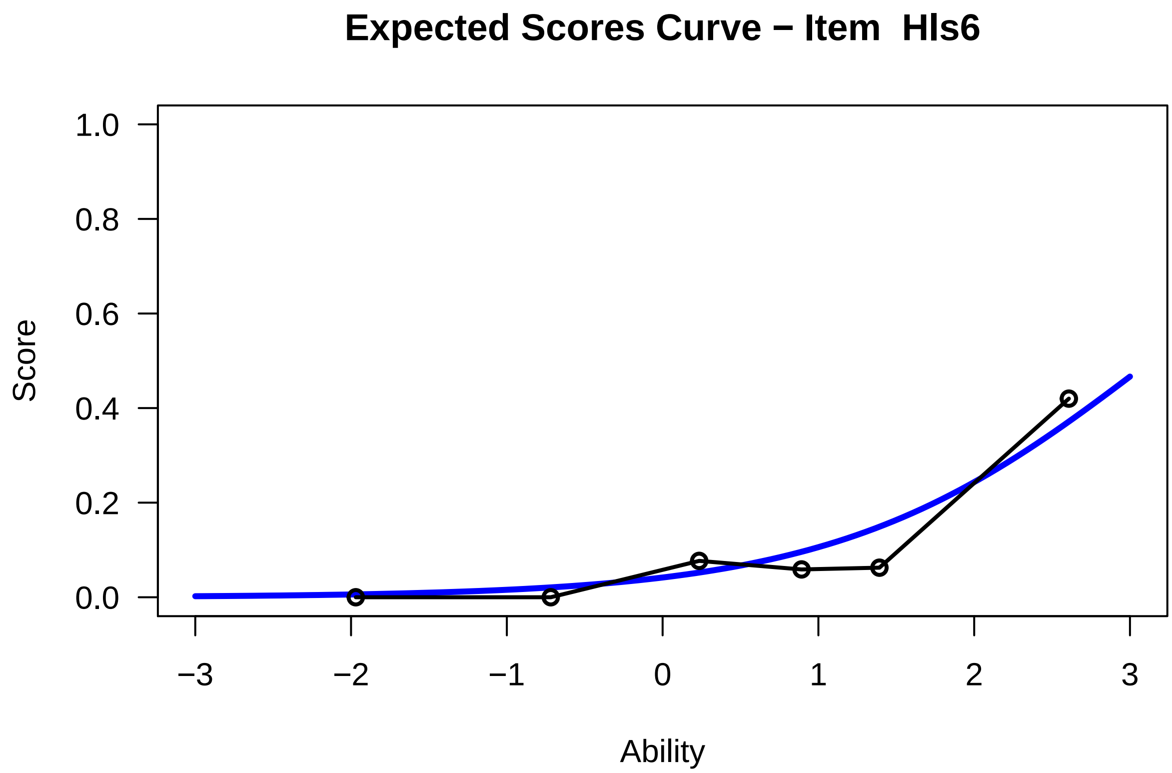

Expected Scores Curve − Item Hls2
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability

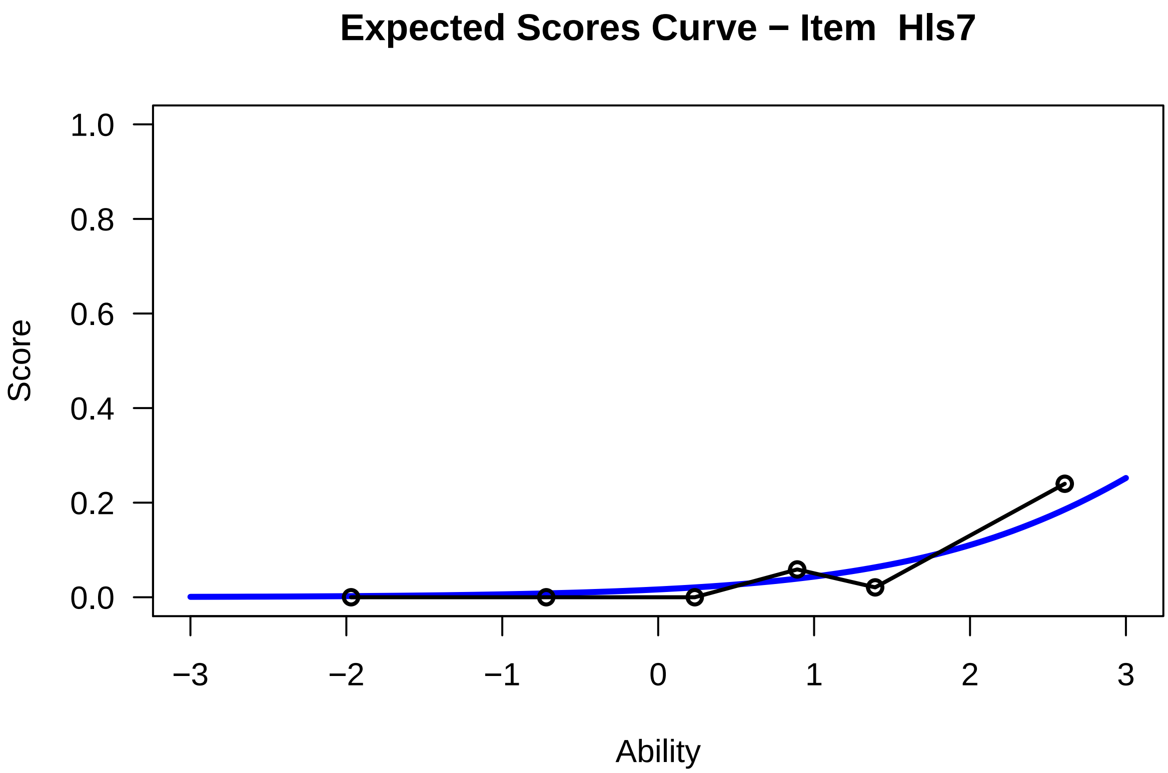

Expected Scores Curve − Item Hls3
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability



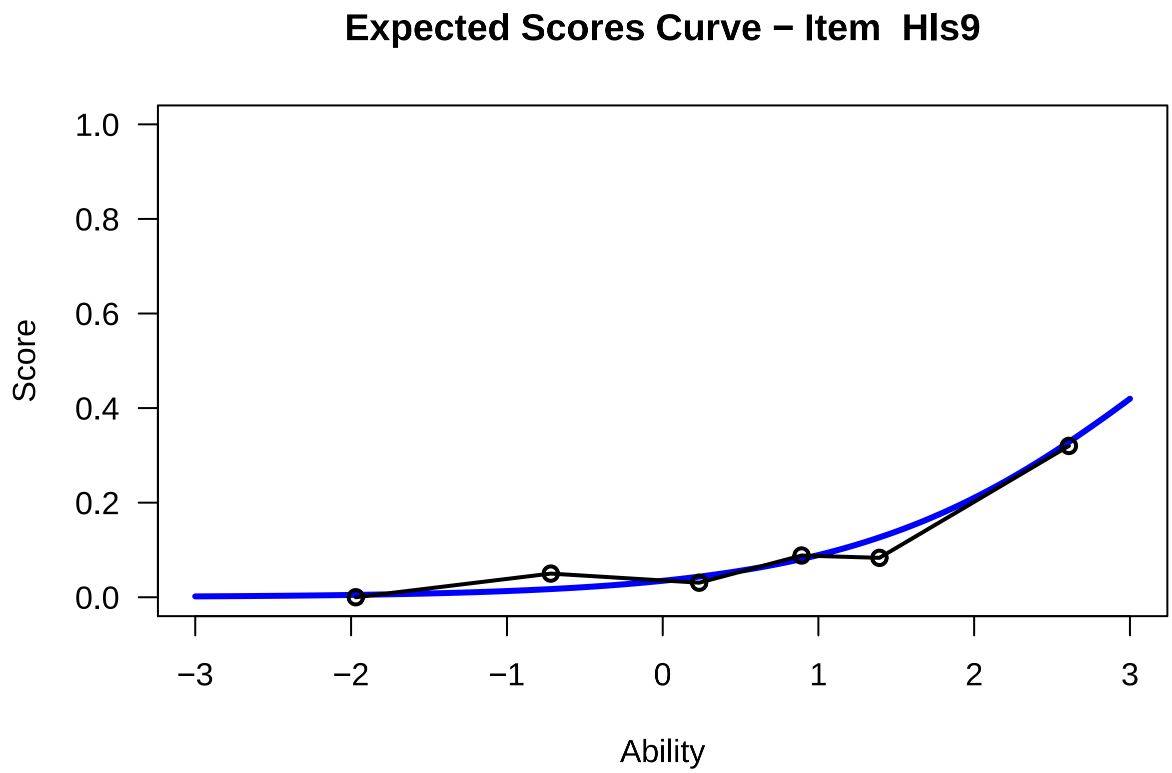
Expected Scores Curve − Item Hls4
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability

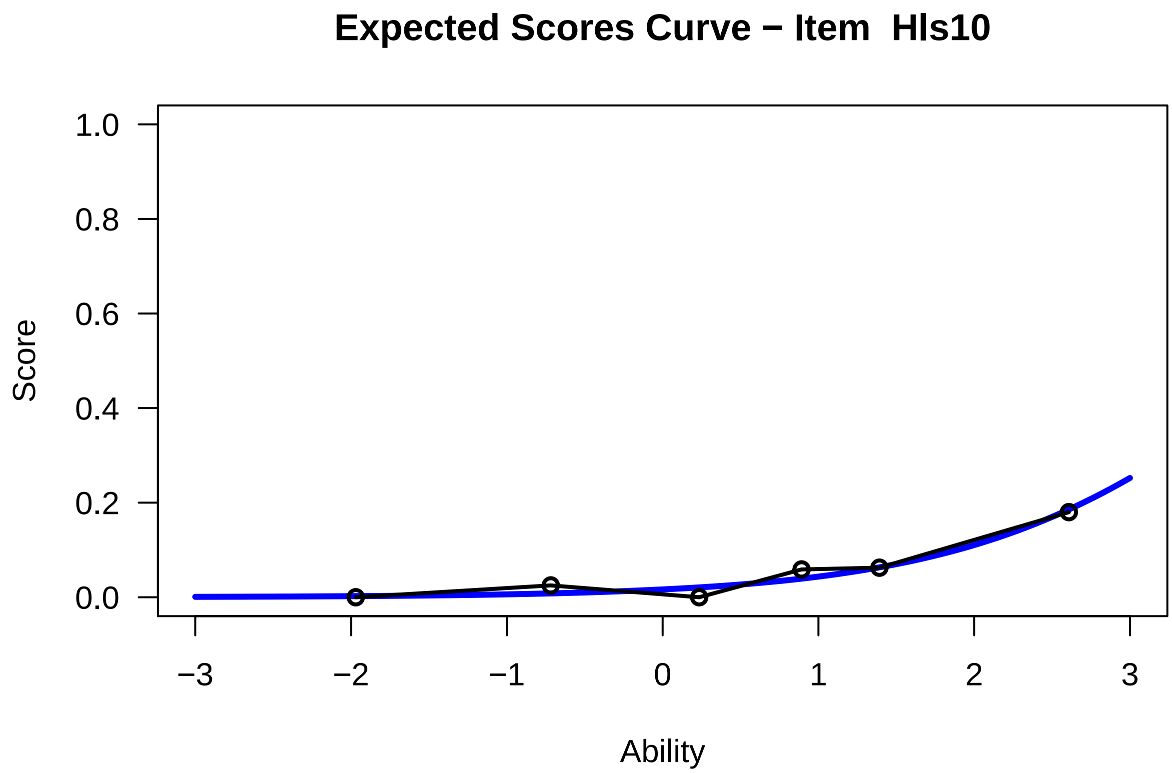

Expected Scores Curve − Item Hls5
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability

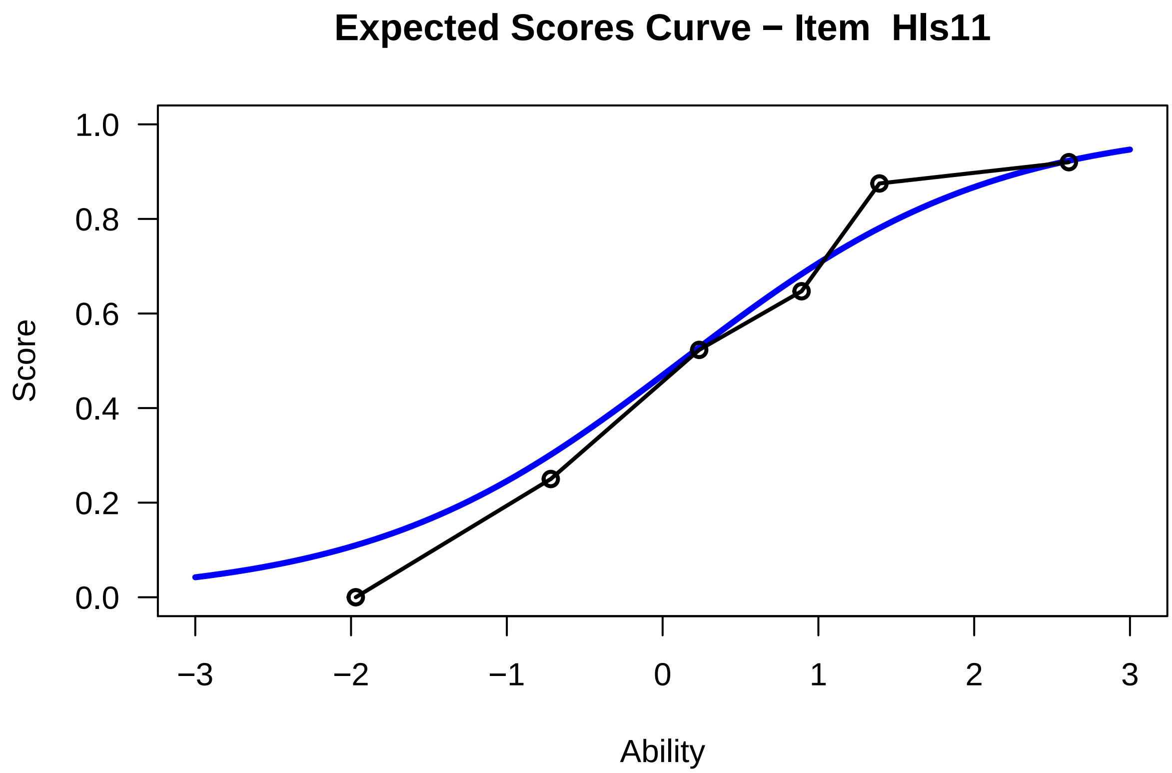

Expected Scores Curve − Item Hls6
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability

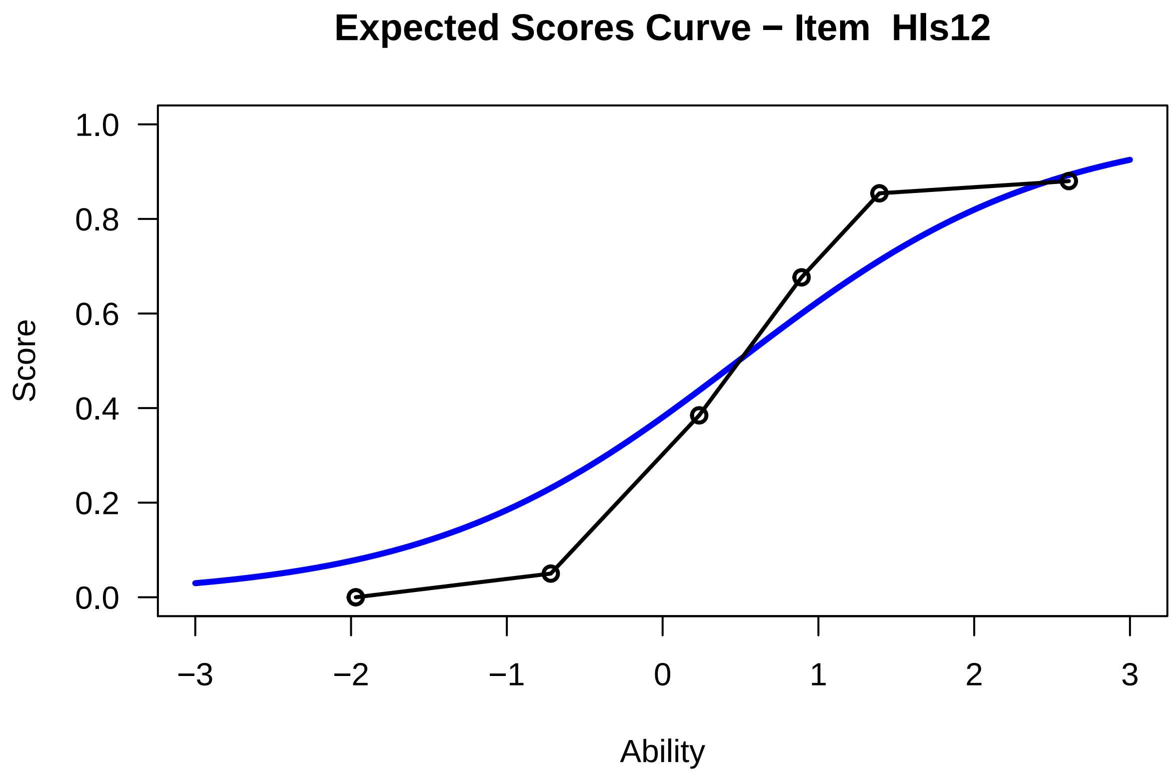

Expected Scores Curve − Item Hls7
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability

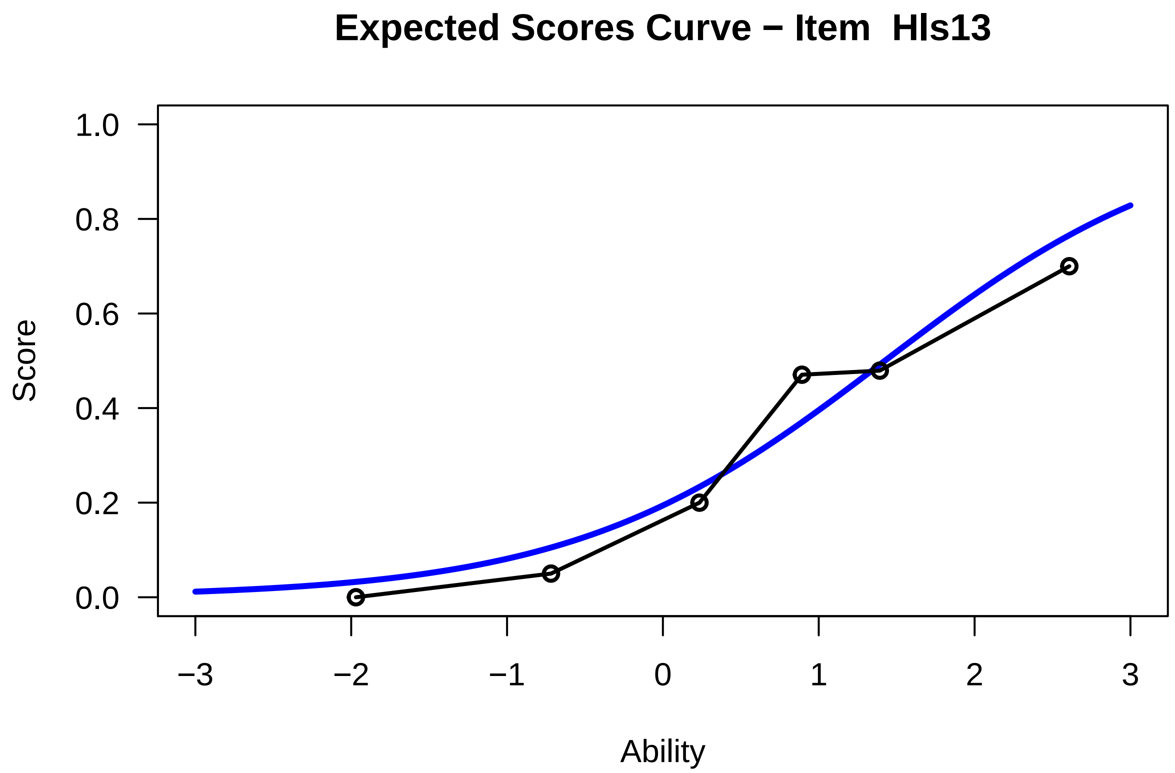

Expected Scores Curve − Item Hls8
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability

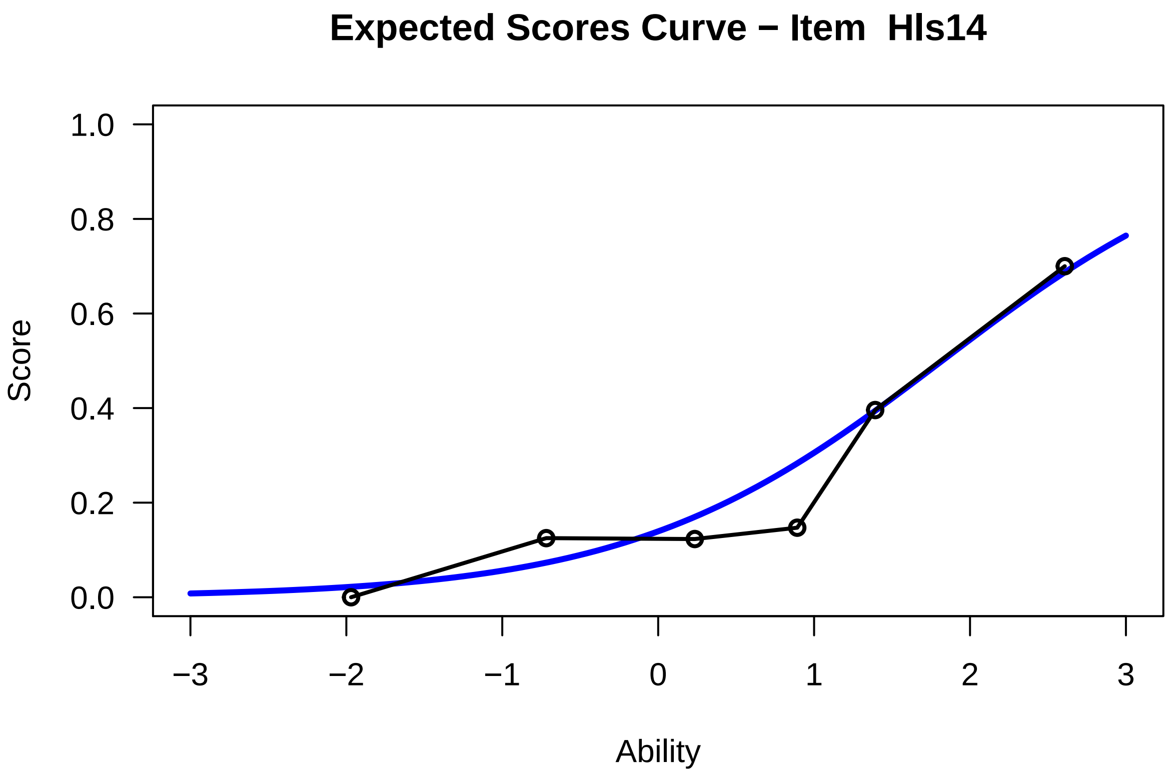

Expected Scores Curve − Item Hls9
1.0
0.8
0.6
0.4
0.2
0.0
Score
−3
−2
−1
0
1
2
3
Ability

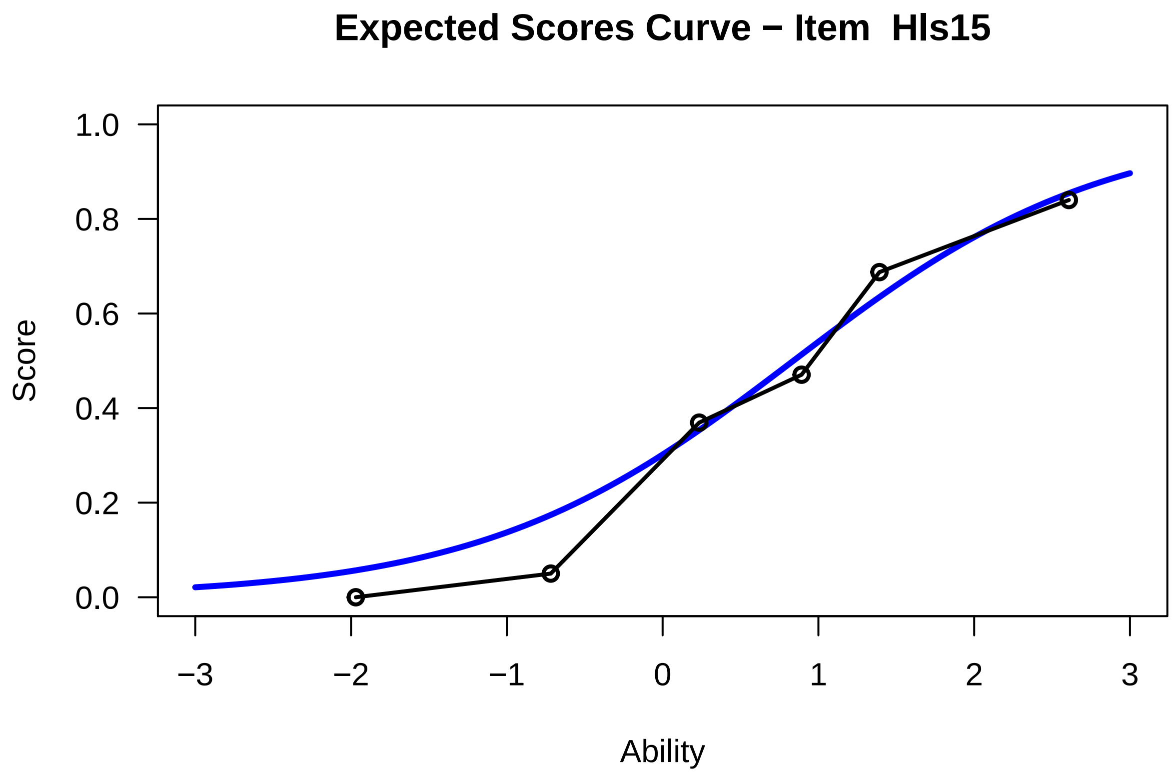

Expected Scores Curve − Item Hls10
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability

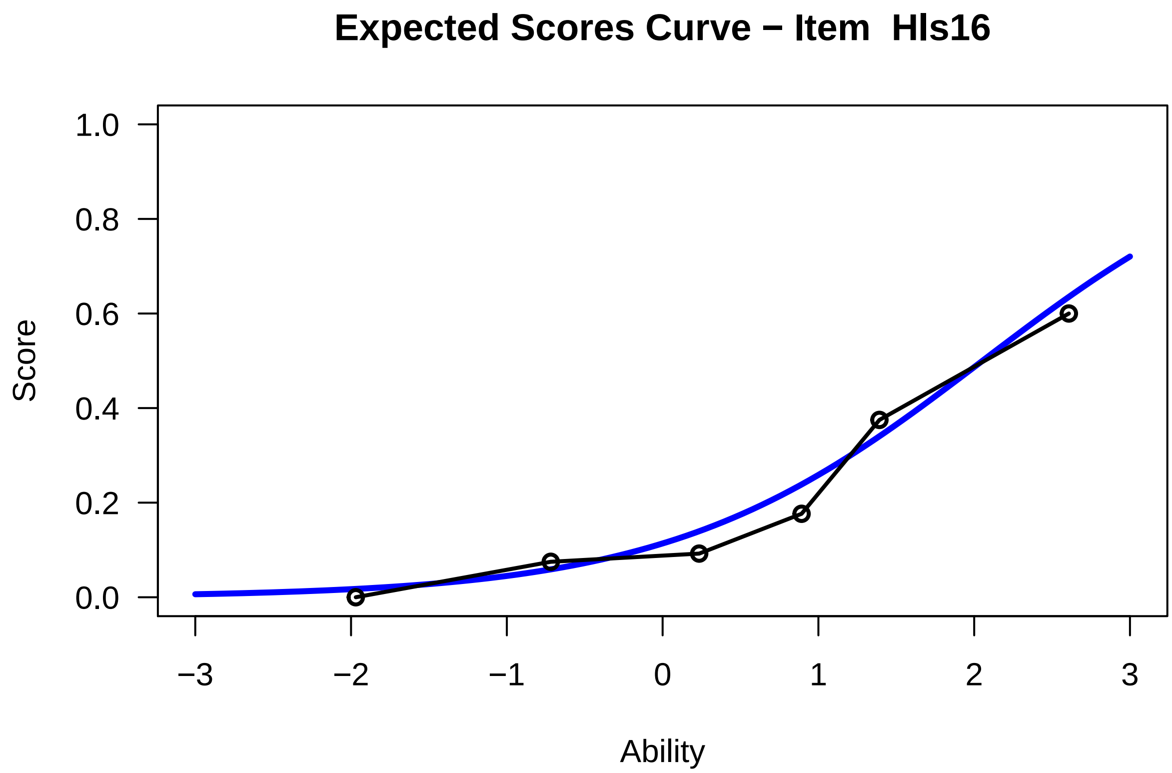

Expected Scores Curve − Item Hls11
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability


Expected Scores Curve − Item Hls12
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability

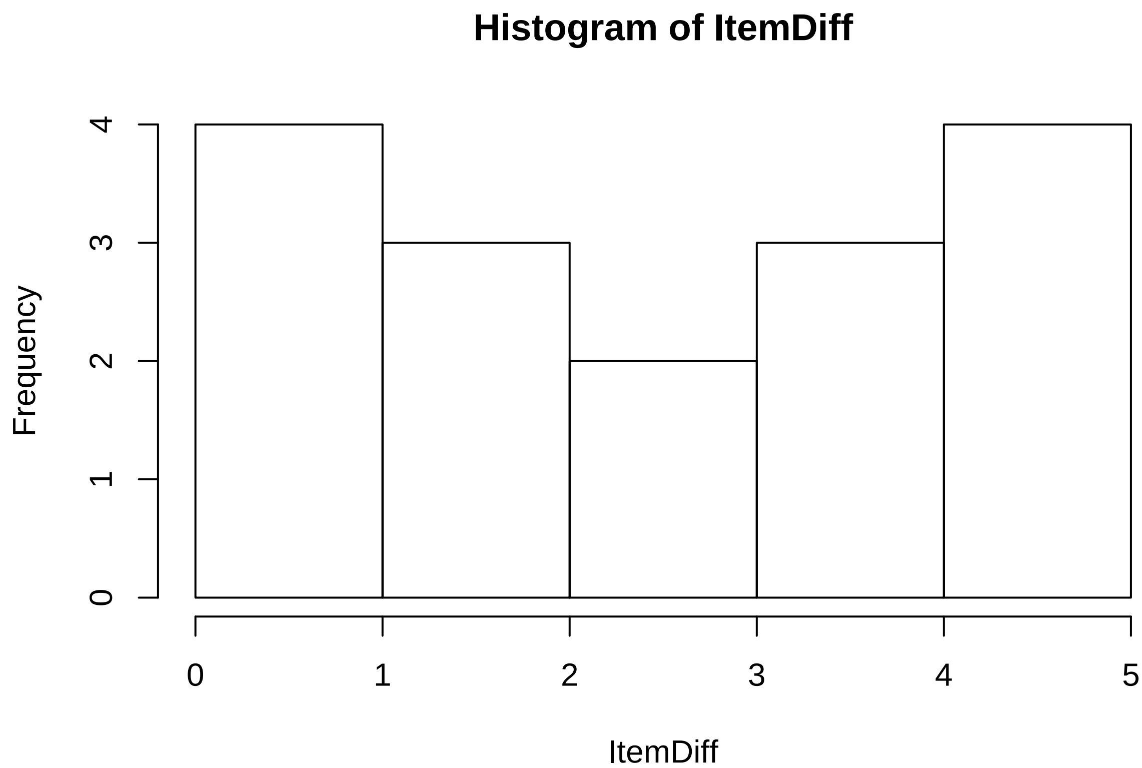

Expected Scores Curve − Item Hls13
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability


Expected Scores Curve − Item Hls14
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability


Expected Scores Curve − Item Hls15
Score
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
Ability




```
hist(ItemDiff)
```




```r
mean(ItemDiff)
```

```
## [1] 2.393147
```

```r
sd(ItemDiff)
```

```
## [1] 1.468218
```

Exercise 1:

1. Which item is the hardest? The easiest? The closest to the mean? **Hint**: try to use the commands such as `max()`, `min()`.

3.5 Person Abilities

Person abilities are also of interest. We can look at the person side of the model by computing person abilities. Compute person abilities using the `tam.wle` function and assign to an object called `Abil`. Extract person abilities (θ_p) from `Abil` and create an object in the `environment` called `PersonAbility` which will essentially be a column vector. **Note**: You may want more information than this at times (such as standard errors) so you may not always want to subset this way.

```r
#generates a data frame
Abil <- tam.wle(mod1)
```

```
## Iteration in WLE/MLE estimation  1   | Maximal change  2.5494
## Iteration in WLE/MLE estimation  2   | Maximal change  2.2203
## Iteration in WLE/MLE estimation  3   | Maximal change  0.4122
## Iteration in WLE/MLE estimation  4   | Maximal change  0.0234
## Iteration in WLE/MLE estimation  5   | Maximal change  0.003
```

```
## Iteration in WLE/MLE estimation  6   | Maximal change  4e-04
## Iteration in WLE/MLE estimation  7   | Maximal change  1e-04
## ----
##  WLE Reliability= 0.615
```

See the first few rows of Abil. Notice you get:

1. `pid`: person id assigned by TAM.
2. `N.items`: Number of items the person was given (this becomes interesting when you have linked test forms where students may not all see the same number of items)
3. `PersonScores`: Number of items the student got right/selected an option for in the survey case.
4. `PersonMax`: Max that person could have gotten right/selected an option for
5. `theta`: estimated person ability
6. `error`: estimated measurement error
7. `WLE.rel`: estimated person seperation reliability.

```
head(Abil)

# or

View(Abil)
```

```
## Object of class 'tam.wle'
## Call: tam.wle(tamobj = mod1)
##
##   WLEs for 317 observations and 1 dimension
##
##   WLE Reliability=0.615
##   Average error variance=1.012
##   WLE mean=0.178
##   WLE variance=2.629
```

The column in the `Abil` data.frame corresponding to person estimates is the `theta` column. Pull out the ability estimates, theta, column if you would like, though, this creates a list. This makes it a little easier for a few basic tasks below.

```
PersonAbility <- Abil$theta
```

```
# Only the first 20 shown
PersonAbility
```

```
##   [1] -0.04587787  2.01460924 -0.04587787  0.46070913  1.28373867
##   [6]  1.28373867 -1.96956632  0.46070913  1.28373867  2.01460924
##  [11] -0.71791902  1.65424265  2.73309925  2.37233153 -1.96956632
##  [16]  0.46070913 -1.96956632 -1.96956632 -0.04587787 -1.96956632
##  [21]  1.28373867  0.89201500 -0.71791902 -0.71791902  1.28373867
##  [26]  3.10211354  0.46070913 -0.04587787  1.28373867  0.46070913
##  [31]  0.89201500 -1.96956632 -1.96956632 -0.71791902  1.65424265
##  [36] -0.04587787 -0.71791902 -1.96956632 -1.96956632  0.46070913
##  [41] -0.04587787  2.73309925 -1.96956632 -1.96956632  0.89201500
##  [46] -1.96956632 -0.71791902 -1.96956632  0.89201500 -1.96956632
##  [51] -1.96956632 -1.96956632 -0.71791902 -1.96956632 -0.71791902
##  [56]  0.89201500  4.86023062  3.48553489  0.46070913 -1.96956632
##  [61]  1.28373867  0.46070913 -1.96956632  1.28373867  0.89201500
##  [66] -1.96956632  0.89201500  0.89201500  2.01460924 -1.96956632
##  [71] -1.96956632  1.28373867 -1.96956632 -1.96956632 -0.04587787
##  [76] -1.96956632  1.28373867  0.46070913 -1.96956632  1.65424265
```

```
##  [81]  1.28373867 -1.96956632  0.89201500 -1.96956632  3.10211354
##  [86]  1.28373867  0.89201500  0.89201500 -0.04587787  1.28373867
##  [91]  0.46070913 -1.96956632  4.33952720  0.46070913 -1.96956632
##  [96]  0.46070913 -1.96956632 -1.96956632  2.01460924  0.89201500
## [101]  3.48553489 -1.96956632  1.65424265  0.46070913  1.65424265
## [106] -1.96956632 -0.04587787 -0.71791902  3.10211354 -0.71791902
## [111]  1.65424265  0.89201500  0.46070913  6.80308915 -0.04587787
## [116] -1.96956632  0.46070913  1.28373867  2.01460924 -0.04587787
## [121]  0.89201500  1.28373867 -0.04587787 -1.96956632  1.65424265
## [126] -0.04587787  0.89201500 -0.71791902 -1.96956632  0.46070913
## [131] -0.71791902  2.01460924  0.46070913  0.89201500  2.73309925
## [136] -0.71791902 -1.96956632 -1.96956632  2.73309925 -0.71791902
## [141] -0.04587787 -0.71791902 -1.96956632  0.46070913  1.65424265
## [146]  1.28373867  1.65424265  1.28373867  0.46070913 -1.96956632
## [151]  2.01460924 -1.96956632  2.01460924 -1.96956632 -0.71791902
## [156] -0.04587787 -1.96956632  0.46070913 -1.96956632 -0.71791902
## [161] -1.96956632  1.28373867 -0.71791902 -1.96956632  0.89201500
## [166] -1.96956632  0.46070913  0.89201500  0.46070913  0.89201500
## [171] -1.96956632 -1.96956632  0.46070913  3.48553489  0.89201500
## [176] -0.71791902 -1.96956632  0.89201500 -0.71791902  1.28373867
## [181]  2.01460924  4.33952720 -1.96956632 -1.96956632  2.73309925
## [186] -1.96956632  2.37233153  0.46070913  2.01460924 -1.96956632
## [191]  2.01460924 -0.71791902  1.65424265 -1.96956632  0.89201500
## [196] -0.04587787 -1.96956632  0.89201500  0.46070913  2.73309925
## [201] -0.04587787 -0.71791902  1.28373867  1.65424265  0.46070913
## [206]  2.01460924 -1.96956632 -0.71791902 -1.96956632  2.37233153
## [211]  1.65424265  2.01460924  2.01460924  0.89201500 -0.71791902
## [216] -1.96956632  0.46070913 -0.04587787  3.10211354 -1.96956632
## [221] -0.71791902  0.46070913  1.28373867 -1.96956632 -0.71791902
## [226]  0.89201500  2.01460924 -1.96956632 -0.71791902  0.46070913
## [231]  0.46070913 -1.96956632  0.89201500  0.89201500 -1.96956632
## [236]  0.46070913 -0.71791902 -0.04587787 -1.96956632 -0.04587787
## [241] -1.96956632 -1.96956632 -1.96956632 -1.96956632  2.01460924
## [246] -0.71791902 -0.71791902  1.28373867  1.28373867 -1.96956632
## [251] -0.04587787  1.65424265  1.28373867 -0.04587787 -0.71791902
## [256]  1.28373867  1.28373867 -0.71791902  2.01460924  1.28373867
## [261]  2.01460924  2.01460924 -0.71791902 -1.96956632 -1.96956632
## [266] -1.96956632  0.89201500  2.73309925  0.46070913  2.37233153
## [271] -0.71791902 -0.04587787  2.73309925 -0.04587787  2.73309925
## [276]  2.01460924 -1.96956632  0.89201500  0.89201500  0.46070913
## [281] -1.96956632  2.37233153  1.28373867 -0.71791902 -1.96956632
## [286]  1.65424265 -1.96956632  0.46070913  2.01460924 -1.96956632
## [291] -0.71791902 -0.71791902 -0.71791902 -0.04587787 -0.04587787
## [296]  0.89201500  2.01460924 -0.71791902  1.28373867  1.28373867
## [301] -1.96956632  0.46070913  0.89201500 -0.04587787 -0.04587787
## [306]  2.01460924  2.37233153  1.28373867  0.46070913  0.89201500
## [311]  1.28373867  0.89201500 -0.71791902  1.28373867  1.28373867
## [316] -0.04587787  2.01460924
```

You can export those estimated abilites to a .csv to save (you can also save directly in R, if you need to).

```
write.csv(Abil,"HLSmod1_thetas.csv")
```

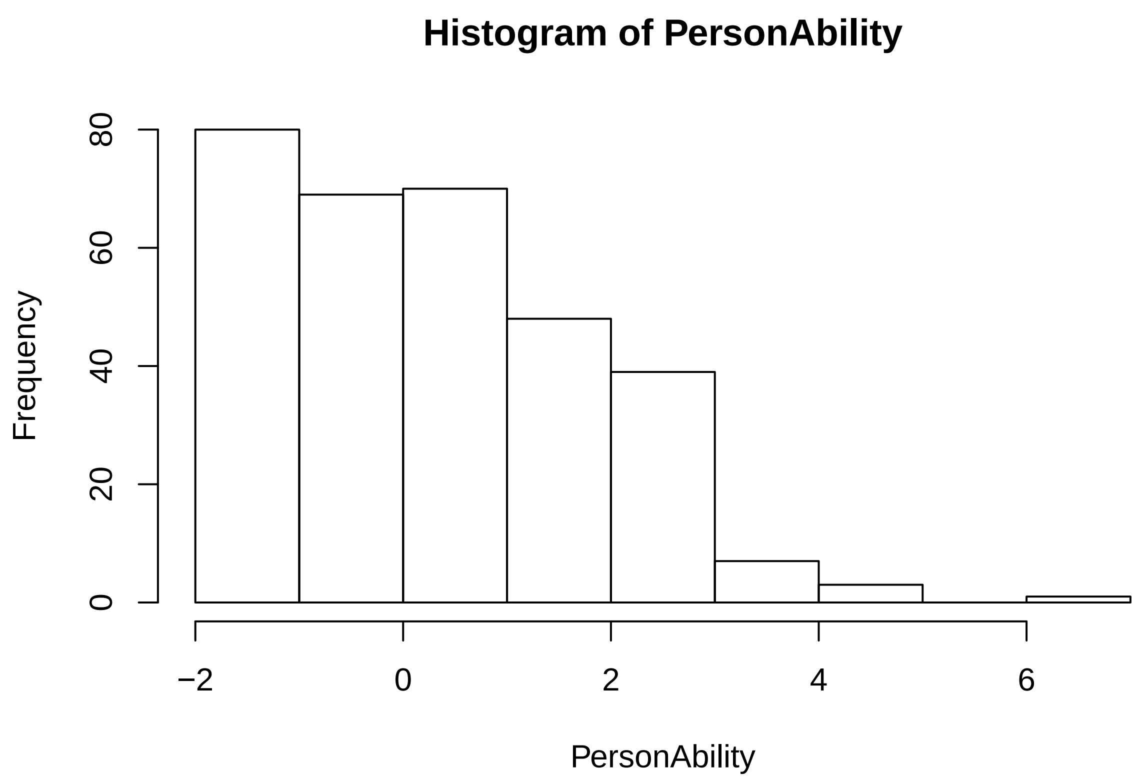
You can find the CSV file in your `Working Directory`. If you need help finding where your `working directory` is:

```r
getwd()
```

```
## [1] "C:/Users/katzd/Desktop/Rprojects/DBER_plain_biome/DBER_Rasch"
```

Descriptives for person ability

```r
hist(PersonAbility)
```




```r
mean(PersonAbility)
```

```
## [1] 0.178227
```

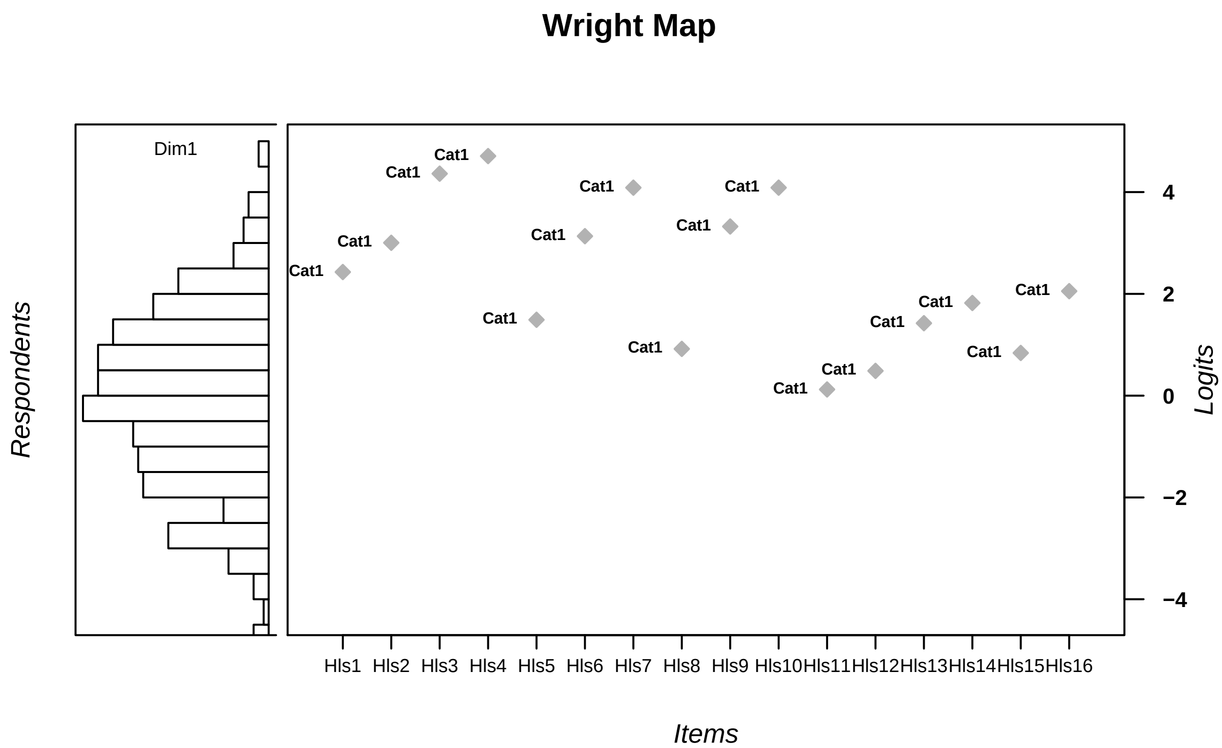
```r
sd(PersonAbility)
```

```
## [1] 1.621543
```

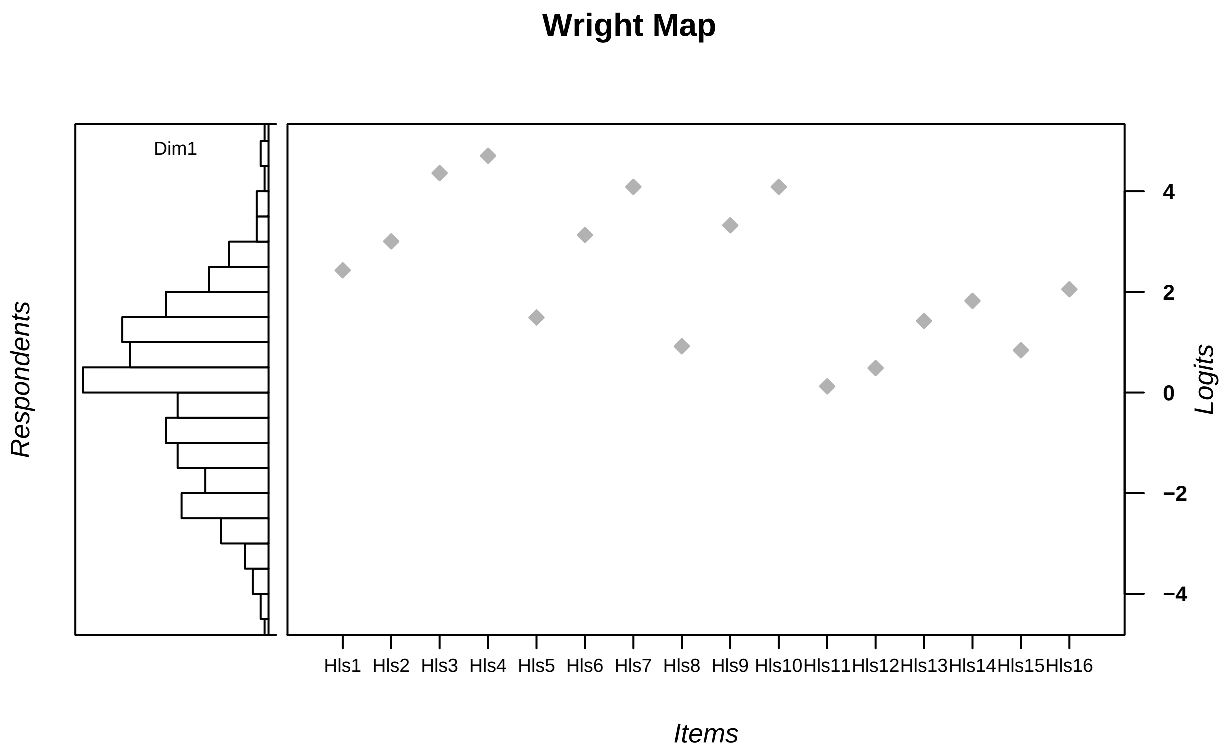
3.6 Wright Map

To visualize the relationship between item difficulty and person ability distributions, call the WrightMap package installed previously. We'll generate a simple WrightMap. We'll clean it up a little bit by removing some elements

```r
library(WrightMap)
IRT.WrightMap(mod1)
```




```
IRT.WrightMap(mod1, show.thr.lab=FALSE)
```



Exercise 2:

1. Are the items appropriately targeted to the ability level of the population?
2. Why do you think?

3.7 Item Fit

Let's find out if the data fit the model. Use the `tam.fit` function to compute fit statistics, then display.

```
fit <- tam.fit(mod1)
```

```
## Item fit calculation based on 100 simulations
## |**********|
## |----------|
```

```
##    parameter    Outfit   Outfit_t     Outfit_p Outfit_pholm     Infit
## 1       Hls1 1.3594498  3.0273606 0.0024669949  0.037004924 1.1265145
## 2       Hls2 1.1679221  0.9366360 0.3489458387  1.000000000 1.0658013
## 3       Hls3 1.3615061  1.1085383 0.2676293865  1.000000000 0.9531203
## 4       Hls4 0.5014529 -2.0514823 0.0402200001  0.442420001 0.8706204
## 5       Hls5 0.9004467 -1.4792745 0.1390669695  1.000000000 0.9828806
## 6       Hls6 0.8948356 -0.8580967 0.3908390783  1.000000000 0.9835958
## 7       Hls7 0.5611632 -2.3607741 0.0182368363  0.237078871 0.9363685
## 8       Hls8 1.0464602  0.5864684 0.5575607855  1.000000000 1.0527404
## 9       Hls9 1.5819631  2.9602266 0.0030741287  0.043037801 0.9961030
## 10     Hls10 1.4195828  1.2651585 0.2058145077  1.000000000 1.0858460
## 11     Hls11 0.9023657 -1.5990286 0.1098142517  1.000000000 0.9267805
```

```
## 12     Hls12 0.7888542 -3.5904089 0.0003301596  0.005282553 0.8705295
## 13     Hls13 0.9244575 -1.1444456 0.2524389016  1.000000000 1.0271382
## 14     Hls14 1.1436088  1.5440291 0.1225812718  1.000000000 1.0197745
## 15     Hls15 0.8698210 -2.1737812 0.0297215696  0.356658835 0.9657435
## 16     Hls16 0.9741745 -0.3797952 0.7040974487  1.000000000 1.0340360
##         Infit_t    Infit_p Infit_pholm
## 1   1.20967112 0.22640512   1.0000000
## 2   0.52207268 0.60161973   1.0000000
## 3  -0.11262383 0.91032879   1.0000000
## 4  -0.36760798 0.71316556   1.0000000
## 5  -0.22262077 0.82383067   1.0000000
## 6  -0.07965113 0.93651473   1.0000000
## 7  -0.22756484 0.81998456   1.0000000
## 8   0.79980749 0.42382234   1.0000000
## 9   0.01693975 0.98648468   1.0000000
## 10  0.44127624 0.65901302   1.0000000
## 11 -1.15384414 0.24856408   1.0000000
## 12 -2.11519056 0.03441371   0.5506194
## 13  0.39041291 0.69623124   1.0000000
## 14  0.26188067 0.79341344   1.0000000
## 15 -0.52297700 0.60099026   1.0000000
## 16  0.40292828 0.68700098   1.0000000
```

Exercise 3:

1. Look at the `Wright Map` and the histograms of person abilities (θ_p) and item difficulties (δ_i). Do you think this instrument is well-targeted for this sample? 2. How might it be optimized?
2. Relative to other items, which item fit our model the worst?

This concludes the planned instruction in the Rasch model for our workshop.

However, we've provided working code for a few other concepts in which you might be interested, including using the Rasch model with polytomous items and with multidimensional models.

Polytomous Items

Polytymous item types (anything with a rating Scale)

We can use the Rasch Partial Credit Model (PCM) to look at polytomous data too. We'll start by bringing in the polytomous items from the survey. Note that TAM needs the bottom category to be coded as 0, so you may need to recode.

```
hls2 <- read.csv("hls_poly_scale.csv")
```

```
head(hls2)
```

```
##   Hls1 Hls2 Hls3 Hls4 Hls5 Hls6 Hls7 Hls8 Hls9 Hls10 Hls11 Hls12 Hls13
## 1    1    1    1    0    1    1    0    2    1     1     2     1     1
## 2    2    1    1    1    2    1    1    2    1     1     2     2     2
## 3    0    1    1    1    1    1    1    2    1     0     1     2     1
## 4    1    1    0    0    2    1    0    1    0     0     2     1     1
## 5    1    1    0    0    1    0    0    2    0     0     2     2     2
## 6    1    1    1    1    2    1    1    1    1     0     2     2     2
```

```
##   Hls14 Hls15 Hls16
## 1     0     1     1
## 2     1     1     2
## 3     1     1     1
## 4     1     2     1
## 5     1     1     2
## 6     1     2     1
```

```
View(hls2)
```

TAM will automatically run the PCM when our data is polytomous. There are other model-types for polytomous data such as the rating scale model. This may be more appropriate for Likert-type items. For more information, read TAM documentation or see the reference list (Bond & Fox, 2007)

```
mod2 <- tam(hls2)
```

```
summary(mod2)
```

```
## ------------------------------------------------------------
## TAM 3.3-10 (2019-08-23 13:42:07)
## R version 3.5.2 (2018-12-20) x86_64, mingw32 | nodename=LAPTOP-K7402PLE | login=katzd
##
## Date of Analysis: 2020-01-13 09:47:33
## Time difference of 0.1500039 secs
## Computation time: 0.1500039
##
## Multidimensional Item Response Model in TAM
##
## IRT Model: 1PL
## Call:
## tam.mml(resp = resp)
##
## ------------------------------------------------------------
## Number of iterations = 57
## Numeric integration with 21 integration points
##
## Deviance = 8371.25
## Log likelihood = -4185.63
## Number of persons = 317
## Number of persons used = 317
## Number of items = 16
## Number of estimated parameters = 49
##     Item threshold parameters = 48
##     Item slope parameters = 0
##     Regression parameters = 0
##     Variance/covariance parameters = 1
##
## AIC = 8469  | penalty = 98    | AIC=-2*LL + 2*p
## AIC3 = 8518  | penalty = 147    | AIC3=-2*LL + 3*p
## BIC = 8653  | penalty = 282.19    | BIC=-2*LL + log(n)*p
## aBIC = 8497  | penalty = 126.15    | aBIC=-2*LL + log((n-2)/24)*p  (adjusted BIC)
## CAIC = 8702  | penalty = 331.19    | CAIC=-2*LL + [log(n)+1]*p  (consistent AIC)
## AICc = 8488  | penalty = 116.35    | AICc=-2*LL + 2*p + 2*p*(p+1)/(n-p-1)  (bias corrected AIC)
##
## ------------------------------------------------------------
## EAP Reliability
```

```
## [1] 0.914
## ------------------------------------------------------------
## Covariances and Variances
##       [,1]
## [1,] 2.615
## ------------------------------------------------------------
## Correlations and Standard Deviations (in the diagonal)
##       [,1]
## [1,] 1.617
## ------------------------------------------------------------
## Regression Coefficients
##      [,1]
## [1,]    0
## ------------------------------------------------------------
## Item Parameters -A*Xsi
##     item   N     M xsi.item AXsi_.Cat1 AXsi_.Cat2 AXsi_.Cat3 B.Cat1.Dim1
## 1   Hls1 317 0.978    1.427     -1.846      0.452      4.282           1
## 2   Hls2 317 0.874    2.074     -1.502      1.263      6.221           1
## 3   Hls3 317 0.634    2.903     -0.381      3.581      8.709           1
## 4   Hls4 317 0.530    2.809      0.176      4.500      8.428           1
## 5   Hls5 317 1.091    1.198     -1.684     -0.302      3.595           1
## 6   Hls6 317 0.830    1.818     -1.155      1.784      5.455           1
## 7   Hls7 317 0.615    2.455     -0.166      3.587      7.364           1
## 8   Hls8 317 1.237    0.781     -2.136     -1.239      2.342           1
## 9   Hls9 317 0.644    2.098     -0.008      2.982      6.295           1
## 10 Hls10 317 0.615    2.630     -0.183      3.511      7.890           1
## 11 Hls11 317 1.413    0.325     -2.106     -1.934      0.974           1
## 12 Hls12 317 1.338    0.512     -2.318     -1.805      1.537           1
## 13 Hls13 317 1.136    1.162     -2.127     -0.789      3.487           1
## 14 Hls14 317 1.063    1.070     -1.762      0.000      3.209           1
## 15 Hls15 317 1.243    0.792     -2.043     -1.232      2.376           1
## 16 Hls16 317 1.000    1.434     -1.629      0.278      4.303           1
##    B.Cat2.Dim1 B.Cat3.Dim1
## 1            2           3
## 2            2           3
## 3            2           3
## 4            2           3
## 5            2           3
## 6            2           3
## 7            2           3
## 8            2           3
## 9            2           3
## 10           2           3
## 11           2           3
## 12           2           3
## 13           2           3
## 14           2           3
## 15           2           3
## 16           2           3
##
## Item Parameters Xsi
##               xsi se.xsi
## Hls1_Cat1  -1.846  0.177
## Hls1_Cat2   2.298  0.174
```

```
## Hls1_Cat3   3.830  0.464
## Hls2_Cat1  -1.502  0.164
## Hls2_Cat2   2.765  0.200
## Hls2_Cat3   4.958  0.780
## Hls3_Cat1  -0.381  0.138
## Hls3_Cat2   3.962  0.321
## Hls3_Cat3   5.127  1.117
## Hls4_Cat1   0.176  0.133
## Hls4_Cat2   4.325  0.378
## Hls4_Cat3   3.927  0.853
## Hls5_Cat1  -1.684  0.178
## Hls5_Cat2   1.382  0.147
## Hls5_Cat3   3.897  0.394
## Hls6_Cat1  -1.155  0.154
## Hls6_Cat2   2.939  0.211
## Hls6_Cat3   3.671  0.519
## Hls7_Cat1  -0.166  0.136
## Hls7_Cat2   3.753  0.295
## Hls7_Cat3   3.776  0.689
## Hls8_Cat1  -2.136  0.199
## Hls8_Cat2   0.897  0.139
## Hls8_Cat3   3.581  0.324
## Hls9_Cat1  -0.008  0.136
## Hls9_Cat2   2.990  0.229
## Hls9_Cat3   3.313  0.482
## Hls10_Cat1 -0.183  0.136
## Hls10_Cat2  3.695  0.292
## Hls10_Cat3  4.379  0.822
## Hls11_Cat1 -2.106  0.210
## Hls11_Cat2  0.173  0.138
## Hls11_Cat3  2.907  0.235
## Hls12_Cat1 -2.318  0.212
## Hls12_Cat2  0.513  0.136
## Hls12_Cat3  3.342  0.282
## Hls13_Cat1 -2.127  0.194
## Hls13_Cat2  1.339  0.144
## Hls13_Cat3  4.276  0.449
## Hls14_Cat1 -1.761  0.178
## Hls14_Cat2  1.762  0.155
## Hls14_Cat3  3.208  0.331
## Hls15_Cat1 -2.042  0.197
## Hls15_Cat2  0.810  0.138
## Hls15_Cat3  3.609  0.323
## Hls16_Cat1 -1.629  0.172
## Hls16_Cat2  1.907  0.160
## Hls16_Cat3  4.025  0.457
##
## Item Parameters in IRT parameterization
##     item alpha  beta tau.Cat1 tau.Cat2 tau.Cat3
## 1   Hls1     1 1.427   -3.274    0.871    2.403
## 2   Hls2     1 2.074   -3.575    0.691    2.885
## 3   Hls3     1 2.903   -3.284    1.059    2.224
## 4   Hls4     1 2.809   -2.633    1.515    1.118
## 5   Hls5     1 1.198   -2.883    0.184    2.699
```

```
## 6   Hls6     1 1.818   -2.974    1.121    1.853
## 7   Hls7     1 2.455   -2.620    1.299    1.322
## 8   Hls8     1 0.781   -2.917    0.116    2.800
## 9   Hls9     1 2.098   -2.106    0.891    1.215
## 10 Hls10     1 2.630   -2.814    1.065    1.749
## 11 Hls11     1 0.325   -2.431   -0.152    2.583
## 12 Hls12     1 0.512   -2.830    0.001    2.830
## 13 Hls13     1 1.162   -3.290    0.176    3.113
## 14 Hls14     1 1.070   -2.831    0.692    2.139
## 15 Hls15     1 0.792   -2.835    0.018    2.816
## 16 Hls16     1 1.434   -3.063    0.472    2.590
```

Item Difficulties

Now we'll get item and person characteristics just like before

```
mod2$xsi
```

```
##                      xsi    se.xsi
## Hls1_Cat1  -1.846046710 0.1770841
## Hls1_Cat2   2.298334842 0.1736858
## Hls1_Cat3   3.830385868 0.4635448
## Hls2_Cat1  -1.501715752 0.1640914
## Hls2_Cat2   2.764560868 0.2004621
## Hls2_Cat3   4.958324889 0.7804292
## Hls3_Cat1  -0.380834628 0.1378983
## Hls3_Cat2   3.962213547 0.3205437
## Hls3_Cat3   5.127428610 1.1165768
## Hls4_Cat1   0.175976124 0.1331498
## Hls4_Cat2   4.324558567 0.3775359
## Hls4_Cat3   3.927492901 0.8526291
## Hls5_Cat1  -1.684131262 0.1781274
## Hls5_Cat2   1.381933373 0.1467223
## Hls5_Cat3   3.897482578 0.3943440
## Hls6_Cat1  -1.155351142 0.1543659
## Hls6_Cat2   2.939344680 0.2113184
## Hls6_Cat3   3.671463609 0.5189345
## Hls7_Cat1  -0.165823435 0.1358793
## Hls7_Cat2   3.753328170 0.2949727
## Hls7_Cat3   3.776319530 0.6885511
## Hls8_Cat1  -2.135935885 0.1992731
## Hls8_Cat2   0.896643565 0.1385362
## Hls8_Cat3   3.581083599 0.3235907
## Hls9_Cat1  -0.008019089 0.1360316
## Hls9_Cat2   2.989853095 0.2288433
## Hls9_Cat3   3.313295753 0.4819018
## Hls10_Cat1 -0.183297684 0.1360561
## Hls10_Cat2  3.694746057 0.2921816
## Hls10_Cat3  4.379242422 0.8215605
## Hls11_Cat1 -2.106058995 0.2097751
## Hls11_Cat2  0.172650186 0.1377271
## Hls11_Cat3  2.907183948 0.2353937
## Hls12_Cat1 -2.317865929 0.2117123
## Hls12_Cat2  0.513325662 0.1362435
## Hls12_Cat3  3.342199604 0.2821645
```

```
## Hls13_Cat1 -2.127336182 0.1938394
## Hls13_Cat2  1.338677184 0.1444056
## Hls13_Cat3  4.275574749 0.4493420
## Hls14_Cat1 -1.761463128 0.1777770
## Hls14_Cat2  1.762102813 0.1550751
## Hls14_Cat3  3.208316423 0.3314447
## Hls15_Cat1 -2.042459839 0.1969236
## Hls15_Cat2  0.810466042 0.1380178
## Hls15_Cat3  3.608588261 0.3230206
## Hls16_Cat1 -1.628691156 0.1721913
## Hls16_Cat2  1.906727817 0.1604086
## Hls16_Cat3  4.024764112 0.4574405
```

```
ItemDiff2 <- mod2$xsi$xsi
ItemDiff2
```

```
##  [1] -1.846046710  2.298334842  3.830385868 -1.501715752  2.764560868
##  [6]  4.958324889 -0.380834628  3.962213547  5.127428610  0.175976124
## [11]  4.324558567  3.927492901 -1.684131262  1.381933373  3.897482578
## [16] -1.155351142  2.939344680  3.671463609 -0.165823435  3.753328170
## [21]  3.776319530 -2.135935885  0.896643565  3.581083599 -0.008019089
## [26]  2.989853095  3.313295753 -0.183297684  3.694746057  4.379242422
## [31] -2.106058995  0.172650186  2.907183948 -2.317865929  0.513325662
## [36]  3.342199604 -2.127336182  1.338677184  4.275574749 -1.761463128
## [41]  1.762102813  3.208316423 -2.042459839  0.810466042  3.608588261
## [46] -1.628691156  1.906727817  4.024764112
```

```
#note, if you want to see this in your viewer, you can also use View().
```

Person ability (theta) estimates

```
person.ability.poly <- tam.wle(mod2)
```

```
## Iteration in WLE/MLE estimation  1   | Maximal change  2.6967
## Iteration in WLE/MLE estimation  2   | Maximal change  2.1777
## Iteration in WLE/MLE estimation  3   | Maximal change  0.368
## Iteration in WLE/MLE estimation  4   | Maximal change  0.0135
## Iteration in WLE/MLE estimation  5   | Maximal change  3e-04
## Iteration in WLE/MLE estimation  6   | Maximal change  0
## ----
##  WLE Reliability= 0.9
```

```
head(person.ability.poly)
```

```
## Object of class 'tam.wle'
## Call: tam.wle(tamobj = mod2)
##
##   WLEs for 317 observations and 1 dimension
##
##   WLE Reliability=0.9
##   Average error variance=0.307
##   WLE mean=-0.02
##   WLE variance=3.071
```

Item fit statistics

```r
Fit.poly <- tam.fit(mod2)
```

```
## Item fit calculation based on 100 simulations
## |**********|
## |----------|
```

Fit.poly$itemfit

```r
kable(Fit.poly$itemfit)
```

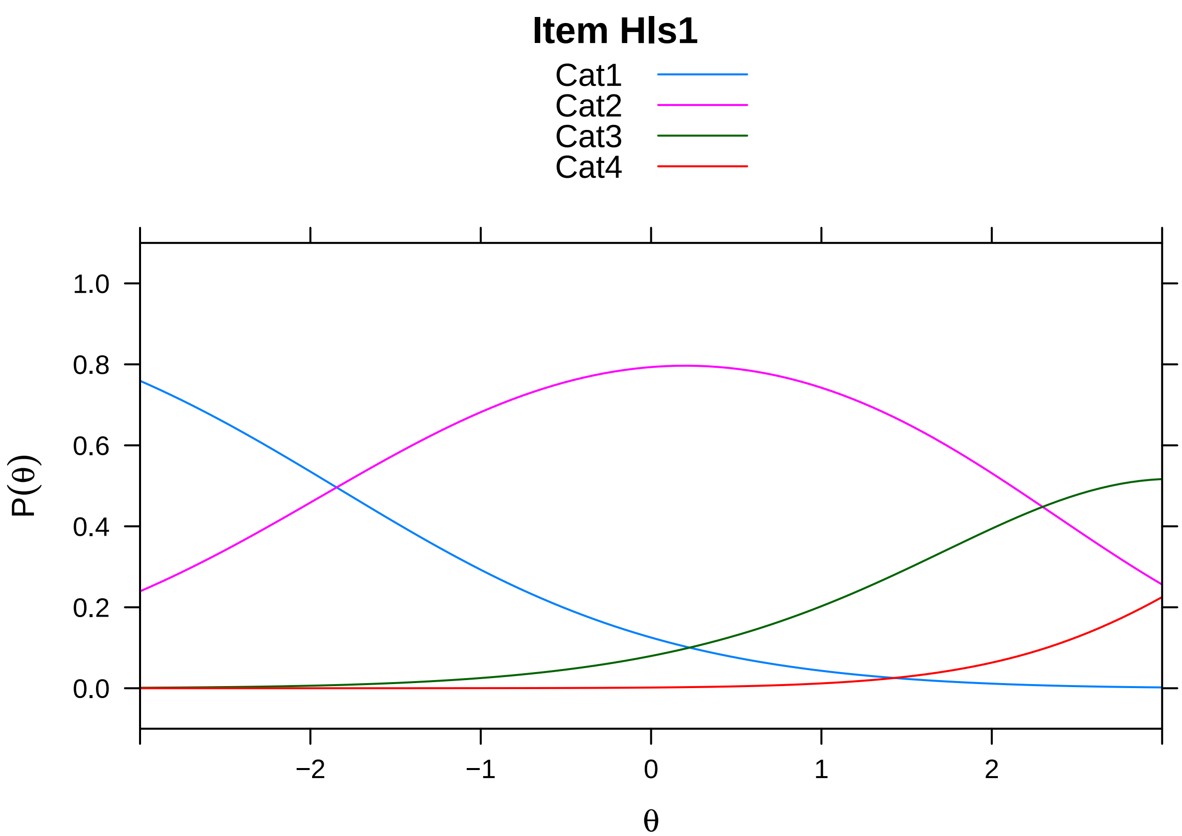
parameter	Outfit	Outfit_t	Outfit_p	Outfit_pholm	Infit	Infit_t	Infit_p	Infit_pho
Hls1_Cat1	3.5099428	12.6468060	0.0000000	0.0000000	1.0315791	0.3168336	0.7513698	
Hls1_Cat2	4.0679640	15.0800196	0.0000000	0.0000000	1.1248225	1.2010932	0.2297151	
Hls1_Cat3	3960.6060497	89.0796806	0.0000000	0.0000000	0.9624986	0.0158218	0.9873765	
Hls2_Cat1	20.7651487	47.8333283	0.0000000	0.0000000	1.0738025	0.7694442	0.4416296	
Hls2_Cat2	0.9188821	-0.6233827	0.5330330	1.0000000	1.0405502	0.3366623	0.7363715	
Hls2_Cat3	0.8360986	-0.2459324	0.8057346	1.0000000	1.3874375	0.7569263	0.4490940	
Hls3_Cat1	0.9338249	-1.0609081	0.2887317	1.0000000	0.9606121	-0.6099498	0.5418951	
Hls3_Cat2	0.8325491	-0.6691008	0.5034312	1.0000000	0.9070141	-0.2886781	0.7728277	
Hls3_Cat3	0.0140062	-2.4509223	0.0142491	0.3704759	0.8401390	0.0301593	0.9759400	
Hls4_Cat1	0.8610097	-2.6691771	0.0076037	0.2205083	0.9262990	-1.3727888	0.1698180	
Hls4_Cat2	0.6403760	-1.3257322	0.1849284	1.0000000	0.8834026	-0.2913151	0.7708104	
Hls4_Cat3	0.0106654	-3.4864353	0.0004895	0.0171326	0.4191678	-1.0265639	0.3046259	
Hls5_Cat1	0.8346624	-1.7940411	0.0728066	1.0000000	0.8418430	-1.5173829	0.1291700	
Hls5_Cat2	1.5522438	6.4456435	0.0000000	0.0000000	1.0785033	1.1498713	0.2501969	
Hls5_Cat3	1.9047738	1.8603985	0.0628292	1.0000000	1.0668029	0.3034766	0.7615267	
Hls6_Cat1	0.9372606	-0.7650567	0.4442378	1.0000000	0.8679647	-1.5868662	0.1125430	
Hls6_Cat2	1.3559125	2.0659682	0.0388315	0.8931240	1.0196059	0.1761030	0.8602130	
Hls6_Cat3	2.9291135	2.8075561	0.0049919	0.1597408	1.2954841	0.7826149	0.4338533	
Hls7_Cat1	0.8345010	-2.9353241	0.0033320	0.1099558	0.9097382	-1.5438804	0.1226173	
Hls7_Cat2	0.5428751	-2.3754463	0.0175277	0.4381936	0.9019374	-0.3716201	0.7101757	
Hls7_Cat3	0.8458629	-0.5671425	0.5706174	1.0000000	1.1414942	0.4112924	0.6808582	
Hls8_Cat1	0.6356179	-3.3814029	0.0007212	0.0245197	0.8436515	-1.2897225	0.1971470	
Hls8_Cat2	2.4388729	13.2713003	0.0000000	0.0000000	1.0764485	1.3103106	0.1900908	
Hls8_Cat3	1.3798180	1.0980025	0.2722034	1.0000000	0.9492692	-0.1260393	0.8997008	
Hls9_Cat1	0.8887407	-1.9755524	0.0482055	1.0000000	0.9513511	-0.8296680	0.4067265	
Hls9_Cat2	1.5042831	2.5715405	0.0101247	0.2834921	1.0811179	0.5446293	0.5860085	
Hls9_Cat3	0.7745219	-0.8537194	0.3932605	1.0000000	0.9354260	-0.0576325	0.9540414	
Hls10_Cat1	1.0188431	0.3044023	0.7608214	1.0000000	1.0152305	0.2641651	0.7916527	
Hls10_Cat2	86.8095905	35.4191808	0.0000000	0.0000000	1.1405478	0.6565621	0.5114625	
Hls10_Cat3	0.6552080	-0.8472339	0.3968648	1.0000000	1.0200217	0.2286678	0.8191271	
Hls11_Cat1	0.8018616	-1.7893762	0.0735543	1.0000000	0.8538543	-1.1487867	0.2506439	
Hls11_Cat2	0.9301360	-1.2435350	0.2136707	1.0000000	1.0233912	0.4102879	0.6815948	
Hls11_Cat3	2.1724986	4.6474536	0.0000034	0.0001243	1.0011438	0.0553416	0.9558663	
Hls12_Cat1	0.7092339	-2.6812690	0.0073344	0.2200306	0.7313001	-2.2015447	0.0276975	
Hls12_Cat2	1.0592966	0.8902006	0.3733582	1.0000000	0.9908589	-0.1538375	0.8777379	
Hls12_Cat3	3.3486457	6.0327213	0.0000000	0.0000001	1.0400848	0.2499802	0.8026027	
Hls13_Cat1	0.9611484	-0.4742819	0.6352989	1.0000000	0.7708498	-2.0119398	0.0442263	
Hls13_Cat2	1.0062064	0.0852376	0.9320725	1.0000000	1.0505484	0.7766358	0.4373736	
Hls13_Cat3	2.4389935	2.2508934	0.0243923	0.5854149	0.9039416	-0.1505466	0.8803334	
Hls14_Cat1	0.8239855	-1.7423115	0.0814539	1.0000000	0.9063179	-0.8613223	0.3890606	
Hls14_Cat2	2.0910391	9.6300006	0.0000000	0.0000000	1.0792708	1.0044574	0.3151582	
Hls14_Cat3	0.5881476	-2.0216287	0.0432147	0.9507239	0.9707354	-0.0425327	0.9660740	
Hls15_Cat1	0.7292479	-2.4920071	0.0127023	0.3429634	0.8114747	-1.6090098	0.1076142	
Hls15_Cat2	1.1101321	1.8550086	0.0635950	1.0000000	1.0759142	1.3207173	0.1865956	
Hls15_Cat3	4.9768611	6.8811888	0.0000000	0.0000000	0.9591085	-0.0868033	0.9308279	
Hls16_Cat1	1.5516620	4.4227489	0.0000097	0.0003508	0.9442892	-0.5172678	0.6049693	
Hls16_Cat2	1.9544537	8.2381738	0.0000000	0.0000000	1.0625095	0.7399902	0.4593059	
Hls16_Cat3	0.2891700	-2.7089202	0.0067503	0.2092580	0.8162838	-0.4153244	0.6779045	

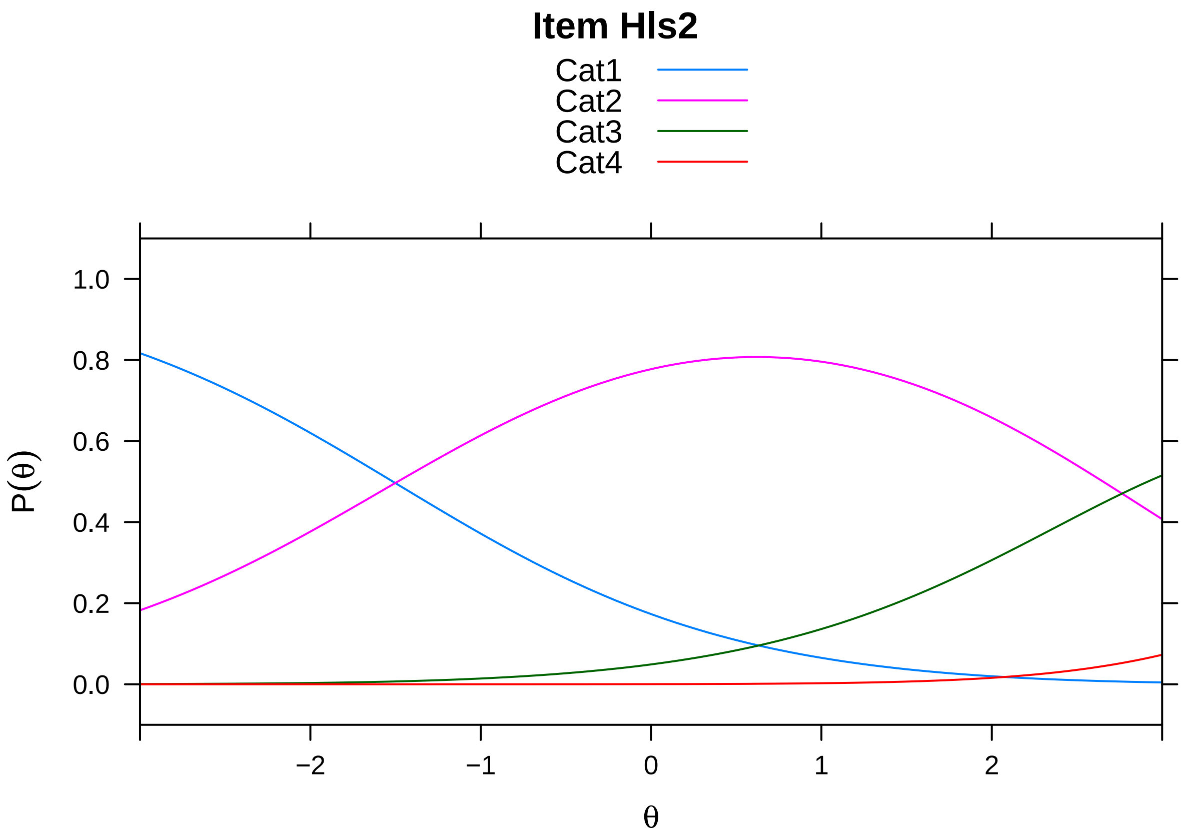
Item characteristic curves (but now as thresholds).

There are item characteristic curves (ICCs) for each item choice

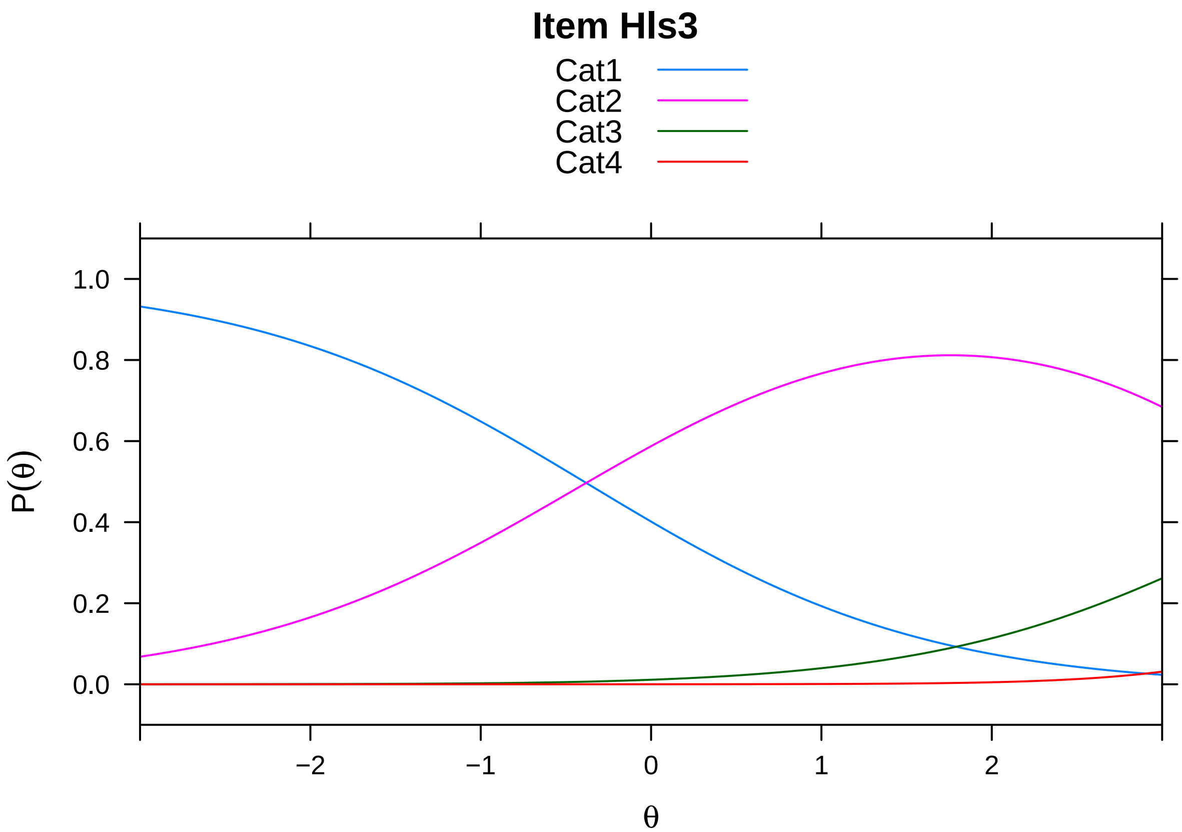
```r
tthresh.poly <- tam.threshold(mod2)
plot(mod2, type = "items")
```

```
## Iteration in WLE/MLE estimation  1   | Maximal change  2.6967
## Iteration in WLE/MLE estimation  2   | Maximal change  2.1777
## Iteration in WLE/MLE estimation  3   | Maximal change  0.368
## Iteration in WLE/MLE estimation  4   | Maximal change  0.0135
## Iteration in WLE/MLE estimation  5   | Maximal change  3e-04
## Iteration in WLE/MLE estimation  6   | Maximal change  0
## ----
##  WLE Reliability= 0.9
```

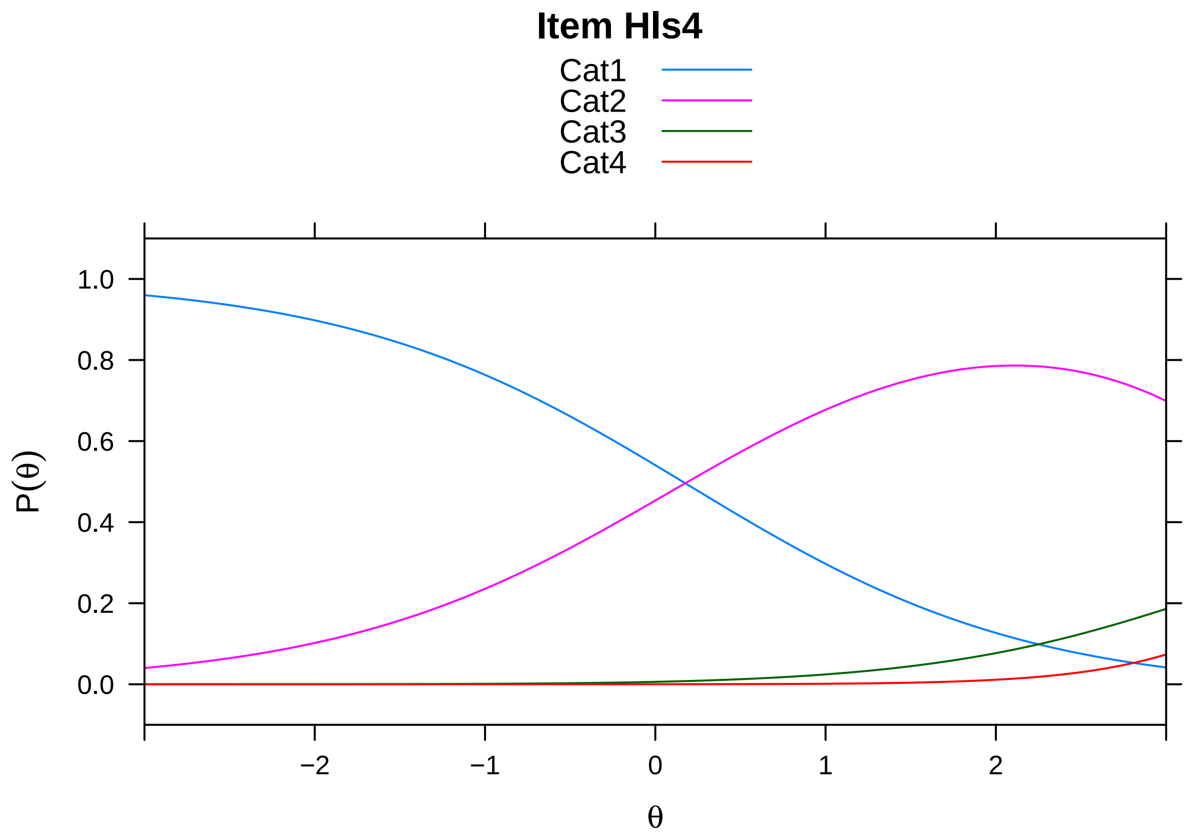


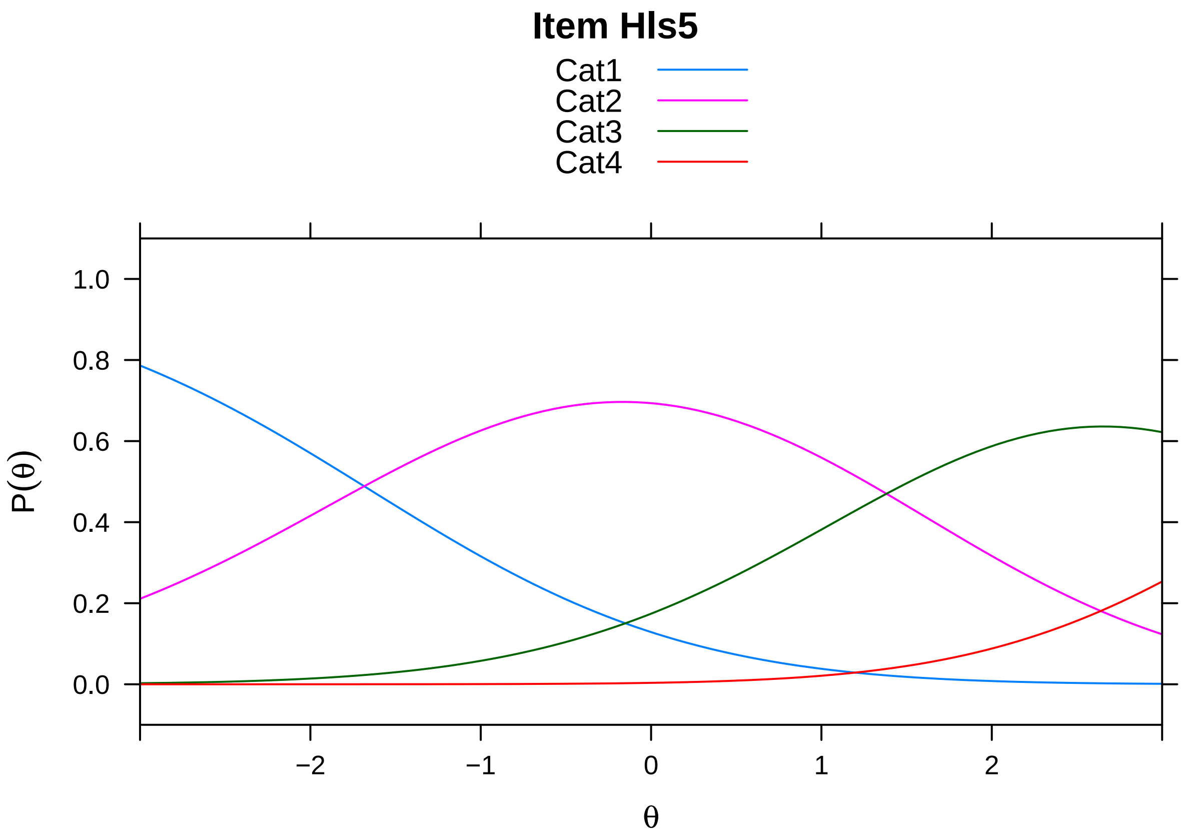


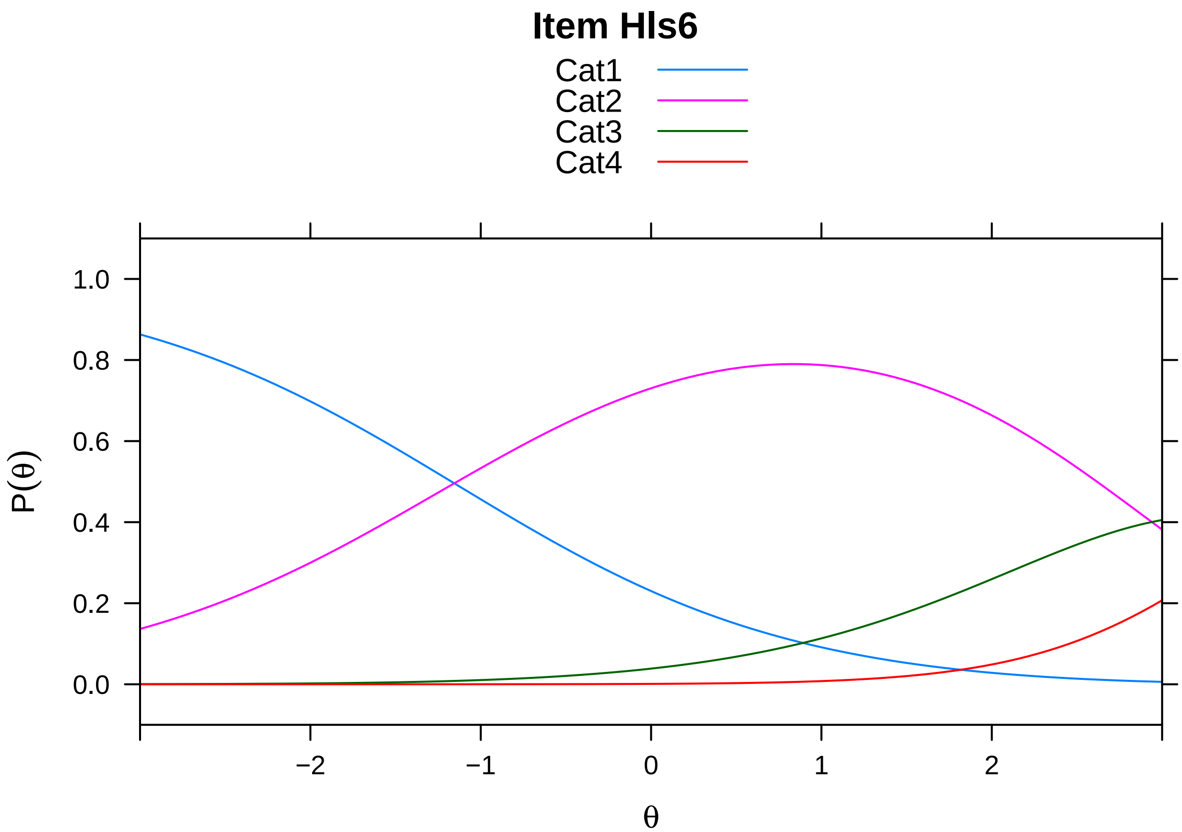
Item Hls2
Cat1
Cat2
Cat3
Cat4
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−2
−1
0
1
2
θ



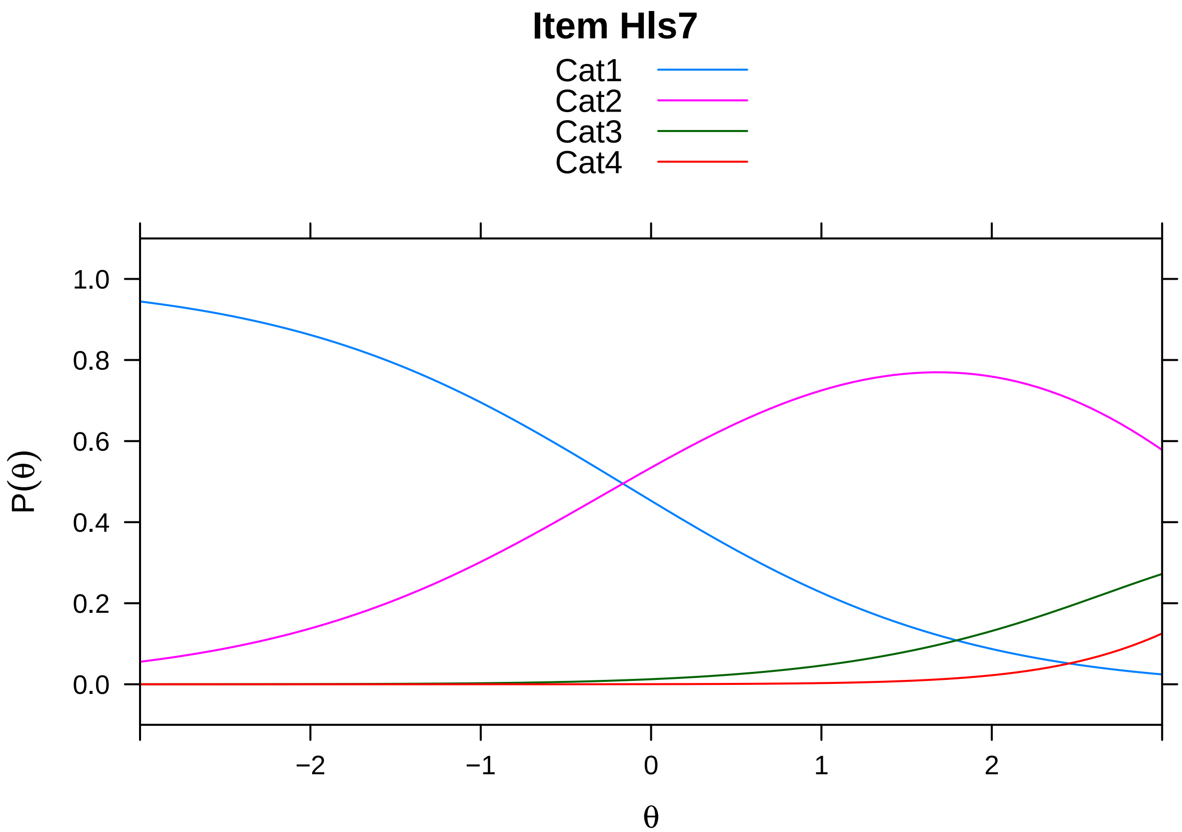
Item Hls3
Cat1
Cat2
Cat3
Cat4
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−2
−1
0
1
2
θ

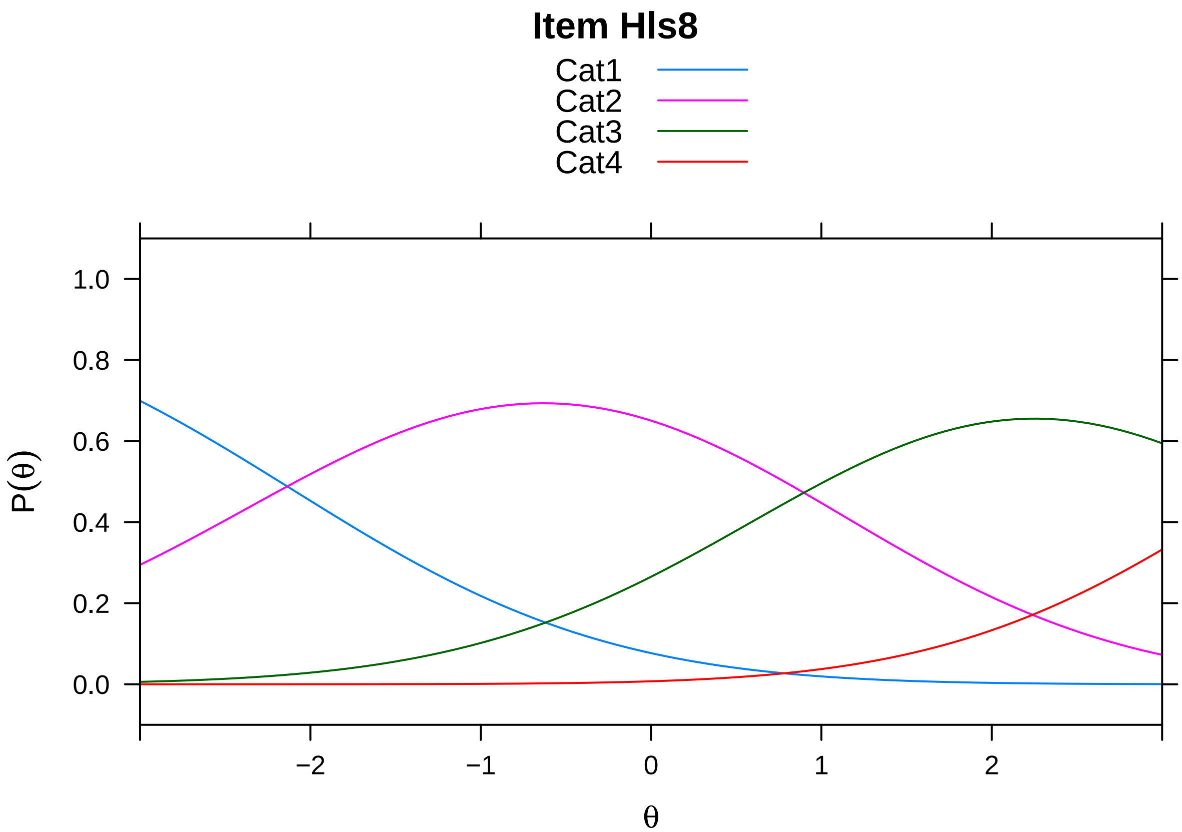

Item Hls4
Cat1
Cat2
Cat3
Cat4
P(θ)
1.0
0.8
0.6
0.4
0.2
0.0
−2
−1
0
1
2
θ

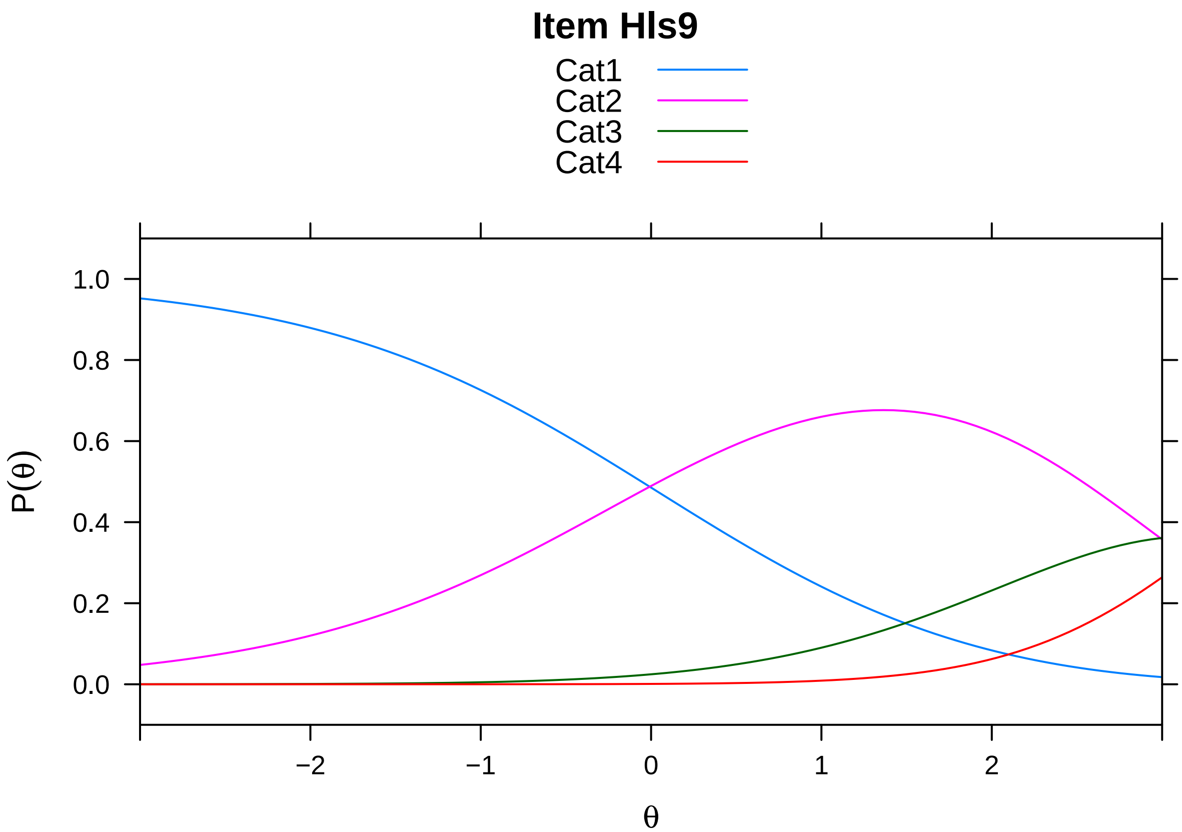

Item Hls5
Cat1
Cat2
Cat3
Cat4
P(θ)
1.0
0.8
0.6
0.4
0.2
0.0
−2
−1
0
1
2
θ

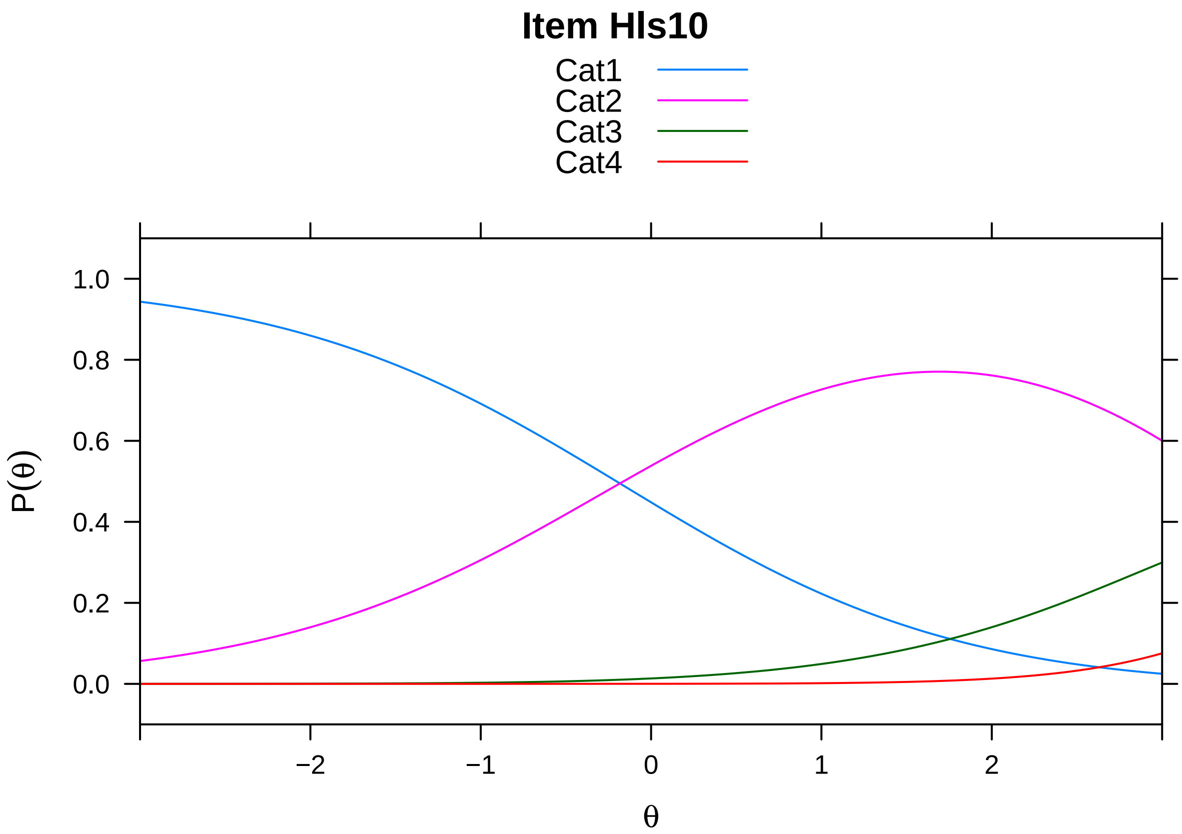


Item Hls6
Cat1
Cat2
Cat3
Cat4
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−2
−1
0
1
2
θ

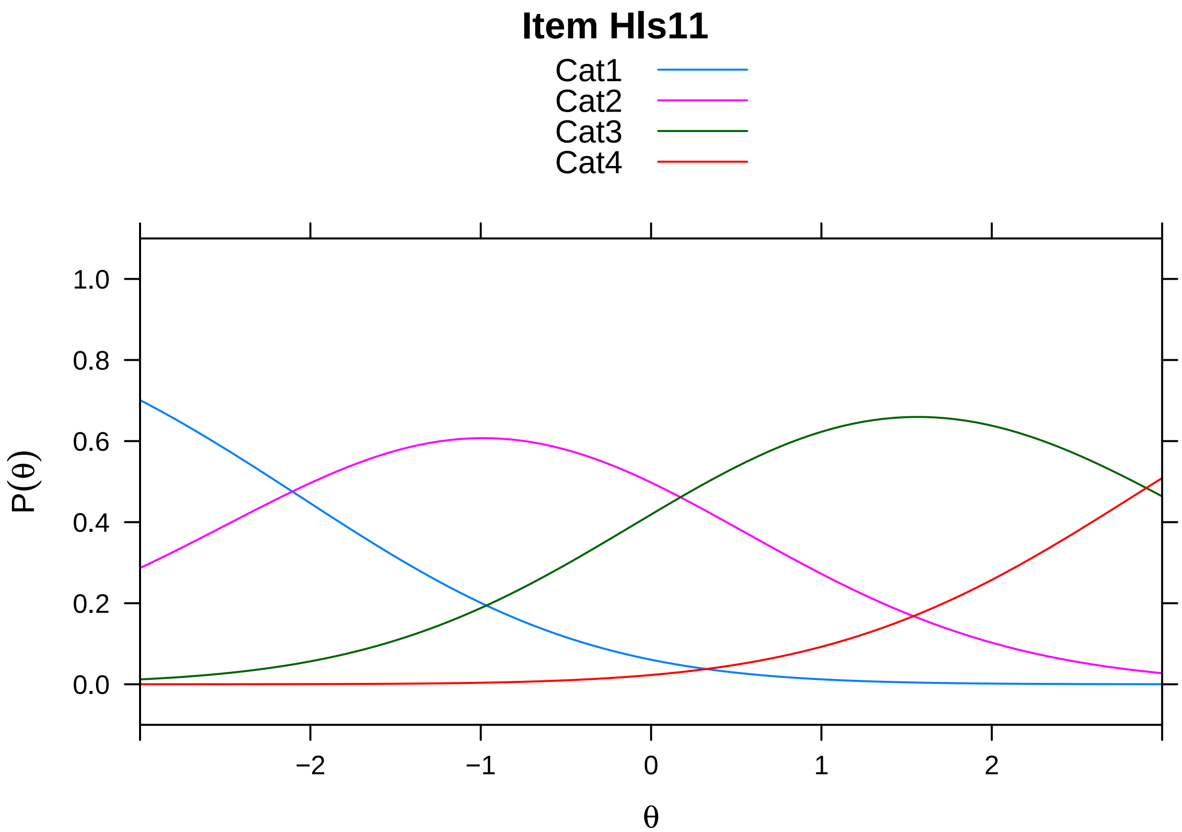

Item Hls7
Cat1
Cat2
Cat3
Cat4
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−2
−1
0
1
2
θ

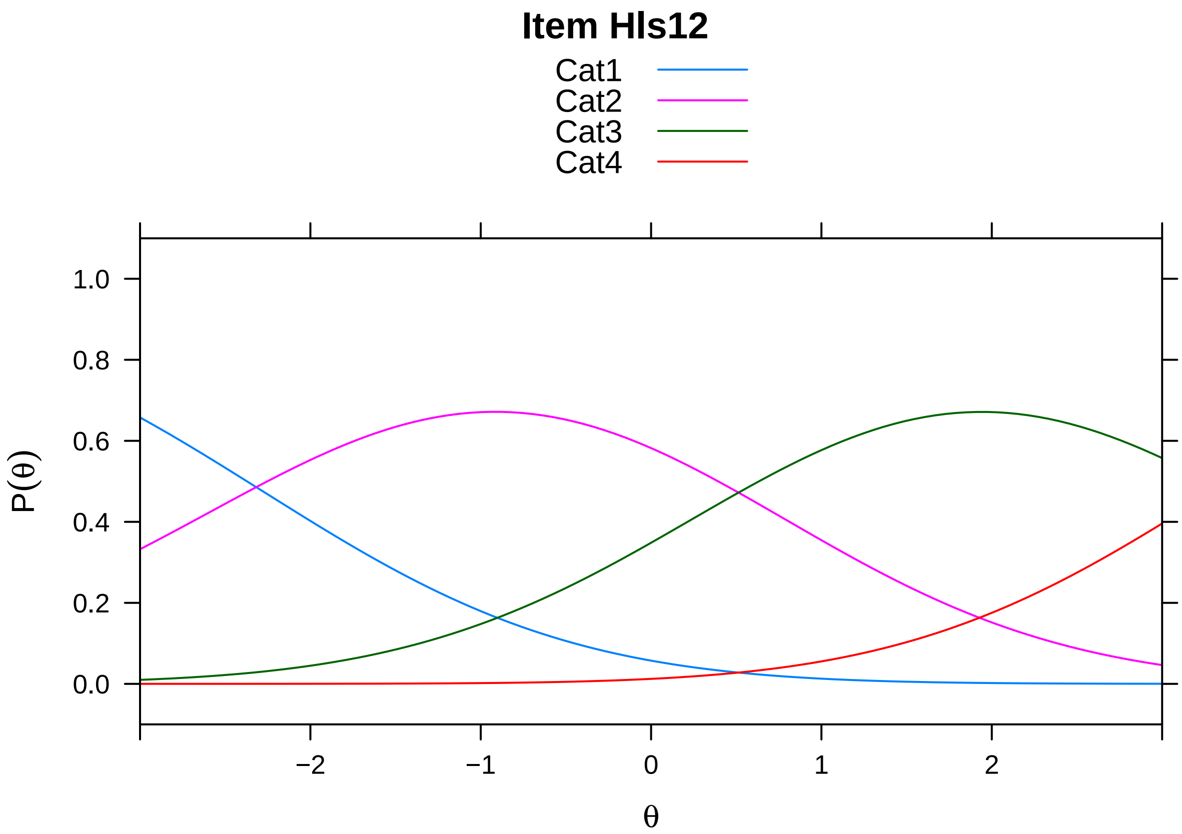

Item Hls8
Cat1
Cat2
Cat3
Cat4
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−2
−1
0
1
2
θ


Item Hls9
Cat1
Cat2
Cat3
Cat4
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−2
−1
0
1
2
θ

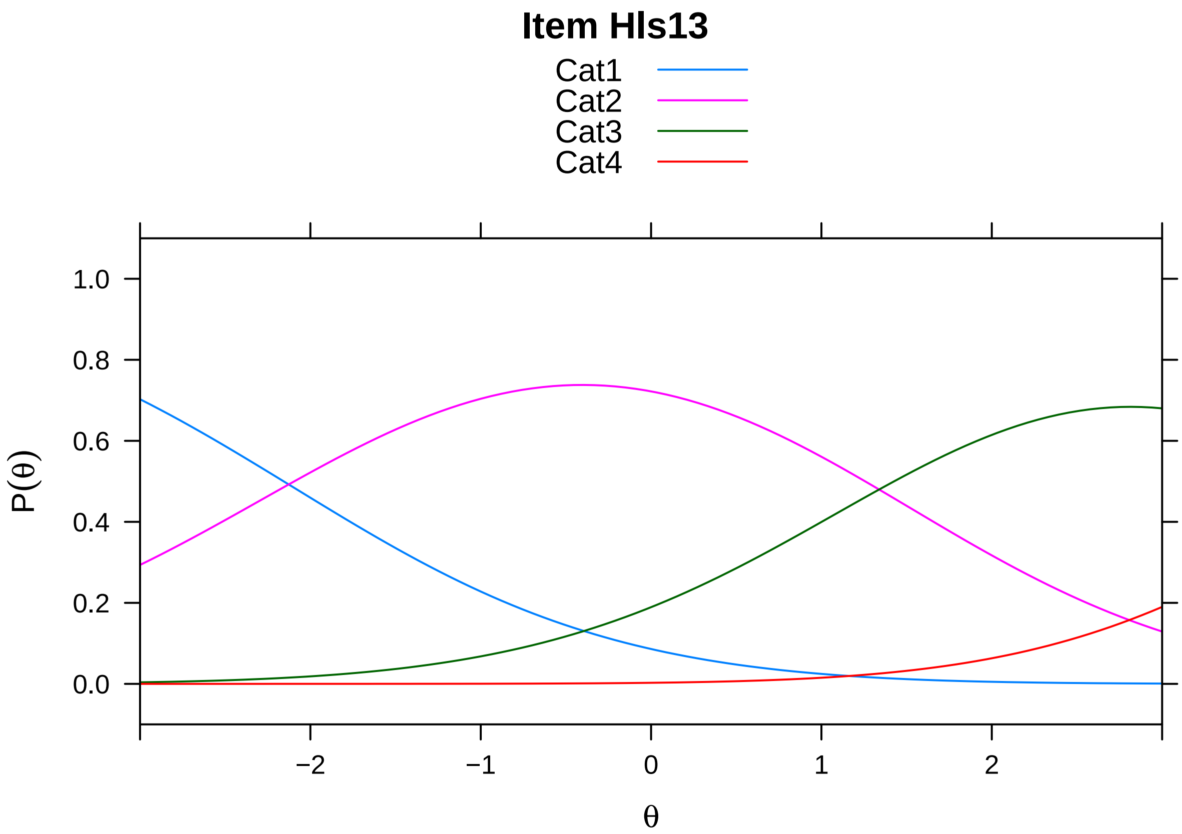


Item Hls10
Cat1
Cat2
Cat3
Cat4
P(θ)
1.0
0.8
0.6
0.4
0.2
0.0
−2
−1
0
1
2
θ

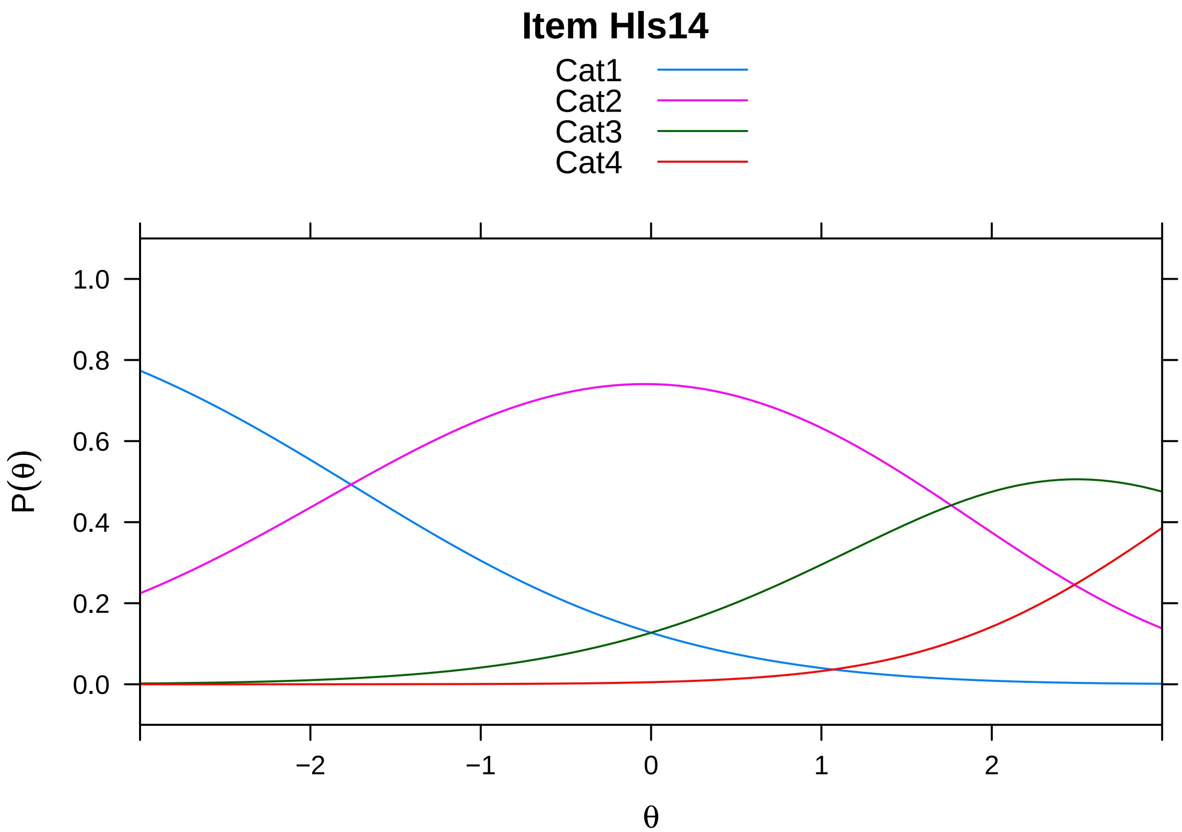

Item Hls11
Cat1
Cat2
Cat3
Cat4
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−2
−1
0
1
2
θ



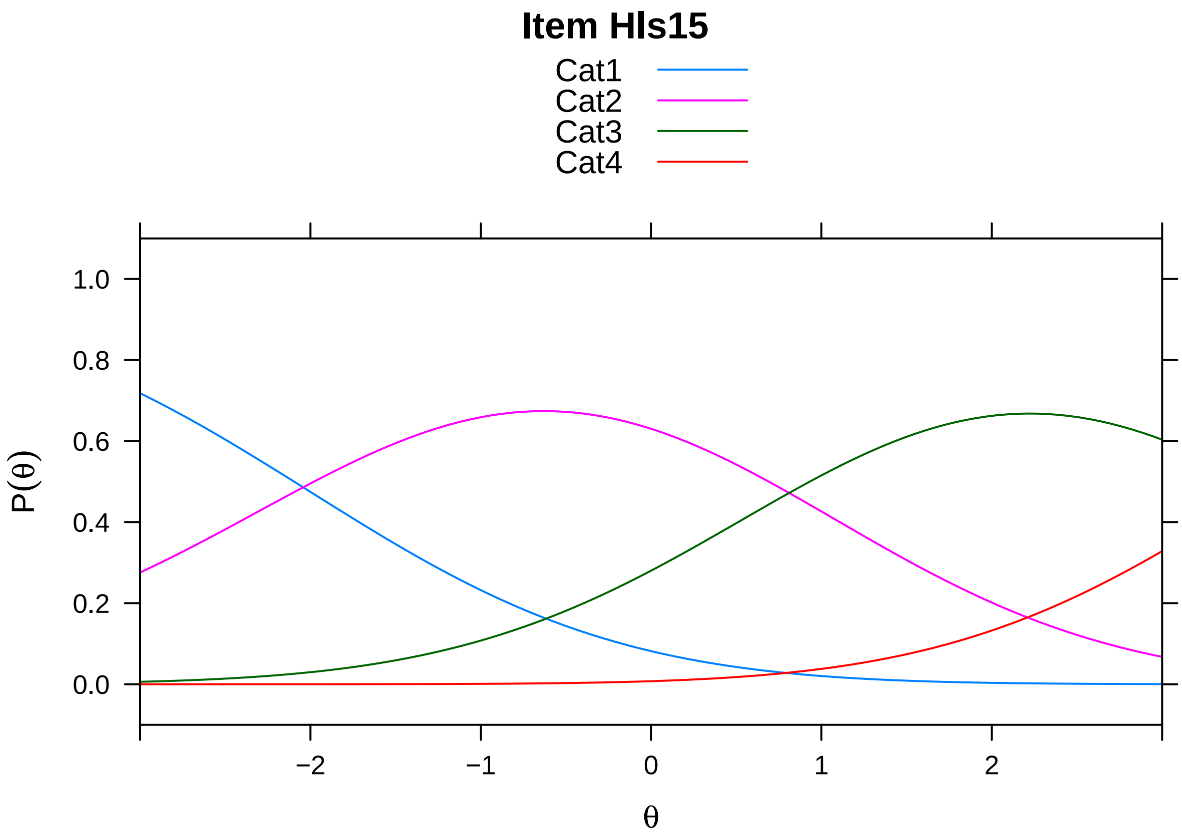
Item Hls12
Cat1
Cat2
Cat3
Cat4
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−2
−1
0
1
2
θ



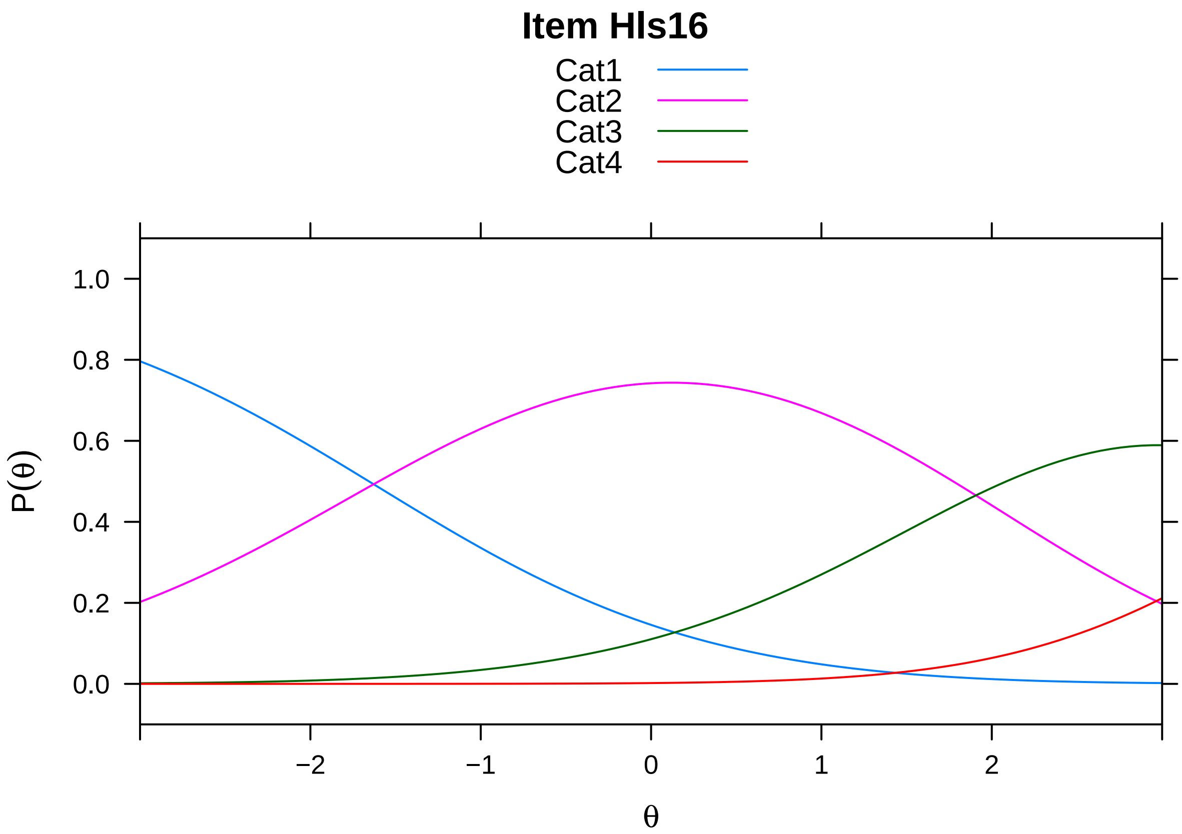
Item Hls13
Cat1
Cat2
Cat3
Cat4
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−2
−1
0
1
2
θ



Item Hls14
Cat1
Cat2
Cat3
Cat4
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−2
−1
0
1
2
θ



Item Hls15
Cat1
Cat2
Cat3
Cat4
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−2
−1
0
1
2
θ

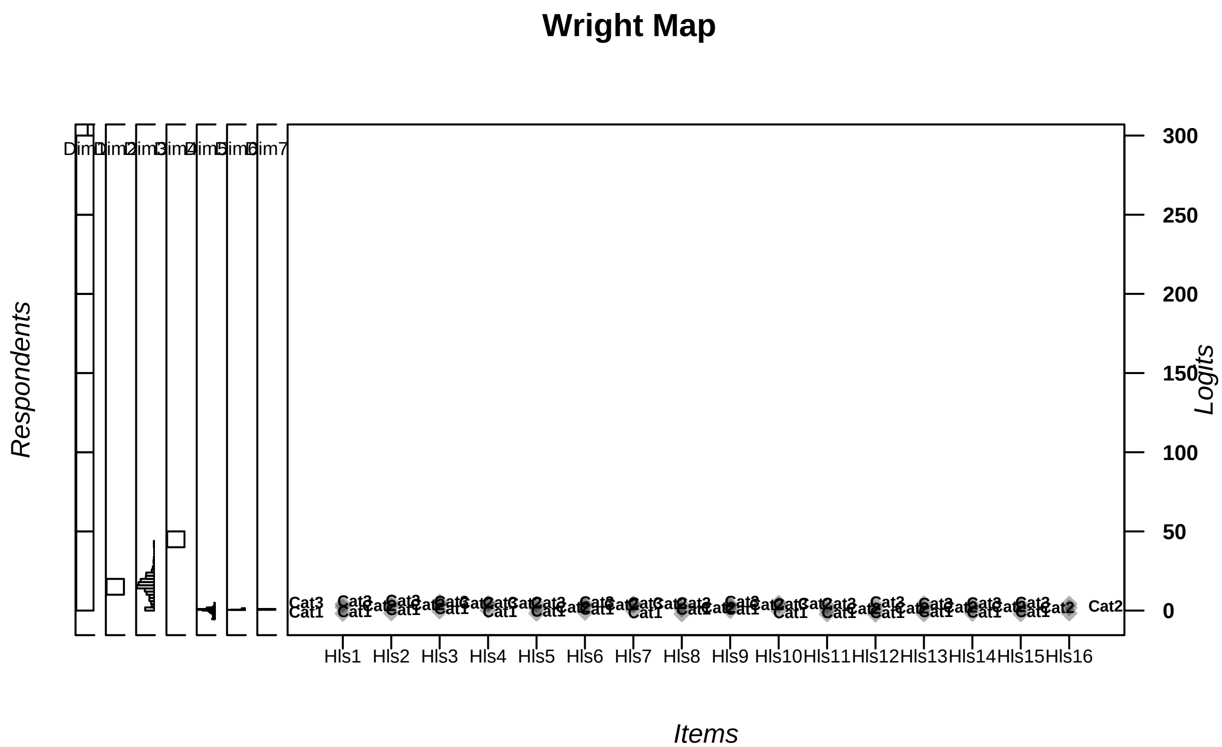



```
## ...................................................
##  Plots exported in png format into folder:
##  C:/Users/katzd/Desktop/Rprojects/DBER_plain_biome/DBER_Rasch/Plots
```

Wright Map

Here's a polytomous Wright Map

```
wrightMap(person.ability.poly, tthresh.poly)
```



```
##                Cat1      Cat2     Cat3
## Hls1  -1.86154175 2.1462708 3.998566
## Hls2  -1.51547241 2.6820374 5.054901
## Hls3  -0.39358521 3.7526550 5.350800
## Hls4   0.16012573 3.7458801 4.529205
## Hls5  -1.72787476 1.3532410 3.970184
## Hls6  -1.17178345 2.6529236 3.976410
## Hls7  -0.18539429 3.3007507 4.254547
## Hls8  -2.18106079 0.8795471 3.643341
## Hls9  -0.05612183 2.6443176 3.715851
## Hls10 -0.20352173 3.4025574 4.694550
## Hls11 -2.19607544 0.2026062 2.966949
## Hls12 -2.37240601 0.5131531 3.396881
## Hls13 -2.15725708 1.3193665 4.324860
## Hls14 -1.78994751 1.6114197 3.388824
## Hls15 -2.09591675 0.8077698 3.664764
## Hls16 -1.65664673 1.8318787 4.128021
```

Exercises:

1. Find an item for which Cat 3 is actually easier than the Cat 2 of another item.
2. Find an item that has two categories that are extremely close in severity.
3. Look at the ICC for item 14. Describe what is happening with Cat 3.

Model Comparison

say we want to compare the two models we just ran (note, these aren't really comparable since it's a completely different model - not nested data)

```
logLik(mod1)
```

```
## 'log Lik.' -1880.882 (df=17)
```

```
logLik(mod2)
```

```
## 'log Lik.' -4185.626 (df=49)
```

```
anova(mod1, mod2)
```

```
##   Model   loglike Deviance Npars      AIC      BIC     Chisq df  p
## 1  mod1 -1880.882 3761.763    17 3795.763 3859.665 -4609.489 32  1
## 2  mod2 -4185.626 8371.252    49 8469.252 8653.438        NA NA NA
```

Log likelihood is the foundation of both AIC and BIC. AIC and BIC allow you to compare non-nested models while penalizing for model complexity (BIC penalizes more). In general, the model with a smaller AIC/BIC is the one that the data fit better. The two criteria sometimes disagree.

Multidimensional Rasch Model

What if we envision something that's multidimensional? We can model that with TAM. IN fact, this is one of TAM's great strengths. Do read package documentation, though. As the number of dimensions grows, you'll have to use particular estimation methods else the model will take to long to run.

we start by assigning the items to a dimension using a Q-matrix

If we want to have two dimensions, we'll create a matrix with two columns. A 1 or 0 denotes whether that item belongs to dimension 1 or 2 (or both!)

```
Q <-
  matrix(c(1,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0
           ,0),ncol=2)

Q
```

```
##       [,1] [,2]
##  [1,]    1    0
##  [2,]    0    1
##  [3,]    0    1
##  [4,]    0    1
##  [5,]    0    1
##  [6,]    0    1
##  [7,]    0    1
##  [8,]    1    0
##  [9,]    1    0
## [10,]    1    0
## [11,]    1    0
## [12,]    1    0
## [13,]    1    0
## [14,]    1    0
## [15,]    1    0
## [16,]    1    0
```

click on the “Q” object in the environment pane to see what we just made

Run the multidimensional Rasch model

```
multi <- TAM::tam.mml(resp=hls, Q=Q)
```

θ and δ

```
persons.multi <- tam.wle(multi)
```

```
## Iteration in WLE/MLE estimation  1   | Maximal change  2.6468
## Iteration in WLE/MLE estimation  2   | Maximal change  1.3204
## Iteration in WLE/MLE estimation  3   | Maximal change  2.3637
## Iteration in WLE/MLE estimation  4   | Maximal change  0.4899
## Iteration in WLE/MLE estimation  5   | Maximal change  0.0964
## Iteration in WLE/MLE estimation  6   | Maximal change  0.0224
## Iteration in WLE/MLE estimation  7   | Maximal change  0.0071
## Iteration in WLE/MLE estimation  8   | Maximal change  0.0023
## Iteration in WLE/MLE estimation  9   | Maximal change  7e-04
## Iteration in WLE/MLE estimation  10   | Maximal change  2e-04
## Iteration in WLE/MLE estimation  11   | Maximal change  1e-04
##
## -------
## WLE Reliability (Dimension1)=0.548
## WLE Reliability (Dimension2)=-0.683
```

```
WLEestimates.multi <- persons.multi$theta
thresholds.multi <- tam.threshold(multi)
```

```
#Fit and reliabilities
Fit.multi <- tam.fit(multi)
```

```
## Item fit calculation based on 100 simulations
## |**********|
## |----------|
```

```
Fit.multi$itemfit
```

```
##    parameter    Outfit   Outfit_t     Outfit_p Outfit_pholm     Infit
## 1       Hls1 1.4233683  3.4482889 5.641503e-04 0.0084622546 1.1524232
## 2       Hls2 1.1498487  0.7663490 4.434687e-01 1.0000000000 1.0719542
## 3       Hls3 0.7767545 -1.3287783 1.839211e-01 1.0000000000 0.9090698
## 4       Hls4 0.3603962 -3.0756936 2.100135e-03 0.0273017559 0.8406098
## 5       Hls5 0.9052404 -1.4444913 1.486008e-01 1.0000000000 1.0019654
## 6       Hls6 0.6984289 -2.5142231 1.192949e-02 0.1431539207 0.9410298
## 7       Hls7 0.7969676 -1.2504172 2.111472e-01 1.0000000000 0.9313144
## 8       Hls8 1.0500456  0.6403940 5.219165e-01 1.0000000000 1.0680429
## 9       Hls9 1.3979713  2.0890578 3.670252e-02 0.3670251803 0.9765253
## 10     Hls10 1.3293173  1.0559681 2.909828e-01 1.0000000000 1.0920158
## 11     Hls11 0.8668431 -2.1794481 2.929840e-02 0.3222823807 0.9151552
## 12     Hls12 0.7539335 -4.2177287 2.467754e-05 0.0003948407 0.8463454
## 13     Hls13 0.9302283 -1.0329475 3.016285e-01 1.0000000000 1.0409923
## 14     Hls14 1.1321535  1.4036866 1.604122e-01 1.0000000000 1.0085641
## 15     Hls15 0.8044582 -3.2178279 1.291653e-03 0.0180831408 0.9382870
## 16     Hls16 1.0113582 -0.0256253 9.795562e-01 1.0000000000 1.0160832
```

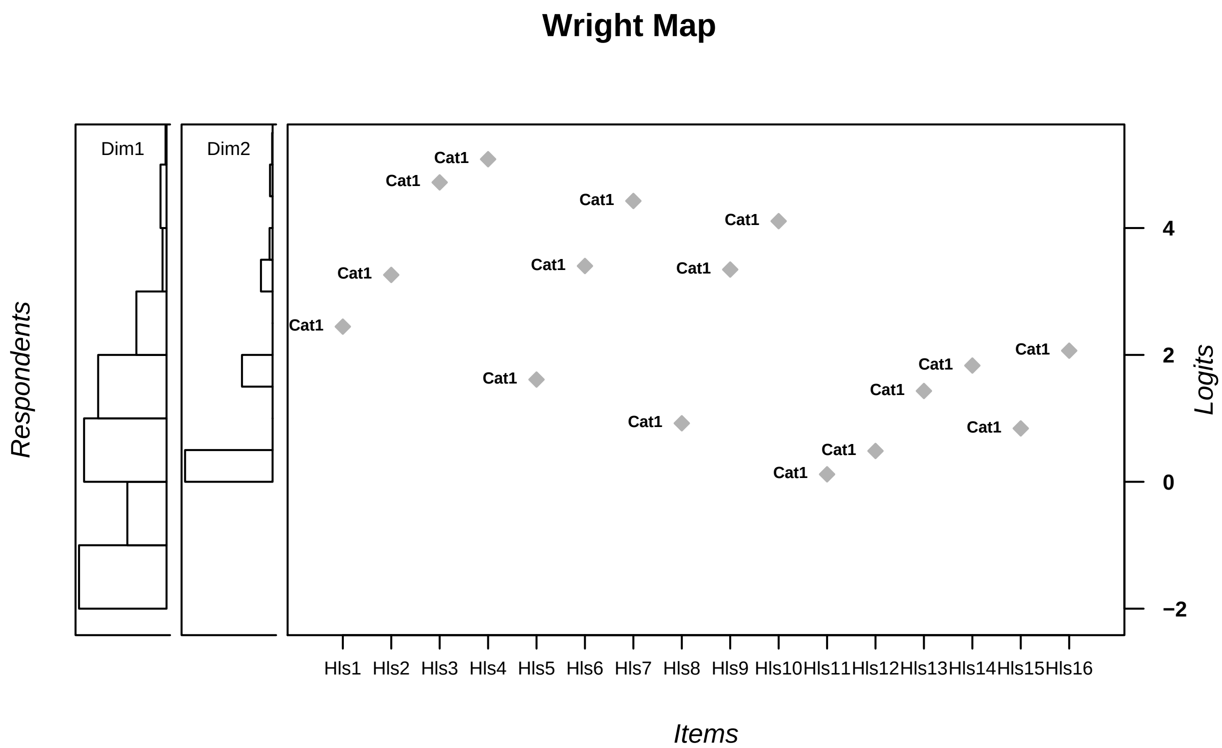
```
##          Infit_t    Infit_p Infit_pholm
## 1   1.43226025 0.15206936   1.0000000
## 2   0.55311662 0.58018356   1.0000000
## 3  -0.30991905 0.75662252   1.0000000
## 4  -0.50653070 0.61248414   1.0000000
## 5   0.03023834 0.97587697   1.0000000
## 6  -0.39027055 0.69633650   1.0000000
## 7  -0.26638310 0.78994418   1.0000000
## 8   1.01588004 0.30968652   1.0000000
## 9  -0.10842681 0.91365713   1.0000000
## 10  0.47405395 0.63546143   1.0000000
## 11 -1.34061339 0.18004601   1.0000000
## 12 -2.51814763 0.01179739   0.1887582
## 13  0.56984306 0.56878414   1.0000000
## 14  0.12025336 0.90428245   1.0000000
## 15 -0.95549596 0.33932695   1.0000000
## 16  0.19921390 0.84209542   1.0000000
```

```
multi$EAP.rel #EAP reliabilities
```

```
##      Dim1      Dim2
## 0.7648527 0.6680034
```

Wright Map

```
MDthetas.multi <-
  cbind(persons.multi$theta.Dim01,persons.multi$theta.Dim02) #one line
wrightMap(MDthetas.multi, thresholds.multi) #second line
```




```
##          Cat1
## Hls1  2.4458313
## Hls2  3.2610168
## Hls3  4.7192688
## Hls4  5.0847473
## Hls5  1.6114197
## Hls6  3.4018250
## Hls7  4.4266663
## Hls8  0.9223938
## Hls9  3.3454285
## Hls10 4.1084290
## Hls11 0.1181946
## Hls12 0.4860535
## Hls13 1.4316101
## Hls14 1.8324280
## Hls15 0.8414612
## Hls16 2.0666199
```

Compare the first unidimensional model to the multidimensional one

```r
logLik(mod1)
```

```
## 'log Lik.' -1880.882 (df=17)
```

```r
logLik(multi)
```

```
## 'log Lik.' -1873.238 (df=19)
```

```r
anova(mod1, multi)
```

```
##   Model   loglike Deviance Npars      AIC      BIC    Chisq df       p
## 1  mod1 -1880.882 3761.763    17 3795.763 3859.665 15.28774  2 0.00048
## 2 multi -1873.238 3746.476    19 3784.476 3855.895       NA NA      NA
```

Alternatively, you can use `IRT.compareModels`

```r
compare <- CDM::IRT.compareModels(mod1, multi)
compare
```

```
## $IC
##   Model   loglike Deviance Npars Nobs      AIC      BIC     AIC3     AICc
## 1  mod1 -1880.882 3761.763    17  317 3795.763 3859.665 3812.763 3797.810
## 2 multi -1873.238 3746.476    19  317 3784.476 3855.895 3803.476 3787.035
##       CAIC
## 1 3876.665
## 2 3874.895
##
## $LRtest
##   Model1 Model2     Chi2 df            p
## 1   mod1  multi 15.28774  2 0.0004789713
##
## attr(,"class")
## [1] "IRT.compareModels"
```

```r
summary(compare)
```

```
## Absolute and relative model fit
##
##   Model   loglike Deviance Npars Nobs      AIC      BIC     AIC3     AICc
## 1  mod1 -1880.882 3761.763    17  317 3795.763 3859.665 3812.763 3797.810
## 2 multi -1873.238 3746.476    19  317 3784.476 3855.895 3803.476 3787.035
##       CAIC
## 1 3876.665
## 2 3874.895
##
## Likelihood ratio tests - model comparison
##
##   Model1 Model2    Chi2 df     p
## 1   mod1  multi 15.2877  2 5e-04
```

We see that model `multi` fits slightly better. However, the log likelihood difference test shows the difference is statististically significant.

```
##   Model1 Model2     Chi2 df            p
## 1   mod1  multi 15.28774  2 0.0004789713
```

```r
compare$LRtest
```

Exercises

1. what evidence points towards multidimensionality?
2. compare the multidimensional model to the PCM model