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

Maciej J. Dańko

26 I 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 fit of *hopit* models as well as functions used to analyse heterogenity in reporting behaviour.

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 to fit and analyze ordered response data in the context of heterogeneity in self reporting behavior.

The ordered response data classifies a measure of interest into ordered categories collected during a survey. For example, if the dependent variable were a happiness rating, then a respondent typically answers a question like: "Taking all things together, would you say you are ... ?" and then selects from response options along the lines of:"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, such as "very good", "good", "fair", "bad", and "very bad" (King et al. 2004; Jürges 2007; Rebelo and Pereira 2014). 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 categories like: "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 be used during a survey and to which the methodology described below could be applied to.

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 reporting behavior modeling perspective, one of the main tasks is to compute this continuous estimate of individuals' underlying, latent measure 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 measure 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 measure and threshold coefficients) can be used to calculate the differences in reporting behavior among groups of people having different contextual characteristics realized by calculation of differences between expected and reported ordinal response measures (Jürges 2007).

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 continues latent measure of investigated response variable	Latent health, latent happiness
Latent index	H	Standardized latent measure	Health index, happiness index
Latent variables	—	Variables used to model latent measure	Health variables, happiness variables
Latent terms	X	Terms of design matrix used to modeled 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 latent measure	Cut-points
Threshold variables	—	Variables used to model thresholds	Socio-demographic, cultural, contextual variables
Threshold terms	Y	Terms of design matrix used to model latent measure	
Threshold coefficients	γ, λ	Coefficients corresponding to each threshold term	

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 self-rated health example, the response variable is self-rated health and latent measure h_i can depend on different health conditions and diseases (health variables). These variables 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 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 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 $\alpha_{i,j}$ assuming that thresholds of lower order are never greater than thresholds of higher orders (hierarchical assumption):

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

The thresholds (cut points, α) are modeled by threshold variables coded as design matrix Y, their coefficients γ, and intercepts λ. It is assumed that they model contextual characteristics of the respondent (e.g. country, 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 $y_i = j \;\Leftrightarrow\; \alpha_{j-1,i} \leq h_i < \alpha_{j,i}$ 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 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 heterogeneity

The model estimates are used to determine reporting behavior, i.e., how the continuous latent measure is projected onto the categorical response. Practically, this 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 measure 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 measure 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 H_i. 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. In other words, disability weight reduces H_i by some given amount or percentage (i.e. of every individual is reduced by the same amount if heart attack or other heart problems are present)(Jürges 2007).

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

While the latent index H_i is intend to reflect underlying health, happiness or other status across individuals, the standardized coefficients D_k (e.g. 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 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 (2) reclassification based on estimated thresholds.

In the first method, the classification is based on calculated latent index H_i and is thus adjusted for inter-individual differences in reporting behavior. The Jürges' percentile method is based 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 $\mathbf{1}$ 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 method 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 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 I use *healthsurvey*, which is completely artificial data set simulated using distributions of some major health and socio-demographic characteristics. The distributions and data structure is roughly based on WAVE1 SHARE database (DOIs: 10.6103/SHARE.w1.600) see Börsch-Supan et al. (2013) for technical details. See also acknowledgements.

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

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

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

Generalized ordered probit model can be fitted using the *hopit*() function. The function takes two kinds of formulas: (1) *latent.formula* that models the impact of latent variables on categorical health and (2) *thresh.formula that models 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 best health to the worst health)
# 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 varcov).
##
##                            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  0.43112
## (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).ageclass60-69.1|2      -0.01835    0.03383   -0.54  0.58763
## (G).ageclass60-69.2|3       0.05336    0.04068    1.31  0.18962
## (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      0.23234    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        0.22378    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)

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

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

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

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

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 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 ~ 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 interactions, like 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 interpretation they can be added to the latent or 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 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 ~ 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)
```

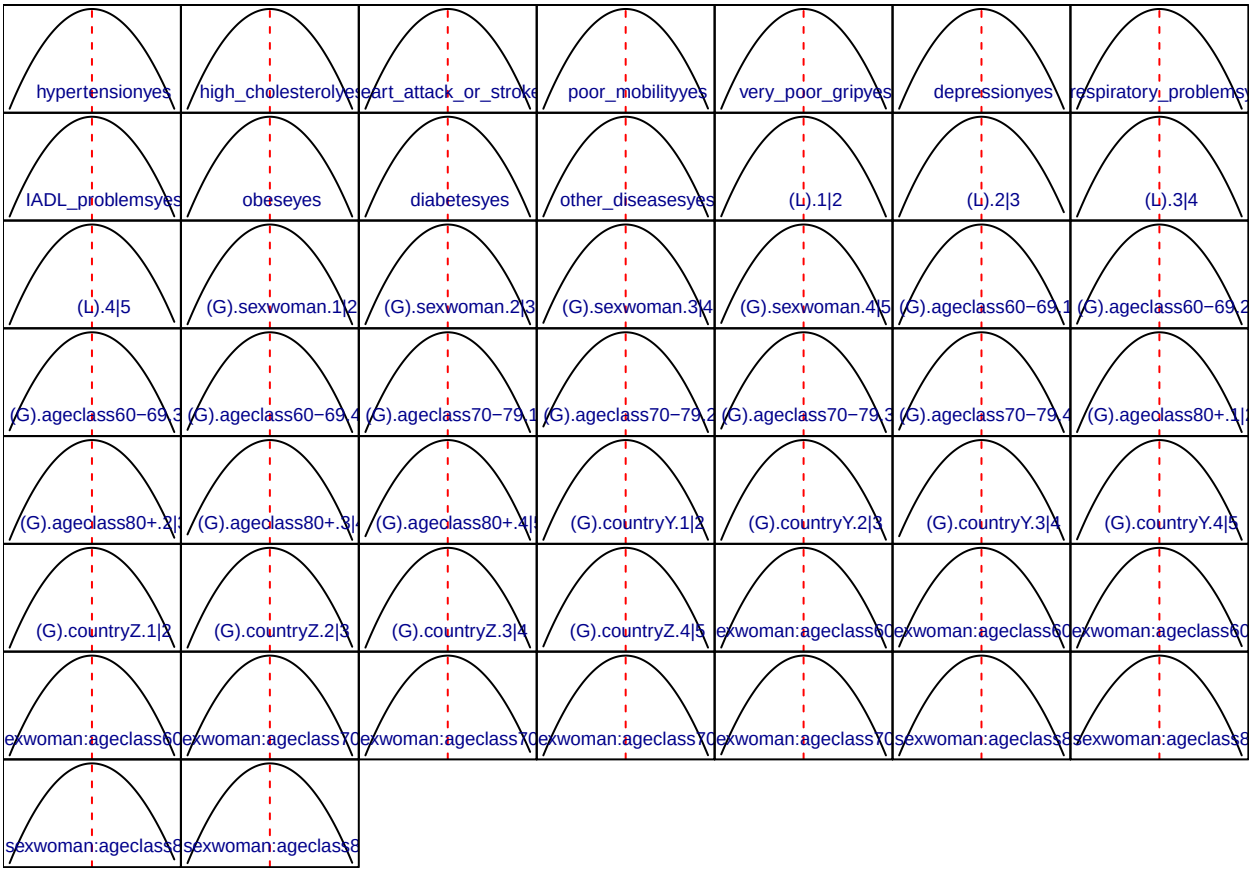
Generally, ignoring survey design could lead to biased results. In the example presented here, it has low importance, which is seen by comparing coefficients of latent variable for both models:

```
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
## hypertensionyes                 0.18475332        0.18777954
## high_cholesterolyes             0.08972562        0.09368829
## heart_attack_or_strokeyes       0.34659838        0.34676964
## poor_mobilityyes                0.70346456        0.70603463
## very_poor_gripyes               0.51424418        0.54768857
## depressionyes                   0.24998274        0.24922296
## respiratory_problemsyes         0.37863461        0.37984845
## IADL_problemsyes                0.59262343        0.60999390
## obeseyes                        0.19041874        0.18874124
## diabetesyes                     0.32839067        0.32328488
## other_diseasesyes               0.32936970        0.32876109
```

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 heterogeneity using *hopit* package

Let's look at latent health variables of *model3*.

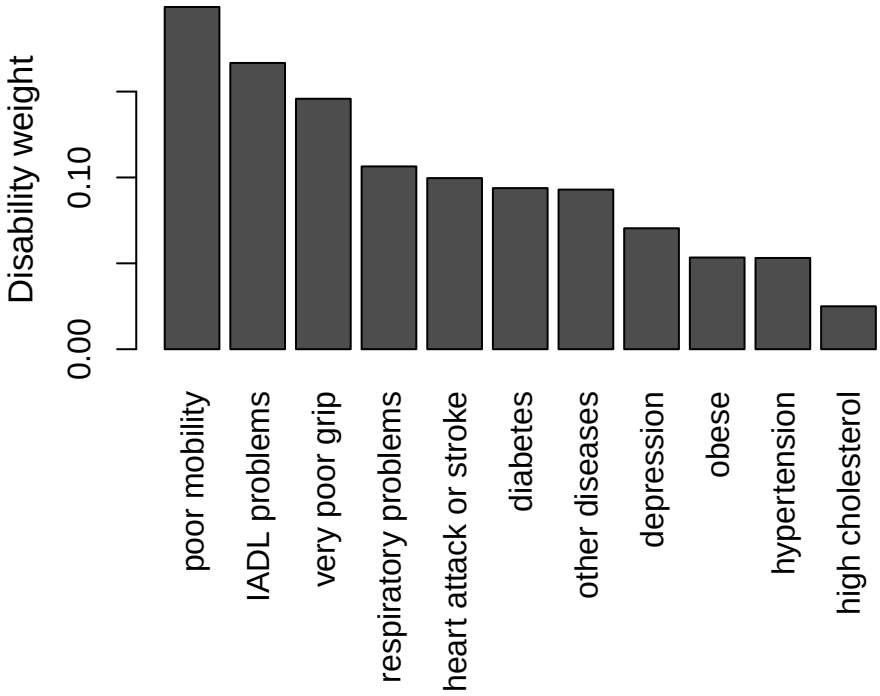
```
model3$coef.ls$latent.params
```

```
##           hypertensionyes       high_cholesterolyes
##                0.18807939                0.08845807
## heart_attack_or_strokeyes          poor_mobilityyes
##                0.35272888                0.70549450
##         very_poor_gripyes             depressionyes
##                0.51631806                0.24936181
##   respiratory_problemsyes          IADL_problemsyes
##                0.37683709                0.59016970
##                 obeseyes               diabetesyes
##                0.18900807                0.33214901
##         other_diseasesyes
##                0.32904701
```

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

```
# 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)
```



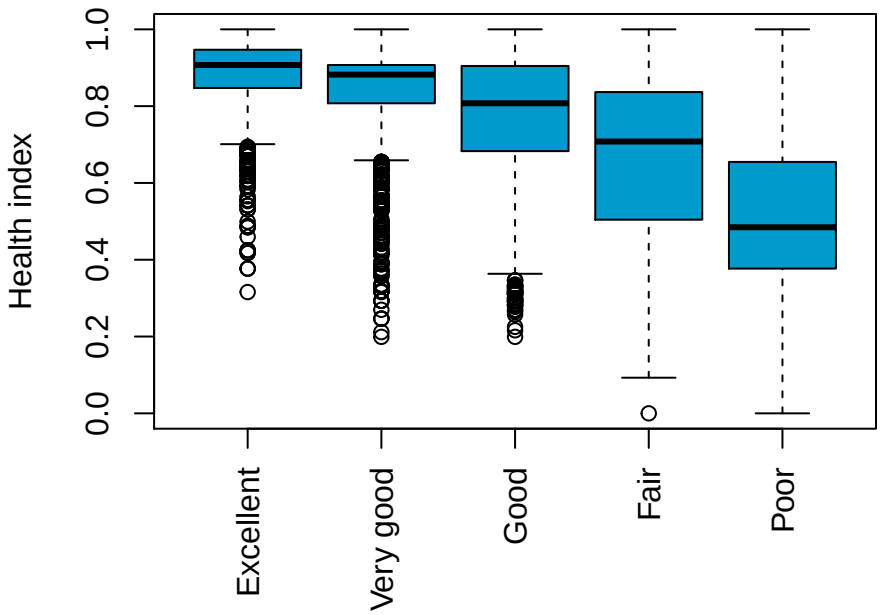

```
sc
```

```
##                              [,1]
## poor mobility          0.19924639
## IADL problems          0.16667625
## very poor grip         0.14581901
## respiratory problems   0.10642667
## heart attack or stroke 0.09961801
## diabetes               0.09380582
## other diseases         0.09292975
## depression             0.07042499
## obese                  0.05337983
## hypertension           0.05311755
## high cholesterol       0.02498241
```

The *namesf* argument is a function or a character vector that is used to rename the coefficient 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 above shows reported health status vs. health index. It is also possible to plot expected categorical health status on Y axis calculated according to the eq. (2).

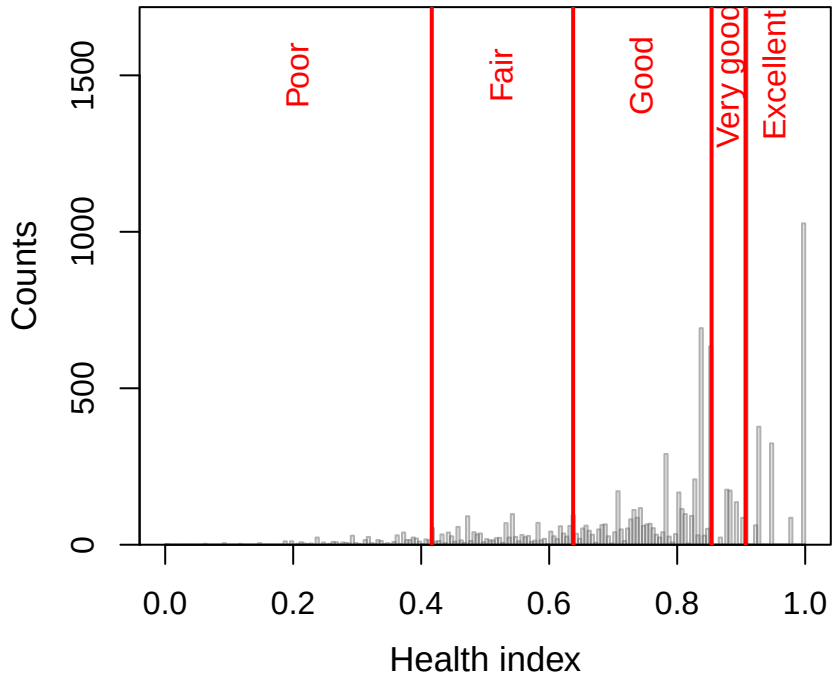
```
hi <- latentIndex(model3, plotf = TRUE, response = "fitted",
                  ylab = 'Health index', col='deepskyblue3')
```

or according to Jürges (2007) method:

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

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

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




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

```
##     4.41%    17.68%    52.34%    77.63%
## 0.4165472 0.6373989 0.8536904 0.9070702
```

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

```
##
## Excellent Very good      Good      Fair      Poor
##      1876      2804      3506      1368       444
```

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

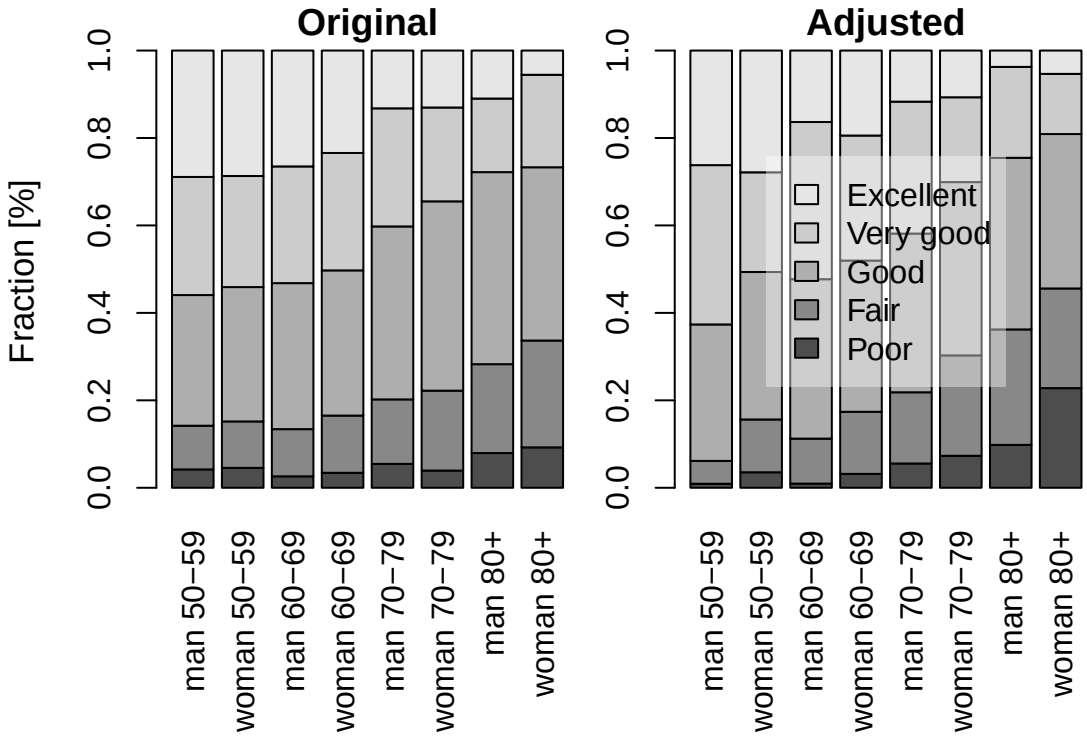
```
##
## Excellent Very good      Good      Fair      Poor
##      2237      2529      3466      1327       441
```

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

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

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

```
# 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 frequencies can be calculated directly using *getLevels* output:

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

```
##
##                 Poor   Fair   Good Very good Excellent
##   man 50-59     3.28   4.77  -1.32     -9.43      2.70
##   woman 50-59   1.01  -1.47  -2.99      2.60      0.85
##   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 to calculate confidence intervals for any measure derived from the model. In the example below, we calculate confidence intervals of the difference between original

and adjusted frequencies of bad health. The latter is determined by the presence of "*Poor*" or "*Fair*" self-rated health categories.

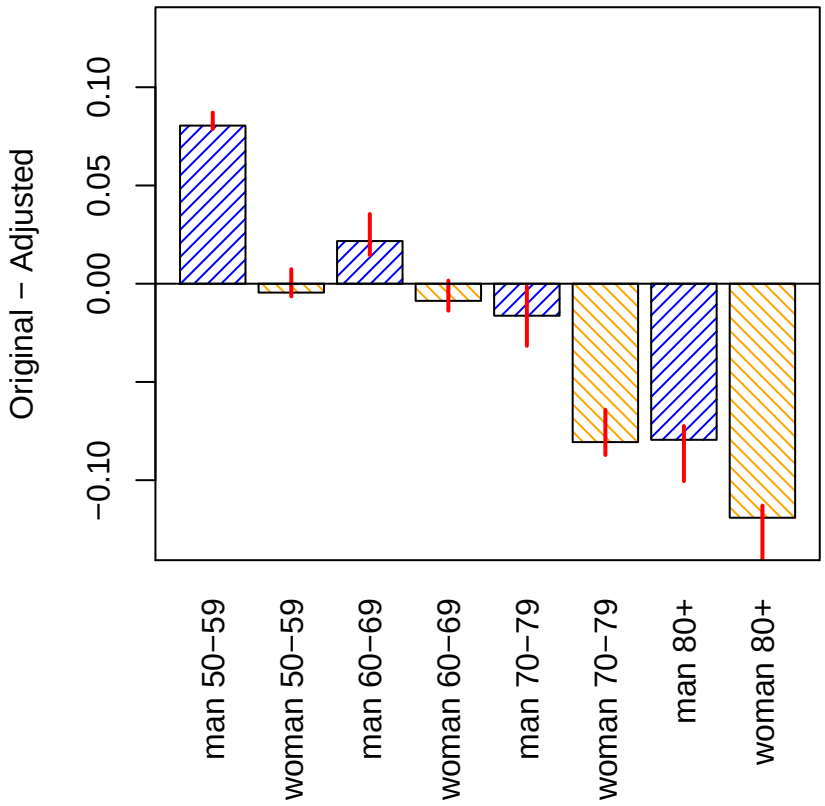
```
# the function to be bootstraped
diff_BadHealth <- function(model, data) {
  hl <- getLevels(model=model, formula=~ sex + ageclass, data = data,
                  sep=' ', plotf=FALSE)
  hl$original[,1] + hl$original[,2] - hl$adjusted[,1]- hl$adjusted[,2]
}

# 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 percentile method
est.CI <- percentile_CI(B)

# plot 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 = '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 at ages [70,80) and both sexes at ages (80, 120] under-report bad health.

8. Acknowledgements

I thank Anna Oksuzyan, Marius Pascariu, Christian Dudel, Oskar Burger, and Laszlo Nemeth for comments on the package and the vignette.

The artificially generated data used in the package examples is based on distributions of some major socjo-demographic and health related characteristics of 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. 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.

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.