Analysis of reporting behavior using generalized ordered probit models: an introduction to the *hopit* package

Maciej J. Dańko

29 III 2019

Abstract

The *hopit* package provides R functions to fit and analyze ordered response data in the context of reporting heterogeneity. In this vignette I describe the formulation and the fit of *hopit* models as well as functions that can be used to analyze heterogenity in reporting behavior.

1. Introduction

hopit is an open source software library written in the R (R-Core-Team 2018) and C++ (Bates and Eddelbuettel 2013; Eddelbuettel and François 2011) programming languages. The *hopit* package provides versatile methods for fitting and analyzing ordered response data in the context of heterogeneity in self reporting behavior.

The ordered response data classify a measure of interest into ordered categories collected during a survey. For example, if the dependent variable is a happiness rating, then a respondent typically answers a question such as: "Taking all things together, would you say you are . . . ?"; and then selects from response options such as:"very happy“,"pretty happy“,"not too happy“, and"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, such as "very good", "good", "fair", "bad", and "very bad" (King et al. 2004; Jürges 2007; Rebelo and Pereira 2014; Oksuzyan et al. 2019). In political science, 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 categories like "strongly oppose", "mildly oppose", "indifferent", "mildly support", and "strongly support" (Greene and Hensher 2010). It is easy to imagine other multi-level ordinal variables that might be used during a survey and to which the methodology described below could be applied.

In practice, it is assumed that when responding to a survey question about their general happiness, health, feelings, attitudes or other status, participants are assessing their true value of this unobserved continuous variable, and project it onto the discrete scale provided. The thresholds that individuals use to categorize their true status by selecting a specific response option may be affected by the reference group chosen, their earlier life experiences, and cross-cultural differences in using scales. Thus, the responses of individuals may differ depending on their gender, age, cultural background, education, and personality traits; among other factors.

From the perspective of reporting behavior modeling, one of the main tasks researchers face is to compute this continuous estimate of the underlying, latent measures of individuals based on several specific characteristics of the responses considered (e.g., health variables or happiness variables), and to account for variations in reporting across socio-demographic and cultural groups. More specifically, to build a latent, underlying measure, a generalized hierarchical ordered threshold model is fitted that regresses the reported status/attitude/feeling on two sets of independent variables (Boes and Winkelmann 2006; Greene et al. 2014). When the dependent reported ordered variable is self-rated health status, then the first set of variables – i.e., health variables – assess specific aspects of individuals' health, such as measures of chronic conditions, mobility, difficulties with a range of daily activities, grip strength, anthropometric characteristics, and lifestyle behaviors. Using the second set of independent variables (threshold variables), the model also adjusts for differences across socio-demographic and cultural groups, such as differences in cultural background, gender, age, and education (King et al. 2004; Jürges 2007; Oksuzyan et al. 2019; but see Rebelo and Pereira 2014).

Once the model is fitted, its estimates (latent measure and threshold coefficients) can be used to calculate the differences in reporting behavior among groups of people with different contextual characteristics through the calculation of differences between the expected and the reported ordinal response measures (Jürges 2007; Oksuzyan et al. 2019).

Table 1. Glossary.

Term	Symbol	Definition	Exemplary case specific synonyms
Categorical response	y	Dependent variable obtained during the survey	Self-rated health, self-rated happiness
Latent measure	h	Modeled continuous latent measure of the investigated response variable	Latent health, latent happiness
Latent index	H	Standardized latent measure	Health index, happiness index
Latent variables	—	Variables used to model the latent measure	Health variables, happiness variables
Latent terms	X	Terms of the design matrix used to model the latent measure	
Latent coefficients	β	Coefficients corresponding to each latent term	
Standardized coefficient	D	Standardized value of a coefficient	Disability weights
Thresholds	α	Thresholds used to group the latent measure	Cut-points
Threshold variables	—	Variables used to model the thresholds	Socio-demographic, cultural, contextual variables
Threshold terms	Y	Terms of the design matrix used to model the latent measure	
Threshold coefficients	γ, λ	Coefficients corresponding to each threshold term	

2. Generalized (hierarchical) ordered threshold model

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

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

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

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

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

The thresholds (cut points, α) are modeled using threshold variables coded as design matrix Y, their coefficients γ, and intercepts λ. It is assumed that they model the contextual characteristics of the respondent (e.g. country, gender, and age). The threshold variables are modeled without applying the 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, parametrization of King et al. (2004) and Jürges (2007) is used, which assumes that:

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

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

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

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 behavior analyses the probit model is typically chosen. This model 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 as:

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

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 heterogeneity

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

One of the first steps of the analysis is the standardization of the latent measure to obtain the 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 the true underlying health of an individual, and varies from 0, representing the (model-based) worst health state in the sample, to 1, representing the (model-based) best health state in the sample.

The predicted latent measure h_i obtained from the model is also used to standardize the latent variable coefficients. In the self-rated health example the standardized coefficients are called disability weights D_k (Jürges 2007; Oksuzyan et al. 2019) and are calculated for each health variable to provide information about the impact of a specific health measure on the latent index H_i. The disability weight for a health variable is equal to the ratio of the corresponding health coefficient and the difference between the lowest and the highest values of predicted latent health. In other words, the disability weight reduces H_i by some given amount or percentage (i.e. the H_i of every individual is reduced by the same amount if the person had a heart attack or other heart problems)(Jürges 2007; Oksuzyan et al. 2019).

$$D_k = \frac{\beta_k}{\max_i h_i - \min_i h_i} \tag{9}$$

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

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

Reporting behavior analysis is based on the reclassification of individuals into new response categories. There are two methods of reclassification: (1) Jürges (2007) percentile method (see also Rebelo and Pereira 2014, and @OKSUZYAN2019) and (2) reclassification based on estimated thresholds.

In the first method, the classification is based on the calculated latent index H_i and is thus adjusted for inter-individual differences in reporting behavior. Jürges' percentile method is based on the original distribution of the 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 $\mathbf{1}$ is an indicator function taking 1 if the condition is true, or is 0 otherwise. The calculated cumulative frequencies of the latent index H_i are used as percentiles (cut-points), so that each individual i can be reclassified into new response categories.

In the second method the reclassification is based on eq. (2), so that each individual has her/his own, model-derived cut-points.

4. Installing and loading the package

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

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

```
library(hopit)
```

In the examples presented below I use *healthsurvey*, which is a completely artificial data set yhat is simulated using the distributions of some major health and socio-demographic characteristics. The distributions and the data structure is roughly based on the WAVE1 SHARE database (DOIs: 10.6103/SHARE.w1.600) see Börsch-Supan et al. (2013) for technical details. See also the acknowledgements.

```r
# load *healthsurvey* dataset
data(healthsurvey)

# horizontal view of the dataset (omitting ID)
print(t(healthsurvey[1:6,-1]), quote=FALSE, na.print='NA', right=TRUE)
```

```
##                                1       2      3      4         5      6
## health                 Very good    Good   Good   Good Excellent   Good
## diabetes                      no      no    yes     no        no     no
## obese                         no      no     no     no        no     no
## IADL_problems                 no      no     no     no        no     no
## hypertension                  no     yes     no     no        no    yes
## high_cholesterol              no     yes     no     no        no    yes
## respiratory_problems          no      no     no     no        no    yes
## heart_attack_or_stroke        no     yes     no     no        no     no
## poor_mobility                 no      no     no     no        no    yes
## very_poor_grip                no      no     no     no        no     no
## depression                    no      no     no    yes        no     no
## other_diseases               yes     yes     no     no        no    yes
## sex                          man     man    man    man     woman    man
## ageclass                     80+   70-79  50-59  60-69       80+    80+
## education                  prim-   prim-  prim-   sec+     prim-  prim-
## country                        Y       Y      X      Y         Z      Y
## csw                      2407.48 1198.12 885.26 772.04   1304.24 917.16
## psu                           YB      YB     XC     YA        ZB     YD
```

The first variable on the list (*health*) is the categorical self-reported health status. This variable is followed by 11 determinants of health, which includes information on the presence of chronic diseases and other health conditions. The *sex*, *ageclass*, *education*, and *country* variables describe the contextual characteristics of individuals. The last group of variables (*csw*, *psu*, and *ssu*) describes the survey design.

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

The generalized ordered probit model can be fitted using the *hopit*() function. The function takes two kinds of formulas: (1) *latent.formula*, which models the impact of the latent variables on categorical health; and (2) *thresh.formula, which models the thresholds.

```r
# first determine the order of the dependent variable
levels(healthsurvey$health)
```

```
## [1] "Excellent" "Very good" "Good"      "Fair"      "Poor"
```

```r
# the order is decreasing (from the best to the worst health state)
# so we set: decreasing.levels = TRUE
model1<- hopit(latent.formula = health ~ hypertension + high_cholesterol +
                             heart_attack_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 ~ hypertension + high_cholesterol + heart_attack_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 variance-covariance matrix).
##
##                            Estimate Std. Error z value Pr(>|z|)
## hypertensionyes             0.19232    0.02478    7.76  8.4e-15 ***
## high_cholesterolyes         0.09780    0.02918    3.35  0.00080 ***
## heart_attack_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  0.43111
## (G).sexwoman.2|3            0.01366    0.03460    0.39  0.69304
## (G).sexwoman.3|4            0.03661    0.02869    1.28  0.20193
## (G).sexwoman.4|5            0.11847    0.05039    2.35  0.01872 *
## (G).ageclass60-69.1|2      -0.01835    0.03383   -0.54  0.58763
## (G).ageclass60-69.2|3       0.05336    0.04068    1.31  0.18963
## (G).ageclass60-69.3|4       0.06003    0.03616    1.66  0.09693 .
## (G).ageclass60-69.4|5      0.16842    0.06492    2.59  0.00949 **
## (G).ageclass70-79.1|2     -0.32157    0.04391   -7.32  2.4e-13 ***
## (G).ageclass70-79.2|3      0.17131    0.04774    3.59  0.00033 ***
## (G).ageclass70-79.3|4      0.19360    0.03777    5.13  3.0e-07 ***
## (G).ageclass70-79.4|5      0.23235    0.06653    3.49  0.00048 ***
## (G).ageclass80+.1|2       -0.33134    0.07274   -4.56  5.2e-06 ***
## (G).ageclass80+.2|3        0.14976    0.07590    1.97  0.04848 *
## (G).ageclass80+.3|4        0.17851    0.05025    3.55  0.00038 ***
## (G).ageclass80+.4|5        0.22379    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 includes βs ("latent.params", first 11 coefficients in the summary), λs ("thresh.lambda", threshold intercepts, "(L)" prefix in the summary), and γs ("thresh.gamma", parameters related to threshold covariates, "(G)" prefix in the summary). The model coefficients can be accessed by the *coef*() function.

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

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

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

```
# extracting the latent health coefficients
cm1$latent.params
```

```
##           hypertensionyes       high_cholesterolyes
##                0.19231663                0.09780304
## heart_attack_or_strokeyes          poor_mobilityyes
##                0.34400564                0.72832334
##         very_poor_gripyes             depressionyes
##                0.49720377                0.25322841
##   respiratory_problemsyes          IADL_problemsyes
##                0.36776716                0.61579072
##                  obeseyes               diabetesyes
##                0.18990979                0.33726056
##         other_diseasesyes
##                0.33532985
```

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

```
model2<- hopit(latent.formula = health ~ hypertension + high_cholesterol +
                      heart_attack_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 the two models can be compared using the AIC() function:

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

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

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

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

```
## Full model:
## -- Formula (latent variables):
## health ~ hypertension + high_cholesterol + heart_attack_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 ~ hypertension + high_cholesterol + heart_attack_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 the user to specify the interactions, like the interaction between gender (*sex*) and age (*ageclass*):

```
model3<- hopit(latent.formula = health ~ hypertension + high_cholesterol +
                      heart_attack_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)
##  26.498 12    0.00912 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interactions between latent and threshold variables can also be modeled. Depending on how an interaction is interpreted, it can be added to either the latent or the threshold formula:

```
model4<- hopit(latent.formula = health ~ hypertension + high_cholesterol +
                      heart_attack_or_stroke + poor_mobility +
                      very_poor_grip + depression + respiratory_problems +
                      IADL_problems + obese + diabetes + other_diseases +
                      sex : respiratory_problems,
               thresh.formula = ~ sex * ageclass + country + sex : depression,
               decreasing.levels = TRUE,
               control=list(trace=FALSE),
               data = healthsurvey)

print(anova(model3,model4), short=TRUE)
```

```
##
## Likelihood ratio test:
##   Chi^2 df Pr(>Chi^2)
##  15.221  5    0.00946 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

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

```
model2s<- hopit(latent.formula = health ~ hypertension + high_cholesterol +
                       heart_attack_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)
```

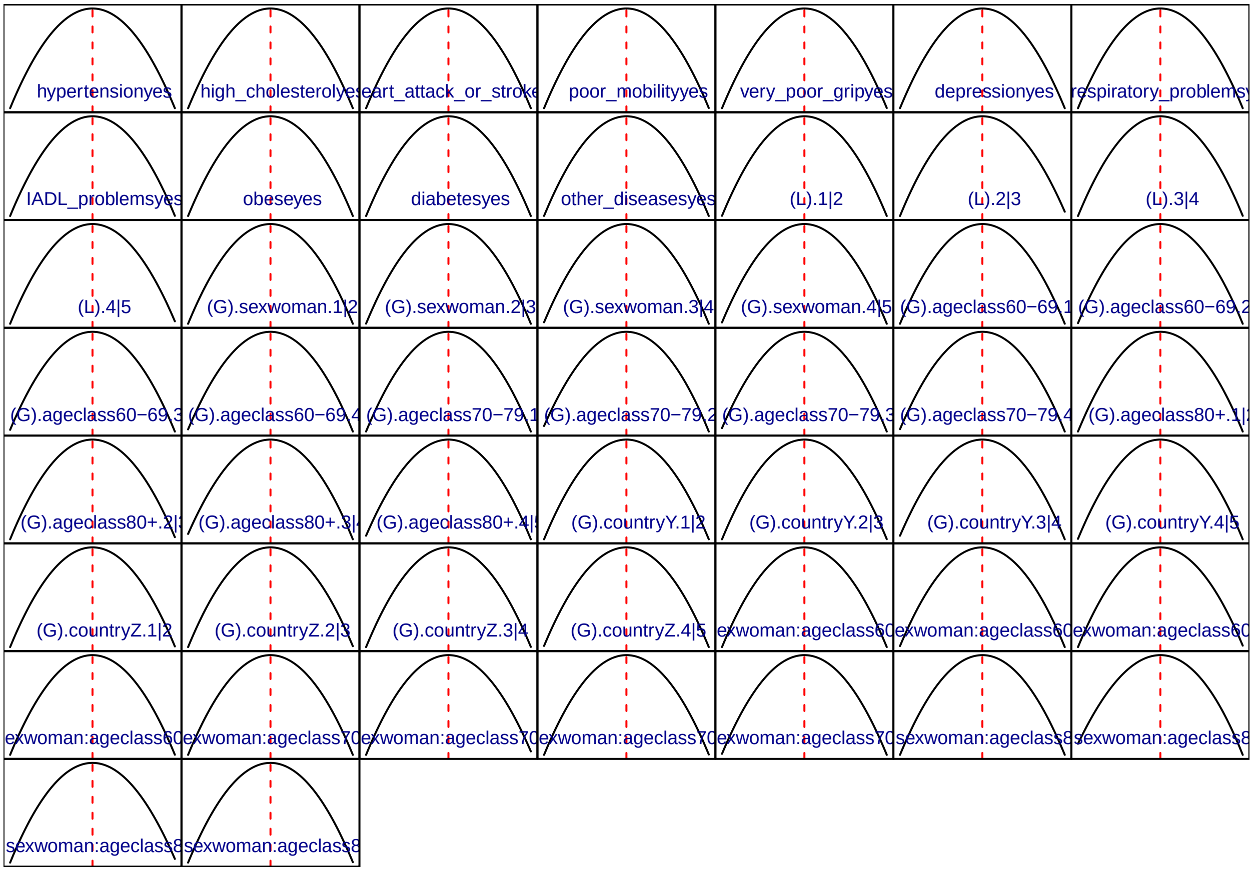
Generally, ignoring the survey design can lead to biased results. In the example presented here, the design has little importance, which is seen by comparing the coefficients of the latent variable for the two models:

```
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
## hypertensionyes                   0.18475331        0.18777953
## high_cholesterolyes               0.08972564        0.09368831
## heart_attack_or_strokeyes         0.34659837        0.34676967
## poor_mobilityyes                  0.70346452        0.70603472
## very_poor_gripyes                 0.51424239        0.54768701
## depressionyes                     0.24998271        0.24922293
## respiratory_problemsyes           0.37863461        0.37984846
## IADL_problemsyes                  0.59262348        0.60999391
## obeseyes                          0.19041873        0.18874123
## diabetesyes                       0.32839064        0.32328477
## other_diseasesyes                 0.32936965        0.32876101
```

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

```
profile(model3)
```



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

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

Let us now look at the latent health variables in *model3*.

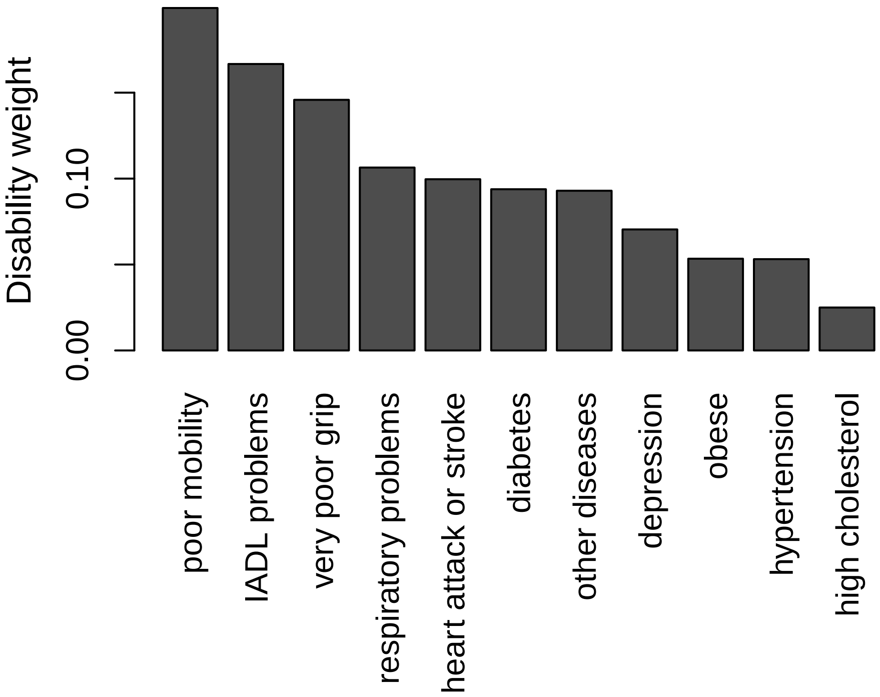
```
model3$coef.ls$latent.params
```

```
##            hypertensionyes       high_cholesterolyes
##                 0.18807936                0.08845808
## heart_attack_or_strokeyes          poor_mobilityyes
##                 0.35272886                0.70549437
##          very_poor_gripyes             depressionyes
##                 0.51631929                0.24936182
##    respiratory_problemsyes          IADL_problemsyes
##                 0.37683713                0.59016958
##                   obeseyes               diabetesyes
##                 0.18900796                0.33214901
##          other_diseasesyes
##                 0.32904712
```

We can standardize the coefficients of these variables using Jürges' approach (Jürges 2007) to obtain disability weights. The standardization can be done using the *standardizeCoef*() function.

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

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



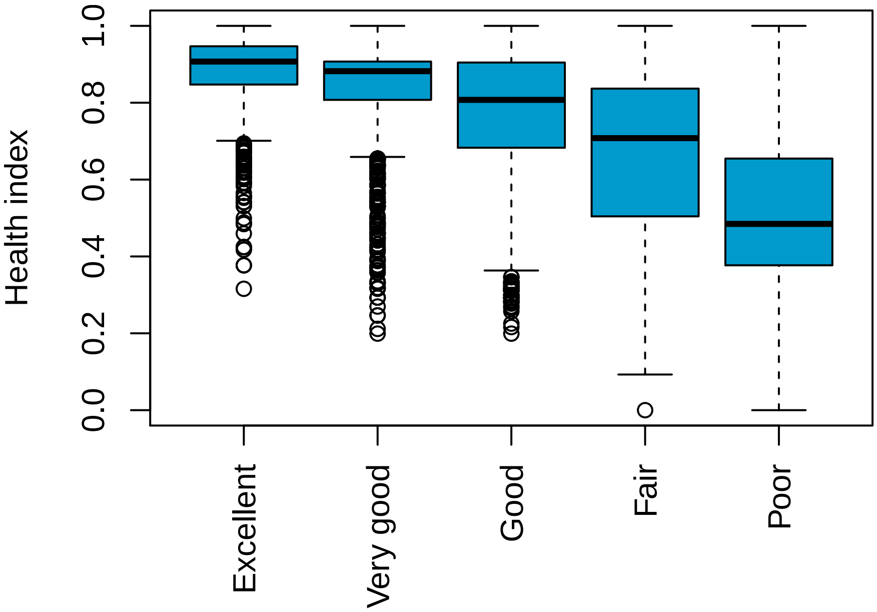
```
sc
```

```
##                              [,1]
## poor mobility          0.19924630
## IADL problems          0.16667618
## very poor grip         0.14581932
## respiratory problems   0.10642665
## heart attack or stroke 0.09961797
## diabetes               0.09380580
## other diseases         0.09292976
## depression             0.07042497
## obese                  0.05337978
## hypertension           0.05311753
## high cholesterol       0.02498240
```

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

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

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



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

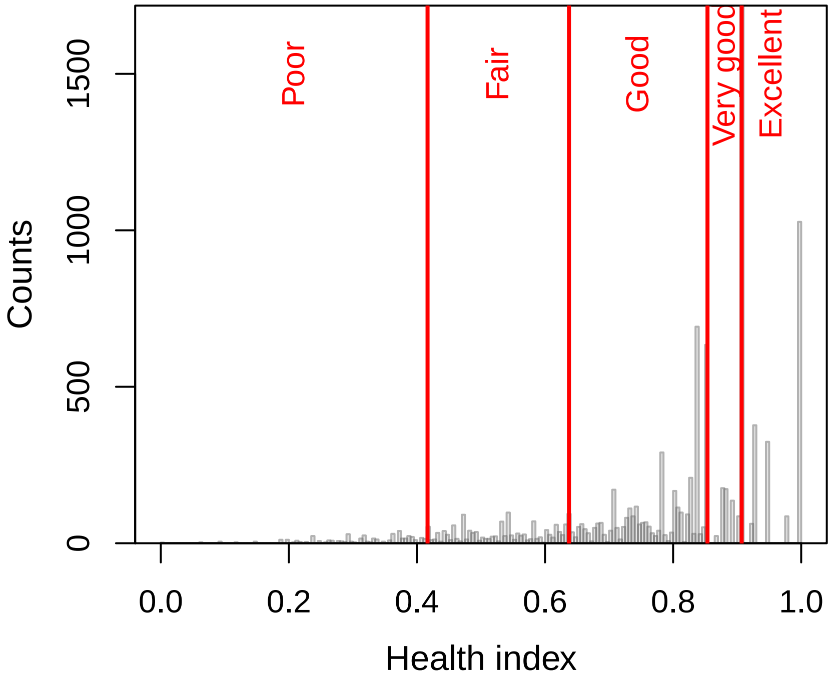
```
hi <- latentIndex(model3, plotf = TRUE, response = "fitted",
                  ylab = 'Health index', col='deepskyblue3')
```

or according to Jürges (2007) method:

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

The main aim of reporting heterogeneity analyses is to determine the cut-points used to calculate the adjusted health status for each individual. The calculation and the plotting of cut-points is realized using the *getCutPoints*() function.

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



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

```
##      4.41%     17.68%     52.34%     77.63%
## 0.4165474 0.6373990 0.8536905 0.9070702
```

```
# Adjusted health levels for individuals (using the Jürges method)
rev(table(z$adjusted.levels))
```

```
##
## Excellent Very good      Good      Fair      Poor
##      1876      2804      3506      1368       446
```

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

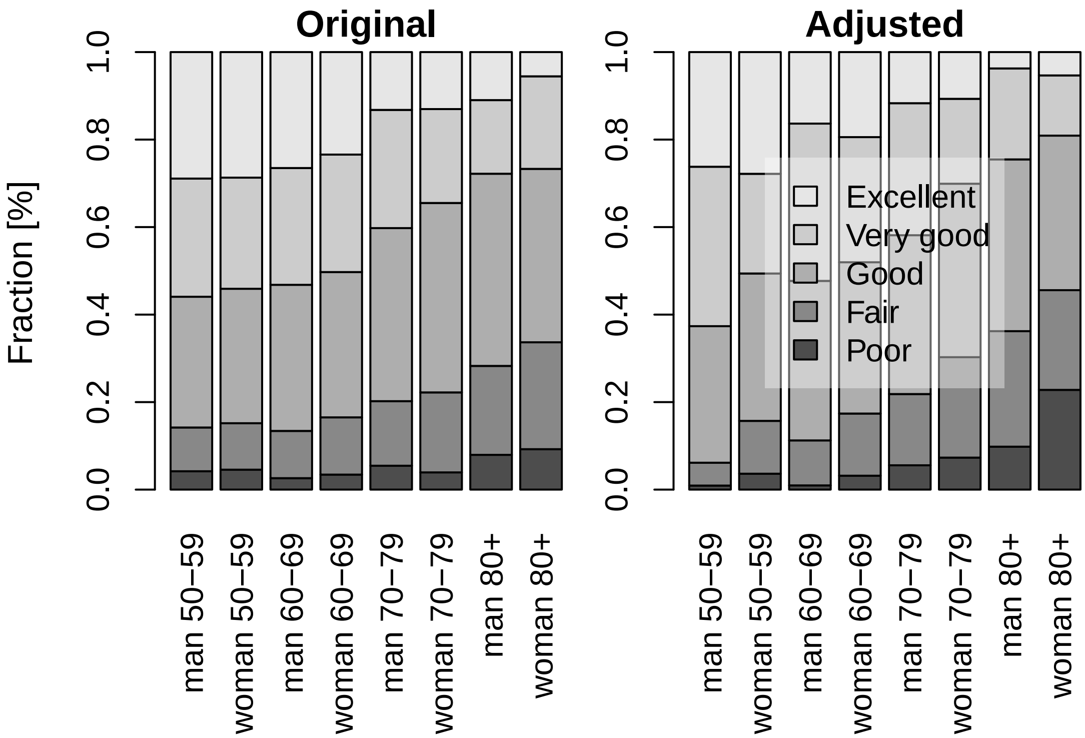
```
##
## Excellent Very good      Good      Fair      Poor
##      2237      2529      3466      1327       441
```

```
# Adjusted health levels for individuals (using estimated model thresholds)
table(model3$Ey_i)
```

```
##
## Excellent Very good      Good      Fair      Poor
##       734      4438      3959       805        64
```

The analysis of health levels is performed using the *getLevels*() function

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



The differences between the original and the adjusted frequencies can be calculated directly using the *getLevels* output:

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

```
##
##                 Poor  Fair  Good Very good Excellent
##   man 50-59     3.28  4.77 -1.32     -9.43      2.70
##   woman 50-59   0.92 -1.45 -2.96      2.62      0.87
##   man 60-69     1.66  0.51 -3.07     -9.26     10.15
##   woman 60-69   0.25 -1.12 -1.37     -1.74      3.99
##   man 70-79    -0.10 -1.53  3.26     -3.16      1.53
##   woman 70-79  -3.37 -4.69  3.66      2.06      2.34
##   man 80+      -1.87 -6.07  4.67     -3.97      7.24
##   woman 80+   -13.55  1.64  4.31      7.39      0.21
```

7. Bootstrapping Confidence Intervals

The package offers functions for calculating the confidence intervals for any measure derived from the model. In the example below, we calculate the confidence intervals of the difference between the original and the adjusted frequencies of bad health. The adjusted frequencies are determined by the presence of "*Poor*" or "*Fair*" self-rated health categories.

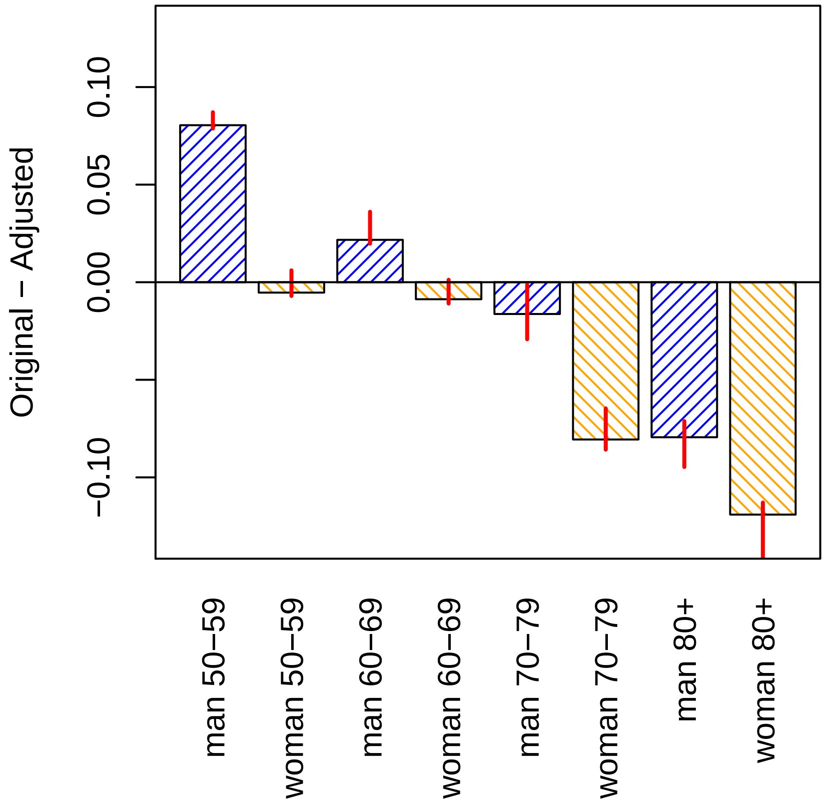
```r
# the function to be bootstraped
diff_BadHealth <- function(model, data) {
  hl <- getLevels(model = model, formula = ~ sex + ageclass, data = data,
                  sep = ' ', plotf = FALSE)
  hl$original[,1] + hl$original[,2] - hl$adjusted[,1]- hl$adjusted[,2]
}

# estimate 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)

# calculate lower and upper bounds using the percentile method
est.CI <- percentile_CI(B)

# plot the difference and its (asymmetrical) 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 = 'Original - 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], whereas women tend to over-report bad health at ages [70,80); and that both sexes at ages (80, 120] tend to under-report bad health.

8. Acknowledgements

I thank Anna Oksuzyan, Christian Dudel, Marius Pascariu, Laszlo Nemeth, and Oskar Burger for their help and suggestions on the package and the vignette. I also thank the Max-Planck Institute for Demographic Research for all their support.

The artificially generated data used in the package examples is based on the distributions of some major socio-demographic and health related characteristics of the WAVE 1 SHARE database. None of the records represent any part of the true data.

The SHARE data collection has been primarily funded by the European Commission through FP5 (QLK6-CT-2001-00360), FP6 (SHARE-I3: RII-CT-2006-062193, COMPARE: CIT5-CT-2005-028857, SHARELIFE: CIT4-CT-2006-028812) and FP7 (SHARE-PREP: N°211909, SHARE-LEAP: N°227822, SHARE M4: N°261982). Additional funding from the German Ministry of Education and Research, the Max Planck Society for the Advancement of Science, the U.S. National Institute on Aging (U01_AG09740-13S2, P01_AG005842, P01_AG08291, P30_AG12815, R21_AG025169, Y1-AG-4553-01, IAG_BSR06-11, OGHA_04-064, HHSN271201300071C) and from various national funding sources is gratefully acknowledged (see www.share-project.org).

9. References

Bates, Douglas, and Dirk Eddelbuettel. 2013. "Fast and Elegant Numerical Linear Algebra Using the RcppEigen Package." *Journal of Statistical Software* 52 (5): 1–24. http://www.jstatsoft.org/v52/i05/.

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.

Börsch-Supan, Axel, Martina Brandt, Christian Hunkler, Thorsten Kneip, Julie Korbmacher, Frederic Malter, Barbara Schaan, Stephanie Stuck, and Sabrina Zuber. 2013. "Data Resource Profile: The Survey of Health, Ageing and Retirement in Europe (Share)." *International Journal of Epidemiology* 42 (4): 992–1001. https://doi.org/10.1093/ije/dyt088.

Eddelbuettel, Dirk, and Romain François. 2011. "Rcpp: Seamless R and C++ Integration." *Journal of Statistical Software* 40 (8): 1–18. https://doi.org/10.18637/jss.v040.i08.

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.

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.

Oksuzyan, Anna, Maciej J. Dańko, Jennifer Caputo, Domantas Jasilionis, and Vladimir M. Shkolnikov. 2019. "Is the Story About Sensitive Women and Stoical Men True? Gender Differences in Health After Adjustment for Reporting Behavior." *Social Science & Medicine* 228: 41–50. https://doi.org/https://doi.org/10.1016/j.socscimed.2019.03.002.

R-Core-Team. 2018. "R: A Language and Environment for Statistical Computing [Internet]. Vienna, Austria. R Version 3.4.4 (2018-03-15)." https://www.r-project.org.

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.