SUPPLEMENTARY MATERIALS

Allostatic load scoring using item response theory

Shelley H. Liu[1], Robert-Paul Juster[2], Kristen Dams-O'Connor[3], & Julie Spicer[4]

[1]Department of Population Health Science and Policy, Icahn School of Medicine at Mount Sinai
[2]Department of Psychiatry and Addiction, University of Montreal
[3]Department of Rehabilitation Medicine, Department of Neurology, Icahn School of Medicine at Mount Sinai
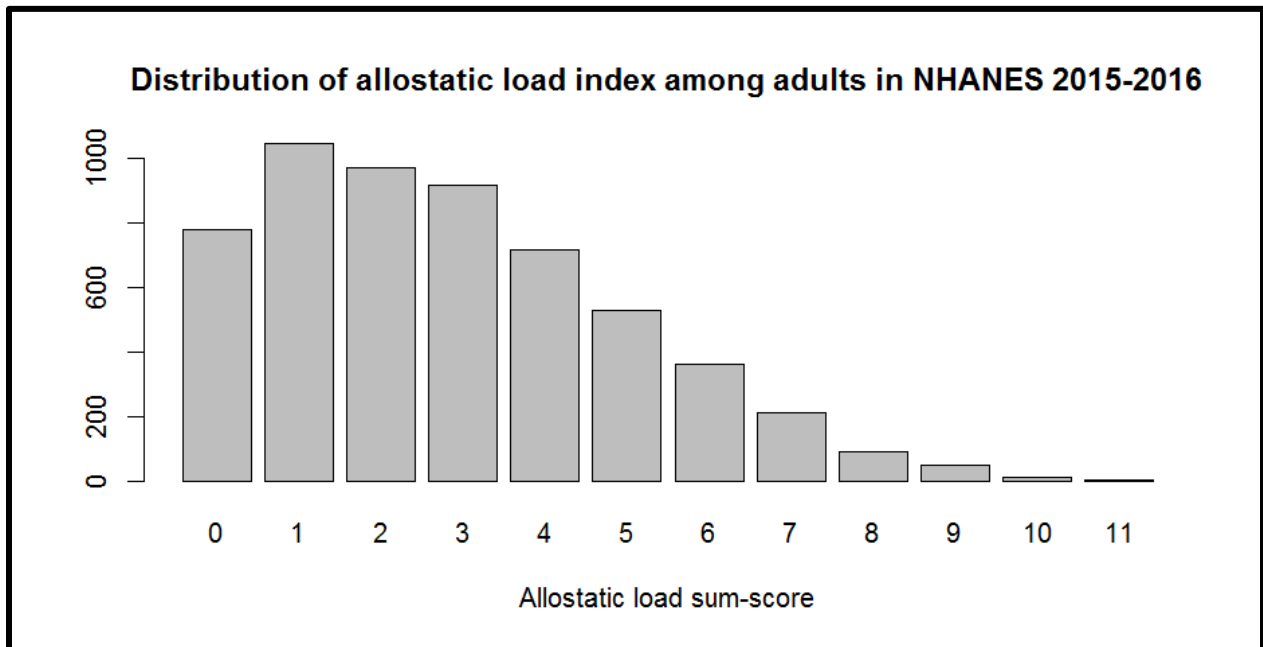[4]Department of Psychiatry, Icahn School of Medicine at Mount Sinai
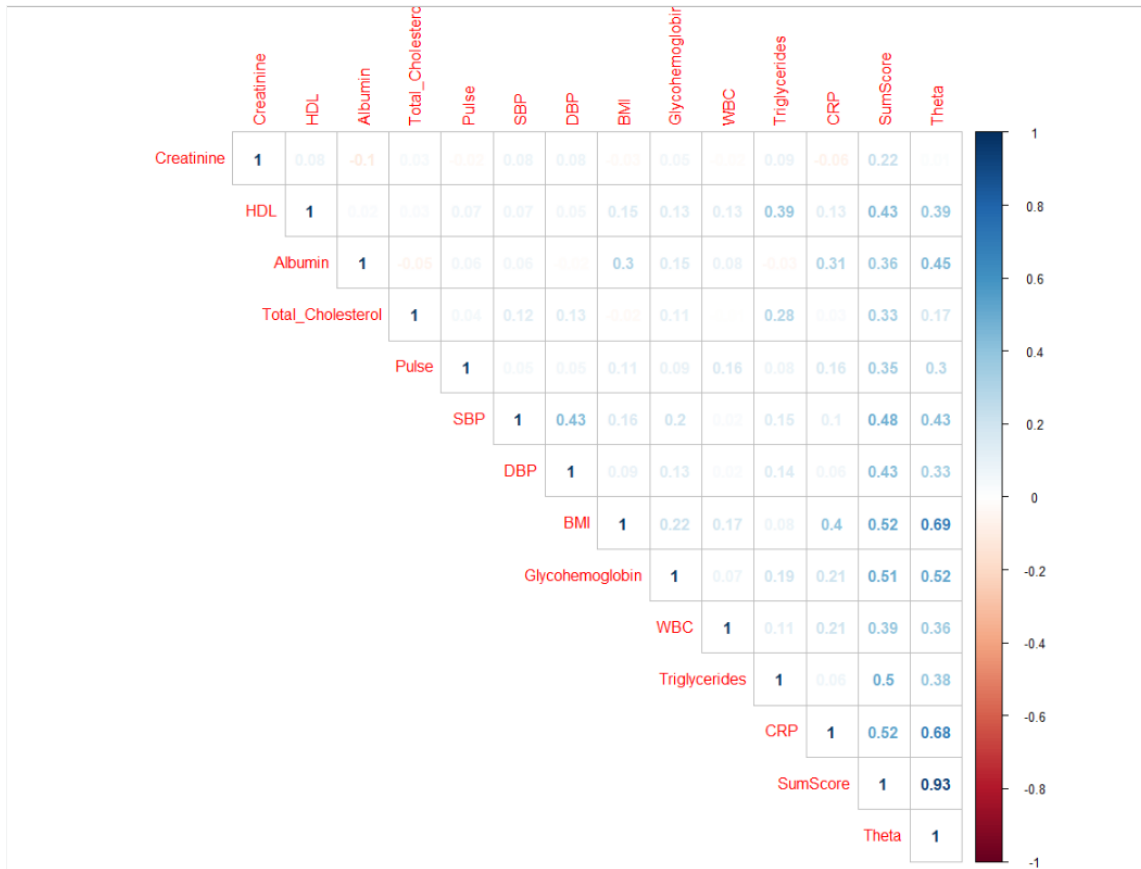
Corresponding author:
Shelley H. Liu
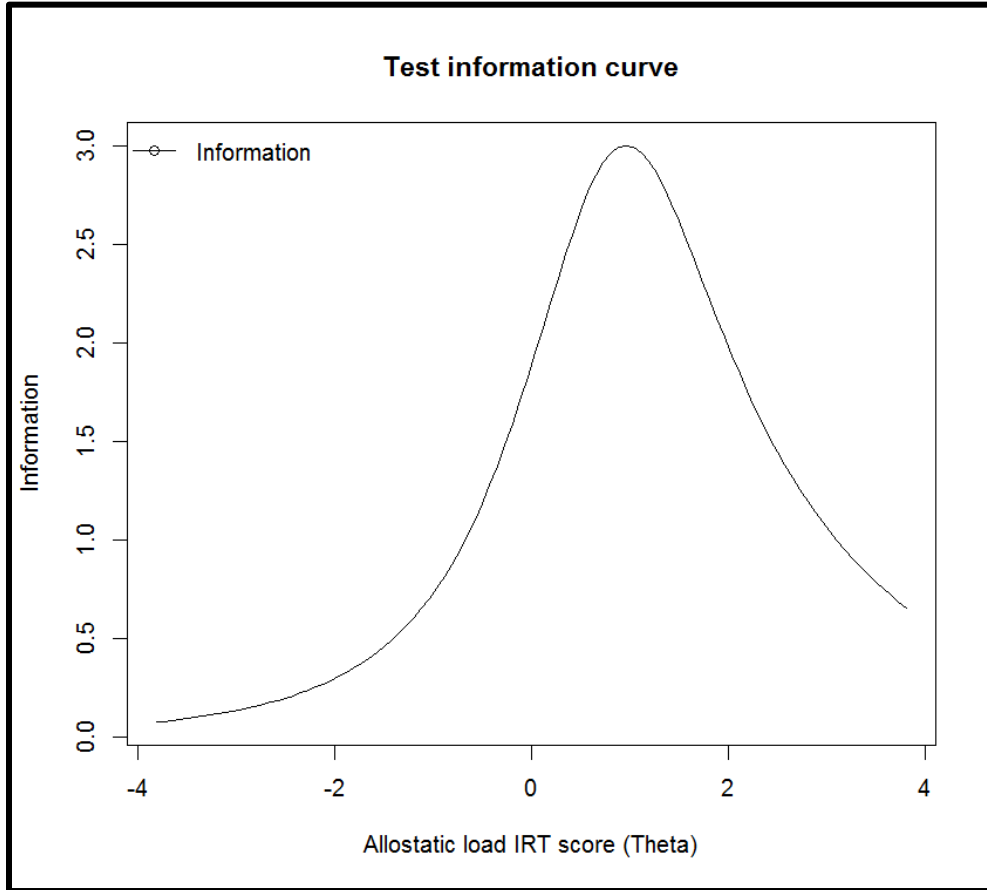Email: shelley.liu@mountsinai.org

I. Supplementary Figure 1. Distribution of the allostatic load sum-score in adults aged 20-59 in NHANES 2015-2016.

II. Supplementary Figure 2. Correlation between dichotomized (high/low risk) allostatic load biomarkers and the allostatic load sum-score and IRT score.

III. Supplementary Figure 3. Test information curve. The scale for the IRT score in the dataset remains between approximately -1 to 2. However, the test information curve provides a larger range, as it shows the information supplied by the model at different theta scores (but in our dataset, we only have theta scores ranging from -1 to 2).

IV. Supplementary Figure 4. Distribution of the allostatic load IRT score in adults aged 20-59 in NHANES 2015-2016.

Histogram of AL.mat$Theta
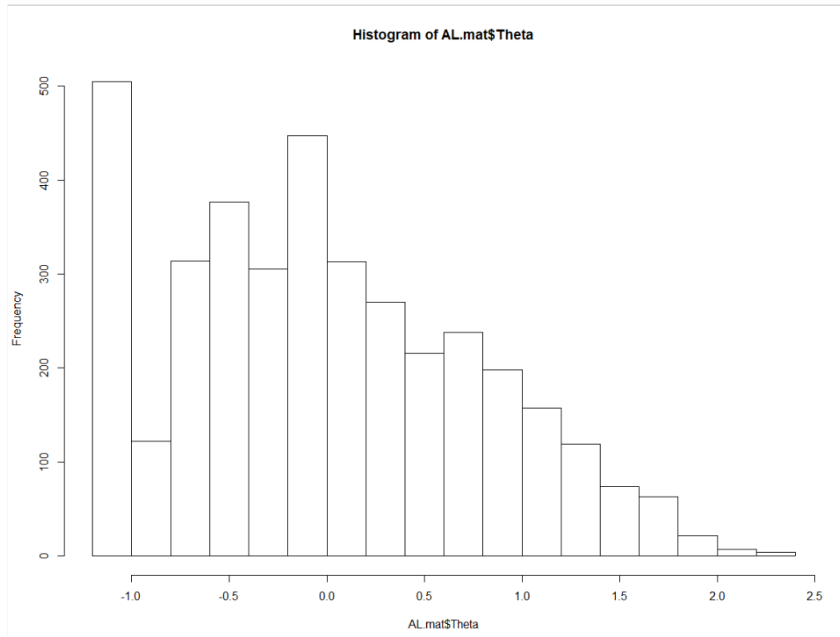
V. Supplementary Table 1. Associations of family income to poverty ratio and allostatic load.

Predictors	AL IRT score			AL Sum score		
	β	95% CI	p	β	95% CI	p
(Intercept)	0.1	0.05 – 0.15	**<0.001**	3.11	2.96 – 3.26	**<0.001**
Family income to poverty ratio	-0.04	-0.05 – -0.02	**<0.001**	-0.07	-0.12 – -0.02	**0.008**
Observations	3391			2835		

Socio-economic status was measured using family income to poverty ratio. The models have different sample sizes, as IRT does not require complete data on all AL biomarkers in order to calculate scores, unlike the sum-score approach.

VI. Simulated data and reproducible IRT code in R.

Here, we simulate data for n=500 participants. There are four correlated binary variables (e.g. four biomarkers, each classified as high/low risk).

```
# Load packages
library(MultiRNG)
library(lavaan)
library(ltm)

set.seed(123) # For reproducibility of these simulation results

# Simulate some data for n=500, with 4 correlated binary indicators
```

4

cmat<-matrix(c(1,0.4,0.3,0.25, 0.4,1,0.4,0.4, 0.3,0.4,1,0.2, 0.25,0.4,0.2,1), nrow=4, ncol=4)
propvec=c(0.4,0.4,0.3, 0.2)

mydata=draw.correlated.binary(no.row=500,d=4,prop.vec=propvec,corr.mat=cmat)
colnames(mydata) = c("V1", "V2", "V3", "V4")

head(mydata) # print first six roles of mydata

```
> head(mydata)
     V1 V2 V3 V4
[1,]  0  0  0  0
[2,]  0  0  0  0
[3,]  0  0  0  0
[4,]  0  0  0  0
[5,]  1  1  1  0
[6,]  0  1  0  1
```

Check for sufficient uni-dimensionality

CFA.model <- 'Construct =~ V1 + V2 + V3 + V4'
fit <- cfa(CFA.model, data=mydata)
summary(fit, fit.measures=TRUE)

```
> summary(fit, fit.measures=TRUE)
lavaan 0.6-5 ended normally after 24 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                          8

  Number of observations                           500

Model Test User Model:

  Test statistic                                14.862
  Degrees of freedom                                 2
  P-value (Chi-square)                           0.001

Model Test Baseline Model:

  Test statistic                               309.707
  Degrees of freedom                                 6
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.958
  Tucker-Lewis Index (TLI)                       0.873

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -1073.791
  Loglikelihood unrestricted model (H1)      -1066.360

  Akaike (AIC)                                2163.582
  Bayesian (BIC)                              2197.298
  Sample-size adjusted Bayesian (BIC)         2171.906
```

```
Root Mean Square Error of Approximation:

  RMSEA                                           0.113
  90 Percent confidence interval - lower          0.064
  90 Percent confidence interval - upper          0.170
  P-value RMSEA <= 0.05                           0.019

Standardized Root Mean Square Residual:

  SRMR                                            0.037

Parameter Estimates:

  Information                                  Expected
  Information saturated (h1) model           Structured
  Standard errors                              Standard

Latent Variables:
                  Estimate  Std.Err  z-value  P(>|z|)
  Construct =~
    V1              1.000
    V2              1.418    0.165    8.575    0.000
    V3              0.873    0.109    8.015    0.000
    V4              0.686    0.087    7.876    0.000

Variances:
                  Estimate  Std.Err  z-value  P(>|z|)
   .V1             0.165    0.013   12.771    0.000
   .V2             0.093    0.015    6.159    0.000
   .V3             0.151    0.011   13.359    0.000
   .V4             0.101    0.007   13.555    0.000
    Construct      0.072    0.013    5.435    0.000
```
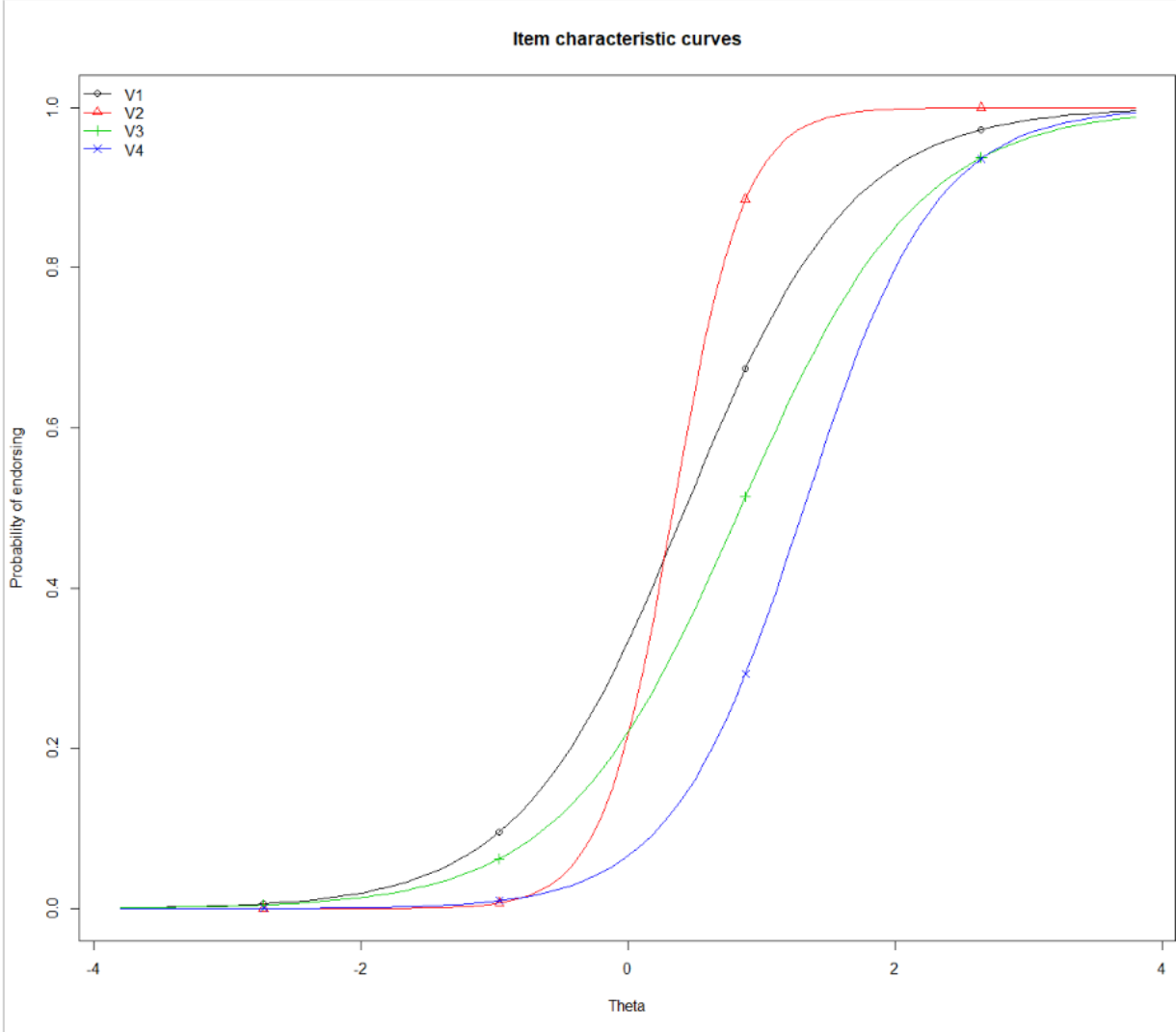
Run IRT

irt.dat = ltm(mydata ~ z1, IRT.param = TRUE)
coef(irt.dat)

```
> coef(irt.dat)
      Dffclt    Dscrmn
V1 0.4328137 1.611859
V2 0.3429488 3.787745
V3 0.8456207 1.501197
V4 1.3186840 2.019666
```
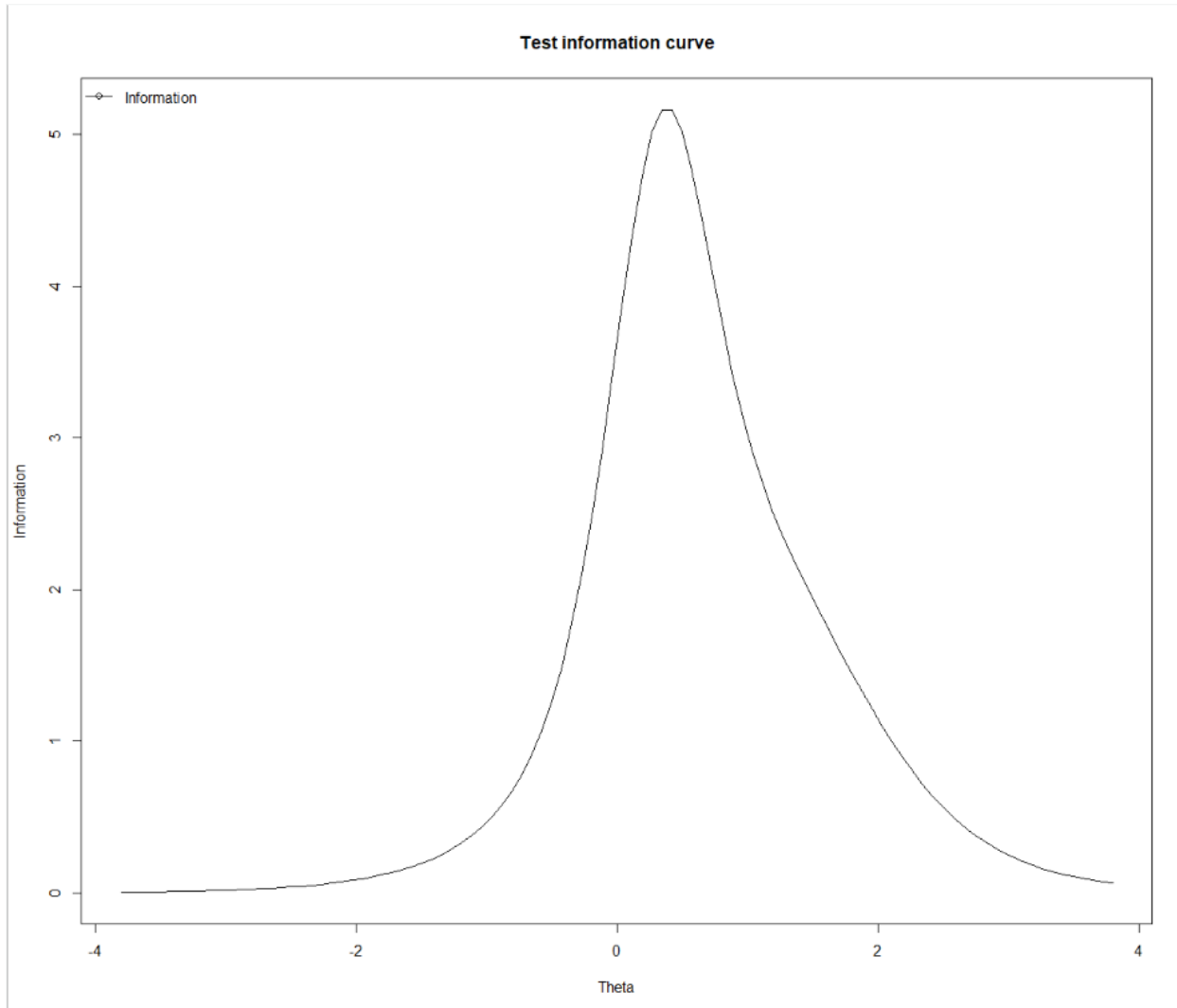
plot(irt.dat, pch = c(1:4), lty = 1, type = "ICC", legend = TRUE, xlab = "Theta", main = "Item characteristic curves", ylab ="Probability of endorsing")

plot(irt.dat, pch = c(1:4), item = 1:4, lty = 1, type = "IIC", legend = TRUE, xlab = "Theta", main = "Item information curves", ylab ="Information")

Item information curves
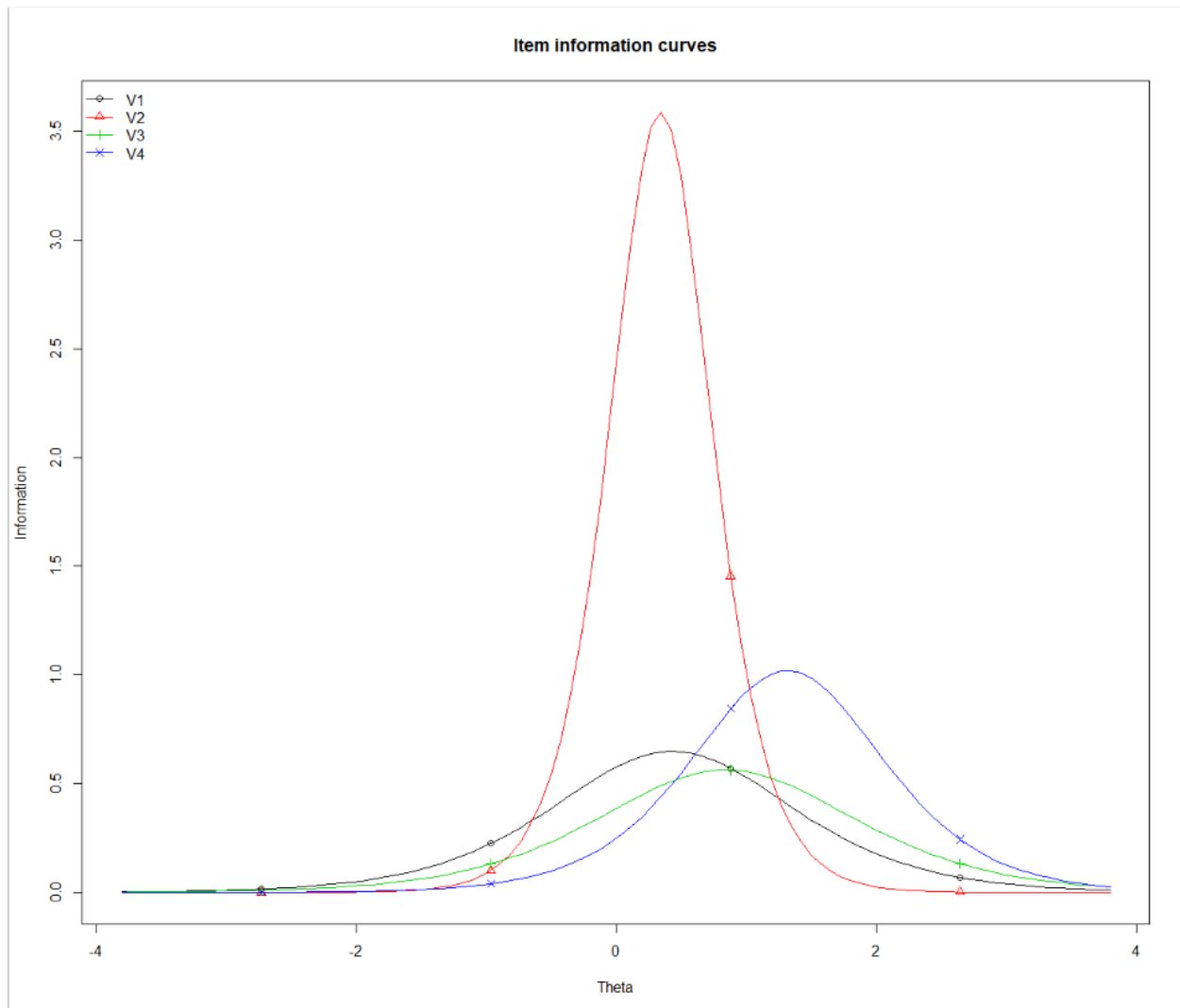
plot(irt.dat, pch = c(1:4), item = 0, type = "IIC", legend = TRUE, xlab = "Theta", main = "Test information curve", ylab ="Information")

```
Theta = factor.scores(irt.dat, method = "EAP", resp.patterns = mydata)$score.dat$z1
sum.score = apply(mydata, 1, sum)
plot(sum.score ~ Theta, xlab = "Theta", ylab = "Sum score")
```
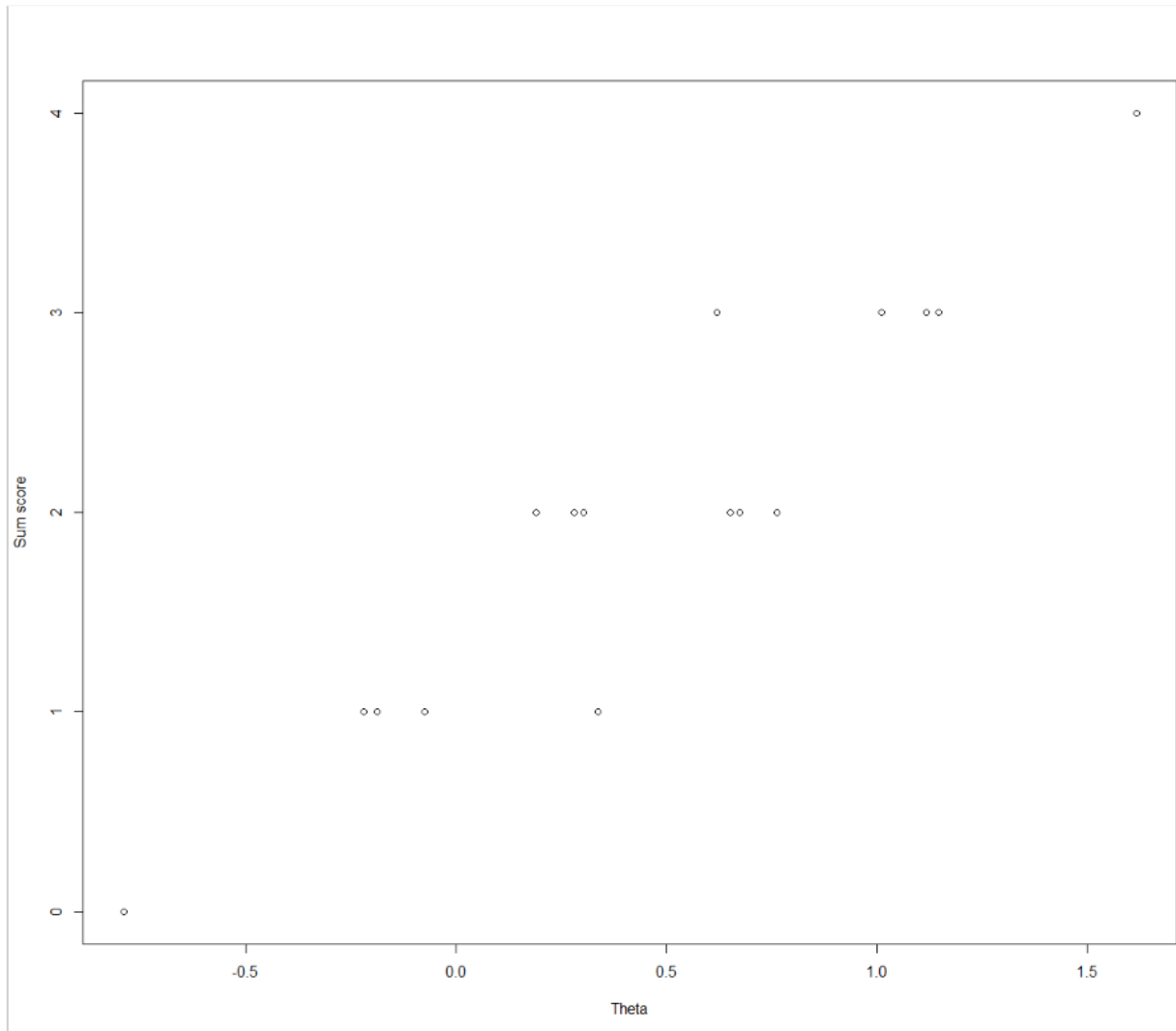


```
mydata2 = cbind(mydata, Theta, sum.score)
cor(sum.score, Theta) # Correlation = 0.98
```