DRAFT: Analysis of the reporting styles using generalized ordered threshold models.

Maciej J. Dańko

16 I 2019

Abstract

The *hopit* package provides R functions to fit and analyze ordered response data in the context of reporting styles. In this vignette we describe the formulation and fit of *hopit* models as well as functions used to analyse reporting styles.

1. Introduction

The *hopit* package provides R functions to fit and analyze ordered response data in the context of reporting styles.

The ordered response data classifies a measure of interest into ordered categories collected during a survey. If the dependent variable is a happiness then a respondent typically answers a question: "Taking all things together, would you say you are . . . ? " and have some response options e.g."very happy","pretty happy","not too happy","very unhappy" (Liao, Fu, and Yi 2005). Similarly if interviewees are asked to evaluate their health in general (e.g. "Would you say your health is . . . ?") they may choose among several categories, e.g. very good, good, fair, bad, and very bad (Jürges 2007). In political sciences a respondent may be asked for an opinion about recent legislation (e.g. "Rate your feelings about the proposed legislation") and asked to choose among several categories "strongly oppose", "mildly oppose", "indifferent", "mildly support", "strongly support" (Greene and Hensher 2010). It is easy to imagine other multi-level ordinal variables that might by use during the survey and to which methodology described below could be applied with.

Practically, it is assumed that when responding to a survey question about their general happiness, health, feeling, attitude or other status, participants assess their true value of this unobserved continuous variable, and project it to a provided discrete scale. The thresholds that each individual uses to categorize their true status into a specific response option may be affected by the choice of a reference group, earlier life experiences, and cross-cultural differences in using scales, and thus, may differ across individuals depending on their gender, age, cultural background, education, and personality traits, among other factors.

From the modeling perspective, one of the main tasks is to compute this continuous estimate of individuals' underlying, latent variable based on several specific characteristics of the considered response (e.g. health variables or happiness variables) and accounting also for variations in reporting across socio-demographic and cultural groups. More specifically, to build the latent, underlying variable a generalized hierarchical ordered threshold model is fitted, which regresses the reported status/attitude/feeling on two sets of independent variables (Boes and Winkelmann 2006; Greene et al. 2014). When a dependent reported ordered variable is self-rated health status then the first set of variables - health variables - assesses individuals' specific aspects of health, and might include chronic conditions, mobility level, difficulties with a range of daily activities, performance on grip strength test, anthropometric measures, lifestyle behaviors, etc. Using the second set of independent variables (threshold variables), the model also adjusts for the differences across socio-demographic and cultural groups like cultural background, gender, age, education, etc. (King et al. 2004; Jürges 2007; but see Rebelo and Pereira 2014).

Once the model is fitted its estimates (latent variable and threshold coefficients) can be used to calculate the differences in reporting styles among groups of people having different contextual characteristics realized by calculation of differences between expected and reported ordinal response measures (Jürges 2007).

2. Generalized (hierarchical) ordered threshold model

Ordered threshold models are used with ordered categorical dependent variables. The generalized ordered threshold models (Ierza 1985; Boes and Winkelmann 2006; Greene et al. 2014) are an extension to the ordered threshold models (McKelvey and Zavoina 1975). In the latter models the thresholds are constant, whereas generalized models allows thresholds to be dependent on covariates. Greene and Hensher (2010) and Greene et al. (2014) pointed out that also thresholds must be ordered so that a model has a sense. This motivated Greene and coauthors to call this models **HOPIT**, which stands for hierarchical ordered probit models.

In the the self-rated health example, the response variable is self-rated health and latent variable h_i can depend on different health conditions and diseases (health variables X). Variables X are modeled with parallel regression assumption. According to the assumption, coefficients, which describe the relationship between lowest and all higher response categories, are the same as those coefficients, which describe the relationship between another (e.g. adjacent) lowest and the remaining higher response categories. In the considered case h_i is modeled as a linear function of X and their coefficients β:

$$h_i = \sum_{k=1}^{K} \beta_k X_{i,k} = X'\beta \tag{1}$$

where index $i \in 1...N$ is number of cases (e.g. respondents), X is in the form of model matrix, and K is number of columns in X. As described above, the categorization (response mechanism) of the latent variable h_i is modeled in terms of thresholds $\alpha_{i,j}$ assuming that thresholds of lower order are never greater than thresholds of higher orders:

$$\begin{cases} y_i = 1 \;\Leftrightarrow\; \alpha_{i,0} \le h_i < \alpha_{i,1} \\ y_i = 2 \;\Leftrightarrow\; \alpha_{i,1} \le h_i < \alpha_{i,2} \\ \dots \\ y_i = j \;\Leftrightarrow\; \alpha_{i,j-1} \le h_i < \alpha_{i,j} \\ \dots \\ y_i = J \;\Leftrightarrow\; \alpha_{i,J-1} \le h_i < \alpha_{i,J} \end{cases} \tag{2}$$

The thresholds (cut points, α) are modeled by threshold variables γ and intercepts λ. It is assumed that they model contextual characteristics of the respondent (e.g. country of origin, gender, age, etc.). Threshold variables are modeled without parallel regression assumption, thus each threshold is modeled by a variable independently (Boes and Winkelmann 2006; Greene et al. 2014).

Different parametrizations of thresholds exist (Greene et al. 2014; Rebelo and Pereira 2014; Jürges 2007). In the package, King et al. (2004) and Jürges (2007) parametrization is used, which assumes that:

$$\alpha_{i,\ j} = \begin{cases} -\infty & for\ j = 0 \\ \lambda_1 + \sum_{m=1}^{M} \gamma_{1,m} Y_{i,m} & for\ j = 1 \\ \alpha_{i,\ j-1} + exp(\lambda_j + \sum_{m=1}^{M} \gamma_{j,m} Y_{i,m}) & for\ J-1 \ge j \ge 2 \\ \infty & for\ j = J \end{cases} \tag{3}$$

The condition $y_i = j \;\Leftrightarrow\; \alpha_{j-1,i} \le h_i < \alpha_{j,i}$ can be easily expressed in terms of the probability, which leads to:

$$P(y_i = j) = P(\alpha_{j-1,i} \le h_i < \alpha_{j,i}), \tag{4}$$

hence

$$P(y_i = j) = \Phi(\alpha_{i,\ j} - h_i) - \Phi(\alpha_{i,\ j-1} - h_i), \tag{5}$$

where Φ is a distribution function (cdf, cumulative density function). For example, for probit regression it is standard normal cdf $\Phi(x) = \frac{1}{2} + \frac{1}{2} * erf\left(\frac{x}{\sqrt{2}}\right)$ whereas for logit regression it takes the form $\Phi(x) = \frac{1}{1+e^{-x}}$.

In reporting styles analyses the typical choice is the probit model. It simply assumes that h_i is affected by a random noise ϵ_i having standard normal distribution $\epsilon_i \sim \mathcal{N}(0,1)$.

Using all definitions presented above the log likelihood function can be constructed

$$\ln L = \sum_{i=1}^{N} \sum_{j=1}^{J} z_{i,\ j} \ln \Big[\Phi(\alpha_{i,\ j} - h_i) - \Phi(\alpha_{i,\ j-1} - h_i) \Big], \tag{6}$$

where $z_{i,j}$ is an indicator function defined as:

$$z_{i,\ j} = \begin{cases} 0 & for\ y_i = j \\ 1 & for\ y_i \neq j \end{cases} \tag{7}$$

3. Analysis of reporting styles

The model estimates are used to determine reporting behavior, i.e in how the continuous latent variable is projected onto categorical response measure. Practically, it is done by comparing actual categorical ordered responses with theoretical ones that are adjusted for heterogeneity in reporting behaviors and are more comparable across individuals.

One of the first steps of the analysis is standardization of the latent variable to obtain latent index H_i.

$$H_i = 1 - \frac{h_i - \min\limits_i h_i}{\max\limits_i h_i - \min\limits_i h_i} \tag{8}$$

In the self-rated health example H_i is a proxy for true underlying health of an individual, and varies from 0 representing the (model-based) worst health state to 1 representing the (model-based) best health in the sample.

The predicted latent variable h_i obtained from the model is also used to standardize latent variable coefficients. In the self-rated health example the standardized coefficients are called disability weights D_k (Jürges 2007) and are calculated for each health variable to provide information about the impact of a specific health measure on the latent index. The disability weight for a health variable is equal to the ratio of corresponding health coefficient and the difference between the lowest and highest values of predicted latent health.

$$D_k = \frac{\beta_k}{\max\limits_i h_i - \min\limits_i h_i} \tag{9}$$

While the latent index H_i is intend to reflect underlying health, happiness or other status across individuals, the standardized coefficients D_k, like disability weights, are computed for an average individual in the study population. The relation between H_i and D_k follows the equation:

$$H_i = C - \sum_{k=1}^{K} D_k X_{i,k}, \quad \text{where } C = \frac{\max\limits_i h_i}{\max\limits_i h_i - \min\limits_i h_i} \tag{10}$$

Reporting styles analysis is based on the reclassification of individuals into new response categories. The classification is based on calculated latent index H_i and is thus adjusted for inter-individual differences in reporting behavior. There are two methods of reclassification: (1) Jürges (2007) percentile method (see also Rebelo and Pereira 2014) and (2) reclassification based on estimated thresholds.

The Jürges' percentile method is based on on original distribution of categorical response variable. First for each category j an empirical distribution function is constructed.

$$\hat{F}(j) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{y_i} \leq j \tag{11}$$

Where **1** is indicator function taking 1 if the condition is true or 0 otherwise. The calculated cumulative frequencies of latent index H_i are used as percentiles (cut points), so each individual i can be reclassified into new response categories.

In the second case the reclassification is based on eq. (2), so each individual has its own, model-derived cut-points.

4. Installing and loading the package

The newest available version of the package is always available from GitHub. It can be installed using *devtools* package

```
library(devtools)
install_github("maciejdanko/hopit")
```

```
library(hopit)
```

In examples presented below we will use artificially generated data set.

```
data(healthsurvey)
head(healthsurvey)
```

```
##   ID    health diabetes obese IADL_problems hypertenssion high_cholesterol
## 1  1 Very good       no    no            no            no               no
## 2  2      Good       no    no            no           yes              yes
## 3  3      Good      yes    no            no            no               no
## 4  4      Good       no    no            no            no               no
## 5  5 Excellent       no    no            no            no               no
## 6  6      Good       no    no            no           yes              yes
##   respiratory_problems heart_atack_or_stroke poor_mobility very_poor_grip
## 1                   no                    no            no             no
## 2                   no                   yes            no             no
## 3                   no                    no            no             no
## 4                   no                    no            no             no
## 5                   no                    no            no             no
## 6                  yes                    no           yes             no
##   depression other_diseases   sex ageclass education contHM country
## 1         no            yes   man [80,120)     prim-    3.6       Y
## 2         no            yes   man  [70,80)     prim-    4.4       Y
## 3         no             no   man  [50,60)     prim-    4.5       X
## 4        yes             no   man  [60,70)      sec+    5.1       Y
## 5         no             no woman [80,120)     prim-    3.3       Z
## 6         no            yes   man [80,120)     prim-    5.1       Y
##       csw psu   ssu
## 1 2407.48  YB  <NA>
## 2 1198.12  YB  <NA>
## 3  885.26  XC XCgis
## 4  772.04  YA  <NA>
## 5 1304.24  ZB  <NA>
## 6  917.16  YD  <NA>
```

5. Fitting the model using the *hopit*() function

Generalized ordered probit model can be fitted using *hopit*() function. The function takes two kinds of formulas: (1) *latent.formula* that models the latent variable and (2) *thresh.formula that models thresholds.

```
model1<- hopit(latent.formula = health ~ hypertenssion + high_cholesterol +
                               heart_atack_or_stroke + poor_mobility + very_poor_grip +
                               depression + respiratory_problems +
                               IADL_problems + obese + diabetes + other_diseases,
               thresh.formula = ~ sex + ageclass,
               decreasing.levels = TRUE,
               control=list(trace=FALSE),
               data = healthsurvey)

summary(model1)
```

```
## Formula (latent variables):
## health ~ hypertenssion + high_cholesterol + heart_atack_or_stroke +
##     poor_mobility + very_poor_grip + depression + respiratory_problems +
##     IADL_problems + obese + diabetes + other_diseases
## Formula (threshold variables): ~sex + ageclass
## Link: probit
## Number of cases: 10000
## Response levels: Excellent, Very good, Good, Fair, Poor
##
## Robust SE were used (sandwich estimator of varcov).
##
##                           Estimate Std. Error z value Pr(>|z|)
## hypertenssionyes           0.19232    0.02478    7.76  8.4e-15 ***
## high_cholesterolyes        0.09780    0.02918    3.35  0.00080 ***
## heart_atack_or_strokeyes   0.34401    0.03183   10.81  < 2e-16 ***
## poor_mobilityyes           0.72832    0.03564   20.44  < 2e-16 ***
## very_poor_gripyes          0.49720    0.12299    4.04  5.3e-05 ***
## depressionyes              0.25323    0.02390   10.59  < 2e-16 ***
## respiratory_problemsyes    0.36777    0.03337   11.02  < 2e-16 ***
## IADL_problemsyes           0.61579    0.03637   16.93  < 2e-16 ***
## obeseyes                   0.18991    0.03295    5.76  8.3e-09 ***
## diabetesyes                0.33726    0.04010    8.41  < 2e-16 ***
## other_diseasesyes          0.33533    0.02370   14.15  < 2e-16 ***
## (L).1|2                   -0.09248    0.03194   -2.90  0.00379 **
## (L).2|3                   -0.26826    0.03236   -8.29  < 2e-16 ***
## (L).3|4                    0.07514    0.02905    2.59  0.00968 **
## (L).4|5                   -0.20346    0.05222   -3.90  9.8e-05 ***
## (G).sexwoman.1|2           0.02373    0.03015    0.79  0.43112
## (G).sexwoman.2|3           0.01366    0.03460    0.39  0.69304
## (G).sexwoman.3|4           0.03661    0.02869    1.28  0.20192
## (G).sexwoman.4|5           0.11848    0.05039    2.35  0.01872 *
## (G).ageclass[60,70).1|2   -0.01835    0.03383   -0.54  0.58763
## (G).ageclass[60,70).2|3    0.05336    0.04068    1.31  0.18962
## (G).ageclass[60,70).3|4    0.06003    0.03616    1.66  0.09693 .
## (G).ageclass[60,70).4|5    0.16842    0.06492    2.59  0.00949 **
## (G).ageclass[70,80).1|2   -0.32157    0.04391   -7.32  2.4e-13 ***
## (G).ageclass[70,80).2|3    0.17131    0.04774    3.59  0.00033 ***
## (G).ageclass[70,80).3|4    0.19360    0.03777    5.13  3.0e-07 ***
```

```
## (G).ageclass[70,80).4|5   0.23234    0.06653    3.49  0.00048 ***
## (G).ageclass[80,120).1|2 -0.33134    0.07274   -4.56  5.2e-06 ***
## (G).ageclass[80,120).2|3  0.14976    0.07590    1.97  0.04848 *
## (G).ageclass[80,120).3|4  0.17851    0.05025    3.55  0.00038 ***
## (G).ageclass[80,120).4|5  0.22378    0.07674    2.92  0.00354 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Theta: 1
## Log-likelihood: -12945.98
## Deviance: 25891.96
## AIC: 25953.96
```

model1 contains 11 dichotomous health variables and two threshold variables. The fitted coefficient can be accessed by *coef*() function

```
# extract parameters in a form of list
cm1 <- coef(model1, aslist = TRUE)

# types of returned coefficients
names(cm1)
```

```
## [1] "latent.params" "thresh.lambda" "thresh.gamma"  "logTheta"
```

```
# latent health variables
cm1$latent.params
```

```
##         hypertenssionyes      high_cholesterolyes heart_atack_or_strokeyes
##                0.1923166                0.0978032               0.3440052
##         poor_mobilityyes        very_poor_gripyes            depressionyes
##                0.7283236                0.4972017               0.2532285
##  respiratory_problemsyes         IADL_problemsyes                obeseyes
##                0.3677676                0.6157910               0.1899097
##              diabetesyes        other_diseasesyes
##                0.3372606                0.3353300
```

model1 can be further extended by adding country of origin to the treshold formula to control for cultural differences.

```
model2<- hopit(latent.formula = health ~ hypertenssion + high_cholesterol +
                              heart_atack_or_stroke + poor_mobility +
                              very_poor_grip + depression + respiratory_problems +
                              IADL_problems + obese + diabetes + other_diseases,
               thresh.formula = ~ sex + ageclass + country,
               decreasing.levels = TRUE,
               control=list(trace=FALSE),
               data = healthsurvey)
```

The fit of both models can be compared using AIC() function:

```
AIC(model2, model1)
```

```
##   model2   model1
## 25154.19 25953.96
```

or using Likelihood Ratio Test (LRT) as models are nested:

```
anova(model2, model1)
```

```
## Full model:
## -- Formula (latent variables):
## health ~ hypertenssion + high_cholesterol + heart_atack_or_stroke +
##     poor_mobility + very_poor_grip + depression + respiratory_problems +
##     IADL_problems + obese + diabetes + other_diseases
## -- Formula (threshold variables): ~sex + ageclass + country
## -- Theta: FALSE
##
## Nested model:
## -- Formula (latent variables):
## health ~ hypertenssion + high_cholesterol + heart_atack_or_stroke +
##     poor_mobility + very_poor_grip + depression + respiratory_problems +
##     IADL_problems + obese + diabetes + other_diseases
## -- Formula (threshold variables): ~sex + ageclass
## -- Theta: FALSE
##
## Likelihood ratio test:
##   Chi^2 df Pr(>Chi^2)
##  815.78  8     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both *latent.formula* and *thresh.formula* allow to specify interactions, like interaction between gender (*sex*) and age (*ageclass*):

```
model3<- hopit(latent.formula = health ~ hypertenssion * high_cholesterol +
                              heart_atack_or_stroke + poor_mobility +
                              very_poor_grip + depression + respiratory_problems +
                              IADL_problems + obese + diabetes + other_diseases,
               thresh.formula = ~ sex * ageclass + country,
               decreasing.levels = TRUE,
               control=list(trace=FALSE),
               data = healthsurvey)

print(anova(model3,model2), short=TRUE)
```

```
##
## Likelihood ratio test:
##   Chi^2 df Pr(>Chi^2)
##  28.923 13    0.00671 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The *hopit*() function has also an option to include survey design using the *survey* package. The example below fit a model using simple two level cluster sampling design

```
design <- svydesign(ids = ~ country + psu, weights = healthsurvey$csw, data = healthsurvey)
```

```
model2s<- hopit(latent.formula = health ~ hypertenssion + high_cholesterol +
                                 heart_atack_or_stroke + poor_mobility +
                                 very_poor_grip + depression + respiratory_problems +
                                 IADL_problems + obese + diabetes + other_diseases,
                thresh.formula = ~ sex + ageclass + country,
                decreasing.levels = TRUE,
                design = design,
                control=list(trace=FALSE),
                data = healthsurvey)
```

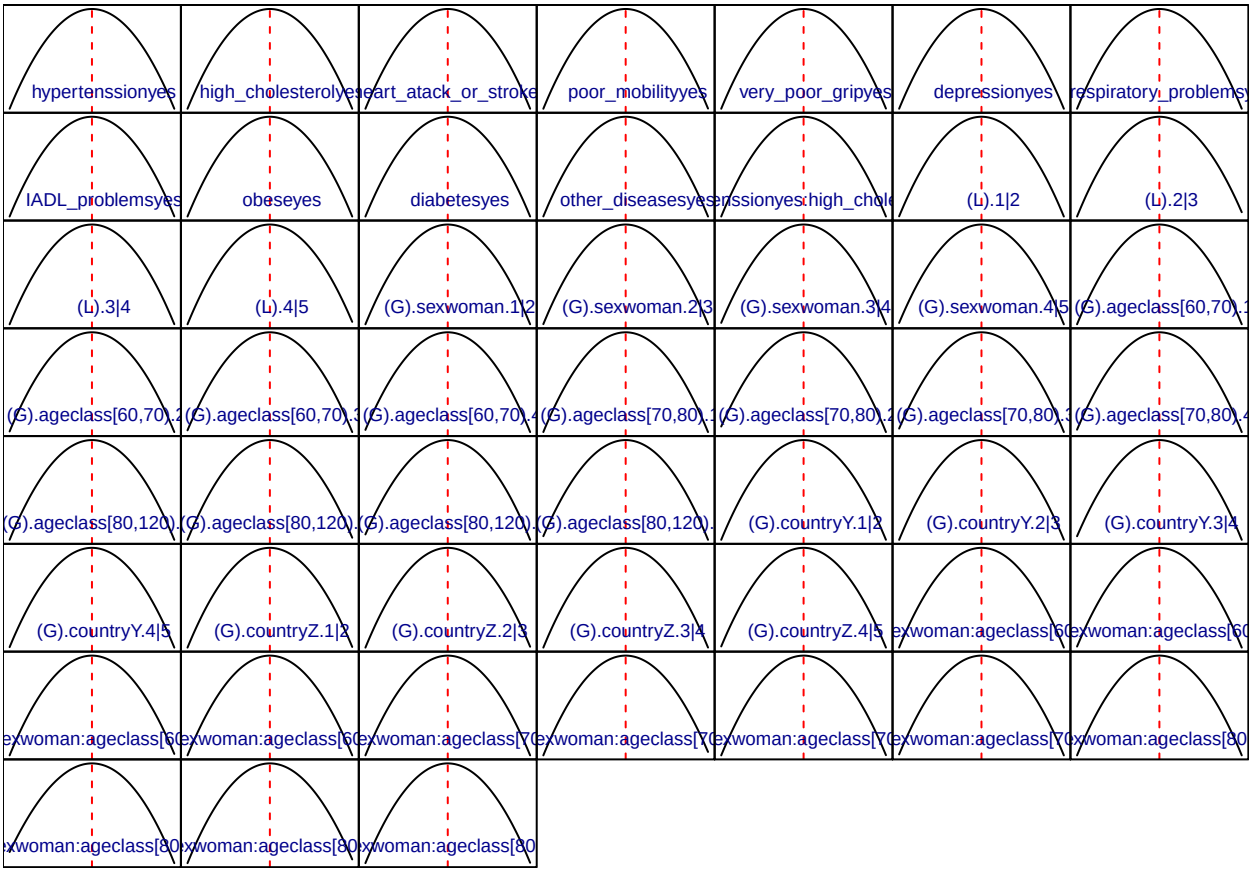
Ignoring survey design could lead to biased results. Here, in presented examples it has however an minor importance, which is seen by comparing coefficients of latent variable for both models:

```
cbind('No survey design'=coef(model2,aslist=TRUE)$latent.par,
      'Has survey design'=coef(model2s,aslist=TRUE)$latent.par)
```

```
##                          No survey design Has survey design
## hypertenssionyes               0.18475332        0.18777955
## high_cholesterolyes            0.08972562        0.09368829
## heart_atack_or_strokeyes       0.34659838        0.34676962
## poor_mobilityyes               0.70346456        0.70603471
## very_poor_gripyes              0.51424418        0.54768793
## depressionyes                  0.24998274        0.24922297
## respiratory_problemsyes        0.37863461        0.37984848
## IADL_problemsyes               0.59262343        0.60999389
## obeseyes                       0.19041874        0.18874130
## diabetesyes                    0.32839067        0.32328477
## other_diseasesyes              0.32936970        0.32876106
```

The fit accuracy of the model can be assessed using *profile*() function, which calculate and plot profile of the log likelihood function around fitted coefficient values.

```
profile(model3)
```




```
## All parameters seem to be at arg.max (at optimum).
```

6. Analyses of the reporting styles using *hopit* package

Let's look at latent health variables of *model2*.

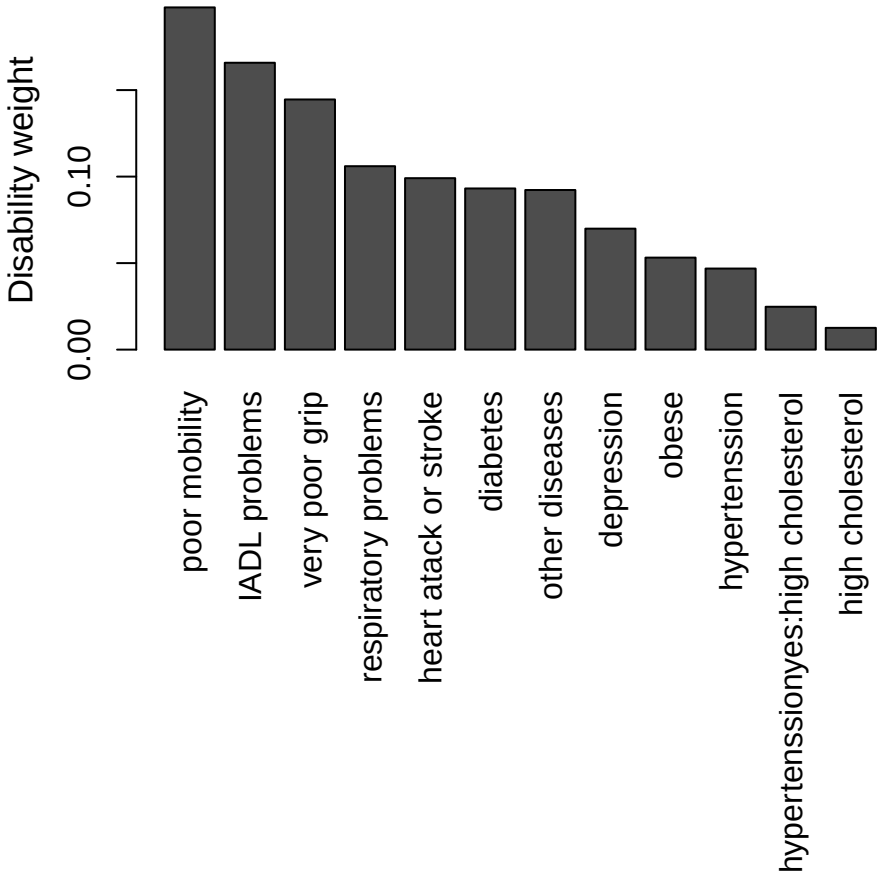
```
model3$coef.ls$latent.params
```

```
##                   hypertenssionyes                   high_cholesterolyes
##                         0.16692322                           0.04478911
##              heart_atack_or_strokeyes                      poor_mobilityyes
##                         0.35296700                           0.70450393
##                  very_poor_gripyes                         depressionyes
##                         0.51500758                           0.24905559
##               respiratory_problemsyes                      IADL_problemsyes
##                         0.37763441                           0.59062305
##                           obeseyes                         diabetesyes
##                         0.18939406                           0.33180729
##                 other_diseasesyes hypertenssionyes:high_cholesterolyes
##                         0.32861717                           0.08822596
```

We can standardize them using Jürges' approach (Jürges 2007) to obtain disability weights. The standardization can be doen using *standardizeCoef*() function.

```r
# A function that modifies coefficient names.
txtfun <- function(x) gsub('_',' ',substr(x,1,nchar(x)-3))

# Calcualte and plot disability weights
sc <- standardizeCoef(model3, plotf = TRUE, namesf = txtfun)
```



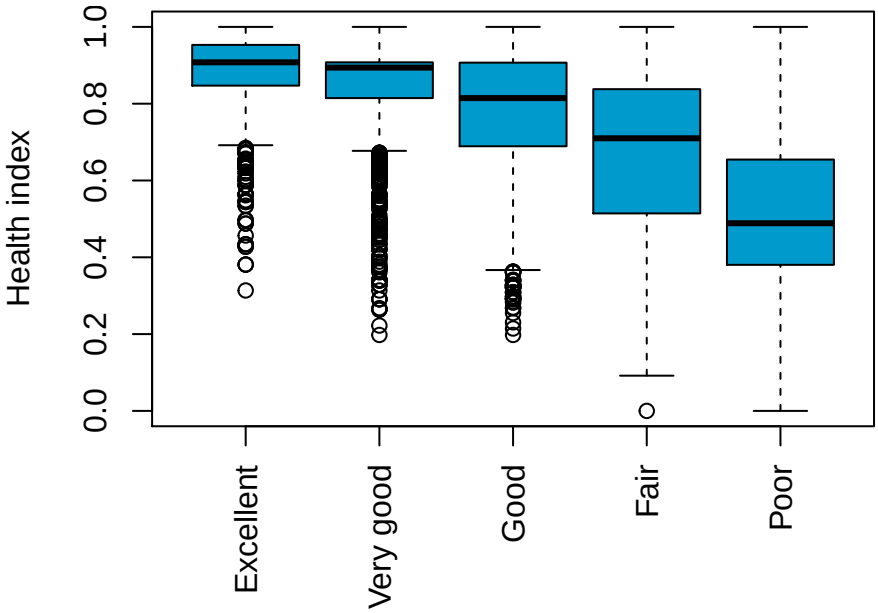

```r
sc
```

```
##                                      [,1]
## poor mobility                  0.19778802
## IADL problems                  0.16581620
## very poor grip                 0.14458732
## respiratory problems           0.10602008
## heart atack or stroke          0.09909476
## diabetes                       0.09315421
## other diseases                 0.09225859
## depression                     0.06992184
## obese                          0.05317199
## hypertenssion                  0.04686335
## hypertenssionyes:high cholesterol 0.02476926
## high cholesterol               0.01257445
```

The *namesf* argument is a function or a character vector that is used to rename the coefficent names. Here, it removes last 3 letters ("yes"), which is a reference level for each variable and exchanges "_" with spaces in variable names.

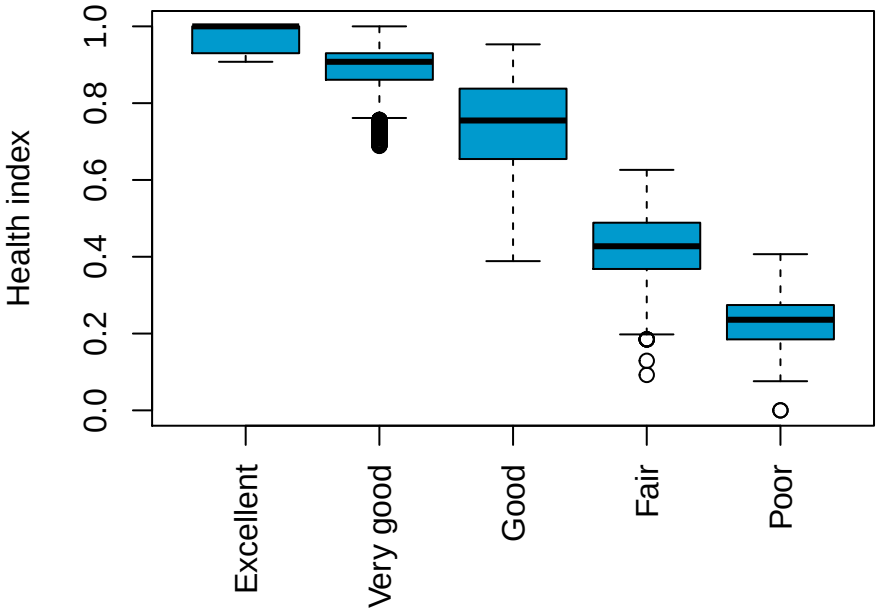
The latent index is simply calculated using *latentindex*() function.

```
hi <- latentIndex(model3, plotf = TRUE, response = "data",
                  ylab = 'Health index', col='deepskyblue3')
```



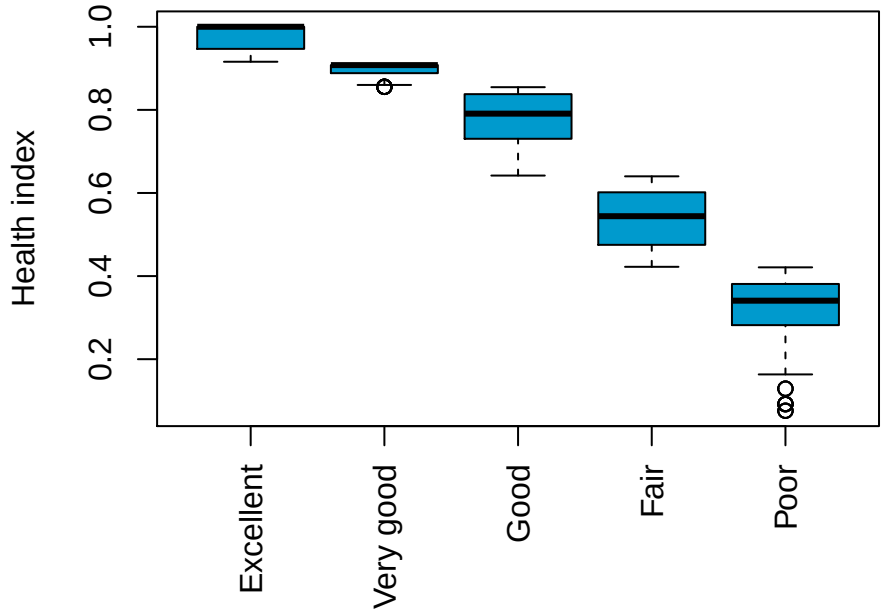
The boxplot shows reported health status vs. health index. It is also possible to plot expected categorical health status on Y axis calcualted according to the eq. (2).

```
hi <- latentIndex(model3, plotf = TRUE, response = "fitted",
                  ylab = 'Health index', col='deepskyblue3')
```



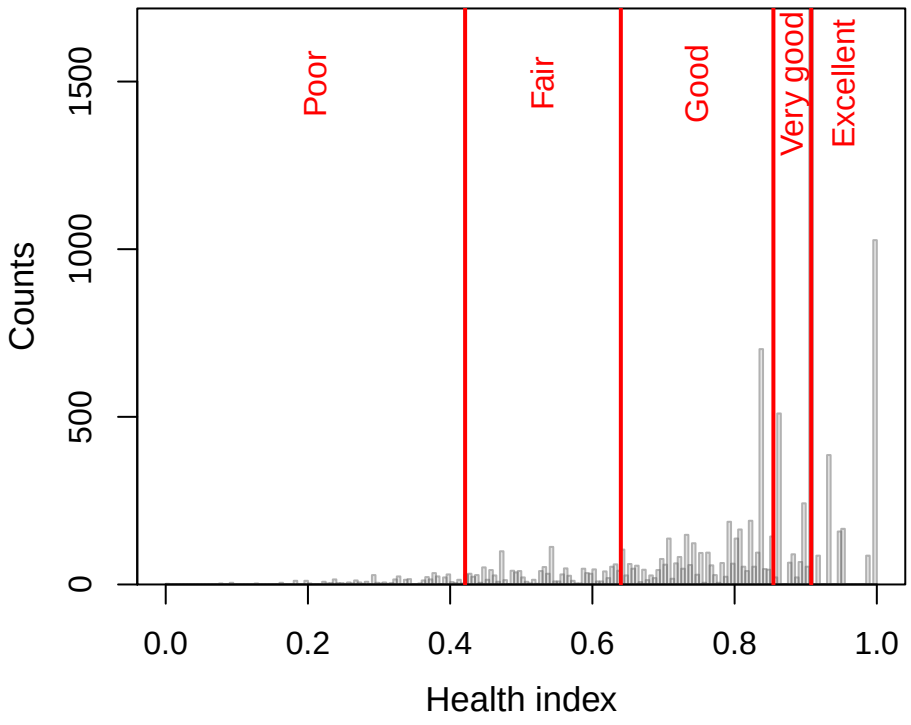
or according to Jürges' method:

```
hi <- latentIndex(model3, plotf = TRUE, response = "Jurges",
                  ylab = 'Health index', col='deepskyblue3')
```



The central part of reporting styles analyses is to determine the cut-points used to calcualte adjusted health status for each individual. The calcualtion and ploting of cut-points is realized by *getCutPoints*() function.

```
z=getCutPoints(model=model3)
```



```r
# Health index cut-points
z$cutpoints
```

```
##     4.41%    17.68%    52.34%    77.63%
## 0.4210433 0.6400315 0.8545694 0.9077414
```

```r
#Adjusted health levels for individuals: Jurges method
table(z$adjused.health.levels)
```

```
##
##      Poor      Fair      Good Very good Excellent
##       450      1347      3505      2787      1909
```

```r
#Adjusted health levels for individuals: Estimated model thresholds
table(model3$Ey_i)
```

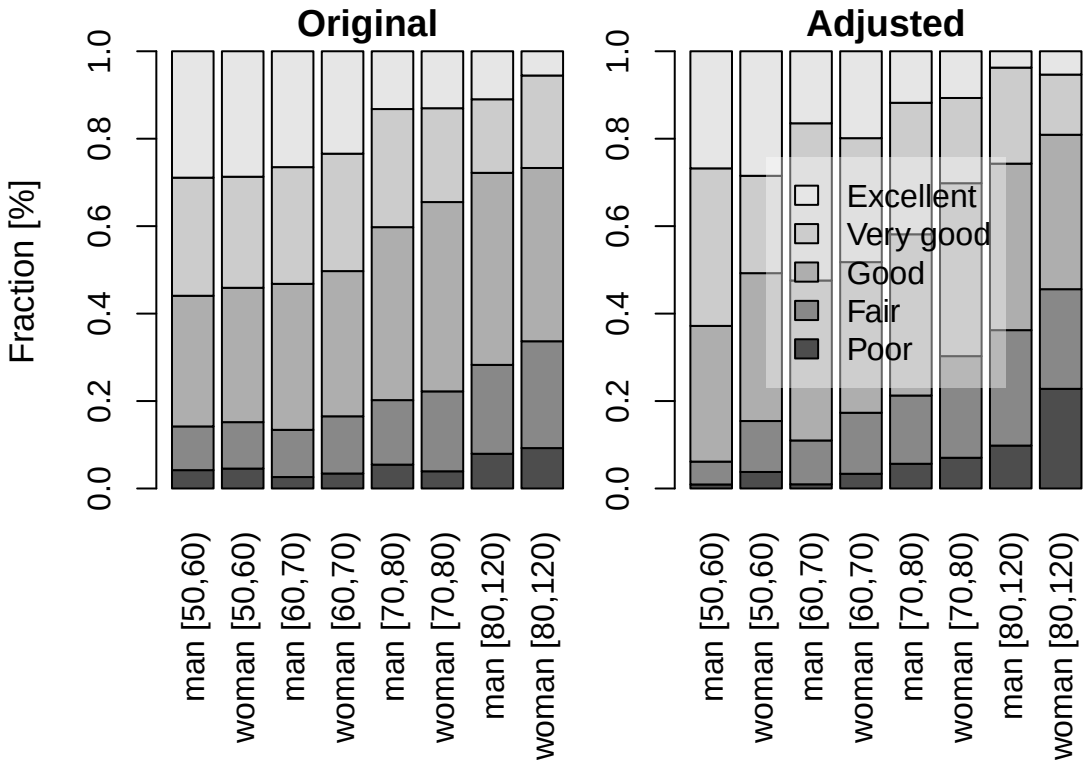
```
##
## Excellent Very good      Good      Fair      Poor
##       737      4361      4029       807        66
```

```r
#Original health levels for individuals
table(model3$y_i)
```

```
##
## Excellent Very good      Good      Fair      Poor
##      2237      2529      3466      1327       441
```

The analysis of health levels is done by *getLevels*() function

```r
# Health levels for combination of age and gender, and pooled country of origin.
hl <- getLevels(model=model3, formula=~ sex + ageclass, data = healthsurvey,
                      sep=' ', plotf=TRUE)
```



The differences between original and adjusted frequncies can be calcualted directly using *getLevels* output:

```
round(100*(hl$original - hl$adjusted),2)
```

```
##
##                     Poor   Fair   Good Very good Excellent
##    man [50,60)      3.28   4.77  -1.15     -9.02      2.13
##    woman [50,60)    0.77  -1.03  -3.09      3.13      0.21
##    man [60,70)      1.66   0.77  -3.19     -9.26     10.03
##    woman [60,70)    0.06  -0.87  -1.25     -1.50      3.55
##    man [70,80)     -0.19  -0.86   2.68     -3.07      1.44
##    woman [70,80)   -3.09  -4.97   3.75      1.97      2.34
##    man [80,120)    -1.87  -6.07   5.84     -5.14      7.24
##    woman [80,120) -13.55   1.64   4.31      7.39      0.21
```

7. Bootstraping Confidence Intervals

The package offers functions to calculate confidence intervals for any measure derived from the model. As an example we show calcualtion of confidence intervals of the difference between original and adjusted frequencies of combined "Poor" + "Fair" healt categories.

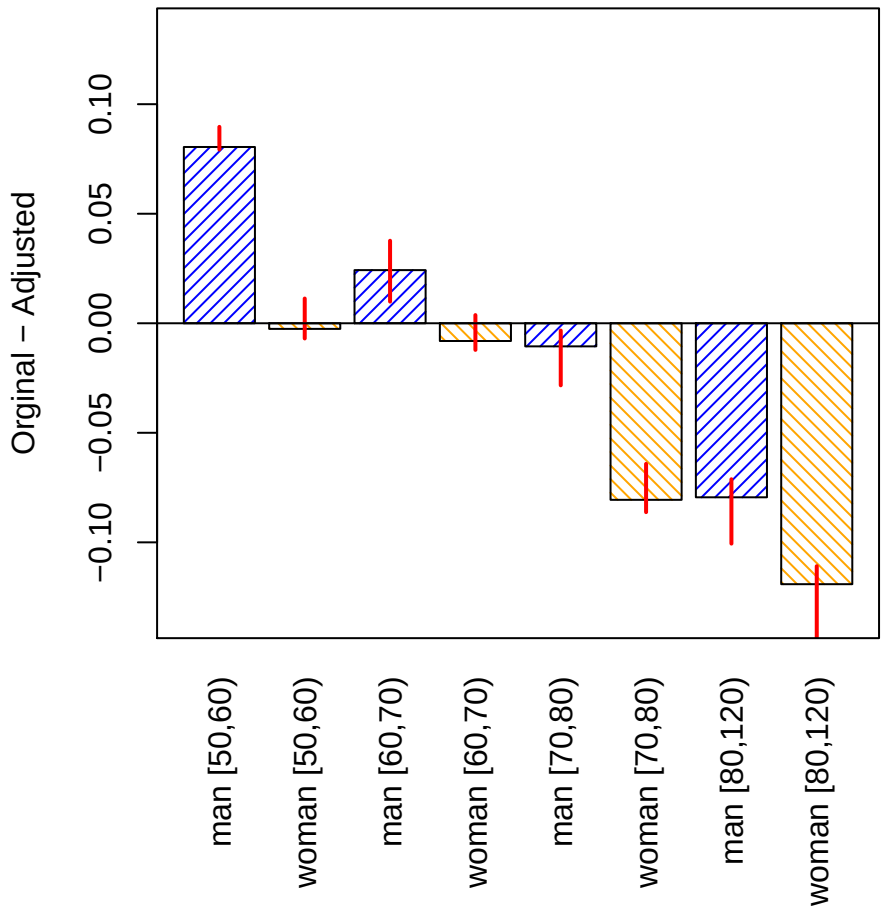
```
# Function to be bootstraped
diff_BadHealth <- function(model, data) {
  hl <- getLevels(model=model, formula=~ sex + ageclass, data = data,
                  sep=' ', plotf=FALSE)
  hl$original[,1] + hl$original[,2] - hl$adjusted[,1]- hl$adjusted[,2]
}
```

```r
# Estimate of the difference
est.org <- diff_BadHealth(model = model3, data = healthsurvey)

# Perform the bootstrap
B <- boot_hopit(model = model3, data = healthsurvey,
                func = diff_BadHealth, nboot = 100)

# Calcualte lower and upper bounds using percentile method
est.CI <- boot_hopit_CI(B)

# Plotting the difference and its (assymetrical) confidence intervals
pmar <- par('mar'); par(mar = c(9.5,pmar[2:4]))
m <- max(abs(est.CI))
pos <- barplot(est.org, names.arg = names(est.org), las = 3, ylab = 'Orginal - Adjusted',
               ylim=c(-m, m), density = 20, angle = c(45, -45), col = c('blue', 'orange'))
for (k in seq_along(pos)) lines(c(pos[k,1],pos[k,1]), est.CI[,k], lwd = 2, col = 2)
abline(h = 0); box(); par(mar = pmar)
```



The results show that men tend to over-report bad health at ages (50,60] and (50,70], wherease women at ages [70,80) and both sexes at ages (80, 120] under-report bad health.

8. References

Boes, Stefan, and Rainer Winkelmann. 2006. "Ordered Response Models." *Allgemeines Statistisches Archiv* 90 (1): 167–81. https://doi.org/10.1007/s10182-006-0228-y.

Greene, William, Mark N. Harris, Bruce Hollingsworth, and Timothy A. Weterings. 2014. "Heterogeneity in Ordered Choice Models: A Review with Applications to Self-Assessed Health." *Journal of Economic Surveys* 28 (1): 109–33. https://doi.org/10.1111/joes.12002.

Greene, William, and David Hensher. 2010. *Modeling Ordered Choices*. Cambridge University Press. https://EconPapers.repec.org/RePEc:cup:cbooks:9780521142373.

Ierza, Joseph V. 1985. "Ordinal Probit: A Generalization." *Communications in Statistics - Theory and Methods* 14 (1). Taylor & Francis: 1–11. https://doi.org/10.1080/03610928508828893.

Jürges, Hendrik. 2007. "True Health Vs Response Styles: Exploring Cross-Country Differences in Self-Reported Health." *Health Economics* 16 (2): 163–78. https://doi.org/10.1002/hec.1134.

King, Gary, Christopher J. L. Murray, Joshua A. Salomon, and Ajay Tandon. 2004. "Enhancing the Validity and Cross-Cultural Comparability of Measurement in Survey Research." *American Political Science Review* 98 (1). Cambridge University Press: 191–207. https://doi.org/10.1017/S000305540400108X.

Liao, Pei-Shan, Yang-Chih Fu, and Chin-Chun Yi. 2005. "Perceived Quality of Life in Taiwan and Hong Kong: An Intra-Culture Comparison." *Journal of Happiness Studies* 6 (1): 43–67. https://doi.org/10.1007/s10902-004-1753-6.

McKelvey, R. D., and W. Zavoina. 1975. "A Statistical Model for the Analysis of Ordinal Level Dependent Variables." *Journal of Mathematical Sociology* 4 (1): 103–20.

Rebelo, Luis Pina, and Nuno Sousa Pereira. 2014. "Assessing Health Endowment, Access and Choice Determinants: Impact on Retired Europeans' (in)activity and Quality of Life." *Social Indicators Research* 119 (3). Germany: Springer: 1411–46. https://doi.org/10.1007/s11205-013-0542-1.