Arcara 2024 - Regression Methodfor normative data - Tutorial

2025-07-16

Introduction

This tutorials explain how to use the code for applying the method in *Arcara G.(2024) Improving Equivalent scores: a new method for regression model selection.*

Set up R

To work with the method you need to load some functions that will be used. Here I included all the required functions in a folder called `R functions` (these will be wrapped in an R package in the future).

```
source("R_functions/adjscores_A2024_v3.R")
source("R_functions/formula_transf_text.R")
source("R_functions/model_transf_text.R")
source("R_functions/transf_functions.R")
source("R_functions/tolLimits.adjscores.R")
source("R_functions/tolLimits.obs.R")
source("R_functions/ES.R")
```

I then load some required packages.

```
require(effects)
require(car)
require(performance)
```

Import data

In the following lines I import the data and make some check everything is ok

```
Test.dat = read.csv("Original_Data/MOCA_Dataset.csv", sep=",", dec=".")

dim(Test.dat)
```

```
## [1] 440   5
```

```
head(Test.dat)
```

```
##    ID Age Education Sex Score
## 1 482  58      27.0   M    25
## 2 428  70      26.0   F    24
## 3 236  59      23.0   F    26
## 4 217  58      23.0   F    28
## 5  34  89      22.0   M    23
## 6 414  51      21.5   F    26
```

```
str(Test.dat)
```

```
## 'data.frame':    440 obs. of  5 variables:
##  $ ID       : int  482 428 236 217 34 414 116 336 104 85 ...
##  $ Age      : int  58 70 59 58 89 51 65 54 61 63 ...
```

```
##  $ Education: num  27 26 23 23 22 21.5 21 21 20 20 ...
##  $ Sex      : chr  "M" "F" "F" "F" ...
##  $ Score    : int  25 24 26 28 23 26 24 27 26 26 ...
```

```r
# fix values for participants with zero Education otherwise some transformations (e.g. 1/x, log) could
Test.dat[Test.dat$Education==0, "Education"] = 1

Test.dat = na.omit(Test.dat)
```

Use the method to select regression model

```r
Test.ARC.res = adjscores_A2024(df = Test.dat, dep="Score",
                              age="Age", edu="Education", sex="Sex",
                              dep.range = c(0, 100))
```

```
## Sex converted to numeric
```

Inspect model results. The results is a list of objects (see `adjscores_A2024.R` file for details):

- the first is a dataset in which a new column with adjusted scores is added

```r
head(Test.ARC.res$new.df)
```

```
##    ID Age Education Sex Score dep age  edu sex sex.or age_tr     edu_tr
## 1 482  58      27.0   M    25  25  58 27.0   0      M   3364 0.03703704
## 2 428  70      26.0   F    24  24  70 26.0   1      F   4900 0.03846154
## 3 236  59      23.0   F    26  26  59 23.0   1      F   3481 0.04347826
## 4 217  58      23.0   F    28  28  58 23.0   1      F   3364 0.04347826
## 5  34  89      22.0   M    23  23  89 22.0   0      M   7921 0.04545455
## 6 414  51      21.5   F    26  26  51 21.5   1      F   2601 0.04651163
##   ADJ_SCORES RESIDUALS
## 1   23.41360 -2.466498
## 2   23.31524 -2.564851
## 3   24.59679 -1.283304
## 4   26.53007  0.649978
## 5   24.16439 -1.715703
## 6   24.14983 -1.730263
```

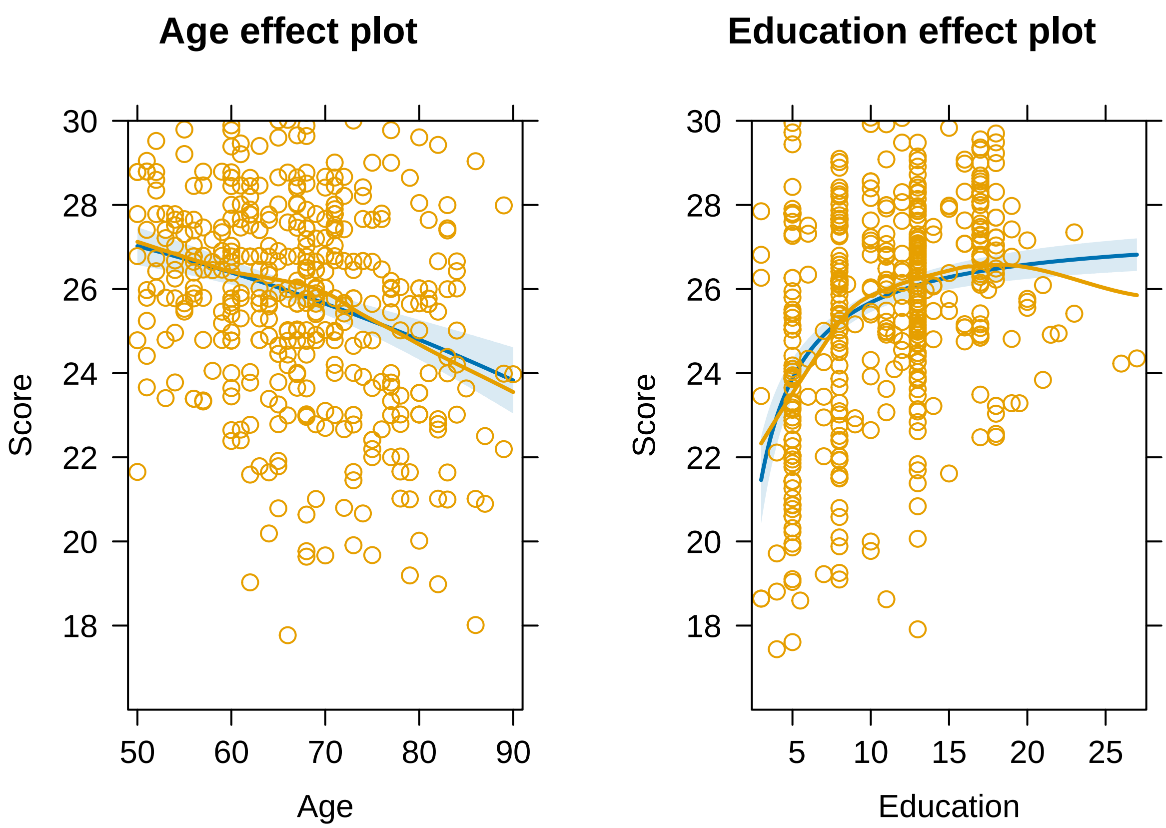
- the second is the linear model estimated (here I use `summary` to better inspect the model) and I plot the partial effects

```r
summary(Test.ARC.res$lm.model)
```

```
##
## Call:
## lm(formula = Score ~ quadr(Age) + inv(Education), data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1796 -1.4153  0.2001  1.5930  6.3890
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.005e+01  4.407e-01  68.201  < 2e-16 ***
## quadr(Age)     -5.702e-04  1.027e-04  -5.553 4.89e-08 ***
## inv(Education) -1.808e+01  2.272e+00  -7.959 1.50e-14 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.401 on 437 degrees of freedom
## Multiple R-squared:  0.2729, Adjusted R-squared:  0.2696
## F-statistic: 82.02 on 2 and 437 DF,  p-value: < 2.2e-16
```

```
plot(allEffects(Test.ARC.res[[2]], partial.residuals=TRUE))
```



- I can also print the final formula

```
print(Test.ARC.res$model_text)
```

```
## [1] "30.1 + quadr(Age) * -0.00057 + inv(Education) * -18.1"
```

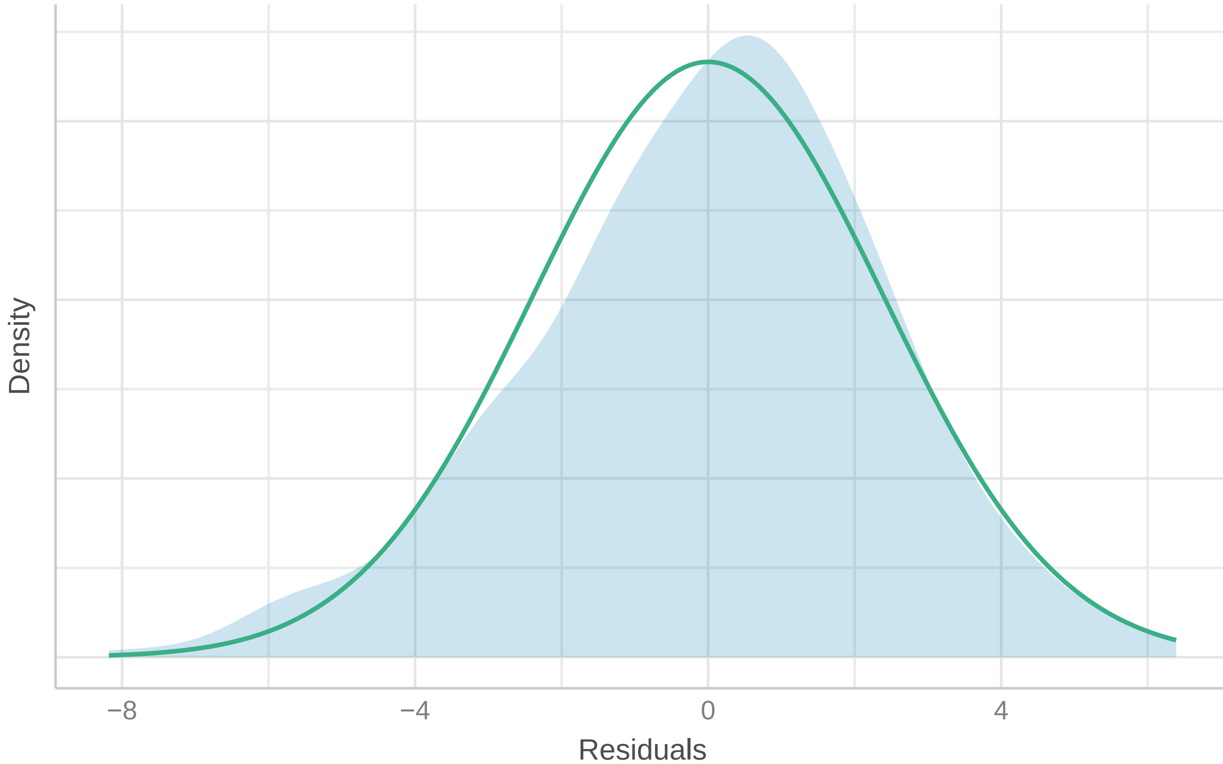
Model diagnositc

An appropriate model selection should also include some diagnostics

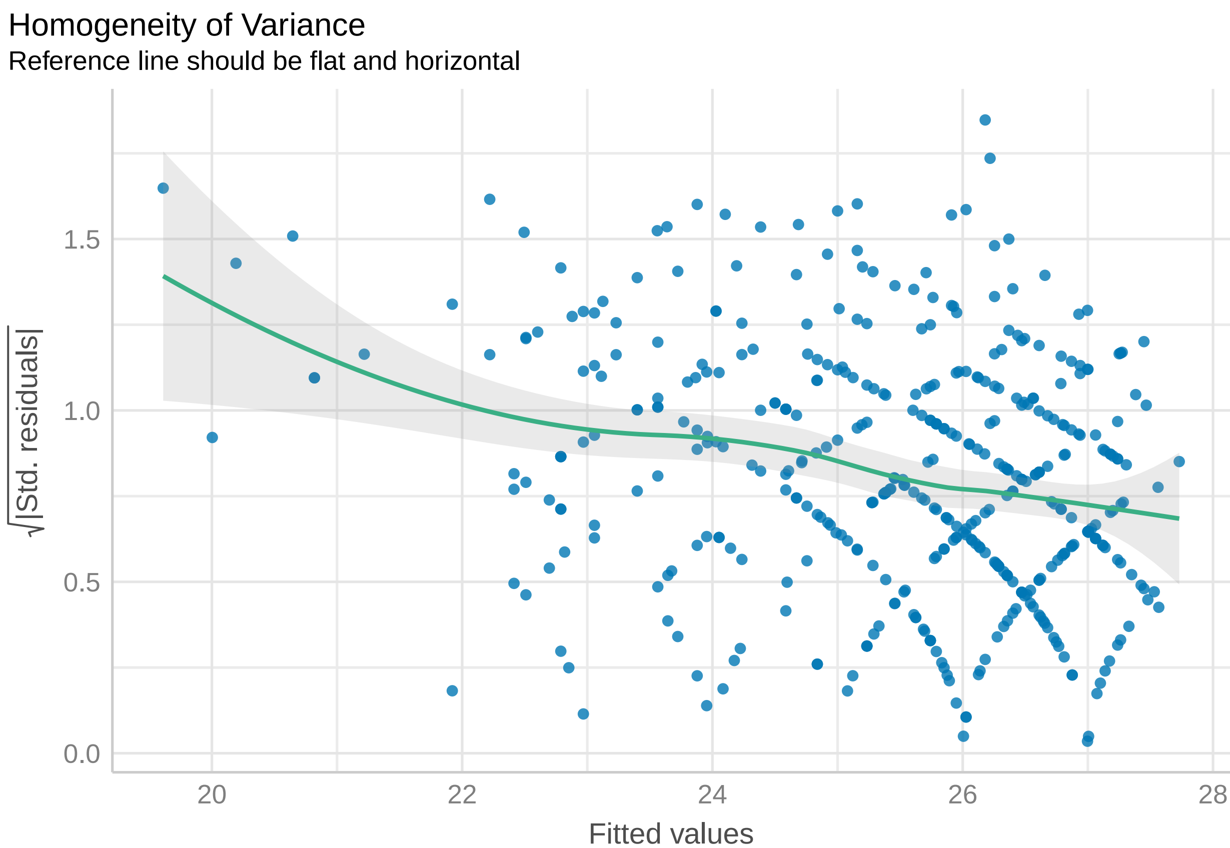
```
plot(check_normality(Test.ARC.res$lm.model))
```

Normality of Residuals

Distribution should be close to the normal curve




```
plot(check_heteroscedasticity(Test.ARC.res$lm.model))
```




```
# some disomogeneity in variance, but overall a good fit
```

Calculate Equivalent Scores

To calculate thresholds for Equivalent Scores I can use some some few additional code.

```
Test.ES = ES(adjscores=Test.ARC.res$new.df$ADJ_SCORES)

print(Test.ES)
```

```
## $Observations
## ES0(oTL)-ES1      ES1-ES2      ES2-ES3      ES3-ES4
##           15           49          119          219
##
## $Adjusted_Scores
## ES0(oTL)-ES1      ES1-ES2      ES2-ES3      ES3-ES4
##     21.09169     22.87938     24.51027     26.04739
```

Citing the Method

If you use the regression method script please cite: *Arcara G. (2024) Improving Equivalent Scores: A new method for regression model selection*

If you use the ES, please cite: *Aiello, E. N., & Depaoli, E. G. (2022). Norms and standardizations in neuropsychology via equivalent scores: software solutions and practical guides. Neurological Sciences, 43(2), 961-966.*