There are **loads** of other analyses I could have done but considering that this is a portfolio, I have kept things as simple as possible for now.

I used the following code to recode my qualitative responses into quantitative numbers.

```
data(package = "ltm")
library(tidyverse)
df<- Environment
x<-df %>% mutate_all(~ str_replace(., "^$", NA_character_)) %>% mutate_all(.funs = ~
as.integer(recode(.x = .,"not very concerned"=0,"slightly concerned"=1, "very concerned"=2)))
library("writexl" write_xlsx(df,"D:\\df.xlsx")
```

Checking assumptions

The unidimensionality assumption was checked through factor analysis and parallel analysis.

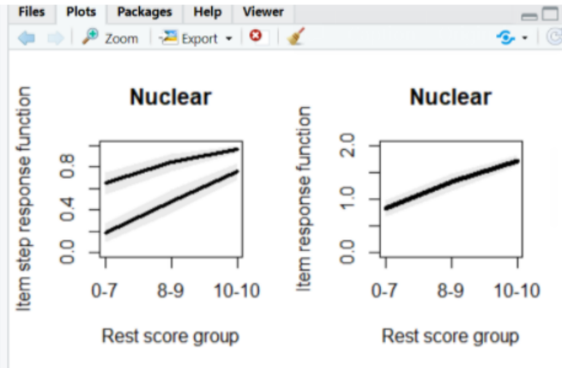
The local dependence assumption was checked through the following test. In practical terms, a correlation of r=0.40 is low dependency. The two items only have 0.4*0.4=0.16 of their variance in common. Correlations need to be around 0.7 before we are really concerned about dependency.

```
            LeadPetrol RiverSea RadioWaste AirPollution
LeadPetrol          NA    0.148     -0.082        0.106
RiverSea        12.666       NA     -0.123        0.127
RadioWaste       3.867    8.851         NA       -0.080
AirPollution     6.515    9.418      3.707           NA
Chemicals        2.746    4.760      8.896        4.335
Nuclear          2.319    5.092     10.314        4.877
            Chemicals Nuclear
LeadPetrol     -0.069  -0.063
RiverSea       -0.090  -0.094
RadioWaste      0.124   0.133
AirPollution   -0.086  -0.092
Chemicals          NA   0.074
Nuclear         3.175      NA
> 
```

The monotonicity assumption can be checked through the mokken package

As the ability level increases, the probability of getting the item correct increases monotonicially.

```
error in evaluating the argument 'x' in selecting a method for f
unction 'plot': object 'monotonicity.list' not found
> summary(monotonicity.list)
Error in h(simpleError(msg, call)) :
  error in evaluating the argument 'object' in selecting a method
 for function 'summary': object 'monotonicity.list' not found
>
> monotonicity.list <- check.monotonicity(E)
> plot(monotonicity.list)
Hit <Return> to see next plot: summary(monotonicity.list)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
>
>
>
```



I have fit two models(1d and 2d) and compared them using Anova

```
Iteration: 404, Log-Lik: -1069.019, Max-Change: 0.00010
> anova(zar1d, zar2d)

Model 1: mirt(data = itzareki, model = 1, itemtype = "graded")
Model 2: mirt(data = itzareki, model = 2, itemtype = "graded")

       AIC     AICc    SABIC       HQ      BIC    logLik     X2
1 2217.035 2219.550 2226.073 2243.523 2283.155 -1090.518    NaN
2 2184.037 2188.172 2195.586 2217.883 2268.524 -1069.019 42.998
   df   p
1 NaN NaN
2   5   0
> |
```

P value is low so we can say that 2 is better than 1. We can also see that AIC is lower in second model

```
> M2(x)
Error: M2() statistic cannot be calculated due to too few degrees
 of freedom
>
> x <- mirt(f, 1, itemtype = 'graded')
Iteration: 23, Log-Lik: -1090.518, Max-Change: 0.00004
>
> M2(x)
            M2 df           p     RMSEA    RMSEA_5  RMSEA_95
stats 12.84331  3 0.004988012 0.106368 0.05117397 0.1689135
           SRMSR       TLI       CFI
stats 0.07354776 0.9155455 0.9718485
> |
```

I ran an M2 test on 1D data as it would not run on 2D data due to less df. This portfolio is just for conceptual purpose so i decided to use 1D model instead of 2D.

Item fit statistics

After evaluating the test-level goodness of fit of the model, we can proceed to an item-level fit analysis. If the fit of the overall model is poor, then an item fit analysis might help to uncover the sources of misfit. Even if the overall fit seems adequate, an item fit analysis should be carried out, as it may improve the model with regard to interpretation, usefulness, and other important psychometric qualities. Item fit analysis involves comparing the model predictions with the actual response patterns. In UIRT, item fit can be evaluated by computing the S-X2 statistic for each item. Essentially, this item fit statistic compares the observed and expected proportions of correct and incorrect responses for each total score k in the sample. If the observed and expected proportions are similar, then we conclude that the item fits well. A smaller p value indicates a lack of fit which is the Air pollution item. We would not decide to delete the item in one go as p values can occur due to random variation. If we get same results in subsequent analysis i might consider revising the item

```
>
> itemfit(x, fit_stats = 'S_X2')
          item   S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1   LeadPetrol 14.671      10      0.040  0.145
2     RiverSea  3.099       3      0.011  0.377
3   RadioWaste  4.815       6      0.000  0.568
4 AirPollution  7.566       2      0.098  0.023
5    Chemicals  8.113       6      0.035  0.230
6      Nuclear  6.515       9      0.000  0.687
> 
```

```
> coef(x)
$LeadPetrol
        a1    a2    d1    d2
par -1.193 1.273 3.903 0.736

$RiverSea
        a1    a2    d1    d2
par -2.554 2.427 7.755 3.312

$RadioWaste
        a1    a2    d1    d2
par -4.645 0.097 7.627 3.297

$AirPollution
        a1   a2   d1    d2
par -3.489 2.47 8.56 1.826

$Chemicals
```

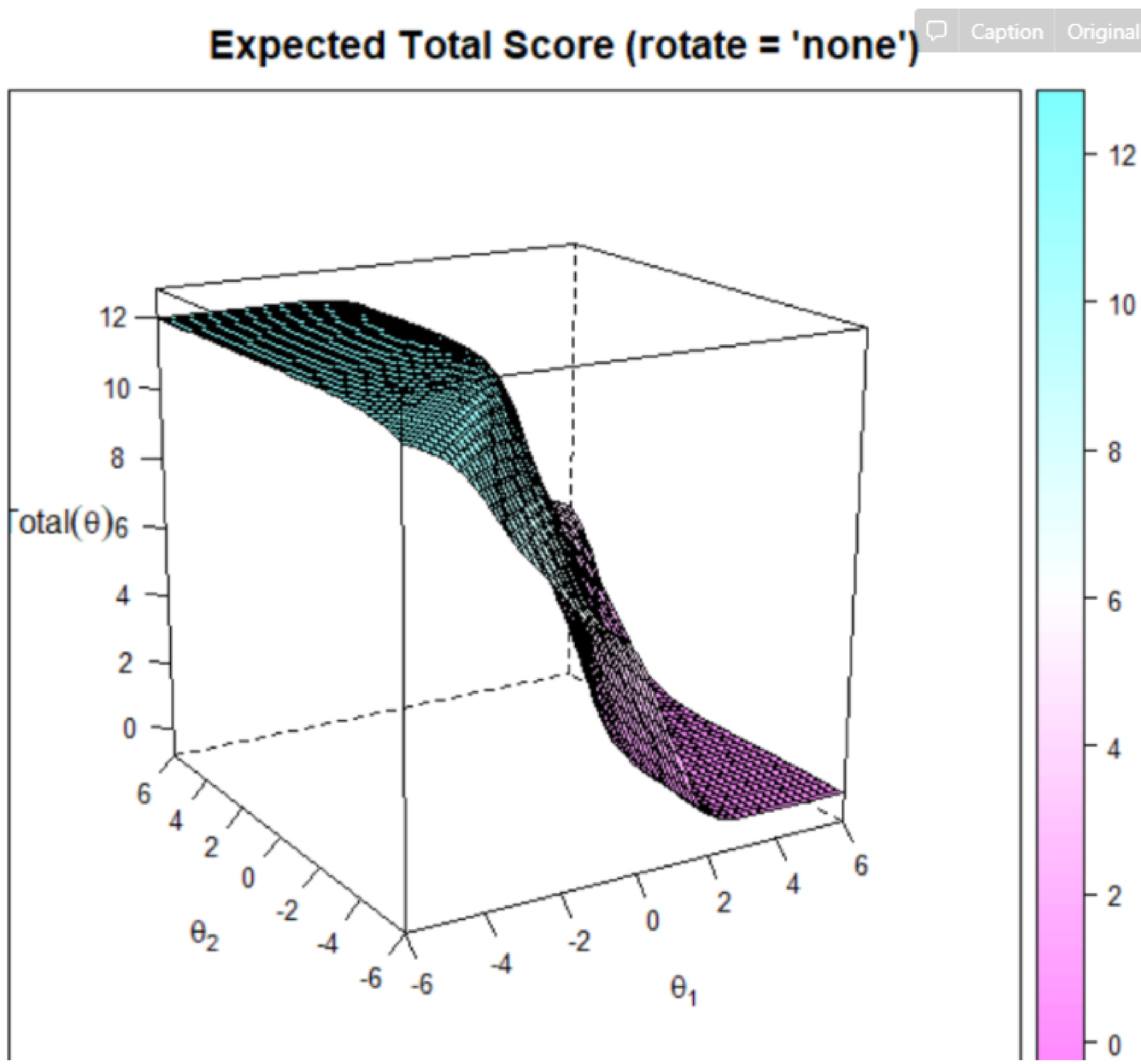
The multidimensional item location.

```
> head(MDIFF(x))
                MDIFF_1     MDIFF_2
LeadPetrol    -2.238020 -0.42225602
RiverSea      -2.201538 -0.94010701
RadioWaste    -1.641747 -0.70973219
AirPollution  -2.002388 -0.42719159
Chemicals     -1.799065 -0.77580373
Nuclear       -1.300202 -0.05499871
>
```

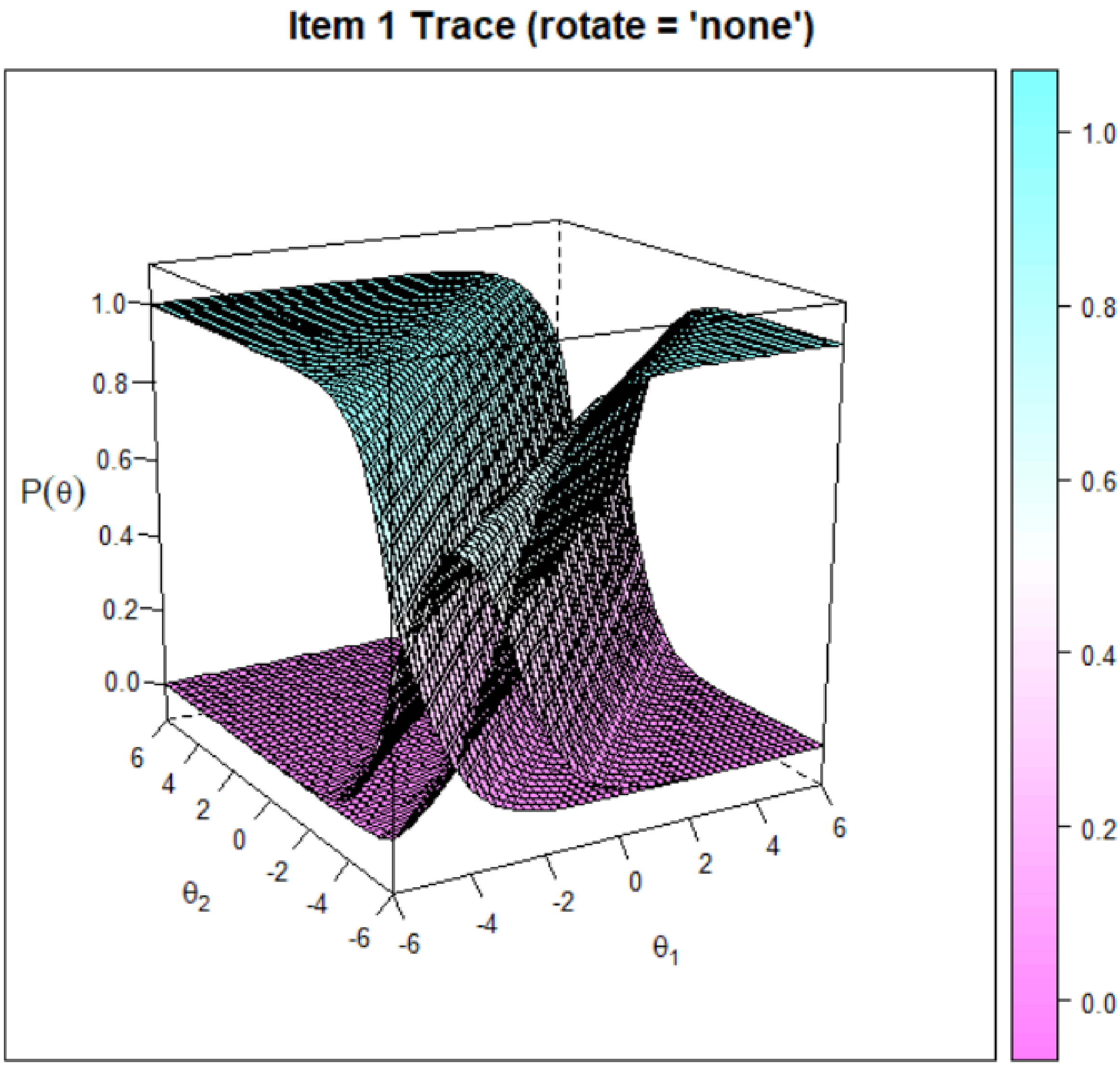
Finally, we can compute the person parameters, one for each dimension (only first six persons shown here):

```
Nuclear       -1.300202 -0.0549
> head(fscores(x))
             F1        F2
[1,] -0.7991555 0.7777467
[2,] -0.7991555 0.7777467
[3,] -0.7991555 0.7777467
[4,] -0.7991555 0.7777467
[5,] -0.7991555 0.7777467
[6,] -0.7991555 0.7777467
>
```

Plots

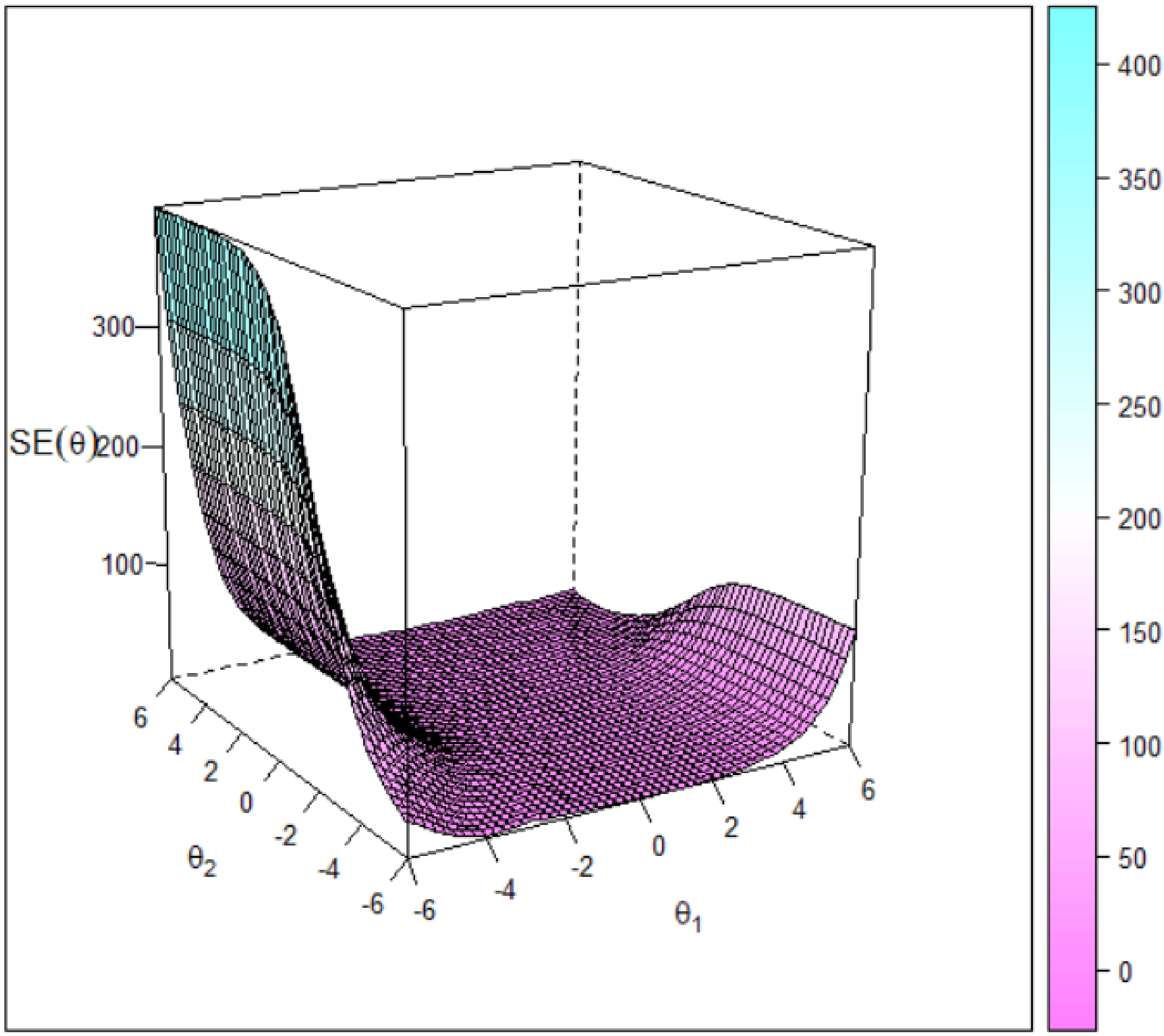


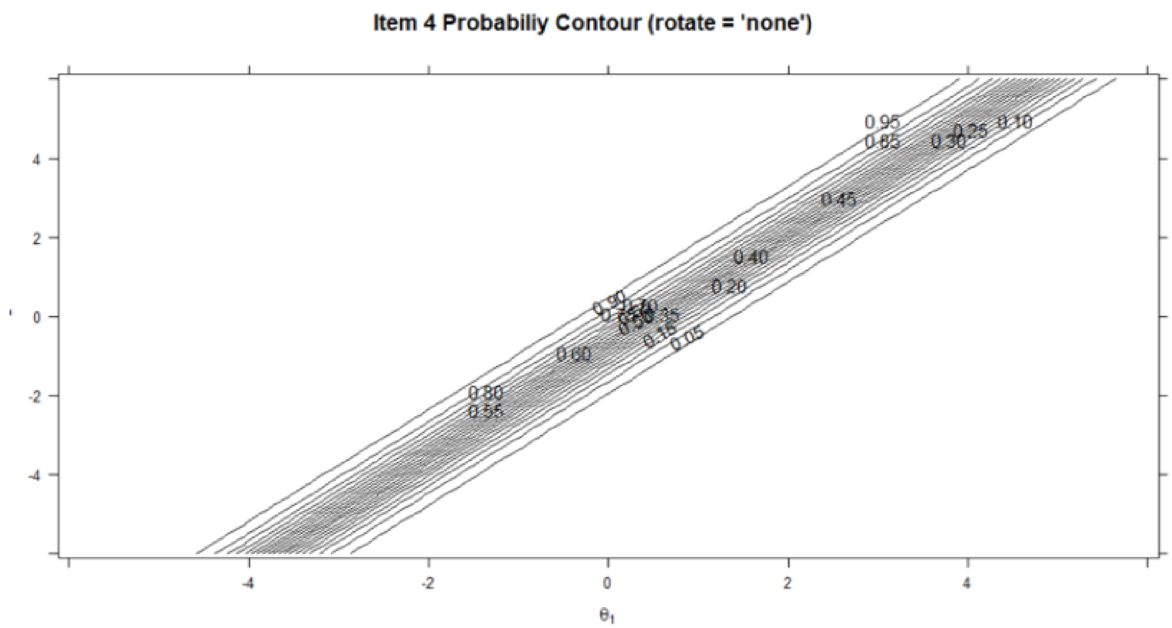
Caption
Original
Expected Total Score (rotate = 'none')
Total(θ)
θ1
θ2
12
10
8
6
4
2
0
-6
-4
-2
0
2
4
6

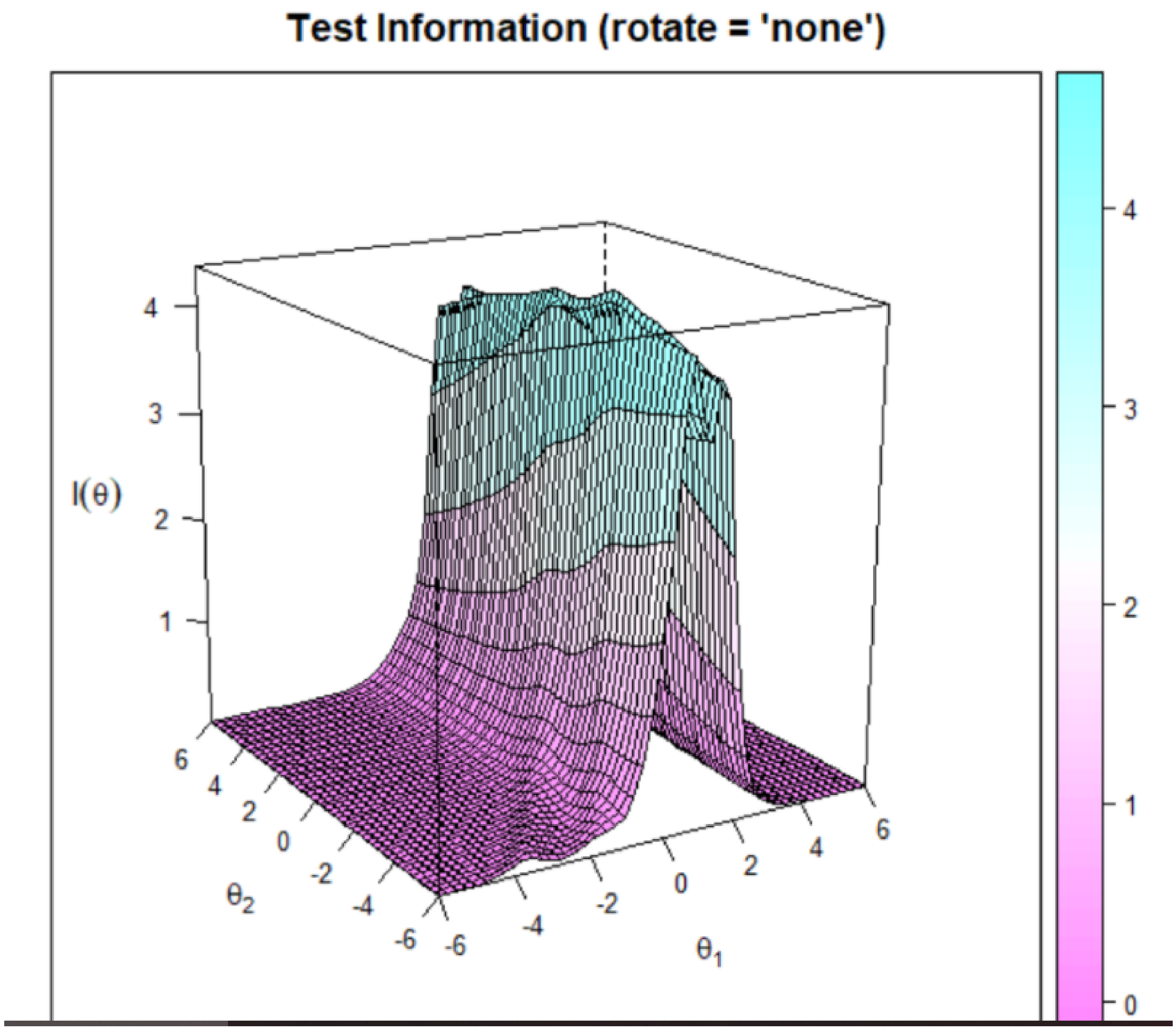


Item 1 Trace (rotate = 'none')
P(θ)
1.0
0.8
0.6
0.4
0.2
0.0
6
4
2
0
-2
-4
-6
θ2
θ1

Test Standard Errors (rotate = 'none')






Test Information (rotate = 'none')
I(θ)
θ2
θ1