

ANCHORING VIGNETTETS IN R: A (DIFFERENT KIND OF) VIGNETTE

JONATHAN WAND AND GARY KING

ABSTRACT. The `anchors` package in R implements the techniques described in King et al. (2004), King and Wand (2007), and Wand (2007). The procedures include methods both for evaluating and choosing anchoring vignettes, and for analyzing the resulting data. This document provides a quick introduction to setting up and using `anchors`. A companion article is also available (Wand, King, and Lau, 2007), providing details on the logic of the analysis and results for the same data used in this document. The latest version of this software and related materials are available at the `anchors` website:

<http://wand.stanford.edu/anchors/>.

1. A QUICK OVERVIEW

This section assumes that you have already have `anchors` installed and want a quick introduction/overview. Information on installation, background, and examples of `anchors` are provide in detail in subsequent sections. All examples and objects described in this document assume that you have loaded the package in an R session,

```
> library(anchors)
```

A list of the functions and datasets with help pages can be found using,

```
> help(package = "anchors")
```

For a list of datasets of vignette surveys included in `anchors`, see

```
> data(package = "anchors")
```

For a list of demonstrations of functions, uses of data, and replications of published results,

```
> demo(package = "anchors")
```

The main model functions are,

<code>anchors</code>	Nonparameteric analysis of surveys with anchoring vignettes
<code>chopit</code>	Compound Hierarchical Ordered Probit
<code>cpolr</code>	Censored ordered probit

Date: February 9, 2007, Version: 2.0.

Our thanks to NIA/NIH (grant P01 AG17625-01), NSF (grants SES-0112072 and IIS-9874747), and the Weatherhead Center for International Affairs, and the Robert Wood Johnson Foundation's Health Policy Scholars program for research support.

The main combinatoric and diagnostic functions include,

entropy	Calculate entropy of C distribution for subsets of vignettes
vignette.order	How do respondent actually order vignettes
anchors.intervals	Calculate rate of ties in C for subsets of vignettes

Datasets with anchoring vignettes that are made available by the `anchors` package include

chopitsim	Simulated Data for test chopit function
mexchn	China-Mexico political efficacy data
poleff	Simulated Political Efficacy Data
poleffna	Simulated Political Efficacy Data with NA (demo only, don't use)
freedom	Individual freedom of speech data
sleep	Sleep data for china
selfcare	Self-care data for china
table1	Reference from Table 1 of King and Wand (2007)
table1src	Specific response values that have inequalities to create table1

Any of these can be loaded with `data()`, for example,

```
> data(freedom)
```

Demonstration files are available, both to provide examples of the use of functions and as an aid to those who would simply like to re-compute published results that have used versions of the `anchors` package,

anchors	Demo of anchors: summary, plot
chopit	Demo of chopit: summary, plot
chopit.re	Demo of chopit: with random effects
chopit.mexchn	King et al (2004) Table 2 (non-linear taus)
anchors.freedom3	Wand et al (2007) Figure 2 histogram with 3 vignettes
anchors.freedom6	Wand et al (2007) Figure 1 histogram with 6 vignettes
anchors.mexchn	King and Wand (2007) Figure 1 histogram
entropy.mexchn	King and Wand (2007) Figure 2 entropy()
entropy.sleep	King and Wand (2007) Figure 3 entropy()
entropy.self	King and Wand (2007) Figure 4 entropy()
anchors.vign2	King and Wand (2007) Table 1 anchors()

Any of these can be invoked with `demo()`, for example,

```
> demo(anchors)
```

2. GETTING STARTED: INSTALLATION AND THE BASICS

We begin by walking through how to set-up `anchors` on your computer to facilitate the interactive use of the examples that follow. There are many introductions to R available on the R site, <http://www.r-project.org>, and this is only intended as a brief summary with an emphasis on helping you to specifically get started with `anchors`.

Prior to installing `anchors`, you will need to install the R statistical package available via <http://www.r-project.org>. Use at least R 2.4. For details on installing R the FAQ at <http://cran.r-project.org/faqs.html> are helpful.

Once you have R installed, and given you have an active internet connection, the easiest way to install the `anchors` package is from the R command line,

```
> install.packages("anchors", dependencies = TRUE,
  repos=c("http://wand.stanford.edu/R/CRAN", "http://cran.r-project.org"))
```

which will also install the `rgenoud` package if it is not already installed on your system. Alternatively, for *nix systems, you can also install the package manually by

- (1) downloading the source code from the anchors website: `anchors_2.0.tar.gz`.
- (2) running from the *nix shell, in the same directory as the downloaded file,


```
% R CMD INSTALL anchors_2.0.tgz
```

Once the `anchors` package is installed, and an R session is begun, the package is made available by invoking on the R command-line,

```
> library(anchors)
```

The full list of functions and datasets made available by `anchors` can be found by invoking at any time,

```
> help(package = "anchors")
```

and each individual object with a help file can be queried individually, for example,

```
> help(anchors.overview)
> help(anchors)
```

The page for `help(anchors.overview)` is a particularly useful place to start since it also has a link to the main functions of the `anchors` package. It may be helpful to reference the `help` pages for functions at the same time as reading about their use here.

Similarly, the list of demonstration files can be shown by,

```
> demo(package = "anchors")
```

each of which can be run individually, for example,

```
> demo(anchors)
```

but the others are also helpful for showing additional examples of applications. Replications of published using this package results are included among the demonstration files; a list of replication files is provide at the end of this document.

Additionally, there are a number of datasets that are included as part of the `anchors` package. The list can be found by,

```
> data(package = "anchors")
```

each of which comes with its own `help()` file describing the variables. A dataset can be loaded by,

```
> data(freedom)
```

which loads the political freedom dataset, which will be a running example.

Lists summarizing the functions, datasets, and demo files that are part of `anchors` are also provided at the end of this document.

3. INTRODUCTION TO ANCHORING VIGNETTES

Consider a survey question along with response categories that is asked as a self-assessment,

How free do you think you are to express yourself without fear of government reprisal? (1) Completely Free, (2) Very Free, (3) Moderately Free, (4) Slightly Free, (5) Not Free at All

One key difficulty of analyzing the results from such a survey question is the possibility that individuals apply different standards in the selection of a response category. Researchers have tried to ameliorate the problems of interpersonal and cross-cultural incomparability in survey research with careful question wording, translation (and back translation), focus groups, cognitive debriefing, and other techniques, most of which are designed to improve the survey question. In contrast, anchoring vignettes is a technique that seeks to bring additional data to bear on the problem.

For example, vignettes corresponding to the above political freedom question attempt to describe hypothetical individuals who have different levels of freedom from government reprisal. The following six vignettes are intended to correspond to distinct levels of political freedom in order of decreasing freedom,

vign1: [Kay] does not like many of the government’s policies. She frequently publishes her opinion in newspapers, criticizing decisions by officials and calling for change. She sees little reason these actions could lead to government reprisal.

vign2: [Michael] disagrees with many of the government’s policies. Though he knows criticism is frowned upon, he doesn’t believe the government would punish someone for expressing critical views. He makes his opinion known on most issues without regard to who is listening.

vign3: [Bob] has political views at odds with the government. He has heard of people occasionally being arrested for speaking out against the government, and government leaders sometimes make political speeches condemning those who criticize. He sometimes writes letters to newspapers about politics, but he is careful not to use his real name.

vign4: [Connie] does not like the government’s stance on many issues. She has a friend who was arrested for being too openly critical of governmental leaders, and so she avoids voicing her opinions in public places.

vign5: [Vito] disagrees with many of the government’s policies, and is very careful about whom he says this to, reserving his real opinions for family and close friends only. He knows several men who have been taken away by government officials for saying negative things in public.

vign6: [Sonny] lives in fear of being harassed for his political views. Everyone he knows who has spoken out against the government has been arrested or

taken away. He never says a word about anything the government does, not even when he is at home alone with his family.

After each of these vignettes, a corresponding evaluation question is asked with the same response categories as for the self-assessment.

How free do you think [name] is to express [him/her]self without fear of government reprisal? (1) Completely Free, (2) Very Free, (3) Moderately Free, (4) Slightly Free, (5) Not Free at All

Note: In the case where there are missing values for responses to the self-assessment or the vignettes, it is important that these be coded as '0' (zero), instead of NA or some other missing value if you wish to retain the other (non-missing) responses of an individual in the parametric model to be described shortly (see `chopit`). For all non-parametric analysis that rely on `anchors` or `vignette.order`, cases with missing responses (either NA or zero) must be listwise deleted. We provide a handy function, `replacevalue`, that facilitates the alteration of the coding of missing values for subsets of variables.

4. INDEXING NOTATION

Our notation is a generalization of King et al. designed to accommodate our enhancements to the various models. We index survey questions, response categories, and respondents as follows:

- We index *survey questions* by the pair (s, j) , where question set s ($s = 1, \dots, S$) corresponds to the self-assessment question number and refers to the set of questions that includes the self-assessment question (indicated by $j = 0$) and, optionally, one or more vignette questions (indicated by $j = 1, \dots, J_s$).
- We index *response categories* by k ($k = 1, \dots, K_s$) separately for each survey question since they can each have different response categories. Each set of questions (self-assessment and vignettes) must have the same number of choice categories (coded as increasing sequential integers starting with 1). *Missing values* (whether structural, because the question was not asked, or due to nonresponse) should be coded as $k = 0$.
- We index *respondents* by i or ℓ . Respondent i ($i = 1, \dots, n$) is asked all of the self-assessment questions. Respondent ℓ ($\ell = 1, \dots, N$) is asked all of the vignette questions. (Respondents are indexed for self-assessment and vignette questions separately since each could be asked of independent samples; if they are asked of the same individuals, then $i = \ell$ and $n = N$.) If your survey design asks each set of vignette questions in separate samples (and separate from the self-assessment question), then index each set of vignettes according to unique values of ℓ and use the missing value code ($k = 0$) for vignettes that are not asked of a subgroup; in other words, stack the data in block diagonal format.

Thus, every mathematical symbol in the model could be indexed by s , j , k , and either i or ℓ . In practice, we drop indexes that are constant.

5. A NONPARAMETRIC APPROACH

5.1. **Definition.** Define C_{is} as the self-assessment *relative* to the corresponding set of vignettes. Let y_i be the self-assessment response and z_{i1}, \dots, z_{iJ} be the J vignette responses, for the i th respondent. For respondents with consistently ordered rankings on all vignettes ($z_{j-1} < z_j$, for $j = 2, \dots, J$), we create the DIF-corrected self-assessment C_i

$$(1) \quad C_i = \begin{cases} 1 & \text{if } y_i < z_{i1} \\ 2 & \text{if } y_i = z_{i1} \\ 3 & \text{if } z_{i1} < y_i < z_{i2} \\ \vdots & \vdots \\ 2J + 1 & \text{if } y_i > z_{iJ} \end{cases}$$

Respondents who give tied or inconsistently ordered vignette responses may have an interval values of C , if the tie/inconsistency results in multiple conditions in equation 1 appearing to be true. A more general definition of C is defined as the minimum to maximum values among all the conditions that hold true in equation 1. Values of C that are intervals, rather than scalar, represent the set of inequalities over which the analyst cannot distinguish without further assumption.

5.2. EXAMPLE CODE: `anchors()`. This example again first loads the library and example dataset, and then `anchors()` calculates C for each individual. In the non-parametric estimation, only *one* self-question and corresponding set of vignettes are analyzed at a time.

```
> library(anchors)
> data(freedom)
> a1 <- anchors(self ~ vign2 + vign3 + vign4 + vign5 +
+   vign6, freedom)
> summary(a1)
```

NON-PARAMETRIC ANCHORS SUMMARY:

Number of cases: 1763 with interval value, 1737 with scalar value, 0 dropped due to missing

Maximum possible C value: 11

C : Frequency and proportions
Cs to Ce

	N	Prop
1 to 1	387	0.1105714286
2 to 2	279	0.0797142857
3 to 3	336	0.0960000000
4 to 4	81	0.0231428571
5 to 5	59	0.0168571429
6 to 6	28	0.0080000000
7 to 7	11	0.0031428571
8 to 8	31	0.0088571429

```

9 to 9 22 0.0062857143
10 to 10 164 0.0468571429
11 to 11 339 0.0968571429
1 to 4 16 0.0045714286
1 to 5 12 0.0034285714
1 to 6 25 0.0071428571
1 to 7 5 0.0014285714
1 to 8 31 0.0088571429
1 to 9 5 0.0014285714
1 to 10 32 0.0091428571
1 to 11 19 0.0054285714
2 to 4 15 0.0042857143
2 to 5 11 0.0031428571
2 to 6 22 0.0062857143
2 to 7 4 0.0011428571
2 to 8 51 0.0145714286
2 to 9 19 0.0054285714
2 to 10 177 0.0505714286
2 to 11 91 0.0260000000
3 to 6 31 0.0088571429
3 to 7 3 0.0008571429
3 to 8 93 0.0265714286
3 to 9 29 0.0082857143
3 to 10 16 0.0045714286
3 to 11 11 0.0031428571
4 to 6 16 0.0045714286
4 to 7 2 0.0005714286
4 to 8 94 0.0268571429
4 to 9 39 0.0111428571
4 to 10 175 0.0500000000
4 to 11 39 0.0111428571
5 to 8 80 0.0228571429
5 to 9 38 0.0108571429
5 to 10 9 0.0025714286
5 to 11 6 0.0017142857
6 to 8 107 0.0305714286
6 to 9 61 0.0174285714
6 to 10 242 0.0691428571
6 to 11 52 0.0148571429
7 to 10 1 0.0002857143
7 to 11 1 0.0002857143
8 to 10 44 0.0125714286
8 to 11 39 0.0111428571

```

The names of vignettes must be passed to the function in the same order as the direction of the responses. In the example, `vign2` is in the same (highest) direction as the response category 1, while the `vign6` is in the same direction (lowest) as the response category 5. (We drop `vign1` here for space reason when printing the summary—with the different combinations of intervals of C can be numerous.)

If `anchors` produces many ties you should check that you passed the vignettes in the correct order, but we also offer a function that investigates the ordering of vignettes in detail.

5.3. EXAMPLE CODE: `vignette.order()`. The function `vignette.order`, and the associated methods `summary.vignette.order` and `plot.vignette.order` investigate the relationship between vignette responses *without* reference to the self-assessment question.

```
> vo1 <- vignette.order(~vign2 + vign3 + vign4 +
+   vign5 + vign6, freedom)
> summary(vo1, top = 10, digits = 3, verbose = TRUE)
```

```
Vignette Orderings Frequencies
Grouping Ties
```

```
Deleted 0 Observations Due to Missing Data, 3500 Observations Remain
```

```
Number of cases with at least two distinct vignette responses: 3223
and with no violations of natural ordering: 1178
and with no more than 1 violation of natural ordering: 1959
and with no more than 2 violation of natural ordering: 2621
```

Ngroup	Nviolation					
	0	1	2	3	4	5+
1	277	0	0	0	0	0
2	409	129	126	78	68	37
3	513	376	326	122	80	93
4	254	264	204	64	17	40
5	2	12	6	0	0	3

```
Proportion of cases a vignette (row) is less than another (column):
  <1  <2  <3  <4  <5
1  NA 0.663 0.732 0.707 0.754
2 0.121 NA 0.457 0.363 0.575
3 0.080 0.138 NA 0.183 0.374
4 0.068 0.198 0.339 NA 0.495
5 0.070 0.081 0.100 0.103 NA
```

```
Upper tri =   p_{ij} - p_{ji} (negative values suggest misorderings)
Lower tri = 1 - p_{ij} - p_{ji} (big numbers means many ties)
  1  2  3  4  5
1  NA 0.542 0.652 0.639 0.684
2 0.215 NA 0.320 0.165 0.494
3 0.188 0.440 NA -0.156 0.275
4 0.405 0.477 0.345 NA 0.392
5 0.225 0.176 0.526 0.402 NA
```

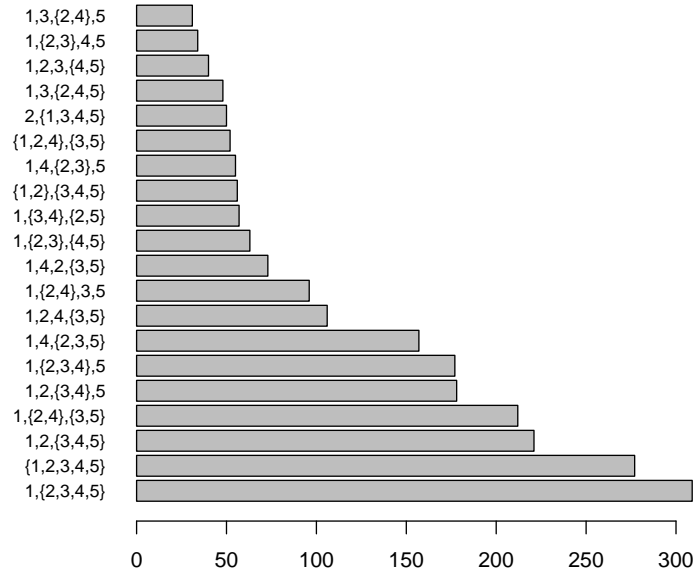
Top 10 orderings (out of 262 unique orderings):

	Frequency	Proportion	Ndistinct	Nviolation
1,{2,3,4,5}	309	0.0883	2	0
{1,2,3,4,5}	277	0.0791	1	0
1,2,{3,4,5}	221	0.0631	3	0
1,{2,4},{3,5}	212	0.0606	3	1
1,2,{3,4},5	178	0.0509	4	0
1,{2,3,4},5	177	0.0506	3	0
1,4,{2,3,5}	157	0.0449	3	2
1,2,4,{3,5}	106	0.0303	4	1
1,{2,4},3,5	96	0.0274	4	1
1,4,2,{3,5}	73	0.0209	4	2

```
> plot(vo1)
```

Deleted 0 Observations Due to Missing Data, 3500 Observations Remain

**Vignette Orderings Frequencies
Grouping Ties**



Details of how to interpret and use the output of the summary are provided in Wand, King, and Lau (2007), where it is discussed in detail how `vign6` is given the highest response almost half the time, however `vign4` is more often given the highest response than `vign5`.

In light of this it is worth reestimating C using the consensus ordering of the vignettes,

```
> a2 <- anchors(self ~ vign2 + vign3 + vign5 + vign4 +
+ vign6, freedom)
> summary(a2)
```

NON-PARAMETRIC ANCHORS SUMMARY:

Number of cases: 1654 with interval value, 1846 with scalar value, 0 dropped due to missing

Maximum possible C value: 11

C : Frequency and proportions

Cs to Ce

	N	Prop
1 to 1	387	0.1105714286
2 to 2	279	0.0797142857
3 to 3	336	0.0960000000
4 to 4	81	0.0231428571
5 to 5	59	0.0168571429
6 to 6	80	0.0228571429
7 to 7	38	0.0108571429
8 to 8	61	0.0174285714
9 to 9	22	0.0062857143
10 to 10	164	0.0468571429
11 to 11	339	0.0968571429
1 to 4	16	0.0045714286
1 to 5	12	0.0034285714
1 to 6	20	0.0057142857
1 to 7	1	0.0002857143
1 to 8	39	0.0111428571
1 to 9	6	0.0017142857
1 to 10	32	0.0091428571
1 to 11	19	0.0054285714
2 to 4	15	0.0042857143
2 to 5	11	0.0031428571
2 to 6	31	0.0088571429
2 to 7	6	0.0017142857
2 to 8	51	0.0145714286
2 to 9	8	0.0022857143
2 to 10	177	0.0505714286
2 to 11	91	0.0260000000
3 to 6	63	0.0180000000
3 to 7	19	0.0054285714
3 to 8	67	0.0191428571
3 to 9	7	0.0020000000
3 to 10	16	0.0045714286
3 to 11	11	0.0031428571
4 to 6	59	0.0168571429
4 to 7	17	0.0048571429
4 to 8	60	0.0171428571

```

4 to 9 15 0.0042857143
4 to 10 175 0.0500000000
4 to 11 39 0.0111428571
5 to 8 28 0.0080000000
5 to 9 11 0.0031428571
5 to 10 9 0.0025714286
5 to 11 6 0.0017142857
6 to 8 107 0.0305714286
6 to 9 31 0.0088571429
6 to 10 158 0.0451428571
6 to 11 50 0.0142857143
7 to 10 3 0.0008571429
7 to 11 1 0.0002857143
8 to 10 126 0.0360000000
8 to 11 41 0.0117142857

```

Changing the assumed ordering of the vignettes increased the number of cases without any order violation by 60 percent. With respect to the top sets of types of ordering,

The analysis of vignettes is useful both at the stage of evaluating a pilot study of survey instruments, as well at the stage of choosing how (and whether) to use particular vignettes. The results of non-parametric anchoring vignettes analysis using C are entirely dependent on which vignettes are included and the order in which they are specified.

5.4. Example Code: `cpolr()`. Utilizing the output from `anchors`, the user may wish to estimate censored ordered probit as described in King and Wand (2007). `cpolr()` fits a parametric censored ordered probit model, and a detailed few comments are warranted about this function. First, this is simply a slightly modified version of `polr` from Venables and Ripley (2006), and returns objects of class `c("cpolr", "polr")`. Generic methods for `polr` class work for `cpolr` class objects with the exception of two that are preempted by two new `cpolr` methods: `fitted.cpolr()` and `vcov.cpolr()`.

To use `cpolr` requires combining the estimated C with the original dataset containing the covariates that are of interest for the parametric imputation. This is done with the helper function `insert`, for example,

```

> freedom2 <- insert(freedom, a1)
> ca1 <- cpolr(cbind(Cs, Ce) ~ sex + age + educ +
+   as.factor(country), data = freedom2)
> summary(ca1)

```

```

Call:
cpolr(formula = cbind(Cs, Ce) ~ sex + age + educ + as.factor(country),
      data = freedom2)

```

Coefficients:

	Value	Std. Error
sex	0.118894657	0.037956811

```

age                -0.001567604 0.001135618
educ               -0.054405146 0.011423430
as.factor(country)Eurasia 0.534584422 0.087879398
as.factor(country)Oceania -0.642338446 0.050301618
                    t value
sex                 3.132367
age                -1.380397
educ               -4.762593
as.factor(country)Eurasia 6.083160
as.factor(country)Oceania -12.769737

```

Intercepts:

	Value	Std. Error	t value
1 2	-1.4803	0.0723	-20.4720
2 3	-1.0535	0.0709	-14.8532
3 4	-0.6157	0.0703	-8.7566
4 5	-0.4747	0.0705	-6.7287
5 6	-0.3570	0.0709	-5.0359
6 7	0.0097	0.0789	0.1231
7 8	0.0800	0.0792	1.0100
8 9	0.5521	0.0723	7.6349
9 10	0.6159	0.0721	8.5435
10 11	1.0412	0.0716	14.5349

Residual Deviance: 9871.161

AIC: 9901.161

(53 observations deleted due to missingness)

Fitted values can be obtain in a number of different ways, conditional or unconditional, by case or on average for each value of \hat{C} . See Wand, King, and Lau (2007) for details.

In order, we have here the average probabilities (by invoking *Cvec=TRUE*) for the unconditional, conditional, and the unconditional again.

```
> fitted(ca1, Cvec = TRUE)
```

	1	2	3	4	5
0.11873838	0.10022019	0.14104697	0.05145078	0.04423212	
	6	7	8	9	10
0.13899572	0.02587968	0.15578361	0.01788735	0.09644667	
	11				
0.10931853					

```
> fitted(ca1, a1, Cvec = TRUE)
```

	1	2	3	4	5
0.11843925	0.10100212	0.14163318	0.05137898	0.04404892	
	6	7	8	9	10
0.13847180	0.02579515	0.15532959	0.01782125	0.09643172	

```

11
0.10964804
> fitted(ca1, a1, Cvec = TRUE, unconditional.override = TRUE)
      1      2      3      4      5
0.11873838 0.10022019 0.14104697 0.05145078 0.04423212
      6      7      8      9     10
0.13899572 0.02587968 0.15578361 0.01788735 0.09644667
11
0.10931853

```

By including the original `anchors` object as the second argument of `fitted` results in conditional calculations unless the `unconditional.override=TRUE` is invoked as well.

To get the fitted values for each case, we would simply omit the `Cvec` option (since by it is `FALSE`),

```

> fitted(ca1, Cvec = TRUE)
> fitted(ca1, a1, Cvec = TRUE)
> fitted(ca1, a1, Cvec = TRUE, unconditional.override = TRUE)

```

`fitted.cpolr()` uses a combination of an object with class `cpolr` and an object of class `anchors`. `fitted.cpolr()` finds the intersection of common cases on the basis of matching row names of the two objects. In the current case, the censored ordered probit was fit using the entire sample, but fitted values are extracted for the subsets of each region by passing the `anchors` class object for the separate regions. Cases that are dropped by `cpolr()` due to missing covariates are also handled correctly when finding the intersection of the two objects.

5.5. **Example Code: `entropy()`.** Calculating entropy as suggested by Wand and King (2007) is straightforward. The `entropy()` function simply calls `anchors()`, and optionally `cpolr()` for every permutation of vignettes and calculates the minimum entropy and estimated entropy (based on the fitted values from `cpolr`) for each permutation. For more details, please see `help(entropy)` in R and King and Wand (2007).

```

> data(freedom)
> ent <- entropy(self ~ vign1 + vign3 + vign6, covar = ~as.factor(country) +
+   sex + age + educ, data = freedom)
> summary(ent, type = 1, digits = 3)

```

Summary of Entropy Calculation

Top 7 (out of 7) combinations of vignettes, ranked by minimum entropy:

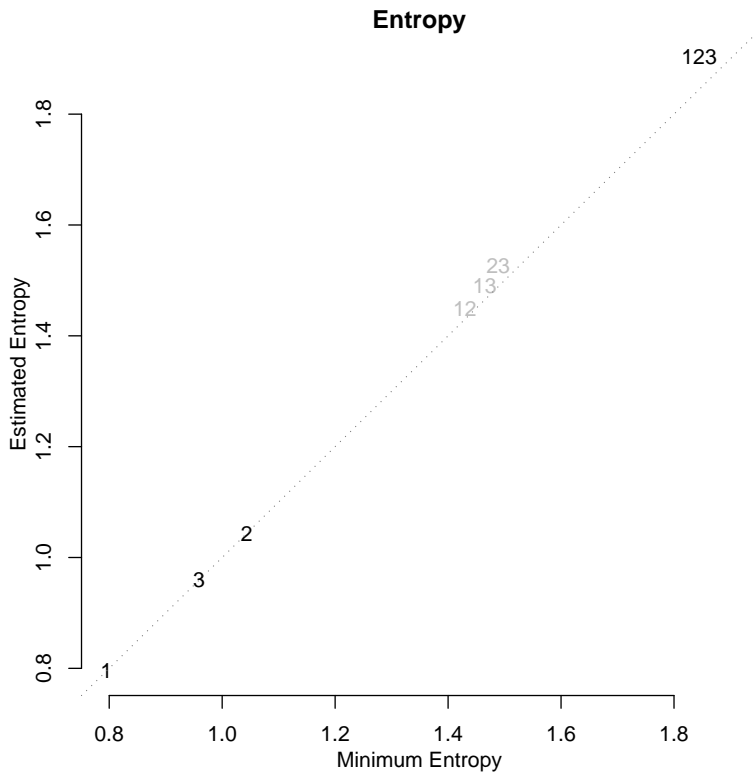
	Vignettes	Estimated entropy	Minimum entropy	N.interval
1	123	1.903	1.844	522
2	23	1.527	1.489	418
3	13	1.490	1.465	195

4	12	1.450	1.430	220
5	2	1.044	1.044	0
6	3	0.959	0.959	0
7	1	0.795	0.795	0
Avg span 2J+1				
1	1.47	7		
2	1.29	5		
3	1.14	5		
4	1.15	5		
5	1.00	3		
6	1.00	3		
7	1.00	3		

One important feature to be noted about including `covar=` variables is that cases with any missing value in the covariates will be listwise deleted for both both the estimated and minimum entropy calculations to ensure a common basis for comparisons. As such, the minimum entropy values may change as a function of what variables (if any) are included in `covar=`.

The `plot()` method for class `entropy` plots either minimum entropy by number of cases with interval C ($type=1$), or minimum entropy against estimated entropy ($type=2$). Obviously, ($type=2$) requires that the `entropy` object originally had a formula in `covar` to work.

```
> plot(ent, type = 2)
```



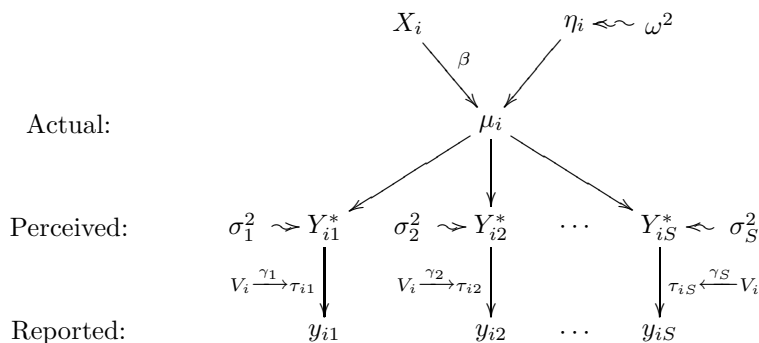


FIGURE 1. Self-Assessment Component: All levels vary over observations i . Each solid arrow denotes a deterministic effect; a squiggly arrow denotes the addition of normal random error with variance indicated at the arrow's source.

6. PARAMETRIC MODEL

This section describes the Compound Hierarchical Ordered Probit (chopit) model.

6.1. Self-assessment component. Figure 1 summarizes the self-assessment component of the model.

The *actual* level for respondent i is μ_i , a continuous unidimensional variable (with higher values indicating more freedom, better health, etc., defined by the order of the vignettes). Respondent i perceives μ_i only with random normal error so that

$$(2) \quad Y_{is}^* \sim N(\mu_i, \sigma_s^2)$$

is respondent i 's unobserved *perceived* level. The actual level is a linear function of observed covariates X_i , the first column of which can be a constant term (if it is not needed for identification) and an independent normal random effect η_i :

$$(3) \quad \mu_i = X_i\beta + \eta_i$$

with parameter β and

$$(4) \quad \eta_i \sim N(0, \omega^2).$$

The *reported* survey response category is y_{is} and is generated by the model via this observation mechanism:

$$(5) \quad y_{is} = k \quad \text{if } \tau_{is}^{k-1} \leq Y_{is}^* < \tau_{is}^k$$

with a vector of thresholds τ_{is} (where $\tau_{is}^0 = -\infty$, $\tau_{is}^{K_s} = \infty$, and $\tau_{is}^{k-1} < \tau_{is}^k$, with indexes for categories $k = 1, \dots, K_s$ and self-assessment questions $s = 1, \dots, S$) that vary over the observations as a function of a vector of covariates, V_i (the first column of which can be a constant term), and unknown parameter vectors γ_s (with

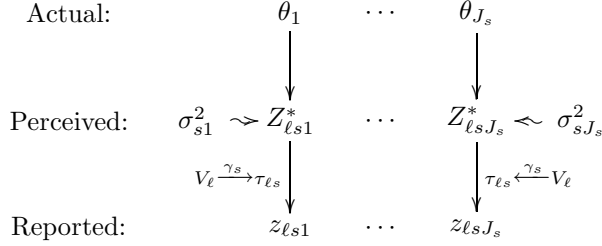


FIGURE 2. Vignette Component for question set s ($s = 1, \dots, S'$, $S' \leq S$). All levels vary over observations ℓ . Each solid arrow denotes a deterministic effect; a squiggly arrow denotes the addition of normal random error with variance indicated at the arrow's source.

elements the vector γ_s^k):

$$\begin{aligned}
(6) \quad \tau_{is}^1 &= \gamma_s^1 V_i \\
\tau_{is}^k &= \tau_{is}^{k-1} + e^{\gamma_s^k V_i} \quad (k = 2, \dots, K_s - 1)
\end{aligned}$$

6.2. Vignette Component. Figure 2 summarizes the vignette component of the model for question set s ($s = 1, \dots, S$). Under the model, one or more of the self-assessment questions have corresponding vignettes.

The actual level for vignette j is θ_j ($j = 1, \dots, J_s$), measured on the same scale as μ_i and the τ 's. Respondent ℓ perceives θ_j with random normal error so that

$$(7) \quad Z_{\ell s j}^* \sim N(\theta_j, \sigma_{s j}^2)$$

represents respondent ℓ 's unobserved assessment of the level of vignette j for question set s .

The perception of respondent ℓ about the level of vignette j elicited via a survey question s with the same K_s ordinal categories as the corresponding self-assessment question. Thus, the respondent turns the continuous $Z_{\ell s j}^*$ into a categorical answer to the survey question $z_{\ell s j}$ via this observation mechanism:

$$(8) \quad z_{\ell s j} = k \quad \text{if } \tau_{\ell s}^{k-1} \leq Z_{\ell s j}^* < \tau_{\ell s}^k$$

with thresholds determined by the same γ_s coefficients as in (6) for y_{i1} , and the same explanatory variables but with values measured for units ℓ , V_ℓ :

$$\begin{aligned}
(9) \quad \tau_{\ell 1}^1 &= \gamma_s^1 V_\ell \\
\tau_{\ell 1}^k &= \tau_{\ell 1}^{k-1} + e^{\gamma_s^k V_\ell} \quad (k = 2, \dots, K_1 - 1).
\end{aligned}$$

6.3. Identification. The model as specified above has an infinite number of equivalent maximum likelihood solutions. To identify the model, two choices must be made:

- (1) The mean of the actual level must be set, by choosing one point. This can be done by setting the constant term $\beta_0 = 0$ (in which case be aware of your choice of the scale of the variables in X), or one of the θ 's.
- (2) The variance of the actual level must also be set. This can be done by setting all the self-assessment variances (such as $\sigma_s^2 = 1$, for all s) or by setting another point among β_0 or the θ 's.

Two common parameterizations are as follows:

- (1) The ordinal probit parameterization is useful for comparing chopit to this simpler model. Set $\beta_0 = 0$ and $\sigma_1^2 = \dots = \sigma_S^2 = 1$.
- (2) Another option is parameterization defined by the extreme vignettes. Let $\theta_1 = 0$ and $\theta_{J_s} = 1$. This lets estimates of μ be interpreted on the scale of the vignettes, with 0 being the level of the lowest vignette and 1 the level of the highest. Note that μ can still be higher than 1 or lower than 0, but the units are easily interpretable.

6.4. EXAMPLE CODE: `chopit()`. The `chopit()` function provided by `anchors` at its most basic simply requires specifying the formula's defining ys , zs , and τs . For example, using variables from the `data(freedom)` dataset, we have the *named* list.

```
> fo <- list(self = self ~ sex + age + educ + factor(country),
+          vign = cbind(vign1, vign2, vign3, vign4, vign5,
+                      vign6) ~ 1, tau = ~sex + age + educ +
+                      factor(country))
```

The names `self=`, `vign=`, and `tau=` as written, are required. On the LHS of the equality signs are the variables of the dataset that specify the details of the models as for other models (e.g., `lm()`).

The self-assessment variable `self` is modeled to have a mean that is a linear additive function of `sex`, `age`, `educ` and `country` dummies. The vignettes are specified as a vector of outcomes `cbind(vign1,vign2,vign3,vign4,vign5,vign6)` as a function of only an intercept '`~ 1`'. This is both a simple and accurate way to describe the model of θ s which are the mean locations of the vignettes. The τ cutpoints shared by the self-assessment and the vignettes are specified as their own formula without a LHS variable.

Beyond the formula and data, the rest will be set by default in the basic invocation,

```
> out <- chopit(fo, data = freedom)
```

which can be summarized by the `summary.chopit` method,

```
> summary(out)
```

Summary of chopit model

Model formula:

```
$self
self ~ sex + age + educ + factor(country)
```

```
$vign
cbind(vign1, vign2, vign3, vign4, vign5, vign6) ~ 1
```

```
$tau
~sex + age + educ + factor(country)
```

Coefficients:

	chopit.coeff	chopit.se
gamma1.cut1.(Intercept)	-1.6697	0.0774
gamma1.cut1.sex	0.0570	0.0228
gamma1.cut1.age	-0.0028	0.0007
gamma1.cut1.educ	0.0109	0.0068
gamma1.cut1.factor(country)Eurasia	0.0447	0.0504
gamma1.cut1.factor(country)Oceania	-0.1262	0.0309
gamma1.cut2.(Intercept)	0.6655	0.0388
gamma1.cut2.sex	-0.0426	0.0205
gamma1.cut2.age	0.0013	0.0006
gamma1.cut2.educ	-0.0140	0.0061
gamma1.cut2.factor(country)Eurasia	-0.0286	0.0449
gamma1.cut2.factor(country)Oceania	0.0260	0.0274
gamma1.cut3.(Intercept)	0.7068	0.0319
gamma1.cut3.sex	-0.0211	0.0167
gamma1.cut3.age	-0.0001	0.0005
gamma1.cut3.educ	0.0112	0.0051
gamma1.cut3.factor(country)Eurasia	0.0250	0.0374
gamma1.cut3.factor(country)Oceania	-0.0985	0.0218
gamma1.cut4.(Intercept)	0.5937	0.0294
gamma1.cut4.sex	0.0436	0.0159
gamma1.cut4.age	0.0007	0.0005
gamma1.cut4.educ	0.0163	0.0049
gamma1.cut4.factor(country)Eurasia	0.0605	0.0365
gamma1.cut4.factor(country)Oceania	0.0166	0.0211
lnse.re	1.0000	NaN
lnse.self	1.0000	NaN
lnse.vign.vign1	0.7951	0.0183
lnse.vign.vign2	0.9974	0.0239
lnse.vign.vign3	0.7546	0.0173
lnse.vign.vign4	0.8336	0.0208
lnse.vign.vign5	0.7246	0.0171
lnse.vign.vign6	1.3307	0.0420
theta1.vign1	-1.0863	0.0721
theta1.vign2	-1.2051	0.0734
theta1.vign3	-0.2478	0.0706
theta1.vign4	0.1660	0.0715
theta1.vign5	-0.0562	0.0706
theta1.vign6	0.9519	0.0820
beta.(Intercept)	0.0000	NaN

beta.sex	0.1434	0.0388
beta.age	-0.0019	0.0012
beta.educ	-0.0569	0.0117
beta.factor(country)Eurasia	0.4600	0.0897
beta.factor(country)Oceana	-0.7019	0.0517

-Log-likelihood of CHOPIT: 32421.69

Partition of CHOPIT -Log-likelihood by question:

	-LL	N
Self (self)	5154.965	3447
vign1	5032.314	3447
vign2	5207.052	3447
vign3	4766.234	3447
vign4	4340.710	3447
vign5	4485.543	3447
vign6	3434.877	3447

Number of observations, pre- and post-listwise deletion:

- a) full dataset 3500
- b) self-responses cases used 3447
- c) vign-responses cases used 3447

The default invocation uses the the ordinal probit normalization, which identifies/normalizes the model by omitting the intercept in μ , and setting $\sigma_1 = 1$ (the variance of the first self-assessment question). If one specified the explanatory variables of `self=` to include an intercept, then that intercept parameter would be constrained to be zero as will be `beta.(Intercept)` in this example.

7. HELPER FUNCTIONS

7.1. EXAMPLE CODE: `insert()`. Insert the DIF corrected variable into the original data frame, with missing values for observations for which it was impossible to calculate DIF correction (due to missingness in either the self-response or one or more of the vignette responses).

```
> data(freedom)
> names(freedom)

 [1] "sex"      "age"      "educ"     "country"  "self"
 [6] "vign1"    "vign2"    "vign3"    "vign4"    "vign5"
[11] "vign6"

> C <- anchors(self ~ vign1 + vign3 + vign6, data = freedom)
> freedom2 <- insert(freedom, C = C)
> names(freedom2)

 [1] "sex"      "age"      "educ"     "country"  "self"
 [6] "vign1"    "vign2"    "vign3"    "vign4"    "vign5"
[11] "vign6"    "Ce"      "Cs"
```

The new columns in freedom are Cs and Ce.

7.2. EXAMPLE CODE: `replacevalue()`. Not surprisingly, this function replaces a single value in a set of columns with another given value. This makes it easy to change the default missing value indicator. For example,

```
> data(poleff)
> data(poleffna)
> cat("convert NA to 0\n")

convert NA to 0

> dd <- replacevalue(poleffna, c("xsayself", "xsay1",
+   "xsay2", "xsay3", "xsay4", "xsay5"))
> cat("convert 0 to NA\n")

convert 0 to NA

> dd2 <- replacevalue(poleff, c("xsayself", "xsay1",
+   "xsay2", "xsay3", "xsay4", "xsay5"), 0, as.double(NA))
```

where `data(poleff)` has missing values for responses coded as 0 while, `data(poleffna)` has missing values for responses coded as NA.

8. LIST OF FUNCTIONS

Here is a complete list of function available in `anchors`, and help files are available for each of them.

<code>allequal.test</code>	all.equal with expected outcome test
<code>anchors</code>	Non-parametric analysis of surveys with vignette anchors
<code>anchors.intervals</code>	Frequency of intervals in C for subsets of vignettes
<code>anchors.minent</code>	Distribution of C by remapping interval values to scalars by minimum entropy
<code>chopit</code>	Compound Hierarchical Ordered Probit (CHOPIT)
<code>chopit.density</code>	Calculate density of mu from CHOPIT Analysis
<code>cpolr</code>	Censored ordered probit
<code>entropy</code>	Calculate known minimum or estimated entropy for survey vignettes
<code>fitted.cpolr</code>	Conditional and unconditional prediction for censored ordered probit
<code>insert</code>	Insert DIF-corrected variable into original data frame
<code>minimum.entropy</code>	Calculate known minimum entropy for C
<code>plot.anchors.intervals</code>	Plot frequency of intervals in C for subsets of vignettes
<code>plot.entropy</code>	Plot entropy calculations
<code>plot.vignette.order</code>	Plot frequency of vignette orderings
<code>print.anchors.intervals</code>	Frequency of intervals in C for subsets of vignettes
<code>print.summary.anchors</code>	Summary of non-parameteric anchors analysis
<code>replacevalue</code>	Replaces occurrences of a value with another value in set of columns
<code>summary.anchors</code>	Summary of non-parameteric anchors analysis
<code>summary.chopit</code>	Summary of CHOPIT Analysis
<code>summary.entropy</code>	Summary of entropy function
<code>summary.vignette.order</code>	Summary of frequency of vignette orderings
<code>vignette.order</code>	Calculate frequency of vignette orderings

REFERENCES

- King, Gary, C.J.L. Murray, J.A. Salomon, and A. Tandon. 2004. "Enhancing the Validity and Cross-Cultural Comparability of Measurement in Survey Research." *American Political Science Review* 98 (01): 191–207.
- King, Gary, and Jonathan Wand. 2007. "Comparing Incomparable Survey Responses: New Tools for Anchoring Vignettes." *Political Analysis* 15: 46–66.
- Venables, William N., and Brian Ripley. 2006. *MASS*. R package ver. 7.2-31.
URL: <http://www.r-project.org>
- Wand, Jonathan. 2007. "Surveys with Anchoring Vignettes: Theory and Practice." Stanford University, mimeo.
- Wand, Jonathan, Gary King, and Olivia Lau. 2007. "Anchors: Software for Anchoring Vignette Data." *Journal of Statistical Software*. Submitted for review.