

# Adjusted Limited Dependent Variable Mixture Models of Health State Utilities in R

Mark Pletscher

2021-02-23

## 1 Introduction

Health-related quality of life is a key outcome in health technology assessments because it is patient-relevant and it is needed to calculate quality adjusted life years in economic evaluations of health care interventions. Quality of life instruments usually measure health problems in multiple domains using ordinal Likert scales. Value sets or valuation functions are then used to convert these profiles of ordinal measures into cardinal health-related utilities between 1 (perfect health) and minus infinity, where 0 represents death and negative values represent health states worse than death. Because perfect health is defined as 100% quality of life, health state utilities are limited at 1. The lowest possible utility in a local value set or valuation function further defines a lower limit of health state utilities in a local population. Thus, health state utilities are limited dependent variables. In addition, health state utilities often show gaps between 1 and the next smaller utility in a value set or valuation function. These gaps occur particularly often in quality of life instruments with few levels in the ordinal measures of health problems such as the EQ-5D-3L (Mulhern et al. 2018). A last but important particularity of health state utilities is that they can be the consequence of multiple latent classes, or they can exhibit multi-modal marginal densities (Hernández Alava et al. 2014).

Adjusted limited dependent variable mixture models are finite mixtures of multiple normal distributions that account for multi-modality or latent classes, limits and gaps between 1 and the next smaller utility value (Hernández Alava, Wailoo, and Ara 2012; Hernández Alava et al. 2013, 2014; Hernández Alava and Wailoo 2015; Mukuria et al. 2019). These features can improve empirical fit, parameter identification and predictive accuracy compared to standard regression models which makes adjusted limited dependent variable mixture models particularly useful for mapping studies (Gray, Wailoo, and Alava 2018; Gray, Alava, and Wailoo 2018; Dixon, Hollingworth, and Sparrow 2020; Yang et al. 2019; Xu et al. 2020; Fuller et al. 2017; Pennington et al. 2020) and the identification of incremental effects and average marginal effects of medical interventions or health problems (e.g. Hvidberg (2016)).

The R package ‘aldvmm’ is an implementation of the adjusted limited dependent variable model mixture model proposed by Hernández Alava and Wailoo (2015) using normal component distributions and a multinomial logit model of probabilities of component membership.

The objective of this vignette is to demonstrate the usage of the ‘aldvmm’ package, show important challenges of fitting adjusted limited dependent variable mixture models and to compare its results to the published results from the STATA<sup>®</sup> package (Hernández Alava and Wailoo 2015) using publicly available data.

## 2 Methods

Adjusted limited dependent variable mixture models are finite mixtures of normal distributions in  $K$  components  $c$  with conditional expectations  $E[y|X, c] = X\beta^c$  and standard deviations  $\sigma^c$ . The probabilities of component membership are estimated using a multinomial logit model as  $P[c|X] = \exp(X\delta^c) / \sum_{k=1}^K \exp(X\delta^k)$ . The model accumulates the density mass of the finite mixture below a minimum value  $\Psi_1$  at the value  $\Psi_1$ ,

and the density mass above a maximum value  $\Psi_2$  at 1. If the maximum value  $\Psi_2$  is smaller than 1, the model emulates a value set with a gap between 1 and the next smaller value.

$$y_i|c = \begin{cases} 1 & \text{if } y_i|c > \Psi_2 \\ \Psi_1 & \text{if } y_i|c \leq \Psi_1 \\ y_i|c & \text{if } \Psi_1 < y_i|c \leq \Psi_2 \end{cases} \quad (1)$$

In this vignette, we estimate the same models of post-operative EQ-5D-3L utilities as Hernández Alava and Wailoo (2015) and include post-operative Oxford Hip Scores (divided by 10) as the only explanatory variable  $x$ .

$$\begin{aligned} \text{Model 1: } E[y|c, X] &= \beta_0^c + \beta_1^c x \\ P[c|X] &= \text{mlogit}(\delta_0^c) \\ \text{Model 2: } E[y|c, X] &= \beta_0^c + \beta_1^c x \\ P[c|X] &= \text{mlogit}(\delta_0^c + \delta_1^c x) \end{aligned} \quad (2)$$

The function `aldvmm::aldvmm()` fits an adjusted limited dependent variable mixture model using the likelihood function from Hernández Alava and Wailoo (2015). The function calls `optimr::optimr` to minimize the negative log-likelihood function using numerical gradients from `numDeriv::grad()`. The function `aldvmm::aldvmm()` accepts all optimization methods available in `optimr::optimr()` except for “nlm,” which requires a different implementation of the likelihood function.

The model formula in `aldvmm::aldvmm()` is a object of class “formula” with two parts on the right-hand side. The first part on the left of the `|` delimiter represents the model of expected values of  $K$  normal distributions in components  $c$  with standard deviations  $\sigma^c$ . The second part on the right of the `|` delimiter represents the model of probabilities of component membership from a multinomial logit model.

The function `aldvmm::aldvmm()` provides four options for the generation of starting values for the optimization algorithm.

1. “zero”: A vector of zeroes (default).
2. “random”: A vector of standard normal random values.
3. “constant”: Parameter estimates of a constant-only model as starting values for intercepts and standard deviations, and zeroes for all other parameters.<sup>1</sup>
4. “sann”: Parameter estimates of a simulated annealing algorithm.

The package ‘aldvmm’ obtains fitted values using the expected value function from Hernández Alava and Wailoo (2015). Covariance matrices and standard errors of parameters are obtained from a numerical approximation of the hessian matrix using `numDeriv::hessian()`. Standard errors of fitted values in the estimation data  $SE_{fit}$  and standard errors of predicted values in new data  $SE_{pred}$  are calculated using the delta method (Dowd, Greene, and Norton 2014; Whitmore 1986).  $G$  denotes the gradient of fitted values with respect to changes in parameter estimates,  $\Sigma$  denotes the covariance matrix of parameters, and  $MSE$  denotes the mean squared error of fitted versus observed values in the estimation data.

$$SE_{fit} = \sqrt{G'\Sigma G} \quad (3)$$

$$SE_{pred} = \sqrt{MSE + G'\Sigma G} \quad (4)$$

The function `aldvmm::aldvmm()` returns an object of S3 class “aldvmm” for which methods for generic functions `summary()` and `predict()` are available.

---

<sup>1</sup>The auxiliary models for obtaining starting values are fitted using zero starting values.

## 3 Installation

The package ‘aldvmm’ can be installed from github using the function `install_github()` from the ‘devtools’ package.

```
# install.packages("devtools")
devtools::install_github("pletschm/aldvmm", ref = "main")
```

## 4 Data

We analyze the same publicly available EQ-5D-3L utility data from English patients after hip replacement in 2011 and 2012 (NHS Digital 2013) as Hernández Alava and Wailoo (2015) in their description of the STATA® ALDVMM package.

```
temp <- tempfile()

url <- paste0("https://files.digital.nhs.uk/publicationimport/pub11xxx/",
             "pub11359/final-proms-eng-apr11-mar12-data-pack-csv.zip")

download.file(url, temp)

df <- read.table(unz(description = temp,
                    filename = 'Hip Replacement 1112.csv'),
               sep = ',',
               header = TRUE)

unlink(temp)
rm(temp)

df <- df[, c("AGEBAND", "SEX", "eq5d", "HR_Q2_SCORE")]
df <- df[df$AGEBAND!='*' & df$SEX!='*', ]
d[, "hr"] <- d[, "HR_Q2_SCORE"]/10
df <- df[stats::complete.cases(df), ]

set.seed(101010101)
df <- df[sample(1:nrow(df), size = nrow(df)*0.3), ]
```

The data includes 35’166 observations with complete information on patients’ post-operative utilities, Oxford Hip Scores, age and sex. Like Hernández Alava and Wailoo (2015), we draw a 30% sub-sample of 10’549 observations from the population of complete observations of these variables. Although we follow the same approach as Hernández Alava and Wailoo (2015) in the preparation of the data, our random sample is not identical to the data used in their study. Post-operative EQ-5D-3L utilities from English value sets (Dolan 1997) show a bimodal distribution, limits at -0.594 and 1 and a gap between 1 and 0.883 (figure 1).

## 5 Results

### 5.1 Model 1

#### 5.1.1 BFGS optimization method with zero-only starting values

We first fit model 1 with the “BFGS” optimization method and “zero” initial values. The values in the argument ‘psi’ represent the maximum and minimum values smaller than 1 in the English value set (Dolan 1997). As the data showed a bi-modal distribution (figure 1), we estimate a mixture of 2 normal distributions (‘ncmp’ = 2). `aldvmm::aldvmm()` returns an object of class “aldvmm.”

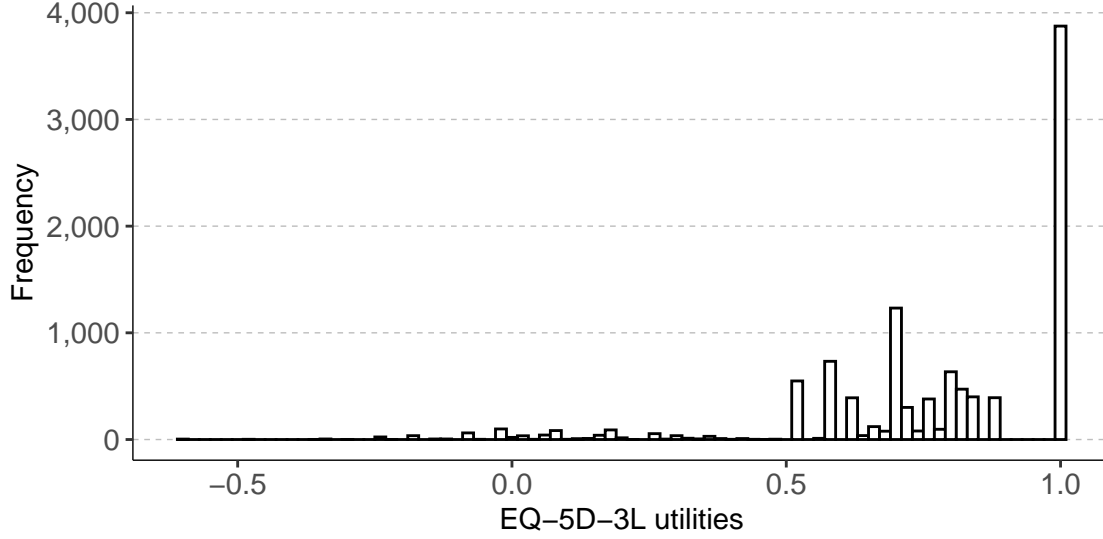


Figure 1: Frequency distribtution of observed EQ-5D-3L utilities

```
library("aldevmm")

fit <- aldevmm::aldevmm(data = df,
  formula = eq5d ~ hr | 1,
  psi = c(0.883, -0.594),
  ncmp = 2,
  init.method = "zero",
  optim.method = "BFGS")

summary(fit)

pred <- predict(fit,
  newdata = df,
  se.fit = TRUE,
  type = "fit")
```

We obtain a summary table of regression results using the generic function `summary()`. The model converges at a log-likelihood of 706.32 and an Akaike information criterion value of -1'398.65 (table 1).

The coefficients of the intercept and covariates for the expected values  $E[y|c, X]$  of the normal distributions can be interpreted as marginal effects on component means. 'lnsigma' denotes the natural logarithm of the estimated standard deviation  $\sigma^c$ . The coefficients of covariates in the multinomial logit model of probabilities of component membership are log-transformed relative probabilities. Our model only includes two components, and the multinomial logit model collapses to a binomial logit model. The intercept of 0.7283 means that the average probability of an observation in the data to belong to the first component is  $\exp(0.7283)$  or 2.071556 times the probability to belong to component 2.

We obtain expected values of observations in the estimation data using the generic function `predict()`. Standard errors of fitted (estimation data) or predicted (new data) values are calculated using the delta method. Expected values exhibit a smoother distribution than observed values and do not show a gap between 1 and 0.883, because they are averages of 1 and the expected values of the finite mixture weighted by the density mass above and below 0.883.

Table 1: Regression results from model1 with "BFGS" optimization method and "zero" starting values

		Estimate	Std. Err.	z	P> z	[95% Conf. Interval]
E[y c, X]						
Comp1	(Intercept)	0.2358	0.0069	34.1794	0.0000	0.2223 0.2493
	hr	0.1459	0.0019	76.3624	0.0000	0.1422 0.1496
	lnsigma	-2.4623	0.0178	-138.1313	1.0000	-2.4972 -2.4273
Comp2	(Intercept)	-0.4306	0.0224	-19.2308	1.0000	-0.4745 -0.3868
	hr	0.3135	0.0065	47.9417	0.0000	0.3007 0.3263
	lnsigma	-1.2480	0.0215	-57.9891	1.0000	-1.2902 -1.2058
P[c X]						
Comp1	(Intercept)	0.7283	0.0607	12.0056	0.0000	0.6094 0.8472
N = 10549 ll = 706.32 AIC = -1398.65 BIC = -1398.65						

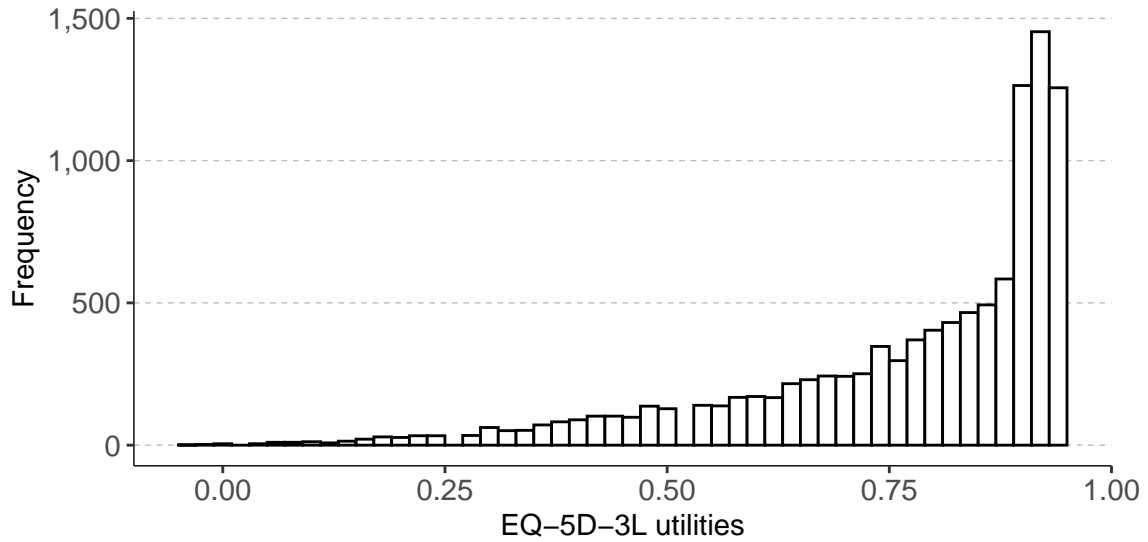


Figure 2: Expected values from base case model

### 5.1.2 Comparison of optimization methods

```

init.method <- c("zero", "random", "constant", "sann")

optim.method <- c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin",
                  "Rvmmin", "hjn")

fit1_all <- list()

for (i in init.method) {
  for (j in optim.method) {
    set.seed(101010101) # Seed for random starting values
    fit1_all[[i]][[j]] <- aldvmm::aldvmm(data      = df,
                                         formula   = formula,
                                         psi       = c(0.883, -0.594),
                                         ncmp      = 2,
                                         init.method = i,
                                         optim.method = j)
  }
}

```

```
}
```

Hernández Alava and Wailoo (2015) suggested that the likelihood function of the adjusted limited dependent variable mixture model with the English EQ-5D-3L data might have multiple local optima, and that the estimation is sensitive to initial values. We thus fit model 1 with all optimization algorithms and methods for generating initial values available in `aldvmm::aldvmm()` to assess the sensitivity of the model to optimization settings and to find the maximum likelihood estimates.

The maximum likelihood varies considerably across optimization methods and initial values which confirms the sensitivity of the model to initial values and optimization algorithms (table 2). The most frequent log-likelihood is 706.32, but the Hooke and Jeeves Pattern Search Optimization (“hjn”) with “zero” initial values converges at a log-likelihood 33’057.43.

The optimization methods “Nelder-Mead,” “BFGS,” “CG,” and “L-BFGS-B” are particularly sensitive to starting values. The methods “Rvmmin” converge at a log-likelihood of 706.32 with three of four sets of initial values, and the method “hjn” with two of four sets of initial values. The methods “nlminb” and “Rcgmin” converge at a log-likelihood of 706.32 regardless of the sets of initial values.

Table 2: Log-likelihood by optimization method

	Nelder-Mead	BFGS	CG	L-BFGS-B	nlminb	Rcgmin	Rvmmin	hjn
zero	-259.21	706.32	223.96	-4513.33	706.32	706.32	-8393.14	33057.43
random	-434.21	-623.98	65.03	-2830.18	706.32	706.32	-11074.58	-627.78
constant	354.36	706.32	-601.52	-634.81	706.32	706.32	-10867.67	706.32
sann	706.19	706.32	576.51	706.32	706.32	706.32	706.32	706.32

The computation times of optimization routines vary considerably across methods (table 3). The optimization methods “Nelder-Mead,” “BFGS,” “L-BFGS-B” and “Rvmmin” are the fastest methods, but this speed comes at the cost of a higher risk of convergence at local optima. Naturally, the generation of initial values using simulated annealing (“sann”) is the slowest method for generating initial values which results in long overall computation times of `aldvmm::aldvmm()`. The Hooke and Jeeves Pattern Search Optimization (“hjn”) with “zero” starting values that converges at the largest log-likelihood is the slowest approach with a computation time of 15.83 minutes.

Table 3: Estimation time [minutes] by optimization method

	Nelder-Mead	BFGS	CG	L-BFGS-B	nlminb	Rcgmin	Rvmmin	hjn
zero	0.13	0.30	0.98	0.09	0.52	3.87	0.04	14.98
random	0.12	0.37	1.02	0.18	0.51	3.89	0.04	0.41
constant	0.19	0.45	1.20	0.29	0.77	2.80	0.05	0.92
sann	1.79	1.88	2.68	2.85	2.19	5.04	1.92	3.76

As a next step, we compare the parameter estimates from the “Nelder-Mead,” “BFGS” and “hjn” algorithms with “zero” initial values. The “Nelder-Mead” and “BFGS” estimates differ clearly in terms of the split between the intercept and the coefficient of the Oxford Hip Score, the probabilities of component membership and the standard deviations (table 4). The solution of the “hjn” method is rather extreme with no effect of the Oxford Hip Score in component 1, a standard deviation of almost 0 in component 1 and a very low probability of membership of component 1.

To get a better understanding of the differences between the results of the “Nelder-Mead,” “BFGS” and “hjn” algorithms we plot the densities of each component weighted by the probability of component membership.

```
nsim <- 100
hr <- 3.825244 # Population average Oxford Hip Score

# Nelder-Mead parameter estimates
```

Table 4: Regression results of model1 with zero starting values in "Nelder-Mead", "BFGS" and "hjn" algorithms

		Nelder-Mead	BFGS	hjn
E[y c, X]				
Comp1	(Intercept)	-0.0575	0.2358	0.6910
	hr	0.2233	0.1459	0.0000
	lnsigma	-1.8381	-2.4623	-36.7370
Comp2	(Intercept)	4.4022	-0.4306	-0.1490
	hr	-0.9974	0.3135	0.2500
	lnsigma	0.1250	-1.2480	-1.5890
P[c X]				
Comp1	(Intercept)	3.8489	0.7283	-2.1900
N = 10549   ll = -259.21   AIC = 532.42   AIC = -1398.65   AIC = -66100.87				

```

n1    <- nsim*exp(3.8489)/(1 + exp(3.8489))
mean1 <- -0.0575 + 0.2233 * hr
sd1   <- exp(-1.8381)
n2    <- nsim*(1 - exp(3.8489)/(1 + exp(3.8489)))
mean2 <- 4.4022 + -0.9974 * hr
sd2   <- exp(0.1250)

# Make plot
ggplot2::ggplot(data = data.frame(x = c(-1, 1)), aes(x)) +
  ggplot2::stat_function(fun = dnorm,
                        n = n1,
                        args = list(mean = mean1, sd = sd1)) +
  ggplot2::stat_function(fun = dnorm,
                        n = n2,
                        args = list(mean = mean2, sd = sd2))

```

The densities in the solution of the “Nelder-Mead” method resemble the bi-model distribution observed in the data (figure 3). The densities in the solution of the “BFGS” method include two distributions with similar means but different standard deviations (figure 4). The densities in the solution of the “hjn” method include two distributions with similar means, but the distribution in component 1 shows low density mass due to the low probability of membership to component 1 (figure 5). The density plots also suggest that the model fit benefits more from improving the modeling of the distribution at the more frequently observed higher utilities rather than replicating the bi-modal distribution observed in the data. Overall, the differences between optimization methods show that it is very difficult to fit a two-component model to the data. We suspect that a simple one-component model would be more likely to converge towards a global optimum and would fit the data similarly well as the two-component model.

The differences in parameter estimates from different optimization methods show that the choice of the optimization algorithm and initial values is very important for parameter identification. As adjusted limited dependent variable mixture models are frequently used for tasks that rely on predictions, we also compare expected values from the “Nelder-Mead” and “BFGS” methods to the expected values from the “hjn” method. Expected values from the “Nelder-Mead” and “BFGS” methods differed from expected values from the “hjn” method in particular for observations with lower expected values (figure 6). Interestingly, the expected values from the “Nelder-Mead” method deviate less from the expected values from the “hjn” method than the expected values from the “BFGS” method although the “BFGS” method yielded a higher log-likelihood in the optimum, and the “BFGS” results are mixtures of densities with higher means similar to the solution of the “hjn” algorithm.

An visual inspection of mean residuals over deciles of expected values shows that model 1 fits the data poorly, and that the patterns of over- and under-predictions of observed values are similar across optimization

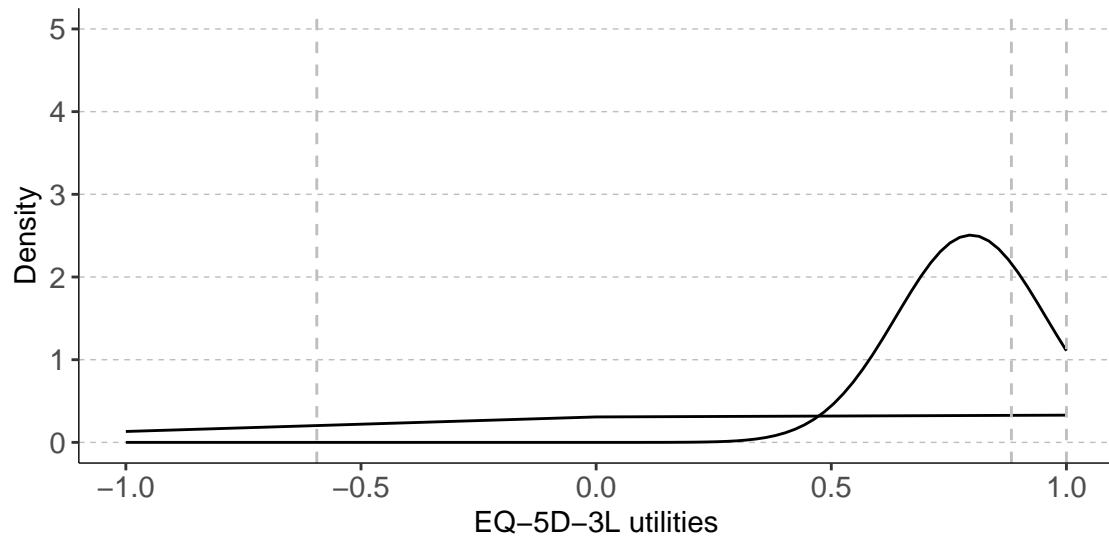


Figure 3: Densities in components based on “Nelder-Mead” parameter estimates (observation with population average Oxford Hip Score 3.8489)

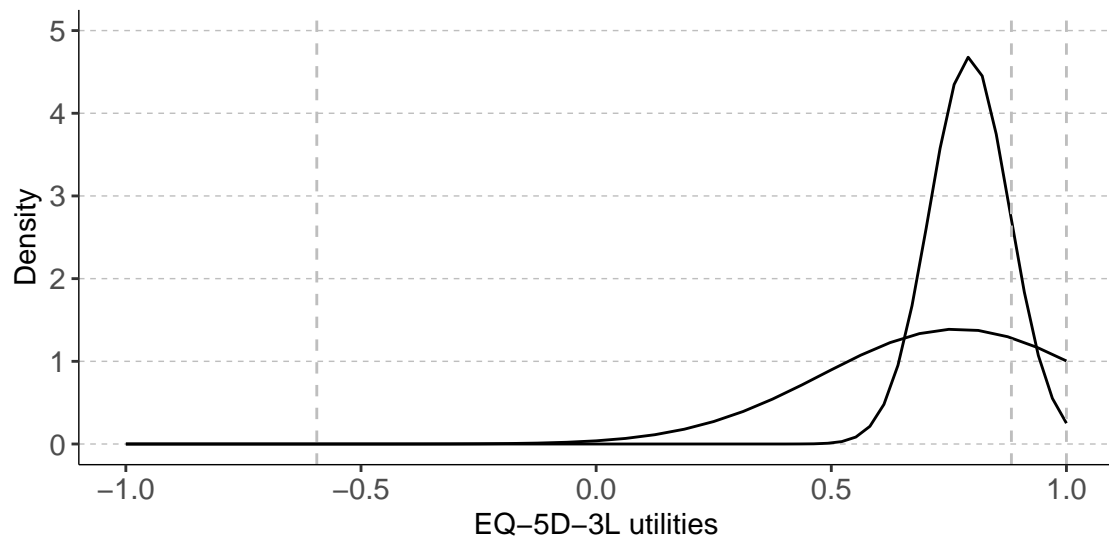


Figure 4: Densities in components based on “BFGS” parameter estimates (observation with population average Oxford Hip Score 3.8489)



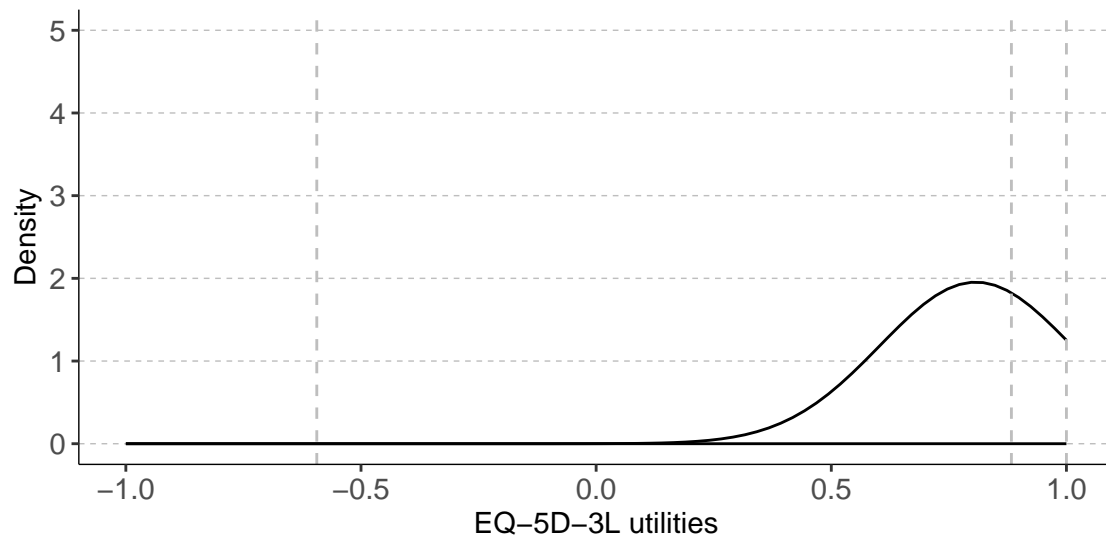


Figure 5: Densities in components based on “hjn” parameter estimates (observation with population average Oxford Hip Score 3.8489)

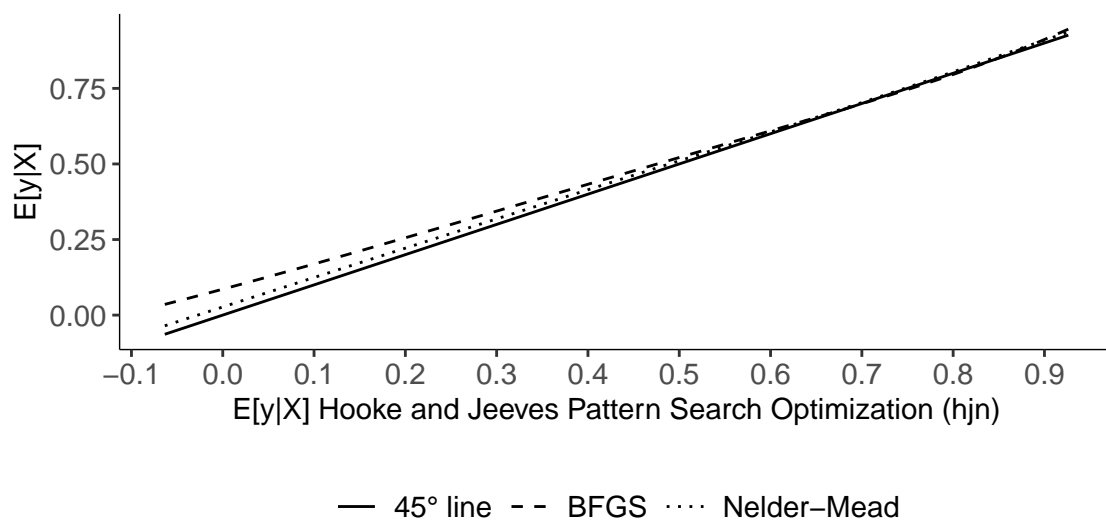


Figure 6: Expected values from model 1, “Nelder-Mead” and “BFGS” versus “hjn” with zero starting values

methods with different log-likelihoods (figure 7, figure 8 and figure 9 in the appendix).

### 5.1.3 Constrained optimization with user-defined initial values

We can also fit model 1 with user-defined starting values and box constraints. When constraints are imposed the function `aldvmm::aldvmm()` uses the optimization method “L-BFGS-B.” We set the initial value of the intercept of the probability of belonging to component 1 to 0.7283 as in the solution of the “BFGS” optimization method with “zero” starting values (table 4). We impose a lower limit of -3 to the log-standard deviations in both components. The function `aldvmm::aldvmm()` returns a warning that the covariance matrix included negative values on the diagonal. We see that these values are the variances of the intercept and the log-standard deviation in component 2 (table 5). The log-likelihood amounts to -627.78, which is lower than the log-likelihood in the solution of the “Nelder-Mead” optimization method with “zero” starting values. The parameter estimates do not resemble any of the solutions of the unconstrained “Nelder-Mead,” “BFGS” or “hjn” optimization methods with “zero” values (table 4), which further emphasizes the difficulties in finding a global optimum of the likelihood with English EQ-5D-3L utilities after hip replacement.

```
init <- c(0, 0, 0, 0, 0, 0, 0.7283)
lo <- c(-Inf, -Inf, -3, -Inf, -Inf, -3, -Inf)
hi <- c(Inf, Inf, Inf, Inf, Inf, Inf, Inf)

fit1_ctr <- aldvmm::aldvmm(data = df,
                           formula = formula,
                           psi = c(0.883, -0.594),
                           ncmp = 2,
                           init.est = init,
                           init.lo = lo,
                           init.hi = hi)

summary(fit1_ctr)
```

Table 5: Regression results of model1 with "hjn" optimization method and constrained user-defined starting values

		Estimate	Std. Err.	z	P> z	[95% Conf. Interval]
E[y c, X]						
Comp1	(Intercept)	-0.0920	0.0085	-10.8048	1.0000	-0.1087 -0.0753
	hr	0.2325	0.0023	101.8493	0.0000	0.2280 0.2370
	lnsigma	-1.6406	0.0089	-183.9809	1.0000	-1.6581 -1.6231
Comp2	(Intercept)	0.3931	NaN	NaN	NaN	NaN NaN
	hr	2.5249	7.4794	0.3376	0.3678	-12.1345 17.1843
	lnsigma	-0.7651	NaN	NaN	NaN	NaN NaN
P[c X]						
Comp1	(Intercept)	6.8520	0.5388	12.7168	0.0000	5.7960 7.9081
N = 10549 ll = -627.78 AIC = 1269.55 BIC = 1269.55						

### 5.1.4 Single-component model

As the solution of the “hjn” algorithm included a component with very low probability, we also estimate a single-component model.

```
fit <- aldvmm::aldvmm(data = df,
                      formula = eq5d ~ hr | 1,
                      psi = c(0.883, -0.594),
                      ncmp = 1,
                      init.method = "zero",
```

```
optim.method = "hjn")
```

```
summary(fit)
```

The coefficients of the single-component model are relatively similar to the solution of the “hjn” algorithm for the two-component model (table 6). The Akaike information criterion amounts to 1275.61 which is larger than the values of the “BFGS” (-1398.65) and “hjn” (-66100.87) solutions of the two-component model and thus suggests worse fit.

Table 6: Regression results of model1 with 1 component, zero starting values in "hjn" algorithm

		Estimate	Std. Err.	z	P> z	[95% Conf. Interval]
E[y c, X]						
Comp1	(Intercept)	-0.0886	0.0085	-10.4100	1.0000	-0.1053 -0.0719
	hr	0.2318	0.0023	101.4681	0.0000	0.2273 0.2362
	lnsigma	-1.6363	0.0089	-184.2421	1.0000	-1.6537 -1.6189
N = 10549 ll = -634.81 AIC = 1275.61 BIC = 1275.61						

## 5.2 Model 2

Because model 1 is difficult to fit and shows poor fit to the data, we also estimate model 2 with a coefficient of the Oxford Hip score in the multinomial logit model of component membership. For this fit, we use the method “hjn” with estimates from Hernández Alava and Wailoo (2015) as starting values.

```
init <- c(-.40293118, .30502755, .22614716, .14801581, -.70755741, 0,
         -1.2632051, -2.4541401)
```

```
fit2 <- aldvmm::aldvmm(data = df,
                      formula = eq5d ~ hr | hr,
                      psi = c(0.883, -0.594),
                      ncmp = 2,
                      init.est = init,
                      optim.method = "hjn")
```

```
summary(fit2)
```

The log-likelihood of model 2 in the “hjn” optimum is larger than the log-likelihood of model 1 in the “hjn” optimum due to the extra parameter (table 7). However, the Akaike information criterion of model 2 (-1866.97) is larger than the Akaike information criterion of model 1 (-66100.87), which suggests that the increase in the log-likelihood does not justify the extra parameter in the multinomial logit part.<sup>2</sup>

## 5.3 Comparison to results by Hernández Alava and Wailoo (2015)

To validate the R implementation of adjusted limited dependent variable models we compare the results of three models to the results from the STATA<sup>®</sup> ALDVMM package published in Hernández Alava and Wailoo (2015).

- Model 1 with default optimization settings.
- Model 1 initial values from constant-only model.
- Model 2 with user-defined initial values.

Because we draw a random sub-sample of the data, we do not expect identical results, but similar findings would be expected if the model is implemented in a similar manner as in STATA<sup>®</sup>.

<sup>2</sup>In the aldvmm::aldvmm() output, smaller values of the Akaike information criterion indicate better goodness of fit.

Table 7: Regression results of model 2 with user-defined starting values in the "hjn" algorithm

		Estimate	Std. Err.	z	P> z	[95% Conf. Interval]
E[y c, X]						
Comp1	(Intercept)	0.0041	0.0281	0.1444	0.4426	-0.0511 0.0592
	hr	0.0962	0.0115	8.3296	0.0000	0.0735 0.1188
	lnsigma	-1.2674	0.0313	-40.4818	1.0000	-1.3288 -1.2061
Comp2	(Intercept)	0.1817	0.0074	24.5211	0.0000	0.1672 0.1962
	hr	0.1609	0.0019	85.4232	0.0000	0.1572 0.1646
	lnsigma	-2.2810	0.0126	-181.2022	1.0000	-2.3057 -2.2563
P[c X]						
Comp1	(Intercept)	2.4527	0.1720	14.2598	0.0000	2.1155 2.7898
	hr	-1.3925	0.0566	-24.5956	1.0000	-1.5034 -1.2815

N = 10549 ll = 941.36 AIC = -1866.73 BIC = -1866.73

We use the results of the “BFGS” method with “zero” starting values as default estimates of model 1, as this solution is found most frequently by different optimization methods (table 2), and the results of the “hjn” method with “zero” starting values are rather extreme and difficult to interpret.

```
# Model 1 with default optimization settings
fit1_default <- aldvmm::aldvmm(data = df,
                              formula = eq5d ~ hr | 1,
                              psi = c(0.883, -0.594),
                              ncmp = 2,
                              init.method = "zero",
                              optim.method = "BFGS")

# Model 1 initial values from constant-only model
fit1_const <- aldvmm::aldvmm(data = df,
                              formula = eq5d ~ hr | 1,
                              psi = c(0.883, -0.594),
                              ncmp = 2,
                              init.method = "constant",
                              optim.method = "hjn")

# Model 2 with user-defined initial values.
init <- c(-.40293118, .30502755, .22614716, .14801581, -.70755741, 0,
          -1.2632051, -2.4541401)

fit2 <- aldvmm::aldvmm(data = df,
                       formula = eq5d ~ hr | hr,
                       psi = c(0.883, -0.594),
                       ncmp = 2,
                       init.est = init,
                       optim.method = "hjn")
```

The results of model 1 from the “BFGS” method with “zero” starting values differed considerably from the default estimates reported by Hernández Alava and Wailoo (2015) (table 8). However, the default estimates by Hernández Alava and Wailoo (2015) include extreme estimates without standard errors in component 2 (table 10 in the appendix). Our estimates of parameters, standard errors and the log-likelihood of model 1 with initial values from a constant-only model are very similar to the results from Hernández Alava and Wailoo (2015) (tables 12 and 11 in the appendix). The parameter estimates of model 2 with user-defined starting values are somewhat similar, and the standard errors of parameters with similar point estimates are fairly consistent (tables 14 and 13 in the appendix). Differences in the analyzed sub-samples could be a

possible reason for the observed discrepancies. We do not deem these findings indications of errors in the ‘aldvmm’ implementation of adjusted limited dependent variable mixture models.

Table 8: Comparison of point estimates of parameters in R and STATA extsuperscript extregistered

		default		constant		model 2	
		R	STATA	R	STATA	R	STATA
E[y c, X]							
Comp1	(Intercept)	0.2358	-0.0883	-0.4307	-0.4029312	-0.1192	0.0129138
	hr	0.1459	0.2308	0.3135	0.3050275	0.0802	0.0887522
	lnsigma	-2.4623	-1.6461	-1.2481	-1.263205	-1.8720	-1.319285
Comp2	(Intercept)	-0.4306	3930832	0.2358	0.2261472	0.2010	0.1763049
	hr	0.3135	885110.9	0.1459	0.1480158	0.1562	0.1619008
	lnsigma	-1.2480	-164.8169	-2.4624	-2.45414	-2.2133	-2.271434
P[c X]							
Comp1	(Intercept)	0.7283	7.3206	-0.7280	-0.7075574	1.7744	2.431909
	hr					-1.3526	-1.395115
N = 10549		ll = 706.32	ll = -577.38	ll = 706.32	ll = 685.79	ll = 939.22	ll = 942.71

## 6 Discussion

Adjusted limited dependent variable mixture models are powerful tools for regression analysis of health state utility data. Unlike standard regression models, adjusted limited dependent variable mixture models account for limits, gaps and multi-modal distributions.

The comparison of different optimization methods in EQ-5D-3L utility data from English patients after hip replacement in 2011 and 2012 (NHS Digital 2013) shows that the likelihood function can be challenging to maximize and can converge at extreme solutions. The package ‘aldvmm’ offers a variety of optimization algorithms and methods for generating initial values which is an important strength in such challenging situations. It is essential to assess different optimization algorithms and methods for initial values before interpreting the parameter estimates or predictions of adjusted limited dependent variable mixture models.

The analysis of the EQ-5D-3L utility data also suggests that simpler models with fewer components should be considered when multi-component models are difficult to fit. Even in single-component models, the adjusted limited dependent variable mixture model accounts for limits and gaps which can improve fit compared to traditional regression techniques.

Although coefficients of expected values can be interpreted as marginal effects within each component, they cannot be interpreted in terms of overall expected values. Thus, average marginal effects and average treatment effects need to be calculated from predictions using the generic function `predict()`. Standard errors of marginal effects or average treatment effects can be calculated using the standard errors of fitted values for observed and counterfactual covariate values.

In situations with repeated utility measures, the package ‘aldvmm’ only allows fixed effect estimations with individual/group and time fixed effects. This might be an important limitation in the analysis of clinical trial data when the statistical analysis plan of the clinical study specifies models with individual/group and time random effects. However, causal inference on treatment effects with repeated measures requires an adjustment of treatment effects for general time trends when there is dynamic selection in the population, e.g. because treated individuals survive longer and thus are over-represented in later measurements. The researcher who aims to make causal inference using repeated measures faces a trade-off between the efficiency of the random effects model and the causal interpretation of treatment effects in a fixed-effects model. It is recommended to assess the uncorrelatedness of random effects with the treatment using the Hausman test in a generalized linear model before making an informed modeling decision.

Mixed effects implementations of ‘aldvmm’ would be an important improvement over the current implementation. However, a mixed effects implementation would require other optimization techniques than the

package ‘aldvmm.’ A Bayesian implementation of the likelihood from Hernández Alava and Wailoo (2015) would be a pragmatic and powerful approach to introduce random effects to any parameter in the model. Obviously, Bayesian models would also provide relevant information about the full posterior distributions, but the definition of informative priors can be challenging with the fairly complex adjusted limited dependent variable mixture models.

Possible extensions of ‘aldvmm’ could include the following features.

- Adjusted limited dependent variable beta mixture models (Gray and Alava 2018).
- Mixed model implementation for repeated measures.
- Generic function for the calculation of average marginal effects and their standard errors.

Any suggestions are welcome.

## 7 References

- Dixon, Pdraig, William Hollingworth, and John Sparrow. 2020. "Mapping to Quality of Life and Capability Measures in Cataract Surgery Patients: From Cat-Prom5 to EQ-5d-3l, EQ-5d-5l, and ICECAP-o Using Mixture Modelling." *MDM Policy & Practice* 5 (1): 2381468320915447.
- Dolan, Paul. 1997. "Modeling Valuations for EuroQol Health States." *Medical Care*, 1095–1108.
- Dowd, Bryan E, William H Greene, and Edward C Norton. 2014. "Computation of Standard Errors." *Health Services Research* 49 (2): 731–50.
- Fuller, Gordon Ward, Monica Hernandez, David Pallot, Fiona Lecky, Mathew Stevenson, and Belinda Gabbe. 2017. "Health State Preference Weights for the Glasgow Outcome Scale Following Traumatic Brain Injury: A Systematic Review and Mapping Study." *Value in Health* 20 (1): 141–51.
- Gray, Laura A, and Mónica Hernández Alava. 2018. "A Command for Fitting Mixture Regression Models for Bounded Dependent Variables Using the Beta Distribution." *The Stata Journal* 18 (1): 51–75.
- Gray, Laura A, Mónica Hernández Alava, and Allan J Wailoo. 2018. "Development of Methods for the Mapping of Utilities Using Mixture Models: Mapping the AQLQ-s to the EQ-5d-5l and the Hui3 in Patients with Asthma." *Value in Health* 21 (6): 748–57.
- Gray, Laura A, Allan J Wailoo, and Monica Hernandez Alava. 2018. "Mapping the FACT-b Instrument to EQ-5d-3l in Patients with Breast Cancer Using Adjusted Limited Dependent Variable Mixture Models Versus Response Mapping." *Value in Health* 21 (12): 1399–1405.
- Hernández Alava, Mónica, and Allan Wailoo. 2015. "Fitting Adjusted Limited Dependent Variable Mixture Models to EQ-5d." *The Stata Journal* 15 (3): 737–50.
- Hernández Alava, Mónica, Allan J Wailoo, and Roberta Ara. 2012. "Tails from the Peak District: Adjusted Limited Dependent Variable Mixture Models of EQ-5d Questionnaire Health State Utility Values." *Value in Health* 15 (3): 550–61.
- Hernández Alava, Mónica, Allan Wailoo, Fred Wolfe, and Kaleb Michaud. 2013. "The Relationship Between EQ-5d, HAQ and Pain in Patients with Rheumatoid Arthritis." *Rheumatology* 52 (5): 944–50.
- . 2014. "A Comparison of Direct and Indirect Methods for the Estimation of Health Utilities from Clinical Outcomes." *Medical Decision Making* 34 (7): 919–30.
- Hvidberg, Michael Falk. 2016. "A Framework for Identifying Disease Burden and Estimating Health-Related Quality of Life and Prevalence Rates for 199 Medically Defined Chronic Conditions."
- Mukuria, Clara, Donna Rowen, Sue Harnan, Andrew Rawdin, Ruth Wong, Roberta Ara, and John Brazier. 2019. "An Updated Systematic Review of Studies Mapping (or Cross-Walking) Measures of Health-Related Quality of Life to Generic Preference-Based Measures to Generate Utility Values." *Applied Health Economics and Health Policy*, 1–19.
- Mulhern, Brendan, Yan Feng, Koonal Shah, Mathieu F Janssen, Michael Herdman, Ben van Hout, and Nancy Devlin. 2018. "Comparing the UK EQ-5d-3l and English EQ-5d-5l Value Sets." *Pharmacoeconomics* 36 (6): 699–713.
- NHS Digital. 2013. "Finalised Patient Reported Outcome Measures (PROMs) in England - April 2011 to March 2012. Patient Reported Outcome Measures (PROMs)." <https://Digital.nhs.uk/Data-and-Information/Publications/Statistical/Patient-Reported-Outcome-Measures-Proms/Finalised-Patient-Reported-Outcome-Measures-Proms-in-England-April-2011-to-March-2012> October 15, 2013.
- Pennington, Becky M, Mónica Hernández-Alava, Philip Hykin, Sobha Sivaprasad, Laura Flight, Abualbishr Alshreef, and John Brazier. 2020. "Mapping from Visual Acuity to EQ-5d, EQ-5d with Vision Bolt-on, and VFQ-UI in Patients with Macular Edema in the LEAVO Trial." *Value in Health* 23 (7): 928–35.
- Whitmore, GA. 1986. "Prediction Limits for a Univariate Normal Observation." *The American Statistician* 40 (2): 141–43.

- Xu, Richard Huan, Eliza Lai Yi Wong, Jun Jin, Ying Dou, and Dong Dong. 2020. “Mapping of the EORTC QLQ-C30 to EQ-5d-5l Index in Patients with Lymphomas.” *The European Journal of Health Economics* 21 (9): 1363–73.
- Yang, Fan, Carlos KH Wong, Nan Luo, James Piercy, Rebecca Moon, and James Jackson. 2019. “Mapping the Kidney Disease Quality of Life 36-Item Short Form Survey (KDQOL-36) to the EQ-5d-3l and the EQ-5d-5l in Patients Undergoing Dialysis.” *The European Journal of Health Economics* 20 (8): 1195–1206.



## 8 Appendix

### 8.1 Covariance matrices across optimization methods

Covariance matrices were incomplete or missing entirely (**FALSE**) in multiple optimization approaches (table 9)

Table 9: Covariance matrix by optimization method

	Nelder-Mead	BFGS	CG	L-BFGS-B	nlminb	Rcgmin	Rvmmin	hjn
zero	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
random	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
constant	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
sann	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

## 8.2 Modified Hosmer-Lemeshow test

```
# Number of percentiles
ngroup <- 10

# Extract expected values and residuals
yhat <- fit1_all[["zero"]][["Nelder-Mead"]][["pred"]][["yhat"]]
res <- fit1_all[["zero"]][["Nelder-Mead"]][["pred"]][["res"]]

# Make groups
group <- as.numeric(cut(yhat, breaks = ngroup), na.rm=TRUE)

# Auxiliary regression
aux <- stats::lm(res ~ factor(group))

# Data set of predictions from auxiliary regressions
newdf <- data.frame(group = unique(group)[order(unique(group))])
predict <- predict(aux,
                    newdata = newdf,
                    se.fit = TRUE,
                    interval = 'confidence',
                    level = 0.95)

plotdat <- as.data.frame(rbind(
  cbind(group = newdf$group,
        outcome = "mean",
        value = predict$fit[, 'fit']),
  cbind(group = newdf$group,
        outcome = "ll",
        value = predict$fit[, 'lwr']),
  cbind(group = newdf$group,
        outcome = "ul",
        value = predict$fit[, 'upr'])
))

# Make plot
plot <- ggplot2::ggplot(plotdat, aes(x = factor(as.numeric(group)),
                                     y = as.numeric(value),
                                     group = factor(outcome))) +
  geom_line(aes(linetype = factor(outcome)))
```

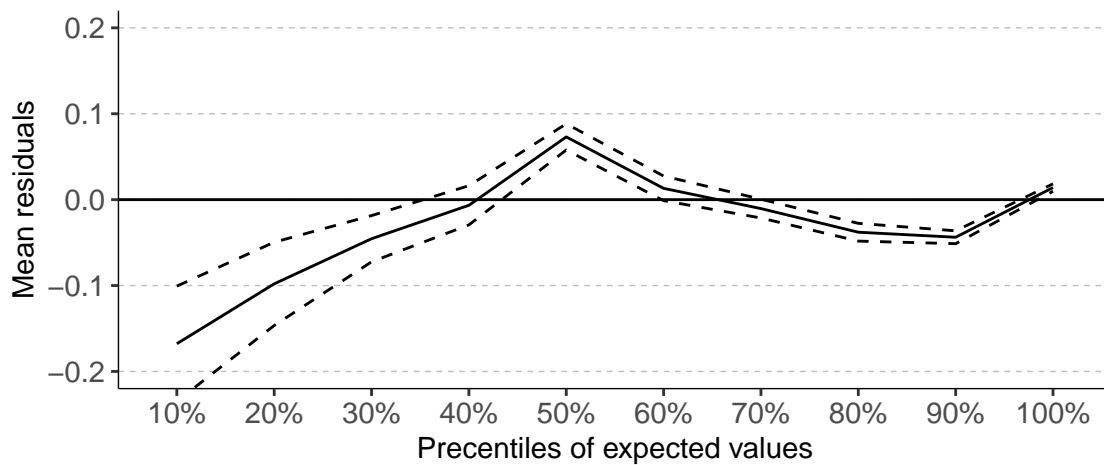


Figure 7: Mean residuals over deciles of expected values, “Nelder-Mead” with “zero” starting values

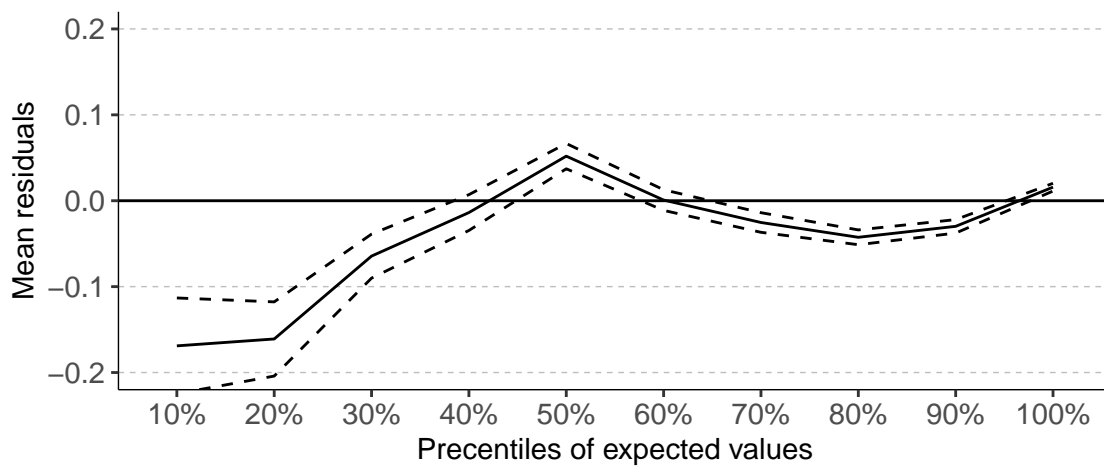


Figure 8: Mean residuals over deciles of expected values, “BFGS” with “zero” starting values

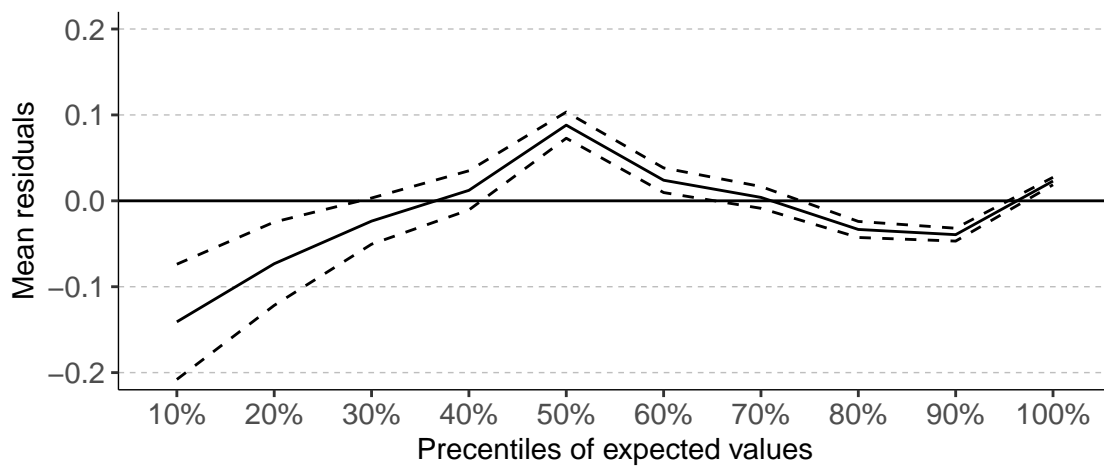


Figure 9: Mean residuals over deciles of expected values, “hjn” with “zero” starting values

### 8.3 Comparison to results by Hernández Alava and Wailoo (2015)

Table 10: Regression results from Hernandez Alava and Wailoo 2015, model 1 with default initial values from function ml

		Estimate	Std. Err.	z	P> z	[95% Conf. Interval]
E[y c, X]						
Comp1	(Intercept)	-0.0883	0.0084	-10.54	0	-0.1048 -0.0719
	hr	0.2308	0.0022	102.84	0	0.2264 0.2352
	lnsigma	-1.6461	0.0089	-184.92	0	-1.6636 -1.6287
Comp2	(Intercept)	3930832				
	hr	885110.9				
	lnsigma	-164.8169				
P[c X]						
Comp1	(Intercept)	7.3206	0.7314	10.01	0	5.8871 8.7542
N = 10565 ll = -577.38						

Table 11: Regression results of model 2 with user-defined starting values in the "hjn" algorithm

		Estimate	Std. Err.	z	P> z	[95% Conf. Interval]
E[y c, X]						
Comp1	(Intercept)	-0.4307	0.0224	-19.2350	1.0000	-0.4746 -0.3868
	hr	0.3135	0.0065	47.9501	0.0000	0.3007 0.3263
	lnsigma	-1.2481	0.0215	-58.0028	1.0000	-1.2903 -1.2060
Comp2	(Intercept)	0.2358	0.0069	34.1839	0.0000	0.2223 0.2494
	hr	0.1459	0.0019	76.3631	0.0000	0.1421 0.1496
	lnsigma	-2.4624	0.0178	-138.1256	1.0000	-2.4973 -2.4274
P[c X]						
Comp1	(Intercept)	-0.7280	0.0607	-12.0015	1.0000	-0.8469 -0.6091
N = 10549 ll = 706.32 AIC = -1398.65 BIC = -1398.65						

Table 12: Regression results from Hernandez Alava and Wailoo 2015, model 1 with initial values from a constant-only model

		Estimate	Std. Err.	z	P> z	[95% Conf.	Interval]
E[y c, X]							
Comp1	(Intercept)	-0.4029312	0.0215507	-18.7	0	-0.4451698	-0.3606925
	hr	0.3050275	0.0063324	48.17	0	0.2926162	0.3174389
	lnsigma	-1.263205	0.0211453	-59.74	0	-1.304649	-1.221761
Comp2	(Intercept)	0.2261472	0.0069948	32.33	0	0.2124377	0.2398566
	hr	0.1480158	0.0019441	76.13	0	0.1442053	0.1518263
	lnsigma	-2.45414	0.0177415	-138.33	0	-2.488913	-2.419367
P[c X]							
Comp1	(Intercept)	-0.7075574	0.061444	-11.52	0	-0.8279855	-0.5871293
N = 10565 ll = 685.79							

Table 13: Regression results of model 2 with user-defined starting values in the "hjn" algorithm

		Estimate	Std. Err.	z	P> z	[95% Conf.	Interval]
E[y c, X]							
Comp1	(Intercept)	-0.1192	0.0150	-7.9316	1.0000	-0.1487	-0.0897
	hr	0.0802	0.0059	13.6648	0.0000	0.0687	0.0917
	lnsigma	-1.8720	0.0349	-53.6924	1.0000	-1.9403	-1.8036
Comp2	(Intercept)	0.2010	0.0063	31.7730	0.0000	0.1886	0.2134
	hr	0.1562	0.0017	94.6157	0.0000	0.1529	0.1594
	lnsigma	-2.2133	0.0109	-202.4097	1.0000	-2.2347	-2.1918
P[c X]							
Comp1	(Intercept)	1.7744	0.1311	13.5344	0.0000	1.5175	2.0314
	hr	-1.3526	0.0442	-30.5738	1.0000	-1.4393	-1.2659
N = 10549 ll = 939.22 AIC = -1862.44 BIC = -1862.44							

Table 14: Regression results from Hernandez Alava and Wailoo 2015, model 2 with user-defined initial values

		Estimate	Std. Err.	z	P> z	[95% Conf.	Interval]
E[y c, X]							
Comp1	(Intercept)	0.0129138	0.0269556	0.48	0.632	-0.0399183	0.0657459
	hr	0.0887522	0.0108455	8.18	0	0.0674954	0.110009
	lnsigma	-1.319285	0.0340906	-38.7	0	-1.386101	-1.252468
Comp2	(Intercept)	0.1763049	0.0075321	23.41	0	0.1615422	0.1910675
	hr	0.1619008	0.0019171	84.45	0	0.1581433	0.1656582
	lnsigma	-2.271434	0.012582	-180.53	0	-2.296095	-2.246774
P[c X]							
Comp1	(Intercept)	2.431909	0.1748113	13.91	0	2.089285	2.774533
	hr	-1.395115	0.0557792	-25.01	0	-1.50444	-1.28579
N = 10565 ll = 942.71							